INTERSPEECH 2017
August 20–24, 2017, Stockholm, Sweden

Deep Least Squares Regression for Speaker Adaptation

Younggwan Kim, Hyungjun Lim, Jahyun Goo, Hoirin Kim

School of Electrical Engineering, KAIST, Daejeon, Republic of Korea
{cleanthink, hyungjun.lim, jahyun.goo, hoirkim}@kaist.ac.kr

Abstract

Recently, speaker adaptation methods in deep neural networks (DNNs) have been widely studied for automatic speech recognition. However, almost all adaptation methods for DNNs have to consider various heuristic conditions such as mini-batch sizes, learning rate scheduling, stopping criteria, and initialization conditions because of the inherent property of a stochastic gradient descent (SGD)-based training process. Unfortunately, those heuristic conditions are hard to be properly tuned. To alleviate those difficulties, in this paper, we propose a least squares regression-based speaker adaptation method in a DNN framework utilizing posterior mean of each class. Also, we show how the proposed method can provide a unique solution which is quite easy and fast to calculate without SGD. The proposed method was evaluated in the TED-LIUM corpus. Experimental results showed that the proposed method achieved up to a 4.6% relative improvement against a speaker independent DNN. In addition, we report further performance improvement of the proposed method with speaker-adapted features.

Index Terms: deep neural network, speaker adaptation, class-dependent posterior mean, deep least squares regression

1. Introduction

Deep neural networks (DNNs) have become an essential tool for modeling acoustic models (AMs) and shown dramatic gains in various automatic speech recognition (ASR) tasks [1–3]. Despite the progress, DNN-based AMs still suffer from the mismatches between the training and testing conditions, which may cause performance degradations. To overcome the mismatches, various adaptation techniques have been widely studied and speaker adaptation is one of the adaptation techniques to minimize the mismatch between the training and testing conditions caused by speaker variability.

To adapt Gaussian mixture model-hidden Markov model (GMM-HMM)-based AMs, maximum a posteriori (MAP), maximum likelihood linear regression (MLLR), and eigenvoice approach are commonly used [4–7]. The main advantage of these approaches is that they can provide closed form solutions for model parameter adaptation. Thus, when we apply those methods, we need to consider only a few hyper parameters.

As mentioned earlier, deep neural network-hidden Markov model (DNN-HMM)-based acoustic models has been widely used in these days. Since, however, adaptation methods developed for GMM-HMMs cannot be easily applied to DNN-HMMs, speaker adaptation for DNN-HMMs has been actively explored by many researchers. Actually, speaker adaptation methods for DNN-HMMs can be categorized into 3 classes: speaker-adapted layer insertion, conservative training, and auxiliary feature augmentation.

For speaker-adapted layer insertion (SALI), adding linear transformation (LT) layer is the simplest and the most popular way to adapt DNN-HMMs [8, 9]. For this method, a single LT layer is inserted between any two layers of a speaker independent DNN. After the insertion, only the LT layer is trained by adaptation data. Finally, each speaker can obtain speaker-adapted LT layer used for a decoding process. Learning hidden unit contribution (LHUC) is another type of SALI and inserts different type of speaker-adapted layer which can control the output of hidden layers in non-linear way [10].

Conservative training (CT) is conducted by adding a regularization term to the cross entropy (CE) criterion. L_2 or L_1 weight decay and Kullback-Leibler divergence (KLD) regularization are well known for the CT-based speaker adaptation methods [11, 12]. To apply CT-based speaker adaptation more simply, we can just train several layers of the speaker-independent DNN with or without regularization and assign those layers to each speaker for the decoding process.

For auxiliary feature augmentation (AFA), acoustic features are concatenated with speaker-specific vector information such as i-vectors and bottleneck features obtained from a DNN for speaker classification. The most representative way of AFA is to concatenate the acoustic features with speaker-dependent i-vectors [13–15].

In fact, almost all of adaptation techniques for DNN-HMMs are optimized by the CE criterion. Stochastic gradient descent (SGD) is the most popular method to minimize the CE criterion for DNN training process. However, SGD essentially needs to consider various heuristic conditions such as mini-batch sizes, learning rate scheduling, stopping criteria, and initialization conditions. In general, those heuristic conditions are not easy to be properly tuned and the tuning process may require many trial and errors. To alleviate those difficulties, in this paper, we propose a new speaker adaptation technique for DNN-HMMs, which is called deep least squares regression (DLSR). The motivation of the proposed method came from MLLR. Thus, different from other adaptation methods, DLSR can be optimized by a closed form solution and naturally does not require SGD. In this paper, we also address how the proposed method can provide the closed form solution for DNN-HMM adaptation, which is easy and fast to obtain. The proposed method was evaluated in the TED-LIUM release-1 corpus.

The rest of the paper is organized as follows. Section 2 describes maximum likelihood linear regression for GMM-HMM acoustic model adaptation from which the motivation of the proposed method came. In Section 3, we briefly review DNN-based acoustic models. In Section 4, deep least squares regression with the posterior mean and the derivation of the closed form solution are represented. In Section 5, we give the details of our experimental setup and results. We conclude our work in Section 6.

http://dx.doi.org/10.21437/Interspeech.2017-783

2. Maximum likelihood linear regression

MLLR is the most well-known speaker adaptation method for GMM-HMM [5, 6]. MLLR computes a set of transformations that will reduce the mismatch between a speaker-independent model and the speaker-specific speech data. More specifically, MLLR estimates a set of affine transformation matrices for the means and variances of GMM-HMM. It generally uses regression classes that classified the HMM states to give different affine transforms for each states. To briefly review MLLR, at first, we define a GMM-based distribution on a HMM state s, which is given by

$$b_s(\mathbf{x}) = \sum_{g=1}^{G} w_g \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_{s,g}, \boldsymbol{\Sigma}_{s,g}) \tag{1}$$

where $\mathbf{x}$ is an input feature vector, G is the total number of Gaussian components, w_g is the mixture weight of the g^{th} Gaussian, and $\mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_{s,g}, \boldsymbol{\Sigma}_{s,g})$ is the normal distribution with mean $\boldsymbol{\mu}_{s,g}$ and covariance $\boldsymbol{\Sigma}_{s,g}$. For the simplest way of MLLR-based speaker adaptation, the mean vectors of the all Gaussian components are transformed by a global transformation matrix $\mathbf{W}_k \in \mathbb{R}^{d\times(d+1)}$ for speaker k. Thus, the transformed mean vector is given by

$$\boldsymbol{\mu}_{s,g}^{k} = \mathbf{W}_k \hat{\boldsymbol{\mu}}_{s.g} \tag{2}$$

where $\hat{\boldsymbol{\mu}}_{s.g}$ is the augmented mean vector of $\boldsymbol{\mu}_{s,g}$, which is given by

$$\hat{\boldsymbol{\mu}}_{s,g} = \begin{bmatrix} \boldsymbol{\mu}_{s,g} \\ 1 \end{bmatrix}. \tag{3}$$

$\mathbf{W}_k$ can be obtained by maximizing the following auxiliary function:

$$Q(\mathbf{W}_k) \sim -\frac{1}{2}\sum_{n,s,g} \gamma_{s,g}(\mathbf{x}_n)[(\mathbf{x}_n - \mathbf{W}_k\hat{\boldsymbol{\mu}}_{s,g})^{\mathrm{T}} \boldsymbol{\Sigma}_{s,g}^{-1}(\mathbf{x}_n - \mathbf{W}_k\hat{\boldsymbol{\mu}}_{s,g})] \tag{4}$$

where $\gamma_{s,g}(\mathbf{x}_n)$ is *a posterior* probability that $\mathbf{x}_n$ is occupied by state s and the g^{th} Gaussian component. By setting $\partial Q(\mathbf{W}_k)/\partial \mathbf{W}_k = 0$, we can reach the following closed form solution:

$$\hat{\mathbf{W}}_k = \left(\sum_{n,s,g} \gamma_{s,g}(\mathbf{x}_n)\boldsymbol{\Sigma}_{s,g}^{-1}\mathbf{x}_n\hat{\boldsymbol{\mu}}_{s,g}^{\mathrm{T}}\right)\left(\sum_{n,s,g} \gamma_{s,g}(\mathbf{x}_n)\boldsymbol{\Sigma}_{s,g}^{-1}\hat{\boldsymbol{\mu}}_{s,g}\hat{\boldsymbol{\mu}}_{s,g}^{\mathrm{T}}\right)^{-1}. \tag{5}$$

As shown in (5), the main advantage of MLLR is that we can obtain the closed form solution without any gradient-based search algorithms.

3. Deep neural network-based acoustic models

In this section, we briefly review the DNN-based acoustic models trained by the CE criterion. First of all, the output of each hidden layer is given by

$$\left.\begin{aligned} \mathbf{h}_n^l &= \sigma(\mathbf{z}_n^l) \\ \mathbf{z}_n^l &= \mathbf{W}^l\mathbf{h}_n^{l-1} + \mathbf{b}^l \end{aligned}\right\}, \quad \text{for } 1 \le l \le L \tag{6}$$

where n and l are frame and layer index, respectively, $\mathbf{W}^l$ and $\mathbf{b}^l$ are weight matrix and bias vector for affine transformation, $\sigma(\cdot)$ denotes an activation function (sigmoid function in this paper), and L is the total number of the hidden layers. For $l = 1$, $\mathbf{h}_n^0 = \mathbf{x}_n$ where $\mathbf{x}_n$ is an input observation which typically consists of 9-15 frames of feature vectors. Next, the output layer located above the L^{th} hidden layer typically utilizes the affine transformation-based softmax function. More specifically, the output of the i^{th} output node is given by

$$o_n^i = p(i \mid \mathbf{h}_n^L) = \frac{\exp(d_n^i)}{\sum_{j=1}^{O}\exp(d_n^j)} \tag{7}$$

$$d_n^i = \mathbf{w}_i^T\mathbf{h}_n^L + b_i \tag{8}$$

where d_n^i is the conventional linear output excitation, $\mathbf{w}_i$ and b_i are the weight vector and the bias for node i, respectively and O is the total number of nodes in the output layer. Typically, each output node indicates a hidden Markov model (HMM) tied triphone state. The CE criterion is typically used for DNN training and is given by

$$\mathcal{L}_{CE}(\boldsymbol{\theta}) = -\sum_{n=1}^{N}\sum_{i=1}^{O} I(i = c_n)\log p(i \mid \mathbf{h}_n^L) \tag{9}$$

where $\boldsymbol{\theta}$ represents the all set of the DNN parameters, N is the total number of training data, and the indicator function $I(i = c_n)$ has 1 if i is equal to class information c_n and 0 for otherwise. To train DNN parameters based on (9), SGD update rule is commonly used and given by

$$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \eta\frac{\partial \mathcal{L}_{CE}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \tag{10}$$

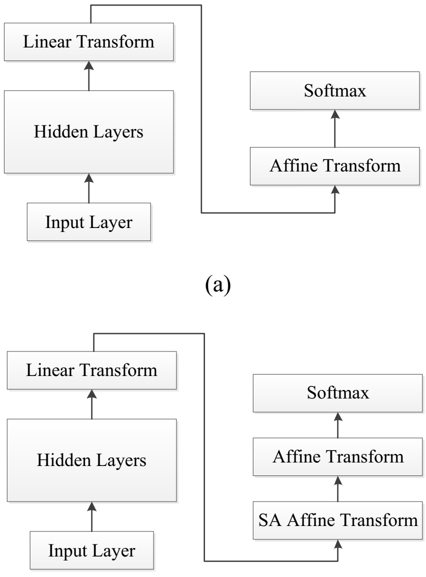
where η is a learning rate.

4. Proposed method

Most of DNN-HMM-based speaker adaptation methods add certain layer-wise components into speaker-independent DNNs, certain regularization terms into the CE criterion or both of them. When minimizing the CE criterion for speaker adaptation, SGD is an essential search algorithm. However, SGD may require several heuristic conditions such as mini-batch sizes, learning rate scheduling, stopping criteria, and initialization conditions. In general, those heuristic conditions are hard to find optimal values. To alleviate those difficulties, in this paper, we propose deep least squares regression for speaker adaptation in a DNN framework utilizing class-dependent posterior means obtained by the last hidden layer output. From the proposed method, we can obtain speaker-adapted transformation matrix from a closed form equation without SGD.

4.1. Class-dependent posterior mean

As mentioned in Section 2, MLLR was developed in a generative model framework. However, since DNN is a discriminative model constituting decision boundaries, there is no way to represent data distribution with DNN. Thus, in this section, we introduce a class-dependent posterior mean which can represent class-dependent center. To do so, we slightly modify the DNN introduced in previous section by adding an



(a)

(b)

Figure 1: *Modified DNN architecture. (a) speaker-independent DNN and (b) DNN with a speaker-adapted transformation matrix*

additional linear transformation above the last hidden layer and the transformed last hidden layer output is given by

$$\tilde{\mathbf{h}}_n^L = \mathbf{W}_{LT}\mathbf{h}_n^L \tag{11}$$

where $\mathbf{W}_{LT}$ is the linear transformation matrix. The modified DNN architecture is shown in Figure 1 (a). From the modified DNN architecture, we can replace almost binary representation with unbounded representation. By using $\tilde{\mathbf{h}}_n^L$, the class-dependent posterior mean on i^{th} node can be calculated by

$$\boldsymbol{\mu}_i^{DNN} = \frac{\sum_{n=1}^{N_i} p(i\,|\,\tilde{\mathbf{h}}_n^L)\tilde{\mathbf{h}}_n^L}{\sum_{n=1}^{N_i} p(i\,|\,\tilde{\mathbf{h}}_n^L)} \tag{12}$$

where N_{i} is the total number of training data for class *i*.

4.2. Deep least squares regression

Least squares regression-based speaker adaptation is shown in [5, 16] for a template-based speech recognition system. By using the class-dependent posterior mean, least squares regression can be applied to DNN adaptation. However, in this paper, we transform $\tilde{\mathbf{h}}_n^L$ instead of $\boldsymbol{\mu}_i^{DNN}$. Actually, this approach is much more similar to feature-level MLLR [17]. The proposed learning criterion with the class-dependent posterior mean is given by

$$\mathcal{L}_{DLSR}(\mathbf{W}_k^{DNN}) = \frac{1}{2}\sum_{n=1}^{N_k} \| \mathbf{W}_k^{DNN}\hat{\mathbf{h}}_n^L - \boldsymbol{\mu}_{c_n}^{DNN} \|_2^2 \tag{13}$$

where N_k is the total number of adaptation data for speaker *k* and $\hat{\mathbf{h}}_n^L$ is the augmented vector of $\tilde{\mathbf{h}}_n^L$. The differential of (13) with $\mathbf{W}_k^{DNN}$ is given by

$$\frac{\partial \mathcal{L}_{DLSR}(\mathbf{W}_k^{DNN})}{\partial \mathbf{W}_k^{DNN}} = \sum_{n=1}^{N_k} (\mathbf{W}_k^{DNN}\hat{\mathbf{h}}_n^L - \boldsymbol{\mu}_{c_n}^{DNN})(\hat{\mathbf{h}}_n^L)^{\text{T}}. \tag{14}$$

By setting $\partial \mathcal{L}_{DLSR}(\mathbf{W}_k^{DNN}) / \partial \mathbf{W}_k^{DNN} = 0$, we can reach a closed form solution for the speaker-adapted transformation matrix $\mathbf{W}_k^{DNN}$, which is given by

$$\tilde{\mathbf{W}}_k^{DNN} = \left(\sum_{n=1}^{N_k} \boldsymbol{\mu}_{c_n}^{DNN} (\hat{\mathbf{h}}_n^L)^{\text{T}} \right) \left(\sum_{n=1}^{N_k} \hat{\mathbf{h}}_n^L (\hat{\mathbf{h}}_n^L)^{\text{T}} \right)^{-1}. \tag{15}$$

In addition, we also consider a diagonalized form of $\tilde{\mathbf{W}}_k^{DNN}$ which is given by

$$\tilde{\mathbf{W}}_k^{DNN,diag} = \begin{bmatrix} \tilde{w}_{1,1}^{DNN} & 0 & \cdots & 0 & \tilde{w}_{1,d+1}^{DNN} \\ 0 & \tilde{w}_{2,2}^{DNN} & \ddots & \vdots & \tilde{w}_{2,d+1}^{DNN} \\ \vdots & \ddots & \ddots & 0 & \vdots \\ 0 & \cdots & 0 & \tilde{w}_{d,d}^{DNN} & \tilde{w}_{d,d+1}^{DNN} \end{bmatrix} \tag{16}$$

where $\tilde{w}_{i,j,k}^{DNN}$ is the element at i^{th} row and j^{th} column of $\tilde{\mathbf{W}}_k^{DNN}$. Finally, we can obtain the speaker-adapted transformation matrix after applying an essential regularization step which is given by $\hat{\mathbf{W}}_k^{DNN} = \lambda\tilde{\mathbf{W}}_k^{DNN} + (1-\lambda)\tilde{\mathbf{I}}$ where λ is a regularization parameter and $\tilde{\mathbf{I}}$ is an augmented identity matrix composed of an identity matrix and a zero vector, $[\mathbf{I}\ \mathbf{0}]$. After obtaining the final speaker-adapted transformation matrix, we can apply the matrix as shown in Figure 1 (b).

5. Experimental results

To evaluate the proposed method, we conducted additional experiments on TED-LIUM corpus [18]. This corpus contains 774 TED talks which are 118 hours of speech data and this corpus was used for training. For test, other 8 TED talks (dev 2010) that amount to 1.7 hours were used, which are not included in the aforementioned TED-LIUM corpus. 50% of these talks were used for supervised adaptation and the other 50% is used for evaluation. For decoding, we used 150K word vocabulary and a pruned "Cantab" trigram language model [19].

The speech was analyzed using a 25-ms Hamming window with a 10-ms fixed frame rate. We used two types of feature vectors for DNN input. The first one was a speaker-independent Fourier-transform-based log filterbank energy with 40 coefficients distributed on a Mel-scale, together with their first and second temporal derivatives. The second one was based on 12th-order Mel frequency cepstral coefficients (MFCCs) and energy with their first and second temporal derivatives. Then, 7 frames of the 39 dimensional MFCCs were concatenated and projected down to 40 dimensions through linear discriminant analysis (LDA). After that, the projected results were finally transformed by feature space maximum likelihood linear regression (fMLLR) for each speaker.

We used the open-source Kaldi toolkit for all experiments [20]. The Gaussian mixture model (GMM) systems were built by the TEDLIUM recipe that has been released together with Kaldi. Our GMM model has 3986 context dependent triphone

Table 1: *ASR performance (word error rate in %) with the filterbank energy features.*

Models	WER
SI DNN-2048	17.5%
SI DNN-1024	17.6%
SI DNN-512	17.7%
DLSR-2048 (Diag)	17.1%
DLSR-2048 (Full)	**16.8%**
DLSR-1024 (Diag)	17.1%
DLSR-1024 (Full)	16.9%
DLSR-512 (Diag)	17.2%
DLSR-512 (Full)	16.9%

Table 2: *ASR performance (word error rate in %) with the MFCC+LDA+fMLLR.*

Models	WER
SI DNN-2048	17.0%
SI DNN-1024	17.0%
SI DNN-512	17.1%
DLSR-2048 (Diag)	16.8%
DLSR-2048 (Full)	**16.6%**
DLSR-1024 (Diag)	16.9%
DLSR-1024 (Full)	**16.6%**
DLSR-512 (Diag)	16.8%
DLSR-512 (Full)	16.7%

states. The phonetic label on each speech frame is generated by the GMM model through forced alignment. The baseline DNN model was also trained by Kaldi recipe. The DNN had 6 hidden layers, each of which has 2048 nodes and sigmoid function for non-linear activation. The last softmax output layer has 3986 nodes corresponding to the context dependent triphone states. The inputs are 11 frames of features (5 frames on each side of current frame) with global mean variance normalization. The DNN parameters are initialized with restricted Boltzmann machines (RBMs) [21]. For fine-tuning, we optimized the CE criterion with the exponentially decaying “newbob” learning rate schedule with an initial learning rate 0.008. For the modified DNN configuration, we considered various sizes of the output dimension of $\mathbf{W}_{LT}$. The regularization parameter λ was set to 0.1.

At first, recognition performance with filterbank energy feature in word error rate (WER) is reported in Table 1. For every test condition, there are three numbers which are 2048, 1024, and 512. Those numbers indicate the output dimension of $\mathbf{W}_{LT}$. SI DNN denotes the speaker-independent DNN. For DLSR, “Diag” and “Full” mean $\hat{\mathbf{W}}_k^{DNN,diag}$ and $\hat{\mathbf{W}}_k^{DNN}$, respectively. From this experiment, we found that the proposed adaptation method can increase the SI DNN recognition performance even though DLSR is a type of generative approaches. In addition, DLSR-2048 (Full) shows the best performance whose relative improvement is 4.6% against SI DNN-1024. In particular, note that mild bottleneck conditions reducing the number of speaker-specific model parameters do not affect the recognition accuracy same as the results reported in [22].

We additionally performed the recognition test with the speaker-adapted MFCC features which were followed by LDA and fMLLR. Table 2 reports the WER results according to the MFCC features. Basically, the proposed adaptation method with filterbank energy features outperforms SI-DNN with the speaker-adapted MFCC features. Combined with the MFCC features, DLSR still shows further improvements.

6. Conclusions

In this paper, we proposed a new affine transform-based speaker adaptation method for DNN-HMM acoustic models. For the method, we introduced a modified DNN architecture and class-dependent posterior mean. By the proposed method, it was shown how the DLSR can provide the closed form solution. In our experiments, the proposed adaptation method improved the recognition accuracy of SI DNN. With filterbank energy features, we achieved relative improvements of 4.6% against SI DNN. Furthermore, it was also observed that the mild bottleneck conditions on the $\mathbf{W}_{LT}$ output do not degrade the recognition performance. In our future work, we will apply the proposed adaptation method with various speaker adaptation methods such as LHUC and i-vector augmentation.

7. Acknowledgements

This work was partly supported by the ICT R&D program of MSIP/IITP [R7119-16-1001, Core technology development of the real-time simultaneous speech translation based on knowledge enhancement] and the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIP) (No. 2014R1A2A2A01007650).

8. References

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

[2] A. Mohamed, G. Dahl, and G. Hinton, “Acoustic modeling using deep belief networks,” *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 20, no. 1, pp. 14–22, 2012.

[3] G. Hinton, L. Deng, D. Yu, G. Dahl, A.-R. Mohamed, N. Jaitly, V. Vanhoucke, P. Nguyen, T. Sainath, and B. Kingsbury, “Deep Neural Networks for Acoustic Modeling in Speech Recognition,” *IEEE Signal Process. Mag.*, vol. 29, no. 6, pp. 82–97, 2012.

[4] J. Gauvain and C. 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.

[5] C. Leggetter and P. 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.

[6] M. Gales, “Maximum likelihood linear transformations for hmm-based speech recognition,” *Computer speech & language*, vol. 12, no. 2, pp. 75–98, 1998.

[7] R. Kuhn, J.-C. Junqua, P. Nguyen, and N. Niedzielski, "Rapid speaker adaptation in eigenvoice space", *IEEE Trans. Speech Audio Process.*, vol. 8, no. 6, pp. 695-707, Nov. 2000.

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

[9] J. Trmal, J. Zelinka, and L. Müller, “Adaptation of a feedforward artificial neural network using a linear transform,” in *Text, Speech and Dialogue*, Springer, 2010, pp. 423–430.

[10] P. Swietojanski and S. Renals, “Learning hidden unit contributions for unsupervised speaker adaptation of neural network acoustic models,” in *SLT,* 2014, pp. 171–176.

[11] 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. IEEE Acoust., Speech, Signal Process. (ICASSP)*, 2013, pp. 7893–7897.

[12] Y. Huang and Y. Gong, “Regularized sequence-level deep neural network model adaptation,” in *Proc. Interspeech*, 2015.

[13] V. Gupta, P. Kenny, P. Ouellet, and T. Stafylakis, “I-vector-based speaker adaptation of deep neural networks for french broadcast audio transcription,” in *Proc. IEEE Acoust., Speech, Signal Process. (ICASSP)*, 2014, pp. 6334–6338.

[14] G. Saon, H. Soltau, D. Nahamoo, and M. Picheny, “Speaker adaptation of neural network acoustic models using i-vectors, in *Proc. IEEE Workshop Autom. Speech Recogn. Understand. (ASRU)*, 2013, pp. 55–59.

[15] A. Senior and I. Lopez-Moreno, “Improving DNN speaker independence with I-vector inputs, in *Proc. IEEE Acoust., Speech, Signal Process. (ICASSP)*, 2014.

[16] A. J. Hewett, “Training and speaker adaptation in template-based speech recognition,” Ph.D. dissertation, Cambridge Univ., Cambridge, U.K., 1989.

[17] D. Povey and G. Saon, “Feature and model space speaker adaptation with full covariance Gaussians,” in *Proc. Annu. Conf. Int. Speech Communication Association (INTERSPEECH)*, 2006, pp. 1145–1148.

[18] A. Rousseau, P. Deléglise, and Y. Estève, “TED-LIUM: An automatic speech recognition dedicated corpus,” in *Proc. LREC*, pp. 125–129, 2012.

[19] W. Williams, N. Prasad, D. Mrva, T. Ash, and T. Robinson, “Scaling recurrent neural network language models,” in *Proc. Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, 2015, pp. 5391–5395.

[20] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, J. Silovsky, G. Stemmer, and K. Vesely, “The Kaldi speech recognition toolkit,” in *Proc. IEEE Workshop Autom. Speech Recogn. Understand. (ASRU)*, 2011, pp. 1–4.

[21] G. E. Hinton, “A practical guide to training restricted Boltzmann machines,” in *Neural Networks: Tricks of the Trade*. Springer, 2012, pp. 599–619.

[22] T. 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 *Proc. of ICASSP*, 2013.