LEARNING EFFECTIVE FACTORIZED HIDDEN LAYER BASES USING STUDENT-TEACHER TRAINING FOR LSTM ACOUSTIC MODEL ADAPTATION

Lahiru Samarakoon[◇⋆1] *Brian Mak*[◇] *Khe Chai Sim*[†]

[◇] Hong Kong University of Science and Technology
[⋆] Fano Labs, Hong Kong
[†] Google, Inc

lahiru@fano.ai, mak@cse.ust.hk, khechai@google.com

ABSTRACT

Factorized Hidden Layer (FHL) has been proposed for the adaptation of deep neural network (DNN) and Long Short-Term Memory (LSTM) based acoustic models (AMs). In FHL, a speaker-dependent (SD) transformation matrix and an SD bias are included in addition to the standard affine transformation. The SD transformation is a linear combination of rank-1 matrices whereas the SD bias is a linear combination of vectors. However, the adaptation of LSTMs is challenging and often reports modest gains. In this paper, we propose to use student-teacher training to estimate more efficient FHL bases for LSTM AMs using an FHL adapted DNN as the teacher model. For both AMI IHM and AMI SDM tasks, FHL achieves 3.2% absolute improvement over the frame-level cross entropy trained LSTM baselines. Moreover, FHL results 3.0% and 3.8% absolute improvements over sequentially trained LSTM baselines for the AMI IHM and AMI SDM tasks respectively.

***Index Terms*—** Long Short-Term memory (LSTM), Recurrent Neural Networks (RNNs), Speaker Adaptation, Student-teacher training, Acoustic Modeling

1. INTRODUCTION

In state-of-the-art automatic speech recognition (ASR) systems, recurrent neural networks (RNNs) have been found to significantly outperform the feedforward deep neural networks (DNNs) due to better modeling of temporal dependencies. Both RNNs and DNNs suffer from performance degradations due to mismatch between training and testing conditions. To address this problem, adaptation techniques are developed. These techniques reduce the mismatch between training and testing conditions by transforming the models and / or features.

The commonly used maximum a posteriori (MAP) adaptation [1], maximum likelihood linear regression (MLLR) [2, 3] and speaker adaptive training (SAT) [4, 5] were first developed for conventional Gaussian mixture model (GMM)–hidden Markov model (HMM) systems. Then, adaptation techniques were developed for deep neural network (DNN)-HMM hybrid systems with significant performance improvements [6, 7, 8, 9, 10, 11, 12]. Since RNNs consistently outperform DNNs, it is important to develop adaptation methods for RNN acoustic models (AMs). However, unsupervised adaptation of RNN AMs has been recognized as a difficult problem with modest gains reported in the literature [13, 14, 15]. This can be mainly due to the increased complexity of RNNs in comparison to DNNs. It has also been suggested that RNNs perform implicit normalization of the speaker variability due to their effectiveness at capturing and normalizing long-range characteristics and consequently adaptation has a limited impact [15].

Recently, student-teacher training which is also known as knowledge distillation has been used to transfer knowledge between models [16, 17, 18]. Student-teacher training is performed using two steps. First, teacher models are trained and second, student models are trained to mimic output distributions of the teacher models. In [19], student-teacher training is used to build multilingual systems in low-resource settings. In addition, that work shows student models can achieve comparable recognition accuracy to teacher networks. Moreover, student-teacher training is used to avoid overfitting when the model is adapted with a limited amount of data to different domains [20]. Furthermore, student-teacher paradigm is successfully used for speech enhancement [21]. In [21], a teacher model is trained with enhanced features while a student model learns to perform speech enhancement implicitly by mimicking the teacher's output distribution.

In this paper, we propose to employ the student-teacher paradigm to improve the factorized hidden layer (FHL) adaptation of LSTM AMs. FHL adaptation is first proposed to adapt DNNs and has shown superior performance over other adaptation methods [22]. In FHL adaptation, a speaker-dependent (SD) transformation matrix and an SD bias are estimated in addition to the standard affine transformation. The SD transformation is a linear combination of rank-1 matrices whereas the SD bias is a linear combination of vectors. In [13], the effectiveness of FHL is investigated for LSTM AMs. Even though FHL is enjoying significant improvements when used for DNNs [22], gains are modest for LSTMs [13]. Therefore, we claim that it is more difficult to estimate effective FHL bases for LSTMs than DNNs. Based on these findings, we propose to use an FHL adapted DNN as a teacher when estimating FHL bases for LSTM AMs. We have evaluated our approach in two benchmark ASR tasks from the Augmented Multi-party Interaction (AMI) [23]: individual headset microphone (IHM) and the AMI single distant microphone (SDM) tasks, respectively. Results are reported for both frame-wise and sequentially trained systems.

The rest of the paper is organized as follows. Section 2 reviews the LSTM acoustic models and Section 3 discusses the FHL adaptation for LSTMs. Section 4 briefly describes the student-teacher training and details of its usage in this paper. In Section 5 we give the details of our experimental setup. The results are reported in Section 6 and we conclude our work in Section 7.

[1] Work performed at Hong Kong University of Science and Technology.

2. LSTM-RNNS

To mitigate the vanishing gradient problem in RNN training when using stochastic gradient descent method LSTM is proposed [24]. LSTM has memory blocks with self-connections which enable it to model temporal dependencies. The information flow to each LSTM memory cell is controled by a set of units called gates. There are three types of gates called input, output and forget. As the names suggest, the input gate controls the inflow to the memory while an output gate controls the outflow. Forget gates decide how much information to forget during each time step [25]. In some architectures, peephole connections are used to connect gates and cell state information [26]. For ASR, it is more effective to use LSTMP models where a projection layer is used to reduce the network complexity [27]. In this paper, we perform adaptation experiments on LSTMP AMs. A summary of LSTMP formulas are given below:

$$\mathbf{i}_t = \sigma(\mathbf{W}_{xi}\mathbf{x}_t + \mathbf{W}_{ri}\mathbf{r}_{t-1} + \mathbf{W}_{ci}\mathbf{c}_{t-1} + \mathbf{b}_i) \tag{1}$$
$$\mathbf{f}_t = \sigma(\mathbf{W}_{xf}\mathbf{x}_t + \mathbf{W}_{rf}\mathbf{r}_{t-1} + \mathbf{W}_{cf}\mathbf{c}_{t-1} + \mathbf{b}_f) \tag{2}$$
$$\mathbf{o}_t = \sigma(\mathbf{W}_{xo}\mathbf{x}_t + \mathbf{W}_{ro}\mathbf{r}_{t-1} + \mathbf{W}_{co}\mathbf{c}_{t-1} + \mathbf{b}_o) \tag{3}$$
$$\mathbf{c}_t = \mathbf{f}_t \circ \mathbf{c}_{t-1} + \mathbf{i}_t \circ \tanh(\mathbf{W}_{xc}\mathbf{x}_t + \mathbf{W}_{rc}\mathbf{r}_{t-1} + \mathbf{b}_c) \tag{4}$$
$$\mathbf{m}_t = \mathbf{o}_t \circ \tanh(\mathbf{c}_t) \tag{5}$$
$$\mathbf{r}_t = \mathbf{W}_{mr}\mathbf{m}_t \tag{6}$$

where t is the timestep, σ is the sigmoid funtion, $\mathbf{i}_t, \mathbf{f}_t, \mathbf{o}_t, \mathbf{c}_t, \mathbf{m}_t, \mathbf{r}_t$ are vectors with input gate, forget gate, output gate, cell state, cell output, and projection values respectively. $\mathbf{W}_{**}$ are weight matrices and $\mathbf{b}_*$ are biases. All peephole weight matrices $\mathbf{W}_{c*}$ are diagonal.

3. FHL ADAPTATION

In this section, we first review the FHL adaptation for DNNs. Then, FHL adaptation for LSTMP AMs is presented.

3.1. FHL Adaptation for DNNs

FHL adaptation can be used to estimate an SD weight matrix $\mathbf{W}^s$ and an SD bias $\mathbf{b}^s$ as given below:

$$\mathbf{W}^s = \mathbf{W} + \sum_{i=1}^{|\mathbf{d}^s|} \mathbf{d}^s(i)\mathbf{B}(i) \tag{7}$$

where $\{\mathbf{B}(1), \mathbf{B}(2), .., \mathbf{B}(|\mathbf{d}_s|)$ is the set of basis matrices for the SD transformation and $\mathbf{d}^s$ is the SD interpolation vector. Similarly, the SD bias vector, $\mathbf{b}^s$ is given by:

$$\mathbf{b}^s = \mathbf{b} + \sum_{i=1}^{|\mathbf{v}^s|} \mathbf{v}^s(i)\mathbf{u}(k) = \mathbf{b} + \mathbf{U}\mathbf{v}^s \tag{8}$$

where $\mathbf{v}^s$ is the SD interpolation vector.

Furthermore, in [22] $\mathbf{B}(i)$ weight bases are constrained to be rank-1 matrices. This allows us to formulate the SD transformation:

$$\begin{aligned}\mathbf{W}^s &= \mathbf{W} + \sum_{i=1}^{|\mathbf{d}^s|} \mathbf{d}^s(i)\boldsymbol{\gamma}(i)\boldsymbol{\psi}^\top(i) \\ &= \mathbf{W} + \boldsymbol{\Gamma}\mathbf{D}^s\boldsymbol{\Psi}^\top\end{aligned} \tag{9}$$

where $\mathbf{B}(i) = \boldsymbol{\gamma}(i)\boldsymbol{\psi}^\top(i)$ and $\mathbf{D}^s$ is a diagonal matrix ($\mathbf{D}^s = \mathrm{diag}(\mathbf{d}^s)$) and $\boldsymbol{\gamma}(i)$, $\boldsymbol{\psi}(i)$ are the i-th column vectors for $\boldsymbol{\Gamma}$, $\boldsymbol{\Psi}$ respectively.

3.2. FHL Adaptation for LSTM-RNNs

FHL adaptation for LSTMP can be applied by modelling SD transformations and SD biases for various $\mathbf{W}_{**}$ and $\mathbf{b}_*$ in the LSTMPs (Equations (1) - (6)). For instance, we can estimate the SD transformations on the input feature ($\mathbf{x}_t$) as given below:

$$\mathbf{W}^s_{x*} = \mathbf{W}_{x*} + \boldsymbol{\Gamma}_{x*}\mathbf{D}^s_{x*}\boldsymbol{\Psi}^{l\top}_{x*} \tag{10}$$

where $\mathbf{D}^s_{x*} \in \mathbb{R}^{|\mathbf{d}^s| \times |\mathbf{d}^s|}$ is a diagonal matrix ($\mathbf{D}^s_{x*} = \mathrm{diag}(\mathbf{d}^s)$).

Similarly, an SD transformation is estimated for the recurrence connections as given below:

$$\mathbf{W}^s_{r*} = \mathbf{W}_{r*} + \boldsymbol{\Gamma}_{r*}\mathbf{D}^s_{r*}\boldsymbol{\Psi}^{l\top}_{r*}. \tag{11}$$

However, as found in [13], it is sufficient to estimate SD transformations on input features. Therefore, in this work we only estimates SD transformations on input features. Furthermore, we do not estimate any SD transformations for diagonal peephole weight matrices ($\mathbf{W}_{c*}$). Similar to the FHL adaptation for DNNs, the SD bias vector, $\mathbf{b}^s_*$ can be estimated for LSTMPs (Equation 8).

For both DNN and LSTMP-RNN adaptation, SD interpolation vectors are initialized with speaker-level i-vectors. During the second pass adaptation, these SD interpolation vectors are updated while keeping all other parameters fixed.

4. STUDENT-TEACHER TRAINING

Student-teacher training was first used to investigate the depth in deep neural networks [16]. Then, this method was used to compress a large DNN to a smaller DNN which can be deployed in devices with limited computational and storage resources [17]. Later, Hinton et al. [18] coined the term "knowledge distillation" and provided further evidence of the effectiveness of the student-teacher training algorithm.

In general, frame-level cross entropy (CE) criterion is used for DNN training :

$$\mathcal{F}_{CE} = -\sum_t \sum_{i=1}^{C} P^{ref}(i|\mathbf{x}_t)\log(P^{model}(i|\mathbf{x}_t)) \tag{12}$$

where C is the total number of context dependent (CD) HMM states and $P^{ref}(i|\mathbf{x}_t)$ is the probability of feature frame $\mathbf{x}_t$ belonging to class i in the reference distribution while $P^{model}(i|\mathbf{x}_t)$ is the probability of feature frame $\mathbf{x}_t$ belonging to class i according to the model being trained.

In standard training, the reference distribution is obtained from the forced alignment of the training data. In that case, $P^{ref}(i|\mathbf{x}_t)$ becomes a one-hot vector which is also known as training with hard labels. The simplified formulation is given below:

$$\mathcal{F}_{CE-Hard} = -\sum_t \log(P^{model}(i = c|\mathbf{x}_t)) \tag{13}$$

where c is the correct label.

In student-teacher training, instead of using the hard labels, a student model is trained to mimic the distribution of the teacher network as given below:

$$\mathcal{F}_{CE-Soft} = -\sum_t \sum_{i=1}^{C} P^{teacher}(i|\mathbf{x}_t)\log(P^{student}(i|\mathbf{x}_t)). \tag{14}$$

In general [20, 21], the student network is trained to minimize the following loss function which is an interpolation between the soft and hard CE losses:

$$\mathcal{F} = (1-\alpha)\mathcal{F}_{CE-Hard} + \alpha\mathcal{F}_{CE-Soft} \tag{15}$$

where α is the interpolation weight.

In this work, we incoporate student-teacher training to estimate FHL bases for LSTMP AMs. We start with a well-trained LSTMP AM and then an FHL-adapted DNN model is used as the teacher to estimate the FHL bases for the LSTMP AM. We keep all other weights fixed when estimating the FHL bases. Therefore, student-teacher training is only used to estimate the FHL bases. Furthermore, we do not interpolate teacher labels with the original hard targets. Therefore, we use

$$P^{teacher} = P^{FHL-DNN} \text{ and } P^{student} = P^{FHL-LSTMP}$$

during the FHL bases estimation in Equation 14.

5. EXPERIMENT SETUP

We use the AMI corpus which contains about 100 hours of meetings conducted in English. In the experiments, we use the IHM data and the speech from the first microphone in the array which is known as the SDM. We use the ASR split [28] of the corpus where 78 hours of the data are used for training while about 9 hours each are used for evaluation and development. We use 90% of the training set for training, and the rest is used as the validation set. The results are reported on the evaluation set.

For both the IHM and SDM datasets, we extract Mel-frequency cepstral coefficients (MFCCs) from the speech using a 25 ms window and a 10 ms frame shift. Then linear discriminant analysis (LDA) features are obtained by first splicing 7 frames of 13-dimensional MFCCs and then projecting down to 40 dimensions using LDA. A single semi-tied covariance (STC) transformation [29] is applied on top of the LDA features. We further extract speaker-normalized CMLLR (also known as fMLLR) features after applying speaker specific CMLLR transforms on top of these LDA+STC features. The GMM-HMM system for generating the alignments for DNNs and LSTMPs is trained on these 40 dimensional CMLLR features. We train the DNN-HMM baselines on the CMLLR features that span a context of 11 neighboring frames. Before being presented to the DNN, features are globally normalized to have zero mean and unit variance. DNNs have 6 sigmoid hidden layers with 2048 units per layer, and around 4000 outputs.

We train RNNs consist of 3 unidirectional LSTMP layers with 1024 memory cells and 512 dimensional projection as in [30]. The input feature is a single frame with a 5 frames shift. For the training, we use truncated back propagation through time (BPTT) with sequences of 20 frames. We process 40 sequences in parallel.

We conduct experiments on models trained to optimize the cross-entropy criterion as well as the state-level minimum Bayes risk (sMBR) criterion. All the DNNs and LSTMPs are trained using CNTK [31]. Kaldi [32] is used to build GMM-HMM systems and for i-vector extraction. The UBM consists of 128 full Gaussians. For decoding, we use the trigram language model as used in Kaldi, which is an interpolation of trigram language models trained on AMI and Fisher English transcripts. We do not use any data cleaning or frame-level dropout as used in Kaldi.

Table 1. *Word error rates (WER %) for baseline models trained on CMLLR features.*

Model	IHM	SDM
DNN	25.9	52.7
+ sMBR	24.3	50.0
LSTMP	25.3	49.6
+ sMBR	24.6	48.4

Table 2. *IHM : WER % for various models when FHL adaptation is applied to different layers of the LSTMP model trained on CMLLR features.*

Layer	First Pass	Second Pass
None (SD bias)	25.0	24.4
1	25.0	24.2
2	24.7	24.1
3	24.9	24.4

6. RESULTS

Table 1 shows the results for baseline DNNs and baseline LSTMP models trained on the IHM and SDM tasks. For both tasks, LSTMP models trained using the cross entropy criterion outperform the corresponding DNNs. However, the LSTMP model trained using the sMBR criterion performs slightly worse than the corresponding DNN for the IHM task. It is evident that DNNs benefit more from the sMBR criterion than LSTMP models. Furthermore, all LSTMP models trained on the SDM task perform significantly better than the corresponding DNNs. This can be because the superior temporal dependency modelling of LSTMPs is more beneficial for the noisy distant microphone speech in the SDM task.

In Table 2, we present the results when FHL adaptation is applied to different layers of the LSTMP model. First row results are for the case where only an SD bias is connected to the first hidden layer. As can be seen, the effectiveness of SD transformations in FHL adaptation is not evident from the results. However, in [13], FHL adaptation reported more gains when models are trained on LDA+STC features. Therefore, gains of the FHL adaptation diminish when AMs are trained on speaker normalized CMLLR features.

Table 3 presents results when FHL adaptation is applied to DNNs trained on both cross entropy and sMBR criterions. As can be clearly seen, FHL reports significant improvements. More specifically, gains we observe from the second pass over the first pass are significantly higher for DNNs than the that of LSTMPs shown in Table 2. This observation suggests that the estimated FHL bases for DNNs are more effective than the LSTMP FHL bases. Therefore, we employ student-teacher approach to estimate the FHL bases for LSTMP models by using FHL adapted DNNs as teachers.

Table 3. *IHM : WER % for FHL adapted DNN models.*

Model	First Pass	Second Pass
DNN	25.9	-
+ sMBR	24.3	-
+ FHL	25.2	23.8
+ sMBR	23.4	22.1

Table 4 presents the results for FHL adaptation of LSTMPs where FHL bases are estimated using student-teacher training. For all experiments, the FHL adapted sMBR DNN (WER of 23.4% in

Table 4. *IHM : WER % for LSTMP FHL adaptation where an FHL adapted DNN is used as the teacher.*

Model	First Pass	Second Pass
Baseline	25.3	-
SD bias only	27.9	25.8
Layer 1 (with SD bias)	25.8	24.0
Layer 2 (with SD bias)	24.1	23.0
Layer 3 (with SD bias)	23.5	22.6
Layer 3 (without SD bias)	23.3	22.8
All Layers (without SD bias)	23.3	22.1

Table 5. *IHM : Summary of results for LSTMP adaptation with student-teacher training.*

Model	First Pass	Second Pass
LSTMP	25.3	-
+ sMBR	24.6	-
+ FHL (ST)	23.3	22.1
+ sMBR	22.5	21.6

Table 3) is used as the teacher. It is worth highlighting the considerable degradation in performance of the model where an SD bias is connected to the first hidden layer. However, when SD transformations are estimated performance improves significantly. We get the best performance among the first passes when only SD transformations are estimated for layer 3 of the LSTMP model. However, the model with the SD biases connected to the first layer along with the SD transformations in the third layer (22.6%) outperforms the corresponding model without SD biases after the second pass (22.8%). This observation suggests that performing adaptation at multiple layers improves the second pass adaptation performance. Therefore, we train a model with SD transformation connected to all LSTMP layers. As expected this model enjoys the best performance of 22.1% which is a 3.2% absolute improvement over the LSTMP baseline.

Table 5 summarizes the adaptation results of LSTMP models trained on the IHM task. As can be clearly seen, FHL enjoys 3.2% and 3.0% absolute performance improvements over both cross entropy and sMBR trained LSTMP baselines respectively. According to the best of our knowledge, WER of 21.6% is the best result available for unidirectional LSTMP models. We use student-teacher training only when estimating the FHL bases for the LSTMP AM with cross entropy criterion.

Next, we report the results of adaptation experiments on the SDM task in Table 6. For both DNNs and LSTMPs, FHL improves the performance significantly. As expected, sMBR training delivers more gains over DNNs which is also in congruence with the IHM results. We obtain 2.3% and 4.1% absolute gains over the baseline DNN systems for the cross entropy and sMBR criterions, respectively. Furthermore, the FHL adapted LSTMP systems achieve 3.2% and 3.8% absolute improvements over the baselines trained using cross entropy and sMBR criterions, respectively.

Finally, in Table 7, we investigate the effectiveness of FHL adaptation when SDM models are trained using IHM alignments. As can be clearly seen, FHL enjoys significant improvements over DNNs as well as LSTMPs. It is worth highlighting that the performance gains from using IHM alignments are significantly better for DNNs than the that of LSTMPs. This is understandable as LSTMPs are more robust to errors in alignments due to their superior temporal

Table 6. *SDM : WER % for various adaptation experiments with student-teacher training.*

Model	First Pass	Second Pass
DNN	52.7	-
+ sMBR	50.0	-
+ FHL	51.7	50.4
+ sMBR	48.5	45.9
LSTMP	49.6	-
+ sMBR	48.4	-
+ FHL (ST)	47.9	46.4
+ sMBR	45.7	44.6

Table 7. *SDM : WER % for various models trained with IHM alignments.*

Model	First Pass	Second Pass
DNN	47.6	-
+ sMBR	44.9	-
+ FHL	47.2	46.4
+ sMBR	44.6	43.1
LSTMP	46.8	-
+ sMBR	46.6	-
+ FHL (ST)	44.8	43.8
+ sMBR	44.2	42.3

modeling capacity and consequently have report smaller gains when IHM alignments are used. Therefore, the FHL adaptation gains over DNNs trained with IHM alignments are smaller compared to that of LSTMPs. In summary, FHL obtains 3.0% and 4.3% absolute improvements over the LSTMP baselines trained using cross entropy and sMBR criterions, respectively.

7. CONCLUSIONS

Factorized Hidden Layer (FHL) was proposed for the adaptation of deep neural network (DNN) and then later extended to Long Short-Term Memory (LSTM) acoustic models (AMs). In FHL, a speaker-dependent (SD) transformation matrix and an SD bias are included in addition to the standard affine transformation. The SD transformation is a linear combination of rank-1 matrices whereas the SD bias is a linear combination of vectors. Even though FHL reported significant performance improvements for DNN adaptation, when applied for the LSTMPs, the gains were small. Therefore, this paper proposed to employ student-teacher training paradigm to estimate more efficient FHL bases for LSTMP AMs using an already FHL adapted DNN as the teacher model. Evaluations are performed on AMI IHM and AMI SDM tasks. FHL achieved 3.2% absolute improvement over the frame-level cross entropy trained LSTMP baselines in both IHM and SDM tasks. Moreover, FHL also reported significant improvements over sequentially trained LSTMP baselines with 3.0% and 3.8% absolute improvements for the IHM and SDM tasks respectively. Furthermore, when IHM alignments are used in training SDM models, FHL obtained 3.0% and 4.3% absolute improvements over the LSTMP baselines trained using cross entropy and sMBR criterions, respectively. As a future work, we plan to extend this approach to bidirectional LSTMP AMs.

This work was partially supported by grants from the RGC of the Hong Kong SAR, China (Project Nos. HKUST16215816 and HKUST16206714).

8. REFERENCES

[1] J. Gauvain and C.H. 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.

[2] 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.

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

[4] T. Anastasakos, J. McDonough, R. Schwartz, and J. Makhoul, "A compact model for speaker-adaptive training," in *ICSLP*. ISCA, 1996, vol. 2, pp. 1137–1140.

[5] M.J.F. Gales, "Cluster adaptive training of hidden Markov models," *IEEE Transactions on Speech and Audio Processing*, vol. 8, no. 4, pp. 417–428, 2000.

[6] L. Samarakoon and K.C. Sim, "Learning factorized transforms for speaker normalization," in *ASRU*. IEEE, 2015.

[7] L. Samarakoon and K.C. Sim, "On combining i-vectors and discriminative adaptation methods for unsupervised speaker normalization in DNN acoustic models," in *ICASSP*. IEEE, 2016.

[8] 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 *ICASSP*. IEEE, 2013, pp. 7893–7897.

[9] V. Gupta, P. Kenny, P. Ouellet, and T. Stafylakis, "I-vector-based speaker adaptation of deep neural networks for French broadcast audio transcription," in *ICASSP*. IEEE, 2014, pp. 6334–6338.

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

[11] 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 *ICASSP*. IEEE, 2013, pp. 7942–7946.

[12] Dong Yu, , and Li Deng, *Automatic Speech Recognition - A Deep Learning Approach*, Springer London, New York, 2015.

[13] L. Samarakoon, B. Mak, and K.C. Sim, "Learning factorized transforms for unsupervised adaptation of LSTM-RNN acoustic models," in *Interpeech*. ISCA, 2017.

[14] Chaojun Liu, Yongqiang Wang, Kshitiz Kumar, and Yifan Gong, "Investigations on speaker adaptation of LSTM RNN models for speech recognition," in *ICASSP*. IEEE, 2016, pp. 5020–5024.

[15] Y. Zhao, J. Li, K. Kumar, and Y. Gong, "Extended low-rank plus diagonal adaptation for deep and recurrent neural networks," in *ICASSP*, 2017.

[16] Jimmy Ba and Rich Caruana, "Do deep nets really need to be deep?" in *Advances in neural information processing systems*, 2014, pp. 2654–2662.

[17] Jinyu Li, Rui Zhao, Jui-Ting Huang, and Yifan Gong, "Learning small-size DNN with output-distribution-based criteria," in *Interspeech*, 2014.

[18] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean, "Distilling the knowledge in a neural network," *arXiv preprint arXiv:1503.02531*, 2015.

[19] J. Cui, B. Kingsbury, B. Ramabhadran, G. Saon, T. Sercu, K. Audhkhasi, A. Sethy, M. Nussbaum-Thom, and A. Rosenberg, "Knowlege distillation across ensembles of multilingual models for low-resource languages," in *ICASSP*, 2017.

[20] T. Asami, R. Masumura, Y. Yamaguchi, H. Masataki, and Y. Aono, "Domain adaptation of DNN acoustic models using knowledge distillation," in *ICASSP*, 2017.

[21] S. Watanabe, T. Hori, J. Le Roux, and J. R. Hershey, "Student-teacher network learning with enhanced features," in *ICASSP*, 2017.

[22] L. Samarakoon and K.C. Sim, "Factorized hidden layer adaptation for deep neural network based acoustic modeling," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2016.

[23] I. McCowan, J. Carletta, W. Kraaij, S. Ashby, S. Bourban, M. Flynn, M. Guillemot, T. Hain, J. Kadlec, V. Karaiskos, et al., "The AMI meeting corpus," in *Proceedings of the 5th International Conference on Methods and Techniques in Behavioral Research*, 2005, vol. 88.

[24] Sepp Hochreiter and Jürgen Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[25] Felix A Gers, Jürgen Schmidhuber, and Fred Cummins, "Learning to forget: Continual prediction with LSTM," *Neural computation*, vol. 12, no. 10, pp. 2451–2471, 2000.

[26] Felix A Gers, Nicol N Schraudolph, and Jürgen Schmidhuber, "Learning precise timing with LSTM recurrent networks," *Journal of machine learning research*, vol. 3, no. Aug, pp. 115–143, 2002.

[27] Hasim Sak, Andrew W Senior, and Françoise Beaufays, "Long short-term memory recurrent neural network architectures for large scale acoustic modeling," in *Interspeech*, 2014, pp. 338–342.

[28] P. Swietojanski, A. Ghoshal, and S. Renals, "Hybrid acoustic models for distant and multichannel large vocabulary speech recognition," in *ASRU*. IEEE, 2013, pp. 285–290.

[29] M.J.F. Gales, "Semi-tied covariance matrices for hidden Markov models," *IEEE Transactions on Speech and Audio Processing*, vol. 7, no. 3, pp. 272–281, 1999.

[30] Yu Zhang, Guoguo Chen, Dong Yu, Kaisheng Yaco, Sanjeev Khudanpur, and James Glass, "Highway long short-term memory RNNs for distant speech recognition," in *ICASSP*. IEEE, 2016.

[31] D. Yu, A. Eversole, M. Seltzer, K. Yao, Z. Huang, B. Guenter, O. Kuchaiev, Y. Zhang, F. Seide, H. Wang, et al., "An introduction to computational networks and the computational network toolkit," Tech. Rep., Tech. Rep. MSR, Microsoft Research, 2014, http://codebox/cntk, 2014.

[32] D. Povey, A. Ghoshal, G. Boulianne, N. Goel, M. Hannemann, Y. Qian, P. Schwarz, and G. Stemmer, "The Kaldi speech recognition toolkit," in *ASRU*. IEEE, 2011.