DEEP NEURAL NETWORK BASED DISCRIMINATIVE TRAINING FOR I-VECTOR/PLDA SPEAKER VERIFICATION

Zheng Tieran, Han Jiqing, Zheng Guibin

School of Computer Science and Technology, Harbin Institute of Technology, Harbin, China
{zhengtieran,jqhan,zhengguibin}@hit.edu.cn

ABSTRACT

In the studies of i-vector based speaker verification, the discriminative training of probabilistic linear discriminative analysis (PLDA) model has been proven to be an effective way to improve performance. This paper focuses on using a deep neural network (DNN) to strengthen the original discriminatively trained classifiers by its strong capability of nonlinear modeling representation. We first propose a deep neural network based dimensionality reduction model to replace the linear discriminant analysis (LDA) process, and then a discriminative training algorithm is also proposed to jointly optimize the network and PLDA scoring function under single discriminative criterion. Our experiments show that performance improvements are achieved in the male trials of short2-short3 core data set of NIST SRE08.

Index Terms— DNN, Discriminative training, PLDA, Speaker verification

1. INTRODUCTION

Over the last decade, some methods based on i-vector extraction and PLDA have been widely used in state-of-the-art speaker verification systems [1, 2]. I-vector is an information-rich low-dimensional fixed length vector extracted from the feature sequence representing a speech segment. A PLDA verification score between two i-vectors is then approximated by the Log-Likelihood Ratio (LLR) between the "same-speaker" and "different-speaker" hypotheses.

PLDA attempts to decompose speech data into a speaker component and a channel component, and assumes that they obey the Gauss distribution, thus the PLDA model can be optimized by generative training under the maximum likelihood (ML) criterion. However, such prior Gaussian assumptions have been proved inaccurate [1]. For this, some discriminatively trained affine transformations of the scores [3, 4] were firstly proposed to address the problem of inaccurate verification score, which is the result of the inaccurate assumptions. Subsequently, the discriminative training of PLDA model were shown to outperform the ML training in speaker verification [5-8]. Their proposed discriminative training scheme optimize the LLR score function of the PLDA model directly, instead of training the PLDA model explicitly. Those optimization processes allow the score function to be more general than the score function of a standard (ML trained) PLDA model.

In state-of-the-art speaker verification systems, before the PLDA scoring stage, i-vectors are typically post-processed to generate dimensionality reduced and channel-compensated features, in order to annihilate the directions not informative for speaker and to improve the computational efficiency of PLDA. Nowadays LDA is widely used for this purpose. However, LDA is also limited by its Gaussian assumption. Especially when speech recordings are collected in the presence of noise and channel distortions, the inaccurate assumption can be more problematic [9, 10]. Some studies that attempt to alleviate the limitations of LDA have been reported to bring performance improvements [9, 11, 12].

Despite the application of DNN are very successful in automatic speech recognition (ASR) field, a direct transition to speaker recognition is much more challenging, as speakers are often unknown during system training or each speaker only has very little training data. Recently, in DNN based speaker verification approaches, feature representation instead of classifier has become the main research focus. They can be roughly divided into two categories, one is to use the neural networks to assist in the i-vector extraction [13, 14], another is to directly learn embedding speaker feature [15-17].

In this paper, we propose a DNN based dimensionality reduction model to replace the LDA, and then optimize the DNN and the PLDA scoring function under single discriminative criterion by our proposed discriminative training algorithm. Consequently, the DNN can be directly embedded into the scoring function and be used to strengthen the original linear discriminatively trained classifiers by its strong capability of nonlinear modeling representation. Moreover, a more general classifier can be achieved by alleviating the limitations of Gaussian assumption in both dimensionality reduction and scoring stages.

2. PROPOSED SPEAKER VERIFICATION SYSTEM

The i-vector/PLDA approach has become state of the art in speaker verification field. Generally, it contains three processing stages: i-vector extraction, dimensionality reduction and scoring. In this paper, our speaker verification system also follows this framework and a DNN based dimensionality reduction is introduced into the second stage.

2.1. I-vector extraction [2]

It is assumed that a high-dimensional GMM supervector $\mathbf{v}$ corresponding to a speech utterance can be modeled as:

$$\mathbf{v}=\mathbf{u}+\mathbf{T}\mathbf{x} \tag{1}$$

where $\mathbf{x}$ is a low-dimensional random feature vector known as the i-vector, $\mathbf{T}$ is a matrix of a low rank referred as the total variability

matrix, **u** is the mean of **v**. It is assumed that **x** follows a standard Gaussian distribution and its dimension is *d*. To learn the bases for the total variability subspace, Baum-Welch statistics are computed from a Universal Background Model (UBM).

2.2. DNN based dimensionality reduction

The i-vector approach models both signal (i.e. speaker) and noise （i.e. channel, session, etc.）variabilites in the same total variability subspace, therefore an intersession compensation should be adopted to reduce dimensionality and to annihilate the undesired noisy directions. However, we do not use the widely adopted Fisher LDA compensation method in this stage. There are two reason for doing this. Firstly, the Fisher criterion based optimization which attempts to maximize the between-class scatter while minimizing the within-class variation is abandoned here, because we want to integrate the dimensionality reduction stage and the subsequent scoring stage to obtain a more discriminative classifier by using the discriminative training method presented in section 3. Secondly, nonlinear projection is also considered to have more powerful ability in seeking a reasonable low-dimensional feature subspace than the linear projection conducted by LDA.

Our DNN based dimensionality reduction, shown as Fig.1, can be viewed as a nonlinear projection $\mathbf{f}:\Re^d \mapsto \Re^n$, which takes i-vector feature **x** as input, and the dimension *n* of output vector $\mathbf{f}(\mathbf{x},\Theta)$ is much smaller than *d*, Θ is the set of network parameters. Rectified linear units (ReLUs) are adopted as activations in the hidden layers. Without the activation function, only linear summation is used in the output layer, hence negative value can be available in the output vector. Note that if there is 0 hidden layer, the network will degenerate into a linear projection. The network parameters Θ will be jointly optimized with the PLDA model by our discriminative training method. Before the PLDA scoring stage, mean subtraction, length normalization and whitening are adopted as a pre-processing:

$$\mathbf{z} = \mathbf{M}\frac{\mathbf{f}(\mathbf{x},\Theta)-\boldsymbol{\mu}_f}{\left\|\mathbf{f}(\mathbf{x},\Theta)-\boldsymbol{\mu}_f\right\|_2} \tag{2}$$

where **M** is the whitening transformation matrix and $\boldsymbol{\mu}_f$ is the mean of **f**(**x**).

2.3. PLDA scoring

Kenny [1] has given a modified PLDA model, it decomposes total variability into between-class (speaker) and within-class (channel) variability as follow:

$$\mathbf{z}=\mathbf{m}+\mathbf{U}_1\mathbf{y}_1+\mathbf{U}_2\mathbf{y}_2 \tag{3}$$

where $\mathbf{y}_1$ and $\mathbf{y}_2$ are random vectors depending, respectively, on the speaker and the channel. Speaker variability is given by $\mathbf{U}_1$ and channel variability is given by $\mathbf{U}_2$. In our case, **m** is a zero vector after the pre-processing.

For scoring two vectors $\mathbf{z}_i$ and $\mathbf{z}_j$, an LLR s_{ij} should be calculated between the hypothesis of being from the same speaker and the hypothesis of being from the different speakers. A closed-form expression of the LLR was also given by [8]

$$s_{ij} = \mathbf{z}_i^T\mathbf{P}\mathbf{z}_j + \mathbf{z}_j^T\mathbf{P}\mathbf{z}_i + \mathbf{z}_i^T\mathbf{Q}\mathbf{z}_i + \mathbf{z}_j^T\mathbf{Q}\mathbf{z}_j + (\mathbf{z}_i+\mathbf{z}_j)^T\mathbf{c} + k \tag{4}$$

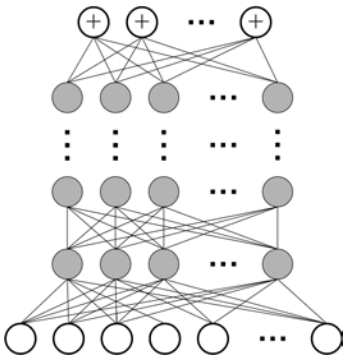

Fig.1 Architecture of the DNN used in dimensionality reduction

where

$$\mathbf{P} = \frac{1}{2}\Sigma_{tot}^{-1}\Sigma_b(\Sigma_{tot}-\Sigma_b\Sigma_{tot}^{-1}\Sigma_b)^{-1}$$

$$\mathbf{Q} = \frac{1}{2}\Sigma_{tot}^{-1} - (\Sigma_{tot}-\Sigma_b\Sigma_{tot}^{-1}\Sigma_b)^{-1}$$

$$\mathbf{c}=-2(\mathbf{P}+\mathbf{Q})\mathbf{m}$$

$$k = \frac{1}{2}(\log\left|\Sigma_{tot}\right| - \log(\left|\Sigma_{tot}-\Sigma_b\Sigma_{tot}^{-1}\Sigma_b\right|) + \mathbf{m}^T 2(\mathbf{P}+\mathbf{Q})\mathbf{m}$$

where $\Sigma_b = \mathbf{U}_1\mathbf{U}_1^T$, $\Sigma_w = \mathbf{U}_2\mathbf{U}_2^T$ are between- and within-class covariance matrices, respectively. $\Sigma_{tot} = \Sigma_b + \Sigma_w$. Equation (4) can also be written as a dot product of a vector of weights $\mathbf{W}^T$, and an expanded vector $\boldsymbol{\varphi}(\mathbf{z}_i,\mathbf{z}_j)$ representing a trail:

$$s_{ij} = \begin{bmatrix} vec(\mathbf{P}) \\ vec(\mathbf{Q}) \\ \mathbf{c} \\ k \end{bmatrix}^T \begin{bmatrix} vec(\mathbf{z}_i\mathbf{z}_j^T + \mathbf{z}_j\mathbf{z}_i^T) \\ vec(\mathbf{z}_i\mathbf{z}_i^T + \mathbf{z}_j\mathbf{z}_j^T) \\ \mathbf{z}_i + \mathbf{z}_j \\ 1 \end{bmatrix} = \mathbf{W}^T\boldsymbol{\varphi}(\mathbf{z}_i,\mathbf{z}_j) \tag{5}$$

In the above linear expression, *vec*(·) stacks the columns of a matrix into a vector.

3. DNN BASED DISCRIMINATIVE TRAINING

Instead of using the ML criterion for training the PLDA model, some discriminative training methods [5-8] have been proposed to directly optimize the parameters **W** for discriminating between the same-speaker trial and a different-speaker trial. One of those methods is Support Vector Machine (SVM) based discriminative training [6], which refers to maximize the margin separating the scores of same-speaker trails and different-speaker trials. In this study, we follow the approach since it is computationally inexpensive on training and extremely fast on testing, and then modify it to jointly optimize **W** and Θ.

3.1. Objective function

The set of training examples comprises both same-speaker and different-speaker trials. Let $t_{ij} \in \{1,-1\}$ be the corresponding labels. We consider the training of our discriminatively trained classifier

as a nested optimization problem, its objective function can be written as

$$\min_{\Theta} E(\Theta) \tag{6}$$

where

$$E(\Theta) = \min_{\mathbf{W}} \frac{1}{2}\|\mathbf{W}\|_2^2 + \lambda \sum_{i,j} \max(0, 1 - t_{ij}\mathbf{W}^T \boldsymbol{\varphi}(\mathbf{z}_i(\Theta)\mathbf{z}_j(\Theta))) \tag{7}$$

Equation (7) denotes the original SVM based discriminative training which is to obtain an optimal $\mathbf{W}$ for a given network parameter set Θ. Equation (6) is to select an optimal Θ based on the margins and the hinge losses of the SVMs. Given Θ, let

$$\hat{\mathbf{W}} = \arg\min_{\mathbf{W}} E(\Theta)$$

and

$$\hat{E}(\Theta) = \frac{1}{2}\|\hat{\mathbf{W}}\|_2^2 + \lambda \sum_{i,j} \max(0, 1 - t_{ij}\hat{\mathbf{W}}^T \boldsymbol{\varphi}(\mathbf{z}_i(\Theta), \mathbf{z}_j(\Theta)))$$

According to the Theorem 4.1 in [18], since $\boldsymbol{\varphi}$ and its derivatives are smoothly varying functions of Θ, and $\hat{\mathbf{W}}$ is unique, $E(\Theta)$ can be proven differentiable and have derivatives given by

$$\frac{\partial E(\Theta)}{\partial \Theta} = \frac{\partial \hat{E}(\Theta)}{\partial \Theta}$$

Hence, our optimization problem can be solved by a gradient algorithm, and then $\hat{\mathbf{W}}$ must be calculated for the current Θ in each gradient step.

3.2. Optimization of PLDA model

Even though the Equation (7) is a standard SVM problem, we address the optimization by its gradient instead of any SVM optimization package, since there are hundreds of thousands of pairs of i-vectors in our training dataset. According to Cumani's work [6], the gradient can be calculated as:

$$\frac{\partial E}{\partial \mathbf{W}} = \begin{bmatrix} vec(\mathbf{P}) \\ vec(\mathbf{Q}) \\ \mathbf{c} \\ k \end{bmatrix} + \lambda \begin{bmatrix} vec\left(\sum_{ij} g_{ij}(\mathbf{z}_i\mathbf{z}_j^T + \mathbf{z}_j\mathbf{z}_i^T)\right) \\ vec\left(\sum_{ij} g_{ij}(\mathbf{z}_i\mathbf{z}_i^T + \mathbf{z}_j\mathbf{z}_j^T)\right) \\ \sum_{ij} g_{ij}(\mathbf{z}_i + \mathbf{z}_j) \\ \sum_{ij} g_{ij} \end{bmatrix} \tag{8}$$

where g_{ij} is the derivative of the loss function with respect to the dot product

$$g_{ij} = \begin{cases} 0 & if \quad t_{ij}\mathbf{W}^T \boldsymbol{\varphi}(\mathbf{z}_i, \mathbf{z}_j) \geq 1 \\ -t_{ij} & otherwise \end{cases} \tag{9}$$

3.3. Optimization of DNN parameters

As we know, mini-batch stochastic gradient descent (SGD) techniques have almost been the standard algorithm for training DNNs. Thus, considering the embedded DNN in our approach, it is natural to design a SGD based algorithm to implement the optimization of Equation (6). For each mini-batch $\boldsymbol{B}$, based on Equation (4), the gradient can be estimated as:

$$\begin{aligned} \frac{\partial E(\Theta)}{\partial \Theta} &= \lambda \sum_{(i,j)\in \boldsymbol{B}} g_{ij} \frac{\partial s_{ij}}{\partial \Theta} \\ &= \lambda \sum_{(i,j)\in \boldsymbol{B}} g_{ij} (\boldsymbol{\gamma}_{ij}^T \frac{\partial \mathbf{f}_i}{\partial \Theta} + \boldsymbol{\gamma}_{ji}^T \frac{\partial \mathbf{f}_j}{\partial \Theta}) \end{aligned} \tag{10}$$

where

$$\boldsymbol{\gamma}_{ij}^T = \left(\mathbf{z}_j^T(\mathbf{P} + \mathbf{P}^T) + \mathbf{z}_i^T(\mathbf{Q} + \mathbf{Q}^T) + \mathbf{c}^T\right)\mathbf{K}_i$$

$$\mathbf{K}_i = \mathbf{M}\left(\frac{\mathbf{I}}{\|\mathbf{f}_i - \boldsymbol{\mu}_f\|_2} - \frac{(\mathbf{f}_i - \boldsymbol{\mu}_f)(\mathbf{f}_i - \boldsymbol{\mu}_f)^T}{\|\mathbf{f}_i - \boldsymbol{\mu}_f\|_2^3}\right)$$

Obviously, two dot product terms inside the summation formula of Equation (10) can be separately calculated by Back Propagation (BP) algorithms. Note that those trials whose g_{ij} is equal to 0 do not participate in the gradient calculation, and therefore will not appear in the mini-batches. Adam algorithm [19] is employed here to update Θ. All the samples are shuffled in each epoch, and Early Stopping technique is used to judge convergence via a validation set. Our algorithm is listed below. $\mathbf{W}$ is initialized with a generatively trained PLDA. Θ is initialized as Θ_0

$$\Theta_0 = \arg\min_{\Theta} \sum_i \|\mathbf{f}(\mathbf{x}_i, \Theta) - \mathbf{a}_i\|_2^2 \tag{11}$$

where $\mathbf{a}_i$ is the output of a standard Fisher LDA approach.

Algorithm 1 SGD Based training Algorithm

Input: Training set, penalty factor λ of the SVM
Output: the optimal parameters $\hat{\Theta}$ and its corresponding $\hat{\mathbf{W}}$

1: Initialize Θ by a SGD algorithm based on Eq. (11)
2: Initialize $\mathbf{W}$ by the EM algorithm of PLDA
3: **for** each epoch
4: Estimate whitening matrix and calculate $\mathbf{z}_i$ by Eq.(2)
5: Calculate the LLR s_{ij} by Eq.(5) and g_{ij} by Eq.(9)
6: Filter out all the trials with g_{ij}=0
7: Run a Batch Gradient Descent algorithm based on Eq.(8) to Obtain $\hat{\mathbf{W}}$ (up to the maximum number of iterations or convergence)
8: Shuffle and fill the mini-batches
9: **for** each mini-batch $\boldsymbol{B}$
10: Calculate the gradient of the DNN using BP algorithm based on Eq. (10)
11: Update Θ using Adam algorithm
12: **end for**
13: **if** maximum epoch number is arrived **or** Early Stopping condition is met **then** stop
14: **end for**

4. EXPERIMENTS

4.1. Experiment setup

The performance of our DNN based discriminatively trained classifier is evaluated on the NIST SRE08 male short2-short3 core

data set. The common evaluation conditions (C1-C8) which respectively refer to microphone or telephone speech under channel matched or mismatched conditions are all covered in our evaluation experiments to test the generalization ability of our method. They contain 39433 trails from 1270 male speakers in total. We make use of the data from NIST SRE03-06 evaluation datasets as the development set. Both equal error rate (EER) and minimum detection cost function (DCF) are used for evaluation.

In the frame-level acoustic feature extraction, the speech is segmented by a 25ms hamming window shifting with 10ms frame rate. The passing frequency band is restricted to 300-3400 Hz. 19 Mel frequency cepstral coefficients (MFCC) with log energy are calculated with their first and second derivatives to form a 60-dimension feature. A full-covariance gender-independent UBM with 2,048 mixtures is trained and then the total variability subspace for i-vector extractor is estimated from the development set by using the Kaldi toolkit [20]. The dimension of i-vector is set to 600 in our experiments.

Baseline LDA and generatively trained PLDA models are trained on the male utterances of the development set, which include 10389 utterances for 1268 speakers. For training our DNN based discriminatively trained classifier, 300,000 different-speaker trails and 10,000 same-speaker trails are randomly selected from the above dataset as training data. Another 5000 trails are also randomly selected as validation set that consists of 4500 different-speaker trails and 500 same-speaker trails. The DNN includes 256 ReLUs in each hidden layer. The dimension of its inputs is 600, and the dimension of its outputs is set to 50.

The penalty factor λ of the SVM is set to 100. The size of mini-batches and the maximum number of epochs are separately set to 100 and 10000. For the Adam algorithm, the default setting given by paper [19] is adopted.

4.1. Results

Table 1 compares the obtained results from 4 i-vector speaker verification systems. The system denotes as LDA+GTPLDA, which serves as our baseline, is based on a LDA based dimensionality reduction and a generatively trained PLDA model. The baseline system is compared with three discriminatively trained systems. The system denotes as LDA+DTPLDA is based on a LDA based dimensionality reduction and a discriminatively trained PLDA model, which is obtained by running only the step 7 in Algorithm 1. For consistency, the LDA transformation reduces i-vector dimensionality to 50, and before the PLDA training and scoring stage, mean subtraction, length normalization and whitening are also adopted just as Equation (2) in the two systems above.

The system denotes as DNN+GTPLDA is based on a DNN based dimensionality reduction and a generatively trained PLDA model, which can be obtained by replacing the step 7 with a standard generative training algorithm of PLDA in Algorithm 1.

The system denotes as DNN+DTPLDA is based on a DNN based dimensionality reduction and a discriminatively trained PLDA model, which can be obtained by Algorithm 1. A DNN with 3 hidden layers is employed in the last two systems.

The results show that, compared with the baseline system, the LDA+DTPLDA system does bring a little bit performance improvements, as what have been reported in the previous studies. Even though the adopted PLDA model is generatively trained, the performance of the DNN+GTPLDA system is still better than the baseline, since the DNN embedded PLDA scoring function is indeed iteratively adjusted under the discriminative criterion. The best performance is achieved by the DNN+DTPLDA system, it proves that our proposed DNN based discriminative training method is effective.

Tabel 1. Performance of 4 i-vector speaker verification systems

Method	EER(%)	miniDCF
LDA+GTPLDA	8.52	0.041
LDA+DTPLDA	8.41	0.040
DNN+GTPLDA	8.26	0.037
DNN+DTPLDA	7.76	0.034

Tabel 2. Performance of the systems with different hidden layers

Method	EER(%)	miniDCF
DNN+DTPLDA (0 hidden layer)	8.33	0.039
DNN+DTPLDA (1 hidden layer)	8.17	0.036
DNN+DTPLDA (2 hidden layers)	7.78	0.034
DNN+DTPLDA (3 hidden layer)	7.76	0.034

In the next set of experiments, we investigated the impact of the number of hidden layers of the DNN on speaker verification performance. Table 2 shows speaker verification results of the DNN+DTPLDA systems using 0-3 hidden layers respectively on the same evaluation dataset with Table 1. Two observations can be made from this table. First, the larger the number of hidden layers, the better the performance. Second the performance of the system with 0 hidden layer is better than the LDA+DTPLDA system, although both of them adopt linear projection on dimensionality reduction. This shows that it is meaningful to integrate the dimensionality reduction process into the scoring stage.

5. CONCLUSION AND FUTURE WORK

We have presented a DNN based discriminative training method for i-vector and PLDA based speaker verification. It follows the idea of traditional discriminative training of PLDA and provides an effective way on embedding a DNN into the discriminatively trained PLDA scoring function. On NIST SRE evaluation data set, the resulting systems perform better. In the future, the effect of deeper networks and other forms of networks will be investigated. End to end systems that follow the strategy of this article will also be considered.

ACKNOWLEDGEMENTS

This research was supported by National Key Technologies Research and Development Program of China under Grant 2017YFB1002102 and National Natural Science Foundation of China under Grant U1736210

REFERENCES

[1] P. Kenny, “Bayesian speaker verification with heavy-tailed priors.” *in Proceedings of Odyssey, The Speaker and Language Recognition Workshop*, 2010, p. 14.

[2] N. Dehak, P. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, “Front-end factor analysis for speaker verification,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 4, pp. 788–798, 2011.

[3] N. Brummer, L. Burget, J. Cernocky, O. Glembek, F. Grezl, M. Karafiat, D. A. van Leeuwen, P. Matejka, P. Schwarz, and A. Strasheim, “Fusion of heterogeneous speaker recognition systems in the STBU submission for the NIST speaker recognition evaluation 2006,” *IEEE Transactions on Audio, Speech and Language Processing,* vol. 15, no. 7, pp. 2072–2084, 2007.

[4] N. Brummer, “Measuring, refining and calibrating speaker and language information extracted from speech,” *PhD thesis*, Stellenbosch: University of Stellenbosch, 2010.

[5] L. Burget, O. Plchot, S. Cumani, O. Glembek, P. Matejka, and N. Brummer, “Discriminatively trained probabilistic linear discriminant analysis for speaker verification,” *IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 4832–4835, 2011.

[6] S. Cumani, N. Brummer, L. Burget, and P. Laface., “Fast discriminative speaker verification in the i-vector space,” *IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 4852–4855, 2011.

[7] Q.Wang, and T. Koshinaka, "Unsupervised discriminative training of PLDA for domain adaptation in speaker verification." *INTERSPEECH*, 2017: 3727-3731.

[8] J. Rohdin, Johan, S. Biswas, and K. Shinoda, "Constrained discriminative PLDA training for speaker verification." *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2014:1670-1674.

[9] S. O. Sadjadi, J. W. Pelecanos, and W. Zhu, “Nearest neighbor discriminant analysis for robust speaker recognition,” in Proc. *INTERSPEECH*, Singapore, September 2014, pp. 1860–1864

[10] S. O. Sadjadi, S. Ganapathy, and J.W. Pelecanos, “Nearest neighbor discriminant analysis for language recognition,” in Proc. *IEEE International Conference on Acoustics, Speech and Signal Processing*, Brisbane, Australia, April 2015, pp. 4205–4209

[11] S. O. Sadjadi, S. Ganapathy, and J. W. Pelecanos. "The IBM 2016 speaker recognition system." *n Proceedings of Odyssey, The Speaker and Language Recognition Workshop*, 2016, pp.174-180.

[12] A. Khosravani, and M. M. Homayounpour, "Nonparam-etrically trained probabilistic linear discriminant analysis for i-Vector speaker verification." *INTERSPEECH* 2017:1019-1023.

[13] Y. Lei, N. Scheffer, L. Ferrer, and M. McLaren, “A novel scheme for speaker recognition using a phonetically aware deep neural network,” *IEEE International Conference on Acoustics, Speech and Signal Processing*, Florence, Italy, May 2014, pp. 1695–1699

[14] P. Kenny, V. Gupta, T. Stafylakis, P. Ouellet, and J. Alam, “Deep neural networks for extracting baum-welch statistics for speaker recognition,” *in Proceedings of Odyssey, The Speaker and Language Recognition Workshop*, 2014.

[15] M. Senoussaoui, N. Dehak, P. Kenny, R. Dehak, P. Dumouchel, “First attempt of boltzmann machines for speaker verification”, *In Proceedings of Odyssey-The Speaker and Language Recognition Workshop*，2012

[16] V. Ehsan, L. Xin, M. Erik, L. M. Ignacio, and G.-D. Javier, “Deep neural networks for small footprint text-dependent speaker verification,” *IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. 28, no. 4, 2014, pp. 357–366.

[17] L. Li, Y. Chen, Y. Shi, Z. Tang, and D. Wang， "Improved Deep Speaker Feature Learning for Text-Dependent Speaker Recognition." *Computer Science*, 2015: 426-429.

[18] J.F. Bonnans and A. Shapiro, “Optimization problems with perturbation: A guided tour”, *SIAM Review*, 40(2):202–227, 1998.

[19] D. P. Kingma, and J. Ba, “Adam: a method for stochastic optimization”, *Computer Science*, 2014.

[20] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz et al., “The Kaldi speech recognition toolkit,” *in Proc. of ASRU*, 2011, pp. 1–4.