Speech Graphics - Speech Mode Classification

Sambit Paul

September 24, 2018

Abstract

This report outlines the project done for speech mode classification. The idea is to use deep learning methods to classify audio files for types of speech modes - rap, singing and speech. The approaches explored here use features from the data for classification using simple classifiers like K-nearest neighbours and SVMs, to deep learning models like feed-forward networks, LSTM-Recurrent Networks and Convolutional Neural Networks. The features used for almost all the models were the MFCC features except for the CNN which used the spectrogram of the audio files. From the experiments, it was observed that the LSTM-RNN using the MFCC features was able to classify with the highest accuracy.

1. Problem Description

The objective is to classify audio segments on basis of three modes — rap, singing and speech. The dataset provided here contains a total of 1440 audio recording each of which has a length of 3 seconds and are recorded at 16kHz. Each mode of speech has 480 examples associated to them.

The idea is to extract features from the audio files and use them for classifying between the three given modes. The data needs to be distributed such that 10% of the dataset is in the balanced test set.

2. Feature Extraction

One of the key challenges for speech recognition is selecting the optimal feature set which will generate the best results. The problem as stated in Section 1 is a basic problem in speech classification which will work well given the same features as most problems in Automated Speech Recognition.

The first part discusses about Mel-Frequency Cepstral Coefficients (MFCC) and Linear Prediction Cepstral Coefficient (LPCC) and in the second part we look into spectrograms and their potential use in this classification problem.

MFCC is based on human hearing such that the filters below 1kHz is linearly spaced and the ones above 1kHz are logarithmically spaced. MFCC operates by converting the signal from time domain to frequency domain using Fourier Transform and then, each Mel filter computes a weighted sum of its spectral components. Discrete Cosine Transform (DCT) applied to the log Mel spectrum produces the MFC coefficients [Muda et al., 2010].

LPCC works on the principle that a speech sample at the current time can be determined by a linear combination of previous speech samples. LPCC operates by applying Linear Predictive

Analysis on the speech signal followed by a Cepstrum Analysis which finds the cepstrum coefficients of the speech sequence [Gulzar et al., 2014].

MFCC is considered better than LPCC for most problems in Automatic Speech Recognition [Rana and Miglani, 2014]. Figure 1 shows MFCC samples taken from each speech mode with 13 cepstrums.

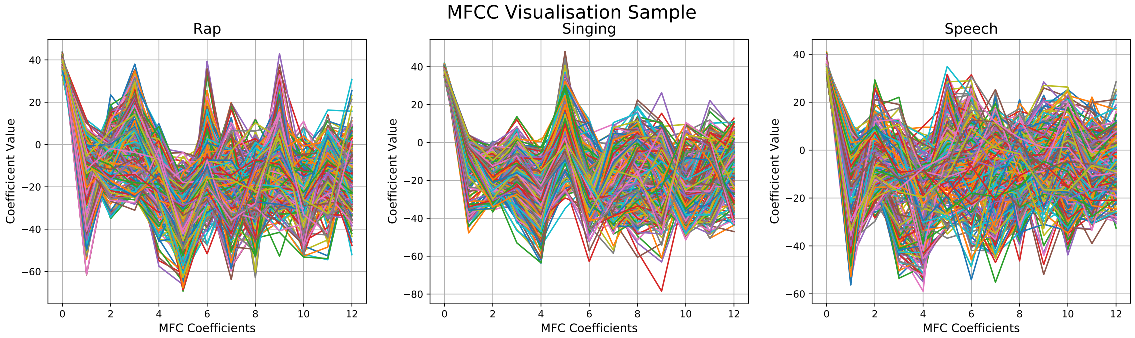


Figure 1: Visualisation of MFCC for each speech mode

One of the features that could potentially also be useful is the spectrogram. There is a significant difference in the spectrograms of the different speech modes. Figure 2 shows sample spectrograms of the different speech modes. It can be that there are singing has a dominant higher frequency throughout the sample compared to other modes, while speech has gaps at some intervals in the frequency map.

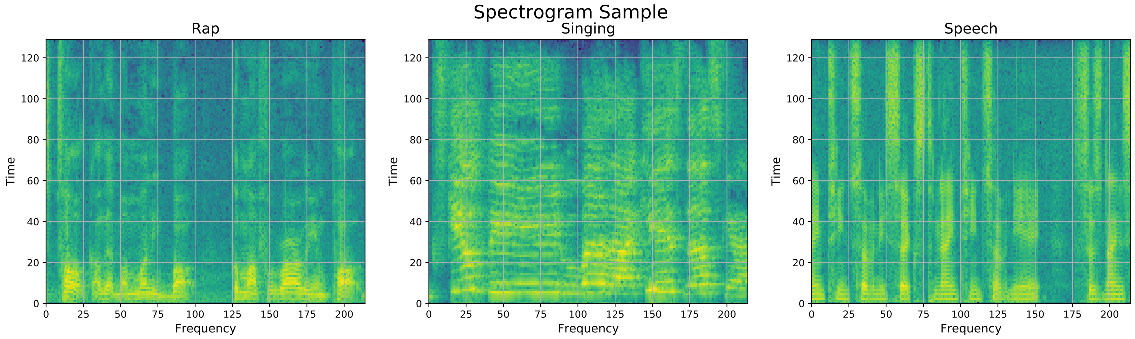


Figure 2: Sample spectrogram for each speech mode

3. Traditional Methods

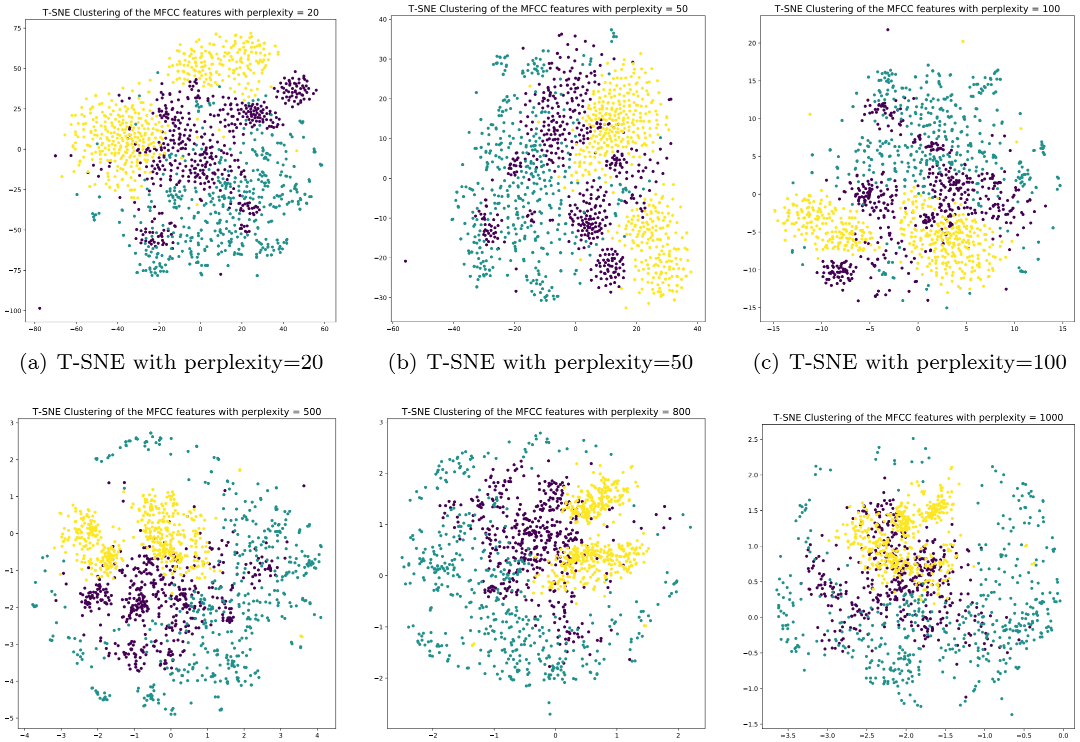
This section aims to visualise the data using T-distributed Stochastic Neighbour Embedding (t-SNE) and explores a couple of traditional machine learning methods to classify among the three speech modes.

In this section, we break the dataset into 1296 examples in the training set and 144 examples in the test set. The features used are the MFCC features as extracted using the python-speech-features library [Lyons, 13].

3.1. Visualisation with t-SNE

t-SNE is a tool for visualising high dimensional data [Maaten and Hinton, 2008]. We are trying to cluster each audio file based on their MFCC values. Due to the high-dimensionality of the data (299 × 13 for each file), we are using Principal Component Analysis (PCA) to reduce dimensions due to computational limitations. We use 200 components in the PCA which has an explainability factor of 77.13%.

Figure 3 shows projections of the data for different perplexities but as is visible, there are no distinct clusters visible for any value of perplexity.



(a) T-SNE with perplexity=20

(b) T-SNE with perplexity=50

(c) T-SNE with perplexity=100

(d) T-SNE with perplexity=500

(e) T-SNE with perplexity=800

(f) T-SNE with perplexity=1000

Figure 3: T-SNE projections of the dataset for perplexity values of 20, 50, 100, 500, 800 and 1000

3.2. K-Nearest Neighbours

The first traditional method used is the K-nearest neighbours classifier which for 2 number of neighbours setting. The first experiment was using 5 neighbours which yielded a test accuracy of 63.19%. The second experiment was using 7 neighbours which yielded a test accuracy of 61.81%.

3.3. Support Vector Machine

The second method used support vector machines based on Linear Support Vector Classification with L2 regularisation. This method yielded a test accuracy of 71.53%. We can consider this as the baseline for further experiments.

4. Deep Learning Models

This section explores different deep learning models and their performance compared to traditional machine learning models.

In this section, we break down the dataset into three parts — the training set which contains 1101 examples, the validation set which contains 195 examples and the test set which contains 144 examples.

For the feed-forward network and LSTM-RNN, we use the MFCC features and for the CNN, we use images of the spectrogram as inputs. The loss function for all the models is the Cross Entropy loss which means that the class with the highest energy value is the most probable class.

4.1. Feed-Forward Model

This section describes the feed-forward model used for classification using a flattened form of the 299×13 dimensional MFCC. The architecture has 5 fully connected layers starting with 3887 (input layer) followed by 1024, 256, 64, 8 and finally 3 output nodes each holding the cross-entropy energies of each class. Figure 4 shows the feed-forward architecture used here.

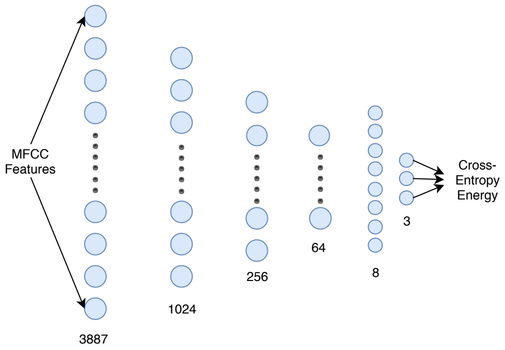


Figure 4: Feed-Forward Network Architecture

The training-validation loss curve for the feed-forward architecture is shown in Figure 5. The test set accuracy for this model was 84.03%.

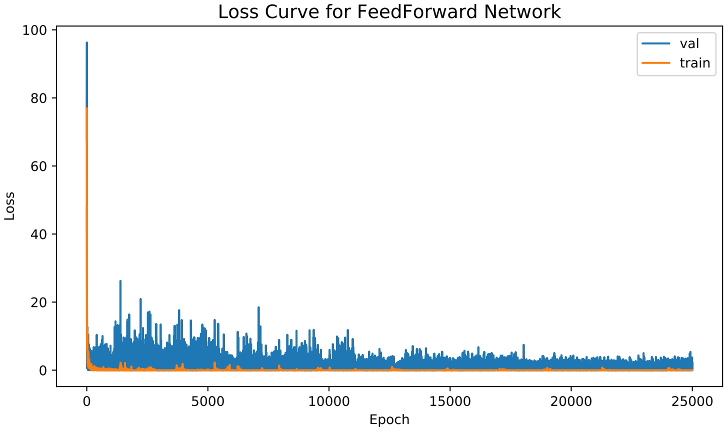


Figure 5: Training-Validation Loss Curve

4.2. Convolutional Neural Network

This section describes the CNN used for classification using the spectrogram images. The model has 5 convolutional layers followed by a fully-connected dense network which gives the cross entropy energies of each class at the end. A detailed architecture is given in Figure 6.

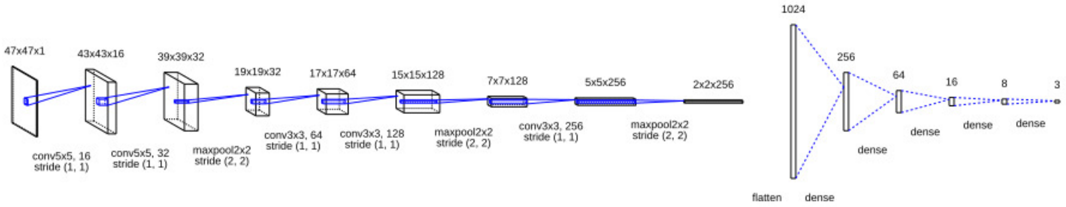


Figure 6: CNN Architecture

The training-validation loss curve for the feed-forward architecture is shown in Figure 7. It can be observed, on comparison with the feed-forward model loss (Figure 5), that the loss values have a much smaller range of variance. This can be used to infer that this model should give a better performance.

The test set accuracy for this model was 88.19%.

4.3. Long Short-Term Memory RNN

This section describes the LSTM - Recurrent Neural Network used for classification using the MFCC features. We have used two models here, one with a single LSTM layer and another with 2 LSTM layers. The LSTM takes 13 dimensional input with a sequence length of 299. It was observed that

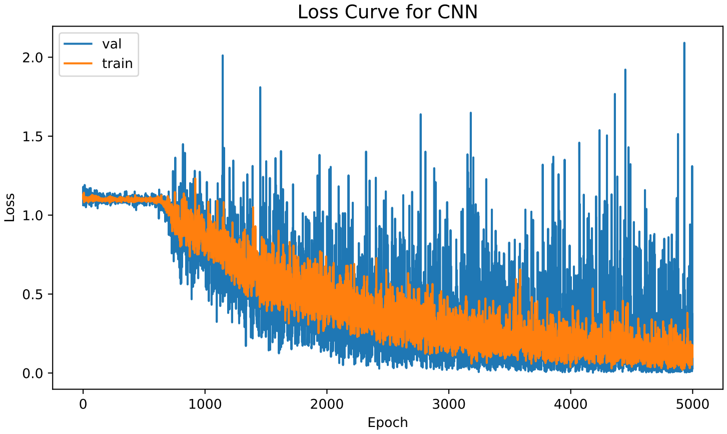


Figure 7: Training-Validation Loss Curve

the 2 layer LSTM performed better than the single layer LSTM. The architecture of the 2 layer LSTM network is given in Figure 6.

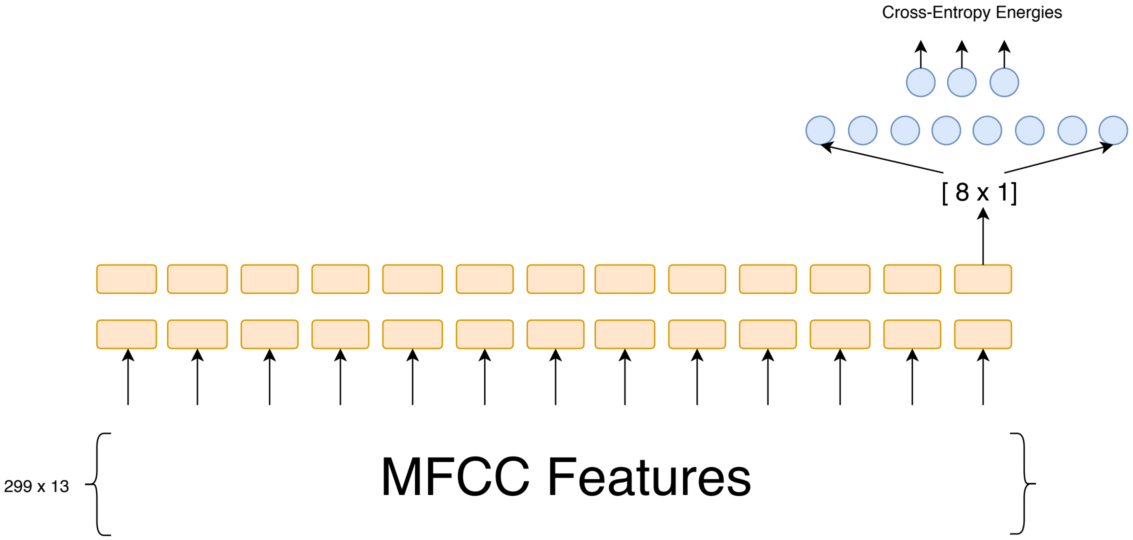


Figure 8: LSTM Architecture

The training-validation loss curve for the feed-forward architecture is shown in Figure 9. Comparing to the CNN performance as shown in Figure 7, it can be observed that the losses are almost similar and hence, the performance can also be expected to be nearly same.

The test set accuracy for the single layer model was 86.11% while the 2-layer model was 89.58%.

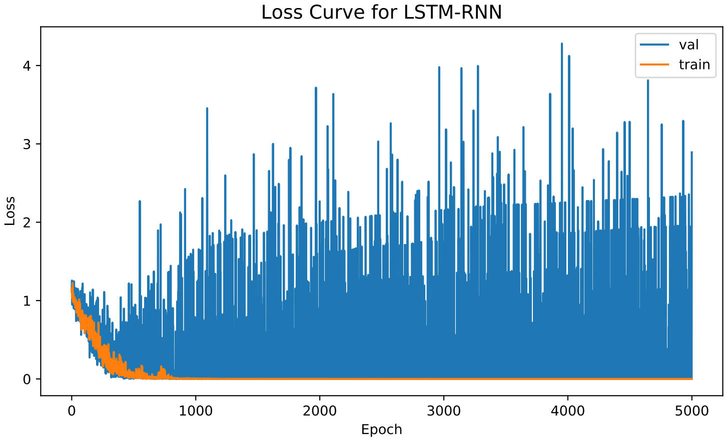


Figure 9: Training-Validation Loss Curve

5. Results & Analysis

Table 1 gives a summary of the test performance of all the methods used.

As is visible, the traditional methods like KNN and SVM do not perform at par with deep learning methods. This implies that for this dataset, the classification boundaries are not linear and therefore, introduction of the non-linear activations in the deep networks helped achieve much better performance.

Moreover, the fact that LSTM networks generated $\sim$ 5% improvement over the feed-forward network reveals the importance of exploiting the sequential nature of the data.

Although, we can observe that LSTMs have performed better than CNNs by $\sim$ 1%, further analysis of hyper-parameters and architecture needs to be done to see if CNNs with spectrogram inputs can perform better than LSTM networks.

Method	**Test Accuracy**
KNN-5	63.19%
KNN-7	61.81%
SVM-LinearSVC	71.53%
Feed Forward Network	84.03%
Convolutional Neural Network	88.19%
LSTM-RNN-1	86.11%
LSTM-RNN-2	**89.58%**

Table 1: Test accuracies of the different methods explored in this project. Best performing model in bold.

6. Real-Time Classification

For real-time classification, using an LSTM model would be of useful since they do not require fixed length inputs. If each second is recorded at 16kHz, we would get 16000 datapoints which when passed through MFCC algorithm (frame size = 0.025, frame step = 0.01, number of cepstrum = 13), should return a 98 × 13 feature matrix based on Equation 1.

$$n_{frame} = \frac{|seq_{len} - (frame_{size} * SR)|}{(frame_{stride} * SR)} \tag{1}$$

where,
$n_{frame} = Number of frames$
$seq_{len} = SequenceLength$
$frame_{size} = FrameSize$
$frame_{stride} = FrameStride$
$SR = SamplingRate$

Another technique that would be interesting to try out would be a recurrent convolutional network which can take in spectrograms generated in real-time and find out correlations between subsequent spectrograms. It will potentially be able to learn how transitions between different speech modes work, which can be critical for classification.

References

[Gulzar et al., 2014] Gulzar, T., Singh, A., and Sharma, S. (2014). Comparative analysis of lpcc, mfcc and bfcc for the recognition of hindi words using artificial neural networks. *International Journal of Computer Applications*, 101(12):22–27.

[Lyons, 13] Lyons, J. (2013–). python-speech-features: Provides common speech features for asr including mfccs and filterbank energies for Python.

[Maaten and Hinton, 2008] Maaten, L. v. d. and Hinton, G. (2008). Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605.

[Muda et al., 2010] Muda, L., Begam, M., and Elamvazuthi, I. (2010). Voice recognition algorithms using mel frequency cepstral coefficient (mfcc) and dynamic time warping (dtw) techniques. *arXiv preprint arXiv:1003.4083*.

[Rana and Miglani, 2014] Rana, M. and Miglani, S. (2014). Performance analysis of mfcc and lpcc techniques in automatic speech recognition. *International Journal Of Engineering And Computer Science*, 3(08).