

Métodos usados en el entrenamiento de la red neuronal

Se introducirán brevemente algunos elementos que fueron claves para el diseño y entrenamiento de la red neuronal. Cada uno de estos componentes jugó un papel esencial en el rendimiento, la estabilidad y la eficiencia del modelo.

1. Inicializador HeNormal. La inicialización adecuada de los pesos es crucial para evitar problemas como el desvanecimiento del gradiente o explosión del gradiente. Este método inicializa los pesos de la red con valores aleatorios siguiendo una distribución normal (gaussiana).

Se hizo uso de este inicializador ya que fue diseñado específicamente para redes que utilizan funciones de activación ReLu, como es el caso.

2. Regularizadores. Son técnicas usadas utilizadas para prevenir el sobreajuste (*overfitting*), que ocurre cuando un modelo se ajusta demasiado a los datos de entrenamiento y por ende no reconoce o generaliza a nuevos datos.

2.2 Regularización L2. La regularización Ridge o decaimiento de peso, funciona penalizando los valores grandes de los pesos del modelo al añadir un término de penalización a la función de pérdida.

2.3 Dropout. Es otro método de regularización que desactiva aleatoriamente algunas neuronas durante el entrenamiento. Esto ayuda a, justamente, regularizar el uso de las neuronas presentes en el entrenamiento y que así el modelo no dependa demasiado de algunas de ellas en específico.

3. Funciones de activación. Son funciones matemáticas que transforman la salida lineal de una neurona en una forma no lineal. Esto es importante porque le otorga a la red la flexibilidad de aprender relaciones complejas.

3.1 ReLu (Rectified Linear Unit) Simplemente hace que para valores de entrada positivos la salida es igual al valor de la entrada, mientras que para valores negativos en la entrada la salida es 0. Esta función de activación es ampliamente usada gracias a su simplicidad computacional, lo que acelera el entrenamiento del modelo.

3.2 Softmax. Se usa específicamente en la última capa de las redes neuronales convolucionales diseñadas para la clasificación. Esto gracias a que convierte los valores de la salida en probabilidades normalizadas.

4. Optimizador Adam. Son algoritmos que se encargan de ajustar los pesos y sesgos de la red neuronal durante el proceso de entrenamiento para minimizar lo máximo posible la función de pérdida.

Adam (*Adaptive Moment Estimation*), es de los optimizadores más utilizados para el entrenamiento de redes neuronales profundas, esto se debe básicamente a su fácil implementación, adaptabilidad y eficiencia, pues tiene un bajo costo computacional y requiere menos memoria, algo de vital importancia al momento de diseñar una red neuronal para usarse en un microcontrolador.

5. Callbacks. son herramientas que monitorizan y controlan el entrenamiento en tiempo real.

5.1 EarlyStopping. Básicamente detiene el entrenamiento del modelo de forma anticipada si no se presenta una mejora en la métrica seleccionada. Esto es de vital importancia para prevenir el *overfitting* y ahorrar tiempo al momento de realizar múltiples pruebas para mejorar la red neuronal.

5.2 ModelCheckpoint. Permite guardar el modelo en puntos específicos al encontrar la mejor métrica de validación durante el entrenamiento. Lo que facilita la utilización del mejor modelo entrenado al momento de evaluarlo en los datos de prueba.