CS753 Project - Speaker Verification

Rahul Chunduru 160050072

Suraj Narra 160050087

Sunchu Rohit 160050097

1 Introduction

Speaker verification is the task of verifying the speaker of an utterance in the known database. It is different from speaker recognition as it can choose to output no speaker found. Although, this capability is not robust enough to be used as the sole mode of identity verification, it is still useful for certain purposes such as:

1. **Bio-metric Security**: Speech verification can be combined with other methods for bio-metric identification.

2. **Voice Assistants**: Speech verification offers a good level of security for domestic and personal touch-free voice assistants.

In this project, we have implemented and analysed the effectiveness of a neural network based speaker verification system.

2 Problem Statement

We consider the mechanism in which the speakers enroll themselves first in our system and then use them gets verified.

Therefore, the problem can be stated as, Given voice samples of enrolled speakers and a sample claiming to be one of them (the purported identity is provided), verify the claim. Note that this is distinct from Speaker Identification, where you don't know what the voice sample's claimed identity is.

3 Related Work

Before the advent of Neural Nets, the task of speaker verification is classically approached using Adapted Gaussian Models. In this approach, the speakers are assumed to be coming from the UBM model and the hypothesis regarding a particular speaker is tested using Likelihood tests. The likelihood of an utterance given a speaker is modelled using GMMs.

4 Outline of the Project

Our project starts with the paper by Wan et al (2017). They use neural networks to create embeddings of each speaker, and given a test utterance, compare it with the claimed speaker's embedding. If the similarity (cosine similarity) lies within a certain threshold, they declare the utterance accepted. They train the model in small batches, with a generalized loss function GE2E (described below). They use LSTMs to extract relevant features from utterances.

The metric used is Equal Error Rate - vary the threshold until $FAR = FRR$, then $EER = (FAR + FRR)/2$. Here FAR is the False Acceptance Rate, and FRR the False Rejection Rate.

Our work in this project is a comparison of the following modifications to this system:

1. Use convolutional neural networks instead of LSTMs to extract features from utterances

2. Use attention on top of LSTMs

3. Try large-margin softmax (described below) error instead of GE2E.

We report and discuss our results after describing the architecture and experimental settings.

5 Architectures

We used the following architectures in our implementations.

5.1 Linear Attention Network

We first use LSTMs on each utterance $x_i = \{x_{i1}, x_{i2}, .., x_{iT}\}$ to extract features $h_i = \{h_{i1}, .., h_{iT}\}$. Then we compute an attention scores on h_i as

$$\alpha_t = \frac{e^{Wh_{it}}}{\sum_{j=0}^{T} e^{Wh_{ij}}}$$

where W is a feedforward Attention network.

Finally, we obtain embedding ω_i for the utterance as,

$$\omega_i = \sum_{t=0}^{T} \alpha_t h_{it}$$

5.2 Convolution Networks

The CNN architecture consists of two Convolutional Layers with stride and zero padding. We used ReLU as the activation function. The output is finally fed to a fully connected layer which represents the embeddings of the speaker's utterance.

6 Loss Functions

6.1 Generalized End-to-End Loss

The Generalized End-to-End Loss proposed by Wan et al (2017). In a batch of N speakers each with M utterances, calculate cosine distance from each utterance's embedding vector to the centroid of each speaker, and compute a loss function penalizing proximity to wrong centroids and rewarding proximity to the right centroid.

There are two ways to go about this. In the current batch, let $\mathbf{e}_{ji}$ be the embedding vector of the jth speaker's ith utterance, and $S_{ji,j}$ its cosine similarity with the jth speaker's centroid.

1. Softmax loss:

$$L(e_{ji}) = -\mathbf{S}_{ji,j} + log \sum_{k=1}^{N} exp(\mathbf{S}_{jk,k})$$

2. Contrast loss:

$$L(e_{ji}) = 1 - \sigma(\mathbf{S}_{ji,j}) + \underset{1 \leq k \leq N, k \neq j}{max} \sigma(\mathbf{S}_{ji,k})$$

6.2 Softmax with large margin

This is a modification on the popular softmax loss with the cosine distance replaced by

$$\phi(\theta) = cos(n_1\theta + n_2) + n_3$$

Since θ typically has doesn't usually span its entire possible range, using softmax with large margin can result in better performance.

Although we implemented this loss function, we ran into some trouble while training, so we cannot report the results.

7 Implementation Details

We use the TIMIT dataset, with 900 speakers. We first extract mel-spectrogram using **librosa**. Then batches of 4 speakers each with 5 utterances are fed into the networks described above, randomly sampled in each epoch. This takes around 2 minutes to train. Testing is done on a set of 63 speakers, and the metric is EER (Equal Error Rate).

8 Results and Analysis

The following table contains EER (Equal Error rate) values of various architectures when trained using GE2E loss function:

LSTM	LSTM+Attention	CNN
0.0572	0.0426	0.0097

As expected, attention shows a slight improvement over the baseline. The most notable detail of our results is that convolutional feature extraction outperforms all others. It is also significantly faster to train. Wan et al (2018) report an EER of 0.03, but this is on a different, larger dataset than ours. We are yet to test our model on this dataset.

9 Scope for Future Work

Our work can be extended in the following ways. The system can be made noise resilient. This has be achieved using Attention Networks. The current model requires huge amounts of enrolment data for verification. This can be reduced via **Transfer Learning**.

10 Conclusion

In this work, we looked into deep neural net techniques for the task of Speaker verification. We compared the performance of model across different architectures and loss functions. We noted that the common goal for all these architectures is to compute efficient embeddings for the speaker utterances, such that in the projected dimension, the speaker verification problem becomes simple.

References

[Wan2018] Wan. *Generalized end-to-end loss for speaker verification*. 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)

[Rey1995] Reynolds, D. A., Speaker identification and verification using Gaussian mixture speaker models, Speech Commun. 17 (1995), 91–108.

Github link to code of Wan et al