

An abstract graphic featuring three light blue circles of varying sizes. Two thin, light blue lines intersect: one runs diagonally from the top-left towards the center, and the other runs diagonally from the top-right towards the bottom-left. The circles are positioned such that they appear to be connected or influenced by these lines.

# **PRÁCTICA 1**

## **NEUROCOMPUTACIÓN**

**José Antonio Guerrero Avilés**  
**Nº Lista: 14**  
**cany@correo.ugr.es**

# INDICE

<b>1.- Descripción de los ficheros. ....</b>	<b>2</b>
<b>2.- Obtención de la topología (2 neuronas).....</b>	<b>3</b>
2.1- Prueba 1 .....	3
2.2- Prueba 2.....	3
2.3- Prueba 3.....	4
2.4- Prueba 4.....	4
2.5- Prueba 5.....	5
2.6- Prueba 6.....	5
2.7- Prueba 7.....	6
2.8- Conclusión.....	6
<b>3.- Obtención del mejor tipo de entrenamiento (2 neuronas).....</b>	<b>7</b>
3.1 – Trainlm .....	7
3.2 – Traingdx .....	7
3.3 – Traincgp.....	8
3.4 – Trainbfg .....	8
3.5 – Trainrp .....	8
3.6 – Trainscg .....	9
3.7 – Conclusiones.....	9
<b>4. – Pruebas con la mejor red (2 neuronas).....</b>	<b>9</b>
<b>5.- Obtención de la topología (3 neuronas).....</b>	<b>11</b>
5.1- Prueba 1 .....	11
5.2- Prueba 2.....	11
5.3- Prueba 3.....	12
5.4- Prueba 4.....	12
5.5- Prueba 5.....	13
5.6- Prueba 6.....	13
5.7- Prueba 7.....	14
5.8- Conclusión.....	14
<b>6.- Obtención del mejor tipo de entrenamiento (3 neuronas).....</b>	<b>15</b>
6.1 – Trainlm .....	15
6.2 – Traingdx .....	15
6.3 – Traincgp.....	16
6.4 – Trainbfg .....	16
6.5 – Trainrp .....	16
6.6 – Trainscg .....	17
6.7 – Conclusiones.....	17
<b>7. – Pruebas con la mejor red (3 neuronas).....</b>	<b>17</b>

# 1.- Descripción de los ficheros.

**Iris1.txt** – aparecen los ejemplos que deberán ser usados para el entrenamiento y validación de RNs Backpropagation que permitan la clasificación de datos Iris.

**Iris2.txt** – están contenidos los modelos de test que deberán ser utilizados en el proceso de evaluación de las RNs.

**Entrena.txt** – se obtiene a partir de Iris1.txt, eliminando las cabeceras y el formato será de 4 columnas de datos correspondientes a las entradas que aparecen en Iris1.txt

**Salida2.txt** – se obtiene a partir de Iris1.txt, eliminando las cabeceras; el formato será de 2 columnas de datos correspondientes a las salidas que aparecen en Iris1.txt. Estas salidas son el resultado de sustituir cada string de salida “sestosa”, “versicolor”, “virginica” por 2 valores binarios (1 0), (0 1), (1 1) dependiendo del caso.

**Salida3.txt** – se obtiene a partir de Iris1.txt, eliminando las cabeceras; el formato será de 3 columnas de datos correspondientes a las salidas que aparecen en Iris1.txt. Estas salidas son el resultado de sustituir cada string de salida “sestosa”, “versicolor”, “virginica” por 3 valores binarios (1 0 0), (0 1 0), (0 0 1) dependiendo del caso.

**TestEntrena.txt** – se obtiene a partir de Iris2.txt, eliminando las cabeceras y el formato será de 4 columnas de datos correspondientes a las entradas que aparecen en Iris2.txt

**TestSalida2.txt** – se obtiene a partir de Iris2.txt, eliminando las cabeceras; el formato será de 2 columnas de datos correspondientes a las salidas que aparecen en Iris2.txt. Estas salidas son el resultado de sustituir cada string de salida “sestosa”, “versicolor”, “virginica” por 2 valores binarios (1 0), (0 1), (1 1) dependiendo del caso.

**TestSalida3.txt** – se obtiene a partir de Iris2.txt, eliminando las cabeceras; el formato será de 3 columnas de datos correspondientes a las salidas que aparecen en Iris2.txt. Estas salidas son el resultado de sustituir cada string de salida “sestosa”, “versicolor”, “virginica” por 3 valores binarios (1 0 0), (0 1 0), (0 0 1) dependiendo del caso.

**ProcesamientoIris.m** – programa que lee los ficheros Iris1.tx e Iris2.txt y genera los ficheros Entrena.txt, Salida2.txt, Salida3.txt, TestEntrena.txt, TestSalida2.txt y TestSalida3.txt eliminando las cabeceras.

**PracticaIris.m** – Archivo con la mejor topología y tipo de entrenamiento (tanto para una salida con 2 neuronas como para 3) tras haber realizado todas las pruebas pertinentes contenidas en el fichero MedicionesPruebas.xls. Crea los archivos Entrena.txt, Salida2.txt, Salida3.txt, TestEntrena.txt, TestSalida2.txt y TestSalida3.txt porque llama a la función ProcesamientoIris(). Además, contiene un bucle de tamaño 10 y realiza automáticamente la media de las 10 ejecuciones, que es lo que muestra por pantalla y que son los datos que uso para calcular la mejor topología y el mejor tipo de entrenamiento (la media de las 10 iteraciones del bucle).

**MedicionesPruebas.xls** – Contiene el resultado de haber realizado pruebas con diferentes topologías y diferentes tipos de entrenamiento. Consta de 2 hojas, una para la salida con 2 neuronas y otra para la salida con 3 neuronas.

## 2.- Obtención de la topología (2 neuronas).

Para obtener la topología más adecuada, voy a realizar una serie de pruebas aumentando y/o disminuyendo tanto el número de capas de la red, como el número de neuronas de dichas capas dependiendo de los resultados que vayamos obteniendo. Cada prueba constará de 10 ensayos y como resultado de la prueba, nos quedaremos con la media de los resultados obtenidos en los ensayos realizados.

### 2.1- Prueba 1

Los datos de la tipología que voy a usar para esta prueba son:

<b>Nº Capas</b>	1
<b>Neuronas Capa</b>	10

Con esta topología, he obtenido los siguientes datos:

	<b>MSE</b>
<b>MEDIA</b>	0,0133

A partir de estos datos obtenidos (voy a usar la media de los ensayos realizados) iré cambiando el número de neuronas de la capa y el número de capas para ver si mejoran o empeoran los resultados. Si mejoran, me quedaré con la tipología que mejore estos resultados, si no, la desecharé.

### 2.2- Prueba 2

Para esta prueba, voy a probar, por ejemplo, a disminuir el número de neuronas de la capa y ver cuáles son los resultados.

<b>Nº Capas</b>	1
<b>Neuronas Capa</b>	5

Con esta topología he obtenido los datos que aparecen en la siguiente tabla:

	<b>MSE</b>
<b>MEDIA</b>	0,0167

Comparando las medias obtenidas en esta prueba (prueba 2) con las obtenidas en la prueba 1, se ve que teniendo una sola capa, si reduzco el número de neuronas de la capa, los resultados empeoran, por tanto, de momento, me quedo con la topología de la prueba 1.

### 2.3- Prueba 3

Para esta prueba, voy a probar a aumentar el número de neuronas respecto de la prueba 1. Los datos para esta prueba son:

<b>Nº Capas</b>	1
<b>Neuronas Capa</b>	15

Y los resultados que obtengo son los siguientes:

	<b>MSE</b>
<b>MEDIA</b>	0,0183

Comparando con los resultados obtenidos en la primera prueba (que es la mejor hasta ahora), se observa que aumentando el número de neuronas de la capa, no se mejoran los resultados, por lo tanto, la topología con la que nos quedamos momentáneamente es la de la primera prueba.

Como continuar aumentando el número de neuronas de la capa, no mejorará los resultados, lo que voy a probar ahora es a añadir una capa más (ahora tendré 2 capas) e ir aumentando y/o disminuyendo el número de neuronas de las capas para ver si los resultados que voy obteniendo mejoran mi mejor topología hasta el momento.

### 2.4- Prueba 4

Para esta prueba voy a usar los siguientes datos:

<b>Nº Capas</b>	2
<b>Neuronas Capa 1</b>	10
<b>Neuronas Capa 2</b>	10

He decidido empezar con estos datos, porque 10 es el número de neuronas con el que empecé la primera prueba.

Los datos obtenidos en esta prueba son los siguientes:

	<b>MSE</b>
<b>MEDIA</b>	0,0217

Comparando con la topología de la prueba 1, que era la mejor hasta el momento, se observa que añadiendo 2 capas con 10 neuronas cada capa, no se mejoran los resultados; por tanto, mantenemos la anterior topología.

Aunque añadiendo 2 capas con 10 neuronas cada capa no se mejoran los resultados obtenidos, voy a probar a aumentar y/o disminuir el número de neuronas de ambas capas para ver si alguna posible combinación puede mejorar los resultados de la primera prueba.

## 2.5- Prueba 5

Voy a aumentar el número de neuronas de la primera capa:

Nº Capas	2
Neuronas Capa 1	5
Neuronas Capa 2	10

Viendo los resultados de la tabla que pongo a continuación, se puede ver como disminuir el número de neuronas de la primera capa, mejora los resultados y por tanto, esta será la nueva topología que usaremos:

	MSE
MEDIA	0,01

No obstante, voy a probar a aumentar y disminuir el número de neuronas de la segunda capa para ver lo que sucede.

## 2.6- Prueba 6

Voy a mantener las neuronas de la capa 1 y a aumentar las de la capa 2:

Nº Capas	2
Neuronas Capa 1	5
Neuronas Capa 2	5

Los resultados que he obtenido son los siguientes:

	MSE
MEDIA	0,025

Y como se puede ver, tampoco mejora la topología que ya tenía.

## 2.7- Prueba 7

Por último, voy a probar a aumentar el número de neuronas de la segunda capa a 15.

<b>Nº Capas</b>	2
<b>Neuronas Capa 1</b>	5
<b>Neuronas Capa 2</b>	15

Los resultados de esta prueba son:

	<b>MSE</b>
<b>MEDIA</b>	0,02

Como se ve, tampoco mejoran los resultados de la topología de la prueba 5.

## 2.8- Conclusión

Viendo los resultados de las pruebas, se puede determinar que la mejor topología para la red es:

<b>Nº Capas</b>	2
<b>Neuronas Capa 1</b>	5
<b>Neuronas Capa 2</b>	10

Obteniendo como resultado:

<b>Nº Ensayo</b>	<b>MSE</b>	<b>%Aciertos</b>
<b>MEDIA (10 ensayos)</b>	<b>0,01002</b>	<b>98,00002</b>

### 3.- Obtención del mejor tipo de entrenamiento (2 neuronas)

Con la mejor topología obtenida en las pruebas anteriores, voy a realizar ahora una batería de pruebas, cambiando el tipo de entrenamiento en cada una de las pruebas para ver cuál es el entrenamiento con el que la mejor topología obtiene los mejores resultados.

La topología será:

Nº Capas	2
Neuronas Capa 1	5
Neuronas Capa 2	10

#### 3.1 – Trainlm

Tipo de entrenamiento	trainlm
-----------------------	---------

Este entrenamiento es: Levenberg-Marquardt backpropagation

Los resultados que he obtenido son (los mismos que en la obtención de la topología ya que usé este entrenamiento como entrenamiento base para todas las pruebas anteriores):

	MSE
MEDIA	0,0117

#### 3.2 – Traingdx

Tipo de entrenamiento	traingdx
-----------------------	----------

Este entrenamiento es: Gradient descent with momentum and adaptive learning rate backpropagation.

Los resultados con este tipo de entrenamiento son peores, como se puede observar en la siguiente tabla:

	MSE
MEDIA	0,0183

Como se puede ver, no mejoran a los resultados obtenidos con el tipo de entrenamiento “trainlm”.



### 3.3 – *Traincgp*

Tipo de entrenamiento	traincgp
-----------------------	----------

Este tipo de entrenamiento es: Conjugate gradient backpropagation with Polak-Ribière updates (restarts).

Los resultados que obtengo, y que una vez más no mejoran a los obtenidos con trainlm son:

	<b>MSE</b>
<b>MEDIA</b>	0,015

Como se puede ver, no mejoran a los resultados obtenidos con el tipo de entrenamiento “trainlm”.

### 3.4 – *Trainbfg*

Tipo de entrenamiento	trainbfg
-----------------------	----------

Este tipo de entrenamiento es: BFGS quasi-Newton backpropagation.

Con él, tampoco se mejoran los resultados obtenidos hasta el momento con trainlm:

	<b>MSE</b>
<b>MEDIA</b>	0,0133

Como se puede ver, no mejoran a los resultados obtenidos con el tipo de entrenamiento “trainlm”.

### 3.5 – *Trainrp*

Tipo de entrenamiento	trainrp
-----------------------	---------

Este tipo de entrenamiento es: Resilient backpropagation.

Los resultados obtenidos con este tipo de entrenamiento son:

	<b>MSE</b>
<b>MEDIA</b>	0,015

Como se puede ver, no mejoran a los resultados obtenidos con el tipo de entrenamiento “trainlm”.

### 3.6 – Trainscg

<b>Tipo Entrenamiento</b>	trainscg
---------------------------	----------

Este tipo de entrenamiento es: Scaled conjugate gradient backpropagation

Los resultados que he obtenido usando este tipo de entrenamiento son:

	<b>MSE</b>
<b>MEDIA</b>	0,0167

Como se puede ver, no mejoran a los resultados obtenidos con el tipo de entrenamiento “trainlm”.

### 3.7 – Conclusiones

Como se puede ver en los resultados que he obtenido, el tipo de entrenamiento con el que obtengo mejores resultados es el entrenamiento **Levenberg-Marquardt** (“trainlm”):

<b>Tipo de entrenamiento</b>	trainlm
<b>MEDIA (MSE)</b>	0,0117

## 4. – Pruebas con la mejor red (2 neuronas)

Después de realizar todas las pruebas anteriores, he obtenido la siguiente red como la mejor:

<b>Nº Capas</b>	2
<b>Neuronas Capa 1</b>	5
<b>Neuronas Capa 2</b>	15
<b>Tipo Entrenamiento</b>	trainlm

Con esta red voy a realizar una batería de 10 pruebas: mi programa PracticaIris.m incluye un bucle de 10 ensayos, por lo que el resultado que te da tras ejecutarlo es la media de los 10 ensayos; pero además, voy a ejecutar PracticaIris.m 10 veces, por lo que realmente habré realizado 100 ejecuciones, de esta manera, obtendré unos datos más fiables ya que el número de ejecuciones será bastante alto.

Los resultados que he obtenido al realizar estas pruebas son los que aparecen en la siguiente tabla:

Nº Ejecución	MSE	%Acierto
1	0,0117	97,667
2	0,01	98
3	0,0117	97,667
4	0,015	97
5	0,015	97
6	0,01	98
7	0,01	98
8	0,0117	97,667
9	0,0117	97,667
10	0,01	98
<b>MEDIA</b>	<b>0,01168</b>	<b>97,6668</b>

## 5.- Obtención de la topología (3 neuronas).

Para obtener la topología más adecuada, voy a realizar una serie de pruebas aumentando y/o disminuyendo tanto el número de capas de la red, como el número de neuronas de dichas capas dependiendo de los resultados que vayamos obteniendo. Cada prueba constará de 10 ensayos y como resultado de la prueba, nos quedaremos con la media de los resultados obtenidos en los ensayos realizados.

### 5.1- Prueba 1

Los datos de la tipología que voy a usar para esta prueba son:

<b>Nº Capas</b>	1
<b>Neuronas Capa</b>	10

Con esta topología, he obtenido los siguientes datos:

	<b>MSE</b>
<b>MEDIA</b>	0,0144

A partir de estos datos obtenidos (voy a usar la media de los ensayos realizados) iré cambiando el número de neuronas de la capa y el número de capas para ver si mejoran o empeoran los resultados. Si mejoran, me quedaré con la tipología que mejore estos resultados, si no, la desecharé.

### 5.2- Prueba 2

Para esta prueba, voy a probar, por ejemplo, a disminuir el número de neuronas de la capa y ver cuáles son los resultados.

<b>Nº Capas</b>	1
<b>Neuronas Capa</b>	5

Con esta topología he obtenido los datos que aparecen en la siguiente tabla:

	<b>MSE</b>
<b>MEDIA</b>	0,0178

Comparando las medias obtenidas en esta prueba (prueba 2) con las obtenidas en la prueba 1, se ve que teniendo una sola capa, si reduzco el número de neuronas de la capa, los resultados empeoran, por tanto, de momento, me quedo con la topología de la prueba 1.

### 5.3- Prueba 3

Para esta prueba, voy a probar a aumentar el número de neuronas respecto de la prueba 1. Los datos para esta prueba son:

<b>Nº Capas</b>	1
<b>Neuronas Capa</b>	15

Y los resultados que obtengo son los siguientes:

	<b>MSE</b>
<b>MEDIA</b>	0,0189

Comparando con los resultados obtenidos en la primera prueba (que es la mejor hasta ahora), se observa que aumentando el número de neuronas de la capa, no se mejoran los resultados, por lo tanto, la topología con la que nos quedamos momentáneamente es la de la primera prueba.

Como continuar aumentando el número de neuronas de la capa, no mejorará los resultados, lo que voy a probar ahora es a añadir una capa más (ahora tendré 2 capas) e ir aumentando y/o disminuyendo el número de neuronas de las capas para ver si los resultados que voy obteniendo mejoran mi mejor topología hasta el momento.

### 5.4- Prueba 4

Para esta prueba voy a usar los siguientes datos:

<b>Nº Capas</b>	2
<b>Neuronas Capa 1</b>	10
<b>Neuronas Capa 2</b>	10

He decidido empezar con estos datos, porque 10 es el número de neuronas con el que empecé la primera prueba.

Los datos obtenidos en esta prueba son los siguientes:

	<b>MSE</b>
<b>MEDIA</b>	0,0222

Comparando con la topología de la prueba 1, que era la mejor hasta el momento, se observa que añadiendo 2 capas con 10 neuronas cada capa, no se mejoran los resultados; por tanto, mantenemos la anterior topología.

Aunque añadiendo 2 capas con 10 neuronas cada capa no se mejoran los resultados obtenidos, voy a probar a aumentar y/o disminuir el número de neuronas de ambas capas para ver si alguna posible combinación puede mejorar los resultados de la primera prueba.

### 5.5- Prueba 5

Voy a disminuir el número de neuronas de la primera capa:

<b>Nº Capas</b>	2
<b>Neuronas Capa 1</b>	5
<b>Neuronas Capa 2</b>	10

Viendo los resultados de la tabla que pongo a continuación, se puede ver como disminuir el número de neuronas de la primera capa, tampoco mejora los resultados:

	<b>MSE</b>
<b>MEDIA</b>	0,0267

No obstante, voy a probar a aumentar y disminuir el número de neuronas de la segunda capa y a mantener a 10 el número de neuronas de la primera para ver lo que sucede.

### 5.6- Prueba 6

Voy a mantener las neuronas de la capa 1 y a aumentar las de la capa 2:

<b>Nº Capas</b>	2
<b>Neuronas Capa 1</b>	10
<b>Neuronas Capa 2</b>	5

Los resultados que he obtenido son los siguientes:

	<b>MSE</b>
<b>MEDIA</b>	0,0244

Y como se puede ver, tampoco mejora la topología que ya tenía.

### 5.7- Prueba 7

Por último, voy a probar a aumentar el número de neuronas de la segunda capa a 15.

<b>Nº Capas</b>	2
<b>Neuronas Capa 1</b>	10
<b>Neuronas Capa 2</b>	15

Los resultados de esta prueba son:

	<b>MSE</b>
<b>MEDIA</b>	0,0167

Como se ve, tampoco mejoran los resultados de la topología de la prueba 1.

### 5.8- Conclusión

Viendo los resultados de las pruebas, se puede determinar que la mejor topología para la red es:

<b>Nº Capas</b>	1
<b>Neuronas Capa 1</b>	10

Obteniendo como resultado:

<b>Nº Ensayo</b>	<b>MSE</b>	<b>%Aciertos</b>
<b>MEDIA (10 ensayos)</b>	<b>0,0144</b>	<b>98</b>

## 6.- Obtención del mejor tipo de entrenamiento (3 neuronas)

Con la mejor topología obtenida en las pruebas anteriores, voy a realizar ahora una batería de pruebas, cambiando el tipo de entrenamiento en cada una de las pruebas para ver cuál es el entrenamiento con el que la mejor topología obtiene los mejores resultados.

La topología será:

Nº Capas	1
Neuronas Capa	10

### 6.1 – Trainlm

Tipo de entrenamiento	trainlm
-----------------------	---------

Este entrenamiento es: Levenberg-Marquardt backpropagation

Los resultados que he obtenido son (los mismos que en la obtención de la topología ya que usé este entrenamiento como entrenamiento base para todas las pruebas anteriores):

	MSE
MEDIA	0,0144

### 6.2 – Traingdx

Tipo de entrenamiento	traingdx
-----------------------	----------

Este entrenamiento es: Gradient descent with momentum and adaptive learning rate backpropagation.

Los resultados con este tipo de entrenamiento son peores, como se puede observar en la siguiente tabla:

	MSE
MEDIA	0,0256

Como se puede ver, no mejoran a los resultados obtenidos con el tipo de entrenamiento “trainlm”.



### 6.3 – *Traincgp*

Tipo de entrenamiento	traincgp
-----------------------	----------

Este tipo de entrenamiento es: Conjugate gradient backpropagation with Polak-Ribière updates (restarts).

Los resultados que obtengo, y que una vez más no mejoran a los obtenidos con trainlm son:

	<b>MSE</b>
<b>MEDIA</b>	0,0156

Como se puede ver, no mejoran a los resultados obtenidos con el tipo de entrenamiento “trainlm”.

### 6.4 – *Trainbfg*

Tipo de entrenamiento	trainbfg
-----------------------	----------

Este tipo de entrenamiento es: BFGS quasi-Newton backpropagation.

Con él, tampoco se mejoran los resultados obtenidos hasta el momento con trainlm:

	<b>MSE</b>
<b>MEDIA</b>	0,0189

Como se puede ver, no mejoran a los resultados obtenidos con el tipo de entrenamiento “trainlm”.

### 6.5 – *Trainrp*

Tipo de entrenamiento	trainrp
-----------------------	---------

Este tipo de entrenamiento es: Resilient backpropagation.

Los resultados obtenidos con este tipo de entrenamiento son:

	<b>MSE</b>
<b>MEDIA</b>	0,02

Como se puede ver, no mejoran a los resultados obtenidos con el tipo de entrenamiento “trainlm”.

## 6.6 – Trainscg

Tipo Entrenamiento	trainscg
--------------------	----------

Este tipo de entrenamiento es: Scaled conjugate gradient backpropagation

Los resultados que he obtenido usando este tipo de entrenamiento son:

	<b>MSE</b>
<b>MEDIA</b>	0,0178

Como se puede ver, no mejoran a los resultados obtenidos con el tipo de entrenamiento “trainlm”.

## 6.7 – Conclusiones

Como se puede ver en los resultados que he obtenido, el tipo de entrenamiento con el que obtengo mejores resultados es el entrenamiento **Levenberg-Marquardt** (“trainlm”):

Tipo de entrenamiento	trainlm
<b>MEDIA (MSE)</b>	0,0144

## 7. – Pruebas con la mejor red (3 neuronas)

Después de realizar todas las pruebas anteriores, he obtenido la siguiente red como la mejor:

<b>Nº Capas</b>	1
<b>Neuronas Capa</b>	10
<b>Tipo Entrenamiento</b>	trainlm

Con esta red voy a realizar una batería de 10 pruebas: mi programa PracticaIris.m incluye un bucle de 10 ensayos, por lo que el resultado que te da tras ejecutarlo es la media de los 10 ensayos; pero además, voy a ejecutar PracticaIris.m 10 veces, por lo que realmente habré realizado 100 ejecuciones, de esta manera, obtendré unos datos más fiables ya que el número de ejecuciones será bastante alto.

Los resultados que he obtenido al realizar estas pruebas son los que aparecen en la siguiente tabla:

Nº Ejecución	MSE	%Acierto
1	0,0183	96,3333
2	0,0144	98
3	0,0117	97,667
4	0,0117	97,667
5	0,0117	97,667
6	0,0189	96,3333
7	0,0144	98
8	0,0144	98
9	0,0117	97,667
10	0,0144	98
<b>MEDIA</b>	<b>0,01416</b>	<b>97,53346</b>