

# Redes de Neuronas Artificiales

GRADO EN INGENIERÍA INFORMÁTICA

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

### **Parte II**

**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>Perceptrón Multicapa</b>	<b>3</b>
<b>Red Neuronal Convolucional</b>	<b>5</b>
<b>Comparación de los modelos</b>	<b>7</b>
<b>Conclusiones</b>	<b>7</b>

# Introducción

En esta parte de la práctica procederemos a usar perceptrón multicapa y redes convolucionales sobre un problema de clasificación de imágenes, en este caso a diferencia de la primera parte no serán una serie de atributos determinados de las imágenes, sino que trataremos directamente con estas. Aunque como en la primera parte debemos ajustar los hiperparámetros de los modelos, PM y CNN, para que se ajusten a la tarea que vamos a realizar y se trabajara en Google Colab.

Este problema consistirá en dado el conjunto de datos CIFAR10, compuesto por 60000 imágenes de 32x32 a color y etiquetadas, construir modelos de clasificación que sean capaces de determinar a cuál de las 10 clases pertenece (airplane, automobile, bird, cat, deer, dog, frog, horse, ship o truck). Para ello generaremos una serie de modelos en cada caso y analizaremos a fondo el mejor modelo de cada técnica.

Se han empleado un total de 50000 imágenes para el entrenamiento de las redes y las 10000 restantes para comprobar la generalización del modelo. Además, como se trata de imágenes a color cada una está compuesta por 3 canales de color.

En ambos casos evaluaremos los modelos con la métrica de loss *Sparse Categorical Crossentropy*, que es la mejor para los casos de clasificación en los que la salida viene dada como un valor y no como un vector binario.

## Perceptrón Multicapa

Para esta técnica deberemos tener en cuenta dos hiperparámetros: el n.º de capas y el n.º de neuronas por cada capa. A continuación de muestra una tabla con los modelos generados:

Núm. capas	Núm. neuronas	test_loss	test_accuracy
2	50,20	1.48E+00	4.69E-01
3	50,35,20	1.50E+00	4.69E-01
2	20,10	1.64E+00	4.08E-01
3	20,35,75	1.58E+00	4.27E-01
3	50,10,5	1.86E+00	2.75E-01

Tabla 1. Resultados de los modelos generados de Perceptrón Multicapa  
Loss: sparse categorical crossentropy

Como podemos ver en la Tabla 1, hemos ido variando el n.º de capas entre 2 y 3. Aparte de esto hemos ido aplicando la técnica de la parte anterior, intentando hacer un ‘embudo’ para el n.º de neuronas por capa. Hemos ido iterando estos valores hasta que hemos advertido que los valores no mejoraban, en total se le ha dado 500, pero los resultados se muestran corresponden con el modelo que da menor loss de validación.

Los mejores resultados han sido con 2 y 3 neuronas y siguiendo una estructura de disminuir las neuronas por capa empezando por 50. Para el modelo de 3 capas se intentó meter una capa intermedia para ver si el resultado era más preciso, sin embargo, la diferencia con el modelo de dos capas era mínima y además el tiempo de procesamiento era mayor.

Dicho esto se muestran las gráficas referentes al mejor modelo:

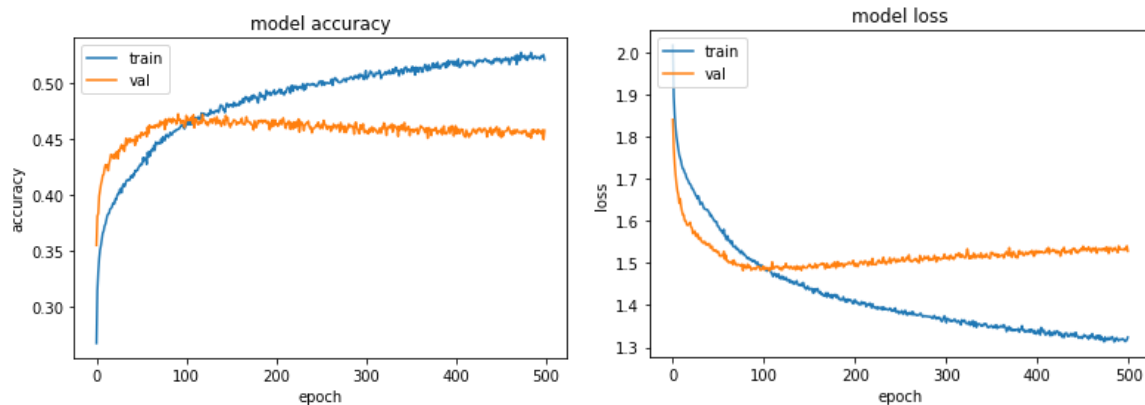


Ilustración 1. Precisión vs. Épocas y Perdida vs. Épocas para el mejor modelo de Perceptrón Multicapa  
Perdida: sparse categorical crossentropy

Como advertimos los mejores resultados se obtienen alrededor de la época 100. Además, se puede apreciar como comienza a sobreaprender y deja de generalizar tan bien, empeorando el de validación.

	airplane	automobile	bird	cat	deer	dog	frog	horse	ship	truck
airplane	522	53	52	43	18	15	13	42	182	60
automobile	27	649	11	23	3	25	14	24	71	153
bird	99	55	241	127	175	76	98	88	23	18
cat	23	42	79	304	42	243	126	64	25	52
deer	56	22	106	83	416	47	120	111	25	14
dog	16	36	75	238	56	383	80	70	25	21
frog	7	29	54	154	129	61	495	29	14	28
horse	41	44	52	97	53	88	22	537	20	46
ship	100	80	12	21	13	37	8	9	651	69
truck	41	265	2	42	4	16	24	48	64	494

Tabla 2. Matriz de confusión mejor modelo de Perceptrón multicapa

Como se puede ver en la matriz de confusión, Tabla 2, el modelo comete muchos errores, sobre todo en bird, cat y dog, en los que falla más de la mitad de las veces. Sin embargo, en clases como automobile o ship funciona realmente bien, considerando que elige entre 10 clases. Con la aproximación por redes neuronales convolucionales esperamos mejorar estos resultados.

# Red Neuronal Convolutacional

Este caso ha sido bastante más complicado de ajustar debido a la gran cantidad de hiperparámetros a tener en cuenta. Debido a su gran número se mostrarán directamente en la Tabla 3.

Núm. capa conv	Núm. filtros	Tam. kernel	Núm. capas dense	Tam. Dense	test_loss	test_accuracy
1ª FASE DE GENERACIÓN DE MODELOS						
1	16	3,3	1	32	1.0838	0.6255
2	16,32	3,3	1	32	0.8420	0.7161
3	16,32,64	3,3	1	32	0.6496	0.7834
3	16,32,64	3,3	2	32,64	0.7872	0.7361
4	32,16,8,4	3,3	1	32	0.9830	0.6506
4	16,16,32,32	3,3	1	32	0.7290	0.7430
2ª FASE DE GENERACIÓN DE MODELOS						
3	16,32,64	3,3	2	50,20	0.7970	0.7309
3	16,32,64	3,3	3	50,35,20	0.8660	0.7176
3ª FASE DE GENERACIÓN DE MODELOS						
3	16,32,64	2,2	1	32	0.7760	0.7295
3	16,32,64	4,4	1	32	0.6540	0.7830

Tabla 3. Resumen de los resultados de las generaciones de modelos de Red Neuronal Convolutacional  
Loss: sparse categorical crossentropy

Como se puede ver, hemos dividido el proceso de generación de modelos en 3 fases. La primera de ellas fue exclusivamente dedicada a ajustar los hiperparámetros referentes a las redes convolucionales, sobre todo nos hemos centrado en el n.º de capas convolucionales y de filtros.

La segunda fase ha servido como ‘unificación’ del perceptrón multicapa y la red convolutacional, ya que los atributos de los dos mejores modelos obtenidos en PM se ponen como capa densa en la red convolutacional.

Por último, para probar diferentes tamaños de kernel, se hacen otras 2 pruebas extra.

Cabe destacar que las épocas utilizadas para la generación de modelos fueron inicialmente de 10, pero se volvió a probar con 100, que es cómo se obtienen los resultados mostrados.

Podemos observar en la Tabla 3 que el mejor modelo ha resultado ser uno con 3 capas convolucionales, aumentando el n.º filtros por capa, llegando a tener la última capa convolutacional 32 filtros, además solo hay una capa densa con 32 neuronas. Este modelo

obtiene una precisión de un 78 % que es bastante alta y consideramos que es un buen resultado. Se muestran a continuación las gráficas correspondientes a dicho modelo y su matriz de confusión:

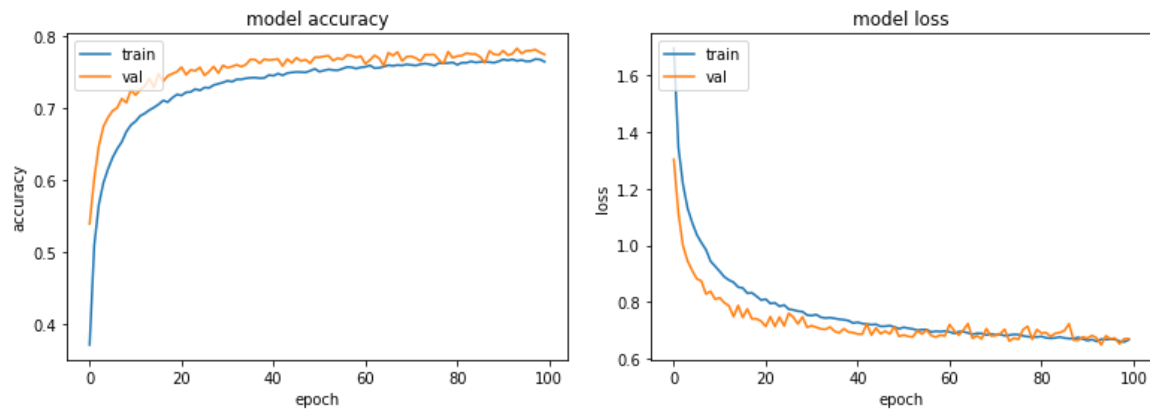


Ilustración 2. Precisión vs. Épocas y Perdida vs. Épocas para el mejor modelo de Red Neuronal Convolutacional

Loss: sparse categorical crossentropy

Se puede apreciar que el modelo aprende correctamente, las curvas van a la par, partiendo de un comienzo con un mayor error van progresivamente mejorando. El modelo va mejorando, no obstante alrededor de la época 90 converge.

	airplane	automobile	bird	cat	deer	dog	frog	horse	ship	truck
airplane	787	16	28	20	24	5	9	16	59	36
automobile	11	874	0	5	4	4	8	4	21	69
bird	59	1	667	39	83	57	50	28	11	5
cat	21	4	57	582	67	142	75	29	10	13
deer	8	2	52	29	808	26	31	37	6	1
dog	10	2	53	125	60	685	19	43	1	2
frog	4	1	50	32	34	10	862	4	2	1
horse	13	1	30	37	63	35	8	805	2	6
ship	44	10	3	16	10	2	2	5	895	13
truck	20	42	3	20	3	6	3	13	21	869

Tabla 4. Matriz de confusión sobre test del mejor modelo de Red Neuronal Convolutacional

En la matriz de confusión, Tabla 4, podemos ver que el modelo es bastante bueno, obteniendo un gran número de aciertos con respecto a las que falla. Inicialmente, al observar el porcentaje de precisión del 78,8 % puede dar la sensación de que no es tan bueno, pero hay que tener en cuenta que hay 10 clases y que acierte entre las 10 no es fácil.

# Comparación de los modelos

Los dos mejores modelos que tenemos han dado los siguientes resultados:

- **Modelo de PM:** Con una precisión de un 46 %, da resultados bastante malos, seguramente debido a que el n.º de tipos de foto que hay que clasificar es muy alto, y este tipo de arquitectura no es muy recomendable para clasificación de imágenes.
- **Modelo de CNN:** El modelo genera una precisión del 78 % que es algo muy bueno teniendo en cuenta el anterior modelo.

Como se puede contemplar, el modelo de redes convolucionales es algo óptimo para clasificación de imágenes y una opción superior a usar un perceptrón multicapa. Esto se debe principalmente al uso de filtros, que sirven bastante para localizar diferencias claves entre imágenes, un ejemplo por ejemplo es la generación de un filtro que reconozca límites verticales. Esta técnica es aún mejor que la generación manual de filtros debido a que se van iterando los valores de estos para encontrar uno útil.

## Conclusiones

Como conclusiones finales, queremos destacar que esta práctica nos ha servido para conocer algo más del ajuste de hiperparámetros que hemos ido desarrollando en la práctica anterior y mejorar nuestras técnicas para encontrar los mejores resultados, ya que estos dependen totalmente de los datos de entrada.

Por otro lado, ha sido bastante útil para conocer mejor los conceptos de las redes convolucionales y el uso de filtros, ya que los demás conceptos manejados en la práctica eran algo más conocidos.