CLUSTER ADAPTIVE TRAINING FOR DEEP NEURAL NETWORK

Tian Tan *Yanmin Qian* *Maofan Yin* *Yimeng Zhuang* *Kai Yu*

Key Laboratory of Shanghai Education Commission for Intelligent Interaction and Cognitive Engineering
SpeechLab, Department of Computer Science and Engineering
Shanghai Jiao Tong University, Shanghai, China
{tantian,yanminqian,__root__,yimmon,kai.yu}@sjtu.edu.cn

ABSTRACT

Although context-dependent DNN-HMM systems have achieved significant improvements over GMM-HMM systems, there still exists big performance degradation if the acoustic condition of the test data mismatches that of the training data. Hence, adaptation and adaptive training of DNN are of great research interest. Previous works mainly focus on adapting the parameters of a *single* DNN by regularized or selective fine-tuning, applying linear transforms to feature or hidden-layer output, or introducing vector representation of non-speech variability into the input. These methods all require relatively large number of parameters to be estimated during adaptation. In contrast, this paper employs the *cluster adaptive training* (CAT) framework for DNN adaptation. Here, *multiple* DNNs are constructed to form the bases of a canonical parametric space. During adaptation, an interpolation vector, specific to a particular acoustic condition, is used to combine the multiple DNN bases into a single adapted DNN. The DNN bases can also be constructed at layer level for more flexibility. The CAT-DNN approach was evaluated on an English switchboard task in unsupervised adaptation mode. It achieved significant WER reductions over the unadapted DNN-HMM, relative 6% to 8.5%, with only 10 parameters.

Index Terms— Cluster Adaptive Training, Deep Neural Network, Adaptation

1. INTRODUCTION

In recent years, context-dependent deep neural network HMM (CD-DNN-HMM) systems have achieved significant performance improvements compared to the conventional GMM-HMM systems [1, 2, 3]. Although the improvements are consistent over all types of acoustic conditions (e.g. speaker, channel or environments), it has been observed that there still exists significant performance degradation if the acoustic condition of the test data mismatches that of the training data [4]. This reveals that there is still a large room for DNN-HMM to improve under mismatched conditions. Therefore, adaptation and adaptive training, being successful in dealing with the acoustic condition mismatch problem in the GMM-HMM era, are attracting more and more research interest. Speaker adaptation is most widely investigated and is also the focus of this paper. It is worth noting that the proposed approaches can be readily used for other acoustic conditions.

Over the past years, various adaptation schemes for DNN have been proposed. Many adaptation approaches for GMM-HMM [5, 6, 7] have been applied on DNN. For example, similar to the idea of maximum a posteriori (MAP) adaptation, the KL-divergence regularization adaptation uses the KL distance to keep the adapted DNN close to a well trained unadapted DNN [8]. Linear transforms are also widely used to adapt feature or hidden-layer output [9, 10, 11, 12]. Layer-wise adaptive training is also implemented for DNN, where the updates of a speaker-dependent layer and the rest speaker-independent layers are interleaved [13]. In addition, a number of adaptation approaches specific to DNN have also been proposed. The basic idea of these approaches is to introduce some speaker-dependent vector representations as input to DNN and allow DNN to learn how to effectively combine the speaker representations with the normal acoustic features. The representation can be estimated either independent of the DNN, such as the iVector adaptation [14, 15, 16], or dependent on the DNN, such as speaker code adaptation [17, 18]. Although the previous adaptation approaches have yielded good gains over unadapted DNN-HMM systems, they all require relatively large number of adaptation parameters to obtain satisfactory gains for large ASR systems. This is largely because all adaptation power is mostly encapsulated in the adaptation parameters. It is worth noting that the iVector and the speaker code approaches introduce connections between the speaker representation vectors and the original DNN layers. These additional parameters also encapsulate useful information for adaptation and only need to be updated during training. Hence, the actual number of parameters for effective adaptation is relatively small, but still at the level of hundred.

In contrast to the previous approaches where only a *single* DNN is used, this paper applies the *cluster adaptive training* (CAT) [19, 20, 21] framework to DNN adaptation. In this framework, *multiple* model sets are used to form the bases of a speaker-independent, canonical parametric space. A speaker-dependent interpolation vector is used to combine the multiple model sets into a single adapted model set. Here, the weight matrices of DNN are regarded as the model set. Since DNN has many layers, the multiple DNN bases can also be constructed at layer level for more flexibility. During training, the bases and the interpolation vectors specific to each training speaker are updated. During adaptation, only the interpolation vectors for each test speaker need to be estimated. With this framework, since the multiple DNN bases bear rich information for speaker adaptation, only very few adaptation parameters, i.e. interpolation vectors, can yield significant performance gains, as shown in later experiments. Also, richer acoustic factors may be easily incorporated by employing DNN bases corresponding to the desired acoustic factors, although this is not the focus of this paper.

The rest of the paper is organized as follows. Section 2 describes the details of applying cluster adaptive training to DNN, followed by the experiments in section 3. Finally, section 4 concludes the whole paper and discusses future works.

This work was supported by the Program for Professor of Special Appointment (Eastern Scholar) at Shanghai Institutions of Higher Learning, the China NSFC project No. 61222208, JiangSu NSF project No. 201302060012 and the EU Framework 7 Program (No. 247619).

2. CLUSTER ADAPTIVE TRAINING FOR DNN

Adaptive training is a structured modelling approach for building systems on non-homogeneous training data. Two distinct sets of model parameters are usually used: a global, speaker-independent *canonical model* representing the desired speech variability; a set of speaker-dependent transforms representing characteristics for each training speaker respectively. The two sets of parameters need to be combined to form the adapted model with standard format. Specifically, in *cluster adaptive training* (CAT) for GMM-HMM [19], the adapted mean of Gaussian component m for speaker s, $\boldsymbol{\mu}^{(sm)}$, is obtained by

$$\boldsymbol{\mu}^{(sm)} = \sum_{c=1}^{P} \lambda_c^{(sr_m)} \boldsymbol{\mu}_c^{(m)}$$

where P is the number of clusters, r_m is the regression base class to which the Gaussian component m belongs and $\boldsymbol{\mu}_c^{(m)}$ is the canonical mean vector of Gaussian m for cluster c. Therefore, the two sets of CAT parameters are:

- **Canonical model:** Each Gaussian has multiple mean vectors and a shared covariance matrix.

$$\mathcal{M} = \left\{ \{\mathbf{M}^{(1)} \ldots \mathbf{M}^{(N)}\}, \{\boldsymbol{\Sigma}^{(1)} \ldots \boldsymbol{\Sigma}^{(N)}\} \right\}$$

  where $\mathbf{M}^{(m)} = [\boldsymbol{\mu}_1^{(m)} \ldots \boldsymbol{\mu}_P^{(m)}]$ is the multiple mean vectors for Gaussian component m and N is the total number of Gaussians. The canonical model parameters are updated using all training data.

- **Speaker interpolation vectors:** A set of interpolation vectors, each specific to speaker s and associated with base class r_m, are used. Each vector is estimated only using the data from the corresponding speaker.

$$\boldsymbol{\lambda}^{(sr_m)} = [\lambda_1^{(sr_m)} \ldots \lambda_P^{(sr_m)}]^\top$$

2.1. DNN with CAT Layers

CAT can be extended to DNN by introducing multiple canonical weight matrices for a DNN layer as depicted in Fig. 1.

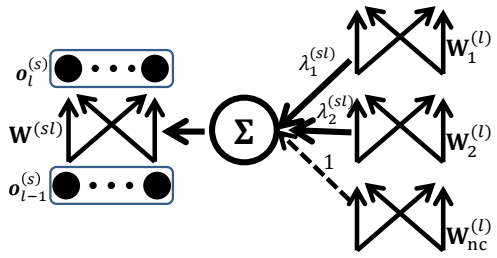

Fig. 1. Architecture of CAT-DNN for one layer

The adapted DNN matrices of layer l for speaker s, $\mathbf{W}^{(sl)}$, is then represented as an interpolation of the canonical DNN matrices:

$$\mathbf{W}^{(sl)} = \sum_{c=1}^{P} \lambda_c^{(sl)} \mathbf{W}_c^{(l)} \tag{1}$$

If multiple layers are constructed as CAT-layers, the parameter sets of CAT-DNN can be written as

$$\begin{aligned} \mathcal{M} &= \left\{ \{\mathbf{M}^{(l_1)} \ldots \mathbf{M}^{(l_L)}\}, \{\mathbf{W}^{(k_1)}, \ldots \mathbf{W}^{(k_K)}\} \right\} \\ \boldsymbol{\lambda}^{(sl)} &= [\lambda_1^{(sl)} \ldots \lambda_P^{(sl)}]^\top \end{aligned}$$

where $\mathbf{M}^{(l)} = [\mathbf{W}_1^{(l)} \ldots \mathbf{W}_P^{(l)}]$ is the set of weight bases of layer l, L is the total number of CAT-layers, $\mathbf{W}^{(k)}$ is the weight matrix of non-CAT layer k and K is the total number of non-CAT layers. Similar to CAT for GMM-HMM, $\boldsymbol{\lambda}^{(sl)}$ denotes the speaker dependent interpolation vectors for layer l and speaker s.

As depicted in Fig.1, a neutral cluster, whose interpolation coefficient is always 1.0, can be introduced to represent cluster-independent aspects. The CAT-layer parameters then become

$$\begin{aligned} \mathbf{M}^{(l)} &= [\mathbf{W}_1^{(l)} \ldots \mathbf{W}_P^{(l)}, \mathbf{W}_{\mathtt{nc}}^{(l)}] \\ \boldsymbol{\lambda}^{(sl)} &= [\lambda_1^{(sl)} \ldots \lambda_P^{(sl)}, 1]^\top \end{aligned}$$

Neutral cluster does not affect the parameter update formula except for always fixing the neutral cluster coefficient to be 1.0.

With the above definitions, the output of CAT-layer l for speaker s, $\boldsymbol{o}_l^{(s)}$, can be defined as below:

$$\boldsymbol{o}_l^{(s)} = \sigma\left(\boldsymbol{s}_l^{(s)}\right), \quad \boldsymbol{s}_l^{(s)} = \mathbf{W}^{(sl)} \boldsymbol{o}_{l-1}^{(s)} + \boldsymbol{b}^{(l)} \tag{2}$$

where $\sigma(\cdot)$ is an element-wise sigmoid function, $\mathbf{W}^{(sl)}$ is constructed using equation (1), $\boldsymbol{b}^{(l)}$ is a speaker independent bias for layer l. The update formula of the CAT-DNN parameters can then be obtained by using the Back-Propagation algorithm with the minimum cross entropy criterion $\mathcal{L}_{\mathtt{ce}}$. For a mini-batch $\mathcal{B}$, the gradients w.r.t. the CAT parameters can be derived as

- **CAT-layer canonical weight matrix for cluster** c

$$\frac{\partial \mathcal{L}_{\mathtt{ce}}}{\partial \mathbf{W}_c^{(l)}} = \frac{1}{N_{\mathcal{B}}} \sum_s \lambda_c^{(sl)} \sum_{\boldsymbol{o}_0^{(s)} \in \mathcal{B}} \frac{\partial \mathcal{L}_{\mathtt{ce}}}{\partial \boldsymbol{s}_l^{(s)}} \boldsymbol{o}_{l-1}^{(s)\top} \tag{3}$$

  where $N_{\mathcal{B}}$ is the number of frames in the mini-batch, $\boldsymbol{o}_0^{(s)}$ denotes the input observation (0^{th} layer) from speaker s, the derivative w.r.t. the combined input $\boldsymbol{s}_l^{(s)}$ can be obtained using standard BP. It is worth noting that all training data need be used to update $\mathbf{W}_c^{(l)}$.

- **Speaker-specific interpolation coefficient**

$$\frac{\partial \mathcal{L}_{\mathtt{ce}}}{\partial \lambda_c^{(sl)}} = \frac{1}{N_{\mathcal{B}}^{(s)}} \sum_{\boldsymbol{o}_0^{(s)} \in \mathcal{B}} \left(\frac{\partial \mathcal{L}_{\mathtt{ce}}}{\partial \boldsymbol{s}_l^{(s)}} \right)^\top \mathbf{W}_c^{(l)} \boldsymbol{o}_{l-1}^{(s)} \tag{4}$$

  where $N_{\mathcal{B}}^{(s)}$ is the number of observations of speaker s in the mini-batch $\mathcal{B}$. Note that only the data of speaker s is used to calculate the gradient for $\lambda_c^{(sl)}$.

The two sets of parameters are updated simultaneously in this paper. After training, only the CAT-DNN is retained for further use and the interpolation vectors for training speakers may be discarded. During adaptation, given the transcriptions of the adaptation data, interpolation vectors are re-estimated for each test speaker. Then the CAT-DNN is interpolated to form a standard DNN for decoding. Compared to the number of parameters required by the other DNN adaptation and adaptive training approaches, e.g. 1000-dimensional vector of speaker code [18], only P parameters are needed in CAT-DNN, which is far fewer than the other methods.

2.2. Initialization of CAT-DNN

Since CAT-DNN makes use of multiple weight bases, it is important to discuss how to initialize the CAT-DNN parameters. Training for CAT-DNN starts after the RBM pretraining, the initialization of weight bases is simply duplicating RBM weight matrices. Initialization of the speaker-dependent interpolation vectors are required for both training and adaptation. It can be done in three ways:

- **Prior knowledge**. This allows the clusters to be associated with meaningful acoustic condition labels. For example, gender information can be used for two clusters, channel information can be used for additional clusters.
- **Automatic data clustering**. Data driven clustering can be performed to construct D homogeneous data block. Then a 1-of-D vector can be used for each data sample as the initial weights. For example, k-means can be applied to iVectors to form speaker groups for cluster initialization.
- **Random initialization**. The cluster interpolation vector can also be randomly initialized. For a P dimensional vector, c is randomly chosen from $\{1, \cdots, P\}$ and $\lambda_c^{(sl)}$ is set 1 and the others are set 0.

It is also possible to initialize CAT-DNN by directly using well trained DNN models with the same structure as the bases. Then the estimation of interpolation vectors can be regarded as a speaker-dependent DNN model combination approach. Since model combination is not the focus of this paper, this is not further discussed.

2.3. CAT-DNN with Different Structures

The CAT-DNN derivation in section 2.1 assumes that each CAT layer has a distinct set of interpolation vectors. A variation of this structure is to tie the interpolation vectors of all CAT layers together. With the tying structure, the cluster interpolation vector becomes $\boldsymbol{\lambda}^{(sl)} = [\lambda_1^{(s)} \ ... \lambda_P^{(s)}]$, where $\lambda_c^{(s)}$ is a global value for all CAT-layers. Consequently, the gradient for $\lambda_c^{(s)}$ becomes

$$\frac{\partial \mathcal{L}_{\texttt{ce}}}{\partial \lambda_c^{(s)}} = \frac{1}{N_{\mathcal{B}}^{(s)} N_{\mathcal{C}}} \sum_{l \in \mathcal{C}} \sum_{\boldsymbol{o}_0^{(s)} \in \mathcal{B}} \left(\frac{\partial \mathcal{L}_{\texttt{ce}}}{\partial \boldsymbol{s}_l^{(s)}} \right)^{\top} \mathbf{W}_c^{(l)} \boldsymbol{o}_{l-1}^{(s)} \quad (5)$$

where $\mathcal{C}$ is the set of CAT-layers, $N_{\mathcal{C}}$ is the number of CAT-layers. As interpolation vector is the only set of parameters to be estimated during adaptation, tying reduces the number of adaptation parameter. However, the adaptation performance may also be affected as the model flexibility is reduced. Hence, whether to choose the tying structure is a trade-off between complexity and performance.

Another variation of CAT-DNN structure is to apply CAT to the bias parameters of DNN. Compared to equation (2) where a speaker-independent bias $\boldsymbol{b}^{(l)}$ is used, introducing bias bases leads to a new formulae to calculate the combined input of layer l

$$\boldsymbol{o}_l^{(s)} = \sigma\left(\boldsymbol{s}_l^{(s)}\right), \quad \boldsymbol{s}_l^{(s)} = \mathbf{W}^{(sl)} \boldsymbol{o}_{l-1}^{(s)} + \boldsymbol{b}^{(l)} + \sum_{c=1}^{P} \lambda_c^{(sl)} \boldsymbol{b}_c^{(l)} \quad (6)$$

where $\boldsymbol{b}_c^{(l)}$ are the bias bases and can be estimated using BP, similar to $\mathbf{W}_c^{(l)}$. Note that for the systems with neutral clusters, since there is a constant 1 in cluster interpolation vector, the independent bias $\boldsymbol{b}^{(l)}$ is redundant and can be removed.

3. EXPERIMENTS

CAT-DNN was evaluated on a 310-hr English Switchboard dataset with 4870 channels. A subset of 51-hr data, 810 channels, was randomly chosen to form a small training set to investigate different CAT-DNN configurations. The full 310-hr Switchboard dataset is then used to evaluate the final performance of CAT-DNN using the configurations learned from 51-hr dataset. The NIST 2000 Hub5e set (referred to as `swb`, 1831 utterances with 40 speakers) and Rich Transcription 2003 set (referred to as `fsh`, 3940 utterances with 72 speakers) were used as the test sets.

13-dimensional PLP features with per-speaker CMN and CVN, along with first and second derivatives were extracted. Two triphone GMM-HMMs model were trained to generate the original state level alignment. The first one with 3001 tied states was used for 51 hours task, another with 9296 tied states was used for 310 hours task. Two types of DNN systems, RBM initialized, 7-hidden layers with 2048 nodes per layer, were trained for 51-hours task and 310-hours task respectively. The initial learning rate is set as 1.6 and reduced by half after four iterations and the learning rates for canonical weight bases and cluster interpolation vectors are the same. A trigram language model which was trained on the transcription of the 2000h Fisher corpus and interpolated with a background trigram model was used for decoding.

During recognition, *unsupervised self adaptation* was used. Hypotheses was first generated using the baseline speaker-independent DNN (SI-DNN) system. State level alignment is then obtained given these hypotheses. Interpolation vectors were then estimated using equation (4) or (5). The initial learning rate for adaptation was 0.2 and reduced by half after the second iteration. Adaptation was stopped after the 8th iteration.

3.1. Investigation of Different Aspects of CAT-DNN

The small 51-hr training set was used for investigation in this section. 2-cluster CAT-DNN systems, initialized using gender information and without neutral cluster and bias bases, were constructed. To make fair comparison, in addition to the SI-DNN baseline, two gender-dependent DNN systems were also built. The GD-DNN systems were directly trained using the male and the female training data respectively and the KL-GD-DNN systems employed the KL-divergence adaptation [8] for each gender to get more robust GD systems. During adaptation, the gender information of the test speakers was assumed to be known and also used for CAT-DNN adaptation initialization.

3.1.1. Effect of Layer Position for a Single CAT Layer

The performance of the baseline systems and the 2-cluster CAT-DNN systems with different layer positions are shown in table 1.

System	CAT Layer	`swb`	`fsh`
SI		25.3	28.9
GD	—	26.1	28.9
GD-KL		24.9	28.6
CAT-DNN	H1	24.1	27.1
	H2	24.2	27.6
	H4	24.4	28.0
	H7	24.9	28.5
	Output	24.8	28.2

Table 1. WER (%) of CAT-DNN with Single CAT Layer

It can be observed that straightforward GD-DNN system is worse than the SI-DNN system, due to reduced training data. KL-divergence adaptation is more robust and yielded better results than the SI-DNN baseline. In contrast, all CAT-DNN systems with a single CAT layer outperformed all baselines. The performance becomes worse as the CAT layer becomes higher. The best CAT-DNN performance is obtained for the first hidden layer. Note that when CAT is applied to the output layer, the dynamic range of the state log-likelihood changes and hence language model scaling factor should be tuned during decoding. In this paper, this effect is not of interest and hence CAT was not applied to the output layer in the below experiments.

3.1.2. Effect of Tying for Multiple CAT Layers

As indicated in section 2.3, when multiple CAT layers are used, tying is an option to reduce adaptation parameter number.

CAT Layer	Tying	# Adapt Param.	swb	fsh
H1	—	2	24.1	27.1
H1-H2	×	4	24.2	27.2
	√	2	24.1	27.7
H1-H7	×	14	24.1	26.5
	√	2	24.4	27.8

Table 2. WER (%) of CAT-DNN with Multiple CAT Layers

From table 2, the use of multiple CAT layers yielded improvements over the best single CAT layer when all hidden layers are CAT layers. Interpolation vector tying always degraded the performance. Hence, in the rest experiments, *no* tying were used.

3.1.3. Effect of Neutral Cluster and Bias Bases

Here, CAT-DNN systems with neutral cluster and speaker dependent bias bases were built to investigate more complicated CAT structure.

CAT Layer	Neutral Cluster	Bias Bases	swb	fsh
H1	×	×	24.1	27.1
	×	√	24.7	27.5
	√	×	24.3	27.3
H1-H2	×	×	24.2	27.2
	×	√	23.8	26.9
	√	×	23.7	26.7
H1-H7	×	×	24.1	26.5
	×	√	24.0	26.4
	√	×	23.6	26.7

Table 3. WER (%) of CAT-DNN w/o Neutral Cluster and Bias Bases

From table 3, the introduction of speaker dependent bias bases and neutral cluster both yielded performance improvements for systems with multiple CAT layers. This shows that more complicated CAT structures are useful. In the rest of experiments, neutral cluster was only employed for CAT-DNN with multiple CAT layers and bias bases were not used, unless explicitly stated.

3.1.4. Effect of Cluster Number and Cluster Initialization

In this section, different number of clusters were investigated. From table 4, for single CAT layer, the performance generally becomes better with more clusters. Especially for 10 clusters, the best performance was obtained. WER for swb reduce from 25.3 to 23.3 (about 7.9% relative error reduction) and from 28.9 to 25.8(about 10.7% relative error reduction) for fsh set. However, for multiple CAT layer, not as expected, more clusters led to increased WER. This may be because 51-hr data is too small to well train complicated models.

CAT Layer	# Cluster	# Adapt Param.	swb	fsh
H1	2	2	24.1	27.1
	5	5	23.7	26.4
	10	10	23.3	25.8
	20	20	23.3	26.1
H1-H2	2	4	23.7	26.7
	5	10	24.1	26.9

Table 4. WER (%) of CAT-DNN with Different Number of Clusters

CAT Layer	# Cluster	Initialization	swb	fsh
H1	2	gender	24.1	27.1
		random	24.2	27.1
	5	kmeans	23.7	26.4
		random	23.6	26.5

Table 5. WER (%) of CAT-DNN with Different Initialization Approaches for Both Training and Adaptation

Table 5 shows the effect of different initialization approaches on a single CAT layer. It can be observed that the CAT-DNN systems performance is not sensitive to initialization. Even random initialization can yield satisfactory results.

3.2. CAT-DNN Performance on Large Training Set

Finally, CAT-DNN systems were built on the full 310-hr switchboard dataset. From table 6, all CAT-DNN systems outperformed the SI-DNN baseline. With the large training dataset, 2 CAT layers with 5 clusters obtained the best performance as expected. It achieved significant gains over the SI-DNN system (relative 8.5% for swb and 6.0% for fsh) with only 10 parameters, demonstrating the effectiveness of CAT-DNN.

CAT Layer	# Cluster	Neutral Cluster	swb	fsh
SI	—	—	19.9	21.5
H1	2	×	19.0	20.5
	5	×	18.9	20.3
H1-H2	2	√	18.7	20.5
	5	√	18.2	20.2

Table 6. WER (%) of CAT-DNN Trained on 310-hr Switchboard Dataset

4. CONCLUSIONS

Cluster adaptive training (CAT) for DNN is introduced in this paper. Instead of using a single DNN, multiple DNNs are trained to form the bases of a speaker-independent, canonical parametric space. An interpolation vector is estimated for each speaker to combine the DNN bases during adaptation. Since interpolation vector is a compact representation of acoustic conditions, much fewer parameters are estimated during adaptation than the other adaptation methods. CAT-DNN yielded significant WER reduction on an English switchboard task with only 10 parameters. Future work will look into combination of multiple acoustic factors, more flexible structure and apply on Sequent-based criterion DNN.

5. REFERENCES

[1] George E Dahl, Dong Yu, Li Deng, and Alex Acero, "Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 20, no. 1, pp. 30–42, 2012.

[2] Geoffrey Hinton, Li Deng, Dong Yu, George E Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara N Sainath, et al., "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *Signal Processing Magazine, IEEE*, vol. 29, no. 6, pp. 82–97, 2012.

[3] Xie Chen, Adam Eversole, Gang Li, Dong Yu, and Frank Seide, "Pipelined back-propagation for context-dependent deep neural networks.," in *INTERSPEECH*, 2012.

[4] Yan Huang, Dong Yu, Chaojun Liu, and Yifan Gong, "A comparative analytic study on the gaussian mixture and context dependent deep neural network hidden markov models," *submitted to the Interspeech*, 2014.

[5] Jean-Luc 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.

[6] Christopher J Leggetter and Philip 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.

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

[8] 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 *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. IEEE, 2013, pp. 7893–7897.

[9] Kaisheng Yao, Dong Yu, Frank Seide, Hang Su, Li Deng, and Yifan Gong, "Adaptation of context-dependent deep neural networks for automatic speech recognition.," in *SLT*, 2012, pp. 366–369.

[10] Frank Seide, Gang Li, Xie Chen, and Dong Yu, "Feature engineering in context-dependent deep neural networks for conversational speech transcription," in *Automatic Speech Recognition and Understanding (ASRU), 2011 IEEE Workshop on*. IEEE, 2011, pp. 24–29.

[11] 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.

[12] Bo Li and Khe Chai Sim, "Comparison of discriminative input and output transformations for speaker adaptation in the hybrid nn/hmm systems," 2010.

[13] Tsubasa Ochiai, Shigeki Matsuda, Xugang Lu, Chiori Hori, and Shigeru Katagiri, "Speaker adaptive training using deep neural networks," in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*. IEEE, 2014, pp. 6349–6353.

[14] George Saon, Hagen Soltau, David Nahamoo, and Michael Picheny, "Speaker adaptation of neural network acoustic models using i-vectors," in *Automatic Speech Recognition and Understanding (ASRU), 2013 IEEE Workshop on*. IEEE, 2013, pp. 55–59.

[15] Andrew Senior and Ignacio Lopez-Moreno, "Improving dnn speaker independence with i-vector inputs," in *Proc. ICASSP*, 2014.

[16] Vishwa Gupta, Patrick Kenny, Pierre Ouellet, and Themos Stafylakis, "I-vector-based speaker adaptation of deep neural networks for french broadcast audio transcription," in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*. IEEE, 2014, pp. 6334–6338.

[17] 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 *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. IEEE, 2013, pp. 7942–7946.

[18] Shaofei Xue, Ossama Abdel-Hamid, Hui Jiang, and Lirong Dai, "Direct adaptation of hybrid dnn/hmm model for fast speaker adaptation in lvcsr based on speaker code," in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*. IEEE, 2014, pp. 6339–6343.

[19] Mark JF Gales, "Cluster adaptive training for speech recognition.," in *ICSLP*, 1998, vol. 1998, pp. 1783–1786.

[20] Mark JF Gales, "Cluster adaptive training of hidden markov models," *Speech and Audio Processing, IEEE Transactions on*, vol. 8, no. 4, pp. 417–428, 2000.

[21] Roland Kuhn, Patrick Nguyen, Jean-Claude Junqua, Lloyd Goldwasser, Nancy Niedzielski, Steven Fincke, Ken Field, and Matteo Contolini, "Eigenvoices for speaker adaptation.," .