Batch Normalization based Unsupervised Speaker Adaptation for Acoustic Models

Jiangyan Yi*‡ and Jianhua Tao*†‡
* National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing, China
† CAS Center for Excellence in Brain Science and Intelligence Technology, Institute of Automation, Chinese Academy of Sciences
‡ University of Chinese Academy of Sciences, Beijing, China
E-mail: jiangyan.yi@nlpr.ia.ac.cn, jhtao@nlpr.ia.ac.cn

***Abstract*—This paper proposes a simple yet effective unsupervised speaker adaptation approach for batch normalization based deep neural network acoustic models. The basic idea of this approach is to recompute means and variances in all batch normalization layers over the test data for every speaker. Thus the distribution of the test data can be close to the training data. This approach doesn't need to adjust any trainable parameters of the acoustic model. Experiments are conducted on CHiME-3 datasets. The results show that the proposed adaptation obtains improvement on the real test set by 2.17 % relative average word error rate (WER) reduction when compared with the scaling and shifting factors (SSF) adaptation. Combining our proposed MV adaptation with the SSF adaptation obtains further improvement.**

I. Introduction

Automatic speech recognition systems have achieved significant improvement through using deep neural network (DNN) based acoustic models [1], [2], [3]. However, the performance will degrade dramatically if there exists the mismatch between test and training conditions. The mismatch can be caused by one or many of the factors, such as speaker etc. A lot of efforts have been made to solve this problem.

One of the approaches is to train the acoustic model using multi-condition training technique by data augmentation [4]. However, this approach requires a large amount of various training data that are difficult to obtain. The other is to narrow the distribution gap between the test and training data by adaptation techniques. There are many methods proposed to perform speaker adaptation. These methods can be roughly classified into three categories: feature compensation, auxiliary features and model compensation.

One technique for feature compensation is maximum likelihood linear regression (MLLR) transforms or feature-space variant (fMLLR) [5]. The other technique is linear input network (LIN) which is developed for neural networks based models [6]. The LIN defines an additional speaker-dependent layer between the input features and the first hidden layer.

Auxiliary features adaptation is to augment the input acoustic features by utilizing additional speaker-specific features at both training and test stage. There are about three typical methods: i-vectors [7], [8], speaker-specific bottleneck features [9] and speaker code [10].

Model compensation adaptation mainly relies on adjusting the parameters of DNN based acoustic models. Adapting the parameters of the models results in producing a large amount of speaker-dependent parameters [11], [12]. Some works also are proposed to reduce the adaptation parameters. These methods are performed by adding a linear speaker-dependent layer in the neural networks, such as linear hidden network (LHN) [13], linear output network (LOH) [14], [15] etc. Recently, Pawel et al. propose a learning hidden unit contributions (LHUC) to linearly recombine hidden units for a speaker [16]. Xie et al. [17] further propose a bayesian LHUC adaptation method to improve the performance. Meng et al. [18] propose to use adversarial training to perform speaker adaptation. Wang et al. [19] propose a scaling and shifting factors (SSF) adaptation for batch normalized acoustic models. The SSF adaptation is performed by retraining scaling and shifting factors in all batch normalization (BN) [20] layers for every speaker. The SSF adaptation has achieved the state-of-the-art performance on CHiME-3 datasets [21]. However, one problem of the SSF adaptation is that it uses the means and variances computed over the training data at the test stage. Thus the test data can not better match the distribution of the training data. The other problem is that it will mislead the acoustic model during adaptation when the first-pass decoding results have too many errors.

Therefore, this paper proposes a simple yet effective means and variances (MV) adaptation for batch normalization based DNN acoustic models. The MV adaptation is an unsupervised speaker adaptation. The key idea of the MV adaptation is to recalculate means and variances in all BN layers using the test data for each speaker. Thus the distribution of the test data can be close to the training data. Furthermore, the MV adaptation doesn't require to adjust any trainable parameters of the acoustic model. It only needs little computational cost to recompute means and variances over the test data. So it will not mislead the acoustic model during adaptation. Experiments are conducted on CHiME-3 datasets. The results show that the proposed MV adaptation outperforms the SSF adaptation proposed by [19], with a further average word error rate (WER) reduction of 2.17% relative on the real test set.

The rest of this paper is organized as follows. Section 2 introduces batch normalization briefly. The proposed MV

adaptation is described in Section 3. Experiments are presented in Section 4. The results are discussed in Section 5. This paper is concluded in Section 6.

II. Batch Normalization

Batch normalization (BN) is proposed to train deep neural networks by Ioffe and Szegedy [20]. It has become a standard component in neural networks for many tasks [22], [23]. It not only yields a substantial speedup in training, but also improves the performance.

The BN layer is utilized to address the problem of internal covariate shifting. At first, the activations of each hidden layer are normalized by means and variances computed over each training mini-batch. Then the normalized activations are linearly transformed by scaling factors γ and shifting factors β before applying non-linear functions. The normalization is applied to each activation independently. Formally, given the input to a BN layer $X \in R^{m \times n}$, where m denotes the mini-batch size, and n indicates the input dimension, an input value x_j is transformed by the BN layer:

$$y_j = \gamma_j \frac{x_j - \mu_j^{train}}{\sigma_j^{train}} + \beta_j \tag{1}$$

$$\mu_j^{train} = E[X_{.j}] \tag{2}$$

$$\sigma_j^{train} = \sqrt{Var[X_{.j}]} \tag{3}$$

where j is the dimension of the input, $j \in \{1...n\}$; y_j is an output value transformed by the BN layer; γ_j and β_j are the trainable parameters; mean μ_j^{train} and variance σ_j^{train} are computed over each training mini-batch $X_{.j}$.

This transformation makes the input distribution of each hidden layer stable across different mini-batches. At the test stage, the global statistics mean μ_j^{train} and variance σ_j^{train} computed over all the training data are used to normalize the test data.

III. Proposed means and variances adaptation

The proposed means and variances (MV) adaptation is an unsupervised speaker adaptation. It is performed on batch normalization based acoustic models. The basic idea of the adaptation is that the global statistics mean μ_j^{train} and variance σ_j^{train} are just replaced with a re-computation of the population mean μ_j^{test} and variance σ_j^{test} over the test data for each speaker. Thus it is more easy to generalize training data distributions to test data distributions.

The MV adaptation is to recalculate means and variances in all BN layers using the test data for each speaker. So the test data can better match the distribution of the training data. Thus for one BN layer, we can define

$$\hat{y}_j = \gamma_j \frac{\hat{x}_j - \mu_j^{test}}{\sigma_j^{test}} + \beta_j \tag{4}$$

where $\hat{x}_j$ is an input value of the BN layer; $\hat{y}_j$ is an adapted value of the BN layer output; denotes the j^{th} dimension of the input data; μ_j^{test} and σ_j^{test} are the mean and variance computed over all the test data for each speaker; the scaling factor γ_j and shifting factor β_j are the trainable parameters obtained at the training stage.

From equation (4), we can see that replacing μ_j^{train} and σ_j^{train} with μ_j^{test} and σ_j^{test} is equivalent to narrow the distribution gab between the test data and the training data. When performing speaker adaptation, the proposed MV adaptation is conducted in each BN layer for every speaker. The speaker-independent batch normalization based acoustic models are trained at first.

At the training stage, μ_j^{train} and σ_j^{train} are computed for each mini-batch. The samples in one mini-batch are from the same speaker during training. γ_j and β_j are the trainable parameters.

At the adaptation stage, the population μ_j^{test} and σ_j^{test} in every BN layer are recomputed over the test data for each speaker. All trainable parameters of the speaker-independent batch normalized acoustic model are frozen.

At the test stage, the recomputed μ_j^{test} and σ_j^{test} are directly used to perform forward pass algorithm for each speaker by equation (4). In addition, all the trainable parameters obtained at the training stage are used at the test stage, such as γ_j and β_j.

IV. Difference between MV and SSF adaptation

The MV adaptation is different from the scaling and shifting factors (SSF) adaptation proposed by Wang et al. [19].

The SSF adaptation is proposed to retrain the scaling factor and shifting factor at each hidden layer in batch normalized acoustic models for every speaker. It needs to adapt the acoustic model by adjusting the scaling and shifting factors using first-pass decoding results. In addition, the means and variances computed over the training data are reused at the test stage.

The proposed MV adaptation doesn't need to retrain any trainable parameters of the acoustic model. So it doesn't need the first-pass decoding results. Thus when performing adaptation, it will not mislead the model. Moreover, the MV adaptation replaces the means and variances computed at the training stage with the means and variances recalculated over the test data for each speaker. This may better adapt an acoustic model to a target speaker.

V. Experiments

Our proposed MV adaptation is evaluated by a series of experiments in this section.

A. CHiME-3 datasets

Our experiments are conducted on CHiME-3 datasets [21]. The CHiME-3 datasets consist of real and simulated six-channel audio data in four noisy environments, such public transport (BUS), cafe (CAF), and pedestrian area (PED), street junction (STR). A tablet device with six microphones is designed for collecting real audio recording. The simulated recordings have been generated by artificially mixing clean speech data with four noisy environments. The clean speech corpus includes only read speech from the WSJ0 corpus.

TABLE I
THE WERS(%) OF SPEAKER INDEPENDENT MODELS USING 3-GRAM LM FOR DECODING ON THE DEVELOPMENT (DEV.) SET AND TEST SET.

Dataset	Approaches	Simu					Real					Avg.
		AVG	BUS	CAF	PED	STR	**AVG**	BUS	CAF	PED	STR	
Dev.	DNN	7.62	6.30	9.44	6.84	7.91	7.62	8.63	8.01	6.52	7.33	7.62
	DNN + BN	6.85	5.47	8.39	6.30	7.23	6.92	8.02	7.09	5.80	6.77	6.88
Test	DNN	8.72	7.06	10.20	8.78	8.85	11.20	13.65	11.56	9.87	9.71	9.96
	DNN + BN	8.00	6.48	9.45	7.96	8.09	10.22	12.88	10.53	8.37	9.10	9.11

The training set comprises 1600 real and 7138 simulated utterances, which amount to 18 hours of speech. There are a total of 4 speakers in the real data, and 83 speakers in the simulated data. The development set consists of 1640 (4*410) real and 1640 (4*410) simulated utterances from 4 unseen speakers. Similarly, the test set consists of 1320 (4*330) real and 1320 (4*330) simulated utterances from another 4 unseen speakers. Speaker labels can be used for speaker adaptation in CHiME-3 challenge [21].

B. Experimental setup

Our experiments are conducted using Kaldi speech recognition toolkit [24]. In all experiments, we use enhanced single-channel signals to train acoustic models, perform adaptation and decoding. The enhanced single-channel signals are generated by the generalized eigenvalue (GEV) beamformer toolkit [25]. The code of the GEV toolkit is public available[1]. The real and simulated six-channel audio data are enhanced to single-channel data on the training set, the development set and the test set respectively. We use the real and simulated enhanced data to train acoustic models. The total training data has 8738 (1600+7138) utterances about 18 hours. The total development data has 3280 (1640+1640) utterances. The total test data has 2640 (1320+1320) utterances. The training data is used to update the trainable parameters. The hyper-parameters are selected on the development data.

We follow the officially released Kaldi recipe to build a Gaussian mixture model hidden Markov model (GMM-HMM) model at first. The features are extracted with a 25-ms sliding window with a 10-ms shift. Input features for the GMM-HMM model consist of 13-dimensional MFCC and their delta and delta-delta parameters. The GMM-HMM model has 2024 senones. We use the GMM-HMM model to generate frame-level state alignments for DNN models.

All the DNN models use a sliding context window of 11 consecutive speech frames as inputs. Each frame is represented by 40-dimensional log mel-filter bank (FBANK) features plus their delta and delta-delta. All the DNN models are trained using stochastic gradient descent (SGD) with a momentum term to minimize the cross-entropy criterion. The training terminates on the development set with a little improvement.

The 3-gram language model (LM) is provided by the CHiME-3 challenge. The vocabulary of this language model is 5K. At the decoding stage, decoding is performed using fully composed 3-gram weighted finite state transducers.

[1]https://github.com/fgnt/nn-gev

C. Speaker independent acoustic models

Our DNN based acoustic model is speaker-independent. The DNN model has 7 hidden layers and each hidden layer has 2048 sigmoid units. The output layer of the DNN model has 2024 senones in total. The momentum is set to 0.9. The initial learning rate is set to 0.008. The mini-batch size is 256. Sentence-level mean normalization is used for the input features. Then global mean-variance normalization is applied to the inputs. After training, the 3-gram LM is used for decoding. The average WERs of the Baseline DNN model for all speakers on the real and simulated sets are listed in Table I.

The batch normalized acoustic model is also a speaker-independent model, where additional BN layer is applied to each hidden layer in the DNN model. We use the same configuration of the Baseline DNN model to train the BN-DNN model. The initial learning rate is 0.001. The means and variances are computed for each mini-batch. The mini-batch size is 256. The samples in one mini-batch are from the same speaker during training. The results using 3-gram LM for decoding are reported in Table I. From the results, we can find that the BN model outperforms the DNN model from 11.20% to 10.22% average WER on the real test set.

D. Speaker adaptation

In our experiments, the proposed MV adaptation is compared with the other adaptation methods proposed by previous researchers. The adaptation is performed for every speaker on the development set and the test set. There are about 410 utterances for each speaker in the development set and 330 utterances for each speaker in the test set. We use all the utterances of each speaker for adaptation. The 3-gram LM is utilized to obtain the first-pass decoding results for other adaptation methods. It is also used to decode all the acoustic models. The adaptation methods are as follows.

LHN is proposed by Gemello et al. [13]. It is performed by inserting an additional linear transformation layer after the last hidden layer of the Baseline DNN model for each speaker. At the adaptation stage, we only adapt the linear transformation layer using all the utterances of every speaker.

LIN [6] is performed by adding an extra linear transformation layer between the input features and the first hidden layer of the Baseline DNN model. The linear layer is finetuned for each speaker.

SSF is the scaling and shifting factors adaptation proposed by Wang et al. [19]. This method is performed by finetuning

TABLE II
THE WERS(%) OF SPEAKER ADAPTATION USING 3-GRAM LM FOR DECODING ON THE DEVELOPMENT (DEV.) SET AND TEST SET.

Dataset	Approaches	Simu					Real					Avg.
		AVG	BUS	CAF	PED	STR	**AVG**	BUS	CAF	PED	STR	
Dev.	LIN	5.81	4.74	7.59	5.38	5.51	5.84	6.98	5.92	4.59	5.85	5.82
	LHN	5.65	4.42	7.45	5.21	5.53	5.58	6.49	5.81	4.54	5.49	5.62
	SSF	4.98	4.16	6.68	4.29	4.79	4.60	5.53	4.34	3.75	4.76	4.79
	MV (Ours)	**4.81**	**4.01**	**6.56**	**4.12**	**4.53**	**4.42**	**5.41**	**4.16**	**3.58**	**4.52**	**4.61**
	MV (Ours) + SSF	**4.49**	**3.72**	**6.23**	**3.81**	**4.21**	**4.10**	**5.09**	**3.91**	**3.24**	**4.17**	**4.30**
Test	LIN	6.93	5.62	8.25	6.75	7.09	8.78	10.87	9.04	7.43	7.79	7.86
	LHN	6.69	5.35	7.83	6.63	6.93	8.55	10.51	8.74	7.27	7.66	7.62
	SSF	5.73	4.33	6.85	5.68	6.05	7.37	9.20	7.53	6.20	6.56	6.55
	MV (Ours)	**5.58**	**4.26**	**6.71**	**5.52**	**5.81**	**7.21**	**9.02**	**7.36**	**6.06**	**6.41**	**6.39**
	MV (Ours) + SSF	**5.24**	**3.84**	**6.32**	**5.27**	**5.54**	**6.87**	**8.71**	**6.95**	**5.73**	**6.09**	**6.06**

the scaling and shifting factors of the BN model. All the utterances of each speaker are used for adaptation.

MV is our proposed method which performed on the BN model. The population means and variances are recomputed over the development set or the test data for each speaker.

We implement the above methods based on the Kaldi toolkit. The results are reported in Table II. From the results, we can see that all the adaptation methods outperform the speaker-independent DNN model and BN model obviously. This is because of challenging speaking styles of the test speakers in the CHiME-3 challenge. The LHN adaptation outperforms the LIN adaptation. It is consistent with the conclusions in [13]. The SSF adaptation outperforms all the above-mentioned adaptation methods. Our proposed MV adaptation obtains 2.67% and 2.17% relative average WER reduction on the simulated set and the real test set over the SSF adaptation respectively. Combining our proposed MV adaptation with the SSF adaptation achieves the best performance both on the development set and the test set.

VI. DISCUSSIONS

The above experiments show that our proposed method is simple yet effective. Some interesting observations are made as follows.

The proposed MV adaptation obtains obvious improvement when compared with LHN and LIN adaptation. It indicates that all hidden layers of the DNN model are influenced by speaker shift. It is not enough to perform speaker adaptation only on one hidden layer of the model.

Our proposed MV adaptation also outperforms the SSF adaptation. There are two main reasons. One reason is that the MV adaptation doesn't need the first-pass decoding results. Thus when performing adaptation, it will not mislead the model. The other reason is that the MV adaptation is to recompute the means and variances over the test data for each speaker. This may make the acoustic model to better match the distribution of a target speaker. However, the SSF adaptation will mislead the acoustic model during adaptation when the first-pass decoding results have some errors. Furthermore, the SSF adaptation uses the means and variances computed over the training data at the test stage. Thus the test data can not better match the distribution of the training data.

Combining the MV adaptation with the SSF adaptation can achieve better performance It is because that this method utilizes the strengths of the MV adaptation and the SSF adaptation. However, This method will take more computational cost to perform adaptation when compared to the MV adaptation.

In short, it is more easily to generalize training data distributions to test data distributions just replacing with a recomputation of the means and variances over the test data for each speaker. It means that the means and variances of BN layer contain the characteristics of the speaker. Thus it makes the MV adaptation simpler yet more effective.

VII. CONCLUSIONS

This paper proposes a simple yet effective unsupervised speaker adaptation for batch normalization based DNN acoustic models. The proposed MV adaptation doesn't require to adjust any trainable parameters of the acoustic model. It only needs few computational cost to recompute means and variances. So it will not mislead the acoustic model when performing adaptation. Experiments are conducted on CHiME-3 corpus. The results show that the proposed MV adaptation obtains improvement on the real test set by 2.17% relative average WER reduction over the SSF adaptation. In future work, we plan to apply this adaptation to environment and accent adaptation tasks.

ACKNOWLEDGMENT

This work is supported by the National Key Research & Development Plan of China (No.2017YFC0820602), the National Natural Science Foundation of China (NSFC) (No.61425017, No.61831022, No.61773379, No.61603390), the Strategic Priority Research Program of Chinese Academy of Sciences (No.XDC02050100), and Inria-CAS Joint Research Project (No.173211KYSB20170061).

REFERENCES

[1] G. Hinton, L. Deng, D. Yu, G.E. Dahl, A.R. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, and T.N. Sainath. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Processing Magazine*, 29(6):82–97, 2012.

[2] W. Xiong, J. Droppo, X. Huang, F. Seide, M. Seltzer, A. Stolcke, D. Yu, and G. Zweig. The microsoft 2016 conversational speech recognition system. In *ICASSP*, pages 5255–5259, 2017.

[3] G. Saon, T. Sercu, S. Rennie, and H.K.J. Kuo. The ibm 2016 english conversational telephone speech recognition system. 2016.
[4] K. Zmołłkov, M. Karafit, K. Vesely, M. Delcroix, S. Watanabe, L. Burget, and J. Cernocky. Data selection by sequence summarizing neural network in mismatch condition training. In *INTERSPEECH*, pages 2354–2358, 2016.
[5] M.J.F. Gales. Maximum likelihood linear transformations for hmm-based speech recognition. *Computer Speech and Language*, 12(2):75–98, 1997.
[6] R. Gemello, F. Mana, and D. Albesano. Linear input network based speaker adaptation in the dialogos system. In *IEEE International Joint Conference on Neural Networks Proceedings*, pages 2190–2195 vol.3, 1998.
[7] M. Karafiat, L. Burget, P. Matejka, and O. Glembek. ivector-based discriminative adaptation for automatic speech recognition. *Speech Segmentation*, pages 152–157, 2011.
[8] A. Senior and I. Lopez-Moreno. Improving dnn speaker independence with i-vector inputs. In *ICASSP*, pages 225–229, 2014.
[9] T. Ochiai, M. Delcroix1, K. Kinoshita1, and Ogawa A. Cumulative moving averaged bottleneck speaker vectors for online speaker adaptation of cnn-based acoustic models. In *ICASSP*, pages 5175–5179, 2017.
[10] 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*, 22(12):1713–1725, 2014.
[11] H. Liao. Speaker adaptation of context dependent deep neural networks. In *ICASSP*, pages 7947–7951, 2013.
[12] 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*, pages 7893–7897, 2013.
[13] R. Gemello, F. Mana, S. Scanzio, P. Laface, and R. De Mori. Linear hidden transformations for adaptation of hybrid ann/hmm models. *Speech Communication*, 49(10):827–835, 2007.
[14] B. Li and K.C. Sim. Comparison of discriminative input and output transformations for speaker adaptation in the hybrid nn/hmm systems. In *INTERSPEECH*, pages 526–529, 2010.
[15] K. Yao, D. Yu, F. Seide, H. Su, L. Deng, and Y. Gong. Adaptation of context-dependent deep neural networks for automatic speech recognition. In *Spoken Language Technology Workshop*, pages 366–369, 2012.
[16] P. Swietojanski and S. Renais. Sat-lhuc: Speaker adaptive training for learning hidden unit contributions. In *ICASSP*, pages 197–202, 2016.
[17] 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*, pages 5711–5715, 2019.
[18] Z. Meng, J. Li, and Y. Gong. Adversarial speaker adaptation. In *ICASSP*, pages 5721–5725, 2019.
[19] Zh. Wang and D. Wang. Unsupervised speaker adaptation of batch normalized acoustic models for robust asr. In *ICASSP*, pages 4890–4894, 2017.
[20] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *Computer Science*, 2015.
[21] B. Jon, M. Ricard, V. Emmanuel, and W. Shinji. The third chime speech separation and recognition challenge: Dataset, task and baselines. In *ASRU*, pages 436–443, 2015.
[22] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016.
[23] Y. Zhang, W. Chan, and N. Jaitly. Very deep convolutional networks for end-to-end speech recognition. In *ICASSP*, pages 4845–4849, 2017.
[24] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, and M. Hannemann. The kaldi speech recognition toolkit. In *ASRU*, 2011.
[25] J. Heymann, L. Drude, A. Chinaev, and R. Haeb-Umbach. Blstm supported gev beamformer front-end for the 3rd chime challenge. In *ASRU*, pages 444–451, 2015.