

# Redes de Neuronas Artificiales

GRADO EN INGENIERÍA INFORMÁTICA

## **Práctica 2: Problema de clasificación con Perceptrón Multicapa y Redes Convolucionales**

### **Parte I**

**Curso 2021/2022**

Jorge Rodríguez Fraile, 100405951, Grupo 83, [100405951@alumnos.uc3m.es](mailto:100405951@alumnos.uc3m.es)  
Carlos Rubio Olivares, 100405834, Grupo 83, [100405834@alumnos.uc3m.es](mailto:100405834@alumnos.uc3m.es)

# Índice

<b>Introducción</b>	<b>3</b>
<b>Modelos generados y resultados</b>	<b>3</b>
<b>Análisis Mejor Modelo</b>	<b>6</b>
<b>Conclusiones</b>	<b>7</b>

# Introducción

En la primera parte de esta práctica vamos a resolver un problema de clasificación basado en la predicción de vehículos. Tendremos que variar los hiperparámetros de la red para encontrar el mejor modelo. El código se hará mediante Google Colab y se basará en la clasificación de 4 tipos de vehículos que se traducen en 4 clases: bus, Saab, van, Opel.

En cuanto a la red a utilizar, sabemos que tendremos 18 atributos de entrada, y como salida 4 neuronas (una por cada tipo de vehículo) que generarán números entre 0 y 1. El valor más cercano al 1 de estos 4 será el tipo de vehículo asignado a dicha entrada.

Podremos modificar el n.º de capas ocultas, las neuronas por capas ocultas, épocas y el factor de aprendizaje. A partir de estos campos podremos generar diferentes modelos y deberemos ajustarlos para resolver el problema de manera eficiente. Una vez obtenido el mejor modelo, pasaremos a analizarlo con métodos como una matriz de confusión y el error cuadrático medio con el conjunto de test.

## Modelos generados y resultados

En cuanto a los modelos generados hemos tenido en cuenta tres puntos importantes:

- **Utilización de datos:** Para el conjunto de datos que tenemos, dividiremos un 70 % de los mismos para entrenamiento y otro 30 % para test. Del conjunto de test se coge el 20 % para la validación durante la búsqueda de modelos y cuando se haya encontrado el modelo final se emplea el conjunto completo para entrenar.
- **Ajuste de parámetros:** Para ajustar los parámetros hemos seguido una especie de 'jerarquía' o 'lista de preferencia', primero queremos ajustar el n.º de neuronas por capa para después pasar a las capas ocultas y por último el factor de aprendizaje. Para el mejor modelo generado también ajustaremos los ciclos por época.
- **Resultados dados:** Para definir si un modelo es realmente bueno deberemos hallar las medias de los valores que obtenemos de la curva de validación y de la iteración en la que se obtiene el valor mínimo para una serie de ejecuciones del modelo.
- **Nomenclatura de los modelos:** Los modelos tendrán la nomenclatura del tipo 'LR\_EP\_N1\_N2\_N3' donde:
  - **LR:** Factor de aprendizaje
  - **EP:** Épocas
  - **N1:** Neuronas capa 1
  - **N2:** Neuronas capa 2
  - **N3:** Neuronas capa 3

- **Evaluación:** Nótese que debido a la extensión del problema sólo probaremos 3 capas ocultas en un modelo (como máximo), además, para cada modelo se harán 5 ejecuciones y se ejecutará la media sobre estos, que acabará siendo el valor que utilizamos.

Modelo	0	1	2	3	4	ValLoss Media
0.2_500_5_0_0	6.3983E-02	6.2677E-02	6.2518E-02	6.9217E-02	6.2624E-02	6.4204E-02
0.2_500_10_0_0	5.9950E-02	6.1248E-02	5.7570E-02	6.1247E-02	6.3876E-02	6.0778E-02
0.2_500_20_0_0	6.4039E-02	5.7739E-02	5.8910E-02	6.2367E-02	6.1790E-02	6.0969E-02
0.2_500_35_0_0	6.4301E-02	6.3932E-02	6.6315E-02	6.2477E-02	6.3534E-02	6.4112E-02
0.2_500_50_0_0	5.9557E-02	6.2223E-02	6.3183E-02	6.2316E-02	5.9078E-02	6.1271E-02
0.2_500_75_0_0	6.2972E-02	6.3896E-02	6.1935E-02	6.7051E-02	6.3680E-02	6.3907E-02
0.2_500_100_0_0	6.3108E-02	6.2411E-02	6.3722E-02	6.1772E-02	6.2529E-02	6.2708E-02
0.2_500_50_20_0	5.2153E-02	5.1103E-02	6.5189E-02	5.3601E-02	4.9769E-02	5.4363E-02
0.2_500_20_10_0	5.6835E-02	5.3768E-02	6.6304E-02	5.1255E-02	5.5593E-02	5.6751E-02
0.2_500_5_3_2	6.6286E-02	5.4365E-02	6.4831E-02	6.4817E-02	6.0979E-02	6.2256E-02
0.2_500_20_10_5	6.3181E-02	5.2795E-02	5.7001E-02	5.8884E-02	5.8939E-02	5.8160E-02
0.2_500_50_35_20	5.8773E-02	5.4864E-02	5.6710E-02	5.5650E-02	5.6513E-02	5.6502E-02
0.2_500_10_20_30	6.0629E-02	5.8460E-02	5.3923E-02	6.4554E-02	5.9944E-02	5.9502E-02
0.2_500_20_35_50	5.7463E-02	5.7769E-02	6.1880E-02	5.6420E-02	5.7396E-02	5.8186E-02
0.2_500_20_35_75	5.6212E-02	5.4311E-02	5.5169E-02	5.5455E-02	6.2656E-02	5.6761E-02

Tabla 1. Modelos generados ajustando neuronas y capas ocultas  
Loss: sparse categorical crossentropy

Los valores de la Tabla 1 se corresponden con el error cuadrático medio sobre el conjunto de validación para la mejor época del modelo.

Se puede ver que comenzamos fijando el factor de aprendizaje a 0.2 y con una sola capa de 5 neuronas. Vamos subiendo el número de neuronas hasta llegar a 100 y observamos que los valores de 10 neuronas y 50 eran bastante estables, siendo el de 10 el mejor.

Después de ese set de pruebas pasamos a añadir una capa más haciendo una especie de 'embudo' donde la primera capa tendrá más neuronas que la segunda. Probamos con 50 y 10 neuronas en la primera capa, y 20 y 10 en la segunda respectivamente. Vemos que el valor de 50 nos sigue dando un resultado muy bueno por lo que lo marcamos como posible candidato.

Seguimos con 3 capas y con la misma técnica del embudo, obtenemos resultados buenos, pero no tanto como con 2, por lo que seguimos probando.

Esta vez marcamos 3 capas y hacemos lo opuesto a las pruebas anteriores, el número de neuronas irá aumentando conforme vayamos pasando por las capas ocultas de la red. Obtenemos valores parecidos al anterior pero un poco más estables, aunque no lo tenemos muy en cuenta.

Modelo	0	1	2	3	4	ValLoss Media
0.03_3000_50_20_0	5.5396E-02	5.6936E-02	5.4189E-02	5.2560E-02	5.6244E-02	5.5065E-02
0.1_1000_50_20_0	5.6115E-02	6.2371E-02	5.9546E-02	6.1706E-02	6.1900E-02	6.0328E-02
0.2_500_50_20_0	5.2153E-02	5.1103E-02	6.5189E-02	5.3601E-02	4.9769E-02	5.4363E-02
0.3_500_50_20_0	5.5550E-02	5.5040E-02	5.2929E-02	5.6872E-02	4.7346E-02	5.3548E-02
0.5_500_50_20_0	5.5901E-02	5.2143E-02	5.7389E-02	5.9688E-02	6.4320E-02	5.7888E-02

Tabla 2. Error de validación de los modelos generados ajustando factor de aprendizaje y épocas  
Loss: sparse categorical crossentropy

Modelo	0	1	2	3	4	ValAcc Media
0.03_3000_50_20_0	8.7611E-01	8.7611E-01	8.5841E-01	8.6726E-01	8.4956E-01	8.6549E-01
0.1_1000_50_20_0	8.4956E-01	8.2301E-01	8.3186E-01	8.3186E-01	8.2301E-01	8.3186E-01
0.2_500_50_20_0	8.5841E-01	8.5841E-01	8.2301E-01	8.4071E-01	8.6726E-01	8.4956E-01
0.3_500_50_20_0	8.6726E-01	8.3186E-01	8.5841E-01	8.6726E-01	8.6726E-01	8.5841E-01
0.5_500_50_20_0	8.4956E-01	8.6726E-01	8.4956E-01	8.4071E-01	8.4071E-01	8.4956E-01

Tabla 3. Precisión de los modelos generados ajustando factor de aprendizaje y épocas

Una vez visto que los mejores ajustes son los de 2 capas de neuronas con 50 y 20 neuronas respectivamente se procede a variar las épocas y el factor de aprendizaje. Los cambios hechos en los valores son incrementales y bastante simples, hasta que observamos que el mejor modelo es como el mejor generado en la Tabla 1, solo que con el factor de aprendizaje a 0.3.

Se muestran a continuación las gráficas de error cuadrático medio y precisión para el modelo 0.3\_500\_50\_20\_0, para poder apreciar como los modelos van mejorando hasta sobreaprender.

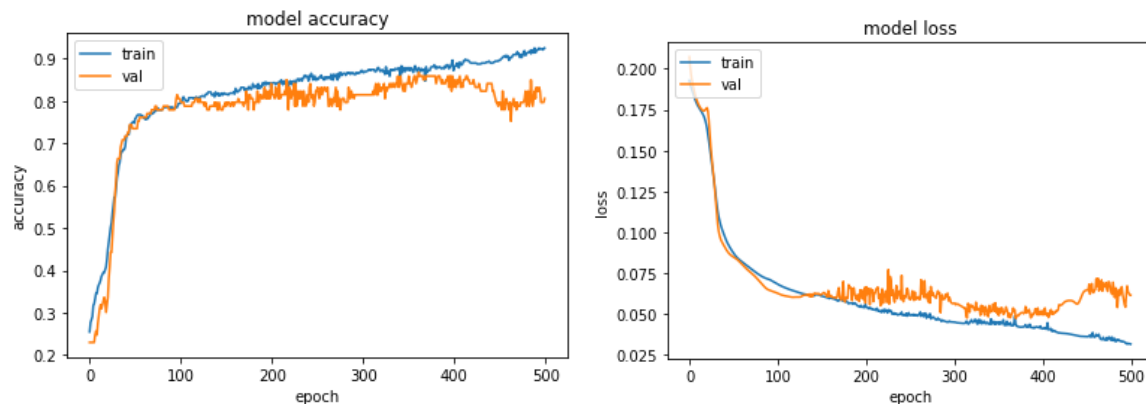


Ilustración 1. Precisión vs. Épocas y Perdida vs. Épocas de entrenamiento y validación para el modelo 0.3\_500\_50\_20\_0  
Loss: sparse categorical crossentropy

## Análisis Mejor Modelo

En el apartado anterior fijamos que el modelo con mejores resultados nos otorga es el de:

- **Factor de aprendizaje:** 0.3
- **Ciclos:** 341
- **N.º capas ocultas:** 2
- **Neuronas en capa 1:** 50
- **Neuronas en capa 2:** 20

A partir de este modelo mostraremos las gráficas de la curva de precisión y error, junto con la matriz de confusión asociada con los datos de test que se separaron previamente. Una vez mostradas, concluimos sobre la calidad del modelo y si fuera posible aplicar ciertas mejoras.

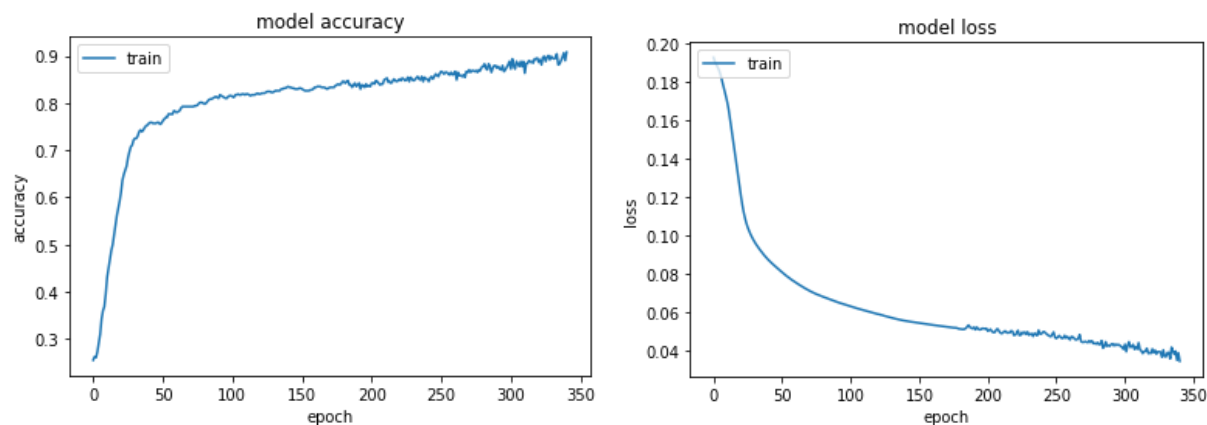


Ilustración 2. Precisión vs. Épocas y Perdida vs. Épocas de test para el mejor modelo  
Loss: sparse categorical crossentropy

Las gráficas de la Ilustración 2 constan de una sola curva al ser el test para el mejor modelo. Como se puede ver el modelo mejora rápidamente al principio y llegado a un punto, 50 épocas, va mejorando poco a poco.

A continuación se muestra la matriz de confusión y los valores de loss y accuracy, Tabla 4, que se obtienen con este modelo:

	bus	opel	saab	van
bus	61	0	2	0
opel	0	31	27	1
saab	0	4	62	0
van	0	0	0	66

	0.3_341_50_20_0
Loss	4.68E-02
Accuracy	8.66E-01

Tabla 4. Matriz de confusión sobre test mejor modelo y Resultados de perdida y precisión mejor modelo  
Loss: sparse categorical crossentropy

Los resultados con bastante buenos, equivocándose muy poco en bus y van, cosa que no podemos decir con opel y saab en los que se produce algo más de error al ser ambos turismos y tener características más parecidas.

Ahora mostraremos 5 predicciones y sus correspondientes valores para las 5 primeras instancias de test.

```
[[1.9930536e-03 1.7733844e-02 3.3050150e-02 9.4722289e-01]
[4.7664238e-07 4.8660431e-02 9.5133907e-01 2.0419886e-08]
[3.4751163e-03 9.5768424e-04 6.9675798e-06 9.9556017e-01]
[3.4179765e-04 1.4456519e-04 5.9844592e-06 9.9950767e-01]
[1.2978218e-01 1.7165761e-04 2.4466109e-04 8.6980158e-01]]
```

Que se traduce mediante argmax en [3 2 3 3 3] que equivale a las clases ['van' 'saab' 'van' 'van' 'van'], que como se puede comprobar en el fichero test.csv las predicciones han sido correctas.

## Conclusiones

Para concluir, se debe destacar que esta parte nos ha servido para ver lo efectivo que puede llegar a ser un clasificador de imágenes (o más bien, de atributos referentes a las imágenes) y de qué variables e hiperparámetros depende. Ha sido interesante también modificar el código para ver los resultados que nos ha ido dando y poder razonar por qué se han generado de dicha manera.

Hay que recalcar también que no nos hemos encontrado problemas mayores durante esta parte de la práctica, ya que ha estado bastante estructurada y las explicaciones y objetivos estaban claros.