

Práctica 6 - parte II

Categorización de imágenes de histopatología mediante Redes Neuronales Convolucionales

Mario Berrios Carmona

*Fundamentos de Sistemas Inteligentes en Visión
Grado en Ingeniería Informática*

Diciembre 2022

1. Introducción

A continuación se muestran cuatro modelos distintos para clasificación mediante una red neuronal convolucional (CNN). Se expone además la precisión y la pérdida de cada uno de los modelos en la fase de entrenamiento así como el resultado obtenido con el set *test* una vez se ha concluido el entrenamiento.

2. Modelo 1

Este modelo consta de un solo bloque convolucional consistente en una capa convolucional 2D, una capa de activación mediante la función ReLU y una capa pool mediante máximos. Segudamente al bloque convolucional se ha añadido una única capa densa adicional a la capa de *output*. Dicha capa utiliza a su vez una activación ReLU.

Model: "Model1"

Layer (type)	Output Shape	Param #
Conv1 (Conv2D)	(None, 124, 124, 20)	1520
Relu1 (Activation)	(None, 124, 124, 20)	0
Pool1 (MaxPooling2D)	(None, 62, 62, 20)	0
flatten_2 (Flatten)	(None, 76880)	0
Dense1 (Dense)	(None, 640)	49203840
Ouput (Dense)	(None, 4)	2564

=====
Total params: 49,207,924
Trainable params: 49,207,924
Non-trainable params: 0

Figura 1: Resumen del modelo 1

Los hiperparámetros de cada capa han ido cambiando a medida que se han ido haciendo pruebas, dejando el mejor resultado obtenido en el Jupyter Notebook que se adjunta. El entrenamiento para estos parámetros se puede analizar mediante la figura 2. Como se observa, mientras *Train* aumenta su precisión y reduce su pérdida, *Validation* tiene un comportamiento errático, aunque siempre se cumple que a mayor precisión menor pérdida y viceversa.

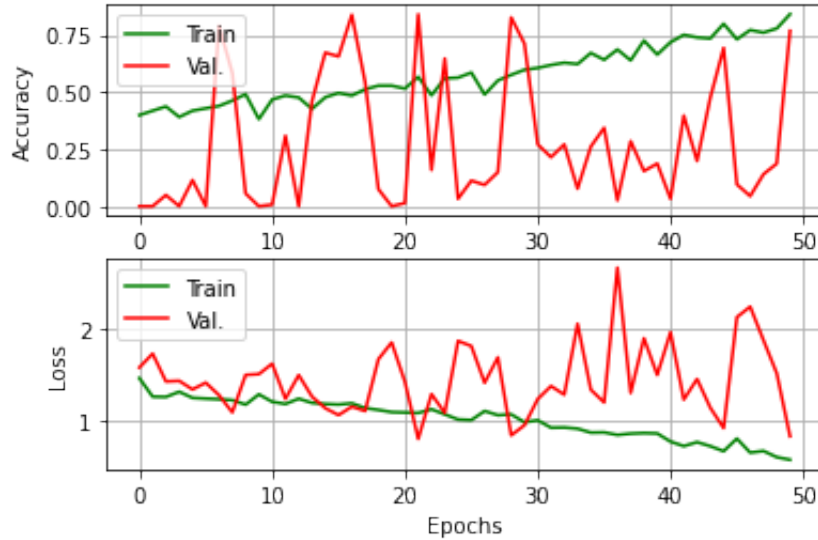


Figura 2: Entrenamiento del modelo 1

Para terminar, el resultado obtenido al utilizar el modelo con el conjunto *Test* obtenemos una precisión de 0,599 y una pérdida de 0,804

3. Modelo 2

Este modelo difiere con el modelo 1 en que se han añadido dos capas convolucionales y dos capas densas adicionales. Se sigue manteniendo ReLU como función tanto en la capa de activación en el bloque de convolución así como en las capas densas finales.

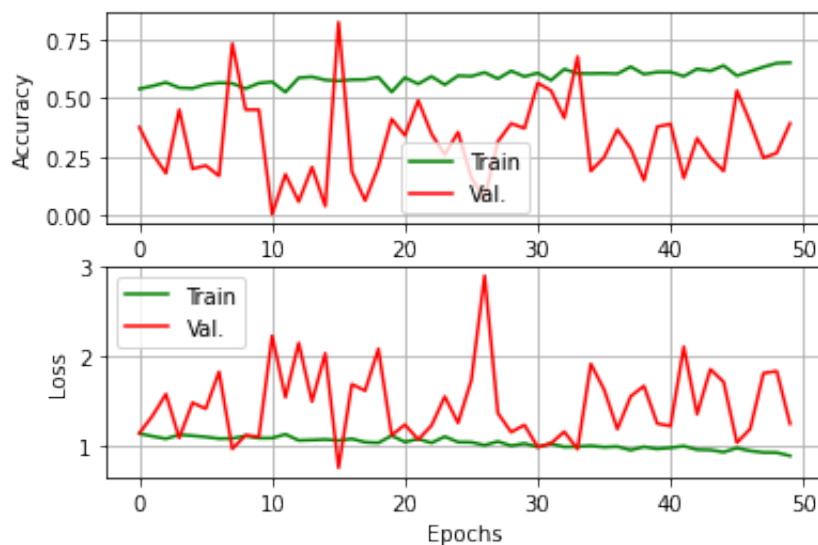


Figura 3: Entrenamiento del modelo 2

Model: "Model2"

Layer (type)	Output Shape	Param #
Conv1 (Conv2D)	(None, 124, 124, 20)	1520
Relu1 (Activation)	(None, 124, 124, 20)	0
Pool1 (MaxPooling2D)	(None, 62, 62, 20)	0
Conv2 (Conv2D)	(None, 58, 58, 20)	10020
Relu2 (Activation)	(None, 58, 58, 20)	0
Pool2 (MaxPooling2D)	(None, 29, 29, 20)	0
Conv3 (Conv2D)	(None, 25, 25, 20)	10020
Relu3 (Activation)	(None, 25, 25, 20)	0
Pool3 (MaxPooling2D)	(None, 12, 12, 20)	0
flatten_3 (Flatten)	(None, 2880)	0
Dense1 (Dense)	(None, 640)	1843840
Dense2 (Dense)	(None, 640)	410240
Dense3 (Dense)	(None, 640)	410240
Ouput (Dense)	(None, 4)	2564

=====
Total params: 2,688,444
Trainable params: 2,688,444
Non-trainable params: 0

Figura 4: Resumen del modelo 2

Al igual que en el modelo anterior, en la figura 3 se observa el mismo comportamiento, si bien esta vez la mejoría en la precisión y pérdida está menos acusada. También se puede observar que aunque sea menor la mejoría a lo largo del entrenamiento, el punto de partida tanto de la precisión como de la pérdida es mejor que la obtenida en el modelo 1.

Al realizar la clasificación con el conjunto *Test* se obtiene una precisión de 0,618 y una pérdida de 0,951.

4. Modelo 3

Este modelo tiene los mismos componentes que el modelo 2, con la diferencia en la capa de *pooling* del bloque convolucional. En este modelo se usa la media de los valores en lugar del máximo. Se intenta así dar la misma importancia a todos los valores encontrados.

En la figura 6 se observa la evolución del entrenamiento a lo largo de las épocas de este modelo. Se puede observar que al iniciar el entrenamiento, la generalización no ha sido buena debido al cambio que se ha realizado (utilizar la media en lugar del valor máximo para *pooling*). Además de esto el valor de pérdida medio es el mayor de los modelos vistos hasta ahora.

Como resultado con el conjunto *Test* tenemos una precisión de 0,449 y una pérdida de 1,374

Model: "Model3"

Layer (type)	Output Shape	Param #
Conv1 (Conv2D)	(None, 124, 124, 20)	1520
Relu1 (Activation)	(None, 124, 124, 20)	0
Pool1 (AveragePooling2D)	(None, 62, 62, 20)	0
Conv2 (Conv2D)	(None, 58, 58, 20)	10020
Relu2 (Activation)	(None, 58, 58, 20)	0
Pool2 (AveragePooling2D)	(None, 29, 29, 20)	0
Conv3 (Conv2D)	(None, 25, 25, 20)	10020
Relu3 (Activation)	(None, 25, 25, 20)	0
Pool3 (AveragePooling2D)	(None, 12, 12, 20)	0
flatten_5 (Flatten)	(None, 2880)	0
Dense1 (Dense)	(None, 640)	1843840
Dense2 (Dense)	(None, 640)	410240
Dense3 (Dense)	(None, 640)	410240
Ouput (Dense)	(None, 4)	2564

=====
 Total params: 2,688,444
 Trainable params: 2,688,444
 Non-trainable params: 0

Figura 5: Resumen del modelo 3

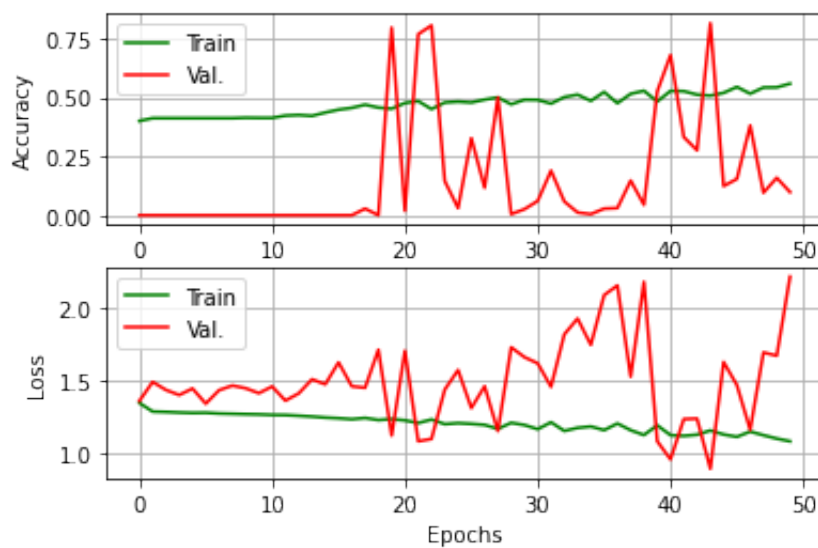


Figura 6: Entrenamiento del modelo 3

5. Modelo 4

Este modelo vuelve a basarse en el segundo modelo, pero añadiendo una capa *dropout* entre dos bloques convolucionales. Esta capa se ha probado en distintos sitios sin llegar a dar un cambio sustancial en el resultado final.

Model: "Model4"

Layer (type)	Output Shape	Param #
Conv1 (Conv2D)	(None, 124, 124, 20)	1520
Relu1 (Activation)	(None, 124, 124, 20)	0
Pool1 (MaxPooling2D)	(None, 62, 62, 20)	0
Drop1 (Dropout)	(None, 62, 62, 20)	0
Conv2 (Conv2D)	(None, 58, 58, 20)	10020
Relu2 (Activation)	(None, 58, 58, 20)	0
Pool2 (MaxPooling2D)	(None, 29, 29, 20)	0
Conv3 (Conv2D)	(None, 25, 25, 20)	10020
Relu3 (Activation)	(None, 25, 25, 20)	0
Pool3 (MaxPooling2D)	(None, 12, 12, 20)	0
flatten_1 (Flatten)	(None, 2880)	0
Dense1 (Dense)	(None, 640)	1843840
Dense2 (Dense)	(None, 640)	410240
Dense3 (Dense)	(None, 640)	410240
Ouput (Dense)	(None, 4)	2564

=====
Total params: 2,688,444
Trainable params: 2,688,444
Non-trainable params: 0

Figura 7: Resumen del modelo 4

En cuanto al entrenamiento el valor de precisión en el conjunto *Validation* es bastante bajo en todo el trascurso. Esto puede deberse al bajo número de imágenes, lo que hace que eliminar algunas neuronas sea contraproducente, ya que es difícil que el modelo sobreentrene.

Los resultados obtenidos con el conjunto *Test* son una precisión de 1,235 y una pérdida de 0,455

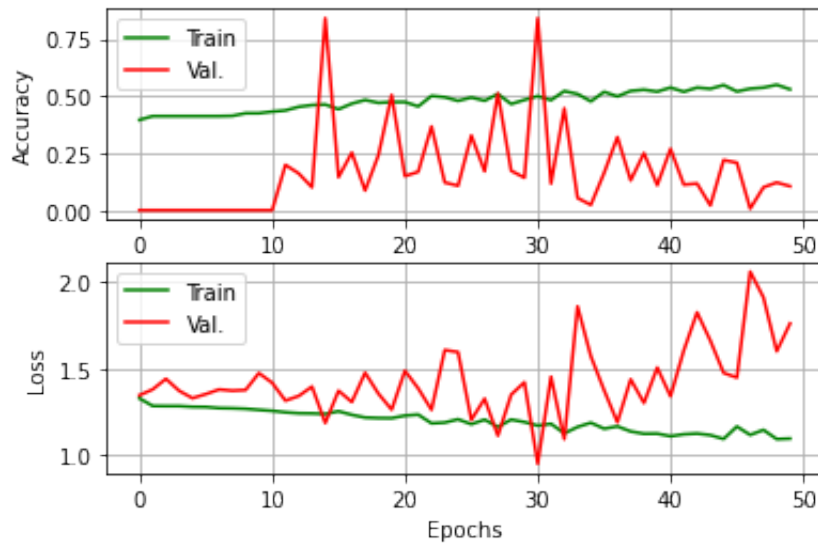


Figura 8: Entrenamiento del modelo 4

6. Conclusión

Para terminar se comparan los resultados tanto en precisión como en pérdida de los distintos modelos para obtener una idea clara de su desempeño y poder elegir de manera correcta el mejor modelo.

	Precisión	Pérdida
Modelo 1	0.599	0.804
Modelo 2	0.615	0.951
Modelo 3	0.449	1.374
Modelo 4	0.455	1.235

Cuadro 1: Tabla comparativa

En el cuadro 1 se puede observar como el modelo 3 y 4 deben ser descartados ya que su precisión es muy inferior al de los otros modelos, y su pérdida muy superior. Por otro lado, el modelo 2 tiene el valor de precisión más alto, aunque con una diferencia mínima con respecto al modelo 1. A su vez, la pérdida del modelo 2 es superior en 0,15 al modelo 1. Esta diferencia de perdida y la proximidad en su valor de precisión hace que el modelo 1 sea el más adecuado.