Informe Trabajo Final: Generación de Sonidos de Pájaros

Introducción

En la última década, las investigaciones en temas de inteligencia artificial han alcanzado un mayor auge. De acuerdo con el motor de búsqueda Scopus, en el año 2022 se publicaron por lo menos 125 mil artículos sobre redes neuronales y 50 mil en inteligencia artificial general. Debe notarse que al día de hoy existen diversas arquitecturas de redes neuronales, cada una diseñada para desempeñarse en labores específicas como clasificación, procesamiento de lenguaje natural, traducción por máquina, análisis no supervisado de datos o generación de imágenes, texto y series de tiempo.

Dentro de las estructuras más conocidas y empleadas están las redes neuronales convolucionales (CNN), que en general, son empleadas para tareas de clasificación. Este tipo de red nació con la creación de Neocognitron, un modelo de red neuronal autoorganizada para un mecanismo de reconocimiento de patrones no afectado por el cambio de posición, elaborado por Kunihiko Fukushima en 1980 [1]. Luego, el modelo de Fukushima fue mejorado por Yann LeCun en 1998 proponiendo LeNet-5, una arquitectura de CNN diseñada para reconocimiento de caracteres escritos a mano, donde introdujo un método de aprendizaje basado *backpropagation* para poder entrenar el sistema correctamente [2]. Posterior al trabajo hecho por LeCun, surgieron diferentes estructuras de redes convolucionales que implementaban unidades de procesamiento gráfico (GPU por sus siglas en inglés) para la optimización de los tiempos de entrenamiento. Entre ellas se encuentran la CNN sobre GPU de K. Chellapilla en 2006 [3], que fue 4 veces más rápido que una implementación equivalente en una unidad de procesamiento central (CPU por sus siglas en inglés), y la deep CNN de Dan Cireșan en 2011, que era 60 veces más rápida [4], superando a sus predecesores en agosto del mismo año. Al año siguiente, esta estructura de red ganaría cuatro concursos de clasificación de imágenes, además de mejorar significativamente el rendimiento para múltiples bases de datos de imágenes. Sin embargo, este mismo año Krizhevsky y colaboradores crearían AlexNet y concursarían en el *ImageNet Large Scale Visual Recognition Challenge* en el mes de Septiembre ganando la competencia [5]. Para el año 2015, AlexNet fue superado por la *very deep* CNN de *Microsoft Research Asia* con más de 100 capas que ganó el concurso ImageNet 2015 [6]. Las mejoras en el campo de las redes neuronales convolucionales, como se aprecia a lo largo de la historia, se debieron a la implementación de GPUs para potenciar el rendimiento y la eficiencia de los tiempos de ejecución y la velocidad de procesamiento en la clasificación de imágenes.

Aunque las CNN se conocen por su aplicación a la clasificación de imágenes, también se han utilizado para la clasificación de audios. Aún así, es importante destacar que, en última instancia, los datos de audio deben convertirse en imágenes, ya que las entradas que utilizan en las redes convolucionales son tensores. Este proceso de conversión implica mapear la forma de onda del audio a un espectrograma, siendo estos últimos las entradas que recibirá la red.

Con base en esta aplicación de las CNN, el objetivo de este trabajo final es desarrollar una clasificación de cantos de pájaros grabados alrededor del norte del Monte Kenia. Esta es una tarea importante para los científicos de *NATURAL STATE* que monitorean las poblaciones de aves con fines de conservación. La base de datos es tomada de *kaggle*, una plataforma en línea de competiciones de ciencias de datos y de machine learning. Esta base consiste de un conjunto de entrenamiento de 264 carpetas, donde cada carpeta corresponde a un ave diferente con sus respectivos audios, y 3 archivos *csv*, que para nuestro interés, solo se usará *train_metadata.csv* que contiene columnas con la etiqueta y la ruta de cada audio. Además, estos archivos están reducidos a 32 kHz y en formato *ogg*.

En las siguientes secciones se presentan los métodos y el procedimiento usado para lograr el objetivo, los resultados obtenidos de la clasificación al aplicar una red convolucional a este conjunto de datos y las conclusiones respecto a los valores alcanzados de precisión y posibles formas de mejorarlos.

Métodos y Resultados

Para comenzar, la información que se tiene sobre el sonido de las aves se presenta en forma de series de tiempo. Como el esquema de extracción de características y clasificación incluye el uso de redes convolucionales, utilizadas generalmente para clasificación de imágenes, se extrajeron los espectrogramas de las series de tiempo. Los espectrogramas, son gráficas que dan cuenta de la intensidad de diferentes frecuencias en un intervalo temporal; en ese sentido, como los sonidos de las aves se caracterizan por la intensidad de las frecuencias y las regularidades del audio, los espectrogramas son ideales como información para la clasificación de aves a través del sonido. Debe notarse que debido a limitaciones computacionales, no se utilizaron todos los archivos de la base de datos, se seleccionaron 10 carpetas, que corresponden a 10 especies de aves diferentes, cada una con alrededor de 90 audios. A partir de los metadatos de los audios, se utilizó el módulo de audio de PyTorch (torchaudio) para: 1) cargar el archivo, 2) ajustar el número de canales de audio, 3) truncar la señal hasta un tiempo determinado en milisegundos y 4) generar el espectrograma de cada señal.

Ya con eso, se construyó un conjunto de datos de sonido con el tipo de datos requerido para el procesamiento en PyTorch. Además, se aseguró de que todas los datos tuvieran el mismo número de canales de audio. Nótese que todos estos pasos de procesamiento son tomados de un artículo del blog *Towards Data Science* [7]. Luego de eso, se construyó un modelo de clasificación de los espectrogramas que cuenta con cinco bloques residuales convolucionales, con cada iteración aumentando la cantidad de características: se partió de 2 canales de audio, pasando por una CNN con un kérnel 3 x 3 que lo llevó a 8, y así sucesivamente hasta llegar a 128 características (Ver Fig. 3). Como su nombre lo indica, estos bloques residuales cuentan con un corto circuito que suma los dos canales originales al resultado de la capa convolucional.

El modelo se entrenó en 60 *epochs*, siendo este número un dato empírico para la convergencia del valor de pérdida. Por último, como métricas para evaluar el desempeño se utilizaron: la entropía cruzada, al tratarse de un problema de clasificación no binario; y la precisión absoluta en el conjunto de datos de validación, pues interesa que el modelo sea capaz de clasificar correctamente la mayor cantidad de sonidos.

Las métricas obtenidas para el conjunto de datos de entrenamiento se muestran en la Fig. 1 y Fig. 2.

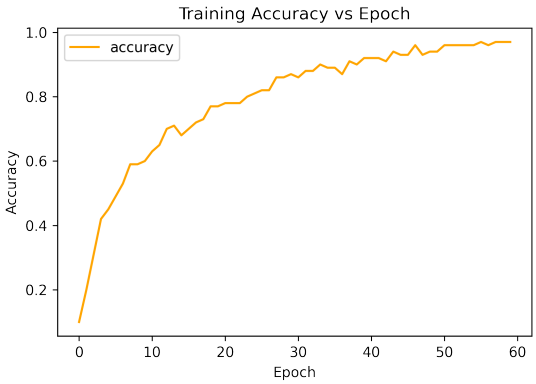

Fig. 1: *Accuracy* de entrenamiento.

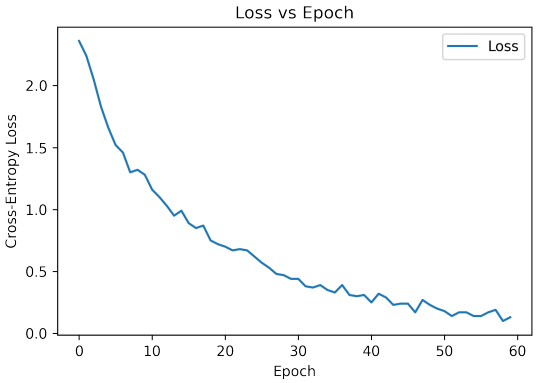

Fig. 2: *Cross-Entropy Loss*

Los valores de precisión y de pérdida para el entrenamiento son de 0.97 y 0.13 respectivamente. La precisión obtenida para los datos de validación fue del 0.73.

Conclusiones

Con el procesamiento de audio, el entrenamiento y los resultados obtenidos se observa que el modelo tiene

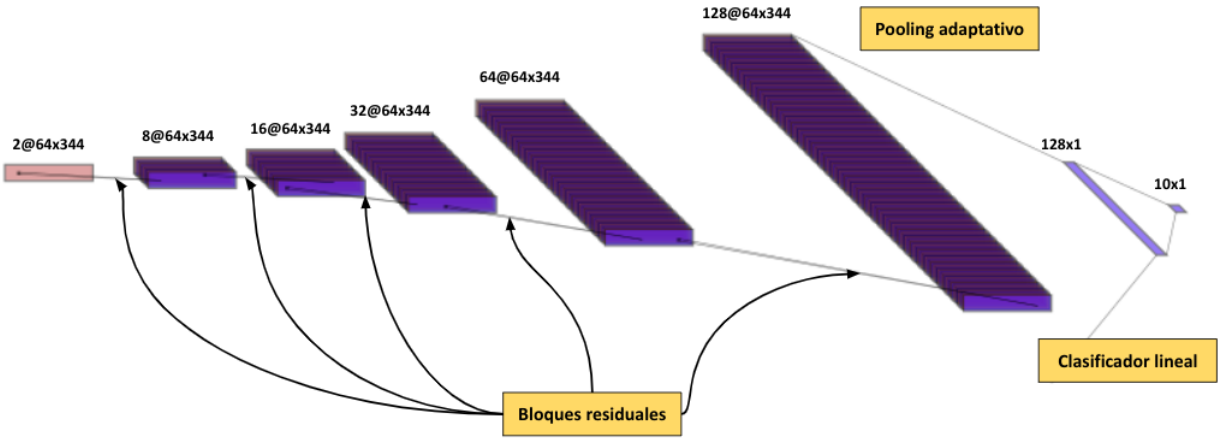

Fig. 3: Esquema de la CNN implementada. Como se ve, se trata de 5 bloques residuales que duplican la cantidad de features en cada iteración. Además, al final de la red se tiene un aplanamiento, un *pooling* global y un clasificador lineal.

la capacidad de clasificar correctamente el 73% de los audios, específicamente los espectrogramas asociados a ellos. A pesar de haber realizado diversas modificaciones en la estructura de la red, como la adición de capas o bloques residuales, no se logró superar el valor de precisión de validación mencionado anteriormente. El siguiente paso lógico es usar transfer learning para intentar mejorar las métricas. Se encontró una arquitectura de red llamada *YamNet*, sin embargo, esta red está implementada con *TensorFlow* mientras que el modelo de este proyecto está desarrollado con *PyTorch*, dificultando la implementación del transfer learning.

Por otro lado, una alternativa para mejorar la precisión de validación es usar más carpetas de audios asociadas a distintas aves. Como se mencionó en la sección de métodos, debido a la limitación de hardware, solo se utilizaron 10 (RAM de Colab) carpetas para entrenar el modelo. Esto significa que se entrenó en promedio con alrededor de 900 audios, lo cual es un número pequeño para considerar modificar la estructura de la red utilizando técnicas como dropout o maxpooling. Probablemente, cargando todos los datos de entrenamiento disponibles y modificando un poco la estructura de la red (agregando capas y métodos mencionados anteriormente), el modelo se entrene mucho mejor y alcance una mayor precisión. Además, podría considerarse tratar el modelo con añadidos a la arquitectura como cabezas de auto-autotención múltiple; de esta forma, más allá de eliminar la cantidad de capas de convolución, se podrían identificar mejor las zonas claves en los espectrogramas.

Referencias

[1] K. Fukushima, "Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position," *Biological cybernetics*, vol. 36, no. 4, pp. 193–202, 1980.

[2] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[3] K. Chellapilla, S. Puri, and P. Simard, "High performance convolutional neural networks for doc-

ument processing," in *Tenth international workshop on frontiers in handwriting recognition*, Suvisoft, 2006.

[4] D. C. Ciresan, U. Meier, J. Masci, L. M. Gambardella, and J. Schmidhuber, "Flexible, high performance convolutional neural networks for image classification," in *Twenty-second international joint conference on artificial intelligence*, Citeseer, 2011.

[5] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, 2017.

[6] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

[7] K. Doshi, "Audio Deep Learning Made Simple: Sound Classification, step-by-step — towardsdatascience.com." `https://rb.gy/61vlc`, 2021. [Accessed 21-May-2023].