MULTIPLE TEMPORAL SCALES BASED SPEAKER EMBEDDINGS LEARNING FOR TEXT-DEPENDENT SPEAKER RECOGNITION

Wenchao Wang[1,2] *Yike Zhang*[1,2] *Ji Xu*[1,2] *Yonghong Yan*[1,2,3]

[1] Institute of Acoustics, Chinese Academy of Sciences, China
[2] University of Chinese Academy of Sciences, China
[3]Xinjiang Technical Institute of Physics and Chemistry, Chinese Academy of Sciences, China

ABSTRACT

To extract high speaker-sensitive embeddings from deep neural networks is still a challenge in the field of speaker recognition. This paper proposes a novel network that learns speaker embeddings from multiple temporal scales. This idea comes from the recent biological research that the human auditory system has a mechanism of fusing multi-timescale information together to encode sound information. A two-pathway neural network is presented, in which one pathway focuses on short-time (or *local*) traits and the other focuses on long-range (or *global*) scale. Both traits are fused into one feature vector and the utterance-level speaker embeddings are extracted from these features. Experimental results show that different timescale traits can complement each other. And their fusion, which refer to as t-vector, outperforms i-vector and other deep embeddings. Moreover, with the end-to-end training, t-vectors can obtain excellent performance even using simple scoring approach like cosine distance.

Index Terms— speaker recognition, speaker embedding, biological research, different timescales, t-vector, end-to-end

1. INTRODUCTION

The factor analysis methods[1, 2] have been the state-of-the-art speaker representation learning method in the last decades. Using phonetic-discriminative deep neural networks (DNNs) in place of Gaussian mixture model (GMM)[3] to extract Baum-Welch statistics[4, 5] has made considerable progress in some situations. In recent, it has become a hot research topic to learn efficient and sensitive speaker embeddings using speaker-discriminative DNNs. The d-vectors, which are first proposed in [6], are extracted from the last hidden layer to replace i-vectors but they get suboptimal performance. Since then, many related works are proposed to improve performance. There are kinds of neural networks[7, 8, 9, 10, 11, 12] and some with end-to-end loss functions[13, 14, 15] that show advantages on text dependent or independent tasks. These results confirm that speaker embeddings extracted from DNN outperform i-vectors on short duration utterances.

Studies mentioned above do not reveal the auditory processing mechanism in human brains, which is proved in the newest biological research[16] that the human auditory system employs at least a 2-timescale processing mode. This motivates us to learn speaker embeddings from multiple temporal scales. Inspiration also comes from[17], in which the authors present a Two-Pathway Generative Adversarial Network (TP-GAN) for photorealistic frontal view synthesis. The generator of this TP-GAN contains two pathways, one for global transformation processing and the other for local landmarks. In this work, we also propose a two-pathway neural network model to simulate the human auditory system. This model is constructed with paired global-focused and local-focused pathways together to learn distinct perceptual traits of sound, as the paired short and long timescales employed by our brains. Then speaker embeddings are derived by averaging frame-level feature vectors, which are fused from perceptual traits of each pathway. The proposed embeddings (or *t-vectors*) are also set as input samples for the triplet-loss scheme, in order to study the performance of these embeddings in an end-to-end system. Experiments show that t-vectors contain more speaker characteristics and the end-to-end training makes them more speaker sensitive.

2. TWO-PATHWAY NEURAL NETWORK

2.1. Network Architecture

This work focuses on the effect of multiple learning timescales, which are represented by different learning window sizes instead of different speech lengths. Therefore in this model, two timescale pathways are designed to learn different perceptual traits from the same speech. One of them is the local perceptual learning pathway, which is designed to simulate the short timescale processing mode, focusing on correlations inside the local field. The other is the global perceptual learning pathway, which is designed to simulate the long timescale,

This work is partially supported by the National Natural Science Foundation of China (Nos.11590770-4, U1536117,11504406,11461141004), the National Key Research and Development Program (No.2016YFC0800503), the Key Science and Technology Project of the Xinjiang Uygur Autonomous Region (No.2016A03007-1).

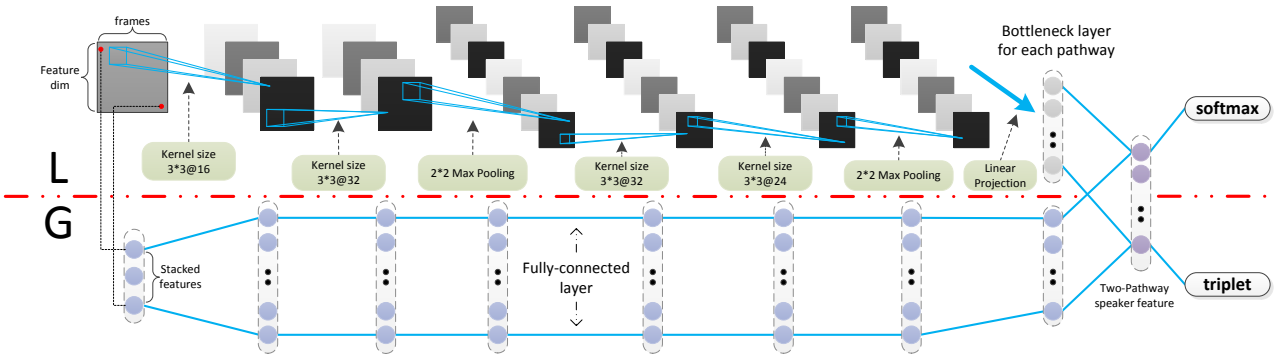


Fig. 1. Structure of the two-pathway neural network. The green blocks indicate convolution kernel windows and filter channels. The light blue lines indicate fully-connected linear layers.

aiming at representing correlations among entire features. Because of the two timescales interaction in our brains, different perceptual traits are supposed to play different roles on recognition decisions. Therefore both global and local traits are fused into one feature vector. In this paper, outputs from two pathways are called "traits" to prevent the confusion of "features", which "features" represents the fusion of "traits".

The architecture of the proposed two-pathway neural network model is shown in Fig.1. This model is split into two parts by the red line. The upper part, which is marked as "L", represents the local perceptual learning pathway. It is consisted of a CNN with the VGG-style[18]. It involves 4 convolution layers. And every two of them are followed by a max-pooling layer. Outputs from the last convolution layer are projected to a bottleneck layer to extract the local perceptual traits. In this pathway, small convolution window is regarded as the short timescale.

The lower part, which is marked as "G", represents the global perceptual learning pathway. In this pathway, two kinds of architectures are considered at first. One is the same as "L" but with a larger window size, which represents the long timescale. The alternative one is a fully-connected DNN that learns correlations from all input dimensions. For the first one, using a convolution window covering entire features will make the rest network equivalent to the second one. Ultimately the second one is adopted and the effectiveness is proved in the experiment part. This DNN has seven hidden layers, corresponding to six convolution layers and the last bottleneck layer in "L". The last bottleneck layer in this pathway is designed to produce global perceptual traits.

The feature producing part aims at simulating the encoding mechanism that integrates different timescales into one representation. The "Two-Pathway speaker feature" layer receives the concatenations of global and local traits as input and projects them into discriminative feature vectors.

2.2. Classifier Model

The softmax network with cross-entropy loss is first applied to this model. When this model is well trained, the last softmax layer is discarded and the speaker embeddings are extracted from the "Two-Pathway speaker feature" layer. These embeddings will replace of i-vectors for backend models scoring.

Then deep embeddings extracted above are set as input to end-to-end training. In this paper, we take the widely applied triplet-loss function[19, 20] to verify the effectiveness of these embeddings in end-to-end system . Triplet-loss function aims at minimizing the distance between embeddings of the same speaker and maximizeing the distance of different speakers. In its training process, a triplet $\tau = (x_a^\tau, x_p^\tau, x_n^\tau)$ is selected as input. Samples in this triplet are called anchor, positive and negative embeddings. The anchor and positive are selected from the same speaker while the anchor and negative are not. This loss can achieve a better separation between positive and negative samples by adding a margin coefficient α. For any triplet, the training goal is $L(\tau) + \alpha < 0$ and $L(\tau)$ is defined as follow:

$$L(x_a^\tau, x_p^\tau, x_n^\tau) = \left\| f(x_a^\tau) - f(x_p^\tau) \right\|_2^2 - \| f(x_a^\tau) - f(x_n^\tau) \|_2^2 \quad (1)$$

where $\|*\|_2^2$ is the Euclidean norm and f is the embedding extracted from the network. In this paper, α is set to 0.2.

3. BASELINE SYSTEM

3.1. I-Vector

The i-vector is a compact representation of the speech utterance, containing both the speaker and channel characteristics. The i-vector extractor projects high dimensional statistic features onto a total variability space T to form fixed-length low dimensional embeddings. It is defined as:

$$M = m + T\omega \quad (2)$$

where M is the GMM supervector and m is the mean supervector of the training data, ω is the so-called i-vector.

3.2. Deep Embedding

In order to compare with the t-vector, two baseline deep embedding systems are built from two subparts of the proposed model. Speaker embeddings produced by "L", which is consisted of CNN, are represented as the *c-vectors*. Because of the network in "G" is simplified to the fully-connected DNN, speaker embeddings are represented as the *d-vectors*. The d-vector has shown advantages on text-dependent tasks in [20]. However that work is done only on female data and they get worse d-vector baseline than i-vector. In this paper, we take the similar d-vector network with more speakers.

For each of these deep embedding baselines training, in order to prevent the interaction of the other pathway, one pathway is temporarily discarded and the other is maintained to construct the system. Actually in our entire model training process, well trained baseline models of each pathway are adopted as the pre-trained models, then the entire model is fine-tuned to produce t-vectors.

In addition to cosine distance, PLDA[21] is also trained as backend classifier. Some studies noted that PLDA may not fit well to d-vectors[8], so results of linear discriminant analysis (LDA) are also employed. All embeddings are length normalized[22] prior to these approaches.

4. EXPERIMENTS

4.1. Database

Experiments are conducted on the Part3 of RSR2015 text-dependent database[23]. In this portion, speakers are prompted with random sequences of digits. Sequences from enrollment sessions have ten-digit content and sequences from test sessions have five random digits. This database is divided into background (bkg), development (dev) and evaluation (eval) subsets. Bkg and dev subsets are used for neural networks, PLDA and LDA training, while the data used for the i-vector extractor and UBM model are merged from Switchboard, Fisher and NIST SRE 2004-2008 databases as the valid data is lack for the i-vector extractor training only on digits. The standard eval trials are used for experiments. Sequences in RSR2015 database are down-sampled to 8kHz to match other databases. And in this paper, female and male speakers are merged together to make gender-independent experiments.

4.2. System Settings

Acoustic cepstral features are estimated by a 20 ms window with the 10 ms shift. The 60-dimensional MFCC feature vector consists of 19 cepstral coefficients and the log energy, along with their first and second derivations. These features are used for the i-vector system and network input. A gender-independent UBM model is composed of 1024 Gaussian components and the dimensionality of the total variability space is set to 400. The i-vector is projected to a 150-dimensional vector by the LDA using kaldi toolkit[24].

For network input, a symmetric 5-frames window constructed 11 frames are used. One-stride shift across the time and frequency and zero-padding are adopted for convolution network. Batch normalization is applied to improve the training convergence. For "G", neurons in the first six hidden layers are 1024, corresponding to the UBM components. Traits from the bottleneck layer with 512 neurons in each pathway are concatenated to a 1024-dimensional vector. Then they are projected into the deep feature layer, whose neurons are 400 to match the i-vector dimensionality. The softmax non-linear layer has 194 neurons, corresponding to speakers in the training set. ReLU activation function is employed by all hidden layers. The Adam optimizer is employed with the learning rate set to 0.001 and batch size set to 512. For the end-to-end system, learning rate is set to 0.0001 in case of overfitting and the offline sampling approach[25] is adopted for triplets.

4.3. Experimental Results

The performance of t-vector system compared with the baseline systems is shown in this section. The results are presented in terms of equal error rate (EER). In this paper, we just show the effect of learning embeddings from two timescales, so complicated networks are not involved.

Table 1. EER Results with Different Backend Classifiers

Systems	Cosine(%)	PLDA(%)	LDA(%)
i-vector	17.83	9.26	7.13
c-vector	16.98	10.12	7.42
d-vector	9.15	8.64	6.59
t-vector	**8.98**	**8.02**	**5.84**

As is shown in Table.1, it can be observed that both PLDA and LDA efficiently improve performance. The best performance obtained by LDA suggests that it is still effective to make good use of the intra- and inter-class information for deep embeddings. The d-vector system obtains performance improvements on the i-vector system while the c-vector system does not. A possible reason for this is that the MFCC features are suboptimal for CNN learning. We keep this input for all networks because this work tends to learn information from multiple timescales rather than kinds of features. Compared with the i-vector system, the t-vector system excellently improves performance. It achieves relative improvements of 13.4% and 19.4% for EER using PLDA and LDA respectively. Even fused with a suboptimal trait, the t-vector still obtains relative gains of 7.2% and 11.4% for EER compared with the d-vector. This improvement is credible since we get a better baseline than other studies[20].

In this part, the effect of different timescale combinations are discussed. 3 convolution windows are selected, including 3×3, 5×5 and 7×7. DNN with 2048 hidden neurons is also experimented to show the effect of large parameters. The results using LDA are shown in Table.2. In these paired timescales, the smaller one is adopted as "L" and the larger is adopted as "G". From the results, we can find that the larger convolution window is adopted, the better performance is obtained for the single pathway. When the smallest convolution window is adopted as "L", fusing with a large window obtains better performance than c-vector but worse than the larger window. It suggests that simplely fusing traits of two timescales can not get ideal performance. On the other hand, when DNN is adopted as "G", fusing with a short timescale obtains better performance, but a larger convolution window leads to a worse performance. These results confirm that fusing the local and global traits efficiently strengthen the embeddings discriminative. And the shortest and longest timescales complement each other best, because of the minimum overlapped correlations learned from features.

Table 2. Results of different timescale combinations

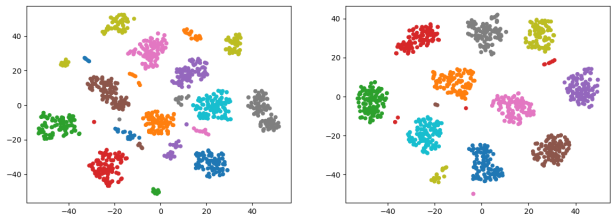
Systems	L-Pathway	G-Pathway	EER(%)
i-vector	–	–	7.13
d-vector	–	1024 DNN	6.59
–	–	2048 DNN	6.75
–	7×7	1024 DNN	6.51
–	5×5	1024 DNN	6.32
–	7×7	–	6.67
–	5×5	–	6.90
c-vector	3×3	–	7.42
–	3×3	5×5	6.97
–	3×3	7×7	6.85
t-vector	3×3	1024 DNN	**5.84**

We also noted that this model is somehow like a feature fusion. However, the proposed embeddings should have their unique identified characteristics after traits fusing by the "two-pathway speaker feature" layer. Therefore, fusion results of the score domain are presented here. From the LDA results in Table.3, we can observe that the t-vector is still complementary to other traits. The fusion of t-vector and both two traits not only obtains improvements of 17.8% for EER compared with the single t-vector system, but also outperforms the fusion of these two traits by 9.4%.

To study the performance of embeddings in an end-to-end system, the triplet-loss is applied for t-vectors. And we use t-SNE[26] to draw embedding samples from 10 random selected speakers, intending to show the impact of different training approaches. From the Fig.2, it can be observed that embeddings extracted from the end-to-end training have high discriminative than from the conventional approach. Embeddings from the same speakers gather more closely while those from different speakers distinguish more clearly.

Table 3. Results of embedding fusions

Fusion Vectors	c & d	c & t	d & t	c & d & t
EER(%)	5.30	5.28	5.12	**4.80**



(a) Cross-Entropy Training (b) End-to-End Training

Fig. 2. T-SNE visualization of embeddings learned from different training approaches. Each color represents a speaker.

There are still sequences not well distinguished in Fig.2, so backend models should also play a role. These results are presented in Table.4. The LDA performance comfirms that it is effective to utilize the intra- and inter-class variations even for embeddings extracted from end-to-end system. It reduces the EER by 7.9% than original t-vectors. The impact of end-to-end training is shown by the cosine distance result that t-vectors are able to directly get excellent performance without other backend classifiers. It obtains improvement of 26% for EER compared with t-vectors training conventionally, and it even outperforms PLDA backend of 13%.

Table 4. T-Vector Results with Different Training Method

Methods	Cosine(%)	PLDA(%)	LDA(%)
Cross-Entropy	8.98	8.02	5.84
End-to-End	**6.65**	**7.64**	**5.38**

5. CONCLUSION

In this paper, we present a novel two-pathway neural network to extract deep speaker embeddings. These embeddings are fused from two sub-traits, which represent the information learned by short-time (or *local*) and long-scale (or *global*) processing modes in the human auditory system. Experiments are conducted to verify the proper timescale combinations and the results confirm that traits learned from different temporal scales are complementary and embeddings (or *t-vectors*) extracted from the best timescale combination achieve better performance than other representations. Then triplet-loss is used to show the impact of end-to-end learning for t-vectors and finally more speaker-sensitive embeddings are produced.

6. REFERENCES

[1] N. Dehak, P. Kenny, R. Dehak, P. Ouellet, and P. Dumouchel, "Front end factor analysis for speaker verification," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 19, no. 4, pp. 788–798, May 2011.

[2] P. Kenny, G. Boulianne, P. Ouellet, and P. Dumouchel, "Joint factor analysis versus eigenchannels in speaker recognition," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 15, no. 4, pp. 1435–1447, 2007.

[3] D.A. Reynolds, T.F. Quatieri, and R.B. Dunn, "Speaker verification using adapted gaussian mixture models," *Digital signal processing*, vol. 10, no. 1, pp. 19–41, 2000.

[4] P. Kenny, V.Gupta, T. Stafylakis, P. Ouellet, and J. Alam, "Deep neural networks for extracting baum-welch statistics for speaker recognition," in *Proc. IEEE Odyssey*, 2014, pp. 293–298.

[5] Y. Lei, N. Scheffer, L. Ferrer, and M. McLaren, "A novel scheme for speaker recognition using a phonetically-aware deep neural network," in *Proc. ICASSP*, 2014, pp. 1714–1718.

[6] E. Variani, X. Lei, E. McDermott, I. L. Moreno, and J. Gonzalez-Dominguez, "Deep neural networks for small footprint text-dependent speaker verification," in *Proc. ICASSP*, 2014, vol. 28, pp. 357–366.

[7] F.A.R.R Chowdhury, Q. Wang, I.L. Moreno, and L. Wan, "Attention-based models for text-dependent speaker verification," in *Proc. ICASSP*, 2018.

[8] L. Li, Y. Chen, Y. Shi, Z. Tang, and D. Wang, "Deep speaker feature learning for text-independent speaker verification," in *Proc. Interspeech*, 2017, pp. 1542–1546.

[9] Sergey S. Novoselov, O. Kudashev, V. Schemelinin, I. Kremnev, and G. Lavrentyeva, "Deep cnn based feature extractor for text-prompted speaker recognition," in *Proc. ICASSP*, 2018.

[10] D. Snyder, D. Garcia-Romero, D. Povey, and S. Khudanpur, "Deep neural network embeddings for text-independent speaker verification," in *Proc. Interspeech*, 2017, pp. 999–1003.

[11] Y. Liu, Y. Qian, N. Chen, T. Fu, Y. Zhang, and K. Yu, "Deep feature for text-dependent speaker verification," *Speech Communication*, vol. 73, pp. 1–13, 2015.

[12] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "X-vectors: robust dnn embeddings for speaker recognition," in *Proc. ICASSP*, 2018.

[13] S. Zhang, Z. Chen, Y. Zhao, J. L, and Y. G, "End-to-end attention based text-dependent speaker verification," in *Proc. IEEE SLT Workshop*, 2016, pp. 171–178.

[14] G. Heigold, I. Moreno, S. Bengio, and N. Shazeer, "End-to-end text-dependent speaker verificaiton," in *Proc. ICASSP*, 2016.

[15] D. Snyder, P. Ghahremani, D. Povey, D. Garcia-Romero, Y. Carmiel, and S. Khudanpur, "Deep neural network-based speaker embeddings for end-to-end speaker verification," in *Proc. SLT Workshop*, 2016.

[16] X. Teng, X. Tian, J. Rowland, and D. Poeppel, "Concurrent temporal channels for auditory processing: Oscillatory neural entrainment reveals segregation of function at different scales," *Plos Biology*, vol. 15, no. 11, 2017.

[17] R. Huang, Z. Shu, T. Li, and R. He, "Beyond face rotation: global and local perception gan for photorealistic and identity preserving frontal view synthesis," in *Proc. ICCV*, 2017, pp. 2458–2467.

[18] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. ICLR*, 2015.

[19] H. Bredin, "Tristounet: triplet loss for speaker turn embedding," in *Proc. ICASSP*, 2017, pp. 5430–5434.

[20] S. Dey, T. Koshinaka, P. Motlicek, and S. Madikeri, "Dnn based speaker embedding using content information for text-dependent speaker verificaiton," in *Proc. ICASSP*, 2018.

[21] S.J.D. Prince and J.H. Elder, "Probabilistic linear discriminant analysis for inferences about identity," in *IEEE 11th International Conference on Computer Vision*, 2007, pp. 1–8.

[22] D. Garcia-Romero and C.Y. Espy-Wilson, "Analysis of i-vector length normalization in speaker recognition systems," in *Proc. Interspeech*, 2011, pp. 249–252.

[23] A. Larcher, K.A. Lee, B. Ma, and H. Li, "Text-dependent speaker verification: Classifiers, databases and rsr2015," *Speech Communication*, vol. 60, pp. 56–77, 2014.

[24] 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. ASRU Workshop*, 2011.

[25] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *Proc. CVPR*, 2015, pp. 815–823.

[26] L. v. d. Maaten and G. Hinton, "Visualizing data using t-sne," *Machine Learning Research*, 2008.