FEATURE CLASSIFICATION BY MEANS OF DEEP BELIEF NETWORKS FOR SPEAKER RECOGNITION

Pooyan Safari, Omid Ghahabi, Javier Hernando

TALP Research Center, Department of Signal Theory and Communications
Universitat Politecnica de Catalunya - BarcelonaTech, Spain
pooyan.safari@tsc.upc.edu, {omid.ghahabi, javier.hernando}@upc.edu

ABSTRACT

In this paper, we propose to discriminatively model target and impostor spectral features using Deep Belief Networks (DBNs) for speaker recognition. In the feature level, the number of impostor samples is considerably large compared to previous works based on i-vectors. Therefore, those i-vector based impostor selection algorithms are not computationally practical. On the other hand, the number of samples for each target speaker is different from one speaker to another which makes the training process more difficult. In this work, we take advantage of DBN unsupervised learning to train a global model, which will be referred to as Universal DBN (UDBN). Then we adapt this UDBN to the data of each target speaker. The evaluation is performed on the core test condition of the NIST SRE 2006 database and it is shown that the proposed architecture achieves more than 8% relative improvement in comparison to the conventional Multilayer Perceptron (MLP).

Index Terms— Speaker Recognition, Deep Belief Network, Restricted Boltzmann Machine, Feature Classification

1. INTRODUCTION

GMM-UBM is the conventional state-of-the-art method in speaker recognition. The mean vectors of MAP adapted GMMs are concatenated to form high dimensional vectors called supervectors. Supervectors are then represented by low dimensional vectors using an effective factor analysis technique well-known as i-vector [1]. On the other hand, different types of neural networks, including Multilayer Perceptron (MLP), have been used for speaker recognition (e.g., in [2–4]). Deep Belief Networks (DBNs) have recently shown effective alternative solutions for different machine learning tasks in speech processing (e.g., [5–10]). The network parameters in DBN are pre-trained using Restricted Boltzmann Machines (RBMs). Unsupervised pre-training phase helps the network to converge faster and avoid local minima in the supervised discriminative training phase.

This work has been funded by the Spanish project SpeechTech4All (TEC2012-38939-C03-02) and the European project CAMOMILE (PCIN-2013-067).

RBMs and DBNs have also been used in speaker recognition for different purposes. Different combinations of RBMs have been used in [11, 12] to model i-vectors. In [13] speaker factors are extracted using RBMs. In [14] and [15] RBMs have been used to extract pseudo-i-vectors from acoustic features and i-vectors, respectively. They have also been employed in [16] as a non-linear transformation and dimension reduction stage for GMM supervectors. DBNs have recently been used to extract Baum-Welch statistics for supervector and i-vector extraction [17, 18]. Other deep learning techniques have also been used in speaker recognition (e.g., [19, 20]).

In this paper, the authors propose to use speaker spectral features as the inputs to DBNs in order to build discriminative target speaker models. In the feature level, the number of impostor samples is considerably larger than when i-vectors are used as the inputs [10, 21, 22]. Therefore, the impostor selection techniques proposed in [10, 22] are not computationally practical. On the other hand, as the number of samples for each target is different from one speaker to another, training DBNs for different speakers will be more difficult. We take advantage of DBN unsupervised learning to train a global model, which is referred to as Universal DBN (UDBN). Then the UDBN is adapted to each target data to build discriminative speaker models. Experimental results show that the proposed architecture achieves more than 8% relative improvement in comparison to the conventional MLP.

2. DEEP BELIEF NETWORKS

Deep Belief Networks (DBNs) comprise multiple layers of stochastic hidden units above a single layer of visible units (Fig. 1). They form originally a generative model which is able to capture higher-order statistics of input data [23]. They can be trained efficiently using a greedy layer-wise algorithm in which every two adjacent layers are considered as a Restricted Boltzmann Machine (RBM). The output of each RBM is considered as the input to its above RBM (Fig. 1.a). RBMs are constructed from two layers of stochastic hidden and visible units where there is no intra-layer connection between units (Fig. 2.a). Units can be either stochastic binary or

978-0-9928626-3-3/15/$31.00 ©2015 IEEE

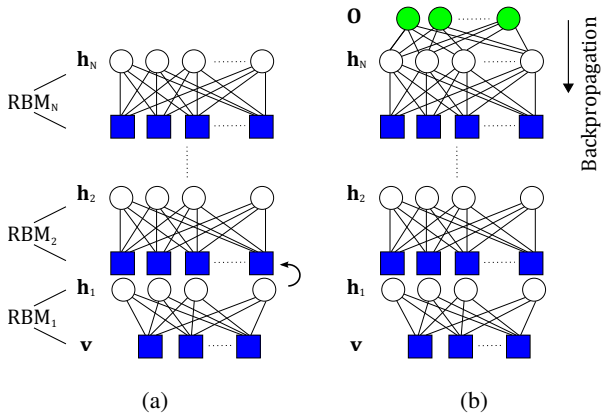

Fig. 1. Unsupervised (a), and supervised (b) DBN training.

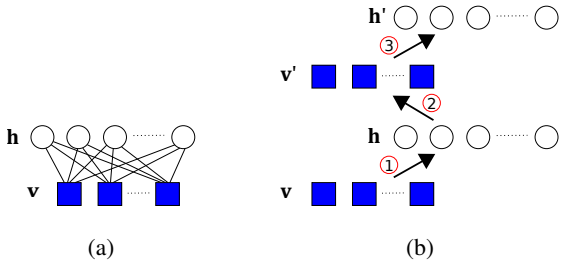

Fig. 2. RBM (a), and RBM training using CD-1 algorithm (b).

Gaussian real-valued. RBMs are trained via an approximated version of the Contrastive Divergence (CD) algorithm called CD-1 [23]. As it is shown in Fig. 2.b, CD-1 is carried out in a three-step procedure. First, the hidden layer values are computed given the visible units with the posterior probability distribution,

$$p\left(h_j = 1 \mid \mathbf{v}, \theta\right) = \sigma\left(a_j + \sum_{i=1}^{V} w_{ij} v_i\right) \tag{1}$$

where $\theta = (\mathbf{w}, \mathbf{b}, \mathbf{a})$ is the set of RBM parameters, including weights, visible and hidden biases, respectively. $\sigma(x) = (1 + e^{-x})^{-1}$ is the sigmoid function and V is the number of visible units. Second, the values for visible units are reconstructed given the hidden units. Depending on the type of the visible units, the posterior probability of the reconstructed values will be,

$$p\left(v_i = 1 \mid \mathbf{h}, \theta\right) = \sigma\left(b_i + \sum_{j=1}^{H} w_{ij} h_j\right) \tag{2}$$

$$p\left(v_i = 1 \mid \mathbf{h}, \theta\right) = \mathcal{N}\left(b_i + \sum_{j=1}^{H} w_{ij} h_j, 1\right) \tag{3}$$

where H is the number of hidden units. $\mathcal{N}(\mu, \delta^2)$ is a Gaussian with mean μ and variance δ^2. It is important that hidden unit likelihoods are converted to binary values before being used in (2) and (3) [24]. Third, the first step is repeated given the values of the reconstructed visible units.

Once the procedure is completed, the network weights will be modified by,

$$\Delta w_{ij} \approx -\varepsilon\left(\langle v_i h_j \rangle_{data} - \langle v_i h_j \rangle_{recons}\right) \tag{4}$$

where ε is the learning rate, w_{ij} is the weight between a pair of visible v_i and hidden unit h_j. $\langle \cdot \rangle_{data}$ and $\langle \cdot \rangle_{recons}$ denote the expectations when the hidden state values are driven respectively from the input data and the reconstructed one. This process is iterated until the algorithm converges. Each iteration is called an epoch. In order to accelerate the parameter-updating process, it is recommended to divide the whole training dataset into smaller ones, called mini-batches.

When the generative DBN is trained (Fig. 1.a), it can be converted to a discriminative one by adding a label layer on top of the network and performing a standard backpropagation algorithm (Fig. 1.b). Actually, the greedy layer-wise RBM-based training is considered as a pre-training phase for the discriminative DBN. It is shown in [23] that in practice the pre-training phase outperforms the random initialization of the network and avoids local minima for the supervised training phase.

3. SPEAKER FEATURE CLASSIFICATION WITH DBN

Fig. 3 shows the block diagram of the proposed architecture in this paper. First, features for impostor and target utterances are extracted. In the next step, a speaker-dependent mean-variance normalization is applied. Impostor samples are then subject to an impostor sample selection step. As it is shown in Fig. 3, impostor sample selection is applied before both Universal DBN (UDBN) and discriminative target model training.

In [10, 21, 22] the authors apply a similar method to discriminatively model the target and impostor speaker i-vectors. In this paper, we use spectral features. The use of speaker features gives rise to new problems. The amount of impostor speakers, and therefore the number of impostor samples, are considerably large. Different impostor selection methods have been proposed for i-vectors [10, 22]. However, they are not practical in the frame feature selection due to the computational complexity and memory usage. Therefore, we perform simply a random impostor sample selection. In the case of UDBN, we select as many samples as possible. The number of selected samples is constrained by the resource limitations. In the case of speaker models, we select randomly the same impostor samples for all target speakers. In order to keep balanced the number of target and impostor samples for each target speaker model, the number of selected impostor samples is almost equal to the average number of target samples.

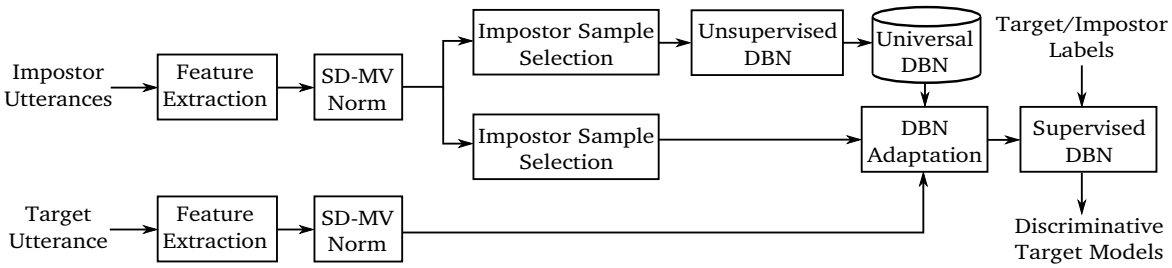

Fig. 3. Block-diagram of the proposed Adapted DBN (ADBN) method (SD-MV Norm: Speaker-Dependent Mean-Variance Normalization).

On the other hand, the number of target samples varies from one speaker to another, which is not the case when i-vectors are the inputs to the network [10, 21, 22]. We fix the number of mini-batches for all target speakers instead of using fixed mini-batch size. In this way, the number of times that the parameters of each network is updated in each iteration (epoch) will remain constant for all target speakers.

Unlike the previous works [10,21] in which very few input i-vector samples are available for each target speaker model, in the feature vector level we have enough data to train networks. Therefore, the overfitting problem will be less probable in this case, whereas the training time will be considerably higher. Moreover, in [10, 21] the authors proposed to balance manually the number of target and impostor samples in each mini-batch, which is not necessary in case of feature inputs.

The aim is to capture the information of all available background data by training a generative DBN in an unsupervised manner, and then to adapt the background model to few available data of each target speaker. The two main steps of this process are described as follows.

3.1. Universal DBN

As it was mentioned in section 2, DBN can be trained in an unsupervised manner without labelled data. As it is shown in Fig. 3, we train a DBN model based on the background data which is referred to as Universal DBN (UDBN). This model is further adapted to the data of each target speaker. UDBN can also tackle the imbalance between the two classes of impostor and target speaker samples by incorporating the information lies in the huge amount of impostor data in a single universal model. The UDBN should be built on the whole available impostor samples. However, due to resource limitations we select randomly as many impostor samples as possible.

3.2. Unsupervised DBN adaptation

As it was mentioned in section 2, the DBN pre-training can lead to better initialization of the network parameters which are further used in the supervised phase [23]. However, the DBN pre-training may be insufficient to build a good model, when few samples are available.Therefore, we propose to adapt the UDBN model of section 3.1 to both target and impostor samples of each target model. The adaptation is carried out by pre-training each network initialized by the UDBN parameters. In order to avoid overfitting, fewer number of epochs is used for the unsupervised training phase of the network. We refer to this method as Adapted DBN (ADBN).

4. EXPERIMENTAL RESULTS

4.1. Database and setup

Two different types of features have been used for the experiments. One is the log filter bank energies (FBE), and the other is frequency filtering (FF) [25]. FF features, like MFCCs, are a decorrelated version of FBEs [25]. It has been shown that FF features achieve equal or better performance than MFCCs [25]. Both FBE and FF features are extracted every 10 ms with a 30 ms Hamming window. The size of static FBE and FF features are 18 and 16, respectively. We use 5 frames (2-1-2) of FBEs or FFs in order to compose 90- or 80-dimensional feature inputs for the networks. Before feature extraction, speech signals are subject to an energy-based silence removal process. All the features are mean-variance normalized per each utterance.

The whole core test condition of the NIST 2006 SRE evaluation [26] is used in all experiments. It comprises of 816 target speakers, with $51,068$ trials. Each signal consists of about two minutes of speech. The inputs to the networks consist of target samples and $10,000$ impostor samples (close to the average number of samples of all target speakers), which are randomly selected from impostor speakers. As it was mentioned in section 3 we consider a fixed number of mini-batches for all target models, which is equal to 200 in all of the experiments. However, UDBN is trained using four million randomly selected impostor samples with a fixed mini-batch size of 100.

All the architectures used in this paper comprise one hidden layer with 128 hidden units. The fixed momentum and weight decay for all the systems are set to 0.9 and 10^{-7}, respectively. A sparsity target of 0.05 and a sparsity penalty of 10^{-4} are used for all the systems. A learning rate of

Classifier	Feature	EER (%)
MLP	FBE	19.81
MLP	FF	18.12
DBN	FF	17.19
ADBN	**FF**	**16.65**

Table 1. Results obtained on the core test condition of NIST SRE 2006 evaluation. ADBN is referred to the proposed Adapted DBN approach.

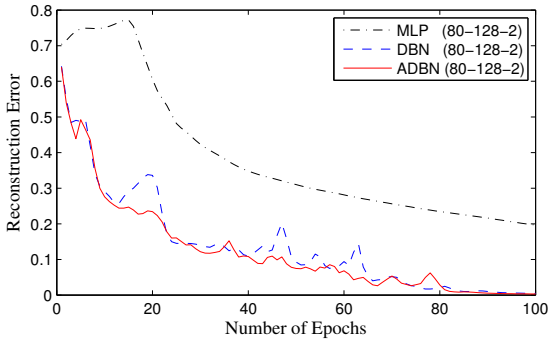

Fig. 4. Comparison of the convergence speeds of MLP, DBN, and ADBN configurations for a given target speaker.

0.05 is employed for both MLP configurations. For MLP with FBE and FF features 700 and 400 epochs are used, respectively. Since the input samples are real-valued data, a Gaussian-Bernoulli RBM is used for all DBNs [24, 27]. The unsupervised part of the DBN is trained by a learning rate of 0.0001 with 100 epochs which is then followed by a supervised training with $\varepsilon = 0.09$ and 150 epochs. The UDBN is trained in an unsupervised fashion with $\varepsilon = 0.0001$ and 200 epochs. Adaptation process in ADBN is carried out by 5 epochs with $\varepsilon = 0.001$, and supervised phase is trained with $\varepsilon = 0.06$ and 250 epochs.

4.2. Results

The obtained results have been shown in Table 1 for different methods and configurations. There is more than 8.5% relative improvement using the MLP network with FF features in comparison to the MLP with FBE features. It should also be mentioned that the number of epochs needed for an MLP system to converge with FBE features is much higher than the one for FF features. This may be due to the fact that FF features are more decorrelated than the FBE features as stated in [25]. Therefore, the network needs more iterations to learn the correlation among input components. Regarding both convergence speed and better classification performance, we decided to continue our experiments with FF features. As it was expected and is shown in Table 1, DBN pre-training improves the equal error rate (EER). The results obtained for ADBN reveals the effectiveness of the proposed method in section 3. The result is better than the DBN pre-training and more than 8% relative improvement is achieved comparing

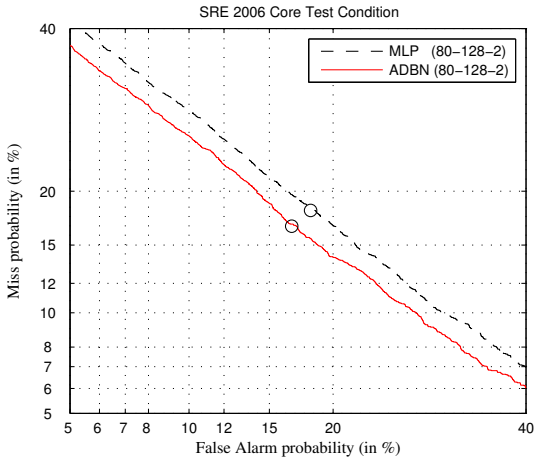

Fig. 5. Comparison of DET curves for MLP and the proposed ADBN.

with MLP using FF features.

Fig. 4 compares the convergence speeds of MLP, DBN, and ADBN configurations for a given target speaker model in the supervised training phase. Each of these architectures is trained with the network parameters of the corresponding results reported in Table 1. The number of samples of the selected target speaker is close to the average number of samples for all target speakers. There is a considerable difference between MLP and DBN-based methods in terms of convergence time. For approximately the same level of reconstruction error, MLP needs more than twice as many epochs as the one for DBNs. In other words, the convergence speed of DBN-based models is much higher than the one of conventional MLP. This is another advantage of DBNs which is of great importance especially when dealing with large amount of data with few available resources. Fig. 4 also reveals that both DBN and ADBN are converged in a similar way, but the ADBN has less fluctuations.

Fig. 5 shows the detection error trade-off (DET) curves for the MLP and ADBN techniques. It shows that not only at the EER but also at all other working points ADBN performs better than MLP configuration.

5. CONCLUSION

We discriminatively train target speaker models with the speaker spectral features using DBNs. Two new issues are addressed in this paper, namely the large amount of impostor data and the difference among the number of samples for different target speakers. The authors take advantage of a global model which is referred to as Universal DBN (UDBN). The UDBN is adapted to data of each target speaker. The preliminary results on the core test condition of the NIST SRE 2006 database show that this UDBN adaptation together with discriminative training of target speaker models outperforms both the conventional MLP and DBN.

REFERENCES

[1] 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, no. 4, pp. 788–798, 2011.

[2] K.R. Farrell, R.J. Mammone, and K.T. Assaleh, "Speaker recognition using neural networks and conventional classifiers," *IEEE Transactions on Speech and Audio Processing*, vol. 2, no. 1, pp. 194–205, 1994.

[3] S.E. Fredrickson and L. Tarassenko, "Text-independent speaker recognition using neural network techniques," in *Proc. Fourth International Conference on Artificial Neural Networks*, 1995, pp. 13–18.

[4] L. Wang, K. Chen, and H. Chi, "Capture interspeaker information with a neural network for speaker identification," *IEEE Transactions on Neural Networks*, vol. 13, no. 2, pp. 436–445, 2002.

[5] A. Mohamed, G. Dahl, and G.E. Hinton, "Deep belief networks for phone recognition," in *Proc. of NIPS 2009 Workshop on Deep Learning for Speech Recognition and Related Applications*, 2009.

[6] P. Hamel and D. Eck, "Learning features from music audio with deep belief networks," in *Proc. 11th International Society for Music Information Retrieval Conference (ISMIR)*, 2010, pp. 339–344.

[7] X. Yang, Q. Chen, S. Zhou, and X. Wang, "Deep belief networks for automatic music genre classification," in *Proc. Twelfth Annual Conference of the International Speech Communication Association*, 2011, pp. 2433–2436.

[8] H. Lee, Y. Largman, P. Pham, and a. Ng, "Unsupervised feature learning for audio classification using convolutional deep belief networks," *Advances in Neural Information Processing Systems*, vol. 22, pp. 1096–1104, 2009.

[9] X. Zhang and J. Wu, "Deep belief networks based voice activity detection," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 21, no. 4, pp. 697–710, Apr. 2013.

[10] O. Ghahabi and J. Hernando, "Deep belief networks for i-vector based speaker recognition," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2014, pp. 1700–1704.

[11] T. Stafylakis and P. Kenny, "Preliminary investigation of Boltzmann machine classifiers for speaker recognition," in *Proc. Odyssey Speaker and Language Recognition Workshop*, 2012.

[12] M. Senoussaoui, N. Dehak, P. Kenny, and R. Dehak, "First attempt of Boltzmann machines for speaker verification," in *Proc. of the Odyssey: The Speaker and Language Recognition Workshop*, 2012.

[13] T. Stafylakis, P. Kenny, M. Senoussaoui, and P. Dumouchel, "PLDA using Gaussian restricted Boltzmann machines with application to speaker verification," in *Proc. Interspeech*, 2012.

[14] V. Vasilakakis, S. Cumani, P. Laface, and P. Torino, "Speaker recognition by means of deep belief networks," in *Proc. Biometric Technologies in Forensic Science*, 2012.

[15] S. Novoselov, T. Pekhovsky, K. Simonchik, and A. Shulipa, "RBM-PLDA subsystem for the NIST i-vector challenge," *system*, vol. 8, pp. 9, 2014.

[16] O. Ghahabi and J. Hernando, "Restricted Boltzmann machine supervectors for speaker recognition," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2015.

[17] W.M. Campbell, "Using deep belief networks for vector-based speaker recognition," in *Proc. Fifteenth Annual Conference of the International Speech Communication Association*, 2014, pp. 676–680.

[18] P Kenny, V Gupta, T Stafylakis, P Ouellet, J Alam, Patrick Kenny, Vishwa Gupta, Themos Stafylakis, Pierre Ouellet, and Jahangir Alam, "Deep neural networks for extracting Baum-Welch statistics for speaker recognition," .

[19] 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. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2014.

[20] Y. Lei, N. Scheffer, L. Ferre, and M. Mclaren, "A novel scheme for speaker recognition using a phonetically-aware deep neural network," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2014.

[21] O. Ghahabi and J. Hernando, "i-vector modeling with deep belief networks for multi-session speaker recognition," in *Proc. Odyssey*, 2014, pp. 305–310.

[22] O. Ghahabi and J. Hernando, "Global impostor selection for DBNs in multi-session i-vector speaker recognition," in *Advances in Speech and Language Technologies for Iberian Languages*, pp. 89–98. Springer, 2014.

[23] G.E. Hinton, S. Osindero, and Y. Teh, "A fast learning algorithm for deep belief nets," *Neural Computation*, vol. 18, no. 7, pp. 1527–1554, 2006.

[24] G.E. Hinton, "A practical guide to training restricted Boltzmann machines," *Momentum*, vol. 9, no. 1, 2010.

[25] C. Nadeu, Dušan Macho, and J. Hernando, "Time and frequency filtering of filter-bank energies for robust HMM speech recognition," *Speech Communication*, vol. 34, no. 1, pp. 93–114, 2001.

[26] "The NIST Year 2006 Speaker Recognition Evaluation Plan," 2006.

[27] G.E. Dahl, D. Yu, L. Deng, and A. Acero, "Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 1, pp. 30–42, 2012.