

**UNIVERSIDAD CARLOS III DE MADRID**

*Escuela Politécnica Superior*



**Proyecto Fin de Carrera**

*Ingeniería Técnica Informática de Gestión*

Analisis de Técnicas para Medir la  
Influencia de las Variables del  
Vector de Entrada de un  
Perceptron Multicapa

Tutor:

González Carrasco, Israel

Alumno:

Romero Martínez, Roberto

Mayo de 2012



## RESUMEN

El objetivo primordial de cualquier proyecto relacionado con la simulación es obtener resultados fiables a partir de experimentos planteados. La principal cualidad que debe cumplir cualquier simulación reside en lograr resultados fidedignos con un menor coste de recursos (tiempo, personal, monetario . . .) que los que acarrearía la realización del experimento propiamente dicha.

Centrándonos en el ámbito que nos ocupa, el problema que se presenta en el entorno de los impactos en materiales es su no linealidad, y el elevado coste que requiere su estudio en modo experimental. Por ello, se trata de resolver estos problemas mediante una herramienta capaz de obtener resultados válidos mediante simulaciones, lo que reduce de forma drástica los costes asociados al análisis de los impactos.

Para este proyecto se ha creado una herramienta, en adelante Simulador, que automatiza el entrenamiento y testeo de una RNA, así como la generación de informes y resúmenes que permiten extrapolar datos del comportamiento de una red en base a su entrenamiento y datos de entrada. Con ayuda de esta información se ha buscado la mejor configuración de la RNA para el problema que nos atañe.

Uno de los principales objetivos de este proyecto es ayudar al lector a comprender el funcionamiento de la red mediante el análisis de los resultados que esta proporciona. Es decir, como influyen en los resultados la cantidad y variedad de los datos de entrada, así como la configuración de los diferentes métodos de entrenamiento.

Otro de los objetivos de todo proyecto de simulación es conocer la influencia que tienen cada una de las variables de entrada en los resultados obtenidos por la RNA. En este proyecto se utilizan dos métodos de estudio para la valoración de la criticidad de las variables de entrada “Fortaleza frente a valores atípicos” y “Sensitivity about the mean”. Se va a analizar y confrontar los resultados de ambos métodos para ver si sus resultados coinciden, y de no ser así, valorar cuál de los métodos parece más fiable.



## ABSTRACT

The primary objective of any project related to the simulation is to obtain reliable results from experiments presented. The best quality to be met by any simulation lies in achieving reliable results with a lower cost of resources (time, personnel, money ...) that would lead to the realization that the experiment itself.

Focusing on the subject who concerns us, the problem presented around of impacts on materials is their non-linearity, and the high cost necessary for his experimental study. Therefore, the question is resolve this problems with a tool to be able to have a good result through simulations, which dramatically reduces costs associated with the analysis of impacts

For this project has been made a tool, hereinafter referred to as "Simulator", which automates the training and the testing of an ANN, as well as the generation of simulation reports and summaries that allow extrapolating the behavior of a network based on its training and input data. Using this information we have tried to achieve for the best ANN configuration to the problem that concerns us.

One of the project main objectives is to help the reader to understand network's behavior by analyzing the results that it provides. That is, how influence in the results the quantity and the variety of the input data, as well as the configuration of the distinct training methods.

Another goal of any simulation project is to determine the influence of each of the input variables on the results obtained by ANN. This project will study two methods for assessing the criticality of the input variables "Fortress before outliers" and "Sensitivity about the mean". It will analyze and compare the results of both methods to see if their results match, and if not, assess which method seems more reliable.



## **ÍNDICE DE CONTENIDOS**

<b>Capítulo 1. INTRODUCCIÓN .....</b>	<b>9</b>
1.1. Contexto.....	9
1.2. Objetivo .....	11
1.3. Principales herramientas utilizadas.....	13
<b>Capítulo 2. ESTADO DEL ARTE.....</b>	<b>15</b>
2.1. Introducción .....	15
2.2. Historia de las redes neuronales.....	16
2.3. Perspectiva futura .....	21
2.4. Red neuronal .....	24
2.4.1. Definición de una red neuronal.....	24
2.4.2. Ventajas de las redes neuronales .....	25
2.4.3. Tipos y aplicación de las redes más importantes .....	28
2.4.4. Elementos y estructura de una red neuronal artificial.....	30
2.4.5. Estructura de una red neuronal. ....	36
2.4.6. Proceso de aprendizaje.....	38
2.4.7. Redes neuronales biológicas.....	40
2.4.8. Analogía con las redes neuronales biológicas. ....	42
2.5. El Perceptrón.....	44
2.5.1. El perceptron multicapa .....	44
2.5.2. Algoritmo de retropropagación (backpropagation) .....	49
<b>Capítulo 3. ESTUDIO DE VIABILIDAD DEL SISTEMA .....</b>	<b>55</b>
3.1. Introducción .....	55
3.2. Establecimiento del alcance del sistema.....	55
3.2.1. Estudio de la solicitud.....	56
3.2.2. Identificación del alcance del sistema .....	57
3.2.3. Identificación de los interesados en el sistema (stakeholders).....	58
3.3. Estudio de la situación actual.....	58
3.3.1. Valoración del estudio de la situación actual.....	58
3.3.2. Realización del diagnóstico de la situación actual.....	59
3.4. Definición de los Requisitos del Sistema .....	60
3.4.1. Especificación de los casos de uso.....	61
3.4.1.1. Especificación de los casos de uso.....	62
3.4.2. Definición de los requisitos del sistema .....	72
3.4.2.1. Identificación de requisitos.....	72
3.4.2.2. Requisitos de capacidad.....	74
3.4.2.3. Requisitos de Restricción.....	82
3.4.3. Estudio de alternativas de solución.....	86
3.4.4. Valoración de las alternativas .....	86
3.4.5. Selección de la solución.....	86



<b>Capítulo 4. GESTIÓN DEL PROYECTO .....</b>	<b>87</b>
4.1. Ciclo de vida .....	87
4.2. Organización del proyecto .....	89
4.3. Planificación .....	95
4.4. Estimación de costes .....	102
<b>Capítulo 5. MANUAL DE USUARIO “SIMULADOR” .....</b>	<b>104</b>
5.1. Introducción .....	104
5.2. Generación de ficheros .....	105
5.2.1. Introducción .....	105
5.2.2. Parámetros de entrada .....	106
5.2.3. Proceso de generación .....	109
5.3. Generación de informes .....	112
5.3.1. Introducción .....	112
5.3.2. Parámetros de entrada .....	113
5.3.3. Proceso de entrenamiento e informes .....	114
5.4. Generación del Resumen .....	117
5.4.1. Introducción .....	117
5.4.2. Parámetros de entrada .....	117
5.4.3. Proceso de resumen .....	118
<b>Capítulo 6. EVALUACIÓN Y EXPERIMENTACIÓN.....</b>	<b>120</b>
6.1. Datos iniciales.....	120
6.2. Simulaciones de clasificación.....	122
6.2.1. Introducción.....	122
6.2.2. Simulaciones para pruebas.....	124
6.2.3. Simulaciones para Epocs, MSE y CV .....	126
6.2.3.1. Ajustando el número de Epocs I.....	128
6.2.3.2. Ajustando el número de Epocs II.....	129
6.2.3.3. Ajustando el valor del MSE.....	130
6.2.3.4. Incluyendo datos para CV I .....	132
6.2.3.5. Incluyendo datos para CV II.....	134
6.2.4. número de reg. en Entrenamiento, Test y CV.....	137
6.2.4.1. Modificando el nº de reg. para Entrenamiento .....	137
6.2.4.2. Modificando el nº de reg. para Test.....	138
6.2.4.3. Modificando el nº de reg. para Test y CV .....	139
6.3. Simulaciones de predicción .....	140
6.3.1. Introducción.....	140
6.3.2. Simulaciones para pruebas.....	145
6.3.2.1. Mfinal .....	145
6.3.2.2. Mi .....	146
6.3.2.3. Vres .....	146
6.3.2.4. DOP .....	147
6.3.3. Simulaciones para Epocs, MSE y CV .....	149
6.3.3.1. Mfinal .....	149
6.3.3.2. Mi .....	158
6.3.3.3. Vres .....	164



6.3.3.4. DOP .....	173
6.4. Análisis de sensibilidad .....	182
6.4.1. Introducción .....	182
6.4.2. Sensitivity about the mean .....	183
6.4.2.1. Informe Sensitivity About the Mean (SAM) .....	184
6.4.3. Fortaleza Frente a Valores Atípicos (FFVA).....	190
6.4.3.1. Material de proyectil, material de la placa y radio del proyectil. ....	194
6.4.3.2. Grosor de la placa .....	195
6.4.3.3. Velocidad del proyectil.....	197
6.4.3.4. Masa inicial del proyectil.....	198
6.4.3.5. Longitud del proyectil.....	199
6.4.3.6. Resumen General.....	200
<b>Capítulo 7. CONCLUSIONES .....</b>	<b>201</b>
<b>REFERENCIAS.....</b>	<b>203</b>



## **ÍNDICE DE FIGURAS**

Ilustración 1: Elementos de una RNA .....	30
Ilustración 2: Salida de una Neurona .....	34
Ilustración 3: Neurona Biológica .....	40
Ilustración 4: Unidad de Proceso Típica .....	42
Ilustración 5: Perceptrón Multicapa .....	45
Ilustración 6: Función Lineal .....	47
Ilustración 7: Función Sigmoidal logística .....	47
Ilustración 8: Función Sigmoidal Tangente Hiperbólica .....	48
Ilustración 9: Superficie del Error 1 .....	51
Ilustración 10: Superficie del Error 2 .....	52
Ilustración 11: Casos de uso .....	61
Ilustración 12: Ciclo de vida .....	87
Ilustración 13: RBS del proyecto .....	90
Ilustración 14: WBS Proceso Estudio de viabilidad .....	91
Ilustración 15: WBS Gestión del proyecto .....	91
Ilustración 16: WBS Primer prototipo .....	92
Ilustración 17: WBS Segundo prototipo .....	93
Ilustración 18: WBS de la documentación .....	93
Ilustración 19: WBS Evaluación y experimentación .....	94
Ilustración 20: Diagrama de Gantt principal .....	95
Ilustración 21: Diagrama de Gantt del estudio de la viabilidad .....	95
Ilustración 22: Diagrama de Gantt de la gestión del proyecto .....	96
Ilustración 23: Diagrama de Gantt del primer prototipo .....	98
Ilustración 24: Diagrama de Gantt del segundo prototipo .....	99
Ilustración 25: Diagrama de Gantt de la evaluación y experimentación .....	100
Ilustración 26: Diagrama de Gantt del manual de usuario .....	101
Ilustración 27: Bloques de Simulador .....	104
Ilustración 28: Generación de ficheros .....	105
Ilustración 29: Selección de datos de entrada .....	106
Ilustración 30: Generación de ficheros iterativo .....	108
Ilustración 31: Generación de informes .....	112
Ilustración 32: Generación de resumen .....	117
Ilustración 33: Tabla de ensayos .....	121
Ilustración 34: Resumen resultados SAM .....	184
Ilustración 35: Resultados SAM grosor .....	185
Ilustración 36: Resultados SAM radio .....	186
Ilustración 37: Resultados SAM longitud .....	187
Ilustración 38: Resultados SAM velocidad .....	187
Ilustración 39: Resultados SAM masa inicial .....	188
Ilustración 40: Resultados SAM material placa .....	188
Ilustración 41: Resultados SAM material proyectil .....	189



## **ÍNDICE DE TABLAS**

Tabla 1: Redes y Aplicaciones.....	29
Tabla 2: Comparación entre las Neuronas Biológicas y Artificiales.....	43
Tabla 3: Caso de Uso 001 .....	62
Tabla 4: Caso de Uso 002 .....	63
Tabla 5: Caso de Uso 003 .....	64
Tabla 6: Caso de Uso 004.....	64
Tabla 7: Caso de Uso 005 .....	65
Tabla 8: Caso de Uso 006.....	65
Tabla 9: Caso de Uso 007 .....	66
Tabla 10: Caso de Uso 008.....	66
Tabla 11: Caso de Uso 009.....	67
Tabla 12: Caso de Uso 010.....	68
Tabla 13: Caso de Uso 011.....	68
Tabla 14: Caso de Uso 012.....	69
Tabla 15: Caso de Uso 013.....	69
Tabla 16: Caso de Uso 014.....	70
Tabla 17: Caso de Uso 015.....	71
Tabla 18: Caso de Uso 016.....	71
Tabla 19: Caso de Uso 017.....	72
Tabla 20: Requisito de usuario C001.....	74
Tabla 21: Requisito de usuario C002.....	75
Tabla 22: Requisito de usuario C003.....	75
Tabla 23: Requisito de usuario C004.....	76
Tabla 24: Requisito de usuario C005.....	76
Tabla 25: Requisito de usuario C006.....	77
Tabla 26: Requisito de usuario C007.....	77
Tabla 27: Requisito de usuario C008.....	78
Tabla 28: Requisito de usuario C009.....	78
Tabla 29: Requisito de usuario C010.....	78
Tabla 30: Requisito de usuario C011.....	79
Tabla 31: Requisito de usuario C012.....	79
Tabla 32: Requisito de usuario C013.....	80
Tabla 33: Requisito de usuario C014.....	80
Tabla 34: Requisito de usuario C015.....	80
Tabla 35: Requisito de usuario C016.....	81
Tabla 36: Requisito de usuario C017.....	81
Tabla 37: Requisito de usuario C018.....	82
Tabla 38: Requisito de usuario R001.....	82
Tabla 39: Requisito de usuario R002.....	83
Tabla 40: Requisito de usuario R003.....	83
Tabla 41: Requisito de usuario R004.....	84
Tabla 42: Requisito de usuario R005.....	84
Tabla 43: Requisito de usuario R006.....	84
Tabla 44: Requisito de usuario R007.....	85
Tabla 45: Requisito de usuario R008.....	85
Tabla 46: Requisito de usuario R009.....	86





Tabla 47: Coste del software.....	102
Tabla 48: Coste de infraestructura .....	102
Tabla 49: Coste humano .....	102
Tabla 50: Coste total .....	103
Tabla 51: Contenido de ficheros .....	111
Tabla 52: Detalle de un informe clasificación .....	115
Tabla 53: Detalle de un informe predicción .....	116
Tabla 54: Resumen de clasificación 1 .....	119
Tabla 55: Resumen de clasificación 2 .....	119
Tabla 56: Resumen de predicción.....	119
Tabla 57: Resultados de clasificación.....	124
Tabla 58: Resultados de clasificación.....	127
Tabla 59: Resultados de clasificación.....	128
Tabla 60: Resultados de clasificación.....	129
Tabla 61: Resultados de clasificación.....	131
Tabla 62: Resultados de clasificación.....	133
Tabla 63: Resultados de clasificación.....	134
Tabla 64: Resultados de clasificación.....	137
Tabla 65: Resultados de clasificación.....	138
Tabla 66: Resultados de clasificación.....	139
Tabla 67: Resultados de predicción .....	145
Tabla 68: Resultados de predicción .....	146
Tabla 69: Resultados de predicción .....	147
Tabla 70: Resultados de predicción .....	147
Tabla 71: Resultados de predicción .....	150
Tabla 72: Resultados de predicción .....	151
Tabla 73: Resultados de predicción .....	152
Tabla 74: Resultados de predicción .....	153
Tabla 75: Resultados de predicción .....	155
Tabla 76: Resultados de predicción .....	159
Tabla 77: Resultados de predicción .....	160
Tabla 78: Resultados de predicción .....	161
Tabla 79: Resultados de predicción .....	162
Tabla 80: Resultados de predicción .....	165
Tabla 81: Resultados de predicción .....	167
Tabla 82: Resultados de predicción .....	167
Tabla 83: Resultados de predicción .....	169
Tabla 84: Resultados de predicción .....	170
Tabla 85: Resultados de predicción .....	174
Tabla 86: Resultados de predicción .....	175
Tabla 87: Resultados de predicción .....	176
Tabla 88: Resultados de predicción .....	177
Tabla 89: Resultados de predicción .....	179
Tabla 90: Resumen resultados SAM .....	185
Tabla 91: Resumen fortaleza frente a valores atípicos (FFVA) .....	193
Tabla 92: Informe FFVA .....	194
Tabla 93: Informe FFVA resumen epocs .....	196
Tabla 94: Valoración FFVA .....	200



## Capítulo 1. INTRODUCCIÓN

### 1.1. Contexto

Durante los cincuenta últimos años, los científicos y los 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 casi exclusivamente entorno a colisiones de sólidos de geometría simple e impactos de baja velocidad, con la deformación plástica limitada, dada de la complejidad matemática de las ecuaciones implicadas en estos fenómenos y a la limitada información disponible sobre el comportamiento de los materiales en su deformación a alta velocidad [1]. La creación de potentes computadores, con capacidades de cálculo antaño impensables, y las nuevas técnicas instrumentales han proporcionado nuevos datos de prueba que han ayudado a solucionar muchos de los problemas sobre el impacto de proyectiles.

Existen numerosos elementos que pueden sufrir el impacto de proyectiles a gran velocidad durante el desempeño de su actividad, estos elementos pueden ser partes esenciales de un sistema, por lo que es vital conocer como se comportarían tales elementos tras el impacto de un proyectil. Los ejemplos más característicos se encuentran en el campo militar aunque 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.

El diseño de estructuras que no fuesen vulnerables a determinados impactos de proyectiles, han sido realizadas tradicionalmente de forma empírica, confiando en las pruebas de impacto reales realizadas con un proyectil y blanco determinado. La completad matemática de solucionar las ecuaciones que gobiernan el fenómeno del impacto, y a la ignorancia relativa del comportamiento mecánico de los materiales en altos rango de tensión, desalentó el intento de cualquier simulación del problema. El método empírico es fiable pero los resultados solamente 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. Es aún más costoso cuanto más se afinen los descubrimientos que se desean realizar con las pruebas.

Con el objetivo de disminuir costes en la realización de pruebas, se desarrollaron algunas técnicas semi-empíricas. Estas ecuaciones algebraicas con base teórica, calibradas con una gran cantidad de datos experimentales permitieron cierta interpolación o incluso extrapolación de resultados.

La necesidad de herramientas de diseño capaces de simular y predecir las consecuencias de los impactos accionó el desarrollo de gran cantidad de modelos de simulación. Así, el uso de costosas pruebas experimentales se ha relegado a la etapa final del diseño. Todas las etapas anteriores se pueden cubrir por el uso de las herramientas de la simulación.

Para poder predecir las posibles consecuencias del impacto de un determinado proyectil sobre una superficie vamos a realizar una herramienta que nos permita simular las consecuencias del impacto en base a una colección de datos de entrada. Para ello, vamos a ayudarnos de una Red Artificial de Neuronas. La utilización de las RNA puede orientarse para la resolución de problemas prácticos como la clasificación de patrones y la predicción de funciones. Las RNA han sido aplicadas de forma satisfactoria en la predicción de diversos problemas en diferentes áreas de conocimiento --biología, medicina, economía, ingeniería, psicología, etc. [2] [3] [4] -- obteniendo excelentes resultados respecto a los modelos derivados de la estadística clásica [5] [6] [7] [8] [9] . El paralelismo de cálculo, la memoria distribuida y la adaptabilidad al entorno, han convertido a las RNA en potentes instrumentos con capacidad para aprender relaciones entre variables sin necesidad de imponer presupuestos o restricciones de partida en los datos.

Actualmente, existen unos 40 modelos de RNA que son usados en diversos campos de trabajo [10] . Entre estos paradigmas, el más ampliamente utilizado es el



Perceptrón multicapa asociado al algoritmo de aprendizaje backpropagation error (propagación del error hacia atrás), también denominado red backpropagation [11] La popularidad del Perceptrón multicapa se debe principalmente a que es una herramienta eficiente para la clasificación y la predicción en problemas no lineales, es decir, a que es capaz de actuar como un aproximador universal de funciones [12] [13] . Más concretamente, una red backpropagation conteniendo al menos una capa oculta con suficientes unidades no lineales puede aprender cualquier tipo de función o relación continua entre un grupo de variables de entrada y salida. Esta propiedad convierte a las redes Perceptrón multicapa en herramientas de propósito general, flexibles y no lineales.

La habilidad del Perceptrón multicapa para aprender a partir de un conjunto de ejemplos, aproximar relaciones no lineales, filtrar ruido en datos, etc. hace que sea un modelo adecuado para abordar problemas reales. Dentro del marco de las redes de neuronas, el Perceptrón multicapa es en la actualidad una de las arquitecturas más utilizadas en la resolución de problemas de estructuras y de materiales, y más concretamente en el tipo de problema que vamos solucionar, es decir, la predicción de las posibles consecuencias del impacto de un proyectil sobre una superficie determinada [14] [15] [16] [17] .

Para la resolución de nuestro problema, todo lo expuesto hasta ahora, nos lleva a la determinación de modelar una Red de Neuronas del tipo Perceptrón Multicapa con algoritmo de aprendizaje backpropagation con una función de transferencia de tipo identidad. Para su desarrollo utilizaremos un software que trabaja en un ambiente Excel, la herramienta Neurosolutions.

## 1.2. Objetivo

Este proyecto surge como respuesta a una serie de nuevas necesidades que surgen de la utilización de los proyectos anteriormente creados en el Departamento de Informática para la predicción de impactos. Los proyectos anteriormente realizados son: “Simulación de impactos en materiales mediante redes neuronales”, “Parametrización del aprendizaje de una red neuronal para la predicción de impactos sobre materiales” y



“Aplicación de un perceptron multicapa para la resolución de problemas de impacto balístico”.

El objetivo principal de cualquier proyecto relacionado con la simulación es obtener resultados fiables a partir de experimentos planteados, es decir, obtener los resultados sin que los estos tengan que ser llevados a cabo por su método tradicional. La principal cualidad que debe cumplir cualquier simulación reside en lograr resultados fidedignos con un menor coste de recursos (tiempo, personal, monetario . . .) que los que acarrearía la realización del experimento propiamente dicha.

Centrándonos en el ámbito que nos ocupa, el problema que se presenta en el entorno de los impactos en materiales es su no linealidad, y el elevado coste que requiere su estudio en modo experimental. La experimentación en esta materia consume un gran número de recursos tanto espaciales, temporales y materiales, debido al gran número de parámetros influyentes en los resultados que necesitan ser controlados. Por ello, se trata de resolver estos problemas mediante una herramienta capaz de obtener resultados válidos mediante simulaciones, lo que reduce de forma drástica los costes asociados al análisis de los impactos.

El proyecto no busca ser tratado como una evolución de uno de los proyectos anteriores, si no como un compendio de las mejores posibilidades que aquellos nos ofrecen, incluyendo en él, nuevas necesidades, y nuevos métodos para el análisis de las variables.

A partir de ahora nos referiremos al software creado para este proyecto con el nombre de Simulador.

Uno de los principales objetivos del proyecto es, comprobar el grado de fiabilidad que muestra una red de neuronas, a la hora de predecir el patrón de comportamiento, que se genera en el lanzamiento de proyectiles ante una superficie. Se intentará obtener la mejor configuración de la red para el problema que nos atañe. No existe ningún método que nos permita afirmar que una red, y/o configuración de una



red, es la más óptima para solucionar un determinado problema, por lo que, para fijar la mejor configuración, nos basaremos, únicamente, en los resultados proporcionados por la herramienta desarrollada.

Otro de los principales objetivos de todo proyecto de simulación, es el de lograr conocer cómo afectan las variables de entrada en los resultados obtenidos por la RNA. Los resultados los obtendremos mediante Simulador por el método conocido como “Fortaleza frente a valores atípicos” o “Criticalidad de las variables”. Para contrastar estos resultados, realizaremos un estudio paralelo de las variables de entrada, con Neurosolutions; este estudio recibe el nombre de “Sensitivity about the mean”

Una de las principales cualidades que se ha buscado implementar en Simulador ha sido la amigabilidad del programa. Simulador pretende ser una herramienta de fácil manejo a ojos, no de un informático, si no de cualquier usuario no relacionado con la informática que pretenda obtener la mejor configuración de una red para cualquier tipo estudio. Se ha intentado dotar a Simulador de la mayor flexibilidad posible (extensiones de ficheros de entrada, formateo, rutas, variables que intervienen . . .); además, se ha modularizado de forma que el proceso total, es decir, generación de ficheros, entrenamiento de la red, y obtención de los resultados, pueda realizarse, bien como un proceso lineal unidireccional, o bien, como pequeños procesos independientes, es decir, que cada parte del proceso total pueda ser ejecutada como un modulo estanco, totalmente independiente del resto de los módulos de Simulador, procurando así dotar de mayor funcionalidad a la herramienta.

### 1.3. Principales herramientas utilizadas

A continuación se detallan las principales herramientas utilizadas para el desarrollo de Simulador:

- NeuroSolutions 5.0
- Microsoft Visual Studio 6.0
- Microsoft Windows XP Professional Version 2002 Service Pack 3



**Proyecto:** Analisis de Técnicas para Medir la Influencia de las Variables del Vector de Entrada de un Perceptron Multicapa

**Autor:** Roberto Romero Martínez

**Capítulo:** Introducción

- Microsoft Office Professional Edition 2003 y Microsoft Office Ultimate 2007
- Otras herramientas de apoyo necesarias para la ejecución del proyecto:
- Mozilla Firefox
- Biblioscape
- UltraEdit
- JEdit



## Capítulo 2. ESTADO DEL ARTE

### 2.1. Introducción

Durante muchos siglos una de las principales ocupaciones de la humanidad ha sido el desarrollo de máquinas. Éstas se diseñaron para intentar procurarnos una vida más cómoda y despreocupada. La era de las máquinas comenzó con la invención de la rueda, la polea... Muchas invenciones después, los ingenieros y los científicos de hoy en día están intentando desarrollar máquinas inteligentes. Los sistemas artificiales de neuronas son ejemplos actuales del desarrollo de tales intentos.

Una de las principales características diferenciadoras del cerebro humano es la capacidad de procesar a gran velocidad grandes cantidades de información procedentes de los sentidos, combinarla o compararla con la información almacenada y dar respuestas adecuadas incluso ante nuevas situaciones. Por ejemplo, el cerebro humano es mucho mejor y más rápido reconociendo imágenes que las más avanzada de las computadoras.

Aunque todavía se ignora mucho sobre la forma en que el cerebro realiza estas funciones, se están desarrollando modelos matemáticos que tratan de mimetizar tales habilidades, una de ellas es el modelo que nos ocupa, los sistemas de Redes de Neuronas Artificiales. Con evolución de estos sistemas se ha logrado realizar investigaciones fiables que nos ayuden a la predicción del comportamiento de un proyectil y su blanco durante el impacto. La utilización de estos sistemas, nos ha permitido aventurar los resultados de los impactos sin tener que realizarlos de forma empírica, obteniendo un elevado grado de fiabilidad. Esto nos ha permitido reducir los costes y los tiempos en el estudio de impactos de proyectiles de una forma más que considerable.





## 2.2. Historia de las redes neuronales

Puesto que no es posible cubrir la historia en un breve capítulo introductorio, sólo vamos a mencionar los logros más importantes. Esta ojeada a la historia debe intentar hacernos ver como ha sido el desarrollo de las redes de neuronas artificiales durante los años. El resumen histórico que se muestra a continuación no es exhaustivo; se omiten algunos hechos y otros sólo se mencionan brevemente.

El año 1943 a menudo es considerado como el año de comienzo del desarrollo de los sistemas de neuronas artificiales. McCulloch y Pitts [18], neurofisiólogo y matemático respectivamente, realizaron el primer modelo matemático de unas Redes de Neuronas Artificiales. El modelo de McCulloch y Pitts propuso la teoría general de procesamiento de la información basada en redes de trabajo de elementos de conmutación o decisión, los cuales eufemísticamente fueron llamados “neuronas”, aunque fuesen de lejos, mucho más simples que sus homónimos biológicos. Aunque este modelo generó gran interés al proporcionar medidas de comportamiento sofisticadas a través de cálculos sencillos, el factor clave del modelo está en la capacidad de aprendizaje.

Donald Hebb [19] fue el primero en proponer un esquema de aprendizaje basado en las conexiones de neuronas, al cual ahora nos referimos como aprendizaje hebbiano. Hebb estimó que la información se puede almacenar utilizando las conexiones entre neuronas y unos pesos asociados a ellas, y postuló técnicas de aprendizaje que tuvieron un impacto profundo en el futuro desarrollo de este campo [20].

Durante la década de los 50 la primera neurocomputadora fue construida y probada (Minsky 1954). Más tarde, Frank Rosenblatt [21] generalizó el modelo de McCulloch y Pitts; llamó a este *Perceptron*. El Perceptron red neuronal artificial de neuronas más antigua, y se usa hoy para el reconocimiento de patrones. Primero se desarrolló un modelo de dos niveles, que ajustaba los pesos de las conexiones entre los niveles de entrada y salida, en proporción al error entre la salida deseada y la obtenida.



Este modelo era capaz de generalizar patrones; es decir, después de entrenarlo con una serie de patrones modelo era capaz de reconocer otros similares, aunque no se los hubieran presentado anteriormente. Rosenblatt intentó extender su procedimiento de aprendizaje a un Perceptron de tres niveles, pero no encontró un método matemático sólido para entrenar la capa de conexiones oculta. El modelo tenía varias limitaciones, la más conocida era su incapacidad para resolver el problema de la función OR-exclusiva y, en general, no era capaz de clasificar clases no separables linealmente

En 1959, Bernard Widrow y Marcial Hoff diseñaron una Red Neuronal Artificial muy similar al Perceptron, llamada Adaptative Linear Element o *Adaline*. Introduce unas nuevas reglas de aprendizaje, llamadas reglas de Widrow-Hoff. La diferencia entre los dos modelos son muy pequeñas, pero las aplicaciones a las que van dirigidas son muy distintas. Dentro de las distintas aplicaciones del Perceptron se pueden destacar el reconocimiento de patrones y la predicción, mientras que Adaline ha tenido como principal aplicación el filtrado de señales. Fue la primera red neuronal aplicada a un problema real, se utilizó como filtro adaptativo para eliminar ecos en las líneas telefónicas, y se ha usado comercialmente durante varias décadas [22] .

A pesar de los éxitos y el entusiasmo existente en los inicios y mediados de los años sesenta, los métodos de aprendizaje eran demasiado débiles para utilizarse como apoyo en la solución de problemas complejos. Aunque los cuellos de botella fueron identificados por los trabajos de Nilsson y las redes de neuronas conocidas como layered networks eran bien conocidas, seguían sin existir esquemas de aprendizaje más potentes. La investigación en redes de neuronas entró en una fase de estancamiento. Otra razón que contribuyó al estancamiento en la investigación durante aquella época fue la modesta disponibilidad de recursos computacionales [20] .

Sobre el 1969 surgieron grandes y numerosas críticas que frenaron parcialmente, hasta aproximadamente 1982, el crecimiento que estaban experimentando las investigaciones sobre redes neuronales. Marvin Minsky y Seymour Papera, de Instituto Tecnológico de Massachussets (MIT) publicaron un libro, *Perceptrons* [23] , que, además de contener un análisis matemático detallado del Perceptron, consideraba que



las extensión a Perceptron multinivel era completamente estéril. También se dieron cuenta de su incapacidad para representar la función XOR debido a su naturaleza lineal. Estas limitaciones eran muy importantes en aquella época, sobre todo su incapacidad de resolver muchos problemas interesantes. Gran cantidad de las investigaciones dieron un giro hacia la inteligencia artificial, que tenía mejores augurios por aquellos entonces [24]

A pesar de las conclusiones obtenidas en el libro *Perceptrons*, algunos investigadores continuaron su trabajo estudiando, desarrollando y creando nuevos modelos de redes de neuronas. Los nuevos descubrimientos que se llevaron a cabo durante estos años propiciaron el verdadero reconocimiento por parte de todos los científicos del potencial real de este tipo de aplicaciones [22].

- Stephen Grossberg: medico y matemáticos considerado como el más influyente y formal de todos los investigadores en Redes de Neuronas Artificiales. Grossberg realizó importantes trabajos tratando de basarse en principios fisiológicos, que incluyen estrictos análisis matemáticos, que permitieron la realización de nuevos paradigmas de Redes de Neuronas. Éstos permitían tener acceso directo a la información mientras se operaba a tiempo real.
- Shun-Ichi Amari combinó la actividad de las redes neuronales biológicas con rigurosos modelos matemáticos de Redes de Neuronas Artificiales. Sus estudios incluyen el tratamiento de Redes de Neuronas Artificiales dinámicas y aleatoriamente conectadas [25] [26] [27], estudios de aprendizaje competitivo [28] [29] [30], así como el análisis matemático de memorias asociativas [31].
- James Anderson, que trabajó con un modelo de memoria basado en la asociación de activaciones de las sinapsis de una neurona, realizó un modelo de memoria asociativa lineal (Asociador Lineal), siguiendo el planteamiento de Hebb. Empleó un nuevo método de corrección de error, y sustituyó la



función umbral lineal por otra en rampa, creando un nuevo modelo llamado *Brain-State-in-a-Box (BSB)* [32] .

- Kuniyiko Fukushima creó un nuevo tipo de arquitectura de red de neuronas conocido como *Neocognitron* [33] . Este era un modelo de red neuronal para el reconocimiento de patrones visuales.
- Teuvo Kohonen, sus trabajos se centraron en memorias asociativas y matrices de correlación, de manera semejante a los trabajos de Anderson. Extendió el modelo de memoria lineal asociativa, que requería vectores linealmente independientes para un buen rendimiento, en uno que buscaba los óptimos entre vectores linealmente dependientes, llamado *Asociador Óptimo de Memoria Lineal (OLAM)*. Más tarde, realizó investigaciones en métodos de aprendizaje y desarrolló el “*Learning Vector Quantization*” (*LVQ*), un sistema de aprendizaje competitivo.

Durante el periodo de 1982 a 1986, varias publicaciones potenciaron de forma significativa el potencial de las redes de neuronas. La era del renacimiento comenzó con John Hopfield [34] [35] introduciendo una arquitectura de red de neuronas para memorias asociativas. Sus trabajos formularon las características de cómputo de una red de unidades completamente conectada, pero, además, mostró tanto entusiasmo y claridad de presentación que estos trabajos produjeron un nuevo impulso al campo y provocaron el incremento de las investigaciones.

A partir de los estudios de James McClelland y David Rumelhart, seguidos con mucho interés por otros investigadores, nació el grupo PDP (Parallel Distributed Processing). La culminación de los estudios de los investigadores del grupo PDP fue la publicación, en 1986, de un libro editado en dos volúmenes titulado *Parallel Distributed Processing : Explorations in the Microstructure of Cognition* [36] . Entre los muchos autores de este libro cabe destacar a los siguientes: McClelland, Rumelhart, Hinton, Zipser, Smolensky, Sejnowsky, Willians, Jordan, Stone, Rabin, Elman, Kawamoto, Crack, Asanuma y Munro. La parte principal del libro es el capítulo



dedicado al algoritmo de retropropagación, que soluciona los problemas planteados por Minsky y Papert., eliminando así, una de las más importantes barreras que habían echado por tierra los esfuerzos de la corriente de pensamiento de mediados de los años sesenta. Esta publicación abrió un nuevo periodo para, el una vez subestimado, potencial de computación de las redes en capa.

A partir de 1986-1987, comenzaron muchos programas nuevos de investigación de redes neuronales. El aumento en la intensidad de investigación en la disciplina de la neurocomputación, a partir de entonces, puede medirse con facilidad observando el rápido crecimiento del número de conferencias y periódicos dedicados a este campo. Además, de los muchos volúmenes editados que contienen colecciones de “*papers*”, algunos libros ya han se han publicado. Asimismo, los problemas que pueden ser solucionados con redes neuronales se ha expandido desde pequeños ejemplos de prueba hasta grandes programas con aplicación práctica real.



### 2.3. Perspectiva futura

La explosión actual del interés en sistemas de neuronas artificiales se basa en el elevado número de expectativas científicas y económicas creadas. Uno debe de conocer, sin embargo, que algunas de las expectativas populares puede ser exageradas, y no del todo realistas.

Las redes de neuronas pueden aplicarse para la solución de una amplia variedad de problemas, aunque la mayor parte de ellos suelen tener como fondo en reconocer tendencias en grandes cantidades de datos. Las RNA satisfacen mejor que las arquitecturas tradicionales de computadoras los problemas donde los seres humanos suelen ser buenos, y las computadoras tradicionales malas. Por ejemplo, en el reconocimiento de imágenes, haciendo generalizaciones, etc.

La simulación de los sentidos y las emociones humanas sigue siendo parte del reino de la ciencia ficción. ¿Se podrá lograr esta simulación algún día? Esto es un tema que no discutiremos aquí, porque, por supuesto, existe además de la discusión propiamente científica, otra filosófica, que se postula demasiado profunda, y en un extremo, quizás imposible de resolver.

El futuro de las redes de neuronas no reside por completo en tentativas de simular los sentidos. De hecho, por el momento esta preocupación es relativamente pequeña, ya que donde más hincapié se hace es en mejorar los sistemas con los que contamos actualmente.

Avances recientes y usos futuros de las RNA:

- Integración de la lógica confusa: es una lógica que reconoce más valores que el verdadero y falso, por lo tanto es más eficiente para describir el mundo real. Por ejemplo, si no hay nubes la posibilidad de que haya sol es del 100%, si hay alguna nube la posibilidad de que haya sol es del 80 %... . Por lo tanto esta lógica es capaz de considerar conceptos siempre, casi siempre ...



La lógica confusa y las redes de neuronas se han integrado para aplicaciones tan diversas como la ingeniería automotora, control de grúas y la supervisión del glaucoma.

- Redes de neuronas implementadas en hardware: La mayoría de las RNA están basadas en un modelo de computación que implica la propagación de variables continuas de una unidad de proceso a otra. En estos últimos años, los datos obtenidos en experimentos neurobiológicos han dejado cada vez más claro que las redes de neuronas biológicas, las cuales se comunican a través de pulsos, utilizan la sincronización de pulso para transmitir información y realizar cómputos. Este hecho, ha estimulado de forma significativa la investigación de redes de neuronas, además del análisis teórico y del desarrollo de modelos neurobiológicos, y su implementación en hardware.

Algunas redes están siendo codificadas en chips o dispositivos análogos, esta tecnología llegará a ser muy útil cuando aprendamos a hacerlo con redes complejas. La principal ventaja de este avance es la velocidad que tendrán estas redes al obtener el resultado de un problema. Las RNA implementadas en hardware se utilizan ya en algunas áreas como en el reconocimiento de la voz, y en aquellas que se requiere un alto rendimiento.

Muchas redes de neuronas tienen alrededor de 100 neuronas y necesitan entrenamiento ocasional. En estas situaciones, el software de simulación generalmente es eficiente. Pero cuando en un futuro los algoritmos de RNA estén diseñados para tareas más complejas, las redes podrán tener miles de neuronas y cientos de miles de sinapsis, por lo que será esencial tener desarrollado hardware de alto rendimiento de redes de neuronas, para su utilización práctica.

- Mejora de las tecnologías existentes: Todas las tecnologías actuales de RNA, muy probablemente, serán sumamente mejoradas en el futuro. Se desarrollarán



nuevos métodos de entrenamiento y nuevas arquitecturas que mejoraran los sistemas de reconocimiento de voz, escritura, imágenes ...

Las RNA quizá en el futuro permitan:

- Crear robots que puedan ver, sentir, y predecir cosas sobre el mundo que les rodea.
- Crear coches que se autoconduzcan.
- Componer música.
- Transformar automáticamente documentos manuscritos en documentos de texto de ordenador.
- Ayudar a comprender los datos recopilados en el proyecto genoma humano.
- Realizar exámenes médicos-
- ...





## 2.4. Red neuronal

### 2.4.1. DEFINICIÓN DE UNA RED NEURONAL

Existen diferentes formas de responder a la pregunta de cómo definir una Red de Neuronas Artificial. Por un lado, se podría definir, simplemente una red neuronal, como una clase de algoritmos matemáticos. Mientras que en el otro extremo, se podrían definir, como redes sintéticas que imitan redes neuronales biológicas encontradas en organismos vivos. En luz al todavía limitado conocimiento de hoy en día sobre el sistema nervioso y las redes neuronales biológicas, puede considerarse como más plausible que la respuesta este algo más cercana al sentido algorítmico de su posible definición.

Estas definiciones van desde cortas y genéricas, hasta las que intentan explicar más detalladamente lo que significa red neuronal o computación neuronal. Se muestran a continuación algunos casos representativos:

*“Una nueva forma de computación, inspirada en modelos biológicos.” [24] .*

*“Un modelo matemático compuesto por un gran número de elementos procesales organizados en niveles.” [24] .*

*“... un sistema de computación hecho por un gran número de elementos simples, elementos de proceso muy interconectados, los cuales procesan información por medio de su estado dinámico como respuesta a entradas externas”[37]*

*“Redes neuronales artificiales son redes interconectadas masivamente en paralelo de elementos simples (usualmente adaptativos) y con organización jerárquica, las cuales intentan interactuar con los objetos del mundo real del mismo modo que lo hace el sistema nervioso biológico.” [38]*



#### 2.4.2. VENTAJAS DE LAS REDES NEURONALES

Debido a su constitución y a sus fundamentos, las redes neuronales artificiales presentan gran número de características similares a las del cerebro. Por ejemplo, son capaces de aprender de la experiencia, de generalizar de casos anteriores nuevos casos, de abstraer características esenciales a partir de entradas que representan información irrelevante, etc. Esto hace que ofrezcan numerosas ventajas y que este tipo de tecnología se esté aplicando a múltiples áreas. Estas ventajas incluyen [39] :

- **Aprendizaje Adaptativo:** Capacidad de aprender a realizar tareas basadas en un entrenamiento o una experiencia. Esta es una de las características más atractivas de las redes de neuronas. Las redes neuronales son sistemas dinámicos autoadaptativos. Son adaptables debido a la capacidad de autoajustarse de las neuronas. Son dinámicos, ya que son capaces de estar constantemente cambiando para adaptarse a las nuevas condiciones.

La función del diseñador es únicamente la obtención de la arquitectura apropiada. Nos es problema del diseñador el cómo aprenderá a discriminar la red; sin embargo, sí es necesario que desarrolle un buen algoritmo de aprendizaje que proporcione la capacidad de discriminar de la red mediante un entrenamiento con patrones.

- **Autoorganización:** Las redes neuronales usan su capacidad de aprendizaje adaptativo para autoorganizar la información que reciben durante el aprendizaje y/o la operación. Mientras que el aprendizaje es una modificación de cada elemento procesal (neurona), la autoorganización consiste en la modificación de la red neuronal completa para llevar a cabo un objetivo específico.

Cuando las redes neuronales se usan para reconocer ciertas clases de patrones, ellas se autoorganizan la información usada. Esta autoorganización provoca la *generalización*: facultad de las redes neuronales de responder apropiadamente cuando se les presentan datos o situaciones a los que no había



sido expuestas anteriormente. El sistema puede generalizar la entrada para obtener una respuesta cuando se tienen que solucionar problemas en los cuales la información de entrada es poco clara o incluso incompleta.

- **Tolerancia a fallos:** La destrucción parcial de una red conduce a una degradación de su estructura; sin embargo, algunas capacidades de la red se pueden retener, incluso sufriendo un gran daño. Las redes neuronales son los primeros métodos computacionales con capacidad inherente de tolerancia a fallos, esto se debe a que tienen su información distribuida en las conexiones entre neuronas, existiendo cierto grado de redundancia en este tipo de almacenamiento.

Estas características permiten distinguir dos aspectos respecto a la tolerancia a fallos: primero, las redes pueden aprender a reconocer patrones con ruido, distorsionados o incompletos, ésta es una tolerancia a fallos respecto a los datos. Segundo, puede seguir realizando su función (con cierta degradación) aunque se destruya parte de la red.

- **Operación en tiempo real:** Los computadores neuronales pueden ser realizados en paralelo, y se diseñan y fabrican máquinas con hardware especial para obtener esta capacidad.

Las Redes de Neuronas Artificiales pueden realizar grandes procesos con datos de forma muy rápida, esto se debe a su implementación paralela. De todos los métodos posibles, es la mejor alternativa para el reconocimiento y clasificación de patrones a tiempo real.

- **Fácil inserción dentro de la tecnología existente:** Se pueden obtener chips especializados para redes neuronales que mejoran su capacidad en ciertas tareas. Ello facilitará la integración modular en los sistemas existentes.



Debido a que una red puede ser rápidamente entrenada, comprobada, verificada y trasladada a una implementación hardware de bajo coste, es fácil insertar redes neuronales en aplicaciones específicas dentro de sistemas existentes. De esta manera, las redes neuronales se pueden utilizar para mejorar sistemas de forma incremental, y cada paso puede ser evaluado antes de acometer un desarrollo más amplio.



### 2.4.3. TIPOS Y APLICACIÓN DE LAS REDES MÁS IMPORTANTES

Existen un gran número y variedad de aplicaciones de las redes neuronales, es decir, son una tecnología computacional emergente que puede tener tanto aplicaciones comerciales como militares. Esto se debe en gran parte a que se pueden desarrollar en un periodo razonable de tiempo, y pueden realizar tareas concretas mejor que otras tecnologías convencionales. Cuando se implementa mediante hardware (redes neuronales en chips VLSI), presentan un alto grado de paralelismo en el procesamiento de datos y una elevada tolerancia a fallos del sistema, lo que hará posible insertar redes neuronales de bajo coste en sistemas existentes y recientemente desarrollados.

Hay muchos tipos diferentes de redes neuronales, cada uno de los cuales tiene una aplicación particular más apropiada. A continuación mostraremos aquellas áreas de aplicación para las cuales los computadores neuronales se están utilizando, o pueden ser utilizados en un futuro inmediato [24] :

- a) Reconocimiento de patrones.
- b) Bases de datos de conocimiento para información estocástica.
- c) Control de robots.
- d) Toma de decisiones.
- e) Filtrado de señales.
- f) Segmentación, compresión y fusión de datos.
- g) Interfaces adaptativas para sistemas hombre/máquina.

A continuación se muestra una tabla resumen de los tipos de redes más conocidos, comentando de forma esquematizada las aplicaciones más importantes de cada una, sus ventajas e inconvenientes y quién o quiénes fueron sus creadores y en qué año.



Nombre de la red	Año	Aplicaciones más importantes	Comentarios	Limitaciones	Inventada / Desarrollada por
<b>Avalancha</b>	1967	Reconocimiento de habla continúa. Control brazos robot.	Ninguna red sencilla puede hacer todo esto.	No es fácil alterar la velocidad o interpolar el movimiento.	Stephen Grossberg.
<b>Teoría Resonancia Adaptativa (ART)</b>	1986	Reconocimiento de patrones (radar, sonar, etc.).	Sofisticada. Poco utilizada.	Sensible a la translación, distorsión y escala.	Gail Carpenter, Stephen Grossberg.
<b>Adaline/Madeline</b>	1960	Filtrado de señales. Ecuilizador adaptativo. Modems.	Rápida, fácil de implementar con circuitos analógicos o VLSI	Sólo es posible clasificar espacios linealmente separados.	Bernard Widrow.
<b>Back propagation</b>	1974-85	Síntesis de voz desde texto. Control de robots. Predicción. Reconocimiento de patrones.	Red más popular. Numerosas aplicaciones con éxito. Facilidad de aprendizaje. Potente.	Necesita mucho tiempo para aprendizaje y muchos ejemplos.	Paul Werbos, David Parker, David Rumelhart.
<b>Memoria asociativa bidireccional</b>	1985	Memoria heteroasociativa de acceso por contenido.	Aprendizaje y arquitecturas simples.	Baja capacidad de almacenamiento. Los datos deben ser codificados.	Bart Kosko.
<b>Máquinas de Boltzmann y Cauchy</b>	1985-86	Reconocimiento de patrones (imágenes, radar y sonar). Optimización	Redes simples. Capacidad de representación óptima de patrones.	La máquina de Boltzmann necesita un tiempo muy largo de aprendizaje.	Jeffrey Hinton, Terry Sejnowski, Harold Szu.
<b>Brain-Estate-in-a-Box</b>	1977	Extracción de conocimiento de bases de datos	Posiblemente mejor realización que las redes de Hopfield.	Realización y potenciales aplicaciones no estudiadas totalmente.	James Anderson
<b>Cerebellatron</b>	1969	Control de movimiento de los brazos de un robot	Semejante a la Avalancha.	Requiere complicadas entradas de control.	David Marr, James Albus, Andres Pellionez.
<b>Counter-propagation</b>	1986	Compresión de imágenes .	Combinación de Perceptron y TPM	Numerosas neuronas y conexiones.	Robert Hecht-Nielsen
<b>Hopfield</b>	1982	Reconstrucción de patrones y optimización	Puede implementarse en VLSI. Fácil de conceptualizar.	Capacidad y estabilidad.	John Hopfield
<b>Neocognitron</b>	1978-84	Reconocimiento de caracteres manuscritos.	Insensible a la translación, rotación y escala.	Requiere muchos elementos de proceso, niveles y conexiones.	K. Fukushima
<b>Perceptron</b>	1957	Reconocimiento de caracteres impresos	La red más antigua. Construida en HW.	No puede reconocer caracteres complejos.	Frank Rosenblatt
<b>Self-Organizing-Map (SOM). Topology-Preserving-Map (TPM)</b>	1980-84	Reconocimiento de patrones, codificación de datos, optimización.	Realiza mapas de características comunes de los datos aprendidos.	Requiere mucho entrenamiento.	Teuvo Kohonen

**Tabla 1: Redes y Aplicaciones**

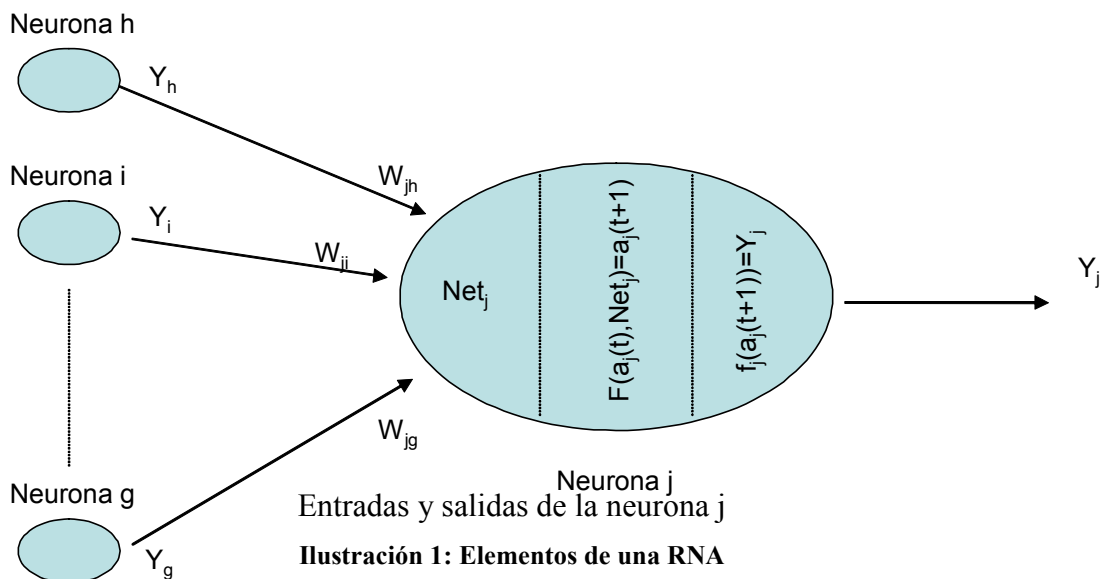


#### 2.4.4. ELEMENTOS Y ESTRUCTURA DE UNA RED NEURONAL ARTIFICIAL

La gran diferencia entre una máquina conexionista, es decir, una máquina neuronal y los programas de computador convencionales es que éstas “elaboran”, en cierta medida, la información de entrada para obtener una salida o respuesta. No se trata de la aplicación ciega y automática de un algoritmo; de hecho, el proceso de elaboración de la información recibida depende de las distintas características, tanto estructurales como funcionales, de la red.

Existen modelos muy diversos de redes de neuronas en los cuales se siguen filosofías de diseño, reglas de aprendizaje y funciones de construcción de las respuestas muy distintas [21] .

#### Elementos de una RNA



$a_i(t)$ : valor o estado de activación.

$f_i$ : función de salida.

$y_i$ : señal de salida.

$W_{ji}$ : peso.

$Net_j$ : entrada total.  $Net_j: \sum y_i * W_{ji}$ .

**Función de activación (F): determina el nuevo estado de activación  $a_j(t+1)$ .**



La dinámica que rige la actualización de estados de las neuronas puede ser de dos tipos: asíncrono y síncrono. De la primera forma, las neuronas, de forma independiente, evalúan su estado continuamente según le va llegando la información. En la segunda opción, la información también llega de forma continua, pero los cambios en los estados se realizan simultáneamente, como si existiera un reloj interno que decidiera cuando deben cambiar su estado.

A continuación se describirán los conceptos básicos de los distintos componentes de un modelo de red neuronal artificial.

- **Unidades de proceso: la neurona artificial:**

Su trabajo es simple y único, y consiste en recibir las entradas de las células vecinas y calcular un valor de salida, el cual es enviado a las células oportunas. Normalmente, cualquier sistema está modelado en tres tipos de unidades: entradas (reciben señales desde el entorno), salidas (envían señales fuera del sistema) y ocultas (aquellas cuyas entradas y salidas están dentro del sistema).

Se denomina capa o nivel al conjunto de neuronas cuyas entradas provienen de la misma fuente (pudiendo ser otra capa de neuronas) y cuyas salidas se dirigen al mismo destino (pudiendo ser también otra capa de neuronas).

- **Estado de activación:**

El estado de activación viene representado por un vector de  $N$  números reales ( $A_t$ ). Cada elemento representa la activación de una neurona en el tiempo  $t$ .  $A_t = (a_1(t), a_2(t), \dots, a_n(t))$ . El procesamiento que realiza la red se ve como la evolución de un patrón de activación en el conjunto de unidades que lo componen a través del tiempo.

Podemos afirmar que toda neurona que se encuentra en una red se halla en cierto estado. Simplificando, los estados posibles son el de reposo y el de excitado. Se





puede diferenciar entre dos tipos de valores de activación: continuos o discretos. Los valores discretos suelen tomar un conjunto pequeño de valores o valores binarios; en estos, el valor 1 indicaría un estado activo, y se caracteriza por la emisión de un impulso por parte de la neurona, mientras que el valor 0 indica una neurona pasiva. En el caso de los valores de activación continuos se considera un conjunto de valores continuos, a los que se les suele asignar un valor que oscile entre el intervalo  $[0,1]$  o en el intervalo  $[-1,1]$ , generalmente siguiendo una función sigmoideal.

- **Función de salida o de transferencia:**

La función de salida o transferencia transforma estado actual de activación  $a_1(t)$  en una señal de salida  $y_1(t)$ , es decir:  $y_1(t) = f_1(a_1(t))$ . El vector que contiene las salidas de todas las neuronas en un instante  $t$  es:  $Y(t) = (f_1(a_1(t)), f_2(a_2(t)), \dots, f_n(a_n(t)))$ .

Las funciones de salida más típicamente utilizadas son cuatro:

1. Función escalón: se asocia a neuronas binarias, ya que cuando la suma de las entradas es mayor o igual que el umbral de la neurona, la activación es 1, y si es menor, la activación es 0.
2. Función lineal (identidad) y mixta: la identidad responde a la función  $f(x) = x$ . En las neuronas con función mixta hay un límite inferior y otro superior. Si la suma de las señales de entrada es menor que el límite inferior, la activación es 0. Si es mayor que el límite superior, es 1, y si está comprendida entre ambos límites, es la función identidad de la suma de las señales de entrada.
3. La función continua (sigmoideal): la mayoría de los valores del estímulo de entrada (variable independiente), el valor dado por la función es cercano a uno de los valores asintóticos. Esto hace que en la mayoría de los casos, el valor de salida este comprendido en la zona alta o baja del sigmoide. De hecho, cuando la pendiente es



elevada esta función tiende a la función escalón. La derivada de esta función es siempre positiva y cercana a 0 para los valores grandes positivos y negativos; además, toma su valor máximo cuando  $X$  es 0. Esto permite utilizar las reglas de aprendizaje definidas para la función escalón, con la ventaja, de que en esta función la derivada sí está definida en todo el intervalo.

4. La función de transferencia Gaussiana: los centros y anchuras de estas funciones se pueden adaptar más que las funciones sigmoideas. Mapeos que suelen requerir dos niveles ocultos utilizando neuronas con funciones de transferencia sigmoideas.

- **Conexiones entre neuronas:**

Todas las conexiones de una RNA tienen asociado un peso, que es lo que permite a la neurona adquirir conocimiento.

Cada conexión entre una neurona  $i$  con una  $j$  esta ponderada por un peso  $w_{ij}$ . Como regla general, se considera que el efecto de cada señal es aditivo, de manera que, la entrada neta que recibe una neurona ( $net_j$ ) es la suma del producto de cada señal individual por el valor del peso que conecta ambas neuronas:

$$net_j = \sum_i^n W_{ij} \cdot y_i$$

Esta regla es conocida como *regla de propagación*.

Suele utilizarse una matriz de pesos  $W$  con todos los pesos que reflejan la influencia de una neurona sobre otra. Si el peso  $w_{ij}$  es positivo, la neurona  $i$  está activada, por lo que la neurona  $j$  recibirá una señal de  $i$  que tenderá a activarla. Si por el contrario  $w_{ij}$  es negativo, la neurona  $i$  estará activada, entonces esta enviará una señal a  $j$  que tenderá a desactivarla. Por último  $w_{ij}$  es igual a 0 podemos considerar que no existe conexión entre neuronas.



• **Función activación:**

Es necesario tener una regla que combine las entradas con el estado actual de la neurona para producir un nuevo estado de activación, esta es la función de activación. Esta función  $F$  produce un nuevo estado de activación en la neurona a partir del estado que existía ( $a_i$ ) y la combinación de las entradas con sus pesos asociados ( $Net_i$ ):

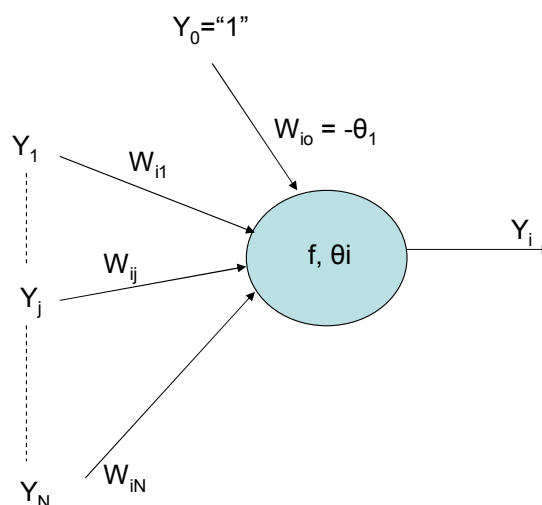
$$a_i(t+1) = F(a_i(t), Net_i)$$

En muchas ocasiones  $F$  es la función identidad, lo que implica que el estado anterior no se tenga en cuenta, por lo que el estado de activación de una neurona en  $t+1$  coincidirá con el  $Net$  de la misma  $t$ . En este caso, se le pasará directamente  $Net$  a la función de salida  $f$ . Al añadirle el umbral de activación de la neurona  $i$  ( $\theta_i$ ), la salida de  $i$  ( $y_i$ ) quedará:

$$y_i(t+1) = f(Net_i - \theta_i) = f(\sum w_{ij} y_j(t) - \theta_i).$$

Para simplificar la salida de la neurona se suele considerar una neurona ficticia con valor de salida con  $y_o=1$  y con un peso asociado a la conexión de  $-\theta_i$ . Por lo que la expresión de la salida quedaría:

$$y_i(t+1) = f(\sum w_{ij} y_j(t) - \theta_j * 1) = f(\sum w_{ij} y_j(t)) = f(Net_i)$$



**Ilustración 2: Salida de una Neurona**



- **Regla de aprendizaje:**

En las Redes de Neuronas Artificiales se puede considerar que el conocimiento reside en los pesos que tiene las conexiones que existen entre las diferentes neuronas de nuestro sistema. De lo que podemos concluir que, todo proceso de aprendizaje implica cierto número de cambios en estas conexiones. Lo que significa, que una RNA aprende modificando los pesos de sus interconexiones. Cada modelo de red dispone de su o sus propias técnicas de aprendizaje.



#### 2.4.5. ESTRUCTURA DE UNA RED NEURONAL.

La distribución de neuronas dentro de la red se realiza formando niveles o capas de un número determinado de neuronas cada una. Existen tres tipos de capas:

- **De entrada:** Reciben información proveniente de las fuentes externas de la red.
- **Ocultas:** Son internas a la red y no tienen contacto con el exterior. Puede haber de cero a “n” capas ocultas en una red. Las neuronas de las capas ocultas pueden estar interconectadas de distintas maneras, lo que determina, junto con su número, las distintas tipologías de redes neuronales.
- **De salida:** Transfieren información de la red al exterior.

Se dice que una red está totalmente conectada si todas las neuronas si todas las salidas desde un nivel llegan a todos y cada uno de los nodos del nivel siguiente.

En función de cómo se interconectan unas capas con otras, podemos hablar de *redes con propagación hacia atrás o recurrentes* (feed-back) y de *redes con propagación hacia delante o redes en cascada* (feed-forward). En las redes en cascada la información fluye unidireccionalmente de una capa a otra (desde la capa de entrada a las capas ocultas y de éstas a la capa de salida), y además, no se admiten conexiones intracapa. En las redes recurrentes la información puede volver a lugares por los que ya había pasado, formando bucles, y se admiten las conexiones intracapa (laterales), incluso de una unidad consigo misma [20] .

Las conexiones entre una capa y otra pueden ser totales, es decir, que cada unidad se conecta con todas las unidades de la capa siguiente, o parciales, en las cuales una unidad se conecta con sólo algunas de las capas de la unidad siguiente.

En función de cómo las unidades computan su activación con respecto al tiempo, podemos distinguir entre redes síncronas o asíncronas (también conocidas como probabilísticas). Hablaremos de redes sincronías si en cada ciclo, o epoc, de aprendizaje



se calcula la activación de todas las unidades de una capa; por el contrario, nos referiremos a redes asíncronas si cada unidad de proceso tiene una cierta posibilidad de computar su activación en cada ciclo de aprendizaje.

Esta distinción tiene bastante relación con la anterior, ya que las redes síncronas suelen emplear reglas de aprendizaje deterministas, y los métodos estadísticos en cambio, se aplican en las redes asíncronas. Algunos autores hablan de métodos de aprendizaje deterministas haciendo referencia al modo en que computan, paso a paso, las modificaciones en los pesos de las conexiones. Los métodos estadísticos hacen cambios pseudoaleatorios en las conexiones y retienen los cambios sólo si mejoran la respuesta del sistema, pueden resultar útiles en cuanto permiten alcanzar soluciones globalmente óptimas, a diferencia de las soluciones localmente óptimas típicas de los métodos deterministas [21] .



#### 2.4.6. PROCESO DE APRENDIZAJE.

En cualquier red neuronal cabe distinguir la fase o proceso de aprendizaje, opcionalmente una fase de test, y la fase de aplicación[40] .

El aprendizaje requiere la presentación repetida de un número relativamente amplio de ejemplos diferenciados, hasta lograr una respuesta satisfactoria del sistema (según la medida de error o función de energía utilizada). Cada una de estas presentaciones se las denomina *Ciclos o Epocs*. A este proceso se le denomina entrenamiento y es el encargado de realizar modelos eficaces y obtener las características de la red, que ante un conjunto de entradas obtiene una salida. De esta manera ante una entrada semejante con los parámetros de la red aprendidos en la etapa de entrenamiento, esta será capaz de obtener una salida.

Es usual disponer de un conjunto de datos distintos a los usados para el entrenamiento, para los cuales se conoce la respuesta correcta, y se usan como test, evaluando con ello si la red responde adecuadamente frente a datos distintos a los usados durante el aprendizaje (generalización), si es así se considera que la red funciona bien y se puede pasar a la fase de aplicación (recall), es decir, se puede usar para obtener un resultado frente a datos totalmente nuevos para los que no se conoce la respuesta correcta.

Independientemente del tipo de aprendizaje usado, una característica esencial de la red es la regla de aprendizaje usada, que indica cómo se modifican los pesos de las conexiones en función de los datos usados en la entrada, es decir, de la historia de aprendizaje de la red.

Cualquiera que sea el tipo de aprendizaje usado, una característica esencial de la red es la regla de aprendizaje usada, que indica cómo se modifican los pesos de las conexiones en función de los datos usados en la entrada. Existen dos tipos de reglas: las que responden a lo que se conoce como *aprendizaje supervisado*, y las correspondientes



a un *aprendizaje no supervisado*. La diferencia entre estos dos tipos radica en si existe o no un agente externo (supervisor) que controle el proceso aprendizaje de la red.

Otro criterio para clasificar la red es que el aprendizaje sea *ON-LINE*, la red puede aprender durante su funcionamiento, es decir, los pesos pueden variar siempre que se presente una nueva información en el sistema, que sea un aprendizaje *OFF-LINE*, donde el aprendizaje supone la desconexión de la red. Los pesos permanecen fijos una vez que acaba la fase de entrenamiento. En la de funcionamiento no cambian.

- **Redes con aprendizaje supervisado:**

Entrenamiento controlado por un agente externo que determina la respuesta que debería generar la red a partir de una determinada entrada. El supervisor comprueba la salida y si no coincide modifica los pesos para conseguir que la salida se aproxime a la deseada.

- **Redes con aprendizaje no supervisado:**

No requieren influencia externa para ajustar los pesos de las conexiones entre sus neuronas. Se dice por tanto, que estas redes son capaces de autoorganizarse. Existen varias posibilidades en cuanto a la interpretación de la salida de estas redes, que dependen de su estructura y del algoritmo de aprendizaje empleado.

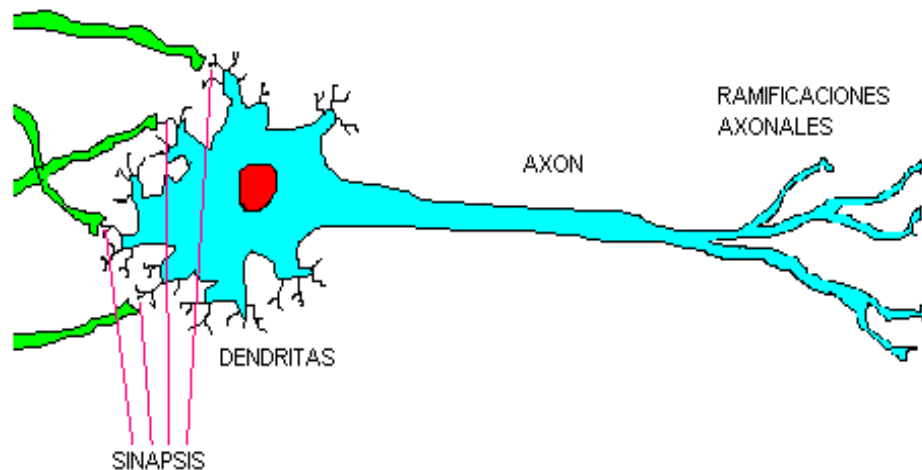




#### 2.4.7. REDES NEURONALES BIOLÓGICAS.

A grandes rasgos, recordemos que el cerebro humano se compone de decenas de billones de neuronas interconectadas entre sí formando circuitos o redes que desarrollan funciones específicas.

Una neurona típica recoge señales procedentes de otras neuronas a través de una pléyade de delicadas estructuras llamadas dendritas. La neurona emite impulsos de actividad eléctrica a lo largo de una fibra larga y delgada denominada axón, que se escinde en millares de ramificaciones.



**Ilustración 3: Neurona Biológica**

Las extremidades de estas ramificaciones llegan hasta las dendritas de otras neuronas y establecen unas conexiones llamadas Sinapsis, en las cuales se produce una transformación del impulso eléctrico en un mensaje neuroquímico, mediante la liberación de unas sustancias llamadas neurotransmisor.

El efecto de los neurotransmisores sobre la neurona receptora puede ser excitatorio o inhibitorio, y es variable (la intensidad del efecto depende de numerosos factores que no sería oportuno describir aquí), de manera que podemos hablar de la fuerza o efectividad de una Sinapsis. Las señales excitatorias e inhibitorias recibidas por una neurona se combinan, y en función de la estimulación total recibida, la neurona toma un cierto nivel de activación, que se traduce en la generación de breves impulsos



nerviosos con una determinada frecuencia o tasa de disparo, y su propagación a lo largo del axón hacia las neuronas con las cuales sinapta.

De esta manera la información se transmite de unas neuronas a otras y va siendo procesada a través de las conexiones sinápticas y las propias neuronas. El aprendizaje de las redes neuronales se produce mediante la variación de la efectividad de las Sinopsis, de esta manera cambia la influencia que unas neuronas ejercen sobre otras, de aquí se deduce que la arquitectura, el tipo y la efectividad de las conexiones en un momento dado, representan en cierto modo la memoria o estado de conocimiento de la red.



#### 2.4.8. ANALOGÍA CON LAS REDES NEURONALES BIOLÓGICAS.

Las neuronas se modelan mediante unidades de proceso. Cada unidad de proceso se compone de una red de conexiones de entrada, una función de red (de propagación), encargada de computar la entrada total combinada de todas las conexiones, un núcleo central de proceso, encargado de aplicar la función de activación, y la salida, por dónde se transmite el valor de activación a otras unidades [40].

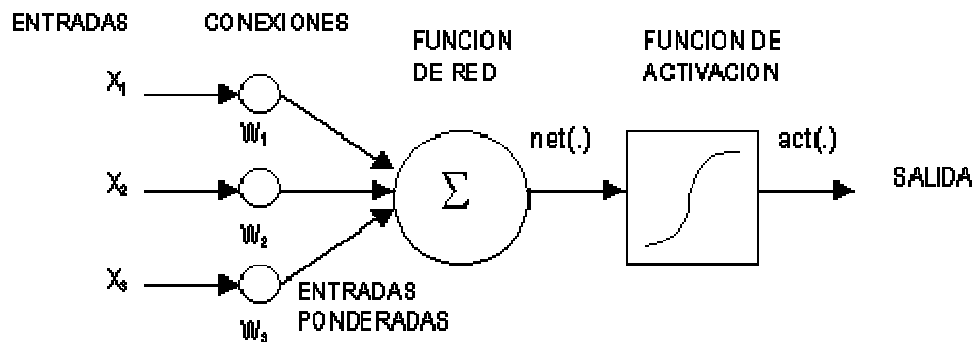


Ilustración 4: Unidad de Proceso Típica

La función de red es típicamente el sumatorio ponderado, mientras que la función de activación suele ser alguna función de umbral o una función sigmoide.

- Función de propagación de red: Calcula el valor de base o entrada total a la unidad, generalmente como simple suma ponderada de todas las entradas recibidas, es decir, de las entradas multiplicadas por el peso o valor de las conexiones. Equivale a la combinación de las señales excitatorias e inhibitorias de las neuronas biológicas.
- Función de activación: Es quizás la característica principal o definitoria de las neuronas, la que mejor define el comportamiento de la misma. Se usan diferentes tipos de funciones, desde simples funciones simples de umbral a funciones no lineales. Se encarga de calcular el nivel o estado de activación de la neurona en función de la entrada total.



- **Conexiones ponderadas:** hacen el papel de las conexiones sinápticas, el peso de la conexión equivale a la fuerza o efectividad de la Sinopsis. La existencia de conexiones determina si es posible que una unidad influya sobre otra, el valor de los pesos y el signo de los mismos definen el tipo (excitatorio/inhibitorio) y la intensidad de la influencia.
- **Salida:** calcula la salida de la neurona en función de la activación de la misma, aunque normalmente no se aplica más que la función identidad, y se toma como salida el valor de activación. El valor de salida cumpliría la función de la tasa de disparo en las neuronas biológicas.

<i>Redes Neuronales Biológicas</i>	<i>Redes Neuronales Artificiales</i>
Neuronas	Unidades de proceso
Conexiones sinápticas	Conexiones ponderadas
Efectividad de las sinopsis	Peso de las conexiones
Efecto excitatorio o inhibitorio de una conexión	Signo del peso de una conexión
Efecto combinado de las sinapsis	Función de propagación o de red
Activación -> tasa de disparo	Función de activación -> Salida

**Tabla 2: Comparación entre las Neuronas Biológicas y Artificiales**



## 2.5. El Perceptrón

### 2.5.1. EL PERCEPTRON MULTICAPA

Rumelhart, Hinton y Williams (1986) [11] formalizaron un método para que una red del tipo perceptrón multicapa aprendiera la asociación que existe entre un conjunto de patrones de entrada y sus salidas correspondientes. Este método, conocido como *backpropagation error* (propagación del error hacia atrás) --también denominado método de gradiente decreciente--, ya había sido descrito anteriormente por Werbos en 1974 [41] , Parker en 1982 [42] y Le Cun en 1985 [43] , aunque fue el *Parallel Distributed Processing Group* (grupo PDP) --Rumelhart y colaboradores--, quien realmente lo popularizó.

La importancia de la red *backpropagation* consiste en su capacidad de organizar una representación interna del conocimiento en las capas ocultas de neuronas, a fin de aprender la relación que existe entre un conjunto de entradas y salidas. Posteriormente, aplica esa misma relación a nuevos vectores de entrada con ruido o incompletos, dando una salida activa si la nueva entrada es parecida a las presentadas durante el aprendizaje. Esta característica importante es la capacidad de generalización, entendida como la facilidad de dar salidas satisfactorias a entradas que el sistema no ha visto nunca en su fase de entrenamiento.

Un perceptrón multicapa está compuesto por una capa de entrada, una capa de salida y una o más capas ocultas; aunque se ha demostrado que para la mayoría de problemas bastará con una sola capa oculta [12] [13] .

En la ilustración 5 podemos observar un perceptrón típico formado por una capa de entrada, una capa oculta y una de salida.

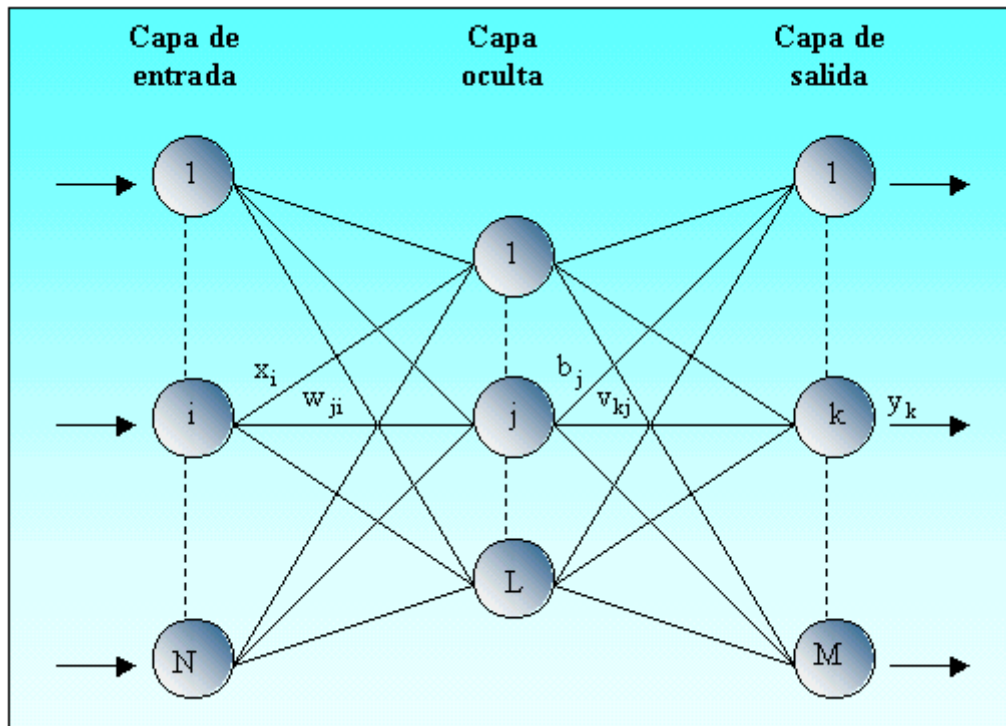


Ilustración 5: Perceptrón Multicapa

En este tipo de arquitectura, las conexiones entre neuronas son siempre hacia delante, es decir, las conexiones van desde las neuronas de una determinada capa hacia las neuronas de la siguiente capa; no hay conexiones laterales --esto es, conexiones entre neuronas pertenecientes a una misma capa--, ni conexiones hacia atrás --esto es, conexiones que van desde una capa hacia la capa anterior. Por tanto, la información siempre se transmite desde la capa de entrada hacia la capa de salida.

En adelante será considerado  $w_{ji}$  como el peso de conexión entre la neurona de entrada  $i$  y la neurona oculta  $j$ , y  $v_{kj}$  como el peso de conexión entre la neurona oculta  $j$  y la neurona de salida  $k$ .

Cuando se presenta un patrón  $p$  de entrada  $X_p: x_{p1}, \dots, x_{pi}, \dots, x_{pN}$ , éste se transmite a través de los pesos  $w_{ji}$  desde la capa de entrada hacia la capa oculta. Las neuronas de esta capa intermedia transforman las señales recibidas mediante la aplicación de una función de activación proporcionando, de este modo, un valor de salida. Este se transmite a través de los pesos  $v_{kj}$  hacia la capa de salida, donde aplicando la misma



operación que en el caso anterior, las neuronas de esta última capa proporcionan la salida de la red. Este proceso se puede explicar matemáticamente de la siguiente manera:

La entrada total o neta que recibe una neurona oculta  $j$ ,  $net_{pj}$ , es:

$$net_{pj} = \sum_{i=1}^N w_{ji} x_{pi} + \theta_j$$

donde  $\theta$  es el umbral de la neurona que se considera como un peso asociado a una neurona ficticia con valor de salida igual a 1.

El valor de salida de la neurona oculta  $j$ ,  $b_{pj}$ , se obtiene aplicando una función  $f(\cdot)$  sobre su entrada neta:

$$b_{pj} = f(net_{pj})$$

De igual forma, la entrada neta que recibe una neurona de salida  $k$ ,  $net_{pk}$ , es:

$$net_{pk} = \sum_{j=1}^L v_{kj} b_{pj} + \theta_k$$

Por último, el valor de salida de la neurona de salida  $k$ ,  $y_{pk}$ , es:

$$y_{pk} = f(net_{pk})$$

El algoritmo *backpropagation* exige que la función de activación sea continua y, por tanto, derivable para poder obtener el error o valor delta de las neuronas ocultas y de salida. Se disponen de dos formas básicas que cumplen esta condición: la función lineal (o identidad) y la función sigmoideal (logística o tangente hiperbólica). En las ilustraciones 6, 7 y 8 se presentan las expresiones matemáticas y la correspondiente representación gráfica de la función lineal, la sigmoideal logística (con límites entre 0 y 1) y la sigmoideal tangente hiperbólica (con límites entre -1 y 1):

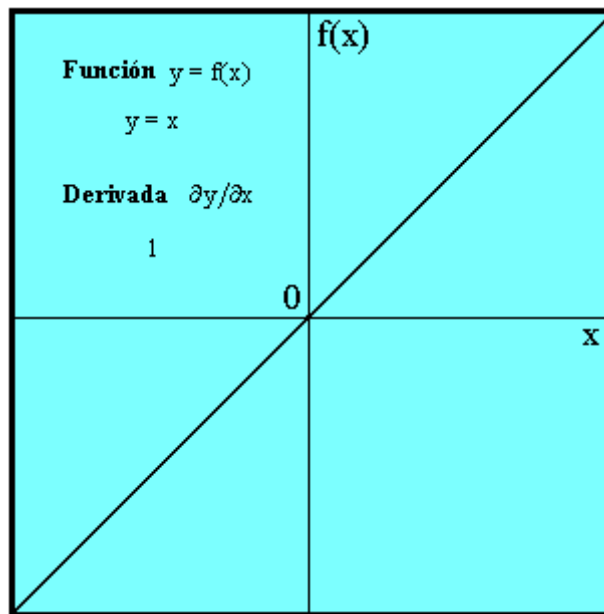


Ilustración 6: Función Lineal

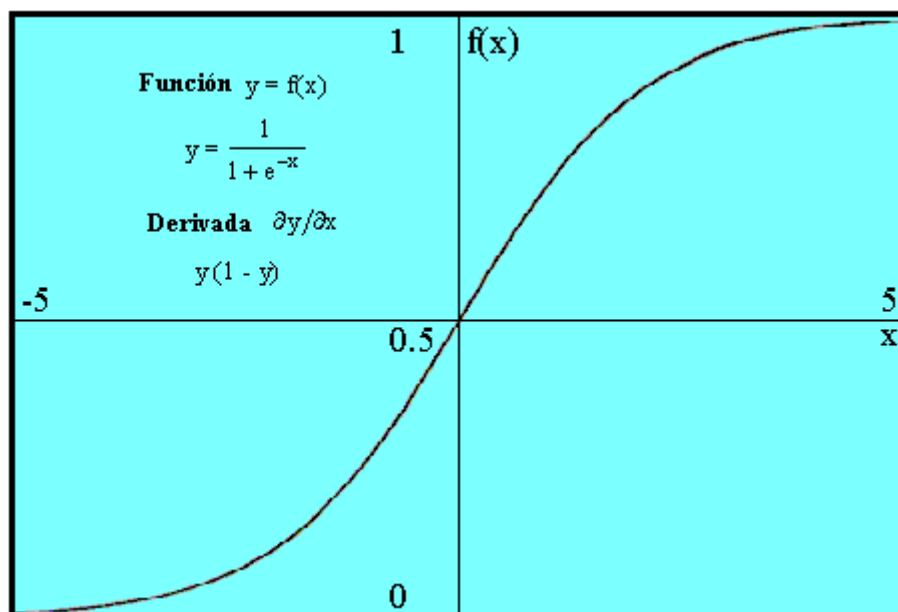
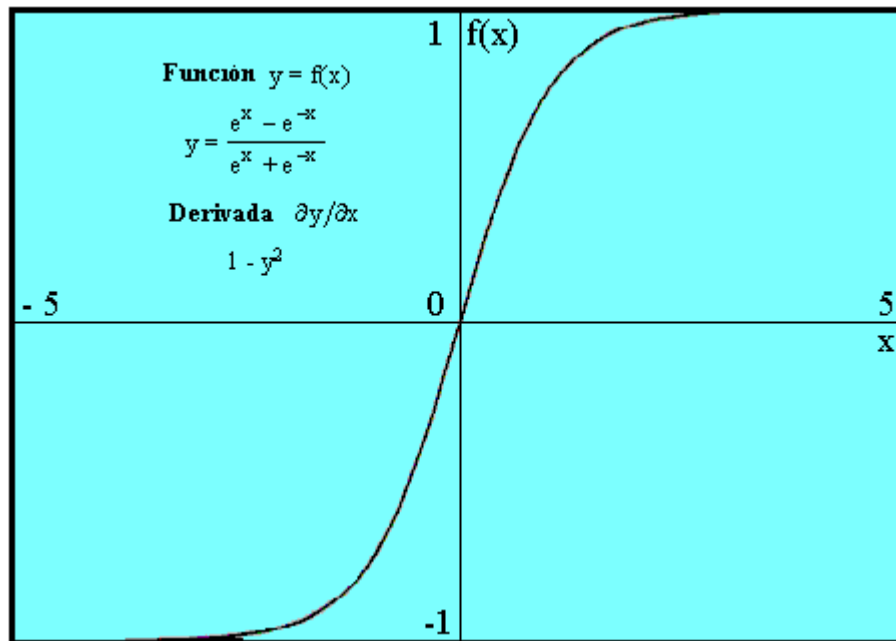


Ilustración 7: Función Sigmoidal logística





**Ilustración 8: Función Sigmoidal Tangente Hiperbólica**

Debemos tener en cuenta que para aprovechar la capacidad de las RNA de aprender relaciones complejas o no lineales entre variables, es absolutamente imprescindible la utilización de funciones no lineales al menos en las neuronas de la capa oculta [44]. Las RNA que no utilizan funciones no lineales, se limitan a solucionar tareas de aprendizaje que implican únicamente funciones lineales o problemas de clasificación que son linealmente separables. Por tanto, en general se utilizará la función sigmoidal (logística o tangente hiperbólica) como función de activación en las neuronas de la capa oculta.

Por su parte, la elección de la función de activación en las neuronas de la capa de salida dependerá del tipo de tarea impuesto. En tareas de clasificación, las neuronas normalmente toman la función de activación sigmoidal.



### 2.5.2. ALGORITMO DE RETROPROPAGACIÓN (BACKPROPAGATION)

El primer algoritmo importante de aprendizaje, para la arquitectura del perceptron, fue propuesta por Rosenblatt en 1958 [45] , aunque debido a sus limitaciones han sido surgiendo diferentes variantes más evolucionadas. Su versión más conocida es sin duda el *algoritmo de retropropagación* o *regla delta generalizada* propuesta por Rumelhart en 1986 [11] . Se trataba de una generalización del algoritmo de la regla delta propuesta por Widrow [46] para poder aplicarla a la redes con conexiones hacia delante (feedforward) con capas ocultas que no tienen conexión con el exterior.

Al hablar de redes de retropropagación o redes de propagación hacia atrás se hace referencia a un algoritmo de aprendizaje más que a una arquitectura determinada. La retropropagación consiste en propagar el error hacia atrás, es decir, de la capa de salida hacia la capa de entrada, pasando por las capas ocultas intermedias y ajustando los pesos de las conexiones con el fin de reducir dicho error.

En la etapa de aprendizaje, el objetivo que se persigue es hacer mínima la discrepancia o error entre la salida obtenida por la red y la salida deseada por el usuario ante la presentación de un conjunto de patrones de entrenamiento. Por este motivo, se dice que el aprendizaje en las redes *backpropagation* es de tipo supervisado, debido a el usuario (o supervisor) determina la salida deseada ante la presentación de un determinado patrón de entrada.

La función de error que se pretende minimizar para cada patrón p viene dada por:

$$E_p = \frac{1}{2} \sum_{k=1}^M (d_{pk} - y_{pk})^2$$

donde  $d_{pk}$  es la salida deseada para la neurona de salida k ante la presentación del patrón p. A partir de esta expresión se puede obtener una medida general de error mediante:



$$E = \sum_{p=1}^P E_p$$

La base matemática del algoritmo *backpropagation* para la modificación de los pesos es la técnica conocida como gradiente decreciente [11]. Teniendo en cuenta que  $E_p$  es función de todos los pesos de la red, el gradiente de  $E_p$  es un vector igual a la derivada parcial de  $E_p$  respecto a cada uno de los pesos. El gradiente toma la dirección que determina el incremento más rápido en el error, mientras que la dirección opuesta -- es decir, la dirección negativa--, determina el decremento más rápido en el error. Por tanto, el error puede reducirse ajustando cada peso en la dirección:

$$-\sum_{p=1}^P \frac{\partial E_p}{\partial w_{ji}}$$

Vamos a ilustrar el proceso de aprendizaje de forma gráfica: El conjunto de pesos que forma una red neuronal puede ser representado por un espacio compuesto por tantas dimensiones como pesos tengamos. Supongamos para simplificar el problema que tenemos una red formada por dos pesos, el paisaje se puede visualizar como un espacio de dos dimensiones. Por otra parte, hemos comentado que el error cometido es función de los pesos de la red; de forma que en nuestro caso, a cualquier combinación de valores de los dos pesos, le corresponderá un valor de error para el conjunto de entrenamiento. Estos valores de error se pueden visualizar como una superficie, que denominaremos superficie del error. Como se muestra en la ilustración 9, la superficie del error puede tener una topografía arbitrariamente compleja.

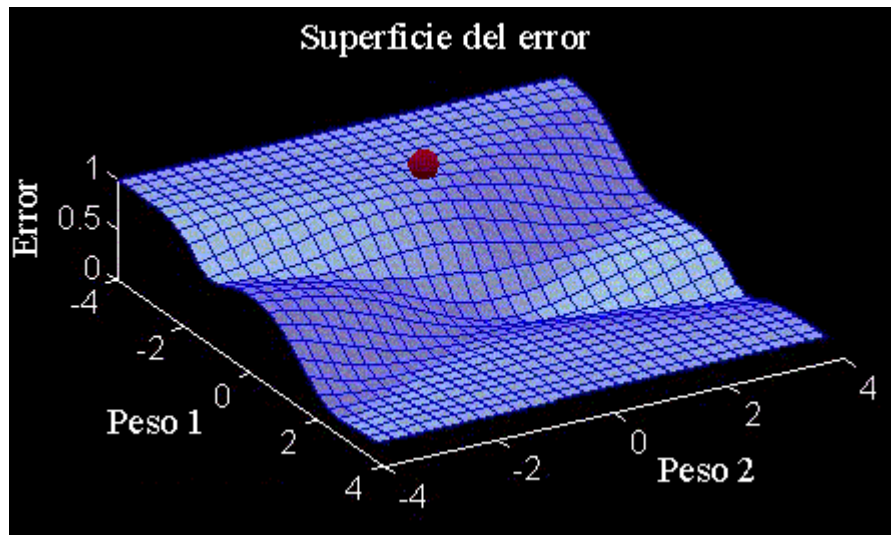


Ilustración 9: Superficie del Error 1

Con la imagen de la superficie del error en mente, el proceso de entrenamiento comienza en un determinado punto, representado por la bola roja, definido por los pesos iniciales de la red (Ilustración 9). El algoritmo de aprendizaje se basa en obtener información local de la pendiente de la superficie --esto es, del gradiente--, y a partir de esa información modificar iterativamente los pesos de forma proporcional a dicha pendiente, a fin de asegurar el descenso por la superficie del error hasta alcanzar el mínimo más cercano desde el punto de partida. La Ilustración 10 muestra el proceso descrito mediante la representación del descenso de la bola roja hasta alcanzar una llanura.

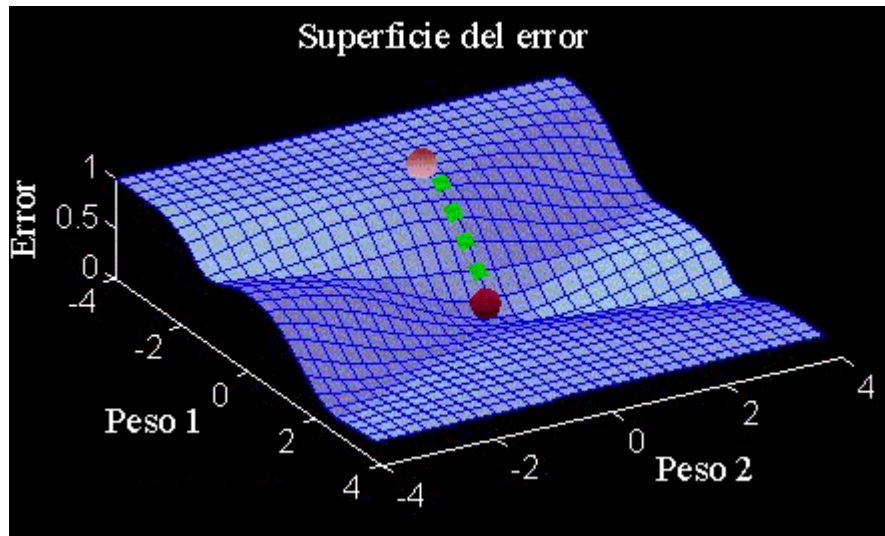


Ilustración 10: Superficie del Error 2

Con un número mayor de pesos el espacio se convierte en un plano multidimensional inimaginable, aunque se seguirían aplicando los mismos principios comentados en el ejemplo.

Un peligro que puede surgir al utilizar el método de gradiente decreciente es que el aprendizaje converja en un punto bajo, sin ser el punto más bajo de la superficie del error. Tales puntos se denominan mínimos locales para distinguirlos del punto más bajo de esta superficie, denominado mínimo global. Sin embargo, el problema potencial de los mínimos locales se dan en raras ocasiones en datos reales [44] .

A nivel práctico, la forma de modificar los pesos de forma iterativa consiste en aplicar la regla de la cadena a la expresión del gradiente y añadir una tasa de aprendizaje  $\eta$ . Así, cuando se trata del peso de una neurona de salida:

$$\Delta v_{kj}(n+1) = \eta \sum_{p=1}^P \delta_{pk} b_{pj}$$

donde

$$\delta_{pk} = (d_{pk} - y_{pk}) f'(net_{pk})$$

y n indica la iteración. Cuando se trata del peso de una neurona oculta:



$$\Delta w_{ji}(n+1) = \eta \sum_{p=1}^P \delta_{pj} x_{pi}$$

donde

$$\delta_{pj} = f'(\text{net}_{pj}) \sum_{k=1}^M \delta_{pk} v_{kj}$$

Se puede observar que el error o valor delta asociado a una neurona oculta  $j$ , viene determinado por la suma de los errores que se cometen en las  $k$  neuronas de salida que reciben como entrada la salida de esa neurona oculta  $j$ . De ahí que el algoritmo también se denomine propagación del error hacia atrás.

Para la modificación de los pesos, la actualización se realiza después de haber presentado todos los patrones de entrenamiento. Este es el modo habitual de proceder y se denomina aprendizaje por lotes o modo *batch*. Existe otra modalidad denominada aprendizaje en serie o modo *on-line* consistente en actualizar los pesos tras la presentación de cada patrón de entrenamiento. En este modo, se debe tener presente que el orden en la presentación de los patrones debe ser aleatorio, puesto que si siempre se siguiese un mismo orden, el entrenamiento estaría viciado a favor del último patrón del conjunto de entrenamiento, cuya actualización, por ser la última, siempre predominaría sobre las anteriores [47].

Con el fin de acelerar el proceso de convergencia de los pesos, Rumelhart, Hinton y Williams [11] sugirieron añadir en la expresión del incremento de los pesos un factor momento,  $\alpha$ , el cual tiene en cuenta la dirección del incremento tomada en la iteración anterior. Así, cuando se trata del peso de una neurona de salida:

$$\Delta v_{kj}(n+1) = \eta \left( \sum_{p=1}^P \delta_{pk} b_{pj} \right) + \alpha \Delta v_{kj}(n)$$

Cuando se trata del peso de una neurona oculta:



$$\Delta w_{ji}(n+1) = \eta \left( \sum_{p=1}^P \delta_{pj} x_{pi} \right) + \alpha \Delta w_{ji}(n)$$



## Capítulo 3. ESTUDIO DE VIABILIDAD DEL SISTEMA

### 3.1. Introducción

El objetivo del Estudio de Viabilidad del Sistema es el análisis de un conjunto concreto de necesidades para proponer una solución a corto plazo, que tenga en cuenta restricciones económicas, técnicas, legales y operativas. La solución obtenida como resultado del estudio puede ser la definición de uno o varios proyectos que afecten a uno o varios sistemas de información ya existentes o nuevos. Para ello, se acota el alcance del sistema, se estudia, si procede, la situación actual, y se realiza de definición de los requisitos del sistema que serán la base para la construcción de la solución.

### 3.2. Establecimiento del alcance del sistema

En esta actividad se estudia el alcance de la necesidad planteada por el cliente o usuario, realizando una descripción general de la misma. Se determinan los objetivos, se inicia el estudio de los requisitos y se identifican las unidades organizativas afectadas estableciendo su estructura.

Se analizan las posibles restricciones, tanto generales como específicas, que puedan condicionar el estudio y la planificación de las alternativas de solución que se propongan. Si la justificación económica es obvia, el riesgo técnico bajo, se esperan pocos problemas legales y no existe ninguna alternativa razonable, no es necesario profundizar en el estudio de viabilidad del sistema, analizando posibles alternativas y realizando una valoración y evaluación de las mismas, sino que éste se orientará a la especificación de requisitos, descripción del nuevo sistema y planificación.

Se detalla la composición del equipo de trabajo necesario para este proceso y su planificación. Finalmente, con el fin de facilitar la implicación activa de los usuarios en la definición del sistema, se identifican sus perfiles, dejando claras sus tareas y responsabilidades.





### 3.2.1. ESTUDIO DE LA SOLICITUD

Este proyecto surge como respuesta a una serie de nuevas necesidades que surgen de la utilización de los proyectos anteriormente creados en el Departamento de Informática para la predicción de impactos. La idea final de todos estos proyectos es la automatización de los procesos de entrenamiento y testeo de redes de neuronas.

La finalidad de este proyecto es proporcionar al usuario/investigador una herramienta que le ayude a intentar comprender el comportamiento de las redes, y como varia este en función de su configuración, entrenamiento, y datos del mismo. Un punto importante es ayudar al usuario a conocer como influye cada una de las variables en el resultado final de la predicción/simulación.

Se solicita desde el Departamento de Informática una nueva herramienta que cubra los siguientes objetivos:

- Automatización de los procesos de entrenamiento y testeo de una red de neuronas implementada en NeuroSolutions.
- Valida en procesos de simulación y clasificación.
- Debe proporcionar la información necesaria para que el investigador pueda analizar el comportamiento de la red
- Solución modularizada, intuitiva y sencilla.

Las restricciones del proyecto inicialmente proporcionadas son:

- Económico: No aplica.
- Técnico:
  - La herramienta debe ser una aplicación de escritorio sobre entorno Windows.
  - Debe interactuar con Neurosolutions.
- Operativo: Un solo programa debe proporcionar toda la funcionalidad.
- Legal: No aplica.

Inicialmente todo hace indicar que el proyecto es perfectamente viable.



### 3.2.2. IDENTIFICACIÓN DEL ALCANCE DEL SISTEMA

La solución a desarrollar debe ser una aplicación de escritorio. Debe estar implementada mediante un único lenguaje de programación, lo que hará más sencillo su mantenimiento y el desarrollo de futuras adaptaciones.

Para permitir al usuario el análisis del comportamiento de una red el sistema debe generar información y datos que el investigador pueda explotar. Todo esto se proporcionará mediante ficheros Excel. Esta información debe cubrir los siguientes aspectos:

- Resultados de ensayo
- Resúmenes de estudios
- Resumen de varios estudios.
- Análisis de la Fortaleza frente a valores atípicos

Estos informes deben adaptarse en función de si el proceso es de clasificación o predicción.

El sistema debe poder recibir como de fichero de entrada cualquiera formato que sea accesible desde Excel. A partir del mismo, el sistema generará las baterías de pruebas que formarán parte de un estudio. El estudio se ejecutará desde la misma herramienta, para ello, esta debe poder comunicarse con Neurosolutions, y una vez finalizado, proporcionará todos los informes anteriormente citados que permitirán al usuario obtener conclusiones del estudio.

El sistema debe estar compuesto por tres grandes módulos que deben poder ejecutarse tanto de forma secuencial como atómica. Cada modulo tendrá los siguientes objetivos:

- Generación de los ficheros para un estudio
- Entrenamiento y generación de los informes del estudio.
- Generación de resúmenes transversales a diferentes estudios.



### 3.2.3. IDENTIFICACIÓN DE LOS INTERESADOS EN EL SISTEMA (STAKEHOLDERS)

En este punto se lleva a cabo una identificación de todas aquellas personas o entidades interesadas de alguna forma en el proyecto. Se describen a continuación:

- **Cliente:** Es la persona que realiza la solicitud de desarrollo del sistema, en este caso, el cliente es Israel González Carrasco, profesor de la Universidad Carlos III de Madrid.
- **Usuarios gestores:** Son aquellas personas que harán uso de la aplicación.
- **Tutor y coordinador del proyecto:** Israel González Carrasco
- **Autor del Proyecto:** Roberto Romero Martínez. Es la persona encargada de la realización del proyecto, siendo además el máximo responsable del mismo y de todos los productos generados durante su desarrollo.

### 3.3. Estudio de la situación actual

La situación actual es el estado en el que se encuentran los sistemas de información existentes en el momento en el que se inicia su estudio. Teniendo en cuenta el objetivo del estudio de la situación actual, se realiza una valoración de la información existente acerca de los sistemas de información afectados. En función de dicha valoración, se especifica el nivel de detalle con que se debe llevar a cabo el estudio.

Como resultado de esta actividad se genera un diagnóstico, estimando la eficiencia de los sistemas de información existentes e identificando los posibles problemas y las mejoras.

#### 3.3.1. VALORACIÓN DEL ESTUDIO DE LA SITUACIÓN ACTUAL

El Departamento de Informática utiliza una serie de herramientas desarrolladas en anteriores PFC para interactuar de forma masiva con RNAs, con ello, se intenta comprender de una forma más exacta el funcionamiento de las redes, su



comportamiento en distintos ambientes, y como estos pueden afectar a sus resultados. Algunos de los proyectos/estudios anteriormente utilizados han sido: “Simulación de impactos en materiales mediante redes neuronales”, “Parametrización del aprendizaje de una red neuronal para la predicción de impactos sobre materiales” y “Aplicación de un perceptron multicapa para la resolución de problemas de impacto balístico”.

El objetivo principal de cualquier proyecto relacionado con la simulación/predicción es obtener resultados fiables a partir de experimentos planteados, es decir, obtener los resultados sin que los estos tengan que ser llevados a cabo por su método tradicional. La principal cualidad que debe cumplir cualquier simulación reside en lograr resultados fidedignos con un menor coste de recursos (tiempo, personal, monetario...) que los que acarrearía la realización del experimento propiamente dicha.

Los proyectos existentes buscan comprobar el grado de fiabilidad que muestra una red de neuronas a la hora de predecir el patrón de comportamiento que se genera en el lanzamiento de proyectiles ante una superficie. Buscan lograr encontrar la mejor configuración de la red puesto que no existe ningún método exacto que permita afirmar que una red, y/o configuración de la misma, es la más óptima para solucionar un determinado problema.

### 3.3.2. REALIZACIÓN DEL DIAGNÓSTICO DE LA SITUACIÓN ACTUAL

La situación actual tiene una serie de inconvenientes. El principal de ellos es que las herramientas actuales permiten en el entrenamiento y testeo de redes de forma casi automática en la realización de un estudio, pero no para la realización de una batería de estudios. Además no ayudan de ninguna manera al usuario/investigador a la explotación de los resultados de los estudios. Obligan al usuario a trabajar con una gran cantidad de ficheros de salida por estudio, lo que hace muy complejo sacar conclusiones tanto unitarias como transversales a un conjunto de ellos.

Como método para intentar descubrir cómo influyen las variables de un estudio en los resultados del mismo se han utilizarlo el método de “Fortaleza frente a valores



atípicos” o “Criticidad de las variables”. En ninguno de los estudios se ha contrastado de forma fehaciente la exactitud o validez de este método.

A nivel técnico, algunas de las herramientas obligan al usuario a trabajar con distintas herramientas para poder ejecutar un único estudio, lo que añade complejidad a su uso.

### 3.4. Definición de los Requisitos del Sistema

Una vez realizado el estudio de la solicitud del cliente se realiza la extracción de requisitos con el fin de cumplir con las funcionalidades debe proporcionar la aplicación.

Esta actividad incluye la determinación de los requisitos generales, mediante una serie de sesiones de trabajo con los usuarios participantes, que hay que planificar y realizar. Una vez finalizadas, se analiza la información obtenida definiendo los requisitos y sus prioridades, que se añaden al catálogo de requisitos que servirá para el estudio y valoración de las distintas alternativas de solución que se propongan.

Esta extracción de requisitos se realiza también con el objetivo de orientar al cliente, permitiendo al equipo de desarrollo obtener una lista de requisitos detallada, completa y sin ambigüedades. Estos requisitos proporcionarán una visión general de la aplicación, sin ahondar en aspectos técnicos, estableciendo las principales funcionalidades y restricciones, sirviendo de base a posteriores procesos del ciclo de vida.

Previamente a la redacción de los requisitos han sido identificados los casos de uso con el fin de lograr una mejor definición.



### 3.4.1. ESPECIFICACIÓN DE LOS CASOS DE USO

A continuación se presentan los casos de uso del sistema a desarrollar.



Ilustración 11: Casos de uso



### 3.4.1.1. ESPECIFICACIÓN DE LOS CASOS DE USO

En este apartado se realiza la especificación de los casos de uso. A continuación se presenta la información que completará cada caso de uso:

- **Nombre:** especificación textual del caso de uso.
- **Identificador:** estará formado por el literal “CU-XXX”, siendo XXX un identificador numérico único para cada caso de uso.
- **Actores:** tipo de usuario de la aplicación.
- **Objetivo:** finalidad del caso de uso.
- **Precondiciones:** estado que se debe cumplir para poder realizar una operación.
- **Postcondiciones:** estado en el que queda el sistema tras realizar una operación.
- **Escenario básico:** especifica cómo interactúa un actor con el sistema y cuál es la respuesta que el sistema le ofrece.
- **Escenarios alternativos:** condiciones excepcionales que afectan al escenario y respuestas del sistema ante esas situaciones.

Identificador	CU - 001
Nombre	Arrancar sistema.
Actores	Usuario.
Objetivo	Iniciar los componentes de la aplicación
Precondiciones	Ninguna
Poscondiciones	Aplicación arrancada
Escenario Básico	1. El usuario inicia la ejecución de la aplicación 2. Se muestra la pantalla para la configuración de generación de ficheros
Escenarios Alternativos	

**Tabla 3: Caso de Uso 001**



<b>Identificador</b>		<b>CU - 002</b>
Nombre	Configurar la generación de ficheros.	
Actores	Usuario.	
Objetivo	Configurar todos los parámetros necesarios para la generación de los ficheros de estudio (entrenamiento, test y CV)	
Precondiciones	Sistema inicializado	
Poscondiciones	Parámetros configurados	
Escenario Básico	<ol style="list-style-type: none"><li>1. El usuario introduce los parámetros que configuran la generación de los ficheros que se someterán a estudio.</li><li>2. El sistema va presentado opciones en función de la configuración que el usuario realice.</li></ol>	
Escenarios Alternativos		

**Tabla 4: Caso de Uso 002**

<b>Identificador</b>		<b>CU - 003</b>
Nombre	Configurar fichero origen.	
Actores	Usuario.	
Objetivo	Seleccionar el fichero origen e indicar las columnas de este que van a formar parte del estudio.	
Precondiciones	Sistema inicializado	
Poscondiciones	Fichero origen cargado	
Escenario Básico	<ol style="list-style-type: none"><li>1. El usuario selecciona el fichero origen.</li><li>2. El sistema muestra las columnas del fichero origen.</li><li>3. El usuario selecciona aquellas que van a ser objetivo del estudio</li></ol>	
Escenarios Alternativos	<ol style="list-style-type: none"><li>1. 2a El fichero de entrada es incorrecto.</li><li>1. El sistema muestra un mensaje de error indicándolo.</li></ol>	





**Tabla 5: Caso de Uso 003**

<b>Identificador</b>		<b>CU - 004</b>
Nombre	Configurar proceso iterativo.	
Actores	Usuario.	
Objetivo	Generar n baterías de ficheros de prueba, test y CV.	
Precondiciones	Sistema inicializado	
Poscondiciones	Parámetros configurados	
Escenario Básico	<ol style="list-style-type: none"><li>1. El usuario selecciona el proceso iterativo.</li><li>2. El sistema muestra los campos para la configuración de este proceso.</li></ol>	
Escenarios Alternativos		

**Tabla 6: Caso de Uso 004**

<b>Identificador</b>		<b>CU - 005</b>
Nombre	Generar ficheros.	
Actores	Usuario.	
Objetivo	Generar los ficheros de prueba, test y CV siguiendo la configuración definida por el usuario.	
Precondiciones	<ul style="list-style-type: none"><li>• Sistema inicializado</li><li>• Campos rellenos correctamente.</li></ul>	
Poscondiciones	Ficheros generados.	
Escenario Básico	<ol style="list-style-type: none"><li>1. El usuario ejecuta la generación de ficheros</li><li>2. El sistema genera los ficheros siguiendo la configuración indicada por el usuario.</li><li>3. El sistema muestra una barra de progreso del proceso.</li></ol>	
Escenarios Alternativos	<ol style="list-style-type: none"><li>1. 2a Alguno de los parámetros de entrada es incorrecto. La aplicación muestra un mensaje de error indicando el parámetro erróneo y el porqué del error.</li></ol>	



**Tabla 7: Caso de Uso 005**

<b>Identificador</b>		<b>CU - 006</b>
Nombre	Acceder a generación de informes.	
Actores	Usuario.	
Objetivo	Pasar a la sección de generación de informes.	
Precondiciones	<ul style="list-style-type: none"><li>• Sistema inicializado</li><li>• Si es un proceso iterativo, entonces los campos para la configuración de generación de ficheros deben de estar rellenos correctamente.</li></ul>	
Poscondiciones	Pantalla de generación de informes.	
Escenario Básico	<ol style="list-style-type: none"><li>1. El usuario pulsa sobre el enlace que permite pasar a la pantalla de generación de informes</li><li>2. El sistema muestra la pantalla.</li></ol>	
Escenarios Alternativos	<ol style="list-style-type: none"><li>1. 2a Si es un proceso iterativo y los campos para la configuración de generación de ficheros no están rellenos correctamente. La aplicación muestra un mensaje de error indicando el parámetro erróneo y el porqué del error.</li></ol>	

**Tabla 8: Caso de Uso 006**

<b>Identificador</b>		<b>CU - 007</b>
Nombre	Configurar RNA e informes	
Actores	Usuario.	
Objetivo	Configurar la RNA y los informes.	
Precondiciones	<ul style="list-style-type: none"><li>• Sistema inicializado</li><li>• Ficheros generados</li></ul>	
Poscondiciones	Parámetros configurados.	
Escenario Básico	<ol style="list-style-type: none"><li>1. El usuario rellena los campos necesarios para el entrenamiento y testeo de la red, así como las rutas donde dejar resultados e informes.</li></ol>	



<b>Identificador</b>		<b>CU – 007</b>
		2. Selecciona el tipo de estudio, clasificación o predicción
Escenarios Alternativos		

**Tabla 9: Caso de Uso 007**

<b>Identificador</b>		<b>CU – 008</b>
Nombre		Crear una nueva RNA.
Actores		Usuario.
Objetivo		Crear una nueva red de neuronas sobre la que realizar un estudio.
Precondiciones		<ul style="list-style-type: none"><li>• Sistema inicializado</li><li>• Ficheros generados</li></ul>
Poscondiciones		Nueva red creada.
Escenario Básico		<ol style="list-style-type: none"><li>1. El usuario desea crear una nueva red de neuronas. Pulsa sobre el enlace</li><li>2. El sistema le lleva directamente a a NeuroSolutions donde puede crear la red.</li></ol>
Escenarios Alternativos		<ol style="list-style-type: none"><li>1. 2a NeuroSolutions no está disponible. La aplicación informa al usuario de que no es posible generar una nueva RNA con NeuroSolutions.</li></ol>

**Tabla 10: Caso de Uso 008**

<b>Identificador</b>		<b>CU – 009</b>
Nombre		Ejecutar generación de informes
Actores		Usuario.
Objetivo		Realizar el entrenamiento y el testeo de la red, y generar informes a partir de los resultados obtenidos.
Precondiciones		<ul style="list-style-type: none"><li>• Sistema inicializado</li></ul>



<b>Identificador</b>		<b>CU – 009</b>
		<ul style="list-style-type: none"><li>• Ficheros generados</li><li>• Sistema configurado</li></ul>
Poscondiciones		Ficheros de resultados e informes.
Escenario Básico		<ol style="list-style-type: none"><li>1. El usuario desea ejecutar el proceso de generación de informes a partir de la configuración establecida para la red y los ficheros generados anteriormente. Pulsa sobre el enlace de ejecución</li><li>2. El sistema realiza el entrenamiento y testeo sobre la red, y genera los informes a partir de los resultados obtenidos.</li><li>3. Los resultados e informes se dejan en los paths indicados.</li></ol>
Escenarios Alternativos		<ol style="list-style-type: none"><li>1. 2a La configuración no es correcta. La aplicación muestra un mensaje de error indicando el parámetro erróneo y el porqué del error.</li></ol>

**Tabla 11: Caso de Uso 009**

<b>Identificador</b>		<b>CU – 010</b>
Nombre		Ejecutar entrenamiento y testeo de RNA
Actores		Usuario.
Objetivo		Realizar el entrenamiento y el testeo de la red con la configuración definida.
Precondiciones		<ul style="list-style-type: none"><li>• Sistema inicializado</li><li>• Ficheros generados</li><li>• Sistema configurado</li></ul>
Poscondiciones		Ficheros de resultados.
Escenario Básico		<ol style="list-style-type: none"><li>1. El usuario desea ejecutar el proceso de entrenamiento y testeo de la red con la configuración establecida y los ficheros generados anteriormente. Pulsa sobre el enlace de ejecución</li><li>2. El sistema realiza el entrenamiento y testeo sobre la red.</li></ol>



<b>Identificador</b>		<b>CU – 010</b>
		3. Los resultados se dejan en el path indicado.
Escenarios Alternativos		1. 2a La configuración no es correcta. La aplicación muestra un mensaje de error indicando el parámetro erróneo y el porqué del error.

**Tabla 12: Caso de Uso 010**

<b>Identificador</b>		<b>CU – 011</b>
Nombre		Generar informes de resultados
Actores		Usuario.
Objetivo		Generar informes de resultados del entrenamiento y testeo realizado a la red.
Precondiciones		<ul style="list-style-type: none"><li>• Sistema inicializado</li><li>• Ficheros generados</li><li>• Sistema configurado</li><li>• Resultados generados.</li></ul>
Poscondiciones		Ficheros de informe generado.
Escenario Básico		1. El usuario obtener un informe de resultados del proceso de entrenamiento y testeo de la red con la configuración establecida y los ficheros generados anteriormente. Pulsa sobre el enlace de ejecución 2. El sistema realiza el entrenamiento y testeo sobre la red. 3. El sistema va generando el informe de resultados a medida que se va desarrollando el proceso de testeo.
Escenarios Alternativos		2. 2a La configuración no es correcta. La aplicación muestra un mensaje de error indicando el parámetro erróneo y el porqué del error.

**Tabla 13: Caso de Uso 011**



<b>Identificador</b>		<b>CU – 012</b>
Nombre	Volver a generación de ficheros	
Actores	Usuario.	
Objetivo	Volver a la pantalla de generar ficheros manteniendo los datos que esta tuviese anteriormente.	
Precondiciones	<ul style="list-style-type: none"><li>• Sistema inicializado</li><li>• Pantalla de ejecutar generación de informes</li></ul>	
Poscondiciones	Mostrar la pantalla de generar ficheros con los datos que anteriormente tuviese.	
Escenario Básico	<ol style="list-style-type: none"><li>1. El usuario desea volver a la pantalla de generar ficheros por lo que pulsa sobre el enlace de retroceder.</li><li>2. El sistema salva los datos de la pantalla actual, y recupera los de la anterior.</li><li>3. El sistema presenta la pantalla generar ficheros con los datos que ha recuperado.</li></ol>	
Escenarios Alternativos		

**Tabla 14: Caso de Uso 012**

<b>Identificador</b>		<b>CU – 013</b>
Nombre	Acceder a generación del resumen.	
Actores	Usuario.	
Objetivo	Pasar a la sección de generación de resumen.	
Precondiciones	<ul style="list-style-type: none"><li>• Sistema inicializado</li></ul>	
Poscondiciones	Mostrar la pantalla de generación de resumen.	
Escenario Básico	<ol style="list-style-type: none"><li>1. El usuario pulsa sobre el enlace que permite pasar a la pantalla de generación de resumen</li><li>2. El sistema muestra la pantalla.</li></ol>	
Escenarios Alternativos		

**Tabla 15: Caso de Uso 013**



<b>Identificador</b>		<b>CU – 014</b>
Nombre	Configurar el resumen.	
Actores	Usuario.	
Objetivo	Configurar las variables del resumen a ejecutar.	
Precondiciones	<ul style="list-style-type: none"><li>• Sistema inicializado</li></ul>	
Poscondiciones	Ninguna	
Escenario Básico	<ol style="list-style-type: none"><li>1. El usuario rellena todos los campo de la pantalla de generación de resumen</li><li>2. Selecciona el tipo de resumen, clasificación o predicción</li></ol>	
Escenarios Alternativos		

**Tabla 16: Caso de Uso 014**

<b>Identificador</b>		<b>CU – 015</b>
Nombre	Ejecutar resumen	
Actores	Usuario.	
Objetivo	Realizar un resumen de todos los ficheros de informes.	
Precondiciones	<ul style="list-style-type: none"><li>• Sistema inicializado</li><li>• Ficheros de informes generados</li><li>• Sistema configurado</li></ul>	
Poscondiciones	Fichero de resumen generado.	
Escenario Básico	<ol style="list-style-type: none"><li>1. El usuario desea ejecutar el proceso de generación del resumen de los informes indicados en la configuración. Pulsa sobre el enlace de ejecución</li><li>2. El sistema genera un resumen en base al tipo de estudio y los informes seleccionados</li><li>3. El resumen se dejan en el path indicado.</li></ol>	
Escenarios Alternativos	<ol style="list-style-type: none"><li>3. 2a La configuración no es correcta. La aplicación muestra un mensaje de error indicando el</li></ol>	



<b>Identificador</b>	<b>CU – 015</b>
	parámetro erróneo y el porqué del error.

**Tabla 17: Caso de Uso 015**

<b>Identificador</b>	<b>CU – 016</b>
Nombre	Mostrar resumen
Actores	Usuario.
Objetivo	Mostrar el fichero de resumen.
Precondiciones	<ul style="list-style-type: none"><li>• Sistema inicializado</li><li>• Ejecutar resumen realizado</li></ul>
Poscondiciones	Mostrar el resumen mediante Excel.
Escenario Básico	<ol style="list-style-type: none"><li>1. El usuario desea ver el resumen de los informes. Pulsa sobre el icono de Excel que se muestra cuando el resumen ha sido generado.</li><li>2. El sistema presenta el resumen mediante Excel.</li></ol>
Escenarios Alternativos	<ol style="list-style-type: none"><li>4. 2a Cualquier problema derivado de VBA. La aplicación muestra un mensaje de error indicando el parámetro erróneo y el porqué del error.</li></ol>

**Tabla 18: Caso de Uso 016**

<b>Identificador</b>	<b>CU – 017</b>
Nombre	Cerrar sistema.
Actores	Usuario.
Objetivo	Cerrar la aplicación.
Precondiciones	Sistema inicializado
Poscondiciones	
Escenario Básico	<ol style="list-style-type: none"><li>1. El usuario selecciona el aspa de la barra de la aplicación.</li><li>2. La aplicación se cierra</li></ol>





**Identificador**

**CU – 017**

Escenarios Alternativos

**Tabla 19: Caso de Uso 017**

### 3.4.2. DEFINICIÓN DE LOS REQUISITOS DEL SISTEMA

En este apartado se realiza una extracción de requisitos del sistema con el fin de presentar las principales funcionalidades deseadas por el cliente, sirviendo de base a posteriores fases del ciclo de vida del proyecto. Los requisitos identificados proporcionarán al cliente una visión general de la aplicación, de forma completa y sin ambigüedades.

La toma de requisitos tiene un valor añadido, ya que ayuda a orientar al cliente y deshacer ambigüedades que hubieran podido presentarse. Con los requisitos se presentará una visión general de la aplicación.

#### 3.4.2.1. IDENTIFICACIÓN DE REQUISITOS

En esta tarea se realiza la obtención detallada de requisitos de usuario mediante sesiones de trabajo con el cliente. La lista de requisitos obtenida no es definitiva y podrá ser modificada durante el proyecto, añadiendo requisitos no contemplados, o bien eliminando o modificando requisitos incorrectos, de forma que se terminen cubriendo todas las funcionalidades expuestas por el cliente.

La definición de requisitos recoge lo que quiere el cliente y lo que necesita, englobando los requisitos obtenidos del usuario en dos grandes categorías:

- **Requisitos de capacidad:** representan lo que necesitan los usuarios para resolver un problema o lograr un objetivo.



- **Requisitos de restricción:** son las restricciones impuestas por los usuarios sobre cómo se debe resolver el problema o cómo se debe alcanzar el objetivo.

Si se considera necesario se podrán englobar los requisitos en nuevas categorías. Cada requisito de usuario debe ser definido con una serie de atributos, que le añadan toda la información necesaria para su seguimiento posterior y su clasificación. A continuación se detallan:

- **Identificador:** Cada requisito de usuario debe estar identificado de forma única. Este identificador tendrá el siguiente formato: RU-C-nnn o RU-R-nnn, donde:
  - RU: Indica que se trata de un requisito de usuario.
  - C: Indica que se trata de un requisito de usuario de capacidad.
  - R: Indica que se trata de un requisito de usuario de restricción.
  - nnn: Tomará valores numéricos dentro del rango 000-999.
- **Prioridad:** Se asignará una prioridad a cada requisito con el fin de poder realizar una planificación correcta durante fases posteriores. Su clasificación puede tomar los valores: alta, media y baja.
- **Necesidad:** Los requisitos clasificados como esenciales para el usuario no pueden ser eliminados, mientras que los demás requisitos estarán siempre sujetos a modificación en el caso de que exista una causa que lo justifique. Por lo tanto, se clasificarán descendientemente de acuerdo a su necesidad de la siguiente forma: esencial, deseable y opcional.
- **Claridad:** Identifica la falta o existencia de ambigüedad de un requisito, esto es, si puede ser interpretado de varias formas dependiendo del contexto. Los requisitos serán clasificados de acuerdo a su claridad de forma descendente de la siguiente forma: alta, media y baja.
- **Fuente:** Identifica el origen del requisito, que puede estar en el usuario, una fuente externa como un documento, o el propio equipo de desarrollo del proyecto.



- **Estabilidad:** Algunos requisitos pueden no estar sujetos a cambios durante el proyecto debido a su naturaleza, mientras que otros pueden estar sujetos a determinados cambios por el desarrollo de la etapa de diseño o los requisitos software. Los requisitos poco estables deberán ser observados con mayor cuidado durante el desarrollo del proyecto al ser susceptibles a cambios. Cada requisito se clasificará en: estable o inestable.
- **Verificabilidad:** Indica si el cumplimiento de un requisito en el sistema puede ser susceptible de comprobación, esto es, si se puede verificar que el requisito se ha incorporado en el diseño y que en el sistema se puede verificar su cumplimiento. La verificabilidad de cada requisito se clasificará en: alta, media y baja.

### 3.4.2.2. REQUISITOS DE CAPACIDAD

A continuación se presentan los requisitos de capacidad, los cuales representan lo que necesita el usuario para resolver un problema o lograr un objetivo.

Identificador		RU- C- 001	
Prioridad	<input checked="" type="radio"/> Alta <input type="radio"/> Media <input type="radio"/> Baja.	Fuentes	Cliente
Necesidad	<input checked="" type="radio"/> Esencial <input type="radio"/> Deseable <input type="radio"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="radio"/> Alta <input type="radio"/> Media <input type="radio"/> Baja.	Verificabilidad	<input checked="" type="radio"/> Alta <input type="radio"/> Media <input type="radio"/> Baja.
Descripción	Al iniciar el sistema la aplicación mostrará la pantalla de generación de ficheros.		

Tabla 20: Requisito de usuario C001

Identificador		RU- C- 002	
Prioridad	<input checked="" type="radio"/> Alta <input type="radio"/> Media	Fuentes	Cliente
Departamento de Informática		Página 74 de 207	



<b>Identificador</b>		<b>RU- C- 002</b>	
	<input type="radio"/> Baja.		
Necesidad	<input checked="" type="radio"/> Esencial <input type="radio"/> Deseable <input type="radio"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="radio"/> Alta <input type="radio"/> Media <input type="radio"/> Baja.	Verificabilidad	<input checked="" type="radio"/> Alta <input type="radio"/> Media <input type="radio"/> Baja.
Descripción	Configurar todos los paths tanto de los ficheros de entrada como de salida en la generación de ficheros.		

**Tabla 21: Requisito de usuario C002**

<b>Identificador</b>		<b>RU- C- 003</b>	
Prioridad	<input checked="" type="radio"/> Alta <input type="radio"/> Media <input type="radio"/> Baja.	Fuentes	Equipo de desarrollo
Necesidad	<input type="radio"/> Esencial <input checked="" type="radio"/> Deseable <input type="radio"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="radio"/> Alta <input type="radio"/> Media <input type="radio"/> Baja.	Verificabilidad	<input checked="" type="radio"/> Alta <input type="radio"/> Media <input type="radio"/> Baja.
Descripción	Indicar que columnas del fichero de entrada van a pasar a formar parte de los ficheros generados para el estudio.		

**Tabla 22: Requisito de usuario C003**

<b>Identificador</b>		<b>RU- C- 004</b>	
Prioridad	<input checked="" type="radio"/> Alta <input type="radio"/> Media <input type="radio"/> Baja.	Fuentes	Cliente
Necesidad	<input checked="" type="radio"/> Esencial <input type="radio"/> Deseable <input type="radio"/> Opcional	Estabilidad	Estable



<b>Identificador</b>		<b>RU- C- 004</b>	
Claridad	<input checked="" type="radio"/> Alta <input type="radio"/> Media <input type="radio"/> Baja.	Verificabilidad	<input checked="" type="radio"/> Alta <input type="radio"/> Media <input type="radio"/> Baja.
Descripción	Indicar el número de registros destinados a entrenamiento, testeo y validación cruzada que deben contener los ficheros de una pruebas, así como el porcentaje de perforaciones que debe de cumplirse en todos y cada uno de ellos.		

**Tabla 23: Requisito de usuario C004**

<b>Identificador</b>		<b>RU- C- 005</b>	
Prioridad	<input checked="" type="radio"/> Alta <input type="radio"/> Media <input type="radio"/> Baja.	Fuentes	Cliente
Necesidad	<input checked="" type="radio"/> Esencial <input type="radio"/> Deseable <input type="radio"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="radio"/> Alta <input type="radio"/> Media <input type="radio"/> Baja.	Verificabilidad	<input checked="" type="radio"/> Alta <input type="radio"/> Media <input type="radio"/> Baja.
Descripción	Indicar el número de iteraciones que se van a realizar en la generación de ficheros..		

**Tabla 24: Requisito de usuario C005**

<b>Identificador</b>		<b>RU- C- 006</b>	
Prioridad	<input checked="" type="radio"/> Alta <input type="radio"/> Media <input type="radio"/> Baja.	Fuentes	Cliente
Necesidad	<input checked="" type="radio"/> Esencial <input type="radio"/> Deseable <input type="radio"/> Opcional	Estabilidad	Estable



<b>Identificador</b>		<b>RU- C- 006</b>	
Claridad	<input checked="" type="radio"/> Alta <input type="radio"/> Media <input type="radio"/> Baja.	Verificabilidad	<input checked="" type="radio"/> Alta <input type="radio"/> Media <input type="radio"/> Baja.
Descripción	Generar n baterías de ficheros para un estudio en una misma ejecución, incluyendo variaciones en el número de registros destinados a entrenamiento, testeo y CV.		

**Tabla 25: Requisito de usuario C006**

<b>Identificador</b>		<b>RU- C- 007</b>	
Prioridad	<input checked="" type="radio"/> Alta <input type="radio"/> Media <input type="radio"/> Baja.	Fuentes	Cliente
Necesidad	<input checked="" type="radio"/> Esencial <input type="radio"/> Deseable <input type="radio"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="radio"/> Alta <input type="radio"/> Media <input type="radio"/> Baja.	Verificabilidad	<input checked="" type="radio"/> Alta <input type="radio"/> Media <input type="radio"/> Baja.
Descripción	Configurar todos los paths tanto de los ficheros de entrada como de salida en la generación de informes.		

**Tabla 26: Requisito de usuario C007**

<b>Identificador</b>		<b>RU- C- 008</b>	
Prioridad	<input checked="" type="radio"/> Alta <input type="radio"/> Media <input type="radio"/> Baja.	Fuentes	Cliente
Necesidad	<input checked="" type="radio"/> Esencial <input type="radio"/> Deseable <input type="radio"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="radio"/> Alta <input type="radio"/> Media <input type="radio"/> Baja.	Verificabilidad	<input checked="" type="radio"/> Alta <input type="radio"/> Media <input type="radio"/> Baja.
Descripción	Realizar procesos de entrenamiento y testeo tanto de clasificación como de predicción.		



**Tabla 27: Requisito de usuario C008**

<b>Identificador</b>		<b>RU- C- 009</b>	
Prioridad	<input checked="" type="radio"/> Alta <input type="radio"/> Media <input type="radio"/> Baja.	Fuentes	Cliente
Necesidad	<input checked="" type="radio"/> Esencial <input type="radio"/> Deseable <input type="radio"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="radio"/> Alta <input type="radio"/> Media <input type="radio"/> Baja.	Verificabilidad	<input checked="" type="radio"/> Alta <input type="radio"/> Media <input type="radio"/> Baja.
Descripción	Generar informes con los resultados del entrenamiento y testeo de la red.		

**Tabla 28: Requisito de usuario C009**

<b>Identificador</b>		<b>RU- C- 010</b>	
Prioridad	<input checked="" type="radio"/> Alta <input type="radio"/> Media <input type="radio"/> Baja.	Fuentes	Cliente
Necesidad	<input checked="" type="radio"/> Esencial <input type="radio"/> Deseable <input type="radio"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="radio"/> Alta <input type="radio"/> Media <input type="radio"/> Baja.	Verificabilidad	<input checked="" type="radio"/> Alta <input type="radio"/> Media <input type="radio"/> Baja.
Descripción	Incluir en los informes los resultados para el estudio de “fortaleza frente a valores atípicos.		

**Tabla 29: Requisito de usuario C010**

<b>Identificador</b>		<b>RU- C- 011</b>	
Prioridad	<input checked="" type="radio"/> Alta <input type="radio"/> Media <input type="radio"/> Baja.	Fuentes	Equipo de desarrollo



<b>Identificador</b>		<b>RU- C- 011</b>	
Necesidad	<input type="radio"/> Esencial <input type="radio"/> Deseable <input checked="" type="radio"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="radio"/> Alta <input type="radio"/> Media <input type="radio"/> Baja.	Verificabilidad	<input checked="" type="radio"/> Alta <input type="radio"/> Media <input type="radio"/> Baja.
Descripción	Incluir en los informes los tiempos consumidos durante el entrenamiento y testeo de la red		

**Tabla 30: Requisito de usuario C011**

<b>Identificador</b>		<b>RU- C- 012</b>	
Prioridad	<input checked="" type="radio"/> Alta <input type="radio"/> Media <input type="radio"/> Baja.	Fuentes	Cliente
Necesidad	<input checked="" type="radio"/> Esencial <input type="radio"/> Deseable <input type="radio"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="radio"/> Alta <input type="radio"/> Media <input type="radio"/> Baja.	Verificabilidad	<input checked="" type="radio"/> Alta <input type="radio"/> Media <input type="radio"/> Baja.
Descripción	Configurar los criterios de parada de testeo desde la aplicación. Estos serán Epocs, MSE y CV		

**Tabla 31: Requisito de usuario C012**

<b>Identificador</b>		<b>RU- C- 013</b>	
Prioridad	<input checked="" type="radio"/> Alta <input type="radio"/> Media <input type="radio"/> Baja.	Fuentes	Cliente
Necesidad	<input checked="" type="radio"/> Esencial <input type="radio"/> Deseable <input type="radio"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="radio"/> Alta	Verificabilidad	<input checked="" type="radio"/> Alta





<b>Identificador</b>		<b>RU- C- 013</b>	
	<input type="radio"/> Media <input type="radio"/> Baja.		<input type="radio"/> Media <input type="radio"/> Baja.
Descripción	En procesos iterativos el entrenamiento y testeo de la red de todas las baterías de pruebas creadas en la generación de ficheros deben llevarse a cabo si intervención del usuario.		

**Tabla 32: Requisito de usuario C013**

<b>Identificador</b>		<b>RU- C- 014</b>	
Prioridad	<input checked="" type="radio"/> Alta <input type="radio"/> Media <input type="radio"/> Baja.	Fuentes	Cliente
Necesidad	<input checked="" type="radio"/> Esencial <input type="radio"/> Deseable <input type="radio"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="radio"/> Alta <input type="radio"/> Media <input type="radio"/> Baja.	Verificabilidad	<input checked="" type="radio"/> Alta <input type="radio"/> Media <input type="radio"/> Baja.
Descripción	Configurar todos los paths tanto de los ficheros de entrada como de salida en la generación del resumen.		

**Tabla 33: Requisito de usuario C014**

<b>Identificador</b>		<b>RU- C- 015</b>	
Prioridad	<input checked="" type="radio"/> Alta <input type="radio"/> Media <input type="radio"/> Baja.	Fuentes	Cliente
Necesidad	<input checked="" type="radio"/> Esencial <input type="radio"/> Deseable <input type="radio"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="radio"/> Alta <input type="radio"/> Media <input type="radio"/> Baja.	Verificabilidad	<input checked="" type="radio"/> Alta <input type="radio"/> Media <input type="radio"/> Baja.
Descripción	Generar el fichero de resumen en formato Excel.		

**Tabla 34: Requisito de usuario C015**



<b>Identificador</b>		<b>RU- C- 016</b>	
Prioridad	<input checked="" type="radio"/> Alta <input type="radio"/> Media <input type="radio"/> Baja.	Fuentes	Cliente
Necesidad	<input checked="" type="radio"/> Esencial <input type="radio"/> Deseable <input type="radio"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="radio"/> Alta <input type="radio"/> Media <input type="radio"/> Baja.	Verificabilidad	<input checked="" type="radio"/> Alta <input type="radio"/> Media <input type="radio"/> Baja.
Descripción	Generar resúmenes diferentes para procesos de predicción y clasificación.		

**Tabla 35: Requisito de usuario C016**

<b>Identificador</b>		<b>RU- C- 017</b>	
Prioridad	<input type="radio"/> Alta <input checked="" type="radio"/> Media <input type="radio"/> Baja.	Fuentes	Cliente
Necesidad	<input type="radio"/> Esencial <input checked="" type="radio"/> Deseable <input type="radio"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="radio"/> Alta <input type="radio"/> Media <input type="radio"/> Baja.	Verificabilidad	<input checked="" type="radio"/> Alta <input type="radio"/> Media <input type="radio"/> Baja.
Descripción	Acceder al resumen desde la aplicación.		

**Tabla 36: Requisito de usuario C017**

<b>Identificador</b>		<b>RU- C- 018</b>	
Prioridad	<input checked="" type="radio"/> Alta <input type="radio"/> Media <input type="radio"/> Baja.	Fuentes	Cliente
Necesidad	<input checked="" type="radio"/> Esencial <input type="radio"/> Deseable	Estabilidad	Estable



<b>Identificador</b>		<b>RU- C- 018</b>	
	<input type="radio"/> Opcional		
Claridad	<input checked="" type="radio"/> Alta <input type="radio"/> Media <input type="radio"/> Baja.	Verificabilidad	<input checked="" type="radio"/> Alta <input type="radio"/> Media <input type="radio"/> Baja.
Descripción	Cerrar la aplicación en cualquier momento desde el aspa.		

**Tabla 37: Requisito de usuario C018**

### 3.4.2.3. REQUISITOS DE RESTRICCIÓN

Son las restricciones impuestas acerca de cómo se debe resolver el problema o como se debe alcanzar el objetivo.

<b>Identificador</b>		<b>RU- R- 001</b>	
Prioridad	<input checked="" type="radio"/> Alta <input type="radio"/> Media <input type="radio"/> Baja.	Fuentes	Cliente
Necesidad	<input checked="" type="radio"/> Esencial <input type="radio"/> Deseable <input type="radio"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="radio"/> Alta <input type="radio"/> Media <input type="radio"/> Baja.	Verificabilidad	<input checked="" type="radio"/> Alta <input type="radio"/> Media <input type="radio"/> Baja.
Descripción	El texto de la aplicación debe ser en castellano.		

**Tabla 38: Requisito de usuario R001**

<b>Identificador</b>		<b>RU- R- 002</b>	
Prioridad	<input checked="" type="radio"/> Alta <input type="radio"/> Media <input type="radio"/> Baja.	Fuentes	Cliente



Identificador		RU- R- 002	
Necesidad	<input checked="" type="radio"/> Esencial <input type="radio"/> Deseable <input type="radio"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="radio"/> Alta <input type="radio"/> Media <input type="radio"/> Baja.	Verificabilidad	<input checked="" type="radio"/> Alta <input type="radio"/> Media <input type="radio"/> Baja.
Descripción	Recibir ficheros de entrada con cualquier extensión que sea reconocida por Excel.		

Tabla 39: Requisito de usuario R002

Identificador		RU- R- 003	
Prioridad	<input checked="" type="radio"/> Alta <input type="radio"/> Media <input type="radio"/> Baja.	Fuentes	Cliente
Necesidad	<input checked="" type="radio"/> Esencial <input type="radio"/> Deseable <input type="radio"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="radio"/> Alta <input type="radio"/> Media <input type="radio"/> Baja.	Verificabilidad	<input checked="" type="radio"/> Alta <input type="radio"/> Media <input type="radio"/> Baja.
Descripción	Todos los ficheros de salida que proporciona la aplicación deben de ser formato Excel.		

Tabla 40: Requisito de usuario R003

Identificador		RU- R- 004	
Prioridad	<input checked="" type="radio"/> Alta <input type="radio"/> Media <input type="radio"/> Baja.	Fuentes	Cliente
Necesidad	<input checked="" type="radio"/> Esencial <input type="radio"/> Deseable <input type="radio"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="radio"/> Alta <input type="radio"/> Media <input type="radio"/> Baja.	Verificabilidad	<input type="radio"/> Alta <input checked="" type="radio"/> Media <input type="radio"/> Baja.



<b>Identificador</b>		<b>RU- R- 004</b>
Descripción	Todo el sistema debe estar generado con un único lenguaje de programación.	

**Tabla 41: Requisito de usuario R004**

<b>Identificador</b>		<b>RU- R- 005</b>	
Prioridad	<ul style="list-style-type: none"> <li>● Alta</li> <li>○ Media</li> <li>○ Baja.</li> </ul>	Fuentes	Cliente
Necesidad	<ul style="list-style-type: none"> <li>● Esencial</li> <li>○ Deseable</li> <li>○ Opcional</li> </ul>	Estabilidad	Estable
Claridad	<ul style="list-style-type: none"> <li>● Alta</li> <li>○ Media</li> <li>○ Baja.</li> </ul>	Verificabilidad	<ul style="list-style-type: none"> <li>● Alta</li> <li>○ Media</li> <li>○ Baja.</li> </ul>
Descripción	El sistema debe integrarse de forma transparente para el usuario con NeuroSolutions.		

**Tabla 42: Requisito de usuario R005**

<b>Identificador</b>		<b>RU- R- 006</b>	
Prioridad	<ul style="list-style-type: none"> <li>● Alta</li> <li>○ Media</li> <li>○ Baja.</li> </ul>	Fuentes	Cliente
Necesidad	<ul style="list-style-type: none"> <li>● Esencial</li> <li>○ Deseable</li> <li>○ Opcional</li> </ul>	Estabilidad	Estable
Claridad	<ul style="list-style-type: none"> <li>● Alta</li> <li>○ Media</li> <li>○ Baja.</li> </ul>	Verificabilidad	<ul style="list-style-type: none"> <li>● Alta</li> <li>○ Media</li> <li>○ Baja.</li> </ul>
Descripción	El sistema será una aplicación será de escritorio. Se proporcionará un ejecutable para su instalación en entorno Windows.		

**Tabla 43: Requisito de usuario R006**



<b>Identificador</b>		<b>RU- R- 007</b>	
Prioridad	<input checked="" type="radio"/> Alta <input type="radio"/> Media <input type="radio"/> Baja.	Fuentes	Cliente
Necesidad	<input checked="" type="radio"/> Esencial <input type="radio"/> Deseable <input type="radio"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="radio"/> Alta <input type="radio"/> Media <input type="radio"/> Baja.	Verificabilidad	<input checked="" type="radio"/> Alta <input type="radio"/> Media <input type="radio"/> Baja.
Descripción	Los formatos de los ficheros de salida y entrada vienen definidos por los usuarios		

**Tabla 44: Requisito de usuario R007**

<b>Identificador</b>		<b>RU- R- 008</b>	
Prioridad	<input checked="" type="radio"/> Alta <input type="radio"/> Media <input type="radio"/> Baja.	Fuentes	Cliente
Necesidad	<input checked="" type="radio"/> Esencial <input type="radio"/> Deseable <input type="radio"/> Opcional	Estabilidad	Estable
Claridad	<input checked="" type="radio"/> Alta <input type="radio"/> Media <input type="radio"/> Baja.	Verificabilidad	<input checked="" type="radio"/> Alta <input type="radio"/> Media <input type="radio"/> Baja.
Descripción	La interfaz debe de ser sencilla e intuitiva, accesible para usuarios que no tenga un perfil técnico avanzado.		

**Tabla 45: Requisito de usuario R008**

<b>Identificador</b>		<b>RU- R- 009</b>	
Prioridad	<input checked="" type="radio"/> Alta <input type="radio"/> Media <input type="radio"/> Baja.	Fuentes	Cliente
Necesidad	<input checked="" type="radio"/> Esencial <input type="radio"/> Deseable <input type="radio"/> Opcional	Estabilidad	Estable



Identificador		RU- R- 009	
Claridad	<input checked="" type="radio"/> Alta <input type="radio"/> Media <input type="radio"/> Baja.	Verificabilidad	<input checked="" type="radio"/> Alta <input type="radio"/> Media <input type="radio"/> Baja.
Descripción	Mostrar el avance del proceso de generación de ficheros, informes y resumen con una barra de progreso.		

**Tabla 46: Requisito de usuario R009**

### 3.4.3. ESTUDIO DE ALTERNATIVAS DE SOLUCIÓN

Con el desarrollo de este proyecto se pretende desarrollar una herramienta que ayude a usuario a comprender mejor el comportamiento de las redes, y a ver cómo responden cuando se varían las distintas variables que intervienen en su entrenamiento. La solución ha sido impuesta por el cliente y no aplica realizar un estudio de diferentes alternativas.

Para poder comunicar el sistema con NeuroSolutions se ha utilizado la librería que proporciona la herramienta, dicha librería se integra como un componente de Visual Basic, por lo que este ha sido el lenguaje seleccionado para realizar el proyecto. Para la comunicación con Excel se ha utilizar Visual Basic for Applications, igualmente de natural integración con el lenguaje base.

### 3.4.4. VALORACIÓN DE LAS ALTERNATIVAS

No aplica porque la solución viene dada.

### 3.4.5. SELECCIÓN DE LA SOLUCIÓN

No aplica porque la solución viene dada.



## Capítulo 4. GESTIÓN DEL PROYECTO

### 4.1. Ciclo de vida

Se entiende por ciclo de vida software el conjunto de fases, procesos y actividades requeridas para ofertar, desarrollar, probar, integrar, explotar y mantener un producto software indicando como funciones principales la de por un lado determinar el orden de las fases y procesos y por otro el de establecer los criterios de transición para pasar de una fase a otra.

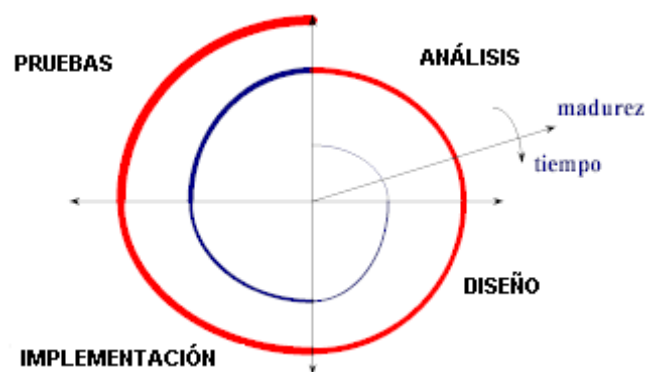


Ilustración 12: Ciclo de vida

Para el presente proyecto el ciclo de vida elegido ha sido el modelo “ciclo de vida prototipado” (también llamado ciclo de vida iterativo por prototipos), que es el que más se adapta a las necesidades de desarrollo en este caso, ya que el propio producto a lo largo de su desarrollo puede considerarse una sucesión de prototipos que progresan hasta llegar al estado deseado. En cada ciclo (prototipo), las especificaciones se van resolviendo paulatinamente, completando las del ciclo anterior y ganando “madurez” hasta que se alcanza el punto en el cuál se completa y puede dejarse de entrar en un nuevo ciclo.

En cada ciclo se completan varias fases, que se explican brevemente a continuación:

- **Análisis:** En esta fase se determinan (o amplían) los requisitos software del sistema, especificando a un alto nivel la arquitectura de la solución que se propone para dichos requisitos.





- **Diseño:** En esta segunda fase, se diseñan tanto los interfaces de usuario de la aplicación, como la arquitectura a un nivel más bajo de especificación, detallando los procesos del sistema.
- **Implementación:** En esta etapa se codifican los interfaces de usuario, se codifican los procesos, y se documenta el manual de usuario.
- **Pruebas:** En esta última fase, se definen las pruebas a realizar por el prototipo en cuestión y se llevan a cabo.

Este ciclo ha sido considerado como el más adecuado debido a diferentes motivos:

- Permite evaluar distintas posibilidades de desarrollo, ya que en los prototipos se puede fácilmente añadir o eliminar funcionalidades.
- Permite al tutor o usuario final realizar un seguimiento periódico del proyecto para verificar si el alumno ha comprendido bien los requisitos y si está desarrollando la aplicación de acuerdo con éstos.
- El prototipo es un documento vivo de buen funcionamiento del producto final.
- Ayuda para determinar requerimientos expresados en el prototipo.
- Experimenta sobre los aspectos del sistema que representan mayor complejidad.
- El usuario reacciona mucho mejor ante el prototipo, sobre el que puede experimentar, que no sobre una especificación escrita.
- Se incrementa la calidad del producto final, ya que el prototipo permite trabajar.

Se debe realizar un buen estudio para determinar que este ciclo de vida es el necesario ya que si se comete algún error, se producirán los siguientes inconvenientes:

- Fuerte inversión en un producto que es desechable, los prototipos se descartan.
- Tendencia a tratar de convertir el prototipo mismo en el sistema de producción.
- Aumento del coste.
- Se arrastran decisiones del diseño de prototipos al producto final.



## 4.2. Organización del proyecto

A la hora de llevar a cabo cualquier proyecto, se hace necesaria la tarea de realizar una planificación, detallando el tiempo estimado necesario para cumplir las distintas actividades y tareas que forman parte del proyecto, así como los recursos asignados a cada una de ellas. Por ello, se ha desarrollado una planificación de trabajo en función del ciclo de vida seleccionado anteriormente, ciclo de vida por prototipos.

Se ha decidido realizar tres iteraciones completas, tras cada una de las cuales el desarrollador se reúne con el tutor para corregir o matizar distintos aspectos de la aplicación.

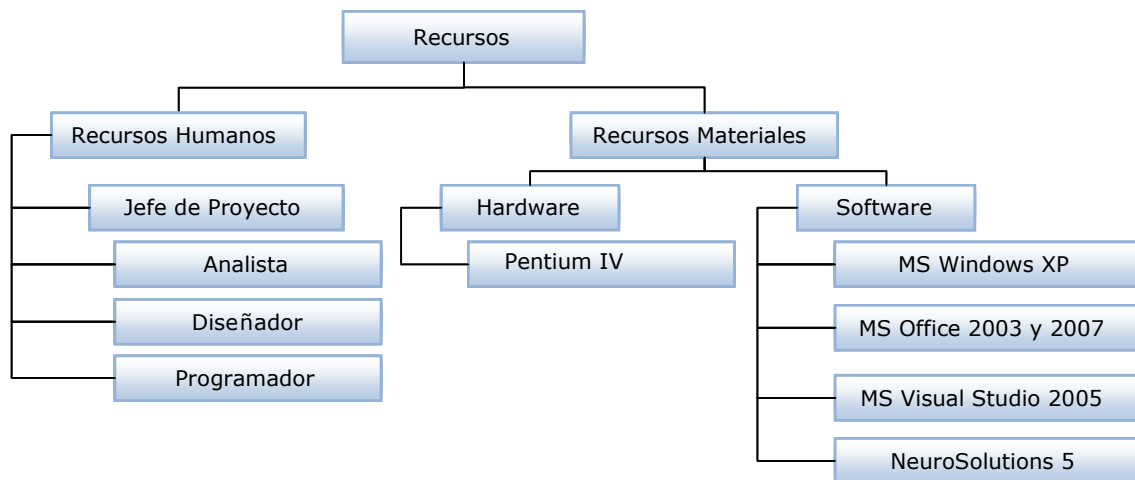
En lo que se refiere a los recursos humanos que intervienen en el desarrollo del proyecto, a continuación se muestran las personas y roles que participan en el proyecto. Las funciones de las que se encarga cada uno de ellos son:

- **Jefe de Proyecto:** Se encarga de la gestión del proyecto, su organización, planificación y supervisión a lo largo de todo el desarrollo del mismo.
- **Analista:** Se encarga de obtener y redactar los requisitos de usuario, además de modelar los procesos y tareas a codificar.
- **Diseñador:** Su tarea es el diseño de las interfaces, la arquitectura del sistema y el plan de verificación y validación.
- **Programador:** Se encarga de la codificación del sistema, así como de llevar a cabo las pruebas necesarias sobre el mismo.

El RBS (Resource Breakdown Structure) es una representación jerárquica de los recursos, tanto humanos como materiales, necesarios para la planificación de un proyecto. Esta técnica de organización de proyectos software tiene por objeto representar la organización humana del proyecto, así como la estructura de recursos tecnológicos y materiales.



A continuación se muestra el RBS del proyecto:

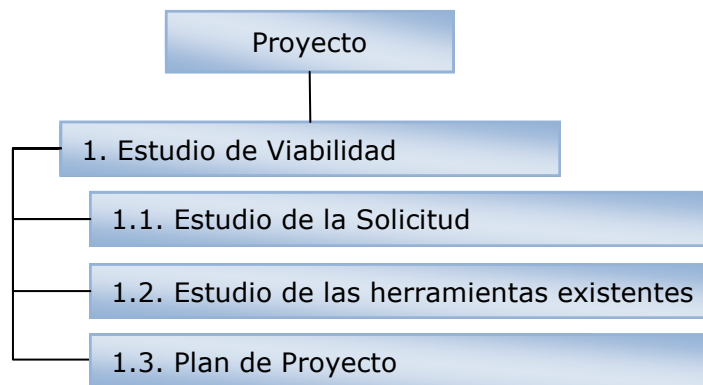


**Ilustración 13: RBS del proyecto**

El WBS (World Breakdown Structure) es una herramienta simple y práctica para definir el alcance de un proyecto. Es la base para poder saber qué se requiere hacer, para después poder realizar la planificación del proyecto. Se trata de identificar y definir las tareas que se van a realizar en el proyecto.

En el WBS se puede observar la descomposición de los procesos en subprocesos, y éstos a su vez en tareas, agrupadas por funcionalidad. La base para en la que está basada el WBS es el modelo de proceso software IEEE 1074. A continuación se muestra la división de tareas del proyecto mediante el WBS:

El primer proceso que se debe llevar a cabo es el estudio de viabilidad del proyecto, mediante el cual se analiza la solicitud del cliente, la situación actual y si el desarrollo del proyecto es viable. Toda la información se recoge en el documento de Estudio de Viabilidad.



**Ilustración 14: WBS Proceso Estudio de viabilidad**

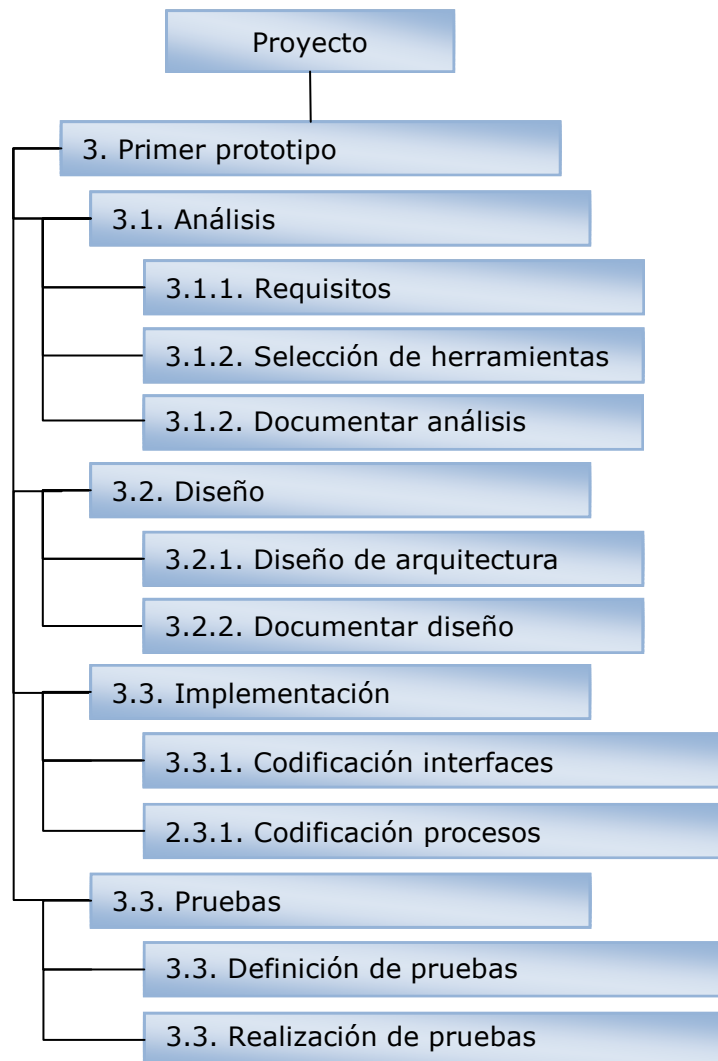
Si una vez finalizado el estudio de la viabilidad se concluye que el proyecto es factible el próximo paso a seguir es realizar el estudio de la gestión del proyecto. En este se seleccionará el modelo de ciclo de vida, se realizará la planificación a seguir durante el desarrollo del proyecto y estimación de costes del mismo.



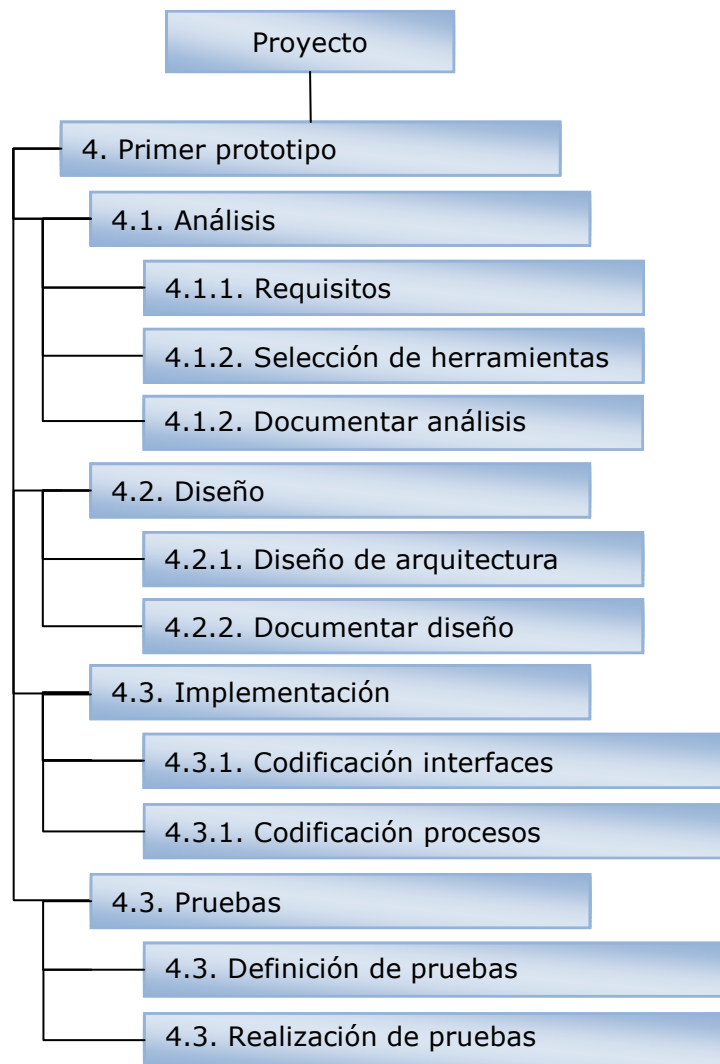
**Ilustración 15: WBS Gestión del proyecto**



Lo siguiente es presentar el WBS de cada una de las iteraciones realizadas en el proyecto de desarrollo, Por iteración se deben desarrollar todas las tareas de análisis, diseño, implementación y pruebas siguiendo el orden establecido. Esto no implica que su desarrollo podrá solaparse, fin e inicio de tareas, al contrario, esto confiere una mayor flexibilidad en la ejecución del proyecto, mejorando la calidad del software y los tiempos de desarrollo del mismo.

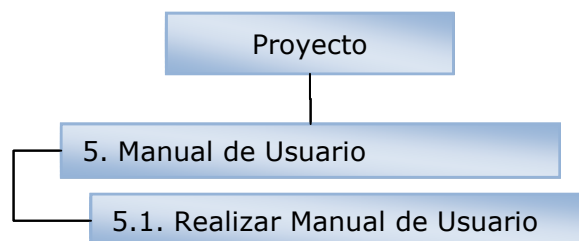


**Ilustración 16: WBS Primer prototipo**



**Ilustración 17: WBS Segundo prototipo**

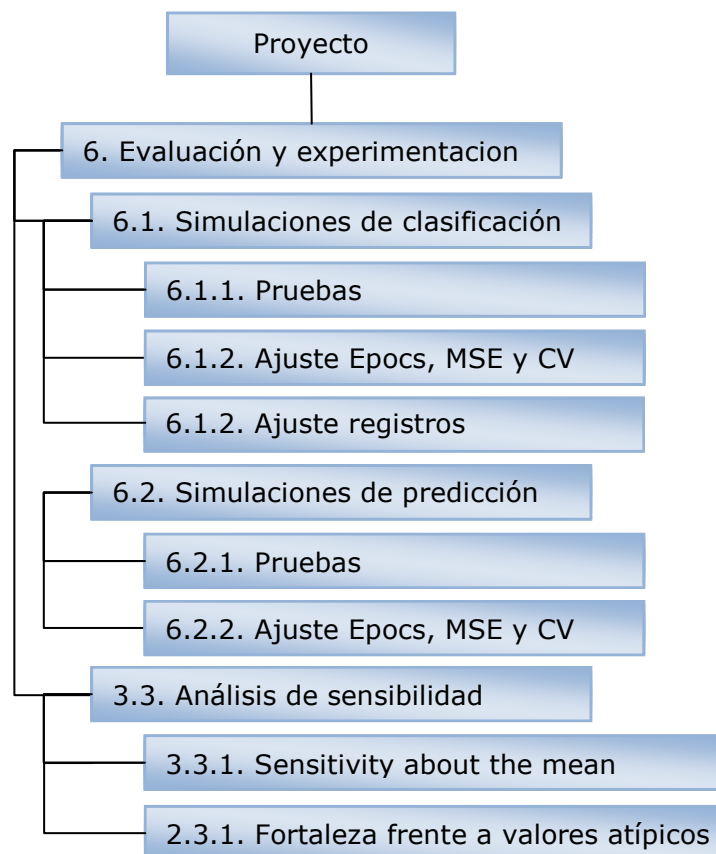
Una vez desarrollado el último prototipo se realiza el manual de usuario



**Ilustración 18: WBS de la documentación**



Con la herramienta desarrolla comenzará la fase más importante del proyecto, la fase de evaluación y experimentación. Esta fase será donde se llevan a cabo todas las tareas propias al proyecto, es decir, la búsqueda de la mejor configuración de una RNA perceptron multicapa para los datos de entrada suministrado para el proyecto. También se evaluarán 2 métodos para el análisis de sensibilidad de las variables de entrada.

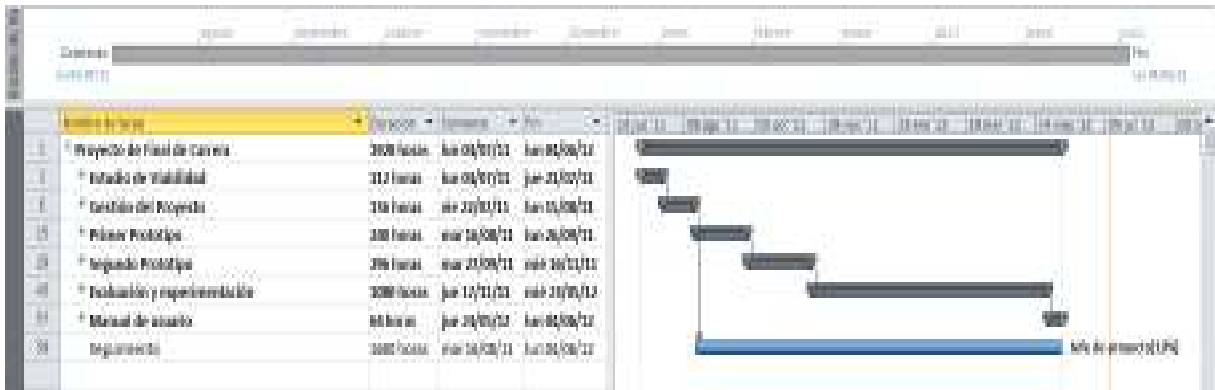


**Ilustración 19: WBS Evaluación y experimentación**



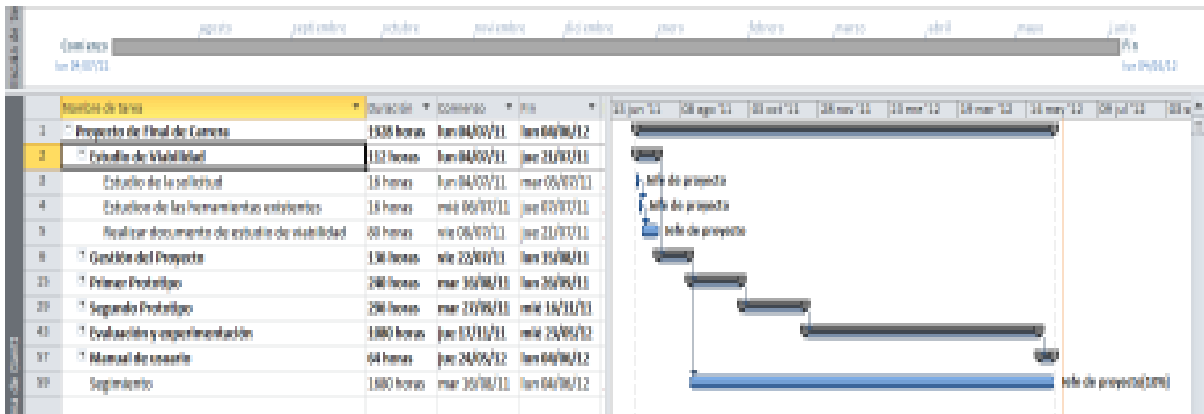
### 4.3. Planificación

La duración total de este proyecto es de 1.928 horas, lo que equivale a 241 días laborables, desde el 4 de julio de 2011, fecha en la que se inicia el proyecto, hasta el 4 de junio de 2012, fecha en la que el proyecto se da por finalizado y es entregado al cliente.



**Ilustración 20: Diagrama de Gantt principal**

El primer paso se debe acometer a la hora de realizar un proyecto es realizar el estudio de la viabilidad del mismo. Para ello, el jefe de proyecto realiza un estudio de la solicitud, de la situación actual y elabora un documento con dicha información, su principal finalidad es indicar si la solicitud realizada se puede llevar a cabo o no.



**Ilustración 21: Diagrama de Gantt del estudio de la viabilidad**



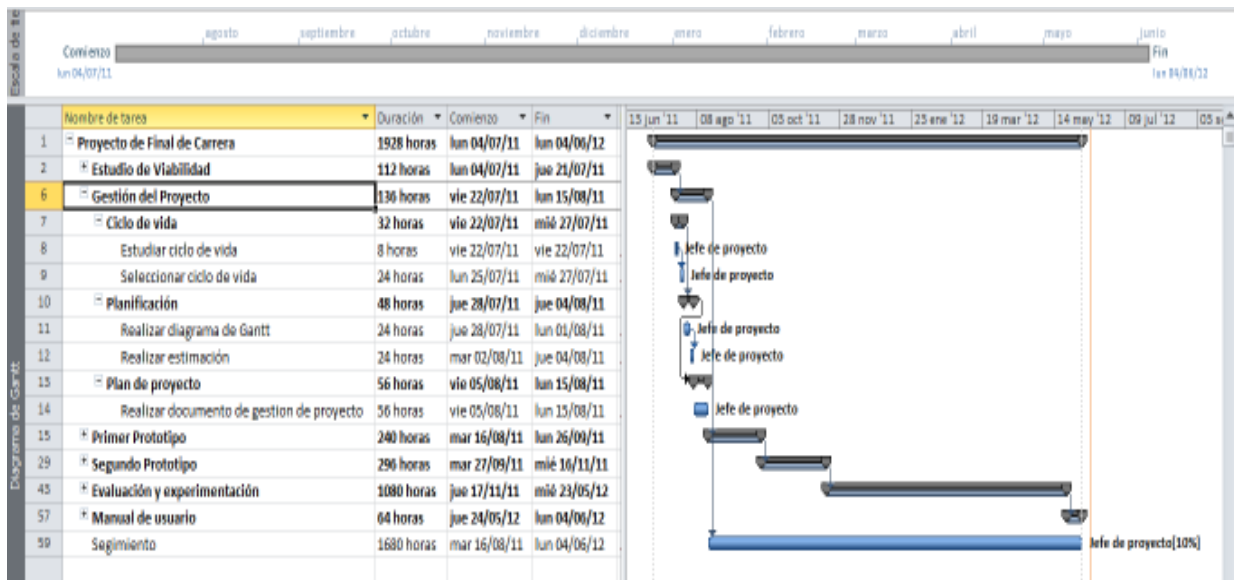


Si el proyecto es considerado viable, comienza la fase de gestión del proyecto. Ésta también es llevada a cabo por el jefe de proyecto, el cual realiza un estudio sobre los ciclos de vida existentes, selecciona el más idóneo para el desarrollo de la aplicación, documentando claramente las razones y motivos de su elección.

A continuación, se realiza el diagrama Gantt en el cual figurarán todas las tareas y actividades necesarias para realizar el desarrollo de la aplicación, así como la duración estimada de cada una de ellas y los recursos necesarios que se utilizarán en cada una de las tareas. En este diagrama las tareas ya aparecen como finalizadas, puesto que ya se ha llegado al final del proyecto.

Por último, el jefe de proyecto ha de realizar el correspondiente documento de Gestión de Proyecto, en el que se incluyan toda la planificación y estimaciones realizadas, a su vez, comienza a realizar su seguimiento que se llevará a cabo en paralelo a todas las actividades a realizar hasta la entrega final.

En la siguiente figura se muestran las tareas que conforman la Gestión de Proyecto:



**Ilustración 22: Diagrama de Gantt de la gestión del proyecto**



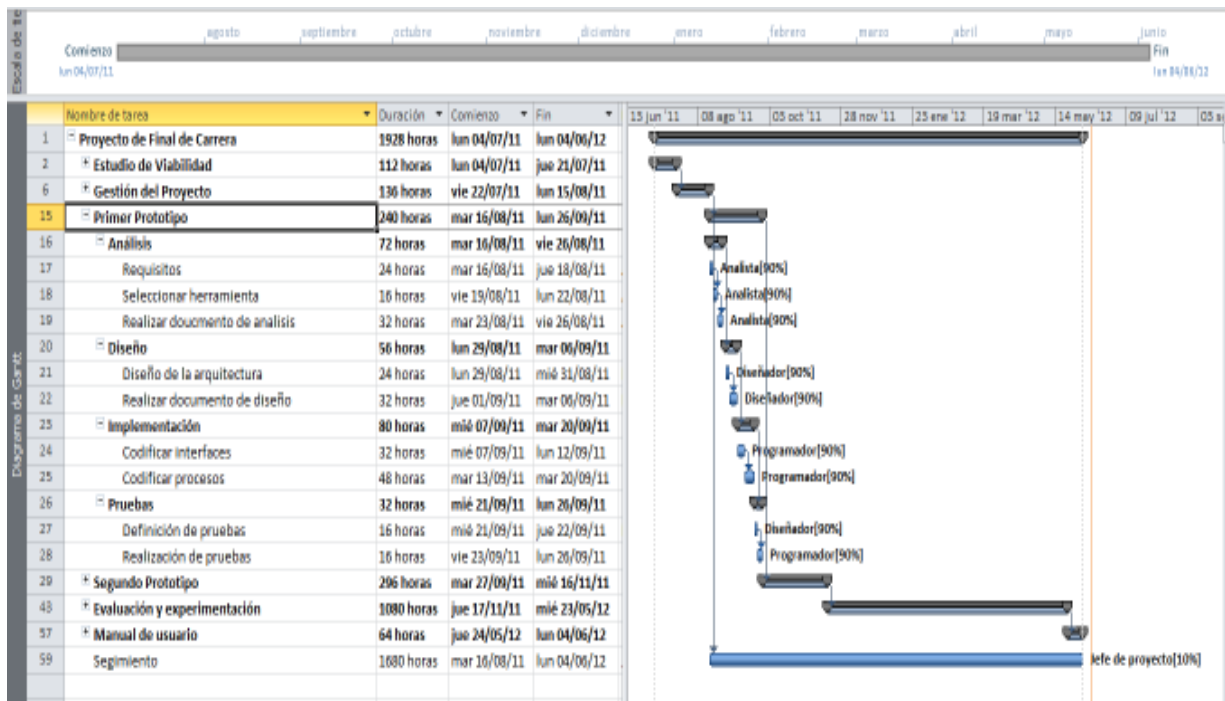
Una vez finalizadas todas las tareas de gestión del proyecto se comienza a desarrollar el primer el primer prototipo de la aplicación. En primer lugar el analista debe recopilar los requisitos del usuario para obtener los requisitos software y seleccionar las herramientas necesarias para satisfacerlos. Es muy importante que antes de crear un prototipo, el analista y el cliente o tutor trabajen estrechamente en la correcta identificación de los requerimientos a satisfacer.

El diseñador lee la documentación generada por el analista y diseña la arquitectura del sistema, definiendo para ello tanto las interfaces como los procesos a implementar.

A continuación será el programador el encargado de implementar la aplicación conforme al diseño realizado a fin de satisfacer los requisitos de usuario definidos en el proyecto.

El primer prototipo se concluye con una batería de pruebas, para verificar su correcto funcionamiento y la evaluación por parte del cliente o tutor. Es responsabilidad del cliente trabajar con el prototipo y evaluar sus características y operación. La experiencia del sistema, bajo condiciones reales, permite obtener la familiaridad indispensable para determinar los cambios o mejoras que sean necesarios, así como las características inadecuadas.

Las pruebas serán redactadas por el diseñador y ejecutadas por el programador para comprobar el correcto funcionamiento de la aplicación.



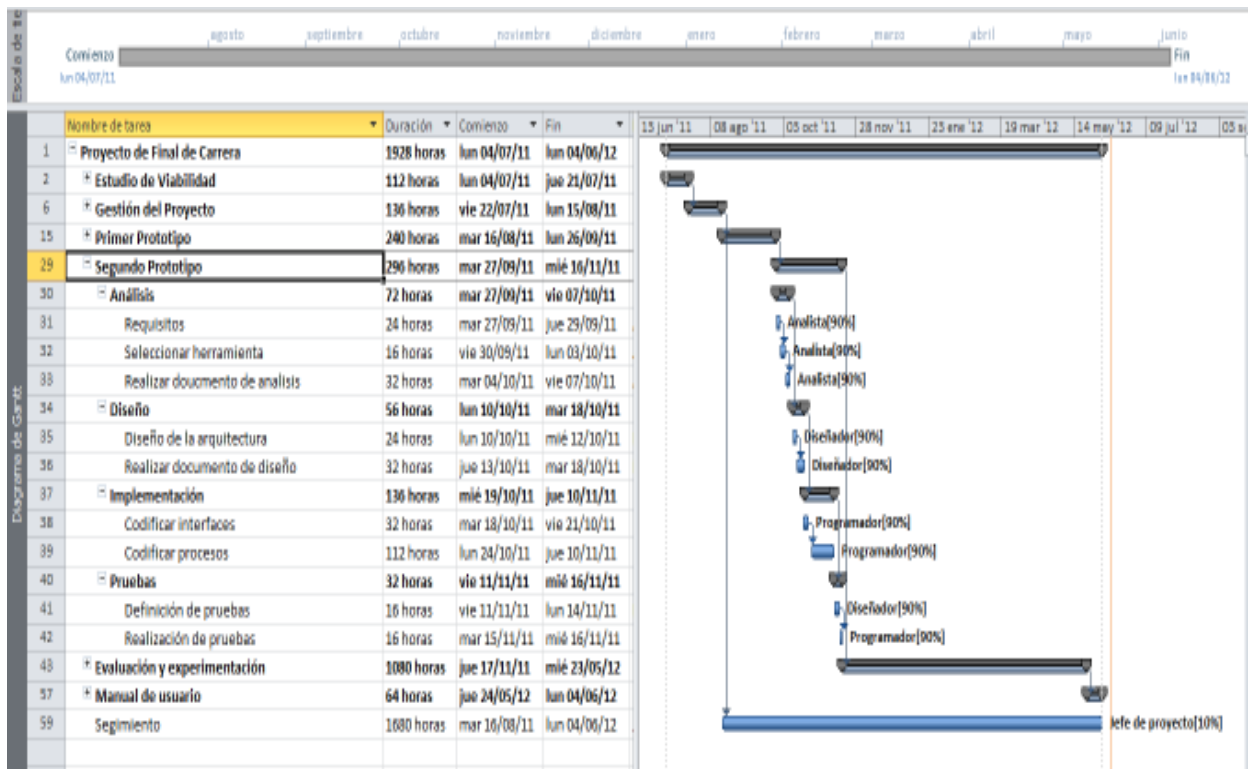
**Ilustración 23: Diagrama de Gantt del primer prototipo**

Finalmente se comienza con el último prototipo, en el que se intentarán satisfacer todos los requisitos expuestos por el cliente a lo largo del proyecto.

El analista es el encargado de recoger todos los requisitos y comentarios del usuario, elabora la documentación definitiva, y de requerirse busca, evalúa y selecciona cualquier herramienta que hasta ahora no hubiese sido necesario utilizar.

El diseñador plantea las interfaces y funcionalidades que todavía no han sido desarrolladas y el programador las implementa obteniendo una versión casi definitiva de la aplicación.

La fase de pruebas es crucial para determinar que los requerimientos han sido satisfechos y que el sistema se comporta como se esperaba. El prototipo debe pasar las pruebas de funcionalidad para validar su correcto funcionamiento y la evaluación por parte del cliente, siendo éste el que certifique la aceptación del sistema que utiliza.



**Ilustración 24: Diagrama de Gantt del segundo prototipo**

Finalizado el desarrollo en el que se va a apoyar el proyecto comienza la fase de evaluación y experimentación. En esta fase es la base del proyecto actual, ya que tal y como se ha indicado, el fin del mismo es encontrar la mejor configuración, teniendo en cuenta la relación eficiencia/consume, de un perceptron multicapa para el problema del lanzamiento de proyectiles a partir de los datos iniciales del proyecto. Durante el proceso se debe razonar y poner de manifiesto el por qué de cada una de las acciones realizadas hasta encontrar la dicha configuración. Eso objeto también de este proyecto evaluar dos métodos para el análisis de sensibilidad, así como comparar sus resultados.

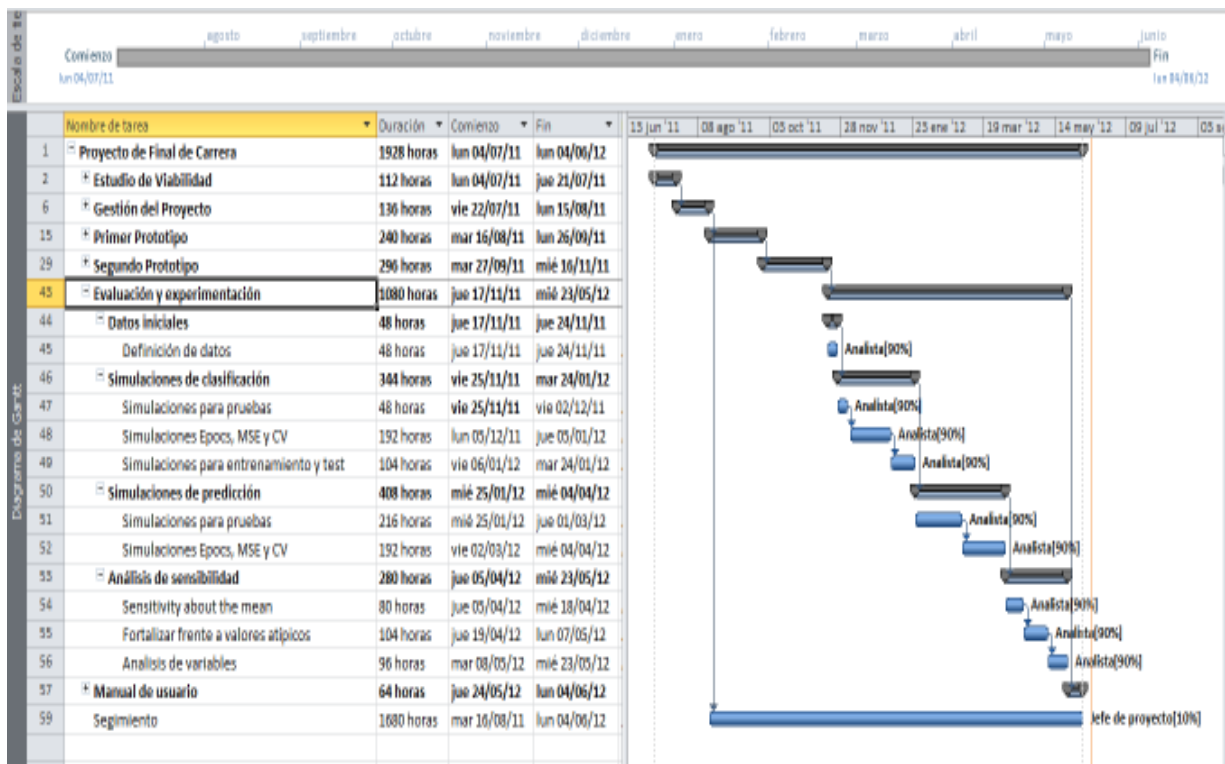
Inicialmente se analizarán los datos de entrada sobre los que se va a sustentar el proyecto.

A continuación se comenzará con el proceso para identificar la mejor configuración para la red para el problema de clasificación tomando como base la variable que indica si el proyectil ha atravesado la placa o no.



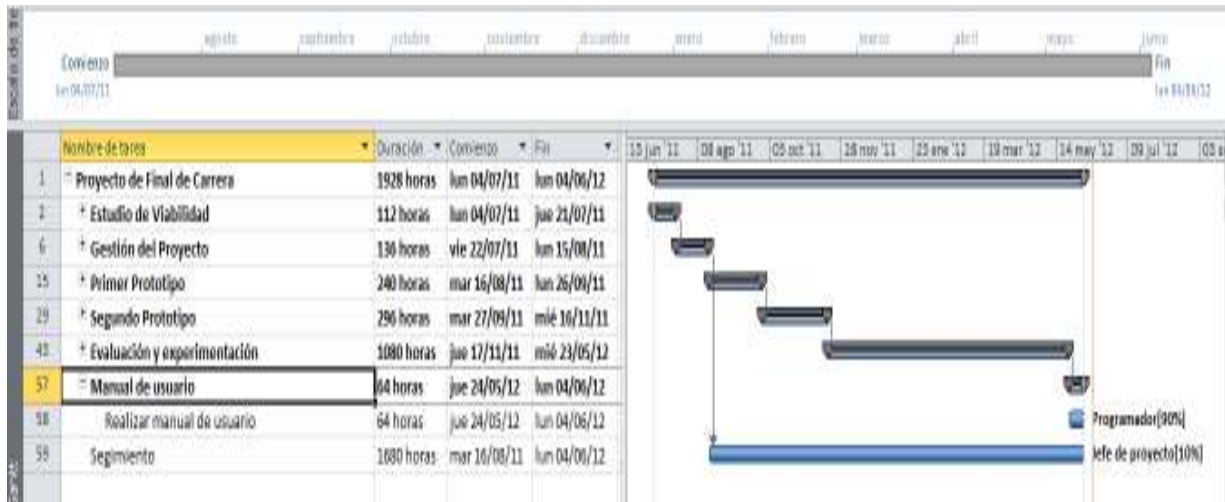
Se realizará un proceso análogo pero esta vez para un proceso de simulación para las variables de masa fina, masa inicial, velocidad residual y profundidad de perforación de la placa.

Por último se llevará a cabo los análisis de sensibilidad, se obtendrán sus resultados y se compararán intentando averiguar cuál de ellos es más fiable y en qué circunstancias.



**Ilustración 25: Diagrama de Gantt de la evaluación y experimentación**

Finalmente se genera un manual de usuario indicando todos los aspectos de la aplicación, tanto sus funciones como la manera de ejecutarlas o seleccionarlas, para que el usuario final u otra persona sea capaz de utilizar la nueva aplicación sin problemas, y no tener que recurrir a los desarrolladores.



**Ilustración 26: Diagrama de Gantt del manual de usuario**

El último paso es la entrega de la documentación y el software desarrollado al cliente. Tras este paso se da por concluido el proyecto el día 4 de Junio de 2012.



#### 4.4. Estimación de costes

Para calcular el coste estimado del proyecto se deben tener en cuenta tanto los gastos materiales como los costes humanos del desarrollo.

A continuación se muestra el presupuesto en forma de tabla de las herramientas utilizadas para el desarrollo del proyecto:

<i>Herramientas</i>	<i>Coste</i>
Microsoft Office 2007	379 €
Microsoft Windows XP	199 €
Microsoft Visual Studio 2005	550 €
Neurosolutions	573 €
<b>Total</b>	<b>1.701 €</b>

**Tabla 47: Coste del software**

En lo referente al tema de hardware y comunicaciones se ha estimado el siguiente coste. Se ha obviado el coste de las instalaciones por ser un proyecto a realizar fuera de la oficina, y se ha depreciado el coste de luz.

<i>Infraestructura</i>	<i>Coste</i>
Equipo	500 €
Internet	528 €
<b>Total</b>	<b>1.028 €</b>

**Tabla 48: Coste de infraestructura**

Para el cálculo del coste humano

<i>Perfil</i>	<i>horas</i>	<i>Precio/hora</i>	<i>Total</i>
Jefe de proyecto	466,4	37 €	17.257 €
Analista	1.101,6	33 €	36.353 €
Diseñador	129,6	33 €	4.277 €
Programador	230,4	19 €	4.378 €
<b>Total</b>	<b>1928</b>	<b>30,5 €</b>	<b>62.264 €</b>

**Tabla 49: Coste humano**



Por tanto, el coste total del proyecto sería

<i>Costes</i>	<i>Coste</i>
Herramientas	1.701 €
Infraestructura	1.028 €
Humanos	62.264 €
<b>Total</b>	<b>64.993 €</b>

**Tabla 50: Coste total**

El precio total del proyecto una vez incluido el IVA sería de **76.691,74 €**





## Capítulo 5. MANUAL DE USUARIO "SIMULADOR"

### 5.1. Introducción

Simulador es la herramienta desarrollada en el proyecto para permitir la automatización de los procesos de creación de ficheros de pruebas, entrenamiento/testeo de la red, y la generación de informes y resúmenes de los resultados proporcionados por esta. El software ha sido desarrollado con Visual Basic 6.0 y Visual Basic for Applications.

La aplicación está diseñada conforme al flujo lógico del proceso a automatizar con la herramienta.



**Ilustración 27: Bloques de Simulador**

Es decir, una primera fase para la generación de ficheros, la siguiente centrada en el entrenamiento y testeo de la red, y por último, una fase cuya función es la de generar informes de resumen en base a los resultados obtenidos en el testeo de la red. Aún así, cualquiera de los tres bloques de la aplicación es autónomo, no necesita de la ejecución del proceso anterior para realizar su cometido. De esta manera no se le obliga al usuario a realizar el flujo completo, al contrario, se le da la posibilidad de dividir en distintas ejecuciones las tareas que desee llevar a cabo en función de sus necesidades.

Todos los módulos siguen un mismo patrón, primero se configuran los parámetros necesarios para su ejecución, se comprueba si son correctos, y de ser así, se ejecutan los procesos asociados al módulo en cuestión.

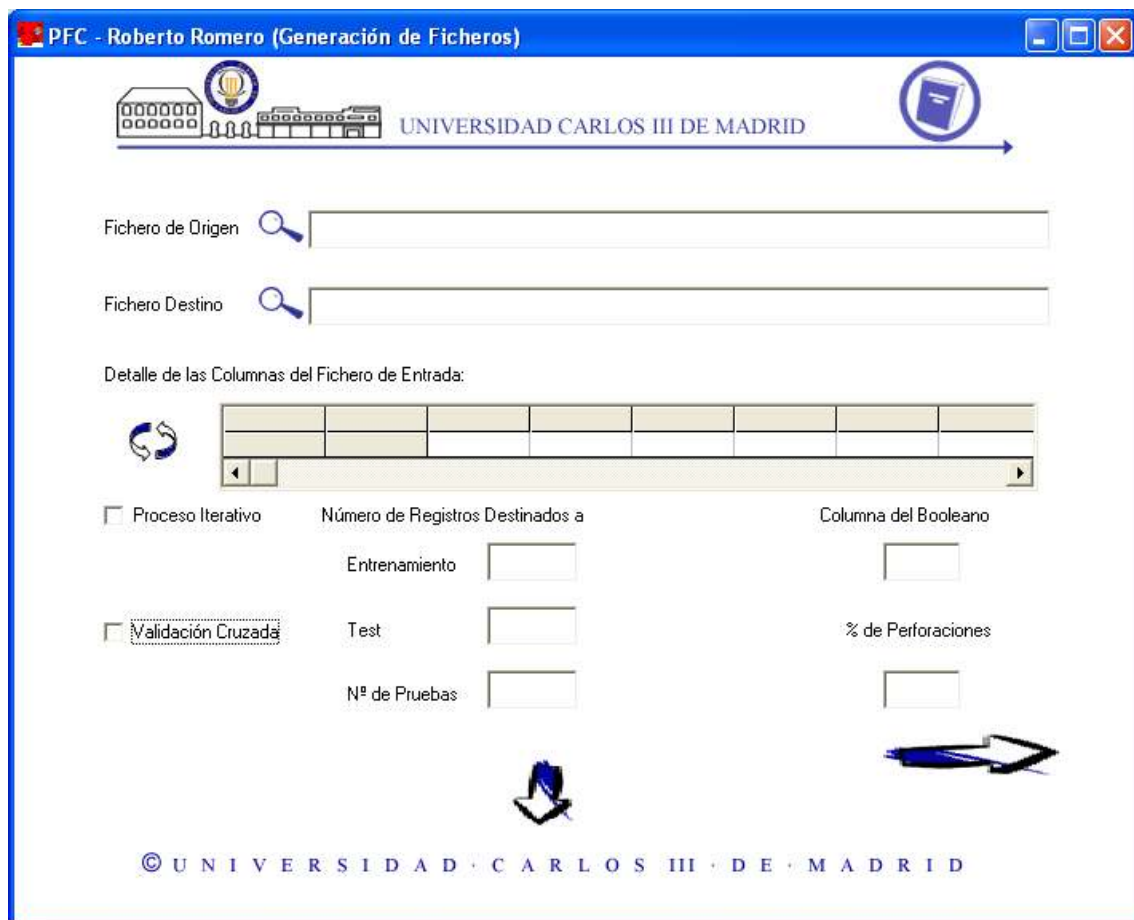


## 5.2. Generación de ficheros

### 5.2.1. INTRODUCCIÓN

Este es el modulo inicial de la aplicación. Su misión es generar los ficheros necesarios para realizar pruebas de entrenamiento y testeo de una red a partir de los datos proporcionados en un fichero origen.

Este modulo tiene dos comportamientos básicos, uno diseñado para la creación de baterías de pruebas unitarias, y otra, para la creación iterativa de baterías de pruebas. La configuración iterativa permite crear "n" baterías de pruebas incluyendo variaciones en la configuración de cada una de ellas respecto al número de pruebas de la batería, la cantidad de registros de los ficheros de entrenamiento y de testeo.



**Ilustración 28: Generación de ficheros**




### 5.2.2. PARÁMETROS DE ENTRADA

A continuación se detallan los parámetros de configuración necesarios para la ejecución del módulo:

- **Fichero de entrada:** ruta del fichero que contiene los datos de origen. Cualquier formato valido para Excel lo será también para Simulador. El fichero debe estar compuesto por una primera fila que debe contener el nombre de las variables que tiene una prueba, siendo siempre la primera el identificador de la misma (id), y 'n' filas sucesivas con los valores obtenidos en cada una de ellas.
- **Fichero de destino:** se debe especificar la ruta y el nombre base que se le desea asignar a la carpeta donde se van a guardar los ficheros generados
- **Detalle de las columnas de fichero de entrada:** Una vez seleccionado el fichero de entrada, se debe indicar los campos que se desean incluir en la batería de pruebas a generar. Para ello, se debe pulsar sobre el icono de refrescar, entonces la tabla adyacente muestra la cabecera del fichero origen; debajo de cada campo de la cabecera existe un checks que permite indicar si se desea que la variable bajo la que está situado debe aparecer en los ficheros de pruebas que se van a generar.

Detalle de las Columnas del Fichero de Entrada:



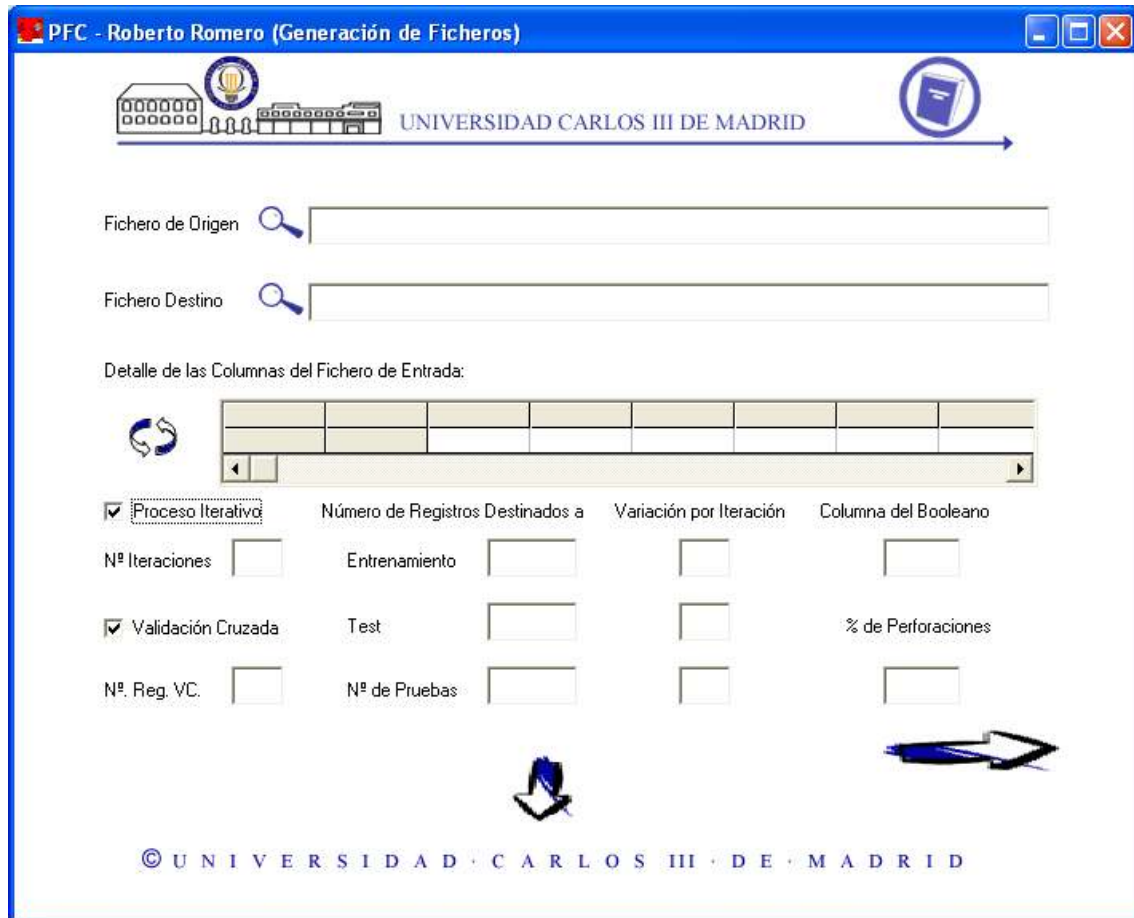
Columnas ==>	NºEnsayo	GROSORPLACA(mm)	RADIOPROY(mm)	LONGPROY(mm)	VEL(m/s)	MASAIT
Checks ==>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>

**Ilustración 29: Selección de datos de entrada**

- **Proceso iterativo:** Indica si se va a realizar un proceso unitario o no. Cuando está seleccionado el proceso es iterativo, y entonces se muestran los campos adicionales para completar esta configuración.



- **Validación cruzada:** Indica si se van a generar ficheros para la validación cruzada en la batería de pruebas. Si está seleccionado, se muestra una caja de texto donde se debe introducir el número de registros que se destinarán a CV.
  
- **Numero de registros destinados a**
  - **Entrenamiento:** campo donde se indica la cantidad de registros que debe contener cada fichero de entrenamiento de la batería de pruebas.
  - **Test:** campo donde se indica la cantidad de registros que debe contener cada fichero de testeo de la batería de pruebas.
  
- **Nº de pruebas:** se deben indicar la de pruebas que debe contener la batería. Es decir, el numero de registros que se deben generar de entrenamiento, test y de validación cruzada si procediese.
  
- **Columna del booleano:** Si se quiere realizar un estudio de clasificación es necesario que se índice en este campo el numero de la columna del fichero de origen donde se encuentra la variable booleana.
  
- **% de perforación:** Si se quiere realizar un estudio de clasificación es necesario que se indique el porcentaje de registros que cada fichero de la batería debe de contener de registros con la variable booleana a "Si".



**Ilustración 30: Generación de ficheros iterativo**

- **Variación por iteración**

- **Entrenamiento:** campo donde se indica el número de registros que deben sumarse o restarse por iteración al número de registros destinados a entrenamiento.
- **Test:** campo donde se indica el número de registros que deben sumarse o restarse por iteración al número de registros destinados a test.
- **Nº de pruebas:** campo donde se indica el número en el que debe aumentar o disminuir la cantidad de pruebas generadas en cada iteración.



### 5.2.3. PROCESO DE GENERACIÓN

A continuación se describe el proceso que realiza Simulador durante la creación de los ficheros de una batería de pruebas. Ficheros de entrenamiento, testeo y validación cruzada, si procediese. En adelante el proceso se explicará como si CV estuviese seleccionada.

Para evitar, en la medida de lo posible, el sobreaprendizaje de la red se impone una restricción la restricción lógica de que los ficheros generados para una prueba de una batería deben ser disjuntos, es decir, en dos ficheros de una misma prueba nunca puede aparecer un mismo registro. Por ello, el proceso lo primero que hace es comprobar que la suma de los registros destinados entrenamiento, test y CV, es inferior o igual a la cantidad de registros que tiene el fichero origen. Esto es necesario independientemente de si la batería está diseñada para el entrenamiento de una red de clasificación o de predicción.

Además, si se tratase de una batería de pruebas para clasificación se debe cumplir un requisito adicional. Los registros de cada fichero destino debe mantener la proporción necesaria, es decir, el usuario debe indicar cuál debe ser el porcentaje de los registros en función de la columna discriminante de fichero origen (columna booleana) que deben cumplir los ficheros resultantes. Para poder comprobarlo es necesario conocer la siguiente información: el número de registros que debe contener cada uno de los ficheros con la columna booleana a true, y por ende, los que la deben tener .false. Sólo, si el total de los registros con valor true, es decir, el sumatorio de los registros necesarios para generar los ficheros resultantes en la proporción indicada por el usuario, es inferior o igual al total de registros con valor true del fichero de origen, el proceso podrá ejecutarse. Y de forma recíproca, para valores false.

El proceso lo primero que realiza es una copia del fichero origen para trabajar sobre ella. Esto se hace para evitar que una terminación inesperada de la aplicación pudiese dejar corrupto el fichero de origen. El siguiente paso es crear contador con el número de registros que tiene el fichero origen y otro con el número de registros que



debe tener el fichero destino. Seguidamente se entra en un bucle del que no sé saldrá hasta que se hayan generado los ficheros de entrenamiento, test y CV para la prueba del bloque en cuestión.

Para baterías de pruebas de predicción por cada iteración del bucle, se lee un registro del fichero origen, el número del registro leído viene dado por una función random. Esta función siempre debe dar un valor que se encuentre entre 1 y el valor del contador del número total de registros en el fichero origen. El registro se copia en el fichero destino, se elimina del origen, y se resta uno al contador de origen y destino. Cuando el contador de destino es 0, el fichero se da por finalizado.

Para baterías de prueba de clasificación el proceso es básicamente igual pero se necesitan dos contadores para el fichero destino, uno con el número de registros que debe contener con la variable discriminante a "Verdadero" y otro para "Falso". La diferencia radica en que cuando un registro de origen es seleccionado se comprueba el valor de su columna booleana, y en función de su valor se resta uno a su contador correspondiente de destino. Cuando los contadores de registros del fichero destino llegan ambos a 0 el fichero ha finalizado. Pero, ¿Qué se hace cuando el contador del número de registro con valor, "Verdadero" o "Falso", es 0 y se lee en origen un registro con ese valor?, Pues el registro leído, no se elimina, si no que se coloca al final del fichero origen, y se resta uno al valor del contador del número total de registros en origen. Esto se hace para que ese registro, que para el fichero actual no es válido, pueda ser seleccionado en la creación de los siguientes ficheros (Entrenamiento, Testeo, CV). Al restar uno al contador de registros totales en origen, nos aseguramos que el random no nos vuelva indicar que accedamos a ese registro, ya que está en una posición superior al valor del contador de registros en origen

Cuando se pasa a generar el segundo fichero, el de testeo, se actualizan los contadores con los valores necesarios para el nuevo fichero y se realiza el proceso descrito anteriormente de nuevo. Se debe tener en cuenta que el fichero de origen tendrá los registros sobrantes de la iteración anterior, lo que nos asegura que cada uno de los



ficheros creados contiene un conjunto disjunto de registro en su interior. De forma recíproca se haría para el registro de CV

Cuando se han generado los tres ficheros (entrenamiento, test y CV) se ha concluido una prueba de la batería. El proceso precedente se debe repetir en su totalidad por cada prueba a generar.

Los ficheros tienen exactamente el mismo formato, se diferencian en el número de filas que contienen y la palabra clave por la que comienza su nombre "Entrenamiento\_...", "Test\_..." y "CV\_...".

NºEnsayo	GROSOR	RADIO	LONGITUD	VEL(m/s)	MASA	MAT.	MAT.	Penetración
	PLACA(mm)	PROY(mm)	PROY(mm)		INICIAL(g)	PROY	PLACA	
104	14	5	35	943	22	2	3	no
7	8	8	47	476	74	2	2	no
178	45	5	26	277	0	4	2	no
5	9	5	30	529	18	2	2	no
207	10	4	22	672	0	3	2	si
52	11	6	37	912	33	2	2	si
192	10	6	41	699	0	3	2	si
144	9	4	31	1187	29	3	2	si
142	7	4	29	884	11	2	2	si
68	6	4	30	716	12	2	2	si
107	8	8	39	648	146	3	2	si
106	17	7	19	870	54	4	2	si
85	8	5	35	982	22	2	2	si
154	10	6	31	980	65	4	2	si
112	4	5	24	1007	35	3	1	si
...	...	...	...	...	...	...	...	...

**Tabla 51: Contenido de ficheros**



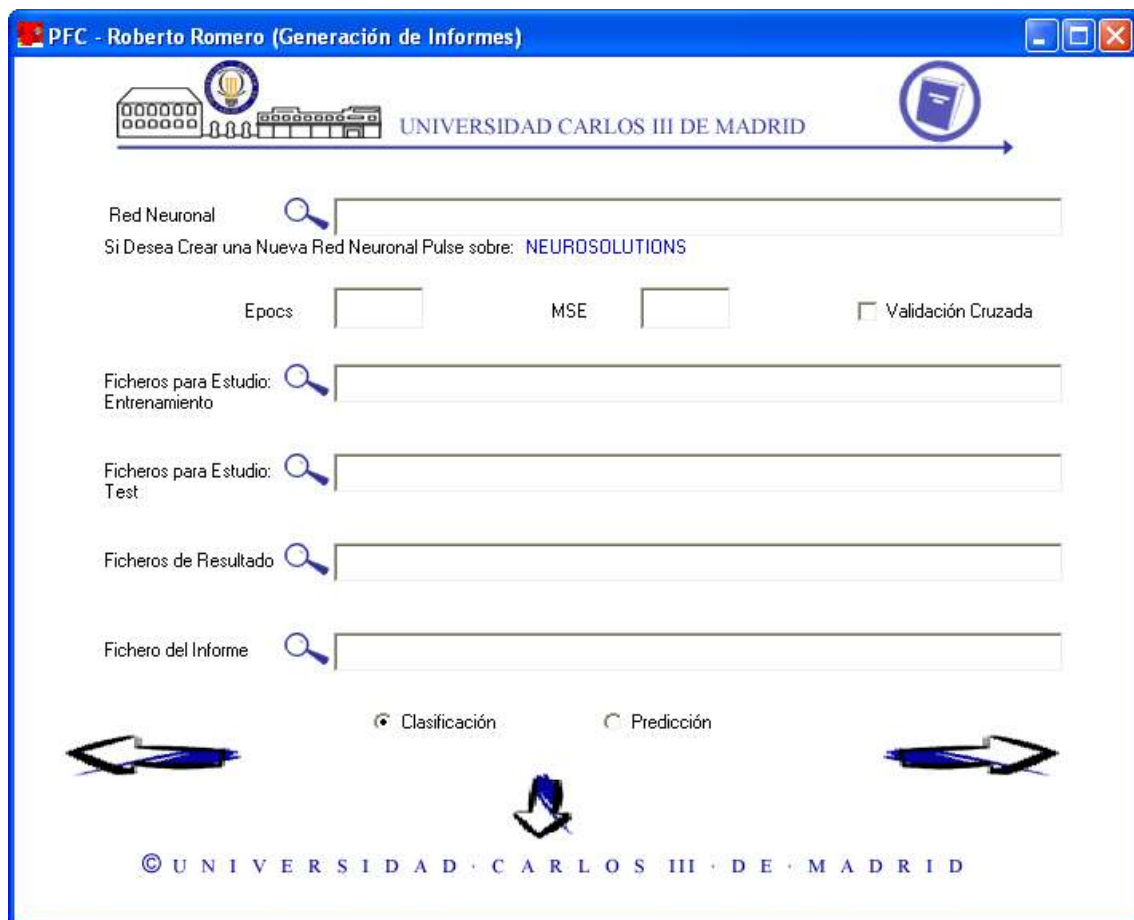


## 5.3. Generación de informes

### 5.3.1. INTRODUCCIÓN

Este es el modulo encargado de entrenar y probar la red con las baterías de pruebas realizadas en la fase anterior. Con información del proceso devuelta por la red Simulador debe generar unos informes de resultados para cada batería de pruebas ejecutada en el proceso.

Este modulo tiene dos comportamientos básicos que vendrán discriminados por la configuración establecida en la fase anterior. Si la generación de informes es configurada para un proceso iterativo, esta fase se comportará de la misma manera, es decir, realizará el entrenamiento, testeo e informe de resultados de cada una de las baterías definidas en la fase anterior, en caso contrario, lo hará únicamente de la batería seleccionada.



**Ilustración 31: Generación de informes**



### 5.3.2. PARÁMETROS DE ENTRADA

A continuación se detallan los parámetros de configuración necesarios para la ejecución del módulo:

- **Red Neuronal:** ruta del fichero “.nbs” que contiene la red de neuronas sobre la que se va ejecutar el proceso. Si se desea crear la red en el momento de la configuración del proceso, basta con pulsar sobre el enlace “[NEUROSOLUTIONS](#)”, entonces se abrirá la herramienta y permitirá crear la red sobre la que después se ejecutará el proceso.
- **Epocs (ciclos):** es un criterio de parada de la red, especifica el número de veces que se la van a presentar las pruebas de un fichero de entrenamiento a la misma, siempre y cuando, ningún otro criterio de parada se haya cumplido antes.
- **MSE (Error cuadrático medio):** Es un criterio de parada de la red, permite medir el error global del pronóstico. Cuando esta cota sea menor o igual al valor indicado en este campo el proceso de entrenamiento de la prueba termina, siempre y cuando, ningún otro criterio de parada se haya cumplido antes.
- **Validación Cruzada:** Es un criterio de parada de la red. Este método controla el error en un conjunto independiente de datos, su principal característica es la de detectar el sobreaprendizaje de la red. Detiene el entrenamiento cuando detecta que el error comienza a aumentar, siempre y cuando, ningún otro criterio de parada se haya cumplido antes.
- **Ficheros para estudio:**
  - **Entrenamiento:** Ruta donde se encuentra el primer fichero de entrenamiento de la red de una batería de pruebas, Si el proceso es iterativo, este campo estará relleno por defecto.



- **Test:** Ruta donde se encuentra el primer fichero de testeo de la red de una batería de pruebas, Si el proceso es iterativo, este campo estará relleno por defecto.
- **Ficheros de resultados:** Se debe especificar la ruta y la palabra clave con la que se desea que se genere el nombre del fichero de resultados de una prueba. El nombre de un fichero de resultados estará formado por "*palabraclave\_Salida\_n*" Siendo *n* el número de la prueba a la que pertenecen los resultados.
- **Fichero del informe:** Se debe especificar la ruta y la palabra clave con la que se desea que se genere el nombre del fichero de informe de resultados de una batería de pruebas. El nombre del fichero de informe estará formado por "*palabraclave\_Informe\_n\_i\_j*" Siendo *n* el número de registros destinados a entrenamiento en la batería de pruebas, *i* el número de registros destinados a testeo, y *j* el número de pruebas ejecutadas en la batería.
- **Clasificación / Predicción:** Es un campo booleano que permite indicar el tipo de entrenamiento que debe realizar la red neuronal.

### 5.3.3. PROCESO DE ENTRENAMIENTO E INFORMES

A continuación se describe el proceso que realiza Simulador durante el entrenamiento de la red, la obtención de resultados durante el testeo de la misma y la generación de informe asociado a cada batería de pruebas lanzada en la herramienta.

La explicación que a continuación se va a realizar abarca una batería de pruebas, para un proceso iterativo, se repite el proceso tantas veces como baterías hayan sido configuradas en la fase anterior. Con el objetivo de que la explicación abarque todas las posibilidades el supuesto de la misma tendrá ficheros de validación cruzada.



Después de realizar las comprobaciones lógicas pertinentes, campos obligatorios, datos correctos, directorios y ficheros accesibles, formatos validos..., lo primero que hace Simulador es iniciar el proceso de entrenamiento y testeo de la red. Simulador mediante el API de Neurosolutions carga los ficheros de entrenamiento, y testeo, a continuación se cargan los distintos criterios de parada, el fichero de CV, el número de Elocs y el MSE. Seguidamente se ejecuta el entrenamiento, y después el testeo de la red; el resultado producido se analiza con el fin de ir preparando el informe de resultados.

En función del tipo de estudio que se realice, clasificación o regresión, la información que presenta el informe y su formato varía.

**Clasificación:** para cada fichero de entrenamiento se obtiene el máximo y mínimo de cada una de las variables de entrada Se analiza el fichero de resultados para obtener el número de aciertos y fallos. Para cada acierto o fallo se ejecuta la comprobación de fuera de rango, es decir, se contabilizan el número de veces que el valor de la variable en el testeo está fuera del rango de valores sobre los que se entrenado la red, y si esta acierte o falla en su clasificación. Esta información es muy útil en los análisis de sensibilidad de las variables. Toda esta información, más totales y tiempos, es lo que se presenta en un informe de una batería de pruebas para un estudio de clasificación.

N. Ensayo	Aciertos	Fallos	Test	GROSOR PLACA(mm)	RADIO PROY(mm)	LONG PROY(mm)	VEL(m/s)	MASA INICIAL(g)	MATERIAL PROY	MATERIAL PLACA
1	37	3	40	0 0	0 0	1 0	2 0	0 0	0 0	0 0
2	36	4	40	0 0	0 0	0 0	2 0	0 0	0 0	0 0
...	...	...	...	...	...	...	...	...	...	...
49	39	1	40	0 0	0 0	0 0	2 0	1 0	0 0	0 0
50	39	1	40	1 0	0 0	0 0	0 0	0 0	0 0	0 0
Totales	1896	104	2000	19 0	0 0	25 1	17 1	15 1	0 0	0 0
Elocs	10000	MSE	0,1	VC	15	Pruebas	50	Entrenamiento	150	Duración 190,47 s.

Tabla 52: Detalle de un informe clasificación



**Predicción:** en este proceso tiene como finalidad el cálculo del coeficiente de correlación entre el resultado real y el predicho por la red en cada proceso de testeo. Este coeficiente determinará el grado de fiabilidad con el que ha aprendido la RNA. La correlación media obtenida tras la ejecución de cada prueba es la que se incluye en el informe, así como las media total de la batería y el tiempo requerido hasta la finalización del proceso.

<i>Ensayo</i>	<i>Test</i>	<i>(Mi-Mf)/Mi(%)</i>
1	40	0,88242776
2	40	0,926911875
...	...	...
49	40	0,875894054
50	40	0,934087284
Totales	2000	0,887516684
Epocs	10000	MSE
0,1 VC 15 Pruebas 50 Reg. Entrenamiento 150 Duración 152,89 segundos		

**Tabla 53: Detalle de un informe predicción**



## 5.4. Generación del Resumen

### 5.4.1. INTRODUCCIÓN

Este es el modulo encargado generar el fichero de resumen de un estudio, en otras palabras, genera un resumen a partir de los ficheros de informes obtenidos en la fase anterior.



**Ilustración 32: Generación de resumen**

### 5.4.2. PARÁMETROS DE ENTRADA

A continuación se detallan los parámetros de configuración necesarios para la ejecución del módulo:



- **Ficheros de informes:** ruta donde Simulador va a recoger todos los ficheros de informes con lo que elaborar un resumen de los mismo.
- **Fichero de resumen:** ruta y nombre del fichero de resumen que va a generar Simulador a partir de los ficheros de informes
- **Clasificación / Predicción:** Es un campo booleano que permite indicar el tipo de resumen que debe generar la herramienta.

#### 5.4.3. PROCESO DE RESUMEN

A continuación se describe el proceso que realiza Simulador para la generación del fichero resumen de un estudio. Por lógica, es obligatorio que todos los ficheros de informes pertenezcan al mismo estudio. Lo que se pretende con este resumen es facilitar al investigador la obtención de conclusiones, se trata de permitir ver en un mismo documento la configuración con la que se ha ejecutado cada batería de pruebas del estudio, así como los resultados obtenidos, medias y tiempos.

Tras realizar las comprobaciones lógicas pertinentes, campos obligatorios, directorios y ficheros accesibles, formatos validos..., y siempre que todo se a correcto, se inicia el proceso de generación de resumen. En función del tipo de estudio que se realice, clasificación o regresión, la información que presenta el informe y su formato varía.

**Clasificación:** Este formato de resumen consta de dos partes, una que se centra en el grado de fiabilidad de la clasificación de la red y otra en la influencia que tienen los valores atípicos durante el testeo de la misma.

La primera parte obtiene cada uno de los informes sobre los que se va a generar el resumen la configuración de la red, los totales generales y los tiempos. Estos valores, más los porcentajes de acierto/fallos calculados, son los datos que se presentan. La segunda parte calcula a partir de los informes el porcentaje de valores atípicos que han producido error respecto del total de ellos. Este porcentaje ayuda a discernir el posible peso que tiene cada variable en el estudio en cuestión.



Cód.	Entrena.	Test	Nº. Ficheros	Epocs	MSE	CV	Aciertos	Fallos	Total	%Aciertos	%Fallos	Duración
1	150	40	100	1	0,01	No	2812	1188	4000	70,3	29,7	314,13 s.
2	150	40	150	1	0,01	No	4468	1532	6000	74,46666667	25,53333333	490,48 s
...	...	...	...	...	...	...	...	...	...	...	...	...
47	150	40	200	10000	0,1	15	7678	322	8000	95,975	4,025	879,05 s.
48	150	40	50	10000	0,1	15	1896	104	2000	94,8	5,2	190,47 s.

**Tabla 54: Resumen de clasificación 1**

Código	Entrena.	Test	%GROSOR PLACA(mm)	%RADIO PROY(mm)	%LONG PROY(mm)	%VEL(m/s)	%MASA INICIAL(g)	%MATERIAL PROY	%MATERIAL PLACA
1	150	40	22,85714286	0	57,14285714	20,45454545	0	0	0
2	150	40	7,142857143	0	44,68085106	6,451612903	15,90909091	0	0
...	...	...	...	...	...	...	...	...	...
47	150	40	0	0	1,298701299	7,575757576	27,65957447	0	0
48	150	40	0	0	3,846153846	5,555555556	6,25	0	0

**Tabla 55: Resumen de clasificación 2**

**Predicción:** Este resumen presenta todos los resultados generales obtenidos en cada uno de los informes generados en el estudio. Esto permite de un simple vistazo sacar conclusiones generales sobre las configuraciones más óptimas para el aprendizaje de la red.

Código	Entrenamiento	Test	Pruebas	Epocs	MSE	CV	Total	MasaFinal(g)	Duración
1	150	40	100	1	0,01	No	4000	0,352632119	65,47 s.
2	150	40	150	1	0,01	No	6000	0,35332745	97,61 s.
...	...	...	...	...	...	...	...	...	...
47	150	40	200	10000	0,1	15	8000	0,992464529	1795,50 s.
48	150	40	50	10000	0,1	15	2000	0,993821669	423,20 s.

**Tabla 56: Resumen de predicción**





## Capítulo 6. EVALUACIÓN Y EXPERIMENTACIÓN

### 6.1. Datos iniciales

Para la realización de este proyecto se ha utilizado los datos obtenidos mediante la ejecución de 220 ensayos de impactos de proyectiles sobre placas. Debemos tener en cuenta la criticidad de los datos iniciales, a la hora de la realización de un proceso de prueba y testeo de una RNA, tanto en calidad como en variedad. Cuanto más fidedignos sean estos, más fiables serán los datos obtenidos por la red. Cuanto más variados sean los datos, más patrones aprenderá la red, mayor capacidad de generalización, lo que le permitirá predecir de forma eficiente una mayor cantidad de los mismos.

La información está dividida en las siguientes variables:

- N° Ensayo: Es el número de ensayo (de 1 a 220).
- Grosor Placa (mm): Es el grosor de la placa expresado en milímetros.
  - Rango de valores : 3 - 47
- Radio Proy (mm): Es el radio del proyectil expresado en milímetros.
  - Rango de valores : 3 - 8
- Long Proy (mm): Es la longitud del proyectil expresada en milímetros.
  - Rango de valores : 12 - 62
- Vel (m/s): Es la velocidad del proyectil expresada en metros por segundo.
  - Rango de valores : 270 - 1195
- Masa Inicial (g): Es la masa del proyectil antes del impacto expresada en gramos.
  - Rango de valores : 0 - 221
- Material Proy: Es el material del proyectil, codificado mediante un número.
  - Rango de valores : 2 - 4
- Material Placa: Es el material de la placa, codificado mediante un número, al igual que en el caso del material del proyectil.
  - Rango de valores : 1 - 3



- Masa Final (g): Es la masa del proyectil una vez que ha impactado sobre la placa expresada en gramos.
  - Rango de valores : 0 – 148,4
- (Mi-Mf)/Mi (%): Es el porcentaje de pérdida de masa del proyectil tras impactar sobre la placa.
  - Rango de valores : 0 – 89,6
- Penetración: Es un campo booleano que indica si el proyectil ha atravesado completamente la placa (perforación), indicándolo mediante un “si” en ese caso, y un “no” en caso contrario.
  - Rango de valores : Si - No
- Vres (m/s): Es la velocidad residual del proyectil cuando sí atraviesa la placa completamente, expresada en metros por segundo.
  - Rango de valores : 0 - 1170
- DOP (mm): Representa la profundidad de perforación en la placa expresada en milímetros, en el caso de que el proyectil no la atraviese completamente.
  - 0 – 25

Detalle del fichero con los datos iniciales usados en el proyecto.

A	B	C	D	E	F	G	H	I	J	K	L	M
	GROSOR PLACA [mm]	RADIO PROY [mm]	LONG. PROY [mm]	VEL [m/s]	MASA INICIAL [g]	MATERIAL PROY	MATERIAL PLACA	Masa Final [g]	(Mi-Mf)/Mi (%)	Penetración	Vres [m/s]	DOP [mm]
1	1	13	6	45	593	40	2	27	32	no	0	0.75
2	2	7	6	34	990	30	2	22	27	si	805	
3	3	4	4	21	654	8	2	7	16	si	265	
4	4	11	6	33	856	29	2	15	49	si	413	
5	5	9	5	30	529	18	2	14	24	no	0	6.4
6	6	9	5	31	1101	19	2	11	42	si	850	
7	7	8	8	47	476	74	2	68	8	no	0	20
8	8	11	6	34	569	30	2	22	27	no	0	8.5
9	9	12	6	45	499	40	2	31	22	no	0	6.5
10	10	13	7	55	934	66	2	44	34	si	712	
11	11	10	6	40	447	25	2	21	16	no	0	4
12	12	5	5	31	948	19	2	15	22	si	785	
13	13	15	7	56	453	68	2	57	16	no	0	4.8
14	14	6	4	25	608	18	2	7	29	no	0	9.8
15	15	15	7	48	668	58	2	31	47	no	0	14.8
16	16	4	4	24	1128	9	2	7	29	si	975	
17	17	9	8	44	827	89	2	62	25	si	611	
18	18	7	5	26	512	16	2	14	13	no	0	8.2
19	19	6	5	33	902	20	2	15	20	si	713	
20	20	7	5	28	645	17	2	12	32	no	0	25
21	21	5	4	27	910	11	2	6	30	si	704	
22	22	11	8	62	956	98	2	75	23	si	809	
23	23	7	7	33	601	40	2	32	20	si	208	
24	24	7	6	41	582	36	2	29	22	si	194	
25	25	12	7	42	543	51	2	38	25	no	0	11.1
26	26	11	7	52	1133	63	2	45	28	si	866	
27	27	16	8	53	1073	84	2	50	40	si	832	
28	28	8	7	33	1190	40	2	27	32	si	1002	
29	29	8	4	29	1004	11	2	6	48	si	680	
30	30	10	8	53	420	84	2	80	4	no	0	15.5
31	31	4	4	25	1076	10	2	7	27	si	920	
32	32	10	6	41	880	36	2	24	35	si	619	
33	33	9	5	30	1016	18	2	10	44	si	713	
34	34	5	4	24	1166	9	2	6	35	si	582	
35	35	4	3	14	804	3	2	1	57	no	0	13
36	36	14	7	49	1122	58	2	37	38	si	910	

Ilustración 33: Tabla de ensayos

El desglose de estos datos puede observarse en el anexo de la documentación



## 6.2. Simulaciones de clasificación

### 6.2.1. INTRODUCCIÓN

Nos encontramos ante un problema de clasificación siempre que la variable de salida a predecir sea de tipo booleano. Para las simulaciones de clasificación realizadas en este proyecto se ha tomado como referencia la variable “Penetración”. Centrando el estudio en dicha variable, y con la ayuda de Simulador, se ha realizado un conjunto de simulaciones para intentar identificar la mejor configuración para una RNA, de tipo Perceptrón multicapa, con algoritmo de aprendizaje backpropagation, para los datos actuales, e indicar, la notabilidad que tiene cada una de las variables de entrada en el resultado obtenido por la red.

El proceso realizado persigue definir la mejor configuración de la red en función de los resultados obtenidos, y, de los recursos empleados para ello. En nuestro caso, los recursos empleados lo mediremos en función del tiempo que necesite la red para conseguir dichos resultados. Dado que la capacidad computacional de las máquinas está en constante, e imparable aumento, se le ha asignado una mayor importancia a la fiabilidad de los resultados que a los recursos consumidos.

En el 70% de los registros utilizados en cualquier fichero de una prueba de simulación, es decir, de un fichero de entrenamiento, test o CV, tendrá la variable “Penetración” con el valor ‘Si’, y por ende, el resto a “No. Esta proporción se obtiene como aproximación a la proporción que guarda el fichero de datos de entrada, y deberá cumplirse en cualquiera de los ficheros utilizados.

El criterio inicial que se ha tomado para la asignación de las variables de entrenamiento, test y CV es el siguiente: El fichero de datos de entrada tiene 220 registros, y la experiencia nos dice, que para un buen testeo, se necesita una buena base de entrenamiento, por ello, se van a destinar 150 registros para entrenamiento, más o menos el 70 por cien del total; para el testeo de la red serán 40 registros, lo que viene a ser el 70 por ciento de los registros restantes; y, aplicando idéntico criterio, 15 para CV.



*Nota: La distribución inicial del número de registros se ha ajustado a semejanza de los estudios anteriores para los se disponían de 205 pruebas diferentes. Esto permitirá comparar resultados, si fuese necesario, de forma más ecuánime. [48] [49] [50] [51]*

Las variables que se modifican durante este proceso para la configuración de la RNA son las siguientes:

- Número de pruebas
- Número de Epocs
- Número de MSE
- Número de registro de entrenamiento
- Número de registro de test
- Número de registros de CV

Descripción del proceso para la obtención de la configuración de la red:

- Selección del número de pruebas óptimas para una correcta simulación.
- Selección de la configuración y valores óptimos para las variables de Epocs, MSE y CV.
- Selección del número de registros óptimos para el entrenamiento.
- Selección del número de registros óptimos para el test.

A continuación, se va a presentar en detalle el estudio realizado.



### 6.2.2. SIMULACIONES PARA PRUEBAS

Lo primero que vamos a intentar delimitar es el número de pruebas que debe tener cada simulación. Se han realizado 4 simulaciones, la primera con 50 pruebas, y el resto, incrementando este número en 50.

A las variables de Epocs y MSE, se les ha asignado los mismos valores que Neurosolutions da por defecto al crear una red, 1000 Epocs y 0,1 MSE.

Resumen de los valores definidos para:

- Número de pruebas: 50 | 100 | 150 | 200
- Número de registro de entrenamiento: 150
- Número de registro de test: 40
- Número de registros de CV: 15
- Número de Epocs: 1.000
- Número de MSE: 0,1

Código	Entrena.	Test	Pruebas	Epocs	MSE	CV	Aciertos	Fallos	Total	%Aciertos	%Fallos	Duración
1	150	40	50	1000	0,1	15	1902	98	2000	95,1	4,9	191,52 s.
2	150	40	100	1000	0,1	15	3837	163	4000	95,925	4,075	391,86 s.
3	150	40	150	1000	0,1	15	5718	282	6000	95,3	4,7	599,53 s.
4	150	40	200	1000	0,1	15	7581	419	8000	94,7625	5,2375	814,34 s.

**Tabla 57: Resultados de clasificación**

En función de los resultados que proporciona Simulador, salta a la vista la gran capacidad que tiene este tipo de RNA para el reconocimiento de patrones, simplemente con las características iniciales definidas en la red, esta obtiene un porcentaje muy elevado de fiabilidad. A parte de lo anteriormente expuesto, no podemos sacar grandes conclusiones de la tabla anterior, como mucho, podemos aventurarnos a insinuar que cuando se utilizan 200 pruebas para el entrenamiento la RNA puede estar sufriendo



**Proyecto:** Analisis de Técnicas para Medir la Influencia de las Variables del Vector de Entrada de un Perceptron Multicapa

**Autor:** Roberto Romero Martínez

**Capítulo:** Evaluación y experimentación

sobreaprendizaje (overfitting). Pero no tenemos pruebas concluyentes sobre la última observación.



### 6.2.3. SIMULACIONES PARA EPOCS, MSE Y CV

Dado que entre los resultados de las 4 primeras simulaciones, existe poco más de un punto de diferencia, para la siguiente parte del proceso, volveremos a variar el número de pruebas realizadas en cada simulación, además claro está, de variar el valor de las variables de CV, Epocs y MSE.

Esta parte del proceso se centra en las variables que marcan los criterios de parada del entrenamiento de la RNA. Ajustando bien estas variables, podremos disminuir los recursos necesarios para realizar una simulación, obteniendo incluso, mejores resultados.

Resumen de los valores definidos para:

- Número de pruebas: 50 | 100 | 150 | 200
- Número de registro de entrenamiento: 150
- Número de registro de test: 40
- Número de registros de CV: 0 - 15
- Número de Epocs: 1 | 1.000 | 10.000
- Número de MSE: 0,01 | 0,1

Código	Entrena.	Test	Pruebas	Epocs	MSE	CV	Aciertos	Fallos	Total	%Aciertos	%Fallos	Duración
1	150	40	100	1	0,01	No	2812	1188	4000	70,3	29,7	314,13 s.
2	150	40	150	1	0,01	No	4468	1532	6000	74,46666667	25,53333333	490,48 s.
3	150	40	200	1	0,01	No	5611	2389	8000	70,1375	29,8625	692,83 s.
4	150	40	50	1	0,01	No	1397	603	2000	69,85	30,15	149,05 s.
5	150	40	100	1	0,01	15	2890	1110	4000	72,25	27,75	327,78 s.
6	150	40	150	1	0,01	15	4417	1583	6000	73,61666667	26,38333333	517,38 s.
7	150	40	200	1	0,01	15	5703	2297	8000	71,2875	28,7125	643,31 s.
8	150	40	50	1	0,01	15	1373	627	2000	68,65	31,35	159,66 s.
9	150	40	100	1	0,1	No	2902	1098	4000	72,55	27,45	308,84 s.
10	150	40	150	1	0,1	No	4205	1795	6000	70,08333333	29,91666667	486,31 s.
11	150	40	200	1	0,1	No	5604	2396	8000	70,05	29,95	673,39 s.
12	150	40	50	1	0,1	No	1467	533	2000	73,35	26,65	152,27 s.
13	150	40	100	1	0,1	15	2789	1211	4000	69,725	30,275	335,11 s.



Código	Entrena.	Test	Pruebas	Epocs	MSE	CV	Aciertos	Fallos	Total	%Aciertos	%Fallos	Duración
14	150	40	150	1	0,1	15	4695	1305	6000	78,25	21,75	525,05 s.
15	150	40	200	1	0,1	15	6079	1921	8000	75,9875	24,0125	721,33 s.
16	150	40	50	1	0,1	15	1495	505	2000	74,75	25,25	161,36 s.
17	150	40	100	1000	0,01	No	3849	151	4000	96,225	3,775	434,72 s.
18	150	40	150	1000	0,01	No	5772	228	6000	96,2	3,8	664,41 s.
19	150	40	200	1000	0,01	No	7736	264	8000	96,7	3,3	916,98 s.
20	150	40	50	1000	0,01	No	1901	99	2000	95,05	4,95	210,72 s.
21	150	40	100	1000	0,01	15	3814	186	4000	95,35	4,65	379,20 s.
22	150	40	150	1000	0,01	15	5739	261	6000	95,65	4,35	586,73 s.
23	150	40	200	1000	0,01	15	7587	413	8000	94,8375	5,1625	894,50 s.
24	150	40	50	1000	0,01	15	1895	105	2000	94,75	5,25	186,34 s.
25	150	40	100	1000	0,1	No	3657	343	4000	91,425	8,575	321,81 s.
26	150	40	150	1000	0,1	No	5517	483	6000	91,95	8,05	502,75 s.
27	150	40	200	1000	0,1	No	7379	621	8000	92,2375	7,7625	701,17 s.
28	150	40	50	1000	0,1	No	1853	147	2000	92,65	7,35	155,09 s.
29	150	40	100	1000	0,1	15	3837	163	4000	95,925	4,075	391,86 s.
30	150	40	150	1000	0,1	15	5718	282	6000	95,3	4,7	599,53 s.
31	150	40	200	1000	0,1	15	7581	419	8000	94,7625	5,2375	814,34 s.
32	150	40	50	1000	0,1	15	1902	98	2000	95,1	4,9	191,52 s.
33	150	40	100	10000	0,01	No	3759	241	4000	93,975	6,025	773,64 s.
34	150	40	150	10000	0,01	No	5743	257	6000	95,71666667	4,283333333	1209,95 s.
35	150	40	200	10000	0,01	No	7644	356	8000	95,55	4,45	1585,88 s.
36	150	40	50	10000	0,01	No	1854	146	2000	92,7	7,3	361,42 s.
37	150	40	100	10000	0,01	15	3849	151	4000	96,225	3,775	383,33 s.
38	150	40	150	10000	0,01	15	5682	318	6000	94,7	5,3	604,75 s.
39	150	40	200	10000	0,01	15	7618	382	8000	95,225	4,775	1064,61 s.
40	150	40	50	10000	0,01	15	1917	83	2000	95,85	4,15	181,42 s.
41	150	40	100	10000	0,1	No	3663	337	4000	91,575	8,425	324,02 s.
42	150	40	150	10000	0,1	No	5488	512	6000	91,46666667	8,533333333	501,88 s.
43	150	40	200	10000	0,1	No	7314	686	8000	91,425	8,575	697,98 s.
44	150	40	50	10000	0,1	No	1827	173	2000	91,35	8,65	154,97 s.
45	150	40	100	10000	0,1	15	3826	174	4000	95,65	4,35	422,53 s.
46	150	40	150	10000	0,1	15	5704	296	6000	95,06666667	4,933333333	627,05 s.
47	150	40	200	10000	0,1	15	7678	322	8000	95,975	4,025	879,05 s.
48	150	40	50	10000	0,1	15	1896	104	2000	94,8	5,2	190,47 s.

**Tabla 58: Resultados de clasificación**

Ya que el número de simulaciones para esta parte del estudio es elevado, concretamente 48 simulaciones, y para que al lector le sea más fácil de seguir el por qué de las conclusiones a las que se ha llegado, a continuación, se presentan aquellos subconjuntos de la tabla anterior que nos permiten mostrar claramente las informaciones extrapolables a esta.





### 6.2.3.1. AJUSTANDO EL NÚMERO DE EPOCS I

Resumen de los valores definidos para el primer subconjunto:

- Número de pruebas: 50 | 100 | 150 | 200
- Número de registro de entrenamiento: 150
- Número de registro de test: 40
- Número de registros de CV: 0 - 15
- Número de Epocs: 1
- Número de MSE: 0,01 | 0,1

Código	Entrena.	Test	Pruebas	Epocs	MSE	CV	Aciertos	Fallos	Total	%Aciertos	%Fallos	Duración
4	150	40	50	1	0,1	No	1467	533	2000	73,35	26,65	152,27 s.
1	150	40	100	1	0,1	No	2902	1098	4000	72,55	27,45	308,84 s.
2	150	40	150	1	0,1	No	4205	1795	6000	70,08333333	29,91666667	486,31 s.
3	150	40	200	1	0,1	No	5604	2396	8000	70,05	29,95	673,39 s.
8	150	40	50	1	0,1	15	1495	505	2000	74,75	25,25	161,36 s.
5	150	40	100	1	0,1	15	2789	1211	4000	69,725	30,275	335,11 s.
6	150	40	150	1	0,1	15	4695	1305	6000	78,25	21,75	525,05 s.
7	150	40	200	1	0,1	15	6079	1921	8000	75,9875	24,0125	721,33 s.
12	150	40	50	1	0,01	No	1397	603	2000	69,85	30,15	149,05 s.
9	150	40	100	1	0,01	No	2812	1188	4000	70,3	29,7	314,13 s.
10	150	40	150	1	0,01	No	4468	1532	6000	74,4666667	25,53333333	490,48 s.
11	150	40	200	1	0,01	No	5611	2389	8000	70,1375	29,8625	692,83 s.
16	150	40	50	1	0,01	15	1373	627	2000	68,65	31,35	159,66 s.
13	150	40	100	1	0,01	15	2890	1110	4000	72,25	27,75	327,78 s.
14	150	40	150	1	0,01	15	4417	1583	6000	73,6166667	26,38333333	517,38 s.
15	150	40	200	1	0,01	15	5703	2297	8000	71,2875	28,7125	643,31 s.

**Tabla 59: Resultados de clasificación**

Lo primero que salta a la vista al fijarnos en los resultados de las simulaciones expuestas en la tabla anterior, es el escaso índice de acierto que tiene la RNA cuando el número de Epocs es 1, independientemente del valor que tengan el resto de los criterios de parada. Esta observación, es más que comprensible, ya que al realizar un único ciclo, a la red sólo se le presentan los entrenamientos una vez, por lo que no puede ajustar los



pesos de sus neuronas, impidiendo por tanto, alcanzar el mínimo global del error, punto en el que la red obtiene su mayor grado de fiabilidad.

El grado de fiabilidad de la red es bastante bajo, en torno al 70%. Para intentar aumentar esta cifra, y acorde con la explicación anterior de los resultados obtenidos, vamos a incrementar el número de la variable Epocs.

### 6.2.3.2. AJUSTANDO EL NÚMERO DE EPOCS II

Resumen de los valores definidos para el segundo subconjunto:

- Número de pruebas: 50 | 100 | 150 | 200
- Número de registro de entrenamiento: 150
- Número de registro de test: 40
- Número de registros de CV: 0
- Número de Epocs: 1.000 | 10.000
- Número de MSE: 0,1

Código	Entrena	Test	Pruebas	Epocs	MSE	CV	Aciertos	Fallos	Total	%Aciertos	%Fallos	Duración
28	150	40	50	1000	0,1	No	1853	147	2000	92,65	7,35	155,09 s.
44	150	40	50	10000	0,1	No	1827	173	2000	91,35	8,65	154,97 s.
25	150	40	100	1000	0,1	No	3657	343	4000	91,425	8,575	321,81 s.
41	150	40	100	10000	0,1	No	3663	337	4000	91,575	8,425	324,02 s.
26	150	40	150	1000	0,1	No	5517	483	6000	91,95	8,05	502,75 s.
42	150	40	150	10000	0,1	No	5488	512	6000	91,4666667	8,53333333	501,88 s.
27	150	40	200	1000	0,1	No	7379	621	8000	92,2375	7,7625	701,17 s.
43	150	40	200	10000	0,1	No	7314	686	8000	91,425	8,575	697,98 s.

**Tabla 60: Resultados de clasificación**

Podemos ver notable incremento del porcentaje de acierto de la red al aumentar el número de Epocs. Algo era más que lógico, pues ahora en el proceso de entrenamiento se presentan los entrenamientos 1.000 ó 10.000 veces, lo que posibilita la reconfiguración de los pesos en la RNA Si continuamos el análisis de los resultados



presentados en la tabla actual, una de las principales cosas que podemos destacar es, que cuando a la variable de MSE se le asigna el valor 0,1, la variable de número de Epocs no tiene, o tiene escasísima influencia, en el entrenamiento de la red. Esto se debe a que el criterio de parada que se cumple en la mayoría de las ocasiones es el de MSE menor 0,1. Esto se deduce claramente al observar que simulaciones con idénticas características, pero con distinto número de Epocs, obtienen unos resultados y tiempos muy similares.

El porcentaje de acierto de la red para esta batería de simulaciones está en torno al 92%. Para mejorar este porcentaje se van a afinar los criterios de parada actuales. Según lo anteriormente expuesto, deberíamos afinar el criterio del Error Cuadrático (MSE), que es el que actualmente tiene influencia sobre el entrenamiento, y ver qué pasa.

### 6.2.3.3. AJUSTANDO EL VALOR DEL MSE

Resumen de los valores definidos para el tercer subconjunto:

- Número de pruebas: 50 | 100 | 150 | 200
- Número de registro de entrenamiento: 150
- Número de registro de test: 40
- Número de registros de CV: 0
- Número de Epocs: 1.000 | 10.000
- Número de MSE: 0,01

Código	Entrena	Test	Pruebas	Epocs	MSE	CV	Aciertos	Fallos	Total	%Aciertos	%Fallos	Duración
20	150	40	50	1000	0,01	No	1901	99	2000	95,05	4,95	210,72 s.
36	150	40	50	10000	0,01	No	1854	146	2000	92,7	7,3	361,42 s.
17	150	40	100	1000	0,01	No	3849	151	4000	96,225	3,775	434,72 s.
33	150	40	100	10000	0,01	No	3759	241	4000	93,975	6,025	773,64 s.
18	150	40	150	1000	0,01	No	5772	228	6000	96,2	3,8	664,41 s.



Código	Entrena	Test	Pruebas	Epocs	MSE	CV	Aciertos	Fallos	Total	%Aciertos	%Fallos	Duración
34	150	40	150	10000	0,01	No	5743	257	6000	95,7166667	4,28333333	1209,95 s.
19	150	40	200	1000	0,01	No	7736	264	8000	96,7	3,3	916,98 s.
35	150	40	200	10000	0,01	No	7644	356	8000	95,55	4,45	1585,88 s.

Tabla 61: Resultados de clasificación

Según los valores de la tabla anterior, cuando el valor de la variable MSE es 0,01, la variable Epocs ahora sí pasa a tener influencia en el entrenamiento de la RNA. Al contrario de lo que sucede cuando el error cuadrático es 0,1, entrenamientos con idénticas características, pero con distinto número de Epocs, obtienen unos resultados muy diferentes, tanto en porcentaje de acierto, como en recursos consumidos, lo que nos indica que, con esta configuración, el criterio de parada por número de ciclos sí se cumple en determinadas ocasiones.

Es más, existen dos razones que nos permiten a afirmar que en la mayoría de las ocasiones el criterio de parada que se cumple es el de Epocs:

1. Si nos fijamos en los tiempos consumidos por cada simulación, se puede ver que para las que tienen el número de ciclos igual a 10000, el tiempo consumido es alrededor de un 43% más elevado que para las que tienen 1000, si se tiene en cuenta que el resto de variables tienen exactamente los mismo valores, podemos deducir lo anteriormente afirmado, que criterio de parada por número de ciclos se cumple en gran cantidad de ocasiones.
2. Por otro lado, al observar los porcentajes de acierto de la red, se denota que cuando el numero de Epocs es 10.000 la red siempre obtiene porcentajes de acierto inferiores a cuando es 1.000, lo que puede deberse a que la RNA está aprendiendo los ejemplos de memoria, por lo que su capacidad de generalización disminuye, y responde peor ante los supuestos no entrenados (sobreaprendizaje u overfitting). Esta información sugiere que el número de ciclos que se están realizando por entrenamiento en la gran mayoría de las ocasiones es mayor que 1.000;



y, por lo recursos consumidos en los entrenamientos con 10.000 ciclos, se podría decir este número es bastante elevado.

Al disminuir el error cuadrático, se ha logrado que la cantidad Epochs tenga influencia sobre el entrenamiento, además de aumentar el porcentaje de acierto de la red, llegando en torno al 95%. Pese a esto, no se debe pasar por alto que el consumo de recursos es como mínimo un 25% más elevado, incluso llegando en ocasiones al 60%. Este aumento se debe a que, en ocasiones, se agotan todos los ciclos definidos para el entrenamiento antes de llegar a un error cuadrático inferior al 0,01

Para intentar continuar mejorando los resultados, se podría pensar en aumentar el número de Epochs, lo cual no tendría mucho sentido por problema del sobreaprendizaje, comentado anteriormente, al que podría estar sometiéndose a la red. Disminuir el MSE tampoco tendría mucha lógica, ya que como se ha demostrado anteriormente, muchos de los entrenamientos no paraban por ese criterio, si no por el de Epochs, por tanto, si siguiésemos disminuyendo el valor del error cuadrático, este dejaría de tener influencia sobre el entrenamiento. Por tanto, ¿Qué hacer para intentar mejorar los resultados?, probemos incluyendo datos para Cross Validation (CV).

#### 6.2.3.4. INCLUYENDO DATOS PARA CV I

Resumen de los valores definidos para el tercer subconjunto:

- Número de pruebas: 50 | 100 | 150 | 200
- Número de registro de entrenamiento: 150
- Número de registro de test: 40
- Número de registros de CV: 0 - 15
- Número de Epochs: 1.000 | 10.000
- Número de MSE: 0,1



Código	Entrena	Test	Pruebas	Epocs	MSE	CV	Aciertos	Fallos	Total	%Aciertos	%Fallos	Duración
28	150	40	50	1000	0,1	No	1853	147	2000	92,65	7,35	155,09 s
25	150	40	100	1000	0,1	No	3657	343	4000	91,425	8,575	321,81 s
26	150	40	150	1000	0,1	No	5517	483	6000	91,95	8,05	502,75 s
27	150	40	200	1000	0,1	No	7379	621	8000	92,2375	7,7625	701,17 s
32	150	40	50	1000	0,1	15	1902	98	2000	95,1	4,9	191,52 s
29	150	40	100	1000	0,1	15	3837	163	4000	95,925	4,075	391,86 s
30	150	40	150	1000	0,1	15	5718	282	6000	95,3	4,7	599,53 s
31	150	40	200	1000	0,1	15	7581	419	8000	94,7625	5,2375	814,34 s
44	150	40	50	10000	0,1	No	1827	173	2000	91,35	8,65	154,97 s
41	150	40	100	10000	0,1	No	3663	337	4000	91,575	8,425	324,02 s
42	150	40	150	10000	0,1	No	5488	512	6000	91,4666667	8,533333333	501,88 s
43	150	40	200	10000	0,1	No	7314	686	8000	91,425	8,575	697,98 s
48	150	40	50	10000	0,1	15	1896	104	2000	94,8	5,2	190,47 s
45	150	40	100	10000	0,1	15	3826	174	4000	95,65	4,35	422,53 s
46	150	40	150	10000	0,1	15	5704	296	6000	95,0666667	4,933333333	627,05 s
47	150	40	200	10000	0,1	15	7678	322	8000	95,975	4,025	879,05 s

Tabla 62: Resultados de clasificación

Este primer subconjunto de pruebas para el análisis de la influencia del criterio de CV nos va a permitir observar que ocurre cuando se introduce dicha variable en un estudio cuyo criterio de parada era, en la mayoría de las ocasiones, MSE. Como quedó demostrado en el apartado ‘6.2.3.2’, cuando MSE es 0,1, este es el criterio de parada básicamente utilizado por la red.

Lo primero que desvelan los datos de la Tabla 62 es que el porcentaje de acierto de la RNA aumenta entorno al 3,5% de media en todos los ensayos al introducir validación cruzada, por lo que se puede afirmar que, en estas condiciones la introducción de CV aumenta la capacidad de aprendizaje de la red. Los datos también reflejan que las simulaciones ahora tardan alrededor de un 22% más en finalizar. En un principio sería sensato pensar que este aumento en los tiempos se debe al coste computacional que acarrea CV.

¿Cuál es el criterio de parada ahora, CV ó MSE? A partir de los datos anteriores no es fácil aseverar cual sería el que se está dando en la mayoría de las ocasiones, a priori, no existen datos que lo pongan de relieve. Pero, basándonos en las conclusiones anteriores, podríamos inclinarnos a pensar que el criterio MSE es el que se da en la mayoría de las ocasiones, ya que quedó demostrado que 0,1 es un valor muy alto de



MSE para las pruebas utilizadas en este estudio. De todas formas, vamos a ver si fijándonos en la segunda parte del estudio podemos afirmar con datos lo anteriormente sugerido.

### 6.2.3.5. INCLUYENDO DATOS PARA CV II

Resumen de los valores definidos para el tercer subconjunto:

- Número de pruebas: 50 | 100 | 150 | 200
- Número de registro de entrenamiento: 150
- Número de registro de test: 40
- Número de registros de CV: 0 - 15
- Número de Epocs: 1.000 | 10.000
- Número de MSE: 0,01

Código	Entrena	Test	Pruebas	Epocs	MSE	CV	Aciertos	Fallos	Total	%Aciertos	%Fallos	Duración
20	150	40	50	1000	0,01	No	1901	99	2000	95,05	4,95	210,72 s
17	150	40	100	1000	0,01	No	3849	151	4000	96,225	3,775	434,72 s
18	150	40	150	1000	0,01	No	5772	228	6000	96,2	3,8	664,41 s
19	150	40	200	1000	0,01	No	7736	264	8000	96,7	3,3	916,98 s
24	150	40	50	1000	0,01	15	1895	105	2000	94,75	5,25	186,34 s
21	150	40	100	1000	0,01	15	3814	186	4000	95,35	4,65	379,20 s
22	150	40	150	1000	0,01	15	5739	261	6000	95,65	4,35	586,73 s
23	150	40	200	1000	0,01	15	7587	413	8000	94,8375	5,1625	894,50 s
36	150	40	50	10000	0,01	No	1854	146	2000	92,7	7,3	361,42 s
33	150	40	100	10000	0,01	No	3759	241	4000	93,975	6,025	773,64 s
34	150	40	150	10000	0,01	No	5743	257	6000	95,7166667	4,28333333	1209,95 s
35	150	40	200	10000	0,01	No	7644	356	8000	95,55	4,45	1585,88 s
40	150	40	50	10000	0,01	15	1917	83	2000	95,85	4,15	181,42 s
37	150	40	100	10000	0,01	15	3849	151	4000	96,225	3,775	383,33 s
38	150	40	150	10000	0,01	15	5682	318	6000	94,7	5,3	604,75 s
39	150	40	200	10000	0,01	15	7618	382	8000	95,225	4,775	1064,61 s

**Tabla 63: Resultados de clasificación**

En este apartado, se va a tratar de describir cómo actúa la introducción de la variable de validación cruzada en un subconjunto de ensayos cuyo MSE es 0,01. Se



debe recordar que esta configuración de la red hace que con estos datos el criterio de parada sea básicamente MSE.

Si nos fijamos en los ensayos con 10.000 Epocs se puede ver que se ha producido una reducción muy drástica en la cantidad de recursos necesarios, de media, el 45% menos. Por el contrario, hemos obtenido una tasa de aprendizaje similar, en realidad, esta ha aumentado en los ensayos con menos pruebas entorno al 2,5%. La notable disminución de tiempo indica que ahora el criterio de parada más frecuente no es MSE, es CV. ¿Por qué? Si se tiene en cuenta los resultados obtenidos en el apartado '6.2.3.3', es decir, que la red con MSE 0,01 y 10.000 Epocs estaba sobreaprendiendo, es fácil llegar a esta conclusión, ya que una de las principales tareas de CV es tratar de evitar el sobreaprendizaje, por lo que cuando detecta que esto puede estar sucediendo, detiene el proceso de entrenamiento; y tal hecho, es lo que ha acontecido en estos ensayos, lo que ha llevado a que el tiempo consumido disminuya alrededor de un 45%. La misma razón puede explicar que la red haya mejorado sus resultados en los ensayos con un menor número de pruebas, especialmente en el de 50 pruebas, al haber menor número de pruebas, es más fácil que la red las memorice antes, produciéndose el temido sobreaprendizaje.

Cuando el número de Epocs es de 1.000, CV igualmente reduce la cantidad de recursos necesarios, esta vez, en torno a un 10%, obteniendo una tasa de aprendizaje algo inferior, cerca de un 1% de media.

Los datos obtenidos para 1.000 ciclos siguen idéntico razonamiento al expuesto para 10.00. Pero podríamos preguntarnos, si esto es cierto, ¿por qué se ha pasado de un 45%, a un 10% en el ahorro de recursos? Teniendo presente que ahora el número de Epocs es de 1.000, en vez de 10.000, es natural pensar que la tasa de sobreaprendizaje en estos ensayos sea menor, por tanto, el número de veces que se para el proceso de entrenamiento a causa de la validación cruzada también será menor, en consecuencia, el entrenamiento es más largo, y esto, claro está, ha producido que el ahorro de recursos haya pasado de un 45% a un 10%. La disminución cercana al 1% en la tasa de aprendizaje puede deberse en gran medida a que validación cruzada es un criterio de





parada anticipada, lo que hace que en algunas ocasiones, intentado evitar un mal mayor como sería el sobreaprendizaje, detenga el proceso de entrenamiento, provocando esto, que en determinadas ocasiones el porcentaje de aprendizaje de la red no sea el más óptimo. Al igual que con 10.000 ciclos, y por la misma explicación, a menor cantidad de registros por prueba, la introducción de validación cruzada es más recomendable.

Como ha quedado evidenciado en este apartado, cuando CV actúa como criterio de parada habitual, los tiempos necesarios para el entrenamiento de la red disminuyen; esta información permite contestar la pregunta que quedó planteada en el apartado anterior, “¿Cuál es el criterio de parada?”, si nos fijamos que en los ensayos del anterior apartado, al introducir validación cruzada aumenta la necesidad de recursos, lo que indica que CV no está actuando como criterio de parada. El aumento del porcentaje de acierto de la RNA, indica que CV se está evaluando, aunque, como ya ha quedado reflejado, no actúe como criterio de parada.

Después de toda la información aportada, se ha seleccionado como la mejor configuración para el estudio la número 37. Es decir,

- 10.000 Epocs
- 0,01 MSE
- 15 CV
- 100 pruebas

Aunque existe una configuración que obtiene un porcentaje de acierto algo mejor, la número 19, el coste de recursos es casi tres veces mayor. Además, en función de las conclusiones a las que se ha llegado, queda claro que la mejor configuración es aquella en la que intervengan los tres criterios de parada, pero que esté guiada por CV, criterio que ha demostrado ser el que nos da una mejor relación entre el porcentaje de acierto y los recursos necesarios, sin olvidar, que logra impedir el nefasto sobreaprendizaje.



#### 6.2.4. NÚMERO DE REG. EN ENTRENAMIENTO, TEST Y CV.

Una vez fijada la mejor configuración de la RNA para el problema en cuestión, se va a modificar el número de registros que contienen los ficheros de entrenamiento, test y CV en cada una de las pruebas, para comprobar cómo influyen estas variaciones sobre la red

##### 6.2.4.1. MODIFICANDO EL N° DE REG. PARA ENTRENAMIENTO

Resumen de los valores definidos:

- Número de pruebas: 100
- Número de registro de entrenamiento: 50 | 60 | 70 | 80 | 90 | 100 | 110 | 120 | 130 | 140 | 150
- Número de registro de test: 40
- Número de registros de CV: 15
- Número de Epocs: 10.000
- Número de MSE: 0,01

<b>Código</b>	<b>Entrena</b>	<b>Test</b>	<b>Pruebas</b>	<b>Epocs</b>	<b>MSE</b>	<b>CV</b>	<b>Aciertos</b>	<b>Fallos</b>	<b>Total</b>	<b>%Aciertos</b>	<b>%Fallos</b>	<b>Duración</b>
1	50	40	100	10000	0,01	15	3437	563	4000	85,925	14,075	303,80 s.
2	60	40	100	10000	0,01	15	3577	423	4000	89,425	10,575	270,16 s.
3	70	40	100	10000	0,01	15	3680	320	4000	92	8	303,27 s.
4	80	40	100	10000	0,01	15	3660	340	4000	91,5	8,5	332,19 s.
5	90	40	100	10000	0,01	15	3652	348	4000	91,3	8,7	359,13 s.
6	100	40	100	10000	0,01	15	3742	258	4000	93,55	6,45	385,45 s.
7	110	40	100	10000	0,01	15	3757	243	4000	93,925	6,075	443,05 s.
8	120	40	100	10000	0,01	15	3763	237	4000	94,075	5,925	466,00 s.
9	130	40	100	10000	0,01	15	3741	259	4000	93,525	6,475	526,78 s.
10	140	40	100	10000	0,01	15	3675	325	4000	91,875	8,125	535,81 s.
11	150	40	100	10000	0,01	15	3907	93	4000	97,675	2,325	605,23 s.

**Tabla 64: Resultados de clasificación**



Como era de esperar a mayor número de registros destinados al entrenamiento, siempre que estos sean distintos entre sí, más posibilidades existe de que la red aprenda con mayor fiabilidad. Esta es la norma general, aunque no tiene por que cumplirse siempre, como se puede verse en el ensayo nº 10, que pese realizarse con 140 registros, el segundo valor más alto en el conjunto, ha obtenido un pobre resultado. Este detalle nos indica que en determinadas circunstancias, pequeñas variaciones en la disposición de los registros que van en los ficheros que se le pasan a la red durante el proceso, pueden hacer variar su capacidad de aprendizaje notablemente. La experiencia aconseja destinar al menos el 70% de los registros al entrenamiento de la RNA.

#### 6.2.4.2. MODIFICANDO EL Nº DE REG. PARA TEST

Resumen de los valores definidos:

- Número de pruebas: 100
- Número de registro de entrenamiento: 150
- Número de registro de test: 15 | 20 | 25 | 30 | 35 | 40 | 45
- Número de registros de CV: 15
- Número de Epocs: 10.000
- Número de MSE: 0,01

Código	Entrena	Test	Pruebas	Epocs	MSE	CV	Aciertos	Fallos	Total	%Aciertos	%Fallos	Duración
1	150	15	100	10000	0,01	15	1432	68	1500	95,46666667	4,533333333	464,13 s.
2	150	20	100	10000	0,01	15	1904	96	2000	95,2	4,8	490,77 s.
3	150	25	100	10000	0,01	15	2394	106	2500	95,76	4,24	453,20 s.
4	150	30	100	10000	0,01	15	2875	125	3000	95,83333333	4,166666667	437,05 s.
5	150	35	100	10000	0,01	15	3329	171	3500	95,11428571	4,885714286	421,16 s.
6	150	40	100	10000	0,01	15	3880	120	4000	97	3	431,20 s.
7	150	45	100	10000	0,01	15	4341	159	4500	96,46666667	3,533333333	444,06 s.

**Tabla 65: Resultados de clasificación**

La cantidad de datos destinados a test nos va a permitir ver como de satisfactorio ha sido el aprendizaje de la red, por lo tanto, esta cantidad debe ser suficiente, para



evitar obtener conclusiones incorrectas. Es decir, si el número de registros destinados a test es insuficiente, puede ocurrir que obtengamos unos porcentajes de aprendizaje de la red erróneos, tanto por exceso como por defecto. Tampoco sería correcto destinar datos en exceso a test, esto reduciría los destinados a entrenamiento y/o CV, fundamentales el aprendizaje de la red.

Al estar todos los resultados dentro de una horquilla muy pequeña, existen más posibilidades de que los porcentajes se ajusten fielmente a la realidad del aprendizaje de la red. La experiencia aconseja destinar a test cerca del 20% de los datos

#### 6.2.4.3. MODIFICANDO EL Nº DE REG. PARA TEST Y CV

Código	Entrena	Test	Pruebas	Epocs	MSE	CV	Aciertos	Fallos	Total	%Aciertos	%Fallos	Duración
1	150	40	100	10000	0,01	15	3895	105	4000	97,375	2,625	375,61 s.
2	150	35	100	10000	0,01	20	3335	165	3500	95,28571429	4,714285714	394,41 s.
3	150	30	100	10000	0,01	25	2825	175	3000	94,16666667	5,833333333	366,08 s.
4	150	25	100	10000	0,01	30	2421	79	2500	96,84	3,16	369,59 s.
5	150	20	100	10000	0,01	35	1925	75	2000	96,25	3,75	342,70 s.
6	150	15	100	10000	0,01	40	1399	101	1500	93,26666667	6,733333333	390,44 s.

**Tabla 66: Resultados de clasificación**

En esta apartado se va a intentar comprobar si el número de registros destinados a CV es suficiente. Se ha destinado a test un menor número de registros, para poder destinar un mayor número a validación cruzada. La cantidad de registros destinados a entrenamiento no se ha variado, y no es aconsejable que varíe, puesto que dicha cantidad se encuentra en el umbral mínimo aconsejable para el correcto aprendizaje de una RNA.

Según los datos obtenidos, el número inicial de registros destinados a CV parece ser suficiente. Al aumentar los registros para CV no apreciamos cambios sustanciales en los porcentajes, aunque se debe tener en cuenta que según disminuye el valor de test, menos fiables son los porcentajes de acierto obtenidos.



## 6.3. Simulaciones de predicción

### 6.3.1. INTRODUCCIÓN

Nos encontramos ante un problema de predicción siempre que la variable de salida a predecir sea de tipo numérico. Para las simulaciones de predicción realizadas en este proyecto se han tomado como referencia las variables “Masa final”, “ $(Mi-Mf)/Mi$ ”, “Vres” y “DOP”.

Recordemos:

- Masa Final (g): Es la masa del proyectil una vez que ha impactado sobre la placa expresada en gramos. En adelante también referenciaremos a dicha variable como “Mfinal”
- $(Mi-Mf)/Mi$  (%): Es el porcentaje de pérdida de masa del proyectil tras impactar sobre la placa. En adelante referenciaremos a dicha variable mediante “Mi”
- Vres (m/s): Es la velocidad residual del proyectil cuando sí atraviesa la placa completamente, expresada en metros por segundo.
- DOP (mm): Representa la profundidad de perforación en la placa expresada en milímetros, en el caso de que el proyectil no la atraviese completamente.

Al igual que se hizo para los estudios de clasificación, utilizaremos Simulador para realizar unas baterías de ensayos sobre la RNA con distintas configuraciones, y con los resultados obtenidos, evaluar el comportamiento del Perceptrón multicapa con algoritmo de aprendizaje backpropagation para los datos de que disponemos.

Simulador utilizar el coeficiente de correlación para evaluar la relación lineal existente entre los datos reales y los proporcionados por la RNA. A continuación se detalla el proceso.

De cada variable, en cada prueba, se va a calcular el coeficiente de correlación lineal ( $r$ ) que existe entre los valores reales de esa variable y los producidos por la red neuronal. Este coeficiente indica la aproximación de las dos gráficas de cada variable, una con los datos reales y la otra con los datos producidos por la red.



Tomando  $x$  como los valores reales de la variable e  $y$  como los valores producidos por la red de esa variable, el coeficiente de correlación viene dado por:

$$r = \frac{s_{xy}^2}{s_x s_y}$$

La covarianza entre dos variables Var1 y Var2 viene dada por:

$$s_{xy}^2 = \frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})$$

Donde  $x_i$  indica el valor de la variable Var1 para el individuo  $i$ ,  $y_i$  indica el valor de la variable Var2 para el individuo  $i$ ,  $\bar{x}$  la media de Var1 e  $\bar{y}$  la media de Var2.

Las desviaciones típicas muestrales  $s_x$  y  $s_y$  se calculan a partir de las expresiones siguientes:

$$s_x = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^2} \quad s_y = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (y_i - \bar{y})^2}$$

Siendo  $s_x$  la desviación típica de la variable Var1 y  $s_y$  la desviación típica de la variable Var2.

En la interpretación del coeficiente de correlación se debe tener en cuenta que:

- $r = \pm 1$  indica una relación lineal exacta positiva (creciente) o negativa (decreciente),
- $r = 0$  indica la no existencia de relación lineal estocástica, pero no indica independencia de las variables ya que puede existir una relación no lineal,
- valores intermedios de  $r$  ( $0 < r < 1$  ó  $-1 < r < 0$ ) indican la existencia de una relación lineal estocástica, más fuerte cuanto más próximo a  $+1$  (ó  $-1$ ) sea el valor de  $r$ .

Es decir, cuanto más se aproxime a 1 el valor de  $r$  mayor será la relación entre los valores producidos por la red y los reales de una determinada variable.



Estudiando los resultados proporcionados por Simulador podremos seleccionar la configuración más óptima de la red, es decir, aquella que proporcione la mejor relación entre el porcentaje de acierto y los recursos consumidos para conseguir el resultado. En nuestro caso, los recursos empleados lo mediremos en función del tiempo que necesite la red para obtener los resultados.

Por la misma razón que para las simulaciones de clasificación, es decir, el constante e imparable aumento de la capacidad computacional de las máquinas, se le ha asignado una mayor importancia a la fiabilidad de los resultados que a los recursos consumidos.

Dado que la variable Vres sólo obtiene valores cuando el proyectil perfora la red, para su estudio se ha utilizado el subconjunto de los datos iniciales en los que el proyectil atraviesa la placa. Al contrario ocurre con DOP, en la que sólo se han utilizado datos en los que el proyectil no perfora la plancha. Para el resto de variables se han utilizado todos los datos, y se ha mantenido la proporción de 70/30, recordemos que esta proporción indica que el 70 de los datos que incluye cualquier fichero de “prueba, test o CV” son de pruebas que han perforado la placa.

Como ya se ha apuntado, la experiencia nos dice, que para un buen testeo, se necesita una buena base de entrenamiento, por ello, se van a destinar alrededor del 70 por cien del total de los registros de datos de entrada para entrenamiento; para el testeo se destinarán el 70 por ciento, más o menos, de los registros restantes; y, el resto, se destinarán para CV.

Por estudios anteriores [48] [49] [50] [51] , se considera que más de 150 registros de datos para generar los ficheros de entrenamiento, test y CV es una muestra representativa. Por ello, y para que todos los ficheros de los estudios sean lo más heterogéneos posibles, cuando la muestra se ha considerado representativa, no se han utilizado todos los datos disponibles.



Asignación de datos de entrada para cada fichero de entrenamiento, test y CV en cada variable sometida a estudio

- Masa Final y Mi: Existen 220 registros validos.
  - 150 entrenamiento
  - 40 test
  - 15 CV

*Nota: La distribución del número de registros se ha realizado a semejanza de estudios anteriores para los se disponían de 205 pruebas diferentes. Esto permitirá comparar resultados, si fuese necesario, de forma más ecuánime. [48] [49] [50] [51]*

- Vres: Existen 157 registros validos para el estudio de la velocidad residual. Y se utiliza como distribución inicial la siguiente.
  - 110 entrenamiento
  - 25 test
  - 15 CV
- DOP: Existen 63 registros validos para el estudio de la velocidad residual. Y se utiliza como distribución inicial la siguiente.
  - 48 entrenamiento
  - 10 test
  - 5 CV

Las variables que se modifican durante este proceso de configuración de la RNA son las siguientes:

- Número de pruebas
- Número de Epocs
- Número de MSE
- Número de registro de entrenamiento
- Número de registro de test
- Número de registros de CV





Descripción del proceso para la obtención de la configuración de la red:

- Selección del número de pruebas óptimas para una correcta simulación.
- Selección de la configuración y valores óptimos para las variables de Epochs, MSE y CV.
- Selección del número de registros óptimos para el entrenamiento.
- Selección del número de registros óptimos para el test.

A continuación, se va a presentar en detalle el estudio realizado.



### 6.3.2. SIMULACIONES PARA PRUEBAS

Lo primero que vamos a intentar delimitar es el número de pruebas que debe tener cada simulación. Se han realizado 4 simulaciones, la primera con 50 pruebas, y el resto, incrementando este número en 50.

A las variables de Epocs y MSE, se les ha asignado los mismos valores que Neurosolutions da por defecto al crear una red, 1000 Epocs y 0,1 MSE.

#### 6.3.2.1. MFINAL

Resumen de los valores definidos para:

- Número de pruebas: 50 | 100 | 150 | 200
- Número de registro de entrenamiento: 150
- Número de registro de test: 40
- Número de registros de CV: 15
- Número de Epocs: 1.000
- Número de MSE: 0,1

Código	Entrena.	Test	Pruebas	Epocs	MSE	CV	Total	r(Mfinal)	Duración
1	150	40	50	1000	0,1	15	2000	0,991136996	97,53 s.
2	150	40	100	1000	0,1	15	4000	0,993228923	219,25 s.
3	150	40	150	1000	0,1	15	6000	0,9924569	404,22 s.
4	150	40	200	1000	0,1	15	8000	0,990370273	577,39 s.

**Tabla 67: Resultados de predicción**

Con la configuración inicial de la red obtenemos unos resultados muy buenos. La relación entre los datos reales y los proporcionados por Simulador es prácticamente lineal.



### 6.3.2.2. Mi

Resumen de los valores definidos para:

- Número de pruebas: 50 | 100 | 150 | 200
- Número de registro de entrenamiento: 150
- Número de registro de test: 40
- Número de registros de CV: 15
- Número de Epocs: 1.000
- Número de MSE: 0,1

Código	Entrena.	Test	Pruebas	Epocs	MSE	CV	Total	r(Mi)	Duración
1	150	40	50	1000	0,1	15	2000	0,897908477	121,33 s.
2	150	40	100	1000	0,1	15	4000	0,895784056	252,08 s.
3	150	40	150	1000	0,1	15	6000	0,895898045	386,66 s.
4	150	40	200	1000	0,1	15	8000	0,9087493	514,27 s.

**Tabla 68: Resultados de predicción**

Volvemos a obtener unos buenos datos con la configuración inicial aunque sensiblemente inferiores los de la variable masa final.

### 6.3.2.3. VRES

Resumen de los valores definidos para:

- Número de pruebas: 50 | 100 | 150 | 200
- Número de registro de entrenamiento: 110
- Número de registro de test: 25
- Número de registros de CV: 15
- Número de Epocs: 1.000



- Número de MSE: 0,1

Código	Entrena.	Test	Pruebas	EpoCs	MSE	CV	Total	r(Vres)	Duración
1	110	25	50	1000	0,1	15	1250	0,97056254	119,47 s.
2	110	25	100	1000	0,1	15	2500	0,966806248	237,91 s.
3	110	25	150	1000	0,1	15	3750	0,965994923	368,53 s.
4	110	25	200	1000	0,1	15	5000	0,965828219	448,09 s.

**Tabla 69: Resultados de predicción**

Pese a tener un conjunto datos de entrada más reducido que para las dos anteriores variables, la RNA vuelve a obtener unos muy buenos resultados.

#### 6.3.2.4. DOP

Resumen de los valores definidos para:

- Número de pruebas: 50 | 100 | 150 | 200
- Número de registro de entrenamiento: 48
- Número de registro de test: 10
- Número de registros de CV: 5
- Número de EpoCs: 1.000
- Número de MSE: 0,1

Código	Entrena.	Test	Pruebas	EpoCs	MSE	CV	Total	r(Vres)	Duración
1	48	10	50	1000	0,1	5	500	0,897002488	80,45 s.
2	48	10	100	1000	0,1	5	1000	0,841649172	156,50 s.
3	48	10	150	1000	0,1	5	1500	0,877839694	248,42 s.
4	48	10	200	1000	0,1	5	2000	0,880308306	336,86 s.

**Tabla 70: Resultados de predicción**

Pese a disponer de la menor cantidad de datos reales para las predicciones de la variable DOP, Simulador devuelve unos valores muy buenos. La correlación entre los



datos reales y los predichos por la RNA, con la configuración básica y con tan sólo, 63 registros de datos reales, sigue siendo bastante alta.



### 6.3.3. SIMULACIONES PARA EPOCS, MSE Y CV

De la misma forma que en las simulaciones de clasificación, los resultados obtenidos en todos los estudios de predicción al variar el número de pruebas presentadas a la red son despreciables para los 3 primero estudios, y relativamente bajas, alrededor de 5 puntos en el último. Por ello, para la siguiente parte del proceso, volveremos a variar el número de pruebas realizadas en cada simulación, además claro está, del valor de las variables de CV, Epocs y MSE.

Esta parte del proceso se centra en las variables que marcan los criterios de parada del entrenamiento de la RNA. Ajustando bien estas variables, podremos disminuir los recursos necesarios para realizar una simulación, pudiendo llegar a obtener incluso mejores resultados.

#### 6.3.3.1. MFINAL

Resumen de los valores definidos para:

- Número de pruebas: 50 | 100 | 150 | 200
- Número de registro de entrenamiento: 150
- Número de registro de test: 40
- Número de registros de CV: 0 - 15
- Número de Epocs: 1 | 1.000 | 10.000
- Número de MSE: 0,01 | 0,1

<b>Código</b>	<b>Entrena.</b>	<b>Test</b>	<b>Pruebas</b>	<b>Epocs</b>	<b>MSE</b>	<b>CV</b>	<b>Total</b>	<b>r(Mfinal)</b>	<b>Duración</b>
1	150	40	100	1	0,01	No	4000	0,352632119	65,47 s.
2	150	40	150	1	0,01	No	6000	0,35332745	97,61 s.
3	150	40	200	1	0,01	No	8000	0,690794842	145,05 s.
4	150	40	50	1	0,01	No	2000	0,348385569	34,31 s.
5	150	40	100	1	0,01	15	4000	0,367613859	81,30 s.
6	150	40	150	1	0,01	15	6000	0,618875292	120,17 s.



Código	Entrena.	Test	Pruebas	Epocs	MSE	CV	Total	r(Mfinal)	Duración
7	150	40	200	1	0,01	15	8000	0,490130372	188,97 s.
8	150	40	50	1	0,01	15	2000	0,519562648	43,00 s.
9	150	40	100	1	0,1	No	4000	0,517470437	292,75 s.
10	150	40	150	1	0,1	No	6000	0,538621399	257,70 s.
11	150	40	200	1	0,1	No	8000	0,686915181	187,39 s.
12	150	40	50	1	0,1	No	2000	0,470280743	136,31 s.
13	150	40	100	1	0,1	15	4000	0,474418213	250,64 s.
14	150	40	150	1	0,1	15	6000	0,617975575	204,86 s.
15	150	40	200	1	0,1	15	8000	0,580386965	321,16 s.
16	150	40	50	1	0,1	15	2000	0,432298055	73,55 s.
17	150	40	100	1000	0,01	No	4000	0,95096399	99,03 s.
18	150	40	150	1000	0,01	No	6000	0,952064439	170,70 s.
19	150	40	200	1000	0,01	No	8000	0,952371436	239,88 s.
20	150	40	50	1000	0,01	No	2000	0,957125068	54,33 s.
21	150	40	100	1000	0,01	15	4000	0,992045783	433,13 s.
22	150	40	150	1000	0,01	15	6000	0,992864929	708,64 s.
23	150	40	200	1000	0,01	15	8000	0,991322071	869,23 s.
24	150	40	50	1000	0,01	15	2000	0,991868869	150,20 s.
25	150	40	100	1000	0,1	No	4000	0,914487872	97,13 s.
26	150	40	150	1000	0,1	No	6000	0,907168641	145,91 s.
27	150	40	200	1000	0,1	No	8000	0,896199775	224,91 s.
28	150	40	50	1000	0,1	No	2000	0,903546514	48,17 s.
29	150	40	100	1000	0,1	15	4000	0,993228923	219,25 s.
30	150	40	150	1000	0,1	15	6000	0,9924569	404,22 s.
31	150	40	200	1000	0,1	15	8000	0,990370273	577,39 s.
32	150	40	50	1000	0,1	15	2000	0,991136996	97,53 s.
33	150	40	100	10000	0,01	No	4000	0,945549091	79,38 s.
34	150	40	150	10000	0,01	No	6000	0,943646378	147,41 s.
35	150	40	200	10000	0,01	No	8000	0,947070109	228,27 s.
36	150	40	50	10000	0,01	No	2000	0,95735251	39,56 s.
37	150	40	100	10000	0,01	15	4000	0,994521644	930,69 s.
38	150	40	150	10000	0,01	15	6000	0,993821251	1358,64 s.
39	150	40	200	10000	0,01	15	8000	0,990981737	2053,84 s.
40	150	40	50	10000	0,01	15	2000	0,994860583	423,20 s.
41	150	40	100	10000	0,1	No	4000	0,872361224	79,41 s.
42	150	40	150	10000	0,1	No	6000	0,889040432	138,84 s.
43	150	40	200	10000	0,1	No	8000	0,917008653	239,36 s.
44	150	40	50	10000	0,1	No	2000	0,906063389	39,34 s.
45	150	40	100	10000	0,1	15	4000	0,994703732	885,00 s.
46	150	40	150	10000	0,1	15	6000	0,992568715	1222,89 s.
47	150	40	200	10000	0,1	15	8000	0,992464529	1795,50 s.
48	150	40	50	10000	0,1	15	2000	0,993821669	285,02 s.

Tabla 71: Resultados de predicción

Ya que el número de simulaciones para esta parte del estudio es elevado, concretamente 48 simulaciones, y para que al lector le sea más fácil de seguir el por qué



de las conclusiones a las que se ha llegado, a continuación, se presentan aquellos subconjuntos de la tabla anterior que nos permiten mostrar claramente las informaciones extrapolables a esta.

Resumen de los valores definidos para el subconjunto 1:

- Número de pruebas: 50 | 100 | 150 | 200
- Número de registro de entrenamiento: 150
- Número de registro de test: 40
- Número de registros de CV: 0 - 15
- Número de Epocs: 1
- Número de MSE: 0,01 | 0,1

El primer subconjunto que se define claramente en la tabla de resultados, es el de aquellas simulaciones en las que el número de Epocs es 1. Para todos ellos el coeficiente de correlación entre los resultados reales y los proporcionados por la RNA es bastante bajo, siempre por debajo del 0,7.

Código	Entrena.	Test	Pruebas	Epocs	MSE	CV	Total	r(Mfinal)	Duración
3	150	40	200	1	0,01	No	8000	0,690794842	145,05 s.
11	150	40	200	1	0,1	No	8000	0,686915181	187,39 s.
6	150	40	150	1	0,01	15	6000	0,618875292	120,17 s.
14	150	40	150	1	0,1	15	6000	0,617975575	204,86 s.
15	150	40	200	1	0,1	15	8000	0,580386965	321,16 s.
10	150	40	150	1	0,1	No	6000	0,538621399	257,70 s.
8	150	40	50	1	0,01	15	2000	0,519562648	43,00 s.
9	150	40	100	1	0,1	No	4000	0,517470437	292,75 s.
7	150	40	200	1	0,01	15	8000	0,490130372	188,97 s.
13	150	40	100	1	0,1	15	4000	0,474418213	250,64 s.
12	150	40	50	1	0,1	No	2000	0,470280743	136,31 s.
16	150	40	50	1	0,1	15	2000	0,432298055	73,55 s.
5	150	40	100	1	0,01	15	4000	0,367613859	81,30 s.
2	150	40	150	1	0,01	No	6000	0,35332745	97,61 s.
1	150	40	100	1	0,01	No	4000	0,352632119	65,47 s.
4	150	40	50	1	0,01	No	2000	0,348385569	34,31 s.

Tabla 72: Resultados de predicción





Al igual que ocurría para el estudio de clasificación, y tal y como era de esperar, los valores más bajos de capacidad de predicción de la RNA se obtienen cuando el número de Epochs es 1, independientemente del valor que tengan el resto de los criterios de parada. Esta observación, es más que comprensible, ya que al realizar un único ciclo, a la red sólo se le presentan los entrenamientos una vez, por lo que no puede ajustar los pesos de sus neuronas, impidiendo por tanto, alcanzar el mínimo global del error, punto en el que la red obtiene su mayor grado de fiabilidad.

Resumen de los valores definidos para el subconjunto 2:

- Número de pruebas: 50 | 100 | 150 | 200
- Número de registro de entrenamiento: 150
- Número de registro de test: 40
- Número de registros de CV: 0
- Número de Epochs: 1000 | 10000
- Número de MSE: 0,1

En el siguiente subconjunto de datos se encuentran aquellas simulaciones cuyos valores de correlación se oscilan entre el 0,87 y el 0,91.

<b>Código</b>	<b>Entrena.</b>	<b>Test</b>	<b>Pruebas</b>	<b>Epochs</b>	<b>MSE</b>	<b>CV</b>	<b>Total</b>	<b>r(Mfinal)</b>	<b>Duración</b>
43	150	40	200	10000	0,1	No	8000	0,917008653	239,36 s.
25	150	40	100	1000	0,1	No	4000	0,914487872	97,13 s.
26	150	40	150	1000	0,1	No	6000	0,907168641	145,91 s.
44	150	40	50	10000	0,1	No	2000	0,906063389	39,34 s.
28	150	40	50	1000	0,1	No	2000	0,903546514	48,17 s.
27	150	40	200	1000	0,1	No	8000	0,896199775	224,91 s.
42	150	40	150	10000	0,1	No	6000	0,889040432	138,84 s.
41	150	40	100	10000	0,1	No	4000	0,872361224	79,41 s.

**Tabla 73: Resultados de predicción**

Simplemente aumentando el número de Epochs los resultados mejoran muy notablemente. En este bloque se obtienen resultados muy próximos independientemente



de número de ciclos máximos establecidos, esto puede indicar que el criterio de parada que prevalece es el de MSE. Al centrarse en la duración de las simulaciones, los resultados corroboran las sospechas, ya que la duración de un proceso con las mismas características pero con distinto número de Epocs es muy similar, e incluso en ocasiones, superior en las simulaciones con menor número de Epocs.

### Resumen de los valores definidos para el subconjunto 3:

- Número de pruebas: 50 | 100 | 150 | 200
- Número de registro de entrenamiento: 150
- Número de registro de test: 40
- Número de registros de CV: 0
- Número de Epocs: 1000 | 10000
- Número de MSE: 0,01

Para el siguiente bloque de datos los valores de la correlación oscilan entre el 0,94 y el 0,95. Son unos datos excelentes, esto supone una mejor mínima superior a 4 puntos para todas las simulaciones respecto al subconjunto anterior.

<b>Código</b>	<b>Entrena.</b>	<b>Test</b>	<b>Pruebas</b>	<b>Epocs</b>	<b>MSE</b>	<b>CV</b>	<b>Total</b>	<b>r(Mfinal)</b>	<b>Duración</b>
36	150	40	50	10000	0,01	No	2000	0,95735251	39,56 s.
20	150	40	50	1000	0,01	No	2000	0,957125068	54,33 s.
19	150	40	200	1000	0,01	No	8000	0,952371436	239,88 s.
18	150	40	150	1000	0,01	No	6000	0,952064439	170,70 s.
17	150	40	100	1000	0,01	No	4000	0,95096399	99,03 s.
35	150	40	200	10000	0,01	No	8000	0,947070109	228,27 s.
33	150	40	100	10000	0,01	No	4000	0,945549091	79,38 s.
34	150	40	150	10000	0,01	No	6000	0,943646378	147,41 s.

**Tabla 74: Resultados de predicción**

Esta mejora se debe al aumento de la precisión de la variable MSE. El valor ahora para todas las simulaciones expuestas es de 0,01, mientras que el resto criterios toman idénticos valores que en el bloque anterior.



Al retrasar este criterio de parada, y no haber indicios de sobreaprendizaje, la red distribuye mejor los pesos de sus neuronas, lo que la permite predecir con mayor fiabilidad.

El criterio de parada dominante sigue siendo el error cuadrático, puesto que al igual que en el bloque anterior (Tabla 73), la duración de los procesos con idénticas características pero con distinto número de Epocs sigue siendo muy similar, e incluso en ocasiones, superior para las simulaciones con menor número de Epocs.

Resumen de los valores definidos para el subconjunto 4:

- Número de pruebas: 50 | 100 | 150 | 200
- Número de registro de entrenamiento: 150
- Número de registro de test: 40
- Número de registros de CV: 15
- Número de Epocs: 1000 | 10000
- Número de MSE: 0,1 | 0,01

En este último bloque obtenemos unos resultados de correlación excelentes, todos los valores para r aumentan al menos 4 puntos. Los valores de coeficiente varían entre 0,990 y 0,994, todos ellos muy próximos a 1 lo que indica una relación prácticamente directa los datos reales y los predichos por la RNA

<b>Código</b>	<b>Entrena.</b>	<b>Test</b>	<b>Pruebas</b>	<b>Epocs</b>	<b>MSE</b>	<b>CV</b>	<b>Total</b>	<b>r(Mfinal)</b>	<b>Duración</b>
40	150	40	50	10000	0,01	15	2000	0,994860583	423,20 s
45	150	40	100	10000	0,1	15	4000	0,994703732	885,00 s.
37	150	40	100	10000	0,01	15	4000	0,994521644	930,69 s.
48	150	40	50	10000	0,1	15	2000	0,993821669	285,02 s.
38	150	40	150	10000	0,01	15	6000	0,993821251	1358,64 s.
29	150	40	100	1000	0,1	15	4000	0,993228923	219,25 s.
22	150	40	150	1000	0,01	15	6000	0,992864929	708,64 s.
46	150	40	150	10000	0,1	15	6000	0,992568715	1222,89 s.



Código	Entrena.	Test	Pruebas	Epocs	MSE	CV	Total	$r(M_{final})$	Duración
47	150	40	200	10000	0,1	15	8000	0,992464529	1795,50 s.
30	150	40	150	1000	0,1	15	6000	0,9924569	404,22 s.
21	150	40	100	1000	0,01	15	4000	0,992045783	433,13 s.
24	150	40	50	1000	0,01	15	2000	0,991868869	150,20 s.
23	150	40	200	1000	0,01	15	8000	0,991322071	869,23 s.
32	150	40	50	1000	0,1	15	2000	0,991136996	97,53 s.
39	150	40	200	10000	0,01	15	8000	0,990981737	2053,84 s.
31	150	40	200	1000	0,1	15	8000	0,990370273	577,39 s.

Tabla 75: Resultados de predicción

Este subconjunto lo componen aquellas simulaciones en los que se ha incluido como criterio de parada la validación cruzada.

Lo primero que destaca es que una vez incluido este criterio, los valores asignados para el resto de criterios de parada dejan de ser determinantes para el valor  $r$  que proporciona Simulador; sin importar el resto de valores que toman los demás criterios de parada todas las simulaciones aumentan de media 5 centésimas el valor de la correlación, esta es una mejora más que significativa dado el margen de mejora posible. La evaluación de CV durante el entrenamiento permite a la RNA mejorar su capacidad de aprendizaje y permite controlar el sobreaprendizaje

Si se analizan los tiempos de las simulaciones se puede ver con claridad que aquellas con 1000 Epocs son las más rápidas, y particularmente, las que tiene un MSE igual a 0,1; simulaciones con las mismas características pero con distinto número de Epocs tienen un coste, al menos, del doble de tiempo, llegando incluso en ocasiones a cuadruplicarlo. Esto indica que en ellas el número de ciclos es el principal criterio de parada, cediendo es si, algo de protagonismo a MSE, cuando este tiene el valor de 0,1.

Para las simulaciones con 10000 Epocs, mismas características, pero con distinto valor para MSE, los consumos no tienen unas diferencias muy pronunciadas, suelen rondar el 10 por cien, por lo que todo hace pensar que en estas simulaciones el número de Epocs y CV, son los criterios de parada que se dan en la mayoría de las ocasiones.



Es reseñable que el consumo de recursos en las simulaciones con CV es mucho mayor, aunque más que justificable, ateniéndose a los resultados obtenidos. Pero llama la atención ver que esto es totalmente distinto a lo que ocurría en las simulaciones de clasificación en las que al introducir la validación cruzada la duración de los procesos disminuía de forma notable. ¿Cómo se explica esto? ¿Existe alguna diferencia en CV para las simulaciones de predicción y las de clasificación?

En principio, CV supone un consumo extra de recursos, ya que se debe evaluar un nuevo criterio de parada durante el entrenamiento. Una de las principales características de la validación cruzadas es evitar que la RNA sufra sobreaprendizaje, esto es, CV es capaz de detectarlo y detener el entrenamiento, si esperar a que se cumplan cualquiera de los restantes criterios, lo que provoca un entrenamiento más corto, y por tanto, una disminución del consumo de recursos.

A partir de las premisas anteriores se pueden aclarar todas las dudas planteadas.

- En el caso del estudio de clasificación quedó claro que al introducir CV además de mejorar los porcentajes, lo que se producía era un notable descenso en los tiempos de proceso, y que esto se debía CV acortaba el entrenamiento para evitar el sobreaprendizaje. Sin embargo, en el estudio de predicción de Masa Final, se ha producido un aumento de costes, lo que indica que no existía sobre aprendizaje en las simulaciones sin CV. Por tanto, al introducir la validación cruzada el tiempo de entrenamiento no disminuye, sino que aumenta, al evaluar CV.
- No, CV funciona de la misma forma para simulaciones de clasificación y predicción.

Dados los excelentes resultados que obtiene simulador para la predicción de la masa final todo hace indicar que la configuración más óptima debe ser una rápida, no parece tener mucho sentido alargar el proceso de entrenamiento para obtener unos



resultados muy similares.; y más, cuando cualquier simulación ofrece una relación prácticamente lineal con los resultados reales.

Partiendo de la anterior premisa, se ha seleccionado como la configuración más óptima la de la simulación 29.

- 1.000 Epocs
- 0,1 MSE
- 15 CV
- 100 pruebas

Esta configuración es la que guarda mejor relación entre el coeficiente de correlación y la cantidad de recursos consumidos para lograrlo.

Pese a existir configuraciones capaces de lograr un  $r$  mayor, la mejora lograda no es relevante, y aún lo es menos, si se compara la cantidad de tiempo que se ha necesitado para lograrla. Concretando, en ningún caso la mejora máxima de  $r$  que se ha obtenido es superior a 1,5 milésimas, mientras que el incremento en el consumo de recursos oscila entre el 25 y el 600 por cien. Además

Todas las simulaciones con mejor coeficiente tienen 10000 ciclos, por 1000 de la seleccionada, y como se demostró, al no existir sobreaprendizaje, el número de Epocs es un criterio de parada básico en el estudio de esta variable, por eso, el tiempo de entrenamiento aumenta tanto, aunque no se obtengan unos resultados significativamente mejores.



### 6.3.3.2. Mi

Resumen de los valores definidos para:

- Número de pruebas: 50 | 100 | 150 | 200
- Número de registro de entrenamiento: 150
- Número de registro de test: 40
- Número de registros de CV: 0 - 15
- Número de Epocs: 1 | 1.000 | 10.000
- Número de MSE: 0,01 | 0,1

Código	Entrena.	Test	Pruebas	Epocs	MSE	CV	Total	r(Mi)	Duración
1	150	40	100	1	0,01	15	4000	0,522375621	131,25s.
2	150	40	150	1	0,01	15	6000	0,210357367	202,64s.
3	150	40	200	1	0,01	15	8000	0,555828658	276,16s.
4	150	40	50	1	0,01	15	2000	0,246113201	66,03s.
5	150	40	100	1	0,01	No	4000	0,576223579	69,30s.
6	150	40	150	1	0,01	No	6000	0,415035776	128,69s.
7	150	40	200	1	0,01	No	8000	0,38264209	194,06s.
8	150	40	50	1	0,01	No	2000	0,037501681	35,20s.
9	150	40	100	1	0,1	15	4000	0,573291253	142,20s.
10	150	40	150	1	0,1	15	6000	0,432592766	216,58s.
11	150	40	200	1	0,1	15	8000	0,438697512	291,20s.
12	150	40	50	1	0,1	15	2000	0,279528405	70,75s.
13	150	40	100	1	0,1	No	4000	0,541341873	105,34s.
14	150	40	150	1	0,1	No	6000	0,467302441	161,06s.
15	150	40	200	1	0,1	No	8000	0,593941676	220,53s.
16	150	40	50	1	0,1	No	2000	0,491549599	53,20s.
17	150	40	100	1000	0,01	15	4000	0,915933645	241,91s.
18	150	40	150	1000	0,01	15	6000	0,894427451	367,16s.
19	150	40	200	1000	0,01	15	8000	0,903776729	470,97s.
20	150	40	50	1000	0,01	15	2000	0,913562334	128,16s.
21	150	40	100	1000	0,01	No	4000	0,905390744	338,86s.
22	150	40	150	1000	0,01	No	6000	0,901305091	500,78s.
23	150	40	200	1000	0,01	No	8000	0,904510996	675,08s.
24	150	40	50	1000	0,01	No	2000	0,890412297	170,64s.
25	150	40	100	1000	0,1	15	4000	0,895784056	252,08s.
26	150	40	150	1000	0,1	15	6000	0,895898045	386,66s.
27	150	40	200	1000	0,1	15	8000	0,9087493	514,27s.



Código	Entrena.	Test	Pruebas	Epocs	MSE	CV	Total	r(Mi)	Duración
28	150	40	50	1000	0,1	15	2000	0,897908477	121,33s.
29	150	40	100	1000	0,1	No	4000	0,803581702	140,41s.
30	150	40	150	1000	0,1	No	6000	0,786413122	212,19s.
31	150	40	200	1000	0,1	No	8000	0,806075199	292,88s.
32	150	40	50	1000	0,1	No	2000	0,794020015	70,39s.
33	150	40	100	10000	0,01	15	4000	0,892749483	325,19s.
34	150	40	150	10000	0,01	15	6000	0,882617597	489,53s.
35	150	40	200	10000	0,01	15	8000	0,894874367	607,80s.
36	150	40	50	10000	0,01	15	2000	0,891539126	168,89s.
37	150	40	100	10000	0,01	No	4000	0,887913369	1318,20s.
38	150	40	150	10000	0,01	No	6000	0,908459306	1963,52s.
39	150	40	200	10000	0,01	No	8000	0,896386682	2425,20s.
40	150	40	50	10000	0,01	No	2000	0,880185114	740,56s.
41	150	40	100	10000	0,1	15	4000	0,910565684	321,78s.
42	150	40	150	10000	0,1	15	6000	0,889318085	458,97s.
43	150	40	200	10000	0,1	15	8000	0,901988673	614,66s.
44	150	40	50	10000	0,1	15	2000	0,887516684	152,89s.
45	150	40	100	10000	0,1	No	4000	0,811082116	184,03s.
46	150	40	150	10000	0,1	No	6000	0,771212212	280,84s.
47	150	40	200	10000	0,1	No	8000	0,792454058	385,00s.
48	150	40	50	10000	0,1	No	2000	0,789185617	90,64s.

Tabla 76: Resultados de predicción

Al igual en los estudios anteriores, el número de simulaciones para esta parte del estudio es elevado, concretamente 48 simulaciones, para que al lector le sea más fácil de seguir el por qué de las conclusiones a las que se ha llegado, a continuación, se presentan aquellos subconjuntos de la tabla anterior que nos permiten mostrar claramente las informaciones extrapolables a esta.

Resumen de los valores definidos para el subconjunto 1:

- Número de pruebas: 50 | 100 | 150 | 200
- Número de registro de entrenamiento: 150
- Número de registro de test: 40
- Número de registros de CV: 0 - 15
- Número de Epocs: 1
- Número de MSE: 0,01 | 0,1





De la misma forma que en el estudio anterior, el primer subconjunto que se define claramente en la tabla de resultados es el de aquellas simulaciones en las que el número de Epochs es 1. Para todos ellos el coeficiente de correlación entre los resultados reales y los proporcionados por la RNA es bastante bajo, siempre inferiores a 0,6.

Código	Entrena.	Test	Pruebas	Epochs	MSE	CV	Total	r(Mi)	Duración
15	150	40	200	1	0,1	No	8000	0,59394168	220,53 s.
5	150	40	100	1	0,01	No	4000	0,57622358	69,30 s.
9	150	40	100	1	0,1	15	4000	0,57329125	142,20 s.
3	150	40	200	1	0,01	15	8000	0,55582866	276,16 s.
13	150	40	100	1	0,1	No	4000	0,54134187	105,34 s.
1	150	40	100	1	0,01	15	4000	0,52237562	131,25 s.
16	150	40	50	1	0,1	No	2000	0,4915496	53,20 s.
14	150	40	150	1	0,1	No	6000	0,46730244	161,06 s.
11	150	40	200	1	0,1	15	8000	0,43869751	291,20 s.
10	150	40	150	1	0,1	15	6000	0,43259277	216,58 s.
6	150	40	150	1	0,01	No	6000	0,41503578	128,69 s.
7	150	40	200	1	0,01	No	8000	0,38264209	194,06 s.
12	150	40	50	1	0,1	15	2000	0,2795284	70,75 s.
4	150	40	50	1	0,01	15	2000	0,2461132	66,03 s.
2	150	40	150	1	0,01	15	6000	0,21035737	202,64 s.
8	150	40	50	1	0,01	No	2000	0,03750168	35,20 s.

Tabla 77: Resultados de predicción

Al igual que en los estudios anteriores, y tal y como era de esperar, los valores más bajos de capacidad de predicción de la RNA se obtienen cuando el número de Epochs es 1, independientemente del valor que tengan el resto de los criterios de parada. Como ya se explicó, esta observación es más que comprensible, al realizar un único ciclo a la red sólo se le presentan los entrenamientos una vez, por lo que no puede ajustar los pesos de sus neuronas, impidiendo por tanto, alcanzar el mínimo global del error, punto en el que la red obtiene su mayor grado de fiabilidad.

Resumen de los valores definidos para el subconjunto 2:

- Número de pruebas: 50 | 100 | 150 | 200
- Número de registro de entrenamiento: 150
- Número de registro de test: 40



- Número de registros de CV: 0
- Número de Epocs: 1000 | 10000
- Número de MSE: 0,1

En el siguiente subconjunto de datos se encuentran aquellas simulaciones cuyos valores de correlación se oscilan entre el 0,77 y el 0,81.

Código	Entrena.	Test	Pruebas	Epocs	MSE	CV	Total	r(Mi)	Duración
45	150	40	100	10000	0,1	No	4000	0,81108212	184,03s.
31	150	40	200	1000	0,1	No	8000	0,8060752	292,88s.
29	150	40	100	1000	0,1	No	4000	0,8035817	140,41s.
32	150	40	50	1000	0,1	No	2000	0,79402001	70,39s.
47	150	40	200	10000	0,1	No	8000	0,79245406	385,00s.
48	150	40	50	10000	0,1	No	2000	0,78918562	90,64s.
30	150	40	150	1000	0,1	No	6000	0,78641312	212,19s.
46	150	40	150	10000	0,1	No	6000	0,77121221	280,84s.

**Tabla 78: Resultados de predicción**

Simplemente aumentando el número de Epocs los resultados mejoran muy notablemente.

En este bloque se obtienen resultados muy próximos independientemente de número de ciclos máximos establecidos, al igual que para el estudio anterior, esto hace pensar que el criterio de parada que se da en la mayoría de las ocasiones es el de MSE. Sin embargo, esto no es del todo cierto, a diferencia del estudio Mfinal, en este siempre existe una diferencia notable entre la cantidad de recursos consumidos en las simulaciones con 1000 y 10000 ciclos, en concreto, 25 por cien mayores en estos últimos. Esto indica que en el caso de las simulaciones con 1000 ciclos este criterio tiene un peso no desdeñable. Para las simulaciones con 10000 Epocs, los datos no nos revelan directamente cual es el peso de cada uno de los criterios de parada, pero en base a los estudios anteriores sabemos que simulaciones sin CV y un MSE de 0,1 suelen conllevar un entrenamiento corto, es un criterio fácilmente alcanzable; y por el contrario, 10000 Epocs sin CV se traduce en un entrenamiento largo, por ello, todo hace pensar que el criterio de parada básico para estas simulaciones es el MSE.



### Resumen de los valores definidos para el subconjunto 3:

- Número de pruebas: 50 | 100 | 150 | 200
- Número de registro de entrenamiento: 150
- Número de registro de test: 40
- Número de registros de CV: 0 -15
- Número de Epocs: 1000 | 10000
- Número de MSE: 0,01

En este último bloque obtenemos unos resultados de correlación bastante buenos, todos los valores para r aumentan al menos 12 puntos, lo que supone una mejora muy importante. Los valores de coeficiente varían entre 0,89 y 0,91, todos ellos muy próximos entre sí, y relativamente próximos a 1.

Código	Entrena.	Test	Pruebas	Epocs	MSE	CV	Total	r(Mi)	Duración
17	150	40	100	1000	0,01	15	4000	0,915933645	241,91s.
20	150	40	50	1000	0,01	15	2000	0,913562334	128,16s.
41	150	40	100	10000	0,1	15	4000	0,910565684	321,78s.
27	150	40	200	1000	0,1	15	8000	0,9087493	514,27s.
38	150	40	150	10000	0,01	No	6000	0,908459306	1963,52s.
21	150	40	100	1000	0,01	No	4000	0,905390744	338,86s.
23	150	40	200	1000	0,01	No	8000	0,904510996	675,08s.
19	150	40	200	1000	0,01	15	8000	0,903776729	470,97s.
43	150	40	200	10000	0,1	15	8000	0,901988673	614,66s.
22	150	40	150	1000	0,01	No	6000	0,901305091	500,78s.
28	150	40	50	1000	0,1	15	2000	0,897908477	121,33s.
39	150	40	200	10000	0,01	No	8000	0,896386682	2425,20s.
26	150	40	150	1000	0,1	15	6000	0,895898045	386,66s.
25	150	40	100	1000	0,1	15	4000	0,895784056	252,08s.
35	150	40	200	10000	0,01	15	8000	0,894874367	607,80s.
18	150	40	150	1000	0,01	15	6000	0,894427451	367,16s.
33	150	40	100	10000	0,01	15	4000	0,892749483	325,19s.
36	150	40	50	10000	0,01	15	2000	0,891539126	168,89s.
24	150	40	50	1000	0,01	No	2000	0,890412297	170,64s.
42	150	40	150	10000	0,1	15	6000	0,889318085	458,97s.
37	150	40	100	10000	0,01	No	4000	0,887913369	1318,20s.
44	150	40	50	10000	0,1	15	2000	0,887516684	152,89s.
34	150	40	150	10000	0,01	15	6000	0,882617597	489,53s.
40	150	40	50	10000	0,01	No	2000	0,880185114	740,56s.

Tabla 79: Resultados de predicción



Llama la atención de este último subconjunto que esté formado por las simulaciones con CV, como era de esperar, más simulaciones sin CV. Esto nunca había ocurrido, en los estudios anteriores siempre existía una diferencia representativa en la capacidad de predicción o clasificación entre simulaciones con y sin validación cruzada. Todas las simulaciones sin CV que está dentro de este bloque de las mejores simulaciones son aquellas que tiene un error cuadrático de 0,01. ¿Por qué?

Un error cuadrático de 0,01 sin CV garantiza un entrenamiento largo, es más, por lo datos se puede afirmar que en muchas ocasiones se agotarán el número de ciclos antes de llegar al MSE establecido, sobre todo en las simulaciones con 1000 Epocs. Con este tipo de entrenamiento se pueden obtener muy buenos resultados siempre y cuando la RNA no sobreaprenda, ya que una vez que esto ocurre la capacidad predicción de la red disminuye por cada nuevo ciclo de entrenamiento. En función de los datos obtenidos mediante Simulador todo parece indicar que el grado de sobreaprendizaje que puede estar sufriendo las red con este tipo de entrenamientos no de ser muy alto, pues los resultados de  $r$  son tan buenos como cuando se incluye el criterio de validación cruzada.

En las simulaciones con CV vemos que los consumos de recursos disminuyen drásticamente, esto quiere decir, que cuando este criterio entra en juego se muestra como el principal criterio de parada. Esto se certifica al observar que todas las simulaciones con CV y mismo número de pruebas, obtienen unos tiempos similares independientemente de los valores de MSE y Epocs.

Para obtener los mejores resultados en la predicción de  $M_i$  existen dos posibilidades, un entrenamiento largo o un entrenamiento controlado por CV. En aras de mejorar el consumo de recursos se debe seleccionar una configuración que proporcione un entrenamiento de duración baja/media regulado por CV. Esto garantiza la mejor relación entre el valor de  $r$  y los recursos utilizados.



Partiendo de la anterior premisa, se ha seleccionado como la configuración más óptima la de la simulación 20.

- 1.000 Epocs
- 0,01 MSE
- 15 CV
- 50 pruebas

Esta configuración es la que guarda mejor relación entre el coeficiente de correlación y la cantidad de recursos consumidos para lograrlo. La duración del entrenamiento está por debajo de la media y obtiene el segundo mejor coeficiente de correlación. El entrenamiento puede parecer un poco corto, pero para la predicción de esta variable la RNA se comporta de igual o mejor manera que los entrenamientos más largos. Otra muy buena opción, si se deseara un entrenamiento algo más largo, sería la configuración de la simulación 20. Es el mismo tipo de entrenamiento pero con más pruebas; obtiene un mejor coeficiente pero aumentan los recursos necesarios al doble, aunque sigue siendo eficiente si la comparamos con el resto de las simulaciones.

#### 6.3.3.3. VRES

Resumen de los valores definidos para:

- Número de pruebas: 50 | 100 | 150 | 200
- Número de registro de entrenamiento: 110
- Número de registro de test: 25
- Número de registros de CV: 0 - 15
- Número de Epocs: 1 | 1.000 | 10.000
- Número de MSE: 0,01 | 0,1



Código	Entrena.	Test	Pruebas	Epocs	MSE	CV	Total	r(Mi)	Duración
1	110	25	100	1	0,01	No	2500	0,652925718	61,66 s.
2	110	25	150	1	0,01	No	3750	0,528939215	95,59 s.
3	110	25	200	1	0,01	No	5000	0,537038136	154,14 s.
4	110	25	50	1	0,01	No	1250	0,39396188	30,41 s.
5	110	25	100	1	0,01	15	2500	0,498152667	119,86 s.
6	110	25	150	1	0,01	15	3750	0,716298738	179,30 s.
7	110	25	200	1	0,01	15	5000	0,547053276	246,97 s.
8	110	25	50	1	0,01	15	1250	0,727712244	60,81 s.
9	110	25	100	1	0,1	No	2500	0,607551747	89,17 s.
10	110	25	150	1	0,1	No	3750	0,775980646	141,84 s.
11	110	25	200	1	0,1	No	5000	0,504491208	206,88 s.
12	110	25	50	1	0,1	No	1250	0,746323545	43,86 s.
13	110	25	100	1	0,1	15	2500	0,593825018	118,77 s.
14	110	25	150	1	0,1	15	3750	0,671769936	181,17 s.
15	110	25	200	1	0,1	15	5000	0,67637883	238,67 s.
16	110	25	50	1	0,1	15	1250	0,629253802	60,69 s.
17	110	25	100	1000	0,01	No	2500	0,939013145	131,33 s.
18	110	25	150	1000	0,01	No	3750	0,936258524	201,20 s.
19	110	25	200	1000	0,01	No	5000	0,935367959	266,19 s.
20	110	25	50	1000	0,01	No	1250	0,946705453	69,08 s.
21	110	25	100	1000	0,01	15	2500	0,967952083	236,55 s.
22	110	25	150	1000	0,01	15	3750	0,963716501	359,80 s.
23	110	25	200	1000	0,01	15	5000	0,969729922	467,56 s.
24	110	25	50	1000	0,01	15	1250	0,969252972	111,58 s.
25	110	25	100	1000	0,1	No	2500	0,905129587	117,78 s.
26	110	25	150	1000	0,1	No	3750	0,926250676	183,48 s.
27	110	25	200	1000	0,1	No	5000	0,931926527	241,86 s.
28	110	25	50	1000	0,1	No	1250	0,924091666	56,94 s.
29	110	25	100	1000	0,1	15	2500	0,966806248	237,91 s.
30	110	25	150	1000	0,1	15	3750	0,965994923	368,53 s.
31	110	25	200	1000	0,1	15	5000	0,965828219	448,09 s.
32	110	25	50	1000	0,1	15	1250	0,97056254	119,47 s.
33	110	25	100	10000	0,01	No	2500	0,934891309	150,89 s.
34	110	25	150	10000	0,01	No	3750	0,936656172	227,88 s.
35	110	25	200	10000	0,01	No	5000	0,940143605	311,70 s.
36	110	25	50	10000	0,01	No	1250	0,94918083	78,31 s.
37	110	25	100	10000	0,01	15	2500	0,966970416	360,97 s.
38	110	25	150	10000	0,01	15	3750	0,964667859	607,91 s.
39	110	25	200	10000	0,01	15	5000	0,968572164	657,27 s.
40	110	25	50	10000	0,01	15	1250	0,965548574	188,31 s.
41	110	25	100	10000	0,1	No	2500	0,909580728	145,95 s.
42	110	25	150	10000	0,1	No	3750	0,913310274	209,47 s.
43	110	25	200	10000	0,1	No	5000	0,917065534	280,05 s.
44	110	25	50	10000	0,1	No	1250	0,922622933	71,52 s.
45	110	25	100	10000	0,1	15	2500	0,960449184	442,41 s.
46	110	25	150	10000	0,1	15	3750	0,966811954	584,22 s.
47	110	25	200	10000	0,1	15	5000	0,965554068	798,09 s.
48	110	25	50	10000	0,1	15	1250	0,972016817	157,77 s.

**Tabla 80: Resultados de predicción**



Al igual en los estudios anteriores, el número de simulaciones para esta parte del estudio es elevado, concretamente 48 simulaciones, para que al lector le sea más fácil de seguir el por qué de las conclusiones a las que se ha llegado, a continuación, se presentan aquellos subconjuntos de la tabla anterior que nos permiten mostrar claramente las informaciones extrapolables a esta.

Resumen de los valores definidos para el subconjunto 1:

- Número de pruebas: 50 | 100 | 150 | 200
- Número de registro de entrenamiento: 110
- Número de registro de test: 25
- Número de registros de CV: 0 - 15
- Número de Epocs: 1
- Número de MSE: 0,01 | 0,1

De la misma forma que en el resto de los estudios, el primer subconjunto que se define claramente en la tabla de resultados es el de aquellas simulaciones en las que el número de Epocs es 1. Para todos ellos el coeficiente de correlación entre los resultados reales y los proporcionados por la RNA es siempre inferior a 0,78. Pese a todo son unos valores bastante altos en comparación a los obtenidos para el resto de variables.

<i>Código</i>	<i>Entrena.</i>	<i>Test</i>	<i>Pruebas</i>	<i>Epocs</i>	<i>MSE</i>	<i>CV</i>	<i>Total</i>	<i>r(Mi)</i>	<i>Duración</i>
10	110	25	150	1	0,1	No	3750	0,77598065	141,84 s.
12	110	25	50	1	0,1	No	1250	0,74632355	43,86 s.
8	110	25	50	1	0,01	15	1250	0,72771224	60,81 s.
6	110	25	150	1	0,01	15	3750	0,71629874	179,30 s.
15	110	25	200	1	0,1	15	5000	0,67637883	238,67 s.
14	110	25	150	1	0,1	15	3750	0,67176994	181,17 s.
1	110	25	100	1	0,01	No	2500	0,65292572	61,66 s.
16	110	25	50	1	0,1	15	1250	0,6292538	60,69 s.
9	110	25	100	1	0,1	No	2500	0,60755175	89,17 s.
13	110	25	100	1	0,1	15	2500	0,59382502	118,77 s.
7	110	25	200	1	0,01	15	5000	0,54705328	246,97 s.
3	110	25	200	1	0,01	No	5000	0,53703814	154,14 s.
2	110	25	150	1	0,01	No	3750	0,52893921	95,59 s.
11	110	25	200	1	0,1	No	5000	0,50449121	206,88 s.



Código	Entrena.	Test	Pruebas	Epocs	MSE	CV	Total	r(Mi)	Duración
5	110	25	100	1	0,01	15	2500	0,49815267	119,86 s.
4	110	25	50	1	0,01	No	1250	0,39396188	30,41 s.

Tabla 81: Resultados de predicción

Al igual que en los estudios anteriores, y tal y como era de esperar, los valores más bajos de capacidad de predicción de la RNA se obtienen cuando el número de Epocs es 1, independientemente del valor que tengan el resto de los criterios de parada. Como ya se a explicado, esta observación es más que comprensible, al realizar un único ciclo a la red sólo se le presentan los entrenamientos una vez, por lo que no puede ajustar los pesos de sus neuronas, impidiendo por tanto, alcanzar el mínimo global del error, punto en el que la red obtiene su mayor grado de fiabilidad.

Resumen de los valores definidos para el subconjunto 2:

- Número de pruebas: 50 | 100 | 150 | 200
- Número de registro de entrenamiento: 110
- Número de registro de test: 25
- Número de registros de CV: 0
- Número de Epocs: 1000 | 10000
- Número de MSE: 0,1

En el siguiente subconjunto de datos se encuentran aquellas simulaciones cuyos valores de correlación se oscilan entre el 0,905 y el 0,931.

Código	Entrena.	Test	Pruebas	Epocs	MSE	CV	Total	r(Mi)	Duración
27	110	25	200	1000	0,1	No	5000	0,93192653	241,86 s.
26	110	25	150	1000	0,1	No	3750	0,92625068	183,48 s.
28	110	25	50	1000	0,1	No	1250	0,92409167	56,94 s.
44	110	25	50	10000	0,1	No	1250	0,92262293	71,52 s.
43	110	25	200	10000	0,1	No	5000	0,91706553	280,05 s.
42	110	25	150	10000	0,1	No	3750	0,91331027	209,47 s.
41	110	25	100	10000	0,1	No	2500	0,90958073	145,95 s.
25	110	25	100	1000	0,1	No	2500	0,90512959	117,78 s.

Tabla 82: Resultados de predicción





Simplemente aumentando el número de Epocs los resultados mejoran muy notablemente. En este bloque se dan unas circunstancias muy similares a las del bloque 2 de estudio de la variable  $M_i$

Los resultados obtenidos revelan valores de  $r$  muy cercanos sin importar el número de Epocs, aunque parece comportarse mejor con un menor número de ciclos, lo que podría indicar que la RNA empieza a sufrir sobreaprendizaje en las simulaciones con más de 1000 ciclos.

El obtener coeficientes de correlación muy próximos con independencia del número de ciclos induce a pensar que el criterio de parada que se da en la mayoría de las ocasiones es el de MSE. Sin embargo, para las simulaciones con 1000 Epocs el consumo de recursos siempre es inferior, entre el 10 y 20 por cien, por lo que para estas simulaciones, en no pocas ocasiones, el criterio de parada que se cumple es el del número de ciclos. Para el resto, las simulaciones con 10000 Epocs, los datos no nos revelan directamente cual es el peso de cada uno de los criterios de parada, pero en base a los estudios anteriores sabemos que simulaciones sin CV y un MSE de 0,1 suelen conllevar un entrenamiento corto, es un criterio fácilmente alcanzable; y por el contrario, 10000 Epocs sin CV se traduce en un entrenamiento largo, por ello, todo hace pensar que el criterio de parada fundamental para estas simulaciones es el MSE.

Resumen de los valores definidos para el subconjunto 3:

- Número de pruebas: 50 | 100 | 150 | 200
- Número de registro de entrenamiento: 110
- Número de registro de test: 25
- Número de registros de CV: 0
- Número de Epocs: 1000 | 10000
- Número de MSE: 0,01



Para el siguiente bloque de datos los valores de la correlación oscilan entre el 0,935 y el 0,949.

Código	Entrena.	Test	Pruebas	Epocs	MSE	CV	Total	r(Mfinal)	Duración
36	110	25	50	10000	0,01	No	1250	0,94918083	78,31 s.
20	110	25	50	1000	0,01	No	1250	0,94670545	69,08 s.
35	110	25	200	10000	0,01	No	5000	0,94014361	311,70 s.
17	110	25	100	1000	0,01	No	2500	0,93901315	131,33 s.
34	110	25	150	10000	0,01	No	3750	0,93665617	227,88 s.
18	110	25	150	1000	0,01	No	3750	0,93625852	201,20 s.
19	110	25	200	1000	0,01	No	5000	0,93536796	266,19 s.
33	110	25	100	10000	0,01	No	2500	0,93489131	150,89 s.

Tabla 83: Resultados de predicción

Se mejoran alrededor de 3 centésimas los resultados del bloque anterior. Esta mejora se debe al aumento de la precisión de la variable MSE. El valor ahora para todas las simulaciones expuestas es de 0,01, mientras que el resto criterios toman idénticos valores que en el bloque anterior. La mejora es significativa dado el margen posible.

En el bloque anterior se ha planteado la posibilidad de la red estuviese sobre aprendiendo en las simulaciones con mayor número de Epocs. Esta posibilidad debe quedar descartada; este bloque de simulaciones tiene las mismas características a excepción del MSE, que es de 0,01, lo que se traduce en un entrenamiento más largo, y sin embargo la RNA ha continuado mejorando sus resultados, e incluso, en las simulaciones de este bloque, se comporta igual de bien independientemente del número de Epocs.

En cuanto al criterio de parada dominante se dan idénticas circunstancias que en las simulaciones del bloque anterior. Resumiendo, para aquellas con 10000 ciclos es MSE y para las de 1000 es Epocs y MSE, aunque este último en la mayoría de las ocasiones.



Resumen de los valores definidos para el subconjunto 4:

- Número de pruebas: 50 | 100 | 150 | 200
- Número de registro de entrenamiento: 110
- Número de registro de test: 25
- Número de registros de CV: 15
- Número de Epocs: 1000 | 10000
- Número de MSE: 0,1 | 0,01

En este último bloque están los mejores resultados, la mejora media es de 2 puntos. Los valores de coeficiente varían entre 0,960 y 0,972, todos ellos muy próximos a 1, por tanto, se los resultados se pueden catalogar como excelentes.

Código	Entrena.	Test	Pruebas	Epocs	MSE	CV	Total	r(Mfinal)	Duración
48	110	25	50	10000	0,1	15	1250	0,97201682	157,77 s.
32	110	25	50	1000	0,1	15	1250	0,97056254	119,47 s.
23	110	25	200	1000	0,01	15	5000	0,96972992	467,56 s.
24	110	25	50	1000	0,01	15	1250	0,96925297	111,58 s.
39	110	25	200	10000	0,01	15	5000	0,96857216	657,27 s.
21	110	25	100	1000	0,01	15	2500	0,96795208	236,55 s.
37	110	25	100	10000	0,01	15	2500	0,96697042	360,97 s.
46	110	25	150	10000	0,1	15	3750	0,96681195	584,22 s.
29	110	25	100	1000	0,1	15	2500	0,96680625	237,91 s.
30	110	25	150	1000	0,1	15	3750	0,96599492	368,53 s.
31	110	25	200	1000	0,1	15	5000	0,96582822	448,09 s.
47	110	25	200	10000	0,1	15	5000	0,96555407	798,09 s.
40	110	25	50	10000	0,01	15	1250	0,96554857	188,31 s.
38	110	25	150	10000	0,01	15	3750	0,96466786	607,91 s.
22	110	25	150	1000	0,01	15	3750	0,9637165	359,80 s.
45	110	25	100	10000	0,1	15	2500	0,96044918	442,41 s.

Tabla 84: Resultados de predicción

Este subconjunto lo componen aquellas simulaciones en los que se ha incluido como criterio de parada la validación cruzada. La características de este último bloque del estudio son muy similares a las que se han obtenido para su recíproco en el estudio de la Masa Final.



Es fácil observar que una vez se ha incluido la validación cruzada, los valores asignados para el resto de criterios de parada dejan de ser determinantes para el valor  $r$  que proporciona Simulador. Sin importar el resto de valores que toman los demás criterios de parada, todas las simulaciones aumentan de media 2 centésimas el valor de la correlación, esta es una buena mejora aunque no tan acentuada como la que se produjo para la “Mfinal”. Lo que vuelve a quedar claro es que la evaluación de CV durante el entrenamiento ayuda a mejorar la capacidad de predicción de la red, además de controlar el sobreaprendizaje

En cuanto a los criterios de parada, todas las simulaciones con 1000 Epocs tienen un tiempo de duración muy similares sin importar el valor de MSE. Esto indica claramente que para estas simulaciones el número de ciclos es el criterio de parada dominante. Algo que queda confirmado al observar que las mismas simulaciones pero con 10000 Epocs aumentan, como mínimo, su consumo de recursos un 50 por cien.

Si se analizan los tiempos de las simulaciones se puede ver con claridad que aquellas con 1000 Epocs son las más rápidas, y particularmente, las que tiene un MSE igual a 0,1; simulaciones con las mismas características pero con distinto número de Epocs tienen un coste, al menos, del doble de tiempo, llegando incluso en ocasiones a cuadruplicarlo. Esto indica que en ellas el número de ciclos es el principal criterio de parada, cediendo, eso sí, algo de protagonismo a MSE cuando este tiene el valor de 0,1.

Para las simulaciones con 10000 Epocs, mismas características, pero con distinto valor para MSE, los consumos no tienen unas diferencias significativas, es más, en algunos casos, las simulaciones con error cuadrático mayor tardan algo más. Esto indica claramente que en estas simulaciones el número de Epocs y CV, son los criterios de parada que se dan en la mayoría de las ocasiones.

Se debe volver a destacar, que al igual que ocurría en el estudio de la Masa Final, el consumo de recursos al introducir CV en las simulaciones ha aumentado. El aumento en este estudio no es tan notable, lo que indica que el grado de



sobreaprendizaje que existe ahora es superior al que se dio en “Mfinal”; aunque de igual manera este es mínimo.

Tal y como ya se explicó en el apartado reciproco de Masa Final, en principio, CV supone un consumo extra de recursos, ya que se debe evaluar un nuevo criterio de parada durante el entrenamiento. Una de las principales características de la validación cruzadas es evitar que la RNA sufra sobreaprendizaje, esto es, CV es capaz de detectarlo y detener el entrenamiento, sin esperar a que se cumpla cualquiera de los restantes criterios, lo que provoca un entrenamiento más corto, y por tanto, una disminución del consumo de recursos. Por ello, si el entrenamiento aumenta al introducir CV, en principio, se puede afirmar que no existe sobre aprendizaje, o de existir, este no es significativo.

Todos los valores del coeficiente de correlación en este bloque son muy buenos, y se mueven en torno a una horquilla de una milésima, por ello, parece lógico seleccionar como más óptima aquella configuración que proporciona un entrenamiento rápido, pues no tiene mucho sentido alargar el proceso de entrenamiento para obtener unos resultados tan similares.

Partiendo de la anterior premisa, se ha seleccionado como la configuración más óptima la de la simulación 48.

- 10.000 Epocs
- 0,1 MSE
- 15 CV
- 50 pruebas

Esta configuración es la que guarda mejor relación entre el coeficiente de correlación y la cantidad de recursos consumidos para lograrlo. El entrenamiento, aunque puede parecer corto, obtiene el mejor valor de  $r$ .



### 6.3.3.4. DOP

Resumen de los valores definidos para:

- Número de pruebas: 50 | 100 | 150 | 200
- Número de registro de entrenamiento: 48
- Número de registro de test: 10
- Número de registros de CV: 0 - 5
- Número de Epocs: 1 | 1.000 | 10.000
- Número de MSE: 0,01 | 0,1

Código	Entrena.	Test	Pruebas	Epocs	MSE	CV	Total	r(Mi)	Duración
1	48	10	100	1	0,01	No	1000	0,241253259	46,13 s.
2	48	10	150	1	0,01	No	1500	0,415751782	73,91 s.
3	48	10	200	1	0,01	No	2000	0,537905511	132,42 s.
4	48	10	50	1	0,01	No	500	0,67976534	23,09 s.
5	48	10	100	1	0,01	5	1000	0,001499757	92,78 s.
6	48	10	150	1	0,01	5	1500	0,660626787	143,63 s.
7	48	10	200	1	0,01	5	2000	0,612408631	198,36 s.
8	48	10	50	1	0,01	5	500	0,688534688	48,23 s.
9	48	10	100	1	0,1	No	1000	0,514333791	46,02 s.
10	48	10	150	1	0,1	No	1500	0,555638344	71,73 s.
11	48	10	200	1	0,1	No	2000	0,417788422	127,27 s.
12	48	10	50	1	0,1	No	500	0,480577441	23,13 s.
13	48	10	100	1	0,1	5	1000	0,328015558	102,19 s.
14	48	10	150	1	0,1	5	1500	0,41730586	153,69 s.
15	48	10	200	1	0,1	5	2000	0,627038102	217,28 s.
16	48	10	50	1	0,1	5	500	0,043273583	51,38 s.
17	48	10	100	1000	0,01	No	1000	0,933438379	81,80 s.
18	48	10	150	1000	0,01	No	1500	0,929720747	180,41 s.
19	48	10	200	1000	0,01	No	2000	0,885422929	274,28 s.
20	48	10	50	1000	0,01	No	500	0,942005575	37,95 s.
21	48	10	100	1000	0,01	5	1000	0,879478363	165,34 s.
22	48	10	150	1000	0,01	5	1500	0,886749619	245,20 s.
23	48	10	200	1000	0,01	5	2000	0,876621299	338,50 s.
24	48	10	50	1000	0,01	5	500	0,891259856	82,98 s.
25	48	10	100	1000	0,1	No	1000	0,833154463	92,73 s.
26	48	10	150	1000	0,1	No	1500	0,816944102	141,19 s.
27	48	10	200	1000	0,1	No	2000	0,783223014	187,19 s.
28	48	10	50	1000	0,1	No	500	0,798126168	46,91 s.
29	48	10	100	1000	0,1	5	1000	0,841649172	156,50 s.



Código	Entrena.	Test	Pruebas	Epocs	MSE	CV	Total	r(Mi)	Duración
30	48	10	150	1000	0,1	5	1500	0,877839694	248,42 s.
31	48	10	200	1000	0,1	5	2000	0,880308306	336,86 s.
32	48	10	50	1000	0,1	5	500	0,897002488	80,45 s.
33	48	10	100	10000	0,01	No	1000	0,915943532	159,05 s.
34	48	10	150	10000	0,01	No	1500	0,91296793	241,84 s.
35	48	10	200	10000	0,01	No	2000	0,925024767	320,67 s.
36	48	10	50	10000	0,01	No	500	0,934901668	80,08 s.
37	48	10	100	10000	0,01	5	1000	0,859631741	206,72 s.
38	48	10	150	10000	0,01	5	1500	0,891091021	324,17 s.
39	48	10	200	10000	0,01	5	2000	0,887715722	476,64 s.
40	48	10	50	10000	0,01	5	500	0,887149975	81,27 s.
41	48	10	100	10000	0,1	No	1000	0,799980734	114,23 s.
42	48	10	150	10000	0,1	No	1500	0,804636897	179,55 s.
43	48	10	200	10000	0,1	No	2000	0,783512968	241,95 s.
44	48	10	50	10000	0,1	No	500	0,784339943	55,53 s.
45	48	10	100	10000	0,1	5	1000	0,878443278	218,44 s.
46	48	10	150	10000	0,1	5	1500	0,883388387	362,50 s.
47	48	10	200	10000	0,1	5	2000	0,879251997	471,61 s.
48	48	10	50	10000	0,1	5	500	0,888771826	108,83 s.

Tabla 85: Resultados de predicción

Al igual en los estudios anteriores, el número de simulaciones para esta parte del estudio es elevado, concretamente 48 simulaciones, a continuación, para que al lector le sea más fácil de seguir el por qué de las conclusiones a las que se ha llegado, se presentan aquellos subconjuntos de la tabla anterior que nos permiten mostrar claramente las informaciones extrapolables a esta.

Resumen de los valores definidos para el subconjunto 1:

- Número de pruebas: 50 | 100 | 150 | 200
- Número de registro de entrenamiento: 48
- Número de registro de test: 10
- Número de registros de CV: 0 - 5
- Número de Epocs: 1
- Número de MSE: 0,01 | 0,1



El primer subconjunto que se define claramente en la tabla de resultados es el de aquellas simulaciones en las que el número de Epochs es 1. Para todos ellos el coeficiente de correlación entre los resultados reales y los proporcionados por la RNA es bastante bajo, siempre inferiores a 0,69.

Código	Entrena.	Test	Pruebas	Epochs	MSE	CV	Total	r(Mi)	Duración
8	48	10	50	1	0,01	5	500	0,68853469	48,23 s.
4	48	10	50	1	0,01	No	500	0,67976534	23,09 s.
6	48	10	150	1	0,01	5	1500	0,66062679	143,63 s.
15	48	10	200	1	0,1	5	2000	0,6270381	217,28 s.
7	48	10	200	1	0,01	5	2000	0,61240863	198,36 s.
10	48	10	150	1	0,1	No	1500	0,55563834	71,73 s.
3	48	10	200	1	0,01	No	2000	0,53790551	132,42 s.
9	48	10	100	1	0,1	No	1000	0,51433379	46,02 s.
12	48	10	50	1	0,1	No	500	0,48057744	23,13 s.
11	48	10	200	1	0,1	No	2000	0,41778842	127,27 s.
14	48	10	150	1	0,1	5	1500	0,41730586	153,69 s.
2	48	10	150	1	0,01	No	1500	0,41575178	73,91 s.
13	48	10	100	1	0,1	5	1000	0,32801556	102,19 s.
1	48	10	100	1	0,01	No	1000	0,24125326	46,13 s.
16	48	10	50	1	0,1	5	500	0,04327358	51,38 s.
5	48	10	100	1	0,01	5	1000	0,00149976	92,78 s.

Tabla 86: Resultados de predicción

Al igual que en los estudios anteriores, y tal y como era de esperar, los valores más bajos de capacidad de predicción de la RNA se obtienen cuando el número de Epochs es 1, independientemente del valor que tengan el resto de los criterios de parada. Como ya se a explicado, esta observación es más que comprensible, al realizar un único ciclo a la red sólo se le presentan los entrenamientos una vez, por lo que no puede ajustar los pesos de sus neuronas, impidiendo por tanto, alcanzar el mínimo global del error, punto en el que la red obtiene su mayor grado de fiabilidad.

Resumen de los valores definidos para el subconjunto 2:

- Número de pruebas: 50 | 100 | 150 | 200
- Número de registro de entrenamiento: 48
- Número de registro de test: 10
- Número de registros de CV: 0





- Número de Epocs: 1000 | 10000
- Número de MSE: 0,1

En el siguiente subconjunto de datos se encuentran aquellas simulaciones cuyos valores de correlación se oscilan entre el 0,78 y el 0,84.

<i>Código</i>	<i>Entrena.</i>	<i>Test</i>	<i>Pruebas</i>	<i>Epocs</i>	<i>MSE</i>	<i>CV</i>	<i>Total</i>	<i>r(Mi)</i>	<i>Duración</i>
25	48	10	100	1000	0,1	No	1000	0,83315446	92,73 s.
26	48	10	150	1000	0,1	No	1500	0,8169441	141,19 s.
42	48	10	150	10000	0,1	No	1500	0,8046369	179,55 s.
41	48	10	100	10000	0,1	No	1000	0,79998073	114,23 s.
28	48	10	50	1000	0,1	No	500	0,79812617	46,91 s.
44	48	10	50	10000	0,1	No	500	0,78433994	55,53 s.
43	48	10	200	10000	0,1	No	2000	0,78351297	241,95 s.
27	48	10	200	1000	0,1	No	2000	0,78322301	187,19 s.

**Tabla 87: Resultados de predicción**

Al aumentar el número de Epocs los resultados mejoran de forma significativa llegando a una cota máxima 10 puntos superior a la del bloque anterior.

El obtener coeficientes de correlación muy próximos con independencia del número de ciclos induce a pensar que el criterio de parada que se da en la mayoría de las ocasiones es el de MSE. Sin embargo, para las simulaciones con 1000 Epocs el consumo de recursos siempre es inferior, entre el 15 y 22 por cien, por lo que para estas simulaciones, en no pocas ocasiones, el criterio de parada que se cumple es el del número de ciclos. Para el resto, las simulaciones con 10000 Epocs, los datos no nos revelan directamente cual es el peso de cada uno de los criterios de parada, pero en base a los estudios anteriores sabemos que simulaciones sin CV y un MSE de 0,1 suelen conllevar un entrenamiento corto, es un criterio fácilmente alcanzable; y por el contrario, 10000 Epocs sin CV se traduce en un entrenamiento largo, por ello, todo hace pensar que el criterio de parada fundamental para estas simulaciones es el MSE.



Se debe reseñar que pese al menor número de registros para test y entrenamiento que dispone la RNA en el estudio actual, esta se comporta de forma análoga a los estudios anteriores. Por ahora, esta reducción no parece influir en sus resultados.

Resumen de los valores definidos para el subconjunto 3:

- Número de pruebas: 50 | 100 | 150 | 200
- Número de registro de entrenamiento: 48
- Número de registro de test: 10
- Número de registros de CV: 5 - 0
- Número de Epocs: 1000 | 10000
- Número de MSE: 0,1 | 0,01

Para el siguiente bloque de datos los valores de la correlación oscilan entre el 0,842 y el 0,897.

Código	Entrena.	Test	Pruebas	Epocs	MSE	CV	Total	r(Mfinal)	Duración
32	48	10	50	1000	0,1	5	500	0,897002488	80,45 s.
24	48	10	50	1000	0,01	5	500	0,891259856	82,98 s.
38	48	10	150	10000	0,01	5	1500	0,891091021	324,17 s.
48	48	10	50	10000	0,1	5	500	0,888771826	108,83 s.
39	48	10	200	10000	0,01	5	2000	0,887715722	476,64 s.
40	48	10	50	10000	0,01	5	500	0,887149975	81,27 s.
22	48	10	150	1000	0,01	5	1500	0,886749619	245,20 s.
19	48	10	200	1000	0,01	No	2000	0,885422929	274,28 s.
46	48	10	150	10000	0,1	5	1500	0,883388387	362,50 s.
31	48	10	200	1000	0,1	5	2000	0,880308306	336,86 s.
21	48	10	100	1000	0,01	5	1000	0,879478363	165,34 s.
47	48	10	200	10000	0,1	5	2000	0,879251997	471,61 s.
45	48	10	100	10000	0,1	5	1000	0,878443278	218,44 s.
30	48	10	150	1000	0,1	5	1500	0,877839694	248,42 s.
23	48	10	200	1000	0,01	5	2000	0,876621299	338,50 s.
37	48	10	100	10000	0,01	5	1000	0,859631741	206,72 s.
29	48	10	100	1000	0,1	5	1000	0,841649172	156,50 s.

Tabla 88: Resultados de predicción



En este bloque de simulaciones las mejoras en el coeficiente de correlación son muy notables, entre 6 y 11 décimas, respecto al anterior. Hasta ahora, en las variables anteriores el salto cualitativo en los resultados respecto al bloque 2 se producía al aumentar la precisión de error cuadrático de 0,1 a 0,01, pero en este caso, no es así. El salto cualitativo se produce al introducir CV, sin importar el valor que tenga el MSE. Este escenario no se había planteado en ninguno de los anteriores estudios, aunque también es cierto, que esta variable es aquella en el que se ha contado con menor número de datos de entrada. ¿Por qué aparecen ahora todos los registros de CV en vez de en el último bloque como era lo habitual?

Al analizar el consumo de recursos respecto al bloque anterior, vemos que la incursión de la validación cruzada con MSE 0,1 no ha propiciado una disminución de estos respecto al bloque anterior, al contrario, lo que invita a pensar que pese al menor número de datos para el estudio de esta variable, con MSE igual 0,1 el grado de sobreaprendizaje no es relevante, por tanto, CV no es un criterio de parada habitual frente al MSE 0,1.

Al centrarse únicamente en las simulaciones del bloque actual cabe destacar que simulaciones con idénticas características pero con distinto MSE consume una cantidad de recursos muy similares, esto indica que MSE no es un criterio de parada relevante. Esto se corrobora al fijarse en que simulaciones idénticas pero con distinto número de ciclos siempre necesitan una cantidad muy distinta de recursos, en la mayoría de ocasiones las simulaciones con 10000 Epochs consumen cerca de un 30% más de tiempo. Por tanto, se puede asegurar que el número de Epochs es un criterio muy relevante en este bloque del estudio.

Como nota, cabe destacar la aparición de un registro sin CV en este subconjunto. ¿A qué se puede deber esto?

Mediante el análisis del próximo, y último subconjunto de datos, se intentará aclarar las cuestiones que se han planteado.



#### Resumen de los valores definidos para el subconjunto 4:

- Número de pruebas: 50 | 100 | 150 | 200
- Número de registro de entrenamiento: 48
- Número de registro de test: 10
- Número de registros de CV: 0
- Número de Epocs: 1000 | 10000
- Número de MSE: 0,01

En este último bloque están los mejores resultados, con una mejora mínima de 2 décimas sobre el anterior bloque, llegando a una máxima de entorno a las 10 décimas. Los valores de coeficiente varían entre 0,913 y 0,942, todos ellos bastante próximos a 1; si se tiene en cuenta la cantidad de datos, se podrían catalogar los resultados como muy buenos.

Código	Entrena.	Test	Pruebas	Epocs	MSE	CV	Total	r(Mfinal)	Duración
20	48	10	50	1000	0,01	No	500	0,94200558	37,95 s.
36	48	10	50	10000	0,01	No	500	0,93490167	80,08 s.
17	48	10	100	1000	0,01	No	1000	0,93343838	81,80 s.
18	48	10	150	1000	0,01	No	1500	0,92972075	180,41 s.
35	48	10	200	10000	0,01	No	2000	0,92502477	320,67 s.
33	48	10	100	10000	0,01	No	1000	0,91594353	159,05 s.
34	48	10	150	10000	0,01	No	1500	0,91296793	241,84 s.

**Tabla 89: Resultados de predicción**

Este cuarto subconjunto está compuesto por simulaciones con un MSE de 0,01 y sin validación cruzada, unas características totalmente distintas a las de los bloques de simulaciones análogos de todas las variables anteriores.

La mejora mínima respecto al conjunto anterior es de 2 puntos, y la máxima de casi 10. Una mejora notable y muy peculiar, puesto que se ha producido al eliminar la validación cruzada. Todos los conjuntos de pruebas respecto a sus análogos del bloque anterior pero sin CV, obtienen unos resultados mejores y consumen alrededor del 35%



menos de recursos. Por esta disminución de costes se puede concluir que no se está produciendo sobre aprendizaje. Pero ¿A qué se debe la mejora de los resultados? La respuesta es válida también para la pregunta planteada en el bloque anterior “¿Por qué aparecen ahora todos los registros de CV en vez de en el último bloque como era lo habitual?”. Partimos de la realidad de que para el estudio de esta variable contamos con bastantes menos ensayos reales que para el resto. También es cierto, que hasta ahora no ha existido problema de sobre aprendizaje en esta red; CV sólo ha aportado un consumo extra de recursos sin mejorar resultados, ni ser el criterio de parada dominante en ninguno de los subconjuntos. Por ello, si los ensayos dedicados a CV se destinan para el entrenamiento de la RNA, la variedad muestral de este aumenta, lo que hace mejorar la calidad del mismo, y por ende, el resultado final del test.

Al centrarse en los resultados del conjunto actual es fácil comprobar que cuando el número de ciclos es 1000 este es el criterio de parada que rige en todas las pruebas. Esto se comprueba al ver que idénticas simulaciones pero con un mayor número de ciclos consumen siempre muchos más recursos, algunas hasta más del 50%. En cuanto a las simulaciones con 10000 ciclos no se puede saber cuál es el criterio de parada dominante, si MSE o Epocs.

Las simulaciones más cortas son aquellas que de media obtiene un mejor resultado. Esto indica que en los entrenamientos más largos la red puede estar sufriendo sobreaprendizaje, si bien, este debe ser mínimo, puesto que CV pasa de puntillas sobre él y los valores de  $r$  son muy similares para todas las pruebas.

Partiendo de la anterior premisa, se ha seleccionado como la configuración más óptima la de la simulación 48.

- 1.000 Epocs
- 0,01 MSE
- 0 CV
- 50 pruebas



**Proyecto:** Analisis de Técnicas para Medir la Influencia de las Variables del Vector de Entrada de un Perceptron Multicapa

**Autor:** Roberto Romero Martínez

**Capítulo:** Evaluación y experimentación

Esta es la configuración que obtiene un coeficiente de correlación superior, y además, es la que necesita un entrenamiento más corto.



## 6.4. Análisis de sensibilidad

### 6.4.1. INTRODUCCIÓN

Esta parte del estudio tiene como objetivo averiguar cómo influye cada una de las variables del fichero de entrada en la capacidad de aprendizaje de la red, es decir, en la salida que esta nos proporciona. El análisis de sensibilidad debe establecer la jerarquía de las variables, lo que permitirá direccionar el esfuerzo y los recursos en la obtención de datos para aquellas que sean más importantes y/o necesarias para mejorar los resultados, basándose en la realidad actual.

Este tipo de análisis es la base sobre la que se debe apoyar la decisión sobre la conveniencia de eliminar, o no, una variable de entrada. Es pertinente recordar que la eliminación de variables de entrada irrelevantes para un estudio produce efectos beneficiosos sobre la red. La hará más pequeña, por tanto, simple y rápida, y mejorará su capacidad de generalización.

Para establecer el peso de la relación entre las variables de entrada y la salida obtenida se han utilizado dos métodos, uno implementado en Neurosolutions, “Sensitivity about the mean”, y otro implementado en Simulador, “Fortaleza frente a valores atípicos”.



#### 6.4.2. SENSITIVITY ABOUT THE MEAN

Este proceso de análisis proporciona una medida sobre la importancia relativa entre las diferentes variables de entrada de una red artificial de neuronas, e ilustra como el modelo de salida varía en respuesta a la variación de un entrada.

El valor de la primera variable de entrada va ser modificado dentro del rango fijado por +/-un número de veces la desviación típica; dicho número es indicado por el usuario. De igual forma, este debe indicar la cantidad de fluctuaciones, por encima y por debajo de la media, que desea que se realicen. El resto de variable son fijadas. Este proceso se repite por cada una de las entradas.

Como resultado del análisis se obtiene un informe que detalla cómo ha variado la salida en función de las variaciones realizadas sobre cada entrada. Es resumen está compuesto por:

- Un gráfico de barras indicando la importancia relativa de cada una de las entradas sobre la salida.
- Un tabla con las desviación típica de la salida dividida por la desviación típica de la entrada variada para obtener dicha salida
- Una gráfica por cada variable de entrada indicando la relación entre la salida obtenida y el rango de variaciones utilizado para dicha entrada.





### 6.4.2.1. INFORME SENSITIVITY ABOUT THE MEAN (SAM)

A continuación se presenta el informe obtenido mediante Sensitivity About the Mean.

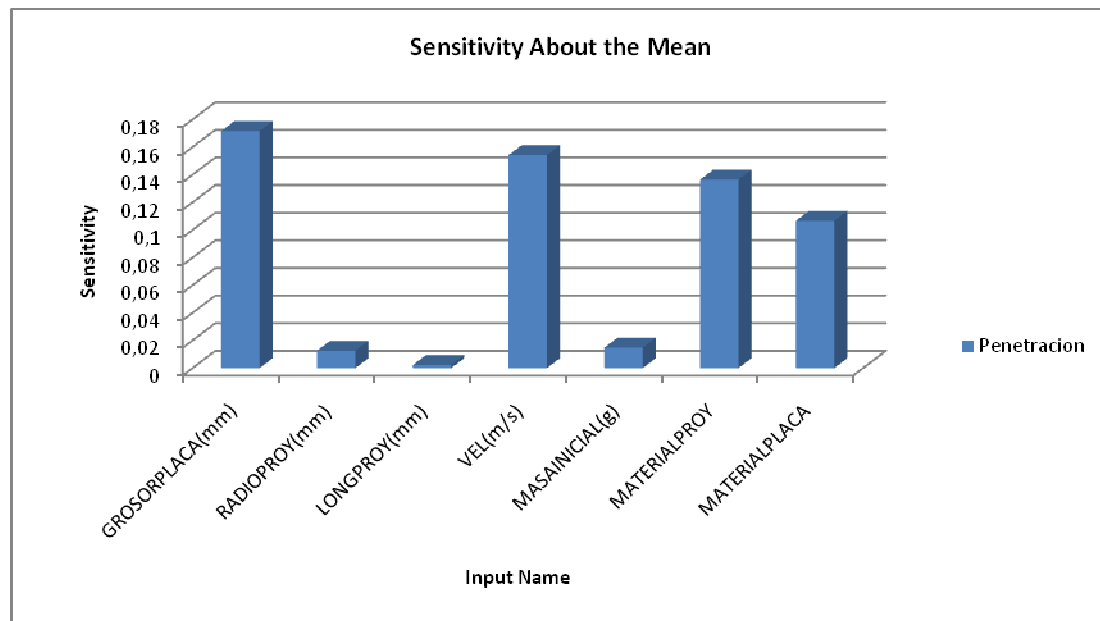


Ilustración 34: Resumen resultados SAM

La Ilustración 34 muestra mediante un gráfico de barras la influencia relativa que tiene cada una de las variables de entrada en la salida propuesta por la red. Según el gráfico las variables “Grosor de la placa”, “Velocidad”. “Material del proyectil” y “Material de la placa”, son las que tienen influencia tienen en la salida pronosticada por la red. Por el contrario, las variables, “Radio del proyectil”, “Longitud del proyectil” y “Masa inicial del proyectil” no tienen un peso significativo en el proceso.

<i>Sensitivity</i>	<i>Penetracion</i>
GROSORPLACA (mm)	0,171645684
RADIOPROY (mm)	0,012602047
LONGPROY (mm)	0,002229894
VEL (m/s)	0,154256945
MASAINICIAL (g)	0,014824647



<i>Sensitivity</i>	<i>Penetracion</i>
MATERIALPROY	0,136492558
MATERIALPLACA	0,106665648

Tabla 90: Resumen resultados SAM

La Tabla 90 muestra la desviación típica de la salida dividida por la desviación típica de la entrada variada para obtener dicha salida. Cuanto más cerca de 0 esté el valor obtenido, más uniforme será la salida con respecto a la modificación de la entrada, por tanto, mayor será la precisión de la salida, menor el grado de incertidumbre, lo que se puede traducir como, menor será la influencia de dicha variable en la salida generada por la red.

Las gráficas que se presentan a continuación muestran la relación entre valores de entrada de una variable y la salida producida por la red.

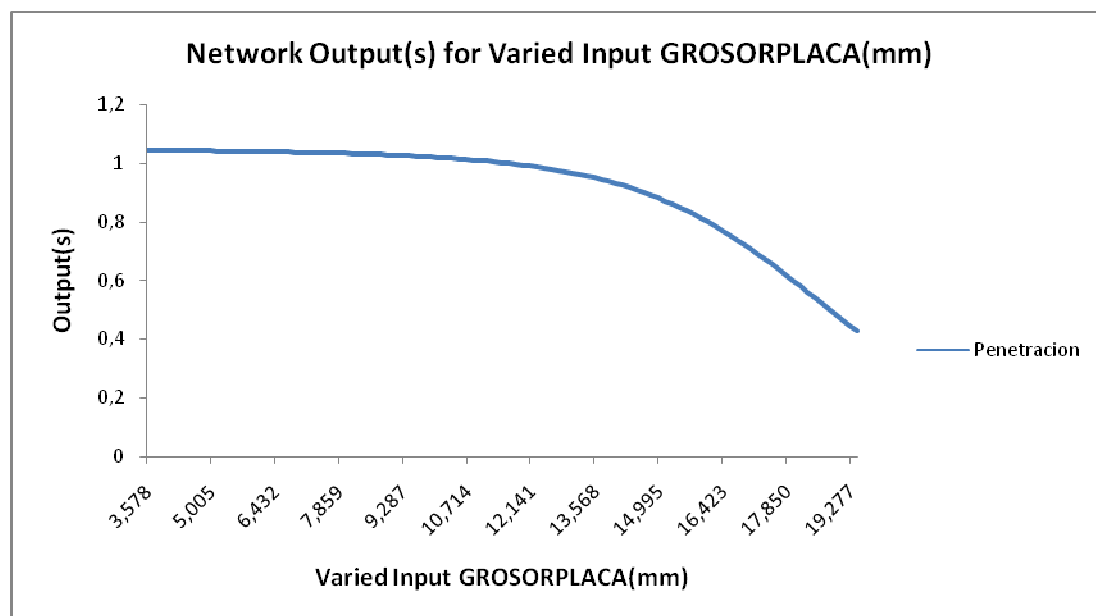


Ilustración 35: Resultados SAM grosor

La relación expresada por la Ilustración 35 indica que a mayor grosor de la placa mayor es la posibilidad de que el proyectil no la atraviese. Esta relación se acentúa claramente al sobrepasar los 12 cm de espesor. Esto también ayuda a contrastar lo



aseverado anteriormente, que el espesor de la placa es altamente influyente en los resultados obtenidos por la RNA.

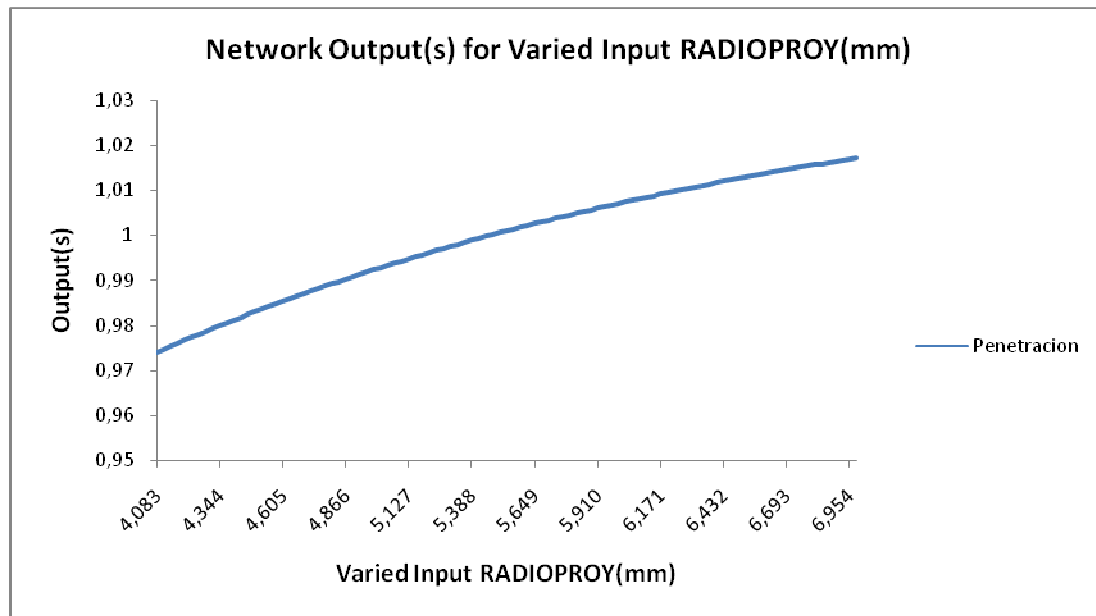


Ilustración 36: Resultados SAM radio

La relación expresada por la Ilustración 36 indica que a mayor radio del proyectil mayor es la posibilidad de que el proyectil atraviese la placa. Aunque la relación parece clara en la gráfica, al fijarse en el eje Y se puede observar que el rango de variación de la salida es mínimo, una horquilla de 0,03, por tanto, esta relación es prácticamente lineal, por lo que no se le debe otorgar mucha credibilidad. Las características de la relación expresada en la gráfica ayudan a confirmar lo anteriormente indicado, el radio del proyectil no tiene un peso importante en el proceso predicción.

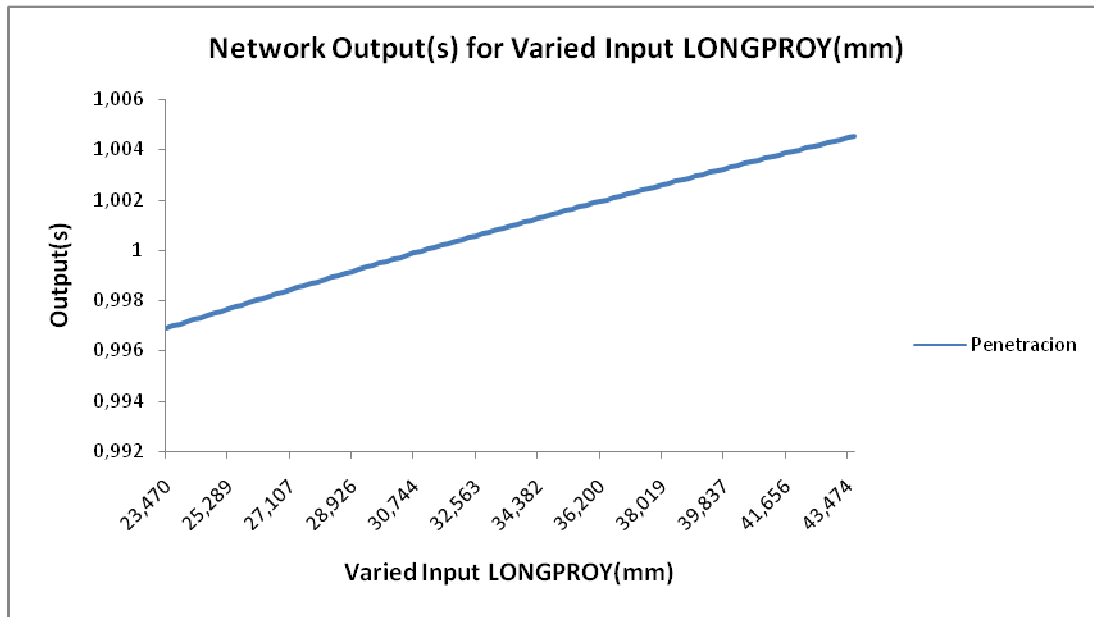


Ilustración 37: Resultados SAM longitud

Con esta variable ocurre exactamente lo mismo que con la anterior, incluso, se acentúa un poco más lo anteriormente expresado, puesto que ahora la horquilla es de 0.003 entorno a 1, lo que hace convierte a esta relación en precitamente lineal. Por ello, la variable longitud del proyectil, tampoco tiene una incidencia reseñable sobre el proceso.

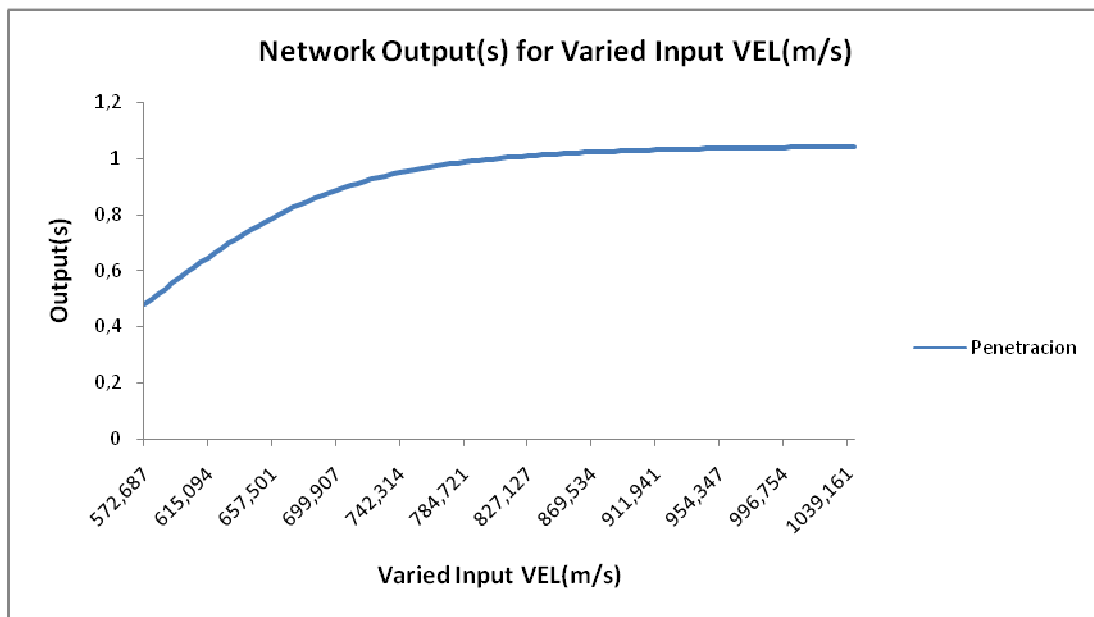


Ilustración 38: Resultados SAM velocidad



Como era de esperar, a mayor velocidad existen más posibilidades de que el proyectil atraviese la placa. Cabe destacar como norma general, que cuando la velocidad supera los 800 m/s los proyectiles atraviesan la placa.

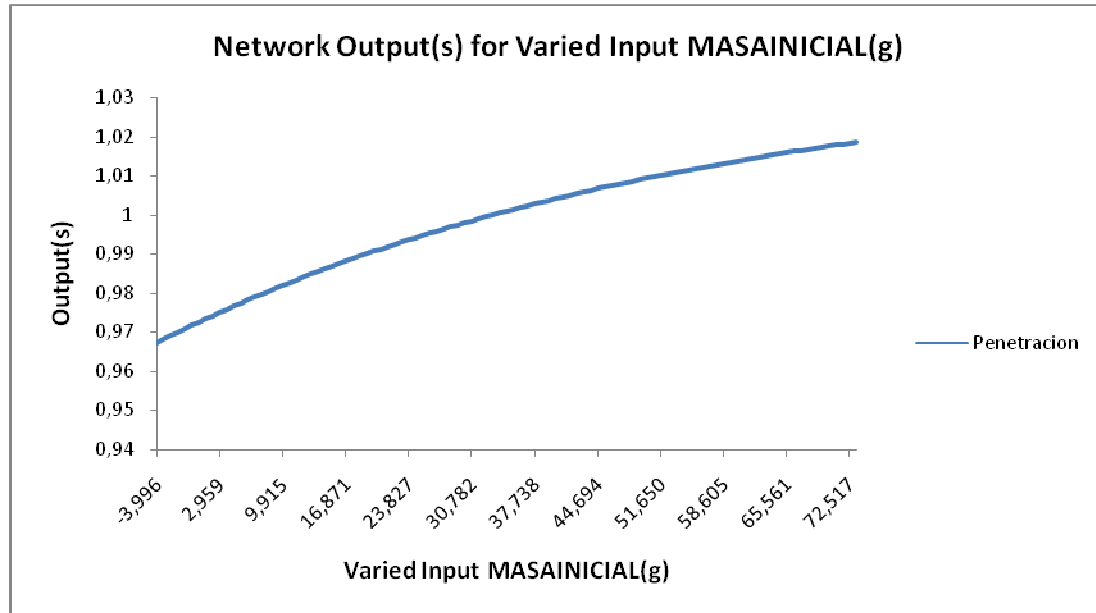


Ilustración 39: Resultados SAM masa inicial

Con la variable Masa inicial nos encontramos en idéntico caso que con todas aquellas variables que no tienen incidencia sobre la salida de la RNA.

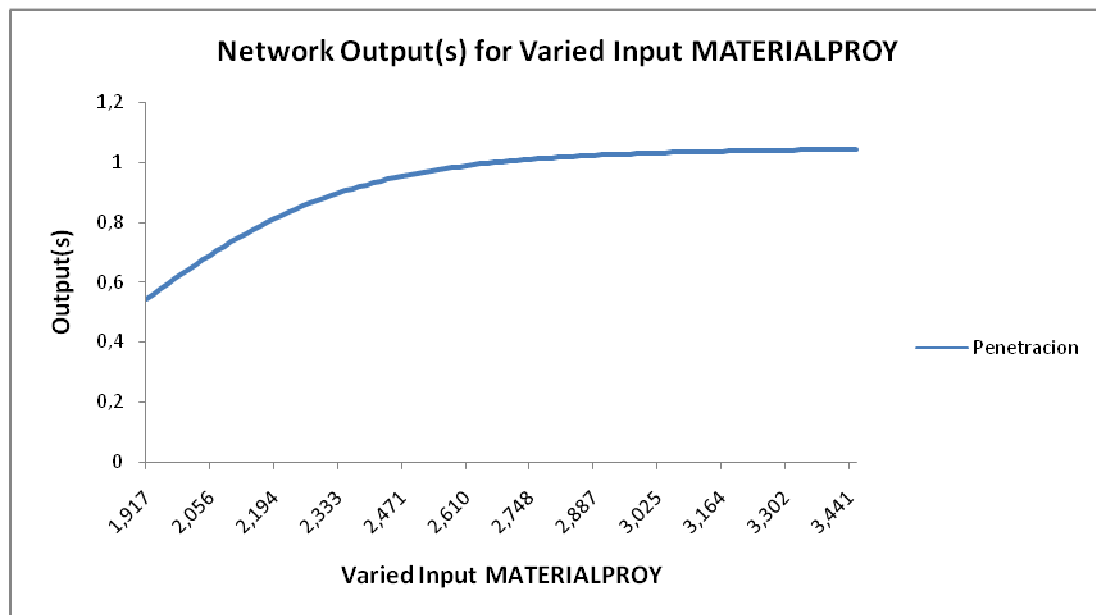


Ilustración 40: Resultados SAM material placa



Como material del proyectil existen tres tipos, tipo 2, 3 y 4. Los datos de la gráfica inducen a pensar que la numeración de los tipos de materiales es inversamente proporcional a su dureza, es decir, cuanto más alto es el número asignado al tipo de material, menor es la dureza de este.

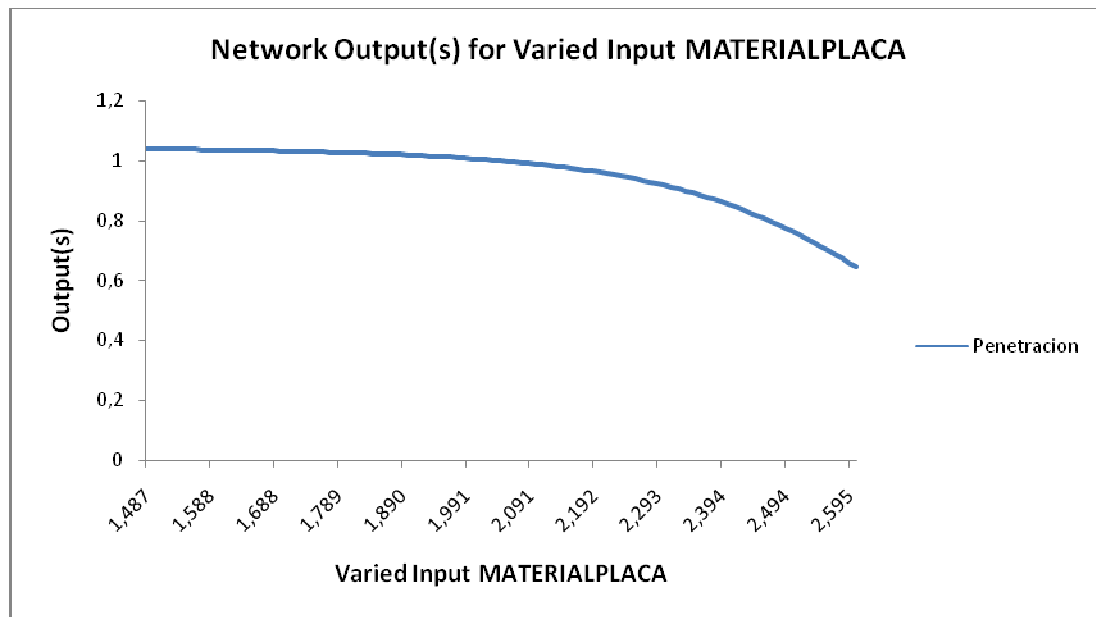


Ilustración 41: Resultados SAM material proyectil

Existen tres tipos de materiales para la placa, tipo 1, 2 y 3. Ahora ocurre exactamente lo contrario a lo sucedido para la variable anterior, en este caso, los resultados mostrados por la gráfica, llevan a pensar que cuanto mayor sea la numeración del tipo de material mayor será la dureza de este.



### 6.4.3. FORTALEZA FRENTE A VALORES ATÍPICOS (FFVA)

Este estudio pretende mostrar cómo se comporta la red al enfrentarse ante un valor de una variable que se encuentra fuera del rango de valores sobre el que ha sido entrada. Para obtener dicha información, primero hay que saber en qué rango de valores se mueve dicha variable dentro de los ensayos destinados al entrenamiento. Es decir, hay que conocer el valor mínimo y máximo que toma la variable dentro de los ensayos de entrenamiento.

Una vez que se calcula el rango de valores para cada una de las variables que intervienen en el entrenamiento, hay que averiguar qué valor toma cada variable en cada uno de los ensayos destinados al test. De esta forma se puede comprobar para cada test qué variables están dentro del rango con el que se entrenó la red neuronal y cuáles están fuera y que, por tanto, son valores que pueden hacer fallar el test.

Este método de análisis ha sido empleado con anterioridad en otros estudios [48] [49] [50] [51] y a partir de él se obtenían las siguientes conclusiones, y cito textualmente:

*“Algunas de las conclusiones que se pueden sacar con estos datos son las siguientes:*

- *Si la red falla un ensayo de test (la red “dice” que el proyectil sí penetra cuando en realidad no lo hace o viceversa), se puede comprobar si hay alguna variable cuyo valor está fuera del rango con el que se entrenó la red. Si esto ocurre con la misma variable en varios test, se puede concluir que dicha variable es crítica en el aprendizaje de la red neuronal y por tanto hay que tener cuidado con los valores que adopta.*
- *Cuando la red acierta un ensayo de test, se puede comprobar si alguna de las variables está fuera del rango con el que se entrenó. Si esto ocurre con la misma variable en varios test, se puede concluir que dicha variable no es determinante en el aprendizaje de la red y por tanto el valor que adopte tanto en el entrenamiento como en el test no influye en el resultado final.”*



Con los resultados obtenidos en los diferentes estudios realizados, y ayudándonos con “Sensitivity about the mean” se va a demostrar que estos criterios para obtener conclusiones son demasiado genéricos y poco exactos.

Para un mejor seguimiento de las conclusiones la tabla que se presenta a continuación detalla los resultados obtenidos en los estudios con un n° de Epochs mayor que 1 para el proceso “Fortaleza frente a valores atípicos”.





**Proyecto:** Analisis de Técnicas para Medir la Influencia de las Variables del Vector de Entrada de un Perceptron Multicapa

**Autor:** Roberto Romero Martínez

**Capítulo:** Analisis de sensibilidad

Cód.	%Aciertos	Entrena.	Test	G.PLACA	R.PROY	L.PROY	VEL(m/s)	M. INICIAL(g)	MAT.PROY	MAT.PLACA
17	96.2	150	40	0	0	2,857142857	6,818181818	0	0	0
18	96.7	150	40	2,380952381	0	2,127659574	0	2,272727273	0	0
19	95.05	150	40	0	0	3,896103896	18,18181818	0	0	0
20	95.35	150	40	0	0	0	22,22222222	6,25	0	0
21	95.65	150	40	0	0	2,857142857	0	0	0	0
22	94.8375	150	40	0	0	4,255319149	9,677419355	2,272727273	0	0
23	94.75	150	40	0	0	1,298701299	1,515151515	0	0	0
24	91.425	150	40	0	0	3,846153846	5,555555556	12,5	0	0
25	91.95	150	40	0	0	2,857142857	0	0	0	0
26	92.2375	150	40	0	0	6,382978723	9,677419355	0	0	0
27	92.65	150	40	0	0	2,597402597	4,545454545	0	0	0
28	95.925	150	40	0	0	3,846153846	0	0	0	0
29	95.3	150	40	0	0	2,857142857	6,818181818	0	0	0
30	94.7625	150	40	0	0	6,382978723	0	0	0	0
31	95.1	150	40	0	0	1,298701299	1,515151515	0	0	0
32	93.975	150	40	0	0	0	5,555555556	0	0	0
33	95.71666667	150	40	5,714285714	0	2,857142857	2,272727273	0	0	0
34	95.55	150	40	0	0	6,382978723	0	18,18181818	0	0
35	92.7	150	40	0	0	12,98701299	15,15151515	10,63829787	0	0
36	96.225	150	40	0	0	0	33,33333333	0	0	0
37	94.7	150	40	0	0	2,857142857	9,090909091	3,225806452	0	0
38	95.225	150	40	0	0	6,382978723	0	0	0	0
39	95.85	150	40	0	0	3,896103896	4,545454545	31,91489362	0	0
40	91.575	150	40	0	0	3,846153846	5,555555556	6,25	0	0
41	91.46666667	150	40	0	0	2,857142857	0	0	0	0
42	91.425	150	40	0	0	6,382978723	9,677419355	0	0	0
43	91.35	150	40	0	0	1,298701299	4,545454545	0	0	0
44	95.65	150	40	0	0	3,846153846	0	0	0	0
45	95.06666667	150	40	0	0	2,857142857	0	12,90322581	0	0



**Proyecto:** Analisis de Técnicas para Medir la Influencia de las Variables del Vector de Entrada de un Perceptron Multicapa

**Autor:** Roberto Romero Martínez

**Capítulo:** Análisis de sensibilidad

46	95,975	150	40	0	4,255319149	0	6,818181818	0
47	94,8	150	40	0	1,298701299	7,575757576	27,65957447	0
48	96,2	150	40	0	3,846153846	5,555555556	6,25	0

**Tabla 91: Resumen fortaleza frente a valores atípicos (FFVA)**

Nota: los datos de las columnas “*G.PLACA*”, “*R.PROY*”, “*L.PROY*”, “*VEL (m/s)*”, “*M. INICIAL(g)*”, “*MAT.PROY*” y “*MAT.PLACA*” representan el porcentaje de valores atípicos de cada variable en los que la calificación falló.



### 6.4.3.1. MATERIAL DE PROYECTIL, MATERIAL DE LA PLACA Y RADIO DEL PROYECTIL.

Lo primero que llama la atención al observar la tabla son los resultados obtenidos para las variables “material de proyectil”, “material de la placa” y “radio del proyectil”. Para estas variables el valor obtenido en todos los estudios es de 0. Dada la peculiaridad de la situación se ha tratado de encontrar el porqué de esta.

Después de repetir los estudios y obtener idénticos resultados en el fichero de resumen, se pasó a analizar los informes de cada uno de los estudios realizados, y ahí se hallaba la respuesta. Durante el proceso de testeo de la red, en cualquier ensayo de los estudios realizados, el valor de las variables en cuestión nunca estaba fuera del rango para el cual había sido entrenada, por tanto, nunca podrían existir valores atípicos.

N. Ensayo	Aciertos	Fallos	Total	G.PLACA	R.PROY	L.PROY	VEL(m/s)	M. INICIAL(g)	MAT.PROY	MAT.PLACA
1	38	2	40	1 0	0 0	0 0	0 0	0 0	0 0	0 0
2	38	2	40	0 0	0 0	0 0	0 0	0 0	0 0	0 0
3	37	3	40	0 0	0 0	0 0	0 0	0 0	0 0	0 0
4	39	1	40	0 0	0 0	0 0	1 0	1 0	0 0	0 0
. . .										
97	37	3	40	1 0	0 0	0 0	0 0	0 0	0 0	0 0
98	39	1	40	0 0	0 0	1 0	1 0	0 0	0 0	0 0
99	40	0	40	0 0	0 0	0 0	0 0	0 0	0 0	0 0
100	38	2	40	1 0	0 0	0 0	0 0	1 0	0 0	0 0
<b>Totales</b>	<b>3849</b>	<b>151</b>	<b>4000</b>	<b>35 0</b>	<b>0 0</b>	<b>34 1</b>	<b>40 4</b>	<b>30 1</b>	<b>0 0</b>	<b>0 0</b>
				<b>35</b>	<b>0</b>	<b>35</b>	<b>44</b>	<b>31</b>	<b>0</b>	<b>0</b>
<b>Epocs</b>	<b>10000</b>	<b>MSE</b>	<b>0,01</b>	<b>VC</b>	<b>15</b>	<b>Pruebas</b>	<b>100</b>	<b>Reg. Entrenamiento</b>	<b>150</b>	<b>383,33 s.</b>

**Tabla 92: Informe FFVA**

El que para estas variables nunca existiesen valores atípicos debía deberse a alguna situación peculiar en sus datos. Analizando los datos de entrada del estudio se obtuvieron los siguientes resultados.



**Referencias**

- Grosor de la placa → 34 valores diferentes
- Radio del proyectil → 6 “ “
- Longitud del proyectil → 43 “ “
- Velocidad del proyectil → 191 “ “
- Masa inicial del proyectil → 82 “ “
- Material del proyectil → 3 “ “
- Material de la placa → 3 “ “

A la luz de los resultados se observa que la peculiaridad reside en la distribución de los datos en estas tres variables. La distribución de sus datos es mínima, por lo que es casi imposible que la red no haya sido entrenada con todos los valores posibles de cada una de las 3 variables.

Esta es una de las grandes limitaciones de este tipo de estudio, si la distribución muestral de alguna de las variables no es lo suficientemente amplia, no se podrá obtener información de los resultados que produzca. Lo que no quiere decir que la variable sean o no críticas, sino que el estudio es incapaz de obtener ningún tipo de información a través de ellas.

#### 6.4.3.2. GROSOR DE LA PLACA

Si llamaba la atención notablemente la existencia de 3 variables sin valores atípicos, no menor es la sorpresa al ver que para la variable Grosor de la placa, pese a existir 34 valores distintos, sólo en 2 de los estudios se obtenían valores distintos de 0. Según el anterior criterio, esto significaría que el grosor de la placa no es determinante en el aprendizaje de la red, algo, que como se va a demostrar, no puede estar más lejos de la realidad.



**Proyecto:** Analisis de Técnicas para Medir la Influencia de las Variables del Vector de Entrada de un Perceptron Multicapa

**Autor:** Roberto Romero Martínez

**Referencias**

Apoyándose en los resultados de los estudios con número de Epocs igual a 1 será más fácil la exposición

Cód.	%Aciertos	Entrena.	Test	G.PLACA	R.PROY	L.PROY	VEL(m/s)	M. INI(g)	MAT.PROY	MAT.PLACA
14	78,25	150	40	9,523809524	0	59,57446809	0	34,09090909	0	0
15	75,9875	150	40	16,66666667	0	9,090909091	3,03030303	42,55319149	0	0
16	74,75	150	40	0	0	30,76923077	16,66666667	25	0	0
2	74,46666667	150	40	7,142857143	0	44,68085106	6,451612903	15,90909091	0	0
6	73,61666667	150	40	11,9047619	0	38,29787234	24,19354839	6,818181818	0	0
12	73,35	150	40	10,52631579	0	30,76923077	22,22222222	18,75	0	0
9	72,55	150	40	5,714285714	0	57,14285714	4,545454545	6,451612903	0	0
5	72,25	150	40	28,57142857	0	62,85714286	15,90909091	16,12903226	0	0
7	71,2875	150	40	76,19047619	0	61,03896104	10,60606061	0	0	0
1	70,3	150	40	22,85714286	0	57,14285714	20,45454545	0	0	0
3	70,1375	150	40	78,57142857	0	57,14285714	30,3030303	0	0	0
10	70,08333333	150	40	71,42857143	0	59,57446809	12,90322581	0	0	0
11	70,05	150	40	78,57142857	0	55,84415584	13,63636364	0	0	0
4	69,85	150	40	68,42105263	0	34,61538462	27,77777778	0	0	0
13	69,725	150	40	82,85714286	0	57,14285714	29,54545455	0	0	0
8	68,65	150	40	68,42105263	0	42,30769231	50	37,5	0	0

**Tabla 93: Informe FFVA resumen epocs**

Para los estudios cuyo número de Epocs es 1, la cantidad y el porcentaje de los valores obtenidos en el estudio es mucho mayor, algo que era de esperar al sólo presentársele una vez los datos para el entrenamiento. En la mayoría de las ocasiones los valores obtenidos para el grosor de la placa son bastante elevados, más de un 55%, o relativamente inferiores a un 15%, es decir, o la red ha aprendido una regla fiable para el grosor, y acierta generalmente, o si no falla en la mayoría de las ocasiones. Sólo 3 de los 16 valores están fuera de ese rango, llamado de ambigüedad, lo que lleva a pensar que para esta variable se puede fijar una regla sencilla y muy fiable.

Simplemente aumentado el número de Epocs la mayoría de los porcentajes bajan hasta 0. Si se comprueban los informes individualmente, se observa que se siguen produciendo valores atípicos, con la diferencia de que estos ahora raramente generan errores. ¿Qué es lo que está pasando? ¿Pasa la variable a no ser determinante al aumentar el número de Epocs?



Lo que ocurre es que al aumentar la calidad del entrenamiento, la red se da cuenta de la gran importancia del grosor de la placa en el resultado final, y se marca un regla a la que otorga gran peso para el resultado de su predicción, a mayor grosor mayor posibilidad de que el proyectil no la atravesase, y viceversa. Eso explica que ahora, por norma general, los valores atípicos en esta variable no produzcan error en la predicción. Si el valor está por encima de lo que ha sido entrenado en la mayoría de las ocasiones dirá que el proyectil no atraviesa la placa, y al contrario. Algo que ya nos advertía el estudio “Sensitivity about the mean”.

Como se ha demostrado, no parece fiable el criterio inicial para el análisis de los resultados obtenidos mediante “Fortaleza frente a valores atípicos”. Este tipo de estudio requiere un análisis mucho más complejo para poder obtener informaciones fiables.

#### 6.4.3.3. VELOCIDAD DEL PROYECTIL

Para la variable Velocidad del proyectil cuando el numero de Epocs es 1 se obtienen unos resultados bastantes más ambiguos que para el grosor de la placa, ahora son 9 de los 16 estudios los que se encuentran dentro del rango de ambigüedad. Esto indica que, en contra de lo que se podía imaginar, la red no ha encontrado un patrón claro que aplicar. Se ha analizado los datos de entrada para comprobar si existe alguna anomalía que no permita a la red obtener un patrón claro para la velocidad, y no es así, al revés, los datos son muy variados y tienen una distribución muy uniforme.

Son 4 los resultados del estudio que después de aumentar el número de Epocs obtienen un valor fuera del rango 15-55, un número a priori elevado. Pese a ver mejorado exponencialmente el entrenamiento, en determinadas ocasiones la red no encuentra una conducta clara sobre la que obtener un patrón. También es cierto que en 17 de los 32 estudios se obtienen valores inferiores al 5%, siendo 11 de un 0% y siempre existiendo valores atípicos. Este hecho muestra que la red sí puede lograr obtener una muy regla fiable para la velocidad, y que lo hace en la mayoría de las ocasiones. Es cierto que cuando no lo logra, el porcentaje de fallos que producen los



valores atípicos aumenta considerablemente, llegando incluso a ser de un 33,33%. Esto indica que la red otorga bastante importancia a esta variable, y que se debe ser muy cuidadoso con los valores, y la distribución de estos, que se le presenta a la RNA.

#### 6.4.3.4. MASA INICIAL DEL PROYECTIL

En esta variable son 7 los valores ambiguos para entrenamientos con 1 Epocs, un valor parecido al obtenido por Velocidad del proyectil, pero con la diferencia de que para la masa 7 de los valores fuera de rango son inferiores al 5%, concretamente son 0. Con estos datos, en principio, se podría asegurar que a la RNA le debe ser relativamente fácil obtener un patrón para esta variable.

Cuando aumentamos la cantidad de Epocs los resultados, como era de esperar, mejoran. Son 22 los valores inferiores a 5%, siendo 19 de ellos 0. Esto confirmaría la capacidad de la red de obtener un patrón fiable con estos datos para la variable Masa inicial del proyectil. Peso a esto, 3 resultados del análisis continúan en el rango de ambigüedad. A priori, esta situación puede resultar parecida a la acontecida en Velocidad del proyectil, donde eran 4 los valores ambiguos, pero existe una diferencia importante que radica en que desde el entrenamiento más básico para la masa del proyectil la red mostraba gran facilidad para obtener un patrón válido, o al menos, es lo que se pensaba inicialmente. ¿Por qué si la RNA tiene esa facilidad existen todavía valores ambiguos incluso cuando la red ha sido bien entrenada? ¿Por qué para el estudio “Sensitivity about the mean” otorga un importancia tan alta a la velocidad en comparación con la masa?

Estas dudas obligan a analizar los datos de entrada para la variable Masa del proyectil. Al analizar los datos se observan graves anomalías, la primera, y más importante, es que en 60 de los 220 registros la masa es 0. Además, el 75% de los valores posibles se concentra en el primer 25% de total del rango de valores, es decir, su distribución es poco uniforme. Estos datos indican que lo más probable es que la red



no esté logrando obtener un patrón fiable, como se pensaba inicialmente, y por ello, le está otorgando a la variable un peso muy bajo en la mayoría de los estudios. Si esto es cierto, ¿cómo explicar los valores ambiguos cuando la red ha sido bien entrenada? Esto ocurre cuando, debido al tipo de entrenamiento y los datos suministrados para este, presumiblemente, un entrenamiento con pocos valores a 0, la RNA cree obtener un patrón fiable para la masa, a mayor masa, mayor posibilidad de que atraviese, y le otorga un peso superior al habitual, provocando que la red falle durante el proceso de test al encontrarse pruebas con masa 0 que sí atraviesan la placa.

#### 6.4.3.5. LONGITUD DEL PROYECTIL

Con esta variable son 6 los valores que se encuentran en el rango de ambigüedad cuando el número de Epocs es 1, cabe destacar que existen 7 valores que se encuentran muy próximos al rango de ambigüedad. Todo indica que la red puede obtener un patrón con relativa facilidad.

Al aumentar el número de Epocs desaparecen todos los valores ambiguos, y 26 de los 32 valores obtienen un porcentaje inferior al 5%, aunque sólo en 3 es 0. Los datos de entrada para esta variable siguen una distribución muy uniforme. Todo esto sugiere que la RNA logra obtener un patrón fiable para la variable Longitud del proyectil.

Según “Sensitivity about the mean”, la red le otorga un peso ligeramente mayor a la masa que la longitud del proyectil. Sin embargo, la información que nos proporciona “Fortaleza frente a valores atípicos” invita a discrepar.

Todo indica que a la red le es más fácil obtener un patrón fiable para la variable longitud del proyectil que para la masa, la distribución de sus datos es más homogénea, por todo ello, lo normal es que la red otorgue un peso superior a la longitud en la mayoría de las ocasiones. Además, los resultados del estudio muestran que la masa del proyectil no tiene una influencia nula en la mayoría de las ocasiones, por eso abundan





los valores iguales a 0, mientras que la longitud del proyectil, obtiene porcentajes muy bajos en todos los casos, pero en su mayoría distintos de 0, lo que sugiere que la RNA le otorga un peso bajo, aunque superior al que le da a la masa.

#### 6.4.3.6. RESUMEN GENERAL

El orden de importancia que se deriva de los resultados del análisis del estudio “Fortaleza frente a valores atípicos” sería el siguiente:

<i>Posición</i>	<i>Variable</i>	<i>Grado de Criticidad</i>
1	Grosor de la Placa	Muy alta
2	Velocidad del proyectil	Alta
3	Longitud del proyectil	Baja
4	Masa del proyectil	Muy baja
5	Radio del proyectil	No evaluable
6	Material del proyectil	No evaluable
7	Material de la placa	No evaluable

**Tabla 94: Valoración FFVA**

Sería muy interesante mejorar los datos que se tienen de la variable Masa del proyectil, a priori algo poco costoso. Como se ha demostrado, cuando la red cree conseguir un patrón fiable para esta variable le otorga un peso reseñable, esto indica, que de lograrse, la red mejoraría su capacidad de predicción con toda seguridad.



## Capítulo 7. CONCLUSIONES

El proyecto ha logrado alcanzar los objetivos marcados al inicio del mismo:

- Acercamiento al mundo de la RNA, origen, filosofía,, estructura, utilidades, tipos ...
- Realizar un estudio para la identificación de la mejor configuración de una RNA Perceptron Multicapa para un sistema de clasificación a partir de los datos de entrada proporcionados.
- Realizar un estudio para la identificación de la mejor configuración de una RNA Perceptron Multicapa para un sistema de predicción a partir de los datos de entrada proporcionados.
- Comprender y razonar el comportamiento de una RNA a partir de análisis pormenorizado de los estudios anteriores.
- Analizar el peso de las variables de entrada con el método Sensitivity About the Mean (SAM).
- Analizar el peso de las variables de entrada con el método Fortaleza Frente a Valores Atípicos (FFVA).
- Valorar los resultados de los métodos anteriores, enfrentarlos, y determinar el grado de fiabilidad de los mismos.

Se ha creado una herramienta para automatizar el entrenamiento y testeo de las RNA, así como para la generación de informes y resúmenes de resultados. Esto ha permitido realizar de forma masiva procesos de entrenamiento y testeo en tiempos razonable. De la misma forma, los informes y resúmenes de los resultados obtenidos han permitido analizar y explicar el comportamiento de la red, como varía este en función de la configuración del entrenamiento, del testeo y de los datos de entrada.



También ha permitido implementar el método de medición de criticidad de las variables de entrada “FFVA”, y con ello, analizar sus resultados, compararlos a los obtenidos con SAM.

En el ámbito que nos ocupa, predecir las posibles consecuencias del impacto de un determinado proyectil sobre una superficie, los datos arrojados por los estudios de las simulaciones de predicción y clasificación no admiten lugar a dudas, una red de neuronas del tipo perceptron multicapa asociado al algoritmo de aprendizaje backpropagation tiene un comportamiento excelente en este tipo de procedimientos, más aún, teniendo en cuenta la calidad de los datos de entrada.

La red tiene un comportamiento notable con una configuración básica. Realizando un análisis pormenorizado de los datos de entrada, combinándolo con un proceso iterativo para la configuración de los distintos criterios de parada de forma que se evite el sobreaprendizaje se obtienen un método con un grado de fiabilidad máximo.

Como ya se ha concluido, para obtener unos buenos resultados es crucial una correcta configuración de la red, pero aún lo es más, el disponer de una muestra de datos de entrada de calidad y variada. Cuanto más exactos sean estos, más fiables serán los resultados obtenidos por la red; cuanto más variados, más patrones aprenderá la red, lo que le permitirá predecir de forma eficiente una mayor diversidad de ensayos.

La calidad de los datos también influye de forma directa en la fiabilidad de métodos para el análisis de sensibilidad de las variables de entrada. Esto influye especialmente en el proceso de “Fortaleza frente a valores atípicos”. Una de las grandes limitaciones de este tipo de estudio reside en la distribución muestral de los datos de entrada, si en alguna de las variables no es lo suficientemente amplia, no se podrá obtener información sobre valores atípicos, lo que no quiere decir que las variables sean o no críticas, sino que el estudio es incapaz de obtener ningún tipo de información a través de ellas.



## REFERENCIAS

- [1] Goldsmith W. *Impact*. The theory and physical behaviour of colliding solids. Edward Arnold Publishers Ltd., London, 1960.
- [2] Arbib, M.A. *The handbook of brain theory and neural networks*. Cambridge, Mass.: MIT Press, Ed.1995.
- [3] Simpson, P.K. *Neural networks technology and applications: theory, technology and implementations*. New York: IEEE, Ed.1995.
- [4] Arbib, M.A., Erdi, P. y Szentagothai, J. *Neural organization: structure, function and dynamics*. Cambridge, Mass.: MIT Press, 1997.
- [5] De Lillo, A. y Meraviglia, C. The role of social determinants on men's and women's mobility in Italy. A comparison of discriminant analysis and artificial neural networks. *Substance Use and Misuse*, 33(3), 751-764, 1998.
- [6] Jang, J. Comparative analysis of statistical methods and neural networks for predicting life insurers' insolvency (bankruptcy) (The University of Texas at Austin, 1997). *Dissertation Abstracts International, DAI-A*, 59/01, 228, 1998.
- [7] Waller, N.G., Kaiser, H.A., Illian, J.B. y Manry, M. A comparison of the classification capabilities of the 1-dimensional Kohonen neural network with two partitioning and three hierarchical cluster analysis algorithms. *Psychometrika*, 63(1), 5-22, 1998.
- [8] Arana, E., Delicado, P., Martí-Bonmatí, L. Validation procedures in radiologic diagnostic models. Neural network and logistic regression. *Investigative Radiology*, 34(10), 636-642, 1999.
- [9] Takahashi, K., Hayasawa, H. y Tomita, M. A predictive model for affect of atopic dermatitis in infancy by neural network and multiple logistic regression. *Japanese Journal of Allergology*, 48(11), 1222-1229, 1999.
- [10] Sarle, W.S. *Neural network FAQ*. Periodic posting to the Usenet newsgroup comp.ai.neural-nets, URL: <ftp://ftp.sas.com/pub/neural/FAQ.html>, 1998.
- [11] Rumelhart, D.E., Hinton, G.E. y Williams, R.J. Learning internal representations by error propagation. En: D.E. Rumelhart y J.L. McClelland (Eds.). *Parallel distributed processing* (pp. 318-362). Cambridge, MA: MIT Press, 1986.
- [12] Funahashi, K. On the approximate realization of continuous mapping by neural networks. *Neural Networks*, 2, 183-192, 1989.



**Referencias**

- [13] Hornik, K., Stinchcombe, M. y White, H. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5), 359-366, 1989
- [14] García-Crespo, B. Ruiz-Mezcua, D. Fernández-Fdz, R. Zaera Prediction of the response under impact of steel armours using a multilayer perceptron, *Neural Computing & Applications*, 2006.
- [15] Takehiko Ogawa, Hajime Kanada, Hideaki Kasano. *Neural Network Localization of a Steel Ball in Impact Perforation Images*, SICE Annual Conference in Fukui, August 4-6,2003 Fukui University, Japan, 2003.
- [16] S.W. Liu, Jin H. Huang,, J.C. Sung, C.C. Lee. *Detection of cracks using neural networks and computational mechanics*, *Comput. Methods Appl. Mech. Engrg.* 191 (2002) 2831–2845, 2002.
- [17] Zenon Waszczyszyn, Leonard Ziemianski. *Neural network in mechanics of structures and materials- new result and prospects of applications*, *Computers and Structures* 79 (2001) 2261-2276, 2001.
- [18] McCulloch, W. & Pitts, W. A logical calculus of ideas immanent in nervous activity. *Bull Math Biophys* 115-133, 1943.
- [19] Hebb, D. *Organization of Behaviour*. John Wiley & Sons, New York: Science Editions, 1949.
- [20] Jacek M. Zurada. *Introduction to Artificial Neural Systems.*: West Publishin, 1992.
- [21] Rosenblatt, F. The perceptron: A perceiving and recognizing automaton. *Technical Report 85-460-1, Cornell Aeronautical Laboratory*, 1957.
- [22] P. Isasi and I. Galván. *Redes De Neuronas Artificiales : Un Enfoque Práctico.*: Pearson Prentice Hal, 2004.
- [23] Minsky, M. & Papert, S. *Perceptrons: An Introduction to Computational Geometry*. The MIT Press, 1969
- [24] J.R. Hilera and J. Martínez. *Redes Neuronales Artificiales : Fundamentos, Modelos Y Aplicaciones.*: Ra-Ma, 1995.
- [25] Amari, S. Characteristics of randomly connected threshold-element networks and network systems. *Proceeding of the IEEE*, 59(1):35-47, 1971.
- [26] Amari, S. Characteristics of random nets of analog neuron-like elements. *IEEE Transactions on Systems, Man and Cybernetics*, SMC-2:643-657, 1972.



**Referencias**

- [27] Amari, S. A method of statistical neurodynamics. *Ky-bernetic*, 14:201-215, 1974
- [28] Amari, S. Dynamics of pattern formation in lateral-inhibition type neural fields. *Biological Cybernetics*, 27:77-87, 1977.
- [29] Amari, S. and Takeuchi, M. Mathematical theory on formation of category detecting nerve cells. *Biological Cybernetics*, 29:127-136, 1978.
- [30] Amari, S. Field theory of self-organization neural nets. *IEEE Transaction on Systems, Man and Cybernetics*, SMC-13:741-748, 1983.
- [31] Amari, S. *A mathematical theory of self-organizing nerve systems*. Biomathematics: Current Status and Perspectives. North-Holland, Amsterdam, 1982.
- [32] Anderson, J. A., J. W. Silverstein, S. A. Rite, and R. S. Jones. Distinctive Features, Categorical Perception, and Probability Learning: Some Applications of a Neural Model, *Psychological Review*. 84: 413-451, 1977.
- [33] Fukushima, K. And S. Mikaya. Neocognitron: A Self-Organizing Neural Network Model for a Mechanism of Pattern Recognition Unaffected by Shift in Position. *Biological Cybernetics*. 36(4): 193-202, 1980.
- [34] Hopfield, J. J. Neural Networks and Physical Systems with Emergent Collective Computational Abilities, *Proceedings of the National Academy of Science USA*. 79: 2554-58, 1982.
- [35] Hopfield, J. J.. Neurons with Graded Response Have Collective Computational Properties Like Those of Two State Neurons, *Proceedings of the National Academy of Science USA* 8: 3088-3092, 1984..
- [36] McClelland, T. L. , D. E. Rumelhart. *Parallel Distributed Processing*. Cambridge: MIT Press and PDP Research Group, 1982.
- [37] R.Hecht-Nielsen. Neurocomputing: Picking the Human Brain. *IEEE Spectrum*, 25, pags. 36-41, Marzo, 1988.
- [38] R.Hecht-Nielsen. Applications of Counterpropagation Networks. *Neural Networks*, 1, pags. 131-140, 1988.
- [39] A. J. Maren , C.T. Harston y R. M. Pap. *Handbook of Neural Computing Applications*. Ed. Academic Press, 1990.
- [40] José C. Principe, Neil R. Euliano, W. Curt Lefebvre. *Neural and Adaptive Systems*, 1999.



**Referencias**

- [41] Werbos, P. Beyond regression: New tools for prediction and analysis in the behavioural sciences. Tesis doctoral. Harvard University, 1974.
- [42] Parker, D. *Learning logic*. Invention Report, S81-64, File 1. Office of Technology Lising, Stanford University, 1982.
- [43] Le Cun, Y. *Modèles connexionnistes de l'apprentissage*. Tesis doctoral. Université Pierre et Marie Curie, París VI, 1985.
- [44] Rzepoluck, E.J. *Neural network data analysis using Simulnet*. New York: Springer-Verlag, 1998.
- [45] Rosenblatt, F. The Perceptron: A Theory of Statistical Separability in Cognitive Systems. *Technical Report VG-1196-G-1, Cornell Aeronautical Laboratory*, 1958.
- [46] Widrow, B. (1990) 30 years of adaptive neural networks: perceptron, madaline, and backpropagation. *Proceedings of the IEEE* 78 9 1415-1442.
- [47] Martín del Brío, B. y Sanz, A. (1997). *Redes neuronales y sistemas borrosos*. Madrid: Ra-Ma.
- [48] Alejandro Moreno Torres (2008): *Aplicación de un perceptrón multicapa para resolución de problemas de impacto balístico*
- [49] David Martín Muñoz (2007): *Estudio de la viabilidad de la aplicación de las redes de neuronas artificiales para la resolución del problema del impacto sobre materiales*
- [50] Fernando Calderón Valiente (2005): *Simulación de impactos en materiales mediante redes neuronales*.
- [51] Diego Terroba Jarque (2007): *Parametrización del aprendizaje de una red neuronal para la predicción de impactos sobre materiales*.