INTERSPEECH 2020
October 25–29, 2020, Shanghai, China

Adaptive Speaker Normalization for CTC-Based Speech Recognition

Fenglin Ding, Wu Guo, Bin Gu, Zhenhua Ling, Jun Du

National Engineering Laboratory for Speech and Language Information Processing
University of Science and Technology of China, Hefei, China
{flding,bin2801}@mail.ustc.edu.cn, {guowu,zhling,jundu}@ustc.edu.cn

Abstract

In this paper, we propose a new speaker normalization technique for acoustic model adaptation in connectionist temporal classification (CTC)-based automatic speech recognition. In the proposed method, for the inputs of a hidden layer, the mean and variance of each activation are first estimated at the speaker level. Then, we normalize each speaker representation independently by making them follow a standard normal distribution. Furthermore, we propose using an auxiliary network to dynamically generate the scaling and shifting parameters of speaker normalization, and an attention mechanism is introduced to improve performance. The experiments are conducted on the public Chinese dataset AISHELL-1. Our proposed methods present high effectiveness in adapting the CTC model, achieving up to 17.5% character error rate improvement over the speaker-independent (SI) model.

Index Terms: speaker normalization, speech recognition, connectionist temporal classification

1. Introduction

With the widespread use of deep learning in automatic speech recognition (ASR), recognition accuracy has been greatly improved over the past several years [1, 2]. However, the performance of deep neural network (DNN)-based ASR will still deteriorate under the mismatches between training and test conditions, which are caused by the different characteristics of acoustic variability, such as speakers, channels and environmental noise. In ASR, speaker normalization (SN) techniques are used to minimize the mismatch between the training and testing conditions due to speaker variability. Typical normalization techniques transform the model to match the testing condition or the inputs to match the model.

Speaker normalization techniques for DNNs can be categorized into two broad approaches: adaptation and adaptive training. Speaker adaptation methods address speaker variability by estimating speaker-dependent (SD) parameters from a trained speaker-independent (SI) model on additional adaptation data. Speaker adaptive training attempts to address the speaker mismatch during training on the fly.

For the adaptation method, a straightforward idea is to retrain all SI model parameters. To avoid overfitting, regularization approaches such as L2 regularization using weight decay [3], Kullback-Leibler divergence (KLD) [4] and adversarial multitask learning (MTL) [5] were proposed. There are also many approaches in which only small subsets of the network parameters are adapted [6, 7, 8]. Recently, adaptation schemes using parameterized hidden activation functions have been widely explored [9, 10, 11] and have achieved good improvements.

In adaptive training, a traditional technique is to transform the acoustic features to a normalized space, and then the adapted features are used to train DNN models. Typical methods include MLLR transforms and the feature-space variant (fMLLR) [12, 13]. Another effective method is to provide the network with auxiliary features that characterize speaker information such as i-vector [14, 15, 16] and speaker code [17, 18]. In addition, cluster adaptive training (CAT) has been applied for speaker normalization [19, 20].

Despite the great success of these methods in hybrid systems, there has been limited investigation in speaker normalization for the end-to-end (E2E) ASR. In [21], two regularization-based approaches were shown to be effective for connectionist temporal classification (CTC) [22]-based E2E ASR. In [23], several conventional adaptation methods were integrated to adapt the attention-based encoder-decoder (AED) model.

In this paper, we propose a novel speaker normalization technique for CTC-based ASR. The CTC models take all utterances as input and produce a sequence of activations. These allow us to make use of better context modeling capabilities and statistical information of hidden activations for a speaker. Additionally, inspired by the idea of batch normalization (BN) [24], we propose to normalize each speaker representation independently by making each activation follow the standard normal distribution. The mean and variance of each activation are estimated at the speaker level. Then, a pair of scaling and shifting parameters are introduced to transform the normalized value, which are learned along with the original model parameters. Furthermore, motivated by dynamic layer normalization (DLN) [25] and attentive batch normalization (ABN) [26], we also use an auxiliary network with an attention mechanism to dynamically generate the normalization parameters, which we call adaptive speaker normalization (ASN). However, unlike DLN and ABN, we propose to generate the parameters at the batch level and at the speaker level to fulfill speaker adaptation. We evaluated the proposed algorithms on the AISHELL-1 corpus [27], an open-source Mandarin ASR task. Experimental results show that the proposed methods present high effectiveness in adapting the CTC model, achieving up to 17.5% character error rate (CER) improvement over the speaker-independent (SI) model.

2. Relation to prior work

Well-known normalization techniques for reducing the train-test mismatch include the application of input normalization, such as the mean normalization (MN) [28] and mean and variance normalization (MVN) [29]. MN assumes that the data mean is invariant, and MVN uses the stronger assumption that the mean and variance of data are invariant, so standardizing the mean and/or variance removes irrelevant information [30]. In deep learning, un-normalized features with greater variance dominate the DNN learning process, so scaling the inputs is a standard procedure that can improve DNN performance.

Similar normalization techniques can be found in DNN

http://dx.doi.org/10.21437/Interspeech.2020-1390

training. Batch normalization (BN)[24] and layer normalization (LN) [31] are two well-known methods for normalizing the activations of the hidden layers. BN was originally designed to alleviate the issue of internal covariate shifting, a common problem in DNN training. BN addresses the problem by normalizing each dimension of activations in a mini-batch by making it follow a standard normal distribution. LN has the same idea as batch normalization, but the difference is that LN normalizes each node of a neural layer, which is independent of the size of each batch.

However, BN or LN is a DNN training technique that is not targeted at speaker adaptation. There has still been limited investigation in speaker adaptation using similar normalization techniques. In [32], researchers used the auxiliary network to learn speaker-specific information and then performed normalization at the speaker level. Although the method achieves a lower word error rate (WER) than the unadapted models, it only uses the mean information of activations and performs normalization at a specific layer. The novelty of our proposed methods lies in the following aspects: first, we use the idea of BN to perform normalization for activations at the speaker level, which is layer-wise and makes use of the mean and variance information of activations. Furthermore, we introduce an attention mechanism to the auxiliary network and use it to dynamically generate the normalization parameters. Finally, we investigate our approaches in connectionist temporal classification (CTC)-based end-to-end speech recognition and demonstrate competitive performance in the speaker-adapted scenario.

3. Proposed methods

We first introduce the modified speaker normalization (SN) method in speech recognition. Moreover, its application in Bi-LSTM is discussed. In the following sections, the details of the proposed speaker-level and batch-level adaptive speaker normalization (ASN) are discussed.

3.1. Speaker normalization

For a neural layer with p-dimensional input feature $x_s = \{x_s^{(1)}, ..., x_s^{(i)}, ..., x_s^{(p)}\}$, where s means the feature belongs to a certain speaker s, the proposed speaker normalization for each dimension is defined as:

$$\hat{x}_s^{(i)} = \frac{x_s^{(i)} - \mathrm{E}[x_s^{(i)}]}{\sqrt{\mathrm{Var}[x_s^{(i)}]}} \tag{1}$$

where the expectation $\mathrm{E}[x_s^{(i)}]$ and variance $\mathrm{Var}[x_s^{(i)}]$ are computed over all training samples belonging to speaker s.

However, in most instances, training DNNs uses stochastic optimization. Parameter updates are on a mini-batch basis. It is impractical to use the whole set to normalize activations. Therefore, we make the simplification as BN that each mini-batch produces estimates of the mean and variance in each activation of speaker s. Eq. (1) is rewritten as:

$$\hat{x}_s^{(i)} = \frac{x_s^{(i)} - \mu_s}{\sqrt{\sigma_s^2 + \varepsilon}} \tag{2}$$

where ε is a small positive constant to prevent numerical instability, and the mini-batch speaker mean μ_s and variance σ_s are given by

$$\mu_s = \frac{1}{\sum_k \mathbf{1}[s_k = s]} \sum_k \mathbf{1}[s_k = s] x_k \tag{3}$$

and

$$\sigma_s^2 = \frac{1}{\sum_k \mathbf{1}[s_k = s]} \sum_k \mathbf{1}[s_k = s](x_k - \mu_s)^2 \tag{4}$$

where s_k denotes the speaker label of the k^th sample in the mini-batch and $\mathbf{1}[.]$ is the indicator function that evaluates to 1 when its argument holds.

However, according to BN, simply normalizing each input of a layer may change what the layer can represent. To account for this, we also introduce additional learnable parameters γ and β, which respectively scale and shift the normalized activation to enhance the representational power of the layer, leading to a layer of the form:

$$y_s^{(i)} = \gamma^{(i)} \hat{x}_s^{(i)} + \beta^{(i)} \tag{5}$$

where γ and β are parameters to be trained along with the original model parameters. By setting $\gamma^{(i)}$ to σ_s and $\beta^{(i)}$ to μ_s, the network can recover the original layer representation.

Note that SN normalizes the activations at the speaker level, which can be considered a subset of BN. When all samples of a mini-batch belong to the same speaker, SN is equal to the standard BN. However, the proposed SN may solve the drawbacks of BN to some extent. According to [33], the effectiveness of BN diminishes when the training mini-batches are small or do not consist of independent samples. For small mini-batches, the estimates of the mean and variance become less accurate. These inaccuracies are compounded with depth and reduce the quality of the resulting models. In SN, however, we estimate the mean and variance of each speaker instead of the entire training set. This allows us to make more accurate estimates from activations in smaller batches. In addition, similar to the definition of BN, SN also requires that the samples have the assumption of independent and identical distribution (i.i.d.). However, the connectionist temporal classification (CTC) [22] criterion has the same assumptions of i.i.d., which makes the proposed SN better match the scenario of CTC-based ASR.

For a standard feedforward layer in a neural network, speaker normalization can be applied easily before an arbitrary activation function as in BN. However, we are more concerned with its application in the long short-term memory (LSTM) model since LSTM is widely used to model the temporal information of acoustic features in speech recognition, especially in CTC-based speech recognition. However, according to the investigations in BN, it is quite challenging to perform normalization in such recurrent neural networks due to their complicated framework. In this paper, we apply speaker normalization to the input-to-hidden transitions of LSTMs as the researchers did in [34]. This has proven to be effective in our experiments. For a bidirectional LSTM, speaker normalization is equally applied to the forward and backward LSTM.

3.2. Adaptive speaker normalization

The scaling and shifting parameters of SN can be computed at the speaker level and batch level, respectively. For the speaker-level strategy, normalizing activations of each speaker corresponds with specific scaling and shifting parameters. For the batch-level strategy, one pair of scaling and shifting parameters are generated for all mini-batch samples.

3.2.1. Speaker level

Assume that $\mathbf{h}_t^{l-1}$ denotes the p-dimensional hidden activation of the $l-1^th$ layer at time step t. For the normalization pa-

rameter generation network, a nonlinear transformation is first applied to the normalized hidden activation as:

$$\mathbf{g}_t^{l-1} = \tanh(\mathbf{W}_g \mathbf{h}_t^{l-1} + \mathbf{b}_g) \tag{6}$$

where $\mathbf{W}_g$ is a $d_g \times p$ weight matrix and d_g is set to be less than p. This nonlinear transformation is designed to project the hidden activation to a low dimensional space. In this way, the computational cost of the auxiliary network can be greatly reduced.

We use the weighted summation of all frames belonging to speaker s to generate the normalization parameters for that speaker. The mean of all the elements in $\mathbf{g}_t^{l-1}$ can measure the importance of the t^{th} frame-level representation. With the softmax function, the attention weight for each frame of the speaker s can be calculated as:

$$\alpha_{s,t} = \frac{\exp(\text{mean}(\mathbf{g}_{s,t}^{l-1}))}{\sum_\tau \mathbf{1}[s_\tau = s] \exp(\text{mean}(\mathbf{g}_\tau^{l-1}))} \tag{7}$$

where s_τ denotes the speaker label of the τ^{th} frame in the mini-batch and $\mathbf{1}[.]$ is the indicator function that evaluates to 1 when its argument holds.

The context vector of speaker s can be easily computed with $\alpha_{s,t}$ serving as the combination weights.

$$\mathbf{c}_s = \sum_t \alpha_{s,t} \mathbf{1}[s_t = s] \mathbf{g}_t^{l-1} \tag{8}$$

Finally, the scaling and shifting parameters for speaker s are generated as a linear transformation of the context vector:

$$\gamma_s^l = \mathbf{W}_\gamma^l \mathbf{c}_s + \mathbf{b}_\gamma^l \tag{9}$$

$$\beta_s^l = \mathbf{W}_\beta^l \mathbf{c}_s + \mathbf{b}_\beta^l \tag{10}$$

Assume $\hat{\mathbf{h}}_{s,t}^{l-1}$ denotes the activation normalized by the mean and variance of speaker s. The final speaker normalized activation is given by

$$\tilde{\mathbf{h}}_{s,t}^l = \hat{\mathbf{h}}_{s,t}^{l-1} \odot \gamma_s^l + \beta_s^l \tag{11}$$

where $\odot$ denotes the elementwise product.

3.2.2. Batch level

In batch-level speaker normalization, all activations of all speakers in the mini-batch share one pair of scaling and shifting parameters indiscriminately as with the standard format. The difference is that the parameters are dynamically generated by the activations.

A straightforward idea is to use the weighted mean of all activated frames to generate the parameters. Then, Eq. (7) and Eq. (8) are rewritten as:

$$\alpha_t = \frac{\exp(\text{mean}(\mathbf{g}_t^{l-1}))}{\sum_\tau \exp(\text{mean}(\mathbf{g}_\tau^{l-1}))} \tag{12}$$

$$\mathbf{c} = \sum_t \alpha_t \mathbf{g}_t^{l-1} \tag{13}$$

where the symbols denote the same meaning as above. Then, the context vector is used to generate the scaling and shifting parameters as Eq. (9-10).

To take advantage of the discriminative information among different speakers, we further propose speaker interclass attention to combine the activations of different speakers. After the context vectors of all speakers are computed in Eq. (8). The attention weight for each speaker context vector can be calculated as:

$$\alpha_s = \frac{\exp(\text{mean}(\mathbf{c}_s))}{\sum_m \exp(\text{mean}(\mathbf{c}_m))} \tag{14}$$

where m denotes the m^{th} speaker in the mini-batch.

Then, the weighted mean of all speaker context vectors is formed with α_s serving as the combination weights:

$$\mathbf{u} = \sum_s \alpha_s \mathbf{c}_s \tag{15}$$

Finally, the weighted mean is used to generate the scaling and shifting parameters as Eq. (9-10).

4. Experiments

4.1. Dataset

We evaluated our proposed methods on an open-source Mandarin speech corpus AISHELL-1 [27]. All speech files are sampled at 16 K Hz with 16 bits. We trained our models on the training set which contains 150 hours of speech (120,098 utterances) recorded by 340 speakers. The development set contains 20 hours of speech (14,326 utterances) recorded by 40 speakers was used for early-stopping. And the test set contains 10 hours of speech (7,176 utterances) recorded by 20 speakers was used for the final evaluation. The speakers of the training set, development set, and test set are not overlapped.

4.2. SI system

We used connectionist temporal classification (CTC)-based speech recognition systems in our experiments. The input acoustic feature was 108-dimensional filter-bank features (36 filter-bank features, delta coefficients, and delta-delta coefficients) with mean and variance normalization. All neural acoustic models in the experiments had three bidirectional LSTM hidden layers with 512 LSTM cells. To improve recognition performance and training efficiency, we appended a convolutional neural network (CNN) before the LSTM layers. For the SI model, the bottom two layers were 2D convolution layers with output channels of 64 and 256. Each convolution layer was followed by a max-pooling layer with a stride of 2 in the time dimension for finally downsampling utterances to a quarter of the original length. We used a dropout rate of 0.3 for the LSTM layers to avoid overfitting.

For the output of the Mandarin acoustic model, according to the statistical information of the transcripts, we collected 4,294 Chinese characters in the training and development sets. With the special symbol blank involved, 4,295 modeling units were used for the output inference. Additionally, to further improve the performance of the SI model, the trigram language model, which was trained by using the transcription of the training set, was used in the decoding procedure.

4.3. Network training setups

The CTC-based acoustic model used the whole utterance as input, while utterances varied in length. Therefore, we sorted all the utterances of the training set in descending order by length. For the input features of the network, each utterance was represented as a sequence of frames. We set a maximum number of frames of $f_{\max}$ to control the batch size. The number of utterances included in each mini-batch was $f_{\max}/l_{\max}$, where/means rounding operation, and $l_{\max}$ denotes the length

Table 1: *The CERs (%) of SI and SN models under different learning rates. "–" denotes that the model did not converge.*

Learning rate	0.0001	0.0002	0.0004
SI	9.96	-	-
SN	9.44	9.04	8.89

of the longest utterance of the mini-batch. All utterances whose lengths were less than $l_{\max}$ were unified by a zero-padding operation. Therefore, the size of each mini-batch was variable but did not exceed $f_{\max}$.

PyTorch toolkits [35] were used in our model training process. All the model parameters were randomly initialized and updated by Adam [36]. The network was trained to minimize the CTC loss function with an initial learning rate of 0.0001. The development set was used for learning rate scheduling and early stopping. We started to halve the learning rate when the relative improvement fell below 0.004, and the training ended if the relative improvement was lower than 0.0005.

4.4. Results of SN

The standard speaker normalization described in section 3.1 was first applied to the input of all LSTM layers. We investigated the effect of normalization on the learning rate of network training, where $f_{\max}$ was set to 5,000.

Table 1 shows the character error rate (CER) of different acoustic models with and without speaker normalization under different initial learning rates. It can be seen that with a learning rate greater than 0.0001, the model without speaker normalization did not converge. The model with speaker normalization obtained a 5.2% reduction in CER under a small learning rate of 0.0001. In the case of a larger learning rate, the speaker normalized model significantly outperformed the SI model, achieving a CER of 8.89% and up to 10.7% relative improvement.

According to the analysis of Table 1, we found that standard speaker normalization enables higher learning rates and makes the model perform better. Speaker normalization makes model training more resilient to the parameter scale. Normally, large learning rates may increase the scale of layer parameters, which then amplify the gradient during backpropagation and lead to model explosion. However, with speaker normalization, backpropagation through a layer is unaffected by the scale of its parameters.

We further explored the influence of batch size during model training. Since SN uses similar ideas as BN, we compared the proposed SN with the BN algorithm. The initial learning rates for the SN and BN models were set to 0.0002. As shown in Table 2, in the case of a smaller batch size, BN greatly deteriorated the ASR performance. This is an inherent flaw of BN, as mentioned in [39]. For the proposed SN, a smaller batch size had little impact on the model performance, resulting in a comparable performance with a favorable batch size of 5,000. This shows that SN can allow the model to be trained at a smaller batch size without significantly reducing performance, thereby adapting to scenarios with sparse data.

Table 2: *The CERs (%) of the SN and BN models with different training batch sizes.*

Batch size($f_{\max}$)	2000	5000	8000
SN	10.97	9.71	9.90
BN	9.01	9.04	9.38

4.5. Results of ASN

In the ASN model, the size of the nonlinear transformation in the auxiliary network, i.e., d_g, was set to 256 to speed up the training. The hidden activation of the previous layer was used to generate the scaling and shifting parameters for the current layer. We also used dropout for the auxiliary network to improve performance. Note that the main network was more adaptable to smaller learning rates due to the influence of the auxiliary network. Therefore, the learning rate was set to 0.0001 for the ASN models, and $f_{\max}$ was set to 5000.

Table 3: *The CERs (%) of SI, SN and different ASN models*

Model	Model Size(M)	CER(%)
SI	85.82	9.96
SN	85.84	8.89
ASN-S	92.75	**8.22**
ASN-B1	92.75	8.51
ASN-B2	92.75	8.39

Table 3 summarizes the experimental results of the SI, SN and ASN models. ASN-S indicates that the scaling and shifting parameters in SN were generated at the speaker level. ASN-B1 and ASN-B2 denote the parameters that were generated by using the weighted mean of all activated frames and all speaker context vectors, respectively. As shown in Table 3, all ASN models outperformed the SN models. In batch-level ASN, since the proposed speaker interclass attention utilized the discriminative information among different speakers, ASN-B1 performed better than ASN-B2. In speaker-level ASN, since specific scaling and shifting parameters were generated for each speaker to provide more discriminative information, ASN-S further outperformed ASN-B. Finally, ASN-S achieved the best CER of 8.22%, resulting in 17.5% relative improvement over the SI model.

5. Conclusions

In this work, we propose a novel speaker normalization technique for neural acoustic model adaptation in CTC-based ASR. Unlike previous work, we use the idea of BN to normalize hidden activations at the speaker level. The method performs a layer-wise normalization for hidden activations and utilizes the mean and variance information of each speaker. Experimental results show that the proposed SN enables the model to be trained with a higher learning rate, resulting in a better performance. Additionally, SN makes model training more resilient to the batch size, which makes it possible to use it in different scenarios. Furthermore, based on SN, we propose ASN, in which the scaling and shifting parameters are dynamically generated by using an auxiliary network with an attention mechanism. We generate the parameters at the speaker level and batch level. The two strategies both outperform the standard SN, finally achieving up to a 17.5% relative reduction in CER.

6. Acknowledgements

This work was partially funded by the National Key Research and Development Program of China (Grant No. 2016YFB100 1303) and the National Natural Science Foundation of China (Grant No. U1836219).

7. References

[1] G. Hinton, L. Deng, D. Yu *et al.*, "Deep neural networks for acoustic modeling in speech recognition," *IEEE Signal processing magazine*, vol. 29, 2012.

[2] D. Yu and J. Li, "Recent progresses in deep learning based acoustic models," *IEEE/CAA Journal of Automatica Sinica*, vol. 4, no. 3, pp. 396–409, 2017.

[3] L. Hank, "Speaker adaptation of context dependent deep neural networks," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2013, pp. 7947–7951.

[4] D. Yu, K. Yao, H. Su *et al.*, "Kl-divergence regularized deep neural network adaptation for improved large vocabulary speech recognition," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2013, pp. 7893–7897.

[5] Z. Meng, J. Li, and Y. Gong, "Adversarial speaker adaptation," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 5721–5725.

[6] K. Yao, D. Yu, F. Seide *et al.*, "Adaptation of context-dependent deep neural networks for automatic speech recognition," in *2012 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2012, pp. 366–369.

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

[8] 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*, vol. 24, no. 12, pp. 2241–2250, 2016.

[9] 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.

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

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

[12] A.-r. Mohamed, T. N. Sainath, G. E. Dahl *et al.*, "Deep belief networks using discriminative features for phone recognition." in *ICASSP*, 2011, pp. 5060–5063.

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

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

[15] Y. Miao, H. Zhang, and F. Metze, "Towards speaker adaptive training of deep neural network acoustic models," in *Fifteenth Annual Conference of the International Speech Communication Association*, 2014.

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

[17] 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 *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2013, pp. 7942–7946.

[18] 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 *2014 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2014, pp. 6339–6343.

[19] T. Tan, Y. Qian, M. Yin *et al.*, "Cluster adaptive training for deep neural network," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2015, pp. 4325–4329.

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

[21] J. Li, R. Zhao, Z. Chen *et al.*, "Developing far-field speaker system via teacher-student learning," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 5699–5703.

[22] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks," in *Proceedings of the 23rd international conference on Machine learning*. ACM, 2006, pp. 369–376.

[23] F. Weninger, J. Andrés-Ferrer, X. Li, and P. Zhan, "Listen, attend, spell and adapt: Speaker adapted sequence-to-sequence asr," *arXiv preprint arXiv:1907.04916*, 2019.

[24] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv preprint arXiv:1502.03167*, 2015.

[25] T. Kim, I. Song, and Y. Bengio, "Dynamic layer normalization for adaptive neural acoustic modeling in speech recognition," *arXiv preprint arXiv:1707.06065*, 2017.

[26] F. Ding, W. Guo, L. Dai, and J. Du, "Attentive batch normalization for lstm-based acoustic modeling of speech recognition," *arXiv preprint arXiv:2001.00129*, 2020.

[27] H. Bu, J. Du, X. Na *et al.*, "Aishell-1: An open-source mandarin speech corpus and a speech recognition baseline," in *2017 20th Conference of the Oriental Chapter of the International Coordinating Committee on Speech Databases and Speech I/O Systems and Assessment (O-COCOSDA)*. IEEE, 2017, pp. 1–5.

[28] F.-H. Liu, R. M. Stern, X. Huang, and A. Acero, "Efficient cepstral normalization for robust speech recognition," in *Proceedings of the workshop on Human Language Technology*. Association for Computational Linguistics, 1993, pp. 69–74.

[29] O. Viikki and K. Laurila, "Cepstral domain segmental feature vector normalization for noise robust speech recognition," *Speech Communication*, vol. 25, no. 1-3, pp. 133–147, 1998.

[30] Y. Obuchi and R. M. Stern, "Normalization of time-derivative parameters using histogram equalization," in *Eighth European Conference on Speech Communication and Technology*, 2003.

[31] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," *arXiv preprint arXiv:1607.06450*, 2016.

[32] L. Sarı, S. Thomas, and M. A. Hasegawa-Johnson, "Learning speaker aware offsets for speaker adaptation of neural networks," *Proc. Interspeech 2019*, pp. 769–773, 2019.

[33] S. Ioffe, "Batch renormalization: Towards reducing minibatch dependence in batch-normalized models," in *Advances in neural information processing systems*, 2017, pp. 1945–1953.

[34] C. Laurent, G. Pereyra, P. Brakel *et al.*, "Batch normalized recurrent neural networks," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2016, pp. 2657–2661.

[35] A. Paszke, S. Gross, S. Chintala *et al.*, "Automatic differentiation in pytorch," 2017.

[36] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.