Hugo Arnoldo Oliva. Procesamiento de Datos

Modelo:

Debido a que se plantea tratar de clasificar la emoción imperante en dos audios con etiquetas de “triste” o “feliz”, se recurrión a un modelo de Redes Neuronales Convolucionales (RNC), y el porque de esto es el siguiente: una manera adecuada de representar la información de un audio en su totalidad, incluyendo las frecuencias de onda y sus amplitudes a través del eje del tiempo del audio es el espectograma, el cual es una especie de gráfico que puede ser exportado a imagen. Por lo tanto, se hicieron espectogramas de los diversos audios en sus dos categorías (se utilizaron las librerías de Torchaudio para leer los audios y obtener sus características y en conjunto con Scipy se hizo la remoción de ruido, y se uso Matplotlib para generar las imágenes de los espectogramas) e incluso, bajo un tratamiento de remoción de ruido de fondo con un cutoff frequency (frecuencia a la cual se establece un límite para el filtrado de una señal) vs sin remoción de ruido, para comparar si las muestras tratadas tienen más accuracy en su modelo.

Las RNC, son redes neuronales que trabajan en imágenes, para tratar de predecir cualquiera que sea la categoría a evaluar. Un ejemplo de sus aplicaciones es para identificar objetos. Este modelo se divide en dos partes, una primera red neuronal convolucional que obtiene características importantes de las imágenes, y la segunda red neuronal Dense. La red convolucional con sus neuronas que son matrices cuadradas, mapean la información de la imagen (la imagen ya habrá sido transformada en información numérica -array/matriz- con escala de 0 a 255 cada píxel, siendo esta escala una escala de colores, pueden haber dos escalas, blanco y negro, y RGB, que son rojo, verde y azul, en este caso cada pixel puede ser representado en 3 escalares, uno de cada color), con operaciones matriciales, reduciendo las dimensiones del array, pasandolo a la siguiente capa. En medio de esas capas pueden haber operaciones de “pooling”, el cuales son matrices que resumen las características encontradas en arrays incluso aun más pequeños con operaciones matriciales aun más sencillas, como encontrar el máximo valor en una neurona. Dependiendo de la estructura que plantes, se puede llegar a tener un mayor accuracy en el modelo final. En mi caso, utilice un modelo de TransferLearning, el cual es el VGG16 de la librería Tensorflow/Keras, la cual cuenta con 13 capas convolucionales y 5 de pooling, y se optó por mapear las imágenes en sus 3 canales de RGB, las imágenes originales fueron acotadas a 224x244 pixeles y se usaron kernels (neuronas mapeadoras) de tamaño 3 x 3. También cabe aclarar que se utilizó la librería CV2 para la manipulación de las imágenes.

Después estos arrays fueron mandados a la segunda red neuronal Dense también de Keras, La primera capa consto de 2048 neuronas. Se trabaja con lotes de 32 inputs (batch size). Todas las capas, a excepción de la última, contaban con función de activación “relu” (para forzar que los valores sean positivos, o de lo contrario,

cero). De esta capa a la segunda (de 1024 neuronas) se implementa “Dropout”, el cual es una técnica de regularización que desactiva aleatoriamente un porcentaje de las neuronas durante el entrenamiento (el 10% en este caso), lo que ayuda a evitar el sobreajuste (overfitting). La tercera capa consta de 512 neuronas, la cuarta de 64, la quinta de 10 y la última de 2, debido a que solo son dos categorías, Happy siendo 1, Sad siendo 0, se pudo representar con la función de activación “sigmoid”, para obtener la predicción dicotómica. Cabe destacar que la última capa se le aplicó una regularización L2 a los pesos para penalizar los valores grandes en los pesos de la capa, lo que ayuda a prevenir el sobreajuste y mejora la generalización del modelo. También es notorio que a cada capa de la red se le aplicó “BatchNormalization”, la cual normaliza las activaciones del lote en función de su media y varianza, lo que ayuda a estabilizar y acelerar el entrenamiento y mejorar la generalización del modelo. Se recorrió el dataset 10 veces (epochs).