Speaker Diarization: A Comparative Analysis using different types of Deep Neural Networks

Asmita Pal, Pramesh Pandey, Manish Meshram, Abhinav Pandey

ABSTRACT

Speaker diarization is the process of annotating an input audio channel with information that attributes tomporal regions of energy to their specific sources. Deep learning, has exhibited prolific improvements in pattern recognition and related fields which involves *learning* systems. In this project, we exploit the inevitable learning attribute of neuron-based algorithms, to perform speaker diarization. We present a comparative analysis of different types of present day state-of-the-art neural networks, (A) Multi-layer Deep Neural Network, (B) Convolutional Neural Network. The essence of this study brings out the prediction accuracy in networks with marked difference in training behavior, in the framework of speaker diarization task.

1. INTRODUCTION

Speaker diarization has been an extensive area of research ever since the need arised for human-machine interaction for meetings, group conversations and any arena which involves rapid identification of speakers. Fox et al. proposed a hierarchical Dirichlet process hidden Markov model (HDP-HMM) to identify the number of speakers on the NIST speaker diarization database [1]. Speaker diarization can be defined as the process of separating an input audio signal into homogeneous segments according to the speaker identity. It is essentially not speech recognition, but a enhancement on the same. It combines the principles of speaker segmentation and speaker clustering, through identification of speaker change points in an audio signal followed by grouping of similar speech characteristics.

Additionally, deep learning has emerged as a promising arena to solve problems where even human experts are left befuddled. It attempts to mimic the activity co-ordinated by the layers of neurons in the neocortex. The key idea of exploiting an array of neurons in the artificial "neural network" has been there since the 1970s, but has remained fairly unexplored due to the limitations of computational capacity. With the advent of multicore systems and GPUs, neural networks have gained considerable popularity and eventually we are progressing into a deep learning era. Deep neural networks have shown remarkable advances in speech and image recognition. Salakhutdinov and Hinton explored the behavior of a multi-layered feedforward neural network by pre-training one layer at a time, treating each layer in turn as an unsupervised restricted Boltzmann machine, then fine-tuning it using supervised backpropagation [6]. Through decades, Gaussian Mixture Model (GMM) was used for acoustic modelling and was considered state of the art in speech recognition. However, DNNs revolutionized this field by employing multi-layer networks and using *discriminative training.* Recently, Google's voice transcription system took the advantage of the successes demonstrated by Long Short Term Memory Recurrent Neural Networks (LSTM RNNs), in their new Android Speech recognizer. Graves et al. showed a remarkable improvement by using RNNs for end-to-end learning for acoustic models [4].

As speaker diarization involves idenfying and separating the speakers based on their features, deep neural networks seem like a viable option. The multi-class problem often used with DNNs is analogous to the multiple speaker identification using specific features. Contrarily, Hidden Markov Models have been used consistently to model each of the speakers and there already exists a large variety of open source repositories for this problem. A tandem HMM-LSTM system was proposed to improve overlap detection in speech [3]. Further, Romero et al. employs DNNs instead of conventional i-vector extraction and demonstrates better performance than the latter [2]. Another interesting approach has used spectrograms as input to a Convolutional Neural Network (CNN) for speaker identification and clustering [5]. Hence, in this project we aim to exploit the learning ability of neural networks for speaker diarization.

Our contributions in this project are as follows:

- We explore the characteristics of speech data in order to feed it to a neural network. Following which, we extract some features from the speech to make it functional for a deep neural network.
- We test the speaker diarization problem with different architectures such as the deep neural network and the convolutional neural network.
- Our project makes a comparative analysis of performance evaluation and reveals the adaptive prowess of DNNs for the same two-speaker problem.

2. SPEAKER DIARIZATION DESIGN

In this section, we explore the basic design of three different approaches to the speaker diarization problem. We model our speech data accordingly to be fed to each Neural Network. We use a multi-layer deep neural network and a convolutional deep neural network to evaluate the accuracy achieved for a given data. Figure 1 depicts the control flow architecture that we adopted to accomplish the task of Speaker Diarization.

2.1 Task 1: DNN Approach

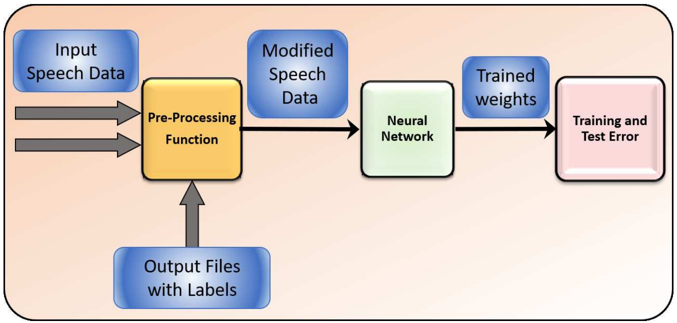

Figure 1: *Control Flow for Speaker Diarization.*

As mentioned earlier, DNN is a viable approach for the speaker diarization problem. Figure 2 presents an overview of a multi-layer deep neural net. In this context, we would like to slightly outline how the speaker diarization problem works in DNNs. Classification using neural network is a supervised learning method and therefor requires a tagged dataset. DNN comprises of a set of interconnected layers, in which the inputs lead to outputs by a series of weighted edges and nodes. The weights on the edges are learned when training the neural network on the input data. The direction of the graph from the inputs through the hidden layer, with all nodes of the graph connected by the weighted edges to nodes in the next layer. To compute the output of the network for any given input, a value is calculated for each node in the hidden layers and in the output layer. For each node, the value is set by calculating the weighted sum of the values of the nodes in the previous layer and applying an activation function to that weighted sum. This is the feed-forward part of the DNN. With the outputs obtained from the feed-forward network, we calculate the error, given a set of pre-defined labels. This error is then back-propagated to modify the weights at neurons. The process continues for a certain number of iteration until and unless we are satisfied with the modfied weights. These modified weights are used in a feed-forward way with a different set of data, often referred to as the test data, to evaluate how well we have trained our DNN.

In this approach, the initial sound file is in the format .wav and we process this to separate the data into channels. Each channel is then transformed into a real-valued array of data,

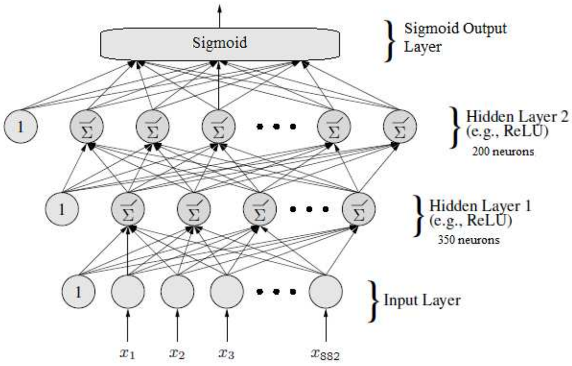

Figure 2: *DNN architecture.*

instead of the raw sound. Each value in this array of data represents the sound characteristics in a certain window. The numerical details of the pre-processing are presented in Section 3.1. As with the MNIST data set or any classification problem, we feed this processed data in the DNN. It must be noted, that the DNN is trained for both the channels, since we consider a two speaker classification problem. Our neural net trained using this method, shows considerable improvement in training accuracy, which justifies the efficacy of our approach.

2.2 Task 2: DNN Approach 2

FFT on an audio series leads us to the general Time-Frequency spectrum of the audio. It has been found out that all the frequency elements are not clearly discernable by the human ears. Mel-Frequency (MF) analysis of speech is based on human perception experiments. It is observed that human ear acts as filter - It concentrates on only certain frequency components. These filters are non-uniformly spaced on the frequency axis. The case is that we have more filters in the low frequency regions and less no. of filters in high frequency regions. Mel-frequency scale represents subjective (perceived) pitch. It is one of the perceptually motivated frequency scales.

Hence, MF coefficients model the spectral energy distribution in a perceptually meaningful way. Mel scale is the most widely-used acoustic feature for speech recognition, speaker recognition, and audio classification. It has also been used for Speaker Diarization in [5]. It takes into account certain properties of the human auditory system. This work of Mel Frequency scaling was based on extensive experiments on human subjects and has been widely accepted since then [5].

Mel-frequency scale represents subjective (perceived) pitch. It is one of the perceptually motivated frequency scales. The visual representation of the Mel-scale with respect to frequency scale is given in the Figure 3.

$$f_{Mel} = 2595 \log_{10}(1 + \frac{f_{Hz}}{700})$$

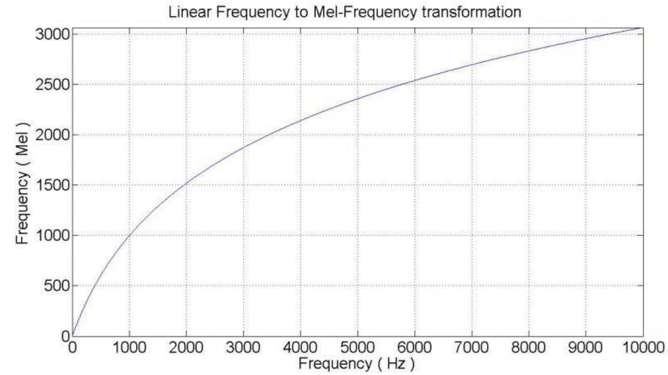

Figure 3: *Mel-scale visual representation.*

Due to this wide acceptance of the scale in speech recognition community, we thought to process out input to achieve a Mel-Spectrogram and feed it to a neural network, in expectation that neural network would have more practical information to process on.

2.3 Task 3: CNN Approach

Convolutional Neural Networks are a variant of deep learning approaches that are designed to take advantage of the 2D structure of an input image. It consists of initial sparsely connected layers followed by one or more fully connected layers, as in a standard multi-layer neural network. The convolution layer has some kernels which is smaller than the dimension of the input image and a certain number of channels less than equal to the number of channels in the input image. The size of the kernels give rise to a locally connected structure which are each convolved to produce feature maps. Each map is then subsampled with mean or max pooling. CNNs function extremely well with images because of its ability to allow specification of channels and modifiable kernels. CNNs extend the well-known idea from image processing of filters as weighted sums of pixels, by making the filter coefficients learnable.

3. METHODOLOGY

In this section, we discuss the methodology parameters for each of our approaches to analyze their efficacy.

3.1 Task 1: Parameters for 1st DNN Approach

Property	Parameters
No. of Layers	2
No. of neurons in 1st Hidden Layer	350
No. of neurons in 2nd Hidden Layer	100
Activation Function for Hidden Layers	ReLU
Activation Function for Output Layer	Sigmoid
Learning Rate	0.01
No. of epochs	20
Batch size	100

Table 1: *DNN configuration*

Table 1 enlists the numerical values for defining each layer in our multi-layer neural network. We use inbuilt functions in Tensorflow to model our neural network. The use of ReLU as an activation function for the hidden layers, can be attributed to the fact that it has a reduced likelihood of vanishing gradient. Hence, ReLU stops the inactive neurons. In order to identify the speech and non-speech character of a given data point we use sigmoid as the output function. Evidently, it is a binary classification, and sigmoid gives us the probabilities of each class (namely, speech and non-speech). Mean square error is used to estimate the error in our predictions, as it incorporates the variance as well as the bias. The sampling rate as observed in the given *.wav* files is 44100 samples/sec. For pre-processing the initial sound file we have used the pysoundfile package and extracted data every 20 milliseconds. This way each data vector represents 20 milliseconds. For the given output file, we have split the labels, each to represent the 20 millisecond window. This pre-processing is needed for proper mapping of training vectors and the corresponding class labels. We have trained the DNN on 30 samples and tested it for 7 samples of audio input stream. In the upcoming section we describe the prediction accuracy of our DNN.

3.2 Task 2: Parameters for 2nd DNN Approach

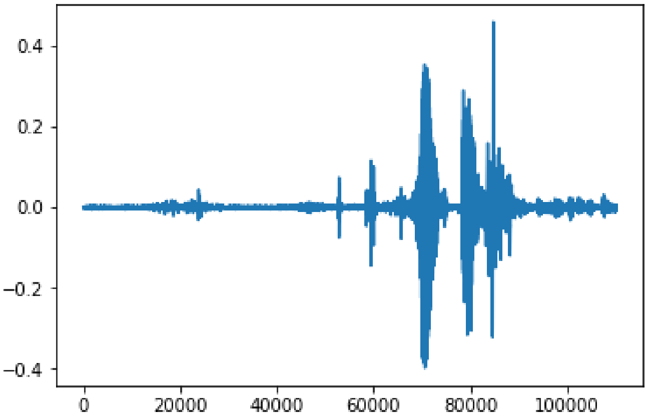

Figure 4: *Audio time series plot.*

The architecture for this approach is same as in 3.1, except the inputs to the DNN adopts a spectral approach. We used python package ‘librosa’, which is a comprehensive package for extensive audio processing. We have 16kHz sampling rate, 1024 samples FFT window length and 160 samples as hop length. For Mel-Spectrogram, we used 128 elements in the Mel-scale. We obtained the following plots for the audio file ‘HS_D04.wav’. Figure 4 is the audio time series plot.

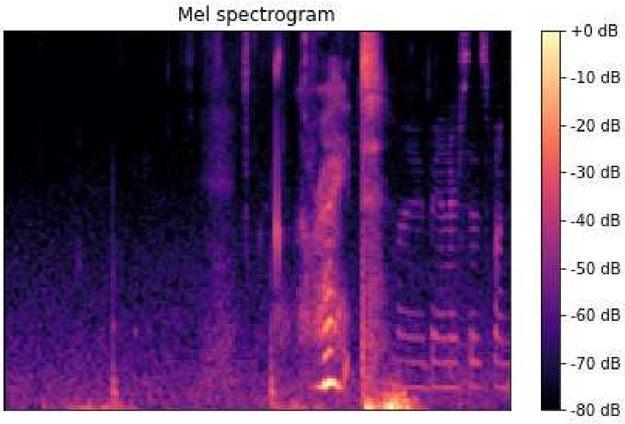

Figure 5: *Mel-spectrogram for 5s.*

Figure 5 is the Mel-spectrogram for 5s of the audio stream. Figure 6 is for 100ms of the stream. After extensive discussion in class, we had decided 20ms to be the period for one data

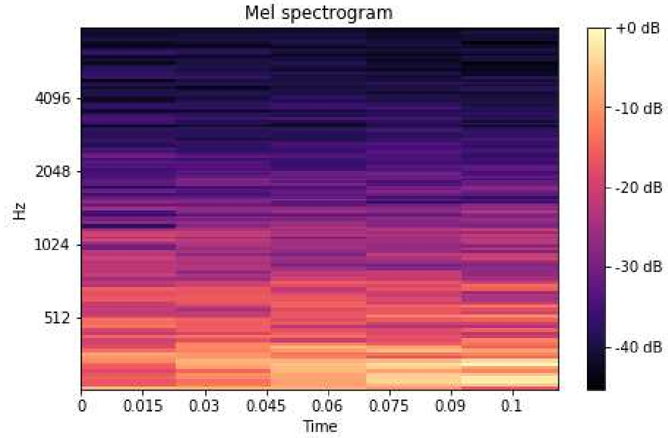

Figure 6: *Mel spectrogram for 100ms.*

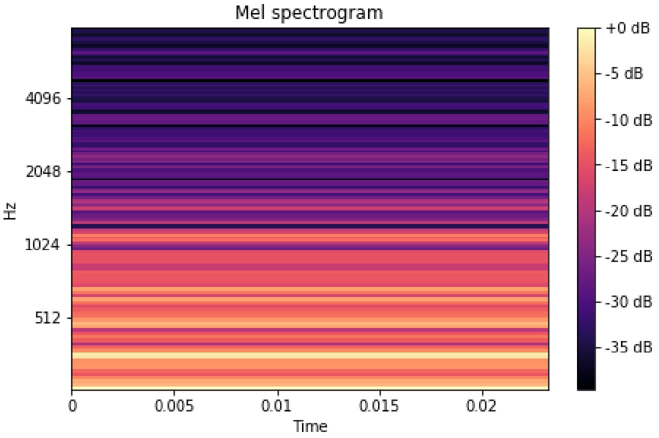

Figure 7: *Mel spectrogram for 20ms.*

point. So, when we tried to plot the Mel-spectrogram for 20ms, we obtained the plot of Figure 7. The figure clearly shows that the units of decibels(dB) for all 128 mel scales for entire 20ms is the same. Hence, the 2D image for 20ms, just represented 1D data, that values of dB at 128 mel scales. Rechecking the figure for 100ms, we see that it is the composition of about 5 smaller images representing 20ms each. This 1D nature of the data can be exploited to feed into a DNN. In the next section, we assess the results when this Mel-spectrogram is passed as an input to a DNN and why it does not stand as a viable option.

3.3 Task 2: Parameters for CNN Approach

Table 2 defines the convolutional neural network. We use inbuilt functions in Tensorflow to model our neural network. The packages used for pre-processing are slightly different from those used in Task 1 and Task 2. The use of ReLU function has the same logical inference as in DNN. Tensorflow is used for modelling the CNN architecture.

Property	Parameters
No. of Layers	3
1st Layer	Convolution Layer
2nd Layer	ReLU Activation Function
3rd Layer	Max Pooling

Table 2: *DNN configuration*

4. EXPERIMENTAL RESULTS

In this section, we demonstrate a comprehensive analysis of speaker diarization using DNN and CNN. With the methodology followed in 3, we have generated the output file for the test data similar to the given output file used in training. Figure 8 shows the gradual drop in training error rate. The initial decrease is steep during training and stabilizes after a few epochs. The mean square error for test data is initially 13.08% and drops down to 11.12%. This performance is achieved using the first DNN approach. For the second DNN approach, the error rate achieved is 35% for the Mel-spectrogram generated every 20ms. This may be attributed to the fact that the expected output we are feeding for backpropagation, reflects more of amplitude changes, which is inherently adopted by the humans while identifying speakers, in a specific channel. The spectral approach is essentially not adopted for classififcation, for speech input from multiple channels. In our study, the DNN stands out with a marked improvement in performance and an accuracy of 88.88%, since its input is an audio series reflecting amplitude changes with time. This confirms our faith in deep neural networks and their ability to learn.

5. CONCLUSION

Which approach is better?

6. REFERENCES

[1] Fox, E. B. and others An HDP-HMM for systems with state persistence. vol. 307, pp. 312–319.

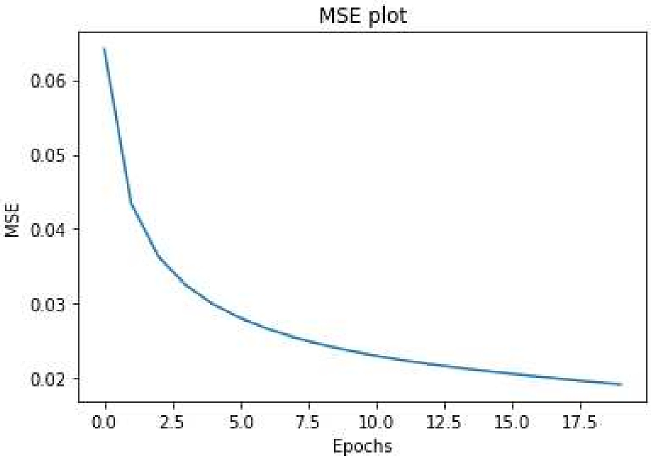

Figure 8: *Traininf Error (MSE) using DNN.*

[2] Garcia-Romero, D. and others Speaker diarization using deep neural network embeddings. pp. 4930–4934.

[3] Geiger, J. T. and others Detecting overlapping speech with long short-term memory recurrent neural networks. pp. 1668–1672.

[4] Graves, A. and others Speech recognition with deep recurrent neural networks. pp. 6645–6649.

[5] Lukic, Y. and others Speaker identification and clustering using convolutional neural networks. pp. 1–6.

[6] Salakhutdinov, R. and others Restricted Boltzmann machines for collaborative filtering. pp. 791–798.