SPEAKER ADAPTIVE TRAINING FOR DEEP NEURAL NETWORKS EMBEDDING LINEAR TRANSFORMATION NETWORKS

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

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

dun0139@mail4.doshisha.ac.jp,
{hideyuki.watanabe,xugang.lu,chiori.hori}@nict.go.jp,
{shmatsud,skatagir}@mail.doshisha.ac.jp

ABSTRACT

Recently, a novel speaker adaptation method was proposed that applied the Speaker Adaptive Training (SAT) concept to a speech recognizer consisting of a Deep Neural Network (DNN) and a Hidden Markov Model (HMM), and its utility was demonstrated. This method implements the SAT scheme by allocating one Speaker Dependent (SD) module for each training speaker to one of the intermediate layers of the front-end DNN. It then jointly optimizes the SD modules and the other part of network, which is shared by all the speakers. In this paper, we propose an improved version of the above SAT-based adaptation scheme for a DNN-HMM recognizer. Our new training adopts a Linear Transformation Network (LTN) for the SD module, and such LTN employment leads to more appropriate regularization in both the SAT and adaptation stages by replacing an empirically selected anchorage of a network for regularization in the preceding SAT-DNN-HMM with a SAT-optimized anchorage. We elaborate the effectiveness of our proposed method over TED Talks corpus data. Our experimental results show that a speaker-adapted recognizer using our method achieves a significant word error rate reduction of 9.2 points from a baseline SI-DNN recognizer and also steadily outperforms speaker-adapted recognizers, each of which originates from the preceding SAT-based DNN-HMM.

Index Terms— Speaker Adaptive Training, Deep Neural Network, Linear Transformation Network

1. INTRODUCTION

Speaker adaptation is one of the most important approaches to achieving high performing speech recognition. Recently, with the advent of a new hybrid approach that combines a Deep Neural Network (DNN) and a Hidden Markov Model (HMM) to speech recognition [1, 2, 3], several speaker adaptation methods using the hybrid DNN-HMM recognizer have been investigated [4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14]. An approach that applies the Speaker Adaptive Training (SAT) concept [15] to the DNN-HMM recognizer has been proved especially effective for increasing DNN's adaptability [4, 5, 6].

The SAT-based DNN-HMM speech recognition approaches are mainly divided into the following two groups. In the first group [5, 6], the SAT scheme was implemented by combining the Gaussian Mixture Model (GMM)-based speaker normalization in such input feature space as cMLLR [16] with the DNN-learning-based classifier design. In the second group [4], the SAT scheme was implemented by allocating one Speaker Dependent (SD) module for each training speaker to one of the intermediate DNN layers. The method then jointly optimized the SD modules and the remaining part of the network[1], which was shared by all the training speakers, with changing a pair comprised of the SD module and its corresponding training speaker in a speaker-by-speaker manner. The effectiveness of this SAT-based DNN training scheme, which we call the SAT-DNN-SD method in this paper, was clearly demonstrated through systematic experiments [4]. Also, in comparison with the first-group methods, the SAT-DNN-SD method has an obvious advantage. It can consistently apply the DNN's high learning capability to speaker normalization and classifier design in the SAT-based framework.

However, the SAT-DNN-SD method still has room for improvement. For example, it used a Speaker Independent (SI) DNN, which was trained in a speaker independent mode, as an anchorage state in the regularization of the SAT stage. This might have decreased the adaptability of SAT due to a large restriction from the SI-DNN. In addition, in the speaker adaptation stage, it just empirically initialized the SD module using a one-layer network that was extracted from the SI-DNN and retrained with the SAT-optimized SI network over the speech data of all the training speakers. This initialized SD module was denoted using the term "mean" because it represents a certain kind of mean speaker model [4]. Due to the lack of theoretical rationale in such usage, alternatives to SI-DNN and the mean speaker model will probably further improve the adaptation results.

Motivated by the above research, we extend in this paper our previously proposed SAT-based speaker adaptation scheme with the original DNN [4], i.e., the SAT-DNN-SD method, by embedding a Linear Transformation Network (LTN) [17, 18, 19] in DNN. In contrast to the SAT-DNN-SD method [4], our new method with LTN dynamically changes the anchorage state of the network weight and bias parameters in the regularization along the training progress of SAT and automatically provides, in a natural way, the initial status of an SD module that is used for a new speaker in the speaker adaptation step.

In subsequent sections, we formulate our proposed method and experimentally elaborate its effectiveness in a difficult TED Talks corpus data task. We also compare the results of our proposed method and the SAT-DNN-SD method.

This work was supported in part by Grant-in-Aid for Scientific Research (B) No. 26280063.

[1] For descriptive purposes, we refer to this remaining part of DNN as a Speaker Independent (SI) network.

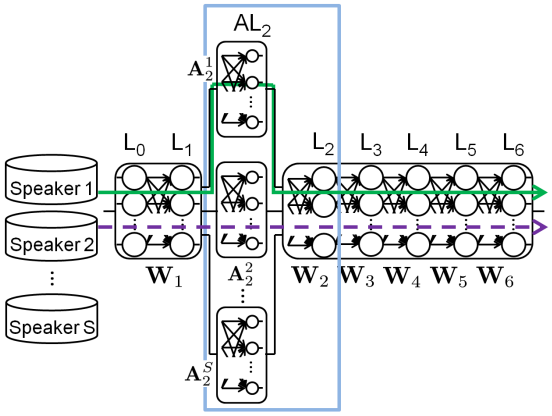

Fig. 1. SAT procedure with Linear Transformation Network

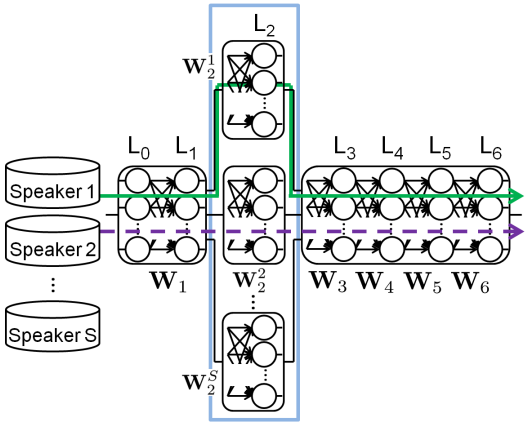

Fig. 2. SAT procedure with the original DNN

2. SAT-BASED SPEAKER ADAPTATION WITH LTN-EMBEDDED DNN

The structure of an LTN-embedded DNN is illustrated in Fig. 1, where the SAT procedure for this new type of network is also shown. We assume that our DNN has seven layers ($L_0, L_1, \ldots, L_6$), and for illustration simplicity, no biases are depicted. Because of the multi-layer structure, LTN can be allocated to any of the layers through L_1 to L_6. In the figure, as an example, we allocate LTN as SD modules, i.e., $A_2^1, A_2^2, \ldots, A_2^S$, to the second layer ($L_2$), where S is the number of speakers in the training dataset and $\mathbf{A}_2^s$ is the weight matrix of the LTN inserted into L_2 for training speaker s; $\mathbf{W}_l$ is the weight matrix of the original DNN between L_l and L_{l-1}.

2.1. Speaker adaptive training step

In the following for discussion simplicity, we denote the layer to which the SD modules are allocated as $L_{l_{SD}}$. From Fig. 1, the outputs from $L_{l_{SD}}$, which is referred to as $\mathbf{z}_{l_{SD}}$, is given as follows:

$$\mathbf{z}_{l_{SD}} = \mathbf{W}_{l_{SD}}(\mathbf{A}_{l_{SD}}\mathbf{z}_{l_{SD}-1} + \mathbf{a}_{l_{SD}}) + \mathbf{b}_{l_{SD}}, \quad (1)$$

where $\mathbf{A}_{l_{SD}}$ and $\mathbf{a}_{l_{SD}}$ are the weight matrix of the LTN inserted into L_2, and its corresponding bias vector, respectively.

In the SAT procedure, we first initialize our DNN's parameters, $\{\mathbf{W}_l, \mathbf{b}_l\}(l = 1, 2, \cdots, L)$, using the network parameters of SI-DNN. We next insert LTNs as SD modules, one for each training speaker, into one of the inner layers of our DNN. Then the parameters to be updated for the SD module corresponding to training speaker s are represented as $\mathbf{A}_{l_{SD}}^s, \mathbf{a}_{l_{SD}}^s$, and based on the SAT concept, they are updated by the following equation:

$$(\overline{\mathbf{\Lambda}}, \overline{\mathbf{A}}_{l_{SD}}^{SDs}, \overline{\mathbf{a}}_{l_{SD}}^{SDs}) = \underset{(\mathbf{\Lambda}, \mathbf{A}_{l_{SD}}^{SDs}, \mathbf{a}_{l_{SD}}^{SDs})}{\arg\min} E(\mathbf{\Lambda}, \mathbf{A}_{l_{SD}}^{SDs}, \mathbf{a}_{l_{SD}}^{SDs}) + \frac{\beta}{2} R(\mathbf{A}_{l_{SD}}^{SDs}, \mathbf{a}_{l_{SD}}^{SDs}), \quad (2)$$

where

$$R(\mathbf{A}_{l_{SD}}^{SDs}, \mathbf{a}_{l_{SD}}^{SDs}) = \sum_{s=1}^{S} \left(\|\mathbf{A}_{l_{SD}}^s - \mathbf{I}_{l_{SD}}\|^2 + \|\mathbf{a}_{l_{SD}}^s - \mathbf{0}_{l_{SD}}\|^2 \right), \quad (3)$$

$\mathbf{\Lambda} = \{\mathbf{W}_1, \cdots, \mathbf{W}_L, \mathbf{b}_1, \cdots, \mathbf{b}_L\}$, $\mathbf{A}_{l_{SD}}^{SDs} = \{\mathbf{A}_{l_{SD}}^1, \cdots, \mathbf{A}_{l_{SD}}^S\}$, $\mathbf{a}_{l_{SD}}^{SDs} = \{\mathbf{a}_{l_{SD}}^1, \cdots, \mathbf{a}_{l_{SD}}^S\}$, E is a loss function, R is a regularization term, β is its regularization coefficient, L is the number of network layers except the input layer, $\mathbf{I}_{l_{SD}}$ is an identity matrix, $\mathbf{0}_{l_{SD}}$ is a zero vector, $\|\cdot\|^2$ is L^2 norm, and $\overline{\mathbf{\Lambda}}$ is a resultant trained state of $\mathbf{\Lambda}$.

Since the size of the speech data from one speaker is usually limited, this often makes the SD modules over-fit the data. To alleviate this over-fitting problem, we introduce for LTN the regularization term shown in Eq. 3. The term works so that $\mathbf{A}_{l_{SD}}^s$ and $\mathbf{a}_{l_{SD}}^s$ do not differ too much from the identity matrix and the zero vector, respectively. Note here that the identity matrix and the zero vector work as an anchorage for regularization, which frees our SAT-optimized DNN from SI-DNN constraints. The linearity of the SD module virtually replaces the anchorage state of the network in the regularization term, which is originally the identity matrix (i.e., $\mathbf{I}_{l_{SD}}$), with the DNN's weight matrix (i.e., $\mathbf{W}_{l_{SD}}$). Clearly, the DNN's weight matrix alters along the course of the SAT progress. Then the network's anchorage state in the regularization also alters and comes close to a SAT-optimized network, which is different from the part of SI-DNN that was used in a previously proposed SAT-DNN-SD method [4]. This new feature of dynamically changing the anchorage state is expected to make the training of the linear SD module more suited to speaker adaptation, because the anchorage itself is optimized for SAT. The above virtual replacement holds for the bias vector.

Similar to the training for SAT-DNN-SD [4], we dynamically switched the node connections between an inserted SD module and its adjacent layers in conjunction with the speaker selection in the training data. For example, in Fig. 1, we only execute the training along the green solid line when using the data of speaker 1. Note that 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. Also, we adopt Error Back Propagation (EBP) training [20] with Cross Entropy (CE) loss.

2.2. Speaker adaptation step

In the speaker adaptation step, the above SAT procedure is expected to increase the adaptability of the entire network by only adapting the LTN module, which corresponds to the SD module, using a target speaker's speech data. The parameters of the SD module for target speaker t are represented as $\mathbf{A}_{l_{SD}}^t$ and $\mathbf{a}_{l_{SD}}^t$. Then the speaker adaptation procedure is defined as follows:

$$(\overline{\mathbf{A}}_{l_{SD}}^t, \overline{\mathbf{a}}_{l_{SD}}^t) = \underset{(\mathbf{A}_{l_{SD}}^t, \mathbf{a}_{l_{SD}}^t)}{\arg\min} E(\overline{\mathbf{\Lambda}}, \mathbf{A}_{l_{SD}}^t, \mathbf{a}_{l_{SD}}^t) + \frac{\beta}{2} R(\mathbf{A}_{l_{SD}}^t, \mathbf{a}_{l_{SD}}^t), \quad (4)$$

where

$$R(\mathbf{A}^t_{l_{SD}}, \mathbf{a}^t_{l_{SD}}) = \|\mathbf{A}^t_{l_{SD}} - \mathbf{I}_{l_{SD}}\|^2 + \|\mathbf{a}^t_{l_{SD}} - \mathbf{0}_{l_{SD}}\|^2, \quad (5)$$

In the adaptation step, we only adapt $\mathbf{A}^t_{l_{SD}}$ and $\mathbf{a}^t_{l_{SD}}$ using the target speaker's speech data. Note that the other weights all are fixed. In this step too, we adopt the EBP training with the CE loss.

2.3. Advantage of proposed scheme over previous SAT-DNN-SD

A previously proposed SAT-DNN-SD procedure [4] is illustrated in Fig. 2. The difference between this original SAT-DNN-SD and our proposed scheme, which we call SAT-DNN-LTN, originates from different SD module types.

In the original SAT-DNN-SD scheme, the SD module parameters for training speaker s are represented as $\mathbf{W}^s_{l_{SD}}$ and $\mathbf{b}^s_{l_{SD}}$. Then the SAT stage is formulated as follows:

$$(\overline{\mathbf{\Lambda}}^*, \overline{\mathbf{W}}^{SDs}_{l_{SD}}, \overline{\mathbf{b}}^{SDs}_{l_{SD}}) = \underset{(\mathbf{\Lambda}^*, \mathbf{W}^{SDs}_{l_{SD}}, \mathbf{b}^{SDs}_{l_{SD}})}{\arg\min} E(\mathbf{\Lambda}^*, \mathbf{W}^{SDs}_{l_{SD}}, \mathbf{b}^{SDs}_{l_{SD}}) + \frac{\beta}{2} R(\mathbf{W}^{SDs}_{l_{SD}}, \mathbf{b}^{SDs}_{l_{SD}}), \quad (6)$$

where

$$R(\mathbf{W}^{SDs}_{l_{SD}}, \mathbf{b}^{SDs}_{l_{SD}}) = \sum_{s=1}^{S} \Big(\|\mathbf{W}^s_{l_{SD}} - \mathbf{W}^{SI}_{l_{SD}}\|^2 + \|\mathbf{b}^s_{l_{SD}} - \mathbf{b}^{SI}_{l_{SD}}\|^2 \Big), \quad (7)$$

$\mathbf{\Lambda}^* = \{\mathbf{W}_1, \cdots, \mathbf{W}_{l_{SD}-1}, \mathbf{W}_{l_{SD}+1}, \cdots, \mathbf{W}_L, \mathbf{b}_1, \cdots, \mathbf{b}_{l_{SD}-1}, \mathbf{b}_{l_{SD}+1}, \cdots, \mathbf{b}_L\}$, $\mathbf{W}^{SDs}_{l_{SD}} = \{\mathbf{W}^1_{l_{SD}}, \cdots, \mathbf{W}^S_{l_{SD}}\}$, and $\mathbf{b}^{SDs}_{l_{SD}} = \{\mathbf{b}^1_{l_{SD}}, \cdots, \mathbf{b}^S_{l_{SD}}\}$. The regularization term here depends on the SI-DNN parameters.

In successive adaptation steps, the parameters of the SD module for target speaker t are also represented as $\mathbf{W}^t_{l_{SD}}$ and $\mathbf{b}^t_{l_{SD}}$, and then the speaker adaptation procedure is formulated as follows:

$$(\overline{\mathbf{W}}^t_{l_{SD}}, \overline{\mathbf{b}}^t_{l_{SD}}) = \underset{(\mathbf{W}^t_{l_{SD}}, \mathbf{b}^t_{l_{SD}})}{\arg\min} E(\overline{\mathbf{\Lambda}}^*, \mathbf{W}^t_{l_{SD}}, \mathbf{b}^t_{l_{SD}}) + \frac{\beta}{2} R(\mathbf{W}^t_{l_{SD}}, \mathbf{b}^t_{l_{SD}}), \quad (8)$$

where

$$R(\mathbf{W}^t_{l_{SD}}, \mathbf{b}^t_{l_{SD}}) = \|\mathbf{W}^t_{l_{SD}} - \mathbf{W}^{mean}_{l_{SD}}\|^2 + \|\mathbf{b}^t_{l_{SD}} - \mathbf{b}^{mean}_{l_{SD}}\|^2, \quad (9)$$

$\mathbf{W}^{mean}_{l_{SD}}$ and $\mathbf{b}^{mean}_{l_{SD}}$ are the initial states of the SD module in the adaptation step. See [4] for the initialization process.

In the SAT step, the SAT-DNN-SD method used the regularization term based on the SI-DNN parameters. However, since there are no theoretical bases for directly linking the SAT step with the SI-DNN parameters, they are not necessarily suited for the anchorage state of the network parameters in the SAT step's regularization. On the other hand, our proposed SAT-DNN-LTN uses the regularization term in Eq. 3, leading to a merit where the regularization term can use a network's anchorage state that better fits the SAT optimization framework.

In the speaker adaptation step, the SAT-DNN-SD method must prepare the initial status of SD module parameters $\mathbf{W}^{mean}_{l_{SD}}$ and $\mathbf{b}^{mean}_{l_{SD}}$. In contrast, our proposed SAT-DNN-LTN method automatically produces, through the SAT step, parameters $\overline{\mathbf{W}}_{l_{SD}}$ and $\overline{\mathbf{b}}_{l_{SD}}$ that correspond to the initial status of the SD module for the speaker adaptation step (Eq. 2) . The speaker adaptation step of our SAT-DNN-LTN method is expected to perform based on a more suitable anchorage state of the network parameters in the regularization.

3. EXPERIMENTS

3.1. Conditions

3.1.1. Speech data corpus and acoustic feature representation

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

The training dataset consisted of the speech data of 300 speakers; each speaker's data were about 30 minutes. The total length of the training data was about 150 hours. The evaluation dataset consisted of the speech data of ten speakers. The testing dataset consisted of the speech data of 28 speakers, which was used for the IWSLT2013 testing dataset.

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 the 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 11 concatenated acoustic feature vectors (429 dimensions) were used as input to the DNN's front-end. Each element of the 429-dimensional input vector was normalized so that its mean and variance became 0 and 1, respectively.

3.1.2. Adopted recognizers

To evaluate our SAT-DNN-LTN method, we compared the performance derived by the Speaker-Adapted SAT-DNN-LTN (SA-SAT-L) recognizer with those produced by the baseline SI-DNN recognizer, the Speaker-Adapted SI with the LTN (SA-SI-L) recognizer, the Speaker-Adapted SI (SA-SI-SD) recognizer, and the Speaker-Adapted SAT-DNN-SD (SA-SAT-SD) recognizer.

The SA-SI-L and SA-SAT-L recognizers were adapted using LTN. The SA-SI-L recognizer was developed using SI-DNN as an initial state of the recognizer in the speaker adaptation step. The SA-SAT-L recognizer was developed based on the SAT-DNN-LTN method. In the speaker adaptation step with LTN, the SA-SI-L recognizer was implemented by inserting LTN into one of the SI-DNN's intermediate network layers, which corresponded to an SD module and adapting it using the speech data of an adaptation target speaker that was selected from the 28 testing speakers. In this adaptation step, we applied the regularization term of Eq. (5) to the update of the weights and biases of layer l_{SD}.

The SA-SI-SD and SA-SAT-SD recognizers were adapted using the original DNN. The SA-SI-SD recognizer was developed using SI-DNN as an initial state of the recognizer in the speaker adaptation step. The SA-SAT-SD recognizer was developed using the SAT-DNN-SD method. In the speaker adaptation step with the original DNN, the SA-SI-SD recognizer was implemented by adapting one of the SI recognizer's intermediate network layers, which corresponded to a SD module. In this adaptation, we applied the regularization term of Eq. (9) to the update of the weights and biases of layer l_{SD}, changing $\mathbf{W}^{mean}_{l_{SD}}$ to $\mathbf{W}^{SI}_{l_{SD}}$ and $\mathbf{b}^{mean}_{l_{SD}}$ to $\mathbf{b}^{SI}_{l_{SD}}$.

The DNN module in our recognizers had seven layers and consisted of 429 input nodes, 4909 output nodes, and 512 nodes for all of the intermediate layers. We selected one from the five intermediate layers ($\mathrm{L}_1, \mathrm{L}_2, \ldots, \mathrm{L}_5$) as an SD module allocation or insertion layer in the adaptation stage of either the SA-SI recognizers (SA-SI-SD and SA-SI-L) or the SA-SAT recognizers (SA-SAT-SD and SA-SAT-L). We also elaborated the layer selection effect in the speaker adaptation by changing a selected layer from the 1st through the 5th intermediate layers to analyze the role of the intermediate layer in feature representation.

Table 1. Experimental results with SI-DNN recognizer and four different speaker-adapted DNN-based recognizers (word error rate [%])

l_{SD}	SI-DNN	SA-SI-SD	SA-SAT-SD	SA-SI-L	SA-SAT-L
1	26.4	20.0	18.9	20.8	20.5
2	26.4	19.0	18.2	19.3	17.2
3	26.4	18.7	18.0	19.0	17.5
4	26.4	19.0	18.4	19.0	17.5
5	26.4	19.5	19.0	19.2	18.0

In all of our recognizers, the HMM part used a 4-gram language model that was trained over the transcriptions of TED Talks, News Commentary, and English Gigaword [21] and used a context-dependent acoustic model that was trained with Boosted MMI training. During the DNN training, all of the HMM parameters were fixed, such as the language model and the state transition probabilities.

To circumvent the problem of closed-form evaluation, 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 it, we used one of the subgroups for testing and the three remaining subgroups for adaptation and obtained the average recognition accuracies by changing a subgroup for the four testings.

3.1.3. Hyper-parameter settings

Since DNN training sometimes requires careful control of the learning rate, we controlled it at each training epoch, which denoted the EBP training that used all of the training samples just once, using the following rule based on the 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, it was halved, and the network parameters, i.e., the weights and biases, were replaced with those that produced the minimum recognition error rate in the preceding training epochs, and the training for these replaced weights and biases were restarted using the halved learning rate. In the SAT step, we adapted each SAT-DNN (SAT-DNN-SD or SAT-DNN-LTN) recognizer and evaluated its performance over the evaluation data at each training epoch. We controlled the learning rate based on the recognition accuracies over the evaluation data. 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.

The hyper-parameters of baseline SI-DNN, SA-SI-SD, and SA-SAT-SD were set as previously described [4]. In the SAT step, the initial value of the learning rate was set to 0.004, the number of training epochs was 50, and the regularization coefficient (β) was set to 0.1 when inserting LTN into L_1 and to 10.0 when inserting LTN into L_2 through L_5. In the speaker adaptation step, we selected a learning rate of 0.00001 and a regularization coefficient of 0.1 for inserting LTN into L_1. On the other hand, we selected a learning rate of 0.00005 and a regularization coefficient of 10.0 in the case of inserting LTN into L_2 through L_5. These adaptation procedures were repeated ten times, corresponding to ten epochs.

3.2. Results and discussions

Table 1 shows the recognition performances of the five tested recognizers: SI-DNN, SA-SI-SD, SA-SAT-SD, SA-SI-L, and SA-SAT-L (our method proposed in this paper). Each error rate for the SA-SI-SD, SA-SAT-SD, SA-SI-L, and SA-SAT-L 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 or inserted.

The SA-SAT-L recognizer achieved the lowest error rate, 17.2%, which corresponded to 9.2 point reduction from the error rate of the baseline SI-DNN recognizer.

Comparisons of the SA-SI-SD and SA-SAT-SD recognizers and comparisons of the SA-SI-L and SA-SAT-L recognizers clearly demonstrate the effectiveness of the SAT training concept. Regardless of the layer to which the SD module was allocated or inserted, the SA-SAT recognizers outperformed the SA-SI recognizers.

A comparison between the speaker adaptation scheme with LTN and that with the original DNN also shows the following two aspects: 1) the SA-SI-L recognizer achieved almost the same performance as the SA-SI-SD recognizer, although the allocation of the SD module slightly affected their performances; 2) our proposed SA-SAT-L recognizer stably outperformed its counterpart SA-SAT-SD recognizer in all cases except when the SD module was allocated or inserted to L_1. As described in section 2.3, our proposed SAT-DNN-LTN dynamically estimates $\mathbf{W}_{l_{SD}}$ and $\mathbf{b}_{l_{SD}}$, which are anchorage states of network parameters in the regularization of the SAT step, and automatically estimates $\overline{\mathbf{W}}_{l_{SD}}$ and $\overline{\mathbf{b}}_{l_{SD}}$, which are the initial states of a SD module in the speaker adaptation step. Based on these properties, we expect that our SA-SAT-L performed the SAT and speaker adaptation steps using a more appropriate anchorage state of network parameters in the regularization.

When inserting LTN as an SD module into L_1, speaker adapted recognizers with LTN (SA-SI-L and SA-SAT-L) were slightly degraded, perhaps because at L_1, the number of LTN parameters was smaller than that of the original DNN parameters.

The table also shows a quite interesting finding. The adaptation using the SD module allocated to the central inner layers such as the 3rd layer outperformed that using 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 such speaker-adapted recognizers as SA-SAT-SD and SA-SAT-L. The result clearly suggests that it is important to balance the layers, or their corresponding weight and bias parameters, to the front and back of the SD module. This must be further investigated by analyzing such functional roles as feature transformation and classification played by the inner layers.

4. CONCLUSION

In this paper, we proposed an alternative speaker adaptation scheme for a DNN embedding Linear Transformation Network (LTN) that applied the SAT concept (SAT-DNN-LTN) and experimentally elaborated its effectiveness in a difficult TED Talks corpus data task. Our experimental results showed that our proposed SAT-based speaker adaptation scheme that embedded LTN stably outperformed the previously proposed SAT-based speaker adaptation scheme with the original DNN (SAT-DNN-SD) [4]. Based on the formulation of our proposed scheme, our SAT-DNN-LTN dynamically estimated the anchorage states of network parameters in the regularization of the SAT step and automatically estimated the initial states of a SD module in the speaker adaptation step. Therefore, we can expect that our SAT-DNN-LTN performed the SAT and speaker adaptation steps using a more appropriate anchorage state of network parameters in the regularization. We also consider that this new mechanism played a key role in increasing the performances of our proposed method.

Future work will include an evaluation under unsupervised adaptation setups. Applying a sequence training concept [22] to our proposed SAT scheme will also be an interesting research topic for achieving higher performance.

5. REFERENCES

[1] 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.

[2] G. Dahl, D. Yu, L. Deng, and A. Acero, "Context-dependent pre-trained deep deural networks for large-vocabulary speech recognition," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 1, pp. 30–42, 2012.

[3] A. Mohamed, G. Dahl, and G. Hinton, "Acoustic modeling using deep belief networks," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 1, pp. 14–22, 2012.

[4] T. Ochiai, S. Matsuda, X. Lu, C. Hori, and S. Katagiri, "Speaker adaptive training using deep neural networks," in *Proc. ICASSP*, 2014, pp. 6399–6403.

[5] S. P. Rath, D. Povey, K. Vesely, and J. H. Cernocky, "Improved feature processing for deep neural network," in *Proc. Interspeech*, 2013, pp. 109–113.

[6] T. Yoshioka, A. Ragni, and M. J. F. Gales, "Investigation of unsupervised adaptation of DNN acoustic models with filter bank input," in *Proc. ICASSP*, 2014, pp. 6394–6398.

[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] 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 *Proc. ICASSP*, 2013, pp. 7942–7946.

[10] 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 *Proc. ICASSP*, 2014, pp. 6339–6343.

[11] S. M. Siniscalchi, J. Li, and C.-H. Lee, "Hermitian based hidden activation functions for adaptation of hybrid HMM/ANN models," in *Proc. Interspeech*, 2012.

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

[13] F. Seide, G. Li, X. Chen, and D. Yu, "Feature engineering in context-dependent deep neural networks for conversational speech transcription," in *Proc. ASRU*, 2011, pp. 24–29.

[14] 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. ICASSP*, 2014, pp. 6359–6363.

[15] 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.

[16] M. J. F. Gales, "Maximum likelihood linear transformation for HMM-based speech recognition," *Computer Speech and Language*, vol. 12, no. 2, pp. 75–98, 1998.

[17] 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.

[18] 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.

[19] B. Li and K. C. Sim, "Comparison of discriminative input and output transformations for speaker adaptation in the hybrid NN/HMM systems," in *Proc. Interspeech*, 2010, pp. 526–529.

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

[21] 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 *Proc. IWSLT*, 2012.

[22] K. Vesely, A. Ghoshal, L. Burget, and D. Povey, "Sequence-discriminative training of deep neural networks," in *Proc. Interspeech*, 2013, pp. 2345–2349.