DIRECT ADAPTATION OF HYBRID DNN/HMM MODEL FOR FAST SPEAKER ADAPTATION IN LVCSR BASED ON SPEAKER CODE

Shaofei Xue[1] *Ossama Abdel-Hamid*[2] *Hui Jiang*[2] *Lirong Dai*[1]*

[1]National Engineering Laboratory of Speech and Language Information Processing,
University of Science and Technology of China, Hefei, P. R. China
[2]Department of Electrical Engineering and Computer Science, York University, Toronto, Canada
Email: {xuesf,lrdai}@mail.ustc.edu.cn, {ossama,hj}@cse.yorku.ca

ABSTRACT

Recently an effective fast speaker adaptation method using discriminative speaker code (SC) has been proposed for the hybrid DNN-HMM models in speech recognition [1]. This adaptation method depends on a joint learning of a large generic adaptation neural network for all speakers as well as multiple small speaker codes using the standard back-propagation algorithm. In this paper, we propose an alternative direct adaptation in model space, where speaker codes are directly connected to the original DNN models through a set of new connection weights, which can be estimated very efficiently from all or part of training data. As a result, the proposed method is more suitable for large scale speech recognition tasks since it eliminates the time-consuming training process to estimate another adaptation neural networks. In this work, we have evaluated the proposed direct SC-based adaptation method in the large scale 320-hr Switchboard task. Experimental results have shown that the proposed SC-based rapid adaptation method is very effective not only for small recognition tasks but also for very large scale tasks. For example, it has shown that the proposed method leads to up to 8% relative reduction in word error rate in Switchboard by using only a very small number of adaptation utterances per speaker (from 10 to a few dozens). Moreover, the extra training time required for adaptation is also significantly reduced from the method in [1].

Index Terms— Deep Neural Network (DNN), Hybrid DNN-HMM, Speaker Code, Fast Speaker Adaptation,

1. INTRODUCTION

Speaker adaptation has been an important research topic in automatic speech recognition (ASR) for decades. Speaker adaptation techniques attempt to optimize ASR performance by transforming speaker-independent models towards one particular speaker or modifying the target speaker features to match the given speaker-independent models based on a relatively small amount of adaptation data. Several successful speaker adaptation techniques have been proposed for the conventional HMM/GMM based speech recognition systems, such as MAP [2, 3], MLLR [4, 5], and CMLLR [6]. As the hybrid deep neural networks (DNN) and HMM models revives in acoustic modelling for large vocabulary continuous speech recognition systems, it now becomes a very interesting problem to perform effective speaker adaptation for DNNs. Recently, a number of speaker adaptation methods have been proposed for neural networks. For example, linear input network (LIN) method in [7] and linear hidden network (LHN) method in [8] both attempt to add additional transforming layers to the initial speaker-independent neural networks. On the other hand, retrained sub-set hidden units (RSHU) method in [9] tries to retrain only weights connected with active hidden nodes. And Hermitian-based MLP (HB-MLP) method in [10] achieves the adaptive capability of hidden activation function through the use of orthonormal Hermite polynomials. More recently, in [11] Kullback-Leibler (KL) divergence is used as regularization for the adaptation criterion and it forces the state distribution estimated from the adapted model to stay close enough to the original model to avoid over-fitting. In [12], it explores how to adapt deep neural networks (DNNs) to new speakers by other retraining and regularization tricks. In spite of these, speaker adaptation remains as a very challenging task for the hybrid DNN-HMM models, especially when only a very small amount of adaptation data is available per speaker, because adaptation of DNNs is very prone to over-fitting due to a large number of model parameters in DNNs. In [1] and [13], a fast speaker adaptation method based on the so-called speaker codes has been proposed for hybrid DNN/HMM models, which is capable of adapting large size DNNs with only a few adaptation utterances. This method relies on a joint training procedure to learn a generic adaptation neural network (NN) from the whole training set as well as many small speaker codes for all different speakers. In this way, the learned adaptation NN is capable of transforming each speaker features into a generic speaker-independent feature space when a small speaker code is given. Adaptation to a new speaker can be simply done by learning a new speaker code without changing any NN weights. This method is appealing because the large adaptation network can be reliably learned from the entire training data set while only a small speaker code is learned from adaptation data for each speaker. Moreover, the speaker code size can be freely adjusted according to the amount of available adaptation data. In [1], the speaker-code based adaptation has been found quite effective for fast speaker adaptation in small scale speech recognition tasks, like TIMIT. However, this method introduces additional adaptation neural networks for feature transformation and it takes a very long time to train prior to adaptation, especially in large vocabulary continuous speech recognition tasks.

In this paper, we extend the idea of speaker-code based adaptation in [1] and propose an alternative direct adaptation method that performs speaker adaptation in model space without using adaptation NNs. Some similar ideas have been previously investigated for shallow neural networks in [14, 15]. The basic idea is to connect speaker codes directly to all hidden and output layers of the original DNNs through a set of new connection weights, which can be

*This work was partially funded by the National Nature Science Foundation of China (Grant No. 61273264) and the National 973 program of China (Grant No. 2012CB326405).

efficiently learned from all or part of training data using additional information of speaker labels. In test stage, a new speaker code is estimated for each new speaker from a small amount of adaptation data and the estimated speaker code is directly fed to the original DNN to form a nonlinear transformation in model space. Since there is no need to estimate the entire generic adaptation neural network as in [1], the additional training time prior to adaptation is reduced significantly. Moreover, experimental results on the Switchboard task have shown that it can achieve up to 8% relative reduction in word error rate with only a few adaptation utterances per speaker (from 10 to several dozens).

2. SPEAKER CODE ADAPTATION

The speaker code based adaptation method proposed in [1] and [13] for DNN-HMM based models is shown as in Fig. 1. This method relies on learning another generic adaptation neural network as well as some speaker specific codes. The adaptation neural network consists of weights matrices $\mathbf{A}^{(l)}$ and $\mathbf{B}^{(l)}$ (for all l), where l stands for the l-th layer of the adaptation neural network. All layers of the adaptation neural network are standard fully connected layers. The top layer of the adaptation neural network represents the transformed features and its size matches the input size. Each layer of the adaptation neural network receives all activation output signals of the lower layer along with a speaker specific input vector $\mathbf{S}^{(c)}$, named as speaker code for speaker c, as follows:

$$O^{(l)} = \sigma(\mathbf{A}^{(l)} O^{(l-1)} + \mathbf{B}^{(l)} \mathbf{S}^{(c)}) \quad (\forall\, l) \tag{1}$$

where $O^{(l)}$ denotes outputs from l-th layers of adaptation neural networks and $\sigma(\cdot)$ stands for sigmoid based nonlinear activation function.

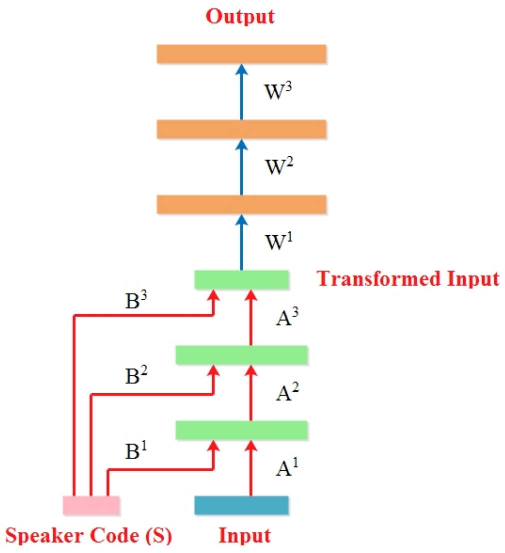

Fig. 1. Speaker adaptation of the hybrid NN-HMM model based on speaker code for feature transformation as in [1].

Assume we need to adapt a well-trained DNN (represented by $\mathbf{W}^{(l)}$), we estimate the adaptation neural network using the back-propagation (BP) algorithm to minimize the cross entropy between the target state labels and the DNN outputs of all training data. The derivatives of cross entropy with respect to all adaptation weights (all $\mathbf{A}^{(l)}$ and $\mathbf{B}^{(l)}$) and speaker code ($\mathbf{S}^{(c)}$) can be easily derived (see [1] for details). In this stage, all adaptation weights (all $\mathbf{A}^{(l)}$ and $\mathbf{B}^{(l)}$) are learned from training data without changing the original DNN (all $\mathbf{W}^{(l)}$). Meanwhile, a number of speaker codes ($\mathbf{S}^{(c)}$) are simultaneously learned with BP for all speakers in the training data based on the available information of speaker labels in training data. In other words, all speaker codes are first randomly initialized and speaker code $\mathbf{S}^{(c)}$ is only updated by training data from speaker c. In this way, we rely on training data as well as the associated speaker labels to learn a generic adaptation neural network that serves as a nonlinear feature transformation to normalize speaker variations in speech signals.

Next, in the adaptation stage, we need to estimate a new speaker code for each new test speaker from a very small amount of adaptation data. During this phase, only the small speaker code is learned from adaptation utterances of the target speaker based on the similar BP algorithm. The whole neural networks (including the initial speaker independent neural network and the adaptation neural network) are kept unchanged. When testing a new utterance, we import the speaker code to adaptation neural network to transform the utterance into a generic space prior to feeding it to the original speaker-independent DNN for final recognition.

3. DIRECT ADAPTATION OF DNNS BASED ON SPEAKER-CODE

In this work, we study the speaker-code based adaptation method for large scale speech recognition tasks and propose an alternative direct adaptation method that conducts speaker adaptation in model space of DNNs. As show in Fig. 2, instead of stacking an adaptation neural network below the initial speaker independent neural network and normalizing speakers features with speakers codes, we propose to feed the speaker codes directly to the hidden layers and the output layer of the initial neural network through a set of new connection weights (all $\mathbf{B}^{(l)}$). In this way, speaker codes are directly used to adapt the speaker-independent DNNs towards new target speakers. A main advantage of this new adaptation scheme is that the computation complexity is dramatically reduced in training because we have no need to learn another set of weight matrices, i.e. all $\mathbf{A}^{(l)}$, from training data. In many cases, $\mathbf{A}^{(l)}$ is significantly bigger than $\mathbf{B}^{(l)}$ since $\mathbf{B}^{(l)}$ is related to speaker codes ($\mathbf{S}^{(c)}$) that has smaller size than hidden layers.

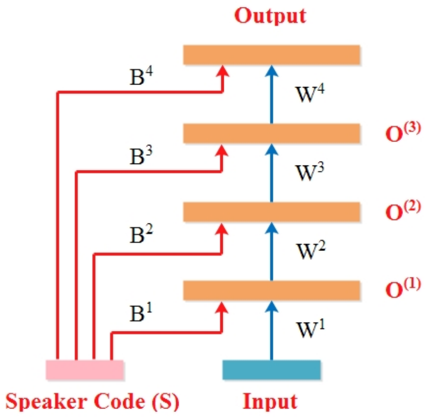

Fig. 2. The proposed direct adaptation of DNNs based on speaker code.

Let us denote $\mathbf{W}^{(l)}$ as the l-th layer weights in the initial neural network that consists of n layers (including input and output layer), and $\mathbf{B}^{(l)}$ as weight matrix to connect speaker code to l-th layer in DNNs, and $\mathbf{S}^{(c)}$ stands for the speaker code specific to c-th speaker. In this case, output signals of l-th layer can be computed as follows:

$$\mathbf{O}^{(l)} = \sigma(\mathbf{W}^{(l)}\mathbf{O}^{(l-1)} + \mathbf{B}^{(l)}\mathbf{S}^{(c)}) \quad (\forall\ l) \tag{2}$$

In the following, we investigate how to estimate connection weights, $\mathbf{B}^{(l)}$, and speaker codes, $\mathbf{S}^{(c)}$, from training data for this new adaptation scheme. For simplicity, we use the cross entropy criterion for adaptation. Assume E denotes the objective function for DNN training or adaptation, such as frame-level cross-entropy (CE) or sequence-level minimum mutual information (MMI) criterion [16]. During the adaptation procedure, we only estimate $\mathbf{B}^{(l)}$ (for all l) and speaker codes $\mathbf{S}^{(c)}$ (for all speakers in the training set) using the stochastic gradient descent algorithm while keeping all $\mathbf{W}^{(l)}$ unchanged. Therefore, the derivative with respect to any element in $\mathbf{B}^{(l)}$, i.e., $B_{kj}^{(l)}$, that connects between the k-th node in the speaker code and the j-th node in l-th layer of initial neural network can be computed as:

$$\frac{\partial E}{\partial B_{kj}^{(l)}} = \frac{\partial E}{\partial O_j^{(l)}}(1 - O_j^{(l)})O_j^{(l)}S_k^{(c)} \tag{3}$$

where $S_k^{(c)}$ that stands for the k-th node in speaker code of c-th speaker.

Similarly, we compute the derivative of E with respect to each element of all speaker codes based on the chain rule. Since the propagation errors from all layers in the neural network contribute to the derivative of $S_k^{(c)}$, we need to summarize all as follows:

$$\frac{\partial E}{\partial S_k^{(c)}} = \frac{1}{n-1}\sum_{l=1}^{n-1}\sum_{j=1}^{J}\frac{\partial E}{\partial O_j^{(l)}}(1 - O_j^{(l)})O_j^{(l)}B_{kj}^{(l)}. \tag{4}$$

In learning, we first randomly initialize all $\mathbf{B}^{(l)}$ and $\mathbf{S}^{(c)}$. Next, we run several epochs of stochastic gradient descents over the training data to update $\mathbf{B}^{(l)}$ and $\mathbf{S}^{(c)}$ based on the gradients computed in eqs.(3) and (4). For speaker codes, $\mathbf{S}^{(c)}$ is only updated by data from c-th speaker. At the end, we have learned all weight matrices $\mathbf{B}^{(l)}$, which are capable of adapting the speaker-independent DNN to any new speaker given a suitable speaker code.

The next step in adaptation is to learn a speaker code for each new speaker. During this phase, only the speaker code is estimated based on eq.(4) for the new speaker from a small number of adaptation utterances while all $\mathbf{B}^{(l)}$ and $\mathbf{W}^{(l)}$ remain unchanged. After the speaker code is learned for each test speaker, the speaker code is imported into the neural network through $\mathbf{B}^{(l)}$ as in eq.(2) to compute posterior probabilities of test utterances for final recognition.

4. EXPERIMENTS

In this section, we evaluate the proposed direct adaptation method for rapid speaker adaptation in two speech recognition tasks: i) the small-scale TIMIT phone recognition task; ii) the well-known large-scale 320-hr Switchboard task.

4.1. TIMIT Phone Recognition

We use the standard 462-speaker training set and remove all SA records (i.e., identical sentences for all speakers in the database) since they may bias the results. A separate development set of 50 speakers is used for tuning all of the meta parameters. Results are reported using the 24-speaker core test set, which has no overlap with the development set. Each speaker in the test set has eight utterances. We use 39 dimensional PLP features (static, first and second derivatives) and 183 target class labels (3 states for each one of the 61 phones) for neural network training. After decoding, the 61 phone classes were mapped to a set of 39 classes as in [17] for scoring purpose. In our experiments, a bi-gram language model in phone level, estimated from the training set, is used in decoding.

For training the weight matrices $\mathbf{B}^{(l)}$, an annealing and early stopping strategies are utilized as in [18] with an initial learning rate of 0.5, the momentum is kept as 0.9. The bunch size is set to 128 and speaker code size is 500. The neural network input layer includes a context window of 11 consecutive frames. Since each test speaker has eight utterances in total. Testing is conducted for each speaker based on a cross validation method. In each run, for each speaker, eight utterances are divided into n_a utterances for adaptation and the remaining $8 - n_a$ utterances for test. The overall recognition performance is the average of all runs. In the learning process of speaker code for each new speaker, the learning rate is set as 0.02 and the bunch size is 32. Two baseline DNNs with various sizes are built: i) 3 hidden layers with 1024 nodes in each hidden layer; ii) 6 hidden layers with 1024 nodes in each hidden layer.

In this section, we evaluate the direct adaptation method for fast speaker adaptation in TIMIT. The results in Table 1 shows that for 3-layer DNN, the direct adaptation using 7 utterances can reduce phone error rate from 23.4% down to 21.5% (about 8.1% relative error reduction). Moreover, for 6-layer DNNs, it reduces PER from 22.9% down to 21.2% (7.4% relative error reduction).

Table 1. *PER (in%) of direct adaptation (using 1, 4 and 7 adaptation utterances) on different DNNs.*

DNN	baseline	1 utt.	4 utt.	7 utt.
3hid*1024node	23.4	22.3	21.8	21.5 (8.1%)
6hid*1024node	**22.9**	22.0	21.4	**21.2** (7.4%)

4.2. Switchboard (SWB)

The SWB training data consists of 309 hour Switchboard-I training set and 20 hour Call Home English training set (1540 speakers in total). In this work, we use the NIST 2000 Hub5e set (containing 1831 utterances from 40 speakers) as the evaluation set. We use 39 dimensional PLP features to train a standard triphone GMM-HMMs model consisting of 8991 tied states based on the maximum likelihood (ML) criterion, which is used to obtain the state level alignment labels for both training and evaluation set.

The baseline DNNs are trained as described in [19, 20, 21, 22] with RBM-based pretraining and BP-based fine-tuning. Three baseline DNNs with various sizes are built: i) 3 hidden layers with 1024 nodes in each hidden layer; ii) 3 hidden layers with 2048 nodes in each hidden layer; iii) 6 hidden layers with 2048 nodes in each hidden layer. We also perform an MMI-based sequence training to refine DNNs as described in [23]. For training connection matrices $\mathbf{B}^{(l)}$ and speakers codes $\mathbf{S}^{(l)}$, we use an initial learning rate of 0.5 and it is halved after three epochs, the momentum is kept as 0.9. The bunch size is set to 1024 and speaker code size is 1000. The training process typically converges after only 4-6 epochs. In the evaluation set (Hub5e00), each test speaker has different number of utterances. The test is conducted for each speaker based on cross validation (CV). In each CV run, a fixed number of utterances (to say 10, 20) is used as adaptation data and the remaining utterances from the same speaker is used to evaluate performance. The process is rotated for many runs until all test utterances are covered. The overall recognition performance is computed as the average of all runs. In the learning process of speaker code for each new speaker, the learning rate is set as 0.02 and the bunch size is 128 and learning is stopped after 5 epochs.

4.2.1. Extra training time of direct adaptation

The speaker-code based adaptation method requires an extra training process to estimate connection matrices from training data. As shown in Table 2, the proposed direct adaptation scheme significantly reduce the extra training time, especially for large DNNs. For the 6-layer DNN in the 3rd row, it needs about 150 hours to train the baseline DNN. The method in [1] requires similar amount of time to train an adaptation DNN. In the proposed direct adaptation scheme, it only needs about 60 hours to estimate $\mathbf{B}^{(l)}$. The training time can be further reduced to only 6 or 12 hours by using only 10% or 20% of randomly selected training data (not the whole training set) to estimate $\mathbf{B}^{(l)}$.

Table 2. *Extra training time (in hr) of direct adaptation in different models structures (using single core of GTX 690).*

DNN models	baseline	training data size		
		10%	20%	100%
3hid*1024node	43	+3.5	+7	+35
3hid*2048node	125	+4.5	+9	+45
6hid*2048node	150	+6	+12	+60

4.2.2. Performance of Fast Speaker Adaptation

In this section, we evaluate the direct adaptation method for fast speaker adaptation in Switchboard. In the first experiment, we use 10 adaptation utterances per speaker to generate speaker codes. The results in Table 3 show that the speaker-code based direct adaptation scheme is very effective to adapt large DNNs by only 10 utterances from each speaker. For example, for 6-layer DNN, the direct adaptation using 10 utterances can reduce word error rate from 16.2% down to 15.2% (about 6.2% relative error reduction). Moreover, the direct adaptation can also be used to adapt sequence-trained DNNs as well, reducing WER from 14.0% to 13.4% (4.3% relative error reduction). The results also show that using 20% of training data to estimate $\mathbf{B}^{(l)}$ still yields comparable performance but it can significantly reduce extra training time as shown in Table 2.

Table 3. *WER (in%) of direct adaptation on different DNNs using 10 adaptation utterances per speaker.*

DNN models	baseline	training data size		
		10%	20%	100%
3hid*1024node	18.9	18.1	18.0	17.8 (5.8%)
3hid*2048node	17.4	16.8	16.6	16.4 (5.7%)
6hid*2048node	**16.2**	15.8	15.6	**15.2** (6.2%)
+ Seq Training	**14.0**	13.7	13.5	**13.4** (4.3%)

Next, we consider to use more utterances to adapt DNNs. As shown in Table 4, we use 20 adaptation utterances per speaker. Since four (4) speakers in the evaluation set (Hub5e00) have fewer than 30 utterances, they are removed from this part of evaluation because it is hard to do CV. Therefore, the baseline performance slightly differs in Table 4. As expected, the results in Table 4 show that adaptation using 20 utterances gives slightly better performance, especially for cross-entropy (CE) trained DNNs. For example, for 6-layer CE DNNs, it reduces WER from 16.3% down to 15.2% (6.8% relative error reduction). For 3-layer DNNs, the gain is much larger than that of 10 adaptation utterances.

Table 4. *WER (in%) of direct adaptation on different DNNs using 20 adaptation utterances per speaker.*

DNN models	baseline	training data size		
		10%	20%	100%
3hid*1024node	19.0	18.2	18.0	17.8 (6.3%)
3hid*2048node	17.4	16.6	16.4	16.1 (7.5%)
6hid*2048node	**16.3**	15.5	15.4	**15.2** (6.8%)
+ Seq Training	**14.0**	14.0	13.8	**13.4** (4.3%)

At last, we consider to use maximum number of utterances per speaker for adaptation, called *max adaptation*. For every test utterance in Hub5e00, we use all remaining utterances from the same speaker to adapt DNNs that is in turn used to recognize only this test utterance. The process is repeated for all utterances in Hub5e00. Since the number of utterances is different for each speaker in test set, adaptation utterances used in this case varies from minimal 25 utterances to maximal 67 utterances per speaker (46 utterances per speaker in average). The results in Table 5 shows that direct adaptation for CE-trained 6-layer DNN can reduce WER from 16.2% down to 14.9%, accounting for about 8.0% relative error reduction. On the other hand, it does not give performance gain by adding more adaptation utterances to adapt sequence-trained DNNs. The main reason is due to the mismatch between the maximum mutual information (MMI) criterion [24, 25] (used for training the baseline DNN) and the cross entropy criterion (used for adaptation). The work to use MMI-based sequence training criterion for adaptation is under way. The results will be reported in the future.

Table 5. *WER (in%) of direct adaptation on different DNNs for max adaptation.*

DNN models	baseline	training data size		
		10%	20%	100%
6hid*2048node	**16.2**	15.3	15.1	**14.9** (8.0%)
+ Seq Training	14.0	13.7	13.8	13.5 (3.6%)

In summary, comparing with other adaptation methods in [11, 12], this direct adaptation method using speaker codes is quite effective not only for small and shallow neural networks but also for large and deep neural networks.

5. CONCLUSION

In this paper, we have proposed an alternative direct adaptation method for DNNs in model space. This method relies on speaker specific compensation that is achieved from learning various speaker codes. Results on large vocabulary Switchboard task show that it can achieve 8% relative reduction in word error rate with only a small number of adaptation utterances. Meanwhile, the proposed direct adaptation scheme also helps to reduce extra training time required for adaptation. We are currently exploring speaker code adaptation using the MMI based sequence training criterion, which will be reported in the near future.

6. REFERENCES

[1] Ossama Abdel-Hamid and Hui Jiang, "Fast speaker adaptation of hybrid NN/HMM model for speech recognition based on discriminative learning of speaker code," in *IEEE International Conference of Acoustics,Speech and Signal Processing (ICASSP)*, 2013.

[2] J. L. Gauvain and Chin-Hui Lee, "Maximum a posteriori estimation for multivariate Gaussian mixture observations of Markov chains," *IEEE Transactions on Speech and audio processing*, vol. 2, no. 2, pp. 291–298, 1994.

[3] S. M. Ahadi and P. C. Woodland, "Combined Bayesian and predictive techniques for rapid speaker adaptation of continuous density hidden Markov models," *Computer speech & language*, vol. 11, no. 3, pp. 187–206, 1997.

[4] Christopher 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.

[5] Mark J. F. Gales, "Maximum likelihood linear transformations for HMM-based speech recognition," *Computer speech & language*, vol. 12, no. 2, pp. 75–98, 1998.

[6] Vassilios V Digalakis, Dimitry Rtischev, and Leonardo G Neumeyer, "Speaker adaptation using constrained estimation of Gaussian mixtures," *IEEE Transactions on Speech and Audio Processing*, vol. 3, no. 5, pp. 357–366, 1995.

[7] Joao Neto, Lus Almeida, Mike Hochberg, Ciro Martins, Lus Nunes, Steve Renals, and Tony Robinson, "Speaker-adaptation for hybrid HMM-ANN continuous speech recognition system," in *EUROSPEECH*, 1995.

[8] Roberto Gemello, Franco Mana, Stefano Scanzio, Pietro Laface, and Renato De Mori, "Linear hidden transformations for adaptation of hybrid ANN/HMM models," *Speech Communication*, vol. 49, no. 10, pp. 827–835, 2007.

[9] Jan Stadermann and Gerhard Rigoll, "Two-stage speaker adaptation of hybrid tied-posterior acoustic models," in *IEEE International Conference of Acoustics,Speech and Signal Processing (ICASSP)*, 2005.

[10] Sabato Marco Siniscalchi, Jinyu Li, and Chin-Hui Lee, "Hermitian based hidden activation functions for adaptation of hybrid HMM/ANN models," in *INTERSPEECH*, 2012.

[11] Dong Yu, Kaisheng Yao, Hang Su, Gang Li, and Frank Seide, "KL-divergence regularized deep neural network adaptation for improved large vocabulary speech recognition," in *IEEE International Conference of Acoustics,Speech and Signal Processing (ICASSP)*, 2013.

[12] Hank Liao, "Speaker adaptation of context dependent deep neural networks," in *IEEE International Conference of Acoustics,Speech and Signal Processing (ICASSP)*, 2013.

[13] Ossama Abdel-Hamid and Hui Jiang, "Rapid and effective speaker adaptation of convolutional neural network based models for speech recognition," in *INTERSPEECH*, 2013.

[14] Bridle J. S. and S. J. Cox, "RecNorm: simultaneous normalization and classification applied to speech recognition," *Advances in Neural Information Processing Systems*, vol. 3, 1991.

[15] Nikko Strom, "Speaker adaptation by modeling the speaker variation in a continuous speech recognition system," in *Spoken Language, 1996. ICSLP 96. Proceedings., Fourth International Conference on*. IEEE, 1996, vol. 2, pp. 989–992.

[16] Shaofei Xue, Ossama Abdel-Hamid, Hui Jiang, and Lirong Dai, "Speaker adaptation of deep neural network based on discriminant codes," *submitted to IEEE Transactions on Acoustics, Speech and Signal Processing*, Feb 2014.

[17] K-F Lee and H-W Hon, "Speaker-independent phone recognition using hidden Markov models," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 37, no. 11, pp. 1641–1648, 1989.

[18] Abdel-rahman Mohamed, George E Dahl, and Geoffrey Hinton, "Acoustic modeling using deep belief networks," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 1, pp. 14–22, 2012.

[19] Jia Pan, Cong Liu, Zhiguo Wang, Yu Hu, and Hui Jiang, "Investigation of deep neural networks (DNN) for large vocabulary continuous speech recognition: Why DNN surpasses GMMs in acoustic modeling," in *8th International Symposium on Chinese Spoken Language Processing (ISCSLP)*, 2012, pp. 301–305.

[20] Yebo Bao, Hui Jiang, Cong Liu, Yu Hu, and Lirong Dai, "Investigation on dimensionality reduction of concatenated features with deep neural network for LVCSR systems," in *IEEE 11th International Conference on Signal Processing (ICSP)*, 2012, vol. 1, pp. 562–566.

[21] Yebo Bao, Hui Jiang, Lirong Dai, and Cong Liu, "Incoherent training of deep neural networks to de-correlate bottleneck features for speech recognition," in *IEEE International Conference of Acoustics,Speech and Signal Processing (ICASSP)*, 2013.

[22] Shiliang Zhang, Yebo Bao, Pan Zhou, Hui Jiang, and Lirong Dai, "Improving deep neural networks for LVCSR using dropout and shrinking structure," in *IEEE International Conference of Acoustics,Speech and Signal Processing (ICASSP)*, 2014.

[23] Hang Su, Gang Li, Dong Yu, and Frank Seide, "Error back propagation for sequence training of context-dependent deep networks for conversational speech transcription," in *IEEE International Conference of Acoustics,Speech and Signal Processing (ICASSP)*, 2013.

[24] Pan Zhou, Lirong Dai, Hui Jiang, Yu Hu, and Qingfeng Liu, "A state-clustering based multiple deep neural networks modelling approach for speech recognition," in *IEEE International Conference of Acoustics,Speech and Signal Processing (ICASSP)*, 2013.

[25] Pan Zhou, Lirong Dai, and Hui Jiang, "Sequence training of multiple deep neural networks for better performance and faster training speed," in *IEEE International Conference of Acoustics,Speech and Signal Processing (ICASSP)*, 2014.