# FACTORIZED ADAPTATION FOR DEEP NEURAL NETWORK

*Jinyu Li, Jui-Ting Huang, and Yifan Gong*

Microsoft Corporation, Redmond, 98052, WA, USA

## ABSTRACT

In this paper, we propose a novel method to adapt context-dependent deep neural network hidden Markov model (CD-DNN-HMM) with only limited number of parameters by taking into account the underlying factors that contribute to the distorted speech signal. We derive this factorized adaptation method from the perspectives of joint factor analysis and vector Taylor series expansion, respectively. Evaluated on Aurora 4, the proposed method can get 19.0% and 10.6% relative word error rate reduction on test set B and D with only 20 adaptation utterances, and can have decent improvement with as few as two adaptation utterances. We also show that the proposed method is better than feature discriminative linear regression (fDLR), an existing DNN adaptation method. Its small number of parameters and short training time offer an attractive solution to low-footprint speech applications.

***Index Terms***— deep neural network, factorized adaptation, joint factor analysis, vector Taylor series

## 1. INTRODUCTION

Recently, a new acoustic model, referred to as the context-dependent deep neural network hidden Markov model (CD-DNN-HMM), has been developed. It has been shown, by many groups [1][2][3][4][5][6], to outperform the conventional GMM-HMMs in many automatic speech recognition (ASR) tasks. Although popular now in general ASR tasks, there are only very few works to investigate the effectiveness of CD-DNN-HMM on noise-robust ASR tasks [7][8][9][10]. With the excellent modeling power of DNN, in [7] it is shown that the DNN-based acoustic models can easily match state-of-the-art performance of GMM systems without any explicit noise compensation. This is because its layer-by-layer setup provides a feature extraction strategy that automatically derives powerful noise-resistant features from primitive raw data for senone classification. In [8][9][10] robust front-end is investigated to see whether it is still helpful to CD-DNN-HMM.

In contrast to the above works which study the applicability of noise-robust front-end technologies to CD-DNN-HMM, we investigate the effectiveness of DNN adaptation methods in this paper. Specifically, we are interested in small foot-print adaption, in which a relatively small number of parameters are used to adjust the existing DNN to fit new environments. This is closely related to our real-world application requirements: we have data from large amount of environments, and we want to build specific models for these environments. It is not realistic to build and store a specific DNN with huge number of parameters for each environment. Instead, low-footprint environment-specific models are acceptable.

There are several types of methods to adapt neural networks. The first type of method, linear input network (LIN) [11][12], applies affine transforms to the input of a neural network to map the speaker-dependent input feature to the speaker-independent feature. Similarly, the linear output network (LON) adds a linear layer at the output layer of the neural network, right before the softmax functions are applied. However, LON is reported to give worse results than the baseline neural network [12]. In the context of DNN, feature discriminative linear regression (fDLR) [13], an example of LIN, is proposed to adapt a DNN with decent gains. The second type of method, linear hidden layer (LHN) [14], adds a linear transform network before the output layer. The rationale behind LHN is that the added linear layer generates discriminative features of the input pattern suitable for the classification performed at the output of the neural network. This LHN method is used in [15] as the output-feature discriminative linear regression (oDLR) method to adapt a DNN by transforming the outputs of the final hidden layer of a DNN, but with worse performance than fDLR. The third type of method [16][17] changes the shape of the hidden activation function instead of the network weights to better fit the speaker-specific features. The fourth type of method adds some regularization terms to the objective function to prevent the network weights from moving too far away from the baseline model. In [18], L2 regularization on network weights is used, while Kullback–Leibler divergence (KLD) on output probabilities is used in [19][20] as the regularization term for DNN adaptation. Finally, in [21] a method called speaker code is proposed to adapt a DNN by putting all the speakers together to train individual speaker codes and several adaptation DNN layers which transform speaker-dependent features into speaker-independent ones before feeding them into the original DNN.

All the above-mentioned methods could be applied to noise-robust ASR tasks. However, these methods except the speaker code method only adapt or add the network weights without differentiating the underlying factors that cause the mismatch between training and testing. In the literature of noise-robust ASR [22], there are acoustic factorization methods [23][24][25] which separate the clean speech feature/model from the multiple speaker and environment factors irrelevant to the phonetic classification. In this paper, we propose a novel DNN adaptation method by taking into account the underlying factors that contribute to the distorted speech signal.

The rest of this paper is organized as follows. We will first briefly introduce CD-DNN-HMM in Section 2. Then, in Section 3, we propose the factorized adaptation method with small amount of parameters and link this method with well-established technologies such as joint factor analysis (JFA) [26] and vector Taylor series (VTS) expansion [27][28]. In Section 4, we show that the proposed method can get up to 19.0% and 10.6% relative word error rate reduction on test set B and D of Aurora 4 [29] with only 20 adaptation utterances, and can beat fDLR in most cases when using different number of adaptation utterances. We discuss the relation to prior work in Section 5, and then conclude the study and propose the future research direction in Section 6.

## 2. CD-DNN-HMM

A deep neural network (DNN) can be considered as a conventional multi-layer perceptron (MLP) with many hidden layers (thus deep) as illustrated in the left side of Figure 1, in which the input and output of the DNN are denoted as $x$ and $o$, respectively. The three major components contributing to the excellent performance of CD-DNN-HMM are: modeling senones directly even though there might be thousands or even tens of thousands of senones; using DNNs instead of shallow MLPs; and using a long context window of frames as the input.

Denote the input vector at layer $l$ as $v^l$ (with $v^0 = x$), the weight matrix as $W^l$, and bias vector as $a^l$. Then for a DNN with $L$ hidden layers, the output of the $l$-th hidden layer is

$$\mathrm{v}^{\mathrm{l}+1} = \sigma\left(\mathrm{z}(\mathrm{v}^{\mathrm{l}})\right), \qquad 0 \le \mathrm{l} < \mathrm{L} \tag{1}$$

where $z(\mathrm{v}^{\mathrm{l}}) = W^l \mathrm{v}^{\mathrm{l}} + a^l$ and $\sigma(\mathrm{x}) = 1/(1+\mathrm{e}^{\mathrm{x}})$ is the sigmoid function applied element-wise. The posterior probability is

$$p_{o|x}(o = s|x) = softmax(z(v^L)), \tag{2}$$

where $s$ is the tied triphone states (also known as senones).

We compute the HMM's state emission probability density function $p_{x|o}(x|o = s)$ by converting the state posterior probability $p_{o|x}(o = s|x)$ to

$$p_{x|o}(x|o = s) = \frac{p_{o|x}(o = s|x)}{p_o(o = s)} \cdot p(x), \tag{3}$$

where $p_o(o = s)$ is the prior probability of state $s$, and $p(x)$ is independent of state and can be dropped during evaluation.

## 3. ACOUSTIC FACTORIZATION FOR DNN

Denote $r = z(v^L)$ as the output vector right before the softmax activation in Eq-(2). Now we consider the case that the input feature, *x*, has been distorted by environment factors to become *y*. Merging the layer-by-layer $z$ and $\sigma$ functions, we can denote $r = R(y)$ , where $R(\cdot)$ represents the overall nonlinear function in a DNN. In this study, to adapt an existing DNN to a new environment, we propose to compensate the vector $r$ by removing those unwanted parts in the network outputs caused by acoustic factors, as shown in Figure 1. Specifically, the modified vector $r'$is obtained by

$$r' = R(y) + \sum_{n=1}^{N} Q_n f_n, \tag{4}$$

where $f_n$ is the underlying *n*-th acoustic factor and $Q_n$ is the corresponding loading matrix. Then $r'$ is used to calculate the posterior probability as

$$p_{o|x}(o = s|x) = softmax(r'). \tag{5}$$

When adapting the existing DNN to a new environment, we extract the factors $[f_1, \dots\dots f_N]$ from adaptation utterances, and then train the loading matrixes $[Q_1, \dots\dots Q_N]$ using standard back-propagation. Then this adapted DNN can be used to decode the utterance from the same environment.

In the following, we examine what acoustic factors should be used for adaption and link our proposed method with well-established technologies from different perspectives.

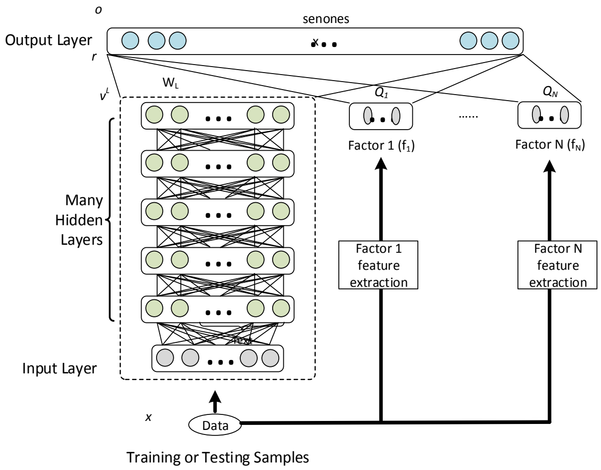


Figure 1. The flow chart of factorized adaptation for a DNN.

### 3.1. From the view of joint factor analysis

Joint factor analysis (JFA) [26] is a very successful technology in speaker recognition by denoting speaker-dependent mean supervector *M* as

$$M = m + Aa + Bb + Cc, \tag{6}$$

where *m* is the speaker- and session-independent mean supervector, *A* and *C* define a speaker subspace (eigenvoice matrix and diagonal residual, respectively), and B defines a session subspace (eigenchannel matrix). The speaker, speaker-specific residual and session factors are *a, c,* and *b*, respectively.

Similarly, when considering nuisance factors that affect DNN's prediction, we can use noise, channel, and speaker as factors in Eq-(4), and relate the factor-independent $r'$ and dependent vectors *R(y)* in Eq-(4) to the speaker/session- independent and dependent mean supervectors *m* and *M*, respectively. While sharing the concept of decomposing speech into several factors with JFA, our method takes a fundamentally different approach for the purpose of acoustic model adaptation. This study focuses on estimating the "compensation" matrix using the discriminative training criterion given the pre-determined acoustic factors, whereas JFA jointly estimates the JFA matrices and the speaker/session factors. In our experiments, we refer to the factorized adaptation method considering any environment-related factors (noise, channel, or speaker) as JFA-style adaption. We next introduce a second method that considers additional inputs which are derived from the view of vector Taylor series expansion.

### 3.2. From the view of vector Taylor series expansion

Vector Taylor series (VTS) expansion is a very successful noise robust method [27][28] which uses a parsimonious nonlinear physical model to describe the environmental distortion and uses the VTS approximation technique to find closed-form HMM adaptation and noise/channel parameter estimation formulas. We can relate our proposed adaption method to VTS technologies by the following derivation.

Suppose clean speech $x$, distorted speech $y$, and noise $n$ are in the log filter-bank domain, which has been proved to be better than MFCC as the input of a DNN [30][31]. They have the following relationship,

$$x = y + \log(1 - \exp(n - y)) \tag{7}$$

and can be expanded with first-order VTS at $(y_0, n_0)$ as

$$x \approx y + \log(1 - \exp(n_0 - y_0)) + A(y - y_0) + B(n - n_0), \tag{8}$$

where

$$A = \frac{\partial \log(1-\exp(n-y))}{\partial y}\Big|_{y_0,n_0}$$
$$B = \frac{\partial \log(1-\exp(n-y))}{\partial n}\Big|_{y_0,n_0}.$$

Then, $R(x)$ can also be expanded with first-order VTS as

$$\begin{aligned} R(x) &= R(y + \log(1 - \exp(n - y))) \\ &\approx R(y) + \frac{\partial R}{\partial y}\log(1 - \exp(n - y)) \\ &\approx R(y) + \frac{\partial R}{\partial y}(\log(1 - \exp(n_0 - y_0)) + A(y - y_0) + \\ &\quad B(n - n_0)) \\ &\approx R(y) + \frac{\partial R}{\partial y}(Ay + Bn + const.) \end{aligned} \tag{9}$$

If we make a rather imprecise assumption that $\partial R/\partial y$ is constant, Eq-(9) can be simplified as

$$R(x) \approx R(y) + Cy + Dn + const. \tag{10}$$

Eq-(10) is closely related to Eq-(4) and shows that in addition to using the noise $n$ as a factor, we should also use the distorted input feature $y$ as a factor to adjust $R(y)$. Similar VTS-type inference can also be made by introducing additional factors such as channel, and can be done in other domains such as MFCC. In our experiments, we refer to the factorized adaptation method considering not only environment factors but also distorted input features as VTS-style adaptation.

## 4. EXPERIMENTS

We evaluate the effectiveness of our proposed methods with Aurora 4 [29], a noise-robust medium-vocabulary task based on Wall Street Journal corpus (WSJ0). The 16kHz clean-condition training set consists of 7138 utterances recorded with the Sennheiser microphone, corresponding to 14 hours of speech data.

There are totally 14 evaluation sets. Two clean evaluation sets (A and C) are recorded with the Sennheiser microphone and the secondary microphone, respectively. The remaining 12 subsets are divided into two groups (B and D), recorded with two types of microphone respectively. Inside each group, 6 types of noise (car, babble, restaurant, street, airport, and train) are added with randomly chosen SNRs between 5 and 15 dB for each of the microphone types. There is also a multi-condition training set of Aurora 4 with 14 subsets, each with the same types of noise and microphone as the sub test sets, but with added SNR between 10 and 20 dB.

The baseline GMM-HMM system has 1206 senones, each with 16 Gaussians trained using maximum likelihood estimation criterion on the clean training set. The GMM-HMM system is used to align the training data to get the forced alignment for training the DNN-HMM system. Decoding is performed with the task-standard WSJ0 bigram language model.

The clean-trained DNN is trained with 24-dimensional log Mel filter-bank features and their first- and second-order derivative features. The input layer is formed from a context window of 11 frames, which means the dimension of input layer is 792. The DNN has 5 hidden layers with 2048 hidden units in each layer and the final soft-max output layer has 1206 units, corresponding to the senones of the HMM system. The network is initialized with pre-training and then fine-tuned using 25 iterations of back propagation. This DNN obtains 4.4% word error rate (WER) on test set A, the matched test set. On the mismatched test sets B, C, and D, it gets 24.3%, 20.7%, and 41.3% WER, respectively.

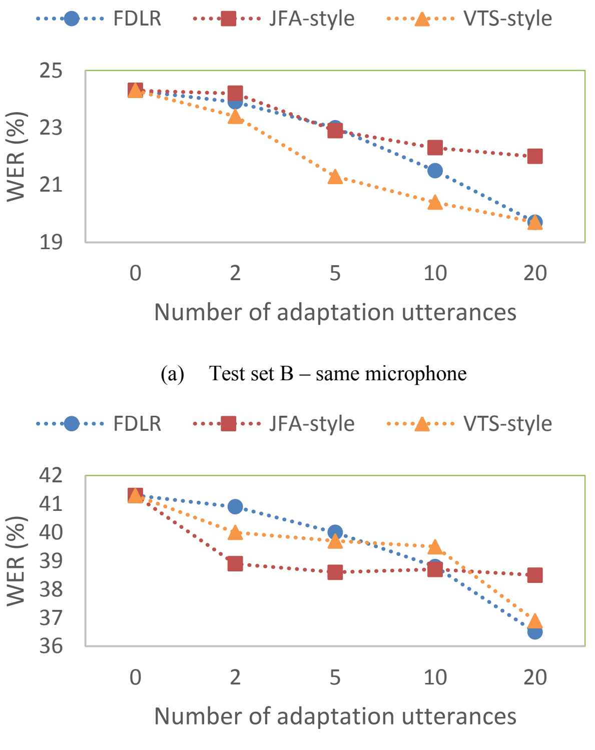
In this study, we use the noise-distorted test sets B and D to evaluate the method to adapt the clean-trained DNN to noise environments. The only difference between B and D is the type of microphone used; B uses the same microphone as the clean training set, while D uses a secondary one. We randomly sample 2, 5, 10, and 20 utterances from each noise-distorted sub training set to adapt the clean DNN, and then evaluate on the corresponding test set.

For JFA-style adaptation, we extract the 72-dimension noise factor by averaging the first and last 20 frames of each utterance, which means we use the same noise factor for every frame within an utterance. For VTS-style adaptation, in addition to the noise factor used in the JFA-style adaptation, we also use the noise-distorted input $y$ whose dimension is 72. This means for every frame, we have a frame-invariant noise factor $n$ and a frame-variant factor $y$ within an utterance. Since we are interested in limited-footprint adaptation, either fDLR [13] or oDLR [15] is also a possible solution for using limited number of parameters for adaptation. As fDLR has been reported to be better than oDLR [15], we include the results of the fDLR approach as a comparison.

Figure 2 (a) and Figure 2(b) show the averaged WER across all six types of noises versus the number of adaptation utterances for test set B and test set D, respectively. We plot the WERs of the JFA- and VTS-style adaptation methods together with fDLR. Overall, the VTS-style method performs the best. In test set B, it gets 19.0%, 15.8%, 12.3%, and 3.4% relative WER reduction (WERR) with 20, 10, 5, and 2 adaptation utterances, respectively. In test set D, the WERRs are 10.6%, 4.4%, 4.0%, and 3.3%, respectively. For test set B, the VTS-style approach is better than fDLR and the JFA-style approach regardless of the number of adaptation utterances. It shows that when the noise is the main source of variability in speech, the VTS-style adaptation can effectively learn the necessary modification to noisy features for more accurate DNN outputs. For test set D where microphone/channel mismatch occurs, the VTS-style adaptation is slightly worse than fDLR when the number of adaptation utterances is ten or larger. We think that it is possible to further improve the VTS-style adaptation if we could add some estimated channel factors for the adaptation framework, besides the noise and the raw feature factor.

For both cases, the JFA-style method always reaches its plateau much earlier than the other two. It suggests that the “noise” factor extracted in the current experiments hardly captures any further useful information after more adaptation utterances are available. Indeed, currently we just use the noise-mean vector, which does not change much after more utterances are collected.

The factorization approach also possesses an advantage in terms of the training speed. Unlike fDLR where back-propagation computation are conducted through every hidden layer to the input feature level, the factorization approach only requires one layer of computation on relative smaller matrices.



(a) Test set B – same microphone

(b) Test set D – microphone mismatch

Figure 2. Compare JFA- and VTS-style methods with fDLR for DNN adaptation on Aurora 4. The averaged WERs across 6 noise sub test sets after adaptation for Test set B (a) and Test set D (b). “0” for the number of adaptation utterances means the un-adapted clean-trained DNN model.

## 5. RELATION TO PRIOR WORK

As discussed in the introduction section, most existing neural network adaptation technologies only adapt or add the network weights without differentiating the underlying factors that cause the mismatch between training and testing [11]-[20]. The proposed method is fundamentally different from these methods by taking into account the underlying factors that contribute to the distorted speech signal.

The most related work to our proposed method is the speaker code method [21] in which the speaker factor is addressed by training speaker-dependent codes. However, the detail is very different. The speaker code method needs to add several layers to connect the speaker code and the input feature to the bottom hidden layer of the original DNN. These new layers are trained with all the training data and shared by all speakers, while only the speaker code, a vector, is speaker dependent. This somehow restricts the scalability when more adaptation utterances can be used. As shown in [21], the improvement got saturated with 7 adaptation utterances, and there was only very small WER difference between using 2 and 7 adaptation utterances. Our method differs from the speaker code method in these aspects: 1) We use factor-dependent matrices, as opposed to a speaker-dependent vector in the speaker code method; 2) We directly modify the weight matrixes connecting the output layer and the factors for every environment, while the speaker code method needs to train speaker-specific codes and several additional DNN layers connected to the bottom layer of the original network; 3) As shown in Figure 2, our proposed method doesn’t have the fast saturation issue observed with the speaker code method.

As described in Section 3.1 and 3.2, we can view our proposed method from the perspectives of JFA [26] and VTS [27][28]. Different from JFA which works on the mean supervector of GMM and VTS which works on either input features or model parameters of GMM, our proposed method modifies the output vector right before the softmax function in a DNN by adding the impacts from multiple acoustic factors.

It should be noted that although we are using the term of JFA- and VTS-style to describe our methods, it doesn’t mean we strictly follow the formulation of JFA or VTS in this initial study. For example, speaker factor in JFA is not used in JFA-style in this study although we plan to model speaker factor in the future.

## 6. CONCLUSIONS AND FUTURE WORKS

In this paper, we have proposed a novel factorized adaptation method to adapt a DNN with only limited number of parameters by taking into account the underlying factors that contribute to the distorted speech signal. The proposed has two variants of implementation: the JFA-style and the VTS-style adaption methods. In the JFA-style adaptation, only noise is used as a factor while both noise and distorted speech are used as factors in the VTS-style adaptation. Evaluated on Aurora 4 test set B where the speech is only distorted by noise, the VTS-style adaptation gets 19.0%, 15.8%, 12.3%, and 3.4% relative WER reduction (WERR) with 20, 10, 5, and 2 adaptation utterances, respectively. It is consistently better than fDLR and the JFA-style approach in all cases. On test set D where the speech is distorted by noise and channel, the VTS-style approach behaves similarly as fDLR while the JFA-approach is best when only 2 and 5 utterances are used. Despite some imprecise assumption when deriving from the perspective of VTS, overall the VTS-style performs the best among the three methods, suggesting its great potential advantage when more precise modeling is used.

This paper presents our initial study of factor adaption method for a DNN. We are now working on several ways to improve the proposed method. First, as shown in Section 4, the VTS-style adaptation method can get only 10.6% WERR with 20 adaptation utterances on test set D, compared to 19.0% on test set B. In addition to noise distortion as in test set B, there is also channel distortion that needs to be addressed in test set D while our current VTS-style method doesn’t include a channel factor. We expect to get further improvement by including channel factors into the formulation. Second, the VTS-style adaptation in Eq-(10) is approximated by assuming the gradient $\partial R/\partial y$ is constant in Eq-(9). While significant WER improvement has been established under this assumption, we expect to get better performance with precise modeling in Eq-(9). This can be achieved by calculating the gradient using back-propagation during training and testing. Another way to avoid the gradient term in Eq-(9) is to apply VTS-style adaptation at the DNN input layer instead of the output layer, using Eq-(8) with the factor loading matrixes directly. fDLR can be considered as a special case that only the matrix related with the distorted input is used. Last, we will examine the possibility of using i-vector [32] to model total variability instead of using individual factors.

## ACKNOWLEDGEMENT

We would like to thank Dr. Mike Seltzer in Microsoft for providing the baseline setup of Aurora 4.

# REFERENCES

[1] D. Yu, L. Deng, and G. Dahl, “Roles of pretraining and fine-tuning in context-dependent DBN-HMMs for real-world speech recognition,” in *Proc. NIPS Workshop on Deep Learning and Unsupervised Feature Learning*, 2010.

[2] T. N. Sainath, B. Kingsbury, B. Ramabhadran, P. Fousek, P. Novak, and A. Mohamed, “Making deep belief networks effective for large vocabulary continuous speech recognition,” in *Proc. Workshop on Automatic Speech Recognition and Understanding*, pp. 30–35, 2011.

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

[4] N. Jaitly, P. Nguyen, and V. Vanhoucke, “application of pretrained deep neural networks to large vocabulary speech recognition”, in *Proc. Interspeech*, 2012.

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

[6] L. Deng, J. Li, J. -T. Huang et al. “Recent advances in deep learning for speech research at Microsoft,” in *Proc. ICASSP*, 2013.

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

[8] B. Li and K. C. Sim, “Noise adaptive front-end normalization based on vector Taylor series for deep neural networks in robust speech recognition,” in *Proc. ICASSP*, pp. 7408–7412, 2013.

[9] B. Li, Y. Tsao, and K. C. Sim, “An investigation of spectral restoration algorithms for deep neural networks based noise robust speech recognition,” in *Proc. Interspeech*, pp. 3002–3006, 2013.

[10] 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. Interspeech*, pp. 2992–2996, 2013.

[11] 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. Eurospeech*, pp. 2171–2174, 1995.

[12] B. Li and K. C. Sim, “Comparison of discriminative input and output transformations for speaker adaptation in the hybrid NN/HMM systems,” in *Proc. Interspeech*, pp. 526–529, 2010.

[13] F. Seide, G. Li, X. Chen, and D. Yu, “Feature engineering in context-dependent deep neural networks for conversational speech transcription,” in *Proc. ASRU*, 2011.

[14] R. Gemello, F. Mana, S. Scanzio, P. Laface, and R. De Mori, “Linear hidden transformations for adaptation of hybrid ANN/HMM models,” *Speech Communication*, vol. 49, no. 10-11, pp. 827–835, 2007.

[15] 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 *IEEE Spoken Language Technology Workshop (SLT),* pp. 366-369, 2012.

[16] S. M. Siniscalchi, J. Li, and C.-H. Lee, “Hermitian based hidden activation functions for adaptation of hybrid HMM/ANN models,” in *Proc. Interspeech*, 2012.

[17] S. M. Siniscalchi, J. Li, and C.-H. Lee, “Hermitian polynomial for speaker adaptation of connectionist speech recognition systems,” *IEEE Transactions on Audio, Speech, and Language Processing,* vol. 21, no. 10, pp. 2152-2161, 2013.

[18] X. Li and J. Bilmes, “Regularized adaptation of discriminative classifiers,” in *Proc. ICASSP,* 2006.

[19] 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. ICASSP*, 2013.

[20] 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. ICASSP*, 2014.

[21] O. Abdel-Hamid and H. Jiang, “Fast speaker adaptation of hybrid NN/HMM model for speech recognition based on discriminative learning of speaker code,” in *Proc. ICASSP*, 2013.

[22] J. Li, L. Deng, Y. Gong, and R. Haeb-Umbach, “An overview of noise-robust automatic speech recognition, in *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2014.

[23] M. J. F. Gales, “Acoustic factorisation,” in *Proc. ASRU*, pp. 77–80, 2001.

[24] M. L. Seltzer and A. Acero, “Separating speaker and environmental variability using factored transforms,” in *Proc. Interspeech*, pp. 1097–1100, 2011.

[25] M. Rouvier, M. Bouallegue, D. Matrouf, and G. Linares, “Factor analysis based session variability compensation for automatic speech recognition,” in *Proc. ASRU*, pp. 141–145, 2011.

[26] P. Kenny, G. Boulianne, P. Ouellet, and P. Dumouchel, “Joint factor analysis versus eigenchannels in speaker recognition,” *IEEE Transactions on Audio, Speech, and Language Processing,* vol. 15, no. 4, pp. 1435-1447, 2007.

[27] P. J. Moreno, “Speech recognition in noisy environments,” *Ph.D. thesis*, Carnegie Mellon University, 1996.

[28] J. Li, L. Deng, D. Yu, Y. Gong, and A. Acero, “A unified framework of HMM adaptation with joint compensation of additive and convolutive distortions,” *Computer, Speech and Language*, vol. 23, no. 3, pp. 389–405, 2009.

[29] N. Parihar and J. Picone, “Aurora working group: DSR front end LVCSR evaluation AU/384/02,” *Tech. Rep.*, Institute for Signal and Information Processing, Mississippi State Univ., 2002.

[30] A. Mohamed, G. Hinton, and G. Penn, “Understanding how deep belief networks perform acoustic modelling,” in *Proc. ICASSP*, pp. 4273–4276, 2012.

[31] J. Li, D. Yu, J. T. Huang, and Y. Gong, “Improving wideband speech recognition using mixed-bandwidth training data in CD-DNN-HMM,” in *Proc. IEEE SLT*, pp. 131–136, 2012.

[32] 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, 2011.