Deep Learning for Single and Multi-Session i-Vector Speaker Recognition

Omid Ghahabi and Javier Hernando

Abstract—The promising performance of Deep Learning (DL) in speech recognition has motivated the use of DL in other speech technology applications such as speaker recognition. Given i-vectors as inputs, the authors proposed an impostor selection algorithm and a universal model adaptation process in a hybrid system based on Deep Belief Networks (DBN) and Deep Neural Networks (DNN) to discriminatively model each target speaker. In order to have more insight into the behavior of DL techniques in both single and multi-session speaker enrollment tasks, some experiments have been carried out in this paper in both scenarios. Additionally, the parameters of the global model, referred to as universal DBN (UDBN), are normalized before adaptation. UDBN normalization facilitates training DNNs specifically with more than one hidden layer. Experiments are performed on the NIST SRE 2006 corpus. It is shown that the proposed impostor selection algorithm and UDBN adaptation process enhance the performance of conventional DNNs 8-20% and 16-20% in terms of EER for the single and multi-session tasks, respectively. In both scenarios, the proposed architectures outperform the baseline systems obtaining up to 17% reduction in EER.

Index Terms—Deep Neural Network, Deep Belief Network, Restricted Boltzmann Machine, i-Vector, Speaker Recognition.

I. INTRODUCTION

THE recent compact representation of speech utterances known as i-vector [1] has become the state-of-the-art in the text-independent speaker recognition. Given speaker labels for the background data, there are also some post-processing techniques such as Probabilistic Linear Discriminant Analysis (PLDA) [2], [3] to compensate speaker and session variabilities and, therefore, to increase the overall performance of the system.

On the other hand, the success of deep learning techniques in speech processing, specifically in speech recognition (e.g., [4]–[8]), has inspired the community to make use of those techniques in speaker recognition as well. Three main commonly used techniques are Restricted Boltzmann Machines (RBM), Deep Belief Networks (DBN), and Deep Neural Networks (DNN). Different combinations of RBMs have been used in [9], [10] to classify i-vectors and in [11] to learn speaker and channel factor subspaces in a PLDA simulation. RBMs and DBNs have been used to extract a compact representation of speech signals from acoustic features [12] and i-vectors [13]. RBMs have also been employed

in [14] as a non-linear transformation and dimension reduction stage for GMM supervectors. DBNs have been used in [15] as unsupervised feature extractors and in [16] as speaker feature classifiers. Furthermore, in [17]–[19] they have been integrated in an adaptation process to provide a better initialization for DNNs. DNNs have been utilized to extract Baum-Welch statistics for supervector and i-vector extraction [20]–[23]. DNN bottleneck features are recently employed in the i-vector framework [24], [25]. Additionally, different types of i-vectors represented by DNN architectures are proposed in [26], [27].

The main attention of the National Institute of Standard and Technology (NIST) over the last two years to combine i-vectors with new machine learning techniques [28], [29] encouraged the authors to extend the prior works developed in [17], [18]. The authors took advantage of unsupervised learning of DBNs to train a global model referred to as Universal DBN (UDBN) and DNN supervised learning to model each target speaker discriminatively. To provide a balanced training, an impostor selection algorithm and to cope with few training data a UDBN-adaptation process was proposed.

In this work, deep architectures with different number of layers are explored for both single and multi-session speaker enrollment tasks. The parameters of the global model are normalized before adaptation. Normalization helps to facilitate training networks specifically where more than one hidden layer is used. The top layer pre-training proposed in [17] is not used in this work. The reason is that it emphasizes on the top layer connection weights and avoids the lower hidden layers to learn enough from the input data. This fact is of more importance where higher number of hidden layers are used. It is supposed, in this work, that there is no labeled background data. Moreover, no unsupervised labeling technique (e.g., [13], [30]) is employed. This work shows how DNN architectures can be more efficient in this particular task. Experimental results performed on the NIST SRE 2006 corpus [31] show that the proposed architectures outperform the baseline systems in both single and multi-session speaker enrollment tasks.

II. DEEP LEARNING

Deep Learning (DL) refers to a branch of machine learning techniques which attempts to learn high level features from data. Since 2006 [32], [33], DL has become a new area of research in many applications of machine learning and signal processing. Various deep learning architectures have been used in speech processing (e.g., [7], [8], [34]–[36]). Deep Neural Networks (DNN), Deep Belief networks (DBN), and Restricted Boltzmann Machines (RBM) are three main

techniques we have used in this work to discriminatively model each target speaker given input i-vectors.

DNNs are feed-forward neural networks with multiple hidden layers (Fig. 1a). They are trained using discriminative back-propagation algorithms given class labels of input vectors. The training algorithm tries to minimize a loss function between the class labels and the outputs. For classification tasks, cross entropy is often used as the loss function and the softmax is commonly used as the activation function at the output layer [37]. Typically, the parameters of DNNs are initialized with small random numbers. Recently, it has been shown that there are more efficient techniques for parameter initialization [38]–[40]. One of those techniques consists in initializing DNN with DBN parameters, which it is often referred to as unsupervised pre-training or just hybrid DBN-DNN [5], [41]. It has empirically been shown that this pre-training stage can set the weights of the network closer to an optimum solution than random initialization [38]–[40].

DBNs are generative models with multiple hidden layers of stochastic units above a visible layer which represents a data vector (Fig. 1b). The top two layers are undirected and the other layers have top-down directed connections to generate the data. There is an efficient greedy layer wised algorithm to train DBN parameters [33]. In this case, DBN is divided in two-layer sub-networks and each one is treated as an RBM (Fig. 1c). When the first RBM corresponding to the visible units is trained, its parameters are frozen and the outputs are given to the RBM above as input vectors. This process is repeated until the top two layers are reached.

RBMs are generative models constructed from two undirected layers of stochastic hidden and visible units (Fig. 2a). RBM training is based on maximum likelihood criterion using the stochastic gradient descent algorithm [5], [33]. The gradient is estimated by an approximated version of the Contrastive Divergence (CD) algorithm which is called CD-1 [32], [33]. As it is shown in Fig. 2b, CD-1 consists of three steps. At first, hidden states (**h**) are computed given visible unit values (**v**). Secondly, given **h**, **v** is reconstructed. Thirdly, hidden unit values are computed given the reconstructed **v**. Finally, the change of connection weights is given as follows,

$$\Delta_{w_{ij}} \approx -\eta \left(\langle v_i h_j \rangle_{data} - \langle v_i h_j \rangle_{recon} \right) \qquad (1)$$

where η is the learning rate, w_{ij} represents the weight between the visible unit i and the hidden unit j, and $\langle . \rangle_{data}$ and $\langle . \rangle_{recon}$ denote the expectations when the hidden state values are driven from the input visible data and the reconstructed data, respectively. The biases are updated in a similar way. More theoretical and practical details can be found in [32], [33], [42]. The whole training algorithm is given in [14].

In all of these networks, it is possible to update the parameters 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 update the parameters by averaging the gradients over each minibatch. The parameter updating procedure is repeated when the whole available input data are processed. Each iteration is called an epoch.
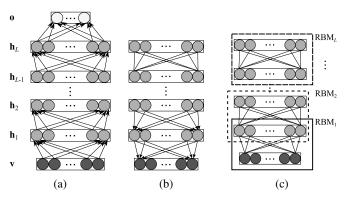

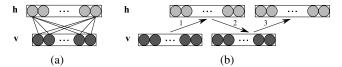Fig. 1: (a) DNN ,(b) DBN , and (c) DBN training/DNN pre-training.

Fig. 2: (a) RBM and (b) RBM training.

III. Deep Learning for i-Vectors

The success of the i-vector approach in speaker recognition and DL techniques in speech processing applications has encouraged the research community to combine those techniques for speaker recognition [17]–[19], [21], [22], [24], [27], [43]. Two kinds of combination can be considered. DL techniques can be used in the i-vector extraction process [21], [22], [24], [27], [43], or applied after i-vector computation [9]–[11], [17]–[19]. In order to have more insight into the behavior of DL techniques on i-vectors, in this work the authors extend the preliminary study developed in [17], [18].

An i-vector [1] is a low rank vector, typically between 400 and 600, representing a speech utterance. Feature vectors of a speech signal can be modeled by a set of Gaussian Mixtures (GMM) adapted from a Universal Background Model (UBM). The mean vectors of the adapted GMM are stacked to build the supervector **m**. The supervector can be further modeled as follows,

$$\mathbf{m} = \mathbf{m}_u + \mathbf{T}\boldsymbol{\omega} \qquad (2)$$

where $\mathbf{m}_u$ is the speaker- and session-independent mean supervector typically from UBM, $\mathbf{T}$ is the total variability matrix, and $\boldsymbol{\omega}$ is a hidden variable. The mean of the posterior distribution of $\boldsymbol{\omega}$ is referred to as i-vector. This posterior distribution is conditioned on the Baum-Welch statistics of the given speech utterance. The $\mathbf{T}$ matrix is trained using the Expectation-Maximization (EM) algorithm given the centralized Baum-Welch statistics from background speech utterances. More details can be found in [1].

In the state-of-the-art speaker recognition systems, all the speech utterances, including background, train and test, are converted to i-vectors. Background and train i-vectors are typically called impostor and target i-vectors, respectively. The main objective in this work is to train a two-class DNN for each target speaker given target and impostor i-vectors. In the single-session target speaker enrollment task, only one i-vector is available, meanwhile in the multi-session one, several i-vectors are available per each target speaker. DNNs and in
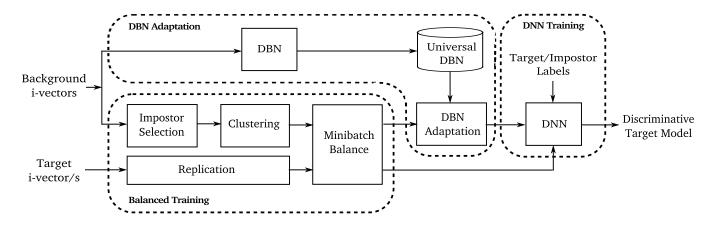
Fig. 3: Block-diagram of the proposed DNN based speaker recognition system.

general neural networks usually need a large number of input samples to be trained, and deeper networks require more input data. The lack of enough target samples for training each DNN yields two main problems. Firstly, the number of target and impostor samples will be highly unbalanced, one or some few target samples against thousands of impostor samples. Learning from such unbalanced data will result in biased DNNs towards the majority class. In other words, DNNs will have a much higher prediction accuracy over the majority class. Secondly, as we need to decrease the number of impostor samples to solve the first problem, the total number of samples for training the network will be very few. A network trained with such few data is highly probable to be overfitted.

Fig. 3 shows the block diagram of the proposed approach to discriminatively model target speakers. In this block diagram, we propose two main contributions to tackle the above problems. The balanced training block attempts to decrease the number of impostor samples and, in the contrary, to increase the number of target ones in a reasonable and effective way. The most informative impostor samples for target speakers are first selected by the proposed impostor selection algorithm. After that, the selected impostors are clustered and the cluster centroids are considered as final impostor samples for each target speaker model. Impostor centroids and target samples are then divided equally into minibatches to provide balanced impostor and target data in each minibatch.

On the other hand, the DBN adaptation block is proposed to compensate the lack of enough input data. As DBN training does not need any labeled data, the whole background i-vectors are used to build a global model, which is referred to as Universal DBN (UDBN). The parameters of UDBN are then adapted to the balanced data obtained for each target speaker. It is worth noting that as the minimum divergence training algorithm [44] is used in the i-vector extraction process, i-vectors will have a standard normal distribution $\mathcal{N}(0,1)$. Therefore, they will be compatible with Gaussian RBMs (GRBM) in DBN architectures, which assume a zero-mean unit-variance normal distribution for inputs. At the end, given target/impostor labels, adapted DBN, and balanced data, a DNN is discriminatively trained for each target speaker. In the two following sections, we will describe these two main

contributions in more details.

IV. BALANCED TRAINING

As 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 highly unbalanced in this case which yields biasing towards the majority class. Some of the most straightforward ways to deal with unbalanced data problem are explored in [45]–[47] [48], [49]. One of the simplest commonly used method is data sampling. The data of the majority class is undersampled and, in the contrary, the data of the minority class is oversampled. The effectiveness of these techniques is highly dependent on the data structure.

In the proposed approach in Fig. 3, the amount of impostors is decreased in two steps, namely selection and clustering. On the other hand, the amount of target samples is increased by either replication or combination. After that, balanced target and impostor samples are distributed equally among minibatches.

A. Impostor Selection and Clustering

The objective is to decrease the large number of negative samples in a reasonable way. Our proposal has two main steps. First, only those impostor i-vectors which are more informative for the training dataset are selected. Informative impostor means, in this case, the impostor which is not only representative to a given target but also is close statistically to other targets in the dataset. For a real application, it makes sense to select those impostors who are globally close to all enrolled speakers. When the target speakers are changed, the selected impostors can be reselected according to the new target dataset. Second, as the number of selected impostor samples is still high in comparison to the number of target ones, they are clustered by the k-means algorithm using the cosine distance criterion. The centroids of the clusters are then used as the final negative samples.

The selection method is inspired from a data-driven background data selection technique proposed in [50]. In that technique, given all available impostors a Support Vector Machine
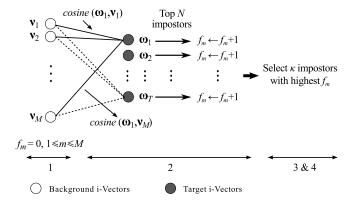
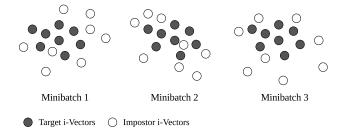Fig. 4: Steps of proposed impostor selection algorithm.

Fig. 5: Balanced training for DNNs in multi-session speaker verification task. In each minibatch the same target i-vectors but different impostors are shown to DNNs.

(SVM) classifier is trained for each target speaker. The number of times each impostor is selected as a support vector, in all training SVM models, is called impostor support vector frequency [50]. Impostor examples with higher frequencies are then selected as the refined impostor dataset. However, SVM training for each target speaker would be computationally costly. Moreover, as our final discriminative models will be DNNs, it would not be worth to employ this technique as such. Instead, we have proposed to use cosine distance as an efficient and a fast criterion for comparing i-vectors. We compare each target i-vector with all impostor ones in the background dataset. Those N impostors which are close to each target i-vector are treated like support vectors in [50]. Then the κ impostors with higher frequencies are selected as the most informative ones. The parameters N and κ are determined experimentally. The whole algorithm is shown in Fig. 4 and can be summarized as follows,

1) Set impostor frequencies $f_m = 0$, $1 \leq m \leq M$
2) For each target i-vector $\boldsymbol{\omega}_t$, $1 \leq t \leq T$
   a) Compute $cosine\,(\boldsymbol{\omega}_t, \boldsymbol{\nu}_m)$, $1 \leq m \leq M$
   b) Select the N impostors with the highest scores
   c) For the selected impostors $f_m \leftarrow f_m + 1$
3) Sort impostors descendingly based on their f_m
4) Select the first κ impostors as the final ones.

where $cosine\,(\boldsymbol{\omega}_t, \boldsymbol{\nu}_m)$ is the cosine score between target i-vector $\boldsymbol{\omega}_t$ and the impostor i-vector $\boldsymbol{\nu}_m$ in the background dataset, M and T are the number of impostor and target speakers, respectively. It is worth noting that in the case of multi-session target enrollment, the average of the i-vectors available per each target speaker will be considered in the above selection algorithm.

B. Target Replication or Combination

In order to balance positive and negative samples, the number of target samples is increased as many as the number of impostor cluster centroids obtained in section IV-A. In the single session enrollment task, the i-vector of each target speaker is simply replicated as many as the number of cluster centroids. Replicated target i-vectors will not act exactly the same as each other in the pre-training process of DNNs due to the sampling noise created in RBM training [42]. Moreover, in

both adaptation and supervised learning stages the replicated versions make the target and impostor classes having the same weights when the network parameters are being updated. In multi-session task, the available i-vectors of each target speaker can be combined, i.e., the average of every n i-vectors is considered as a new target i-vector.

Once the number of positive and negative samples are balanced, they are distributed equally among minibatches. In other words, each minibatch contains the same number of impostors and targets. If target samples in the multi-session task are not combined, the same target samples but different impostor ones are shown to the network in each minibatch (Fig. 5). The optimum numbers of impostor clusters and minibatches will be determined experimentally in section. VI.

V. UNIVERSAL DBN AND ADAPTATION

Unlike DNNs, which need labeled data for training, DBNs do not necessarily need such labeled data as inputs. Hence, they can be used for unsupervised training of a global model referred to as Universal DBN (UDBN) [17]. UDBN is trained by feeding background i-vectors from different speakers. The training procedure is carried out layer by layer using RBMs as described in Sec. II.

It is shown that pre-training techniques can initialize DNNs better than simply random numbers [38]–[40]. However, when a few numbers of input samples are available, just pre-training may not be enough to achieve a good model. In this case, we have proposed in [17] to adapt UDBN parameters to the balanced data obtained for each target speaker. Adaptation is carried out by pre-training each network initialized by UDBN parameters. In order to avoid overfitting, only a few iterations will be used for adaptation. It is supposed that UDBN can learn both speaker and channel variabilities from the background data. Therefore, UDBN will provide a more meaningful initial point for each target model than a simple random initialization. The study in [39] has shown that pre-training is robust with respect to the random initialization seed. The use of UDBN parameters makes target models almost independent from the random seeds.

In contrast to [17], [18], in this work we normalize the UDBN parameters before adaptation. Normalization is carried out by simply scaling down the maximum absolute value of connection weights to 0.01. In this way, connection weights will have a dynamic range similar to that typically used
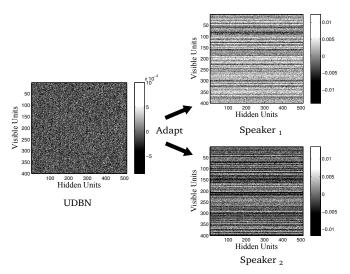
Fig. 6: Comparison of the adapted connection weights between the visible and the first hidden units for two different speakers.

for random initialization. Correspondingly, bias terms are multiplied by 0.01 to be closer to zero. This is because the bias terms are usually set to zero when the connection weights are randomly initialized. UDBN parameter normalization facilitates training the networks specifically where more than one hidden layer is used. In this way, the same learning rates and the number of epochs tuned for random initialized DNNs can also be used for adapted DNNs in the supervised learning stage.

Fig. 6 compares the adapted UDBN connection weights between the input layer and the first hidden one for two different speakers. As it can be seen in this figure, adaptation sets speaker-specific initial points for each model.

Once the adaptation process is completed, a DNN is initialized with the adapted DBN parameters for each target speaker. Then the minibatch stochastic gradient descent back-propagation is carried out for fine-tuning. The softmax and the logistic sigmoid will be the activation functions of the top label layer and the other hidden layers, respectively.

If the input labels in the training phase are chosen as $(1, 0)$ and $(0, 1)$ 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,

$$LLR = \log(o_1) - \log(o_2) \qquad (3)$$

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 regularization is used to penalize large weights. The top layer pre-training proposed in [17] is not used in this work. The reason is that it gives the emphasis on the top layer connection weights and avoids the lower layers, closer to the input, to learn enough from the input data. This fact will be more important when higher number of hidden layers are used.

VI. EXPERIMENTAL RESULTS

The block-diagram of Fig. 3 has been implemented for both single and multi-session speaker verification tasks. The effectiveness of two main contributions proposed in the figure will be shown in this section.

A. Baseline and Dataset

All the experiments in this work are carried out on the NIST SRE 2006 evaluation [31]. In both training and testing phases signals have approximately two-minute total speech duration. The whole core condition has been used for the single session task, in which there are 816 target models and 51,068 trials. On the other hand, 8 conversation sides are available per each target speaker in the multi-session task and the protocol contains 699 target models and 31,080 trials. NIST SRE 2004 and 2005 are used as the background data. It is worth noting that in the case of NIST 2005 only the speech signals of those speakers who do not appear in NIST SRE 2006 are used.

Frequency Filtering (FF) features [51] are used in the experiments. FFs, like MFCCs, are decorrelated version of log Filter Bank Energies (FBE) [51]. It has been shown that FF features achieve a performance equal to or better than MFCCs [51]. Features are 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 log energy, 33-dimensional feature vectors are produced. Before feature extraction, speech signals are subjected to an energy-based silence removal process.

The gender-independent UBM is represented as a diagonal covariance, 512-component GMM. All the i-vectors are 400-dimensional. The i-vector extraction process is carried out using ALIZE open source software [52]. UBM and **T** matrix are trained using more than 6,000 speech signals collected from NIST SRE 2004 and 2005. Performance is evaluated using the Detection Error Tradeoff (DET) curves, the Equal Error Rate (EER), and the minimum of the Decision Cost Function (minDCF) calculated using $C_M = 10, C_{FA} = 1, P_T = 0.01$.

It is supposed in all experiments that there is no labeled background data and, therefore, no channel compensation technique is used. The aim of the work is to show how DNN architectures can be more efficient in this particular task. In the baseline systems, whitened and length normalized i-vectors are classified using cosine distance. In the multi-session task, the average of the available i-vectors per each target speaker is first length normalized and then compared with the test i-vector using cosine distance. In DNN experiments, raw i-vectors without whitening and length normalization are used.

B. Single-Session Experiments

At first, we need to choose the size of DNNs in terms of the hidden layer size and the number of layers. The number of hidden units in each layer is taken as a power of 2 greater than the input layer size. Since the input layer size is 400, the hidden layer size is chosen as 512. We explore DNNs with up to three hidden layers in all experiments. We do not go further than three layers because the computational complexity is increased considerably and also no significant gain is observed.
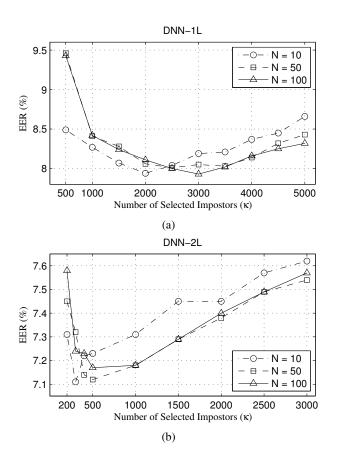
DNN–1L

DNN–2L

Fig. 7: Determination of the parameters of the proposed impostor selection algorithm for (a) 1 hidden layer and (b) 2 hidden layer DNNs. N and κ are, respectively, the number of local and global nearest impostor i-vectors to the target i-vectors.

As it is shown in Fig. 3, the first step to train DNNs is to balance the number of target and impostor input i-vectors in each minibatch. The number of minibatches and the number of impostor centroids are set experimentally to 3 and 12, respectively. Each minibatch will include four impostor centroids and four replicated target samples.

After that, we train a DNN for each target speaker using the whole impostor background data and random initialization. In this case, the whole background i-vectors are clustered using the k-means algorithm and the centroids are considered as impostor samples. In this work, we use the uniform distribution $\mathcal{U}(0, 0.01)$ for random initialization as the experimental results showed that it achieves slightly better performance than the normal distribution $\mathcal{N}(0, 0.01)$ used in the prior work [17]. We tune the parameters of the networks and keep them fixed in all other experiments. One, two, and three hidden layer DNNs are trained with the learning rates of 0.001, 0.005, and 0.08 and with the number of epochs of 30, 100, and 500, respectively. Momentum and weight decay are set, respectively, to 0.9 and 0.0012 for all DNNs. In the following, we show the effect of each contribution proposed in Sec. III.

Background i-vectors are extracted from the same speech signals used for training UBM and **T** matrix. The two parameters N and κ, the number of local and global selected impostors in the proposed impostor selection method, need to be determined experimentally. Figures. 7a and 7b illustrate the variability of EER in terms of these two parameters for one and two hidden layer DNNs, respectively. The same behavior can be observed for minDCF curves. DNN with three hidden layers act similar to two-layer ones. DNN examples shown in these two figures are initialized randomly. As it can be seen, DNNs with more than one hidden layer tend to achieve better results with fewer number of global selected impostors in comparison to networks with only one hidden layer. In all cases, the best performance is obtained by selecting fewer local impostors. Hence, for all DNNs N is set to 10 and κ is set to 2,000, 300, and 500 for one, two, and three layer DNNs, respectively.

UDBN is trained with the same background i-vectors used for impostor selection. As the input i-vectors are real-valued, a Gaussian-Bernoulli RBM (GRBM) [5], [42] is used to train the connection weights between the visible and the first hidden layer units. The rest of the connection weights are trained with Bernoulli-Bernoulli RBMs. The learning rate and the number of epochs are set to 0.014 and 200 for GRBM, and to 0.06 and 120 for the rest of RBMs in UDBN, respectively. Momentum and weight decay are set, respectively, to 0.9 and 0.0002 for all RBMs. Unlike in [17] and [18] where the authors used the UDBN parameters as such, in this work we normalize the connection weights so that the maximum absolute value is 0.01. This is the maximum value of the random numbers we used to initialize DNNs. Additionally, we scale down the bias terms by 0.01. Normalization helps to facilitate training DNNs with higher number of layers. Moreover, we can use the same learning rates we tuned for random initialized DNNs.

Experimental results showed that the most part of the improvement due to the adaptation process comes from the adaptation of the connection weights between the input layer and the first hidden layer for all DNNs. The adaptation of the other layers has no positive effect or it improves the performance slightly. We adapted the networks up to two layers. The learning rate of adaptation is set to 0.001 and 0.0001 for the first and the second layers, respectively. The number of epochs for the first layer is set to 10, 20, and 15 for DNN-1L, -2L, and -3L, respectively. DNN-1L stands for a one hidden layer DNN. The number of epochs for the second layer is set, respectively, to 15 and 20 for DNN-2L and DNN-3L.

Table I summarizes the effect of each proposed contribution. In the first row of the table, DNNs are initialized randomly and the impostor cluster centroids are obtained on the whole background data. As it can be seen in this row, adding more hidden layers to the network improves the performance. However, they still work worse than the baseline system in which i-vectors are classified using cosine distance. EER and minDCF for the baseline system are 7.18% and 0.0324, respectively. Impostor selection improves the performance to a great extent for all the networks. However, the biggest improvement due to the adaptation process is observed in DNNs with one hidden layer. The best results are obtained using both impostor selection and adaptation techniques which show an 8-20% and 10-17% relative improvements in terms of EER and minDCF, respectively, in comparison to the conventional DNNs. The

TABLE I: The effect of each proposed idea of Fig. 3 on the performance of the DNN target models. Results are obtained on the core test condition of NIST SRE 2006. Baseline is classification of i-vectors using cosine distance with EER=7.18 and minDCF=324.

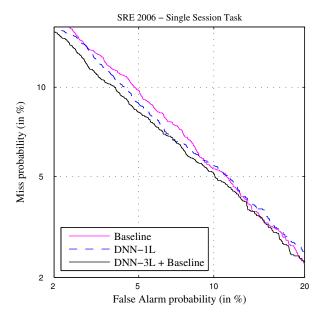| Impostor Selection	Adaptation	EER (%)			minDCF ($\times 10^4$)		
		# Hidden Layers			# Hidden Layers		
		1	2	3	1	2	3
No	No	8.55	7.76	7.59	381	353	351
Yes	No	8.06	7.12	7.09	360	327	326
No	Yes	7.43	7.47	7.45	339	343	339
Yes	Yes	**6.81**	6.97	6.99	315	317	313
Fusion with Baseline		6.83	**6.88**	**6.64**	**308**	**309**	**299**

Fig. 8: Comparison of the performance of the proposed DNN based systems with the baseline system (i-vector + cosine). DET curves are obtained on the core test condition of NIST SRE 2006.

last row of the table shows the fusion of DNN systems with the baseline in the score level. Scores of each system are first mean and variance normalized and then summed. Although DNNs with one hidden layer yield slightly better results, DNNs with more layers provide complementary information to the baseline system. This confirms the theoretical hypothesis which states that more hidden layers more abstractions from the input data. The fusion of baseline and DNNs with three hidden layers achieves the best results corresponding to an 8% relative improvement for both EER and minDCF in comparison to the baseline system. We have also combined the scores of DNNs with different number of hidden layers, but no gain is observed.

The DET curve in Fig. 8 compares the best systems in all operating points. As it is shown in this figure, DNNs with one hidden layer achieve better results than the baseline and the combination of 3-layer DNNs with the baseline works the best in all operating points.

C. Multi-Session Experiments

The same configuration used for the single session task is also applied for the multi-session one. The number of minibatches is set to 3. In each minibatch, all 8 target i-vectors accompanying with 8 impostor cluster centroids are shown to the network. Therefore, the size of each minibatch and the total number of impostor clusters will be 16 and 24, respectively. As the combination of the i-vectors of each target speaker did not help the training of the networks, we replicated the target i-vectors in every minibatch as it was shown in Fig. 5.

We start training the networks with the same parameters tuned for the single session experiments. However, as the target i-vector samples per each training speaker are different from each other in the multi-session task, the number of epochs and the learning rates need to be slightly re-tuned. We have set the learning rates to 0.001, 0.01, and 0.08 and the number of epochs to 30, 100, and 500 for one layer to three layer DNNs, respectively.

Results are summarized in Table II. Around 12% relative improvements are achieved in all DNNs employing impostor selection technique proposed in this work. With the same parameters obtained for the single session task, we re-selected the impostors for the new multi-session data set. The adaptation process improves the performance up to 8%. As in the single session task, adaptation is more effective in the one hidden layer DNNs. For all the networks, only the parameters of the first hidden layer are adapted because no more improvement was observed adapting the other layers. Adaptation is carried out by the learning rate of 0.001 for all DNNs and the number of epochs of 10, 10, and 25 for DNNs with one to three layers, respectively. The best results are obtained with three layer DNNs when the two proposed techniques are used together. It shows more than 20% improvement of EER and minDCF in comparison to the conventional three layer DNNs. Compared to the baseline system in which EER and minDCF are obtained 4.2% and 0.0191, respectively, the proposed three hidden layer DNNs achieve more than 17% and 10% improvements in terms of EER and minDCF, respectively. Fusion with the baseline system at the score level improves the results in all cases. Fusion is effective mostly on the minDCF which increase the improvement from 10% to 15%.

Fig. 9 compares the DET curves of the best results obtained in table II. As it can be seen in this figure, DNN-3L outperforms clearly the baseline and the DNN-1L in all operating points. However, fusion with the baseline system improves the performance only for the operating points with higher false alarm probabilities.

VII. CONCLUSION

A hybrid system based on Deep Belief Networks (DBN) and Deep Neural Networks (DNN) has been proposed in this work for speaker recognition to discriminatively model target speakers with available i-vectors. In order to have more insight into the behavior of these techniques in both single and multi-session speaker enrollment tasks, the experiments are carried out in both scenarios. Two main contributions have been proposed to make DNNs more efficient in this

TABLE II: The effect of each proposed idea of Fig. 3 on the performance of the DNN target models. Results are obtained on NIST SRE 2006, 8-session enrollment task. Baseline is classification of i-vectors using cosine distance with EER=4.2 and minDCF=191.

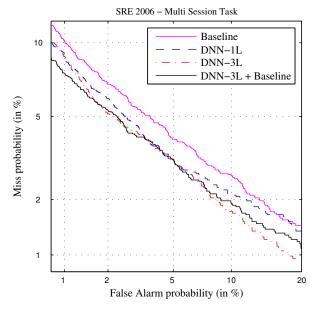| Impostor Selection	Adaptation	EER (%)			minDCF ($\times 10^4$)		
		# Hidden Layers			# Hidden Layers		
		1	2	3	1	2	3
No	No	4.58	4.58	4.38	208	213	217
Yes	No	4.02	4.07	3.86	183	201	194
No	Yes	4.24	4.30	4.20	202	207	202
Yes	Yes	3.68	3.83	3.50	170	189	172
Fusion with Baseline		**3.61**	**3.77**	**3.45**	**161**	**169**	**162**

Fig. 9: Comparison of the performance of the proposed DNN based systems with the baseline system (i-vector + cosine). DET curves are obtained on the 8-session enrollment task of NIST SRE 2006.

particular task. Firstly, the most informative impostors have been selected and clustered to provide a balanced training. Secondly, each DNN has been initialized with the speaker specific parameters adapted from a global model, which has been referred to as Universal DBN (UDBN). The parameters of UDBN are normalized before adaptation, which facilitates the training of DNNs specifically with more than one hidden layer. Experiments are performed on the NIST SRE 2006 corpus. It was shown that in both scenarios the proposed architectures outperform the baseline systems with up to 17% and 10% in terms of EER and minDCF, respectively.

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] 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.

[5] 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.

[6] 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.

[7] 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.

[8] A. Senior, H. Sak, and I. Shafran, "Context Dependent Phone Models For LSTM RNN Acoustic Modelling," in *Proc. ICASSP*, 2015, pp. 4585–4589.

[9] T. Stafylakis, P. Kenny, M. Senoussaoui, and P. Dumouchel, "Preliminary investigation of boltzmann machine classifiers for speaker recognition," in *Proc. Odyssey*, 2012.

[10] M. Senoussaoui, N. Dehak, P. Kenny, R. Dehak, and P. Dumouchel, "First attempt of boltzmann machines for speaker verification," in *Proc. Odyssey*, 2012.

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

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

[13] S. Novoselov, T. Pekhovsky, and K. Simonchik, "STC speaker recognition system for the NIST i-vector challenge," in *Odyssey: The Speaker and Language Recognition Workshop*, 2014, pp. 231–240.

[14] O. Ghahabi and J. Hernando, "Restricted boltzmann machine supervectors for speaker recognition," in *Proc. ICASSP*, 2015, pp. 4804–4808.

[15] 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. 10961104, 2009.

[16] P. Safari, O. Ghahabi, and J. Hernando, "Feature classification by means of deep belief networks for speaker recognition," in *Proc. EUSIPCO*, 2015, pp. 2162–2166.

[17] O. Ghahabi and J. Hernando, "Deep belief networks for i-vector based speaker recognition," in *Proc. ICASSP*, May 2014, pp. 1700–1704.

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

[19] 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*, Lecture Notes in Artificial Intelligence. Springer International Publishing, Nov. 2014.

[20] W. M. Campbell, "Using deep belief networks for vector-based speaker recognition," in *Proc. Interspeech*, 2014, pp. 676–680.

[21] Y. Lei, N. Scheffer, L. Ferre, and M. Mclaren, "A novel scheme for speaker recognition using a phonetically-aware deep neural network," in *Proc. ICASSP*, 2014.

[22] P. Kenny, V. Gupta, T. Stafylakis, P. Ouellet, and J. Alam, "Deep neural networks for extracting baum-welch statistics for speaker recognition," in *Proc. Odyssey*, 2014, pp. 293–298.

[23] D. Garcia-Romero, Xiaohui Zhang, A. McCree, and D. Povey, "Improving speaker recognition performance in the domain adaptation challenge using deep neural networks," in *2014 IEEE Spoken Language Technology Workshop (SLT)*, Dec. 2014, pp. 378–383.

[24] M. Mclaren, Y. Lei, and L. Ferre, "Advances In Deep Neural Network Approaches To Speaker Recognition," in *Proc. ICASSP*, 2015.

[25] F. Richardson, D. Reynolds, and N. Dehak, "Deep Neural Network Approaches to Speaker and Language Recognition," *IEEE Signal Processing Letters*, vol. 22, no. 10, pp. 1671–1675, Oct. 2015.

[26] E. Variani, Xin Lei, E. McDermott, I. Lopez Moreno, and J. Gonzalez-Dominguez, "Deep neural networks for small footprint text-dependent speaker verification," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2014, pp. 4052–4056.

[27] Yuan Liu, Yanmin Qian, Nanxin Chen, Tianfan Fu, Ya Zhang, and Kai Yu, "Deep feature for text-dependent speaker verification," *Speech Communication*, vol. 73, pp. 1–13, Oct. 2015.

[28] "The NIST speaker recognition i-vector machine learning challenge," 2014, [Online]. Available: http://nist.gov/itl/iad/mig/upload/sre-ivectorchallenge_2013-11-18_r0.pdf.

[29] "The NIST language recognition i-vector machine learning challenge," 2015, [Online]. Available: http://www.nist.gov/itl/iad/mig/upload/lre_ivectorchallenge_rel_v2.pdf.

[30] E. Khoury, L. El Shafey, M. Ferras, and S. Marcel, "Hierarchical speaker clustering methods for the nist i-vector challenge," in *Odyssey: The Speaker and Language Recognition Workshop*, 2014, pp. 254–259.

[31] "The NIST year 2006 speaker recognition evaluation plan," 2006, [Online]. Available: http://www.nist.gov/speech/tests/spk/2006/index.htm.

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

[33] 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.

[34] 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.

[35] 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.

[36] Tara N. Sainath, Brian Kingsbury, George Saon, Hagen Soltau, Abdel-rahman Mohamed, George Dahl, and Bhuvana Ramabhadran, "Deep Convolutional Neural Networks for Large-scale Speech Tasks," *Neural Networks*, vol. 64, pp. 39–48, Apr. 2015.

[37] Zhen-Hua Ling, Shi-Yin Kang, Heiga Zen, A. Senior, M. Schuster, Xiao-Jun Qian, H.M. Meng, and Li Deng, "Deep Learning for Acoustic Modeling in Parametric Speech Generation: A systematic review of existing techniques and future trends," *IEEE Signal Processing Magazine*, vol. 32, no. 3, pp. 35–52, May 2015.

[38] H. Larochelle, Y. Bengio, J. Louradour, and P. Lamblin, "Exploring Strategies for Training Deep Neural Networks," *Journal of Machine Learning Research*, vol. 10, pp. 1–40, June 2009.

[39] E. Dumitru, P. Manzagol, Y. Bengio, S. Bengio, and P. Vincent, "The Difficulty of Training Deep Architectures and the Effect of Unsupervised Pre-Training," in *The Twelfth International Conference on Artificial Intelligence and Statistics (AIST ATS09)*, pp. 153–160.

[40] D. Erhan, Y. Bengio, A. Courville, P. Manzagol, P. Vincent, and S. Bengio, "Why Does Unsupervised Pre-training Help Deep Learning?," *J. Mach. Learn. Res.*, vol. 11, pp. 625–660, Mar. 2010.

[41] L. Deng and D. Yu, *Deep Learning: Methods and Applications*, Now Publishers Inc, June 2014.

[42] 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.

[43] R. Fred, R. Douglas, and N. Dehak, "A Unified Deep Neural Network for Speaker and Language Recognition," *arXiv:1504.00923 [cs, stat]*, 2015, arXiv: 1504.00923.

[44] 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.

[45] H. He and E.A. Garcia, "Learning from imbalanced data," *IEEE Transactions on Knowledge and Data Engineering*, vol. 21, no. 9, pp. 1263–1284, Sept. 2009.

[46] N. Thai-Nghe, Z. Gantner, and L. Schmidt-Thieme, "Cost-sensitive learning methods for imbalanced data," in *The 2010 International Joint Conference on Neural Networks (IJCNN)*, July 2010, pp. 1–8.

[47] T.M. Khoshgoftaar, J. Van Hulse, and A. Napolitano, "Supervised neural network modeling: An empirical investigation into learning from imbalanced data with labeling errors," *IEEE Transactions on Neural Networks*, vol. 21, no. 5, pp. 813–830, May 2010.

[48] V. Lpez, A. Fernndez, S. Garca, V. Palade, and F. Herrera, "An insight into classification with imbalanced data: Empirical results and current trends on using data intrinsic characteristics," *Information Sciences*, vol. 250, pp. 113–141, Nov. 2013.

[49] S. Barua, M.M. Islam, Xin Yao, and K. Murase, "MWMOTE-majority weighted minority oversampling technique for imbalanced data set learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 2, pp. 405–425, Feb. 2014.

[50] M. McLaren, R. Vogt, B. Baker, and S. Sridharan, "Data-driven background dataset selection for SVM-based speaker verification," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 6, pp. 1496–1506, Aug. 2010.

[51] 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.

[52] 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.

Omid Ghahabi received the M.Sc. Degree in electrical engineering from Shahid Beheshti University, Tehran, Iran, in 2009. From 2009 to 2011, he has been with the speech processing group of the Research Center for Intelligent Signal Processing (RCISP), Tehran, Iran. He is now a Ph.D. candidate at Technical University of Catalonia (UPC)-BarcelonaTech, Spain. Since 2011, he has been working as a researcher in the speech processing group of the Signal Theory and Communications Department of UPC. He is also a member of the Research Center for Language and Speech Technologies and Applications (TALP), Barcelona, Spain. His research interests include, but not limited to, speaker recognition, speech signal processing, and deep learning. He is the author and coauthor of several journal and conference papers on these topics.

Javier Hernando received the M.S. and Ph.D. degrees in telecommunication engineering from the Technical University of Catalonia (UPC), Barcelona, Spain, in 1988 and 1993, respectively. Since 1988, he has been with the Department of Signal Theory and Communications, UPC, where he is a Professor and a member of the Research Center for Language and Speech (TALP). He was a Visiting Researcher at the Panasonic Speech Technology Laboratory, Santa Barbara, CA, during the academic year 2002-2003. His research interests include robust speech analysis, speaker recognition, speaker verification and localization, oral dialogue, and multimodal interfaces. He is the author or coauthor of about two hundred publications in book chapters, review articles, and conference papers on these topics. He has led the UPC team in several European, Spanish and Catalan projects. Prof. Hernando received the 1993 Extraordinary Ph.D. Award of UPC.