Capstone Project

Machine Learning Engineer
Nanodegree

Choppa Vivek Krishna

April 7, 2018

Recognizing Speech Commands

Definition

Project Overview

Building a system to recognize few set of speech commands based on the Speech Commands Dataset provided by Tensorflow.

Problem Statement

Using the Speech Commands Dataset, we should train a system which is able to identify and classify, if any word in a specific set of words for which we have trained has been uttered

Evaluation

We have a test set as well against which we can calculate accuracy. Kaggle can provide us the score based on the submission file for the test set.

Analysis

Data Exploration and Visualization

Our training set contains .wav files with variants of 30 class labels, where each class label is a word. Out of these 30, we need to recognize only 10 words and label other words as unknown or silence. Number of training .wav files provided for each wave is around 2000 on average which is large enough for training a good model.

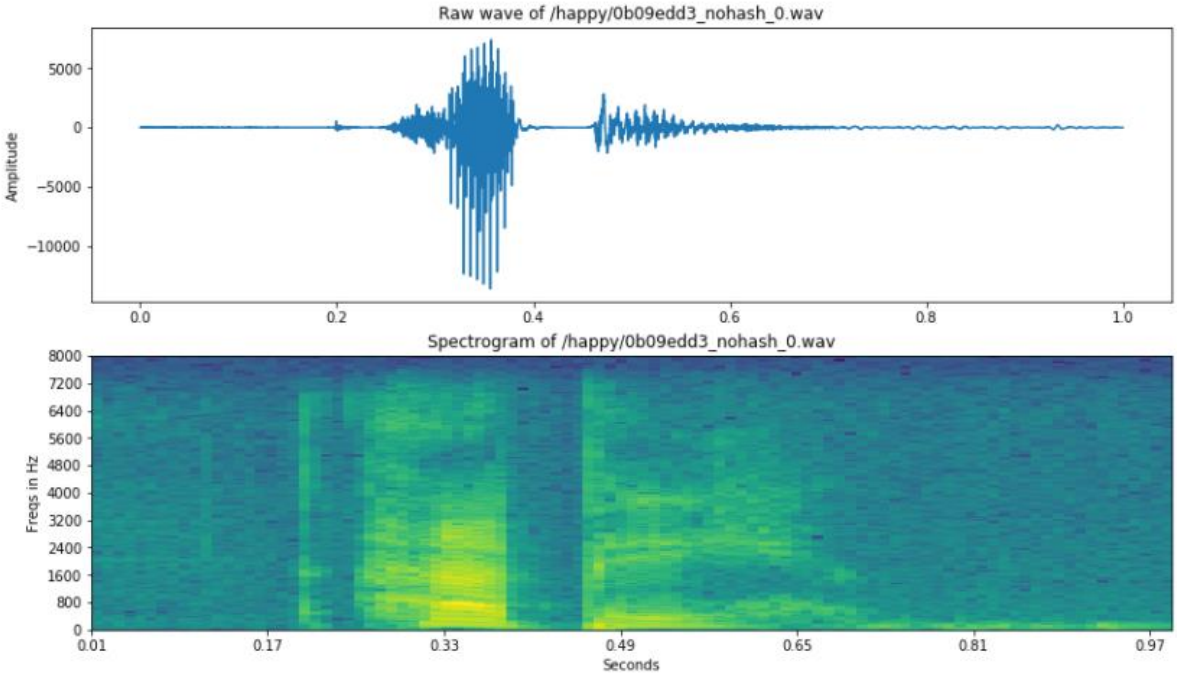
Labels to predict are:

```
['down', 'go', 'left', 'no', 'off', 'on', 'right',
'stop', 'up', 'yes', 'unknown', 'silence']
```

Out of the 64000 .wav files, nearly 4000 files were detected by VAD (voice Activation Detector) as silent or noise files. We have excluded these from training

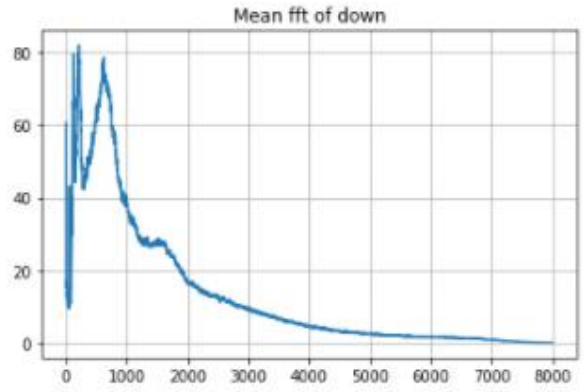
set. Few of the word pairs in the dataset like (go, no) etc. have very close pronunciation and predictions to compete closely with each other as well.

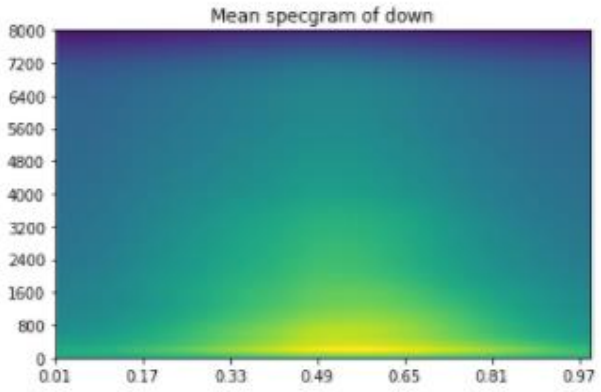
Each wave had some silence/noise padded before and after the word for which we had to use VAD and strip off frames of that sort. One such wave for example:

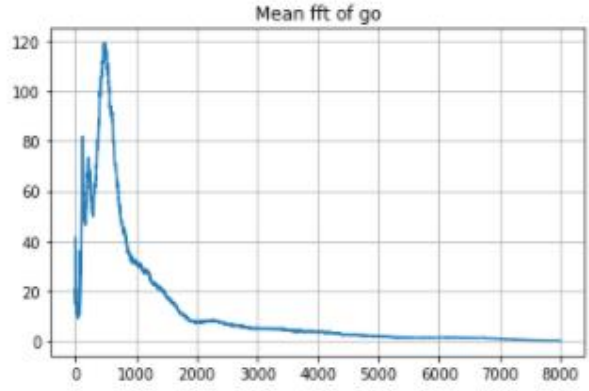


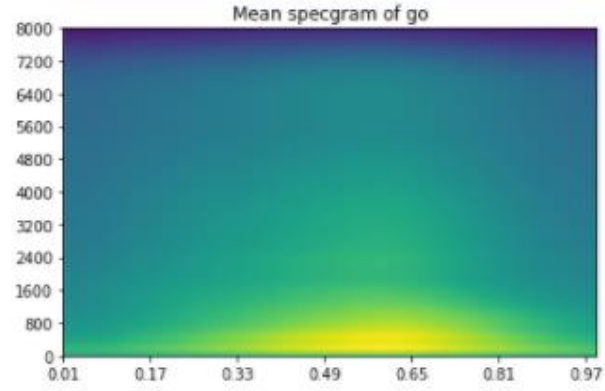
We have plotted mean fft and spectrogram of different words that needs to be classified, to decide on which features to pick for classification.

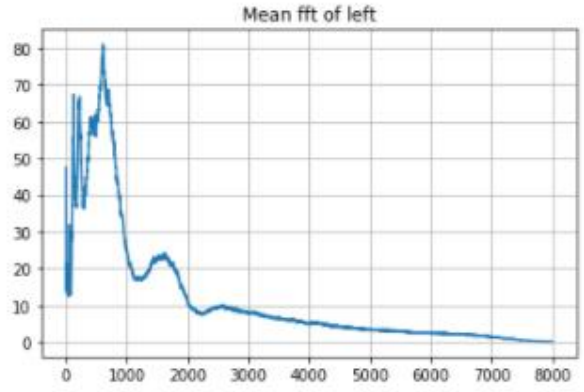
['down', 'go', 'left', 'no', 'off', 'on', 'right', 'stop', 'up', 'yes']

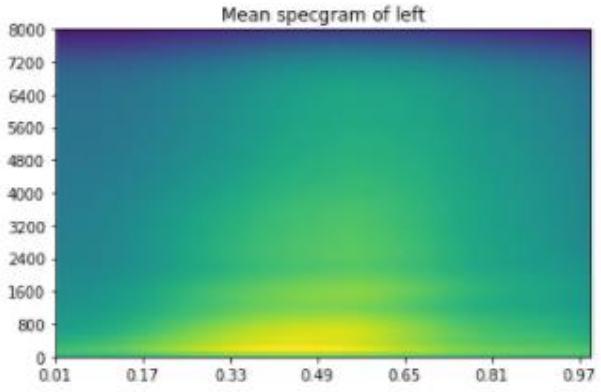


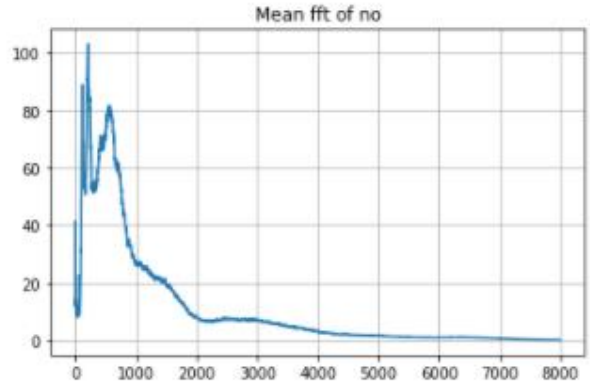


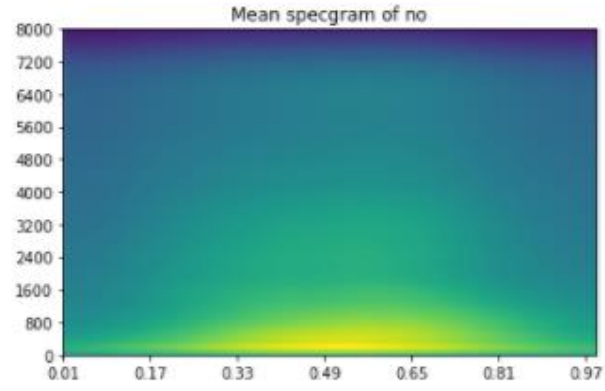


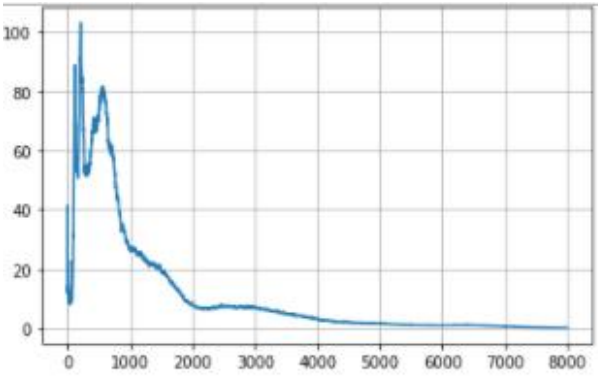


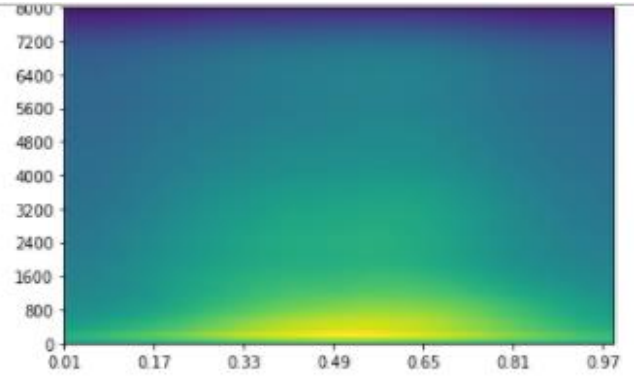


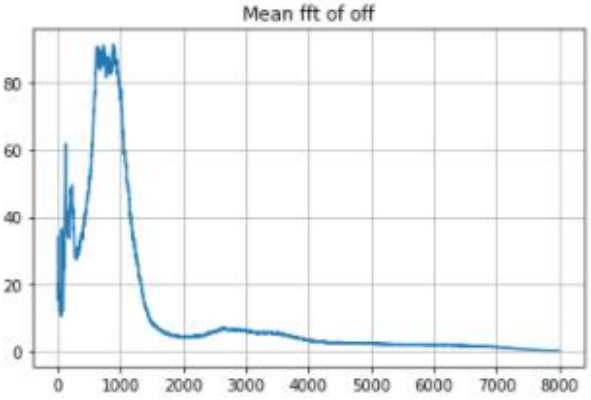


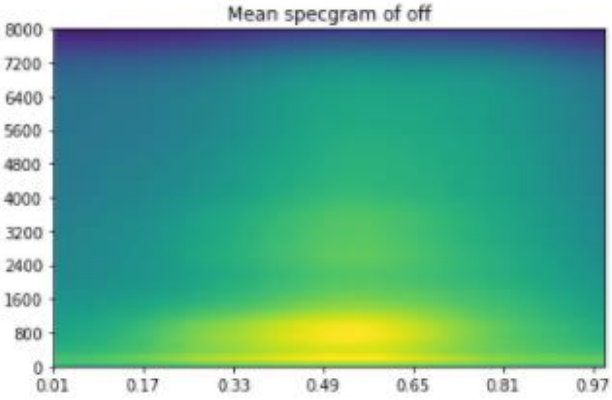


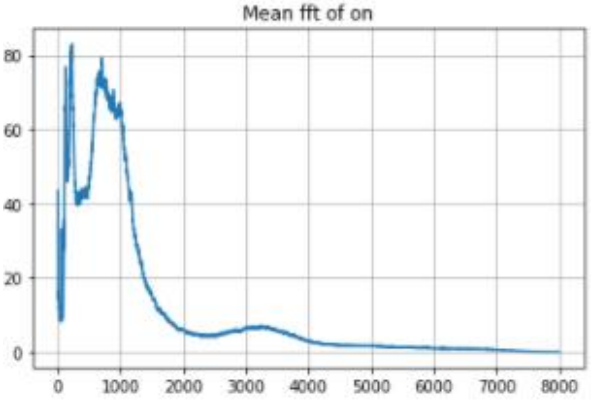


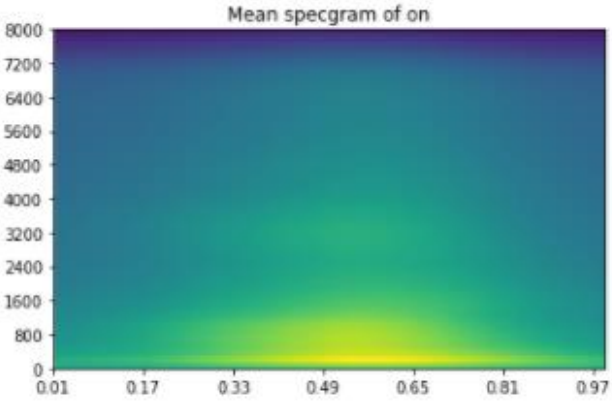

100
80
60
40
20
0
0
1000
2000
3000
4000
5000
6000
7000
8000

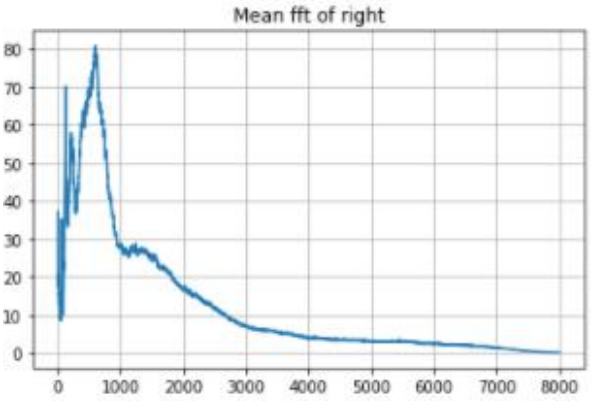

8000
7200
6400
5600
4800
4000
3200
2400
1600
800
0
0.01
0.17
0.33
0.49
0.65
0.81
0.97

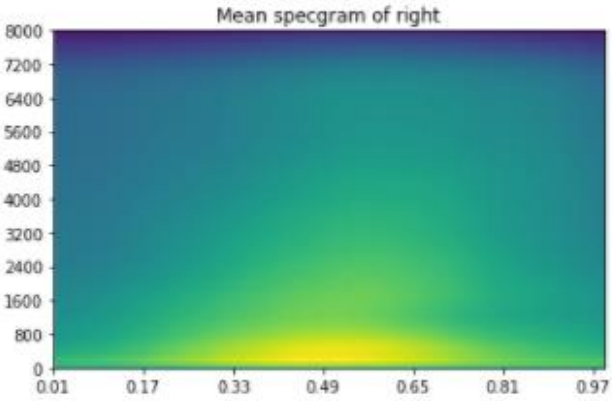

Mean fft of off
80
60
40
20
0
0
1000
2000
3000
4000
5000
6000
7000
8000

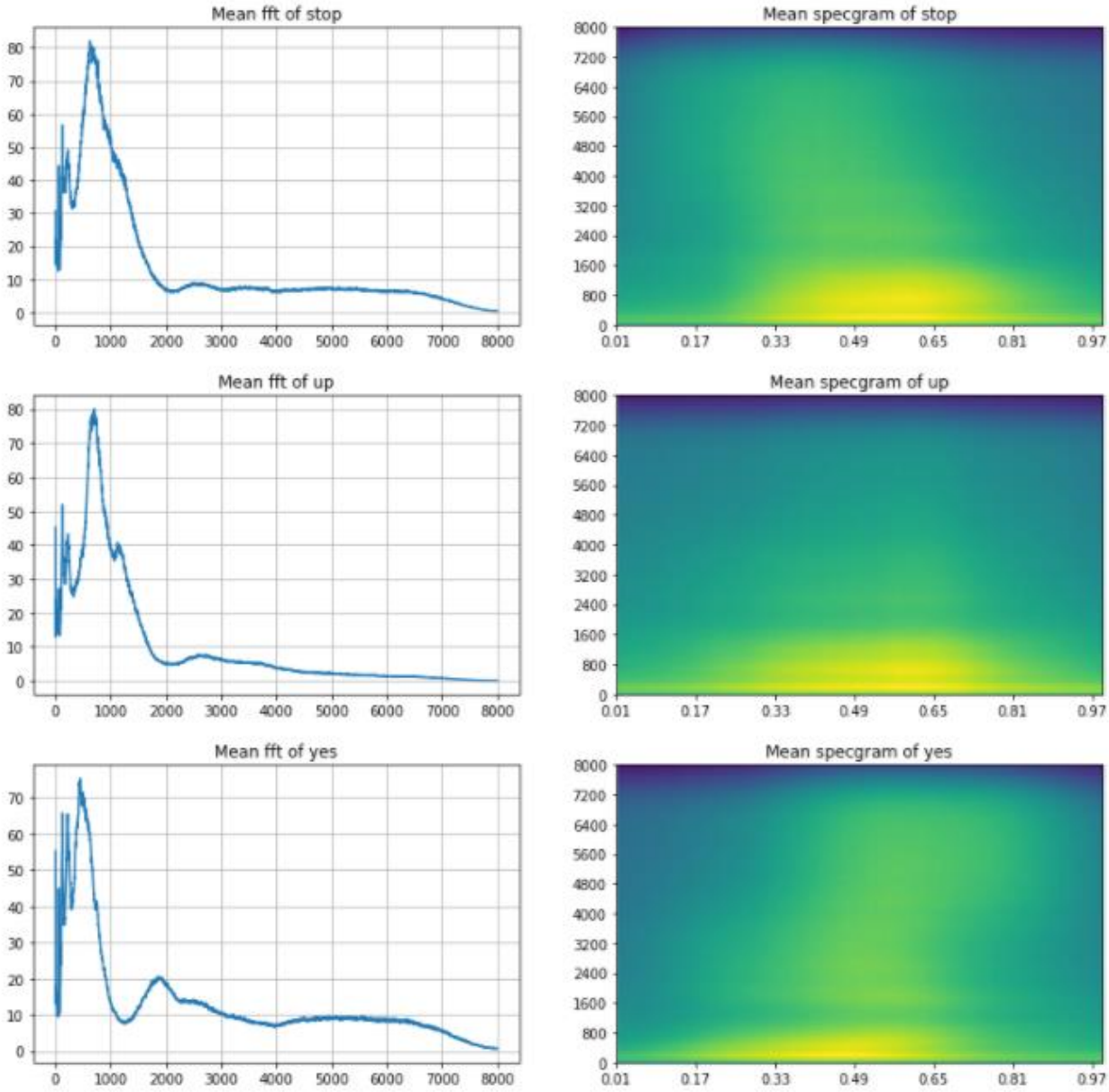

Mean specgram of off
8000
7200
6400
5600
4800
4000
3200
2400
1600
800
0
0.01
0.17
0.33
0.49
0.65
0.81
0.97


Mean fft of on
80
60
40
20
0
0
1000
2000
3000
4000
5000
6000
7000
8000


Mean specgram of on
8000
7200
6400
5600
4800
4000
3200
2400
1600
800
0
0.01
0.17
0.33
0.49
0.65
0.81
0.97

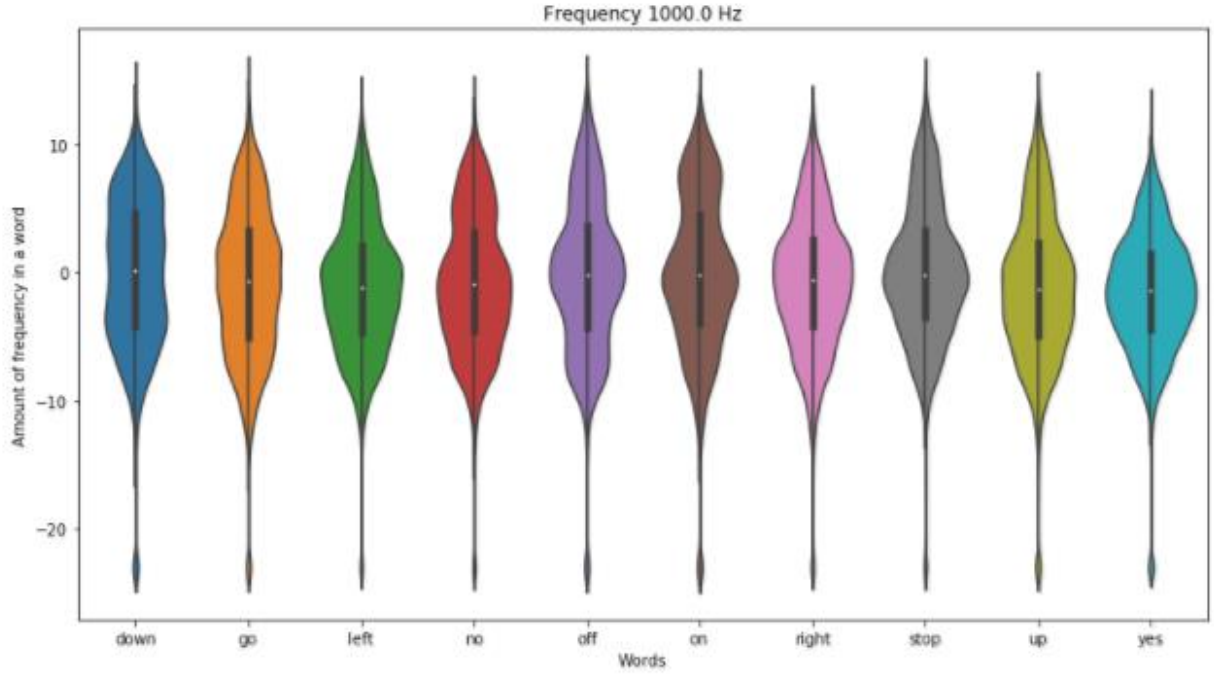

Mean fft of right
80
70
60
50
40
30
20
10
0
0
1000
2000
3000
4000
5000
6000
7000
8000


Mean specgram of right
8000
7200
6400
5600
4800
4000
3200
2400
1600
800
0
0.01
0.17
0.33
0.49
0.65
0.81
0.97



Based on the above information, fft seem to show more variation from word to word than spectrogram and possibly be better features for classification.

Then we did violin plot to identify amount of different frequencies in each word.



Feature Extraction:

After considering, spectrogram, fft and mfcc for Features to train a model, We chose to build features based on mel scale, which is inspired by how humans process speech in ears. Hence built mfcc features for each wave which is stripped off with silence using Voice Activation Detection.

We have used Librosa library to build mfcc features from a raw sound wave. It includes

1. Converting wave file into smaller frames.
2. Find the power spectrum of each frame
3. Apply mel filter bank to the spectra and sum power inside each filter.
4. Take logarithm of few filterbank energies.
5. Convert them to DCT and pick few important coefficients of the same.

These coefficients are called Mel-frequency cepstral coefficients and state of the art in Automatic Speech Recognition systems.

Later on, considering Speech information could be present in dynamics of spectral frames, rather than just the spectral envelope of frames, we added delta mfcc features as well.

Stacking both mfcc and delta_mfcc features for each frame and doing it for all 16 frames of the wave, we have ended up with features of size (16, 26).

Model:

Since speech is a time series where sentences are sequences of words and words are sequences of phonemes, we chose Long short term memory networks which can hold the memory of recent phonemes and predict the next phoneme considering the memory and subsequently word.

Though we could have tackled this problem using convolutional neural network, since the problem is restricted to words, eventually when we need to scale to sentences it makes sense to approach with LSTM.

We chose binary cross-entropy and categorical accuracy as it is a Multi-classification problem to predict our labels. Our labels are one-hot encoded and passed on to model for training.

We have probabilities for each label coming out of the model and picked the label with maximum probability after passing through a threshold value. If none of the labels is more than the threshold, we guessed the word as unknown.

For silence labels, we have passed the wave through the VAD even before passing it model for prediction and labelled accordingly.

Training:

Models tried:

0.63

```
Layer (type)                 Output Shape              Param #
=================================================================
lstm_1 (LSTM)                (None, 16, 39)            8268
_________________________________________________________________
lstm_2 (LSTM)                (None, 26)                6864
_________________________________________________________________
dense_1 (Dense)              (None, 11)                810
=================================================================
Total params: 15,942
Trainable params: 15,942
Non-trainable params: 0
```

0.65

```
Layer (type)                 Output Shape              Param #
=================================================================
lstm_1 (LSTM)                (None, 16, 39)            8268
_________________________________________________________________
lstm_2 (LSTM)                (None, 26)                6864
_________________________________________________________________
dense_1 (Dense)              (None, 30)                810
=================================================================
Total params: 15,942
Trainable params: 15,942
Non-trainable params: 0
```

Then, we have added delta features and drop out layers for regularization to avoid overfitting.

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
lstm_11 (LSTM)               (None, 16, 52)            16432
_________________________________________________________________
dropout_6 (Dropout)          (None, 16, 52)            0
_________________________________________________________________
lstm_12 (LSTM)               (None, 45)                17640
_________________________________________________________________
dropout_7 (Dropout)          (None, 45)                0
_________________________________________________________________
dense_6 (Dense)              (None, 30)                1380
=================================================================
Total params: 35,452
Trainable params: 35,452
Non-trainable params: 0
```

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
lstm_9 (LSTM)                (None, 16, 52)            16432
_________________________________________________________________
dropout_9 (Dropout)          (None, 16, 52)            0
_________________________________________________________________
lstm_10 (LSTM)               (None, 16, 45)            17640
_________________________________________________________________
dropout_10 (Dropout)         (None, 16, 45)            0
_________________________________________________________________
lstm_11 (LSTM)               (None, 45)                16380
_________________________________________________________________
dense_5 (Dense)              (None, 45)                2070
_________________________________________________________________
dropout_11 (Dropout)         (None, 45)                0
_________________________________________________________________
dense_6 (Dense)              (None, 30)                1380
=================================================================
Total params: 53,902
Trainable params: 53,902
Non-trainable params: 0
_________________________________________________________________
```

Initially we were working with threshold value of 0.5, which skipped many words that are recognized around 0.25 and 0.3 max probabilities. After reducing threshold to 0.1, score has crossed 0.7 on the leaderboard test set.

Initially we assumed that for some words in test set that are other than these 30 words (among which we have to recognize 10 words as is and other twenty as unknown), the model will probably not return probabilities more than the threshold for any of the labels and hence we can consider that word as unknown. However, since it is hard to find proper threshold of that kind even if that exists, we introduced a new label called Unknown unknowns which are some words apart from these 30.

It is probably tough for the model to converge all other words with varied features to aggregate as single class. We also need to prepare some test data with random words to this Unknown unknown's class.

It took around 30 minutes to train each epoch in my CPU and the model used to take around 15 to 20 epochs to converge. It is very time consuming to experiment each time to train the model.

Conclusion

After trying out with different models and feature data, the best model obtained has the unknown unknowns (words not in training set considered) and have been trained on 31 labels with MFCC features. One more binary classification for detecting Silence in the wave file is employed. We have stripped off silent frames in the audio file before training. Model with LSTM layers considering sequences and memory states with dropout layers for generalization have been fruitful.