

Desarrollo de nuevos algoritmos neurogliales que modelizan la interacción astrocito-neurona en sistemas de altas prestaciones

Francisco Abel Cedrón Santaefemia

Tesis doctoral UDC / 2019

Directora: Ana Belén Porto Pazos

Programa de doctorado en Tecnologías de la Información y las Comunicaciones



UNIVERSIDADE DA CORUÑA

Dña. Ana Belén Porto Pazos profesora titular del área de Ciencias de la Computación e Inteligencia Artificial de la Universidade da Coruña,

AUTORIZA:

La presentación para su depósito de la tesis que dirige y que fue realizado por D. Francisco Abel Cedrón Santaefemia con DNI 33549761-Y con título “Desarrollo de nuevos algoritmos neurogliales que modelizan la interacción astrocito-neurona en sistemas de altas prestaciones”.

Y para que así conste, firma esta autorización en A Coruña, a 12 de Septiembre de 2018

La directora de la tesis

Fdo.: Ana B. Porto Pazos

*A todas esas personas que han perdido parte
de su valioso tiempo en enseñarme cosas nuevas*

Agradecimientos

En primer lugar, agradecer a mi directora de tesis la Dra. Ana Belén Porto Pazos por su buen trato desde el primer día, por sus sabios consejos y la dedicación prestada durante las casi infinitas horas de tutorías sin las cuales difícilmente habría sido posible la realización de esta tesis.

A Carlos Fernández Lozano y Adrián Carballal Mato, por las *masterclass* que me han dado sobre el mundo de la investigación y de las cuales he aprendido tanto.

Por supuesto, quiero agradecer a todos los miembros del laboratorio RNASA-IMEDIR por la buena acogida que tuve desde el primer día, el cual no habría sido posible sin el Dr. Marcos Gestal Pose ayudándome a dar mis primeros pasos hacia el mundo de la investigación.

También me gustaría agradecer al grupo RNASA-IMEDIR por todos los recursos que he podido utilizar para la presente tesis:

- Proyecto “GERIA-TIC”, Proyecto cofinanciado por GAIN- Conecta PEME (3ª edi) (IN852A 2016/10) y fondos FEDER UE
- “Proyecto Colaborativo de Integración de datos Genómicos” (CICLOGEN) PI17/01826, “Técnicas de data mining y docking molecular para análisis de datos integrativos en cáncer de colon” del MINECO.
- “Red Gallega de Investigación en Cáncer Colorrectal” (REGICC) ED431D2017/23, “Red Gallega de medicamentos” (REGID) ED431D2017/16
- Ayudas para la Consolidación y estructuración de unidades de investigación competitivas del SUGA-Xunta de Galicia y Centros Singulares (ED431G/01) dotados de fondos FEDER de la UE

Además, también me gustaría agradecer al Centro de Supercomputación de Galicia (CESGA) por toda la inmensa cantidad de horas que he podido usar de su infraestructura.

Por último, cabe destacar que esta tesis ha sido posible por las ayudas a la etapa predoctoral financiadas por la Unión Europea a través del Fondo Social Europeo en un 80% y el 20% restante de la Secretaría Xeral de Universidades, perteneciente a la Consellería de Cultura, Educación y Ordenación Universitaria de la Xunta de Galicia.

“La ciencia es simplemente sentido común en su mejor momento”

Thomas Henry Huxley. (1825-1895)

Biólogo Británico

Resumo

Estamos vivindo una era de evolución constante debido aos avances tecnolóxicos. Moitos deles están sendo posibles grazas á Intelixencia Artificial (IA) e a que se están almacenando grandes volumes de datos. Pódense construír ferramentas con componentes intelixentes que están revolucionando diversos campos grazas á gran cantidade de datos que poden ser analizados mediante devanditos modelos intelixentes. Sen embargo, moitas investigación só céntranse na cantidade e a calidade dos datos dos que se dispón, e apenas se realizan esforzos en mellorar as propias técnicas de IA.

A proposta desta tese é a mellora de sistemas intelixentes conexionistas que ata hai pouco estaban formados por Redes de Neuronas Artificiais (RNA). Para levar a cabo tan ambiciosa labor, considerouse, do mesmo xeito que ocorreu con investigación noutras áreas, ter en conta como resolve o problema a natureza. Para iso, centrarémonos na estrutura máis complexa e máis eficiente coñecida polo home, o cerebro humano. Con ese fin, é necesario apoiarse no campo da Neurociencia onde se pode tratar de levar os avances que se descubren ou as hipóteses que se xeran ao campo da IA.

O núcleo desta tese vira contorna as investigación que evidencian que as neuronas non son os únicos elementos do cerebro humano que participan no procesamento da información, si non que os astrocitos do Sistema Glial (SG) xogan un papel esencial. De feito, a comunicación sináptica sábese xa que se produce con participación de neuronas e astrocitos, o cal coñécese como sinapsis tripartita. Isto levou á inclusión de novos elementos de procesado nas RNA que simulan o comportamento das células gliales creando as Redes NeuroGliais Artificiais (RNGA).

Para demostrar a utilidade dos astrocitos artificiais e colaborar en demostrar a capacidade do SG, leváronse a cabo novos algoritmos de modulación astrocítica que se puxeron a proba en diferentes problemas de clasificación e regresión, obténdose resultados significativos con respecto a redes sen astrocitos. Ademais, desenvolveuse graza a esta tese una aplicación web opensource para que a comunidade científica poida usar estas redes libremente.

Resumen

Estamos viviendo una era de evolución constante debido a los avances tecnológicos. Muchos de ellos están siendo posibles gracias a la Inteligencia Artificial (IA) y a que se están almacenando grandes volúmenes de datos. Se pueden construir herramientas con componentes inteligentes que están revolucionando diversos campos gracias a la gran cantidad de datos que pueden ser analizados mediante dichos modelos inteligentes. Sin embargo, muchas investigaciones solo se centran en la cantidad y la calidad de los datos de los que se dispone, y apenas se realizan esfuerzos en mejorar las propias técnicas de IA.

La propuesta de esta tesis es la mejora de sistemas inteligentes conexionistas que hasta hace poco estaban formados por Redes de Neuronas Artificiales (RNA). Para llevar a cabo tan ambiciosa labor, se ha considerado, al igual que ha ocurrido con investigación en otras áreas, tener en cuenta cómo resuelve el problema la naturaleza. Para ello, nos centraremos en la estructura más compleja y más eficiente conocida por el hombre, el cerebro humano. Con ese fin, es necesario apoyarse en el campo de la Neurociencia donde se puede tratar de llevar los avances que se descubren o las hipótesis que se generan al campo de IA.

El núcleo de esta tesis gira entorno las investigaciones que evidencian que las neuronas no son los únicos elementos del cerebro humano que participan en el procesamiento de la información, si no que los astrocitos del sistema glial juegan un papel esencial. De hecho, la comunicación sináptica se sabe ya que se produce con participación de neuronas y astrocitos, lo cual se conoce como sinapsis tripartita. Esto llevó a la inclusión de nuevos elementos de procesado en las RNA que simulan el comportamiento de las células gliales creando las Redes NeuroGliales Artificiales (RNGA).

Para demostrar la utilidad de los astrocitos artificiales y colaborar en demostrar la capacidad del Sistema Glial (SG), se han llevado a cabo nuevos algoritmos de modulación astrocítica que se han puesto a prueba en diferentes problemas de clasificación y regresión, obteniéndose resultados significativos con respecto a redes sin astrocitos. Además, se ha desarrollado gracias a esta tesis una aplicación web opensource para que la comunidad científica pueda usar estas redes libremente.

Abstract

We are living an era of constant evolution due to technological advances. Many of them are being possible thanks to Artificial Intelligence (AI) and the large volumes of data are being stored. You can build tools with intelligent components that are revolutionizing various fields thanks to a large amount of data that can be analyzed by these intelligent models. However, many investigations only focus on the quantity and quality of the data available, and little effort is made to improve the AI techniques themselves.

The proposal of this thesis is the improvement of connectionist intelligent systems that until recently were formed by Artificial Neural Networks (ANN). To carry out such ambitious work, it has been considered, as has happened with research in other areas, to consider how nature solves the problem. For this, we will focus on the most complex and efficient structure known, the human brain. To that end, it is necessary to rely on the field of Neuroscience where one can try to take the advances that are discovered or the hypotheses that are generated in the AI field.

The core of this thesis is based on research that shows that neurons are not the only elements of the human brain that participate in the processing of information. Astrocytes of the glial system play an essential role in the treatment of information. In fact, it is known that synaptic communication occurs with the participation of neurons and astrocytes, which is known as tripartite synapses. This led to the inclusion of new elements that simulate the behaviour of glial cells in ANN. The addition of the new elements, artificial astrocytes, originated the Artificial NeuroGial Networks (ANGN).

To demonstrate the usefulness of artificial astrocytes and collaborate in demonstrating the capacity of the Glial System (GS), new algorithms of astrocytic modulation have been used. These algorithms have been tested in different classification and regression problems, obtaining significant results with respect to networks that do not use the GS. In addition, an open source web application has been developed thanks to this thesis so that the scientific community can use these networks freely.

Tabla de contenidos

1.	Introducción	1
1.1.	Contexto	1
1.2.	Breve descripción del problema	2
1.3.	Objetivos	3
1.3.1.	Objetivo general	3
1.3.2.	Objetivos específicos	3
1.4.	Hipótesis	3
1.5.	Estructura de la tesis	4
2.	Fundamentos	7
2.1.	Fundamentos biológicos	7
2.1.1.	Sistema Nervioso	7
2.1.2.	Sistema Glial	8
2.1.3.	Astroцитos	10
2.1.4.	Sinapsis Tripartita	11
2.2.	Fundamentos computacionales	13
2.2.1.	Redes de Neuronas Artificiales	14
2.2.2.	Algoritmos genéticos	18
3.	Antecedentes	23
3.1.	Antecedentes biológicos	23
3.1.1.	Excitabilidad de los astroцитos	23
3.1.2.	Receptores de endocannabinoides en astroцитos	24
3.2.	Antecedentes Computacionales: Redes NeuroGliales Artificiales	27

3.2.1.	Metodología	28
3.2.2.	Verificación.....	33
4.	Solución propuesta	35
4.1.	Implementación de la herramienta de simulación de RNGA	35
4.1.1.	Búsqueda automática de los parámetros gliales.....	36
4.1.2.	Rendimiento	40
4.1.3.	Herramientas auxiliares	51
4.2.	Algoritmos astrocíticos	52
4.2.1.	Potenciación	53
4.2.2.	Depresión	56
4.2.3.	Receptores de endocannabinoides.....	56
4.3.	Metodología	58
4.3.1.	Metodología del software.....	58
4.3.2.	Metodología de diseño y construcción de los sistemas conexionistas.....	58
4.4.	Validación de resultados	59
4.5.	Análisis estadístico de los datos	60
4.5.1.	Test no paramétrico de Wilcoxon de rangos signados.	61
4.6.	Entorno de simulación	62
4.6.1.	Herramienta web.....	67
5.	Experimentación y resultados	75
5.1.	Entorno de simulación.....	75
5.1.1.	Problemas de clasificación	77
5.1.2.	Problemas de regresión	79

5.2. Experimentación y resultados en problemas de clasificación	80
5.2.1. Análisis de astrocitos artificiales en redes con 1 capa oculta.....	80
5.2.2. Análisis de astrocitos artificiales en redes con 2 capas ocultas.....	81
5.2.3. Análisis de astrocitos artificiales en redes con 3 capas ocultas.....	82
5.2.4. Comparativa global del uso de RNGA en problemas de clasificación.....	83
5.3. Experimentación y resultados en problemas de regresión	86
5.3.1. Análisis de astrocitos artificiales en redes con 1 capa oculta.....	86
5.3.2. Análisis de astrocitos artificiales en redes con 2 capas ocultas.....	87
5.3.3. Análisis de astrocitos artificiales en redes con 3 capas ocultas.....	87
5.3.4. Comparativa global del uso de RNGA en problemas de regresión.....	88
5.4. Discusión	90
6. Conclusiones.....	97
7. Conclusions.....	103
8. Futuros Desarrollos.....	107
9. Referencias	111
Apéndice A. Resultados obtenidos en la experimentación	123
A.1 . Experimentación y resultados en problemas de clasificación	123
A.1.1. Análisis de astrocitos artificiales en redes con 1 capa oculta	123
A.1.2. Análisis de astrocitos artificiales en redes con 2 capas ocultas	132
A.1.3. Análisis de astrocitos artificiales en redes con 3 capas ocultas	141
A.2.1. Análisis de astrocitos artificiales en redes con 1 capa oculta	150
A.2.2. Análisis de astrocitos artificiales en redes con 2 capas ocultas	159
A.2.3. Análisis de astrocitos artificiales en redes con 3 capas ocultas	171

Apéndice B. Aplicación de simulación de RGA y RNA	181
B.1. Ficheros de entrada	181
B.1.1. Poblaciones de individuos	181
B.1.2. Fichero de parámetros	183
B.2. Fichero de salida	190
B.2.1. Fichero de Log	190
B.3. Diccionario de nodos xml	193
B.3.1. Algoritmos de selección de individuos.....	193
B.3.2. Algoritmos de mutación de individuos	194
B.3.3. Algoritmos de cruce de individuos	194
B.3.4. Algoritmos de reemplazo de individuos.....	195
B.4. Diccionario de identificadores	195
B.4.1. Algoritmo genético	196
B.4.2. Redes Neuronales.....	197
Apéndice C. Abreviaturas	199
Apéndice D. Listado de figuras	201
Apéndice E. Listado de tablas	205

1. Introducción

El propósito de esta introducción es realizar una breve descripción del problema tratado en esta tesis doctoral y del contexto en el que se enmarca. Se definen los objetivos perseguidos por esta investigación y la hipótesis de partida de la misma. Por último, se resume el contenido de cada uno de los capítulos en los que está estructurado el presente documento.

1.1. Contexto

Desde los inicios de la Neurociencia se ha establecido como verdad incuestionable que las neuronas son los elementos más nobles del cerebro y que son los únicos responsables de cómo se procesa la información. El resto de elementos que se encuentran dentro del cráneo se pensaba que estaban para ayudar a las neuronas, tratándose de elementos de soporte estructural o elementos tróficos. De hecho, cuando el tejido glial fue descubierto en los años 50 por Rudolf Virchow lo consideró como un pegamento nervioso (la palabra proviene del griego *γλία* que significa pegamento) (Virchow & CHANCE, 1860). Para Virchow las células gliales eran elementos que no tenían ninguna función relevante dentro del cerebro. Sin embargo, en 1891 Santiago Ramón y Cajal fue capaz de diferenciar las células gliales dentro de la maraña de células que se encontraban en el cerebro y las clasificó como parte del sistema nervioso (De Felipe, 2005). Aunque Ramón y Cajal pensaba que las células gliales tomaban algún papel dentro del procesamiento de información, no fue capaz de demostrarlo y las células gliales fueron consideradas durante muchos años como elementos estructurales, tróficos y metabólicos de las neuronas.

En los últimos años, gracias a los grandes avances de la ciencia, se ha podido estudiar más en detalle las células que componen el cerebro y no resulta extraño que parte de esas investigaciones se centraran en el sistema glial ya que ocupa casi el 50% del volumen total del cerebro humano (Verkhatsky & Butt, 2013b). Derivado de las investigaciones que se han realizado, se pudo demostrar que los astrocitos tienen un papel activo en el procesamiento de la información modulando la plasticidad y el paso de neurotransmisores durante la sinapsis a través de una comunicación existente entre neuronas y astrocitos (Perea, Navarrete, & Araque, 2009). Esto supuso un gran punto de inflexión en el mundo de la Neurociencia ya que cambiaba sus teorías de cómo el cerebro procesa la información, ya que ahora entra el juego el sistema glial, que se coordina con las neuronas para coordinar los procesos cognitivos.

Aunque se están logrando avances sobre el papel de las células gliales, aún queda mucho por descubrir sobre cómo los astrocitos influyen en el cerebro para el tratamiento y procesamiento de información. Los estudios sobre el papel que éstos

desempeñan se están haciendo en su mayoría mediante la experimentación en los laboratorios de Neurociencia. Sin embargo, debido a la mayor potencia computacional de los ordenadores actuales, se está empleando cada vez más la modelización de los circuitos nerviosos y dado que se está obteniendo mucha información sobre el funcionamiento, se puede colaborar en el estudio desde el punto de vista de la modelización computacional. Una de las posibilidades de esta modelización es mediante la rama conexionista de la IA. Esta investigación trae beneficios al campo de la Neurociencia, pero también abre una perspectiva muy importante en el campo de las ciencias de la computación, para observar si los sistemas conexionistas que integran nuevos elementos de procesamiento pueden resolver problemas de un modo más eficaz. Este punto de vista puede llegar a crear una simbiosis perfecta entre la Neurociencia y la IA (Haykin, 2009).

1.2. Breve descripción del problema

Los sistemas conexionistas son una rama de la IA que trata de simular el comportamiento del cerebro mediante las Redes de Neuronas Artificiales (RNA) que tratan de simular el funcionamiento de las neuronas del cerebro (Pastur-Romay, Porto-Pazos, Cedron, & Pazos, 2017). Las RNA permiten resolver problemas del mundo real que no son abordables mediante el uso de la informática clásica o mediante la rama simbólica de la IA como pueden ser problemas clasificación, regresión, agrupamiento (clustering) o predicción.

Desde que McCulloch y Pitts hicieron aportaciones fundamentales para la generación de la rama conexionista se ha podido avanzar en el diseño de las RNA (McCulloch & Pitts, 1943). Sin embargo, todos estos avances aún presentan ciertas limitaciones en el paradigma del procesamiento de información (Freeman, 1991; Lapedes & Farber, 1988). Algunos investigadores tratan de suplir las carencias de las RNA mediante la creación de modelos matemáticos que controlen su funcionamiento (Buhmann, 2003; Cortes & Vapnik, 1995; Dorado, 1999; Haykin, 2009; Porto, Araque, Rabuñal, Dorado, & Pazos, 2007; Xu & Wunsch, 2008). Por otro lado, hay otros investigadores que dicen que las limitaciones se deben a que estos obstáculos podrían ser ocasionadas por que las RNA no reflejan los comportamientos de las neuronas dentro del Sistema Nervioso (SN) (Sudheer & Jain, n.d.). Dado a que las RNA pretenden simular el SN, no es descabellado pensar que al acercar su diseño hacia uno que represente más fielmente el modelo biológico se tendrán mejoras significativas. Una representación más realista puede cumplir el objetivo de la creación de redes con un comportamiento más inteligente y que sean capaces de abordar problemas más complejos de una manera más eficiente.

Ya que durante las últimas décadas la Neurociencia ha desvelado un papel muy importante del Sistema Glial (SG), y más concretamente el relevante papel de los astrocitos, se ha considerado la importancia de que los SC incorporen nuevos elementos que simulen las células del SG. Se ha creado un modelo más realista: se da paso a las Redes NeuroGliales Artificiales (RNGA). La idea es que a las RNA actuales se le añadan astrocitos artificiales que interactúen con las neuronas artificiales tratando de emular con mayor fidelidad lo que sucede biológicamente en el SN. Para ello se estudian en este trabajo de tesis los diferentes comportamientos que tienen los astrocitos biológicos, lo que permite implementar algoritmos neurogliales diferentes de los existentes hasta ahora que modulen a las neuronas artificiales en el procesamiento de información. Todos estos nuevos algoritmos neurogliales artificiales se prueban en diferentes problemas de clasificación y por primera vez, en problemas de regresión, para comparar su eficacia y eficiencia con el uso de sistemas conexionistas sin astrocitos artificiales que tratarán de resolver los mismos problemas.

Además, se ha desarrollado una herramienta web de software libre con librerías que contienen todo lo relacionado con el funcionamiento de una RNGA. Toda la comunidad científica podrá de este modo utilizar las RNGA para analizar sus datos.

1.3. Objetivos

1.3.1. Objetivo general

El objetivo general de esta tesis es analizar cómo el sistema glial y en concreto los astrocitos artificiales influyen en el procesamiento de información en las computadoras.

1.3.2. Objetivos específicos

- Analizar la influencia de los astrocitos artificiales en las RNGA mediante el uso de nuevos y diferentes algoritmos astrocíticos.
- Comprobar la utilidad de las RNGA en la resolución de problemas de clasificación y regresión.
- Analizar el uso de las RNGA en redes con distintas arquitecturas.
- Implementar la primera herramienta opensource para las simulaciones y obtención de resultados con RNGA.

1.4. Hipótesis

Resultados previos a esta tesis muestran que simulando el comportamiento biológico observado en los circuitos neurogliales se puede superar a los sistemas

conexionistas entrenados mediante algoritmos genéticos simples en problemas de clasificación (Porto, 2004; Veiguela, 2015).

Se piensa que la mejora de eficiencia podría darse de modo generalizado en otro tipo de problemas e implementando otros comportamientos que se dan en el sistema glial, ya que se ha demostrado que los astrocitos tienen más influencia en los procesos cognitivos más complejos (Pereira & Furlan, 2010). Además, se pueden usar distintos algoritmos neurogliales en las RNGA, si consideramos que los astrocitos biológicos influyen de distintas maneras sobre las sinapsis neuronales y aún a día de hoy no se conoce cómo proceden para actuar en cada situación.

Se puede considerar que la hipótesis de esta tesis doctoral es: “Los astrocitos participan en el tratamiento y procesamiento de la información influyendo en la sinapsis neuronal haciendo que ésta se potencie, se decremente o combine ambas conductas que al simularlas en los sistemas conexionistas mejorarán su funcionamiento en problemas de clasificación y regresión”.

1.5. Estructura de la tesis

Esta tesis doctoral se compone de los siguientes capítulos:

- **Introducción.** Se trata de la presente Introducción, en la cual se contextualiza el trabajo desarrollado junto con los objetivos y la hipótesis de la tesis doctoral.
- **Fundamentos** En este capítulo se explican las nociones fundamentales para comprender el resto del contenido de la tesis. Dado el carácter multidisciplinar de esta investigación se divide en los siguientes apartados:
 - **Fundamentos biológicos.** En esta sección se detallan las bases biológicas en las que se apoya la investigación.
 - **Fundamentos computacionales.** En este subapartado se exponen las bases computacionales.
- **Antecedentes.** En esta sección se exponen los antecedentes que han servido de base a esta investigación. La sección se divide en dos apartados:
 - **Antecedentes biológicos.** En este apartado se exponen los antecedentes biológicos que se han tenido en cuenta a la hora de realizar la presente tesis.
 - **Antecedentes Computacionales: Redes NeuroGliales Artificiales.** En este apartado se indican los antecedentes computacionales en que se ha basado la investigación llevada a cabo.
- **Solución propuesta.** En este capítulo se indican diferentes aspectos que se tuvieron en cuenta para desarrollar la investigación y el software

empleado. Se indican detalles de optimización de código e implementación de algoritmos gliales, entre otros aspectos.

- **Experimentación y resultados.** En este capítulo se detalla toda la experimentación llevada a cabo con los diversos problemas que se emplearon para evaluar los algoritmos propuestos y los resultados que se obtuvieron al realizar las pruebas.
- **Conclusiones.** Se exponen las conclusiones derivadas de la investigación realizada en esta tesis.
- **Futuros Desarrollos.** Se indican las futuras líneas de trabajo que seguirán a partir del desarrollo de esta tesis doctoral.
- **Referencias.** En este capítulo se indica la lista de referencias bibliográficas.
- **Apéndices:** Se ha incluido una sección de apéndices con información relevante sobre el trabajo realizado que se divide en los siguientes apartados.
 - **Apéndice A. Resultados obtenidos en la experimentación.** En este apéndice se muestran los datos obtenidos durante la experimentación para los 6 problemas analizados con las tres arquitecturas de red empleadas.
 - **Apéndice B. Aplicación de simulación de RNGA y RNA.** En esta sección se indica cómo funciona la aplicación de simulación de redes y se detalla cómo son los archivos de configuración para mostrar todas las opciones que dispone la herramienta desarrollada.
 - **Apéndice C. Abreviaturas.** En este apéndice se han añadido la lista de abreviaturas que aparecen en la presente tesis.
 - **Apéndice D. Listado de figuras.** En esta sección se muestra el listado de todas las figuras que forman parte de este documento.
 - **Apéndice E. Listado de tablas.** En este último apéndice está el listado de las tablas que hay en esta tesis doctoral.

2. Fundamentos

En este capítulo se indican cuáles son los fundamentos necesarios para comprender el contenido de la presente tesis. Dado que este es un trabajo multidisciplinar se exponen los fundamentos biológicos y computacionales.

2.1. Fundamentos biológicos

2.1.1. Sistema Nervioso

El Sistema Nervioso (SN) es un conjunto organizado de células que permite, a grandes rasgos, que los animales capten y procesen rápidamente las señales que pueden provenir de la interacción de su entorno o de su propio cuerpo (Noback, Strominger, Demarest, & Ruggiero, 2005). Toda esa información que recibe se necesita poder procesarla y el SN se encarga de coordinar sus elementos para que ésta se procese y genere una respuesta, que puede ser voluntaria o refleja.

El SN se clasifica en dos partes, el Sistema Nervioso Central (SNC) y el Sistema Nervioso Periférico (SNP). En los animales vertebrados el SNC está formado por la médula espinal y el encéfalo. Éste a su vez se divide en el cerebro, cerebelo, bulbo raquídeo y el tallo encefálico. El SNP se divide en el sistema periférico visceral, que contiene los nervios que residen o se encuentran en los órganos internos, los vasos sanguíneos y las glándulas, y el sistema periférico somático que se compone de los nervios que inervan la piel, los músculos y las articulaciones.

Gracias a la técnica creada por Camillo Golgi, donde se tenía un porcentaje de neuronas con una solución de cromado de plata, se hizo posible poder observar las neuronas con mayor detalle y permitió diferenciar las partes que conformaban la estructura de la neurona pudiéndose diferenciar las dendritas y el axón (figura 1) (Nicholls, 2001; Torres-Fernández, Yepes, & Gómez, 2007).

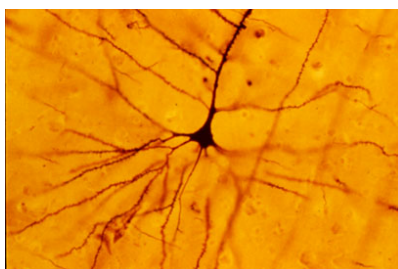


Figura 1. Neurona teñida por la técnica de Golgi

Gracias a esta técnica, Santiago Ramón y Cajal pudo estudiar el SN para comprenderlo mejor. Aunque ambos usaron las mismas técnicas, sus teorías diferían

ya que Ramón y Cajal identificó la neurona como elemento básico que conforma una red, mientras que Golgi indicaba que todo era un sistema continuo (Mazzarello, 2007). Aunque ambos tuvieron teorías distintas, se mantuvieron como relevantes en la comunidad científica y al final acabaron compartiendo el premio nobel de medicina en 1906 y de sus estudios surgió la Neurociencia moderna (Bear, Connors, & Paradiso, 2007; Glickstein, 2006).

Desde esos primeros estudios se pudo avanzar mucho analizando y observando el cerebro y las células que componen el SN. Con esto se definió la estructura que conocemos actualmente de las neuronas que se componen de tres partes (figura 2): las dendritas, el soma y el axón. Las dendritas son las ramificaciones de la neurona por las que recibe los estímulos y la información proveniente de los axones de otras neuronas. El soma representa el cuerpo celular y es donde se encuentra el núcleo de la célula. El axón es la parte de mayor longitud de la célula y es el que cumple la función de transmitir el potencial de acción a otras neuronas o células del cuerpo.

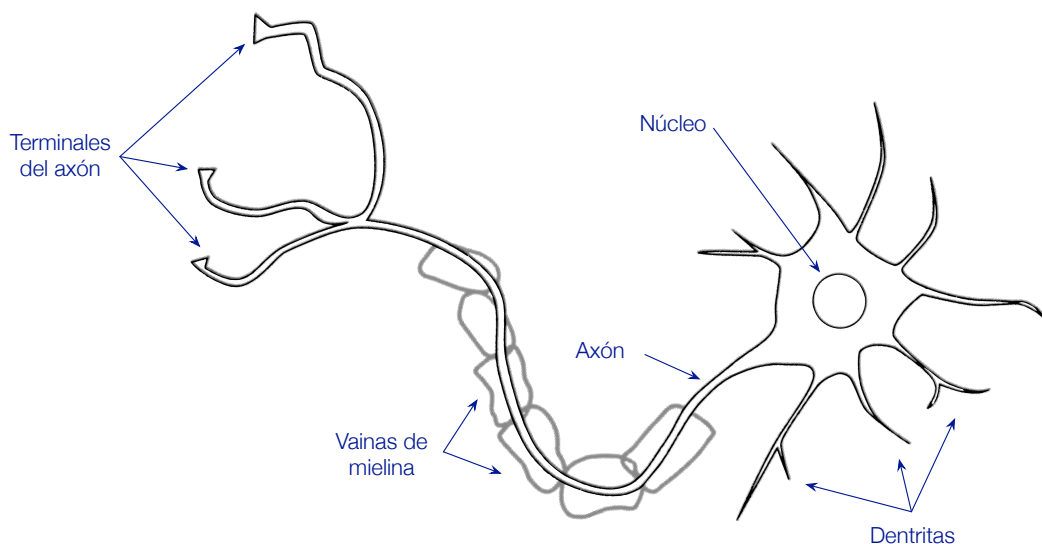


Figura 2. Estructura de una neurona biológica

2.1.2. Sistema Glial

Las células gliales o glía forman parte del SN. La primera descripción de este tipo de células fue realizada por Rudolf Virchow que le atribuyó al sistema glial la función de pegamento. No fue hasta el siglo XX que se descubrió que las células gliales tienen más funciones y se llegó a clasificar las neuronas.

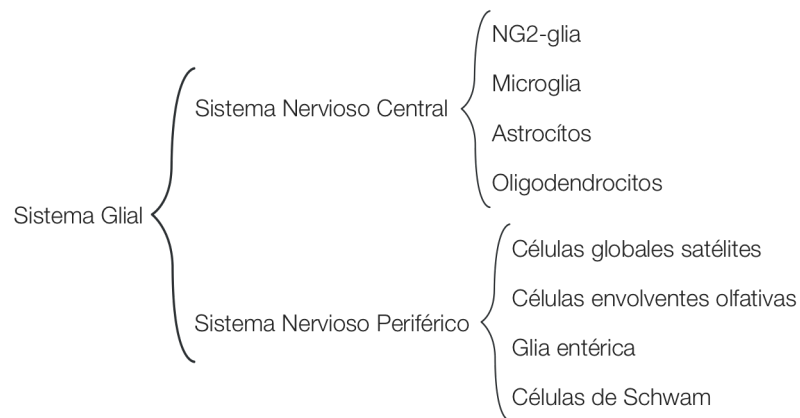


Figura 3. Clasificación del Sistema Glial

La glía en el SN de los mamíferos se divide dentro del SNP y el SNC (figura 3). Las células gliales del SNP incluyen: Células de Schwann mielinizantes que mielinizan los axones periféricos, las células de Schwann no mielinizantes que rodean múltiples axones no mielinizantes y las células de Schwann perisináptica que envuelven las sinapsis periféricas, como por ejemplo las uniones neuromusculares (Verkhratsky & Butt, 2013b). Las células gliales del SNP que rodean a las neuronas en los ganglios periféricos se conocen como células gliales satélites, y las del sistema olfativo que se conocen como células envoltantes olfativas. Por último, el SNP también incluye la glía entérica que regula la movilidad gastrointestinal (Gershon & Taube, 1991). Las células gliales del SNC se subdividen en astrocitos, oligodendrocitos, NG2-glia y microglia. Los astrocitos son los tipos de células más comunes del sistema glial. Los oligodendrocitos son las células que pueden recubrir los axones de las neuronas para mielinizarlas y permitir que los potenciales de acción se transmitan de manera eficiente (Verkhratsky & Butt, 2013d). La NG2-glia se consideran que son las precursoras de las neuronas, oligodendrocitos y los astrocitos, aunque todavía no está del todo clara la función que estas desempeñan (Verkhratsky & Butt, 2013c). Por último, la microglia es la que se encarga del sistema inmune en el SNC (Verkhratsky & Butt, 2013a).

Las células importantes para esta tesis son las que se encuentran en el SNC que es la parte del SN que se encarga de procesar la información (Perea & Araque, 2010). Dentro de las células de la glía las que se pueden considerar más relevantes son los astrocitos, también llamados astroglia, que son las más numerosas de las células gliales en los organismos más evolucionados de la escala filogenética (Verkhratsky & Butt, 2013b).

2.1.3. Astrocitos

Durante mucho tiempo se pensó que las neuronas eran las únicas células que tenían el papel de transmitir y procesar la información ya que se consideró que eran las únicas células excitables. Sin embargo, a finales del siglo XX se empezaron a realizar los primeros descubrimientos sobre la excitabilidad de los astrocitos gracias a los avances en los marcadores fluorescentes y las técnicas de imagen (Charles, Merrill, Dirksen, & Sandersont, 1991; A H Cornell-Bell, Finkbeiner, Cooper, & Smith, 1990). Gracias a este descubrimiento se dejó de pensar que los astrocitos tenían una función pasiva dentro del SN lo que supuso una revolución en el estudio del cerebro.

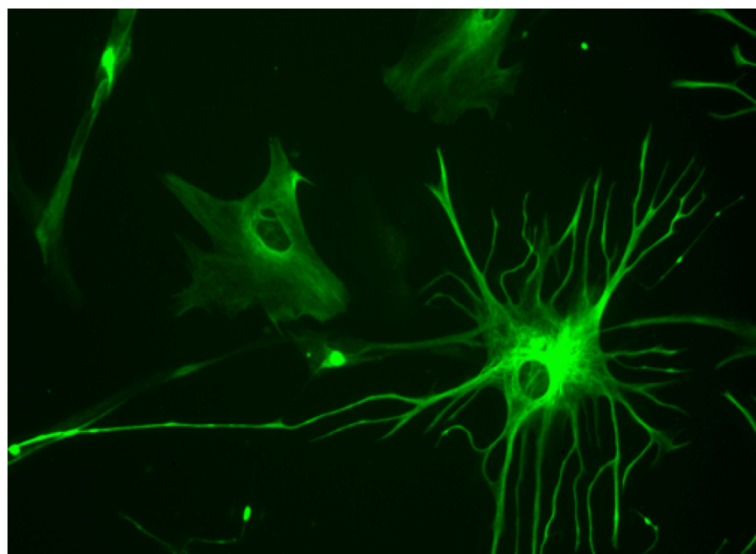


Figura 4. Astrocito humano del cerebro de un feto de 23 semanas

En general hay menos células gliales que neuronas, sobre 86 billones de neuronas y 85 billones de células gliales en el cerebro humano, aunque estas últimas representan la mitad del volumen del cerebro y de la médula espinal (Azevedo et al., 2009). Lo que es de importante relevancia es que la relación glía/neurona varía en función de la parte del cerebro, en la corteza cerebral es de 3,7 (60.84 billones son células gliales y 16,34 son neuronas), mientras que por ejemplo en el cerebelo es de solo 0.2 (16,04 billones de células gliales y 69,03 billones de neuronas) (Azevedo et al., 2009).

A través de diversos estudios se pudo ver que se puede estimular la excitabilidad de los astrocitos usando glutamato, lo que hace que estos respondan con un aumento en la concentración del calcio intracelular (Charles et al., 1991; A H Cornell-Bell et al., 1990; Perea & Araque, 2002). Además, se pudo observar que el glutamato puede generar una propagación de ondas de calcio en los astrocitos cercanos. A mayores, la

señal de calcio, producida por un astrocito, puede propagarse entre diferentes astrocitos (Innocenti, Parpura, & Haydon, 2000). Esas ondas de calcio representan una nueva forma de comunicación intercelular a larga distancia en el SN (Alfonso Araque et al., 2014; Schipke & Kettenmann, 2004).

Es importante destacar que la forma de transmisión entre neuronas y astrocitos difiere, ya que las primeras lo hacen de una manera muy rápida mediante impulsos eléctricos (entre 10-100m/s) y los segundos producen ondas de calcio que son más lentas (entre 4-20m/s) quedando patente la comunicación entre astrocitos (A Araque, Parpura, Sanzgiri, & Haydon, 1999).

2.1.4. Sinapsis Tripartita

Conocida la excitabilidad de los astrocitos y que éstos provocan una señal que afecta a la transmisión de información entre neuronas, se acuñó un nuevo término sináptico conocido como sinapsis tripartita (figura 5) (A Araque et al., 1999). La sinapsis tripartita considera que existe una comunicación sináptica que se produce entre dos neuronas y un astrocito. La manera de actuar es, cuando en una sinapsis una neurona libera una cantidad adicional de neurotransmisores el astrocito se activa y éste puede liberar una cantidad de gliotransmisores que también recibirá la neurona postsináptica (Theodosis, Poulain, & Oliet, 2008). El astrocito también puede tener la función de aislar una neurona de liberar los neurotransmisores de una sinapsis para que lo reciban otras neuronas a través de la producción de filopodios en la superficie de los astrocitos (Ann H. Cornell-Bell, Thomas, & Smith, 1990; Theodosis et al., 2008).

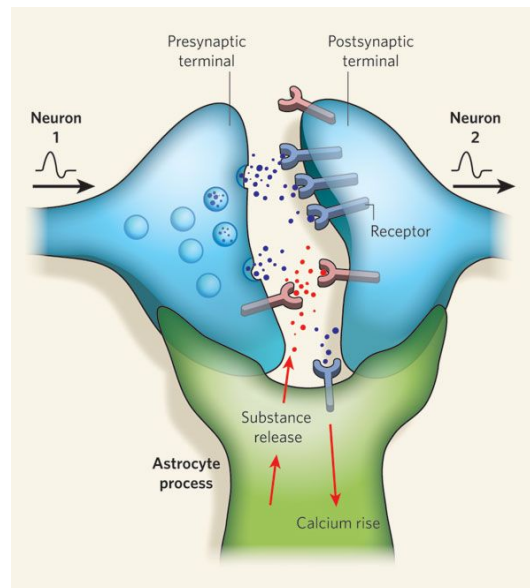


Figura 5. Sinapsis tripartita (Allen & Barres, 2009)

Además, el astrocito puede potenciar la conexión de neuronas al reducir el espacio sináptico entre ellas como consecuencia de la capacidad del astrocito de aumentar de tamaño (Genoud et al., 2006). Esto implica que los astrocitos son capaces de interferir en la capacidad del SN para modificar de manera temporal o permanente la eficacia de la conexión sináptica entre neuronas, capacidad que también se conoce como plasticidad sináptica. Dicha capacidad se considera responsable de los procesos celulares de la memoria y el aprendizaje.

Todos estos estudios indican que los astrocitos influyen en el procesamiento de la información a través de la sinapsis tripartita. Sin embargo, se llevaron a cabo pruebas para observar el efecto de las ondas de calcio astrocíticas en las sinapsis del hipocampo, observando que solo un subconjunto de alrededor de un 40% experimentaron potenciación inducida por los astrocitos (Perea & Araque, 2007). Este estudio indica que la modulación neuronal no es el resultado de una liberación masiva de gliotransmisores, sino que sugiere la existencia de caminos de señalización específicos entre astrocitos y neuronas. Esto afirma la postulación de que no todas las sinapsis son tripartitas (Ventura & Harris, 1999).

Como conclusión se puede determinar que los astrocitos tienen un papel importante en el procesamiento de información ya que actúa en la sinapsis, regulando la transmisión y modificando la plasticidad sináptica. Es sensato indicar que los astrocitos están involucrados en las tareas cognitivas del SN y que deben ser considerados, junto a las neuronas, como elementos involucrados en la actividad cerebral.

2.2. Fundamentos computacionales

Desde hace mucho tiempo, el ser humano ha querido introducir el concepto de inteligencia a diferentes máquinas que le hiciera la vida más fácil, se percataron de que dotar a una máquina con comportamiento inteligente no era fácil, pero con el paso del tiempo diferentes personas han tratado de añadirle ese comportamiento. Un ejemplo de ello puede ser la paloma voladora de Architas de Trente, el caracol animado de Demetrio de Farleria, o el sistema de defensa de Siracusa de Arquímedes (Russell, Norvig, & Rodríguez, 2004).

Es cierto que los autómatas sorprendían por lo que eran capaces de realizar, llegando a existir ejemplos más que notables como puede ser el mayordomo de San Alberto Magno o la creación del talentoso Leonardo da Vinci con su león de pecho florido. A pesar de que se llegaron a construir auténticas obras de arte todos los autómatas tenían una fuerte limitación, su estructura. Todo el comportamiento de un autómata estaba definido por su estructura interna, imposibilitando realizar algo que no estuviera en su predefinido, con anterioridad, en su mecanismo.

Con el nacimiento de la informática se llegó a pensar casi desde sus inicios que las computadoras podrían tener un comportamiento inteligente y los investigadores empezaron a trabajar para crear modelos de IA que dependiendo de su modo del trabajo pertenecerán a una u otra rama (Smolensky, 1987). Los modelos de IA simbólica se caracterizan por estar basados en la representación de un alto nivel de los problemas, la lógica, la matemática y la búsqueda de soluciones. También existen modelos de IA conexionistas, que también son llamados subsimbólicos, que tratan de simular los elementos de más bajo nivel dentro de los procesos cognitivos, a fin de evidenciar el comportamiento inteligente. La rama conexionista trabaja con conceptos fuertemente relacionados con la autonomía, aprendizaje y adaptación.

La rama simbólica tiene un diseño *top-down*, es decir, el diseño del sistema sigue un esquema de arriba hacia abajo, ya que es imprescindible disponer de una aproximación de la solución para poder diseñar el sistema. Sin embargo, la rama conexionista tiene un diseño *bottom-up* ya que los sistemas diseñados recogen las características físicas de los sistemas que tratan de emular y proporcionar una solución de forma automática mediante técnicas prefijadas de aprendizaje, sin que sea necesario proporcionar un conocimiento previo del problema a resolver.

Dentro de la rama conexionista, una de las técnicas que más éxito ha tenido ha sido la RNA la cual lleva años aplicándose con éxito en distintos ámbitos. Los precursores de las RNA fueron McCulloch y Walter Pitts que fueron reconocidos como

los autores del primer trabajo de IA con su aportación en 1943 (McCulloch & Pitts, 1943). Ambos partieron de tres fuentes:

- Conocimiento sobre filosofía básica y el funcionamiento de las neuronas en el cerebro.
- El análisis formal de la lógica proposicional de Russell y Whitehead
- La teoría de computación de Alan M. Turing

2.2.1. Redes de Neuronas Artificiales

Las Redes de Neuronas Artificiales (RNA) son un paradigma, de la rama conexionista de la IA, para el aprendizaje automático y el procesamiento de información. La idea que subyace detrás de las RNA es la de un sistema con elementos de procesado, conocidos como neuronas artificiales, que están interconectados entre ellos para que al recibir un estímulo de entrada este pueda generar una salida en función del estado en el que se encuentra en el sistema.

Como hemos mencionado anteriormente, el primer modelo de IA fue propuesto por McCulloch y Pitts y en concreto fue un modelo conexionista muy simple (McCulloch & Pitts, 1943). Dicho modelo se basaba en una aproximación muy simplificada del cerebro donde consideraba que la neurona era un dispositivo binario que puede producir los estados activa o inactiva. Las interconexiones de varias de esas neuronas conformaban una RNA. El problema que tenían los primeros modelos era el de no disponer de un método adecuado que estableciese de manera automática los valores de las interconexiones de las neuronas.

No fue hasta la década de los 80 cuando se publicó el algoritmo de retropropagación que permitía un establecer el valor de las conexiones de la red mediante un algoritmo de aprendizaje (Rumelhart, McClelland, & PDP Research Group, 1986). Este algoritmo ajusta los pesos de las conexiones entre las neuronas de tal manera que minimiza la diferencia entre el valor que debería dar la red y el valor que está realmente devuelve. También en la década de los 80, Kohonen publicó unas redes autoorganizativas con un aprendizaje adaptativo que era capaz de generalizar de una manera muy rápida (Kohonen, 1989).

En definitiva, una RNA se compone de neuronas artificiales que están interconectados entre sí. La neurona está considerada un elemento de procesado que recibe unos parámetros de entrada, a modo de estímulos, que pueden provenir del entorno o de otras neuronas y que es capaz de procesarla para emitir una salida hacia el entorno o hacia otras neuronas. La interconexión en una RNA se le llama peso y es un valor que determina la relevancia que tiene la conexión de una neurona a otra.

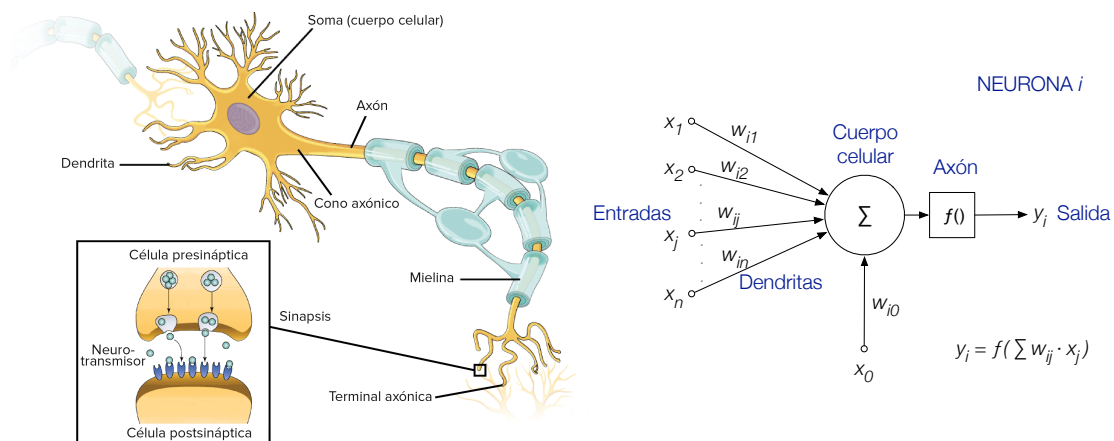


Figura 6. Sinapsis biológica (izquierda) y modelo de neurona artificial (derecha) (Khan Academy, 2018)

Como se puede ver en la figura 6 las conexiones de las neuronas, también denominadas pesos, se etiquetan de la manera w_{ij} , donde la i indica cual es la neurona que emite la salida y la j es la neurona que recibe la respuesta. Cada una de las conexiones de la red tienen un valor asociado que son lo que permite que la neurona emita el valor neto, que es el procesamiento que realiza la neurona según el estímulo recibido en la entrada. Después, a ese valor se le aplica una función de activación que es como la neurona trata la información procesada para, por último, aplicar la función de distribución que es la que se le aplica al enviar los datos a las demás neuronas (Viñuela, 2004).

La creación de una RNA está ligada a dos fases. La primera de ellas se corresponde a la modelización de la red y la segunda se corresponde a la fase ejecución de red. Estos pasos se detallan a continuación:

1. Etapa de modelización.

1.1. Diseño de la arquitectura de red. En este paso se tiene que determinar cómo se va a definir la red. Hay que definir el número de capas ocultas que tendrá la red, el número de neuronas que habrá dentro ellas además de indicar las funciones de activación y distribución que emplearán. Esta es una fase muy delicada, ya que la topología de la red está asociada a la manera de cómo se va adaptar en la representación del problema que abarcará. A día de hoy no hay una única forma de establecer una arquitectura adecuada para cada caso, algunos autores indican algunos consejos, como el que el número de neuronas en las capas ocultas tiene que estar entre las características de entrada y las de salida, o incluso hay autores que dicen que se tiene que tener en cuenta el número de patrones disponibles (Hagan, Demuth, & Beale, 2014; Heaton, 2008). El no existir un modo adecuado de escoger una arquitectura se suele

recurrir al ensayo-error o utilizar técnicas de IA para tratar de optimizar las topologías (Shen, Wang, & Li, 2007).

1.2. **Ajuste de los valores de las conexiones.** Una vez que tenemos la topología de la red, es necesario que la red adquiera el conocimiento y para eso se la somete a un proceso de entrenamiento para que pueda aprender cómo dar las respuestas adecuadas para los estímulos, o patrones de entrada, que se le presenta en esta fase (Pazos-Sierra, 1996). Para ello, se le presenta un subconjunto de los posibles valores a los que la red se va a tener que enfrentar en un futuro. Es importante que todos estos datos sean una muestra representativa de la casuística del problema y también es importante que se le presenten contraejemplos. Todos esos valores que se le presentan a la red en esta fase condicionará su aprendizaje, que será el de establecer los valores adecuados a todos los pesos de la red. Aunque existen diversos métodos el más habitual es el de retropropagación del error (o *backpropagation*) que se puede aplicar a las redes totalmente conectadas hacia delante o, dicho de otra manera, que la conectividad entre las capas sea solo entre la actual y la siguiente y que todas las neuronas de una capa esté conectada con la siguiente (Rumelhart et al., 1986).

A la hora de que una red aprenda existen tres grandes técnicas: aprendizaje supervisado, aprendizaje no supervisado y aprendizaje por refuerzo.

El aprendizaje supervisado se caracteriza porque existe un mecanismo que es capaz de indicarle a la red la respuesta que tiene que dar a cada estímulo y con esto es capaz de determinar el error que está cometiendo. Durante este entrenamiento se le presenta a la red, de manera repetida, el conjunto de entrenamiento para comparar su salida con la que realmente debería de dar para después ajustar el valor de sus conexiones. El ajuste de los pesos de la red tiene como objetivo que cada vez se acerque más a las respuestas deseadas. Después de iterar, y pasarle varias veces el conjunto de entrenamiento a la red, los valores de sus pesos acabarán convergiendo hacia unos valores óptimos y, es cuando se dice que el algoritmo de aprendizaje converge y se da por finalizada la fase de entrenamiento ya que la red fue capaz de aprender.

El aprendizaje no supervisado se caracteriza porque no existe un mecanismo que le indique a la red cual es la respuesta adecuada. En estos casos se intenta que la red sea capaz de encontrar unas relaciones en los datos de entrada. El ejemplo más común de redes que usan este tipo de aprendizaje son las redes con mapas autoorganizativos cuya función es realizar una tarea de *clustering*, o agrupación, sobre los datos (Kohonen, 1989; Prudent & Ennaji, 2005).

El aprendizaje por refuerzo tiene la peculiaridad de que no existe un mecanismo que le indique a la red cual es la respuesta adecuada, pero sí que hay un mecanismo que le indica la idoneidad de la respuesta dada a la red y así

puede saber si su aprendizaje va por buen camino (Williams, 1992). Este entrenamiento es mucho más realista ya que la red tiene un comportamiento para las señales que recibe del entorno y cuando la respuesta es buena tiene una recompensa. Un ejemplo, del mundo real es cuando a un perro se le indica que se siente y al realizar la acción se le da una recompensa, como puede ser una galleta.

Estos métodos de entrenamiento definen las formas básicas de entrenamiento, pero se pueden mezclar para obtener otros tipos de entrenamiento. Un ejemplo, podría ser un entrenamiento híbrido que une el aprendizaje supervisado y el no supervisado, donde cada método ajusta un determinado número de conexiones (Kohonen, 1995). Otro ejemplo podría ser el entrenamiento evolutivo, donde con técnicas de computación evolutiva se intenta optimizar el proceso de aprendizaje en las RNA (Holland, 1975; J. R. Rabuñal, Dorado, Pazos, Pereira, & Rivero, 2004; Rivero, Dorado, Fernández-Blanco, & Pazos, 2009).

- 1.3. **Validación de red.** Es habitual que la validación de la red se realice a la vez con la fase de entrenamiento empleando un conjunto de datos distintos. El objetivo de esta fase es que la red no tenga problemas durante el entrenamiento, como puede ser que la red sobreentrene, especializándose en los patrones de entrenamiento y que pierda la capacidad de generalizar, lo que llevaría a que la red funcionase bien con los datos de entrenamiento, pero no con el resto de datos. Al usar la fase de validación en conjunto con el entrenamiento, se puede observar si la red, con los datos de validación, empieza a cometer más errores. En el momento en que esto suele suceder, se para el entrenamiento, para coger la red en el punto en el que su conocimiento es capaz de generalizar sus datos (figura 7). La validación, por tanto, es un método que se puede emplear para que la red entrene de manera satisfactoria (Viñuela, 2004).

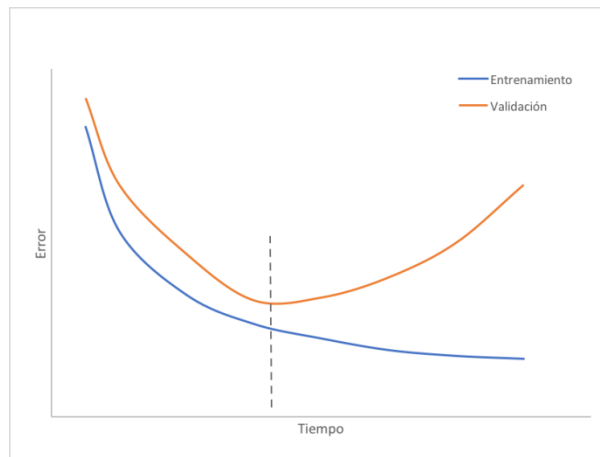


Figura 7. Evolución del error de entrenamiento y validación de una RNA con sobreentrenamiento

2. **Fase de test.** Cuando la red se tiene entrenada y validada, se realiza un test usando un subconjunto de datos que no contenga los datos que se le han pasado durante el entrenamiento o la validación. Esta etapa sirve para poder tener una medida de la capacidad de generalización de la red y poder conocer cómo se comporta con información que no se le ha pasado anteriormente. Esto nos dará una indicación de cómo se comporta con el problema para el cual ha sido creada.
3. **Fase de ejecución.** Esta etapa la red ya está lista para que se enfrente con datos del entorno real. En este paso lo que se le pide a la red es que responda a información que no se le ha presentado previamente y que gracias a las relaciones encontradas anteriormente y que esta fue capaz de generalizar lo realizara adecuadamente.

2.2.2. Algoritmos genéticos

Los Algoritmos Genéticos (AG) pertenecen a una rama de IA cuyo aprendizaje se basa en técnicas que simulan el concepto de evolución observado en la naturaleza. La idea es tomar las bases de la teoría de la evolución que postuló Darwin en 1859 (Darwin, 1859). El objetivo es tratar de reproducir las capacidades que tienen los seres vivos de adaptarse al entorno que los rodea para poder crear técnicas de computación evolutiva. Concretamente, el AG es un algoritmo de búsqueda que basa en la reproducción donde los individuos tienen que adaptarse al medio y solo sobrevivirán los más aptos.

Una parte fundamental del AG son los individuos ya que estos tienen la representación de los parámetros necesarios para resolver un problema dentro de su cromosoma. Al igual que en la naturaleza, en los AG el cromosoma de los individuos

suele estar codificado. Lo más habitual es usar una representación binaria aunque existen representaciones que codifican directamente cada parámetro (Goldberg, 1991; Holland, 1975). Si la representación del cromosoma es binaria, se usa una cadena de bits donde cada uno de los bits se conoce como alelo.

El AG trata de buscar la solución a un problema partiendo de una población inicial de individuos que suele crearse de manera aleatoria. A cada uno de esos individuos se le asigna un valor de bondad o *fitness* en función de cuan bueno es resolviendo el problema planteado (Goldberg & Holland, 1988). Este valor, que se calculará de manera diferente para cada problema, es un valor cuantitativo que usa el AG durante todo el proceso de búsqueda hacia la solución adecuada.

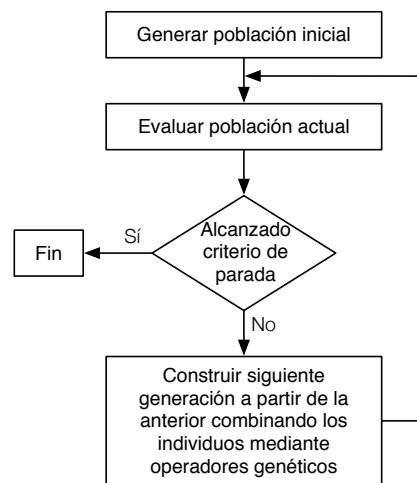


Figura 8. Funcionamiento general de un algoritmo genético

El algoritmo comienza con una población inicial de individuos generados de manera aleatoria. A continuación, se evalúan todos los individuos y, si uno ya es adecuado para el problema se termina el algoritmo y se selecciona dicho individuo, en caso contrario se continúa seleccionando individuos usando algún algoritmo de selección como puede ser mediante torneo o ruleta (De Jong, 1975; Gen & Cheng, 2000). Una vez seleccionado los individuos se realiza una operación de cruce para tener una descendencia cuyo cromosoma sea una mezcla de los cromosomas de los progenitores. Al igual que sucede en la naturaleza, los descendientes pueden tener alguna mutación y, en el caso de los AG, solo se realiza con una pequeña probabilidad la cual provoca el cambio aleatorio en alguno de sus genes. La ventaja de la mutación es que permite que un individuo pueda saltar a una zona del espacio de búsqueda que no se puede llegar mediante la combinación de ellos, aunque esto puede proporcionar una ventaja no se puede abusar de ello ya que si no se realizaría una búsqueda basa en el azar. Después de realizar iterar varias veces los pasos de selección, cruce y

mutación, si esta procede, se alcanzará una nueva generación en la población. El AG continuará realizando nuevas generaciones hasta que se cumpla alguna condición de parada. Dentro de la condición de parada está la de que un individuo tenga una buena solución a un problema, pero se le puede añadir más comprobaciones como el de no realizar un número máximo de generaciones o que este finalice si no continúa convergiendo.

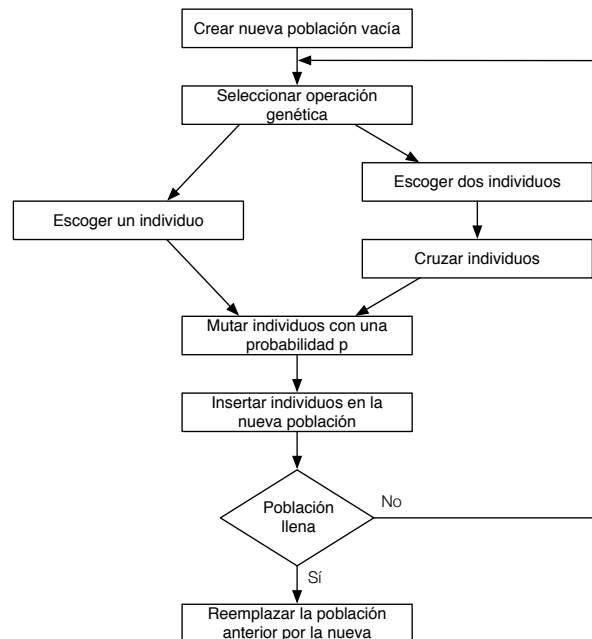


Figura 9. Creación de una nueva generación en un algoritmo genético canónico

Con este algoritmo lo que se consigue es que los individuos vayan evolucionando y mejorando los valores de bondad usando las bases que postuló Darwin.

Los AG se pueden emplear para entrenar las RNA y optimizar distintos aspectos como puede ser la arquitectura de la red, las conexiones, las funciones que emplean las neuronas, etc (Dorado, 1999). En el trabajo realizado en esta tesis se ha empleado entre otros un AG para el entrenamiento de RNA donde el AG busca los valores adecuados para los pesos de la red (figura 10).

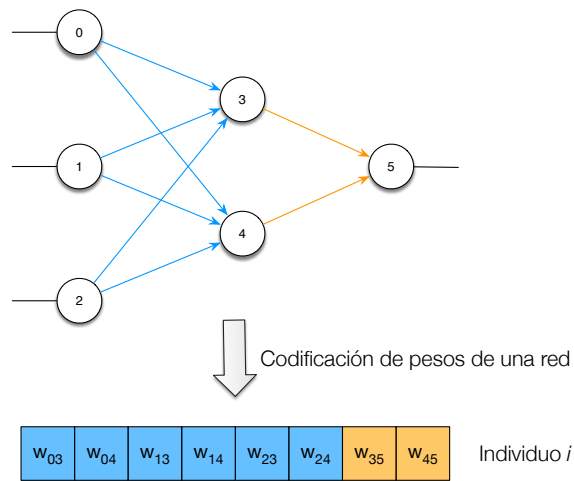


Figura 10. Codificación de los pesos de una RNA para un individuo del algoritmo genético

3. Antecedentes

En este capítulo se exponen los antecedentes biológicos y computacionales considerados para la realización de este trabajo de tesis. Se hace especial hincapié en el aspecto clave de la investigación que es la excitabilidad de los astrocitos lo que implicará una comunicación entre astrocitos y neuronas.

3.1. Antecedentes biológicos

3.1.1. Excitabilidad de los astrocitos

Gracias a las nuevas técnicas de pigmentación de imagen se pudo ver que el SG tiene una gran importancia dentro del SN y, que no todas las células gliales, carecen de un comportamiento activo en el tratamiento de la información. Estas nuevas técnicas permitieron evidenciar que dentro del SG las células conocidas como astrocitos tienen una excitabilidad celular que se apoya en las variaciones que se producen en el espacio intracelular.

Mientras que la excitabilidad en las neuronas se realiza mediante un cambio en su potencial eléctrico, también conocido como *spike*, los astrocitos presentan su excitabilidad celular como una variación intracelular del ion calcio lo que se manifiesta como un aumento de concentración citoplasmática de calcio (Charles et al., 1991; A H Cornell-Bell et al., 1990; Perea & Araque, 2002). Esta respuesta por calcio puede deberse a una respuesta a la liberación de neurotransmisores de la actividad sináptica (Navarrete & Araque, 2008; Perea & Araque, 2002, 2005a). Además, el cambio de concentración de calcio se puede transmitir tanto en el interior celular como a diferentes astrocitos (Perea & Araque, 2005a). Esa transmisión a otros astrocitos es una señal intercelular que deja como prueba que existe una comunicación entre los astrocitos. Esto es posible dado a que los astrocitos son capaces de generar ondas de calcio lo cual puede incluso generar una comunicación a larga distancia (Bazargani & Attwell, 2016; Schipke & Kettenmann, 2004).

Además, se observó que los astrocitos del hipocampo pueden responder a selectivamente y discriminar la actividad de sinapsis específicas que usan diferentes neurotransmisores (Perea & Araque, 2005a). A mayores, también se descubrió que los astrocitos responden selectivamente a caminos sinápticos que usan glutamato como neurotransmisor en las zonas del hipocampo y del tálamo ventrobasal (Parri, Gould, & Crunelli, 2004).

Otra de las observaciones interesantes es que los astrocitos no emiten una señal aditiva como respuesta a la activación de varios caminos sinápticos. Si varios caminos se activan simultáneamente a frecuencias muy altas se observa una depresión en la

señal de calcio que provocan los astrocitos mientras que con frecuencias bajas ocurre todo lo contrario, la respuesta astrocítica es el aumento de calcio (Perea & Araque, 2005a). Dicho de otra manera, cuando la actividad sináptica producida por diferentes neuronas es alta la respuesta se atenúa, mientras que cuando la actividad sináptica es baja su respuesta se incrementa.

3.1.2. Receptores de endocannabinoides en astrocitos

Existe una controversia en cuanto a la presencia de receptores de cannabinoides en los astrocitos (Nephi, 2004). Mientras que algunos estudios no detectan los efectos de los cannabinoides en astrocitos, otros estudios los han localizado en el hipocampo, en la médula espinal y varios estudios muestran su presencia en células gliales. Consecuentemente la funcionalidad de los cannabinoides en los astrocitos está pobremente definida (Lisa & Nephi, 2003; Molina-Holgado et al., 2003; Nephi, 2004; Rodriguez, Mackie, & Pickel, 2001; Salio, Doly, Fischer, Franzoni, & Conrath, 2002).

El sistema endocannabinoide (SEC) es un sistema de señal intercelular que está envuelto en una gran variedad de procesos fisiológicos. El SEC consiste en receptores de cannabinoides, transmisores endógenos (llamados endocannabinoides), y todos los enzimas necesarios para su síntesis, liberación y degradación. Hay dos tipos de receptores de cannabinoides denominados CB₁ y CB₂ donde el primero está expresado en el SN y el segundo se encuentra en algunas células del sistema inmune.

Los receptores CB₁ tienen un papel neuromodulador fundamental en la fisiología del cerebro (Chevalleyre, Takahashi, & Castillo, 2006; FREUND, KATONA, & PIOMELLI, 2003; Rodriguez et al., 2001). Este receptor interviene en la inhibición retrógrada de la liberación de transmisores, controla la excitabilidad neuronal y regula la plasticidad neuronal a largo y corto plazo (Alger, 2002; Chevalleyre et al., 2006; Wilson & Nicoll, 2002).

Elevación de calcio intracelular provocada por astrocitos del hipocampo que expresan funcionalmente receptores de endocannabinoides.

Gracias al uso de técnicas de imagen usadas en el cerebro de ratones se pudo estudiar la expresión de los receptores de cannabinoides en los astrocitos alojados en el hipocampo y su participación en la comunicación entre neuronas y astrocitos (Navarrete & Araque, 2008). Los estudios revelaron que los astrocitos del hipocampo expresan CB₁ funcionales que al activarse conducen la movilización de ondas de calcio. Estos receptores pueden ser activados por endocannabinoides liberados por neuronas piramidales incrementando los niveles de calcio y estimulando la liberación de glutamato.

La respuesta de los astrocitos puede ser indirectamente evocada por los receptores CB₁ donde pueden inhibir la liberación de transmisores sinápticos (Alger, 2002; Chevaleyre et al., 2006; FREUND et al., 2003). Uno de los estudios que se realizaron para probar la respuesta de los astrocitos mediante el receptor CB₁ fue la aplicación de WIN 55,212-2, que es un compuesto de la familia de los cannabinoidérgicos que produce efectos similares a los cannabinoides, y se bloquearon otros receptores para prevenir la acción de liberar neurotransmisores (Michalski, 2008; Navarrete & Araque, 2008). Las elevaciones de calcio que se observaron indican que la respuesta estaba mediada por la activación de los receptores CB₁ alojados en los astrocitos. El estudio confirmó que los receptores CB₁ pueden ser responsables de las elevaciones de calcio producidas por astrocitos.

Los endocannabinoides liberados por neuronas piramidales provocan una elevación de calcio astrocítica

Algunos científicos estudiaron los mecanismos celulares envueltos en la despolarización neuronal que provocan la elevación de calcio astrocítico comparando las señales de calcio antes y después de diferentes tratamientos farmacológicos (Navarrete & Araque, 2008). Se pudo ver que el aumento provocado por la despolarización neuronal en el incremento de las señales de calcio astrocíticas no se previno por la presencia de sustancias que suprimen los potenciales de acción ni de sustancias que bloquea los canales de sodio, lo que sugirió que la despolarización neuronal no se debió a los potenciales de acción provocados por la actividad neuronal (BERGILLOS, 2013; Butterworth, Cole, & Marlow, 1993). Los resultados que se obtuvieron indican que el incremento de las oscilaciones astrocíticas de calcio producidas por la despolarización neuronal son debidas por la activación de los receptores CB₁ (Navarrete & Araque, 2008).

La despolarización neuronal estimula la liberación de endocannabinoides, los cuales reducen durante un corto periodo de tiempo la liberación de transmisores inhibitorios de la activación del receptor presináptico CB₁. Este proceso se conoce como la supresión de la inhibición inducida por la despolarización (Alger, 2002; Chevaleyre et al., 2006; FREUND et al., 2003). Usando esta técnica como una prueba biológica de la liberación de endocannabinoides, se puede comprobar los efectos de la despolarización neuronal en la supresión de la inhibición inducida por la despolarización en las neuronas y la probabilidad del pico de calcio en los astrocitos simultáneamente registrando corrientes inhibitorias postsinápticas por las neuronas piramidales y los niveles de calcio astrocítico (Porto-Pazos et al., 2011). Incrementando la duración en la inducción de despolarización neuronal se observó un incremento gradual en la probabilidad de picos de calcio y la supresión de la inhibición inducida

por la despolarización. Ambos parámetros muestran una gran correlación lo que sugiere que la señal astrocítica de calcio resulta como la liberación de endocannabinoides de neuronas piramidales y que indica que los endocannabinoides son igualmente efectivos para inducir efectos neuronales y astrocíticos.

Los resultados indican que las elevaciones de calcio en las neuronas piramidales son necesarios y suficientes para inducir elevaciones de calcio en los astrocitos y, a mayores, apoyan la idea de que los receptores CB₁ median las respuestas de calcio astrocítico hacia las neuronas despolarizadas que son evocadas por la liberación de endocannabinoides liberados por las neuronas piramidales (Navarrete & Araque, 2008).

Corrientes internas lentas en las neuronas piramidales provocadas por las elevaciones de calcio en astrocitos medidas por el receptor CB₁.

Algunos informes indican que los estímulos exógenos experimentales elevan el calcio intracelular en los astrocitos e inducen corrientes lentas internas mediadas por el glutamato en neuronas contiguas (Alfonso, Vladimir, P., & G., 1998; Angulo, Kozlov, Charpak, & Audinat, 2004; Fellin et al., 2004; Perea & Araque, 2005b). Un estudio registró las corrientes de dos neuronas piramidales cercanas y los niveles de calcio de astrocitos y resultó que las neuronas analizadas no fueron estimuladas porque la observación de corrientes lentas internas durante la despolarización neuronal impidió la estimulación neuronal. La despolarización de una neurona piramidal aumento la frecuencia de oscilación de calcio en astrocitos y de las corrientes lentas internas de manera simultánea en las neuronas contiguas (Navarrete & Araque, 2008; Vyklicky et al., 2014). La despolarización neuronal provocó el incremento de la frecuencia en las oscilaciones de calcio y en las corrientes lentas internas. Ese estudio indicó que las elevaciones de calcio en astrocitos mediados por los receptores CB₁ son necesarias para el aumento de la frecuencia de las corrientes lentas internas inducidas por la despolarización neuronal.

A través de varios experimentos se pudo indicar que la despolarización neuronal de las neuronas estimula la liberación de endocannabinoides, los cuales activan los receptores CB₁ en los astrocitos, elevando el calcio intracelular e induciendo la

liberación de glutamato que a su vez sirve como una señal de retroalimentación por la activación de NMDA¹ en las neuronas (Navarrete & Araque, 2008).

Se puede concluir que en el hipocampo los receptores de endocannabinoides generan una alta actividad sináptica, lo que provoca una depresión en las sinapsis cercanas debido a la despolarización de las neuronas. Además, esta actividad sináptica hace que los astrocitos potencien las sinapsis de las conexiones más lejanas al variar las concentraciones de calcio en los astrocitos adjuntos. Este comportamiento se puede trasladar a los SC donde un astrocito artificial deprima las conexiones de la neurona asociada mientras que el resto de conexiones de las neuronas de la misma capa se potencien. Así se simulará en este trabajo de tesis.

3.2. Antecedentes Computacionales: Redes NeuroGliales Artificiales

Los descubrimientos mencionados en capítulos anteriores revelan la existencia de una comunicación entre neuronas y astrocitos y ponen de manifiesto la participación de los segundos en el procesamiento de la información en el SN lo que hizo viable su incorporación en las RNA. La incorporación de astrocitos artificiales a RNA multicapa dio lugar a la creación de Redes NeuroGliales Artificiales (RNGA) (Porto, 2004; Porto, Pazos, & Araque, 2005). Como consecuencia también se creó una metodología de construcción para estos SC donde se incorporan dos elementos de procesamiento diferenciados (Porto, 2004) y será explicada en detalle en el apartado 3.2.1.

Las redes multicapa totalmente conectadas hacia delante, sin conexiones laterales ni conexiones recurrentes, fueron las escogidas para la metodología diseñada por A. B. Porto-Pazos (Porto, 2004). En estas redes se le asocia una neurona artificial a un astrocito, donde cada astrocito es el encargado de modular la sinapsis de su neurona asociada (figura 11).

¹ El NMDA son unos tipos de receptores de glutamato.

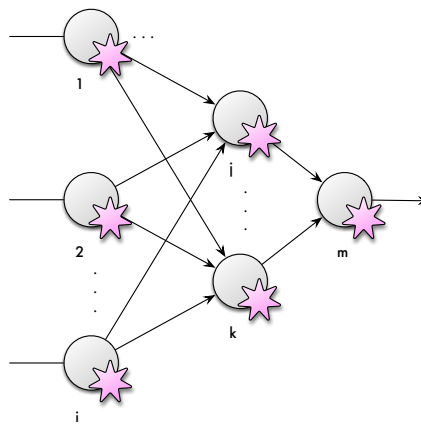


Figura 11. Red NeuroGlial Artificial (RNGA). Los círculos representan las neuronas y las estrellas representan los astrocitos.

Las pruebas sobre esta nueva aproximación en los SC se llevaron a cabo en redes de clasificación. Esta elección se hizo ya que el control de los astrocitos en las conexiones para proceder a su refuerzo o debilitación, observado en el cerebro, está relacionado con la adaptación o plasticidad de las conexiones, generándose caminos de activación como resultado de dicha adaptación. Este proceso podría, por tanto, facilitar la clasificación de los patrones o su reconocimiento en la RNGA implementada.

3.2.1. Metodología

La metodología creada por Ana Porto y el grupo RNASA-IMEDIR para la construcción de las RNGA es muy similar a la de otros SC. El primer paso es el diseño de la arquitectura que conformará la red y la elección de las funciones asociadas a esta (función de activación y distribución). A continuación, se lleva a cabo la fase de entrenamiento, la cual se hace en paralelo con la fase de validación. Finalmente se realiza la fase de test para poder comprobar si la red generaliza o no.

Fase de diseño

Como se ha mencionado anteriormente las RNGA se han diseñado para la resolución de problemas de clasificación. Su arquitectura está basada en capas que están totalmente conectadas hacia delante.

Fase de entrenamiento.

En la etapa de entrenamiento en las RNGA se emplea un entrenamiento híbrido en dos fases. La primera etapa emplea un aprendizaje no supervisado con un algoritmo

neuroglial que modifica los valores de las conexiones en función de la actividad neuronal. La segunda etapa con aprendizaje supervisado donde se emplea un AG para buscar unos valores adecuados a los pesos de las conexiones obtenidos en la primera parte. Esto se corresponde a los pasos 3-9 de la figura 12.

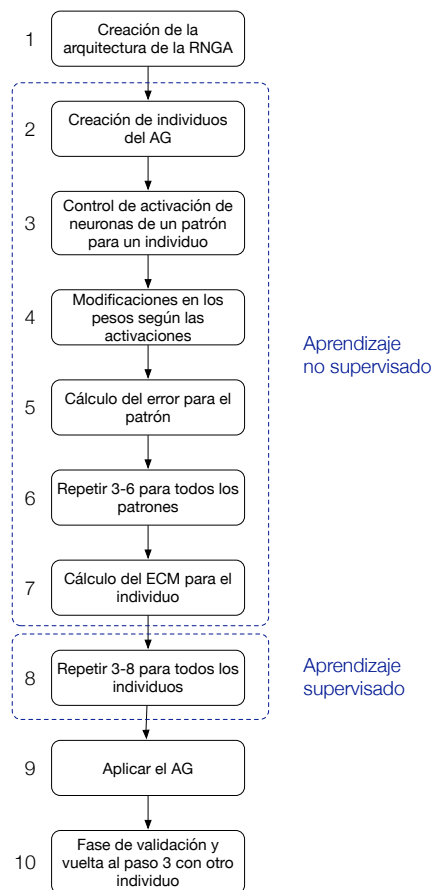


Figura 12. Diagrama de flujo del funcionamiento de una RNGA

Fase de Aprendizaje no supervisado

Como se puede observar en la figura 12 la primera fase del entrenamiento se basa en el comportamiento de los astrocitos que conlleva un aprendizaje no supervisado en la que hace uso de la sinapsis tripartita observada en el cerebro.

En la metodología planteada, esta fase de aprendizaje no supervisada se lleva en conjunto con la fase supervisada del AG. Al emplear un AG se tienen unos individuos que, en este caso, representan los valores de los pesos de las conexiones de una RNGA. El primer paso es la creación de dichos individuos, o lo que es lo mismo, la creación de los pesos de la RNGA. Cada una de las redes que representa los individuos modifica los pesos cada vez que a dicha red se le presenta un patrón de entrada. Cada patrón se le pasa como entrada un número determinado de iteraciones las cuales

representan la lentitud de la actuación de los astrocitos frente a las neuronas. Además, durante las iteraciones en la que se presenta el mismo patrón el astrocito puede modificar las conexiones de acuerdo a ciertas reglas establecidas en base a las activaciones que sufren las neuronas (Porto-Pazos et al., 2011; Porto, 2004; Porto et al., 2007, 2005; Veiguela, 2015).

El funcionamiento de un astrocito artificial es el siguiente: la red evalúa n veces un patrón de entrada y si la neurona que monitoriza el astrocito se ha activado² k veces el astrocito lo tendrá en cuenta para modificar las conexiones de la neurona que monitoriza con las neuronas conectadas de la siguiente capa. Los valores que se consideraron adecuados durante las experimentaciones son los siguientes:

- Incrementar un 25% el valor del peso con las conexiones salientes de la neurona activada k veces durante las n iteraciones.
- Decrementar a la mitad el valor del peso de las conexiones que salen de las neuronas que no se han activado k veces durante las n iteraciones.

Por ejemplo, si se considera que el algoritmo glial emplea una combinación 2-6 de activaciones iteraciones, el patrón de entrada se debería presentar 6 veces a la red y se tendrá en cuenta si las neuronas se activan 2 veces. Si se activan 2 veces se considera que hubo la suficiente actividad neuronal como para que el astrocito se activase y este modifica los pesos en un 25% mientras que si no se activa se considera que no hubo la suficiente actividad neuronal como para que se excite el astrocito y se debilitan las conexiones salientes en un 50%.

Esta manera de actuar representa una de las maneras que se ha observado que realizan los astrocitos en el cerebro, donde necesitan que las neuronas tengan cierta actividad neuronal como para que se exciten los astrocitos y se realice la sinapsis tripartita. El número de iteraciones que se le pasa el mismo patrón trata de simular la actuación de los astrocitos en una escala temporal distinta de las neuronas.

Según las experimentaciones llevadas a cabo se comprobó que el uso de más de ocho iteraciones incrementa notablemente el tiempo de simulación y que no se

² Que un astrocito se active implica que su salida esté por encima de un determinado umbral.

obtenían mejores resultados (Porto, 2004). Sin embargo, con dos iteraciones el astrocito no tiene el tiempo suficiente para actuar. En la tabla 1 se muestran los valores indicados por Ana Porto para la combinación de activación-iteración adecuados.

Las experimentaciones también mostraron que los valores adecuados para el incremento y el decremento de los pesos son el 25% y el 50% respectivamente (Porto, 2004). Esta elección es justificada por los fundamentos biológicos ya que, si el incremento por refuerzo es menor que el decremento cuando no hay actividad constante, solo permanecería reforzada las conexiones de neuronas que representan una actividad continuada (Haydon & Araque, 2002).

Tabla 1. Combinaciones analizadas de activación-iteración

Activaciones	Iteraciones
2	4
2	6
3	6
3	8

Siguiendo con la etapa de entrenamiento, una vez que se acabó el ciclo de presentación del patrón a la red, se calcula el error para el patrón dado. Una vez que se pasan todos los patrones a la red se calcula el error cuadrático medio (ECM) para la red. Este proceso es igual para todos los individuos del AG y constituye un entrenamiento no supervisado ya que todas las modificaciones que realizan los astrocitos no son tomadas en función del error que comete la red, sino que tienen lugar en función de las activaciones de las neuronas, los cuales tratan de simular los refuerzos e inhibiciones que son provocados por los astrocitos biológicos en el cerebro (Kang, Jiang, Goldman, & Nedergaard, 1998).

A modo de resumen el algoritmo que se emplea en el aprendizaje no supervisado es el siguiente:

1. Generación aleatoria de todos los individuos donde cada individuo representa las conexiones que posee una red.
2. Se introduce un patrón a la red donde cada neurona calcula su salida.
3. Se contabiliza las neuronas activadas.
4. El astrocito aplicará las modificaciones en función de las activaciones que tenga la neurona. Es importante mencionar que estas modificaciones son temporales y no perduran los cambios a los siguientes patrones.

5. Tras llevar a cabo todas las iteraciones se calcula el error de la red para el patrón dado.
6. El proceso se realiza con todos los patrones y con todos los errores cometidos se calcula el ECM de la red que representa el individuo para tener su valor de bondad.
7. Se realiza el proceso anterior para todos los individuos de la población y con todos los valores de bondad de los individuos el AG podrá realizar los cruces necesarios en la etapa de aprendizaje supervisado.

Fase de aprendizaje supervisado

La fase de aprendizaje supervisado es llevada a cabo por el AG donde los individuos son ordenados por su valor de bondad, que en este caso son redes cuyo valor de bondad es el ECM que han cometido con unos patrones de entrada. En esta segunda fase el AG emplea las operaciones genéticas necesarias para llevar a cabo para ir mejorando el valor fitness de los individuos, dicho de otra manera, ir reduciendo el error que comete la red con los patrones empleados en la fase de entrenamiento. Los individuos serán seleccionados teniendo en cuenta el error que cometen para después cruzarlos y, solo con cierta probabilidad, mutarlos para generar nuevos individuos. El AG se ejecutará hasta que se alcance un determinado error o se realice un número determinado de generaciones.

Fase de validación

La etapa de validación se realiza conjuntamente con la fase de entrenamiento. Durante esta fase se emplea un conjunto de datos diferente del empleado en la fase de entrenamiento y diferente del que se empleará en la fase de test.

El objetivo de esta fase es poder evitar el sobreentrenamiento y que la red generalice y no se especialice en generar buenos resultados para un subconjunto de datos. Los valores que se obtienen en validación pueden emplearse para evitar la “memorización” de los datos de entrenamiento. Para poder realizar la parada del entrenamiento hay que analizar los errores que comete la red en el conjunto de validación de la red y en el conjunto de entrenamiento para ver que ambos errores

decrecen³ y, si el error de validación aumenta mientras que el error de entrenamiento baja se da el caso en que la red pierde la capacidad de generalizar. También es necesario guardar las configuraciones de la red para una vez acabado el proceso poder seleccionar la configuración que genere el menor error de validación.

Fase de test

Durante la etapa que ejecutó el entrenamiento y validación se construyó una red con una configuración que consigue generalizar la red la cual consigue el error mínimo con los conjuntos de validación y entrenamiento. En la fase de test se utiliza la red para evaluar la capacidad de generalizar de la red. Para ello, se emplea un conjunto de datos que la red no ha visto durante las fases de entrenamiento y validación.

Una diferencia significativa de las RGA con otros modelos de redes es que el algoritmo de modulación astrocítica se sigue ejecutando durante la fase de test y la fase de ejecución y, además continúan actuando de la misma manera que en fases anteriores. Por tanto, los astrocitos de los que dispone la red actúan activamente durante el procesamiento de información. Los datos de entrada que se presentan a la red se ejecutaran durante las mismas iteraciones que en fases anteriores para permitir que estos puedan actuar sobre las conexiones de la red. El comportamiento de aprendizaje no supervisado que tienen los astrocitos se sigue aplicando durante todo el ciclo de vida de la red, algo que simula el comportamiento biológico ya que guarda relación con los procesos de aprendizaje que se dan en el cerebro. Esto implica que la red siempre está realizando un proceso activo de aprendizaje cuyo símil biológico está en la propiedad de plasticidad que tiene el cerebro y tiene la capacidad constante de aprender y adaptarse a nuevas situaciones (Ota, Zanetti, & Hallock, 2013).

3.2.2. Verificación

La metodología que ha sido desarrollada por Ana Porto y el grupo RNASA-IMEDIR ha sido evaluado y comparado con otras RNA en varios trabajos (Porto, 2004; Porto et

³ Aunque en alguna ocasión se da que el error de entrenamiento y validación aumenten puede tomarse la decisión de realizar la parada por sobreentrenamiento o dejar que la red siga entrenando para intentar salir de un mínimo local en el que se encuentra.

al., 2007, 2005). Para ello se realizaron comparaciones con problemas de clasificación sencillos, como el caso de un MUX o la clasificación de la flor de iris, donde se empleó RNGA con un aprendizaje híbrido y RNA con un aprendizaje empleando únicamente AG para la comparación. Las arquitecturas de ambas redes y los parámetros del AG fueron igual para ambos casos y fueron los mismos que los que empleo Rabuñal obteniendo muy buenos resultados para estos problemas (J. Rabuñal, 1998).

Los test fueron llevados a cabo manteniendo para ambos tipos de red las mismas poblaciones de individuos y las ejecuciones usaron la misma semilla para la generación de números aleatorios que requiere el AG para la selección, cruce y mutación de individuos. De esta forma se aseguraban que el AG funcionase de la misma manera para la RNA como para la RNGA.

Para el problema del MUX se había decidido usar dos arquitecturas, con una y tres capas ocultas, fijando el número de generaciones máximas en el entrenamiento a 4000. En la comparación de las redes con una capa oculta se obtuvo que la RNGA obtuvo mejores resultados para test en todas las poblaciones mientras que en el caso de las redes con tres capas ocultas los resultados con RNGA mejoraban notablemente los obtenidos con RNA en todas las poblaciones. Estos resultados daban a entender que el aprendizaje híbrido es mucho más efectivo con redes con una arquitectura más compleja.

El otro problema empleado para comprobar el funcionamiento de las RNGA fue el de flor de iris el cual es más complejo que el anterior ya que empleaba tareas de clasificación múltiple (Fisher, 1936). Los resultados alcanzados por las RNGA fueron de nuevo mejores que los obtenidos por redes sin astrocitos donde además se lograba alcanzar un ECM empleando menos generaciones.

4. Solución propuesta

En este capítulo, se propone la solución que ha permitido validar la hipótesis planteada en esta tesis doctoral. Para ello primero se indica como se ha creado la herramienta desde cero y qué aspectos se han tenido en consideración. Después, se indican los algoritmos neuroastrocíticos que se han implementado. A continuación, se detalla la metodología usada para la construcción del software y la metodología usada para el diseño de los modelos de RNA y RNGA. Luego, se explica qué análisis estadístico se emplea para comparar las diferencias obtenidas con RNA y RNGA. Finalmente, se detalla cuál es el entorno de simulación que se ha empleado para la realización de esta tesis doctoral.

4.1. Implementación de la herramienta de simulación de RNGA

El grupo RNASA-IMEDIR ya disponía de una serie de herramientas que permitían realizar simulaciones de RNGA. Las herramientas que tenían realizaban su labor, pero eran muy rudimentarias, generaban un gran volumen de ficheros de salida, el código para ejecutar la fase de test era diferente y estaba en otra aplicación distinta de la usada para la fase de entrenamiento. El flujo de trabajo se hacía largo. Además, el código que estaba desarrollado estaba básicamente en un fichero, donde todo era adhoc, el algoritmo genético tenía los operadores genéticos adaptados para realizarse sobre las estructuras de las redes. Estas son algunas de las razones que hacían que el uso de los programas fuese difícil, provocando que la decisión final fuese realizar una nueva aplicación desde cero.

Una de las primeras decisiones fue que tenía que existir una única herramienta que fuese capaz de realizar todas las tareas necesarias para la ejecución de las simulaciones. Todo se redujo a usar únicamente tres parámetros de entrada donde el primero indica el modo de funcionamiento de la herramienta, el segundo indica donde está el fichero de configuración y el último parámetro indica donde debe almacenarse la salida. Otra de las opciones que se tomó de diseño, es que no se creasen cosas adhoc, todo se debía de construir creando librerías, lo que dio como resultado librerías para el uso de RNA, RNGA, AG entre otras. Esto posibilitará que se pueda usar el trabajo desarrollado en otros proyectos y que, además, se puedan añadir fácilmente nuevas funcionalidades.

La herramienta que se desarrolló, tiene ficheros de configuración, al igual que la anterior, pero con una gran diferencia, el formato usado. La herramienta de la que disponía el grupo RNASA-IMEDIR usaba los ficheros de entrada como ficheros de texto donde en cada línea estaba el valor del parámetro que se quería leer, algo que dificultaba bastante la creación de un fichero de parámetros, o la observación del

fichero que contenía la población inicial de pesos. La nueva herramienta, todos los ficheros que necesita para funcionar tienen el formato XML, a excepción de los ficheros de los conjuntos de datos, que tienen el mismo formato que anteriormente, un formato CSV. La ventaja de usar archivos XML es que no requieren un orden estricto al añadir la configuración y además ésta es fácilmente entendible.

Otra de las cosas que se quiso añadir desde el principio, y que carecía la anterior herramienta, es la posibilidad de añadir test de prueba. Para ello se creó un *framework* en C que permite, de una manera muy sencilla la creación de test unitarios y de integración en la nueva herramienta. Además, esta característica se añadió en el script de compilación, permitiendo dos modos, uno de *debug*, donde se ejecutan todos los test y otra de *release* que genera la aplicación no sin antes comprobar todos los test. Si alguno de ellos falla, no creará el ejecutable de la aplicación. A mayores, el framework genera un informe indicando todos los test ejecutados, cuánto han tardado cada uno de ellos y el global, y lo más importante, notifica qué ha fallado y muestra el motivo de porqué ha fallado.

La última característica importante que se quiso añadir es la comprobación del funcionamiento de la aplicación. La anterior aplicación se ponía a ejecutar y no devolvía ningún mensaje de error si había algún problema, como puede ser la lectura de una población que no se corresponde o algún valor inválido de un parámetro. La única forma de saber si se produjo un error era mirar si se generaba el log de salida y este tenía sentido.

4.1.1. Búsqueda automática de los parámetros gliales

Se añadió la búsqueda automática de los parámetros de los astrocitos en el AG.

Hasta ahora se empleaba un AG de estado estable para la búsqueda de unos valores adecuados en las conexiones de la red (Porto, 2004; Veigueta, 2015). Como se ha podido comprobar en trabajos anteriores este método funciona correctamente y es una manera adecuada y consolidada de encontrar buenas conexiones para una red neuronal (Dorado, 1999). Los trabajos que se habían realizado anteriormente conllevaban una búsqueda manual en la tarea de asignar unos valores adecuados en los parámetros lo que lleva a un gran consumo de tiempo por tener que usar un método de prueba y error.

La búsqueda de las conexiones adecuadas para la red resultan en un difícil problema de optimización NP (Blum & Rivest, 1992). El añadirle la interrelación de los parámetros de los astrocitos y los valores de los pesos dibuja un espacio de una gran

dimensionalidad. La técnica que permita buscar a la vez los valores adecuados para las conexiones de la red y los parámetros gliciales sería muy beneficiosa.

La técnica que se emplea para la búsqueda de los parámetros de una RNGA la lleva a cabo un Algoritmo Genético Coevolutivo (AGC). Este tipo de AG provee una serie de interesantes características para la optimización de valores. En el AGC el valor de bondad se calcula con una medida de cómo interactúan los individuos. Esta hipótesis permite el potencial de usar problemas más complejos ya que los individuos de los AG tienen que evolucionar en conjunto haciendo que la aptitud de un individuo dependa de la relación de éste con otros individuos (Wiegand, 2003). Esa interacción de los algoritmos coevolutivos puede ser de manera competitiva o cooperativa. Ambas interacciones están muy relacionadas en el mundo real, ya que antiguamente las especies tenían que competir con el resto, mientras que en los últimos años de la historia todas las especies colaboran para construir una solución para el problema (Potter & De Jong, 2000; Rosin & Belew, 1997).

Originalmente los Algoritmos Genéticos Coevolutivos Cooperativos (AGCC) tenían como objetivo en resolver un problema mediante la evolución de distintas partes que evolucionan por separado y luego se ensamblan para tener algo funcional. Existen trabajos que indican que se tienen que cumplir unas características para que el uso de los AGCC sea recomendable (Pena-Reyes & Sipper, 2001). Estas características con las siguientes:

1. Un espacio de búsqueda complejo
2. El problema tiene que poder separarse por definición
3. Se deben usar diferentes tipos de valores y conceptos
4. Tiene que existir una fuerte interconexión entre los componentes de la solución.

Todos estos requisitos se cumplen con las RNGA al intentar de evitar la búsqueda manual de los parámetros de la glía para tener un método de aprendizaje completamente automático.

AGCC para la optimización en RNGA

Como mencionamos anteriormente, los AGCC tienen la particularidad de que el ajuste de un individuo depende de la colaboración con otros individuos y que la solución del sistema tiene que ser la unión de piezas distintas. Con otras palabras, lo que se indica es que este AG tiene la peculiaridad que tiene individuos de distintas especies y éstas evolucionan en base a la colaboración otros individuos de otras especies. Estas especies están separadas en poblaciones y, las operaciones genéticas

se realizan dentro de la misma población, no se permite el cruce de especies. Esta colaboración hace que se puedan encontrar mejores soluciones en problemas complejos.

El funcionamiento de los AGCC no difiere mucho de otros AG, la diferencia está en que hay que tener en cuenta que se emplean varias poblaciones. El AGCC se basa en la evolución de generaciones que se realizan de manera iterada hasta que se cumpla un criterio de parada. En la figura 13, a la derecha, se muestra un diagrama de cómo se realiza la generación de un AGCC empleando una configuración de AG de estado estable. En la figura se puede observar que para cada población se realizan una determinada cantidad de veces lo operadores genéticos para generar nuevos individuos y la manera de evaluar la bondad de los nuevos individuos es el resultado de evaluar varios individuos de otras poblaciones y calcular el error que cometen la solución que conforman para al final asignarle al individuo una medida de todos los errores calculados. Las medidas más comunes para asignar al individuo su valor de fitness es usar la media de los errores, el valor mínimo o el valor máximo.

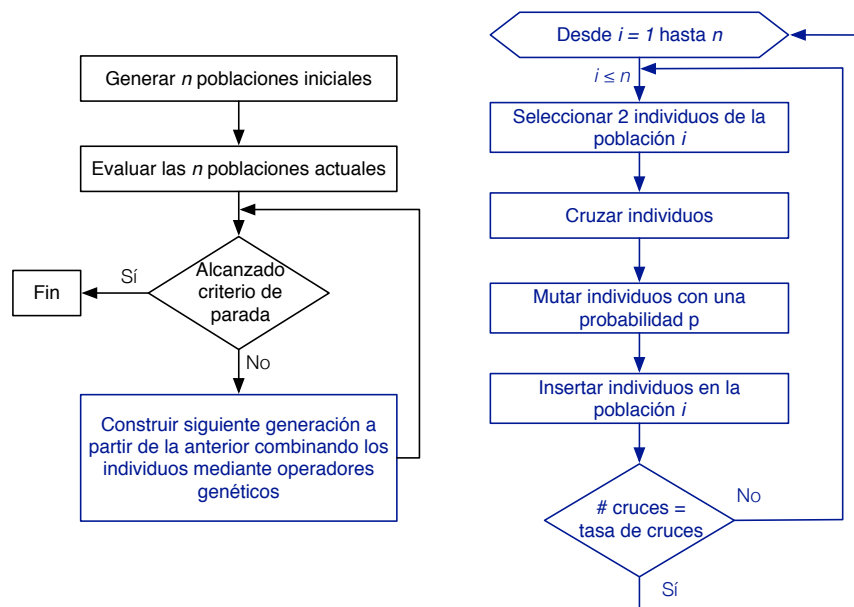


Figura 13. Diagrama de flujo de un Algoritmo Genético Cooperativo Coevolutivo (AGCC).

Para el problema que queremos abordar, que es la búsqueda simultánea de las conexiones de la red y los parámetros giales, se emplearon dos poblaciones, una para los pesos y otra para los parámetros giales. La codificación de ambas poblaciones está basada en una codificación real. La población de pesos de la red, tienen en el cromosoma de sus individuos un vector con todas las conexiones de la red, como se muestra en la figura 14. Por otro lado, la población de parámetros de la glía, tiene todos los parámetros necesarios que necesitan los astrocitos, pero se pueden usar dos

aproximaciones, una donde se emplea la misma configuración astrocítica para toda la red, como se muestra en la figura 15, y otra donde cada capa de la red puede tener una configuración glial diferente, tal y como se detalla en la figura 16. Los parámetros que se codifican en esta última población son el tipo de algoritmo a usar (g), el número de iteraciones (k) que se le pasa el mismo patrón a la red, el número de activaciones (μ) que se tiene que activar la neurona para excitar y provocar una respuesta del astrocito y los porcentajes de incremento (a) y decremento (b) que emplean los algoritmos astrocitos para modificar las conexiones de la red. Como restricciones establecemos que el número de iteraciones tiene que ser mayor que el número de activaciones ($k > \mu$). El valor de iteraciones está entre tres y nueve ($k \in [3,9]$) y el valor de activaciones está entre dos y ocho ($\mu \in [2,8]$). Los valores del porcentaje de incremento y decremento son valores del uno al nueve que indica decenas de porcentaje que se aplicará sobre los pesos ($incremento = (ax10)\%$, $decremento = (bx10)\%$). El valor del algoritmo glial (g) es un entero entre cero y siete ($g \in [0,7]$) que funciona a modo de identificador de que algoritmo emplear.

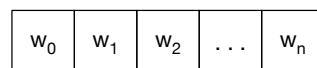


Figura 14. Cromosoma de un individuo de la población de pesos.

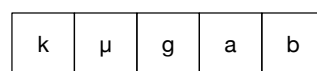


Figura 15. Cromosoma de un individuo de la población del sistema glial donde toda la red usa la misma configuración

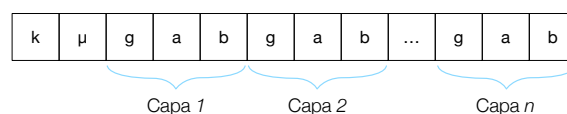


Figura 16. Cromosoma de un individuo de la población del sistema glial donde cada capa usa su propia configuración

Otro aspecto importante a considerar en los AGC es mecanismo de interacción entre las especies. En el caso de los AGCC el individuo de una especie, o población, no tiene la información necesaria para representar la totalidad de la solución a un problema, y se necesita seleccionar individuos de otras poblaciones para formar la solución completa y también es necesaria esa interacción para indicar cuan bueno es el individuo de una población. Entonces la cuestión es como seleccionar a los individuos colaboradores. Una posible solución sería evaluar el individuo con todos los individuos de otras especies la cual no es una buena opción computacionalmente. La opción

implementada es la de seleccionar el número de individuos que se van a usar de otras poblaciones e indicar como se van a seleccionar usando cualquier combinación de las tres siguientes opciones:

- Seleccionar los mejores individuos de otras poblaciones.
- Seleccionar aleatoriamente los individuos de las otras poblaciones.
- Usar cualquier operador genético de selección implementado⁴ sobre las otras poblaciones.

En cuanto a la manera de asignar el valor de fitness de un individuo se implementaron las siguientes:

- Mínimo valor de los errores obtenidos.
- Máximo valor de los errores obtenidos.
- Media aritmética de los errores obtenidos.

4.1.2. Rendimiento

Una de las partes fundamentales del programa desarrollado es la ejecución de las redes ya que la mayor parte del tiempo se pasa ejecutando una red neuronal (en la figura 17 se puede ver que el 90% del tiempo está calculando el error de la red). Mejorar esta parte supondrá entonces una mejora importante en la ejecución de todo el programa. Para llevar a cabo esta tarea lo primero que habría que realizar es analizar cómo se usan las RNA. El tipo de red que se ha usado en la presente tesis es un perceptrón multicapa donde las neuronas artificiales están conectadas con todas las neuronas de la siguiente capa.

⁴ Los operadores genéticos de selección que se han implementado para los AG son: selección aleatoria, selección por ruleta, selección por torneo determinístico y selección por torneo probabilístico.

Weight	Self Weight	Symbol Name
3.68 min	100.0%	0 s
3.68 min	100.0%	0 s
3.68 min	99.9%	0 s
3.68 min	99.9%	0 s
3.67 min	99.9%	0 s
3.38 min	92.0%	1.40 s
3.33 min	90.7%	4.93 s
1.62 s	0.7%	1.62 s
20.00 ms	0.0%	20.00 ms
11.21 s	5.0%	28.00 ms
6.06 s	2.7%	4.00 ms
124.00 ms	0.0%	0 s
5.00 ms	0.0%	0 s
1.00 ms	0.0%	0 s
19.00 ms	0.0%	0 s
7.00 ms	0.0%	0 s
1.00 ms	0.0%	0 s
3.00 ms	0.0%	0 s

Figura 17. Profiling de la herramienta de simulación de RGA

Al momento de guardar la configuración de una RNA se presta una atención esencial en la manera en la que se guardan las conexiones, ya que estas son las partes más importantes y las que más ocupan. La opción que se ha convertido en un estándar de facto es la utilización de una matriz bidimensional donde una de las dimensiones indica la neurona presináptica y la otra dimensión la neurona postsináptica. Así, en la matriz podemos tener una manera muy sencilla de poder representar los valores de los pesos de una red, o lo que es lo mismo, sus conexiones. Para ello se crea una matriz cuya dimensión está delimitada por el número de neuronas de una red.

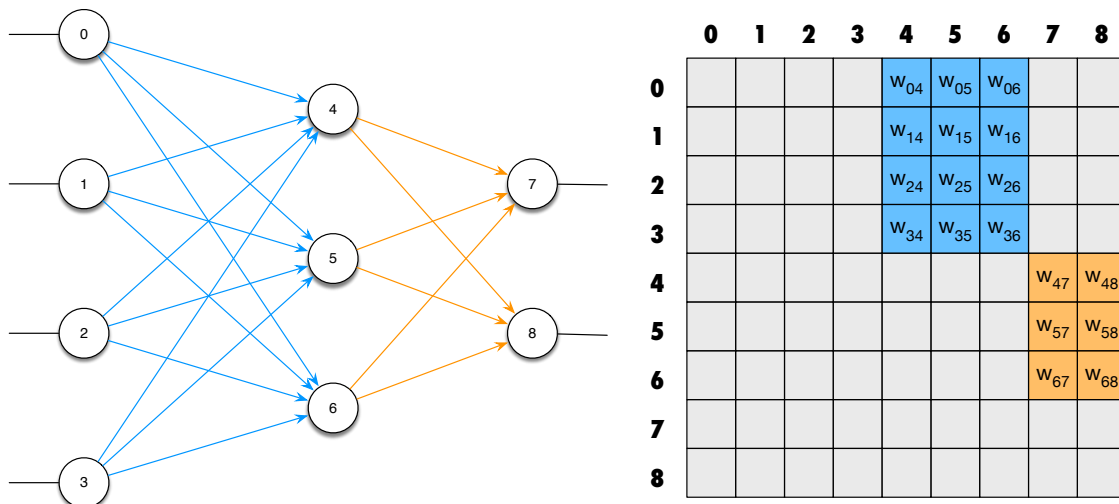


Figura 18. Representación de una RNA en una matriz

Como se puede ver en la figura 18, las celdas azules representan las conexiones de la capa de entrada con la capa oculta mientras que las celdas naranjas representan las conexiones de la capa oculta con la capa de salida. Las celdas grises de la matriz representan posiciones que no se están empleando para almacenar las conexiones de la red. La forma de acceder a una conexión usando la matriz es usando la numeración

de las neuronas donde el primer índice indica la neurona presináptica y el segundo indica la neurona postsináptica, de tal manera que la posición (0,4) representa la conexión de la neurona 0 hacia la neurona 4. En la figura se puede ver fácilmente que gran parte de la matriz no se está usando.

Una mejora que se puede ver fácilmente es el uso de una matriz triangular superior ya que este tipo de redes totalmente conectadas hacia delante no usa la parte inferior. Con esta aproximación se estaría usando la mitad de memoria que se empleaba anteriormente tal y como se muestra en la figura 19. Sin embargo, como se puede observar en la matriz, no es una buena aproximación ya que 18 de las 36 posiciones se siguen sin emplear, lo que supone que se desperdicia el 50% del espacio.

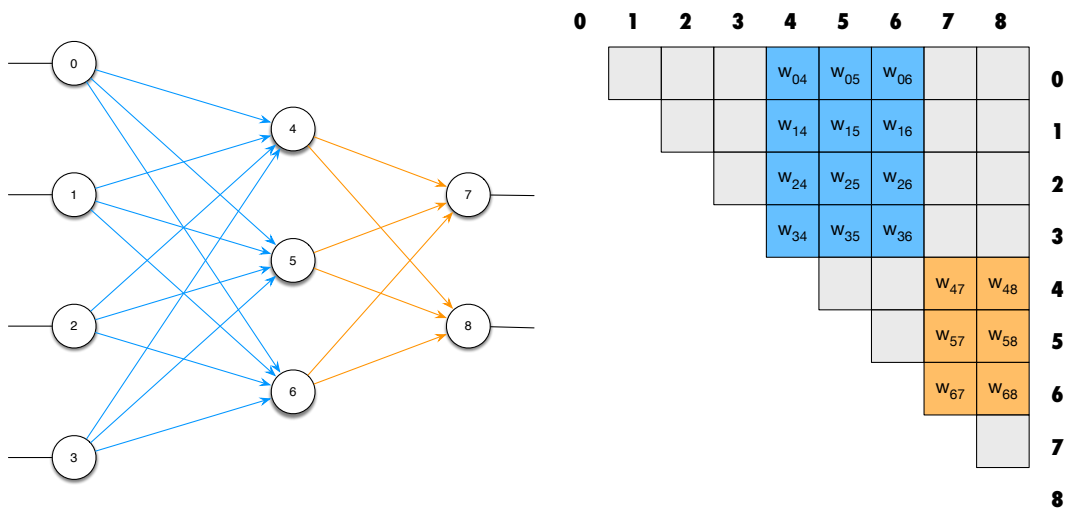


Figura 19. Representación de una RNA en una matriz superior

La opción de emplear matrices *sparse* para el almacenamiento de las conexiones no es viable ya que éstas solo tienen sentido usarlas si se cumplen dos condiciones. La primera condición es que se tenga miles de registros y la segunda es que como mucho se emplee un 10% de las posiciones. Como se pueden ver en las imágenes de ejemplo, no se cumple ninguna de las dos condiciones, ya que no se llega a tener tantos registros y el porcentaje de posiciones usadas no es tan bajo.

La aproximación que se ha desarrollado en esta tesis para las redes totalmente conectadas hacia delante es la de guardar las submatrices que se forman al guardar este tipo de redes y almacenarlo en un vector. En la figura 20 se puede ver en la parte superior las aproximaciones con las matrices, la matriz entera y la triangular, en la parte superior mientras que en la parte baja se ve el vector que se propone usar. Como se puede ver en la figura, el vector solo emplea las posiciones que realmente se

necesitan y en la tabla 2 se muestra las celdas empleadas para cada uno de los métodos.

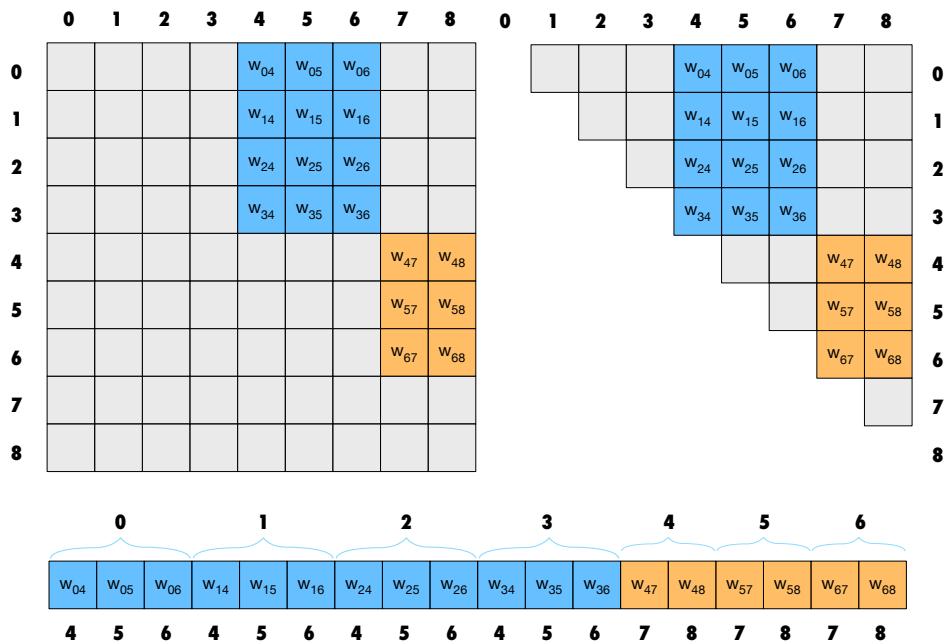


Figura 20. Representaciones para almacenar las conexiones

Tabla 2. Uso de memoria para las diferentes aproximaciones para almacenar las conexiones

Aproximación	Celdas	Ahorro
Matriz entera	81	-
Matriz triangular superior	36	55,5 %
Método propuesto	18	77,7 %

Implementación

El método que se propone, donde se almacena todas las conexiones de manera consecutiva, hace que se pierda una parte fundamental que tenían las matrices y son los índices ya que en las matrices cualquier conexión se podía obtener usando únicamente sus índices. Almacenar los datos en un vector supone tener que almacenar adicionalmente la posición de cada una de las conexiones. Así pues, para poder usar el método propuesto se propone el uso de las siguientes variables:

- **num layer**: número de capas que tiene la red (se incluye las capas de entrada, las de salida y las ocultas).
- **num neurons**: número de neuronas que tiene la red.
- **layer** (array): la posición i indica el número de elementos de procesado que hay en la capa i .

- **position** (array): indica a que capa pertenece el elemento de procesado i .
- **index** (array): indica la posición que ocupa el elemento de procesado dentro de la capa.
- **num weights**: número de pesos que tiene la red.
- **weight** (array): valor de las conexiones de la red.
- **stride** (array): la posición i indica donde empieza los pesos salientes de la neurona i .

A continuación, se detalla cómo obtener el tamaño de cada uno de los arrays necesarios:

- **layer**: El tamaño de este array viene dado por el valor de `num_layer`, es decir, el número de capas.
- **index** y **position**: El tamaño de estos arrays viene dado por el valor de `num_neurons`, es decir, el número de neuronas de la red.
- **stride**: El tamaño de este array viene dado por el número de neuronas que tienen conexiones salientes, es decir, todas las neuronas a excepción de la capa de salida.

$$size(stride) = num_neurons - layer[num_layer - 1]$$

- **weight**: El tamaño de este array viene determinado por el número de conexiones salientes que puede existir que viene determinado por la variable `num_weight`

$$num_weights = \sum_{i=0}^{num_layers-1} layer[i] \cdot layer[i + 1]$$

Una vez que tenemos guardada toda la información necesaria, solo queda saber cómo se puede acceder a un peso referenciándolo por la neurona presináptica y postsináptica. Esto se puede hacer usando la fórmula que aparece a continuación:

$$w_{ij} = weight[stride[i] + index[j]]$$

En la figura 21 se puede observar el valor que tendría cada uno de los vectores necesarios para usar una red con el método propuesto.

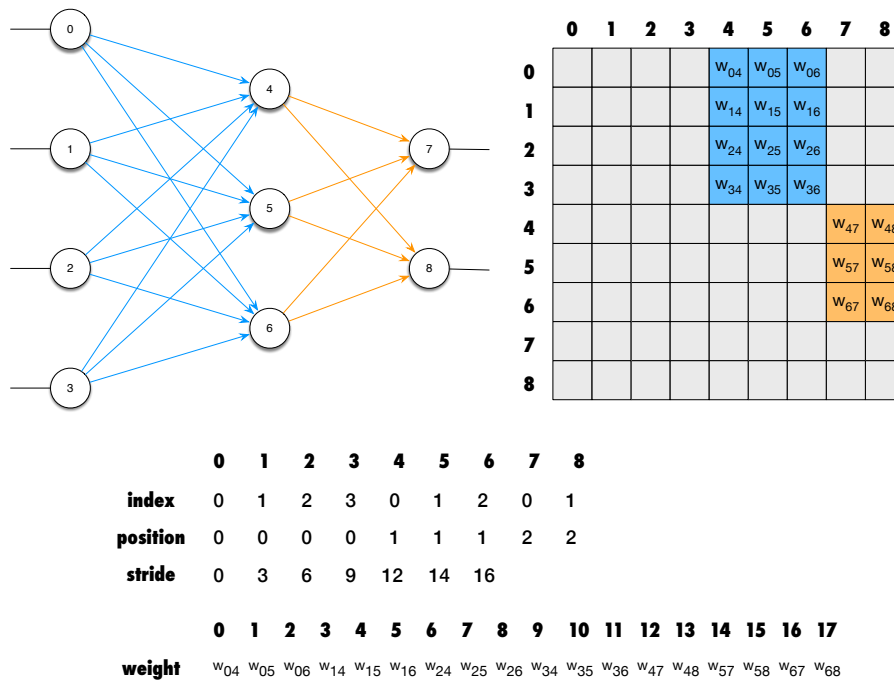


Figura 21. Red con los valores de los arrays para el método propuesto

Por último, queda por comprobar si toda esta información auxiliar que se necesita mejora el consumo de memoria que necesita, para ello se puede usar las siguientes fórmulas:

$$\text{Traditional approach} = (2 + \text{num_neurons}) \cdot \text{sizeof}(\text{int}) + \text{num_neurons}^2 \cdot \text{sizeof}(\text{float})$$

$$\begin{aligned} \text{Proposed approach} \\ = (3 + 3 \cdot \text{num_neurons} + \text{num_layers} + \text{layer}[\text{num_layers} - 1]) \cdot \text{sizeof}(\text{int}) \\ + \text{num_weights} \cdot \text{sizeof}(\text{float}) \end{aligned}$$

Codificación

A continuación, en la figura 22, se muestra como se inicializan las nuevas variables indicadas para emplear este método

```
POSITION = getArrayInt(NUM_NEURONS);
INDEX    = getArrayInt(NUM_NEURONS);

for (l = i = 0; i < NUM_LAYERS; i++) {
    for (j = 0; j < LAYER[i]; j++, l++) {
        POSITION[l] = i;
        INDEX[l] = j;
    }
}

STRIDE = getArrayInt(NUM_NEURONS - LAYER[NUM_LAYERS - 1]);
STRIDE[0] = 0;
for (i = 1; i < NUM_NEURONS - LAYER[NUM_LAYERS - 1]; i++)
    STRIDE[i] = STRIDE[i - 1] + LAYER[POSITION[i - 1] + 1];

NUM_WEIGHTS = 0;
for (i = 0; i < NUM_LAYERS - 1; i++)
    NUM_WEIGHT += LAYER[i] * LAYER[i + 1];
```

Figura 22. Inicialización de arrays para el método propuesto de almacenamiento de RNA

Para finalizar, en la figura 23 se muestra el método propuesto para calcular la salida de la red, donde se asume que la función de distribución es la función entidad.

```
first_pe = in_pe = 0;
for (l = 1; l < NUM_LAYERS; l++) {
    first_pe += LAYER[l - 1];
    for (in = 0; in < LAYER[l - 1]; in++, in_pe++) {
        pe = first_pe;
        for (out = 0; out < LAYER[l]; out++, pe++) {
            NET[pe] = NET[pe] + NET[in_pe] * WEIGHT[out + STRIDE[in-pe]];
        }
    }
    for (tmp = first_pe; tmp < pe; tmp++) {
        NET[tmp] = ACTIVATION_FUNCTION[tmp](NET[tmp]);
    }
}
```

Figura 23. Cálculo de la salida de la RNA usando el método propuesto

Beneficios al usar el método propuesto

El primer beneficio que se observa es un menor uso de memoria comparado con el método tradicional de usar la matriz entera para guardar los pesos. En la tabla 3 se indica la memoria que se ahorra en las redes que se emplearon para la realización de la presente tesis.

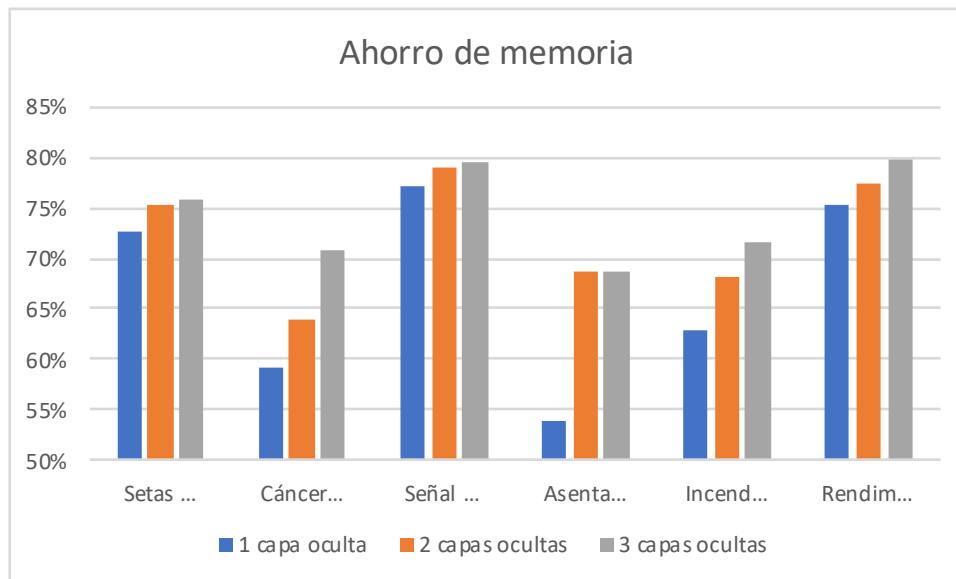


Figura 24. Ahorro de memoria al usar el método propuesto para guardar las redes

Tabla 3. Memoria en bytes para almacenar las diferentes redes

Data set	Estructura de red	Método tradicional	Método propuesto	Ahorro de memoria
Setas venenosas	22,7,1	3740	1028	72,51%
	22,7,4,1	4784	1180	75,33%
	22,12,7,4,1	8676	2104	75,75%
Cáncer de mama	9,7,1	1244	508	59,16%
	9,7,3,1	1704	616	63,85%
	9,12,8,4,1	4788	1400	70,76%
Señal de ionosfera	34,9,1	7940	1812	77,18%
	34,9,4,1	9432	1988	78,92%
	34,12,8,4,1	14188	2900	79,56%
Asentamiento de hormigón	7,4,1	644	296	54,04%
	7,4,3,1	984	308	68,70%
	7,4,5,3,1	1708	536	68,62%
Fuegos forestales	12,8,1	1868	692	62,96%
	12,8,5,1	2832	904	68,08%
	12,8,5,3,1	3508	996	71,61%
Rendimiento estudiantil	30,9,1	6580	1620	75,38%
	30,9,4,1	7944	1796	77,39%
	30,9,6,3,1	9828	2000	79,65%

Una de las mejoras que suele tener asociado el no emplear tanta memoria es una reducción en el tiempo de ejecución que se necesita para ejecutar la red. En la tabla 4 se muestran los resultados obtenidos de ejecutar un millón de patrones sobre la red en un Macbook Pro del 2017 con un procesador i5-7360U de 2.3GHz al usar el compilador clang-900.0.39.2. La forma en la que se muestran los tiempos es ejecutar diez veces cada red y tomar la media de las mediciones (Intel, 2017).

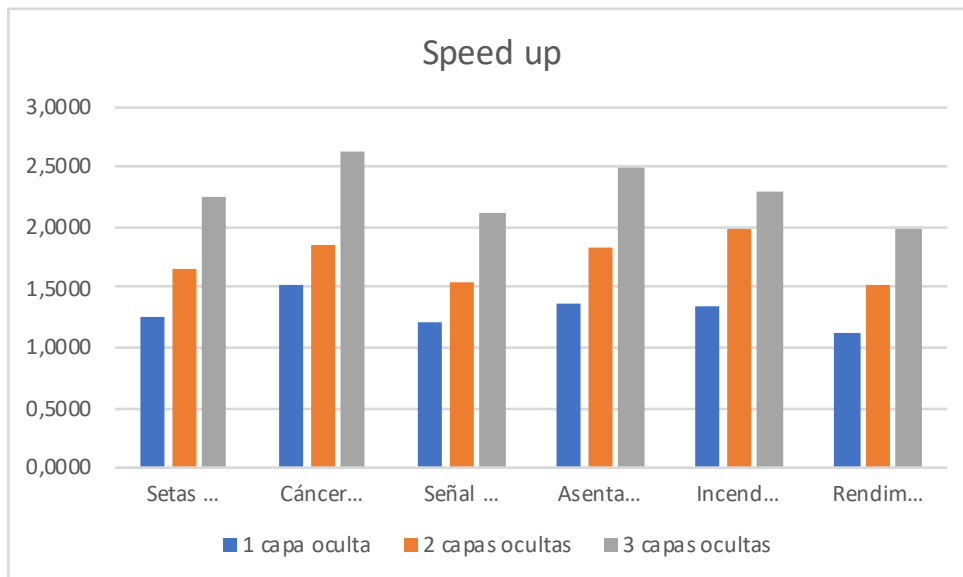


Figura 25. Comparación del speed up al usar el método propuesto frente al tradicional sin utilizar las opciones de optimización en el compilador

Tabla 4. Comparación de speed up usando el método tradicional y el propuesto sin opciones de optimización en el compilador

Data set	Estructura de la red	Método tradicional	Método propuesto	Speed up
Setas venenosas	22,7,1	0,9582	0,7665	1,2501
	22,7,4,1	1,468	0,888	1,6532
	22,12,7,4,1	3,7602	1,6684	2,2538
Cáncer de mama	9,7,1	0,5568	0,369	1,5089
	9,7,3,1	0,8677	0,4695	1,8481
	9,12,8,4,1	2,9614	1,1292	2,6226
Señal de ionosfera	34,9,1	1,681	1,3907	1,2087
	34,9,4,1	2,3799	1,5479	1,5375
	34,12,8,4,1	4,9259	2,3174	2,1256
Asentamiento de hormigón	7,4,1	0,2897	0,2138	1,3550
	7,4,3,1	0,5223	0,2862	1,8249
	7,4,5,3,1	1,04	0,4157	2,5018
Fuegos forestales	12,8,1	0,7722	0,577	1,3383
	12,8,5,1	1,4132	0,7142	1,9787
	12,8,5,3,1	1,8595	0,81	2,2957
Rendimiento estudiantil	30,9,1	1,5723	1,4018	1,1216
	30,9,4,1	2,301	1,5225	1,5113
	30,9,6,3,1	3,3071	1,6601	1,9921

Por último, queda por comprobar cómo funciona el método propuesto a la hora de usar las opciones de optimización del compilador para comprobar que nuestra mejora de tiempos la puede conseguir el compilador. Para ello, se han tomado los tiempos obtenidos al emplear las opciones de optimización del compilador. En concreto se usaron los parámetros: “-O3 -ffast-math -funroll-loops -ftree-vectorize -march=native”. Los resultados se pueden ver en la tabla 5.

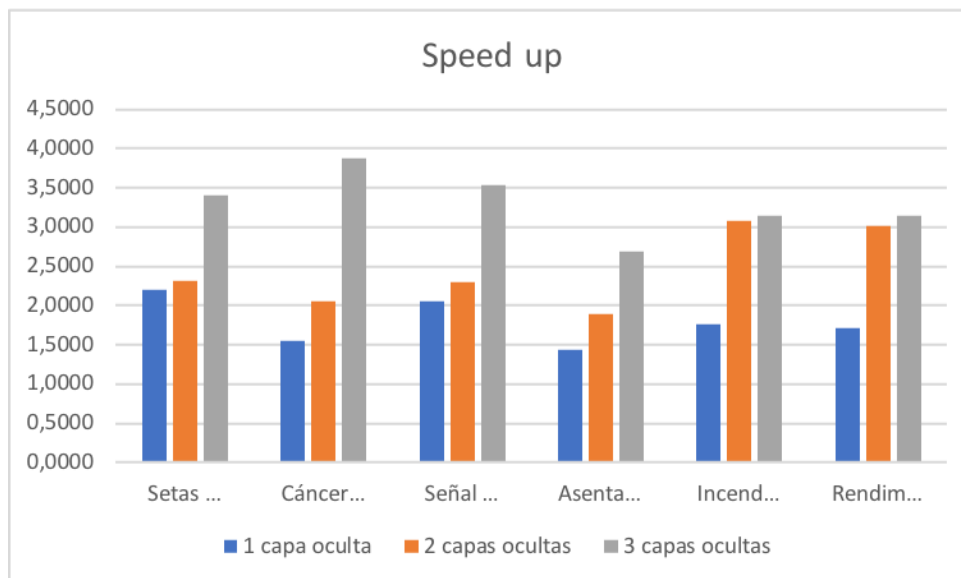


Figura 26. Comparación del speed up al usar el método propuesto frente al tradicional empleando las opciones de optimización en el compilador

Se puede concluir que, el método propuesto no siempre tiene un ahorro de memoria constante. Esto se debe a que depende mucho de la tipología escogida, ya que solamente se almacenan las conexiones que hay entre las neuronas y para una red con las mismas neuronas, dependiendo de su tipología puede tener diferente número de conexiones. Por ejemplo, para dos redes con estructuras 9,7,3,1 y 9,6,4,1 donde ambas tienen 20 neuronas, tienen un número de conexiones distinto, la primera tiene 87 conexiones y la segunda tiene 82 conexiones.

En lo referente a la mejora en la velocidad de ejecución hay que indicar que se consiguen beneficios y que, la mejora que se consigue con el método propuesto se debe a la organización, ya que el compilador no es capaz de conseguir igualarlo usando sus opciones de optimización, sino que el speed up guardando las conexiones en vector es mayor usando las optimizaciones de código.

Tabla 5. Comparación de speed up usando el método tradicional y el propuesto usando las opciones de optimización disponibles en el compilador

Data set	Estructura de la red	Método tradicional	Método propuesto	Speed up
Setas venenosas	22,7,1	0,3412	0,1548	2,2041
	22,7,4,1	0,414	0,1784	2,3206
	22,12,7,4,1	1,0623	0,3115	3,4103
Cáncer de mama	9,7,1	0,1459	0,0942	1,5488
	9,7,3,1	0,2295	0,1117	2,0546
	9,12,8,4,1	0,8527	0,2204	3,8689
Señal de ionosfera	34,9,1	0,5276	0,2569	2,0537
	34,9,4,1	0,6511	0,2823	2,3064
	34,12,8,4,1	1,4572	0,412	3,5369
Asentamiento de hormigón	7,4,1	0,0826	0,0578	1,4291
	7,4,3,1	0,1443	0,076	1,8987
	7,4,5,3,1	0,2802	0,1042	2,6891
Fuegos forestales	12,8,1	0,21	0,1187	1,7692
	12,8,5,1	0,4441	0,144	3,0840
	12,8,5,3,1	0,5226	0,166	3,1482
Rendimiento estudiantil	30,9,1	0,4407	0,2579	1,7088
	30,9,4,1	0,8223	0,2725	3,0176
	30,9,6,3,1	0,9892	0,3147	3,1433

Beneficios del método propuesto al combinar las redes con los AG

Una de las ventajas de este método es si se desea usar AG para buscar los valores adecuados de las conexiones de la red. Si se usa una codificación real no se requiere tiempo para codificar y decodificar los pesos ni tampoco es necesario usar el método que se usaba en el grupo RNASA-IMEDIR donde los operadores genéticos estaban hechos adhoc para los pesos de la red.

Con esta nueva propuesta solamente hay que poner todos los vectores de pesos de las redes en un único array de manera consecutiva. La forma de “descodificar” los pesos de un individuo es que el puntero de los pesos de la red apunte a posición del genotipo del individuo, es decir, solo se necesita realizar una asignación.

Esto proporcionó la ventaja de que se ha creado una librería para los AG que es completamente genérica y que no depende del problema que se quiera resolver, haciendo que esta sea fácilmente usable en otras aplicaciones y, además es fácilmente

extensible para añadir nuevas operaciones genéticas o usar otros tipos de algoritmos genéticos. De hecho, la propia librería que se ha implementado en C tiene diferentes versiones de AG.

4.1.3. Herramientas auxiliares

Para asegurarse de que los resultados obtenidos usando la aplicación de simulación para RNGA son independientes del orden y de los conjuntos de datos, se usa el método de *cross-validation* (Cantu-Paz & Kamath, 2005). Así se dispone de 10 conjuntos de datos donde cada uno incluye el 25% de datos para entrenamiento, el 50% para validación y el 25% restantes son para test. También se emplean 10 poblaciones genéticas distintas que constituyen la generación inicial y han sido creadas aleatoriamente. Esto implica que se tienen 100 ejecuciones para cada simulación debido a la configuración de la validación cruzada.

Generación masiva de archivos de parámetros

Generar 100 archivos de configuración que requieren las 100 ejecuciones para cada simulación puede ser una tarea repetitiva, aburrida y sobre todo puede generar configuraciones diferentes al generar archivos distintos. Para evitar ese problema se ha creado una herramienta en Java que pregunta por cada uno de los parámetros y, que para los parámetros que son distintos, permite la introducción de un patrón. Esto facilita introducir una única configuración de la simulación y la herramienta se encarga de generar automáticamente todos los archivos necesarios. Esta herramienta permite generar la configuración para la simulación donde el usuario introduce manualmente los parámetros de la glía y también para las simulaciones donde un AGCC busca de manera automática dichos parámetros de la glía. Además, en las simulaciones donde el usuario puede introducir manualmente los parámetros, la herramienta permite generar automáticamente distintos archivos para diferentes combinaciones de activación-iteración.

A mayores, esta herramienta también tiene una opción para crear la estructura del fichero de configuración para las poblaciones iniciales que necesita el AG.

Generación de informe de resultados

Al igual que se tienen los 100 archivos de configuración que requiere cada simulación, se obtienen como resultado 100 archivos que se corresponde con cada ejecución realizada del proceso de *cross-validation*. Para ello se creó otra herramienta en Java que permite leer de un fichero de propiedades, o preguntándole interactivamente al usuario, la configuración de cómo se quiere generar el log. Esto

acaba generando un único documento, una hoja de cálculo que resume los resultados obtenidos. Además, la herramienta permite la creación de gráficas, donde analiza los resultados de todas las ejecuciones que se emplearon durante el *cross-validation*, y genera las gráficas mediante las interpolaciones de todos los valores almacenados en los registros.

Cabe destacar que, en el caso de las simulaciones donde el usuario introduce los parámetros de la glía, se puede agrupar las ejecuciones donde se prueban distintos valores de activación-iteración para así generar un único documento resumido donde a mayores muestras la comparativa de las RNGAs con la RNA.

4.2. Algoritmos astrocíticos

Como aparece mencionado en el apartado de antecedentes computacionales: redes neurogliales artificiales, el comportamiento del astrocito está determinado por los parámetros de iteración (k) y activación (μ) y que, para cada patrón de entrada, independientemente si es de entrenamiento, validación o test, se le presenta a la red k iteraciones. El astrocito artificial registra la actividad de la neurona durante las k iteraciones aplicando la función $u : \mathbb{R} \rightarrow \mathbb{Z}$ sobre la salida de la neurona $y_j(t)$ donde la función u indica si la neurona se activó usando la siguiente fórmula:

$$u(x) \begin{cases} -1 & x \leq 0 \\ 1 & x > 0 \end{cases}$$

Por lo tanto, el astrocito monitoriza la actividad de la neurona en una ventana temporal de k iteraciones. Observando la actividad, el astrocito modificará las conexiones asociadas a las neuronas cuando el contador de activaciones alcance el valor μ . Aunque el astrocito podría modificar las conexiones de entrada y de salida, en esta tesis solo se modifican las conexiones salientes.

La modificación astrocítica de los pesos de las conexiones salientes de la neurona j , denotados como w_i , se define como $w_i(t + \Delta t) = w_i(t) + \Delta w_i(t)$ donde $\Delta w_i(t) = |w_i(t)|z(t)$ y $z : \mathbb{N} \setminus \{0\} \rightarrow \mathbb{R}$ es una función que se define de la siguiente manera

$$z(t) = \begin{cases} a & \text{si } r_j = \mu \\ -b & \text{si } r_j = -\mu \end{cases}$$

siendo $r_j : \mathbb{N} \setminus \{0\} \rightarrow [-\mu, \mu]$ la función que devuelve el número de veces que una neurona se ha activado. Si la neurona estuvo activa μ veces, los pesos de las conexiones, dependiendo de su signo, aumentarán o disminuirá en un porcentaje a , mientras que se reducirán/aumentarán en un porcentaje b si la neurona permaneció inactiva durante esas μ iteraciones.

Este es el funcionamiento básico que se puede asumir de un astrocito artificial, sin embargo, en esta tesis se han empleado 8 algoritmos astrocíticos que introducen alguna variación específica sobre la implementación.

4.2.1. Potenciación

Es conocido que en el SNC de los vertebrados el aprendizaje está relacionado con el refuerzo de la eficacia sináptica (que también se conoce como potenciación sináptica) (Bliss & Collingridge, 1993; Cooper, 2005). Cuando existe una actividad adecuada que dura varios milisegundos, incluso algunas que llegan a segundos, puede originarse un reforzamiento sináptico cuya duración puede ser de horas o incluso varios días.

Existen evidencias de que el glutamato liberado por los astrocitos potencia la fuerza sináptica, que viene dada por el cambio de potencial de membrana que ocurre cuando se activan los receptores de neurotransmisoras postsinápticos, en diferentes sinapsis y además se hace con distintos mecanismos. La recepción de receptores presinápticos de glutamato activados por el glutamato liberado por los astrocitos incrementan la frecuencia de corrientes postsinápticas excitatorias evocadas y espontáneas EPSCs lo que provoca un aumento en la probabilidad de liberación de neurotransmisores (Fiacco & McCarthy, 2004; Perea & Araque, 2007).

Efecto de atenuación astrocítica

Este algoritmo tiene un comportamiento específico en la función $z(t)$ que es la que indica cómo se modifican las conexiones salientes de la red y cuya definición es la siguiente:

$$z(t) = \begin{cases} a & r_j(t) = \mu \\ -b & r_j(t) = -\mu \\ 1 & \text{en otro caso} \end{cases}$$

Recordemos que $r_j: \mathbb{N} \setminus \{0\} \rightarrow [-\mu, \mu]$ es la función que devuelve el número de veces que una neurona se ha activado. Si la neurona estuvo activa μ veces, los pesos de las conexiones se verán incrementados en un porcentaje a , mientras que éstos se decrementarán en un porcentaje b si la neurona permaneció inactiva durante esas μ iteraciones.

$$r_j(t) = \begin{cases} u(y_j(t)) + r_j(t-1) & t > 0, \quad r_j(t-1) \in (-\mu, \mu) \\ r_j(t-1) & t > 0, \quad r_j(t-1) \in \{-\mu, \mu\} \\ u(y_j(t)) & \text{en otro caso} \end{cases}$$

Un comportamiento interesante en este algoritmo es que cuando una neurona alcanza el nivel de activación $\{-\mu, \mu\}$ el contador no se resetea a cero como consecuencia de la excitación del astrocito. El comportamiento consigue tener un refuerzo que perdura en el tiempo de las neuronas que más se activan y hacen que el astrocito esté actuando más tiempo y que su efecto desaparezca gradualmente con el tiempo. Esto se consigue porque una vez que una neurona alcanza el nivel de activación necesaria para excitar a un astrocito éste modifica el peso de las conexiones y permite que en la siguiente iteración la neurona pueda volver a excitar el astrocito para que este continúe modificando los pesos.

Activaciones no consecutivas y modificación de conexiones sin límite

El comportamiento de este algoritmo es que se permite modificar las conexiones, es decir, los valores de los pesos y éstos pueden alcanzar cualquier valor. En lo que se refiere a la forma de contar las activaciones no hay una restricción de que tengan que ser consecutivas, de tal manera que las activaciones de las neuronas pueden estar dispersas durante las iteraciones que se le presente el mismo dato de entrada. Por lo tanto, la función $r_j(t)$ queda de la siguiente manera:

$$r_j(t) = \begin{cases} u(y_j(t)) + r_j(t-1) & t > 0, \quad r_j(t-1) \in (-\mu, \mu) \\ u(y_j(t)) & \text{en otro caso} \end{cases}$$

Hay que mencionar que, mientras que el algoritmo anterior no reiniciaba el contador de activaciones al llegar, éste algoritmo astrocítico resetea el valor de activación de las neuronas cuando el astrocito es excitado.

Efecto de proceso global

Este algoritmo tiene la particularidad de que en lugar de procesar cada instancia durante k iteraciones considera todas las instancias como un conjunto y el astrocito procesa todo el conjunto durante las k iteraciones. La idea detrás de este algoritmo está basado en cómo se procesa la información visual por el cerebro el cual recibe un conjunto de imágenes y las procesa como un conjunto y extrae las características de las imágenes para dar a la secuencia un significado. Los demás algoritmos consideran

cada imagen independientemente y tratan de extraer las características de cada imagen de forma separada.

La manera en la que este algoritmo modifica las conexiones es la siguiente:

$$w_i(t + \Delta t) = w_i(t) + \Delta w_i(t)$$

donde

$$\Delta w_i(t) = \frac{n + \operatorname{sgn}(w_{ji}(t))s}{n}$$

En esta fórmula el valor n indica el número de instancias que hay en el conjunto de datos mientras que s indica el número de veces que la neurona se activó durante las k iteraciones. Una parte peculiar de este algoritmo es que la modificación de las conexiones es directamente proporcional a las veces que la neurona se activa.

Activaciones consecutivas y modificación de conexiones con límite

La particularidad de este algoritmo es que solo contempla las activaciones de una neurona si esta se realiza de manera consecutiva y, si la neurona alcanza el valor de actividad o inactividad hace que el astrocito actúe y el contador de activaciones es reseteado. El nivel de actividad neuronal sigue las restricciones descritas en la siguiente fórmula:

$$r_j(t) = \begin{cases} u(y_j(t)) + r_j(t-1) & t > 0, u(y_j(t)) = u(y_j(t-1)), r_j(t-1) \in (-\mu, \mu) \\ u(y_j(t)) & \text{en otro caso} \end{cases}$$

Además, este algoritmo los astrocitos tiene en cuenta el valor límite de los pesos que las conexiones pueden alcanzar de tal manera que la modificación de pesos se calcula de la siguiente manera:

$$w_i(t + \Delta t) = \min\{w_i(t) + \Delta w_i(t), wM_i\}$$

Activaciones no consecutivas y modificación de conexiones con límite

El comportamiento de los astrocitos artificiales en este algoritmo es permitir que las activaciones de la neurona que monitoriza pueda están dispersas por las k iteraciones y, aun así, exciten el astrocito.

En cuanto a la modificación de las conexiones que puede provocar el astrocito, éste tiene que respetar el valor máximo que se estable que los pesos pueden alcanzar.

Activaciones consecutivas y modificación de conexiones sin límite

Con este algoritmo, el astrocito solo se excitará si la neurona que monitoriza alcanza las μ activaciones de manera consecutiva. A mayores, este algoritmo permite que los valores que alcanzan los pesos tras las modificaciones en las conexiones provocadas por el astrocito no tengan un valor límite.

4.2.2. Depresión

La disminución de la amplitud de los potenciales postsinápticos en respuesta a sucesivos impulsos presinápticos y que se produce como resultado de cambios en la cantidad de neurotransmisores liberados por cada impulso presináptico se conoce como depresión sináptica. Además de que los astrocitos biológicos participen en la potenciación sináptica se cree que pueden participar en la depresión sináptica. Esto se debe a que se ha podido observar que el glutamato que librea los astrocitos puede causar depresión sináptica y un ejemplo de ello se da al activar los receptores de glutamato del grupo II/III en los terminales inhibitorios del hipocampo (Liu, Xu, Arcuino, Kang, & Nedergaard, 2004).

Para simular este comportamiento de depresión sináptica se ha implementado un algoritmo similar al de atenuación astrocítica. La diferencia está en que en este caso se reducirá en un porcentaje b el peso de las conexiones salientes de la neurona j únicamente cuando el contador de activaciones alcance el umbral μ . Las funciones matemáticas que afectan a este algoritmo se indican a continuación:

$$z(t) = \begin{cases} -b & r_j(t) = \mu \\ 1 & \text{en otro caso} \end{cases}$$

Este comportamiento implica que las conexiones de las neuronas que más se activen serán las que más se debiliten.

4.2.3. Receptores de endocannabinoides

En el hipocampo los endocannabinoides generan una alta actividad sináptica, lo que provoca una depresión en las sinapsis cercanas debido a la despolarización de las neuronas. A mayores, esta actividad sináptica hace que los astrocitos potencien las sinapsis de las conexiones más lejanas al variar las concentraciones de calcio astrocítico en astrocitos adjuntos.

Para simular este comportamiento en el que están involucrados los receptores CB₁ de los astrocitos se implementó un algoritmo donde el astrocito que monitoriza la activación de una neurona tiene un efecto de depresión en las conexiones de la propia neurona, pero a su vez potencia las conexiones de las demás neuronas que se encuentran en la misma capa.

La principal diferencia de este algoritmo, con los mostrados anteriormente es que no afecta únicamente a las conexiones salientes de la neurona j que se ha activado, sino que también afecta a todas las neuronas que forman parte del subconjunto c , que son las neuronas pertenecientes a la misma capa que la neurona j . El algoritmo de modificación de pesos quedaría de la siguiente manera $w_{li}(t + \Delta t) = w_{li}(t) + \Delta w_{li}(t)$ donde $\Delta w_{li}(t) = |w_{li}(t)|z_l(t)$ y $z: \mathbb{N} \setminus \{0\} \rightarrow \mathbb{R}$ es una función que se define de la siguiente manera:

$$z_l(t) = \begin{cases} b & l \neq j, l \in c, r_j(t) = \mu \\ a & l = j, r_j(t) = \mu \\ 1 & \text{en otro caso} \end{cases}$$

Recordemos que $r_j: \mathbb{N} \setminus \{0\} \rightarrow [-\mu, \mu]$ es la función que devuelve el número de veces que una neurona se ha activado. Si la neurona estuvo activa μ veces, los pesos de las conexiones se verán incrementados en un porcentaje a , mientras que éstos se decrementarán en un porcentaje b si la neurona permaneció inactiva durante esas μ iteraciones.

$$r_j(t) = \begin{cases} u(y_j(t)) + r_j(t-1) & t > 0, \quad r_j(t-1) \in (-\mu, \mu) \\ r_j(t-1) & t > 0, \quad r_j(t-1) \in \{-\mu, \mu\} \\ u(y_j(t)) & \text{en otro caso} \end{cases}$$

Un comportamiento interesante en este algoritmo es que cuando una neurona alcanza el nivel de activación μ el contador no se resetea a cero como consecuencia de la excitación del astrocito. El comportamiento tiene un efecto que se mantiene en el tiempo de los astrocitos que monitorean a las neuronas. Este comportamiento se trata de simular el comportamiento biológico de los astrocitos que se excitaban por la despolarización neuronal estos volvían a responder en las siguientes despolarizaciones.

4.3. Metodología

Para poder seguir una correcta planificación, diseño y desarrollo se decidió usar metodologías de software y creación de RNA y RNGA que se detallarán a continuación. En lo referente a la realización de los experimentos se ha recurrido al uso de la validación cruzada para garantizar la independencia de los conjuntos de datos. Además, para el análisis de los resultados obtenidos se ha aplicado el test de wilcoxon de rangos signados (Wilcoxon, 1945).

4.3.1. Metodología del software

Para el análisis, diseño y desarrollo tanto del software de simulación de RNGA y RNA, como aplicación web para disponer de un entorno sencillo para realizar las ejecuciones se ha seguido el enfoque basado por la metodología SCRUM (Pohl & Hof, 2015). De forma progresiva las aplicaciones se fueron construyendo creando entregables en cada uno de los sprints que se llevaron a cabo teniendo una herramienta con la que ir probando las simulaciones en etapas muy cercanas. Todos los sprints generaban aplicaciones listas, en ningún momento eran prototipos, con las que se permitió testear la validez del software diseñado, sin tener que esperar a la última etapa del desarrollo software.

Frente a otros modelos, esta metodología permite un desarrollo ágil de las aplicaciones y además una gran flexibilidad para incorporar nuevas características que se descubren que son necesarias que se incorporen en el software.

4.3.2. Metodología de diseño y construcción de los sistemas conexionistas

Para las etapas de diseño y construcción de los dos tipos de redes, la RNA y RNGA, se ha empleado una metodología que ha sido desarrollada por el grupo de investigación RNASA-IMEDIR. El uso de esta metodología es que no existe ninguna formalmente aceptada científicamente y el grupo tiene una gran experiencia en RNA.

Dentro del desarrollo de SC no existe ninguna metodología bien definida, pero es aconsejable seguir un proceso meticuloso para evitar saltarse pasos que pongan en riesgo la evolución del desarrollo. La metodología está basada en la propia experiencia y en el uso de otras disciplinas, como la ingeniería de conocimiento, que tienen una estrecha relación (Alonso-Betanzos & Guijarro-Berdiñas, 2004). La metodología se define en los siguientes pasos:

- Identificación del problema a resolver
- Adecuación – Justificación del uso del SC

- Diseño y construcción del SC neuroglial
- Preparación de los datos y construcción de los conjuntos de datos.
- Entrenamiento del SC
- Organización y análisis de resultados
- Ajuste de parámetros
- Validación del sistema.

4.4. Validación de resultados

Muchos estudios usan técnicas de computación evolutiva combinadas con redes neuronales donde usan los algoritmos evolutivos para entrenar las redes o incluso diseñar su estructura (Castillo et al., 2003; Yao, 1999). Muchos estudios usan una pequeña combinación de datos de pruebas y muy pocos comparan diferentes combinaciones de algoritmos evolutivos y redes del mismo dominio (Grönroos, 1998; Siddiqi & Lucas, 1998). Sin embargo, muchas de las comparaciones que se realizan no se hacen de la manera adecuada o no usan test estadísticos apropiados que apoyen las conclusiones que se realizan (Cantu-Paz & Kamath, 2005).

En la presente tesis se ha empleado la técnica de *cross-validation* 10cv para estimar cuan buenos son los algoritmos desarrollados (Cantu-Paz & Kamath, 2005). Esta técnica se emplea para evitar que los resultados de un problema estén influenciados por los conjuntos de datos, es decir, esto se realiza para comprobar hasta qué punto los resultados son deudores de un determinado conjunto de patrones.

El método de *cross-validation* consiste en dividir un conjunto de datos D en k subconjuntos no solapados lo que se denomina *k-fold cross-validation*. Por cada uno de esos subconjuntos $i \in [1, k]$ se entrena el método usando un subconjunto de datos para entrenamiento D_{tr} , otro subconjunto de datos D_{va} para validación y un último subconjunto D_{te} para la fase de test, donde ninguno de los tres subconjuntos se solapa y la unión de los tres hacen el conjunto total de los datos. Hay que tener en cuenta lo que algunos estudios indican un ligero aumento del error de tipo I (falso positivos) al comparar algoritmos de clasificación usando los *t-test* de *cross-validation* (Herrera, Hervás, Otero, & Sánchez, 2004).

En la presente tesis se realiza 10 ejecuciones de un test con diferentes permutaciones de los datos. En todos los casos, la base de la comparación es la precisión media de los 10 test realizados usando el método de *cross-validation*. Además, para que no exista dependencia en la inicialización de las redes, se emplean 10 poblaciones iniciales distintas y, cada una de ellas, realizan los 10 test. Al final, en este trabajo se analizan 100 test distintos para cada red.

Mientras que hay métodos de *cross-validation* como 5x2cv donde los patrones se dividen en dos mitades generando un subconjunto de entrenamiento y otro subconjunto para test en este trabajo la validación separa los datos en tres conjuntos para tener un conjunto de validación (Alpaydin, 1998). El reparto para todos los datos generados ha sido el 25% para entrenamiento el 25% de test y los datos restantes se emplearon para validación. La razón por la que se añade este subconjunto de validación es para poder evitar los problemas de sobreentrenamiento que se generan al tener solo conjuntos de entrenamiento y test.

4.5. Análisis estadístico de los datos

El análisis estadístico se ha realizado usando el test de Wilcoxon de rangos asignados sobre los resultados obtenidos para la comparación de RNA y RGA (Mesejo et al., 2015; Porto-Pazos et al., 2011; Wilcoxon, 1945). Aplicando un contraste de hipótesis sobre este estadístico se puede saber si la comparación de los resultados obtenidos por las redes es estadísticamente significativa.

La hipótesis que se contrasta se denomina Hipótesis Nula (H_0) y la que se acepta si se rechaza H_0 es la Hipótesis Alternativa (H_1). La medida de discrepancia entre H_0 y la información de la muestra se conoce como estadístico de contraste y debe seguir una distribución conocida cuando H_0 es cierta de modo que se puede distinguir una discrepancia grande que tenga probabilidad muy pequeña de ocurrir cuando H_0 sea cierta y una discrepancia pequeña que tenga una probabilidad grande de ocurrir cuando H_0 sea cierta. A continuación, se debe indicar que discrepancias no se consideran admisibles cuando H_0 sea correcta, dicho de otra manera, se debe fijar un valor de significación del contraste, que comúnmente se denota por α , donde $\alpha = P(\text{rechazar } H_0 | H_0 \text{ es cierta})$.

Para finalizar, se debe calcular el valor del estadístico \hat{d} asociado a una muestra tomada de manera que si es \hat{d} es pequeño se acepta H_0 , porque pertenece a la región de aceptación, y si \hat{d} es grande se rechaza H_0 , porque pertenece a la región de rechazo.

Se denomina *p-valor* como la probabilidad p de obtener una discrepancia con H_0 mayor o igual que el valor crítico \hat{d} cuando es H_0 correcta. Esto indica que si $\alpha > p - \text{valor}$ entonces se indica que se ha de rechazar H_0 . Comúnmente de manera orientativa un *p-valor* inferior a 0,01 supone rechazar H_0 , si es superior a 0,1 implica aceptar H_0 y si se encuentra entre ambos valores se considera que no es determinante.

En el estudio la hipótesis nula sería: “la media de los errores obtenidos con RNA es menor o igual a la media de las precisiones obtenidas con RGA”. En este caso las muestras son pareadas ya que aparecen como distintas observaciones realizadas sobre

los mismos datos. Concretamente, en la presente tesis, las observaciones pareadas consisten en considerar las 100 combinaciones, que resultan de la validación cruzada de 10 conjuntos y 10 poblaciones, a las que por un alazo se les aplica una RNA y por otro lado se les aplica una RNGA como sistemas de resolución. Si se denomina d_i a la diferencia entre las observaciones antes y después de incluir el astrocito artificial ($d_i = rna_i - rnga_i$) siendo μ la media de la muestra, H_0 sería igual a:

$$H_0: \mu_{rna} = \mu_{rnga} \leftrightarrow \mu_{rna} - \mu_{rnga} = 0 \leftrightarrow \mu_d = 0$$

Si se quiere contrastar que los resultados obtenidos con RNGA suponen una mejora o no (H_0) se pueden emplear diferentes test de hipótesis y en este trabajo se usara el test de Wilcoxon de rangos signados.

4.5.1. Test no paramétrico de Wilcoxon de rangos signados.

El test de Wilcoxon es una técnica no paramétrica que resulta apropiada cuando no se puede suponer la suposición de normalidad (Wilcoxon, 1945). Para usar test se debe disponer de muestras bidimensionales pareadas, es decir, se necesita tener n parejas de valores (x_i, y_i) , que en el caso que interesa en esta tesis, las parejas son los valores medidos por las dos redes (la RNA y la RNGA). Hay que destacar que en algunas ocasiones, las hipótesis necesarias para el test paramétrico (normalidad de las diferencias pareadas, d_i) no se verifican y es rigurosamente imprescindible hacer el contraste que se presenta a continuación (Moore, 1999; Walpole, Myers, & Myers, 1999):

1. Ordenar todos los valores absolutos de las diferencias ($|d_i|$) entre las observaciones pareadas de menos a mayor.
2. Asignar la diferencia en valor absoluto más pequeña rango 1, a la segunda más pequeña rango 2 y así sucesivamente. Si el valor absoluto de dos o más diferencias es el mismo se asignará a cada uno el promedio de los rangos que se asignarían si las diferencias fuesen diferentes.
3. A cada rango ($r(|d_i|)$:valor del orden del dato en el conjunto) se le pone el signo de la diferencia.
4. Se calcula la suma de rangos positivos (w_+) y la de rangos negativos (w_-).
5. La suma de los rangos con diferencias negativas omitido su signo es el estadístico de contraste que se suele denotar con la letra W (aunque en ocasiones aparece con la letra T).

Finalmente, las hipótesis manejadas por este contraste para indicar si existen diferencias entre las muestras son:

H_0 : No hay diferencias entre las medianas de las muestras pareadas ($Med_{RNGA} = Med_{RNA}$)
 H_1 : Si hay diferencia entre las medianas de las muestras pareadas ($Med_{RNGA} < Med_{RNA}$)

Cuanto más grande sea el sumatorio con respecto a la otra muestra, más grande será la evidencia de que las dos poblaciones son distintas (Morillo, 2008). En este caso, si w_+ es pequeña y w_- es grande la hipótesis nula se puede rechazar a favor de la alternativa.

Como se puede observar, la medida a contrastar con este test es la mediana. La razón de usar la mediana es que es una medida de posición central robusta y que tiene poca sensibilidad a la presencia de valores atípicos observados en las muestras.

4.6. Entorno de simulación

Una parte importante en la solución propuesta es la facilidad para poder tener los resultados de las simulaciones. La propuesta que tenía anteriormente el grupo RNASA-IMEDIR implicaba que primero se ejecutaba la fase de entrenamiento y validación de las RNGA. A continuación, era necesario ejecutar un script en Matlab para poder tener un informe de los resultados. Para la fase de test era necesario usar otro script para seleccionar que red es la que se debe usar. También se tenía que generar un archivo de parámetros para finalmente ejecutar la fase de test. Por último, era necesario estructurar los datos para poder tener otro informe con los datos de test.

Con la nueva propuesta, todo se ha vuelto más simple. Ahora solo es necesario preparar los datos de entrada del programa y este se encarga de realizar todas las etapas del SC, siendo únicamente necesario ejecutar un último programa en java con los resultados para poder tener un único informe en Excel con toda la información relevante.

Una parte relevante para el entorno de las simulaciones es el número de ejecuciones necesarias a realizar debido al uso de la técnica de *cross-validation*, por ese motivo se sugiere ejecutar las simulaciones en un entorno de HPC. La infraestructura de HPC empleada fue el *Finisterrae2* del Centro de Supercomputación de Galicia (CESGA, 2016). Actualmente el *Finisterrae2* cuenta con SLURM, un sistema de gestión de recursos, que permite usar un sistema de colas para la ejecución de programas (SLURM, 2018). La infraestructura cuenta actualmente con 400 nodos de cómputo que están organizados por particiones a las que hay que asociar los trabajos usados. La partición sugerida para estos trabajos es la *shared* ya que es la que permite mantener un mayor número de trabajos en cola (300 trabajos) y también es la que

permite tener un mayor número de trabajos ejecutándose a la vez (hasta 200 trabajos).

A continuación, se detallan los scripts utilizados para lanzar las simulaciones y la propuesta de una estructura de directorios para la ejecución de simulaciones.

Estructura de directorios

Para la realización de un gran número de pruebas es mejor tener una estructura organizada de directorios para el correcto funcionamiento de las simulaciones. Para ello se propone usar la siguiente estructura de directorios:

- **`\${HOME}/bin** En este directorio se encuentran alojados los programas necesarios para la ejecución de las simulaciones. Además, es recomendable añadir esta ruta en el *path* para poder usar los programas en cualquier directorio sin indicar la ruta hacia la aplicación.
- **database** En esta carpeta, que se puede situar donde se desee, tiene incluidos las siguientes subcarpetas:
 - **dataset** Dentro de esta carpeta estará una carpeta por cada problema de datos que se emplee. Dentro se propone usar 10 subdirectorios para cada división proveniente de la técnica de *cross-validation* que tendrán los conjuntos de entrenamiento, validación y test.
 - **population** Dentro de esta carpeta se alojarán las poblaciones que usará el AG. Se propone poner dentro de una subcarpeta las que son para las conexiones de la red y en otra carpeta distinta las que son para los parámetros de la glía.
- **sim** En este directorio se alojará el script que enviará los trabajos a la cola de ejecución del CESGA. Se recomienda usar el nombre del problema a resolver y un identificador indicando el número de prueba que se realiza, por ejemplo *Cancer-123*. Dentro de este directorio se encuentra el subdirectorio **params** que contiene los ficheros de parámetros de entrada para el programa que realizará la simulación.
- **`\${STORE}/angn/sim/results** Este directorio se empleará para guardar los resultados de la simulación y lo creará automáticamente el script de ejecución, donde *sim* es el nombre de la carpeta que contiene la configuración de una simulación específica. La razón de usar *`\${STORE}* es no llegar al límite de la cuota que tiene el CESGA en el directorio *`\${HOME}*.

Script de ejecución para RNA y RNGA

Para ejecutar las simulaciones, se emplea un script de bash con las variables adecuadas para lanzar el *cross-validation* y para poder probar diferentes combinaciones de activación iteración de la RNGA y también añadir una RNA para después comparar los resultados. En *MIN_POP* y *MAX_POP* se indican los índices mínimo y máximo de las poblaciones respectivamente mientras que en *MIN_SET* y *MAX_SET* representa los valores para los índices de los conjuntos de datos. Estos valores de los índices están relacionados con la validación cruzada que se realiza en las simulaciones. El array *AC* e *IT* representan los valores de activación e iteración que tendrá la red. Como podemos ver en el atributo *-p* del comando *sbatch* se emplea la partición *shared* y la calidad del servicio y el tiempo de ejecución se indican en las variables *QOS* y *TIME* respectivamente. Estos dos valores están puestos como variables porque dependiendo del valor que se les de los trabajos tendrán otorgados un nivel de prioridad distinta, de tal manera que, si usamos la calidad de servicio más restrictiva, y ponemos el menor tiempo posible, los trabajos entrarán antes a ejecutarse.

Si hacemos el cálculo del número de ejecuciones que realizará el script tenemos 500, lo que supera los 300 trabajos que permite el CESGA tener su cola. Para ello, el script agrupa las ejecuciones por combinaciones de activación-iteración y por las poblaciones para lanzar los trabajos. Con esta agrupación el script envía 50 trabajos a la cola, lo que nos permitiría enviar 6 simulaciones para ejecutar en los nodos de cómputo.

```

#!/bin/bash

MIN_POP=1
MAX_POP=10

MIN_SET=1
MAX_SET=10

declare -a AC=(0 2 2 3 3)
declare -a IT=(0 4 6 6 8)

PROG=energlia
OPP=ann_fixed_glia

TIME="12:00:00"
QOS="shared_short"

PRE=Cancer
DIR_CONFIG=params/
DIR_OUT=$STORE/angnClassification/`basename $PWD`/results/

mkdir -p ${DIR_OUT}

for ((i = 0; i < ${#AC[@]}; i++))
do
    for p in `seq ${MIN_POP} ${MAX_POP}`
    do
        TMP_SCRIPT=${DATASET}-p${p}-ac${AC[$i]}it${IT[$i]}

        echo "#!/bin/bash" > ${TMP_SCRIPT}.sh
        echo "" >> ${TMP_SCRIPT}.sh
        echo "module load intel/2018.3.222" >> ${TMP_SCRIPT}.sh

        for d in `seq ${MIN_SET} ${MAX_SET}`
        do
            TMP=${PRE}_p${p}d${d}ac${AC[$i]}it${IT[$i]}

            echo "" >> ${TMP_SCRIPT}.sh
            echo "printf \"\${PROG} \${OPP} --config=\${DIR_CONFIG}\${TMP}.xml \"" >> ${TMP_SCRIPT}.sh
            echo "echo \"--out=\${DIR_OUT}\${TMP}.xml\"" >> ${TMP_SCRIPT}.sh
            echo "\${PROG} \${OPP} --config=\${DIR_CONFIG}\${TMP}.xml --out=\${DIR_OUT}\${TMP}.xml" >> ${TMP_SCRIPT}.sh

        done

        sbatch -p shared --qos=${QOS} --time=${TIME} --mem=5G -c 1 --nodes=1 ${TMP_SCRIPT}.sh

        rm ${TMP_SCRIPT}.sh
    done
done

```

Figura 27. Script para enviar una simulación al Finisterrae2 donde el usuario establece los parámetros de la glía

Script de ejecución para RNGA con AGCC

Al igual que el script anterior, este dispone de las variables adecuadas para emplear la técnica de validación cruzada y, además, las variables son las mismas que en el caso contrario. En este script, no se incluyen las combinaciones de iteración y activación de la red, porque es el propio algoritmo genético el que se encarga de buscar los valores adecuados.

```
#!/bin/bash

MIN_POP=1
MAX_POP=10

MIN_SET=1
MAX_SET=10

PROG=energla
OPP=angn_search_glia

TIME="08:00:00"
QOS="shared_short"

PRE=Cancer
DIR_CONFIG=params/
DIR_OUT=$STORE/angnClassification/`basename ${PWD}`/results/

mkdir -p ${DIR_OUT}

for p in `seq ${MIN_POP} ${MAX_POP}`
do
    for d in `seq ${MIN_SET} ${MAX_SET}`
    do
        TMP=${PRE}_p${p}d${d}

        echo "#!/bin/bash" > ${TMP}.sh
        echo "" >> ${TMP}.sh
        echo "module load intel/2018.3.222" >> ${TMP}.sh

        echo "" >> ${TMP_SCRIPT}.sh
        echo "printf \"${PROG} ${OPP} --config=${DIR_CONFIG}${TMP}.xml \"" >> ${TMP_SCRIPT}.sh
        echo "echo \"--out=${DIR_OUT}${TMP}.xml\"" >> ${TMP_SCRIPT}.sh
        echo "${PROG} ${OPP} --config=${DIR_CONFIG}${TMP}.xml --out=${DIR_OUT}${TMP}.xml" >> ${TMP_SCRIPT}.sh

        sbatch -p shared --qos=${QOS} --time=${TIME} --mem=5G -c 1 --nodes=1 ${TMP_SCRIPT}.sh

        rm ${TMP}.sh
    done
done
```

Figura 28. Script para enviar una simulación al Finisterrae2 donde el AGCC es el encargado de establecer los parámetros de la glía

Generación automática de informes sobre los resultados generados

Como se mencionó anteriormente, cada una de las simulaciones necesita ejecutar un el programa de simulación de las redes varias veces para usar *cross-validation*. En concreto para cada simulación se realizan 100 ejecuciones (10 poblaciones x 10 conjuntos de datos). Esto genera 100 archivos de logs. En el caso donde el usuario indique los parámetros de la glía tendremos 5 simulaciones, con lo que tendremos 500 ficheros de log, es decir, 100 ficheros por cada combinación de iteración activación. Mirar todos estos archivos uno a uno, es una tarea muy lenta, y no permite ver la información global obtenida por las simulaciones lo que realizarlo manualmente es algo inviable. Para ello, se puede emplear la herramienta de generación de informes, que analiza todos los archivos para crear una hoja de cálculo Excel, que se puede ser ejecutada desde cualquier SO, lo que implica que incluso puede ser ejecutada desde el CESGA.

El análisis de los resultados se puede configurar. El test se puede analizar toda la evolución del AG, o limitarlo tanto por número de generación o tiempo consumido. Además, se puede indicar que seleccione, dentro del rango de búsqueda que se

especifique, el mejor o el último individuo. Como aliciente, se puede indicar que de los X mejores individuos seleccione el más óptimo de test.

En el caso de que se genere el informe sobre la simulación en la que se especifica manualmente los parámetros de la glía, estos se pueden agrupar para generar un informe conjunto que compare cada combinación de la RNGA con la RNA. Un ejemplo del informe generado automáticamente se puede ver en la figura 29.

	GLIA										SIN GLIA									
	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10
P1	0.076246	0.061404	0.058551	0.049708	0.352952	0.228577	0.028319	0.070175	0.05848	0.070175	0.123167	0.093367	0.108504	0.102339	0.428153	0.190508	0.108187	0.108187	0.111111	0.134503
P2	0.085044	0.102339	0.102639	0.061404	0.398527	0.192982	0.048708	0.102339	0.087719	0.078947	0.111437	0.105263	0.079179	0.093567	0.410557	0.350577	0.087719	0.108187	0.116559	0.114335
P3	0.093942	0.076233	0.064516	0.076233	0.365569	0.122957	0.055556	0.084795	0.105263	0.054327	0.122235	0.119883	0.096774	0.073369	0.422267	0.266262	0.266262	0.266262	0.134503	0.131579
P4	0.096774	0.096491	0.134507	0.054755	0.407626	0.125731	0.070175	0.076233	0.119883	0.078947	0.111332	0.093367	0.087717	0.116559	0.392982	0.266262	0.132983	0.084491	0.108187	0.128555
P5	0.140762	0.099415	0.093842	0.05848	0.398527	0.321637	0.046784	0.052632	0.111111	0.078923	0.1261	0.084795	0.073314	0.111111	0.445748	0.230394	0.102339	0.108187	0.143275	0.119883
P6	0.099707	0.087719	0.099707	0.05848	0.41349	0.154971	0.067251	0.087719	0.073369	0.078947	0.102639	0.093367	0.067449	0.067251	0.428153	0.24269	0.076233	0.122607	0.119883	0.119883
P7	0.102639	0.087719	0.093942	0.084795	0.381267	0.151579	0.078947	0.084795	0.087719	0.083567	0.123167	0.111111	0.093942	0.081671	0.404982	0.22607	0.266262	0.114335	0.119883	0.128555
P8	0.082111	0.093943	0.076246	0.07251	0.410557	0.149123	0.05848	0.084795	0.078947	0.054327	0.108504	0.099415	0.085344	0.093567	0.422267	0.217154	0.087719	0.111111	0.152047	0.116559
P9	0.064516	0.087719	0.076246	0.084795	0.352952	0.263158	0.061404	0.076233	0.087719	0.081671	0.111437	0.105263	0.079179	0.102339	0.41349	0.526316	0.111111	0.105263	0.116559	0.108187
P10	0.070381	0.092643	0.078179	0.070175	0.401176	0.215526	0.05848	0.108187	0.092643	0.076233	0.099707	0.096491	0.082111	0.076233	0.41349	0.349029	0.093943	0.094415	0.114335	0.125731
	0.091202	0.086102	0.079177	0.096491	0.397037	0.190358	0.05731	0.094211	0.092643	0.076318	0.114337	0.102262	0.095337	0.091813	0.418162	0.265398	0.095329	0.111894	0.132352	0.121345
	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10
P1	0.035088	0.046784	0.040936	0.02924	0.02924	0.354094	0.023352	0.035088	0.05848	0.017544	0.055846	0.070175	0.087719	0.02924	0.093567	0.356725	0.087719	0.070175	0.093567	0.099415
P2	0.052632	0.05848	0.054327	0.02924	0.046784	0.338181	0.046784	0.046784	0.064327	0.02924	0.046784	0.087719	0.076233	0.02924	0.087719	0.374269	0.064327	0.064327	0.105263	0.081871
P3	0.02924	0.040936	0.05848	0.046784	0.05848	0.333333	0.02924	0.026232	0.046784	0.02924	0.052632	0.111111	0.081671	0.035088	0.105263	0.350577	0.099415	0.099415	0.076233	0.076233
P4	0.046784	0.040936	0.093567	0.023352	0.070175	0.338181	0.052632	0.02924	0.046784	0.054327	0.052632	0.076233	0.070175	0.087719	0.094415	0.35117	0.076233	0.046784	0.076233	0.087719
P5	0.105263	0.040936	0.040936	0.02924	0.023352	0.327485	0.040936	0.02924	0.046784	0.052632	0.046784	0.070175	0.052632	0.064327	0.093567	0.356725	0.087719	0.076233	0.070175	0.081871
P6	0.035088	0.052632	0.054327	0.035088	0.052632	0.345029	0.076233	0.040936	0.064327	0.052632	0.076233	0.081871	0.052632	0.099415	0.084327	0.356725	0.099415	0.070175	0.070175	0.064327
P7	0.025332	0.040936	0.046784	0.05848	0.05848	0.280702	0.040936	0.046784	0.040936	0.035088	0.070175	0.081871	0.070175	0.046784	0.040936	0.350111	0.070175	0.035088	0.081871	0.070175
P8	0.02924	0.035088	0.046784	0.02924	0.046784	0.321637	0.064327	0.02924	0.02924	0.035088	0.05848	0.076233	0.070175	0.040936	0.070175	0.356725	0.070175	0.05848	0.111111	0.070175
P9	0.035088	0.052632	0.040936	0.05848	0.046784	0.321637	0.040936	0.035088	0.064327	0.087719	0.040936	0.070175	0.076233	0.046784	0.076233	0.356725	0.128555	0.052632	0.081871	0.064327
P10	0.023352	0.040936	0.040936	0.046784	0.076233	0.352952	0.040936	0.023352	0.017544	0.046784	0.064327	0.070175	0.05848	0.052632	0.084327	0.350111	0.081871	0.064327	0.087719	0.070175
	0.044444	0.045029	0.033811	0.035088	0.052632	0.322222	0.046784	0.036263	0.017544	0.046784	0.05848	0.076233	0.065931	0.049123	0.076233	0.350111	0.065931	0.054327	0.035088	0.076233

Figura 29. Ejemplo de informe generado automáticamente con la herramienta de informes.

4.6.1. Herramienta web

La principal motivación de desarrollar una herramienta web es la de poder automatizar y simplificar un proceso que se realiza completamente manual. La herramienta que se realizó permite a los usuarios olvidarse de algunos aspectos como la gestión de logs, o el poder compartir datasets con otros usuarios y etc, porque es la aplicación web la que se encarga de gestionar cualquier parámetro de entrada y salida necesario. Además, la generación de informes de resultados es de gran apoyo para poder visualizar los resultados obtenidos.

Otro de los puntos que se ha tenido en cuenta es que la aplicación desarrollada puede gestionar varios HPC⁵, lo que hace que se disminuya el tiempo de gestión, lo que conlleva a una disminución del tiempo necesario para poder obtener los resultados.

⁵ Un HPC es un sistema para computación de altas prestaciones

El objetivo general de esta aplicación es permitir que la herramienta sea capaz de lanzar simulaciones de *machine learning* sobre servidores con sistemas de colas. Este desarrollo incluye una arquitectura web capaz de responder a las necesidades de respuesta de los usuarios. Este objetivo general se detalla en los siguientes objetivos particulares:

- Proveer un servicio de almacenamiento para los conjuntos de datos y las poblaciones del AG.
- Crear una cola virtual para almacenar las tareas que no cojan en las colas de los HPC.
- Generación de informes fácilmente comprensibles
- Interfaz web sencilla bajo un front end, independiente de la arquitectura del servidor.

Fases y servicios ofrecidos por la aplicación web

Estructura de la aplicación

El primer paso necesario para la creación de la aplicación web es la de establecer la estructura donde se ha empleado *yeoman* que es un generador de código que generó una estructura inicial para *Java 8* que usa *Spring Boot* y emplea una BD *MariaDB*

Para la parte de registro de usuarios se escogió una autenticación API basada en tokens de sesión sin estado. En concreto se empleó el sistema de autenticación basado en JWT.

Almacenamiento de datasets y poblaciones

Durante esta etapa se añadieron los servicios que permiten añadir, mostrar, listar, borrar y compartir con otros usuarios y datasets.

Creación de simulaciones

Las tareas que se hicieron en esta fase fueron las de mostrar, listar y crear simulaciones. Para la creación de la simulación cabe destacar que se permite que los usuarios indiquen todos los parámetros, tanto de la red como los necesarios por los algoritmos gliales. También se permite que el usuario seleccione todos los parámetros relacionados con el AG.

Acceso a los servidores mediante ssh

Esta tarea está relacionada con las funcionalidades que puede realizar el administrador de la aplicación web. Los servicios que se añadieron fueron los de dar de alta un servidor ssh, listar los servidores ssh, dar de alta las cuentas de los servidores, mostrar las cuentas.

También se añadieron servicios internos que usa la aplicación, como el de abrir y cerrar conexiones ssh y la ejecución de comandos en los servidores de cómputo.

La aplicación para las simulaciones

Esta tarea permite subir a la aplicación web versiones de la aplicación de simulación para poder emplear siempre las versiones más actualizadas que se dispongan, sin necesidad de configurar el servicio web.

Uso de las colas de los CPD

En esta fase se creó lo necesario para consultar los gestores de las aplicaciones como puede ser consultar el número de trabajos que tiene una cuenta en el cpd, consultar el estado de los trabajos y comprobar los cambios de estado de estos.

Creación de la cola virtual y envío de trabajos

Cuando se necesita hacer uso de aplicaciones de HPC es habitual usar un servidor con un sistema de colas. La razón por la que se usa el sistema de colas es para asegurarse de que se está usando unos recursos determinados del sistema de manera exclusiva. El sistema de colas es el que se encarga de poner en ejecución las tareas de manera ordenada. Es habitual, que en los CPD con un sistema de colas, los usuarios solo puedan poner un número limitado de tareas en la cola. Esto se hace así para evitar que un usuario colapse todo el sistema.

La idea de la aplicación web es poder ignorar la limitación del número de tareas que se pueden meter en la cola. Para eso, se crea una cola virtual que almacenará los procesos que se quieran ejecutar y esta cola virtual será la encargada de ir introduciendo las tareas en la cola del CPD para que más adelante se ejecuten en el CPD.

En ocasiones los administradores de CPD pueden otorgar algunos privilegios a mayores como puede ser el aumento de tareas que se pueden meter en cola pero, estos privilegios suelen ser durante un periodo de tiempo. En ocasiones, los

administradores pueden otorgar acceso a alguna cuenta más para poder hacer pruebas en ella.

La idea la idea es que la aplicación web pueda usar todas las cuentas que te puedan proporcionar los administradores de CPD y usarlas de manera eficiente para poder usar de la mejor manera los recursos de los que se disponen. Para ello la cola virtual deberá repartir los trabajos entre las distintas cuentas.

Una de las opciones más fáciles para implementar es enviar todas las tareas a una cuenta y una vez que esté llena la cola de esa cuenta pasar a la siguiente. Esta opción no es muy buena idea, porque algún CPD tienen como limitación de que el número de trabajos que puedes tener en ejecución es mucho más pequeño que los puedes tener en cola. Para evitar este problema, se podría ir repartiendo los procesos de uno en uno por las distintas cuentas. El problema de esta opción es que se pasa mucho tiempo abriendo sesiones hacia el CPD para enviar las tareas a la cola.

La opción escogida ha sido una mezcla de las dos opciones. La idea es repetir el proceso de escoger la cuenta que menos tareas tenga en cola y enviar en esa cuenta un pequeño grupo de trabajos. Con esto hacemos que todas las cuentas tengan un número similar de trabajos en su colas y se evita perder mucho tiempo al no abrir y cerrar tantas conexiones contra el CPD.

En la figura 30 se puede ver un diagrama de la idea propuesta de la cola virtual implementada.

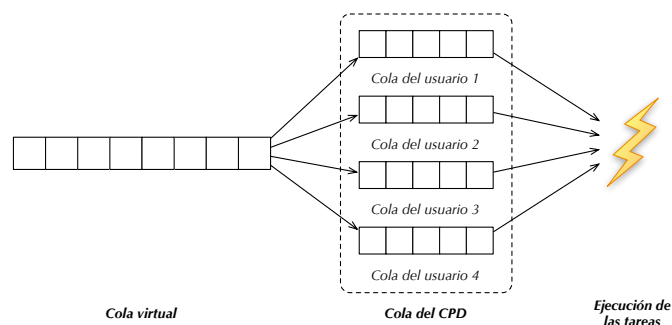


Figura 30. Idea en la que se basa la cola virtual

Recuperación de logs

En esta fase se incorporaron las funcionalidades necesarias para saber si una simulación ha finalizado su ejecución. También se agregó la función que permite recuperar el fichero de log y persistir su información en la aplicación web para después poder consultarse.

Generación de informes

Como se mencionó anteriormente, cada vez que se lanza una simulación se prueba con varias combinaciones de dataset y varias combinaciones de poblaciones. Si se dispone de una simulación con ocho combinaciones de dataset y diez de poblaciones se tendrá ochenta ejecuciones. Dicho de otra manera, se tendrá ochenta logs para el conjunto de entrenamiento y si el dataset tiene particiones para validación y test serían ochenta logs más para cada una de las particiones. Es decir, se podría tener 240 logs para analizar y a mayores hay que multiplicar por las generaciones del AG para tener los registros necesarios, lo que supone que si se lanza el AG con 7000 generaciones se podría llegar a tener hasta 1680000 registros para mostrar al usuario. Mover tal cantidad de datos hasta el usuario final puede suponer un problema de rendimiento o darle al usuario la sensación de que la aplicación no funciona como debería.

Por la razón mencionada anteriormente, se ha decidido simplificar los datos que se envían al usuario. No se envían 1680000 datos para dibujar las gráficas, que además supondría dibujar 80 series. Se ha optado por mostrar como mucho 4 gráficas, una general donde se muestra la evolución global de los 80 logs de la partición de entrenamiento, de la de validación y la de test. A mayores para cada partición se muestra la evolución global de los 80 logs de error y la desviación media. Con lo propuesto anteriormente evitamos enviar las 80 series y enviamos como mucho 6 series, una para el error global de cada partición y otra para la desviación media de cada partición. Aún así, enviando solo esas 6 series, si el AG generó 7000 registros se necesitaría enviar 42000 datos. Para simplificar esta situación, se optó por enviar una pequeña cantidad de puntos es decir dividir los 7000 registros en tan sólo una pequeña cantidad. Se ha optado que sean 40 puntos equidistantes. Con esta propuesta estamos simplificando los 1680000 registros en tan sólo 240 datos, lo que en este caso supone enviar 7000x menos datos.

En la figura 31 se muestra las gráficas de evolución de error que muestra la aplicación.



Figura 31. Ejemplo de las gráficas de evolución del error de la aplicación web

Arquitectura del sistema

La arquitectura de la aplicación está desarrollada bajo la base de datos relacional *MariaDB*, dando lugar a una fácil escalabilidad gracias a las opciones que tiene para la replicación. A pesar de esto, para las búsquedas relacionadas con texto no se emplea directamente la BD si no que se emplea el motor de búsqueda *Elasticsearch*.

La aplicación provee un servicio web bajo una arquitectura *RESTful* en el lado del servidor, o *back end*, creada con *Spring Boot*. Mediante el componente de *Spring Security* y un sistema de autenticación basada en *token* de sesión sin estado asegura el acceso a la API. La aplicación de del lado cliente, o *front end*, realiza las peticiones *HTTP* mediante el uso de *AngularJS*. Los usuarios pueden realizar peticiones directamente hacía el servicio web *RESTful* o usando la interfaz web.

En la figura 32 se muestra la arquitectura global de la aplicación con todos sus componentes en un despliegue. En la parte inferior a la derecha se encuentran las BD para el almacenamiento y los motores de búsqueda *Elasticsearch*. A continuación, en la parte superior derecha, se muestran los componentes del *back end*. Se puede ver que la capa de servicios se comunica con los *CPD* que tiene las colas de trabajo para ejecutar las simulaciones. Por último la parte del *front end* está escrita con *AngularJS* para mostrarle a los usuarios una interfaz con la que usar la aplicación.

La arquitectura del sistema muestra en la figura 33

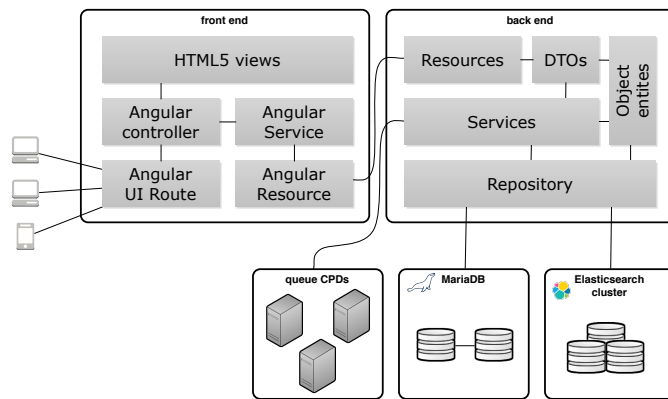


Figura 32. Arquitectura global de la aplicación web

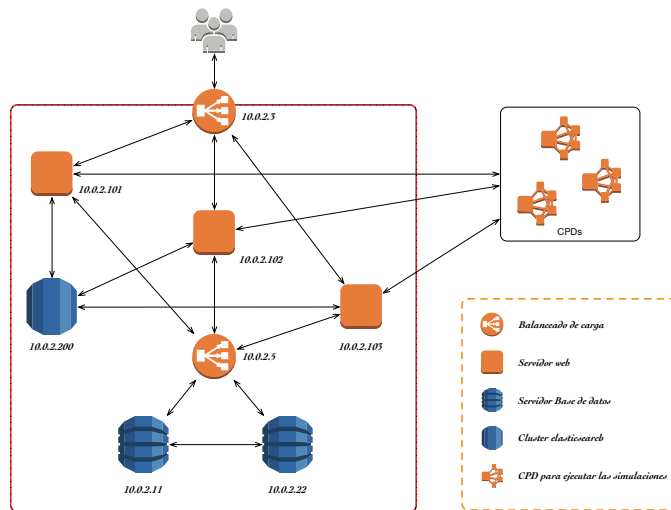


Figura 33. Arquitectura propuesta del sistema para la aplicación web

5. Experimentación y resultados

En este capítulo se expondrán y se analizarán todos los resultados obtenidos, utilizando la herramienta web opensource desarrollada y descrita en el capítulo anterior, durante la experimentación con las RNGA, así como los resultados de su comparación con las RNA, usando una, dos y tres capas ocultas.

5.1. Entorno de simulación

Antes de iniciar la exposición de los resultados conviene indicar diferentes aspectos relativos al entorno que se ha empleado, considerando que se experimenta con problemas de clasificación y de regresión:

Para cada problema de clasificación, los valores de comparación de cada RNGA con su correspondiente RNA serán los porcentajes de acierto en entrenamiento, validación y test. Para cada problema de regresión los valores de comparación serán el ECM en entrenamiento, validación y test. El valor *medio* tomado se refiere a la media aritmética obtenido en los resultados de las 100 ejecuciones, que fueron realizado con una red entrenada con 10 conjuntos distintos de patrones de entrada debido al *cross-validation* y empleando diez poblaciones iniciales de pesos.

La manera de seleccionar los individuos del AG que forman parte de los resultados se realizó simulando un proceso industrializado, donde no se selecciona el mejor individuo de validación. El proceso de selección reúne el top 5 de los individuos que obtuvieron el mejor error de validación (en el caso de que empate el error de validación se usa el error de entrenamiento para discriminar) y de esos cinco individuos se selecciona el que tenga mejor error de test.

En la resolución de cada uno de los problemas aplicando RNGA se han utilizado dos algoritmos genéticos distintos para usar dos problemáticas distintas:

1. Se establecieron los resultados de la glía a mano usando cuatro combinaciones neurogliales de activación-iteración distintas (2-4, 2-6, 3-6, 3-8). Para los algoritmos de potenciación se empleó un porcentaje de incremento del 25% y un porcentaje de decremento del 50%. Para los algoritmos de depresión y de endocannabinoides se empleó un 25% tanto para el porcentaje de incremento como para el porcentaje de decremento.
2. Se empleó el uso de un AGCC, donde es el propio AG el que busca los parámetros adecuados para la configuración de la glía. El AG busca para el porcentaje de incremento y decremento los valores con una resolución de 10% siendo el valor mínimo un 10% y el valor máximo un 90%. También se ha

establecido que el valor de la iteración tiene que ser mayor que el valor de activación teniendo la restricción que el valor de activación tiene que estar entre 2 y 8, mientras que el valor de iteración tiene que estar entre 3 y 9.

Para que las comparaciones sean válidas y las redes a comparar estén en igualdad de condiciones se ha preestablecido una configuración. La configuración que se ha empleado para el AG es la siguiente:

- El algoritmo de selección empleado ha sido el torneo determinístico con un tamaño de ventana 2.
- El algoritmo de cruce empleado fue SBX (*simulated binary crossover*) con un parámetro $\mu = 0,3$. Además, la tasa de cruce para cada generación ha sido del 90%.
- Se empleó una tasa de mutación del 5% en la generación de nuevos individuos donde la mutación provocaba un cambio en el gen de cómo mucho un $\pm 10\%$.
- El algoritmo de reemplazo hacía que los nuevos individuos sustituyeran a los peores de la población.
- En cuanto al número de generaciones para el caso de las RNA se emplearon un 30000, mientras que para las RGA donde se establecieron los parámetros de la glía a mano se usaron 5000 y las RGA creadas con un AGCC se usaban 7500 generaciones.
- En el uso del AGCC se seleccionaron dos individuos de la otra población para el cálculo de fitness donde uno se seleccionaba de manera aleatoria y el otro era el mejor de la población. El valor de bondad que se usaba finalmente era el valor mínimo de fitness.
- El tamaño de la población de pesos fue de 150 individuos mientras que el tamaño para la población de los parámetros de la glía fue de 30 individuos.
- La población de pesos tiene la restricción de que los pesos tienen que tener su valor entre -1 y 1.

Se emplean redes totalmente conectadas hacia delante que usan la función hiperbólica tangente como función de transferencia y la función de distribución es la identidad.

Para indicar el nivel de significación se emplearon asteriscos que indican el nivel obtenido al aplicar el test de Wilcoxon alcanzado por las RGA y las RNA. Así en función del p-valor obtenido se puede establecer la siguiente correspondencia:

p-valor < 0,05 → * las diferencias son significativas

p-valor < 0,01 → ** las diferencias son muy significativas

p-valor < 0,001 → *** las diferencias son extremadamente significativas

Las tablas de los resultados donde la configuración de la glía ha sido establecida manualmente, y que se pueden consultar en la sección “*apéndice a. resultados obtenidos en la experimentación*”, la columna mejor indica el resultado de añadir al *cross-validation* las combinaciones introducidas manualmente de activación-iteración, de tal manera que la columna mejor usa 400 ejecuciones (10 conjuntos de datos, 10 población iniciales y 4 combinaciones de activación iteración). Hay que indicar que, aunque esa columna, tenga 400 ejecuciones para la RNGA en lugar de las 100 de la RNA, la RNA usa muchas más generaciones en el AG.

Para los resultados obtenidos con el AGCC, se emplean cuatro configuraciones diferentes, ya que es necesario saber si influye usar el mismo algoritmo glial para toda la red o se puede aplicar uno distinto para cada capa. También se quiere comprobar si es necesaria la restricción de que el porcentaje de incremento no pueda superar el de decremento. En definitiva las cuatro configuraciones que se necesitan probar son:

- Usar el mismo algoritmo glial en toda la red y establecer la restricción de que el porcentaje de incremento glial no ha de ser superior al porcentaje de decremento.
- Usar el mismo algoritmo glial en toda la red y no establecer restricción en el porcentaje de la modificación glial.
- Permitir emplear un algoritmo glial en cada capa de la red y establecer la restricción de que el porcentaje de incremento glial no ha de ser superior al porcentaje de decremento.
- Permitir emplear un algoritmo glial en cada capa de la red y no establecer restricción en el porcentaje de la modificación glial.

5.1.1. Problemas de clasificación

Para la comparación de RNGA y RNA en problemas de clasificación se utilizaron tres problemas que se encuentran en UCI-MLR: cáncer de mama, señales de ionosfera y setas venenosas (Asuncion & Newman, 2007).

Setas venenosas

Este conjunto de datos tiene clasificadas setas en venenosas o no a través de 22 variables (Schlimmer, 1987). Para ello se disponen de 5644 casos donde el 38,2% corresponden a setas venenosas. Las características de este problema representan características físicas como la forma, color, superficie de las diferentes partes tales como el sombrero, el himenio, el tallo o el velo. A mayores, están incluidas características con información sobre el hábitat o la distribución de la población. Es importante mencionar que todos los datos son nominales, pero han sido normalizados con valores entre 0 y 1.

Cáncer de mama

El conjunto de datos sobre cáncer de mama consiste en poder catalogar los tumores en benignos o malignos en función de variables histológicas que habitualmente usan los especialistas para poder dar un diagnóstico (Tan & Eshelman, 1988). Los datos fueron tomados de 683 donde el 65,5% se corresponde a pacientes cuyo tumor es benigno y el 34,5% restante tienen el tumor maligno. Las nueve variables histológicas de las que consta el dataset son las siguientes:

- Espesor (*clump thickness*)
- Uniformidad del tamaño de la célula (*uniformity of cell size*)
- Uniformidad de la forma de la célula (*uniformity of cell shape*)
- Adhesión marginal (*marginal adhesion*)
- Tamaño individual de la célula (*single epithelial cell size*)
- Núcleo desnudo (*bare nuclei*)
- Cromatina laxa (*bland chromatin*)
- Nucléolos normales (*blanc chromatin*)
- Mitosis (*mitoses*)

Señales de ionosfera

El datos relacionado con la señal de ionosfera fueron obtenidos mediante un conjunto de 16 antenas de alta frecuencia con un poder de transmisión total de 6,4 kilowatios (Sigillito, Wing, Hutton, & Baker, 1989). El sistema de antenas se encuentra alojado en Labrador, Canadá, tiene como objetivo detectar los electrones libres en la ionosfera. La finalidad de este problema es poder clasificar la señal recibida en *buena* o *mala*. Si una señal muestra evidencia de algún tipo de estructura se considera como *buena* y si no se detecta la evidencia de algún tipo de estructura se considerará como una *mala* señal.

El conjunto de datos dispone de 351 señales, donde 64% han sido etiquetadas como buenas señales, que están representadas por un total de 34 variables.

5.1.2. Problemas de regresión

Para la comparación de RNGA y RNA en problemas de regresión se emplearon tres problemas que se encuentran en UCI-MLR: rendimiento estudiantil, incendios forestales y caída de cemento (Asuncion & Newman, 2007).

Asentamiento de hormigón

El objetivo de este problema es poder identificar asentamiento de hormigón según las medidas de su composición (Yeh, 1998). Se disponen de 107 datos, donde para cada uno de ellos tenemos la siguiente información:

- Cemento (*cement*)
- Escombros (*slag*)
- Cenizas volátiles (*fly ash*)
- Agua (*watter*)
- Superplastificante (*superplasticizer*)
- Roca gruesa (*coarse aggr.*)
- Roca fina (*coarse aggr.*)

Incendios forestales

El objetivo de este problema es el de predecir la superficie quemada en incendios forestales en la región noroeste de Portugal para el que se recogieron datos como la vegetación o información meteorológica (Cortez & Morais, 2007). Se dispone de unas 520 muestras en el conjunto de datos que disponen de 12 características.

La importancia de este problema es que los creadores de la base de datos no pueden determinar la cantidad de outliers que hay e indican que es un buen problema debido a los valores atípicos que tiene. Es importante mencionar, que la cantidad de ejemplos de incendios con una gran área quemada es muy pequeña.

Rendimiento estudiantil

El enfoque de estos datos es poder representar el rendimiento estudiantil en la educación secundaria de dos escuelas portuguesas (Cortez & Silva, 2008). Los datos contienen atributos relacionados con los estudiantes como pueden ser características demográficas, sociales e incluso se recogieron datos relacionados con la escuela. Todos

estos datos fueron obtenidos mediante informes y cuestionarios escolares. El atributo de destino es la calificación final que obtienen los estudiantes al final del año.

5.2. Experimentación y resultados en problemas de clasificación

En este apartado se analizan los resultados obtenidos para los tres problemas de clasificación probados, es decir, los problemas de: setas venenosas, cáncer de mama y señal de ionosfera. Para ello se probarán los resultados obtenidos con la RNA y con la RNGA para poder comparar si el uso de astrocitos artificiales es beneficioso. La RNGA se creará siguiendo dos aproximaciones, donde en la primera aproximación consiste establecer los algoritmos gliales y sus parámetros manualmente. Los algoritmos gliales a probar son de tres tipos, algoritmos de potenciación astrocítica donde los astrocitos potencian las conexiones, algoritmo de depresión astrocítica que debilitan las conexiones, mientras que el último tipo es el algoritmo de receptores de endocannabinoides donde los astrocitos tienen a la vez la actuación de potenciar y deprimir las conexiones. La segunda aproximación es establecer el algoritmo glial y sus parámetros de manera automática mediante el uso de un AGCC.

5.2.1. Análisis de astrocitos artificiales en redes con 1 capa oculta

Comparando los datos de los tres problemas usando una arquitectura de red con una capa oculta (apéndice “A.1.1. Análisis de astrocitos artificiales en redes con 1 capa oculta”) se puede ver que la configuración manual de los parámetros de la glía en la RNGA es mejor para el problema de ionosfera, que es el que tiene más características de entrada, que para los otros problemas. Esto sucede para los algoritmos de potenciación, depresión y receptores de endocannabinoides. Si nos fijamos en los resultados de la RNA, el problema más complejo es el de la ionosfera, que es justo el problema donde más ha destacado la RNGA usando la configuración manual de los parámetros de la glía.

Destacar que usando el AGCC, se observa que la RNGA obtiene mejores resultados que la RNA en el problema de cáncer de mama, con el que se obtuvo un p-valor de 0.65 lo que indica que la mejora introducida por la RNGA no es significativa.

La comparativa de los problemas de clasificación usando una capa oculta (apéndice “A.1.1. Análisis de astrocitos artificiales en redes con 1 capa oculta”) muestra que la RNGA empieza a ser efectiva para los problemas más complejos, y que donde hay muy buenos resultados con la RNA la RNGA no aportó mejoras. También, es importante mencionar que se pueden conseguir mejores resultados para las RNGAs usando un algoritmo glial diferente en cada capa de la red.

5.2.2. Análisis de astrocitos artificiales en redes con 2 capas ocultas

Analizando todos los resultados obtenidos con la información recogida para los datos de los tres problemas de clasificación y usando una arquitectura de red de dos capas ocultas (apéndice “A.1.2. Análisis de astrocitos artificiales en redes con 2 capas ocultas”) se puede observar que el algoritmo de efecto de atenuación astrocítica supera siempre el error de entrenamiento, algo que no sucede con los demás algoritmos.

Al fijarse en el problema que tiene el número de aciertos más alto para RNA, que es el de setas venenosas, se puede ver que la RNGA con el algoritmo de receptores de endocannabinoides supera a la RNA en entrenamiento, validación y test. Esto parece indicar que para problemas donde funcione muy bien la RNA es necesario que exista una gran actividad por parte del sistema glial, ya que este algoritmo es el único de los implementados en el que la acción de un astrocito se extiende a otras neuronas que no monitoriza.

En lo que respecta al problema de cáncer de mama, la RNGA tiene un mayor rendimiento que la RNA. De hecho, la RNGA con el algoritmo glial de depresión indica que obtiene unas diferencias extremadamente significativas al compararlo con la RNA ya que se obtiene un p-valor de 10^{-7} con el test de wilcoxon.

En el problema de señal de ionosfera, que es el que tiene el mayor número de características, la RNGA con la combinación de activación-iteración 2-4 de endocannabinoides es capaz de superar a la RNA en los aciertos de entrenamiento, validación y test, pero no es significativo ya que obtiene en test un p-valor de 0,93. Los demás algoritmos de la RNGA son capaces de superar los aciertos de la RNA. La RNGA con el algoritmo de potenciación activaciones no consecutivas y modificación con límite obtiene mejor desempeño que la RNA en entrenamiento. La RNGA con los algoritmos de potenciación de efecto de atenuación, los de actuaciones consecutivas, actuaciones no consecutivas con modificación sin límite y el algoritmo de depresión astrocítica logran una mayor cantidad de aciertos en las etapas de entrenamiento y validación si la comparamos con la RNA. Finalmente el algoritmo de potenciación de efecto de proceso global logra que la RNGA supere a la RNA en las etapas de entrenamiento y test logrando un p-valor de 0,1 lo que refleja que las diferencias obtenidas casi son significativas.

En lo que respecta al uso del AG para la búsqueda automática de los parámetros de la glía, se puede observar que en el problema de cáncer lo hace perfecto, ya que la RNGA en las cuatro aproximaciones empleadas supera a la RNA llegando a obtener un p-valor de 0,19. Para el problema de señal de ionosfera, la RNGA supera a la RNA en el

número de aciertos en entrenamiento y validación, pero la RNA consigue generalizar mejor que la RNGA. Para el problema de señal de setas venenosas, únicamente en la configuración del AG donde se usa la restricción de que el porcentaje de incremento no puede ser superior al porcentaje de decremento y donde cada capa tiene su propio algoritmo glial la RNGA supera los aciertos de RNA en entrenamiento y test, se puede decir que la RNGA supera a la RNA, ya que para los datos con los que se creaba el modelo la RNGA se adaptó mejor y también fue capaz de tener mejor generalización del modelo.

Al fijarse en las casuísticas probadas con el AGCC, se puede observar que, usando una única configuración del algoritmo glial en toda la red, los resultados mejoran en dos de los tres problemas si la restricción del porcentaje de incremento no supera al de decremento, pero sucede lo contrario si cada capa puede tener una configuración específica de los algoritmos gliales en cada capa en los tres problemas. Como es de esperar, se observa que la RNGA tiene mejores resultados si puede tener diferentes configuraciones de algoritmos gliales.

5.2.3. Análisis de astrocitos artificiales en redes con 3 capas ocultas

El uso de las RNGA usando una arquitectura de tres capas ocultas es beneficioso para el problema de cáncer de mama. Los resultados (apéndice "*Análisis de astrocitos artificiales en redes con 3 capas ocultas*") se ven mejorados de una manera increíble debido a las modificaciones astrocíticas generadas por la gran estimulación neuronal que se produce dentro de la red. Además, el beneficio se logra con los algoritmos de potenciación, de depresión y de receptores de endocannabinoides. Las diferencias entre la RNGA y la RNA son extremadamente significativas, por ejemplo se llega a obtener con el algoritmo de depresión un p-valor de 10^{-13} .

Los resultados obtenidos con el problema de señal de ionosfera reflejan que la RNA no es capaz de adaptarse tan bien a los datos que se le presentan ya que la RNGA supera a la RNA con los datos que se les pasan a las redes en la fase de entrenamiento. Usando un AGCC para la búsqueda del algoritmo glial y sus parámetros la RNGA logra mejores resultados en fase de entrenamiento y validación. En cuanto a la fase de test, se puede indicar que la RNA y la RNGA tienen el mismo desempeño ya que el p-valor obtenido, por ejemplo con el algoritmo glial de efecto de proceso global, es de 0,57 lo que indica que las diferencias no son significativas.

Sin embargo, el problema de setas venenosas es más simple y el nivel cognitivo necesario es más simple, lo que indica que los astrocitos no realizan sus funciones y

que se puede prescindir de ellos y usar simplemente la RNA ya que el modelo obtenido por la RNA obtiene muy buenos porcentajes de acierto y es extremadamente difícil superarlos.

Al fijarse en las casuísticas probadas con el AGCC, se puede observar que, usando una única configuración del algoritmo glial en toda la red, los resultados mejoran si la restricción del porcentaje de incremento no supera al de decremento, pero sucede lo contrario si cada capa puede tener una configuración específica del SN. Como es de esperar, se observa que la RNGA tiene mejores resultados si puede tener diferentes configuraciones de algoritmos gliales.

5.2.4. Comparativa global del uso de RNGA en problemas de clasificación

En este apartado se analiza el uso del algoritmo glial para los tres problemas de clasificación y se tendrá en cuenta usando las diferentes arquitecturas de red. Toda la información de los resultados expuestos se puede consultar en el apéndice “a.1. experimentación y resultados en problemas de clasificación”.

Algoritmo glial y sus parámetros establecidos manualmente

El problema de setas venenosas, usando diferentes arquitecturas de red, es un problema en el que se obtienen muy buenos resultados con la RNA, ya que consigue un porcentaje de aciertos de más del 93%. Como solo se emplea un 25% de los datos para entrenamiento se puede ver que no es un problema muy complicado ya que con un porcentaje pequeño de datos se consiguen buenos resultados en las fases de entrenamiento, validación y test. Que la RNA tenga un porcentaje alto de acierto deja un margen muy pequeño de mejora a la RNGA. Este hecho hace pensar que el uso de algoritmos gliales no es necesario en problemas sencillos, algo que biológicamente tiene sentido ya que los astrocitos están involucrados el proceso que tienen una gran actividad cognitiva.

El problema de cáncer de mama, es un problema con muy pocas características de entrada y donde el porcentaje de acierto en entrenamiento es muy bueno con RNA ya que está alrededor del 90% mientras que en las fases de validación y test el porcentaje de aciertos ronda el 85%. Los aciertos conseguidos con la RNA son buenos, pero al existir un salto de un 5% entre el porcentaje de acierto en entrenamiento, se puede decir que la RNA se especializa en esos datos de entrenamiento. Ese salto parece que ayuda a los algoritmos gliales a mejorar los resultados ya que las RNGA mejoran los resultados cuando hay más de una capa oculta. Fijándose en el algoritmo de depresión astrocítica para las combinaciones 3-8 las RNGA con dos y tres capas consiguen un porcentaje de acierto superior al 93,5%, en la fase de validación se consigue alrededor

de un 90% de acierto mientras que en la fase de test se consigue un porcentaje superior al 91%. Esto indica que la RNGA mejora todos los resultados que se consiguen con la RNA y, además, la RNGA no tiene ese salto de especialización que tiene la RNA con los datos de entrenamiento. Además, esta mejora se ve reflejada en el p-valor obtenido al comparar los resultados de la RNA y RNGA ya que se obtiene valores inferiores de 10^{-11} , tanto en entrenamiento, como validación y test, lo que indica que las diferencias son muy significativas.

Los datos obtenidos por la RNA con el problema de la señal de ionosfera tienen un porcentaje de acierto que está alrededor del 82%, 76% y 80% en entrenamiento, validación y test. Ese salto tan grande de un 6% entre el porcentaje de acierto en entrenamiento y validación puede sugerir que no hay suficientes datos en el conjunto de entrenamiento y es probable que al usar más de un 25% de los datos de entrenamiento se consiga mejores resultados, ya que los aciertos en son similares a los de entrenamiento (es importante recordar que la partición de datos para test es un 25% del total, el mismo porcentaje que la partición de entrenamiento). Esa poca cantidad de datos en entrenamiento también afecta a la RNGA ya que la diferencia del porcentaje de acierto es mucho mayor, por ejemplo, usando una RNA con la arquitectura de tres capas ocultas usando el algoritmo de potenciación de activaciones consecutivas se consiguen unos porcentajes del acierto del 86%, 81% y 83% para entrenamiento, validación y test respectivamente. Esta mejora por parte de la RNGA se ve también en el test paramétrico donde el p-valor para entrenamiento y validación es inferior a 10^{-7} mientras que en la fase de test es de 0,01 lo que indica que los cambios son muy significativos.

AGCC para la selección del algoritmo glial y sus parámetros

Aunque a primera vista el uso del AGCC se pueda usar para que busque el mejor algoritmo glial y los parámetros necesarios para este, se puede definir casos para mirar que configuración beneficia más a una RNGA. Con el AGCC se quisieron analizar dos aspectos importantes, el primer aspecto de estudio es si una red debe usar el mismo algoritmo glial en toda la red o si se puede discriminar por ejemplo por capas. El segundo supuesto a analizar es comprobar si lo que postulación hecha por Ana Porto donde sugiere que el porcentaje de incremento de modificación astrocítica no debe de ser superior al porcentaje de decremento (Porto, 2004).

Analizando los resultados obtenidos en los problemas de clasificación (apéndice “a.1 . experimentación y resultados en problemas de clasificación”), se observa que la mayoría de los casos funciona mejor una RNGA que tiene una configuración específica para cada capa, solamente en dos problemas con la arquitectura de una

capa oculta y en un problema con arquitectura de dos capas ocultas se obtiene algún porcentaje mejor usando el mismo algoritmo glial en toda la red.

En lo referente a la restricción expuesta sobre el porcentaje de incremento y decremento que el astrocito modifica sobre las conexiones en 2 de los 9 casos se obtienen mejores resultados haciendo que el porcentaje de decremento sea mayor o igual que el de incremento. Si se analizan las RNGA donde esos dos casos en los que se obtuvo mejores resultados al no tener la restricción se puede observar que la gran mayoría de las 100 ejecuciones que requiere una simulación (10 particiones de datos diferentes y 10 poblaciones distintas) el AGCC situó los valores de incremento y decremento siguiendo esas restricciones. Sabiendo que el AGCC establece esa restricción de manera interna se puede indicar que para los problemas de clasificación la hipótesis formulada por Ana Porto se cumple y la restricción de que el porcentaje de incremento no puede superar al porcentaje de decremento. Si se analiza el uso de la restricción en la casuística donde al AGCC puede establecer una configuración del algoritmo glial específica en cada capa se observa que en 5 de los 9 casos se obtiene mejores resultados si el AGCC no tiene la restricción de los porcentajes; y si se analizan las redes resultantes se puede observar que en algunas capas efectivamente el porcentaje de incremento astrocítico es superior. Esto lleva a plantear que la hipótesis de Ana Porto sobre la restricción de los porcentajes se cumple en el caso de que se empleó el mismo algoritmo glial en toda la red.

En lo referente a los resultados obtenidos usando el AGCC, se observa que funciona mejor cuantas más capas ocultas tenga la red. También se puede ver que, en el problema de setas venenosas, donde la RNGA le cuesta superar los buenos resultados obtenidos por la RNA, al usar el AGCC para la creación de la RNGA se observa que en la configuración de dos capas se logra superar los resultados de la RNA.

Otro hecho curioso, es que en los problemas en los que se ve que se usan pocos datos para entrenamiento y se genera un *overfitting* el AGCC hace que el porcentaje de validación sea un poco superior en las redes con el sistema glial incorporado.

5.3. Experimentación y resultados en problemas de regresión

En este apartado se analizan los resultados obtenidos para los tres problemas de regresión probados, es decir, los problemas de: asentamiento de hormigón, fuegos forestales, y rendimiento estudiantil. Para ello se compararon los resultados obtenidos con la RNA y con la RNGA para poder analizar si el uso de astrocitos artificiales es beneficioso. La RNGA se creará siguiendo dos aproximaciones, donde en la primera aproximación consiste establecer los algoritmos gliales y sus parámetros manualmente. Los algoritmos gliales a probar son de tres tipos, algoritmos de potenciación astrocítica donde los astrocitos potencian las conexiones, algoritmo de depresión astrocítica que debilitan las conexiones, mientras que el último tipo es el algoritmo de receptores de endocannabinoides donde los astrocitos tienen a la vez la actuación de potenciar y deprimir las conexiones. La segunda aproximación es establecer el algoritmo glial y sus parámetros de manera automática mediante el uso de un AGCC.

5.3.1. Análisis de astrocitos artificiales en redes con 1 capa oculta

Al comparar los resultados obtenidos en los tres problemas de regresión donde se emplean redes con una única capa oculta (apéndice “A.2.1. Análisis de astrocitos artificiales en redes con 1 capa oculta”) se puede observar que al establecer los parámetros de la glía manualmente la RNGA obtiene mejores resultados que la RNA. Además se da la situación de que para cada problema es mejor un algoritmo glial concreto que otro, y en muchas ocasiones este algoritmo es diferente al que es mejor para otros problemas. Por ejemplo, los algoritmos de potenciación que cuentan las activaciones de manera consecutiva funcionan mejor que los de depresión en el problema de rendimiento estudiantil. En cambio la depresión astrocítica funciona mejor que la potenciación con activaciones consecutivas en el problema de fuegos forestales.

Con todos los problemas la RNGA obtiene mejores resultados que la RNA, con un p-valor inferior a 10^{-7} , lo que refleja que la mejora introducida por la RNGA es extremadamente significativa.

En definitiva, al fijarse en los resultados de la RNGA con la configuración del algoritmo glial establecida desde el inicio se puede observar que funciona mejor que la RNA, pero al usar el AGCC los resultados no son tan esperanzadores como podría parecer. El AGCC funciona correctamente en el problema de asentamiento de hormigón, pero el resultado en el problema de rendimiento estudiantil indica que la RNA es más precisa. Al analizar la evolución del AGCC se puede observar que esta situación se debe a que los parámetros de un algoritmo glial no tienen que ser bueno

en otro algoritmo glial, lo que provoca que del cruce de dos buenos individuos se tenga una mala descendencia si los nuevos individuos generados no emplean el mismo algoritmo glial que los progenitores. Esta situación hace que el AGCC converja hacia un único algoritmo glial que puede acabar en dos situaciones que se tendrán en cuenta para futuros trabajos de investigación:

- La población converge hacia un algoritmo glial con el que no se obtiene buenos resultados para el problema específico con el que se está entrenando
- Se obtiene un algoritmo glial adecuado para el problema, pero se han gastado muchas generaciones en la convergencia del AG y no tiene margen de actuación para situar los valores adecuados en los parámetros del algoritmo glial que escogió el AG.

5.3.2. Análisis de astrocitos artificiales en redes con 2 capas ocultas

Analizando los resultados que obtienen al usar una configuración del algoritmo glial preestablecida desde el inicio para los problemas de regresión usando redes con una arquitectura de dos capas ocultas (apéndice “A.2.2. Análisis de astrocitos artificiales en redes con 2 capas ocultas”) se observa que para los problemas de asentamiento de hormigón y fuegos forestales se obtienen mejores resultados que los obtenidos con la RNA y se ve reflejado en el test estadístico empleado donde el p-valor, en los conjuntos de test, son inferiores al 10^{-6} lo que indica que la mejora de generalización obtenida por la RNGA es extraordinariamente significativa. Para el otro problema, el de rendimiento estudiantil, se obtienen resultados muy buenos con la RNA (0.0167, 0.0196 y 0.0221) lo que obligaría a la RNGA a acertar los valores con gran precisión para poder superar los resultados. Esto permite concluir que el algoritmo glial no merece la pena usarlo en problemas simples y que es más adecuado en problemas más complejos.

Si se observan los resultados obtenidos con el AGCC (apéndice “A.2.2. Análisis de astrocitos artificiales en redes con 2 capas ocultas”) se puede observar que no concuerdan con los datos obtenidos estableciendo el algoritmo glial y sus parámetros de manera manual. El problema es el mismo que el observado usando una capa oculta en problemas de regresión.

5.3.3. Análisis de astrocitos artificiales en redes con 3 capas ocultas

Al comparar los resultados obtenidos en los tres problemas de regresión donde se emplean redes con tres capas ocultas (apéndice “A.2.3. Análisis de astrocitos artificiales en redes con 3 capas ocultas”) se puede observar que al establecer los

parámetros de la glía manualmente la RNGA obtiene mejores resultados que la RNA únicamente en el problema de fuegos forestales. La diferencia de este problema con los demás radica en que la RNA comete bastante error en la predicción de valores y la RNGA tiene margen de mejora, algo que no sucede con los otros dos problemas.

5.3.4. Comparativa global del uso de RNGA en problemas de regresión

En este apartado se analiza el uso del algoritmo glial para los tres problemas de regresión usando las diferentes arquitecturas de red.

Algoritmo glial y sus parámetros establecidos manualmente

Los datos para el problema de la superficie quemada por los incendios forestales es un buen problema para usar técnicas de machine learning ya que ni los creadores de la base de datos pueden determinar qué datos son atípicos e indican que es un problema muy complejo y con pocos datos (Cortez & Morais, 2007). El ECM que comete la RNA es mucho mayor que el error que comete con los otros tres problemas y la RNGA se comporta mucho mejor en este problema. El algoritmo de receptores de endocannabinoides es capaz de mejorar a la RNA en los resultados de entrenamiento usando las tres arquitecturas de la red y, a mayores, usando la arquitectura de tres capas la RNGA logra superar el error de test, con lo que en una arquitectura más compleja logra generalizar mejor que la RNA. En este mismo problema la RNGA con el algoritmo de depresión es mejor en todos los aspectos que la RNA usando las arquitecturas de una y dos capas ocultas, pero sorprendentemente con tres capas ocultas la RNGA tiene un error de entrenamiento y validación un poco inferior, pero pierde mucha capacidad de generalización si comparamos el error de test. En lo referente a los algoritmos de potenciación, la RNGA con dos y tres capas ocultas es capaz de mejorar a la RNA en los tres tipos de error.

Analizando el problema que indica cuánto se asienta el hormigón se puede ver que la RNA comete unos errores de aproximadamente 0.055, 0.077 y 0.080 en entrenamiento, validación y test respectivamente para las arquitecturas de una y dos capas ocultas, pero al usar tres capas ocultas reduce los errores a 0.032, 0.062 y 0.062. La RNA consigue muy buenos resultados y hace que la RNGA tenga que dar valores extremadamente exactos para superarla. Las RNGA con los algoritmos de potenciación es capaz de mejorar a la RNA en los tres tipos de error con arquitecturas de una y dos capas, pero usando tres capas la RNGA solo mejora a la RNA con el error de entrenamiento. Las RNGA que usan el algoritmo de depresión astrocítica pierden su efecto con el incremento de capas ocultas, ya que con una capa oculta mejora los tres tipos de error, con dos capas ocultas mejora el error de entrenamiento y validación

mientras que con tres capas ocultas solo logra mejorar a la RNA en el error de entrenamiento.

El problema de rendimiento estudiantil se adapta muy bien a la RNA ya que esta consigue unos errores que están sobre el 0.017, 0.019 y 0.022 en entrenamiento, validación y test con las tres arquitecturas de red. Este problema también le pone muy difícil la capacidad de mejora que pueda lograr la RNGA y, de hecho, las RNGA con dos y tres capas ocultas no logra superar en ningún momento a la RNA. Sin embargo, los algoritmos de potenciación que usan las activaciones consecutivas para activar la actividad del astrocito hacen que la RNGA con una capa oculta reduzca los tres tipos de errores obteniendo mejoras muy significativas, al compararlas con la RNA, ya que el p-valor obtenido en el conjunto de test es de 0,039.

AGCC para la selección del algoritmo glial y sus parámetros

Analizando los resultados que consigue en los problemas de regresión, en especial los de rendimiento estudiantil y erupciones solares, la RNGA usando el AGCC para la búsqueda de los parámetros de la glía con la RNGA que tiene esos parámetros preestablecidos de manera fija desde un inicio se observa que sucede algo que se puede considerar anómalo. El AGCC no es capaz de conseguir buenos resultados ni de mejorar la configuración que se estableció a mano en simulaciones anteriores. Lo primero que se puede pensar es que al usar una resolución de 10% en los porcentajes de incremento y decremento, no es posible que el AGCC alcance el valor de 25% que se emplea en la configuración manual. Pero analizando los datos, por ejemplo con los algoritmos de potenciación, en algunos problemas un algoritmo glial específico tiene malos resultados, lo que indica que unos buenos parámetros de la glía para un algoritmo glial no tienen que funcionar bien en otro algoritmo glial y esta situación provoca que en el AG el cruce de dos buenos individuos genere una mala descendencia algo que se puede observar al analizar los individuos que se obtienen en la evolución del AGCC donde las primeras iteraciones los individuos convergen a tener un único algoritmo glial. Después de que el AGCC “seleccione” el tipo de algoritmo glial a usar se pueden dar dos situaciones, por un lado, si el algoritmo glial no es el adecuado para ese problema no se obtendrán buenos resultados y por otro lado si el algoritmo glial es el adecuado el AG no es capaz de encontrar los parámetros adecuados en las generaciones que le quedan por realizar.

Aun así, con este problema, el AGCC logra que la RNGA con una capa oculta mejore los errores de entrenamiento y validación que consigue la RNA en los problemas de fuegos forestales, erupciones solares y asentamiento de hormigón donde la RNGA en este último también mejora el error de test que consigue la RNA.

Con redes de dos capas ocultas el AGCC consigue que la RNGA mejore a la RNA en algunos casos. En el problema de asentamiento de hormigón hay casuísticas en la que la RNGA mejora el error de entrenamiento, el de entrenamiento y validación y también el de entrenamiento y test, pero en ninguno de los casos se mejora los tres errores en un único caso. Con el problema de fuegos forestales, en todos los casos la RNGA logra superar los errores de entrenamiento y validación de la RNA, pero no supera a esta en el error de test. Por último, la RNA consigue tener mejores resultados en los tres tipos de error para el problema de rendimiento estudiantil.

La última arquitectura de red, es decir, la que consta de tres capas ocultas el AGCC consigue que la RNGA tenga mejor error de entrenamiento en el problema de asentamiento de hormigón. Analizando el problema de los incendios forestales el AGCC logra que la RNA siempre tenga mejores errores de entrenamiento y validación. En el caso concreto donde el AGCC establece el mismo algoritmo glial para toda la red y no hay una restricción en los porcentajes la RNGA también logra cometer un error menor en la fase de test. Por último, el AGCC no logra que la RNGA consiga mejores resultados que la RNA en el problema de rendimiento estudiantil.

5.4. Discusión

En la figura 34 se muestra los resultados obtenidos para un problema de clasificación, el de cáncer de mama, y en la figura 35 se muestran los datos obtenidos para el problema de asentamiento de hormigón que es un problema de regresión. Ambas figuras muestran que la RNGA es capaz de lograr un gran desempeño en todas las fases por las que pasa la creación de un modelo, y que además también logra superar significativamente a los datos obtenidos por su red análoga menos realista, la RNA. Las figuras muestran que los tres tipos de algoritmos gliales, los de potenciación, depresión y receptores de endocannabinoides, no obtienen los mismos resultados. En algunos problemas funciona mejor uno y en otros problemas puede funcionar mejor otro tipo de algoritmo glial. Un ejemplo de ello es el algoritmo de receptores de endocannabinoides, los cuales no obtienen buen desempeño en problemas de regresión ya que solamente en una de las nueve casuísticas probadas con este tipo de algoritmo se consiguió obtener un modelo que generalizase mejor que la RNA.

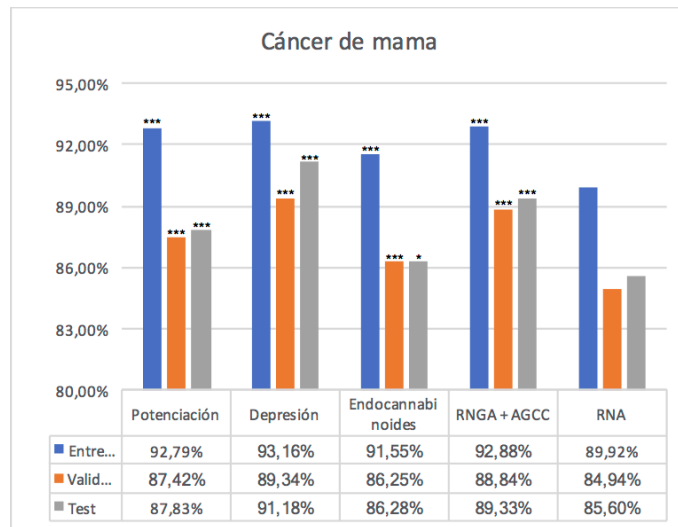


Figura 34. Resultados obtenidos para el problema de cáncer de mama con redes de tres capas ocultas. La combinación activación-iteración empleada para los algoritmos de potenciación, depresión y endocannabinoides es 3-6. El AGCC establece en cada capa una configuración distinta en cada capa de la RNGA. Los valores mostrados corresponden al porcentaje de acierto.

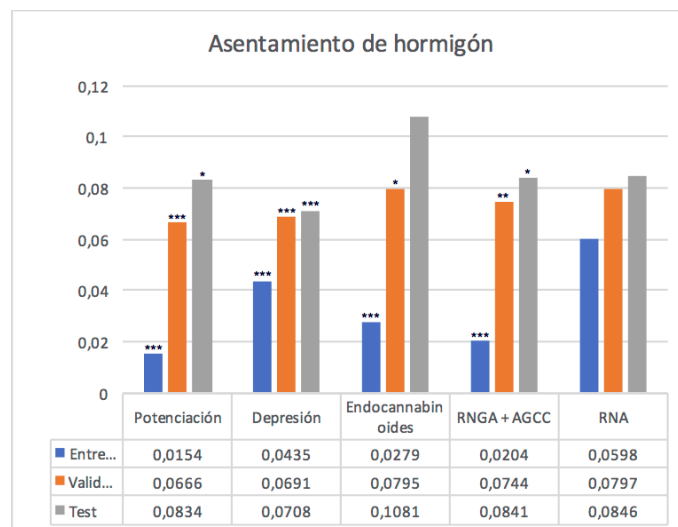


Figura 35. Resultados obtenidos para el problema de asentamiento de hormigón con redes de una capa oculta. La combinación activación-iteración empleada para los algoritmos de potenciación, depresión y endocannabinoides es 3-6. El AGCC establece en la RNGA la misma configuración glial y restringe a que el porcentaje de incremento no supere al de decremento. Los valores mostrados corresponden al ECM.

Los algoritmos de potenciación astrocítica que generalizan mejor en problemas de clasificación son los algoritmos de efecto de atenuación astrocítica y efecto de proceso global, ya que cada uno fue capaz de obtener en cuatro de las 12 casuísticas los mejores resultados en la etapa de test. Aunque empataron ambos algoritmos en la cantidad de veces que fueron los mejores dentro de su tipo, el de efecto de proceso global no era capaz de superar a la RNA en las tres etapas (entrenamiento, validación y

test), algo que el algoritmo de atenuación si fue capaz de lograr. Sin embargo, en regresión el algoritmo glial de potenciación que consigue los mejores resultados es el que hace que el astrocito se excite si las activaciones son consecutivas, y el siguiente a éste es el de efecto de proceso global. Hay que mencionar que el algoritmo de efecto de proceso global se comporta igual que en clasificación, en ningún caso logra superar a la RNA en las etapas de entrenamiento, validación y test, algo que si que pueden hacer los otros algoritmos.

Al observar los resultados establecer el algoritmo glial que mejor funciona depende no solo del tipo de problema, si es de clasificación o regresión, si no que depende del problema específico.

- En problemas de clasificación las redes con dos y tres capas, los algoritmos que mejor funcionan en el problema de señal de ionosfera es el de potenciación, sin embargo, en cáncer el mejor algoritmo glial es el de depresión astrocítica. En cuanto al problema de setas venenosas, con dos capas el mejor algoritmo es de potenciación mientras que con tres capas ocultas el que mejor funciona es el de receptores de endocannabinoides.
- En problemas de regresión, los algoritmos de depresión astrocítica consiguieron mejores resultados en redes de una capa para los problemas de asentamiento de hormigón e incendios forestales, mientras que en el otro caso las mejores RNGA han sido las que han empleado algoritmos de potenciación.

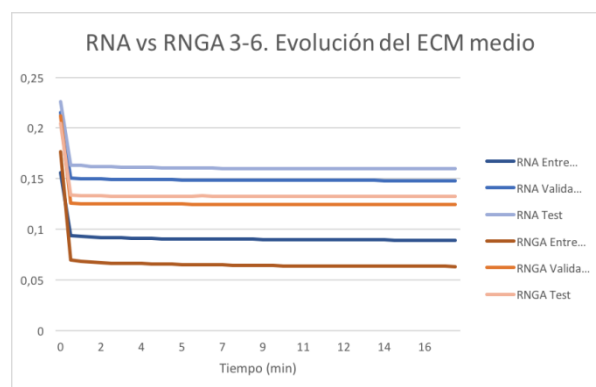


Figura 36. Evolución del error para el problema de cáncer de mama con redes de tres capas ocultas donde la RNGA emplea el algoritmo de efecto de atenuación astrocítica con una combinación activación-iteración 3-6.

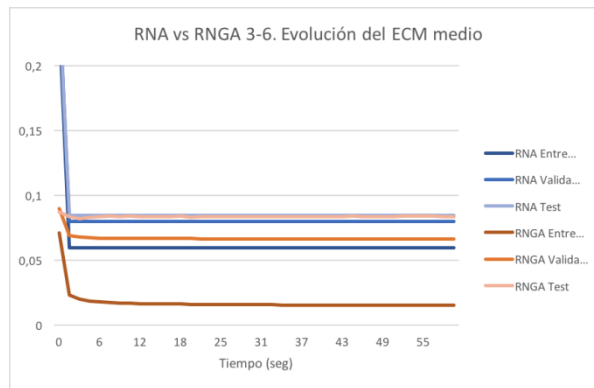


Figura 37. Evolución del error para el problema de asentamiento de hormigón con redes de una capa oculta donde la RGA emplea el algoritmo de efecto de atenuación astrocítica con una combinación activación-iteración 3-6.

La figura 36 muestra como evoluciona el error en un problema de clasificación, en concreto para cáncer de mama, mientras que la figura 37 muestra la evolución para el problema de regresión asentamiento de hormigón. En ambas gráficas se muestra que la capacidad de la RGA es asombrosa, ya que en el caso de las figuras, la RGA requiere como tiempo de ejecución de 6 veces lo que le lleva la RNA (la RGA de las imágenes tienen como parámetro de iteración el valor 6). Esto indica que la capacidad de adaptación de la RGA a los problemas, tanto de clasificación como de regresión, es asombrosa.

Otro aspecto importante que se observa en las RGA durante los problemas de clasificación es que tiende a funcionar mejor cuanto más grande y compleja sea la red. Además, con el problema de setas venenosas, los resultados llevan a indicar que en problemas simples donde la RNA obtiene buenos resultados los algoritmos gliales no mejoran los resultados. En cuanto a los problemas de regresión, se puede observar que la RGA tiene un plus de mejora en los entrenamientos de la red, logrando que en problemas donde la RNA tiene unos resultados increíbles los algoritmos gliales pueden mejorar incluso esos resultados. En cuanto a la complejidad de la red, en los problemas de regresión sucede algo diferente a lo que se observa en clasificación, las redes con algoritmos gliales no funcionan mejor que las RNA en las redes más grandes.

En lo relacionado a la búsqueda de los parámetros gliales de la red usando AGCC no se consiguen las mejoras que se podrían esperar ya que por la definición de AGCC, la búsqueda automática del de las conexiones de la red y los parámetros gliales, es un problema de búsqueda que se adapta al AGCC. En clasificación existen mejoras si se compara con la RNA, pero al comparar los resultados obtenidos empleando un AGCC y usando unos parámetros gliales preestablecidos las mejoras que obtiene el AGCC son muy pequeñas. Lo que sorprende es que en regresión, la mayoría de los modelos que se obtienen no superan la capacidad de generalización de la RNA. Al analizar los

registros que se obtienen empleando AGCC, se puede ver que el AGCC durante las primeras generaciones está confuso y trata de encontrar que algoritmo glial debe seleccionar y ,después de seleccionar uno, busca los parámetros adecuados para este. Esta confusión la obtiene porque los parámetros gliales que son buenos para un algoritmo puede que no sean buenos para otro, lo que provoca que al cruzar dos buenos individuos, salga como resultado una descendencia con malos valores de bondad. En clasificación este hecho no se nota tanto porque más o menos se tienen desempeños similares con los diferentes algoritmos gliales, pero en regresión la elección de estos parámetros es mucho más crítica, lo que hace que el AG pueda seleccionar un algoritmo glial que no sea adecuado y no se ajuste bien al problema.

A pesar de ese problema detectado que tiene el AGCC con el cruce de los individuos, se pueden comparar los resultados obtenidos en las cuatro casuísticas probadas, que son las siguientes:

- Uso de los mismos parámetros gliales en toda la red, y con la restricción de que el porcentaje de incremento que pueden modificar las conexiones a los astrocitos no sea superior al porcentaje de decremento.
- Uso de los mismos parámetros gliales en toda la red.
- Uso de diferentes parámetros gliales para cada capa de la red.
- Uso de diferentes parámetros gliales para cada capa de la red, y con la restricción de que el porcentaje de incremento que pueden modificar las conexiones a los astrocitos no sea superior al porcentaje de decremento.

Al observar en los resultados si la restricción puesta en los porcentajes de incremento y decremento obtiene mejores resultados, se puede observar que, al usar la misma configuración glial en toda la red, en la gran mayoría de los casos establecer esa restricción genera mejores modelos de RNGA, pero lo importante son los pocos casos cuando el AGCC obtienen mejores resultados sin la restricción de los porcentajes de incremento y decremento ya que las RNGA resultantes de manera interna cumplen la restricción. Esto tiene una justificación biológica, ya que si el incremento por refuerzo es menos que el decremento cuando no hay actividad constante, solo permanecería reforzada las conexiones de neuronas que representan una actividad continuada. Pero si se observa los resultados obtenidos donde la configuración glial de cada red puede ser diferente para cada capa, lo que es una versión más realista, se observa que la mayoría de las veces se obtienen mejores resultados no poniendo la restricción; sin embargo, al fijarse en las RNGA resultantes se obtiene que en muchas ocasiones únicamente la capa entrante tiene el porcentaje de incremento superior al decremento. Teniendo en cuenta de que la capa de entrada en una RNGA se emplea

para introducir los datos a la red se puede indicar que también cumple la explicación biológica.

En lo relacionado a si es mejor emplear una configuración glial en toda la red o una distinta en cada capa, los resultados de entrenamiento muestran que es mejor tener una configuración glial diferente en cada capa de la RNGA. Al observar, los datos obtenidos en clasificación, la generalización del modelo se ve mejorada cuando se usa una configuración diferente en cada capa de la red. Sin embargo, el uso de una configuración en cada capa de la RNGA en los problemas de regresión hacen que el modelo sufra de overfitting siendo mejor usar la misma configuración glial en toda la red, pero esto puede ser debido al problema que tiene el AGCC al cruzar los individuos que contienen la configuración de la glía que se ve más acusado cuando tiene varios algoritmos para seleccionar.

6. Conclusiones

Los algoritmos desarrollados están inspirados en los recientes descubrimientos sobre el papel de células gliales denominadas astrocitos en el procesamiento de la información. En concreto, el concepto de la sinapsis tripartita fue el que asentó las bases de esta tesis. Los algoritmos desarrollados representan una simulación adaptada de astrocitos artificiales para interactuar con neuronas artificiales y funcionar en los sistemas conexionistas.

En la primera fase del estudio de los algoritmos gliales, se han desarrollado diferentes comportamientos biológicos de los astrocitos sobre cómo modifican las conexiones de las neuronas. Los algoritmos desarrollados se han probado por primera vez sobre dos tipos diferentes de problemas, dando lugar a una experimentación en 3 problemas de clasificación y 3 problemas de regresión. El análisis de los resultados de este estudio ha permitido extraer las siguientes conclusiones:

- Se ha demostrado que la eficiencia de las redes de neuronas artificiales, en arquitecturas multicapa conectadas totalmente hacia delante y entrenadas con algoritmos genéticos, aumenta al introducir en ellas los astrocitos artificiales, lo que refuerza los descubrimientos que indican que existe un rol crucial de los astrocitos en el procesamiento de la información en el cerebro mediante la sinapsis tripartita donde existe una comunicación bidireccional entre las neuronas y los astrocitos.
- Se ha podido comprobar que dependiendo del tipo del problema, es decir, si es de clasificación o de regresión, algunos algoritmos gliales funcionan mejor que otros. El algoritmo de efecto de atenuación astrocítico, es de los que mejor funciona en problemas de clasificación, mientras que en regresión hay otros algoritmos, como los que excitan una respuesta por parte de los astrocitos únicamente si las activaciones neuronales se producen de manera consecutiva, que tienen mejor desempeño. Esto se ha observado en el cerebro donde en los procesos cognitivos de aprendizaje y de memoria, unos caminos sinápticos se ven afectados por distintos procesos moduladores y se desconoce bajo qué condiciones tienen lugar los posibles efectos moduladores.
- Además, se ha comprobado que dependiendo del tipo de salida del problema puede ocurrir que un algoritmo no funcione de una manera óptima, como es el caso del algoritmo de receptores de endocannabinoides en el caso de los problemas de regresión, es necesario ajustar el tipo de algoritmo glial a cada problema específico.

- En los problemas de clasificación se puede ver fácilmente que cuando más compleja y grande es la red, mejores resultados logran los algoritmos gliales. En el caso de regresión, debido a que la RNA consigue errores muy bajos, se ve que la tendencia es que en una y dos capas ocultas los algoritmos gliales también tienen buenos resultados. En regresión también se puede ver, aunque menos que en clasificación, que la complejidad de la red es un factor que beneficia a los algoritmos gliales, ya que los resultados obtenidos en regresión con RGA de dos capas ocultas son mejores que los de una capa oculta. Esta observación tiene una explicación biológica, ya que la actuación de los astrocitos tiene lugar en los procesos cognitivos más complejos.
- La evolución de los errores de entrenamiento muestra que las redes con los algoritmos astrocíticos tienen un aprendizaje más reforzado ya que, a pesar de que los procesos de iteración en las redes neurogliales artificiales las hace más lentas, aprenden en menor tiempo y cometiendo menor error. Esto reafirma que los astrocitos están involucrados en los procesos de aprendizaje y, aunque en algún problema de regresión parezca que empieza generar overfitting por la especialización en el entrenamiento realmente no hay problemas de sobreentrenamiento al analizar los resultados y la evolución de los errores de entrenamiento y validación. Esta cuestión lleva a pensar que, aunque aprenda muy rápido la red aún necesita tiempo para poder generalizar los conceptos específicos del problema, haciendo que, después, del “aprendizaje” los altos procesos cognitivos sean necesarios para generalizar los conceptos en los problemas realmente complejos.
- Se observa que al modificar los parámetros de la relación astrocito-neurona, los valores de activación-iteración, se puede lograr mejor rendimiento de la RGA que al establecer unos valores fijos. Esto indica que las propiedades de la interacción entre neurona y astrocito en la sinapsis tripartita son extremadamente relevantes y se corresponde a las observaciones vistas biológicamente entre neurona y astrocito. En los resultados se puede observar que los datos de la columna “mejor” se consiguen mejores resultados que la RNA en el 60% de los casos mostrados en las tablas. Sin embargo, al analizar los datos obtenidos el uso de una única combinación neuroglial, por ejemplo, la combinación de activación-iteración 3-6, no siempre obtiene datos superiores en todos los problemas. Esto muestra que, al igual que sucede biológicamente, los parámetros adecuados de activación-iteración dependen del problema analizado y de las condiciones de partida (conjuntos de datos y configuraciones iniciales de la red).

- Mientras que los modelos más complejos necesitan muchos datos en la fase de entrenamiento para empezar a aprender y no tener problemas de sobreentrenamiento, como es el ejemplo de las redes de Deep learning que son devoradores de datos de entrenamiento, los problemas que se han usado aquí no contienen una gran cantidad de datos. Además el uso de los algoritmos gliales que añaden más complejidad a la red les permiten aprender fácilmente ya que solo se les pasaba una cuarta parte de los datos disponibles en entrenamiento. Probablemente el uso de más datos durante la fase de aprendizaje haría que las redes pudiesen generalizar mucho mejor los conceptos y tendrían mejores resultados.

La siguiente fase de estudio de los algoritmos gliales, se basó en usar una técnica de IA que buscase a la vez los valores adecuados de las conexiones de la red y los parámetros gliales. Para ello se ha empleado la técnica de aprendizaje cooperativo del cual se ha extraído las siguientes conclusiones:

- El uso de un AGCC teóricamente se ajusta al problema que busca simultáneamente las conexiones de la red y la configuración glial de una red. Ya que se cumplen las cuatro condiciones necesarias, un espacio de búsqueda complejo, el problema se puede separar por definición, se usan diferentes tipos de valores y conceptos, existe una fuerte interconexión entre los componentes de la solución. Esto hace pensar que, al usar esta aproximación, los resultados de la RNGA tienen que ser mejores que los obtenidos con una red cuya configuración glial se ha establecido de antemano. Sin embargo, no siempre se obtienen mejores resultados. Al analizar la evolución de las redes se ha podido ver que buscar todos los parámetros gliales no es simple usando la estrategia de un algoritmo de estado estable en la población que contiene la configuración glial. Esto se debe porque al poner el algoritmo glial como un parámetro de búsqueda hace que los demás valores de búsqueda sean sensibles al algoritmo glial seleccionado, de tal manera que de dos buenos individuos que tienen algoritmos gliales diferentes tengan posibilidades de generar descendientes con malos ajustes, donde la única forma que tiene el algoritmo genético de salir es converger primero hacia un algoritmo glial, que no tiene por qué ser el adecuado, y después ajustar el resto de parámetros gliales. Por esta razón la aproximación coevolutiva no siempre obtiene mejor rendimiento que la preestablecida de antemano.
- **El uso de la computación coevolutiva reflejó que la modificación astrocítica que realiza en la red funciona mejor con una restricción y es que el porcentaje de incremento sobre las conexiones no puede ser**

superior al de decremento. De hecho, cuando el algoritmo genético no tiene puesta esa restricción en la codificación de sus individuos las soluciones resultantes tras varias iteraciones cumplen la condición de restricción, siendo el único caso donde no se cumple en los problemas de regresión con una configuración diferente por cada capa de la red donde en la primera capa el incremento que realizan los astrocitos sobre las conexiones son superiores al decremento que inducen. Que los astrocitos tengan la restricción de que el incremento sea inferior tiene una justificación biológica ya que si el incremento por refuerzo es menor que el decremento cuando no hay actividad constante solamente permanecerían reforzadas las conexiones neuronales que tienen actividad constante (Haydon & Araque, 2002).

- El uso de la computación coevolutiva también mostró que dentro de una red se pueden emplear diferentes combinaciones gliales. De hecho, reveló que para problemas de clasificación es excelente tener diferentes algoritmos gliales por capas y, que en regresión el uso de diferentes parámetros gliales para una red aprenda muy rápido en la fase de entrenamiento, lo que con un 25% de datos de entrenamiento son muy pocos datos para poder generalizar bien los modelos. El uso de diferentes comportamientos astrocitos está en consonancia con los observado biológicamente en el cerebro.

Las conclusiones obtenidas son coherentes y acordes con los descubrimientos biológicos expuestos en el apartado de antecedentes y sirven para ayudar a consolidar más el rol de las Redes NeuroGliales Artificiales en el campo de la IA.

Por último, mencionar la importancia de haber desarrollado una herramienta de código abierto que se pondrá a disposición de toda la comunidad. Ha permitido la experimentación así como analizar cómo se ejecutan las aplicaciones, ya que gracias a ello se obtuvo una medida de cuánto tiempo se emplea en la ejecución de las redes en la simulación. Esto llevo a centrarse en el funcionamiento interno de la red haciendo posible la creación de una nueva implementación en las RNA y RNgAs, donde el ahorro de memoria que se consigue es muy notable ya que solo se almacena lo estrictamente necesario, pasando a tener un modelo que se centra en las conexiones de las neuronas y no en el número de elementos. Además de este ahorro de memoria, y del orden que se sigue para calcular la salida de la red se ha conseguido incrementar la velocidad de la red de manera considerable con respecto a las implementaciones existentes hasta ahora para comparar RNgA

con RNA. Se ha observado que cuanto más compleja y grande es la red más rápido se ejecuta comparado con el método tradicional.

7. Conclusions

The algorithms developed are inspired by recent discoveries about the role of glial cells in information processing. In particular, the concept of the tripartite synapse was the one that laid the foundations of this thesis. However, it is important to mention that the developed algorithms are not coded so as to faithfully emulate the biological behaviour of the glial cells but that the algorithms represent a simulation adapted to work with the connectionist systems.

In the first phase of the study of glial algorithms, different biological behaviours that contain astrocytes have been developed on how they modify the connections of neurons. The developed algorithms have been tested on two different types of problems (3 problems for classification and 3 problems for regression). The analysis of the results of this study has allowed us to draw the following conclusions:

- It has been shown that the efficiency of artificial neuron networks, in multilayer architectures totally connected forward and trained with genetic algorithms, increases with the introduction of artificial astrocytes, which reinforces the findings that indicate a crucial role of the Astrocytes in the processing of information in the brain through the tripartite synapse where there is a bidirectional communication between neurons and astrocytes.
- It has been proven that depending on the type of problem, that is, whether it is classification or regression, some glial algorithms work better than others. The algorithm of astrocyte attenuation effect is the one that works best in classification problems, while in regression there are other algorithms, such as those that excite a response by astrocytes only if the neuronal activations occur consecutively, which have better performance. This has been observed in the brain where it is not known that in the cognitive processes of learning and memory synaptic pathways are affected by different modulatory processes, it is unknown under what conditions the possible modulatory effects take place.
- In addition, it has been found that depending on the type of output of the problem it can be indicated that an algorithm does not work in an optimal way, as is the case of the endocannabinoid receptor algorithm in the case of regression problems, it is necessary to adjust the type of glial algorithm to each specific problem.
- In classification problems, it can easily be seen that the more complex and larger the network, the better the glial algorithms achieve. In the case of regression, because the ANN gets very low errors, it is seen that the tendency that until the use of two hidden layers the glial algorithms also

have good performances. In other words, in regression, you can also see, although not as easily as in classification, that the complexity of the network is a factor that benefits the glial algorithms since the results obtained in regression with ANGN of two hidden layers are better than those of a hidden layer. This observation has a biological explanation since the action of the astrocytes takes place in the most complex cognitive processes.

- The evolution of training errors shows that networks with astrocytic algorithms have a more reinforced learning because, despite the fact that iteration processes in artificial neuronglia networks slow them down, they learn in less time and make fewer mistakes. This reaffirms that astrocytes are involved in learning processes and, although some regression problem seems to start generating overfitting by specialization in training there are really no overtraining problems when analyzing the results and the evolution of training errors and validation, which leads to think that, although it learns very fast the network still needs time to be able to generalize the specific concepts of the problem, making that, after "learning" the high cognitive processes are necessary to generalize the concepts in the problems really complex.
- It is observed that by modifying the parameters of the astrocyte-neuron relationship, the activation-iteration values, better performance of the ANGN can be achieved than by establishing fixed values. This indicates that the properties of the interaction between neuron and astrocyte in the tripartite synapse are extremely relevant and corresponds to the observations seen biologically between neuron and astrocyte. In the results, it can be seen that the data of the "best" column achieves better results than the RNA in the vast majority of the almost 200 cases shown in the tables. However, when analyzing the obtained data the use of a single neuroglial combination, for example, the combination of activation-iteration 3-6, does not always obtain superior data in all the analyzed problems and, it can also be seen that, in some cases, the best combination obtained in training is different from the best for validation and testing. This shows that, as happens biologically, the appropriate activation-iteration parameters depend on the problem analyzed and on the initial conditions (data sets and initial network configurations).
- While the more complex models need a lot of data in the training phase to start learning and not have overtraining problems, as is the case of the Deep Learning networks that are devouring training data, the problems that have been used here they do not contain a large amount of data, where in addition the use of glial algorithms that add more complexity to the

network can easily be learned since only a quarter of the data available in training was passed to them. Probably the use of more data during the learning phase would make the networks could generalize the concepts much better and would have better performance since during all the problems they have wanted to put the glial algorithms in trouble so that they do not generalize.

The next phase of the study of the glial algorithms was based on using an artificial intelligence technique that looked for the appropriate values of the network connections and the glial parameters at the same time. To this end, the cooperative learning technique has been used, from which the following conclusions have been drawn:

- The use of a coevolutionary cooperative genetic algorithm theoretically fits the problem that simultaneously searches for network connections and the glial configuration of a network since the four necessary conditions are met, a complex search space, the problem can be separated by definition, different types of values and concepts are used, there is a strong interconnection between the components of the solution. This suggests that, when using this approach, the results of the ANGN have to be better than those obtained with a network whose glial configuration has been put to hand. However, better results are not always obtained despite the use of cross-validation. When analyzing the evolution of the networks it has been possible to see that looking for all the glial parameters is not simply using the strategy of a stable state algorithm in the population that contains the glial configuration. This is because putting the glial algorithm as a search parameter makes the other search values sensitive to the selected glial algorithm, so that two good individuals with different glial algorithms have the possibility of generating descendants with poor settings, where the only way for the genetic algorithm to exit is to first converge towards a glial algorithm, which does not have to be the right one, and then adjust the rest of the glial parameters. For this reason, the coevolutionary approach does not always obtain better performance than when establishing the glial configuration by hand.
- The use of coevolutionary computation reflected that the astrocytic modification performed in the network works better with a restriction and that the percentage of increase over the connections can not be higher than the decrease. In fact, when the genetic algorithm does not have that restriction in the coding of its individuals the resulting solutions after several iterations fulfil the restriction condition, being the only case where

the regression problems with a different configuration for each layer are not met. of the network where in the first layer the increase that the astrocytes perform on the connections are greater than the decrease they induce. That the astrocytes have the restriction that the increase is lower has a biological justification because if the increment by reinforcement is smaller than the decrease when there is no constant activity only the neural connections that have constant activity would be reinforced(Haydon & Araque, 2002).

The use of coevolutionary computation also indicated that different glial combinations can be used within a network. In fact, it revealed that for classification problems it is excellent to have different glial algorithms by layers and, that in regression the use of different glial parameters for a network learn very fast in the training phase, which with 25% of training data There are very few data to be able to generalize well the models. The use of different astrocyte behaviours is in line with those observed biologically in the brain.

Finally, mention the importance of analyzing how applications are executed since thanks to this a measure of how much time is spent in the execution of the networks in the simulation was obtained. This led to focus on the internal functioning of the network making possible the creation of a new implementation in the networks of artificial neurons, where the saving of memory that is achieved is very remarkable since only the strictly necessary is stored, passing to have a model that focuses on the connections of the neurons and not on the number of elements. In addition to this saving of memory, and the order that is followed to calculate the output of the network, it has been possible to increase the speed of the network in a considerable way, where the more complex and larger the network the faster it is executed compared to the method traditional.

8. Futuros Desarrollos

Los resultados obtenidos con las RNGA afianzan las hipótesis biológicas en las que se ha basado el desarrollo de la presente tesis, a la vez que aportan más evidencia sobre la utilidad de la incorporación de astrocitos artificiales en los modelos de Inteligencia Artificial. El trabajo desarrollado propone que esta reciente línea de investigación tenga que ser tomada en cuenta a la hora de desarrollar modelos inteligentes. Algunos de los futuros desarrollos que se plantean como continuación de este trabajo son los siguientes:

- **Capacidad de modificación astrocítica.** Los estudios analizados por Ana Porto, reflejaban que después de realizar diversas pruebas, los parámetros que parecían adecuados para el porcentaje de modificación sobre las conexiones neuronales eran un 25% y 50% para incrementar y decrementar los valores respectivamente. Durante la computación coevolutiva se ha dejado que los porcentajes variasen entre el 10% y el 90% con una resolución del 10% y, curiosamente diferentes redes, tanto las que usaban una configuración glial en toda la red como las que podían tener una configuración distinta en cada capa, llegaron a tener porcentajes del 90%. Un estudio interesante, tanto biológicamente como computacionalmente podría ser el tratar de averiguar cuál es la modificación máxima que se podría obtener con la actuación de los astrocitos.
- **Búsqueda automática de configuración glial.** El uso de un algoritmo de estado estable en la población de parámetros gliales presenta problemas de convergencia hacia un algoritmo glial que no busca que sea óptimo. Una posible solución sería usar un algoritmo basado en islas donde cada una de las islas exista individuos pertenecientes a un algoritmo glial y que se busque los parámetros adecuados. Sería necesario analizar la carga computacional de esta aproximación, ya que la búsqueda simultánea de las conexiones de la red y la configuración glial es computacionalmente costoso, y el añadir una isla por cada algoritmo glial incrementaría el tiempo de cómputo necesario haciendo que el algoritmo genético no fuese viable si tiene buscar una configuración glial diferente para cada capa. Una aproximación menos costosa computacionalmente podría ser tener dentro del individuo varias configuraciones de los valores gliales dentro de su genotipo para que dependiendo del algoritmo glial que tengan seleccionen una configuración u otra y así no se rompa la configuración de todos los parámetros en la descendencia. Dicho de otra manera, el genotipo tendría una configuración de todos los parámetros gliales necesarios para una red

repetidos para cada algoritmo glial, así de esta manera la primera sección del genotipo se corresponderá con el algoritmo glial uno y solo se modificará con las configuraciones gliales que sean buenas para el algoritmo uno. Con esta aproximación puede ser necesario tener que introducir algo de elitismo y se podría situar en los mejores individuos la mejor configuración de todos los algoritmos gliales que no tienen seleccionado, de tal manera que los mejores individuos tengan buenas configuraciones de todos los algoritmos gliales, lo que permitiría que el algoritmo genético pudiese variar los algoritmos gliales de manera más libre y sin que converja necesariamente a uno para tener que después encontrar los buenos valores para este. En la figura 38 se muestra el ejemplo de la codificación propuesta.

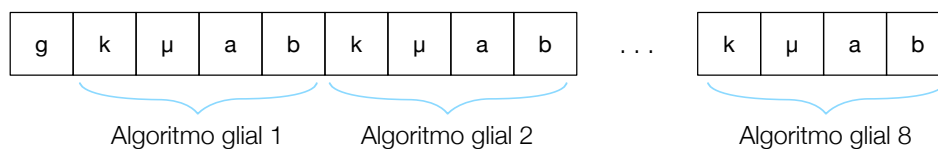


Figura 38. Propuesta de codificación para los parámetros gliales

- **Incorporación de astrocitos sobre redes ya configuradas.** Uno de los pasos computacionalmente más costosos en esta aproximación conexionista es la búsqueda de los pesos de la red. Para esta tesis se ha empleado más de 25 años de cómputo en las máquinas del CESGA. Se puede reducir este tiempo de cómputo analizando si sobre redes ya entrenadas se puede añadir astrocitos artificiales y ver si el desempeño de este se puede separar. Esta aproximación podría reducir notablemente los tiempos de cómputo necesarios para la creación de los modelos de RGA y, también para la búsqueda de nuevas aproximaciones.
- **Dropout en los astrocitos artificiales.** Los estudios recientes confirman la existencia de una sinapsis tripartita, pero indican que no en todas las sinapsis neuronales participan los astrocitos. Es posible que en algunos problemas que el no usar astrocitos en todas las neuronas mejore los modelos de RGA para algunos problemas. Esta observación se puede ver en el uso de los problemas de regresión, donde usar una tercera capa oculta no mejora muchas veces los resultados que se obtienen con dos capas ocultas, y donde además el algoritmo de receptores de

endocannabinoides no generaliza bien en los problemas de regresión, siendo probable que tanta modificación en las conexiones haga que el modelo matemático que forma la RNA se desplace mucho en el espacio debido a la actuación de los astrocitos. Hay que comprobar si la eliminación de unos pocos astrocitos dentro de la red es beneficiosa.

- **Redes de astrocitos.** El uso de RGA presenta un modelo más realista que la RNA pero aún se podría acercar todavía más a la realidad cerebral haciendo que los astrocitos se pudieran comunicar entre ellos e interactuar. Esta aproximación está apoyada biológicamente ya que es conocido que en el sistema nervioso los astrocitos se comunican mediante ondas de calcio y tienen una red paralela a la formada por las neuronas. Esto puede favorecer la producción de mayor cantidad de efectos en el procesamiento de información ya que ambas redes, la astrocítica y la neuronal, pueden compartir el conocimiento almacenado en ambas a la hora de generar una respuesta ante un estímulo.
- **Modelos de astrocitos y memoria.** Las redes LSTM, Long-Short Term Memory, son redes recurrentes que contienen una memoria interna, lo que las hacen unas redes con unas conexiones más realistas que las RNA. La incorporación de astrocitos en ellas ocasionaría redes biológicamente más realistas y que en problemas de series temporales podrían obtener buenos resultados.
- **Deep Learning con astrocitos.** Las redes de Deep Learning están logrando un gran desempeño en estos últimos años. Al ser redes con muchos más elementos de procesado que las RNA y al estar diseñadas para trabajar con problemas complejos son un buen objetivo para añadirles astrocitos artificiales.
- **Sistemas embebidos con modelos conexionistas.** La implementación llevada a cabo en esta tesis de las RNA permiten que el modelo tenga dos grandes ventajas. Estas dos ventajas, la de ahorro de memoria y la rapidez en ejecución, permiten que las simulaciones se ejecuten de manera más rápida y, que se pueda empezar a pensar en usar las redes en pequeños procesadores, como los que usa arduino o la raspberry pi, para pensar en dotar al IoT una capacidad de comportamiento más inteligente.

9. Referencias

- Alfonso, A., Vladimir, P., P., S. R., & G., H. P. (1998). Glutamate-dependent astrocyte modulation of synaptic transmission between cultured hippocampal neurons. *European Journal of Neuroscience*, *10*(6), 2129–2142. <https://doi.org/10.1046/j.1460-9568.1998.00221.x>
- Alger, B. E. (2002). Retrograde signaling in the regulation of synaptic transmission: focus on endocannabinoids. *Progress in Neurobiology*, *68*(4), 247–286. [https://doi.org/10.1016/S0301-0082\(02\)00080-1](https://doi.org/10.1016/S0301-0082(02)00080-1)
- Allen, N. J., & Barres, B. A. (2009). Glia — more than just brain glue. *Nature*, *457*, 675. Retrieved from <http://dx.doi.org/10.1038/457675a>
- Alonso-Betanzos, A., & Guijarro-Berdiñas, B. (2004). *Ingeniería del conocimiento: aspectos metodológicos*. Pearson Educación. Retrieved from <https://books.google.pt/books?id=SrzzaQAACAAJ>
- Alpaydin, E. (1998). Combined 5x2cv F-Test for Comparing Supervised Classification Learning Algorithms.
- Angulo, M. C., Kozlov, A. S., Charpak, S., & Audinat, E. (2004). Glutamate Released from Glial Cells Synchronizes Neuronal Activity in the Hippocampus. *Journal of Neuroscience*, *24*(31), 6920–6927. <https://doi.org/10.1523/JNEUROSCI.0473-04.2004>
- Araque, A., Carmignoto, G., Haydon, P. G., Oliet, S. H. R., Robitaille, R., & Volterra, A. (2014). Gliotransmitters travel in time and space. *Neuron*, *81*(4), 728–739. <https://doi.org/10.1016/j.neuron.2014.02.007>
- Araque, A., Parpura, V., Sanzgiri, R. P., & Haydon, P. G. (1999). Tripartite synapses: glia, the unacknowledged partner. *Trends in Neurosciences*, *22*(5), 208–215. Retrieved from <http://www.ncbi.nlm.nih.gov/pubmed/10322493>
- Asuncion, A., & Newman, D. J. (2007). {UCI} Machine Learning Repository. Retrieved July 17, 2018, from <http://mlr.cs.umass.edu/ml/>
- Azevedo, F. A. C., Carvalho, L. R. B., Grinberg, L. T., Farfel, J. M., Ferretti, R. E. L., Leite, R. E. P., ... Herculano-Houzel, S. (2009). Equal numbers of neuronal and nonneuronal cells make the human brain an isometrically scaled-up primate brain. *Journal of Comparative Neurology*, *513*(5), 532–541. <https://doi.org/10.1002/cne.21974>
- Bazargani, N., & Attwell, D. (2016). Astrocyte calcium signaling: the third wave. *Nature Neuroscience*, *19*, 182. Retrieved from <http://dx.doi.org/10.1038/nn.4201>

- Bear, M. F., Connors, B. W., & Paradiso, M. A. (2007). *Neuroscience*. Lippincott Williams & Wilkins. Retrieved from <https://books.google.es/books?id=75NgwLzueikC>
- BERGILLOS, F. (2013). *TOXINOLOGIA CLINICA. LESIONES POR PICADURAS Y MORDEDURAS DE ANIMALES*. Bubok Publishing. Retrieved from <https://books.google.es/books?id=BZMIWtbjav8C>
- Bliss, T. V. P., & Collingridge, G. L. (1993). A synaptic model of memory: long-term potentiation in the hippocampus. *Nature*, *361*, 31. Retrieved from <http://dx.doi.org/10.1038/361031a0>
- Blum, A. L., & Rivest, R. L. (1992). Training a 3-node neural network is NP-complete. *Neural Networks*, *5*(1), 117–127. [https://doi.org/https://doi.org/10.1016/S0893-6080\(05\)80010-3](https://doi.org/https://doi.org/10.1016/S0893-6080(05)80010-3)
- Buhmann, M. D. (2003). *Radial Basis Functions: Theory and Implementations*. Cambridge University Press. Retrieved from <https://books.google.es/books?id=TRMf53opzlsC>
- Butterworth, J., Cole, L., & Marlow, G. (1993). Inhibition of brain cell excitability by lidocaine, QX314, and tetrodotoxin: a mechanism for analgesia from infused local anesthetics? *Acta Anaesthesiologica Scandinavica*, *37*(5), 516–523. <https://doi.org/10.1111/j.1399-6576.1993.tb03758.x>
- Cantu-Paz, E., & Kamath, C. (2005). An empirical comparison of combinations of evolutionary algorithms and neural networks for classification problems. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, *35*(5), 915–927. <https://doi.org/10.1109/TSMCB.2005.847740>
- Castillo, P. A., Arenas, M. G., Castillo-Valdivieso, J. J., Merelo, J. J., Prieto, A., & Romero, G. (2003). Artificial Neural Networks Design using Evolutionary Algorithms. In J. M. Benítez, O. Cordón, F. Hoffmann, & R. Roy (Eds.), *Advances in Soft Computing* (pp. 43–52). London: Springer London.
- CESGA. (2016). Finisterrae2. Retrieved August 20, 2008, from <https://www.cesga.es/es/infraestructuras/computacion/FinisTerrae2>
- Charles, A. C., Merrill, J. E., Dirksen, E. R., & Sandersont, M. J. (1991). Intercellular signaling in glial cells: Calcium waves and oscillations in response to mechanical stimulation and glutamate. *Neuron*, *6*(6), 983–992. [https://doi.org/https://doi.org/10.1016/0896-6273\(91\)90238-U](https://doi.org/https://doi.org/10.1016/0896-6273(91)90238-U)
- Chevaleyre, V., Takahashi, K. A., & Castillo, P. E. (2006). ENDOCANNABINOID-MEDIATED SYNAPTIC PLASTICITY IN THE CNS. *Annual Review of Neuroscience*, *29*(1), 37–76. <https://doi.org/10.1146/annurev.neuro.29.051605.112834>

- Cooper, S. J. (2005). Donald O. Hebb's synapse and learning rule: a history and commentary. *Neuroscience & Biobehavioral Reviews*, 28(8), 851–874. <https://doi.org/https://doi.org/10.1016/j.neubiorev.2004.09.009>
- Cornell-Bell, A. H., Finkbeiner, S. M., Cooper, M. S., & Smith, S. J. (1990). Glutamate induces calcium waves in cultured astrocytes: long-range glial signaling. *Science*, 247(4941), 470–473. <https://doi.org/10.1126/science.1967852>
- Cornell-Bell, A. H., Thomas, P. G., & Smith, S. J. (1990). The excitatory neurotransmitter glutamate causes filopodia formation in cultured hippocampal astrocytes. *Glia*, 3(5), 322–334. <https://doi.org/10.1002/glia.440030503>
- Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine Learning*, 20(3), 273–297. <https://doi.org/10.1007/BF00994018>
- Cortez, P., & Morais, A. (2007). A Data Mining Approach to Predict Forest Fires using Meteorological Data.
- Cortez, P., & Silva, A. (2008). Using data mining to predict secondary school student performance. *EUROSIS*.
- Darwin, C. (1859). *The Origin of Species by Means of Natural Election, Or the Preservation of Favored Races in the Struggle for Life*.
- De Felipe, J. (2005). Cajal y sus dibujos: ciencia y arte. In *Arte y Neurología* (pp. 213–230). Saned. Retrieved from <http://hdl.handle.net/10261/12879>
- De Jong, K. A. (1975). *Analysis of the behaviour of a class of genetic adaptive systems*.
- Dorado, J. (1999). *Modelo de un sistema para la selección automática en dominios complejos con una estrategia cooperativa, de conjuntos de entrenamiento y arquitecturas ideales de redes de neuronas artificiales utilizando algoritmos genéticos*. Universidade da Coruña. Retrieved from <https://ruc.udc.es/dspace/handle/2183/1063>
- Fellin, T., Pascual, O., Gobbo, S., Pozzan, T., Haydon, P. G., & Carmignoto, G. (2004). Neuronal Synchrony Mediated by Astrocytic Glutamate through Activation of Extrasynaptic NMDA Receptors. *Neuron*, 43(5), 729–743. <https://doi.org/https://doi.org/10.1016/j.neuron.2004.08.011>
- Fiacco, T. A., & McCarthy, K. D. (2004). Intracellular Astrocyte Calcium Waves In Situ Increase the Frequency of Spontaneous AMPA Receptor Currents in CA1 Pyramidal Neurons. *Journal of Neuroscience*, 24(3), 722–732. <https://doi.org/10.1523/JNEUROSCI.2859-03.2004>
- Fisher, R. A. (1936). THE USE OF MULTIPLE MEASUREMENTS IN TAXONOMIC

- PROBLEMS. *Annals of Eugenics*, 7(2), 179–188. <https://doi.org/10.1111/j.1469-1809.1936.tb02137.x>
- Freeman, J. (1991). *Neural networks: algorithms, applications, and programming techniques*. Reading, Mass: Addison-Wesley.
- FREUND, T. F., KATONA, I., & PIOMELLI, D. (2003). Role of Endogenous Cannabinoids in Synaptic Signaling. *Physiological Reviews*, 83(3), 1017–1066. <https://doi.org/10.1152/physrev.00004.2003>
- Gen, M., & Cheng, R. (2000). *Genetic Algorithms and Engineering Optimization*. Wiley. Retrieved from <https://books.google.pt/books?id=U7MuV1q6P1oC>
- Genoud, C., Quairiaux, C., Steiner, P., Hirling, H., Welker, E., & Knott, G. W. (2006). Plasticity of Astrocytic Coverage and Glutamate Transporter Expression in Adult Mouse Cortex. *{PLoS} Biology*, 4(11), e343. <https://doi.org/10.1371/journal.pbio.0040343>
- Gershon, M. D., & Taube, R. P. (1991). Enteric glia. *Glia*, 4(2), 195–204. <https://doi.org/10.1002/glia.440040211>
- Glickstein, M. (2006). Golgi and Cajal: The neuron doctrine and the 100th anniversary of the 1906 Nobel Prize. *Current Biology*, 16(5), R147–R151. <https://doi.org/10.1016/j.cub.2006.02.053>
- Goldberg, D. E. (1991). Real-coded genetic algorithms, virtual alphabets, and blocking. *Complex Systems*, 5(2), 139–167.
- Goldberg, D. E., & Holland, J. H. (1988). Genetic algorithms and machine learning. *Machine Learning*, 3(2), 95–99.
- Grönroos, M. A. (1998). Evolutionary design of neural networks. *Master of Science Thesis in Computer Science. Dept. of Mathematical Sciences. University of Turku*.
- Hagan, M. T., Demuth, H. B., & Beale, M. H. (2014). *Neural network design*. Martin Hagan. Retrieved from <https://books.google.pt/books?id=4EW9oQEACAAJ>
- Haydon, P. G., & Araque, A. (2002). Astrocytes as integrators and modulators of synaptic transmission. In *The tripartite synapse: glia in synaptic transmission* (pp. 185–198). Oxford; New York: Oxford University Press. Retrieved from <https://trove.nla.gov.au/work/17051957?q&versionId=20006811>
- Haykin, S. O. (2009). *Neural Networks and Learning Machines*. Prentice Hall. Retrieved from https://books.google.pt/books?id=K7P36lKzl_QC
- Heaton, J. (2008). *Introduction to Neural Networks with Java*. Heaton Research. Retrieved from <https://books.google.pt/books?id=Swlcw7M4uD8C>

- Herrera, F., Hervás, C., Otero, J., & Sánchez, L. (2004). Un estudio empírico preliminar sobre los tests estadísticos más habituales en el aprendizaje automático. *Tendencias de La Minería de Datos En Espana, Red Espanola de Minería de Datos y Aprendizaje (TIC2002-11124-E)*, 403–412.
- Holland, J. (1975). Adaptation in natural and artificial systems: an introductory analysis with application to biology. *Control and Artificial Intelligence*.
- Innocenti, B., Parpura, V., & Haydon, P. G. (2000). Imaging Extracellular Waves of Glutamate during Calcium Signaling in Cultured Astrocytes. *Journal of Neuroscience*, 20(5), 1800–1808. <https://doi.org/10.1523/JNEUROSCI.20-05-01800.2000>
- Intel. (2017). Intel® Core™ i5-7360U Processor (4M Cache, up to 3.60 GHz) Product Specifications.
- Kang, J., Jiang, L., Goldman, S. A., & Nedergaard, M. (1998). Astrocyte-mediated potentiation of inhibitory synaptic transmission. *Nature Neuroscience*, 1, 683. Retrieved from <http://dx.doi.org/10.1038/3684>
- Khan Academy. (2018). Biología humana. Función y estructura de la Neurona. Retrieved from <https://es.khanacademy.org/science/biology/human-biology/modal/a/overview-of-neuron-structure-and-function>
- Kohonen, T. (1989). *Self-Organization and Associative Memory*. Springer Berlin Heidelberg. <https://doi.org/10.1007/978-3-642-88163-3>
- Kohonen, T. (1995). Learning Vector Quantization. In *Self-Organizing Maps* (pp. 175–189). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-97610-0_6
- Lapedes, A. S., & Farber, R. M. (1988). How neural nets work. In *Neural information processing systems* (pp. 442–456).
- Lisa, W., & Nephi, S. (2003). Endothelin-1 increases 2-arachidonoyl glycerol (2-AG) production in astrocytes. *Glia*, 44(1), 85–90. <https://doi.org/10.1002/glia.10270>
- Liu, Q., Xu, Q., Arcuino, G., Kang, J., & Nedergaard, M. (2004). Astrocyte-mediated activation of neuronal kainate receptors. *Proceedings of the National Academy of Sciences*, 101(9), 3172–3177. <https://doi.org/10.1073/pnas.0306731101>
- Mazzarello, P. (2007). Net without nodes and vice versa, the paradoxical Golgi–Cajal story: A reconciliation? *Brain Research Bulletin*, 71(4), 344–346. <https://doi.org/https://doi.org/10.1016/j.brainresbull.2006.09.022>
- McCulloch, W. S., & Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The Bulletin of Mathematical Biophysics*, 5(4), 115–133. <https://doi.org/10.1007/BF02478259>

- Mesejo, P., Ibáñez, O., Fernández-Blanco, E., Cedrón, F., Pazos, A., & Porto-Pazos, A. B. (2015). Artificial Neuron–Glia Networks Learning Approach Based on Cooperative Coevolution. *International Journal of Neural Systems*, 1550012.
- Michalski, C. W. (2008). Cannabinoids Reduce Markers of Inflammation and Fibrosis in Pancreatic Stellate Cells. <https://doi.org/10.1371/journal.pone.0001701>
- Molina-Holgado, F., Pinteaux, E., Moore, J. D., Molina-Holgado, E., Guaza, C., Gibson, R. M., & Rothwell, N. J. (2003). Endogenous Interleukin-1 Receptor Antagonist Mediates Anti-Inflammatory and Neuroprotective Actions of Cannabinoids in Neurons and Glia. *Journal of Neuroscience*, 23(16), 6470–6474. <https://doi.org/10.1523/JNEUROSCI.23-16-06470.2003>
- Moore, D. (1999). *Introduction to the practice of statistics*. New York: W.H. Freeman.
- Morillo, A. (2008). Contrastes No Paramétricos (II). Universidad de Málaga. Retrieved from <http://campusvirtual.uma.es/morillas/CNOPARII.pdf>.
- Navarrete, M., & Araque, A. (2008). Endocannabinoids Mediate Neuron-Astrocyte Communication. *Neuron*, 57(6), 883–893. <https://doi.org/https://doi.org/10.1016/j.neuron.2008.01.029>
- Nephi, S. (2004). Cannabinoid signaling in glial cells. *Glia*, 48(4), 267–277. <https://doi.org/10.1002/glia.20084>
- Nicholls, J. (2001). *From neuron to brain*. Sunderland, Mass: Sinauer Associates.
- Noback, C. R., Strominger, N. L., Demarest, R. J., & Ruggiero, D. A. (2005). *The Human Nervous System: Structure and Function*. Humana Press. Retrieved from https://books.google.pt/books?id=UnRO3A_cS44C
- Ota, Y., Zanetti, A. T., & Hallock, R. M. (2013). The Role of Astrocytes in the Regulation of Synaptic Plasticity and Memory Formation. *Neural Plasticity*, 2013, 1–11. <https://doi.org/10.1155/2013/185463>
- Parri, R., Gould, T. M., & Crunelli, V. (2004). A heterogeneity of responses to synaptic stimulation in astrocytes in thalamic astrocytes. In *Society for Neuroscience Abstracts* (Vol. 30, pp. 974–976).
- Pastur-Romay, L. A., Porto-Pazos, A. B., Cedron, F., & Pazos, A. (2017). Parallel Computing for Brain Simulation. *Current Topics in Medicinal Chemistry*, 17(14). Retrieved from <https://www.ingentaconnect.com/content/ben/ctmc/2017/00000017/00000014/art00009>
- Pazos-Sierra, A. (1996). *Redes de neuronas artificiales y algoritmos genéticos*. La Coruña: Universidade da Coruña, Servicio de Publicacions.

- Pena-Reyes, C. A., & Sipper, M. (2001). Fuzzy CoCo: a cooperative-coevolutionary approach to fuzzy modeling. *IEEE Transactions on Fuzzy Systems*, 9(5), 727–737. <https://doi.org/10.1109/91.963759>
- Perea, G., & Araque, A. (2002). Communication between astrocytes and neurons: a complex language. *Journal of Physiology-Paris*, 96(3), 199–207. [https://doi.org/https://doi.org/10.1016/S0928-4257\(02\)00007-4](https://doi.org/https://doi.org/10.1016/S0928-4257(02)00007-4)
- Perea, G., & Araque, A. (2005a). Glial calcium signaling and neuron–glia communication. *Cell Calcium*, 38(3), 375–382. <https://doi.org/https://doi.org/10.1016/j.ceca.2005.06.015>
- Perea, G., & Araque, A. (2005b). Properties of Synaptically Evoked Astrocyte Calcium Signal Reveal Synaptic Information Processing by Astrocytes. *Journal of Neuroscience*, 25(9), 2192–2203. <https://doi.org/10.1523/JNEUROSCI.3965-04.2005>
- Perea, G., & Araque, A. (2007). Astrocytes Potentiate Transmitter Release at Single Hippocampal Synapses. *Science*, 317(5841), 1083–1086. <https://doi.org/10.1126/science.1144640>
- Perea, G., & Araque, A. (2010). GLIA modulates synaptic transmission. *Brain Research Reviews*, 63(1), 93–102. <https://doi.org/https://doi.org/10.1016/j.brainresrev.2009.10.005>
- Perea, G., Navarrete, M., & Araque, A. (2009). Tripartite synapses: astrocytes process and control synaptic information. *Trends in Neurosciences*, 32(8), 421–431. <https://doi.org/https://doi.org/10.1016/j.tins.2009.05.001>
- Pereira, A., & Furlan, F. A. (2010). Astrocytes and human cognition: Modeling information integration and modulation of neuronal activity. *Progress in Neurobiology*, 92(3), 405–420. <https://doi.org/https://doi.org/10.1016/j.pneurobio.2010.07.001>
- Pohl, C., & Hof, H.-J. (2015). Secure Scrum: Development of Secure Software with Scrum. *ArXiv E-Prints*.
- Porto-Pazos, A. B., Veiguela, N., Mesejo, P., Navarrete, M., Alvarellos, A., Ibáñez, O., ... Araque, A. (2011). Artificial astrocytes improve neural network performance. *PLoS ONE*, 6(4). <https://doi.org/10.1371/journal.pone.0019109>
- Porto, A. (2004). *Modelos Computacionales para optimizar el Aprendizaje y el Procesamiento de la Información en Sistemas Adaptativos: Redes Neurogliales Artificiales (RR.NG.AA)*. Universidade da Coruña.
- Porto, A., Araque, A., Rabuñal, J., Dorado, J., & Pazos, A. (2007). A new hybrid evolutionary mechanism based on unsupervised learning for Connectionist

- Systems. *Neurocomputing*, 70(16), 2799–2808.
- Porto, A., Pazos, A., & Araque, A. (2005). Artificial neural networks based on brain circuits behaviour and genetic algorithms. In *Computational Intelligence and Bioinspired Systems* (pp. 99–106). Springer.
- Potter, M. A., & De Jong, K. A. (2000). Cooperative Coevolution: An Architecture for Evolving Coadapted Subcomponents. *Evol. Comput.*, 8(1), 1–29. <https://doi.org/10.1162/106365600568086>
- Prudent, Y., & Ennaji, A. (2005). An incremental growing neural gas learns topologies. In *Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005.* (Vol. 2, pp. 1211–1216 vol. 2). <https://doi.org/10.1109/IJCNN.2005.1556026>
- Rabuñal, J. (1998). *Entrenamiento de redes de neuronas artificiales con algoritmos genéticos*. Tesis de Licenciatura. Dep. Computación. Facultad de Informática. Universidade da Coruña.
- Rabuñal, J. R., Dorado, J., Pazos, A., Pereira, J., & Rivero, D. (2004). A New Approach to the Extraction of {ANN} Rules and to Their Generalization Capacity Through {GP}. *Neural Computation*, 16(7), 1483–1523. <https://doi.org/10.1162/089976604323057461>
- Rivero, D., Dorado, J., Fernández-Blanco, E., & Pazos, A. (2009). A Genetic Algorithm for ANN Design, Training and Simplification. In J. Cabestany, F. Sandoval, A. Prieto, & J. M. Corchado (Eds.), *Bio-Inspired Systems: Computational and Ambient Intelligence* (pp. 391–398). Berlin, Heidelberg: Springer Berlin Heidelberg.
- Rodriguez, J. J., Mackie, K., & Pickel, V. M. (2001). Ultrastructural Localization of the CB1 Cannabinoid Receptor in μ -Opioid Receptor Patches of the Rat Caudate Putamen Nucleus. *Journal of Neuroscience*, 21(3), 823–833. <https://doi.org/10.1523/JNEUROSCI.21-03-00823.2001>
- Rosin, C. D., & Belew, R. K. (1997). New methods for competitive coevolution. *Evolutionary Computation*, 5(1), 1–29.
- Rumelhart, D. E., McClelland, J. L., & PDP Research Group, C. (1986). *Parallel distributed processing: explorations in the microstructure of cognition, vol. 1: foundations*. MIT Press. Retrieved from <http://dl.acm.org/citation.cfm?id=104279>
- Russell, S. J., Norvig, P., & Rodríguez, J. M. C. (2004). *Inteligencia artificial: un enfoque moderno*. Pearson Educación. Retrieved from <https://books.google.es/books?id=yZCVPwAACAAJ>
- Salio, C., Doly, S., Fischer, J., Franzoni, M. F., & Conrath, M. (2002). Neuronal and astrocytic localization of the cannabinoid receptor-1 in the dorsal horn of the rat

- spinal cord. *Neuroscience Letters*, 329(1), 13–16. [https://doi.org/https://doi.org/10.1016/S0304-3940\(02\)00549-9](https://doi.org/https://doi.org/10.1016/S0304-3940(02)00549-9)
- Schipke, C. G., & Kettenmann, H. (2004). Astrocyte responses to neuronal activity. *Glia*, 47(3), 226–232. <https://doi.org/10.1002/glia.20029>
- Schlimmer, J. (1987). *Concept acquisition through representational adjustment*. Department of Information and Computer Science, University of California. Retrieved from <http://hdl.handle.net/1822/8039>
- Shen, C., Wang, L., & Li, Q. (2007). Optimization of injection molding process parameters using combination of artificial neural network and genetic algorithm method. *Journal of Materials Processing Technology*, 183(2), 412–418. <https://doi.org/https://doi.org/10.1016/j.jmatprotec.2006.10.036>
- Siddiqi, A. A., & Lucas, S. M. (1998). A comparison of matrix rewriting versus direct encoding for evolving neural networks. In *1998 IEEE International Conference on Evolutionary Computation Proceedings. IEEE World Congress on Computational Intelligence (Cat. No.98TH8360)* (pp. 392–397). <https://doi.org/10.1109/ICEC.1998.699787>
- Sigillito, V. G., Wing, S. P., Hutton, L. V., & Baker, K. B. (1989). Classification of radar returns from the ionosphere using neural networks. *Johns Hopkins APL Tech. Dig*, 10, 262–266.
- SLURM. (2018). Simple Linux Utility for Resource Management. Retrieved from <https://slurm.schedmd.com/>
- Smolensky, P. (1987). Connectionist AI, symbolic AI, and the brain. *Artificial Intelligence Review*, 1(2), 95–109. <https://doi.org/10.1007/BF00130011>
- Sudheer, K. P., & Jain, A. (n.d.). Explaining the internal behaviour of artificial neural network river flow models. *Hydrological Processes*, 18(4), 833–844. <https://doi.org/10.1002/hyp.5517>
- Tan, M., & Eshelman, L. J. (1988). Using Weighted Networks to Represent Classification Knowledge in Noisy Domains. In *ML* (p. 121).
- Theodosis, D. T., Poulain, D. A., & Oliet, S. H. R. (2008). Activity-Dependent Structural and Functional Plasticity of Astrocyte-Neuron Interactions. *Physiological Reviews*, 88(3), 983–1008. <https://doi.org/10.1152/physrev.00036.2007>
- Torres-Fernández, O., Yepes, G. E., & Gómez, J. E. (2007). Neuronal dendritic morphology alterations in the cerebral cortex of rabies-infected mice: a Golgi study. *Biomédica*, 27(4), 605–613. Retrieved from http://www.scielo.org.co/scielo.php?script=sci_arttext&pid=S0120-41572007000400015&lng=en&nrm=iso&tlng=es

- Veigueta, N. (2015). *Procesamiento de información mediante Redes NeuroGliales Artificiales en Clasificación y predicción*. Universidade da Coruña. Retrieved from <http://hdl.handle.net/2183/14716>
- Ventura, R., & Harris, K. M. (1999). Three-Dimensional Relationships between Hippocampal Synapses and Astrocytes. *Journal of Neuroscience*, 19(16), 6897–6906. <https://doi.org/10.1523/JNEUROSCI.19-16-06897.1999>
- Verkhatsky, A., & Butt, A. (2013a). Microglia. In *Glial Physiology and Pathophysiology* (pp. 343–380). Wiley-Blackwell. <https://doi.org/10.1002/9781118402061.ch7>
- Verkhatsky, A., & Butt, A. (2013b). Neuroglia: Definition, Classification, Evolution, Numbers, Development. In *Glial Physiology and Pathophysiology* (pp. 73–104). Wiley-Blackwell. <https://doi.org/10.1002/9781118402061.ch3>
- Verkhatsky, A., & Butt, A. (2013c). NG2-Glial Cells. In *Glial Physiology and Pathophysiology* (pp. 321–341). Wiley-Blackwell. <https://doi.org/10.1002/9781118402061.ch6>
- Verkhatsky, A., & Butt, A. (2013d). Oligodendrocytes. In *Glial Physiology and Pathophysiology* (pp. 245–319). Wiley-Blackwell. <https://doi.org/10.1002/9781118402061.ch5>
- Viñuela, P. (2004). *Redes de neuronas artificiales: un enfoque práctico*. Madrid: Prentice Hall.
- Virchow, R., & CHANCE, F. (1860). *Cellular Pathology as based upon physiological and pathological histology. Twenty lectures delivered in ... 1858. Translated from the second edition of the original by F. Chance. With notes and numerous emendations principally from MS. notes of the author*. Retrieved from <https://books.google.es/books?id=59pUAAAacAAJ>
- Vyklicky, V., Korinek, M., Smejkalova, T., Balik, A., Krausova, B., Kaniakova, M., ... Vyklicky, L. (2014). Structure, function, and pharmacology of NMDA receptor channels. *Physiological Research*, 63 Suppl 1, S191-203.
- Walpole, R. E., Myers, R. H., & Myers, S. L. (1999). *Probabilidad y estadística para ingenieros*. Pearson Educación. Retrieved from <https://books.google.pt/books?id=9DWw696jLbMC>
- Wiegand, R. P. (2003). *An analysis of cooperative coevolutionary algorithms*. Citeseer.
- Wilcoxon, F. (1945). Individual Comparisons by Ranking Methods. *Biometrics Bulletin*, 1(6), 80–83. Retrieved from <http://www.jstor.org/stable/3001968>
- Williams, R. J. (1992). Simple statistical gradient-following algorithms for connectionist reinforcement learning. In *Machine Learning* (pp. 229–256).

- Wilson, R. I., & Nicoll, R. A. (2002). Endocannabinoid Signaling in the Brain. *Science*, 296(5568), 678–682. <https://doi.org/10.1126/science.1063545>
- Xu, R., & Wunsch, D. (2008). *Clustering*. Wiley. Retrieved from https://books.google.es/books?id=kYC3YCyl_tkC
- Yao, X. (1999). Evolving artificial neural networks. *Proceedings of the IEEE*, 87(9), 1423–1447. <https://doi.org/10.1109/5.784219>
- Yeh, I.-C. (1998). Modeling of strength of high-performance concrete using artificial neural networks. *Cement and Concrete Research*, 28(12), 1797–1808. [https://doi.org/https://doi.org/10.1016/S0008-8846\(98\)00165-3](https://doi.org/https://doi.org/10.1016/S0008-8846(98)00165-3)

Apéndice A. Resultados obtenidos en la experimentación

En este apéndice se muestran las tablas con los resultados obtenidos durante la experimentación y que se han usado para la redacción de esta tesis doctoral. Para los problemas de clasificación se muestran los datos de los porcentajes de acierto con dos decimales, mientras que en regresión se emplean cuatro decimales para indicar el ECM obtenido. Para hacer una lectura más rápida de las tablas, las columnas de las RNGA tienen el color verde si funcionan mejor que la RNA y rojo si no lo logran. Como en las tablas no se usan todos los decimales se da el caso de que se muestra el mismo valor para la RNGA y la RNA, cuando realmente la RNGA y la RNA no empatan y en ese caso el código de colores indica cual es la que tiene mejor desempeño. Además, dentro de las columnas RNGA se muestra cual es el par activación-iteración que mejores resultados tiene indicando el valor en negrita y con el color negro.

A.1. Experimentación y resultados en problemas de clasificación

A.1.1. Análisis de astrocitos artificiales en redes con 1 capa oculta

Algoritmos de potenciación astrocítica

Setas venenosas

Como se puede observar en la siguiente tabla, se muestran los porcentajes de acierto medio alcanzados por la RNA y las RNGA en este problema de setas venenosas usando algoritmos de potenciación. Como se puede ver, el resultado obtenido para las combinaciones de activación-iteración 2-4, 2-6, 3-6 y 3-8 en la RNGA no logra superar los valores obtenidos por la RNA a excepción en el algoritmo de atenuación astrocítica donde la combinación de activación 3-8 en entrenamiento consiguió un resultado mejor. Sin embargo, añadiendo en el cross-validation las combinaciones de activación-iteración se logra obtener mejores resultados en todos los algoritmos y para todas las particiones del conjunto de datos. También es importante resaltar el hecho de que, en todos los algoritmos, la configuración de activación-iteración que obtuvo el mejor valor de entrenamiento, también lo ha hecho en la partición de test, y la mayoría de las veces también se consiguió para los datos de validación.

Tabla 6. Resultados en el problema de setas venenosas de la RNGA con el alg. de potenciación y la RNA, obtenidos en el entrenamiento, validación y test con la arquitectura de 1 capa oculta. Los valores que se muestran corresponden al % de acierto medio.

	Alg glial	Conjunto de datos	RNGA					RNA
			2-4	2-6	3-6	3-8	Mejor	
Setas venenosas - 1 capa oculta - 22,7,1	Efecto de atenuación astrocítica	Entre...	94,17%	94,32%	94,22%	95,21%	96,69%	94,39%
		Valid...	92,78%	92,83%	92,75%	93,78%	95,52%	93,84%
		Test	92,66%	92,80%	92,72%	93,80%	95,48%	93,83%
	Act. no consecutivas, modificación sin límite	Entre...	93,51%	92,92%	93,15%	93,17%	95,45%	94,39%
		Valid...	92,12%	91,47%	91,79%	91,95%	94,22%	93,84%
		Test	92,06%	91,29%	91,73%	91,87%	94,21%	93,83%
	Efecto de proceso global	Entre...	93,53%	93,47%	93,74%	93,90%	95,67%	94,39%
		Valid...	92,98%	92,87%	93,06%	93,03%	95,17%	93,84%
		Test	92,95%	92,86%	93,17%	93,39%	95,23%	93,83%
	Act. consecutivas, modificación con límite	Entre...	93,72%	93,83%	93,48%	93,69%	95,79%	94,39%
		Valid...	93,19%	93,30%	92,94%	93,09%	95,38%	93,84%
		Test	93,15%	93,23%	92,86%	93,04%	95,30%	93,83%
	Act. no consecutivas, modificación con límite	Entre...	93,49%	93,27%	93,54%	93,04%	95,46%	94,39%
		Valid...	92,19%	91,93%	92,10%	91,85%	94,27%	93,84%
		Test	92,07%	91,87%	92,10%	91,85%	94,21%	93,83%
	Act. consecutivas, modificación sin límite	Entre...	93,53%	94,04%	93,55%	93,90%	96,07%	94,39%
		Valid...	93,03%	93,35%	92,96%	93,27%	95,50%	93,84%
		Test	93,02%	93,37%	92,91%	93,27%	95,60%	93,83%

Cáncer de mama

Como se puede observar en la siguiente tabla, se muestran los porcentajes de acierto medio alcanzados por la RNA y las RNGA en el problema de cáncer de mama usando algoritmos de potenciación. Como se puede observar, ninguna de las combinaciones de activación-iteración 2-4, 2-6, 3-6 y 3-8 de la RNGA obtienen mejores resultados que los obtenidos con la RNA. Al fijarse en la columna mejor, se puede ver que el error de validación es mejor en la RNGA y que el algoritmo de efecto de atenuación es el único que logra mejorar los resultados en entrenamiento. Se puede observar que para esta configuración de RNGA el resultado de test se le resiste ya que

ninguno de los algoritmos consigue mejorar los resultados. Una peculiaridad es que la combinación que consiguió los mejores resultados en validación también los hizo en la partición de test.

Tabla 7. Resultados en el problema de cáncer de mama de la RNGA con el alg. de potenciación y la RNA, obtenidos en el entrenamiento, validación y test con la arquitectura de 1 capa oculta. Los valores que se muestran corresponden al % de acierto medio.

	Alg glial	Conjunto de datos	RNGA					RNA
			2-4	2-6	3-6	3-8	Mejor	
Cáncer de mama - 1 capa oculta - 9,7,1	Efecto de atenuación astrocítica	Entre...	89,26%	89,60%	89,04%	89,40%	90,12%	90,03%
		Valid...	82,57%	83,03%	82,80%	82,92%	85,35%	85,13%
		Test	82,02%	83,00%	82,32%	82,31%	85,44%	85,83%
	Act. no consecutivas, modificación sin límite	Entre...	89,30%	89,50%	89,71%	89,18%	89,98%	90,03%
		Valid...	82,99%	82,86%	82,58%	83,08%	85,39%	85,13%
		Test	82,04%	82,22%	81,32%	83,16%	84,57%	85,83%
	Efecto de proceso global	Entre...	89,51%	89,42%	89,12%	89,27%	89,95%	90,03%
		Valid...	82,78%	82,85%	83,03%	83,33%	85,49%	85,13%
		Test	81,81%	82,03%	82,42%	82,84%	84,62%	85,83%
	Act. consecutivas, modificación con límite	Entre...	89,00%	89,35%	89,50%	89,53%	89,99%	90,03%
		Valid...	83,23%	83,38%	82,50%	82,72%	85,51%	85,13%
		Test	82,12%	82,98%	81,59%	81,66%	84,72%	85,83%
	Act. no consecutivas, modificación con límite	Entre...	89,56%	89,40%	89,14%	89,47%	89,96%	90,03%
		Valid...	83,55%	82,58%	82,36%	82,63%	85,21%	85,13%
		Test	82,34%	81,83%	81,94%	81,74%	84,35%	85,83%
	Act. consecutivas, modificación sin límite	Entre...	89,46%	89,40%	89,01%	89,73%	89,91%	90,03%
		Valid...	83,11%	83,01%	82,42%	83,02%	85,30%	85,13%
		Test	82,26%	81,66%	82,16%	81,99%	84,47%	85,83%

Señal de ionosfera

Como se puede observar en la siguiente tabla, se muestran los porcentajes de acierto medio alcanzados por la RNA y las RNGA con el problema de ionosfera usando algoritmos de potenciación. Las RNGA con las combinaciones de activación-iteración para el algoritmo de efecto de atenuación y los algoritmos de activaciones no consecutivas tienen mejores resultados en entrenamiento y en algunas de sus combinaciones de activación-iteración también consiguen mejores resultados en

validación. También se puede observar que los algoritmos de activaciones consecutivas consiguen en dos combinaciones mejores resultados en test. Observando la columna de mejor, el algoritmo de efecto atenuación astrocítica y los algoritmos de activaciones no consecutivas consiguen mejores resultados en todas las particiones del conjunto de datos. El algoritmo de efecto de proceso global en ninguno de los casos consigue mejorar el resultado de la RNA y los algoritmos de activaciones no consecutivas consiguen tener mejores resultados en la columna de mejor para las particiones de validación y test, consiguiendo una mejor generalización que la RNA, aunque no pasa lo mismo para los datos presentados en la fase de entrenamiento.

Tabla 8. Resultados en el problema señal de ionosfera de la RNGA con el alg. de potenciación y la RNA, obtenidos en el entrenamiento, validación y test con la arquitectura de 1 capa oculta. Los valores que se muestran corresponden al % de acierto medio.

	Alg grial	Conjunto de datos	RNGA					RNA
			2-4	2-6	3-6	3-8	Mejor	
Señal de ionosfera - 1 capa oculta - 34,9,1	Efecto de atenuación astrocítica	Entre...	87,40%	87,90%	87,52%	87,48%	89,37%	82,23%
		Valid...	76,52%	77,61%	77,94%	76,74%	81,77%	76,57%
		Test	77,91%	79,57%	78,48%	77,58%	82,93%	80,94%
	Act. no consecutivas, modificación sin límite	Entre...	85,69%	86,26%	85,93%	85,78%	87,31%	82,23%
		Valid...	75,91%	76,80%	75,07%	75,04%	80,25%	76,57%
		Test	77,51%	77,00%	76,72%	76,60%	81,24%	80,94%
	Efecto de proceso global	Entre...	80,98%	81,77%	80,70%	83,07%	81,47%	82,23%
		Valid...	71,62%	60,72%	60,38%	51,83%	74,64%	76,57%
		Test	80,83%	80,87%	81,59%	79,88%	80,92%	80,94%
	Act. consecutivas, modificación con límite	Entre...	78,97%	77,93%	79,02%	78,60%	80,94%	82,23%
		Valid...	75,39%	75,12%	75,09%	75,57%	79,07%	76,57%
		Test	81,12%	81,86%	80,59%	80,73%	84,38%	80,94%
	Act. no consecutivas, modificación con límite	Entre...	85,66%	86,66%	85,89%	85,55%	87,53%	82,23%
		Valid...	75,56%	76,87%	75,43%	75,37%	80,27%	76,57%
		Test	77,40%	77,43%	76,20%	77,71%	82,19%	80,94%
	Act. consecutivas, modificación sin límite	Entre...	79,39%	78,81%	78,59%	79,40%	81,38%	82,23%
		Valid...	75,50%	74,24%	74,80%	75,79%	79,24%	76,57%
		Test	82,54%	80,61%	80,25%	81,65%	84,98%	80,94%

Algoritmo de depresión astrocítica

Setas venenosas

Como se puede observar en la siguiente tabla, se muestran los porcentajes de acierto medio alcanzados por la RNA y las RNGA con el problema de setas venenosas usando el algoritmo de depresión. Se puede ver que la combinación 2-4 de activación-iteración consiguió los mejores resultados para las tres particiones del conjunto de datos. Se puede observar que el único resultado que mejora la RNGA frente a la RNA está en la columna mejor y solo se obtiene para los datos de entrenamiento.

Tabla 9. Resultados en el problema de setas venenosas de la RNGA con el alg. de depresión y la RNA, obtenidos en el entrenamiento, validación y test con la arquitectura de 1 capa oculta. Los valores que se muestran corresponden al % de acierto medio.

Conjunto de datos	RNGA					RNA
	2-4	2-6	3-6	3-8	Mejor	
Entre...	93,20%	92,30%	92,33%	92,48%	94,88%	94,39%
Valid...	91,77%	90,85%	91,02%	90,95%	93,62%	93,84%
Test	91,77%	90,67%	90,99%	90,94%	93,49%	93,83%

Cáncer de mama

Como se puede observar en la siguiente tabla, se muestran los porcentajes de acierto medio alcanzados por la RNA y las RNGA con el problema de cáncer de mama usando el algoritmo de depresión. Para esta arquitectura de red, ninguno de los casos de RNGA mejora los resultados de RNA.

Tabla 10. Resultados en el problema cáncer de mama de la RNGA con el alg. de depresión y la RNA, obtenidos en el entrenamiento, validación y test con la arquitectura de 1 capa oculta. Los valores que se muestran corresponden al % de acierto medio.

Conjunto de datos	RNGA					RNA
	2-4	2-6	3-6	3-8	Mejor	
Entre...	89,34%	89,08%	89,15%	89,20%	89,71%	90,03%
Valid...	83,28%	83,58%	82,86%	83,23%	85,43%	85,13%
Test	83,20%	83,36%	83,70%	83,08%	84,98%	85,83%

Señal de ionosfera

Como se puede observar en la siguiente tabla, se muestran los porcentajes de acierto medio alcanzados por la RNA y las RNGA con el problema de ionosfera usando el algoritmo de depresión. Se puede observar que en todas las combinaciones la RNGA supera los resultados de la RNA, pero no logra generalizar tanto como la RNA. Si se observa la columna mejor, se observa que la RNGA no logra generalizar para los datos de test como lo hace la RNA.

Tabla 11. Resultados en el problema señal de ionosfera de la RNGA con el alg. de depresión y la RNA, obtenidos en el entrenamiento, validación y test con la arquitectura de 1 capa oculta. Los valores que se muestran corresponden al % de acierto medio.

Conjunto de datos	RNGA					RNA
	2-4	2-6	3-6	3-8	Mejor	
Entre...	86,14%	86,08%	86,78%	85,89%	87,40%	82,23%
Valid...	76,38%	76,13%	75,74%	74,58%	80,22%	76,57%
Test	75,75%	76,32%	75,60%	74,41%	80,07%	80,94%

Algoritmo de receptores endocannabinoides astrocíticos

Setas venenosas

Como se puede observar en la siguiente tabla, se muestran los porcentajes de acierto medio alcanzados por la RNA y las RNGA con el problema de setas venenosas usando el algoritmo de receptores de endocannabinoides. Se puede observar que la mejor combinación de activación-iteración de RNGA es la 3-6, ya que consigue tener mejores resultados en entrenamiento, validación y test que los obtenidos con las otras combinaciones de RNGA; sin embargo, solo consigue mejorar a la RNA en los datos de entrenamiento. Si nos fijamos en la columna mejor, los datos de la RNGA superan con creces los resultados obtenidos con la RNA.

Tabla 12. Resultados en el problema de setas venenosas de la RNGA con el alg. de receptores de endocannabinoides y la RNA, obtenidos en el entrenamiento, validación y test con la arquitectura de 1 capa oculta. Los valores que se muestran corresponden al % de acierto

Conjunto de datos	RNGA					RNA
	2-4	2-6	3-6	3-8	Mejor	
Entre...	94,27%	93,83%	94,62%	93,82%	96,74%	94,39%
Valid...	93,08%	92,61%	93,34%	92,67%	95,66%	93,84%
Test	92,94%	92,59%	93,28%	92,58%	95,58%	93,83%

Cáncer de mama

Como se puede observar en la siguiente tabla, se muestran los porcentajes de acierto medio alcanzados por la RNA y las RNGA con el problema de cáncer de mama usando el algoritmo de receptores de endocannabinoides. Se puede ver que las combinaciones de activación-iteración 2-4, 2-6, 3-6, 3-8 no logran mejorar la RNA. Además, en la columna mejor solo en el caso de validación se consiguen mejores resultados que los obtenidos con la RNA.

Tabla 13. Resultados en el problema cáncer de mama de la RNGA con el alg. de receptores de endocannabinoides y la RNA, obtenidos en el entrenamiento, validación y test con la arquitectura de 1 capa oculta. Los valores que se muestran corresponden al % de acierto medio.

Conjunto de datos	RNGA					RNA
	2-4	2-6	3-6	3-8	Mejor	
Entre...	89,10%	89,17%	89,32%	89,47%	89,91%	90,03%
Valid...	83,72%	83,98%	83,69%	83,49%	85,79%	85,13%
Test	83,64%	83,13%	83,05%	83,73%	84,93%	85,83%

Señal de ionosfera

Como se puede observar en la siguiente tabla, se muestran los porcentajes de acierto medio alcanzados por la RNA y las RNGA con el problema de ionosfera usando el algoritmo de receptores de endocannabinoides. Se puede observar que la mejor combinación de activación-iteración de RNGA es la 3-8 y que al igual que las combinaciones de RNGA obtienen más aciertos que la RNA para los datos de entrenamiento. Al fijarse en la columna mejor, se puede ver que la RNGA obtiene mejores resultados en los errores de entrenamiento y validación que la RNA, pero la RNA consigue generalizar mejor con los datos de test que la RNGA.

Tabla 14. Resultados en el problema señal de ionosfera de la RNGA con el alg. de receptores de endocannabinoides y la RNA, obtenidos en el entrenamiento, validación y test con la arquitectura de 1 capa oculta. Los valores que se muestran corresponden al % de acierto medio.

Conjunto de datos	RNGA					RNA
	2-4	2-6	3-6	3-8	Mejor	
Entre...	86,65%	84,97%	86,09%	86,85%	88,03%	82,23%
Valid...	75,69%	74,28%	75,56%	76,02%	80,72%	76,57%
Test	76,27%	74,13%	76,24%	76,56%	80,34%	80,94%

Búsqueda de algoritmo astrocítico usando computación coevolutiva

Setas venenosas

Como se puede observar en la siguiente tabla, se muestran los porcentajes de acierto medio alcanzados por la RNA y las RNGA empleando un AGCC para el problema de setas venenosas. Se puede observar que todos los resultados obtenidos con la RNA son mejores que los obtenidos con RNGA. Aun así, es importante ver que eliminar la restricción de que el porcentaje de incremento sea menor que el porcentaje de decremento no mejora los resultados de la RNGA, pero sí que la RNGA donde cada capa usa su propio algoritmo glial es mejor que la RNGA que tienen el mismo algoritmo glial para toda la red.

Tabla 15. Resultados en el problema de setas venenosas de la RNGA creada con AGCC y la RNA, obtenidos en el entrenamiento, validación y test con la arquitectura de 1 capa oculta. Los valores que se muestran corresponden al % de acierto medio.

Setas venenosas - 1 capa oculta - 22,7,1	Info	Conjunto de datos	RNGA + AGCC	RNA
	Mismo Alg glial toda la red. %Dec ≥ %Inc	Entre...	93,70%	94,39%
		Valid...	93,22%	93,84%
		Test	93,12%	93,83%
	Mismo Alg glial toda la red	Entre...	93,54%	94,39%
		Valid...	93,04%	93,84%
		Test	93,08%	93,83%
	Mismo Alg glial por capas	Entre...	94,17%	94,39%
		Valid...	93,67%	93,84%
		Test	93,66%	93,83%
	Mismo Alg glial por capas. %Dec ≥ %Inc	Entre...	93,97%	94,39%
		Valid...	93,45%	93,84%
		Test	93,39%	93,83%

Cáncer de mama

Como se puede observar en la siguiente tabla, se muestran los porcentajes de acierto medio alcanzados por la RNA y las RNGA empleando un AGCC para el problema de cáncer de mama. Se puede observar que la RNGA es mejor que la RNA salvo la excepción donde la RNGA usa el mismo algoritmo glial para toda la red y no hay restricción en los porcentajes de incremento y decremento.

Tabla 16. Resultados en el problema cáncer de mama de la RNGA creada con AGCC y la RNA, obtenidos en el entrenamiento, validación y test con la arquitectura de 1 capa oculta. Los valores que se muestran corresponden al % de acierto medio.

Cáncer de mama – 1 capa – 9,7,1	Info	Conjunto de datos	RNGA + AGCC	RNA
	Mismo Alg glial toda la red. %Dec \geq %Inc	Entre...	90,41%	90,03%
		Valid...	86,14%	85,13%
		Test	86,15%	85,83%
	Mismo Alg glial toda la red	Entre...	90,30%	90,03%
		Valid...	85,87%	85,13%
		Test	85,52%	85,83%
	Mismo Alg glial por capas	Entre...	90,56%	90,03%
		Valid...	86,09%	85,13%
		Test	86,21%	85,83%
	Mismo Alg glial por capas. %Dec \geq %Inc	Entre...	90,21%	90,03%
		Valid...	86,20%	85,13%
		Test	86,21%	85,83%

Señal de ionosfera

Como se puede observar en la siguiente tabla, se muestran los porcentajes de acierto medio alcanzados por la RNA y las RNGA empleando un AGCC para el problema de ionosfera. Se puede observar que el uso de un algoritmo glial por capa en las RNGA mejora los errores de entrenamiento para las RNA. En el caso específico donde la RNGA usa el mismo algoritmo glial para toda la red y tiene la restricción en la que el porcentaje de incremento no puede superar el de decremento se obtiene mejores datos en validación que la RNA. Para esta arquitectura y para este problema, al observar los valores obtenidos con los datos de test, se puede ver que la RNA tiene una mejor capacidad de generalización que la RNGA.

Tabla 17. Resultados en el problema señal de ionosfera de la RNGA creada con AGCC y la RNA, obtenidos en el entrenamiento, validación y test con la arquitectura de 1 capa oculta. Los valores que se muestran corresponden al % de acierto medio.

Señal de ionosfera - 1 capa oculta - 34,9,1	Info	Conjunto de datos	RNGA + AGCC	RNA
	Mismo Alg glial toda la red. %Dec \geq %Inc	Entre...	81,81%	82,23%
		Valid...	76,99%	76,57%
		Test	80,78%	80,94%
	Mismo Alg glial toda la red	Entre...	81,15%	82,23%
		Valid...	76,53%	76,57%
		Test	79,84%	80,94%
	Mismo Alg glial por capas	Entre...	82,40%	82,23%
		Valid...	76,54%	76,57%
		Test	80,45%	80,94%
	Mismo Alg glial por capas. %Dec \geq %Inc	Entre...	83,17%	82,23%
		Valid...	76,47%	76,57%
		Test	80,31%	80,94%

A.1.2. Análisis de astrocitos artificiales en redes con 2 capas ocultas

Algoritmos de potenciación astrocítica

Setas venenosas

Como se puede observar en la siguiente tabla, se muestran los porcentajes de acierto medio alcanzados por la RNA y las RNGA con el problema de setas venenosas usando algoritmos de potenciación. Se puede observar que los resultados obtenidos con la RNGA usando el algoritmo de efecto de atenuación astrocítica con la combinación de activación-iteración 3-8 son muy buenos y mejoran los que se obtienen con la RNA. Se puede observar que, en el resto de los casos, solo los algoritmos de efecto de atenuación y activaciones no consecutivas tienen mejores resultados en entrenamiento que las RNA. Al fijarse en la columna mejor, se puede observar que todos los algoritmos gliales usados en la RNGA mejoran los valores obtenidos por la RNA.

Tabla 18. Resultados en el problema de setas venenosas de la RNGA con el alg. de potenciación y la RNA, obtenidos en el entrenamiento, validación y test con la arquitectura de 2 capas ocultas. Los valores que se muestran corresponden al % de acierto medio.

	Alg glial	Conjunto de datos	RNGA					RNA
			2-4	2-6	3-6	3-8	Mejor	
Setas venenosas - 2 capas ocultas - 22,7,4,1	Efecto de atenuación astrocítica	Entre...	94,03%	94,58%	94,21%	94,73%	96,37%	93,76%
		Valid...	92,69%	92,87%	92,93%	93,32%	95,24%	93,27%
		Test	92,72%	92,91%	92,81%	93,25%	95,19%	93,13%
	Act. no consecutivas, modificación sin límite	Entre...	93,80%	93,86%	93,86%	93,68%	95,93%	93,76%
		Valid...	92,52%	92,42%	92,21%	92,31%	94,80%	93,27%
		Test	92,44%	92,24%	92,29%	92,26%	94,72%	93,13%
	Efecto de proceso global	Entre...	93,67%	93,12%	93,34%	93,25%	95,43%	93,76%
		Valid...	92,99%	92,39%	92,58%	92,20%	94,71%	93,27%
		Test	93,05%	92,49%	92,72%	92,60%	94,90%	93,13%
	Act. consecutivas, modificación con límite	Entre...	92,85%	93,02%	92,90%	92,82%	95,19%	93,76%
		Valid...	92,08%	92,02%	92,17%	92,07%	94,43%	93,27%
		Test	92,00%	91,96%	92,06%	91,99%	94,36%	93,13%
	Act. no consecutivas, modificación con límite	Entre...	93,98%	93,72%	93,91%	93,86%	96,08%	93,76%
		Valid...	92,41%	91,99%	92,37%	92,47%	94,81%	93,27%
		Test	92,41%	91,91%	92,32%	92,27%	94,69%	93,13%
	Act. consecutivas, modificación sin límite	Entre...	93,07%	93,22%	92,96%	93,17%	95,26%	93,76%
		Valid...	92,30%	92,30%	92,23%	92,47%	94,51%	93,27%
		Test	92,28%	92,21%	92,03%	92,41%	94,37%	93,13%

Cáncer de mama

Como se puede observar en la siguiente tabla, se muestran los porcentajes de acierto medio alcanzados por la RNA y las RNGA con el problema de cáncer de mama usando algoritmos de potenciación. Al fijarse en la columna mejor, se puede ver que la RNGA tiene mejores resultados para la RNA, y si observa que en las combinaciones de activación-iteración 2-4, 2-6, 3-6 y 3-8 de la RNGA en 5 de los 6 algoritmo gliales se consiguen mejores resultados que la RNA. El único algoritmo que flagea es el de efecto de proceso global, donde únicamente en la combinación 3-8 de activación-iteración se consiguen mejores resultados para los datos de entrenamiento y de test.

Tabla 19. Resultados en el problema cáncer de mama de la RNGA con el alg. de potenciación y la RNA, obtenidos en el entrenamiento, validación y test con la arquitectura de 2 capas ocultas. Los valores que se muestran corresponden al % de acierto medio.

	Alg glial	Conjunto de datos	RNGA					RNA
			2-4	2-6	3-6	3-8	Mejor	
Cáncer de mama - 2 capas ocultas - 9,7,3,1	Efecto de atenuación astrocítica	Entre...	92,68%	92,81%	92,74%	93,16%	93,06%	89,91%
		Valid...	87,68%	88,27%	87,69%	87,91%	89,94%	84,53%
		Test	88,77%	88,70%	88,40%	89,23%	90,44%	85,45%
	Act. no consecutivas, modificación sin límite	Entre...	92,40%	92,53%	92,16%	92,09%	92,87%	89,91%
		Valid...	87,60%	87,59%	87,09%	87,14%	89,41%	84,53%
		Test	88,37%	88,01%	87,20%	87,35%	89,81%	85,45%
	Efecto de proceso global	Entre...	89,39%	89,70%	89,60%	90,49%	90,10%	89,91%
		Valid...	83,78%	83,98%	83,67%	83,13%	85,91%	84,53%
		Test	84,78%	85,12%	85,29%	85,90%	86,17%	85,45%
	Act. consecutivas, modificación con límite	Entre...	91,38%	91,46%	90,91%	91,13%	92,15%	89,91%
		Valid...	86,62%	87,07%	86,77%	86,47%	88,59%	84,53%
		Test	86,98%	87,55%	86,99%	87,36%	89,00%	85,45%
	Act. no consecutivas, modificación con límite	Entre...	92,30%	92,42%	91,95%	92,00%	92,82%	89,91%
		Valid...	87,45%	86,97%	87,02%	87,34%	89,33%	84,53%
		Test	87,80%	87,83%	87,22%	87,94%	89,25%	85,45%
	Act. consecutivas, modificación sin límite	Entre...	91,14%	91,44%	90,77%	91,07%	92,27%	89,91%
		Valid...	86,58%	86,60%	86,00%	86,33%	88,56%	84,53%
		Test	86,84%	87,17%	87,03%	86,70%	88,46%	85,45%

Señal de ionosfera

Como se puede observar en la siguiente tabla, se muestran los porcentajes de acierto medio alcanzados por la RNA y las RNGA con el problema de ionosfera usando algoritmos de potenciación. Como podemos ver en la tabla, todas las combinaciones de activación-iteración para los algoritmos de activaciones consecutivas y el algoritmo de atenuación consiguen mejores resultados en entrenamiento y validación para la RNGA, aunque si nos fijamos en los datos de test, la RNGA no consigue generalizar tanto como la RNA. Para los algoritmos de activaciones no consecutivas, las combinaciones de activación-iteración mejoran únicamente a la RNA en los datos de entrenamiento. Al fijarse en el algoritmo de proceso global, se puede ver que la RNGA en tres de cuatro combinaciones de activación-iteración consigue mejores resultados

de entrenamiento y fijándonos en los datos de test también es capaz de generalizar mejor que la RNA. Si se observa la información de la columna mejor la RNGA es mejor que la RNA a excepción de un único caso en el que el error de validación usando la RNGA con el algoritmo de efecto de proceso global es inferior que el acierto obtenido con RNA.

Tabla 20. Resultados en el problema señal de ionosfera de la RNGA con el alg. de potenciación y la RNA, obtenidos en el entrenamiento, validación y test con la arquitectura de 2 capas ocultas. Los valores que se muestran corresponden al % de acierto medio.

	Alg glial	Conjunto de datos	RNGA					RNA
			2-4	2-6	3-6	3-8	Mejor	
Señal de ionosfera - 2 capas ocultas - 34,9,4,1	Efecto de atenuación astrocítica	Entre...	87,72%	86,36%	87,85%	87,07%	88,32%	81,82%
		Valid...	76,15%	76,64%	76,00%	76,77%	81,31%	75,75%
		Test	76,36%	77,37%	77,66%	77,91%	81,31%	80,01%
	Act. no consecutivas, modificación sin límite	Entre...	86,34%	86,39%	86,70%	86,14%	87,25%	81,82%
		Valid...	74,72%	76,27%	74,92%	75,02%	79,79%	75,75%
		Test	74,74%	76,97%	75,56%	75,23%	80,59%	80,01%
	Efecto de proceso global	Entre...	81,06%	82,36%	83,64%	85,16%	82,55%	81,82%
		Valid...	67,57%	63,16%	64,12%	54,32%	74,78%	75,75%
		Test	82,18%	79,94%	80,91%	78,56%	82,43%	80,01%
	Act. consecutivas, modificación con límite	Entre...	82,81%	83,89%	82,91%	83,03%	84,49%	81,82%
		Valid...	76,06%	76,45%	76,84%	76,44%	81,01%	75,75%
		Test	78,31%	78,48%	80,00%	78,65%	84,08%	80,01%
	Act. no consecutivas, modificación con límite	Entre...	86,58%	86,25%	86,49%	86,11%	87,55%	81,82%
		Valid...	75,44%	75,72%	75,36%	75,21%	79,42%	75,75%
		Test	74,00%	76,12%	76,49%	75,28%	80,24%	80,01%
	Act. consecutivas, modificación sin límite	Entre...	83,62%	84,02%	82,65%	83,06%	85,61%	81,82%
		Valid...	75,92%	76,06%	76,43%	76,64%	81,15%	75,75%
		Test	78,38%	78,08%	79,79%	79,49%	83,99%	80,01%

Algoritmo de depresión astrocítica

Setas venenosas

Como se puede observar en la siguiente tabla, se muestran los porcentajes de acierto medio alcanzados por la RNA y las RNGA con el problema de setas venenosas usando el algoritmo de depresión. Como se puede observar, la combinación de activación-iteración 3-8 de la RNGA, que es la mejor combinación de RNGA, es la única que consigue mejorar la RNA y solo lo hace para los datos de entrenamiento. Sin embargo, al fijarse en la columna mejor, se observa que los datos obtenidos por la RNGA superan notablemente los resultados obtenidos por la RNA.

Tabla 21. Resultados en el problema de setas venenosas de la RNGA con el alg. de depresión y la RNA, obtenidos en el entrenamiento, validación y test con la arquitectura de 2 capas ocultas. Los valores que se muestran corresponden al % de acierto medio.

Conjunto de datos	RNGA					RNA
	2-4	2-6	3-6	3-8	Mejor	
Entre...	92,90%	93,46%	93,37%	93,88%	95,79%	93,76%
Valid...	91,79%	92,10%	91,90%	92,62%	94,58%	93,27%
Test	91,73%	92,07%	91,86%	92,62%	94,54%	93,13%

Cáncer de mama

Como se puede observar en la siguiente tabla, se muestran los porcentajes de acierto medio alcanzados por la RNA y las RNGA con el problema de cáncer de mama usando el algoritmo de depresión. Se puede ver que la mejor combinación de RNGA es la que tiene la combinación de activación-iteración 3-8, pero como se puede observar todas las RNGA han superado los resultados de la RNA.

Tabla 22. Resultados en el problema cáncer de mama de la RNGA con el alg. de depresión y la RNA, obtenidos en el entrenamiento, validación y test con la arquitectura de 2 capas ocultas. Los valores que se muestran corresponden al % de acierto medio.

Conjunto de datos	RNGA					RNA
	2-4	2-6	3-6	3-8	Mejor	
Entre...	91,96%	93,21%	92,91%	93,73%	93,86%	89,91%
Valid...	87,50%	89,75%	88,85%	90,43%	91,47%	84,53%
Test	88,03%	91,08%	90,10%	91,18%	92,31%	85,45%

Señal de ionosfera

Como se puede observar en la siguiente tabla, se muestran los porcentajes de acierto medio alcanzados por la RNA y las RNGA con el problema de ionosfera usando el algoritmo de depresión. Se puede ver que todas las combinaciones de RNGA obtienen un mejor resultado para entrenamiento y solo las combinaciones de activación-iteración 2-6 y 3-8 tienen más aciertos en validación que la RNA. Sin embargo, al fijarse en la columna mejor, se puede observar que la RNGA obtienen mejores resultados que la RNA.

Tabla 23. Resultados en el problema señal de ionosfera de la RNGA con el alg. de depresión y la RNA, obtenidos en el entrenamiento, validación y test con la arquitectura de 2 capas ocultas. Los valores que se muestran corresponden al % de acierto medio.

Conjunto de datos	RNGA					RNA
	2-4	2-6	3-6	3-8	Mejor	
Entre...	85,69%	85,77%	86,59%	86,10%	87,53%	81,82%
Valid...	75,07%	76,76%	75,15%	77,19%	80,93%	75,75%
Test	75,14%	78,02%	76,18%	79,20%	83,06%	80,01%

Algoritmo de receptores endocannabinoides astrocíticos

Setas venenosas

Como se puede observar en la siguiente tabla, se muestran los porcentajes de acierto medio alcanzados por la RNA y las RNGA con el problema de setas venenosas usando el algoritmo de receptores de endocannabinoides. La mejor RNGA es la que tiene la combinación 2-4 que al igual que la combinación 3-6 supera a la RNA en los datos de entrenamiento, validación y test, mientras que las otras combinaciones mejoran solo mejoran a la RNA en la fase de entrenamiento. En cuando a la columna mejor, se puede observar que la RNGA obtiene muchos más aciertos que la RNA.

Tabla 24. Resultados en el problema de setas venenosas de la RNGA con el alg. de receptores de endocannabinoides y la RNA, obtenidos en el entrenamiento, validación y test con la arquitectura de 2 capas ocultas. Los valores que se muestran corresponden al % de acierto

Conjunto de datos	RNGA					RNA
	2-4	2-6	3-6	3-8	Mejor	
Entre...	94,91%	94,44%	94,70%	94,13%	96,80%	93,76%
Valid...	93,58%	93,08%	93,42%	92,97%	95,67%	93,27%
Test	93,51%	92,98%	93,28%	92,95%	95,58%	93,13%

Cáncer de mama

Como se puede observar en la siguiente tabla, se muestran los porcentajes de acierto medio alcanzados por la RNA y las RNGA con el problema de setas venenosas usando el algoritmo de receptores de endocannabinoides. Como se puede observar para este problema usando los datos de endocannabinoides obtenemos mejores resultados con la RNGA que con la RNA. La única excepción donde la RNA supera la RNGA es usando la combinación 2-6 donde el valor de test es un poco mejor con la RNA.

Tabla 25. Resultados en el problema cáncer de mama de la RNGA con el alg. de receptores de endocannabinoides y la RNA, obtenidos en el entrenamiento, validación y test con la arquitectura de 2 capas ocultas. Los valores que se muestran corresponden al % de acierto medio.

Conjunto de datos	RNGA					RNA
	2-4	2-6	3-6	3-8	Mejor	
Entre...	92,30%	92,53%	92,29%	92,23%	93,25%	89,91%
Valid...	86,43%	85,11%	86,10%	85,89%	88,28%	84,53%
Test	86,50%	84,97%	86,46%	85,58%	88,28%	85,45%

Señal de ionosfera

Como se puede observar en la siguiente tabla, se muestran los porcentajes de acierto medio alcanzados por la RNA y las RNGA con el problema de ionosfera usando el algoritmo de receptores de endocannabinoides. La RNGA obtiene mejores resultados en entrenamiento que la RNA. Además, la combinación 2-4 obtiene mejores resultados para todas las particiones de los datos. Además, para las combinaciones que tienen tres activaciones de la glía, se consiguen mejores resultados para entrenamiento y validación, pero la RNGA

Tabla 26. Resultados en el problema señal de ionosfera de la RNGA con el alg. de receptores de endocannabinoides y la RNA, obtenidos en el entrenamiento, validación y test con la arquitectura de 2 capas ocultas. Los valores que se muestran corresponden al % de acierto medio.

Conjunto de datos	RNGA					RNA
	2-4	2-6	3-6	3-8	Mejor	
Entre...	86,01%	85,75%	85,82%	86,22%	87,89%	81,82%
Valid...	77,30%	75,64%	77,10%	77,89%	81,71%	75,75%
Test	80,25%	76,96%	79,26%	79,26%	83,14%	80,01%

Búsqueda de algoritmo astrocítico usando computación coevolutiva

Setas venenosas

Como se puede observar en la siguiente tabla, se muestran los porcentajes de acierto medio alcanzados por la RNA y las RNGA empleando un AGCC para el problema de setas venenosas. Para la arquitectura de dos capas en el problema de setas venenosas, la RNGA generaliza mejor para la RNA en las simulaciones que se estableció la restricción de que el porcentaje de incremento tiene que ser menor que el de decremento. En el caso donde no se establece la restricción de los porcentajes la RNA obtiene mejores resultados que la RNGA.

Tabla 27. Resultados en el problema de setas venenosas de la RNGA creada con AGCC y la RNA, obtenidos en el entrenamiento, validación y test con la arquitectura de 2 capas ocultas. Los valores que se muestran corresponden al % de acierto medio.

Setas venenosas - 2 capas ocultas - 22,7,4,1	Info	Conjunto de datos	RNGA + AGCC	RNA	
	Mismo Alg glial toda la red. %Dec \geq %Inc	Entre...		93,75%	93,76%
		Valid...		93,07%	93,27%
		Test		93,15%	93,13%
	Mismo Alg glial toda la red	Entre...		93,63%	93,76%
		Valid...		93,03%	93,27%
		Test		93,11%	93,13%
	Mismo Alg glial por capas	Entre...		93,58%	93,76%
		Valid...		93,05%	93,27%
		Test		92,93%	93,13%
	Mismo Alg glial por capas. %Dec \geq %Inc	Entre...		93,85%	93,76%
		Valid...		93,16%	93,27%
		Test		93,21%	93,13%

Cáncer de mama

Como se puede observar en la siguiente tabla, se muestran los porcentajes de acierto medio alcanzados por la RNA y las RNGA empleando un AGCC para el problema de cáncer de mama. Se puede observar que, para todas las simulaciones, el acierto es mayor empleando la RNGA en comparación con la RNA.

Tabla 28. Resultados en el problema cáncer de mama de la RNGA creada con AGCC y la RNA, obtenidos en el entrenamiento, validación y test con la arquitectura de 2 capas ocultas. Los valores que se muestran corresponden al % de acierto medio.

Cáncer de mama - 2 capas ocultas - 9,7,3,1	Info	Conjunto de datos	RNGA + AGCC	RNA
	Mismo Alg glial toda la red. %Dec \geq %Inc	Entre...	91,25%	89,91%
		Valid...	87,38%	84,53%
		Test	87,07%	85,45%
	Mismo Alg glial toda la red	Entre...	91,36%	89,91%
		Valid...	87,42%	84,53%
		Test	87,78%	85,45%
	Mismo Alg glial por capas	Entre...	91,72%	89,91%
		Valid...	87,65%	84,53%
		Test	87,62%	85,45%
	Mismo Alg glial por capas. %Dec \geq %Inc	Entre...	91,65%	89,91%
		Valid...	86,80%	84,53%
		Test	86,87%	85,45%

Señal de ionosfera

Como se puede observar en la siguiente tabla, se muestran los porcentajes de acierto medio alcanzados por la RNA y las RNGA empleando un AGCC para el problema de señal de ionosfera. Para esta arquitectura de dos capas la RNGA es capaz de obtener mejores resultados de entrenamiento y validación en todas las simulaciones, sin embargo, la RNA es capaz de generalizar mejor.

Tabla 29. Resultados en el problema señal de ionosfera de la RNGA creada con AGCC y la RNA, obtenidos en el entrenamiento, validación y test con la arquitectura de 2 capas ocultas. Los valores que se muestran corresponden al % de acierto medio.

Señal de ionosfera - 2 capas ocultas - 34,9,4,1	Info	Conjunto de datos	RNGA + AGCC	RNA
	Mismo Alg glial toda la red. %Dec \geq %Inc	Entre...	83,38%	81,82%
		Valid...	77,18%	75,75%
		Test	78,42%	80,01%
	Mismo Alg glial toda la red	Entre...	83,81%	81,82%
		Valid...	76,24%	75,75%
		Test	78,40%	80,01%
	Mismo Alg glial por capas	Entre...	85,24%	81,82%
		Valid...	76,42%	75,75%
		Test	77,96%	80,01%
	Mismo Alg glial por capas. %Dec \geq %Inc	Entre...	85,58%	81,82%
		Valid...	77,01%	75,75%
		Test	78,66%	80,01%

A.1.3. Análisis de astrocitos artificiales en redes con 3 capas ocultas

Algoritmos de potenciación astrocítica

Setas venenosas

Como se puede observar en la siguiente tabla, se muestran los porcentajes de acierto medio alcanzados por la RNA y las RNGA con el problema de setas venenosas usando algoritmos de potenciación. Se puede observar que la RNGA con la combinación 3-6 es la única que supera a la RNA y lo hace en los aciertos sobre los datos de test.

Si se consulta la columna mejor se puede ver que los algoritmos de gliales que activaciones no consecutivas solo superan a la RNA en los aciertos de entrenamiento, mientras que los otros cinco algoritmos son capaces de superar a la RNA en entrenamiento, validación y test.

Tabla 30. Resultados en el problema de setas venenosas de la RNGA con el alg. de potenciación y la RNA, obtenidos en el entrenamiento, validación y test con la arquitectura de 3 capas ocultas. Los valores que se muestran corresponden al % de acierto medio.

	Alg glial	Conjunto de datos	RNGA					RNA
			2-4	2-6	3-6	3-8	Mejor	
Setas venenosas - 3 capas ocultas - 22,12,7,4,1	Efecto de atenuación astrocítica	Entre...	93,11%	93,60%	93,68%	93,44%	95,48%	94,11%
		Valid...	91,00%	91,49%	91,50%	91,40%	93,71%	93,51%
		Test	90,91%	91,45%	91,51%	91,34%	93,64%	93,38%
	Act. no consecutivas, modificación sin límite	Entre...	92,73%	92,60%	93,05%	92,71%	94,88%	94,11%
		Valid...	90,51%	90,55%	90,96%	90,63%	93,06%	93,51%
		Test	90,54%	90,49%	90,80%	90,58%	92,94%	93,38%
	Efecto de proceso global	Entre...	93,92%	93,44%	93,96%	93,40%	95,93%	94,11%
		Valid...	93,23%	92,59%	93,04%	91,86%	95,15%	93,51%
		Test	93,18%	92,69%	93,50%	92,73%	95,46%	93,38%
	Act. consecutivas, modificación con límite	Entre...	93,80%	93,55%	93,39%	93,76%	95,64%	94,11%
		Valid...	92,63%	92,26%	92,24%	92,51%	94,74%	93,51%
		Test	92,68%	92,19%	92,22%	92,55%	94,72%	93,38%
	Act. no consecutivas, modificación con límite	Entre...	92,90%	92,71%	92,61%	92,89%	94,77%	94,11%
		Valid...	90,80%	90,42%	90,54%	90,68%	93,14%	93,51%
		Test	90,65%	90,35%	90,32%	90,73%	92,99%	93,38%
	Act. consecutivas, modificación sin límite	Entre...	93,90%	93,55%	93,61%	93,94%	95,88%	94,11%
		Valid...	92,88%	92,21%	92,42%	92,78%	94,92%	93,51%
		Test	92,75%	92,00%	92,32%	92,77%	94,82%	93,38%

Cáncer de mama

Como se puede observar en la siguiente tabla, se muestran los porcentajes de acierto medio alcanzados por la RNA y las RNGA con el problema de cáncer de mama usando algoritmos de potenciación. Se puede observar que la RNGA funciona mejor que la RNA salvo con el algoritmo de efecto de proceso global, donde la RNGA en las combinaciones de activación 2-6 y 3-8 no mejora a la RNA en los aciertos de la partición de datos correspondiente a validación.

Tabla 31. Resultados en el problema cáncer de mama de la RGA con el alg. de potenciación y la RNA, obtenidos en el entrenamiento, validación y test con la arquitectura de 3 capas ocultas. Los valores que se muestran corresponden al % de acierto medio.

	Alg grial	Conjunto de datos	RNGA					RNA
			2-4	2-6	3-6	3-8	Mejor	
Cáncer de mama - 3 capas ocultas - 9,12,8,4,1	Efecto de atenuación astrocítica	Entre...	92,79%	92,71%	92,79%	92,39%	93,49%	89,92%
		Valid...	87,54%	87,06%	87,42%	85,70%	89,32%	84,94%
		Test	88,85%	87,38%	87,83%	86,58%	89,88%	85,60%
	Act. no consecutivas, modificación sin límite	Entre...	91,75%	92,30%	92,43%	92,81%	93,06%	89,92%
		Valid...	87,26%	86,85%	87,48%	87,81%	89,71%	84,94%
		Test	87,54%	88,18%	88,18%	88,30%	90,91%	85,60%
	Efecto de proceso global	Entre...	89,70%	90,16%	90,13%	90,96%	90,70%	89,92%
		Valid...	84,13%	84,14%	84,15%	83,16%	86,19%	84,94%
		Test	84,97%	85,96%	85,54%	87,39%	87,29%	85,60%
	Act. consecutivas, modificación con límite	Entre...	91,06%	91,78%	91,22%	91,28%	92,42%	89,92%
		Valid...	85,81%	86,18%	86,16%	86,19%	88,70%	84,94%
		Test	85,91%	87,10%	86,56%	86,64%	89,08%	85,60%
	Act. no consecutivas, modificación con límite	Entre...	92,53%	92,36%	92,11%	92,16%	92,90%	89,92%
		Valid...	87,48%	87,11%	87,35%	87,52%	89,86%	84,94%
		Test	88,57%	88,23%	88,17%	87,87%	90,34%	85,60%
	Act. consecutivas, modificación sin límite	Entre...	91,35%	91,97%	91,38%	91,29%	92,35%	89,92%
		Valid...	86,37%	86,59%	86,25%	85,97%	88,86%	84,94%
		Test	87,06%	86,84%	86,83%	85,98%	89,27%	85,60%

Señal de ionosfera

Como se puede observar en la siguiente tabla, se muestran los porcentajes de acierto medio alcanzados por la RNA y las RNGA con el problema de ionosfera usando algoritmos de potenciación. Como se puede observar en la tabla la RNGA lo hace mejor que la RNA en la fase de entrenamiento.

En la columna mejor, se puede ver que los algoritmos de activaciones no consecutivas la RNGA obtiene más aciertos en los datos de entrenamiento y validación que los obtenidos por la RNA, pero esta última tiene una mejor capacidad de generalización. La RNGA con el algoritmo de efecto de proceso global tiene más aciertos para las particiones de entrenamiento y test, pero la RNA tiene mejores

resultados para los datos de validación. Las RNGA con los otros tres algoritmos, el de atenuación y el de activaciones consecutivas, tienen mejores resultados que la RNA en entrenamiento, validación y test.

Tabla 32. Resultados en el problema señal de ionosfera de la RNGA con el alg. de potenciación y la RNA, obtenidos en el entrenamiento, validación y test con la arquitectura de 3 capas ocultas. Los valores que se muestran corresponden al % de acierto medio.

	Alg glial	Conjunto de datos	RNGA					RNA
			2-4	2-6	3-6	3-8	Mejor	
Señal de ionosfera - 3 capas ocultas - 34,9,6,3,1	Efecto de atenuación astrocítica	Entre...	87,49%	88,53%	87,45%	88,24%	89,72%	83,13%
		Valid...	75,80%	76,56%	76,49%	76,60%	80,87%	78,08%
		Test	76,54%	78,00%	76,85%	76,88%	80,99%	81,79%
	Act. no consecutivas, modificación sin límite	Entre...	87,27%	87,45%	87,33%	87,44%	88,84%	83,13%
		Valid...	75,83%	75,63%	75,99%	76,32%	80,17%	78,08%
		Test	76,37%	76,50%	76,78%	77,59%	81,14%	81,79%
	Efecto de proceso global	Entre...	81,78%	85,52%	83,40%	85,48%	84,26%	83,13%
		Valid...	70,77%	65,33%	66,69%	58,46%	75,55%	78,08%
		Test	81,02%	80,13%	81,57%	77,84%	81,84%	81,79%
	Act. consecutivas, modificación con límite	Entre...	85,02%	85,61%	84,95%	84,91%	86,78%	83,13%
		Valid...	77,71%	77,05%	76,98%	76,99%	81,22%	78,08%
		Test	80,39%	79,53%	79,45%	79,80%	83,99%	81,79%
	Act. no consecutivas, modificación con límite	Entre...	86,90%	87,43%	87,62%	86,95%	88,86%	83,13%
		Valid...	76,08%	75,91%	75,44%	75,56%	79,90%	78,08%
		Test	76,61%	75,89%	76,44%	74,72%	80,57%	81,79%
	Act. consecutivas, modificación sin límite	Entre...	85,05%	84,66%	84,44%	84,22%	86,34%	83,13%
		Valid...	77,03%	77,20%	77,28%	77,17%	81,50%	78,08%
		Test	79,92%	78,78%	79,56%	79,25%	83,18%	81,79%

Algoritmo de depresión astrocítica

Setas venenosas

Como se puede observar en la siguiente tabla, se muestran los porcentajes de acierto medio alcanzados por la RNA y las RNGA con el problema de setas venenosas usando el algoritmo depresión. En la tabla se puede observar que la mejor RNGA es la

de la combinación de activación-iteración 3-8 pero que no supera la RNA. Sin embargo, al consultar la columna mejor se puede ver que la RNGA supera los resultados obtenidos por la RNA.

Tabla 33. Resultados en el problema de setas venenosas de la RNGA con el alg. de depresión y la RNA, obtenidos en el entrenamiento, validación y test con la arquitectura de 3 capas ocultas. Los valores que se muestran corresponden al % de acierto medio.

Conjunto de datos	RNGA					RNA
	2-4	2-6	3-6	3-8	Mejor	
Entre...	92,62%	92,59%	92,74%	93,22%	95,07%	94,11%
Valid...	91,06%	90,86%	91,23%	91,77%	93,73%	93,51%
Test	91,05%	90,88%	91,15%	91,72%	93,73%	93,38%

Cáncer de mama

Como se puede observar en la siguiente tabla, se muestran los porcentajes de acierto medio alcanzados por la RNA y las RNGA con el problema de cáncer de mama usando el algoritmo depresión. Con esta arquitectura de tres capas la RNGA es capaz de superar los resultados obtenidos por la RNA en todas las situaciones probadas.

Tabla 34. Resultados en el problema de cáncer de mama de la RNGA con el alg. de depresión y la RNA, obtenidos en el entrenamiento, validación y test con la arquitectura de 3 capas ocultas. Los valores que se muestran corresponden al % de acierto medio.

Conjunto de datos	RNGA					RNA
	2-4	2-6	3-6	3-8	Mejor	
Entre...	92,68%	93,39%	93,16%	93,59%	93,78%	89,92%
Valid...	89,08%	89,36%	89,34%	89,49%	91,49%	84,94%
Test	90,25%	90,70%	91,18%	91,72%	92,82%	85,60%

Señal de ionosfera

Como se puede observar en la siguiente tabla, se muestran los porcentajes de acierto medio alcanzados por la RNA y las RNGA con el problema de ionosfera usando el algoritmo depresión. La RNGA es capaz de superar a la RNA en la fase de entrenamiento y al fijarse en la columna mejor la RNGA tiene mejores resultados en entrenamiento y validación, pero la RNA es capaz de obtener mejor capacidad de generalización.

Tabla 35. Resultados en el problema señal de ionosfera de la RNGA con el alg. de depresión y la RNA, obtenidos en el entrenamiento, validación y test con la arquitectura de 3 capas ocultas. Los valores que se muestran corresponden al % de acierto medio.

Conjunto de datos	RNGA					RNA
	2-4	2-6	3-6	3-8	Mejor	
Entre...	86,24%	86,17%	86,67%	86,82%	88,12%	83,13%
Valid...	76,04%	75,61%	76,58%	76,54%	81,06%	78,08%
Test	76,13%	75,40%	76,89%	76,81%	81,63%	81,79%

Algoritmo de receptores endocannabinoides astrocíticos

Setas venenosas

Como se puede observar en la siguiente tabla, se muestran los porcentajes de acierto medio alcanzados por la RNA y las RNGA con el problema de setas venenosas usando el algoritmo de receptores de endocannabinoides. Se puede observar que la RNGA con las combinaciones de activación-iteración 2-4 y 3-8 supera los aciertos de la RNA en entrenamiento, validación y test.

Tabla 36. Resultados en el problema de setas venenosas de la RNGA con el alg. de receptores de endocannabinoides y la RNA, obtenidos en el entrenamiento, validación y test con la arquitectura de 3 capas ocultas. Los valores que se muestran corresponden al % de acierto

Conjunto de datos	RNGA					RNA
	2-4	2-6	3-6	3-8	Mejor	
Entre...	94,65%	94,49%	93,97%	94,73%	96,56%	94,11%
Valid...	93,54%	93,29%	92,75%	93,77%	95,63%	93,51%
Test	93,48%	93,18%	92,67%	93,69%	95,59%	93,38%

Cáncer de mama

Como se puede observar en la siguiente tabla, se muestran los porcentajes de acierto medio alcanzados por la RNA y las RNGA con el problema de cáncer de mama

usando el algoritmo de receptores de endocannabinoides. Los resultados para la RNGA son muy satisfactorios ya que supera a la RNA en todas las casuísticas probadas.

Tabla 37. Resultados en el problema cáncer de mama de la RNGA con el alg. de receptores de endocannabinoides y la RNA, obtenidos en el entrenamiento, validación y test con la arquitectura de 3 capas ocultas. Los valores que se muestran corresponden al % de acierto medio.

Conjunto de datos	RNGA					RNA
	2-4	2-6	3-6	3-8	Mejor	
Entre...	91,54%	91,18%	91,55%	91,37%	92,49%	89,92%
Valid...	85,70%	85,59%	86,25%	86,13%	88,01%	84,94%
Test	85,77%	85,70%	86,28%	86,52%	87,38%	85,60%

Señal de ionosfera

Como se puede observar en la siguiente tabla, se muestran los porcentajes de acierto medio alcanzados por la RNA y las RNGA con el problema de ionosfera usando el algoritmo de receptores de endocannabinoides. La RNGA es capaz de superar a la RNA en la fase de entrenamiento y es necesario acudir a la columna mejor donde la RNGA obtiene mejores resultados en todas las particiones de los datos.

Tabla 38. Resultados en el problema señal de ionosfera de la RNGA con el alg. de receptores de endocannabinoides y la RNA, obtenidos en el entrenamiento, validación y test con la arquitectura de 3 capas ocultas. Los valores que se muestran corresponden al % de acierto medio.

Conjunto de datos	RNGA					RNA
	2-4	2-6	3-6	3-8	Mejor	
Entre...	86,86%	85,86%	87,47%	86,28%	88,33%	83,13%
Valid...	77,29%	76,56%	77,53%	76,48%	81,73%	78,08%
Test	78,99%	76,71%	79,31%	77,87%	82,10%	81,79%

Búsqueda de algoritmo astrocítico usando computación coevolutiva

Setas venenosas

Como se puede observar en la siguiente tabla, se muestran los porcentajes de acierto medio alcanzados por la RNA y las RNGA empleando un AGCC para el problema de setas venenosas. Con redes que tienen tres capas ocultas, la RNA supera a la RNGA.

Lo que se puede ver es que una vez más es que la RNGA funciona mejor cuando cada capa usa un algoritmo glial distinto.

Tabla 39. Resultados en el problema de setas venenosas de la RNGA creada con AGCC y la RNA, obtenidos en el entrenamiento, validación y test con la arquitectura de 3 capas ocultas. Los valores que se muestran corresponden al % de acierto medio.

Setas venenosas - 3 capas ocultas - 22,12,7,4,1	Info	Conjunto de datos	RNGA + AGCC	RNA	
	Mismo Alg glial toda la red. %Dec ≥ %Inc	Entre...		93,84%	94,11%
		Valid...		93,05%	93,51%
		Test		93,17%	93,38%
	Mismo Alg glial toda la red	Entre...		93,44%	94,11%
		Valid...		92,45%	93,51%
		Test		92,81%	93,38%
	Mismo Alg glial por capas	Entre...		93,93%	94,11%
		Valid...		93,20%	93,51%
		Test		93,13%	93,38%
	Mismo Alg glial por capas. %Dec ≥ %Inc	Entre...		93,30%	94,11%
		Valid...		92,46%	93,51%
		Test		92,41%	93,38%

Cáncer de mama

Como se puede observar en la siguiente tabla, se muestran los porcentajes de acierto medio alcanzados por la RNA y las RNGA empleando un AGCC para el problema de cáncer de mama. Se puede observar que el uso de una arquitectura compleja con tres capas ocultas beneficia notablemente a la RNGA ya que consigue resultados mucho mejores que la RNA. Además, se puede observar que el uso de un algoritmo glial distinto para cada capa beneficia a los resultados que puede obtener la RNGA.

Tabla 40. Resultados en el problema cáncer de mama de la RNGA creada con AGCC y la RNA, obtenidos en el entrenamiento, validación y test con la arquitectura de 3 capas ocultas. Los valores que se muestran corresponden al % de acierto medio.

Cáncer de mama - 3 capas ocultas - 9,12,8,4,1	Info	Conjunto de datos	RNGA + AGCC	RNA
	Mismo Alg glial toda la red. %Dec ≥ %Inc	Entre...	91,13%	89,92%
		Valid...	86,80%	84,94%
		Test	87,10%	85,60%
	Mismo Alg glial toda la red	Entre...	90,83%	89,92%
		Valid...	86,23%	84,94%
		Test	86,55%	85,60%
	Mismo Alg glial por capas	Entre...	92,88%	89,92%
		Valid...	88,84%	84,94%
		Test	89,33%	85,60%
	Mismo Alg glial por capas. %Dec ≥ %Inc	Entre...	92,48%	89,92%
		Valid...	88,59%	84,94%
		Test	89,15%	85,60%

Señal de ionosfera

Como se puede observar en la siguiente tabla, se muestran los porcentajes de acierto medio alcanzados por la RNA y las RNGA empleando un AGCC para el problema de señal de ionosfera. En este problema tan complejo, la RNGA supera a la RNA en la fase de entrenamiento, pero la RNA es capaz de crear un modelo más genérico que el conseguido por la RNGA.

Un hecho interesante es que la RNGA con la restricción de que el porcentaje de incremento no sea mayor que el porcentaje de decremento y el poder establecer un algoritmo glial específico por capa la RNGA se beneficia y obtiene mejores resultados, lo que hace que supere a la RNA en los aciertos obtenidos en la fase de validación.

Tabla 41. Resultados en el problema señal de ionosfera de la RNGA creada con AGCC y la RNA, obtenidos en el entrenamiento, validación y test con la arquitectura de 3 capas ocultas. Los valores que se muestran corresponden al % de acierto medio.

Señal de ionosfera - 3 capas ocultas - 34,9,6,3,1	Info	Conjunto de datos	RNGA + AGCC	RNA
	Mismo Alg glial toda la red. %Dec ≥ %Inc	Entre...	84,49%	83,13%
		Valid...	77,85%	78,08%
		Test	79,76%	81,79%
	Mismo Alg glial toda la red	Entre...	84,74%	83,13%
		Valid...	76,93%	78,08%
		Test	78,31%	81,79%
	Mismo Alg glial por capas	Entre...	86,84%	83,13%
		Valid...	77,98%	78,08%
		Test	79,35%	81,79%
	Mismo Alg glial por capas. %Dec ≥ %Inc	Entre...	88,34%	83,13%
		Valid...	78,84%	78,08%
		Test	79,69%	81,79%

A.2. Experimentación y resultados en problemas de regresión

A.2.1. Análisis de astrocitos artificiales en redes con 1 capa oculta

Algoritmos de potenciación astrocítica

Asentamiento de hormigón

Como se puede observar en la siguiente tabla, se muestra la media de los ECM alcanzados por la RNA y la RNGA con el problema de asentamiento de hormigón usando algoritmos de potenciación. Como se puede ver, tanto los algoritmos de activaciones no consecutivas como los de activaciones consecutivas todas las combinaciones de activación-iteración de la RNGA supera a la RNA. El caso particular de la RNGA con el algoritmo de efecto de atenuación astrocítica comete un mayor error de test en las combinaciones 2-6 y 3-8 de activación-iteración. Como peculiaridad, la red que emplea el algoritmo glial de efecto de proceso global puede obtener menores errores en la fase de entrenamiento y validación, pero la RNA tiene un mejor desempeño en la fase de validación. También es importante mencionar que

en este problema cuatro de los seis algoritmos tienen su mejor rendimiento usando la combinación de activación-iteración 2-6.

Tabla 42. Resultados en el problema asentamiento de hormigón de la RNGA con el alg. de potenciación y la RNA, obtenidos en el entrenamiento, validación y test con la arquitectura de 1 capa oculta. Los valores que se muestran corresponden a la media del ECM.

	Alg glial	Conjunto de datos	RNGA				Mejor	RNA
			2-4	2-6	3-6	3-8		
Asentamiento de hormigón - 1 capa oculta - 7,4,1	Efecto de atenuación astrocítica	Entre...	0,0197	0,0140	0,0154	0,0118	0,0140	0,0598
		Valid...	0,0681	0,0691	0,0666	0,0671	0,0561	0,0797
		Test	0,0771	0,0864	0,0834	0,0948	0,0814	0,0846
	Act. no consecutivas, modificación sin límite	Entre...	0,0310	0,0246	0,0317	0,0316	0,0274	0,0598
		Valid...	0,0695	0,0664	0,0694	0,0711	0,0622	0,0797
		Test	0,0739	0,0728	0,0735	0,0739	0,0714	0,0846
	Efecto de proceso global	Entre...	0,0400	0,0214	0,0220	0,0139	0,0385	0,0598
		Valid...	0,1041	0,1587	0,1547	0,2236	0,1025	0,0797
		Test	0,0763	0,0766	0,0766	0,0889	0,0760	0,0846
	Act. consecutivas, modificación con límite	Entre...	0,0428	0,0358	0,0427	0,0425	0,0378	0,0598
		Valid...	0,0721	0,0701	0,0726	0,0722	0,0671	0,0797
		Test	0,0754	0,0738	0,0753	0,0747	0,0741	0,0846
	Act. no consecutivas, modificación con límite	Entre...	0,0323	0,0235	0,0323	0,0324	0,0269	0,0598
		Valid...	0,0708	0,0667	0,0698	0,0717	0,0623	0,0797
		Test	0,0720	0,0720	0,0720	0,0727	0,0699	0,0846
	Act. consecutivas, modificación sin límite	Entre...	0,0429	0,0355	0,0424	0,0429	0,0382	0,0598
		Valid...	0,0719	0,0704	0,0723	0,0727	0,0667	0,0797
		Test	0,0746	0,0733	0,0740	0,0744	0,0725	0,0846

Fuegos forestales

Como se puede observar en la siguiente tabla, se muestra la media de los ECM alcanzados por la RNA y la RNGA con el problema de fuegos forestales usando algoritmos de potenciación. Se puede ver que la RNA siempre obtiene mejores resultados en la fase de validación, pero la RNGA, a excepción de la usada con el algoritmo de efecto de atenuación astrocítica, comete menores errores al predecir los valores con las particiones de datos para entrenamiento y test.

Tabla 43. Resultados en el problema fuegos forestales de la RNGA con el alg. de potenciación y la RNA, obtenidos en el entrenamiento, validación y test con la arquitectura de 1 capa oculta. Los valores que se muestran corresponden a la media del ECM.

	Alg glial	Conjunto de datos	RNGA					RNA
			2-4	2-6	3-6	3-8	Mejor	
Incendios Forestales - 1 capa oculta - 12,8,1	Efecto de atenuación astrocítica	Entre...	0,2961	0,2861	0,2867	0,2817	0,2870	0,3395
		Valid...	0,3678	0,3694	0,3682	0,3696	0,3608	0,3643
		Test	0,3952	0,4122	0,4089	0,4217	0,4088	0,3831
	Act. no consecutivas, modificación sin límite	Entre...	0,3123	0,3072	0,3136	0,3117	0,3125	0,3395
		Valid...	0,3670	0,3678	0,3664	0,3667	0,3619	0,3643
		Test	0,3825	0,3849	0,3813	0,3799	0,3793	0,3831
	Efecto de proceso global	Entre...	0,3395	0,3378	0,3376	0,3326	0,3379	0,3395
		Valid...	0,3649	0,3699	0,3696	0,3755	0,3627	0,3643
		Test	0,3832	0,3805	0,3807	0,3787	0,3816	0,3831
	Act. consecutivas, modificación con límite	Entre...	0,3236	0,3133	0,3235	0,3239	0,3212	0,3395
		Valid...	0,3667	0,3678	0,3666	0,3665	0,3631	0,3643
		Test	0,3816	0,3851	0,3817	0,3822	0,3824	0,3831
	Act. no consecutivas, modificación con límite	Entre...	0,3132	0,3055	0,3142	0,3139	0,3117	0,3395
		Valid...	0,3663	0,3677	0,3670	0,3668	0,3617	0,3643
		Test	0,3801	0,3866	0,3795	0,3809	0,3803	0,3831
	Act. consecutivas, modificación sin límite	Entre...	0,3238	0,3141	0,3239	0,3239	0,3208	0,3395
		Valid...	0,3668	0,3681	0,3676	0,3676	0,3633	0,3643
		Test	0,3820	0,3845	0,3818	0,3816	0,3829	0,3831

Rendimiento estudiantil

Como se puede observar en la siguiente tabla, se muestra la media de los ECM alcanzados por la RNA y la RNGA con el problema de rendimiento estudiantil usando algoritmos de potenciación. Los valores muestran un gran trabajo por parte de la RNGA usando los algoritmos de activaciones consecutivas, donde obtiene mejores valores que la RNA en entrenamiento, validación y test. Los algoritmos de activaciones no consecutivas y efecto de atenuación astrocítica solamente hacen que la RNGA supere a la RNA en la fase de entrenamiento mientras que el algoritmo de efecto de proceso global no logra superar en ningún caso a la RNA. Además, los casos en los que

las combinaciones de activación-iteración 2-4, 2-6, 3-6 y 3-8 no superan la RNA la columna mejor también indica que la RNGA tampoco puede superar los resultados añadiendo dentro del *cross-validation* las distintas combinaciones de activación-iteración.

Tabla 44. Resultados en el problema rendimiento estudiantil de la RNGA con el alg. de potenciación y la RNA, obtenidos en el entrenamiento, validación y test con la arquitectura de 1 capa oculta. Los valores que se muestran corresponden a la media del ECM.

	Alg glial	Conjunto de datos	RNGA					RNA
			2-4	2-6	3-6	3-8	Mejor	
Rendimiento estudiantil - 1 capa oculta - 30,9,1	Efecto de atenuación astrocítica	Entre...	0,0165	0,0171	0,0167	0,0170	0,0168	0,0175
		Valid...	0,0211	0,0217	0,0217	0,0219	0,0201	0,0195
		Test	0,0253	0,0268	0,0266	0,0275	0,0255	0,0221
	Act. no consecutivas, modificación sin límite	Entre...	0,0163	0,0165	0,0162	0,0162	0,0161	0,0175
		Valid...	0,0208	0,0213	0,0207	0,0205	0,0196	0,0195
		Test	0,0245	0,0247	0,0244	0,0241	0,0237	0,0221
	Efecto de proceso global	Entre...	0,0180	0,0181	0,0180	0,0179	0,0180	0,0175
		Valid...	0,0211	0,0246	0,0238	0,0276	0,0200	0,0195
		Test	0,0274	0,0470	0,0403	0,0546	0,0270	0,0221
	Act. consecutivas, modificación con límite	Entre...	0,0168	0,0163	0,0168	0,0168	0,0165	0,0175
		Valid...	0,0194	0,0192	0,0193	0,0193	0,0188	0,0195
		Test	0,0220	0,0220	0,0219	0,0220	0,0219	0,0221
	Act. no consecutivas, modificación con límite	Entre...	0,0162	0,0165	0,0162	0,0162	0,0162	0,0175
		Valid...	0,0205	0,0213	0,0208	0,0205	0,0196	0,0195
		Test	0,0246	0,0255	0,0241	0,0241	0,0239	0,0221
	Act. consecutivas, modificación sin límite	Entre...	0,0168	0,0164	0,0167	0,0168	0,0166	0,0175
		Valid...	0,0193	0,0193	0,0194	0,0193	0,0189	0,0195
		Test	0,0220	0,0220	0,0219	0,0219	0,0219	0,0221

Algoritmo de depresión astrocítica

Asentamiento de hormigón

Como se puede observar en la siguiente tabla, se muestra la media de los ECM alcanzados por la RNA y la RNGA con el problema de asentamiento de hormigón usando el algoritmo de depresión. Se puede ver que la RNA tiene un poco de *overfitting* teniendo un error un incremento de un 33% en el error de validación y un

40% en test si se compara con el que obtiene en entrenamiento. El overfitting también lo tiene la RNGA y aunque es más acusado, ya que el incremento en validación y test está entorno al 60% del error de entrenamiento, la RNGA obtiene mejoras increíblemente significativas en los ECM cometidos.

Tabla 45. Resultados en el problema asentamiento de hormigón de la RNGA con el alg. de depresión y la RNA, obtenidos en el entrenamiento, validación y test con la arquitectura de 1 capa oculta. Los valores que se muestran corresponden a la media del ECM.

Conjunto de datos	RNGA					RNA
	2-4	2-6	3-6	3-8	Mejor	
Entre...	0,0484	0,0409	0,0435	0,0392	0,0395	0,0598
Valid...	0,0718	0,0701	0,0691	0,0696	0,0626	0,0797
Test	0,0730	0,0715	0,0708	0,0696	0,0659	0,0846

Fuegos forestales

Como se puede observar en la siguiente tabla, se muestra la media de los ECM alcanzados por la RNA y la RNGA con el problema de fuegos forestales usando el algoritmo de depresión. En la tabla se observa que la RNGA obtiene siempre mejores resultados que la RNA a excepción del ECM en la fase de validación en la simulación 2-4 el cual no es significativamente peor que la RNA ya que obtiene un p-valor de 0,56 al realizar el test paramétrico de wilcoxon.

Tabla 46. Resultados en el problema fuegos forestales de la RNGA con el alg. de depresión y la RNA, obtenidos en el entrenamiento, validación y test con la arquitectura de 1 capa oculta. Los valores que se muestran corresponden a la media del ECM.

Conjunto de datos	RNGA					RNA
	2-4	2-6	3-6	3-8	Mejor	
Entre...	0,3310	0,3252	0,3272	0,3209	0,3252	0,3395
Valid...	0,3647	0,3640	0,3641	0,3634	0,3598	0,3643
Test	0,3763	0,3749	0,3745	0,3750	0,3749	0,3831

Rendimiento estudiantil

Como se puede observar en la siguiente tabla, se muestra la media de los ECM alcanzados por la RNA y la RNGA con el problema de rendimiento estudiantil usando el algoritmo de depresión. Como muestra la tabla, la RNGA de una capa oculta con el

algoritmo de depresión solamente es capaz de superar a la RNA en la fase de entrenamiento.

Tabla 47. Resultados en el problema rendimiento estudiantil de la RNGA con el alg. de depresión y la RNA, obtenidos en el entrenamiento, validación y test con la arquitectura de 1 capa oculta. Los valores que se muestran corresponden a la media del ECM.

Conjunto de datos	RNGA					RNA
	2-4	2-6	3-6	3-8	Mejor	
Entre...	0,0174	0,0174	0,0174	0,0175	0,0171	0,0175
Valid...	0,0208	0,0209	0,0210	0,0209	0,0198	0,0195
Test	0,0232	0,0235	0,0233	0,0236	0,0232	0,0221

Algoritmo de receptores endocannabinoides astrocíticos

Asentamiento de hormigón

Como se puede observar en la siguiente tabla, se muestra la media de los ECM alcanzados por la RNA y la RNGA con el problema de asentamiento de hormigón usando el algoritmo de receptores de endocannabinoides. La gran actividad que genera este algoritmo glial dentro de la red hace que el ECM en entrenamiento sea mucho más bajo que la RNA (0,059 frente al 0,018 que consigue la RNGA), donde también es capaz de llegar tener un buen desempeño en la fase de validación. Sin embargo, la RNGA comete un gran *overfitting* (en la combinación 3-8 el error de test es 350% más grande que el de entrenamiento), lo que hace que la RNA obtenga un modelo que generaliza mejor.

Tabla 48. Resultados en el problema asentamiento de hormigón de la RNGA con el alg. de receptores de endocannabinoides y la RNA, obtenidos en el entrenamiento, validación y test con la arquitectura de 1 capa oculta. Los valores que se muestran corresponden a la media del ECM.

Conjunto de datos	RNGA					RNA
	2-4	2-6	3-6	3-8	Mejor	
Entre...	0,0370	0,0312	0,0279	0,0184	0,0222	0,0598
Valid...	0,0799	0,0826	0,0795	0,0782	0,0625	0,0797
Test	0,1011	0,1112	0,1081	0,1207	0,1004	0,0846

Fuegos forestales

Como se puede observar en la siguiente tabla, se muestra la media de los ECM alcanzados por la RNA y la RNGA con el problema de fuegos forestales usando el algoritmo de receptores de endocannabinoides. La gran actividad que genera este algoritmo glial dentro de la red hace que el ECM en entrenamiento sea más bajo que el obtenido con RNA, y que usando la combinación de activación-iteración 3-8 la RNA obtiene peor ECM en validación. Sin embargo, este algoritmo no logra que la RNGA obtenga un modelo que generalice mejor que la RNA.

Tabla 49. Resultados en el problema fuegos forestales de la RNGA con el alg. de receptores de endocannabinoides y la RNA, obtenidos en el entrenamiento, validación y test con la arquitectura de 1 capa oculta. Los valores que se muestran corresponden a la media del ECM.

Conjunto de datos	RNGA					RNA
	2-4	2-6	3-6	3-8	Mejor	
Entre...	0,2918	0,3111	0,2937	0,2901	0,2960	0,3395
Valid...	0,3648	0,3653	0,3658	0,3631	0,3586	0,3643
Test	0,40582	0,3961	0,4087	0,4127	0,4016	0,3831

Rendimiento estudiantil

Como se puede observar en la siguiente tabla, se muestra la media de los ECM alcanzados por la RNA y la RNGA con el problema de rendimiento estudiantil usando el algoritmo de receptores de endocannabinoides. La gran actividad que genera este algoritmo glial dentro de la red hace que el ECM en entrenamiento sea más bajo que el obtenido con RNA. Sin embargo, este algoritmo no logra que la RNGA obtenga un modelo que sea mejor que la RNA en la fase de validación y test.

Tabla 50. Resultados en el problema rendimiento estudiantil de la RNGA con el alg. de receptores de endocannabinoides y la RNA, obtenidos en el entrenamiento, validación y test con la arquitectura de 1 capa oculta. Los valores que se muestran corresponden a la media del ECM.

Conjunto de datos	RNGA					RNA
	2-4	2-6	3-6	3-8	Mejor	
Entre...	0,0170	0,0209	0,0173	0,0182	0,0170	0,0175
Valid...	0,0216	0,0234	0,0216	0,0220	0,0204	0,0195
Test	0,0260	0,0277	0,0263	0,0286	0,0261	0,0221

Búsqueda de algoritmo astrocítico usando computación coevolutiva

Asentamiento de hormigón

Como se puede observar en la siguiente tabla, se muestran los porcentajes de acierto medio alcanzados por la RNA y las RNGA empleando un AGCC para el problema de asentamiento de hormigón. En la tabla se muestra que en todas las casuísticas la RNGA supera a la RNA en las fases de entrenamiento y validación. La mejor ejecución es donde el AGCC tiene la restricción de usar una única configuración del algoritmo glial dentro de la red y que el astrocito tenga un porcentaje de incremento que no supere al de decremento ya que es la única simulación donde la RNGA genera un modelo que generaliza mejor que la RNA. Usar una configuración diferente de algoritmos por cada capa de la RNGA hace que la RNGA tenga mejores resultados en la fase de entrenamiento y un desempeño similar en la fase de validación, pero como no es capaz de mejorar el ECM tiene un *overfitting* en los datos más acusado.

Tabla 51. Resultados en el problema asentamiento de hormigón de la RNGA creada con AGCC y la RNA, obtenidos en el entrenamiento, validación y test con la arquitectura de 1 capa oculta. Los valores que se muestran corresponden a la media del ECM.

Asentamiento de hormigón - 1 capa oculta - 7,4,1	Info	Conjunto de datos	RNGA + AGCC	RNA	
	Mismo Alg glial toda la red. %Dec ≥ %Inc	Entre...		0,0204	0,0598
		Valid...		0,0744	0,0797
		Test		0,0841	0,0846
	Mismo Alg glial toda la red	Entre...		0,0175	0,0598
		Valid...		0,0740	0,0797
		Test		0,0913	0,0846
	Mismo Alg glial por capas	Entre...		0,0149	0,0598
		Valid...		0,0761	0,0797
		Test		0,0932	0,0846
	Mismo Alg glial por capas. %Dec ≥ %Inc	Entre...		0,0170	0,0598
		Valid...		0,0763	0,0797
		Test		0,0907	0,0846

Fuegos forestales

Como se puede observar en la siguiente tabla, se muestran los porcentajes de acierto medio alcanzados por la RNA y las RNGA empleando un AGCC para el problema de fuegos forestales. En todas las simulaciones realizadas la RNGA tiene un mejor

desempeño que la RNGA en la fase de entrenamiento, y en la mitad de las simulaciones la RNGA también mejora a la RNA en la etapa de validación. Sin embargo, en ninguno de los casos fue capaz de superar a la RNA en la etapa de test.

Tabla 52. Resultados en el problema fuegos forestales de la RNGA creada con AGCC y la RNA, obtenidos en el entrenamiento, validación y test con la arquitectura de 1 capa oculta. Los valores que se muestran corresponden a la media del ECM.

Incendios Forestales - 1 capa oculta - 12,8,1	Info	Conjunto de datos	RNGA + AGCC	RNA
	Mismo Algorítmica toda la red. %Dec ≥ %Inc	Entre...	0,3088	0,3395
		Valid...	0,3640	0,3643
		Test	0,3980	0,3831
	Mismo Algorítmica toda la red	Entre...	0,3073	0,3395
		Valid...	0,3647	0,3643
		Test	0,3975	0,3831
	Mismo Algorítmica por capas	Entre...	0,3029	0,3395
		Valid...	0,3641	0,3643
		Test	0,4021	0,3831
	Mismo Algorítmica por capas. %Dec ≥ %Inc	Entre...	0,3011	0,3395
		Valid...	0,3657	0,3643
Test		0,4068	0,3831	

Rendimiento estudiantil

Como se puede observar en la siguiente tabla, se muestran los porcentajes de acierto medio alcanzados por la RNA y las RNGA empleando un AGCC para el problema de rendimiento estudiantil. En tres de las simulaciones realizadas la RNA tiene un mejor funcionamiento que la RNGA. Al fijarse en la simulación donde el AGCC tiene la restricción de los porcentajes, pero no tiene impedimento para establecer una configuración específica en cada capa de la RNGA esta obtiene un mejor error de entrenamiento que la RNA. Fijándose en las filas del ECM de entrenamiento se ve que los valores de RNGA y RNA son muy similares, lo que no sucede con validación y test, lo que indica que la RNGA tiene problemas de *overfitting*.

Tabla 53. Resultados en el problema rendimiento estudiantil de la RNGA creada con AGCC y la RNA, obtenidos en el entrenamiento, validación y test con la arquitectura de 1 capa oculta. Los valores que se muestran corresponden a la media del ECM.

Rendimiento estudiantil - 1 capa oculta - 30,9,1	Info	Conjunto de datos	RNGA + AGCC	RNA
	Mismo Alg glial toda la red. %Dec \geq %Inc	Entre...	0,0176	0,0175
		Valid...	0,0203	0,0195
		Test	0,0287	0,0221
	Mismo Alg glial toda la red	Entre...	0,0178	0,0175
		Valid...	0,0203	0,0195
		Test	0,0309	0,0221
	Mismo Alg glial por capas	Entre...	0,0176	0,0175
		Valid...	0,0208	0,0195
		Test	0,0332	0,0221
	Mismo Alg glial por capas. %Dec \geq %Inc	Entre...	0,0175	0,0175
		Valid...	0,0204	0,0195
		Test	0,0393	0,0221

A.2.2. Análisis de astrocitos artificiales en redes con 2 capas ocultas

Algoritmos de potenciación astrocítica

Asentamiento de hormigón

Como se puede observar en la siguiente tabla, se muestra la media de los ECM alcanzados por la RNA y la RNGA con el problema de asentamiento de hormigón usando algoritmos de potenciación. En la tabla se puede ver el desempeño que tiene la que los algoritmos gliales de activaciones consecutivas aportan a la RNGA ya que supera en todos los casos y de manera significativa los resultados obtenidos por la RNA. Los algoritmos de la RNGA de activaciones no consecutivas también tienen un buen desempeño y logran superar a la RNA, aunque solo en algunos casos específicos algunas combinaciones de activación-iteración de la RNGA con estos algoritmos no tiene el mismo rendimiento que la RNA en la fase de test. En cuanto a las RNGA con el algoritmo de efecto de atenuación astrocítica las fases de entrenamiento y validación superan a la RNA, de hecho, en la fase de entrenamiento consigue una mejora increíblemente significativa (se pasa de un ECM del 0,05 al 0,01) lo que provoca una

especialización en los datos de entrenamiento ya que en la fase de test no logra acercarse a los valores obtenidos por la RNA.

Tabla 54. Resultados en el problema asentamiento de hormigón de la RNGA con el alg. de potenciación y la RNA, obtenidos en el entrenamiento, validación y test con la arquitectura de 2 capas ocultas. Los valores que se muestran corresponden a la media del ECM.

	Alg glial	Conjunto de datos	RNGA					RNA
			2-4	2-6	3-6	3-8	Mejor	
Asentamiento de hormigón - 2 capas ocultas - 9,7,3,1	Efecto de atenuación astrocítica	Entre...	0,0171	0,0124	0,0137	0,0114	0,0122	0,0507
		Valid...	0,0701	0,0734	0,0730	0,0718	0,0575	0,0750
		Test	0,0870	0,0995	0,0929	0,1070	0,0860	0,0764
	Act. no consecutivas, modificación sin límite	Entre...	0,0252	0,0215	0,0246	0,0249	0,0219	0,0507
		Valid...	0,0696	0,0696	0,0685	0,0678	0,0587	0,0750
		Test	0,0781	0,0835	0,0784	0,0764	0,0734	0,0764
	Efecto de proceso global	Entre...	0,0312	0,0197	0,0199	0,0135	0,0297	0,0507
		Valid...	0,1167	0,1714	0,1755	0,2236	0,1135	0,0750
		Test	0,0712	0,0822	0,0777	0,0944	0,0712	0,0764
	Act. consecutivas, modificación con límite	Entre...	0,0343	0,0266	0,0330	0,0343	0,0271	0,0507
		Valid...	0,0677	0,0651	0,0668	0,0667	0,0596	0,0750
		Test	0,0709	0,0715	0,0702	0,0694	0,0666	0,0764
	Act. no consecutivas, modificación con límite	Entre...	0,0250	0,0207	0,0256	0,0234	0,0218	0,0507
		Valid...	0,0681	0,0715	0,0706	0,0692	0,0578	0,0750
		Test	0,0752	0,0839	0,0782	0,0756	0,0705	0,0764
	Act. consecutivas, modificación sin límite	Entre...	0,0333	0,0262	0,0329	0,0337	0,0282	0,0507
		Valid...	0,0665	0,0672	0,0669	0,0670	0,0606	0,0750
		Test	0,0696	0,0726	0,0696	0,0704	0,0676	0,0764

Fuegos forestales

Como se puede observar en la siguiente tabla, se muestra la media de los ECM alcanzados por la RNA y la RNGA con el problema de fuegos forestales usando algoritmos de potenciación. Se puede observar fácilmente que la RNGA cuyo algoritmo glial es el de activaciones consecutivas con modificación sin límite en los pesos supera en todo y en todas las casuísticas a la RNA. La RNGA con el otro algoritmo de activaciones no consecutivas también tiene un buen desempeño solo en una de las

cuatro combinaciones probadas de activación-iteración supera a la RNA en fase de test.

La RNGA con los algoritmos de activaciones no consecutivas tienen también un gran desempeño. De hecho, la RNA solo es capaz de superar a la RNGA en la fase de validación y únicamente lo hace en casos específicos.

Los otros dos algoritmos gliales, el de efecto de atenuación y el de efecto de proceso global no logran un beneficio tan notable en la RNGA como los otros algoritmos de potenciación. Aun así, son dignos de observar ya que el primero, el de atenuación, logra una especialización de la red muy buena que logra superar siempre a la RNA en la fase de entrenamiento para este problema tan complejo, pero lo hace a costa de perder generalización mientras que el otro caso, el de efecto de proceso global la RNGA no tiene mejores resultados en la fase de entrenamiento y validación pero logra generalizar el modelo y tener un buen desempeño ante datos nuevos, lo que hace que supere siempre a la RNA en la fase de test.

Tabla 55. Resultados en el problema fuegos forestales de la RNGA con el alg. de potenciación y la RNA, obtenidos en el entrenamiento, validación y test con la arquitectura de 2 capas ocultas. Los valores que se muestran corresponden a la media del ECM.

	Alg glial	Conjunto de datos	RNGA				RNA	
			2-4	2-6	3-6	3-8		Mejor
Incendios Forestales - 2 capas ocultas - 12,8,5,1	Efecto de atenuación astrocítica	Entre...	0,2956	0,2880	0,2911	0,2887	0,2905	0,3317
		Valid...	0,3685	0,3680	0,3685	0,3719	0,3609	0,3657
		Test	0,4036	0,4208	0,4113	0,4279	0,4137	0,3905
	Act. no consecutivas, modificación sin límite	Entre...	0,3121	0,3048	0,3122	0,3131	0,3106	0,3317
		Valid...	0,3649	0,3685	0,3652	0,3647	0,3606	0,3657
		Test	0,3877	0,3902	0,3864	0,3841	0,3875	0,3905
	Efecto de proceso global	Entre...	0,3357	0,3375	0,3370	0,3338	0,3362	0,3317
		Valid...	0,3679	0,3700	0,3686	0,3786	0,3626	0,3657
		Test	0,3897	0,3820	0,3819	0,3770	0,3834	0,3905
	Act. consecutivas, modificación con límite	Entre...	0,3322	0,3286	0,3318	0,3325	0,3304	0,3317
		Valid...	0,3631	0,3630	0,3639	0,3627	0,3596	0,3657
		Test	0,3771	0,3787	0,3779	0,3759	0,3753	0,3905
Act. no consecutivas, modificación con límite	Entre...	0,3114	0,3056	0,3128	0,3119	0,3101	0,3317	
	Valid...	0,3653	0,3686	0,3657	0,3651	0,3607	0,3657	
	Test	0,3873	0,3894	0,3839	0,3859	0,3831	0,3905	
Act. consecutivas,	Entre...	0,3313	0,3260	0,3286	0,3306	0,3284	0,3317	

	modificación sin límite	Valid...	0,3629	0,3630	0,3632	0,3637	0,3598	0,3657
		Test	0,3778	0,3782	0,3782	0,3778	0,3767	0,3905

Rendimiento estudiantil

Como se puede observar en la siguiente tabla, se muestra la media de los ECM alcanzados por la RNA y la RNGA con el problema de rendimiento estudiantil usando algoritmos de potenciación. Se puede observar que en las todas las combinaciones de activación-iteración 2-4, 2-6, 3-6 y 3-8 las RNGA no logran mejorar en ningún momento los resultados obtenidos por la RNA. Es necesario acudir a la columna mejor para ver que la RNGA únicamente con tres algoritmos y solo en mejora el resultado de la RNA solo en la fase de validación.

Tabla 56. Resultados en el problema rendimiento estudiantil de la RNGA con el alg. de potenciación y la RNA, obtenidos en el entrenamiento, validación y test con la arquitectura de 2 capas ocultas. Los valores que se muestran corresponden a la media del ECM.

	Alg glial	Conjunto de datos	RNGA					RNA
			2-4	2-6	3-6	3-8	Mejor	
Rendimiento estudiantil - 2 capas ocultas - 30,9,4,1	Efecto de atenuación astrocítica	Entre...	0,0183	0,0185	0,0182	0,0187	0,0181	0,0167
		Valid...	0,0216	0,0221	0,0219	0,0221	0,0205	0,0196
		Test	0,0247	0,0260	0,0260	0,0260	0,0252	0,0221
	Act. no consecutivas, modificación sin límite	Entre...	0,0177	0,0183	0,0177	0,0177	0,0175	0,0167
		Valid...	0,0212	0,0219	0,0214	0,0214	0,0201	0,0196
		Test	0,0241	0,0251	0,0246	0,0247	0,0241	0,0221
	Efecto de proceso global	Entre...	0,0173	0,0180	0,0181	0,0181	0,0177	0,0167
		Valid...	0,0208	0,0217	0,0225	0,0235	0,0196	0,0196
		Test	0,0265	0,0350	0,0369	0,0547	0,0265	0,0221
	Act. consecutivas, modificación con límite	Entre...	0,0176	0,0176	0,0177	0,0176	0,0174	0,0167
		Valid...	0,0201	0,0203	0,0200	0,0200	0,0193	0,0196
		Test	0,0227	0,0230	0,0227	0,0227	0,0224	0,0221
	Act. no consecutivas, modificación con límite	Entre...	0,0175	0,0183	0,0178	0,0177	0,0175	0,0167
		Valid...	0,0213	0,0218	0,0215	0,0214	0,0202	0,0196
		Test	0,0244	0,0254	0,0247	0,0244	0,0242	0,0221
	Act. consecutivas, modificación sin límite	Entre...	0,0176	0,0174	0,0176	0,0176	0,0174	0,0167
		Valid...	0,0202	0,0203	0,0201	0,0201	0,0193	0,0196
		Test	0,0227	0,0228	0,0226	0,0227	0,0223	0,0221

Algoritmo de depresión astrocítica

Asentamiento de hormigón

Como se puede observar en la siguiente tabla, se muestra la media de los ECM alcanzados por la RNA y la RNGA con el problema de asentamiento de hormigón usando el algoritmo de depresión. Aunque en la columna mejor se pueda ver que la RNGA supera a la RNA, aunque en las combinaciones individuales de activación-iteración la RNGA supera a la RNA en fase de entrenamiento y validación no lo hace en la etapa final de test.

Tabla 57. Resultados en el problema asentamiento de hormigón de la RNGA con el alg. de depresión y la RNA, obtenidos en el entrenamiento, validación y test con la arquitectura de 2 capas ocultas. Los valores que se muestran corresponden a la media del ECM.

Conjunto de datos	RNGA					RNA
	2-4	2-6	3-6	3-8	Mejor	
Entre...	0,0528	0,0461	0,0490	0,0386	0,0397	0,0507
Valid...	0,0800	0,0741	0,0766	0,0752	0,0649	0,0750
Test	0,0815	0,0781	0,0806	0,0834	0,0713	0,0764

Fuegos forestales

Como se puede observar en la siguiente tabla, se muestra la media de los ECM alcanzados por la RNA y la RNGA con el problema de asentamiento de fuegos forestales el algoritmo de depresión. El algoritmo de depresión en redes con dos capas para este problema específico logra siempre mucho mejor desempeño que la RNA.

Tabla 58. Resultados en el problema fuegos forestales de la RNGA con el alg. de depresión y la RNA, obtenidos en el entrenamiento, validación y test con la arquitectura de 2 capas ocultas. Los valores que se muestran corresponden a la media del ECM.

Conjunto de datos	RNGA					RNA
	2-4	2-6	3-6	3-8	Mejor	
Entre...	0,3258	0,3261	0,3199	0,3160	0,3219	0,3317
Valid...	0,3630	0,3626	0,3635	0,3647	0,3586	0,3657
Test	0,3788	0,3808	0,3839	0,3879	0,3810	0,3905

Rendimiento estudiantil

Como se puede observar en la siguiente tabla, se muestra la media de los ECM alcanzados por la RNA y la RGA con el problema de rendimiento estudiantil forestales el algoritmo de depresión. Se puede observar que el algoritmo de depresión en RGA para este problema específico tiene un desempeño aceptable, pero no puede superar a la RNA.

Tabla 59. Resultados en el problema rendimiento estudiantil de la RGA con el alg. de depresión y la RNA, obtenidos en el entrenamiento, validación y test con la arquitectura de 2 capas ocultas. Los valores que se muestran corresponden a la media del ECM.

Conjunto de datos	RGA					RNA
	2-4	2-6	3-6	3-8	Mejor	
Entre...	0,0192	0,0193	0,0191	0,0192	0,0189	0,0167
Valid...	0,0218	0,0219	0,0217	0,0218	0,0206	0,0196
Test	0,0244	0,0244	0,0244	0,0248	0,0240	0,0221

Algoritmo de receptores endocannabinoides astrocíticos

Asentamiento de hormigón

Como se puede observar en la siguiente tabla, se muestra la media de los ECM alcanzados por la RNA y la RNGA con el problema de asentamiento de hormigón forestales el algoritmo de receptores de endocannabinoides. En la tabla se muestra que la RNGA consigue tener un error tres veces más bajo que la RNA en la fase de entrenamiento, lo que hace que la red tenga *overfitting* y, aunque en algún caso la RNGA también funcione mejor en la fase de validación, se ve afectado en la fase de test ya que no supera a la RNA.

Tabla 60. Resultados en el problema asentamiento de hormigón de la RNGA con el alg. receptores de endocannabinoides y la RNA, obtenidos en el entrenamiento, validación y test con la arquitectura de 2 capas ocultas. Los valores que se muestran corresponden a la media del ECM.

Conjunto de datos	RNGA					RNA
	2-4	2-6	3-6	3-8	Mejor	
Entre...	0,0222	0,0195	0,0189	0,0179	0,0149	0,0507
Valid...	0,0736	0,0772	0,0768	0,0803	0,0588	0,0750
Test	0,0927	0,1072	0,1036	0,1158	0,0943	0,0764

Fuegos forestales

Como se puede observar en la siguiente tabla, se muestra la media de los ECM alcanzados por la RNA y la RNGA con el problema de fuegos forestales el algoritmo de receptores de endocannabinoides.

Tabla 61. Resultados en el problema fuegos forestales de la RNGA con el alg. receptores de endocannabinoides y la RNA, obtenidos en el entrenamiento, validación y test con la arquitectura de 2 capas ocultas. Los valores que se muestran corresponden a la media del ECM.

Conjunto de datos	RNGA					RNA
	2-4	2-6	3-6	3-8	Mejor	
Entre...	0,3057	0,3177	0,3079	0,3048	0,3112	0,3317
Valid...	0,3689	0,3639	0,3709	0,3677	0,3597	0,3657
Test	0,4029	0,3963	0,4078	0,4078	0,4049	0,3905

Rendimiento estudiantil

Como se puede observar en la siguiente tabla, se muestra la media de los ECM alcanzados por la RNA y la RNGA con el problema de rendimiento estudiantil forestales el algoritmo de receptores de endocannabinoides. Se puede observar que este tipo de algoritmo en la RNGA para este problema específico tiene un desempeño aceptable, pero no puede superar a la RNA.

Tabla 62. Resultados en el problema rendimiento estudiantil de la RNGA con el alg. receptores de endocannabinoides y la RNA, obtenidos en el entrenamiento, validación y test con la arquitectura de 2 capas ocultas. Los valores que se muestran corresponden a la media del ECM.

Conjunto de datos	RNGA					RNA
	2-4	2-6	3-6	3-8	Mejor	
Entre...	0,0195	0,0198	0,0194	0,0197	0,0192	0,0167
Valid...	0,0220	0,0223	0,0220	0,0222	0,0207	0,0196
Test	0,0257	0,0255	0,0263	0,0263	0,0249	0,0221

Búsqueda de algoritmo astrocítico usando computación coevolutiva

Asentamiento de hormigón

Como se puede observar en la siguiente tabla, se muestran los porcentajes de acierto medio alcanzados por la RNA y las RNGA empleando un AGCC para el problema de asentamiento de hormigón. Se puede observar que el AGCC que usa la misma configuración glial en toda la red únicamente logra superar a la RNA en la fase de validación. Al indicarle al AGCC que los individuos pueden tener un algoritmo glial diferente en cada capa de la red y que tiene libertad al establecer los valores de los porcentajes de incremento/decremento se consigue que la RNGA supere a la RNA en la fase de validación. La tabla también muestra que la RNGA no logra generalizar, probablemente por la poca cantidad de datos que hay en la partición de entrenamiento.

Tabla 63. Resultados en el problema asentamiento de hormigón de la RNGA creada con AGCC y la RNA, obtenidos en el entrenamiento, validación y test con la arquitectura de 2 capas ocultas. Los valores que se muestran corresponden a la media del ECM.

Asentamiento de hormigón - 2 capas ocultas - 9,7,3,1	Info	Conjunto de datos	RNGA + AGCC	RNA
	Mismo Alg glial toda la red. %Dec \geq %Inc	Entre...	0,0175	0,0507
		Valid...	0,0885	0,0750
		Test	0,0930	0,0764
	Mismo Alg glial toda la red	Entre...	0,0165	0,0507
		Valid...	0,0776	0,0750
		Test	0,0975	0,0764
	Mismo Alg glial por capas	Entre...	0,0118	0,0507
		Valid...	0,0749	0,0750
		Test	0,0989	0,0764
	Mismo Alg glial por capas. %Dec \geq %Inc	Entre...	0,0137	0,0507
		Valid...	0,0758	0,0750
		Test	0,1040	0,0764

Fuegos forestales

Como se puede observar en la siguiente tabla, se muestran los porcentajes de acierto medio alcanzados por la RNA y las RNGA empleando un AGCC para el problema de fuegos forestales. Se puede observar en la tabla que la RNGA supera a la RNA en las etapas de entrenamiento y validación en todas las casuísticas. La tabla también refleja que las RNGA que están forzadas a usar la misma configuración de glial en toda la red tienen un desempeño similar en todas las fases de la generación del modelo mientras que si la RNGA puede tener una configuración diferente del algoritmo glial en cada capa los valores de test se resienten ligeramente al conseguir un mayor muy notable en la fase de entrenamiento.

Tabla 64. Resultados en el problema fuegos forestales de la RNGA creada con AGCC y la RNA, obtenidos en el entrenamiento, validación y test con la arquitectura de 2 capas ocultas. Los valores que se muestran corresponden a la media del ECM.

Incendios Forestales - 2 capas ocultas - 12,8,5,1	Info	Conjunto de datos	RNGA + AGCC	RNA
	Mismo Algoritmo toda la red. %Dec \geq %Inc	Entrenamiento	0,3104	0,3317
		Validación	0,3635	0,3657
		Test	0,3967	0,3905
	Mismo Algoritmo toda la red	Entrenamiento	0,3073	0,3317
		Validación	0,3646	0,3657
		Test	0,3984	0,3905
	Mismo Algoritmo por capas	Entrenamiento	0,2995	0,3317
		Validación	0,3633	0,3657
		Test	0,4091	0,3905
	Mismo Algoritmo por capas. %Dec \geq %Inc	Entrenamiento	0,2982	0,3317
		Validación	0,3639	0,3657
		Test	0,4052	0,3905

Rendimiento estudiantil

Como se puede observar en la siguiente tabla, se muestran los porcentajes de acierto medio alcanzados por la RNA y las RNGA empleando un AGCC para el problema de rendimiento estudiantil. La tabla indica que los errores tan bajos obtenidos por la RNA no permiten mucha mejora y la RNGA no logra mejorar sus resultados.

Tabla 65. Resultados en el problema rendimiento estudiantil de la RNGA creada con AGCC y la RNA, obtenidos en el entrenamiento, validación y test con la arquitectura de 2 capas ocultas. Los valores que se muestran corresponden a la media del ECM.

Rendimiento estudiantil - 2 capas ocultas - 30,9,4,1	Info	Conjunto de datos	RNGA + AGCC	RNA
	Mismo Algoritmo toda la red. %Dec \geq %Inc	Entrenamiento	0,0178	0,0167
		Validación	0,0208	0,0196
		Test	0,0348	0,0221
	Mismo Algoritmo toda la red	Entrenamiento	0,0181	0,0167
		Validación	0,0207	0,0196
		Test	0,0308	0,0221
	Mismo Algoritmo por capas	Entrenamiento	0,0185	0,0167
		Validación	0,0213	0,0196
		Test	0,0275	0,0221
	Mismo Algoritmo por capas. %Dec \geq %Inc	Entrenamiento	0,0183	0,0167
		Validación	0,0212	0,0196
		Test	0,0325	0,0221

A.2.3. Análisis de astrocitos artificiales en redes con 3 capas ocultas

Algoritmos de potenciación astrocítica

Asentamiento de hormigón

Como se puede observar en la siguiente tabla, se muestran los valores medios de ECM en el problema de asentamiento de hormigón obtenidos con la RNA y la RNGA usando algoritmos de potenciación. En la tabla se puede observar que todos los algoritmos de potenciación en la RNGA generan muy buenos resultados en entrenamiento, superando por creces los obtenidos por la RNA. Esta gran mejora genera *overfitting*, lo que hace que en validación y test la RNA tenga mejores valores. De hecho, al fijarse en la columna mejor se puede ver que la RNGA puede tener mejores resultados que la RNA, pero no llega a generalizar lo suficiente como para que eso llegue a reducir los valores del ECM en la fase de test.

Tabla 66. Resultados en el problema asentamiento de hormigón de la RNGA con el alg. de potenciación y la RNA, obtenidos en el entrenamiento, validación y test con la arquitectura de 3 capas ocultas. Los valores que se muestran corresponden a la media del ECM.

	Alg glial	Conjunto de datos	RNGA					RNA
			2-4	2-6	3-6	3-8	Mejor	
Asentamiento de hormigón - 3 capas ocultas - 7,4,5,3,1	Efecto de atenuación astrocítica	Entre...	0,0142	0,0095	0,0107	0,0089	0,0107	0,0327
		Valid...	0,0709	0,0768	0,0748	0,0785	0,0585	0,0620
		Test	0,0967	0,1081	0,1082	0,1241	0,0974	0,0621
	Act. no consecutivas, modificación sin límite	Entre...	0,0220	0,0154	0,0207	0,0213	0,0174	0,0327
		Valid...	0,0729	0,0712	0,0700	0,0733	0,0584	0,0620
		Test	0,0809	0,0878	0,0781	0,0804	0,0725	0,0621
	Efecto de proceso global	Entre...	0,0279	0,0180	0,0167	0,0127	0,0243	0,0327
		Valid...	0,1235	0,1752	0,1765	0,2459	0,1125	0,0620
		Test	0,0765	0,0918	0,0856	0,0985	0,0781	0,0621
	Act. consecutivas, modificación con límite	Entre...	0,0226	0,0198	0,0255	0,0270	0,0213	0,0327
		Valid...	0,0677	0,0720	0,0690	0,0697	0,0594	0,0620
		Test	0,0760	0,0839	0,0772	0,0743	0,0700	0,0621
	Act. no consecutivas, modificación con límite	Entre...	0,0210	0,0184	0,0220	0,0213	0,0180	0,0327
		Valid...	0,0720	0,0719	0,0734	0,0721	0,0594	0,0620
		Test	0,0816	0,0898	0,0843	0,0805	0,0718	0,0621
	Act. consecutivas, modificación sin límite	Entre...	0,0241	0,0199	0,0264	0,0268	0,0222	0,0327
		Valid...	0,0705	0,0729	0,0709	0,0708	0,0602	0,0620
		Test	0,0783	0,0862	0,0749	0,0775	0,0720	0,0621

Fuegos forestales

Como se puede observar en la siguiente tabla, se muestran los valores medios de ECM en el problema de los fuegos forestales obtenidos con la RNA y la RNGA usando algoritmos de potenciación. En la tabla se puede observar que los algoritmos gliales que cuentan las activaciones de manera consecutiva y las que no lo hacen de manera consecutiva se adaptan mejor en todas las fases de la creación del modelo que la RNA. Sin embargo, los algoritmos que simulan un efecto más prolongado del SG, los algoritmos de atenuación y proceso global, mejoran a la RNA en alguna de sus fases; el primero de ellos tiene genera algo de *overfitting* y tiene mejor desempeño en la fase de test, mientras que el segundo no se adapta también como la RNA en las dos

primeras fases de la creación de la red, pero generaliza muy bien haciendo que los valores en la fase de test sean mejores que los obtenidos por la RNA.

Tabla 67. Resultados en el problema fuegos forestales de la RNGA con el alg. de potenciación y la RNA, obtenidos en el entrenamiento, validación y test con la arquitectura de 3 capas ocultas. Los valores que se muestran corresponden a la media del ECM.

	Alg glial	Conjunto de datos	RNGA					RNA
			2-4	2-6	3-6	3-8	Mejor	
Incendios Forestales - 3 capas ocultas - 12,8,5,3,1	Efecto de atenuación astrocítica	Entre...	0,3036	0,2969	0,3005	0,3008	0,3007	0,3296
		Valid...	0,3681	0,3672	0,3663	0,3667	0,3595	0,3655
		Test	0,4084	0,4149	0,4083	0,4190	0,4119	0,3935
	Act. no consecutivas, modificación sin límite	Entre...	0,3194	0,3132	0,3188	0,3163	0,3191	0,3296
		Valid...	0,3631	0,3658	0,3644	0,3630	0,3592	0,3655
		Test	0,3883	0,3928	0,3862	0,3888	0,3868	0,3935
	Efecto de proceso global	Entre...	0,3344	0,3392	0,3400	0,3382	0,3358	0,3296
		Valid...	0,3666	0,3731	0,3718	0,3810	0,3620	0,3655
		Test	0,3922	0,3813	0,3796	0,3761	0,3861	0,3935
	Act. consecutivas, modificación con límite	Entre...	0,3288	0,3195	0,3322	0,3318	0,3286	0,3296
		Valid...	0,3630	0,3638	0,3632	0,3630	0,3584	0,3655
		Test	0,3788	0,3873	0,3776	0,3775	0,3798	0,3935
	Act. no consecutivas, modificación con límite	Entre...	0,3173	0,3131	0,3157	0,3177	0,3150	0,3296
		Valid...	0,3627	0,3654	0,3631	0,3638	0,3590	0,3655
		Test	0,3858	0,3919	0,3851	0,3878	0,3915	0,3935
	Act. consecutivas, modificación sin límite	Entre...	0,3265	0,3223	0,3299	0,3319	0,3271	0,3296
		Valid...	0,3639	0,3647	0,3639	0,3638	0,3591	0,3655
		Test	0,3807	0,3843	0,3786	0,3785	0,3798	0,3935

Rendimiento estudiantil

Como se puede observar en la siguiente tabla, se muestran los valores medios de ECM en el problema de rendimiento estudiantil obtenidos con la RNA y la RNGA usando algoritmos de potenciación. Se puede observar que en las todas las combinaciones de activación-iteración 2-4, 2-6, 3-6 y 3-8 las RNGA no logran mejorar en ningún momento los resultados obtenidos por la RNA. Incluso si se acude a la columna mejor la RNGA no supera a los valores obtenidos por la RNA.

Tabla 68. Resultados en el problema rendimiento estudiantil de la RNGA con el alg. de potenciación y la RNA, obtenidos en el entrenamiento, validación y test con la arquitectura de 3 capas ocultas. Los valores que se muestran corresponden a la media del ECM.

	Alg glial	Conjunto de datos	RNGA					RNA
			2-4	2-6	3-6	3-8	Mejor	
Rendimiento estudiantil - 3 capas ocultas - 7,4,5,3,1	Efecto de atenuación astrocítica	Entre...	0,0188	0,0188	0,0187	0,0192	0,0185	0,0161
		Valid...	0,0224	0,0222	0,0221	0,0223	0,0207	0,0196
		Test	0,0257	0,0257	0,0256	0,0258	0,0251	0,0225
	Act. no consecutivas, modificación sin límite	Entre...	0,0185	0,0188	0,0185	0,0184	0,0182	0,0161
		Valid...	0,0219	0,0220	0,0219	0,0220	0,0205	0,0196
		Test	0,0248	0,0255	0,0248	0,0249	0,0242	0,0225
	Efecto de proceso global	Entre...	0,0171	0,0178	0,0179	0,0181	0,0172	0,0161
		Valid...	0,0206	0,0227	0,0238	0,0262	0,0198	0,0196
		Test	0,0279	0,0415	0,0453	0,0662	0,0299	0,0225
	Act. consecutivas, modificación con límite	Entre...	0,0182	0,0186	0,0184	0,0184	0,0179	0,0161
		Valid...	0,0207	0,0210	0,0208	0,0208	0,0197	0,0196
		Test	0,0236	0,0238	0,0231	0,0236	0,0230	0,0225
	Act. no consecutivas, modificación con límite	Entre...	0,0183	0,0189	0,0185	0,0183	0,0179	0,0161
		Valid...	0,0217	0,0221	0,0217	0,0216	0,0205	0,0196
		Test	0,0247	0,0251	0,0247	0,0249	0,0243	0,0225
	Act. consecutivas, modificación sin límite	Entre...	0,0182	0,0185	0,0185	0,0184	0,0181	0,0161
		Valid...	0,0207	0,0211	0,0207	0,0208	0,0196	0,0196
		Test	0,0233	0,0236	0,0231	0,0233	0,0228	0,0225

Algoritmo de depresión astrocítica

Asentamiento de hormigón

Como se puede observar en la siguiente tabla, se muestran los valores medios de ECM en el problema de asentamiento de hormigón obtenidos con la RNA y la RNGA usando el algoritmo de depresión. La tabla muestra que este algoritmo glial no es el adecuado para este problema específico.

Tabla 69. Resultados en el problema asentamiento de hormigón de la RGA con el alg. de depresión y la RNA, obtenidos en el entrenamiento, validación y test con la arquitectura de 3 capas ocultas. Los valores que se muestran corresponden a la media del ECM.

Conjunto de datos	RGA					RNA
	2-4	2-6	3-6	3-8	Mejor	
Entre...	0,0379	0,0277	0,0309	0,0254	0,0244	0,0327
Valid...	0,0740	0,0746	0,0755	0,0767	0,0618	0,0620
Test	0,0814	0,0884	0,0924	0,0950	0,0825	0,0621

Fuegos forestales

Como se puede observar en la siguiente tabla, se muestran los valores medios de ECM en el problema de fuegos forestales obtenidos con la RNA y la RGA usando el algoritmo de depresión. El algoritmo de depresión en redes con tres capas para este problema específico no logra un mejor desempeño que la RNA.

Tabla 70. Resultados en el problema fuegos forestales de la RGA con el alg. de depresión y la RNA, obtenidos en el entrenamiento, validación y test con la arquitectura de 3 capas ocultas. Los valores que se muestran corresponden a la media del ECM.

Conjunto de datos	RGA					RNA
	2-4	2-6	3-6	3-8	Mejor	
Entre...	0,3729	0,3722	0,3763	0,3721	0,3675	0,3296
Valid...	0,5767	0,5764	0,5802	0,5776	0,5542	0,3655
Test	0,5848	0,5913	0,5859	0,5840	0,5847	0,3935

Rendimiento estudiantil

Como se puede observar en la siguiente tabla, se muestran los valores medios de ECM en el problema de rendimiento estudiantil obtenidos con la RNA y la RNGA usando el algoritmo de depresión. El algoritmo de depresión en redes con tres capas para este problema específico no logra un mejor desempeño que la RNA.

Tabla 71. Resultados en el problema rendimiento estudiantil de la RNGA con el alg. de depresión y la RNA, obtenidos en el entrenamiento, validación y test con la arquitectura de 3 capas ocultas. Los valores que se muestran corresponden a la media del ECM.

Conjunto de datos	RNGA					RNA
	2-4	2-6	3-6	3-8	Mejor	
Entre...	0,0195	0,0194	0,0197	0,0195	0,0191	0,0161
Valid...	0,0218	0,0219	0,0222	0,0219	0,0207	0,0196
Test	0,0242	0,0246	0,0249	0,0246	0,0241	0,0225

Algoritmo de receptores endocannabinoides astrocíticos

Asentamiento de hormigón

Como se puede observar en la siguiente tabla, se muestran los valores medios de ECM en el problema de asentamiento de hormigón obtenidos con la RNA y la RNGA usando el algoritmo de receptores de endocannabinoides. La tabla muestra que los errores obtenidos por la RNA son muy bajos y el algoritmo glial de receptores de endocannabinoides no logra superarlos.

Tabla 72. Resultados en el problema asentamiento de hormigón de la RNGA con el alg. de receptores de endocannabinoides y la RNA, obtenidos en el entrenamiento, validación y test con la arquitectura de 3 capas ocultas. Los valores que se muestran corresponden a la media del ECM.

Conjunto de datos	RNGA					RNA
	2-4	2-6	3-6	3-8	Mejor	
Entre...	0,0188	0,0192	0,0188	0,0160	0,0155	0,0327
Valid...	0,0787	0,0812	0,0794	0,0810	0,0610	0,0620
Test	0,1104	0,1152	0,1183	0,1284	0,1026	0,0621

Fuegos forestales

Como se puede observar en la siguiente tabla, se muestran los valores medios de ECM en el problema de fuegos forestales obtenidos con la RNA y la RGA usando el algoritmo de receptores de endocannabinoides. La tabla muestra que en todas las combinaciones de iteración-activación probadas, la RGA obtiene mejores resultados que la RNA en la fase de entrenamiento y lo hace manera notable. En cuanto a la fase de validación, solamente la RGA con la combinación 3-8 obtiene mejores valores que la red que no emplea algoritmos gliales. La RNA se ve superada en la fase de test por las RGA que necesitan dos iteraciones para activar la excitabilidad del SG.

Tabla 73. Resultados en el problema fuegos forestales de la RGA con el alg. de receptores de endocannabinoides y la RNA, obtenidos en el entrenamiento, validación y test con la arquitectura de 3 capas ocultas. Los valores que se muestran corresponden a la media del ECM.

Conjunto de datos	RGA					RNA
	2-4	2-6	3-6	3-8	Mejor	
Entre...	0,3196	0,3208	0,3174	0,3173	0,3210	0,3296
Valid...	0,3678	0,3666	0,3674	0,3655	0,3598	0,3655
Test	0,3889	0,3893	0,3989	0,3972	0,3920	0,3935

Rendimiento estudiantil

Como se puede observar en la siguiente tabla, se muestran los valores medios de ECM en el problema de rendimiento estudiantil obtenidos con la RNA y la RGA usando el algoritmo de receptores de endocannabinoides. Este algoritmo glial en redes con tres capas para este problema específico no logra un mejor desempeño que la RNA.

Tabla 74. Resultados en el problema rendimiento estudiantil de la RGA con el alg. de receptores de endocannabinoides y la RNA, obtenidos en el entrenamiento, validación y test con la arquitectura de 3 capas ocultas. Los valores que se muestran corresponden a la media del ECM.

Conjunto de datos	RGA					RNA
	2-4	2-6	3-6	3-8	Mejor	
Entre...	0,0197	0,0203	0,0198	0,0198	0,0194	0,0161
Valid...	0,0224	0,0227	0,0225	0,0224	0,0210	0,0196
Test	0,0257	0,0259	0,0258	0,0257	0,0247	0,0225

Asentamiento de hormigón

Como se puede observar en la siguiente tabla, se muestran los valores medios de ECM alcanzados por la RNA y las RNGA empleando un AGCC para el problema de asentamiento de hormigón. La tabla muestra que la RNGA obtiene valores mucho mejores en la fase de entrenamiento que la RNA lo que provoca que tenga overfitting y no logre superar los valores de la RNA en las etapas de validación y test.

Tabla 75. Resultados en el problema asentamiento de hormigón de la RNGA creada con AGCC y la RNA, obtenidos en el entrenamiento, validación y test con la arquitectura de 3 capas ocultas. Los valores que se muestran corresponden a la media del ECM.

Asentamiento de hormigón - 3 capas ocultas - 7.4.5.3.1	Info	Conjunto de datos	RNGA + AGCC	RNA
	Mismo Alg glial toda la red. %Dec \geq %Inc	Entre...	0,0171	0,0327
		Valid...	0,0879	0,0620
		Test	0,1023	0,0621
	Mismo Alg glial toda la red	Entre...	0,0163	0,0327
		Valid...	0,0884	0,0620
		Test	0,1078	0,0621
	Mismo Alg glial por capas	Entre...	0,0081	0,0327
		Valid...	0,0781	0,0620
		Test	0,1079	0,0621
	Mismo Alg glial por capas. %Dec \geq %Inc	Entre...	0,0101	0,0327
		Valid...	0,0783	0,0620
		Test	0,1154	0,0621

Fuegos forestales

Como se puede observar en la siguiente tabla, se muestran los valores medios de ECM alcanzados por la RNA y las RNGA empleando un AGCC para el problema de fuegos forestales. La tabla muestra que cuando el AGCC emplea un único algoritmo glial en toda la red y no se impone la restricción en los porcentajes que los astrocitos modifican las conexiones se obtiene en todas las etapas de la red mejores valores que los obtenidos con la RNA. En las demás casuísticas la RNGA obtiene mejores en la fase de entrenamiento y validación que los obtenidos por la RNA pero no generaliza tanto. Esto se debe a que estas tres casuísticas, la RNGA mejora bastante los valores obtenidos en la fase de entrenamiento que la RNA que obtiene siempre mejores valores que la RNA.

Tabla 76. Resultados en el problema fuegos forestales de la RNGA creada con AGCC y la RNA, obtenidos en el entrenamiento, validación y test con la arquitectura de 3 capas ocultas. Los valores que se muestran corresponden a la media del ECM.

Incendios Forestales - 3 capas ocultas - 12,8,5,3,1	Info	Conjunto de datos	RNGA + AGCC	RNA
	Mismo Alg glial toda la red. %Dec \geq %Inc	Entre...	0,3164	0,3296
		Valid...	0,3614	0,3655
		Test	0,3976	0,3935
	Mismo Alg glial toda la red	Entre...	0,3179	0,3296
		Valid...	0,3622	0,3655
		Test	0,3929	0,3935
	Mismo Alg glial por capas	Entre...	0,2980	0,3296
		Valid...	0,3624	0,3655
		Test	0,4146	0,3935
	Mismo Alg glial por capas. %Dec \geq %Inc	Entre...	0,3007	0,3296
		Valid...	0,3625	0,3655
		Test	0,4107	0,3935

Rendimiento estudiantil

Como se puede observar en la siguiente tabla, se muestran los valores medios de ECM alcanzados por la RNA y las RNGA empleando un AGCC para el problema de rendimiento estudiantil. En este problema concreto y con la arquitectura en tres capas, la RNGA no supera en ninguna casuística los resultados obtenidos por la RNA.

Tabla 77. Resultados en el problema rendimiento estudiantil de la RNGA creada con AGCC y la RNA, obtenidos en el entrenamiento, validación y test con la arquitectura de 3 capas ocultas. Los valores que se muestran corresponden a la media del ECM.

Rendimiento estudiantil - 3 capas ocultas - 7,4,5,3,1	Info	Conjunto de datos	RNGA + AGCC	RNA
	Mismo Alg glial toda la red. %Dec ≥ %Inc	Entre...	0,0187	0,0161
		Valid...	0,0218	0,0196
		Test	0,0347	0,0225
	Mismo Alg glial toda la red	Entre...	0,0186	0,0161
		Valid...	0,0220	0,0196
		Test	0,0330	0,0225
	Mismo Alg glial por capas	Entre...	0,0185	0,0161
		Valid...	0,0210	0,0196
		Test	0,0277	0,0225
	Mismo Alg glial por capas. %Dec ≥ %Inc	Entre...	0,0185	0,0161
		Valid...	0,0212	0,0196
		Test	0,0355	0,0225

Apéndice B. Aplicación de simulación de RNGA y RNA

La aplicación de simulación de redes no tiene ninguna interacción con el usuario. El motivo de que la aplicación se ejecute en modo bash es poder permitir ejecutarla sin ningún problema en plataformas de HPC. La forma de interactuar con la herramienta es mediante ficheros de configuración cuya estructura se detalla en este apéndice, pero antes se procede a explicar cómo funciona.

La aplicación desarrollada tiene el nombre de *energlia.out* y la ejecución del programa es la siguiente:

```
$ ./energlia.out option --config=configfile.xml --out=outfile.xml
```

Donde:

- **option**: Es la opción a ejecutar del programa
 - **help**: Muestra una ayuda indicando cuales son los parámetros del programa y para que sirven. Es la única opción que no requiere los parámetros *config* y *out*.
 - **ann_fixed_glia**: Comando para ejecutar un AG que busca los pesos adecuados para una RNGA.
 - **angn_search_glia**: Comando para ejecutar un AG que busca los pesos de la red y los parámetros de la glia.
- **configfile.xml**: Es la ruta del archivo XML que contiene la configuración para ejecutar la opción escogida.
- **outfile.xml**: Es la ruta del archivo XML que generará el programa con la salida que genera la aplicación.

B.1. Ficheros de entrada

En este apartado se indica cuáles son los ficheros de entrada que puede recibir la aplicación *energlia* y como es su definición.

B.1.1. Poblaciones de individuos

Población binaria.

Como se puede ver en el siguiente fichero el nodo principal se llama *population* en donde se tienen que indicar el número de individuos que tiene la población mediante el atributo *num_individuals*. El atributo *s_genobool* indica cuanto ocupa el genotipo de toda la población.

Dentro del nodo *population* habrá tantos nodos *individual* como se halla especificado en el atributo *num_individuals*. Dentro de cada nodo *individual* hay nodos *genobool* para indicar el valor del genotipo. Dentro del *genobool* están los atributos *value* e *id*⁶ para indicar el valor del alelo y su posición respectivamente.

```
<?xml version="1.0"?>
<population version="1.0" encode="1" num_individuals="" s_genobool="">
  <individual version="1.0" encode="1">
    <genobool id="" value="" />
    <!-- more genobool nodes -->
  </individual>
  <!-- more individual nodes -->
</population>
```

Población real

Como se puede ver en el siguiente que aparece a continuación, el nodo principal se llama *population* en donde tenemos que indicar el número de individuos que tiene la población mediante el atributo *num_individuals*. El atributo *s_genofloat* indica el tamaño de números decimales que tiene toda la población, mientras que el atributo *s_genoshort* difiere en que se refiere a los números enteros.

Dentro del nodo *population* habrá tantos nodos *individual* como se halla especificado en el atributo *num_individuals*. Dentro de cada nodo *individual* hay nodos *genofloat* y *genoshort*, donde los primeros son para indicar los genes que usan los números decimales y los otros para los números enteros. Tanto *genoshort* como *genofloat* tienen los atributos *value* e *id* donde se indica el valor del gen y su posición dentro del genotipo respectivamente.

⁶ El valor de *id* debe comenzar en cero y el valor máximo será la longitud del genotipo menos uno

```
<?xml version="1.0"?>

<population version="1.0" encode="2" num_individuals="" s_genofloat=""
s_genoshort="">
  <individual version="1.0" encode="2">
    <genofloat id="" value="" />
    <!-- more genofloat nodes -->

    <genoshort id="" value="" />
    <!-- more genoshort nodes -->

  </individual>
  <!-- more individual nodes -->
</population>
```

B.1.2. Fichero de parámetros

El fichero de parámetros es el fichero que se le pas al programa *energía* para que se ejecute el AG con las redes

En el fichero que se muestra a continuación se puede ver la estructura genérica de este archivo cuyo nodo principal se llama *parameters* y contiene tres nodos (*ga*, *dataset* y *problem*). Como este fichero es muy grande se explicará que hay en cada subnodo dentro de diferentes subapartados.

```
<?xml version="1.0"?>

<parameters >
  <ga ... />
  <dataset ... />
  <problem .../>
</parameters>
```

Nodo para la definición del algoritmo genético

Para indicar la configuración del algoritmo genético hay que usar un nodo xml que se muestra a continuación.

```
<ga version="1.0" generations="" goal="">
  <!-- info about specific algorithm -->
</ga>
```

Se puede ver ver que el nodo principal se llama *ga* donde están los siguientes atributos:

- **generations:** Aquí se indica el número máximo de generaciones que realizará el algoritmo genético
- **goal:** indica el valor de fitness que se quiere conseguir con el algoritmo genético

Dentro de este nodo se especifican la configuración específica para cada algoritmo genético.

Algoritmo de estado estable

En el nodo que se muestra a continuación se puede ver la definición para usar un algoritmo genético de estado estable.

```
<steadystate version="1.0">
  <population pathname="" encode="">
    <selectmother algorithm="" window="" prob="" />
    <selectfather algorithm="" window="" prob="" />
    <mutation algorithm="" rate="" />
    <crossover algorithm="" rate="" mu="" />
    <replace algorithm="" />
  </population>
</steadystate>
```

Se puede ver que la definición es muy simple. El significado de los nodos `selectmother`, `selectfather`, `mutation`, `crossover` y `mutation` aparece definido en apartados posteriores.

En el nodo `population` se tiene que indicar la ruta del archivo donde está el fichero xml que contiene la aplicación usando el atributo `pathname` mientras que en el atributo `encode` se indica la codificación que usa la población donde se emplean los valores **1** y **2** para indicar si la codificación es binaria o real respectivamente. El valor del atributo `encode` sirve para comprobar rápidamente si los algoritmos indicados dentro del nodo `population` son los adecuados.

Algoritmo cooperativo coevolutivo

En el nodo que se muestra a continuación se puede ver la definición para usar un algoritmo genético cooperativo coevolutivo de estado estable.

```
<cooperativecoevolutionarysteadystate version="1.0">
  <selectfitness best="" random="" function="">
    <selectfunction algorithm="" window="" prob="" />
  </selectfitness>

  <fitness algorithm="" />

  <population id="" pathname="" encode="">
    <selectmother algorithm="" window="" prob="" />
    <selectfather algorithm="" window="" prob="" />
    <mutation algorithm="" rate="" />
    <crossover algorithm="" rate="" mu="" />
    <replace algorithm="" />
  </population>
  <!-- more population nodes -->
</cooperativecoevolutionarysteadystate>
```

Se puede ver que la definición es muy sencilla. El significado de los nodos *selectmother*, *selectfather*, *mutation*, *crossover* y *mutation* aparece definido en apartados posteriores.

En el nodo *population* se tiene que indicar la ruta del archivo donde está el fichero xml que contiene la aplicación usando el atributo *pathname* mientras que en el atributo *encode* se indica la codificación que usa la población donde se emplean los valores **1** y **2** para indicar si la codificación es binaria o real respectivamente. El valor del atributo *encode* sirve para comprobar rápidamente si los algoritmos indicados dentro del nodo *population* son los adecuados. El último atributo del nodo *population* que queda por indicar es *id* que sirve para indicar el orden de las poblaciones. Para la población de pesos se usará el identificador *0* mientras que para la población con los parámetros de la glia se emplea el identificador *1*.

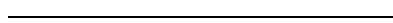
El nodo *selectfitness* es para indicar la selección de individuos que se emplean para calcular el fitness. El atributo *best* indica la cantidad de los mejores individuos de la población que se seleccionan. Para indicar el número de individuos que se seleccionan de manera aleatoria se emplea el atributo *random*. Con el atributo *function* se indica el número de individuos que se selecciona usando la función de selección que se quiera emplear y que se indica en el nodo interno llamado *selectfitness*⁷.

En el nodo *fitness* se indica el algoritmo de asignación de fitness a emplear los valores que puede tomar aparecen en la sección B.4.1)

Nodo para los conjuntos de datos

Para indicar los conjuntos de datos que se usarán en el programa por la RNGA hay que usar un nodo xml específico que se detalla a continuación.

```
<dataset version="1.0">
  <train inputs="" outputs="" />
  <validation inputs="" outputs="" />
  <test inputs="" outputs="" />
</dataset>
```



⁷ La definición del nodo *selectfitness* aparece en la sección B.3.1

Como podemos ver el nodo necesario para indicar los conjuntos de datos se llama *dataset* donde dentro de el se puede especificar diferentes nodos:

- **train:** Es para indicar el conjunto de datos para entrenamiento. Este nodo es obligatorio.
- **validation:** Es para indicar el conjunto de datos para validación. Este nodo es opcional.
- **test:** Es para indicar el conjunto de datos para test. Este nodo es opcional, pero si está definido también tiene que estarlo *validation*.

Dentro de los nodos *train*, *validation* y *test* se encuentran los atributos *inputs* y *outputs* con los que se indica donde se encuentran el fichero de patrones y las salidas deseadas respectivamente.

Nodo para la especificación del problema

Este nodo es el más específico que hay. Su estructura cambiará dependiendo del tipo de problema a resolver.

Algoritmo de estado estable. Búsqueda de los pesos de la RNGA.

En este apartado se explica como defenir la configuración de los parámetros de la RNGA.

```
<annFixedGlia version="1.0" num_neurons="9" num_layers="3"
    num_activation="2" num_iteration="4">
  <layer id="0" value="4" />
  <layer id="1" value="3" />
  <layer id="2" value="2" />

  <neuron id="0" activation_function="2">
    <astrocyte algorithm="1" per_increase="0.5" per_decrease="0.25" />
  </neuron>
  <neuron id="1" activation_function="2">
    <astrocyte algorithm="1" per_increase="0.5" per_decrease="0.25" />
  </neuron>
  <!-- more neuron nodes -->

  <output type="">
    <!-- Maybe need to add something right here-->
  </output>

  <outerror type="1" /> <!-- MSE or MAE -->

  <constraints >
    <weight min=" -10.2" max="1.45" />
  </constraints>
</annFixedGlia>
```

Como se puede ver el nodo principal se llama *annFixedGlia* y en el hay varios atributos:

- **num_neurons:** Indica el número de neuronas que tiene la red **num_layers:** Indica el número de capas que posee la red
- **num_activation:** Número de activaciones necesarias para que entre en funcionamiento el astrocito.
- **num_iteration:** Número de veces que se introduce el mismo patrón a la red para que le de tiempo a actuar al astrocito.

Dentro del nodo principal podemos encontrar cinco nodos distintos dentro de el que se detallan a continuación:

- **layer:** Relacionado con las capas de la red. Con estos atributos se puede indicar cuantas neuronas tiene una capa. El número de capa se identifica por el atributo *id* mientras que las neuronas que posee se indican con el valor de *value*
- **neuron:** Relacionado con las neuronas de la red. El atributo *id* indica el número de neurona dentro de la red mientras que con *activation_function* se indica la función de activación que posee (ver sección B.4.1). Dentro de el encontramos otro nodo llamado *astrocyte* con el que se especifica el algoritmo glial a emplear (ver sección B.4.1), el porcentaje de incremento y decremento de pesos mediante los atributos *algorithm*, *per_increase* y *per_decrease* respectivamente.
- **output:** Relacionado con la salida de la red. En el atributo *type* se indica el tipo la red (Ver la sección B.4.2 para consultar los valores que puede tomar).
 - Si la red es para regresión no hay que indicar nada más
 - Si la red es para clasificación hay que añadir un nodo interno identificado *classify* donde el atributo *step* indica la separación numérica entre clases y *classes* indica el número de clases a clasificar.

```
<classify step="0.5" class="2"/>
```

- Si la red es de tipo competitiva el nodo que se inserta se identifica por *competitive* y el valor de salida de la neurona ganadora se indica mediante el atributo *yes* mientras que las otras tomarán el valor especificado en *no*.

```
<competitive yes="1.2" no="0.2"/>
```

- **outerror:** Sirve para indicar como se calculará el error de la red. Se puede consultar los valores que puede tomar en la sección B.4.2).
- **constraints:** Relacionado con las restricciones que aplicará el algoritmo genético. A día de hoy sólo se puede indicar que el valor de las conexiones de los pesos tienen que estar entre un mínimo y un máximo. Esto está indicado en el nodo *weight* donde el atributo *min* indica el valor mínimo que puede tomar el peso y con *max* se indica el valor máximo que pueda tomar el peso.

Algoritmo cooperativo coevolutivo. Búsqueda de los parámetros de la RGA.

En este apartado se explica como definir la configuración para la simulación que busca los pesos y los parámetros de la glia de la RGA.

```
<angnSearchGlia version="1.0" num_neurons="9" num_layers="3">
  <!-- for layers -->
  <layer id="0" value="4"/>
  <layer id="1" value="3"/>
  <layer id="2" value="2"/>

  <!-- for neurons -->
  <neuron id="0" activation_function="2"/>
  <neuron id="1" activation_function="2"/>
  <neuron id="2" activation_function="2"/>
  <neuron id="3" activation_function="2"/>
  <neuron id="4" activation_function="2"/>
  <neuron id="5" activation_function="2"/>
  <neuron id="6" activation_function="2"/>
  <neuron id="7" activation_function="2"/>
  <neuron id="8" activation_function="2"/>

  <output type="">
    <!-- maybe you need to add something right here-->
  </output>

  <outerror type="0"/>

  <constraints >
    <weight min="-50.500000" max="100.500000"/>
    <same_glia_algorithm value="1"/>
    <iteration min="3" max="9"/>
    <activation min="2" max="8"/>
    <per_increase min="0.100000" max="0.900000"/>
    <per_decrease min="0.100000" max="0.900000"/>
    <per_decrease_greater_than_per_increase value="1"/>
  </constraints>
</angnSearchGlia>
```

Dentro del nodo principal podemos encontrar cinco nodos distintos dentro de el que se detallan a continuación:

- **layer:** Relacionado con las capas de la red. Con estos atributos puedes indicar cuantas neuronas tiene una capa. El número de capa se identifica por el

atributo *id* mientras que las neuronas que posee se indican con el valor de *value*

- **neuron:** Relacionado con las neuronas de la red. El atributo *id* indica el número de neurona dentro de la red mientras que con *activation_function* se indica la función de activación que posee (ver sección B.4.2)
- **output:** Relacionado con la salida de la red. En el atributo *type* se indica el tipo la red (Ver la sección B.4.2 para consultar los valores que puede tomar).
 - Si la red es para regresión no hay que indicar nada más
 - Si la red es para clasificación hay que añadir un nodo interno identificado *classify* donde el atributo *step* indica la separación numérica entre clases y *classes* indica el número de clases a clasificar.

```
<classify step="0.5" class="2"/>
```

- Si la red es de tipo competitiva el nodo que se inserta se identifica por *competitive* y el valor de salida de la neurona ganadora se indica mediante el atributo *yes* mientras que las otras tomarán el valor especificado en *no*.

```
<competitive yes="1.2" no="0.2"/>
```

- **outerror:** Sirve para indicar como se calculará el error de la red. Se puede consultar los valores que puede tomar en la sección B.4.2).
- **constraints:** Relacionado con las restricciones que aplicará el algoritmo genético.
 - Se puede indicar que el valor de las conexiones de los pesos tienen que estar entre un mínimo y un máximo. Esto está indicado en el nodo *weight* donde el atributo *min* indica el valor mínimo que puede tomar el peso y con *max* se indica el valor máximo que pueda tomar el peso.
 - Para indicar que se usará el mismo algoritmo glial en todas los elementos de procesado se tendrá que poner el valor *1* en el atributo *value* dentro del nodo *same_glia_algorithm*. Si queremos que el algoritmo glial pueda ser distinto a nivel de capas de la red se ha de poner el valor *0* en el atributo *value*.
 - Se puede especificar el valor máximo y mínimo que pueden tomar el valor de iteración de la red neuroglial. Esto se hace dentro del nodo

iteration con los atributos *min* y *max* para indicar el valor mínimo y máximo que pueden tomar respectivamente.

- Se puede especificar el valor máximo y mínimo que pueden tomar el valor de activación de la red neuroglial. Esto se hace dentro del nodo *activation* con los atributos *min* y *max* para indicar el valor mínimo y máximo que pueden tomar respectivamente.
- Con el nodo *per_increase* se puede especificar el porcentaje máximo y mínimo de incremento de pesos que el algoritmo glial puede tomar usando los atributos *max* y *min* respectivamente.
- Con el nodo *per_decrease* se puede especificar el porcentaje máximo y mínimo de decremento de pesos que el algoritmo glial puede tomar usando los atributos *max* y *min* respectivamente.
- Estableciendo dentro del nodo *per_decrease_greater_than_per_increase* el atributo *value* a *1* se indica que el porcentaje de decremento tiene que ser siempre mayor que el de incremento. Para quitar esta restricción hay que poner el valor de *value* a *0*.

B.2. Fichero de salida

En este apartado se explica como es la salida que se obtiene con el programa *energía*.

B.2.1. Fichero de Log

La salida de la aplicación únicamente genera un fichero de salida que contiene un log en el que se muestra la salida del AG indicando el error que se comete a través de las generaciones. Además, se muestra la información que corresponde con el fenotipo del mejor individuo. A continuación se muestra un ejemplo de cómo es el archivo de log:

```
<?xml version="1.0"?>
<logs version="1.0" code_version="1.2.1" >
  <log generation="0" train="0.50" validation="0.60" test="0.64" ms="0" >
    <!-- individual -->
  </log>
  <log generation="1" train="0.49" validation="0.62" test="0.63" ms="2" >
    <!-- individual -->
  </log>
  <log generation="3" train="0.44" validation="0.59" test="0.61" ms="6" >
    <!-- individual -->
  </log>
  <!-- more log nodes -->
</logs>
```

Como se puede ver, en el fichero de ejemplo, no se muestran todas las generaciones. Eso se debe a que sólo se almacena la información cuando el cambio es relevante, dicho de otra manera, cuando el cambio que se produce es a mejor. Esa es la razón por la que no se muestra el nodo para la generación 2 del algoritmo genético.

El atributo *code_version* indica cual es la versión del programa que genero el log.

Los errores que se pueden mostrar aquí son los que se obtienen de los dataset. En el ejemplo expuesto anteriormente se muestra el error para los conjuntos de entrenamiento (*train*), de validación (*validation*) y test (*test*). Es importante mencionar que no tienen que aparecer siempre los tres conjuntos de datos. El que es obligatorio que aparezca es el de entrenamiento, mientras que el de validación y el de test son opcionales, aunque si aparece el conjunto de test también tiene que aparecer el de validación.

El último atributo que se muestra, *ms*, es el que indica los milisegundos que han pasado desde la primera generación del algoritmo genético. Este parámetro sirve para ver cuán rápido converge el algoritmo, y nos da una idea de lo que se tarda en realizar una generación del algoritmo genético.

Dentro del nodo *log* se añade el fenotipo del mejor individuo de una generación específica. A día de hoy sólo se almacenan redes neurogliales, pero el día de mañana podría emplearse individuos de distinto tipo, como podría ser otras técnicas de machine learning para comparar resultados. Lo que se encuentra dentro del nodo *log* se especifica en los siguientes apartados.

Fenotipo de la RNGA

A diferencia con la versión anterior, ahora en los ficheros de salida se almacenan los fenotipos. La razón por la que no se guarda el genotipo de la RGNA es porque se necesita tener cierta información para su descodificación, como podría ser por ejemplo el número de elementos de procesado que hay en cada capa. Antes, toda esa información necesaria para saber descodificar a los individuos estaba almacenada en la configuración del fichero de parámetros. La principal ventaja de usar el fenotipo, es que permite tanto a un parser como a un humano leer toda la documentación que se encuentra sobre las RNGAs y entenderla.

La estructura que tienen que seguir los nodos que representan las RNGAs es la siguiente:

```
<angn version="1.0" type="feedforward" num_neurons="" num_layers=""  
  activations="" iterations="">
```

```
<output type="">
<!-- info about output -->
</output>

<neuron id="" layer="" activation_function="" per_increase=""
per_decrease="" algorithm="" />
<!-- more neuron nodes -->

<layers >
  <layer id="" value="" />
  <!-- more layer nodes -->
</layers>

<weights num="">
  <weight id="" value="" />
  <!-- more weight nodes -->
</weights>

</angn>
```

Podemos ver que en el nodo principal hay dos atributos con un valor específico. Esos atributos son *version* y *type*. Por un lado, el valor del atributo *version* indica la versión con la que se ha almacenado la RGA, mientras que por otro lado, el valor del atributo *type* avisa sobre que tipo de red se almacena. A día de hoy sólo se trabaja con redes *feedforward*.

Los otros valores del nodo principal indican el número de neuronas que tiene la red (*num_neurons*), el número de capas que tiene (*num_layers*), el número de activaciones para que haga efecto la glía (*activations*) y el número de veces que se procesa un patrón (*iterations*).

Dentro del nodo principal aparece el nodo *output*, este nodo indica para se tiene que emplear esta red mediante el atributo *type* que puede tomar los siguientes valores:

- **0** para clasificación. Además aquí hay que insertar el siguiente nodo:

```
<classify step="" class="" />
```

Donde el atributo *step* indica el paso entre clases, es decir lo que dista el valor de una clase entre otra, y *class* el número de clases a categorizar.

- **1** para regresión. En este caso no hay que añadir nada más.
- **2** para redes competitivas. A mayores hay que incluir el nodo:

```
<competitive yes="" no="" />
```

donde el atributo *yes* indica el valor que tomará el elemento de procesado que haya “ganado” mientras que los otros tomarán el valor indicado por el atributo *no*.

Los siguientes nodos que aparecen son los *neuron*. Aparece uno por cada elemento de procesado de la RNGA. A continuación se indica a que corresponde cada atributo:

- *id*: Posición que ocupa dentro de la red.
- *layer*: Número de capa a la que pertenece la red.
- *activation_function*: Función de activación que empleará la neurona. Los valores que puede tomar se pueden consultar en la sección B.4.2.
- *per_increase*: porcentaje de incremento que usará el astrocito que controla a esta neurona. Se expresa en tanto por uno.
- *per_decrease*: porcentaje de decremento que usará el astrocito que controla a esta neurona. Se expresa en tanto por uno.
- *algorithm*: Algoritmo glial que emplea el astrocito que controla a esta neurona. Los valores que puede tomar están en la sección B.4.2.

El nodo que aparece a continuación es el de *layers* donde dentro de el hay un nodo *layer* para cada capa de la RNGA. El atributo *id* indica a que capa de la RNGA se refiere mientras que el atributo *value* indica los elementos de procesado que hay dentro de esa capa.

Después de *layers* aparece el nodo *weights* para los pesos de la RNGA donde el atributo *num* indica el número de pesos que tiene la RNGA. Dentro tiene un nodo *weight* para cada peso que tiene la red donde, en el atributo *id*, se indica la posición de ese peso en la red mientras que el atributo *value* indica el valor de dicho peso.

B.3. Diccionario de nodos xml

En este apartado se explica la estructura de algunos nodos xml que se pueden emplear en diferentes secciones de los documentos.

B.3.1. Algoritmos de selección de individuos

La definición del nodo xml para definir los algoritmos de selección se muestra a continuación:

```
<selectX algorithm="" window="" prob="" />
```

El nodo principal se puede llamar *selectmother*, *selectfather* o *selectfitness*. Si el nodo se llama *selectmother* especifica el algoritmo que se empleará para escoger al individuo que hará de madre en la reproducción mientras que el otro, es decir *selectfather*, hará de padre. El nodo *selectfitness* es para indicar el algoritmo de selección para escoger los individuos que se emplearán para calcular el fitness en los algoritmos coevolutivos.

El atributo *algorithm* especifica el algoritmo de selección que se usará. Este atributo puede tomar los siguientes valores:

- **1:** Algoritmo de selección aleatoria. En este caso no se usan los atributos *window* y *prob*.
- **2:** Algoritmo de selección por ruleta. En este caso no se usan los atributos *window* y *prob*.
- **3:** Algoritmo de selección por torneo determinista. En este caso hay que darle un valor al atributo *window* especificando el tamaño de ventana para la selección. Con este algoritmo no se emplea el atributo *prob*.
- **4:** Algoritmo de selección por torneo probabilístico. En este caso hay que darle un valor al atributo *window* especificando el tamaño de ventana para la selección. Además hay que especificar el valor del atributo *prob* que se usa para indicar la probabilidad de coger al mejor.

B.3.2. Algoritmos de mutación de individuos

La definición del nodo xml para definir el algoritmo de mutación a emplear se muestra a continuación:

```
<mutation algorithm="" rate="" />
```

Como se puede ver sólo hay que especificar el valor de dos atributos. En el atributo *rate* se indica la probabilidad de que los descendientes sufran una mutación en alguno de sus genes. El otro atributo que hay que especificar es *algorithm* que indica el algoritmo a emplear:

- *Para codificación binaria*
 - **1:** bit inversion. Cambia el valor del alelo.
 - **2:** bit swapping. Intercambia el valor de dos alelos del genotipo
- *Para codificación real*
 - **10:** Cambio de valor: Cambia el valor del gen por otro aleatorio
 - **11:** Cambio en porcentaje: Como mucho cambia un 10 % el valor del gen.

B.3.3. Algoritmos de cruce de individuos

La definición del nodo xml para definir el algoritmo de cruce a emplear se muestra a continuación :

```
<crossover algorithm="" rate="" mu="" />
```

Como se puede ver, el primer atributo que hay que especificar es el algoritmo a emplear que puede tomar los siguientes valores:

- *Para codificación binaria*
 - **1**: SPX. Cruce de un punto.
 - **2**: DPX. Cruce de dos puntos.
 - **3**: UPX. Cruce uniforme.
- *Para codificación real*
 - **10**: SPX. Cruce de un punto.
 - **11**: DPX. Cruce de dos puntos.
 - **12**: UPX. Cruce uniforme.
 - **13**: SBX. Cruce que simula los cruces de codificación binaria. Sólo en este caso hay que especificar el valor de *mu* mediante el atributo *mu*.

El atributo *rate* sirve para indicar el porcentaje de cruces que se emplean en una generación del algoritmo genético.

El último atributo, *mu*, sólo se emplea con el algoritmo SBX.

B.3.4. Algoritmos de reemplazo de individuos

La definición del nodo xml para definir el algoritmo de reemplazo a emplear se muestra a continuación:

```
<replace algorithm="" />
```

En este nodo sólo tenemos que especificar el valor del atributo *algorithm* el cual puede tomar los valores:

- **1**. Siempre reemplaza a los padres.
- **2**. Reemplaza a los peores individuos.
- **3**. Reemplaza a individuos que tengan un valor de fitness similar.

B.4. Diccionario de identificadores

En este apartado se muestran cuales son los valores que pueden tomar los distintos identificadores y que significan cada uno.

B.4.1. Algoritmo genético

Algoritmos de Fitness

- **1:** Mínimo (*FITNESS_MIN*)
- **2:** Media (*FITNESS_AVG*)
- **3:** Máximo (*FITNESS_MAX*)

Algoritmo de Cruce

- Codificación binaria
 - **1:** SPX (*BINARY_SPX*)
 - **2:** DPX (*BINARY_DPX*)
 - **3:** UPX (*BINARY_UPX*)
- Codificación real
 - **10:** SPX (*REAL_SPX*)
 - **11:** DPX (*REAL_DPX*)
 - **12:** UPX (*REAL_UPX*)
 - **13:** SBX (*REAL_SBX*)

Algoritmo de Mutación

- Codificación binaria
 - **1:** inversión de bit (*BIT_INVERSION*)
 - **2:** intercambio de bits (*BIT_SWAPPING*)
- Codificación real
 - **10:** Cambio de valor (*CHANGE_VALUE*)
 - **11:** Cambio un porcentaje (*CHANGE_PERCENT*)

Codificación de individuos

- **1:** Codificación binaria (*BINARY_ENCODING*)
- **2:** Codificación real (*REAL_CODE*)

Algoritmo de selección

- **1:** Aleatorio (*SELECT_RANDOM*)
- **2:** Ruleta (*SELECT_ROULETTE*)
- **3:** Torneo determinístico (*SELECT_TOURNAMENT_DETERMINISTIC*)
- **4:** Torneo probabilístico (*SELECT_TOURNAMENT_DETERMINISTIC*)

Algoritmo de reemplazo

- **1:** Siempre reemplaza a los padres (*ALWAYS_REPLACE*)
- **2:** Reemplaza a los peores individuos (*WORST_REPLACING*)
- **3:** Reemplaza a individuos que tengan un valor de fitness similar (*SIMILAR_REPLACE*)

B.4.2. Redes Neuronales

Función de activación

- **0:** Función identidad. $f(x) = x$. (*IDENTITY_ACTIVATION*)
- **1:** Función umbral. $f(x) = (x \geq 0)$. (*THRESHOLD_ACTIVATION*)
- **2:** Función hiperbólica tangente. $f(x) = (1 - e^x)/(1 + e^x)$ (*HTANGENT_ACTIVATION*)
- **3:** Función sigmoidea. $f(x) = 1/(1 + e^x)$. (*SIGMOID_ACTIVATION*)

Función de error

- **0:** MSE (*OUT_ERROR_MSE*)
- **1:** MAE (*OUT_ERROR_MAE*)

Tipo de salida

- **0:** Clasificación (*OUTPUT_TYPE_CLASSIFY*)
- **1:** Regresión (*OUTPUT_TYPE_REGRESSION*)
- **2:** Competitiva (*OUTPUT_TYPE_COMPETITIVE*)

Algoritmo glial

- **0:** (*ATTENUATED EFFECT ASTROCYTE*)
- **1:** (*NON CONSECUTIVE UNLIMITED*)
- **2:** (*GLOBAL PROCESSING EFFECT*)
- **3:** (*CONSECUTIVE LIMITED*)
- **4:** (*NON CONSECUTIVE LIMITED*)
- **5:** (*CONSECUTIVE UNLIMITED*)
- **6:** (*DEPRESSION*)
- **7:** (*CANNABINOIDES*)

Apéndice C. Abreviaturas

Tabla 78. Tabla de abreviaturas.

AG	Algoritmo Genético
AGC	Algoritmo Genético Cooperativo
AGCC	Algoritmo Genético Cooperativo Coevolutivo
CESGA	Centro de Supercomputación de Galicia
CPD	Centro de Procesado de datos
ECM	Error Cuadrático Medio
HPC	High Performance Computing
IA	Inteligencia Artificial
RNA	Redes de Neuronas Artificiales
RNGA	Redes NeuroGliales Artificiales
SC	Sistema Conexionista
SEC	Sistema EndoCannabinoide
SG	Sistema Glial
SN	Sistema Nervioso
SNC	Sistema Nervioso Central
SNP	Sistema Nervioso Periférico

Apéndice D. Listado de figuras

Figura 1. Neurona teñida por la técnica de Golgi	7
Figura 2. Estructura de una neurona biológica	8
Figura 3. Clasificación del Sistema Glial	9
Figura 4. Astrocito humano del cerebro de un feto de 23 semanas	10
Figura 5. Sinapsis tripartita (Allen & Barres, 2009).....	12
Figura 6. Sinapsis biológica (izquierda) y modelo de neurona artificial (derecha) (Khan Academy, 2018)	15
Figura 7. Evolución del error de entrenamiento y validación de una RNA con sobreentrenamiento.....	18
Figura 8. Funcionamiento general de un algoritmo genético.....	19
Figura 9. Creación de una nueva generación en un algoritmo genético canónico ..	20
Figura 10. Codificación de los pesos de una RNA para un individuo del algoritmo genético.....	21
Figura 11. Red NeuroGlial Artificial (RNGA). Los círculos representan las neuronas y las estrellas representan los astrocitos.....	28
Figura 12. Diagrama de flujo del funcionamiento de una RNGA	29
Figura 13. Diagrama de flujo de un Algoritmo Genético Cooperativo Coevolutivo (AGCC).....	38
Figura 14. Cromosoma de un individuo de la población de pesos.....	39
Figura 15. Cromosoma de un individuo de la población del sistema glial donde toda la red usa la misma configuración	39
Figura 16. Cromosoma de un individuo de la población del sistema glial donde cada capa usa su propia configuración	39
Figura 17. Profiling de la herramienta de simulación de RNGA	41

Figura 18. Representación de una RNA en una matriz	41
Figura 19. Representación de una RNA en una matriz superior	42
Figura 20. Representaciones para almacenar las conexiones.....	43
Figura 21. Red con los valores de los arrays para el método propuesto.....	45
Figura 22. Inicialización de arrays para el método propuesto de almacenamiento de RNA	46
Figura 23. Cálculo de la salida de la RNA usando el método propuesto	46
Figura 24. Ahorro de memoria al usar el método propuesto para guardar las redes	47
Figura 25. Comparación del speed up al usar el método propuesto frente al tradicional sin utilizar las opciones de optimización en el compilador	48
Figura 26. Comparación del speed up al usar el método propuesto frente al tradicional empleando las opciones de optimización en el compilador	49
Figura 27. Script para enviar una simulación al Finisterrae2 donde el usuario establece los parámetros de la glía	65
Figura 28. Script para enviar una simulación al Finisterrae2 donde el AGCC es el encargado de establecer los parámetros de la glía.....	66
Figura 29. Ejemplo de informe generado automáticamente con la herramienta de informes.	67
Figura 30. Idea en la que se basa la cola virtual	70
Figura 31. Ejemplo de las gráficas de evolución del error de la aplicación web.....	72
Figura 32. Arquitectura global de la aplicación web.....	73
Figura 33. Arquitectura propuesta del sistema para la aplicación web	73
Figura 34. Resultados obtenidos para el problema de cáncer de mama con redes de tres capas ocultas. La combinación activación-iteración empleada para los algoritmos de potenciación, depresión y endocannabinoides es 3-6. El AGCC establece en cada	

capa una configuración distinta en cada capa de la RNGA. Los valores mostrados corresponden al porcentaje de acierto. 91

Figura 35. Resultados obtenidos para el problema de asentamiento de hormigón con redes de una capa oculta. La combinación activación-iteración empleada para los algoritmos de potenciación, depresión y endocannabinoides es 3-6. El AGCC establece en la RNGA la misma configuración glial y restringe a que el porcentaje de incremento no supere al de decremento. Los valores mostrados corresponden al ECM. 91

Figura 36. Evolución del error para el problema de cáncer de mama con redes de tres capas ocultas donde la RNGA emplea el algoritmo de efecto de atenuación astrocítica con una combinación activación-iteración 3-6. 92

Figura 37. Evolución del error para el problema de asentamiento de hormigón con redes de una capa oculta donde la RNGA emplea el algoritmo de efecto de atenuación astrocítica con una combinación activación-iteración 3-6. 93

Figura 38. Propuesta de codificación para los parámetros gliales 108

Apéndice E. Listado de tablas

Tabla 1. Combinaciones analizadas de activación-iteración.....	31
Tabla 2. Uso de memoria para las diferentes aproximaciones para almacenar las conexiones.....	43
Tabla 3. Memoria en bytes para almacenar las diferentes redes.....	47
Tabla 4. Comparación de speed up usando el método tradicional y el propuesto sin opciones de optimización en el compilador	48
Tabla 5. Comparación de speed up usando el método tradicional y el propuesto usando las opciones de optimización disponibles en el compilador	50
Tabla 6. Resultados en el problema de setas venenosas de la RNGA con el alg. de potenciación y la RNA, obtenidos en el entrenamiento, validación y test con la arquitectura de 1 capa oculta. Los valores que se muestran corresponden al % de acierto medio.	124
Tabla 7. Resultados en el problema de cáncer de mama de la RNGA con el alg. de potenciación y la RNA, obtenidos en el entrenamiento, validación y test con la arquitectura de 1 capa oculta. Los valores que se muestran corresponden al % de acierto medio.	125
Tabla 8. Resultados en el problema señal de ionosfera de la RNGA con el alg. de potenciación y la RNA, obtenidos en el entrenamiento, validación y test con la arquitectura de 1 capa oculta. Los valores que se muestran corresponden al % de acierto medio.	126
Tabla 9. Resultados en el problema de setas venenosas de la RNGA con el alg. de depresión y la RNA, obtenidos en el entrenamiento, validación y test con la arquitectura de 1 capa oculta. Los valores que se muestran corresponden al % de acierto medio.	127
Tabla 10. Resultados en el problema cáncer de mama de la RNGA con el alg. de depresión y la RNA, obtenidos en el entrenamiento, validación y test con la arquitectura de 1 capa oculta. Los valores que se muestran corresponden al % de acierto medio.	127
Tabla 11. Resultados en el problema señal de ionosfera de la RNGA con el alg. de depresión y la RNA, obtenidos en el entrenamiento, validación y test con la	

arquitectura de 1 capa oculta. Los valores que se muestran corresponden al % de acierto medio. 128

Tabla 12. Resultados en el problema de setas venenosas de la RNGA con el alg. de receptores de endocannabinoides y la RNA, obtenidos en el entrenamiento, validación y test con la arquitectura de 1 capa oculta. Los valores que se muestran corresponden al % de acierto 128

Tabla 13. Resultados en el problema cáncer de mama de la RNGA con el alg. de receptores de endocannabinoides y la RNA, obtenidos en el entrenamiento, validación y test con la arquitectura de 1 capa oculta. Los valores que se muestran corresponden al % de acierto medio. 129

Tabla 14. Resultados en el problema señal de ionosfera de la RNGA con el alg. de receptores de endocannabinoides y la RNA, obtenidos en el entrenamiento, validación y test con la arquitectura de 1 capa oculta. Los valores que se muestran corresponden al % de acierto medio. 129

Tabla 15. Resultados en el problema de setas venenosas de la RNGA creada con AGCC y la RNA, obtenidos en el entrenamiento, validación y test con la arquitectura de 1 capa oculta. Los valores que se muestran corresponden al % de acierto medio. 130

Tabla 16. Resultados en el problema cáncer de mama de la RNGA creada con AGCC y la RNA, obtenidos en el entrenamiento, validación y test con la arquitectura de 1 capa oculta. Los valores que se muestran corresponden al % de acierto medio. 131

Tabla 17. Resultados en el problema señal de ionosfera de la RNGA creada con AGCC y la RNA, obtenidos en el entrenamiento, validación y test con la arquitectura de 1 capa oculta. Los valores que se muestran corresponden al % de acierto medio. 132

Tabla 18. Resultados en el problema de setas venenosas de la RNGA con el alg. de potenciación y la RNA, obtenidos en el entrenamiento, validación y test con la arquitectura de 2 capas ocultas. Los valores que se muestran corresponden al % de acierto medio. 133

Tabla 19. Resultados en el problema cáncer de mama de la RNGA con el alg. de potenciación y la RNA, obtenidos en el entrenamiento, validación y test con la arquitectura de 2 capas ocultas. Los valores que se muestran corresponden al % de acierto medio. 134

Tabla 20. Resultados en el problema señal de ionosfera de la RNGA con el alg. de potenciación y la RNA, obtenidos en el entrenamiento, validación y test con la

arquitectura de 2 capas ocultas. Los valores que se muestran corresponden al % de acierto medio. 135

Tabla 21. Resultados en el problema de setas venenosas de la RNGA con el alg. de depresión y la RNA, obtenidos en el entrenamiento, validación y test con la arquitectura de 2 capas ocultas. Los valores que se muestran corresponden al % de acierto medio. 136

Tabla 22. Resultados en el problema cáncer de mama de la RNGA con el alg. de depresión y la RNA, obtenidos en el entrenamiento, validación y test con la arquitectura de 2 capas ocultas. Los valores que se muestran corresponden al % de acierto medio. 136

Tabla 23. Resultados en el problema señal de ionosfera de la RNGA con el alg. de depresión y la RNA, obtenidos en el entrenamiento, validación y test con la arquitectura de 2 capas ocultas. Los valores que se muestran corresponden al % de acierto medio. 137

Tabla 24. Resultados en el problema de setas venenosas de la RNGA con el alg. de receptores de endocannabinoides y la RNA, obtenidos en el entrenamiento, validación y test con la arquitectura de 2 capas ocultas. Los valores que se muestran corresponden al % de acierto..... 138

Tabla 25. Resultados en el problema cáncer de mama de la RNGA con el alg. de receptores de endocannabinoides y la RNA, obtenidos en el entrenamiento, validación y test con la arquitectura de 2 capas ocultas. Los valores que se muestran corresponden al % de acierto medio..... 138

Tabla 26. Resultados en el problema señal de ionosfera de la RNGA con el alg. de receptores de endocannabinoides y la RNA, obtenidos en el entrenamiento, validación y test con la arquitectura de 2 capas ocultas. Los valores que se muestran corresponden al % de acierto medio..... 139

Tabla 27. Resultados en el problema de setas venenosas de la RNGA creada con AGCC y la RNA, obtenidos en el entrenamiento, validación y test con la arquitectura de 2 capas ocultas. Los valores que se muestran corresponden al % de acierto medio. . 139

Tabla 28. Resultados en el problema cáncer de mama de la RNGA creada con AGCC y la RNA, obtenidos en el entrenamiento, validación y test con la arquitectura de 2 capas ocultas. Los valores que se muestran corresponden al % de acierto medio. 140

Tabla 29. Resultados en el problema señal de ionosfera de la RNGA creada con AGCC y la RNA, obtenidos en el entrenamiento, validación y test con la arquitectura de 2 capas ocultas. Los valores que se muestran corresponden al % de acierto medio. . 141

Tabla 30. Resultados en el problema de setas venenosas de la RNGA con el alg. de potenciación y la RNA, obtenidos en el entrenamiento, validación y test con la arquitectura de 3 capas ocultas. Los valores que se muestran corresponden al % de acierto medio. 142

Tabla 31. Resultados en el problema cáncer de mama de la RNGA con el alg. de potenciación y la RNA, obtenidos en el entrenamiento, validación y test con la arquitectura de 3 capas ocultas. Los valores que se muestran corresponden al % de acierto medio. 143

Tabla 32. Resultados en el problema señal de ionosfera de la RNGA con el alg. de potenciación y la RNA, obtenidos en el entrenamiento, validación y test con la arquitectura de 3 capas ocultas. Los valores que se muestran corresponden al % de acierto medio. 144

Tabla 33. Resultados en el problema de setas venenosas de la RNGA con el alg. de depresión y la RNA, obtenidos en el entrenamiento, validación y test con la arquitectura de 3 capas ocultas. Los valores que se muestran corresponden al % de acierto medio. 145

Tabla 34. Resultados en el problema de cáncer de mama de la RNGA con el alg. de depresión y la RNA, obtenidos en el entrenamiento, validación y test con la arquitectura de 3 capas ocultas. Los valores que se muestran corresponden al % de acierto medio. 145

Tabla 35. Resultados en el problema señal de ionosfera de la RNGA con el alg. de depresión y la RNA, obtenidos en el entrenamiento, validación y test con la arquitectura de 3 capas ocultas. Los valores que se muestran corresponden al % de acierto medio. 146

Tabla 36. Resultados en el problema de setas venenosas de la RNGA con el alg. de receptores de endocannabinoides y la RNA, obtenidos en el entrenamiento, validación y test con la arquitectura de 3 capas ocultas. Los valores que se muestran corresponden al % de acierto..... 146

Tabla 37. Resultados en el problema cáncer de mama de la RNGA con el alg. de receptores de endocannabinoides y la RNA, obtenidos en el entrenamiento, validación

y test con la arquitectura de 3 capas ocultas. Los valores que se muestran corresponden al % de acierto medio..... 147

Tabla 38. Resultados en el problema señal de ionosfera de la RNGA con el alg. de receptores de endocannabinoides y la RNA, obtenidos en el entrenamiento, validación y test con la arquitectura de 3 capas ocultas. Los valores que se muestran corresponden al % de acierto medio..... 147

Tabla 39. Resultados en el problema de setas venenosas de la RNGA creada con AGCC y la RNA, obtenidos en el entrenamiento, validación y test con la arquitectura de 3 capas ocultas. Los valores que se muestran corresponden al % de acierto medio. . 148

Tabla 40. Resultados en el problema cáncer de mama de la RNGA creada con AGCC y la RNA, obtenidos en el entrenamiento, validación y test con la arquitectura de 3 capas ocultas. Los valores que se muestran corresponden al % de acierto medio..... 149

Tabla 41. Resultados en el problema señal de ionosfera de la RNGA creada con AGCC y la RNA, obtenidos en el entrenamiento, validación y test con la arquitectura de 3 capas ocultas. Los valores que se muestran corresponden al % de acierto medio. . 150

Tabla 42. Resultados en el problema asentamiento de hormigón de la RNGA con el alg. de potenciación y la RNA, obtenidos en el entrenamiento, validación y test con la arquitectura de 1 capa oculta. Los valores que se muestran corresponden a la media del ECM. 151

Tabla 43. Resultados en el problema fuegos forestales de la RNGA con el alg. de potenciación y la RNA, obtenidos en el entrenamiento, validación y test con la arquitectura de 1 capa oculta. Los valores que se muestran corresponden a la media del ECM. 152

Tabla 44. Resultados en el problema rendimiento estudiantil de la RNGA con el alg. de potenciación y la RNA, obtenidos en el entrenamiento, validación y test con la arquitectura de 1 capa oculta. Los valores que se muestran corresponden a la media del ECM. 153

Tabla 45. Resultados en el problema asentamiento de hormigón de la RNGA con el alg. de depresión y la RNA, obtenidos en el entrenamiento, validación y test con la arquitectura de 1 capa oculta. Los valores que se muestran corresponden a la media del ECM. 154

Tabla 46. Resultados en el problema fuegos forestales de la RNGA con el alg. de depresión y la RNA, obtenidos en el entrenamiento, validación y test con la

arquitectura de 1 capa oculta. Los valores que se muestran corresponden a la media del ECM. 154

Tabla 47. Resultados en el problema rendimiento estudiantil de la RNGA con el alg. de depresión y la RNA, obtenidos en el entrenamiento, validación y test con la arquitectura de 1 capa oculta. Los valores que se muestran corresponden a la media del ECM. 155

Tabla 48. Resultados en el problema asentamiento de hormigón de la RNGA con el alg. de receptores de endocannabinoides y la RNA, obtenidos en el entrenamiento, validación y test con la arquitectura de 1 capa oculta. Los valores que se muestran corresponden a la media del ECM..... 155

Tabla 49. Resultados en el problema fuegos forestales de la RNGA con el alg. de receptores de endocannabinoides y la RNA, obtenidos en el entrenamiento, validación y test con la arquitectura de 1 capa oculta. Los valores que se muestran corresponden a la media del ECM. 156

Tabla 50. Resultados en el problema rendimiento estudiantil de la RNGA con el alg. de receptores de endocannabinoides y la RNA, obtenidos en el entrenamiento, validación y test con la arquitectura de 1 capa oculta. Los valores que se muestran corresponden a la media del ECM..... 156

Tabla 51. Resultados en el problema asentamiento de hormigón de la RNGA creada con AGCC y la RNA, obtenidos en el entrenamiento, validación y test con la arquitectura de 1 capa oculta. Los valores que se muestran corresponden a la media del ECM. 157

Tabla 52. Resultados en el problema fuegos forestales de la RNGA creada con AGCC y la RNA, obtenidos en el entrenamiento, validación y test con la arquitectura de 1 capa oculta. Los valores que se muestran corresponden a la media del ECM. 158

Tabla 53. Resultados en el problema rendimiento estudiantil de la RNGA creada con AGCC y la RNA, obtenidos en el entrenamiento, validación y test con la arquitectura de 1 capa oculta. Los valores que se muestran corresponden a la media del ECM. 159

Tabla 54. Resultados en el problema asentamiento de hormigón de la RNGA con el alg. de potenciación y la RNA, obtenidos en el entrenamiento, validación y test con la arquitectura de 2 capas ocultas. Los valores que se muestran corresponden a la media del ECM. 160

Tabla 55. Resultados en el problema fuegos forestales de la RNGA con el alg. de potenciación y la RNA, obtenidos en el entrenamiento, validación y test con la arquitectura de 2 capas ocultas. Los valores que se muestran corresponden a la media del ECM. 161

Tabla 56. Resultados en el problema rendimiento estudiantil de la RNGA con el alg. de potenciación y la RNA, obtenidos en el entrenamiento, validación y test con la arquitectura de 2 capas ocultas. Los valores que se muestran corresponden a la media del ECM. 162

Tabla 57. Resultados en el problema asentamiento de hormigón de la RNGA con el alg. de depresión y la RNA, obtenidos en el entrenamiento, validación y test con la arquitectura de 2 capas ocultas. Los valores que se muestran corresponden a la media del ECM. 164

Tabla 58. Resultados en el problema fuegos forestales de la RNGA con el alg. de depresión y la RNA, obtenidos en el entrenamiento, validación y test con la arquitectura de 2 capas ocultas. Los valores que se muestran corresponden a la media del ECM. 164

Tabla 59. Resultados en el problema rendimiento estudiantil de la RNGA con el alg. de depresión y la RNA, obtenidos en el entrenamiento, validación y test con la arquitectura de 2 capas ocultas. Los valores que se muestran corresponden a la media del ECM. 165

Tabla 60. Resultados en el problema asentamiento de hormigón de la RNGA con el alg. receptores de endocannabinoides y la RNA, obtenidos en el entrenamiento, validación y test con la arquitectura de 2 capas ocultas. Los valores que se muestran corresponden a la media del ECM..... 166

Tabla 61. Resultados en el problema fuegos forestales de la RNGA con el alg. receptores de endocannabinoides y la RNA, obtenidos en el entrenamiento, validación y test con la arquitectura de 2 capas ocultas. Los valores que se muestran corresponden a la media del ECM..... 166

Tabla 62. Resultados en el problema rendimiento estudiantil de la RNGA con el alg. receptores de endocannabinoides y la RNA, obtenidos en el entrenamiento, validación y test con la arquitectura de 2 capas ocultas. Los valores que se muestran corresponden a la media del ECM..... 167

Tabla 63. Resultados en el problema asentamiento de hormigón de la RNGA creada con AGCC y la RNA, obtenidos en el entrenamiento, validación y test con la arquitectura de 2 capas ocultas. Los valores que se muestran corresponden a la media del ECM. 168

Tabla 64. Resultados en el problema fuegos forestales de la RNGA creada con AGCC y la RNA, obtenidos en el entrenamiento, validación y test con la arquitectura de 2 capas ocultas. Los valores que se muestran corresponden a la media del ECM. 169

Tabla 65. Resultados en el problema rendimiento estudiantil de la RNGA creada con AGCC y la RNA, obtenidos en el entrenamiento, validación y test con la arquitectura de 2 capas ocultas. Los valores que se muestran corresponden a la media del ECM. 170

Tabla 66. Resultados en el problema asentamiento de hormigón de la RNGA con el alg. de potenciación y la RNA, obtenidos en el entrenamiento, validación y test con la arquitectura de 3 capas ocultas. Los valores que se muestran corresponden a la media del ECM. 172

Tabla 67. Resultados en el problema fuegos forestales de la RNGA con el alg. de potenciación y la RNA, obtenidos en el entrenamiento, validación y test con la arquitectura de 3 capas ocultas. Los valores que se muestran corresponden a la media del ECM. 173

Tabla 68. Resultados en el problema rendimiento estudiantil de la RNGA con el alg. de potenciación y la RNA, obtenidos en el entrenamiento, validación y test con la arquitectura de 3 capas ocultas. Los valores que se muestran corresponden a la media del ECM. 174

Tabla 69. Resultados en el problema asentamiento de hormigón de la RNGA con el alg. de depresión y la RNA, obtenidos en el entrenamiento, validación y test con la arquitectura de 3 capas ocultas. Los valores que se muestran corresponden a la media del ECM. 175

Tabla 70. Resultados en el problema fuegos forestales de la RNGA con el alg. de depresión y la RNA, obtenidos en el entrenamiento, validación y test con la arquitectura de 3 capas ocultas. Los valores que se muestran corresponden a la media del ECM. 175

Tabla 71. Resultados en el problema rendimiento estudiantil de la RNGA con el alg. de depresión y la RNA, obtenidos en el entrenamiento, validación y test con la

arquitectura de 3 capas ocultas. Los valores que se muestran corresponden a la media del ECM. 176

Tabla 72. Resultados en el problema asentamiento de hormigón de la RNGA con el alg. de receptores de endocannabinoides y la RNA, obtenidos en el entrenamiento, validación y test con la arquitectura de 3 capas ocultas. Los valores que se muestran corresponden a la media del ECM..... 176

Tabla 73. Resultados en el problema fuegos forestales de la RNGA con el alg. de receptores de endocannabinoides y la RNA, obtenidos en el entrenamiento, validación y test con la arquitectura de 3 capas ocultas. Los valores que se muestran corresponden a la media del ECM..... 177

Tabla 74. Resultados en el problema rendimiento estudiantil de la RNGA con el alg. de receptores de endocannabinoides y la RNA, obtenidos en el entrenamiento, validación y test con la arquitectura de 3 capas ocultas. Los valores que se muestran corresponden a la media del ECM..... 177

Tabla 75. Resultados en el problema asentamiento de hormigón de la RNGA creada con AGCC y la RNA, obtenidos en el entrenamiento, validación y test con la arquitectura de 3 capas ocultas. Los valores que se muestran corresponden a la media del ECM. 178

Tabla 76. Resultados en el problema fuegos forestales de la RNGA creada con AGCC y la RNA, obtenidos en el entrenamiento, validación y test con la arquitectura de 3 capas ocultas. Los valores que se muestran corresponden a la media del ECM..... 179

Tabla 77. Resultados en el problema rendimiento estudiantil de la RNGA creada con AGCC y la RNA, obtenidos en el entrenamiento, validación y test con la arquitectura de 3 capas ocultas. Los valores que se muestran corresponden a la media del ECM. 180

Tabla 78. Tabla de abreviaturas..... 199