



Universidad Carlos III de Madrid
Campus de Colmenarejo

Redes de Neuronas Artificiales

Curso 2021-22

Práctica 2: Parte 2. Problema de Clasificación con Redes Convolucionales

Fernando Bermúdez Albarrán (100405854@alumnos.uc3m.es)

Andrei García Cuadra (100405803@alumnos.uc3m.es)

Jorge Ríos Marfil (100405942@alumnos.uc3m.es)

Índice

Introducción	2
Perceptrón Multicapa	2
Descripción de los experimentos realizados	2
Resultados obtenidos	2
Evolución de los valores de accuracy y loss de entrenamiento y test del mejor modelo	2
Matriz de confusión de los datos de test del mejor modelo	3
Resultados obtenidos de todos los modelos	3
Red Neuronal Convolutiva	4
Descripción de los experimentos realizados	4
Resultados obtenidos	4
Evolución de los valores de accuracy y loss de entrenamiento y test del mejor modelo	4
Matriz de confusión de los datos de test del mejor modelo	6
Resultados obtenidos de todos los modelos	6
Comparación entre PM y CNN	8
Conclusión	8

Introducción

En el presente informe se detallan los resultados obtenidos de realizar distintos experimentos sobre el problema de clasificación de imágenes del conjunto CIFAR10 utilizando tanto Perceptrón Multicapa (PM) como Redes de Neuronas Convolucionales (CNN). Para cada una de las dos técnicas, se realizará una breve descripción de los experimentos realizados, seguida de la presentación de los resultados obtenidos y el mejor modelo. Finalmente, se hará una comparación entre las dos técnicas.

Perceptrón Multicapa

Descripción de los experimentos realizados

Hemos realizado un total de cinco experimentos distintos, cambiando el número de capas y neuronas ocultas con el objetivo de encontrar una arquitectura adecuada. Para ello, hemos partido de la arquitectura proporcionada por el tutorial de la práctica, manteniendo como constantes el número de ciclos (épocas) y la tasa de aprendizaje, con los valores 10 y 0.001, respectivamente.

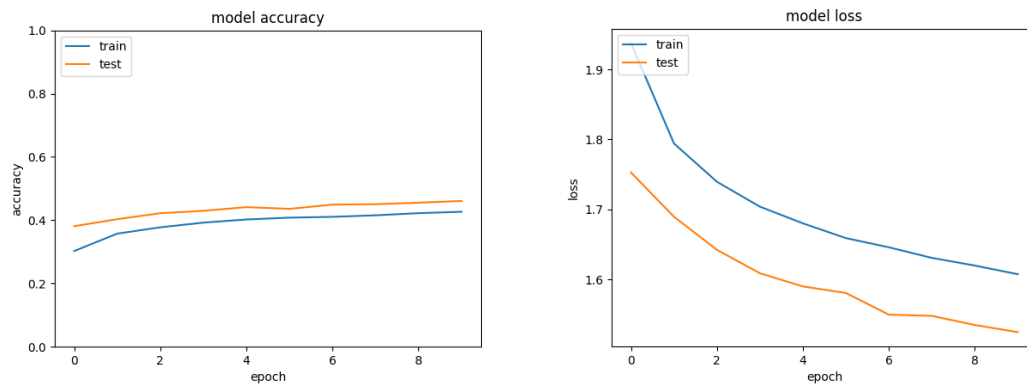
Resultados obtenidos

Evolución de los valores de accuracy y loss de entrenamiento y test del mejor modelo

De los cinco experimentos realizados, el que mejores resultados nos ha dado ha sido el tercero. A continuación se muestran los resultados de dicho experimento:

Ciclo	Accuracy (Train)	Loss (Train)	Accuracy (Test)	Loss (Test)
1	3.024600148200988770e-01	1.937758326530456543e+00	3.810999989509582520e-01	1.752858042716979980e+00
2	3.574999868869781494e-01	1.794334888458251953e+00	4.032000005245208740e-01	1.689637541770935059e+00
3	3.775399923324584961e-01	1.739578604698181152e+00	4.219999909400939941e-01	1.642356991767883301e+00
4	3.923600018024444580e-01	1.704001665115356445e+00	4.296000003814697266e-01	1.608875155448913574e+00
5	4.022200107574462891e-01	1.680180668830871582e+00	4.413000047206878662e-01	1.590079545974731445e+00
6	4.080399870872497559e-01	1.659105420112609863e+00	4.357999861240386963e-01	1.580728769302368164e+00
7	4.105800092220306396e-01	1.645964860916137695e+00	4.492000043392181396e-01	1.549734711647033691e+00
8	4.154999852180480957e-01	1.630873560905456543e+00	4.505000114440917969e-01	1.548024177551269531e+00
9	4.222399890422821045e-01	1.619885683059692383e+00	4.553000032901763916e-01	1.534986972808837891e+00
10	4.266400039196014404e-01	1.607538223266601562e+00	4.607999920845031738e-01	1.524798989295959473e+00

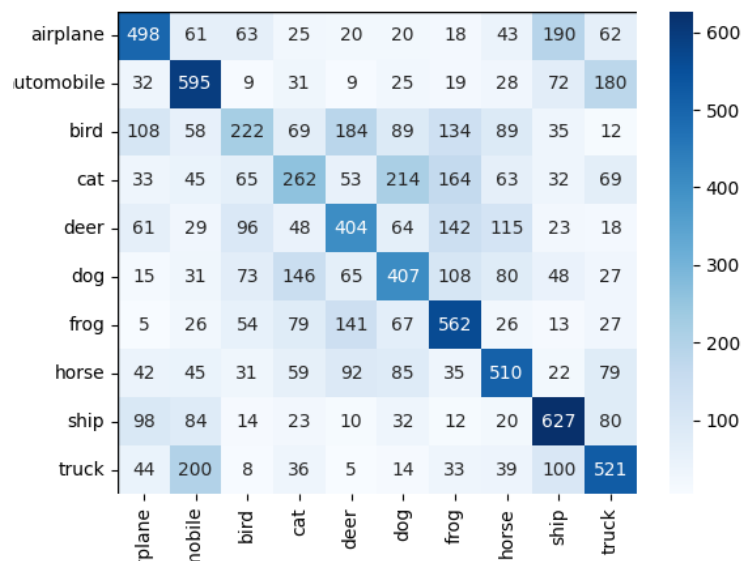
A continuación se pueden ver las gráficas de la evolución del mejor experimento (experimento 3):



A la izquierda se puede observar la gráfica de la evolución del accuracy, mientras que a la derecha se puede ver el loss.

Matriz de confusión de los datos de test del mejor modelo

A continuación se puede observar la matriz de confusión generada por el programa del mejor modelo (experimento 3):



Como podemos observar, el modelo tiene muchos errores a la hora de clasificar correctamente las imágenes de test.

Resultados obtenidos de todos los modelos

A continuación se pueden ver los valores de accuracy y loss de entrenamiento y test al finalizar el aprendizaje de todos los experimentos realizados:

Nº de Experimento	Hiperparámetros usados	Accuracy (Train)	Loss (Train)	Accuracy (Test)	Loss (Test)
1	- Capas ocultas: 1 - Neuronas por capa oculta: 50	4.280200004577636719e-01	1.609699964523315430e+00	4.544000029563903809e-01	1.554758310317993164e+00
2	- Capas ocultas: 2 - Neuronas por capa oculta: 50	4.034599959850311279e-01	1.681309700012207031e+00	4.359999895095825195e-01	1.592381000518798828e+00
3	- Capas ocultas: 2 - Neuronas por capa oculta: 75	4.266400039196014404e-01	1.607538223266601562e+00	4.607999920845031738e-01	1.524798989295959473e+00
4	- Capas ocultas: 2 - Neuronas por capa oculta: 25	3.454999923706054688e-01	1.806937336921691895e+00	3.896000087261199951e-01	1.706998586654663086e+00
5	- Capas ocultas: 3 - Neuronas por capa oculta: 50	3.769800066947937012e-01	1.725610136985778809e+00	4.162999987602233887e-01	1.618427276611328125e+00

Para el primer experimento usamos los hiperparámetros que venían con el ejemplo del tutorial. Luego intentamos aumentar el número de capas a 2, sin mucho éxito ya que el modelo empeoró. Probamos también a subir el número de neuronas por capa a 75, consiguiendo el mejor modelo. Probamos también a reducir las neuronas a 25 por capa, resultando en el peor modelo de todos. Finalmente, probamos a poner 3 capas con 50 neuronas, consiguiendo un modelo ligeramente mejor al anterior.

Red Neuronal Convolutiva

Descripción de los experimentos realizados

De la misma manera que con el Perceptrón Multicapa, hemos realizado un total de cinco experimentos, cambiando el número de capas convolucionales, así como el número de filtros y el tamaño del kernel de dichas capas convolucionales, y también el tamaño y número de capas “Fully Connected”. Hemos decidido mantener constante el número de ciclos (épocas) a 30 y la tasa de aprendizaje a 0.001, es decir, el valor del ejemplo del tutorial.

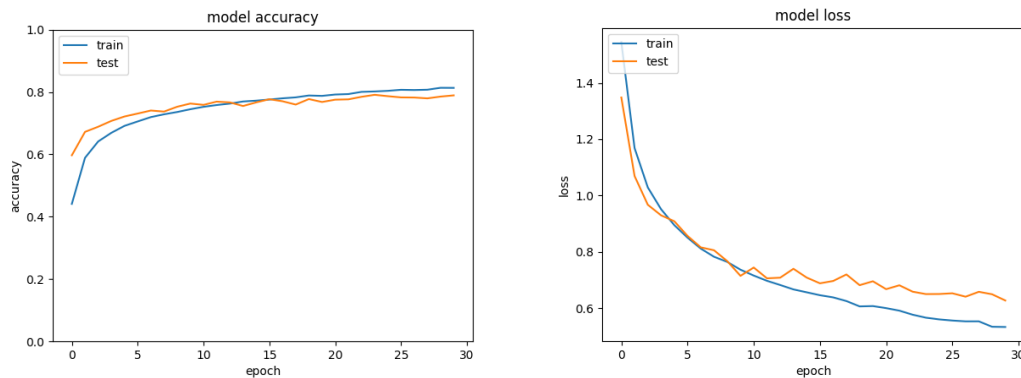
Resultados obtenidos

Evolución de los valores de accuracy y loss de entrenamiento y test del mejor modelo

De los cinco experimentos realizados, el que mejores resultados nos ha dado ha sido el quinto. A continuación se muestran los resultados de dicho experimento:

Ciclo	Accuracy (Train)	Loss (Train)	Accuracy (Test)	Loss (Test)
1	4,41E-1	1,54E+00	5,97E-01	1,35E+00
2	5,89E-1	1,17E+00	6,72E-01	1,07E+00
3	6,41E-1	1,03E+00	6,88E-01	9,67E-01
4	6,69E-01	9,52E-01	7,07E-01	9,30E-01
5	6,91E-01	8,95E-01	7,21E-01	9,09E-01
6	7,05E-01	8,51E-01	7,31E-01	8,57E-01
7	7,19E-01	8,12E-01	7,40E-01	8,16E-01
8	7,28E-01	7,83E-01	7,37E-01	8,06E-01
9	7,36E-01	7,64E-01	7,53E-01	7,67E-01
10	7,45E-01	7,37E-01	7,63E-01	7,15E-01
11	7,52E-01	7,16E-01	7,59E-01	7,45E-01
12	7,58E-01	6,97E-01	7,69E-01	7,06E-01
13	7,63E-01	6,82E-01	7,67E-01	7,08E-01
14	7,69E-01	6,67E-01	7,55E-01	7,40E-01
15	7,72E-01	6,57E-01	7,67E-01	7,09E-01
16	7,76E-01	6,46E-01	7,77E-01	6,88E-01
17	7,80E-01	6,39E-01	7,70E-01	6,97E-01
18	7,83E-01	6,25E-01	7,60E-01	7,20E-01
19	7,89E-01	6,07E-01	7,77E-01	6,82E-01
20	7,87E-01	6,08E-01	7,68E-01	6,96E-01
21	7,92E-01	6,00E-01	7,75E-01	6,68E-01
22	7,93E-01	5,91E-01	7,76E-01	6,81E-01
23	8,00E-01	5,77E-01	7,85E-01	6,59E-01
24	8,02E-01	5,66E-01	7,91E-01	6,50E-01
25	8,04E-01	5,60E-01	7,86E-01	6,50E-01
26	8,07E-01	5,56E-01	7,83E-01	6,53E-01
27	8,06E-01	5,53E-01	7,82E-01	6,41E-01
28	8,07E-01	5,53E-01	7,80E-01	6,58E-01
29	8,13E-01	5,34E-01	7,85E-01	6,50E-01
30	8,13E-01	5,33E-01	7,89E-01	6,28E-01

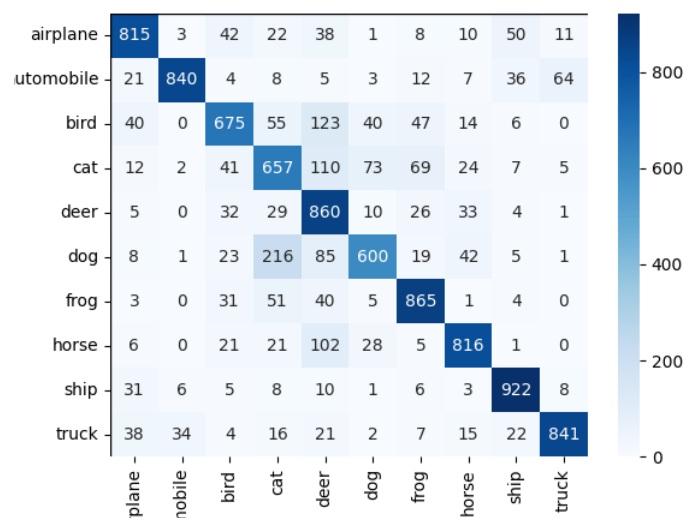
A continuación se pueden ver las gráficas de la evolución del mejor experimento (experimento 5):



A la izquierda se puede observar la gráfica de la evolución del accuracy, mientras que a la derecha se puede ver el loss.

Matriz de confusión de los datos de test del mejor modelo

A continuación se puede observar la matriz de confusión generada por el programa del mejor modelo (experimento 5):



Como podemos observar, el modelo tiene muchos menos errores a la hora de clasificar correctamente las imágenes de test que en el caso del PM.

Resultados obtenidos de todos los modelos

A continuación se pueden ver los valores de accuracy y loss de entrenamiento y test al finalizar el aprendizaje de todos los experimentos realizados:

Nº Exp	Hiperparámetros usados	Accuracy (Train)	Loss (Train)	Accuracy (Test)	Loss (Test)
1	- Capas convolucionales: 1 - Filtros en capas convolucionales: 16 - Kernel en capas convolucionales: (3,3) - Capas "Fully Connected": 1 - Tamaño capas "Fully Connected": 32	5.238000154495239258e-01	1.287524580955505371e+00	5.871000289916992188e-01	1.187501192092895508e+00
2	- Capas convolucionales: 2 - Filtros en capas convolucionales: 65, 50 - Kernel en capas convolucionales: (8,8), (4,4) - Capas "Fully Connected": 1 - Tamaño capas "Fully Connected": 32	7.884399890899658203e-01	5.880052447319030762e-01	7.318000197410583496e-01	8.055219054222106934e-01
3	- Capas convolucionales: 2 - Filtros en capas convolucionales: 16, 32 - Kernel en capas convolucionales: (3,3), (3,3) - Capas "Fully Connected": 2 - Tamaño capas "Fully Connected": 32, 16	5.812399983406066895e-01	1.149540543556213379e+00	6.557000279426574707e-01	1.044198393821716309e+00
4	- Capas convolucionales: 2 - Filtros en capas convolucionales: 80, 100 - Kernel en capas convolucionales: (7,7), (3,3) - Capas "Fully Connected": 1 - Tamaño capas "Fully Connected": 50	8.217599987983703613e-01	4.893662035465240479e-01	7.314000129699707031e-01	8.081874251365661621e-01
5	- Capas convolucionales: 3 - Filtros en capas convolucionales: 32, 64, 128 - Kernel en capas convolucionales: (3,3), (3,3), (3,3) - Capas "Fully Connected": 1 - Tamaño capas "Fully Connected": 128	8.131600022315979004e-01	5.334455370903015137e-01	7.890999913215637207e-01	6.275117993354797363e-01

Para el primer experimento usamos los hiperparámetros que venían con el ejemplo del tutorial. Luego intentamos aumentar el número de capas y de filtros, así como el kernel, mejorando considerablemente el modelo con respecto al anterior. Luego probamos a también aumentar el número de capas "Fully Connected", así como variar el tamaño de dichas capas. Esto nos generó un peor resultado, pero mejor que el inicial. Decidimos dejar el número de capas de este tipo a 1 en el resto de experimentos. Más tarde probamos a dejar 2 capas convolucionales con muchos más filtros y kernels algo más pequeños que en el segundo experimento y subiendo el número de capas "Fully Connected". Esto nos generó un modelo ligeramente mejor al segundo. Finalmente, probamos una configuración inspirada en información que encontramos en internet, dejando 3 capas con filtros secuencialmente creciendo (32,64,128) y subiendo el tamaño de la capa "Fully Connected" a 128 también. Esto nos generó el mejor modelo.

Comparación entre PM y CNN

Con los resultados vistos en las anteriores secciones, podemos ver que la técnica de Redes de Neuronas Convolucionales es muy superior al Perceptrón Multicapa a la hora de clasificar imágenes, ya que la configuración más simple del CNN ya era mejor que el mejor modelo que pudimos sacar del PM (recordamos que el valor de accuracy de test para el mejor PM era del orden de 0.46, mientras que el del ejemplo de CNN era del orden de 0.59)

Conclusión

La práctica nos ha servido para trabajar con redes de neuronas para la clasificación de imágenes, nos ha parecido muy interesante ya que nos muestra de una forma práctica lo aprendido en las clases de teoría, además de un enfoque práctico que es lo de verdad útil para un futuro. El hecho de contar con un pdf con los pasos detallados de cómo funciona el código, nos parece un gran acierto.

Nos hubiera gustado que hubiese habido algún tipo de competición o concurso para que se valorará el número de entrenamientos, pruebas y resultados obtenidos.