Redes de neuronas artificiales

Grado en Ingeniería Informática



Universidad Carlos III of Madrid

**Práctica 02:**

**Problema de clasificación**

Ivan Miguelez García - 100383387

José Manuel Frías Salvachua - 100383533

# Introducción

Para esta práctica se quiere crear un modelo que permita distinguir los tipos de nubes que existen. En este caso, se dispone de un conjunto de datos con 717 instancias de nubes que han sido clasificadas en tres tipos: cielo despejado, nube y multinube.

# Preparación de los datos

Debido al número de datos tan pequeño, no se podía hacer una división con grupo de entrenamiento, validación y test y se ha tenido que recurrir a la Validación Cruzada. Para ello se crearon 4 folds iniciales y entremezclándolos se obtuvieron 4 grupos de Training y Test.

Antes de nada, se realizó una normalización de cada uno de los atributos para poder evitar sesgos a la hora de aprender.

Para la creación de los folds iniciales, se separaron los datos en cada una de las tres clases y se aleatorizó su orden. Después se dividió cada conjunto en 4 (el único en presentar algún problema fue Nube que tenía un resto de 1 instancia, pero se añadió al último fold) y se fue añadiendo a cada fold hasta quedarnos sin posibles instancias de esa clase. Eso se hizo por cada una de las tres clases hasta tener los folds con las instancias correspondientes. De esta forma los datos dentro de los folds quedan lo más balanceados posible entre ellos. Por último se realizó una aleatorización de cada uno de los folds para que las instancias fuesen presentando las distintas clases a la vez y no primero todas las instancias de una clase y luego de las siguientes, con el fin de facilitar el aprendizaje de la red.

Una vez terminada esta división en folds iniciales, se hizo combinatoria para juntar tres folds como conjuntos de entrenamiento y el fold sobrante como conjunto de test. De esta forma se consiguen los cuatro folds que serán utilizados para enseñar a la red a clasificar los tres tipos de cielo que se presentan en el conjunto: cielo despejado, nube y multinube.

# Experimentación con Perceptrón Multicapa

Para llevar a cabo la experimentación con el algoritmo de perceptrón multicapa se prueban distintas combinaciones de hiperparámetros. La *Tabla 1* muestra un resumen de estos resultados de forma creciente y se ha marcado en negrita el mejor modelo. Este orden ha sido establecido para mantener cierta claridad. Por esto mismo, se ha decidido incluir únicamente los accuracies obtenidos en test, ya que son los usados para elegir el mejor de los modelos.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Arquitectura | Razón de aprendizaje | FOLD1 | FOLD2 | FOLD3 | FOLD4 | Media Test | Ciclos |
| 10 | 0.01 | 0,7472 | 0,7584 | 0,7528 | 0,7318 | 0,7476 | 5000 |
| 0,7416 | 0,7416 | 0,7584 | 0,7430 | 0,7461 | 3500 |
| 0.05 | 0,7528 | 0,7584 | 0,7697 | 0,7598 | 0,7602 | 1000 |
| 0,7528 | 0,7640 | 0,7640 | 0,7542 | 0,7588 | 800 |
| 0.2 | 0,7191 | 0,7640 | 0,7640 | 0,7486 | 0,7489 | 500 |
| 0,7528 | 0,2753 | 0,7753 | 0,7765 | 0,6450 | 200 |
| 0.6 | 0,7360 | 0,7528 | 0,7472 | 0,7765 | 0,7531 | 250 |
| 0,7416 | 0,7753 | 0,7472 | 0,7654 | 0,7574 | 150 |
| **20** | 0.01 | 0,7416 | 0,7697 | 0,7921 | 0,7709 | 0,7686 | 10000 |
| 0,7416 | 0,7584 | 0,7753 | 0,7821 | 0,7644 | 8000 |
| **0.05** | 0,6910 | 0,7640 | 0,7022 | 0,7430 | 0,7251 | 5000 |
| **0,7416** | **0,7809** | **0,7809** | **0,7765** | **0,7700** | **2000** |
| 0.2 | 0,7528 | 0,7809 | 0,7753 | 0,7598 | 0,7672 | 1000 |
| 0,7303 | 0,7584 | 0,7753 | 0,7765 | 0,7601 | 300 |
| 0.4 | 0,7584 | 0,7640 | 0,7753 | 0,7598 | 0,7644 | 500 |
| 0,7303 | 0,7528 | 0,7697 | 0,7933 | 0,7615 | 250 |
| 40 | 0.2 | 0,7360 | 0,7472 | 0,7360 | 0,7765 | 0,7489 | 1500 |
| 0,7584 | 0,7584 | 0,7697 | 0,7877 | 0,7686 | 600 |
| 0.4 | 0,7528 | 0,7416 | 0,7303 | 0,7542 | 0,7447 | 1000 |
| 0,7528 | 0,7584 | 0,7753 | 0,7821 | 0,7672 | 200 |
| 0.6 | 0,7079 | 0,7809 | 0,7753 | 0,7821 | 0,7615 | 500 |
| 0,7528 | 0,7528 | 0,7753 | 0,7765 | 0,7644 | 150 |
| 0.8 | 0,7303 | 0,7753 | 0,7697 | 0,7486 | 0,7560 | 300 |
| 0,7472 | 0,7640 | 0,7753 | 0,2793 | 0,6415 | 100 |
| 60 | 0.05 | 0,7360 | 0,7978 | 0,7865 | 0,7542 | 0,7686 | 2500 |
| 0,7303 | 0,7528 | 0,7640 | 0,7765 | 0,7559 | 1500 |
| 0.2 | 0,7584 | 0,7584 | 0,7416 | 0,7654 | 0,7559 | 1500 |
| 0,7416 | 0,7753 | 0,7584 | 0,7654 | 0,7602 | 500 |
| 0.4 | 0,7303 | 0,7528 | 0,7640 | 0,7374 | 0,7462 | 1000 |
| 0,7472 | 0,7640 | 0,7809 | 0,7821 | 0,7686 | 200 |
| 10-10 | 0.001 | 0,2360 | 0,2360 | 0,0674 | 0,0670 | 0,1516 | 10000 |
| 0,2584 | 0,2697 | 0,2865 | 0,2737 | 0,2721 | 20000 |
| 0.05 | 0,6854 | 0,7528 | 0,7303 | 0,7877 | 0,7391 | 5000 |
| 0,7360 | 0,7584 | 0,7360 | 0,7709 | 0,7503 | 1000 |
| 0.2 | 0,6742 | 0,7360 | 0,7416 | 0,7039 | 0,7139 | 2000 |
| 0,2697 | 0,2809 | 0,7528 | 0,2737 | 0,3943 | 500 |
| 0.4 | 0,6685 | 0,7247 | 0,7360 | 0,7598 | 0,7222 | 1000 |
| 0,2697 | 0,7584 | 0,7472 | 0,7598 | 0,6338 | 300 |
| 20-20 | 0.01 | 0,7360 | 0,7472 | 0,7809 | 0,7654 | 0,7574 | 5000 |
| 0,7472 | 0,7584 | 0,7809 | 0,7542 | 0,7602 | 3000 |
| 0.05 | 0,7528 | 0,7753 | 0,7360 | 0,7821 | 0,7615 | 1000 |
| 0,7654 | 0,7640 | 0,2753 | 0,7472 | 0,6380 | 600 |
| 0.2 | 0,7528 | 0,7584 | 0,7528 | 0,7598 | 0,7560 | 500 |
| 0,2737 | 0,7640 | 0,2809 | 0,2697 | 0,3971 | 250 |

*Tabla 1*

Aunque la experimentación esté mostrada de forma secuencial, la experimentación fue realizada de manera ligeramente distinta. El orden de las capas sí se mantiene pero durante la ejecución se empezaba primero por una razón de aprendizaje de 0,2 y luego se iban probando distintas posibilidades. También se iniciaba siempre con un número de ciclos similar a los usados en anteriores experimentos y luego se revisaba las gráficas obtenidas para intentar obtener una segunda pasada con un número de ciclos que pareciesen mejores. Por ello todas las razones de aprendizaje tienen dos modelos, la estimación original y los ciclos consiguientes. Estos segundos ciclos se obtenían intentando ver lo que funcionaba mejor en las 4 gráficas aunque se puede comprobar fácilmente que no tenían porqué acabar siendo mejor que la primera estimación. Esto es debido a que lo que puede ser mejor para un fold puede ser desastroso para el resto (mirar resultados 20-20 con razón 0.2). Al no tener un conjunto de validación para elegir estos parámetros, esta fue la mejor opción

También se intentaba que las razones de aprendizaje fuesen las mismas entre los experimentos con distintas capas, aunque podrían sufrir algún ajuste en caso de que lo viésemos oportuno, ya sea porque creíamos que creciendo o decreciendo ese valor, alcanzaríamos un resultado mejor. Aún así, se intentó una estandarización de estas razones siendo las más usadas 0.2 (nuestro punto inicial), 0.05 (nuestro punto base por lo bajo) y 0.4 (aproximación para ver si crecer daba mejores resultados).

La *Tabla 2* muestra los mejores 10 resultados clasificados:

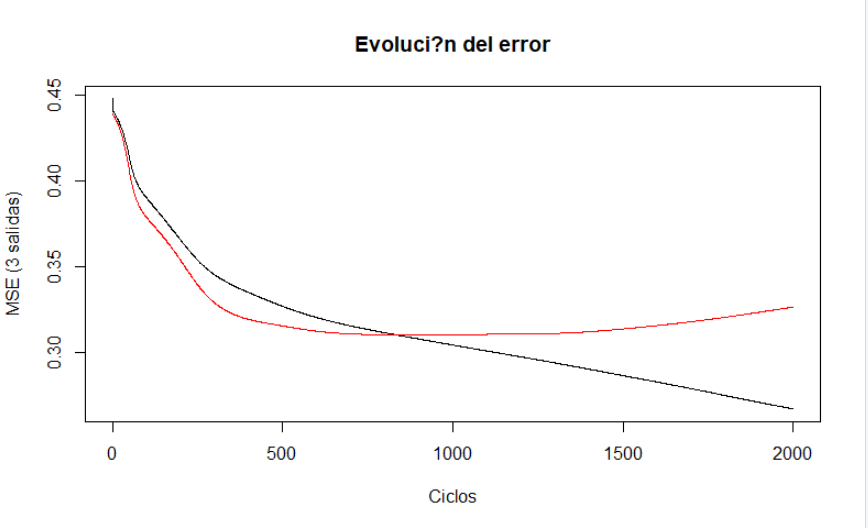
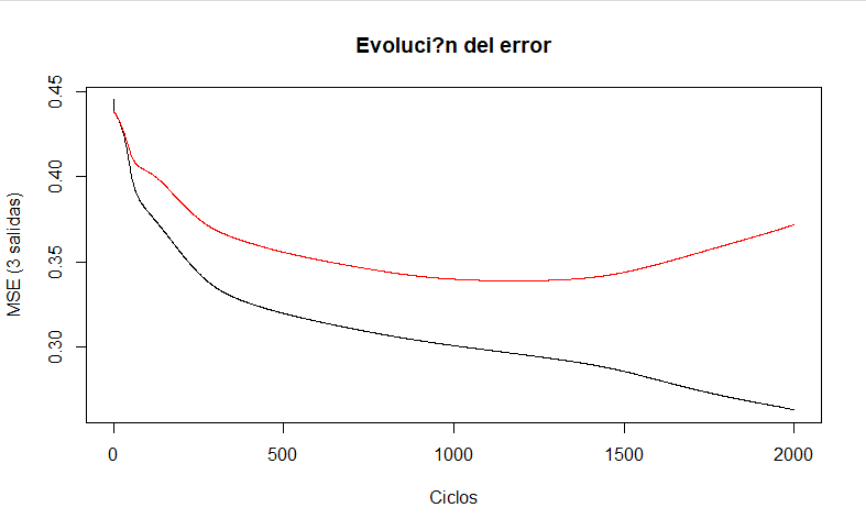
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Posición | Modelo | Media Test | Posición | Modelo | Media Test |
| 1 | 20 capas Razón 0.05 | **0,77** | 6 | 20 capas Razón 0.2 | 0,7672 |
| 2 | 20 capas Razón 0.01 | 0,7686 | 7 | 40 capas Razón 0.4 | 0,7672 |
| 3 | 40 capas Razón 0.2 | 0,7686 | 8 | 20 capas Razón 0.01 | 0,7644 |
| 4 | 60 capas Razón 0.05 | 0,7686 | 9 | 20 capas Razón 0.4 | 0,7644 |
| 5 | 60 capas Razón 0.4 | 0,7686 | 10 | 40 capas Razón 0.6 | 0,7644 |

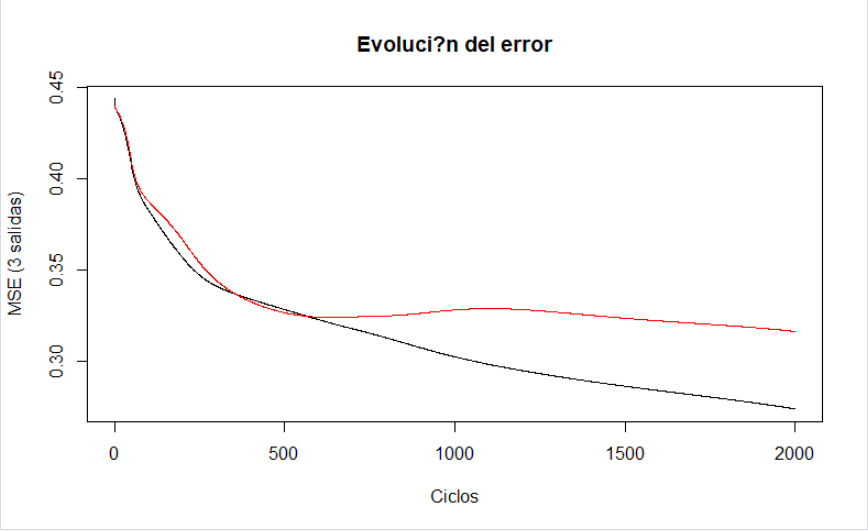
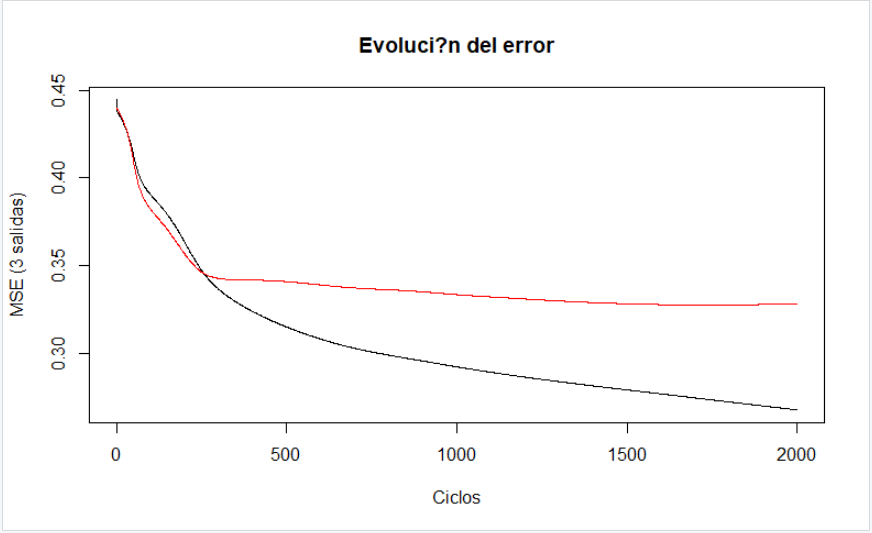
*Tabla 2*

En la experimentación original se obtenían hasta 10 decimales pero para la presentación de la tabla se han decidido mantener únicamente 4. Es por ello que vemos varias veces el mismo resultado ya que al redondear este se trunca. También podemos ver en la tabla que no aparece ningún modelo de más de una capa. Esto es algo que guió la experimentación ya que veíamos que no obteníamos ningún modelo de 2 capas que alcanzase las accuracies obtenidas en con los modelos de una capa; haciendo que desistiésemos en seguir ese camino.

La *Tabla 2* también sirve para indicarnos que razones de aprendizajes más pequeñas parecen ofrecer mejores resultados

Por último, en las siguientes gráficas podemos ver cómo ha sido la evolución del error en los distintos folds del mejor modelo. Es interesante ver cómo estos varían para formar un conjunto habiendo algunos que están entrando en overfitting mientras que otros folds aún necesitan unos cuantos ciclos más para poder estabilizarse.

Fold 1 Fold 2



Fold 3 Fold 4

Al revisar las gráficas de otros modelos, se observó que esto es algo que pasaba siempre, haciendo que la elección de unos ciclos óptimos para todos los folds sea algo casi imposible debido a la aleatoriedad en la creación de estos.

Lo siguiente sería ver las matrices de confusión de cada uno de los folds (las matrices de la izquierda son las de entrenamiento y las matrices de la derecha son las de test):

Fold 1

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Train | Cielo Despejado | Multi Nube | Nube |  | Test | Cielo Despejado | Multi Nube | Nube |
| Cielo Despejado | 34 | 0 | 1 |  | Cielo Despejado | 9 | 0 | 3 |
| Multi Nube | 0 | 50 | 67 |  | Multi Nube | 0 | 8 | 31 |
| Nube | 2 | 19 | 364 |  | Nube | 5 | 7 | 115 |

Fold 2

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Train | Cielo Despejado | Multi Nube | Nube |  | Test | Cielo Despejado | Multi Nube | Nube |
| Cielo Despejado | 32 | 0 | 4 | Cielo Despejado | 10 | 0 | 2 |
| Multi Nube | 0 | 33 | 84 | Multi Nube | 0 | 10 | 29 |
| Nube | 4 | 16 | 364 | Nube | 1 | 6 | 120 |

Fold 3

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Train | Cielo Despejado | Multi Nube | Nube |  | Test | Cielo Despejado | Multi Nube | Nube |
| Cielo Despejado | 33 | 0 | 3 | Cielo Despejado | 11 | 0 | 1 |
| Multi Nube | 0 | 42 | 75 | Multi Nube | 0 | 12 | 27 |
| Nube | 4 | 17 | 363 | Nube | 2 | 15 | 110 |

Fold 4

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Train | Cielo Despejado | Multi Nube | Nube |  | Test | Cielo Despejado | Multi Nube | Nube |
| Cielo Despejado | 30 | 0 | 5 | Cielo Despejado | 11 | 0 | 1 |
| Multi Nube | 0 | 37 | 80 | Multi Nube | 0 | 11 | 28 |
| Nube | 3 | 14 | 367 | Nube | 2 | 9 | 117 |

|  |  |
| --- | --- |
|  | Porcentaje de acierto total por clase |
| Cielo Despejado | 0.895 |
| Multi Nube | 0.325 |
| Nube | 0.938 |

Aquí se ve con claridad cómo afecta que en el conjunto de datos haya una clase mayoritaria, en este caso la clase Nube. En todas las matrices, vemos que al intentar clasificar Cielo Despejado o Multi Nube, se clasifica siempre de manera correcta o eligiendo la clase mayoritaria Nube; nunca intercambiando alguna de las clases Cielo Despejado por Multi Nube o viceversa. También se puede observar que CieloDespejado suele clasificarse extremadamente bien, quitando algunas pocas instancias que se desvían a la clase mayoritaria. Sin embargo, con Multi Nube presenta muchos problemas ya que hay más probabilidades de que sea asignada como la clase mayoritaria que de manera correcta.

Por último, Nube se suele clasificar de manera correcta la mayoría de ocasiones aunque algunas veces acabe siendo alguna de las otras dos clases (apareciendo más en Multi Nube que en Cielo Despejado).

# Conclusiones

Para terminar, se podría comentar que aunque el modelo obtenido es relativamente bueno (con accuracy general de 0,77) presenta un sesgo bastante importante y los resultados que ofrezca no deberán ser tomados como absolutos; sobretodo cuando ofrezca como resultado Multi Nube ya que, como se puede observar en las matrices de confusión, es existe una mediana posibilidad de que sea en realidad Nube. El resto de clases sin embargo, ofrecen una posibilidad de acierto bastante elevada por lo que no deberían dar problemas.

Como posible mejora, se podría partir de un conjunto de datos más grande que no se encuentre tan desbalanceado ya que el sesgo que esto produce se ha podido apreciar con claridad (problemas con Multi Nube). Para evitarlo se podría intentar conseguir nuevas instancias de las otras clases o realizar un sobre sampleo de ellas. No recomendaríamos un infrasampleo de la clase Nube ya que podríamos acabar con pocas instancias haciendo el aprendizaje algo extremadamente complicado.