INTERSPEECH 2017
August 20–24, 2017, Stockholm, Sweden

Joint Training of Expanded End-to-end DNN for Text-dependent Speaker Verification

Hee-soo Heo[1], *Jee-weon Jung*[1], *IL-ho Yang*[1], *Sung-hyun Yoon*[1], *and Ha-jin Yu*[1]

[1] School of Computer Science, University of Seoul, Korea

zhasgone@naver.com, aberforth@naver.com, heisco@hanmail.net,
ysh901108@naver.com, hjyu@uos.ac.kr

Abstract

We propose an expanded end-to-end DNN architecture for speaker verification based on b-vectors as well as d-vectors. We embedded the components of a speaker verification system such as modeling frame-level features, extracting utterance-level features, dimensionality reduction of utterance-level features, and trial-level scoring in an expanded end-to-end DNN architecture. The main contribution of this paper is that, instead of using DNNs as parts of the system trained independently, we train the whole system jointly with a fine-tune cost after pre-training each part. The experimental results show that the proposed system outperforms the baseline d-vector system and i-vector PLDA system.

Index Terms: Speaker verification, i-vector PLDA, end-to-end DNN

1. Introduction

Speaker verification is performed in units of utterance, assuming only one speaker for one utterance. Therefore, most of the systems with high performance in the conventional studies on speaker verification extract utterance-level features and utilize them [2-8, 10, 11]. For example, the operation of i-vector probabilistic linear discriminant analysis (PLDA) system [9, 10], which is a state-of-the-art system in the field of speaker verification, is as follows. First, frame-level features such as mel-filterbank cepstral coefficients (MFCCs) are extracted from the input speech signal and modeled by Gaussian mixtures model (GMM). Utterance-level features such as the i-vectors are extracted by combining the information of frame-level features from each utterance. In order to increase the discrimination power of the utterance-level features or to reduce the dimension, the feature enhancement techniques such as linear discriminant analysis (LDA) or within class covariance normalization (WCCN) are applied. The enhanced utterance-level features are modeled and scored at trial-level for speaker verification. The trial is a unit for speaker verification consisting of two utterance-level features. A simple trial-level scoring can be performed by calculating cosine similarity from two utterance-level features. PLDA technique is applied to perform trial-level modeling and more sophisticated scoring. Therefore, the processes from frame-level feature extraction to trial-level scoring of the i-vector PLDA system can be considered as five stages constituting the speaker verification system. Figure 1 shows the five stages described above and components that perform each stage based on i-vector PLDA system.

In recent years, deep neural networks (DNNs) have been applied to many machine learning fields, and considerably improved the performance [1]. Based on these results, some studies have been carried out to apply DNN to some of the stages required for speaker verification. In [2, 3], a method to calculate Baum-Welch statistics corresponding to the second and third stage with DNN is proposed. There have also been studies that performed utterance-level feature enhancement by applying DNN to the fourth stage [4]. In [5], b-vector based system that can perform the fifth stage using DNN as a binary classifier is proposed. Although an end-to-end architecture that performs speaker verification with DNN was introduced in [6], the proposed architecture only performs the second and fifth stages. The previous researches on end-to-end DNN architectures limits the usages to some stages of speaker verification. Therefore, we propose an expanded end-to-end DNN (EEEnet) architecture that covers from the second to fifth stage of speaker verification. We embed layers into the EEEnet that perform similar operations to all the components used in the i-vector PLDA system. In addition, we propose a joint-training technique that can effectively train all of the layers of the EEEnet simultaneously.

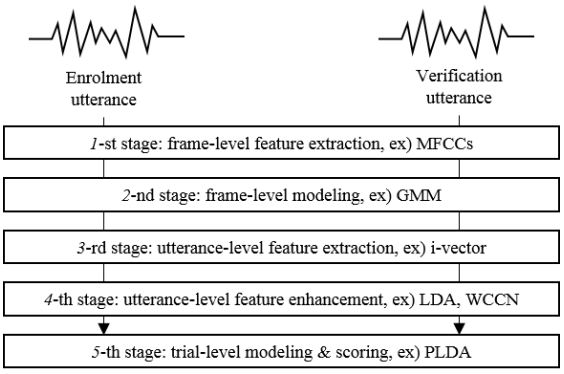

Figure 1: *General flow of speaker verification*

In section 2, we describe the conventional speaker verification systems based on DNNs. Section 3 describes the proposed system. Finally, section 4, 5, and 6 describe the experiments, conclusions, and future works respectively.

2. Conventional systems

2.1. D-vector based systems

D-vector based systems use the activations of the specific hidden layer of DNN as an utterance-level feature [6-7]. In general, a DNN used for extracting d-vectors is trained as a speaker identifier. A d-vector is derived by averaging node output values which are the results of feed-forwarding frame-level features into a fully-connected DNN. A d-vector $\boldsymbol{d}$ of each utterance is defined as the following equation.

http://dx.doi.org/10.21437/Interspeech.2017-1050

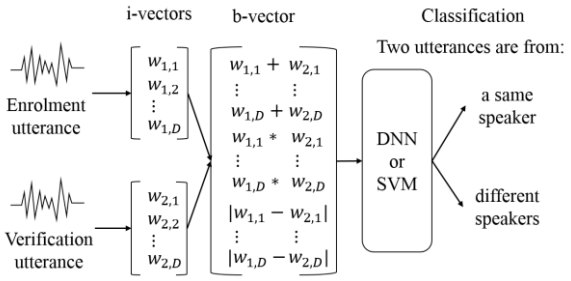

Figure 2: *Flow of speaker verification using the b-vector based system.*

$$\boldsymbol{d} = \frac{\sum_{i=1}^{T} \boldsymbol{h}_i}{T}, \tag{1}$$

where T is the number of frames in each utterance and $\boldsymbol{h}_i$ is the activation vector of the hidden layer, as a result of feed-forwarding the i-th frame.

In the d-vector system, the first step is to train the DNN as a speaker identifier (or classifier) with a development set. Next, the enrollment phase estimates a speaker model by averaging d-vectors over time. Then the evaluation phase, the cosine similarities between speaker model and the d-vector of an evaluation utterance are calculated. The key assumption underlying in the d-vector system is that the DNN can represents the frame-level features properly in the process of training the DNN as a speaker identifier. However, d-vector based system has a limitation that DNN is used only in the second and the third stage.

2.2. B-vector based system

A b-vector based system classifies a trial composed of two utterances into the two classes [5]. The two classes mean that the utterances are spoken from "a same speaker" or "different speakers", respectively. In the system, each utterance is represented by a feature such as an i-vector which is an utterance-level feature, and a b-vector is extracted by applying simple binary operations. For example, element-wise addition, subtraction, or multiplication of the two i-vectors can be used for extracting a b-vector such as the following equations.

$$\boldsymbol{b_a} = \boldsymbol{w_1} \oplus \boldsymbol{w_2}, \tag{2}$$

$$\boldsymbol{b_m} = \boldsymbol{w_1} \otimes \boldsymbol{w_2}, \tag{3}$$

$$\boldsymbol{b_s} = |\boldsymbol{w_1} \ominus \boldsymbol{w_2}|, \tag{4}$$

where $\boldsymbol{w}_1$ and $\boldsymbol{w}_2$ are i-vectors from each utterance and $\oplus$, $\otimes$ and $\ominus$ are the element-wise addition, multiplication and subtraction operations, respectively. The b-vectors based on binary operations describe the relationship between the two utterances, so that a classifier such as a DNN can perform speaker verification [5]. The flow of SV using the b-vector is shown in figure 2.

3. Proposed system

Most of the studies for speaker verification apply DNN to only some stages of speaker verification. In this paper, we propose EEEnet, an end-to-end DNN architecture that can replace from the second to the last stage of speaker verification at once.

3.1. Architecture

The EEEnet is composed of three groups of layers as shown in figure 3: frame-level, utterance-level, and trial-level layers. Frame-level layers perform the second and third stage shown in figure 1, and have similar role with i-vector extraction in the conventional i-vector PLDA system. Utterance-level layers perform the fourth stage, and have similar role of LDA in i-vector PLDA system which enhance the discriminability power of utterance-level features. Trial-level layers perform the fifth stage, and have the role of PLDA which is for trial-level scoring.

The structure of the frame-level layers is the same as the DNN d-vector system aforementioned in section 2.1. The frame-level layers extract the acoustic features and transform them for speaker identification or verification. Utterance-level features are extracted by averaging the last frame-level layer outputs.

The utterance-level layers enhance the utterance-level features. We apply the residual learning method by using identity mapping [12] for the effective feature enhancement. The residual learning is a technique to train DNNs that only find the residual values needed for the feature enhancement, rather than finding new feature vectors. It is known from previous studies that this technique can transform or enhance the input features effectively [12]. The enhanced vectors from the conventional DNNs and those from the DNNs with residual learning are expressed as:

$$\mathbf{y} = \mathcal{F}(\mathbf{x}, \mathrm{W}), \tag{5}$$

$$\boldsymbol{y}_{res} = \mathcal{F}(\mathbf{x}, \mathrm{W}) + \mathbf{x}, \tag{6}$$

where $\mathbf{y}$ and $\boldsymbol{y}_{res}$ denote feature vectors enhanced by conventional DNN and residual learning DNN, respectively, and $\mathbf{x}$ denotes an input to the DNNs. The output of the DNN defined by weight parameter W, is denoted by function $\mathcal{F}(\cdot)$. The residual learning can be implemented on the network by the identity connections shown in Figure 3.

Finally, the trial-level layers similar to the b-vector system make final decision for speaker verification. In the trial-level layers, we define the b-vector operations as:

$$\boldsymbol{b}'_a = (\boldsymbol{w_1} \oplus \boldsymbol{w_2})/2, \tag{7}$$

$$\boldsymbol{b}'_m = \sqrt{|\boldsymbol{w_1} \otimes \boldsymbol{w_2}|} \otimes sgn(\boldsymbol{w_1} \otimes \boldsymbol{w_2}), \tag{8}$$

$$\boldsymbol{b}'_s = |\boldsymbol{w_1} \ominus \boldsymbol{w_2}| \otimes sgn(\boldsymbol{w_1} \oplus \boldsymbol{w_2}) * 2, \tag{9}$$

where the function sgn returns the signs of each elements in the vector. The operations are defined to compensate for the changes of the scales of the feature vectors caused by each binary operation.

3.2. Joint-training method

It is difficult and inefficient to train the entire network at a time. Therefore we insert two new output layers that are different with conventional one and propose a novel training method. The newly inserted output layers are activated by the soft-max function and perform speaker identification at the frame-level and utterance-level, respectively. We expected that it would provide additional information to the network by adding output layers that perform operations related to the original purpose (identification related to speaker verification), such as multitask learning [13]. In addition, we expected that the problem of gradient vanishing that might occur when

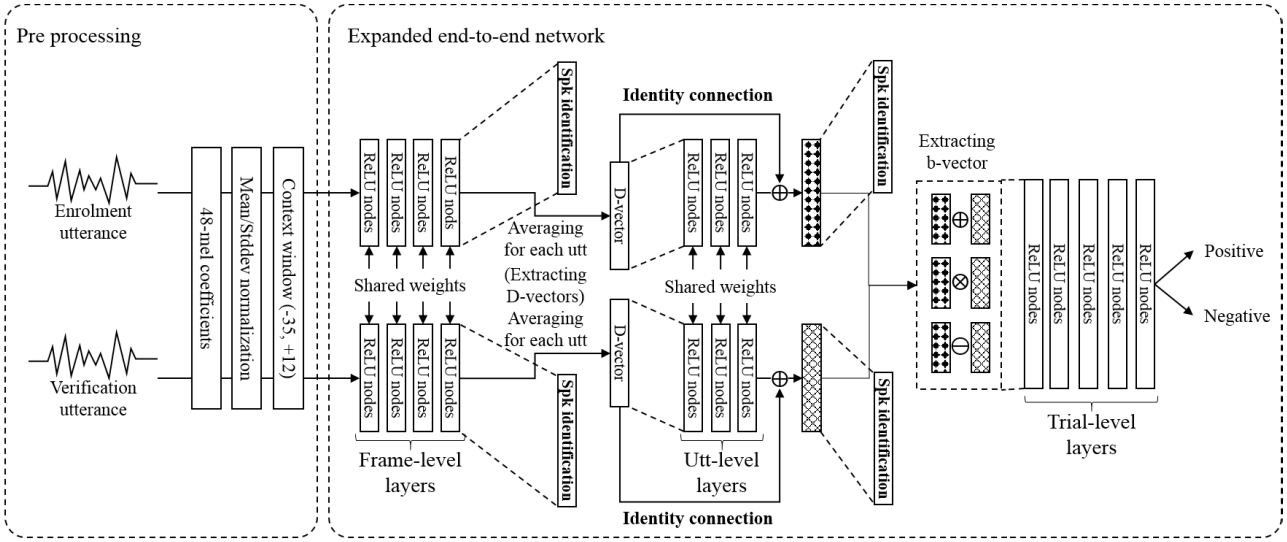

Figure 3: *Flow of speaker verification using the expanded end-to-end network.*

learning the weights of layers that are far away from the conventional output layer [14], could be solved to some extent. This structure is inspired by the colorization study related to image processing [15]. The entire architecture including newly inserted output layers is joint-trained by the novel method as follows.

We pre-train the components of the EEEnet in two steps. The first pre-training is fine-tuning the frame-level layers with fine-tune cost defined as

$$F_{frame} = NLL_{spkid\_frame}, \quad (10)$$

where NLL_{spkid_frame} means a negative log likelihood (NLL) for speaker identification of each frame calculated from the output layer right next the frame-level layers in figure 3. The second pre-training is done by fine-tuning the frame-level layers and utterance-level layers simultaneously with fine-tune cost defined as

$$F_{utt} = \alpha * NLL_{spkid\_frame} + (1-\alpha) * NLL_{spkid\_utt}, \quad (11)$$

where NLL_{spkid_utt} means an NLL for speaker identification of each utterance calculated from the output layer next to the utterance-level layers in figure 3 and α means a weight factor between the two NLLs. Finally, the entire network is fine-tuned with fine-tune cost defined as

$$F_{EEEnet} = \alpha * \frac{NLL_{spkid\_frame} + NLL_{spkid\_utt}}{2} + (1-\alpha) * NLL_{spkvrf}, \quad (12)$$

where NLL_{spkvrf} means NLL for SV of each trial calculated from the final output layer .

We expect that by using the fine-tune cost defined with various NLLs, more information will be provided to the network. For example, the information about speaker identity for each frame or utterance is additionally provided to the EEEnet. Also, the gradient vanishing problems can be mitigated, because the errors calculated in the new inserted output layers are directly back-propagated to the frame- or utterance-level layers.

EEEnet is an architecture designed to perform from the second stage to the final stage of speaker verification with only DNNs. Therefore, it can be expected that each stage will be more suitable for speaker verification. For example, in the i-vector PLDA system, frame-level modeling is performed by the GMM that simply represent the distribution of frame-level features. However, frame-level layers of the EEEnet can perform more suitable frame-level modeling for speaker verification in the course of being trained with fine-tune cost defined by Eq. (12).

4. Experiments

To evaluate performances of the EEEnet, we designed text-dependent SV experiments. D-vector based systems are proposed for text-dependent task. All the systems with the DNN is implemented in the Theano environment [17, 18]. We use Kaldi [19] which is an open-source speech and speaker recognition toolkit for i-vector PLDA system.

4.1. Database

All the experiments in this study were carried out using Korean speech database for text-dependent speaker recognition distributed by Electronics and Telecommunication Research Institute (ETRI). This database contains the speech of 250 speakers, and each speaker read 10 sentences, 20 times. The voices of 150 speakers were used as a development set, and the voices of 100 speakers was used as an evaluation set.

Ten and forty utterances with length of approximately one second (and one sentence) were used for speaker enrolment and verification respectively.

4.2. I-vector PLDA system

60-dimensional feature vectors (19 MFCCs + energy + Δ + ΔΔ) were extracted using a 25-ms window with 10-ms shifts, and then cepstral mean normalization (CMN) was applied.

A gender-independent UBM, containing 128 Gaussian components, and a TVM with dimensionality 200 were trained, both with 5 iterations. LDA was applied to reduce the dimensions of the i-vectors to 100. Length normalizations were applied to the i-vector before and after applying the LDA.

Table 1: *Performances in EER of the baseline and proposed systems.*

	Systems	EER (%)
Baselines	D-vector	2.95
	I-vector PLDA	1.15
Proposed	EEEnet	1.87
	I-vector + EEEnet	**1.02**

For the baseline system, a PLDA model was estimated using the dimensionality-reduced i-vectors.

4.3. D-vector based system

The baseline d-vector system is composed of 4 fully-connected hidden layers, 3 having 1024 nodes and the last layer having 512 nodes. Input layer size is 48*48, a 48-dimensional feature extracted by mel-filterbank method and then concatenated with 35 previous frames and 12 upcoming frames. Output layer has 150 nodes, same with the number of speakers in development set. We used rectified linear unit (ReLU) as the activation function for the hidden layers. Most d-vector based system configurations, including context frames (35 previous and 12 upcoming frames), are described in [6].

4.4. Proposed EEEnet

The frame-level layers of EEEnet including the output layer has the same structure as the d-vector based system. The utterance-level layers contain three hidden layers with 512 nodes, and the trial-level layers contains five hidden layers with 1024 nodes. All nodes in the hidden layers are activated by ReLU function. The frame-level layers and the utterance-level layers were pre-trained with thirty epochs. Then the whole network was trained with learning rate of 0.1, batch size of 100, α of 0.1, "drop-out" technique, and approximately 100,000 trials selected from the development set.

4.5. Results

Table 1 shows the results of the baseline and the proposed systems in terms of equal error rate (EER). "D-vector" and "EEEnet" in the table 1 mean the systems introduced in the section 2.1 and 3, respectively. "I-vector + EEEnet" means the system using the i-vectors instead of the d-vectors of the EEEnet as the front-end. This experiment was carried out to compare the performance according to the front-end of the EEEnet. The results showed that the relative error reduction (RER) of the EEEnet over that of the d-vector system was 36.6%, and the RER of the i-vector + EEEnet over that of the i-vector PLDA system was 11.3%.

4.6. Analysis

The d-vector based system among baseline systems shows worse performance than i-vector PLDA system and shows the same tendency with previous studies [6]. EEEnet using d-vector as the front-end shows better performance than pure d-vector system, and EEEnet using i-vector ("I-vector + EEEnet") as the front-end shows higher performance than single i-vector PLDA system. The results show that the back-end expanded with the EEEnet can perform desired operation properly. Therefore, it is expected that the EEEnet may perform well as back-end in LSTM-based experiments not covered in this paper.

5. Conclusions

In this paper, we proposed an expanded end-to-end DNN architecture which embeds the layers that can perform from the second stage to the final stage in speaker verification. In the proposed EEEnet, each layer is effectively joint-trained using the fine-tune cost defined on the basis of speaker verification. The results of the experimental evaluation showed that the RER of the EEEnet over that of the baseline system was up to 36.6%.

The contributions of this paper related to prior works are as follows. In this paper, we embed DNNs, which were partially applied in conventional speaker verification studies, into one architecture. All components of the proposed architecture are joint-trained rather than trained independently as in the conventional systems. This joint-training makes each component more suitable for speaker verification.

6. Future works

In our study, basic experiments were carried out by using database of small size for evaluation of the proposed systems. It is necessary to carry out further studies to expand the proposed architecture and to carry out large scale experiments in the future. The experiments should be carried again with common databases such as RSR [16], and the EEEnet can be reconstructed by replacing the frame-level layers with recurrent layers such as LSTM [6]. Based on the DNN-related research results so far, it can be expected that the EEEnet, DNN-based system, will show excellent performance in large scale experiments by simply adjusting a few parameters. We expect that replacing the front-end of the EEEnet with LSTM will improve the performance, because our experiments show that the EEEnet works well as a back-end.

7. Acknowledgment

This work was supported by the IT R&D program of MOTIE/KEIT. [10041610, The development of the recognition technology for user identity, behavior and location that has a performance approaching recognition rates of 99% on 30 people by using perception sensor network in the real environment]

8. References

[1] Graves, A., and Jaitly, N., "Towards End-To-End Speech Recognition with Recurrent Neural Networks." *International Conference on Machine Learning, Beijing, China*, 2014.

[2] Lei, Y., Scheffer, N., Ferrer, L., and McLaren, M., "A novel scheme for speaker recognition using a phonetically-aware deep neural network," Acoustics, *Speech and Signal Processing (ICASSP)*, pp. 1695-1699, 2014.

[3] Kenny, P., Gupta, V., Stafylakis, T., Ouellet, P., and Alam, J., "Deep neural networks for extracting baum-welch statistics for speaker recognition," *Odyssey: Speaker Lang. Recognit. Workshop*, 2014.

[4] Richardson, F., Reynolds, D., and Dehak, N., "A Unified Deep Neural Network for Speaker and Language Recognition," *Interspeech, Dresden, Germany,* 2015.

[5] Lee, H. S., Tso, Y., Chang, Y. F., Wang, H. M., and Jeng, S. K., "Speaker verification using kernel-based binary classifiers with binary operation derived features," *Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1660-1664, 2014.

[6] Heigold, G., Moreno, I., Bengio, S., and Shazeer, N., "End-to-end text-dependent speaker verification," *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2016.

[7] Chen, Y. H., Lopez-Moreno, I., Sainath, T. N., Visontai, M., Alvarez, R., and Parada, C., "Locally-connected and convolutional neural networks for small footprint speaker recognition," *Interspeech, Dresden, Germany,* 2015.

[8] Dehak, N., Kenny, P. J., Dehak, R., Dumouchel, P., and Ouellet, P., "Front-end factor analysis for speaker verification," *Audio, Speech, and Language Processing*, pp. 788-798, 2011.

[9] Ioffe, S., "Probabilistic linear discriminant analysis," *Computer Vision–ECCV*, pp. 531-542, 2006.

[10] Kenny, P., "Bayesian Speaker Verification with Heavy-Tailed Priors," *Odyssey: Speaker Lang. Recognition Workshop*, 2010.

[11] Campbell, W. M., Sturim, D. E., and Reynolds, D. A., "Support vector machines using GMM supervectors for speaker verification," *IEEE Signal Process. Lett*, vol. 13, no. 5, pp. 308-311, 2006.

[12] He, K., Zhang, X., Ren, S., and Sun, J., "Deep residual learning for image recognition," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[13] Camana, R., "Multitask Learning: A Knowledge-Based Source of Inductive Bias," *Proceedings of the tenth international conference of machine learning*, pp. 41-48, 1993

[14] Glorot, X., and Bengio, Y., "Understanding the difficulty of training deep feedforward neural networks," *In Proceedings of the International Conference on Artificial Intelligence and Statistics* , Vol. 9, pp. 249-256, 2010.

[15] Iizuka, S., Simo-Serra, E., and Ishikawa, H., "Let there be color!: joint end-to-end learning of global and local image priors for automatic image colorization with simultaneous classification," ACM Transactions on Graphics (TOG), (2016).

[16] Larcher, A., Lee, K. A., Ma, B., and Li, H., "The RSR2015: database for text-dependent speaker verification using multiple pass-phrases," *Proceedings of Interspeech*, Portland (Oregon), USA, 2012.

[17] Bastien, F., Lamblin, P., Pascanu, R., Bergstra, J., Goodfellow, I., Bergeron, A., Bouchard, N., Warde-Farley, D., and Bengio, Y., "Theano: new features and speed improvements," *NIPS deep learning workshop, 2012*.

[18] James, B., Olivier, B., Frédéric, B., Pascal, L., and Razvan, P., "Theano: A CPU and GPU Math Expression Compiler," *Proceedings of the Python for Scientific Computing Conference (SciPy)*, 2010.

[19] Povey, D., Ghoshal, A., Boulianne, G., Burget, L., Glembek, O., Goel, N., Hannemann, M., Motlicek, P., Qian, Y., and Silovsky, J., "The kaldi speech recognition toolkit," *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding*, 2011.