

1. Explicación del problema elegido.

El problema es que necesitamos clasificar imágenes de forma automática y con alta fiabilidad. Esto no se puede lograr con programación convencional ya que tenemos que leer los píxeles de las imágenes y estas pueden estar movidas, rotadas, difuminadas, etc... Por lo que debemos usar un modelo de inteligencia artificial.

Podemos usar varios tipos de modelos para esta tarea. En esta práctica utilizaremos las RNA (Redes Neuronales Alimentadas hacia adelante), debido a su capacidad de aproximación universal y las CNN (Convolutional Neural Networks), ya que son más eficientes en el procesamiento de imágenes.

Este problema es importante ya que puede ser utilizado no solo para clasificar números o ropa, sino también para reconocer obstáculos en vehículos de conducción automático, detección de patologías en el campo médico, en sistemas de seguridad, sistemas de reconocimiento facial, y demás, que dentro de unos años serán completamente necesarios y realizados de forma automática para lograr resultados más rápidos y fiables (Eliminando el factor humano).

2. Implementación de una RNA Feedforward.

Para realizar predicciones sobre la base de datos MNIST, hemos construido dos modelos de RNA Feedforward.

El primero consta de 1 capa oculta y 128 neuronas con función de activación ReLU, y el segundo con 3 capas ocultas de 16 neuronas cada una. Los dos modelos tienen 784 neuronas de entrada (Imágenes de 28x28, escala de grises) y 10 neuronas de salida (1 por cada categoría) con función de activación softmax. Para ambos modelos hemos utilizado el optimizador ADAM y la función de pérdida de Entropía Cruzada Categórica. Hemos elegido esta función de pérdida ya que es una elección común para problemas de clasificación múltiple. Esta función de pérdida compara la distribución de probabilidad predicha por el modelo con la distribución real de las etiquetas.

Hemos entrenado los modelos de forma iterativa y hemos llegado a la conclusión de que para la tarea específica de clasificación de dígitos en el conjunto de datos MNIST, este modelo es adecuado y proporciona resultados razonables. Sin embargo, para tareas más complejas o conjuntos de datos más desafiantes, podría ser necesario explorar arquitecturas más avanzadas y realizar ajustes adicionales para lograr un rendimiento óptimo. En general, la elección del modelo depende de la naturaleza de la tarea y la complejidad de los datos.

3. Implementación de una Red Convolutiva.

Para comprobar si existe una mejora de rendimiento utilizando redes CNN en lugar de RNA, hemos construido otros dos modelos, esta vez con la topología de redes convolutivas.

Para el primer modelo hemos utilizado una capa convolutiva 3x3 con 32 filtros, una capa max pooling 2x2, una capa plana y una capa densa de 32 neuronas, y para el segundo modelo hemos creado una capa convolutiva 3x3 con 32 filtros, una capa max pooling 2x2, otra capa convolutiva 3x3 con 64 filtros, otra capa max pooling 2x2, una capa convolutiva más de 3x3 con 64 filtros, una capa plana y una capa densa de 64 neuronas. Ambos modelos utilizan la función de activación ReLU para las capas convolutivas y densas, y tienen 784 neuronas de entrada y 10 neuronas de salida (Las mismas que los modelos RNA). El segundo modelo tiene más complejidad que el

primero, ya que hemos añadido un mayor número de filtros, lo que le añade capacidad de aprendizaje al modelo.

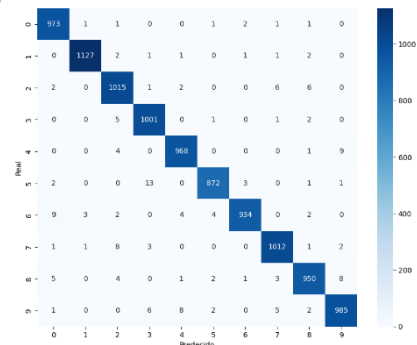
Tal como en los modelos RNA, hemos utilizado el optimizador ADAM y la función de pérdida de Entropía Cruzada Categórica, por las mismas razones anteriormente mencionadas.

Hemos comparado los resultados obtenidos para ambos modelos. Vamos a valorar la exactitud para decir que un modelo es mejor que otro. Los CNN han resultado ser mucho más exactos que los RNA, con una precisión de $\approx 97\%$ para los RNA y del $\approx 98\%$ para los CNN (Siendo las variantes más complejas las que obtienen mejores resultados, con la contraparte de ser más costosos computacionalmente).

4. Evaluación de la capacidad de entrenamiento y test de la Red Convolutiva.

Si evaluamos los modelos calculados anteriormente, podemos observar que cuanto más complejo es la arquitectura del modelo, existe una precisión mayor, una función de pérdida menor y el tiempo en que tarda en desarrollarse el modelo en cuestión será mayor. Por lo tanto, podemos observar que si la red convolutiva es bastante compleja nos dará una precisión mayor, pero esto hará que el modelo sea más lento. En nuestro caso, como nos importa más que sea preciso y el tiempo no es excesivo para que nos bloquee, nos parece un buen modelo.

Por otro lado, si vemos la matriz de confusión, vemos que se genera una diagonal de colores muy oscuros comparados con los demás cuadrados blancos. Esto es positivo porque nos indica que está siendo bastante preciso. Podemos observar que puede encontrar problemas con números parecidos como pueden ser 0 y 6, 7 y 2 o 5 y 3.



5. Posibles mejoras.

Una posible mejora para intentar ayudar al modelo detectar de una manera más correcta los números en cuestión, sería poner más ejemplos de entrenamiento para los números que tenga más problemas de detectar (Por ejemplo, más 0 y 6), y decrementar aquellos en los que el modelo tenga menos problemas (Por ejemplo, el 1).

Otra posible mejora pensada puede ser que se añadan imágenes rotadas, movidas, o incluso con imágenes con ruido, para poder hacer que el modelo no se sobreajuste y tenga bastante más flexibilidad.

Reducir el tiempo de ejecución, sería bastante importante, por lo que quedarnos con el primer modelo de red convolucional no sería tan mala idea ya que la precisión, aunque sea menor, es bastante parecida y el tiempo de ejecución es la mitad comparado con la más compleja.