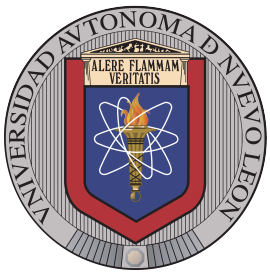

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN

FACULTAD DE CIENCIAS FÍSICO MATEMÁTICAS

MAESTRÍA EN CIENCIA DE DATOS

RECONOCIMIENTO DE COMANDOS DE VOZ

POR:

ACT. DAMIÁN ATILANO MARTÍNEZ ALVARADO

SAN NICOLÁS DE LOS GARZA, N.L. 2024

1. Introducción

En este proyecto, implementamos un modelo de reconocimiento de comandos de voz utilizando un conjunto de datos de comandos de voz *(speech commands)*. El objetivo principal es clasificar correctamente diferentes comandos de voz como *yes, no, up, down, left, right, on, off, stop, y go*. Utilizamos técnicas de procesamiento de señales y deep learning para construir y entrenar el modelo, y evaluamos su desempeño en términos de precisión y pérdida.

2. Descripción de los datos

El conjunto de datos *speech commands* proporcionado por *TensorFlow Datasets* contiene miles de grabaciones de voz, cada una correspondiente a uno de los comandos mencionados anteriormente. Cada grabación es un archivo de audio en formato WAV, con una duración de un segundo. Los datos están etiquetados, lo que significa que cada archivo de audio tiene una etiqueta correspondiente que indica el comando que se está diciendo.

- Formato: Archivos WAV.
- Duración: 1 segundo por archivo.
- Etiquetas: Comandos de voz específicos como ”yes”, ”no”, etc.
- Cantidad de Datos: Miles de grabaciones distribuidas equitativamente entre los comandos.

Durante este proyecto, utilizamos el 10 % del total de los audios disponibles en el conjunto de datos. Esto significa que trabajamos con aproximadamente 3500 grabaciones para el entrenamiento y validación del modelo.

3. Metodología

3.1. Preprocesamiento de Datos

Utilizamos *TensorFlow Datasets* para descargar y cargar el conjunto de datos *speech commands*. Las señales de audio se ajustan a una longitud fija de 16000 muestras; si una señal es más corta, se rellena con ceros, y si es más larga, se trunca. Para extraer características relevantes, aplicamos la transformada *wavelet* discreta (*DWT*) a cada señal de audio, convirtiéndola en un conjunto de coeficientes que representan las características de frecuencia de la señal.

3.2. Construcción del modelo

La arquitectura del modelo incluye dos capas convolucionales 1D con 64 y 128 filtros, respectivamente, cada una seguida de una capa de *max pooling* para reducir la dimensionalidad. También cuenta con tres capas de *dropout* con una tasa del 50 % para evitar el sobreajuste. Finalmente, el modelo tiene una capa densa con 256 neuronas y activación *ReLU*, seguida de una capa de salida densa con activación *softmax* para la clasificación.

3.3. Entrenamiento y Evaluación del Modelo

Entrenamos el modelo con un tamaño de *batch* de 32 y por 15 épocas. Utilizamos un conjunto de datos de validación para evaluar el modelo durante el entrenamiento y ajustar los hiperparámetros si es necesario. El modelo tardó más de 3 horas en completarse debido a la complejidad del modelo y la cantidad de datos procesados. Después del entrenamiento, evaluamos el modelo en un conjunto de datos de prueba separado para determinar su precisión y pérdida en datos no vistos durante el entrenamiento.

4. Resultados

Los resultados obtenidos muestran una precisión mayor en comparación con intentos anteriores. La precisión final en el conjunto de validación es aproximadamente 0.1997. La precisión en el conjunto de entrenamiento es ligeramente mejor, alcanzando aproximadamente 0.2450.

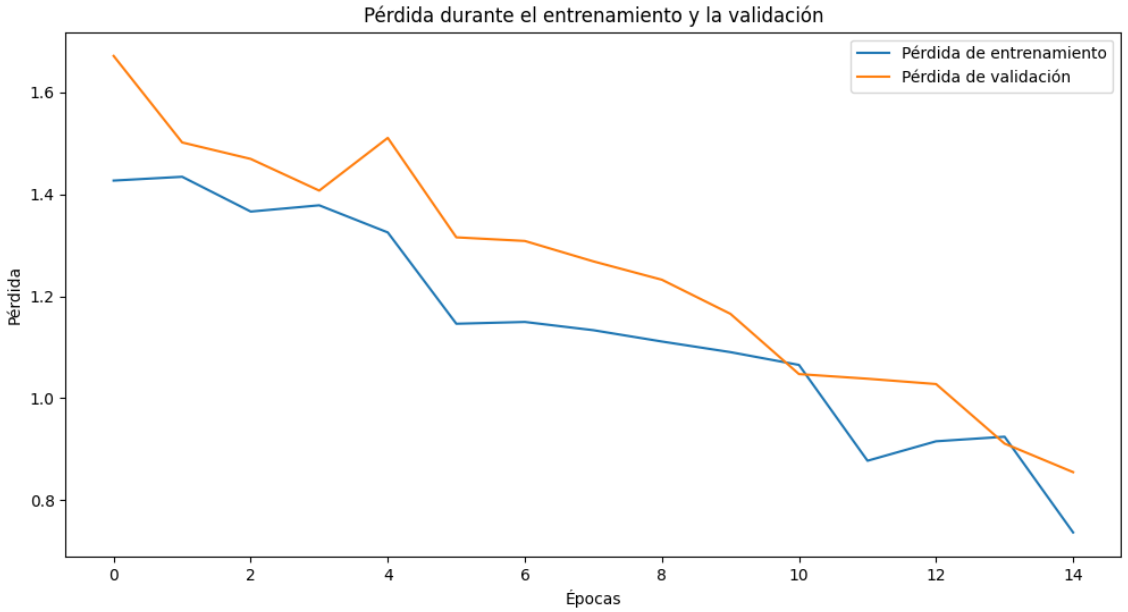

Figura 1: Graficas de Pérdida

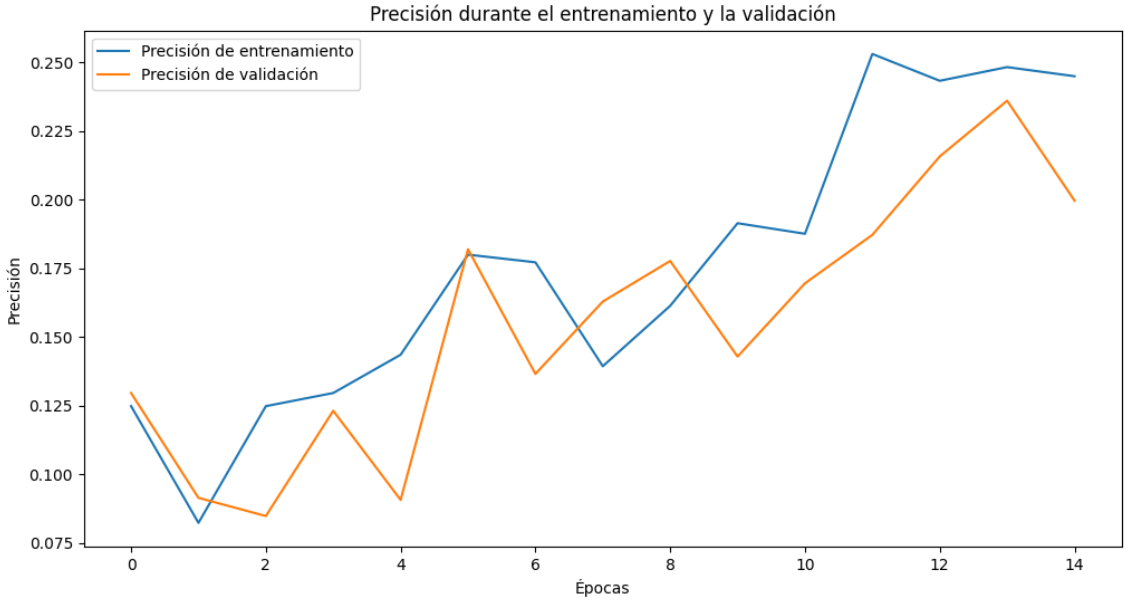

Figura 2: Graficas de Precisión

5. Conclusiones

El bajo rendimiento del modelo podría estar relacionado con varios factores. Al utilizar solo el 10 % del conjunto de datos original, puede que no haya sido suficiente para que el modelo aprenda todas las características distintivas de cada comando de voz. La arquitectura del modelo, aunque compleja y robusta, puede necesitar ajustes adicionales para estos datos. Estos factores, combinados, pueden haber contribuido a la precisión más baja observada durante el entrenamiento y la validación.

Para mejorar el rendimiento del modelo, se recomienda utilizar el conjunto de datos completo en lugar del 10 %, reevaluar y ajustar el preprocesamiento de las señales de audio, experimentar con diferentes arquitecturas de redes neuronales, implementar y aumentar el número de épocas de entrenamiento.

El reconocimiento de comandos de voz requiere un preprocesamiento adecuado y un modelo robusto. Los resultados actuales indican problemas en el aprendizaje del modelo, posiblemente debido a datos insuficientes y una arquitectura no optimizada. Para mejorar, es esencial experimentar con diferentes técnicas de extracción de características, ajustar hiperparámetros, usar el conjunto de datos completo, aplicar técnicas de regularización y aumentar los datos.