TENSORFLOW SPEECH RECOGNITION CHALLENGE

Sound is represented in the form of an **audio** signal having parameters such as frequency, bandwidth, decibel, etc.

A typical audio signal can be expressed as a function of Amplitude and Time.

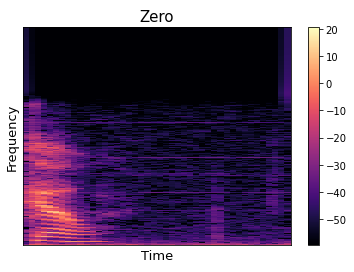
**Sampling an Audio Signal**

Sampling refers to the process of converting a continuous-time signal to a discrete-time signal.

Sampling frequency/sample rate is the average number of samples per second.

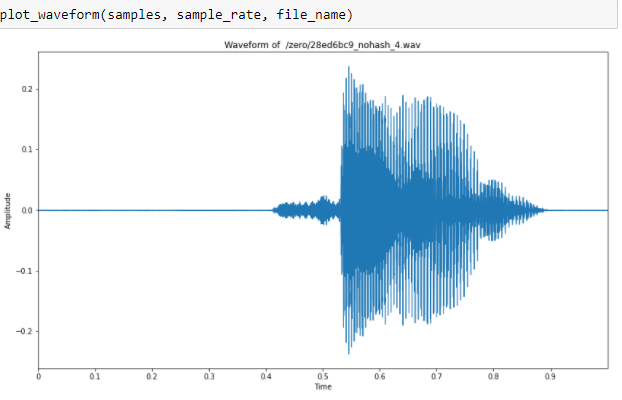
We will also represent our audio signals using spectrograms. Spectrograms are a way of visually representing the strength of the signal over time at different frequencies available in the waveform of a particular audio signal.  
  
**Reading a Spectrogram**  
Spectrograms are two-dimensional graphs in which the third dimension is represented by a colour bar. Time is represented along the X axis and the vertical axis represents the pitch of the sound wave. The amplitude is represented in the third dimension (Dark shade corresponds to low amplitude while a lighter shade is for lower amplitude signal.)

In our project, the following spectrogram graph was obtained when processing an audio signal for the ‘Zero’ sound.

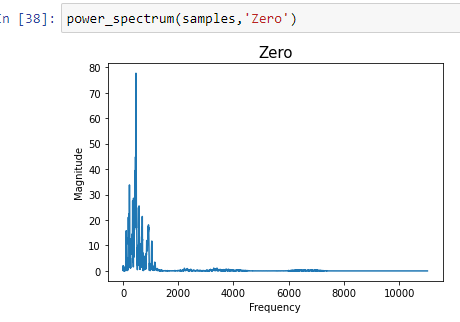


For audio analysis in python, we used librosa package.

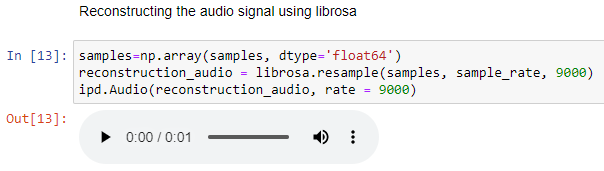
* Librosa provides the building blocks for retrieval of sound waves
* Librosa version used: 0.8.0
* Librosa functions are used for the following purposes.
  + Sampling an audio wave



* + Obtaining power spectrums of the audio waveforms.



* + Reconstructing the audio signals using librosa.resample to obtain the audio signal back from the samples and sample rate.



* + Finally, librosa is used to obtain the signals in the form of numpy arrays to process them in machine learning models.

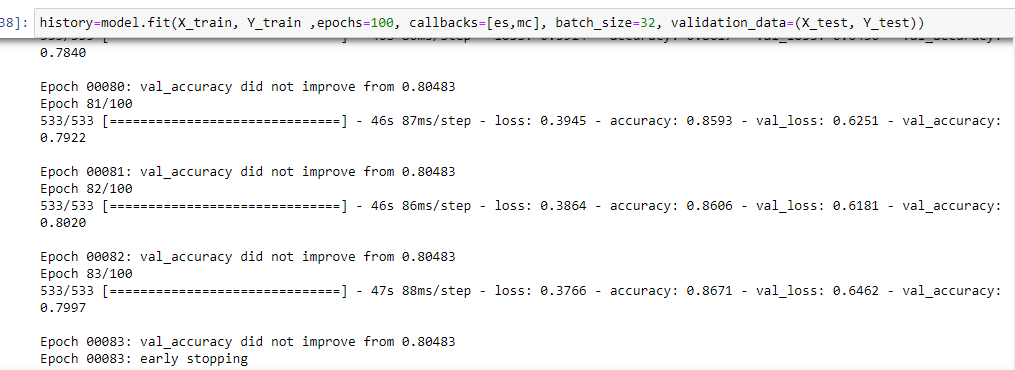
**MACHINE LEARNING MODELS ATTEMPTED**

**LSTM Model**

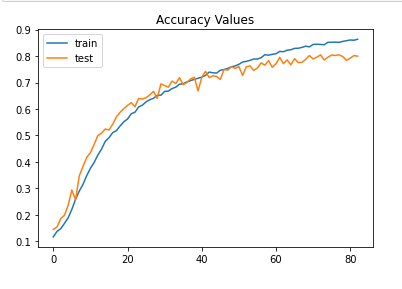
1. The first model consisted of long-term short-term memory (LSTM) layers and Dense layers in an attempt to process the data successfully.
2. Represented by ‘model\_LSTM’ in the jupyter notebook.
3. Model optimized by Adam optimizer having a learning rate of 0.01 while its loss was measured by categorical\_crossentropy.
4. Result: The runtime process of the model was very high. Hence, I was unable to train the model.
5. The model architecture is available at the end of the jupyter notebook.

**Convolutional Model**

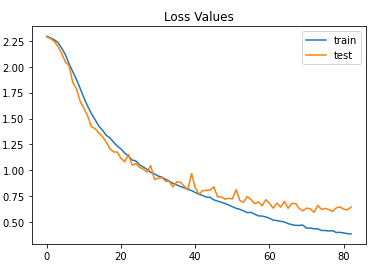
1. This model consists of 1D convolutional layers, MaxPooling Layers, Dense and Dropout layers.
2. Represented by ‘model’ in the jupyter notebook.
3. Model optimized by Stochastic Gradient Descent (sgd) optimizer having a default learning rate while its loss was measured by categorical\_crossentropy.
4. Result: The model was processed successfully having a validation accuracy exceeding 80%.



1. The model architecture is available in the jupyter notebook.
2. The model weights and parameters can be downloaded from the github repository for replication of the model.
3. The accuracy metric while training the said is model is as follows:



1. The loss metric while training the said model is as follows:



**Bibliography/ Source Citations**

* <https://en.wikipedia.org/wiki/Sampling_(signal_processing)>
* <https://pnsn.org/spectrograms/what-is-a-spectrogram#:~:text=A%20spectrogram%20is%20a%20visual,energy%20levels%20vary%20over%20time>
* <https://librosa.org/doc/latest/index.html>
* <https://www.kaggle.com/c/tensorflow-speech-recognition-challenge>
* <https://www.tensorflow.org/tutorials/audio/simple_audio>