Práctica 5: Keras

Evelyn G. Coronel
Redes Neuronales y Aprendizaje Profundo para Visión Artificial
Instituto Balseiro

(3 de noviembre de 2020)

EJERCICIO 3

En este ejercicio se utilizó la red `MobileNet` [1] seguida de una capa densa de 10 valores de salida. Se utilizó está red por ser una red convolucional profunda con una arquitectura liviana. Esta red ya está implementa en `Keras` y tiene la opción de inicializarla con pesos ya entrenados con el conjunto de datos *imagenet*. Este conjunto consiste en imágenes clasificadas en 1000 categorías. En este ejercicio se estudió la posibilidad de usar los pesos pre-entrenados para clasificar la base de datos del CIFAR-10.

La red `MobileNet` alcanza un precisión del 70.6 % con *imagenet*[1], es de esperar que la red obtenga una precisión similar con el CIFAR-10. En las Figs.1 y 2 se muestran las curvas de entrenamiento (E) y validación (V) para distintas condiciones de entrenamiento.

Primeramente, utilizando la red `MobileNet` con pesos aleatorio, luego se entrenó el modelo implementado durante 60 épocas mediante el optimizador Adam con una tasa de entrenamiento de 10^{-4}, todo esto con la función de costo del `CategoricalCrossentropy` y la métrica `CategoricalAccuracy`. Para este caso la red llegó a una precisión de 35.4 % después de 60 épocas, esto es de esperarse ya que la red empieza sin noción de los pesos en las capas convolucionales.

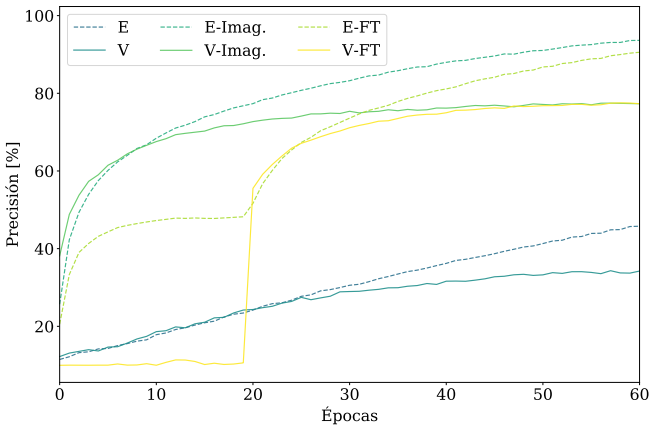

Fig. 1: Precisión en función de las épocas

En segunda instancia, con los mismos parámetros del primer entrenamiento se inicializaron los pesos de la red convolucional ya entrenados con *imagenet*. En las figuras anteriores, estas curvas se identifican con *Imag.*, se observa además que la red alcanza una mejor precisión que el caso anterior como era de esperarse, donde la red empieza con una precisión alta del 38 % y llega al orden del 78 % de precisión. Las redes convolucionales con los pesos ya entrenados tiene la facilidad de reconocer los features propios de imágenes naturales.

Por último se entrenó la red usando *Fine-tuning*. En este trabajo, este proceso consistió en entrenar las últimas dos capas de la red durante 20 épocas con las mismas condiciones que los casos anteriores, luego se permite a la red en modificar los pesos de las capas convolucionales durante 40 épocas con una tasa de aprendizaje de 10^{-5}. En las Figs.1 y 2 las curvas que corresponden a este caso se identifican con *FT*. Este trabajo esperaba que con los pesos ya entrenados la red alcanzaría una mejor precisión superior al primer caso, pero se observó que no sucedió en el caso de la curva de validación, esto puede deberse a que los features que aprendió la red clasificando *imagenet* en 1000 categorías confunde a las últimas capas que buscan una clasificación de solamente 10. Una vez que la red puede modificar los pesos de las capas convolucionales, la precisión a partir de la época 40 aumenta considerablemente y llega a la misma precisión que el segundo caso.

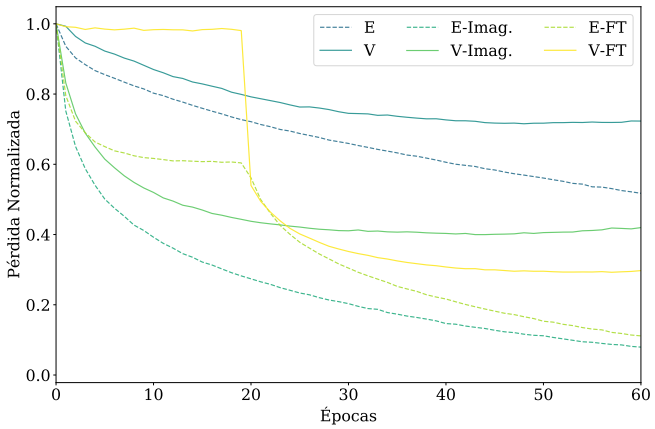

Fig. 2: Pérdida en función de las épocas

Este ejemplo de clasificar las imágenes del CIFAR-10 alcanza una precisión alta dado que la red está entrenada en reconocer features en imágenes naturales. Si utilizáramos una red pre-entrenada en reconocer features en lenguaje natural u otro conjunto de datos, la red tendría una pre-

cisión pobre en la misma cantidad de épocas con respecto a la utilizada en este ejercicio.

EJERCICIO 4

Siguiendo con la red utilizada en el ejercicio anterior, ahora se busca visualizar lo que la red va analizando durante las capas convolucionales. En particular, nos centramos en lo que se observa en las antepenúltima y penúltima capa convolucional.

Para lograr esto se alimentó la red con tres tipos de imágenes: una imagen con píxeles de intensidad aleatoria que se muestra en la Fig.3, otra con píxeles que sigue un patrón sinusoidal como la Fig.5 por último una imagen natural, en este caso el perro de la Fig.7. Las imágenes de los que se observa en las capas convolucionales son las Figs. 4, 6 y 8 respectivamente, las imágenes de la izquierda y derecha corresponden a la antepenúltima y última capa.

Para realizar este ejercicio, este trabajo se basó en el ejemplo implementado por F. Chollet que se encuentra en la página de Keras [2].

Primeramente se instancia un red que tenga como entrada la misma que la red `MobileNet` y que tenga como salida la penúltima capa de la misma red. Una vez elegida las dimensiones de las imágenes de entrada, se aplica un algoritmo similar al gradiente descendente donde ahora se intenta maximizar la función de costo. La función de costo en este problema es la media entre la imagen de entrada y la imagen de salida en cada capa.

El algoritmo de gradiente ascendente se realiza con una tasa de aprendizaje de 10 que está por encima de las utilizadas usualmente, y se maximiza la función de pérdida durante 50 épocas. Para obtener las Figs. 4, 6 y 8 se realiza el proceso inverso al preprocesado y se transforman las imágenes a RGB.

Con la imagen aleatoria de la Fig.3 se obtiene la Fig.4, donde se observan patrones en la parte del centro de la imagen, esto puede deberse a que la mayor parte de la información de las imágenes se encuentran en esta área.

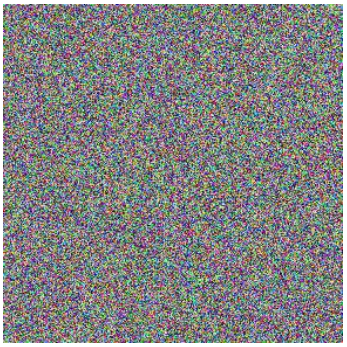

Fig. 3: Imagen con valores aleatorios en los tres canales RGB.

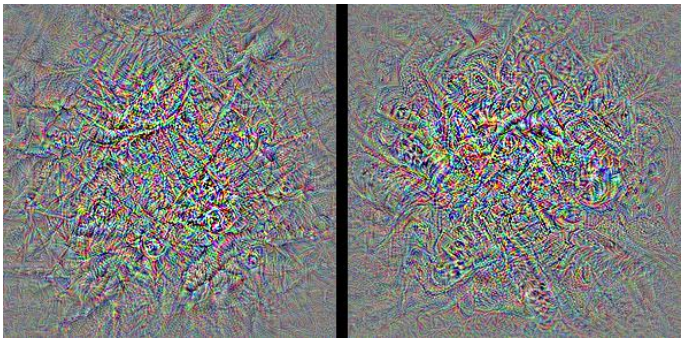

Fig. 4: Imágenes obtenidas mediante el gradiente ascendente a a partir de la imagen aleatoria.

Con la imagen con patrón sinusoidal de la Fig.5 se obtiene la Fig.6, donde además de observarse los patrones en la parte del centro de la imagen, la red intenta obtener información sobre las líneas de la imagen original. Este es el proceso por el cual la red convolucional intenta reconocer features en las imágenes.

Fig. 5: Imagen con un patrón sinusoidal en las componentes RGB.

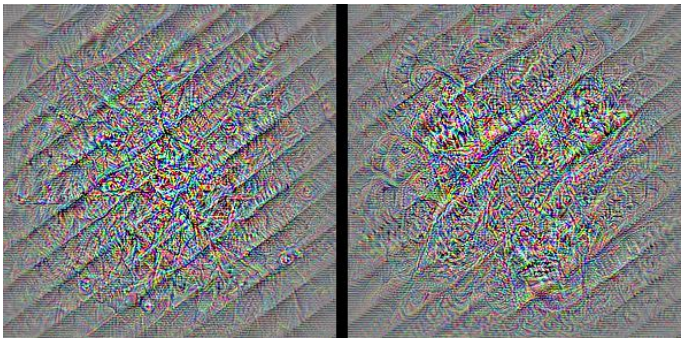

Fig. 6: Imágenes obtenidas mediante el gradiente ascendente a a partir de la imagen con el patrón sinusoidal.

Con la imagen del animal de la Fig.7 se obtiene la Fig.8. Esta imagen de entrada es un ejemplo de como la red

intenta obtener información sobre el outline sobre la parte central de la imagen original. Por ejemplo, en este caso la red omite información del fondo del animal, esto es posible después del entrenamiento de las capas.

Fig. 7: Imagen de un perro del conjunto de datos dogs vs cats.

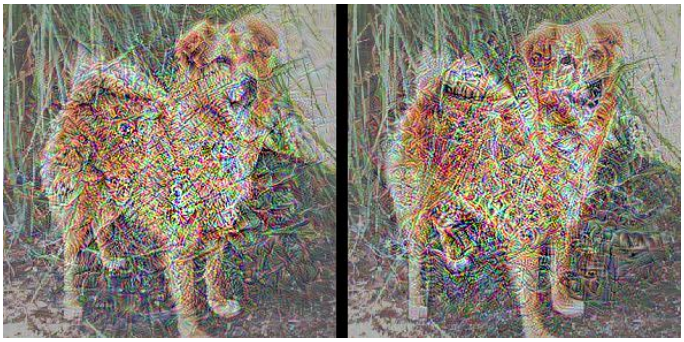

Fig. 8: Imágenes obtenidas mediante el gradiente ascendente a a partir de la imagen de un perro.

[1] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, CoRR **abs/1704.04861** (2017), arXiv:1704.04861.
[2] F. Chollet, Visualizing what convnets learn .