International Journal of Research in Engineering, Science and Management
Volume 5, Issue 1, January 2022
https://www.ijresm.com | ISSN (Online): 2581-5792

Tamil Natural Language Voice Classification using Recurrent Neural Networks

Nishaanth Kanna Ravichandran*
Student, Department of Computer Science and Engineering, PSG College of Technology, Coimbatore, India

***Abstract*: Audio Classification systems are used to classify the given audio into N outputs. Various models can accurately classify a given sound. But few can accurately classify a given Natural Language (Tamil) Voice, especially Tamil Vowels pronounced by Children with learning disabilities. Voice classification is done by recording the audio and converting it into digital data. The audio samples then undergo a feature extraction process to extra Mel Frequency Cepstral Coefficients. These coefficients are then used as input to the Deep Recurrent Neural Networks (DRNN) (LSTM) to accurately classify them into Tamil vowels. This paper focuses on a learning system (android app), with an on-device inference model developed using TensorFlow Lite, that records the children's voice data, classifies them, and provides accuracy and training to improve their pronunciation.**

***Keywords*: Deep Recurrent Neural Network (DRNN), Long-Short Term Memory (LSTM), Mel Frequency Cepstral Coefficient (MFCC), TensorFlow Lite.**

1. Introduction

Natural language voice classification is a difficult problem to solve especially for languages that have a similar sounding vowel. With the advent of Deep Learning Algorithms that excel at identifying minor differences across the given data and powerful hardware to handle the requirements of the algorithm, the voice classification problem is possible to solve. Natural language voice classification is a very useful system for helping children with learning disabilities who find it difficult to pronounce vowels that sound similar. It Helps them by showing if their pronunciation is accurate and understandable.

Till now, a variety of machine learning and signal processing methods have been used for audio classification including using matrix factorization [1]–[3], filter banks [4], [5] hidden Markov models, and Deep Neural Networks [6]–[9]. In a particular approach to classifying audio from construction sites [10], Deep Recurrent Neural Networks (DRNN) based on Long-Short Term Memory (LSTM) was used to achieve an accuracy of 97% across five classes. This approach used a series of spectral features, like MFCCs, Mel-scaled Spectrogram, Chroma, and Spectral contrast, as an input for the DRNN.

Another challenge in working with Deep neural networks is that they are particularly dependent on the availability of large quantities of training data, especially bigger models, to learn a function and generalize well and provide high classification accuracy on unseen data. Unfortunately, there are no open source large data sets for the Tamil language with vowels. To solve the Tamil natural language voice data insufficiency problem, we used *data augmentation*, that is, the application of one or more deformations to a collection of annotated training samples which result in new, additional training data. An important concept of data augmentation is that the transformations applied to the labeled data do not change its meaning of it. For example, Images used in computer vision problems [11]–[13]can be rotated, translated, mirrored, or scaled image and it will still be a coherent image, so it is possible to apply these transformations to produce additional training data while maintaining the meaning of it. By training the network on the additional transformed data, the hope is that the network becomes invariant to these transformations and generalizes better to unseen data. Semantics-preserving deformations have also been proposed for the audio domain, and have been shown to increase model accuracy for music classification tasks [13]. However, in the case of natural language classification, the application of data augmentation has been relatively limited, with the author of [7] using different random combinations of time-shifting, pitch shifting, and time stretching for data augmentation on environmental sounds. But they reported that the simple augmentation techniques were unsatisfactory for the UrbanSound8k dataset given the considerable increase in training time and a minor impact on accuracy. We have outlined the augmentations performed on natural language to produce additional data and help the deep recurrent neural network to generalize and recognize Tamil voices with different pitches, speeds, and noise better.

In this paper, a Natural Language Voice Classification system is proposed that is using MFCCs as feature extraction and as an input to the DRNN, and instead of simple homogeneous construction audio, complex natural language audio is used thus increasing the complexity of the problem. We have trained our DRNN on natural language voice recorded with correct pronunciations with noise. We show that the proposed DRNN architecture yields state-of-the-art performance for Tamil natural language voice classification.

2. Method

A. Deep Recurrent Neural Network

Recurrent Neural Networks are widely used to detect patterns

*Corresponding author: nishaanthkanna@gmail.com

in a sequence of Data like handwriting, speech, text, genomes, or numerical time series because they have cyclical connections to nearby neurons. But the major drawback with Simple Recurrent Neural Networks is the range of context that can be assessed is limited. That is, the network output either reduces or increases significantly as it cycles around the network's recurrent connections. This is referred to as the *vanishing gradient problem* [14].To avoid this problem, we are using Long Short-Term Memory cells. The LSTM avoids the vanishing gradient problem by replacing summation units with memory blocks. These blocks help store and access information over long periods, helping reduce the vanishing gradient problem. LSTMs have been applied to various problems such as music generation, speech recognition, and handwriting recognition. The advantages of LSTM are more pronounced for problems requiring the use of long contextual information. We are covering the audible frequency of 0-22050 Hz, using a window size of 1024 samples at 44.1 kHz and a hop size of the same duration. We are extracting 40 MFCCs from the audio file. We fix the size of the input x to 4 seconds, i.e. $x \in \mathbb{R}^{40}$.

Given our input x, the network is trained to learn the parameters Θ of a composite nonlinear function $f(.|\theta)$ which maps x to the output (prediction) z:

$$z = f(x|\theta) = f_L(\dots f_2(f_1(X|\theta_1)|\theta_2)|\theta_L),$$

Where each operation $f_l(.|\theta_l)$ is referred to as a layer of the network, with L = 4 layers in our proposed architecture. The four layers are Long-Short Term Memory cells expressed as:

$$f_t = \sigma(X_t * U_f + H_{t-1} * W_f)$$
$$\bar{C_t} = tanh(X_t * U_c + H_{t-1} * Wc)$$
$$l_t = \sigma(X_t * U_i + H_{t-1} * W_i)$$
$$O_t = \sigma(X_t * U_o + H_{t-1} * W_o)$$

$$C_t = f_t * C_{t-1} + l_t * \bar{C_t}$$
$$H_t = \mathrm{O_t} * \tanh(\mathrm{C_t})$$

Where X_l is a 1-dimensional input tensor consisting of N *features*, H_{t-1} is the previous cell output, C_{t-1} is the previous cell memory and W, U are weight vectors (simplified for display). Similar to the proposed learning features applied to audio samples from construction specified in this paper [10], we use a constant learning rate of 0.01. Dropout [15] is applied to the input of the first two layers

B. Mel-frequency cepstral coefficients

The feature extraction of the acoustic signals is an important task to produce a better recognition performance. The efficiency of this phase is important for the next phase since it affects its behavior. Many algorithms such as Linear Predictive Cepstral Coefficient (LPCC), Human Factor Cepstral Coefficient (HFCC), and Mel Frequency Cepstral Coefficients (MFCC) can be used for feature extraction. MFCC has been selected because it is less complex and has produced good results for other audio classifications tasks [16]–[18]. MFCC is based on human hearing perceptions that cannot perceive frequencies over 1Khz. In other words, MFCC is based on a known variation of the human ear's critical bandwidth with frequency [19]. MFCC has two types of filters which are spaced linearly at a low frequency below 1000 Hz and logarithmic spacing above 1000Hz. A subjective pitch is present on Mel Frequency Scale to capture important characteristics of phonetics in speech. The complete process of feature extraction of MFCCs is illustrated in figure 1.

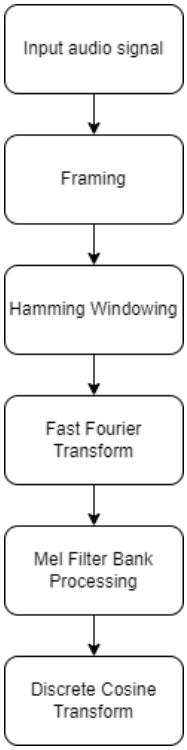

Fig. 1. Extracting Mel Frequency Cepstral Coefficient

C. Data Augmentation

We used four different audio data augmentations similar to the data augmentations performed on environment sounds [20]. Each transformation is applied directly to the audio signal before passing it to the feature extraction model (Mel-Frequency Cepstrum Coefficients). Note that for each augmentation it is important that we choose the transformation parameters such that the meaning and pronunciation of the natural language audio is maintained. The transformations and resulting augmentation sets are described below:

- *Time Stretching:* Each sample was time-stretched by 2 factors: 0.8x and 1.2x. The samples were slowed or speed up while maintaining the same pitch.
- *Pitch Shifting 1:* Each sample was pitch-shifted by 2 values: -1; 1. The sample was either lowered or raised while keeping the duration of the audio sample unchanged.
- *Pitch Shifting 2:* pitch shifting was a particularly beneficial augmentation, so we created a second augmentation set. This time each sample was pitch-shifted by 4 larger values: -3.5*;* -2.5*;* 2.5*;* 3.5.
- *Background Noise:* we start by mixing the sample with another recording containing background sounds. Each

sample was mixed with 3 acoustic scenes: street traffic, ceiling fan noise, people talking. Each augmented audio sample y was generated using $y = (1 - w) \cdot x + w \cdot y$ where $x$is the audio signal of the original sample, y is the signal of the background noise and w is the weighting parameter that was chosen randomly from a uniform distribution ranging from [0.2,0.8].

D. Android App using TensorFlow Lite

For ease of use for kids with learning disabilities a companion app (Figure 2 and 3) was developed that showcases the sound of the Tamil vowels and recorded their voices and passed the data to a natural voice classification system which extracted the MFCCs from the audio files.

Fig. 2. Homepage of the app with intuitive art

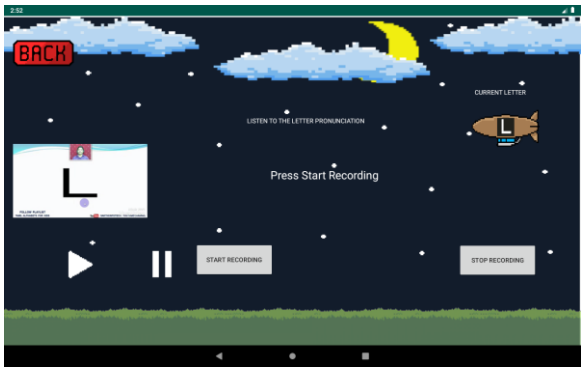

Fig. 3. Where the user learns the pronunciation of the vowels and speaks it to the classification system

The Deep Recurrent Neural Network developed using TensorFlow was trained on a PC with an Intel CPU. The model was then exported to be used in Android using TensorFlow Lite. In the app, the extracted MFCCs are passed to the TensorFlow Lite model as inputs and it outputs the confidence for the four vowels and the vowel with the highest confidence if picked. If that vowel does not match the user-selected vowel a pop-up is shown asking them to re-try.

E. Technical walkthrough of architecture formation

All audio was recorded in a lossless format in natural environments and with common school background noise (spinning fan, faint talking sounds). The recorded wav format was augmented, as detailed before, making the final dataset count to 1200 with each vowel having 300, 250 for training, and 50 for testing.

The Python Library Librosa was used to manipulate the wav files. The mfcc function from *librosa.feature* was used to extract the 40 MFCCs from the audio file with the following parameters: sampling_rate = 22050, n_mfcc = 40 and hop_length = 1024.

Using Librosa, the audio waveforms of all the vowels were visualized (as shown in figure 4).

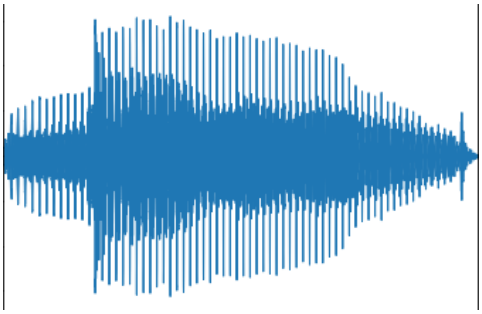

Fig. 4. Sample waveform for a vowel

To develop the Deep Recurrent Neural Network (LSTM), TensorFlow a Python Library developed by Google was used. The architecture of the DRNN is displayed as a table below.

Table 1
The architecture of the proposed Deep Recurrent Neural Network

Proposed Deep Recurrent Neural Network
LSTM – 40
DROPOUT
LSTM – 20
DROPOUT
LSTM – 20
LSTM – 10
SOFTMAX

3. Output Results

Tuning the hyperparameters, the highest accuracy was achieved. Using the Adam optimizer, an accuracy of 0.62 was reached over 20 epochs and a batch size of 20. The Stochastic Gradient Descent optimizer gave an accuracy of 0.57 and it could not be increased significantly by further tuning of the hyperparameters.

4. Conclusion

This paper explains the Natural Language Speech Classification for the Tamil language that has two major phases, using python Librosa to extract MFCCs from the recorded audio, providing the extracted MFCCs as input to the deep recurrent neural network, and developing an Android app to utilize these features. Thus, children with learning disabilities can use this app to learn the pronunciation of Tamil vowels.

References

[1] V. Bisot, R. Serizel, S. Essid, and G. Richard, "Acoustic scene classification with matrix factorization for unsupervised feature learning," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2016, pp. 6445–6449.

[2] E. Benetos, G. Lafay, M. Lagrange, and M. D. Plumbley, "Detection of overlapping acoustic events using a temporally-constrained probabilistic

model," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2016, pp. 6450–6454.

[3] A. Mesaros, T. Heittola, O. Dikmen, and T. Virtanen, "Sound event detection in real life recordings using coupled matrix factorization of spectral representations and class activity annotations," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2015, pp. 151–155.

[4] J. Salamon and J. P. Bello, "Feature learning with deep scattering for urban sound analysis," in *2015 23rd European Signal Processing Conference (EUSIPCO)*, 2015, pp. 724–728.

[5] J. T. Geiger and K. Helwani, "Improving event detection for audio surveillance using Gabor filterbank features," in *2015 23rd European Signal Processing Conference (EUSIPCO)*, 2015, pp. 714–718.

[6] E. Cakir, T. Heittola, H. Huttunen, and T. Virtanen, "Polyphonic sound event detection using multi label deep neural networks," in *2015 International Joint Conference on Neural Networks (IJCNN)*, 2015, pp. 1–7.

[7] K. J. Piczak, "Environmental sound classification with convolutional neural networks," in *2015 IEEE 25th International Workshop on Machine Learning for Signal Processing (MLSP)*, 2015, pp. 1–6. doi: 10.1109/MLSP.2015.7324337.

[8] D. Stowell, D. Giannoulis, E. Benetos, M. Lagrange, and M. D. Plumbley, "Detection and Classification of Acoustic Scenes and Events," *IEEE Transactions on Multimedia*, vol. 17, no. 10, pp. 1733–1746, 2015.

[9] S. Sigtia, A. M. Stark, S. Krstulović, and M. D. Plumbley, "Automatic Environmental Sound Recognition: Performance Versus Computational Cost," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 24, no. 11, pp. 2096–2107, 2016.

[10] M. Scarpiniti, D. Comminiello, A. Uncini, and Y.-C. Lee, "Deep Recurrent Neural Networks for Audio Classification in Construction Sites," in *2020 28th European Signal Processing Conference (EUSIPCO)*, 2021, pp. 810–814.

[11] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," in *Advances in Neural Information Processing Systems*, 2012, vol. 25.

[12] P. Y. Simard, D. Steinkraus, and J. C. Platt, "Best practices for convolutional neural networks applied to visual document analysis," in *Seventh International Conference on Document Analysis and Recognition, 2003. Proceedings.*, 2003, pp. 958–963.

[13] B. McFee, E. J. Humphrey, and J. P. Bello, "A software framework for musical data augmentation," in *Proceedings of the 16th International Society for Music Information Retrieval Conference, ISMIR 2015*, 2015, pp. 248–254.

[14] A. Graves, Supervised Sequence Labeling with Recurrent Neural Networks, vol. 12, no. 1. 2013.

[15] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, 2014.

[16] S. Dhingra, G. Nijhawan, and P. Pandit, "Isolated speech recognition using MFCC and DTW," *International Journal of Advanced …*, vol. 2, no. 8, 2013.

[17] C. Ittichaichareon, "Speech recognition using MFCC," *… Conference on Computer*, 2012.

[18] A. Bala, "Voice Command Recognition System Based on Mfcc and Dtw," *International Journal of Engineering Science and Technology*, vol. 2 (12), no. 9, 2010.

[19] Gordon E. Carlson, Signal and Linear System Analysis, 2nd Edn., 1998 New York, Chichester, John Wiley & Sons, *European Journal of Engineering Education*, vol. 23, no. 3, 1998.

[20] J. Salamon and J. P. Bello, "Deep Convolutional Neural Networks and Data Augmentation for Environmental Sound Classification," Aug. 2016.