SPEECH RECOGNITION

MULTILAYER PERCEPTRON FOR SPEECH RECOGNITION

DATASET

- **FSDD Dataset** A simple audio/speech dataset consisting of recordings of spoken digits (0 - 9) in wav files at 8kHz
- **Recordings** There are 4 speakers in total with 2000 recordings (50 of each digit per speaker)

All About an Audio Signal

- **Audio Signal** is a three-dimensional signal in which three axes represent time, amplitude and frequency. It is a form of Analog signal
- **Analog Signal** is a continuous representation of a signal over a period of time. In an analog signal, an infinite number of samples exist between any two-time intervals
- **Sampling** is a process of converting an analog signal to a digital signal by selecting a certain number of samples per second from the analog signal, so that it can be stored and processed efficiently in memory

MFCC
Mel-Frequency Cepstral Coefficients

MFCCs of a signal are a small set of features (usually about 10-20) which concisely describe the overall shape of a spectral envelope

MFCC is one of the most important method to extract a feature of an audio signal and is used majorly whenever working on audio signals and it works even if voice is buried in harmonic noise such as echo

PROJECT OUTLINE

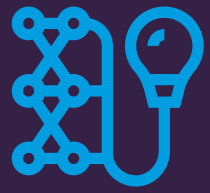
STEPS I TOOK TO SOLVE THE PROBLEM AND MAKE A MODEL

DATA PREPROCESSING

FEATURE EXTRACTION

DATA PREPARATION

VECTORIZE THE SOUND FILES



DEEP LEARNING

CONVOLUTIONAL NEURAL NETWORK

What and Why?

- **Convolution Neural Network** Sound data is spatial data, which means positions of the data points is important. In order to capture the variations of all the parts of a sound wave and preserve the spatial dependency, I made use of Convolution Neural Network

- **RELU Activation Function** I have used RELU activation function so that all the non-linearity in the data is captured. Signals less than 0 will be dropped and the good thing about RELU is that its gradient doesn’t vanish as we increase X

What and Why?

- **Max Pooling** To reduce the dimensionality of feature maps and make it more computationally efficient while preserving its features, I have used Max Pooling

- **Dropout** To train the model on several architectures of a Neural Network, we add a dropout to make it more generalized and avoid overfitting. It turns off the neurons randomly during training process

- **Categorical Cross entropy** Loss function is set to categorical cross-entropy since it is a multi-classification problem

What and Why?

- **Categorical Cross entropy** Network efficiency will be evaluated by this function and Derivative of this loss function is the back propagative error, network

- **Optimizer** I have used Adam Optimizer as Adam combines the good properties of Adadelta and RMSprop and hence tend to do better for most of the problems

- **Batch Normalization** allows each layer of the network to learn by itself a little bit more independently of other layers, it normalizes the input layer by adjusting and scaling the activations

What and Why?

- **Hidden Layers** Most of the problems are solved with 2 hidden layers, situations in which performance improves with 3rd hidden layer are very few

- **Size of hidden layer** Optimal size of hidden layer is usually between the size of the input and size of the output layer

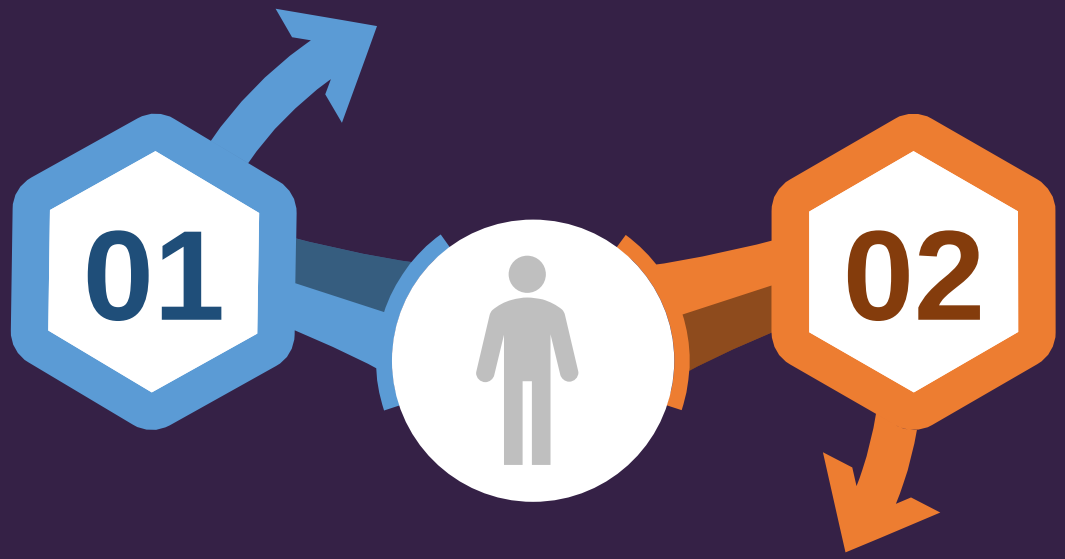
- **Batch Normalization** allows each layer of the network to learn by itself a little bit more independently of other layers, it normalizes the input layer by adjusting and scaling the activations

RESULTS

- **Correct set of weights** Correct set of weights is where the model has achieved the peak validation accuracy and minimized the loss
- **Training Accuracy** 92%
- **Test Accuracy** 91%

IMPROVEMENTS

ADDING MORE WORDS



DEPLOY IT USING FLASK

THANK YOU