INTERSPEECH 2019
September 15–19, 2019, Graz, Austria

Fast DNN Acoustic Model Speaker Adaptation by Learning Hidden Unit Contribution Features

Xurong Xie[1,2]*, Xunying Liu*[1,2]*, Tan Lee*[1]*, Lan Wang*[2]

[1]Chinese University of Hong Kong, Hong Kong, China
[2]Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen, China
{xrxie, tanlee}@ee.cuhk.edu.hk, xyliu@se.cuhk.edu.hk, lan.wang@siat.ac.cn

Abstract

Speaker adaptation techniques play a key role in reducing the mismatch between automatic speech recognition (ASR) systems and target users. Deep neural network (DNN) acoustic model adaptation by learning speaker-dependent hidden unit contributions (LHUC) scaling vectors has been widely used. The standard LHUC method not only requires multiple decoding passes in test time but also a substantial amount of adaptation data for robust parameter estimation. In order to address the issues, an efficient method of predicting and compressing the LHUC scaling vectors directly from acoustic features using a time-delay DNN (TDNN) and an online averaging layer is proposed in this paper. The resulting LHUC vectors are then used as auxiliary features to adapt DNN acoustic models. Experiments conducted on a 300-hour Switchboard corpus showed that the DNN and TDNN systems using the proposed predicted LHUC features consistently outperformed the corresponding baseline systems by up to about 9% relative reductions of word error rate. Being combined with i-Vector based adaptation, the LHUC feature adapted TDNN systems demonstrated consistent improvement over comparable i-Vector adapted TDNN system.
Index Terms: LHUC, speaker adaptation, online, time-delay

1. Introduction

Speaker adaptation techniques play a vital role in speech recognition systems for reducing the mismatch against target users. Current approaches to speaker adaptation for deep neural network (DNN) based speech recognition systems can be divided into three categories. Auxiliary input feature based DNN adaptation techniques encode speaker-dependent (SD) characteristics in a compact vector to facilitate model adaptation, such as i-Vectors [1, 2, 3], speaker codes [4, 5, 6], and bottleneck features [7]. In model based DNN speaker adaptation techniques, SD parameters represented by, for example, hidden layers or input layer linear transforms of each speaker are either separately learned from GMM-HMMs [8, 9, 10] or jointly estimated with the remaining DNN parameters [11, 12]. A set of scaling vectors for learning hidden layer unit contributions (LHUC) [13, 14, 15] among different speakers are also possible to be used. Interpolation based DNN adaptation with multiple basis of sub-network hidden outputs has also been proposed [16, 17], inspired by the speaker cluster based adaptation techniques originally proposed for GMM-HMM systems [18].

Many state of the art speaker adaptation techniques, for example, the LHUC based adaptation, have to use words transcribed by decoding passes as supervision to explicitly and iteratively estimate the SD parameters. In addition, the parameter estimation may be sensitive to supervision quality. When using limited amount of adaptation data, mismatch against later data caused by data sparsity problem may lead to performance degradation. Although the Bayesian LHUC proposed in the previous work [19] for gaining generalization has partially solved the data sparsity problem, it still requires multiple decoding passes and explicit parameter estimation in test time.

This paper proposes an efficient method of predicting and compressing the LHUC scaling vectors from acoustic features on the fly. A special designed feature prediction network is built with a time-delay DNN (TDNN) [20] and an online averaging layer, which can compute accumulated average of history hidden vectors for each speaker. The resulting LHUC feature vectors are then used as auxiliary features to adapt DNN acoustic models, via feature concatenation similar to i-Vector adaptation, or via the the previously proposed feature based LHUC [21] to restore to the full dimension of standard LHUC scaling vector.

The main contribution of this proposed method is summarized below. LHUC features of test data can be directly predicted for decoding on the fly. Neither the multiple decoding passes nor explicit parameter estimation is required. On the contrary, the standard LHUC based adaptation requires both multiple decoding passes and sufficient amount of adaptation data for robust adaptation. Moreover, the proposed LHUC features are found to be complementary to the standard LHUC based adaptation and used in combination for dealing with data sparsity.

In the experiments, conventional cross entropy (CE) trained DNN and lattice-free maximum mutual information (LF-MMI) trained TDNN acoustic models adapted by the proposed LHUC features on the fly were built on a 300-hour Switchboard setup, and evaluated on the Hub5' 00 data set. Consistent performance improvements up to about 9% relative reductions in terms of word error rate (WER) over the baseline system were obtained with the LHUC feature based adaptation. When applying i-Vector based speaker adaptive training (SAT) [3] together, the best LHUC feature adapted TDNN systems significantly outperformed the baseline system and comparable i-Vector SAT TDNN system on the **CallHome** data set by WER reduction of 1.8% and 0.6% absolute respectively.

In the next section, the standard LHUC method is reviewed. Section 3 introduces the proposed fast LHUC feature prediction and acoustic model adaptation techniques. In section 4, LHUC feature adapted DNN/TDNN systems are evaluated on the Switchboard databases. Section 5 presents the conclusion.

2. LHUC based adaptation

Learning hidden unit contribution (LHUC) [13] based speaker adaptation learns speaker-dependent (SD) scaling vectors to modify the activation amplitude of DNN hidden units for each speaker. In the lth hidden layer, let $\boldsymbol{r}^{l,s} \in \mathbb{R}^D$ denote the parameter set of scaling vector for speaker s. The hidden layer

http://dx.doi.org/10.21437/Interspeech.2019-2050

output is computed as

$$\boldsymbol{h}^{l,s} = \xi(\boldsymbol{r}^{l,s}) \otimes \psi(\boldsymbol{W}^{l\top}\boldsymbol{h}^{l-1,s} + \boldsymbol{b}^{l}) \quad (1)$$

where $\boldsymbol{W}^l$ and $\boldsymbol{b}^l$ were the DNN weight matrix and bias vector, ψ denotes the hidden unit activation function, and $\otimes$ denotes the Hadamard product. The scaling vectors are modeled by the function $\xi : \mathbb{R}^D \rightarrow \{\mathbb{R}^+\}^D$ on $\boldsymbol{r}^{l,s}$. In this work the element-wise function $\xi(\cdot) = 2\text{sigmoid}(\cdot)$ [13, 14] is utilized, while linearity, ReLU [15], or EXP [13] functions can be used alternatively. Figure 1 shows an example of applying LHUC adaptation in a DNN acoustic model.

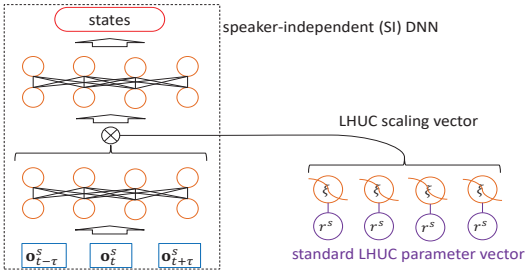

Figure 1: *Using LHUC adaptation in a DNN acoustic model.*

To adapt DNNs, LHUC parameters $\boldsymbol{r}^{l,s}$ for adaptation data can be appended to the well trained speaker-independent (SI) DNN acoustic model in the test stage. Then, a few fine-tuning epochs based on the first-pass of decoding result is used to optimize the SD parameters for next decoding. Alternatively, the LHUC parameters for training data can be fine-tuned together with the SI DNN acoustic model via speaker adaptive training (SAT), and the trained SI DNN part can then be used in test time adaptation. The test time adaptation requires at least two passes of decoding and iterative SD parameter estimation on the adaptation data. In practice, decoding results may be requested in a few seconds after speaking, and the test time adaptation may be inefficient enough for usage.

3. LHUC feature for fast adaptation

3.1. Proposed LHUC feature prediction network

The idea of fast LHUC adaptation is to directly predict LHUC parameters for test data from acoustic features on the fly, without requiring pre-decoding and iterative parameter estimation. First of all, LHUC parameters of training data have to be estimated on a DNN acoustic model via supervised manner. Subsequently, a feature prediction network is trained with the acoustic features to predict the estimated LHUC parameters $\boldsymbol{r}^s$ for each frame of data. However, the size of LHUC parameter vector may be large such that the training is slow and difficult to converge. Compression of the vector size can be first achieved using principle component analysis (PCA). When applying PCA all speakers are assumed to have equal probability. The data size for PCA becomes the number of speakers, as each speaker has only one set of LHUC parameters. This makes the PCA compression much faster. Finally, the compressed LHUC parameter vector for each frame is utilized as training labels. The mean squared error (MSE) between LHUC feature vector outputted from the prediction network and the compressed LHUC parameter vector is utilized as objective function for training.

To capture the relationship among individual frames of speech, recurrent neural network (RNN) or time-delay DNN (TDNN) [20] can be employed to realize the LHUC feature prediction network. In this work TDNN is used because of its relative high efficiency. The prediction network architecture is shown in Figure 2. For each context splicing layer (red block), the input is a splicing of the outputs from the preceding layer in fixed context, and the output is activated by the sigmoid function. Between two context splicing layers, a bottleneck (purple block) is employed to constrict the spliced vector size by linear transformation.

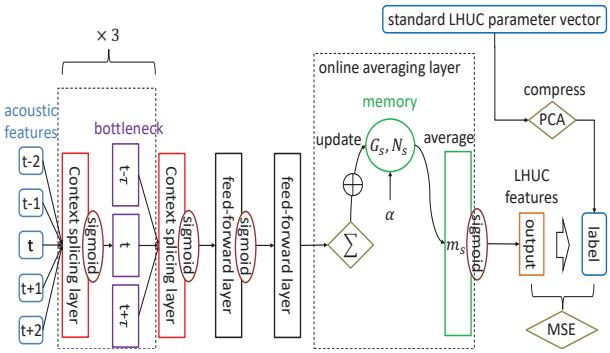

Figure 2: *Proposed LHUC feature prediction network.*

In this work, the LHUC feature prediction network consists of 4 context splicing layers followed by 2 feed-forward layers. Each of these hidden layers and bottlenecks has 500 nodes and 300 nodes respectively. The splicing indices of the context splicing layers are $[-2, 2], \{-2, 0, 2\}, \{-3, 0, 3\}, \{-4, 0, 4\}$. Since the LHUC parameter vector represents information over all data for each speaker, considering a short context in the prediction network may not be enough. Therefore, an online averaging layer can be employed preceding the output layer.

3.2. Online averaging layer

In order to facilitate the efficient and online adaptation, an alternative online method computing the accumulated average of hidden history vectors in the prediction network is employed. This is shown as the green part in Figure 2. This special designed online averaging layer encodes memory of preceding speech segments associated with the same speaker. This memory stores the accumulated history hidden vector and frame count for each speaker, which can be calculated by

$$\boldsymbol{G}_s^{(k)} = \sum_{t=1}^{T_k} \boldsymbol{h}_t^{(k)} + \alpha \boldsymbol{G}_s^{(k-1)}, \quad N_s^{(k)} = T_k + \alpha N_s^{(k-1)} \quad (2)$$

where $\boldsymbol{h}_t^{(k)}$ is the tth frame of hidden vector input in the kth segment, T_k denotes the number of frames in the segment, $\boldsymbol{G}_s^{(k)}$ and $N_s^{(k)}$ denote the accumulated history hidden vector and frame count for speaker s when computing the memory of the kth segment, and $\alpha \in [0, 1]$ is the history interpolation weight for trade off between history memory of the previous and current segments. $\boldsymbol{G}_s^{(0)} = 0$ and $N_s^{(0)} = 0$ are utilized as initialization for all speakers. Subsequently, the online averaging layer output is the accumulated average history memory of all speech history up to the kth segment of the current speaker, and computed as $\boldsymbol{m}_s^{(k)} = \boldsymbol{G}_s^{(k)} / N_s^{(k)}$. Compared to using alternative methods, for example, RNN for use of history memory over all history frames, the designed online averaging layer using accumulated average of history memory for segments is not

only found to be efficient, but also allow additional flexibility to balance the current and previous segments for adaptation by a tunable weight α. The gradient for each frame is the copy of that back-propagated from the following layer.

3.3. LHUC feature based adaptation

After the LHUC feature prediction network trained, training data for acoustic modeling can be fed into the prediction network and produce LHUC features. Then, LHUC features can be utilized as auxiliary features to be simply concatenated with acoustic features as input. Alternatively, without PCA compression, the prediction network can produce standard form of parameters for standard LHUC adaptation. However, by applying PCA compression this is no long feasible. Feature based LHUC (f-LHUC) [21] approach can thus be employed to restore the compressed LHUC features to the standard form. These approaches are shown in Figure 3. For fast adaptation in test time, LHUC features can be directly computed by feeding acoustic features of test speech segments on the fly into the LHUC feature prediction network and used for decoding.

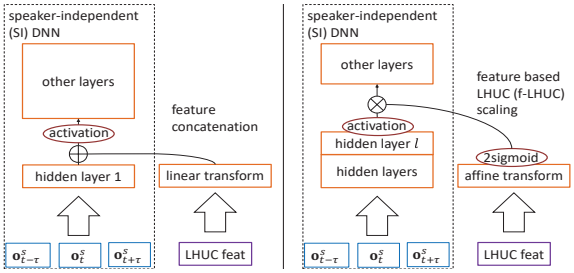

Figure 3: *Left: feature concatenation; Right: f-LHUC.*

4. Experiments

4.1. Experimental setup

The proposed LHUC feature is investigated for both conventional CE trained DNN-HMM acoustic model adaptation and LF-MMI trained TDNN-HMM acoustic model adaptation on a 300-hour Switchboard setup. A four-gram language model with 30-thousand words was employed for evaluation on the Hub5' 00 data set with **SWBD** and **CallHome** test sets. In order to obtain the LHUC parameters, feed-forward DNN consisting of 6 hidden layers and LHUC parameters in the first hidden layer was trained by minimizing CE with the alignments of 8929 tied tri-phone states. Each of the hidden layer contained 2000 hidden nodes and was activated by sigmoid function. 9 successive frames of 80 dimensional filter-bank features with the first order difference were used as the DNN input. Back propagation with stochastic gradient descent (SGD) was employed to update the 800-frame mini-batches.

After the above DNN trained, the 2000 dimensional LHUC parameter vectors was compress to 100 dimensional vectors by PCA. Then, the LHUC feature prediction network with architecture mentioned in section 3.1 is trained on 40 dimensional filter-bank features and updated by RMSProp. The history interpolation weight in online averaging layer was set as $\alpha = 0.9$.

The LHUC feature adapted conventional DNN acoustic model had the same architecture with the above DNN for LHUC parameter estimation. For TDNN acoustic models, the baseline was trained following the default setup of example script *"egs/swbd/s5c/local/chain/tuning/run_tdnn_7q.sh"* in the Kaldi tookit [22]. The TDNN consisted of 15 context splicing layers with 1536 nodes and output layer with 4456 nodes. Each context splicing layer contained a 160 dimensional bottleneck and was activated by ReLU function, followed by batch normalization, dropout, and scaled summation with the preceding context splicing layer output. The splicing indices were $\{-1, 0, 1\}$ for the first 4 context splicing layers, and $\{-3, 0, 3\}$ for the last 10 context splicing layers. For optimization, the lattice-free MMI criterion using leaky HMM was employed together with the CE regularization. Natural-gradient was utilized for SGD update. In the experiments, 40 dimensional filter-bank features were used as input. Because of the restriction on computing resource, 4 epochs of training were run with single thread without using speed perturbation and high resolution features. About 280 hours of data were selected for training by removing over-represented transcriptions. All systems were trained and evaluated with a modified version of Kaldi tookit and HTK [23].

4.2. Performance on conventional DNN acoustic model

Performance of different conventional CE trained DNN-HMM systems using or without using the predicted LHUC features for adaptation were evaluated on the **SWBD** and **CallHome** test sets. Both the standard LHUC (test time adaptation only) and LHUC SAT systems were decoded by two passes of decoding, where the first pass is to generate supervision for SD parameter estimation. When using the predicted LHUC features proposed in this paper, there were two choices: using them in test time adaptation, or in SAT training framework. For either of this two choice, the utilization of the predicted LHUC features can be further implemented by using two methods, feature concatenation shown in left side of Figure 3, or the f-LHUC shown in right side of Figure 3. These two level of choices led to four systems in Table 1, which were denoted by Sys (4) and Sys (6).

Table 1: *Performance of LHUC feature adapted DNNs with or without using SAT, via either feature concatenation in the left of Figure 3, or f-LHUC in the right of Figure 3.*

Sys	CE DNN adaptation	Usage of LHUC feat	SAT	WER (%)	
				SWBD	CallHome
(1)	SI baseline	-	-	15.3	27.6
(2)	LHUC	-	×	14.6	25.8
(3)		-	√	13.2	23.5
(4)	LHUC feat	feat concat	×	14.4	26.8
		f-LHUC		14.5	26.6
(5)	LHUC feat + LHUC	f-LHUC	×	14.0	25.1
(6)	LHUC feat	feat concat	√	14.1	26.8
		f-LHUC		14.0	26.4
(7)	LHUC feat + LHUC	f-LHUC	√	**12.9**	**23.0**

Figure 4 shows the performance of standard LHUC and LHUC feature adaptation of DNN systems when the first utterance only, the first 10%, 25%, 50% and up to 100% of all test data were used as adaptation data, respectively. Two trends can be found in Table 1 and Figure 4. First, fast adaptation by LHUC features (purple solid line and purple dash line) outperformed the standard LHUC adaptation (blue solid line and blue dash line) when limited amount of adaptation data, for example, 10% of test data is used. When using test time adaptation only, LHUC feature based adaptation (purple solid line) consistently performed better than the standard LHUC adaptation (blue solid line) on the **SWBD** test set, irrespective the amount of adaptation data. On the more challenging and mismatch **CallHome** test set, LHUC features (purple solid line) outperformed

the standard LHUC adaptation (blue solid line) up to 25% data amount. This demonstrated that LHUC features could handle the data sparsity problem. Second, LHUC feature based adaptation could be used in combination with standard LHUC adaptation for improvement, if fast adaptation is not necessary. In this case, standard LHUC SD parameters were appended to the LHUC feature adapted DNNs as description in section 2. As expected, the standard LHUC adaptation got further improvement by combining LHUC features in test time only (Sys (5); red solid line) over the standard LHUC adapted systems (Sys (2); blue solid line). Consistent trend can be observed on the SAT version of corresponding adaptation.

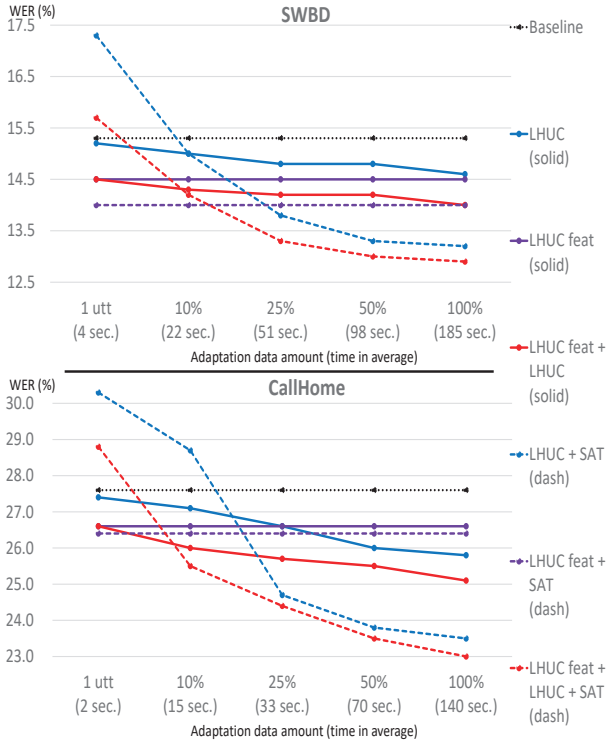

Figure 4: *Performance contrast of DNN systems adapted by LHUC features and standard LHUC adaptation using various amounts of adaptation data on* **SWBD** *and* **CallHome** *test sets. LHUC features were all used in f-LHUC framework.*

4.3. Performance on TDNN acoustic model

Performance of different LF-MMI trained TDNN-HMM systems using or without using the predicted LHUC features for adaptation were further evaluated on the **SWBD** and **CallHome** test sets and shown in Table 2. The baseline system (Sys (1)) obtained similar results to the LF-MMI trained TDNN system in [24]. All TDNN systems here were adapted via SAT. Three trends can be observed from Table 2. First, when online averaging layer was used in the prediction network with history interpolation weight $\alpha > 0$, all LHUC feature adapted systems consistently outperformed the baseline TDNN system (Sys (1)). In contrast, when online averaging layer was not used, the LHUC feature based adaptation (Sys (2)) got no improvement over the baseline TDNN system. The best performance was obtained by setting the history interpolation weight as $\alpha = 0.9$. Second, the best LHUC feature adapted TDNN system (Sys (10)) by using f-LHUC in both the first and third hidden layers obtained comparable performance to the i-Vector based SAT TDNN system (Sys (11)), and slightly outperformed it on the **CallHome** test set. Third, when combining the i-Vector and LHUC feature based adaptation, the TDNN systems ((13) and (14)) outperformed both the baseline TDNN system and i-Vector SAT TDNN system, especially on **CallHome** test set by WER reduction of 1.8% and 0.6% absolute respectively.

Table 2: *Performance on LHUC feature adapted TDNNs. The label α denotes the history interpolation weight in online averaging layer. $\alpha =$ "-" means no online averaging layer is used. The label "l" denotes the hidden layer using f-LHUC.*

Sys	LF-MMI TDNN adaptation	Usage of LHUC feat	α	WER (%) SWBD	WER (%) CallHome
(1)	SI baseline	-	-	10.1	20.6
(1.1)	Povey *et al [24]*	-	-	10.2	20.5
(2)	LHUC feat	feat concat	-	10.1	21.5
(3)			0.0	9.6	21.0
(4)			0.5	9.4	20.5
(5)			0.7	9.4	20.3
(6)			0.8	9.3	20.0
(7)			0.9	**9.2**	**19.7**
(8)			1.0	9.3	19.8
(9)		f-LHUC in $l = \{1\}$	0.9	9.7	19.5
(10)		f-LHUC in $l = \{1, 3\}$	0.9	**9.4**	**19.1**
(11)	i-Vector	-	-	9.2	19.4
(12)	LHUC feat + i-Vector	feat concat	0.9	**8.9**	19.5
(13)		f-LHUC in $l = \{1\}$	0.9	9.1	**18.8**
(14)		f-LHUC in $l = \{1, 3\}$	0.9	9.1	**18.8**

5. Conclusions

This paper proposed a fast adaptation approach using prediction of the compressed LHUC speaker-dependent parameters directly from the acoustic features on the fly without additional decoding passes being requaired. The prediction network consisting a time-delay DNN (TDNN) and an online averaging layer was also proposed. The predicted LHUC features were then used as auxiliary features to adapt DNN acoustic models. Moreover, LHUC feature is found to be complementary to the standard LHUC adaptation for dealing with data sparsity problem, and also complementary to the i-Vector adaptation as well. Experiments conducted on a 300-hour Switchboard corpus showed that the DNN and TDNN systems adapted by the proposed predicted LHUC features obtained consistent improvements over the corresponding baseline DNN and TDNN systems by relative reductions of word error rate up to about 9%. After being used together with i-Vector based adaptation, the LHUC feature adapted TDNN systems consistently outperformed the comparable i-Vector adapted TDNN system. Future works will focus on investigating different architectures of the LHUC feature prediction network.

6. Acknowledgements

This research was partially supported by a direct grant from Research Committee of the Chinese University of Hong Kong (CUHK), GRF grants (Ref.: 14227216 and 14200218) from the Hong Kong Research Grants Council, the Major Program of National Social Science Fund of China (Ref: 13&ZD189), Shun Hing Institute of Advanced Engineering Project No. MMT-p1-19, Natural Science Foundation of China U1736202, and Shenzhen Fundamental Research Program JCYJ20160429184226930 and KQJSCX20170731163308665.

7. References

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

[2] A. Senior and I. Lopez-Moreno, "Improving dnn speaker independence with i-vector inputs," in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*. IEEE, 2014, pp. 225–229.

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

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

[5] 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 *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*. IEEE, 2014, pp. 6339–6343.

[6] 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 Transactions on Audio, Speech and Language Processing (TASLP)*, vol. 22, no. 12, pp. 1713–1725, 2014.

[7] H. Huang and K. C. Sim, "An investigation of augmenting speaker representations to improve speaker normalisation for dnn-based speech recognition," in *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*. IEEE, 2015, pp. 4610–4613.

[8] 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.

[9] V. V. Digalakis, D. Rtischev, and L. G. Neumeyer, "Speaker adaptation using constrained estimation of gaussian mixtures," *IEEE Transactions on speech and Audio Processing*, vol. 3, no. 5, pp. 357–366, 1995.

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

[11] B. Li and K. C. Sim, "Comparison of discriminative input and output transformations for speaker adaptation in the hybrid nn/hmm systems," 2010.

[12] J. Trmal, J. Zelinka, and L. Müller, "Adaptation of a feedforward artificial neural network using a linear transform," in *International Conference on Text, Speech and Dialogue*. Springer, 2010, pp. 423–430.

[13] P. Swietojanski, J. Li, and S. Renals, "Learning hidden unit contributions for unsupervised acoustic model adaptation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 24, no. 8, pp. 1450–1463, 2016.

[14] P. Swietojanski and S. Renals, "Differentiable pooling for unsupervised speaker adaptation," in *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*. IEEE, 2015, pp. 4305–4309.

[15] C. Zhang and P. C. Woodland, "Dnn speaker adaptation using parameterised sigmoid and relu hidden activation functions," in *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*. IEEE, 2016, pp. 5300–5304.

[16] C. Wu and M. J. Gales, "Multi-basis adaptive neural network for rapid adaptation in speech recognition," in *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*. IEEE, 2015, pp. 4315–4319.

[17] T. Tan, Y. Qian, and K. Yu, "Cluster adaptive training for deep neural network based acoustic model," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 24, no. 3, pp. 459–468, 2016.

[18] M. J. Gales, "Cluster adaptive training of hidden markov models," *IEEE transactions on speech and audio processing*, vol. 8, no. 4, pp. 417–428, 2000.

[19] X. Xie, X. Liu, T. Lee, S. Hu, and L. Wang, "Blhuc: Bayesian learning of hidden unit contributions for deep neural network speaker adaptation," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 5711–5715.

[20] V. Peddinti, D. Povey, and S. Khudanpur, "A time delay neural network architecture for efficient modeling of long temporal contexts," in *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[21] X. Xie, X. Liu, T. Lee, and L. Wang, "Rnn-lda clustering for feature based dnn adaptation." in *INTERSPEECH*, 2017, pp. 2396–2400.

[22] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz *et al.*, "The kaldi speech recognition toolkit," in *IEEE 2011 workshop on automatic speech recognition and understanding*, no. EPFL-CONF-192584. IEEE Signal Processing Society, 2011.

[23] S. Young, G. Evermann, M. Gales, T. Hain, D. Kershaw, X. Liu, G. Moore, J. Odell, D. Ollason, D. Povey *et al.*, "The htk book," *Cambridge university engineering department*, vol. 3, p. 175, 2002.

[24] D. Povey, V. Peddinti, D. Galvez, P. Ghahremani, V. Manohar, X. Na, Y. Wang, and S. Khudanpur, "Purely sequence-trained neural networks for asr based on lattice-free mmi," in *Interspeech*, 2016.