Memoria de practicas de RNA

Raúl García Crespo

December 2020

1 MNIST

El objetivo de esta primera practica es el de entrenar una red neuronal con una estructura MLP a partir del dataset MNIST, el cual está compuesto por imágenes de 10 clases de dígitos de dimensiones 28x28 píxeles.

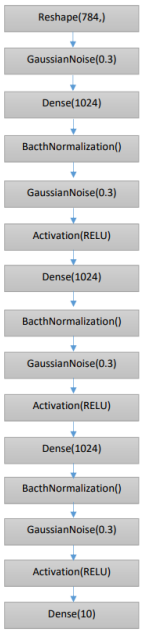

Figure 1: Arquitectura usada

Para el entrenamiento de la red se ha empleado el entorno que ofrece Google Colabs y como plantilla de inicio para la red que vamos a modificar se ha empleado este fichero, el cual ofrece un porcentaje de acierto sobre el dataset MNIST del 98.9%. Esta red inicial se compone de 1 capa de entrada, 3 capas ocultas y una capa de salida, las cuales incluyen Batch Normalization, Gaussian Noise y funciones de activación "relu". En esta red también se ha implementado un proceso de data augmentation, con un desplazamiento en los margenes de la imagen del 10%. A partir de esta plantilla se han ido realizando experimentos con el propósito de aumentar este porcentaje el porcentaje de acierto hasta el 99.2%. Los cambios y los resultados obtenidos son los siguientes:

Cambios en la estructura de la red : Se ha modificado la dimensión de las capas ocultas de la red, experimentando con distintas combinaciones, pero ninguna consiguió un porcentaje de error inferior al obtenido a partir de la estructura de partida.

Modificaciones sobre el proceso de Data Augmentation: Se ha incluido la posibilidad de rotar las imágenes de entrenamiento 10 grados junto al desplzamiento lateral, superior e inferior del 10%, obteniendo un porcentaje de acierto del 98.96%. Otros intentos en los que se ha aumenta la magnitud del desplazamiento han llevado a un descenso del acierto.

Disminución del ruido Gaussiano: Simplemente con reducir el parametro de las capas Bacth Normalization de 0.3 a 0.2 se ha conseguido de forma consistente alcanzar un porcentaje de acierto del 99.2%. Un valor de 0.1 en este parámetro sin embargo ofrece mejores resultados que la red de partida pero sin llegar al porcentaje de acierto objetivo. La gráfica obtenida para el mejor resultado es la siguiente:

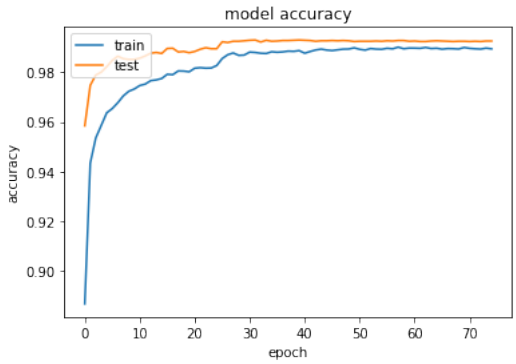

Figure 2: Evolución del entrenamiento sobre mnist

2 CIFAR10

En esta segunda práctica se pretende desarrollar una red convolucional entrenada con el dataset CIFAR10 cuyo porcentaje de acierto se sitúe por encima del 90%. De nuevo, se ha empleado como plantilla inicial la red que puede encontrase en este fichero. Esta red de partida proporciona un porcentaje de acierto en torno al 84.42% sobre el dataset CIFAR10 y su arquitectura se compone de bloques de capas con los siguientes elementos: capa convolucional, batch normalization, gaussian noise, función de activación "relu" y maxpooling; la red incluye también un preproceso para incluir data augmentation similar al empleado en la practica anterior. A partir de esta red inicial se han realizado modificaciones con el fin de elevar su porcentaje de acierto. Estas son las modificaciones realizadas:

Reducción del batch size: La primera modificación que se realizó sobre la red de partida fue la disminución del tamaño de los batches, los cuales ini-

cialmente tomaban el valor 100. Reducciendo este valor hasta 32 encontramos que el resultado del entrenamiento porporciona ahora un 85% de acierto.

Eliminación del ruido Gaussiano: En los distintos experimentos realizados, los mejores resultados suelen obtenerse con valores muy bajos para las capas de ruido Gaussiano o incluso en ocasiones sin incluir estas capas. Partiendo de las modificaciones del tamaño del batch y aplicando un valor de 0.2 a los parámetros de las capas de ruido Gaussiano se obtuvo un porcentaje de acierto del 86.9%.

Data augmentation: Adicionalmente al desplzamiento horizontal y vertical, y a emplear imagenes reflejedas horizontalmente para generar nuevas imagenes de test, se han añadido la posibilidad de ampliar o reducir la imagen con "zoom_range = 0.2" y la posibilidad de rotar la imagen 10 grados como máximo con "rotation_range = 10".

Aumentar el número de convoluciones por bloque: Inicialmente cada bloque de la red neuronal se componia de una capa convolucional, un "batch normalization", una capa de ruido Gaussiano, una función de activación "relu" y por último un "maxPool2d". Duplicando las capas previas al "maxPooling" en el mismo orden se consigue un aumento muy significativo del porcentaje de acierto, llegando a alcanzar el 90%. En posteriores experimentos y en el caso de la mejor red obtenida se ha llegado incluso a triplicar el número de convoluciones por bloque lo que, con otros cambios necesarios, llega a proporcionar un porcentaje de acierto del 91.5%.

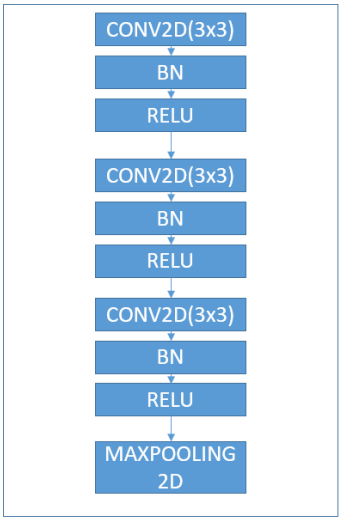

Figure 3: Estructura interna de los bloques convolucionales usados

Cambios en el "learning rate scheduler": El "learning rate" con el que estabamos trabjando hasta ahora tan solo realizaba 3 reducciones, desde 0.1 a 0.001 en 50 epochs con un "learning rate scheduler". Empleando en su lugar un "scheduler del tipo "ReduceLROnPlateau" que reduzca el "learning rate" en un orden de 0.1 hasta un mínimo de 0.00001 se llegan a alzcanzar de forma consistente valores superiores al 91%.

Reinicio del entrenamiento: Realizando dos procesos contiguos de entrenamiento de los pesos de la red, es decir, obtener unos pesos como resultado de un primer entrenamiento y emplear dichos pesos como el estado inicial de la red en un segundo entrenamiento se ha llegado a la meta de esta práctica del 92% de acierto. Este proceso no es arbitrario ni equivalente a realizar un solo entrenamiento con mayor número de epochs, pues con esto conseguimos reiniciar el "learning rate" en mitad del proceso, así como introducir nuevas imágenes de entrenamiento gracias al imageDataGenerator el cual se invoca con cada proceso de entrenamiento. Todo esto nos permite afinar mucho más la red sin abusar de sobre-entrenamiento.

Implementación de "mix-up generator": Basado en el código que podemos encontrar en este repositorio se implementó un "mix-up generator" que funcionase en conjunto con la implementación que ya teníamos del "data augmentation". Sin embargo experimentando con distintos valores de alfa para este generador no se consiguió ninguna mejora de la precisión de la red con respecto a la misma red sin "mix-up", por lo que no se acabo empleando en el mejor resultado obtenido.

Para resumir, la red que ha alcanzado los mejores resultados posee:

- Batch size = 32/16 y epochs = 50/75.
- Sin ruido Gaussiano.
- Data augmentation con desplazamientos horizontales y verticales, zoom, rotaciones e imágenes invertidas.
- Una evolución del "learning rate" empleando "ReduceLROnPlateau".
- 3 convoluciones por bloque manteniendo el número de filtros.
- Reinicio del "learning rate" tras 50 epochs.

2.1 Resultados

Como ya se ha comentado, se han realizado dos entrenamientos con el propósito de reiniciar el valor del "learning rate". La primera gráfica muestra la evolución de la red en la primera fase de entrenamiento, y la segunda muestra la evalución de la red a partir del reinicio del "learning rate":

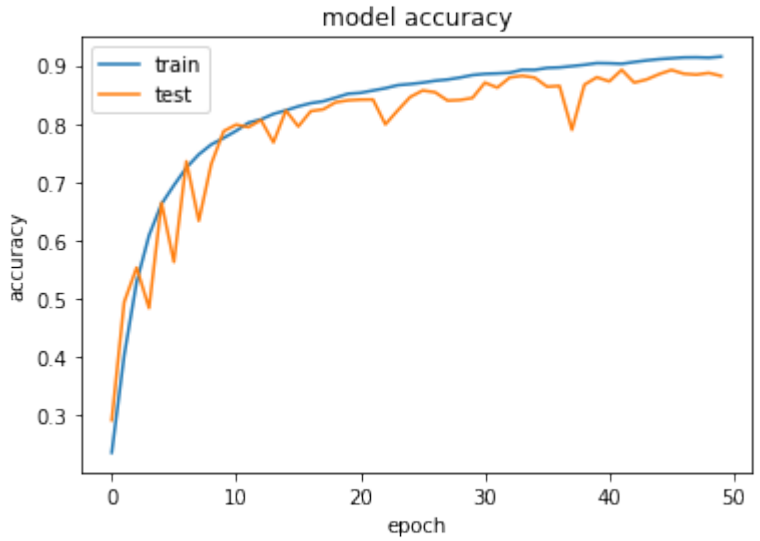

Figure 4: Evolución de la 1ª mitad del entrenamiento sobre cifar10

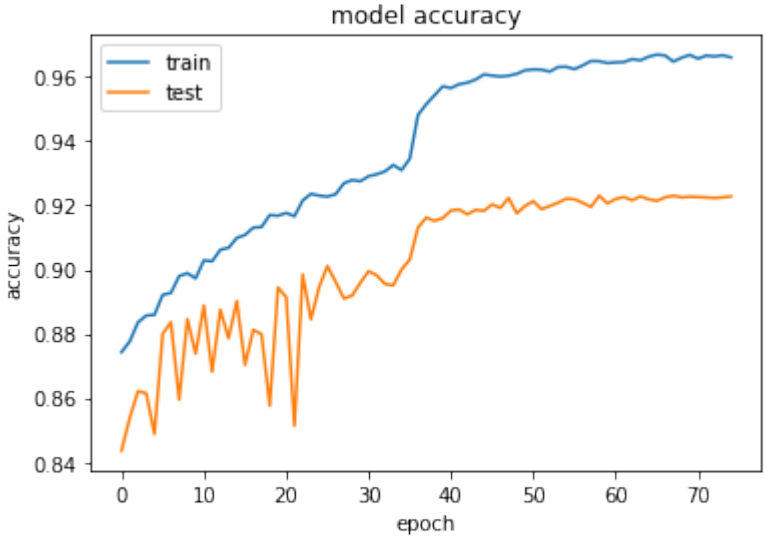

Figure 5: Evolución de la 2ª mitad del entrenamiento sobre cifar10