Universidad Nacional Autónoma de México
(UNAM)

Instituto de Investigaciones en Matemáticas Aplicadas y en Sistemas
(IIMAS)

Escuchando Emociones

Materia:
Introducción Al Aprendizaje Profundo

Profesores:
M. en C. Berenice Montalvo Lezama
M. en C. Ricardo Montalvo Lezama

Autor:
Ortega Ibarra Jaime Jesús

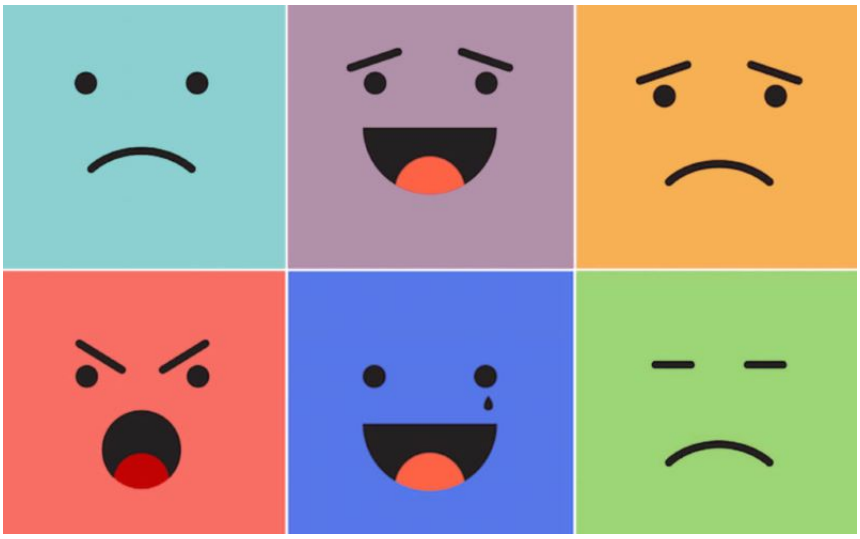
Junio 12, 2021

Índice

1. Introducción

En los últimos años, la cantidad de datos que se generan día a día ha crecido de manera exponencial, pero no solo la cantidad, si no los diversos formatos en que podemos encontrar estos, es decir, al día se capturan millones de fotografías, videos, se realizan millones de llamadas, mensajes, entre otros.¿

Esto a su vez impulsa el desarrollo de nuevas tecnologías y estrategias para tratar los datos, ¿Te imaginas poder medir la satisfacción de un cliente a través de una llamada telefónica?, muchas empresas hoy en día desearían poder hacerlo o poder detectar el estado de ánimo de una persona a través de su voz, sería algo de mucho impacto, por ello, en este proyecto se plantea poder clasificar audios, de acuerdo a la emoción que el hablante presente.



2. Herramientas

Para llevar a cabo este proyecto utilizaremos el lenguaje de programación *Python*, específicamente dentro de Google Colab, pues dicho entorno nos permitirá tener de manera virtual los recursos necesarios para llevar a cabo nuestro proyecto.

Para el tratamiento de los datos utilizaremos diversas librerías, entre ellas:

- Matplotlib y Seaborn: Para visualización de los datos.
- Numpy y Pandas: Para el tratamiento y organización de los datos.
- Librosa: Para el procesamiento de los datos de tipo Audio.
- PyTorch: Para nuestro Modelo.

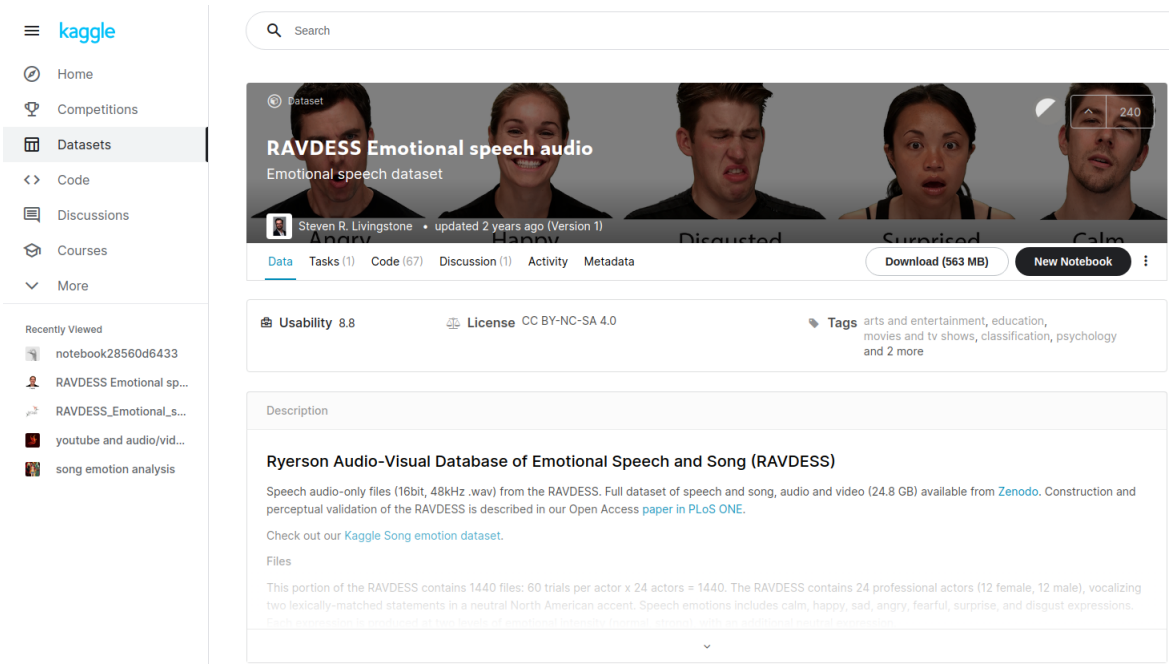
3. Datos

Para llevar a cabo este proyecto, se utilizará la base de datos The Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS) Dicha base de datos contiene tanto los videos como audios de

diversos actores, los cuales fueron capturados al momento de decir diversos discursos utilizando diferentes emociones, en nuestro caso nos enfocaremos únicamente en el apartado de Audio.

Dichos datos los vamos a extraer desde el portal de Kaggle, cuyo url es el siguiente:

- `https://www.kaggle.com/uwrfkaggler/ravdess-emotional-speech-audio`



3.1. Formato y Etiquetado

Dentro de nuestro conjunto de datos encontramos 24 Carpetas diferentes, cada una de estas corresponde a cada uno de los 24 actores, dentro de estas podemos encontrar los 60 audios en formato *.wav*, dando así un total de 1440 audios para todo el conjunto de datos.

Dentro del etiquetado de nuestros datos, lo encontramos dentro del mismo nombre de cada uno de los archivos, tal como lo podemos observar en el siguiente ejemplo:

$$03 - 01 - 03 - 02 - 02 - 01 - 01.wav$$

Donde cada un de los valores corresponden a la siguiente lista, en el mismo orden que se presenta.

- Modalidad
- Canal Vocal
- Emoción
- Intensidad
- Discurso
- Repetición
- Actor

Para cada uno de los atributos, contamos con un diccionario de los diversos valores que pueden llegar a presentarse.

3.2. Diccionario de Etiquetado

- Modalidad
  - 01: AV Completo
  - 02: Solo video
  - 03: Solo audio
- Canal Vocal
  - 01: Habla
  - 02: Canción
- Repetición
  - 01: Primera repetición
  - 02: Segunda repetición
- Intensidad emocional
  - 01: Normal
  - 02: Fuerte
- Actor
  - Impar: Hombre
  - Par: Mujer
- Discurso
  - 01: “Kids are talking by the door”
  - 02: “Dogs are sitting by the door”
  - Emoción
    - 01: Neutral
    - 02: Calma
    - 03: Feliz
    - 04: Triste
    - 05: Enojado
    - 06: Temeroso
    - 07: Disgustado
    - 08: Sorprendido

4. Desarrollo

Es importante mencionar que dado el tipo de problema, es decir, en el cual deseamos detectar una posible emoción a partir de un audio, este se puede denominar como un problema de Clasificación. Pues mediante los datos entrenados con diversas etiquetas, deseamos extraer ciertas características para poder clasificar datos posteriores.

4.1. Exploración de los Datos

4.2. Extracción de los Datos

Para poder explorar nuestro conjunto de datos, he decidido utilizar la API de Kaggle, pues esta me permitirá colocar los datos dentro de mi notebook de manera sencilla y sobre todo cumplir con la replicabilidad del ejercicio, anteriormente he cargado a github el archivo *.json* correspondiente para poder realizar la conexión, dando así el siguiente resultado:

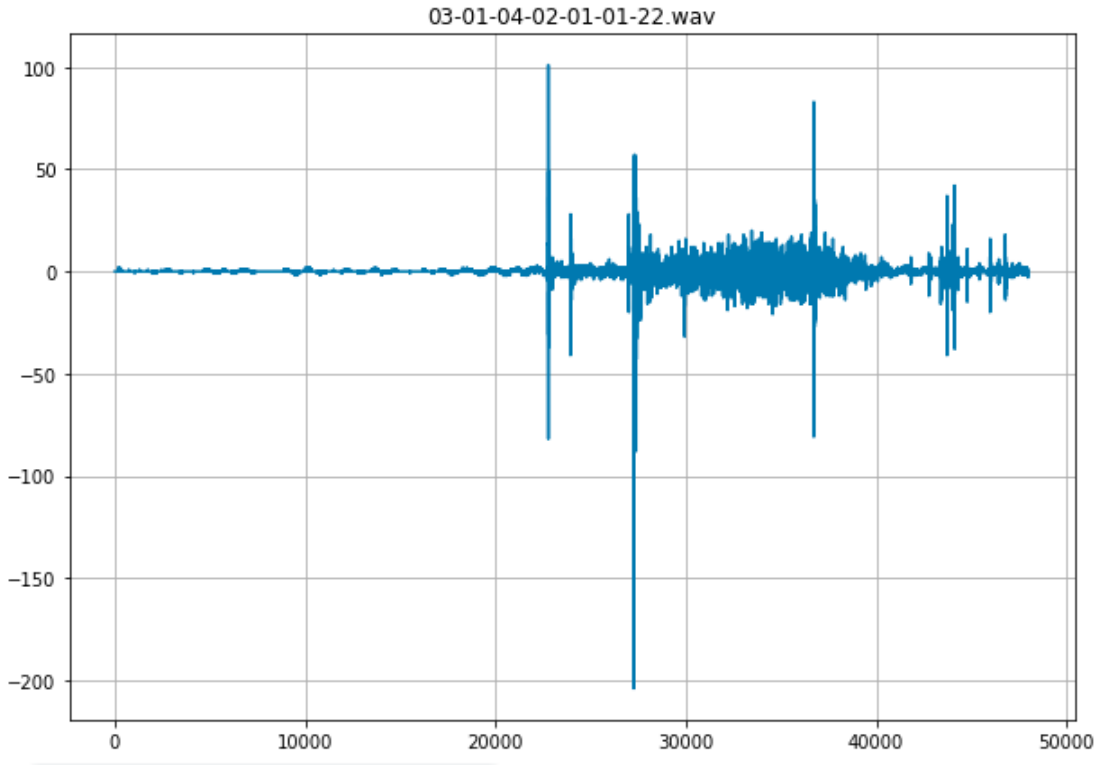
```
[2]  !wget https://raw.githubusercontent.com/joiortega1/Deep_Learning/main/Proyecto/kaggle.json
     ! pip install -q kaggle
     ! mkdir ~/.kaggle
     ! cp kaggle.json ~/.kaggle/
     !chmod 600 ~/.kaggle/kaggle.json
     !kaggle datasets download -d uwrfkaggler/ravdess-emotional-speech-audio
     !unzip 'ravdess-emotional-speech-audio.zip'
```

```
  inflating: audio_speech_actors_01-24/Actor_24/03-01-01-01-01-02-24.wav
  inflating: audio_speech_actors_01-24/Actor_24/03-01-01-01-02-01-24.wav
```

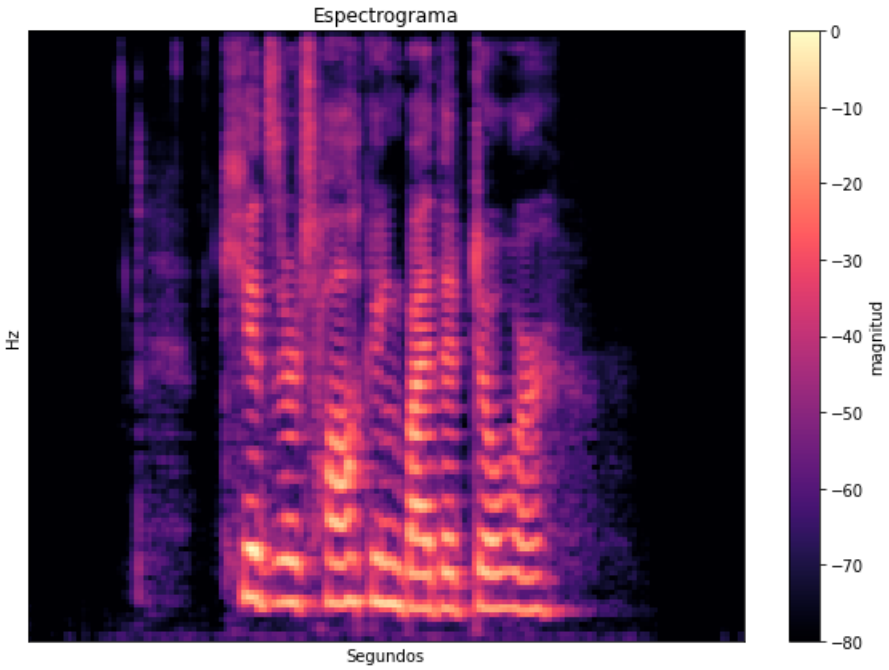
Como podemos observar en la captura anterior, nuestros datos ya se encuentran alojados en nuestro entorno.

4.3. Frecuencia y espectrograma

Ya alojados nuestros datos, se ha utilziado tanto librosa como waves, para poder extraer la información de nuestros audios y así poder visualizar algunas muestras, tal como vemos a continuación:



En la imagen anterior, podemos observar la frecuencia presentada dentro del audio $03-01-04-02-01-01-22.wav$ cuya duración es de 3,73 segundos, ahora observaremos el espectrograma mediante coeficientes de mel, para dicho audio:



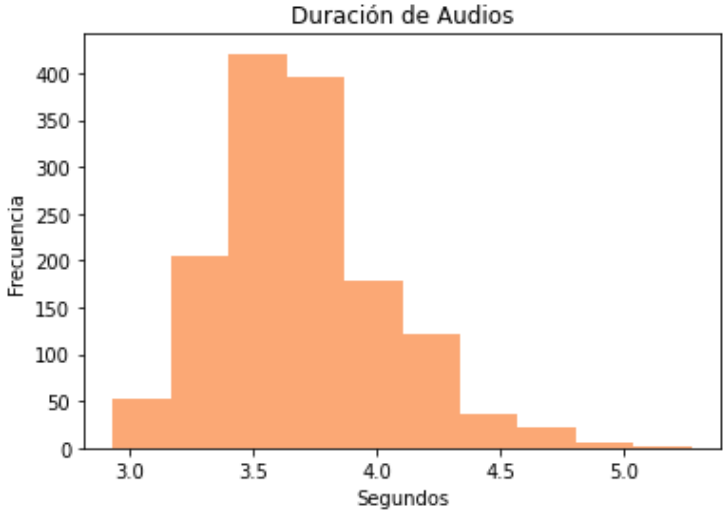
4.4. Clases y Duración

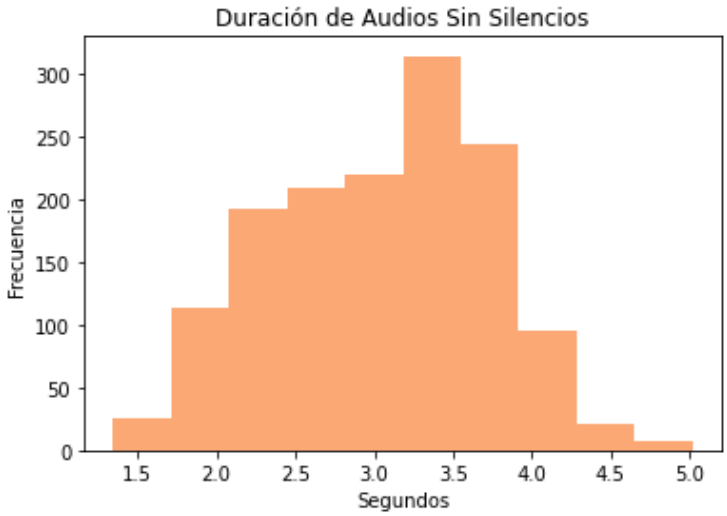
Para llevar mas a fondo nuestra exploración, he decidió crear un dataframe a partir de los datos correspondientes a cada uno de los audios, de esta manera poder observar como se encuentran distribuidos nuestros datos.

	Archivo	Modalidad	Canal	Emocion	Intensidad	Discurso	Repeticion	Actor	
0	03-01-07-02-02-01-01.wav	3	1	7	2	2	1	1	audio_speech_actor
1	03-01-06-02-01-02-01.wav	3	1	6	2	1	2	1	audio_speech_actor
2	03-01-06-01-01-01-01.wav	3	1	6	1	1	1	1	audio_speech_actor
3	03-01-06-02-01-01-01.wav	3	1	6	2	1	1	1	audio_speech_actor
4	03-01-07-01-01-01-01.wav	3	1	7	1	1	1	1	audio_speech_actor
5	03-01-04-02-02-02-01.wav	3	1	4	2	2	2	1	audio_speech_actor
6	03-01-04-01-01-02-01.wav	3	1	4	1	1	2	1	audio_speech_actor
7	03-01-03-01-02-02-01.wav	3	1	3	1	2	2	1	audio_speech_actor
8	03-01-05-02-01-02-01.wav	3	1	5	2	1	2	1	audio_speech_actor
9	03-01-05-02-01-01-01.wav	3	1	5	2	1	1	1	audio_speech_actor

Para poder observar los intervalos de duración de nuestros audios, se ha creado un histograma, pues nos permitirá observar el rango en que estos se encuentran, cabe mencionar que se han implementado dos

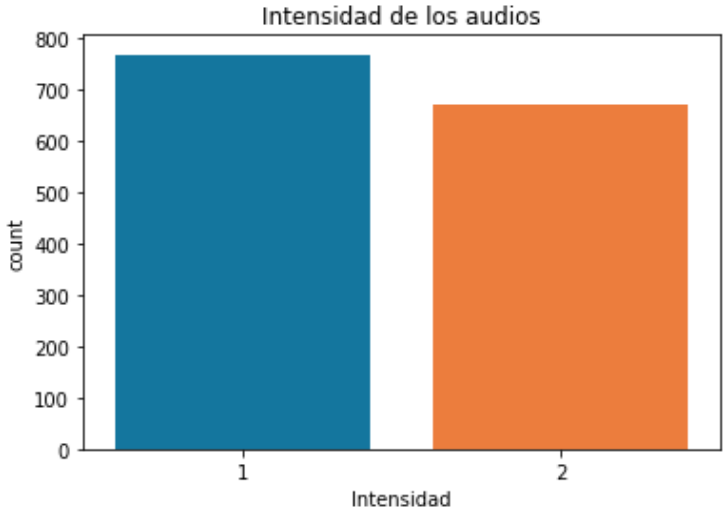
diferentes, pues un histograma ha sido con los audios en crudo y otro ha sido con los audios pero una vez procesados, pues se han eliminado los silencios, ya que encontrábamos muchos silencios ya sea al inicio o al final de estos, de esta manera podemos evaluar cual será la mejor manera de construir nuestro dataset.



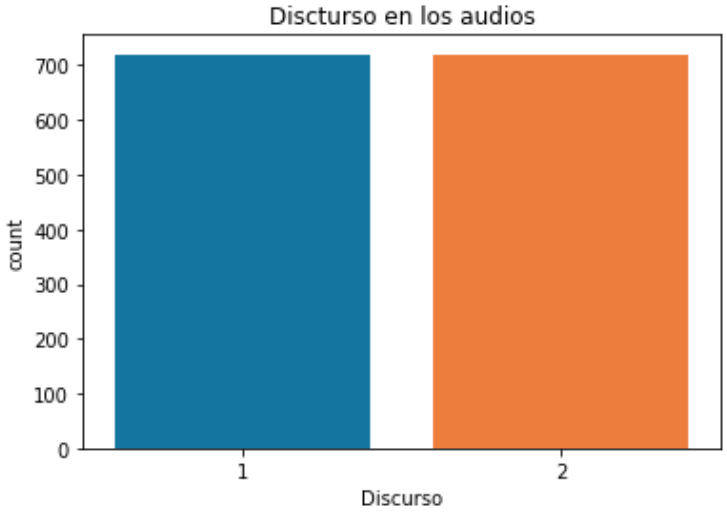


Tal como podemos observar, al eliminar los silencios, el rango de duración de nuestros audios se ve sumamente afectado, pues encontramos audios con 1,5 segundos de duración.

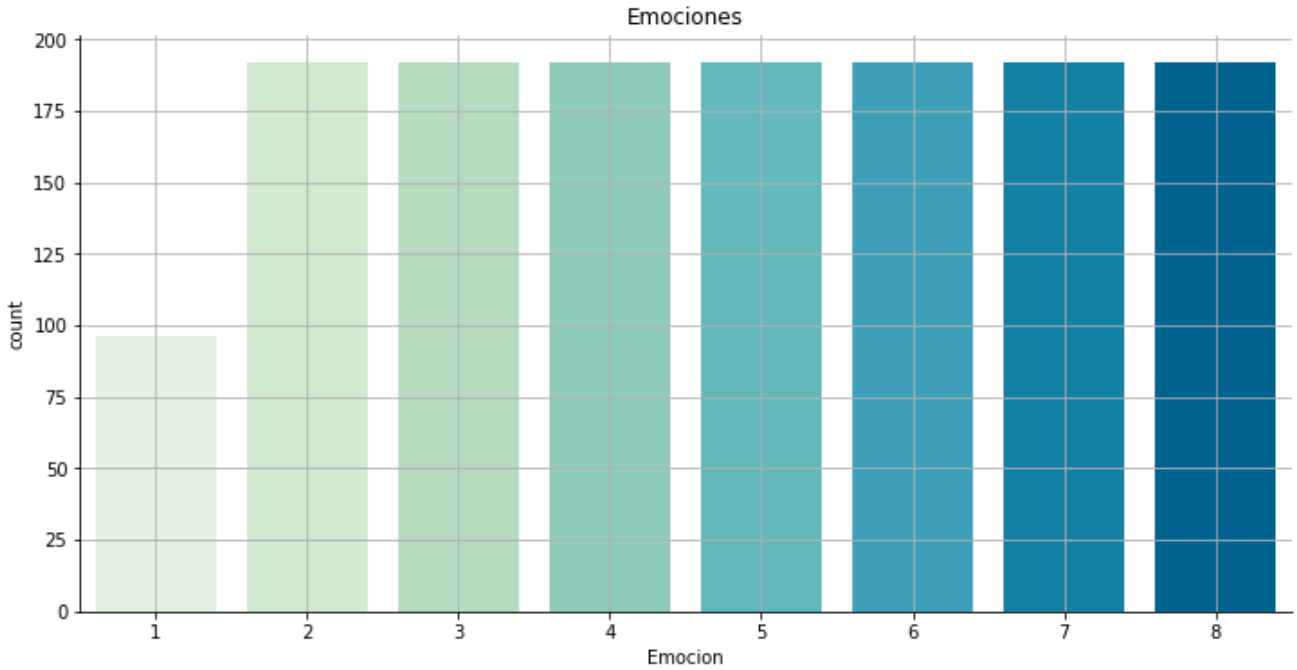
Siguiendo con nuestro análisis exploratorio, es importante saber como se encuentra el balance en nuestros datos, es decir la cantidad de clases existentes y los registros en cada una de ellas, además de las diferentes características proporcionadas en el etiquetado.



Como podemos observar en la gráfica anterior, dentro del atributo de intensidad encontramos mayor tendencia por el número 1, esto nos indica que tenemos mayor cantidad de audios cuya intensidad es Normal, con respecto a la cantidad de auidos cuya intensidad es Fuerte.



En la gráfica anterior, encontramos el balance del discurso, es decir la frase que se dice en cada uno de los audios, como podemos observar existe la misma cantidad de audios para cada frase, es decir 720 audios para "Kids are talking by the doorz 720 para "Dogs are sitting by the door".



Como observamos en esta última gráfica, dentro de nuestras diversas clases (emociones), encontramos un buen balance, a excepción de la clase número 1, correspondiente a la emoción "Neutral", pues presenta la mitad de la cantidad de audios que el resto de las emociones.

4.5. Creación de DataLoader

Para crear nuestro Dataloader, se han extraído los audios mediante la librería librosa, la cual nos permitirá almacenar el valor de cada uno de nuestros arreglos, una vez almacenados, de igual manera se han extraído las etiquetas correspondientes, de acuerdo al nombre de cada uno de los archivos. Ya que debemos entrenar, validar y posteriormente probar nuestro modelo, con ayuda de *ScikitLearn*, se ha utlizado la función *Train_test_split*, de esta manera podemos dividir nuestros datos tanto en prueba como en entrenamiento.

Es importante mencionar que en este procedimiento se utilizaron diversas estrategias para el procesamiento de nuestros datos, entre ellas:

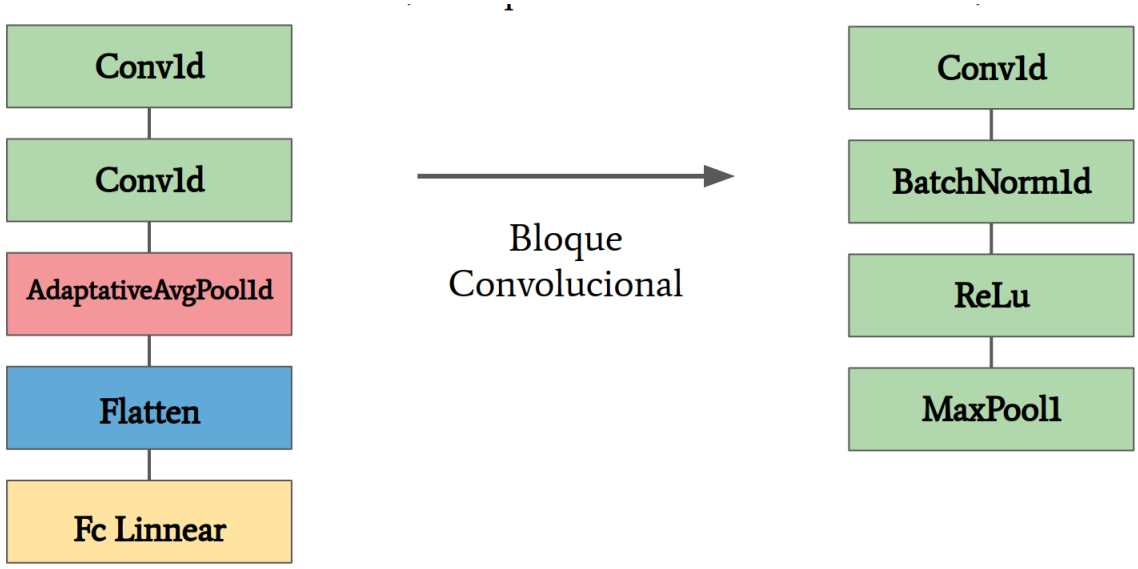
- Eliminar Silencios en Nuestros audios
- Extraer Coeficientes Cepstrales
- Procesar únicamente los audios en crudo

5. Modelo

Para nuestro modelo me he enfocado en las redes neuronales convolucionales ya que desearemos realizarlo con formas de ondas sin procesar, es decir utilizaremos nuestros audios en crudo, ya que como vimos en nuestro histograma, el procesar nuestros audios para eliminar silencios, nos generaría un desequilibrio en la duración de estos, lo cual podría afectar a nuestros tensores.

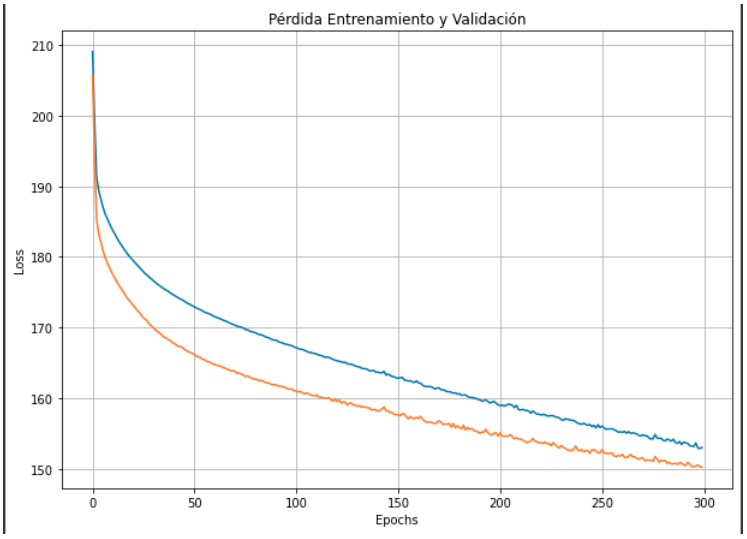
Para este caso, específicamente utilizaremos la Arquitectura M5 vista anteriormente en clase, ya que de acuerdo al formato en que se encontraban nuestros audios y el hecho de poder entrenar mediante audio en crudo.

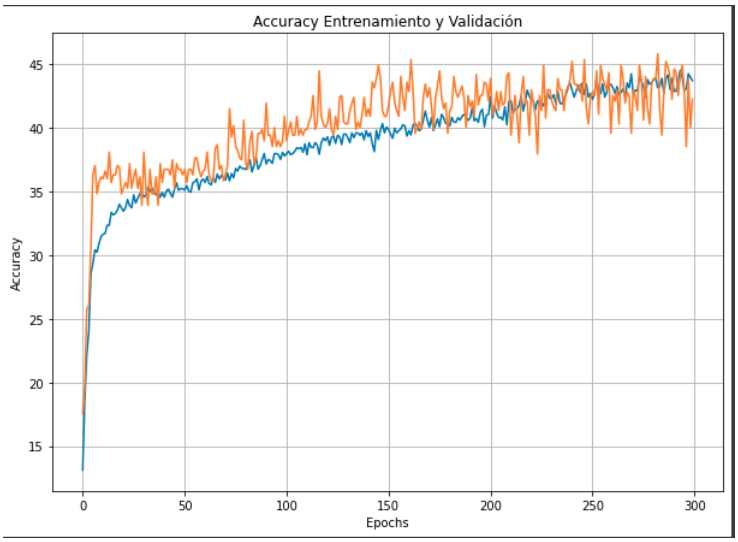
Específicamente encontramos la siguiente arquitectura:



6. Resultados

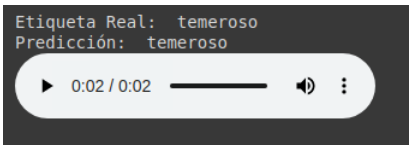
Una vez establecida nuestra arquitectura, he entrenado por 300 épocas, pues noté que era una cantidad que daba un mejor rendimiento con respecto a menores o mayores cantidades, dando así los siguientes resultados:

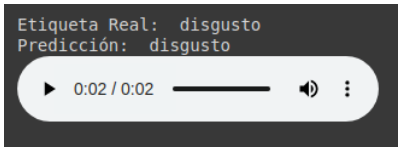


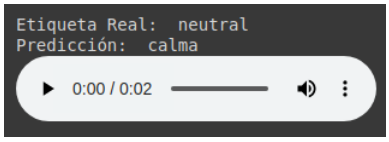


6.1. Predicción

En las siguientes capturas, podemos observar tanto la etiqueta real como la etiqueta de predicción mediante el conjunto de prueba, cabe mencionar que la emoción que mas ha costado trabajo para el modelo es "Neutral", se quiere suponer que es debido a la poca cantidad de datos para dicho atributo.







7. Conclusiones

Se encontraron diversas dificultades al momento de tratar los audios, pues no todas las técnicas para procesar son favorables, tal como lo vimos al quitar los silencios, pues nuestro modelo daba peores resultados, además la falta de datos puede influir de cierta manera, ya que un audio de mayor duración, podría proporcionar mayor cantidad de características para un entrenamiento más preciso, cabe mencionar que la etiqueta en la que mayores fallas tenía, era en la etiqueta neutral, lo cual lo podemos relacionar con la falta de datos en dicho atributo, pues si recordamos la gráfica de barras que se había mostrado en un inicio, la etiqueta "Neutralçontaba con la mitad de audios con respecto al resto de las etiquetas.

7.1. Limitaciones y Dificultades encontradas

Entre las limitaciones me he dado cuenta que al tratar con audios debemos ser sumamente específicos, pues si tenemos algún desajuste en nuestra duración o en valores como sample rate, podemos llegar a tener varios conflictos al procesar nuestros datos, en mi caso al probar utilizar espectrogramas, no lograba entrenar adecuadamente, de igual manera al eliminar los silencios, desequilibraba la duración de mis audios y de esta manera no podía construir mis tensores de la mejor manera.

7.2. Formas de mejorar o expandir el Proyecto

Una posible idea de expandir el proyecto, es impulsar la generación de nuevas bases de datos como RAVDESS, pues realmente existe una cierta escasez de datos realmente bien etiquetados y en este caso, considero que la base de datos es un tanto pequeña y por lo mismo los resultados pueden llegar a ser de cierta manera poco precisos, pues la duración de estos es muy pequeña y el entrenamiento no llega a ser tan extenso como se desea.

8. Referencias

- `https://es.wikipedia.org/wiki/MFCC`
- https://www.kaggle.com/uwrfkaggler/ravdess-emotional-speech-audio
- `https://platzi.com/tutoriales/1794-pandas/6926-usando-la-api-de-kaggle-con-google-colab-para-carga`
- `http://man.hubwiz.com/docset/LibROSA.docset/Contents/Resources/Documents/generated/librosa.effects.trim.html` https://librosa.org/doc/main/generated/librosa.feature.mfcc.html