SPEAKER ADAPTIVE TRAINING USING DEEP NEURAL NETWORKS

Tsubasa Ochiai[1,2]*, Shigeki Matsuda*[1]*, Xugang Lu*[1]*, Chiori Hori*[1]*, and Shigeru Katagiri*[2]

[1] Spoken Language Communication Laboratory,
National Institute of Information and Communications Technology, Kyoto, Japan
[2] Graduate School of Engineering, Doshisha University, Kyoto, Japan

dun0139@mail4.doshisha.ac.jp,
{shigeki.matsuda,xugang.lu,chiori,hori}@nict.go.jp,
skatagir@doshisha.ac.jp

ABSTRACT

Among many speaker adaptation embodiments, Speaker Adaptive Training (SAT) has been successfully applied to a standard Hidden-Markov-Model (HMM) speech recognizer, whose state is associated with Gaussian Mixture Models (GMMs). On the other hand, recent studies on Speaker-Independent (SI) recognizer development have reported that a new type of HMM speech recognizer, which replaces GMMs with Deep Neural Networks (DNNs), outperforms GMM-HMM recognizers. Along these two lines, it is natural to conceive of further improvement to a preset DNN-HMM recognizer by employing SAT. In this paper, we propose a novel training scheme that applies SAT to a SI DNN-HMM recognizer. We then implement the SAT scheme by allocating a Speaker-Dependent (SD) module to one of the intermediate layers of a seven-layer DNN, and elaborate its utility over TED Talks corpus data. Experiment results show that our speaker-adapted SAT-based DNN-HMM recognizer reduces the word error rate by 8.4% more than that of a baseline SI DNN-HMM recognizer, and (regardless of the SD module allocation) outperforms the conventional speaker adaptation scheme. The results also show that the inner layers of DNN are more suitable for the SD module than the outer layers.

Index Terms—
Speaker Adaptative Training, Deep Neural Network

1. INTRODUCTION

Speaker adaptation is one key concept for achieving high performing speech recognition. A typical example is adapting a present speech recognizer, such as a Speaker-Independent (SI) speech recognizer, to a particular speaker to accurately recognize his/her speech data only using a limited volume of his/her data. Among various approaches to this important concept, Speaker Adaptive Training (SAT) has been successfully applied to a standard Hidden-Markov-Model (HMM) speech recognizer that adopts Gaussian Mixture Models (GMMs) for estimating the emission probability at every state of the HMM structure [1]. SAT is a training scheme for producing a compact, easy-to-adapt speaker model through the concurrent attempts of normalizing speaker-dependent acoustic variability in speech signals and optimizing such recognizer parameters as GMMs to achieve high recognition accuracy. The recent embodiments of this scheme include applications to the Shallow-Neural-Network (SNN)-based speech recognizer [2].

A new type of HMM recognizer that uses a Deep Neural Network (DNN) in place of GMMs has attracted great research interest in SI speech recognizer development [3]. Probably because DNN gains high discriminative power based on its discriminative training paradigm and its high feature representation capability, DNN-HMM recognizers generally achieve accurate recognition.

In light of the above two recent trends, we propose in this paper a new training scheme for achieving increased recognition accuracy by applying SAT to a SI DNN-HMM recognizer. The contrivance of our proposed scheme is three-fold: it allocates a Speaker-Dependent (SD) module to one of the intermediate layers of DNN; it optimizes the entire network as well as the SD module in the SAT framework that synchronizes the selection of the SD module and the speaker data for training; it only adapts the SD module to a target speaker. Some of the authors of this paper previously outlined our proposed scheme [4]. However, its details have not yet been reported. We define our scheme in detail and experimentally elaborate its effectiveness in the difficult, TED Talks corpus data task.

2. PROPOSED METHOD

2.1. Overview

Several speaker adaptation techniques using a hybrid of Neural Network (NN) and HMM were proposed outside the SAT framework (e.g., [5, 6, 7, 8]). Some early cases among the techniques include the conventional SNN that attempted speaker adaptation by simply adding a linear projection network to either an input or an output layer of SI network [5, 6]. Recent cases employed DNN and adapted the SI network by incorporating regularization constraints [7, 8].

Unlike the above cases, our proposed scheme adopts the following three SAT-based steps: 1) DNN initialization, 2) DNN retraining by assigning one intermediate layer to the SD module, and 3) adaptation of the SD module. Note here that the HMM part of the DNN-HMM recognizer is assumed to be already trained by such appropriate criterion as Maximum Likelihood and Maximum Mutual Information. Similar to the preceding DNN-HMM recognizer architecture, the DNN part of our recognizer has the same number of output nodes as the HMM states of triphones, i.e., senones. Each output node produces a probability estimate that is used for calculating the emission probability for its corresponding senones. In addition, DNN training is executed by minimizing the Cross-Entropy (CE) loss for every input acoustic feature vector, which is converted from an input speech signal. The definition of the input vectors will be given below in 3.1.

Fig. 1 illustrates the structure of our DNN, the procedure of conducting SAT with allocating SD modules, and the procedure of adapting an SD module to a selected speaker. Our DNN is basically

978-1-4799-2893-4/14/$31.00 ©2014 IEEE

a standard Multi-Layer Perceptron (MLP) network whose node has connection weights and bias. In the figure, we assume that our DNN has seven layers ($L_0, \ldots, L_6$), and for illustration simplicity, it has just two nodes at every layer. The weights between layers L_l and L_{l-1} are represented in the matrix form such as $\mathbf{W}_l$, but the details of this definition will be given in subsequent subsections where we describe each step of the training scheme in detail. In the figure, again for illustration simplicity, no biases are depicted.

2.2. Initialization

Fig. 1 (a) illustrates the initial status of our DNN, which works as part of the baseline SI DNN-HMM recognizer.

For effective network training, the DNN must be appropriately initialized. A standard initialization is using the Restricted Boltzmann Machine (RBM). However, this non-discriminative training is not necessarily suitable for recognition. Therefore, an alternative, somewhat advanced initialization can be considered. An example of such advanced initialization is to discriminatively train the RBM-pre-trained DNN with Error Back Propagation (EBP) training [9] using the minimum CE loss criterion.

In Fig. 1 (a), $\mathbf{W}_l^{\mathrm{SI}}$ represents the weight matrix of L_l (and L_{l-1}), which is produced by the above CE loss minimization and works for the SI DNN-HMM recognizer.

2.3. Re-training with speaker-dependent module allocation

In Fig. 1, we assume the allocation of SD modules, i.e., $SD_1, \ldots, SD_S$, to the second layer (L_2), where S is the number of speakers in the training dataset and $\mathbf{W}_2^s$ is the weight matrix of SD_s.

First, as illustrated in Fig. 1 (b), we initially set S SD modules to layer L_2 by copying $\mathbf{W}_2^{\mathrm{SI}}$ of the initial network. Note that the node connection between an added SD module (at L_2) and its adjacent layers (L_1 and L_3) is dynamically controlled in conjunction with the speaker-by-speaker selection of the training speech data. Fig. 1 (b) illustrates two example cases: one for using the speech data of speaker 1 and one for speaker 2. When using the data of speaker 1, only the nodes of SD_1 are connected with the nodes of the adjacent layers; the nodes of the other SD modules, $SD_2, \ldots, SD_S$, are disconnected with the nodes in the adjacent layers. The green dashed line depicts this situation, and training is executed only along this path. Similarly, the purple solid line depicts the situation in the case of using the data of speaker 2. Clearly, each SD module is trained only using its corresponding speaker's data, but the other part of the network is trained using the data of all speakers.

We conducted the re-training using EBP training, as in the initialization stage. One additional device here is to incorporate regularization into the EBP training. The size of each bit of speaker data is usually limited, and this restriction easily causes over-fitting to the training data, or in other words, a decrease in the robustness to the unseen data for the SD modules. The regularization details will be explained in 2.5.

Here, to circumvent the over-fitting problem, alternatives to regularization are possible, e.g., Cluster Adaptive Training (CAT) [10], which increases the amount of speech data that can be used to train the SD module by clustering all the available speakers into groups of acoustically similar speakers.

As reported in previous studies on SAT, the above speaker-by-speaker re-training is expected to increase the adaptability of the network to some selected speaker's data.

2.4. Speaker adaptation using speaker-dependent modules

In the final stage of the SAT scheme, we adapt a preset speaker module, which is placed in the re-trained DNN, using the speech data

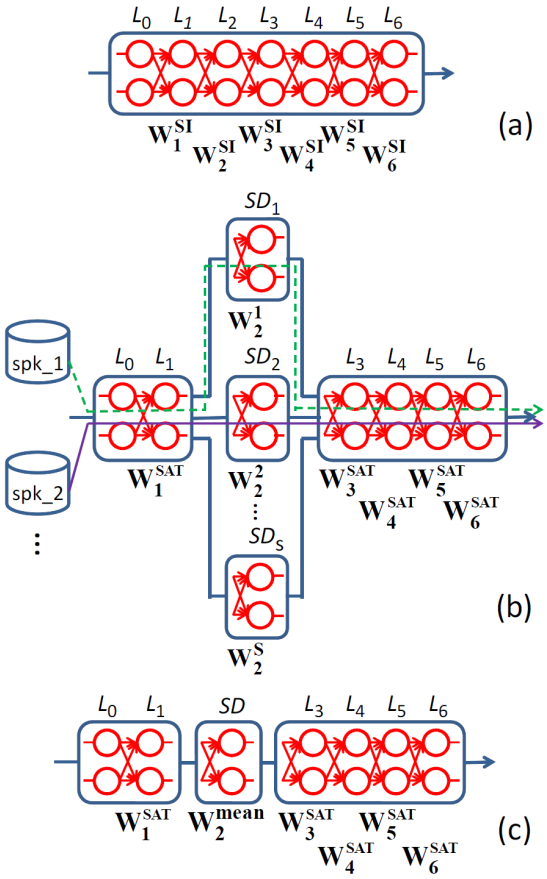

Fig. 1. Network configurations and training procedures of Speaker Adaptive Training (SAT) for DNN. (a) Speaker-Independent (SI) DNN, (b) Training procedure with Speaker-Dependent (SD) module allocation, (c) Pre-Trained Speaker Adaptive Training (PT-SAT) DNN.

of a particular target speaker. In the scenario of Fig. 1, the preset speaker module is embedded into L_2 and adapted.

Fig. 1 (c) illustrates the status in which a preset speaker module, represented by $\mathbf{W}_2^{\mathrm{mean}}$, is set to L_2 of the re-trained DNN, where $\mathbf{W}_l^{\mathrm{SAT}}$ represents the weight matrix generated through re-training. Here, there are many possible ways of preparing the preset speaker module. In the figure, we make the following assumption about the module: 1) place the 2nd layer's weight matrix of SI DNN as an initial state of the SD module, and 2) re-train the initial matrix using all the training speech data using non-regularized EBP training with the CE loss minimization criterion. The Term $\mathbf{W}_2^{\mathrm{mean}}$ represents the situation where all the training speech data is used for updating its corresponding parameters (weights), and we refer to the resulting network (Fig. 1 (c)) as Pre-Trained SAT (PT-SAT) network. We adopt non-regularized EBP training because all the training data, which are usually large, can be used.

In the adaptation, we adapt $\mathbf{W}_2^{\mathrm{mean}}$ using the target speaker's speech data. Note that the other weights all are fixed. Similar to the former re-training stage using all the training data, the update of $\mathbf{W}_2^{\mathrm{mean}}$ is executed with the regularized EBP training, where the regularization is incorporated into the EBP training using the CE loss. Since the data size for the adaptation is often limited, we need to take the over-fitting problem into account. Also note that the adaptation of only the SD modules probably helps circumvent the over-fitting

problem because it reduces the number of parameters to train by not re-training many weights of the other part of the network.

2.5. Regularization

As described in the previous subsection, since the amount of speech data for one speaker is often limited, our re-training and adaptation of the SD module uses regularized EBP training in place of standard (non-regularized) training. There are several possible definitions of the regularization term. Among them, we generally use the L_2 norm of the difference between the initial weight matrix ($\mathbf{W}_{l_{\rm SD}}^{\rm SI}$ for producing the network of Fig. 1 (b), and $\mathbf{W}_{l_{\rm SD}}^{\rm mean}$ for producing a (speaker-adapted) SAT network that is further modified from the network of Fig. 1 (c)) and the weight matrix of the SD module.

Even if the training data are limited in the re-training and adaptation stages for the SD modules, the weights of the other network layers can be trained using a sufficient amount of training data. The data of all the speakers can be used for training the weights outside the SD modules. Therefore, we only incorporate the regularization term to the SD module.

The regularization term for training the SAT recognizer is defined as follows [7]:

$$R(\Lambda) = \frac{1}{2}\|\mathbf{W}_{l_{\rm SD}}^{t} - \mathbf{W}_{l_{\rm SD}}^{\rm mean}\|_2^2 + \frac{1}{2}\|\mathbf{b}_{l_{\rm SD}}^{t} - \mathbf{b}_{l_{\rm SD}}^{\rm mean}\|_2^2 \quad (t = 1, 2, ..., T) \tag{1}$$

where $\mathbf{W}_{l_{\rm SD}}^{t}$ and $\mathbf{b}_{l_{\rm SD}}^{t}$ are the weight matrix of the t-th SD module and its corresponding bias vector in the $l_{\rm SD}$ layer, respectively; T is the number of speakers for adaptation; $\mathbf{W}_{l_{\rm SD}}^{\rm mean}$ and $\mathbf{b}_{l_{\rm SD}}^{\rm mean}$ are an initial setting of the weight matrix and the bias vector at the time position right before the adaptation, which corresponds to Fig. 1 (c), respectively; Λ represents all the trainable network parameters such as the weights and biases.

For producing the network of Fig. 1 (b), $\mathbf{W}_{l_{\rm SD}}^{\rm mean}$ and $\mathbf{b}_{l_{\rm SD}}^{\rm mean}$ in (1) are replaced with $\mathbf{W}_{l_{\rm SD}}^{\rm SI}$ and $\mathbf{b}_{l_{\rm SD}}^{\rm SI}$, both of which are the weight matrix of SI DNN and its corresponding bias vector in the l_{SD} layer, respectively. Also, the speaker set consisting of T speakers used in (1) is replaced by the training data set consisting of S speakers.

3. EXPERIMENTS

3.1. Conditions

3.1.1. Data

We tested our proposed method on the difficult, lecture speech data of the TED Talks corpus, under the supervised adaptation setups. We prepared three data sets: training, evaluation, and testing.

The training data set consisted of the speech data of 300 speakers; each speaker's data was about 30 minutes. The total length of the training data was about 150 hours. The evaluation data set consisted of the speech data of eight speakers, each of whom was different from the speakers in the training data. This set was used for finding the optimal values of the hyper-parameters, which produced high recognition accuracies over the set itself, such as the learning rate of CE minimization and the regularization coefficient.

The testing data set consisted of the speech data of 28 speakers, which was used for the IWSLT2013 testing data set, each of whom was different from the speakers both in the training and evaluation data sets.

3.1.2. Adopted recognizers

To evaluate our proposed SAT-based DNN-HMM recognizer, we compared its performance with those produced by a baseline SI DNN-HMM recognizer, a Speaker-Adapted (SA) DNN-HMM recognizer, and a SAT-based DNN-HMM recognizer.

Here, the baseline recognizer simply adopted the seven-layer DNN as its front-end, and the whole network was first initialized by RBM training and trained using CE minimization optimization over the training data (see Fig. 1 (a)).

The SA recognizer was implemented by adapting one of the SI recognizer's intermediate network layers, which corresponds to a speaker-dependent (SD) module, using the speech data of an adaptation target speaker that was selected from the 28 testing speakers. To circumvent the problem of closed-form training, we divided the speech data of every testing speaker into four subgroups and obtained recognition results in the four-times cross-validation (CV) scheme. In this CV scheme, we used one of the subgroups for testing and the three remaining subgroups for training and obtained the average recognition accuracies by changing a subgroup for four trainings.

The SAT procedure first adopted the baseline SI recognizer as the initial status of the SAT-based recognizer and next prepared SD modules, whose numbers were the same as those of the training speakers, i.e., 300; the procedure next generated a PT-SAT network along the course of Fig. 1 (b) to Fig. 1 (c). Finally, in the adaptation stage, the SAT procedure generated the SAT recognizer by training only the SD module in the speaker-by-speaker mode, where an adaptation target speaker was selected from the 28 testing speakers.

When adapting the SI recognizer to the SA recognizer, we must take the over-fitting problem into account, because the SD module set in layer l_{SD} is adapted using a limited volume of the data of a selected speaker. Therefore, in this adaptation, we applied the regularization term of (1) to the update of the weights and biases of layer l_{SD}, changing $\mathbf{W}_{l_{\rm SD}}^{\rm mean}$ and $\mathbf{b}_{l_{\rm SD}}^{\rm mean}$ to $\mathbf{W}_{l_{\rm SD}}^{\rm SI}$ and $\mathbf{b}_{l_{\rm SD}}^{\rm SI}$, respectively.

In all of our recognizers, the HMM part used the 4-gram language model that was trained over the transcriptions of TED Talks, News Commentary, and English Gigaword [11] and used the context-dependent acoustic model that was trained with the Boosted MMI training. During the DNN training, all of the HMM parameters were fixed, such as the language model and the state transition probabilities.

The DNN module in our recognizers used 429 input nodes, 4909 output nodes, and 512 nodes for all of the intermediate layers.

As above, we selected one from the five intermediate layers as an SD module in the adaptation stage of either the SA or SAT recognizer and elaborated the layer selection effect in the speaker adaptation by changing a selected layer from the 1st intermediate layer through the 5th intermediate layer. This is motivated by our research interest in attempts to reveal the roles of intermediate layers for (speaker) feature representation.

3.1.3. Acoustic feature representation

The input speech was first converted to a series of acoustic feature vectors, each of which was calculated through a 20-ms Hamming window that was shifted at 10-ms intervals. The acoustic feature vector consisted of 12 MFCCs, logarithmic power (log-power), 12 Δ MFCCs, Δ log-power, 12 $\Delta\Delta$ MFCCs, and $\Delta\Delta$ log-power, where MFCC stands for Mel-scale Frequency Cepstrum Coefficient, Δ is the first derivative, and $\Delta\Delta$ is the second derivative. The dimensions of the acoustic feature vectors were 39. Then the 11 concatenated acoustic feature vectors (429 dimensions) were used as input for the DNN's front-end. From the viewpoint of the Hamming window positioning, these 11 vectors were considered a concatenation of the acoustic feature vectors at the Hamming window position, five acoustic feature vectors at its preceding positions, and five acous-

Table 1. Experimental result (word error rate [%]). The upper row shows the names of the tested recognizers, i.e., the SI recognizer (baseline), the SA recognizer (adapted from SI), the PT-SAT recognizer (before adaptation), and the SAT recognizer (adapted from PT-SAT). The left-end column shows the number of a network layer, to which the SD module placed.

l_{SD}	SI	SA	PT−SAT	SAT
1	26.4	20.0	27.2	18.9
2	26.4	19.0	26.9	18.2
3	26.4	18.7	27.0	18.0
4	26.4	19.0	26.6	18.4
5	26.4	19.5	26.5	19.0

tic feature vectors at its succeeding positions. Each element of the 429-dimensional input vector was normalized so that its mean and variance became 0 and 1, respectively.

3.1.4. Hyper-parameter settings

DNN training sometimes requires careful control of the learning rate. Therefore we controlled it at each training epoch using the following rule based on recognition accuracies over the evaluation data. If the recognition error decreased over the evaluation data, the learning rate was kept the same as in the previous epoch. Otherwise, the learning rate was halved, and the network parameters, i.e., the weights, were replaced with those that produced the minimum recognition error rate in the preceding training epochs, and the training for these replaced weights was restarted using the halved learning rate.

The training of the SI and PT-SAT recognizers was started by setting the initial value of the learning rate to 0.004 and repeated 20 times, corresponding to 20 epochs, using the above learning rate updating rule. Similarly, when producing the network of Fig. 1 (b), the initial value of the learning rate was set to 0.004, the number of training epochs was 20, and additionally the regularization coefficient was set to 0.1.

In contrast, in the adaptation stage where only the SD module was updated, the learning rate was simply set to a fixed value that was selected based on the recognition accuracies over the evaluation data. We selected a learning rate of 0.005 for the adaptation of the SA recognizer and 0.001 for the adaptation of the SAT recognizer. Both of these adaptation procedures were repeated ten times, corresponding to ten epochs, with a regularization coefficient of 0.1, which was selected again using the recognition rates over the evaluation data.

3.2. Results

Table 1 shows the recognition performance in word error rate of the four tested recognizers, i.e., the SI recognizer produced with the non-regularized EBP training over all the training data, the SA recognizer produced by adapting only the SD module of the SI recognizer to a selected speaker, the PT-SAT recognizer produced with the SD module allocation and the re-training using the data of all of the speakers for training, and the SAT recognizer produced by adapting only the SD modules of the PT-SAT recognizer to a selected speaker. Each error rate for the SA and SAT recognizers is the average value obtained by the previously described CV scheme. In the table, l_{SD} is the number of layers to which the SD module was allocated. Because the baseline SI recognizer did not have an SD module, the same error rate value, 26.4%, was shown in all the corresponding columns.

The SAT DNN-HMM recognizer achieved the lowest error rate, 18.0%, which is an 8.4% reduction from that of the baseline SI DNN-HMM recognizer. The SA recognizer results show that even the speaker adaptation that re-trained the selected SD module embedded into the SI DNN produced an assured improvement in error reduction. Its error reductions from that of the baseline SI recognizer ranged from 6.4% to 7.7%. However, comparing the SI and SAT recognizers clearly demonstrates the effectiveness of the SAT training concept. Regardless of the layer to which the SD module was allocated, the SAT DNN-HMM recognizer outperformed the SA DNN-HMM recognizer. The results of the PT-SAT recognizer were not promising. However, this recognizer was just produced as an initial setting for successive adaptation, and therefore these high error rates should not be a problem.

Table 1 might suggest that the differences between the SA and SAT recognizers were not so large. However, from detailed analyses, we found that our SAT recognizer surely outperformed the SA recognizer: The SAT recognizer won at least in 75% of the 28 speakers for each placement of the SD module; 93%, which was the best rate, of the 28 speakers for the case of the SD module set in the first layer.

The table also shows a quite interesting finding. The adaptation that allocated the SD module to the inner layers such as the 3rd layer outperforms the case of allocating the SD module to the layers near the input or output of the network, such as the 1st and 5th layers. This phenomenon appeared commonly in both the SA DNN-HMM and SAT DNN-HMM recognizers. This suggests that DNN abstracts the input information or extracts some features from the input as data feed-forwarding progresses from the input layer to the upper layers, and when effectively controlling the use of the SD module and the speech data for adaptation, speaker-dependent features are concentrated in the inner layers of the network. Taking this point into account, we believe that using DNN is more suitable for speaker adaptation (probably also for other types such as speaking environment and transmission channel adaptations) than the conventional SNN or any simple front-end architecture that has no deep layer structure.

4. CONCLUSION

We proposed a new speaker adaptation training scheme that applies the Speaker Adaptive Training (SAT) concept to the training of a Deep Neural Network (DNN) front-end that is incorporated into the DNN-HMM recognizer. In this scheme, the HMM part of the recognizer is pre-trained independently of the DNN training, a Speaker-Dependent (SD) module is embedded into a selected layer of the DNN, and only the SD module is adapted in the adaptation stage. We evaluated our proposed scheme with the TED Talks corpus data task and clearly demonstrated its high utility. In addition, we successfully revealed that the SD module allocated into the inner layers worked better than that allocated into the outer layers such as the input and output layers, suggesting that DNN has a function that extracts abstract information or useful features from the input at its inner layers.

We have not yet fully explored the optimal settings for the hyper-parameters of DNN, such as the depth of the network (the number of network layers) and the number of nodes at each layer. This point must be further investigated. In addition, we will elaborate the definition of the regularization term and investigate the effect of incorporating the Cluster Adaptive Training (CAT) approach into our training scheme. The evaluation of the proposed method under the unsupervised conditions will be also an interesting research topic.

5. REFERENCES

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

[2] J. Trmal, J. Zelinka, and L. Muller, "On speaker adaptive training of artificial neural networks," in *Proc. Interspeech*, 2010, pp. 554–557.

[3] G. Hinton, L. Deng, D. Yu, G. Dahl, A. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. Sainath, and B. Kingsbury, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *Signal Processing Magazine, IEEE*, vol. 29, no. 6, pp. 82–97, 2012.

[4] C. L. Huang, P. R. Dixon, S. Matsuda, Y. Wu, X. Lu, M. Saiko, and C. Hori, "The NICT ASR System for IWSLT2013," in *Proceedings of IWSLT2013*, 2013.

[5] 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*, 1995, pp. 2171–2174.

[6] R. Gemello, F. Mana, S. Scanzio, P. Laface, and R. De Mori, "Linear hidden transformations for adaptation of hybrid ANN/HMM models," *Speech Communication*, vol. 49, no. 10-11, pp. 827–835, 2007.

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

[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. ICASSP*, 2013, pp. 7893–7897.

[9] D. Rumelhart, G. Hinton, and R. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, pp. 533–536, 1986.

[10] M. J. F. Gales, "Cluster adaptive training for speech recognition," in *Proc. ICSLP*, 1998, pp. 1783–1786.

[11] H. Yamamoto, Y. Wu, C. L Huang, X. Lu, P. R. Dixon, S. Matsuda, C. Hori, and H. Kashioka, "The NICT ASR System for IWSLT2012," in *Proceedings of IWSLT2012*, 2012.