CONTEXT ADAPTIVE DEEP NEURAL NETWORKS FOR FAST ACOUSTIC MODEL ADAPTATION

Marc Delcroix, Keisuke Kinoshita, Takaaki Hori, Tomohiro Nakatani

NTT Communication Science Laboratories, NTT corporation,
2-4, Hikaridai, Seika-cho (Keihanna Science City), Soraku-gun, Kyoto 619-0237 Japan
{marc.delcroix, kinoshita.k, hori.t, nakatani.tomohiro}@lab.ntt.co.jp

ABSTRACT

Deep neural networks (DNNs) are widely used for acoustic modeling in automatic speech recognition (ASR), since they greatly outperform legacy Gaussian mixture model-based systems. However, the levels of performance achieved by current DNN-based systems remain far too low in many tasks, e.g. when the training and testing acoustic contexts differ due to ambient noise, reverberation or speaker variability. Consequently, research on DNN adaptation has recently attracted much interest. In this paper, we present a novel approach for the fast adaptation of a DNN-based acoustic model to the acoustic context. We introduce a context adaptive DNN with one or several layers depending on external factors that represent the acoustic conditions. This is realized by introducing a factorized layer that uses a different set of parameters to process each class of factors. The output of the factorized layer is then obtained by weighted averaging over the contribution of the different factor classes, given posteriors over the factor classes. This paper introduces the concept of context adaptive DNN and describes preliminary experiments with the TIMIT phoneme recognition task showing consistent improvement with the proposed approach.

Index Terms— Automatic speech recognition, Deep neural networks, Acoustic model adaptation, Context adaptive DNN, Factorized DNN

1. INTRODUCTION

Recently, the introduction of deep neural network (DNN) based acoustic modeling has greatly improved the performance of automatic speech recognition (ASR) for various tasks [1]. However, there still remains a great performance gap between top performances obtained under well-controlled conditions and performances achieved in the presence of noise, reverberation or speaker mismatch.

Acoustic model adaptation is usually used to adjust the acoustic model to the testing conditions. For example, maximum likelihood linear regression (MLLR) has been shown to be very effective for speaker or environment adaptation when employing legacy Gaussian mixture model hidden Markov model (GMM-HMM) based ASR systems [2]. Research on adaptation for DNN-HMM acoustic models has attracted a lot of attention [3–18]. However, there is still no consensus on how to perform efficient adaptation in the context of DNN-based acoustic models. Several approaches for DNN adaptation have been investigated, including input feature normalization [4–8], direct adaptation of the DNN parameters [9–14, 19] and the use of rich input features that explicitly characterize acoustic conditions such as i-vectors or noise features [15–18].

In this paper, we propose a different approach that uses a DNN whose parameters are directly dependent on factors that characterize the acoustic context. We refer to this network as *context adaptive DNN*. Note that here the term context denotes the long-term acoustic conditions that are typically defined on an utterance level, e.g. speakers or acoustic environments. The structure of a context adaptive DNN is realized by dividing one or several hidden layers of the network into a set of parallel sub-layers each associated with a different factor class. By an abuse of terminology, we call such a layer a *factorized layer*. The input of a factorized layer is the output activation of the previous layer as with conventional DNN. The input is processed with each sub-layer in parallel. The output of the factorized layer is then obtained by the *weighted averaging* of the output of each sub-layer, weighted by the posterior probabilities of the factor classes. During training, the parameters of the factorized layer are trained in a soft manner, using training data and the associated factor class posteriors. During testing, a DNN adapted to the test conditions can be obtained given the class posteriors, by the weighted averaging of the parameters of the factorized layer. Consequently, this makes fast adaptation possible even when there are many parameters to adapt. Moreover, the factors can be estimated blindly during testing, enabling unsupervised adaptation.

The topology of the proposed context adaptive DNN is similar to that of networks employed for committee machines, which are used to combine the outputs of different experts [20, 21]. In particular, a similar weighted averaging was used for a of mixture of expert models that employ a gating network to calculate the weights used to combine the outputs of different experts. An equivalent approach referred to as disjoint factorized DNN was investigated in relation to acoustic model adaptation for ASR [22]. A notable difference between the proposed context adaptive DNN and the approaches proposed in [20–22] is that instead of using a gating network to obtain the posteriors, we use posteriors calculated externally, which enables us to represent the long-term acoustic context.

In this paper, we introduce the concept of context adaptive DNNs and detail our implementation. We also provide preliminary experimental results for gender adaptation on the TIMIT corpus. The proposed approach can perform similarly to a gender dependent system, without using prior knowledge about genders.

The remainder of this paper is as follows. In Section 2, we introduce the proposed context adaptive DNN. Section 3 elaborates on the relationship between the proposed approach and previous work on DNN adaptation. We then present preliminary experimental results using the TIMIT corpus in Section 4. Finally, Section 5 concludes the paper and discusses future work directions.

978-1-4673-6997-8/15/$31.00 ©2015 IEEE

2. CONTEXT ADAPTIVE DNN

2.1. Overview

Before introducing the proposed context adaptive DNN, we first review a conventional DNN to introduce the notations used in this paper. Figure 1-(a) is a schematic diagram of a conventional DNN. To emphasize the differences between a conventional DNN and the proposed context adaptive DNN, Fig. 1 explicitly shows the linear transformation and the activation function associated with each hidden layer. We use $\mathbf{x}^{(i-1)}$ to denote the input of the i^{th} layer of a DNN, where by definition $\mathbf{x}^{(0)}$ corresponds to the input features or input layer. The output of the i^{th} layer is given by,

$$\begin{aligned} \mathbf{x}^{(i)} &= \sigma(\mathbf{z}^{(i)}), \\ \mathbf{z}^{(i)} &= \mathbf{W}^{(i)}\mathbf{x}^{(i-1)} + \mathbf{b}^{(i)}, \end{aligned} \qquad (1)$$

where $\mathbf{W}^{(i)}$ and $\mathbf{b}^{(i)}$ are the weight matrix and bias vector of the linear transformation associated with the i^{th} layer, and $\sigma()$ is the activation function, which is typically a sigmoid function [23].

The proposed context adaptive DNN replaces one or several layers of the DNN with factorized layers. A factorized layer is realized by decomposing the linear transformation of a hidden layer into sub transformations each associated with a different factor class. During propagation, the parameters of the factorized layer are obtained as the weighted sum of the parameters associated with each factor class, weighted by the posterior probabilities of the factor classes. The parameters associated with the i^{th} factorized layer can thus be expressed as,

$$\begin{cases} \mathbf{W}^{(i)} &= \sum_{k=1}^{K} \alpha_k \mathbf{W}^{(i)}{}_k, \\ \mathbf{b}^{(i)} &= \sum_{k=1}^{K} \alpha_k \mathbf{b}^{(i)}{}_k, \end{cases} \qquad (2)$$

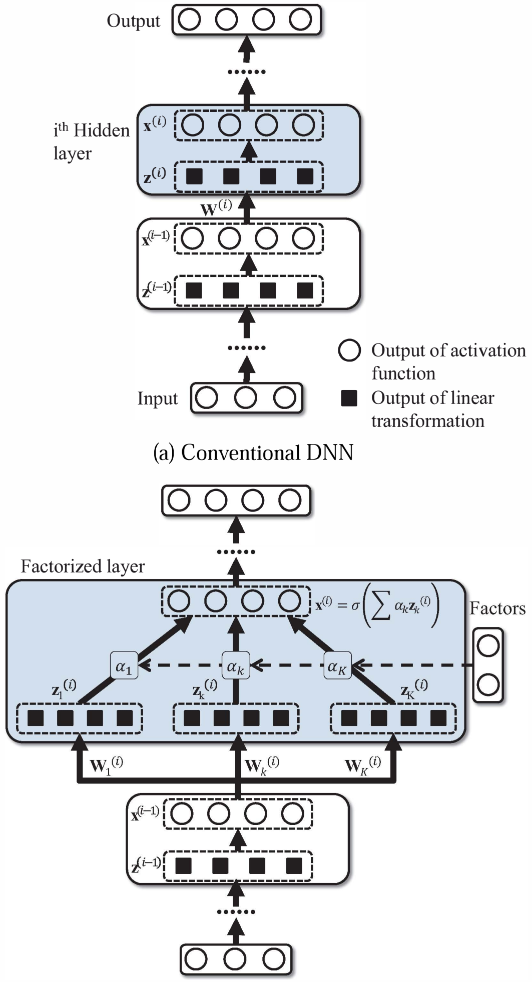
where $\mathbf{W}^{(i)}{}_k$ and $\mathbf{b}^{(i)}{}_k$ and α_k are the weight matrix bias vector and posterior associated with the k^{th} factor class, respectively, and K is the number of factor classes considered. By definition, we have $\sum_k \alpha_k = 1$. $\{\alpha_k\}_{k=1\ldots,K}$ characterize the acoustic context of a given utterance, which depends on the task, e.g. the gender, speaker or acoustic environment (noise or reverberation). For example, α_k can be obtained as the posteriors derived from speaker or environment clustering. In this paper we use context posteriors that are constant across an utterance, but the same formulation could be used for a context that varies within an utterance.

We can express the output of the i^{th} factorized layer by processing its input by using K parallel sub-layers, followed by the weighted averaging of the outputs of each sub-layer before applying the activation function, i.e.,

$$\mathbf{z}^{(i)} = \sum_{k=1}^{K} \alpha_k \underbrace{(\mathbf{W}^{(i)}{}_k \mathbf{x}^{(i-1)} + \mathbf{b}^{(i)}{}_k)}_{\triangleq \mathbf{z}_k^{(i)}}. \qquad (3)$$

Although exactly identical to using Eq. (2), this latter interpretation makes implementation easier if the α_k values are allowed to vary on a frame basis or when training using mini-batches that are randomized over acoustic conditions and consequently have a different α_k per input feature.

Figure 1-(b) is a schematic diagram of a context adaptive DNN with the i^{th} layer replaced by a factorized layer. Note that in principle we could factorize any layer or several layers of the network, although in the following we will present results obtained when factorizing only a single layer.



(a) Conventional DNN

(b) Context adaptive DNN

Fig. 1. Schematic diagram of (a) a conventional DNN and (b) the proposed context adaptive DNN with the i^{th} layer replaced by a factorized layer. Note that the dotted boxes are included to emphasize intermediate steps in the computation of the output of a hidden layer (i.e. linear transformation and activation function) and are not actual hidden layers.

2.2. Training procedure

Let us now briefly describe how to train context adaptive DNNs. The parameters of the factorized layers, $\Theta \triangleq \{\mathbf{W}^{(i)}{}_k, \mathbf{b}^{(i)}{}_k\}$, can be obtained with the back-propagation algorithm. The implementation of the training algorithm requires simple modifications to an existing DNN training implementation. In particular, the gradients of the factorized layer parameters are given by,

$$\begin{cases} \dfrac{\partial J(\Theta)}{\partial \mathbf{W}^{(i)}{}_k} &= \alpha_k \boldsymbol{\delta}^{(i)} (\mathbf{x}^{(i-1)})^T, \\ \dfrac{\partial J(\Theta)}{\partial \mathbf{b}^{(i)}{}_k} &= \alpha_k \boldsymbol{\delta}^{(i)}, \end{cases} \qquad (4)$$

where $J(\Theta)$ is the objective function (typically the cross entropy), and δ is the back-propagated error that is expressed as,

$$\boldsymbol{\delta}^{(i)} = ((\mathbf{W}^{(i+1)})^T \boldsymbol{\delta}^{(i+1)}) \odot \sigma(\mathbf{z}^{(i)})', \tag{5}$$

where $\odot$ is the Hadamard product and $\sigma'(\mathbf{z}^{(i)})$ is the derivative of the activation function w.r.t. $\mathbf{z}^{(i)}$. Equation (4) is similar to the expression of the gradient for a conventional neural network [24] except for the introduction of the weighting term α_k. Moreover, Eq. (5) is identical to the expression for a conventional DNN but $\mathbf{z}^{(i)}$ should be calculated with Eq. (3) and $\mathbf{W}^{(i+1)}$ should be calculated with Eq. (2) if layer $i+1$ is factorized.

There are several training strategies that could be used to train the proposed context adaptive DNN. One can start from a network pre-trained for conventional DNN training (e.g. obtained using restricted Boltzmann machine (RBM) pre-training), and create an initial factorized layer by duplicating the layer corresponding to the factorized layer by the number of factors. However, we found in our experiments that better results could be obtained by using a *warm-start* approach, where the initial value for the adaptive DNN consists of a conventional factor independent DNN trained (i.e. fine-tuned) with all the training data. The factorized layer was obtained by duplicating the original layer by the number of factors and the remaining layers are kept unchanged. The network is then retrained using a small learning rate. Note that all the layers of the network are retrained but only the factorized layer becomes context dependent in a similar way to the approach described in [13].

3. RELATIONSHIP TO PREVIOUS WORK

The proposed context adaptive DNN shares similarities with other approaches to DNN adaptation. Factorized DNNs were investigated in [22], where the authors proposed factorizing the last layer of a DNN by introducing a weight tensor to combine the DNN output and the factors. Our implementation is especially similar to the disjoint factorized model proposed in [22]. However, we apply the factorization to the weights and biases of the hidden layers instead of the softmax layer, which may be more general as it can be extended to any layer of the network. Moreover, another difference is that in [22] the same features were used for recognition and for estimating the speaker and environment factors. In our case, we use factors estimated in a separate process that enables us to use different features that may be better suited to representing the acoustic context.

The proposed context adaptive DNN is also related to the speaker adaptive training approach proposed in [13], where a given DNN layer was made speaker dependent during training, while maintaining the other layers speaker independent. The speaker dependent layer was then retrained in a supervised manner using adaptation data of the corresponding speaker. Our approach uses posteriors to train the DNN and therefore implements a soft version of [13]. Moreover, instead of retraining the speaker dependent layer to adapt the DNN to each test speaker, we use posteriors to compute the adapted DNN directly. As the posteriors we use can be calculated blindly on an utterance basis, we can achieve fast unsupervised adaptation. We could potentially combine the proposed approach with [13] by using a few adaptation utterances to retrain the context adaptive DNN to the test conditions.

Another promising approach for fast unsupervised adaptive DNN consists of using rich features obtained by concatenating the original input features with additional features representing acoustic conditions such as speaker [15], noise [16] or both [17]. Such approaches are simple to implement and have been shown to be effective for many tasks [15, 16]. However, they make it necessary to train a network from scratch for a set of rich features. The proposed context adaptive DNN employs a different approach for integrating acoustic condition information. Moreover, a potential advantage of the proposed approach is that we can use an already trained factor independent DNN as the initial model, which may speed up the training.

Finally, [12, 18] proposed including additional features in the input of intermediate hidden layers [18] or in the last layer [12]. However, these approaches require two passes for adaptation (one pass to generate labels and one pass to recognize them after adapting the DNNs), while the proposed context adaptive DNN operates in a single pass.

4. EXPERIMENTS

In this section we describe preliminary experiments based on the TIMIT continuous phoneme recognition task [25]. In this preliminary experiment we used two factor classes, which therefore corresponds to gender adaptation. Note that TIMIT is probably not the best corpus with which to demonstrate the potential of the proposed approach as it has already been shown that conventional DNN based acoustic models could perform speaker normalization on this task [26]. However we used this task as it enabled us to perform simple experiments to test our concept.

4.1. Settings

4.1.1. Recognizer configuration

Our baseline system consists of a DNN-HMM recognizer, trained using all the training data. We refer to this system as a gender independent DNN (GI-DNN). The DNN consists of 6 hidden layers with 2048 hidden units per layer and 144 output units. The input features consist of MFCC features with delta and acceleration (39 dimensions in total). We used the 11 concatenated frames as input to the DNN (i.e. 429 input units). As is common practice for DNNs, the input features were normalized using mean and variance normalization parameters calculated using the training data set. The DNNs were trained with conventional layer-wise pre-training using RBMs followed by fine tuning using SGD [23, 27]. For the fine tuning, we used an initial learning rate of 0.1, a momentum of 0.9 and a batch size of 128. Moreover, the learning rate was gradually decreased when the frame accuracy would not improve for a validation set (i.e. here the development set).

In addition to the above GI-DNN, we also trained gender dependent DNNs (GD-DNNs). The GD-DNNs were obtained by retraining the GI-DNN (warm start) using only male and female data respectively. Note that we observed that the performance of this retraining strategy was superior to that of training GD-DNNs created from scratch. We used the same parameters for retraining as for fine tuning of GI-DNN except for the learning rate that we set at a smaller value of 0.001. We also employed the same retraining strategy to train the proposed context adaptive DNNs.

We used monophone HMMs for all the experiments. For decoding, we used a phoneme bigram language model and fixed the language model weight at 1 in all the experiments. The results are expressed in terms of the phone error rate (PER) for the development (dev) and evaluation (eval) sets.

Table 1. Phone error rate for TIMIT experiment. The results are shown for the baseline gender independent DNN (GI-DNN), gender dependent DNN (GD-DNN), DNN with rich input features (RF-DNN) and the proposed context adaptive DNN (CA-DNN). The best results are highlighted with bold font.

	posteriors	dev	eval
GI-DNN	-	22.12 %	22.99 %
GD-DNN	-	**21.69 %**	22.61 %
RF-DNN	w/ LDA	21.76 %	22.98 %
RF-DNN	w/o LDA	21.88 %	22.84 %
CA-DNN	w/ LDA	21.75 %	22.66 %
CA-DNN	w/o LDA	21.71 %	**22.50 %**

4.1.2. Posterior calculation

In this experiment, the posteriors α_k were obtained from the clustering of i-vectors using GMMs [28]. Here, we deal with two factor classes and therefore the number of Gaussian components for i-vector clustering was set at 2. The i-vectors consisted of 400 components. We used two types of posteriors, one obtained by applying dimensionality reduction using linear discriminant analysis (LDA) before clustering the i-vectors (w/ LDA) and one obtained by clustering i-vectors directly (w/o LDA). We used LDA to obtain more speaker-discriminant vectors. In that case, we reduced the dimensionality of the i-vectors to 4 using an LDA projection matrix estimated by employing the speaker IDs of the training data to define the classes used by LDA. Note that the posteriors obtained without LDA are somewhat smoother than those obtained with LDA. The latter posteriors tend to take binary values. These two types of posteriors were used to confirm the effect of soft training.

4.2. Results

Table 1 shows the PER the baseline GI-DNN and GD-DNN systems, systems using rich input features (RF-DNN) obtained by concatenating the input features with posteriors and systems using the proposed context adaptive DNN (CA-DNN).

The GI-DNNs performance is comparable to that obtained by others on the same task when using the same input features and network topology [23]. The small performance differences may be due to minor differences in the training strategy. We observe a small but consistent improvement with the GD-DNNs. Note that when using GD-DNNs we assume prior knowledge about gender during decoding. The other systems in Table 1 do not use such prior information.

Table 1 also shows the results for rich features (RF-DNN) acquired by concatenating the input features with posterior probabilities obtained with i-vectors processed with and without LDA. This is similar in principle to adding i-vectors to the input of DNNs, but the dimension of the posteriors is smaller than that usually used for i-vectors. In this experiment, we obtained better performance using the posteriors than with i-vectors directly. We observed a small improvement over GI-DNN when using RF-DNN, however the performance did not match that obtained with GD-DNNs.

The last part of Table 1 shows the results of the proposed context adaptive DNN (CA-DNN) for the two different types of posteriors (i.e. 'w/ LDA' and 'w/o LDA'). The results were obtained when using a single factorized layer. Table 1 shows the results for the factorized layer that gave the best performance on the development set, i.e. the second layer for posteriors with LDA and the third layer for posteriors without LDA. Note that similar results could be obtained when factorizing another layer or several layers of the network. However, the performance tended to degrade slightly when the last layer was factorized. This suggests that some extra layers may be needed on top of the factorized layer to compensate for perturbations that may occur when the posteriors observed during testing and training differ.

We observed that CA-DNNs can achieve performance comparable to that of GD-DNNs without using prior knowledge about gender. Both types of posteriors achieve similar performance levels, but the use of smoother posteriors obtained without LDA provides slightly better performance than when using LDA or GD-DNNs. This confirms that the soft training strategy is effective. It is noticeable that we could achieve some performance improvement on TIMIT although it is known that DNNs can perform speaker normalization on this task [26]. Note that we confirmed that the performance improvement is not due to the increased number of CA-DNN parameters. Indeed, a GI-DNN with 7 layers (which has the same number of parameters as the CA-DNNs), achieved poorer PERs of 22.17 % and 23.03 % on the development and evaluation sets, respectively. In addition, we also found that increasing the number of units of a given hidden layer did not improve performance.

We also tested the proposed approach using 4 and 8 factor classes to extend the experiment to speaker as well as gender classes. The performance with 4 and 8 classes cases was very similar to that using 2 classes with a slight degradation with 8 classes. The number of model parameters increases with the number of factor classes. Consequently, when increasing the number of classes we may need more training data to accurately train the parameters of the factorized layer. In addition, we should investigate approaches to for reducing the number of parameters of the factorized layers when increasing the number of classes using e.g. bottleneck layers [11, 29, 30].

5. CONCLUSION

In this paper, we introduced a novel approach for DNN adaptation that we called context adaptive DNN. The proposed DNN adapts its model parameters using a set of posteriors that describe the acoustic context. This enables the rapid unsupervised adaptation of DNNs even when the number of parameters is large. The proposed context adaptive DNN was tested for gender adaptation on the TIMIT continuous phoneme recognition task. We observed small but consistent improvements using the proposed method. In particular, we achieved similar performance to that of gender dependent DNNs. These are preliminary results for a simple task. We expect to observe larger gains for tasks with more training data or when dealing with other acoustic conditions such as noise or reverberation that may be more difficult to represent by a conventional DNN [19, 31]. These investigations will constitute part of our future work. We will also investigate approaches for reducing the number of parameters of the context adaptive DNN to enable us to perform experiments with a larger number of factor classes.

6. REFERENCES

[1] G. Hinton, L. Deng, D. Yu, G.E. Dahl, A. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T.N. Sainath, and B. Kingsbury, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal Process. Mag.*, vol. 29, no. 6, pp. 82–97, 2012.

[2] C.J. Leggetter and P.C. Woodland, "Maximum likelihood linear regression for speaker adaptation of continuous density hidden Markov models," *Computer Speech & Language*, vol. 9, no. 2, pp. 171–185, 1995.

[3] J. Neto, L. Almeida, M. Hochberg, C. Martins, L. Nunes, S. Renals, and T. Robinson, "Speaker-adaptation for hybrid HMM-ANN continuous speech recognition system," in *Proc. of EUROSPEECH'95*, 1995, pp. 2171–2174.

[4] F. Seide, G. Li, X. Chen, and D. Yu, "Feature engineering in context-dependent deep neural networks for conversational speech transcription," in *Proc. of ASRU'11*, 2011, pp. 24–29.

[5] K. Yao, D. Yu, F. Seide, H. Su, L. Deng, and Y. Gong, "Adaptation of context-dependent deep neural networks for automatic speech recognition," in *Proc. of SLT'12*, 2012, pp. 366–369.

[6] T. Yoshioka, A. Ragni, and M. J. F. Gales, "Investigation of unsupervised adaptation of DNN acoustic models with filter bank input," in *Proc. of ICASSP'14*, 2014, pp. 6344–6348.

[7] O. Abdel-Hamid and H. Jiang, "Rapid and effective speaker adaptation of convolutional neural network based models for speech recognition," in *Proc. of INTERSPEECH'13*, 2013, pp. 1248–1252.

[8] Y. Miao, H. Zhang, and F. Metze, "Towards speaker adaptive training of deep neural network acoustic models," in *Proc of INTERSPEECH'14*, 2014, pp. 2189–2193.

[9] D. Yu, K. Yao, H. Su, G. Li, and F. Seide, "KL-divergence regularized deep neural network adaptation for improved large vocabulary speech recognition," in *Proc. of ICASSP'13*, 2013, pp. 7893–7897.

[10] H. Liao, "Speaker adaptation of context dependent deep neural networks," in *Proc. of ICASSP'13*, 2013, pp. 7947–7951.

[11] J. Xue, J. Li, D. Yu, M. Seltzer, and Y. Gong, "Singular value decomposition based low-footprint speaker adaptation and personalization for deep neural network," in *Proc of ICASSP'14*, 2014, pp. 6359–6363.

[12] J. Li, J.-T. Huang, and Y. Gong, "Factorized adaptation for deep neural network," in *Proc of ICASSP'14*, 2014, pp. 5537–5541.

[13] T. Ochiai, S. Matsuda, X. Lu, C. Hori, and S. Katagiri, "Speaker adaptive training using deep neural networks," in *Proc of ICASSP'14*, 2014, pp. 6349–6353.

[14] R. Doddipatla, M. Hasan, and T. Hain, "Speaker dependent bottleneck layer training for speaker adaptation in automatic speech recognition," in *Proc of INTERSPEECH'14*, 2014, pp. 2199–2203.

[15] G. Saon, H. Soltau, D. Nahamoo, and M. Picheny, "Speaker adaptation of neural network acoustic models using i-vectors," in *Proc. ASRU'13*, 2013, pp. 55–59.

[16] M.L. Seltzer, D. Yu, and Y. Wang, "An investigation of deep neural networks for noise robust speech recognition," in *Proc. of ICASSP'13*, 2013, pp. 7398–7402.

[17] P. Karanasou, Y. Wang, M. J. F. Gales, and P. C. Woddland, "Adaptation of deep neural network acoustic models using factorized i-vectors," in *Proc of INTERSPEECH'14*, 2014, pp. 2180–2184.

[18] S. Xue, O. Abdel-Hamid, H. Jiang, and L. Dai, "Direct adaptation of hybrid DNN/HMM model for fast speaker adaptation in LVCSR based on speaker code," in *Proc of ICASSP'14*, 2014, pp. 6339–6343.

[19] M. Delcroix, T. Yoshioka, A. Ogawa, Y. Kubo, M. Fujimoto, N. Ito, K. Kinoshita, M. Espi, T. Hori, T. Nakatani, and A. Nakamura, "Linear prediction-based dereverberation with advanced speech enhancement and recognition technologies for the REVERB challenge," in *Proc. of REVERB'14*, 2014.

[20] S. Haykin, *Neural Networks: A Comprehensive Foundation*, Prentice Hall PTR, Upper Saddle River, NJ, USA, 2nd edition, 1999.

[21] V. Tresp, "Committee machines," in *Handbook of Neural Network Signal Processing*, Y.H. Hu and J.-N. Hwang, Eds., chapter 5. CRC Press, Inc., Boca Raton, FL, USA, 2001.

[22] D. Yu, X. Chen, and L. Deng, "Factorized deep neural networks for adaptive speech recognition," in *Proc. of IWSML'12*, 2012.

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

[24] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification (2Nd Edition)*, Wiley-Interscience, 2000.

[25] J. Garofolo, L. Lamel, W. Fisher, J. Fiscus, D. Pallett, N. Dahlgren, and V. Zue, "TIMIT acoustic-phonetic continuous speech corpus LDC93S1," in *Proc. of SNL'92*. 1993, Linguistic Data Consortium.

[26] A. Mohamed, G. Hinton, and G. Penn, "Understanding how deep belief networks perform acoustic modelling," in *Proc. of ICASSP'12,*, 2012, pp. 4273–4276.

[27] G. Hinton, "A practical guide to training restricted Boltzmann machines," Tech. Rep., 2010.

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

[29] T. N. Sainath, B. Kingsbury, V. Sindhwani, E. Arisoy, and B. Ramabhadran, "Low-rank matrix factorization for deep neural network training with high-dimensional output targets," in *Proc. of ICASSP'13*, 2013, pp. 6655–6659.

[30] S. Wiesler, A. Richard, R. Schluter, and H. Ney, "Mean-normalized stochastic gradient for large-scale deep learning," in *Proc. of ICASSP'14*, 2014, pp. 180–184.

[31] M. Delcroix, Y. Kubo, T. Nakatani, and A. Nakamura, "Is speech enhancement pre-processing still relevant when using deep neural networks for acoustic modeling?," in *Proc. of INTERSPEECH'13*, 2013, pp. 2992–2996.