LOW-RANK PLUS DIAGONAL ADAPTATION FOR DEEP NEURAL NETWORKS

Yong Zhao, Jinyu Li, and Yifan Gong

Microsoft Corporation, One Microsoft Way, Redmond, WA 98052, USA

{yonzhao; jinyli; ygong}@microsoft.com

ABSTRACT

In this paper, we propose a scalable adaptation technique that adapts the deep neural network (DNN) model through the low-rank plus diagonal (LRPD) decomposition. It is desired that an adaptation method can properly accommodate the available development data with a variable amount of adaptation parameters. Thus, the resulting models neither over-fit nor under-fit as the development data vary in size for different speakers. The technique developed in this paper is inspired by observing that adaptation matrices are very close to an identity matrix or diagonally dominant. The LRPD restructures the adaptation matrix as a superposition of a diagonal matrix and a low-rank matrix. By varying the low-rank values, the LRPD contains the full and the diagonal adaptation matrix as its special cases. Experimental results demonstrated that the LRPD adaptation of the full-size DNN obtains improved accuracy over the standard linear adaptation. The LRPD bottleneck adaptation can reduce the speaker-specific footprint by 82% over an already very compact SVD bottleneck adaptation, at an expense of 1% relative WER increase.

Index Terms: automatic speech recognition, deep neural network, speaker adaptation

1. INTRODUCTION

The discovery of the strong modeling capabilities of deep neural networks (DNNs) [1], [2], [3], [4] and the availability of high-speed hardware has made it feasible to train large networks with tens of millions of parameters. In the framework of context-dependent DNN hidden-Markov-models (CD-DNN-HMMs) [1], the conventional Gaussian Mixture Model (GMM) is replaced by a DNN to evaluate the senone log-likelihood.

Despite its outstanding performance, CD-DNN-HMMs may still suffer from the accuracy degradation due to the potential acoustic mismatch between the training and test conditions. A simple and popular approach to combat the mismatch problem is to adapt the DNN model by applying a linear transformation on either input features, hidden layers, or the input to the softmax layer [5, 6, 7, 8, 9]. One issue is that, due to the high dimensionality of the DNN layers, the transformation matrix in a full form would result in a large speaker-dependent parameter set. When the amount of adaptation data is limited, overfitting becomes a severe issue.

To address this issue, [10] adds Kullback-Leibler divergence (KLD) regularization to the training criterion to prevent the adapted model from straying too far from the speaker independent (SI) model. Many other methods have been proposed as low-footprint adaptation techniques, which require a small number of speaker-dependent parameters. Feature discriminative linear regression (fDLR) [9] transforms the input features by sharing the same transform between each of the input frames. This amounts to replace the full transformation matrix with a tied block-diagonal matrix. In [11], the original large full-size DNN model is converted to a much smaller low-rank DNN model by using singular value decomposition (SVD). Then, SVD bottleneck adaptation is done by applying the linear transform to the bottleneck layer. Thus, only a few matrices of much lower dimension need to be updated for each speaker.

Furthermore, we can, instead of adapting the whole matrix, only adapt the diagonal matrix, when the adaptation data is limited [9, 12]. A DNN has been adapted by adjusting amplitude, bias and slopes in the node activation functions, such as learning hidden unit contribution (LHUC) [13] and sigmoid adaptation [14]. Likewise, this is equivalent to adding a linear layer with diagonal transform after or before the activation functions . The LHUC and sigmoide adaptation have much smaller number of parameters to adapt comparing to the SVD adaptation, while they may not yield similar accuracy improvement along with the increase of the adaptation data.

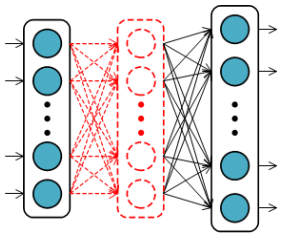
In an extreme case, the adaptation matrix can be fixed to an identity matrix and only bias vectors are adjusted. Interestingly, a family of adaptation methods using auxiliary features falls into this category. The speaker-specific auxiliary features can be derived in many different ways, such as i-vector [15, 16, 17] and speaker code [18, 19, 20], and then are concatenated with the standard acoustic feature vectors. Because i-vectors or speaker codes are of low dimension, this method works well for rapid speaker adaptation with a limited number of utterances. After simple algebraic manipulation, it can be shown that the augmentation of auxiliary features can be regarded as adjusting the bias vector on the first hidden layer [21].

It is remarkable that the employed adaptation methods highly relies on the topology of the underlying SI DNN model. When a full-size DNN is used, a full transformation matrix is prone to overfitting the limited adaptation data, and a diagonal matrix, or just bias vector, is preferred. However, the use of the diagonal matrix may sacrifice the adaptation gain, especially when adaptation data become sufficient. To make the model scalable, it is desired that the model footprint can be adjusted according to the amount of available adaptation data.

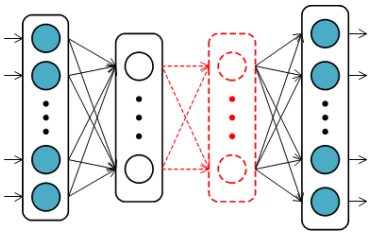
In this paper, we first examine adaptation matrices and discover that they are very close to an identity matrix or diagonally dominant. Therefore, we decompose the adaptation matrix into two parts, one is a diagonal matrix and the other is a low-rank matrix which is further decomposed into two smaller matrices by SVD. This method can be applied to adapt either the full-size or the SVD SI DNN model. The latter leads to a combination of the SVD bottleneck and the LRPD adaptation. The rest of this paper is organized as follows. We will first briefly introduce the SVD-based DNN adaptation in Section 2. In Section 3, we propose the low-rank plus diagonal method for DNN adaptation. Then, we evaluate our proposed method and compare it with the existing adaptation methods in Section 4, and conclude the study in Section 5.

2. SVD-BASED DNN ADAPTATION

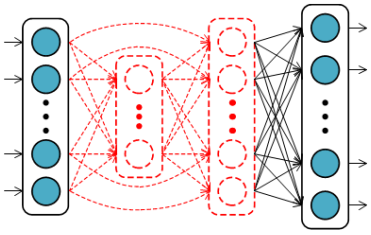
A DNN can be considered as a conventional multi-layer perceptron (MLP) with many hidden layers, where the input feature is concatenated from multiple consecutive frames and the output predicts the posterior probabilities of thousands of senones. Given a DNN with L hidden layers, the activation output at the l-th hidden layer, $\boldsymbol{h}^l$, is recursively defined as the nonlinear transformation of the $(l-1)$-th



(a) General linear adaptation



(b) SVD bottleneck adaptation



(c) LRPD adaptation

Fig. 1: Illustration of network structures of different adaptation methods. Shaded nodes denote nonlinear units, unshaded nodes for linear units. Red dashed links indicate the transformations that are introduced during adaptation.

layer:

$$\boldsymbol{h}^l = \sigma(\boldsymbol{W}^l \boldsymbol{h}^{l-1} + \boldsymbol{b}^l) \tag{1}$$

where $\boldsymbol{W}^l$ is the weight matrix, $\boldsymbol{b}^l$ is the bias vector, and $\sigma(\cdot)$ is an element-wise sigmoid activation function. $\boldsymbol{h}^0 = \boldsymbol{x}$ is the input observation vector. For CD-DNN-HMMs [1], the output layer is normalized by the softmax function to produce the posterior probability of senone id s, $p(s|\boldsymbol{x})$.

Many DNN adaptation techniques have been developed in the past. The most popular approach for adapting DNNs is applying a linear transformation to a certain DNN layer to account for the mismatch between the training and test conditions. The adaptation parameters consist of a speaker-specific transformation matrix $\boldsymbol{W}_s$ and a bias vector $\boldsymbol{b}_s$. The basic idea of this model is illustrated in Fig. 1a. Note that the parameters corresponding to the red dashed links are trained using the adaptation set, while other weights of the model are fixed during adaptation. One main issue in these adaptation techniques is that they typically need to update and store a large amount of adaptation parameters due to the high dimensionality of the DNN layers.

2.1. SVD bottleneck adaptation

We recently presented a SVD-based method in [22] to restructure the DNN model in a significantly small size while maintaining the recognition accuracy. Given an $m \times n$ weight matrix $\boldsymbol{W}$ in a DNN, we approximate it as the product of two low-rank matrices by applying SVD

$$\boldsymbol{W}_{m\times n} \approx \boldsymbol{U}_{m\times k}\boldsymbol{V}_{k\times n} \tag{2}$$

where k is the number of retained singular values. Since $k \ll min(m,n)$, the number of parameters is significantly reduced from mn to $(m+n)k$. Applying this decomposition to the weight matrix, it acts as if inserting a linear bottleneck layer of fewer units between the original nonlinear layers. After the SVD decomposition, it is also possible to finetune $\boldsymbol{U}$ and $\boldsymbol{V}$ to recover the accuracy loss.

One important application of the SVD DNN is to allow efficient adaptation by applying the linear transformation on the bottleneck layers [11], as illustrated in Fig. 1b. We have

$$\boldsymbol{W}_{s,m\times n} = \boldsymbol{U}_{m\times k}\boldsymbol{S}_{s,k\times k}\boldsymbol{V}_{k\times n} \tag{3}$$

where $\boldsymbol{S}_{s,k\times k}$ is the adaptation matrix for speaker s and is initialized to be an identity matrix $\boldsymbol{I}_{k\times k}$. The advantage of this approach is that only a couple of small matrices need to be updated for each speaker. Consider $k = 256$ and $m = n = 2048$ as an instance. Directly adapting the initial matrix $\boldsymbol{W}$ needs to update $2048 \times 2048 = 4M$ parameters, while adapting $\boldsymbol{S}$ only updates $256 \times 256 = 64K$ parameters, reducing the footprint to 1.6%. This dramatically reduces the deployment cost for speaker personalization while producing more reliable estimate of the adaptation model [11].

3. LOW-RANK PLUS DIAGONAL (LRPD) ADAPTATION FOR DEEP NEURAL NETWORKS

To make the model scalable, it is desired that the model footprint can be adjusted according to the amount of available adaptation data. In this section, we propose the LRPD adaptation method to control the number of adaptation parameters in a flexible way while maintaining the same accuracy. The LRPD can be applied on either full-size or SVD DNNs.

3.1. Properties of adaptation matrix

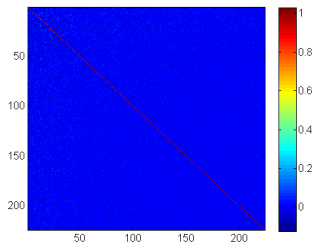
One simple heuristic is to re-apply the SVD to decompose the adaptation matrix $\boldsymbol{W}_s$. However, such decomposition would not yield expected performance.

Fig. 2a shows an example of 224×224 adaptation matrix, trained using 50 speaker-specific utterances starting from an SVD DNN. Clearly, the matrix is very close to an identity matrix and of full rank. This is indeed reasonable as the adapted model should not deviate too far from the SI model given the limited number of adaptation data. Even if the adaptation data are sufficient, there should exist a one-to-one correspondence between the speaker-dependent excitation/activation and that of the average speaker in the SI model. Thus the adaptation matrix should be diagonally dominant. It is found that after subtracting an identify matrix from the adaptation matrix, the magnitudes of all the elements are below 0.2, and 99% below 0.05.

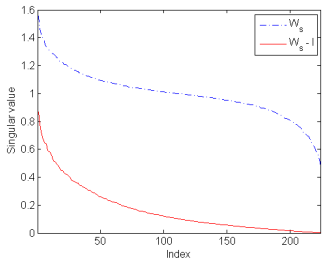
Fig. 2b compares the singular values between an adaptation matrix $\boldsymbol{W}_s$ and the one minus an identity matrix $(\boldsymbol{W}_s - \boldsymbol{I})$. Because $\boldsymbol{W}_s$ is close to an identity matrix, singular values of $\boldsymbol{W}_s$ center around 1 and decrease slowly. We cannot effectively compress such a high-rank matrix by SVD. In contrast, the singular values of $(\boldsymbol{W}_s - \boldsymbol{I})$ steadily decrease approaching zero. Around 17% of singular values contribute 50% of total values, and around 40% contribute 80% of total values. This indicates that we should be able to apply SVD to $(\boldsymbol{W}_s - \boldsymbol{I})$ to reduce the speaker-dependent footprint.

3.2. LRPD adaptation

In the following, we describe in detail the proposed low-rank plus diagonal (LRPD) adaptation method. Given an adaptation matrix



(a) An adaptation matrix



(b) Singular values of the adaptation matrix and the one after subtracting an identity matrix from it.

Fig. 2: Illustration of an adaptation matrix and its corresponding singular values.

$\boldsymbol{W}_{s,k\times k}$, which can be $\boldsymbol{S}_{s,k\times k}$ in the SVD bottleneck adaptation, we approximate it as a superposition of a diagonal matrix $\boldsymbol{D}_{s,k\times k}$ and a low-rank matrix $\boldsymbol{L}_{s,k\times k}$ as

$$\boldsymbol{W}_{s,k\times k} \approx \boldsymbol{D}_{s,k\times k} + \boldsymbol{L}_{s,k\times k} \quad (4)$$

The low-rank matrix $\boldsymbol{L}_{s,k\times k}$ can be represented as a product of two smaller matrices $\boldsymbol{P}_{s,k\times c}$ and $\boldsymbol{Q}_{s,c\times k}$. Hence

$$\boldsymbol{W}_{s,k\times k} \approx \boldsymbol{D}_{s,k\times k} + \boldsymbol{P}_{s,k\times c}\boldsymbol{Q}_{s,c\times k} \quad (5)$$

The number of elements in the LRPD decomposition is $k(2c+1)$, while the original $\boldsymbol{W}_s$ has k^2 elements. If $c \ll k$, this can significantly reduce the adaptation model footprint.

We can see that adaptation in an LRPD form amounts to inserting two linear layers above the layer being adapted, as shown in Fig. 1c. The first layer has c units with weight matrix $\boldsymbol{Q}_s$ and the second one has k units with weight matrix $\boldsymbol{P}_s$. The one-to-one skip-layer connections correspond to the diagonal component $\boldsymbol{D}_s$.

Note that the LRPD adaptation can not only be applied to a full-size DNN, but also be applied to the bottleneck layer of a SVD DNN, leading to a combination of the SVD bottleneck adaptation and LRPD adaptation. This essentially forms a cascade of two low-rank models for different purposes. The SVD decomposition is aimed to restructure the SI DNN model, whereas the LRPD decomposition is to restructure the speaker-specific components.

The LRPD bridges the gap between the full and diagonal transformation matrices. When $c = 0$, the LRPD is reduced to adaptation with diagonal matrix. Specifically, if we apply the diagonal transforms before or after all non-linear layers, we may achieve the sigmoid [14] or LHUC [13] adaptation. On the other hand, if we fix the diagonal components $\boldsymbol{D}_s$ to an identity matrix, we achieve a form of low-rank plus identity (LRPI). Further setting $c = 0$ leads to the adaptation of only bias vectors.

3.3. Training procedure

The parameters of the LRPD adaptation can be estimated using the error backpropagation (BP) algorithm [23]. One issue is how to initialize the LRPD model. The first approach is to apply the SVD decomposition to a well trained full adaptation matrix minus an identity matrix, as we have illustrated in Section 3.1. Then we may keep a portion, either in percentage η or absolute number c, of largest singular values, and retrain the model to recover the accuracy loss. Alternatively, we can simply initialize the model by setting $\boldsymbol{D}_s$ to an identity matrix, bias to zero, and low-rank $\boldsymbol{P}_s$ and $\boldsymbol{Q}_s$ randomly with a uniform distribution. This method requires less training iterations. One limitation is that the low-rank value c is not automatically determined from the characteristic of the adaptation data.

4. EXPERIMENTS AND RESULTS

4.1. Experimental setup

The proposed adaptation method was evaluated using a Microsoft internal Windows Phone short message dictation (SMD) task. The Computational Network Toolkit (CNTK) [24] is used for neural network training. The baseline SI models were trained with 300hr voice search (VS) and SMD data. The experiments were conducted on data from 7 speakers. The total number of test set words is 20,203. There is no overlap among the development and test data.

The SI GMM-HMM acoustic model has approximately 288K Gaussian components and 5976 senones trained with the MLE procedure, followed by fMPE and BMMI. The baseline SI CD-DNN-HMM system takes as input a 22-dimension mean-normalized log-filter bank feature with up to second-order derivatives and a context window of 11 frames, forming a vector of 726-dimension (66×11) input. On top of the input layer there are 5 hidden layers with 2,048 units for each. The output layer has a dimension of 5,976. We convert the full-size DNN model to the SVD DNN model by doing SVD on all the matrices except the one between the input and the first hidden layer, and keep 40% of total singular values. The numbers of units on the linear layers after SVD are 256, 272, 224, 256, and 368, from bottom to top. We then retrained the SVD model and obtained comparable accuracy to the full-size model. The baseline SI full-size DNN and SVD DNN systems achieved 21.48% and 21.43% WER, respectively, on the 7-speaker test set. More details of SVD-based DNN model training can be found in [22].

The DNN models are adapted in a supervised way, where the SI model is used to align the development data. We varied the number of adaptation utterances from 5 (32 seconds) to 100 (11 minutes) for each speaker.

4.2. Adaptation of full-size DNN

The first experiment is conducted to adapt the excitation to the fourth hidden layer of the full-size DNN, as it is shown in [25] that adapting intermediate layer provides more benefits than adapting boundary layers or all hidden layers. The KLD regularization [10] is applied to prevent the adapted model from straying too far from the SI model, since the adaptation data is scarce compared with the scale of the adaptation model. The regularization weight ρ is empirically set and varies with tasks.

Fig. 3 shows the recognition performance (in WER%) for the linear and LRPD adaptation. The standard linear adaption employs a full transformation matrix with $2048 \times 2048 = 4M$ parameters. In contrast, the LRPD adaptation with $c = 10$ reduces the transformation to $2K \times 21 = 42K$ parameters, 10% footprint of the standard linear adaption. If we set $c = 0$, only diagonal and bias components are adapted. This is equivalent to the sigmoid adaptation except that one hidden layer, instead of all 5 layers, is updated.

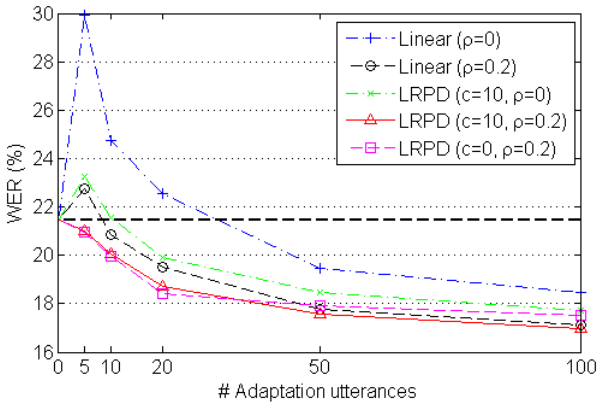


Fig. 3: WERs (%) against the number of adaptation utterances for linear and LRPD adaptation of the full-size DNN model. The dashed level line is the SI DNN baseline.

It is observed that the LRPD ($c = 10,\ \rho = 0.2$) produces 4.1-21.0% relative WER reduction (WERR) with 5-100 adaptation utterances, which outperforms the corresponding linear adaptation (-5.9-20.3% WERR). Second, the regularization weight ρ has an observable impact on the adaptation performance [10], especially for the standard linear adaptation. Without regularization, the linear adaptation shows gains with at least 50 adaptation utterances, and LRPD with 20 utterance. Third, the sigmoid ($c = 0,\ \rho = 0.2$) adaption yields the similar WER to the LRPD ($c = 10,\ \rho = 0.2$) adaptation with 5-20 adaptation utterances, and then become worse when more adaptation data is available. These observations indicate that the LRPD adaption is capable to scale according to the adaptation data, even if the underlying SI mode does not expose an amenable structure for adaptation.

Table 1: Footprints and WERs (in %) for the LRPD adaptation of the SVD DNN model, trained by starting from random initial conditions.

Model	# Adapt. params.	# Adapt. utts. 5	20	100	Avg.
Linear	50.4K	19.80	18.26	16.22	17.69
LRPD, c=0	0.4K	20.92	20.12	18.75	19.69
LRPD, c=5	2.6K	20.26	18.64	17.47	18.46
LRPD, c=10	4.8K	20.09	18.43	16.92	18.18
LRPD, c=20	9.2K	19.89	18.13	16.91	17.97
LRPD, c=50	22.3K	19.37	18.28	16.67	17.83
LRPI, c=0	0.2K	21.01	20.09	19.01	19.80
LRPI, c=20	9.0K	19.94	18.20	16.97	18.01

4.3. Adaptation of SVD DNN

The second experiment was conducted to adapt a bottleneck layer of the SVD DNN models. The goal is to see whether the LRPD model can further improve beyond an already very compact SVD bottleneck adaptation. The third bottleneck layer is chosen for adaptation. Initially, the LRPD is trained by choosing a low-rank value c and randomly initializing the low-rank matrices with a uniform distribution. LRPD with $c = 0$ is equivalent to only adjusting the scaling factor on the SVD bottleneck layer. The regularization weight ρ is set to 0, at which all adaptation methods are found to achieve the best performance.

Table 1 compares the speaker-specific footprints and WERs for the standard bottleneck (denoted as linear) and LRPD bottleneck adaptation of different configurations. WERs are selectively shown at 5, 20, and 100 adaptation utterances, and averaged over 5, 10, 20, 50, and 100 utterances.

First, we observed that the adaptation on the SVD bottleneck layer produces more robust model and obtains improved accuracy over the adaptation on the layer of the full-size DNN. The bottleneck adaptation does not exhibit accuracy degradation, even without imposing the KLD regularization. The LRPD ($c = 20$) bottleneck adaptation decreases WER of the SI model by 7.2-24.3% relative with 5-100 adaptation utterances, which is better than LRPD ($c = 20$) for the full-size DNN with 1.9-20.9% relative. This is 1.6% relatively worse (18.18% vs. 17.69% WER) in average than the standard SVD bottleneck adaptation, while reducing the adaptation footprint by 82%. Second, the LRPD adaptation progressively improves performance as the low-rank value c increases from 0 to 20, and then flattens after 20. Finally, comparing the results of LRPD and LRPI shows that finetuning the diagonal components is not essential, only producing marginal WER improvement.

Table 2: Footprints and WERs (in %) for the LRPD adaptation of the SVD DNN model, trained by decomposing a well trained full adaptation matrix as the seed model.

Singular value ratio	# Adapt. params.	# Adapt. utts. 5	20	100	Avg.
$\eta = 0.1$	3.9K	19.53	18.34	17.19	18.11
$\eta = 0.2$	5.9K	19.53	18.43	16.97	17.97
$\eta = 0.3$	8.9K	19.56	18.25	16.75	17.85
$\eta = 0.4$	12.3K	19.52	18.10	16.64	17.75

The LRPD adaptation in the previous experiments, which starts from a random initial condition with the same low-rank value for different amounts of adaptation data, may not be the best way to generate the model. In the following experiment, we generate the LRPD model by decomposing a well trained full adaptation matrix as described in Section 3.3. The same third bottleneck layer is adapted. Taking the adapted bottleneck model in line 1, Table 1 as the seed model, we keep a portion (η) of the total of the singular values and then finetune the LRPD model. Since we start from a decent seed model, a relatively small learning rate is used in the model finetune stage. Results are summarized in Table 2. It is shown that the model keeping 30% of total singular values only incurs 1% relative WER increase (17.85% vs. 17.69% WER) in average than the standard bottleneck adaptation, while reducing the speaker-specific footprint by 82% (8.9K vs. 50.4K parameters).

5. CONCLUSION

In this paper, we presented a scalable adaptation technique that adapts the DNN model through the LRPD decomposition. The proposal is inspired by observing that adaptation matrices are very close to an identity matrix or diagonally dominant. The LRPD restructures the adaptation matrix as a superposition of a diagonal matrix and a low-rank matrix. By varying the low-rank values, the LPRD contains the full and the diagonal adaptation matrix as its special cases. Thus it is capable to yield a compact model that makes full use of the amount of available development data. Experimental results demonstrated that the LRPD adaptation of the full-size DNN obtains improved accuracy over the standard liner adaptation. The LRPD bottleneck adaptation can reduce the speaker-specific footprint by 82% over an already very compact SVD bottleneck adaptation, at an expense of 1% relative WER increase.

6. REFERENCES

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

[2] N. Jaitly, P. Nguyen, A. Senior, and V. Vanhoucke, "Application of pretrained deep neural networks to large vocabulary speech recognition," in *Proc. Interspeech*, 2012, pp. 2578–2581.

[3] L. Deng, J. Li, J.-T. Huang, et al., "Recent advances in deep learning for speech research at Microsoft," in *Proc. ICASSP*, 2013, pp. 8604–8608.

[4] 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. ASRU*, 2011, pp. 30–35.

[5] 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.

[6] 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*, 2010, pp. 526–529.

[7] F. Seide, G. Li, X. Chen, and D. Yu, "Feature engineering in context-dependent deep neural networks for conversational speech transcription," in *Proc. Workshop on Automatic Speech Recognition and Understanding*, 2011, pp. 24–29.

[8] 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, pp. 827–835, 2007.

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

[10] 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.

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

[12] 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, pp. 1–5.

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

[14] Y. Zhao, J. Li, J. Xue, and Y. Gong, "Investigating online low-footprint speaker adaptation using generalized linear regression and click-through data," in *Proc. ICASSP*, 2015, pp. 4310–4314.

[15] G. Saon, H. Soltau, D. Nahamoo, and M. Picheny, "Speaker adaptation of neural network acoustic models using i-vectors," in *Proc. Workshop on Automatic Speech Recognition and Understanding*, 2013, pp. 55–59.

[16] A. Senior and I. Lopez-Moreno, "Improving DNN speaker independence with i-vector inputs," in *Proc. ICASSP*, 2014, pp. 225–229.

[17] Y. Miao, H. Zhang, and F. Metze, "Towards speaker adaptive training of deep neural network acoustic models," in *Proc. Interspeech*, 2014.

[18] 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.

[19] O. Abdel-Hamid and H. Jiang, "Direct adaptation of hybrid DNN/HMM model for fast speaker adaptation in LVCSR based on speaker code," in *Proc. Interspeech*, 2014.

[20] 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," *Audio, Speech, and Language Processing, IEEE/ACM Transactions on*, vol. 22, no. 12, pp. 1713–1725, 2014.

[21] D. Yu and L. Deng, "Adaptation of deep neural networks," in *Automatic speech recognition: a deep learning approach*, pp. 193–215. Springer, 2015.

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

[23] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," in *Cognitive modeling*. MIT Press, Cambridge, MA, 1988.

[24] A. Agarwal, E. Akchurin, C. Basoglu, et al., "An introduction to computational networks and the computational network toolkit," Tech. Rep. MSR-TR-2014-112, Microsoft, 2014.

[25] K. Kumar, C. Liu, K. Yao, and Y. Gong, "Intermediate-layer DNN adaptation for offline and session-based iterative speaker adaptation," in *Proc. Interspeech*, 2015, pp. 1091–1095.