

Práctica Autoencoders

Javier Franco y Pablo Guilló

Introducción

En esta práctica se crea un supresor de ruido para las imágenes del conjunto de imágenes “MNIST”. Se añade ruido a las imágenes de este conjunto, y mediante el uso de autoencoders, se trata de suprimir dicho ruido de forma que las imágenes reconstruidas se parezcan lo máximo posible a las originales. En la segunda parte se trata de hacer lo mismo con una versión de la imagen con menor dimensión, pero sin ruido.

Metodología

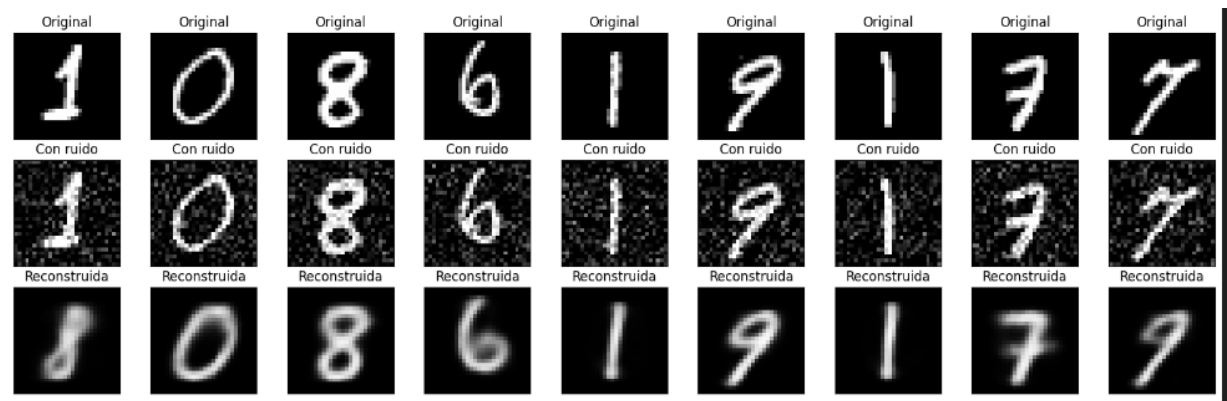
Se diseñó un autoencoder con una arquitectura simétrica. Tras probar con diferentes tipos de capas, se acabó usando un autoencoder lineal que reduce las dimensiones de la imagen hasta 5 veces, con capas de batchnormalization y usando LeakyRelu como función de activación, que favorecía bastante el aprendizaje. En la parte de la súper resolución, el autoencoder contaba con un menor número de capas, y se usó Relu en lugar de LeakyRelu.

Para el proceso de entrenamiento, se probaron numerosas funciones de pérdidas y optimizadores distintos, pero con los que se obtuvieron mejores resultados fueron BCELoss como función de pérdida y Adam como optimizador. Fue un modelo de entrenamiento habitual, añadiendo ruido a las imágenes o reduciendo su dimensionalidad al principio de este para poder pasárselas como parámetro al modelo. De esta manera se calculaba la pérdida (mostrándola después de cada epoch) y se realizaba el Backpropagation correctamente.

Por último se imprimía una muestra de las imágenes conseguidas tras la reconstrucción, comparándolas con las imágenes originales y con las imágenes modificadas. Esto servía para comprobar visualmente la eficacia del modelo, y no depender solo de lo que dictaba la función de pérdida.

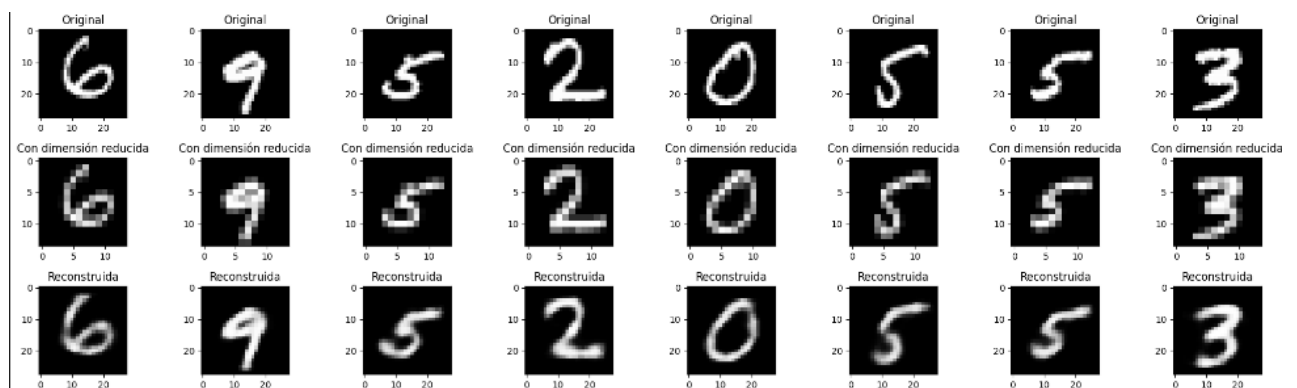
Resultados

Como resultado del entrenamiento, las imágenes reconstruidas eran similares a las originales y se mostraba el mismo número, aunque no acertaba siempre. Se consiguió reducir la función de pérdida a lo largo de las iteraciones de entrenamiento, aunque no en gran medida, hasta que el valor de pérdida llegó a 0.15.



Viendo las imágenes, estaba claro que el supresor hacía un buen trabajo ya que en la mayoría de los casos mostraba el mismo número escrito de la misma manera. No obstante, las imágenes reconstruidas eran mucho menos nítidas, lo que no impedía que los números fuesen reconocibles, pero tampoco es el resultado óptimo.

En el apartado de la súper resolución, los resultados fueron mejores, con imágenes mucho más nítidas y con una pérdida que bajaba de 0.02.



Vemos en que en esta segunda parte, los números ya no presentan esa cualidad borrosa, y se asemejan en gran medida a las imágenes originales

Conclusiones y trabajo futuro

La implementación de un supresor de ruido para imágenes del conjunto MNIST ha demostrado ser viable usando autoencoders, logrando reducir la función de pérdida y mostrando los mismos números que en las imágenes originales, aunque la nitidez de las imágenes reconstruidas podría mejorar. La segunda parte se ha beneficiado mucho más del uso de los autoencoders, mostrando una reconstrucción muy fiel a las imágenes originales.

Hay margen de mejora, y el trabajo futuro se podría enfocar en probar el modelo con otros conjuntos de datos, además de experimentar con distintas capas, funciones de pérdida, optimizadores y los distintos parámetros que puedan tener estas herramientas.