Ensemble Speaker Modeling using Speaker Adaptive Training Deep Neural Network for Speaker Adaptation

Sheng Li[1], *Xugang Lu*[2], *Yuya Akita*[1], *Tatsuya Kawahara*[1]

[1]School of Informatics, Kyoto University, Sakyo-ku, Kyoto 606-8501, Japan

[2]National Institute of Information and Communications Technology, Kyoto, Japan

lisheng@ar.media.kyoto-u.ac.jp

Abstract

In this paper, we introduce an ensemble speaker modeling using a speaker adaptive training (SAT) deep neural network (SAT-DNN). We first train a speaker-independent DNN (SI-DNN) acoustic model as a universal speaker model (USM). Based on the USM, a SAT-DNN is used to obtain a set of speaker-dependent models by assuming that all other layers except one speaker-dependent (SD) layer are shared among speakers. The speaker ensemble matrix is created by concatenating all of the SD neural weight matrices. With matrix factorization technique, an ensemble speaker subspace is extracted. When testing, an initial model for each target speaker is selected in this ensemble speaker subspace. Then, adaptation is carried out to obtain the final acoustic model for testing. In order to reduce the number of adaptation parameters, low-rank speaker subspace is further explored. We test our algorithm on lecture transcription task. Experimental results showed that our proposed method is effective for unsupervised speaker adaptation.

Index Terms: speaker adaptation, deep neural networks, ensemble modeling, lecture transcription

1. Introduction

Speaker adaptation is very important for achieving high recognition performance in automatic speech recognition (ASR). Great successes have been achieved in the traditional GMM-HMM framework by using speaker adaptation techniques. However, these techniques cannot be applied to the DNN-HMM framework straightforwardly since these two frameworks are fundamentally different. Two typical approaches have been proposed for DNN-HMM adaptation framework. In one approach, which is analogous to fMLLR and global MLLR techniques, a linear transformation is applied to weights of links to input and/or output nodes for DNN adaptation [1, 2, 3, 4]. Another approach is inspired by speaker adaptive training (SAT) in GMM-HMM framework. Speaker adaptive training deep neural network (SAT-DNN) has been proposed to achieve speaker normalization at training time [5, 6, 7]. In order to explicitly incorporate speaker information in adaptation, features related to speaker characteristics, for example i-vectors, are integrated as a specific network layer for SAT [5, 6]. In another SAT-DNN approach for speaker adaptation, adopting the ideas from multi-task learning, one layer is specified as a speaker-dependent (SD) layer and all other layers are shared by all speakers in the DNN architecture [7].

No matter what techniques are used in adaptation, model generalization problem must be taken into consideration. DNN model has a huge number of free parameters, and thus is easy to fall into overfitting with limited adaptation data. Regularization technique has been proposed to avoid overfitting in adaptation and adaptive training. For example, Yu et al. [8] proposed a method to control the adaptation procedure by monitoring the KL-divergence from the baseline model. Liao et al. [9] introduced L2-regularization to effectively control speaker adaptation. These adaptation techniques try to update model parameters with regularization constraints that keep the updated models from deviating too far away from the "good" model. In most studies, the "good" model is a speaker-independent DNN (SI-DNN) model, i.e., an average model for all speakers. Recently, a theoretically attractive approach for DNN adaptation has been proposed based on low-rank approximation techniques for matrices. For example, a singular value decomposition (SVD)-based low-rank matrix adaptation method for DNN is proposed [10]. The SVD based low-ranking matrix approximation [11] can prune the vast number of parameters to obtain a compressed DNN model without accuracy loss. In their approach, however, the initial model in adaptation is an SI-DNN model, i.e., an averaged model for all speakers.

Doing adaptation either with a regularization technique or with a low-rank approximation technique based on an averaged acoustic model may not be good enough due to the large variations of the speaker acoustic space in real applications. In ensemble speaker and environment modeling technique [12, 13, 14], when choosing an initial model for adaptation, it is possible to choose one single speaker or a subgroup of speakers for adaptation. Inspired by this, we propose an ensemble speaker modeling framework for speaker adaptation using SAT-DNN.

In the proposed framework, we first train an SI-DNN acoustic model as a universal speaker model (USM). Based on this USM, a SAT-DNN architecture is used to obtain a set of SD models by making all layers shared by all speakers except one SD layer. A speaker ensemble matrix is composed by concatenating all of the SD neural weight matrices. By applying SVD to the ensemble matrix, a full-rank or low-rank speaker subspace representation is extracted. Every SD weight matrix can be approximated in this speaker subspace. When testing, we select an initial model for each target speaker in this speaker subspace. And then, the adapted model is used for testing. We apply our algorithm to unsupervised speaker adaptation for lecture speech transcription. Experimental results show that our proposed method is effective for unsupervised speaker adaptation.

The rest of this paper is organized as follows. Section 2 introduces our proposed ensemble speaker modeling and adaptation scheme. Section 3 shows the implementation and evaluation of the proposed scheme, and conclusion is given in section 4.

2. Ensemble speaker modeling using speaker adaptive training DNN

Rather than using only one SI-DNN model as an initial model in adaptation, we prepare many SD-DNN models, and choose the best one among them as an initial model for adaptation. The basic procedure is as follows:

a) Train a USM, i.e., SI-DNN.
b) Taking the USM as an initial model, train speaker-dependent models, i.e., SD-DNNs. For training, a multi-task learning architecture for SAT-DNN is adopted.
c) Factorize the speaker-dependent weight matrices using SVD and obtain speaker-specific coefficient matrices. Then, perform low-rank matrix approximation to reduce the number of adaptation model parameters.
d) Perform adaptation for a testing speaker by picking up an initial model in the speaker subspace.

In the following subsections, each stage of the procedure is described in details.

2.1. Multi-task learning architecture for SAT-DNN

In multi-task learning, we suppose that some model parameters are shared by all tasks and each task has its own task-dependent parameters. It is shown that this multi-task learning strategy achieves better generalization than single-task learning strategy in various task domains such as phone recognition and multilingual speech recognition [15, 16]. The SAT-DNN proposed in [7] can be regarded as a multi-task learning. In NICT-SAT-DNN [7], the DNN architecture is configured as shown in Figure 1. All of the DNN layers are shared among speakers except one SD layer. The parameters in the SD layer are updated only for a specific speaker while the parameters for all of the shared layers are updated for all speakers. Explicitly specifying one layer as an SD layer in training makes training focus much more on speaker adaptation in DNN. In speaker adaptive training, the initial model parameters are set as the model parameters of an SI-DNN model.

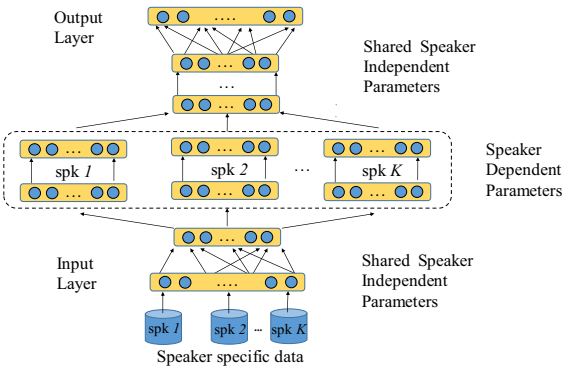

Figure 1: *Multi-task learning architecture for SAT-DNN.*

2.2. Ensemble speaker matrix factorization

From SAT-DNN introduced in section 2.1, we obtain a set of SD-DNN models (with shared neural weight matrices). Suppose the SD-DNN model is represented as the neural weight matrix of the SD layer as

$$\left\{\mathbf{W}_{sd}^{i} \in R^{m*n}, i = 1,2,...,K\right\}$$

where K is the total number of speakers, m and n are the numbers of neurons for input and output, respectively, of the SD layers. The ensemble speaker matrix is composed by concatenating these matrices as

$$\mathbf{W}_{sd} \overset{\Delta}{=} [\mathbf{W}_{sd}^{1}, \mathbf{W}_{sd}^{2}, ..., \mathbf{W}_{sd}^{K}] \in R^{m*l}, l = n*K$$

Based on SVD matrix decomposition [17, 18], it is decomposed as

$$\mathbf{W}_{sd} = \mathbf{U} * \mathbf{S} * [(\mathbf{V}_{sd}^{1})^{T}, (\mathbf{V}_{sd}^{2})^{T}, ..., (\mathbf{V}_{sd}^{K})^{T}] \quad (1)$$

In this equation, $\mathbf{U} \in R^{m*n}$ is the left singular matrix, $\mathbf{S} \in R^{n*n}$ is a diagonal matrix with elements as singular values. $(\mathbf{V}_{sd}^{i})^{T} \in R^{n*n}$ is the speaker coefficient matrix of the i-th speaker that satisfies:

$$\mathbf{W}_{sd}^{i} = \mathbf{U} * \mathbf{S} * (\mathbf{V}_{sd}^{i})^{T} \quad (2)$$

In DNN, this matrix factorization can be implemented as in Figure 2.

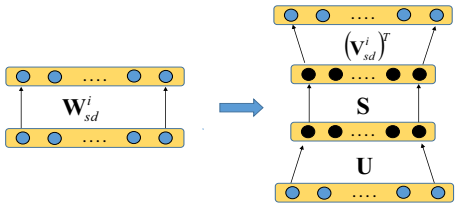

Figure 2: *Before (left) and after (right) matrix factorization in one layer of DNN.*

In this figure, black balls represent linear response neurons. The total transform effect of the factorized matrix is the same as only using one transform matrix $\mathbf{W}_{sd}^{i}$ [11].

2.3. Low-rank matrix approximation

In order to reduce the number of model parameters in adaptation, low-rank approximation techniques are used. The ensemble speaker matrix can be approximated in a low-rank form as:

$$\mathbf{W}_{sd} \approx \widetilde{\mathbf{U}} * \widetilde{\mathbf{S}} * [(\widetilde{\mathbf{V}}_{sd}^{1})^{T}, (\widetilde{\mathbf{V}}_{sd}^{2})^{T}, ..., (\widetilde{\mathbf{V}}_{sd}^{K})^{T}] \quad (3)$$

where $\widetilde{\mathbf{S}} \in R^{d*d}$ is a diagonal matrix with top d largest singular values of $\mathbf{S}$, and $\widetilde{\mathbf{U}} \in R^{m*d}$ is a matrix with column vectors corresponding to singular values in $\widetilde{\mathbf{S}}$.

$(\widetilde{\mathbf{V}}_{sd}^{i})^{T} \in R^{d*n}$ is the speaker coefficient matrix and $d << \min\{m,n\}$ is the low-rank value of the matrix. The advantage of using this low-rank approximation is that we can generate a small bottleneck layer in implementation which may make the model much more robust (or with better generalization ability) than using the full-rank matrix [10].

2.4. Adaptation on SAT-DNN ensemble models

Under the SAT-DNN ensemble model framework, many adaptation strategies can be applied. In this study, we introduce two adaptation algorithms.

2.4.1. Updating speaker coefficient matrix of the SD layer

As we have shown in section 2.2, one direct physical explanation of the ensemble matrix factorization (refer to Eqs. (1) and (2)) is that: **U*S** is the weighted speaker subspace bases and $\left(\mathbf{V}_{sd}^{i}\right)^{T}$ is the speaker coefficient matrix. For a test speaker, we can regard the adapted model as one point in this speaker subspace, and then the weight matrix for the SD layer of the target speaker should be in the form of

$$\mathbf{W}_{sd}^{test} = \mathbf{U}*\mathbf{S}*\left(\mathbf{V}_{sd}^{test}\right)^{T} \tag{4}$$

This $\left(\mathbf{V}_{sd}^{test}\right)^{T}$ needs to be estimated in the adaptation model. This matrix is a function of training speakers as:

$$\mathbf{V}_{sd}^{test} \overset{\Delta}{=} \mathbf{F}\left(\mathbf{V}_{sd}^{1}, \mathbf{V}_{sd}^{2}, ..., \mathbf{V}_{sd}^{K}; \Theta\right) \tag{5}$$

where **F(.)** is a function matrix with parameter Θ. It is difficult to obtain the solution if there is no prior knowledge of this F(**.**). If we suppose this mapping function is a linear regression of all training speakers, it is formulated as (for simplicity, the bias in linear regression model is omitted):

$$\mathbf{V}_{sd}^{test} \overset{\Delta}{=} \sum_{i=1}^{K} \mathbf{A}_i \mathbf{V}_{sd}^{i} \tag{6}$$

where $\mathbf{A}_i$ is a regression matrix. If $\mathbf{A}_i$ is an identity matrix ($\mathbf{A}_i = \mathbf{I}$) for i=1, 2 … K, the adaptation model is the average of all training speakers as

$$\mathbf{V}_{sd}^{test} \overset{\Delta}{=} \overline{\mathbf{V}}_{sd}^{train} = \frac{1}{K}\sum_{i=1}^{K} \mathbf{V}_{sd}^{i} \tag{7}$$

If $\mathbf{A}_i$=**0** for all i except when $i \neq best$, then

$$\mathbf{V}_{sd}^{test} \overset{\Delta}{=} \mathbf{A}_{best} \mathbf{V}_{sd}^{best} \tag{8}$$

This means only picking up the "best" speaker's model $\mathbf{V}_{sd}^{best}$ for adaptation. In implementation, rather than using the linear regression in Eq.(8), a direct parameter update algorithm for non-linear regression was applied for more accurate estimation. The matrix in DNN is decomposed into two components as shown in Figure 3. Only matrix parameters in $\mathbf{V}_{sd}^{test}$ are updated from an initial model of $\mathbf{V}_{sd}^{best}$ using adaptation data.

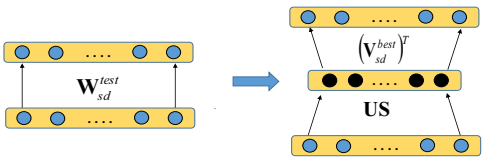

Figure 3: *Decomposition of weight matrix* $\mathbf{W}_{sd}^{best}$ *for speaker coefficient matrix adaptation.*

In order to reduce the number of adaptation parameters, low-rank form as introduced in section 2.2 can be used. All of the equations and formulations in Eqs.(4), (5), (6), (7), (8) hold by changing corresponding matrix to its low-rank form.

2.4.2. Updating singular values in the SD layer

After picking up a "best" speaker's model for adaptation, we can suppose that the left and right singular vectors are fixed, only the singular values are adjusted to weight these two singular vectors for a testing speaker. We formulate this idea as follows.

For an initial model (the "best" one from SAT-DNN ensembles), the factorization of the SD matrix is:

$$\mathbf{W}_{sd}^{best} = \mathbf{U}_{sd}^{best} \Sigma_{\alpha} \left(\mathbf{V}_{sd}^{best}\right)^{T} \tag{9}$$

where $\Sigma_{\alpha} = \mathrm{diag}\left(\alpha_1, \alpha_2, ..., \alpha_i, ..., \alpha_p\right)$, p=min{m,n}.

For a test speaker, we suppose that the $\mathbf{U}^{best}$ and $\left(\mathbf{V}_{sd}^{best}\right)^{T}$ are kept the same and only the singular value matrix is updated as:

$$\mathbf{W}_{sd}^{test} = \mathbf{U}_{sd}^{best} \Sigma_{\beta} \left(\mathbf{V}_{sd}^{best}\right)^{T} \tag{10}$$

where $\Sigma_{\beta} = \mathrm{diag}\left(\beta_1, \beta_2, ..., \beta_i, ..., \beta_p\right)$. Then purpose of adaptation is to find a mapping function as:

$$\beta_i = g_i\left(\alpha_i\right), i = 1,2,..., p \tag{11}$$

In real implementation, it is accomplished by inserting a linear transformation matrix **M** between $\mathbf{U}_{sd}^{best}$ and Σ_{α} according to Eq.(10). Figure 4 shows the decomposition structure in DNN implementation. The transformation matrix **M** can be initialized by using identity matrix.

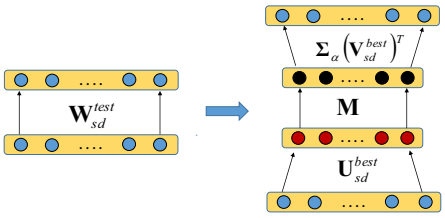

Figure 4: *Decomposition to weight matrix* $\mathbf{W}_{sd}^{best}$ *for singular values adaptation.*

In this paper, we only update the diagonal elements of **M**. The advantage of singular value adaptation strategy is that only a small number of p parameters are involved in adaptation, i.e., the number of adaptation parameters is drastically reduced.

3. Implementation and evaluations

3.1. Corpus of spoken lectures

We organize our speech data of Chinese lectures [19] into three parts as listed in Table 1.

- TRN: For acoustic model training.
- DEV: For validation when DNN training and adaptation.
- TST: For evaluation the results.

Table 1. *Data descriptions.*

Data set	#Speakers	Duration (hours)
TRN	184	97.2
DEV	12	7.2
TST	19	11.9

3.2. Baseline SAT-DNN model

In baseline modeling, a GMM-HMM with 3000 tied-triphone-states model was built by using the TRN set. For DNN model training, 40 dimensional filterbank features, plus their first and second derivatives were used as a feature set. The DNN has 1320 neuron nodes in the input layer (5 frames on each side of the current frame), 3000 neuron nodes in the output layer, and 1024 neuron nodes in each hidden layer (7 hidden layers). Training of SI-DNN consists of the unsupervised pre-training step and the supervised fine-tuning step. 184 speakers in the training set were used in SAT training based on the SI-DNN model. And finally, 184 SD-DNN models were obtained. Although in SAT-DNN, choosing the second or third hidden layer as the SD layer in adaptation could obtain a better result than choosing other layers as shown in [7], there is no clear theoretical support on which layer should be used as the SD layer in SAT training. In this study, we only perform SAT training on the second hidden layer. Kaldi DNN toolkit (nnet1) [20] and theano library [21] were used in our implementation. And the training procedures used in [7] were followed in our implementation.

The dictionary consists of 53K lexical entries from the TRN together with Hub4 and TDT4 corpora. The OOV rate on the TST is 0.368%. The pronunciation entries were derived from the CEDICT open dictionary. We adopt 113 phonemes (consonants and 5-tone vowels).

A word trigram language model was built for decoding with Julius [22]. We complemented the small sized text of the TRN with lecture texts collected from the web, whose size is 1.07M words. Then, this lecture corpus was interpolated with other three corpora (Hub4 of 0.34M, TDT4 of 4.75M, GALE of 1.03M) and lecture text archive from Phoenix TV station (Hong Kong) of 4.12M. The interpolated weights were determined to get the lowest perplexity on the DEV set.

We conducted recognition experiments on the TST set to see whether the adaptation is effective or not. We modified Julius for fast decoding with the DNN acoustic model. This baseline system achieved an average Character Error Rate (CER) of 28.5% with the DNN-HMM model on the TST set.

3.3. Experimental setups for ensemble speaker modeling

By concatenating the weight matrices (1024×1024) of these SD layers, we got a super matrix (1024×188416). SVD was applied on this super matrix for factorization. Based on the factorization, globally shared speaker subspace **U** (1024 × rank), singular value matrix S (rank×rank), and the coefficient matrix related to each speaker $(\mathbf{V}_{sd}^{i})^{T}$ (rank × 1024) were obtained. In experiments, four rank values (1024, 500, 300 and 100) were tested and full rank value is 1024.

When selecting the initial model for each testing speaker, we choose the SD layer with highest frame accuracy on the testing data compared to the labels derived in an unsupervised way.

3.4. Experimental evaluations

By gradually reducing the adaptation data for each testing speaker from 50 utterances (1 minute on average), to 30 utterances (half a minute on average), and then to 10 utterances (10 seconds on average), we carried out experiments to test the two adaptation algorithms as introduced in section 2.4, i.e., speaker coefficient matrix adaptation, and singular value matrix adaptation. Table 2 shows the results for different experimental conditions. In this table, SAT means baseline SAT-DNN model. SAT-SVD-V denotes adaptation on speaker coefficient matrix **V** (with rank of the matrix specified in bracket), and SAT-SVD-S represents adaptation on the singular values.

Table 2. *Adaptation performances (CER% on TST).*

	Parameter size for adaptation	w/o adaptation	#utterances for adaptation		
			50	30	10
SAT (baseline)	1024*1024	28.5	26.8	27.1	27.7
SAT-SVD-V (r=1024)	1024*1024	28.5	26.8	**26.9**	**27.5**
SAT-SVD-V (r=500)	1024*500	28.5	**26.6**	**26.9**	**27.5**
SAT-SVD-V (r=300)	1024*300	**28.4**	**26.5**	**27.0**	27.6
SAT-SVD-V (r=100)	1024*100	29.6	26.8	27.5	28.4
SAT-SVD-S	1024	28.5	27.1	27.4	27.9

The utterances are randomly selected from those sentences with the averaged word confidence score larger than 0.8. The improvements compared to the baseline with statistical significance (by the NIST Scoring Toolkit) are shown in bold fonts.

From Table 2, we observed the rank and the adaptation data size exerted large influence to the adaptation results.

For the first method (SAT-SVD-V), most of its performances are higher than or equivalent with the baseline SAT adaptation method, except when the rank is too small (100). SAT-SVD-V (rank=1024) outperforms SAT baseline on small data cases (30 utterances and 10 utterances), although they have the same number of parameters for adaptation. This result shows selecting the "best" initial model for adaptation is effective.

We also notice that SAT-SVD-V is better than the baseline SAT adaptation at rank=300 and rank=500 after adaptation with all data cases. Especially for the rank=300, the bottleneck structure seems to introduce more robustness even without adaptation. The low-rank approximation based adaptation technique shows better accuracy with large reduction on number of adaptation parameters.

The second method (SAT-SVD-S) is more sensitive to the adaptation data size due to its very limited number of parameters for adaptation (1024). But it still mostly outperforms the speaker coefficient matrix adaptation with low-rank case of SAT-SVD-V (r=100) which holds 100*1024 model parameters.

4. Conclusions

In this paper, we proposed an ensemble speaker modeling framework for speaker adaptation using speaker adaptive training DNN with two different kinds of implementations. A speaker ensemble matrix is composed by concatenating all of the SD neural weight matrices. By applying SVD to the ensemble matrix, a full-rank or low-rank speaker subspace representation is extracted. When testing, we select an initial model for each target speaker in this speaker subspace. The experimental results showed the effectiveness of the proposed scheme in improving the recognition performance. Our proposed scheme will be fully explored in our future work.

5. Acknowledgements

The authors would like to thank Dr. Shinsuke Sakai for his collaboration and suggestions throughout this work.

6. References

[1] J. Neto, L. Almeida, M. Hochberg, C. Martins, L. Nunes, S. Renals, and T. Robinson, "Speaker adaptation for hybrid HMM–ANN continuous speech recognition system," *in Proc. Eurospeech*, pp. 2171–2174, 1995.

[2] V. Abrash, H. Franco, A. Sankar, and M. Cohen, "Connectionist speaker normalization and adaptation," *in Proc. Eurospeech*, pp. 2183–2186, 1995.

[3] F. Seide, X. Chen, and D. Yu, "Feature engineering in context-dependent deep neural networks for conversational speech transcription," *in Proc. IEEE ASRU*, 2011.

[4] 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 Proc. IEEE SLT*, 2012.

[5] Y. Miao, H. Zhang, F. Metze. "Towards Speaker Adaptive Training of Deep Neural Network Acoustic Models", *in Proc. INTERSPEECH*, 2014.

[6] Y. Miao, L. Jiang, H. Zhang, F. Metze, "Improvements to Speaker Adaptive Training of Deep Neural Networks", *in Proc. IEEE SLT*, 2014.

[7] T. Ochiai, S. Matsuda, X. Lu, C. Hori and S. Katagiri, "Speaker adaptive training using deep neural networks," *in Proc. IEEE ICASSP*, 2014.

[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 Proc. IEEE ICASSP*, pp. 7893-7897, 2013.

[9] H. Liao, "Speaker adaptation of context dependent deep neural networks," *in Proc. IEEE ICASSP*, pp. 7947–7951, 2013

[10] J. Xue, J. Li, D. Yu, M. Seltzer, and Y. Gong, "Singular value decomposition based low-footprint speaker adaptation and personalization for deep neural network," *in Proc. IEEE ICASSP*, 2014.

[11] J. Xue, J. Li, and Y. Gong, "Restructuring of deep neural network acoustic models with singular value decomposition," *in Proc. INTERSPEECH*, pp. 2365-2369, 2013.

[12] Y. Tsao, P. Lin, T. Hu, X. Lu, "Ensemble environment modeling using affine transform group," *Speech Communication* 68: 55-68 (2015).

[13] Y. Tsao and C.-H. Lee, "An Ensemble Speaker and Speaking Environment Modeling Approach to Robust Speech Recognition," *IEEE ASLP*, vol.17, pp.1025-1037, 2009.

[14] X. Lu, Y. Tsao, S. Matsuda, and C. Hori, "Ensemble Modeling of Denoising Autoencoder for Speech Spectrum Restoration," *in Proc. INTERSPEECH* , 2014.

[15] M. Seltzer and J. Droppo, "Multi-task learning in deep neural networks for improved phoneme recognition," *in Proc. IEEE ICASSP*, pp. 6965–6968, 2013.

[16] J. Huang, J. Li, D. Yu, L. Deng, and Y. Gong, "Cross-language knowledge transfer using multilingual deep neural network with shared hidden layers," *in Proc. IEEE ICASSP*, pp. 7304 – 7308, 2013.

[17] E. Anderson, Z. Bai, C. Bischof, S. Blackford, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, D. Sorensen, "LAPACK Users' Guide," *Society for Industrial and Applied Mathematics*,1999.

[18] L. Trefethen and D. Bau, "Numerical Linear Algebra," *Society for Industrial and Applied Mathematics*, 1997.

[19] S. Li, Y. Akita and T. Kawahara, "Corpus and transcription system of Chinese lecture room," *In Proc. ISCSLP*, 2014.

[20] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, J. Silovsky, G. Stemmer, and K. Vesely, "The Kaldi speech recognition toolkit," *in Proc. IEEE ASRU*, 2011.

[21] J. Bergstra, F. Bastien, O. Breuleux, P. Lamblin, R. Pascanu, O. Delalleau, et al., "Theano: Deep learning on gpus with python," *in Proc. Big Learning Workshop, NIPS*, vol.11, pp.1–6, 2011.

[22] A. Lee and T. Kawahara. "Recent development of open-source speech recognition engine Julius," *In Proc. APSIPA ASC*, pp.131-137, 2009.