

CONVOLUTIONAL NEURAL NETWORK

TRANSFER LEARNING

Carlos Castillo Rodríguez

ccastillo.py@gmail.com

1.- INTRODUCCIÓN

Este proyecto consiste en llevar a cabo el manejo de imágenes y entrenamiento de modelos de redes neuronales convolucionales a través de la técnica conocida como transfer learning.

Este método se basa en importar una red existente y ya entrenada a partir de un conjunto de datos iniciales (imágenes en este caso) y aprovechar su distribución y configuración de capas para predecir las etiquetas de otro conjunto de datos diferentes.

En particular, se trata de congelar todas las capas y parámetros (pesos y bias) de los modelos precargados, modificando únicamente la última capa de salida para adaptarla a las necesidades del problema a resolver.

2.- OBJETO

Partiendo del dataset “Dogs & Cats Images” de la plataforma Kaggle, se persigue generar modelos específicos a partir de la arquitectura de modelos convolucionales de referencia y, de este modo, realizar predicciones sobre las etiquetas de las imágenes que contiene para determinar si lo que se presenta se corresponde a un perro o a un gato. Se trata pues de un problema de clasificación binaria.

Los modelos de referencia utilizados para la confección del transfer learning son:

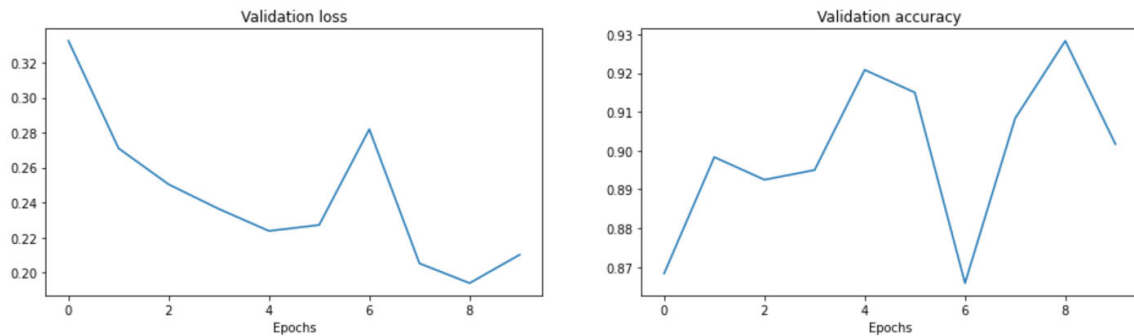
- VGG16, red con más de 138 millones de parámetros
- MobilNet V2, red con más de 3,5 millones de parámetros
- ResNet V2, red con 23,5 millones de parámetros

Todos estos modelos han sido pre-entrenados utilizando el dataset de ImageNet como datos train.

3.- RESULTADOS

3.1.- RED VGG16

Analizando el modelo para el conjunto de datos de Perros y Gatos de Kaggle, se obtienen las siguientes gráficas de pérdida y precisión para los datos de validación:



Atendiendo a las gráficas, se comprueba que la pérdida de validación presenta valores irregulares, aunque con una tendencia a la reducción en todo momento. Sin embargo, se observa que comienza a aumentar justo a partir de la octava epoch; además, se comprueba que la precisión de la validación comienza a reducirse a partir de la misma época.

Esto sugiere que nuestro modelo comienza a sobreajustarse en ese punto y que el mejor rendimiento se alcanza deteniendo el entrenamiento antes. El óptimo se conseguiría, por tanto, realizándose 8 epochs.

En general, aplicando Transfer-Learning se obtiene una precisión superior al 90% por lo que podemos concluir que el modelo VGG16 aplicado al dataset objeto de estudio da unos buenos resultados.

Una vez entrenado el modelo, se realiza la evaluación del mismo para los datos de Test, obteniéndose:

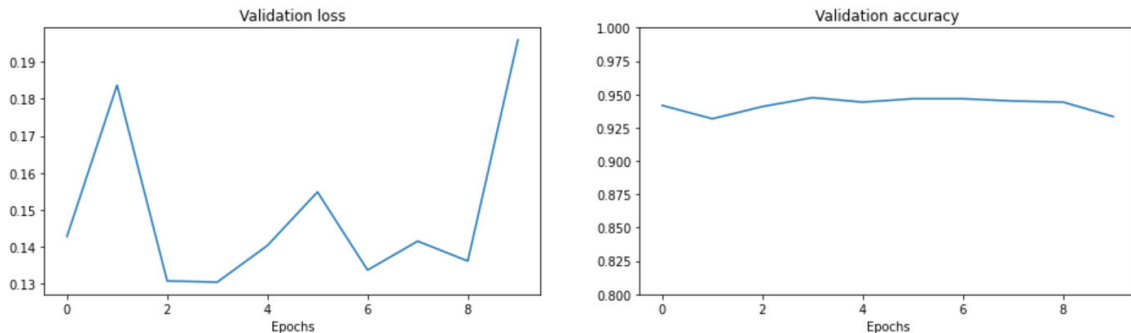
Pérdida sobre los datos Test: 23,44%

Precisión sobre los datos Test: 89,30%

Lo que arroja unos valores bastante próximos a los obtenidos para los datos de validación.

3.2.- RED MOBILENET V2

Analizando el modelo para el conjunto de datos de Perros y Gatos de Kaggle, se obtienen las siguientes gráficas de pérdida y precisión para los datos de validación:



Según las imágenes mostradas se observa que la pérdida de validación comienza a aumentar justo a partir de la octava epoch; además, se comprueba que la precisión de la validación máxima se obtiene en esa misma época, alcanzando un valor del 95%.

Esto sugiere que nuestro modelo comienza a sobreajustarse en ese punto y que el mejor rendimiento se alcanza deteniendo el entrenamiento antes. El óptimo se conseguiría, por tanto, realizándose 8 epochs.

En general, aplicando Transfer-Learning se obtiene una precisión de aproximadamente el 95% tanto para los datos de entrenamiento como para los datos de validación, por lo que podemos concluir que el modelo MobilNet aplicado al dataset analizado arroja resultados muy precisos.

Una vez entrenado el modelo, se realiza la evaluación del mismo para los datos de Test, obteniéndose:

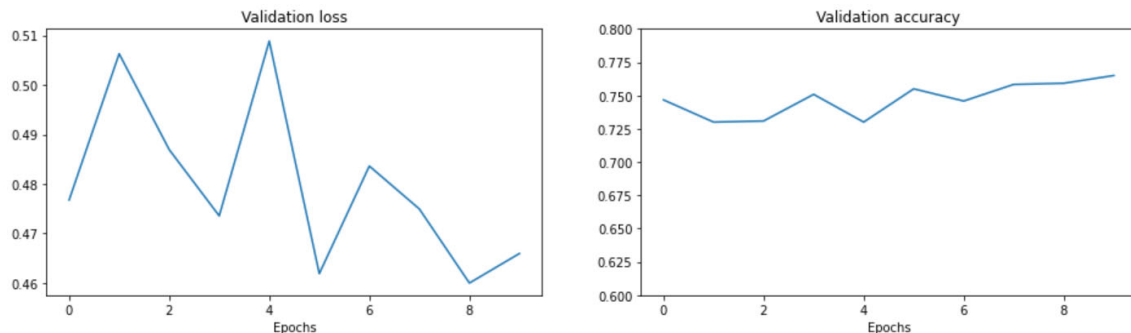
Pérdida sobre los datos Test: 18,18%

Precisión sobre los datos Test: 93,65%

Lo que arroja unos valores bastante próximos a los obtenidos para los datos de validación.

3.3.- RED RESNET50 V2

Analizando el modelo para el conjunto de datos de Perros y Gatos de Kaggle, se obtienen las siguientes gráficas de pérdida y precisión para los datos de validación:



Atendiendo a las gráficas arriba mostradas, la pérdida de validación obtiene su mínimo valor en la octava epoch. Del mismo modo, para esa misma epoch se alcanza la mayor precisión del modelo. A partir de dicha epoch todo parece indicar que se reduce la precisión, así como se aumenta la pérdida de validación, lo que se traduce en sobreajuste, aunque serían necesarias más epochs para terminar de determinar ese escenario con acierto.

En general, aplicando Transfer-Learning se obtiene una precisión alrededor del 80% para los datos de entrenamiento y del 75% para los de validación.

Una vez entrenado el modelo, se realiza la evaluación del mismo para los datos de Test, obteniéndose:

Pérdida sobre los datos Test: 46,96%

Precisión sobre los datos Test: 77,45%

Lo que arroja unos valores bastante próximos a los obtenidos para los datos de validación.

4.- CONCLUSIONES

Se ha comprobado que aprovecharse de arquitecturas preestablecidas facilita enormemente el desarrollo de nuevos modelos predictivos al reutilizarse un gran porcentaje de las capas y parámetros de dichos modelos de referencia.

Para nuestro caso particular, se ha podido observar que se han conseguido unas predicciones considerablemente acertadas aplicando los modelos de referencia VGG16 y MobileNet V2. En el apartado de ResNet, por desgracia, la precisión obtenida es ligeramente inferior.

De todos los modelos estudiados, el que mejores resultados ha dado ha sido el modelo MobileNetV2 que, además, ha requerido de menos tiempos de procesamiento con diferencia. Esto es debido a que, además de contar con una estructura más sencilla, también presenta muchos menos parámetros que las demás. Se comprueba que una red más compleja no implica necesariamente rendimientos y precisiones mayores.

Como propuesta de mejora sobre los modelos ensayados, sería recomendable aplicar algunas técnicas adicionales tales como:

- Early Stopping, a fin de determinar el número máximo de intervalos de entrenamiento que se pueden asumir antes de detener el aprendizaje y atajar los escenarios de overfitting.
- Herramientas de regularización como Dropout (donde se apagan algunas capas para evitar que el modelo aprenda demasiado rápido y alcance el sobreajuste) o Data Augmentation, con objeto de añadir mayor volumen de datos de entrada y dificultar el aprendizaje de los modelos.