REDES DE NEURONAS ARTIFICIALES

**REDES DE NEURONAS ARTIFICIALES:**

**PRÁCTICA 2 PARTE 1**

Pablo Salcedo

100408773

Jorge Sousa

100408805

Ángel Del Viejo Casillas

100405898

****

**ÍNDICE**

1. **INTRODUCCIÓN**

En este documento vamos a explicar todo lo que hemos realizado en la parte 1 de la práctica 2 de la asignatura. En esta primera parte vamos a tratar un problema de clasificación con el Perceptrón multicapa. Dado que todo el código de preparación de datos se nos ha dado hecho, solo vamos a realizar pruebas para encontrar los hiperparámetros que nos den el modelo con el mínimo fallo de validación y la mayor precisión.

1. **Parte 1: Perceptrón multicapa**
   1. **Pruebas**

Hemos utilizado la opción 2 de ModelCheckPoint para ejecutar todos los ciclos y poder garantizar que se encuentra el mejor modelo y guardarlo. Creemos que esta opción es mejor, ya que puede que en tras empeorar en varios ciclos consecutivos, al final de la ejecución mejore y si usamos EarlyStopping, habría parado antes y no habríamos encontrado este mejor modelo.

| **MODELO DE RED NEURONAL** | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Tasa de aprendizaje** | **N.º ciclos** | **loss** | **accuracy** | **mse** | **val\_loss** | **val\_accuracy** | **val\_mse** |
| **0.001** | 100 | 2.6377 | 0.0836 | 0.6108 | 0.6128 | 0.0807 | 0.6128 |
| **0.005** | 100 | 2.6017 | 0.0818 | 0.6121 | 0.6127 | 0.0807 | 0.6127 |
| **0.01** | 100 | 2.6342 | 0.0835 | 0.611 | 0.6127 | 0.0807 | 0.6127 |
| **0.05** | 100 | 2.6052 | 0.0819 | 0.612 | 0.6127 | 0.0807 | 0.6127 |
| **0.1** | 100 | 2.615 | 0.0834 | 0.611 | 0.6127 | 0.0807 | 0.6127 |
| **0.5** | 100 | 2.6265 | 0.0837 | 0.6108 | 0.6127 | 0.0807 | 0.6127 |

Para estos experimentos hemos usado los pesos que venían por defecto en el código, pero viendo los resultados nos damos cuenta de que deberíamos cambiarlos para intentar alcanzar resultados más concluyentes.

A pesar de que los pesos no sean los más apropiados podemos intuir que en esta práctica la tasa de aprendizaje no va a ser el factor más importante, sino que puede ser el número de capas, ya que aquí solo hemos usado 1 para intentar acotar la tasa de aprendizaje.

| **MODELO DE RED NEURONAL** | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Tasa de aprendizaje** | **N.º ciclos** | **N.º de capas** | **loss** | **accuracy** | **mse** | **val\_loss** | **val\_accuracy** | **val\_mse** |
| **0.0001** | 100 | 1 | 0.1108 | 0.8955 | 0.0529 | 0.0736 | 0.8596 | 0.0763 |
| 100 | 2 | 0.1176 | 0.8911 | 0.0522 | 0.0689 | 0.8632 | 0.0689 |
| 100 | 3 | 0.1526 | 0.863 | 0.066 | 0.0698 | 0.8561 | 0.0678 |
| 200 | 2 | 0.1479 | 0.8604 | 0.0695 | 0.0654 | 0.8456 | 0.0654 |
| **0.0005** | 100 | 1 | 0.1073 | 0.8911 | 0.0551 | 0.0707 | 0.8526 | 0.0707 |
| 100 | 2 | 0.1479 | 0.8604 | 0.0695 | 0.0654 | 0.8456 | 0.0654 |
| 100 | 3 | 0.2130 | 0.8183 | 0.0886 | 0.0687 | 0.8561 | 0.0687 |
| 200 | 2 | 0.1479 | 0.8604 | 0.0695 | 0.0654 | 0.8456 | 0.0654 |
| **0.001** | 100 | 1 | 0.1073 | 0.8911 | 0.0551 | 0.0707 | 0.8526 | 0.0707 |
| 100 | 2 | 0.1479 | 0.8604 | 0.0695 | 0.0654 | 0.8456 | 0.0654 |
| 100 | 3 | 0.2130 | 0.8183 | 0.0886 | 0.0687 | 0.8561 | 0.0687 |
| 200 | 2 | 0.1479 | 0.8604 | 0.0695 | 0.0654 | 0.8456 | 0.0654 |
| **0.005** | 100 | 1 | 0.1061 | 0.8964 | 0.0499 | 0.0735 | 0.8596 | 0.0735 |
| 100 | 2 | 0.1263 | 0.8771 | 0.0617 | 0.0707 | 0.8351 | 0.0707 |
| 100 | 3 | 0.1867 | 0.8428 | 0.0732 | 0.0837 | 0.8246 | 0.0837 |
| 200 | 2 | 0.1066 | 0.8955 | 0.0544 | 0.0464 | 0.9088 | 0.0464 |
| **0.01** | 100 | 1 | 0.1073 | 0.8911 | 0.0511 | 0.0707 | 0.8526 | 0.0707 |
| 100 | 2 | 0.1479 | 0.8604 | 0.0695 | 0.0654 | 0.8456 | 0.0654 |
| 100 | 3 | 0.2130 | 0.8183 | 0.0886 | 0.0687 | 0.8561 | 0.0687 |
| 200 | 2 | 0.1479 | 0.8604 | 0.0695 | 0.0654 | 0.8456 | 0.0654 |
| **0.05** | 100 | 1 | 0.1073 | 0.8911 | 0.0551 | 0.0707 | 0.8526 | 0.0707 |
| 100 | 2 | 0.1479 | 0.8604 | 0.0695 | 0.0654 | 0.8456 | 0.0654 |
| 100 | 3 | 0.2130 | 0.8183 | 0.0886 | 0.0687 | 0.8561 | 0.0687 |
| 200 | 2 | 0.1479 | 0.8604 | 0.0695 | 0.0654 | 0.8456 | 0.0654 |

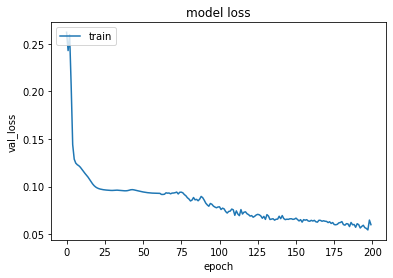
Analizando estos resultados vemos que todos los experimentos dan lo mismo según el número de capas salvo uno, con una tasa de 0.005, vamos a probar con tasas más extremas para analizar mejor el comportamiento del modelo.

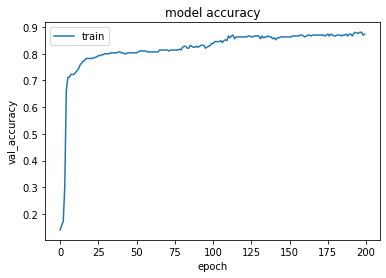
| **MODELO DE RED NEURONAL** | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Tasa de aprendizaje** | **N.º ciclos** | **N.º de capas** | **loss** | **accuracy** | **mse** | **val\_loss** | **val\_accuracy** | **val\_mse** |
| **0.0000005** | 100 | 2 | 0.1479 | 0.8604 | 0.0695 | 0.0654 | 0.8456 | 0.0654 |
| **1.5** | 100 | 2 | 0.1479 | 0.8604 | 0.0695 | 0.0654 | 0.8456 | 0.0654 |

Viendo esta tabla también, podemos concluir que en este modelo la tasa de aprendizaje no es el factor más importante, sino que es el número de capas, siendo 2 capas el óptimo.

**Gráficas de los mejores modelos:**

* Tasa de aprendizaje = 0.005
* Ciclos = 200
* Capas = 2



****

En la primera gráfica podemos ver como en los primeros ciclos la precisión aumenta en gran medida, ya que los primeros modelos deben de tener unos pesos muy lejanos a los correctos, por lo tanto, al ajustar y acercarse mejora considerablemente. También podemos ver como al final del experimento vemos que la curva va oscilando y eso es porque a pesar de haber encontrado ya los valores óptimos, como hemos elegido que no pare aunque empeore, sigue modificando los pesos buscando un modelo mejor, lo que hace que vaya oscilando.

En la segunda gráfica, correspondiente al error del modelo, pasa algo parecido a la precisión, en los primeros modelos, con unos pesos muy alejados de los correctos se tiene un fallo muy grande y a medida que se ajustan los pesos alcanzamos valores bastante más bajos, hasta llegar al valor mínimo, pero dado que no para aunque empeore el fallo, seguirá buscando modelos mejore, los cuales producirán las oscilaciones que se producen al final de la gráfica.

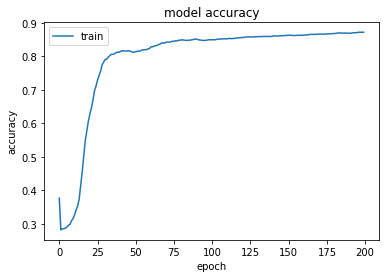
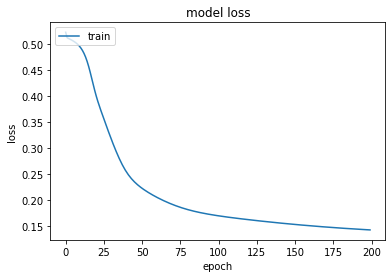
El mejor modelo que hemos encontrado es con una tasa de aprendizaje de **0.0005**, **2** capas y **200** ciclos. Vamos a usar estos valores para el modelo final y buscaremos el mejor número de neuronas. Además, vamos a usar los otros 2 mejores modelos de entrenamiento para poder alcanzar el mejor modelo final

| **MODELO FINAL** | | | | | | |
| --- | --- | --- | --- | --- | --- | --- |
| **Tasa de aprendizaje** | **N.º ciclos** | **N.º neuronas** | **loss** | **accuracy** | **mse** | **binary\_crossentropy** |
| **0.001** | **200** | 25 | 0.2644 | 0.8055 | 0.1139 | 0.4370 |
| 50 | 0.2542 | 0.8287 | 0.1014 | 0.4153 |
| 100 | 0.2488 | 0.8202 | 0.0972 | 0.4585 |
| 150 | 0.2416 | 0.8315 | 0.0943 | 0.4706 |
| **0.005** | **200** | 25 | 0.1485 | 0.8799 | 0.0594 | 0.3160 |
| 50 | 0.1664 | 0.8596 | 0.0672 | 0.4080 |
| 100 | 0.1730 | 0.8553 | 0.0688 | 0.5227 |
| 150 | 0.1759 | 0.8518 | 0.07 | 0.5670 |
| **0.01** | **200** | 25 | 0.1449 | 0.8912 | 0.0602 | 0.3102 |
| 50 | 0.1521 | 0.8764 | 0.0644 | 0.4434 |
| 100 | 0.1619 | 0.8652 | 0.0658 | 0.5430 |
| 150 | 0.1644 | 0.8624 | 0.0669 | 0.5842 |

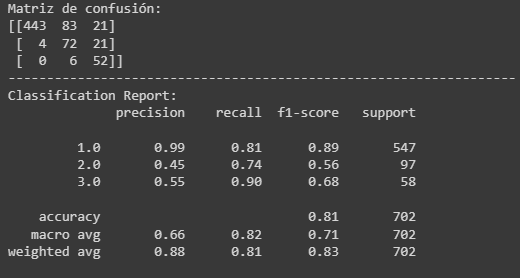
Analizando los resultados de esta tabla podemos concluir que la mejor tasa de aprendizaje es de 0.005 con 200 ciclos, 25 neuronas y 2 capas. También sacamos la conclusión de que el resultado sería mejor si aumentamos el número de ciclos porque el modelo final es un modelo que va mejorando constantemente.

**Gráfica del mejor modelo final con:**

* Tasa de aprendizaje = 0.005
* Ciclos = 200
* Capas = 2
* Neuronas = 25

****

Analizando estas gráficas podemos llegar a la misma conclusión que llegamos anteriormente analizando la tabla de resultado, este modelo mejoraría constantemente si aumentamos los ciclos que ejecuta, la mejora sería cada vez más pequeña, pero seguiría mejorando, tanto en la precisión como en la tasa de fallo.

****

Aquí tenemos la matriz de confusión y el classification report del mejor modelo.

En cuanto a la matriz de confusión:

* Tenemos un número muy alto de verdaderos positivos en la clase N, que son los 443 que fueron predichos correctamente. Destaca también que ningún valor ha sido predicho como clase P y que en realidad fueran clase N.
* También podemos observar un número considerable de valores de clase S que han sido predichos como clase N. El resto son valores bastante equilibrados para clase S y P donde se predice bien alrededor del 50%.

En cuanto al classification report:

* Hay valores muy altos para la clase N, el tipo 1, como podemos observar con el f1-score, que tiene en cuenta tanto la ‘precision’ como el ‘recall’. Mientras que para las otras dos clases tenemos valores bastante bajos, por encima del 50% pero muy lejos de la clase N.

1. **Conclusiones parte 1:**

Esta práctica nos ha parecido demasiado parecida a la parte 2 de la práctica 1, ya que se trabaja también con un perceptrón multicapa, la única diferencia es la introducción del parámetro de precisión para ver si realiza la clasificación correctamente, pero debido a su parecido se podría haber incluido como un sub-apartado en la anterior práctica.