Deep Discriminative Embeddings for Duration Robust Speaker Verification

Na Li, Deyi Tuo, Dan Su, Zhifeng Li, and Dong Yu

Tencent AI lab

lina011779@126.com,{deyituo,dansu,michaelzfli,dyu}@tencent.com

Abstract

The embedding-based deep convolution neural networks (CNNs) have demonstrated effective for text-independent speaker verification systems with short utterances. However, the duration robustness of the existing deep CNNs based algorithms has not been investigated when dealing with utterances of arbitrary duration. To improve robustness of embedding-based deep CNNs for longer duration utterances, we propose a novel algorithm to learn more discriminative utterance-level embeddings based on the Inception-ResNet speaker classifier. Specifically, the discriminability of embeddings is enhanced by reducing intra-speaker variation with center loss, and simultaneously increasing inter-speaker discrepancy with softmax loss. To further improve system performance when long utterances are available, at test stage long utterances are segmented into shorter ones, where utterance-level speaker embeddings are extracted by an average pooling layer. Experimental results show that when cosine distance is employed as the measure of similarity for a trial, the proposed method outperforms ivector/PLDA framework for short utterances and is effective for long utterances.

Index Terms: deep convolution, speaker embedding, i-vector, center loss, duration

1. Introduction

Although the i-vector/PLDA [1–3] framework performs well when long utterances are available, it suffers performance reduction when handling short utterances. The problem of duration variability in utterances has attracted attention in the community because an i-vector extracted from a short utterance should not be treated as being equally reliable as an i-vector extracted from a long utterance. The reason is that the posterior distribution of hidden variables in the i-vector extractor is a Gaussian whose covariance matrix is related to the utterance duration. The shorter the utterance is, the larger the covariance will become, leading to greater uncertainty in the estimated i-vector.

The issue of duration variability has been addressed to a certain extent in the past. For example, by propagating the uncertainty arising from the i-vector extraction process into a PLDA model, the resulting PLDA model better handled the duration variability than the conventional PLDA model [4]. Recently there have been some efforts to replace i-vector with speaker embedding learned from deep neural networks [5–8]. These approaches outperform i-vector/PLDA framework in text-dependent or short-duration text-independent tasks. In [9], the speaker embeddings were created by averaging bottle-neck layer activations of a feed-forward DNN which was trained to classify speakers at the frame-level. [7] further demonstrates that utterance-level embdedding is more reliable than frame-level representation. Based on triplet loss, deep convolutional neural network (CNN) based architectures were employed to learn speaker embeddings for short utterances in [9, 10]. However, compared to training samples, the number of training pairs or triplets dramatically grows. This inevitably results in slow convergence and instability. Work in [11] also exhibits better performance compared to i-vector/PLDA framework for short utterances by training a deep CNN similar to VGG net with softmax loss. Though these works achieved promising performance for short-duration speaker verification, the investigation of robustness to duration variant is still limited. An attractive exception is the work in [12]. The utterance-level speaker embeddings were obtained by using a statistics pooling layer to aggregate the frame-level inputs. However, this method needs large amount of speech segments covering different durations that may be met in test stage, causing much inconvenience in preparing training set.

Inspired by recent progress in face recognition and short-duration speaker verification, this work proposes an Inception-ResNet [13, 14] based architecture to learn deep speaker discriminative embedding for utterances of variable duration. Unlike [10], where Inception-ResNet-v1 [14] with triplet loss was employed to learn speaker embeddings, we modify the stem component of Inception-ResNet-v1 according to the characteristics of input speech. In the training stage, speech segments of short duration were used as inputs and the deep CNN network were trained to learn segment-level speaker embeddings by optimizing softmax loss and center loss simultaneously. With the joint supervision of softmax loss and center loss, the intra-speaker variability is suppressed and the inter-speaker variability is emphasized in the embedding space. Given the trained network, the utterance-level embedding for an utterance of variable duration can be obtained via an average pooling layer.

Compared to the state-of-the-art approaches, the contributions of this work include: (1) apply deep convolutional network to text-independent speaker verification for utterances of variable durations, (2) learn deep discriminative embeddings by training the network with joint supervision of softmax loss and center loss, the resulting embeddings have less intra-speaker variability compared to those extracted from a deep speaker classifier with softmax loss function, and (3) avoid suffering from dramatic data expansion when constituting sample triplets from the training set, leading to fast convergence.

The remainder of this paper is organized as follows: Section 2 introduces the scheme of our proposed system. Experimental results and discussions are described in Section 3. Finally, we make our conclusions in Section 4.

2. Deep Discriminative Embedding based Speaker Verification

We argue that segment-level speaker embedding is much more crucial to speaker verification compared to frame-level embeddings. So we aim to use speech segments as network inputs in this work. Since Inception-ResNet architecture has demon-

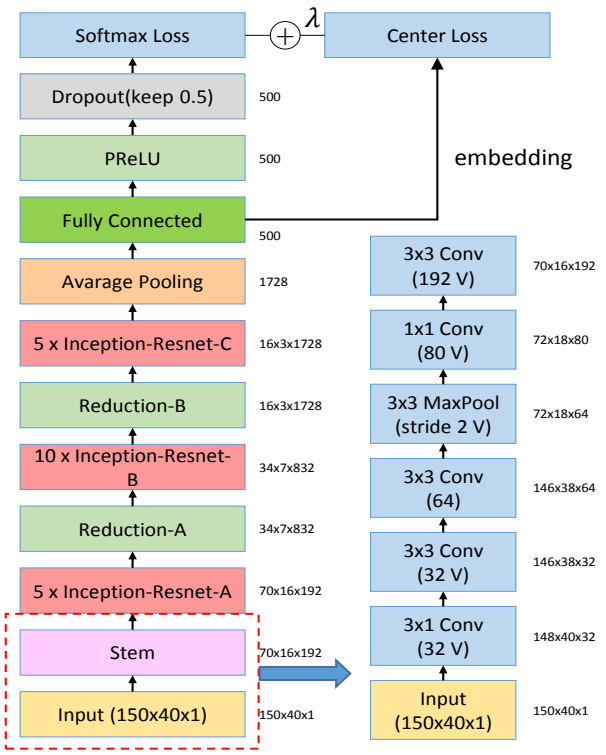

Figure 1: *Architecture of deep convolutional network. The left is the overall schema, the right details the Stem block. V denotes 'Valid' padding.*

strated excellent performance in face recognition, we settle on Inception-ResNet based architecture. The following details the deep convolutional network, the loss function, and verification process.

2.1. Deep Convolutional Architecture

We made modifications to Inception-ResNet-v1 [14] considering the characteristics of our inputs. Specifically, we use unsymmetrical convolutional kernels with 1 stride in the first convolutional layer. More aggressively convolution is performed along the duration direction as it is much larger than the feature dimension. In addition, we employ convolutional layer with 1 stride at the top of stem block in Inception-ResNet-v1. While Inception-ResNet architecture are retained in our work. The overall schema of the deep convolutional network and the input/output size of each module are shown in Fig 1.

2.1.1. Loss Function

The total loss is a combination of softmax loss and center loss [15]. The softmax loss is defined as:

$$\mathcal{L}_s = -\sum_{i=1}^{M} \log \frac{e^{W_{y_i}^{\top}\mathbf{x}_i + b_{y_i}}}{\sum_{j=1}^{N} e^{W_j^{\top}\mathbf{x}_i + b_j}}, \tag{1}$$

where $\mathbf{x}_i \in \mathbb{R}^d$ denotes the i-th deep embedding, belonging to the y_i-th speaker. d is the feature dimension. $W_j \in \mathbb{R}^d$ denotes the j-th column of the weights $W \in \mathbb{R}^{d \times N}$ in the last fully connected layer and $\mathbf{b} \in \mathbb{R}^N$ is the bias term. The size of mini-batch and the number of speakers is M and N, respectively.

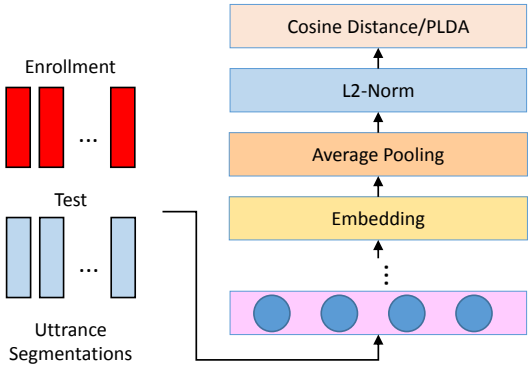

Figure 2: *Schematic diagram of verification process.*

The center loss is defined as:

$$\mathcal{L}_c = \frac{1}{2}\sum_{i=1}^{M} \|\mathbf{x}_i - \mu_{y_i}\|_2^2, \tag{2}$$

where $\mu_{y_i} \in \mathbb{R}^d$ denotes the center of deep embeddings from speaker y_i. Particularly, the centers are updated based on mini-batch. The centers are estimated by averaging the embeddings from the corresponding speakers in each iteration step. The centers are updated according to Eq. 3 and Eq. 4, shown as follows:

$$\mu_j^{t+1} = \mu_j^t - \alpha\Delta\mu_j^t, \tag{3}$$

and

$$\Delta\mu_j = \frac{\sum_{i=1}^{M} \delta(y_i = j)(\mu_j - \mathbf{x}_i)}{1 + \sum_{i=1}^{M} \delta(y_i = j)}, \tag{4}$$

where the scalar α is used to control the learning rate of the centers, and $\Delta\mu_j$ is the gradient of $\mathcal{L}_c$ with respect to $\mathbf{x}_i$. If the condition is satisfied, we have $\delta = 1$, otherwise, $\delta = 0$. The range of α is $[0, 1]$.

Finally, softmax loss and center loss are combined together by a weight λ to construct the total loss, shown as:

$$\begin{aligned}\mathcal{L}_t &= \mathcal{L}_s + \lambda\mathcal{L}_c \\ &= -\sum_{i=1}^{M} \log \frac{e^{W_{y_i}^{\top}\mathbf{x}_i + b_{y_i}}}{\sum_{j=1}^{N} e^{W_j^{\top}\mathbf{x}_i + b_j}} + \frac{\lambda}{2}\Sigma_{i=1}^{M}\|\mathbf{x}_i - \mu_{y_i}\|_2^2.\end{aligned} \tag{5}$$

2.2. Verification

The verification process is illustrated in Fig 2. Given the trained network, the utterance-level embeddings of the enrollment and test utterances are extracted from the L2-norm layer. Specifically, if the duration of an utterance is shorter than the duration of input segments utilized in the training stage, we will pad some frames to the short utterance. Otherwise, we divide the long utterance into multiple short segments by employing a sliding-window without overlap. Then the utterance-level speaker embedding is obtained by performing averaging pooling followed by L2 normalization as shown in Fig 2. After extracting the utterance-level speaker embeddings, cosine distance or PLDA is adopted as the scoring method.

Table 1: *Configurations of different networks. As 40-dimensional log mel-filter bank features are too small to train Inception-ResNet-v1 [14] smoothly, we use 120-dimensional features consisting of 40-dimensional log mel-filter bank features and their first and second derivatives to train Net1, Net2, and Net3.*

Network Name	Architecture	Loss Type	Fea-Dim	Input Duration(s)/Size
Net1	Inception-ResNet-v1	Softmax+Center	120	1.5/150x120
Net2	Inception-ResNet-v1	Softmax+Center	120	2.0/200x120
Net3	Inception-ResNet-v1	Softmax+Center	120	2.5/250x120
Net4	Proposed	Softmax	40	1.5/150x40
Net5	Proposed	Softmax+Center	40	1.5/150x40

3. Experiments and Results

This section details our experimental setup, results and analysis. The system performance is measured in terms of equal error rate (EER).

3.1. Speech Data and Front-end Processing

Experiments were performed on a large collection of speakers from Android cellphones. The corpus consists of about 760,220 utterances from 2,500 Chinese speakers, each speaker has 300 utterances. Most of the data set are short utterances with mean duration of 2.6s. We divided the speech data into three categories: (1) training set, (2) validation set, and (3) evaluation set.

- *Training Set*: This data set was employed to train the neural networks, i-vector extractor, and PLDA models. It includes 2,000 speakers, each speaker has 295 utterances. For training the deep network, we extracted one speech segment of fixed-duration from each utterance. This amounts to 590,000 training segments in total.
- *Validation Set*: The validation set comprises 10,000 utterances from the same 2,000 speakers as in training data. Each speaker has 5 utterances.
- *Evaluation Set*: All of the utterances from the other 500 speakers were used as valuation set for the system performance evaluation. For each speaker, 3 utterances were sampled as the enrollment data. Other than the enrollment data, we sampled 25 utterances from each speaker and 800 utterances from other speakers. This resulted in 12,500 target trials and 400,000 impostor trials in total. For different experiments, we sampled the corresponding utterances meeting the specific requirement for duration.

For each speech utterance, a VAD [16, 17] was applied to prune out silence regions. Then the speech regions were segmented into 25-ms Hamming windowed frames with 10-ms frame shift. For i-vector/PLDA systems, the first 19 Mel frequency cepstral coefficients (MFCC) with log energy were calculated with their first and second derivatives to form a 60-dimensional acoustic vector, followed by cepstral mean normalization. While the neural networks in this work were trained on 40-dimensional log mel-filter bank features. As a comparison, we also used 120-dimensional features consisting of 40-dimensional log mel-filter bank features and their first and second derivatives to train neural networks in some experiments.

3.2. I-vector Baseline

The i-vector extractor is based on a phonetically-aware DNN with 2786 output nodes and a gender-independent total variability matrix with 500 total factors. Similar to [18], we applied within-class covariance normalization (WCCN) [19] and i-vector length normalization (LN) [20] to the 500-dimensional i-vectors. Then, linear discriminant analysis (LDA) [21] and WCCN were used to further reduce intra-speaker variability and reduce the dimension to 200. Then PLDA models with 150 latent identity factors were trained. Note that the total variability matrix was trained using long utterances.

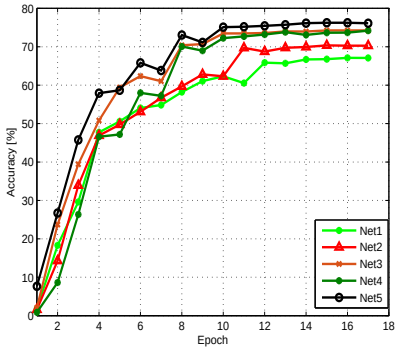

Figure 3: *Accuracy of different networks on validation data set at varying epoches. The same training strategy was applied to each neural network (e.g., initial learning rate, optimizer, dropout and batch normalization).*

3.3. Network Training

The proposed deep neural network was trained under joint supervision of softmax loss and center loss. We employed the RMSProp [22] optimizer with an initial learning rate of 0.1 in all our networks. The learning rate was decayed based on validation set performance. To accelerating the training process, batch normalization and dropout were employed during training stage. The batch size was set to 128. The values of λ in Eq. 5 and α in Eq. 3 were set to 0.001 and 0.2, respectively.

3.4. Performance Comparison between Different Networks on Validation Set

To validate our deep convolutional architecture, we compare the performance of Inception-ResNet-v1 and our modified version under different conditions. The details about each system are listed in Table 1. Three Inception-ResNet-v1 networks with the same setup as in [14] were trained with different input sizes. The accuracy of each network on validation data set is shown in Fig 3. Considering the first three networks, we can see that the network performs better when it sees a larger image, this finding agrees with the results in [10, 11]. Compared to Net3 which is trained using 2.5s speech segments, our proposed Net5 achieves higher accuracy when 40-dimensional log-mel features extract-

ed from only 1.5s speech segments were used as inputs. One possible reason is that our modification to Inception-ResNet-v1 is suitable for such input speech segments. Another observation is that Net5 outperforms Net4 at almost each epoch, this should be due to the effect of center loss. To balance of the flexibility of the network in handling utterances of arbitrary duration and the burden of training multiple duration-dependent networks, we fix the input size to 150×40 in the following experiments.

3.5. Short Enrollment versus Short Test

In this section, we focus on the condition where both the enrollment and test utterances of a verification trial are short. Only short utterances ($< 10s$) were selected as the valuation set. One speech segment of 1.5s was extracted from the speech region of each enrollment or test utterance, the corresponding embedding obtained from the proposed deep network was utilized as the utterance-level speaker embedding. 'i-vector-1.5s' in Table 2 denotes that both the enrollment and test i-vectors were extracted from the same set of extracted speech segments. While 'i-vector' denotes the full utterance duration i-vector. When PLDA models were trained using embeddings, we did not perform any pre-processing. Because we found that if we trained the PLDA model with the same pipelines as that used for training PLDA with i-vectors, the performance of embedding based system degraded a lot. [11] also found the similar issue.

The results in Table 2 show that the embedding based systems outperform i-vector based ones. I-vectors of full utterance duration is more reliable compared to i-vectors extracted from speech segments of only 1.5s. Compared the results of PLDA, we can see that PLDA could improve the performance of i-vector/PLDA systems a lot.

Table 2: *Performance of i-vector and embedding based systems in terms of EER(%), short-short. CD denotes cosine distance measurement.*

Method	Feature Representation		
	Proposed	i-vector-1.5s	i-vector
CD	2.10	8.25	4.88
PLDA	1.96	5.09	2.76

3.6. Long Enrollment versus Short Test

The experiments in this section investigate the system performance when enrollment utterance is long and the test utterance is short. Utterances longer than 15s were sampled as enrollment data, and only utterances of short duration were used as test utterances in verification trials. The average duration of the enrollment and test utterances is about 20s and 2s, respectively. The PLDA in i-vector/PLDA system was trained using long utterances, as we found that PLDA trained using short utterances degraded the system performance.

From Table 3, we can see that the best performance is still achieved by embedding/PLDA approach. The performance of i-vector/PLDA on full duration i-vector is better than that on i-vectors extracted from speech segments of 1.5s. PLDA only improves the performance of embedding approach just a little. The performance of embedding approach with cosine distance measurement is comparable to that of i-vector/PLDA system.

Table 3: *Performance of i-vector and embedding based systems in terms of EER(%), long-short. CD denotes cosine distance measurement.*

Method	Feature Representation		
	Proposed	i-vector-1.5s	i-vector
CD	2.32	4.70	2.80
PLDA	2.27	3.49	2.43

3.7. Long Enrollment versus Long Test

The experiments in this section compare the proposed approach with i-vector/PLDA system when both enrollment and test utterances are long. Utterances longer than 15s were sampled as evaluation set. The average duration of the evaluation set is about 20s. Utterance-level speaker embedding was extracted from each utterance in the evaluation set according to the strategy described in Section 2.2. The PLDA models were trained using long utterances.

Compared to the above experiments, both embedding and i-vector based systems achieve significant performance improvement. For speaker embeddings, PLDA degrades the system performance. Though i-vector based systems outperform deep embedding based systems, the embedding system's performance is still promising. Results also indicate that our proposed utterance-level speaker embedding is a duration robust feature representation for speaker verification.

Table 4: *Performance of i-vector and embedding based systems in terms of EER(%), long-long. CD denotes cosine distance measurement.*

Method	Feature Representation	
	Proposed	i-vector
CD	0.46	0.19
PLDA	0.52	0.11

4. Conclusions

A duration robust deep convolutional network based text-independent speaker verification system is presented. It is designed to improve the robustness of speaker verification systems when the enrollment or test utterances exhibit a wide range of duration. By using joint supervision of softmax loss and center loss to train a modified Inception ResNet architecture, the deep discriminative speaker embedding is learned. The utterance-level speaker embedding for an utterance of variable-duration is extracted by averaging multiple embeddings of short segments. Experiments on a large data set demonstrate the effectiveness of the proposed method.

5. References

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

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

[3] P. Kenny, "Bayesian speaker verification with heavy-tailed priors," in *Proc. of Odyssey: Speaker and Language Recognition Workshop*, Brno, Czech Republic, June 2010.

[4] P. Kenny, T. Stafylakis, P. Ouellet, M. J. Alam, and P. Dumouchel, "PLDA for speaker verification with utterances of arbitrary duration," in *Proc. ICASSP*. IEEE, 2013, pp. 7649–7653.

[5] 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, pp. 4052–4056.

[6] S.-X. Zhang, Z. Chen, Y. Zhao, J. Li, and Y. Gong, "End-to-end attention based text-dependent speaker verification," in *Proc. SLT*, 2016, pp. 171–178.

[7] G. Heigold, I. Moreno, S. Bengio, and N. Shazeer, "End-to-end text-dependent speaker verification," in *Proc. ICASSP*, 2016, pp. 5115–5119.

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

[9] C. Li, X. Ma, B. Jiang, X. Li, X. Zhang, X. Liu, Y. Cao, A. Kannan, and Z. Zhu, "Deep speaker: an end-to-end neural speaker embedding system," *arXiv preprint arXiv:1705.02304*, 2017.

[10] C. Zhang and K. Koishida, "End-to-end text-independent speaker verification with triplet loss on short utterances," in *Proc. of Interspeech*, 2017.

[11] G. Bhattacharya, J. Alam, and P. Kenny, "Deep speaker embeddings for short-duration speaker verification," in *Proc. Interspeech*, 2017, pp. 1517–1521.

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

[13] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the Inception architecture for computer vision," in *Proc. ICCV*, 2016, pp. 2818–2826.

[14] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, "Inception-v4, Inception-ResNet and the impact of residual connections on learning." in *Proc. AAAI*, 2017, pp. 4278–4284.

[15] Y. Wen, K. Zhang, Z. Li, and Y. Qiao, "A discriminative feature learning approach for deep face recognition," in *Proc. ECCV*. Springer, 2016, pp. 499–515.

[16] M. W. Mak and H. B. Yu, "A study of voice activity detection techniques for NIST speaker recognition evaluations," *Computer, Speech and Language*, vol. 28, no. 1, pp. 295–313, Jan 2014.

[17] H. Yu and M. Mak, "Comparison of voice activity detectors for interview speech in NIST speaker recognition evaluation," in *Proc. of Interspeech*, 2011, pp. 2353–2356.

[18] M. McLaren, M. Mandasari, and D. Leeuwen, "Source normalization for language-independent speaker recognition using i-vectors," in *Odyssey 2012: The Speaker and Language Recognition Workshop*, 2012, pp. 55–61.

[19] A. Hatch, S. Kajarekar, and A. Stolcke, "Within-class covariance normalization for SVM-based speaker recognition," in *Proc. of the 9th International Conference on Spoken Language Processing*, Pittsburgh, PA, USA, Sep. 2006, pp. 1471–1474.

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

[21] C. Bishop, *Pattern Recognition and Machine Learning*. New York: Springer, 2006.

[22] T. Tieleman and G. Hinton, "Lecture 6.5 - RMSProp," Tech. Rep., 2012.