

Universidad Carlos III de Madrid

Escuela Politécnica Superior



Tesis Doctoral
Doctorado en Ciencia y Tecnología Informática
Departamento de Informática

Análisis, Optimización y Evaluación de Modelos de Redes de Neuronas Artificiales para la Clasificación y Predicción de Impactos de Alta Velocidad sobre Distintos Materiales

Autor: Israel González Carrasco
Tutores: Dr. Ángel García Crespo
Dra. Belén Ruiz Mezcuca

Leganés, 12 de Febrero de 2010

TESIS DOCTORAL

ANÁLISIS, OPTIMIZACIÓN Y EVALUACIÓN DE MODELOS DE REDES DE NEURONAS ARTIFICIALES PARA LA CLASIFICACIÓN Y PREDICCIÓN DE IMPACTOS DE ALTA VELOCIDAD SOBRE DISTINTOS MATERIALES

Autor: Israel González Carrasco

Director/es: Ángel García Crespo y Belén Ruiz Mezcuca

Firma del Tribunal Calificador:

Firma

Presidente:

Vocal:

Vocal:

Vocal:

Secretario:

Calificación:

Leganés, de de

A Cristina

Agradecimientos

Odisea es quizás la palabra que mejor refleja el trabajo realizado ya que considero el desarrollo de esta tesis doctoral como un largo viaje, de varios años, en los cuales se han sucedido aventuras unas veces favorables y otras adversas. Sin embargo, gracias al apoyo de todas las personas que me han acompañado durante este largo trayecto he podido superar las dificultades que han ido apareciendo. Los siguientes párrafos están escritos en primera persona para reflejar mi más sincero agradecimiento.

En primer lugar, a mis padres por todos aquellos sacrificios que mi formación ha supuesto para ellos. A mi hermana, y a toda mi familia, por el cariño y apoyo moral que siempre me han prestado.

A mis directores, Ángel García y Belén Ruiz, por su confianza durante todos estos años y por su tiempo y dedicación durante el desarrollo de la tesis doctoral.

A Ramón Zaera por su continua disposición para resolver mis dudas. A los alumnos de los que he sido su tutor de proyecto fin de carrera y que han colaborado en esta investigación: Diego, Alejandro y David. A María por su ayuda en la traducción a lengua inglesa de varios pasajes de este documento. A Rafaela, Alberto y María, de la secretaría del Departamento de Informática, por ayudarme a superar todas las barreras administrativas.

A John Breslin y al centro de investigación DERI de Irlanda por acogerme para realizar una estancia de investigación de tres meses con ellos.

A los profesores, Inés María Galván y Álvaro García, por contribuir a la predefensa de esta tesis y aportar puntos de vista y comentarios de gran utilidad. A Luciano García y Genoveva Vargas por su detallada revisión y valiosas observaciones.

A Jorge Pleite y Carlos García, del departamento de Tecnología Electrónica, por su confianza y enseñanzas en el mundo de los transformadores.

A mis compañeros y amigos, Isabel, Ricardo, Lourdes, Rosa, Alberto (Cana), Mayte, David, Jessica, Alberto, Antonio y a todo el grupo SINTONIA por sus consejos y comentarios. A Jose por compartir conmigo las mismas dudas y alegrías, a Fernando por hacer más llevadero el viaje y a Juanmi por su infinita paciencia e innumerables contactos.

A mis amigos de siempre, Amador, Ana, Jesús, Vanesa y Carlos, por empujar la nave con su continuo apoyo. Gracias también a Luis, José, Elsa, Raúl, Miguel, Roberto, Virginia, Ana y a todos los que me han demostrado su sincero interés.

Y finalmente, a Cristina por su optimismo y generosidad, por animarme en los peores momentos y acompañarme por todos los caminos recorridos, ella sin duda es la que más ha sufrido conmigo y la que más valora lo que he conseguido. Esta tesis es en gran medida suya. Gracias por enseñarme que siempre hay luz al final del túnel.

En resumen, a todas aquellas personas que de una u otra forma han colaborado para que Ulises vuelva a casa: Gracias.

Summary

In order to satisfy the requirements for the European doctorate mention, the summary of this phd thesis has been written in English. This summary includes methodology, main contributions and conclusions. The Chapter §6 has also been written in English.

The problems of impact of solids, both low and medium speed, the so-called Crashworthiness, as well as those of high speed, known as ballistic impact, are of great relevance in the field of structural engineering and of advanced materials. In these kind of problems the behavior of materials under extreme conditions of work is analyzed and predicted, allowing the design and manufacture of new components and structures subject to impact. The design of these structures is generally a complex and costly process that requires a high consumption of technical and human resources. Due to the fact that these structures can be essential parts of a system, and to the fact that they can suffer impacts from other bodies during the performance of their activity, it is vital to know how they would behave after suffering the shock of a projectile.

To try to minimize the damage caused by the impact, many efforts have been directed to study the process of penetration of a projectile on a target or goal. Among the proposed alternatives highlight the empirical evidence and, in recent years, the computational techniques for the representation of the process, known as classic or traditional: numerical simulation and analytical modeling. However, these techniques have certain problems that prevent them from being implemented in all possible scenarios. Therefore, during recent years multidisciplinary methodologies have begun to be used in many branches of industrial and structural engineering, for improving and optimizing processes in charge of modeling this phenomenon. Among these multidisciplinary methodologies, the use of predictive and machine learning techniques based on Artificial Neural Networks (ANN) stands out.

This research has been defined as an integral framework in which different components work together for the solution and optimization of the engineering problem of impact of solids:

- The numerical simulation tools and the experimentation of ballistic tests have been the first components used, being two crucial elements as they serve as a link with the Multilayer Perceptron (MLP) models designed. The numerical simulation, has allowed from the analysis of the physical phenomenon of the problem, and based on a series of mathematical models, to recreate by means of numerical simulations, the material's characteristics and the behavior of the objects involved. Thus, thanks to its use a series of missile tests aimed to train and validate the MLP have been simulated. Empirical tests conducted in the laboratory have been useful for checking the plausibility of the numerical simulation models with reality.

- The next component corresponds to the techniques of Artificial Intelligence (AI) based on ANN, more specifically the MLP architecture. Its main objective has been to replace the former components to recreate the impact process of solids at high speed. These ANN models have been used to solve three of the most representative scenarios, i.e. to know the impact's result and to study the behavior of objects, proving its validity as an alternative solution to traditional techniques.
- Optimization of neural models has been carried out by defining an optimization methodology based on a series of activities and tasks aimed at fulfilling the targets and hypothesis outlined at the beginning of this doctoral thesis. To homogenize the large number of existing alternatives and to simplify its implementation on each of the scenarios, these are grouped into three large blocks or phases: patterns, architecture and algorithm. The variants that according to the literature best adapt to the problem treated, have been included and tested in each of them, only taking those that have shown better results.

The scope of this research is accordingly multidisciplinary, due to the fact that the theoretical basis of the problem comes from the engineering branch of solid mechanics and IA-based computing techniques will be implemented for its resolution. Thus, the solution of the problem of ballistic impact discussed in this research requires the collaboration of heterogeneous techniques coming from different branches of science and engineering in search of a common benefit. Consequently, this doctoral thesis provides a methodology of optimization of neural models based on MLP architecture for different scenarios of ballistic impact.

The research methodology followed in this doctoral thesis to achieve the objectives and demonstrate the hypotheses initially set out has consisted of:

1. State of the art. The main objective of this activity is to detail the works performed with ANN within the field of impact of solids. The main issues addressed in this section will be:
 - Definition of the problem to be solved. This will involve a detailed description of the phenomenon of penetration treated, focusing on its key elements: projectile and protection.
 - Physical description of the problem, focused on the physics of the materials involved and the dynamics of the ballistic phenomenon.
 - Critical analysis of the main classic alternatives for solving problems of impact.
 - Identification of the main parameters that influence on the predictive ability of the ANN within the domain treated.
 - Precedents of the implementation of ANN-based solutions within the domain studied.

2. Numerical simulation of the tests. The behavior of projectiles and protections in the impacts has been recreated by means of numerical simulation tools. It has been generated a set of cases of impact with different values for the descriptive parameters of each of the structures, projectile and protection, which take part in the problem.
3. Development of new models of ANN. The work to be carried out by the author in this section will involve creating new ANN designs in order to solve two variants of the problem of impact: result's classification and prediction of the final velocity and residual mass.
4. Simulation of the new ANN designed models. The ANN will be tested for different geometries, both projectile and protector panel, velocities and materials.
5. Analysis of possible alternatives for the optimization of learning and results. There are many parameters of an ANN, both of the architecture and of the training algorithm, which modify its generalization capacity and therefore affect its accuracy of prediction. The search for optimal values for these parameters is a point of importance in the work to be carried out as it will allow optimizing learning and performance.
6. Evaluation and Validation. The assessment and validation activities pretend to estimate the effectiveness of different ANN solutions raised and the validity of the optimization methodology proposed.
7. Documentation and Conclusions. In this activity each aspect of the thesis will be documented. The different alternatives of ANN will be described as well as their contributions to the results obtained. Finally, the conclusions drawn from the evaluation will be exposed.

The evaluation of the neural models has demonstrated the viability of the MLP to solve different ballistic scenarios. Thanks to its use, traditional simulation or modeling tools can be used as a complement to the MLP to improve the design of the protection systems. The use of an optimized MLP allows predicting both the result of an impact and the behavior of the objects which take part with a higher success rate than the other proposals evaluated. To confirm these conclusions, the MLP has been trained and tested under various combinations of geometries and materials for the projectile and the protection, and different impact velocities.

The advantage of using a MLP to solve the different scenarios is that the results are obtained in real time and the computational cost is thus much lower than the one corresponding to the traditional techniques of simulation and modeling. Furthermore the comparison with other proposals of learning machines has shown that the MLP is the best alternative for the different ballistic scenarios set out. Besides the optimization methodology aimed at the systematization of the different methods and techniques existing when it comes to configuring an ANN, has been validated for different ballistic

scenarios.

In summary, the conclusions obtained show that thanks to this research new ballistic tests can be recreated in the future in a reliable way. This will facilitate the subsequent design of protections and study of the consequences associated with the introduction of different configurations of projectiles and protective panels with different characteristics. Finally, the findings obtained by using in the first place AI-based solutions and in the second place the application of an optimization methodology can be extrapolated to other areas such as the design of structures subject to low-speed impacts or the passive safety of vehicles.

Resumen

A fin de cumplir los requisitos necesarios para la obtención de la mención europea del doctorado, el resumen de esta tesis fue originalmente escrito en inglés. A continuación se incluye la versión en lengua española.

Los problemas de impacto sobre estructuras tanto de baja y media velocidad, denominados *Crashworthiness*, como de alta velocidad, conocidos como impacto balístico, son de gran relevancia en el ámbito de la ingeniería estructural y de materiales avanzados. En ellos, se analiza y predice el comportamiento de materiales en condiciones extremas de trabajo, lo que permite el diseño y fabricación de nuevos componentes y estructuras sujetas a impactos. El diseño de estas estructuras es por lo general un proceso complejo y costoso, que requiere un alto consumo de recursos técnicos y humanos. Debido a que estas estructuras pueden ser partes esenciales de un sistema, y a que pueden sufrir durante el desempeño de su actividad impactos de otros cuerpos, es vital conocer como se comportarían tras sufrir el choque de un proyectil.

Para intentar minimizar los daños producidos por el impacto, se han destinado muchos esfuerzos a estudiar el proceso de penetración de un proyectil sobre un blanco u objetivo. Entre las alternativas clásicas empleadas por los científicos destacan, en primer lugar las pruebas empíricas, y durante las últimas décadas las técnicas computacionales para la representación del proceso, conocidas como clásicas o tradicionales: simulación numérica y modelado analítico. No obstante, estas técnicas presentan ciertos problemas que impiden su aplicación en todos los escenarios posibles. Por ello, durante los últimos años ha comenzado a utilizarse, en numerosas ramas de la ingeniería industrial y estructural, metodologías multidisciplinares para mejorar y optimizar los procesos encargados de modelar este fenómeno. Entre ellas destaca el empleo de técnicas predictivas y de aprendizaje automático basadas en Redes Neuronales Artificiales (RNA).

Esta investigación ha sido definida como una estructura integral de trabajo donde colaboran diferentes componentes para la resolución y optimización del problema ingenieril del impacto de sólidos:

- Las herramientas de simulación numérica y la experimentación de ensayos balísticos han sido los primeros componentes empleados. Se trata de dos elementos cruciales al servir de nexo de unión con los modelos de Perceptrón MultiCapa (PMC) diseñados. La simulación numérica ha permitido a partir del análisis del fenómeno físico del problema, y basándose en una serie de modelos matemáticos, recrear mediante simulaciones numéricas las características de los materiales y el comportamiento de los objetos que intervienen. Por tanto gracias a su empleo se han simulado una serie de ensayos balísticos encaminados a entrenar y validar los PMC. Las pruebas empíricas realizadas en laboratorio han servido para comprobar la verosimilitud de

los modelos de simulación numérica con la realidad.

- El siguiente componente se corresponde con las técnicas de Inteligencia Artificial (IA) basadas en RNA, más concretamente de la arquitectura PMC. Su objetivo fundamental ha sido sustituir a los componentes anteriores para recrear el proceso de impacto de sólidos a alta velocidad. Estos modelos de RNA han sido empleados para resolver tres de los escenarios más representativos, i.e. conocer el resultado del impacto y estudiar el comportamiento de los objetos, demostrando su validez como solución alternativa a las técnicas tradicionales.
- La optimización de los modelos neuronales se ha llevado a cabo mediante la definición de una metodología de optimización, basada en una serie de actividades y tareas, orientadas a cumplir los objetivos e hipótesis expuestas al comienzo de esta tesis doctoral. Para homogeneizar el gran número de alternativas existentes y simplificar su aplicación sobre cada uno de los escenarios, éstas se han agrupado en tres grandes bloques o fases: patrones, arquitectura y algoritmo. En cada uno de ellos se han incluido y probado las variantes que según la literatura mejor se adaptan al problema tratado, tomándose únicamente aquellas que mejores resultados han mostrado.

El ámbito de esta investigación es en consecuencia multidisciplinar, ya que la base teórica del problema a resolver proviene de la rama ingenieril de la mecánica de sólidos y para su resolución se aplicarán técnicas informáticas basadas en IA. De esta forma, la resolución del problema del impacto balístico, tratado en esta investigación, requiere de la colaboración de técnicas heterogéneas provenientes de diferentes ramas de la ciencia y la ingeniería en busca de un beneficio común. Como elemento final, la presente tesis doctoral aporta una metodología de optimización de modelos neuronales basados en la arquitectura PMC para diferentes escenarios del impacto balístico.

La metodología de investigación seguida en esta tesis doctoral para alcanzar los objetivos y demostrar las hipótesis planteadas inicialmente ha consistido en:

1. **Estado del arte.** El principal objetivo de esta actividad se ha centrado en detallar los trabajos realizados con RNA dentro del ámbito del impacto de sólidos. Las principales cuestiones tratadas en este apartado han sido:
 - Definición del problema a resolver, centrándose en los elementos fundamentales del mismo: proyectil y protección.
 - Descripción física del problema, centrada en la física de los materiales que intervienen y la dinámica del fenómeno balístico.
 - Análisis crítico de las principales alternativas clásicas para la resolución de problemas de impacto.
 - Identificación de los principales parámetros que influyen en la capacidad de predicción de las RNA dentro del dominio tratado.

- Antecedentes de la aplicación de soluciones basadas en RNA dentro del dominio estudiado.
2. **Simulación numérica de los ensayos.** El comportamiento de los proyectiles y de las protecciones ante los impactos se ha recreado mediante herramientas de simulación numérica. Se ha generado un conjunto de casos de impacto con diferentes valores de los parámetros descriptivos de cada una de las estructuras, proyectil y protección, que intervienen en el problema.
 3. **Desarrollo de nuevos modelos de RNA.** El trabajo realizado por el autor en este apartado ha consistido en crear nuevos diseños de RNA para solventar dos variantes del problema del impacto: clasificación del resultado y predicción de la velocidad final y masa residual.
 4. **Simulación de los nuevos modelos de RNA diseñados.** Las RNA han sido probadas para distintas geometrías, tanto de proyectil como de panel protector, velocidades y materiales.
 5. **Análisis de posibles alternativas para la optimización del aprendizaje y de los resultados.** Existen múltiples parámetros de una RNA, tanto de la arquitectura como del algoritmo de entrenamiento y de los datos empleados para extraer el conocimiento del dominio, que modifican su capacidad de generalización y que afectan por tanto a su precisión de predicción. La búsqueda de valores óptimos para estos parámetros, es un punto de relevancia dentro del trabajo ya que ha permitido optimizar tanto el aprendizaje como los resultados.
 6. **Evaluación y Validación.** Las actividades de evaluación y validación incluidas han determinado la efectividad de las diferentes soluciones de RNA planteadas y demostrado la validez de la metodología de optimización propuesta.
 7. **Documentación y Conclusiones.** Esta actividad ha consistido en la documentación de cada uno de los aspectos relativos a la tesis doctoral y presenta las conclusiones extraídas a raíz de la investigación.

La evaluación de los modelos neuronales ha demostrado la viabilidad del PMC para resolver diferentes escenarios balísticos. Gracias a su utilización las herramientas tradicionales de simulación o modelado pueden utilizarse como complemento al PMC para mejorar el diseño de sistemas de protección. El empleo de un PMC optimizado permite predecir, tanto el resultado de un impacto, como el comportamiento de los objetos que intervienen, con un porcentaje de acierto superior al resto de propuestas evaluadas. Para ratificar estas conclusiones, el PMC ha sido entrenado y probado bajo diversas combinaciones geométricas y de materiales para el proyectil y la protección, y diferentes velocidades de impacto.

La ventaja de emplear un PMC para resolver los diferentes escenarios es que los resultados se obtienen en tiempo real y por tanto el coste computacional es muy

inferior al de las técnicas tradiciones de simulación y modelado. Además la comparación realizada con otras propuestas de maquinas de aprendizaje ha demostrado que un PMC, correctamente configurado, es la mejor alternativa para los diferentes escenarios balísticos planteados. Asimismo la metodología de optimización, orientada a la sistematización de los diferentes métodos y técnicas existentes a la hora de configurar y parametrizar una RNA, ha sido validada para diferentes escenarios balísticos.

En consecuencia, gracias a las conclusiones obtenidas en esta investigación se podrán recrear en el futuro de una manera fiable nuevos ensayos balísticos. Esto facilitará el posterior diseño de protecciones y estudio de las consecuencias asociadas a la introducción de diferentes configuraciones de proyectiles y paneles protectores con diferentes características. Por último, las deducciones obtenidas al emplear en primer lugar soluciones basadas en IA y en segundo lugar la aplicación de una metodología de optimización pueden extrapolarse a otras áreas como el diseño de estructuras sujetas a impactos de baja velocidad o la seguridad pasiva de vehículos.

Índice general

1. Introducción	1
1.1. Contexto	1
1.2. Breve descripción del problema	3
1.3. Motivación	4
1.4. Objetivos	6
1.4.1. Objetivo general	6
1.4.2. Objetivos específicos	6
1.5. Hipótesis	8
1.6. Justificación	9
1.7. Metodología de la investigación	12
1.8. Estructura del documento	14
1.9. Convenciones	16
2. Estado del Arte	17
2.1. Impactos de proyectiles	17
2.1.1. Interés del problema	17
2.1.2. Impacto balístico	19
2.1.2.1. Nociones de impacto balístico	21
2.1.2.2. Variables relativas al impacto balístico	22
2.1.2.3. Espesor de los blindajes	24
2.1.2.4. Composición de los blindajes	25
2.1.2.5. Material de los blindajes	25
2.1.3. Proceso de penetración	26
2.1.3.1. Modos de fallo	28
2.1.4. El blindaje ligero en la actualidad	29

2.1.5.	Metodologías clásicas para modelar el problema	31
2.1.5.1.	Simulación Numérica	33
2.1.5.2.	Método de Elementos Finitos	35
2.2.	Redes de Neuronas Artificiales	36
2.2.1.	Modelos biológicos	36
2.2.2.	Redes de Neuronas Artificiales e Inteligencia Artificial	38
2.2.3.	Definición de una Red de Neuronas Artificial	40
2.2.3.1.	Modelo matemático	41
2.2.4.	Clasificación de una Red de Neuronas Artificial	45
2.2.5.	Proceso de aprendizaje	45
2.2.5.1.	Criterio de convergencia	47
2.2.5.2.	Otros criterios de parada	49
2.2.6.	Ventajas e inconvenientes de las Redes de Neuronas Artificiales	51
2.3.	Familia del Perceptrón	53
2.3.1.	Perceptrón Multicapa	53
2.3.2.	Arquitectura del Perceptrón Multicapa	54
2.3.2.1.	Función de activación	57
2.3.3.	Aprendizaje del Perceptrón Multicapa	57
2.3.3.1.	Aplicación del algoritmo RP	58
2.3.3.2.	Modos de entrenamiento	60
2.3.3.3.	Tasa de aprendizaje y constante <i>momentum</i>	61
2.3.3.4.	Criterios de parada	62
2.3.4.	Conjuntos de entrenamiento y validación	63
2.4.	Técnicas para la optimización	65
2.4.1.	Importancia de la optimización	65
2.4.2.	Tratamiento de los patrones	67
2.4.2.1.	Cantidad de patrones necesarios	67
2.4.2.2.	Maximizar la información disponible	69
2.4.2.3.	Pre-procesado de la información	71
2.4.2.4.	Distribución de patrones	73
2.4.3.	Arquitectura de la red	73
2.4.3.1.	Capas y neuronas ocultas	75

2.4.3.2.	Crecimiento y poda de la red	77
2.4.3.3.	Selección de las variables descriptivas	79
2.4.3.4.	Funciones de activación	86
2.4.3.5.	Inicialización de los pesos de red	87
2.4.3.6.	Comité de redes	88
2.4.4.	Algoritmo de aprendizaje	90
2.4.4.1.	Parámetros del algoritmo de aprendizaje	91
2.4.4.2.	Técnicas para la resolución de problemas en el algoritmo de RP	93
2.4.4.3.	Criterio de parada	94
2.4.4.4.	Modos de entrenamiento	99
2.4.5.	Resumen de reglas y parámetros de optimización	100
2.5.	Aplicaciones de las RNA	100
2.5.1.	Redes de Neuronas Artificiales en ingeniería	100
2.5.2.	Redes de Neuronas Artificiales para la resolución de problemas en ingeniería industrial	103
2.5.3.	El problema del impacto y las Redes de Neuronas Artificiales	105
2.6.	Algoritmos genéticos	107
2.6.1.	Parámetros del Algoritmo Genético	108
2.6.2.	Codificación de las variables	109
2.6.3.	Selección	109
2.6.4.	<i>Crossover</i> o cruce	110
2.6.5.	Mutación	110
2.7.	Alternativas basadas en el aprendizaje automático	111
2.8.	Sumario	112
2.9.	Discusión	113
3.	Solución Propuesta	117
3.1.	Introducción	117
3.2.	Marco integral de la investigación	118
3.2.1.	Introducción	118
3.2.2.	Ubicación de la investigación	121
3.3.	Planteamiento del problema	123

3.3.1.	Resultado del impacto	126
3.3.2.	Comportamiento del proyectil y del panel protector	127
3.4.	Simulación Numérica	128
3.4.1.	Introducción	128
3.4.2.	Justificación de la necesidad de la simulación numérica	130
3.4.3.	Ensayos experimentales	132
3.4.4.	Simulación numérica realizada	134
3.4.5.	Materiales metálicos empleados	138
3.4.6.	Validación del código de simulación	141
3.4.7.	Ensayos balísticos obtenidos	141
3.5.	Modelos Neuronales	145
3.5.1.	Arquitectura de red	146
3.5.1.1.	Arquitecturas neuronales en clasificación	149
3.5.1.2.	Arquitecturas neuronales en regresión	152
3.5.2.	Patrones de entrada	154
3.5.3.	Algoritmo de entrenamiento de la red	158
3.5.4.	Herramienta para la configuración y automatización de las simulaciones	160
3.6.	Alternativas basadas en el aprendizaje automático	162
3.7.	Sumario	163
4.	Metodología de Optimización	165
4.1.	Introducción	165
4.2.	Fase Previa. Comparativa	166
4.3.	Fase 1. Tratamiento de los patrones	170
4.3.1.	Tarea 1.1: Normalización de los patrones	171
4.3.2.	Tarea 1.2: Distribución de los patrones	174
4.3.3.	Tarea 1.3: Técnicas para trabajar con datos limitados	175
4.3.3.1.	Validación Cruzada <i>Hold Out</i> (HOCV)	177
4.3.3.2.	Validación Cruzada <i>K-Fold</i> (KFCV)	178
4.3.3.3.	Validación Cruzada <i>Leave-One-Out</i> (LOOCV)	179
4.3.3.4.	Remuestreo <i>Bootstrap</i> (RBP)	181

4.3.3.5.	Variante Iterativa Equilibrada de Validación Cruzada (VIECV)	183
4.3.3.6.	Incorporación de ruido a los datos	185
4.3.3.7.	Resumen de las técnicas	188
4.4.	Fase 2: Arquitectura de la red	188
4.4.1.	Tarea 2.1: Inicialización los pesos de la red	190
4.4.2.	Tarea 2.2: Elección de la función de activación	191
4.4.3.	Tarea 2.3: Selección de la arquitectura óptima	195
4.4.4.	Tarea 2.4: Selección del vector de entradas óptimo	196
4.5.	Fase 3: Algoritmo de entrenamiento	199
4.5.1.	Tarea 3.1: Criterio de convergencia	201
4.5.2.	Tarea 3.2: Modos de entrenamiento	202
4.5.3.	Tarea 3.3: Heurísticas asociadas a los parámetros del algoritmo de entrenamiento	203
4.5.4.	Tarea 3.4: Optimización numérica de algoritmo del entrenamiento	205
4.5.4.1.	Función de coste	207
4.5.4.2.	Método del Gradiente	208
4.5.4.3.	Aprendizaje con momentum	208
4.5.4.4.	Regla Delta-Bar Extendida	208
4.5.4.5.	Método del Gradiente Conjugado	209
4.5.4.6.	Métodos de segundo orden	210
4.5.4.6.1.	Algoritmo Quick Propagation	211
4.5.4.6.2.	Algoritmo Levenberg-Marquardt	211
4.6.	Fase 4: Medidas de calidad	212
4.6.1.	Tarea 4.1: Indicadores de la teoría de la información	213
4.6.2.	Tarea 4.2: Test estadístico	215
4.7.	Algoritmo genético	224
4.8.	Sumario	225
5.	Evaluación y Validación	227
5.1.	Introducción	227
5.2.	Problema de clasificación	229
5.2.1.	Modelo neuronal base	229

5.2.2.	Fase Previa. Comparativa	230
5.2.2.1.	Medidas de calidad de los clasificadores	231
5.2.3.	Fase 1. Tratamiento de los patrones	237
5.2.3.1.	Tarea 1.1: Normalización de los datos	237
5.2.3.1.1.	Medidas de calidad de los resultados	239
5.2.3.2.	Tarea 1.2: Distribución de los patrones	240
5.2.3.3.	Tarea 1.3: Técnicas para trabajar con datos limitados . . .	245
5.2.3.3.1.	Tarea 1.3.1. Parada del entrenamiento por <i>epochs</i>	246
5.2.3.3.2.	Tarea 1.3.2. Parada del entrenamiento por MSE .	248
5.2.3.3.3.	Tarea 1.3.3. Parada del entrenamiento por error de validación (parada temprana)	251
5.2.3.3.4.	Medidas de calidad de los resultados	253
5.2.3.4.	Sumario	256
5.2.4.	Fase 2: Arquitectura de la red	256
5.2.4.1.	Tarea 2.1. Inicialización los pesos de la red	257
5.2.4.1.1.	Medidas de calidad de los resultados	258
5.2.4.2.	Tarea 2.2: Elección de la función de activación	261
5.2.4.2.1.	Medidas de calidad de los resultados	262
5.2.4.3.	Tarea 2.3: Selección de la arquitectura óptima	262
5.2.4.3.1.	Medidas de calidad de los resultados	264
5.2.4.4.	Tarea 2.4: Selección del vector de entradas óptimo	266
5.2.4.4.1.	Medidas de calidad de los resultados	269
5.2.4.5.	Sumario	270
5.2.5.	Fase 3: Algoritmo de entrenamiento	270
5.2.5.1.	Tarea 3.1. Criterio de convergencia	271
5.2.5.2.	Tarea 3.2: Modos de entrenamiento	272
5.2.5.2.1.	Medidas de calidad de los resultados	273
5.2.5.3.	Tarea 3.3: Heurísticas asociadas a los parámetros del algoritmo de aprendizaje	274
5.2.5.3.1.	Medidas de calidad de los resultados	275
5.2.5.4.	Tarea 3.4: Optimización numérica del algoritmo de aprendizaje	278
5.2.5.4.1.	Tarea 3.4.1: Funciones de coste	279

5.2.5.4.2.	Medidas de calidad de los resultados	281
5.2.5.4.3.	Tarea 3.4.2: Algoritmos de aprendizaje	284
5.2.5.4.4.	Medidas de calidad de los resultados	285
5.2.5.5.	Comparativa Tarea 3.3 y Tarea 3.4	288
5.2.5.6.	Sumario	289
5.2.6.	Fase 4. Medidas de calidad	290
5.2.6.1.	Tarea 4.1: Indicadores de la teoría de la información . . .	290
5.2.6.2.	Tarea 4.2: Test estadístico	291
5.2.7.	Sumario final para el problema de clasificación	292
5.3.	Problema de regresión	294
5.3.1.	Modelo neuronal base	294
5.3.2.	Fase Previa. Comparativa	296
5.3.2.1.	Escenario uno: Perforación de la placa	297
5.3.2.1.1.	Medidas de calidad de las alternativas	297
5.3.2.2.	Escenario dos: Detención del proyectil	302
5.3.2.2.1.	Medidas de calidad de las alternativas	302
5.3.3.	Fase 1. Tratamiento de los patrones	305
5.3.4.	Fase 2. Arquitectura de la red	306
5.3.5.	Fase 3. Algoritmo de entrenamiento	307
5.3.6.	Sumario final para el problema de regresión	308
5.4.	Sumario	309
6.	Conclusions and Future Research	311
6.1.	Future Research	315
6.2.	Related Publications	316
7.	Conclusiones y Trabajos Futuros	319
7.1.	Futuras líneas de investigación	324
7.2.	Publicaciones realizadas	325
	Apéndices	325
	A. Abreviaturas	327

B. Listado de ensayos balísticos	329
B.1. Listado de ensayos balísticos	330
B.2. Listado de ensayos balísticos con ruido	338

Índice de figuras

1.1. Secuencia del impacto de un proyectil	3
1.2. Objetivos fijados para esta investigación	6
1.3. Pasos necesarios para completar la simulación del problema mediante RNA	12
1.4. Método de trabajo diseñado para la tesis doctoral	15
2.1. Definiciones de impacto balístico según los diferentes autores	19
2.2. Ángulo de impacto	23
2.3. Posibles resultados del impacto de un proyectil sobre una protección	27
2.4. Modos de fallo más comunes en placas impactadas	28
2.5. Curva de probabilidad de perforación frente a la velocidad de impacto para un proyectil-protección determinado	32
2.6. Esquema de una neurona biológica	36
2.7. Áreas de interés relacionadas con las RNA	38
2.8. Unidad de Proceso Elemental o Neurona Artificial	43
2.9. Funciones de activación típicas	44
2.10. Representación de la superficie del error	48
2.11. Evolución estacionaria del error	50
2.12. Clases separables linealmente y Clases no separables linealmente	55
2.13. Arquitectura de un PMC	56
2.14. Aprendizaje supervisado en un PMC	59
2.15. Evoluciones del error para los conjuntos de entrenamiento y validación durante el proceso de entrenamiento	64
2.16. Estimación del error usando la alternativa KFCV	70
2.17. Diferentes categorías para cuantificar la contribución de las variables de entrada sobre los modelos neuronales	82
2.18. Comité de RNA	89

2.19. Evolución de los errores de entrenamiento y validación	98
2.20. Aplicaciones de las RNA en en el dominio ingenieril	105
3.1. Estructura global del trabajo, donde se detallan los componentes fundamentales y las interacciones entre los mismos	119
3.2. Disciplinas implicadas en esta tesis doctoral	121
3.3. Esquema del impacto del proyectil sobre la protección	124
3.4. Relación entre simulaciones numéricas, teorías y experimentos	128
3.5. Procedimiento para realizar una simulación numérica	130
3.6. Relaciones entre la simulación numérica, experimentación y RNA	132
3.7. Dos imágenes superpuestas de un proyectil esférico tras perforar un panel de material compuesto con velocidad de salida de 480 m/s	133
3.8. Sistemas de cámaras digitales ultrarápidas	134
3.9. Útil de sujeción de paneles instalado dentro de la cámara principal del cañón de gas	135
3.10. Simulación mediante el código ABAQUS/Explicit del impacto oblicuo de un proyectil de acero sobre una placa metálica de aluminio	137
3.11. Proyectil antes (izquierda) y después del impacto (derecha)	139
3.12. Secuencia de la simulación de un caso de perforación: vistas anteriores (izquierda y centro) y vista final posterior (derecha)	140
3.13. Proyectil de aluminio contra placa de uranio a velocidad alta (izquierda) y Proyectil de acero contra placa de uranio a velocidad media (derecha) . . .	140
3.14. Diagramas de caja asociados al conjunto de variables del problema	144
3.15. Diagrama de bloques con las fases clave operacionales utilizadas para cada modelo neuronal	145
3.16. Pasos para la definición de la arquitectura básica de una RNA	146
3.17. Estructura inicial de la RNA diseñada para el problema de clasificación . .	150
3.18. Estructuras iniciales de la RNA diseñadas para el problema de regresión en los escenarios de perforación y detención	152
3.19. Arquitectura del <i>framework</i> desarrollado para automatizar las simulaciones	160
3.20. Detalles de la interfaz de usuario (arriba tratamiento de los datos y abajo entrenamiento iterativo)	162
4.1. Pasos definidos para la metodología de optimización	167
4.2. Detalle de la Fase Previa al proceso de optimización	168

4.3.	Detalle del proceso de optimización asociado a la Fase 1: Tratamiento de los patrones	171
4.4.	Ubicación de la capa de procesado (pre y post-procesado) de los ensayos dentro del marco de trabajo	173
4.5.	Relación entre el tamaño del conjunto de <i>train</i> y la capacidad de predicción de un modelo neuronal	176
4.6.	Proceso de creación de conjuntos con HOCV	178
4.7.	Proceso de selección aleatoria para k subconjuntos con KFCV y parada temprana	180
4.8.	Proceso de selección aleatoria con LOOCV	181
4.9.	Proceso de selección aleatoria con <i>bootstrapping</i>	183
4.10.	Proceso de selección aleatoria con <i>VIECV</i>	184
4.11.	Proceso de incorporación de ruido a los patrones de entrenamiento	185
4.12.	Detalle del proceso de optimización asociado a la Fase 2: Arquitectura de la red	190
4.13.	Funciones de activación disponibles para su empleo sobre las neuronas de las capas ocultas y salida	192
4.14.	Utilización de los indicadores de sensibilidad para la optimización del vector de entradas	199
4.15.	Detalle del proceso de optimización asociado a la Fase 3: Algoritmo de entrenamiento	201
4.16.	Clasificación de los diferentes métodos de optimización de la función del error utilizados	206
4.17.	Detalle del proceso de optimización asociado a la Fase 4: Medidas de calidad	213
4.18.	Esquema de los test de contraste de hipótesis realizados durante la fase de validación de los resultados	223
4.19.	Ubicación del componente de AG dentro de la metodología de optimización	224
5.1.	Fase Previa. Comparación entre los diferentes clasificadores con diferentes criterios de parada (20 repeticiones)	232
5.2.	Fase Previa. Diagramas de dispersión y de caja	233
5.3.	Fase Previa. Análisis de los residuos y la media	234
5.4.	Tarea 1.1. Diagramas de histogramas y de caja	239
5.5.	Tarea 1.2. Evolución del porcentaje medio de acierto del clasificador en función del número de datos para entrenar para #1 y #2 (20 repeticiones)	242

5.6. Tarea 1.2. Evolución del porcentaje medio de acierto del clasificador en función del número de datos para entrenar para #3 (20 repeticiones) . . .	243
5.7. Tarea 1.3.1 (criterio #1). (arriba) Desviación de los diferentes métodos sobre HOCVE (en porcentaje de acierto) y (abajo) resultados obtenidos para los mejores candidatos de cada método (en porcentaje de acierto) (20 repeticiones)	247
5.8. Tarea 1.3.1 (criterio #1). Media de la desviación obtenida (en porcentaje de acierto) usando ruido Gaussiano y ruido Uniforme sobre HOCVE (20 repeticiones)	248
5.9. Tarea 1.3.2 (criterio #2). (arriba) Desviación de los diferentes métodos sobre HOCVE (en porcentaje de acierto) y (abajo) resultados obtenidos para los mejores candidatos de cada método (en porcentaje de acierto) (20 repeticiones)	249
5.10. Tarea 1.3.2 (criterio #2). Media de la desviación obtenida (en porcentaje de acierto) usando ruido Gaussiano y ruido Uniforme sobre HOCVE (20 repeticiones)	250
5.11. Tarea 1.3.3 (criterio #3). (arriba) Desviación de los diferentes métodos sobre HOCVE (en porcentaje de acierto) y (abajo) resultados obtenidos para los mejores candidatos de cada método (en porcentaje de acierto) (20 repeticiones)	252
5.12. Tarea 1.3.3 (criterio #3). Media de la desviación obtenida (en porcentaje de acierto) usando ruido Gaussiano y ruido Uniforme sobre HOCVE (20 repeticiones)	253
5.13. Tarea 1.3.3. Diagrama de caja y dispersión	255
5.14. Tarea 2.1. Diagramas de histogramas y de caja	258
5.15. Tarea 2.3. Promedio del error MSE para las diferentes arquitecturas	265
5.16. Tarea 2.4 (arriba) Influencia de cada variable sobre las salidas usando PSA (abajo izquierda) Sensibilidad media usando SAM y (abajo derecha) Indicador de criticidad de usando IVC.	266
5.17. Tarea 2.4. Estudio comparativo del incremento del error utilizando SWM (20 repeticiones)	268
5.18. Tarea 3.2. Diagramas de histogramas y de caja	273
5.19. Tarea 3.3. Diagramas de dispersión y caja	276
5.20. Tarea 3.3. Análisis de los residuos y la media	277
5.21. Tarea 3.4.1. Diagramas de dispersión y caja	281
5.22. Tarea 3.4.1. Análisis de los residuos y la media	282
5.23. Tarea 3.4.2. Diagramas de dispersión y de caja	286
5.24. Tarea 3.4.2. Análisis de los residuos y la media	287

5.25. Resumen de tareas utilizadas dentro de la metodología	295
5.26. Fase Previa. Diagramas de caja para los escenarios de perforación	298
5.27. Fase Previa. Análisis de la media para los escenarios de perforación	299
5.28. Fase Previa. Diagramas de caja y análisis de la media para el escenario de detención	303

Índice de tablas

2.1. Características de la IA clásica frente a las RNA	39
2.2. Rangos más utilizados en la tasa de aprendizaje	93
2.3. Rangos más utilizados en la constante <i>momentum</i>	93
2.4. Efectos de utilizar valores extremos en los parámetros de diseño	101
3.1. Variables descriptivas del problema del impacto	142
3.2. Variables de salida para los modelos diseñados	142
4.1. Fase Previa. Descripción de los clasificadores utilizados	169
4.2. Procedimientos para generar vectores artificiales con ruido	186
4.3. Características más interesantes de las técnicas analizadas para trabajar con pocos datos	189
4.4. Funciones de activación utilizadas	194
4.5. Valores para $t_{N,v}$ para intervalos de confianza de dos colas	219
4.6. Tipos de error estadísticos	219
4.7. Parámetros y criterios empleados en el AG	225
5.1. Resumen de características de la red base para clasificación	229
5.2. Resultados preliminares obtenidos para el PMC base (% acierto tras 20 repeticiones)	230
5.3. Fase Previa. Análisis aplicando la Tarea 4.1: Indicadores AIC y MDL . . .	231
5.4. Fase Previa. Análisis aplicando la Tarea 4.2: Contraste de la varianza . . .	235
5.5. Fase Previa. Análisis aplicando la Tarea 4.2: Método ANOVA	235
5.6. Fase Previa. Análisis mediante la Tarea 4.2: Test CRM	236
5.7. Fase Previa. Análisis aplicando la Tarea 4.2: Grupos homogéneos	237
5.8. Tarea 1.1. Resultados obtenidos por criterio de parada (20 repeticiones) . .	238

5.9. Tarea 1.1 Análisis estadístico para el criterio de parada temprana (20 repeticiones)	238
5.10. Tarea 1.2. Valores aproximados por el modelo matemático	244
5.11. Tarea 1.3. Detalle de los experimentos incluidos	245
5.12. Tarea 1.3.3. Análisis aplicando la Tarea 4.2: Contraste de la varianza . . .	254
5.13. Tarea 1.3.3. Análisis aplicando la Tarea 4.2: Test de Krustal Wallis	254
5.14. Tarea 2.1 Análisis estadístico (20 repeticiones)	259
5.15. Tarea 2.2. Resultados obtenidos para cada topología (20 repeticiones). . . .	261
5.16. Tarea 2.3. Mejores topologías para una y dos capas utilizando el AG	263
5.17. Tarea 2.3. Resultados obtenidos para cada topología con el proceso <i>batch</i> (20 repeticiones)	264
5.18. Tarea 2.3. Análisis aplicando la Tarea 4.1: Indicadores AIC y MDL	264
5.19. Tarea 2.4. Mejor vector de entrada localizado utilizando el AG	269
5.20. Tarea 2.4. Análisis aplicando la Tarea 4.1: Indicadores AIC y MDL	269
5.21. Tarea 3.2 Análisis estadístico (20 repeticiones)	272
5.22. Tarea 3.3. Mejores valores de η y μ utilizando el AG	275
5.23. Tarea 3.3. Resultados obtenidos utilizando diferentes heurísticas (20 repeticiones)	275
5.24. Tarea 3.3. Análisis aplicando la Tarea 4.2: Contraste de la varianza	278
5.25. Tarea 3.3. Análisis aplicando la Tarea 4.2: Método ANOVA	278
5.26. Tarea 3.3. Análisis aplicando la Tarea 4.2: Test CRM	278
5.27. Tarea 3.3. Análisis aplicando la Tarea 4.2: Grupos homogéneos	279
5.28. Tarea 3.4. Listado de propuestas para la mejora del algoritmo de aprendizaje	279
5.29. Tarea 3.4.1. Resultados obtenidos utilizando diferentes funciones de coste (20 repeticiones)	280
5.30. Tarea 3.4.1. Análisis aplicando la Tarea 4.2: Contraste de la varianza . . .	280
5.31. Tarea 3.4.1. Análisis aplicando la Tarea 4.2: Método ANOVA	281
5.32. Tarea 3.4.1. Análisis aplicando la Tarea 4.2: Test CRM	283
5.33. Tarea 3.4.1. Análisis aplicando la Tarea 4.2: Grupos homogéneos	284
5.34. Tarea 3.4.2. Resultados obtenidos utilizando diferentes algoritmos de aprendizaje (20 repeticiones)	285
5.35. Tarea 3.4.2. Análisis aplicando la Tarea 4.2: Contraste de la varianza . . .	285
5.36. Tarea 3.4.2. Análisis aplicando la Tarea 4.2: Método ANOVA	286
5.37. Tarea 3.4.2. Análisis aplicando la Tarea 4.2: Test CRM	288

5.38. Tarea 3.4.2. Análisis aplicando la Tarea 4.2: Grupos homogéneos	288
5.39. Tarea 3.3 vs.Tarea 3.4. Análisis estadístico (20 repeticiones)	289
5.40. Listado de tareas donde se ha utilizado la Fase 4: Medidas de calidad . . .	290
5.41. Comparativa de resultados para todas las alternativas de red respecto al PMC optimizado	293
5.42. Resumen de características de la red base para regresión	296
5.43. Resultados preliminares obtenidos para el PMC base en cada escenario (20 repeticiones)	297
5.44. Fase Previa. Resultados obtenidos para todas las alternativas en el escenario de perforación (20 repeticiones)	297
5.45. Fase Previa. Análisis aplicando la Tarea 4.2: Contraste de la varianza . . .	300
5.46. Fase Previa. Análisis aplicando la Tarea 4.2: Método ANOVA	300
5.47. Fase Previa. Análisis aplicando la Tarea 4.2: Test CRM	301
5.48. Fase Previa. Análisis aplicando la Tarea 4.2: Grupos homogéneos	302
5.49. Fase Previa. Resultados obtenidos para todas las alternativas en el escenario de detención (20 repeticiones)	303
5.50. Fase Previa. Análisis aplicando la Tarea 4.2: Contraste de la varianza . . .	304
5.51. Fase Previa. Análisis aplicando la Tarea 4.2: Test de Krustal Wallis	305
5.52. Fase 1. Resultados obtenidos por el PMC en cada escenario (20 repeticiones)	305
5.53. Fase 2. Resultados obtenidos por el PMC en cada escenario (20 repeticiones)	306
5.54. Fase 3. Resultados obtenidos por el PMC base en cada escenario (20 repeticiones)	307
5.55. Sumario y comparativa de los resultados obtenidos en cada escenario . . .	308
5.56. Comparativa de resultados para todas las alternativas de red respecto al PMC optimizado	308
B.1. Listado de ensayos generados con simulación numérica	330
B.1. Continuación de página anterior	331
B.1. Continuación de página anterior	332
B.1. Continuación de página anterior	333
B.1. Continuación de página anterior	334
B.1. Continuación de página anterior	335
B.1. Continuación de página anterior	336
B.1. Continuación de página anterior	337
B.2. Listado de ensayos generados con ruido	338

B.2. Continuación de página anterior 339
B.2. Continuación de página anterior 340
B.2. Continuación de página anterior 341

Capítulo 1

Introducción

El propósito de este capítulo es realizar una breve descripción del problema tratado en esta tesis doctoral: el impacto balístico entre dos cuerpos. Si bien este tipo de problemas ha sido estudiado desde hace varias décadas, la complejidad para recrear el comportamiento de los materiales en este fenómeno, unido a la aparición de potentes herramientas de cálculo, ha llevado a los científicos e ingenieros a utilizar nuevas alternativas computacionales para su simulación. En este sentido las aproximaciones conexionistas presentan unas características de gran interés para la resolución de problemas como el tratado.

Una vez explicada la motivación de este trabajo y definidos los objetivos perseguidos, se describen las hipótesis de partida fijadas para esta tesis doctoral. A continuación se justifica la necesidad de su resolución mediante técnicas basadas en el aprendizaje automático. Para finalizar se detalla la metodología diseñada para la consecución de los objetivos enunciados.

1.1. Contexto

Los problemas de impacto sobre estructuras tanto de baja y media velocidad, denominados *Crashworthiness*, como de alta velocidad, conocidos como impacto balístico, son de gran relevancia en el ámbito de la ingeniería estructural y de materiales avanzados. En ellos, se analiza y predice el comportamiento de materiales en condiciones extremas de trabajo, lo que permite el diseño y fabricación de nuevos componentes y estructuras susceptibles a impactos.

El diseño de estas estructuras es por lo general un proceso complejo y costoso, que requiere un alto consumo de recursos técnicos y humanos. Debido a que estas estructuras pueden ser partes esenciales de un sistema, y a que pueden sufrir durante el desempeño

de su actividad impactos de otros cuerpos, es vital conocer como se comportarían tras sufrir el choque de un proyectil.

Durante años se han realizado grandes esfuerzos para estudiar el proceso de penetración de un proyectil sobre un blanco u objetivo e intentar minimizar los daños producidos por el impacto. La mayoría de estos pueden agruparse en dos categorías, la primera se centra en estudiar el proyectil y su trayectoria, mientras que la segunda se encarga de analizar el mecanismo de impacto.

Este proceso de penetración, que se produce al impactar el proyectil sobre un objetivo, incluye fenómenos de alta complejidad que han sido investigados de forma experimental durante más de dos siglos. Sin embargo, debido a la gran cantidad de variables que intervienen, las conclusiones obtenidas mediante este método empírico son difíciles de extrapolar fuera de las condiciones de laboratorio donde se han desarrollado. En un intento de solucionar este problema, durante las últimas décadas se han desarrollado otras técnicas computacionales para la representación del proceso, conocidas como clásicas o tradicionales: simulación numérica y modelado analítico.

No obstante, estas técnicas presentan ciertas limitaciones que impiden su aplicación en todos los escenarios posibles. Por un lado, las simulaciones numéricas han demostrado su eficacia para predecir la respuesta de los objetivos frente al impacto de proyectiles. Sin embargo, necesitan una gran cantidad de recursos en términos de tiempo de computación y recursos lo que dificulta su empleo para obtener resultados a corto plazo. Por otro lado, el modelado analítico consigue respuestas en poco tiempo pero a costa de obtener una precisión inferior.

Por ello, durante los últimos años ha comenzado a utilizarse, en numerosas ramas de la ingeniería industrial y estructural, metodologías multidisciplinares para mejorar y optimizar los procesos encargados de modelar este fenómeno. En este sentido, la aplicación de técnicas predictivas y de aprendizaje automático, basadas en Redes Neuronales Artificiales (RNA, también denominadas *Artificial Neural Networks* o ANN), consiguen solventar las carencias expuestas con anterioridad para las técnicas clásicas. Estas estructuras computacionales aúnan las ventajas de las técnicas analíticas y numéricas: alta velocidad de cálculo y precisión elevada. Sus principales cualidades son, por tanto, su gran adaptabilidad dinámica y su capacidad de aprendizaje a partir de experiencias anteriores, lo que les permite adaptar su comportamiento ante situaciones dinámicas, generalizando resultados acertados al enfrentarse a nueva información.

Por último, las investigaciones encontradas en la literatura han constatado que el ámbito de aplicación de las RNA encaja con las características del problema descrito. Su utilización se presenta por tanto como una solución novedosa y atractiva para determinar, a pesar de la complejidad física del problema, el resultado de un impacto sobre una protección o blindaje.

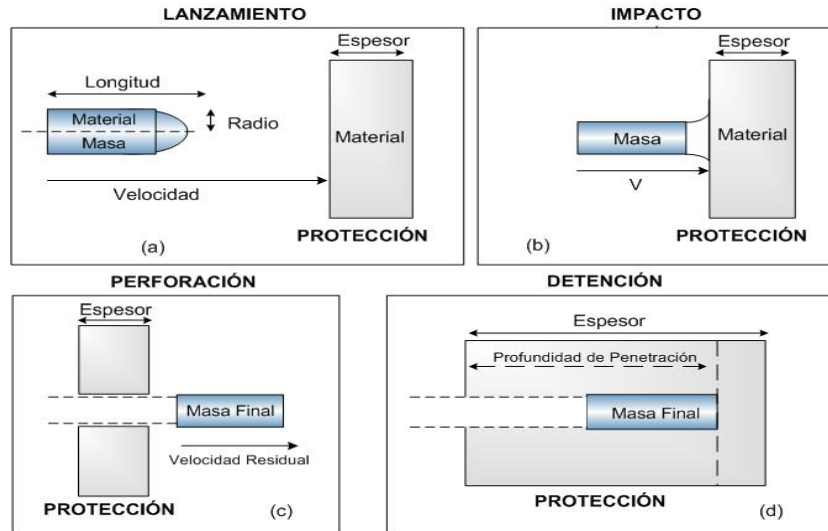


Figura 1.1: Secuencia del impacto de un proyectil

1.2. Breve descripción del problema

En terminología balística, un impacto entre sólidos se produce cuando un cuerpo diseñado para uso balístico, conocido como proyectil, golpea un cuerpo o estructura específica, denominada objetivo. Los materiales que forman parte en este fenómeno pueden ser muy variados, siendo los más utilizados los metales, las cerámicas y las aleaciones.

Desde el punto de vista puramente físico, el impacto de un proyectil sobre un panel protector o blindaje comienza con el proceso denominado penetración, que se corresponde con la entrada del proyectil en la protección una vez producido el impacto. A continuación, los dos escenarios fundamentales que se pueden originar son perforación y detención. Se produce una perforación cuando el proyectil consigue atravesar completamente la protección, siendo en este caso de interés analizar la velocidad y masa residual o final del mismo. Por otro lado, cuando el proyectil entra en la protección pero no consigue traspasarla es interesante conocer tanto la masa final como la profundidad alcanzada por el proyectil dentro de la protección, conociéndose este proceso como detención. Esta secuencia de un impacto se representa en la Figura 1.1. En §3.3 se ofrece una exposición ampliada del problema del impacto balístico tratado en esta tesis doctoral.

Debido al interés que este tipo de fenómenos ha suscitado desde su aparición, científicos e ingenieros han perfeccionado durante años diferentes propuestas para su estudio y comprensión. Éstas pueden englobarse en dos categorías diferenciadas: experimentación en laboratorio y simulación computacional (cf. §2.1.5). Sin embargo, la creciente necesidad de comprender mejor el fenómeno, unido a la utilización de nuevos materiales y el diseño de estructuras más complejas, está dificultando la aplicación de las técnicas

tradicionales en la simulación de impactos con estas características.

Comentar por último, que el ámbito de esta tesis doctoral se sitúa dentro de un marco multidisciplinar, al ser necesaria la colaboración de técnicas provenientes de diferentes ramas de la ciencia y la ingeniería. A lo largo de este trabajo se concreta la posición de esta investigación dentro de este marco y las influencias que recibe de otras disciplinas.

1.3. Motivación

Las RNA han constituido un centro focal de intensa actividad durante los últimos años, siendo un hecho demostrado su utilización cada vez más extendida en la resolución de problemas científicos y técnicos. Sin embargo, quedan todavía numerosas aplicaciones en las que no se ha aprovechado toda su potencialidad, especialmente aquellas en las que el número de variables que definen el problema es muy elevado, la respuesta del sistema altamente no lineal y la simulación compleja, ya sea por ausencia de modelos o por el elevado coste computacional asociado.

El análisis del estado del arte revela, que dentro del ámbito de las ingenierías de estructuras y materiales, se han estado empleando técnicas basadas en RNA para la modelización del comportamiento de los materiales bajo diferentes situaciones. Sin embargo apenas existen investigaciones sobre su aplicación a problemas de impacto de sólidos, ya sea impacto balístico o de tipo *Crashworthiness*. En §2.5 se detallan estas publicaciones y sus principales aportaciones.

Las bases de esta tesis doctoral provienen del proyecto de investigación, subvencionado por la Comunidad de Madrid, GR/MAT/0507/2004: “Predicción de la respuesta frente a impacto de elementos estructurales de absorción de energía utilizando técnicas basadas en redes neuronales”. Durante el mismo se desarrolló un sistema, basado en RNA, para la predicción de la respuesta de elementos estructurales de absorción de energía sometidos a impacto. Las conclusiones obtenidas, permitieron establecer la viabilidad de los sistemas de RNA como alternativa a las metodologías clásicas para la resolución de problemas de impacto de sólidos.

La presente tesis doctoral se plantea, por una parte, como una continuación del proyecto anteriormente mencionado y, por otra, como una extensión hacia un campo afín. Se pretende avanzar en el tratamiento de las propiedades de los materiales de los sólidos involucrados en el problema de impacto. Durante el proceso de entrenamiento de las RNA empleadas en el anterior proyecto, se utilizaron siempre como datos de entrada parámetros correspondientes a materiales reales. Como resultado, la RNA aprendió cuál debía ser la respuesta de cada material concreto, reconociendo el conjunto de parámetros asociado a cada uno de ellos. Sin embargo, esta forma de aprendizaje imposibilita la

utilización de la RNA para predecir la respuesta de protecciones y proyectiles realizados con materiales virtuales, con propiedades variando de forma continua aunque no presenten correspondencia con materiales metálicos reales. Aunque la utilización de estos materiales virtuales carezca de sentido para etapas finales de diseño, en las que hay que ajustarse ya a las aleaciones disponibles, sí tiene un gran interés para realizar análisis de tendencias y profundizar en la influencia que cada parámetro de la ecuación constitutiva tiene en la respuesta global. Este tipo de análisis en los que se simula el impacto utilizando materiales con propiedades variando de forma continua suele ser muy utilizado por los ingenieros en las etapas previas de diseño [Rosenberg and Dekel, 2004].

La principal motivación de esta investigación es facilitar a científicos e ingenieros una serie de modelos neuronales capaces de aproximar el efecto del impacto de un proyectil sobre una protección y conocer el comportamiento de ambos objetos después del mismo. Esto se consigue gracias a que los modelos neuronales diseñados extraen los patrones característicos del problema y cuando se les presenta nueva información en forma de ensayo balístico, son capaces de facilitar una respuesta precisa de forma inmediata.

Dentro del fenómeno del impacto balístico esta investigación se ha centrado en resolver una serie de escenarios de interés para la comunidad científica. El primero se encargará de analizar el posible resultado del impacto, y el segundo estudiará el comportamiento del proyectil y de la protección. Para ello, se diseñarán una serie de RNA para la representación de estas situaciones. Cada una de éstas, en función de su arquitectura, algoritmo de entrenamiento y aspectos relativos a los patrones utilizados, conforman lo que se denomina en el ámbito de esta investigación un *modelo*. Los modelos neuronales a diseñar tendrán como objetivo, para el primer escenario, clasificar el impacto entre perforante o no, y para el segundo, predecir la masa final y velocidad residual cuando se perfora la protección y la masa final si se detiene antes. Además se han incluido nuevas condiciones de entorno, i.e. nuevas características del proyectil y protección, y se han establecido una serie de tareas con el fin de optimizar los resultados anteriores.

Por último, un aspecto primordial que se espera conseguir con esta tesis doctoral es no solo resolver las nuevas situaciones planteadas, sino conseguir definir tareas que permitan optimizar tanto los factores inherentes al entrenamiento, como al diseño de las redes, obteniendo el máximo beneficio de los datos disponibles. Para ello se creará una metodología de optimización que ordenará y guiará las diferentes alternativas disponibles con este fin. Esto permitirá mejorar la precisión y reducir tiempo necesario para alcanzar las cotas máximas de aprendizaje.

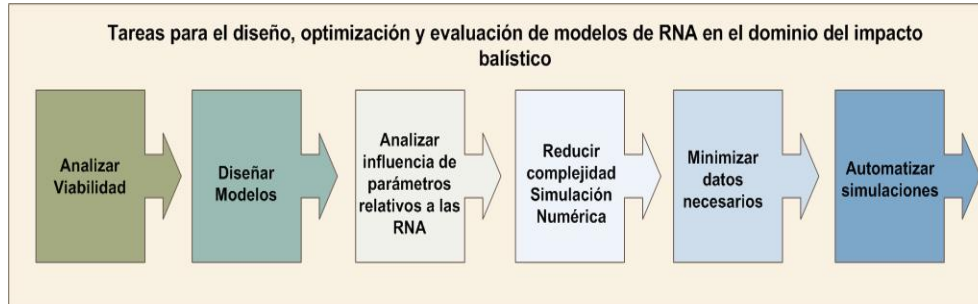


Figura 1.2: Objetivos fijados para esta investigación

1.4. Objetivos

1.4.1. Objetivo general

El objetivo final de esta tesis doctoral es mejorar el diseño de sistemas de protección ante impactos dentro del rango de la alta velocidad. Para ello, se recurrirá a la utilización de estructuras computacionales basadas en el aprendizaje automático con el fin de solventar los complejos mecanismos y recrear las leyes físicas que rigen este fenómeno.

Para lograr la consecución de este objetivo, será preciso definir una serie de tareas encaminadas en primer lugar a diseñar los modelos de RNA necesarios para su recreación, pero sobre todo a optimizar y evaluar su comportamiento en diferentes situaciones. Gracias a esto, se podrán extrapolar las conclusiones obtenidas a otras áreas ingenieriles de interés.

1.4.2. Objetivos específicos

De forma desglosada, los objetivos individuales que se pretenden obtener en esta investigación son (cf. Figura 1.2):

1. Confirmar la viabilidad de la aplicación de sistemas RNA para la solución del problema del impacto balístico.
2. Utilizar sistemas basados en RNA para clasificar el resultado del impacto balístico y predecir la respuesta de los sistemas de protección. Para ello será necesario desarrollar una serie de modelos de RNA, encargados de resolver las siguientes situaciones:
 - Clasificación: Predecir el resultado del impacto del proyectil sobre una placa metálica, i.e perforación o detención.

-
- Regresión: Predecir tras el impacto y a partir de la masa final y velocidad residual el comportamiento del proyectil y de la protección.
3. Minimizar la cantidad de información necesaria en cada modelo neuronal para mantener resultados satisfactorios. Se estudiarán diferentes técnicas para la distribución de los patrones disponibles entre los conjuntos asociados al entrenamiento y validación de una RNA. Como consecuencia, se reducirán la cantidad de ensayos que tienen que ser obtenidos mediante las técnicas clásicas.
 4. Diseñar las nuevas arquitecturas de RNA necesarias para recrear las posibles variantes del fenómeno del impacto balístico de sólidos. Se detallarán las decisiones de diseño necesarias para su completitud. Para ello se deben determinar los siguientes aspectos en cada una de las redes diseñadas:
 - Valor de inicialización de los pesos asociados a las conexiones entre las neuronas.
 - Función de activación de cada neurona.
 - Niveles o número de capas.
 - Número de neuronas de entrada, ocultas y de salida.
 5. Estudiar los modelos de RNA creados para resolver las diferentes variantes del problema del impacto balístico. Se analizará la influencia de diferentes parámetros relativos a la arquitectura de la RNA y al algoritmo de entrenamiento. Para cada modelo de red se definirán:
 - Valor del coeficiente de inercia o *momentum* y valor para la *tasa de aprendizaje*.
 - Criterio de parada para el proceso de aprendizaje.
 - Modo de entrenamiento.
 - Variantes del algoritmo de aprendizaje.
 6. Realizar un análisis paramétrico de las diferentes variantes del problema balístico. La meta es disponer de una herramienta capaz de dar información sobre los factores y variables más relevantes, tanto geométricos como relativos a las propiedades del material, que influyen en el proceso de aprendizaje de la RNA. La consecución de este apartado conllevará:
 - Obtener conclusiones y realizar inferencias sobre los parámetros más relevantes del fenómeno, así como conocer las propiedades óptimas de los materiales involucrados. Esto incentivará el diseño de nuevos materiales con esas propiedades y su utilización para la construcción de protecciones y proyectiles.
 - Reducir la complejidad asociada a la simulación mediante técnicas tradicionales de los ensayos de impactos, eliminando aquellas características de los objetos sin influencia significativa en el aprendizaje de la red.
-

7. Facilitar el proceso de entrenamiento y aprendizaje de la RNA, con el diseño de una herramienta para la automatización de las simulaciones. Dicha herramienta permitirá realizar múltiples simulaciones de forma automática y transparente para el usuario final. Los requisitos a cumplir serán:
 - Agrupar bajo una misma simulación diferentes configuraciones de: arquitecturas de red, cantidad y reparto de los datos y parámetros relativos al algoritmo de entrenamiento.
 - Simplificar la evaluación de los modelos neuronales agrupando resultados individuales.
8. Obtener un estudio comparativo de los problemas analizados, tratando de mostrar las ventajas inherentes al empleo de la metodología basada en RNA en comparación con otras alternativas de aprendizaje automático.

1.5. Hipótesis

En §3.3 se describieron los principales interrogantes derivados del problema que se pretende resolver en esta tesis doctoral. Las hipótesis de partida formuladas de acuerdo a los mismos son las siguientes:

- Los problemas de impacto balístico sobre estructuras son de gran relevancia en el ámbito de la ingeniería estructural y de materiales avanzados. Por ello, es de especial interés en el ámbito ingenieril, analizar y predecir el comportamiento de materiales en condiciones extremas de trabajo.
- El análisis convencional de los problemas de impacto balístico presenta un elevado coste computacional, debido a su carácter fuertemente no lineal y al elevado número de parámetros que deben ser considerados.
- Las técnicas predictivas basadas en aprendizaje automático, como las RNA, pueden aportar soluciones atractivas en problemas que cumplan las citadas características.
- Las RNA pueden utilizarse junto a las técnicas clásicas de modelado y resolución de casos de impactos balísticos, como las simulaciones numéricas, para resolver problemas como el planteado.
- La capacidad de las RNA para adaptarse a entornos cambiantes permite su utilización en este dominio bajo diversas configuraciones de proyectiles y protecciones.
- La optimización de factores relativos a la arquitectura de la red, algoritmo de entrenamiento y datos disponibles, beneficia la capacidad de generalización de los modelos de RNA creados, y por tanto mejora los resultados obtenidos, tanto en tiempo de ejecución como en precisión.

1.6. Justificación

Las protecciones diseñadas para sufrir posibles impactos deben presentar una notable capacidad de absorción de energía para evitar o paliar los daños en el sistema que preservan. La simulación del comportamiento de estas protecciones constituye uno de los problemas de diseño de elementos estructurales que mayor coste computacional presenta. La fuerte no linealidad de estos problemas, debida a las grandes deformaciones y contactos, unido al gran número de parámetros influyentes en el proceso, hace que los tiempos de cálculo sean excesivos para las fases previas de diseño.

En un intento de proponer herramientas de diseño alternativas para este tipo de sistemas, en esta tesis doctoral en primer lugar se pretende desarrollar un sistema para la predicción de la respuesta de elementos estructurales de absorción de energía sometidos a impacto, basado en la utilización de RNA. En segundo lugar se propone una metodología de optimización orientada a localizar los modelos neuronales que mejor se adapten a cada uno de los escenarios a estudiar.

Gracias a estas aportaciones, mientras que la simulación numérica de un problema de impacto puede conllevar varias horas de ejecución, la RNA, una vez realizado el proceso de aprendizaje y optimización, permite obtener el resultado en una fracción de segundo.

¿Por que es importante el estudio del fenómeno del impacto balístico?

Como ya se ha comentado, la simulación mediante códigos numéricos del impacto entre sólidos en el rango de la alta velocidad constituye uno de los problemas de diseño de elementos estructurales que mayor coste computacional presenta. Durante los últimos cincuenta años, los científicos e ingenieros han dedicado cada vez más esfuerzo a solucionar problemas de este tipo, intentando conocer el comportamiento de los materiales y estructuras ante el impacto de un objeto.

En §2.1.5 se describen las diferentes metodologías, basadas en las técnicas clásicas, utilizadas por la comunidad científica para la resolución de los problemas de impacto balístico.

¿Por qué utilizar técnicas de aprendizaje automático basadas en RNA?

En los últimos años, las RNA han sido ampliamente utilizadas para modelar las actividades humanas en diversas áreas de la ciencia y de la ingeniería. Las características de estas estructuras computacionales permiten una inducción del conocimiento, i.e. son capaces de generalizar comportamientos a partir de una información no estructurada suministrada en forma de ejemplos.

Por otro lado, las RNA pueden manejar ciertos cambios en la información de entrada. Esto es debido a que son capaces de extraer las patrones esenciales de las entradas aprendidas, por lo que pueden procesar correctamente datos incompletos o distorsionados. En definitiva, la gran variedad de redes disponibles, permiten la resolución de una amplia variedad de problemas complejos. La tipología de red elegida para esta investigación, el *Perceptrón MultiCapa* (PMC), o *MultiLayer Perceptron* (MLP), tiene una reconocida capacidad como aproximador universal de funciones, lo que avala su utilización para el modelado de problemas no lineales como el tratado en esta investigación [Hornik et al., 1989].

La viabilidad de este tipo de soluciones para resolver problemas de impacto de sólidos, ha sido verificada mediante los resultados obtenidos en el desarrollo del proyecto de investigación financiado por la Comunidad de Madrid con denominación GR/MAT/0507/2004. En relación con dicha investigación, el autor ha realizado varias contribuciones en forma de ponencia en congreso [Gonzalez-Carrasco et al., 2008], capítulo de libro [Garcia-Crespo et al., 2008] y revista internacional [Garcia-Crespo et al., 2006, Gonzalez-Carrasco et al., 2009]. Además se ha presentado una tesis doctoral [Fernandez, 2007] donde se aborda el problema del impacto de sólidos, desde el punto de vista del modelado de los objetos, mediante sus ecuaciones constitutivas, su simulación numérica y su posterior comparación con las RNA.

Por otra parte, las búsquedas bibliográficas realizadas hasta la fecha han mostrado un escaso número de referencias dedicadas a la aplicación de RNA a problemas de impacto balístico o de *Crashworthiness* [Carpenter and Barthelemy, 1993, Hajela and Lee, 1997, Lanzi et al., 2004, Fernandez et al., 2008].

Gracias a lo expuesto en este apartado, se puede subrayar que la temática propuesta en esta investigación presenta un notable interés, tratando de mejorar el diseño de sistemas de protección ante impactos mediante la utilización de estructuras computacionales basadas en el aprendizaje automático.

¿Por qué es importante optimizar un modelo neuronal

Partiendo de la premisa $modelo = patrones + arquitectura + algoritmo$ y teniendo en cuenta el elevado número de parámetros, técnicas y métodos disponibles para entrenar una RNA, se hace necesaria un serie de pasos para guiar, estructurar y priorizar las diferentes alternativas existentes. Además muchas de estas alternativas no tienen justificación teórica o matemática por lo que son dependientes del problema lo que implica que el rendimiento de una RNA depende de una gran cantidad de factores que necesitan ser probados y evaluados tanto de forma individual como conjunta.

Por ello, la optimización de los modelos neuronales se llevará a cabo mediante la

definición de una metodología de investigación, basada en una serie de actividades y tareas orientadas a cumplir los objetivos e hipótesis expuestos para esta tesis doctoral. Para homogeneizar el gran número de alternativas existentes y simplificar su aplicación sobre cada uno de los escenarios, éstas se han agrupado en tres grandes bloques o fases: patrones, arquitectura y algoritmo. En cada uno de ellos se incluyen las variantes que según la literatura analizada mejor se adaptan a problemas ingenieriles similares al tratado en este estudio.

¿Que aportaciones tendrá esta tesis doctoral?

El desarrollo de una metodología para el diseño, optimización y evaluación de modelos de RNA, ayudará a la creación de protecciones para ámbitos impactos balísticos, solventando los problemas asociados a las técnicas clásicas empleadas hasta el momento.

A lo largo de la investigación se mostrará la capacidad de estas estructuras para determinar, a pesar de la complejidad física del problema, por un lado el resultado de un impacto de alta velocidad sobre blindajes y por otro el comportamiento de los objetos que intervienen en el fenómeno. Gracias a esto, y recreando la mecánica del impacto, se podrá determinar con gran fiabilidad dos aspectos fundamentales inherentes al diseño de protecciones: si conseguirá detener un determinado proyectil y, en caso contrario, la velocidad y la masa residual del mismo.

Los diferentes modelos neuronales diseñados para resolver estas cuestiones serán validados para distintas geometrías de proyectil y de panel protector, velocidades y materiales. Además, las tareas definidas en la metodología permitirán optimizar por un lado aquellos factores relativos al diseño y entrenamiento de las redes, y por otro a la calidad y velocidad de las respuestas. Con esto, se conseguirá definir aquellos modelos de RNA más exactos para la resolución del problema del impacto balístico.

Asimismo, la utilización de técnicas de aprendizaje como las RNA, con gran capacidad predictiva, presentan la ventaja frente a los códigos de elementos finitos o diferencias finitas, de su rápida respuesta. Mientras que la simulación de un problema de impacto puede conllevar varias horas de ejecución, una RNA, una vez realizado el proceso de aprendizaje, permite obtener el resultado de forma casi inmediata. Gracias a esto, los ingenieros e investigadores podrán ensayar con nuevas configuraciones de proyectiles y paneles protectores con rapidez, sin necesidad de recurrir a costosas pruebas experimentales.

Por último, esta investigación abrirá nuevas posibilidades, tanto en el mismo área de la protección frente a impacto de alta velocidad como en áreas afines, entre la que cabe destacar la de la seguridad pasiva de vehículos. En ésta, es de especial relevancia el diseño de estructuras cuya misión es absorber energía en choques a baja y media velocidad, como

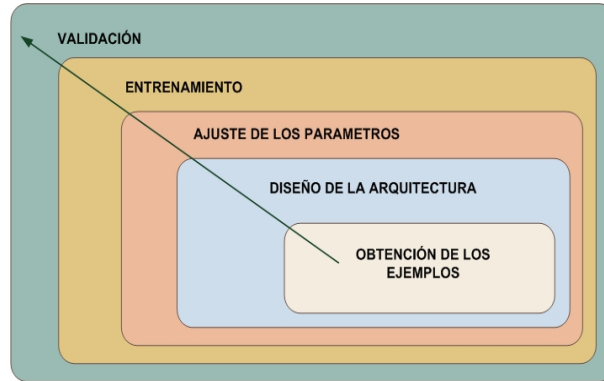


Figura 1.3: Pasos necesarios para completar la simulación del problema mediante RNA

los que pueden aparecer en accidentes de tráfico o ferroviarios, aterrizajes de emergencia de helicópteros, etc. Se trata de diseñar estructuras capaces de absorber energía para evitar o paliar los daños a los pasajeros de los vehículos afectados. Por las características de los vehículos para los que se diseñan estas estructuras, deben usarse materiales ligeros como aleaciones de aluminio y materiales compuestos de matriz polimérica reforzados con fibras.

1.7. Metodología de la investigación

Este apartado detalla las actividades en las que se ha dividido la investigación para alcanzar los objetivos expuestos en §1.4. Para la resolución de un problema ingenieril mediante la utilización de RNA es necesaria la realización de una serie de pasos claves (cf. Figura 1.3). El primero de ellos es la obtención de los ejemplos a partir de los cuales se va a extraer el conocimiento, posteriormente se procede al diseño de las arquitecturas de red necesarias, el ajuste de los parámetros correspondientes, y para finalizar el entrenamiento y validación del aprendizaje.

A continuación se definen las actividades o métodos necesarios para adaptar estos pasos genéricos al dominio del impacto balístico, definiendo de esta forma una metodología para conseguir los objetivos fijados al comienzo de la investigación.

El cumplimiento de las siguientes actividades permitirá afinar tanto el algoritmo de entrenamiento como las arquitecturas diseñadas y reducir los patrones necesarios para alcanzar un nivel óptimo de precisión en cada uno de los modelos de red (cf. Figura 4.1).

1. **Estado del arte.** El principal objetivo de esta actividad consiste en detallar los trabajos realizados con RNA dentro del ámbito del impacto de sólidos. Las principales cuestiones tratadas en este apartado serán:

- Definición del problema a resolver. Para ello se realizará una descripción pormenorizada del fenómeno de penetración tratado, centrándose en los elementos fundamentales del mismo: proyectil y protección.
 - Descripción física del problema, enfocada en la física de los materiales que intervienen y la dinámica del fenómeno balístico.
 - Análisis crítico de las principales alternativas clásicas para la resolución de problemas de impacto.
 - Identificación de los principales parámetros que influyen en la capacidad de predicción de las RNA dentro del dominio tratado.
 - Antecedentes de la aplicación de soluciones basadas en RNA dentro del dominio estudiado.
2. **Simulación numérica de los ensayos.** El comportamiento de los proyectiles y de las protecciones ante los impactos será recreado mediante herramientas de simulación numérica. Para ello, se generará un conjunto de casos de impacto con diferentes valores de los parámetros descriptivos de cada una de las estructuras, proyectil y protección, que intervienen en el problema.
 3. **Desarrollo de nuevos modelos de RNA.** El trabajo a realizar por el autor en este apartado, consistirá en crear nuevos diseños RNA para resolver dos variantes del problema del impacto: clasificación del resultado y predicción de la velocidad final y masa residual. Se han ampliado además las variables de entorno que definen cada problema, incluyendo nuevas características tanto del proyectil como de la placa de protección. Los resultados provenientes de las simulaciones, serán de gran utilidad para conocer la capacidad de extrapolación de las conclusiones obtenidas para nuevas características o incluso su aplicación en otros dominios similares.
 4. **Simulación de los nuevos modelos de RNA diseñados.** Las RNA serán probadas para distintas geometrías, tanto de proyectil como de panel protector, velocidades y materiales. Los resultados iniciales han mostrado una gran capacidad predictiva con la ventaja, frente a los códigos de elementos finitos o diferencias finitas, de su rápida respuesta. Mientras que la simulación de un problema de impacto puede conllevar varias horas de ejecución, la RNA, una vez realizado el proceso de aprendizaje, permite obtener el resultado de forma inmediata.
 5. **Análisis de posibles alternativas para la optimización del aprendizaje y de los resultados.** Existen múltiples parámetros de una RNA, tanto de la arquitectura como del algoritmo de entrenamiento, que modifican su capacidad de generalización y que afectan por tanto a su precisión de predicción. La búsqueda de valores óptimos para estos parámetros, es un punto de importancia dentro del trabajo a realizar, ya que permitirá optimizar el aprendizaje y los resultados. Dentro de esta actividad destacan las que se enuncian a continuación, ya que van a permitir reducir la complejidad de la simulación numérica asociada a la obtención de los ensayos balísticos:

- **Análisis de la influencia de las variables que caracterizan el proyectil y la protección.** Una vez diseñado el modelo de RNA, es posible que ciertas variables de entrada tengan una influencia mayor que otras en su capacidad de generalizar. Esto puede deberse a posibles interacciones entre ellas o a que alguna no aporte información relevante para el entrenamiento. Debido al elevado coste computacional de las simulaciones numéricas, es interesante incluir en la arquitectura final de una RNA únicamente aquellas variables explicativas, i.e. que contribuyan de forma significativa al aprendizaje de la red.
 - **Minimización de los datos necesarios para entrenar la RNA.** El aprendizaje de la red se realizará a partir de costosos ensayos simulados numéricamente. Es por tanto de gran interés minimizar la utilización de estos datos en el proceso de entrenamiento de la RNA.
6. **Evaluación y Validación.** Las actividades de evaluación y validación incluidas pretenden, estimar la efectividad de las diferentes soluciones de RNA planteadas, y probar la validez de la metodología de optimización propuesta.
7. **Documentación y Conclusiones.** En esta actividad se procederá a documentar cada uno de los aspectos de la tesis. Se describirán las diferentes alternativas de RNA así como sus contribuciones a los resultados obtenidos. Por último, se presentarán las conclusiones que se extraigan de la evaluación y validación realizada.

1.8. Estructura del documento

Para una mejor comprensión de la investigación realizada, este documento ha sido estructurado en una serie de capítulos, los cuales se explican a continuación de forma resumida.

- En el Capítulo 2 se describen las principales características del problema a resolver, analizándose los antecedentes de los sistemas de RNA para los problemas de impactos de sólidos. Además se discuten las principales bases teóricas utilizadas durante la investigación: simulación numérica y RNA. A continuación se hace especial hincapié en las referencias encontradas en la literatura acerca del uso de las RNA en el ámbito ingenieril y más concretamente dentro del fenómeno del impacto balístico. Por último, se incluye una revisión de las técnicas más utilizadas para la optimización de los resultados obtenidos por un sistema de RNA.
- En el Capítulo 3 se especifica el marco integral de la solución, exponiéndose las diferentes disciplinas que se engloban dentro de esta investigación. Se detalla la solución propuesta, basada en un sistema PMC, entrenado a partir de ensayos

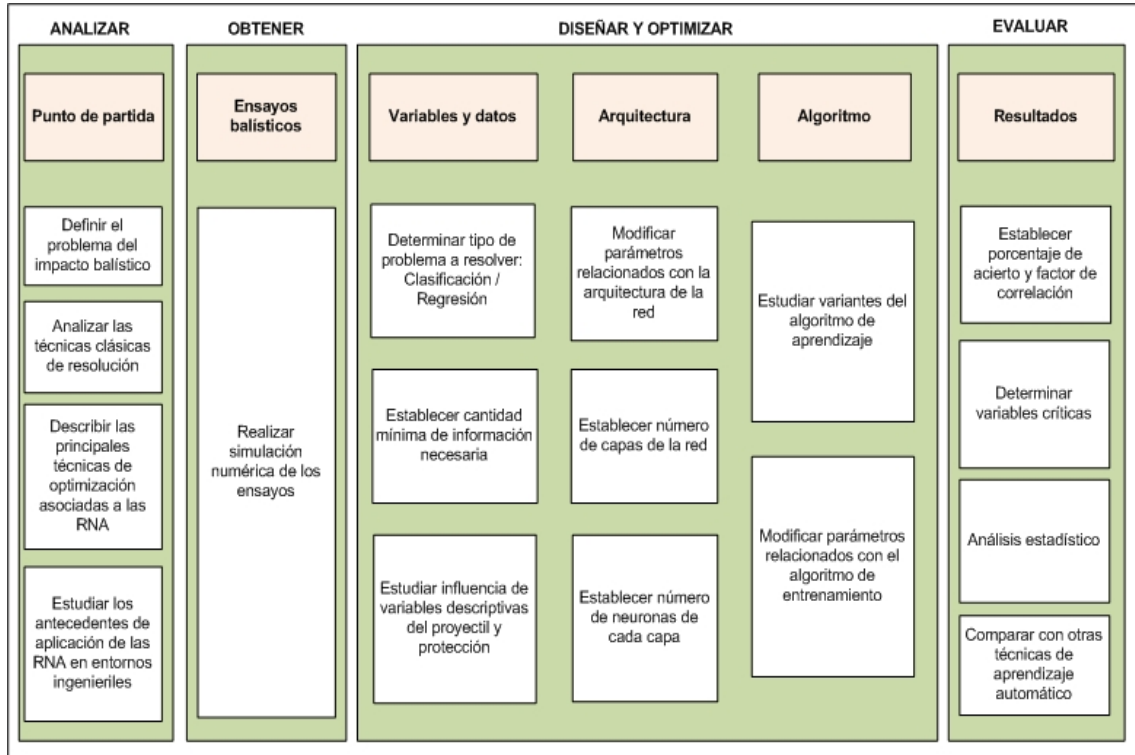


Figura 1.4: Método de trabajo diseñado para la tesis doctoral

simulados numéricamente y las diferentes alternativas basadas en el aprendizaje automático que serán empleadas para evaluar la validez de la investigación propuesta.

- En el Capítulo 4 se pormenoriza la metodología de investigación, basada en una serie de actividades y tareas, orientada a la optimización de los modelos neuronales. Partiendo de la premisa $modelo = arquitectura + patrones + algoritmo$ se aplicarán sobre cada término de la ecuación aquellas técnicas y métodos que han demostrado, en la literatura analizada, mejores resultados para problemas ingenieriles similares al tratado en esta investigación.
- En el Capítulo 5 se aplica la metodología de optimización diseñada sobre cada uno de los diferentes escenarios propuestos. Los resultados obtenidos sobre el PMC serán comparados con otras alternativas de RNA y analizados siguiendo una serie de pruebas estadísticas para asegurar su validez.
- En los Capítulos 6 y 7 se muestran y razonan las conclusiones obtenidas, se aportan posibles nuevas líneas de investigación a seguir y se listan las publicaciones realizadas en diferentes ámbitos como consecuencia de esta investigación. Para

adecuar esta tesis al marco europeo de doctorado se incluyen tanto en español como en inglés.

- El Anexo A desglosa los diferentes acrónimos que han aparecido a lo largo del documento. Por último, el Anexo B contiene el listado de los ensayos balísticos obtenidos mediante simulación numérica y los generados mediante la introducción de ruido Gaussiano.

1.9. Convenciones

En este apartado se exponen una serie de aclaraciones acerca del estilo seguido en la redacción de esta tesis:

- Las citas textuales aparecerán encerradas entre dobles comillas (“”), con letra cursiva y con la cita bibliográfica donde fueron obtenidas.
- Los términos que se mantengan en su idioma original, bien por ser aceptados dentro del dominio, para preservar su semántica o por no existir traducción directa al español, aparecerán en cursiva.
- Los términos técnico-científico, de interés dentro de la investigación, se mostrarán en cursiva en su primera aparición en el documento para facilitar su identificación.
- Las citas bibliográficas incluidas en este documento siguen el sistema autor-año.
- Las notas al pie de página se han reservado para realizar aclaraciones sobre ciertos términos referidos en este trabajo o para facilitar recursos web de interés.

Capítulo 2

Estado del Arte

El impacto balístico es una situación altamente compleja, con implicaciones termodinámicas, estados tensionales tridimensionales y velocidades de deformación elevadas. Los científicos han dedicado, desde hace más de cincuenta años, muchos esfuerzos a resolver problemas donde se presentan estos fenómenos. Para ello han recurrido a tres aproximaciones clásicas, conocidas como: predicción empírica o experimental, simulación numérica y modelado analítico. No obstante, estas técnicas presentan ciertos problemas que impiden su utilización en todos los escenarios posibles.

La utilización de técnicas predictivas y de aprendizaje automático basadas en RNA aúnan las ventajas de los modelos analíticos y numéricos: velocidad de cálculo y precisión elevadas. Durante los últimos veinticinco años, las RNA han sido utilizadas con éxito para la resolución de una amplia gama de problemas ingenieriles. Su capacidad para tareas como el reconocimiento de patrones, la clasificación y la aproximación de funciones, ha permitido que en la actualidad su uso se haya extendido al dominio del impacto de sólidos.

2.1. Impactos de proyectiles sobre blindajes metálicos a alta velocidad

2.1.1. Interés del problema

Durante los últimos cincuenta años, los científicos e ingenieros han dedicado cada vez más esfuerzo a solucionar problemas referentes al impacto de sólidos. Hasta hace un par de décadas, el análisis de estos problemas giraba de forma casi exclusiva en torno a colisiones de sólidos, de geometría simple, con impactos de baja-media velocidad y deformación plástica limitada. Los impactos a alta velocidad, debido a las complejas ecuaciones que los caracterizan y a la limitada información disponible sobre el comportamiento de los materiales en su deformación, eran recreados generalmente en pruebas de laboratorio

[Goldsmith, 1960].

La tendencia actual hacia computadoras con una gran capacidad de procesamiento, junto con la aparición de nuevas técnicas instrumentales dentro del campo de la mecánica, ha permitido resolver muchos problemas relacionados con el impacto sobre materiales. En la actualidad, existen modelos definidos que permiten a los ingenieros realizar estudios en diferentes ámbitos, como accidentes de coches, impactos sobre aspas de turbinas, optimización estructural, etc. El estudio de este fenómeno es de gran relevancia en el ámbito de la ingeniería estructural y de materiales avanzados, donde se analiza y predice el comportamiento de materiales en condiciones extremas de trabajo.

Dentro del campo balístico, con proyectiles que alcanzan velocidades superiores a los 500 m/s, los ejemplos más característicos se encuentran en el ámbito militar, donde se estudia el comportamiento de blindajes, diseñados para la protección de vehículos y personas. Sin embargo durante las últimas décadas, este tipo de problema ha migrado con interés a los usos civiles, en los cuales se requieren elementos estructurales para absorber la energía del proyectil de modo que no dañe partes críticas del sistema global.

Por otro lado, existen numerosos elementos que pueden sufrir el impacto de dichos proyectiles durante el desempeño de su actividad, ya sea militar o civil. Estos pueden ser partes esenciales de un sistema, por lo que es vital conocer su comportamiento durante y después del impacto.

Las investigaciones realizadas por otros autores dentro de este ámbito, se han centrado en conocer dos aspectos del estado del proyectil tras el impacto: su velocidad residual [Sun and Potti, 1996, Borvik et al., 1999, Mimaroglu et al., 2006] y masa final [Silling and Forrestal, 2007]. El primero establece la velocidad del proyectil tras el impacto. En el caso de no conseguir atravesar el blindaje ésta siempre es cero, pero cuando lo consigue el valor de este parámetro es de especial interés para comprender el comportamiento del conjunto proyectil-protección. Si se produce una perforación completa, el proyectil se convierte en una amenaza, por lo que el cálculo de su velocidad residual es vital para conseguir proteger a los sistemas u ocupantes de la estructura. El segundo parámetro, permite conocer la deformación que ha sufrido el proyectil a través de la masa perdida durante el proceso de penetración. Otro parámetro, ampliamente estudiado, es la profundidad de penetración tras el impacto (DOP o *Depth of Penetration*) [Yarin et al., 1995, Franzen et al., 1997, Wen, 2001]. Éste permite averiguar, en los casos en los que no hay perforación, cuánto se ha deformado la parte posterior de la placa de blindaje.

Gracias a las investigaciones llevadas a cabo en este dominio se han abierto nuevas posibilidades en áreas afines, entre las que cabe destacar la seguridad pasiva de vehículos. En este caso es de especial relevancia el diseño de estructuras cuya misión es absorber energía en choques del tipo *Crashworthiness*, como los que pueden aparecer en accidentes de tráfico o ferroviarios, aterrizajes de emergencia de helicópteros, etc. Se entiende por

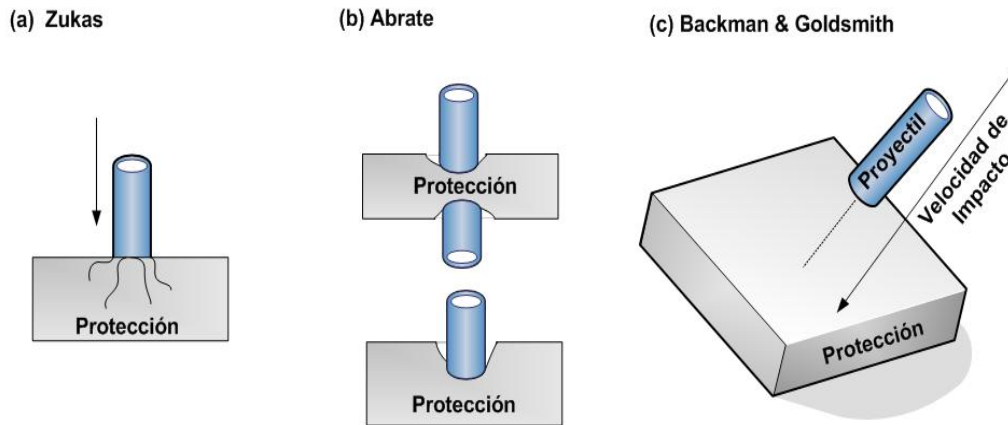


Figura 2.1: Definiciones de impacto balístico según los diferentes autores

Crashworthiness, la habilidad de las diferentes estructuras que intervienen en el impacto para proteger a los ocupantes de un vehículo. Estos impactos suelen producirse en el régimen de baja velocidad, entre 200 y 500 m/s, y el objetivo es por tanto diseñar estructuras capaces de absorber energía para evitar o paliar los daños en el sistema que intentan preservar.

Por último comentar que el diseño de estructuras, dentro del dominio de los impactos a alta o baja velocidad, es una actividad compleja que se aborda, tradicionalmente, desde tres metodologías complementarias: *el método empírico*, *la simulación numérica* y *la modelización analítica*. Las características de cada una, así como sus ventajas e inconvenientes serán analizadas en §2.1.5.

2.1.2. Impacto balístico

En terminología balística, un impacto entre sólidos se produce cuando un cuerpo diseñado para uso balístico, conocido como proyectil, golpea un objeto específico, denominado objetivo.

Hasta el momento, desde el punto de vista mecánico se han dado múltiples definiciones de impacto balístico en la literatura. Según Zukas [Zukas et al., 1982], se entiende por impacto balístico la colisión entre un proyectil y un blanco a una velocidad tal que deben considerarse efectos inerciales y de propagación de ondas de tensión en el material (cf. Figura 2.1 caso a).

Por su parte, Backman y Goldsmith [Backman and Goldsmith, 1978] utilizan la velocidad del impacto para establecer otra clasificación alternativa. Aquellos choques con

una velocidad de impacto superior a 500 m/s, son etiquetados como de alta velocidad, mientras que los que presentan valores inferiores se denominan de baja velocidad (cf. Figura 2.1 caso b). En este sentido, Abrate [Abrate, 1998] comenta que una de las clasificaciones posibles es la que distingue entre los impactos con una penetración completa, denominados impactos de alta velocidad o impactos balísticos y los que no lo logran, clasificados como impactos de baja velocidad (cf. Figura 2.1 caso c). Señala además que otra posible clasificación, relacionada con la propuesta por Zukas, consiste en denominar impactos de baja velocidad aquellos en los que la propagación de las ondas de tensión a través del espesor del material no tienen una importancia significativa. Por consiguiente los impactos de alta velocidad se producen cuando la relación existente entre la velocidad de impacto y la velocidad de las ondas compresivas que se propagan a través del espesor del material, son superiores a la máxima tensión de fallo que la estructura puede soportar en esa dirección.

Con el paso del tiempo estas definiciones han ido combinándose, dando como resultado que en la actualidad la denominación más utilizada por la comunidad científica, es aquella que describe el impacto balístico como el impacto de alta velocidad, superior a 500 m/s, de un cuerpo con poca masa impulsado por un dispositivo externo, sobre un objetivo determinado [Naik et al., 2005].

Desde el punto de vista del diseño de estructuras, una tarea fundamental es dotar de cierta dosis de protección a aquellos objetivos, ya sean militares o civiles, que debido a su función son susceptibles de sufrir impactos por parte de proyectiles. Los dispositivos o sistemas de seguridad pasiva permiten reducir las consecuencias de estos choques, destacando desde el punto de vista balístico, la coraza o blindaje. Su objetivo es dispersar los efectos de un impacto concentrado y absorber las consecuencias térmicas producidos por el mismo.

El proceso de entrada de un proyectil dentro de estas protecciones durante el impacto es denominado penetración. En la mayoría de las ocasiones, el choque da como resultado el empotramiento del proyectil y la formación de un cráter o cavidad en el objetivo. Se trata de un proceso en el que no existe situación de equilibrio y, en consecuencia, distintas zonas de la estructura impactada pueden estar sometidas a estados tensionales muy distintos. En este sentido, las variables que caracterizan la respuesta de una estructura ante un impacto dependen de forma acusada de la velocidad de impacto. Como norma general puede decirse que a medida que esta aumenta, el comportamiento de la estructura se hace secundario frente al comportamiento del material [Viechnicki et al., 1991].

El fenómeno del impacto balístico se detallará durante los siguientes apartados, prestando especial atención tanto a los elementos que intervienen en el mismo, proyectil y protección, como al resultado del impacto.

2.1.2.1. Nociones de impacto balístico

La balística se define como el estudio científico de todo lo relativo al movimiento de los proyectiles. Dicho movimiento, desde el momento del disparo hasta su impacto en el blanco, se divide en tres fases distintas, examinadas en sus correspondientes disciplinas: *Balística Interior*, que estudia el movimiento del proyectil mientras se encuentra dentro del cañón, *Balística Exterior*, que considera el movimiento del proyectil desde el momento en que abandona el cañón hasta que alcanza el blanco, y *Balística Terminal o de Efecto*, que analiza el efecto del proyectil sobre el blanco [Manzano-Trovamala et al., 2001]. El desarrollo de esta última disciplina ha permitido identificar una serie de factores o variables inherentes a la conformación del proyectil y al medio en el que incide una vez disparado. Para ello explora los resultados producidos en el cuerpo u objeto sobre el que impacta la bala o proyectil y el estado final del proyectil [Malcolm, 2005].

La velocidad inicial o velocidad en la boca del cañón es uno de los factores más importantes, ya que afecta a muchos de los mecanismos de deformación que gobiernan los materiales que se ven involucrados en el choque. Se expresa en m/sg y para su cálculo se mide la distancia que recorre en metros el proyectil en el primer segundo, inmediatamente después de salir por la boca de fuego.

La mayoría de los dispositivos utilizados en este campo son pistolas convencionales, en las cuales el proyectil adquiere una velocidad que varía entre los 500 y 1300 m/s, siendo este el rango usado como norma en municiones. El rango inferior de 25-500 m/s se alcanza mediante pistolas neumáticas u otros aparatos de laboratorio. Existen rangos superiores de hasta 3000 m/s, conocidos como ultra e hipervelocidad, que se consiguen mediante pistolas de propósito específico o aceleradores [Backman and Goldsmith, 1978, Farrar and Leeming, 1983].

Por otro lado, la velocidad con la que impacta un proyectil sobre un cuerpo, denominada velocidad de impacto, es un parámetro fundamental para conocer el comportamiento de las estructuras y materiales que intervienen en el proceso de penetración. Cuando la velocidad de impacto es relativamente baja, inferior a 500 m/s, la geometría y rigidez de la estructura puede resultar determinante en el resultado del choque. Sin embargo, si la velocidad de impacto supera los 500 m/s, las propiedades mecánicas y dinámicas de los materiales y el flujo hidrodinámico controlan la respuesta del sólido. Por último, si la velocidad de impacto pertenece al rango de la hipervelocidad, el material se comportará como un fluido sometido a presiones enormes, siendo la tensión ejercida por el impacto varios órdenes superior a la resistencia del material [Viechnicki et al., 1991, Naik and Shrirao, 2004].

El propósito con el que se diseña un proyectil es impactar en un punto deseado sobre un objetivo, siendo obvio que a mayor velocidad y peso lleve, más energía será capaz de transportar y ceder. En este sentido, la energía cinética en el momento del impacto (cf.

Ecuación 2.1) es directamente proporcional a la masa y al cuadrado de la velocidad, lo cual, aplicado al proyectil en movimiento, establece que a mayor velocidad, más energía es capaz de transportar.

$$E_c = (1/2)mv^2 \quad (2.1)$$

Durante el impacto balístico, existe una transferencia de energía entre el proyectil y el cuerpo impactado, ya que ésta es dispersada y absorbida por el objetivo. Existen varios comportamientos posibles en función de la geometría del objetivo, las propiedades de los materiales y los parámetros del proyectil:

1. El proyectil perfora el objetivo y sale con una cierta velocidad residual. En este caso la energía cinética inicial del proyectil era superior a la cantidad de energía que podía absorber el objetivo.
2. El proyectil penetra parcialmente en el objetivo. Esto indica que la energía cinética inicial del proyectil era inferior que la cantidad que podía absorber el objetivo.
3. El proyectil perfora el objetivo completamente con una velocidad de salida o residual cero. Cuando esto sucede la energía cinética del proyectil es absorbida en su totalidad por el objetivo.

2.1.2.2. Variables relativas al impacto balístico

El fenómeno del impacto balístico tiene lugar entre dos cuerpos, el proyectil y la protección, apareciendo entre ambos un gran número de variables que pueden influir en el proceso del choque y posterior perforación del objetivo. Todos ellos pueden agruparse en torno a cuatro puntos:

1. Velocidad del impacto.
2. Ángulo de incidencia del impacto.
3. Configuración, características y material constituyente del proyectil.
4. Configuración, características y material constituyente del objetivo.

Según comentaron Backman y Goldsmith en 1978 [Backman and Goldsmith, 1978], esta primera suele ser la más importante ya que la velocidad afecta con tanta intensidad a los fenómenos que forman parte del impacto que prácticamente anula cualquier otra consideración. Esto es debido a que la velocidad cuando se produce el impacto determina la presión hidrodinámica del mismo, i.e., la resistencia que encuentra el proyectil a la hora de penetrar en el objetivo. Por otro lado la velocidad de impacto tiene un efecto significativo sobre la deformación del proyectil, creando un efecto de *cavitación* entre el proyectil y el objetivo al comportarse ambos como fluidos.

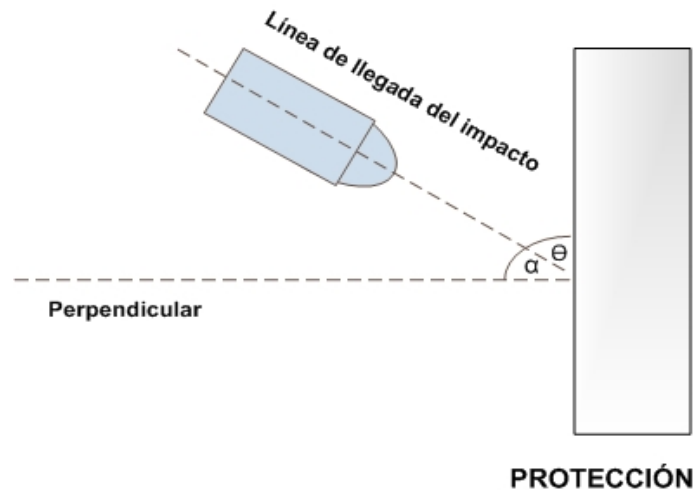


Figura 2.2: Ángulo de impacto

La cavitación se produce por una vaporización local del líquido, inducido por una reducción hidrodinámica de la presión. Esta produce la formación de bolsas localizadas de vapor o cavidades dentro del líquido, apareciendo casi siempre en las proximidades de las superficies solidas que limitan con el. Esta cavidad se extiende de forma radial a lo largo del camino del proyectil sobre el objetivo durante la penetración. A mayor cavitación, el proyectil consigue menos penetración.

Otra consideración esencial a la hora de analizar los problemas balísticos es conocer el ángulo de impacto del proyectil sobre el blanco, i.e., el ángulo entre la línea de llegada del proyectil y la perpendicular al plano del objetivo. Éste dependerá de varios factores externos como la aerodinámica y estabilidad del proyectil en vuelo y el tipo de proyectil y blanco, por lo que representa un factor de gran complejidad para modelar (cf. Figura 2.2).

Con respecto al proyectil, la densidad y longitud del mismo en la dirección de penetración son factores a tener en cuenta durante su construcción. Este proceso de fabricación determinará si la tensión del impacto permitirá al mismo superar la capacidad de resistencia balística del objetivo, estableciendo si el proyectil es suficientemente duro para sobrevivir al impacto y penetrar, o en caso contrario su destrucción. En este último caso es posible que los fragmentos resultantes tengan suficiente energía para poder penetrar en el objetivo.

Por su parte el diseño de protecciones metálicas, se basa en la premisa de crear placas que se comportan como una barrera rígida ante posibles cuerpos que puedan chocar contra su superficie. Existen una serie de factores decisivos para lograr resistir estos impactos y su posterior penetración: densidad, resistencia de los materiales y grosor final de la protección. Los avances en el tratamiento metalúrgico de los proyectiles han hecho que los diseños contemporáneos sean superiores a cualquiera de los usados en el siglo pasado.

Como es lógico, los materiales utilizados para el diseño de las protecciones determinan en gran medida los empleados en la fabricación de los proyectiles. En este sentido, durante su diseño es preferible que las estructuras presenten una gran maleabilidad, no siendo tan importante la dureza final del proyectil. Siguiendo estas pautas, los proyectiles más comunes hoy en día consisten en una bala de plomo o acero, recubierta por una coraza de cobre que lo hace de forma parcial o total (*jacketed y partial jacketed*), cilindros de plomo y postas o balas pequeñas de acero o plomo [Manzano-Trovamala et al., 2001].

Por último, la forma y tamaño del proyectil determinarán el modo de fallo y la resistencia del objetivo ante el impacto (cf. §2.1.3.1). Los proyectiles afilados muestran una acción cortante, en la cual la fractura del blanco se produce sobre el eje del proyectil. Las formas planas, por su parte, muestran una penetración a modo de tapón. La investigación realizada por Wilkins en 1978 [Wilkins, 1978], determinó que las dimensiones más influyentes en el fenómeno del impacto balístico eran el grosor de la protección y el radio del proyectil. Sus resultados confirman la relación existente entre ambos factores, junto con el ángulo, para determinar el modo de fallo de la protección sobre la que impacta el proyectil.

2.1.2.3. Espesor de los blindajes

Wilkins en 1978 [Wilkins, 1978] introdujo una primera clasificación de los blindajes en función de su espesor. Si el cociente entre el espesor y el radio es mayor que uno, se trata de un blindaje grueso, mientras que si es menor que uno se considera un blindaje fino. Las conclusiones obtenidas por Wilkins, determinaron que el conocimiento de este cociente, junto con la velocidad del impacto, permite aproximar cual será el modo de fallo que resultará del impacto. De la misma forma si el cociente de protección y radio es cercano a uno, la forma del proyectil puede influenciar en el modo de fallo.

Por otro lado, Backman y Goldsmith en el mismo año [Backman and Goldsmith, 1978], propusieron una forma alternativa de clasificar dichos blindajes en función de su espesor:

- Semi-infinito: Estas protecciones se caracterizan porque la distancia al contorno más alejado al lugar del impacto no influye en el proceso de penetración.
- Grueso: En este tipo de protección la distancia al contorno más alejado al impacto influye en el proceso de penetración, después de un sustancial recorrido dentro del mismo.
- Intermedio: En este caso la parte posterior de la superficie ejerce una influencia considerable sobre el proceso de deformación durante el movimiento del proyectil.
- Fino: Se trata de protecciones que no presentan cambios de presión ni deformaciones en todo su espesor.

2.1.2.4. Composición de los blindajes

La evolución en la composición de los blindajes se ha visto ligada históricamente a la de los proyectiles. Las modificaciones realizadas en estos últimos conlleva una constante revisión de la composición de los blindajes con el objetivo de conseguir una protección eficiente en todo momento. La clasificación cronológica de los blindajes en este apartado es la siguiente:

- Los blindajes de fundición fueron de los primeros en aparecer y estaban formados a partir de grandes planchas de acero fundido.
- Los blindajes laminados consisten en una serie de placas de acero, aluminio o aleaciones de distinta dureza y tenacidad. La primera de las capas se encarga de romper o desviar los proyectiles perforantes, mientras que las demás deben ser capaces de deformarse lo suficiente como para absorber la energía residual.
- Los blindajes espaciados, se componen de diferentes placas, generalmente acero con baños de cromo, níquel, vanadio, molibdeno, etc., con espacios vacíos entre ellas.
- Los blindajes compuestos surgieron como mejora de los laminados. Su eficacia se basa en el empleo de planchas superpuestas de diferentes materiales (acero, cerámica, aluminio, materias plásticas, etc.), cuya finalidad es desviar y frenar sucesivamente el proyectil, desestabilizándolo en cada etapa.
- Los blindajes modulares y reactivos han surgido como respuesta para los modernos proyectiles creados en los últimos años. El modular es una modificación del blindaje compuesto que permite sustituir partes de blindaje debido a su diseño en módulos. El reactivo por su parte, incorpora el concepto novedoso de pequeñas cargas explosivas entre las planchas del blindaje para desestabilizar a los proyectiles al impactar.

Por último comentar que en función del acabado superficial del blindaje su forma puede ser curva, lisa irregular, etc. La incidencia de este apartado es crucial para conseguir que los proyectiles no impacten de forma directa sobre el blindaje, consiguiendo en muchos casos, en función del acabado y del tipo de proyectil, que este último rebote sobre la superficie.

2.1.2.5. Material de los blindajes

Un apartado fundamental dentro del mecanismo de penetración, es la descripción de las características de los materiales que constituyen la protección o blindaje. Esto puede ser conseguido de forma cualitativa mediante la distinción entre sustancias naturales como la tierra, las rocas, madera y agua, materiales manufacturados como los metales, aleaciones, plásticos, cerámicas y compuestos mezclados como las gomas o cauchos.

Es también conveniente clasificar los materiales en función de su peso. Por un lado los materiales ligeros, con gravedad específica de hasta tres, incluyen madera, plásticos,

cerámicas, derivados de la tierra, varios tipos de rocas, metales de baja densidad y aleaciones. Los materiales de peso intermedio, con gravedad específica hasta ocho, incluyen la mayoría de los compuestos metálicos como el acero, cobre y estaño. Por último los materiales pesados, con gravedad específica por encima de ocho, incluyen el plomo, uranio y tungsteno.

Con respecto a la resistencia del objetivo, ésta puede ser descrita en términos de penetrabilidad para un proyectil estándar disparado bajo condiciones de impacto idénticas. En balística es muy interesante tener una clasificación de la penetrabilidad de un proyectil estándar sobre diferentes materiales. Por un lado los materiales de baja resistencia los cuales suelen estar formados por compuestos de tierra. Los materiales de resistencia media suelen ser hormigón y aleaciones metálicas de baja resistencia. Por último los materiales de alta resistencia se caracterizan por ser metales de alta resistencia, aleaciones y cerámica [Backman and Goldsmith, 1978].

Por otro lado la mayoría de los objetivos están caracterizados como sólidos, ya sea un sistema continuo o formado por partículas. Sin embargo, en algunas situaciones el blanco podría comportarse como un fluido. Esto ocurre cuando el objetivo se ve afectado por las altas presiones producidas por impactos a hipervelocidad. Las propiedades de los objetivos, especialmente en los sólidos, están determinadas por su homogeneidad, isotropía y la naturaleza de su composición, incluyendo combinaciones iniciales de sólidos/fluidos.

Algunos modelos de materiales sólidos pueden incluir uno o más comportamientos: elástico, plástico, viscoso o hidrodinámico. Las deformaciones permanentes en estos materiales ocurren cuando la presión supera los niveles críticos o el límite de elasticidad. Una especificación precisa de la respuesta de estos materiales sólidos es complicada de obtener debido a la alta presión a la que es sometido durante la penetración y por las diversas deformaciones y modos de fallos que pueden producirse.

2.1.3. Proceso de penetración

La combinación de los factores comentados con anterioridad: materiales, dimensiones, geometría, forma, velocidad y ángulo de impacto, hace que existan una amplia variedad de comportamientos posibles una vez que se produce el contacto entre el objetivo y el proyectil.

El proceso de entrada del proyectil dentro de la protección durante un impacto es denominado penetración. Desde el punto de vista puramente físico, el impacto de un proyectil sobre un blindaje o protección puede originar tres resultados distintos: perforación, parada o rebote. Se produce una perforación cuando el proyectil consigue atravesar completamente la protección (Figura 3.3, caso a). Por otro lado, cuando el proyectil entra en ella pero no consigue traspasarla se habla de parada o detención

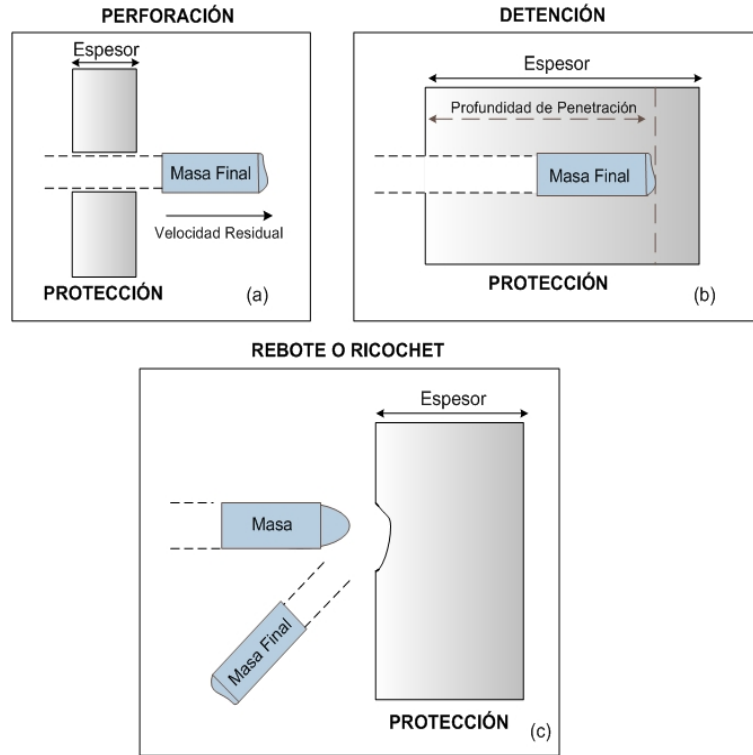


Figura 2.3: Posibles resultados del impacto de un proyectil sobre una protección

(Figura 3.3, caso b). Por último, cuando aparece un ángulo de ataque en el impacto, i.e. el impacto no es perpendicular, el proyectil puede salir rebotado, lo que se conoce como rebote o *ricochet* (Figura 3.3, caso c).

En las pruebas realizadas en laboratorio, se habla de perforación o penetración completa cuando el proyectil impactante, o cualquier fragmento del mismo o cualquier fragmento de la muestra de ensayo perfora la placa testigo. Esta placa es una lámina delgada localizada detrás y en forma paralela a la muestra de ensayo balístico usada para detectar la penetración de proyectiles.

Un parámetro de especial interés para caracterizar el fenómeno del impacto es el denominado *límite balístico*. Se trata de la velocidad mínima a la cual un proyectil particular se espera penetre completamente un blindaje de espesor y propiedades físicas dadas a un ángulo de oblicuidad especificado. El límite balístico también puede ser definido como la máxima velocidad a la cual un proyectil consigue una perforación completa pero con una velocidad de salida, velocidad residual, igual a cero. Conseguir calcular este valor de forma precisa ha sido siempre una tarea de especial dificultad. Esto es debido a varios factores como la naturaleza destructiva de los ensayos balísticos, la gran cantidad de variables que intervienen, los complejos modos de fallo que aparecen durante el proceso de penetración, la aparición de cierto grado de variabilidad y la

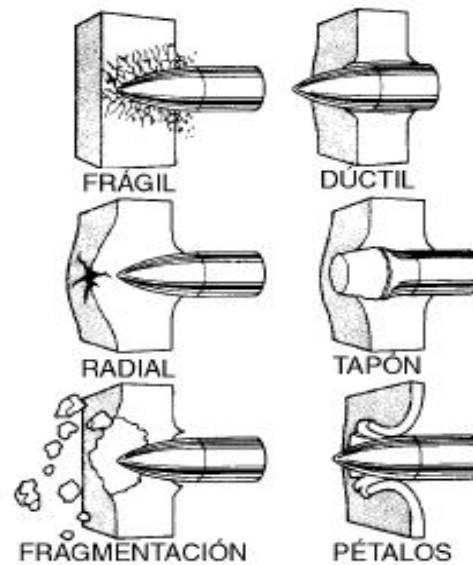


Figura 2.4: Modos de fallo más comunes en placas impactadas

dificultad de controlar la velocidad de impacto.

Un método que ha encontrado gran aceptación, sobre todo en armas de pequeño calibre, es el conocido como límite balístico v_{50} . En este caso se intenta localizar la velocidad inicial del proyectil que resultará en una penetración completa para el 50% de las muestras [Abrate, 1998]. Otros métodos menos utilizados, intentan aproximar el punto v_0 que es la máxima velocidad a la cual no ocurrirá una penetración completa o aproximar el punto v_{100} , que es la mínima velocidad a la cual todos los proyectiles penetrarán completamente.

En general, la interacción a alta velocidad de un proyectil cilíndrico contra un blanco metálico suele estar dividida en tres partes.

1. La fase de impacto donde las ondas de tensión y deformación son críticas para iniciar la penetración.
2. En esta fase tanto el proyectil como la placa se comportan como fluidos.
3. La tercera fase es la penetración del proyectil dentro de la protección metálica.

2.1.3.1. Modos de fallo

Durante la penetración de un objetivo por un proyectil, tienen lugar diferentes mecanismos de fallo en la protección o blindaje. Sin embargo, no es posible definir una

conducta general de fallo en el proceso de impacto debido a los numerosos factores que intervienen como la diversa geometría de los proyectiles, la gran cantidad de materiales empleados en el diseño de las protecciones y el amplio abanico de velocidades posibles.

Los tipos más frecuentes de fallos en materiales sólidos se muestran en la Figura 2.4, adaptada a partir de los trabajos de Backman y Goldsmith [Backman and Goldsmith, 1978] y Zukas [Zukas et al., 1982]. Cada uno de los tipos de fallos aparece de forma representativa ante blindajes de ciertas características. Los materiales frágiles y de baja densidad suelen presentar fallos del tipo: fractura frágil, fractura radial y fragmentación. En todos ellos, la fractura se produce debido a las ondas iniciales de tensión. Por otra parte, en los blindajes formados por placas de espesor delgado y materiales de mayor dureza destaca la aparición de fallos como el taponamiento. Éste ocurre una sección de la placa es arrancada alrededor del movimiento del proyectil. El fallo dúctil, por el contrario, aparece si el blindaje contiene placas gruesas de elevada dureza, donde la perforación es acompañada por una expansión radial del material de la placa. Por último, el fallo basado en pétalos, es frecuente que se produzca en planchas finas ante proyectiles cónicos a velocidades relativamente lentas. Como el material se deforma en su parte trasera más que el propio proyectil, se exceden las propiedades elásticas del material y comienza a romperse alrededor de la punta del proyectil.

No obstante, en un impacto real es normal que se produzca una combinación de los mismos sobre el material o materiales que forman el blindaje.

2.1.4. El blindaje ligero en la actualidad

La protección de personas y estructuras para prevenir los daños causados por los impactos de proyectiles de diferentes tipos, que pueden desplazarse en amplios rangos de velocidad, es una de las principales preocupaciones hoy en día, tanto en aplicaciones militares como civiles [Radin and Goldsmith, 1988]. En sus orígenes, el concepto de blindaje estaba estrechamente relacionado con el ámbito militar, utilizándose para dotar de diferentes grados de protección a los vehículos de este campo. Sin embargo en los últimos años, su uso se ha ampliado al ámbito civil o comercial debido a las necesidades de protección existentes en la sociedad actual.

Uno de los factores fundamentales del diseño de blindajes destinados a vehículos, es la capacidad de movilidad asociada a la protección. Un punto de especial importancia en este sentido, es el peso final de la estructura a proteger como resultado del blindaje elegido y los efectos de éste sobre su maniobrabilidad. En el ámbito militar esta consideración tiene un interés menor en algunos casos, e.g. tanques y naves, ya que su movilidad es reducida debido a su diseño original. Gracias a esto se puede optar por equiparlos con fuertes blindajes de peso significativo y alta capacidad de protección, conocidos como blindajes pesados. Lógicamente cuando la estructura a proteger es estática, el peso

puede no tenerse en cuenta durante la fase de diseño de la protección, recurriendo por tanto a protecciones de este tipo. Sin embargo en otros entornos, militares y civiles, la movilidad es un factor determinante por lo que es necesario prestar atención al peso final del conjunto, utilizando en este caso blindajes ligeros. Esto se une además a la creciente necesidad de mejorar el rendimiento y reducir los costes asociados a la construcción de blindajes, por lo que se están utilizando cada vez con mayor asiduidad blindajes ligeros laminados. Ejemplos representativos de este concepto es la construcción de protecciones personales, donde destaca la fabricación de chalecos antibalas, o la fabricación de elementos para la industria aeroespacial, como cabinas de aviones o helicópteros militares.

Desde el punto de vista de la efectividad, en sus comienzos los blindajes solo eran útiles contra armas ligeras. Estaban contruidos mediante chapas de acero remachado, las cuales al recibir impactos proyectaban fragmentos metálicos hacia el interior del vehículo. Con el paso de los años se mejoró su diseño, pasando a crearse blindajes de láminas de metal con espesores cercanos a los 100mm. Sin embargo la tendencia de mejorar la protección añadiendo láminas más gruesas aumentaba el peso lo que reducía la movilidad del vehículo. Desde entonces, se han ido incorporando otras formas de blindaje, incorporando espacios vacíos entre las láminas y materiales como cerámicas o uranio empobrecido junto con el acero. Hasta entonces se había recurrido únicamente a planchas monolíticas de cierto metal y de un espesor determinado.

En la actualidad el blindaje laminado, el más común en el ámbito militar sobre todo en carros de combate y vehículos blindados ligeros. Se diseña para resistir la penetración frente proyectiles de alta velocidad, con formas esbeltas y puntiagudas. Su composición en una serie de placas de acero u otros metales de distinta dureza y tenacidad. La primera placa se encarga de romper o desviar los proyectiles perforantes, mientras que las demás deben de ser capaces de deformarse lo suficiente como para absorber la energía residual. Esta capacidad para soportar impactos en el ámbito balístico, se evalúa normalmente en función del límite balístico de referencia V_{50} , que se define como la velocidad en la que existe un 50 % de posibilidades de detención para un determinado tipo de proyectil.

La introducción del aluminio en el conjunto de los materiales de protección balística dio lugar a una mejora considerable en el diseño de blindajes compuestos por varios materiales. El aluminio se había mostrado como un material apropiado para absorber los impactos de metralla dada su ductilidad y para la construcción de vehículos dada su ligereza. El problema que presenta es su vulnerabilidad frente al impacto de proyectiles de núcleo duro como el acero o wolframio, lo que llevo que en algunos casos se añadiera al aluminio una capa exterior de acero.

Otros tipos de blindajes de gran aceptación son los blindajes compuestos con componente cerámico, ya que han demostrado grandes resultados en los diseños donde se han incluido. Sin embargo, su coste asociado es mayor que el de las placas metálicas por lo que su utilización es más reducida por parte de la industria [Viechnicki et al., 1991].

Como se ha podido constatar, los diferentes tipos de blindajes metálicos han demostrado sobradamente su eficacia en múltiples problemas de impacto balístico y hoy en día son insustituibles [Zukas, 1990]. Dentro de estos, el acero y el aluminio son los materiales más utilizados para la construcción de blindajes y protecciones tanto militares como civiles, destacando su uso en la fabricación de vehículos y aviones. Su amplia difusión se debe a su gran rendimiento balístico y al bajo coste asociado.

2.1.5. Metodologías clásicas para modelar el problema

El impacto y los problemas relacionados con el mismo han sido un punto de investigación relevante durante décadas, llevándose a cabo esfuerzos considerables para comprender desde el punto de vista físico y describir desde el matemático el fenómeno que tiene lugar durante la penetración balística [Jonas and Zukas, 1978, Zukas et al., 1982, Brown, 1986]. Este fenómeno fue descrito por Zukas, centrándose en la dinámica del impacto para el rango de la alta velocidad, al comienzo de los años noventa [Zukas, 1990].

Hasta hace un par de décadas, el diseño de estructuras que no fuesen vulnerables a impactos de proyectiles se realizaban tradicionalmente de forma empírica, confiando en pruebas de impacto reales con proyectiles y blancos específicos. Esto se debía a la complejidad matemática de solucionar las ecuaciones que aparecen durante un impacto y el relativo desconocimiento del comportamiento mecánico de los materiales en altos rangos de tensión, lo que desalentaba el intento de cualquier simulación del problema [Goldsmith and Finnegan, 1971, Averbuch and Bodner, 1974]. Las múltiples investigaciones experimentales realizadas, permitieron conseguir avances importantes en el intento de modelar el fenómeno del impacto y su posterior penetración, apareciendo en la literatura científica múltiples referencias. Una amplia revisión de los estudios realizados por la comunidad científica, puede encontrarse en la publicación realizada por Borvik et al. en el año 1999 [Borvik et al., 1999].

A través de un estudio estadístico, estos resultados puramente experimentales permitían establecer curvas de probabilidad de perforación en función de la velocidad de impacto. Sobre éstas se definían diferentes valores de velocidad de límite balístico, V_p . En ellas se especificaba el valor crítico por debajo del cual existe una probabilidad de perforación inferior a un cierto valor de velocidad P (cf. Figura 2.5).

Si bien las conclusiones que se pueden obtener a través de este método empírico son fiables, los resultados sólo son válidos con los parámetros probados en el experimento, y estos no son fácilmente extrapolables. Cualquier variación de la velocidad del impacto o de las características del proyectil o del blanco invalida los datos de prueba, y con ello los resultados. Es también un procedimiento costoso, ya que las pruebas en estas condiciones requieren instalaciones y equipos sofisticados, aumentando su coste cuanto más se desean

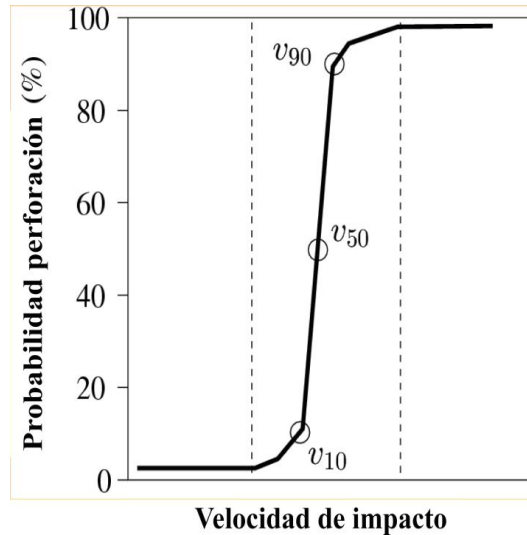


Figura 2.5: Curva de probabilidad de perforación frente a la velocidad de impacto para un proyectil-protección determinado

afinar los descubrimientos a realizar con dichas pruebas.

Con el objetivo de disminuir los gastos asociados a estas simulaciones se desarrollaron algunas técnicas semi-empíricas que permitían extender la validez de las mismas a otras estructuras. Éstas se fundamentan en ecuaciones algebraicas con cierta base teórica, las cuales calibradas con una gran cantidad de datos experimentales permiten cierta interpolación o incluso extrapolación de resultados.

La necesidad de herramientas de diseño para simular este proceso, ha desencadenado en los últimos años el desarrollo de un gran número de modelos de diferentes tipos, todos ellos pertenecientes a dos categorías: modelado analítico y simulación numérica. De este modo, el uso de costosos test experimentales se ha relegado para la última fase del diseño, ya que todas las etapas previas pueden ser cubiertas usando estas herramientas de simulación.

En el modelado analítico, se asumen una serie de hipótesis simplificadas para recrear el mecanismo físico que define el comportamiento de los sólidos, produciendo mediante ecuaciones simples un modelo que permite solventar en pocos minutos o incluso segundos el problema tratado. Estos métodos no son tan precisos como los numéricos, pero son tan rápidos que pueden calcular en muy poco tiempo un gran número de variantes [Awerbuch and Bodner, 1974, Ravid and Bodner, 1983]. En este sentido, el trabajo publicado por Ben-Dor et al. en 2005 ofrece una amplia revisión de los últimos avances relacionados con el modelado analítico [Ben-Dor et al., 2005]. Estas herramientas, también conocidas como modelos de ingeniería, son útiles en el diseño de blindajes o protecciones; ya que además de su rapidez y sencillez, ofrecen información detallada del proceso de penetración

del blindaje.

Por su parte, los métodos de simulación numérica se basan en la utilización de programas de elementos finitos o de diferencias finitas para la simulación del proceso completo de penetración del proyectil (*hydrocodes*). Estos métodos conducen al resultado del problema a través de la solución completa de las ecuaciones diferenciales de la mecánica de medios continuos, siendo mucho más fiable este resultado que el obtenido con el modelado analítico. En el caso de los blindajes metálicos han demostrado sobradamente su eficacia [Zukas, 1990, 2004]. Sin embargo, su capacidad de aproximación a la realidad depende más de las ecuaciones constitutivas usadas para los materiales, que en los errores inherentes en el propio método numérico [Anderson, 1987, Anderson and Bodner, 1988]. No obstante, son pocos los códigos numéricos comerciales que pueden abordar este tipo de simulaciones dado su marcado carácter no lineal: contactos, deformación finita o comportamiento inelástico de los materiales. Su principal desventaja es la gran cantidad de tiempo preciso para solucionar cada caso de impacto de un proyectil, siendo necesario generalmente horas o incluso días de proceso. Su alto coste y elevado tiempo de ejecución, necesario para una sola simulación, obstaculiza el poder realizar un número de cálculos suficiente para conseguir optimizar los resultados.

Por último, si bien durante años se han propuesto una gran cantidad de modelos analíticos para la resolución del problema del impacto balístico, la complejidad de los eventos que participan han limitado el uso general de soluciones de este tipo. Por ello, los modelos analíticos para estos fenómenos han sido construidos generalmente solo para situaciones idealizadas, concentrándose en un solo modo de deformación o fallo. Es por tanto preferible usar los métodos numéricos para intentar resolver esta clase de problemas. Una revisión de los mismos, asociados al fenómeno del impacto, fue publicada por Borvik et al. en 1999 [Borvik et al., 1999].

2.1.5.1. Simulación Numérica

En muchos problemas de ingeniería existe una gran dificultad para obtener una solución exacta debido a la alta complejidad asociada a las ecuaciones diferenciales o las dificultades inherentes a las condiciones de contorno iniciales. Con el fin de superar estas barreras y lograr resolver el problema con la ayuda de computadoras, los ingenieros han optado por presentar el problema de una manera puramente algebraica mediante aproximaciones numéricas. En contraste con las soluciones analíticas, que reflejan el comportamiento exacto de un sistema en cualquier punto del mismo, las soluciones numéricas aproximan la solución exacta solo en puntos discretos, llamados nodos. Por tanto, el primer paso necesario en cualquier procedimiento numérico es la discretización, este proceso divide el medio de interés en un número de pequeñas subregiones y nodos. Mediante este proceso de discretización, el conjunto infinito de números que representan la función o funciones incógnitas es reemplazado por un número finito de parámetros incógnita. En función del proceso de discretización elegido, existen diferentes formas de métodos numéricos, destacando entre estos el método de las diferencias finitas y método

de los elementos finitos.

El método de diferencias finitas, define una ecuación diferencial para cada nodo y las derivadas son reemplazadas por aproximaciones en diferencias finitas, convirtiendo entonces un problema de ecuaciones diferenciales en un problema algebraico fácilmente resoluble por medios comunes (especialmente matriciales). Si bien este método presenta las ventajas de su facilidad de comprensión y utilización en problemas simples, manifiesta ciertas dificultades al aplicarlo a geometrías o condiciones de contorno complejas. Esta situación es real para problemas con materiales con propiedades no isotrópicos, i.e. aquellos que no tienen iguales propiedades en todas las direcciones.

En contraste, el método de los elementos finitos recurre a formulaciones integrales en lugar de ecuaciones diferenciales para crear un sistema de ecuaciones algebraicas. En esencia se trata de una técnica que sustituye el problema diferencial por otro algebraico, aproximadamente equivalente, para el cual se conocen técnicas generales de resolución. Para ello hace uso de la discretización o subdivisión de una región sobre la cual están definidas las ecuaciones en formas geométricas simples denominadas elementos finitos.

Dentro del ámbito de esta investigación, la simulación mediante códigos numéricos del impacto entre sólidos en el rango de la alta velocidad ($v > 500$ m/s), constituye uno de los problemas de diseño de elementos estructurales que mayor coste computacional presenta [Anderson, 1987]. Confluyen en éstos numerosas causas de no-linealidad, tales como comportamiento plástico del material, grandes deformaciones y contactos, que aumentan considerablemente los tiempos de cálculo. Así mismo, los elevados gradientes que aparecen en las variables obligan a emplear mallas muy refinadas y la gran distorsión de elementos precisan habitualmente de un remallado. Todo ello hace que las simulaciones numéricas requieran de herramientas de diseño extremadamente sofisticadas (no más de 3 ó cuatro códigos comerciales son capaces de abordar este tipo de problemas (LS-DYNA¹, PAM-CRASH², ABAQUS³) y computadoras de gran capacidad de cálculo.

Uno de los programas comerciales más usados en la actualidad para el modelado y la simulación es ABAQUS/Explicit v6.4.1. Se trata de un código de elementos finitos explícito, ampliamente empleado en la simulación de problemas dinámicos con un alto grado de no linealidad como el que se pretende resolver [Oguibe and Webb, 1999, Arias et al., 2007, Zhou and Stronge, 2007].

Así pues, la realización de un elevado número de simulaciones tendentes a optimizar parámetros geométricos o de material, práctica habitual durante el diseño de protecciones frente a impacto balístico, obliga a destinar numerosos recursos.

¹<http://www.ls-dyna.com/>

²<http://www.esi-group.com/SimulationSoftware/NumericalSimulation/index.html>

³http://www.simulia.com/products/abaqus_explicit.html

Mientras que un componente estructural convencional, suele ser diseñado para que sea capaz de soportar las cargas sin que se produzca plastificación en ninguno de sus puntos, los sistemas de protección frente a impacto deben disipar energía del proyectil mediante grandes deformaciones plásticas. Ésta es la fuente principal de las no-linealidades comentadas anteriormente y del elevado coste computacional que conlleva su simulación [Zaera and Sanchez-Galvez, 1998, Zaera, 2001].

2.1.5.2. Método de Elementos Finitos

El Método de los Elementos Finitos o MEF, es un método de simulación numérica usado para la aproximación de soluciones de ecuaciones diferenciales parciales. Su aplicación es de gran utilidad en diversos problemas de ingeniería y física, destacando de forma especial en aquellos relacionados con la ingeniería mecánica [Stasa, 1984, Zienkiewics et al., 2005]. Debido a la complejidad creciente asociada a los análisis que se realizan en todas las ramas de la ingeniería industrial, el cálculo asistido por ordenador ha ido adquiriendo mayor protagonismo con el paso del tiempo. En este sentido, las simulaciones numéricas a través de mecanismos computacionales son superiores a las experimentaciones en flexibilidad, mejorando en precisión y velocidad los resultados para fines prácticos.

Entre los métodos numéricos, el MEF es superior al resto debido a su versatilidad para modelar defectos de forma arbitraria y por la naturaleza anisotrópicas del material [Liu et al., 2002]. Cuando se trata de un problema de mecánica de sólidos deformables o más generalmente un problema de mecánica de medios continuos, este método permite calcular el campo de desplazamientos, para posteriormente y a través de relaciones cinemáticas y constitutivas obtener las deformaciones y tensiones relacionadas.

Este método es muy usado debido a su generalidad y a la facilidad de introducir dominios de cálculo complejos, en dos o tres dimensiones. Dada la imposibilidad práctica de encontrar la solución analítica de problemas ingenieriles relacionados con el ámbito de la mecánica de materiales, con frecuencia en la práctica los métodos numéricos y en particular los MEF, han sido durante mucho tiempo la única alternativa práctica de cálculo.

Sin embargo, en los últimos años se ha empezado a considerar a los técnicas de RNA como un complemento e incluso en algunos entornos como una alternativa al MEF. Esto es debido a la superioridad de ambos sistemas, para el modelado y resolución de problemas complejos, frente a las soluciones matemáticas clásicas y procedimientos tradicionales. Asimismo las similitudes estructurales que comparten, con elementos distribuidos, ha llevado a los investigadores a utilizar las RNA para paliar las deficiencias que presenta el MEF en algunos tipos de problemas [Takeuchi and Kosugi, 1993] (cf. §2.5.2).

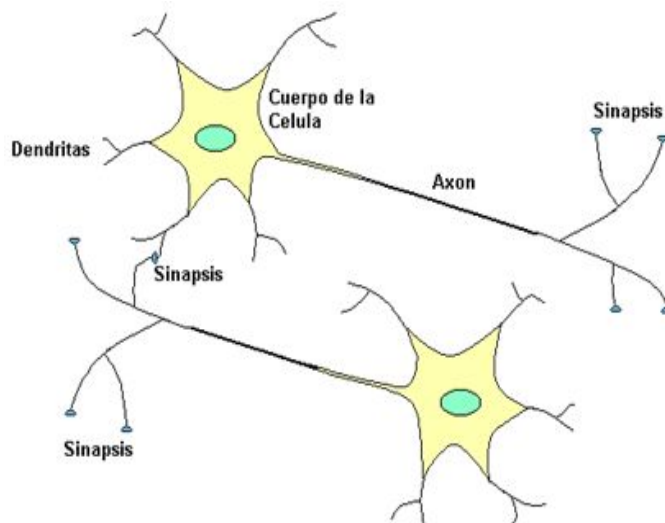


Figura 2.6: Esquema de una neurona biológica

2.2. Redes de Neuronas Artificiales

2.2.1. Modelos biológicos

La teoría y modelado de las RNA está inspirada en la estructura y funcionamiento de los sistemas nerviosos biológicos donde la neurona es el elemento fundamental. Las bases del sistema nervioso, tal y como se conocen en la actualidad, fueron fijadas por el científico español Santiago Ramón y Cajal, el cual a lo largo de su carrera investigadora consiguió demostrar los mecanismos que gobiernan la morfología y los procesos conectivos de las células nerviosas.

Según estas premisas, el cerebro humano continuamente recibe señales de entrada de muchas fuentes y las procesa para crear una salida apropiada. El cerebro cuenta con millones de neuronas que se interconectan para elaborar redes neuronales. Estas redes ejecutan los millones de instrucciones necesarias para mantener una vida normal [Ramon-y Cajal, 1894]. Dentro de las redes, las neuronas son las células que forman la corteza cerebral de los seres vivos, cada una está formada por elementos llamados cuerpo, axón y dendritas (cf. Figura 2.6).

Las dendritas forman una estructura de filamentos muy fina que rodean el cuerpo de la neurona. Se trata de sus prolongaciones terminales y están implicadas en la recepción de los estímulos, pues sirven como receptores de impulsos nerviosos provenientes desde un axón perteneciente a otra neurona. El axón es una prolongación filiforme de la

célula nerviosa, a través de la cual viaja el impulso nervioso de forma unidireccional, estableciendo contacto con otra célula mediante ramificaciones terminales. Las sinapsis por su parte, son uniones especializadas mediante las cuales las células del sistema nervioso envían señales de unas a otras, permitiendo a las neuronas del sistema nervioso central formar una red de circuitos neuronales.

Una de las características que diferencian a las neuronas del resto de células vivas es su capacidad para comunicarse. Para ello utilizan impulsos eléctricos y reacciones químicas, lo que facilita que cada neurona pueda intercambiar información con las demás. Éstos viajan por el axón que hace contacto con las dendritas de la neurona vecina mediante la sinapsis.

Las RNA son una abstracción funcional de estas estructuras neuronales biológicas del sistema nervioso central [Arbib, 1995]. Como su nombre indica, una RNA consiste en una red de neuronas artificiales interconectadas. El concepto se basa en el funcionamiento del cerebro animal, donde existen sistemas de células interconectadas, las cuales son aparentemente responsables de la conciencia, memoria y pensamientos. Las neuronas consiguen realizar estas funciones gracias, entre otras cualidades, a su plasticidad, i.e. habilidad de cambiar de respuesta a los estímulos en el tiempo y aprender, siendo esto precisamente lo que intenta imitar una RNA mediante el software de un ordenador.

Para establecer una analogía entre ambas, neurona biológica y artificial, se pueden fijar los siguientes aspectos: las señales que llegan a la sinapsis son las entradas a la neurona, éstas pueden ser ponderadas (atenuadas o amplificadas) a través de un parámetro, denominado peso. La señal que se transmite a la neurona puede ser inhibirla (peso negativo) o excitarla (peso positivo). El efecto es la suma de las entradas ponderadas, la cual si es mayor o igual que el umbral de la neurona, entonces produce que la salida de la neurona se active. Este funcionamiento conlleva situaciones de todo o nada, cada una de las neuronas se activa o no se activa.

Por otro lado, la conectividad entre las células del cerebro es muy elevada. A través de las investigaciones realizadas en el campo de la neurofisiología, se calcula que en un cerebro existen alrededor de 100 billones de neuronas, conectadas cada una de ellas con alrededor de 10000 neuronas, lo cual forma una red de proporciones considerables [Adeli and Yeh, 1989]. Un dispositivo de características similares es difícil de fabricar con la tecnología actual, consiguiéndose sistemas varios órdenes de magnitud por debajo.

Con este alto grado de complejidad, no es de extrañar por tanto que las similitudes entre los sistemas biológicos y artificiales sean en muchos casos superficiales. Sin embargo los procesos cognitivos procedentes de la neurobiología y psicología, han sido tomados como punto de partida por la comunidad investigadora para desarrollar las diferentes arquitecturas de RNA existentes.

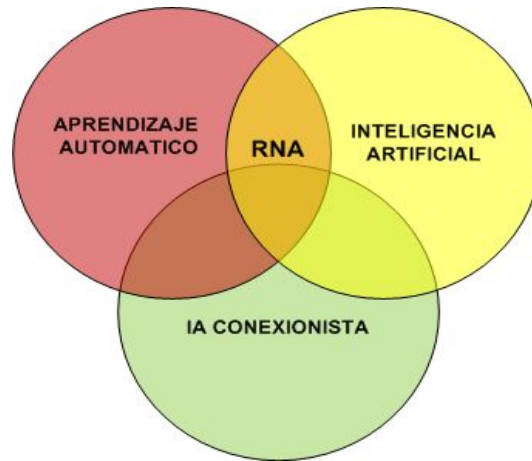


Figura 2.7: Áreas de interés relacionadas con las RNA

2.2.2. Redes de Neuronas Artificiales e Inteligencia Artificial

El término *Inteligencia Artificial* (IA) fue formalizado durante la Conferencia de Dartmouth en 1956, en la que se reunieron los padres fundadores de esta disciplina, entre los que destacan John McCarthy (1927), Marvin Minsky (1927) y Claude Shannon (1916-2001).

El propósito de este congreso era reunir a un grupo de investigadores que quisieran trabajar sobre la conjetura de que cada aspecto del aprendizaje y cada característica de la inteligencia, podían ser tan precisamente descritos que se podrían crear máquinas que los simularan. Este congreso supuso la postulación de los objetivos prioritarios de la IA: “*entender la inteligencia natural humana, y usar máquinas inteligentes para adquirir conocimientos y resolver problemas considerados como intelectualmente difíciles*” [Klassner, 1996].

Para resolver estas cuestiones, se han seguido dos enfoques fundamentales que dan lugar a los diferentes modos de trabajo en IA: el enfoque de emulación, conocido como conexionista, y el enfoque de ejecución, denominado también simbolista o IA clásica. En el primero, sus representantes pretenden emular el cerebro, incluida, si fuera posible, su estructura. En el segundo, sus integrantes pretenden crear sistemas cuyo comportamiento sea tal que, si lo llevase a cabo una persona, sería considerada como inteligente [Borrajo et al., 1993].

A pesar de que ambas escuelas, simbolista y conexionista, emplean al menos en una gran parte técnicas similares, su intención y objetivos están claramente diferenciados. La concepción simbolista se basa en que la inteligencia puede reducirse al procesamiento simbólico de la información, i.e., la manipulación de símbolos de acuerdo con ciertas reglas. Por el contrario, la hipótesis conexionista rechaza la idea de que la inteligencia

pueda reducirse a la suma de procesos elementales basados en el manejo de símbolos. Esta hipótesis parte de la existencia de una red de neuronas, esto es, una densa red formada por componentes, denominados neuronas por analogía con el cerebro, interconectados con otros componentes similares. Los conexionistas parten del estudio de las propiedades computacionales de las neuronas y de las redes construidas con ellas, y tratan de averiguar la forma en que estas redes deben ser construidas para lograr sistemas inteligentes capaces de aprender.

Dentro de la rama conexionista, destaca el área de aprendizaje automático, cuyo objetivo es desarrollar técnicas que permitan a las computadoras aprender. Gracias a esto, se consiguen desarrollar sistemas capaces de generalizar comportamientos a partir de una información no estructurada suministrada en forma de ejemplos. Esta capacidad de aprendizaje es considerada una de las características más apreciables en un sistema inteligente, siendo las técnicas basadas en RNA unas de las más utilizadas por los científicos e investigadores (cf. Figura 2.7). Su empleo tan prolífico es debido a que las RNA permiten solventar uno de los objetivos centrales de la IA: construir sistemas que sean capaces de adaptarse de forma dinámica a situaciones nuevas y aprender como resultado de resolver el problema que estas situaciones presentan.

Un resumen de las principales características de estas dos corrientes, simbolista y RNA, puede observarse en la Tabla 2.1.

IA simbolista	RNA
Enfoque descendente	Enfoque ascendente
Basado en la psicología	Basado en la biología
Qué hace el cerebro	Cómo lo hace el cerebro
Reglas Si/Entonces	Generalización a partir de ejemplos
Sistemas programados	Sistemas entrenados
Lógica, conceptos y reglas	Reconocimiento de patrones
Arquitectura Von Neumann. Separación hardware/software	Arquitecturas paralelas, distribuidas, adaptativas. Autoorganización

Tabla 2.1: Características de la IA clásica frente a las RNA

Durante la década de los setenta, la corriente simbólica cobró un gran auge en detrimento de la corriente conexionista, más concretamente de las RNA, que se vio sumida en un declive que duraría más de diez años. Esto fue debido, principalmente, al estudio presentado por Minsky y Papert en 1969 [Minsky and Papert, 1969], en el que mostraron las graves limitaciones de la arquitectura Perceptrón, el modelo neuronal por excelencia de los años sesenta, lo que supuso una enorme pérdida de confianza en este nuevo campo.

Ya a finales de los años 60 se señaló como solución a las limitaciones del Perceptrón la introducción de capas ocultas en la red, pero el problema estribaba en que no se disponía

de ningún algoritmo de aprendizaje que permitiera obtener los pesos sinápticos en una red multicapa con neuronas ocultas. Esto no se consiguió hasta mediados de los años 80, cuando D. E. Rumelhart et al. presentaron el algoritmo de Retropropagación (RP) de errores o *Backpropagation* (BP) [Rumelhart et al., 1986]. Gracias a este algoritmo comenzó el resurgimiento de las RNA ya que solventaba las limitaciones de la arquitectura Perceptrón para resolver problemas de separabilidad no lineal (cf. §2.3.3).

Hoy en día, vuelven a coexistir estas dos corrientes, conexionista y simbolista, dentro de la búsqueda de inteligencia, ya que ambas son alternativas complementarias que completan la IA en sus tres facetas fundamentales: artificial, biológica y computacional [Martin-del Brio and Sanz-Molina, 2006].

2.2.3. Definición de una Red de Neuronas Artificial

Las RNA son un paradigma de aprendizaje y procesamiento automático, inspirado en el funcionamiento del sistema nervioso biológico. Son además consideradas como un modelo matemático, compuesto por un gran número de elementos o unidades de procesamiento, llamados neuronas, que trabajan al mismo tiempo para encontrar solución a problemas específicos. En cuanto a su estructura se trata de un sistema de interconexión de neuronas, organizado en red y distribuido en niveles, que colaboran para producir un estímulo de salida.

Quizás una de las definiciones que mejor explica las características, estructura y pretensiones de las RNA es la que dio Kohonen en 1988 [Kohonen, 1988]: “*Las Redes Neuronales Artificiales son redes de elementos simples, usualmente adaptativos, y de sus organizaciones jerárquicas interconectadas masivamente en paralelo, que se pretende interactúen con los objetos del mundo real de la misma manera que lo hacen los sistemas nerviosos biológicos.*”

Durante los últimos años las RNA han constituido un centro focal de intensa actividad, dando lugar a una maduración de las ideas primitivas desarrolladas en los años 60 [Minsky and Papert, 1969]. La principal cualidad de estos sistemas es su adaptabilidad dinámica, i.e., su capacidad para variar de comportamiento en situaciones cambiantes [Ellias and Grossberg, 1975], utilizando para ello técnicas como el aprendizaje, generalización o auto-organización. Estas técnicas se inspiran en el modelo de la neurona biológica, recreándolas mediante unidades elementales de procesamiento o neuronas artificiales que mimetizan algunas de las características de las primeras. Dichas neuronas se agrupan formando estructuras más o menos complejas, denominadas *Redes de Neuronas Artificiales*. El comportamiento global de dicha red determina su capacidad para ensayar hipótesis, detectar patrones estadísticos y regularidades o ajustar dinámicamente un modelo implícito implementado en la misma arquitectura [Grossberg, 1988]. Dicho comportamiento va más allá de la suma de las potencialidades de las neuronas que la componen. En estos sistemas la información no se almacena en un emplazamiento físico

único. Por el contrario, la información aparece distribuida por toda la estructura de la red, concentrándose en las uniones de los distintos elementos [Pao, 1989].

Esta es la base del procesamiento distribuido, las funciones proporcionadas en estos sistemas, vienen dadas por la globalidad del sistema más que por la aportación individual de cada elemento. El resultado es la aparición de propiedades nuevas que pertenecen al sistema como un todo.

La segunda cualidad de una RNA es su capacidad de aprendizaje, que permite a la red modificar su propia estructura (matriz de pesos) adaptándola hasta conseguir un algoritmo de ejecución. Este algoritmo se construye a partir de las características extraídas de los ejemplos con los que se realiza el entrenamiento.

Otro aspecto de gran importancia, es que esta nueva forma de computación es capaz de manejar las imprecisiones e incertidumbres que aparecen al resolver problemas relacionados con el mundo real, ofreciendo soluciones robustas y de fácil implementación.

La utilización de las RNA puede orientarse para la resolución de problemas prácticos como la clasificación de patrones, la predicción de funciones y la modelización. Su paralelismo de cálculo, memoria distribuida y adaptabilidad al entorno, han convertido a estas estructuras en potentes instrumentos con capacidad para aprender relaciones entre variables sin necesidad de imponer presupuestos o restricciones de partida en los datos. Gracias a esto, las RNA han podido ser aplicadas de forma satisfactoria en la resolución de problemas en diferentes áreas de conocimiento: biología, medicina, economía, ingeniería, psicología, etc., obteniendo excelentes resultados respecto a los modelos derivados de la estadística clásica.

2.2.3.1. Modelo matemático

De manera análoga al funcionamiento de las neuronas biológicas, las neuronas artificiales se distribuyen en capas unidas mediante conexiones que tienen asociadas un peso, W_{ij} , donde el subíndice i corresponde a la neurona receptora, mientras que el subíndice j identifica a la emisora. Cada neurona artificial i -ésima (descrita en la Figura 2.8) recibe las señales X_j de las adyacentes, provenientes la capa anterior. Si el peso es positivo, la conexión se denomina excitadora; mientras que si es negativo, se denomina inhibitoria. Con unos valores adecuados en los pesos, una RNA puede modelar prácticamente cualquier función. Cada nodo de la red multiplica sus entrada por el peso asociado la conexión, lo que se conoce como función o regla de propagación. Normalmente, como simplificación, se considera que el efecto de cada señal es aditivo, de tal forma que la entrada neta que recibe una neurona, Net_i , es la suma de cada señal individual por el

valor de la sinapsis que conecta ambas neuronas:

$$Net_i = \sum_{j=1}^N (W_{ij}X_j) \quad (2.2)$$

donde: X_i se corresponde con cada uno de los valores de entrada a la neurona S y W_{ij} es el peso asociado a cada una de ellas.

A esta función de propagación se le suele sustraer lo que se denomina valor umbral (u_i). Este permite controlar la activación de cada neurona en cuestión, dando como resultado el llamado potencial sináptico de la neurona (S):

$$S = \sum_{j=1}^N (W_{ij}X_j) - u_i \quad (2.3)$$

De igual forma que es necesaria una regla que combine las entradas de una neurona con los pesos de las conexiones, también se requiere una regla que integre las entradas con el estado actual de la neurona, produciendo así un nuevo estado de activación. Así para cada neurona, dado el estado de activación $a_i(t)$ y la entrada total que llega, Net_i , el siguiente estado de activación, $a_i(t + 1)$, se obtiene aplicando una función F , llamada función de activación:

$$a_i(t + 1) = F(a_i(t), Net_i) \quad (2.4)$$

La función de activación tiene como función limitar la amplitud de la salida de la neurona. Generalmente, el rango de amplitud de salida de una neurona se normaliza en un intervalo cerrado unitario $[0,1]$ o alternativamente $[-1,1]$. En la mayoría de los casos la función F es la función identidad, por lo que el estado de activación en $t + 1$ coincidirá con el Net de la misma t . En este caso, el parámetro que se le pasa a la función de salida φ será directamente el Net . Además el estado de activación anterior no se tiene en cuenta, por lo que la salida de una neurona, y_i , quedará según la expresión:

$$y_i(t + 1) = \varphi(Net_i) = \varphi\left(\sum_{j=1}^N (W_{ij}y_j(t))\right) \quad (2.5)$$

Por tanto, y en lo sucesivo, se considera sólo la función φ , conocida como función de transferencia o de activación. Asimismo, es necesario ajustar esta función mediante el umbral de activación de la neurona i:

$$y_i(t + 1) = \varphi(Net_i) - u_i = \varphi\left(\sum_{j=1}^N (W_{ij}y_j(t)) - u_i\right) \quad (2.6)$$

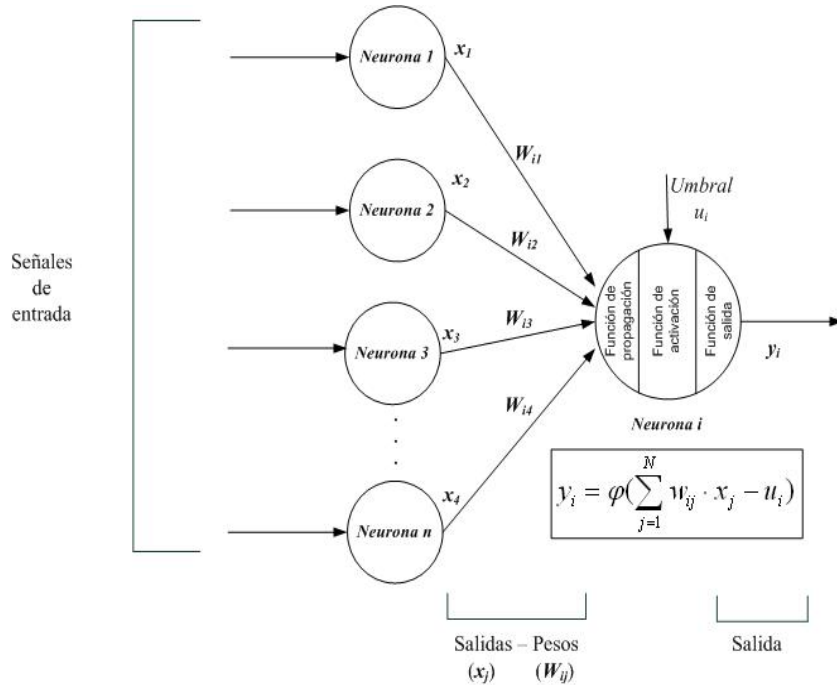


Figura 2.8: Unidad de Proceso Elemental o Neurona Artificial

Esta salida transmite su valor al resto de unidades mediante las conexiones existentes en la arquitectura de la red (cf. Figura 2.8). El valor obtenido depende de la función de activación seleccionada para cada neurona, siendo las más utilizadas las siguientes:

- **Función escalón:** Se usa cuando las salidas de la red son binarias, ya que cuando la suma de las entradas (Net_i) es mayor o igual que el umbral de la neurona (u_i), la activación es uno, y si es menor, la activación es cero (cf. Figura 2.9):

$$y(t+1) = \begin{cases} 0 & \text{si } Net_i < u_i \\ 1 & \text{si } Net_i \geq u_i \end{cases} \quad (2.7)$$

- **Función lineal o identidad:** El estado de activación coincide con la entrada total que ha llegado a la neurona (cf. Figura 2.9):

$$y(t+1) = Net_i - u_i \quad (2.8)$$

- **Función lineal-mixta:** Presenta un límite inferior y otro superior, lo que controla la salida dentro de unos valores prefijados. Si la suma de las señales de entrada es menor que el límite inferior, la activación es cero. Si es mayor que el límite superior, es uno, y si está comprendida entre ambos límites, se produce la función identidad

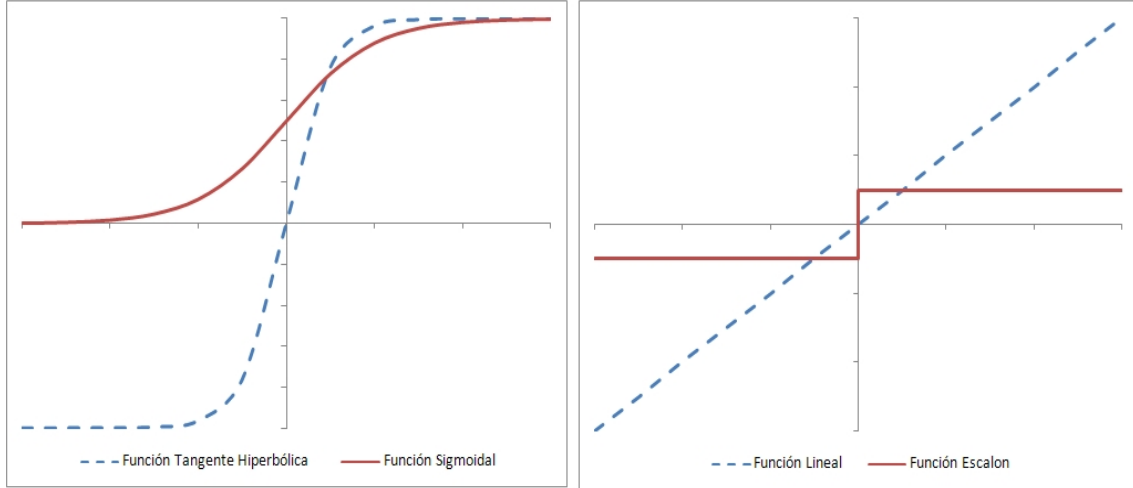


Figura 2.9: Funciones de activación típicas

de la suma de las señales de entrada.

$$y(t+1) = \begin{cases} 0 & \text{si } Net_i \leq b+u_i \\ Net_i - u_i & \text{si } b+u_i < Net_i < B+u_i \\ 1 & \text{si } Net_i \geq B \end{cases} \quad (2.9)$$

- Función continua o sigmoideal: Esta función toma las señales de entrada, las cuales pueden oscilar entre más y menos infinito, y restringe la salida a valores entre cero y uno, de acuerdo a la expresión (cf. Figura 2.9):

$$y(t+1) = \frac{1}{1 + e^{-(Net_i - u_i)}} \quad (2.10)$$

- Función tangente hiperbólica: Presenta un rango continuo de valores dentro del intervalo $[-1,1]$ y se ajusta a la siguiente expresión (cf. Figura 2.9):

$$y(t+1) = \frac{1 - e^{-(Net_i - u_i)}}{1 + e^{-(Net_i - u_i)}} \quad (2.11)$$

Para simplificar la expresión de la salida de una neurona i , es habitual considerar la existencia de una neurona ficticia, con valor de salida unidad, asociada a la entrada de cada neurona i mediante una conexión con peso de valor $-u_i$. De esta forma la expresión de salida quedará:

$$y_i(t+1) = \varphi(Net_i) \quad (2.12)$$

2.2.4. Clasificación de una Red de Neuronas Artificial

Las diferentes clasificaciones de RNA existentes se agrupan en base las conexiones que presentan y al tipo de aprendizaje que se utiliza en cada una de ellas [Hertz et al., 1991]. Si se clasifican en función de las conexiones, los tres grandes conjuntos son los siguientes:

- Redes *feedforward*: Estas redes presentan conexiones en un sólo sentido. Dentro de esta categoría destacan las Redes de Base Radial (RBF), el Perceptrón simple y el Perceptrón Multicapa.
- Redes recurrentes: Estas redes presentan conexiones en todas las direcciones.
- Redes parcialmente recurrentes: En este caso solo algunas de las conexiones de la red son recurrentes. Dentro de esta categoría destacan la red de Jordan y la red de Elman.

Por otro lado, en función del aprendizaje aparecen las siguientes clasificaciones:

- Redes supervisadas: Este tipo de redes utilizan un aprendizaje denominado aprendizaje supervisado. En este para cada patrón presentado a la red existe una respuesta deseada. La respuesta de la red se compara con su salida deseada y en base a esa comparación se ajustan los pesos de la red. Dentro de esta se engloban las redes *feedforward* y las redes recurrentes. En §2.3.3 se explicará en detalle su utilización para un PMC.
 - Una variante de este tipo de aprendizaje es el conocido como aprendizaje por refuerzo. En este caso la salida deseada exacta es desconocida, por lo que algoritmo de aprendizaje sólo se puede basar en si la información referente a la salida actual es o no correcta.
- Redes no supervisadas: Este tipo de redes utilizan un aprendizaje denominado aprendizaje no supervisado. En éste no se especifica a la red cual es la respuesta correcta, sino que a través de unas reglas de aprendizaje, la red descubre las relaciones presentes en los ejemplos. Dentro de esta se engloban los mapas de Kohonen y las redes ART (*Adaptive Resonance Theory*).

2.2.5. Proceso de aprendizaje

El cerebro biológico sobre el que se inspiran las RNA, aprende mediante la reorganización de las conexiones sinápticas entre las neuronas que lo componen. De igual forma, la inteligencia de las RNA reside en las conexiones existentes entre sus unidades de proceso o neuronas. Esto es debido a que las variaciones en el valor de los pesos producen cambios en el comportamiento de cada neurona y, en consecuencia, en el de toda la red. En estas estructuras artificiales, la reorganización se realiza mediante el ajuste de los pesos asociados a las conexiones establecidas entre las neuronas, en lo que

se denomina fase de aprendizaje. Este proceso necesita para comenzar un conjunto de datos de entrada, denominado conjunto de entrenamiento o conjunto de aprendizaje. Su objetivo es conseguir que la red aprenda de manera automática las propiedades intrínsecas de los patrones que se le facilitan, por lo que el conocimiento extraído, se incorpora a la RNA mediante un aprendizaje a partir de ejemplos.

Para que el entrenamiento sea capaz de extraer conocimiento de los ejemplos que se le presentan, es necesaria una cantidad significativa de los mismos. Conocer el número exacto de datos es una tarea dependiente del problema y de la arquitectura de red utilizada, pero si se dispone de pocos, el rango completo de relaciones que la RNA debería aprender no será cubierto. Por el contrario si se le presenta a la red una cantidad suficiente, ésta podrá reconocer todos los casos, tanto normales como anómalos, que encontrará durante la validación [Tarassenko, 1998]. En este sentido, los ejemplos deben ser representativos de todo el dominio del problema, conteniendo dentro de lo posible una cantidad equitativa de todas las clases a identificar. Gracias esto se evita que la red se especialice sobre un subconjunto de los datos y tienda a clasificar la nueva información sobre una clase o clases concretas.

De forma detallada, el entrenamiento de la red consiste en la introducción paulatina de todos los patrones del conjunto de entrenamiento. A través de un determinado esquema de aprendizaje, los pesos de las conexiones son modificados, comprobando al finalizar si se ha alcanzado un criterio de convergencia prefijado. De no ser así se repite el proceso, volviendo a insertar todos los ejemplos a la red, buscando una nueva actualización de los pesos. La modificación de los valores de los pesos puede realizarse después de la introducción de cada patrón de entrenamiento, o bien al finalizar la inserción de todo el conjunto (cf. §2.4.4.4)

En resumen, el entrenamiento de una red neuronal consiste en la presentación repetida de un número relativamente amplio de ejemplos diferenciados, hasta lograr una respuesta que se considere satisfactoria. Un punto esencial dentro de este proceso, es determinar cuándo detener el entrenamiento, ya que de no hacerlo se podría dilatar en el tiempo de forma indeterminada o incluso empeorar los resultados.

La capacidad de aprendizaje de una RNA se mide en función del conocido como error cuadrático medio de la función del error, criterio utilizado por el algoritmo de mínimos cuadrados (*Least Mean Square* o LMS) [Widrow and Hoff, 1959] para definir el error cometido por la red. Este error mide la diferencia entre los valores obtenidos por la red y los que deberían de obtenerse en realidad. Para ello la red trata de determinar los valores de los pesos asociados a las conexiones sinápticas que consiguen que las salidas de la red sean lo más parecidas a las salidas deseadas para el conjunto dado de los patrones de entrenamiento. Es decir, se trata de determinar los pesos sinápticos de manera que se minimice la función de error cuadrático.

Si se define el vector de las salidas reales, R^p , y las que proporciona la red, Y^p , para cada uno de los patrones, p , como:

$$R^p = (r_1, \dots, r_n) \quad p = 1, \dots, n \quad (2.13)$$

$$Y^p = (y_1, \dots, y_n) \quad p = 1, \dots, n \quad (2.14)$$

Y se aplican estos vectores sobre la fórmula del error cuadrático sobre un patrón determinado p , E^p , se obtiene:

$$E^p = \frac{1}{2} \sum_{j=1}^n (y_j^p - t_j^p)^2 = \frac{1}{2} \sum_{j=1}^n (e^p)^2 \quad p = 1, \dots, n \quad (2.15)$$

siendo $e^p = (y_j^p - t_j^p)$ el vector de los errores de la red para el patrón p y la neurona j . Por tanto a partir de la Ecuación 2.15, el error cuadrático total, E , para todos los patrones será:

$$E = \frac{1}{2} \sum_{p=1}^n \sum_{j=1}^n (e^p)^2 \quad (2.16)$$

2.2.5.1. Criterio de convergencia

Desde el punto de vista matemático un problema general de optimización no lineal sin restricciones se puede expresar como:

$$\min f(y) \quad y \in \mathfrak{R}^n \quad (2.17)$$

donde la función real $f : \mathfrak{R}^n \rightarrow \mathfrak{R}$ es la función objetivo que se desea minimizar sobre el espacio de búsqueda de \mathfrak{R}^n .

Si se tiene en cuenta el concepto de vector gradiente de una función f , la cual se supone derivable, se obtiene a partir de la primera derivada de la función objetivo:

$$\nabla f(y) = \left(\frac{\partial f(y)}{\partial y_1}, \dots, \frac{\partial f(y)}{\partial y_n} \right) \quad (2.18)$$

Este vector gradiente certifica las siguientes propiedades:

- $\nabla f(y) = 0$ es un punto estacionario de la función f . Puede tratarse de un máximo o mínimo, local o global, y un punto de silla.
- $\nabla f(y)$ apunta en la dirección en la que localmente, i.e. en un entorno de y , f crece más rápidamente.

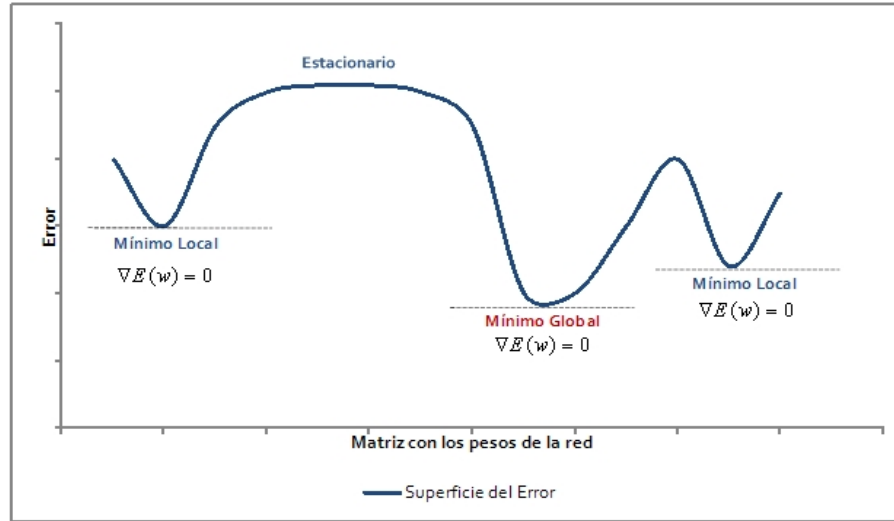


Figura 2.10: Representación de la superficie del error

Por otro lado, como se acaba de comentar, el conocimiento de una RNA recae sobre todas las conexiones de la arquitectura de red, y de forma más precisa sobre los pesos asociados a las mismas. Para conseguir encontrar el valor óptimo de los pesos que minimizan la función del error, el algoritmo RP estima dicho error, basado en mínimos cuadrados, entre el valor devuelto y el esperado (cf. Ecuación 2.16).

Si se aplica la Ecuación 2.18 al error cuadrático total definido en la Ecuación 2.16 para un vector de pesos W , se obtiene:

$$\nabla E(W) = \left(\frac{\partial E}{\partial W_1}, \dots, \frac{\partial E}{\partial W_n} \right) \quad (2.19)$$

Si E es constante de un ciclo a otro, se detiene el aprendizaje porque se considera que el entrenamiento de la red ha convergido y como consecuencia de esto los parámetros dejan de sufrir modificaciones. Por tanto, si se tiene en cuenta la derivada de la función de error E en ese punto para un vector de pesos determinado W , ésta será cero al localizar el valor mínimo del error:

$$\frac{\partial E}{\partial W} = 0 \Leftrightarrow \text{Min}E \quad (2.20)$$

En resumen, para un funcionamiento ideal, el proceso de aprendizaje de una RNA debería terminar cuando se detectará el mínimo global de la función que describe el error. Para ello el algoritmo recurre a un concepto geométrico muy útil, la derivada en un punto de dicha función. La derivada de una función en un punto es el valor que tiene la pendiente de la recta tangente en dicho punto. Dicha pendiente está dada por la tangente del ángulo que forma la recta tangente a la curva con el eje de las abscisas en ese punto.

Cuando el valor de la función del error en punto dado presenta una tangente horizontal, caso de valores máximos y mínimos, la derivada en dicho punto es cero, luego $\frac{\partial E}{\partial w} \approx 0$, siendo w los pesos de la red.

Lógicamente el algoritmo de aprendizaje intenta disminuir el error por lo que solo tiene en cuenta valores negativos de esta derivada. Para ello se mueve siempre en la dirección negativa del gradiente del error, lo que conlleva que los parámetros de la red sólo son modificados cuando esto sucede.

Sin embargo, el problema que puede encontrar el algoritmo, tal como se observa en la Figura 2.10, es detectar un mínimo local y tomarlo como global. Esto es debido a que el método detecta que cualquier pequeño cambio en los pesos, positivo o negativo, incrementa el error. Por tanto el algoritmo no es capaz de determinar en qué dirección deben moverse los pesos para que el error vuelva a decrecer dando por finalizado el proceso de aprendizaje.

2.2.5.2. Otros criterios de parada

Como se acaba de observar el proceso de aprendizaje puede presentar ciertos problemas en su rendimiento con la aparición de mínimos locales. No obstante este caso no es el único que puede presentar una complicación en el aprendizaje. Otro caso de especial relevancia para el proceso de aprendizaje y que limita la capacidad de convergencia de la RNA puede observarse en la Figura 2.11. En esta situación la función del error presenta un comportamiento ligeramente descendiente a lo largo del tiempo pero sin terminar de alcanzar el valor mínimo. El problema de esta situación es que se ralentiza el aprendizaje de forma indeterminada lo que afecta al rendimiento y eficiencia del sistema.

Para atajar estas situaciones, las técnicas convencionales más utilizadas, de forma individual o conjunta, para determinar cuándo detener el proceso intentando evitar situaciones no deseadas son las siguientes [Isasi and Galvan, 2004]:

- Detectar una modificación irrelevante de los pesos: Cuando los pesos de las conexiones no sufren variaciones durante el aprendizaje, se puede detener el entrenamiento ya que la red ha convergido. Este es el comportamiento estándar del algoritmo, sin embargo en §2.2.5.1 ha quedado claro el posible riesgo que se corre cuando se utiliza de forma aislada.
- Fijar un número de ciclos: Se decide a priori cuántas veces será introducido todo el conjunto y, una vez superado dicho número, se detiene el proceso y se da por aceptada la red resultante. Éste método es muy útil cuando la función de error presenta un comportamiento ligeramente descendiente a lo largo del tiempo pero sin terminar de alcanzar el valor mínimo 2.11. Si esto sucede el criterio anterior

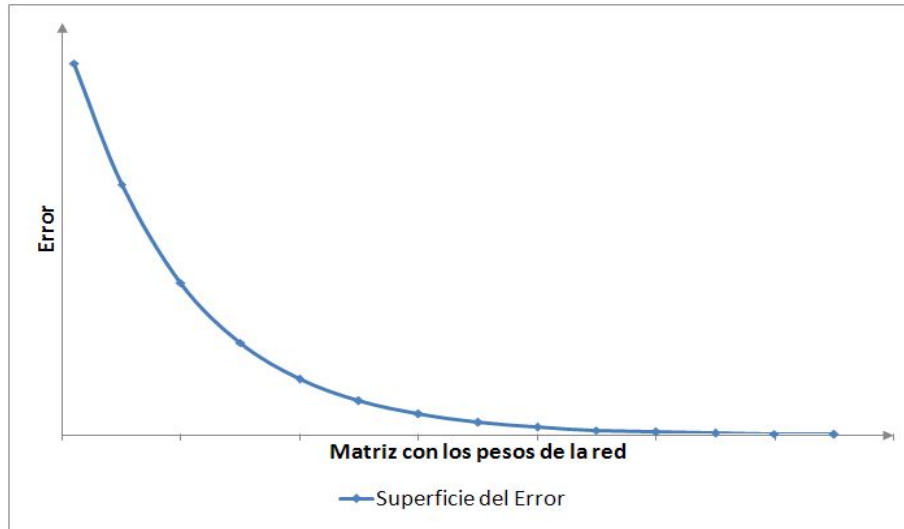


Figura 2.11: Evolución estacionaria del error

no es capaz de localizar el mínimo de la función y el proceso de aprendizaje se alargará de forma irremediable en el tiempo.

- Establecer un valor mínimo para el error: Cuando el error descienda por debajo de una cantidad preestablecida, se detiene el proceso de aprendizaje. Para este criterio puede suceder que la red jamás consiga bajar del nivel prefijado, en cuyo caso se debe disponer de un criterio adicional de parada, lo que significa que la red no ha convergido. En este caso, la red no ha sido capaz de obtener una solución y será necesario realizar pruebas adicionales cambiando algunos parámetros.

No existe, por tanto, un criterio de convergencia establecido, ya que su elección depende en gran medida de las características del problema a resolver y de la arquitectura de red. No obstante, es necesario detener el aprendizaje en algún punto, de no hacerlo el entrenamiento podría alargarse en el tiempo sin obtener resultados concluyentes si no apareciera ningún mínimo en la función del error.

En §2.4.4.3, se detallan otras técnicas alternativas encaminadas a reducir y mejorar el proceso de aprendizaje de una RNA. Su utilización en algunas situaciones es fundamental, debido a la gran cantidad de tiempo y resultados poco precisos que podría conllevar un entrenamiento estándar.

2.2.6. Ventajas e inconvenientes de las Redes de Neuronas Artificiales

En la actualidad existen más de cuarenta paradigmas distintos que utilizan las bases teóricas anteriores para formar el conjunto de soluciones de RNA [Moreno et al., 2002]. En función de la topología y características de las neuronas artificiales que lo forman, así como la manera en que se conectan entre ellas, se pueden citar una serie de ventajas comunes a todas ellas [Maren et al., 1990]:

- **Aprendizaje inductivo:** Las RNA tienen la habilidad de aprender a través de ejemplos, por lo que no necesitan ser programadas. Esto supone un enfoque radicalmente distinto al de los sistemas de computación tradicional, en los que cada paso que ejecuta el computador debe ser especificado con anterioridad por el programador, lo que absorbe tiempo y recursos humanos. En sistemas basados en RNA no se indican las reglas para dar la solución, sino que ellas mismas extraen sus propias reglas a partir de ejemplos de aprendizaje y modifican su comportamiento en función de su experiencia. Dichas reglas quedan almacenadas en las conexiones sinápticas.
- **Autoorganización:** Una RNA puede crear su propia representación de la información en su interior, descargando al usuario de esto.
- **Tolerancia a fallos:** Las RNA codifican su información de forma distribuida, i.e., el conocimiento acumulado por la red es compartido por varias de sus elementos de procesamiento. Este tipo de esquemas de almacenamiento distribuido presenta la ventaja de almacenar la información de forma redundante, lo que las convierte en un sistema tolerante a fallos. Una RNA podría seguir operando adecuadamente a pesar de sufrir lesiones con destrucción de neuronas o de sus conexiones.
- **Flexibilidad o tolerancia al ruido:** Una RNA puede manejar ciertos cambios en la información de entrada. Son capaces de extraer las características esenciales de las entradas aprendidas por lo que pueden procesar correctamente datos incompletos o distorsionados.
- **Procesamiento paralelo:** Su modo de trabajo es en paralelo, por lo que son especialmente adecuadas para el cálculo multiproceso.
- **Generalización:** Este término hace referencia a la capacidad de abstracción de las RNA para facilitar respuestas coherentes ante diferentes estímulos de entrada. Si una red ha sido convenientemente entrenada, es capaz de obtener resultados satisfactorios ante entradas distintas a las facilitadas durante su entrenamiento.
- **Operación en tiempo real:** Los computadores basados en arquitecturas neuronales, pueden ser diseñados para funcionamiento paralelo, utilizando hardware especial para obtener esa capacidad.

Pese a las ventajas que se acaban de exponer, las RNA también presentan una serie de inconvenientes, entre ellos destacan [Isasi and Galvan, 2004]:

- El tiempo de aprendizaje de la red, y por tanto de computación, no puede ser conocido a priori. El aprendizaje de la red se produce a partir de los patrones usados en su entrenamiento. Éstos son presentados a la red de forma iterativa modificándose los pesos de las conexiones en función del esquema de aprendizaje utilizado (cf. §2.3.3.2). Sin embargo no se puede determinar el número de veces que será necesario introducir todo el conjunto de datos para que la red aprenda. El tiempo necesario para completar el aprendizaje puede no presentar una relación directa con el tamaño del conjunto de patrones. Los conjuntos pequeños pueden requerir de muchas pasadas, en función de las características de los datos, lo que redundaría en un mayor tiempo de aprendizaje. Por el contrario, los conjuntos grandes pueden contener elementos no significativos y redundantes que hagan que el aprendizaje se produzca rápidamente.
- El diseño de una red para resolver un problema con éxito puede ser una tarea compleja y larga: El método común para diseñar una buena RNA suele ser mediante prueba y error. Aunque la comunidad investigadora ha intentado fijar reglas y heurísticas que permitan diseñar estructuras, los resultados demuestran que la creación de una RNA sigue siendo un proceso altamente dependiente del problema a resolver. En §2.4.3, se muestran los trabajos más relevantes en este sentido.
- Algunos de los algoritmos de aprendizaje utilizados por estos paradigmas, como el algoritmo RP del PMC (cf. §2.3.1), debido las características asociadas a su forma genérica presentan el inconveniente detenerse en un mínimo local u otro punto estacionario (cf. §2.3.3.1) al intentar minimizar la función de error. Esto puede provocar que no se alcance el mínimo global de la función de error y por tanto que la solución encontrada no sea la óptima.
- Las RNA son consideradas como modelos de caja negra ya que no se puede encontrar fácilmente una relación explícita entre las variables de entrada y las respuestas. El modelo neuronal produce una expresión altamente no lineal donde los pesos que se optimizan en el entrenamiento no tienen interpretación alguna, a diferencia de los coeficientes de los modelos estadístico que sí se pueden interpretar.
- No existe un procedimiento respaldado teóricamente para determinar el número de neuronas de la capa oculta ni las variables de entrada con mayor relevancia. Además no es sencillo realizar una inferencia estadística para medir la importancia de los pesos, aunque algunos autores han estado trabajando en esta posibilidad [Sarle, 1995].

2.3. Familia del Perceptrón

2.3.1. Perceptrón Multicapa

El PMC surge como intento de solucionar las limitaciones del Perceptrón simple mediante la introducción en la arquitectura de capas intermedias entre las de entrada y salida. De esta forma la intersección de los planos definidos por las capas ocultas proporcionaban soluciones a problemas de clasificación de gran envergadura. Los estudios realizados permitieron comprobar de forma experimental que el PMC resolvía estos problemas de una manera eficaz y relativamente simple.

No obstante, el proceso para llegar a esta arquitectura de RNA fue arduo, siendo necesaria la colaboración de numerosos investigadores durante un largo periodo de tiempo. En 1943, McCulloch y Pitts [McCulloch and Pitts, 1943] fueron los primeros que mostraron que su modelo de neurona, que consistía esencialmente en un dispositivo de umbral, podía representar cualquier función booleana.

Años más tarde, en 1987, Denker et al. [Denker et al., 1987] mostraron que toda función booleana podía ser representada por una red unidireccional multicapa de una sola capa oculta. Por esas mismas fechas, Lippmann [Lippmann, 1987] demostró que un Perceptrón con dos capas ocultas bastaba para representar regiones de decisión arbitrariamente complejas. Por otro lado, Lapedes y Farmer en 1988 [Lapedes and Farber, 1988] certificaron que un Perceptrón de dos capas ocultas es suficiente para representar cualquier función arbitraria, y no necesariamente booleana.

Posteriormente, Hecht-Nielsen en 1987 [Hecht-Nielsen, 1987] y 1990 [Hecht-Nielsen, 1990], aplicando el teorema de Kolmogorov [Kolmogorov, 1957], demostró que una arquitectura de características similares al PMC, con una única capa oculta, resultaba ser un aproximador universal de funciones. A finales de la década de los 80, surgieron diversos grupos que demostraron matemáticamente que el PMC de una única capa oculta era en efecto un aproximador universal de funciones, como por ejemplo Funahashi [Funahashi, 1989], Hornik et al. [Hornik et al., 1989] y Cybenko [Cybenko, 1989].

Por tanto, y aunque existen numerosos tipos de RNA, clasificadas en función de diversas características como la topología, el tipo de aprendizaje o la clase de entrada que reciben, las de tipo Perceptrón MultiCapa entrenadas con el algoritmo RP han sido las más utilizadas dentro de las aplicaciones de ingeniería [Ince, 2004]. Esto es debido a que las estructuras computacionales de este tipo, presentan una serie de cualidades que permite que se ajusten a los problemas típicos dentro de este dominio, las cuales pueden resumirse en:

1. Por una parte, a su capacidad como aproximador universal de funciones y clasificador de patrones. Las bases definidas por Lippmann en [Lippmann, 1987], han sido usadas

por diferentes autores para demostrar que el PMC es un aproximador universal sobre varios dominios, ya que cualquier función continua puede aproximarse con un PMC con al menos una capa oculta. Esto permite situar al PMC como una nueva clase de función capaz de aproximar e interpolar relaciones no lineales entre datos de entrada y de salida.

2. Por otra parte, es importante tener en cuenta la facilidad de su uso y aplicabilidad respecto de otras arquitecturas mucho más complejas. Así, el PMC se ha aplicado con éxito a la resolución de problemas en campos tan diversos como el reconocimiento del habla, el control de procesos, la conducción de vehículos, predicción de series temporales o los diagnósticos médicos.

No obstante algunos investigadores han recurrido a otras arquitecturas como la de Hopfield [Omar et al., 1998, 2000, Highsmith and Keshav, 1997, Waszczyszyn and Ziemianski, 2001], RNA probabilística [Jiang et al., 2006] o redes de Kohonen [Rafiq et al., 2001] para resolver problemas dentro del ámbito de la ingeniería estructural y de materiales.

En §2.5 se detallarán las investigaciones encontradas, donde se han utilizado las RNA, y más concretamente la arquitectura PMC, para la resolución de problemas relacionados con el dominio de esta tesis doctoral.

2.3.2. Arquitectura del Perceptrón Multicapa

Se denomina topología o arquitectura de una RNA, el modo en que se conectan los diferentes elementos de la misma. En este sentido, un PMC presenta una topología de tipo *feedforward* basada en el Perceptrón propuesto por Roseblatt en 1958 [Roseblatt, 1958], uno de los primeros modelos desarrollados. Una red es *feedforward* si existe un método que numere todos los nodos en la red de forma tal que no exista ninguna conexión desde un nodo con número grande hacia un nodo con un número más pequeño. Todas las conexiones fluyen desde nodos con números pequeños hacia nodos con números más grandes, i.e. la salida se propaga en cada nodo desde las entradas hacia las salidas, por lo que este tipo de red también es conocido como con conexiones hacia delante. La otra denominación posible en este sentido, *feedback* o red recurrente, se utiliza cuando el criterio anterior no se cumple, i.e. la red tiene conexiones hacia atrás.

El Perceptrón fue diseñado para, a partir de un conjunto de ejemplos de clases diferentes, determinar las ecuaciones de las superficies que hacían de frontera entre dichas clases. Su función principal era por tanto, realizar tareas de clasificación de forma automática. Para realizar esto el Perceptrón, dados unos datos de entrada pertenecientes a clases distintas, debía determinar la ecuación del plano que dividía a dichos datos en las clases a las que pertenecían, tal y como se muestra en la Figura 2.12.

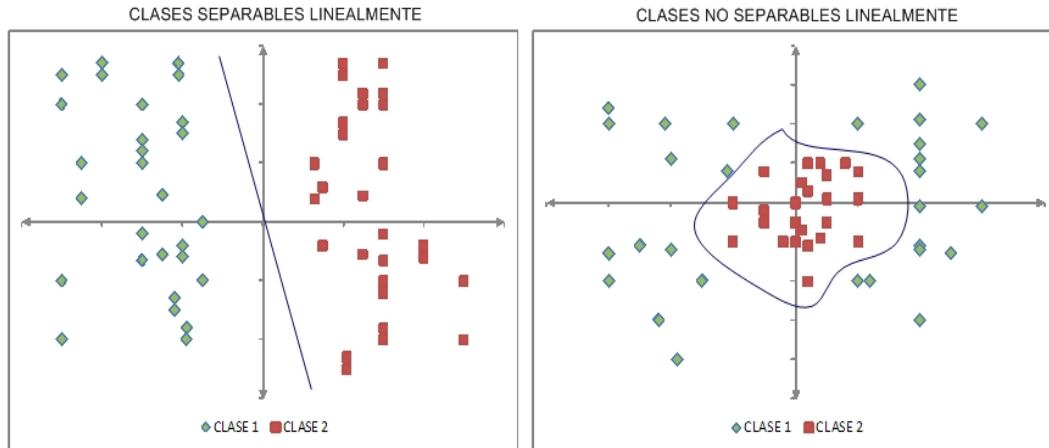


Figura 2.12: Clases separables linealmente y Clases no separables linealmente

El problema que presenta el Perceptrón es que sólo puede resolver problemas linealmente separables, i.e., problemas en los que se pueda trazar una única línea que separe las dos clases que éste permite identificar. Esta limitación viene dada por la arquitectura del Perceptrón, ya que consta sólo de una capa de entrada y otra de salida (arquitectura monocapa), lo que merma bastante su capacidad de representación.

Tras diversos estudios, Minsky y Papert demostraron en 1969 [Minsky and Papert, 1969] que la combinación de varios Perceptrones, esto es, la inclusión de capas ocultas, podía resultar una solución adecuada para tratar problemas no lineales (cf. Figura 2.12). Sin embargo la regla de aprendizaje del Perceptrón simple no podía aplicarse a este escenario, lo que conllevó su estancamiento durante varias décadas.

El PMC ha aprovechado estas conclusiones, utilizando una estructura multicapa con al menos una capa oculta. Además el mecanismo de aprendizaje de la red se completa con el algoritmo de entrenamiento RP, cuyo objetivo es que la red sea finalmente capaz de predecir adecuadamente el patrón de salida. El aprendizaje se realiza mediante la retropropagación de los errores devueltos por la red y el posterior ajuste de los pesos de interconexión entre las diferentes capas. Este algoritmo fue diseñado por Rumelhart, Hinton y Williams en 1986 [Rumelhart et al., 1986].

Otro punto de interés, es que con este modelo de RNA se define una relación entre las variables de entrada y las variables de salida. Esta correspondencia se obtiene propagando hacia adelante los valores de las variables de entrada. Para ello, cada neurona de la red procesa la información recibida por sus entradas y produce una respuesta o activación. Esta se propaga a su vez, a través de las conexiones correspondientes, hacia las neuronas de la siguiente capa (cf. §2.3.3).

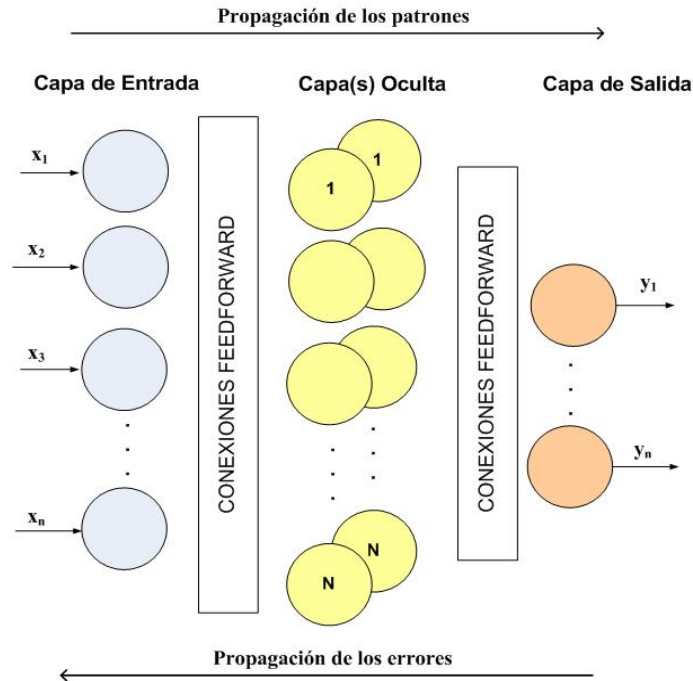


Figura 2.13: Arquitectura de un PMC

Con respecto a la arquitectura del PMC, ésta se caracteriza porque tiene sus neuronas agrupadas en capas de diferentes niveles. Cada una de las capas está formada por un conjunto de neuronas, distinguiéndose tres tipos de capas diferentes (cf. Figura 2.13):

- **Capa de entrada:** En esta capa se sitúan las neuronas que reciben los patrones de entrada en la red. No realizan ningún procesamiento sobre la información, tan sólo recogen los patrones del exterior y los propagan a todas las neuronas de la siguiente capa.
- **Capa(s) oculta(s):** En este caso, pueden existir una o varias capas, donde las neuronas reciben las entradas de las capas anteriores y transfieren sus salidas a neuronas de capas posteriores; realizando un procesamiento no lineal de los patrones recibidos.
- **Capa de salida:** Capa compuesta por las neuronas que proporcionan los valores de salida al entorno.

Las conexiones existentes entre estas capas siempre están dirigidas hacia delante, i.e., están conectadas a todas las neuronas de la siguiente capa, formando así una red totalmente conectada.

2.3.2.1. Función de activación

Durante el proceso de aprendizaje, el cómputo del gradiente local (∇) para cada neurona del PMC requiere el conocimiento de la derivada de la función de activación asociada con esa neurona ($\partial f(y)$). Para que esta derivada exista, se requiere que la función sea continua. En realidad, la diferenciabilidad es el único requerimiento que tiene que satisfacer la función de activación. Un ejemplo de función continua diferenciable y no lineal comúnmente usada en los PMC es la no-linealidad sigmoideal (cf. §2.2.3.1 para conocer las funciones de activación más usadas en RNA). Desde el punto de vista matemático, la utilidad de estas funciones de activación es que ella y sus derivadas son continuas. Por ello las funciones continuas son normalmente las elegidas al tener un mejor comportamiento.

Por último destacar que la ventaja de seleccionar una función de activación sobre otra no está todavía demostrada matemáticamente [Hassoun, 1995].

2.3.3. Aprendizaje del Perceptrón Multicapa

La capacidad de aprendizaje de una RNA reside en el proceso de entrenamiento que se realiza sobre la misma. El objetivo de este entrenamiento es la construcción de un modelo estadístico que represente el proceso que genera los datos, obteniendo buenas predicciones para las nuevas entradas que se le proporcionen a la red y evitando que ésta se especialice en los conjuntos de ejemplos proporcionados. A este problema se le conoce como pérdida de generalidad y para evitar que se produzca, la arquitectura de la RNA y el proceso de entrenamiento de ésta, deben ser diseñados de forma que el aprendizaje sea robusto y eficiente.

Para lograr esto, el aprendizaje en una red neuronal se lleva a cabo en la mayoría de escenarios utilizando un conjunto de ejemplos o patrones que describen el dominio del problema a resolver. En función de cómo se utilicen estos patrones durante el aprendizaje, se pueden encontrar tres clasificaciones diferentes: supervisado, no supervisado y por refuerzo (cf. §2.2.4). Por otro lado la inteligencia de una RNA recae en gran medida sobre las conexiones que unen las unidades de proceso o neuronas. La regla o algoritmo de aprendizaje es el mecanismo mediante el cual se van adaptando y modificando los parámetros asociados a estas conexiones, consiguiendo de esta forma que la RNA aprenda a partir de la información que se la facilita.

El PMC utiliza como regla de aprendizaje el método de retropropagación de errores (RP) o backpropagation (BP), presentado por Rumelhart et al. en 1986 [Rumelhart et al., 1986]. Este algoritmo surge como una generalización de la *regla delta* propuesta por Widrow en 1960 [Widrow, 1960], y su aparición ha ampliado de forma considerable el rango de aplicación de las RNA. La regla delta o regla de Widrow-Hoff, se basa en la

técnica de optimización matemática de mínimos cuadrados y por tanto en el algoritmo LMS. Su objetivo es determinar los pesos de las conexiones sinápticas entre las unidades de proceso de manera que las salidas de la red coincidan con las salidas deseadas, o por lo menos, sean lo más próximas posibles. Es decir, se trata de determinar los pesos de manera que el error total sea mínimo (cf. §2.2.5).

Desde el punto de vista teórico, el punto de partida de la regla delta es el *método del descenso del gradiente* de la función de error definido en la Ecuación 2.19. A partir de esta ecuación la regla delta para un peso W_j del vector de pesos W , se define como:

$$\nabla W_j = -\eta \frac{\partial E}{\partial W_j} \quad j = 1, \dots, n \quad (2.21)$$

donde η es el parámetro de la tasa de aprendizaje del algoritmo RP. Su inclusión permite controlar el ritmo de actualización de los pesos asociados a las conexiones de las neuronas. El uso del signo menos en la Ecuación 2.21 se coloca por ser un gradiente negativo en el espacio de pesos. Es decir se busca una dirección para el cambio de pesos que reduzca el valor del error. Por último la retropropagación se realiza mediante el cálculo del vector gradiente en la dirección opuesta al flujo de la salida en cada nodo.

No obstante, tal y como se ha explicado en §2.2.5.1, éste método se acerca al mínimo de la función de error cuadrático mediante el descenso por el gradiente. El problema es que en algunas situaciones, es posible que se produzca cierto zigzagueo en el descenso por el gradiente que hace que el algoritmo sea lento y por tanto poco eficiente. Para intentar evitar que esto suceda, se modifica la regla de RP teniendo en cuenta el descenso seguido en el paso anterior y descendiendo por una dirección intermedia entre la dirección marcada por el gradiente en sentido opuesto y la dirección seguida en el paso anterior.

Para ello se incluye en el algoritmo RP un término denominado constante *momentum* (μ) con valores $0 \leq \mu < 1$. Gracias a esto se controla el grado de modificación de los pesos teniendo en cuenta el cambio realizado en la etapa anterior. Cuando $\mu = 0$ se vuelve a tener la regla delta por lo que se conoce como regla delta generalizada. La inclusión de este término en el algoritmo RP tiende a acelerar el descenso en las direcciones similares de descenso. Por otro lado si tenemos direcciones con oscilaciones de signo en iteraciones consecutivas se ajustaran los pesos en cantidades pequeñas, i.e., tendrá un efecto estabilizador.

2.3.3.1. Aplicación del algoritmo de Retropropagación

El algoritmo RP utilizado por el PMC es de tipo supervisado por corrección de errores, utilizando para ello una función de error en la salida durante la fase de entrenamiento. A partir de ésta irá corrigiendo los pesos, primero los de la capa de salida, y luego progresivamente hacia atrás, los de las demás capas, de ahí su nombre. La función de

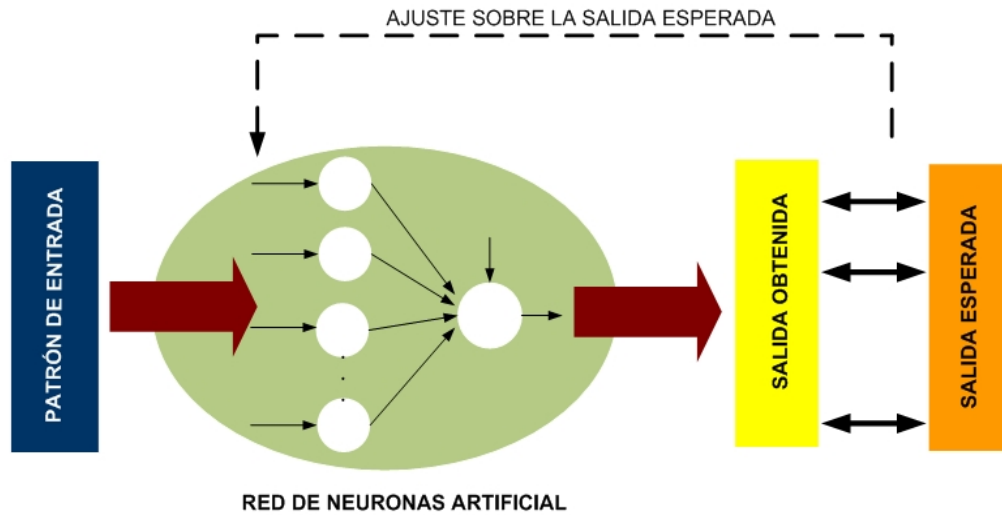


Figura 2.14: Aprendizaje supervisado en un PMC

error es similar a la del problema de mínimos cuadrados expresada en §2.2.5, pero además exige que las funciones de transferencia de las neuronas sean diferenciables (cf. §2.3.2.1).

Al tratarse de un algoritmo de aprendizaje supervisado para cada patrón presentado a la red es necesario que exista una respuesta. La respuesta de la red se compara con su salida deseada y en base a esa comparación se ajustan los pesos de la red. Por tanto la modificación de los parámetros se realiza para conseguir que la salida de la red sea lo más próxima posible a la salida deseada (cf. Figura 2.14, adaptada de [Isasi and Galvan, 2004]) y para cada patrón de entrada a la red, es necesario disponer de un patrón con la salida deseada. Este tipo de entrenamiento se formula a menudo como la minimización de una función de error tal como el error cuadrático medio total entre la salida actual y la salida deseada sumada sobre todos los datos disponibles.

El criterio usado para asegurar la convergencia durante el aprendizaje de un PMC con algoritmo RP es la aproximación por corrección de error. Con esta técnica, se persigue que la salida obtenida por la red sea lo más parecida posible a la salida deseada, minimizando de esta forma los errores obtenidos. Al comienzo del aprendizaje, el PMC recibe un conjunto de patrones de entrada y sobre ellos realiza el entrenamiento. En función de las salidas obtenidas en cada ciclo de aprendizaje, los pesos de las conexiones son ajustados mediante un tamaño de paso (*step-size*) de forma que, al final del proceso, se obtengan las salidas esperadas para cada uno de esos patrones de entrada.

De forma resumida el algoritmo RP consta de dos etapas o pasadas a través de las diferentes capas de la red:

- Etapa hacia delante. Se fijan los parámetros de la red, pesos de las conexiones, y se presenta una señal de entrada a la red, la cual se propaga hacia delante capa a capa para finalmente producir la salida.
- Etapa hacia atrás. El error entre la salida deseada y la producida por la red se propaga hacia atrás y los pesos de la red se ajustan de acuerdo con la regla de corrección del error. De forma específica, la respuesta real de la red se resta de la respuesta deseada para producir una señal de error. Esta señal de error se propaga a continuación hacia atrás a través de la red en contra de la dirección de las conexiones, de aquí el nombre error de retropropagación. El objetivo buscado al ajustar los pesos de las conexiones es conseguir que la respuesta real de la red se mueva más cerca de la respuesta deseada en un sentido estadístico.

La función de error sobre la salida obtenida, permite la corrección de los pesos asociados a las conexiones, comenzando con la capa de salida y continuando hacia atrás, con el resto de capas de la arquitectura de red. El objetivo es encontrar un valor mínimo de error, local o global en función de cuando se detenga, mediante la aplicación de pasos descendientes (gradiente descendiente). Cada punto de la superficie de la función de error se corresponde con un conjunto de valores de los pesos de la red. Con el gradiente descendiente, siempre que se realiza un cambio en todos los pesos de la red, se asegura el descenso por la superficie del error hasta encontrar el valle más cercano, lo que puede hacer que el proceso de aprendizaje se detenga en un mínimo local del error.

Por tanto, uno de los problemas asociados a este algoritmo de entrenamiento es que al intentar minimizar la función de error, encuentre un mínimo local o se detenga en algún punto estacionario (cf. Figura 2.10). Esto puede provocar que no se alcance el mínimo global de la función de error. No obstante, es necesario tener en cuenta que la obtención del mínimo global no siempre es obligatoria, ya que en algunos problemas puede ser suficiente con encontrar un valor mínimo preestablecido para el error.

2.3.3.2. Modos de entrenamiento

Un aspecto fundamental en el funcionamiento del algoritmo RP, es determinar cuándo se tiene que evaluar el error entre la salida obtenida y la salida esperada. Para ello se presentan a la red y de forma repetida los patrones de entrenamiento. Una presentación del conjunto de entrenamiento completo durante el proceso de entrenamiento se denomina ciclo de entrenamiento o *epoch*. El proceso de aprendizaje se mantiene *epoch* a *epoch* hasta que los pesos de las conexiones y los umbrales de la red se estabilizan y el error cuadrático medio sobre el conjunto de entrenamiento completo converge a un valor mínimo.

En función del momento en el que se actualizan los pesos de las conexiones de la red, existen dos esquemas o modos de entrenamiento diferentes:

1. El primero denominado *aprendizaje en serie*, también recibe otras denominaciones como secuencial, *online* o por patrón, se evalúa el error cometido y se modifican los pesos de las conexiones después de la presentación a la red de cada patrón del conjunto de entrenamiento.
2. En el segundo denominado esquema de *aprendizaje por lotes*, también se le conoce como modo *batch* u *offline*, la actualización de los pesos sinápticos se retrasa hasta que han sido presentados todos los patrones, i.e., al finalizar cada ciclo o *epoch*.

En §2.4.4.4, se detallan las ventajas asociadas a la utilización de cada uno en diferentes situaciones.

2.3.3.3. Tasa de aprendizaje y constante *momentum*

El algoritmo RP utilizado dentro de un PMC proporciona una aproximación a la trayectoria del espacio de pesos mediante el uso del método del descenso del gradiente. Para facilitar la convergencia de este algoritmo, la tasa de aprendizaje (η) controla la longitud del paso que el algoritmo debe dar en la dirección opuesta del gradiente, y por tanto se utiliza para ajustar el ritmo de aprendizaje. Cuanto más pequeño es el valor fijado η , más pequeños serán los cambios a los pesos en la red en la siguiente iteración, siendo la trayectoria en el espacio de pesos más suave. Sin embargo, esta mejora se alcanza a costa de un aprendizaje más lento. Si por el contrario η tiene un valor demasiado grande, el aprendizaje se acelera pero podría conducir a la inestabilidad de las estimaciones de los pesos de las conexiones. Por tanto, este parámetro debe ser un valor pequeño (típicamente dentro del rango $0 \leq \eta < 1$) para evitar dar pasos demasiado largos, i.e. que lleve a soluciones peores a las que ya tenía la red. Esto es debido a que el método del gradiente solamente garantiza el decrecimiento de la función de error si se desplaza en la dirección opuesta del gradiente pero dentro de un entorno suficientemente pequeño. Teóricamente, todas las neuronas deberían aprender a la misma velocidad, aunque es habitual que las últimas capas de la red tengan un gradiente mayor que las primeras. Por tanto y como contrapartida es común que el valor de η asignado a las capas del final sea menor. Además las neuronas con muchas entradas deberían tener un valor de η más pequeño que las neuronas con pocas entradas para mantener un ritmo de aprendizaje similar en todas ellas Haykin [1994].

Por otro lado, un método sencillo de incrementar la tasa de aprendizaje y al mismo tiempo evitar el riesgo de inestabilidad es modificar el algoritmo RP incluyendo el término *momentum* (μ) a modo de constante [Rumelhart et al., 1986] (cf. §2.3.3). Esta constante debe estar restringida al rango $0 \leq \mu < 1$, cuando μ es cero, el algoritmo RP opera sin *momentum*. La constante de *momentum* puede ser positiva o negativa, aunque es improbable que se utilice en la práctica un μ negativo.

La incorporación del *momentum* al algoritmo hace que la modificación de los valores

de los pesos sea menor durante su actualización, pudiendo tener efectos beneficiosos en el comportamiento del algoritmo [Watrous, 1987, Jacobs, 1988]:

- Cuando la actualización de los pesos de las conexiones se está realizando con valores altos, la inclusión del *momentum* en el algoritmo RP tiende a acelerar el descenso, sobre la superficie de error, en las direcciones cuesta abajo.
- Si la actualización de los pesos de las conexiones se está realizando con valores bajos, al introducirse la constante *momentum* en el algoritmo RP se produce un efecto estabilizador en las direcciones que oscilan en signo sobre la superficie del error.

La constante *momentum* también puede ser beneficiosa para prevenir que el proceso de aprendizaje termine en un mínimo local sobre la superficie de error.

Por último es destacable el hecho de que en la aplicación del algoritmo RP se puede elegir que todos los pesos en la red sean ajustables, o se puede restringir cualquier número de pesos para que permanezcan fijos durante el proceso de adaptación. En el último caso, las señales de error se propagan hacia atrás a través de la red de la manera usual, sin embargo, los pesos fijos se dejan inalterados. Esto se puede lograr simplemente haciendo que la tasa de aprendizaje η para un peso concreto W_{ij} sea igual a cero.

En §2.4.4.1 se describen las técnicas más utilizadas para optimizar estos parámetros.

2.3.3.4. Criterios de parada

El algoritmo RP por ser un método de gradiente comparte las desventajas de este método: convergencia lenta (son métodos de primer orden) y posibilidad de estancarse en puntos de silla o mínimos locales. Además en general, no puede demostrarse que el algoritmo RP converja y no hay un criterio bien definido para detener su funcionamiento.

Basándose en la teoría intrínseca del algoritmo RP, existen una serie de criterios para detener el aprendizaje y finalizar el ajuste de los pesos. Estos criterios se fundamentan en las propiedades asociadas a los mínimos locales o globales que aparecen en la superficie de error durante el entrenamiento de la red. El primero de estos criterios de convergencia fue especificado por Kramer y Sangiovanni en el año 1989 [Kramer and Sangiovanni-Vincentelli, 1989], donde se exponía que el algoritmo RP alcanza la convergencia cuando la norma euclidiana del vector gradiente alcanza un valor lo suficientemente pequeño. La desventaja de este criterio de convergencia es que, para experimentos que finalizan de forma exitosa, los tiempos de aprendizaje pueden ser grandes.

Como consecuencia de esto, surgió otro criterio de convergencia que solucionaba en parte estos problemas. En este segundo criterio, se considera que el algoritmo RP ha

convergió cuando la tasa absoluta de cambio en el error cuadrático medio por *epoch* es suficientemente pequeña. De forma general, este valor es considerado suficientemente pequeño si cae en el rango de 0.1 a 1 por ciento por *epoch*. Sin embargo a veces se usa un valor incluso menor, como 0.01, en función de las características del problema a resolver. No obstante, este criterio presenta el problema contrario al anterior ya que puede conllevar una finalización prematura del proceso de aprendizaje.

Por último, existe otro criterio basado en la capacidad de generalización del algoritmo. Después de cada iteración de aprendizaje, se evalúa dicha capacidad de generalización de la red en función del error obtenido al realizar el ajuste de la salida obtenida con la salida real. Cuando el rendimiento de generalización se considera adecuado, o cuando aparentemente ha alcanzado un pico el proceso de aprendizaje se detiene.

En §2.4.4.3 se detallan en profundidad estos criterios y se explica cómo realizar ajustes para obtener el rendimiento más eficiente posible en problemas de ingeniería.

2.3.4. Conjuntos de entrenamiento y validación

El siguiente paso una vez diseñada la estructura de la RNA y fijado el algoritmo de aprendizaje, consiste en definir correctamente el conjunto de datos usado para el entrenamiento y la validación. Durante el entrenamiento la red debe ser capaz de extraer las relaciones y el conocimiento necesario acerca del dominio del problema para poder recrearlo posteriormente. Por su parte durante la validación se pone a prueba el aprendizaje de la red comprobando su rendimiento contra un conjunto de ejemplos que no conoce.

En el caso de un PMC entrenado con algoritmo RP, Hecht-Nielsen [Hecht-Nielsen, 1990] distingue los siguientes conjuntos creados a partir de los ejemplos disponibles:

- Conjunto de entrenamiento (*train*): Se utiliza para entrenar la red, i.e. durante el aprendizaje se modifican sus pesos iterativamente hasta cumplir un criterio de parada establecido.
- Conjunto de validación o prueba del entrenamiento (CV): Se utiliza de forma opcional en algunas situaciones para indicar a la red cuándo detener el entrenamiento y controlar de esta forma su capacidad de generalización. De este modo se evitan situaciones anómalas como el entrenamiento excesivo, si este se prolonga demasiado, o el entrenamiento insuficiente, si se detiene antes de tiempo. Aunque se utiliza durante el proceso de entrenamiento no se modifican los pesos de las conexiones al calcular el error asociado a este conjunto.
- Conjunto de test (*test*): Se utiliza para validar la red una vez el entrenamiento ha concluido. Si la red no pasa la validación, deben rehacerse los conjuntos de entrenamiento y validación y volver a entrenar la red.

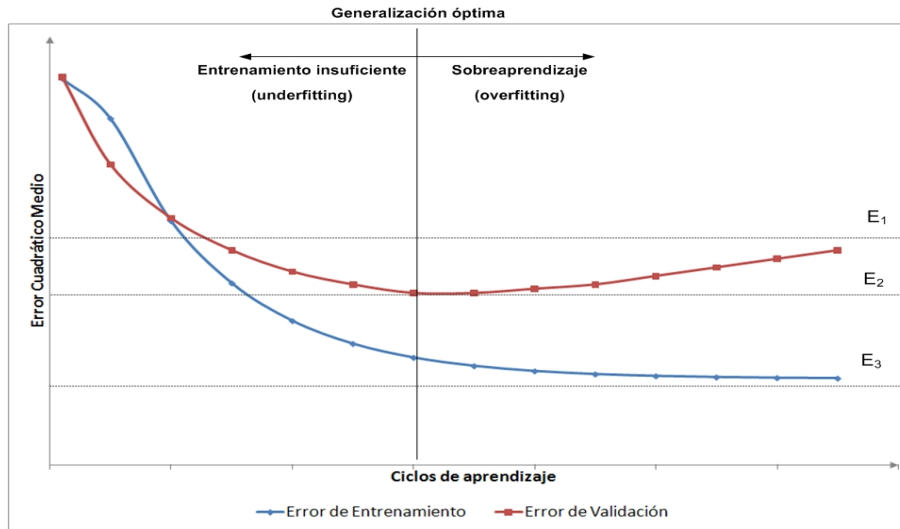


Figura 2.15: Evoluciones del error para los conjuntos de entrenamiento y validación durante el proceso de entrenamiento

- Conjunto de prueba de aceptación: Se utiliza en ciertos escenarios para que el usuario pueda comprobar el desempeño de la red. Este usuario aceptará la RNA solamente si ésta da buenos resultados sobre este conjunto. Lógicamente, el conjunto de test y el de aceptación deben ser distintos.

En algunos textos y aplicaciones informáticas los términos de conjuntos de validación y test se usan a veces de forma opuesta a la expuesta. Esto provoca confusiones innecesarias ya que el conjunto de test debería ser siempre aquel sobre el que se valida el rendimiento de una red ya entrenada respecto a su capacidad de generalizar.

Es crucial que los conjuntos test y prueba de aceptación se usen sólo como comprobación una vez el entrenamiento de la red ha finalizado, pero no durante el mismo. En caso contrario, la RNA aprenderá estos conjuntos específicos pero no será capaz de generalizar. En este contexto, generalizar equivale a realizar una interpolación matemática, i.e. si el nuevo vector está cerca o entre los patrones del conjunto de entrenamiento, la salida debe estar cerca de las salidas para esos patrones. Por otro lado, si el nuevo vector está lejos de cualquier patrón conocido, la salida de la RNA no será significativa.

En la Figura 2.15, se muestra que una vez fijados los conjuntos de entrenamiento y validación, existe un punto marcado como generalización óptima, con un valor asociado de la función de error igual a E_2 , en el que el comportamiento de la red es óptimo. En el caso de no alcanzarse dicho punto, por haberse detenido prematuramente el entrenamiento (zona de entrenamiento insuficiente o *underfitting*), la red dará un resultado pobre pues su ajuste será malo si se sobrepasa dicho punto. Por supuesto que el error en el conjunto de entrenamiento seguirá decreciendo, pero el del validación comenzará a crecer. Esto

se debe a que en la zona de entrenamiento excesivo conocida como *sobreaprendizaje* o *sobreentrenamiento* (*overfitting* u *overtraining*) la red memoriza demasiado los patrones del conjunto de entrenamiento y es incapaz de generalizar a nuevos patrones similares.

El conjunto de validación CV ayuda a evitar esa situación ya que no participa en el entrenamiento activamente (no causa modificaciones en los pesos), pero indica cuándo detenerlo. Dicho de otra manera, el entrenamiento excesivo hace que la RNA desarrolle una superficie en el espacio de pesos que ajusta bien los vectores del entrenamiento (el error en ellos es lo más pequeño posible), pero pierde su capacidad para interpolar bien entre dichos vectores, porque la superficie se curva demasiado entre esos puntos. Esto ocurre con un número de ciclos de entrenamiento infinito (o muy grande) ya que el error en el entrenamiento alcanza su valor límite en E_3 (su valor mínimo), mientras que para el conjunto de validación CV el valor E_1 (que no es su valor mínimo).

Empíricamente se ha observado que la Figura 2.15 es típica de la evolución de los errores en entrenamiento y validación. En la etapa intermedia que rodea al mínimo del error en el conjunto validación, la superficie ajusta aceptablemente los patrones del entrenamiento e interpola bien entre ellos, porque aún no se ha curvado en exceso. En cuanto a la elección de los conjuntos de entrenamiento y validación, Hecht-Nielsen indica que ambos suelen tomarse de tamaño aproximadamente igual, aunque esto no tiene por que seguirse de forma estricta, particularmente cuando el número de patrones disponible es escaso en este caso el conjunto CV debe reducirse e incluso anularse. En este último caso, el entrenamiento debe ser detenido cuando la curva de error en el conjunto de entrenamiento comience a estabilizarse y se tenga un error aceptable.

2.4. Técnicas para la optimización de los resultados en una RNA

2.4.1. Importancia de la optimización

Uno de los pilares sobre los que se sustenta la ingeniería, en cualquiera de sus ramas, es el denominado proceso de optimización ya que la función clásica de los ingenieros es diseñar nuevos sistemas, más eficientes y con menor coste, así como idear proyectos y procedimientos para mejorar las operaciones de los sistemas ya existentes [Reklaitis et al., 1983]. En términos generales la teoría de la optimización está basada en un fuerte componente matemático y el uso de métodos numéricos, y tiene como objetivo encontrar e identificar al mejor candidato entre una colección de alternativas sin necesidad de enumerar de forma explícita y evaluar todas las posibles alternativas.

Desde el punto de vista de las RNA, la optimización de los modelos neuronales pasa por seleccionar dentro de cada opción de configuración existente aquellas que permitan

mejorar la eficiencia del sistema global. Sin embargo la gran cantidad de variantes existentes y las fuertes relaciones que aparecen entre ellas hace que el ensamblado del modelo neuronal final sea un proceso delicado. Para ello es necesario tener en cuenta las posibles interacciones que pueden aparecer al modificar ciertos componentes del modelo neuronal para no afectar al rendimiento de la red.

Debido a la falta de información relativa a los mecanismos que permiten obtener una solución para un problema dado, las RNA han sido catalogadas como cajas negras por múltiples investigadores. No obstante, existe la posibilidad de parametrizar en gran medida su comportamiento y por tanto optimizar su rendimiento a través de una serie de factores y parámetros. Generalmente el cálculo de los valores óptimos se realiza mediante reglas empíricas de uso general basadas en el método de prueba y error donde se prueban diferentes configuraciones. El problema radica en la gran cantidad de parámetros a optimizar y las interacciones que aparecen entre ellos, lo que eleva el tiempo de computación necesario para encontrar la configuración errónea de modelo neuronal y por tanto complica la viabilidad de su utilización en ciertos escenarios.

Existen por tanto varios factores que afectan a esta capacidad de generalizar, los cuales, en función de sus características, pueden modificarse de forma previa al entrenamiento o bien durante el mismo. En la mayoría de casos no existe certeza del valor que debería tomar cada factor, tan sólo algunas reglas e indicaciones facilitadas por las investigaciones de autores en diferentes escenarios, las cuales generalmente son difíciles de extrapolar fuera de su dominio. Dichos factores se agrupan en tres categorías distintas: arquitectura de la red, algoritmo de entrenamiento y conjunto de datos disponibles. Es tarea del diseñador de la RNA, encontrar el diseño ideal de la arquitectura de red, ajustar el algoritmo de entrenamiento y seleccionar la información suficiente y reparto correcto de patrones para asegurar un aprendizaje adecuado.

La necesidad de la optimización de un modelo neuronal, y más concretamente de un PMC, se pone de manifiesto si se tiene en cuenta una de las características que lo convierten en una solución atractiva para resolver problemas ingenieriles: su función de aproximador universal. De forma teórica las investigaciones realizadas por Cybenko [Cybenko, 1989] y Hornik et al. [Hornik et al., 1989] han demostrado que un PMC con una única capa oculta es capaz de aproximar funciones continuas arbitrarias definidas sobre un hipercubo. Sin embargo el número de neuronas ocultas puede ser infinito y desconocido. El número de unidades ocultas por capa para una red neuronal del tipo *feedforward* condiciona la posibilidad de separar el espacio de patrones de entrada en diferentes regiones. Por otro lado la complejidad computacional es dependiente en cierta medida del número de éstas. Este número debe ser suficientemente grande para formar una región de decisión tan compleja como para resolver el problema propuesto. Por tanto, el uso de un número óptimo de neuronas ocultas por capa es perentorio si se desea un entrenamiento eficiente.

En cuanto al algoritmo de aprendizaje utilizado, como ha quedado patente anteriormente, en la mayoría de las situaciones se elige el algoritmo RP debido a sus cualidades para adaptarse a gran cantidad de problemas. No obstante, en algunos casos su rendimiento al intentar optimizar la función del error no es el esperado, ya que puede presentar una convergencia demasiado lenta o caer en mínimos locales. Este funcionamiento poco adecuado puede paliarse gracias a la existencia de ciertos parámetros de optimización y la aparición de variantes al algoritmo RP clásico.

Por otro lado, un PMC con algoritmo RP debería presentar un buen rendimiento sobre un conjunto finito de patrones de entrenamiento, pero no es sencillo determinar el tamaño óptimo de dicho conjunto. La importancia de este punto radica en que el rendimiento de una RNA, al tratarse de sistemas entrenados mediante ejemplos, reside en su habilidad de generalizar resultados fuera de los límites en los que se ha entrenado, técnica conocida como capacidad de generalización. Si el sistema solo memoriza los patrones de entrada, obtendrá unos buenos resultados durante la fase de entrenamiento, pero cuando se le presente una entrada distinta a éstas durante la validación, los resultados serán pobres.

En los siguientes apartados, se detallarán las técnicas más importantes publicadas por la comunidad científica, con el fin de mejorar la capacidad de generalización y por tanto optimizar, tanto en tiempo de cómputo como en precisión, los resultados obtenidos durante la fase de validación.

2.4.2. Tratamiento de los patrones

Al diseñar una RNA se está construyendo un modelo no-lineal que representa el fenómeno físico responsable de la generación de los ejemplos o patrones de entrada y salida usados para entrenar la red. En la medida que el diseño de la red es estadístico por naturaleza, se hace necesario tener un balance adecuado entre la fiabilidad de los datos de entrenamiento y la calidad del modelo diseñado.

En los siguientes apartados se detallan en profundidad los tres factores relativos a los patrones que mayor influencia presentan sobre el rendimiento de los modelos neuronales: número disponible, pre-procesado y reparto entre los conjuntos.

2.4.2.1. Cantidad de patrones necesarios

Uno de los apartados que más atención ha recibido por la comunidad investigadora ha sido el estudio de la relación existente entre la capacidad de generalización de la red, la cantidad de datos disponibles [Boonyanunta and Zeephongsekul, 2004, Crowther and Cox, 2006] y su distribución en los conjuntos de entrenamiento y test [Foody et al., 1995,

Crowther and Cox, 2005].

El número de patrones disponibles es el punto que mayor influencia presenta sobre la capacidad de los modelos neuronales. Si éste no es suficientemente grande, la red nunca conseguirá converger hasta el mínimo error global durante el entrenamiento. Como norma general se considera que para que no se produzca un sobreaprendizaje de los datos usados para entrenar, es necesario que éstos superen en cantidad al número de grados de libertad de la red (pesos de las conexiones). Por otro lado, si el conjunto usado para entrenar es demasiado largo, podría producirse una saturación y la red no alcanzaría el error mínimo global [Flood and Kartam, 1994, Jones et al., 1995]. Por tanto, la cantidad de datos necesarios para asegurar una buena capacidad de generalización depende de forma directa de las características del problema y del diseño de red escogido. Una arquitectura de red compleja necesitará más datos para conseguir completar el aprendizaje de forma correcta.

En algunos dominios la cantidad de información disponible es muy reducida, e.g. medicina, fenómenos meteorológicos o entornos nucleares, lo que limita la capacidad de obtención de patrones para realizar el entrenamiento y validación de la red. La regla general establece que es necesario al menos diez patrones de entrenamiento por cada peso de la red. Sin embargo algunos investigadores han creado reglas propias para aproximar su cálculo [Tarassenko, 1998], mientras que otros recurren a pruebas con diferentes configuraciones hasta alcanzar la ideal. En este sentido, varios autores han logrado a lo largo del tiempo, en el dominio ingenieril, resultados interesantes con diferentes arquitecturas de red y un conjunto de patrones reducido, inferior a 50 muestras [Akhavan et al., 2000, Bisagni et al., 2002, Ince, 2004, Liu et al., 2006]. Esto es de gran utilidad a la hora de enfocar el trabajo a realizar en esta tesis doctoral teniendo en cuenta la dificultad de la simulación numérica de los ensayos de impacto. No obstante, la estimación de la cantidad de datos es dependiente de varios factores como la arquitectura de red y el número de elementos libres. Por ello la extrapolación de conclusiones o la aplicación de las diversas heurísticas disponibles debe de hacerse teniendo en cuenta estas consideraciones.

Cuando el conjunto de datos es reducido, las técnicas de *boosting* y *bagging* y validación cruzada han demostrado su eficacia mejorando la capacidad predictiva de los clasificadores individuales [Opitz and Maclin, 1999, Crowther and Cox, 2005]. No obstante si el conjunto de datos para realizar el entrenamiento es extremadamente pequeño, los resultados del comité pueden no compensar respecto a los modelos individuales ya que cada uno de los modelos ve un conjunto de datos todavía más reducido. Por otro lado si el conjunto de datos es demasiado grande, el coste computacional es tan elevado que un comité de redes no es la solución más adecuada [Kotsianti and Kanellopoulos, 2007].

Por último, Boonyanunta y Zeephongsekul en 2004 [Boonyanunta and Zeephongsekul, 2004], definieron un modelo matemático para explicar la relación entre el tamaño del

conjunto de entrenamiento y la capacidad predictiva de la red. Como conclusión principal se establece que la capacidad de predicción de una RNA aumenta hasta un valor umbral. A partir de este punto no compensa la utilización de nuevos patrones, y por tanto su obtención, ya que la reducción del error es mínima y conlleva un mayor coste de computación durante el entrenamiento. De igual forma, según Bishop [Bishop, 1996] la reducción del número de datos disponibles para el entrenamiento funciona en ciertos problemas a modo de regularización, permitiendo de esta forma mejorar las tasas de generalización y evitar el sobreaprendizaje. Esta regularización ha sido utilizada por varios autores [Worden and Staszewski, 2000, LeClerc et al., 2007] obteniendo redes capaces de generalizar fuera de los límites marcados por un reducido conjunto de datos de entrenamiento.

2.4.2.2. Maximizar la información disponible

El método de validación cruzada (cross validation) fue introducido por Cover en 1969 [Cover, 1969] y aplicado con éxito en RNA por Morgan y Boulard [Morgan and Boulard, 1990] y Weigend et al. [Weigend et al., 1990] en 1990. Se trata de un método estadístico para dividir el conjunto de datos original en diferentes subconjuntos. Su aplicación es de utilidad cuando el número de datos es relativamente pequeño debido a la complejidad o incluso imposibilidad de su obtención. Existen varias alternativas de validación cruzada, entre las que destacan [Kohavi, 1995]:

1. *Hold Out Cross Validation* (HOCV): Esta variante no es considerada de forma estricta como técnica de validación cruzada porque los datos en realidad no son cruzados. Sin embargo, tomando los elementos al azar y repitiendo el procedimiento algunas veces (más de diez), la estimación del error total es aceptable. Una ventaja de esta técnica es que su aplicación no requiere un alto coste computacional. Se toma el conjunto completo de datos disponibles y se divide en dos subconjuntos: el primero de ellos aplicable a la fase de entrenamiento y el segundo para la fase de prueba. Generalmente, el reparto realizado consiste en dos terceras partes para el conjunto de entrenamiento y la otra tercera parte para el conjunto de prueba.

Cabe mencionar que, cuando se dispone de pocos datos o existe alta dispersión en los mismos, puede presentarse alta varianza en la estimación del error total. Por ello, es recomendable aplicar criterios estadísticos adicionales ya que el resultado depende fuertemente de las instancias que hayan sido seleccionadas para formar el conjunto de entrenamiento.

2. *K-Fold Cross Validation* (KFCV): Surge como alternativa para minimizar los efectos de la dispersión de los datos y las debilidades inherentes a la técnica anterior. En este método el conjunto de datos se divide en K particiones que dan lugar a K subconjuntos mutuamente excluyentes. Para cada una de las K estimaciones de error (E), uno de los K subconjuntos es usado como conjunto

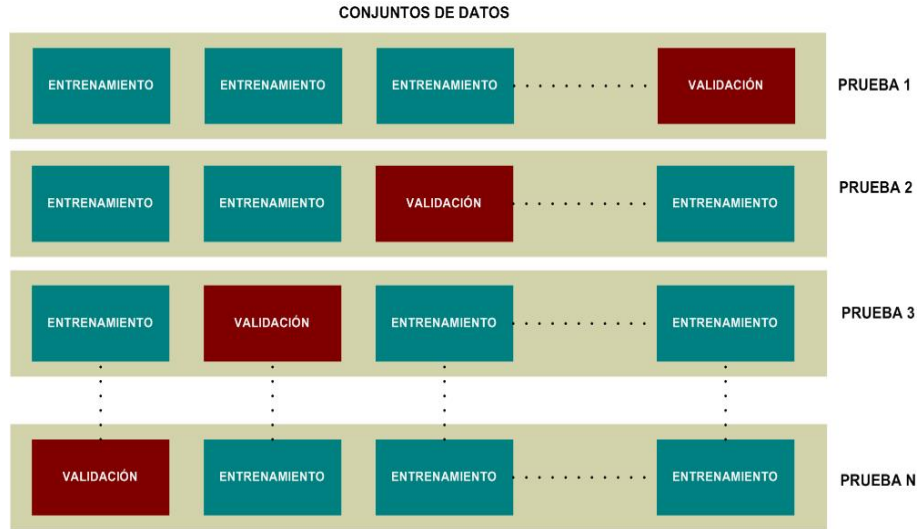


Figura 2.16: Estimación del error usando la alternativa KFCV

de prueba y los otros $K - 1$ restantes son agrupados para formar el conjunto de entrenamiento (cf. Figura 4.7).

El procedimiento es repetido K veces para asegurar que los K subconjuntos han sido utilizados en la fase de prueba, de modo tal, que se tienen K estimaciones de error E_i , que serán promediadas para obtener el error total de predicción E del modelo en cuestión.

$$E = \frac{1}{K} \sum_{i=1}^K E_i \quad (2.22)$$

Una de las ventajas que tiene este método es que, aún cuando se disponga de pocas instancias para ambas fases, entrenamiento y prueba. La estimación total del error E es fiable debido a que todos los patrones son considerados en la fase de prueba, i.e., todos los patrones se consideran una vez en la fase de prueba y $K - 1$ veces en la fase de entrenamiento. Destacar que la varianza en la estimación del error de predicción disminuye según el valor de K aumenta. La desventaja evidente es que la fase de entrenamiento tiene que ejecutarse en K ocasiones, lo que implica K veces más tiempo de computo que la técnica de HOCV.

3. *Leave-One-Out Cross Validation* (LOOCV): Este método de validación cruzada puede verse como la versión KFCV cuando K es igual al número N de patrones disponibles. Esto implica que, para cada una de las N veces que se realice la fase de aprendizaje, será necesario considerar $N - 1$ patrones para entrenamiento y solo uno para prueba. De este modo se obtienen N estimaciones parciales de error, el

cual se calcula usando la siguiente expresión:

$$E = \frac{1}{N} \sum_{i=1}^N E_i \quad (2.23)$$

Una de las ventajas asociadas a este método es que el problema comentado con anterioridad de la aparición de una alta varianza en la estimación del error total E , debido al reducido número de patrones disponibles o la dispersión en los datos, se vuelve casi despreciable. Esto es debido a que según crece el número de particiones, decrece el tamaño de cada una de éstas, ya que el número de patrones involucrados en la fase de aprendizaje es menor, lo que conlleva que la varianza tiende a disminuir.

2.4.2.3. Pre-procesado de la información

La forma en la que se presentan los datos a la RNA afecta a su capacidad de generalización, por lo que en algunas ocasiones es necesario realizar un procesado previo de los mismos antes de utilizarlos. La normalización de los datos se ha demostrado como un paso útil a la hora de afinar los resultados obtenidos por la red y reducir el tiempo de cálculo [Sola and Sevilla, 1997]. Esto permite en muchos casos agilizar el proceso de entrenamiento y su eficiencia ya que los datos que se le pasan a la red se encuentran próximos a la función de activación de cada neurona (cf. §2.4.3.4). Otros autores como Basheer y Hajmeer [Basheer and Hajmeer, 2000], sostienen que la normalización o escalado de los datos dentro de un rango uniforme de valores, e.g. [0-1] para la función sigmoideal, es esencial para prevenir que los valores grandes tengan más importancia que los pequeños y para evitar una saturación prematura de los nodos ocultos, lo que impediría el proceso de aprendizaje. Esto es especialmente importante cuando los datos de entrada toman valores muy grandes. No obstante, no existe un procedimiento estándar para normalizar estos valores de entrada y salida. Una forma es escalar las variables de entrada y salida (z_i) dentro un intervalo $[\lambda_1, \lambda_2]$ que se corresponda con el rango de valores de la función de transferencia:

$$x_i = \lambda_1 + (\lambda_2 - \lambda_1) \frac{z_i - z_i^{min}}{z_i^{max} - z_i^{min}} \quad (2.24)$$

donde x_i es el valor normalizado de z_i y z_i^{max} y z_i^{min} son los valores máximos y mínimos de z . Es recomendable que los datos sean normalizados con valores ligeramente cercanos a los límites de la función de transferencia, e.g. 0.1 y 0.9 en lugar de cero o uno para evitar la saturación de la función sigmoide, lo que llevaría a ralentizar el proceso o incluso a no conseguir un aprendizaje adecuado. En este sentido, Le Cun et al. [Le Cun et al., 1989] proponen que los valores de las variables independientes (variables objetivo o de salida) de la red deberían ser desplazados mediante un offset de valor ϵ , alejándolo de esta forma del valor límite de la función de activación. Dependiendo de si este es positivo o negativo,

se obtendría para un valor límite positivo, $+a$:

$$d_j = a - \epsilon \quad (2.25)$$

Mientras que para un valor negativo, $-a$:

$$d_j = a + \epsilon \quad (2.26)$$

Otras técnicas más complicadas fueron descritas por Masters en 1994 [Masters, 1994], sin embargo en esta misma investigación se indica que estas técnicas más complicadas podrían no producir resultados mejores que los que se consiguen con una normalización lineal. Para parámetros con valores excepcionalmente grandes, podría ser beneficioso tomar el logaritmo de los datos antes de realizar la normalización.

Por su parte Olden y Jackson [Olden and Jackson, 2002], afirman que antes de comenzar la optimización de una RNA, los datos debe ser transformados para que las variables dependientes e independientes manifiesten sus características particulares. La variable dependiente debe ser convertida dentro del rango $[0,1]$ para que de esta forma se ajuste a la función de transferencia usada (típicamente la función sigmoide) durante la construcción de la red de neuronas. Esto se consigue mediante la normalización *min-max* basada en la fórmula:

$$r_n = \frac{y_n - \min(y)}{\max(y) - \min(y)} \quad (2.27)$$

donde r_n es el valor transformado del ejemplar n , y_n es el valor original y $\min(y)$ y $\max(y)$ representan los valores máximos y mínimos de la variable respuesta Y . Es importante resaltar que las variables dependientes del tipo binario (SI/NO) no tienen que ser transformadas porque ya toman valores dentro de este rango. A su vez, las variables independientes deberían ser convertidas en valores *z-score* para eliminar las unidades de los datos, i.e. media=0 y desviación típica=1, usando la formula:

$$z_n = \frac{x_n - \bar{X}}{\sigma_x} \quad (2.28)$$

donde z_n es valor estandarizado y x_n el original para el ejemplar n , mientras que \bar{X} y σ_x son la media y la desviación típica de la variable X .

Dejando a un lado las técnicas de normalización, otras alternativas usadas son la introducción de ruido aleatorio para cada patrón de entrenamiento, la eliminación de datos atípicos o la reducción de la dimensión de cada variable [Stein, 1993, Swingler, 1996]. Con esto se consigue que la red no se ajuste de forma exacta a ellos, mejorando su

capacidad de generalización. Una revisión de los métodos más comunes de normalización y de introducción de ruido, así como su utilización en aplicaciones de ingeniería puede observarse en el trabajo de Rafiq et al. publicado en 2001 [Rafiq et al., 2001].

2.4.2.4. Distribución de patrones

La distribución de estos datos entre los diferentes conjuntos posibles, entrenamiento, validación y test, afecta también al rendimiento de la red. Un conjunto de test grande permite mejorar su capacidad de generalizar, pero a costa de penalizar el aprendizaje de la red debido al reducido tamaño del conjunto de entrenamiento. Existen varios estudios para intentar aclarar esta cuestión, Swingler propone un reparto del 20% para el conjunto de test [Swingler, 1996], mientras que Nelson por su parte considera que la cantidad debería estar entre el 20% y el 30% [Nelson and Illingworth, 1990]. Looney va más lejos y especifica que el reparto debería ajustarse a un 65% para entrenamiento, 25% para test y 15% para validación [Looney, 1993]. Crowther y Cox [2005] establecieron una heurística para estimar el reparto óptimo de datos. En su opinión el reparto aleatorio entre los diferentes conjuntos conduce a RNA que no se pueden considerar como óptimamente entrenadas y generalizadas. En concreto los repartos estándar 80/20 o 50/50 para entrenamiento y test no son los más apropiados en la mayoría de problemas.

No obstante e independientemente de la forma de reparto elegida, es necesario tener en cuenta que el conjunto de entrenamiento debe cubrir los límites superior e inferior de todos los valores usados, así como tener un número suficiente de ejemplos que representen las características particulares de todo el dominio.

Por último, en las tareas de clasificación de patrones usando el aprendizaje RP con entrenamiento secuencial, una técnica sencilla que se utiliza comúnmente es randomizar o barajar el orden en el cual se presentan los ejemplos al PMC de un epoch al siguiente. Idealmente, la randomización asegura que los ejemplos sucesivos en un epoch presentado a la red raramente pertenecen a la misma clase [Le Cun et al., 1989].

2.4.3. Arquitectura de la red

Uno de los aspectos más importantes para conseguir una buena capacidad de generalización es ajustar el tamaño de la red en función de las necesidades del problema a resolver. Este diseño es, todavía hoy, un proceso guiado por el conocimiento del experto sobre el problema. Este experto se basa en su experiencia y en un largo proceso de prueba y error para decidir cuál es la mejor arquitectura para un problema dado. El diseño de la arquitectura de una red de neuronas se puede ver como un problema de búsqueda en el espacio de las arquitecturas, donde cada punto del espacio de búsqueda es

una arquitectura determinada; por lo que se trata de una tarea larga y tediosa. Existen dos aproximaciones fundamentales para controlar la complejidad de un modelo de RNA [Sarle, 1995]: selección del modelo y regularización.

La técnica de selección del modelo requiere elegir el número de elementos o nodos ocultos así como sus conexiones. Dentro de éstas la técnica más habitual consiste en la poda (*pruning*) de la arquitectura de la red (cf. §2.4.3.2), en la que se comienza con un tamaño de red de gran envergadura y se van eliminando diferentes nodos o conexiones durante el entrenamiento siguiendo diversos algoritmos ad hoc. Su funcionamiento por tanto es similar a lo métodos de eliminación por pasos (*stepwise*) utilizados en regresión.

En muchos de los trabajos encontrados en la literatura, estos métodos de poda sólo se han utilizado para eliminar las entradas irrelevantes y reducir la complejidad de la arquitectura necesaria para ese problema. Para ello en primer lugar determinan las variables más significativas y posteriormente aquellas que se encuentren por debajo de un valor umbral son excluidas de la red. Esto permite reducir el tamaño y por tanto minimizar la redundancia de los datos de entrada. No obstante, en muchos dominios no es suficiente con obtener una buena predicción usando RNA, también es necesario conocer la contribución individual de cada variable para la resolución del problema así como las interacciones que aparecen entre ellas y la salida de la red [Gevrey et al., 2003]. El principal problema para obtener esta clase de información radica en las características de las RNA, ya que éstas se caracterizan por que su conocimiento y experiencia esta codificado de forma implícita a través del conjunto de pesos numéricos de las conexiones, lo que no facilita una interpretación directa de su influencia sobre la salida de la red. La aproximación estadística habitual para aproximar la selección del modelo se basa en la estimación del error de generalización cometido por cada uno, eligiéndose aquel con menor valor.

Por su parte la regularización incluye la penalización de la solución del problema de estimación para mejorar la generalización mediante el suavizado (*smoothing*) de las predicciones. Según las pautas dictas por Bishop [Bishop, 1996], los métodos clásicos de regularización son la atenuación de pesos (*weight decay*), la introducción de ruido en las entradas durante el entrenamiento (cf. §2.4.2) y el *método de parada temprana* (*early stopping*) (cf. §2.4.4.3).

Existen por otro lado varias formulaciones teóricas para calcular el tamaño idóneo que necesita una red para aproximar una función concreta [Baum and Haussler, 1989, Haykin, 1994, Lawrence et al., 1996]; y por otro han aparecido trabajos que se basan en las técnicas de computación evolutiva como elemento fundamental de la búsqueda de la arquitectura óptima [Gutierrez et al., 2001]. Por ello el desarrollo de métodos automáticos de determinación de arquitecturas resulta un campo de gran interés entre los investigadores en RNA. Estos métodos utilizan fundamentalmente técnicas de búsqueda, como algoritmos genéticos, enfriamiento simulado, estrategias evolutivas o

incluso técnicas multiagente. La mayoría de estos métodos se basan en la representación directa de los parámetros de la red, lo que hace que sean métodos no escalables, i.e., para representar arquitecturas moderadamente complejas se requieren codificaciones de gran tamaño, lo que afecta al tiempo de convergencia del algoritmo considerado [Guinea et al., 2002]. Debido a esto, en la práctica se recurre como técnica más habitual al método de prueba y error para conseguir diseñar la mejor arquitectura posible. Otra alternativa sencilla consiste en comenzar con un número pequeño de neuronas ocultas y bajo demanda, en función de las necesidades de precisión del modelo, aumentar el número de ellas. Para acelerar la localización del tamaño más adecuado se suele recurrir a la técnica de parada temprana.

Finalmente, algunas de las preguntas más frecuentes que aparecen en este punto son, ¿son mejor dos capas ocultas que una? o ¿las redes grandes generalizan mejor que las pequeñas?. La respuesta en la mayoría de casos se obtiene a partir del razonamiento expuesto por Occam: *“en igualdad de condiciones la solución más simple suele ser la correcta”* [Domingos, 1999]. No obstante, esto no siempre se cumple ya que en función del problema, se ha demostrado que a veces las redes grandes funcionan mejor que las pequeñas [Franco et al., 2005].

En los siguientes apartados, se detallan aquellos aspectos relacionados con la arquitectura de una RNA que influyen en el aprendizaje, como la determinación del número de capas y neuronas ocultas, la cantidad de variables, la elección de las funciones de activación de las neuronas o la modificación de los pesos durante el proceso de entrenamiento.

2.4.3.1. Capas y neuronas ocultas

¿Cuál es el tamaño que debe tener una RNA para realizar una determinada tarea? La respuesta a esta pregunta está directamente relacionada con la capacidad de la red neuronal y debería ser contestada independientemente del algoritmo de aprendizaje utilizado. El tamaño de la red depende de la cantidad de elementos ocultos que se incluyan, i.e. capas y neuronas ocultas, ya que estos pueden variar y no son dependientes de las entradas y salidas de la RNA. Estos elementos ocultos añaden la capacidad de realizar aproximaciones no lineales y discriminar la información en diferentes regiones. Para evitar un elevado coste computacional e incluso obtener resultados poco fiables es necesario localizar el número adecuado de los mismos. Por tanto, la pregunta clave es ¿cuántas neuronas ocultas se necesitan en un PMC para particionar el espacio de datos de entrada en las diferentes regiones a las que pertenecen?.

La primera respuesta satisfactoria la facilitó Lippman en 1987, ya que en sus investigaciones dedujo que un PMC con tres capas permite resolver la mayoría de ellos pudiendo formar cualquier límite complejo arbitrario de decisión [Lippmann, 1987]. Por

su parte Hornik et al. [Hornik et al., 1989] y Cybenko [Cybenko, 1989] establecieron dos años después que un PMC con al menos una capa oculta es capaz de aproximar cualquier función. Sin embargo el número exacto de capas y de nodos en cada una suele elegirse mediante prueba y error o utilizando diferentes metodologías para aproximar a priori el valor óptimo. Sin la capa oculta, la red es esencialmente equivalente a un ajuste de mínimos cuadrados de los datos, esto es debido a dicha capa oculta permite el modelado no lineal de las relaciones de entrada-salida. Además Goh sugirió que para modelar funciones continuas una única capa oculta es adecuada, mientras que para funciones discontinuas podría necesitarse al menos una segunda [Goh, 1994].

En resumen y como norma general una red con una sola capa oculta y un número adecuado de neuronas suele ser suficiente para resolver la mayoría de problemas. No obstante en algunos dominios y problemas específicos se ha demostrado que la introducción de una segunda capa oculta permite mejorar la capacidad de generalización [Haykin, 1994, Hippert et al., 2001]. En relación con esta afirmación existe una fuerte correspondencia entre los patrones de entrada y el número necesario de neuronas ocultas. Según los estudios de Kung y Hwang [Kung and Hwang, 1988] si los datos de entrada son totalmente irregulares serán necesarios al menos tantas neuronas ocultas como patrones.

En cuanto al número de neuronas ocultas, su correcta elección es un punto de especial importancia durante el diseño de la red, ya que los valores seleccionados afectarán drásticamente a su rendimiento. En el caso de incluir pocas neuronas, la red no será capaz de aprender correctamente y su respuesta ante datos nuevos no será la adecuada. Por otro lado si se incluyen demasiadas, además de complicarse el proceso de entrenamiento, la red devolverá valores falsos en ciertos puntos [Flood and Kartam, 1994]. Huang y Huang [Huang and Huang, 1991] detallaron una serie de reglas para conocer los límites superior e inferior del número de neuronas necesario para aproximar funciones no arbitrarias mediante un PMC con una sola capa oculta y algoritmo RP. Por otro lado, Kung y Hwang [Kung and Hwang, 1988] mostraron la relación existente entre el número de neuronas necesario, los patrones de entrada y la tasa de aprendizaje para un problema dado. Otros autores relacionan el número de neuronas ocultas con la cantidad de datos disponibles y las variables de entrada y salida utilizadas [Swingler, 1996, Jenkins, 1997]. Sin embargo estas reglas no pueden ser generalizadas ya que no se pueden escalar a todos los problemas a resolver. No obstante, en la práctica se suele utilizar como límite superior del número de neuronas ocultas, como mínimo el doble del número de neuronas de entrada, i.e. de variables de entrada. Sin embargo es necesario recalcar que esta afirmación tampoco garantiza la mejor capacidad de generalización de la red. Por último, cabe destacar que en la actualidad existen líneas de investigación abiertas basadas en técnicas evolutivas, para tratar de determinar de forma automática el número óptimo tanto de neuronas como de capas ocultas [Isasi and Galvan, 2004]. Estas técnicas se utilizan para realizar la búsqueda en el espacio de arquitecturas de redes guiadas por la optimización del rendimiento de la red.

En conclusión, es tarea del diseñador por tanto elegir el número adecuado de capas

y de neuronas ocultas, pero teniendo en cuenta que un número excesivamente grande o pequeño de las mismas hará que la red no converja bien. Debido a esto, muchos autores han desarrollado sencillas reglas empíricas para el cálculo inicial de estos valores [Carpenter and Barthelemy, 1993, Bishop, 1996, Tarassenko, 1998]. Sin embargo en muchas ocasiones se producen discrepancias sobre su utilización en diferentes dominios lo que conlleva que no sea sencillo decantarse por una u otra. Berke y Hajela [Berke and Hajela, 1992] sugieren que el número de nodos de la capa oculta debería ser un valor intermedio entre la media y la suma de las entradas y las salidas. Por su parte Swift et al. [Swift and Batill, 1991] utilizan un número superior de nodos al especificado en la regla anterior. Además el problema de estas heurísticas es que son aproximaciones del valor más adecuado, i.e., siempre es necesario hacer simulaciones adicionales para comprobar que los valores aproximados son en realidad los mejores.

2.4.3.2. Crecimiento y poda de la red

Una cuestión habitual a la hora de diseñar una RNA es decidir el tamaño de red necesario para resolver un problema particular. Si el entrenamiento se inicia con un tamaño de red demasiado pequeño para el problema en cuestión el aprendizaje no se realizará de forma satisfactoria. El diseñador deberá entonces comenzar un proceso tedioso para solventar esta situación, incrementando el tamaño de la red y volviendo a entrenar. Por el contrario, si la red es demasiado grande para las necesidades del problema el entrenamiento se volverá de forma irremediable más lento. Por tanto un aspecto crucial es conseguir minimizar el tamaño de la red, reduciendo así la complejidad pero manteniendo la precisión de la misma. En un primer momento parece coherente pensar en un número mínimo de nodos ocultos como paso inicial y luego ampliarlo con otro más grande, ya que las redes pequeñas son más rápidas y sencillas de diseñar. No obstante, éstas son más sensibles a las condiciones iniciales y a los parámetros de aprendizaje, y presentan una mayor probabilidad de caer en mínimos locales. En cambio, como norma general, una RNA con tamaño mínimo tiene menos probabilidades de aprender el ruido que puede ir asociados a los patrones sobre los que entrena, y puede así generalizar mejor sobre datos nuevos. Por tanto en función del dominio puede ser conveniente empezar con una red grande y podarla de forma paulatina, i.e. ir suprimiendo un número apropiado de nodos ocultos, hasta lograr un sistema más estable con una mejor generalización.

En resumen, las técnicas incluidas dentro de este apartado están orientadas a reducir la arquitectura de red de un modelo neuronal, consiguiendo de esta forma minimizar la complejidad y agilizar el proceso de aprendizaje. No obstante estas estrategias han servido como base para realizar investigaciones encaminadas a determinar la importancia de las variables de entrada del modelo neuronal y seleccionar las que más influyen por tanto sobre su salida y su capacidad de predicción. Para lograr obtener el tamaño de red óptimo los investigadores han optado de forma mayoritaria por las siguientes estrategias:

- Crecimiento de la red (*growing*): En este caso se comienza con una RNA pequeña,

ya que el tamaño de la red debe ser demasiado pequeño para no resolver el problema tratado, y luego se agregan o bien una nueva neurona o una nueva capa de neuronas ocultas. Este proceso se repite de forma iterativa hasta encontrar la solución ideal.

- Poda de la red (*pruning*): Se trata de la estrategia contraria ya que se comienza con una RNA de grandes proporciones, con mal comportamiento para el problema en cuestión. A partir de este punto se procede a podar eliminando ciertos pesos de forma selectiva y ordenada.

De forma resumida algunos de los procedimientos más utilizados por los algoritmos anteriores para realizar la poda de una red neuronal son los siguientes:

- Eliminar de la red aquellas conexiones cuyos pesos sinápticos sean de pequeña magnitud.
 - Eliminar aquellas conexiones cuya existencia no afecte significativamente a las salidas de la red. Para realizar este tipo de poda basta con ir comparando el error que devuelve la red cuando un peso sináptico es reducido a cero. Si se incrementa demasiado, entonces se vuelve al valor inicial. Éste método también se conoce como de fuerza bruta.
 - Eliminar aquellas neuronas de entrada que producen cambios insignificantes en la salida de la red. Este supone reducir la dimensionalidad de los patrones de entrada al detectar aquellas componentes de entrada que son innecesarias.
- La última opción consiste en una mezcla de las anteriores. En este caso se parte de una red de tamaño suficiente y se podan las conexiones y unidades de proceso que se consideran poco relevantes. A continuación se añaden nuevas unidades de proceso con pesos aleatorios y se vuelve a entrenar la red. Este proceso se continúa hasta que se consigue un tamaño aceptable y un comportamiento satisfactorio.

El objetivo de estas alternativas es conseguir la arquitectura más pequeña posible que consiga mantener buenas condiciones de generalización. Para ello se entrenan de forma sucesiva diferentes redes, con múltiples arquitecturas, hasta que las respuestas devueltas son las correctas. Como es lógico esta técnica consume grandes cantidades de tiempo, que varía en función del tamaño de la red. Para su utilización en sistemas grandes, aparecieron varias revisiones que conseguían paliar en parte este inconveniente. Reed publicó en 1993 una revisión de las alternativas más comunes usadas por la comunidad investigadora [Reed, 1993].

Dentro de estas estrategias, los algoritmos de poda han experimentado una gran actividad investigadora durante los últimos años. A partir de estos trabajos se puede realizar una clasificación de los métodos o algoritmos más utilizados para simplificar las arquitecturas de red mediante la eliminación de neuronas [Sietsma and Dow, 1988, Reed, 1993]. El primero se basa en una estimación de la sensibilidad de la función del error cuando se elimina algún elemento, para lograrlo aquellos elementos que producen el menor efecto pueden ser excluidos de la red. El segundo grupo de algoritmos, conocidos

como métodos de castigo, introduce parámetros adicionales a la función objetivo que permite que la red sea escogida como una solución eficaz. Un ejemplo consiste en introducir un término proporcional a la suma de todos los pesos, lo que favorece la detección de pesos pequeños, ya que aquellos que están cerca de cero es probable que no influyan en la salida y por lo tanto pueden ser eliminados.

En general, los métodos de sensibilidad modifican una red ya entrenada, i.e. la red es entrenada, se estiman los factores de sensibilidad y se eliminan los nodos y pesos necesarios. Por otro lado, los métodos de castigo modifican la función de coste para que el algoritmo RP reduzca a los pesos innecesarios hasta valores cero, eliminándolos de esta forma durante el entrenamiento. En la mayoría de los casos la optimización que utiliza la eliminación de elementos suele ser más efectiva que la introducción de parámetros adicionales, parámetros que son normalmente específicos para un cierto conjunto de datos [Reed, 1993]. Asimismo Bishop [Bishop, 1996] presentó varios algoritmos de poda para eliminar los pesos y las conexiones que no contribuyen al rendimiento de los modelos neuronales mediante diversas heurísticas.

Por último, como se ha comentado anteriormente, estas técnicas orientadas a la localización de una arquitectura óptima han visto extendido su uso junto con otras técnicas, con el objetivo de determinar y cuantificar la importancia de las variables de entrada en un modelo neuronal. Esto se debe a que a diferencia de los modelos estadísticos clásicos, en un modelo neuronal no es evidente conocer el efecto que tiene cada variable explicativa sobre las variables de respuesta. En §2.4.3.3 se detallarán los trabajos más relevantes encontrados en este campo.

2.4.3.3. Selección de las variables descriptivas

Las RNA han demostrado en numerosos ámbitos su gran capacidad predictiva, superior a las aproximaciones estadísticas tradicionales. No obstante, en numerosas ocasiones han sido etiquetadas como cajas negras debido a la poca información que se obtiene acerca de la influencia de las variables independientes o de entrada sobre el proceso de predicción. Esto es debido a que el conocimiento y experiencia de una RNA está codificada de forma implícita a través del conjunto de pesos numéricos repartidos en las conexiones de la arquitectura de la red (cf. §2.4.3.5). Esto conlleva que las variables de entrada son introducidas en el modelo neuronal y la salida obtenida no facilita información acerca de las relaciones existentes entre estas variables, y por tanto, tampoco se obtiene explicación a los mecanismos subyacentes modelados por la red. Esta naturaleza de caja negra es la mayor desventaja de las RNA sobre las aproximaciones tradicionales estadísticas, ya que éstas pueden cuantificar la influencia de las variables independientes sobre el proceso de modelado y además facilitan una medida del grado de confianza teniendo en cuenta la contribución de cada una.

Por otro lado está claramente demostrada la relación existente entre dichos pesos de las conexiones y las variables de entrada de la red. Por ello en ciertos casos disponer de una número elevado de variables de entrada puede complicar el proceso de aprendizaje de la RNA. En este caso la red se encuentra sobreparametrizada, lo que complica el entrenamiento y puede originar sobreaprendizaje [Hippert et al., 2001]. Esto sucede cuando algunas de estas variables no aportan información relevante a la red, o aparecen relaciones entre ellas que afectan a la salida que tiene que calcular [Isasi and Galvan, 2004].

Una de las características más atractivas de las RNA y que permite que se ajuste a gran cantidad de dominios es su capacidad para aproximar funciones no lineales. En estos casos, la variable respuesta depende de las variables independientes según una función polinómica, dando lugar a una regresión múltiple que puede planearse agregando las condiciones polinómicas al modelo lineal básico. De esta forma y aplicando ciertas transformaciones a las variables, se puede convertir el modelo no lineal en uno lineal que puede resolverse entonces por el método de mínimos cuadrados. En este caso el valor de la variable de salida o dependiente (y) se ajusta a partir de una serie de variables independientes o variables de entrada (x_i). Teniendo en cuenta la ecuación asociada a una regresión múltiple para aproximar la función de salida se obtiene [Bishop, 1996]:

$$y = \epsilon + \alpha_1 x_1 + \alpha_2 x_2 + \dots + \alpha_n x_n \quad (2.29)$$

donde los α_i son parámetros desconocidos que deben ser estimados, y ϵ es un componente de error. Por otro lado, es posible que exista cierta colinealidad entre las variables independientes. Si se incluyen en un mismo modelo dos variables relacionadas, es probable que ninguna de las dos sea considerada significativa. Sin embargo si solo se introduce una de ellas, el modelo podría considerarlas más relevantes. El objetivo perseguido por tanto es encontrar las variables explicativas, i.e., incluir en el modelo solo aquellas variables que puedan influir en la respuesta, descartando aquellas que no aporten información. También se busca localizar las posibles interacciones entre las variables independientes que afecten a la variable respuesta.

Además es importante resaltar que, hasta el momento, el mayor esfuerzo en las investigaciones realizadas sobre RNA se ha centrado en el desarrollo de nuevos algoritmos de aprendizaje, la exploración de nuevas arquitecturas de sistemas neuronales y la expansión hacia nuevos campos de aplicación. Sin embargo, se ha dedicado poca atención a desarrollar procedimientos que permitan comprender la naturaleza de las representaciones internas generadas por la red para responder ante un problema determinado. Por el contrario, las RNA se han presentado al usuario como una especie de caja negra cuyo complejo trabajo, en cierta medida difuso, transforma las entradas en salidas predichas. En otras palabras, no se puede saber inmediatamente cómo los pesos de la red o los valores de activación de las neuronas ocultas están relacionados con el conjunto de datos manejados. Así, a diferencia de los modelos estadísticos clásicos, no parece tan evidente conocer en una red el efecto que tiene cada variable explicativa sobre cada una de las variables respuesta.

No obstante, esta percepción acerca de las RNA como una caja negra, no es del todo cierta. De hecho, desde finales de los años 80 han surgido diversos intentos por desarrollar una metodología que permitiera interpretar lo aprendido por la red, aunque son escasos los trabajos orientados a la validación de tales procedimientos. Estos esfuerzos no tienen únicamente por objeto determinar las variables de entrada con mayor peso o importancia sobre la salida de la red, sino también identificar y eliminar del modelo las variables redundantes o irrelevantes, i.e. aquellas variables que pueden expresarse en términos de otras variables de entrada o aquellas que simplemente no contribuyen en la predicción. Este último aspecto no es trivial debido a que las RNA, al igual que otros tipos de modelado, se ven afectadas por la denominada maldición de la dimensionalidad (*curse of dimensionality*) [Devijver and Kittler, 1982, Bishop, 1996]. De forma resumida esta teoría expone que el número de datos necesarios para especificar una función, en general, crece exponencialmente con la dimensión del espacio de entrada. Por tanto, la adecuada selección y por tanto la reducción de la dimensión del espacio de entrada mediante la eliminación de variables redundantes o irrelevantes permite trabajar con un menor número de datos, lo que acelera el proceso de convergencia de los pesos de la red. En base a las demostraciones expuestas en el trabajo de Baum et al [Baum and Haussler, 1989] se espera obtener un error de generalización más bajo. Pero además al identificar únicamente las características más relevantes, el coste de una futura recolección de datos se reduce y, por último, la exclusión de atributos irrelevantes, permite obtener un modelo neuronal más simple, lo que reduce el tiempo necesario para clasificar o predecir futuros patrones.

En este punto es necesario comentar que existe una gran relación entre las técnicas de crecimiento y poda detalladas en §2.4.3, y las orientadas a seleccionar y cuantificar la contribución de las variables de entrada. Las primeras buscan reducir la complejidad de los modelos neuronales utilizados para cada problema seleccionando la mejor arquitectura, i.e. número de neuronas y topología de las conexiones, entre una serie de candidatos, pero sin tener en cuenta posibles interacciones entre las variables de entrada y la salida. En contraste los métodos para cuantificar la contribución de las variables independientes son más complicados y su utilización es por tanto más restringida en los problemas ingenieriles. La mayoría de los trabajos publicados hasta la fecha, para conocer el efecto o importancia de cada variable de entrada sobre la salida de la red, pueden englobarse en tres corrientes o categorías diferenciadas: diagramas de interpretación, análisis basado en la magnitud de los pesos y análisis de sensibilidad.

- Los diagramas de interpretación neuronales (*Neural interpretation diagram*) facilitan una interpretación visual de las conexiones y los pesos entre las neuronas [Olden and Jackson, 2002]. En ellos se representan las magnitudes relativas de cada una de las conexiones mediante líneas de diferente grosor (líneas finas representan conexiones con pesos bajos y líneas gruesas conexiones con pesos altos) y tonalidad (líneas oscuras representan conexiones positivas, i.e. excitadoras, y líneas más claras representan conexiones negativas, inhibitorias). Otra variante también visual a este

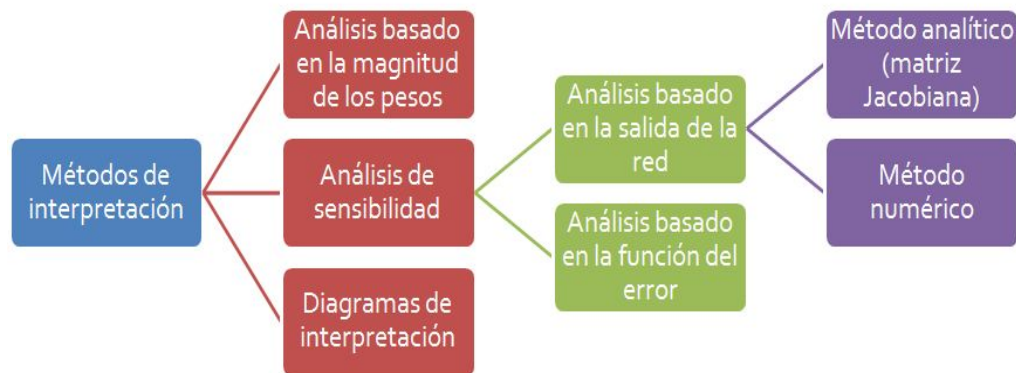


Figura 2.17: Diferentes categorías para cuantificar la contribución de las variables de entrada sobre los modelos neuronales

método son los diagramas de Hilton [Despagne and Massart, 1998], donde los pesos se muestran en forma de caja con tamaño proporcional a su magnitud. Si bien la interpretación y detección de interrelaciones entre las entradas y las salidas en estas técnicas es sencilla al poder realizarse de forma visual, su utilización en ciertos problemas de gran envergadura es inviable debido a la cantidad y complejidad de las conexiones existentes en los modelos neuronales. Por último, muchos investigadores han puesto en duda la utilidad real de estas aproximaciones visuales al carecer de base matemática sobre la que sustentar las conclusiones obtenidas.

- El análisis basado en la magnitud de los pesos, agrupa aquellos procedimientos que se basan exclusivamente en los valores almacenados en la matriz estática de pesos. El propósito de determinar la influencia relativa de cada variable de entrada sobre cada una de las salidas de la red. Se han propuesto diferentes ecuaciones basadas en la magnitud de los pesos, siendo la más representativa la presentada por Garson en 1991 [Garson, 1991]. Esta aproximación facilita una heurística para determinar la influencia global de cada variable, pero utiliza el valor absoluto de los pesos para calcular las contribuciones de cada entrada y por tanto no facilita la dirección ni la fuerza de la relación entre las variables de entradas y salida, lo que complica la detección de interacciones. Por su parte, Tchaban et al. [Tchaban et al., 1998] han presentado una variante para este tipo de análisis denominada *weight product* que incorpora a la información proporcionada por los pesos, el cociente entre el valor de la variable de entrada y el valor de la salida. Por otro lado, un aspecto fundamental es decidir con que valor umbral se debe eliminar o mantener una conexión dentro de la red. Para aclarar este punto, Olden y Jackson [Olden and Jackson, 2002] publicaron una método de prueba aleatorio basado en la poda de la red mediante una aproximación estadística, eliminando aquellas conexiones con pesos nulos y

que por tanto no tienen influencia sobre la salida, y un método de selección para identificar aquellas variables independientes que contribuyan a las predicciones de la red.

- El análisis de sensibilidad permite reflejar el impacto que produce un atributo o variable de entrada sobre la salida o entrada de la red al producirse una variación sobre dicho atributo. Dicha sensibilidad se define como la expectación matemática de las desviaciones sobre la salida o sobre el error con respecto al conjunto global de patrones de entrada. Por tanto el análisis de sensibilidad puede estar basado bien en la medición del efecto que se observa en una salida o bien en el error cometido debido al cambio que se produce en una entrada.

- La primera opción suele ser la más utilizada a la hora de realizar el análisis de sensibilidad, y consiste en estudiar el efecto de las entradas directamente en la salida estimada por la red. Esto se puede realizar de forma similar al análisis aplicado sobre una función de error, tradicionalmente esto conlleva variar los valores de cada una de las variables a través de todo su rango manteniendo el resto constantes. A continuación se valoran las contribuciones individuales de cada una de las entradas. Como es lógico esta aproximación es poco recomendable cuando el número de variables es elevado, ya que el número de combinaciones impediría su utilización dentro de unos plazos razonables [Olden and Jackson, 2002]. Por esta razón otras alternativas limitan el número de valores a medir para cada variable, e.g. máximo, mínimo, cuartiles, percentiles, etc. Gracias a esto, el método de análisis se vuelve más manejable, aunque sigue siendo un gran reto al tener que probar numerosas configuraciones. Lek et al. [Lek et al., 1995] propusieron un algoritmo donde se limitaban a 12 los valores a probar para cada variable mientras el resto se mantenían fijos. Por su parte Song y Ru-Qin [Song and Ru-Qin, 1993] presentaron un método para conocer el grado de influencia de las entradas sobre las salidas, denominado índice de correlación parcial (*partial correlation index*) a partir de la eliminación secuencial de cada una de las variables. Sin embargo su utilización sólo es adecuada para problemas que cuenten con un número restringido de variables. Dimopoulos [Dimopoulos, 1995] y Zurada et al. [Zurada et al., 1994, 1997] postularon sendos métodos basados en la utilización de las derivadas parciales de la salida respecto de las entradas, creando por un lado un perfil de las variaciones que produce las modificaciones sobre cada una de las entradas y por otro clasificando las contribuciones de cada una de ellas sobre la salida de la red.

Otras técnicas con mayor influencia matemática, como la propuesta por Despaigne y Massart [Despaigne and Massart, 1998], ofrecen una estimación cuantitativa de la contribución de cada variable. Para ello calculan la varianza sobre la salida al modificar las entradas de la red, lo que permite obtener un

criterio de estimación global (varianza de la respuesta) en contrapartida con las técnicas tradicionales donde sólo se tiene un criterio local relacionado con la topología de la RNA (magnitudes de los pesos o influencia de su eliminación). Por último, un procedimiento que goza de una mayor fundamentación matemática se basa en la obtención de la matriz Jacobiana mediante el cálculo de las derivadas parciales de las salidas con respecto a las entradas, el cual se trata de la versión analítica del análisis de sensibilidad [Moreno and Pol, 2003].

- La segunda modalidad permite conocer si una determinada entrada es importante en la predicción realizada por el modelo, en el caso de que el error aumente sensiblemente ante el cambio provocado sobre ella. En este caso las estrategias utilizadas de forma habitual son contrarias a las técnicas clásicas de clasificación basadas en árboles de decisión, ya que en lugar de elegir un atributo cada vez, se empieza con un conjunto completo de los mismos y se van eliminando los atributos irrelevantes uno a uno. Para ello se mide la relevancia o irrelevancia de las entradas a través de la fortaleza de las conexiones existentes entre la capa de entrada y la capa oculta de la red. Existen diversas variantes a la hora de realizar este análisis de sensibilidad. Una de ellas consiste en ir variando el valor de una de las variables de entrada a lo largo de todo su rango, mediante la aplicación de pequeños incrementos o perturbaciones, mientras se mantienen los valores originales de las demás variables de entrada. También se puede optar por restringir la entrada de interés a un valor fijo, e.g. el valor promedio, para todos los patrones, eliminar directamente esa entrada o anular los pesos de las conexiones asociadas a cada una de las entradas [Nord and Jacobsson, 1998, Despagne and Massart, 1998]. El trabajo presentado por Anderson et al. [Andersson et al., 2000] es una variante de ésta última técnica, donde se propone un algoritmo para la localización arbitraria de las interacciones entre las variables. Yao et al. [Yao et al., 1998] plantearon un método para analizar el efecto que producen pequeños cambios en cada una de las entradas sobre el error calculado por la RNA. El algoritmo ajusta los valores de una de las variables de entrada mientras mantiene el resto intactas. Aquella variable que más afecte al cálculo del error será la que mayor influencia tenga para el modelo neuronal, por lo que esta propuesta permite obtener una clasificación de las variables por importancia relativa. Una última variante de interés es la que realizaron Principe et al. [Principe et al., 1999], donde se presenta una medida de la influencia que tienen pequeños cambios en las variables independientes sobre los valores de las funciones de salida.

Una de las primeras alternativas en aparecer y una de las más extendidas es la conocida como método paso a paso (*stepwise method*). Éste consiste en añadir o eliminar paso a paso una única variable de entrada y analizar el efecto sobre la salida de la red. En función del tipo de cambios producido sobre el error,

las variables de entrada podrán ser clasificadas bajo diferentes categorías [Sung, 1998]. Por último, otra posibilidad de gran difusión consiste en las llamadas medidas de importancia o notabilidad de las variables (*saliency measure*). Belue y Bauer [Belue and Bauer, 1995] presentaron varias de estas medidas para los atributos basándose en la derivada de la función del error, en los pesos de la red o en ambas. El problema asociado al cálculo de estos indicadores reside en la necesidad de repetir los experimentos al menos 30 veces para encontrar una media y desviación de la medida fiable. A partir de este trabajo Setiono y Liu [Setiono and Liu, 1997], y con el objetivo de reducir los tiempos de computación eligieron una variante del algoritmo RP, el algoritmo de *QuasiNewton* (cf. §2.4.4). Por otro lado, introdujeron términos de penalización sobre la función de error para detectar variables irrelevantes o redundantes, y midieron la precisión de la red mediante una función de error de entropía cruzada.

Olden y Jackson [Olden and Jackson, 2002, Olden et al., 2004] en 2002 y 2004, y Gevrey et al. [Gevrey et al., 2003] en 2003, publicaron una revisión de algunas de las técnicas enumeradas anteriormente para analizar las contribuciones de las variables.

Los métodos descritos con anterioridad han demostrado su utilidad en determinadas tareas de predicción, sin embargo, cuentan con una serie de limitaciones. En los trabajos de simulación realizados, el análisis basado en la magnitud de los pesos no ha sido útil para ordenar las variables de entrada en función de su importancia sobre la salida [Sarle, 2000]. El análisis de sensibilidad sobre el error o sobre la salida consistente en añadir incrementos o perturbaciones se basa en la utilización de variables de entrada cuya naturaleza es cuantitativa, ya que no sería del todo correcto añadir incrementos a variables nominales, i.e. variables que toman valores discretos. Este hecho limita el número de campos de aplicación de las RNA ya que en algunos de ellos es necesario el manejo de variables discretas (e.g. género: 0 = varón, 1 = mujer).

Otra clasificación alternativa de los métodos utilizados para cuantificar la influencia de las variables de entrada sobre las salidas fue facilitada Zeng et al. en 2003 [Zeng et al., 2003] y la ampliada en 2006 por Papadokostantakis et al. [Papadokostantakis et al., 2006]. Esta agrupaba los métodos en función de cuando se aplican al problema, si antes, durante o después de la creación del modelo neuronal. Dentro de la primera categoría se engloban aquellas alternativas que sólo tienen en cuenta a los datos, ya que todavía no existe ningún modelo neuronal, y el objetivo es obtener mayor información de los datos para ayudar al posterior desarrollo de la RNA. Los métodos de información teóricos, basados en la teoría de la información, son útiles para desarrollar medidas no lineales de la asociación entre variables y reducir el vector de variables de entrada [Sridhar et al., 1998]. En la segunda categoría se engloban aquellos métodos encaminados a incorporar mayor conocimiento al proceso, mediante las técnicas de regularización y la adicción de términos de error para penalizar la complejidad de la red [Reed, 1993]. Otras alternativas

son la regularización Bayesiana expresada por MacKay [MacKay, 1992] que permite la determinación de reglas para detener el proceso de poda o crecimiento de la red así como detectar la mejor configuración de variables, penalizando aquellas topologías más complejas. La tercera categoría incluye las aproximaciones que se aplican a los modelos neuronales después de las fases de entrenamiento y validación, i.e. sobre aquellas RNA que se consideran satisfactorias para un problema dado. Dentro de esta podemos encontrar las propuestas de Nord y Jacobsson [Nord and Jacobsson, 1998] o Andersson et al. [Andersson et al., 2000] para revelar la influencia de las variables de entrada y extraer reglas de decisión de los modelos neuronales de caja negra.

Por último, otras técnicas más complejas, no englobadas en las categorías anteriores, son el algoritmo OBS (*Optimal Brain Damage*) propuesto por Le Cun et al. [Le Cun et al., 1990], basado en las segundas derivadas de la función de error, o la regularización de la complejidad [Haykin, 1994], que busca encontrar el equilibrio entre la fiabilidad de los datos y la calidad del modelo mediante una función de riesgo total, o el empleo de algoritmos genéticos o lógica borrosa [Reed, 1993].

2.4.3.4. Funciones de activación

Otro de los factores que afectan a la precisión final de una arquitectura de RNA, es el tipo de función de activación que se selecciona, bajo la premisa continuidad y diferenciabilidad, para cada una de las neuronas [Shi and Hagiwara, 2000].

La utilidad de esta función es realizar la transferencia, dependiendo del estado de activación de la neurona, desde la entrada hasta la salida de la red aplicando dicha función. Es común que la función de activación sea idéntica para todas las neuronas de una misma capa, e incluso en algunos casos para todas las neuronas, siendo el diseñador el que decide cual utilizar. Dicha elección se realiza únicamente basándose en los valores de activación que se desee que alcancen las neuronas.

En la resolución de problemas ingenieriles, como comentan Berke y Hajela [Berke and Hajela, 1992] e Ince [Ince, 2004], la función de transferencia más utilizada es la sigmoideal. Esto se ve ratificado por la gran cantidad de publicaciones dentro del dominio de este trabajo, donde se ha utilizado esta función en los últimos años con resultados satisfactorios [Pu and Mesbahi, 2006, Wang et al., 2006, Liujie et al., 2006]. No obstante, algunos investigadores recurren a la función tangente hiperbólica, [Worden and Staszewski, 2000, Bisagni et al., 2002, Lanzi et al., 2004], debido a las características del problema a resolver y de los datos disponibles para realizar el entrenamiento.

Por último comentar que si bien la función lineal o identidad no suele emplearse de forma habitual [Hilera and Martinez, 1995], su utilización como función de activación para las neuronas de capa de salida ha dado resultados satisfactorios en ciertos estudios [Omar

et al., 2000, Rafiq et al., 2001, Liujie et al., 2006].

2.4.3.5. Inicialización de los pesos de red

La capacidad de adquirir conocimiento de una RNA se encuentra representada en los pesos de las conexiones entre las neuronas. Estos pesos son valores asociados a cada uno de los enlaces que existen entre las neuronas de diferentes capas. Durante el entrenamiento el algoritmo de aprendizaje se encarga de modificarlos para conseguir minimizar los errores devueltos por la red. Cuando el algoritmo detecta que el error ha alcanzado un valor mínimo, congela los pesos y comprueba el rendimiento usando datos nuevos, no mostrados en el entrenamiento.

El valor inicial de los pesos influye de forma considerable en el aprendizaje y velocidad de convergencia de la red. Los pesos se inicializan de forma aleatoria antes de comenzar el entrenamiento, sin embargo es necesario elegir un rango acorde con las dimensiones de los datos de entrada. Debido a la importancia de una buena selección de los pesos para asegurar la convergencia de la red, numerosos autores han propuesto diversas heurísticas para facilitar esta tarea [Fahlman, 1988, Nguyen and Widrow, 1990, Haykin, 1994]. Entre las conclusiones más interesantes destaca que no se debe empezar con valores nulos porque el aprendizaje no progresaría ya que las salidas de las neuronas serían nulas y el incremento de los pesos también. Este fenómeno se conoce como parálisis de la red y sucede cuando los pesos quedan ajustados a valores elevados, por lo que las unidades de proceso tienen que operar con una activación muy próxima a uno. Esto conlleva que el gradiente del error tienda a cero por lo que no se producen modificaciones en los pesos y el aprendizaje queda detenido.

Por otra parte, unos valores altos pueden producir que las neuronas ocultas sean demasiado activas o estén inactivas para todos los patrones del entrenamiento [Fang et al., 2005]. De la misma forma, Le Cun et al. [Le Cun et al., 1989] indican que cuando a los pesos de la red se les asignan valores iniciales grandes, es probable que las neuronas de la red entren en saturación. Si esto ocurre, los gradientes locales en el algoritmo RP asumen valores pequeños, lo cual a su vez causará que el proceso de aprendizaje sea más lento. Sin embargo, si a los pesos se les asignan valores iniciales pequeños, el algoritmo RP puede operar en un área demasiado plana alrededor del origen de la superficie de error. Esta situación se pone de manifiesto en el caso de las funciones de activación antisimétricas como la función tangente hiperbólica, ya que en ellas pueden aparecer puntos de silla sobre la función del error. Por estos motivos, es conveniente evitar el uso de valores demasiado grandes o pequeños para la inicialización de los pesos.

En el trabajo de Le Cun et al. [Le Cun et al., 1989], se propone como estrategia para inicializar los pesos, para la función tangente hiperbólica, utilizar una distribución

uniforme donde se elegirán los pesos con una media de cero y una varianza σ_w igual a:

$$\sigma_w = m^{-1/2} \quad (2.30)$$

donde m es el número de pesos de la neurona.

El trabajo de Basheer y Hajmeer [Basheer and Hajmeer, 2000] ofrece una revisión de las investigaciones más interesantes llevadas a cabo en este aspecto. En ellas se indican posibles rangos de valores y los efectos de una mala elección de los valores iniciales. Por último comentar que la heurística más utilizada dentro de la literatura consultada es la propuesta por Haykin en 1994 [Haykin, 1994], donde los valores para inicializar cada neurona se sitúan dentro del rango $(-\frac{r}{N_j}, \frac{r}{N_j})$, siendo r un número real dependiente de la función de activación y N_j el número de conexiones que alimentan la neurona en cuestión.

2.4.3.6. Comité de redes

Muchos investigadores han desarrollado diversas técnicas para combinar las predicciones de múltiples clasificadores con el objetivo de producir un clasificador más preciso y robusto [Breiman, 1996]. El clasificador resultante, conocido como comité, es generalmente más preciso que cualquiera de los individuales que lo forman. Por tanto, un comité de redes consiste en un conjunto de redes entrenadas que combinan sus predicciones para clasificar nuevos patrones con mayor precisión. Bajo este esquema no se depende de una sola red ya que la respuesta es el promedio entre los miembros del comité. Las dos variantes más utilizadas se conocen como *bagging* [Breiman, 1996] y *boosting* [Freund and Schapire, 1996]. Estos métodos se basan en técnicas de muestreo como el *bootstrapping* para obtener diferentes conjuntos de datos de entrenamiento para cada uno de los clasificadores.

En la Figura 2.18 se puede observar la arquitectura básica de un comité de redes. Cada una de las redes individuales es entrenada usando un conjunto de datos obtenido mediante el muestreo del conjunto original. A continuación la salida de cada red es combinada para obtener la predicción del comité. Muchos trabajos [Alpaydin, 1993, Breiman, 1996] han demostrado que uno de los métodos más efectivos y simples para estimar esta salida es obtener la media de las predicciones de la red. Para que el comité sea realmente útil necesita que las redes individuales difieran para que el conjunto global presente un mejor rendimiento. Dentro de los comités con RNA las alternativas más habituales son la utilización de diferentes topologías, pesos iniciales y parámetros o entrenar sólo sobre partes del conjunto de datos [Hansen and Salamon, 1990, Alpaydin, 1993, Maclin and Shavlik, 1995].

El primero de estos métodos, *bagging*, se basa en la técnica estadística de muestreo *bootstrapping* para obtener los datos necesarios con que entrenar cada red del comité. Para ello realiza una redistribución aleatoria con repeticiones del conjunto de datos

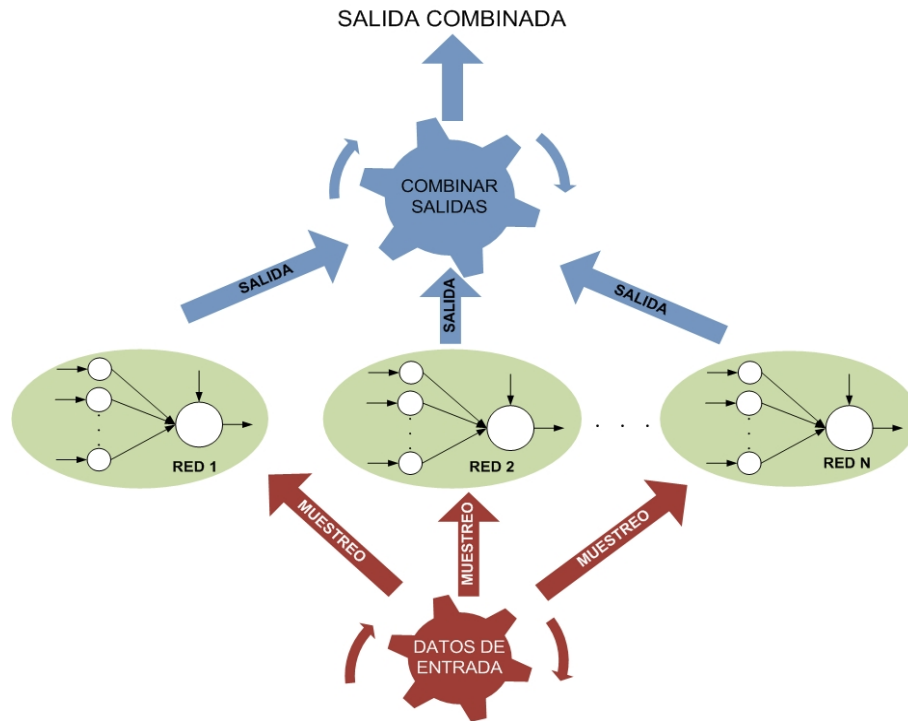


Figura 2.18: Comité de RNA

de entrenamiento, generando tantos subconjuntos de patrones, de igual tamaño que el original, como redes formen el comité. La elección de los patrones para cada conjunto es aleatoria y todos tienen la misma probabilidad de ser seleccionados por lo que es posible que algunos sean elegidos varias veces mientras que otros ninguna.

La segunda técnica, *boosting* presenta un funcionamiento similar, pero en este caso el conjunto de datos que utiliza cada clasificador depende de los resultados obtenidos por el anterior. Los ejemplos que han sido mal clasificados por la red anterior, son seleccionados con una mayor frecuencia. Por tanto este método genera futuros nuevos clasificadores que son capaces de predecir ejemplos con mejor precisión que el actual.

La principal diferencia entre ambos métodos radica en la selección de los datos para entrenar cada red. Mientras que *bagging* es estocástico, *boosting* es dependiente del rendimiento de los clasificadores anteriores. Además el primero es un método de voto equitativo para cada clasificador, mientras que el segundo utiliza voto ponderado en función del rendimiento de cada clasificador.

Finalmente, según los estudios de Opitz y Maclin [Opitz and Maclin, 1999], los comités que utilizan *bagging* presentan mejores resultados que los clasificadores individuales, mientras que los que emplean *boosting* son más precisos en ciertos dominios si los datos presentan ruido. Sin embargo estos últimos, al utilizar RNA como clasificadores

dentro del comité, pueden presentar un rendimiento inferior a los modelos neuronales individuales. Por tanto se considera que los comités con *bagging* son mejores para la mayoría de aplicaciones.

2.4.4. Algoritmo de aprendizaje

El algoritmo de entrenamiento de una RNA es el mecanismo mediante el cual se van adaptando todos los parámetros de la red y gracias a éste las RNA presentan la capacidad de resolver problemas de diferente índole. En el caso del PMC, se suele recurrir al algoritmo RP como mecanismo de aprendizaje. Éste es de tipo supervisado, donde los errores se propagan hacia atrás tratando de localizar el valor mínimo de la función que describe dicho error. Para ello se utiliza la técnica conocida como método de descenso del gradiente (cf. §2.3.3).

Si bien la aparición de este algoritmo supuso el relanzamiento de la arquitectura PMC, su utilización en algunas situaciones conlleva una serie de inconvenientes. En primer lugar es posible que debido a una tasa de convergencia lenta, el proceso de aprendizaje sea poco eficiente ; y en segundo lugar la aparición de mínimos locales puede conllevar resultados poco fiables [Bello, 1992].

Para aquellas situaciones en la que la tasa de convergencia dificulta la resolución del problema tratado, se puede optar por alguna de las aproximaciones conocidas como métodos de segundo de orden surgidas a partir del método del gradiente. Este conjunto de algoritmos que no sólo recurren a la información de la pendiente, sino también a la curvatura de la superficie. Entre ellas destacan por su amplia utilización y ventajas mostradas los algoritmos del gradiente conjugado, el de *Levenberg-Marquardt* o el de *de Quasi-Newton* [Reklaitis et al., 1983]. La limitación de estos métodos es que su uso requiere un alto nivel de experiencia y además algunos son intensivos desde el punto de vista computacional y, como consecuencia, se deberían utilizar en aquellos casos en que el número de pesos a estimar no es muy grande.

Por otro lado para evitar que el entrenamiento se detenga al detectar un mínimo local de la función del error, la técnica más utilizada consiste en incluir el término o constante *momentum* (μ) durante el entrenamiento. Otras investigaciones en este sentido proponen el enfriamiento simulado (*simulated annealing*) propuesto por Kirkpatrick et al. en el año 1983 [Kirkpatrick et al., 1983] o el empleo de técnicas de optimización genética formulada por Yao en 1999 [Yao, 1999].

Por último, como ya se ha comentado, el algoritmo RP tiene como objetivo minimizar la función de error localizando su mínimo global. Sin embargo, ha de tenerse en cuenta que no siempre es necesario o posible alcanzar este punto dentro de unos límites razonables de tiempo y esfuerzo computacional. Para resolver estos casos pueden utilizarse, de

forma conjunta o individualmente, diferentes criterios de parada que acorten el proceso de aprendizaje.

En los siguientes apartados se detallan en profundidad los aspectos relativos al algoritmo de aprendizaje utilizado por una RNA comentados con anterioridad, i.e. parámetros del algoritmo, posibles criterios de parada y modos de entrenamiento disponibles.

2.4.4.1. Parámetros del algoritmo de aprendizaje

Existen varios parámetros asociados al algoritmo RP que se pueden implementar para corregir las carencias del algoritmo RP: tasa de convergencia lenta y caída en mínimos locales. Las dos alternativas más importantes son la tasa o factor de aprendizaje η y la constante *momentum* μ .

Las investigaciones realizadas hasta la fecha han demostrado que el primero de estos parámetros, η , que afecta de forma significativa a la precisión y velocidad del entrenamiento [Zhang and Subbarayan, 2002]. Si se opta por un valor alto, se conseguirá acelerar la búsqueda, favoreciendo una convergencia rápida, pero a costa de oscilar sobre la superficie del error, saltándose el mínimo global. Este fenómeno se conoce como inestabilidad temporal y podría hacer que el error global no fuera detectado y por tanto nunca se alcanzará la convergencia. Sin embargo, un valor bajo implica que la búsqueda se dirigirá de forma constante hacia el mínimo global aunque de forma mucho más lenta, ya que se realizarán un gran número de iteraciones. Rafiq et al. publicaron en 2001 [Rafiq et al., 2001] una forma práctica de conocer en que dirección debe modificarse el valor de dicho parámetro. En su investigación expresaban que si durante el entrenamiento los pesos de la red eran modificados de forma consistente en la misma dirección, entonces η podía ser incrementada para mejorar el proceso. Por otro lado, si los pesos de la red cambiaban en direcciones alternas, se podía decrementar. Por último indicaron que la magnitud del cambio de la tasa de aprendizaje es dependiente de los cambios de los pesos.

En la arquitectura de un PMC idealmente todas las neuronas de un PMC deberían aprender a la misma velocidad. Sin embargo, es habitual que las últimas capas tengan gradientes locales mayores que las capas en la parte delantera de la red. Por lo tanto, se debería asignar un valor más pequeño a η en las últimas capas que en las primeras. Además las neuronas con muchas entradas deberían tener una tasa de aprendizaje menor que las neuronas con pocas entradas de manera de mantener un tiempo de aprendizaje similar para todas ellas. En el trabajo de Le Cun et al. [Le Cun et al., 1989], se sugiere que para una neurona dada el valor de η debería ser inversamente proporcional a la raíz cuadrada de los pesos conectados a esa neurona. Finalmente, Kung y Hwang [Kung and Hwang, 1988] establecieron que la tasa de aprendizaje óptima es inversamente proporcional al número de neuronas ocultas. Más detalladamente, este

parámetro presenta una relación inversamente proporcional a la primera derivada de la función de activación.

Por otro lado, a la variante clásica del algoritmo RP se le puede incluir la constante μ , con lo que el entrenamiento se acelera de manera considerable. La constante *momentum* se utiliza comúnmente en la actualización de pesos para ayudar al escape de la búsqueda de mínimos locales y reducir la probabilidad de inestabilidad. μ acelera la actualización de los pesos cuando existe la necesidad de reducir η para evitar oscilaciones. Un valor alto de μ reduce el riesgo que el entrenamiento quede atrapado en mínimos locales, pero al igual que ocurre con η aumenta el riesgo de sobrepasar la solución óptima. Un valor $\mu > 1.0$ puede implicar un incremento de los pesos excesivo por lo que podría aparecer cierta inestabilidad durante el entrenamiento. Por el contrario un valor excesivamente pequeño de μ conlleva un aprendizaje demasiado lento. Para encontrar un punto de equilibrio se pueden utilizar tanto valores de μ constantes o adaptables en función de las circunstancias del problema a resolver.

Dependiendo del problema a resolver parece que el éxito del entrenamiento varía en función del valor μ seleccionado, por lo que se suele recurrir a un procedimiento de prueba y error. Por último, entre sus desventajas más importantes es necesario tener en cuenta que el empleo de la constante *momentum* implica duplicar el espacio de almacenamiento de los pesos actuales y los de las iteraciones anteriores. Siguiendo las pautas de Haykin [Haykin, 1994], para que todas las neuronas aprendan al mismo ritmo se penalizarán los valores de η de capa a capa con un factor de $\frac{2}{5}$.

Los valores de estos dos parámetros, tasa de aprendizaje (η) y *momentum* (μ), pueden ser constantes o pueden variar durante la optimización de la red, aunque según los estudios realizados por Bishop existen una serie de inconvenientes si estos valores se mantienen fijos [Bishop, 1996]. Por consiguiente, ambos parámetros se modifican frecuentemente, ya sea incrementando o disminuyendo su valor en función de si el error aumenta o disminuye respectivamente, durante la iteración previa del proceso de optimización de la red [Hagan et al., 1996]. En Basheer y Hajmeer [Basheer and Hajmeer, 2000] se puede encontrar una detallada descripción de las alternativas más utilizadas para encontrar los valores idóneos de estos parámetros, así como las principales aportaciones, referentes a su utilización, obtenidas por los investigadores.

En las Tablas 2.2 y 2.3 se muestran el rango de valores más utilizados en la literatura para cada uno de estos parámetros.

No obstante no existe una regla de uso general válida para aproximar desde el comienzo los valores idóneos en todos los problemas, por lo que existen algunas propuestas destinadas a localizar de forma más efectiva los valores óptimos de estos parámetros. Algunas de ellas como *Learning Rate Scheduling* [Principe et al., 1999] o *Delta Bar Delta* [Jacobs, 1988] permiten actualizar los valores de η

Valor mínimo de η	Valor máximo de η	Fuente
0.0	1.0	[Wythoff, 1993]
0.3	0.6	[Zupan, 1995]
0.0	1.0	[Fu, 1995]
-	0.25	[Swingler, 1996]

Tabla 2.2: Rangos más utilizados en la tasa de aprendizaje

Valor μ	Fuente
[0.4, 0.9]	[Wythoff, 1993]
[0.0, 1.0]	[Hassoun, 1995] y [Fu, 1995]
≈ 1.0	[Henseler, 1995] y [Hertz et al., 1991]
$\eta + \mu \approx 1$	[Zupan, 1995]
0.9	[Swingler, 1996]

Tabla 2.3: Rangos más utilizados en la constante *momentum*

y μ de forma dinámica durante la simulación.

2.4.4.2. Técnicas para la resolución de problemas en el algoritmo de Retropropagación

Para conseguir una resolución eficiente de un problema mediante el uso de RNA, es necesario supervisar a lo largo de los ciclos de entrenamiento, el error que devuelve la red tanto en la fase de aprendizaje como en la fase de validación de la misma. Durante el entrenamiento mediante el algoritmo RP, el diseñador se puede encontrar con una serie de situaciones anómalas que podrían obstaculizar el proceso de obtención de la red más adecuada. En la investigación realizada por Basheer y Hajmeer [Basheer and Hajmeer, 2000] se enumeran las incidencias más comunes que podrían aparecer, así como las posibles causas y actuaciones a seguir para resolverlos:

1. Tanto el error de *train* o entrenamiento como el de test decrecen de forma monótona pero lentamente. Esto es un signo que indica que el sistema está aprendiendo de forma adecuada y mejorando su capacidad de generalización sobre el conjunto de test. Para mejorar la velocidad del aprendizaje, se puede aumentar la tasa de aprendizaje y la constante *momentum*.
2. Para acelerar el proceso de aprendizaje se ha aumentado la tasa de aprendizaje pero el error fluctúa durante los ciclos de aprendizaje. Esto podría indicar que el sistema es inestable.
3. El error sobre el conjunto de *train* o entrenamiento ha decrecido pero el error sobre el conjunto de test se estancó en un nivel superior. Esto indica que la red ha extraído toda la información posible de los patrones. Por lo tanto no se obtienen ventajas

adicionales si se continúa con el entrenamiento para intentar reducir el error sobre el conjunto de test.

4. El error sobre el conjunto de *train* ha bajado a un nivel cercano cero, pero el error sobre el conjunto de test empieza a aumentar después de la bajada inicial. Esto indica que la red ha empezado a memorizar los datos, y su capacidad de generalización está empeorando. Es conveniente chequear el punto donde se obtuvo el error mínimo sobre este conjunto de test en la curva del error y devolver la red a esa situación.
5. El error sobre el conjunto de *train* ha alcanzado un valor cercano a cero pero el error sobre el conjunto de test es bastante alto y no ha cambiado desde el comienzo del entrenamiento. Esto puede ser un signo de que los datos con los que se están entrenando la red incluyen patrones no representativos, ejemplos que están fuera del dominio o datos erróneos. El conjunto de test debería ser inspeccionado y cualquier ejemplo no adecuado debería ser removido y si aparecen patrones que no entran en el dominio de los datos de entrenamiento deberían ser transferidos a dicho conjunto.
6. El error sobre el conjunto de *train* disminuye al comienzo del entrenamiento pero es todavía elevado después de un número considerable de ciclos de entrenamiento. Éste comportamiento es común un cuando se utiliza una tasa de aprendizaje pequeña. El incremento de la tasa de aprendizaje podría paliar en parte el problema. Sin embargo, si el problema persiste o si aparecen nuevas situaciones de inestabilidad, será necesario comprobar la presencia de ejemplos inconsistentes (malos o contradictorios) en el conjunto de datos en entrenamiento. En esta situación, puede ser necesario un análisis estadístico exhaustivo de los valores anómalos para limpiar el subconjunto de entrenamiento. En otras situaciones, puede ser conveniente elegir un esquema de representación de los datos diferente o incrementar el número de neuronas de las capas ocultas.
7. Si los errores, tanto de *train* como de test, son considerablemente altos y no cambian durante los ciclos de entrenamiento, podría indicar que la búsqueda se ha estancado en un mínimo local o que el problema tratado no tiene solución. La mayoría de las veces, sin embargo, lo que parece ser un mínimo local podría deberse a valores demasiado grandes en los pesos lo que complica la saturación de las neuronas cuando las derivadas son cercanas a cero. Para solucionar este problema se podría cambiar la inicialización de los valores asociados a los pesos de las conexiones. Por otro lado si lo que sucede en realidad es que se ha encontrado un mínimo local, se podría aumentar el coeficiente *momentum* para lograr salir del mismo.

2.4.4.3. Criterio de parada

El objetivo de un buen entrenamiento es conseguir que la RNA obtenga resultados óptimos ante las nuevas entradas que se le presenten, evitando que se especialice en los conjuntos de ejemplos proporcionados durante dicho entrenamiento, lo que se conoce como pérdida de generalidad o sobreaprendizaje. Para evitar que se produzca, la arquitectura y el proceso de entrenamiento de la red deben ser diseñados de manera que el

aprendizaje sea robusto y eficiente. De nada sirve disponer de una red que haya aprendido correctamente los patrones de entrenamiento y que no responda de forma adecuada ante nuevos patrones. Es necesario que, durante el proceso de aprendizaje, la red extraiga las características generales de las muestras para poder responder adecuadamente frente a nueva información.

En un proceso de entrenamiento se debe considerar, por una parte, el error de aprendizaje cometido por la red, que se suele calcular como el error cuadrático medio (*Mean Square Error* o MSE) de los resultados proporcionados por la red para el conjunto de patrones de aprendizaje. Por otra parte, el error de generalización, se evalúa empleando un conjunto representativo de patrones diferentes a los utilizados en el entrenamiento. De esta manera, se puede entrenar una red neuronal haciendo uso de un conjunto de aprendizaje, y comprobar su eficiencia real, o error de generalización, mediante un conjunto de test.

Este error MSE, basado en la fórmula del error cuadrático de la Ecuación 2.16, establece la medida de las diferencias en promedio entre los valores obtenidos y los observados. Este valor debería ser cercano a cero para expresar que no existen diferencias notables entre los valores calculados y los esperados:

$$MSE = \sum_{i=1}^n \frac{(\phi_i - \phi_{iobs})^2}{N} \quad (2.31)$$

Donde:

- ϕ_i es el valor pronosticado para el ensayo i
- ϕ_{iobs} es el valor observado para el ensayo i
- N es el número de ensayos realizados

Por tanto la base sobre la que se sustenta este estimador estadístico es la diferencia entre el valor obtenido por la red y el esperado: $(\phi_i - \phi_{iobs})$.

Desde el punto de vista teórico, el proceso de aprendizaje termina cuando se ha encontrado el mínimo de la función de error. En ese momento el error de entrenamiento se estabiliza, i.e. las salidas de la red no cambian de una iteración a otra ya que los parámetros no sufren cambios significativos. Sin embargo encontrar ese mínimo en la función de error puede llevar un tiempo considerable en algunos casos, por lo que se pueden utilizar dos criterios complementarios para detener el aprendizaje, conocidos como *criterios de parada o convergencia* (cf. §2.2.5.1):

1. Obtener un valor prefijado en el error MSE: Se fija un valor mínimo que se quiere alcanzar en el error MSE durante el entrenamiento para acortar el aprendizaje.

2. Detener el entrenamiento después de un número de ciclos o *epochs*: Un *epoch* o ciclo de aprendizaje se produce cuando se le presenta a la red el conjunto completo de patrones de entrenamiento.

En ambos casos es importante tener en cuenta que el aprendizaje ha podido detenerse antes de conseguir llegar al mínimo global de la función de error. Para comprobarlo se puede observar si error MSE de entrenamiento se mantiene constante durante las últimas iteraciones o continua decreciendo. Si sucede esto deberían aumentarse el número de ciclos o reducir el valor del error MSE prefijado para prolongar el aprendizaje.

Por otro lado, en algunas situaciones si se entrena la RNA hasta alcanzar un error demasiado pequeño durante el aprendizaje, la eficacia real del sistema medida con el conjunto de test se degrada. En estos casos, al principio la red se adapta progresivamente al conjunto de aprendizaje, acomodándose al problema y mejorando la generalización. Sin embargo, en un momento dado, el sistema se ajusta demasiado a las particularidades de los patrones empleados durante el entrenamiento, por lo que el error cometido al resolver patrones nuevos aumenta, lo que se conoce como sobreaprendizaje (cf. §2.3.4). De manera ideal, una RNA debería entrenarse hasta un punto óptimo en el que el error de generalización sea mínimo, para lo cual sería necesario evaluar a la vez el error de aprendizaje y el error de generalización cometidos por la red.

Las técnicas más populares, para evitar el sobreaprendizaje durante el entrenamiento se conocen como parada temprana (*early stopping*) y validación cruzada (*cross-validation*). En los trabajos realizados por Sarle [Sarle, 1995] y Liu [Liu, 2006] se pone de manifiesto que es una práctica común usarlas de forma conjunta, buscando reducir el tiempo de ejecución del proceso de aprendizaje y afinar la capacidad de generalización de la red.

El método de parada temprana o parada óptima es la estrategia de regularización más utilizada para determinar la mejor arquitectura de red y evitar el sobreaprendizaje. Se basa en dividir el conjunto de patrones disponibles para el entrenamiento en dos subconjuntos: uno para entrenamiento y otro para su validación. Como se ha comentado, el entrenamiento de un modelo de red no lineal se corresponde con una reducción iterativa de la función del error respecto a un conjunto de datos de entrada. Durante una sesión típica de entrenamiento, el error decrece como función del número de iteraciones del algoritmo RP o cualquiera de sus variantes. Por su parte el error medido contra el conjunto de datos independiente, conocido como validación, presenta un comportamiento habitual representativo: decrece al comienzo del proceso para luego aumentar cuando la red sobreaprende. La utilización de esta técnica permite detener el entrenamiento antes de que la red comience a sobreaprender los datos de entrada, lo que reduce el tiempo de ejecución y mejora la posterior capacidad de generalizar datos nuevos. Según las pautas dictadas por Sarle [Sarle, 1995] para aplicar este método será necesario cumplir los siguientes pasos:

1. Dividir los datos disponibles en dos conjuntos disjuntos: entrenamiento y validación.

2. Usar un número elevado de elementos ocultos.
3. Usar un número reducido aleatorio de valores de inicio.
4. Recurrir a una tasa de aprendizaje lenta.
5. Calcular el error de validación de forma periódica durante el entrenamiento.
6. Parar el entrenamiento cuando el error de validación comience a aumentar.

Este nuevo criterio de convergencia es más fiable pero necesita más recursos computacionales y abundantes datos [Basheer and Hajmeer, 2000]. La red es entrenada a partir de los datos del conjunto de entrenamiento y los pesos son modificados de forma iterativa con el objetivo de minimizar el error cometido entre la salida obtenida por la red y los valores deseados. A la vez que se realiza este cálculo, se va obteniendo el error que comete la red ante los datos del conjunto CV, calculándose así los errores totales sobre los conjuntos de entrenamiento y de validación. De esta forma, no sólo se evalúa el error que comete la red una vez que el proceso de aprendizaje ha concluido, sino también durante dicho proceso.

De forma gráfica, en la Figura 2.19 se puede observar que al igual que se analiza la evolución del error de entrenamiento a lo largo de los ciclos de aprendizaje, se debe analizar también la evolución del error de validación. El comportamiento normal de estos errores es que, a medida que se avanza en el proceso de entrenamiento, estos vayan decreciendo progresivamente. Normalmente, el modelo que recrea la RNA no trabaja tan bien sobre el conjunto CV como lo hace sobre el conjunto de entrenamiento, en el cual se basó su diseño. La curva de aprendizaje de entrenamiento decrece monótonamente para un número creciente de *epochs* en la forma acostumbrada. En contraste, la curva de aprendizaje asociada a la validación decrece de forma monótona hasta un mínimo, entonces empieza a incrementarse mientras continua el entrenamiento. Si se observa la curva de aprendizaje para el error de entrenamiento de la Figura 2.19, puede parecer que el entrenamiento puede mejorar si se continúa más allá del punto mínimo sobre la curva de validación. En realidad, más allá de ese punto la red está aprendiendo de forma esencial ruido contenido en el conjunto de entrenamiento. Por tanto esta heurística sugiere que el punto mínimo sobre la curva de aprendizaje de validación, sea utilizado como un criterio para detener la sesión de entrenamiento.

En el trabajo publicado por Isasi y Galván. en [Isasi and Galvan, 2004], se resumen las distintas situaciones que pueden aparecer al entrenar la red con parada temprana:

1. Ambos errores, de entrenamiento y de validación, permanecen estables después de un cierto número de ciclos. En este caso, se puede afirmar que el aprendizaje ha finalizado con éxito, pues la red ha sido capaz de extraer las características del problema, alcanzando un buen nivel de generalización.

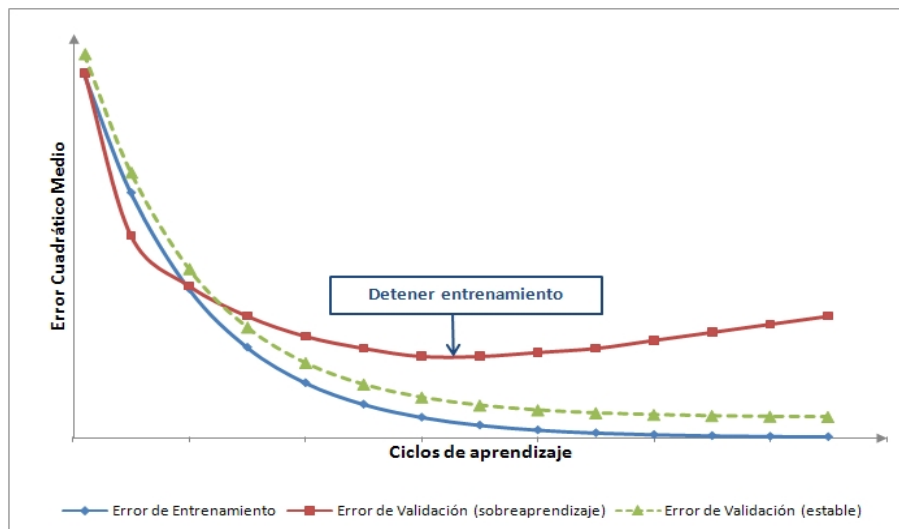


Figura 2.19: Evolución de los errores de entrenamiento y validación

2. A partir de un cierto número de ciclos, el error de validación comienza a aumentar. Este es el momento de detener el entrenamiento, ya que la red está comenzando a especializarse en el conjunto de datos que se le ha proporcionado para extraer conocimiento, siendo menos precisa al incorporar nuevos datos. En este caso no se puede asegurar que el aprendizaje haya acabado con éxito, siendo conveniente detener el proceso en el momento en que el error de validación comenzó a aumentar, ya que se produce un sobreaprendizaje de la red.

Posteriormente, la red es evaluada sobre el conjunto de test, cuyos datos, al no intervenir en el proceso de entrenamiento de la red, sirven para medir de una forma completamente objetiva el nivel de certeza de la respuesta de la RNA ante una nueva entrada, garantizándose así la capacidad de generalización de la red de neuronas. De esta forma, esta medida del error cuadrático total sobre el conjunto de datos del conjunto test proporcionará una estimación insesgada del error de generalización.

Por su parte la validación cruzada es un método con un gran componente estadístico para particionar un conjunto de datos en diferentes subconjuntos. El análisis se realiza sobre el primero de ellos mientras que el resto se utilizan para confirmar y validar el análisis inicial. En teoría de RNA el primero conjunto se denomina de entrenamiento y el resto de validación o test. Su aplicación es de utilidad cuando el número de datos es relativamente pequeño debido a la complejidad o incluso imposibilidad de su obtención. En §2.4.2.2 se detallan las diferentes alternativas existentes para su utilización con RNA en ámbitos con información reducida.

Por tanto, para la utilización de ambas técnicas es necesario dividir los datos disponibles en tres conjuntos disjuntos: entrenamiento, validación y test. Los conjuntos

de entrenamiento y validación se emplean durante la fase de aprendizaje de la red, y el conjunto de test determina de forma objetiva el rendimiento final de la red (cf. §2.3.4). Finalmente comentar que en las situaciones donde no se tienen suficientes patrones para llevar a cabo las dos fases de entrenamiento y prueba con diferentes datos, el método de validación cruzada más comúnmente aplicado en la estimación de clasificadores de patrones es el de KFCV con un valor de $K=10$ [Bishop, 1996].

2.4.4.4. Modos de entrenamiento

El entrenamiento de una RNA implica una reducción gradual del error entre la salida de la red y la salida esperada. Por lo general, existen dos formas diferentes de realizar el proceso de entrenamiento: modo *batch* y modo *secuencial*⁴. En el primero, los pesos se actualizan con el promedio de todos los patrones para cada ciclo o *epoch*⁵, i.e. sólo cuando el conjunto completo de datos de entrenamiento ha sido introducido. En el modo secuencial, el ajuste de los pesos se produce después de que cada patrón o par de datos de entrenamiento es presentado a la red. La elección de un modo u otro es dependiente del problema y decisión del diseñador, aunque como norma general suele utilizarse el modo secuencial [Tarassenko, 1998].

Swingler [Swingler, 1996] dictó una serie de puntos que podían ser tenidos en cuenta a la hora de seleccionar el modo de entrenamiento. El modo *batch* requiere menos actualizaciones de los pesos por lo que tiende a ser más rápido al entrenar. Por otro lado facilita mayor precisión sobre los cambios a realizar en los pesos ya que provee una estimación precisa del vector gradiente. No obstante, cuando los datos de entrenamiento son redundantes (el conjunto de datos contiene varias copias exactas del mismo patrón), se observa que el modo secuencial, a diferencia del modo *batch*, es capaz de tomar ventaja de esta redundancia porque los ejemplos se presentan de a uno en cada pasada. Esto es particularmente efectivo cuando el conjunto de datos es grande y altamente redundante. Finalmente el modo *batch* tiene más probabilidades de quedar atrapado en un mínimo local.

Por su parte, Rafiq et al. [Rafiq et al., 2001] indicaron que el modo *batch* es más sensible al valor inicial de los pesos. Además expusieron una metodología para su utilización, consistente en comenzar el entrenamiento de la red en modo *batch*, analizando para finalizar el error obtenido en la fase de validación. En caso de no ser satisfactorio se podría optar por el modo secuencial.

Por último, se ha observado que una buena práctica para mejorar el comportamiento del algoritmo RP consiste randomizar el orden de presentación de los ejemplos de entrenamiento de un *epoch* al siguiente. Con esto se consigue hacer estocástica la

⁴Algunos autores se refieren a ellos como aprendizaje *offline* y aprendizaje *online* respectivamente

⁵Un *epoch* o ciclo del proceso de entrenamiento comprende la presentación de todos los patrones de entrada a la red

búsqueda en el espacio de pesos, evitando así la posibilidad de limitar al aprendizaje en la evolución de los pesos de la red.

2.4.5. Resumen de reglas y parámetros de optimización

En este apartado dedicado a la optimización se han mostrado las técnicas y heurísticas más utilizadas para mejorar la capacidad de generalización de los modelos neuronales. Entre ellas cabe destacar por su amplia utilización y eficiencia las siguientes:

1. Teoría de la regularización para lograr mejorar la función del error.
2. Método de parada temprana para evitar el sobreaprendizaje.
3. Crecimiento y poda de la red para determinar las mejores arquitecturas de la red.
4. Comité de redes para reducir el esfuerzo de entrenar múltiples redes.
5. Validación cruzada para escoger el mejor modelo usando los mejores datos de entrenamiento posibles.

Por otro lado, para conseguir que el entrenamiento de una RNA sea un proceso eficiente es necesario realizar una correcta selección de los parámetros de diseño que interfieren con el aprendizaje. Éstos pueden englobarse en tres categorías diferentes: arquitectura, algoritmo de aprendizaje y patrones. Si bien se han investigado diferentes reglas para conseguir optimizar su uso, en la mayoría de las ocasiones el cálculo de estos valores se realiza mediante procesos de fuerza bruta o prueba y error.

En este sentido, según Basheer y Jackson [Basheer and Hajmeer, 2000] los seis parámetros más importantes a la hora de realizar el diseño de la red no pueden tomar valores demasiado altos ni demasiado bajos, y por tanto tienen que ser optimizados o seleccionados con precaución. En la Tabla 2.4 se muestran dichos parámetros y los efectos que éstos tienen tanto sobre convergencia del aprendizaje como sobre el rendimiento global de la red.

2.5. Aplicaciones de las Redes de Neuronas Artificiales

2.5.1. Redes de Neuronas Artificiales en ingeniería

En los últimos años, las RNA han sido ampliamente utilizadas para modelar algunas de las actividades humanas en muchas áreas de la ciencia y de la ingeniería. Las primeras

Parámetro de diseño	Valor demasiado grande o alto	Valor demasiado pequeño o bajo
Número de capas ocultas	Sobreaprendizaje de la red (la RNA no generaliza)	Entrenamiento insuficiente o <i>underfitting</i> (la red no es capaz de extraer las reglas incluidas en los datos)
Tasa de aprendizaje (η)	Vuelve la RNA inestable (los pesos oscilan sobre la solución óptima)	Aprendizaje lento
Coefficiente momentum (μ)	Acelera el entrenamiento. Incrementa el riesgo de sobrepasar la solución (inestabilidad)	Ralentiza el entrenamiento. Reduce el riesgo de caer en mínimos locales al utilizar μ .
Número de ciclos de entrenamiento	Mejora la capacidad de memoria de la red (i.e. La red memoriza mejor los datos) pero obtiene una peor generalización sobre los datos no entrenados	Produce una red que es incapaz de representar los datos
Tamaño del conjunto de datos de entrenamiento	Red con una buena capacidad de memoria y generalización	La red es incapaz de explicar totalmente el problema. Red con generalización limitada o pobre
Tamaño del conjunto de test	Confirma la capacidad de generalización de la RNA	Limita las posibilidades de confirmar la capacidad de generalización de la RNA

Tabla 2.4: Efectos de utilizar valores extremos en los parámetros de diseño

aplicaciones de las RNA en ingeniería civil aparecieron a finales de los años 80, siendo la primera publicación al respecto la realizada por Adeli y Yeh en el año 1989 [Adeli and Yeh, 1989]. Estos sistemas son especialmente eficaces para el reconocimiento de patrones y la clasificación. Sin embargo es común referirse a ellos como sistemas de caja negra, ya que facilitan poca información acerca de la posible influencia que tienen las variables independientes a la hora de predecir las salidas [Olden and Jackson, 2002].

Una característica común a la mayoría de problemas de ingeniería, es la necesidad de trabajar con datos incompletos o con ruido. La capacidad de los sistemas de RNA para aprender con este tipo de información les convierte en un candidato prometedor para modelar problemas de este tipo [Rafiq et al., 2001]. Su validez ha sido probada en múltiples entornos ingenieriles relacionados con la predicción, estimación, reconocimiento de patrones y optimización. Adeli realizó, en el año 2001, una amplia retrospectiva de las investigaciones realizadas hasta ese momento relacionadas con la ingeniería civil y estructural [Adeli, 2001].

Por su parte Flood y Kartam publicaron, en el año 1994 [Flood and Kartam, 1994], una revisión del potencial que presentaban este tipo de estructuras computacionales para el desarrollo de proyectos de ingeniería civil. Es de interés la detallada explicación que facilitan, dentro de la resolución de este tipo de problemas, acerca de la influencia de diferentes factores sobre la capacidad de aprender y generalizar de la red.

Otra aplicación interesante de las RNA, de gran difusión en la literatura para la resolución problemas de ingeniería estructural, ha sido su utilización como sustituto o complemento a los métodos de superficie de respuesta [Carpenter and Barthelemy, 1993, Gomes and Awruch, 2004]. La metodología de superficie de respuesta es un conjunto de técnicas matemáticas y estadísticas para modelar y analizar problemas en los que la variable de interés está influenciada por otras. El objetivo es optimizar dicha variable, determinando las condiciones óptimas de operación del sistema [Myers and Montgomery, 1995]. Se trata de un método efectivo para obtener soluciones de diseño óptimas en problemas de gran complejidad y envergadura. De forma convencional éste método utiliza polinomios, por lo que cuando el problema está gobernado por múltiples variables necesita un elevado tiempo de computación para resolverlo.

Una de las tipologías neuronales más utilizadas en el ámbito ingenieril ha sido el PMC. Esto es debido a que puede comportarse como un aproximador universal de cualquier función si tiene un número suficiente de nodos en la capa oculta, lo que le convierte en una herramienta fundamental para poder aproximar funciones en la mayoría de dominios de la ingeniería [Hornik et al., 1989]. Desde el punto de vista estadístico, el PMC es un aproximador universal, ya que puede ser usado en problemas donde no se tiene mucho conocimiento sobre la relación entre las entradas (variables independientes) y las salidas (variables dependientes). Con un pequeño número de neuronas en la capa oculta, el PMC se convierte en un modelo paramétrico que provee una alternativa útil de regresión

polinomial.

2.5.2. Redes de Neuronas Artificiales para la resolución de problemas en ingeniería industrial

Desde la primera publicación realizada por Adeli et al. en 1989, la utilización de las RNA se ha extendido durante los últimos años de forma prolífica en el ámbito de la ingeniería mecánica y de estructuras [Vanluchene and Sun, 1990, Berke and Hajela, 1992, Hajela and Berke, 1992, Waszczyszyn and Ziemianski, 2001]. Los dominios relacionados donde más ha destacado su utilización son: el diseño [Zhang and Subbarayan, 2002], la optimización [Shao and Murotsu, 1997] y el análisis de fiabilidad estructural [Jenkins, 1999, Hosni et al., 2006].

Otras publicaciones de interés donde se ha recurrido a la utilización de RNA han sido la detección de fallos [Kudva et al., 1992, Fang et al., 2005, Jiang et al., 2006], la predicción de fracturas sobre componentes estructurales [Liu et al., 2002, Ince, 2004] y su integración en sistemas de monitorización del desgaste de materiales [Chandrashekhara et al., 1998, Sung et al., 2000].

En la actualidad la gran versatilidad de las RNA ha permitido su utilización en diversos entornos industriales relacionados con lo expuesto anteriormente. Destacan la industria naval [Pu and Mesbahi, 2006], aeroespacial [Kudva et al., 1992, Worden and Staszewski, 2000, Bisagni et al., 2002, LeClerc et al., 2007], automoción [El-Gindy and Palkovics, 1993, Omar et al., 1998, 2000, Shi and Hagiwara, 2000] o incluso la fabricación de barreras contra la onda expansiva de explosiones [Remennikov and Rose, 2007].

Hasta la aparición de las RNA, los tres métodos más utilizados de forma general para la solución de problemas dentro del dominio de ingeniería de materiales habían sido los métodos experimentales, los modelados analíticos y los métodos numéricos como el MEF. Estas técnicas tradicionales empezaron a ser utilizadas de forma conjunta a las RNA, facilitando los datos o patrones necesarios para poder realizar el entrenamiento de la red y su posterior validación.

Los costosos ensayos experimentales han seguido utilizándose en ciertos casos debido a la imposibilidad de realizar simulaciones numéricas o modelados analíticos alternativos por las características del problema; así como para validar los modelos numéricos diseñados. En otras ocasiones, la existencia de ensayos previos ha permitido su reutilización en conjunción con una RNA para estudiar nuevos problemas relacionados. Durante los últimos años, varios investigadores han seguido optando por esta técnica para facilitar los datos necesarios en el aprendizaje de la red [Liu et al., 2006, Wang et al., 2006, LeClerc et al., 2007].

Por otro lado, las técnicas de simulación numérica han sido empleadas extensamente, en conjunción con los sistemas de RNA, en problemas ingenieriles como los tratados en este apartado. Destacan entre ellos los relacionados con el análisis de la fiabilidad de estructuras o la detección de daños en materiales [Fang et al., 2005, Hosni et al., 2006, Jiang et al., 2006, Watkins et al., 2007]. En la actualidad, el MEF es el método de simulación numérica más empleado en el análisis, diseño y optimización estructural. Se trata de una potente herramienta computacional para transformar sistemas continuos en un sistema discreto con elementos finitos y nodos [Stasa, 1984]. Las primeras aproximaciones para aprovechar las similitudes estructurales existentes entre ambas herramientas fueron realizadas por Kudva et al. en el año 1992 [Kudva et al., 1992] y Takeuchi y Kosugi en el año 1993 [Takeuchi and Kosugi, 1993].

Durante los últimos años, la utilización de las RNA de forma combinada con el MEF ha sufrido una gran explosión en estos entornos ingenieriles ya que, gracias a las RNA, se consiguen paliar las desventajas asociadas a los métodos de simulación numérica y más concretamente al MEF. Destaca el estudio realizado por Akhavan et al. [Akhavan et al., 2000] donde se incluye una comparativa de los resultados obtenidos mediante experimentación, RNA y MEF. Por su parte, Liu et al. [Liu et al., 2002] recurren a las RNA para detección y localización de fallos en estructuras. Los datos de entrada en este caso son calculados mediante el MEF y ecuaciones integrales de frontera. Por último en los trabajos de Bisagni et al. [Bisagni et al., 2002] y Lanzi et al. [Lanzi et al., 2004] se reemplazan los costosos análisis de MEF, por herramientas de reanálisis más rápidas como las RNA. Su utilización permite una optimización preliminar para considerar los requisitos de *Crashworthiness* en las primeras fases de un proyecto.

Si bien la comunidad científica ha adoptado el MEF como herramienta general y su utilización conjunta con las RNA ha mejorado los resultados dentro del dominio del impacto de sólidos a alta velocidad, aún no han aparecido estudios relevantes donde se utilicen ambas técnicas. Esto puede ser debido al hecho de que este método de simulación numérica, por sí solo, no tiene una precisión y robustez suficiente para una descripción fiable de los materiales y para los cálculos de fractura. En este sentido, siguiendo las conclusiones obtenidas por Belytschko [Belytschko, 1996] las simulaciones que incluyen inestabilidades materiales y respuestas discontinuas están entre las que más problemas presentan para su resolución.

En la Figura 2.20 se muestran los dominios de aplicación más interesantes de las RNA, en unión con las técnicas clásicas de resolución de problemas en ingeniería industrial y estructural.

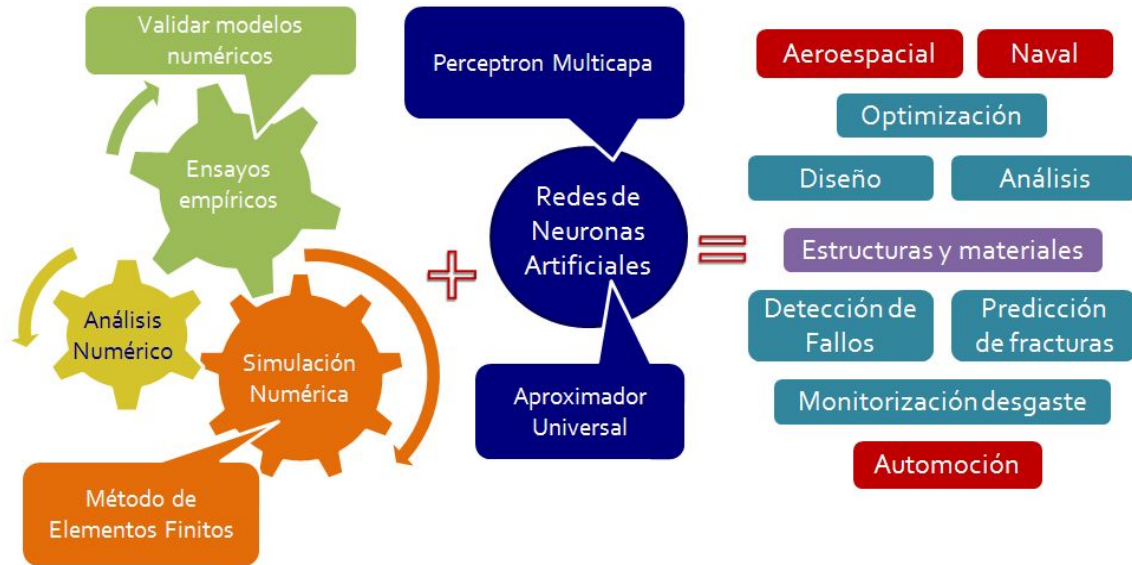


Figura 2.20: Aplicaciones de las RNA en en el dominio ingenieril

2.5.3. El problema del impacto y las Redes de Neuronas Artificiales

La reconocida capacidad de los sistemas de RNA para tareas como el reconocimiento de patrones, la clasificación y la aproximación de funciones, ha permitido que durante los últimos años su uso se haya extendido al dominio de impactos de sólidos.

Sin embargo, las publicaciones realizadas hasta la fecha se han centrado de forma principal en los impactos a baja velocidad o impactos donde es necesaria una absorción de energía, denominados *Crashworthiness*. Dejando a un lado las publicaciones del autor, hasta la fecha las únicas publicaciones referentes a la utilización de RNA para la resolución de problemas balísticos han sido las llevadas a cabo por García et al. 2006 [García-Crespo et al., 2006], así como por Fernández en forma de tesis doctoral en 2007 [Fernández, 2007]. Estos estudios han sido publicados como consecuencia de las tareas realizadas en el proyecto de investigación, financiado por la Comunidad de Madrid, GR/MAT/0507/2004. El resultado de estos trabajos ha puesto de manifiesto la viabilidad de los sistemas de RNA, como alternativa a las metodologías clásicas para la resolución de problemas de impacto de sólidos. Para más información acerca del alcance de los mismos consultar §2.9.

Por otro lado, Fernández et al. en 2008 han realizado una aproximación a la utilización de RNA en el entorno de impactos oblicuos sobre fibra de carbono reforzada con plástico (*Carbon Fiber-Reinforced Plastic* o CFRP) [Fernández et al., 2008].

Dentro del área de impactos a baja velocidad, el uso de las RNA se ha centrado en la supervisión del estado de los materiales que forman los elementos que intervienen en

el fenómeno. Las investigaciones más relevantes han tenido como objetivo determinar la gravedad de los daños producidos en el blanco [Sung et al., 2000, Jiang et al., 2006], analizar la fatiga y el comportamiento de los materiales tras el impacto [Highsmith and Keshav, 1997, Chandrashekhara et al., 1998, Wang et al., 2006] o detectar posibles roturas del material [Liu et al., 2002].

Fuera del rango de la baja velocidad, otro de los puntos de mayor interés en la actualidad, es la utilización de las RNA para la localización de impactos sobre diferentes estructuras [Jones et al., 1995, Sung et al., 2000, Worden and Staszewski, 2000]. Las investigaciones llevadas en este campo, utilizan estos conocimientos como base para realizar tanto la predicción de las posibles coordenadas del impacto, como la clasificación de la estructura donde tendrá lugar el mismo [LeClerc et al., 2007].

Por otro lado, los estudios realizados por Akhavan et al. en el 2000 [Akhavan et al., 2000] y Dua et al. en 2001 [Dua et al., 2001], demostraron la validez de las RNA para el análisis de daños, monitorización de la fatiga y detección de la laminación al producirse impactos sobre estructuras de placas compuestas. En ambos casos se utilizó un modelo de red neuronal basado en el PMC entrenado con el algoritmo RP.

Gracias a los trabajos comentados con anterioridad y a la utilización de tecnologías de detección avanzadas, se ha conseguido predecir y generalizar comportamientos y parámetros desconocidos o difíciles de calcular en sistemas físicos. Los estudios anteriores se han centrado en resolver problemas de impactos inducidos sobre placas de diversos materiales, analizando diferentes factores dentro de este fenómeno. Sin embargo existen trabajos en los que se han aplicado, también con éxito, las RNA para determinar las consecuencias de impactos en vehículos o impactos del tipo *Crashworthiness*. Las investigaciones realizadas dentro de este campo por Omar et al. [Omar et al., 1998, 2000], consiguieron recrear en una red recurrente de Hopfield las características dinámicas no lineales de la estructura del vehículo. Gracias a esto, se demostró que los parámetros del impacto podían ser analizados con un menor coste computacional mediante el uso de RNA. Los resultados verificaron que la velocidad inicial y las características estructurales del vehículo, son los factores principales que afectan a su comportamiento en caso de impacto frontal.

La utilización de RNA para la construcción de estructuras sujetas a impactos de otras estructuras dentro del rango de la baja velocidad, ha sido empleada con éxito dentro del ámbito aeroespacial. Las investigaciones llevadas a cabo por Hajela y Lee en 1997 [Hajela and Lee, 1997] y Bisagni et al. en 2002 [Bisagni et al., 2002], definen el uso de un PMC como herramienta para optimizar el coste computacional que conlleva la selección y el diseño de los materiales que formarán parte del subsuelo de helicópteros. En ambos casos, las RNA diseñadas fueron utilizadas para aproximar el comportamiento de los materiales seleccionados ante diferentes impactos. Lanzi et al. en 2004 [Lanzi et al., 2004] probaron las ventajas de las RNA como herramienta

de aproximación universal en problemas de *Crashworthiness*. Para ello se basaron en un PMC entrenado con el algoritmo RP, consiguiendo, dentro del dominio estudiado, reproducir el comportamiento de los componentes estructurales involucrados en el choque.

Para finalizar, comentar que las simulaciones realizadas mediante elementos finitos, han sido esenciales hasta la fecha para el análisis de problemas de *Crashworthiness*, del diseño de elementos estructurales y de impacto balístico. Sin embargo estos modelos son cada vez más complicados debido al gran número de parámetros que intervienen, lo que aumenta el tiempo y el coste de la simulación. En relación con esto, los trabajos realizados por Borvik et al. [Borvik et al., 2001] y Herle [Herle, 2005] demostraron las dificultades inherentes a la utilización del MEF en ciertos problemas con gran cantidad de variables y respuestas discontinuas.

2.6. Algoritmos genéticos

Los Algoritmos Genéticos (AG) son métodos adaptativos, ideados por Holland en 1975 [Holland, 1975], que pueden usarse para resolver problemas de búsqueda y optimización y están basados en el proceso genético de los organismos vivos. A lo largo de las generaciones, las poblaciones evolucionan en la naturaleza acorde con los principios de la selección natural y la supervivencia de los más fuertes, postulados por Darwin [Darwin, 1859]. Por imitación de este proceso, los AG son capaces de ir creando soluciones para problemas del mundo real. Para ello combina técnicas como la selección, cruces (*crossover*) y operadores de mutación, con el objetivo de encontrar la mejor solución a un problema. La evolución de dichas soluciones hacia valores óptimos del problema depende en buena medida de una adecuada codificación de las mismas, terminando el proceso de búsqueda o evolución cuando se cumple una condición de terminación concreta.

La solución a un problema utilizando AG se denomina cromosoma. Un cromosoma se compone de una colección de genes, que son simplemente los parámetros de la red neuronal que tienen que ser optimizados. El AG genera una población inicial, en este caso un conjunto de cromosomas, y a continuación evalúa esta población mediante la formación de una RNA para cada uno. En el siguiente paso, se desarrolla la población a través de múltiples generaciones, mediante una serie de operadores genéticos, para la búsqueda de los mejores parámetros de una RNA concreta. Gracias a este comportamiento, cabe destacar que los AG han podido usarse en el dominio de las RNA para su optimización en múltiples aspectos, destacando aquellos orientados a estimar su estructura y parámetros óptimos [Lam et al., 2001] o la selección del mejor subconjunto de variables de entrada [Yang and Honavar, 1998] para un problema concreto.

Para comenzar la competición se generan aleatoriamente una serie de cromosomas y el AG procede de la forma siguiente:

1. Evaluar la puntuación o *fitness* de cada uno de los genes.

2. Permitir a cada uno de los individuos reproducirse, de acuerdo con su puntuación.
3. Emparejar los individuos de la nueva población, haciendo que intercambien material genético, y que alguno de los bits de un gen se vea alterado debido a una mutación espontánea.

Cada uno de los pasos consiste en una actuación sobre las cadenas de bits, i.e. la aplicación de un operador a una cadena binaria. Se les denominan, por razones obvias, operadores genéticos, y hay tres principales: selección, *crossover* o reproducción y mutación.

2.6.1. Parámetros del Algoritmo Genético

Un AG tiene contiene una serie de parámetros que es necesario fijar antes de su ejecución. Los más interesantes son los siguientes:

- **Tamaño de la población:** Debe de tener un valor suficiente para garantizar la diversidad de las soluciones, y además tiene que crecer en función del número de bits del cromosoma. No obstante no existe una regla de uso general para determinar su valor óptimo. Es un punto de especial importancia para determinar la complejidad computacional del algoritmo.
- **Condición de terminación:** Lo más habitual es que la condición de terminación sea la convergencia del AG, alcanzar un número prefijado de generaciones o un límite de tiempo.
- **Probabilidad de *crossover* o cruce:** Indica la frecuencia con la que se producen cruces entre los cromosomas padre, i.e., que haya probabilidad de reproducción entre ellos. En caso de que no exista probabilidad de reproducción, los hijos serán copias exactas se los padres. En caso de haberla, los hijos tendrán partes de los cromosomas de los padres. Si la probabilidad de cruce es del 100 % el hijo se crea totalmente por cruce, no por partes. En la mayoría de las ocasiones es conveniente usar valores cercanos a uno (0.9 suele ser una primera aproximación correcta).
- **Probabilidad de mutación:** Indica la frecuencia con la que los genes de un cromosoma son mutados. Si no hay mutación, los descendientes son los mismos que había tras la reproducción. En caso de que haya mutaciones, parte del cromosoma descendiente es modificado y si la probabilidad de mutación es del 100 %, la totalidad del cromosoma se cambia. En este caso, no se cambian únicamente unos bits del cromosoma sino que se cambian todos, lo que significa que se produce una inversión en el cromosoma y no una mutación por lo que la población degenera muy rápidamente. En la mayoría de las ocasiones es conveniente usar valores pequeños (0.01 es suele ser el valor de inicio) ya que si se eligen una probabilidad demasiado elevada la búsqueda se vuelve aleatoria.

Con respecto a la convergencia del AG, existen a su vez varias alternativas para determinar si es necesario parar el algoritmo antes de agotar el número de generaciones. Entre ellas destacan las siguientes:

- Umbral de *fitness* o límite selectivo (*fitness threshold*): Se detiene la evaluación cuando el mejor *fitness* alcanzado por la población en curso es inferior a un valor umbral.
- Convergencia de *fitness*: Se detiene la evaluación cuando se considera que el valor de *fitness* ha convergido. En este caso se pueden utilizar una serie de filtros para suavizar al mejor *fitness* a lo largo de las generaciones. La convergencia aparece cuando el mejor *fitness* suavizado es menor que el porcentaje de umbral del mejor *fitness* suavizado.
- Convergencia de la población: Se detiene la evaluación cuando se considera que la población ha convergido. La población se considera como convergente, cuando el *fitness* promedio de la población actual es menor que un porcentaje de umbral del mejor *fitness* de la población actual.
- Convergencia de genes: Se detiene la evaluación cuando se considera que el porcentaje de genes que componen un cromosoma ha convergido. Esto sucede cuando el valor medio de ese gen en todos los cromosomas en la población actual es menor que el porcentaje de umbral del valor máximo de genes en los cromosomas.

2.6.2. Codificación de las variables

Los individuos, las posibles soluciones del problema, pueden representarse como un conjunto de parámetros, denominados genes, los cuales agrupados forman una ristra de valores, conocidos como cromosomas. Si bien el alfabeto utilizado para representar los individuos no debe necesariamente estar constituido por el $[0, 1]$, buena parte de la teoría en la que se fundamentan los AGs utiliza dicho alfabeto. No obstante la codificación se puede realizar de varias formas, la más utilizada es mediante la comentada de números binarios, pero también se puede realizar la codificación mediante números enteros o incluso cadenas de palabras.

2.6.3. Selección

Durante la evaluación, se decodifica el gen, convirtiéndose en una serie de parámetros de un problema, se halla la solución del problema a partir de esos parámetros, y se le da una puntuación a esa solución en función de lo cerca que esté de la mejor solución. A esta puntuación se le llama *fitness*.

El *fitness* determina siempre los cromosomas que se van a reproducir, y aquellos que se van a eliminar. Una vez evaluado el *fitness*, se tiene que crear la nueva población teniendo en cuenta que los buenos rasgos de los mejores se transmitan a esta. Para ello, hay que seleccionar a una serie de individuos encargados de esta tarea. Esta selección, y la consiguiente reproducción, se puede hacer de tres formas principales:

- Selección por rango: En este esquema se mantiene un porcentaje de la población, generalmente la mayoría, para la siguiente generación. Se coloca toda la población por orden de *fitness*, y los M menos dignos son eliminados y sustituidos por la descendencia de alguno de los M mejores con algún otro individuo de la población.
- Selección por rueda de ruleta: Se crea un conjunto genético formado por cromosomas de la generación actual, en una cantidad proporcional a su *fitness*. Si la proporción hace que un individuo domine la población, se le aplica una operación de escalado. Dentro de este conjunto, se cogen parejas aleatorias de cromosomas y se emparejan.
- Selección por torneo: Se escogen de forma aleatoria un número T de individuos de la población, y el que tiene puntuación mayor se reproduce, sustituyendo su descendencia al que tiene menor puntuación.

2.6.4. *Crossover* o cruce

Una vez se realiza la selección de los cromosomas se procede a realizar cruce o reproducción entre dos de estos cromosomas. Más concretamente, el *crossover* o cruce consiste en el intercambio de material genético entre dos cromosomas. El objetivo del cruce es conseguir que el descendiente mejore la aptitud de sus padres. El operador *crossover*, que consiste en el intercambio de material genético entre dos cromosomas o más, es el principal operador genético. El intercambio genético se puede llevar a cabo de muchas formas, pero hay dos grupos principales:

- *Crossover* n-puntos: Los dos cromosomas se cortan por n puntos, y el material genético situado entre ellos se intercambia. Lo más habitual es un *crossover* de un punto o de dos puntos.
- *Crossover* uniforme: Cada gen del descendiente se obtiene de cualquiera de los padres de forma aleatoria. Una opción es generar un número aleatorio. Si este número supera un cierto umbral se elegirá un padre determinado y si no lo supera se elige al otro.

2.6.5. Mutación

La mutación se considera un operador básico, que proporciona un pequeño elemento de aleatoriedad en la vecindad (entorno) de los individuos de la población. Si bien se admite que el operador *crossover* es el responsable de efectuar la búsqueda a lo largo del espacio

de posibles soluciones, también parece desprenderse de los experimentos efectuados por varios investigadores que el operador de mutación va ganando en importancia a medida que la población de individuos va convergiendo.

Por lo tanto, un AG consiste en determinar los parámetros de que depende el problema, codificarlos en un cromosoma, y aplicar los métodos de la evolución: selección y reproducción con intercambio de información y alteraciones que generan diversidad.

2.7. Alternativas basadas en el aprendizaje automático

El aprendizaje automático, también conocido como máquinas de aprendizaje, es una rama de la IA cuyo objetivo es desarrollar técnicas que permitan a los ordenadores aprender. Desde el punto de vista teórico el aprendizaje automático puede ser visto como un intento de automatizar algunas partes del Método Científico mediante métodos matemáticos mediante un proceso de inducción del conocimiento. De forma más concreta, se trata de crear herramientas computacionales capaces de generalizar comportamientos a partir de una información no estructurada, suministrada en forma de ejemplos.

Existen diversas formas de categorizar las alternativas de maquinas de aprendizaje, pero se suelen agrupar en función del tipo de algoritmo que utilicen, e.g. aprendizaje supervisado, no supervisado, por refuerzo, etc., y de la clase de problema a resolver, e.g. regresión, clasificación, etc. En esta investigación se han elegido las siguientes propuestas que serán evaluadas para medir su calidad frente a los diferentes PMC diseñados:

- Mapas Autoorganizados (*Self Organizing Feature Maps* o SOFMS). Es un tipo de RNA que utiliza aprendizaje no supervisado para producir una representación en bajas dimensiones de los ejemplos de entrenamiento, preservando las propiedades topológicas del espacio de entrada [Lloyd et al., 2009]. Esto hace que SOFMS sea adecuado para la visualización de datos que tienen un gran número de variables. La primera versión fue descrita por Kohonen en finales de los años 80 [Kohonen, 1988] como método para visualizar la relación existente entre patrones del habla y se suele utilizar como un método exploratorio para determinar relaciones entre los patrones. También existen numerosos ejemplos en la literatura sobre su uso como clasificador [Siripatrawan, 2008].
- Máquinas de Vectores de Soporte (*Support Vector Machines* o SVM). Se utilizan con frecuencia tanto para clasificar patrones como para resolver problemas de regresión no lineal. La idea principal asociada a un SVM es construir un hiperplano como dimensión de decisión que maximice el margen de separación entre los ejemplos positivos y negativos dentro del conjunto de datos. Este principio de inducción está basado en el hecho de que el coeficiente de error sobre el conjunto de test, i.e. el coeficiente del error de generalización, está limitado por la suma de los

coeficientes del error de entrenamiento. Este término depende de la dimensión Vapnik-Chervonenkis [Vapnik, 1998]. Un SVM facilita una generalización precisa para problema de clasificación de patrones a pesar de no incorporar un dominio de conocimiento del problema.

- Red de Función de Base Radial (*Radial Basis Function Network* o RBFN). Se trata de un tipo de RNA híbrida y no lineal que utiliza funciones de base radial como funciones de activación. Se emplean en diversos ámbitos como la aproximación de funciones, clasificación de patrones y la predicción de series temporales [Kharoufeh, 2003, Harpham and Dawson, 2006]. Su arquitectura estándar suele estar formada por tres capas: una de entrada, una oculta con una función de activación no lineal de base radial y una capa de salida lineal. En su forma básica la norma empleada es la distancia euclídea y la función base empleada es la Gaussiana. Las funciones de base radial son funciones cuya salida depende de la distancia a un punto denominado centro.
- Red de Neuronas Recurrente (*Recurrent Neural Network* o RNN). En estas redes existen dos tipos de conexiones, las de tipo *feedforward* (hacia delante) y las de tipo *feedback* (hacia atrás), que permiten que la información se propague en las dos direcciones, desde las neuronas de entrada hacia las de salida y viceversa. Las RNNs han atraído la atención de los investigadores durante los últimos años, especialmente en el campo de la predicción de series temporales, la identificación dinámica de sistemas y la clasificación temporal de patrones [Leclercq et al., 2005].
- Red Recurrente de Elman (*Elman Network* o ELN). La red recurrente de Elman es una red de retropropagación de dos capas, con la adición de una retroalimentación de la salida de la capa oculta con la entrada de la red. Esto permite que la función que aprende la red puede estar basada en las correspondientes entradas, pero a su vez en un registro de los estados previos, y las salidas de la red. Su funcionamiento basado en la retroalimentación permite detectar patrones variantes en el tiempo y la regla de aprendizaje, a diferencia de las RNN donde no siempre es posible, puede estar basada en el algoritmo RP estándar [Moghaddamnia et al., 2009].

En la literatura consultada en la mayoría de las ocasiones se hace referencia a estas máquinas de aprendizaje con su denominación en inglés, por lo que para facilitar su identificación dentro de esta investigación se seguirá dicha tendencia.

2.8. Sumario

El propósito de este capítulo ha sido, en primer lugar, acercar al lector al dominio en el que se centra la presente tesis doctoral. Como se ha podido comprobar los procesos físicos involucrados en el proceso de penetración de un material debido al impacto de un proyectil son extremadamente complejos. Además se han descrito las aproximaciones

tradicionales, utilizadas por ingenieros y científicos, para analizar y resolver los problemas de impacto balístico. Cada una de ellas ha sido pormenorizada, mostrando sus utilidades, desventajas y complicaciones ante diferentes situaciones.

A continuación se ha demostrado la utilidad de las RNA para resolver problemas ingenieriles, exponiendo las investigaciones más importantes llevadas a cabo en el ámbito de la ingeniería industrial y de la mecánica de materiales y estructuras. Por otro lado se han examinado los trabajos realizados hasta la fecha con RNA en el área del impacto de sólidos, destacando su utilización como sistemas de monitorización para la fatiga de los materiales, localización de los impactos, análisis de daños, etc. En este sentido, también se ha analizado su aplicación en uno de los entornos de más interés en la actualidad, los choques del tipo *Crashworthiness*, donde es necesario diseñar estructuras que permitan absorber la energía que se produce en este tipo de choques para salvaguardar las estructuras y ocupantes de su interior.

Los resultados obtenidos por los investigadores en estos estudios han confirmado a los sistemas de RNA como una alternativa a tener en cuenta frente a las metodologías clásicas empleadas en la resolución de este tipo de problemas: experimentación empírica, simulación numérica y modelado analítico. Por último, destacar que ha quedado demostrada la falta de investigaciones para la resolución del problema de impactos de proyectiles sobre protecciones, en el rango de la alta velocidad, mediante sistemas de RNA.

En el siguiente apartado se aclarará el ámbito y amplitud de esta tesis doctoral respecto al resto de publicaciones y líneas de investigación existentes en la actualidad.

2.9. Discusión

Como ha quedado patente, las bases de esta tesis doctoral provienen del proyecto de investigación subvencionado por la Comunidad de Madrid GR/MAT/0507/2004. Gracias al trabajo realizado en él, se publicó un artículo [Garcia-Crespo et al., 2006] y una tesis doctoral [Fernandez, 2007] para dar a conocer a la comunidad científica las conclusiones obtenidas. Dicha tesis doctoral abordaba el problema del impacto balístico desde un punto de vista diferente al de esta investigación. El objetivo principal era obtener la modelización de los materiales que componen los objetos del impacto a través de sus ecuaciones constitutivas, su simulación numérica y su posterior comparación con las RNA. Para ello se recrearon una serie de materiales, en este caso compuestos de cerámica y metal, diferentes a los utilizados en esta tesis doctoral. Para conseguirlo, fue necesario definir un nuevo modelo de daño para materiales cerámicos-metálicos, obtenido a partir del modelo de Cortes et al [Cortes et al., 1992]. A raíz de esto, se desarrolló e implementó un algoritmo de integración semi-implícito que posteriormente fue ajustado y validado mediante ensayos instrumentados de impactos sobre placas cerámicas-metal. Como último paso, se diseñó una arquitectura de red neuronal para probar la viabilidad

del uso de estas estructuras computacionales en el fenómeno del impacto balístico. Las simulaciones y experimentos realizados, tenían como meta, facilitar una primera aproximación a la resolución de este tipo de problemas con técnicas de IA como el PMC.

Por tanto, se puede considerar a esta tesis doctoral como una primera aproximación satisfactoria que permite certificar la utilidad de las RNA dentro del dominio del impacto balístico. Sin embargo, dentro de los objetivos fijados no se tuvo en cuenta la utilización de diferentes topologías, configuraciones o parámetros de red. La validación realizada no incluía un estudio destinado a la optimización de los resultados y la minimización de la información necesaria para entrenar la red, tareas fundamentales dentro del diseño ingenieril. Para cubrir este vacío y complementar ambas investigaciones, dicha tesis doctoral será tomada como línea base con el fin de poder ratificar la validez de las soluciones propuestas.

Por otro lado, el artículo realizado por Fernandez et al. [Fernandez et al., 2008] demuestra la validez de las RNA para resolver problemas, de similar dificultad física y complejidad computacional, relacionados con la teoría de estructuras y medios continuos. Este hecho refuerza la necesidad de optimizar el uso de estas estructuras computacionales en otros dominios con el objetivo de mejorar los resultados obtenidos hasta el momento.

Tomando como base lo expuesto, uno de los objetivos fijados para esta tesis doctoral es ratificar y ampliar las conclusiones obtenidas con anterioridad en los trabajos mencionados. Por un lado se definirán nuevas condiciones de contorno que permitirán extrapolar los resultados a nuevos tipos de proyectiles y protecciones. Además, con el fin de conocer si los resultados precedentes son mejorables y para reducir los tiempos asociados a la resolución de las simulaciones, se diseñaran baterías de experimentos destinadas a optimizar aquellos factores que influyen sobre la capacidad de generalización de los modelos de red.

La metodología diseñada pretende localizar aquellas características de los objetos que intervienen en un impacto balístico más importantes para el aprendizaje y posterior validación de los modelos neuronales creados. Gracias a esto, se podrán detectar aquellos factores que menos afectan al modelado del problema y por tanto son susceptibles de no formar parte de las arquitecturas de red propuestas. Esto permitirá reducir en parte, la complejidad asociada a las técnicas tradicionales de generación de impactos balísticos, al reducir el número de variables a recrear. En relación con esto, y para lograr conocer la cantidad mínima de ejemplos necesarios para mantener los resultados en clasificación y regresión, se realizarán experimentos con varias configuraciones de datos para entrenamiento, validación y validación cruzada.

Se espera que las conclusiones obtenidas a partir de esta tesis doctoral, abran nuevas posibilidades en áreas afines, entre las que cabe destacar la seguridad pasiva de vehículos. En este caso es de especial relevancia el diseño de estructuras cuya misión es absorber

energía en choques del tipo *Crashworthiness*, como los que pueden aparecer en accidentes de tráfico o ferroviarios, aterrizajes de emergencia de helicópteros, etc.

Por último, resaltar que los resultados obtenidos en el desarrollo de esta investigación han sido publicados en diversos ámbitos científicos: como una ponencia en un congreso internacional [Gonzalez-Carrasco et al., 2008], en forma de capítulo de libro [García-Crespo et al., 2008] y a través de un artículo en una revista internacional [Gonzalez-Carrasco et al., 2009].

Capítulo 3

Solución Propuesta

En este capítulo se exponen las razones que justifican la creación de un marco multidisciplinar de trabajo, precisando la posición, influencias y contribuciones de cada una de los campos que lo forman. Este intercambio y cooperación entre diversas disciplinas es una de las características de valor añadido y por tanto novedosa de la investigación, sirviendo de impulso para la misma y mejorando las aportaciones individuales de cada una de las disciplinas.

A continuación se expone el planteamiento de partida, que ha servido de motivación para esta tesis doctoral, detallando las particularidades de cada uno de los escenarios estudiados. A partir de las necesidades y de los parámetros característicos detectados, se detallan los modelos de simulación numérica utilizados para recrear el fenómeno del impacto balístico.

En el siguiente paso, y una vez generados los patrones de ensayos balísticos, se describen los modelos neuronales diseñados para cada uno de los escenarios. En ellos se detallan las decisiones de diseño y heurísticas utilizadas, obtenidas a partir del análisis previo realizado en el Capítulo 2.

3.1. Introducción

El proceso de diseño y la posterior optimización de todo proyecto ingenieril de envergadura suele involucrar la descomposición del mismo y la participación de distintas disciplinas. De esta forma, la resolución del problema del impacto balístico tratado en esta investigación requiere la colaboración de técnicas heterogéneas provenientes de diferentes ramas de la ciencia y la ingeniería. Para lograr la consecución de los objetivos marcados en §1.4 se hace necesaria la posterior asociación de las contribuciones individuales en busca de un beneficio común.

Por ello se ha creado una estructura integral de trabajo donde colaboran diferentes componentes para la resolución y optimización del problema ingenieril del impacto de sólidos. En la Figura 3.1 se muestran estos componentes así como las interacciones que se producen entre ellos dentro de la estructura global. Las herramientas de simulación numérica y la experimentación de ensayos balísticos pueden considerarse como los primeros componentes. Se trata de dos elementos cruciales ya que sirven de nexo de unión con los modelos de RNA diseñados. Esta técnica computacional de simulación permite a partir del análisis del fenómeno físico del problema, y basándose en una serie de modelos matemáticos, recrear mediante simulaciones numéricas las características de los materiales y el comportamiento de los objetos que intervienen (cf. §3.4 para más detalle). Por tanto, gracias a su empleo se obtienen una serie de ensayos balísticos simulados encaminados a entrenar y validar los modelos de RNA. Por último, las pruebas empíricas realizadas en laboratorio sirven para comprobar la verosimilitud de los modelos de simulación numérica con la realidad.

El tercer componente se corresponde con las técnicas de IA basadas en RNA (cf. §3.5 para más detalle). Su objetivo fundamental es sustituir a los componentes anteriores para en el futuro poder recrear el proceso de impacto de sólidos a alta velocidad. Estos modelos de RNA se centraran en la resolución de los dos escenarios más representativos, conocer el resultado del impacto y estudiar el comportamiento de los objetos. Si se traslada esta disyuntiva al área de la IA, y más concretamente de las RNA, su resolución implica el diseño de dos modelos diferentes, uno enfocado a la clasificación de patrones y el otro a aproximar los valores de ciertos factores característicos del problema.

El último bloque consiste en la optimización de cada uno de los modelos de RNA con el objetivo de localizar la configuración que mejor se adapte a cada escenario del problema. En este sentido se pretende mejorar no solo la precisión de los resultados, sino mejorar el rendimiento computacional y reducir la complejidad de los mismos. En el Capítulo 4 se detalla la metodología de optimización definida con este fin.

Los resultados obtenidos gracias a este marco integral, mostrados en el Capítulo 5, permitirán conocer la validez y calidad de las RNA, mostrando la potencia de esta técnica en comparación con las estrategias tradicionales, simulación numérica y experimentación, y con otras técnicas de aprendizaje automático.

3.2. Marco integral de la investigación

3.2.1. Introducción

Como se ha determinado en §1.4, el objetivo final de esta tesis doctoral es mejorar el diseño de sistemas de protección ante impactos, para el rango de la alta velocidad, a

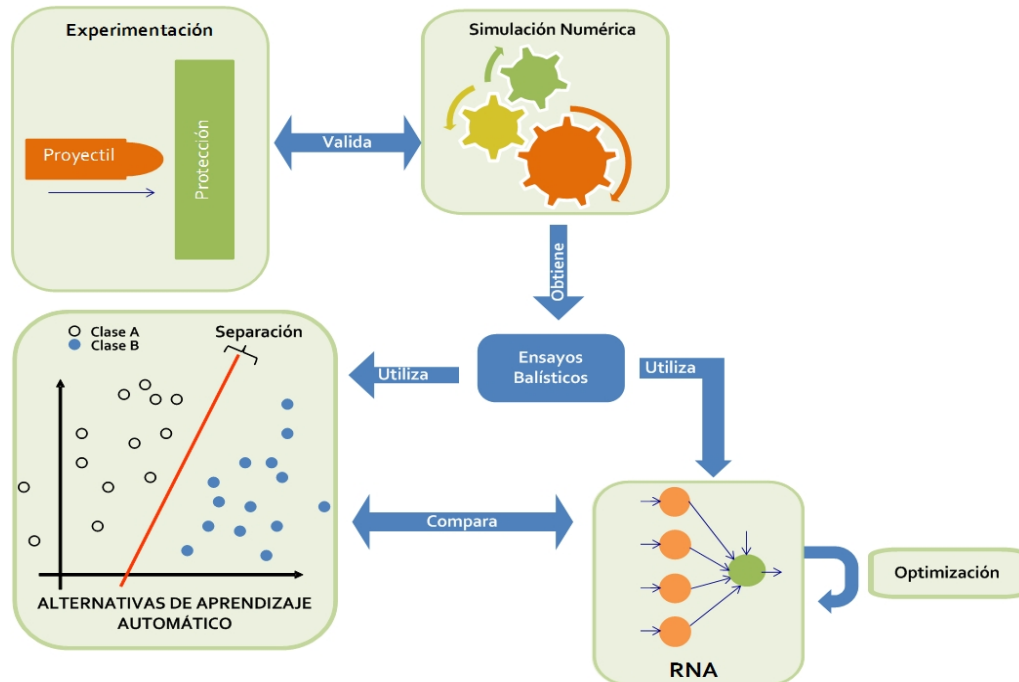


Figura 3.1: Estructura global del trabajo, donde se detallan los componentes fundamentales y las interacciones entre los mismos

través de la aplicación de una aproximación novedosa basada en IA.

El diseño ha sido una actividad humana de gran interés desde hace miles de años. La comunidad científica ha estudiado de forma exhaustiva este proceso durante las últimas décadas para conseguir el establecimiento de una serie de reglas o metodologías de uso general e independientes del dominio. En este sentido, investigadores, científicos y diseñadores han fijado, a partir de la experiencia y la observación, varias definiciones sobre el concepto de diseño. Las más interesantes han sido recopiladas por Evbuomwan et al. [Evbuomwan et al., 1996]. Una de las más extendidas para el diseño ingenieril es la facilitada por Feilden en 1963 [Feilden, 1963]: “*El diseño ingenieril consiste en el uso de principios científicos, información técnica e imaginación para la definición de estructuras o componentes mecánicos, máquinas o sistemas para realizar funciones especificadas de antemano con la máxima economía y eficiencia.*”. El artículo publicado por Finger y Dixon en 1989 [Finger and Dixon, 1989], contiene una revisión de los trabajos más importantes acerca de las metodologías y teorías de diseño, estableciendo tres categorías dentro de los modelos generales de diseño: descriptivo, preceptivo y asistido por ordenador. Más recientemente, debido a la necesidad de recurrir a múltiples disciplinas para la resolución un único problema, han aparecido nuevas estrategias de ingeniería aplicadas al diseño de sistemas complejos de gran escala, cuya solución se ve afectada por los diseños óptimos de los subsistemas que los componen. Éstas estrategias se conocen como técnicas de diseño óptimo multidisciplinario (*Multidisciplinary Design*

Optimization MDO) [Sobieszczanski-sobieski and Haftka, 1997] y se utilizan en el diseño de sistemas, en los que las fuertes interacciones entre las disciplinas, motiva a los diseñadores a manipular variables de diferentes disciplinas de forma simultánea.

En la actualidad, dentro de la comunidad científica, existe gran interés acerca del diseño asistido por ordenador. Esto se debe en parte a que durante los últimos años se ha producido un rápido e intenso desarrollo en el *hardware* de los ordenadores, lo que ha permitido que nuevas técnicas computacionales, como las basadas en IA, hayan proliferado y aprovechado esta circunstancia para crear sistemas más eficientes y robustos [Saridakis and Dentsoras, 2008]. Se ha demostrado que muchas de ellas pueden ser usadas como herramientas en problemas donde las técnicas tradicionales fallan u obtienen resultados no adecuados. Dentro de estas destacan por sus amplias posibilidades: la redes de neuronas artificiales, los sistemas expertos, los algoritmos genéticos y los sistemas basados en lógica borrosa. Un campo ideal donde demostrar esta afirmación, debido a los requisitos y características específicas de cada dominio, es en las diferentes fases del diseño ingenieril. El estudio de las publicaciones existentes ha confirmado este punto, ya que en los últimos años la cantidad de investigaciones dentro de este ámbito que utilizan dichas técnicas ha crecido de forma notable.

Por otro lado, los sistemas típicos en ingeniería están formados por un gran número de variables y parámetros, siendo tarea del diseñador especificar los valores apropiados para ellos. Los diseñadores experimentados utilizan su conocimiento, experiencia y criterio para ajustar los valores y diseñar sistemas ingenieriles efectivos. No obstante, debido al tamaño y complejidad de las tareas de diseño típicas, incluso el diseñador más experimentado es incapaz de tener en cuenta todas las variables y factores simultáneamente. La optimización de estos sistemas ingenieriles consiste en la aplicación de algoritmos numéricos y técnicas que asisten a los diseñadores con el objetivo de mejorar el rendimiento, fiabilidad y/o coste.

En el caso de protecciones balísticas, debido a la amplitud y complejidad del proceso de diseño, ingenieros y diseñadores han recurrido a diferentes técnicas computacionales para analizar y comprender el comportamiento de los objetos que intervienen. Una aproximación novedosa como la que se propone en este trabajo, donde se incluyen técnicas de IA basadas en el aprendizaje automático, necesita también de un proceso de optimización para localizar dentro del conjunto de soluciones la que mejor se adapta al problema. La selección de la mejor solución o solución óptima para la resolución de un problema dado es una de las mayores preocupaciones para los ingenieros. Durante los últimos años, el interés por la aplicación de técnicas de optimización matemáticas en el diseño ingenieril ha ido en aumento. Los algoritmos de optimización ya no son considerados herramientas esotéricas usadas para la resolución únicamente de problemas teóricos. Éstos pueden utilizarse de hecho como efectivas herramientas para el diseño sistemas ingenieriles e industriales de gran complejidad y envergadura [Adeli, 1994].

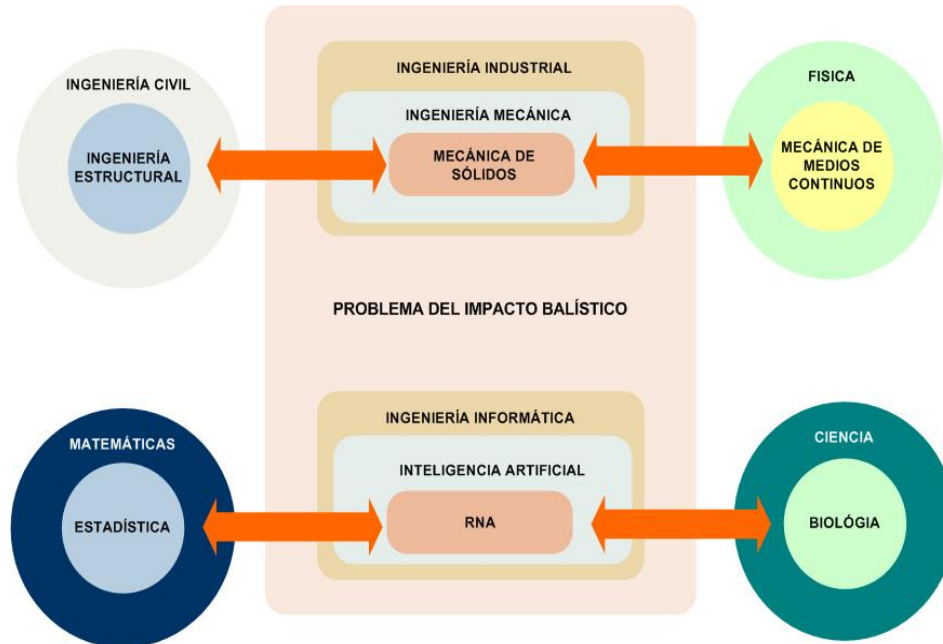


Figura 3.2: Disciplinas implicadas en esta tesis doctoral

Estas premisas han sido tenidas en cuenta en esta investigación, localizando, mediante el empleo de modelos neuronales y su posterior evaluación, optimización y su comparación con otras técnicas, la mejor solución posible para el problema del impacto de sólidos.

3.2.2. Ubicación de la investigación

Cada día es más común que diversas ramas ingenieriles colaboren para la resolución de problemas de elevada complejidad o donde no se terminan de conocer las leyes que los gobiernan. El problema tratado en esta investigación, el impacto balístico de sólidos, se instala dentro del ámbito de la mecánica de sólidos, donde se analiza el comportamiento de las fuerzas sobre sólidos tanto en equilibrio como en movimiento. No obstante, debido a la complejidad de los mecanismos físicos de este fenómeno, es necesario conocer el comportamiento de diferentes estructuras, este problema presenta ramificaciones con otras ingenierías y disciplinas de la ciencia (cf. Figura 3.2).

Desde el punto de vista ingenieril, la mecánica de sólidos tiene relaciones que llegan hasta la ingeniería estructural, rama clásica de la ingeniería civil, que se ocupa del diseño y cálculo de la parte estructural de elementos mecánicos. A su vez, enlaza también con la mecánica de medios continuos, rama de la física, que se encarga de investigar las propiedades de sólidos deformables, sólidos rígidos y fluidos.

Como se ha comentado en §2.1.5, las metodologías clásicas para modelar este fenómeno

encuentran cada vez más problemas para lograr soluciones eficaces en un periodo de tiempo razonable. Debido a esto, era necesaria la aportación de otras técnicas ingenieriles que colaboraran en su resolución. En este caso las técnicas de IA, pertenecientes a la rama informática de la ingeniería, permiten definir sistemas computacionales que se aplican a problemas en los que las reglas que lo rigen no son conocidas o presentan una extrema dificultad. Dentro de éstas, se ha demostrado en §2.5 que las RNA son una alternativa de amplias posibilidades para el diseño de protecciones sujetas a impacto.

La teoría neuronal se integra en los campos de los sistemas dinámicos, control adaptativo y estadística. Inspirada en los sistemas nerviosos biológicos, la característica más destacada de las RNA es su capacidad para aprender de ejemplos. Mediante los llamados algoritmos de aprendizaje, pueden resolver problemas tras el procesamiento de los conjuntos de datos de entrenamiento. Sin embargo, a diferencia de los modelos estadísticos de regresión, estas redes no presuponen una relación concreta entre las variables de entrada y salida, i.e., no necesitan de un modelo matemático previo, por lo que constituyen estimadores independientes del modelo. Sin embargo, para ciertas aplicaciones, hay una característica de las RNA muy inconveniente. Dada su naturaleza conexionista distribuida y no simbólica, la solución aprendida por una RNA no puede expresarse de una manera explícita. La red aprende, pero el usuario de la red no puede aprender de ella.

Las estructuras computacionales del tipo RNA presentan características muy interesantes para la resolución del problema del impacto de sólidos. Éstas pueden aprender a partir de experiencias anteriores, abstraer patrones esenciales de los datos y generalizar respuestas ante nuevos patrones de información. Además su coste computacional, una vez comenzado el entrenamiento, es menor que en las simulaciones hechas con herramientas de elementos finitos.

Para poder realizar estas tareas de forma satisfactoria, las técnicas de aprendizaje automático como las RNA, necesitan ejemplos de los que obtener información, en este caso ensayos de impactos balísticos. En la mayoría de las ocasiones, las redes neuronales obtienen mejores resultados cuando tienen un número elevado de ejemplos de los que extraer el conocimiento del problema. Sin embargo, en el dominio de esta investigación, la obtención de estos ensayos se consigue de manera habitual mediante arduas y lentas técnicas de simulación numérica. Con éstas, es necesario conseguir la solución completa de las ecuaciones diferenciales, de la mecánica de medios continuos, que aparecen durante el proceso de impacto balístico. En la actualidad, son pocos los programas comerciales capaces de realizar aproximaciones fiables, necesitando además una gran cantidad de tiempo, horas o incluso días, para realizar una simulación de un solo ensayo. Por ello, la cantidad de ejemplos disponibles para resolver este tipo de problemas mediante RNA no suele ser demasiado elevado, siendo un punto de especial interés conseguir modelos de red que funcionen bien con poca información.

Como consecuencia de los escasos ejemplos disponibles, al dividirlos en los conjuntos necesarios para realizar el entrenamiento y posterior validación de la red, es posible que los resultados obtenidos presenten cierta dependencia de los datos utilizados. Debido a esto, es de gran interés conocer la cantidad mínima de datos con la que la RNA es capaz de generalizar correctamente. Por otro lado, para evaluar si los modelos diseñados son extrapolables a nuevas condiciones de entorno o incluso otros dominios ingenieriles, es conveniente detectar las variables de entrada más relevantes y descartar aquellas que no aporten información. Por último, es interesante identificar y optimizar aquellos parámetros, tanto de la arquitectura de la red como del algoritmo de entrenamiento utilizado, que pueden influir sobre los resultados que obtiene la RNA.

La combinación de los puntos anteriores, donde aparecen técnicas de diferentes disciplinas, permite la resolución del problema del impacto balístico desde otra perspectiva: la utilización de herramientas computacionales basadas en IA. Además, gracias a los puntos citados, se consigue orientar la investigación hacia el perfeccionamiento tanto del algoritmo de aprendizaje como la arquitectura diseñada, con el objetivo de optimizar los modelos de red y alcanzar un nivel óptimo de precisión y minimizar el tiempo de respuesta.

Por tanto, para la resolución del problema del impacto balístico es necesaria definir una serie de tareas de claro enfoque multidisciplinar que aúnen las ventajas inherentes a cada disciplina. Esta tesis doctoral ha seguido esta tendencia incorporando en su praxis métodos y técnicas de estas disciplinas.

3.3. Planteamiento del problema

El impacto de un proyectil sobre un panel protector o blindaje puede originar dos resultados básicos desde el punto de vista puramente físico: puede traspasarla completamente o quedarse incrustado. En el primer caso, denominado perforación, el proyectil al atravesar la protección sale con una velocidad residual y masa final diferentes a las que tenía antes de comenzar el impacto. Cuando el proyectil entra en la protección, pero no consigue traspasarla, se habla de detención. En este caso, el proyectil queda parado dentro de la estructura a una profundidad concreta, conocida como *DOP* o profundidad de penetración. Por otra parte, el proceso de impacto, desde el momento que el proyectil comienza la entrada en la protección hasta que perfora o se detiene se conoce como penetración. En la Figura 3.3 se representa la secuencia completa del proceso de impacto, desde el momento previo a la penetración hasta la posible perforación (caso c) o detención (caso d).

Como se ha comentado en §2.1.5 y §3.2.2, la resolución tradicional de este tipo de problemas ha sido realizada en la mayoría de los casos mediante técnicas de simulación

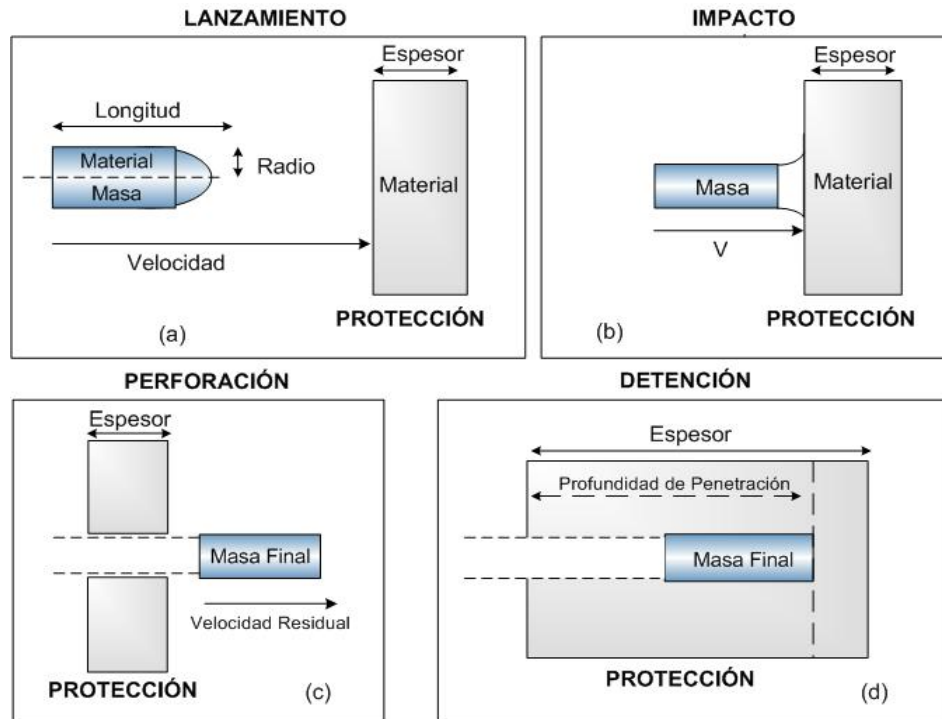


Figura 3.3: Esquema del impacto del proyectil sobre la protección

numérica. La simulación mediante códigos numéricos del impacto entre sólidos en el rango de la alta velocidad ($v > 500$ m/s), constituye uno de los problemas de diseño de elementos estructurales que mayor coste computacional presenta [Anderson, 1987]. Confluyen en éstos numerosas causas de no-linealidad, tales como comportamiento plástico del material, grandes deformaciones y contactos, que gravan considerablemente los tiempos de cálculo. Así mismo, los elevados gradientes que aparecen en las variables obligan a emplear mallas muy refinadas y la gran distorsión de elementos precisan habitualmente de un remallado. Todo ello hace que las simulaciones numéricas requieran de herramientas de diseño extremadamente sofisticadas (no más de tres ó cuatro códigos comerciales son capaces de abordar este tipo de problemas) y computadoras de gran capacidad de cálculo. Así pues, la realización de un elevado número de simulaciones tendentes a optimizar parámetros geométricos o de material, práctica habitual durante el diseño de protecciones frente a impacto balístico, obliga a destinar numerosos recursos. Mientras que un componente estructural convencional suele ser diseñado para que sea capaz de soportar las cargas sin que se produzca plastificación en ninguno de sus puntos, los sistemas de protección frente a impacto deben disipar energía del proyectil mediante grandes deformaciones plásticas, fuente principal de las no-linealidades comentadas anteriormente y del elevado coste computacional que conlleva su simulación [Zaera and Sanchez-Galvez, 1998, Zaera, 2001].

En un intento de proponer herramientas de diseño alternativas para este tipo de sistemas, se pretende desarrollar un sistema para la predicción de la respuesta de

protecciones frente a impacto de alta velocidad, basado en la utilización de RNA. Las RNA han sido durante los últimos años un centro de intensa actividad investigadora, dando lugar a una maduración de las primitivas ideas desarrolladas en los años 60 [Minsky and Papert, 1969]. Su principal ventaja es la adaptabilidad dinámica, i.e. su capacidad para variar de comportamiento en situaciones cambiantes [Ellias and Grossberg, 1975].

Para conseguirlo se utilizan técnicas como el aprendizaje, generalización o auto-organización; por consiguiente, se inspiran en el modelo de neurona biológica, usando unidades elementales de procesamiento que mimetizan algunas de las características de las neuronas biológicas. El comportamiento global de una red determina su capacidad para ensayar hipótesis, detectar patrones estadísticos y regularidades o ajustar dinámicamente un modelo implícito implementado en la misma arquitectura [Grossberg, 1988]. Este comportamiento va más allá de la suma de las potencialidades de las neuronas que la componen. En estos sistemas la información no se almacena en un emplazamiento físico único. Por el contrario, la información aparece distribuida por toda la estructura de la red, concentrándose en las uniones de los distintos elementos [Pao, 1989]. Ésta es la base del procesamiento distribuido: las funciones proporcionadas en estos sistemas, vienen dadas por la globalidad del sistema más que por la aportación individual de cada elemento. El resultado es la aparición de propiedades nuevas que pertenecen al sistema como un todo. La segunda cualidad de una RNA es su capacidad de aprendizaje, que permite a la red modificar su propia estructura (matriz de pesos) adaptándola hasta conseguir un algoritmo de ejecución. Este algoritmo se construye a partir de las características extraídas de los ejemplos, con los que se realiza el entrenamiento.

Aunque la aparición de las RNA ha derivado de ideas ya antiguas y maduras durante décadas, el interés que ha despertado durante los últimos años ha sido uno de los más rápidos en la historia de la computación. Actualmente se emplean para la resolución de un número creciente de problemas científicos y técnicos, aunque quedan todavía numerosas aplicaciones en las que no se ha utilizado toda su potencialidad. Presentan un especial interés, aquellos problemas que cuentan con un elevado número de variables, una respuesta altamente no lineal y experimentan simulaciones complejas, ya sea por ausencia de modelos o por el elevado coste computacional asociado.

El procedimiento que se emplea normalmente en las fases preliminares de modelado de esta clase de problemas, se basa en la utilización de los resultados de simulación de problemas específicos, obtenidos mediante códigos comerciales que resuelven el problema numéricamente mediante el método de los elementos finitos, tales como LS-DYNA¹, PAM-CRASH², ABAQUS³, los cuáles son previamente validados con ensayos estáticos de configuraciones simples. La razón para utilizar ensayos estáticos para contrastar los resultados numéricos es, fundamentalmente la gran complejidad tanto en equipamiento como en interpretación de resultados, que supone la realización de ensayos dinámicos

¹<http://www.ls-dyna.com/>

²<http://www.esi-group.com/SimulationSoftware/NumericalSimulation/index.html>

³http://www.simulia.com/products/abaqus_explicit.html

más realistas. Por otra parte, los trabajos recogidos en la literatura científica ponen de manifiesto la capacidad de los códigos numéricos antes citados para predecir la influencia de los efectos inerciales [Langseth et al., 1999]. Por todo ello, queda clara la dificultad de obtener una elevada cantidad de ensayos balísticos con los que poder entrenar los modelos neuronales. No obstante, dichos ensayos balísticos simulados numéricamente, son fundamentales para poder estudiar, desde la perspectiva de las RNA, las situaciones comentadas en la Figura 3.3. Con el fin de obtener conclusiones de interés, los modelos de red serán validados con diversas geometrías de proyectil y de panel protector, así como distintas velocidades y materiales.

La principal ventaja de los modelos de RNA, frente a los códigos de elementos finitos o diferencias finitas, es su gran capacidad predictiva unida a su rápida respuesta. La simulación de un problema de impacto puede conllevar varias horas de ejecución, mientras que las RNA, una vez realizado el proceso de aprendizaje, obtienen el resultado de forma prácticamente inmediata. A partir de estos modelos neuronales, se definirán una serie de pasos necesarios para mejorar la capacidad de generalización y optimizar los resultados obtenidos durante la fase de validación, tanto en tiempo de cómputo como en precisión.

A continuación se detallan los escenarios de impacto que serán estudiados a lo largo de esta tesis doctoral.

3.3.1. Resultado del impacto

El primero de los escenarios planteados se encarga de estudiar el resultado de un impacto, centrándose en conocer si el proyectil será capaz de perforar o no la protección. La importancia de este escenario radica en que una protección para uso balístico se diseña generalmente con el objetivo de conseguir detener el proyectil cuando recibe el impacto de este. Por tanto, desde el punto de vista de la protección, un aspecto prioritario a la hora de diseñarla es determinar el espesor mínimo necesario para detener un proyectil. De igual manera, el diseño de un proyectil tiene como objetivo fundamental conseguir perforar una protección determinada en caso de impactar contra ella. Puesto que las RNA permiten realizar simulaciones en poco tiempo una vez finalizado su entrenamiento, la ventaja de estas herramientas es que pueden realizar pruebas con diferentes configuraciones de panel protector y proyectil sin consumir elevados recursos.

Desde el enfoque de las RNA, este escenario corresponde con un problema de clasificación ya que la red catalogará cada patrón de entrada, en este caso nuevos ensayos balísticos no utilizados durante el entrenamiento, entre las dos clases posibles, perforación o detención. En este tipo de problemas la salida calculada por la red se interpreta como la probabilidad de pertenencia a un determinado grupo. En función del porcentaje de acierto que presenta la red a la hora de clasificar entre las posibles clases, se determina durante la fase de validación del aprendizaje la calidad del modelo neuronal.

3.3.2. Comportamiento del proyectil y del panel protector

El segundo de los escenarios se dedica a conocer el comportamiento de los objetos una vez comenzado el proceso de impacto. En este caso los aspectos de interés para el diseño, tanto de proyectiles como de protecciones, son la velocidad residual y la masa final de los primeros.

La velocidad residual se define como la velocidad de salida que tiene un proyectil después de perforar completamente una protección. Su estudio permite conocer el comportamiento de diferentes configuraciones de proyectiles sobre una protección. Por su parte, la masa final se define como la masa que presenta el proyectil al perforar completamente una protección. También suele expresarse como el porcentaje de masa que ha perdido el proyectil debido a la abrasión durante el proceso de impacto. De esta forma, conocer la masa final que presenta un proyectil después de perforar una determinada protección permitirá estudiar la viabilidad de ambos diseños para un impacto balístico.

Los dos casos anteriores de este escenario se corresponden con una perforación por parte de un proyectil sobre una protección. En el supuesto de producirse una detención, el punto de interés es conocer con que profundidad ha penetrado el proyectil dentro de ésta. Este es un factor de interés en el diseño de protecciones, ya que permite conocer la combinación de material y espesor mínimo que permite detener un determinado proyectil.

Por tanto para la resolución de estas situaciones, se desarrollarán una serie de modelos de RNA destinados a resolver cada uno un problema de regresión mediante la predicción de los valores de salida requeridos en cada caso. En el primero, se calcula la velocidad residual y la masa final del proyectil cuando este perfora la protección y consigue por tanto perforarla. En el segundo, la red obtendrá únicamente la masa final ya que la velocidad residual es cero.

En regresión, la red aproxima a partir de un patrón de entrada y mediante una función continua no lineal (F) el patrón de salida. En este caso para conocer la calidad del modelo neuronal durante la fase de validación, se calcula y compara para cada variable de salida, el coeficiente de correlación lineal (r) que existe entre los valores reales de esa variable y los obtenidos por la RNA. Este coeficiente permite conocer el grado de similitud y de exactitud de la respuesta de la red después del entrenamiento realizado.

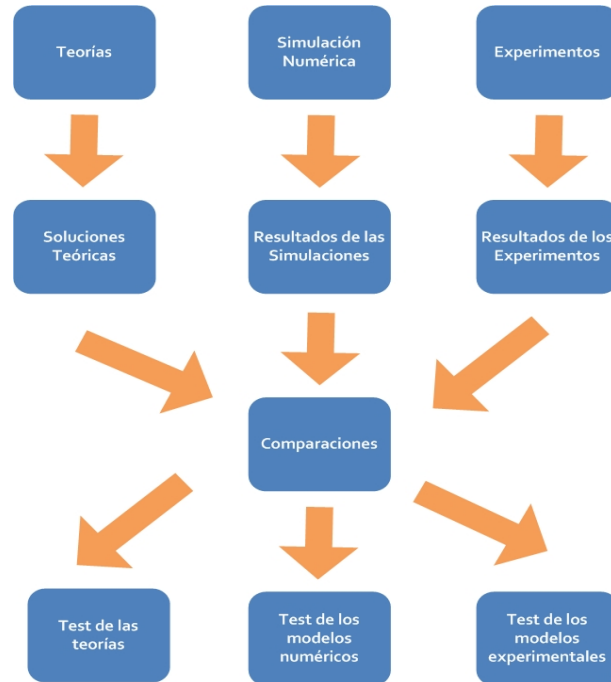


Figura 3.4: Relación entre simulaciones numéricas, teorías y experimentos

3.4. Simulación Numérica

3.4.1. Introducción

La simulación numérica mediante ordenadores, también conocida como simulación computacional, se ha convertido hoy en día en una aproximación muy importante para resolver problemas prácticos y complejos dentro del dominio de la ingeniería y la ciencia. La simulación numérica traslada los aspectos físicos más importantes del problema a resolver a una forma discreta mediante la utilización de descripciones matemáticas.

Dicho problema, por tanto, es recreado y resuelto en un ordenador lo que permite mostrar el fenómeno virtualmente de acuerdo con los requisitos de los analistas. Su utilización es preferible en lugar de adoptar las prácticas teóricas tradicionales que se basan en aproximaciones o suposiciones. La principal ventaja de estas aproximaciones numéricas es que afrontan los problemas originales en todo su detalle, sin tener que realizar demasiadas suposiciones y con la ayuda de la potencia computacional asociada a los ordenadores.

La simulación numérica provee una herramienta alternativa para la ardua tarea de investigación científica, en lugar de tener que llevar a cabo experimentos costosos, de larga duración o incluso peligrosos en laboratorios o en campo. Las herramientas de simulación numérica son a menudo más útiles que los métodos tradicionales de experimentación,

ya que facilitan información intuitiva y completa que no puede ser medida u observada directamente o es difícil de adquirir por otros medios.

Por tanto, la simulación numérica mediante ordenadores juega un valioso papel para facilitar la validación de teorías, ofrecer una mejor comprensión de los resultados y proporcionar la interpretación o incluso el descubrimiento de nuevos fenómenos. Al mismo tiempo, dentro de las investigaciones científicas en las que se utiliza esta técnica, las simulaciones numéricas están tomando progresivamente más importancia como puente de unión entre los modelos experimentales y las predicciones teóricas [Liu and Liu, 2003]. En la Figura 3.4 se pueden observar estas relaciones.

En principio hay una serie de pasos que deben ser cubiertos antes de poder utilizar estas técnicas dentro de un dominio. Como se muestra en la Figura 3.5, a partir del fenómeno físico observado, es necesario establecer una serie de modelos matemáticos a partir de simplificaciones y suposiciones.

Esto se debe a que en la búsqueda de una descripción cualitativa de un determinado fenómeno físico, por lo general el ingeniero plantea un sistema de ecuaciones diferenciales ordinarias o parciales, válidas para determinada región (o dominio), e impone sobre dicho sistema condiciones de borde/contorno e iniciales apropiadas. En esta etapa, el modelo matemático está completo, y es aquí donde aparece la mayor dificultad, dado que solamente la forma más simple de ecuaciones, con fronteras geoméricamente triviales, es capaz de ser resuelta en forma exacta con los métodos matemáticos disponibles. Las ecuaciones diferenciales ordinarias con coeficientes constantes son uno de los pocos ejemplos para los cuales se dispone de procedimientos matemáticos clásicos de solución.

Con el fin de evitar tales dificultades y lograr resolver el problema con la ayuda de computadoras, es necesario presentar el problema de una manera puramente algebraica. Mediante el proceso de discretización, el conjunto infinito de números que representan la función o funciones incógnitas en el continuo es reemplazado por un número finito de parámetros incógnita, y este proceso requiere alguna forma de aproximación. Es decir para resolver numéricamente estas ecuaciones que gobiernan un fenómeno, es necesario dividir en componentes discretos el dominio del problema. Esta discretización del dominio se refiere en la mayoría de los casos a representar el problema de forma continua con un número finito de componentes, que forman el marco computacional para la aproximación numérica.

A continuación se codifica e implementan los algoritmos numéricos en algún lenguaje de programación teniendo en cuenta dos factores determinantes: eficiencia y precisión. Finalmente y una vez comprobado el correcto funcionamiento de los algoritmos, para ello se pueden contrastar los resultados frente a técnicas de *benchmarking*⁴, se realizará la

⁴Técnica utilizada para medir el rendimiento de un sistema o componente de un sistema

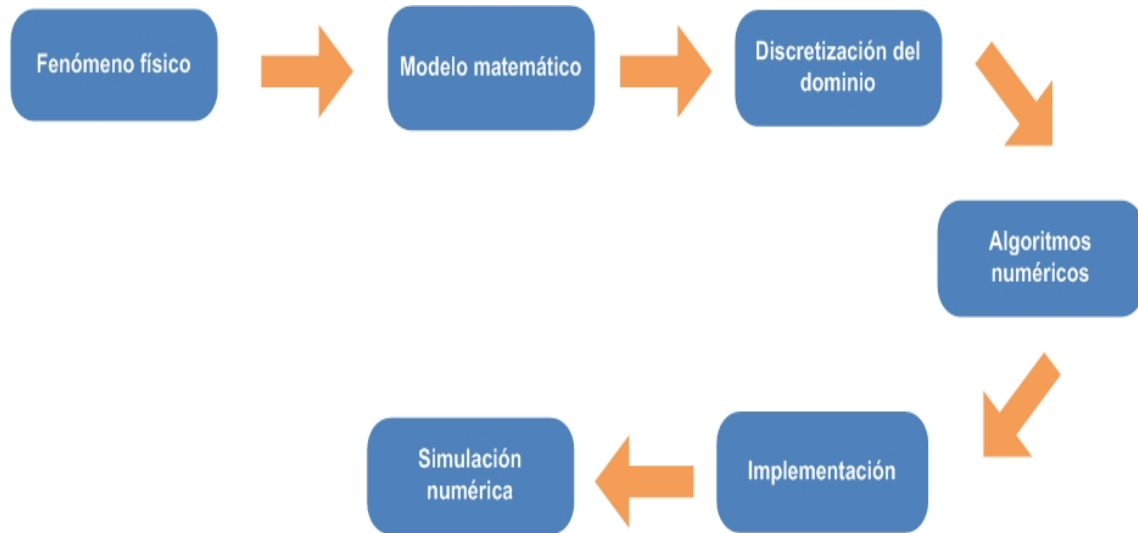


Figura 3.5: Procedimiento para realizar una simulación numérica

simulación numérica.

3.4.2. Justificación de la necesidad de la simulación numérica

Para obtener los resultados para cada caso del conjunto de entrenamiento de la RNA se podría haber optado por una experimentación clásica, i.e. realizar unos cientos de pruebas en el laboratorio para posteriormente medir los resultados. Esto puede ser muy costoso, tanto económicamente como en horas invertidas en el procedimiento, debido a la complejidad de las pruebas a realizar. Sin embargo, es posible simular los casos numéricamente utilizando un computador para obtener resultados fiables con menor esfuerzo. Para ello, la simulación numérica permite obtener la solución del problema del impacto mediante la resolución de las ecuaciones de termodinámica de medios continuos. Sus ventajas frente a las técnicas clásicas han quedado patentes en §2.1.5. En contraposición, los códigos de simulación comerciales no suelen contar con modelos avanzados que recojan adecuadamente el comportamiento frente a impactos de materiales avanzados tales como las cerámicas de altas prestaciones [Fernandez, 2007]. Además, para los modelos existentes, resulta muy difícil encontrar en la literatura científica los valores de sus parámetros para los diferentes materiales empleados en las protecciones, lo que obliga a recurrir a costosas campañas de caracterización mecánica. Por otra parte, en aquellas ocasiones en las que se requiere rapidez en la obtención de resultados, el elevado tiempo de cálculo requerido para realizar la simulación supone un inconveniente. Como consecuencia de esto, la utilización de estas técnicas para resolver futuros ensayos balísticos queda descartada, siendo necesario buscar otras alternativas más eficientes.

En este sentido, como ya se comentó en §2.1.5, la otra alternativa más utilizada

es la modelización analítica. Esta permite resolver el problema, bajo la aplicación de fuertes hipótesis simplificadoras en los mecanismos físicos que intervienen, en tiempos muy reducidos (pocos segundos). Esto unido al bajo coste de esta metodología, supone una gran ventaja frente a los métodos anteriores de experimentación y simulación. Como principal inconveniente, se tiene la menor fiabilidad de los resultados obtenidos y cierta limitación en su capacidad predictiva, determinada por el rango de validez de las hipótesis simplificadoras.

Por tanto, debido a las características especiales del fenómeno balístico, donde es necesaria una respuesta de gran fiabilidad de los objetos, es preferible obtener aproximaciones precisas aunque el tiempo de computación sea más dilatado. Por ello se ha decidido utilizar las técnicas de simulación numérica para obtener un conjunto de ensayos balísticos que van a servir para aprovechar las cualidades ofrecidas por las RNA como herramientas de aprendizaje automático. Gracias a estos ensayos balísticos se realizará el entrenamiento y posterior validación de los modelos de red. El objetivo es conseguir extraer de estos ensayos simulados numéricamente, la información necesaria para que las RNA puedan resolver nuevos casos, sin la necesidad de recurrir a las técnicas de simulación numérica o de experimentación clásica.

A modo de resumen, la Figura 3.6 permite observar las relaciones existentes entre las herramientas de simulación numérica, la experimentación realizada en laboratorio y las heurísticas de aprendizaje automático introducidas en el marco de la investigación.

Dentro de las características de un programa de simulación numérica, éste se divide normalmente en tres partes: el preprocesador, el procesador y el postprocesador. Durante el preproceso se genera la geometría del problema, se definen velocidades iniciales, materiales, temperaturas y el resto de variables que influyen en el problema. Para que el procesador pueda trabajar necesita un conjunto de elementos, cada uno formado por un número determinado de nodos, con sus coordenadas espaciales y propiedades asociadas. Éstos se pueden generar de manera automática o, si es necesario, manualmente.

Posteriormente en el procesado existe la posibilidad de usar diferentes alternativas, como el Método de Elementos Finitos (MEF) o bien el de Diferencias Finitas (MDF), para resolver las distintas ecuaciones para cada instante de tiempo. Dentro de estas destacan las ecuaciones de conservación de masa, momento y energía junto con las relaciones constitutivas, la ecuación de estado para determinación de presiones, criterios de fallo y modelos de post-fallo, e.g. el modelo de Johnson-Cook [Johnson and Cook, 1983]. En problemas estacionarios el proceso finaliza en este punto, sin embargo en problemas dinámicos son necesarios varios pasos adicionales para integrar en el tiempo las ecuaciones.

Una vez procesado se obtiene una salida con todos los nodos del problema, con sus correspondientes velocidades, posiciones, temperaturas, presiones, etc. Para trabajar con ellos y extraer los datos necesarios se recurre a las herramientas de postprocesado, que

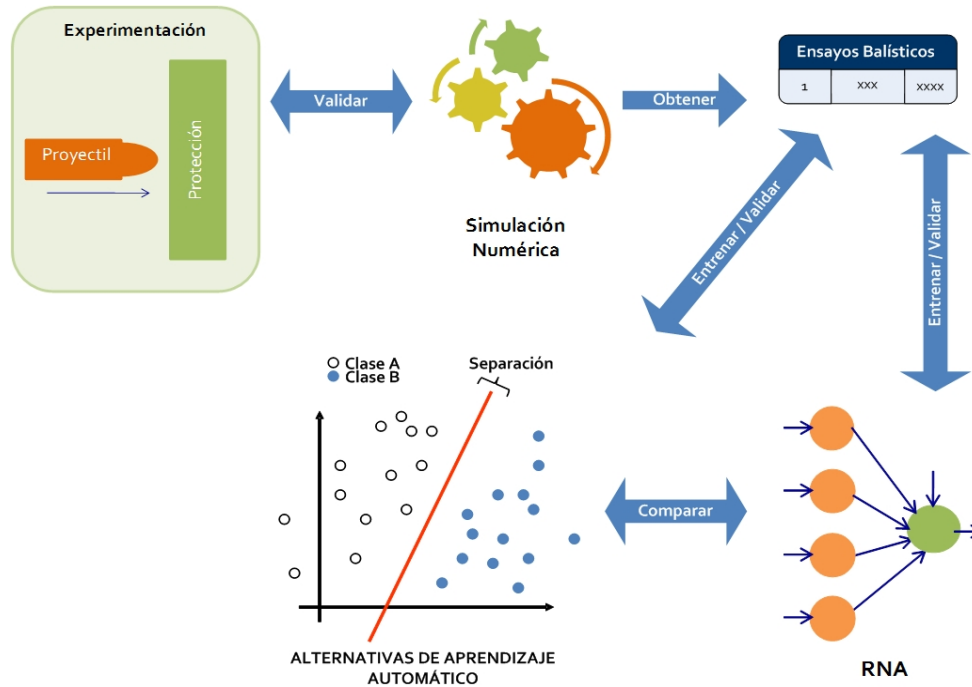


Figura 3.6: Relaciones entre la simulación numérica, experimentación y RNA

permiten obtener de una manera visual la información más interesante para el problema dado.

Es importante resaltar que para obtener resultados fiables es preciso un conocimiento previo de las variables que influyen en el problema, pudiendo de esta forma distinguir las de las secundarias y no perder información debido a las simplificaciones que se realicen. Por último, es recomendable efectuar pruebas comparativas con otro método de simulación, ya sea teórico o un experimento de laboratorio, para validar los datos obtenidos.

3.4.3. Ensayos experimentales

Como se ha comentado en §2.1.5, desde sus comienzos el estudio del fenómeno del impacto de proyectiles se realizó de forma empírica, recurriendo al uso de complejos ensayos experimentales. Hoy en día, el diseño de protecciones metálicas sujetas a impacto, sigue recurriendo a dicho método empírico como herramienta fundamental, ya que rara vez se prescinde de ella para verificar el nivel de protección alcanzado.

Uno de los primeros pasos necesarios para desarrollar estas pruebas es la fabricación de las protecciones metálicas. Para ello se han utilizado placas o losetas con diferentes compuestos metálicos, destinadas a ser impactadas bajo distintas configuraciones de

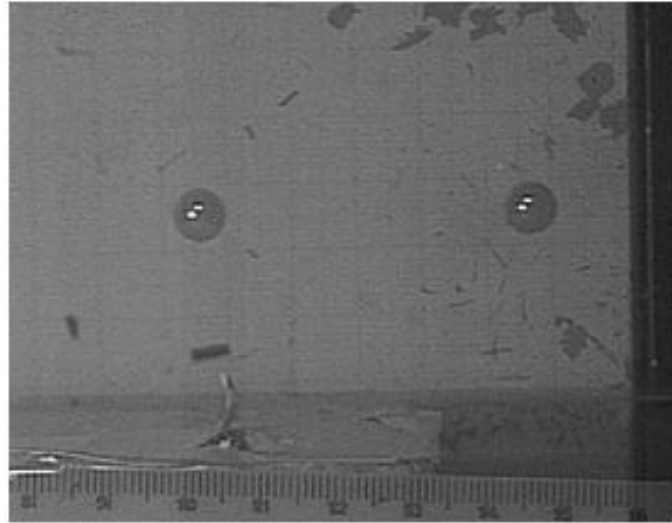


Figura 3.7: Dos imágenes superpuestas de un proyectil esférico tras perforar un panel de material compuesto con velocidad de salida de 480 m/s

velocidad y proyectil.

Mediante un sistema de cámaras digitales de alta velocidad se puede capturar el proyectil en vuelo tras la perforación del blanco y medir, a partir de estas imágenes, su velocidad residual. Todo ello ha exigido diseñar la electrónica de sincronización del sistema, desarrollar elementos de detección del paso del proyectil que sirvan de trigger, y optimizar la óptica de las cámaras para captar el proyectil con la suficiente definición. Ha sido necesario disponer dos cámaras perpendicularmente a la trayectoria por si se producen impactos oblicuos y si el proyectil se desvía una vez perforado el blanco. Este hecho puede producir errores en la interpretación de las medidas de distancias en las fotografías obtenidas, errores que se corrigen con la utilización de dos cámaras. En las Figuras 3.7 y 3.8 pueden observarse el sistema instalado y una imagen con dos proyectiles superpuestos respectivamente.

Otro aspecto de interés ha sido el diseño, construcción y puesta a punto del dispositivo para la sujeción de las protecciones en diferentes orientaciones. Se ha diseñado y fabricado un soporte buscando la máxima versatilidad, de forma que el útil permite sujetar placas de distintos tamaños y espesores. Así mismo, el soporte permite orientar la probeta formando distintos con ángulos con la trayectoria del proyectil. En la Figura 3.9 puede observarse el soporte montado dentro de la cámara principal del cañón de gas.

Los equipos utilizados para la validación de la herramienta de simulación, mediante la realización de los ensayos de compresión de elementos tubulares, fueron los siguientes:

- Máquina universal de ensayo hidráulica INSTRON de 250 kilonewtons (kN).



Figura 3.8: Sistemas de cámaras digitales ultrarápidas

- Equipos de extensometría resistiva.
- Cámara digital de alta velocidad para filmación de ensayos.
- Software de análisis de imagen para determinación de desplazamientos.

3.4.4. Simulación numérica de los patrones de entrada

En esta investigación se ha recurrido a las técnicas de simulación numérica para la generación de los patrones de entradas necesarios para probar la capacidad de los modelos neuronales diseñados. En el problema del impacto balístico, los datos disponibles incluyen información acerca de los parámetros geométricos de los sólidos involucrados, i.e. radio y longitud del proyectil y espesor de la protección, la velocidad de impacto y el tipo de material para el proyectil y la protección.

A la hora de realizar la simulación computacional ha sido necesario detallar, para los objetos que intervienen dentro del fenómeno del impacto balístico, aquellas variables a tener en cuenta para la simulación. Los parámetros considerados hacen referencia a características geométricas y referidas al material y a las condiciones de impacto de acuerdo con lo siguiente:

- Relativas a la geometría de los componentes:
 - Área de la sección recta.
 - Forma de la sección recta.

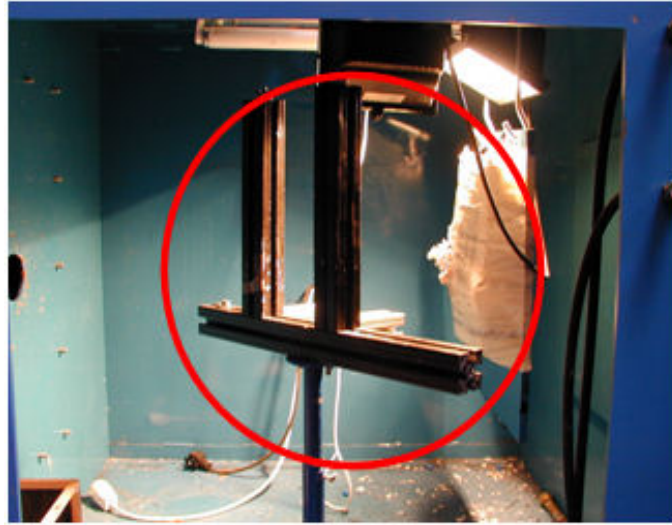


Figura 3.9: Útil de sujeción de paneles instalado dentro de la cámara principal del cañón de gas

- Espesor de la pared del tubo.
- Longitud del tubo.
- Relativas a las propiedades del material.
 - Límite elástico.
 - Coeficiente de endurecimiento por deformación.
 - Coeficiente de endurecimiento por velocidad de deformación.
 - Deformación de rotura.
- Relativas a las condiciones de impacto.
 - Velocidad de impacto.

En cuanto al resultado del impacto y el comportamiento de los objetos durante el mismo, se conocerá para el primero si se ha logrado una perforación y para el segundo, la velocidad residual y masa final.

Con esta distribución de la información, se generarán una serie de ensayos balísticos, mediante la herramienta comercial ABAQUS/Explicit v6.4.1⁵, un código de elementos finitos explícito, de propósito general, ampliamente empleado en la simulación de problemas dinámicos con un alto grado de no linealidad [ABAQUS, 2003]. Entre sus ventajas destaca la capacidad para resolver problemas de mecánica de sólidos, lineales y no lineales, dentro de los rangos estático y dinámico. Incluye modelos de material de tipo

⁵http://www.simulia.com/products/abaqus_explicit.html

elastoplástico y viscoplastico, entre otros. Cuando se refiere a problemas de no linealidad geométrica, incluye formulaciones para grandes deformaciones y problemas de contacto.

Para la simulación de estos datos, se ha llevado a cabo un análisis dinámico adiabático, con tiempos de integración comprendidos entre 120 y 180 μs , intervalos suficientes para garantizar, en función de la velocidad de impacto y de la geometría de los sólidos, bien la detención completa del proyectil o la perforación de la placa.

Las velocidades de impacto se han fijado entre 270 y 1195 m/s, recogiendo así el rango de valores habitual en este tipo de problemas y algunos casos extremos, asegurando así la validez de las simulaciones a realizar. El proyectil se ha modelizado como un cilindro de altura L_p y radio R_p , mientras que la placa como un prisma recto con espesor H_i y una sección cuadrada de lado Y_i . Los valores de estos parámetros geométricos se generan aleatoriamente con una función de densidad de probabilidad uniforme, guardando las siguientes restricciones para mantener las proporciones geométricas:

$$\begin{aligned} 3 \text{ mm} &\leq R_p \leq 8 \text{ mm} \\ 4R_p &\leq L_p \leq 8R_p \\ Y_i &= 2L_p \\ L_p/4 &\leq H_i \leq 0.8L_p \end{aligned}$$

De esta forma se obtienen casos que guarden proporción geométrica y sentido físico. La generación de la malla es paramétrica, tomando como parámetros los siguientes datos de entrada: (L_p, R_p, H_i, Y_i) para conseguir una óptima generación de todos los casos, dado el elevado número de ellos que hay que analizar. Para el mallado se emplean elementos hexaédricos de ocho nodos e integración reducida a un único punto de Gauss, C3D8R en notación del código ABAQUS/Explicit. El mallado es regular, tanto para el proyectil como para la placa, con una longitud característica de elemento $L_c=1.2$ mm. Los lados de la placa se consideran empotrados.

El algoritmo de contacto empleado para modelar el impacto entre el proyectil y el objetivo considera dos superficies, estando la maestra formada por las caras inferiores de todos los elementos del proyectil y la esclava por los todos los nodos de la placa. El coeficiente de fricción entre ambas superficies es $\mu =0.2$.

En notación ABAQUS, el algoritmo utilizado para todas las simulaciones numéricas ha sido el *General contact-kinematic contact algorithm*. Éste predice, en cada incremento de tiempo, una configuración cinemática sin considerar las condiciones de contacto. Posteriormente determina los nodos de una superficie de contacto que penetran en la otra. Por último, calcula las fuerzas de contacto que se oponen a esta penetración en función de la profundidad de ésta, la masa nodal y el incremento de tiempo. El algoritmo aplica una ponderación a ambas superficies para calcular la penetración de los nodos. La ponderación puede ser balanceada, asignando un peso similar a cada superficie de

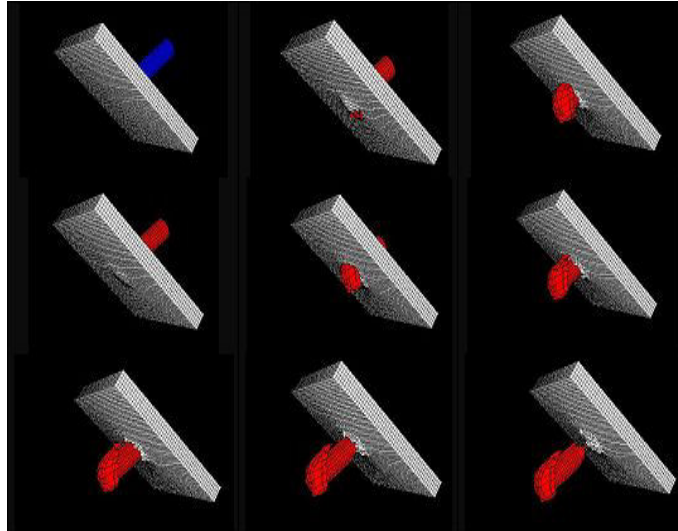


Figura 3.10: Simulación mediante el código ABAQUS/Explicit del impacto oblicuo de un proyectil de acero sobre una placa metálica de aluminio

contacto, o maestra-esclava, en la que los nodos de la superficie esclava no pueden penetrar en la superficie maestra.

En los problemas de impacto, en el que las tensiones son muy elevadas, es fundamental el tratamiento de la excesiva distorsión de los elementos en las mallas de Lagrange. Por tanto se ha elegido un criterio de erosión para borrar cada elemento cuando la deformación plástica equivalente en el punto de integración supera el valor del umbral $\epsilon_{failure}$. Este valor se fijó en 1.5 para el proyectil y el 2.5 para el objetivo para obtener resultados coherentes con las observaciones experimentales. Cuando los elementos de contacto entre el blanco y proyectil se suprimen, las nuevas superficies de contacto se introducen en el algoritmo de contacto para garantizar la continuidad entre ellos. Esto evita los problemas numéricos resultantes de esta distorsión, y la pérdida de masa en el proyectil, un fenómeno típico en los problemas de impacto, es modelado adecuadamente.

En la Figura 3.10 puede observarse la secuencia de un impacto oblicuo sobre una placa de aluminio, obtenida mediante el código ABAQUS/Explicit.

Por último, la simulación ha sido realizada mediante la utilización del MEF como herramienta de simulación numérica en los siguientes equipos:

- Estación de trabajo con dos procesadores Intel Itanium.
- Dos estaciones de trabajo con dos procesadores Intel Pentium IV.

3.4.5. Materiales metálicos empleados

En las décadas anteriores los materiales metálicos han sido los que más atención han recibido, por lo que existen numerosos modelos para ellos tanto fenomenológicos como de origen físico. Presentan generalmente una marcada sensibilidad a la velocidad de deformación y cuando esta alcanza valores elevados, el material puede sufrir un calentamiento adiabático por deformación plástica que, a efectos resistentes, se traduce en un ablandamiento. Es por tanto, necesario emplear una ley de endurecimiento dependiente de la deformación plástica equivalente $\bar{\epsilon}_{pl}$, de la velocidad de la deformación plástica equivalente $\dot{\bar{\epsilon}}_{pl}$ y de la temperatura T .

Los materiales seleccionados para la recreación, tanto del proyectil como de la protección, han sido identificados mediante un código numérico secuencial en lugar de usar los parámetros de sus correspondientes ecuaciones constitutivas. La segunda alternativa hubiera obligado a aumentar considerablemente el número de variables de entrada de la red. Los materiales utilizados han sido los siguientes:

1. Aluminio 6061-T6 (AL).
2. Acero 4340 AISI y Acero templado (AC).
3. Uranio Empobrecido (UE).
4. Wolframio (WHA).

Los materiales empleados en las diferentes placas metálicas han sido AL, AC y UE. Para los proyectiles se ha calculado la masa de cada uno de ellos en función de los materiales utilizados: AC, UE y WHA. Para ello se ha seguido la Ecuación 3.1, donde M_b corresponde a la masa, R_b al radio, L_b a la longitud y d a la densidad, todas relativas al proyectil. Las densidades de los materiales utilizados en los proyectiles son las siguientes:

1. AC, $d = 7850 \text{ kg/m}^3$
2. UE, $d = 18700 \text{ kg/m}^3$
3. WHA, $d = 18560 \text{ kg/m}^3$

$$M_b = \pi * R_b^2 * L_b * d \quad (3.1)$$

Para su modelización se consideró una ley de endurecimiento isótropo dependiente de la deformación plástica $\bar{\epsilon}_{pl}$, la velocidad de deformación plástica $\dot{\bar{\epsilon}}_{pl}$ y la temperatura T . Estas dos últimas variables deben tenerse en cuenta en este tipo de fenómenos, en los que las velocidades de deformación son muy elevadas y el trabajo plástico genera un notable aumento adiabático de la temperatura en el material metálico.

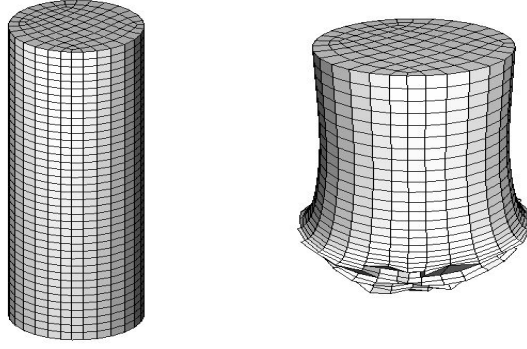


Figura 3.11: Proyectoil antes (izquierda) y después del impacto (derecha)

El modelo de Johnson-Cook [Johnson and Cook, 1983] es probablemente el modelo termoviscoplastico más extendido en la actualidad entre los códigos de simulación numérica, bien por MEF o MDF. Se emplea con éxito en problemas en los que el material está sometido a altas velocidades de deformación, tales como el impacto balístico, mecanizado, *Crashworthiness*, conformado rápido, etc.

Es una ecuación constitutiva fenomenológica con endurecimiento isótropo. La ley viene dada por la expresión:

$$\bar{\sigma}(\bar{\epsilon}_{pl}, \dot{\bar{\epsilon}}_{pl}, T) = (A + B\Delta\bar{\epsilon}_{pl}^n)\Delta(1 + C\Delta Ln\frac{\dot{\bar{\epsilon}}_{pl}}{\epsilon_0})\Delta(1 - \hat{\theta}^m) \quad (3.2)$$

donde $\hat{\theta}$ es la temperatura homologa que viene definida como:

$$\hat{\theta} = \frac{T - T_0}{T_m - T_0} \quad (3.3)$$

Por su parte, T_m and T_0 son las temperaturas de fusión y de referencia respectivamente. A, B, C, n, ϵ_0 y m son constantes del material determinados de forma experimental. El primer y el segundo factor introducen el endurecimiento del material, mientras que el tercero refleja el ablandamiento del material mediante calentamiento adiabático.

La ecuación se encuentra implementada en algunos de los códigos comerciales de simulación más extendidos, como ABAQUS/Explicit, AUTODYN o LS-Dyna. A ello ha contribuido que para las aleaciones metálicas utilizadas con mayor asiduidad, los valores de los parámetros del modelo están disponibles en la bibliografía científica.

Se ha considerado así mismo un criterio de erosión de elementos basado en un valor

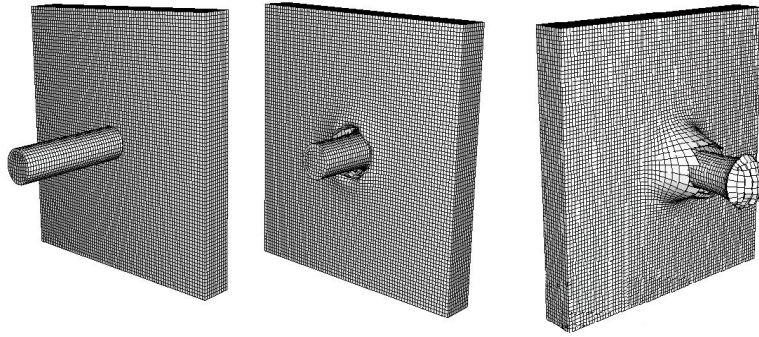


Figura 3.12: Secuencia de la simulación de un caso de perforación: vistas anteriores (izquierda y centro) y vista final posterior (derecha)

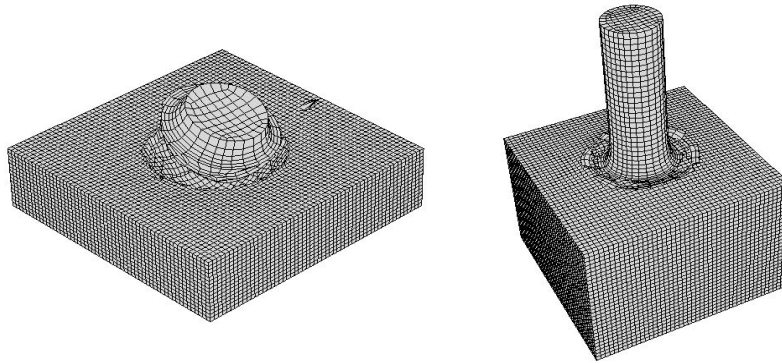


Figura 3.13: Proyectil de aluminio contra placa de uranio a velocidad alta (izquierda) y Proyectil de acero contra placa de uranio a velocidad media (derecha)

crítico de la deformación plástica. Los elementos son eliminados cuando $\bar{\epsilon}_{pl} \geq \epsilon_{failure}$ lo que permite considerar la pérdida de masa del proyectil (cf. Figura 3.11), fenómeno presente en los problemas de impacto a alta velocidad.

En la Figura 3.12 se muestra una secuencia del proceso de impacto en un caso de perforación de la protección, mientras que en la Figura 3.13 se pueden observar dos impactos a alta y media velocidad modelados con la herramienta ABAQUS. Una vez realizada la simulación de los casos, se han calculado la masa residual del proyectil y la velocidad de su centro de masa, parámetros considerados de gran interés para el diseño de protecciones frente a impacto y que deben, por tanto, constituir la salida de la red neuronal. Para ello se ha desarrollado una herramienta de postproceso que permite calcular estos parámetros, a partir de las coordenadas y velocidades nodales.

3.4.6. Validación del código de simulación

Dada la importancia de la esta técnica computacional dentro de esta investigación, es necesario que el código numérico sea validado. Con este fin se han llevado a cabo una serie de ensayos de compresión estática de acuerdo con la siguiente metodología:

1. Selección de un caso de estudio.
2. Simulación numérica del caso seleccionado.
3. Realización del ensayo usando la misma configuración analizada numéricamente.
4. Ajuste de las propiedades mecánicas.
5. Usando las propiedades mecánicas obtenidas en el paso anterior, simulación de otra configuración geométrica.
6. Realización de ensayos sobre la segunda configuración.
7. Comparación de resultados. Validación de la simulación numérica.

Gracias a esto se ha certificado la validez de los modelos de simulación numérica diseñados para recrear el fenómeno del impacto balístico.

3.4.7. Ensayos balísticos obtenidos

Los ensayos balísticos generados mediante simulación numérica recrean el comportamiento del proyectil y de la protección después del impacto. Se ha obtenido un conjunto de casos de impacto con diferentes valores de los parámetros descriptivos de cada una de las estructuras, proyectil y protección, que intervienen en el problema. El conjunto de patrones de entrada contiene un total de 220 ensayos balísticos generados aleatoriamente siguiendo las pautas marcadas en §3.4.4. El reparto de los mismos por escenarios es de 157 para perforación y 63 para detención.

Debido al dominio de valores de los materiales empleados, tanto de la protección como del proyectil, estos han sido identificados mediante un código numérico secuencial en lugar de usar los parámetros de sus correspondientes ecuaciones constitutivas (cf. §3.4.5 para más detalles).

En la Tabla 3.1 pueden observarse las variables descriptivas utilizadas en el problema del impacto balístico, junto con su magnitud y rango de valores. Cada vector de entradas utilizado para el entrenamiento de la red, estará compuesto por un conjunto de valores de estas 7 variables:

Variable descriptiva	Abreviatura	Magnitud	Rango
Longitud proyectil	L_b	mm	$12 \text{ mm} \leq L_p \leq 62 \text{ mm}$
Radio proyectil	R_b	mm	$3 \text{ mm} \leq R_p \leq 8 \text{ mm}$
Masa proyectil	M_b	gr	$3 \text{ gr} \leq M_p \leq 221 \text{ gr}$
Velocidad proyectil	V_b	m/s	$270 \text{ m/s} \leq V_p \leq 1195 \text{ m/s}$
Espesor protección	H_p	mm	$3 \text{ mm} \leq H_i \leq 47 \text{ mm}$
Tipo de material protección	T_p	-	AL, AC o UE
Tipo de material proyectil	T_b	-	AC, UE o WHA

Tabla 3.1: Variables descriptivas del problema del impacto

Variable salida	Abreviatura	Descripción	Magnitud
Perforación	P_b	Indica que el proyectil atraviesa la protección tras el impacto. Utilizado en el problema de clasificación para analizar los resultados.	-
Detención	D_b	Indica que el proyectil no atraviesa la protección tras el impacto. Utilizado en el problema de clasificación.	-
Masa residual	M_{res}	Masa del proyectil después de perforar la protección. Utilizado en el problema de regresión.	gr
Velocidad residual	V_{res}	Velocidad del proyectil al perforar la protección. Utilizado en el problema de regresión.	m/s

Tabla 3.2: Variables de salida para los modelos diseñados

$$V_{ent} = \{M_b, L_b, R_b, V_b, H_p, T_p, T_b\}$$

La Tabla 3.2 muestra las salidas a calcular por la red en cada uno de los modelos creados. En el caso de la clasificación, a partir de un patrón de entrada la RNA predice la ocurrencia de perforación o detención. Para ello se emplean dos neuronas cuya salida es complementaria: penetración SI o penetración NO:

$$S_{clf} = \{P_b, D_b\}$$

Por su parte en la regresión se determinan, mediante el cálculo del coeficiente de correlación, los valores residuales para la velocidad y la masa si el proyectil consigue atravesar la protección:

$$S_{per} = \{V_{res}, M_{res}\}$$

Para el último escenario y también en regresión, si la protección detiene el proyectil durante la perforación, se tiene el siguiente conjunto de variables:

$$S_{det} = \{M_{res}\}$$

En el anexo B se encuentra el listado de los ensayos balísticos obtenidos mediante la simulación numérica durante esta investigación. En la Figura 3.14 se muestran los diagramas de caja asociados al conjunto de datos disponible para este problema. No se han incluido la variable P_b , por ser dicotómica, ni las variables T_p y T_b , por tener ser discretas con rango cerrado. En cada uno de los gráficos se suministra información relativa a los valores mínimo y máximo, los cuartiles Q1, Q2 o mediana y Q3, y sobre la existencia de valores atípicos y la simetría de la distribución.

Un punto de especial interés en este caso es la detección de valores atípicos, también denominados *outliers*. Se trata de detectar la existencia de observaciones que no siguen el mismo comportamiento que el resto. Es necesario tener en cuenta que los casos atípicos pueden deberse a errores en el procedimiento a la hora de introducir los datos o de codificarlos. Una vez detectados los casos atípicos el analista debe saber elegir con criterio entre eliminarlos del análisis o evaluar toda la información incluyéndolos.

Existe una primera categoría de datos atípicos formada por aquellas observaciones que provienen de un error de procedimiento, por ejemplo un error de codificación, error de entrada de datos, etc. Una segunda categoría de casos atípicos contempla aquellas observaciones que ocurren como consecuencia de un acontecimiento extraordinario existiendo una explicación para su presencia en la muestra. Éstos generalmente se retienen salvo que no sean relevantes. Una tercera categoría comprende las observaciones extraordinarias para las que el investigador no tiene explicación, las cuales normalmente se eliminan del análisis. Una última categoría la forman las observaciones que se sitúan fuera del rango ordinario de valores de la variable. Suelen denominarse valores extremos y se eliminan del análisis si se observa que no son elementos significativos para la población [Perez-Lopez, 2004]. Por tanto, las propias características de cada caso atípico, así como los objetivos del análisis que se realiza, determinan cuales van a ser eliminados.

En este caso las distribuciones más simétricas son las de las variables V_b y V_{res} . Además aunque se han detectado valores atípicos para las variables M_b , H_p y M_{res} se han incluido en el conjunto de ensayos final ya que se consideran pertenecientes a la segunda categoría y no se quiere renunciar a ninguna de las simulaciones realizadas. Los diagramas de caja muestran entre otros datos de interés los valores mínimo y máximo, la mediana (línea horizontal), la media (símbolo +) y los intervalos de confianza para cada variable representados por una muesca en la caja de cada distribución. El intervalo de confianza calculado es del 95 % y permite aproximar entre que dos valores se estima que estará cierto valor desconocido para esa determinada probabilidad de acierto. Los bigotes se extienden

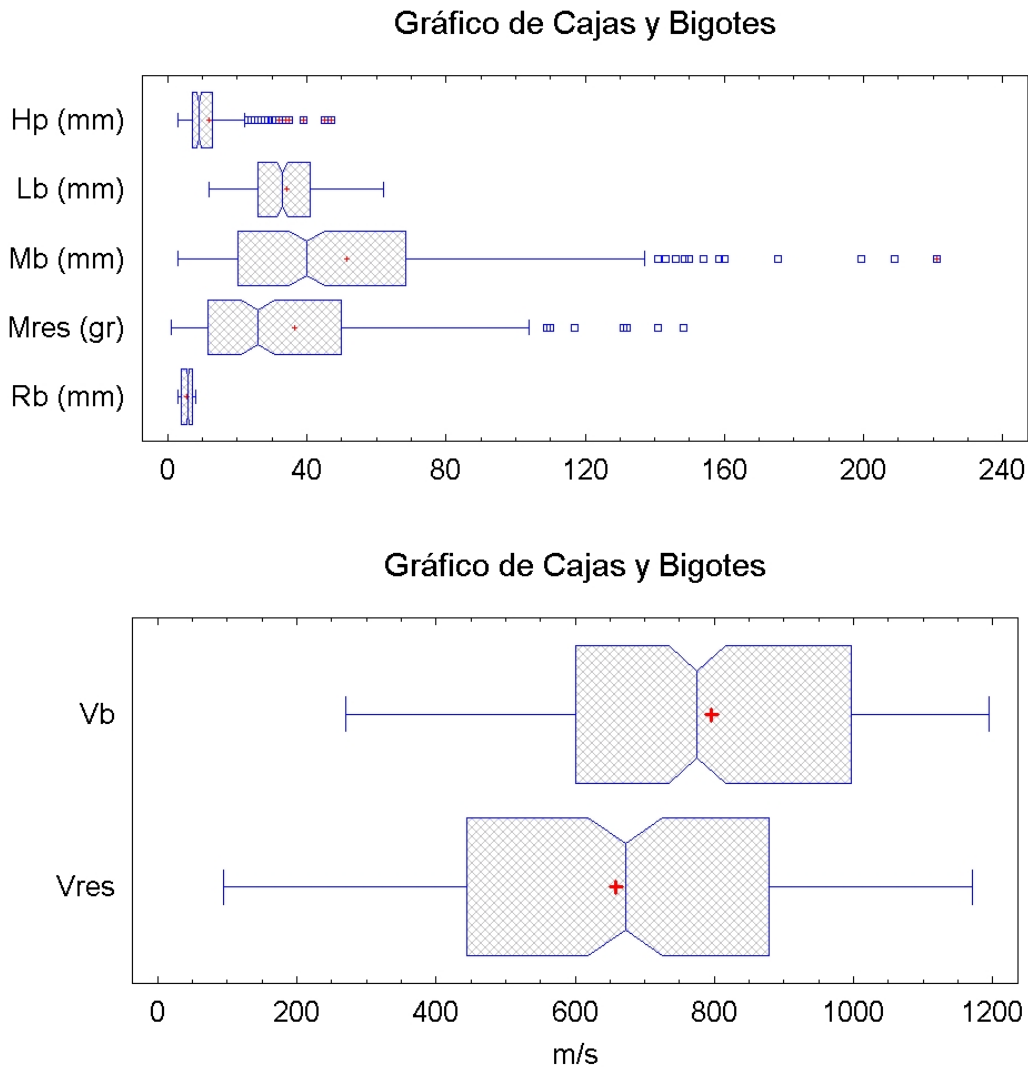


Figura 3.14: Diagramas de caja asociados al conjunto de variables del problema

desde la caja hasta los valores mínimo y máximo de la muestra, excepto los puntos externos o los puntos externos lejanos, los cuales se representan por separado. Los puntos externos son los que quedan a más de 1.5 veces el rango intercuartílico, por encima o por debajo de la caja, y se representan como pequeños cuadrados. Los puntos externos lejanos son los que quedan a más de 3.0 veces el rango intercuartílico, por encima o por debajo de la caja, y se representan como pequeños cuadrados con el signo más en su interior. En este caso, hay 36 puntos externos y 13 puntos externos lejanos asociados a todas las distribuciones. La presencia de puntos externos lejanos puede indicar valores atípicos o una distribución altamente sesgada. Por tanto, este análisis puede ser de utilidad para conocer las causas de los resultados obtenidos durante la evaluación y validación (cf. Capítulo 5).

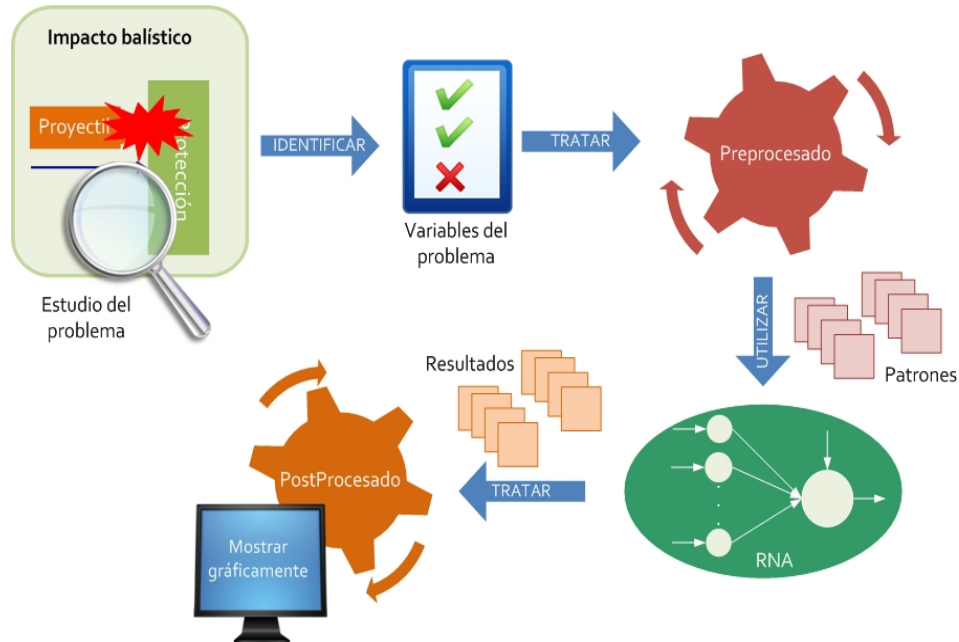


Figura 3.15: Diagrama de bloques con las fases clave operacionales utilizadas para cada modelo neuronal

3.5. Modelos Neuronales

Las RNA son herramientas de gran flexibilidad lo que facilita su aplicación en una amplia variedad de dominios. Esta capacidad de adaptación se consigue gracias a la variedad de alternativas de diseño que aparecen durante la creación de un modelo neuronal. Sin embargo la gran cantidad de posibilidades y la falta de directrices y guías de uso general hace que el diseñador realice una toma de decisiones a veces arbitraria o basada en técnicas de fuerza bruta. En algunos casos se han propuesto una serie de aproximaciones teóricas que pueden facilitar en parte este proceso, pero que no se consideran soluciones analíticas al no poder aplicarse en todos los casos.

Además la resolución de problemas utilizando estrategias basadas en RNA se complementa con una serie de pasos previos y posteriores a su diseño, que facilitan la consecución de resultados eficientes en la mayoría de dominios. En la Figura 3.15 se muestran las etapas más importantes que se desarrollan durante la resolución de un escenario balístico aplicando un modelo neuronal concreto. Todas ellas serán analizadas y justificadas con detalle en los siguientes apartados, agrupando bajo la premisa $\text{modelo} = \text{patrones} + \text{arquitectura} + \text{algoritmo}$ los detalles tenidos en cuenta para cada uno de los términos de la ecuación. Si bien los términos de la ecuación aparecen en el orden lógico de empleo dentro de esta investigación, para definir el primer modelo neuronal o modelo base para cada escenario se alterará el orden, apareciendo primero el término arquitectura. Esto es debido a que la cantidad mínima de patrones necesarios para cada escenario es dependiente de la arquitectura diseñada en cada caso.

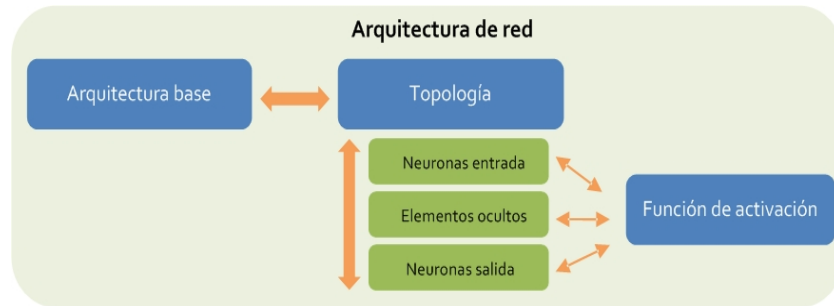


Figura 3.16: Pasos para la definición de la arquitectura básica de una RNA

En los siguientes apartados se justifican las decisiones tomadas para diseñar los modelos base en cada escenario y asegurar la validez del número de patrones disponibles.

3.5.1. Arquitectura de red

El primer paso a la hora de diseñar un sistema basado en RNA es seleccionar la arquitectura apropiada. Actualmente, existen unos 40 paradigmas de RNA que son usados en diversos campos de aplicación. Entre ellos, la arquitectura PMC entrenada con el algoritmo RP se ha mostrado como la más útil dentro del ámbito de la ingeniería. Como demuestran Liu [Liu et al., 2002] e Ince [Ince, 2004] su utilización se ajusta de forma precisa a las características de todo tipo de problemas ingenieriles y es escalable a problemas de grandes dimensiones [Principe et al., 1999]. En §2.3.1 se ha expuesto una justificación más detallada de esta elección.

La arquitectura óptima asociada a un PMC se obtiene mediante un proceso de prueba y error, consistente en el entrenamiento y verificación de diversas configuraciones de elementos de diseño hasta conseguir la que obtiene el menor error. En la Figura 3.16 se exponen las decisiones de diseño más importantes tomadas durante la construcción de las diferentes arquitecturas de red. El objetivo es por tanto localizar la estructura que obtenga, para un conjunto de variables de entrada, el mejor rendimiento a la hora de generalizar nuevos valores. Aunque en §2.4 ha quedado clara la influencia de ciertos parámetros de la RNA en sus resultados, no existen formulaciones analíticas que permitan determinar los valores óptimos de dichos parámetros inherentes al PMC. Esto es debido a que estos dependen de la naturaleza de los datos suministrados a la red, que son propios de cada problema. No obstante, existen ciertas recomendaciones prácticas y aproximaciones útiles a la hora de construir una RNA. Éstas han sido tenidas en cuenta para realizar, tanto el diseño inicial de la red, como las posteriores variantes asociadas al proceso de optimización.

Uno de los aspectos fundamentales de la arquitectura de un PMC son las diferentes capas donde se organizan las neuronas. La capa de entrada y la capa de salida son dependientes del problema a resolver, ya que las características de los patrones de entrada y la naturaleza de la salida a calcular determinan la cantidad de neuronas necesarias en cada una. En el dominio de esta investigación, las variables de entrada y salida utilizadas para cada uno de los escenarios las establecen los ensayos balísticos obtenidos mediante simulación numérica. Las características de estos ensayos pueden observarse en §3.4.7. Por su parte la capa oculta aporta al modelo neuronal la capacidad de aproximar funciones no lineales, i.e. permite situar al PMC como una nueva clase de función capaz de aproximar e interpolar relaciones no lineales entre datos de entrada y de salida. Gracias a los estudios de Hornik et al. [Hornik et al., 1989] y Cybenko [Cybenko, 1989], se conoce que un PMC con al menos una capa oculta es capaz de aproximar cualquier función. No obstante, conviene recordar que en algunos casos específicos se ha demostrado que la introducción de una segunda capa oculta permite mejorar la velocidad de convergencia. Un número superior de capas ocultas no suele ser necesario, siendo preferible realizar una selección de variables de entrada eficiente, ya que la red aprende mejor cuando la correlación entre las entradas y las salidas es más sencilla. En resumen, la tarea del diseñador pasa por elegir el número adecuado de capas y especialmente de neuronas ocultas, teniendo en cuenta que un número excesivamente grande o pequeño de las mismas hará que la red no converja bien.

A raíz de lo anterior la topología de la red estará compuesta, en su capa de entrada, por un conjunto inicial de siete neuronas relativas a las características de los objetos que intervienen en el impacto balístico: proyectil y protección (cf. Tabla 3.1). Uno de los objetivos de la optimización de los modelos neuronales es determinar el conjunto óptimo de variables de entrada, detectando aquellas que no aporten información relevante o las posibles interrelaciones que afecten a las salidas.

La capa intermedia por su parte puede presentar múltiples configuraciones posibles, partiendo de una capa oculta y un número inicial de neuronas. Para aproximar este valor inicial se ha utilizado la formulación propuesta por Tarassenko [Tarassenko, 1998]. Si se aplican sus heurísticas en el dominio investigado se obtiene, aproximando por exceso, el número de neuronas inicial de la capa oculta para asegurar un rendimiento adecuado es:

$$J = \sqrt{IK} \Rightarrow J = \sqrt{7 \times 2} \approx 4 \quad (3.4)$$

donde I , J y K son el número de neuronas de las capas de entrada, oculta y salida respectivamente. Esta aproximación inicial es común para el escenario de clasificación y el de perforación en regresión, ya que el tamaño del conjunto de entrada y el de salida es común ambos. Por último, para el escenario de detención en regresión el valor aproximado es:

$$J = \sqrt{IK} \Rightarrow J = \sqrt{7} \approx 3 \quad (3.5)$$

Por su parte el límite superior según la heurística enunciada por Cover [Cover, 1965] se establece como:

$$H < \frac{0,5 \cdot P - 1}{M + 1} \Rightarrow H < \frac{0,5 \cdot 220 - 1}{7 + 1} \Rightarrow H < 13 \text{ para } S_{clf} \quad (3.6)$$

$$H < \frac{0,5 \cdot P - 1}{M + 1} \Rightarrow H < \frac{0,5 \cdot 157 - 1}{7 + 1} \Rightarrow H < 10 \text{ para } S_{per} \quad (3.7)$$

$$H < \frac{0,5 \cdot P - 1}{M + 1} \Rightarrow H < \frac{0,5 \cdot 63 - 1}{7 + 1} \Rightarrow H < 4 \text{ para } S_{det} \quad (3.8)$$

donde P es el número de patrones y M el número de variables o características de entrada.

Este rango de neuronas ocultas, [4, 12], [4, 9] y [3, 3], para una única capa oculta será ratificado durante la optimización de los modelos. Es importante estimar correctamente el número de neuronas ocultas, con sus respectivos pesos, para mantener en valores bajos el número de parámetros libres; lo que permite reducir la cantidad de patrones de entrenamiento necesarios. A partir de la arquitectura inicial, se realizarán diversas variaciones utilizando el conjunto CV para localizar la que presente un mejor rendimiento en cada escenario.

El número de neuronas de la capa de salida es dependiente del problema. En la Tabla 3.2 se indican las variables salida que se han utilizado en cada escenario. Además, desde el punto de vista conceptual, la principal diferencia entre los modelos neuronales se centra en esta capa ya que en función del tipo de problema las neuronas clasifican o aproximan valores. En el primer caso las salidas se corresponden con el conjunto de clases que forman parte del problema, mientras que en el segundo se asocian con la cantidad de valores a aproximar como salida. En ambos casos el objetivo del proceso de entrenamiento es optimizar el peso de las conexiones para conseguir minimizar el error devuelto por la red, el cuadrado de la distancia euclídea entre las salidas obtenidas y las deseadas. El algoritmo RP es el encargado de minimizar este error hasta cero lo más rápidamente posible.

Otro aspecto relativo a la arquitectura de la red, que es necesario fijar antes de empezar su entrenamiento, es la función de activación asociada a cada neurona. Típicamente todas las neuronas de una capa comparten la misma función de activación, siendo requisito imprescindible al utilizar el algoritmo RP que la función elegida sea derivable. La función más utilizada es sin duda la sigmoideal debido a sus propiedades matemáticas, monotonía, continuidad y diferenciabilidad, las cuales son muy importantes durante el entrenamiento de la red al utilizar algoritmos derivados del descenso del gradiente. Otras funciones muy utilizadas son la tangente hiperbólica, la lineal, la escalón, etc. Sin embargo las ventajas asociadas a cada una de las posibles alternativas no han sido demostradas matemáticamente [Hassoun, 1995]. Por ello, durante la metodología de optimización se

estudiará la influencia de las diversas alternativas dentro de cada escenario.

Finalmente, como se comentó en §3.3 esta tesis doctoral se centra en la resolución de dos problemas: el primero trata de conocer el resultado del impacto desde el punto de vista de la penetración y el segundo estudiar el comportamiento de los objetos que intervienen, proyectil y panel protector. Si se traslada esta disyuntiva al área de la IA, y más concretamente de las RNA, su resolución implica el diseño de dos modelos diferentes, uno enfocado a la clasificación de patrones y el otro a aproximar los valores de ciertos factores característicos del problema: Velocidad Residual (V_{res}) y Masa Final (M_{res}). A continuación se detallan los modelos neuronales diseñados para cada uno de estos escenarios.

3.5.1.1. Arquitecturas neuronales en clasificación

El modelo neuronal diseñado en este apartado tiene como objetivo cubrir el problema del impacto balístico para el escenario descrito en §3.3.1. En la Figura 3.17 se muestra la arquitectura inicial de la red para dicho escenario. Como ha quedado expuesto en el apartado anterior, la cantidad de elementos ocultos deberá localizarse a través de un proceso de optimización iterativo, siendo inicialmente el reparto de neuronas por niveles el siguiente: [7, 4, 2]. Las funciones de activación seleccionadas inicialmente en cada capa, han sido extraídas del trabajo de Lefebvre et al. [Lefebvre et al., 2007]: identidad (f_{iden}) para la de entrada y tangente hiperbólica (f_{tanh}) para la oculta y la de salida.

La fortaleza de un PMC clasificador radica en su capacidad para reconocer patrones complejos del mundo real que pueden presentar ruido o incertidumbre debido a su naturaleza intrínseca. El cálculo de la incertidumbre asociada a cada patrón es probabilístico. Las salidas del modelo neuronal dentro de este ámbito clasificador estiman las probabilidades de que un patrón de entrada pertenezca a una clase u otra. En este dominio cada salida es dicotómica, con valores SI o NO, transformándose internamente a valores binarios [1, 0]. Para valores extremos el comportamiento de la red es obvio, sin embargo cuando se presentan valores cercanos a la frontera de decisión, e.g. 0.5, aparece cierta incertidumbre acerca de su clasificación en alguna de las clases. Para resolver estas situaciones se recurre al teorema probabilístico de Bayes, el cual adaptándolo al entorno de una PMC con k salidas se expresa como [Tarassenko, 1998]:

$$P(C_k|x) = \frac{p(x|C_k)P(C_k)}{\sum_{k=1}^k p(x|C_k)P(C_k)} \quad (3.9)$$

siendo la regla de decisión para asignar cada patrón x a una clase u otra la siguiente:

$$\text{asignar } x \text{ a la clase } C_i \text{ si } P(C_i|x) > P(C_j|x) \text{ para todo } j \neq i \quad (3.10)$$

Por tanto, la salida a calcular en clasificación se define por el conjunto $S_{clf} = \{P_b, D_b\}$ y refleja la probabilidad de pertenencia, a partir del teorema de Bayes enunciado, de

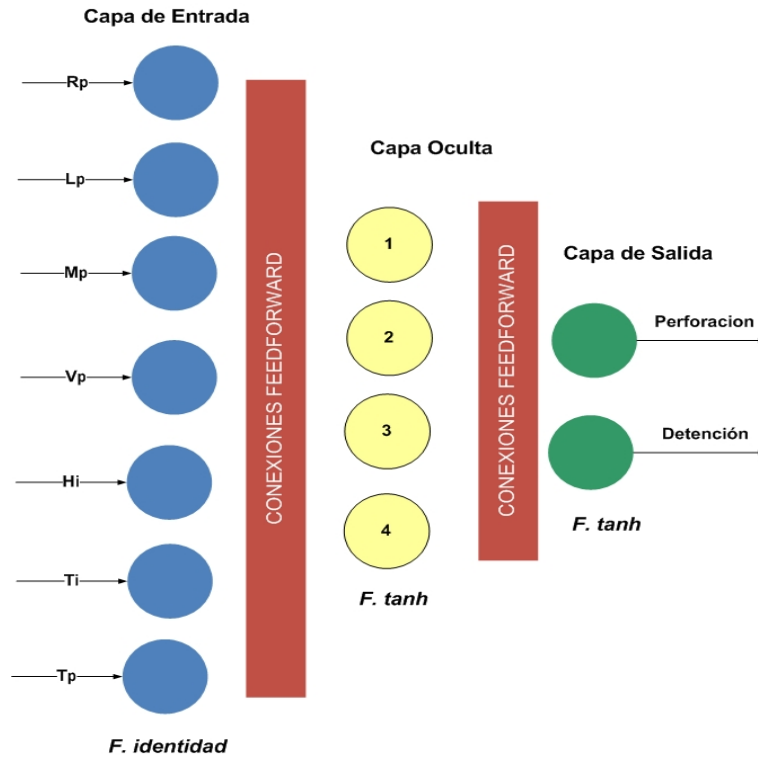


Figura 3.17: Estructura inicial de la RNA diseñada para el problema de clasificación

cada ensayo balístico a la clase perforación o detención.

Como información adicional, durante el entrenamiento y la validación, se calculan dos valores significativos para conocer la validez de cada modelo neuronal: porcentaje de error y error cuadrático medio (MSE). El primero se obtiene a partir de la siguiente ecuación:

$$E \% = \frac{100}{N \cdot P} \sum_{j=0}^P \sum_{i=0}^N \frac{\|dy_{ij} - dd_{ij}\|}{dd_{ij}} \quad (3.11)$$

Donde:

- P es el número de neuronas de salida
- N es el número de patrones del conjunto de datos
- y_{ij} es la salida obtenida desnormalizada para el ensayo i en la salida j
- d_{ij} es la salida esperada desnormalizada para el ensayo i en la salida j

Por otro lado, el error MSE es la medida típica del error usada para el comprobar el rendimiento y la calidad de los resultados generados, presentando la misma magnitud que las variables originales que se analizan, lo que permite obtener conclusiones más cercanas al dominio que se está tratando. Para ello establece la medida de las diferencias en promedio entre los valores estimados por la red y los esperados. Se trata de una medida de la precisión del mismo y su valor debería ser cercano a cero para expresar que no existen diferencias notables entre los valores calculados por la red y los reales.

$$MSE = \frac{\sum_{j=0}^P \sum_{i=0}^N (d_{ij} - y_{ij})^2}{N \cdot P} \quad (3.12)$$

Donde:

- P es el número de neuronas de salida
- N es el número de patrones del conjunto de datos
- y_{ij} es el valor pronosticado para el ensayo i en la salida j
- d_{ij} es el valor esperado para el ensayo i en la salida j

Por tanto, la base sobre la que se sustenta este estimador estadístico es el residuo o error relativo obtenido para cada ensayo de simulación, i.e la diferencia entre la salida calculada por la red y el valor real: $(d_{ij} - y_{ij})$. Al tratarse de un valor acumulado en el momento que uno de los valores calculados tenga una desviación grande, i.e. que la red cometa un error en la predicción, el valor final del MSE se verá afectado. Por último, el valor mínimo óptimo del MSE es dependiente de cada problema y de los patrones utilizados.

Otra alternativa ampliamente utilizada es el error MSE normalizado (NMSE), se trata de un estimador de las desviaciones globales obtenidas entre los valores reales y los obtenidos y viene definido por la siguiente fórmula:

$$NMSE = \frac{P \cdot N \cdot MSE}{\sum_{i=0}^P \frac{N \sum_{i=0}^N d_{ij}^2 - (\sum_{i=0}^N d_{ij})^2}{N}} \quad (3.13)$$

Donde:

- P es el número de neuronas de salida
- N es el número de patrones del conjunto de datos
- MSE es el error MSE
- d_{ij} es el valor esperado para el ensayo i en la salida j

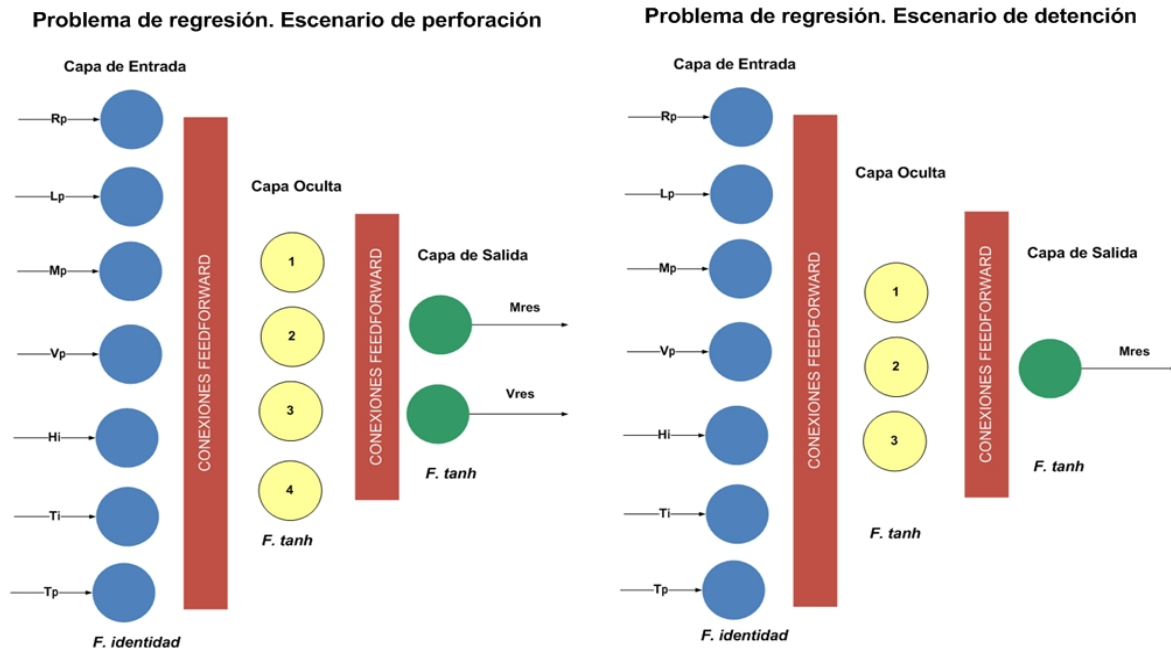


Figura 3.18: Estructuras iniciales de la RNA diseñadas para el problema de regresión en los escenarios de perforación y detención

En el error NMSE las desviaciones (errores absolutos) se suman en lugar de las diferencias. Por esta razón, el error NMSE permite observar diferencias notables entre los modelos. Si un modelo tiene un NMSE muy bajo se puede afirmar que presenta un buen rendimiento tanto en espacio como en tiempo.

3.5.1.2. Arquitecturas neuronales en regresión

Los modelos neuronales diseñados en este apartado tiene como objetivo cubrir el problema del impacto balístico para los escenarios descritos en §3.3.2. En la Figura 3.18 se muestra la arquitectura inicial de la red para cada uno de dichos escenarios. Como ha quedado expuesto anteriormente, la cantidad de elementos ocultos de estas arquitecturas deberá localizarse a través de un proceso de optimización iterativo, siendo el reparto inicial de neuronas por niveles el siguiente: $[7, 4, 2]$ para el escenario de perforación y $[7, 3, 1]$ para el escenario de detención. Las funciones de activación seleccionadas inicialmente en cada capa, han sido extraídas del trabajo de Lefebvre et al. [Lefebvre et al., 2007]: identidad (f_{iden}) para la de entrada y tangente hiperbólica (f_{tanh}) para la oculta y la de salida.

Atendiendo a la información que proporciona como salida cada modelo neuronal, en el primer escenario al tratarse de un caso de perforación, se predice la salida conjunta

$S_{per} = \{V_{res}, M_{res}\}$ que presenta el proyectil al atravesar la protección. En el segundo escenario, el proyectil queda alojado dentro de la protección por lo que la red calcula el conjunto $S_{det} = \{M_{res}\}$ para cada uno de los ensayos.

En estos escenarios de regresión, la red aproxima a partir de un patrón de entrada y mediante una función continua no lineal f (donde $y = f(x)$) los valores asociados a cada una de las salidas esperadas. Para ello, al realizar la predicción de una variable numérica, se calcula durante el entrenamiento y validación el error MSE, el coeficiente de correlación de Pearson (r) para cada una de las variables de salida.

El valor del error MSE expresado en la Ecuación 3.14 puede usarse para determinar como de bien se ajusta la salida de la red a la salida esperada, sin embargo esto no refleja necesariamente si los dos conjuntos de datos se están moviendo en la misma dirección. Por ejemplo con el escalado de las salidas de la red se varía el error MSE sin cambiar la direccionalidad de los datos. Por ello, para los escenarios de regresión, se obtiene de forma adicional el coeficiente r entre las salidas obtenidas por la red y las salidas esperadas o reales.

Este indicador estadístico describe la intensidad de la relación entre dos conjuntos de variables x (salida obtenida por la red) y d (salida esperada), i.e. es la medida de la intensidad de la relación lineal entre ellas. En concreto indica si el valor de una variable x aumenta o disminuye en relación con el aumento o disminución del valor de la otra variable y . Puede tomar valores dentro del rango $[1, -1]$, cuando más próximo a uno es el valor, más fuerte y positiva es la asociación. Por el contrario cuando más cerca está de -1 , más fuerte y negativa, i.e. incrementos en la variable x producen decrementos en d . Por último si el valor es cero o próximo a cero se dice que las variables son incorreladas. Se dice que una correlación es significativa si la misma se encuentra entre $\|0.7, 1\|$.

En los escenarios resueltos mediante modelos neuronales de regresión, se calcula para cada variable de salida el coeficiente de correlación lineal (r) que existe entre los valores reales de esa variable y los obtenidos por la red neuronal. Este coeficiente permite conocer el grado de similitud y de exactitud de la respuesta basándose en la ecuación:

$$r = \frac{\frac{\sum_i (x_i - \bar{x})(d_i - \bar{d})}{N}}{\sqrt{\frac{\sum_i (d_i - \bar{d})^2}{N}} \sqrt{\frac{\sum_i (x_i - \bar{x})^2}{N}}} \quad (3.14)$$

donde el numerador se corresponde con la covarianza y el denominador con el producto de las desviaciones típicas de las variables x (valor obtenido) y d (valor esperado).

Por su parte el porcentaje de error $E\%$, expresado en la Ecuación 3.11, permite observar las salidas de la red dentro de un rango $[0, 100]$ pero esto puede inducir a

equivocación en ciertos casos de regresión ya que pequeñas diferencias entre las salidas llevan asociadas un porcentaje de error elevado. Un ejemplo de esta situación es cuando para un ejemplar cuya salida real es 0.1 y la salida obtenida es 0.2 el error, a pesar de ser valores muy próximos, es del 100 %.

3.5.2. Patrones de entrada

En §2.4.2 se ha evidenciado que los tres factores relativos a los patrones con una mayor influencia sobre el rendimiento de los modelos neuronales son: pre-procesado, cantidad y reparto. Trasladando estas directrices al dominio de la investigación, se exponen a continuación las decisiones más interesantes tomadas en cada uno de estos aspectos.

Debido a que las RNA son herramientas dependientes de los datos, su correcta recolección, preparación y etiquetado es una tarea de gran relevancia durante la construcción de modelos neuronales. En la mayoría de las ocasiones, antes de comenzar, deben considerarse como datos de entrada aquellas variables relevantes o posiblemente relevantes para la red neuronal. No es necesario conocer la naturaleza de las relaciones existentes entre las entradas x y las salida y , solo que existe una gran posibilidad de relación entre ellas. La red neuronal se encargará de determinar una aproximación a la función f que la describe, entendida como $y = f(x)$.

Como condición de partida para la recolección de los datos, es necesario tener en cuenta, que ya sea en clasificación o en regresión, el aprendizaje utilizado en el PMC es de tipo supervisado. Por tanto, se necesita obtener información acerca de las entradas y las salidas a utilizar durante el entrenamiento. En regresión la salida será el valor asociado a cada una de las entradas, mientras que en clasificación se corresponderá con la clase a la que pertenece dicha entrada. Además los datos empleados como patrones es conveniente que se almacenen en su forma bruta, i.e es mejor no realizar sobre ellos ninguna alteración ya que esto podría conllevar la eliminación involuntaria de aspectos de relevancia para el entrenamiento de los modelos neuronales. Por ello, los ensayos incluidos en el Anexo B se muestran en el formato original obtenido directamente mediante las herramientas de simulación numérica.

Las técnicas de pre-procesado a utilizar, e.g. normalización de los valores, se han incluido dentro de la metodología de optimización enunciada en el Capítulo 4. En este punto de la investigación, una vez recolectada toda la información y para evitar comportamientos incorrectos, la única operación realizada tiene como objetivo controlar la validez de los datos basándose en las siguientes pautas:

1. Todos los elementos de cada vector de datos están dentro de los rangos esperados.
2. Todos los elementos de cada vector son mutuamente consistentes.

3. Los vectores objetivo son consistentes con los vectores de entrada, ya que se utilizará un algoritmo de aprendizaje supervisado.
4. No se incluyen elementos ambiguos en el vector de entrada, e.g. entradas idénticas con diferentes etiquetas.

Como la actualización de los pesos en un PMC es directamente proporcional a los valores de las entradas, se asegura que dentro de los datos no aparezcan valores que pueden conllevar un efecto perjudicial sobre el proceso de entrenamiento.

La siguiente cuestión que aparece relacionada con los patrones de entrada, es cuántos son necesarios para asegurar un correcto entrenamiento. Un número pequeño no asegura que se cubran todas las relaciones existentes entre las entradas y las salidas, por lo que el entrenamiento completo de la red no será realizado. En regresión por tanto se necesitan suficientes datos para representar a lo largo de todo el espacio de entradas de la función f , mientras que en clasificación es mejor que todas las clases aparezcan representadas de forma equitativa dentro del conjunto de datos de entrada. En el caso contrario, si se utilizan demasiados datos la red puede presentar problemas de rendimiento y la aparición de saturación por lo que la red no alcanzaría el error mínimo global.

Por tanto la cantidad de datos que se tienen que utilizar se puede resumir en la premisa expuesta por Tarassenko [Tarassenko, 1998]: “se necesita un número suficiente de datos para asegurar que la red será entrenada para reconocer el amplio rango de condiciones, no solo de operación normal sino también posibles condiciones de fallo, que pueda encontrarse en los datos del conjunto de test.”

En esta investigación se dispone de 220 ensayos, con el siguiente reparto de clases: 157 ensayos para perforación y 63 para detención y una dimensión de siete elementos para el vector de entradas. La teoría de la información sugiere que el número de ejemplos, i.e. número de vectores de entrada del conjunto de entrenamiento, debería ser al menos del mismo orden que el número de parámetros libres de la red. Los parámetros libres son los pesos que se ajustan con los datos de entrada que se presentan a la red. La formula asociada es la siguiente:

$$W = (I + 1)J + (J + 1)K \quad (3.15)$$

donde I es el número de entradas, J el número de unidades ocultas y K el número de salidas de la red. Como se puede observar para poder aplicar esta ecuación es necesario conocer ciertos aspectos de la arquitectura de la red. Si se tiene en cuenta la arquitectura de cada escenario explicada en §3.5.1, y las variables de salida incluidas en la Tabla 3.2, se puede adelantar lo siguiente:

$$(7 + 1)4 + (4 + 1)2 = 42 \quad \text{para } S_{clf} \text{ y } S_{per} \quad (3.16)$$

$$(7 + 1)3 + (3 + 1)1 = 28 \quad \text{para } S_{det} \quad (3.17)$$

La siguiente cuestión a resolver es la dimensión del vector de entrada, V_{ent} , para los modelos neuronales diseñados. Si se tiene en cuenta que la complejidad de la red depende de este aspecto, es importante estudiar la influencia real de cada componente del vector sobre la capacidad de la red. Durante la aplicación de la metodología de optimización del Capítulo 4 se realizan variaciones sobre V_{ent} que afectarán a las arquitecturas de red disminuyendo su complejidad.

Los estudios realizados por Tarassenko en [Tarassenko, 1998] establecen que teniendo en cuenta que cada nodo oculto y de salida tiene un peso asociado, se debería usar un número de patrones, P , al menos igual a W , i.e. $P = W \Rightarrow P = 42$ para S_{clf} y S_{per} y $P = 28$ para S_{det} .

Por otro lado, la regla de uso general más extendida para los PMC se obtiene a partir de los trabajos Baum y Haussler [Baum and Haussler, 1989] y Widrow y Lehr [Widrow, 1990]. Esta regla establece que el número de patrones debería ser estimado en función de la capacidad de la red, i.e. el ratio entre el número de pesos y el número de clases, $W/K \Rightarrow 42/2$ para S_{clf} y S_{per} y $28/2$ para S_{det} . De forma precisa el número de patrones debería ser mucho más grande que este ratio, $P \gg W/K \Rightarrow P \gg 21$ para el primer caso y $P \gg 14$ para el segundo. Usando este ratio desde un punto de vista cualitativo, teniendo en cuenta el número de patrones mal clasificados del conjunto de test, se estableció que los patrones a usar deberían ser unas diez veces superior al número de parámetros libres, i.e. $P = 10W \Rightarrow P = 420$ para S_{clf} y S_{per} y $P = 280$ para S_{det} . Por tanto siguiendo estas premisas, unos límites razonables para el número de patrones a utilizar son los siguientes:

$$W \leq P \leq 10W \Rightarrow 42 \leq P \leq 420 \quad \text{para } S_{clf} \text{ y } S_{per} \quad (3.18)$$

$$W \leq P \leq 10W \Rightarrow 28 \leq P \leq 280 \quad \text{para } S_{det} \quad (3.19)$$

Una heurística relacionada con la ecuación anterior es la dimensión VC o *Vapnik-Chervonenkis* [Vapnik, 1995]. Se trata de una medida utilizada para medir la capacidad de los algoritmos de clasificación estadística. Su utilización dentro de los clasificadores neuronales permite inferir el tamaño máximo del conjunto de patrones de entrenamiento, S , que permite separar todas las dicotomías 2^S de dicho conjunto S . En arquitecturas con una única capa oculta el valor mínimo se corresponde con el número de pesos existentes entre la capa de entrada y la oculta, mientras que el límite superior es el doble de pesos. En la práctica los diseñadores de RNA eligen como número de patrones de entrenamiento diez veces el valor de la dimensión VC [Priddy and Keller, 2005]:

$$S < 10W \Rightarrow S < 420 \quad \text{para } S_{clf} \text{ y } S_{per} \quad (3.20)$$

$$S < 10W \Rightarrow S < 280 \quad \text{para } S_{det} \quad (3.21)$$

Otras reglas empíricas utilizadas en clasificación son la regla de Foley [Foley, 1972] la cual expone que la relación entre el número de ejemplos por clase, C , y el tamaño del vector de entrada, N , debería ser superior a tres. Foley demostró que si se cumple esta condición el error resultante será muy cercano al obtenido por un clasificador bayesiano:

$$C/N > 3 \Rightarrow \frac{157}{7} > 3 \quad \text{para } S_{clf} \text{ en perforación} \quad (3.22)$$

$$C/N > 3 \Rightarrow \frac{63}{7} > 3 \quad \text{para } S_{clf} \text{ en detención} \quad (3.23)$$

La regla de Cover [Cover, 1965] por su parte, expone que se necesita un número de patrones superior al menos dos veces al número de parámetros libres, i.e. $P > 2w$, para asegurar que no aparecen generalizaciones ambiguas:

$$P > 2W \Rightarrow P > 84 \quad \text{para } S_{clf} \text{ y } S_{per} \quad (3.24)$$

$$P > 2W \Rightarrow P > 56 \quad \text{para } S_{det} \quad (3.25)$$

Si tenemos en cuenta la problemática asociada a la obtención de ensayos en este dominio y aceptando el reducido número disponible, a continuación se procederá a validar las ecuaciones anteriores. Se dispone de 220 ensayos, con el siguiente reparto de clases: 157 ensayos para perforación y 63 para detención y una dimensión de siete elementos para el vector de entradas, por lo que la resolución de dichas ecuaciones es:

$$42 \leq 220 \leq 420 \quad \text{según Tarassenko para } S_{clf} \text{ (Ecuación 3.18)} \quad (3.26)$$

$$42 \leq 157 \leq 420 \quad \text{según Tarassenko para } S_{per} \text{ (Ecuación 3.18)} \quad (3.27)$$

$$28 \leq 63 \leq 280 \quad \text{según Tarassenko para } S_{det} \text{ (Ecuación 3.19)} \quad (3.28)$$

$$220 < 420 \quad \text{según la dimensión VC para } S_{clf} \text{ (Ecuación 3.20)} \quad (3.29)$$

$$157 < 420 \quad \text{según la dimensión VC para } S_{per} \text{ (Ecuación 3.20)} \quad (3.30)$$

$$63 < 280 \quad \text{según la dimensión VC para } S_{det} \text{ (Ecuación 3.21)} \quad (3.31)$$

$$\frac{157}{7} > 3 \quad \text{según la regla de Fooley para } S_{clf} \text{ en perforación (Ecuación 3.22)} \quad (3.32)$$

$$\frac{63}{7} > 3 \quad \text{según la regla de Fooley para } S_{clf} \text{ en detención (Ecuación 3.23)} \quad (3.33)$$

$$220 > 84 \quad \text{según la regla de Cover para } S_{clf} \text{ (Ecuación 3.24)} \quad (3.34)$$

$$157 > 84 \quad \text{según la regla de Cover para } S_{per} \text{ (Ecuación 3.24)} \quad (3.35)$$

$$63 > 56 \quad \text{según la regla de Cover para } S_{det} \text{ (Ecuación 3.25)} \quad (3.36)$$

Una vez determinado el tamaño del conjunto de patrones, es interesante estudiar la dimensión del vector de entrada, i.e. el número de variables que describen el problema. A medida que crece la dimensión de dicho vector también aumenta la cantidad de información necesaria para poder entrenar la red correctamente. Aunque en principio puede parecer preferible tener un elevado número de características que describan los datos de entrada, puede suceder que la red se comporte peor con más información que con un vector más reducido. Esto es debido a que a medida que se añaden elementos se necesitan más ejemplos para que la red no presente un sobreaprendizaje sobre los datos. Este problema se conoce como la maldición de la dimensionalidad (*curse of dimensionality*), término acuñado por Bellman en 1961 [Bellman, 1961] y según sus palabras: “la maldición de la dimensionalidad es un problema que ha acosado a los científicos e ingenieros desde los primeros tiempos.”

La aparición de esta situación en las RNA se pone de manifiesto cuando cada nueva entrada a la red implica que una nueva dimensión del espacio debe ser localizada por el modelo neuronal. Durante el entrenamiento, cada dato representativo del espacio de entrada ocupado debe ser usado para entrenar la red, creándose una correlación entre las entradas y las salidas. Por otro lado cubrir todas las posibilidades del espacio de entrada presenta el inconveniente de un alto consumo de tiempo y recursos cuando este es demasiado grande.

Para conseguir un tamaño óptimo para cada problema se utilizan técnicas de selección y reducción de parámetros del vector de entradas. Si éstas técnicas se aplican sin control, el rendimiento de las RNA será insuficiente, por lo que para poder tomar la decisión correcta es necesario tener cierto conocimiento sobre el dominio del problema. Inicialmente en esta investigación el vector de entradas, V_{ent} , es el enunciado en §3.4.7. Uno de los objetivos de la metodología de optimización es encontrar aquellos parámetros no correlacionados y sin información redundante.

Por último, los datos obtenidos deberían dividirse de forma aleatoria entre los conjuntos a utilizar: entrenamiento y test o entrenamiento, validación y test. Idealmente la cantidad de patrones destinados a cada conjunto debería ser equitativa, pero esto no siempre es posible debido a las características de los datos. Por tanto, una correcta distribución de los patrones será otro los pasos definidos dentro de la metodología de optimización.

3.5.3. Algoritmo de entrenamiento de la red

El algoritmo RP es un algoritmo supervisado para el entrenamiento de RNA del tipo *feedforward* y en algunas recurrentes. Se trata del más extendido dentro de las redes PMC ya que su utilización es relativamente sencilla y tiene pocos parámetros que ajustar. Además su implementación tiene una complejidad reducida y es aplicable a un

amplio abanico de problemas. Entre sus características de funcionamiento destaca por su capacidad para formar regiones de decisión complejas, i.e. es un aproximador universal.

No obstante, su utilización en ciertos dominios presenta una velocidad de convergencia elevada y cierta dependencia de los valores iniciales, aleatorios, de las conexiones y de la arquitectura de red. Esto hace que sea conveniente entrenar varias redes con distintas valores iniciales y elegir la que mejor funcione. Esto tiene como consecuencia en algunos casos el aumento de tiempo para obtener soluciones sencillas. Por otro lado, al aumentar el tamaño de la red, hacen falta más ejemplos de aprendizaje lo que provoca un aumento mucho mayor del tiempo de aprendizaje. Además, al utilizar la técnica del gradiente descendiente se recorre la superficie del error hacia abajo hasta alcanzar un mínimo local, por lo que no se garantiza que se alcance una solución globalmente óptima. Sin embargo, se ha comprobado que el hecho de alcanzar mínimos locales no impide que se consigan resultados satisfactorios. En §2.2.5 se puede encontrar detallada la base teórica utilizada en la optimización matemática de la función del error que realiza este algoritmo.

Con el objetivo de mejorar la eficiencia del algoritmo original basado en la regla delta, expuesto en §2.3.3, se utilizarán los parámetros tasa de aprendizaje (η) y constante *momentum* (μ). El primero permite controlar la velocidad del aprendizaje, mientras que el segundo reduce el riesgo de inestabilidad durante la estimación de los pesos (cf. §2.3.3.3 para una información más detallada de estos parámetros). En un primer momento es más conveniente utilizar, a pesar de que se ralentice el aprendizaje, una tasa de aprendizaje pequeña y recurrir a inicializar los pesos de las conexiones con valores reducidos. La combinación del algoritmo RP clásico junto con η y μ se denominará, dentro de este trabajo, como algoritmo RP Extendido (RPE).

Durante años se han propuesto múltiples variantes del algoritmo original con el objetivo de mejorar los problemas de velocidad de convergencia y de inestabilidad, e.g. método del gradiente conjugado o algoritmos genéticos. Algunas de ellas hacían pequeñas modificaciones sobre el método del descenso del gradiente utilizado por el algoritmo RP, e.g. modificar la forma de actualizar los pesos, utilizar diferentes funciones de error o incluir ajustes dinámicos a los parámetros η y μ . Otras técnicas más complejas permiten mejorar la velocidad de convergencia mediante la utilización de derivadas de segundo orden, e.g. *Quasi-Newton*, *Levenberg-Marquardt* y *QuickPropagation*.

No obstante, cabe recordar que el algoritmo RP utiliza la derivada del error con respecto a cada peso, combinado con un valor de paso (*step-size*) fijo para adaptar dichos pesos. Este valor de paso sirve como referencia para ajustar el valor inicial y las tasas de incremento y decremento de los valores de los pesos durante el entrenamiento. Al utilizar la información almacenada en la segunda derivada del error respecto a cada peso, el tamaño del paso óptimo puede ser encontrado durante el entrenamiento de la red. Desafortunadamente, la segunda derivada del error respecto a cada peso resulta en un

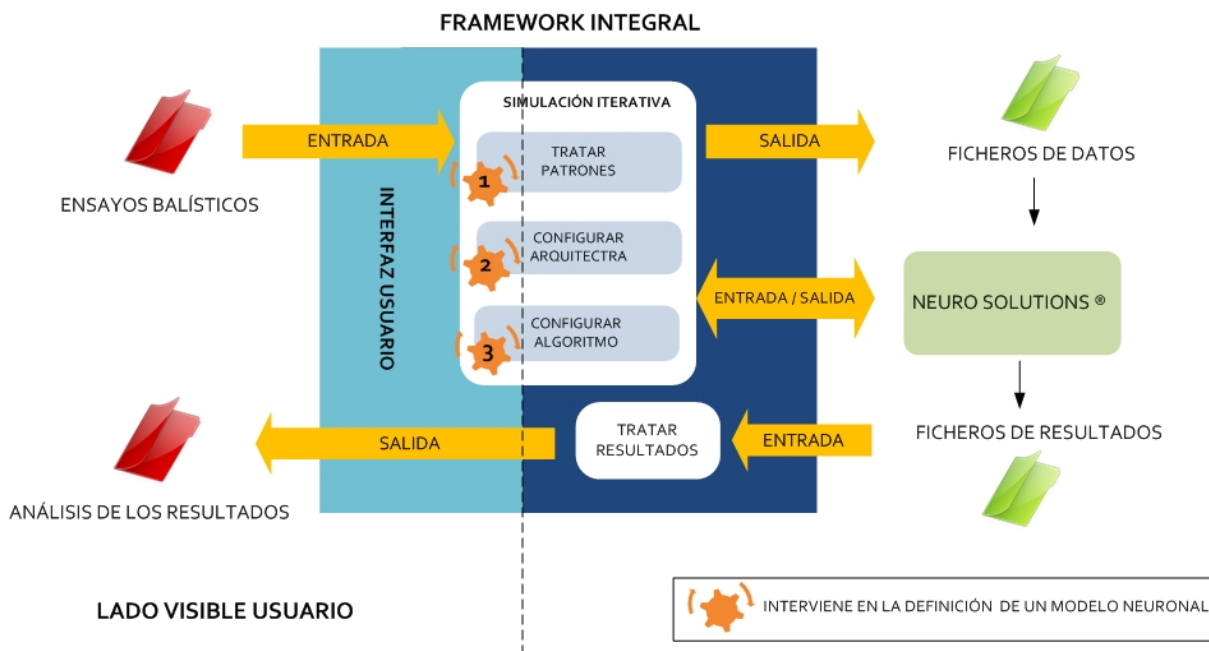


Figura 3.19: Arquitectura del *framework* desarrollado para automatizar las simulaciones

tensor⁶ de segundo orden que necesita ser resuelto, por lo que aumenta la complejidad del algoritmo. Por último, una aproximación diferente para el entrenamiento de un PMC es utilizar algoritmos evolutivos para entrenar los pesos.

Como algoritmo de partida para el entrenamiento y validación de los modelos neuronales se ha utilizado el algoritmo RPE. Las alternativas más interesantes al mismo que se han empleado para la optimización de los resultados y del rendimiento general de cada modelo aparecen incluidas en el Capítulo 4.

3.5.4. Herramienta para la configuración y automatización de las simulaciones

Uno de los problemas inherentes a esta investigación es la gran cantidad de simulaciones a realizar sobre los modelos neuronales para probar las diferentes técnicas y métodos existentes en cada uno de los escenarios balísticos planteados. Con el fin de reducir en la medida de lo posible los recursos y la complejidad de la pruebas a realizar se ha desarrollado un *framework* integral de simulación. Este *framework* incluye una serie de componentes y herramientas orientadas a configurar y automatizar el proceso de aprendizaje sobre diferentes tipos de RNA. Gracias a esto se facilita la capacidad de

⁶Un tensor es una clase de entidad algebraica de varios componentes, representada en forma de matriz, independiente de cualquier sistema de coordenadas elegido.

crear, desarrollar y evaluar series de experimentos de manera secuencial e iterativa, liberando al diseñador de la necesidad de realizar una a una cada una de las simulaciones.

En la Figura 3.19 se muestra cada uno de los componentes del *framework* diseñado y sus interacciones. De forma detallada la función de cada componente es:

1. Configuración de simulaciones iterativas: Este componente permite definir y configurar diferentes opciones relacionadas con los patrones, la arquitectura y el algoritmo de un modelo neuronal. Estas opciones pueden agruparse bajo una configuración de simulación, lo que permite que se puedan ejecutar de forma secuencial y transparente para el usuario múltiples pruebas. Gracias a esto, se puede analizar la influencia de cada una de las técnicas o métodos propuestos de forma individual o conjunta. Para reducir la influencia de la partición aleatoria de los datos, reducir la varianza en los resultados y aumentar la replicabilidad de las simulaciones cada configuración puede repetirse un número de veces fijado por el diseñador. Este componente puede ser desglosado en tres, relacionados cada uno con un término de la ecuación $modelo = patrones + arquitectura + algoritmo$:
 - Tratamiento de los ensayos balísticos: Define el tipo de problema a resolver (clasificación o regresión), así como las entradas y salidas del mismo. Además permite tratar los patrones disponibles mediante diferentes técnicas para su normalización, fijar los porcentajes de reparto para cada conjunto de datos e implementar distintas alternativas para maximizar la información disponible.
 - Configuración de la arquitectura de la RNA: Permite especificar la topología de la red, las funciones de activación de cada una de las capas, elegir diferentes criterios para inicializar los pesos de las conexiones y probar técnicas para la introducción de ruido sobre las entradas.
 - Configuración del entrenamiento: Permite especificar y configurar aquellos parámetros relacionados con el algoritmo de entrenamiento, definir las normas de la función de coste, las características del componente genético y elegir diferentes variantes del algoritmo de aprendizaje.
2. Herramienta de simulación de RNA: Para realizar el entrenamiento de cada modelo neuronal bajo una configuración determinada se ha recurrido a un software externo: NeuroSolutions© versión 5.03⁷.
3. Tratamiento de los resultados: Este último componente no interviene en la creación y configuración de un modelo neuronal, pero permite agrupar los resultados obtenidos para cada simulación realizada. Gracias a esto el diseñador solo tiene que consultar una salida del *framework* para evaluar, comparar y validar el modelo neuronal probado.

⁷<http://www.neurosolutions.com>

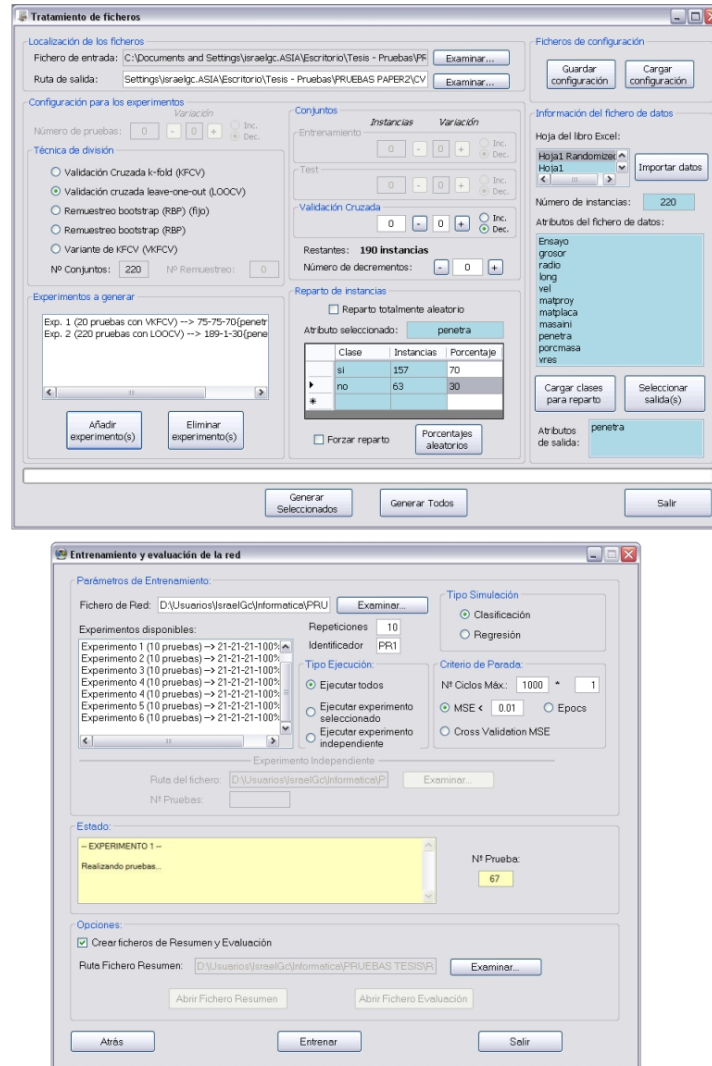


Figura 3.20: Detalles de la interfaz de usuario (arriba tratamiento de los datos y abajo entrenamiento iterativo)

En la Figura 3.20 se muestran dos detalles de la interfaz de usuario, en la zona superior aparece el componente de tratamiento de los patrones y en la inferior la pantalla para la configuración de una serie de pruebas y su posterior simulación.

3.6. Alternativas basadas en el aprendizaje automático

Una parte importante de esta investigación ha sido defender la necesidad de buscar otras soluciones, a las ya existentes, para recrear el fenómeno del impacto balístico. La elección de las RNA ha sido ampliamente justificada durante la investigación pero para

asegurar su viabilidad frente a otras alternativas se ha incluido un estudio comparativo que permita conocer la calidad de otras propuestas basadas en el aprendizaje automático.

En §2.7 se han detallado algunas de las variantes más interesantes dentro del ámbito de las maquinas de aprendizaje o aprendizaje automático: SOMFS, SVM, RBFN, RNN y ELN. Cada una de estas alternativas, junto con los PMC correspondientes, serán diseñadas con el objetivo de cubrir los diferentes escenarios del problema de impacto balístico descritos en §3.3 y comprobar la validez del modelo neuronal mostrado en §3.5.1.1 frente a ellas. El vector de entradas para todas ellas estará formado por 7 elementos, $V_{ent} = \{M_b, L_b, R_b, V_b, H_p, T_p, T_b\}$, y la salida a devolver será la perforación o no del blindaje $S_{clf} = \{P_b, D_b\}$ o bien la regresión de los valores para $S_{det} = \{M_{res}\}$ y $S_{per} = \{V_{res}, M_{res}\}$.

La evaluación realizada en esta tesis doctoral dentro del Capítulo 5 incluye una comparativa del rendimiento del PMC propuesto, frente al resto de técnicas de aprendizaje automático.

3.7. Sumario

El objetivo de este capítulo es definir la estructura de trabajo global, que permita a los ingenieros y científicos conocer el comportamiento de futuros nuevos proyectiles y protecciones, sin la necesidad de recurrir a las técnicas clásicas: experimentación, modelado analítico y simulación numérica. Para ello se han descrito diferentes situaciones relativas al impacto de sólidos, y su tratamiento desde un punto de vista multidisciplinar mediante la utilización de técnicas como la experimentación, simulación numérica y diversas alternativas basadas en el aprendizaje automático.

En primer lugar, para poder trasladar las características del fenómeno del impacto balístico a una herramienta de simulación numérica, ha sido necesario realizar una serie de ensayos experimentales en laboratorios. Gracias a estas pruebas empíricas, se ha podido verificar la precisión real de los modelos de simulación numérica mediante la comparación de los resultados obtenidos.

La utilización de técnicas basadas en la simulación numérica, para recrear el fenómeno del impacto entre sólidos, ha permitido obtener un conjunto de ensayos balísticos, los cuales serán utilizados para probar la capacidad de los modelos de PMC y otras alternativas basadas en la teoría de la máquina de aprendizaje. Dentro del dominio de este tipo de problemas, los datos disponibles incluyen parámetros referentes a la geometría de los sólidos involucrados, e.g. radio y longitud del proyectil, espesor de la placa metálica, al tipo de material utilizado, tanto en el proyectil como en la protección, y por último la velocidad del impacto. Para realizar el modelado y la simulación numérica

computacional se ha recurrido al software comercial ABAQUS/Explicit v6.4.1, un código de elementos finitos explícito, ampliamente empleado en la simulación de problemas dinámicos con un alto grado de no linealidad [ABAQUS, 2003].

Las técnicas de IA, donde se sitúan las RNA, permiten realizar una inducción del conocimiento inherente a la información que se le suministra. Por tanto, una vez realizado el aprendizaje, un modelo de RNA será capaz de resolver con alta precisión aquellas nuevas situaciones que se ajusten a las condiciones sobre las que ha aprendido. Para garantizar la fiabilidad de los resultados, todos los modelos neuronales serán validados para distintas geometrías de proyectil y de panel protector, así como distintas velocidades y materiales. Además de la comentada capacidad predictiva de estas técnicas, las RNA presentan, frente a los códigos de elementos finitos o diferencias finitas, una rápida respuesta a la hora de resolver nuevos ensayos de impacto. Mientras que la simulación de un problema de impacto puede conllevar varias horas de ejecución, la red neuronal, una vez realizado el proceso de aprendizaje, permite obtener el resultado en una fracción de segundo. No obstante, la definición de los modelos de PMC necesarios para resolver los escenarios planteados es el componente de inicio del marco integral diseñado, siendo necesario definir y aplicar una metodología que permita optimizar su funcionamiento. Por ello, a partir de dichos modelos, se definen en el Capítulo 4 los pasos necesarios con el fin de mejorar la capacidad de generalización y optimizar los resultados, tanto en tiempo de cómputo como en precisión.

Finalmente, para verificar la validez de los resultados alcanzados con los modelos neuronales, en el Capítulo 5 se comparará su rendimiento con el que ofrecen otros clasificadores de gran difusión y eficacia.

Capítulo 4

Metodología de Optimización de Modelos Neuronales

Según el diccionario de la RAE¹, una metodología es el conjunto de métodos que se siguen en una investigación científica o en una exposición doctrinal. Mientras que el método es el procedimiento para alcanzar los objetivos, la metodología es el estudio del método.

Partiendo de esta definición, la optimización de los modelos neuronales se ha llevado a cabo mediante la definición de una metodología de investigación, basada en una serie de actividades y tareas orientadas a cumplir los objetivos e hipótesis expuestas al comienzo de esta tesis doctoral. Para homogeneizar el gran número de alternativas existentes y simplificar su aplicación sobre cada uno de los escenarios, éstas se han agrupado en tres grandes bloques o fases: patrones, arquitectura y algoritmo. En cada uno de ellos se incluyen las variantes, que según la literatura analizada, mejor se adaptan al problema tratado.

En resumen, gracias a esta metodología, se posibilita la sistematización de los diferentes métodos y técnicas existentes a la hora de configurar y parametrizar un modelo neuronal basado en una topología PMC y algoritmo RP.

4.1. Introducción

En este capítulo se detallan las diferentes etapas de la metodología ideada para alcanzar los objetivos expresados en §1.4. Como ha quedado expuesto con anterioridad, las técnicas y métodos disponibles para parametrizar y configurar un modelo neuronal,

¹<http://www.rae.es>

basado en la topología PMC con algoritmo RP, pueden agruparse en tres categorías fundamentales: patrones, arquitectura y algoritmo. Partiendo de los modelos iniciales diseñados en §3.5.1 y del algoritmo RP clásico expuesto en §3.5.3, se aplicará esta metodología paso a paso hasta localizar los mejores modelos neuronales en cada uno de los escenarios.

En la Figura 4.1 se muestra la secuencia de pasos seguidos en la metodología para realizar una simulación completa. Se muestra gráficamente la secuencia de aplicación de las diferentes fases junto como sus interacciones y dependencias. La utilización de cualquiera de estos bloques es opcional durante la secuencia de optimización, ya que es posible que no sea necesario aplicar alguno de los métodos asociados. Esto da como resultado lo que se denomina configuración de prueba, estando formada por los siguientes elementos:

$$C_n = \{P_n \vee A_n \vee R_n\} \quad (4.1)$$

donde n es el número de configuración a probar, P_n el bloque de tareas relativo a los patrones, A_n , el bloque de tareas relativo a la arquitectura de red y R_n el bloque de tareas relativo al algoritmo RP.

Para cada una de las tareas dentro de estos bloques se procede a su entrenamiento y validación, para a continuación evaluar los resultados y medir la calidad del modelo neuronal. Si es necesario un nuevo reajuste, se puede iterar el proceso para evaluar una configuración diferente. Por tanto la metodología de optimización M está formada por la evaluación de un conjunto de configuraciones:

$$M = Eval\left(\sum_{n=1}^n C_n\right) \quad (4.2)$$

En los siguientes apartados se detallan las tareas asociadas a cada una de las fases o bloques de la metodología.

4.2. Fase Previa. Comparativa de alternativas al PMC

Una vez determinada la conveniencia de solventar problemas balísticos mediante otras propuestas diferentes a las tradicionales de simulación y modelado, esta etapa tiene como objetivo dilucidar cuál es la mejor solución dentro de las alternativas computacionales basadas en la teoría de las máquinas de aprendizaje. Para ello, se realizará un estudio comparativo que permita asegurar la viabilidad de la utilización de un PMC para este dominio concreto. Las alternativas que se han incluido para esta fase han sido detalladas

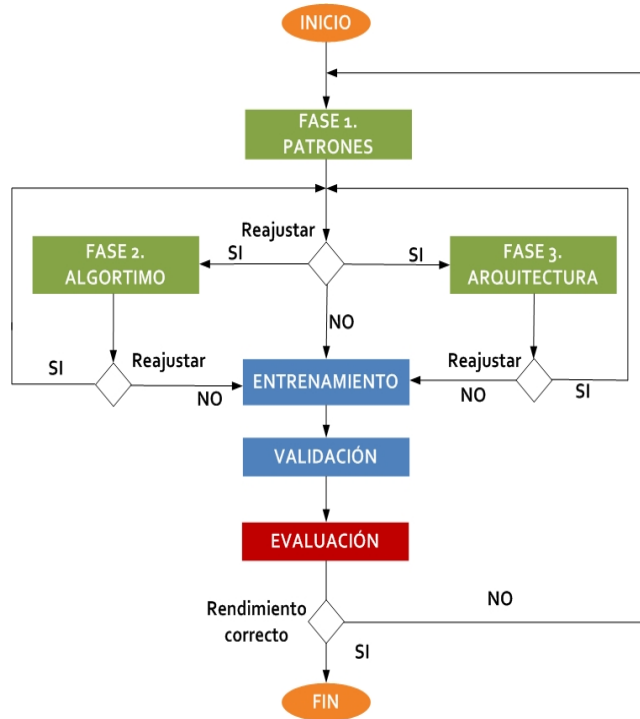


Figura 4.1: Pasos definidos para la metodología de optimización

en §3.6.

En la Figura 4.2 se muestra la ubicación de esta fase previa dentro del proceso de optimización. Una vez seleccionada la mejor alternativa de aprendizaje automático, se procede a aplicar sobre ella la metodología de optimización dividida en tres fases. Al finalizar, se evaluará la mejora obtenida respecto a la configuración básica de la máquina de aprendizaje seleccionada.

En la Tabla 4.1 se detallan las principales características de las alternativas implementadas. Todos las simulaciones realizadas tienen en común que los datos de entrada han sido tratados con una normalización *z-score* (cf. Ecuación 4.5) para mantener los valores usados dentro de un rango controlado. La regla de aprendizaje utilizará los parámetros constante *momentum* μ y tasa de aprendizaje η , siendo los valores aproximados a partir de [Lefebvre et al., 2007] los siguientes:

- SOFMS: $\eta=1$ y $\mu=0.7$ para la capa oculta. $\eta=0.1$ y $\mu=0.7$ para la capa de salida.
- SVM: $\eta=0.01$ para la capa de salida.
- RBFN: $\eta=1.0$ y $\mu=0.7$ para la capa de salida.
- RNN: $\eta=0.01$ y $\mu=0.7$ para la capa oculta. $\eta=0.1$ y $\mu=0.7$ para la capa de salida.

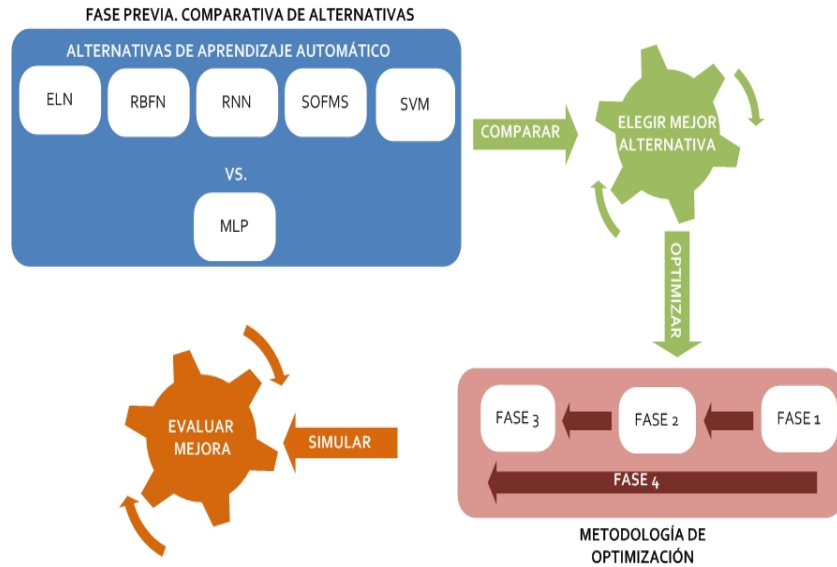


Figura 4.2: Detalle de la Fase Previa al proceso de optimización

- ELN: $\eta=0.01$ y $\mu=0.7$ para la capa oculta. $\eta=0.1$ y $\mu=0.7$ para la capa de salida.
- PMC: $\eta=1$ y $\mu=0.7$ para la capa oculta. $\eta=0.4$ y $\mu=0.3$ para la capa de salida.

Los criterios de parada, los porcentajes de reparto y la técnica de distribución de patrones utilizados serán indicados en cada uno de los escenarios.

En detalle, se ha construido una red neuronal híbrida con un módulo inicial del tipo SOFMS con aprendizaje no supervisado y un PMC con aprendizaje supervisado como sistema de salida. La capa SOFMS se elaboró con una capa de entrada de siete nodos conectada hacia delante (tipo *feedforward*) y una matriz de 5x5 neuronas para todos los datos de entrada. El algoritmo de aprendizaje empleó una función de vecindad del tipo Kohonen-cuadrado con radio de inicio y finalización de dos y cero respectivamente. A su vez, esta matriz constituyó la capa de entrada de una red de Perceptrones contigua, construida con la misma arquitectura que el PMC original (tres o cuatro neuronas en la capa oculta y dos en la de salida).

Para la red SVM el algoritmo de entrenamiento utilizado está basado en el algoritmo Kernel Adatron extendido a redes RBF [Anlauf and Biehl, 1989]. Este algoritmo puede ser fácilmente adaptado a esta clase de redes, mediante la sustitución del producto interior de los patrones en el espacio de entradas de la función kernel, llevando a la resolución de un problema de optimización cuadrático:

$$M = \min_i g(x_i) \quad (4.3)$$

siendo $g(x_i)$ la función de decisión a optimizar que utiliza el kernel Gausiano RBF.

Tabla 4.1: Fase Previa. Descripción de los clasificadores utilizados

Clasificador	Acrónimo	Descripción
Mapa AutoOrganizado	SOFMS	Función de vecindad Kohonen-cuadrado. Regla de aprendizaje <i>Momentum</i> . Topología 7-1-2 para clasificación y perforación en regresión. Topología 7-1-1 para detención.
Maquinas de Vectores de Soporte	SVM	Algoritmo Kernel Adatron. Paso 0.01. Topología 7-0-2 para clasificación y perforación en regresión. Topología 7-0-1 para detención.
Función de Base Radial	RBFN	Regla competitiva Conscience full. Métrica Euclídea. Topología 7-1-2 para clasificación y perforación en regresión. Topología 7-1-1 para detención.
Red de Elman	ELN	Entradas alimentan las unidades de contexto. Parámetro tiempo 0.8. Regla de aprendizaje <i>Momentum</i> . Topología 7-1-2 para clasificación y perforación en regresión. Topología 7-1-1 para detención.
Red Neuronal Recurrente	RNN	Parcialmente recurrente. Regla de aprendizaje <i>Momentum</i> . Topología 7-1-2 para clasificación y perforación en regresión. Topología 7-1-1 para detención.
Perceptrón Multicapa	PMC	Regla de aprendizaje <i>Momentum</i> . Topología 7-4-2 para clasificación y perforación en regresión. Topología 7-3-1 para detención.

Al tratarse de un problema de optimización matemática sobre una función, se pueden utilizar diferentes parámetros para mejorar la eficiencia del algoritmo. En este caso se ha incluido la tasa de aprendizaje η .

La red RBFN desarrollada tiene una capa oculta con funciones de activación Gaussianas y un segmento de aprendizaje no supervisado que es entrenado de forma independiente al aprendizaje supervisado. Para calcular el número de centros a emplear en la RBFN, el número de patrones en el conjunto de entrenamiento afecta de forma directa ya que más patrones implican más Gaussianas. Si los datos están bien agrupados entonces se necesitan menos Gaussianas, pero si están dispersos es necesario incluir un número mayor para lograr un desempeño fiable. Por tanto teniendo en cuenta que la elección de un valor correcto es dependiente del problema, para esta investigación se han seguido las indicaciones anteriores, empleando 35 Gaussianas. Además la red RBFN implementada aplica un aprendizaje competitivo con métrica euclídea. Esta métrica mide la diferencia entre los pesos y los vectores de entrada, de tal manera que se conserven las distancias para el espacio de entrada. El aprendizaje competitivo se encarga también de mantener la distribución de probabilidad intrínseca a los datos de entrada. No obstante, tiene el inconveniente de que algunas neuronas pueden no activarse mientras que otras lo hacen siempre. Para evitar estos dos extremos se ha incluido un mecanismo de conciencia, denominado *conscience full*, que mantiene un recuento de la frecuencia en que cada

neurona gana la competición, aplicando una tasa constante a través de ellas.

La red RNN diseñada tiene una capa oculta y una estructura parcialmente recurrente que añade una conexión *feedforward*, a través de la sinapsis, desde la entrada hasta la siguiente capa después de la capa oculta. En este caso, la estructura recurrente actúa como un estado de dicha estructura *feedforward*. Las estructuras de la capa de entrada están formadas por la función de identidad (f_{iden}) porque los datos de entrada son multidimensionales.

Por último, la red ELN implementada contiene una capa oculta y se presenta como una extensión al PMC con unidades contextuales que recuerdan las actividades pasadas pero con una disminución exponencial. Esto permite a la red extraer información temporal de los datos. Las unidades olvidan el pasado con el paso del tiempo a través de un factor denominado constante de tiempo. Los valores útiles se suelen encontrar entre en el rango [0-1], el valor más cercano a uno implica que la red recuerda más información y el factor de olvido es más lento. En esta investigación el valor empleado ha sido de 0.8.

4.3. Fase 1. Tratamiento de los patrones

Desde la aparición de las RNA, uno de los principales problemas que los investigadores han encontrado, es su aplicación en dominios donde la información existente es reducida. Si bien en algunos trabajos se ha determinado que las RNA pueden enfrentarse a problemas con datos limitados, presentando una mejor capacidad predictiva que las técnicas de análisis discriminante clásicas [Hepner et al., 1990], otros autores han optado por buscar alternativas provenientes de técnicas estadísticas que permitan determinar los requisitos de información necesarios para asegurar la robustez del modelo neuronal o ampliar el conjunto de datos disponibles.

Otra técnica relacionada con el procesado de los patrones, en este caso pre-procesado, es la normalización de los datos. Esta ha demostrado ser, dentro de las RNA, un paso útil para afinar los resultados y reducir el tiempo de cálculo. En muchos casos permite agilizar el proceso de entrenamiento y su eficiencia ya que los datos que se le pasan a la red se encuentran próximos a la función de transferencia de la neurona [Sola and Sevilla, 1997].

Por otro lado, la correcta elección de las características, tamaño y reparto de las clases en los conjuntos de datos determina la precisión final del aprendizaje [Foody et al., 1995]. Es sabido que la precisión de un modelo neuronal incrementa en función del tamaño del conjunto de entrenamiento, hasta un punto donde en el cual no se experimenta mejoras considerables. No obstante, como contrapartida a medida que aumenta este conjunto se incrementa el coste y el tiempo asociado al proceso de desarrollo del modelo neuronal. Por tanto, estas consideraciones hacen necesario determinar la relación entre

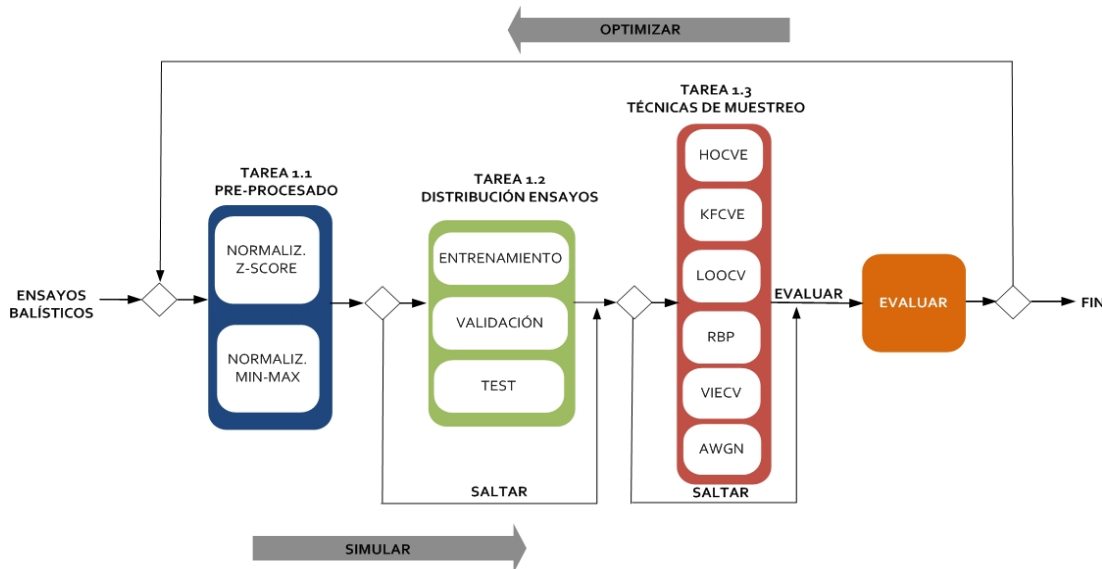


Figura 4.3: Detalle del proceso de optimización asociado a la Fase 1: Tratamiento de los patrones

los conjuntos de datos utilizados y la capacidad predictiva de la red, localizando el punto de inflexión en su rendimiento en función del reparto de ensayos para cada conjunto.

En la Figura 4.3 se muestran las diferentes tareas pertenecientes a la fase del tratamiento de patrones del proceso de optimización. Las diversas técnicas disponibles han sido englobadas en tres grandes bloques para homogeneizar su utilización. El propósito de esta fase consiste en determinar la mejor configuración posible, mediante la evaluación, individual y conjunta, del rendimiento de las alternativas asociadas a cada tarea. En los siguientes apartados se concretan los objetivos y se justifican cada una de estas técnicas.

4.3.1. Tarea 1.1: Normalización de los patrones

Al analizar los ensayos balísticos obtenidos se puede observar que éstos están expresados en diferentes unidades, apareciendo grandes variaciones de magnitud en sus valores. Esto puede producir que la red no será capaz de alcanzar los resultados más precisos y que el tiempo de cómputo sea demasiado elevado. Para solventarlo una de las herramientas más empleadas por los diseñadores de sistemas de reconocimiento automático es la normalización de los datos. Su utilización permite reducir la reducción de los efectos asociados a los valores frontera y la aceleración de la convergencia del entrenamiento, al reducir el espacio de búsquedas sobre el que tiene que actuar el algoritmo RP.

Si bien existen muchos tipos de normalización, todas buscan homogeneizar las variables de entrada con un mismo rango de valores para minimizar la influencia de una variable sobre el resto dentro de la red. Por tanto, la normalización es conveniente para modelar aplicaciones donde las entradas se dividen en un rango de escalas extenso. En esta investigación se han incluido los dos tipos de normalización más utilizados, *min-max* y *z-score*, dependiendo su empleo del criterio del diseñador de la red y de las características de los datos.

La normalización *min-max* es la técnica más simple y es útil cuando se quiere limitar los valores de las entradas y/o salidas a un rango concreto. En este caso los valores mínimos y máximos se desplazan hasta cero y uno, respectivamente, por lo que todas las variables se transforman al rango $[0, 1]$, de manera que la distribución original se mantiene excepto para el factor de escala. En algunos casos se opta por incluir valores negativos ampliando el rango hasta $[-1, 1]$. Este proceso de escalado se consigue normalmente mediante la utilización de una interpolación lineal del tipo:

$$x'_i = (max_{target} - min_{target}) * \left[\frac{(x_i - min_{value})}{(max_{value} - min_{value})} \right] + min_{target} \quad (4.4)$$

donde $(max_{target} - min_{target}) \neq 0$. Cuando en una de las variables se cumple $(max_{target} - min_{target}) = 0$ indica que el valor de esa variable es constante. Cuando sucede esto, se podría eliminar del entrenamiento porque no aporta información adicional a la red. Como puede observarse en la Ecuación 4.4 se calculan los valores máximos y mínimos para cada variable, y los datos son transformados para caer dentro del rango deseado de valores $(min_{target}, max_{target})$. Cuando se aplica la normalización *min-max*, los valores de cada variable estarán acotados dentro de un nuevo rango, pero la distribución subyacente de cada variable se mantiene igual. Esto permite al diseñador más flexibilidad a la hora de crear los modelos neuronales y determinar que características son de mayor importancia para poder tomar decisiones. Esta normalización tiene la ventaja de preservar exactamente las relaciones inherentes a los datos y no introduce ningún tipo de influencia sobre ellos.

Por su parte la normalización *z-score* es la técnica más utilizada y consiste en transformar la distribución original de las variables en una distribución de media cero y variancia unitaria.

$$x'_i = \left[\frac{(x_i - \mu_i)}{\sigma_i} \right] \quad (4.5)$$

La técnica de normalización *z-score* usa dos indicadores estadísticos de los datos, la media (μ) y la desviación típica (σ) de cada variable a lo largo del conjunto de datos de entrenamiento. Con esto se consigue normalizar cada vector de entradas a un rango controlado. Para su empleo se calcula μ y σ para cada variable y se aplica para cada vector la Ecuación 4.5. Esto produce datos donde cada variable tiene una media cero y

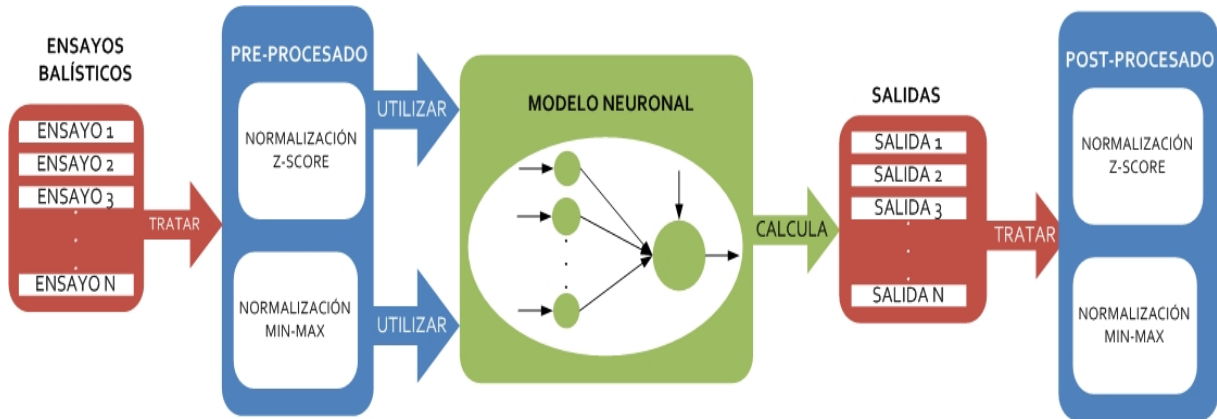


Figura 4.4: Ubicación de la capa de procesado (pre y post-procesado) de los ensayos dentro del marco de trabajo

varianza unitaria.

Una vez que los valores para μ y σ han sido calculados para cada entrada x_i del conjunto de datos, se conservan sus valores para utilizarlos como pesos de los futuros datos a utilizar en el diseño final de la red, datos de validación o test. De otra forma su rendimiento se verá afectado significativamente a la hora de validar su rendimiento, ya que ésta ha sido entrenada sobre una representación de datos diferente a la de los datos sin normalizar.

En esta investigación se aplicarán ambos tipos de normalización a los datos de entrada, dando lugar a un conjunto de datos de entrenamiento nuevo, el cual se utiliza para entrenar la red. Como se observa en la Figura 4.4, la normalización se utiliza por tanto como una capa de preprocesado antes de presentar los datos a la estructura de la red cuando fuera necesario. Una vez que se obtienen las salidas, se sobrepone una capa de postprocesado donde se reescalan los valores de las salidas para que coincidan con el dominio balístico. En el caso de una normalización *min-max* los valores se devuelven al rango original, mientras que si se trata de una normalización *z-score* se cambian los valores unitarios por la escala real. La aplicación de la normalización será necesaria dependiendo de las características de la topología diseñada, i.e. función de activación de las neuronas.

Por último, reseñar que se trata de una tarea transversal que siempre se va incluir en cada modelo neuronal diseñado. Por lo tanto, al menos un criterio de normalización de los pesos será aplicado en todas las simulaciones de esta metodología.

4.3.2. Tarea 1.2: Distribución de los patrones

En el dominio de esta investigación la cantidad de datos disponibles es relativamente pequeña debido a la complejidad asociada a su simulación o experimentación. Por tanto la división de estos ensayos entre los diferentes conjuntos de *train*, test y validación o CV, es crucial ya que afecta de forma clara a los resultados obtenidos por la red.

El objetivo de este apartado es investigar la relación entre el tamaño de los diferentes conjuntos y su influencia sobre capacidad predictiva de los diferentes modelos neuronales. Para ello se busca localizar la mejor distribución de ensayos entre los diferentes conjuntos para cada uno de los modelos.

Como ha quedado claro en §2.4.2.4, la distribución típica es 33/33/33 cuando se emplea parada temprana, mientras que si sólo se utiliza *train* y test el reparto habitual es 50/50 o 80/20 en función del autor de la investigación. Partiendo de la base de que es necesario un tamaño adecuado para que la red converja y no aparezca saturación, algunos autores recomiendan usar conjuntos muy grandes para asegurar resultados aceptables, pero se trata de recomendaciones ad-hoc o subjetivas que dependen de su propia experiencia. Sin embargo, es necesario tener en cuenta que aumentar el tamaño del conjunto de entrenamiento conlleva más tiempo de computación y por tanto el coste. Debido a esto, no existe una forma clara de conocer la mejor distribución posible, por lo que se recurre en muchos casos a pruebas de fuerza bruta o intentar adaptar premisas de otros investigadores. Una de estas premisas de uso general, es que el conjunto de entrenamiento debe ser de mayor extensión que el test para compensar las carencias del proceso de entrenamiento.

Para guiar las diferentes simulaciones destinadas a optimizar los modelos neuronales se adaptarán las pautas expuestas por Boonyanunta y Zeephongsekul [Boonyanunta and Zeephongsekul, 2004] y Crowther y Cox [Crowther and Cox, 2005] en sus respectivos trabajos. El algoritmo diseñado por los primeros ha demostrado que existen distribuciones con mejores resultados que la clásica 50/50 y de menor riesgo que 80/20. Por otro lado el modelo matemático propuesto por los segundos explica la relación entre el conjunto de *train* y la capacidad de predicción de la red, lo que permite determinar su tamaño óptimo.

La metodología propuesta a partir de estos trabajos consiste en realizar múltiples simulaciones, hasta encontrar el umbral de eficiencia de la red. Para ello se probarán diferentes configuraciones de reparto, con variaciones del 1% sobre del conjunto de *train* dentro del rango [10%, 90%] y [10%, 80%] cuando se utilice el conjunto de validación o CV. Gracias a la investigación de Boonyanunta y Zeephongsekul, se sabe que el comportamiento típico asociado al rendimiento predictivo de un RNA es el mostrado en la Figura 4.5. Inicialmente al aumentar el tamaño del conjunto de entrenamiento aumenta el poder predictivo de forma rápida, desde cero hasta el punto *A*, y a continuación empieza a decelerar, desde el punto *A* hasta *B*. En la última etapa, desde *B* hacia delante, no se

observan mejoras significativas produciéndose un estancamiento. Por tanto, el objetivo es localizar este punto B ya que se puede considerar como el umbral de eficiencia de la red.

A partir de estudios empíricos se ha demostrado la relación existente entre el número de patrones (p) y la capacidad predictiva de una RNA para un conjunto p de patrones ($P(p)$), siendo expresada como:

$$\frac{\partial P(p)}{\partial p} = k(T - P(p)) \quad (4.6)$$

donde T es el umbral de eficiencia y k la tasa de eficiencia o tasa de mejora en la capacidad de predicción por unidad de aumento de la eficiencia. Gráficamente el aumento de esta capacidad de predicción en función de p puede verse en la Figura 4.5 (adaptada de [Crowther and Cox, 2006]). Resolviendo la ecuación diferencial se obtiene:

$$P(p) = T(1 - e^{-kp}) + P(0)e^{-kp} \quad (4.7)$$

donde $P(0)$ es el poder predictivo cuando no se facilitan datos de entrenamiento al clasificador, por tanto se espera que $P(0) > 0.5$ ya que su rendimiento en base a la teoría probabilística clásica debe ser al menos superior a la probabilidad de obtener cara o cruz al tirar una moneda.

Este modelo matemático ha sido validado por Crowther y Cox [Crowther and Cox, 2006] para varios conjuntos de datos, reales y experimentales, ratificando la viabilidad del modelo para resolver problemas de clasificación generales dentro del ámbito de las RNA. Debido al desconocimiento y a la dificultad asociada al cálculo de los parámetros k , T y $P(0)$, estos valores se ajustan, a partir de los datos disponibles, mediante un método de optimización no lineal basado en el descenso del gradiente.

Por último, es necesario recalcar que debido a que existe cierta subjetividad en determinar que representa un nivel de mejora insignificante, se deben tener en cuenta ciertos requisitos del dominio de esta investigación como el coste de extracción de los datos o el posible aumento del tiempo de computación.

4.3.3. Tarea 1.3: Técnicas para trabajar con datos limitados

En algunos escenarios debido a las características del proceso que recrea los datos, la obtención y etiquetado de información reseñable se convierte en un proceso intensivo y complicado. En estos casos la cantidad de datos disponibles que el diseñador de una RNA puede utilizar es limitada. Además el entrenamiento y la validación de cada red necesitan la división del conjunto original en una serie de conjuntos disjuntos, lo que reduce todavía

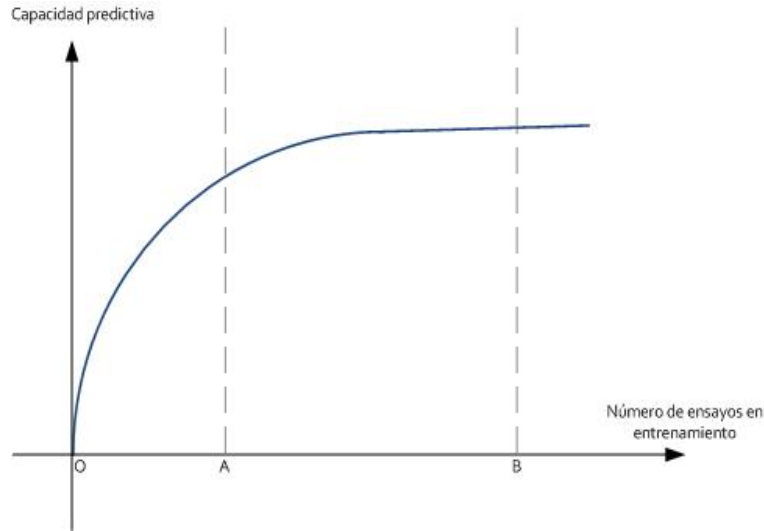


Figura 4.5: Relación entre el tamaño del conjunto de *train* y la capacidad de predicción de un modelo neuronal

más la cantidad de datos disponibles. En el dominio de esta investigación, la obtención de cada ensayo conlleva el uso de grandes recursos y un elevado coste computacional por lo que el conjunto total disponible es de dimensiones reducidas. En §3.4 ha quedado detallado el proceso de obtención de los ensayos balísticos y en el Anexo B se muestra el listado detallado de los mismos.

Para mejorar los resultados sin necesidad de obtener más ensayos balísticos, se han analizado diversas técnicas provenientes del ámbito estadístico, utilizadas en problemas de estimación y clasificación, que permiten el remuestreo del conjunto original aumentando su tamaño. Al tener un claro componente estadístico y probabilístico, las RNA sacan provecho de estas técnicas en aquellas situaciones en las cuales la cantidad de información no permita asegurar unos resultados fiables [Priddy and Keller, 2005]. Al igual que los estadísticos las utilizan en sus problemas para estimar el error de generalización del modelo y elegir su estructura, su aplicación en esta investigación va a permitir elegir realizar múltiples simulaciones para localizar los mejores parámetros, definir la importancia de las variables de entrada y seleccionar la arquitectura óptima.

Las técnicas más conocidas son la validación cruzada y el *bootstrapping*. Se trata de dos alternativas para estimar el error de generalización obtenido por un clasificador a partir de lo que se conoce como remuestreo (*resampling* en terminología inglesa). Esta técnica facilita, por un lado la elección entre diferentes modelos, en el ámbito de las RNA diferentes arquitecturas de red [Weiss and Kulikowski, 1991] y por el otro su empleo ayuda a obtener mejores resultados cuando los datos disponibles son reducidos [Goutte, 1997].

En el primer caso la elección del mejor modelo de RNA se realiza detectando el menor error de generalización obtenido para cada una de las alternativas. Sus aplicaciones más comunes son la estimación del número óptimo de neuronas ocultas o la elección de un subconjunto de variables de entrada. Este punto presenta una clara relación con la segunda de las aplicaciones, el entrenamiento con pocos datos, ya que al elegir un subconjunto menor de variables de entrada, se reduce la complejidad de la red lo que conlleva que se necesite menos información para realizar el aprendizaje.

En los siguientes apartados se detallan las heurísticas derivadas de estas dos técnicas, y que han sido incluidas en esta investigación con el objetivo de estudiar su influencia en el rendimiento de los modelos neuronales.

4.3.3.1. Validación Cruzada *Hold Out* (HOCV)

Esta técnica estadística es la utilizada generalmente para distribuir los datos en RNA debido a su sencillez y velocidad de cálculo. A partir del conjunto completo de datos disponibles se crean dos subconjuntos disjuntos o excluyentes, el primero de ellos aplicable a la fase de entrenamiento y el segundo para la fase de prueba. Una variación del mismo permite generar un nuevo conjunto CV para usarlo durante el entrenamiento. Este proceso se puede observar en la Figura 4.6, utilizándose dos de los conjuntos durante el entrenamiento (*train* y validación), y el otro (test) para probar la arquitectura de la red y determinar el error de generalización. En su forma básica, su aplicación no implica altos costes computacionales al realizarse una sola iteración. Para mejorar su capacidad es necesario asegurar una correcta elección de la distribución de los datos entre los conjuntos y que todas las clases estén representadas de forma equitativa en cada uno de los mismos. Si una clase no está representada en los datos de entrenamiento, es de esperar que el modelo no se desempeñe bien ante nuevos en datos de esa clase.

De forma estricta la alternativa HOCV, no se considera como una técnica de validación cruzada porque los datos en realidad nunca son cruzados. Sin embargo, tomando los elementos al azar y repitiendo el procedimiento algunas veces (más de diez), la estimación del error total, es aceptable gracias al remuestreo o *resampling*. Esta variante se conoce como HOCV Iterativa (HOCVI) y el error estimado se calcula como el promedio de los errores obtenidos en cada iteración. Una segunda alternativa para mejorar la eficiencia de esta técnica se conoce como HOCV Equilibrado o Estratificado (HOCVE). En este caso las clases aparecen con la misma frecuencia en las particiones que en el conjunto original lo que reduce la varianza del clasificador.

En la mayoría de las ocasiones el reparto es de 2/3 de los datos para el entrenamiento y 1/3 para testear la red. En el caso de existir el conjunto CV el reparto pasa a ser 1/3, 1/3, 1/3. Esta técnica es un estimador estadístico pesimista ya que solo una porción de los datos se utiliza para entrenar. Cuantas más instancias se dejen para el conjunto de test, es posible obtener un mayor margen de error en las estimaciones, sin embargo si se

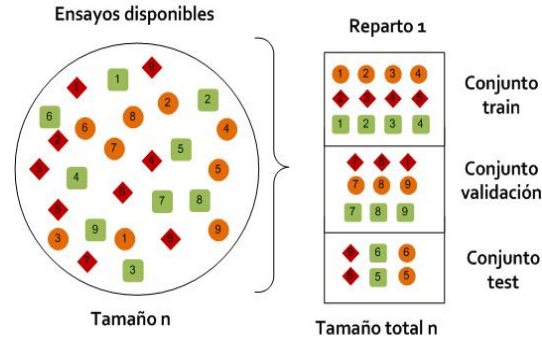


Figura 4.6: Proceso de creación de conjuntos con HOCV

utilizan pocas el intervalo de confianza aumenta [Kohavi, 1995].

Cabe mencionar que, cuando se tienen pocas instancias disponibles o existe alta dispersión en los datos, puede presentarse alta varianza en la estimación del error total; consecuentemente, es recomendable aplicar criterios estadísticos adicionales ya que el resultado depende fuertemente de las instancias que hayan sido seleccionadas para formar el conjunto de entrenamiento.

4.3.3.2. Validación Cruzada *K-Fold* (KFCV)

La validación cruzada *k-fold*, propuesta por Stone en 1974 [Stone, 1974], es una variante más compleja de la técnica anterior. Se trata de la alternativa más empleada para minimizar los efectos de la dispersión de los datos y las debilidades inherentes a la técnica HOCV. La principal diferencia radica en que HOCV solo utiliza un subconjunto de datos, conjunto de test, para estimar el error de generalización durante el entrenamiento, en lugar de los k conjuntos de KFCV. Debido a esto muchos investigadores no tienen en cuenta esta primera propuesta como una de validación cruzada, al no haber realmente ningún cruce de datos entre los subconjuntos. Además, con la alternativa HOCV se produce un evidente desperdicio de datos, ya que no se utilizan todos los datos para entrenar, al destinar un parte de ellos a comprobar el error de generalización.

Esta técnica implica la partición de los datos en k conjuntos separados, donde k es el número de conjuntos elegidos. Esta separación se suele realizar generalmente de forma aleatoria en k disjuntos de un tamaño similar. Dicho proceso se puede observar en la Figura 4.7, utilizándose uno de los conjuntos para testar la arquitectura de la red y determinar el error de generalización y el resto para entrenarla. El conjunto de ensayos de entrenamiento se utiliza para entrenar la red y se puede dividir según el experimento a realizar en entrenamiento y validación.

Este proceso se repite tantas veces como conjuntos se hayan generado, i.e. k , hasta que todos los conjuntos se han utilizado para entrenar y validar la red. De esta forma, se tienen K estimaciones de error E_i , que serán promediadas para obtener el error total de predicción E del modelo en cuestión:

$$E = \frac{1}{k} \sum_{i=1}^k E_i \quad (4.8)$$

En los problemas de estimación estadística, de donde proviene la teoría de esta propuesta, se suele utilizar un valor de k igual a diez [Kohavi, 1995]. En RNA no existe una regla de uso general, pero se suele obtener resultados correctos con valores entre $[5 - 10]$, por lo que será este el rango elegido en las simulaciones. Una variante de esta técnica se conoce como KFCV Equilibrada (KFCVE). En este caso cada uno de los conjuntos contiene un porcentaje de clases similar al del conjunto original, lo que reduce la varianza del clasificador.

Una de las ventajas que tiene esta técnica es que, aún cuando se disponga de pocas instancias para ambas fases (entrenamiento/prueba), la estimación total del error E , es confiable debido a que todos los patrones son utilizados durante la fase de prueba, en otras palabras, todos los patrones se emplean una vez en la fase de prueba y $k - 1$ veces en la fase de entrenamiento. Además facilita una estimación del error poco sesgada y la varianza en la estimación del error de predicción disminuye conforme el valor de k aumenta. Las desventajas de esta técnica son que la fase de entrenamiento tiene que ejecutarse en k ocasiones, lo que implica k veces más tiempo de cómputo que la técnica de HOCV, y que la estimación del error puede ser variable en algunas situaciones. No obstante, la principal motivación que ha llevado a utilizar la técnica KFCV, y las siguientes variantes en esta investigación, es su clara ventaja a la hora de trabajar sobre conjuntos de datos reducidos [Goutte, 1997].

Por último, para paliar la influencia de las posibles particiones aleatorias esta técnica puede repetirse varias veces, lo que implica dividir los datos en k conjuntos y repetir la técnica n veces. Se suele denominar $n * kcv$ [Dietterich, 1998] y un ejemplo típico es el que se repite diez veces sobre diez conjuntos de datos: $10 * 10cv$.

4.3.3.3. Validación Cruzada *Leave-One-Out* (LOOCV)

Esta variante de validación cruzada puede verse como la versión de KFCV cuando k es igual al número n de instancias disponibles. Esto implica que, para cada una de las n veces que se realice la fase de aprendizaje, será necesario considerar $n - 1$ instancias para entrenamiento y solo una instancia para prueba, de modo que, se obtienen n estimaciones parciales de error. Análogamente, se puede calcular el error total en la precisión predictiva usando la siguiente expresión:

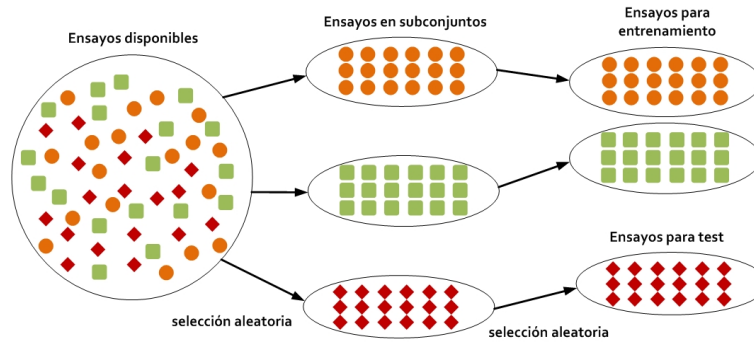


Figura 4.7: Proceso de selección aleatoria para k subconjuntos con KFCV y parada temprana

$$E = \frac{1}{n} \sum_{i=1}^N E_i \quad (4.9)$$

Una de las ventajas de esta propuesta es que el proceso es determinista y en todo momento se usa el máximo posible de datos para la inducción del clasificador. Además, el problema de la alta varianza en la estimación del error total E , derivado ya sea de un número reducido de instancias disponibles o de la dispersión en los datos, se vuelve casi despreciable. Esto no sorprende, ya que conforme crece el número de particiones, decrece el tamaño de cada una de éstas, haciendo que el número de instancias involucradas en la fase de aprendizaje sea menor, consecuentemente, la varianza tiende a disminuir. Contrario a lo que se pudiera pensar, la precisión predictiva de un clasificador no gira en torno a una mayor cantidad de instancias procesadas en la fase de aprendizaje. Algunas investigaciones, sugieren que a medida que aumenta la cantidad de instancias conocidas durante el entrenamiento del clasificador, pueden surgir comportamientos no deseados en la fase de recuperación, i.e. sobreaprendizaje, por lo que únicamente se observará comportamiento adecuado con los patrones conocidos, pero no se asegura que la precisión predictiva sobre nuevas instancias sea la misma [Kohavi, 1995].

Esta versión es similar a KFCV pero solo utiliza un dato para el conjunto de test. Por tanto, es igual a la propuesta KFCV cuando k es igual al tamaño del conjunto de datos, excepto en que consume más recursos que el anterior, ya que necesita utilizar todos los patrones como conjunto de test y se ejecuta tantas veces como patrones tenga el conjunto original, n .

En la Figura 4.8 se muestra un ejemplo de selección aleatoria de ensayos con LOOCV. Uno de los ensayos es destinado al conjunto de test, mientras que el resto se envían al conjunto de entrenamiento. A su vez este conjunto puede dividirse, según las necesidades del experimento, a realizar el entrenamiento y la validación.

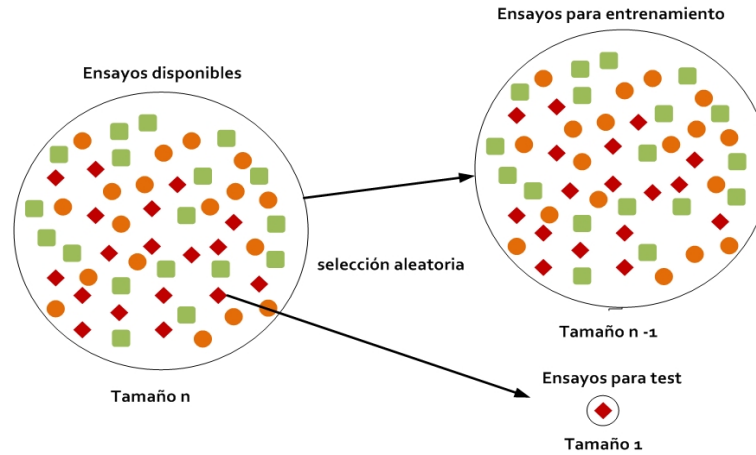


Figura 4.8: Proceso de selección aleatoria con LOOCV

LOOCV presenta como inconveniente, a pesar de sus buenos resultados en ciertos casos, su dificultad de aplicación como seleccionador de modelos. En este caso aparece una falta de continuidad, ya que un pequeño cambio en los datos puede causar una gran perturbación sobre el modelo elegido [Breiman, 1996]. Por otro lado para seleccionar subconjuntos de variables de entrada, Breiman y Spector demostraron en 1992, que KFCV con cinco o diez divisiones presenta mejores resultados que LOOCV en árboles de decisión empíricos (clasificadores C4.5). Además valores de k pequeños, como cinco o dos, funcionan bien en ciertos dominios si se emplean varias divisiones aleatorias de los datos para reducir la variabilidad de la estimación obtenida mediante validación cruzada.

4.3.3.4. Remuestreo *Bootstrap* (RBP)

La técnica de bootstrapping, inventada por Efron en 1979 [Efron, 1979], es una de las más precisas para trabajar con un conjunto de ejemplos reducido. Para ello necesita la creación de un número grande de nuevos ejemplos a partir del conjunto original, remuestreando los datos disponibles para crear una distribución de muestreo. En consecuencia, se entiende por muestreo el proceso de seleccionar un número de observaciones de un grupo en particular de la población. Con esto se asume que la nueva distribución es representativa de la población completa. En estadística se lleva aplicando con éxito durante más de 25 años y los diseñadores de RNA las utilizan desde finales de los años 90.

A partir del conjunto original de n elementos, se seleccionan aleatoriamente b elementos con repetición para producir un nuevo subconjunto de tamaño n . Este proceso se repite varias veces, dando lugar a varios conjuntos de datos diferentes. Los datos que no han sido destinados al conjunto de entrenamiento forman parte del conjunto de test,

mientras que todos los que han sido seleccionados n , con sus repeticiones se utilizan para entrenar la red. Por tanto, el número de patrones destinados a test puede variar, pero como resultado se consigue un conjunto de datos de mayor tamaño en entrenamiento y validación.

Bootstrapping se diferencia de las técnicas anteriores de validación cruzada, KFCV y LOOCV, en que se trabaja sobre subconjuntos de muestreo y no sobre subconjuntos de datos. Además los conjuntos de muestreo tienen el mismo tamaño que el conjunto original, duplicando algunos patrones mientras otros son rechazados. Si se comparan ambas técnicas, este remuestreo aumenta la varianza que puede aparecer en cada conjunto, pero *bootstrapping* facilita una representación más precisa del mundo real del problema [Efron and Tibshirani, 1994]. Para ello preserva las probabilidades a priori de las clases a través del proceso de selección aleatorio. Por último, esta técnica se suele utilizar cuando el conjunto de datos disponible es reducido y no se puede asegurar la fiabilidad de los resultados que se obtienen, además en muchos casos esta alternativa ha mostrado mejor comportamiento que la validación cruzada [Efron, 1983].

En la Figura 4.9 se muestra un ejemplo con dos repartos de ensayos realizados con *bootstrapping* para entrenamiento y test. A su vez el conjunto de entrenamiento se puede dividir según el experimento a realizar en entrenamiento y validación

Como se ha comentado la familia *bootstrap* fue introducida por Efron a finales de los años 70, pero su demostración matemática más utilizada se encuentra descrita en [Efron and Tibshirani, 1994]. Si se tiene un conjunto de datos de tamaño n y se quiere generar un nuevo subconjunto *train* de n muestras, es necesario realizar un remuestreo con repeticiones. Por consiguiente, la probabilidad de que un dato concreto sea elegido entre n es $\frac{1}{n}$. De igual forma, la probabilidad de no serlo es $1 - \frac{1}{n}$. Si se realizan n selecciones la probabilidad de que un ejemplo no sea seleccionado es $(1 - 1/n)^n \approx e^{-1} \approx 0.368$. En definitiva, el número esperado de instancias distintas de la original destinadas al conjunto de test es $0.632n$, siempre que las que se utilicen para entrenar no se incluyan durante la fase de prueba de la red [Kohavi, 1995].

En el trabajo de Efron del año 1983 [Efron, 1983], se compararon mediante experimentos sobre conjuntos reducidos diferentes variantes de validación cruzada, como KFCV y LOOCV, con las técnicas de la familia *bootstrap*. Los resultados más significativos mostraron que RBP funciona mejor que las alternativas derivadas de la validación cruzada en muchos experimentos. Además LOOCV obtiene buenas aproximaciones a los valores reales, pero con una alta variabilidad, especialmente con pocos datos, mientras que *bootstrap* .632 se comporta mejor para las mismas circunstancias.

En el caso más sencillo de *Bootstrapping*, en lugar de analizar repetidamente subconjuntos de datos, se analizan remuestreos de datos. Cada patrón utilizado es un ejemplo aleatorio con repetición a partir del conjunto original. Dependiendo del dominio,

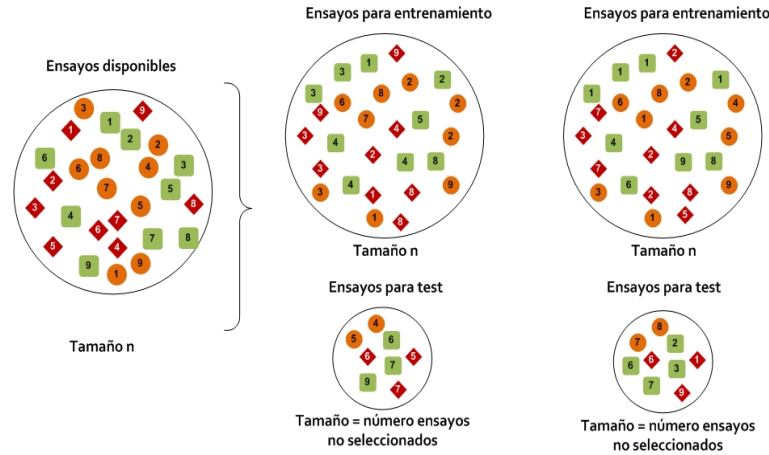


Figura 4.9: Proceso de selección aleatoria con *bootstrapping*

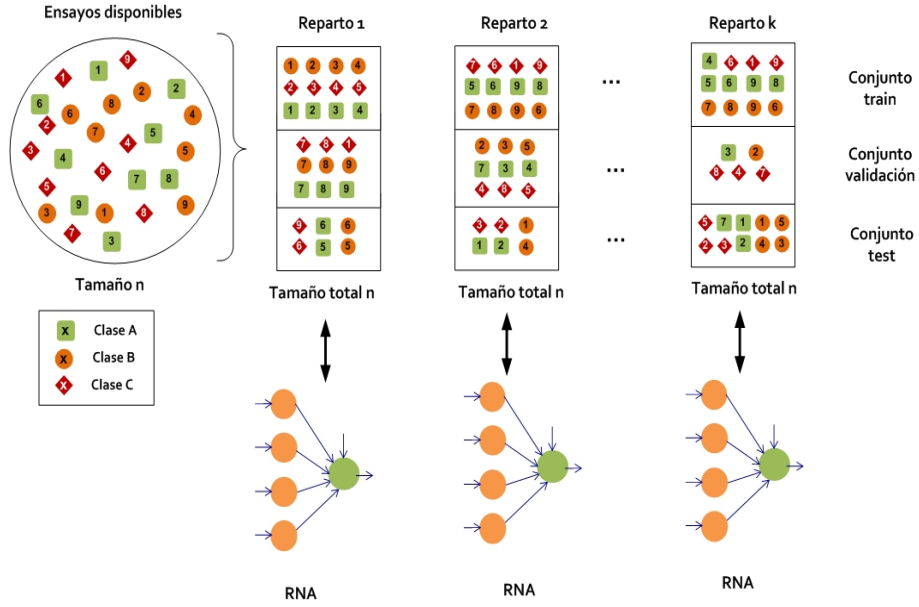
el número de ejemplos obtenidos por muestreo suele oscilar entre 20 y 2000 [Tibshirani, 1996], siendo necesario reentrenar la red tantas veces como remuestreos B se realicen: $20 \leq B \leq 2000$. Según Tibshirani, para utilizar esta técnica en RNA, debido a las características del modelo que representa, el número más adecuado es el límite inferior, i.e. 20 muestreos nuevos. La estimación del error total, E , se realiza mediante la siguiente ecuación:

$$E = \frac{1}{B} \sum_{i=1}^B E_i \quad (4.10)$$

Para la estimación del error de generalización en RNA, se ha recurrido a la variante .632+ de *bootstrap* ya que se trata de un estimador con poco sesgo y varianza inferior a KFCV. Además, se comporta de forma correcta cuando aparecen graves problemas de sobreaprendizaje derivados de la falta de datos [Masters, 1995, Tibshirani, 1996]

4.3.3.5. Variante Iterativa Equilibrada de Validación Cruzada (VIECV)

Se ha implementado una variante, a partir de las técnicas HOCVE y KFCV, para expandir la cantidad de patrones disponibles y eliminar la posible aleatoriedad de los resultados obtenidos por la RNA respecto de los datos usados. Esta propuesta comparte con el KFCV la misma técnica de reparto de patrones, basada en la selección aleatoria para generar conjuntos mutuamente excluyentes. El conjunto original de patrones es dividido en dos o tres subconjuntos, *train*, test y CV, en función de las necesidades de la experimentación a realizar. Esta división se realiza de forma aleatoria pero con la característica de no existir repetición entre los elementos que forman, por lo que se

Figura 4.10: Proceso de selección aleatoria con *VIECV*

trata de conjuntos disjuntos al igual que en HOCV y KFCV:

$$\{C_{train} \cap C_{test}\} = \emptyset \vee \{C_{train} \cap C_{CV} \cap C_{test}\} = \emptyset \quad (4.11)$$

Este proceso es repetido k veces asegurando que en ninguna de las variantes se obtienen conjuntos similares. Por tanto, para el entrenamiento y validación de la red se utilizan conjuntos distintos asegurando la independencia de los resultados de los ensayos balísticos utilizados. A mayor número de repeticiones más confianza se puede tener acerca del rendimiento del modelo neuronal diseñado. Siguiendo las pautas marcadas para HOCV y KFCV, el error de predicción E se calculará como el promedio obtenido para cada uno de los conjuntos:

$$E = \frac{1}{k} \sum_{i=1}^k E_i \quad (4.12)$$

Por otro lado, la cantidad de ensayos destinados a cada uno de los subconjuntos puede ser parametrizada, con el objetivo de obtener diferentes configuraciones de subconjuntos con cada iteración y estudiar la tendencia del error de la RNA. Además, las diferentes clases asociadas a los ensayos (e.g. perforación o penetración) pueden ser distribuidas con diferentes porcentajes para analizar su influencia o asegurar una distribución equilibrada acorde con el mundo real. En este sentido, los ejemplos del conjunto de entrenamiento deben ser representativos de todo el dominio del problema, conteniendo dentro de lo posible una cantidad equitativa de todas las clases a identificar. Gracias a esto, se evita que la red se especialice sobre unos subconjuntos de los datos y tienda a clasificar la

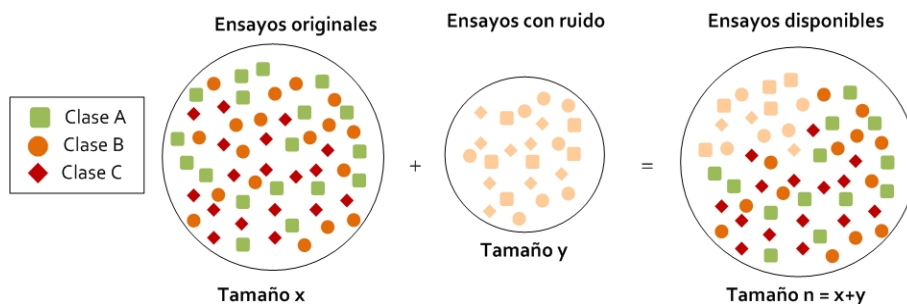


Figura 4.11: Proceso de incorporación de ruido a los patrones de entrenamiento

nueva información sobre una clase o clases concretas.

En la Figura 4.10 se puede observar un ejemplo de reparto con k repeticiones, diferentes tamaños y porcentaje de reparto de clases entre los conjuntos de *train*, *test* y *CV*.

4.3.3.6. Incorporación de ruido a los datos

Uno de los principales problemas asociados a las RNA y a los conjuntos de datos reducidos es la aparición de sobreaprendizaje durante el entrenamiento. Dentro de las alternativas más habituales para solucionar esta situación, destacan las ya comentadas como generar un conjunto de entrenamiento y otro de validación con parada temprana, introducir un término de penalización o podar la red. En este apartado se describe otra posible alternativa que consiste en la introducción de ruido a los patrones para generar nuevos vectores artificiales. Esta técnica ha demostrado ser asintóticamente consistente [Holmstrom and Koistinen, 1992] y gracias a las experimentaciones realizadas se ha verificado la mejora en la capacidad de generalización de las redes en varios dominios [Matsuoka, 1992, An, 1996]. Algunos autores aseguran que la introducción de ruido incluso crea estructuras de RNA más robustas y estables. Estas tienen la característica de mejorar la habilidad de clasificar correctamente clases que no aparecen en el conjunto de entrenamiento y mejorar la clasificación de entradas corruptas y con ruido [Sietsma and Dow, 1991].

Para implementar esta solución en un problema concreto, se generan de forma aleatoria nuevos patrones mediante técnicas estadísticas de generación de datos, que luego son añadidos al conjunto original. Esta generación de datos se basa en la introducción de cierto desplazamiento o ruido sobre los datos originales, lo que permite aumentar el número de patrones efectivos para el entrenamiento (cf. Figura 4.11).

Dentro de las alternativas existentes, la forma más común de ruido en las comunicaciones de datos digitales es el ruido blanco aditivo Gaussiano (más conocido por las siglas AWGN procedentes de Additive White Gaussian Noise), causado por los

múltiples procesos aleatorios de la corriente eléctrica o por las agitaciones térmicas de elementos conductivos. El término Gaussiano se refiere a la distribución de Gauss que siguen los datos debido a su carácter aleatorio. Esta distribución se caracteriza por el valor medio y desviación típica de los datos, presentando la forma de una campana o curva Gaussiana. Por su parte, blanco se refiere a la fuente de ruido de potencia de densidad espectral, que es idealmente plana con la frecuencia. Finalmente, el término aditivo hace referencia al hecho de que el ruido se combina con la señal deseada siendo un importante factor limitante en la transmisión de información. La utilización del ruido AWGN, en problemas con RNA, ha sido estudiada por varios autores para mejorar las prestaciones del entrenamiento si los datos son reducidos [Karystinos and Pados, 2000]. Su aplicación ha permitido expandir el conjunto de datos disponibles, al añadir nuevos vectores de entrada-salida a los ya existentes, con el fin de combatir el sobreaprendizaje.

Pasando al ámbito de las RNA, en función de las características del problema a resolver, si es de clasificación o de regresión, existen dos procedimientos distintos a la hora de añadir ruido a un conjunto de patrones del tipo (x_i, y_i) , donde x_i son las entradas e y_i las salidas. En el dominio de clasificación de patrones, el ruido se añade sólo sobre aquellas variables del vector con valores continuos, no discretos, ya que no tiene sentido añadir ruido a las clases de salida. En el segundo caso, dominio de regresión, si las distribuciones de x e y son continuas, es natural añadir ruido a los dos componentes, entradas y salidas. En ambos casos se genera, a partir de los patrones originales, nuevos vectores aleatorios (x_h^n, y_h^n) , donde el valor escalar h controla la magnitud del ruido.

Estas directrices se pueden observar en el estudio realizado por varios autores como Koistinen y Holmstrom [Koistinen and Holmstrom, 1991, Holmstrom and Koistinen, 1992], donde se especifican los pasos a realizar si se tienen n ensayos originales del tipo (x_i, y_i) y se quieren generar nuevos ensayos artificiales usando ruido aditivo. En la Tabla 4.2 se muestra un resumen de estas directrices.

Procedimiento 1 (añadir ruido sólo a x)	Procedimiento 2 (añadir ruido a x e y)
1. Seleccionar x_i e y_i donde $i \in 1, \dots, n$ equiprobable para cada índice	1. Seleccionar x_i e y_i donde $i \in 1, \dots, n$ equiprobable para cada índice
2. Crear un ejemplo S_x de densidad K_x sobre R^d	2. Crear un ejemplo (S_x, S_y) de densidad $K_{x,y}$ sobre R^{d+p}
3. Crear $x_h^n = x_i + hS_x$ y mantener $y_h^n = y_i$	3. Crear $x_h^n = x_i + hS_x$ e $y_h^n = y_i + hS_y$

Tabla 4.2: Procedimientos para generar vectores artificiales con ruido

En la mayoría de los casos se ha optado por utilizar un ruido AWGN con distribución $N \approx N(0, \sigma^2)$, i.e. se emplea una distribución normal de media cero y varianza aleatoria. La elección de un correcto valor para la varianza es uno de los puntos más sensibles, los investigadores han encontrado diferentes valores en función del problema analizado. Holmstrom [Holmstrom and Koistinen, 1992] y Karystinos y Pados [Karystinos and Pados, 2000] usan el valor 0.0262, Sietsma y Dow aplican el valor uno [Sietsma and Dow, 1988], Koistinen y Holmstrom [Koistinen and Holmstrom, 1991] y Reed et al. [Reed et al., 1992] aplican el valor 0.1.

Otra técnica consiste en la introducción de ruido en los patrones según se introducen en la red, modificando de esta forma el vector de entradas Bishop [1995], o el de entrada-salida [Holmstrom and Koistinen, 1992]. Su aplicación permite conocer la influencia del ruido sobre el aprendizaje de la RNA y evitar que se especialice sobre los patrones usados durante el entrenamiento, i.e. el vector de entradas se mueve (*jitter* en terminología inglesa) sobre el conjunto original reduciendo la posibilidad de sobreaprendizaje. Al mantener los patrones originales e ir cambiando de forma aleatoria el ruido en cada ciclo o *epoch*, los pesos son incapaces de retener la información específica del ruido propio de las medidas.

En esta investigación se han incluido estas dos técnicas para combatir el sobreaprendizaje por un lado, y para aumentar el número de patrones disponibles por el otro. En el primer caso, la creación de nuevos patrones artificiales, se ha optado por ruido AWGN con distribución $N \approx N(0, \sigma^2)$, media cero y varianza variable en función de la distribución asociada a la muestra de cada característica de los ensayos balísticos. Para implementar el segundo caso, la incorporación de ruido a los datos de entrada, se ha incluido un componente en la estructura del modelo neuronal que permite siguiendo una distribución Uniforme (cf. Ecuación 4.13) o Gaussiana (cf. Ecuación 4.14) la inyección de ruido a las entradas de la red. Este ruido es especificado a partir de la media y la varianza de los valores siguiendo las formulaciones:

$$y = \sqrt{3\sigma^2(x - 0,5) + \mu} \quad (4.13)$$

$$y = \sigma^2 \sqrt{-2\log x \cos(2\pi x) + \mu} \quad (4.14)$$

donde

- x es un número pseudoaleatorio entre 0 y 1.
- σ es la raíz cuadrada de la varianza.
- μ es la media.

Al calcular la media y la varianza para cada una de las características del vector de entradas, el ruido se puede aplicar con diferentes distribuciones sobre cada una de las neuronas de entrada. Por último, gracias a la modificación de los patrones de entrada, se puede comprobar la sensibilidad de cada modelo neuronal frente a la aparición de ruido durante el entrenamiento y su influencia sobre su capacidad de generalización. El listado de ensayos generados mediante esta técnica puede observarse en el Anexo B.

4.3.3.7. Resumen de las técnicas

Una vez finalizado el análisis de las técnicas existentes para mejorar la eficiencia de las RNA al trabajar con datos reducidos, se presenta en este apartado a modo de resumen sus características más importantes. Entre sus propiedades cabe destacar si están basadas en técnicas de remuestreo, si los conjuntos de datos son excluyentes y si las clases en cada conjunto están equilibradas respecto al conjunto original y por último el coste computacional asociado. Además para cada técnica se especifica el ámbito idóneo para su utilización en función de la cantidad de datos de que se dispone.

Cabe destacar que las técnicas basadas en la introducción de ruido no dividen los conjuntos, por lo que la exclusión y equilibrado de los datos en cada conjunto es dependiente del diseñador y del reparto elegido.

En la Tabla 4.3 se resumen las versiones analizadas para cada una de las técnicas.

4.4. Fase 2: Arquitectura de la red

La cuestión fundamental a la hora de resolver cualquier problema práctico con estructuras de computación basadas en RNA es dilucidar cómo se comportará el modelo diseñado sobre el conjunto de datos de test. Esta situación se conoce como el problema de la generalización y su resolución determina el rendimiento final del modelo cuando se enfrenta a situaciones desconocidas.

Para asegurar una capacidad de generalización óptima, es primordial conseguir una red tan pequeña como sea posible. Si la red tiene demasiados elementos de procesamiento, se pueden obtener errores pequeños durante el entrenamiento, pero la generalización podría ser pobre ante datos nuevos debido al sobreentrenamiento, i.e. se crea una solución muy específica sobre los datos de entrenamiento. Por su parte una red con dimensiones demasiado reducidas puede no resolver correctamente el problema tratado, por lo que no se trataría de una solución óptima. Por tanto, como en cualquier problema ingenieril, la determinación de la mejor arquitectura posible pasa por localizar una solución de compromiso entre arquitectura y rendimiento en cada uno de los escenarios tratados.

Técnica	Muestreo	Conjuntos Excluyentes	Conjuntos Equilibrados	Coste Computacional	Uso
HOCV	NO	SI	NO	Bajo	Muchos datos
HOCVI	SI	SI	NO	Medio. Depende del número de repeticiones	Muchos datos
HOCVE	NO	SI	SI	Medio. Depende del número de repeticiones	Muchos datos
KFCV	SI	SI	NO	Medio. Depende del valor k	Datos Intermedios
KFCVE	SI	SI	SI	Medio. Depende del valor k	Datos Intermedios
LOOCV	SI	SI	NO	Alto	Pocos datos
RBP	SI	Si con reemplazo	NO	Medio. Depende del número de repeticiones	Pocos datos
VIECV	SI	SI	SI	Alto	Pocos datos
AWGN	NO	Depende del reparto posterior	Depende del reparto posterior	Bajo	Pocos datos

Tabla 4.3: Características más interesantes de las técnicas analizadas para trabajar con pocos datos

Uno de los apartados de mayor dificultad de formulación es la búsqueda de buenos valores iniciales para los pesos. De forma intuitiva, se puede ver que si los pesos iniciales se sitúan cerca del punto óptimo, a continuación buscar el mínimo será más rápido y fiable. Lamentablemente, no se sabe a priori donde se sitúan los pesos óptimos en el espacio de patrones (es por eso que conoce el aprendizaje como un proceso de adaptación). Algunos autores han propuesto una solución lineal para los valores iniciales de los peso, pero esto no siempre es lo más correcto. En muchos casos la solución lineal es exactamente el punto que la búsqueda debe evitar, ya que es un mínimo local fuerte. Es por esto que los valores iniciales elegidos para los pesos tienen una clara influencia sobre la capacidad de aprendizaje de la red. Una inicialización lejos de los valores óptimos, aumenta el tiempo de aprendizaje. Además, otro punto de especial interés asociado a los pesos de las conexiones es que permite que todas las neuronas de la red aprendan a la misma velocidad. Finalmente, para romper posibles simetrías que podría estancar el aprendizaje o saturar las neuronas, es práctica común que los pesos en la red se fijen con condiciones iniciales aleatorias.

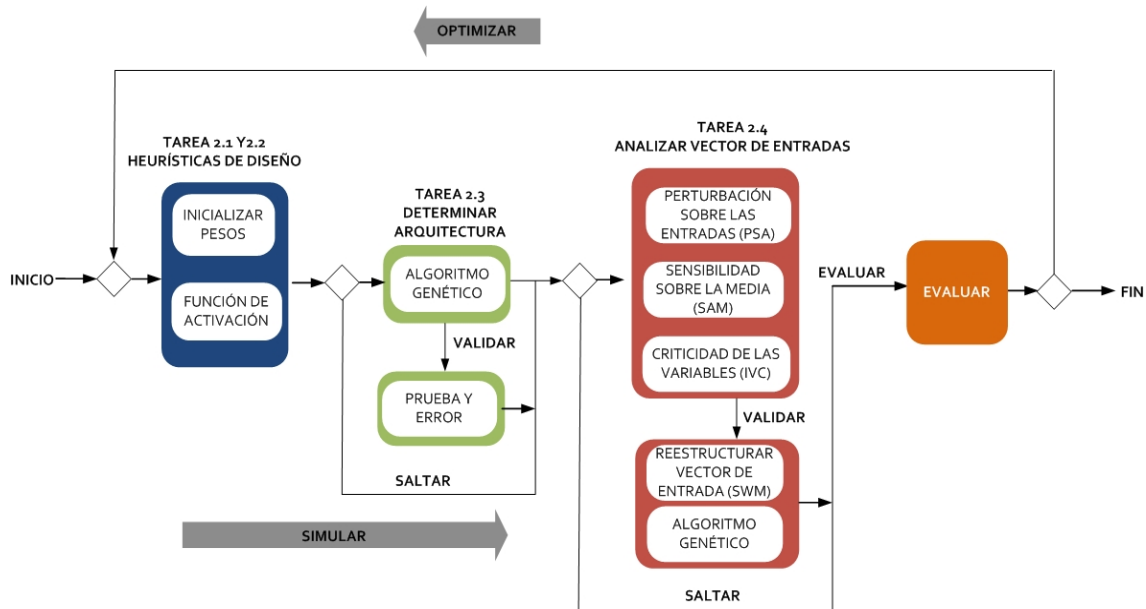


Figura 4.12: Detalle del proceso de optimización asociado a la Fase 2: Arquitectura de la red

En la Figura 4.12 se muestran las diferentes tareas pertenecientes a la fase de optimización de la arquitectura de red. Estas tareas han sido englobadas en tres grandes bloques para homogeneizar su utilización. El propósito de esta fase consiste en determinar la mejor configuración posible, mediante la evaluación, individual y conjunta, del rendimiento de las alternativas asociadas a cada tarea. En los siguientes apartados se concretan los objetivos y se justifica cada una de ellas.

4.4.1. Tarea 2.1: Inicialización los pesos de la red

Los algoritmos de entrenamiento basados en el gradiente, son técnicas de búsqueda local que quedan atrapados en puntos de la superficie del error donde el gradiente sea nulo, condición que asegura que se he encontrado un mínimo, pero que no asegura que se ha llegado al mínimo global de la función. Por ello, los pesos iniciales elegidos para la minimización del error serán determinantes, tanto para la calidad del mínimo encontrado, como para las iteraciones necesarias para encontrarlo. De forma intuitiva la elección de los pesos debería ser cercana al punto óptimo, para que la búsqueda del mínimo global fuera rápida y fiable. Desafortunadamente, no es posible conocer de antemano donde se encuentran los pesos óptimos dentro del espacio de la función del error. El problema es que no existe ninguna aproximación matemática que permita determinar los mejores valores para un problema dado. Por ello, los investigadores recurren a técnicas como los AG para asegurar que la solución final no depende de la posición de inicio o repeticiones de pruebas para quedarse con la mejor.

Con respecto a los valores a utilizar, algunos investigadores como Basheer y Hajmer [Basheer and Hajmeer, 2000] indican que estos tienen una gran influencia sobre la capacidad de convergencia de la RNA, e.g. si estos valores se sitúan sobre una superficie plana del error el entrenamiento se volverá extremadamente lento. No obstante otros estudios, como el de Fahlman [Fahlman, 1988], sostienen que esta circunstancia no debería tenerse en cuenta ya que su influencia real es mínima sobre el rendimiento final de la RNA.

Aunque existen varios algoritmos para asignar los valores a los pesos, casi todos implementan una inicialización aleatoria distribuida uniformemente sobre un rango $[-r, r]$ con media cero. El problema es que si se utilizan valores altos en los pesos se puede llegar a producir una saturación de las funciones de activación de las neuronas a las que están asociados, lo que a su vez provoca la anulación de la primera derivada del algoritmo RP estancando el proceso de entrenamiento. Por otro lado, valores pequeños puede producir que el gradiente del error sea demasiado pequeño lo que puede ralentizar el aprendizaje.

Por tanto la elección de valores pequeños es esencial para reducir la posibilidad de saturación de las neuronas. En esta investigación se estudiarán dos técnicas distintas. La primera expuesta por Haykin [Haykin, 1994] consiste en la inicialización neurona a neurona dentro del rango $(-\frac{r}{N_j}, \frac{r}{N_j})$, donde r es un número real dependiente de la función de activación y N_j es el número de conexiones que alimentan la neurona en cuestión.

La segunda es la utilizada por Principe et al. [Principe et al., 1999], y se basa en la premisa de que una neurona que se encuentra en su región lineal aprende más rápido que una que está en la región saturada. Para un mejor rendimiento del aprendizaje, el objetivo es que cada neurona aprenda aproximadamente a la misma cadencia. Si se establece la variación de las condiciones iniciales al azar basado en el *fan-in* de cada una (i.e. el número de entradas que recibe), cada neurona estará cerca de su región lineal, por lo que todas aprenderán a un ritmo similar. Para ello se utilizará la formulación $(-\frac{2.4}{I}, \frac{2.4}{I})$, donde I es el *fan-in* de la neurona.

Por último reseñar que se trata de una tarea transversal, por lo que siempre se va utilizar en cada modelo neuronal diseñado, i.e. al menos una regla de inicialización será aplicada en todas las simulaciones de esta metodología.

4.4.2. Tarea 2.2: Elección de la función de activación

Como ha quedado expuesto en §2.4.3.4, la aportación de cada una de las funciones de activación sobre la capacidad de generalización de una RNA no puede ser demostrada matemáticamente. Por tanto, la elección de una u otra es decisión del diseñador y

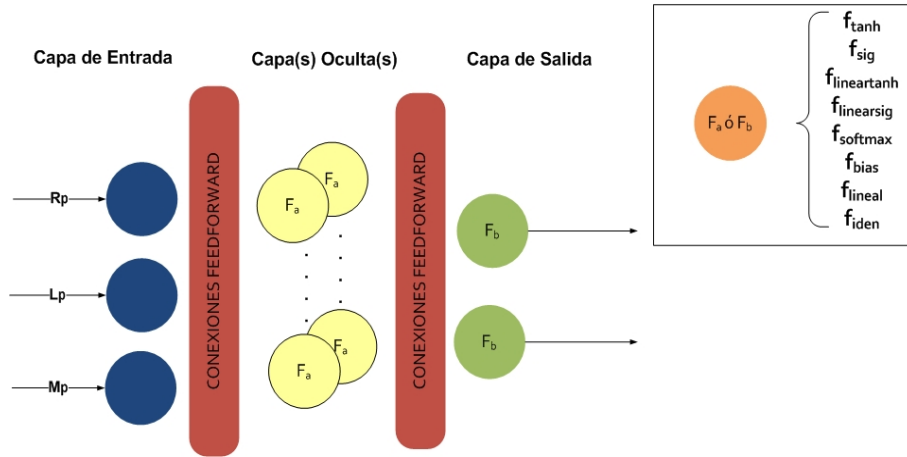


Figura 4.13: Funciones de activación disponibles para su empleo sobre las neuronas de las capas ocultas y salida

dependiente del tipo de problema a resolver. En esta investigación se han incluido diferentes tipos de función de activación, las cuales serán comparadas durante la experimentación a realizar. Como se observa en la Figura 4.13, la elección de una función u otra afecta a todas las neuronas de una misma capa, ya sea la oculta o la de salida. Entre todas las posibles alternativas se han elegido, a partir del trabajo de Principe et al. [Principe et al., 1999], las siguientes debido a sus prestaciones y aportaciones sobre el rendimiento de un modelo neuronal:

- Función tangente hiperbólica (f_{tanh}): Aplica un sesgo junto con la función tangente hiperbólica para cada neurona de la capa. Esto permite agrupar el rango de valores de cada neurona en un rango $[-1, 1]$, facilitando a la red la habilidad de tomar decisiones suaves.

$$f(x_i, w_i) = \frac{1 - e^{-X_i^{lin}}}{1 + e^{-X_i^{lin}}} \quad (4.15)$$

- Función sigmoideal (f_{sig}): Aplica una escala y una función sigmoide en cada neurona de la capa. El rango de valores para cada neurona en la capa variará entre $[0, 1]$, facilitando a la red la habilidad de tomar decisiones suaves.

$$f(x_i, w_i) = \frac{1}{1 + e^{-X_i^{lin}}} \quad (4.16)$$

- La función tangente hiperbólica mixta ($f_{lineartanh}$) sustituye la parte intermedia de la función tangente hiperbólica por una recta de pendiente b , por lo que es una aproximación a dicha función f_{tanh} mediante una función lineal por trozos o escalón. Es una variante menos utilizada pero que ha demostrado buenos resultados,

siendo computacionalmente más eficiente que f_{tanh} , ya que es más fácil de calcular la correspondencia entre entradas y salidas.

$$f(x_i, w_i) = \begin{cases} -1 & \text{si } x_i^{lin} < -1 \\ 1 & \text{si } x_i^{lin} > 1 \\ x_i^{lin} & \text{en caso contrario} \end{cases} \quad (4.17)$$

- La función sigmoideal mixta ($f_{linearsig}$) sustituye la parte intermedia de la función sigmoideal por una recta de pendiente b , por lo que es una aproximación a dicha función f_{sig} mediante una función lineal por trozos o escalón. Es una variante menos utilizada pero que ha demostrado buenos resultados, siendo computacionalmente más eficiente que f_{sig} , ya que es más fácil de calcular la correspondencia entre entradas y salida.

$$f(x_i, w_i) = \begin{cases} 0 & \text{si } x_i^{lin} < 0 \\ 1 & \text{si } x_i^{lin} > 1 \\ x_i^{lin} & \text{en caso contrario} \end{cases} \quad (4.18)$$

- La función sesgo (f_{bias}): Aplica un termino de sesgo que puede ser adaptado.

$$f(x_i, w_i) = x_i + w_i \quad (4.19)$$

- La función softmax ($f_{softmax}$) o logística multinomial interpreta la salida de la red como una probabilidad. Su suma debe ser igual a uno para que constituya una densidad de probabilidad. Como normal general la salida de la RNA produce una medida de similitud. Con el fin de convertir este similitud medida a una probabilidad, se puede utilizar esta función dentro de las neuronas de una misma capa. Su utilización sólo tiene sentido en problemas de clasificación.

$$f(x_i, w_i) = \frac{e^{x_i^{lin}}}{\sum_j e^{x_j^{lin}}} \quad (4.20)$$

- La función lineal f_{lineal} implementa una función lineal pero con pendiente y desplazamiento. Es por tanto más potente que la función f_{bias} al implementar transformaciones afines.

$$f(x_i, w_i) = \beta x_i + w_i \quad (4.21)$$

- La función identidad f_{iden} simplemente realiza la correspondencia entre las entradas y las salidas.

$$f(x_i, w_i) = x_i \quad (4.22)$$

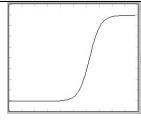
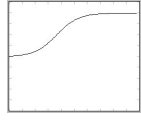
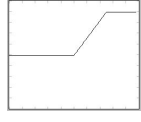
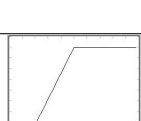
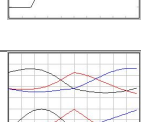
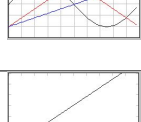
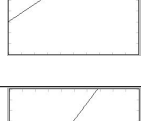
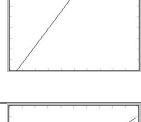
Función	Rango	Características	Gráfica
f_{tanh}	$[-1, 1]$	Función no lineal	
f_{sig}	$[0, 1]$	Función no lineal	
$f_{lineartanh}$	$[-1, 1]$	Aproximación lineal por trozos a la función tangente hiperbólica	
$f_{linearsig}$	$[0, 1]$	Aproximación lineal por trozos a la función sigmoide	
$f_{softmax}$	$[0, 1]$	Su suma debe ser igual a 1. Usada en clasificación	
f_{bias}	Infinito	Función lineal con pendiente y sesgo ajustable	
f_{lineal}	Infinito	Función lineal con desviación ajustable	
f_{iden}	Infinito	Función identidad	

Tabla 4.4: Funciones de activación utilizadas

En estas ecuaciones los parámetros incluidos son: w_i es la matriz de pesos, x_i el vector de entradas de la neurona, β es la pendiente de la recta y x_i^{lin} es igual al coeficiente βx_i heredado de f_{linear} .

En la Tabla 4.4 se muestra un resumen de las funciones de activación disponibles durante la optimización de los modelos neuronales. Las cinco primeras funciones son no lineales. En el caso de utilizarlas es necesario normalizar los datos de entrada antes de introducirlos en la RNA, para ajustar el rango de salida. En los problemas de regresión, donde en la capa de salida la respuesta deseada es una función continua de las entrada, lo mejor es utilizar funciones con un rango infinito como f_{bias} , f_{lineal} , f_{iden} .

4.4.3. Tarea 2.3: Selección de la arquitectura óptima

El primer pensamiento a la hora de diseñar una RNA podría ser que cuanto más grande fuera su estructura, mejor rendimiento obtendríamos (siempre que tengamos datos suficientes para entrenarla). Sin embargo, la teoría de la escalabilidad asociada a las máquinas de aprendizaje automático muestra que las redes más grandes pueden no aprender mejor, ya que necesitan más datos de los que extraer la información representativa del problema.

En §3.5.1 se han aproximado para cada escenario las estructuras iniciales de cada modelo de red. Sin embargo, como ha quedado expuesto anteriormente, un PMC entrenado con el algoritmo RP no controla directamente su capacidad de generalización, siendo necesario un conjunto CV para detener el entrenamiento, lo que permite maximizar la generalización para un determinado tamaño de red. Como se ha demostrado en §2.4.4.3, el método de la parada temprana facilita un criterio para detener el entrenamiento de una RNA en el punto de menor error sobre el conjunto CV, i.e., el punto de mejor generalización para una combinación concreta de topología y complejidad del modelo. Sin embargo, esto no proporciona un mecanismo para establecer la mejor topología de red respecto a la capacidad de generalización. Por tanto, la cuestión es la siguiente: ¿mantiene una red más grande la capacidad de generalización de las más pequeñas?

Si se analiza la manera en que la red realiza su trabajo, se puede ver que el tamaño de la misma (también conocido como complejidad del modelo) presenta una gran relación con el rendimiento: muy pocos grados de libertad (pesos) afectan la capacidad de la red para lograr un buen ajuste sobre la función objetivo. Si la red es demasiado grande, sin embargo, no generalizará correctamente, porque el ajuste es demasiado específico sobre el conjunto de datos de entrenamiento (memorización de los datos). Una red de tamaño intermedio es siempre la mejor opción. Por lo tanto, para lograr un rendimiento óptimo, es indispensable incluir en la metodología de diseño del PMC algún método de control de la complejidad.

El problema del tamaño de la red se puede describir de forma simplificada adaptando el postulado de Occam² a este dominio: *“Cualquier máquina de aprendizaje automático debe ser suficientemente grande para resolver un problema dado, pero no más grande”*.

La cuestión es determinar que significa suficientemente grande. Como se ha expuesto en el apartado §2.4.3.2 existen muchas heurísticas para resolver esta cuestión, pero no dejan de ser meras aproximaciones que no pueden ser tomadas como reglas de uso general. Dentro de todas las alternativas posibles, las más utilizadas para delimitar el tamaño óptimo de un modelo neuronal se pueden agrupar en dos categorías. En la primera se comienza con un tamaño de red pequeño e ir aumentándolo, lo que se conoce

²En su forma original *“entia non sunt multiplicanda praeter necessitatem”* o “no ha de presumirse la existencia de más cosas que las absolutamente necesarias”

como método de crecimiento (*growing method*), mientras que en la segunda se opta por una red de grandes dimensiones e ir reduciendo su tamaño, lo que se conoce como métodos de poda (*pruning method*).

El segundo punto de relevancia durante el diseño de una RNA y más concretamente de un PMC, es determinar la cantidad de elementos ocultos, capas y neuronas, que son necesarias para resolver un problema. En este caso se ha optado por técnicas de fuerza bruta, i.e. prueba y error de diferentes combinaciones, ya que la arquitectura inicial obtenida en §3.5.1 no es excesivamente compleja en ninguno de los escenarios. Para mejorar la convergencia de estas pruebas de fuerza bruta se utiliza el método de parada temprana durante el entrenamiento.

Por último, se ha incluido una heurística basada en AG, para detectar la mejor configuración de elementos ocultos en cada uno de los escenarios. En §4.7 se detalla su utilización dentro de la metodología de optimización.

4.4.4. Tarea 2.4: Selección del vector de entradas óptimo

Al emplear una RNA de cualquier tipo de arquitectura, uno de los objetivos más importantes durante el diseño es limitar el número de parámetros de entrada [Liu and Setiono, 1998]. Según Devijner la maldición de la dimensionalidad o "*the curse of dimensionality*" es la principal razón para limitar estos parámetros [Devijver and Kittler, 1982]. El dilema es que a medida que aumentan el número de parámetros, el número de patrones necesarios para el entrenamiento también aumenta. Por tanto, dadas las características del problema del impacto balístico, uno de los objetivos para esta investigación es determinar la arquitectura óptima para cada una de los PMC diseñadas.

Para lograrlo se han utilizado varias técnicas encaminadas a cuantificar la influencia de cada una de las variables en las predicciones que realiza el modelo de RNA. A medida que se entrena la red, el diseñador puede conocer el efecto que tiene cada una de las entradas sobre las salidas que se obtienen, lo que facilita una retroalimentación para determinar las variables más significantes. Gracias a esto, el diseñador de la red puede decidir si eliminar aquellos canales del vector de entrada menos determinantes. Esto permite reducir el tamaño de la red, lo que conlleva la disminución de la complejidad y del tiempo necesario para finalizar el aprendizaje.

Para evaluar la influencia de las diferentes entradas de la red, esta investigación incluye varias alternativas para el análisis de la sensibilidad. La primera consiste en un método para extraer la causa y el efecto de la relación que existe entre las entradas y las salidas de la red. La idea básica es que las entradas de la red son modificadas ligeramente, analizando el porcentaje la alteración correspondiente que se produce sobre la salida. Para ello se ha incluido un componente de control de activación que genera los datos de

entrada para el análisis de la sensibilidad, aumentando temporalmente la entrada por un pequeño valor o *dither*. El aprendizaje se deshabilita durante esta operación por lo que los pesos de las conexiones no se ven afectados.

El cambio observado en la salida se conoce como dato de sensibilidad, de manera que cada valor representa el porcentaje de efecto que tiene una entrada en concreto en el vector de salida en global (suma de todos los canales de salida). Este método de análisis se conoce como análisis de la perturbación (*Perturbation Sensitivity Analysis* o PSA en literatura inglesa). La ecuación usada para este indicador para una variable k es la siguiente:

$$P_{k,d} \% = \|E \% - E_d \%\| \quad (4.23)$$

donde $E \%$ es el error obtenido sin la perturbacion y $E_d \%$ es el error que devuelve la red usando el método PSA sobre una entrada k .

El segundo método de análisis de sensibilidad recurre al concepto de sensibilidad propuesto por Principe et al. [Principe et al., 1999], y se entiende como una medida de la influencia que tienen pequeños cambios en las variables independientes sobre los valores de las funciones de salida. Por tanto, este análisis de sensibilidad facilita una medida de la importancia relativa entre las entradas de un modelo neuronal, y refleja como la salida del modelo varía en respuesta a las variaciones que se producen sobre las entradas. Esta métrica se denomina sensibilidad sobre la media (*Sensitivity About the Mean* o SAM en literatura inglesa) y permite conocer la influencia de cada una de las entradas sobre la salida que obtiene la red. Su nombre proviene del tipo de modificación que se realiza sobre las entradas para evaluar la importancia de cada unas de las entradas. En este caso, una vez entrenada la red y con los pesos fijos, se modifica de forma aleatoria cada atributo del vector de entradas alrededor de su media, mientras el resto de mantienen fijos en su respectivas medias. A continuación, se miden los cambios producidos sobre la salida de la RNA. El cambio introducido en las entradas se genera añadiendo un valor aleatorio, obtenido a partir de una varianza determinada por el usuario, para cada ejemplo. Posteriormente se calcula la salida correspondiente para cada ejemplo. Este proceso se repite un número de veces a determinar para cada una de las entradas, calculando para finalizar la sensibilidad para una entrada k a través de la siguiente ecuación:

$$S_k = \frac{\sum_{p=1}^P \sum_{i=1}^o (y_{ip} - \bar{y}_{ip})^2}{\sigma_k^2} \quad (4.24)$$

donde \bar{y}_{ip} es la salida i -iesima obtenida con los pesos fijos para el patrón p -iesimo p , o es el número de neuronas de salida, P es el número de patrones y σ_k^2 es la varianza de la perturbación introducida. Este indicador es sencillo de calcular una vez entrenada la red y mide de forma efectiva como afecta una modificación en las entradas a la salida a través del conjunto de datos disponible, *train*, test y validación. Aquellas entradas con una mayor sensibilidad tienen mayor importancia y por tanto deberían ser mantenidas en el modelo neuronal. Por el contrario, aquellas con poca sensibilidad pueden ser tenidas en

cuenta para su eliminación, lo que permite mejorar el entrenamiento al reducir el tamaño de la red, disminuir el coste de obtención de los datos, con un impacto insignificante sobre el rendimiento de la red. En los trabajos de Mandal et al. [Majumder et al., 2007], Sokolova et al. [Sokolova et al., 2006] y Majumder et al. [Majumder et al., 2007], se pueden observar ejemplos de su aplicación en diversos dominios.

Además, se ha ideado e incluido otro indicador en esta investigación que permite conocer la criticidad de cada una de las variables de entrada en función de los valores que toman durante el entrenamiento y validación y la predicción del modelo neuronal. Esta técnica se ha denominado criticidad de las variables (*Indicator of Variable Criticality* o IVC en terminología inglesa) y representa, en un rango $[0, 1]$, la relación entre el número de veces que una variable k estaba fuera de rango³ cuando la predicción ha fallado, y el número total de veces que la variable k ha estado fuera de rango. La ecuación asociada a este indicador para una variable k es la siguiente:

$$C_{k,n} \% = \frac{R_{k,n}}{F_{k,n}} * 100 \quad (4.25)$$

donde $R_{k,n}$ indica el número de veces que la variable k ha salido fuera de rango en n test, y $F_{k,n}$ representa el número de veces que ha fallado la salida o predicción teniendo la variable k valores atípicos en n test.

Por tanto, el objetivo de las técnicas y métodos anteriores es determinar las variables explicativas del modelo matemático que regula la salida de la RNA. Para ello se incluirán sólo aquellas que puedan facilitar información o influenciar a la RNA, descartando aquellas que no lo sean. La ecuación que aproxima la función de salida de una red neuronal está basada en un modelo de regresión múltiple. Para el problema de clasificación, el valor de la variable dependiente, y , perforación, se ajusta a partir de un conjunto de variables independientes, $V_{ent} = \{M_b, L_b, R_b, V_b, H_p, T_p, T_b\}$. Teniendo en cuenta la formulación asociada con esta clase de regresiones y aplicándola sobre este dominio se obtiene la siguiente ecuación:

$$y = \epsilon + \sum_{i=1}^N \alpha_i V_{ent}(i) \quad (4.26)$$

donde α_i son los parámetros desconocidos que deben ser considerados, N es el número de variables independientes y ϵ es un componente de error. Para los problemas de regresión las variables dependientes son M_{res} y V_{res} . Por último, es importante tener en cuenta la aparición de posibles colinealidades e interacciones entre estas variables independientes.

Para probar las deducciones obtenidas con las técnicas y métodos anteriores, al final

³Se entiende por fuera de rango un valor que no ha sido utilizado durante el entrenamiento y por lo tanto es nuevo para la RNA



Figura 4.14: Utilización de los indicadores de sensibilidad para la optimización del vector de entradas

de esta tarea se añade una variación de uno de los métodos más usados en regresión estadística. Se trata de la alternativa al método clásico de regresión paso a paso o *StepWise Method* (SWM) expuesta por Gevrey et al. [Gevrey et al., 2003]. Éste consiste en la eliminación secuencial de variables del vector de entradas y en el análisis del efecto que esto produce sobre la salida. Este procedimiento es repetido usando modelos con menos variables en cada paso, combinando las más influyentes hasta que el rendimiento de la RNA comienza a disminuir de forma notable.

En resumen, el objetivo de esta tarea es analizar la influencia de cada una de las variables sobre el rendimiento de la RNA. Gracias a los indicadores incluidos, se podrá determinar la influencia o importancia relativa de cada una de las variables y tomar las decisiones oportunas para seleccionar aquel vector de entradas óptimo para un modelo neuronal concreto. En la Figura 4.14 se muestra la utilización de ambas técnicas dentro del entrenamiento y validación de un modelo neuronal.

4.5. Fase 3: Algoritmo de entrenamiento

El aprendizaje o adaptación es un paso crucial en la teoría asociada a las RNA, siendo el procedimiento encargado de extraer la información requerida de los datos de entrada. Si el aprendizaje es incompleto, los valores de los pesos no estarán cerca de sus valores óptimos, y el rendimiento se resentirá. Como se ha explicado en §2.4.4, existen técnicas para buscar sobre la superficie de rendimiento el valor óptimo de los pesos, pero sin embargo esta búsqueda tiene que ser controlada de manera heurística. El diseñador de la red puede afectar directamente sobre esta búsqueda mediante los siguientes criterios [Principe et al., 1999]:

- El criterio de parada.
- La tasa de aprendizaje (η) y constante *momentum* (μ).

- Los algoritmos de aprendizaje.

Un aspecto de interés para el diseñador es entender las cuestiones que afectan a cada uno de estos temas para entrenar de manera efectiva una RNA. Además, se debe recordar que el rendimiento final también depende de la selección de los pesos iniciales y de la cantidad y la calidad de los datos utilizados para entrenar el sistema como se ha visto en la fase anterior. Cada uno de estos criterios serán estudiados y analizados durante esta fase para conocer su influencia en el problema del impacto balístico.

Los criterios de parada utilizados incluyen el empleo del método de parada temprana (cf. §2.4.4.3 para una explicación detallada) para combatir el sobreaprendizaje de los modelos neuronales. Esto permite reforzar la capacidad de generalización de la red, evitando que se especialice sólo sobre los patrones sobre los que entrena. Con respecto a los parámetros del algoritmo RP en su versión extendida, μ y η , se tendrán en cuenta las pautas del punto §2.4.4.2 dictadas en el trabajo de Basheer y Hajmeer [Basheer and Hajmeer, 2000], ya que permiten conocer y comprender los problemas de comportamiento típicos del algoritmo RP. Por último, la parte final de esta fase está dedicada a las diferentes versiones del método descenso de gradiente así como a las mejoras que presenta cada una de las alternativas (cf. §2.3.3).

La principal utilidad de esta fase de la metodología reside, en que en ciertos casos, al utilizar el algoritmo básico RP en un problema práctico, el entrenamiento puede implicar días o incluso semanas de computación. Debido a esto se han realizado numerosos esfuerzos en investigar diferentes métodos para acelerar la convergencia del algoritmo. Estas investigaciones pueden agruparse en dos categorías diferentes. La primera incluye el desarrollo de nuevas técnicas heurísticas, que surgen del estudio del rendimiento característico del algoritmo RP estándar. Estas técnicas heurísticas agrupan ideas como la variación de la tasa de aprendizaje y utilizar la constante *momentum*. La otra categoría se ha enfocado al empleo de técnicas de optimización numéricas para mejorar el rendimiento. Como se ha comentado en §2.2.5, el entrenamiento de una RNA se dirige mediante la minimización del error MSE, tratándose de un problema de optimización numérica. Debido a que la optimización numérica ha sido un campo de trabajo importante durante los últimos 30 o 40 años, parece razonable buscar algoritmos de entrenamiento más rápidos dentro de la gran cantidad de alternativas posibles.

En la Figura 4.15 se muestran las diferentes tareas pertenecientes a la fase de optimización del algoritmo de entrenamiento. Estas tareas han sido englobadas en cuatro grandes bloques para homogeneizar su utilización, dos secuenciales y dos que se realizan en paralelo al presentar dependencias entre ellas. El propósito de esta fase consiste en determinar la mejor configuración posible, mediante la evaluación, individual y conjunta, del rendimiento de las alternativas asociadas a cada tarea. En los siguientes apartados se concretan los objetivos y se justifican cada una de ellas.

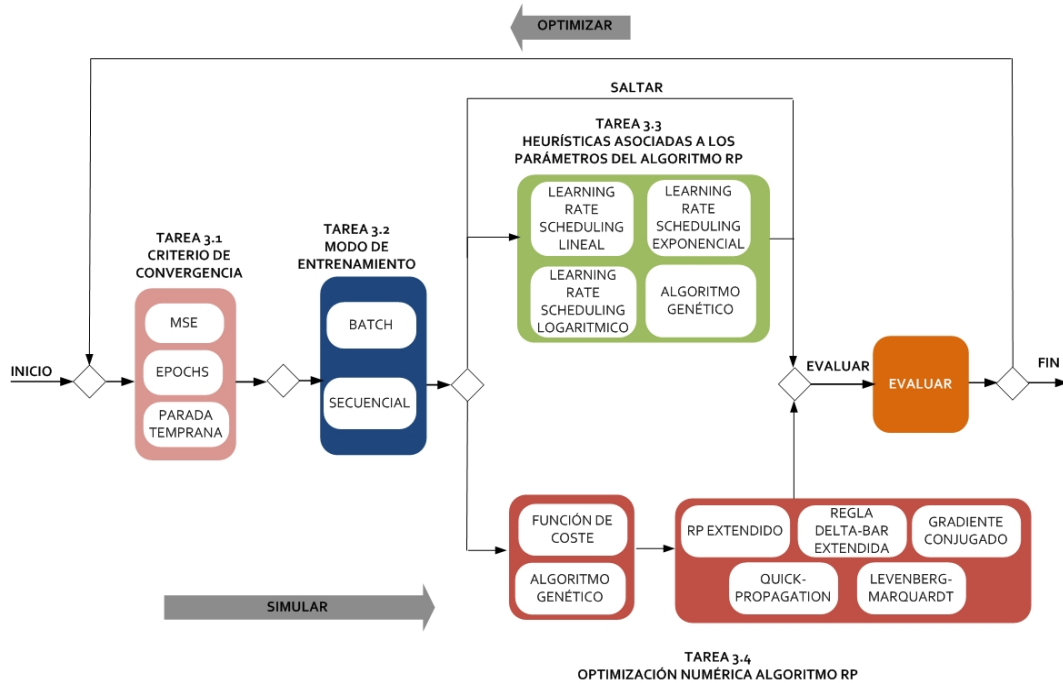


Figura 4.15: Detalle del proceso de optimización asociado a la Fase 3: Algoritmo de entrenamiento

4.5.1. Tarea 3.1: Criterio de convergencia

El objetivo del algoritmo RP es minimizar el error que se obtiene durante el proceso de entrenamiento. Para saber cuándo se ha alcanzado este punto se han incluido una serie de alternativas para su determinación, denominadas criterios de parada o convergencia. En primer lugar, si se utilizan los conjuntos clásicos de *train-test*, se recurrirá a los siguientes criterios:

- Evaluar el error MSE: Permite fijar el valor mínimo que debe alcanzar el error obtenido en el entrenamiento para poder parar el proceso.
- Detener al alcanzar un número de ciclos o *epochs*: Permite fijar el número de veces que se le muestra cada dato de entrenamiento a la red para que extraiga la información necesaria que permita generalizar nuevos valores en el futuro.

Para evitar un entrenamiento infinito en el tiempo, estos criterios pueden emplearse tanto individual como de forma conjunta. Gracias a esto, se evita que se realice un entrenamiento excesivo lo que conllevaría el sobreaprendizaje. En ambos casos, el número de *epochs* necesarios para una correcta generalización suele ser determinado mediante un proceso de prueba y error al no existir una regla de uso general. Por otro lado, estos dos métodos no aseguran que se alcance el mínimo global del error, ya que el valor fijado para el criterio de parada obtenga un mínimo local dentro del entrenamiento. Sin embargo,

tienen la ventaja de reducir el tiempo del proceso de entrenamiento, sacrificando cierta precisión en la generalización.

Como consecuencia de esto, es común utilizar otra alternativa como criterio de parada que implica la utilización del tercer conjunto de datos disponible, el de validación:

- Evaluar el error MSE obtenido en validación: También conocido como error de generalización. Se obtiene evaluando la tasa de clasificación incorrecta del clasificador con datos diferentes a los utilizados en el proceso de entrenamiento. Para ello cada cierto número de ciclos de entrenamiento, se le presentan a la red los patrones de validación, calculando la salida de la red para dichos patrones y evaluando el error obtenido.

El comportamiento típico del error de entrenamiento es que éste decrece monótonamente durante dicha fase. A su vez, el error sobre el conjunto CV decrece hasta un punto a partir del cual aumenta. Esto indica que a partir de este punto se produce un *sobreaprendizaje* sobre los datos de entrenamiento. Por ello, si se utiliza este criterio, el proceso de entrenamiento debe finalizar cuando se alcance el primer mínimo de la función del error de validación. Debido a esto, esta técnica es también conocida como *parada temprana*. Presenta la ventaja de no caer en mínimos locales para el error, pero su tiempo de cómputo puede ser mayor.

Por último reseñar que se trata de una tarea transversal que siempre se va incluir en cada modelo neuronal diseñado. Por lo tanto al menos un criterio de convergencia será aplicado en todas las simulaciones de esta metodología.

4.5.2. Tarea 3.2: Modos de entrenamiento

Después del análisis del estado del arte referente a este apartado, ha quedado claro que la elección de un modo de entrenamiento u otro es decisión del diseñador y suele ser dependiente del problema a resolver. Para simplificar esta tarea se van a seguir las pautas dictadas por Swingler [Swingler, 1996], Basheer y Hajmeer [Basheer and Hajmeer, 2000] y Rafiq et al. [Rafiq et al., 2001]:

- El modo *batch* es más rápido ya que requiere de menos actualizaciones de los pesos.
- El modo *batch* realiza una mejor estimación del vector asociado al error del gradiente y facilita una medida más precisa acerca de las modificaciones necesarias en los pesos.
- El modo *batch* tiende a quedar atrapado en mínimos locales con más facilidad que el modo secuencial.
- El modo *batch* es más sensible al valor inicial de los pesos.

- El modo secuencial consume menos recursos al necesitar menos espacio para almacenar los pesos una búsqueda estocástica más eficiente, lo que evita que caiga en mínimos locales de forma asidua.
- El modo secuencial puede quedar atrapado en un mínimo local al comienzo lo implica seguir una dirección equivocada sobre la superficie del error.

Debido a que el modo *batch* presenta un comportamiento equilibrado entre precisión y velocidad, se comenzará entrenando la red con este modo y se analizará el error obtenido. Si los resultados de generalización antes datos nuevos no son satisfactorios se cambiará a un modo de entrenamiento secuencial.

Por último reseñar que se trata de una tarea transversal que siempre se va incluir en cada modelo neuronal diseñado. Por lo tanto al menos un modo de entrenamiento será aplicado en todas las simulaciones de esta metodología.

4.5.3. Tarea 3.3: Heurísticas asociadas a los parámetros del algoritmo de entrenamiento

Los problemas asociados al algoritmo RP estándar, parada en mínimos locales y velocidad de convergencia lenta, han derivado en una serie de heurísticas para mejorar su eficiencia. Las dos heurísticas más importantes consisten en la modificación del algoritmo mediante la introducción de dos parámetros: constante *momentum* (μ) y factor de aprendizaje (η). Ambos parámetros se revelan como factores muy influyentes para la calidad del modelo neuronal y cuando estos se utilizan el algoritmo de aprendizaje se ha denominado RPE (cf. §2.3.3).

Determinar los mejores valores para cada tipo de red resulta además de costoso, impreciso, ya que los valores óptimos pueden cambiar a la vez que la función objetivo del error (i.e. en cada iteración). No obstante, su principal aportación es que al ajustar η y/o μ en cada iteración del entrenamiento disminuye notablemente el tiempo necesario para encontrar un mínimo en la superficie del error. El primero de estos parámetros, η , permite mejorar la convergencia del algoritmo al suavizar las oscilaciones sobre la trayectoria del error, mientras que el segundo acelera o frena la convergencia dependiendo del valor que tome.

Como algoritmo de aprendizaje de partida se utilizará en las simulaciones el algoritmo RPE, tomando como guía para aproximar los valores iniciales las reglas extraídas de la literatura y expuestas en §2.4.4.1. A modo de resumen se listan a continuación los aspectos más importantes:

- Valores para η dentro del rango $[0, 1]$.

- Valores para μ dentro del rango $[0, 1]$.
- Para que todas las neuronas aprendan al mismo ritmo se penalizarán los valores de η de capa a capa con el factor de penalización de Haykin equivalente a $\frac{2}{5}$ [Haykin, 1994].

No obstante, las propuestas más extendidas en la literatura para mejorar el rendimiento del algoritmo pasan por modificar los valores de estos parámetros durante el proceso de optimización. Para ello se han incluido dos heurísticas encaminadas a localizar los valores óptimos de los parámetros η y μ . La primera técnica se denomina *Learning Rate Scheduling* (LRS) [Principe et al., 1999] y permite adaptar el valor de la tasa de aprendizaje durante el entrenamiento. La teoría asociada a η establece que su valor puede ser alto si la función del error se mantiene estable, de forma que resulta más rápido encontrar algún camino por donde reducir este error. Por el contrario, se debería ir reduciendo conforme se esté más cerca del mínimo del error, cuando la función del error sea más inestable, hasta que se consiga otra vez la estabilidad. Basándose en este principio, la técnica LRS comienza eligiendo un valor elevado de η al comienzo del entrenamiento, para reducir el tiempo de computación, y se disminuye progresivamente para obtener una precisión alta con los valores finales de los pesos. Las alternativas utilizadas para realizar la variación de los parámetros μ y η , son las más comunes en la literatura: lineal, logarítmica y exponencial [Lefebvre et al., 2007]. La variación lineal de los valores para una iteración k se controla mediante la ecuación:

$$\eta(k+1) = \eta(k) - \beta \quad (4.27)$$

En el caso de la variación logarítmica de los valores se recurre a la ecuación:

$$\eta(k+1) = \eta(k) - \frac{\beta}{\eta(k)} \quad (4.28)$$

Y por último la ecuación asociada a la variación exponencial es:

$$\eta(k+1) = \beta\eta(k) \quad (4.29)$$

donde $\eta(0)$ sería el tamaño inicial de la tasa de aprendizaje y β una constante. Con este algoritmo el tamaño de la tasa se reduce linealmente en cada iteración, partiendo de un valor elevado y llegando prácticamente a cero hacia el final del entrenamiento. El tamaño de η puede reducirse paso a paso utilizando una regla lineal, geométrica o logarítmica. Si el valor inicial de $\eta(0)$ es demasiado alto, el aprendizaje puede divergir. La selección de β puede ser incluso más difícil que la selección de $\eta(0)$, ya que es altamente dependiente de la superficie del error. Si β es demasiado grande, los pesos no se puede mover con la suficiente rapidez hacia el mínimo y la búsqueda se podría estancar. Por su parte si la constante β es demasiado pequeña, la búsqueda puede alcanzar el mínimo global de forma rápida y ésta debe esperar mucho tiempo antes de que la tasa de aprendizaje disminuya lo suficiente como para minimizar las oscilaciones.

Por último, se ha incluido como alternativa una técnica evolutiva, basada en un AG, para detectar la mejor configuración de la constante μ y la tasa η en cada uno de los escenarios. Esta técnica ha demostrado su utilidad en otros dominios, para mejorar el comportamiento de diferentes tipos de RNA al optimizar los valores de los parámetros del algoritmo de aprendizaje [Fiszelew, 2002]. En §4.7 se detalla su utilización dentro de la metodología de optimización.

4.5.4. Tarea 3.4: Optimización numérica de algoritmo del entrenamiento

Como se ha comentado en §2.2.5 el entrenamiento de las RNA se basa en la minimización de una función del error E al variar el conjunto de pesos que definen una RNA (cf. Ecuación 4.30). Se trata por tanto de una optimización multivariable sin restricciones ya que no hay ningún otro requisito adicional con respecto a la función o a las variables de entrada. El problema de optimizar una función derivable, continua, multivariable y sin restricciones ha sido estudiado ampliamente fuera del campo de las RNA, y las distintas aproximaciones que se han seguido pueden ser aplicadas casi directamente al problema de minimización del error. Sin embargo, hay dos aspectos a tener en cuenta en este problema específico y que marcan la diferencia respecto a las técnicas convencionales de optimización: el alto número de variables a optimizar en este caso y la gran cantidad de derivadas que hay que calcular en cada paso de la iteración, que hacen los cálculos especialmente costosos.

En la Figura 4.16 se muestran los métodos de optimización alternativos, usados en esta investigación, aplicables a la modificación de los pesos en el proceso de entrenamiento de una RNA. Estos métodos, separados en heurísticos e iterativos, se aplican a problemas multivariables y sin restricciones en los que la solución analítica no es fácil de encontrar.

$$\frac{\partial E}{\partial w} \equiv 0 \Leftrightarrow \nabla E(w^*) = 0 \quad (4.30)$$

Los métodos iterativos presentan una estructura de bucle donde se inicializa el problema, se evalúa la respuesta obtenida y dependiendo de que se cumpla cierto criterio o no, o bien se pasa a la siguiente iteración o se llega al final del proceso. El problema de este tipo de métodos radica, en que cuando el algoritmo converge a una solución, es difícil de reconocer si se trata de un mínimo global o simplemente es una solución local. Este tipo de método sólo alcanza mínimos locales globales en el caso de que la función a minimizar sea convexa, condición que es de difícil cumplimiento para las complejas superficies de error generadas por un PMC. En los métodos de optimización iterativos, hay que determinar por una parte la dirección de búsqueda en el espacio de soluciones posibles, i.e., la dirección en la que deben modificarse los pesos en la iteración siguiente, y también cómo de lejos se ha de avanzar en la dirección elegida, i.e., el tamaño del



Figura 4.16: Clasificación de los diferentes métodos de optimización de la función del error utilizados

paso (*step-size*) de la iteración o lo que se ha denominado tasa de aprendizaje. La elección acertada de estas dos características determinará, tanto la eficiencia global de las optimizaciones, como su velocidad de convergencia. Dentro de estos métodos destacan las variantes del método del gradiente y las del método de Newton, las cuales mantienen problemáticas similares para la elección del tamaño del paso o tasa de aprendizaje, pero se diferencian en la dirección de búsqueda en cada iteración del algoritmo.

Por su parte los heurísticos, a diferencia de los anteriores, representan una alternativa para los casos en los que se quiere asegurar la localización de mínimos globales o para los casos donde los requisitos por los métodos de optimización no sean alcanzables (como la derivabilidad de las funciones, las restricciones, etc.). Un ejemplo típico de este tipo de métodos heurísticos son los AG, que tratan de simular las leyes de selección natural para obtener conjuntos de soluciones que se adaptan mejor a su entorno.

Por otro lado, una manera interesante de aumentar el rendimiento de un PMC es minimizar durante la fase de entrenamiento la función de coste asociada al algoritmo de aprendizaje. Para ello, se incluirán propuestas basadas en la norma L_p que influirán en el proceso de actualización de los pesos que realiza el algoritmo RP. Además, junto a las diferentes funciones de coste, se ha incluido una técnica evolutiva, basada en un AG, para detectar la mejor configuración de la constante μ y la tasa η .

En resumen, esta tarea es paralela a la Tarea 3.3 (cf. 4.5.3), encargada de detectar los mejores parámetros del algoritmo RP, y se divide a su vez en dos subtareas. La primera está orientada a estudiar la influencia de las diferentes funciones de coste, tanto por separado como de forma conjunta con un AG, para aprovechar las ventajas de unos valores óptimos de μ y η . En la segunda se analizarán diversas variantes del algoritmo de aprendizaje para el PMC diseñado. La utilización de estas alternativas en el dominio de las RNA ha sido muy prolífica ya que permiten solucionar los problemas típicos del algoritmo RP estándar. En los trabajos de Bishop [Bishop, 1996], Hagan et al. [Hagan et al., 1996] y Priddy [Priddy and Keller, 2005] pueden encontrarse numerosos ejemplos de las ventajas asociadas a su aplicación en diferentes clases de problemas, así como su

fundamentación matemática.

4.5.4.1. Función de coste

La función de coste más utilizada para un PMC es el error MSE simple [Swingler, 1996], pero hay ciertas situaciones donde podría ser más adecuado utilizar otras funciones de coste para mejorar su rendimiento. Dentro de las diferentes alternativas destacan los modelos de regularización [Girosi et al., 1995], los modelos internos de representación [Mrazova and Wang, 2007] o los modelos múltiples [Ueda, 2000] entre otros. Este estudio incluye una comparación entre las alternativas más usadas dentro del campo de las normas L_p , basadas en la regularización propuesta por Girosi et al. [Girosi et al., 1995] y Lefebvre et al. [Lefebvre et al., 2007].

Con respecto a la utilización de error MSE simple en la función de coste, tiene una serie de ventajas que lo hace muy útil para su uso en problemas de regresión y clasificación [Bishop, 1995, Principe et al., 1999]. Sin embargo, cuando aparecen valores atípicos en los datos la función de coste clásica (error MSE) tiene un comportamiento ineficiente porque la actualización de los pesos de la RNA se vuelve muy sensible. Los valores atípicos por tanto pueden producir un efecto desproporcionado sobre los valores óptimos de los parámetros asociados a la red. Para evitar estas situaciones, la ecuación asociada con la función de coste se modifica como sigue:

$$J_{n,k} = |d_{n,k} - y_{n,k}|^p \quad (4.31)$$

donde k es el índice para las salidas del sistema, n es el índice para los patrones de entrada y p es un número entero, que normalmente se denomina norma p del error. Cuando $p = 2$, se obtiene la norma L_2 o norma euclidiana que conduce al criterio MSE clásico. Cuando $p = 1$, obtenemos la norma L_1 , que es llamada también métrica Manhattan. La norma L_1 se considera como más robusta ya que actualiza los pesos de forma proporcional a su magnitud, por lo que es menos sensible a los atípicos que la norma L_2 . En general, las normas L_p con $p > 2$ producen desviaciones aún más grandes sobre los pesos, lo cual puede ser beneficioso en ciertos problemas en función del conjunto de datos. Cuando $p = \infty$ se obtiene la norma L_∞ o Minimax.

Como esta tarea utiliza el algoritmo RPE, y para aprovechar las ventajas que ofrecen unos valores óptimos de μ y η , se ha incluido además un componente evolutivo, basado en un AG, para localizar aquellos valores que mejor resultado ofrecen asociados a las diferentes funciones de coste.

En resumen, el uso de distintas normas proporciona diferentes soluciones a un problema de aprendizaje, ya que los pesos se modifican con la información que varía en función de la norma elegida. Por ello el posicionamiento de las funciones discriminantes

se ve afectado directamente por la norma, incluso para el mismo conjunto de datos. Se pueden encontrar en la literatura diferentes ejemplos de la utilización de las normas L_p [Girosi et al., 1995, Songwu et al., 1998, He and Sun, 2001, Burger and Neubauer, 2003].

4.5.4.2. Método del Gradiente

El método del gradiente fue introducido por Cauchy el año 1847 [Cauchy, 1847]. Se basa en una aproximación lineal de la función a optimizar, en este caso la función del error explicada en §2.2.5, actualizando la variable a modificar (los pesos de la red) en dirección contraria al gradiente. La aplicación de este método junto con la regla Delta (cf. §2.3.3) es lo que constituye el algoritmo estándar RP (cf. Ecuación 2.21).

La tasa de aprendizaje (η) permite controlar el grado de modificación de los pesos y su valor influye tanto en la rapidez como en la eficiencia del entrenamiento. Los algoritmos del descenso del gradiente presentan la ventaja de ser simples y fáciles de implementar, aunque en algunos casos el tiempo de convergencia puede ser demasiado elevado. Para intentar salvar estos inconvenientes han surgido diversas implementaciones dentro de este tipo de optimización.

4.5.4.3. Aprendizaje con momentum

Para evitar oscilaciones bruscas sobre la superficie del error se puede introducir en la actualización de los pesos el término momentum (μ), que incluye una dependencia con la actualización anterior de los pesos a modo de inercia. De este modo se consigue que la red no se actualice sólo en función del gradiente, sino que tenga en cuenta las tendencias más recientes en las superficies del error. Su inclusión actúa de forma similar a un filtro paso-bajo, ya que evita los pequeños cambios en la superficie del error y disminuye las posibilidades de caer en un mínimo local.

El método del gradiente con la constante *momentum* aplicado como algoritmo de aprendizaje del PMC, se ha denominado en esta investigación algoritmo RPE. Además se trata de una aproximación del gradiente conjugado, aunque en este último caso los valores de η y μ se eligen automáticamente en lugar de ser elegidos por el usuario.

4.5.4.4. Regla Delta-Bar Extendida

La siguiente heurística incluida es una variante de la Regla Delta-Bar (RBD) propuesta por [Jacobs, 1988] a la que se le incluyó la constante μ , conocida como Regla Delta-Bar Extendida (RBDE) [Minai and Williams, 1990]. Se trata de un procedimiento para el control adaptativo de η y μ y la búsqueda de la superficie de rendimiento óptima dentro del modelo neuronal. En el trabajo de Lefebvre et al. se encuentra una explicación detallada del mismo [Lefebvre et al., 2007]. En este caso el tamaño de ambos parámetros se adapta de acuerdo con los valores anteriores del error en cada neurona: $[\eta(t), \mu(t)]$. Si las actualizaciones de los pesos anteriores y las actuales son del mismo signo, se aumenta

la tasa de aprendizaje de forma lineal. Esto se justifica ya que si el peso se está moviendo en la misma dirección que se reduce el error, entonces se llegará más rápido con un mayor tamaño η . Si las actualizaciones tienen diferentes signos, se trata de una indicación de que el peso se ha desplazado demasiado lejos. Cuando esto sucede, la tasa de aprendizaje disminuye geoméricamente para evitar la divergencia. Las ecuaciones encargadas de gobernar el tamaño adaptativo de la tasa de aprendizaje durante el entrenamiento son:

$$\Delta\eta_{ij}(n+1) = \begin{cases} k & \text{si } S_{ij}(n-1)D_{ij}(n) > 0 \\ -\beta\eta_{ij}(n) & \text{si } S_{ij}(n-1)D_{ij}(n) < 0 \\ 0 & \text{en otro caso.} \end{cases} \quad (4.32)$$

donde S_{ij} es la media de los gradientes anteriores y D_{ij} es el gradiente actual. Por su parte k es la constante aditiva, β la constante multiplicativa. Tanto S_{ij} como D_{ij} tienen el mismo signo y su producto será mayor que cero. El primer caso de la ecuación 4.32 se refiere a una convergencia lenta, por lo que el tamaño de la tasa de aprendizaje aumentará en cada iteración (por una constante, lo que implica un proceso lento). El segundo caso se utiliza cuando el peso oscila por lo que el algoritmo decrece el valor η de forma proporcional al valor actual (mediante un decremento geométrico, lo que implica un proceso rápido). El lento decremento y rápido incremento se emplean para evitar la divergencia a cualquier precio, si el algoritmo se aleja, se pierde rápidamente toda la información que ha aprendido y debe empezar de nuevo.

De forma detallada, las condiciones para el incremento o decremento de la tasa de aprendizaje, dependen de los siguientes términos: $D_{ij}(n)$ es la derivada parcial del coste respecto al peso w_{ij} (i.e. el gradiente actual), y $S_{ij}(n)$ es la media en curso de las derivadas parciales actuales y anteriores, dada por la ecuación:

$$S_{ij} = (1 - \varphi)D_{ij}(n-1) + \varphi S_{ij}(n-1) \quad (4.33)$$

donde φ es un número entre cero y uno (denominado constante de tiempo exponencial). El producto de S_{ij} y D_{ij} es chequeado cuando el gradiente actual tiene el mismo signo que la media anterior de gradientes (reflejado como el valor S_{ij}).

4.5.4.5. Método del Gradiente Conjugado

Este método fue inventado por Hestenes y Stiefel en 1952 [Hestenes and Stiefel, 1952] y desde su aparición se han producido múltiples variaciones. Como se trata de un técnica de optimización, se puede aplicar durante el entrenamiento de una RNA para adaptar los valores de los pesos de igual forma que lo hace el algoritmo RP. Entre sus ventajas destaca la posibilidad de trabajar sobre con un elevado número de pesos, al contrario que la versión clásica del algoritmo RP.

Se trata de un método intermedio entre los métodos del gradiente clásicos y los métodos de Newton, ya que por una parte intentan mejorar la convergencia de los métodos del descenso del gradiente y por otro evitan el cálculo y el almacenamiento de la información requerida en un método de Newton. La característica principal de este tipo de métodos es que, asumiendo que la función de error es cuadrática, intentan conseguir direcciones de búsqueda que no interfieran con las realizadas en las iteraciones anteriores. De todas formas este tipo de métodos suelen requerir el doble del cálculo de gradientes que los métodos de Quasi-Newton (QN), pero son capaces de reducir el tiempo y la memoria requerida para determinar la matriz del Hessiano para problemas de altas dimensiones.

Su funcionamiento detallado pasa por realizar una serie de búsquedas lineales sobre la superficie del error. Esto permite determinar la dirección del paso descendente y proyectar una línea en esa dirección para localizar el mínimo de la función, actualizando los pesos una vez por cada *epoch*. La complejidad asintótica asociada a esta variante es del orden de n en función del número de entradas n ($O(n)$).

4.5.4.6. Métodos de segundo orden

El algoritmo RP calcula el Jacobiano, matriz formada por las derivadas parciales de primer orden de la función del error, con respecto a los pesos para localizar la mejor dirección hacia donde ajustarlos. Posteriormente se aplica una actualización de los pesos constante en dicha dirección.

Las técnicas de segundo orden de gradiente por su parte emplean el Hessiano, matriz cuadrada de las segundas derivadas parciales del error, con respecto a los pesos para adaptar el tamaño de paso, *step-size*, en la dirección de la actualización óptima. Expresado de forma matemática, las técnicas de segundo orden tratan de resolver la Ecuación 4.34.

$$\frac{\partial^2 E}{\partial w^2} \equiv 0 \quad (4.34)$$

Parker propuso en 1982 [Parker, 1982] la fórmula original del método de segundo orden para el algoritmo RP. Su principal inconveniente es que necesita un elevado tiempo de computación para encontrar la mejor solución. Por ello, durante años han ido apareciendo diferentes alternativas basadas en diversas suposiciones para aproximar la superficie del error, o al menos una parte de ella, mediante una curva de segundo orden. Una de estas aproximaciones, Quick Propagation (QP), asume una superficie del error cuadrática por lo que la segunda derivada es una constante mientras que otras alternativas como QN y Levenberg-Marquardt (LM) utilizan una variante del método de descenso de Newton.

En general los algoritmos de Newton presentan una complejidad superior, de límite en algunos casos de n^2 para un tamaño del vector de entrada de n ($O(n^2)$). Por ello su

aplicación se suele dar en entornos con arquitecturas y vectores de entrada con tamaños ajustados.

4.5.4.6.1. Algoritmo Quick Propagation

El algoritmo de propagación rápida QP es una variación del algoritmo RP estándar, y fue desarrollado por Fahlman en 1989 [Fahlman, 1988]. Este asume que la superficie de error tiene un comportamiento local cuadrático y emplea una aproximación a la derivada de segundo orden para actualizar los pesos, lo que acelera la búsqueda.

El computo de $\frac{\partial^2 E}{\partial w^2}$ es similar al del algoritmo RP, pero al tratarse de una variante de los métodos de Newton la actualización de los pesos se realiza mediante derivadas de segundo orden y no empleando el método del descenso del gradiente. Se ha demostrado que este algoritmo es más rápido que el RP para algunas aplicaciones, pero no en todos los escenarios. Al igual que éste, también puede quedar atrapado en mínimos locales y presenta los mismos problemas de inestabilidad. Por tanto, aunque no se le considera un método de propósito general, si es especialmente útil para entrenar de forma rápida una RNA en ciertos dominios.

4.5.4.6.2. Algoritmo Levenberg-Marquardt

El algoritmo LM es otro algoritmo de optimización no lineal basado en el uso de derivadas de segundo orden [Levenberg, 1944, Marquardt, 1963]. Se trata de la variante más utilizada en RNA dentro de las diferentes variaciones del método de Newton, siendo un algoritmo iterativo basado en los multiplicadores de Lagrange. Fue diseñado para minimizar funciones que sean la suma de los cuadrados de otras funciones no lineales, por lo que puede utilizar en aplicaciones de estimación. Por ello el algoritmo LM, tiene un excelente desempeño en el entrenamiento de RNA donde el rendimiento de la red esté determinado por el error cuadrático medio.

Con el método anterior comparte su escalado en función del número de pesos, por lo que su empleo queda restringido a redes pequeñas, del orden de pocos cientos de conexiones. El algoritmo LM es una combinación de las características del gradiente descendiente, extraídas del algoritmo RP y el algoritmo de Newton. Asume que la función subyacente es lineal y que su error mínimo se puede encontrar en un solo paso.

El algoritmo LM aproxima la matriz Hessiana usada por el método QN como el producto de la matriz Jacobiana de las derivadas parciales de primer orden con su transpuesta, tal y como se muestra en la Ecuación 4.35. Al emplear el Jacobiano, J , en lugar del Hessiano H , su cálculo es más sencillo lo que beneficia la rapidez de la

convergencia.

$$H \approx J^T J \quad (4.35)$$

4.6. Fase 4: Medidas de calidad de los modelos neuronales

La última fase de la metodología está orientada a medir la calidad y validez de los resultados de las diferentes alternativas analizadas durante el resto de fases. Se trata por tanto de una fase transversal a la que se recurrirá en diferentes tareas de la metodología para contrastar y evaluar los resultados obtenidos.

La evaluación de la calidad de cada modelo neuronal se realizará mediante los criterios expuestos en §3.5.1.1 para las RNA en clasificación y los de §3.5.1.2 para las RNA en regresión. Si bien los parámetros calculados en cada escenario permiten conocer el desempeño y rendimiento de un modelo neuronal concreto, es usual encontrarse con varios modelos alternativos con resultados parejos, lo que no permite dilucidar cual es la mejor elección. Para facilitar esta tarea se han incluido dos tareas, que permiten evaluar y comparar desde el punto de vista estadístico la capacidad de generalización de los modelos neuronales diseñados.

La primera de estas tareas incluye dos estimadores estadísticos que indican la medida de la bondad del ajuste de un modelo estadístico estimado. Estos indicadores, medidas de la calidad en base a la información, son el criterio de información de Akaike (Akaike's Information Criterion o AIC) y el principio de longitud de descripción mínima (Minimum Description Length o MDL). Ambos criterios se ajustan al criterio general usado para el cómputo del error de predicción (*EP*), expresado como la suma de dos términos [Bishop, 1995]:

$$EP = \text{error entrenamiento} + \text{termino complejidad} \quad (4.36)$$

donde el término de complejidad representa una penalización que crece a medida aumentan con los grados de libertad del modelo neuronal. Por tanto se basan en dos elementos de cómputo, uno que disminuye al mejorar el ajuste a los datos (término de fidelidad o ajuste) y otro que crece con el número de parámetros (término de penalización de la complejidad del modelo).

La segunda tarea de esta fase está encaminada a analizar la validez de los resultados obtenidos, tanto en clasificación como en predicción, por las diferentes variantes del sistema de aprendizaje diseñado. Los investigadores del área del aprendizaje automático

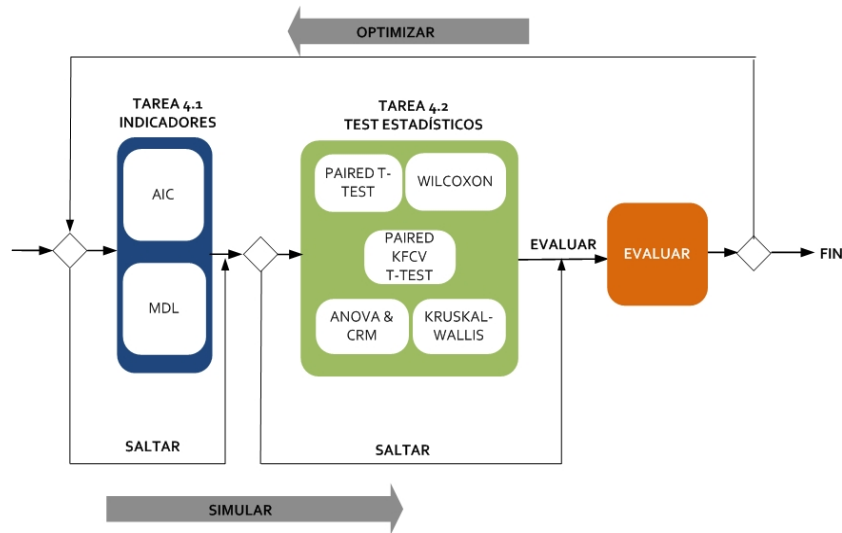


Figura 4.17: Detalle del proceso de optimización asociado a la Fase 4: Medidas de calidad

han dedicado muchos esfuerzos para comparar algoritmos sobre los mismos conjuntos de datos desde el punto de vista de los resultados, pero el estudio estadístico de múltiples alternativas simultáneas ha sido poco explotado hasta el momento. No obstante, en los últimos años han ido apareciendo varios trabajos interesantes, como el de Demsar [Demsar, 2006] y el de García y Herrera [Garcia and Herrera, 2008], donde se analizan los tests estadísticos más habituales en este dominio. Siguiendo esta tendencia, en esta investigación se evaluarán y compararán las diferentes hipótesis mediante propuestas estadísticas basadas en el estimador *t-test* y sus variantes.

Por último, en la Figura 4.17 se muestran las diferentes alternativas incluidas para evaluar la calidad de las diferentes alternativas estudiadas. Al tratarse de una fase transversal, los métodos que han sido englobados en dos bloques, pueden utilizarse durante el resto de fases y tareas de la metodología. En los siguientes apartados se concretan los objetivos y se justifica cada una de estas tareas.

4.6.1. Tarea 4.1: Indicadores de la teoría de la información

El criterio de información de Akaike es un factor estadístico que permite decidir el orden de un modelo. Fue ideado por Akaike en 1974 [Akaike, 1974] y toma en consideración tanto la medida en que el modelo se ajusta a las series observadas como el número de parámetros utilizados en el ajuste. Este criterio se utiliza para medir el compromiso entre el rendimiento y el tamaño de la red. El objetivo en este caso es minimizar este factor para producir la red con una mejor capacidad de generalización. Para su cálculo dentro

de una RNA se utiliza la siguiente ecuación:

$$AIC(k) = N \ln(MSE) + 2k \quad (4.37)$$

donde k es el número de pesos de la red, N es el número de ejemplares del conjunto de entrenamiento y MSE es el error cuadrático medio obtenido. Utilizar el criterio de Akaike para seleccionar un modelo neuronal supone decantarse por aquel que presenta un menor valor del AIC lo cual es representativo de un modelo más parsimonioso (menos parámetros) y que mejor se ajusta a los datos (menor error de predicción). Un AIC pequeño es indicativo de que el estimador representa fielmente a la muestra a la vez que castiga el posible sobreajuste derivado del empleo de un excesivo número de parámetros. Con más parámetros es más fácil conseguir un buen ajuste a la muestra de referencia, pero los resultados del mismo no suelen adaptarse bien a otras muestras distintas, i.e. el modelo no es bueno y sus conclusiones no son extrapolables.

Por su parte, el principio de longitud de descripción mínima introducido por Rissanen en 1978 [Rissanen, 1978] establece que la mejor teoría para describir un conjunto de datos es aquella que minimiza la longitud de descripción del conjunto de datos. El principio MDL ha sido empleado con éxito para la inducción en árboles de decisión, procesamiento de imágenes y modelos de aprendizaje en dominios ingenieriles no homogéneos. Este indicador facilita un criterio para la selección de modelos, teniendo en cuenta su complejidad sin la restricción de asumir que los datos constituyen una muestra de una distribución verdadera. Se trata de un criterio similar a AIC en la medida en que intenta combinar el modelo de error con el número de grados de libertad para determinar el nivel de generalización. Por tanto el objetivo es minimizar este término a partir de la siguiente ecuación:

$$MDL(k) = N \ln(MSE) + 0,5 \ln(N) \quad (4.38)$$

donde k es el número de pesos de la red, N es el número de ejemplares del conjunto de entrenamiento y MSE es el error cuadrático medio obtenido.

La utilización de estos criterios o indicadores en el ámbito de las RNA ha permitido seleccionar, entre varios candidatos, el modelo neuronal óptimo para un problema dado. El indicador AIC ha sido utilizado por los investigadores con diferentes finalidades, como reducir la subjetividad de la elección entre una arquitectura u otra [Fogel, 1990], determinar el número de neuronas ocultas [Murata et al., 1994] e incluso diseñar comités de redes [Zhao et al., 2008]. Por su parte el indicador MDL ha permitido reducir la complejidad de diferentes arquitecturas minimizando el número de pesos [Hinton and Van-Camp, 1993] y determinar el número de neuronas [Zhao and Small, 2006] y capas ocultas [Xu and Chen, 2008] para el modelo óptimo.

Por último hay que tener en cuenta que estos criterios son funciones de la razón del número de parámetros estimados al número de observaciones, k/n , y para el valor $k/n = 0$ el factor de penalización de los criterios vale uno.

4.6.2. Tarea 4.2: Test estadístico

En todo trabajo científico de tipo empírico es muy frecuente que, repitiendo un experimento en condiciones indistinguibles para el investigador, los resultados obtenidos presenten cierta variabilidad, conocida como error experimental. Por ello, en cualquier estudio científico de carácter experimental resulta crucial comparar y evaluar las características de los diferentes conjuntos de muestras o resultados que se obtienen. En el ámbito de las máquinas de aprendizaje, su investigación, desarrollo y simulación por parte de los investigadores ha incluido el empleo de diferentes métodos estadísticos para la evaluación de los resultados [Dietterich, 1998].

En esta investigación se han incluido una serie de estimadores y pruebas estadísticas de contraste, tanto paramétricos como no paramétricos, para comparar las diferentes alternativas estudiadas en cada una de las fases de la metodología de optimización. Los primeros sirven para realizar hipótesis sobre los parámetros que definen la población, e.g. poblaciones normales y tests sobre la media o la desviación típica, mientras que los segundos no se refieren a parámetros de la población; se aplican típicamente cuando no se conoce la distribución de la población o cuando su distribución es no normal. Además los tests no paramétricos son intrínsecamente robustos, i.e. funcionan relativamente bien incluso si no se cumplen sus requisitos. El listado de alternativas utilizadas incluye la prueba t de Student pareada o *t-test pareada*, Análisis de la Varianza (ANOVA o *Analysis Of Variance* en literatura inglesa), Contraste de Rangos Múltiples (CRM) y prueba de Kruskal-Wallis.

La base teórica de todas estas propuestas proviene de la distribución estadística t de Student. Se trata de una distribución de probabilidad, que surge del problema de estimar la media de una población normalmente distribuida cuando el tamaño de la muestra es pequeño, generalmente inferior a 30 [Kiessler, 2006]. Esta es la base de la popular prueba t de Student para la determinación de las diferencias entre dos medias muestrales y para la construcción del intervalo de confianza de la diferencia entre las medias de dos poblaciones. La distribución t surge, en la mayoría de los estudios estadísticos prácticos, cuando la desviación típica de una población se desconoce y debe ser estimada a partir de los datos de una muestra.

La prueba t de Student o *t-test* como todos los estadísticos de contraste se basa en el cálculo de estadísticos descriptivos previos: el número de observaciones (n), la media (\bar{X}) y la desviación típica en cada grupo (S). A través de estos estadísticos previos se calcula el estadístico de contraste experimental, siendo su fórmula para dos muestras de

igual tamaño y varianza:

$$t = \frac{\text{diferencia entre medias}}{\text{variabilidad experimentos}} = \frac{\overline{X_A} - \overline{X_B}}{S_{X_A X_B} \sqrt{2/n}} \quad (4.39)$$

La desviación típica S es calculada mediante:

$$S_{X_A X_B} = \sqrt{\frac{S_{X_A}^2 + S_{X_B}^2}{2}} \quad (4.40)$$

Normalmente en este tipo de análisis se puede establecer una hipótesis de partida (hipótesis nula o H_0), que de forma general asume que el efecto de interés es nulo. A continuación se puede evaluar la probabilidad de haber obtenido los datos observados si esa hipótesis es correcta. El valor de esta probabilidad coincide con el p-valor que nos proporciona cada test estadístico, de modo que cuanto menor sea éste más improbable resulta que la hipótesis inicial se verifique. El p-valor o nivel de significación empírico en un contraste de hipótesis está definido como la probabilidad de obtener un resultado al menos tan extremo como el que realmente se ha obtenido, suponiendo que la hipótesis nula es cierta. Cualquier nivel de significación escogido, α , inferior al p-valor comporta aceptar H_0 :

$$\begin{aligned} \text{si p-valor} &\leq \alpha \rightarrow \text{aceptar } H_1 \\ \text{si p-valor} &> \alpha \rightarrow \text{aceptar } H_0 \end{aligned}$$

Es fundamental tener en cuenta que el p-valor está basado en la asunción de la hipótesis de partida o hipótesis nula.

La prueba de contraste anterior supone desviaciones típicas similares para las poblaciones analizadas, en el caso de no cumplirse esta premisa es necesario recurrir a un análisis de la varianza de las poblaciones, mediante la prueba F o F-test. Gracias a este indicador se pueden probar una gran variedad de hipótesis siendo las más comunes:

- La hipótesis de que las medias de múltiples poblaciones normalmente distribuidas y con la misma desviación estándar son iguales. Esta es, quizás, la más conocida de las hipótesis verificadas mediante el test F y el problema más simple del análisis de varianza.
- La hipótesis de que las desviaciones estándar de dos poblaciones normalmente distribuidas son iguales.

Cuando se disponen de un conjunto con múltiples muestras y se quieren realizar diferentes contrastes de hipótesis sobre ellas bajo las mismas condiciones, utilizando la prueba de la t de Student o *t-test*, el proceso se denomina prueba de t-test pareada

(*paired t-test*). Las pruebas pareadas normalmente producen intervalos de confianza más estrictos, porque las diferencias en los errores observados se deben a diferencias entre las hipótesis. En contraste, cuando las hipótesis se prueban sobre muestras separadas, las diferencias en los dos errores podrían ser parcialmente atribuible a las diferencias en la composición de las dos muestras.

Centrando esta fase en el ámbito de la computación, uno de los problemas más habituales a la hora de comparar el rendimiento de dos algoritmos L_A y L_B , es determinar que prueba es la apropiada para evaluarlos y como determinar cuando la diferencia observada entre dichos algoritmos es significativa. Aunque hay un debate activo acerca de este punto en la comunidad investigadora, una de las alternativas más utilizadas es la expuesta por Mitchell en 1997 [Mitchell, 1997]. Su propuesta se basa en la evaluación del test estadístico t-student combinado con la técnica KFCV. Debido a que los algoritmos se evalúan sobre las mismas muestras se utiliza la variante t-test pareada con $k - 1$ grados de libertad⁴.

Para implementar esta técnica, el conjunto de datos se divide en dos, uno destinado a entrenar a L_A y L_B , conjunto de train o S_0 , y el otro para comparar la precisión, conjunto de test o T_0 . Por tanto, en este caso se mide la cantidad obtenida a partir de la siguiente ecuación:

$$error_{T_0}(L_A(S_0)) - error_{T_0}(L_B(S_0)) \quad (4.41)$$

No obstante, esta ecuación sólo permite medir la diferencia en los errores para un conjunto de entrenamiento, en lugar de tomar el valor esperado de esta diferencia sobre todas las muestras S que podrían derivarse de una distribución D . Una forma de mejorar el estimador de la Ecuación 4.41 es dividir de forma repetida los datos D_0 en conjuntos disjuntos para train y test, y posteriormente tomar la media de los errores de test para esos experimentos. A continuación se muestra, siguiendo las pautas anteriores, el procedimiento para estimar la diferencia entre el error devuelto por dos algoritmos, L_A y L_B , a partir de un conjunto de datos D_0 :

- Partir el conjunto de datos disponible, D_0 , en k conjuntos disjuntos, T_1, T_2, \dots, T_k de igual tamaño y un número de muestras superior a 30.
- Desde $i=1$ hasta k hacer
 - Utilizar T_i como conjunto de test, y el resto de datos como conjunto de train, S_i :
 - $S_i \leftarrow \{D_0 - T_i\}$
 - $h_A \leftarrow L_A(S_i)$

⁴Se trata de un estimador estadístico del número de categorías independientes en un test particular o experimento estadístico. En el caso de la t de Student se utilizan $n-1$ grados de libertad donde n es el número de muestras

- $h_B \leftarrow L_B(S_i)$
- $\delta_i = error_{T_i}(h_A) - error_{T_i}(h_B)$
- Devolver $\bar{\delta}$, donde $\bar{\delta} = \frac{1}{k} \sum_{i=1}^k \delta_i$

De esta forma el algoritmo de aprendizaje es testado sobre k conjuntos independientes de test, y la media de la diferencia de los errores $\bar{\delta}$ se devuelve como una estimación de la diferencia entre los dos algoritmos. De forma más detallada, $\bar{\delta}$ es una estimación de la cantidad:

$$E_{S \subset D_0} [error_D(L_A(S)) - error_D(L_B(S))] \quad (4.42)$$

donde S representa un conjunto de ejemplos aleatorios de tamaño $\frac{k-1}{k} |D_0|$ obtenido de forma uniforme de la distribución D_0 . El intervalo de confianza aproximado para estimar la cantidad de la Ecuación 4.42 usando $\bar{\delta}$ es:

$$\bar{\delta} \pm t_{N,k-1} S_{\bar{\delta}} \quad (4.43)$$

Donde $t_{N,k-1}$ es una constante que varía en función del nivel de confianza deseado y $S_{\bar{\delta}}$ es la desviación típica de la distribución que gobierna $\bar{\delta}$, calculada mediante la ecuación:

$$S_{\bar{\delta}} \equiv \sqrt{\frac{1}{k(k-1)} \sum_{i=1}^k (\delta_i - \bar{\delta})^2} \quad (4.44)$$

Con respecto a $t_{N,k-1}$, el primer subíndice, N , indica el nivel de confianza deseado. El segundo denominado número de grados de libertad (generalmente definido como ν) está relacionado con el número de sucesos aleatorios independientes que van a producir el valor de la variable aleatoria $\bar{\delta}$ y en este caso el valor es $k-1$. Los valores para este parámetro t pueden consultarse en la Tabla 4.5 extraída de [Kiessler, 2006]. En este caso se utilizan intervalos de confianza de dos colas o bilaterales ya que no se establece una dirección específica de la hipótesis alterna H_1 , i.e. se tienen en cuenta valores a ambos lados de la media de la distribución:

$$\begin{aligned} H_0: \mu_1 &= \mu_2 \\ H_1: \mu_1 &< \mu_2 \vee \mu_1 > \mu_2 \end{aligned}$$

Donde H_0 es la hipótesis nula, afirmación acerca del valor de un parámetro poblacional, H_1 la afirmación que se aceptará si los datos muestrales proporcionan evidencia de que la hipótesis nula es falsa. μ_1 y μ_2 son muestras respectivas de cada hipótesis.

Si bien el conjunto de t-test pareado y la validación cruzada $KFCV_{10}$ ha sido ampliamente empleado dentro del ámbito de las RNA [Tang et al., 2000, Ahn et al., 2000, West, 2000, Zhang and Subbarayan, 2002, Feng et al., 2005] para validar hipótesis,

Tabla 4.5: Valores para $t_{N,v}$ para intervalos de confianza de dos colas

	Nivel de confianza (N)			
	90 %	95 %	98 %	99 %
$v=2$	2.92	4.30	6.96	9.92
$v=5$	2.02	2.57	3.36	4.03
$v=10$	1.81	2.23	2.76	3.17
$v=20$	1.72	2.09	2.53	2.84
$v=30$	1.70	2.04	2.46	2.75
$v=120$	1.66	1.98	2.36	2.62
$v=\infty$	1.64	1.96	2.33	2.58

Tabla 4.6: Tipos de error estadísticos

Tipo I	Probabilidad de rechazar la hipótesis nula cuando esta es cierta (falsa diferencia)
Tipo II	Probabilidad de aceptar la hipótesis nula cuando esta es falsa (falsa igualdad)
Replicabilidad	Probabilidad de que dos ejecuciones de un mismo método de comparación produzca los mismos resultados (estabilidad)

su uso presenta una serie de desventajas como su tendencia a subestimar la varianza o la posible aparición de error Tipo I alto (en la Tabla 4.6 se detallan los diferentes tipos de errores estadísticos). Para mejorar esta propuesta algunos investigadores, como Dietterich [Dietterich, 1998] proponen emplear una validación cruzada repetida (*repeated k-fold cross validation*) para reducir la influencia de la partición aleatoria de los datos. Esto permite que se tengan errores Tipo I y Tipo II apropiados y una mayor replicabilidad que con la alternativa tradicional. La propuesta original de Dietterich de $5 \times 2cv$ ha sido ampliada por otros investigadores, los cuales proponen realizar 10 repeticiones sobre $KFCV_{10}$: $10 \times 10cv$ [Grochowski and Dutch, 2007, Bermejo et al., 2008] o incluso 20 repeticiones sobre $KFCV_5$: $20 \times 5cv$ [Ferri et al., 2009].

Por otro lado, en el caso de que las diferencias entre las muestras no sea normal, el t-test no es aplicable y debe recurrirse a contrastar la hipótesis de que las medianas de las distribuciones del error son iguales mediante un test no paramétrico. Para el caso de datos pareados, se puede recurrir al test de los signos para la mediana de las diferencias y al test de rangos de los signos, conocidos como test de Wilcoxon [Wilcoxon, 1945]. La prueba de los signos de Wilcoxon es una prueba no paramétrica, alternativa a la prueba t de Student, que compara la mediana de dos muestras relacionadas para determinar si existen diferencias entre ellas. Las hipótesis a contrastar son por tanto:

$$H_0: \text{Las } t \text{ medianas son todas iguales}$$

$$H_1: \text{Al menos una de las medianas es diferente}$$

En el caso de que el número de poblaciones a comparar sea mayor que dos, el análisis de la varianza de un factor o ANOVA sirve para comparar varios grupos de una variable cuantitativa. Se trata, por tanto, de una generalización de la prueba t de Student para dos muestras independientes al caso de diseños con más de dos muestras. Dicho método estadístico sirve para comparar si los valores de un conjunto de datos numéricos son significativamente distintos a los valores de otro o más conjuntos de datos, siendo por tanto de utilidad para analizar más de dos alternativas de forma simultánea [Fisher, 1950]. El caso de ANOVA con un factor se refiere al modelo (lineal) en el que la variable analizada, Y , depende de un sólo factor, F , de tal manera que las causas de su variabilidad son englobadas en una componente aleatoria que se denomina error experimental:

$$Y = F \pm \text{error} \quad (4.45)$$

De forma detallada, el análisis de la varianza ANOVA surge como una generalización del contraste para dos medias de la t de Student, cuando el número de muestras a contrastar es mayor que dos. Por tanto, el método ANOVA permite contrastar la hipótesis nula H_0 de que las medias de t poblaciones ($t > 2$) son iguales, frente a la hipótesis alternativa de que por lo menos una de las poblaciones difiere de las demás en cuanto a su valor esperado. Este contraste es fundamental en el análisis de resultados experimentales, en los que interesa comparar los resultados de t variables o factores con respecto a la variable dependiente o de interés. El método ANOVA requiere el cumplimiento de los siguientes supuestos, especialmente del último:

- Las poblaciones (distribuciones de probabilidad de la variable dependiente correspondiente a cada factor) son normales.
- Las t muestras sobre las que se aplican los tratamientos son independientes. Tiene que ver con que no haya ningún factor que altere el proceso de recogida de datos, en la práctica, se comprueba contrastando si los residuos son aleatorios.
- Las poblaciones tienen todas igual varianza o σ^2 (homocedasticidad).

Teniendo en cuenta que para cada variable X , la media poblacional es $\mu = E(X)$, entonces la hipótesis nula H_0 que se pone a prueba en ANOVA de un factor es que las medias poblacionales son iguales. Si dichas medias son iguales eso significa que los grupos no difieren y que, en consecuencia el factor analizado es independiente. La estrategia para poner a prueba la hipótesis de igualdad de medias consiste en obtener un estadístico de contraste, llamado F , que refleja el grado de similitud existente entre las medias que se están comparando. Además, cuando la hipótesis nula ha sido rechazada usando el test ANOVA, para comparar diferentes grupos de muestras se empleará el método de las menores diferencias significativas de Fisher (LSD o Least Significant Difference). Por tanto, el contraste a realizar es:

$$H_0 : \text{todas las medias son iguales } \mu_1 = \mu_2 = \dots = \mu_t = \mu$$

$$H_1 : \text{alguna de las medias es diferente } \exists \mu_j \neq \mu \quad j = 1, 2, \dots, t$$

Cuanto más diferentes sean las medias, mayor será el valor de F . Si las poblaciones muestreadas son normales y sus varianzas son iguales, el estadístico F o F -test se distribuye según el modelo de probabilidad F de Fisher-Snedecor [Kiessler, 2006]. Si se supone cierta la hipótesis de igualdad de medias, se puede conocer en todo momento la probabilidad de obtener un valor como el obtenido o mayor. Si el nivel crítico asociado al estadístico F (i.e. si la probabilidad de obtener valores como el obtenido o mayores) es menor que 0.05^5 , se rechazará la hipótesis de igualdad de medias y se concluirá que no todas las medias poblacionales comparadas son iguales. En caso contrario, no se podrá rechazar la hipótesis de igualdad y no se podrá afirmar que los grupos comparados difieran en sus promedios poblacionales.

Por otro lado, el método ANOVA se basa en la descomposición de la variación total de los datos con respecto a la media global (Suma de Cuadrados Totales o SCT). Bajo el supuesto de que H_0 es cierta, es una estimación de la varianza o σ^2 en dos partes, obtenida a partir de toda la información muestral:

- Variación dentro de las muestras (Suma de Cuadrados Residual o SCR) o Intra-grupos, cuantifica la dispersión de los valores de cada muestra con respecto a sus correspondientes medias.
- Variación entre muestras (Suma de Cuadrados Explicada o SCE) o Inter-grupos, cuantifica la dispersión de las medias de las muestras con respecto a la media global.

Además se cumple que $SCT = SCR + SCE$, y las diferencias que aparecen entre los datos, y que no son explicadas por el factor introducido, se consideran debidas al azar. Se llama error experimental a la parte de la variabilidad de los datos debida al azar y puede estimarse como la raíz cuadrada de SCR.

Si H_0 es cierta, entonces SCE será pequeña frente a SCT. De hecho, se llama porcentaje de variabilidad explicada a $(SCE = SCT) * 100$. En general, se puede afirmar que H_0 será aceptada si el porcentaje anterior no es demasiado alto. Si es suficientemente alto, se entiende que el factor que se ha introducido está realmente explicando las diferencias observadas entre los valores de la variable Y , y por tanto que efectivamente hay cierta relación entre Y y F , con lo cual H_0 es falsa. Si H_0 se rechaza, ello no implica que todas las μ_i sean distintas entre sí, sino simplemente que alguna de ellas es diferente a las demás. De hecho, pueden localizarse los diferentes grupos que aparecen entre los niveles del factor (i.e. las medias que se consideran similares entre sí) mediante el empleo del Contraste de Rango Múltiple (CRM).

Una vez contrastado que existen diferencias significativas mediante el análisis de la varianza ANOVA, es interesante conocer que niveles del factor son los que han influido más para que se de este resultado. El método más utilizado, denominado test CRM, consiste en realizar todas las comparaciones por parejas:

⁵Se trata de un nivel de significación de 0.05 o un intervalo de confianza del 95 %

$$\begin{cases} H_0 : \mu_i = \mu_j \\ H_1 : \mu_i \neq \mu_j \end{cases} \quad i, j = 1, \dots, t \quad i \neq j \quad \Rightarrow \binom{t}{2} \text{ contrastes}$$

lo que corresponde a los contrastes de la t de Student comentados con anterioridad, que tienen en este caso como estadístico experimental a (de nuevo suponiendo la homocedasticidad en todas las muestras):

$$T_{exp} = \frac{\bar{x}_i - \bar{x}_j}{\hat{S}_D \sqrt{1/n_i + 1/n_j}} \equiv t_{N-t} \quad (4.46)$$

ya que la intravarianza \hat{S}_D , es un estimador de σ^2 con $N - t$ grados de libertad.

Si los supuestos para aplicar el método ANOVA no se cumplen, se puede recurrir a otras alternativas no paramétricas que no utilicen la media como criterio de contraste. Para comprobar que las varianzas de las poblaciones son iguales se han incluido una serie de contrastes estadísticos de igualdad muy extendidos: Contraste de Bartlett [Snedecor and Cochran, 1980], Contraste C de Cochran [Cochran, 1977] y el Test de Levene [Levene, 1960]. El primero es un contraste general que supone que las poblaciones son normales y que las muestras obtenidas son independientes de igual tamaño o no. El contraste de Cochran es útil cuando la varianza muestral sea mucho mayor que el resto o cuando el número a analizar sea superior a doce. El test de Levene por su parte no necesita normalidad en las distribuciones. En los tres casos si el p-valor resultante es menor que el valor crítico (típicamente 0.05), es poco probable que las diferencias obtenidas en las variaciones de la muestra se hayan producido basándose en un muestreo aleatorio. Así, la hipótesis nula de varianzas iguales es rechazada y se concluye que existe una diferencia entre las varianzas de la población.

Una vez determinado que existe diferencia estadística significativa entre las varianzas, la prueba de Kruskal-Wallis es el método más adecuado para comparar poblaciones cuyas distribuciones no son normales [Kruskal and Wallis, 1952]. Se trata del método no paramétrico, derivado del test F, para testar la igualdad entre las medianas de un grupo de poblaciones. La razón de usar la mediana es que esta es robusta, i.e. poco sensible a datos atípicos, mientras que la media en cambio es muy sensible. Si la distribución es normal, media y mediana coinciden pero si hay discrepancia entre ambas, es preferible la mediana. Por tanto en ausencia de normalidad son relevantes los contrastes no sobre la media, sino sobre la mediana.

Incluso cuando las poblaciones son normales o cuando las desviaciones típicas de los diferentes grupos no son iguales entre sí, este contraste funciona de forma adecuada. Esta prueba, al igual que el ANOVA, realiza el contraste de forma simultánea de si las t muestras vienen de la misma población respecto a un parámetro de centralización, en este caso el rango medio. Ya que es una prueba no paramétrica, la prueba de Kruskal-Wallis no asume normalidad en los datos, en oposición al tradicional ANOVA. Si asume

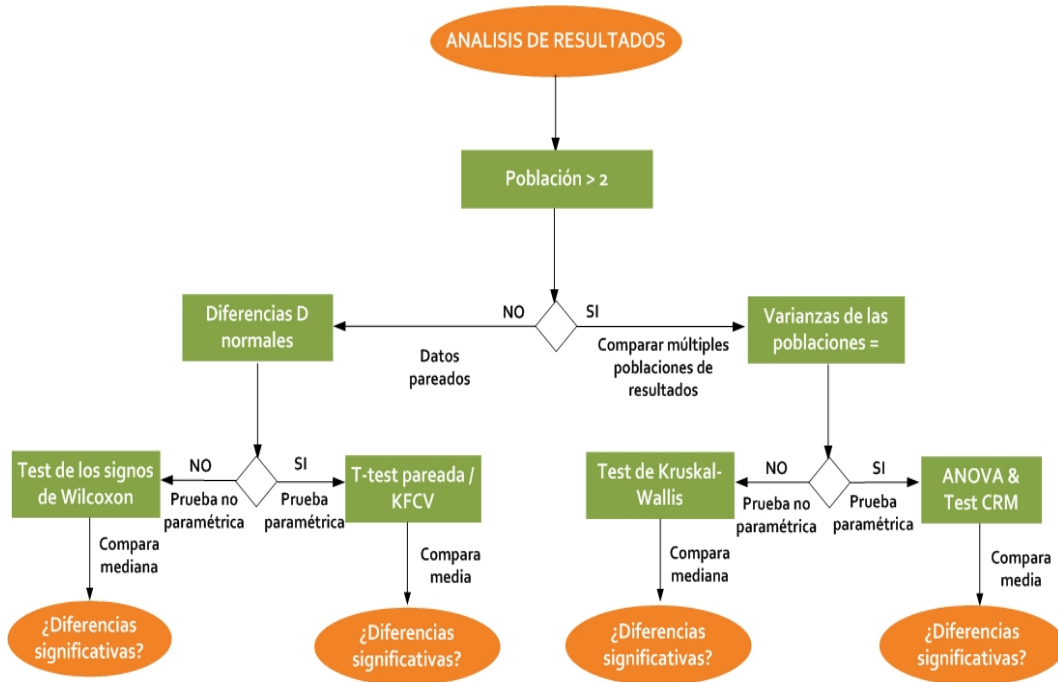


Figura 4.18: Esquema de los test de contraste de hipótesis realizados durante la fase de validación de los resultados

bajo la hipótesis nula que los datos vienen de la misma distribución. Una forma común en que se viola este supuesto es con datos heterocedásticos. No obstante, es conveniente recordar que el método ANOVA de un factor es muy robusto y sólo se ve afectado cuando las desviaciones típicas difieren en gran magnitud. La hipótesis nula de la prueba de Kruskal-Wallis es:

$$H_0: \text{Las } t \text{ medianas son todas iguales}$$

$$H_1: \text{Al menos una de las medianas es diferente}$$

En resumen, para múltiples poblaciones (> 2), el F-test en la tabla de ANOVA comprueba si hay alguna diferencia significativa entre las medias y el test CRM indica las medias que son significativamente diferentes unas de otras. Si aparecen valores atípicos o diferencias en las varianzas, se puede utilizar el test Kruskal-Wallis que compara las medianas en lugar de las medias. En este caso los diferentes gráficos ayudan a juzgar la significación práctica de los resultados, y permiten buscar las posibles violaciones a las asunciones subyacentes en el análisis de la varianza.

Por último en la Figura 4.18 se muestra la secuencia de decisiones tomadas para evaluar cada una de las propuestas incluidas en esta metodología.

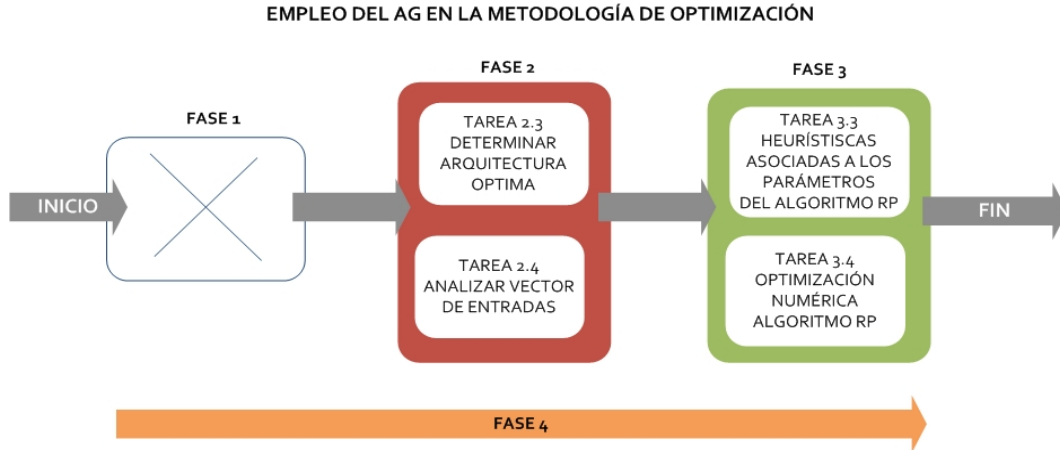


Figura 4.19: Ubicación del componente de AG dentro de la metodología de optimización

4.7. Algoritmo genético

Los AGs son métodos sistemáticos para la resolución de problemas de búsqueda y optimización, que aplican a éstos los mismos métodos de la evolución biológica: selección basada en la población, reproducción y mutación. En un AG, tras parametrizar el problema en una serie de variables, (x_i, \dots, x_n) , éstas se codifican en un cromosoma. Todos los operadores utilizados se aplicarán sobre estos cromosomas, o sobre poblaciones de ellos. Hay que tener en cuenta que un algoritmo genético es independiente del problema y se adapta a diferentes entornos y situaciones. Esto lo hace un algoritmo robusto, por ser útil para cualquier problema, pero a la vez débil, pues no está especializado en ninguno.

Las soluciones codificadas en un cromosoma compiten para ver cuál constituye la mejor solución. El ambiente, constituido por las otras posibles soluciones, ejercerá una presión selectiva sobre la población, de forma que sólo los mejor adaptados (aquellos que resuelvan mejor el problema) sobrevivan o leguen su material genético a las siguientes generaciones, igual que en la evolución de las especies. La diversidad genética se introduce mediante mutaciones y reproducción.

Para aprovechar estas características de los AG, dentro de las diferentes tareas de la metodología de optimización, se ha incluido un componente, basado en un algoritmo de tipo genético. El objetivo es optimizar diferentes parámetros y características asociadas a cada modelo neuronal. Por ello, este apartado aparece independiente de las tareas descritas anteriormente, donde se detallaban el resto de métodos y técnicas empleadas dentro de la metodología. En función de la simulación a realizar, este componente se empleará para determinar el número de neuronas ocultas (Tarea 2.3, cf. §4.4.3), optimizar el vector de entradas (Tarea 2.4, cf. §4.4.4), localizar los mejores parámetros η y μ del algoritmo RPE (Tarea 3.3, cf. §4.5.3) y como complemento a las diferentes funciones de coste basadas en la norma L_p (Tarea 3.4, cf. §4.5.4). En la Figura 4.19 se muestra la

Tabla 4.7: Parámetros y criterios empleados en el AG

Población	50
Generaciones máximas	100
Tiempo máximo de evolución (minutos)	60
Criterio Finalización	Umbral
Criterio Selección	Ruleta
<i>Crossover</i>	De un punto
Probabilidad <i>Crossover</i>	0.9
Probabilidad Mutación	0.01

ubicación de este componente dentro de la metodología de optimización diseñada.

En esta investigación el criterio usado para evaluar la capacidad de cada solución potencial es el coste más bajo obtenido durante el entrenamiento. Los criterios y valores usados se han elegido o fijado en función de las pautas dictadas por Lefebvre et al. [Lefebvre et al., 2007] y su aplicación en problemas prácticos, como la optimización de sistemas eléctricos [Zhao et al., 2008]. Las diferentes posibilidades disponibles en cada caso han sido detalladas en §2.6 y los valores empleados se muestran en la Tabla 4.7.

4.8. Sumario

Como ha quedado constatado existe una gran variedad de alternativas diferentes para parametrizar y configurar un modelo neuronal sobre la topología PMC y algoritmo RP. La literatura analizada ha mostrado que en muchos casos no existen reglas de uso general aplicables a cualquier dominio y problema, lo que hace que el proceso sea largo y tedioso hasta encontrar la solución más apta para cada caso.

Para poder aplicar todas estas variantes en un periodo de tiempo razonable, y asegurar que al finalizar el proceso se obtendrá el mejor modelo neuronal posible en función de las características de cada dominio, es perentoria una metodología que guíe todo el proceso de optimización y permita utilizar de forma razonada las diferentes posibilidades. Además, se ha incluido una serie de controles estadísticos para asegurar la calidad y validez de los modelos neuronales diseñados.

Partiendo de la premisa *modelo = arquitectura + patrones + algoritmo*, se han aplicado sobre cada término de la ecuación aquellas técnicas y métodos que han demostrado, en la literatura analizada, mejores resultados para problemas ingenieriles similares al tratado en esta investigación.

Capítulo 5

Evaluación de los Modelos y Validación de la Metodología

Una vez justificada la elección del PMC como solución de partida, el objetivo de este capítulo es evaluar, mediante la metodología de optimización propuesta, el rendimiento de los modelos diseñados en §3.5 para los escenarios del problema balístico detallados en §3.3.

El primer escenario, definido como un problema de clasificación, permitirá aplicar todas las técnicas y métodos disponibles para cada una de las tareas. Al finalizar cada una de ellas se contrastará la calidad de las diferentes alternativas de modelos neuronales para determinar el mejor candidato.

A continuación para validar la metodología propuesta se ampliará el estudio a los dos escenarios balísticos restantes. Ambos escenarios definen, debido a sus características, dos problemas de regresión que serán resueltos inicialmente mediante el uso de un PMC base y optimizados gracias a la metodología probada sobre el escenario anterior.

Por último, para todos los escenarios estudiados se comparará el rendimiento, tanto de la propuesta original del PMC como del PMC optimizado, frente a otras alternativas de aprendizaje automático.

5.1. Introducción

En el Capítulo 4 se ha definido la estrategia de optimización basada en un flujo secuencial de fases y sus correspondientes tareas. En este capítulo se procede a aplicar y validar dicha metodología sobre una serie de casos de estudio dentro del dominio del impacto balístico. El problema de partida será el caso de clasificación descrito en §3.3.1 y posteriormente las conclusiones obtenidas se validarán sobre otros dos casos de estudio de regresión, descritos en §3.3.2.

En primer lugar será necesario definir una serie de modelos de RNA basados en la arquitectura PMC, que serán tomados como referencia para posteriormente aplicar de forma gradual las diferentes alternativas que se han elegido para cada fase. La configuración inicial de estas redes base ha sido detallada dentro de §3.5, donde se justifican las decisiones tomadas para crear los modelos correspondientes al problema de clasificación y regresión. Como se ha explicado a lo largo de este trabajo, los factores que describen una RNA se pueden separar en tres categorías:

- Distribución de patrones: Detallado en §3.5.2
- Arquitectura: Detallada en §3.5.1
- Algoritmo de entrenamiento: Detallado en §3.5.3

Cada una de estas categorías se corresponde con una fase de la metodología de optimización y dentro de las fases se incluyen tareas con diferentes alternativas que se ejecutan de forma secuencial o en paralelo. El objetivo de este proceso es intentar reducir las posibles interacciones que aparecen si se aplican, de forma combinada y simultánea, varias técnicas y métodos sobre un mismo PMC más complejo. El empleo de una metodología para entrenar y testear RNA ha sido realizada con éxito por otros autores en diferentes dominios, como la predicción del comportamiento de motores diesel [Hernandez-Lopez, 2006] o en cardiología [Ari and Saha, 2009].

Las medidas de calidad utilizadas, comunes a todos los modelos ya sean de clasificación o de regresión, han sido descritas en §4.6. Estos criterios unidos a los tradicionales de clasificación, cf. §3.5.1.1, y predicción numérica, cf. §3.5.1.2, permiten comparar de una manera más eficiente todas las alternativas evaluadas. Los criterios mostrados variarán en función de las características de la tarea analizada en cada momento dentro de esta investigación.

Por último recordar que el aprendizaje es un proceso estocástico que no sólo depende de los parámetros de aprendizaje, sino también de las condiciones iniciales. Por ello, si se quieren comparar los tiempos de la convergencia de red (i.e. cuánto más rápido es una regla de actualización con respecto a otra) o los valores finales del error MSE después de un número de iteraciones, no es suficiente con probar la RNA una sola vez. Lo más conveniente es realizar varias simulaciones sobre cada red, con condiciones iniciales aleatorias. Cuando el objetivo es comparar diferentes algoritmos de entrenamiento o configuraciones de red, es una práctica común mostrar la media de los resultados, i.e. presentar la curva media de aprendizaje a través de los diferentes ensayos o los valores óptimos de los parámetros estudiados. Por tanto, las simulaciones realizadas incluyen un proceso de entrenamiento y testeo que se repite veinte veces con los pesos de las conexiones de la red inicializados de forma aleatoria, esto permite aumentar las validaciones, y mejorar la robustez de los modelos al hacer que los resultados sean

Característica	Descripción
Reparto patrones por conjuntos (#1 y #2 - #3)	80%/20% - 33%/33%/33%
Técnica de distribución de patrones	HOCVE
Entradas	$V_{ent} = \{M_b, L_b, R_b, V_b, H_p, T_p, T_b\}$
Salidas	$S_{clf} = \{P_b, D_b\}$
Arquitectura	PMC
Capas ocultas	1
Neuronas ocultas	4
Función de activación (entrada-oculta-salida)	$f_{iden} - f_{tanh} - f_{tanh}$
Algoritmo de entrenamiento	RPE
Parámetros de entrenamiento	$\mu=0.7$, $\eta = 1$ en capa oculta y $\mu=0.3$, $\eta=0.4$ en capa salida
Función de coste	MSE simple
Actualización de pesos	<i>Batch</i>
Inicialización de los pesos	Heurística de Haykin
Criterio de convergencia (#1 - #2 - #3)	<i>Epochs</i> [1000-150000] - <i>MSE</i> < $10^{-0,5}$ y <i>Epochs</i> [1000-150000] - Parada temprana

Tabla 5.1: Resumen de características de la red base para clasificación

estadísticamente más válidos [Hernandez-Lopez, 2006, Levy-Mangin et al., 2008].

5.2. Evaluación del modelo neuronal para el problema de clasificación

5.2.1. Modelo neuronal base

Uno de los objetivos de esta tesis doctoral es la optimización de los modelos de PMC dentro del dominio del impacto balístico. Para ello es necesario un punto de partida que permita optimizar y evaluar diferentes alternativas. En este caso es necesario el diseño de un PMC inicial y su posterior evaluación sobre un problema específico. Si bien la arquitectura base de este tipo de RNA ha sido explicada en §3.5.1.1, en la Tabla 5.1 se resumen sus características principales para el problema de clasificación. Esta configuración se denomina configuración de prueba base, C_0 , y presenta características divididas en tres bloques: patrones (P_0), arquitectura (A_0) y algoritmo (R_0).

Como algoritmo de aprendizaje se ha elegido la variante RPE debido a sus ventajas y su amplia utilización en toda clase de problemas [Priddy and Keller, 2005]. La reseña #1

Tabla 5.2: Resultados preliminares obtenidos para el PMC base (% acierto tras 20 repeticiones)

<i>Epochs</i> máximos	1000	3000	9000	27000	81000	150000
Parada por <i>Epochs</i>	91.2330	90.2727	91.2734	90.1679	91.3624	90.0597
<i>Epochs</i> máximos	1000	3000	9000	15000	30000	45000
Parada por MSE	90.3330	91.4358	91.5782	92.0459	92.1947	92.1836
<i>Epochs</i> máximos	100	1000	5000	10000	20000	30000
Parada temprana	90.7374	90.8444	91.2251	92.5822	92.6627	92.6154

dentro de la Tabla 5.1 significa entrenamiento convencional, #2 entrenamiento combinado con parada por valor mínimo de MSE o *epochs* y #3 entrenamiento con parada temprana. Para #1 y #2 se usan los conjuntos de *train* y *test*, mientras que para #3 además se incluye el conjunto *CV*. Para #2 el valor MSE de 10^{-5} se ha aproximado a partir de los trabajos de Hernandez et al. [Hernandez-Lopez, 2006] y Levy et al. [Levy-Mangin et al., 2008]. El reparto de patrones en cada conjunto se realizará siguiendo la técnica HOCVE.

Los resultados obtenidos con el clasificador PMC para cada criterio de parada se muestran en la Tabla 5.2. Para cada criterio se incluye la fila *epochs* máximos para conocer el tiempo necesario para realizar el aprendizaje, Como se puede observar la calidad de la clasificación con los tres criterios de parada es superior al 90%. No obstante, al utilizar el criterio por *epochs* el entrenamiento se alarga en el tiempo de forma considerable y el rendimiento es más inestable. Los otros dos criterios, #2 y #3, se utilizan de forma combinada con el número de *epochs*. Cabe destacar que el tiempo de computación se reduce de forma considerable para el criterio #3 cuando el número de *epochs* es elevado (≥ 20000) ya que el entrenamiento finaliza al cumplirse el criterio adicional, aumento del error de validación. Para el criterio #2 el valor máximo de *epochs* es más elevado, pero inferior al primero, ya que la cota mínima para MSE se alcanza con menos de 30000 *epochs*. La ventaja adicional de este comportamiento para #2 y #3 es que el rendimiento de la red es más estable, obteniendo porcentajes de acierto similares para valores altos de ciclos de entrenamiento.

5.2.2. Fase Previa. Comparativa de alternativas al PMC

Una de las preguntas que pueden surgir al comenzar la fase de evaluación es porque se ha elegido un PMC y no otro clasificador basado en la teoría de las máquinas de aprendizaje o aprendizaje automático. Si bien a lo largo de esta tesis doctoral se ha justificado su elección, en este apartado se compara su calidad como clasificador frente a otras alternativas dentro del escenario del impacto balístico.

Los criterios de parada, los porcentajes de reparto y la técnica de distribución de patrones utilizados serán iguales a los especificados en la Tabla 5.1.

La Figura 5.1 muestra la calidad media de los clasificadores con los tres criterios de parada comentados: *epochs*, MSE y parada temprana. En el primer caso, los mejores clasificadores han sido el PMC y el RNN, si bien la primera opción es superior en un 1.3052% en media sobre la segunda y la mejor predicción del PMC (81000 *epochs*) supera en 0.7647% a la del RNN (81000 *epochs*). Con respecto al resto de alternativas cabe destacar el comportamiento del clasificador SOFMS, que obtiene el punto de inflexión en el aprendizaje con 81000 *epochs*, aunque la calidad de las predicciones es inferior al PMC en 1.8102%. Es interesante también la estabilidad que presenta del clasificador SVM, donde la peor predicción (1000 *epochs*) difiere tan solo en 0.3230% respecto de la mejor (27000 *epochs*). Por último, en todos los clasificadores analizados la predicción decae de forma clara a partir de 150000 *epochs*, lo que indica signos de sobreaprendizaje.

Los resultados para el segundo criterio, parada por MSE, corroboran los obtenidos anteriormente pero aportan la ventaja de reducir el tiempo de entrenamiento con algunos clasificadores como PMC, RNN y SOFMS. La calidad del PMC es similar al caso anterior, aunque en este caso el mejor resultado medio se obtiene con 30000 *epochs*, es ligeramente superior al obtenido por el criterio por *epochs* (0.8212% de mejora), presentando un comportamiento muy estable desde los 15000 *epochs*. El resto de clasificadores tiene un comportamiento similar al estudiado previamente pero el sobreaprendizaje se reduce, estabilizándose las predicciones con un elevado número de *epochs*.

El tercer criterio analizado, parada temprana, vuelve a ratificar las conclusiones anteriores. En este caso los tiempos de simulación se reducen para todas los clasificadores de forma considerable al necesitar un menor número *epochs* para finalizar el aprendizaje. La gráfica muestra el número de *epochs* planificados lo que no significa que se alcancen en todos los casos. El comportamiento del PMC vuelve a ser estable a lo largo de las simulaciones, especialmente con un número elevado de *epochs*. En comparación con los otros criterios de parada la calidad del modelo es similar, pero aporta como ventaja la rapidez del entrenamiento y reduce la aparición del sobreaprendizaje. Estudiando el resto de clasificadores cabe resaltar la mejora de ELN y la estabilidad de SOFMS y RBFN al aplicar este criterio de parada.

5.2.2.1. Medidas de calidad de los clasificadores

Tabla 5.3: Fase Previa. Análisis aplicando la Tarea 4.1: Indicadores AIC y MDL

Arquitectura	AIC	MDL	Arquitectura	AIC	MDL
PMC	-25.1340	-23.0344	ELN	49.9282	61.1109
RNN	54.2559	64.7125	SOFM	556.5507	602.1528
RBF	619.9718	671.0927	SVM	1462.1178	1569.5878

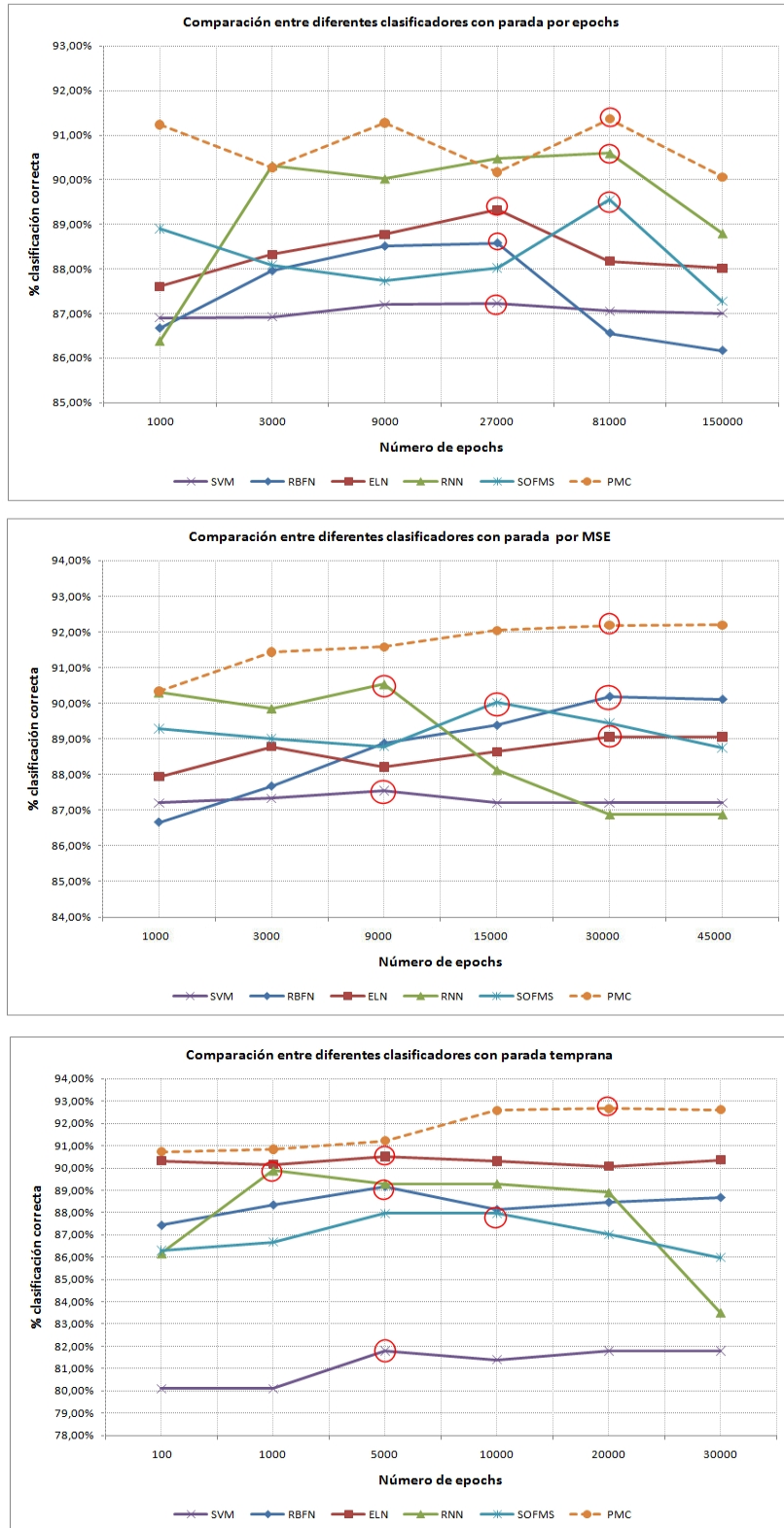


Figura 5.1: Fase Previa. Comparación entre los diferentes clasificadores con diferentes criterios de parada (20 repeticiones)

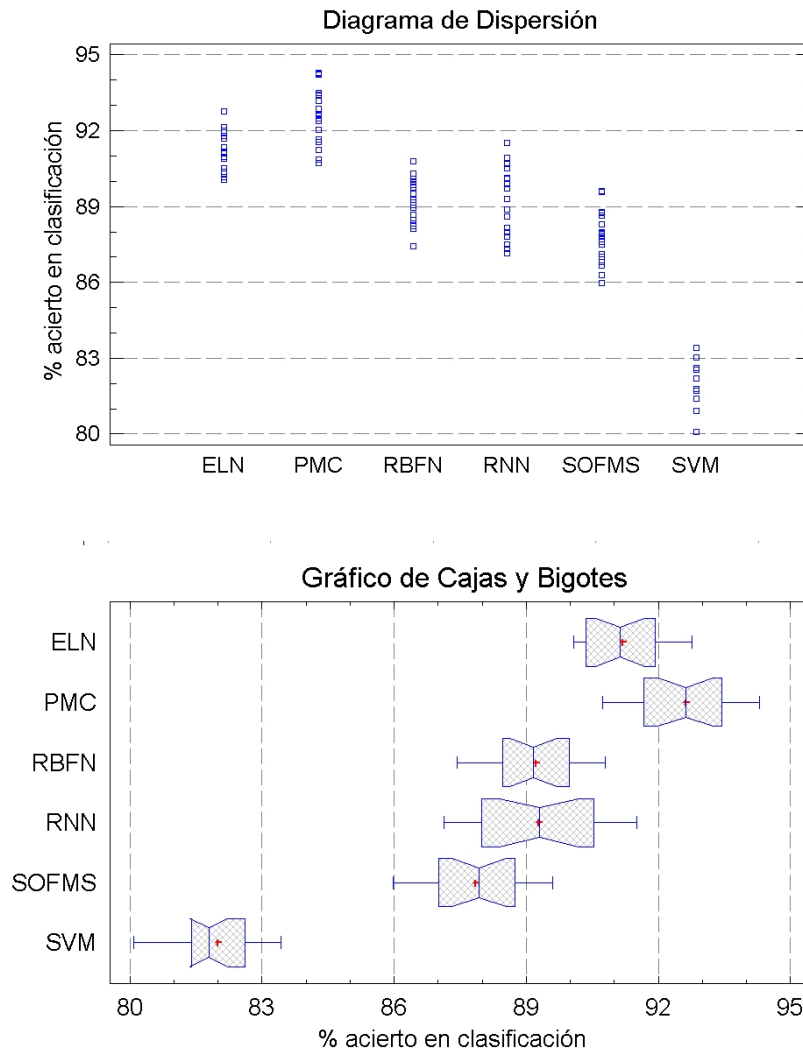


Figura 5.2: Fase Previa. Diagramas de dispersión y de caja

Para corroborar las conclusiones anteriores se ha realizado un análisis enfocado a medir la calidad de cada clasificador. Para comparar los grados de bondad del ajuste de cada propuesta, se incluyen los dos criterios estadísticos basados en la teoría de la información descritos en §4.6. Para ello se han tomado los resultados obtenidos con el criterio de parada temprana, debido a su precisión, rapidez y reducción del sobreaprendizaje. En la Tabla 5.3, se muestran los resultados obtenidos para ambos criterios ordenados por calidad, i.e. un menor AIC y MDL. Como se observa el modelo neuronal basado en el PMC presenta un mejor comportamiento y ajuste sobre los datos desde el punto de vista de estos indicadores estadísticos.

En la Figura 5.2 se incluyen los diagramas de dispersión y de caja asociados a los resultados. El primero describe el comportamiento del conjunto de muestras obtenidas

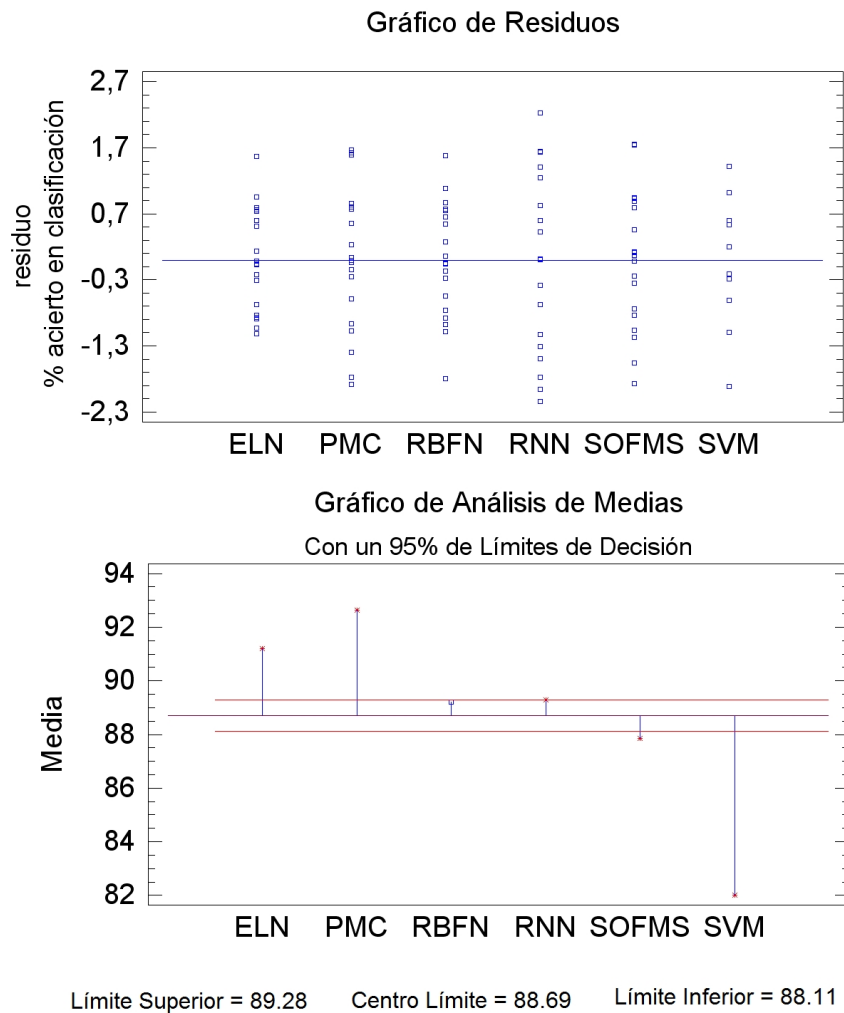


Figura 5.3: Fase Previa. Análisis de los residuos y la media

para cada clasificador a través de una nube de puntos. La utilidad del segundo radica en que permite, mediante una simple inspección visual, tener una idea aproximada de la tendencia central (a través de la mediana), de la dispersión (a través del recorrido intercuartílico), de la simetría de la distribución (a través de la simetría del gráfico) y de los posibles valores atípicos de cada clasificador. La parte rectangular del gráfico de caja se extiende desde el cuartil inferior hasta el superior, cubriendo la mitad central de cada muestra. La línea central dentro de la caja describe la localización de la mediana de la muestra y la media aparece representada por una cruz. En el gráfico también se incluye una muesca para la mediana, cuya anchura indica de forma aproximada el intervalo de confianza del 95%. Como dicha anchura es similar para todos los casos, el clasificador RNN es el único que difiere ligeramente, se induce a pensar que no hay diferencia estadística significativa entre las medianas para dicho nivel de confianza. Los bigotes se extienden desde la caja hasta los valores mínimo y máximo de la muestra,

Tabla 5.4: Fase Previa. Análisis aplicando la Tarea 4.2: Contraste de la varianza

Contraste	Valor	p-valor
C de Cochran	0.2828	0.1162
Bartlett	1.0644	0.2523
Test de Levene	1.6996	0.1408

excepto los puntos externos o los puntos externos lejanos, los cuales se representan por separado. Los primeros quedan a más de 1.5 veces el rango intercuartílico y los segundos a más de 3.0 veces el rango intercuartílico por encima o por debajo de la caja. En este caso, no hay puntos externos ni puntos externos lejanos.

Tabla 5.5: Fase Previa. Análisis aplicando la Tarea 4.2: Método ANOVA

Fuente	Sumas de Cuadrados	Grados Libertad	Cuadrado Medio	Cociente-F	P-Valor
Entre grupos (SCE)	1279.57	5	255.915	149.011	0.0000
Intra grupos (SCR)	185.481	108	1.71742		
Total (SCT)	1465.06	113			

La Figura 5.3 incluye la gráfica de residuos y la del análisis de medias de cada uno de los clasificadores. La primera muestra los residuos obtenidos para cada una de las alternativas. Los residuos son iguales a los valores del porcentaje de acierto menos el valor medio para el grupo del que proceden y muestran que la variabilidad dentro de cada alternativa es aproximadamente la misma. La segunda realiza el análisis de medias para cada una de las seis muestras junto con la media global y el 95% de límites de decisión. Las muestras que se encuentran fuera de los límites de decisión, ELN, PMC y SVM, son significativamente diferentes de la media global. Además el análisis de contraste de la varianza mostrado en la Tabla 5.4, refleja que los tres p-valores calculados son superiores a 0.05, por lo que no hay diferencia estadísticamente significativa entre las desviaciones típicas para un nivel de confianza del 95.0%.

Como se cumplen las condiciones para aplicar el método ANOVA, se ha incluido dicho estudio estadístico paramétrico para determinar una razón entre las diferencias observadas y comprobar diferentes hipótesis a partir del uso de la prueba t-test. Este análisis indicará si existen diferencias significativas entre las diferentes medias y detectará cuales son las diferencias entre los distintos grupos de resultados mediante un test CRM.

La Tabla 5.5 muestra el método ANOVA aplicado sobre los resultados obtenidos por

cada uno de los clasificadores. La varianza del porcentaje de acierto se descompone en dos componentes: un componente entre grupos y un componente dentro de los grupos. El F-ratio, que en este caso es igual a 149.011, es el cociente de la estimación entre grupos y la estimación dentro de los grupos. Puesto que el p-valor del test F es inferior a 0.05, hay diferencia estadísticamente significativa entre las medias del porcentaje de acierto para cada clasificador para un nivel de confianza del 95.0%.

Tabla 5.6: Fase Previa. Análisis mediante la Tarea 4.2: Test CRM

Contraste	Diferencias	+/- Límites
ELN - PMC	*-1.4349	0.8427
ELN - RBFN	*1.9714	0.8427
ELN - RNN	*2.5353	0.8427
ELN - SOFMS	*3.3415	0.8427
ELN - SVM	*9.1845	0.8427
PMC - RBFN	*3.4063	0.8427
PMC - RNN	*3.9702	0.8427
PMC - SOFMS	*4.7764	0.8427
PMC - SVM	*10.619	0.8427
RBFN - RNN	0.5639	0.8427
RBFN - SOFMS	*1.3701	0.8427
RBFN - SVM	*7.2131	0.8427
RNN - SOFMS	0.8062	0.8427
RNN - SVM	*6.6491	0.8427
SOFMS - SVM	*5.8429	0.8427

Para determinar las medias que son significativamente diferentes unas de otras, se ha realizado un test de rangos múltiples o test CRM. La Tabla 5.6 muestra la diferencia estimada entre cada par de medias. En la columna diferencias, el asterisco que se encuentra al lado de los trece pares, del total de $\binom{6}{2}=15$ combinaciones, indica que éstos muestran diferencias estadísticamente significativas a un nivel de confianza 95.0%. La columna límites define la mitad de la anchura de los intervalos de confianza para cada diferencia de medias de los diferentes clasificadores, por lo que dichos intervalos vendrán definidos como *Diferencia* \pm *Límites*. En la Tabla 5.7, se identifican cinco grupos homogéneos según la alineación del símbolo \checkmark en la columna. Dentro de cada columna, los niveles que tienen \checkmark forman un grupo de medias entre las cuales no hay diferencias estadísticamente significativas. El método utilizado para discernir entre las medias es el procedimiento de las menores diferencias significativas de Fisher (LSD). Con este método, hay un 5.0% de riesgo de considerar cada par de medias como significativamente diferentes cuando la diferencia real es igual a cero.

Como ha quedado demostrado los resultados del PMC, además de ser superiores en media, son significativamente diferentes y no presentan homogeneidad con el resto de clasificadores. Además si se tiene en cuenta el estimador de referencia, la media del

Tabla 5.7: Fase Previa. Análisis aplicando la Tarea 4.2: Grupos homogéneos

Clasificador	Media (%)	Grupos homogéneos			
SVM	82.0045	✓			
SOFMS	87.8474		✓		
RNN	88.6537	✓	✓		
RBFN	89.2176			✓	
ELN	91.1891				✓
PMC	92.6239				✓

porcentaje de acierto, el rendimiento del PMC es el más acertado. Por lo tanto se ha demostrado la validez de un PMC para la resolución del problema del impacto balístico, ratificándose las conclusiones obtenidas en los trabajos de Garcia et al. [Garcia-Crespo et al., 2006] y Fernandez [Fernandez, 2007]. En los siguientes apartados se trabajará sobre esta arquitectura para optimizar su utilización dentro de este dominio.

5.2.3. Fase 1. Tratamiento de los patrones

En este apartado se incluyen los resultados relacionados con las tareas encaminadas a estudiar la influencia de los patrones de entrada sobre la calidad de los PMC. Esta fase se ha desglosado en tres tareas secuenciales que toman como configuración de referencia la del modelo neuronal base C_0 (cf. §5.2.1). La configuración asociada a esta fase se denomina configuración de prueba uno, C_1 , y vuelve a estar dividida en tres bloques: patrones (P_1), arquitectura (A_0) y algoritmo (R_0).

5.2.3.1. Tarea 1.1: Normalización de los datos

Como ha quedado claro en §2.4.2.3 la forma en la que se presentan los datos al PMC afecta a su capacidad de generalización. En la mayoría de ocasiones es necesario realizar un procesado previo de los mismos antes de utilizarlos. La normalización de los datos se ha demostrado como un paso útil a la hora de afinar la precisión de la red y reducir el tiempo de cálculo [Sola and Sevilla, 1997]. Esto permite en muchos casos agilizar el proceso de entrenamiento y aumentar su eficiencia ya que los datos que se le pasan a la red se encuentran próximos a la función de activación de cada neurona.

Con el objetivo de homogeneizar los valores usados durante el aprendizaje y prueba de la red se han incluido los dos tipos de normalización detallados en §4.3.1: *z-score* y *min-max*. Los resultados obtenidos con todos los criterios de parada se muestran en la Tabla 5.8.

Aparentemente, la normalización *z-score* es la que mejor comportamiento presenta en

Tabla 5.8: Tarea 1.1. Resultados obtenidos por criterio de parada (20 repeticiones)

Normalización	Criterio de Parada	Mejor %	Peor %	Acierto medio % (epochs)
<i>Z-score</i>	<i>Epochs</i>	91.3624	90.0597	91.3624 (81000)
<i>Z-score</i>	MSE	92.1947	90.3330	92.1947 (26578)
<i>Z-score</i>	Parada temprana	94.0816	91.2374	92.6445 (18868)
<i>Min-Max</i>	Epochs	90.3624	89.1679	89.8948 (81000)
<i>Min-Max</i>	MSE	91.8836	90.2330	90.9119 (21926)
<i>Min-Max</i>	Parada temprana	93.2846	90.9021	92.1499 (19293)

Tabla 5.9: Tarea 1.1 Análisis estadístico para el criterio de parada temprana (20 repeticiones)

Estadístico	(medido en % acierto)		(medido en % error)
	<i>Min-Max</i>	<i>Z-Score</i>	Diferencias MinMax-ZScore
Media	92.1499	92.6445	0.4946
Varianza	0.5099	0.6786	1.0589
Desviación típica	0.7141	0.8237	1.0290
Mínimo	90.9021	91.2374	-0.8222
Máximo	93.2846	94.0816	2.1801
Rango	2.3825	2.8442	3.0023
Asimetría tipificada	-0.0884	-0.2831	0.3953
Curtosis tipificada	-0.8328	-0.84927	-1.3223

el PMC, con un porcentaje de acierto medio superior en todas las variantes analizadas. En el caso de parada por *epochs* el umbral de acierto se obtiene con un valor bajo de ciclos de entrenamiento. Para los otros dos criterios de parada no se llega nunca a completar el número máximo de *epochs*, al cumplirse antes el criterio adicional de finalización, ya sea el error MSE o el error de validación.

En la Figura 5.4 se muestra el diagrama de histogramas y de caja. El primero muestra dos histogramas, uno para cada conjunto de muestras, con el histograma de la segunda muestra invertido debajo del de la primera, lo que permite visualizar la forma de las dos distribuciones asociadas a cada criterio de inicialización. Por su parte en el diagrama de caja se muestran entre otros datos de interés los valores mínimo y máximo, la mediana (línea horizontal), la media (símbolo cruz) y los posibles valores atípicos. La muesca de la caja indica el intervalo de confianza del 95% y permite aproximar entre que dos valores se estima que estará cierto valor desconocido para esa determinada probabilidad de acierto.

5.2.3.1.1. Medidas de calidad de los resultados

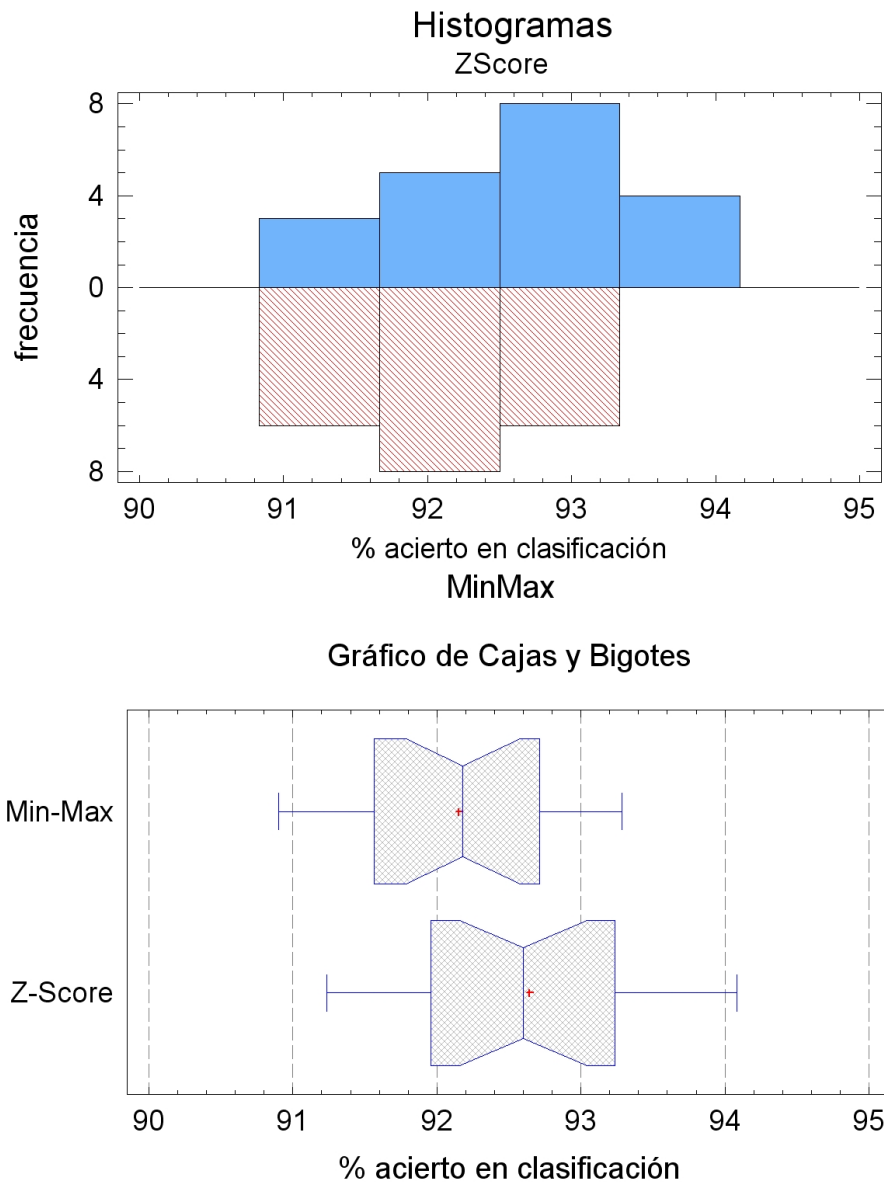


Figura 5.4: Tarea 1.1. Diagramas de histogramas y de caja

Este apartado está destinado a contrastar si dos muestras dadas provienen de la misma población, i.e. si dos muestras dadas provienen de poblaciones con la misma distribución, lo que se conoce como contraste de homogeneidad. En este caso se disponen de 20 muestras pareadas en cada población, una correspondiente al criterio *min-max* y otra al *z-score*, y la hipótesis de partida H_0 será que la diferencia entre los errores muestrales de ambas propuestas tiene media cero.

El objetivo es dar mayor validez a las inferencias obtenidas anteriormente determinando

si existen diferencias significativas entre los resultados obtenidos con el mejor criterio de parada analizado, parada temprana. Los resultados para cada uno de los criterios de inicialización se muestran en la Tabla 5.9 analizados desde un punto de vista estadístico. De particular interés son los estadísticos asimetría estandarizada y curtosis estandarizada, ya que pueden usarse para determinar si las muestras proceden de distribuciones normales. Los valores de estos estadísticos fuera del rango de -2 a +2 indican salidas significativas de normalidad, lo que tendería a invalidar las pruebas que comparan las desviaciones normales. En este caso, ambos valores de asimetría y curtosis estandarizada están dentro del rango esperado, lo que facilita el análisis de calidad de resultados realizado a continuación.

Al tratarse de experimentos que se han realizado bajo las mismas condiciones las muestras pueden considerarse pareadas, por lo que es interesante obtener las diferencias del error para cada una de ellas. Los valores de asimetría y curtosis estandarizada están dentro del rango esperado por lo que se puede determinar que no existen evidencias de un posible alejamiento de la normalidad entre las diferencias de los errores. Por tanto como contraste de homogeneidad para las dos propuestas de normalización se incluye la prueba t-test pareada:

- Contraste t:
 - Hipótesis nula H_0 : media = 0.0
 - Hipótesis alternativa H_1 : no igual
 - Estadístico t = -2.1496
 - P-valor = 0.0416
 - Se rechaza la hipótesis nula para alpha = 0.05

Este contraste de homogeneidad se basa en la prueba t-test cuya hipótesis nula es que la diferencia media de MinMax-ZScore sea igual a 0 frente a la hipótesis alternativa de que no lo sea. Puesto que el p-valor para este test es inferior a 0.05, se puede rechazar la hipótesis nula para un nivel de confianza del 95.0%.

En resumen existen evidencias estadísticas que determinan las diferencias significativas entre los resultados de cada criterio de normalización. Por tanto esta Tarea 1.1 deja como consecuencia el uso de la normalización *z-score* durante el pre-procesado de la información.

5.2.3.2. Tarea 1.2: Distribución de los patrones

La limitada información disponible en ciertos dominios es en ocasiones el mayor problema para intentar recrear esta clase de problemas con herramientas de aprendizaje automático, e.g. medicina, astronomía, ciencia espacial, ciencia nuclear, etc. Además, la necesidad de dividir en varios conjuntos los datos disponibles conlleva que la cantidad

que se puede utilizar para entrenar es aún más reducida.

Para paliar la falta de recursos existen técnicas y métodos estadísticos encaminados a intentar incrementar la información que se dispone. En su ausencia o como complemento a estas alternativas, el diseñador de la red puede decidir como repartir la información disponible para maximizar el rendimiento del PMC. Como se ha explicado en §4.3.2, existen varias propuestas para realizar la distribución de los datos en los diferentes conjuntos, pero sin llegar ninguna a imponerse como norma o pauta a seguir en todas las situaciones. En general se considera que la capacidad predictiva de una RNA aumenta cuantos más patrones se utilizan durante el entrenamiento. Sin embargo, la mejora sigue la ley del rendimiento decreciente a medida que decae la mejora al aumentar el tamaño del conjunto de *train*.

Por tanto, un aspecto de gran utilidad al utilizar un PMC en este tipo de dominios, es estudiar la evolución de su habilidad para predecir y clasificar en función de la distribución y utilización de los datos en cada uno de los conjuntos. La presente tarea tiene como objetivo determinar el mejor reparto posible de ensayos entre los diferentes subconjuntos de *train*, test y validación. Para los criterios de parada #1 y #2 se usarán los conjuntos de *train* y test, mientras que para #3 se añadirá el conjunto CV para implementar la parada temprana. Para cada criterio de parada las simulaciones anteriores han determinado sobre un PMC con HOCVE y normalización *z-score* las siguientes configuraciones clásicas:

- #1. Reparto de 80 %-20 % (train-test) y 81000 *epochs*.
- #2. Reparto de 80 %-20 % (train-test) y 30000 *epochs*.
- #3. Reparto de 33 %-33 %-33 % (train-test-validacion) y 20000 *epochs*.

Para buscar la mejor alternativa en cada caso se seguirán las pautas descritas por Crowther y Cox para la distribución de los ensayos [Crowther and Cox, 2005]. Para #1 y #2 los patrones se repartirán de forma aleatoria entre los conjuntos de *train* y test, mientras que para #3 un 25 % se reserva para validación y los restantes para los otros dos conjuntos. Además se realizarán para cada simulación 20 repeticiones con diferentes patrones en cada conjunto asegurando así la independencia del clasificador sobre los datos usados.

La Figura 5.5 muestra los resultados obtenidos para los criterios de parada *epochs* y MSE. En este caso la evolución del acierto en los criterios estudiados coincide con los postulados de Boonyanunta y Zeephongsekul [Boonyanunta and Zeephongsekul, 2004] y Crowther y Fox [Crowther and Cox, 2005, 2006]. A partir de las líneas de tendencia asociadas a cada criterio de parada se verifica las tesis expuestas en las investigaciones anteriores; el poder predictivo crece rápidamente hasta el punto A, a continuación empieza a decelerar hasta el punto B. Sin embargo en la última etapa, desde B hacia delante, no se observan mejoras significativas produciéndose cierto estancamiento e

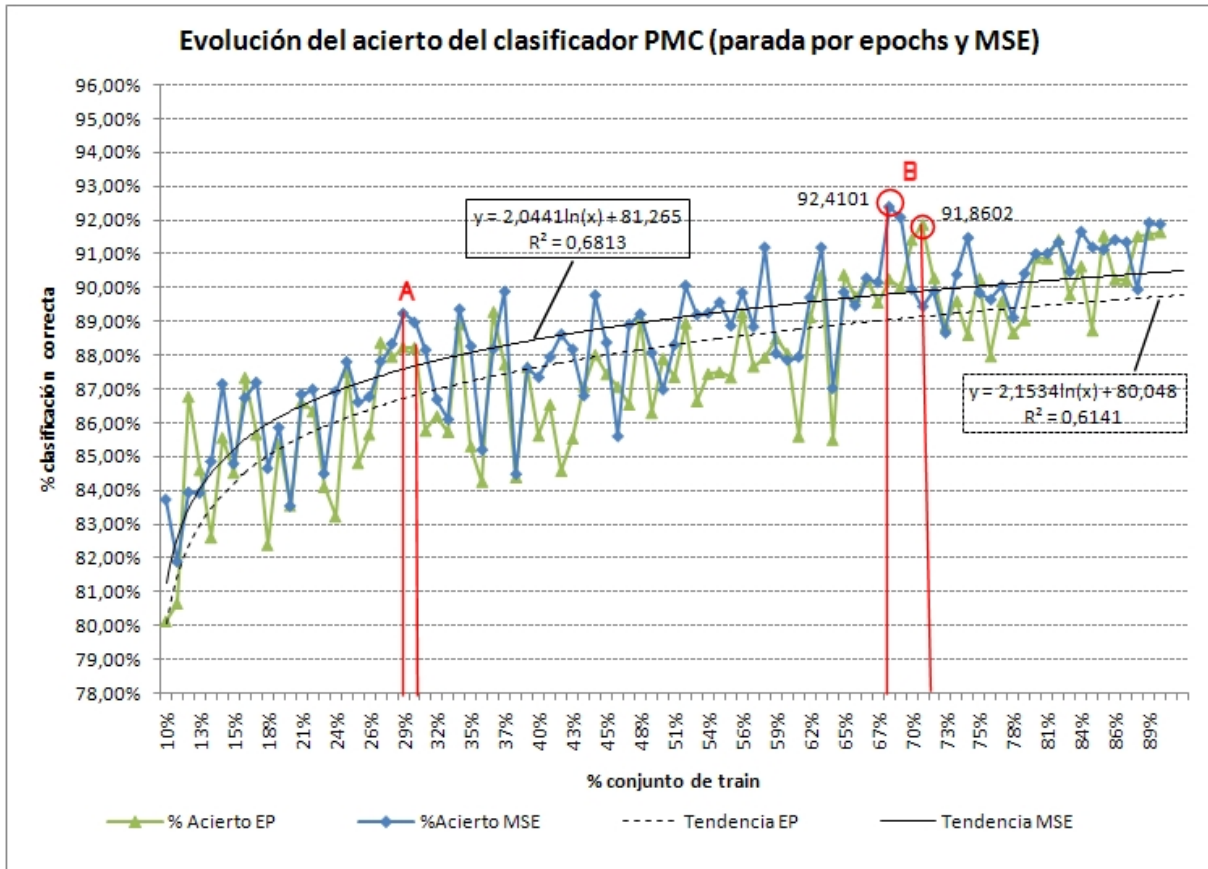


Figura 5.5: Tarea 1.2. Evolución del porcentaje medio de acierto del clasificador en función del número de datos para entrenar para #1 y #2 (20 repeticiones)

incluso empeorando la tasa de acierto.

En los dos casos el valor de máximo acierto coincide con las aserciones de Boonyanunta y Zeephongsekul, al fijarse el ratio para el conjunto de *train* cerca del 70 % de los patrones disponibles (71 % para *epochs* y 68 % para MSE). A partir de este punto B el rendimiento del clasificador se estanca y los resultados se estabilizan.

El criterio de parada temprana analizado mostrado en la Figura 5.6 presenta un comportamiento similar. En este caso el número disponible de ensayos para entrenar es menor porque una parte se utiliza en el conjunto adicional de validación, un 25 % del total. El valor cumbre se consigue con un reparto del 55 % para *train*, dejando el restante para validar el entrenamiento y testear la red. A partir del punto B la red se estabiliza manteniendo su tasa de acierto en valores superiores al 92 %, aunque se muestra cierto estancamiento del clasificador. La mejora que se obtiene desde el punto B hasta la mejor clasificación es del 1.4961 %.

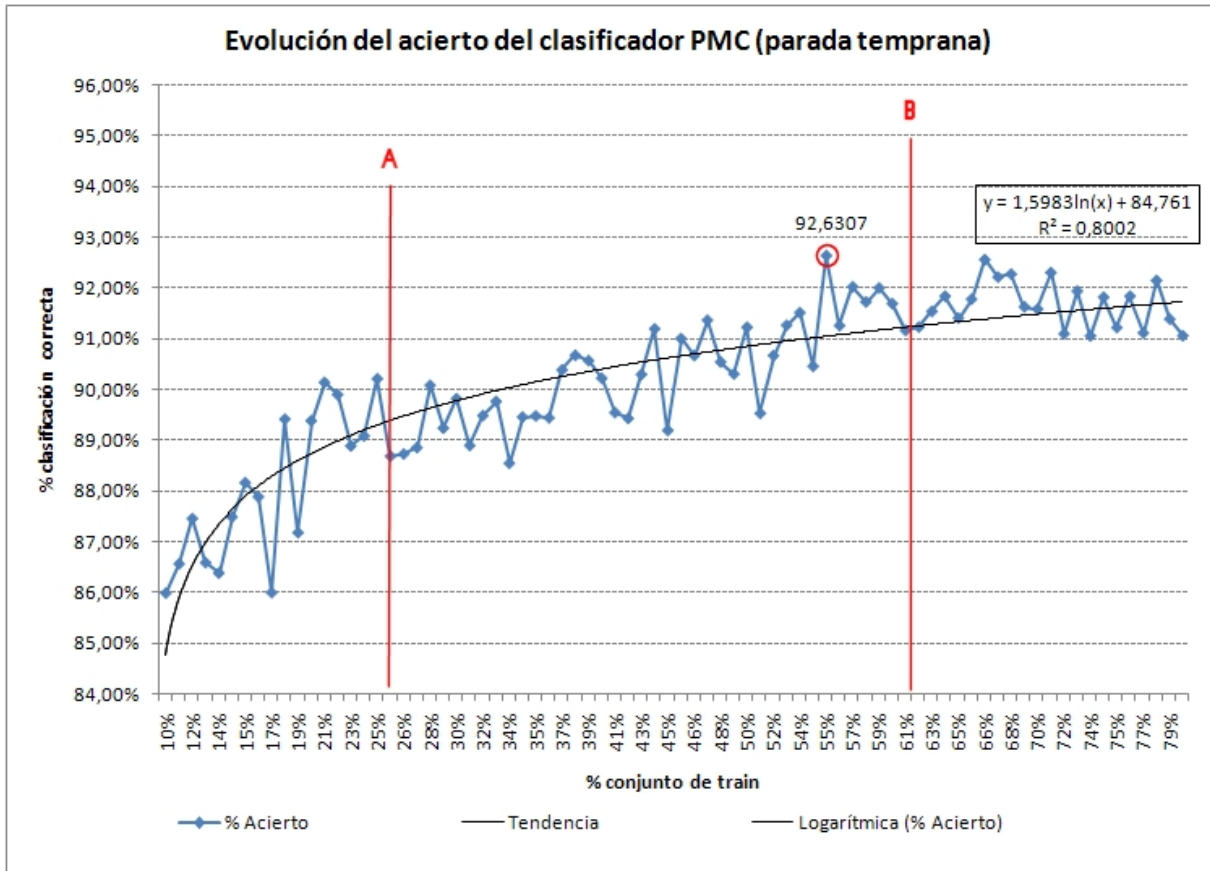


Figura 5.6: Tarea 1.2. Evolución del porcentaje medio de acierto del clasificador en función del número de datos para entrenar para #3 (20 repeticiones)

La ventaja de realizar este análisis es que facilita conocer la cantidad de datos con que los resultados son aceptables, ya que la mejora conseguida a partir de este punto será mínima. Esto permite por un lado reducir el coste de conseguir datos adicionales para aumentar el tamaño del conjunto de entrenamiento y por otro acelerar la convergencia del aprendizaje. En este sentido es necesario razonar si compensa el coste asociado a generar más ensayos para ganar un porcentaje reducido de acierto $\approx 1\%$. Sin embargo, el problema asociado a esta técnica es la subjetividad en determinar que representa un nivel de mejora insignificante. Por ello, es necesario tener en cuenta los otros criterios comentados tales como coste de extracción de la información, tiempo requerido para desarrollar y entrenar el modelo, etc.

A partir de los resultados obtenidos por el PMC en cada criterio de parada, y para probar la evolución del acierto del clasificador según aumenta el número de patrones más allá de los disponibles, se ha recurrido al modelo matemático descrito por Boonyanunta y Zeephongsekul en [Boonyanunta and Zeephongsekul, 2004]. Dicho modelo ha sido

Tabla 5.10: Tarea 1.2. Valores aproximados por el modelo matemático

	T(%)	p en T	k	P0(%)	Max.(%)	p en Max.	Dife. media (%)
#1	94.1285	907	0.000951	72.0128	91.3609	156	5.0621
#2	95.9867	512	0.004153	61.3257	91.9066	150	4.7197
#3	96.4209	431	0.001969	60.2219	92.6307	122	3.2108

explicado en §4.3.2 y se resume en la ecuación:

$$P(p) = T(1 - e^{-kp}) + P(0)e^{-kp} \quad (5.1)$$

donde T es el umbral de eficiencia, k la tasa de mejora en la capacidad de predicción por unidad de aumento de la eficiencia y $P(0)$ es el poder predictivo cuando no se facilitan datos de entrenamiento al clasificador. Se han elegido tres puntos para aproximar el modelo descrito, 50, 100 y 150 patrones. A partir de los valores alcanzados por el clasificador y del método de optimización implementado en Microsoft Excel 2003 (método Solver), se han obtenido los valores mostrados en la Tabla 5.10.

El método Solver muestra el umbral máximo que podría alcanzar el clasificador ($T(\%)$) al aumentar el número de patrones (p). La columna Max. (%) indica el valor máximo alcanzado por el clasificador y p en Max. el número máximo de patrones usados en ese caso. No obstante, al igual que ocurre con la investigación de Crowther y Cox [Crowther and Cox, 2006] el modelo presenta cierto desfase en sus predicciones y no se ajusta correctamente para pocos datos. Las líneas de tendencia sobrepuestas en la Figura 5.5 y en la Figura 5.6 muestran que el valor umbral no se ha alcanzado para ningún criterio de parada. Además la elección de los puntos de referencia usados para ajustar el modelo se presenta fundamental a la hora de usar el método de optimización. Si bien la mejora prevista al aumentar el número de ensayos es superior al 2.5% en todos los casos, hay que tener en cuenta el coste asociado a cada simulación de un solo caso balístico.

Las líneas de tendencia que aparecen son de tipo logarítmico, ya que se trata de una línea curva que se ajusta perfectamente a los datos, siendo de gran utilidad cuando el índice de cambios de los datos aumenta o disminuye rápidamente y después se estabiliza. Los puntos A y B se han aproximado mediante el cálculo de la media de variaciones y la desviación típica que se ha obtenido en las simulaciones.

Finalmente y una vez analizados los resultados se puede definir la cantidad de ensayos y las distribuciones utilizadas en las siguientes fases de la investigación:

- #1. Reparto de 71 %-29% (train-test): Acierto promedio de 91.8602%
- #2. Reparto de 68 %-32% (train-test): Acierto promedio de 92.4101%

Tabla 5.11: Tarea 1.3. Detalle de los experimentos incluidos

Método			Acrónimo	Descripción
Validación Equilibrada	Cruzada	<i>Hold-Out</i>	HOCVE	Entrenamiento convencional.
Validación Equilibrada	Cruzada	<i>K-Fold</i>	$KFCVE_5$, $KFCVE_{10}$	KFCVE con 5 y 10 conjuntos.
Validación Equilibrada	Cruzada	<i>Leave-One-Out</i>	LOOCV	KFCVE con 220 conjuntos.
Remuestreo Bootstrap			BSR_{20} , BSR_{50} , BSR_{100}	Bootstrapping con 20, 50 y 100 ensayos con remuestreo.
Variante Iterativa de Validación Cruzada	Equilibrada	de	$VIECV_{20}$, $VIECV_{50}$, $VIECV_{100}$	Generación aleatoria de 20, 50 y 100 ensayos.
Ruido Blanco Gaussiano Aditivo			$AWGN_{50}$, $AWGN_{100}$, $AWGN_{150}$	Creación de 50, 100 y 150 ensayos con ruido y distribución Gaussiana.
KFCVE & AWGN			$K_{10}A_{100}$	Combinación de los métodos $KFCVE_{10}$ y $AWGN_{100}$.

- #3. Reparto de 55 %-20 %-25 % (train-test-validación): Acierto promedio de 92.6307 %

5.2.3.3. Tarea 1.3: Técnicas para trabajar con datos limitados

Debido a la gran cantidad de simulaciones realizadas esta tarea se divide en tres subtareas. Cada una está destinada al estudio de las diferentes técnicas estadísticas explicadas en §4.3.3 junto con un criterio de parada distinto: *epochs*, MSE y parada temprana. El listado de técnicas y métodos aplicados sobre los patrones de entrada se muestran en la Tabla 5.11. Cada una de las simulaciones será repetida 20 veces para reducir los posibles errores de tipo I y II y conseguir replicabilidad o estabilidad en los resultados. La alternativa combinada $K_{10}A_{100}$ se ha incluido debido al comportamiento de ambas técnicas por separado y a sus ventajas desde el punto de vista de generación de los ensayos y rapidez de cálculo. Sus características permiten que puedan combinarse sin tener que destinar elevados recursos a la generación de los ensayos con ruido y su posterior distribución entre los conjuntos de datos. En primer lugar la técnica AWGN genera un número de nuevos ensayos con ruido Gaussiano a partir de los originales y posteriormente KFCVE trabaja a partir de este nuevo conjunto de ensayos balísticos.

La estrategia de distribución estudiada en la Tarea 1.2 no puede aplicarse en aquellas técnicas incluidas en este apartado, que debido a sus características, empleen criterios propios para la distribución de patrones en los conjuntos de *train*, *test* y *CV*. En estos casos se comparará la eficacia de estas nuevas propuestas frente a los resultados obtenidos

anteriormente.

5.2.3.3.1. Tarea 1.3.1. Parada del entrenamiento por *epochs*

Los resultados de las simulaciones realizadas para este criterio de parada aparecen en la Figura 5.7. Para cada uno de los métodos originales se han propuesto una serie de variaciones listadas en la Tabla 5.11, limitadas por el coste computacional asociada a cada una de ellas y por los resultados obtenidos durante las pruebas.

La parte superior de la figura muestra la desviación obtenida, cuantificada en porcentaje de acierto, entre el método clásico de entrenamiento HOCVE y el resto de métodos analizados. Aunque el método HOCVE debido a su velocidad suele ser el más empleado para entrenar RNA, éste desperdicia una gran cantidad de información ya que solo una parte de los datos se usa para entrenar la red. Cuando la cantidad de datos disponible es reducida, como sucede en este dominio, es necesario estudiar otras alternativas que maximicen y optimicen el uso de los datos. La parte inferior de la gráfica muestra una comparación, por número de *epochs* y porcentaje de acierto, entre las mejores alternativas para cada uno de los métodos analizados.

Como se puede observar, la propuesta $K_{10}A_{100}$ es la que presenta los mejores resultados, con un incremento en media de 4.5731 % después de todas las simulaciones. La mejora máxima respecto a HOCVE se obtiene con 27000 *epochs*, 5.9145 %, coincidiendo con el punto de máximo acierto del clasificador en todas las simulaciones, 96.0750 %.

Las siguientes propuestas que tienen un efecto beneficioso son, por porcentaje de acierto, LOOCV, $KFCVE_{10}$ y $AWGN_{100}$, con una mejora media de 4.3809 %, 3.2607 % y 2.7731 % respectivamente. El primero de ellos, LOOCV, tiene como desventaja su elevado coste computacional, tanto en entrenamiento como en generación de los conjuntos de ensayos necesarios. Si a esto se le une que su utilidad para seleccionar modelos ha sido cuestionada por diversos autores [Breiman, 1996], no es conveniente su aplicación como técnica de muestreo dentro de esta metodología.

Por último los mejores candidatos de las otras técnicas estudiadas, BSR_{100} y $VIECV_{100}$, consiguen también mejorar el comportamiento de la técnica clásica HOCVE. Para el primero el máximo acierto obtenido ha sido de 94.0875 con 3000 *epochs*, con un porcentaje de mejora máximo de 3.0231 % y media de 2.0046 %. Para el segundo el acierto máximo alcanzado ha sido de 93.1667 % con 9000 *epochs*, con un porcentaje de mejora máxima de 2.6387 % y media de 2.0607 %.

La última técnica analizada para mejorar el rendimiento del PMC y evitar el posible sobreaprendizaje ha sido la inyección de ruido a los patrones según se introducen en el modelo. Los efectos derivados de usar esta técnica con ruido Gaussiano y ruido Uniforme



Figura 5.7: Tarea 1.3.1 (criterio #1). (arriba) Desviación de los diferentes métodos sobre HOCVE (en porcentaje de acierto) y (abajo) resultados obtenidos para los mejores candidatos de cada método (en porcentaje de acierto) (20 repeticiones)

y un entrenamiento por *epochs* se muestran en la Figura 5.8. La gráfica muestra el porcentaje medio de desviación para todas las simulaciones dentro del rango de *epochs* [1000-150000]. Su aplicación de forma combinada con las diferentes técnicas estadísticas

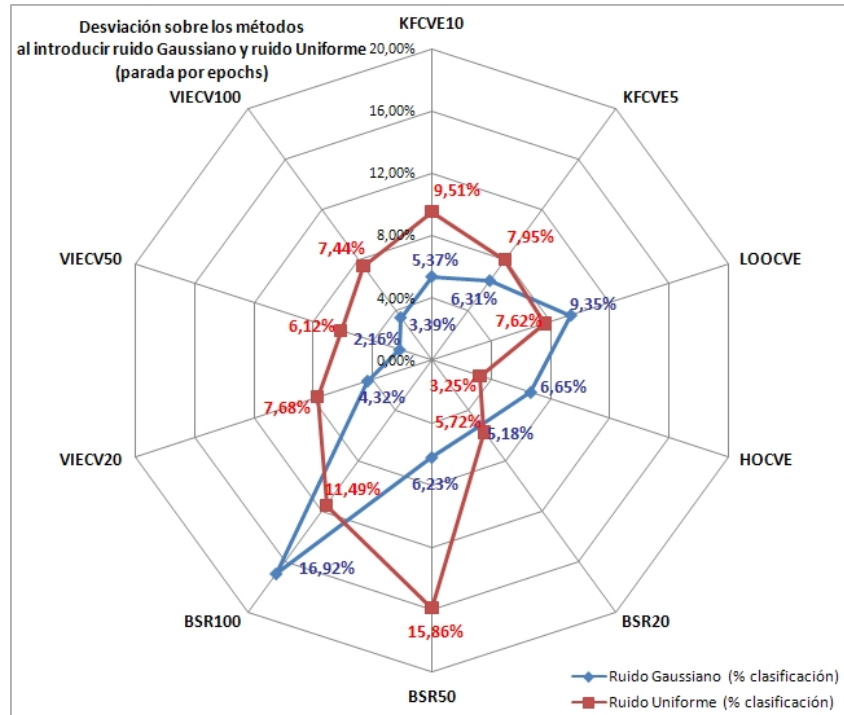


Figura 5.8: Tarea 1.3.1 (criterio #1). Media de la desviación obtenida (en porcentaje de acierto) usando ruido Gaussiano y ruido Uniforme sobre HOCVE (20 repeticiones)

no ha mejorado, ni en media ni en ninguna simulación individual, la calidad de la red. Es interesante el efecto pernicioso que tiene el ruido sobre la técnica BSR, siendo con diferencia la más sensible respecto a los datos de entrada. Además una pauta común para las técnicas BSR, VIECV y KFCVE es que el aumento del número de conjuntos aleatorios o remuestreos no mejora el rendimiento del PMC. Si se comparan los resultados para los dos tipos de ruido usados, en general el ruido Gaussiano presenta un mejor comportamiento dentro de este dominio, pero presentando unas tasas de desviación superior al 2% en el mejor de los casos.

En resumen, la técnica estadística con mejor resultado para este dominio, teniendo en cuenta su precisión y velocidad de simulación ha sido la combinación de KFCVE y AWGN. En este caso el mejor resultado ha sido el siguiente: 96.0750% con la técnica $K_{10}A_{100}$ y 27000 *epochs*.

5.2.3.3.2. Tarea 1.3.2. Parada del entrenamiento por MSE

Los resultados de las simulaciones realizadas para este criterio de parada aparecen en la Figura 5.9. Para cada uno de los métodos originales se han propuesto una serie de variaciones listadas en la Tabla 5.11, limitadas por el coste computacional asociada a cada una de ellas y por los resultados obtenidos durante las pruebas.

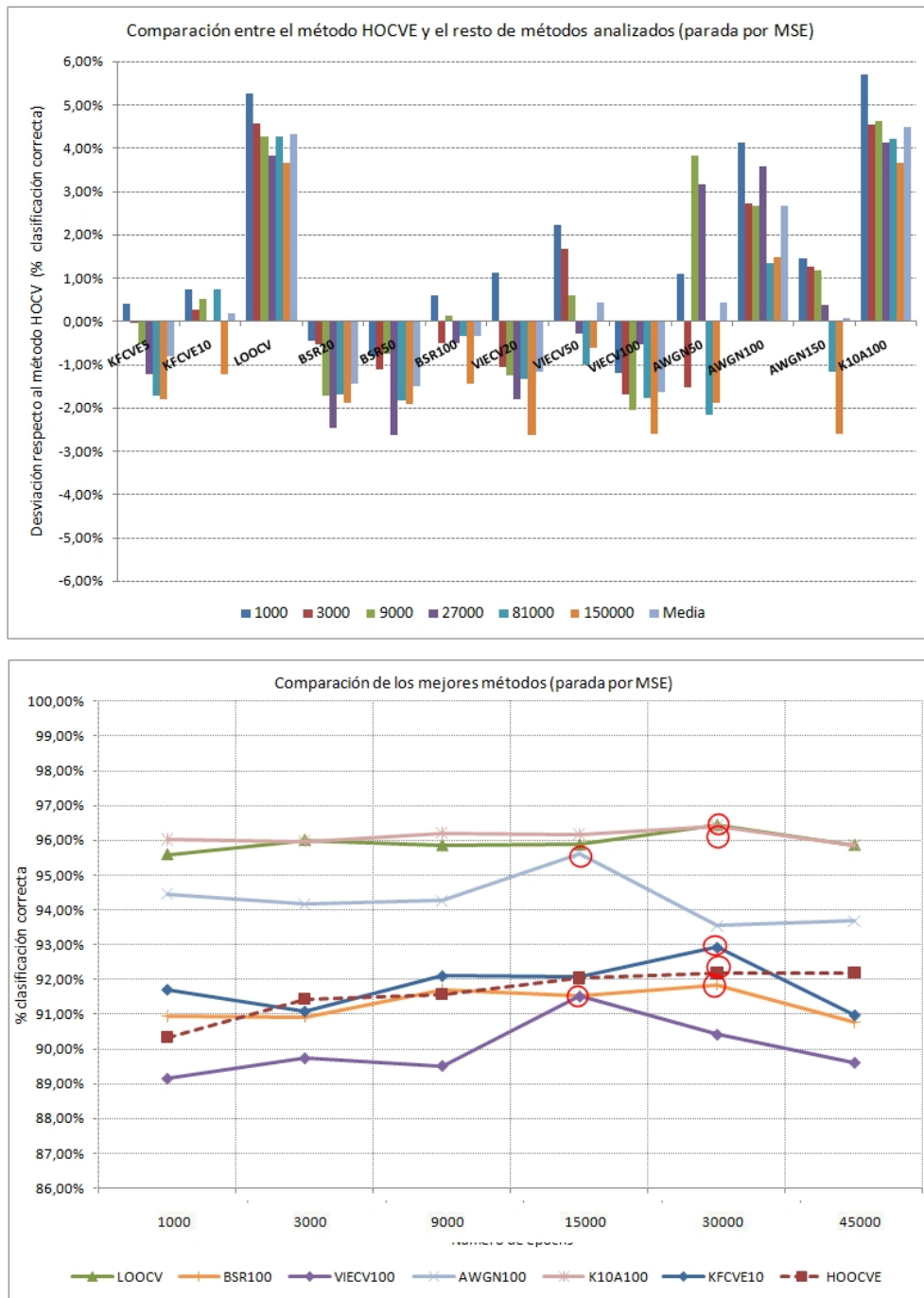


Figura 5.9: Tarea 1.3.2 (criterio #2). (arriba) Desviación de los diferentes métodos sobre HOCVE (en porcentaje de acierto) y (abajo) resultados obtenidos para los mejores candidatos de cada método (en porcentaje de acierto) (20 repeticiones)

La parte superior de la figura muestra la desviación obtenida, cuantificada en

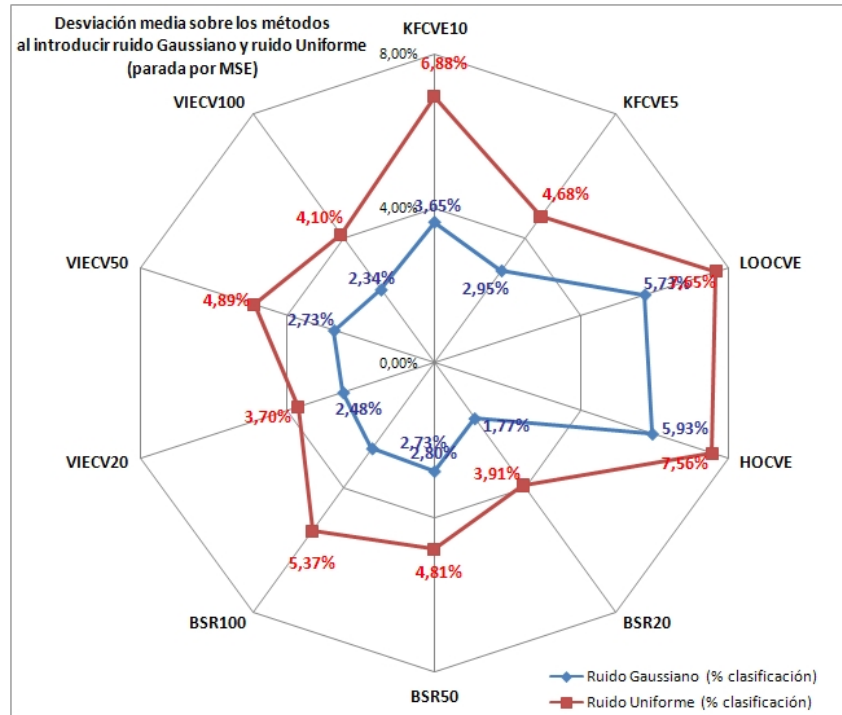


Figura 5.10: Tarea 1.3.2 (criterio #2). Media de la desviación obtenida (en porcentaje de acierto) usando ruido Gaussiano y ruido Uniforme sobre HOCVE (20 repeticiones)

porcentaje de acierto, entre el método clásico de entrenamiento HOCVE y el resto de métodos analizados para el criterio de parada combinado de *epochs* y valor mínimo MSE. La parte inferior muestra una comparación, por número de *epochs* y porcentaje de acierto, entre las mejores alternativas para cada uno de los métodos analizados.

De nuevo las mejores alternativas son $K_{10}A_{100}$, LOOCV y $AWGN_{100}$, con una mejora media respecto a HOCVE de 4.4309 %, 4.3131 % y 2.6605 % respectivamente. El mejor resultado para el primero ha sido de 96.4129 % con 30000 *epochs*, 96.4445 % con 30000 *epochs* para el segundo y para el tercero de 95.6246 % y 15000 *epochs*. El resto de técnicas se sitúan un escalón por debajo en cuanto a la calidad de sus predicciones, siendo en algunos casos, e.g. $VIECV_{100}$, peor que la técnica de referencia HOCVE.

Al igual que con el primer criterio de parada, la inyección de ruido a los patrones mientras se introducen en la red no ha mejorado, ni en media ni en ninguna simulación individual, la calidad del PMC. La Figura 5.10 muestra el porcentaje medio de desviación para todas las simulaciones realizadas en cada uno de los métodos. El comportamiento de todos los métodos analizados es similar para ambos tipos de ruido, siendo preferible el ruido Gaussiano al Uniforme al presentar una menor distorsión. No obstante la pérdida de acierto en porcentaje es 3.3113 % para el primero y de 5.3656 % para el segundo. Las alternativas que presentan una sensibilidad mayor a ambos tipos de ruido son LOOCV y HOCVE.

Por tanto, de nuevo la técnica estadística con mejor resultado para este dominio, teniendo en cuenta su precisión y velocidad de simulación es la combinación de KFVC y AWGN. El mejor resultado ha sido el siguiente: 96.4129% con la técnica $K_{10}A_{100}$ y el criterio combinado de 30000 *epochs* o $MSE < 0.01$.

5.2.3.3.3. Tarea 1.3.3. Parada del entrenamiento por error de validación (parada temprana)

Los resultados de las simulaciones realizadas para este criterio de parada aparecen en la Figura 5.11. Para cada uno de los métodos originales se han propuesto una serie de variaciones listadas en la Tabla 5.11, limitadas por el coste computacional asociado a cada una de ellas y por los resultados obtenidos durante las pruebas.

La parte superior de la figura muestra la desviación obtenida, cuantificada en porcentaje de acierto, entre el método clásico de entrenamiento HOCVE y el resto de métodos analizados para el criterio de parada temprana. La parte inferior muestra una comparación, por número de *epochs* y porcentaje de acierto, entre las mejores alternativas para cada uno los métodos analizados.

De nuevo las mejores propuestas son $K_{10}A_{100}$, LOOCV y $AWGN_{100}$, con una mejora media respecto a HOCVE de 4.5689%, 4.1413% y 1.6485% respectivamente. El mejor resultado para el primero ha sido de 96.7625% y un máximo de 20000 *epochs*, 96.6818% con 20000 *epochs* para el segundo y para el tercero de 94.7962% con 10000 *epochs*. El resto de técnicas se sitúan un escalón por debajo en cuanto a la calidad de sus predicciones pero mejorando en la mayoría de ocasiones a la técnica simple HOCVE.

Al igual que los anteriores criterios de parada, la inyección de ruido a los patrones mientras se introducen en la red no ha mejorado, ni en media ni en ninguna simulación individual, la calidad del PMC. La Figura 5.12 muestra el porcentaje medio de desviación para todas las simulaciones realizadas respecto a los métodos originales sin inyección de ruido. El comportamiento de todas las variantes analizadas es similar para ambos tipos de ruido, siendo preferible el ruido Gaussiano al Uniforme al presentar una menor distorsión. Las alternativas que presentan una sensibilidad mayor con ambos tipo de ruido vuelven a ser LOOCV y HOCVE junto a $KFCVE_{10}$.

Por tanto, de nuevo la técnica estadística con mejor resultado para este dominio, teniendo en cuenta su precisión y velocidad de simulación es la combinación de KFCV y AWGN. El mejor resultado de todas las alternativa analizadas y que por tanto se utilizará como base para el resto de simulaciones ha sido el siguiente: 96.7625% con la técnica $K_{10}A_{100}$ y el criterio de parada temprana y un máximo de 20000. Para

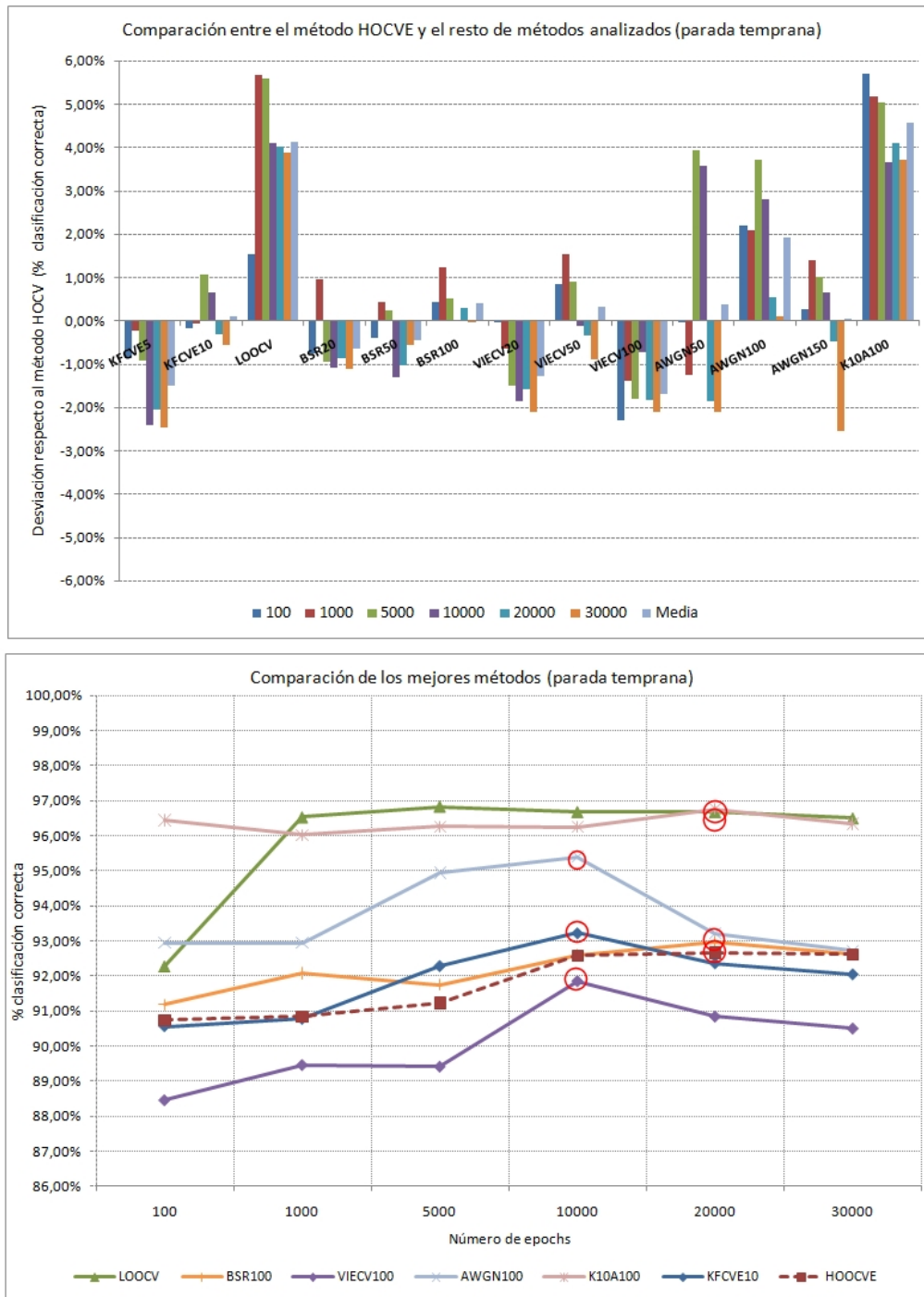


Figura 5.11: Tarea 1.3.3 (criterio #3). (arriba) Desviación de los diferentes métodos sobre HOCVE (en porcentaje de acierto) y (abajo) resultados obtenidos para los mejores candidatos de cada método (en porcentaje de acierto) (20 repeticiones)

ratificar estas conclusiones, en el siguiente apartado se analizarán desde el punto de vista estadístico los resultados asociados a cada técnica bajo esta configuración de parada

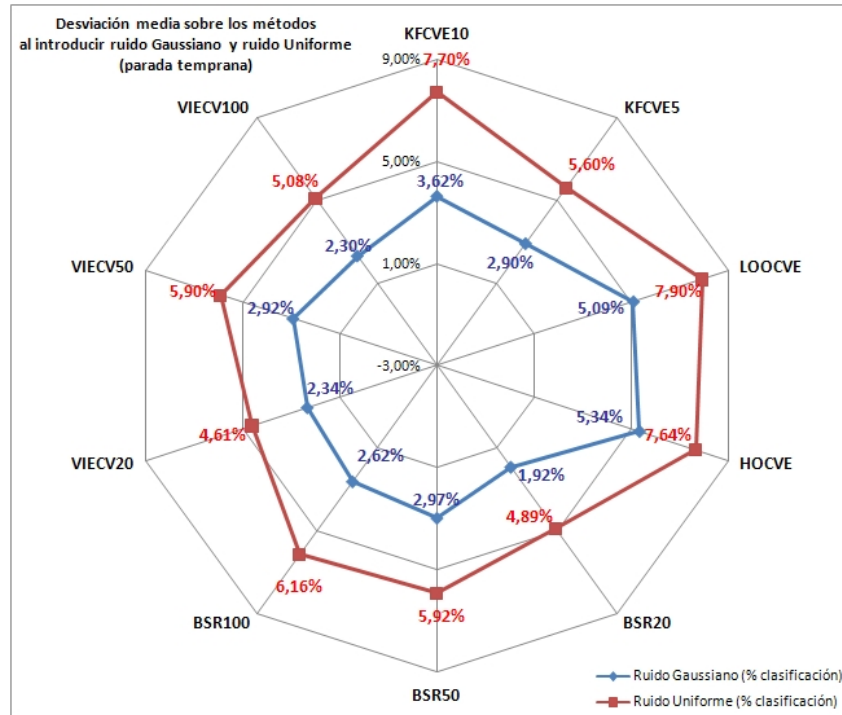


Figura 5.12: Tarea 1.3.3 (criterio #3). Media de la desviación obtenida (en porcentaje de acierto) usando ruido Gaussiano y ruido Uniforme sobre HOCVE (20 repeticiones)

temprana.

5.2.3.3.4. Medidas de calidad de los resultados

Esta tarea pretende corroborar las conclusiones obtenidas anteriormente, realizando un análisis estadístico enfocado a medir la calidad de cada técnica en función de los estimadores y contrastes descritos en §4.6.2. En la Figura 5.13 se muestran los diagramas de dispersión y de caja correspondientes a las mejores alternativas localizadas para cada técnica. En este caso, no hay puntos externos, pero hay tres puntos externos lejanos asociados a la técnica LOOCV, representados como pequeños cuadrados, y su media, representada por una cruz aparece claramente desviada. La presencia de puntos externos lejanos indica valores atípicos y una distribución altamente sesgada hacia la izquierda. Además las anchuras de las muescas de las medianas no se solapan en casi ningún caso, por lo que se podría considerar que hay diferencia estadística significativa entre ellas para un 95 % de nivel de confianza.

Al realizar un análisis de los estimadores estadísticos, se observa que existe una diferencia superior de tres a uno entre la desviación típica más pequeña y la más grande. Esto puede causar problemas puesto que el análisis de la varianza asume que las desviaciones típicas en todos los niveles son iguales. Por ello se ha realizado un contraste

Tabla 5.12: Tarea 1.3.3. Análisis aplicando la Tarea 4.2: Contraste de la varianza

Contraste	Valor	p-valor
C de Cochran	0.4354	0.00026
Bartlett	1.3974	0.00016
Test de Levene	1.6471	0.1423

de la varianza, mostrado en la Tabla 5.12. Dado que dos de los p-valores son inferiores a 0.05, hay diferencia estadística significativa entre las desviaciones típicas para un nivel de confianza del 95.0%. Esto infringe una de las asunciones importantes que subyacen en el análisis de la varianza y podría invalidar el uso del método ANOVA. Además la asimetría y la curtosis estandarizadas esta fuera del rango de -2 a +2 para la técnica LOOCV, lo que indica una no normalidad significativa en los datos. Esto viola la asunción de que los datos proceden de distribuciones normales, lo que provoca que no se pueda aplicar el método ANOVA para comparar los diferentes resultados. En su lugar se ha utilizado el tests no paramétrico de Kruskal-Wallis para comparar las medianas en lugar de las medias.

El test de Kruskal-Wallis mostrado en la Tabla 5.13 es un método no paramétrico para probar si un grupo de datos proviene de la misma población. Intuitivamente es idéntico al ANOVA con los datos reemplazados por categorías. En este caso, se comprueba la hipótesis nula de igualdad de las medianas para el porcentaje de acierto en cada una de las siete alternativas. Los datos de todos los niveles primero se combinan y se ordenan de menor a mayor y luego se calcula el rango medio para los datos en cada nivel. Puesto que el p-valor es inferior a 0.05, hay una gran evidencia estadística en contra de que los resultados obtenidos por todas las técnicas sean similares. Para determinar cuáles son las medianas significativamente diferentes entre si, se puede observar el gráfico de caja de la Figura 5.13, donde la anchura de las muescas indica el intervalo de confianza aproximado del 95.0%.

Tabla 5.13: Tarea 1.3.3. Análisis aplicando la Tarea 4.2: Test de Krustal Wallis

Técnica	Rango Medio %
$K_{10}A_{100}$	89.7333
LOOCV	85.7102
$AWGN_{100}$	67.5333
BSR_{100}	44.6667
$KFCVE_{10}$	37.4333
HOCVE	34.6002
$VIECV_{100}$	11.3333
Estadístico = 81.1271	p-valor = 0.0

Como se puede observar las técnicas LOOCV y $K_{10}A_{100}$ presentan un comportamiento homogéneo y las distribuciones asociadas a los resultados son significativamente distintas

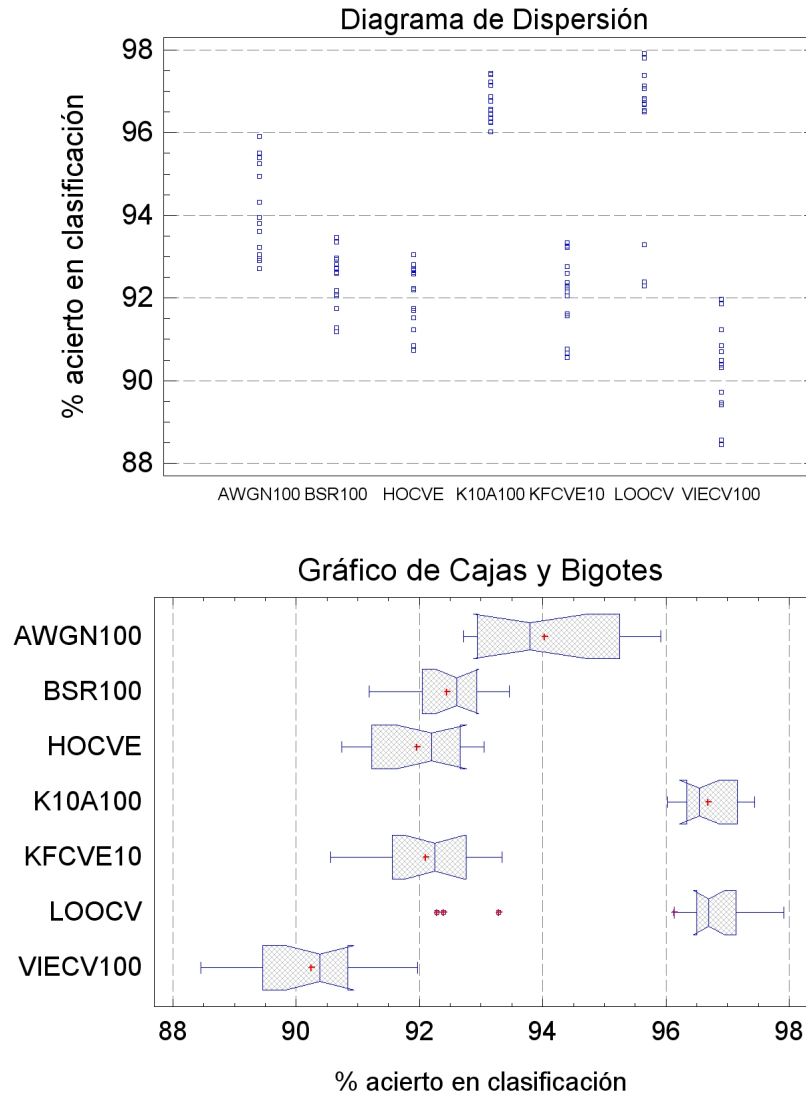


Figura 5.13: Tarea 1.3.3. Diagrama de caja y dispersión

a todas las demás. No obstante la media del porcentaje de acierto de $K_{10}A_{100}$ es superior a LOOCV en 0.4276 % y su rango medio al finalizar el test de Kruskal-Wallis es superior en 4.0231 %. Además la distribución de los resultados de LOOCV no es simétrica y presenta un sesgo hacia la izquierda al aparecer valores atípicos inferiores. Si a esto se le une el hecho de su elevado tiempo de computación en comparación con la alternativa $K_{10}A_{100}$, esta última será la elegida como técnica para maximizar el uso y distribuir los patrones balísticos disponibles.

5.2.3.4. Sumario

Una vez terminada esta primera fase se pueden sacar las siguientes conclusiones que serán aplicadas sobre las siguientes fases y tareas:

- En todas las tareas estudiadas el criterio de parada que presenta un mejor comportamiento es #3 (parada temprana) y un máximo de 20000 *epochs*. Además, los mejores resultados se obtienen bajo este criterio y su utilización permite reducir de forma considerable el tiempo de entrenamiento y prevenir el sobreaprendizaje. Por tanto los otros dos criterios, #1 (*epochs*) y #2 (MSE), no serán tenidos en cuenta en las siguientes fases.
- La normalización *z-score* es la mejor opción para el preprocesado de los patrones.
- Una correcta distribución de los patrones en los conjuntos de *train*, *test* y validación, mejora los resultados que obtienen las configuraciones clásicas (80 %-20 % y 33 %-33 %-33 %) con el entrenamiento HOCVE. Para el criterio de parada seleccionado, #3, el reparto de patrones será de: 55 %-20 %-25 % (*train-test-validación*).
- Los mejores resultados asociados a esta fase se obtienen en la Tarea 1.3 gracias a aplicación de técnicas para maximizar la información. De todas las alternativas analizadas, la que mejor comportamiento presenta es la combinación de la técnica KFCVE junto con la introducción de ruido siguiendo la técnica AWGN. A su vez para esta nueva propuesta se simularon diferentes configuraciones, siendo la mejor candidata la versión $K_{10}A_{100}$.
- Para paliar la influencia de la partición aleatoria de la técnica KFCVE, se repetirá cada simulación de la alternativa $K_{10}A_{100}$ varias veces [Dietterich, 1998]. En este caso se recurre a la alternativa $20 * 10cv$, también denominada *repeated K-fold cross validation*, que indica que se ejecutará 20 veces cada proceso.
- Los valores máximos de acierto obtenidos con cada criterio de parada y la técnica $K_{10}A_{100}$ han sido: #1 con 96.0750 %, #2 con 96.4129 % y #3 con 96.7625 %.

5.2.4. Fase 2: Arquitectura de la red

Esta fase, desglosada en cuatro tareas secuenciales, está orientada a optimizar aquellos aspectos relativos a la arquitectura de la red neuronal. La configuración usada en esta fase parte de la inicial, C_0 , y de la obtenida en la primera fase, C_1 . Por tanto, esta nueva configuración se denomina configuración de prueba dos, C_2 , y vuelve a estar dividida en tres bloques: patrones (P_1), arquitectura (A_1) y algoritmo (R_0). Sus características más interesantes son las siguientes:

- Técnica $K_{10}A_{100}$ con 20 repeticiones ($20 * 10cv$).

- Distribución de patrones en 10 conjuntos de similar tamaño (320 patrones en 10 conjuntos, 9 se usan para entrenar y 1 para testear) y normalización *z-score*.
- Topología 7-4-2 y funciones de activación f_{iden} - f_{tanh} - f_{tanh} para las capas *entrada – oculta – salida*.
- Criterio de parada temprana y un máximo de 20000 *epochs*.

5.2.4.1. Tarea 2.1. Inicialización los pesos de la red

Como ha quedado expuesto en §4.4.1, resulta necesario seleccionar valores no nulos en los pesos de las conexiones para que el proceso de aprendizaje pueda llevarse a cabo. Además la elección de unos valores pequeños en los pesos iniciales de la red es esencial para reducir la posibilidad de saturación en las neuronas. En esta investigación se ha elegido inicialmente la heurística expuesta por Haykin [Haykin, 1994] al ser la más extendida. Para ello la inicialización neurona a neurona ha estado dentro del rango $(-\frac{r}{N_j}, \frac{r}{N_j})$, donde r es un número real dependiente de la función de activación y N_j es el número de conexiones que alimentan la neurona en cuestión. Esta regla de inicialización se ha aplicado en todas las simulaciones realizadas con anterioridad, cambiando en función de la topología de la red diseñada los valores iniciales.

A continuación se comparará el rendimiento del PMC bajo otras condiciones iniciales, regidas por la propuesta de Principe et al. [Principe et al., 1999]. En este caso los valores se calculan mediante la regla $(\frac{-2,4}{I}, \frac{2,4}{I})$, donde I es el *fan-in* de cada neurona.

En la Tabla 5.14 se muestran los resultados obtenidos por el PMC para los dos criterios de condiciones iniciales. La propuesta de Principe consigue un valor máximo similar al criterio de Haykin, sólo un 0.8869% peor, pero su rendimiento al analizar la media obtenida para todas las repeticiones es más voluble, un 1.0237% por debajo en acierto. Además si se tiene en cuenta la distribución asociada a los resultados de ambos criterios, se puede observar en la Figura 5.14 que las dos son asimétricas respecto a la mediana. Por último, el criterio de Principe presenta valores atípicos para el rango inferior de acierto, lo que indica un comportamiento más inestable.

Los resultados analizados desde un punto de vista estadístico se muestran en la Tabla 5.14. De particular interés son los estadísticos asimetría estandarizada y curtosis estandarizada ya que pueden usarse para determinar si las muestras proceden de distribuciones normales. Los valores de estos estadísticos fuera del rango de -2 a +2 indican salidas significativas de normalidad que tenderían a invalidar las pruebas que comparan las desviaciones normales. En este caso, la propuesta de Principe tiene un valor de asimetría estandarizada fuera del rango normal y ambos valores de curtosis estandarizada están dentro del rango esperado. Con respecto a las diferencias obtenidas entre ambos métodos, la asimetría está fuera de rango mientras que la curtosis presenta valores esperados.

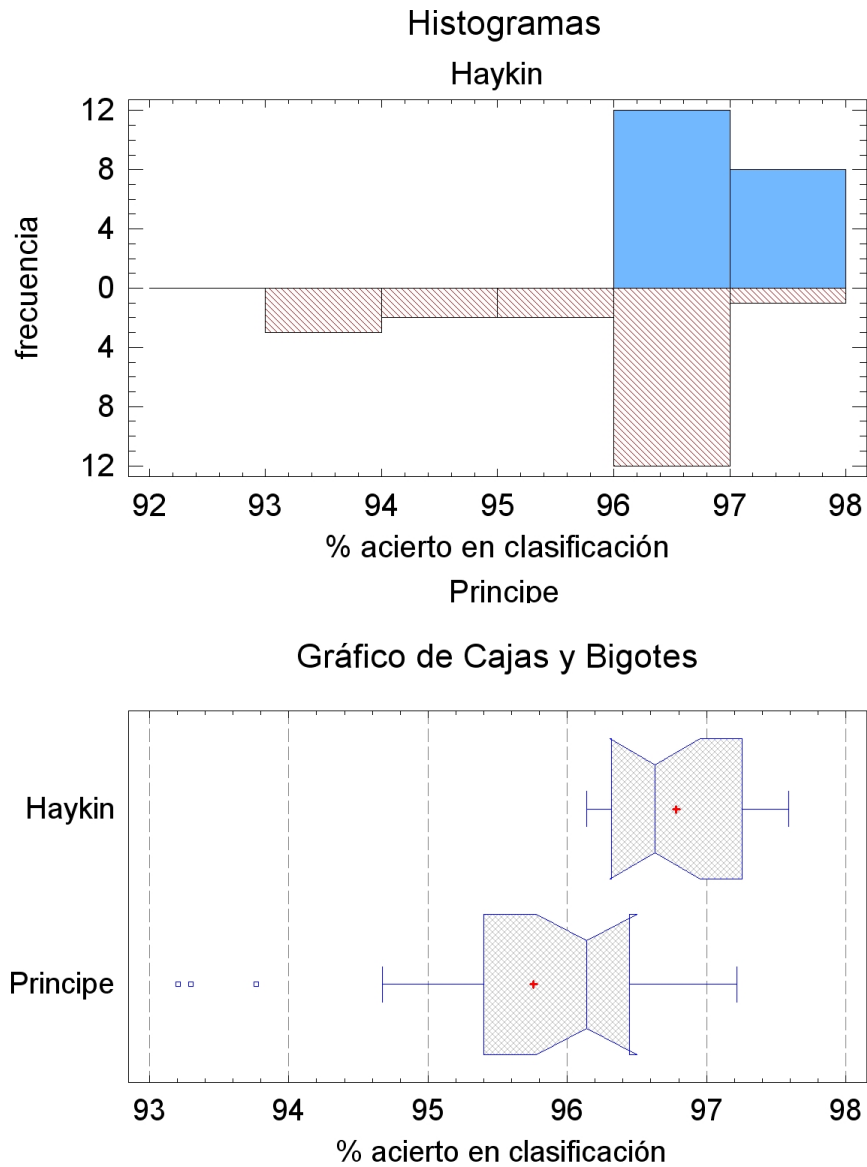


Figura 5.14: Tarea 2.1. Diagramas de histogramas y de caja

5.2.4.1.1. Medidas de calidad de los resultados

En condiciones generales, se puede afirmar que el objeto de la comparación de dos algoritmos es decidir si el valor medio de su medida de error sobre la población completa coincide, o es distinto. En esta investigación se está empleando la técnica $K_{10}A_{100}$ repetida 20 veces, por lo que se dispone de 10 estimaciones del error de cada algoritmo, como resultado de evaluarlo sobre cada uno de los conjuntos T_i con 20 repeticiones. Ese conjunto de valores puede considerarse, a su vez, como una muestra de k realizaciones independientes de una variable aleatoria o error muestral, asociada al algoritmo.

Tabla 5.14: Tarea 2.1 Análisis estadístico (20 repeticiones)

Estadístico	(medido % acierto)		(medido en % error)
	Haykin	Principe	Diferencias Haykin-Principe
Media	96.7824	95.7586	-1.0237
Varianza	0.2628	1.3461	1.0332
Desviación típica	0.5126	1.1602	1.0164
Mínimo	95.5167	92.5874	-3.4335
Máximo	96.9706	96.5991	0.2242
Rango	1.45388	4.0117	3.6577
Asimetría tipificada	0.5642	-2.3799	-2.1051
Curtosis tipificada	-1.4456	0.5615	0.8376
<i>Epochs</i> promedio	12607	8917	

Bajo este punto de vista, si se desea contrastar que dos algoritmos de aprendizaje son distintos, es válido definir como hipótesis nula del contraste la afirmación “Las dos muestras de errores proceden de poblaciones con medias iguales”. Si los datos están apareados, lo que ocurre si los dos algoritmos se han probado sobre las mismas particiones, las dos muestras de errores pueden restarse elemento a elemento, con lo que la hipótesis nula equivalente sería “La diferencia entre los errores muestrales de ambos algoritmos tiene media cero”. Si los errores muestrales de los dos algoritmos siguiesen una distribución normal, el test más potente para contrastar dicha hipótesis, bajo condiciones muy generales, es el test t. En el caso de no cumplirse la premisa de normalidad para las diferencias, se puede recurrir a alternativas no paramétricas como el test de los signos.

A la vista de los resultados obtenidos en la Tabla 5.14 existen evidencias de un posible alejamiento de la normalidad. Por ello además de incluir el contraste t-test pareado se muestra el contraste test de los signos y test de los rangos de los signos, también conocidos como test de Wilcoxon. A continuación se describen los tres tests de contraste de homogeneidad realizados:

1. Contraste t:

- Hipótesis nula H_0 : media = 0.0
- Hipótesis alternativa H_1 : no igual
- Estadístico t = -4.5042
- P-valor = 0.00024
- Se rechaza la hipótesis nula para alpha = 0.05.

2. Contraste de los signos:

- Hipótesis nula H_0 : mediana = 0.0
- Hipótesis alternativa H_1 : no igual

- Número de valores inferiores a la mediana de H_0 : 17
- Número de valores superiores a la mediana de H_0 : 3
- Estadístico para grandes muestras = 2.9068 (aplicada la corrección por continuidad¹). P-valor = 0.0036
- Se rechaza la hipótesis nula para $\alpha = 0.05$.

3. Contraste de rangos con signo:

- Hipótesis nula H_0 : mediana = 0.0
- Hipótesis alternativa H_1 : no igual
- Rango medio de los valores inferiores a la mediana: 11.9412
- Rango medio de los valores superiores a la mediana: 2.3332
- Estadístico para grandes muestras = 3.6399 (aplicada la corrección por continuidad). P-valor = 0.0002
- Se rechaza la hipótesis nula para $\alpha = 0.05$.

El primer contraste es un t-test cuya hipótesis nula es que la diferencia media de Haykin-Principe sea igual a cero frente a la hipótesis alternativa de que no lo sea. Puesto que el p-valor para este test es inferior a 0.05, se puede rechazar la hipótesis nula para un nivel de confianza del 95.0%. Los siguientes tests se han incluido como complemento debido a la aparente no normalidad de la distribución asociada a las diferencias. El segundo contraste es un test de signos cuya hipótesis nula es que la mediana de Haykin-Principe sea igual a cero frente a la hipótesis alternativa de que no lo sea. Se basa en el recuento del número de valores por encima y debajo de la supuesta mediana. Puesto que el p-valor para este test es inferior a 0.05, se puede rechazar la hipótesis nula para un nivel de confianza del 95.0%. El tercer y último contraste es un test de signo-rango cuya hipótesis nula es que la mediana de Haykin-Principe sea igual a cero frente a la hipótesis alternativa de que cuya mediana no lo sea. Se basa en la comparación de rangos medios de valores por encima y por debajo de la mediana hipotética. Puesto que el p-valor para este test es inferior a 0.05, se puede rechazar la hipótesis nula para un nivel de confianza del 95.0%. Los dos últimos contrastes son menos sensibles a la presencia de valores atípicos, pero algo menos potentes que el t-test si todos los datos proceden de una única distribución.

En resumen existen evidencias estadísticas que determinan las diferencias significativas entre los resultados de cada criterio de inicialización. Por tanto y la vista de los resultados y del estudio estadístico, la propuesta de Haykin se seguirá utilizando en las siguientes fases y tareas de la metodología.

¹También conocida como corrección de Yates se aplica al aproximar una variable discreta a una distribución continua.

5.2.4.2. Tarea 2.2: Elección de la función de activación

Como se ha comentado en §4.4.2, casi cualquier función de activación puede ser utilizada dentro de un PMC con algoritmo RP. La única limitación es que dicha función tiene que ser diferenciable. El amplio abanico de posibles funciones disponibles hace difícil poder desarrollar esta tarea dentro de unos plazos razonables. Para reducir el número de simulaciones, pero siguiendo unas pautas establecidas por otros autores, se han tenido en cuenta las siguientes consideraciones [Hassoun, 1995, Priddy and Keller, 2005, Lefebvre et al., 2007]:

- En la mayoría de problemas es conveniente que las funciones utilizadas para todas las neuronas de la capa oculta y de salida sean del mismo tipo.
- Las funciones de activación más utilizadas en clasificación son las del tipo no lineal: f_{tanh} , f_{sig} y $f_{softmax}$.
- Las funciones de activación más utilizadas en regresión son f_{lineal} , f_{bias} , f_{iden} .
- Las funciones de activación de la capa de entrada es del tipo f_{iden} .
- La ventaja de una función sobre otra no ha sido demostrada matemáticamente.

Tabla 5.15: Tarea 2.2. Resultados obtenidos para cada topología (20 repeticiones).

Topología	Epochs realizados	% Acierto medio	AIC	MDL
$f_{iden} - f_{tanh} - f_{tanh}$	12654	96.7988	-26.5886	-25.3691
$f_{iden} - f_{tanh} - f_{sig}$	12103	96.0237	-17.6376	-28.8571
$f_{iden} - f_{tanh} - f_{softmax}$	6401	95.9839	-19.7743	-30.9938
$f_{iden} - f_{tanh} - f_{lineartanh}$	5649	95.9087	28.5000	17.2804
$f_{iden} - f_{tanh} - f_{linearsig}$	3723	92.9500	-5.0692	-16.2887
$f_{iden} - f_{sig} - f_{tanh}$	3937	94.0912	34.3355	23.1160
$f_{iden} - f_{sig} - f_{sig}$	3032	95.6163	-19.2085	-30.4281
$f_{iden} - f_{sig} - f_{softmax}$	6881	95.0273	-13.6740	-24.8936
$f_{iden} - f_{sig} - f_{lineartanh}$	6667	95.1173	32.1927	20.9731
$f_{iden} - f_{sig} - f_{linearsig}$	8678	92.8510	-4.2059	-15.4254
$f_{iden} - f_{softmax} - f_{tanh}$	3604	90.9251	43.5499	32.3304
$f_{iden} - f_{softmax} - f_{sig}$	5760	93.1849	-6.2293	-17.4488
$f_{iden} - f_{softmax} - f_{softmax}$	4079	92.0339	-4.7388	-15.9584
$f_{iden} - f_{softmax} - f_{lineartanh}$	4413	89.7752	48.5849	37.3654
$f_{iden} - f_{softmax} - f_{linearsig}$	9172	92.5940	-4.2374	-15.4570
$f_{iden} - f_{lineartanh} - f_{tanh}$	7404	94.1407	36.0181	24.7986
$f_{iden} - f_{lineartanh} - f_{sig}$	7770	95.8619	-18.7474	-29.9670
$f_{iden} - f_{lineartanh} - f_{softmax}$	3220	94.5986	-12.0421	-23.2616
$f_{iden} - f_{lineartanh} - f_{lineartanh}$	6886	95.9094	28.1601	16.9406
$f_{iden} - f_{lineartanh} - f_{linearsig}$	8303	93.0554	-4.9152	-16.1347

$f_{iden} - f_{linearsig} - f_{tanh}$	8259	93.9331	35.4808	24.2612
$f_{iden} - f_{linearsig} - f_{sig}$	9421	93.5210	-8.7147	-19.9342
$f_{iden} - f_{linearsig} - f_{softmax}$	3463	94.5620	10.0419	-21.2614
$f_{iden} - f_{linearsig} - f_{lineartanh}$	3060	95.2701	31.3153	20.0957
$f_{iden} - f_{linearsig} - f_{linearsig}$	3275	91.8024	-5.7855	-17.0050

En la tabla 5.15 se muestra el listado de simulaciones, agrupadas por las funciones de activación incluidas, junto con el resultado en porcentaje de acierto, el número de *epochs* realizados antes de cumplirse la parada temprana y los criterios AIC y MDL obtenidos después de 20 repeticiones. Como se puede observar la mejor configuración es la obtenida con la función tangente hiperbólica para las dos capas (oculta y salida). Además en promedio las alternativas que mejores resultados obtienen son aquellas que incorporan en la capa oculta este tipo de función de activación, aunque también son las que necesitan un mayor número de *epochs* para completar el aprendizaje. Por el contrario, el peor comportamiento lo presentan las redes que tienen en las neuronas de la capa oculta la función softmax en cualquiera de sus alternativas.

5.2.4.2.1. Medidas de calidad de los resultados

Para determinar cual es la mejor configuración se ha tenido en cuenta la precisión de cada alternativa y los valores de los indicadores de calidad, AIC y MDL, expresados en la Tabla 5.15 (cf. §4.6). El tiempo de computación asociado al número de *epochs* no se ha considerado crítico ya que aunque podría penalizar el tiempo necesario para realizar las siguientes simulaciones de la metodología la diferencia en *epochs* entre la propuesta elegida y el resto de mejores alternativas no es elevada. Además es conveniente recordar que cuando el modelo neuronal se encuentra en producción no es necesario volver a realizar el entrenamiento, lo que hace conveniente dar prioridad al rendimiento en lugar del tiempo. Los valores de AIC y MDL indican el compromiso entre el rendimiento y el tamaño de la red siendo conveniente que se mantengan en valores bajos. Por todo ello, el reparto de funciones de activación para las siguientes simulaciones en las capas entrada-oculta-salida será $f_{iden} - f_{tanh} - f_{tanh}$.

5.2.4.3. Tarea 2.3: Selección de la arquitectura óptima

Esta tarea y la siguiente son las encargadas de optimizar la arquitectura del PMC diseñado. La presente tarea controla los elementos ocultos, capas y neuronas, con el objetivo de mejorar el rendimiento y la calidad del clasificador. En primer lugar se utilizará un método adaptativo, del tipo AG, para localizar dentro del espacio de topologías aquellas que mejor comportamiento presenten. Para validar los candidatos anteriores se recurrirá a un proceso de prueba y error destinado a conocer el rendimiento

Tabla 5.16: Tarea 2.3. Mejores topologías para una y dos capas utilizando el AG

Parámetros AG	cf. §4.7
Topología anterior	C_1N_4
Mejor alternativa 1 capa	C_1N_8
Mejor alternativa 2 capas	$C_1N_5 - C_2N_9$

de cada una de las propuestas.

Desde el punto de vista de la eficacia del aprendizaje no es conveniente crear topologías con muchas capas, siempre es mejor resolver un problema con redes de poca profundidad (i.e. un PMC de una o dos capas ocultas) para favorecer un entrenamiento más rápido. En este caso se parte de la arquitectura PMC básica por lo que inicialmente solo tendrá una capa oculta. Si el error no se considera adecuado se procederá a incluir una segunda capa. Como ha quedado claro en §2.4.3.1, dado que un PMC con dos capas ocultas se considera un aproximador universal, la utilización de más capas ocultas no suele ser necesaria.

El objetivo del primer paso, emplear un AG, es optimizar la arquitectura mediante la localización de aquella configuración que permita reducir al mínimo el error de validación. La topología de partida incluye una sola capa oculta y cuatro neuronas en ella: C_1N_4 . En la Tabla 5.16 se muestran las mejores topologías localizadas por el AG para las capas y neuronas ocultas. La primera de ellas, C_1N_8 , utiliza una capa oculta con siete neuronas mientras que la segunda, $C_1N_5 - C_2N_9$, presenta dos capas ocultas de tres neuronas. Como los resultados obtenidos por la última configuración no son mejores, y el tiempo de computación es más elevado, no se ha seguido buscando topologías más complejas. Los principales parámetros del AG aparecen detallados en dicha tabla, y sus valores han sido fijados siguiendo los razonamientos de Lefebvre et al. [Lefebvre et al., 2007].

Para verificar las propuestas anteriores se ha implementado un proceso *batch* que permite entrenar un conjunto de arquitecturas mediante el criterio de parada temprana. En la Figura 5.15 se muestra la evolución del error MSE asociado al entrenamiento y la validación de cada alternativa. En cada gráfica se muestra el valor mínimo alcanzado en cada caso, siendo el objetivo asociado para optimizar la arquitectura al usar parada temprana, determinar el menor error de validación. La gráfica superior se refiere a las distintas alternativas para una sola capa oculta y diferente número de neuronas. La gráfica central se corresponde a las topologías con dos capas, pero variando las neuronas de la primera capa. Por último, la gráfica inferior incluye las topologías con dos capas donde se modifican las neuronas de la segunda. El número máximo de capas y neuronas se ha fijado en función de la evolución del error MSE, tanto para entrenamiento como validación.

La Tabla 5.17 muestra de forma numérica las opciones más interesantes obtenidas con

Tabla 5.17: Tarea 2.3. Resultados obtenidos para cada topología con el proceso *batch* (20 repeticiones)

Capas-Neuronas	Final MSE <i>Train</i>	Mínimo MSE <i>Train</i>	Mínimo MSE CV	Final MSE CV	% Acierto	<i>Epochs</i> promedio
C_1N_8	0.0429	0.0521	0.0953	0.1856	97.1598	9927
C_1N_9	0.0456	0.0561	0.1534	0.1965	96.1050	10401
$C_1N_5 - C_2N_9$	0.0489	0.0604	0.1137	0.1613	96.2744	12927
$C_1N_7 - C_2N_9$	0.0472	0.0543	0.1179	0.1735	96.8930	14621
C_1N_4 (PMC anterior)	-	-	-	-	96.7988	12654

las estrategias anteriores. Como se puede observar la mejor configuración en cuanto a precisión se obtiene con C_1N_8 , lo que ratifica las ideas extraídas mediante el uso del AG y cumple la regla de Cover enunciada en §3.5.1. Por tanto el resultado de esta tarea deja dicha topología como base para las siguientes simulaciones.

Tabla 5.18: Tarea 2.3. Análisis aplicando la Tarea 4.1: Indicadores AIC y MDL

Arquitectura	AIC	MDL	Arquitectura	AIC	MDL
7-8-2	-42.7309	-61.8190	7-9-2	-13.9035	-11.4459
7-5-9-2	18.6891	15.6438	7-7-9-2	24.8811	20.9275
PMC básico	-25.1340	-23.0344	PMC anterior	-26.5886	-25.3691

5.2.4.3.1. Medidas de calidad de los resultados

Para corroborar las conclusiones obtenidas durante esta tarea, a continuación se incluye un análisis enfocado a medir la calidad del PMC en función de los dos criterios estadísticos, basados en la teoría de la información descritos, en §4.6. En la Tabla 5.18, se muestran los resultados obtenidos para ambos criterios en función de los elementos ocultos (capas y neuronas) de la arquitectura de red. Los modelos neuronales analizados son los mejores candidatos obtenidos a partir del estudio realizado en la Tarea 2.3. La lista de arquitecturas que aparece en la tabla indica el número de neuronas por capa, i.e. *entrada- oculta- salida* para una capa oculta y *entrada- oculta 1 - oculta 2 - salida* para dos capas ocultas. La conclusión principal de este estudio es que el ajuste del modelo empeora a medida que aumenta la complejidad de la arquitectura de red. Teniendo en cuenta que la topología que mejor comportamiento ha mostrado en el análisis de la Tarea 2.3 ha sido la configuración 7-8-2, y siendo la que mejor puntuación ha obtenido en criterios de calidad, dicha alternativa será la candidata seleccionada.

A modo de referencia se incluye en esta tabla la calidad del PMC original, tal y como

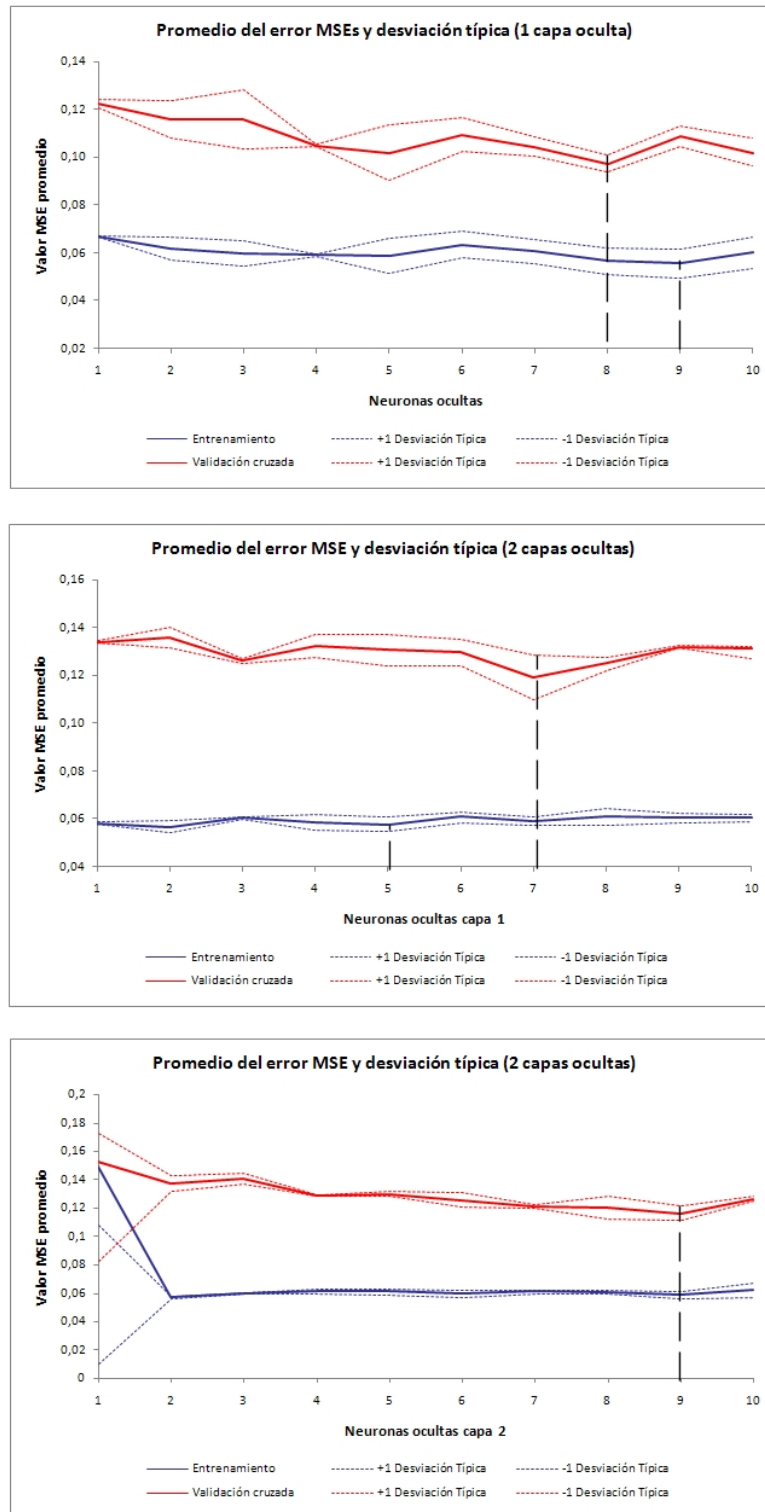


Figura 5.15: Tarea 2.3. Promedio del error MSE para las diferentes arquitecturas

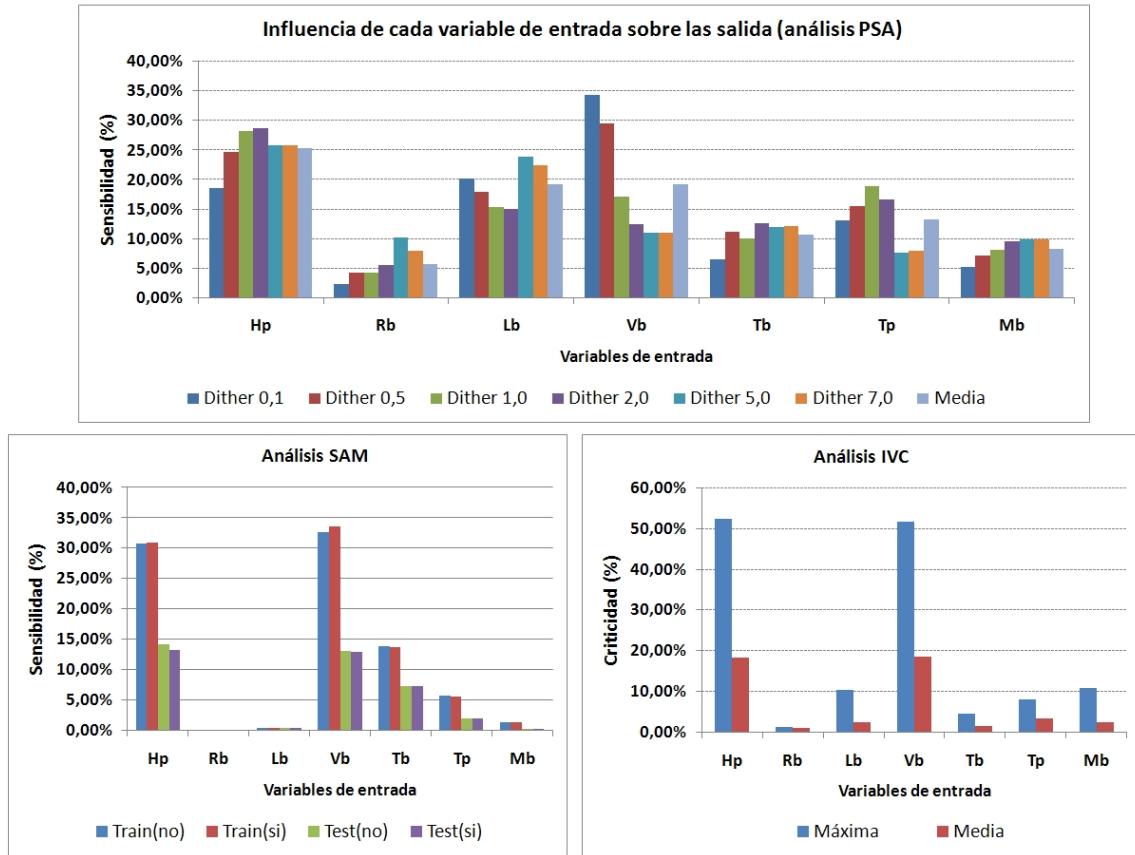


Figura 5.16: Tarea 2.4 (arriba) Influencia de cada variable sobre las salidas usando PSA (abajo izquierda) Sensibilidad media usando SAM y (abajo derecha) Indicador de criticidad de usando IVC.

aparece en §5.2.1, y del PMC anterior obtenido en la Tarea 2.2 (cf. §5.2.4.2). Como se observa los modelos neuronales asociados al PMC optimizado presentan un mejor ajuste sobre los datos desde el punto de vista de estos indicadores estadísticos.

5.2.4.4. Tarea 2.4: Selección del vector de entradas óptimo

Esta tarea está orientada a determinar la influencia que tienen las diferentes variables, que forman el vector de entradas, sobre las predicciones de la red.

Como se mencionó anteriormente, uno de los factores que más influye en la cantidad de datos de necesarios para asegurar el rendimiento de una RNA, es la complejidad de su arquitectura. A fin de reducirla hay varias alternativas tales como la poda de la red o la eliminación de los pesos asociados con las conexiones. En este caso, se ha optado por estudiar la relación existente entre el aprendizaje y el error de generalización en función de la complejidad del modelo, por medio de la reducción del tamaño del vector de entrada. Para ello se analizará la influencia de cada componente de dicho vector

utilizando diferentes métodos de sensibilidad.

El primer análisis, mediante el método PSA, se llevó a cabo aumentando de forma temporal cada uno de los valores de entrada por una pequeña cantidad (*dither*) dentro del rango [0.1-7.0]. Dentro de estos valores, la relación causa-efecto entre las entradas y salidas se refleja con claridad. El segundo análisis, mediante el método SAM, fue ejecutado mediante un proceso por lotes utilizando un PMC diseñado con el vector de entradas completo. Para cada una de las entradas, se realizaron variaciones sobre su media de ± 1 a partir de su desviación estándar, mientras el resto de variables de entrada se mantuvieron constantes. Este proceso se ejecutó 50 veces durante el testeado del PMC, repitiéndose para cada una de los componentes del vector. El análisis se ha llevado a cabo para los conjuntos de *train* y *test*, desglosando los resultados en los valores de la clase perforación (sí / no). Por último, los valores del método IVC se calcularon una vez terminado el aprendizaje, lo que permite determinar la criticidad de los valores que cada variable toma durante las fase de testeado de la red.

La Figura 5.16 muestra el resultado obtenido al aplicar los tres métodos de análisis de sensibilidad tras 20 repeticiones. El gráfico superior refleja la medición de las perturbaciones sobre la salida mediante el análisis de PSA para cada una de las variables de entrada. La parte inferior izquierda presenta los resultados del método de SAM para los conjuntos de *train* y *test*. Por último, la parte inferior derecha del gráfico incluye los resultados del método IVC para cada variable, indicando el valor medio de criticidad y el valor máximo obtenido.

Los resultados obtenidos a través del análisis PSA y SAM, muestran la sensibilidad que las variables de entrada tienen frente a los cambios en sus valores una vez que la red se ha entrenado. Los componentes del vector de entrada, con una sensibilidad más baja son, por una parte, las propiedades geométricas del proyectil, R_b y L_b , y por otro la masa M_b . Por el contrario, las variaciones en los valores de las variables H_p y V_b producen grandes perturbaciones en la salida de la red. El método IVC muestra cómo los valores utilizados en los conjuntos de *train* y *test* influyen en la calidad de la predicción. Cuando los valores H_p y V_b están fuera de rango durante el test, la red no genera aproximaciones correctas. Por lo tanto, una forma de mejorar la salida del modelo neuronal es mediante una correcta distribución de los ensayos balísticos disponibles entre los diferentes conjuntos, prestando especial atención a las variables H_p y V_b . Este hecho se ha comprobado durante las simulaciones, ya que en algunos casos, si la distribución de los ensayos entre los grupos no era correcta, la precisión de la red se ha reducido drásticamente, hasta alcanzar tasas de acierto cercanas al 40 %.

Para comprobar las deducciones anteriores se ha estudiado la influencia de cada componente del vector de entrada mediante el método SWM. En las simulaciones realizadas se ha ido eliminando de forma gradual cada una de las variables, midiendo en cada caso la capacidad de generalización de la red. En la Figura 5.17 se incluye el

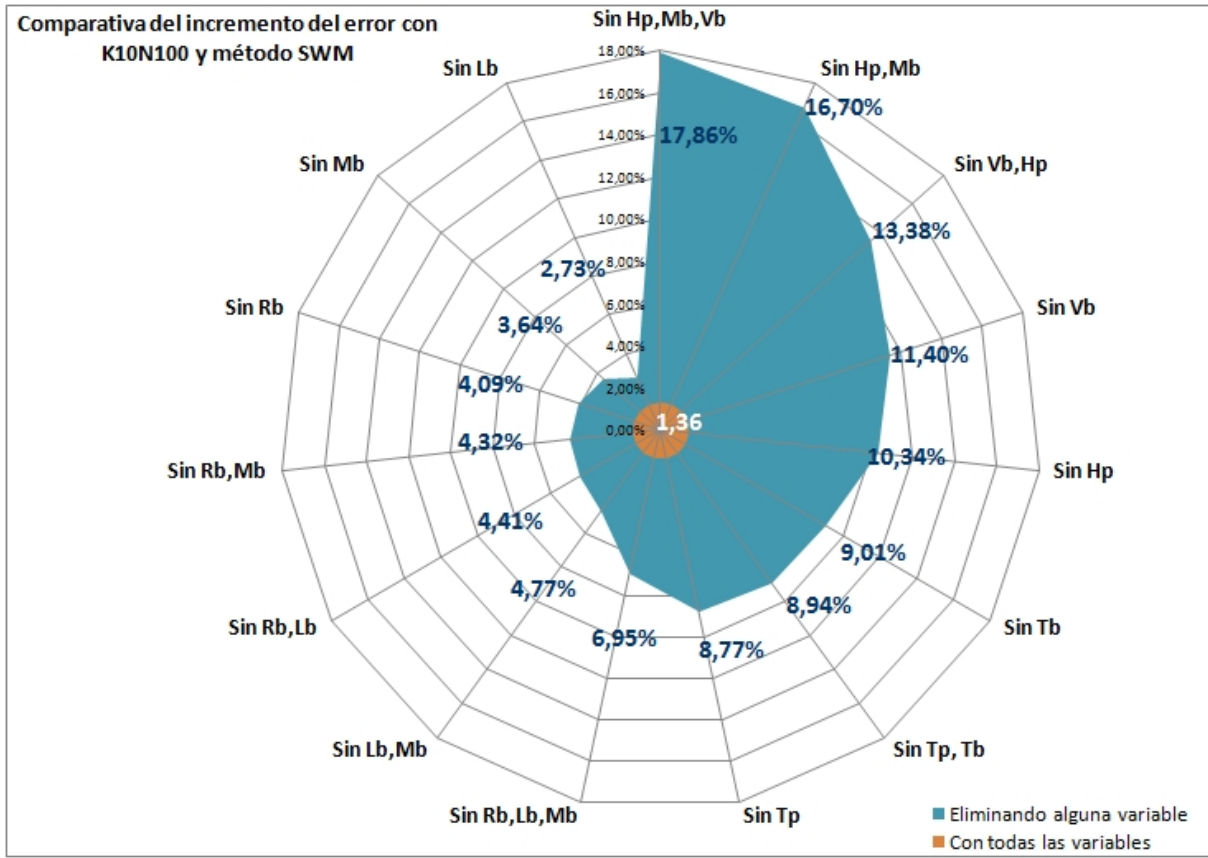


Figura 5.17: Tarea 2.4. Estudio comparativo del incremento del error utilizando SWM (20 repeticiones)

método SWM y se expone el aumento del error que se obtiene con cualquiera de las configuraciones alternativas del vector de entrada respecto a la configuración original con todas las variables. En este caso, los resultados siguen verificando las conclusiones extraídas anteriormente, las variables más influyentes han sido H_p y V_b . En todos los modelos de redes en los que estas dos variables no se han incluido el error de generalización ha crecido de forma considerable. La eliminación de la variable V_b , conlleva un incremento en el error del 11.4018 %, mientras que sin H_p aumenta en 10.3403 %. En promedio, cuando cualquiera de estas variables es eliminada, el aumento del error respecto a la configuración con todas las variables es del 13.9348 %.

En la Tabla 5.19 se muestra el vector de entradas óptimo localizado por el AG implementado. En este caso el mejor comportamiento se consigue sin las variables R_b y L_b lo cual corrobora las conclusiones anteriores. Las características geométricas del proyectil parece que son redundantes dentro del vector de entradas. Esto es lógico ya que, desde el punto de vista físico, el ratio L_b/R_b y la masa M_b están correlacionados con R_b y L_b ya que la densidad es constante:

$$M = \rho \cdot \pi \cdot R^2 \cdot L \cdot d \quad (5.2)$$

Tabla 5.19: Tarea 2.4. Mejor vector de entrada localizado utilizando el AG

Parámetros AG	cf. §4.7
Vector de entradas inicial	$L_b, M_b, R_b, V_b, H_p, T_p, T_b$
Vector de entradas óptimo encontrado	M_b, V_b, H_p, T_p, T_b

donde ρ es una la densidad asociada al material del proyectil.

Si se tiene en cuenta que uno de los objetivos asociado a esta investigación es reducir la complejidad de la simulación numérica, a raíz de los resultados obtenidos con las dos últimas técnicas, AG y método SWM, se pueden considerar como una buena alternativa los vectores reducidos $V_{red1} = \{L_b, R_b, V_b, H_p, T_p, T_b\}$ o $V_{red2} = \{M_b, V_b, H_p, T_p, T_b\}$. Como consecuencia de esto se podría optar por diseñar una arquitectura de PMC de menor complejidad, lo que conllevaría un menor tiempo de aprendizaje y que fueran necesarios menos patrones para obtener un rendimiento correcto.

Tabla 5.20: Tarea 2.4. Análisis aplicando la Tarea 4.1: Indicadores AIC y MDL

Configuración	AIC	MDL	Configuración	AIC	MDL
Sin H_p, M_b, V_b	64.4918	69.8653	Sin T_p	-3.3931	0.8185
Sin H_p, M_b	62.4814	67.4192	Sin R_b, M_b	-5.1617	-5.6263
Sin R_b, L_b, M_b	45.3342	26.2461	Sin L_b, M_b	-8.7363	-5.7790
Sin V_b, H_p	63.2986	44.2105	Sin R_b, L_b	-9.9906	-7.8244
Sin V_b	30.2575	34.4691	Sin M_b	-14.1799	-9.9683
Sin T_p, T_b	13.9889	18.9267	Sin L_b	-19.9906	-15.7790
Sin H_p	0.8974	5.1090	Sin R_b	-38.3859	-50.6442
Sin T_b	-0.5910	3.6205	Todas las variables	-42.7309	-61.8190
PMC básico	-25.1340	-23.0344			

5.2.4.4.1. Medidas de calidad de los resultados

En la Tabla 5.20 se muestran los resultados obtenidos para los criterios AIC y MDL en función del tamaño del vector de entradas. El modelo neuronal sobre el que se ha aplicado esta fase es el mejor modelo obtenido en las fases anteriores y las configuraciones de red analizadas en la Tarea 2.4. La lista de configuraciones de RNA se ordena a partir de los que han mostrado un peor comportamiento, i.e. un mayor AIC y MDL. En este caso, los modelos que mejor se adaptan a los datos utilizados sin sufrir deterioro en el rendimiento (en términos de su capacidad de generalización) son los que incluyen el vector V_{ent} completo o los que no incorporan las características geométricas del proyectil: R_b y L_b y M_b . En contraste, los modelos que no utilizan H_p y V_b , tanto de forma individual o conjunta, tienen un mal ajuste sobre los datos a pesar de tener una menor complejidad

estructural.

A modo de referencia se incluye en esta tabla la calidad del PMC original tal y como aparece en §5.2.1. Como se observa los modelos neuronales asociados al PMC optimizado presentan una calidad superior desde el punto de vista de estos indicadores.

5.2.4.5. Sumario

Una vez terminada esta segunda fase se pueden sacar las siguientes conclusiones que serán aplicadas sobre la última fase y sus correspondientes tareas:

- La inicialización de los pesos de la red y la elección de un conjunto de funciones de activación adecuado mejora la precisión del PMC. La mejor configuración en las capas de entrada-oculta-salida ha sido $f_{iden} - f_{tanh} - f_{tanh}$.
- La mejor topología de red se ha localizado gracias al uso de un AG y su posterior ratificación mediante el diseño y simulación de diferentes PMC. La alternativa con un comportamiento más preciso ha sido: C_1N_8 , i.e. una sola capa oculta con 8 neuronas.
- El estudio del vector de entrada ha revelado que las características geométricas del proyectil aportan información que puede tomarse como redundante, si se utiliza de forma conjunta las variables M_b , R_b y L_b . La utilización de un vector reducido supone resultados menos precisos, entre un 2.7312 % y un 4.7758 % de penalización, pero reduce la complejidad asociada a la simulación numérica. Además permite que la arquitectura del PMC sea más sencilla, lo que conlleva menos tiempo de aprendizaje y la necesidad de un conjunto de patrones más reducido. No obstante, una vez obtenidos los ensayos balísticos los costes computacionales de la simulación numérica no pueden tenerse en cuenta, por lo que para no mermar la calidad del PMC se continuará utilizando el vector V_{ent} completo. Para futuras estudios podría tenerse en cuenta las conclusiones obtenidas a raíz de esta fase de la metodología, lo que permitiría reducir los tiempos de computación derivados de la herramienta de simulación numérica y del entrenamiento del PMC.
- El valor máximo de acierto conseguido al finalizar esta fase ha sido 97.1598 % obtenido con el vector de entradas completo y el reparto de elementos ocultos C_1N_8 .

5.2.5. Fase 3: Algoritmo de entrenamiento

Esta fase está orientada a optimizar aquellos aspectos relativos al algoritmo de aprendizaje del PMC. Se desglosa en dos tareas secuenciales y otras dos que se ejecutan de forma paralela al compartir algunos elementos de la configuración del algoritmo. La configuración usada en esta fase parte de la inicial, C_0 , y de las obtenidas en la primera y segunda fase, C_1 y C_2 . Por tanto, esta nueva configuración se denomina configuración

de prueba tres, C_3 , y vuelve a estar dividida en tres bloques: patrones (P_1), arquitectura (A_1) y algoritmo (R_1). Sus características más interesantes son las siguientes:

- Técnica $K_{10}A_{100}$ con 20 repeticiones ($20 * 10cv$).
- Distribución de patrones en 10 conjuntos de similar tamaño (320 patrones en 10 conjuntos, 9 se usan para entrenar y 1 para testear) y normalización z -score.
- Topología 7-8-2 y función de activación $f_{iden}-f_{tanh}-f_{tanh}$.
- Criterio de parada temprana y un máximo de 20000 *epochs*.
- Vector de entradas: $V_{ent} = \{M_b, L_b, R_b, V_b, H_p, T_p, T_b\}$.

5.2.5.1. Tarea 3.1. Criterio de convergencia

Como ha quedado expuesto en §3.5.3, el algoritmo usado para realizar el aprendizaje del PMC es el RPE. Éste se encarga de minimizar el error que se obtiene durante el proceso de entrenamiento, siendo necesario determinar cuando detenerlo para obtener un aprendizaje adecuado. Para saber cuándo se ha alcanzado este punto se han incluido una serie de alternativas denominadas criterios de parada o convergencia. En §4.5.1 se han descrito las diferentes posibilidades así como sus ventajas y problemas asociados.

En las simulaciones en las que sólo se utilizan los conjuntos clásicos de *train-test*, se recurrirá a los siguientes criterios usados de forma conjunta o individual:

- Evaluar el error MSE.
- Detener al alcanzar un número de ciclos o *epochs*.

Como consecuencia del posible sobreaprendizaje asociado a los criterios anteriores, se utilizará un último criterio de parada que implica la utilización del tercer conjunto de datos disponible, el de validación o CV:

- Evaluar el error MSE obtenido en validación.

Si bien esta tarea se encuentra situada dentro de la fase correspondiente a la optimización del algoritmo de aprendizaje, en todas las simulaciones de esta metodología y en función de la configuración correspondiente, al menos uno de estos criterios ha sido utilizado durante el entrenamiento del PMC.

Tabla 5.21: Tarea 3.2 Análisis estadístico (20 repeticiones)

Estadístico	(medido en % acierto)		(medido en % error)
	<i>Batch</i>	Secuencial	Diferencias <i>Batch</i> -Secuencial
Media	97.1665	92.8844	-4.2821
Varianza	0.4321	1.6681	1.9192
Desviación típica	0.6573	1.2915	1.3853
Mínimo	95.6375	90.025	-7.1875
Máximo	97.8060	94.7125	-2.1751
Rango	2.2625	4.6875	5.0125
Asimetría tipificada	-1.5101	-0.9877	-0.5206
Curtosis tipificada	-0.2245	-0.1198	-0.2864
<i>Epochs</i> promedio	9891	8523	

5.2.5.2. Tarea 3.2: Modos de entrenamiento

En el §4.5.2 se han detallado las ventajas e inconvenientes asociadas a los modos de entrenamiento secuencial y *batch*. Teniendo en cuenta que en el modo elegido de partida ha sido el *batch*, en este apartado se comparará su rendimiento con la otra alternativa, el entrenamiento secuencial.

En la Figura 5.18 se muestra el diagrama de histogramas y de caja. El gráfico superior muestra dos histogramas, uno para cada conjunto de muestras, con el histograma de la segunda muestra invertido debajo del de la primera, lo que permite visualizar la forma de las dos distribuciones asociadas a cada modo de entrenamiento. Por su parte en el diagrama de caja de cada alternativa se muestran entre otros datos de interés los valores mínimo y máximo, la mediana (línea horizontal), la media (símbolo cruz) y los posibles valores atípicos. La muesca de la caja indica el intervalo de confianza del 95 % y permite aproximar entre que dos valores se estima que estará cierto valor desconocido para esa determinada probabilidad de acierto.

Los resultados analizados desde un punto de vista estadístico se muestran en la Tabla 5.21. De particular interés son los estadísticos asimetría estandarizada y curtosis estandarizada ya que usarse para determinar si las muestras proceden de distribuciones normales. Los valores de estos estadísticos fuera del rango de -2 a +2 indican salidas significativas de normalidad, lo que tendería a invalidar las pruebas que comparan las desviaciones normales. En este caso, tanto para cada modo de entrenamiento como para las diferencias obtenidas entre ambos, los valores de asimetría y curtosis estandarizada están dentro del rango esperado, lo que permite un análisis paramétrico de resultados.

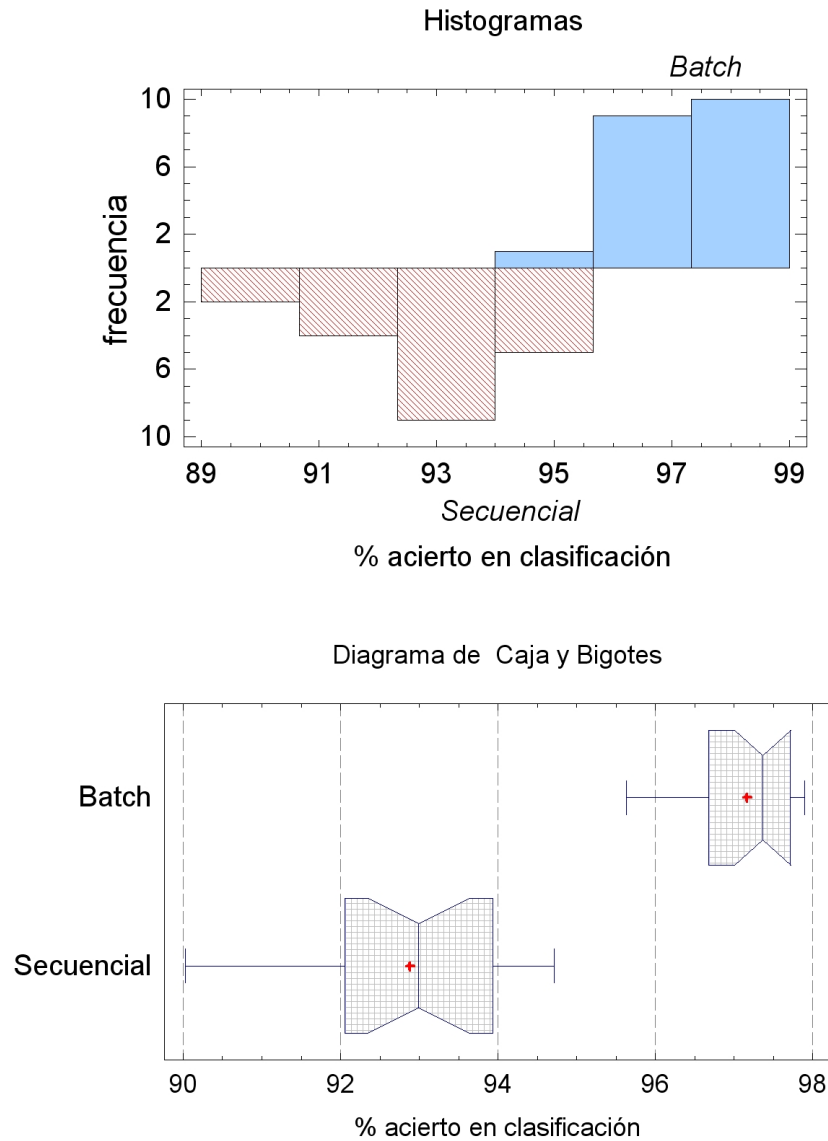


Figura 5.18: Tarea 3.2. Diagramas de histogramas y de caja

5.2.5.2.1. Medidas de calidad de los resultados

A la vista de los resultados obtenidos en la Tabla 5.21, se puede establecer que no existen evidencias de un posible alejamiento de la normalidad, por lo que se incluye la prueba t-test pareada como contraste de homogeneidad entre las poblaciones del error. Como se ha utilizado la técnica $K_{10}A_{100}$, con 10 conjuntos y 20 repeticiones, se tiene la configuración $20 \times 10cv$, siendo la hipótesis de partida que la diferencia entre los errores muestrales de ambos algoritmos tiene media cero. Con este contraste se pretende demostrar que existen diferencias significativas entre los errores asociados a cada modo de entrenamiento.

- Contraste t:

- Hipótesis nula H_0 : media = 0.0
- Hipótesis alternativa H_1 : no igual
- Estadístico $t = -13.8239$
- P-valor = $2.2934 \cdot 10^{-11}$
- Se rechaza la hipótesis nula para $\alpha = 0.05$

Este contraste de homogeneidad se basa en la prueba t-test, cuya hipótesis nula es que la diferencia media de *Batch*-Secuencial sea igual a cero frente a la hipótesis alternativa de que no lo sea. Puesto que el p-valor para este test es inferior a 0.05, se puede rechazar la hipótesis nula para un nivel de confianza del 95.0%. Por tanto, existen evidencias estadísticas que determinan las diferencias significativas entre los resultados de cada modo de entrenamiento. A la vista de los resultados y del estudio estadístico, el modo de entrenamiento *batch* se seguirá utilizando en las siguientes fases y tareas de la metodología.

5.2.5.3. Tarea 3.3: Heurísticas asociadas a los parámetros del algoritmo de aprendizaje

El valor óptimo de la tasa de aprendizaje η y de la constante *momentum* μ pueden establecerse a través de un método de búsqueda lineal o heurístico, aunque es habitual seleccionar un valor concreto ad-hoc para un problema concreto. En esta fase de la investigación se incluyen una serie de alternativas, agrupadas en dos bloques para determinar los valores de η y μ .

El primer bloque engloba diferentes variantes del algoritmo LRS para buscar el valor más adecuado de η (cf. §4.5.3). Las alternativas incluidas han sido LRS Lineal (LRSL), LRS Logarítmico (LRSO) y LRS Exponencial (LRSE). Las ecuaciones asociadas a cada alternativa son Ecuación 4.27 para LRSL, Ecuación 4.28 para LRSO y Ecuación 4.29 para LRSE. Todas las propuestas reciben como parámetro a η y modifican su valor (incrementado o disminuyendo) de forma lineal, exponencial o logarítmica durante un número de *epochs* especificado. Además se especifican una restricción máxima y mínima que debe alcanzarse durante todo el proceso de adaptación. En este caso, los valores empleados para las simulaciones realizadas han sido valor mínimo cero, valor máximo uno y Beta -0.01 (comenzando en el *epoch* uno hasta 1000).

El segundo bloque responde a la utilización de un AG para realizar una búsqueda heurística de los valores óptimos de η y μ . En la Tabla 5.22 se muestran los mejores valores localizados, probándose a continuación la capacidad de una red bajo esta configuración para medir su desempeño respecto a las otras alternativas. Los resultados obtenidos para todas las propuestas aparecen en la Tabla 5.23. Como se puede observar los valores localizados por el AG para η y μ permiten mejorar el comportamiento del PMC, tanto en precisión como en tiempo de entrenamiento.

Tabla 5.22: Tarea 3.3. Mejores valores de η y μ utilizando el AG

Parámetros AG	cf. §4.7
η inicio (oculta-salida)	1.0-0.1
μ inicio (oculta-salida)	0.7-0.7
η encontrado (oculta-salida)	0.2520-0.4219
μ encontrado (oculta-salida)	0.1023-0.6433

Tabla 5.23: Tarea 3.3. Resultados obtenidos utilizando diferentes heurísticas (20 repeticiones)

Heurística	Acierto mínimo (%)	Acierto promedio (%)	Acierto máximo (%)	Epochs promedio
AG	94.8375	97.3875	98.8301	6209
LRSE	92.3375	95.2750	97.3375	7118
LRS�	95.1501	96.7437	98.4901	8102
LRSO	86.4502	88.5809	91.8251	6901
PMC anterior	95.6375	97.1665	97.8060	9891

5.2.5.3.1. Medidas de calidad de los resultados

Para ratificar las conclusiones anteriores, se ha incluido un contraste de homogeneidad de las diferentes heurísticas para comprobar si proceden de la misma distribución de población. Desde el punto de vista gráfico, en la Figura 5.19, se incluye el diagrama de dispersión y el de caja donde se puede observar la simetría, si existen patrones de comportamiento, la independencia entre observaciones y tendencias en general. Como se observa los pares de muescas de la mediana no se solapan en anchura, por lo que podría existir diferencia estadística significativa entre ellas para un 95 % de nivel de confianza. Además en este caso, no hay puntos externos ni puntos externos lejanos. Por otro lado, la Figura 5.20 incluye la gráfica de residuos y la del análisis de medias de cada una de las alternativas. La primera incluye los residuos obtenidos para las diferentes heurísticas y permite observar que la variabilidad dentro de cada una es aproximadamente la misma. La segunda realiza el análisis de medias para cada una de las tres propuestas junto con la media global y el 95 % de límites de decisión. Las muestras que se encuentran fuera de los límites de decisión, AG, LRS� y LRSO, son significativamente diferentes de la media global.

Para poder emplear el método ANOVA, una de los requisitos es que las varianzas de las diferentes poblaciones sean similares. A raíz del diagrama de caja de la Figura 5.19 parece que esto no se cumple. Para ratificarlo se ha incluido un análisis de contraste de la varianza mostrado en la Tabla 5.24. Como los tres p-valores calculados son superiores a 0.05, se puede establecer que no hay diferencia estadística significativa entre las desviaciones típicas para un nivel de confianza del 95.0 %.

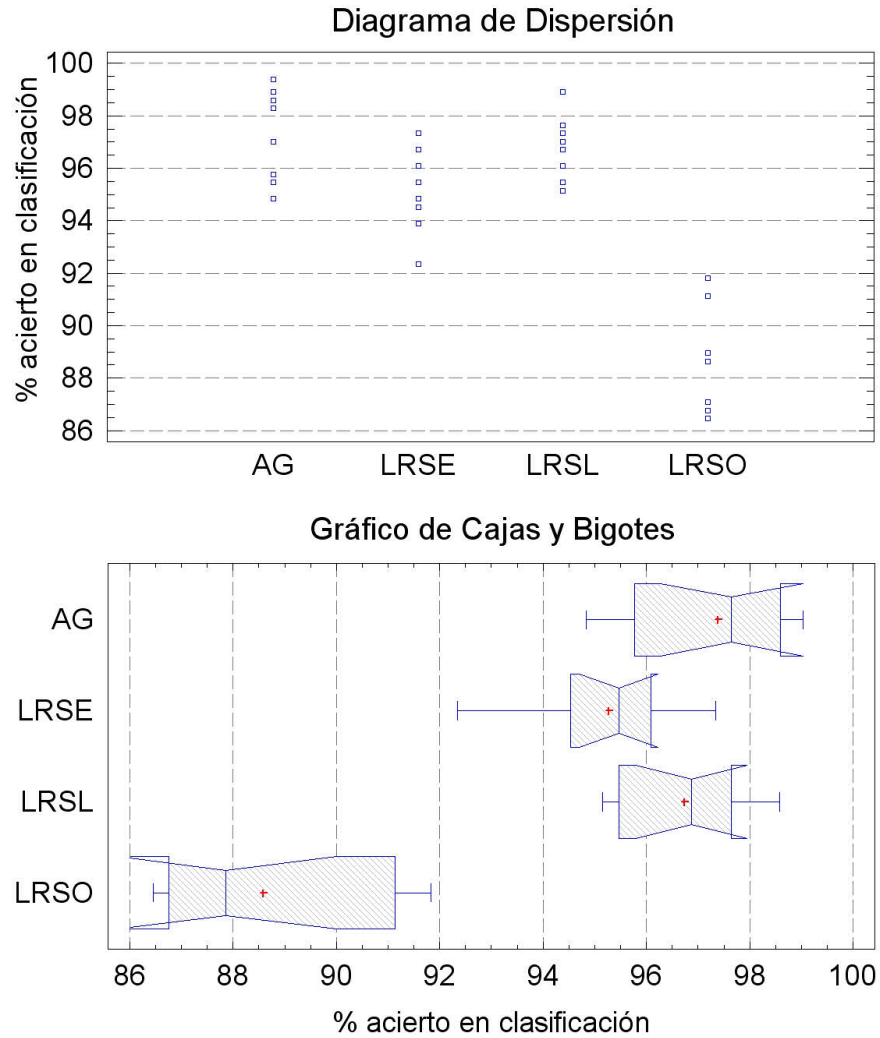


Figura 5.19: Tarea 3.3. Diagramas de dispersión y caja

Como se cumplen las condiciones para aplicar el método ANOVA, en la Tabla 5.25 se muestra dicho método aplicado sobre los resultados obtenidos por cada una de las heurísticas analizadas. La varianza del porcentaje de acierto se descompone en dos componentes: uno entre grupos y otro dentro de los grupos. El F-ratio, que en este caso es igual a 62.25, es el cociente de la estimación entre grupos y la estimación dentro de los grupos. Puesto que el p-valor del test F es inferior a 0.05, hay diferencia significativa entre las medias del porcentaje de acierto para cada una de las alternativas para un nivel de confianza del 95.0 %.

Para determinar las medias que son significativamente diferentes unas de otras, se ha empleado el procedimiento de comparación múltiple o test CRM. La Tabla 5.26 muestra la diferencia estimada entre cada par de medias. El asterisco que se encuentra al lado de cuatro de los pares analizados, del total de $\binom{4}{2}=6$ combinaciones, indica diferencias

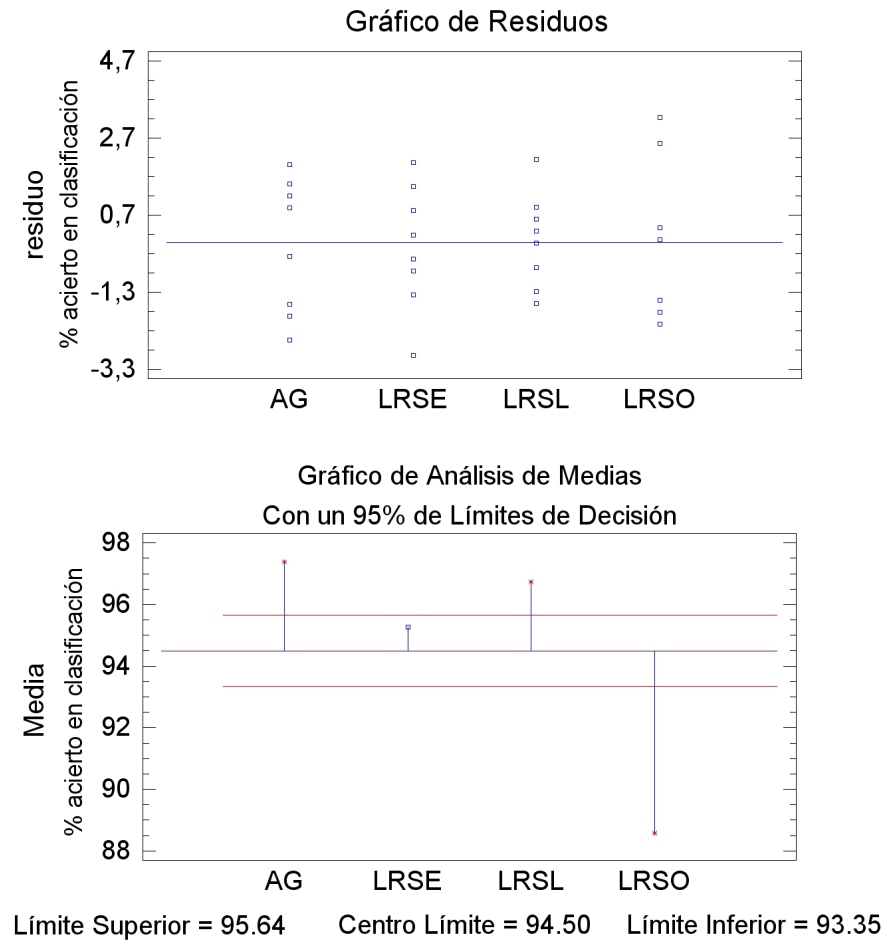


Figura 5.20: Tarea 3.3. Análisis de los residuos y la media

estadísticamente significativas a un nivel de confianza 95,0% y la columna límites define la mitad de la anchura de los intervalos de confianza para cada diferencia de medias de los diferentes pares de alternativas. En la Tabla 5.27, ordenados por su media, se identifican tres grupos homogéneos según la alineación del símbolo \checkmark en la columna. Dentro de cada columna, los niveles que tienen \checkmark forman un grupo de medias entre las cuales no hay diferencias estadísticamente significativas. El método utilizado para discernir entre las medias es el procedimiento LSD.

Los valores de η y μ obtenidos con la heurística del AG presentan diferencias significativas desde el punto estadístico con las alternativas LRSE y LRSO. Con respecto a la técnica LRSL el análisis revela que no se encuentran diferencias entre ambas, pero debido a que los resultados obtenidos por el AG son sensiblemente mejores, se considerará esta propuesta como mejor candidata. La siguiente tarea, orientada a la optimización del algoritmo RP, se ejecuta de forma paralela por lo que los valores obtenidos en este punto no serán utilizados en las próximas evaluaciones. Una vez finalizadas las

Tabla 5.24: Tarea 3.3. Análisis aplicando la Tarea 4.2: Contraste de la varianza

Contraste	Valor	p-valor
C de Cochran	0.4173	0.2409
Bartlett	1.0853	0.4205
Test de Levene	1.4693	0.2391

Tabla 5.25: Tarea 3.3. Análisis aplicando la Tarea 4.2: Método ANOVA

Fuente	Sumas de Cuadrados	Grados Libertad	Cuadrado Medio	Cociente-F	P-Valor
Entre grupos (SCE)	490.043	3	163.348	62.25	0.0000
Intra grupos (SCR)	94.4648	36	2.6240		
Total (SCT)	584.507	39			

dos tareas se evaluarán las mejores alternativas de cada una para contrastar su validez.

5.2.5.4. Tarea 3.4: Optimización numérica del algoritmo de aprendizaje

Esta tarea ha sido dividida en dos subtareas, encaminadas a mejorar el rendimiento del PMC a partir de diferentes criterios relacionados con el algoritmo de aprendizaje. En la primera se analizarán diferentes alternativas utilizadas para minimizar la función de coste incluida en el algoritmo. Una vez determinada la mejor opción, en la segunda se estudiarán diferentes alternativas orientadas a la optimización numérica del algoritmo.

En la Tabla 5.28 se detallan todas las propuestas estudiadas dentro de este apartado. Con respecto a las funciones de coste se incluyen las ecuaciones empleadas para calcular el error de la red para un patrón de entrada n y una salida de la red k . Por su parte, para cada una de las alternativas para la optimización del algoritmo se incluyen sus características y parámetros más importantes.

Tabla 5.26: Tarea 3.3. Análisis aplicando la Tarea 4.2: Test CRM

Contraste	Diferencias	+/- Límites
AG - LRSE	*2.1125	1.4692
AG - LRSL	0.6437	1.4692
AG - LRSO	*8.8062	1.4692
LRSE - LRSL	-1.4687	1.4692
LRSE - LRSO	*6.6937	1.4692
LRSL - LRSO	*8.1625	1.4692

Tabla 5.27: Tarea 3.3. Análisis aplicando la Tarea 4.2: Grupos homogéneos

Heurística	Media (%)	Grupos homogéneos	
LRSO	88.5809	✓	
LRSE	95.2750	✓	
LRSL	96.7437	✓	✓
AG	97.3875		✓

Tabla 5.28: Tarea 3.4. Listado de propuestas para la mejora del algoritmo de aprendizaje

Propuesta	Acrónimo	Descripción
PMC con norma Manhattan	L_1	$J_{n,k} = d_{n,k} - y_{n,k} ^1$
PMC con norma Euclídea	L_2	$J_{n,k} = (d_{n,k} - y_{n,k})^2$
PMC con norma Minmax	L_{inf}	$J_{n,k} = \tan(d_{n,k} - y_{n,k})^2$
PMC con norma Manhattan y AG	L_1AG	$J_{n,k} = d_{n,k} - y_{n,k} ^1$ y AG para η y μ
PMC con norma Euclídea y AG	L_2AG	$J_{n,k} = (d_{n,k} - y_{n,k})^2$ y AG para η y μ
PMC con norma Minmax y AG	$L_{inf}AG$	$J_{n,k} = \tan(d_{n,k} - y_{n,k})^2$ y AG para η y μ
Algoritmo del Gradiente	GR	Método de mejora sólo con parámetro (capa oculta-salida) $\eta = 1.0-0.4$
Regla Delta-Bar Extendida	RBDE	Método de mejora con parámetros η y μ adaptativos. De inicio $\eta = 1.0-0.4$ y $\mu = 0.7-0.3$
Algoritmo Quick-Propagation	QP	Método de mejora con parámetros (capa oculta-salida) $\eta = 1.0-0.4$ y $\mu = 0.7-0.4$
Algoritmo del Gradiente Conjugado	GC	Método de mejora de segundo orden para el gradiente
Algoritmo Levenberg Marquardt	LM	Método de mejora de segundo orden para el gradiente

5.2.5.4.1. Tarea 3.4.1: Funciones de coste

Las normas L_p actúan a modo de regularización y permiten dar diferentes soluciones a un mismo problema modificando los pesos en función de la norma elegida. La norma L_1 implementa la función de coste de error absoluto. Se trata de un criterio utilizado sobre todo para redes que procesan datos de tipo binario. En este caso el error devuelto al procedimiento de entrenamiento es la diferencia entre la salida de la red y la respuesta deseada. Por su parte la norma L_2 implementa la función de coste cuadrática, siendo la función de coste más aplicada en sistemas adaptativos. El error que se le pasa al procedimiento de entrenamiento es el cuadrado de la distancia euclídea entre la salida de la red y la respuesta deseada. Por último la norma L_{inf} enfatiza los errores grandes en cada una de las salidas. Para ello aplica la función tangente a cada uno de los errores obtenidos por la norma L_2 . Por otro lado, como en esta tarea se emplea el algoritmo

Tabla 5.29: Tarea 3.4.1. Resultados obtenidos utilizando diferentes funciones de coste (20 repeticiones)

Función de coste	Acierto mínimo (%)	Acierto promedio (%)	Acierto máximo (%)	Epochs promedio
L_1	93.5466	94.6102	95.6186	10772
L_2	95.4910	97.1069	98.1950	9927
L_{inf}	94.8884	96.0388	97.1880	9978
L_1AG	94.0125	95.0750	96.2001	9023
L_2AG	96.0500	97.3908	98.2375	6846
$L_{inf}AG$	95.875	96.6667	97.4375	8841
PMC Tarea 3.2	95.6375	97.1665	97.8060	9891
PMC Tarea 3.3	94.8375	97.3875	98.8301	6205

RPE y para aprovechar las ventajas que ofrecen los parámetros η y μ , se ha incluido un componente genético para localizar los valores más apropiados para cada una de las normas L_p . De esta forma se podrá comparar el rendimiento del PMC con el algoritmo RPE, diferentes normas y valores de η y μ .

En la Tabla 5.29 se muestran los resultados obtenidos para cada una de las normas, empleando o no la opción del AG para calcular los parámetros del algoritmo RPE. Como se puede observar la norma L_2 es la que presenta un mejor comportamiento en ambos casos, tanto en acierto máximo como en promedio. La presencia de los datos atípicos detectados en §3.4.7 no ha tenido efecto pernicioso sobre ella como podía pensarse en un primer momento (cf. §4.5.4). Además según los estudios de Lefebvre et al. [Lefebvre et al., 2007] apenas existen diferencias entre las normas L_1 y L_2 pudiéndose aplicar cualquiera de ellas en todo tipo de problemas. No obstante la primera se ajusta mejor a problemas de regresión y tanto ella como la norma L_{inf} necesitan más tiempo de entrenamiento para conseguir ajustar los datos. Además la segunda es más sencilla de aplicar en sistemas lineales y no lineales, al minimizar el error y no el valor absoluto del mismo.

Tabla 5.30: Tarea 3.4.1. Análisis aplicando la Tarea 4.2: Contraste de la varianza

Contraste	Valor	p-valor
C de Cochran	0.2020	1.0
Bartlett	1.0295	0.7951
Test de Levene	0.5319	0.7514

Tabla 5.31: Tarea 3.4.1. Análisis aplicando la Tarea 4.2: Método ANOVA

Fuente	Sumas de Cuadrados	Grados Libertad	Cuadrado Medio	Cociente-F	P-Valor
Entre grupos (SCE)	94.2924	5	18.8585	44.6508	0.0000
Intra grupos (SCR)	35.4778	84	0.4223		
Total (SCT)	129.77	89			

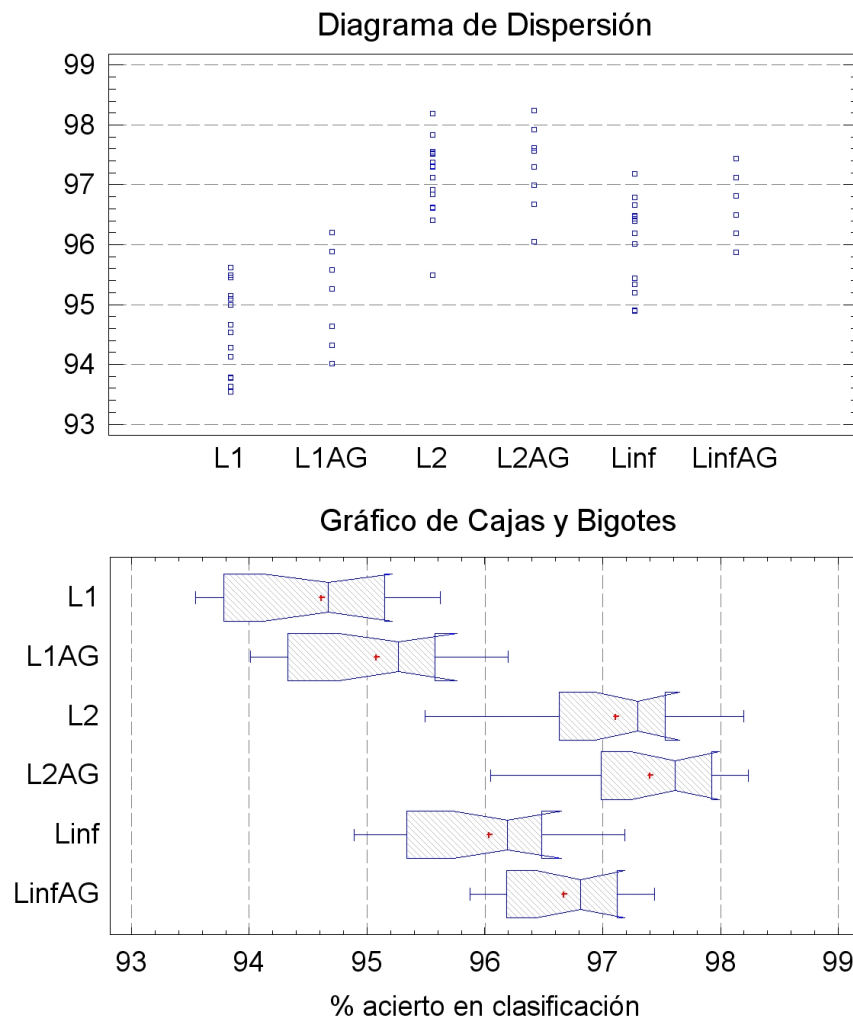


Figura 5.21: Tarea 3.4.1. Diagramas de dispersión y caja

5.2.5.4.2. Medidas de calidad de los resultados

Para ratificar las conclusiones anteriores se ha incluido un estudio estadístico para

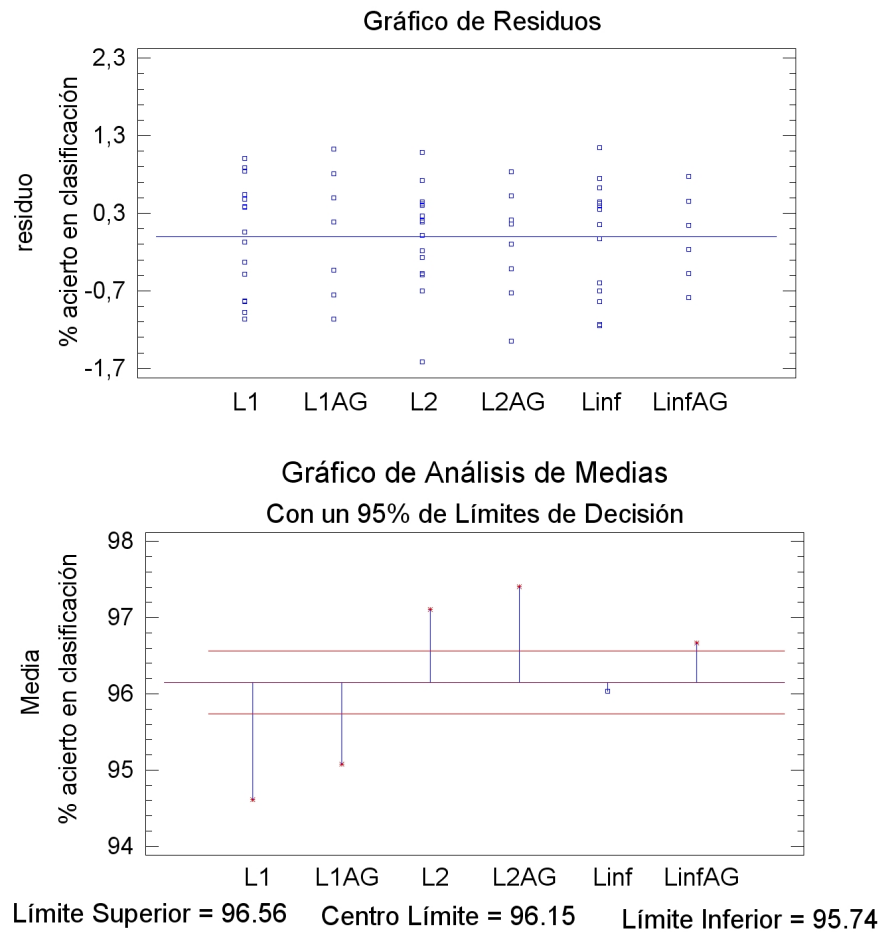


Figura 5.22: Tarea 3.4.1. Análisis de los residuos y la media

comprobar diferentes hipótesis a partir del uso de la prueba t-test. En la Figura 5.21 se incluyen los diagramas de dispersión y de caja asociados a los resultados, donde se puede observar la simetría, si existen patrones de comportamiento, la independencia entre observaciones y tendencias en general. Como se observa los pares de muescas de la mediana se solapan en anchura por lo que no hay diferencia estadística significativa entre ellas para un 95% de nivel de confianza. Además en este caso, no hay puntos externos ni puntos externos lejanos. Por otro lado, la Figura 5.22 incluye la gráfica de residuos y la del análisis de medias de cada una de las alternativas. La primera incluye los residuos de las diferentes normas y muestra que la variabilidad dentro de cada una es aproximadamente la misma. La segunda realiza el análisis de medias para cada una de las tres propuestas junto con la media global y el 95% de límites de decisión. Las muestras que se encuentran fuera de los límites de decisión, L_1 y L_2 , son significativamente diferentes de la media global. Además el análisis de contraste de la varianza mostrado en la Tabla 5.30, muestra que los tres p-valores calculados son superiores a 0.05, por lo que no hay diferencia estadísticamente significativa entre las desviaciones típicas para un nivel de confianza del 95.0%.

Tabla 5.32: Tarea 3.4.1. Análisis aplicando la Tarea 4.2: Test CRM

Contraste	Diferencias	+/- Límites
$L_1 - L_1AG$	-0.4648	0.4719
$L_1 - L_2$	*-2.4966	0.4719
$L_1 - L_2AG$	*-2.7906	0.4719
$L_1 - L_{inf}$	*-1.4286	0.4719
$L_1 - L_{inf}AG$	*-2.0564	0.4719
$L_1AG - L_2$	*-2.0318	0.4719
$L_1AG - L_2AG$	*-2.3258	0.4719
$L_1AG - L_{inf}$	*-0.9638	0.4719
$L_1AG - L_{inf}AG$	*-1.5916	0.4719
$L_2 - L_2AG$	-0.2939	0.4719
$L_2 - L_{inf}$	*1.0681	0.4719
$L_2 - L_{inf}AG$	0.4401	0.4719
$L_2AG - L_{inf}$	*1.3521	0.4719
$L_2AG - L_{inf}AG$	*0.7241	0.4719
$L_{inf} - L_{inf}AG$	*-0.6278	0.4719

Como se cumplen las condiciones para aplicar el método ANOVA en la Tabla 5.31 se muestra el resultado de dicho método aplicado para una de las tres normas analizadas. La varianza del porcentaje de acierto se descompone en dos componentes: uno entre grupos y otro dentro de los grupos. El F-ratio, que en este caso es igual a 44.6508, es el cociente de la estimación entre grupos y la estimación dentro de los grupos. Puesto que el p-valor del test F es inferior a 0.05, hay diferencia estadísticamente significativa entre el porcentaje de acierto para cada norma a un nivel de confianza del 95.0%.

Para determinar las medias que son significativamente diferentes unas de otras, en la Tabla 5.32 se detalla el test CRM de comparación múltiple. El asterisco que se encuentra al lado de 12 de los pares analizados, del total de $\binom{6}{2}=15$ combinaciones, indica que existen diferencias estadísticamente significativas a un nivel de confianza 95,0%. La columna límites define la mitad de la anchura de los intervalos de confianza para cada diferencia de medias entre las dos normas analizadas. En la Tabla 5.33, se identifican cuatro grupos homogéneos según la alineación del símbolo \checkmark en las columnas. Dentro de cada columna, los niveles que tienen \checkmark forman un grupo de medias entre las cuales no hay diferencias estadísticamente significativas. El método utilizado para discernir entre las medias es el procedimiento LSD.

Como se puede observar las propuestas basada en la norma L_2 presentan diferencias significativas con el resto y la media de sus resultados es superior. Entre las dos alternativas, destaca la que emplea el AG para localizar los mejores parámetros de η y μ : L_2AG . Por tanto para la siguiente subtarea se elegirá dicha norma L_2 para calcular la

función de coste asociada a las diferentes heurísticas de algoritmo de aprendizaje.

Tabla 5.33: Tarea 3.4.1. Análisis aplicando la Tarea 4.2: Grupos homogéneos

Función de coste	Media (%)	Grupos homogéneos		
L_1	94.6102	✓		
L_1 -AG	95.0750	✓		
L_{inf}	96.0388		✓	
L_{inf} -AG	96.6667		✓	
L_2	97.1069		✓	✓
L_2 -AG	97.3908			✓

5.2.5.4.3. Tarea 3.4.2: Algoritmos de aprendizaje

Las variantes del algoritmo de aprendizaje pretenden ayudar a localizar el mínimo global sobre la superficie del error del espacio de pesos. Cada una de las alternativas estudiadas, de primer o segundo orden, pretenden optimizar y acelerar dicha búsqueda mediante diferentes aproximaciones, como la forma de actualizar los pesos, la adaptación dinámica de los parámetros o la utilización de las segundas derivadas del error.

La alternativa RBDE propone un proceso de adaptación de η y μ para la búsqueda de la superficie óptima de rendimiento. Los valores se adaptan de acuerdo al valor anterior que han obtenido. Si las actualizaciones de los pesos actual y anterior son del mismo signo, se aumenta la tasa de aprendizaje de forma lineal. El razonamiento empleado es que si los pesos se están moviendo en la misma dirección para disminuir el error, entonces llegarán más rápido con un tamaño de paso grande. Si las actualizaciones tienen signos diferentes, esto es un indicio de que los pesos se han desplazado demasiado lejos sobre la superficie del error. Cuando esto ocurre, la tasa de aprendizaje disminuye geoméricamente para evitar la divergencia. Las ecuaciones asociadas a este proceso adaptativo son la Ecuación 4.32 y la Ecuación 4.33. Los valores empleados para los parámetros de estas ecuaciones han sido los siguientes:

- $k=0.1$, $\beta=0.1$ y $\varphi=0.5$ para la capa oculta.
- $k=0.01$, $\beta=0.1$ y $\varphi=0.5$ para la capa de salida.

Los resultados obtenidos se muestran en la Tabla 5.34. La mejor propuesta es el algoritmo GC, con un rendimiento en precisión y simetría similar al algoritmo LM. Si bien este último es el más rápido a la hora de alcanzar la convergencia, las diferencias en tiempo de aprendizaje, expresado en *epochs*, entre ambos no son excesivas (770 *epochs*).

Tabla 5.34: Tarea 3.4.2. Resultados obtenidos utilizando diferentes algoritmos de aprendizaje (20 repeticiones)

Algoritmo	Acierto mínimo (%)	Acierto promedio (%)	Acierto máximo (%)	Epochs promedio
GR	95.8875	97.4708	98.3875	7710
RBDE	95.8750	97.2438	98.3750	3202
GC	97.4500	98.2417	99.0125	3630
QP	96.2100	97.2959	98.6510	2900
LM	96.8250	98.0751	98.7000	2860
PMC Tarea 3.2	95.6375	97.1665	97.8060	9891
PMC Tarea 3.3	94.8375	97.3875	98.8301	6205

5.2.5.4.4. Medidas de calidad de los resultados

Para asegurar la validez de los resultados anteriores se ha incluido un estudio estadístico que permita comprobar diferentes hipótesis a partir del uso de la prueba t-test. En la Figura 5.23 se incluyen los diagramas de dispersión y de caja para observar la simetría, si existen patrones de comportamiento, la independencia entre observaciones y tendencias en general. Como en el gráfico de cajas los pares de muescas de la mediana se solapan, no hay diferencia estadística significativa entre ellas para un 95 % de nivel de confianza. Además en este caso, no hay puntos externos ni puntos externos lejanos. La Figura 5.24 incluye la gráfica de residuos y la gráfica del análisis de medias de cada uno de los algoritmos. Los residuos son iguales a los valores de porcentaje de acierto menos el valor medio para el grupo del que proceden y muestran que la variabilidad dentro de cada alternativa es aproximadamente la misma. La gráfica realiza el análisis de medias para cada una de las propuestas junto con la media global y el 95 % de límites de decisión. Las muestras que se encuentran fuera de los límites de decisión, GC, LM y RBDE, son significativamente diferentes de la media global. Además el análisis de contraste de la varianza mostrado en la Tabla 5.35, muestra que los tres p-valores calculados son superiores a 0.05, por lo que no hay diferencia estadísticamente significativa entre las desviaciones típicas para un nivel de confianza del 95.0 %.

Tabla 5.35: Tarea 3.4.2. Análisis aplicando la Tarea 4.2: Contraste de la varianza

Contraste	Valor	p-valor
C de Cochran	0.2760	0.2760
Bartlett	1.1081	0.1365
Test de Levene	1.0645	0.3806

Como se cumplen las condiciones para aplicar el método ANOVA en la Tabla 5.36 se muestra el resultado de dicho método aplicado sobre cada uno de los algoritmos.

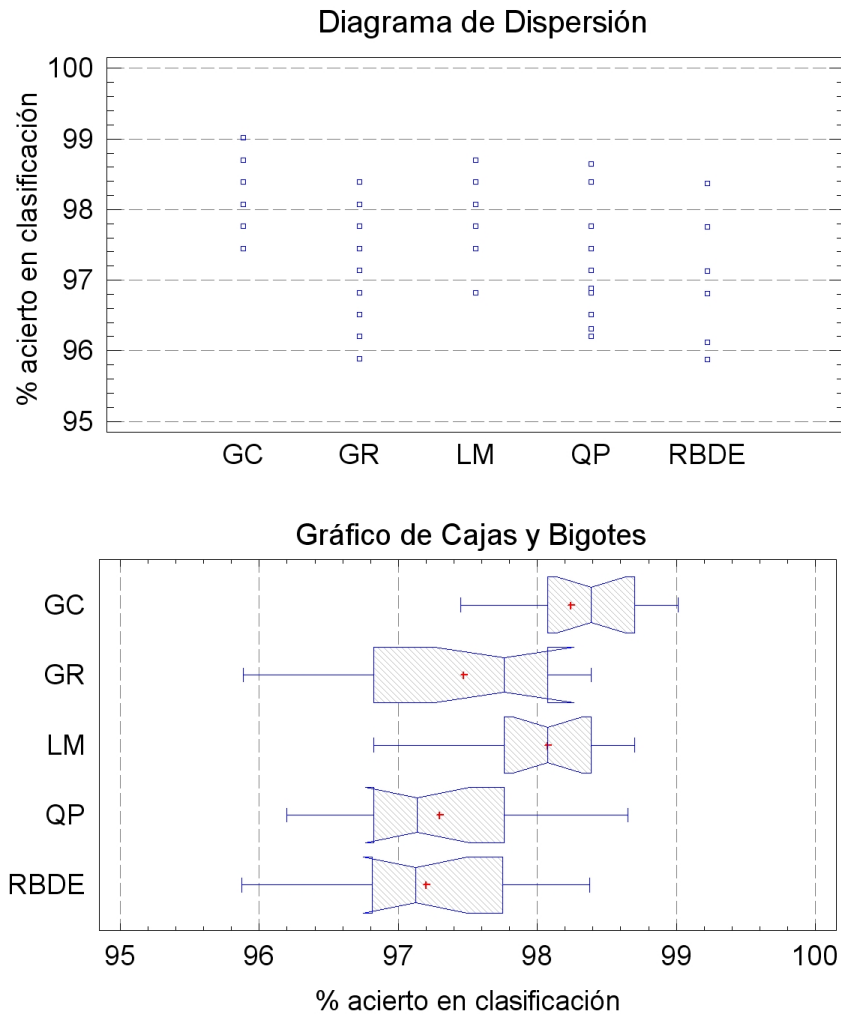


Figura 5.23: Tarea 3.4.2. Diagramas de dispersión y de caja

Tabla 5.36: Tarea 3.4.2. Análisis aplicando la Tarea 4.2: Método ANOVA

Fuente	Sumas de Cuadrados	Grados Libertad	Cuadrado Medio	Cociente-F	P-Valor
Entre grupos (SCE)	13.3569	4	3.33922	7.0578	0.0001
Intra grupos (SCR)	33.1186	70	0.4731		
Total (SCT)	46.4755	74			

La varianza del porcentaje de acierto se descompone en dos componentes: uno entre grupos y otro dentro de los grupos. El F-ratio, que en este caso es igual a 7.0578, es el cociente de la estimación entre grupos y la estimación dentro de los grupos. Puesto que el p-valor del test F es inferior a 0.05, hay diferencia estadísticamente significativa entre

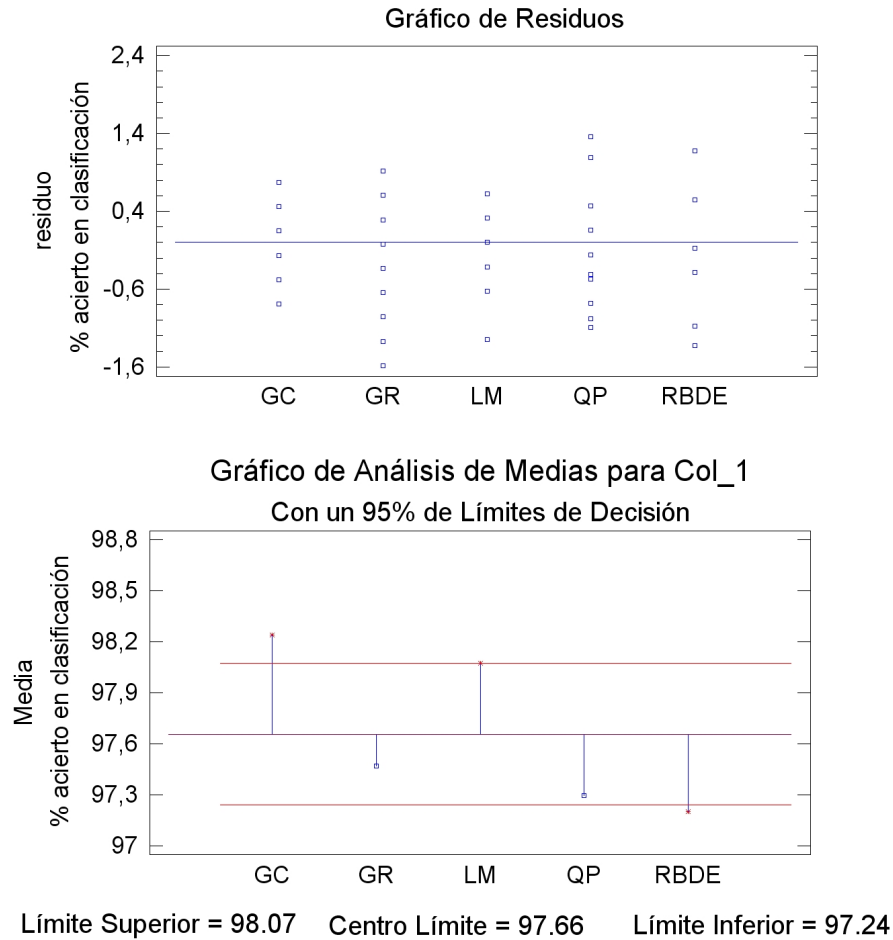


Figura 5.24: Tarea 3.4.2. Análisis de los residuos y la media

la media del porcentaje de acierto de cada algoritmo para un nivel de confianza del 95.0%.

Para determinar las medias que son significativamente diferentes unas de otras, se ha realizado el test CRM que se muestra en la Tabla 5.37. De los 10 pares analizados, $\binom{5}{2}$, seis presentan un asterisco a su izquierda, lo que indica que existen diferencias estadísticamente significativas a un nivel de confianza 95,0%. La columna límites define la mitad de la anchura de los intervalos de confianza para cada diferencia de medias de los diferentes pares de algoritmos. En la Tabla 5.38, ordenados por su media, se identifican dos grupos homogéneos según la alineación del símbolo \checkmark en la columna. Dentro de cada columna, los niveles que tienen \checkmark forman un grupo de medias entre las cuales no hay diferencias estadísticamente significativas. El método utilizado para discernir entre las medias es el procedimiento LSD.

Como se puede observar el algoritmo GC es similar al LM ya que no se encuentran diferencias significativas desde el punto estadístico, pero debido a que los resultados

Tabla 5.37: Tarea 3.4.2. Análisis aplicando la Tarea 4.2: Test CRM

Contraste	Diferencias	+/- Límites
GC - GR	*0.7708	0.5009
GC - LM	0.1666	0.5009
GC - QP	*0.9457	0.5009
GC - RBDE	*1.0416	0.5009
GR - LM	*-0.6041	0.5009
GR - QP	0.1749	0.5009
GR - RBDE	0.2708	0.5009
LM - QP	*0.7791	0.5009
LM - RBDE	*0.875	0.5009
QP - RBDE	0.0959	0.5009

obtenidos por el primero son sensiblemente mejores, la propuesta seleccionada como algoritmo de aprendizaje para este escenario será el primero: algoritmo GC.

Tabla 5.38: Tarea 3.4.2. Análisis aplicando la Tarea 4.2: Grupos homogéneos

Algoritmo	Media (%)	Grupos homogéneos
RBDE	97.2438	✓
QP	97.2959	✓
GRA	97.4708	✓
LM	98.0751	✓
GC	98.2417	✓

5.2.5.5. Comparativa Tarea 3.3 y Tarea 3.4

Para finalizar esta fase se procederá a comparar los mejores resultados obtenidos en las dos últimas tareas, ambas relacionadas con la mejora del algoritmo de entrenamiento del PMC. En este caso, se ha recurrido a un análisis mediante la prueba t-test pareada para determinar la homogeneidad de las distribuciones asociadas a cada una de las alternativas: AG y GC.

A la vista de los resultados obtenidos en la Tabla 5.39 se puede establecer que no existen evidencias de un posible alejamiento de la normalidad, por lo que se incluye la prueba t-test pareada como contraste de homogeneidad entre las poblaciones del error. En este caso se ha utilizado la configuración $20 \times 10cv$ (aplicando la técnica $K_{10}A_{100}$ con 10 conjuntos y 20 repeticiones) y la hipótesis de partida será que la diferencia entre los errores muestrales de ambos algoritmos tiene media cero. Con este contraste se pretende demostrar que existen diferencias significativas entre los errores de cada modo de entrenamiento.

Tabla 5.39: Tarea 3.3 vs.Tarea 3.4. Análisis estadístico (20 repeticiones)

Estadístico	(medido en % acierto)		(medido en % error)
	AG	GC	Diferencias AG-GC
Media	97.3875	98.2417	0.8442
Varianza	2.70033	0.207403	2.9434
Desviación típica	1.64327	0.4554	1.7156
Mínimo	94.8375	97.45	-1.45
Máximo	98.8301	99.0125	3.67
Rango	4.8683	1.5625	5.12
Asimetría tipificada.	-0.3870	-0.4988	0.6432
Curtosis tipificada	-1.112	-0.2964	-0.9535
<i>Epochs</i> promedio	6209	3630	

- Contraste t:

- Hipótesis nula: media = 0.0
- Alternativa: no igual
- Estadístico t = 2.1539
- P-valor = 0.04915
- Se rechaza la hipótesis nula para alpha = 0.05

Este contraste de homogeneidad se basa en la prueba t-test, cuya hipótesis nula es que la diferencia media de AG-GC sea igual a cero, frente a la hipótesis alternativa de que no lo sea. Puesto que el p-valor para este test es inferior a 0.05, se puede rechazar la hipótesis nula para un nivel de confianza del 95.0%. Por tanto, existen evidencias estadísticas que determinan las diferencias significativas entre las dos propuestas para optimizar el algoritmo de aprendizaje del PMC. A la vista de los resultados y del estudio estadístico, la alternativa seleccionada para las Tarea 3.3 y Tarea 3.4 es el algoritmo GC.

5.2.5.6. Sumario

Una vez terminada esta última fase se pueden sacar las siguientes conclusiones:

- A raíz del análisis desarrollado en la Fase 1, el criterio de convergencia utilizado durante toda la metodología ha sido el de parada temprana (#3).
- El modo de entrenamiento seleccionado debido a su comportamiento es el modo *batch*.
- La función de coste que mejor se ajusta al problema del impacto balístico es la que usa la norma L_2 .

Tabla 5.40: Listado de tareas donde se ha utilizado la Fase 4: Medidas de calidad

Tarea	Tarea 4.1	Tarea 4.2		
	AIC & MDL	t-test o Wilcoxon	ANOVA & CRM	Kruskal-Wallis
Análisis clasificadores	✓		✓	
T1.1 Normalización		✓		
T1.3 Maximizar datos				✓
T2.1 Inicialización		✓		
T2.2 Función activación	✓			
T2.3 Arquitectura	✓			
T2.4 Vector entrada	✓			
T3.2 Modos train		✓		
T3.3 η y μ			✓	
T3.4.1 Función coste			✓	
T3.4.2 Algoritmos			✓	
T3.3 vs T3.4		✓		

- El algoritmo de aprendizaje GC permite mejorar el rendimiento del PMC respecto al algoritmo RPE utilizado en las fases y tareas anteriores.
- El valor máximo de acierto conseguido al finalizar esta fase ha sido 98.2417% obtenido con el modo *batch*, algoritmo GC y función de coste L_2 .

5.2.6. Fase 4. Medidas de calidad

Esta última fase la metodología es transversal al resto, por lo que las diferentes técnicas y métodos estadísticos incluidos en este punto han ido utilizándose y apareciendo a lo largo de las tareas analizadas anteriormente. A modo de guía y resumen, en la Tabla 5.40 se muestran las tareas de la metodología donde se han incluido los indicadores de la Tarea 4.1 y las pruebas de significación o contraste de la Tarea 4.2.

5.2.6.1. Tarea 4.1: Indicadores de la teoría de la información

Esta tarea está enfocada a resolver uno de los problemas típicos a la hora de trabajar con RNA, encontrar un criterio general para determinar el orden que ocupa un modelo a la hora de compararlo con otras alternativas. Muchas teorías han sido expuestas para solventar esta cuestión, entre las que destacan los criterios basados en la teoría de la información, AIC y MDL. Ambos indicadores son fundamentales para la extracción de modelos a partir de los datos ya que no es conveniente utilizar sólo el error de entrenamiento como un criterio para optimizar y decidir entre varias RNA.

Gracias a estos parámetros se puede realizar aproximar el mejor modelo, midiendo la bondad del ajuste, mediante dos vías: a través de la topología y los parámetros del PMC

y también por el error de entrenamiento obtenido. Por tanto una de de las aplicaciones más conocidas de estos indicadores es elegir el mejor modelo neuronal en función de la complejidad de su arquitectura [Lefebvre et al., 2007].

Se ha recurrido a esta tarea para analizar diferentes alternativas en varios puntos de la metodología. En concreto se incluye para corroborar las conclusiones obtenidas durante la Fase Previa (cf. §5.2.2) y las Tareas 2.2 (cf. §5.2.4.2), 2.3 (cf. §5.2.4.3) y 2.4 (cf. §5.2.4.4).

5.2.6.2. Tarea 4.2: Test estadístico

Uno de los análisis estadísticos más comunes en la práctica es con probabilidad el utilizado para comparar dos grupos independientes de observaciones con respecto a una variable numérica. La prueba t de Student o t-test permite, basándose en la desviación estándar, determinar si existen diferencias significativas en las variables analizadas. En el caso de tener diferentes evaluaciones sobre un mismo conjunto se denomina prueba t-test pareada. Este tipo de contraste de homogeneidad que analiza las diferencias entre los errores de dos poblaciones, ha sido utilizado en la Tarea 1.1 (cf. §5.2.3.1), la Tarea 3.2 (cf. §5.2.5.2) y por último para comparar las Tareas 3.3 y 3.4 (cf. §5.2.5.5). Gracias al contraste t-test se puede determinar si las muestras del error provienen de la misma población, i.e. si las dos muestras pertenecen a la misma distribución o si existen diferencias significativas entre ambas.

No obstante para poder utilizar el contraste t-test se necesitan que la diferencia entre los errores responda a una distribución normal, en caso de no cumplirse este requisito se ha optado por una alternativa no paramétrica, como el test de Wilcoxon. Su empleo ha sido necesario en la Tarea 2.1 (cf. §5.2.4.1) al detectarse evidencias de un posible alejamiento de la normalidad.

Por su parte, cuando tenemos más de dos conjuntos de mediciones o resultados, el método ANOVA de un factor o simple trata de analizar si dos variables Y (continua, llamada variable respuesta) y F (categórica, llamada factor), son independientes o no. Este método permite conocer si hay relación entre ellas, si hay diferencias significativas en el valor de la primera según el valor que tome la segunda, si el factor influye en la variable respuesta, etc. Un ejemplo puede encontrarse, en la Fase Previa donde la variable Y se define como el porcentaje de acierto, y el factor F como los diferentes clasificadores (cf. §5.2.2). Si se considera que el factor tiene k valores posibles, representado por F_1, \dots, F_t y denominado niveles del factor entonces la hipótesis que se contrasta es:

$$\begin{aligned} H_0 &: \mu_1 = \mu_2 = \dots = \mu_t \\ H_1 &: \text{alguna } \mu_i \text{ es distinta} \end{aligned}$$

donde μ_1, \dots, μ_t representan las medias correspondientes a cada nivel del factor (donde entonces μ_1 es el % medio de acierto para el clasificador F_1). Si se acepta H_0 , se determina que en todos los casos la media de Y es la misma independientemente del valor del factor,

y por tanto que Y y F son independientes (e.g. que el porcentaje de acierto es el mismo siempre, y por lo tanto que el clasificador no influye especialmente en la resolución del problema). La media conjunta de todos los datos se llama media global, μ . En caso de aceptar H_0 , se acepta no sólo la igualdad entre las μ_i , sino también que todas las μ_i son iguales a la media global, μ .

Para contrastar H_0 , se utiliza la noción de variabilidad, que esencialmente coincide con la de dispersión. Para medir la variabilidad de los datos Y_{ij} , donde el primer subíndice i , indica el nivel de factor al que pertenece el dato, y el segundo, j , el orden que ocupa el dato dentro de los recogidos en ese nivel, se utilizará la Suma de Cuadrados Totales (SCT). Esta suma se descompone en dos: Suma de Cuadrados Residual o intra-grupos (SCR), que tiene que ver con la variabilidad dentro de cada nivel de factor, y la Suma de Cuadrados Explicada o entre-grupos (SCE), que tiene que ver con las diferencias entre la media de cada factor y la media global.

Siguiendo esta filosofía el método ANOVA para analizar diferencias significativas entre diferentes variables se ha incluido para la Fase Previa (cf. §5.2.2), la Tarea 3.3 (cf. §5.2.5.3) y las Tareas 3.4.1 y 3.4.2 (cf. §5.2.5.4). Como en el análisis de la varianza en todos los casos ha mostrado diferencias se ha incluido un contraste de rangos múltiples para decidir que medias son significativamente diferentes.

El test CRM permite decidir que medias son significativamente diferentes cuando se comparan por pares las medias para los diferentes niveles del factor o variables. Si distintas variables no son significativamente diferentes (i.e. hay homogeneidad), el test de los rangos múltiples agrupa dichas variables.

Al igual que ocurre con el contraste t-test, para aplicar el método ANOVA es necesario cumplir una serie de criterios detallados en §4.6.2. Cuando éstos no se cumplen se ha recurrido a otras alternativas, no paramétricas, que no utilizan la media como criterio de contraste sino la mediana. En este caso el contraste empleado es el test de Kruskal-Wallis, el cual ha sido incluido en la Tarea 1.3 (cf. §5.2.3.3) para comparar diferentes alternativas, al no cumplirse los requisitos necesarios para emplear el método ANOVA.

5.2.7. Sumario final para el problema de clasificación

Una vez aplicadas todas las fases de la metodología sobre el primer escenario de impacto balístico, se pueden obtener las siguientes conclusiones:

- El mejor porcentaje de acierto promedio obtenido en la fase preliminar fue de 92.6627% para un PMC entrenado con algoritmo RPE y parada temprana (#3).
- En todas las tareas estudiadas el criterio de parada que presenta un mejor comportamiento es #3. Además de obtener los mejores resultados bajo este criterio,

Tabla 5.41: Comparativa de resultados para todas las alternativas de red respecto al PMC optimizado

Clasificador	Acierto promedio (%)	Criterio de parada (<i>epochs</i>)	Diferencia en acierto (%)	Diferencia en <i>epochs</i>
SVM	87.5500	#2 (8927)	-10.6917	+5297
SOFMS	90.0240	#2 (25214)	-8.2177	+21584
RBFN	90.1910	#2 (80321)	-8.0507	+76691
ELN	90.5144	#3 (2983)	-7.7273	-647
RNN	90.5976	#1 (81000)	-7.6441	+77370
PMC base	92.6227	#3 (8868)	-5.6190	+5238
PMC optimizado	98.2417	#3 (3630)	-	-

su utilización permite reducir de forma considerable el tiempo de entrenamiento y previene el sobreaprendizaje.

- Una correcta distribución de los patrones en los conjuntos de *train*, test y validación mejora los resultados que obtienen las configuraciones clásicas. Para el criterio de parada seleccionado, #3, el reparto de patrones será de: 55 %-20 %-25 % (*train-test-validación*).
- De todas las alternativas analizadas para maximizar la información, se ha seleccionado la combinación de la introducción de ruido AWGN y la técnica KFCVE, bajo la configuración $K_{10}A_{100}$.
- La inicialización de los pesos con la técnica *z-score* y la elección del siguiente conjunto de funciones de activación en las capas de entrada-oculta-salida $f_{iden} - f_{tanh} - f_{tanh}$, permite mejorar el rendimiento del PMC.
- La mejor topología de red localizada gracias al uso de un AG ha sido: C_1N_8 , i.e. una sola capa oculta con 8 neuronas.
- El estudio del vector de entrada ha revelado que las características geométricas del proyectil aportan información que puede tomarse como redundante al utilizar de forma conjunta las variables M_b , R_b y L_b . Este apartado es de interés a la hora de realizar futuras simulaciones numéricas ya que puede reducirse el vector de entrada, siempre teniendo en cuenta la diferencia en rendimiento del PMC que esto produciría.
- El modo de entrenamiento aplicado sobre el PMC es el modo *batch*.
- La función de coste que mejor se ajusta al problema del impacto balístico es la que usa la norma L_2 .
- El algoritmo de aprendizaje GC permite mejorar el rendimiento del PMC respecto al algoritmo RPE utilizado en las fases y tareas anteriores.

- A partir de las condiciones anteriores, el mejor porcentaje de acierto promedio obtenido tras aplicar la metodología de optimización ha sido de 98.2417 %.
- Las métricas de calidad introducidas a lo largo de todas las fases de la metodología han permitido asegurar la validez de los resultados, mostrando si existen diferencias significativas entre las diferentes alternativas analizadas en cada momento.
- Al finalizar cada fase se aumenta de forma gradual el porcentaje de acierto del clasificador: 92.6627 - 96.7625 % - 97.1598 % - 98.2417 % (inicial - fase 1 - fase 2 - fase 3)

En la Tabla 5.41 se muestran los resultados obtenidos para cada uno de los clasificadores estudiados y la diferencia obtenida, en porcentaje de acierto y en número de *epochs*, tras la aplicación de la metodología de optimización sobre el PMC optimizado. Gracias a esta investigación la mejora obtenida sobre el PMC base es superior al 5.5 % y además la disminución en el número de *epochs* es considerable, superior a 5000, lo que supone un porcentaje de reducción del 59.0663 %. Con respecto al resto de clasificadores, su aplicación mejora todas las alternativas analizadas en más de un 7.5 % y en cuanto a tiempos de aprendizaje sólo el clasificador ELN reduce el número de ciclos necesarios para obtener la convergencia.

Por último y a modo de resumen gráfico en la Figura 5.25 se muestran las diferentes fases y etapas ganadoras, y por tanto seleccionadas, dentro de las alternativas existentes en cada caso.

5.3. Evaluación del modelo neuronal para el problema de regresión

El apartado anterior ha probado la capacidad de optimización de la metodología, sobre un problema de clasificación, dentro del dominio del impacto balístico. A continuación para asegurar su validez se estudiará su aplicación sobre los dos problemas de regresión detallados en §3.3.2. Gracias a las conclusiones obtenidas con anterioridad se incluirán en cada fase solo aquellas tareas que han tenido un impacto positivo sobre los resultados que obtiene el PMC. Esto permitirá una aplicación más rápida de la metodología de optimización, reduciendo el número de simulaciones a realizar y por tanto el tiempo y el esfuerzo necesario para optimizar un modelo neuronal concreto.

5.3.1. Modelo neuronal base

El problema de regresión analizado se divide a su vez en dos escenarios dedicados a conocer el comportamiento de los objetos una vez comenzado el proceso de impacto.

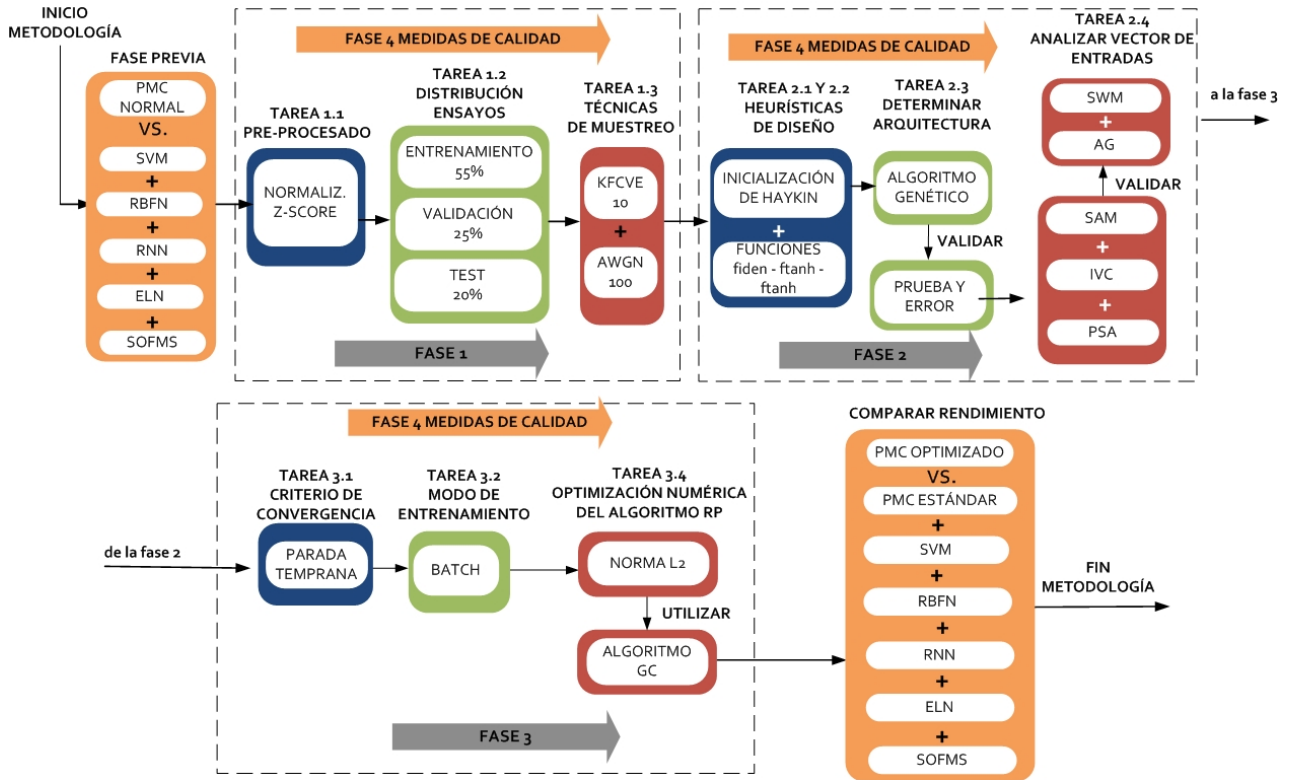


Figura 5.25: Resumen de tareas utilizadas dentro de la metodología

Para el primer escenario, cuando se produce la perforación de la placa, los aspectos de interés son la velocidad residual (V_{res}) y la masa final del proyectil (M_{res}). El segundo escenario se centra en analizar la masa final del proyectil (M_{res}) cuando el proyectil se detiene sin perforar. El reparto de ensayos simulados es menor que en el problema de clasificación lo que penaliza el rendimiento del PMC. Para el primer escenario se dispone de 157 y para el segundo de 63 (cf. §3.5.2).

Las arquitecturas base del PMC para los dos escenarios han sido explicadas en §3.5.1.2. En la Tabla 5.42 se resumen sus características principales dando lugar a la configuración de prueba base, C_0 . Sus características se dividen en tres bloques: patrones (P_0), arquitectura (A_0) y algoritmo (R_0), que serán detallados durante los siguientes apartados.

Gracias a la experiencia obtenida con el problema de clasificación se reducen las simulaciones a realizar en este nuevo problema balístico. En este caso sólo se contemplará el criterio de parada temprana, desechando la parada por *epochs* y MSE al presentar un rendimiento inferior tanto en precisión como en tiempo.

Al igual que en el problema de clasificación, el algoritmo de aprendizaje elegido es la

Característica	Descripción
Reparto patrones por conjuntos (con criterio #3)	33%/33%/33%
Técnica de distribución de patrones	HOCVE
Entradas	$V_{ent} = \{M_b, L_b, R_b, V_b, H_p, T_p, T_b\}$
Salidas	$S_{per} = \{V_{res}, M_{res}\}$ para escenario 1 y $S_{det} = \{M_{res}\}$ para escenario 2
Arquitectura	PMC
Capas ocultas	1
Neuronas ocultas	4 para escenario 1 y 3 para escenario 2
Función de activación (entrada-oculta-salida)	$f_{iden}-f_{tanh}-f_{tanh}$
Algoritmo de entrenamiento	RPE
Parámetros de entrenamiento	$\mu=0.7$, $\eta=1$ en capa oculta y $\mu=0.3$, $\eta=0.4$ en capa salida
Función de coste	MSE simple
Actualización de pesos	Batch
Inicialización de los pesos	Heurística de Haykin
Criterio de convergencia (#3)	Epochs [10000] y Parada temprana

Tabla 5.42: Resumen de características de la red base para regresión

variante RPE debido a sus ventajas y su amplia utilización en toda clase de problemas [Priddy and Keller, 2005]. La reseña #3 indica un entrenamiento con parada temprana por lo que los conjuntos utilizados serán *train*, *test*, y *validación* y la técnica de reparto será HOCVE.

Los resultados obtenidos empleando este PMC base en cada escenario, se detallan en la Tabla 5.43. Para el primer escenario, perforación de la placa, el error MSE es reducido y la factor de correlación positivo y elevado, cercano a uno, especialmente para la variable M_{res} . Esto indica desde un punto de vista matemático que las diferencias entre los valores reales y aproximados por el PMC son reducidos y desde el punto de vista probabilístico que es posible anticipar el valor aproximado de una salida si se conoce el valor de las entradas. Sin embargo, es difícil medir la calidad del PMC si no se tiene una referencia respecto a otras posibles alternativas que permitan evaluar de forma fiable los resultados alcanzados. Por ello en el siguiente apartado se probarán para cada escenario diferentes propuestas de RNA.

5.3.2. Fase Previa: Comparativa de alternativas al PMC

En este apartado se compara la calidad del PMC frente a otras alternativas para dos escenarios de regresión del impacto balístico. En la Tabla 4.1 se han detallado las arquitecturas y principales características de las propuestas. La capa de salida para el

Tabla 5.43: Resultados preliminares obtenidos para el PMC base en cada escenario (20 repeticiones)

Escenario	Variable de salida	r	<i>Epochs</i>
Escenario 1	V_{res}	0.8609	3325
	M_{res}	0.9127	
Escenario 2	M_{res}	0.8936	4731

Tabla 5.44: Fase Previa. Resultados obtenidos para todas las alternativas en el escenario de perforación (20 repeticiones)

Alternativa RNA	<i>Epochs</i>	MSE	NMSE	$r(V_{res})$	$r(M_{res})$	AIC	MDL
SVM	11200	0.0886	0.3871	0.6896	0.7369	98.1310	99.2121
SOFMS	6874	0.0786	0.3393	0.7365	0.8549	85.3278	84.6636
RNN	9253	0.0428	0.1674	0.7971	0.8914	-41.8906	-42.9601
RBFN	9786	0.0423	0.1647	0.8042	0.8981	21.1264	22.1551
ELN	8467	0.0327	0.1191	0.8472	0.9096	-46.7371	-47.8808
PMC	3325	0.0302	0.1067	0.8609	0.9127	-123.0090	-123.6329

primer escenario está formada por dos neuronas, una para cada una de las variables numéricas a aproximar: V_{res} y M_{res} . Para el segundo sólo existe una neurona en dicha capa de salida, asignada a la variable M_{res} . Todos las simulaciones realizadas tienen en común que los datos de entrada han sido tratados con una normalización z-score (cf. Ecuación 4.5) para mantener los valores usados dentro de un rango controlado. El criterio de parada, los porcentajes de reparto y la técnica de distribución de patrones utilizados serán iguales a los especificados en la Tabla 5.42. Por último, los detalles asociados a cada RNA han sido expuestos en §5.2.2.

5.3.2.1. Escenario uno: Perforación de la placa

Para el primer escenario de regresión, perforación de la placa, para cada variable del conjunto $S_{per} = \{V_{res}, M_{res}\}$ se han obtenido los resultados mostrados en la Tabla 5.44. Como se puede observar la alternativa que presenta un menor MSE y NMSE final y un coeficiente r más elevado es el PMC básico analizado en el apartado anterior. Esto indica que las salidas de la red tienen una relación muy estrecha con respecto a los valores reales, al estar el coeficiente r cercano a uno, y que no existen diferencias notables entre los valores calculados por la red y los reales. No obstante, es de destacar que la red ELN obtiene resultados muy próximos, especialmente para la variable M_{res} , por lo que en el siguiente apartado se comprobará, mediante un contraste de homogeneidad, si existen diferencias entre los resultados obtenidos.

5.3.2.1.1. Medidas de calidad de las alternativas

Para comparar el ajuste sobre los datos de cada una de las propuestas analizadas, se

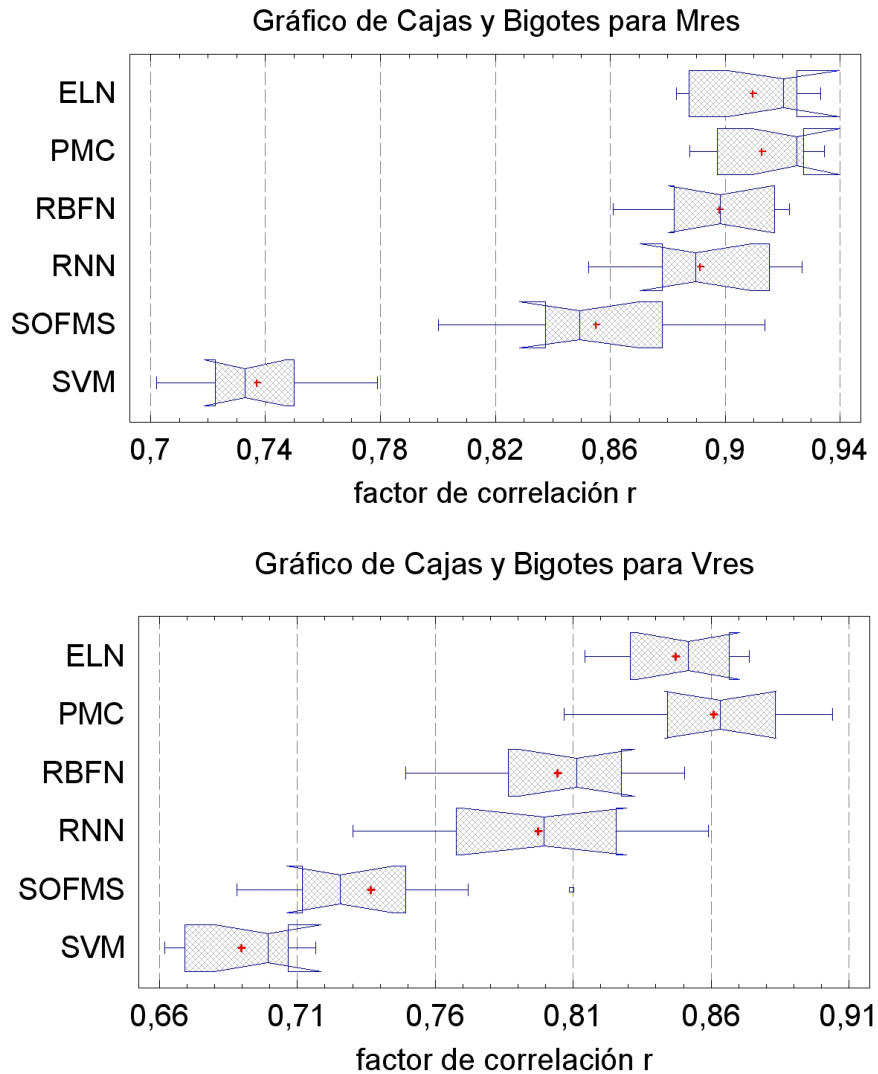


Figura 5.26: Fase Previa. Diagramas de caja para los escenarios de perforación

incluyen los dos criterios estadísticos basados en la teoría de la información descritos en §4.6. En la Tabla 5.44, se muestran los resultados obtenidos para ambos criterios. Como se observa, el modelo neuronal basado en el PMC presenta un mejor comportamiento desde el punto de vista de estos indicadores estadísticos al obtener los valores más bajos.

En la Figura 5.26 se incluyen los diagramas de caja asociados a los resultados para cada variable. Para la variable M_{res} las diferentes cajas nos muestran una asimetría en la distribución de la muestra, a excepción de la red RBFN que es simétrica. Las anchuras de las muescas de la mediana, para un intervalo de confianza del 95 %, son similares para todos los casos. Esto induce a pensar que no hay diferencia estadística significativa entre las medianas para dicho nivel de confianza. Además en el diagrama aparece un punto externo (separado 1.5 veces el rango intercuartílico) en la red SOFMS y ningún punto externo lejano. Para la variable V_{res} las cajas muestran

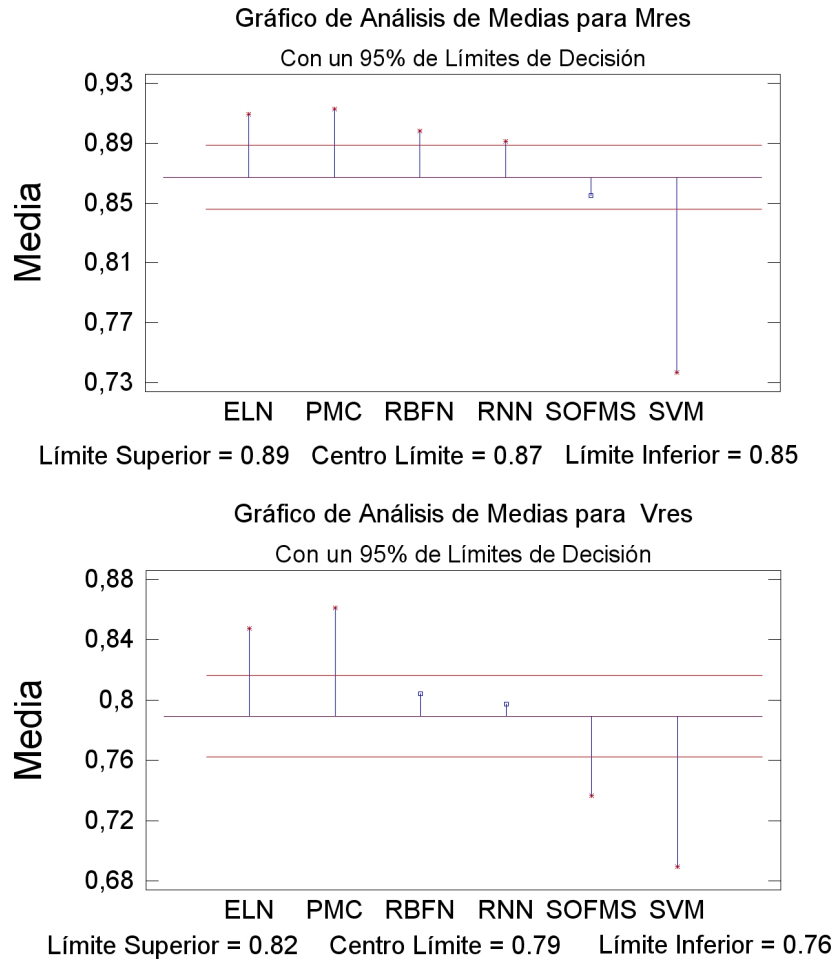


Figura 5.27: Fase Previa. Análisis de la media para los escenarios de perforación

cierta simetría para las redes PMC y RNN, presentando el resto asimetría positiva o negativa. Además vuelve a aparecer un punto externo para la red SOFMS (se trata de la misma muestra de la población) y las muescas de las medianas son similares en tamaño.

La Figura 5.27 incluye las gráficas de análisis de medias para cada una de las seis muestras junto con la media global y el 95 % de límites de decisión. Para la variable M_{res} todas las alternativas, excepto SOFMS, se encuentran fuera de los límites de decisión y son significativamente diferentes de la media global. Por su parte para la variable V_{res} son las redes ELN, PMC, SOFMS y SVM las que están fuera del rango de decisión.

Para poder aplicar el método ANOVA es necesario asegurar que no existen diferencias significativas entre las varianzas de las poblaciones. El análisis de contraste de la varianza para ambas variables, reflejado en la Tabla 5.45, muestra que los tres p-valores calculados son superiores a 0.05, por lo que no hay diferencia estadística reseñable entre las desviaciones típicas para un nivel de confianza del 95.0 %.

Tabla 5.45: Fase Previa. Análisis aplicando la Tarea 4.2: Contraste de la varianza

Salida	Contraste	Valor	p-valor
M_{res}	C de Cochran	0.3668	0.0723
	Bartlett	1.1218	0.3847
	Test de Levene	0.8978	0.4902
V_{res}	C de Cochran	0.2523	0.7895
	Bartlett	1.1183	0.4015
	Test de Levene	0.5864	0.7102

Como se cumplen las condiciones para aplicar el método ANOVA, se ha incluido dicho estudio estadístico paramétrico para determinar una razón entre las diferencias observadas y comprobar diferentes hipótesis a partir del uso de la prueba t-test. Este análisis indicará si existen diferencias significativas entre las diferentes medias y detectará cuales son las diferencias entre los distintos grupos de resultados mediante un test CRM.

Tabla 5.46: Fase Previa. Análisis aplicando la Tarea 4.2: Método ANOVA

Variable	Fuente	Sumas de Cuadrados	Grados Libertad	Cuadrado Medio	Cociente-F	P-Valor
M_{res}	Entre grupos (SCE)	0.2027	5	0.0405	62.0821	0.0000
	Intra grupos (SCR)	0.0313	48	0.0006		
	Total (SCT)	0.2340	53			
V_{res}	Entre grupos (SCE)	0.1933	5	0.0386	36.8641	0.0000
	Intra grupos (SCR)	0.0503	48	0.0011		
	Total (SCT)	0.2437	53			

La Tabla 5.46 muestra el método ANOVA aplicado sobre los resultados obtenidos para cada propuesta. La varianza del porcentaje de acierto se descompone en dos componentes: un componente entre grupos y un componente dentro de los grupos. El cociente de la estimación entre grupos y la estimación dentro de los grupos, F-ratio, es igual a 62.0821 para M_{res} y 36.8641 para V_{res} . Puesto que el p-valor del test F es inferior en los casos a 0.05, hay diferencia estadísticamente significativa entre las medias del coeficiente r para cada alternativa para un nivel de confianza del 95.0 %.

El test CRM aplica un procedimiento de comparación múltiple para determinar las medias que son significativamente diferentes unas de otras. La Tabla 5.47 muestra la

Tabla 5.47: Fase Previa. Análisis aplicando la Tarea 4.2: Test CRM

Contraste	M_{res}		V_{res}	
	Diferencias	+/- Límites	Diferencias	+/- Límites
ELN - PMC	-0.0031	0.0242	-0.01367	0.0306
ELN - RBFN	0.0115	0.0242	*0.0429	0.0306
ELN - RNN	0.0181	0.0242	*0.0501	0.0306
ELN - SOFMS	*0.0546	0.0242	*0.1107	0.0306
ELN - SVM	*0.1726	0.0242	*0.1575	0.0306
PMC - RBFN	0.0147	0.0242	*0.0566	0.0306
PMC - RNN	0.0213	0.0242	*0.0637	0.0306
PMC - SOFMS	*0.0578	0.0242	*0.1244	0.0306
PMC - SVM	*0.1758	0.0242	*0.1712	0.0306
RBFN - RNN	0.0066	0.0242	0.0071	0.0306
RBFN - SOFMS	*0.04311	0.0242	*0.0677	0.0306
RBFN - SVM	*0.1611	0.0242	*0.1146	0.0306
RNN - SOFMS	*0.0364	0.0242	*0.0606	0.0306
RNN - SVM	*0.1544	0.0242	*0.1074	0.0306
SOFMS - SVM	*0.1179	0.0242	*0.0468	0.0306

diferencia estimada entre cada par de medias. Para la variable M_{res} el asterisco que se encuentra al lado de los nueve pares, del total de $\binom{6}{2}=15$ combinaciones, indica que éstos muestran diferencias significativas a un nivel de confianza 95.0%. Por su parte para la variable V_{res} el asterisco aparece al lado de trece pares para indicar las diferencias reseñables desde el punto de vista estadístico. La columna límites define la mitad de la anchura de los intervalos de confianza para cada diferencia de medias de las diferentes propuestas, por lo que dichos intervalos vendrán definidos como *Diferencia* \pm *Límites*. En la Tabla 5.48 según la alineación del símbolo \surd se identifican tres grupos homogéneos para la variable M_{res} y cuatro para la variable V_{res} . Dentro de cada columna, los niveles que tienen \surd forman un grupo de medias entre las cuales no hay diferencias significativas. El método utilizado para discernir entre las medias es el procedimiento de las menores diferencias significativas de Fisher (LSD).

Para la variable M_{res} el test CRM indica que los valores de r para el PMC no son significativamente diferentes con respecto a las alternativas RNN, RBFN y ELN. Esto indica que ambas poblaciones de r provienen de la misma distribución. Sin embargo para la variable V_{res} si que existen diferencias significativas entre el PMC y el resto de propuestas, excepto para la red ELN. No obstante si se tienen en cuenta el resto de criterios analizados se pueden ampliar las deducciones sobre los resultados. Teniendo en cuenta los valores de AIC y MDL la red RBFN puede descartarse debido a los altos valores obtenidos al finalizar las simulaciones. En este sentido las alternativas que mejor representan la muestra de ensayos balísticos, presentan un AIC y MDL más altos, han sido el PMC y las redes ELN y RNN por ese orden. Además con respecto a los errores

MSE y NMSE los valores más bajos los obtienen el PMC y la red ELN. Por último si se tiene en cuenta el estimador de referencia, la media del coeficiente r , el PMC es el que mejor relación obtiene entre los datos reales y los aproximados. Por tanto para este escenario se puede afirmar que estas tanto el PMC como la red ELN son las propuestas más adecuadas, siendo ligeramente superior el PMC.

En los siguientes apartados se procederá a aplicar la metodología de optimización del PMC en este escenario para comprobar posteriormente la mejora obtenida con respecto al PMC original y al resto de alternativas.

Tabla 5.48: Fase Previa. Análisis aplicando la Tarea 4.2: Grupos homogéneos

Alternativa	Media $r(M_{res})$	Grupos homogéneos		Media $r(V_{res})$	Grupos homogéneos	
SVM	0.7369	✓		0.6896	✓	
SOFMS	0.8549		✓	0.7365		✓
RNN	0.8914		✓	0.7971		✓
RBFN	0.8981		✓	0.8042		✓
ELN	0.9096		✓	0.8472		✓
PMC	0.9127		✓	0.8609		✓

5.3.2.2. Escenario dos: Detención del proyectil

Para el segundo escenario de regresión, detención del proyectil dentro de la placa, se han obtenido los resultados mostrados en la Tabla 5.49 para cada variable del conjunto $S_{det} = \{M_{res}\}$. Como se puede observar la alternativa que presenta un mejor MSE y NMSE final y un coeficiente r más elevado es el PMC básico. Esto indica que las salidas de la red tienen una relación muy estrecha con respecto a los valores reales al estar el coeficiente r cercano a uno y que no existen diferencias notables entre los valores calculados por la red y los reales. No obstante es de destacar que la red ELN obtiene resultados muy próximos para la variable M_{res} , por lo que en el siguiente apartado se comprobará mediante un contraste de homogeneidad si existen diferencias entre los resultados obtenidos.

5.3.2.2.1. Medidas de calidad de las alternativas

Para comparar el ajuste sobre los datos de cada una de las propuestas analizadas se incluyen los dos criterios estadísticos basados en la teoría de la información descritos en §4.6. En la Tabla 5.49, se muestran los resultados obtenidos para ambos criterios. Como se observa el modelo neuronal basado en el PMC presenta un mejor comportamiento desde el punto de vista de estos indicadores estadísticos al obtener los valores más bajos.

Tabla 5.49: Fase Previa. Resultados obtenidos para todas las alternativas en el escenario de detención (20 repeticiones)

Alternativa RNA	<i>Epochs</i>	MSE	NMSE	$r(M_{res})$	AIC	MDL
SOFMS	8401	0.3130	0.6635	0.5605	1511.5715	1144.6681
SVM	10125	0.4030	0.7535	0.5695	1361.5715	994.6681
RBFN	13702	0.2088	0.4426	0.7687	598.0663	447.1009
RNN	4423	0.2607	0.5526	0.6844	90.5278	61.8635
ELN	9530	0.1013	0.2148	0.8850	95.6422	61.2450
PMC	4731	0.1009	0.2138	0.8936	25.7882	8.1118

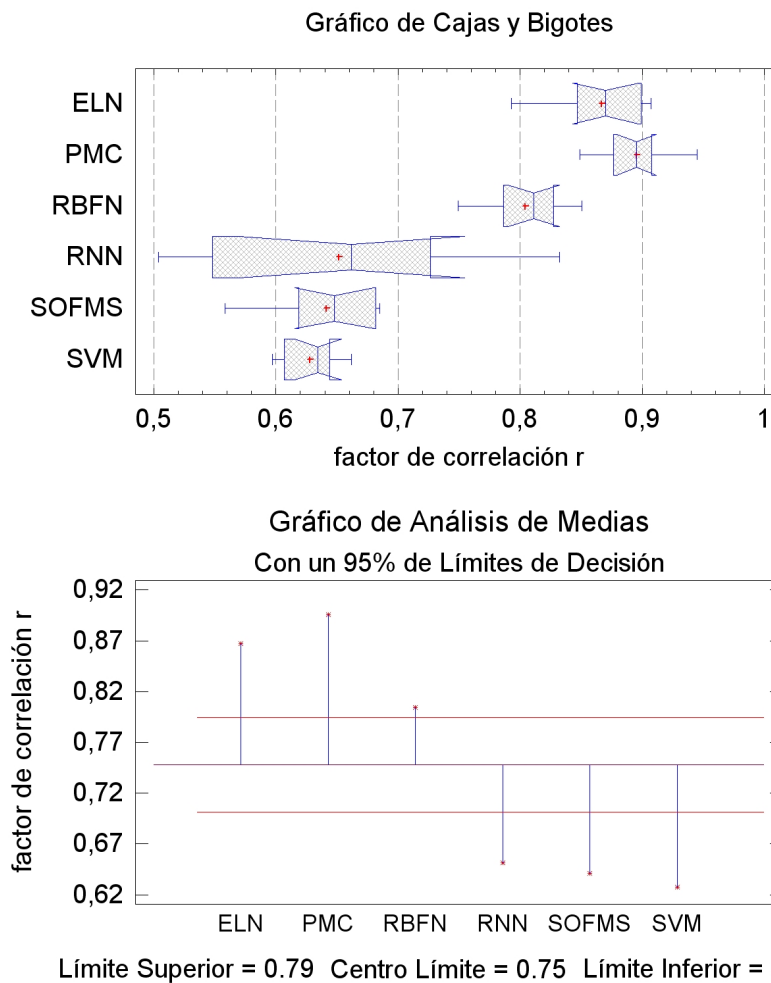


Figura 5.28: Fase Previa. Diagramas de caja y análisis de la media para el escenario de detención

En la Figura 5.28 se incluyen los diagramas de caja asociados a los resultados obtenidos para la media del factor de correlación. Las diferentes cajas muestran una asimetría en la distribución de la muestra, a excepción de la red PMC que es simétrica. En este caso las anchuras de las muescas de la mediana, para un intervalo de confianza del 95 %, no

son similares, especialmente para la red RNN. Esto induce a pensar que existe diferencia estadística significativa entre las medianas para dicho nivel de confianza. Además en el diagrama no aparecen puntos externos ni puntos externos lejanos. Con respecto a la gráfica de análisis de medias, se manifiesta que todas las alternativas se encuentran fuera de los límites de decisión, para un intervalo de confianza del 95 %, y son significativamente diferentes de la media global.

Para poder aplicar el método ANOVA es necesario asegurar que no existen diferencias significativas entre las varianzas de las poblaciones. El análisis de contraste de la varianza, reflejado en la Tabla 5.50, muestra que los tres p-valores calculados son inferiores a 0.05, por lo que hay diferencia estadística reseñable entre las desviaciones típicas para un nivel de confianza del 95.0 %. Esto infringe una de las asunciones importantes que subyacen en el análisis de la varianza y podría invalidar el uso del método ANOVA. En su lugar se ha utilizado el tests no paramétrico de Kruskal-Wallis para comparar las medianas en lugar de las medias.

Tabla 5.50: Fase Previa. Análisis aplicando la Tarea 4.2: Contraste de la varianza

Contraste	Valor	p-valor
C de Cochran	0.6760	$5.6863 \cdot 10^{-7}$
Bartlett	1.8613	0.0000298512
Test de Levene	7.8793	0.000017615

El test de Kruskal-Wallis mostrado en la Tabla 5.51 es un método no paramétrico para probar si un grupo de datos proviene de la misma población. Intuitivamente, es idéntico al ANOVA con los datos reemplazados por categorías. En este caso se comprueba la hipótesis nula de igualdad de las medianas para el porcentaje de acierto para cada una de los seis técnicas. Los datos de todos los niveles primero se combinan y se ordenan de menor a mayor y luego se calcula el rango medio para los datos en cada nivel. Puesto que el p-valor es inferior a 0.05, hay una gran evidencia estadística en contra de que los resultados obtenidos por todas las técnicas sean similares. Para determinar cuáles son las medianas significativamente diferentes entre si se puede observar el gráfico de caja de la Figura 5.28, donde la anchura de las muescas indica el intervalo de confianza aproximado del 95.0 %.

Como ha quedado constatado las redes PMC y ELN son las que mejor comportamiento presentan, teniendo en cuenta la media del factor de correlación, los errores MSE y NMSE. Además los criterios AIC Y MDL ratifican estas conclusiones y el test de Kruskal-Wallis pone de manifiesto que existen diferencias significativas con respecto a la mediana de las distribuciones de la muestra de cada red.

En los siguientes apartados se procederá a aplicar la metodología de optimización del PMC en este escenario, para comprobar tanto la mejora global como la mejora obtenida fase a fase.

Tabla 5.51: Fase Previa. Análisis aplicando la Tarea 4.2: Test de Krustal Wallis

Técnica	Rango Medio
PMC	47.2222
ELN	42.3333
RBFN	32.2222
RNN	16.0003
SOFMS	15.1111
SVM	12.1111
Estadístico = 41.958	p-valor = $6.0068 \cdot 10^{-8}$

Tabla 5.52: Fase 1. Resultados obtenidos por el PMC en cada escenario (20 repeticiones)

Escenario	Variable de salida	r	Epochs
Escenario 1	V_{res}	0.8912	3076
	M_{res}	0.9438	
Escenario 2	M_{res}	0.9098	4328

5.3.3. Fase 1. Tratamiento de los patrones

En primer lugar, se aplicará sobre el PMC base las tareas asociadas a la primera fase que más han afectado al rendimiento del PMC para el problema de clasificación. Las principales características de las técnicas y métodos a ejecutar para cada tarea son las siguientes:

- Tarea 1.1. Normalización *z-score*.
- Tarea 1.2. Distribución de patrones con los porcentajes 55 %-20 %-25 % para los conjuntos de train, test y validación.
- Tarea 1.3. Técnica $K_{10}A_{100}$ para maximizar la información disponible. En este caso se recurre a la alternativa $20 * 10cv$, también denominada *repeated K-fold cross validation*, que indica que se ejecutará 20 veces cada proceso.

La configuración resultante para esta fase se denomina configuración de prueba uno, C_1 , y está dividida en tres bloques: patrones (P_1), arquitectura (A_0) y algoritmo (R_0). Aplicando estas propuestas sobre los diferentes escenarios se obtienen los resultados mostrados en la Tabla 5.52.

Como se puede observar la mejora obtenida es reseñable, aumentando el coeficiente de correlación para todas las variables de cada escenario. El número de *epochs*, para finalizar el entrenamiento, disminuye en todos los casos, lo que reduce el tiempo de entrenamiento. En el siguiente apartado se procederá a aplicar las técnicas y métodos seleccionados para las tareas asociadas a la segunda fase.

Tabla 5.53: Fase 2. Resultados obtenidos por el PMC en cada escenario (20 repeticiones)

Escenario	Variable de salida	r	<i>Epochs</i>
Escenario 1	V_{res}	0.9002	2854
	M_{res}	0.9498	
Escenario 2	M_{res}	0.9203	4091

5.3.4. Fase 2. Arquitectura de la red

Esta fase está orientada a localizar las mejoras arquitecturas posibles para el PMC en cada uno de los problemas. Siguiendo la experiencia obtenida con el problema de clasificación el resumen de técnicas y métodos empleados así como las conclusiones alcanzadas en cada tarea son las siguientes:

- Tarea 2.1. En los problemas de regresión, donde en la capa de salida la respuesta deseada es una función continua de las entrada, lo mejor es utilizar funciones con un rango infinito como f_{bias} , f_{lineal} , f_{iden} (cf. §4.4.2). Siguiendo estas pautas, la configuración de funciones de activación con mejor comportamiento ha sido f_{iden} - f_{lineal} - f_{lineal} para el escenario de perforación y f_{iden} - f_{lineal} - f_{bias} para el escenario de detención.
- Tarea 2.2. La mejor arquitectura localizada mediante el uso del AG ha sido $7-6-2$ para el primer escenario y $7-4-1$ para el segundo. En el caso del segundo escenario, detención, no se cumple la regla expresada por Cover [Cover, 1965] para aproximar el límite superior del número de neuronas ocultas (cf. §3.5.1). Siguiendo la formulación propuesta el rango de neuronas debería ser inferior a cuatro, aunque los mejores resultados se alcanzan precisamente con cuatro elementos dentro de la capa oculta. Esto puede deberse al reducido número de patrones disponibles para este escenario y las características de los datos asociados al problema del impacto balístico. El único inconveniente que presenta esta circunstancia es que complica ligeramente la arquitectura de la red al aumentar el número de parámetros libres. Sin embargo esto no afecta al rendimiento y precisión del modelo, por lo que será la configuración $7-4-1$ la elegida.
- Tarea 2.3. El análisis del vector de entrada ratifica las conclusiones obtenidas para el problema de clasificación. Las variables con una mayor influencia son V_b y H_p , mientras que las que menos afectan al rendimiento del PMC son las características del proyectil: R_b , L_b y M_b .

La configuración resultante para esta fase se denomina configuración de prueba dos, C_2 , y está dividida en tres bloques: patrones (P_1), arquitectura (A_1) y algoritmo (R_0). Tras la aplicación de estas tareas sobre los diferentes escenarios se obtuvieron los resultados mostrados en la Tabla 5.53.

Como se puede observar el coeficiente de correlación ha mejorado para todas las variables de cada escenario. El número de *epochs* para finalizar el entrenamiento también

Tabla 5.54: Fase 3. Resultados obtenidos por el PMC base en cada escenario (20 repeticiones)

Escenario	Variable de salida	r	<i>Epochs</i>
Escenario 1	V_{res}	0.9137	2641
	M_{res}	0.9538	
Escenario 2	M_{res}	0.9314	3927

ha disminuido en todos los casos lo que ha permitido reducir el tiempo de entrenamiento. En el siguiente apartado se procederá a aplicar sobre el modelo neuronal resultante en este punto, las técnicas y métodos correspondientes a las tareas de la última fase de la metodología.

5.3.5. Fase 3. Algoritmo de entrenamiento

El último paso de esta metodología, consiste en aplicar las tareas seleccionadas en el problema de clasificación para la tercera fase, sobre el PMC diseñado en la fase anterior. En este caso la segunda fase aplicada sobre el problema de regresión ha configurado la topología $7 - 6 - 2$ para el escenario primero y $7 - 4 - 1$ para el segundo. En ambos casos el vector de entradas a emplear será el completo: $V_{ent} = \{M_b, L_b, R_b, V_b, H_p, T_p, T_b\}$.

- Tarea 3.1. Se utilizará como criterio de convergencia el criterio de parada temprana (#3).
- Tarea 3.3. El modo de entrenamiento seleccionado será el modo *batch*.
- Tarea 3.3. No se aplicará ya que las propuestas incluidas en esta tarea no mejoran el rendimiento del PMC.
- Tarea 3.4. La función de coste elegida es la norma L_2 y para el algoritmo de aprendizaje la alternativa GC.

La configuración resultante para esta fase se denomina configuración de prueba tres, C_3 y está dividida en tres bloques: patrones (P_1), arquitectura (A_1) y algoritmo (R_1). Aplicando estas tareas sobre los diferentes escenarios se obtienen los resultados mostrados en la Tabla 5.54. En este caso las mejoras obtenidas para los coeficientes de correlación son significativas pero inferiores a las de la primera fase. El número de *epochs* continua disminuyendo pero con un ritmo también inferior.

Tabla 5.55: Sumario y comparativa de los resultados obtenidos en cada escenario

	PMC base	Fase 1	% Var.	Fase 2	% Var.	Fase 3	% Var.	% Var. Total	
Esc. 1	V_{res}	0.8609	0.8912	+3.5196	0.9002	+1.0099	0.9137	+1.4997	+6.1331
	M_{res}	0.9127	0.9438	+3.4075	0.9498	+0.6357	0.9538	+0.4211	+4.5031
	$Epochs$	3325	3076	-7.4887	2854	-7.2172	2641	-7.4632	-20.5714
Esc. 2	M_{res}	0.8936	0.9098	+1.8129	0.9203	+1.1541	0.9314	+1.2061	+4.2301
	$Epochs$	4731	4328	-8.5183	4091	-5.4760	3927	-4.0088	-16.9943

5.3.6. Sumario final para el problema de regresión

En la Tabla 5.55 se muestra un resumen de los valores obtenidos para el factor de correlación y el número de *epochs* completados para cada escenario (filas “Esc.” 1 y “Esc. 2”) y cada variable de salida. Las columnas “% Var.” y “% Var. Total” muestran el porcentaje de variación respecto a la fase anterior y el porcentaje total de variación entre la última fase y el PMC base. Para el primer escenario la mejora media total $r(V_{res})$ y $r(M_{res})$ es del 6.1331 % y 4.5031 % respectivamente. La reducción en el número de *epochs* alcanza el 20.5714 %. Para el segundo escenario la mejora media total para $r(M_{res})$ es del 4.2301 % y la disminución del número de *epochs* llega al 16.9943 %.

Tabla 5.56: Comparativa de resultados para todas las alternativas de red respecto al PMC optimizado

	Alternativa RNA	$Epochs$	Diferencia	$r(V_{res})$	% Var.	$r(M_{res})$	% Var.
Esc.1	SVM	11200	+8559	0.6896	-24.5266	0.7369	-22.7406
	SOFMS	6874	+4233	0.7365	-19.3937	0.8549	-10.3691
	RNN	9253	+6612	0.7971	-12.7613	0.8914	-6.5423
	RBFN	9786	+7145	0.8042	-11.9842	0.8981	-5.8398
	ELN	8467	+5826	0.8472	-7.2781	0.9096	-4.6341
	PMC	3325	+684	0.8609	-5.7787	0.9127	-4.3091
	PMC optimizado	2641	-	0.9137	-	0.9538	-
	Alternativa RNA	$Epochs$	Diferencia	$r(M_{res})$	% Var.		
Esc.2	SVM	10125	+6198	0.6278	-32.5961		
	SOFMS	8401	+4474	0.6416	-31.1184		
	RNN	4423	+496	0.6844	-26.5216		
	RBFN	13702	+9775	0.7687	-17.4709		
	ELN	9530	+5603	0.8850	-4.9776		
	PMC	4731	+804	0.8936	-4.0583		
	PMC optimizado	3927	-	0.9314	-		

Por su parte en la Tabla 5.56 se muestran los resultados obtenidos para cada una de las propuesta de red estudiadas. Además para cada variable de cada escenario se incluye la diferencia obtenida, en porcentaje de variación del coeficiente de correlación y en número de

epochs, de cada propuesta frente al PMC optimizado. Para el primer escenario la red con peor comportamiento frente al PMC optimizado es SVM, con una disminución porcentual sobre $r(V_{res})$ y $r(M_{res})$ del 24.5266 % y 22.7406 % respectivamente. La mejora mínima se obtiene frente a la red ELN con un disminución en porcentajes de 7.2781 % y 4.6341 % para ambas variables. La reducción del número de *epochs* se encuentra dentro del rango [8559-684]. Para el segundo escenario la red SVM vuelve a ser la que peor rendimiento obtiene, con una variación para M_{res} del 32.5961 % frente al PMC optimizado. En el caso contrario, comparando la red ELN, la variación es de 4.9776 %. La reducción del número de *epochs* varía dentro del rango [9775-804].

Por tanto, la evaluación de los resultados obtenidos confirma al PMC optimizado como la mejor opción para el problema de regresión y ratifican la validez de esta metodología para optimizar modelos de PMC sobre diferentes escenarios balísticos.

5.4. Sumario

En el Capítulo 2 se confirmó que los problemas de impacto balístico sobre estructuras son de gran relevancia en el ámbito de la ingeniería estructural y de materiales avanzados. Además, el análisis convencional de los problemas de impacto balístico presenta un elevado coste computacional, debido a su carácter fuertemente no lineal y al elevado número de parámetros que deben ser considerados. Por ello es necesario buscar otras alternativas que permitan mejorar la recreación y resolución de esta clase de problemas. En los Capítulos 3 y 4 se han descrito las soluciones propuestas en esta investigación con este objetivo.

La evaluación y validación realizada en este capítulo ha permitido demostrar el cumplimiento de los objetivos e hipótesis definidas en el Capítulo 1. En primer lugar se ha demostrado, que las técnicas predictivas basadas en aprendizaje automático, pueden aportar soluciones atractivas en problemas de elevada complejidad. Entre las alternativas analizadas el PMC es el que presenta un mejor comportamiento, tanto en el escenario de clasificación como en el de regresión. Además, la capacidad de las RNA para adaptarse a entornos cambiantes permite su utilización en este dominio bajo diversas configuraciones de proyectiles y protecciones, y diferentes escenarios.

Por otro lado, se ha constatado que las RNA, concretamente el PMC, puede utilizarse de forma conjunta con las técnicas clásicas de modelado y resolución, como las simulaciones numéricas, para resolver problemas como el planteado.

Por último, la optimización de factores relativos a la arquitectura de la red, algoritmo de entrenamiento y datos disponibles, beneficia la capacidad de generalización de los modelos neuronales creados y por tanto mejora los resultados obtenidos, tanto en tiempo de ejecución como en precisión.

Capítulo 6

Conclusions and Future Research

In order to satisfy the requirements for the European doctorate mention, this chapter includes the English translation of the conclusions of this phd thesis.

In this chapter the final research conclusions are presented. Together with an overview of the main contributions of this thesis, a set of future trends are proposed in order to continue the current research. Finally, at the end of the chapter appears the list of contributions to congresses and journal papers related to this thesis.

As starting point for this work, the aim has been to approach the reader to the domain in which this doctoral thesis focuses on. As proved, the physical processes involved in the process of penetration of a material due to the impact of a projectile are extremely complex. In addition, traditional approaches have been described, used by engineers and scientists to analyze and solve the problems of ballistic impact. Each of these approaches has been detailed, showing its profits, disadvantages and complications facing different situations.

Next the usefulness of the ANN to solve engineering problems has been demonstrated, exposing the most important research conducted in the field of industrial engineering and in the field of mechanics of materials and structures. On the other hand, attention has been paid to the works carried out to date with ANN in the area of impact of solids, emphasizing their use as monitoring systems for materials fatigue, location of impacts, damage analysis, and so on. In this sense, it has also been discussed its application in one of the most interesting environments at present, the *Crashworthiness* type of impacts, where it is necessary to design structures that permit to absorb the energy produced in such collisions to safeguard the structures and occupants inside.

The results obtained by researchers in these studies have confirmed ANN systems as an alternative to be taken into account opposed to classical methodologies used in solving such problems: empirical experimentation, numerical simulation and analytical modeling. Finally, it is necessary to highlight that the lack of investigations for the resolution, through ANN systems, of the problem of projectile's impact on protections in the high speed range has been

proved.

As for the need to conduct this research, their basis come from the GR/MAT/0507/2004 research project subsidized by the Comunidad of Madrid. Thanks to the work carried out on it, an article [Garcia-Crespo et al., 2006] and a doctoral thesis [Fernandez, 2007] were published in order to announce to the scientific community the findings achieved. This doctoral thesis tackled the problem of the ballistic impact from a different point of view from the one followed in this research. Its main objective was to obtain the modeling of the materials that make up the objects of the impact through its constitutive equations, its numerical simulation and its subsequent comparison with ANN. The simulations and experiments performed were intended to make a first approach to solving such problems. The results obtained demonstrated that the ANN are an alternative to traditional techniques.

Therefore, this doctoral thesis can be considered as a satisfactory first approach which allows certifying the usefulness of the ANN within the domain of ballistic impact. However, it was not taken into account, within the targets set out, the use of different topologies, configurations or network settings. Thus the validation carried out did not include a study aimed at optimizing the performance and results, together with the minimization of the information needed to train the network, key tasks within the engineering design. Moreover, the work done by Fernandez et al. [Fernandez et al., 2008] demonstrates the validity of the ANN to solve problems related to the theory of structures and continuum mechanics. This fact reinforces the benefits of extrapolating to other domains the use of these computational structures and their optimization in order to improve the results reached.

Based on what has been put forward, one of the first targets set for this doctoral thesis was to ratify and extend the conclusions reached previously in the works mentioned. On the one hand, the definition of new boundary conditions that allow extrapolating the results to the new types of projectiles and protections has been set. Furthermore, in order to ascertain whether the precedent results could be improved and to reduce time associated with the resolution of the simulations, batteries of experiments have been designed to optimize those factors that influence capacity of generalization of the network models.

To meet the objectives and prove the original hypothesis, this research has been defined as an integral framework in which different components work together for the solution and optimization of the engineering problem of impact of solids. The numerical simulation tools and the experimentation of ballistic tests have been the first components used, being two crucial elements as they serve as a link with the Multilayer Perceptron (MLP) models designed.

The numerical simulation, has allowed from the analysis of the physical phenomenon of the problem, and based on a series of mathematical models, to recreate by means of numerical simulations, the material's characteristics and the behavior of the objects involved. However, in order to reliably transfer characteristics of the phenomenon of ballistic impact inside a numerical simulation tool has been necessary to conduct a series of experimental tests. These empirical tests are the second component of the integrated framework and allowed to verify the real accuracy of numerical simulation models by comparing the results obtained. As a result of joint use of laboratory testing and numerical simulation techniques to recreate the

phenomenon of impact between solids a series of tests was obtained, which have been used to test the ability of the model MLP and other alternatives based on the theory of machine learning. The available data include parameters relating to solid geometry involved, e.g. radius and length of the projectile, thickness of the metal armour, the type of material used in both the projectile and in the protection, and finally the velocity of impact. To perform the modeling and computer simulation has been used the commercial software ABAQUS/Explicit v6.4.1, an explicit finite element code, widely used in the simulation of dynamic problems with a high degree of nonlinearity.

In summary, thanks to its use a series of missile tests aimed to train and validate the MLP have been simulated. Empirical tests conducted in the laboratory have been useful for checking the plausibility of the numerical simulation models with reality.

The third component corresponds to the techniques of Artificial Intelligence (AI) based on ANN, more specifically the MLP architecture. Its main objective has been to replace the former components to recreate the impact process of solids at high speed. These ANN models have been used to solve three of the most representative scenarios, i.e. to know the impact's result and to study the behavior of objects, proving its validity as an alternative solution to traditional techniques.

AI techniques, where ANN are located, allow an induction of the knowledge inherent to the information supplied to it. Therefore, once learning is carried out, an ANN model will be able to resolve with high precision those new situations that adjust to the conditions on which it has learned. To ensure the reliability of results, all neural models will be validated for different geometries of both projective and protective panel, as well as different velocities and materials. Besides the afore-mentioned capacity of prediction of these techniques, the ANN present, compared with codes of finite elements or finite differences, a fast response when it comes to resolving new impact tests. While the simulation of an impact problem can take several hours of execution, the neural network, once the learning process is performed, allows obtaining the result in a split second. However, the definition of PMC models needed to solve the scenarios raised is the initial component of the integrated framework designed, being necessary to define and implement a methodology that allows optimizing its operation. Thus from these models, the Chapter 4 defines the necessary steps in order to improve the capacity of generalization and optimize the results, both in computation time and accuracy.

The optimization methodology designed is based on a series of activities and tasks aimed at fulfilling the targets and hypothesis outlined at the beginning of this doctoral thesis. As has been found during the analysis of the state of the art, there are a variety of alternatives to parameterize and configure a neural network model with MLP topology and BP algorithm. The literature reviewed has shown that in many cases there are no general rules to use in any domain, which makes the process to find the most suitable solution for each case long and tedious.

To homogenize the large number of existing alternatives and to simplify its implementation on each of the scenarios, these alternatives have been grouped into three large blocks or phases: patterns, architecture and algorithm. Therefore, based on the premise $model = patterns + architecture + algorithm$, on each term of the equation were applied those

techniques and methods that best suit the problem addressed in this research and that have demonstrated superior performance in similar engineering problems. In order to implement all these variants in a reasonable period of time and to ensure that the best possible neural model depending on each domain's characteristics will be obtained, it is imperative a methodology to guide the entire process of optimization and to allow using the different possibilities in a reasonable way. Moreover, a set of statistical controls has also been included to ensure quality and validity of neural models designed. The methodology designed aims to identify the most important features of the objects that take part in a ballistic impact for the learning and subsequent validation of the neural models created. Thanks to this, it will allow detecting those factors that may affect the least to the modeling of the problem and therefore are subject to not being a part of the proposed network architectures. This will reduce in part, the complexity associated with traditional techniques of generation of ballistic impact by reducing the number of variables to recreate. In connection with this, and to get to know the minimum number of examples needed to maintain the results in classification and regression, experiments will be conducted with various configurations of data for training, test and cross-validation.

For all this, this integrated framework has permitted to define a global work structure that enables engineers and scientists to come to know the behavior of future new projectiles and protections without the need of using classic techniques: experimentation, analytical modeling and numerical simulation. For this purpose, the definition of different situations concerning the impact of solids and its treatment from a multidisciplinary point of view has been established by means of techniques such as testing, numerical simulation and diverse alternatives based on machine learning. The evaluation and validation performed brings the following conclusions regarding the objectives and hypotheses defined at the beginning of the work:

- The viability of the MLP to solve different ballistic scenarios has been demonstrated. Thanks to its use, simulation and modeling tools can be used as a complement to the MLP to improve the design of protection systems.
- The use of an optimized MLP allows predicting both the result of an impact and the behavior of the objects that take part with a high percentage of success. To confirm these conclusions, the MLP has been trained and tested under various combinations of geometries and materials for the projectile and protection, and different impact velocities.
- The optimized MLP allows approaching with 98.2417% of accuracy the result of the ballistic impact between a projectile and a protection. For regression problems the correlation coefficient for the first scenario (perforation) is 0.9137 for V_{res} and 0.9538 for M_{res} . For the second scenario of regression (detention or arrest) the values achieved are 0.9314 for M_{res} .
- The percentage improvement for the classification problem between the basis MLP and the optimized MLP is 5.5790%. For regression problems, in the scenario of perforation, the improvement on the correlation coefficient is 6.1331% for V_{res} and 4.5031% for M_{res} . For the scenario of arrest, the improvement is 4.2301% for M_{res} on the correlation coefficient.
- The comparison made with other proposals of learning machines has shown that the MLP is the best alternative for the different ballistic scenarios raised. For the classification problem the improvement that the optimized MLP obtains regarding the best alternative, the RNN classifier, is 7.7273%. For regression problems in the first perforation scenario

the improvement obtained by the optimized MLP regarding the best proposed alternative, ELN network, is 7.2781 % for V_{res} and 4.6341 % for M_{res} respectively. On the other hand, in the arrest scenario the improvement of the optimized MLP is 4.9776 % for M_{res} regarding the ELN network.

- The advantage of using a MLP to resolve the different scenarios is that the results are obtained in real time. The computational cost is therefore much lower for both the traditional techniques of simulation and modeling. Learning time has also been minimized thanks to the methodology, obtaining a percentage of reduction regarding the MLP basis for the number of *epochs* of 59.0663 % 20.5714 % and 16.9943 % for the scenarios of classification, regression with perforation and regression with arrest.
- The maximization of information through the $K_{10}A_{100}$ technique allows using, more efficiently, the MLP based solutions in domains with limited information.
- The techniques proposed to detect the influence of the variables that describe the problem allow selecting only those that provide remarkable information to the learning of the MLP. This reduces the complexity associated with the simulation for the development of new ballistic tests by numerical simulation and reduces the complexity of the architecture of the MLP what entails less effort for its training.
- Quality measures employed throughout the entire methodology ensure the validity of the results from a statistical point of view by reducing the appearance of experimental errors or the appearance of any randomness.
- The conclusions obtained by using in the first place AI-based solutions and in the second place the implementation of an optimization methodology, can be extrapolated to other areas as the design of structures subject to low-speed impacts or the passive safety of vehicles.

In short, thanks to this research, new ballistic tests can be recreated in the future in a reliable way. This will facilitate the further design of protections and the study of the consequences associated with the introduction of different configurations of projectiles and protective panels with different characteristics. Furthermore, the systematization of the different methods and techniques existing when shaping and parameterizing an ANN allows the application of a rapid method of optimization on a neural network model based on a MLP topology and BP algorithm.

6.1. Future Research

The future research lines of this doctoral thesis can be divided into two categories. Being a multidisciplinary work, the first one would be related to the theory associated with ballistic impact and its numerical simulation. The second one would refer to techniques related to the AI and the selected machine learning, the MLP.

For the first category, because the ANN have proved to be a valid tool for the analysis of the behavior to impact on metallic protections, its use may be extended to the study of the behavior to the impact of materials made up of carbon fiber/epoxy matrix. In these situations

it is interesting to make a prediction of the damage's extent, to know the residual kinetic energy or approximate the DOP when the projectile does not perforate the protection.

On the other hand, this work only gathers impact situations without incident angle, i.e. oblique impacts are not addressed. While the simulation of this kind of impacts is more complex the appearance of an incident angle different from zero is actually more likely. Thus a future proposal would be the introduction of the obliquity as input variable in the problem studied for its simulation and subsequent treatment with an optimized MLP.

With regard to the techniques employed to optimize if taken into account the fact that the one that has had the greatest contribution on the performance of the MLP has been the introduction of noise, it would be interesting to investigate the influence that different values of variance have for the AWGN technique. The Gaussian additive noise follows a $N(0, \sigma^2)$ and several researchers have shown that the variance is one of the most sensitive aspects to refine the performance of an ANN [Koistinen and Holmstrom, 1991, Holmstrom and Koistinen, 1992, Karystinos and Pados, 2000].

The conclusions obtained by using in the first place AI-based solutions and in the second place the implementation of an optimization methodology, can be extrapolated to other areas as the design of structures subject to low-speed impacts or the passive safety of vehicles.

Finally, a line of work relevant to the future is to investigate the possibility to extend or adapt this methodology on various alternatives based on learning machines. The aim would be to standardize and systematize all existing alternatives when it comes to optimizing this type of solutions.

6.2. Publications related with this doctoral thesis

As a consequence of this investigation, the following publications have been conducted in different scientific areas:

- Publications in journals with impact factor.
 - Gonzalez-Carrasco, I., Garcia-Crespo, A., Ruiz-Mezcua, B., & Lopez-Cuadrado, J. (2009) Dealing with limited data in ballistic impact scenarios: An empirical comparison of different neural network approaches (in press). Applied Intelligence.

This article describes a part of the optimization methodology developed and includes a comparative of the MLP performance, for the classification scenario, regarding other machine learning alternatives.

- Book Chapters.
 - Garcia-Crespo, A., Ruiz-Mezcua, B., Gonzalez-Carrasco, I. & Lopez-Cuadrado, J. (2009) Multilayer Perceptron Training Optimization for High Speed Impacts

Classification. *Advances in Electrical Engineering and Computational Science* (Vol. 39, pp. 377-388). Springer.

This article demonstrates the possibility to optimize the behavior of a MLP, by implementing a series of heuristics aimed at studying the input vector that defines the neural model.

- Publications in international conferences.
 - Gonzalez-Carrasco, I., Garcia-Crespo, A., Ruiz-Mezcua, B., & Lopez-Cuadrado, J. (2008). Network Application for High Speed Impacts Classification. Paper presented at the World Congress on Engineering (WCE 2008), London, United Kingdom.

In this congress was shown the validity of the ANN to solve the problem of ballistic impact, by replacing or complementing the classical techniques of simulation and modeling.

Capítulo 7

Conclusiones y Trabajos Futuros

A fin de cumplir con los requisitos para la obtención de la mención europea del doctorado, este capítulo sobre las conclusiones y trabajos futuros fue escrito originalmente en inglés. A continuación se incluye la traducción del mismo a la lengua española.

En este capítulo se presentan las conclusiones finales de la investigación expuesta en la presente tesis doctoral. Junto con el repaso de las principales aportaciones se proponen un conjunto de futuras líneas de investigación como continuación de la misma. Para finalizar se enumeran las publicaciones efectuadas a raíz del trabajo realizado en diferentes ámbitos científicos.

Como punto de partida para este trabajo se ha querido aproximar al lector al dominio en el que se centra la presente tesis doctoral: el impacto balístico. Como se ha podido comprobar, los procesos físicos involucrados en el proceso de penetración de un material debido al impacto de un proyectil son extremadamente complejos. Además se han descrito las aproximaciones tradicionales, utilizadas por ingenieros y científicos, para analizar y resolver los problemas de impacto balístico. Cada una de ellas ha sido analizada de forma pormenorizada, mostrando su utilidad, ventajas y desventajas, así como su respuesta ante diferentes situaciones.

A continuación se ha demostrado la utilidad de las RNA para resolver problemas ingenieriles, exponiendo las investigaciones más importantes llevadas a cabo en el ámbito de la ingeniería industrial y de la mecánica de materiales y estructuras. Asimismo se han examinado los trabajos realizados hasta la fecha con RNA en el área del impacto de sólidos, destacando su utilización como sistemas de monitorización para la fatiga de los materiales, localización de los impactos, análisis de daños, etc. En este sentido, también se ha analizado su aplicación en uno de los entornos de más interés en la actualidad, los choques del tipo *Crashworthiness*, donde es necesario diseñar estructuras que permitan absorber la energía que se produce en este tipo de choques para salvaguardar las estructuras y ocupantes de su interior.

Los resultados obtenidos por los investigadores en estos estudios han confirmado a los sistemas de RNA como una alternativa, a tener en cuenta, frente a las metodologías clásicas

empleadas en la resolución de este tipo de problemas: experimentación empírica, simulación numérica y modelado analítico. Por último, destacar que ha quedado demostrada la falta de investigaciones para la resolución, mediante sistemas de RNA, del problema de impactos de proyectiles sobre protecciones en el rango de la alta velocidad.

Como ha quedado patente, las bases de esta investigación y su necesidad provienen del proyecto de investigación subvencionado por la Comunidad de Madrid GR/MAT/0507/2004. Gracias al trabajo realizado en el mismo, se publicó un artículo [García-Crespo et al., 2006] y una tesis doctoral [Fernández, 2007] para dar a conocer a la comunidad científica las conclusiones obtenidas. Gracias al trabajo realizado en estos trabajos, se consiguió obtener la modelización inicial de los materiales que componen los objetos del impacto a través de sus ecuaciones constitutivas, así como su posterior simulación numérica y una primera comparación con las RNA. Las simulaciones y experimentos desarrollados en estos trabajos, permitieron demostrar que las RNA son una alternativa interesante respecto a las técnicas tradicionales. Por otro lado, el artículo publicado por Fernández et al. [Fernández et al., 2008] demuestra la validez de las RNA para resolver problemas, de similar dificultad física y complejidad computacional, derivados de la teoría de estructuras y medios continuos. Este hecho refuerza las ventajas de extrapolar a otros dominios el uso de estas estructuras computacionales y su optimización con el objetivo de mejorar los resultados obtenidos.

No obstante, dentro de los objetivos fijados para los trabajos anteriores no se tuvo en cuenta, al estar fuera del alcance de la investigación, la utilización de diferentes topologías, configuraciones o parámetros de red. La validación realizada no incluía un estudio destinado a la optimización del rendimiento y los resultados, ni tampoco abordaba la minimización de la información necesaria para entrenar la red. Todas estas tareas se consideran fundamentales dentro del diseño y de la optimización ingenieril, ya que la función clásica de los ingenieros es diseñar nuevos sistemas, más eficientes y con menor coste, así como idear proyectos y procedimientos para mejorar las operaciones de los sistemas ya existentes [Reklaitis et al., 1983].

Tomando como base lo expuesto, uno de los primeros objetivos fijados para la presente tesis doctoral fue ratificar y ampliar las conclusiones obtenidas con anterioridad en los trabajos mencionados. Por un lado, se han definido nuevas condiciones de contorno que permiten extrapolar los resultados a nuevos tipos de proyectiles y protecciones. Además, con el fin de conocer si los resultados precedentes son mejorables y para reducir los tiempos asociados a la resolución de las simulaciones, se han diseñado baterías de experimentos destinadas a optimizar aquellos factores que influyen sobre la capacidad de generalización de los modelos de red.

Para cubrir los objetivos y demostrar las hipótesis de partida, esta investigación ha sido definida como una estructura integral de trabajo, donde colaboran diferentes componentes para la resolución y optimización del problema ingenieril del impacto de sólidos: la simulación numérica, la experimentación y las técnicas de IA.

El primero de estos componentes, la simulación numérica, ha permitido, a partir del análisis del fenómeno físico del problema y basándose en una serie de modelos matemáticos, recrear mediante simulaciones en ordenadores las características de los materiales y el comportamiento

de los objetos que intervienen. Sin embargo, para poder trasladar fehacientemente las características del fenómeno del impacto balístico, a una herramienta de simulación numérica, ha sido necesario realizar una serie de ensayos experimentales en laboratorio. Estas pruebas empíricas forman el segundo componente del marco integral, y han permitido verificar la precisión real de los modelos de simulación numérica mediante la comparación de los resultados obtenidos. Como resultado de la utilización conjunta de las pruebas en laboratorio y de las técnicas de simulación numérica, para recrear el fenómeno del impacto entre sólidos, se ha obtenido un conjunto de ensayos balísticos, los cuales han sido empleados para probar la capacidad de los modelos de PMC y otras alternativas basadas en la teoría de la máquina de aprendizaje. Los datos disponibles incluyen parámetros referentes a la geometría de los sólidos involucrados, e.g. radio y longitud del proyectil, espesor de la placa metálica, al tipo de material utilizado, tanto en el proyectil como en la protección, y por último la velocidad del impacto. Para realizar el modelado y la simulación numérica computacional se ha recurrido al software comercial ABAQUS/Explicit v6.4.1, un código de elementos finitos explícito, ampliamente empleado en la simulación de problemas dinámicos con un alto grado de no linealidad.

En resumen, gracias al empleo de la simulación numérica se han obtenido una serie de ensayos balísticos encaminados a entrenar y validar los PMC. Las pruebas empíricas realizadas en laboratorio han servido para comprobar la verosimilitud de los modelos de simulación numérica con la realidad.

El tercer componente se corresponde con las técnicas de IA basadas en RNA, las cuales se han empleado para sustituir a los componentes anteriores y recrear el proceso de impacto de sólidos a alta velocidad. Estos modelos neuronales han sido utilizados para resolver los escenarios más representativos y poder conocer el resultado del impacto y estudiar el comportamiento de los objetos, demostrando de esta manera su validez como alternativa a las técnicas tradicionales. Estas técnicas permiten realizar una inducción del conocimiento inherente a la información que se le suministra. Por tanto, una vez realizado el aprendizaje, un modelo de RNA es capaz de resolver con alta precisión aquellas nuevas situaciones que se ajusten a las condiciones sobre las que ha aprendido. Para garantizar la fiabilidad de los resultados, todos los modelos neuronales han sido validados para distintas geometrías de proyectil y de panel protector, así como distintas velocidades y materiales. Además de la comentada capacidad predictiva de estas técnicas, las RNA presentan, frente a los códigos de elementos finitos o diferencias finitas, una rápida respuesta a la hora de resolver nuevos ensayos de impacto. Mientras que la simulación de un problema de impacto puede conllevar varias horas de ejecución, la red neuronal, una vez realizado el proceso de aprendizaje, permite obtener el resultado en una fracción de segundo. La definición de un modelo neuronal para cada escenario ha sido el punto de partida siendo necesario definir y aplicar una metodología que permita optimizar su funcionamiento. Por ello, a partir de dichos modelos, en el Capítulo 4 se han definido los pasos necesarios con el fin de mejorar la capacidad de generalización y optimizar los resultados, tanto en tiempo de cómputo como en precisión.

La metodología de optimización planteada está basada en una serie de actividades y tareas, orientadas a cumplir los objetivos e hipótesis expuestas al comienzo de esta tesis doctoral. Como ha quedado constatado durante el análisis del estado del arte, existen una gran variedad de alternativas diferentes para parametrizar y configurar un modelo neuronal sobre la topología

PMC y algoritmo RP. La literatura analizada ha mostrado que en muchos casos no existen reglas de uso general aplicables a cualquier dominio y problema, lo que hace que el proceso sea largo y tedioso hasta encontrar la solución más apta para cada caso. Por ello, para homogeneizar el gran número de alternativas existentes y simplificar su aplicación sobre cada uno de los escenarios, éstas se han agrupado en tres grandes bloques o fases: patrones, arquitectura y algoritmo. Por tanto, partiendo de la premisa $\text{modelo} = \text{arquitectura} + \text{patrones} + \text{algoritmo}$ se han aplicado sobre cada término de la ecuación aquellas técnicas y métodos que mejor se adaptan al problema tratado en esta investigación y que han demostrado un mejor comportamiento en problemas ingenieriles similares.

Para poder aplicar todas las variantes existentes en un periodo de tiempo razonable y asegurar que, al finalizar el proceso, se obtendrá el mejor modelo neuronal posible en función de las características de cada dominio, es perentoria una metodología que guíe todo el proceso de optimización y permita utilizar de forma razonada las diferentes posibilidades. Además se ha incluido una serie de controles estadísticos para asegurar la calidad y validez de los modelos neuronales diseñados. La metodología diseñada localiza aquellas características más importantes de los elementos que intervienen en un impacto balístico para el aprendizaje y posterior validación de los modelos neuronales creados. Gracias a esto, se pueden detectar aquellos factores que menos afectan al modelado del problema y por tanto son susceptibles de no formar parte de las arquitecturas de red propuestas. Esto permite reducir en parte la complejidad asociada a las técnicas tradicionales de generación de impactos balísticos, al reducir el número de variables a recrear. En relación con esto, y para lograr conocer la cantidad mínima de ejemplos necesarios para mantener los resultados en clasificación y regresión, se han realizado experimentos con varias configuraciones de datos para los conjuntos de *train*, *test* y validación o CV.

Por todo lo comentado, gracias a este marco integral, se ha definido una estructura de trabajo global que permite a los ingenieros y científicos conocer el comportamiento de futuros nuevos proyectiles y protecciones, sin la necesidad de recurrir a las técnicas clásicas: experimentación, modelado analítico y simulación numérica. Para ello, se han descrito diferentes situaciones relativas al impacto de sólidos y su tratamiento desde un punto de vista multidisciplinar mediante la utilización de técnicas como la experimentación, simulación numérica y diversas alternativas basadas en el aprendizaje automático. La evaluación y validación realizada sobre este marco aporta las siguientes conclusiones, en relación a los objetivos e hipótesis definidas al comienzo del trabajo:

- Se ha demostrado la viabilidad del PMC para resolver diferentes escenarios balísticos. Gracias a su utilización las herramientas de simulación o modelado pueden utilizarse como complemento al PMC para mejorar el diseño de sistemas de protección.
- El empleo de un PMC optimizado predice tanto el resultado de un impacto como el comportamiento de los objetos que intervienen con un alto porcentaje de acierto. Para ratificar estas conclusiones el PMC ha sido entrenado y probado bajo diversas combinaciones de geometrías y materiales para el proyectil y la protección, y diferentes velocidades de impacto.
- El PMC optimizado permite aproximar con un 98.2417% de acierto el resultado del impacto balístico entre un proyectil y una protección. Para los problemas de regresión

el coeficiente de correlación para el primer escenario (perforación) es de 0.9137 para V_{res} y 0.9538 para M_{res} . Para el segundo escenario de regresión (detención) los valores alcanzados son de 0.9314 para M_{res} .

- La mejora porcentual para el problema de clasificación entre el PMC base y el optimizado es de 5.5790 %. Para los problemas de regresión, en el escenario de perforación la mejora sobre el coeficiente de correlación es de 6.1331 % para V_{res} y 4.5031 % para M_{res} . Por su parte para el escenario de detención la mejora es de 4.2301 % para M_{res} sobre el coeficiente de correlación.
- La comparación realizada con otras propuestas de máquinas de aprendizaje ha demostrado que el PMC es la mejor alternativa para los diferentes escenarios balísticos planteados. Para el problema de clasificación la mejora que obtiene el PMC optimizado respecto a la mejor alternativa, el clasificador RNN, es de 7.7273 %. Para los problemas de regresión, en el primer escenario de perforación la mejora que obtiene el PMC optimizado sobre la mejor alternativa propuesta, red ELN, es de 7.2781 % para V_{res} y 4.6341 % para M_{res} respectivamente. Por su parte en el escenario de detención la mejora del PMC optimizado es de 4.9776 % para M_{res} sobre la red ELN.
- La ventaja de emplear un PMC para resolver los diferentes escenarios es que los resultados se obtienen en tiempo real. El coste computacional es muy inferior por tanto al de las técnicas tradiciones de simulación y modelado. El tiempo de aprendizaje también ha sido minimizado gracias a la metodología, obteniendo un porcentaje de reducción respecto al PMC base para el número de *epochs* del 59.0663 %, 20.5714 % y 16.9943 % para los escenarios de clasificación, regresión con perforación y regresión con detención respectivamente.
- La maximización de la información, mediante la técnica $K_{10}A_{100}$, permite utilizar soluciones basadas en el PMC con más eficiencia en dominios con información reducida.
- Las técnicas propuestas, para detectar la influencia de las variables que describen el problema, permiten seleccionar únicamente aquellas que aportan información reseñable al aprendizaje del PMC. Con esto se reduce la complejidad asociada a la simulación para la obtención de nuevos ensayos balísticos mediante simulación numérica y se reduce la complejidad de la arquitectura del PMC lo que conlleva un menor esfuerzo para su entrenamiento.
- Las medidas de calidad empleadas durante toda la metodología aseguran la validez de los resultados desde un punto de vista estadístico, reduciendo la aparición de errores experimentales o la aparición de posibles aleatoriedades.
- Las conclusiones obtenidas al emplear en primer lugar soluciones basadas en IA, y en segundo lugar la aplicación de una metodología de optimización, pueden extrapolarse a otras áreas como el diseño de estructuras sujetas a impactos de baja velocidad o la seguridad pasiva de vehículos.

En resumen, gracias a esta investigación, se pueden recrear en el futuro de forma fiable nuevos ensayos balísticos. Esto facilitará el posterior diseño de protecciones y el estudio de las consecuencias asociadas a la introducción de diferentes configuraciones de proyectiles y paneles protectores con diferentes características. Además, la sistematización de los diferentes métodos

y técnicas existentes a la hora de configurar y parametrizar una RNA, permite la aplicación de una metodología rápida de optimización sobre un modelo neuronal basado en una topología PMC y algoritmo RP.

7.1. Futuras líneas de investigación

Las futuras líneas de investigación de esta tesis doctoral pueden dividirse en dos categorías. Al tratarse de un trabajo multidisciplinar, la primera está relacionada con la teoría asociada al impacto balístico y su simulación numérica. La segunda hace referencia a las técnicas relacionadas con la IA y la máquina de aprendizaje seleccionada, el PMC.

Para la primera categoría, ya que las RNA han demostrado ser una herramienta válida para el análisis del comportamiento de protecciones metálicas sujetas a impactos, se puede extender su empleo al estudio del fenómeno del impacto para materiales compuestos de fibra de carbono/matriz epoxi. En estas situaciones es interesante realizar una predicción de la extensión del daño, conocer la energía cinética residual o aproximar el DOP cuando el proyectil no perfora la protección.

Por otro lado, en este trabajo sólo se han recogido situaciones de impacto sin ángulo de incidencia, i.e. no se contemplan impactos oblicuos. Si bien la simulación de este tipo de impactos es de mayor complejidad, la aparición de un ángulo de incidencia distinto de cero es en realidad más probable. Por ello una futura propuesta sería la introducción de la oblicuidad como variable de entrada en el problema estudiado para su simulación y posterior tratamiento con un PMC optimizado.

Con respecto a las técnicas empleadas para optimizar, si se tiene en cuenta que la que mayor aporte ha tenido sobre el rendimiento del PMC ha sido la introducción de ruido, sería interesante investigar la influencia que tienen diferentes valores de varianza para la técnica AWGN. El ruido aditivo Gaussiano sigue una distribución $N(0, \sigma^2)$, y varios investigadores han demostrado que la varianza es uno de los aspectos más sensibles para refinar el rendimiento de una RNA [Koistinen and Holmstrom, 1991, Holmstrom and Koistinen, 1992, Karystinos and Pados, 2000].

Una línea de trabajo relevante para el futuro es investigar la posibilidad de adaptar esta metodología, para su empleo sobre diversas alternativas basadas en máquinas de aprendizaje. El objetivo sería poder normalizar y sistematizar todas las alternativas existentes a la hora de optimizar este tipo de soluciones. Asimismo se podría ampliar, aprovechando el estudio realizado en el estado del arte, la cantidad de técnicas y métodos incluidos en cada una de las tareas. Se deberían incluir otras variantes, en principio menos adecuadas para el PMC, pero quizás más útiles para el resto de máquinas de aprendizaje.

Por último, las conclusiones obtenidas a partir de esta tesis doctoral abren nuevas posibilidades en áreas afines, entre las que cabe destacar la seguridad pasiva de vehículos. En

este caso es de especial relevancia el diseño de estructuras cuya misión es absorber energía en choques del tipo *Crashworthiness*, como los que pueden aparecer en accidentes de tráfico o ferroviarios, aterrizajes de emergencia de helicópteros, etc.

7.2. Publicaciones realizadas a raíz de esta tesis doctoral

Como consecuencia de esta investigación se han realizado las siguientes publicaciones en diferentes ámbitos científicos:

- Publicaciones en revista con factor de impacto.
 - Gonzalez-Carrasco, I., Garcia-Crespo, A., Ruiz-Mezcua, B., & Lopez-Cuadrado, J. (2009) Dealing with limited data in ballistic impact scenarios: An empirical comparison of different neural network approaches. *Applied Intelligence* (in press).

Este artículo describe una parte de la metodología de optimización desarrollada y se incluye una comparativa del rendimiento del PMC, para el escenario de clasificación, con respecto a otras alternativas de aprendizaje automático.

- Capítulos de libro.
 - Garcia-Crespo, A., Ruiz-Mezcua, B., Gonzalez-Carrasco, I. & Lopez-Cuadrado, J. (2009) Multilayer Perceptron Training Optimization for High Speed Impacts Classification. *Advances in Electrical Engineering and Computational Science* (Vol. 39, pp. 377-388). Springer.

Este artículo demuestra la posibilidad de optimizar el comportamiento de un PMC mediante la aplicación de una serie de propuestas, destinadas a estudiar el vector de entrada que define el modelo neuronal.

- Publicaciones en congresos internacionales.
 - Gonzalez-Carrasco, I., Garcia-Crespo, A., Ruiz-Mezcua, B., & Lopez-Cuadrado, J. (2008). Network Application for High Speed Impacts Classification. Paper presented at the World Congress on Engineering (WCE 2008), London, United Kingdom.

En este congreso se mostró la validez de las RNA para resolver el problema del impacto balístico, sustituyendo o complementando a las técnicas clásicas de simulación y modelado.

Apéndice A

Abreviaturas

Abreviatura	Significado
AC	Acero (material)
AG	Algoritmo Genético
AI	Artificial Intelligence
AL	Aluminio (material)
ANN	Artificial Neural Network
ANOVA	Analysis Of Variance
ART	Adaptive Resonance Theory
AWGN	Additive White Gaussian Noise
BP	BackPropagation (algoritmo de)
CRM	Contraste de Rangos Múltiples
CFRP	Carbon Fiber Reinforced Plastic
CV	Conjunto de Validación
ELN	Elman Network
GC	Gradiente Conjugado (algoritmo del)
GR	Gradiente (algoritmo del)
H_i	Espesor de la protección en mm
HOCV	Hold Out Cross Validation
HOCVE	Hold Out Cross Validation Equilibrado
HOCVI	Hold Out Cross Validation Iterativo
IA	Inteligencia Artificial
IVC	Indicator of Variable Criticality (Indicador de Criticidad de una Variable)
KFCV	K-Fold Cross Validation
KFCVE	K-Fold Cross Validation Equilibrado
LMS	Least Mean Square
LOOCV	Leave-One-Out Cross Validation
L_p	Longitud del proyectil en mm
LM	Levenberg Marquardt (algoritmo de)
LRS	Learning Rate Scheduling
LRSE	Learning Rate Scheduling Exponencial
LRSL	Learning Rate Scheduling Lineal
LRSO	Learning Rate Scheduling Logarítmico

LSD	Least Significant Difference
MDF	Método de Diferencias Finitas
MEF	Método de Elementos Finitos
MLP	MultiLayer Perceptron
MSE	Mean Square Error
M_p	Masa del proyectil en gr
M_{res}	Masa residual o final del proyectil en gr
NMSE	Normalized Mean Square Error
OBS	Optimal Brain Damage
PMC	Perceptron Multicapa
PSA	Perturbation Sensitivity Analysis
QN	Quasi Newton (algoritmo de)
QP	Quick Propagation (algoritmo de)
RBF	Radial Base Function (Función de Base Radial)
RBFN	Radial Base Function Network (Red Neuronal con RBF)
RBP	Remuestreo Bootstrap
RDB	Regla Delta Bar
RDBE	Regla Delta Bar Extendida
RNA	Red de Neuronas Artificiales
RNN	Recurrent Neural Network (Red Neuronal Recurrente)
RP	RetroPropagación (algoritmo de)
RPE	RetroPropagación Extendido (algoritmo de)
R_p	Radio del proyectil en mm
SAM	Sensitivity About the Mean
SCE	Suma de Cuadrados Explicada o entre-grupos
SCR	Suma de Cuadrados Residual o intra-grupos
SCT	Suma de Cuadrados Totales
SOFMS	Self-Organizing Feature Maps (Mapas auto organizados)
SVM	Support Vector Machine (Máquina de Vectores de Soporte)
SWM	Stepwise method
T_i	Tipo de material de la protección
T_p	Tipo de material del proyectil
UE	Uranio Empobrecido
VC	Vapnik-Chervonenkis (dimensión)
VIECV	Variante Iterativa Equilibrada Cross Validation
V_{res}	Velocidad residual del proyectil en m/s
V_p	Velocidad del proyectil en m/s
WHA	Wolframio
Y_i	Lado de la protección en mm

Apéndice B

Listado de ensayos balísticos

B.1. Listado de ensayos obtenidos con simulación numérica

Tabla B.1: Listado de ensayos generados con simulación numérica

N°	H_p (mm)	R_b (mm)	L_b (mm)	V_b (m/s)	M_b (gr)	T_b	T_p	P_b	M_{res} (gr)	(%M)	V_{res} (m/s)
1	13.00	6.00	45.00	593.00	39.95	2	2	no	27.00	32.42	0.00
2	7.00	6.00	34.00	990.00	30.19	2	2	si	22.00	27.12	805.00
3	4.00	4.00	21.00	654.00	8.29	2	2	si	7.00	15.52	265.00
4	11.00	6.00	33.00	856.00	29.30	2	2	si	15.00	48.80	413.00
5	9.00	5.00	30.00	529.00	18.50	2	2	no	14.00	24.31	0.00
6	9.00	5.00	31.00	1101.00	19.11	2	2	si	11.00	42.45	850.00
7	8.00	8.00	47.00	476.00	74.18	2	2	no	68.00	8.33	0.00
8	11.00	6.00	34.00	559.00	30.19	2	2	no	22.00	27.12	0.00
9	12.00	6.00	45.00	499.00	39.95	2	2	no	31.00	22.41	0.00
10	13.00	7.00	55.00	934.00	66.46	2	2	si	44.00	33.80	712.00
11	10.00	5.00	40.00	447.00	24.66	2	2	no	21.00	14.85	0.00
12	5.00	5.00	31.00	940.00	19.11	2	2	si	15.00	21.52	785.00
13	15.00	7.00	56.00	453.00	67.67	2	2	no	57.00	15.77	0.00
14	6.00	4.00	25.00	608.00	9.86	2	2	no	7.00	29.04	0.00
15	15.00	7.00	48.00	668.00	58.00	2	2	no	31.00	46.56	0.00
16	4.00	4.00	24.00	1128.00	9.47	2	2	si	6.70	29.25	975.00
17	9.00	8.00	44.00	827.00	69.45	2	2	si	52.00	25.12	611.00
18	7.00	5.00	26.00	512.00	16.03	2	2	no	14.00	12.66	0.00
19	6.00	5.00	33.00	902.00	20.35	2	2	si	15.00	26.27	713.00
20	7.00	5.00	28.00	645.00	17.26	2	2	no	11.80	31.65	0.00
21	5.00	4.00	27.00	910.00	10.65	2	2	si	7.50	29.60	704.00
22	11.00	8.00	62.00	956.00	97.86	2	2	si	75.40	22.95	809.00
23	7.00	7.00	33.00	601.00	39.88	2	2	si	31.80	20.26	208.00
24	7.00	6.00	41.00	582.00	36.40	2	2	si	28.50	21.70	194.00
25	12.00	7.00	42.00	543.00	50.75	2	2	no	38.00	25.13	0.00
26	11.00	7.00	52.00	1133.00	62.84	2	2	si	45.00	28.39	986.00
27	16.00	8.00	53.00	1073.00	83.65	2	2	si	50.40	39.75	832.00

Tabla B.1: Continuación de página anterior

N°	H_p (mm)	R_0 (mm)	L_b (mm)	V_b (m/s)	M_b (gr)	T_b	T_p	P_b	M_{rcs} (gr)	(%M)	V_{rcs} (m/s)
28	8.00	7.00	33.00	1190.00	39.88	2	2	si	27.00	32.29	1002.00
29	8.00	4.00	29.00	1004.00	11.44	2	2	si	5.90	48.44	660.00
30	10.00	8.00	53.00	420.00	83.65	2	2	no	80.30	4.01	0.00
31	4.00	4.00	25.00	1076.00	9.86	2	2	si	7.20	27.01	920.00
32	10.00	6.00	41.00	880.00	36.40	2	2	si	23.50	35.44	619.00
33	9.00	5.00	30.00	1016.00	18.50	2	2	si	10.40	43.77	713.00
34	5.00	4.00	24.00	1166.00	9.47	2	2	si	6.14	35.16	982.00
35	4.00	3.00	14.00	804.00	3.11	2	2	no	1.33	57.20	0.00
36	14.00	7.00	49.00	1122.00	59.21	2	2	si	36.75	37.94	910.00
37	7.00	8.00	36.00	810.00	56.82	2	2	si	43.50	23.44	596.00
38	10.00	7.00	34.00	1090.00	41.09	2	2	si	25.70	37.45	847.00
39	11.00	8.00	41.00	1015.00	64.71	2	2	si	43.00	33.55	789.00
40	6.00	3.00	18.00	663.00	4.00	2	2	no	2.17	45.68	0.00
41	11.00	6.00	47.00	787.00	41.73	2	2	si	25.20	39.61	455.00
42	11.00	7.00	52.00	930.00	62.84	2	2	si	44.30	29.50	737.00
43	7.00	3.00	24.00	665.00	5.33	2	2	no	2.50	53.07	0.00
44	13.00	6.00	47.00	1188.00	41.73	2	2	si	25.60	38.65	997.00
45	12.00	6.00	37.00	633.00	32.85	2	2	no	20.20	38.51	0.00
46	7.00	6.00	26.00	1133.00	23.08	2	2	si	14.80	35.88	905.00
47	7.00	4.00	22.00	583.00	8.68	2	2	no	6.50	25.12	0.00
48	11.00	7.00	35.00	613.00	42.29	2	2	no	28.90	31.67	0.00
49	11.00	5.00	40.00	863.00	24.66	2	2	si	12.00	51.34	470.00
50	5.00	3.00	21.00	769.00	4.66	2	2	si	2.10	54.95	504.00
51	9.00	6.00	30.00	1078.00	26.63	2	2	si	16.40	38.43	819.00
52	11.00	6.00	37.00	912.00	32.85	2	2	si	18.90	42.46	586.00
53	13.00	6.00	44.00	1094.00	39.06	2	2	si	23.60	39.59	865.00
54	4.00	4.00	23.00	661.00	9.08	2	2	si	7.00	22.87	317.00
55	8.00	6.00	41.00	632.00	36.40	2	2	si	26.30	27.75	227.00
56	8.00	5.00	37.00	781.00	22.81	2	2	si	14.70	35.56	465.00
57	7.00	4.00	24.00	790.00	9.47	2	2	si	4.50	52.48	95.00

Tabla B.1: Continuación de página anterior

N°	H_p (mm)	R_b (mm)	L_b (mm)	V_b (m/s)	M_b (gr)	T_b	T_p	P_b	M_{rcs} (gr)	(%M)	V_{rcs} (m/s)
58	9.00	4.00	29.00	1114.00	11.44	2	2	si	5.40	52.81	772.00
59	10.00	5.00	36.00	732.00	22.20	2	2	no	10.60	52.24	0.00
60	7.00	4.00	27.00	1099.00	10.65	2	2	si	6.40	39.93	858.00
61	3.00	4.00	18.00	798.00	7.10	2	2	si	5.40	23.97	410.00
62	10.00	6.00	47.00	1148.00	41.73	2	2	si	30.90	25.95	1004.00
63	9.00	7.00	29.00	892.00	35.04	2	2	si	22.30	36.37	722.00
64	4.00	4.00	16.00	987.00	6.31	2	2	si	2.02	68.00	629.00
65	8.00	6.00	41.00	1018.00	36.40	2	2	si	27.70	23.90	861.00
66	10.00	6.00	42.00	603.00	37.29	2	2	no	26.50	28.93	0.00
67	7.00	4.00	23.00	1195.00	9.08	2	2	si	3.88	57.25	851.00
68	6.00	4.00	30.00	716.00	11.84	2	2	si	7.17	39.43	263.00
69	15.00	4.00	16.00	1040.00	6.31	2	2	no	1.42	77.51	0.00
70	4.00	5.00	23.00	700.00	14.18	2	2	si	11.10	21.72	411.00
71	5.00	4.00	36.00	959.00	34.00	3	2	si	31.30	7.90	857.00
72	10.00	7.00	34.00	1109.00	97.00	3	1	si	96.00	1.00	1069.00
73	9.00	6.00	50.00	947.00	105.00	4	3	si	76.60	27.00	800.00
74	6.00	3.00	56.00	638.00	29.00	4	2	si	25.00	13.80	442.00
75	13.00	7.00	50.00	526.00	143.00	3	2	si	132.00	7.70	353.00
76	13.00	8.00	24.00	775.00	90.00	3	2	si	84.70	5.90	539.00
77	6.00	8.00	40.00	933.00	150.00	3	1	si	148.40	1.10	907.00
78	10.00	3.00	21.00	599.00	11.00	3	2	si	8.30	24.50	112.00
79	15.00	6.00	26.00	818.00	55.00	4	2	si	46.00	16.40	566.00
80	4.00	7.00	53.00	621.00	64.00	2	3	si	43.00	32.80	342.00
81	5.00	4.00	27.00	1118.00	11.00	2	2	si	3.60	67.30	665.00
82	14.00	8.00	25.00	713.00	94.00	3	2	si	89.00	5.30	466.00
83	9.00	7.00	42.00	905.00	120.00	4	2	si	110.00	8.30	813.00
84	6.00	5.00	31.00	920.00	45.00	3	3	si	34.00	24.40	749.00
85	8.00	5.00	35.00	982.00	22.00	2	2	si	10.50	52.30	644.00
86	7.00	3.00	25.00	1136.00	6.00	2	2	si	2.70	55.00	863.00
87	5.00	6.00	34.00	541.00	30.00	2	3	no	21.00	30.00	0.00

Tabla B.1: Continuación de página anterior

N°	H_p (mm)	R_b (mm)	L_b (mm)	V_b (mm)	V_b (m/s)	M_b (gr)	T_b	T_p	P_b	M_{rcs} (gr)	(%M)	V_{rcs} (m/s)
88	6.00	5.00	28.00	580.00	41.00	4	3	si	31.60	22.90	277.00	
89	7.00	4.00	39.00	737.00	15.00	2	2	si	9.70	35.30	410.00	
90	7.00	6.00	45.00	1078.00	95.00	4	1	si	93.00	2.10	1054.00	
91	14.00	6.00	32.00	855.00	67.00	3	2	si	57.50	14.20	672.00	
92	10.00	4.00	29.00	697.00	27.00	4	2	si	23.00	14.80	484.00	
93	10.00	5.00	12.00	649.00	18.00	3	3	no	13.00	27.80	0.00	
94	5.00	4.00	26.00	1186.00	24.00	3	2	si	21.00	12.50	1098.00	
95	7.00	6.00	23.00	569.00	20.00	2	3	no	12.70	36.50	0.00	
96	4.00	4.00	19.00	916.00	18.00	3	1	si	18.00	0.00	888.00	
97	5.00	7.00	37.00	1003.00	106.00	3	2	si	97.00	8.50	913.00	
98	9.00	7.00	20.00	1194.00	57.00	3	2	si	50.00	12.30	1003.00	
99	7.00	6.00	23.00	1082.00	48.00	4	3	si	31.00	35.40	824.00	
100	16.00	8.00	56.00	905.00	209.00	3	3	si	131.00	37.30	647.00	
101	7.00	5.00	42.00	964.00	26.00	2	3	no	2.70	89.60	0.00	
102	14.00	7.00	32.00	567.00	92.00	3	2	si	88.50	3.80	332.00	
103	6.00	5.00	32.00	847.00	20.00	2	2	si	12.50	37.50	552.00	
104	14.00	5.00	35.00	943.00	22.00	2	3	no	3.70	83.20	0.00	
105	7.00	4.00	42.00	1108.00	39.00	3	2	si	32.00	17.90	988.00	
106	17.00	7.00	19.00	870.00	54.00	4	2	si	47.00	13.00	481.00	
107	8.00	8.00	39.00	648.00	146.00	3	2	si	141.00	3.40	547.00	
108	4.00	6.00	26.00	893.00	55.00	3	2	si	52.00	5.50	804.00	
109	11.00	7.00	28.00	725.00	80.00	4	2	si	74.60	6.80	546.00	
110	5.00	4.00	46.00	892.00	43.00	3	1	si	42.80	0.50	873.00	
111	8.00	7.00	47.00	1043.00	135.00	4	2	si	117.00	13.30	947.00	
112	4.00	5.00	24.00	1007.00	35.00	3	1	si	35.00	0.00	980.00	
113	8.00	3.00	29.00	966.00	15.00	4	2	si	12.70	15.30	840.00	
114	8.00	5.00	55.00	1158.00	80.00	3	1	si	78.80	1.50	1134.00	
115	6.00	4.00	39.00	980.00	36.00	3	2	si	32.00	11.10	879.00	
116	11.00	8.00	59.00	1067.00	221.00	4	3	si	132.00	40.30	905.00	
117	8.00	5.00	41.00	677.00	25.00	2	1	si	24.50	2.00	613.00	

Tabla B.1: Continuación de página anterior

N°	H_p (mm)	R_0 (mm)	L_b (mm)	V_b (m/s)	M_b (gr)	T_b	T_p	P_b	M_{rcs} (gr)	(%M)	V_{rcs} (m/s)
118	4.00	43.00	6.00	560.00	90.00	3	2	si	89.90	0.10	467.00
119	7.00	25.00	7.00	728.00	30.00	2	2	si	21.00	30.00	340.00
120	5.00	16.00	4.00	553.00	15.00	3	2	si	14.60	2.70	329.00
121	8.00	42.00	5.00	897.00	61.00	3	2	si	55.00	9.80	767.00
122	6.00	30.00	5.00	749.00	18.00	2	1	si	18.00	0.00	690.00
123	13.00	23.00	3.00	812.00	12.00	3	3	no	4.20	65.00	0.00
124	7.00	33.00	8.00	596.00	123.00	3	3	si	104.00	15.40	328.00
125	8.00	25.00	6.00	945.00	53.00	4	1	si	53.00	0.00	908.00
126	7.00	24.00	6.00	1095.00	50.00	3	2	si	43.00	14.00	966.00
127	8.00	26.00	4.00	1180.00	24.00	3	2	si	20.00	16.70	1071.00
128	8.00	17.00	7.00	1077.00	49.00	3	2	si	44.80	8.60	890.00
129	9.00	30.00	6.00	745.00	63.00	3	1	si	63.00	0.00	715.00
130	9.00	26.00	7.00	594.00	31.00	2	2	no	24.40	21.30	0.00
131	11.00	34.00	7.00	1073.00	41.00	2	1	si	38.60	5.90	976.00
132	6.00	23.00	4.00	1051.00	9.00	2	3	si	1.80	80.00	261.00
133	8.00	39.00	7.00	822.00	47.00	2	3	no	14.40	69.40	0.00
134	7.00	41.00	6.00	559.00	86.00	3	3	si	73.00	15.10	294.00
135	12.00	35.00	7.00	1134.00	100.00	3	1	si	100.00	0.00	1087.00
136	7.00	30.00	7.00	984.00	86.00	3	1	si	86.00	0.00	955.00
137	3.00	16.00	3.00	1173.00	8.00	3	3	si	8.00	0.00	1170.00
138	5.00	29.00	5.00	888.00	42.00	4	3	si	33.70	19.80	745.00
139	7.00	35.00	5.00	550.00	51.00	4	1	si	51.00	0.00	530.00
140	12.00	40.00	5.00	1040.00	58.00	3	2	si	49.50	14.70	928.00
141	8.00	42.00	5.00	897.00	61.00	3	2	si	55.00	9.80	767.00
142	7.00	29.00	4.00	884.00	11.00	2	2	si	5.80	47.30	508.00
143	7.00	37.00	7.00	1111.00	106.00	4	2	si	98.00	7.50	1032.00
144	9.00	31.00	4.00	1187.00	29.00	3	2	si	24.00	17.20	1081.00
145	7.00	33.00	5.00	527.00	48.00	4	2	si	46.40	3.30	372.00
146	10.00	37.00	6.00	1122.00	78.00	3	3	si	52.00	33.30	919.00
147	6.00	23.00	3.00	1089.00	12.00	4	2	si	10.00	16.70	959.00

Tabla B.1: Continuación de página anterior

N°	H_p (mm)	R_b (mm)	L_b (mm)	V_b (m/s)	M_b (gr)	T_b	T_p	P_b	M_{rcs} (gr)	(%M)	V_{rcs} (m/s)
148	8.00	5.00	23.00	864.00	14.00	2	3	no	3.60	74.30	0.00
149	8.00	7.00	37.00	1161.00	45.00	2	1	si	42.00	6.70	1089.00
150	10.00	5.00	33.00	776.00	20.00	2	2	si	8.70	56.50	100.00
151	8.00	7.00	33.00	1162.00	95.00	4	2	si	85.50	10.00	1065.00
152	8.00	5.00	31.00	507.00	45.00	4	2	si	43.00	4.40	312.00
153	6.00	6.00	27.00	745.00	57.00	4	1	si	57.00	0.00	720.00
154	10.00	6.00	31.00	980.00	65.00	4	2	si	56.00	13.80	855.00
155	6.00	4.00	26.00	629.00	24.00	3	2	si	22.00	8.30	463.00
156	6.00	6.00	31.00	512.00	28.00	2	1	si	28.00	0.00	450.00
157	7.00	4.00	27.00	1110.00	25.00	3	3	si	16.70	33.20	927.00
158	8.00	7.00	38.00	1195.00	109.00	3	1	si	109.00	0.00	1162.00
159	9.00	5.00	28.00	760.00	41.00	4	3	si	25.20	38.50	419.00
160	3.00	3.00	17.00	1002.00	4.00	2	2	si	2.50	37.50	787.00
161	30.00	6.00	45.00	713.00	63.68	2	1	no	25.73	59.60	0.00
162	20.00	6.00	34.00	636.00	19.59	3	3	no	10.25	47.70	0.00
163	32.00	4.00	21.00	511.00	71.91	4	2	no	50.12	30.30	0.00
164	8.00	6.00	33.00	270.00	43.73	4	2	si	42.59	2.60	119.30
165	29.00	5.00	30.00	596.00	69.27	4	1	si	67.19	3.00	503.00
166	18.00	5.00	31.00	575.00	39.95	3	3	no	25.49	36.20	0.00
167	45.00	8.00	47.00	466.00	71.91	4	2	no	55.80	22.40	0.00
168	22.00	6.00	34.00	520.00	175.39	3	2	no	135.75	22.60	0.00
169	32.00	6.00	45.00	692.00	45.53	2	2	no	15.48	66.00	0.00
170	28.00	7.00	55.00	833.00	158.33	3	2	si	111.78	29.40	494.00
171	24.00	5.00	40.00	781.00	58.75	3	3	no	11.57	80.30	0.00
172	35.00	5.00	31.00	429.00	45.53	3	1	si	36.97	18.80	121.20
173	10.00	7.00	56.00	448.00	160.00	4	1	si	151.68	5.20	433.60
174	47.00	4.00	25.00	394.00	137.14	4	3	no	118.22	13.80	0.00
175	21.00	7.00	48.00	675.00	23.32	4	3	no	10.78	53.80	0.00
176	18.00	4.00	24.00	515.00	22.56	3	1	si	22.54	0.10	433.00
177	23.00	8.00	44.00	856.00	17.26	2	2	no	5.39	68.80	0.00

Tabla B.1: Continuación de página anterior

N°	H_p (mm)	R_b (mm)	L_b (mm)	V_b (m/s)	M_b (gr)	T_b	T_p	P_b	M_{rcs} (gr)	(%M)	V_{rcs} (m/s)
178	45.00	5.00	26.00	277.00	48.47	4	2	no	48.27	0.40	0.00
179	11.00	5.00	33.00	520.00	37.90	3	3	no	26.99	28.80	0.00
180	8.00	5.00	28.00	747.00	69.45	2	3	no	23.13	66.70	0.00
181	7.00	4.00	27.00	769.00	39.88	3	1	si	35.93	9.90	739.00
182	10.00	8.00	62.00	834.00	97.86	2	1	si	96.88	1.00	946.00
183	35.00	7.00	33.00	429.00	25.38	2	1	si	25.33	0.20	173.00
184	31.00	6.00	41.00	568.00	86.72	3	2	no	67.29	22.40	0.00
185	18.00	7.00	42.00	611.00	199.28	4	2	si	77.72	61.00	343.00
186	32.00	7.00	52.00	612.00	148.57	4	2	si	120.34	19.00	115.00
187	17.00	8.00	53.00	417.00	120.00	3	1	si	118.80	1.00	439.00
188	28.00	7.00	33.00	603.00	27.06	2	2	no	24.43	9.70	0.00
189	13.00	4.00	29.00	503.00	39.88	4	2	no	35.61	10.70	0.00
190	9.00	8.00	53.00	660.00	199.28	3	2	si	189.32	5.00	635.90
191	34.00	4.00	25.00	623.00	23.50	3	2	no	12.78	45.60	0.00
192	10.00	6.00	41.00	699.00	86.72	3	2	si	84.72	2.30	579.00
193	21.00	5.00	30.00	563.00	43.73	4	2	no	38.66	11.60	0.00
194	19.00	4.00	24.00	982.00	9.47	2	2	no	2.94	69.00	0.00
195	6.00	3.00	14.00	1179.00	3.11	2	2	si	1.53	50.80	765.00
196	11.00	7.00	49.00	653.00	141.06	3	3	si	115.24	18.30	361.00
197	30.00	8.00	36.00	502.00	154.16	4	3	no	111.92	27.40	0.00
198	17.00	7.00	34.00	766.00	97.88	3	3	no	47.18	51.80	0.00
199	25.00	8.00	41.00	514.00	134.34	3	2	no	113.52	15.50	0.00
200	24.00	3.00	18.00	704.00	9.52	3	1	si	9.52	0.00	516.00
201	23.00	6.00	47.00	613.00	41.73	2	2	no	21.70	48.00	0.00
202	39.00	7.00	52.00	498.00	149.70	3	1	si	149.70	0.00	417.00
203	30.00	3.00	24.00	425.00	23.08	2	1	no	3.72	83.90	0.00
204	35.00	6.00	47.00	1104.00	77.67	2	2	no	15.22	80.40	0.00
205	27.00	6.00	37.00	384.00	41.73	4	3	no	35.55	14.80	0.00
206	34.00	6.00	26.00	1086.00	5.33	2	3	no	1.12	79.00	0.00
207	10.00	4.00	22.00	672.00	20.68	3	2	si	19.21	7.10	404.00

Tabla B.1: Continuación de página anterior

N°	H_p (mm)	R_b (mm)	L_b (mm)	V_b (m/s)	M_b (gr)	T_b	T_p	P_b	M_{res} (gr)	(%M)	V_{res} (m/s)
208	14.00	7.00	35.00	680.00	100.00	4	2	si	96.00	4.00	475.00
209	19.00	5.00	40.00	443.00	58.31	4	2	no	57.84	0.80	0.00
210	21.00	3.00	21.00	634.00	11.10	3	2	no	8.13	26.80	0.00
211	7.00	6.00	30.00	636.00	93.06	2	2	si	76.77	17.50	327.00
212	16.00	6.00	37.00	598.00	77.67	4	1	si	77.12	0.70	550.00
213	12.00	6.00	44.00	633.00	26.63	3	2	si	26.10	2.00	544.00
214	23.00	4.00	23.00	453.00	21.46	4	2	no	21.07	1.80	0.00
215	7.00	6.00	41.00	747.00	52.48	3	3	si	45.24	13.80	564.00
216	6.00	5.00	37.00	587.00	27.26	4	2	si	27.26	0.00	504.00
217	27.00	4.00	24.00	779.00	22.39	4	1	si	22.39	0.00	617.00
218	10.00	4.00	29.00	598.00	53.94	3	2	si	53.13	1.50	378.00
219	5.00	5.00	36.00	521.00	86.72	4	2	si	86.72	0.00	445.00
220	11.00	4.00	27.00	619.00	10.65	2	2	no	6.50	39.00	0.00

donde:

- $\%M = \frac{M_b - M_{res}}{M_b}$
- Para T_b y T_p : 1 es Aluminio 6061-T6, 2 es Acero 4340 AISI, 3 es Uranio Empobrecido y 4 es Wolframio.

B.2. Listado de ensayos balísticos con ruido

Tabla B.2: Listado de ensayos generados con ruido

N°	H_p (mm)	R_b (mm)	L_b (mm)	V_b (m/s)	M_b (gr)	T_b	T_p	P_b	M_{res} (gr)	(%M)	V_{res} (m/s)
220	8.54	5.99	42.65	897.51	61.50	3	2	si	55.47	9.80	767.00
221	6.53	3.93	14.55	1179.51	3.61	2	2	si	1.77	50.80	765.00
222	8.54	8.00	33.62	1190.51	40.38	2	2	si	27.34	32.29	1002.00
223	5.53	4.97	24.59	1166.51	9.97	2	2	si	6.46	35.16	982.00
224	8.54	8.00	38.63	1195.51	109.50	3	1	si	109.50	0.00	1162.00
225	17.59	9.00	53.68	417.50	199.78	3	1	si	197.78	1.00	439.00
226	5.53	4.97	27.60	1118.51	11.50	2	2	si	3.76	67.30	665.00
227	13.57	3.93	23.58	812.51	12.50	3	3	no	4.38	65.00	0.00
228	9.55	5.99	30.61	529.50	19.00	2	2	no	14.38	24.31	0.00
229	25.64	9.00	41.64	514.50	154.66	3	2	no	130.69	15.50	0.00
230	30.66	9.00	36.63	502.50	134.84	4	3	no	97.90	27.40	0.00
231	4.52	4.97	19.57	916.51	18.50	3	1	si	18.50	0.00	888.00
232	10.56	8.00	34.62	1090.51	41.59	2	2	si	26.01	37.45	847.00
233	15.58	4.97	16.56	1040.51	6.81	2	2	no	1.53	77.51	0.00
234	11.56	8.00	28.60	725.51	80.50	4	2	si	75.03	6.80	546.00
235	8.54	8.00	39.64	822.51	47.50	2	3	no	14.54	69.40	0.00
236	31.67	7.00	41.64	568.50	87.22	3	2	no	67.68	22.40	0.00
237	7.54	3.93	24.59	665.50	5.83	2	2	no	2.73	53.07	0.00
238	29.66	5.99	30.61	596.50	44.23	4	1	si	42.90	3.00	503.00
239	8.54	5.99	28.60	747.51	17.76	2	3	no	5.92	66.70	0.00
240	32.67	4.97	21.58	511.50	20.09	4	2	no	14.00	30.30	0.00
241	7.54	7.00	30.61	636.50	27.13	2	2	si	22.39	17.50	327.00
242	4.52	3.93	14.55	804.51	3.61	2	2	no	1.54	57.20	0.00
243	45.73	9.00	47.66	466.50	175.89	4	2	no	136.49	22.40	0.00
244	7.54	7.00	23.58	1082.51	48.50	4	3	si	31.33	35.40	824.00
245	6.53	4.97	25.59	608.50	10.36	2	2	no	7.35	29.04	0.00
246	7.54	7.00	41.64	582.50	36.90	2	2	si	28.89	21.70	194.00
247	14.58	9.00	25.59	713.51	94.50	3	2	si	89.49	5.30	466.00

Tabla B.2: Continuación de página anterior

N°	H_p (mm)	R_b (mm)	L_b (mm)	V_b (m/s)	M_b (gr)	T_b	T_p	P_b	M_{rcs} (gr)	(%M)	V_{rcs} (m/s)
248	10.56	5.99	40.64	447.50	25.16	2	2	no	21.43	14.85	0.00
249	4.52	5.99	24.59	1007.51	35.50	3	1	si	35.50	0.00	980.00
250	4.52	4.97	21.58	654.50	8.79	2	2	si	7.42	15.52	265.00
251	7.54	7.00	26.59	1133.51	23.58	2	2	si	15.12	35.88	905.00
252	5.53	4.97	26.59	1186.51	24.50	3	2	si	21.44	12.50	1098.00
253	11.56	4.97	27.60	619.50	11.15	2	2	no	6.80	39.00	0.00
254	10.56	4.97	22.58	672.50	21.18	3	2	si	19.68	7.10	404.00
255	10.56	7.00	41.64	699.51	87.22	3	2	si	85.21	2.30	579.00
256	9.55	8.00	29.60	892.51	35.54	2	2	si	22.62	36.37	722.00
257	11.56	8.00	49.67	653.50	141.56	3	3	si	115.65	18.30	361.00
258	16.59	9.00	56.69	905.51	209.50	3	3	si	131.36	37.30	647.00
259	8.54	8.00	17.56	1077.51	49.50	3	2	si	45.24	8.60	890.00
260	5.53	8.00	37.63	1003.51	106.50	3	2	si	97.45	8.50	913.00
261	16.59	9.00	53.68	1073.51	84.15	2	2	si	50.70	39.75	832.00
262	18.60	5.99	31.61	575.50	46.03	3	3	no	29.37	36.20	0.00
263	6.53	5.99	31.61	920.51	45.50	3	3	si	34.40	24.40	749.00
264	7.54	5.99	26.59	512.50	16.53	2	2	no	14.44	12.66	0.00
265	9.55	7.00	50.67	947.51	105.50	4	3	si	77.02	27.00	800.00
266	10.56	5.99	12.54	649.50	18.50	3	3	no	13.36	27.80	0.00
267	10.56	4.97	29.60	697.51	27.50	4	2	si	23.43	14.80	484.00
268	6.53	3.93	56.69	638.50	29.50	4	2	si	25.43	13.80	442.00
269	13.57	7.00	45.66	593.50	40.45	2	2	no	27.34	32.42	0.00
270	5.53	7.00	34.62	541.50	30.50	2	3	no	21.35	30.00	0.00
271	6.53	5.99	32.61	847.51	20.50	2	2	si	12.81	37.50	552.00
272	20.61	7.00	34.62	636.50	72.41	3	3	no	37.87	47.70	0.00
273	8.54	7.00	41.64	1018.51	36.90	2	2	si	28.08	23.90	861.00
274	28.65	8.00	55.69	833.51	158.83	3	2	si	112.13	29.40	494.00
275	21.61	5.99	30.61	563.50	44.23	4	2	no	39.10	11.60	0.00
276	10.56	7.00	47.66	1148.51	42.23	2	2	si	31.27	25.95	1004.00
277	12.57	7.00	44.65	633.50	93.56	3	2	si	91.69	2.00	544.00

Tabla B.2: Continuación de página anterior

N°	H_p (mm)	R_b (mm)	L_b (mm)	V_b (m/s)	M_b (gr)	T_b	T_p	P_b	M_{rcs} (gr)	(%M)	V_{rcs} (m/s)
278	10.56	3.93	21.58	599.50	11.50	3	2	si	8.68	24.50	112.00
279	12.57	8.00	35.62	1134.51	100.50	3	1	si	100.50	0.00	1087.00
280	11.56	7.00	47.66	787.51	42.23	2	2	si	25.50	39.61	455.00
281	7.54	3.93	25.59	1136.51	6.50	2	2	si	2.93	55.00	863.00
282	21.61	3.93	21.58	634.50	11.60	3	2	no	8.49	26.80	0.00
283	7.54	8.00	37.63	1111.51	106.50	4	2	si	98.51	7.50	1032.00
284	13.57	7.00	47.66	1188.51	42.23	2	2	si	25.91	38.65	997.00
285	11.56	9.00	59.70	1067.51	221.50	4	3	si	132.24	40.30	905.00
286	8.54	7.00	41.64	632.50	36.90	2	2	si	26.66	27.75	227.00
287	28.65	8.00	33.62	603.50	40.38	2	2	no	36.46	9.70	0.00
288	35.69	5.99	31.61	429.50	46.03	3	1	si	37.38	18.80	121.20
289	32.67	7.00	45.66	692.51	40.45	2	2	no	13.75	66.00	0.00
290	7.54	7.00	41.64	747.51	87.22	3	3	si	75.18	13.80	564.00
291	7.54	4.97	42.65	1108.51	39.50	3	2	si	32.43	17.90	988.00
292	8.54	5.99	41.64	677.50	25.50	2	1	si	24.99	2.00	613.00
293	13.57	8.00	55.69	934.51	66.96	2	2	si	44.33	33.80	712.00
294	6.53	9.00	40.64	933.51	150.50	3	1	si	148.84	1.10	907.00
295	10.56	7.00	37.63	1122.51	78.50	3	3	si	52.36	33.30	919.00
296	6.53	4.97	26.59	629.50	24.50	3	2	si	22.47	8.30	463.00
297	4.52	7.00	26.59	893.51	55.50	3	2	si	52.45	5.50	804.00
298	7.54	9.00	36.63	810.51	57.32	2	2	si	43.88	23.44	596.00
299	6.53	4.97	30.61	716.51	12.34	2	2	si	7.47	39.43	263.00
300	7.54	7.00	24.59	1095.51	50.50	3	2	si	43.43	14.00	966.00
301	18.60	8.00	42.65	611.50	120.50	4	2	si	47.00	61.00	343.00
302	9.55	5.99	30.61	1016.51	19.00	2	2	si	10.68	43.77	713.00
303	10.56	7.00	41.64	880.51	36.90	2	2	si	23.82	35.44	619.00
304	8.54	5.99	31.61	507.50	45.50	4	2	si	43.50	4.40	312.00
305	6.53	7.00	27.60	745.51	57.50	4	1	si	57.50	0.00	720.00
306	8.54	9.00	39.64	648.50	146.50	3	2	si	141.52	3.40	547.00
307	4.52	4.97	23.58	661.50	9.58	2	2	si	7.39	22.87	317.00

Tabla B.2: Continuación de página anterior

N°	H_p (mm)	R_b (mm)	L_b (mm)	V_b (m/s)	M_b (gr)	T_b	T_p	P_b	M_{res} (gr)	(%M)	V_{res} (m/s)
308	34.68	4.97	25.59	623.50	24.00	3	2	no	13.06	45.60	0.00
309	6.53	5.99	28.60	580.50	41.50	4	3	si	32.00	22.90	277.00
310	9.55	8.00	20.57	1194.51	57.50	3	2	si	50.43	12.30	1003.00
311	8.54	7.00	25.59	945.51	53.50	4	1	si	53.50	0.00	908.00
312	5.53	5.99	31.61	940.51	19.61	2	2	si	15.39	21.52	785.00
313	7.54	5.99	35.62	550.50	51.50	4	1	si	51.50	0.00	530.00
314	7.54	7.00	41.64	559.50	86.50	3	3	si	73.44	15.10	294.00
315	7.54	5.99	33.62	527.50	48.50	4	2	si	46.90	3.30	372.00
316	9.55	5.99	28.60	760.51	41.50	4	3	si	25.52	38.50	419.00
317	10.56	7.00	42.65	603.50	37.79	2	2	no	26.86	28.93	0.00
318	8.54	9.00	47.66	476.50	74.68	2	2	no	68.46	8.33	0.00
319	8.54	3.93	29.60	966.51	15.50	4	2	si	13.13	15.30	840.00
320	7.54	4.97	23.58	1195.51	9.58	2	2	si	4.09	57.25	851.00

donde:

- Para las entradas se aplica el procedimiento $x_h^n = x_i + hS_x$ y para las salidas $y_h^n = y_i + hS_y$

Bibliografía

- ABAQUS. *ABAQUS/Explicit v6.4 Users Manual*. ABAQUS Inc., Richmond, USA., 2003.
- S. Abrate. *Impact on composite structures*. Cambridge University Press, 1998.
- H. Adeli. *Advances in design optimization*. Chapman & Hall, 1994.
- H. Adeli. Neural networks in civil engineering: 1989-2000. *Computer-Aided Civil and Infrastructure Engineering*, 16:126–142(17), 2001.
- H. Adeli and C. Yeh. Perceptron learning in engineering design. *Microcomputers in Civil Engineering*, 4:247–256, 1989.
- B.S. Ahn, S.S. Cho, and C.Y. Kim. The integrated methodology of rough set theory and artificial neural network for business failure prediction. *Expert Systems with Applications*, 18(2):65–74, 2000.
- H. Akaike. A new look at the statistical model identification. *Automatic Control, IEEE Transactions on*, 19(6):716–723, 1974.
- F. Akhavan, S.E. Watkins, and K. Chandrashekhara. Prediction of impact contact forces of composite plates using fiber optic sensors and neural networks. *Mechanics of Composite Materials and Structures*, 7:195–205(11), 2000.
- E. Alpaydin. Multiple networks for function learning. *Neural Networks, 1993., IEEE International Conference on*, pages 9–14 vol.1, 1993.
- G. An. The effects of adding noise during backpropagation training on a generalization performance. *Neural Comput.*, 8(3):643–674, 1996.
- C.E.J. Anderson. An overview of the theory of hydrocodes. *International Journal of Impact Engineering*, 5(1-4):33–59, 1987.
- C.E.J. Anderson and S.R. Bodner. Ballistic impact: the status of analytical and numerical modeling. *International Journal of Impact Engineering*, 7(1):9–35, 1988.
- F. Andersson, M. Aberg, and S. Jacobsson. Algorithmic approaches for studies of variable influence, contribution and selection in neural networks. *Chemometrics and Intelligent Laboratory Systems*, 51:61–72(12), 2000.
- J.K. Anlauf and M. Biehl. The adatron: an adaptive perceptron algorithm. *Europhysics Letters*, 10:687, 1989.

-
- M. Arbib. *The Handbook of Brain Theory and Neural Networks*. MIT Press, 1995.
- S. Ari and G. Saha. In search of an optimization technique for artificial neural network to classify abnormal heart sounds. *Applied Soft Computing*, 9(1):330–340, 2009.
- A. Arias, J. Rodriguez-Martinez, and A. Rusinek. Numerical simulations of impact behaviour of thin steel plates subjected to cylindrical, conical and hemispherical non-deformable projectiles. *Engineering Fracture Mechanics*, 2007.
- J. Awerbuch and S.R. Bodner. Analysis of the mechanics of perforation of projectiles in metallic plates. *International Journal of Solids and Structures*, 10:671–684, 1974.
- M.E. Backman and W. Goldsmith. The mechanics of penetration of projectiles into targets. *International Journal of Engineering Science*, 16(1):1–99, 1978.
- I. Basheer and M. Hajmeer. Artificial neural networks: fundamentals, computing, design, and application. *Journal of Microbiological Methods*, 43:3–31(29), 2000.
- E.B. Baum and D. Haussler. What size net gives valid generalization? *Neural Computation*, pages 81–90, 1989.
- R. Bellman. *Adaptive Control Processes: A Guided Tour*. Princeton University Press, 1961.
- M.G. Bello. Enhanced training algorithms, and integrated training/architecture selection for multilayer perceptron networks. *IEEE Transactions on Neural Networks*, 3(6):864–875, 1992.
- L.M. Belue and K.W. Bauer. Determining input features for multilayer perceptrons. *Neurocomputing*, 7(2):111–121, 1995.
- T. Belytschko. On difficulty levels in non linear finite element analysis of solids. *Bulletin for the International Association for Computational Mechanics*, 2(6):8, 1996.
- G. Ben-Dor, A. Dubinsky, and T. Elperi. Ballistic impact: Recent advances in analytical modeling of plate penetration dynamics. a review. *Applied Mechanics Reviews*, 58(6):355–371, 2005.
- L. Berke and P. Hajela. Applications of artificial neural nets in structural mechanics. *Structural and Multidisciplinary Optimization*, 4(2):90–98, 1992.
- P. Bermejo, H. Joho, M. Joemon, and R. Villa. Comparison of feature construction methods for video relevance prediction. *Advances in Multimedia Modeling*. 2008.
- C. Bisagni, L. Lanzi, and S. Ricci. Optimization of helicopter subfloor components under crashworthiness requirements using neural networks. *Journal of Aircraft*, 39(2):296–304, 2002.
- C. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, USA, 1996. Diferencias entre RBF and MLP.
- C.M. Bishop. Training with noise is equivalent to tikhonov regularization. *Neural Computation*, 7:108, 1995.
-

- N. Boonyanunta and P. Zeephongsekul. Predicting the relationship between the size of training sample and the predictive power of classifiers. *Knowledge-Based Intelligent Information and Engineering Systems*, 3215:529–535, 2004.
- D. Borrajo, N. Juristo, V. Martinez, and J. Pazos. *Inteligencia Artificial. Métodos y técnicas*. Editorial Centro de Estudios Ramón Areces S.A., 1993.
- T. Borvik, M. Langseth, O. Hopperstad, and K. Malo. Ballistic penetration of steel plates - analysis and experiment. *International Journal of Impact Engineering*, 22(9-10):855–886(32), 1999.
- T. Borvik, O.S. Hopperstad, T. Berstad, and M. Langseth. Numerical simulation of plugging failure in ballistic penetration. *International Journal of Solids and Structures*, 38(34-35): 6241–6264, 2001.
- L. Breiman. Bagging predictors. *Machine Learning*, 26(2):123, 1996.
- S. Brown. Energy release protection for pressurized systems. review of studies into impact terminal ballistics. *Applied Mechanics Reviews*, 39:177–201, 1986.
- M. Burger and A.Ñeubauer. Analysis of tikhonov regularization for function approximation by neural networks. *Neural Networks*, 16(1):79– 90, 2003.
- W. Carpenter and J.-M. Barthelemy. A comparison of polynomial approximations and artificial neural nets as response surfaces. *Structural and Multidisciplinary Optimization*, 5(3):166–174, 1993.
- A. Cauchy. Méthodes générales pour la résolution des systèmes d'équations simultanées. *C. R. Acad. Sci. Paris*, 25:536–538, 1847.
- K. Chandrashekhara, A. Okafor, and Y. Jiang. Estimation of contact force on composite plates using impact-induced strain and neural networks. *Composites Part B: Engineering*, 29:363–370(8), 1998.
- W. Cochran. *Sampling Techniques*. John Wiley, 1977.
- R. Cortes, C.Ñavarro, M. Martinez, J. Rodriguez, and V. Sanchez-Galvez. Numerical modelling of normal impact on ceramic composite armours. *International Journal of Impact Engineering*, 12(4):639–651, 1992.
- M. Cover. Learning in pattern recognition. *Methodologies of Pattern Recognition*, pages 111–132, 1969.
- T.M. Cover. Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition. *Electronic Computers, IEEE Transactions on*, EC-14 (3):326–334, 1965.
- P. Crowther and R. Cox. A method for optimal division of data sets for use in neural networks. In *International Conference on Knowledge-Based Intelligent Information and Engineering Systems*, 2005.
- P. Crowther and R. Cox. *Accuracy of Neural Network Classifiers as a Property of the Size of the Data Set*, volume 4253, pages 1143–1149. 2006.

-
- G. Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals, and Systems*, 2:303–314, 1989.
- C. Darwin. On the origin of species: By means of natural selection or the preservation of favoured races in the struggle for life. 1859.
- J. Demsar. Statistical comparisons of classifiers over multiple data sets. *J. Mach. Learn. Res.*, 7:1–30, 2006.
- J. Denker, D. Schwartz, B. Wittner, S.A. Soils, R. Howard, L. Jackel, and J. Hopfield. Large automatic learning, rule extraction, and generalization. *Complex Systems*, 1:877–922, 1987.
- F. Despagne and D.-L. Massart. Variable selection for neural networks in multivariate calibration. *Chemometrics and Intelligent Laboratory Systems*, 40(2):145–163, 1998.
- P.A. Devijver and J. Kittler. *Pattern Recognition: A Statistical Approach*. Prentice Hall, 1982.
- T.G. Dietterich. Approximate statistical tests for comparing supervised classification learning algorithms. *Neural Computation*, 10:1895–1923, 1998.
- Y. Dimopoulos. Use of some sensitivity criteria for choosing networks with good generalization ability. *Neural processing letters*, 2(6):1, 1995.
- P. Domingos. The role of occam’s razor in knowledge discovery. *Data Min. Knowl. Discov.*, 3(4):409–425, 1999.
- R. Dua, S.E. Watkins, D.C. Wunsch, K. Chandrashekhara, and F. Akhavan. Detection and classification of impact-induced damage in composite plates using neural networks. In *International Joint Conference On Neural Networks (IJCNN 01)*, pages 681–686, 2001.
- B. Efron. Bootstrap methods: Another look at the jackknife. *Annals of Statistics*, 7(1):1–26, 1979.
- B. Efron. Estimating the error rate of a prediction rule: Improvement on cross-validation. *Journal of the American Statistical Association*, 78(382):316–331, 1983.
- B. Efron and Robert J. Tibshirani. *An Introduction to the Bootstrap*. Chapman & Hall/CRC, may edition, 1994. Hardcover.
- M. El-Gindy and L. Palkovics. Possible application of artificial neural networks to vehicle dynamics and control: a literature review. *International Journal of Vehicle Design*, 14(5):592, 1993.
- S.A. Elias and S. Grossberg. Pattern formation, contrast control, and oscillations in short-term memory of shunting on-center off-surround networks. *Biological Cybernetics*, 20:69–98, 1975.
- N.F.O. Evbuomwan, S. Sivaloganathan, and A. Jebb. A survey of design philosophies, models, methods and systems. *Proceedings of the Institution of Mechanical Engineers. Part B. Journal of engineering manufacture*, 210(4):301–320, 1996.
- S.E. Fahlman. Faster-learning variations on back-propagation: an empirical study. *Connectionist Models Summer School*, pages 38–51, 1988.
-

- X. Fang, H. Luo, and J. Tang. Structural damage detection using neural network with learning rate improvement. *Computers & Structures*, 83(25-26):2150–2161, 2005.
- C. Farrar and D. Leeming. *Military Ballistics: A Basic Manual*. Brassey’s Defence Publishers, 1983.
- G.B.R. Feilden. *Engineering design ('The Feilden’s Report')*. Report of Royal Commission. HMSO, 1963.
- C.J. Feng, Z.S. Yu, U. Kingi, and M.P. Baig. Threefold vs. fivefold cross validation in one-hidden-layer and two-hidden-layer predictive neural network modeling of machining surface roughness data. *Journal of Manufacturing Systems*, 24(2):93–107, 2005.
- D. Fernandez. *Desarrollo de una nueva herramienta basada en redes neuronales artificiales para el diseño de protecciones ligeras cerámica-metal frente al impacto de alta velocidad*. PhD thesis, Universidad Carlos III de Madrid, 2007.
- D. Fernandez, J. Lopez-Puente, and R. Zaera. Prediction of the behaviour of cfrps against high-velocity impact of solids employing an artificial neural network methodology. *Composites Part A: Applied Science and Manufacturing*, 39(6):989–996, 2008.
- C. Ferri, J. Hernández-Orallo, and R. Modroiu. An experimental comparison of performance measures for classification. *Pattern Recogn. Lett.*, 30(1):27–38, 2009.
- S. Finger and J. Dixon. A review of research in mechanical engineering design. part i: Descriptive, prescriptive, and computer-based models of design processes. *Research in Engineering Design*, 1(1):51–67, 1989.
- R.A. Fisher. *Statistical methods for research workers*. Oliver & Boyd, Edinburgh :, 11th ed.(rev.) edition, 1950.
- A. Fiszlelew. *Generación automática de redes neuronales con ajuste de parámetros basado en algoritmos genéticos*. PhD thesis, Universidad de Buenos Aires, 2002.
- I. Flood and N. Kartam. Neural networks in civil engineering. i: Principles and understanding. *Journal of Computing in Civil Engineering*, 8(2):131–148, 1994.
- D.B. Fogel. An information criterion for optimal neural network selection. *Signals, Systems and Computers, 1990. 1990 Conference Record Twenty-Fourth Asilomar Conference on*, 2: 998, 1990.
- D. Foley. Considerations of sample and feature size. *Information Theory, IEEE Transactions on*, 18(5):618–626, 1972.
- G. Foody, M.B. McCulloch, and W.B. Yates. The effect of training set size and composition on artificial neural network classification. *International Journal of Remote Sensing*, 16(9): 1707–1723, 1995.
- L. Franco, J. Jerez, and J. Bravo. Role of function complexity and network size in the generalization ability of feedforward networks. *Computational Intelligence and Bioinspired Systems*, 3512:1–8, 2005.

-
- R.R. Franzen, D.L. Orphal, and C. E Anderson. The influence of experimental design on depth-of-penetration (dop) test results and derived ballistic efficiencies. *International Journal of Impact Engineering*, 19(8):727–737, 1997.
- Y. Freund and R.E. Schapire. Experiments with a new boosting algorithm. In *Thirteenth International Conference on Machine Learning*, pages 148–156, 1996.
- L. Fu. *Neural Networks in Computer Intelligence*. McGraw-Hill, 1995.
- K. Funahashi. On the approximate realization of continuous mappings by neural networks. *Neural Networks*, 2:183–192, 1989.
- S. Garcia and F. Herrera. An extension on statistical comparisons of classifiers over multiple data sets for all pairwise comparisons. *Journal of Machine Learning Research*, 9:2677–2694, 2008.
- A. Garcia-Crespo, B. Ruiz-Mezcua, D. Fernandez, and R. Zaera. Prediction of the response under impact of steel armours using a multilayer perceptron. *Neural Computing & Applications*, 16(2):147–154, 2006.
- A. Garcia-Crespo, B. Ruiz-Mezcua, I. Gonzalez-Carrasco, and J. Lopez-Cuadrado. Multilayer perceptron training optimization for high speed impacts classification. volume 39 of *Advances in Electrical Engineering and Computational Science*, pages 377–388. Springer, 2008.
- G.D. Garson. Interpreting neural-network connection weights. *AI Expert*, 6(4):46–51, 1991.
- M. Gevrey, D. I., and S. Lek. Review and comparison of methods to study the contribution of variables in artificial neural network models. *Ecological Modelling*, 160:249–264(16), 2003.
- F. Girosi, M. Jones, and T. Poggio. Regularization theory and neural networks architectures. *Neural Computation*, 7:219, 1995.
- A.T. Goh. Some civil engineering applications of neural networks. *Proceedings of the Institution of Civil Engineers: Structures and Buildings.*, 104(4):463–469, 1994.
- W. Goldsmith. *Impact: The theory and physical behaviour of colliding solids*. Edward Arnold Publishers Ltd., 1960.
- W. Goldsmith and S.A. Finnegan. Penetration and perforation processes in metal targets at and above ballistic velocities. *International Journal of Mechanical Sciences*, 13(10):843–866, 1971.
- H. Gomes and A. Awruch. Comparison of response surface and neural network with other methods for structural reliability analysis. *Structural Safety*, 26(1):49–67, 2004.
- I. Gonzalez-Carrasco, A. Garcia-Crespo, B. Ruiz-Mezcua, and J. Lopez-Cuadrado. Neural network application for high speed impacts classification. In *World Congress on Engineering*, 2008.
- I. Gonzalez-Carrasco, A. Garcia-Crespo, and Lopez-Cuadrado J.L. Ruiz-Mezcua, B. Dealing with limited data in ballistic impact scenarios: An empirical comparison of different neural network approaches. *Applied Intelligence*, (in press), 2009.
-

- C. Goutte. Note on free lunches and cross-validation. *Neural Computation*, 9(6):1245–1249, 1997.
- M. Grochowski and W. Dutch. Learning highly non-separable boolean functions using constructive feedforward neural network. *Artificial Neural Networks. ICANN 2007*. Springer, 2007.
- S. Grossberg. Nonlinear neural networks: Principles, mechanisms, and architectures. *Neural Networks*, 1(1):17–61, 1988.
- M. Guinea, A. Sanchis, and J. Molina. Evolución de gramáticas bidimensionales de contexto libre para el diseño de arquitecturas de redes de neuronas artificiales. *Inteligencia artificial: Revista Iberoamericana de Inteligencia Artificial*, 17:33–48, 2002.
- G. Gutierrez, P. Isasi, J. Molina, A. Sanchis, and I. Galvan. Evolutionary cellular configurations for designing feed-forward neural networks architectures. *Connectionist Models of Neurons, Learning Processes, and Artificial Intelligence*. Springer, 2001.
- M. Hagan, H. Demuth, and M. Beale. *Neural Network Design*. PWS Publishing Co., 1996.
- P. Hajela and L. Berke. Neural networks in structural analysis and design - an overview. *Computing Systems In Engineering*, 3(1-4):525–538, 1992.
- P. Hajela and E. Lee. Topological optimization of rotorcraft subfloor structures for crashworthiness considerations. *Computers structures*, 64(1-4):65, 1997.
- L.K. Hansen and P. Salamon. Neural network ensembles. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 12(10):993–1001, 1990.
- C. Harpham and C.W. Dawson. The effect of different basis functions on a radial basis function network for time series prediction: A comparative study. *Neurocomputing*, 69(16-18):2161–2170, 2006. Brain Inspired Cognitive Systems - Selected papers from the 1st International Conference on Brain Inspired Cognitive Systems (BICS 2004).
- M. Hassoun. *Fundamentals of Artificial Neural Networks*. The MIT Press, 1995.
- S. Haykin. *Neural Networks: A Comprehensive Foundation*. Prentice Hall PTR, 1994.
- Y. He and Y. Sun. Neural network-based l1-norm optimisation approach for fault diagnosis of nonlinear circuits with tolerance. *Circuits, Devices and Systems, IEE Proceedings -*, 148(4):223–228, 2001.
- R. Hecht-Nielsen. Kolmogorov’s mapping neural network existence theorem. *IEEE First Annual International Conference on Neural Networks*, 1987.
- R. Hecht-Nielsen. *Neurocomputing*. Addison-Wesley, 1990.
- J. Henseler. Back propagation. *Artificial Neural Networks: An Introduction to ANN Theory and Practice*. Springer, 1995.
- G.F. Hepner, T. Logan, Ritter, and N. Bryant. Artificial neural network classification using a minimal training set: Comparison to conventional supervised classification. *Photogramm. Eng. Remote Sensing*, 56(4):469–471, 1990.

-
- I. Herle. Difficulties related to numerical predictions of deformations. *Analyzing Uncertainty in Civil Engineering*. Springer, 2005.
- Hernandez-Lopez. *Predicción y optimización de emisores y consumo mediante redes neuronales en motores diesel*. Editorial Reverte, 2006.
- J. Hertz, A. Krogh, and R. Palmer. *Introduction to the Theory of Neural Computation*. Addison-Wesley Publishing Company, 1991.
- R. Hestenes and E. Stiefel. Methods of conjugate gradients for solving linear systems. *Journal of Research of the National Bureau of Standards*, 49:409–436, 1952.
- A. Highsmith and S. Keshav. Using measured damage parameters to predict the residual strength of impacted composites: A neural network approach. *Journal of Composites Technology and Research*, 19(4):203–234, 1997.
- J. Hilera and J. Martinez. *Redes neuronales artificiales : fundamentos, modelos y aplicaciones*. Ra-Ma, 1995.
- G.E. Hinton and D. Van-Camp. Keeping the neural networks simple by minimizing the description length of the weights. In *COLT '93: Proceedings of the sixth annual conference on Computational learning theory*, pages 5–13. ACM, 1993.
- H.S. Hippert, C.E. Pedreira, and R.C. Souza. Neural networks for short-term load forecasting: a review and evaluation. *Power Systems, IEEE Transactions on*, 16(1):44–55, 2001.
- J.H. Holland. *Adaptation in natural and artificial systems*. MIT Press, 1975.
- L. Holmstrom and P. Koistinen. Using additive noise in back-propagation training. *Neural Networks, IEEE Transactions on*, 3(1):24–38, 1992.
- K. Hornik, M. Stinchcombe, and H. White. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5):359–366, 1989.
- A. Hosni, E. Mesbahi, and Y. Pu. Reliability analysis of structures using neural network method. *Probabilistic Engineering Mechanics*, 21(1):44–53, 2006.
- S.C. Huang and Y.F. Huang. Bounds on the number of hidden neurons in multilayer perceptrons. *IEEE Transactions on Neural Networks*, 2(1):47–55, 1991.
- R. Ince. Prediction of fracture parameters of concrete by artificial neural networks. *Engineering Fracture Mechanics*, 71(15):2143–2159, 2004.
- P. Isasi and I. Galvan. *Redes de neuronas artificiales: un enfoque práctico*. Pearson Prentice Hall, 2004.
- R.A. Jacobs. Increased rates of convergence through learning rate adaptation. *Neural Networks*, 1:295–307, 1988.
- W.M. Jenkins. An introduction to neural computing for the structural engineer. *The structural engineer*, 75(3):38–41, 1997.
-

- W.M. Jenkins. A neural network for structural re-analysis. *Computers and Structures*, 72: 687–698(12), 1999.
- S. Jiang, C. Zhang, and C. Koh. Structural damage detection by integrating data fusion and probabilistic neural network. *Advances in Structural Engineering*, 9:445–458(14), 2006.
- G. Johnson and W. Cook. A constitutive model and data for metals subjected to large strains, high strain rates, and temperatures. In *International Symposium Ballistics*, 1983.
- G.H. Jonas and J.A. Zukas. Mechanics of penetration: Analysis and experiment. *International Journal of Engineering Science*, 16(11):879–903, 1978.
- R.T. Jones, J.S. Sirkis, E.J. Friebele, and A.D. Kersey. Location and magnitude of impact detection in composite plates using neural networks. In *Smart Structures and Materials 1995: Smart Sensing, Processing, and Instrumentation*, pages 469–480, 1995.
- G.N. Karystinos and D.A. Pados. On overfitting, generalization, and randomly expanded training sets. *Neural Networks, IEEE Transactions on*, 11(5):1050–1057, 2000.
- J.P. Kharoufeh. Integrated feature and architecture selection for radial basis neural networks. *International Journal of Smart Engineering System Design*, 5:507–516(10), 2003.
- P.C. Kiessler. Probability and statistics for engineers and scientists (3rd ed.). anthony j. hayter. *The American Statistician*, 60:343–343, 2006.
- S. Kirkpatrick, C.D. Gelatt, and M.P. Vecchi. Optimization by simulated annealing. *Science*, 220:671, 1983.
- F. Klassner. Artificial intelligence: introduction. *Crossroads*, 3(1):2, 1996.
- R. Kohavi. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Proceedings of International Joint Conference on Artificial Intelligence*, pages 1137–1143. Morgan Kaufmann, 1995.
- T. Kohonen. An introduction to neural computing. *Neural Networks*, 1(1):3–16, 1988.
- P. Koistinen and L. Holmstrom. Kernel regression and backpropagation training with noise. In *Neural Networks, 1991. 1991 IEEE International Joint Conference on*, pages 367–372 vol.1, 1991.
- A.N. Kolmogorov. On the representation of continuous functions of many variables by superposition of continuous functions of one variable and addition. *Dokl. Akad. Nauk USSR*, 144:953–956, 1957.
- S.B. Kotsianti and D. Kanellopoulos. Combining bagging, boosting and dagging for classification problems. volume 4693 of *Knowledge-Based Intelligent Information and Engineering Systems*. 2007.
- A. Kramer and A. Sangiovanni-Vincentelli. Efficient parallel learning algorithms for neural networks. *Advances in neural Information Processing Systems*, 1:40–48, 1989.
- W.H. Kruskal and W.A. Wallis. Use of ranks in one-criterion variance analysis. *Journal of the American Statistical Association*, 47(260):583–621, 1952.

-
- J. Kudva, N. Munir, and P. Tan. Damage detection in smart structures using neural networks and finite-element analyses. *Smart Materials and Structures*, 1:108–112(5), 1992.
- S.Y. Kung and J.N. Hwang. An algebraic projection analysis for optimal hidden units size and learning rate in backpropagation learning. *IEEE International Conference on Neural Networks*, 1:363–370, 1988.
- H.K. Lam, S.H. Ling, F.H. Leung, and P.K. Tam. Tuning of the structure and parameters of neural network using an improved genetic algorithm. *Industrial Electronics Society, 2001. IECON '01. The 27th Annual Conference of the IEEE*, 1:25–30 vol.1, 2001.
- L. Langseth, O. Hopperstad, and T. Berstad. Crashworthiness of aluminium extrusions: validation of numerical simulation, effect of mass ratio and impact velocity. *International Journal of Impact Engineering*, 22(9-10):829–854, 1999.
- L. Lanzi, C. Bisagni, and S. Ricci. Neural network systems to reproduce crash behavior of structural components. *Computers structures*, 82(1):93, 2004.
- A. Lapedes and R. Farber. How neural nets work. *Neural Information Processing Systems*, pages 442–456, 1988.
- S. Lawrence, C.L. Giles, and A.C. Tsoi. What size neural network gives optimal generalization? convergence properties of backpropagation. Technical report, 1996.
- Y. Le Cun, R. Pfeifer, Z. Schreter, F. Fogelman-Soulie, L. Steels, R. Pfeifer, Z. Schreter, F. Fogelman-Soulie, and L. Steels. Generalization and network design strategies. In *Connectionism in Perspective*, pages 143–155, 1989.
- Y. Le Cun, J.S. Denker, and A. Optimal brain damage. In *Advances in Neural Information Processing*, pages 598–605, 1990.
- J. LeClerc, K. Worden, W. Staszewski, and J. Haywood. Impact detection in an aircraft composite panel. a neural-network approach. *Journal of Sound and Vibration*, 299(3):672–682, 2007.
- E. Leclercq, F. Druaux, D. Lefebvre, and S. Zerkaoui. Autonomous learning algorithm for fully connected recurrent networks. *Neurocomputing*, 63:25– 44, 2005. New Aspects in Neurocomputing: 11th European Symposium on Artificial Neural Networks.
- C. Lefebvre, C. Fancourt, J. Principe, and G. J. *NeuroSolutions Documentation*. 2007.
- S. Lek, A. Belaud, I. Dimopoulos, J. Lauga, and J. Moreau. Improved estimation, using neural networks, of the food consumption of fish populations. *Marine and Freshwater Research.*, 468: 1229–1236, 1995.
- K. Levenberg. A method for the solution of certain non-linear problems in least squares. *Quarterly Journal of Applied Mathematics*, II(2):164–168, 1944.
- H. Levene. Robust tests for equality of variances. *Contributions to probability and statistics: Essays in honor of Harold Hotelling*, pages 278–292, 1960.
-

- J. Levy-Mangin, R. Florez-Lopez, and J. Fernandez-Fernandez. *Las Redes Neuronales Artificiales. Fundamentos Teóricos y Aplicaciones Prácticas*. Metodología y Análisis de Datos en Ciencias Sociales. Netbiblo, 2008.
- R. Lippmann. An introduction to computing with neural nets. *ASSP Magazine, IEEE [see also IEEE Signal Processing Magazine]*, 4(2):4–22, 1987.
- G. Liu and M. Liu. *Smoothed Particle Hydrodynamics*. World Scientific, 2003.
- H. Liu and R. Setiono. Incremental feature selection. *Applied Intelligence*, 9(3):217–230, 1998.
- S. Liu, J. Huang, J. Sung, and C. Lee. Detection of cracks using neural networks and computational mechanics. *Computer methods in applied mechanics and engineering*, 191(25–26):2831, 2002.
- Y. Liu. Create stable neural networks by cross-validation. In *IJCNN '06. International Joint Conference on Neural Networks*, pages 3925–3928, 2006.
- X. Liuji, X. Jiandong, W. Shizhong, Z. Songmin, Z. Yongzhen, and L. Rui. Use of artificial neural network in predicting mechanical properties of high-speed steel (hss). In *Proceedings of the 2006 IEEE International Conference on Mechatronics and Automation*, pages 1872–1877, 2006.
- G.R. Lloyd, K. Wongravee, C.J. Silwood, M. Grootveld, and R.G. Brereton. Self organising maps for variable selection: Application to human saliva analysed by nuclear magnetic resonance spectroscopy to investigate the effect of an oral healthcare product. *Chemometrics and Intelligent Laboratory Systems*, 98(2):149–161, 2009.
- C. Looney. Advances in feedforward neural networks: demystifying knowledge acquiring black boxes. *Transactions on Knowledge and Data Engineering*, 8(2):211–226, 1993.
- D.J.C. MacKay. A practical bayesian framework for backpropagation networks. *Neural Comput.*, 4(3):448–472, 1992.
- R. Maclin and J.W. Shavlik. Combining the predictions of multiple classifiers: Using competitive learning to initialize neural networks. In *Fourteenth International Joint Conference on Artificial Intelligence*, pages 524–530, 1995.
- M. Majumder, P. Roy, and A. Mazumdar. Optimization of the water use in the river damodar in west bengal in india: An integrated multi-reservoir system with the help of artificial neural network. *Engineering Computing and Architecture*, 1(2), 2007.
- J. Malcolm. *Terminal Ballistics: A Text and Atlas of Gunshot Wounds*. 2005.
- F. Manzano-Trovamala, M. Guerrero, and V. Arcaute. Balística: Balística de efectos o balística de las heridas. *Cirujano General*, 23(4), 2001.
- A. Maren, C. Harston, and R. Pap. *Handbook of Neural Computing Applications*. Handbook of Neural Computing Applications, 1990.
- D.W. Marquardt. An algorithm for least-squares estimation of nonlinear parameters. *SIAM Journal on Applied Mathematics*, 11(2):431–441, 1963.

-
- B. Martin-del Brio and A. Sanz-Molina. *Redes neuronales y sistemas borrosos*. RA-MA, 2006.
- T. Masters. *Practical Neural Network Recipes in C++*. Academic Press, 1994.
- .T. Masters. *Advanced algorithms for neural networks: a C++ sourcebook*. John Wiley and Sons, Inc., 1995.
- K. Matsuoka. Noise injection into inputs in back-propagation learning. *Systems, Man and Cybernetics, IEEE Transactions on*, 22(3):436–440, 1992.
- W.S. McCulloch and W. Pitts. A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, 5:115–133, 1943.
- A. Mimaroglu, O. Iyibilgin, and H. Unal. Ballistic penetration into targets: use of f.e. technique. In *MS'06: Proceedings of the 17th IASTED international conference on Modelling and simulation*, pages 579–584. ACTA Press, 2006.
- A.A. Minai and R.D. Williams. Back-propagation heuristics: a study of the extended delta-bar-delta algorithm. *Neural Networks, 1990., 1990 IJCNN International Joint Conference on*, pages 595–600 vol.1, 1990.
- M. Minsky and S. Papert. *Perceptrons: An Introduction to Computational Geometry*. The MIT Press, 1969.
- T.M. Mitchell. *Machine Learning*. McGraw-Hill, 1997.
- A. Moghaddamia, R. Remesan, M.H. Kashani, M. Mohammadi, D. Han, and J. Piri. Comparison of llr, mlp, elman, nnarx and anfis models—with a case study in solar radiation estimation. *Journal of Atmospheric and Solar-Terrestrial Physics*, 71(8-9):975–982, 2009.
- J. Moreno and A. Pol. Numeric sensitivity analysis applied to feedforward neural networks. *Neural Computing & Applications*, 12(2):119–125, 2003.
- J. Moreno, A. Pol, and C. Provencio. Redes neuronales artificiales: abriendo la caja negra. *Metodologia de las ciencias del comportamiento*, 4(1):77–93, 2002.
- N. Morgan and H. Bourlard. Generalization and parameter estimation in feedforward nets: Some experiments. *Advances in Neural Information Processing System*, 2:630–637, 1990.
- I. Mrazova and D. Wang. Improved generalization of neural classifiers with enforced internal representation. *Neurocomputing*, 70(16-18):2940–2952, 2007.
- N. Murata, S. Yoshizawa, and S.-i. Amari. Network information criterion — determining the number of hidden units for an artificial neural network model. *IEEE Transactions on Neural Networks*, 5:865, 1994.
- R.H. Myers and D.C. Montgomery. *Response Surface Methodology: Process and Product in Optimization Using Designed Experiments*. John Wiley & Sons, Inc., 1995.
- N.K. Naik and P. Shirao. Composite structures under ballistic impact. *Composite Structures*, 66(1-4):579–590, 2004.
-

- N.K. Naik, P. Shirao, and B. Reddy. Ballistic impact behaviour of woven fabric composites: Parametric studies. *Materials Science and Engineering*, 412(1-2):104–116, 2005.
- M.Ñelson and W. Illingworth. *A practical guide to neural nets*. Addison-Wesley Publishing Co., 1990.
- D.Ñguyen and B. Widrow. Improving the learning speed of two-layer neural networks by choosing initial values of the adaptive weights. *IEEE International Conference on Neural Networks*, 3: 21–26, 1990.
- L.I. Nord and S.P. Jacobsson. A novel method for examination of the variable contribution to computational neural network models. *Chemometrics and Intelligent Laboratory Systems*, 44 (1-2):153– 160, 1998.
- C. Oguibe and D. Webb. Finite-element modelling of the impact response of a laminated composite plate. *Composites Science and Technology*, 59(12):1913–1922, 1999.
- J.D. Olden and D.A. Jackson. Illuminating the "black box": a randomization approach for understanding variable contributions in artificial neural networks. *Ecological Modelling*, 154: 135–150(16), 2002.
- J.D. Olden, M.K. Joy, and R.G. Death. An accurate comparison of methods for quantifying variable importance in artificial neural networks using simulated data. *Ecological Modelling*, 178(3-4):389–397, 2004.
- T. Omar, A. Eskandarian, and N. Bedewi. Vehicle crash modelling using recurrent neural networks. *Mathematical and Computer Modelling*, 28(9):31, 1998.
- T. Omar, A. Eskandarian, and N. Bedewi. Artificial neural networks for modelling dynamics of impacting bodies and vehicles. *Proceedings of the Institution of Mechanical Engineers. Part A, Journal of Power and Energy*, 214(3):133, 2000.
- D. Opitz and R. Maclin. Popular ensemble methods: an empirical study. *Journal of Artificial Intelligence Research*, 11:169, 1999.
- Y. Pao. *Adaptive pattern recognition and neural networks*. Addison-Wesley, 1989.
- S. Papadokostantakis, A. Lygeros, and S.P. Jacobsson. Comparison of recent methods for inference of variable influence in neural networks. *Neural Netw.*, 19(4):500–513, 2006.
- D. Parker. Learning logic, invention report. Technical report, Office of Technology Licensing, Stanford, 1982.
- C. Perez-Lopez. *Técnicas de análisis multivariante de datos aplicaciones con SPSS*, volume 1. Pearson Educación, 2004.
- K. Priddy and P.E. Keller. *Artificial Neural Networks: An Introduction*. SPIE Press, 2005.
- J.C. Principe, N.R. Euliano, and W.C. Lefebvre. *Neural and Adaptive Systems: Fundamentals through Simulations with CD-ROM*. John Wiley & Sons, Inc., 1999.
- Y. Pu and E. Mesbahi. Application of artificial neural networks to evaluation of ultimate strength of steel panels. *Engineering structures*, 28(8):1190, 2006.

-
- J. Radin and W. Goldsmith. Normal projectile penetration and perforation of layered targets. *International Journal of Impact Engineering*, 7(2):229–259, 1988.
- M.Y. Rafiq, G. Bugmann, and D.J. Easterbrook. Neural network design for engineering applications. *Computers and Structures*, 79(17):1541–1552, 2001.
- S. Ramon-y Cajal. The croonian lecture: La fine structure des centres nerveux. *Proceedings of the Royal Society of London*, 55:444–468, 1894.
- M. Ravid and S. Bodner. Dynamic perforation of viscoplastic plates by rigid projectiles. *International Journal of Engineering Science*, 21:577–591, 1983.
- R. Reed. Pruning algorithms-a survey. *IEEE Transactions on Neural Networks*, 4(5):740–747, 1993.
- R. Reed, O.h. Seho, and R.J.M. Ii. Regularization using jittered training data. In *In Proceedings IJCNN 92 International Joint Conference on Neural Networks*, pages 147–152, 1992.
- G. Reklaitis, A. Ravindran, and K. Ragsdell. *Engineering Optimization: Methods and Applications*. Wiley-Interscience, 1983.
- A. Remennikov and T.A. Rose. Predicting the effectiveness of blast wall barriers using neural networks. *International Journal of Impact Engineering*, 34(12):1907–1923, 2007.
- J. Rissanen. Modeling by shortest data description. *Automatica*, 14:445–471, 1978.
- M. Roseblatt. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65:386–408, 1958.
- Z. Rosenberg and E. Dekel. On the role of material properties in the terminal ballistics of long rods. *International Journal of Impact Engineering*, 30(7):835–851, 2004.
- D. Rumelhart, G. Hinton, and R. Williams. Learning internal representations by back-propagating errors. *Nature*, 323:533–536, 1986.
- K. Saridakis and A. Dentsoras. Soft computing in engineering design - a review. *Advanced Engineering Informatics*, 22(2):202–221, 2008.
- W. Sarle. Stopped training and other remedies for overfitting. In *27th Symposium on Interface*, 1995.
- W. Sarle. *How to measure importance of inputs?*, 2000. in <ftp://ftp.sas.com/pub/neural/importance.html>.
- R. Setiono and H. Liu. Neural-network feature selector. *IEEE Transactions on Neural Networks*, 8(3):654–662, 1997.
- S. Shao and Y. Murotsu. Structural reliability analysis using a neural network. *JSME international journal*, 40(3):242–246, 1997.
- Q. Shi and I. Hagiwara. Optimal design method to automobile problems using holographic neural network’s approximation. *Japan Journal of Industrial and Applied Mathematics*, 17(3):321–339, 2000.
-

- J. Sietsma and R. Dow. Neural net pruning-why and how. In *IEEE International Conference on Neural Networks*, pages 325–333, 1988.
- J. Sietsma and R. Dow. Creating artificial neural networks that generalize. *Neural Networks*, 4(1):67–79, 1991.
- S. Silling and M. Forrestal. Mass loss from abrasion on ogive-nose steel projectiles that penetrate concrete targets. *International journal of impact engineering*, 34:1814–1820, 2007.
- U. Siripatrawan. Self-organizing algorithm for classification of packaged fresh vegetable potentially contaminated with foodborne pathogens. *Sensors and Actuators B: Chemical*, 128(2):435–441, 2008.
- G.W. Snedecor and W.G. Cochran. *Statistical Methods*. Iowa State University Press, 1980.
- J. Sobieszczanski-sobieski and R.T. Haftka. Multidisciplinary aerospace design optimization: survey of recent developments. *Structural Optimization*, 14:1, 1997.
- M. Sokolova, R. Rasras, and D. Skopin. The artificial neural network based approach for mortality structure analysis. *American Journal of Applied Science*, 3(2):1698–1702, 2006.
- J. Sola and J. Sevilla. Importance of input data normalization for the application of neural networks to complex industrial problems. *IEEE Transactions on Nuclear Science*, 44(3):1464–1468, 1997.
- X.-H. Song and Y.u. Ru-Qin. Artificial neural networks applied to the quantitative structure–activity relationship study of dihydropteridine reductase inhibitors. *Chemometrics and Intelligent Laboratory Systems*, 19(1):101–109, 1993.
- L.u. Songwu, S. Member, and T. Basar. Robust nonlinear system identification using neural network models. *IEEE Transactions on Neural Networks*, 9:407, 1998.
- D. Sridhar, E. Bartlett, and R. Seagrave. Information theoretic subset selection for neural network models. *Computers & Chemical Engineering*, 22:613–626(14), 1998.
- F. Stasa. *Applied Finite Element Analysis for Engineers*. Oxford University Press, 1984.
- R. Stein. Seleting data for neural networks. *AI Expert*, 8(2):42–47, 1993.
- M. Stone. Cross-validatory choice and assessment of statistical predictions. *Journal of the Royal Statistical Society B*, 36(1):111–147, 1974.
- C. Sun and S. Potti. A simple model to predict residual velocities of thick composite laminates subjected to high velocity impact. *International Journal of Impact Engineering*, 18(3):339–353, 1996.
- A.H. Sung. Ranking input importance in neural network modeling of engineering problems. *Neural Networks Proceedings, 1998. IEEE World Congress on Computational Intelligence. The 1998 IEEE International Joint Conference on*, 1:316–321 vol.1, 1998.
- D. Sung, J. Oh, C. Kim, and C. Hong. Impact monitoring of smart composite laminates using neural network and wavelet analysis. *Journal of Intelligent Material Systems and Structures*, 11(3):180–190, 2000.

-
- R. Swift and S. Batill. Application of neural networks to preliminary structural design. In *Structures, Structural Dynamics and Materials Conference*, pages 335–343, 1991.
- K. Swingler. *Applying Neural Networks. A Practical Guide*. Academic Press, 1996.
- J. Takeuchi and Y. Kosugi. Neural network representation of finite element method. *Neural Networks*, 7(2):389–395, 1993.
- B. Tang, W. Hsieh, A. Monahan, and F. Tangang. Skill comparisons between neural networks and canonical correlation analysis in predicting the equatorial pacific sea surface temperatures. *Journal of Climate*, 13:287–293, 2000.
- L. Tarassenko. *A guide to neural computing applications*. Arno / NCAF, 1998.
- T. Tchaban, M. Taylor, and J.P. Griffin. Establishing impacts of the inputs in a feedforward neural network. *Neural Computing & Applications*, 7(4):309–317, 1998.
- .R. Tibshirani. A comparison of some error estimates for neural network models. *Neural Comput.*, 8(1):152–163, 1996.
- N. Ueda. Optimal linear combination of neural networks for improving classification performance. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(2):207–215, 2000.
- R. Vanluchene and R. Sun. Microcomputers in civil engineering. *MicroComp. CIV. Eng.*, 5(3): 207–215, 1990.
- V.N. Vapnik. *The Nature of Statistical Learning Theory*. Springer-Verlag, 1995.
- V.N. Vapnik. *Statistical Learning Theory*. John Wiley & Sons, Inc., 1998.
- D. Viechnicki, M. Slavin, and K. M.I. Development and current status of armor ceramics. *Ceramic Bulletin*, 70(6):1035–1039, 1991.
- L. Wang, M. Xu, J. Zhu, and S. Shi. A method of combined shpb technique and bp neural network to study impact response of materials. *Strain*, 42(3):149–158, 2006.
- Z. Waszczyszyn and L. Ziemianski. Neural networks in mechanics of structures and materials - new results and prospects of applications. *Computers and Structures*, 79(16):2261–2276, 2001.
- S.E. Watkins, F. Akhavan, R. Dua, K. Chandrashekhara, and D.C. Wunsch. Impact-induced damage characterization of composite plates using neural networks. *Smart Materials and Structures*, 16:515–524(10), 2007.
- R.L. Watrous. Learning algorithms for connectionist networks: Applied gradient methods of nonlinear optimization. *First IEEE International Conference on Neural Networks*, 2:619–627, 1987.
- A.S. Weigend, B.A. Huberman, and D.E. Rumelhart. Predicting the future: A connectionist approach. *Stanford PDP Research Group Report*, pages 90–91, 1990.
- .S.M. Weiss and .C.A. Kulikowski. *Computer systems that learn: classification and prediction methods from statistics, neural nets, machine learning, and expert systems*. Morgan Kaufmann Publishers Inc., 1991.
-

- H. Wen. Penetration and perforation of thick frp laminates. *Composites Science and Technology*, 61(8):1163–1172, 2001.
- D. West. Neural network credit scoring models. *Computers & Operations Research*, 27(11-12): 1131– 1152, 2000.
- B. Widrow. An adaptive ádalineñeuron using chemical 'memistors'. *Stanford Electronics Laboratories Technical Report*, 1553(2), 1960.
- B. Widrow. 30 years of adaptive neural networks: perceptron, madaline, and backpropagation. *Proceedings of the IEEE* 78, 9:1415–1442, 1990.
- B. Widrow and E. Hoff. Adaptive switch circuits. *IRE WESCON Convention Record*, 55(4): 96–104, 1959.
- F. Wilcoxon. Individual comparisons by ranking methods. *Biometrics Bulletin*, 1(6):80–83, 1945.
- M. Wilkins. Mechanics of penetration and perforation. *International Journal of Engineering Science*, 16(11):793–807, 1978.
- K. Worden and W.J. Staszewski. Impact location and quantification on a composite panel using neural networks and a genetic algorithm. *Strain*, 36(2):61– 70, 2000.
- B.J. Wythoff. Backpropagation neural networks: A tutorial. *Chemometrics and Intelligent Laboratory Systems*, 18:115–155, 1993.
- S. Xu and L. Chen. A novel approach for determining the optimal number of hidden layer neurons for fnns and its application in data mining. In *5th International Conference on Information Technology and Applications*, pages 23–26, 2008.
- J. Yang and V. Honavar. Feature subset selection using a genetic algorithm. *IEEE Intelligent Systems*, 13(2):44–49, 1998.
- J. Yao, N. Teng, H.-l. Poh, and C.L. Tan. Forecasting and analysis of marketing data using neural networks. *Journal of Information Science and Engineering*, 14:843, 1998.
- X. Yao. Evolving artificial neural networks. *Invited Paper*, 1999.
- A.L. Yarin, M.B. Rubin, and I.V. Roisman. Penetration of a rigid projectile into an elastic-plastic target of finite thickness. *International Journal of Impact Engineering*, 16(5):6, 1995.
- R. Zaera. Simulation in high speed impact problems. *Metall. New. Mat. Res*, 9(4):25–52, 2001.
- R. Zaera and V. Sanchez-Galvez. Using an analytical model of simulation in the design of light-weight armours. *Simulation*, 70(3):175–181, 1998.
- X. Zeng, Y. Huang, and D.S. Yeung. Determining the relevance of input features for multilayer perceptrons. *Systems, Man and Cybernetics, 2003. IEEE International Conference on*, 1: 874–879, 2003.

- L. Zhang and G. Subbarayan. An evaluation of back-propagation neural networks for the optimal design of structural systems: Part i. training procedures. *Computer Methods in Applied Mechanics and Engineering*, 191:2873–2886(14), 2002.
- Y.I. Zhao and M. Small. Minimum description length criterion for modeling of chaotic attractors with multilayer perceptron networks. *Circuits and Systems I: Regular Papers, IEEE Transactions on*, 53(3):722–732, 2006.
- Z. Zhao, Y. Zhang, and H. Liao. Design of ensemble neural network using the akaike information criterion. *Engineering Applications of Artificial Intelligence*, 21(8):1182– 1188, 2008.
- D. Zhou and W. Stronge. Ballistic limit for oblique impact of thin sandwich panels and spaced plates. *International Journal of Impact Engineering*, 2007.
- O. Zienkiewics, R. Taylor, P.Ñithiarasu, and J. Zhu. *The Finite Element Method: Its Basis and Fundamentals*. McGraw-Hill, 2005.
- J.A. Zukas. *High velocity impact dynamics*. Wiley, 1990.
- J.A. Zukas. *Introduction to Hydrocodes*. Elsevier, 2004.
- J.A. Zukas, T.Ñicholas, H.F. Swift, L.B. Greszczuk, and D.R. Curran. *Impact Dynamics*. John Wiley & Sons, Inc., 1982.
- J. Zupan. Neural networks in chemistry. *The first European conference on computational chemistry*, 330(1):469–470, 1995.
- J. Zurada, A. Malinowski, and I. Cloete. Sensitivity analysis for minimization of input data dimension for feedforward neural network. *IEEE International Symposium on Circuits and Systems, ISCAS '94.*, 6:447–450 vol.6, 1994.
- J. Zurada, A. Malinowski, and S. Usui. Perturbation method for deleting redundant inputs of perceptron networks. *Neurocomputing*, 14:177–193(17), 1997.