Speaker Identification using Feed Forward Networks

Mohit Agarwal

Abstract:

In the present day, it is self-evident that people can be recognized by their sounds. This research investigates a feedforward neural model-based approach for speaker recognition. First, the study employs the Gaussian Mixture Model to detect voice activity in this investigation. Then after computing the spectrogram for the audio signal, the spectrogram is subjected to two more procedures before silence is discarded. Next, a Melodic-scale (Mel-scale) filter is used to minimize dimensionality, and then the delta feature is added to it. Finally, a multi-layer feed-forward network is used to train the feature vector for a sequence of speech frames from various speakers, each with a speaker label. This experiment found that the system correctly identifies around 90% of the speakers, on test samples.

Introduction:

The practice of recognizing speakers from their voices is known as speaker recognition or identification. The speaker recognition problem, which refers to the general domain of recognizing speakers based on their sounds, can be separated into two subproblems: feature extraction or speech processing and speaker identification. Speech is handled on a frame-by-frame basis in speech processing usually only with the concern that the frame is either speech or silence and then we extract the features only for the speech frames using mel-filter bank and delta feature. Speaker identification is the problem of deciding who is speaking in each audio file.

This paper describes a feed forward network-based speaker recognition system in which a multi-layered neural network model is constructed for each speaker. The neural network models are trained using feature vectors obtained after speech processing. For testing the model, for each session, we read in all the data for that session and predict all frames. Then we fuse the scores by summing the log probabilities and decide class for that session.

Methods:

Dataset Used: The system was trained and tested on the The King corpus [1] dataset. The data consists of 51 Speakers (numbered 01 to 60, with some gaps in the sequence). All speakers in the data set were male. There were ten sessions for each speaker (numbered 01 to 10), and each session was recorded in both a wide-band (wb) and a narrow-band (nb) channel.

Recognition of speech and silence in each frame of the audio files: The speech and silence frames were recognized by training and testing the audio dataset on GMM (Gaussian Mixture Model) which separates the frames based on their RMS energy measurements i.e., Hz.

Computing Spectrogram on each audio files: A spectrogram of each audio file was computed using STFT[3] function of *librosa* module. A Hamming window was used, with 20 ms frame lengths and 10 ms frame advances. A Mel filter bank with n_mels=15 (number of mel bands to generate) was applied on each spectrogram to reduce the dimensionality. Δ features were used on the spectrograms using *librosa.feature.delta* function. A numpy stack operator a was used to join the original Mel-spectrogram with the Δ features. Feature vector for speech frames were retained and feature vector for silence were discarded.

Model training for speaker Identification: A multilayer feed-forward neural networks (with 750 nodes in the hidden layers, ReLU activation functions, batch normalization before each hidden layer, and an L2 regularization with a setting of 0.01) was trained on subset of the data obtained after method 2.

Experiments:

Experiments were conducted by varying number of parameters to check for performance variation. First, width of the network was varied by changing the number of nodes in each hidden layer from 750 to 1000 keeping the depth constant (n=4). Second, depth of the network was varied by changing the number of hidden layers from 3 to 4 keeping the number of nodes (n=750). Third, effect of regularization was studied by varying the *kernel_regularizer* parameter from L2 to L1. Two cases were studied to study the effect of batch normalization. In the first case, batch normalization was performed before each hidden layer and in the second case normalization was not performed.

Results:

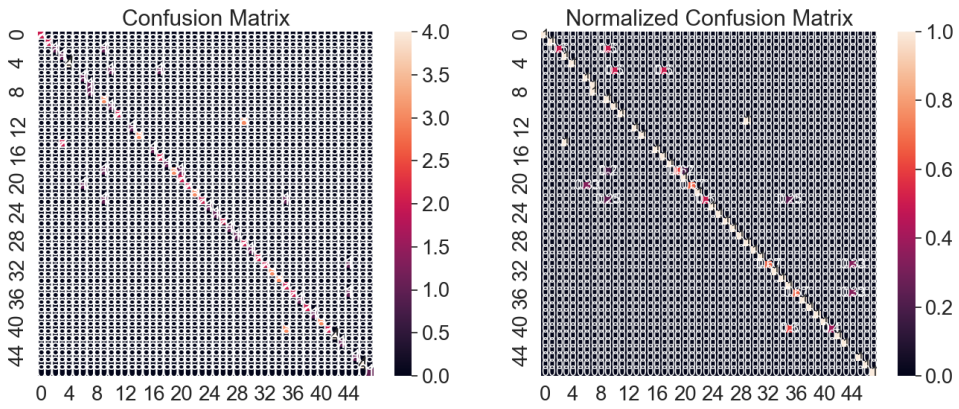
Firstly, an experiment using a 4-hidden layer network with 750 nodes in the hidden layers, ReLU activation functions, batch normalization before each hidden layer, and an L2 regularization with a setting of 0.01 was conducted. This experiment resulted in an accuracy of around 85% on the test dataset.

Secondly, the depth of model from 4 hidden layers to 3 hidden layers was varied, keeping other parameters same as mentioned in first case. The results of the experiment showed that the network doesn't perform well. Decreasing the depth resulted in the decrease of the accuracy by around 10% on test dataset.

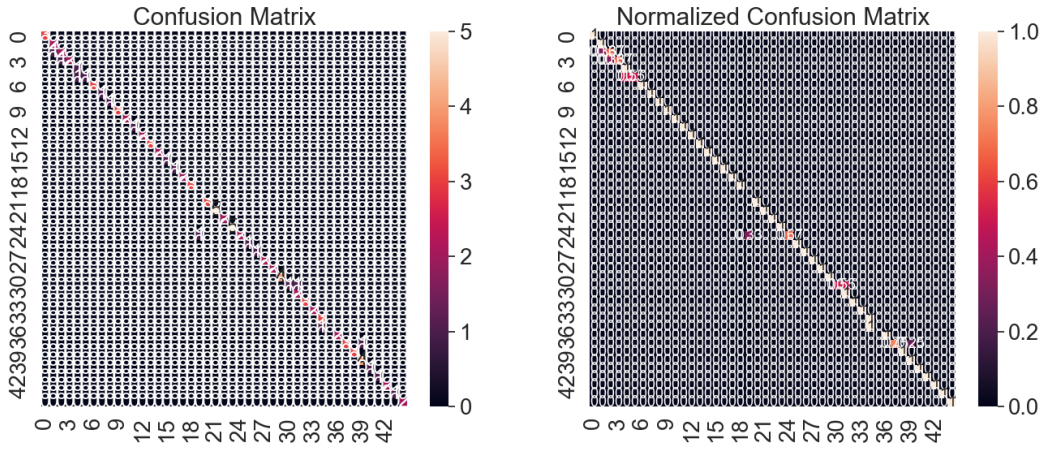
To study the affect of varying the width of the model, the number of nodes in each hidden layer were changed from 750 to 1000, keeping other parameters same as mentioned in first case. The results from this experiment variation depicted that the network performed well and resulted in an increase around 5% in accuracy in the test dataset.

Next, effect of different types of *kernel_regularizer* parameter was studied by changing the parameter from L2 to L1 keeping other parameters same as mentioned in first case. A significant drop in the accuracy by around 75% was observed on test dataset.

By not using the batch normalization before each hidden layer in first case, the accuracy of the test dataset dropped to approximately 40%.



`Figure 1` Confusion matrix for the results obtained on test dataset with the following model parameters, 3-hidden layer network with 750 nodes in the hidden layers, ReLU activation functions, batch normalization before each hidden layer, and an L2 regularization with a setting of 0.01. Gained accuracy of around 77%



`Figure 2` Confusion matrix for the results obtained on test dataset with the following model parameters, 4-hidden layer network with 1000 nodes in the hidden layers, ReLU activation functions, batch normalization before each hidden layer, and an L2 regularization with a setting of 0.01. Gained accuracy of around 90%

Discussion:

Firstly, by decreasing the depth of the network and keeping other parameters same resulted in the decrease of the accuracy. This could be due to decrease capacity of the thinned network.

Secondly, by increasing the width of network and keeping other parameters constant resulted in the increase of the accuracy. This could have happened because the capacity of the network increased as the nodes are increased.

Thirdly, applying batch normalization in each hidden layer improved the accuracy of the model. This might happen because we standardize the inputs to each hidden layer for each mini batch.

After analyzing the results, it was found that by performing batch normalization on data before each hidden layer, using L2 regularizer while keeping the width of the network as 750 nodes and depth as 4 hidden layers, maximum accuracy was achieved. It could suggest that using these parameters, higher accuracy could be reached in comparison to all the other parameter settings used in the study.

References:

1) Higgins, A. and Vermilyea, D. (1995). King speech corpus documentation: Linguistic Data Consortium.
2) Goodfellow, I., Bengio, Y. and Courville, A. (2016). Deep learning. Cambridge, Massachusetts: The MIT Press.
3) http://librosa.org/doc/main/generated/librosa.stft.html