

Aprendizaje Automático

Redes de Neuronas Artificiales

Diego Diaz

Contenido

Introducción.....	2
1. Respuesta a las Actividades	2
Apartado 3.	2
3.1. ¿Por qué el menor error se debería obtener en el primer experimento?.....	2
3.2. ¿Por qué fluctúa el error en los distintas ejecuciones del segundo experimento?.....	3
3.3. ¿Por qué se dice que el resultado más fiable es el obtenido en el tercer experimento, a continuación el obtenido en el segundo y el menos fiable corresponde al primero?	3
3.4. Indique con qué conjunto o subconjunto de datos se construye el modelo de RNA mostrado por Weka en cada uno de los experimentos realizados.	3
3.5. Interprete los resultados de la matriz de confusión obtenida en el experimento 3.	4
Apartado 4.	4
Apartado 5	5
5.1. Estudio de RNA resultantes con diferentes capas ocultas.....	5
5.2. Para la RNA ganadora del apartado anterior, aplique distintos valores (10, 100, 500, 1000 y 10000) al parámetro trainingTime.....	6
5.3. Para la RNA ganadora del apartado anterior, aplique distintos valores (0.001, 0.01, 0.1, 0.3, 1 y 10) al parámetro learningRate.....	7
5.4. Para la RNA ganadora del apartado anterior, aplique distintos valores (0.001, 0.01, 0.1, 0.2 y 1) al parámetro momentum.	7
Apartado 7	8
Apartado 8	9
Apartado 9	10
2. Conclusiones	10

Introducción.

En este documento se encuentran las respuestas a la primera prueba de evaluación a distancia (PED 1) de la asignatura *Aprendizaje Automático* de la UNED del curso 2015-2016.

El enunciado de las preguntas se encuentra en el documento *Practica_RNAs*. Dicho enunciado no se incluye en este documento.

1. Respuesta a las Actividades

Apartado 3.

Experimento 1 = Use Training.

Experimento 2 = Percentage split.

Experimento 3 = Cross-validation.

La **tabla 1** representa el tanto por ciento de instancias correctamente clasificadas para cada experimento realizado.

Experimento	1	2 (R=1)	2 (R=2)	2 (R=3)	2 (R=4)	2 (R=5)	3
% Instancias correctamente clasificadas	98.6667	98.0392	98.0392	94.1176	96.0784	90.1961	97.3333

Tabla 1. Porcentaje de instancias correctamente clasificadas en los experimentos 1,2 y 3.

3.1. ¿Por qué el menor error se debería obtener en el primer experimento?

Esta opción evalúa el clasificador sobre el mismo conjunto de datos con el que se entrenó, por lo que se crea un sobreajuste en la clasificación de los datos.

En la **figura 1** a la derecha se puede observar que el emplear la línea verde como clasificador se adapta mejor a los datos con los que hemos entrenado al clasificador, pero está demasiado adaptada a ellos, de forma que ante nuevos datos probablemente arrojará más errores que la clasificación usando la línea negra.¹

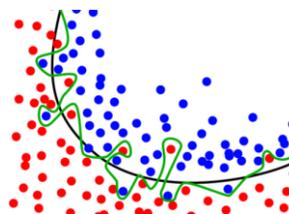


Figura1. Ejemplo de representación con Use Training

¹ «Overfitting» de Chabacano - Trabajo propio. Disponible bajo la licencia GFDL vía Wikimedia Commons - <https://commons.wikimedia.org/wiki/File:Overfitting.svg#/media/File:Overfitting.svg>

3.2. ¿Por qué fluctúa el error en los distintas ejecuciones del segundo experimento?

El experimento 2 se realiza varias veces con diferentes valores de R (Random seed for XVal / % Split). Este valor R, cambia el valor de la semilla responsable de obtener diferentes particiones de los conjuntos entrenamiento/validación.

En la retropropagación del error, el error para cada ejemplo se calcula como la diferencia entre el valor obtenido (para cada salida de la red) y el deseado (ya que estamos en aprendizaje supervisado sabemos el valor deseado). En función del error se van recalculando los pesos hacia atrás. Al cambiar las particiones de los ejemplos provoca que el entrenamiento y la validación se hagan con datos diferentes que provocan que el error fluctúe como se puede observar en la **tabla 2**.

	R=1	R=2	R=3	R=4	R=5
Mean absolute error	0,0239	0,0235	0,0404	0,031	0,0676
Root mean squared error	0,1101	0,0977	0,1651	0,1349	0,229

Tabla 2. Fluctuación del error con los diferentes valores del parámetro R.

3.3. ¿Por qué se dice que el resultado más fiable es el obtenido en el tercer experimento, a continuación el obtenido en el segundo y el menos fiable corresponde al primero?

En la primera opción (use training) se evalúa el clasificador sobre el mismo conjunto de datos con los que se efectuó el entrenamiento. El resultado de evaluación con esta opción puede ser demasiado optimista debido al posible sobreajuste a los datos de entrenamiento (cuanto mayor es el entrenamiento mayor es el desajuste).

La segunda opción (percentage split) permite construir el clasificador a partir de un subconjunto de datos extraído del conjunto de datos original y evaluar dicho clasificador en función del subconjunto restante que no intervino en su construcción. Cuando el número de instancias es suficientemente elevado, esta opción es suficiente para estimar con precisión las prestaciones del clasificador. El porcentaje de instancias correctamente clasificadas es peor que en el ejemplo 1 debido a que la muestra no es muy grande, pero la clasificación está menos afectada por el sobreajuste.

En el tercer ejemplo (cross validation), que es el más fiable, el tipo de validación consiste en dividir el conjunto de instancias de entrenamiento en tantas carpetas como se indica en el campo *Folds* y se realizan tantas evaluaciones como número de carpetas existentes. En cada evaluación se toman las instancias de una carpeta (diferente cada vez) como datos de test y las instancias del resto de carpetas como datos de entrenamiento para construir el clasificador. El error final corresponde al promedio de los errores obtenidos en cada iteración y el clasificador resultante es aquel con el que mejores prestaciones se obtuvo es por ello que este tipo de validación es muy aconsejable cuando el tamaño del conjunto de entrenamiento no es muy grande, como en el caso concreto que se analiza.

3.4. Indique con qué conjunto o subconjunto de datos se construye el modelo de RNA mostrado por Weka en cada uno de los experimentos realizados.

Esta pregunta se complementa con lo indicado en el apartado anterior y de un modo esquemático se indica cómo se utilizan los datos disponibles para construir el modelo de RNA.

Experimento 1: Evalúa el clasificador sobre el mismo conjunto de datos con el que se entrenó.

Experimento 2: 99 instancias para el entrenamiento (66%) y 51 para el test.

Experimento 3: Las instancias están divididas en tantas carpetas como se indica en el campo *Folds* (10 en este caso).

3.5. Interprete los resultados de la matriz de confusión obtenida en el experimento 3.

La matriz de confusión presentada en la **Figura 2** indica que las 50 instancias que pertenecen realmente a la clase *Iris-setosa* fueron clasificadas correctamente. En cambio, de las 50 (48+2) instancias perteneciente realmente a la clase *Iris-versicolor*, 48 fueron clasificadas correctamente y 2 de ellas se clasificó erróneamente en la clase *Iris-virginica*. Finalmente, de las 50 (48+2) instancias pertenecientes realmente a la clase real *Iris-virginica*, 48 fueron clasificadas correctamente y 2 de ellas fueron mal clasificadas asignándose a la clase *Iris-versicolor*.

Matriz de confusión			
a	b	c	Clasificada como
50	0	0	a = <i>Iris-setosa</i>
0	48	2	b = <i>Iris-versicolor</i>
0	2	48	c = <i>Iris-virginica</i>

Figura 2. Representación de la matriz de confusión experimento 3.

Apartado 4.

En la **figura 3** se puede ver el modelo de RNA del experimento 3 del apartado 3. Cada uno de los nodos está etiquetado con un número de nodo. La capa oculta la forman los nodos 3, 4 y 5. La capa de salida está formada por los nodos 0, 1 y 2.

Se han elegido los nodos 3 y 0 para mostrar los pesos de las conexiones que llegan a cada uno de estos nodos con la información obtenida del modelo en Weka.

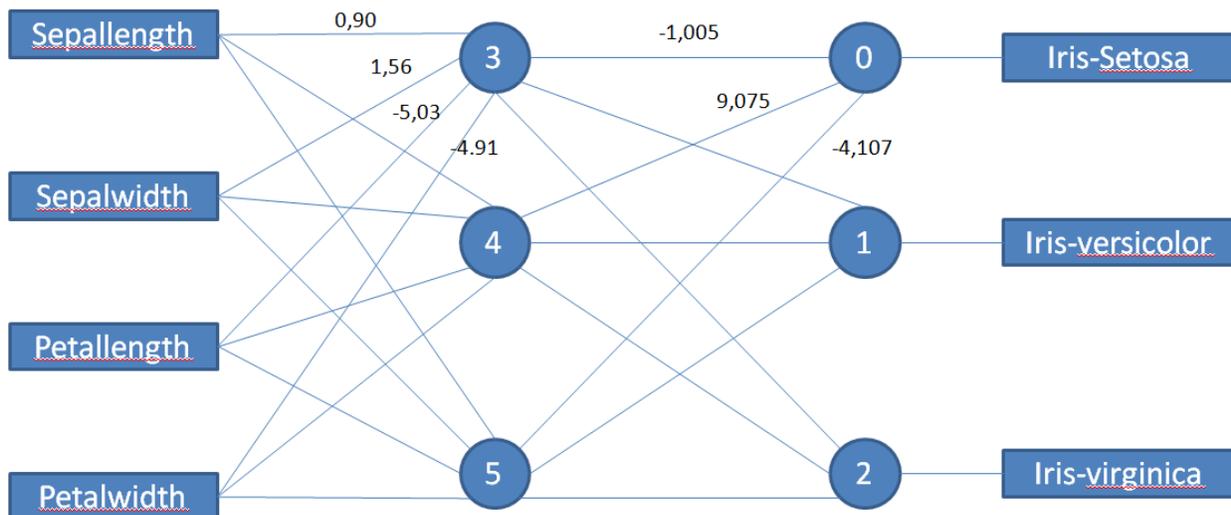


Figura 3. Modelo de RNA del experimento 3. Los nodos 0 y 3 están etiquetados con los pesos de las conexiones que llegan a cada uno de ellos.

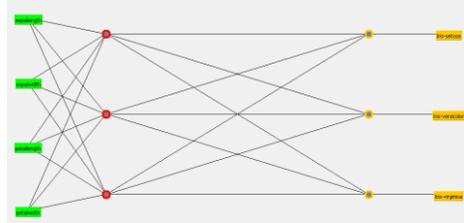
El parámetro Threshold es el umbral que se utiliza para controlar el error en el nodo en cada iteración del entrenamiento. Con el parámetro validationThreshold indicaremos el número de veces seguidas que permitiremos que el error empeore, en dicho caso, se termina el entrenamiento.

Apartado 5

5.1. Estudio de RNA resultantes con diferentes capas ocultas.

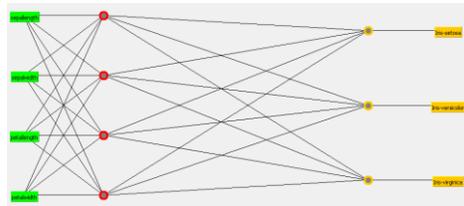
5.1.1. Valor del parámetro capas ocultas (Hidden Layers) utilizado: a

Número de capas ocultas: 1
Neuronas en la capa oculta: 3



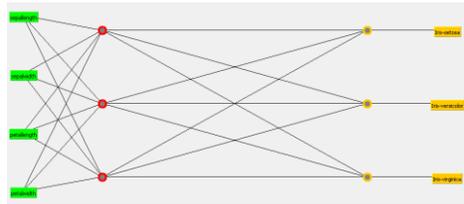
5.1.2. Valor del parámetro capas ocultas (Hidden Layers) utilizado: i

Número de capas ocultas: 1
Neuronas en la capa oculta: 4



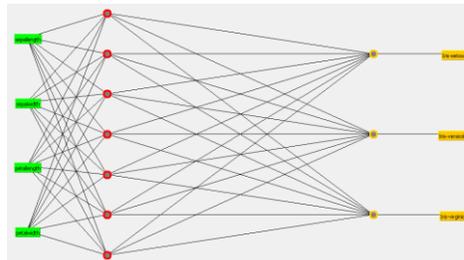
5.1.3. Valor del parámetro capas ocultas (Hidden Layers) utilizado: o

Número de capas ocultas: 1
Neuronas en la capa oculta: 3



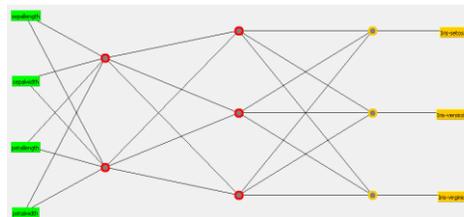
5.1.4. Valor del parámetro capas ocultas (Hidden Layers) utilizado: t

Número de capas ocultas: 1
Neuronas en la capa oculta: 7



5.1.5. Valor del parámetro capas ocultas (Hidden Layers) utilizado: 2,3

Número de capas ocultas: 2
Neuronas en la capa oculta 1: 2
Neuronas en la capa oculta 2: 3

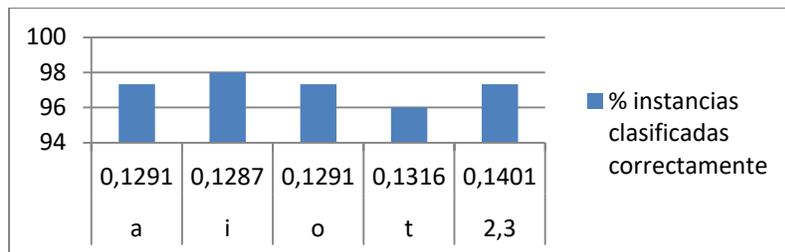


5.1.6. Porcentaje de instancias clasificadas correctamente y el RMSE (root mean square error) según los parámetros de capas ocultas (a, i, o, t, {2,3}).

RMSE (Root mean square error) es un indicador que indica si se ha aprendido bien el modelo. Es una medida que indica la diferencia entre los valores de la muestra (valores de la población) predichos por un modelo o un estimador y los valores realmente observados.

Hidden Layers	a	i	o	t	2,3
RMSE	0,1291	0,1287	0,1291	0,1316	0,1401
% instancias clasificadas correctamente	97,3333	98	97,3333	96	97,3333

Tabla 4. Comparación de los valores de RMSE con el porcentaje de instancias clasificadas correctamente.



Gráfica 1. Porcentaje de instancias clasificadas correctamente y el RMSE según valor de capas ocultas (a, i, o, t, {2,3}).

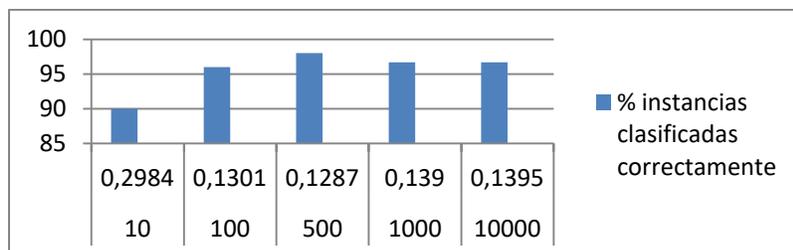
Como se observa en la **Gráfica 1**, la RNA ganadora es la que tiene el menor error, en este caso corresponde con el valor i de la capa oculta (1 capa oculta y cuatro neuronas en dicha capa).

5.2. Para la RNA ganadora del apartado anterior, aplique distintos valores (10, 100, 500, 1000 y 10000) al parámetro trainingTime.

El trainingTime son las épocas que queremos para finalizar el entrenamiento, es decir, el número de veces que se pasará el conjunto de entrenamiento.

trainingTime	10	100	500	1000	10000
RMSE	0,2984	0,1301	0,1287	0,139	0,1395
% instancias clasificadas correctamente	90	96	98	96,6667	96,6667

Tabla 5. Valores de RMSE con el porcentaje de instancias clasificadas correctamente.



Gráfica 2. Porcentaje de instancias clasificadas correctamente y el RMSE según valor de las épocas de entrenamiento (10,100,500,1000,10000).

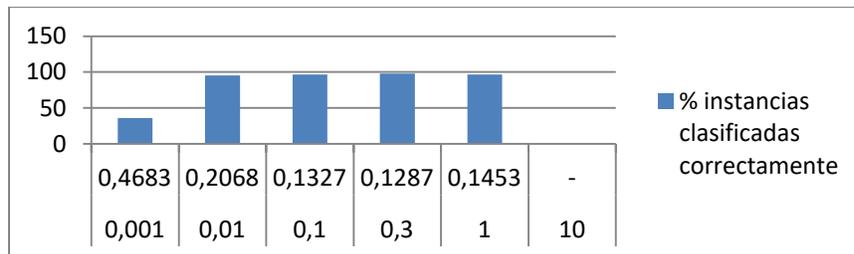
Como se observa en la **tabla 5 y en la gráfica 2**, aunque pasemos un mayor número de veces el entrenamiento, no conseguimos una mejora del error ni en el porcentaje de instancias clasificadas correctamente, por este motivo, la RNA ganadora es la que utiliza 500 épocas.

5.3. Para la RNA ganadora del apartado anterior, aplique distintos valores (0.001, 0.01, 0.1, 0.3, 1 y 10) al parámetro learningRate.

El Learning Rate es la tasa de aprendizaje, especifica el porcentaje en que se permite varíen los pesos de la red en cada época de entrenamiento. Cuanto mayor sea este valor, mayor peso tendrá el conjunto de entrenamiento utilizado. La tasa máxima tendrá valor 1 (el último valor LearningRate no es válido).

LearningRate	0,001	0,01	0,1	0,3	1	10
RMSE	0,4683	0,2068	0,1327	0,1287	0,1453	-
% instancias clasificadas correctamente	36	95,3333	96,6667	98	96,6667	-

Tabla 6. Valores de RMSE con el porcentaje de instancias clasificadas correctamente.



Gráfica 3. Porcentaje de instancias clasificadas correctamente y el RMSE según valor de la tasa de aprendizaje ({0,001}, {0,01}, {0,1}, {0,3}, {1}, {10}).

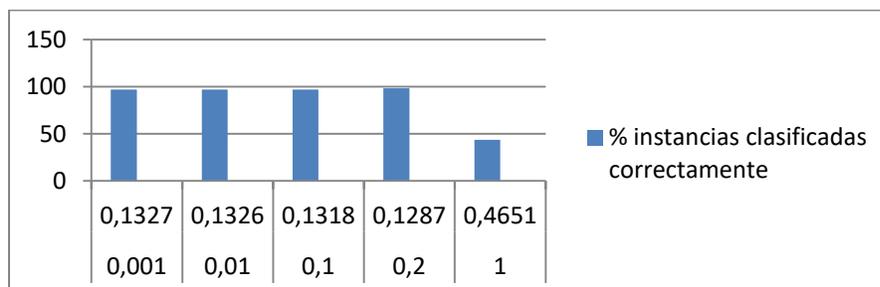
Si la tasa de aprendizaje vale 0,001 supone un lento ajuste de los valores de los pesos provocando la caída de mínimos locales de los que no es posible salir. De los valores restantes, el valor de compromiso en la comparación con el RMSE coincide con el valor por defecto 0,3 como se observa en la **tabla 6**.

5.4. Para la RNA ganadora del apartado anterior, aplique distintos valores (0.001, 0.01, 0.1, 0.2 y 1) al parámetro momentum.

El momentum determina que la tasa de aprendizaje disminuya con el tiempo, lo que provoca que los nuevos valores utilizados en el aprendizaje pierdan peso frente a los anteriores. El momentum tendrá el valor máximo 1 ya que si no provocaría que la tasa de aprendizaje aumentara en vez de disminuirla.

momentum	0,001	0,01	0,1	0,2	1
RMSE	0,1327	0,1326	0,1318	0,1287	0,4651
% instancias clasificadas correctamente	96,6667	96,6667	96,6667	98	43,3333

Tabla 7. Valores de RMSE con el porcentaje de instancias clasificadas correctamente.



Gráfica 4. Porcentaje de instancias clasificadas correctamente y el RMSE según valor de las épocas de entrenamiento (10,100,500,1000,10000).

En la **gráfica 4** se puede observar que manteniendo el valor de la tasa de aprendizaje provoca una caída importante en la clasificación ya que valores altos de tasa de aprendizaje favorecen el acercamiento

rápido a los valores óptimos de los pesos, pero no permiten el ajuste fino de estos pesos provocando un movimiento continuo alrededor del óptimo. El mejor resultado se obtiene con el momentum igual a 0,2.

Apartado 7

Predicción de la clasificación de un conjunto de ejemplos de clasificación desconocida almacenados en el fichero "iris-new.arff".

Se genera un nuevo archivo en el que el contenido coincide con el archivo original "iris-new.arff" con la salvedad de que se ha añadido un nuevo atributo, denominado *predictedClass* que muestra la clasificación predicha por el modelo para cada ejemplo.

```
@relation iris_predicted
```

```
@attribute sepallength numeric
```

```
@attribute sepalwidth numeric
```

```
@attribute petallength numeric
```

```
@attribute petalwidth numeric
```

```
@attribute predictedclass {Iris-setosa,Iris-versicolor,Iris-virginica}
```

```
@attribute class {Iris-setosa,Iris-versicolor,Iris-virginica}
```

```
@data
```

```
5.1,3,1.3,0.1,Iris-setosa,?
```

```
6,3,4,1,Iris-versicolor,?
```

```
6.5,3.3,4.2,1.2,Iris-versicolor,?
```

```
7.1,3,6,2.5,Iris-virginica,?
```

```
4.8,3.6,1.3,0.2,Iris-setosa,?
```

```
5.9,2.6,4.6,1.4,Iris-versicolor,?
```

```
6.2,2.9,4.2,1.5,Iris-versicolor,?
```

```
4.9,3.5,1.2,0.2,Iris-setosa,?
```

```
4.5,3.3,1.4,0.5,Iris-setosa,?
```

```
7.1,3.5,5.8,2.1,Iris-virginica,?
```

```
6.3,3.2,5.1,1.8,Iris-virginica,?
```

```
5.1,3.4,1.2,0.4,Iris-setosa,?
```

```
6.1,3.2,4.1,1.1,Iris-versicolor,?
```

```
6,2.7,6.1,1.9,Iris-virginica,?
```

```
6.7,3.5,5.5,2.2,Iris-virginica,?
```

Apartado 8

En la **tabla 8**, vemos el resultado de predecir la clase de los ejemplos del archivo "iris-new.arff"² utilizando un comando de Weka por consola.

inst#	actual	predicted	error prediction ()
1	1:?	1:Iris-set	0.994
2	1:?	2:Iris-ver	0.976
3	1:?	2:Iris-ver	0.98
4	1:?	3:Iris-vir	0.999
5	1:?	1:Iris-set	0.995
6	1:?	2:Iris-ver	0.973
7	1:?	2:Iris-ver	0.983
8	1:?	1:Iris-set	0.995
9	1:?	1:Iris-set	0.993
10	1:?	3:Iris-vir	0.998
11	1:?	3:Iris-vir	0.716
12	1:?	1:Iris-set	0.994
13	1:?	2:Iris-ver	0.977
14	1:?	3:Iris-vir	0.999
15	1:?	3:Iris-vir	0.998

Tabla 8. Salida obtenida por consola

Ya que solamente son 15 instancias, las cuales se dividen en 10 carpetas en las que se realizarán 10 evaluaciones diferentes, el error final corresponde al promedio de los errores obtenidos en cada iteración y el clasificador resultante es aquel con el que mejores prestaciones se obtuvo. Por este motivo, la predicción del error es muy alta si lo comparamos con la prueba hecha con el archivo iris.arff.

El paso más adecuado a seguir sería evaluar las prestaciones del modelo generado a partir de otro conjunto de datos ajeno a la construcción del modelo y de los que esta vez sí se conoce su clasificación. Esto se hace en el siguiente apartado.

² Mediante el uso del comando `java weka.classifiers.functions.MultilayerPerceptron -p 5 -l directorypath\RNA.model -T directorypath\iris-new.arff` desde la línea de comandos de Weka.

Apartado 9

¿Se puede afirmar que la estimación del error mediante validación cruzada es comparable con el error obtenido al aplicar el modelo a un conjunto de datos no utilizado en el entrenamiento?

Si se compara el RMSE obtenido al aplicar el modelo al conjunto de datos del archivo "iris-test.arff" como se muestra en la **tabla 9**, se puede observar que el error afecta más al valor de RMSE en la nueva clasificación de iris-test.arff, algo que parece lógico ya que solamente se disponen de 15 instancias y un error afectará en mayor medida a la estadística.

	Iris.arff	Iris-test.arff
RMSE	0,1287	0.1999
% instancias clasificadas correctamente	98	93.3333 %

Tabla 9. Comparación de RMSE obtenido con el mismo modelo en los archivos iris.arff e iris-test.arff.

Se ha obtenido un error de 15 en la clasificación con los nuevos datos. Si se comparan los valores de muestra predichos con los observados, podemos ver tal como indica en la Matriz de confusión de la **figura 4**, que de las 5 (4+1) instancias perteneciente realmente a la clase *Iris-versicolor*, 4 fueron clasificadas correctamente y 1 de ellas se clasificó erróneamente en la clase *Iris-setosa*. El resto de las instancias fueron clasificadas correctamente.

Matriz de confusión			
a	b	c	Clasificada como
5	0	0	a = Iris-setosa
1	4	0	b = Iris-versicolor
0	0	5	c = Iris-virginica

Figura 4. Matriz de confusión resultante de aplicar el modelo

Se puede concluir por tanto que el error obtenido es similar al que se obtiene con el conjunto de datos que sí se utilizaron para crear dicho modelo.

2. Conclusiones

Obtener un modelo de clasificación mediante el paradigma RNA resulta sencillo si se considera que el número de entrenamientos totales utilizado no ha sido excesivamente alto.

En el apartado 3 de esta práctica y partiendo de la base de que no se dispone de un número elevado de instancias, se ha considerado que la opción cross-validation era la más acertada.

En el apartado 4 se presenta un gráfico que representa la RNA con la opción cross-validation en la que se pueden ver las capas totales con sus neuronas. También se muestra como ejemplo los pesos de las conexiones que llegan a cada uno de estos nodos de acuerdo con la información obtenida del modelo en Weka.

En el apartado 5 se han estudiado los valores de RMSE (Root mean square error) y el porcentaje de instancias correctamente clasificadas, las cuales nos indican si se ha aprendido bien el modelo y nos sirven como indicadores para comparar diferentes valores en el número de épocas y capas utilizadas, la

tasa de aprendizaje y su decremento (momentum). En base de los resultados obtenidos elegimos un modelo de RNA final.

Posteriormente en el apartado 7, se ha utilizado el modelo de RNA, anteriormente guardado en el apartado anterior, para clasificar un conjunto de ejemplos de clasificación desconocida almacenados en el fichero "iris-new.arff". El modelo clasifica los elementos pero el error observado en dicha clasificación (apartado 8) es alto.

Por este motivo, para evaluar las prestaciones del modelo generado a partir de otro conjunto de datos ajeno a la construcción del modelo y de los que esta vez sí se conoce su clasificación, se utiliza el archivo "iris-test.arff".

Por último se comprueba entre la clasificación que se ha obtenido en el apartado 7/8 y el 9 ya que se trata de los mismos datos como se muestra en la **tabla 10**.³

Resultados predichos con el modelo (apartado 8)	Datos conocidos apartado 9
5.1,3,1.3,0.1,Iris-setosa,?	5.1,3.0,1.3,0.1,Iris-setosa
4.8,3.6,1.3,0.2,Iris-setosa,?	4.8,3.6,1.3,0.2,Iris-setosa
4.9,3.5,1.2,0.2,Iris-setosa,?	4.9,3.5,1.2,0.2,Iris-setosa
4.5,3.3,1.4,0.5,Iris-setosa,?	4.5,3.3,1.4,0.5,Iris-setosa
5.1,3.4,1.2,0.4,Iris-setosa,?	5.1,3.4,1.2,0.4,Iris-setosa
6,3,4,1,Iris-versicolor,?	6.0,3.0,2.5,0.5,Iris-versicolor
6.5,3.3,4.2,1.2,Iris-versicolor,?	6.5,3.3,4.2,1.2,Iris-versicolor
5.9,2.6,4.6,1.4,Iris-versicolor,?	5.9,2.6,4.6,1.4,Iris-versicolor
6.2,2.9,4.2,1.5,Iris-versicolor,?	6.2,2.9,4.2,1.5,Iris-versicolor
6.1,3.2,4.1,1.1,Iris-versicolor,?	6.1,3.2,4.1,1.1,Iris-versicolor
7.1,3,6,2.5,Iris-virginica,?	7.1,3.0,6.0,2.5,Iris-virginica
7.1,3.5,5.8,2.1,Iris-virginica,?	7.1,3.5,5.8,2.1,Iris-virginica
6.3,3.2,5.1,1.8,Iris-virginica,?	6.3,3.2,5.1,1.8,Iris-virginica
6.2,7,6.1,1.9,Iris-virginica,?	6.0,2.7,6.1,1.9,Iris-virginica
6.7,3.5,5.5,2.2,Iris-virginica,?	6.7,3.5,5.5,2.2,Iris-virginica

Tabla 10. Comparación de valores obtenidos (apartado 8) y conocidos (apartado 9).

Conclusiones finales:

- El modelo ha conseguido clasificar instancias nuevas (apartado 7/8).
- El modelo se evalúa con datos no utilizados en el entrenamiento (apartado 9) de los que sí se conoce su clasificación y el error obtenido es similar al que se obtiene con el conjunto de datos que sí se utilizaron para crear dicho modelo.
- Este paradigma resulta bastante efectivo como método de clasificación.

³ Exceptuando una instancia marcada en negrita.