Audio Deep Learning

Van Tuan Nguyen

5/2022

Outline

1.The Basics

- Sound is a pressure wave which is created by a vibrating object.These vibrations create sets of particles in the suronding medium (typically air) in vibration motion, thus transporting energy through medium. Since the particles are moving in the parallel directions to the wave movement, the sound wave is refered to as a longitudinal wave.The results of longitudinal waves is the creation of compressions(C) and rarefractions(R) within the air.

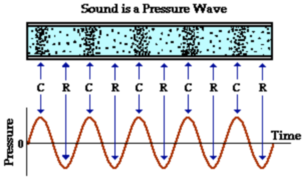


Figure 1: Sound Basics

1. The Basics

The sound wave is fundamentally analogue in the nature. Storing sound analoguely is problematic. In order for computers to manipulate and work with sound, it has to be stored in a digital format. Here are some steps to follow:

1. The analogue sounds must be captured with an input device.
2. The sound is then converted to an electrical analogue signal.
3. Sampling the electrical signals.
4. Storing the digital data.

1. The Basics

- Sampling is the process turning an electrical analog stream of data, into a digital stream of data. The process involves recording the amplitude of an electrical signal at set intervals, rounding the values and converting them to the binary. Sampling rate is the times of sampling in a second.

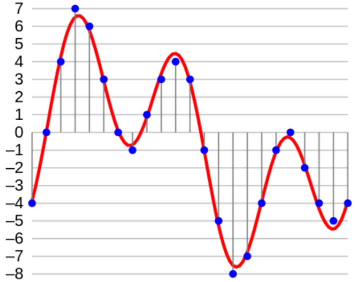


Figure 2: Sampling rate method

1. The Basics

- Bit-rate is number of bits encoded per second or number of bits transmitted or received per second. The higher bit-rate with more sampling rate, the higher quality the resulting audio.
- Channel is the position of each audio source within the audio signal. Each channel contains a sample indicating the audio amplitude of the audio being produced by that source at a given moment of time.
  For instance, in stereo sound , there are two audio sources: one speaker on the left and one speaker on the right. Each of these is represented by one channel.

1. The Basics

- Sound is series of very long signals, but the amount of information in it is not that much. Sound is composed by different waveforms with different frequencies, it is better to modulate a short sound into waveforms with specifics amplitude and frequencies. Fourier Transform(FT) transforms sound in time domain into frequency domain.

1. The Basics

A sound spectrum displays the different frequencies present in a sound.

Spectral representations : the distributions of energy over s set of frequencies

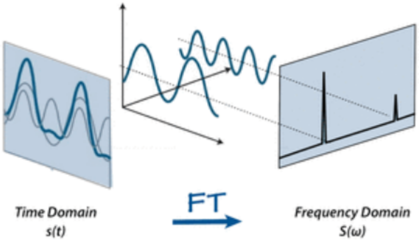


Figure 3: Sound in Time Domain and Frequency Domain

1. The Basics

- A speech spectrogram shows the Fourier Transform of a signal as it varies with time.

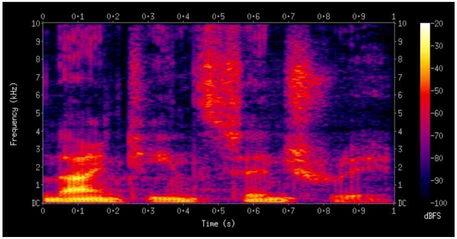


Figure 4: Speech Spectrogram

x axis is in terms of time domain,y axis is in terms of frequency domain.Magnitude is displayed through the colors of the spectrogram. In this spectrogram, it is clearly shown that low frequencies have high magnitude and Low frequencies have low magnitude.

1. The Basics

- Mel Scale is a logarithmic transformation of a signal's frequency. The core idea of this transformation is that sounds of equal distance in Mel - scale are perceived to be equal distance to humans
  For example, people are easy to tell the differences between 100 Hz and 200 Hz sound. However, It's much harder for us to tell the differences between 1000 Hz and 1100 Hz sound. It's actually much harder for people to diffrentiate higher frequencies sound, and easier for lower frequencies. So, Mel Scale mimics our own perceptions of sound.
  Hert(Hz) Scale to Mel Scale formula:

$$m = 2595 * log(1 + \frac{f}{700}) \tag{1}$$

1. The Basics

- Mel-spectrograms are spectrograms that visualize the spectrogram in Mel Scale as opposed the frequency scale aforementioned

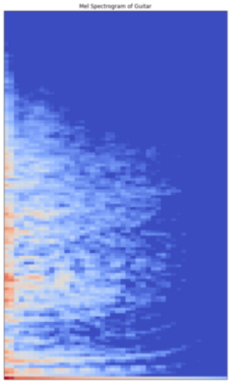


Figure 5: Speech Mel Spectrogram

2. Audio Processing Libraries

1. Librosa
2. Scipy
3. Soundfile
4. Pydub

Librosa

Librosa is a Python package for audio and music signal processing. At high level, librosa provides implementation of a variety of common functions used throughout the file of music information retrieval.

Most commonly used functionalities:

1. Core functionality
2. Spectral Features
3. Display
4. Onsets,Tempo and beats
5. Structural Analysis
6. Decomposition
7. Effects
8. Output

Librosa

- Core functionality:

1. Audio and Time-series operation
   librosa.core.audioread : reading audo from disk
   librosa.core.resample : resampling a signal at a desired rate
   librosa.core.to_mono: stereo to mono conversion
2. Spectrogram calculation
   librosa.core.stft : Short time fourier transformation
   librosa.core.istfr : inverse short time fourier transformation
   librosa.core.logamplitude : log amplitude scaling
3. Time and Frequency conversion
4. Pitch operation

Librosa

- Spectal Feature
  librosa.feature.melspectrogram
  librosa.feature.mfcc
- Display
  librosa.display.waveplot
  librosa.display.spec_how
- Onsets,Tempo and beats
  librosa.onset.onset_strength
  librosa.onset.onset_detect
- Output
  librosa.ouput.times_csv
  librosa.output.annotation
  librosa.output.write_wav

Scipy

Scipy is a open-source scientific computation library that uses Numpy underneath and it provides utility functions for optimization, statistics and signal processing.

1. Scipy Constants : provide many built-in scientific constants
2. Scipy Optimizers : find the minimum value of a function or the root of the equation
3. Scipy Sparse Data: Deals with the sparse data
4. Scipy graphs: working with graphs
5. Scipy Spatial Data
6. Scipy Matlab Arrays
7. Scipy Interpolation
8. Scipy Significance Tests

Soundfile

Soundfile is an audio library based on libsndfile (a free cross-platform open-source library for reading and writing many different sampled sound files format that runs on many platforms including Windows,OS X and Unix) ,CFFI (foreign function interface for Python calling C function) and numpy. Soundfile represent audio data as Numpy arrays

1. Read and Write functions. Soundfile can open all file formats that libsndfile supports: WAV, FLAC, OGG,MAT
   - soundfile.write(file,frames=-1,start=0,stop=None,dtyppe='float64',...) return a two-dimensional(frames x channels).
     Numpy array is returned

Soundfile

- soundfile.read(file,data,samplerate,subtype=None). Data is usually two-dimensional.

2. Block Processing. Sound files can also be read in short, optionally overlapping blocks with soundfile.blocks()
3. Sound files can also be opened as soundfile.SoundFile objects. Every Soundfile has a specific sample rate, data format and a set of number of channels.
    - **class** sound-file.Soundfile(file,mode='r',samplerate=None,channels=None,subtype open a soundfile

Pydub

Pydub is a library work with audio. It can open wav, mp3, ogg, flv file or anything else ffmpeg supports. Pydub can:

- slice audio
- make the begining louder and the end quiter
- concatenate the audio
- Audio Segment
- Crossfade
- Repeat
- Fade
- Save the projects and save the project with tags

3. Audio Data Preparation Technique

1. Fast Fourier Transform (FFT)
2. Short-time Fourier Transform (STFT)
3. Mel Spectrogram
4. MFCC

Fast Fourier Transform

Fourier transform is a mathematical operation that changes the domain (x-axis) of a signal from time to frequency.

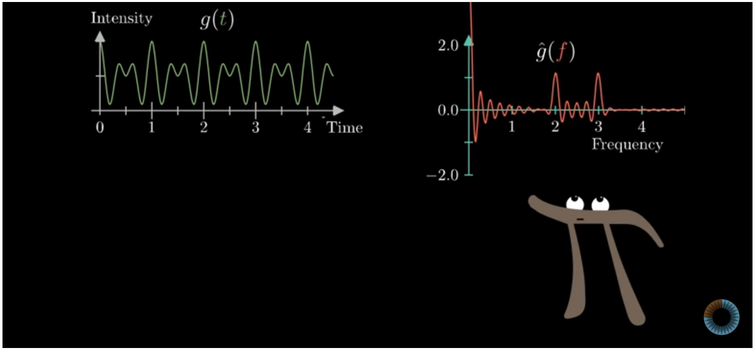


Figure 6: Fourier Transform

Fast Fourier Transform

Discrete Fourier Transform(DCT) can be written as follow:

$$x[k] = \sum_{n=0}^{N-1} x[n] e^{\frac{-j2\pi kn}{N}} \tag{2}$$

- x[n] is a discrete signal
- N size of its domain

Computer needs $0(N^2)$ operations to calculate DCT

Fast Fourier Transform(FFT) is an algorithm that determines DCT of an input significantly faster than computing it directly. FFT reduces the number of operations to $0(NlogN)$

Fast Fourier Transform

Fast Fourier Transform Algorithm:

$$\begin{cases} n = 2r \ \ if \ \ even \\ n = 2r+1 \ \ if \ \ odd \end{cases} \quad wherer = 1,2,3,...,\frac{N}{2}-1 \tag{3}$$

$$x[k] = \sum_{n=0}^{N-1} x[n] e^{\frac{-j2\pi kn}{N}} \tag{4}$$

$$x[k] = \sum_{r=0}^{\frac{N}{2}-1} x[2r] e^{\frac{-j2\pi k(2r)}{N}} + x[k] = \sum_{r=0}^{\frac{N}{2}-1} x[2r+1] e^{\frac{-j2\pi k(2r+1)}{N}} \tag{5}$$

Fast Fourier Transform

$$\text{x[k]=}\sum_{r=0}^{\frac{N}{2}-1} x[2r]e^{\frac{-j2\pi k(2r)}{N}} + x[k] = e^{\frac{-j2\pi k(2r)}{N}} \sum_{r=0}^{\frac{N}{2}-1} x[2r+1]e^{\frac{-j2\pi k(2r)}{N}} \quad (6)$$

$$x[k] = \sum_{r=0}^{\frac{N}{2}-1} x[2r]e^{\frac{-j2\pi k(r)}{N/2}} + x[k] = e^{\frac{-j2\pi k(2r)}{N}} \sum_{r=0}^{\frac{N}{2}-1} x[2r+1]e^{\frac{-j2\pi k(r)}{N/2}} \quad (7)$$

$$x[k] = x_{even}[k] + e^{\frac{-j2\pi k}{N}} x_{odd}[k] \quad (8)$$

Short - Time Fourier Transform(STFT)

Fourier Transform changes the audio data from time domain to frequency domain. However, This transformation loses information about time and STFT is a solution for this problem.

STFT computes several Fourier Transforms at different intervals. Given spectrogram will give us information about : time $+$ frequency $+$ magnitude

Short - Time Fourier Transform(STFT)

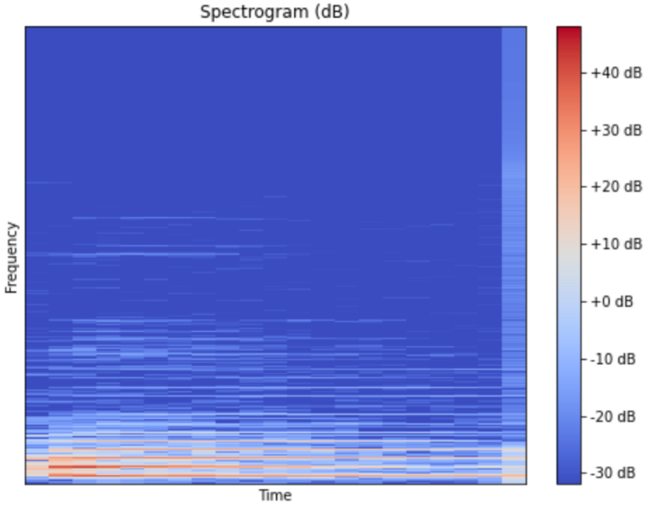
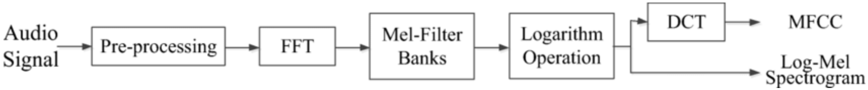


Figure 7: Short Time Fourier Transform

Mel Spectrogram and MFCC

Mel spectrogram is a spectrogram where the frequencies are converted to the mel scale. After computing Mel Spectrogram, we apply DCT to get MFCC.



To get MFCC or Mel Spectrogram. We have to take some following steps:

Mel Spectrogram and MFCC

- Pre-emphasis. Pre-emphasis refers to filtering that emphasizes the higher frequencies. Its purpose is to have a balanced spectrum of voice sounds because the slope of high frequencies voice sounds is smaller than the low frequencies ones.
  Pre-emphasis formula:

  $$y(t) = x(t) - \alpha x(t-1) \tag{9}$$

  Where:

  - x is the input signal
  - $\alpha = 0.95 - 0.97$

Mel Spectrogram and MFCC

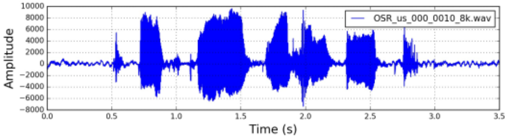


Figure 8: Raw voice sounds

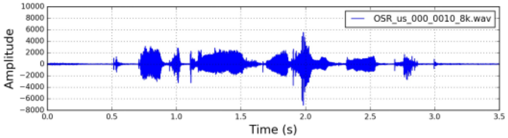


Figure 9: Voice sounds after Pre-emphasis

MFCC and Mel Spectrogram

- Framing and windowing
  The speech signal is slowly time-varying and almost stationary signal. For stable acoustic characteristics , speech needs to be examined over a sufficiently short period of time.Therefore, speech analysis must always be carried out on short segments across which the speech signal is assumed to be stationary. Short-term spectral measurements are typically carried out over 20ms windows and advanced every 10 ms.

Mel Spectrogram and MFCC

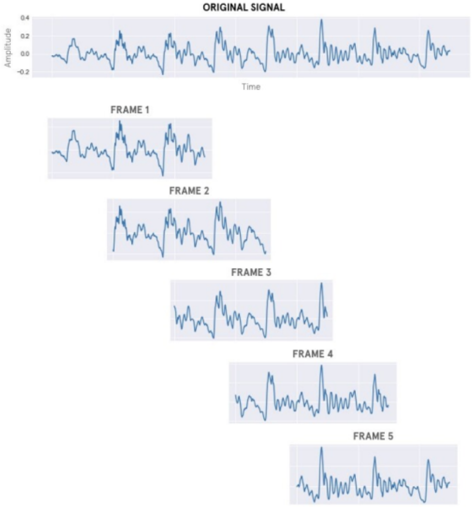


Figure 10: Framing and Windowing mechanism

Mel Spectrogram and MFCC

- FFT

  Because the speech signal is windowed into short-term parts.We apply STFT according to the below formula:

$$X[m,w] = \sum_{k=0}^{N-1} w[k].x[m \times H + k]e^{\frac{-j2\pi kw}{N}} \tag{10}$$

N is the number of points used to compute FFT

Mel Spectrogram and MFCC

- Mel-Filter Banks
  Mel Spectrum is computed by passing the Fourier Transformed signal through a set of bandpass filters known as mel-filter bank. One of the most popular bandpass filters is triangular filters.
  Hertz scale(f) to Mel Scale(m) formula and reverse formula:

$$m = 2595 * log(1 + \frac{f}{700}) \tag{11}$$

$$f = 700 * (10^{\frac{m}{2595}} - 1) \tag{12}$$

Mel Spectrogram and MFCC

Triangular Filter:

$$H_m(K) = \begin{cases} 0; & k < f(m-1) \\ \frac{k-f(m-1)}{f(m)-f(m-1)}; & f(m-1) \leqslant k \leqslant f(m) \\ \frac{f(m+1)-k}{f(m+1)-f(m)}; & f(m) \leqslant k \leqslant f(m+1) \\ 0; & k > f(m+1) \end{cases} \tag{13}$$

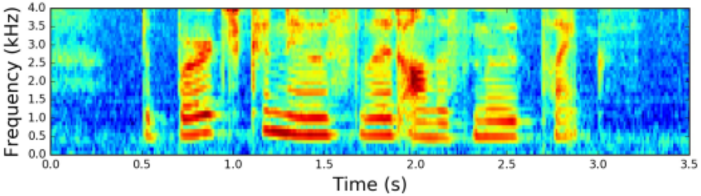


Figure 11: Spectrogram extracted from Filter Banks

Mel Spectrogram and MFCC

- DCT(Discrete Cosine Transform)

$$C_i = \sqrt{\frac{2}{M}} \sum_{j=1}^{M} k_i.cos(\frac{\pi i}{M}.j - 0.5), \forall i \in [1, K] \qquad (14)$$

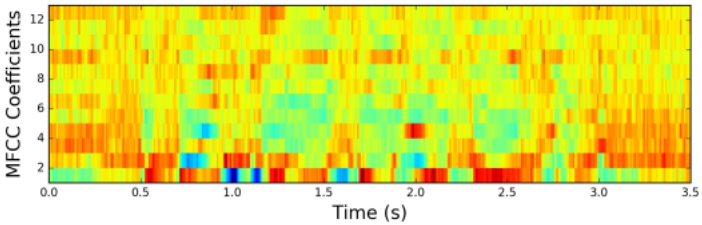


Figure 12: Spectrogram extracted by MFCC

Mel Spectrogram and MFCC

In short, MFCC only keep the K most important features in N features extracted from Filterbaks. As a results, MFCC is better at compressing data. However, DCT is a linear equation, in some special cases, MFCC left some information because speech signals have some non - linear features.

4. Audio Data Augmentation Techniques

1. Spectrogram Augmentation
   1.1 Frequency Masking
   1.2 Time Masking
2. Raw Audio Augmentation
   2.1 Time shift
   2.2 Pitch Shift
   2.3 Time Stretch
   2.4 Add Noise

Spectrogram Augmentation

SpecAugment, an augmentation method that operates on the log mel spectrogram of the input audio, rather than the raw audio. It consists 3 kinds of deformations of the spectrogram.

- Time Masking
  We mask a block of consecutive time steps on the log mel spectrogram of the input audio.
- Frequency Masking
  We mask a block of mel frequency channels on the log mel spectrogram of the input audio.

Frequency Masking

Frequency masking is applied such that f consecutive mel frequency channels [f_o,$f_o + f$] are masked . Where

- f is chosen from 0 to the frequency mask parameter F.
- f_o is chosen from [0,v-f]. v is the number of mel frequency channels.

Frequency Masking

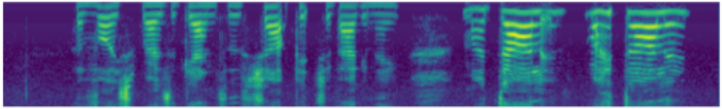


Figure 13: Log Mel spectrogram of the base input with no augmentation

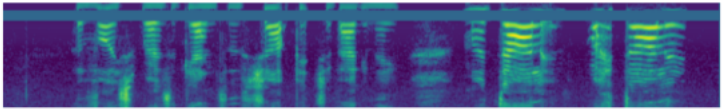


Figure 14: Log Mel spectrogram of the base input with Frequency Masking

Time Masking

Time masking is applied such that t consecutive time steps [t_o,t_o + t] are masked . Where:

- t is chosen from a uniform distribution from 0 to the time mask parameter T.
- t_o is chosen from [0,τ - f]. τ is the number of time steps.

Time Masking

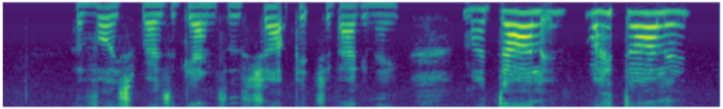


Figure 15: Log Mel spectrogram of the base input with no augmentation

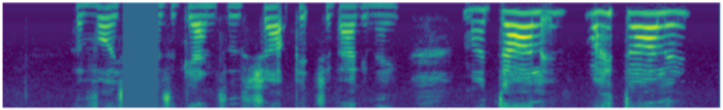


Figure 16: Log Mel spectrogram of the base input with Time Masking

Time Shift

The idea of time shift is shifting the audio to left/right with a random second. If shifting audio to left (fast forward) with x seconds, first x seconds will mark as 0 (i.e. silence). If shifting audio to right (back forward) with x seconds, last x seconds will mark as 0 (i.e. silence)

Time Shift

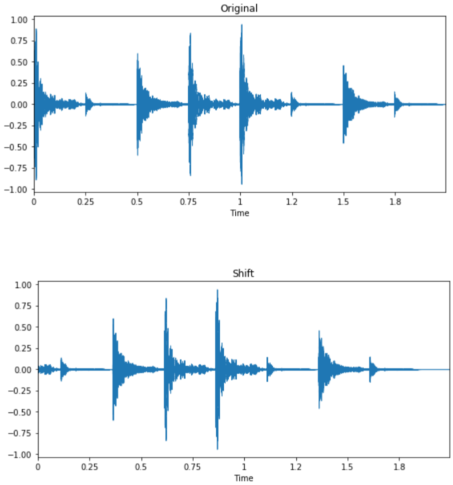


Figure 17: Comparison between original and time-shifted voice

Pitch Shift

Pitch Shift changes the pitch randomly.

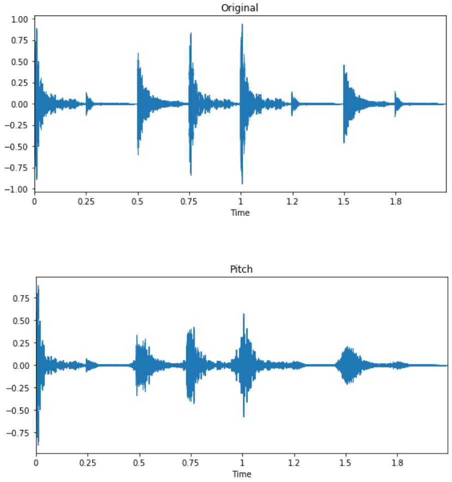


Figure 18: Comparison between original and pitch-shifted voice

Time Stretch

Randomly speed up or slow down the audio.

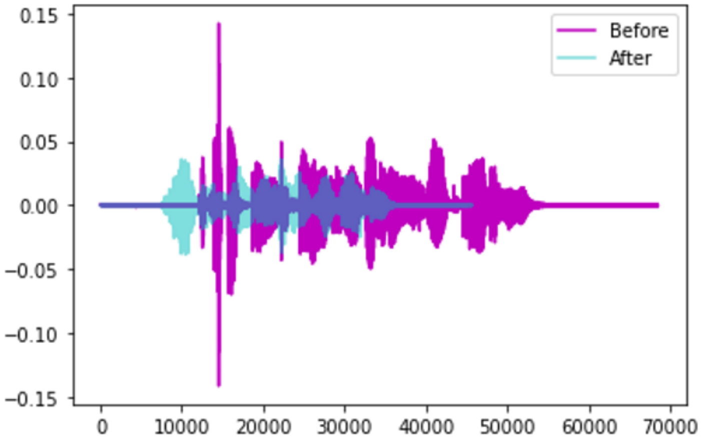


Figure 19: Augmentation by Time Stretch

Add Noise

Add some random values to the sound

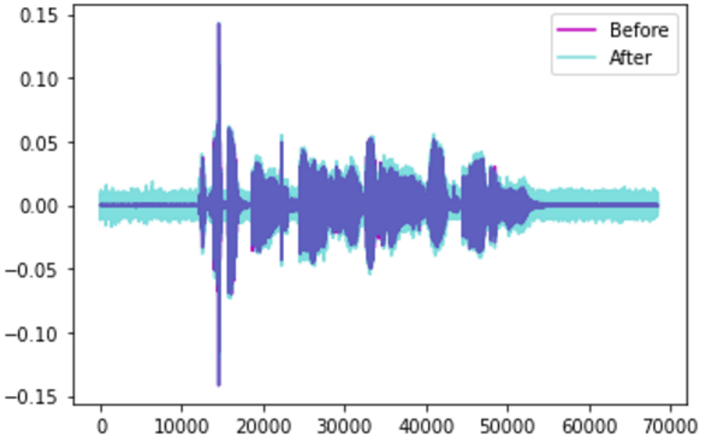


Figure 20: Augmentation by Add Noise

5. Audio Deep Learning Applications

- Text to Speech . The goals of Text to Speech is to have machines talk and make them sounds like humans of different ages and gender
- Keyword Spotting. It deals with the identification of keywords in audio stream.
- Speech To Text(Automated Speech Recognition). It enables real-time or offline transcription of audio streams into text.
- Voice Recognition. It is the ability of a program to identify a person based on their unique voiceprint. It works by scanning the speech and establishing a match with the desired voice fingerprint.

5. Audio Deep Learning Application

Comparisons	Voice Recognition	Speech Recognition
Purposes	to identify the voice owner	to identify the words of the speaker
Requirements	Need a unique voiceprint of the speaker for comparison	Need a huge dictionary to identify the speaker's words

5. Audio Deep Learning Applications

- Audio Classification are tasks to predict tags of audio clips.
- Audio Source Separation. It consists in recovering different unknown signals called sources by filtering their observed mixture.
- Audio Segmentation. Its aims are to remove the problems in ASR by detecting music and noise segment in real-time and replacing them with silence.

6. Common Public Datasets

- LJ Speech: useful for Text to Speech
- LibriVox: useful for Speech to Text, training audio embeddings
- Google Speech Commands: useful for Keyword spotting
- UrbanSound8k: useful for Audio Classification, Audio Segmentation