DEEP NEURAL NETWORKS FOR SMALL FOOTPRINT TEXT-DEPENDENT SPEAKER VERIFICATION

Ehsan Variani[1*], *Xin Lei*[2], *Erik McDermott*[2], *Ignacio Lopez Moreno*[2], *Javier Gonzalez-Dominguez*[2,3]

[1]Johns Hopkins Univ., Baltimore, MD USA
[2]Google Inc., USA
[3]ATVS-Biometric Recognition Group, Universidad Autonoma de Madrid, Spain

variani@jhu.edu {xinlei,erikmcd,elnota,jgd}@google.com

ABSTRACT

In this paper we investigate the use of deep neural networks (DNNs) for a small footprint text-dependent speaker verification task. At development stage, a DNN is trained to classify speakers at the frame-level. During speaker enrollment, the trained DNN is used to extract speaker specific features from the last hidden layer. The average of these speaker features, or d-vector, is taken as the speaker model. At evaluation stage, a d-vector is extracted for each utterance and compared to the enrolled speaker model to make a verification decision. Experimental results show the DNN based speaker verification system achieves good performance compared to a popular i-vector system on a small footprint text-dependent speaker verification task. In addition, the DNN based system is more robust to additive noise and outperforms the i-vector system at low False Rejection operating points. Finally the combined system outperforms the i-vector system by 14% and 25% relative in equal error rate (EER) for clean and noisy conditions respectively.

Index Terms— Deep neural networks, speaker verification.

1. INTRODUCTION

Speaker verification (SV) is the task of accepting or rejecting the identity claim of a speaker based on the information from his/her speech signal. Based on the text to be spoken, the SV systems can be classified into two categories, text-dependent and text-independent. Text-dependent SV systems require the speech to be produced from a fixed or prompted text phrase, while the text-independent SV systems operate on unconstrained speech. In this paper, we focus on a small footprint text-dependent SV task using fixed-text, although the proposed technique may be extended to text-independent tasks.

The SV process can be divided into three phases:

- Development: background models are trained from a large collection of data to define the speaker manifold. Background models vary from simple Gaussian mixture model (GMM) based Universal Background Models (UBMs) [1] to more sophisticated Joint Factor Analysis (JFA) based models [2, 3, 4].
- Enrollment: new speakers are enrolled by deriving speaker specific information to obtain speaker-dependent models. Speakers in the enrollment and development sets are not overlapped.
- Evaluation: each test utterance is evaluated using the enrolled speaker models and background models. A decision is made on the identity claim.

A wide variety of SV systems have been studied using different statistical tools for each of the three phases in verification. The state-of-the-art SV systems are based on i-vectors [5] and Probabilistic Linear Discriminant Analysis (PLDA). In these systems, JFA is used as a feature extractor to extract a low-dimensional i-vector as the compact representation of a speech utterance for SV.

Motivated by the powerful feature extraction capability and recent success of deep neural networks (DNNs) applied to speech recognition [6], we propose a SV technique based on DNN as the speaker feature extractor. A new type of DNN-based background model is used to directly model the speaker space. A DNN is trained to map frame-level features in a given context to the corresponding speaker identity target. During enrollment, the speaker model is computed as the average of activations derived from the last DNN hidden layer, which we refer to as a *deep vector* or "*d-vector*". In the evaluation phase, we make decisions using the distance between the target d-vector and the test d-vector, similar to i-vector SV systems. One significant advantage of using DNNs for SV is that it is easy to integrate them into a state-of-the-art speech recognition system since they can share the same DNN inference engine and simple filterbank energies frontend.

The rest of this paper is organized as follows. In Section 2, previous related work on SV is described. In Section 3 we describe the proposed DNN-based SV system. Section 4 shows the experimental results for a small footprint text-dependent SV system. The DNN-based SV system is compared with an i-vector system in both clean and noisy conditions. We also evaluate the performance with different numbers of enrollment utterances and describe improvements from combination of two systems. Finally, Section 5 concludes the paper and discusses future work.

2. PREVIOUS WORK

The combination of i-vector and PLDA [5, 7] has become the dominant approach for text-independent speaker recognition. The i-vector represents an utterance in a low-dimensional space named total variability space. Given an utterance, the speaker- and session-dependent GMM supervector is defined as follows:

$$M = m + Tw \tag{1}$$

where m is the speaker- and session-independent supervector, usually taken to be the UBM supervector, T is a rectangular matrix of low rank, referred to as the total variability matrix (TVM), and w is a random vector with a standard normal distribution $N(0, I)$. The vector w contains the total factors and is referred to as the i-vector.

*Research conducted as an intern at Google.

978-1-4799-2893-4/14/$31.00 ©2014 IEEE

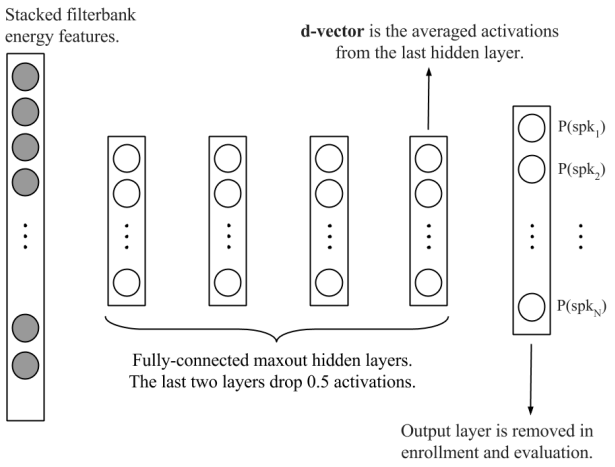

Fig. 1. The background DNN model for speaker verification.

Moreover, the PLDA on the i-vectors can decompose the total variability into speaker and session variability more effectively compared to JFA. The i-vector-PLDA technique and its variants have also been successfully used in text-dependent speaker recognition tasks [8, 9, 10].

In past studies, neural networks have been investigated for speaker recognition [11, 12]. Being nonlinear classifiers, neural networks can discriminate the characteristics of different speakers. The neural network was typically used as a binary classifier of target and non-target speakers, or multicategory classifiers for speaker identification purposes. Auto-associative neural networks (AANN) [13] were proposed to use the reconstruction error difference computed from the UBM-AANN and speaker specific AANN as the verification score. Multi-layer perceptrons (MLPs) with a bottleneck layer have been used to derive robust features for speaker recognition [14]. More recently, some preliminary studies have been conducted on using deep learning for speaker recognition, such as the use of convolutional deep belief networks [15] and Boltzmann machine classifiers [16].

3. DNN FOR SPEAKER VERIFICATION

The proposed background DNN model for SV is depicted in Figure 1. The idea is similar to [15] in the sense that neural networks are used to learn speaker specific features. The main differences are that here we perform supervised training, and use DNNs instead of convolutional neural networks. In addition, in this paper we evaluate on a SV task instead of the simpler speaker identification task.

3.1. DNN as a feature extractor

At the heart of the proposed approach in this work is the idea of using a DNN architecture as a speaker feature extractor. As in the i-vector approach, we look for a more abstract and compact representation of the speaker acoustic frames but using a DNN rather than a generative Factor Analysis model.

With this aim, we first built a supervised DNN, operating at the frame level, to classify the speakers in the development set. The input of this background network is formed by stacking each training frame with its left and right context frames. The number of outputs corresponds to the number of speakers in the development set, N. The target labels are formed as a 1-hot N-dimensional vector where the only non-zero component is the one corresponding to the speaker identity. Figure 1 illustrates the DNN topology.

Once the DNN has been trained successfully, we use the accumulated output activations of the last hidden layer as a new speaker representation. That is, for every frame of a given utterance belonging to a new speaker, we compute the output activations of the last hidden layer using standard feedforward propagation in the trained DNN, and then accumulate those activations to form a new compact representation of that speaker, the *d-vector*. We choose to use the output from the last hidden layer instead of the softmax output layer due to a couple of reasons. First, we can reduce the DNN model size for runtime by pruning away the output layer, and this also enables us to use a large number of development speakers without increasing DNN size at runtime. Second, we have observed better generalization to unseen speakers from the last hidden layer output.

The underlying hypothesis here is that the trained DNN, having learned compact representations of the development set speakers in the output of the last hidden layer, may also be able to represent unseen speakers.

3.2. Enrollment and evaluation

Given a set of utterances $X_s = \{O_{s_1}, O_{s_2}, \ldots, O_{s_n}\}$ from a speaker s, with observations $O_{s_i} = \{o_1, o_2, \ldots, o_m\}$, the process of enrollment can be described as follows. First, we use every observation o_j in utterance O_{s_i}, together with its context, to feed the supervised trained DNN. The output of the last hidden layer is then obtained, $L2$ normalized, and accumulated for all the observations o_j in O_{s_i}. We refer to the resulting accumulated vector as the *d-vector* associated with the utterance O_{s_i}. The final representation of the speaker s is derived by averaging all d-vectors corresponding for utterances in X_s.

During the evaluation phase, we first extract the normalized d-vector from the test utterance. Then we compute the cosine distance between the test d-vector and the claimed speaker's d-vector. A verification decision is made by comparing the distance to a threshold.

3.3. DNN training procedure

Given the low-resource conditions of the scenario explored in this study (see Section 4), we trained the background DNN as a *maxout* DNN using *dropout* [17][18].

Dropout is a useful strategy to prevent over-fitting in DNN fine-tuning when using a small training set [18][19]. In essence, the dropout training procedure consists of randomly omitting certain hidden units for each training token. Maxout DNNs [17] were conceived to properly exploit dropout properties. Maxout networks differ from the standard multi-layer perceptron (MLP) in that hidden units at each layer are divided into non-overlapping groups. Each group generates a single activation via the max pooling operation. Training of maxout networks can optimize the activation function for each unit.

Specifically, in this study, we trained a maxout DNN with four hidden layers and 256 nodes per layer, within the DistBelief framework [20]. A pool size of 2 is used per layer. The first two layers do not use dropout while the last two layers drop 50 percent of activations after dropout, as shown in Figure 1.

Regarding other configuration parameters, we used rectified linear units [21] as the non-linear activation function on hidden units and a learning rate of 0.001 with exponential decay (0.1 every

$5M$ steps). The input of the DNN is formed by stacking the 40-dimensional log filterbank energy features extracted from a given frame, together with its context, 30 frames to the left and 10 frames to the right. The dimension of the training target vectors is 496, which is the same as the number of speakers in the development set (see Section 4). The final maxout DNN model contains about $600K$ parameters, which is similar to the smallest baseline i-vector system.

4. EXPERIMENTAL RESULTS

The experiments are performed on a small footprint text-dependent SV task. The data set contains 646 speakers speaking the same phrase, "ok google", many times in multiple sessions. The gender distribution is balanced on the data set. 496 randomly selected speakers are used for training the background model and the remaining 150 speakers were used for enrollment and evaluation. The number of utterances per speaker for background model training varies from 60 to 130. For the enrollment speakers, the first 20 utterances are reserved for possible use in enrollment and the remaining utterances are used for evaluation. By default, we only use the first 4 utterances of the enrollment set for extracting speaker models. We used one out of 150 trials as a target trial and there are approximately 12750 trials in total.

4.1. Baseline system

In this small footprint text-dependent SV task, we aim to keep the model size small while achieving good performance. The baseline system is an i-vector based SV system similar to [5]. The GMM UBM is trained on 13-dimensional perceptual linear predictive (PLP) features with Δ and $\Delta\Delta$ features appended. We evaluate the equal error rate (EER) performance of the i-vector system with three different model sizes. The number of Gaussian components in the UBM, the dimension of the i-vectors and the dimension of Linear Discriminant Analysis (LDA) output are varied. The TVM is initialized using PCA and further refined using 10 EM iterations, while for UBM training we used 7 EM iterations. As shown in Table 1, the i-vector system performance degrades with reduced model size but not too significantly. The EER results with t-norm [22] for score normalization are consistently much better than with the raw scores. The smallest i-vector system contains about $540K$ parameters and is used as our baseline system.

Table 1. Comparison of EER results of i-vector systems with different number of UBM Gaussian components, i-vector and LDA output dimensions.

#Gaussians	i-vector Dim	LDA Dim	#Params	EER (raw)	EER (t-norm)
1024	300	200	12.2M	2.92%	2.29%
256	200	100	2.1M	3.11%	2.92%
128	100	100	540K	3.50%	2.83%

4.2. DNN verification system

The left plot in Figure 2 shows the detection error tradeoff (DET) curve comparison of the i-vector system and d-vector system. One interesting finding is that in the d-vector system the raw scores are slightly better than the t-norm scores, whereas in the i-vector system the t-norm scores are significantly better. The histogram analysis of the raw scores of the d-vector system indicates the distribution is heavy-tailed instead of a normal distribution. This suggests more sophisticated score normalization methods may be necessary for the d-vector SV system. Moreover, since t-norm requires extra storage and computation at runtime, we evaluate the d-vector systems using raw scores for the following experiments unless specified.

The overall performance of the i-vector system is better than the d-vector system: 2.83% EER using i-vector t-norm scores versus 4.54% with d-vector raw scores. However, in low False Rejection regions, as shown in right bottom part of the plots in Figure 2, the d-vector system outperforms the i-vector system.

We also experiment with different configurations for DNN training. Without maxout and dropout techniques, the EER of the trained DNN is about 2% absolute worse. Increasing the number of nodes to 512 in the hidden layers does not help significantly, while reducing the number of nodes to 128 gives much worse EER at 7.0%. Reducing the context window size to 10 frames on the left and 5 frames on the right also degrades the EER performance to 5.67%.

4.3. Effect of enrollment data

In d-vector SV system, there are no speaker adaptation statistics involved in the enrollment phase. Instead, the background DNN model is used to extract speaker-specific features for each utterance in both enrollment and evaluation phases. In this experiment we investigate how much the verification performance changes in the d-vector system with different numbers of enrollment utterances per speaker. We compare the performance results using 4, 8, 12 and 20 utterances for speaker enrollment.

Table 2. EER results of i-vector and d-vector verification systems using different number of utterances for enrollment.

	# utterances in enrollment			
	4	8	12	20
i-vector	2.83%	2.06%	1.64%	1.21%
d-vector	4.54%	3.21%	2.64%	2.00%

The EER results are listed in Table 2. It shows that both SV systems perform better with increasing numbers of enrollment utterances. The trend is similar for both systems.

4.4. Noise robustness

In practice there is usually a mismatch between development and runtime conditions. In this experiment, we examine the robustness of the d-vector SV system in noisy conditions and compare it with the i-vector system. The background models are trained with clean data. 10 dB cafeteria noise is added to the enrollment and evaluation data. The comparison of DET curves are shown in the right plot in Figure 2. As this figure illustrates, the performance of both systems is degraded by noise, but the performance loss of the d-vector system is smaller. Under 10 dB noisy environment, the overall performance of the d-vector system is very close to the i-vector system. At operating points of 2% or lower False Rejection probability, the d-vector system is in fact better than the i-vector system.

4.5. System combination

The results above show that the proposed d-vector system can be a viable SV approach when compared to the i-vector system. The assessment holds true mostly for noisy environments, or applications that require small footprint model and low False Rejection rates. Alternatively, here we aim to provide an analysis of a combined i/d-vector system. Although more sophisticated combinations can be

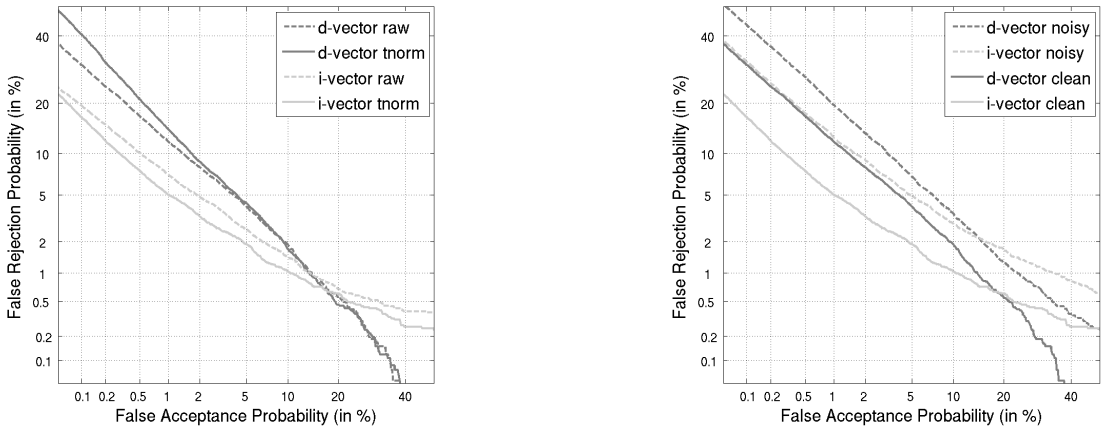

Fig. 2. Left: DET curve comparison between i-vector and d-vector speaker verification systems using raw and t-norm scores. Right: DET curve comparison of the two systems in clean and noisy conditions.

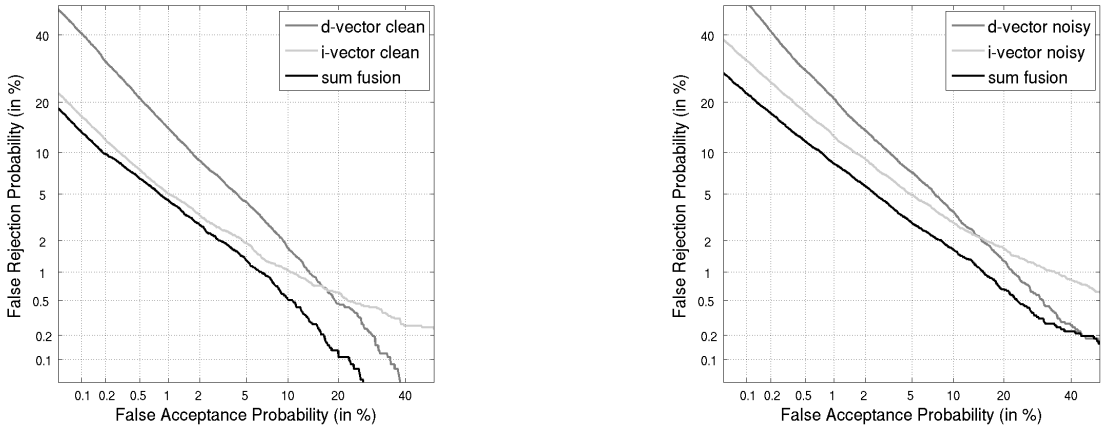

Fig. 3. DET curve for the sum fusion of the i-vector and d-vector systems in clean (left) and noisy (right) conditions.

devised at the feature level, our preliminary results in Figure 3 are obtained using a simple combination named as *sum fusion*, which sums the scores provided by each individual system for each trial. A prior t-norm stage was applied in both systems to facilitate the combination of scores. Results show that the combined system outperforms either component system in essentially all possible operating points and noise conditions. In terms of EER performance, the i/d-vector system beats the i-vector system by 14% and 25% relative, in clean and noisy conditions respectively.

5. CONCLUSIONS

In this paper we have proposed a new DNN based speaker verification method for a small footprint text-dependent speaker verification task. DNNs are trained to classify speakers with frame-level acoustic features. The trained DNN is used to extract speaker specific features. The average of these speaker features, or d-vector, is then used for speaker verification similarly to the popular i-vector. Experimental results show that the performance of the d-vector SV system is reasonably good compared to an i-vector system, and system fusion achieves much better results than the standalone i-vector system. A simple sum fusion of these two systems can improve the i-vector system performance in all operating points. The EER of the combined system is 14% and 25% better than our classical i-vector system in clean and noisy conditions respectively. Furthermore, the d-vector system is more robust to additive noise in enrollment and evaluation data. At low False Rejection operating points, the d-vector system outperforms the i-vector system.

Future work includes improving the current cosine distance scoring, as well as trying normalization schemes such as Gaussianization for the raw scores. We will explore different combination approaches, such as using a PLDA model over the the feature space of the i-vectors and d-vectors stacked. Finally, we aim to investigate the effect of increasing the number of development speakers and how speaker clustering affects performance.

Acknowledgments

The authors would like to thank our colleague Georg Heigold for help with training the DNN models.

6. REFERENCES

[1] D. 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.

[2] P. Kenny, G. Boulianne, P. Ouellet, and P. Dumouchel, "Joint factor analysis versus eigenchannels in speaker recognition," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, pp. 1435–1447, 2007.

[3] P. Kenny, G. Boulianne, P. Ouellet, and P. Dumouchel, "Speaker and session variability in GMM-based speaker verification," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, pp. 1448–1460, 2007.

[4] P. Kenny, P. Ouellet, N. Dehak, V. Gupta, and P. Dumouchel, "A study of interspeaker variability in speaker verification," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 16, pp. 980–988, 2008.

[5] N. Dehak, P. J. 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, pp. 788–798, 2011.

[6] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, and B. Kingsbury, "Deep neural networks for acoustic modeling in speech recognition," *IEEE Signal Processing Magazine*, vol. 29, pp. 82–97, November 2012.

[7] P. Kenny, "Bayesian speaker verification with heavy-tailed priors," in *Proc. Odyssey Speaker and Language Recognition Workshop*, 2010.

[8] T. Stafylakis, P. Kenny, P. Ouellet, P. Perez, J. Kockmann, and P. Dumouchel, "Text-dependent speaker recogntion using PLDA with uncertainty propagation," in *Proc. Interspeech*, 2013.

[9] H. Aronowitz, "Text-dependent speaker verification using a small development set," in *Proc. Odyssey Speaker and Language Recognition Workshop*, 2012.

[10] A. Larcher, K.-A. Lee, B. Ma, and H. Li, "Phonetically-constrained PLDA modeling for text-dependent speaker verification with multiple short utterances," in *Proc. ICASSP*, 2013.

[11] J. Oglesby and J. S. Mason, "Optimisation of neural models for speaker identification," in *Proc. ICASSP*, 1990.

[12] Y. Bennani and P. Gallinari, "Connectionist approaches for automatic speaker recognition," in *ESCA Workshop on Automatic Speaker Recognition, Identification and Verification*, 1994.

[13] B. Yegnanarayana and S.P. Kishore, "AANN: an alternative to GMM for pattern recognition," *Neural Networks*, vol. 15, no. 3, pp. 459–469, 2002.

[14] L.P. Heck, Y. Konig, M.K. Sönmez, and M. Weintraub, "Robustness to telephone handset distortion in speaker recognition by discriminative feature design," *Speech Communication*, vol. 31, no. 2, pp. 181–192, 2000.

[15] H. Lee, Y. Largman, P. Pham, and A. Ng, "Unsupervised feature learning for audio classification using convolutional deep belief networks," in *NIPS*, 2009.

[16] T. Stafylakis, P. Kenny, M. Senoussaoui, and P. Dumouchel, "Preliminary investigation of Boltzmann machine classifiers for speaker recognitin," in *Proc. Odyssey Speaker and Language Recognition Workshop*, 2012.

[17] I. Goodfellow, D. Warde-Farley, M. Mirza, A. Courville, and Y. Bengio, "Maxout networks," in *Proc. JMLR*, 2013, pp. 1319–1327.

[18] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Susskever, and R. R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," in *arXive preprint*, 2012.

[19] G. Dahl, T. N. Sainath, and G. E. Hinton, "Improving deep neural networks for LVCSR using rectified linear units and dropout," in *Proc. ICASSP*, 2013.

[20] J. Dean, G. Corrado, R. Monga, K. Chen, M. Devin, Q. Le, M. Mao, M. Ranzato, A. Senior, P. Tucker, K. Yang, and A. Ng, "Large scale distributed deep networks," in *NIPS*, 2012.

[21] V. Nair and G.E. Hinton, "Rectified linear units improve restricted Boltzmann machines," in *ICML*, 2010.

[22] R. Auckenthaler, M. Carey, and H. Lloyd-Thomas, "Score normalization for text-independent speaker verification systems," *Digital Signal Processing*, vol. 10, pp. 42–54, 2000.