uc3m | Universidad Carlos III de Madrid

Redes Neuronales
Práctica 2
Segunda parte
Problema de Clasificación con Redes Convolucionales

Luis Docampo Mateos-Pardo 100383345
Alejandro Hernández Artiles 100405981

Introducción

En la segunda parte de la segunda práctica se aborda un problema de clasificación de imágenes usando dos modelos, un perceptrón multicapa y una red convolucional. Se trabaja con el conjunto de datos CIFAR10, que está compuesto por 60000 imágenes a color, donde hay 10 posibles valores para la clase, donde la clase puede ser un animal, como un caballo, o un vehículo, como un coche. Se dividirá el conjunto en test y entrenamiento. El conjunto de entrenamiento contendrá 50000 imágenes y el de test 10000 imágenes. Remarcar que cada imagen consta de 32x32 píxeles y 3 canales (RGB) para los colores.
El objetivo de esta práctica es comparar y contrastar el desempeño de los dos modelos, PM y convolucional, a la hora de trabajar con imágenes. También se tratará encontrar modelos lo más adecuados posibles en ambos casos para la clasificación, experimentando con diferentes arquitecturas y optimizaciones.

Perceptrón Multicapa

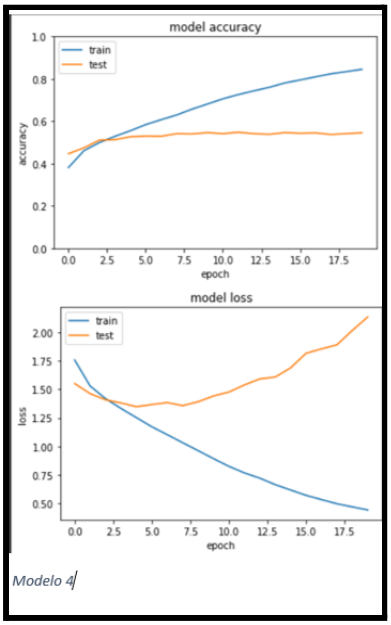
A continuación se explican los experimentos realizados para la clasificación con el modelo de perceptrón multicapa. Antes, el conocimiento de las notaciones utilizadas es necesario. La topología de la red vendrá determinada por las capas, separadas por el símbolo "|". Luego las capas de DropOut serán representadas por DO(x), siendo x el ratio de neuronas apagadas que se quiere. En la fila "Función arq." se representan las funciones utilizadas para las neuronas de cada capa. Por ejemplo, si existe una topología 20|20 la "Función arq." sería relu|sig, indicando que la primera capa usa relu y la segunda sigmoid. Si hubiera una capa de DropOut entre medias de las dos capas "Función arq." seguiría siendo relu|sig, ya que se ignoran las capas DropOut en esta notación para expresar qué funciones se usan.
Se recalca que la capa de entrada y de salida no han sufrido cambios, por ende la capa de entrada tendrá 32 x 32 x 3 neuronas y usará Flatten, y la de salida tendrá 10 neuronas con la función softmax.
Tras muchos y diversos experimentos, se exponen los que mejor resumen el proceso realizado hasta una arquitectura satisfactoria:

	Modelo 1	Modelo 2	Modelo 3	Modelo 4	Modelo 5
Optimización	Rmsprop(0.001)	Adam(0.001)	Adam(0.001)	Adam(0.001)	Adam(0.001)
Ciclos	20.0	50.0	20.0	20.0	20.0
Arquitectura	20|20|20	50|DO(0.25)|50|50|DO(0.25)	50|50|50|DO(0.25)	500|500|500|DO(0.25)	500|DO(0.25)|500|DO(0.25)|500|DO(0.25)
Función arq.	sig|sig|sig	sig|sig|sig	relu|relu|relu	relu|relu|relu	relu|relu|relu
Nº Parámetros	62522.0	159272.0	159272.0	2042522.0	2042522.0
Loss ent.	1.55	1.4	1.17	0.44	0.95
Acc ent.	0.445	0.5	0.58	0.845	0.66
Loss test.	1.62	1.45	1.41	2.13	1.26
Acc test.	0.414	0.484	0.51	0.545	0.565

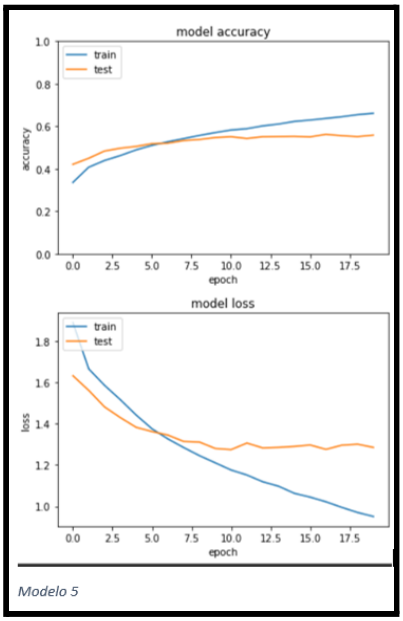
Tras los experimentos se concluyó que por lo general la optimización de Adam funcionaba

mejor que el RMSprop, usado en el Modelo 1, ya que Adam es similar a éste, pero además hace uso del momentum, acelerando el descenso del gradiente en direcciones similares a las anteriores. Luego, se empezó a probar con arquitecturas usando la función sigmoid y capas de DropOut para prevenir el sobreaprendizaje, obteniéndose como mejor resultado el Modelo 2, con una tasa de aciertos del 48%. La función tanh daba resultados similares. Entonces, se experimentó con la función relu usando el Modelo 3, sobrepasando los resultados del Modelo 2 por tres centésimas con una arquitectura sencilla de 3 capas y una capa de DropOut al final. Se dedujo que la función relu trabaja mejor con imágenes, ya que, al contrario que la función sigmoid y tanh, que reducen los valores grandes a valores cercanos a 1, la relu deja las entradas positivas tal y como entran, sin interferir en su valor, y por ende, no modifica los valores de los píxeles, permitiendo que la red aprenda los valores adecuados.

Tras el Modelo 3 no se encontró una arquitectura de tamaño similar que diera mejores resultados, por lo que se aumentó la cantidad de neuronas para mejorar en poder de computación por fuerza bruta. Se obtuvo entonces el Modelo 4, igual que el Modelo 3 pero con 500 neuronas en cada capa, que aportó unos resultados superiores de 54% de aciertos. Sin embargo, se observó que el Modelo 4 sobreaprendía gravemente, como se ve en la gráfica model loss del Modelo 4. Para solucionar este problema se decidió añadir capas de DropOut entre las capas de neuronas, para que en el entrenamiento la red prescindiera de algunas neuronas y profundizara en el entrenamiento, mejorando su capacidad de generalización. Finalmente, se obtuvo el Modelo 5, con una tasa de aprendizaje del 56,5%, la más alta. Este modelo cuenta con 3 capas de 500 neuronas en cada una con función relu, y entre cada capa, una capa de DropOut. Comparando las gráficas del Modelo 4 y el Modelo 5 se comprueba que el problema del sobreaprendizaje se reduce altamente gracias al añadido de capas de DropOut.



Modelo 4



Modelo 5

Se aprecia como el error de validación se mantiene relativamente constante para el Modelo 5, y como en el Modelo 4, sobre el ciclo 7, el error empieza a subir drásticamente. Sin embargo, la tasa de aciertos se mantiene constante en ambos casos, con una ligera ventaja para el Modelo 5.

722	25	50	16	22	9	18	13	91	34
50	695	12	22	4	9	17	12	65	114
101	17	389	74	144	80	106	47	23	19
42	18	74	328	61	241	130	50	20	36
64	6	115	61	446	50	145	74	28	11
28	6	65	166	58	511	71	53	19	23
18	16	60	44	71	51	693	11	17	19
45	11	41	50	80	91	23	609	16	34
142	57	11	25	17	12	8	5	681	42
52	178	16	29	11	19	15	41	65	574

A la derecha se observa la matriz de confusión del Modelo 5, pudiendo apreciarse que, a pesar de que en la diagonal se acumulen los valores más grandes, la red comete muchos errores, sobretodo en la clasificación de la 3º y 4º clase, donde acierta 389 y 328 respectivamente, de las 1000 instancias en el conjunto de test. Estas dos clases son bird y cat. Se puede ver como la red clasifica mejor, por lo general, los vehículos (avión - 1º clase, coche - 2º clase, barco - 9º clase (o penúltima) y camión - 10º (o última)), ya que son más distinguibles de las demás clases. Sin embargo, nótese como en la última fila, la de la clase camión, hay 178 instancias que se clasifican como coche, ya que son más parecidos entre sí.

En conclusión, se obtienen unos resultados muy poco precisos, con una tasa de aciertos del 56,5% como máximo en el Modelo 5, en contraste con la cantidad de poder computacional que se usa en la red, con 3 capas de 500 neuronas, unos $2{,}04\times10^6$ de parámetros entrenables. Un perceptrón multicapa sufre al trabajar con imágenes por su propiedad de arquitectura de capas fully-connected, donde todas las neuronas están conectadas entre sí, que acaba siendo ineficiente para entradas del tamaño y variedad de una imagen. Por tanto, próximamente se trabajará con redes convolucionales, que permiten que haya neuronas que no estén conectadas en capas enfrentadas, permitiendo una mejor generalización para este tipo de entradas.

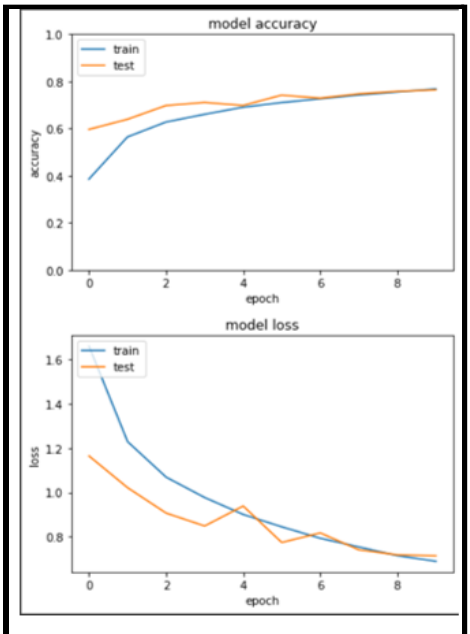
CNN

A continuación se explican los experimentos realizados para la clasificación con el modelo CNN. Antes, el conocimiento de las notaciones utilizadas es necesario. Para representar la arquitectura de las capas fully-connected se ha usado la misma notación de la tabla anterior. Para representar las nuevas capas convolucionales se han usado 3 nuevas filas: “Arq. Capas Conv” que tiene la forma *x*|*y*|*z* siendo *x* el número de filtros para la primera capa, *y* para la segunda y *z* para la tercera. Además, la capa también puede ser DO(DropOut) o MaxP(Capa de max pooling). Luego está la fila “Tamaño filtros” que indica el tamaño de los filtros de las capas en el mismo orden en el que vienen definidas en “Arq. Capas Conv”. Por último, se añade la fila “Función Capas Conv”, en el que se especifica la función usada por cada capa convolucional en el orden establecido en “Arq. Capas Conv”. Se recalca que la capa de entrada y de salida no han sufrido cambios, por ende la capa de entrada tendrá 32 x 32 x 3 neuronas y usará Flatten, y la de salida tendrá 10 neuronas con la función softmax.

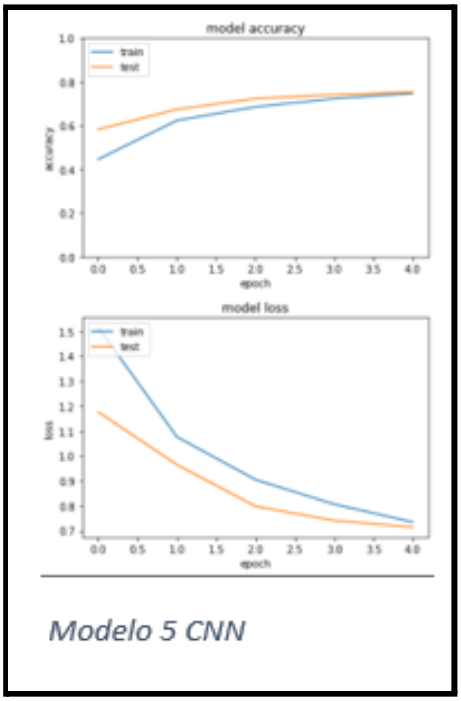
Tras muchos y diversos experimentos, se exponen los que mejor resumen el proceso realizado hasta una arquitectura satisfactoria:

	Modelo 1	Modelo 2	Modelo 3	Modelo 4	Modelo 5
Optimización	Adam(0.001)	Adam(0.001)	Adam(0.001)	Adam(0.001)	Adam(0.001)
Ciclos	10.0	5.0	10.0	5.0	5.0
Arq. CapasConv	16|MaxP|16|MaxP|DO(0.25)	16|32|64|128|MaxP	64|MaxP|128|MaxP|256|MaxP|DO(0.25)	256|MaxP|128|MaxP|64|MaxP|32|DO(0.25)	64|MaxP|256|MaxP|256|MaxP|DO(0.25)
Tamaño Filtros	3x3|2x2|3x3|2x2	3x3|3x3|3x3|3x3|2x2	3x3|2x2|3x3|2x2|3x3|2x2|	3x3|2x2|3x3|2x2|3x3|2x2|3x3|2x2	3x3|2x2|3x3|2x2|3x3|2x2
Función CapasConv	relu|relu	relu|relu|relu|relu	relu|relu|relu	relu|relu|relu|relu	relu|relu|relu
Arq. FullyConnect	32|DO(0.25)|32|DO(0.25)	100|DO(0.25)|100|DO(0.25)	50|DO(0.25)|50|DO(0.25)	50|DO(0.25)|50|DO(0.25)	100|DO(0.25)|100|DO(0.25)
Función FullyConnect	relu|relu	relu|relu	relu|relu	relu|relu|	relu|relu
Nº Parámetros	36966.0	3385462.0	578738.0	403986.0	1160406.0
Loss ent.	1.15	0.68	0.68	1.02	0.734
Acc ent.	0.601	0.762	0.767	0.643	0.747
Loss test.	1.00	0.85	0.71	0.938	0.714
Acc test.	0.675	0.71	0.764	0.672	0.753

A primera vista se aprecia como los resultados generales son bastante mejores que los del perceptrón multicapa. A través de los experimentos se deduce que se obtienen mejores resultados usando un número de filtros divisible entre 32 y que los filtros funcionaban mejor los 3x3 y para MaxPooling los de 2x2, ya que la imagen no es muy grande y filtros mayores eran una segmentación de imagen excesiva. También, la función relu, como se había visto anteriormente, era la mejor para el trabajo con imágenes. Se puede ver como en el Modelo 1 se prueba con 16 filtros en cada capa lo que da una tasa de aprendizaje 1 décima mayor que el mejor Multicapa. Tras ello, se probó con el Modelo 2, que mantenía capas con menos de 32 filtros, que mejoró algo los resultados. En el Modelo 3 se añadieron capas divisibles entre 32, y en orden ascendente de número de filtros, con dos capas fully-connected de 50 capas, con DropOut para evitar sobreaprendizaje. Este modelo dio el mejor resultado con una tasa de aciertos del 76%. Tras este modelo con buenos resultados se trató de mejorarlo con fuerza bruta, como en el caso del Multicapa, añadiendo más número de filtros y capas fully-connected, como se puede ver en los modelos 4 y 5. El Modelo 4, que tiene las capas ordenadas en orden descendente por número de filtros obtuvo unos resultados insatisfactorios, sin embargo, el Modelo 5 obtuvo unos resultados muy parecidos al Modelo 3 (la estructura es similar), con menos número de ciclos, pero mucho más tiempo de procesamiento.



Modelo 3 CNN



Modelo 5 CNN

Se aprecia como ambas gráficas de accuracy y loss para el Modelo 3 y 5 son parecidas, con un descenso pronunciado del error de entrenamiento, y un error de test que va convergiendo. Estas redes rápidamente se estancan, debido a su gran dimensión y altos valores de las imágenes, manteniéndose casi constantes.

Comparación de modelos

Comparando los modelos, se concluye que las redes convolucionales son muy superiores a las multicapa a la hora de clasificar imágenes, ya que con menos conexiones logran mejores resultados en menos ciclos, aunque sus ciclos duren más tiempo. Se aprecia en la matriz de confusión del Modelo 3 CNN, a la derecha, el cual es el mejor modelo de todo el estudio, la superioridad de la red, donde se puede ver, que en casos que el Modelo PM fallaba numerosas veces como la confusión entre el camión y el coche, ésta la reduce mucho, pasando por ejemplo de 574 camiones clasificados correctamente a 881, en un dataset de 1000 camiones. Toda esta mejoría se debe a que la red convolucional está mejor preparada en concepto para trabajar con imágenes, ya que, al contrario que las multicapa que son fully-connected, estas permiten filtros y manipulaciones del tamaño de la imagen gracias a que se puede conectar cada neurona oculta solo con un subconjunto de las entradas, pudiendo dividir entre píxeles adyacentes y no todos en su conjunto, haciendo la imagen más manejable.

823	8	51	19	11	3	6	10	39	30
12	875	2	5	0	0	5	0	13	88
61	4	640	53	75	45	79	26	7	10
19	3	66	544	61	146	99	33	8	21
15	2	68	70	697	9	74	57	6	2
11	1	47	198	40	612	30	46	2	13
3	3	20	41	18	5	897	3	4	6
12	2	24	40	54	38	6	809	1	14
62	16	6	12	6	2	4	5	866	21
28	54	3	10	1	1	3	8	11	881

Conclusión

En conclusión, este estudio ha demostrado por qué las redes convolucionales son las preferidas para clasificación y trabajo con imágenes. Las características intrínsecas que posee, como su capacidad de dividir las uniones entre neuronas en subconjuntos, escalamiento y reducción de la entrada mediante capas de pooling y la posibilidad de añadir capas fully-connected al final, la hacen ideal para esta tarea. También, en este estudio se ha trabajado con redes con DropOut, creando modelos con menos tendencia al sobreaprendizaje, que han acabado siendo los mejores modelos y demostrando no solo por qué el DropOut es útil en las redes de neuronas artificiales, sino también lo crucial que puede ser implementar sucesos que ocurren en las redes de neuronas biológicas a las artificiales.