Rapid Adaptation for Deep Neural Networks through Multi-Task Learning

Zhen Huang[1], *Jinyu Li*[2], *Sabato Marco Siniscalchi*[1,3], *I-Fan Chen*[1], *Ji Wu*[1*,4] , *Chin-Hui Lee*[1]

[1] School of ECE, Georgia Institute of Technology, Atlanta, GA. USA
[2] Microsoft Corporation, One Microsoft Way, Redmond, WA. USA
[3] Department of Telematics, Kore University of Enna, Enna, Italy
[4] Department of Electronic Engineering, Tsinghua University, Beijing, China

huangzhenee@gatech.edu, jinyli@exchange.microsoft.com, marco.siniscalchi@unikore.it, ichen8@gatech.edu, wuji_ee@mail.tsinghua.edu.cn, chl@ece.gatech.edu

Abstract

We propose a novel approach to addressing the adaptation effectiveness issue in parameter adaptation for deep neural network (DNN) based acoustic models for automatic speech recognition by adding one or more small auxiliary output layers modeling broad acoustic units, such as mono-phones or tied-state (often called senone) clusters. In scenarios with a limited amount of available adaptation data, most senones are usually rarely seen or not observed, and consequently the ability to model them in a new condition is often not fully exploited. With the original senone classification task as the primary task, and adding auxiliary mono-phone/senone-cluster classification as the secondary tasks, multi-task learning (MTL) is employed to adapt the DNN parameters. With the proposed MTL adaptation framework, we improve the learning ability of the original DNN structure, then enlarge the coverage of the acoustic space to deal with the unseen senone problem, and thus enhance the discrimination power of the adapted DNN models. Experimental results on the 20,000-word open vocabulary WSJ task demonstrate that the proposed framework consistently outperforms the conventional linear hidden layer adaptation schemes without MTL by providing 3.2% relative word error rate reduction (WERR) with only 1 single adaptation utterance, and 10.7% WERR with 40 adaptation utterances against the un-adapted DNN models.

Index Terms: deep neural networks, speaker adaptation, multi-task learning, CD-DNN-HMM

1. Introduction

Context-dependent deep neural network hidden Markov models (CD-DNN-HMMs) have outperformed conventional context-dependent Gaussian mixture hidden Markov models (CD-GMM-HMMs) [1] in various tasks and data sets [2]. Unfortunately, CD-DNN-HMMs also suffer from the same performance degradations as CD-GMM-HMMs due to potential acoustic mismatches between the training and testing conditions. In recent years, several adaptation approaches have been proposed to address this mismatch issue, such as regularization based [3, 4], subspace based [5, 6], transformation based [7, 8, 9, 10], i-Vector based [11], native neural network adaptation [12, 13, 14], factorized adaptation [15] and speaker code based [16, 13, 17, 18] techniques. Nonetheless, with CD-DNN-HMM adaptation, the key issue is the large number of DNN parameters employed in order to properly model the tied context-dependent triphone HMM states, sometimes referred to as senones [19]. This often implies that a huge number of output branch parameters connected to the output senone nodes need to be adjusted with but only a small amount of adaptation data is available to be able to cover all senones well. Hence the posterior probabilities of both unseen and scarcely observed senones are be pushed towards zero during adaptation. In addition, DNN parameters are updated all together by each training example, which makes it very difficult to modify parameters only for a set of particular senones. As a result, the ability to model scarcely observed or unseen senones is not as well balanced with that for the observed senones, reducing the adaptation effectiveness in improving the overall ASR performance.

The issues related to scarcely observed or unseen senones could be better contrasted by focusing on the output layer. Indeed, the authors in [20] proposed to directly modify the neural structure at the output layer. Specifically, an additional output layer mapping the original set of DNN output classes to a smaller set of phonetic classes was appended to the original senone output layer, and adaptation was carried out by back-propagation of errors from the new output layer. By appending the new output layer to a set of broad acoustic units, this approach successfully reduces the occurrences of unseen senones in the adaptation data. During recognition, the small output layer is removed, and the senone output layer is actually used. For the sake of completeness, it should be remarked that both the hierarchical MAP approach [21] and the KLD-based regularization technique [4] focus on the output layer but in an indirected way since the actual structure of the output layer is not modified.

Inspired by [20], we propose a novel approach to addressing the data sparsity issue by adding to the original DNN structure one or more small auxiliary output layers to model broad acoustic units, such as mono-phones, or senone clusters. With learning the original senone output layer as the primary task and learning the auxiliary output layers as the secondary tasks, DNN is then adapted through multi-task learning (MTL), a machine learning scheme letting a classifier learn related tasks at the same time [22]. When the tasks are properly chosen, what is learned from one task may be usefully for the other tasks. MLT has been proposed for improving the generalization capability of classifiers, and it has been adopted in various speech recognition tasks, such as isolated digit recognition [23, 24] and phoneme recognition [25], and in speech enhancement tasks [26]. By adding the auxiliary architecture to the original DNN, and performing MTL with the secondary task for recognizing broader sense acoustic units than senones, we improve the learning ability of the original DNN structure, en-

* This work is done while Ji Wu was visiting Georgia Tech.

larging the coverage of acoustic space to better deal with the unseen senone problem and thus enhancing the discrimination power of the adapted DNN models with limited adaptation data. In this framework, the secondary mono-phone/senone-cluster classification task serves as an auxiliary information source for the primary senone classification task. This way we can control the influence of the auxiliary information by choosing proper weights for the objective function of the secondary tasks.

Experimental results of supervised adaptation on the 20,000-word open vocabulary WSJ task demonstrate that the proposed approach consistently outperforms the conventional linear hidden layer adaptation schemes without MTL by providing 3.2% relative word error rate reduction (WERR) with only 1 single adaptation utterance, and 10.7% WERR with up to 40 adaptation utterances against un-adapted DNN models.

2. Training of Deep Models

In DNNs, hidden layers are usually constructed by sigmoid units, and the output layer is a softmax layer. The values of the nodes can therefore be expressed as:

$$\mathbf{x}_i = \begin{cases} \mathbf{W}_1\mathbf{o}^t + \mathbf{b}_1, & i = 1 \\ \mathbf{W}_i\mathbf{y}_{i-1} + \mathbf{b}_i, & i > 1 \end{cases}, \quad (1)$$

$$\mathbf{y}_i = \begin{cases} \mathrm{sigmoid}(\mathbf{x}_i), & i < L \\ \mathrm{softmax}(\mathbf{x}_i), & i = L \end{cases}, \quad (2)$$

where $\mathbf{W}_1$, and $\mathbf{W}_1$ are the weight matrices, $\mathbf{b_1}$, and $\mathbf{b_i}$ are the bias vectors, $\mathbf{o}^t$ is the input frame at time t, L is the total number of the hidden layers, and both sigmoid and softmax functions are element-wise operations. The vector $\mathbf{x}_i$ corresponds to pre-nonlinearity activations, and $\mathbf{y}_i$ and $\mathbf{y}_L$ are the vectors of neuron outputs at the i^{th} hidden layer and the output layer, respectively. The softmax outputs were considered as an estimate of the senone posterior probability:

$$p(C_j|\mathbf{o}^t) = \mathbf{y}_L^t(j) = \frac{\exp(\mathbf{x}_L^t(j))}{\sum_i \exp(\mathbf{x}_L^t(i))}, \quad (3)$$

where C_j represents the j^{th} senone and $\mathbf{y}_L(j)$ is the j^{th} element of $\mathbf{y}_L$.

The DNN is trained by maximizing the log posterior probability over the training frames. This is equivalent to minimizing the cross-entropy objective function. Let $\mathcal{X}$ be the whole training set, which contains T frames, *i.e.* $\mathbf{o}_{1:T} \in \mathcal{X}$, then the loss with respect to $\mathcal{X}$ is given by

$$\mathcal{L}^{1:T} = -\sum_{t=1}^{T}\sum_{j=1}^{J} \tilde{\mathbf{p}}^t(j) \log p(C_j|\mathbf{o}^t), \quad (4)$$

where $p(C_j|\mathbf{o}^t)$ is defined in Eq. (3) and $\tilde{\mathbf{p}}^t$ is the target probability of frame t. In real practices of DNN systems, the target probability $\tilde{\mathbf{p}}^t$ is often obtained by a forced alignment by with an existing system resulting in only the target entry that is equal to 1.

The objective function is minimized by using error backpropagation [27] which is a gradient-descent based optimization method developed for neural networks. Specifically, taking partial derivatives of the objective function with respect to the pre-nonlinearity activations of output layer $\mathbf{x}_L$, the error vector to be backpropagated to the previous hidden layers is generated:

$$\boldsymbol{\epsilon}_L^t = \frac{\partial \mathcal{L}_{1:T}}{\partial \mathbf{x}_L} = \mathbf{y}_L^t - \tilde{\mathbf{p}}^t, \quad (5)$$

the backpropagated error vector at previous hidden layer is,

$$\boldsymbol{\epsilon}_i^t = \mathbf{W}_{i+1}^* \boldsymbol{\epsilon}_{i+1}^t \times \mathbf{y}_i \times (\mathbf{1} - \mathbf{y}_i), i < L \quad (6)$$

where $\mathbf{W}_{i+1}^*$ is the transpose of $\mathbf{W}_{i+1}$, and $\times$ denotes element-wise multiplication.

Mini-batch stochastic gradient descent (SGD) [28], with a reasonable size of mini-batches to make all matrices fit into the GPU memory, was used to update all neural network parameters during training. Pre-training methods was used for the initialisation of the DNN parameters [29].

3. DNN adaptation with auxiliary output layers through MTL

A well trained DNN usually employs a large output layer directly modeling senones. As mentioned in Section 1, there are various approaches to perform adaptation for DNN models. But none of these approaches really look into the problem that for some adaptation scenarios the available adaptation data is often very limited, so that the adaptation data might not be able to cover all the senones. In this paper, we address this problem by adding one or more smaller auxiliary output layers to the original DNN architecture.

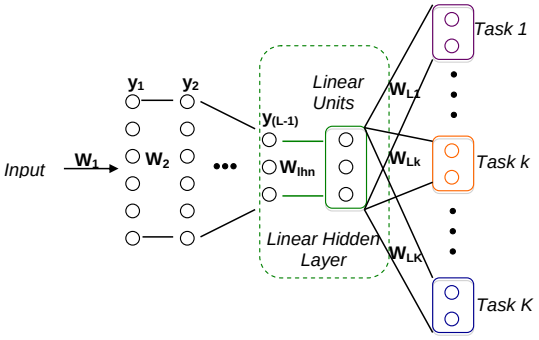

Figure 1: Basic neural architecture for adaptation of HMM/ANN models based on LHN through MTL. The output layers are associated with different tasks. In adaptation, the parameters (weights) of the LHN associated to the links in the dashed rectangle are estimated while all other weights remain unchanged. The activation function of each LHN units (green nodes) is a linear function.

3.1. Multi-task Learning

By adding the auxiliary architecture, the new DNN will have more than one output layers. This kind of multi-output-layer DNN can be trained using the MTL scheme. MTL is a machine learning scheme letting a classifier learn more than one related tasks at a time [22]. As in Figure 1, there are multiple output layers corresponding to different tasks, and the error vector from each output layer will be backpropagated to the same last hidden layer, then the vector will be combined together in a weighted sum fashion as in

$$\boldsymbol{\epsilon}_{MTL} = \sum_{k=1}^{K} w_k \boldsymbol{\epsilon}_k \quad (7)$$

where $\boldsymbol{\epsilon}_k$ is the error vector from the kth output layer and w_k is the weight for the corresponding task. The combined error vector $\boldsymbol{\epsilon}_{MTL}$ is then backpropagated to previous hidden layers.

When the tasks are properly chosen, what is learned from one task will help the other tasks and usually there will be a primary task and several secondary tasks to aid the primary one. In this paper, the primary task is classification of senones and the secondary tasks are set to classify broader sense acoustic units such as mono-phones or senone-clusters. By doing so, the broader sense acoustic units in the added output layers help improve the learning ability of the original DNN structure, enlarge the coverage of acoustic space to better deal with the unseen senone problem and thus enhance discrimination power of adapted DNN models with limited adaptation data.

The secondary tasks in this paper for limited resource adaptation need broader sense acoustic units than senones as classification targets. Obviously mono-phones suits such purpose because a mono-phone is usually corresponds to tens or hundreds of senones. To some extent, a mono-phone can be deemed as a rule-driven cluster of senones.

It is shown in [30], that a log-linear model is equivalent to a Gaussian model. The general form of a log-linear model is

$$p(C_j|\mathbf{x}) = \frac{1}{\mathrm{Z}(\mathbf{x})}\exp(\sum_i \lambda_{ji} f_i(\mathbf{x})), \tag{8}$$

where $f_i(x)$ is the i_{th} feature function for input x, λ_{ji} is the weight for the j_{th} class and i_{th} feature, and $\mathrm{Z}(x)$ is for normalization. The mapping from a log-linear model to a Gaussian model $N(\mathbf{x}|\mu_j, \mathbf{\Sigma}_j)$ is

$$\mathbf{\Sigma}_j = \frac{1}{2}(\boldsymbol{\lambda}_j^2 + \Delta\boldsymbol{\lambda}^2)^{-1}, \quad \boldsymbol{\mu}_j = \mathbf{\Sigma}_j\boldsymbol{\lambda}_j^1 \tag{9}$$

where $\boldsymbol{\lambda}_j^2$ and $\boldsymbol{\lambda}_j^1$ are the second- and first-order weights, $\Delta\boldsymbol{\lambda}^2$ is a constant to make $\mathbf{\Sigma}_j$ positive definite.

The softmax function used in the DNNs' output layer can be considered as a log linear model. From the practice of [31], we know that every senone can be represented by a Gaussian distribution with

$$\mathbf{\Sigma}_j = \mathbf{\Sigma}, \quad \mu_j = \mathbf{\Sigma}[\mathbf{W}_j, \mathbf{b}_j] \tag{10}$$

where $\mathbf{W}_j, \mathbf{b}_j$ are the weights and bias for senone j and $\mathbf{\Sigma}$ can be an arbitrary positive definite matrix, so for any pair of senones, the symmetric KL divergence could be used as their distance measure. With the distance measure, the large set of senones can be clustered into a small set by various clustering technologies. We use the method in [31] to generate senone clusters, which are data-driven acoustic units.

3.2. Linear Hidden Network Adaptation

Nearly all of the adaptation technologies for DNN model mention in Section 1 will fit the proposed MTL adaptation framework. In this paper, we choose the very commonly used method, linear hidden network (LHN), as the basic adaptation approach to demonstrate effectiveness of the proposed MTL adaptation framework. The LHN adaptation is performed by adding an affine transformation network between the last hidden layer and the output layer, and adapting only the augmented LHN's parameters while keeping fixed all of the other DNN parameters [8]. In order to reduce the amount of parameters to adapt, usually the last hidden layer is designed to be a bottleneck (less neurons), and superior results were obtained by this kind of LHN formulation [32, 33]. The LHN adaptation structure can be found in Figure 1. If we deem the hidden layers as a feature extractor and the output layer as the discriminative model, the LHN formulation is quite similar to maximum likelihood linear regression (MLLR) [34]. The difference is that in MLLR the model parameters are Gaussian mean and variance while here the model parameters are the log-linear model's transformation matrix weights.

4. Experiments

4.1. Baseline DNN Setup

This study is concerned with supervised *speaker adaptation*. Experiments are reported on the 20K-word open vocabulary Wall Street Journal task [35] using the Kaldi toolkit [36].

The baseline CD-DNN-HMM system for was trained using the WSJ0 material (SI-84). The standard adaptation set of WSJ0 (si_et_ad, 8 speakers, 40 sentences per speaker) was used to perform adaptation of the affine transformation added to the speaker-independent DNN. The standard open vocabulary 20,000-word (20K) read NVP Senneheiser microphone (si_et_20, 8 speakers x 40 sentences) data were used for evaluation. Training was stopped using an held-out set comprising the si_dt_20 WSJ0 data. A standard trigram language model was adopted during decoding. The ASR system performance was given in terms of the word error rate (WER).

The DNN has six hidden layers. The first five hidden layers have 2048 units, whereas the last hidden layer has 216 units. The output layer has 2022 softmax units corresponding to the senones generated using a CD-GMM-HMM baseline. This DNN architecture follows conventional configurations used in the speech community except for the last hidden layer, which acts as a bottleneck layer. This configuration was chosen, because a too large dimension of the last nonlinear hidden layer might have been harmful to LHN adaptation. The bottleneck based low rank methods has been widely used to achieve more compact DNN models with equivalent performance [37, 38]. The number 216 was chosen to simulate a sort of three-state phone layer thereby obtaining a kind of hierarchical structure between mono-phones in the hidden layer and senones at the output layer. The input feature vector is a 23-dimension mean-normalized log-filter bank feature with up to second-order derivatives and a context window of 11 frames, forming a vector of 759-dimension (69×11) input. The DNN was trained with an initial learning rate of 0.008 using the cross-entropy objective function. It was initialised with the stacked restricted Boltzmann machines (RBMs) by using layer by layer generative pre-training.

4.2. Adaptation Setup

In order to perform MTL adaptation, the auxiliary output layers should first be prepared. For each auxiliary task, we first randomly initialised the additional output layer's affine transformation matrix and then use the training data to fine-tune them while keeping all other parameters in the DNN fixed. Fine-tuning used an initial learning rate of 0.0005 with the cross-entropy based objective function. In this 20K-word open vocabulary WSJ task, the mono-phone output layer has 42 units and so is the senone-cluster output layer.

After we obtained the auxiliary output layers, an LHN was inserted between the last hidden (bottleneck) layer and the output layers' affine transform matrix. The 216×216 LHN is initialized to an identity matrix with zero bias, which gave a starting point equivalent to the unadapted model. Supervised adaptation is then performed updating only the LHN parame-

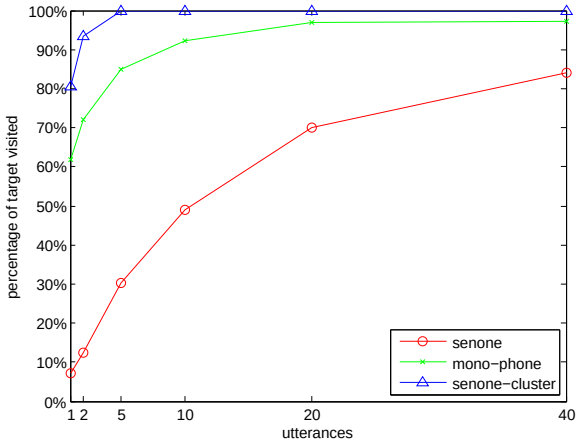

Figure 2: Percentage of the target units (senones/monophones/senone-clusters) visited and adapted, with respect to the number of adaptation utterances

ters. In decoding, only the original senone based outputs were used while the auxiliary architecture was discarded at this stage.

4.3. Adaptation Results

Table 1 shows the adaptation results of MTL adaptation in the 20K-word open vocabulary WSJ experiments. All the results were obtained by adding only one single auxiliary task, either mono-phone or senone-cluster classification, to the original senone classification task. We tried to include both of the two auxiliary tasks, but the results are similar to those by just adding a single task. The w in the tables means the weight of the objective function for the auxiliary task when combined with the original one. As there were only two tasks during each adaptation process, the original senone classification task's weight is therefore $1-w$. $w = 0$ means plain LHN adaptation without the auxiliary task and $w = 1$ means there is only the auxiliary task being performed. The word "phone" means the mono-phone task and word "cluster" means the senone-cluster task.

From Table 1, it is demonstrated that the proposed MTL adaptation consistently outperforms plain LHN adaptation without MTL, especially in the case of limited adaptation data. 3.2% WERR is gained from speaker independent DNN with only 1 single adaptation utterance and 10.7% WERR with up to 40 utterances in the WSJ experiments.

4.4. Result Analysis and Discussions

Figure 2 shows the percentage of visited targets (senones/monophones/senone-clusters) with respect to the number of adaptation utterances. It can be observed that the coverage of the acoustic space reaches about 100% with only 5 utterances by using senone-cluster target units. The mono-phones units also shows a good characteristic by covering more than 90% the acoustic space with only 10 utterances. However, even with the complete adaptation set, i.e. 40 utterances, the adaptation data covers far below 90% of the targets when the senone is used.

From Table 1, the best results for the extreme resource-limited cases, i.e., 1 or 2 utterances, is obtained by using senone-clusters as the auxiliary task's classification targets. This phenomenon is consistent with what is shown in Figure 2, i.e., a high coverage (80%) of the acoustic space can be reached

Table 1: WER obtained by MTL adaptation through monophone/senone-cluster auxiliary task with different weight and different amounts of adaptation data in the 20K-word open vocabulary WSJ experiments experiments

# Adaptation Sentences	w=0 no MTL	w=0.75		w=1	
		phone	cluster	phone	cluster
BASELINE	8.84%				
1	8.79%	8.56%	**8.36%**	8.65%	8.58%
2	8.58%	8.42%	**8.42%**	8.54%	8.52%
5	8.59%	**8.38%**	8.68%	8.44%	8.67%
10	8.52%	**8.33%**	8.47%	8.45%	8.42%
20	8.31%	**8.01%**	8.22%	8.45%	8.47%
40	8.22%	**7.89%**	8.17%	8.28%	8.58%

with only 2 utterances by using senone-clusters. When the adaptation utterances increase, the coverage gap between monophone and senone-cluster decreases, and mono-phone MTL becomes slightly better possibly due to good linguistic knowledge in contrast with the data-driven senone-cluster.

One important point needed to be mentioned is that in decoding only the original senone based output layer is used. The auxiliary architecture was discarded at this stage. But there is an interesting phenomenon that even we adapted the DNN only using the auxiliary task associated with mono-phone/senone-cluster ($w = 1$ case in the tables), there is still an improvement from the baseline DNN, and sometimes even better than using the original primary task in some data-limited cases. This phenomenon further demonstrates that the information introduced by the auxiliary tasks can be quite effective in the limited resource scenarios.

5. Conclusions

We propose a novel approach to addressing the data sparsity problem in CD-DNN-HMM adaptation by adding one or more small auxiliary output layers modeling broad acoustic units, such as mono-phones or senone-clusters, to the original DNN structure, and update the DNN parameters through MTL. By doing so, we improve the learning ability of the original DNN structure, enlarge the coverage of the acoustic space to better deal with the unseen senone problem, and thus enhance the discrimination power of the adapted DNN models with limited adaptation data. We show the effectiveness of the proposed framework in the 20K-word open vocabulary WSJ task. Experimental results shows the proposed method consistently outperforms the conventional linear hidden layer adaptation schemes without MTL. With only 1 single adaptation utterance, a WER reduction of 3.2% is obtained from the speaker independent DNN models and a 10.7% WERR can be achieved by using 40 utterances.

In the future, a combination of the Bayesian adaptation framework [21] and the proposed MTL scheme can be exploited. Other kind of auxiliary tasks such as speech enhancement and speaker verification could be investigated. Another interesting direction is to automatically choose the sizes of the secondary output layers and the weights to combine primary and secondary objective functions according to the data amount, task characteristics and other side information related to ASR.

6. References

[1] L. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, 1989.

[2] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath *et al.*, "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.

[3] X. Li and J. Bilmes, "Regularized adaptation of discriminative classifiers," in *Proc. ICASSP*, vol. 1, 2006, pp. I–I.

[4] 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, pp. 7893–7897.

[5] D. Yu, X. Chen, and L. Deng, "Factorized deep neural networks for adaptive speech recognition," in *Proc. Int. Workshop on Statistical Machine Learning for Speech Processing*, 2012.

[6] D. Yu, L. Deng, and S. Seide, "The deep tensor neural network with applications to large vocabulary speech recognition," *IEEE Trans. Audio, Speech, and Language Processing*, vol. 21, no. 2, pp. 388–396, 2013.

[7] 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*, 1995.

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

[9] 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, pp. 24–29.

[10] 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. Spoken Language Technology Workshop*, 2012, pp. 366–369.

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

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

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

[14] P. Swietojanski and S. Renals, "Learning hidden unit contributions for unsupervised speaker adaptation of neural network acoustic models," in *Proc. IEEE STL*, 2014.

[15] J. Li, J.-T. Huang, and Y. Gong, "Factorized adaptation for deep neural network," in *Proc. ICASSP*, 2014.

[16] 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, pp. 7942–7946.

[17] 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. ICASSP*, 2014, pp. 6339–6343.

[18] S. Xue, O. Abdel-Hamid, H. Jiang, L. Dai, and Q. Liu, "Fast adaptation of deep neural network based on discriminant codes for speech recognition," *IEEE/ACM Trans. on Audio, Speech and Lang. Proc.*, vol. 22, no. 12, pp. 1713–1725, 2014.

[19] M.-Y. M.-Y. Hwang and X. Huang, "Shared-distribution hidden markov models for speech recognition," *IEEE Trans. Speech and Audio Processing*, vol. 1, no. 4, pp. 414–420, 1993.

[20] R. Price, I. Kenichi, and K. Shinoda, "Speaker adaptation of deep neural networks using a hierarchy of output layers," in *Proc. SLT*, 2014.

[21] Z. Huang, S. M. Siniscalchi, I.-F. Chen, J. Wu, and C.-H. Lee, "Maximum a posteriori adaptation of network parameters in deep models," 2015, submitted to INTERSPEECH.

[22] R. Caruna, "Multitask learning: A knowledge-based source of inductive bias," in *Proc. ICML*, 1993, pp. 41–48.

[23] S. Parveen and P. Green, "Multitask learning in connectionist robust asr using recurrent neural networks." in *Proc. INTERSPEECH*, 2003.

[24] Y. Lu, F. Lu, S. Sehgal, S. Gupta, J. Du, C. Tham, P. Green, and V. Wan, "Multitask learning in connectionist speech recognition," in *Proc. Australian International Conference on Speech Science and Technology*, 2004.

[25] M. Seltzer and J. Droppo, "Multi-task learning in deep neural networks for improved phoneme recognition," in *Proc. ICASSP*, 2013, pp. 6965–6969.

[26] Y. Xu, J. Du, Z. Huang, L.-R. Dai, and C.-H. Lee, "Multi-objective learning and mask-based post-processing for deep neural network based speech enhancement," 2015, submitted to INTERSPEECH.

[27] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, *Learning representations by back-propagating errors*. MIT Press, Cambridge, MA, USA, 1988.

[28] O. Dekel, R. Gilad-Bachrach, O. Shamir, and L. Xiao, "Optimal distributed online prediction," in *Proc. ICML*, 2011, pp. 713–720.

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

[30] G. Heigold, H. Ney, P. Lehnen, T. Gass, and R. Schluter, "Equivalence of generative and log-linear models," *IEEE Trans. Audio, Speech & Language Processing*, vol. 19, no. 5, pp. 1138–1148, 2011.

[31] J. Li, R. Zhao, J.-T. Huang, and Y. Gong, "Learning small-size dnn with output-distribution-based criteria," in *Proc. Interspeech*, 2014.

[32] 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."

[33] S. Xue, H. Jiang, and L. Dai, "Speaker adaptation of hybrid NN/HMM model for speech recognition based on singular value decomposition," in *Proc. ISCSLP*, 2014.

[34] 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.

[35] D. B. Paul and J. M. Baker, "The design for the wall street journal-based CSR corpus," in *Proc. Workshop on Speech and Natural Language*, 1992, pp. 899–902.

[36] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, J. Silovskỳ, G. Stemmer, and K. Veselỳ, "The Kaldi speech recognition toolkit," in *Proc. ASRU*, 2011.

[37] 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 *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. IEEE, 2013, pp. 6655–6659.

[38] J. Xue, J. Li, and Y. Gong, "Restructuring of deep neural network acoustic models with singular value decomposition." in *INTERSPEECH*, 2013, pp. 2365–2369.