# DEEP BELIEF NETWORKS FOR I-VECTOR BASED SPEAKER RECOGNITION

*Omid Ghahabi, Javier Hernando*

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

## ABSTRACT

The use of Deep Belief Networks (DBNs) is proposed in this paper to model discriminatively target and impostor i-vectors in a speaker verification task. The authors propose to adapt the network parameters of each speaker from a background model, which will be referred to as Universal DBN (UDBN). It is also suggested to backpropagate class errors up to only one layer for few iterations before to train the network. Additionally, an impostor selection method is introduced which helps the DBN to outperform the cosine distance classifier. The evaluation is performed on the core test condition of the NIST SRE 2006 corpora, and it is shown that 10% and 8% relative improvements of EER and minDCF can be achieved, respectively.

***Index Terms***— Speaker Recognition, i-vector, Deep Belief Network, Neural Network

## 1. INTRODUCTION

Even though the task of speaker recognition has been investigated for several decades, new approaches are still being explored. The i-vector framework recently developed in [1] is an effective factor analysis method for the compact representation of the speaker useful information. The framework maps every speaker utterance to a low dimensional identity vector. The target and test i-vectors can then be compared effectively using the cosine distance metric [1]. There are also some post-processing techniques to compensate speaker and session variabilities (eg., [1][2][3][4]) and, therefore, to increase the overal performance of the system.

On the other hand, Deep Belief Networks (DBN) have recently opened a new research line in image, audio, and speech processing areas (eg., [5][6][7] [8][9]). DBNs are originally generative networks which can be trained by a greedy layer-wise algorithm using Restricted Boltzmann Machines (RBMs) [10][11]. However, by adding a top label layer and using a standard backpropagation algorithm, these generative DBNs can be converted to discriminative ones what is often called a pre-trained discriminative network [11][12].

Acoustic modeling using DBNs has been shown to be effective in speech recognition [5][13][12] [14]. However, few attempts using only RBMs [15][16] or generative DBNs [17] have been carried out in speaker recognition area. In this paper, we propose to use both generative and discriminative DBNs. We take the advantage of the unsupervised learning to model a global DBN which will be called Universal DBN (UDBN) in this paper and the advantage of the supervised learning to model each target model discriminatively. In addition, by using i-vectors as inputs to the network, DBNs will be combined with the recent successful i-vector approach.

Two main ideas will be employed in this paper to make such a structure efficient for speaker recognition. Firstly, with a proposed impostor selection method and clustering, the number of impostor i-vectors are decreased to provide a balanced training. Secondly, a UDBN-adaptation method is proposed to initialize the network parameters. It will be shown that the proposed adaptation method outperforms both random and pre-training initializations.

The rest of the paper is organized as follows. Sections 2 and 3 review respectively the i-vector framework and the background of DBN. The proposed method is described in details in Section 4. Section 5 presents the experimental setup and results. And section 6 concludes the paper.

## 2. I-VECTOR EXTRACTION

This section has a brief overview on the i-vector framework developed in [1]. Given the centralized Baum-Welch statistics from all available speech utterances, the low rank total variability matrix ($T$) is trained in an iterative process. This matrix tries to capture all kinds of variabilities, including speaker and session variabilities, appeared in training utterances. The training process assumes that an utterance can be represented by the Gaussian Mixture Model (GMM) mean supervector,

$$M = m + T\omega \tag{1}$$

where $m$ is the speaker- and session-independent mean supervector from the Universal Background Model (UBM), and $\omega$ is a low rank vector referred to as the identity vector or

This work has been funded by the Spanish project SARAI (TEC2010-21040-C02-01).

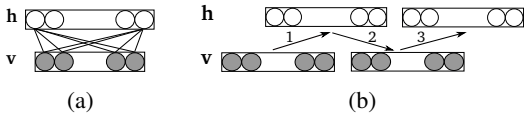


**Fig. 1**: RBM (a) and RBM training (b).

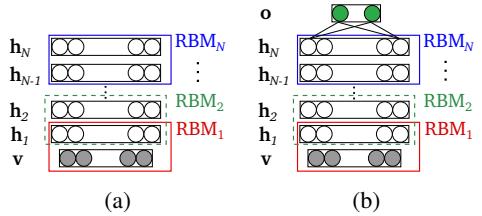


**Fig. 2**: Generative (a) and discriminative (b) DBNs.

i-vector. The supervector $M$ is assumed to be normally distributed with the mean $m$ and the covariance $TT^t$, and the i-vectors have a standard normal distribution $\mathcal{N}(0,1)$. Extracting an i-vector from the total variability subspace is essentially a maximum a-posterior adaptation of $\omega$ in the space defined by $T$. These i-vectors are named raw i-vectors as they are not post-processed by any techniques. More details can be found in [1].

## 3. DEEP BELIEF NETWORKS

DBNs are originally probabilistic generative models with multiple layers of stochastic hidden units above a layer of visible variables. There is an efficient greedy layer-wise algorithm for learning DBNs [11]. The algorithm treats every two adjacent layers as an RBM (Figs. 1a and 2a). The output of each RBM is considered as the input to its above RBM. RBMs are constructed from a layer of binary stochastic hidden units and a layer of stochastic visible units (Fig. 1a).

Training an RBM is based on an approximated version of the Contrastive Divergence (CD) algorithm [10][11] which consists of three steps (Fig. 1b). At first, hidden states (**h**) are computed given visible states (**v**), then given **h**, **v** is reconstructed, and in the third step **h** is updated given the reconstructed **v**. Finally, the change of connection weights is given as follows,

$$\Delta w_{ij} \approx -\alpha \left( \langle v_i h_j \rangle_{data} - \langle v_i h_j \rangle_{recon} \right) \qquad (2)$$

where $\alpha$ is the learning rate, $w_{ij}$ represents the weight between the visible unit $i$ and the hidden unit $j$, $\langle . \rangle_{data}$ and $\langle . \rangle_{recon}$ denote the expectations when the hidden state values are driven respectively from the input visible data and the reconstructed data. Actually, the training process tries to minimize the reconstruction error between the actual input data and the reconstructed one. The parameter updating process is iterated until the algorithm converges. Each iteration is called an epoch. It is possible to perform the above parameter update after processing each training example, but it is often more efficient to divide the whole input data (batch) into smaller size batches (minibatch) and to do the parameter update by an average over each minibatch. More theoretical and practical details can be found in [10][11][18].

When the unsupervised learning is finished, by adding a label layer on top of the network and doing a supervised back-propagation training, it can be converted to a discriminative model (Fig. 2b). In other words, the unsupervised learning can be considered as a pre-training for the supervised stage. It has been shown [11] that this unsupervised pre-training can set the weights of the network to be closer to a good solution than random initialization and, therefore, avoids local minima when using supervised gradient descent.

## 4. PROPOSED DBN-BASED APPROACH

The idea is to model discriminatively the target and impostor i-vectors by a DBN structure. As illustrated in Fig. 3, the proposed method is composed of three main parts which will be described in details as follows.

### 4.1. Impostor Selection and Balanced Training

As the speaker models in the proposed method will be finally discriminative, they need both positive and negative data as inputs. However, the problem is that the amount of positive and negative data are not balanced in this case. There is only one target i-vector as the positive sample and there are many impostor i-vectors as the negative ones. Training a network with such highly unbalanced data will yield overfitting. To avoid this, the number of impostors is reduced in two steps (Fig. 3).

Firstly, among a large number of impostors only the most informative ones are selected. The selection method is inspired from a Support Vector Machine (SVM)-based technique proposed in [19]. The following pseudocode shows how our selection method works,

1. For each client i-vector $s_t \in S$,
   (a) Compute $score\left(s_t, b_m|_{m=1}^{M}\right)$,
   (b) Choose the first $n$ highest scores and add their corresponding impostor indexes to a set named $H$
2. Compute the histogram of $H$ and sort it descendingly,
3. Choose the first $k$ impostors as the selected ones.

where $score\left(s_t, b_m|_{m=1}^{M}\right)$ is the cosine score between $s_t$ and all impostors in the large dataset $B$. The parameters $n$ and $k$ represent, respectively, the number of the closest impostors to each target and the statistically closest ones to all available targets. They will be determined experimentally in section 5.

Secondly, as the number of selected impostors is still high in comparison to the number of target i-vectors, they are clustered by the k-means algorithm using the cosine distance cri-

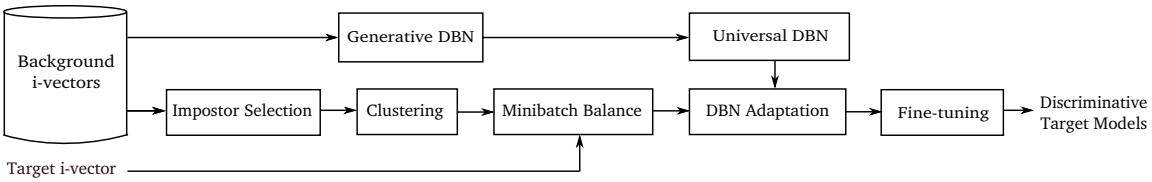


**Fig. 3**: Block-diagram of the proposed method.

teria. The centroids of the clusters are used as the final negative samples.

On the other hand, the target i-vector is repeated as many as the number of impostor centroids. The repeated target i-vectors will not act exactly the same as each other due to the sampling noise created in the pre-training process of the network [18]. Moreover, in both adaptation and supervised learning stages (sections 4.2 and 4.3), the repeated versions make the target and impostor classes having the same weights when the network parameters are being updated. Once the number of positive and negative samples are balanced, they are divided equally among minibatches. The optimum numbers of impostor clusters and minibatches will be determined experimentally in section. 5.

### 4.2. Universal DBN (UDBN) and Adaptation

Unlike the conventional neural networks that need the labeled data to be trained, DBNs do not necessarily need such labeled data as inputs. Actually, they have the ability to be trained unsupervisingly [11][10]. Hence, they can be used for training a global model called in this paper UDBN. By feeding many i-vectors from different background speakers, a UDBN can be trained. The training is carried out layer by layer using RBMs as described in section 3.

In general, neural network parameters are initialized randomly. As it was mentioned in section 3, it is shown that the pre-trained parameters can be a better initialization for training a network. However, when a few numbers of input samples are available, just pre-training will not be enough to achieve a good model. In this case we adapt the UDBN parameters to the new data of each speaker including both target and impostor samples obtained in section 4.1. The adaptation is carried out by pre-training each network initialized by the UDBN parameters. To pre-train, only a few numbers of iterations (epoch) are used, otherwise the network will be led to overfitting. In section 5 it will be shown that the proposed adaptation process outperforms both the conventional random initializing and pre-training started from random numbers.

### 4.3. Fine-tuning

Once the adaptation process is completed, a label layer is added on the top of the network and the stochastic gradient descent backpropagation is carried out on each minibatch as the fine-tuning process. The softmax and the logistic sigmoid will be the activation functions of the top label layer and the rest hidden layer units, respectively. The connection weights between the top label layer and its hidden layer bellow will be initialized by small random numbers ($\mathcal{N}(0, 0.01)$).

To minimize the negative effect of using random numbers, we pre-train the top layer as well. Pre-training is performed by only one layer error backpropagating for a few iterations. Then a full backpropagation will be carried out on the whole layers. If the input labels in the training phase are chosen as ($l_1 = 1, l_2 = 0$) and ($l_1 = 0, l_2 = 1$) for target and impostor i-vectors respectively, the final output score in the testing phase will be computed in a Log Likelihood Ratio (LLR) form as follows,

$$LLR = log(o_1) - log(o_2) \tag{3}$$

where $o_1$ and $o_2$ represent respectively the output of the first and the second units of the top layer. LLR computation helps to gaussianize the true and false score distributions which can be useful for score fusion. In addition, to make the fine-tuning process more efficient a momentum factor is used to smooth out the updates, and the weight decay method is used to penalize large weights.

## 5. EXPERIMENTAL RESULTS

This section gives the details of the databases, the baseline and the proposed DBN-based setups emphasizing on the effect of each main idea proposed in section 4.

### 5.1. Baseline setup

Features used in the experiments are Frequency Filtering (FF) features [20] extracted every 10 ms using a 30 ms Hamming window. The number of static FF features is 16 and together with delta FF and delta energy, they make 33-dimensional feature vectors. Before feature extraction, speech signals are subjected to an energy-based silence removal process.

The whole core test condition of the NIST 2006 SRE evaluation [21] is used in all experiments. It includes 816 target models and 51,068 trials. The signals have around two minutes of speech. Performance is evaluated using the Equal Error Rate (EER) and the minimum Decision Cost Function (minDCF) calculated using $C_M = 10, C_{FA} = 1$ and $P_T = 0.01$.

The i-vectors in all the experiments are 400-dimensional raw i-vectors. Raw i-vectors are compared in the baseline system using the cosine distance classifier. The i-vector framework is carried out using the ALIZE open source software

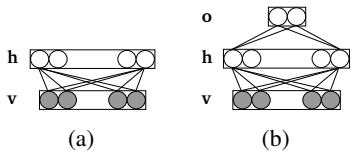


**Fig. 4**: Generative (a) and discriminative (b) 1-layer DBN structures used in the experiments.

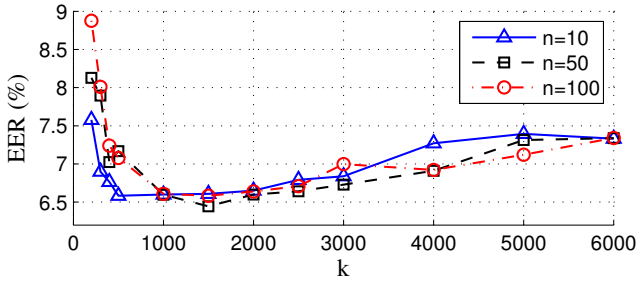


**Fig. 5**: Determination of the parameters $n$ and $k$ defined in sec. 4.1.

[22]. The UBM and the $T$ matrix are trained using more than 6,000 speech files collected from NIST 2004 and 2005 SRE corpora. It is worth noting that in the case of NIST 2005 only the speech files of those speakers which do not appear in NIST 2006 database are collected. The gender-independent UBM is represented as a diagonal covariance, 512-component GMM.

### 5.2. Proposed DBN setup

Figure 4 illustrates the generative and discriminative DBNs used in the experiments. As it is shown in this figure, only one hidden layer networks are explored in this paper. The hidden layer has 512 units.

Experimentally, the number of minibatches and the number of impostor clusters are set respectively to 3 and 12. Each minibatch will include four impostor centroids and four repeated target samples.

The background i-vectors (Fig. 3) are extracted from the same speech files used for training the UBM and the $T$ matrix. Fig. 5 illustrates the variability of EER in terms of the two parameters $n$ and $k$ defined in sec. 4.1. The same behavior can be observed for minDCF. The figure shows that the minimum EER is obtained when $n = 50$ and $k = 1500$. UDBN is trained with the same background i-vectors of the impostor database. As the input i-vectors are real-valued, a Gaussian-Bernoulli RBM [18][12] is employed. Since the minimum divergence training algorithm [23] is used in the i-vector extraction process, i-vectors are already zero-mean unit-variance Normal distributed and, therefore, no post-processing is carried out. The learning rate ($\alpha$), number of epochs (NofE), momentum, and weight decay are set respectively to 0.012, 50, 0.9, and 0.0002.

The generative parts of the speaker models (Fig. 4a) are initialized by the UDBN parameters and then are pre-trained with $\alpha = 0.05$ and NofE $= 25$. The momentum and weight decay values are kept the same as in UDBN. The top connection weights are initialized by $\mathcal{N}(0, 0.01)$ and pre-trained with $\alpha = 1$ and NofE $= 15$ before the whole backpropagation is performed. The momentum is started by 0.4 and is scaled up by 0.1 after each epoch (up to 0.9). The whole backpropagation is then carried out with $\alpha = 1$, NofE $= 30$, and a fixed momentum of 0.9. The weight decay for both top layer pre-tarining and the whole backpropagation is set to 0.0012.

**Table 1**: Results obtained on the core test condition of NIST SRE 2006 evaluation. (Imp: Impostor dataset, Init: Network initialization, T-L Init: Top layer initialization, F: Full, S: Selected, R: Random, P: Pre-training, A: Adaptation, CD: Cosine Distance).

| Setup | Imp | Init | T-L Init | EER (%) | minDCF |
|---|---|---|---|---|---|
| Baseline (i-vector+CD) | - | - | - | 7.18 | 0.0324 |
| i-vector+DBN 1 | F | R | R | 8.80 | 0.0381 |
| i-vector+DBN 2 | S | R | R | 8.29 | 0.0354 |
| i-vector+DBN 3 | S | P | R | 7.62 | 0.0336 |
| i-vector+DBN 4 | S | A | R | 7.04 | 0.0353 |
| i-vector+DBN 5 | S | A | P | **6.44** | **0.0299** |

### 5.3. Results

Table 1 compares the results of the different configurations of the proposed DBN-based structure with the baseline system. Moreover, it shows the influence of each main contribution proposed in sec. 4. For each DBN configuration the learning rates are tuned and the best results are reported in the table. The two first DBN configurations show the effectiveness of the proposed impostor selection method. The influence of the network initialization type is shown in the configurations 2-4. Comparing these configurations shows that the pre-training method outperforms the conventional random initialization and the proposed adaptation method is the best. Finally, the last structure including all three proposed ideas shows that the introduced DBN can achieve respectively 27% and 22% relative improvements of EER and minDCF in comparison with the conventional network (the first DBN configuration). In addition, the proposed DBN configuration can outperform the baseline system, in which the raw i-vectors are compared using cosine distance criteria, with relative improvements of 10% and 8% for EER and minDCF, respectively.

## 6. CONCLUSION

The authors proposed to model discriminately target and impostor i-vectors using Deep Belief Networks (DBNs). The proposed adaptation method can adapt the network parameters of each speaker from a generative background model which has been called Universal DBN (UDBN). Moreover, the proposed impostor selection and balanced training method helped the DBN structure to outperform both conventional neural networks and the cosine distance classifier.

## 7. REFERENCES


[1] N. Dehak, P. 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, May 2011.

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

[3] P. Kenny, "Bayesian speaker verification with heavy tailed priors," in *Speaker and Language Recognition Workshop (IEEE Odyssey)*, 2010.

[4] N. Brummer and E. Villiers, "The speaker partitioning problem," in *Proceedings of the Odyssey Speaker and Language Recognition Workshop, Brno, Czech Republic*, 2010.

[5] A. Mohamed, G.E. Dahl, and G. Hinton, "Acoustic modeling using deep belief networks," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 1, pp. 14–22, Jan. 2012.

[6] Z-H. Ling, L. Deng, and D. Yu, "Modeling spectral envelopes using restricted boltzmann machines and deep belief networks for statistical parametric speech synthesis," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 21, no. 10, pp. 2129–2139, 2013.

[7] X-L. 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, 2013.

[8] V. Nair and G.E. Hinton, "3-d object recognition with deep belief nets," in *Advances in Neural Information Processing Systems 22*, 2009.

[9] G.B. Huang, H. Lee, and E. Learned-Miller, "Learning hierarchical representations for face verification with convolutional deep belief networks," in *2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012, pp. 2518–2525.

[10] G.E. Hinton and R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, July 2006.

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

[12] 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, Jan. 2012.

[13] G. Hinton, L. Deng, D. Yu, G. Dahl, A. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, and T. Sainath, "Deep neural networks for acoustic modeling in speech recognition," *IEEE Signal Processing Magazine*, 2012.

[14] A. Mohamed, D. Yu, and L. Deng, "Investigation of full-sequence training of deep belief networks for speech recognition," in *Proc. Interspeech*, 2010, p. 28462849.

[15] M. Senoussaoui, N. Dehak, P. Kenny, R. Dehak, and P. Dumouchel, "First attempt of boltzmann machines for speaker verification," in *Odyssey 2012-The Speaker and Language Recognition Workshop*, 2012.

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

[17] V. Vasilakakis, S. Cumani, and P. Laface, "Speaker recognition by means of deep belief networks," in *Biometric Technologies in Forensic Science*, 2013.

[18] G.E. Hinton, "A practical guide to training restricted boltzmann machines," in *Neural Networks: Tricks of the Trade*, number 7700 in Lecture Notes in Computer Science, pp. 599–619. Springer Berlin Heidelberg, Jan. 2012.

[19] M. McLaren, B. Baker, R. Vogt, and S. Sridharan, "Improved SVM speaker verification through data-driven background dataset collection," in *IEEE International Conference on Acoustics, Speech and Signal Processing, 2009. ICASSP 2009*, 2009, pp. 4041–4044.

[20] C. Nadeu, D. Macho, and J. Hernando, "Time and frequency filtering of filter-bank energies for robust HMM speech recognition," *Speech Communication*, vol. 34, no. 12, pp. 93–114, Apr. 2001.

[21] "The NIST year 2006 speaker recognition evaluation plan," 2006.

[22] A. Larcher, J-F. Bonastre, B. Fauve, K. Lee, C. Lvy, H. Li, J. Mason, and J-Y. Parfait, "ALIZE 3.0 open source toolkit for state-of-the-art speaker recognition," in *Proc. Interspeech*, 2013, pp. 2768–2771.

[23] 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, no. 5, pp. 980–988, July 2008.