PAPER Special Section on Recent Advances in Machine Learning for Spoken Language Processing

Speaker Adaptive Training Localizing Speaker Modules in DNN for Hybrid DNN-HMM Speech Recognizers

Tsubasa OCHIAI[†a)], ***Student Member***, **Shigeki MATSUDA**[††*], **Hideyuki WATANABE**[††], ***Members***, **Xugang LU**[††], ***Nonmember***, **Chiori HORI**[††**], ***Member***, **Hisashi KAWAI**[††], ***Senior Member***, ***and*** **Shigeru KATAGIRI**[†], ***Member***

SUMMARY Among various training concepts for speaker adaptation, 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, focusing on the high discriminative power of Deep Neural Networks (DNNs), a new type of speech recognizer structure, which combines DNNs and HMMs, has been vigorously investigated in the speaker adaptation research field. Along these two lines, it is natural to conceive of further improvement to a DNN-HMM recognizer by employing the training concept of SAT. In this paper, we propose a novel speaker adaptation scheme that applies SAT to a DNN-HMM recognizer. Our SAT scheme allocates a Speaker Dependent (SD) module to one of the intermediate layers of DNN, treats its remaining layers as a Speaker Independent (SI) module, and jointly trains the SD and SI modules while switching the SD module in a speaker-by-speaker manner. We implement the scheme using a DNN-HMM recognizer, whose DNN has seven layers, and elaborate its utility over TED Talks corpus data. Our experimental results show that in the supervised adaptation scenario, our Speaker-Adapted (SA) SAT-based recognizer reduces the word error rate of the baseline SI recognizer and the lowest word error rate of the SA SI recognizer by 8.4% and 0.7%, respectively, and by 6.4% and 0.6% in the unsupervised adaptation scenario. The error reductions gained by our SA-SAT-based recognizers proved to be significant by statistical testing. The results also show that our SAT-based adaptation outperforms, regardless of the SD module layer selection, its counterpart SI-based adaptation, and that the inner layers of DNN seem more suitable for SD module allocation than the outer layers.

key words: *Deep Neural Networks, Hybrid DNN-HMM, Speaker Adaptation, Speaker Adaptive Training*

1. Introduction

Unavoidably, the development of pattern recognizers has to cope with the training sample finiteness problem. The recognizers are trained using a finite amount of training samples in hand but must accurately work over (practically infinite) unseen testing samples.

In the speech recognition field, this finiteness problem has been particularly investigated in the speaker adaptation framework [1]–[6]. Assuming the problem's existence, speech recognizers are often trained in a speaker independent (SI) mode and adapted to the unseen testing samples of new speakers.

In many speaker adaptation scenarios, only a limited amount of speech samples are available. Since the limitation of training samples makes it difficult to adapt the entire recognizer, usually just some part of it is adapted. In this partial adaptation scheme, SI recognizers are not necessarily the best choice for the initial status for the adaptation. SI training does not assume that part of the trained recognizer will be replaced in the later adaptation stage. As one solution to this inadequacy, Speaker Adaptive Training (SAT) was proposed [5], [6]. If we assume that some part of the recognizer will be replaced later, the remainder should be trained from the start, based on the assumption of such a replacement. Following this understanding, SAT jointly trains the speaker-oriented part of the recognizer and its remainder on the premise that the speaker-oriented part will be replaced in the adaptation stage.

In parallel with the advancement of speaker adaptation technologies, the speech recognizer, which has long been constructed by Gaussian Mixture Models (GMMs) and Hidden Markov Models (HMMs), is welcoming a new hybrid structure of Deep Neural Networks (DNNs) and HMMs [7]–[9]. However, despite the high utility demonstrated by DNN in various tasks (e.g., [10]), the hybrid DNN-HMM has not yet completely solved the sample finiteness problem in speaker adaptation frameworks, i.e., insufficient adaptation to unseen speakers.

Focusing on this situation, various speaker adaptation methods for DNN-HMM recognizers have been extensively studied [11]–[28]. A principal adaptation strategy in these methods is to adapt only the DNN part without changing the pre-trained HMM part. The methods, which also adopt some restriction mechanisms in DNN training to avoid the over-training problem [29], are categorized into the following two main groups: 1) restricting the network's high feature representation capability using additional small-size adaptable parameters [11]–[22], and 2) restricting the network's capability by incorporating some regularization terms in the adaptation stage [23], [24]. The first group of methods are further subdivided as follows: 1) adapting only the linear networks inserted into an SI DNN [11]–[15], 2) adapting such limited size augmented features as *speaker code* [16], [17] or *i-vector* [18]–[20] in SI DNN, and 3) adapting speaker dependent (SD) parameters embedded in the node activation function of SI DNN [21], [22]. A common maneuver by the second group of methods is to secure

Manuscript received February 3, 2016.
Manuscript revised May 24, 2016.
Manuscript publicized July 19, 2016.

[†]The authors are with the Graduate School of Science and Engineering, Doshisha University, Kyotanabe-shi, 610–0394 Japan.
[††]The authors are with the National Institution of Information and Communications Technology, Kyoto-fu, 619–0289 Japan.
[*]Presently, with ATR-Trek.
[**]Presently, with Mitsubishi Electric Research Laboratories.
a) E-mail: eup1105@mail4.doshisha.ac.jp

DOI: 10.1587/transinf.2016SLP0010

a large DNN capability and control it with regularization. Among a number of possibilities of implementing regularization, its effect was studied using the regularization term based on either the L^2 norm of the difference between an initial SI DNN and an adapted DNN [23] or the Kullback-Leibler divergence between the outputs of an initial SI DNN and an adapted DNN [24].

The first group of speaker adaptation methods for DNN-HMM recognizers is probably efficient based on using a limited size of adaptable parameters. However, the intrinsic value of DNN employment is to fully exploit the DNN's veiled power. In addition, the second group of methods simply used SI DNN as an initial condition for adaptation, although it straightforwardly treated DNN's potential. In the light of the SI-DNN-based initialization, the first group methods were also in the same situation as the second group.

Motivated by the above analysis about the preceding speaker adaptation methods for the hybrid DNN-HMM and the advantages of the SAT concept, we proposed a new SAT-based speaker adaptation scheme for the DNN-HMM speech recognizer [30]. Regarding the employment of SAT and DNN, our method shares a certain similarity with the recent SAT-based DNN-HMM recognizers [25]–[28]. However, it is characterized by introducing *modularity*, more precisely localizing *SD module*, in the DNN part. The network weight matrix and bias vector of one layer in the DNN is treated as the SD module and the DNN remainder is treated as the *SI module*[†]. Based on the SAT concept, multiple SD modules for training speakers and the SI module are jointly trained over the training speech data of many speakers, the trained SD modules are replaced by a new SD module for a target speaker, and only the new SD module is adapted using the speech data of the target speaker.

We previously outlined the formalization of our adaptation scheme and obtained its preliminary experimental results in the supervised adaptation scenario [30]. In this paper, we detail the formalization, discuss its relation with other SAT-based approaches, and show the effectiveness of our method in supervised and unsupervised adaptation scenarios.

2. Preparation

2.1 Speaker Adaptation: General Framework and Conventional Training Scheme

Speaker adaptation generally assumes that an acoustic model in the speech recognizer consists of two types of parameter sets: *a seed parameter set* ($\boldsymbol{\Lambda}$) and *an adaptation parameter set* ($\boldsymbol{g}_t$ for adaptation-target speaker t). Seed parameter set $\boldsymbol{\Lambda}$ determines the initial status of the acoustic model before adaptation; $\boldsymbol{g}_t$ adapts the acoustic model state initialized by $\boldsymbol{\Lambda}$ for target speaker t.

[†]The matrices of multiple layers can be used as the SD module. However, because a small-size SD module is clearly favorable for memorization, we focus on the one-layer SD module.

Conventionally, the seed and adaptation parameter sets are trained in the following two-stage manner.

First, in the training stage, the seed parameter set is estimated in the SI training using the data spoken by many speakers as follows:

$$\overline{\boldsymbol{\Lambda}}_{\mathrm{SI}} = \arg\min_{\boldsymbol{\Lambda}_{\mathrm{SI}}} E_{\mathrm{SI}}(\boldsymbol{\Lambda}_{\mathrm{SI}}), \tag{1}$$

where $\boldsymbol{\Lambda}_{\mathrm{SI}}$ is the seed parameter set prepared for the SI training and $\overline{\boldsymbol{\Lambda}}_{\mathrm{SI}}$ is the $\boldsymbol{\Lambda}_{\mathrm{SI}}$'s state that minimizes error function E_{SI} that represents the recognizer's accuracy over the training data of many speakers[††]. No adaptation parameters are involved here.

Next, in the adaptation stage, $\boldsymbol{g}_t$ is optimized as

$$\overline{\boldsymbol{g}}_t = \arg\min_{\boldsymbol{g}_t} E_{\mathrm{SD}t}(\overline{\boldsymbol{\Lambda}}_{\mathrm{SI}}, \boldsymbol{g}_t), \tag{2}$$

where $E_{\mathrm{SD}t}$ is the error function defined over the speech data of target speaker t. In the adaptation, $\overline{\boldsymbol{\Lambda}}_{\mathrm{SI}}$ is used as the seed status of the acoustic model and is fixed. The acoustic model is improved using $\overline{\boldsymbol{g}}_t$ for target speaker t, or in other words, it is adapted for speaker t.

2.2 Speaker Adaptive Training-Based Speaker Adaptation

As for the general speaker adaptation in Sect. 2.1, the SAT-based framework uses both the seed and adaptation parameters. However, unlike the conventional estimation of Eq. (1), it jointly estimates seed parameter set $\boldsymbol{\Lambda}_{\mathrm{SAT}}$ and multiple adaptation parameter sets $\boldsymbol{G} = (\boldsymbol{g}_1, \cdots, \boldsymbol{g}_s, \cdots, \boldsymbol{g}_S)$, where $\boldsymbol{g}_s$ is an adaptation parameter set for training speaker s and S is the number of speakers in the training data pool. This joint estimation procedure is formalized as

$$(\overline{\boldsymbol{\Lambda}}_{\mathrm{SAT}}, \overline{\boldsymbol{G}}) = \arg\min_{(\boldsymbol{\Lambda}_{\mathrm{SAT}}, \boldsymbol{G})} E_{\mathrm{SAT}}(\boldsymbol{\Lambda}_{\mathrm{SAT}}, \boldsymbol{G}), \tag{3}$$

where $E_{\mathrm{SAT}}(\boldsymbol{\Lambda}_{\mathrm{SAT}}, \boldsymbol{G}) = \sum_{s=1}^{S} E_{\mathrm{SD}s}(\boldsymbol{\Lambda}_{\mathrm{SAT}}, \boldsymbol{g}_s)$, and $E_{\mathrm{SD}s}$ is the error function defined over the speech data of training speaker s. Here, $\boldsymbol{\Lambda}_{\mathrm{SAT}}$ is not fixed but trained while $\boldsymbol{g}_s$ is switched for every speaker s, and all of the adaptation parameter sets $\boldsymbol{G}$ are also trained.

In the adaptation stage, all of the adaptation parameter sets in $\boldsymbol{G}$ are replaced with adaptation parameter set $\boldsymbol{g}_t$ that is newly prepared for target speaker t. Then, only $\boldsymbol{g}_t$ is adapted using his/her speech data as follows:

$$\overline{\boldsymbol{g}}_t = \arg\min_{\boldsymbol{g}_t} E_{\mathrm{SD}t}(\overline{\boldsymbol{\Lambda}}_{\mathrm{SAT}}, \boldsymbol{g}_t). \tag{4}$$

Here, similar to Eq. (2), $\overline{\boldsymbol{\Lambda}}_{\mathrm{SAT}}$ is used as the seed for adaptation and is fixed.

In the joint estimation stage of Eq. (3), $\boldsymbol{\Lambda}_{\mathrm{SAT}}$ is optimized on the premise that $\boldsymbol{g}_t$ is adapted in conjunction with the use of $\overline{\boldsymbol{\Lambda}}_{\mathrm{SAT}}$ in the adaptation stage of Eq. (4). The con-

[††]Throughout this paper, *overline* represents the optimized status of its corresponding parameter set.

sistency between the seed model estimation stage and the adaptation stage naturally helps $\overline{\boldsymbol{\Lambda}}_{\mathrm{SAT}}$ work better than $\overline{\boldsymbol{\Lambda}}_{\mathrm{SI}}$ in the adaptation.

In the case of the GMM-HMM recognizer, the mean vectors, the covariance matrices, and the mixture weights of GMM are generally used as $\boldsymbol{\Lambda}_{\mathrm{SAT}}$. In such cases, $\boldsymbol{g}_t$ is the transformation matrices (plus their corresponding bias vectors) for the mean vectors and the covariance matrices and are often adapted using Maximum Likelihood Linear Regression (MLLR) [1] or feature space MLLR (fMLLR) [2] methods.

2.3 Hybrid DNN-HMM Speech Recognizers

2.3.1 Overview

Our SAT-based speaker adaptation scheme uses the hybrid DNN-HMM speech recognizer whose structure is illustrated in Fig. 1. In this hybrid structure, the GMM part of the conventional GMM-HMM recognizer is replaced by the DNN. The adopted DNN is a standard MLP network whose node has trainable connection weights and a trainable bias. We denote the weight matrix and the bias vector between network layers L_{l-1} and L_l as $\boldsymbol{W}_l$ and $\boldsymbol{b}_l$. We also denote the pair of $\boldsymbol{W}_l$ and $\boldsymbol{b}_l$ as $\boldsymbol{\lambda}_l = \{\boldsymbol{W}_l, \boldsymbol{b}_l\}$. The example DNN in the figure has seven layers $\{\mathrm{L}_0, \mathrm{L}_1, \cdots, \mathrm{L}_6\}$. For simplicity, we omit the biases in all of the illustrations in this paper.

Given a speech input, the recognizer first converts it to a sequence of acoustic feature vectors $\boldsymbol{X} = \{\boldsymbol{x}_\tau; \tau = 1, \cdots, \mathcal{T}\}$, where $\boldsymbol{x}_\tau$ is the τ-th acoustic feature vector[†] in $\boldsymbol{X}$ and $\mathcal{T}$ is the length of $\boldsymbol{X}$. Next, the recognizer estimates posterior probability $p(\mathcal{W}_c|\boldsymbol{X})$ for the pair of $\boldsymbol{X}$ and each of the possible word sequences $\{\mathcal{W}_c; c = 1, \cdots, C\}$ and classifies $\boldsymbol{X}$ to the class with the largest posterior probability value among the C classes. Clearly, the recognizer's classification accuracy depends on the estimation quality for $p(\mathcal{W}_c|\boldsymbol{X})$.

In the DNN-HMM recognizer, $p(\mathcal{W}_c|\boldsymbol{X})$ is estimated by $p(\mathcal{W}_c|\boldsymbol{X})_{\{\boldsymbol{\Lambda}_{\mathrm{DNN}}, \boldsymbol{\Lambda}_{\mathrm{HMM}}\}}$, which is a function of $\boldsymbol{\Lambda}_{\mathrm{DNN}}$ (trainable parameters of DNN) and $\boldsymbol{\Lambda}_{\mathrm{HMM}}$ (trainable parameters of HMM). Accordingly, the training seeks the state

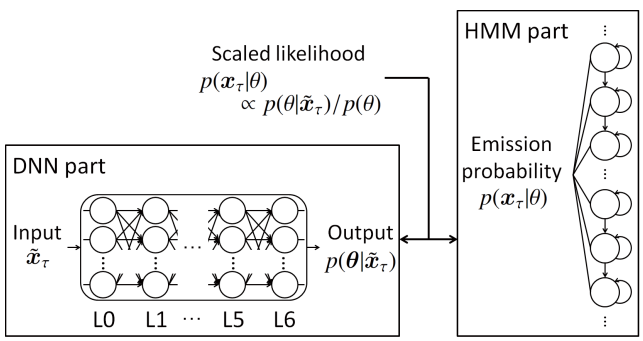


Fig. 1 Structure of hybrid DNN-HMM speech recognizer

[†]The definition of $\tilde{\boldsymbol{x}}_\tau$, used in Fig. 1, will be given in the next subsection.

of $p(\mathcal{W}_c|\boldsymbol{X})_{\{\boldsymbol{\Lambda}_{\mathrm{DNN}}, \boldsymbol{\Lambda}_{\mathrm{HMM}}\}}$ that achieves the highest possible classification accuracies on testing speech data by updating $\boldsymbol{\Lambda}_{\mathrm{DNN}}$ and $\boldsymbol{\Lambda}_{\mathrm{HMM}}$.

2.3.2 Computation of Posterior Probabilities

$p(\mathcal{W}_c|\boldsymbol{X})_{\{\boldsymbol{\Lambda}_{\mathrm{DNN}}, \boldsymbol{\Lambda}_{\mathrm{HMM}}\}}$ is computed in the following divide-and-conquer manner. Based on the Bayes theorem, it is replaced with $p(\boldsymbol{X}|\mathcal{W}_c)_{\{\boldsymbol{\Lambda}_{\mathrm{DNN}}, \boldsymbol{\Lambda}_{\mathrm{HMM}}\}}$ and an estimate of prior probability $p(\mathcal{W}_c)$, which is often calculated by such language models as N-gram. Then $p(\boldsymbol{X}|\mathcal{W}_c)_{\{\boldsymbol{\Lambda}_{\mathrm{DNN}}, \boldsymbol{\Lambda}_{\mathrm{HMM}}\}}$ is calculated using such HMM probabilities as state output probability estimates $\{p(\boldsymbol{x}_\tau|\theta)_{\boldsymbol{\Lambda}_{\mathrm{DNN}}}; \tau = 1, \cdots, \mathcal{T}$, and $\theta = 1, \cdots, \Theta\}$, where Θ is the number of possible states of HMM. Here, again based on the Bayes theorem, $p(\boldsymbol{x}_\tau|\theta)_{\boldsymbol{\Lambda}_{\mathrm{DNN}}}$ is replaced by scaled likelihood $p(\theta|\boldsymbol{x}_\tau)_{\boldsymbol{\Lambda}_{\mathrm{DNN}}}/p(\theta)$. State posterior probability $p(\theta|\boldsymbol{x}_\tau)_{\boldsymbol{\Lambda}_{\mathrm{DNN}}}$ is calculated by DNN, and state prior probability $p(\theta)$ is estimated as $p(\theta)_{\boldsymbol{\Lambda}_{\mathrm{HMM}}}$, based on the frequency of the state assignment produced by HMM's forced alignment.

Estimate $p(\theta|\boldsymbol{x}_\tau)_{\boldsymbol{\Lambda}_{\mathrm{DNN}}}$ must maintain the nature of the probability function. To meet this requirement, the DNN part uses the *softmax* activation functions at its output nodes.

Similar to many recent HMM-based speech recognizers, the DNN-HMM recognizers usually adopt context dependent acoustic models. In this situation, the number of HMM states is too large to appropriately calculate state posterior probability estimate $p(\theta|\boldsymbol{x}_\tau)_{\boldsymbol{\Lambda}_{\mathrm{DNN}}}$. To circumvent this problem, the HMM states are often clustered into several thousands of sub-phonetic units, i.e., *senones*, each representing the HMM tied-state. Following this strategy, in the DNN-HMM framework, the estimate of state posterior probability $p(\theta|\boldsymbol{x}_\tau)_{\boldsymbol{\Lambda}_{\mathrm{DNN}}}$ is replaced with the estimate of senone posterior probability $p(k|\boldsymbol{x}_\tau)_{\boldsymbol{\Lambda}_{\mathrm{DNN}}}$ that is calculated using network output $y_{\tau k}$, where k is the senone class index ($k = 1, \cdots, K$), assuming that K is the number of senones.

In most cases of the recent hybrid DNN-HMM speech recognizer, the concatenation of several acoustic feature vectors $\tilde{\boldsymbol{x}}_\tau = \{\boldsymbol{x}_{\tau-\tau_c}, \cdots, \boldsymbol{x}_\tau, \cdots, \boldsymbol{x}_{\tau+\tau_c}\}$ is used in $p(\theta|\boldsymbol{x}_\tau)_{\boldsymbol{\Lambda}_{\mathrm{DNN}}}$ instead of $\boldsymbol{x}_\tau$, where τ_c is a small natural number. Accordingly, $p(k|\boldsymbol{x}_\tau)_{\boldsymbol{\Lambda}_{\mathrm{DNN}}}$ is further replaced by $p(k|\tilde{\boldsymbol{x}}_\tau)_{\boldsymbol{\Lambda}_{\mathrm{DNN}}}$.

2.3.3 Training Procedure

The DNN-HMM recognizer is basically trained using a set of speech data spoken by multiple speakers in the SI mode.

The training procedure is twofold: one for the HMM part and another for the DNN part. The HMM part is first trained within the training for the GMM-HMM speech recognizer. The DNN part is subsequently trained using the senone labels produced by the forced alignment with the baseline GMM-HMM speech recognizer. These labels are used as teaching signals to train the acoustic feature vector inputs. Using these labels, such an objective function as Cross Entropy (CE) error is defined, and the DNN parameters are optimized under a condition that minimize the defined objective function.

Since the SI-training procedure for the DNN part is the same as the initialization stage of our SAT-based DNN training, we minutely describe it in the framework of our proposed method.

3. New Speaker Adaptive Training Localizing Speaker Modules in DNN

3.1 Overview

In principle, when a speech recognizer is developed for recognizing such continuous speech samples as spoken sentences, the whole recognizer must be optimized to increase the recognition accuracy for such long speech inputs. However, because the number of possible long speech unit classes is often astronomically large and collecting a reasonable amount of training samples is almost impossible for such an extremely large number of classes, most speech recognizers are trained to aim at the accurate classification of such short units as words, phonemes, and senones. Following this general training guideline for large-scale recognizers, we set the evaluation criterion of our recognizer to word recognition accuracy and train/adapt its DNN part to increase its feature-vector-wise senone classification accuracy.

The SAT-based scheme consists of the following three stages: 1) initialization, 2) SAT, and 3) speaker adaptation. All of the training procedures in these stages adopt the criterion of minimizing the CE error function between the DNN outputs and the senone labels that are produced by the forced alignment with the GMM-HMM speech recognizer trained on the Boosted Maximum Mutual Information (BMMI) criterion [31]. In the initialization stage, we train the whole DNN part in the standard SI discriminative training fashion. In the SAT stage, we localize the SD modules in the DNN part and train the entire DNN while switching the SD modules along with the speaker change in the training data set. Finally, in the speaker adaptation stage, we train only a new SD module for a target speaker using his/her speech data.

3.2 Training Procedures

3.2.1 Initialization Stage

We assume that speech samples $\mathcal{X} = \{\boldsymbol{X}_n; n = 1, \cdots, N\}$ are available to train the DNN part, where $\boldsymbol{X}_n$ is the n-th training sample and N is the number of such samples. The samples of $\mathcal{X}$ are spoken by many speakers.

In Fig. 2, we illustrate our initialization procedure for a 7-layer example of a DNN part, where $\boldsymbol{\lambda}_l^{\text{SI-DNN}} = \{\boldsymbol{W}_l^{\text{SI-DNN}}, \boldsymbol{b}_l^{\text{SI-DNN}}\}$ $(l = 1, \cdots, 6)$. As mentioned before, no bias vectors are depicted. Given training feature vector input $\boldsymbol{x}_\tau^{(n)}$ of training sample $\boldsymbol{X}_n$ to input layer L_0, the DNN part emits network outputs $\{y_{\tau k}^{(n)}; k = 1, 2, \cdots, K\}$ at output layer L_6. The largest output represents the senone classification decision, which is evaluated using the correct class information determined by the forced alignment with the BMMI-trained GMM-HMM speech recognizer.

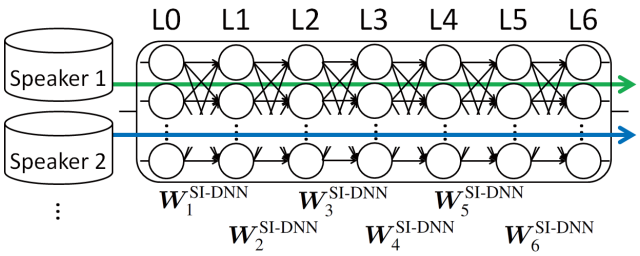


Fig. 2 DNN structure and SI training procedure for initialization stage

For discussion generality, we consider the initialization of L-layer DNN parameters $\boldsymbol{\Lambda}_{\text{SI-DNN}}$ $(= \{\boldsymbol{\lambda}_l^{\text{SI-DNN}}; l = 1, \cdots, L\})$. To accelerate the initialization training, we preliminarily train $\boldsymbol{\Lambda}_{\text{SI-DNN}}$ using the Restricted Boltzmann Machine (RBM) [32] in the greedy layer-wise manner [10]. For later discussions, we denote the RBM-trained state of $\boldsymbol{\Lambda}_{\text{SI-DNN}}$ as $\boldsymbol{\Lambda}_{\text{RBM}}$ $(= \{\boldsymbol{\lambda}_l^{\text{RBM}}; l = 1, \cdots, L\} = \{\{\boldsymbol{W}_l^{\text{RBM}}, \boldsymbol{b}_l^{\text{RBM}}\}; l = 1, \cdots, L\})$. Next, we conduct the following regularization-incorporated CE error minimization:

$$\overline{\boldsymbol{\Lambda}}_{\text{SI-DNN}} = \arg\min_{\boldsymbol{\Lambda}_{\text{SI-DNN}}} \left\{ E_{\text{CE}}(\boldsymbol{\Lambda}_{\text{SI-DNN}}; \mathcal{X}) + \frac{\alpha}{2} R(\boldsymbol{\Lambda}_{\text{SI-DNN}}) \right\}, \tag{5}$$

where E_{CE} is the accumulated CE error defined as

$$E_{\text{CE}}(\boldsymbol{\Lambda}_{\text{SI-DNN}}; \mathcal{X}) = -\sum_{n=1}^{N} \sum_{\tau=1}^{\mathcal{T}} \sum_{k=1}^{K} t_{\tau k}^{(n)} \ln y_{\tau k}^{(n)}. \tag{6}$$

Here, $t_{\tau k}^{(n)}$ is the teaching signal for $y_{\tau k}^{(n)}$ that indicates 1 when $\boldsymbol{x}_\tau^{(n)}$ belongs to senone class k but indicates 0 otherwise, R is a regularization term, and α is a non-negative constant regularization coefficient. In this minimization, $\boldsymbol{\Lambda}_{\text{RBM}}$ works as the initial status of $\boldsymbol{\Lambda}_{\text{SI-DNN}}$.

We adopt the regularization to avoid the over-training problem that is often caused by large-size DNNs. The regularization procedure will be described in Sect. 3.2.4. Then regularization-incorporated minimization is done using the following Error Back Propagation (EBP) parameter update:

$$\boldsymbol{\Lambda}_{\text{SI-DNN}} \leftarrow \boldsymbol{\Lambda}_{\text{SI-DNN}} - \epsilon \frac{\partial \left\{ E_{\text{CE}}(\boldsymbol{\Lambda}_{\text{SI-DNN}}; \mathcal{X}) + \frac{\alpha}{2} R(\boldsymbol{\Lambda}_{\text{SI-DNN}}) \right\}}{\partial \boldsymbol{\Lambda}_{\text{SI-DNN}}}, \tag{7}$$

where ϵ is the positive scalar training rate.

We use the above estimated parameters $\overline{\boldsymbol{\Lambda}}_{\text{SI-DNN}}$ for a baseline SI recognizer and also as an initial status of the subsequent SAT stage.

3.2.2 Speaker Adaptive Training Stage

In Fig. 3, we illustrate the procedure of conducting SAT with SD module allocation. Because of DNN's multi-layer structure, the SD modules can be allocated to any of the intermediate layers. In the figure, as an example, we allocate SD

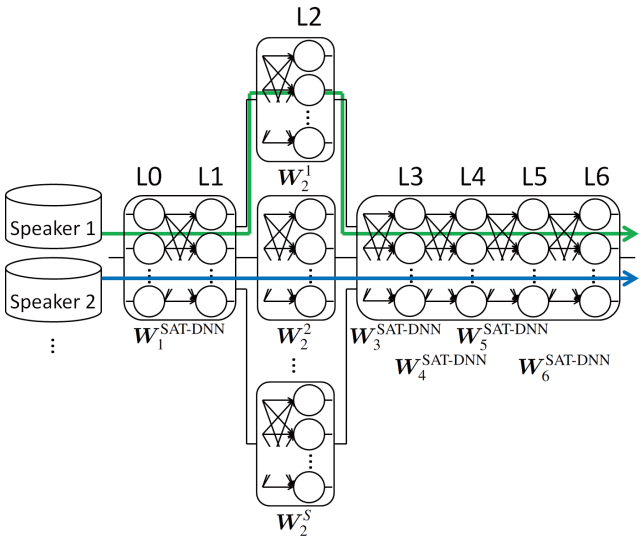


Fig. 3 DNN structure and SAT training procedure for SAT stage

modules $\boldsymbol{G}_2 = \{\boldsymbol{g}_2^1, \boldsymbol{g}_2^2, \cdots, \boldsymbol{g}_2^S\}$ to L_2, where S is the number of speakers in the training data set, $\boldsymbol{g}_2^s$ $(= \{\boldsymbol{W}_2^s, \boldsymbol{b}_2^s\})$ is the SD module parameters for training speaker s, and $\boldsymbol{\Lambda}_{\text{SAT-DNN}}$ $(= \{\boldsymbol{\lambda}_1^{\text{SAT-DNN}}, \boldsymbol{\lambda}_3^{\text{SAT-DNN}}, \cdots, \boldsymbol{\lambda}_L^{\text{SAT-DNN}}\})$ are the parameters of the network layers other than the SD module layer, where $\boldsymbol{\lambda}_l^{\text{SAT-DNN}} = \{\boldsymbol{W}_l^{\text{SAT-DNN}}, \boldsymbol{b}_l^{\text{SAT-DNN}}\}$.

The figure shows two example cases of the training procedure: one for using the speech data of Speaker 1 $(s = 1)$ and another for Speaker 2 $(s = 2)$. When using the data of Speaker 1, only the nodes of the SD module for Speaker 1 are connected with the nodes of the adjacent layers; the nodes of the other SD modules are disconnected from the nodes in the adjacent layers. The green line depicts this situation, and the training is executed only along this path. Similarly, the blue line depicts the situation when using the data of Speaker 2. Each SD module is trained only using its corresponding speaker's data, but the other part of the network, i.e., $\boldsymbol{\Lambda}_{\text{SAT-DNN}}$, is trained using the data of all of the S speakers.

Again for discussion generality, we consider the case of using SD module layer $\mathrm{L}_{l_{\text{SD}}}$ of the L-layer DNN. The SAT procedure for this general setting is formalized as follows:

$$(\overline{\boldsymbol{\Lambda}}_{\text{SAT-DNN}}, \overline{\boldsymbol{G}}_{l_{\text{SD}}}) = \underset{(\boldsymbol{\Lambda}_{\text{SAT-DNN}}, \boldsymbol{G}_{l_{\text{SD}}})}{\arg\min} \left\{ E_{\text{SAT-CE}}(\boldsymbol{\Lambda}_{\text{SAT-DNN}}, \boldsymbol{G}_{l_{\text{SD}}}; \mathcal{X}) + \frac{\beta}{2} R(\boldsymbol{G}_{l_{\text{SD}}}) \right\}, \tag{8}$$

where

$$E_{\text{SAT-CE}}(\boldsymbol{\Lambda}_{\text{SAT-DNN}}, \boldsymbol{G}_{l_{\text{SD}}}; \mathcal{X}) = \sum_{s=1}^{S} E_{\text{CE}}(\boldsymbol{\Lambda}_{\text{SAT-DNN}}, \boldsymbol{g}_{l_{\text{SD}}}^s; \mathcal{X}_s). \tag{9}$$

Here, $\boldsymbol{\Lambda}_{\text{SAT-DNN}} = \{\boldsymbol{\lambda}_l^{\text{SAT-DNN}}; l = 1, \cdots, l_{\text{SD}} - 1, l_{\text{SD}} + 1, \cdots, L\}$, $\boldsymbol{G}_{l_{\text{SD}}} = \{\boldsymbol{g}_{l_{\text{SD}}}^s; s = 1, \cdots, S\}$, $\boldsymbol{g}_{l_{\text{SD}}}^s$ is the parameters of the SD module for training speaker s, $\mathcal{X}_s$ is the speech data spoken by training speaker s, and β is a non-negative scalar regularization coefficient. The definition of $E_{\text{CE}}(\boldsymbol{\Lambda}_{\text{SAT-DNN}}, \boldsymbol{g}_{l_{\text{SD}}}^s; \mathcal{X}_s)$ is basically the same as the accumulated CE error of Eq. (6), except that SD module $\boldsymbol{g}_{l_{\text{SD}}}^s$ is switched here for every training speaker. The training aims to find the optimal states of both $\boldsymbol{\Lambda}_{\text{SAT-DNN}}$ and $\boldsymbol{G}_{l_{\text{SD}}}$ that correspond to the minimum CE error situation achieved in conjunction with the SD-module-based regularization. Here, we define the regularization term only using $\boldsymbol{G}_{l_{\text{SD}}}$, taking into account the training data limitation for each training speaker. The regularization details will be explained in Sect. 3.2.4. With regard to the initial setting of $\boldsymbol{\Lambda}_{\text{SAT-DNN}}$ and $\boldsymbol{G}_{l_{\text{SD}}}$, we initialize them using $\overline{\boldsymbol{\Lambda}}_{\text{SI-DNN}}$, as described in Sect. 3.2.1. The minimization in Eq. (8) is conducted using the EBP parameter update rule.

The formula in Eq. (8) is basically the same as the original SAT formula of Eq. (3) except for the presence of the regularization term.

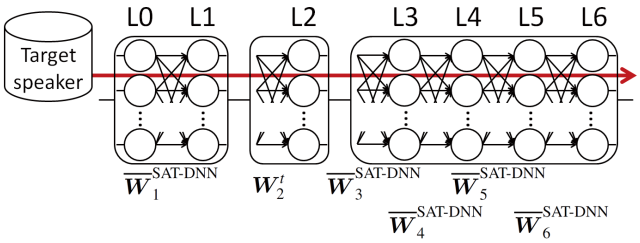


Fig. 4 DNN structure and adaptation training procedure for speaker adaptation stage

3.2.3 Speaker Adaptation Stage

In the adaptation stage, all of the adaptation parameter sets in $\boldsymbol{G}_{l_{\text{SD}}}$ are removed and replaced with a new adaptation parameter set $\boldsymbol{g}_{l_{\text{SD}}}^t$ for target speaker t. Then, only $\boldsymbol{g}_{l_{\text{SD}}}^t$ is adapted using his/her speech data.

In Fig. 4, we illustrate the speaker adaptation procedure, assuming we set the SD modules to L_2 of the DNN trained in the SAT stage. In the figure, $\boldsymbol{g}_2^t$ $(= \{\boldsymbol{W}_2^t, \boldsymbol{b}_2^t\})$ represents the SD module parameters for target speaker t, and $\{\overline{\boldsymbol{\lambda}}_l^{\text{SAT-DNN}}; l = 1, 3, \cdots, L\}$ represents the network parameters optimized in the SAT stage. The inserted SD module $\boldsymbol{g}_2^t$ is adapted to $\overline{\boldsymbol{g}}_2^t$ using the speech data of speaker t. An important point here is that only the inserted SD module is adapted; the remaining DNN part is fixed.

In the scenario where the SD module layer is $\mathrm{L}_{l_{\text{SD}}}$, the adaptation stage is formalized as follows:

$$\overline{\boldsymbol{g}}_{l_{\text{SD}}}^t = \underset{\boldsymbol{g}_{l_{\text{SD}}}^t}{\arg\min} \left\{ E_{\text{CE}}(\overline{\boldsymbol{\Lambda}}_{\text{SAT-DNN}}, \boldsymbol{g}_{l_{\text{SD}}}^t; \mathcal{X}_t) + \frac{\gamma}{2} R(\boldsymbol{g}_{l_{\text{SD}}}^t) \right\}, \tag{10}$$

where $\boldsymbol{g}_{l_{\text{SD}}}^t$ is the SD module parameters for speaker t, $\mathcal{X}_t$ is the speech data spoken by speaker t for adaptation, and γ is a regularization coefficient. The regularized minimization of E_{CE} is conducted only with respect to $\boldsymbol{g}_{l_{\text{SD}}}^t$, just using $\mathcal{X}_t$.

Similar to the SAT stage, the minimization in Eq. (10) is executed with the regularized EBP training. The regularization here will be described in Sect. 3.2.4.

The formula in Eq. (10) is basically the same as the original SAT formula of Eq. (4) except for the presence of the regularization term. Accordingly, as suggested by the effectiveness of the original SAT scheme, the adaptation using $\overline{\boldsymbol{\Lambda}}_{\text{SAT-DNN}}$ is expected to work better than the adaptation using $\overline{\boldsymbol{\Lambda}}_{\text{SI-DNN}}$.

3.2.4 Regularization

To avoid over-training, we adopt regularization-incorporated error minimization in all of our DNN training procedures. Among various possibilities, we especially use the L^2-norm-based regularization term.

For the initialization stage of the SAT-based scheme or the standard SI training, we define the regularization term as follows:

$$R(\boldsymbol{\Lambda}_{\text{SI-DNN}}) = \sum_{l=1}^{L} \left(\| \boldsymbol{W}_l^{\text{SI-DNN}} \|^2 + \| \boldsymbol{b}_l^{\text{SI-DNN}} \|^2 \right), \quad (11)$$

which is often referred to as *weight decay* in the neural network research field.

For the SAT stage, we adopt the following regularization term:

$$R(\boldsymbol{G}_{l_{\text{SD}}}) = \sum_{s=1}^{S} \left(\|\boldsymbol{W}_{l_{\text{SD}}}^{s} - \overline{\boldsymbol{W}}_{l_{\text{SD}}}^{\text{SI-DNN}}\|^2 + \|\boldsymbol{b}_{l_{\text{SD}}}^{s} - \overline{\boldsymbol{b}}_{l_{\text{SD}}}^{\text{SI-DNN}}\|^2 \right), \quad (12)$$

which was previously called L^2 *prior regularization* [23]. This regularization softly ties $\boldsymbol{G}_{l_{\text{SD}}}$ to $\overline{\boldsymbol{\lambda}}_{l_{\text{SD}}}^{\text{SI-DNN}} = \{\overline{\boldsymbol{W}}_{l_{\text{SD}}}^{\text{SI-DNN}}, \overline{\boldsymbol{b}}_{l_{\text{SD}}}^{\text{SI-DNN}}\}$. In the SAT stage, each SD module is trained using just one speaker's data, an amount that is often limited. But the training of the remaining DNN part, $\boldsymbol{\Lambda}_{\text{SAT-DNN}}$, can be done using the data of all of the training speakers. Considering this difference in data size, we define the regularization term only for $\boldsymbol{G}_{l_{\text{SD}}}$.

For the speaker adaptation stage, we also define the L^2 prior-based regularization term for the SD module of target speaker t, $\boldsymbol{g}_{l_{\text{SD}}}^{t}$, as follows:

$$R(\boldsymbol{g}_{l_{\text{SD}}}^{t}) = \|\boldsymbol{W}_{l_{\text{SD}}}^{t} - \boldsymbol{W}_{l_{\text{SD}}}^{anchor}\|^2 + \|\boldsymbol{b}_{l_{\text{SD}}}^{t} - \boldsymbol{b}_{l_{\text{SD}}}^{anchor}\|^2, \quad (13)$$

where $\boldsymbol{g}_{l_{\text{SD}}}^{t}$ is softly tied to its anchor state, $\boldsymbol{g}_{l_{\text{SD}}}^{anchor} = \{\boldsymbol{W}_{l_{\text{SD}}}^{anchor}, \boldsymbol{b}_{l_{\text{SD}}}^{anchor}\}$, such that it does not over-fit $\mathcal{X}_t$.

$\boldsymbol{g}_{l_{\text{SD}}}^{anchor}$ can be prepared in several different ways. Simple ways include using a network initialized by small random numbers and using the SD module of the SI-trained network, $\overline{\boldsymbol{\lambda}}_{l_{\text{SD}}}^{\text{SI-DNN}}$. Compared to the former, the latter would better fit to the anchor for speaker adaptation (or adaptation-oriented regularization): The latter module is already trained for speech recognition. However, the SI training reflects none of the SAT concept. Obviously, the anchor state should import the SAT concept at least to some extent, since the anchor is used for the adaptation of the SAT-based network. Taking this into account, we adopt the following three-step method: 1) remove $\overline{\boldsymbol{G}}_{l_{\text{SD}}}$ from the DNN part trained by Eq. (8); 2) insert $\overline{\boldsymbol{\lambda}}_{l_{\text{SD}}}^{\text{SI-DNN}}$ into the SD module layer of the DNN part; 3) re-train (in the SI training sense) the SD module using all the training speech data, while fixing the remainder of the DNN part, i.e., $\overline{\boldsymbol{\Lambda}}_{\text{SAT-DNN}}$. The resulting state of the SD module is used as $\boldsymbol{g}_{l_{\text{SD}}}^{anchor}$.

The anchor state produced in the above way is also used as the initial status of the target speaker's SD module in the adaptation stage. This initialization is expected to be effective for successive adaptation, because the anchor state is trained so as to utilize the optimized SAT-based network.

3.3 Relations with Preceding SAT-Based DNN-HMM Speech Recognizers

In parallel with our proposed scheme, several speaker normalization techniques for DNN-HMM recognizers have been investigated [25]–[28]. These techniques adopted canonical DNN modeling of a virtual representative speaker, which is different from a standard SI-training-based DNN. The canonical DNN represented one (even a virtual) speaker in a compact form, and this nature is clearly common to the SAT concept.

The design of the canonical DNN was based in common on the speaker normalization applied to the input features, although the normalization was done differently. It was conducted using GMM-HMM recognizers [25], [26], DNN-HMM recognizers [28], and additionally adopting a speaker normalization network whose input was i-vector [27]. Compared with these techniques based on input feature level normalization, our scheme, in which the normalization (or adaptation) is embedded in the deep layers of DNN, is characterized by a positive use of DNN power.

4. Experiments

4.1 Data Preparation

We tested our proposed method on the difficult, lecture speech data of the TED Talks[†] corpus and prepared three data sets: training, validation, and testing.

The training data set consisted of the speech data of 300 speakers; the data of each speaker were about 15 minutes long. The total length of the training data was about 75 hours[††].

The validation data set consisted of the speech data of ten speakers, each of whom was different from the speakers in the training data set. This set was used for finding

[†]http://www.ted.com

[††]In our previous publication [30], we mistakenly reported that this length was 150 hours.

the optimal values of the hyper-parameters, which produced high recognition accuracies over the set itself, such as the learning rate 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 [33]. The speakers in this testing data were different from those both in the training and validation data sets. Each speaker's data ranged from 2.6 to 16.5 minutes, and the average length of the testing data was about 8.5 minutes.

4.2 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 Mel-scale Frequency Cepstrum Coefficients (MFCCs), logarithmic power (log-power), 12 ΔMFCCs, Δlog-power, 12 ΔΔMFCCs, and ΔΔlog-power, where Δ and ΔΔ denote the first and second derivatives. The acoustic feature vectors had 39 dimensions. Next, the 11 adjacent acoustic feature vectors were concatenated as a 429-dimensional input to the DNN part. Each element of the input vectors was normalized so that its mean and variance became 0 and 1.

4.3 Evaluated Recognizers

To evaluate our SAT-based adaptation scheme, we compared the following four recognizers: 1) the SAT-based DNN-HMM recognizer (*SAT recognizer*), 2) its adapted version (for a target speaker) (*Speaker-Adapted SAT (SA-SAT) recognizer*), 3) the SI-based DNN-HMM recognizer (*SI recognizer*), and 4) its adapted version (*Speaker-Adapted SI (SA-SI) recognizer*). Here, the SI recognizer works as a baseline case in the experiments, and the SI and SA-SI recognizers are the counterparts to the SAT and SA-SAT recognizers, both of which are based on our proposed scheme.

We adopted a simple seven-layer DNN as the baseline SI recognizer. The whole network was first pre-trained by layer-wise RBM training and successively trained using the CE error minimization over the training data set (Fig. 2).

The SA-SI recognizer was produced by adapting one of the SI recognizer's intermediate network layers, which corresponded to an SD module, using the speech data of an adaptation-target speaker selected from the 28 testing speakers. To avoid the over-training problem due to the limited amount of adaptation data, we applied the regularization term of Eq. (13) to the update of the weights and biases of layer l_{SD}, setting $\boldsymbol{g}^{anchor}_{l_{\mathrm{SD}}}$ to $\overline{\boldsymbol{\lambda}}^{\text{SI-DNN}}_{l_{\mathrm{SD}}}$, when adapting the SI recognizer to the SA-SI recognizer.

We built the SAT recognizer by the following procedures, as described in Sect. 3. We adopted the baseline SI recognizer as the initial status of the SAT recognizer and inserted 300 SD modules into the baseline recognizer. Here, the number of SD modules is the same as that of the training speakers. We next generated a trained network along the SAT-based optimization course of Eq. (8). Finally, we completed the SAT recognizer by replacing the 300 used SD modules with a new SD module, which was the anchor module described in Sect. 3.2.4. This new SD module worked as the initial status for successive adaptations.

The SA-SAT recognizer was produced by adapting only the SD module of the SAT recognizer in the speaker-by-speaker mode, where an adaptation-target speaker was selected from the 28 testing speakers.

In all of our recognizers, the HMM part used the context-dependent acoustic model and used the 4-gram language model that was trained over the transcriptions of TED Talks, News Commentary, and English Gigaword [34]. The baseline GMM-HMM recognizer was trained with BMMI training, which was used to obtain the senone alignment labels for the DNN training and the adaptation. During the DNN training, HMM's state transition probability was fixed. In the decoding phase, the DNN-HMM recognizers used the scaled likelihood calculated by the DNN in place of the state output probability calculated using the GMM, as described in Sect. 2.3.2. In this experiment, the DNN module in our recognizers had seven layers (Fig. 2) and used 429 input nodes, 4909 output nodes, and 512 nodes for all of the intermediate layers. Here, the number of output nodes was the same as the senone classes. A sigmoid activation function was used for the input and intermediate layer nodes; a softmax activation function was used for the output nodes.

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

4.4 Evaluation Procedures

In terms of the availability of reference word transcriptions, we evaluated our proposed method in two different experimental procedures: supervised adaptation and unsupervised adaptation. In the supervised adaptation procedure, we adapted the SD modules using the (correct) reference transcriptions of the adaptation speech data. In the unsupervised adaptation procedure, we did the adaptation using transcriptions that were generated by decoding with the SI recognizer in place of the reference transcriptions. Here, we calculated the word confidence measure values based on the confusion networks [35] for the decoded transcriptions. Only the speech segments, whose measure values exceeded a preset threshold, were adopted for the adaptation.

To circumvent the problem of a closed-form evaluation, we adopted the four-times cross-validation (CV) experiment paradigm where the speech data of every testing speaker were divided into four groups. In this paradigm, the validation of the adaptation result consisted of the SD module adaptation using three of the four data groups and the testing of the adapted SD module using the remaining

data group. We repeated this validation four times by changing the combination of three groups for adaptation and one group for testing. The recognition accuracies (word error rates) in later discussions are the averages obtained by performing this CV-based evaluation over the speech data of all the adaptation speakers.

4.5 Mini-Batch-Based Error Minimization

In EBP-based CE error minimization for producing SI and SAT recognizers, we repeated a *training epoch*, where every sample in the training data set was used once for the network parameter updates. To accelerate the experiments with GPU's high computation power, we adopted a *mini-batch* mode minimization, which was a mix of the batch and sequential modes that repeated the error calculation and parameter updates over every set of some selected training samples. Especially in the case of using Eq. (8) for the SAT recognizer, because we had to feed the training speech data to the network while switching the SD module, the mini-batch of speech data was required to be only composed of the speech data of a single speaker. To meet the requirements of the speech data preparation, we implemented the SAT procedure in the following way. For every training epoch, we first made mini-batches, each of which consisted of the speech data of a single training speaker, over the whole training data. Next we randomized their order to avoid unexpected convergence to poor local optima in the EBP error minimization and conducted error calculation and parameter updates while switching the randomized mini-batches and the SD modules in the speaker-by-speaker manner. Then we repeated the epochs N_{epoch} times, where N_{epoch} was the maximum number of epoch repetitions.

4.6 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 the frame-level recognition accuracies over the validation data set. If the recognition accuracy increased over the validation data set, 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 maximum recognition accuracy in the preceding training epochs, and the training for these replaced weights and biases was restarted using the halved learning rate.

Especially in the SAT stage, we used the average frame-level recognition accuracy obtained from the trial adaptation using the validation data. At every training epoch in this stage, we first adopted the resultant DNN (the SI module plus the SD module) of the previous epoch as an initial network status for adaptation. Using the speech data in the validation data set, we performed a trial adaptation while switching the SD module and its corresponding speaker data. During the adaptation, the SI module was fixed. After the adaptation, we calculated the frame-level recognition accuracy in the speaker-by-speaker manner. In the same way as the evaluation of the trained recognizers, we repeated the trial adaptation and accuracy calculation in the CV paradigm. The averaged frame-level recognition accuracy we obtained was used to control the learning rate at every epoch.

For the SI recognizer, we set the initial value of the learning rate to 0.004 and repeated 20 epochs, where the learning rate was controlled based on the above update rule. When producing the anchor SD module (for the SAT recognizer) and the speaker adaptive trained network of Eq. (8), we adopted the same settings as in the SI recognizer. For the regularization terms, α in Eq. (5) was set to 0.0 based on preliminary experiments: 0.1 for β in Eq. (8).

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

For each allocation of the SD module layer, the above procedures for setting the hyper-parameters were repeated. Accordingly, regardless of the SD module positioning, all of the training in the initialization, SAT, and adaptation stages was conducted with the tuned hyper-parameters that produced the highest frame-level recognition accuracies over the validation data set.

5. Results and Discussions

5.1 Supervised Adaptation Procedure

Table 1 shows the results in the supervised adaptation procedure. It shows the recognition performances of the word error rate of four evaluated recognizers: the SI recognizer, the SA-SI recognizer, the SAT recognizer, and the SA-SAT recognizer. Each error rate for the SA-SI and SA-SAT recognizers is the average value obtained by the previously described CV paradigm. In the table, l_{SD} is the index of the layer to which the SD module was allocated. Because the baseline SI recognizer did not have an SD module layer, the same error rate value, 26.4%, is shown in all the corresponding columns.

The SA-SI recognizer results show that the conven-

Table 1 Experimental results (word error rate [%]) in supervised adaptation procedure.

l_{SD}	SI	SA-SI	SAT	SA-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

tional way of adapting the SD module in the SI recognizer produced clear improvements. Its error reductions from the rates of the baseline SI recognizer ranged from 6.4 to 7.7 points. However, comparing the SA-SI and SA-SAT recognizers clearly demonstrates the effect of our SAT-based scheme for DNN-HMM recognizers. Regardless of the allocation of the SD module layer, the SA-SAT recognizer outperformed the SA-SI recognizer. Moreover, the SA-SAT recognizer successfully reduced the lowest error rate of the SA-SI recognizer, 18.7%, to 18.0%, which was the best among all of the obtained error rates.

To prove the effectiveness of our SAT-based scheme, we conducted a matched pairs *t*-test for the difference in word error rates between the SA-SAT recognizer and its counterpart SA-SI recognizer. Here, each word error rate was observed for one of the 28 testing speakers through the CV paradigm. From the test results, we found that the error rate reductions between the SA-SAT recognizer and the SA-SI recognizer were significant with $p < 0.01$ when l_{SD} was set to 1, 2, 3, or 4, and with $p < 0.05$ when l_{SD} was set to 5 [t-test].

The results of the SAT recognizer were not promising. However, since the SAT scheme aims to increase the recognition accuracy after the adaptation but not to construct a high performance recognizer without the adaptation, these high error rates are not really a problem.

As shown in Table 1, the effects of the recognizer training/adaptation methods are often evaluated using the average accuracies over multiple testing speakers; such evaluation is obviously important. However, at the same time, it is desirable that the methods accurately work for all of the testing speakers or as many testing speakers as possible. Such reliability (or stability) of the methods is also clearly important. From this viewpoint, we compared the accuracy of the SA-SAT and SA-SI recognizers in the speaker-by-speaker manner and found that our proposed SA-SAT recognizer outperformed the SA-SI recognizer for 75% to 93% of the 28 testing speakers[†].

The table also shows another quite interesting finding. The adaptation allocating the SD module to such inner layers as the 2nd or 3rd layer outperformed the allocation of the SD module to the outer layers near the input or output of the network, such as the 1st and 5th layers. This phenomenon was commonly observed in both the SA-SI and SA-SAT recognizers. The DNN part repeated the feature transformation along with the data feed-forwarding from the input layer to the output layer. Allocating the SD modules to the inner layers allowed a complex feature transformation in both the lower and upper layers. Such a well balanced transformation is probably useful for extracting salient information for recognition/adaptation, although its mechanism remains hidden. Accordingly, we consider the use of DNN more suitable for speaker adaptation (probably also for other types such as speaking environment and transmission channel adaptations) than conventional shallow neural networks or any simple front-end architecture that has no deep layer structure.

[†]The percentage changed according to the selection of the SD module layer.

Table 2 Experimental results (word error rate [%]) in unsupervised adaptation procedure.

l_{SD}	SI	SA-SI	SAT	SA-SAT
1	26.4	21.4	27.2	20.4
2	26.4	20.6	26.9	20.0
3	26.4	20.7	27.0	20.1
4	26.4	21.0	26.6	20.3
5	26.4	21.5	26.5	21.0

5.2 Unsupervised Adaptation Procedure

Table 2 shows the results in the unsupervised adaptation procedure. As in Table 1, Table 2 shows the recognition performances in the word error rate of the four evaluated recognizers. Each error rate for the SA-SI and SA-SAT recognizers is also the average value obtained through the CV paradigm. To select the adaptation data used for the unsupervised adaptation, we set the threshold value for the confidence measure to 0.5 based on the preliminary experiment results. In the preliminary experiments, we tested several different values for the confidence measure and found that the confidence measure of 0.5 achieved a reduction in word error rate of 0.2 (from the rate obtained without the confidence measure) on average for both the SA-SAT recognizer and SA-SI recognizer. The effects of the confidence measure slightly varied according to the selections of the SD module layers and the recognizers.

In the unsupervised adaptation results, we identified a trend similar to the supervised adaptation results. The SA-SAT recognizer outperformed the SA-SI recognizer, regardless of the SD module layer allocation. Moreover, the SA-SAT recognizer achieved the lowest error rate, 20.0%, which was 0.6 point lower than that of the SA-SI recognizer. Compared to the supervised adaptation, the accuracy improvement in the unsupervised adaptation was not large: The reference transcription was not used in the adaptation training. However, comparing the SA-SI and SA-SAT recognizers also clearly demonstrates the effect of the SAT-based DNN-HMM recognizer, even in the unsupervised adaptation procedure.

In the unsupervised adaptation case, we again conducted the matched pairs *t*-test for the differences in the word error rates between the SA-SAT recognizer and the SA-SI recognizer. Here, each word error rate was obtained for one of the 28 testing speakers through the CV paradigm. The test results proved that the error rate reductions between the SA-SAT recognizer and the SA-SI recognizer were significant with $p < 0.01$ when l_{SD} was set to 1 or 4, and with $p < 0.05$ when l_{SD} was set to 2, 3, or 5.

In addition, the adaptation allocating the SD module to the inner layers also outperformed the case of allocating the SD module to the outer layers even in the unsupervised procedure.

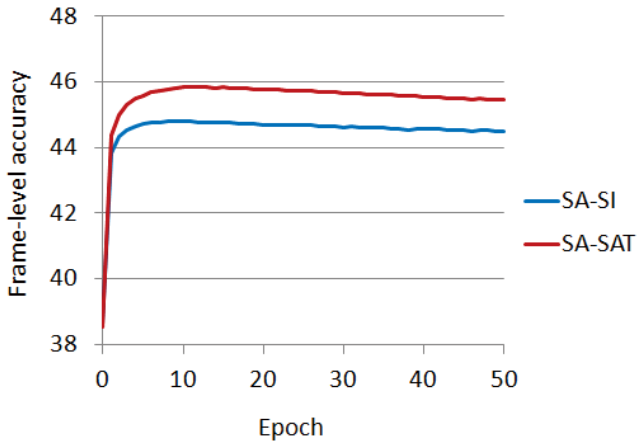


Fig. 5 Frame-level senone recognition accuracy [%] as a function of supervised adaptation epoch.

5.3 Effects of SAT-Based Adaptation: Stability Analysis of Adaptation

To deepen our understanding of our SAT-based adaptation, we elaborate the SA-SI and SA-SAT recognizers produced in the supervised adaptation procedure.

In the experiments of Sect. 5.1, we set the number of adaptation epoch iterations to ten and gained higher performances with our SA-SAT recognizer than with SA-SI recognizer. This setting was done from the viewpoint that fast adaptation was more preferable. However, there is the possibility that the SA-SAT recognizer's advantage was gained by selecting a proper length of the epoch iteration by chance. Actually, as a side effect, a short iteration occasionally increases recognition accuracies. To scrutinize this point, we ran the adaptation by setting the iteration number to 50. Here, except for the number of adaptation epoch iterations, we used the same hyper-parameter settings as in Sect. 5.1.

Figure 5 illustrates the frame-level senone recognition accuracies (in the vertical axis), each of which is a function of the epoch (in the horizontal axis), of the SA-SI and SA-SAT recognizers. This senone recognition accuracy has a close relation with the criterion used in the adaptation stage. Each accuracy curve in the figure was obtained, for its corresponding recognizer, by averaging the accuracies over all the experiment runs conducted by changing the SD module layers in the CV paradigm.

The figure shows that the SA-SAT recognizer stably outperformed the SA-SI recognizer in all the adaptation epochs. The advantage of our SA-SAT recognizer is probably generated by the nature of the SAT-based adaptation mechanism.

5.4 Effects of SAT-Based Adaptation: Comparison with All Layer Adaptation

In the previous subsections, we demonstrated the superiority of our proposed SAT-based adaptation scheme to the SI-based adaptation scheme. In the SI-based adaptation, we adapted only the SD module similarly to the SAT-based adaptation. One may question whether the SAT-based (module-based) adaptation is more effective than the simple (non-module-based) adaptation of the whole DNN of the SI recognizer. To analyze this point, we compared the following three recognizers in the supervised adaptation procedure: (1) the SA-SAT recognizer allocating the SD module in the third layer (*SA-SAT-L3 recognizer*), (2) the SA-SI recognizer allocating the SD module in the third layer (*SA-SI-L3 recognizer*), and (3) a new Speaker-Adapted SI recognizer in which all of the trainable DNN parameters, i.e., the connection weights and biases in all layers, were used for adaptation (*SA-SI-ALL recognizer*). The SA-SAT-L3 recognizer and the SA-SI-L3 recognizer were the best (in terms of SD module layer allocation) SA-SAT and SA-SI recognizers, respectively (Table 1). Furthermore, the SA-SI-ALL recognizer was constructed, adopting the same training/adaptation procedures (e.g., the use of L^2 prior regularization and the CV paradigm) as those for the recognizers in Table 1.

Table 3 Comparisons among module-based adaptation (SA-SI-L3 and SA-SAT-L3 recognizers) and non-module-based adaptation (SA-SI-ALL recognizer).

Recognizer	# adaptation param.	Word error rate (%)
SA-SI-L3	0.26 M	18.7
SA-SI-ALL	2.8 M	18.3
SA-SAT-L3	0.26 M	18.0

Table 3 shows the number of adaptation parameters and the word error rates for the above three recognizers. In the table, "M" represents million. Using a larger number of adaptation parameters, the SA-SI-ALL recognizer gained a word error rate reduction of 0.4 point from that of the SA-SI-L3. However, the rate by the SA-SI-ALL recognizer did not achieve the lowest word error rate, which was reached by the SA-SAT-L3 recognizer at 18.0%. To analyze the effect of the SAT-based (module-based) adaptation against the non-module-based adaptation, we conducted the matched pairs t-test for the difference in word error rates between the SA-SI-ALL recognizer and the SA-SAT-L3 recognizer. The test results proved that improvement of the SA-SAT-L3 recognizer over the SA-SI-ALL recognizer was significant with $p < 0.05$.

Although the t-test proved the statistical difference in word error rates between the SA-SAT-L3 and SA-SI-ALL recognizers, the difference was not so large. A point to note here is that the SA-SAT-L3 recognizer used only 9 % (0.26 M) of the adaptation parameters of the SA-SI-ALL recognizer (2.8 M). This leads to a dramatic reduction in the size of adaptation parameters, which must be stored and adapted for each target speaker. For example, let us assume that a speech recognition system runs on some server and tries to increase its discriminative power through speaker adaptation for a huge number of system users (speakers). The system is expected to handle many different speakers' data simultaneously and thus must store on the server many adaptation

parameter sets, each for a different speaker. Obviously, a small size of speaker-dependent adaptation parameters is favorable in this common scenario. In addition, it is expected that the use of such a small number of adaptation parameters decreases the risk of the over-training problem, especially in the cases where speech data available for adaptation training are severely limited.

6. Conclusion

We proposed a new speaker adaptation scheme that applied the SAT concept to the training of a DNN front-end that was incorporated into a hybrid DNN-HMM speech recognizer. We evaluated our proposed scheme with the TED Talks corpus in supervised and unsupervised adaptation procedures, and our experimental results clearly demonstrated its high utility in both procedures. In addition, we revealed that the SD module allocated into the inner layers worked better than that allocated into the outer layers near the input and output layers. This result suggests that allocating the modules in the inner layers effectively generates salient information for recognition, probably based on the large feature transformation capability gained by the well balanced use of upper and lower layers.

In this paper, we adopted the L^2 norm-based regularization term that restricted the parameters to be trained so that they could stay close to their (initial) anchorage states, e.g., $\overline{\boldsymbol{\lambda}}^{\text{SI-DNN}}_{l_{\text{SD}}}$ in the SAT stage and $\boldsymbol{g}^{anchor}_{l_{\text{SD}}}$ in the speaker adaptation stage. Since anchorage states can be considered a kind of prior information for such successive operations as search, using the above anchorage states is probably one reasonable choice. However, it has no rationale and can be improved by elaborating other types of anchorage states or regularization terms. Our future study will include such deep investigation. The effects of using larger networks and shorter speech data for adaptation will also be important future issues.

Acknowledgments

This work was supported in part by JSPS Grants-in-Aid for Scientific Research No. 26280063, MEXT-Supported Program "Driver-in-the-Loop", and Grant-in-Aid for JSPS Fellows. The authors appreciate their financial support.

References

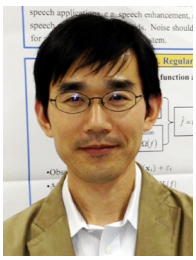
[1] C.J. Leggetter and P.C. Woodland, "Maximum likelihood linear regression for speaker adaptation of continuous density hidden Markov models," Computer Speech and Language, vol.9, no.2, pp.171–185, 1995.
[2] M.J.F. Gales, "Maximum likelihood linear transformations for HMM-based speech recognition," Computer Speech and Language, vol.12, no.2, pp.75–98, 1998.
[3] J.-L. Gauvain and C.-H. Lee, "Maximum a posteriori estimation for multivariate Gaussian mixture observations of Markov chains," IEEE Trans. Speech and Audio Processing, vol.2, no.2, pp.291–298, 1994.
[4] E. Eide and H. Gish, "A parametric approach to vocal tract length normalization," Proc. Int. Conf. Acoust., Speech, Signal Processing, vol.1, pp.346–348, 1996.
[5] T. Anastasakos, J. McDonough, R. Schwartz, and J. Makhoul, "A compact model for speaker-adaptive training," Proc. Int. Conf. Spoken Language Processing, vol.2, pp.1137–1140, 1996.
[6] N. Iwahashi, "Training method for pattern classifier based on the performance after adaptation," IEICE Trans. Inf. & Syst., vol.E83-D, no.7, pp.1560–1566, July 2000.
[7] G. Hinton, L. Deng, D. Yu, G. Dahl, A.-R 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," IEEE Signal Process. Mag., vol.29, no.6, pp.82–97, 2012.
[8] G.E. Dahl, D. Yu, L. Deng, and A. Acero, "Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition," IEEE Trans. Audio, Speech, and Language Processing, vol.20, no.1, pp.30–42, 2012.
[9] A.-R. Mohamed, G.E. Dahl, and G. Hinton, "Acoustic modeling using deep belief networks," IEEE Trans. Audio, Speech, and Language Processing, vol.20, no.1, pp.14–22, 2012.
[10] G.E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," Neural Computation, vol.18, no.7, pp.1527–1554, 2006.
[11] 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," Proc. EUROSPEECH, pp.2171–2174, 1995.
[12] 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, pp.827–835, 2007.
[13] B. Li and K.C. Sim, "Comparison of discriminative input and output transformations for speaker adaptation in the hybrid NN/HMM systems," Proc. Interspeech, pp.526–529, 2010.
[14] F. Seide, G. Li, X. Chen, and D. Yu, "Feature engineering in context-dependent deep neural networks for conversational speech transcription," Proc. IEEE WS Auto. Speech Recognition and Understanding, pp.24–29, 2011.
[15] 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," Proc. Int. Conf. Acoust., Speech, Signal Processing, pp.6359–6363, 2014.
[16] 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," Proc. Int. Conf. Acoust., Speech, Signal Processing, pp.6339–6343, 2014.
[17] O. Abdel-Hamid, and H. Jiang, "Fast speaker adaptation of hybrid NN/HMM model for speech recognition based on discriminative learning of speaker code," Proc. Int. Conf. Acoust., Speech, Signal Processing, pp.7942–7946, 2013.
[18] G. Saon, H. Soltau, D. Nahamoo, and M. Picheny, "Speaker adaptation of neural network acoustic models using i-vectors," Proc. IEEE WS Auto. Speech Recognition and Understanding, pp.55–59, 2013.
[19] V. Gupta, P. Kenny, P. Ouellet, and T. Stafylakis, "I-vector-based speaker adaptation of deep neural networks for French broadcast audio transcription," Proc. Int. Conf. Acoust., Speech, Signal Processing, pp.6384–6388, 2014.
[20] N. Dehak, P.J. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," IEEE Trans. Audio, Speech and Language Processing, vol.19, no.4, pp.788–798, 2010.
[21] S.M. Siniscalchi, J. Li, and C.-H. Lee, "Hermitian based hidden activation functions for adaptation of hybrid HMM/ANN models," Proc. Interspeech, pp.2590–2593, 2012.
[22] P. Swietojanski and S. Renals, "Learning hidden unit contributions for unsupervised speaker adaptation of neural network acoustic models," Proc. IEEE WS Spoken Language Technology, pp.171–176, 2014.

[23] H. Liao, "Speaker adaptation of context dependent deep neural networks," Proc. Int. Conf. Acoust., Speech, Signal Processing, pp.7947–7951, 2013.
[24] D. Yu, K. Yao, H. Su, G. Li, and F. Seide, "KL-divergence regularized deep neural network adaptation for improved large vocabulary speech recognition," Proc. Int. Conf. Acoust., Speech, Signal Processing, pp.7893–7897, 2013.
[25] S.P. Rath, D. Povey, K. Vesely, and J.H. Cernocky, "Improved feature processing for deep neural network," Proc. Interspeech, pp.109–113, 2013.
[26] T. Yoshioka, A. Ragni, and M.J.F. Gales, "Investigation of unsupervised adaptation of DNN acoustic models with filter bank input," Proc. Int. Conf. Acoust., Speech, Signal Processing, pp.6394–6398, 2014.
[27] Y. Miao, H. Zhang, and F. Metze, "Towards speaker adaptive training of deep neural network acoustic models," Proc. Interspeech, pp.2189–2193, 2014.
[28] J. Trmal, J. Zelinka, and L. Muller, "On speaker adaptive training of artificial neural networks," Proc. Interspeech, pp.554–557, 2010.
[29] S.Y. Kung, Digital Neural Networks, Prentice-Hall, Englewood Cliffs, 1993.
[30] T. Ochiai, S. Matsuda, X. Lu, C. Hori, and S. Katagiri, "Speaker adaptive training using deep neural networks," Proc. Int. Conf. Acoust., Speech, Signal Processing, pp.6399–6403, 2014.
[31] D. Povey, D. Kanevsky, B. Kingsbury, B. Ramabhadran, G. Saon, and K. Visweswariah, "Boosted MMI for model and feature space discriminative training," Proc. Int. Conf. Acoust., Speech, Signal Processing, pp.4057–4060, 2008.
[32] G.E. Hinton, "Training products of experts by minimizing contrastive divergence," Neural Computation, vol.14, no.8, pp.1771–1800, 2002.
[33] M. Cettolo, J. Niehues, S. Stuker, L. Bentivogli, and M. Federico, "Report on the 10th IWSLT evaluation campaign," Proc. Int. WS Spoken Language Translation, 2013.
[34] 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," Proc. Int. WS. Spoken Language Translation, pp.34–37, 2012.
[35] L. Mangu, E. Brill, and A. Stolcke, "Finding consensus in speech recognition: word error minimization and other applications of confusion networks," Computer Speech and Language, vol.14, no.4, pp.373–400, 2000.

Tsubasa Ochiai received B.E. and M.E. degrees in Information Engineering from Doshisha University, Kyotanabe, Japan, in 2013 and 2015, respectively. He is currently a Ph.D. student at Doshisha University and a Research Fellow of JSPS. His research interests include pattern recognition, deep learning, and speech recognition. He is a member of the Acoustic Society of Japan, IEICE, and IEEE.

Shigeki Matsuda received his Ph.D. degree from Japan Advanced Institute of Science and Technology (JAIST) in 2003, and joined ATR Spoken Language Communication Laboratories as a researcher. From 2009, he joined Spoken Language Communication Laboratory of the National Institute of Information and Communications Technology (NICT) as a researcher. From 2014, he works for ATR-Trek. He is a manager of speech processing development department. He is engaged in research on speech recognition and speech signal processing, and is a member of the Acoustic Society of Japan, Information Processing Society of Japan, and IEICE.

Hideyuki Watanabe received his Ph.D. degree from Hokkaido university in 1993. From 1993 to 2009, he worked for Advanced Telecommunications Research Institute International (ATR). From 2009, he works for National Institute of Information and Communications Technology (NICT). His current research interests include studies on pattern recognition theory, discriminative training, and speech signal processing. He is a member of the Acoustic Society of Japan, IEICE, and IEEE.



Xugang Lu received his Ph.D. from the National Lab of Pattern Recognition, Chinese Academy of Science, in 1999. During years 1999-2003, he worked as a postdoc fellow in Nanyang Technological University, Singapore, and McMaster University, Canada. From 2003 to 2008, he was an assistant professor at Japan Advanced Institute of Science and Technology. In 2008, he joined in Advanced Telecommunications Research Institute (ATR) as a senior researcher. From 2009 to now, he is a senior researcher at National Institute of Information and Communications Technology (NICT), Japan. His main research interests include speech signal processing and recognition, and statistic pattern recognition and machine learning. He is a member of the Acoustic Society of Japan.

Chiori Hori received her Ph.D. degree from Tokyo Institute of Technology in 2002. She has engaged in research on spoken language processing. She joined the NTT Communication Science Laboratories in Kyoto from 2002 and worked at Carnegie Mellon University in Pittsburgh from 2004. She joined ATR/NICT in 2007 and researched at NICT in Kyoto for 8 years. She was the research manager of Spoken Language Communication Laboratory of NICT from 2012. She is a member of Mitsubishi Electric Research Laboratories in Boston from 2015. She has worked on speech summarization/translation and spoken dialog technologies and standardization on communication protocols for speech interfaces at ITU-T and ASTAP. She is a member of IEEE, the IEICE and ASJ.

Hisashi Kawai received B.E., M.E., and D.E. degrees in electronic engineering from The University of Tokyo, in 1984, 1986, 1989, respectively. He joined Kokusai Denshin Denwa Co. Ltd. in 1989. He worked for ATR Spoken Language Translation Research Laboratories from 2000 to 2004, where he engaged in the development of text-to-speech synthesis system. From Oct. 2004 to Mar. 2009 and from April 2012 to Sept. 2014 he worked for KDDI R&D Laboratories, where he was engaged in the research and development of speech information processing, speech quality control for telephone, speech signal processing, acoustic signal processing, and communication robots. From April 2009 to March 2012 and since Oct. 2014, he has been working for National Institute of Information and Communications Technology (NICT), where he is engaged in development of speech technology for spoken language translation. He is a member of Acoustical Society of Japan (ASJ) and IEEE.

Shigeru Katagiri received B.E. and M.E. degrees in Electrical Engineering and a Dr. Eng. degree in Information Engineering from Tohoku University in 1977, 1979, and 1982, respectively. In 1982, he joined Nippon Telegraph and Telephone Public Corporation (currently NTT). He moved to Advanced Telecommunications Research Institute International (ATR) in 1986 and returned to NTT in 1999. Since 2006, he has been with Doshisha University, where he is currently a Professor in the Graduate School of Science and Engineering. He was awarded the 22nd Sato Paper Award of the Acoustical Society of Japan (ASJ) in 1982, the 27th Sato Paper Award of the ASJ in 1987, and the IEEE Signal Processing Society 1993 Senior Award in 1994. Dr. Katagiri is an IEEE Fellow and an NTT R&D Fellow.