Received: 11 August 2017 | Revised: 3 May 2018 | Accepted: 25 June 2018

DOI: 10.4218/etrij.2017-0087

ORIGINAL ARTICLE

WILEY ETRI Journal

Fast speaker adaptation using extended diagonal linear transformation for deep neural networks

Donghyun Kim | **Sanghun Kim**

SW·Contents Research Laboratory, Electronics and Telecommunications Research Institute, Daejeon, Rep. of Korea.

Correspondence
Donghyun Kim, SW·Contents Research Laboratory, Electronics and Telecommunications Research Institute, Daejeon, Rep. of Korea.
Email: dawnkann@etri.re.kr

This paper explores new techniques that are based on a hidden-layer linear transformation for fast speaker adaptation used in deep neural networks (DNNs). Conventional methods using affine transformations are ineffective because they require a relatively large number of parameters to perform. Meanwhile, methods that employ singular-value decomposition (SVD) are utilized because they are effective at reducing adaptive parameters. However, a matrix decomposition is computationally expensive when using online services. We propose the use of an extended diagonal linear transformation method to minimize adaptation parameters without SVD to increase the performance level for tasks that require smaller degrees of adaptation. In Korean large vocabulary continuous speech recognition (LVCSR) tasks, the proposed method shows significant improvements with error-reduction rates of 8.4% and 17.1% in five and 50 conversational sentence adaptations, respectively. Compared with the adaptation methods using SVD, there is an increased recognition performance with fewer parameters.

KEYWORDS
DNN acoustic modeling, DNN adaptation, DNN speech recognition, fast speaker adaptation, SVD adaptation

1 | INTRODUCTION

Over the past few years, deep neural networks (DNNs) [1,2] which are used to learn acoustic models (AMs), have significantly improved the performance of speech recognition systems. In particular, for large-vocabulary continuous speech recognition (LVCSR) tasks, a DNN requires tens of millions of parameters to increase the accuracy using thousands of hours of training data. Owing to deep multi-layer and fully connected weights used with a non-linear activation function, a DNN systematically and discriminatively performs and overcomes the barrier of a Gaussian mixture model (GMM)-based performance. To prevent a degradation in accuracy owing to differences between speakers, there is a need for speaker adaptation using limited data relative to large parameters.

Fast speaker-adaptation methods can be broadly divided into feature-based and model-based adaptations. Feature-based adaptation methods extract features by minimizing an individual speaker's characteristics using feature transformation methods [3,4], whereas a model-adaptation is a method for transforming a speaker-independent (SI) model into a more speaker-dependent (SD) version. Model-adaptation methods include regularization methods [5,6] that decrease the overfitting by preventing many deviations from the SI model; learning speaker information such as a speaker code [7], bases [8], and i-vectors [9,10] that can effectively combine with general features and other adaptation methods; and transformation-based techniques that convert the DNN layer into a speaker adapted layer. In particular, whereas feature adaptation affects all senones, by applying relatively small transformation parameters to the

input layer, a model adaptation can be applied to all other layers. In addition, because the hidden-layer operation of a DNN can be a process of feature extraction and regression in abstract feature representations, model-adaptation methods using various layers may be more effective than feature adaptation applied solely to the input layer.

Previously, linear-transformation methods [11–13] have introduced an additional linear layer to a particular layer, such as the input and output layers, or to the top hidden layer. In [14] and [15], an affine transformation has also been proposed to adapt the hidden layer by converting an additional scale matrix and bias vector. In particular, [15] showed how to combine Kullback-Leibler divergence (KLD) regularization [16,17]. However, to prevent overfitting, only a diagonal matrix was adapted in the evaluation. In addition, in [18], amplitude adaptation of the activation function in a hidden layer was proposed using fewer parameters under limited conditions.

The fundamental problem of model-adaptation is that there are many parameters to be converted relative to the amount of data. Approaches that use singular-value decomposition (SVD) are able to reduce the transformation parameters using low-rank matrices [19]. In [20], the handling of an intermediate layer for an SVD-based DNN adaptation shows a better performance than using the input and output layers. Nevertheless, the combined performance of each hidden layer, which has a relatively independent nature, is not shown, and only the adaptation of individual layers is investigated. In [21], the authors present a low-rank plus diagonal (LRPD) method combined with decomposed SVD matrices and a diagonal matrix, which considers the data gathered around the diagonal elements of the transformation layer. However, this SVD operation is computational costly, and slows down the online adaptation process, especially when using a large dimensionality of the matrix.

In this study, we first examine the effects of each layer in a DNN model adaptation, after which we investigate the relationship between the performance and adaptation parameters using the basic hidden-layer adaptations. We also attempt to use LRPD methods by changing the position of the additional linear layer, while finding the best rank of the SVD matrix. Then, based on the preceding methods, we propose a hidden-layer-based extended diagonal adaptation as a scale matrix and a bias vector for a fast speaker adaptation. Experiments were conducted in both supervised and unsupervised manners for LVCSR tasks.

The rest of this paper is organized as follows: Section 2 describes the hidden-layer DNN adaptation method. Then, Section 3 introduces the LRPD method, which has a change in position. Section 4 describes the extended diagonal transformation method, while Section 5 provides the experimental results. Finally, Section 6 concludes the paper.

2 | HIDDEN LAYER-BASED DNN ADAPTATION

A DNN requires a sequence of observation vectors extended with left-right window frames as the input. The output layer of the activation function is then used to estimate the posterior probabilities, which are as many as the number of senones that are constructed during the GMM process. A DNN has many layers, and each fully connected layer has the following operation:

$$\mathbf{v}_t^l = f(\mathbf{z}_t^l) = f(\mathbf{W}^l \mathbf{v}_t^{l-1} + \mathbf{b}^l), \tag{1}$$

where $\mathbf{v}_t^l$ is a current l layer output vector at sample frame t. In addition, $\mathbf{z}^l$ is a l layer input vector estimated using the previous layer output vector, $\mathbf{v}^{l-1}$, weight matrix, $\mathbf{W}^l$, and bias vector, $\mathbf{b}^l$. The activation function, $f(\cdot)$, can be either a sigmoid, hyperbolic tangent, or rectified linear unit function. The model parameters such as the weight and bias are estimated using an error back-propagation (BP) algorithm as an iterative fine-tuning operation through the cross-entropy, $J(\cdot)$, criterion [22,23]:

$$J(\mathbf{W}, \mathbf{b}; \mathbf{o}, \mathbf{y}) = \frac{1}{T} \sum_t^T \sum_{c=1}^c \mathbf{y}_t^c \log p(c|\mathbf{o}_t; \mathbf{W}, \mathbf{b}), \tag{2}$$

where $\mathbf{y}_t^c$ is the empirical target probability of the senone (context-dependent phone state), c, aligned to observation, $\mathbf{o}_t$, at frame t, while $p(c \mid \mathbf{o}_t; \mathbf{W}, \mathbf{b})$ is the same hypothesis output probability estimated from the DNN.

Such a BP algorithm is used for fundamental adaptation as well as training. Speaker adaptation differs from AM training because it needs to update large model parameters using a relatively small amount of data. In particular, the output layer that conducts the classification has many parameters that depend on the number of senone nodes. When using a limited amount of adaptation data, the large parameters of the output layer can cause biased changes and an overfitting. Hence, in [20], parameter updates in the hidden layers, where there are relatively few parameters for each layer compared to the output layer, have been studied.

The basic form of a DNN with one hidden layer is shown in Figure 1A. It consists of a fully connected linear layer and a nonlinear layer as an activation function. During adaptation, only the linear parameters of the hidden layers are updated, while the other layers are fixed. However, because this method changes the parameters of the SI model, it must be adapted for each speaker using a copy of the SI model. To reduce the storage space, only the adapted layer is stored for each speaker, and the adapted

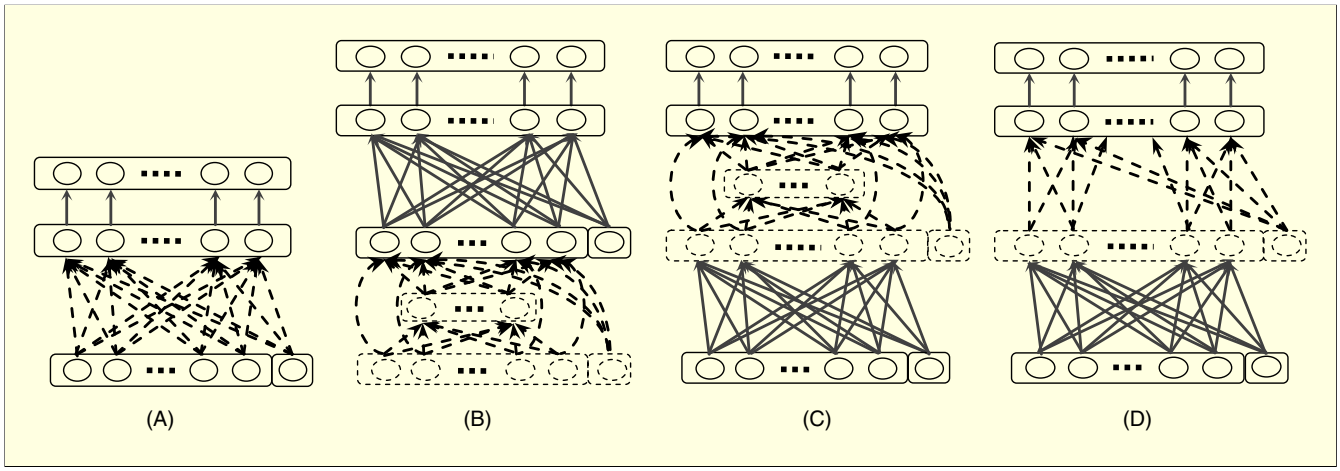

FIGURE 1 Illustration of a DNN structure with one hidden layer using different methods: (A) layer full transfer, (B) down–LRPD, (C) up–LRPD, and (D) linear extended diagonal

W and **b** of the layer parameters that transfer to the SI model are used as the SD parameters. Similar to the transfer learning in [24,25], this is referred to as hidden layer transfer (HLT) adaptation. Although this method is simple, it has sufficient functions for speaker adaptation.

3 | DNN ADAPTATION WITH LOW-RANK

The HLT method shown in Figure 1A may actually update certain parameters more strongly than others may, but storage problems arise because these parameters are not distinguishable. Therefore, transformation-based approaches have also applied an additional linear layer to generate speaker-specific parameters without changing the SI model. In traditional methods [14,15], an additional layer is introduced in a linear layer portion to convert an input vector, $\mathbf{v}^{l-1}$, and (1) is replaced as follows:

$$\mathbf{v}^l = f(\mathbf{W}^l(\boldsymbol{\alpha}_s^{l-1}\mathbf{v}^{l-1} + \boldsymbol{\beta}_s^{l-1}) + \mathbf{b}^l), \tag{3}$$

where $\boldsymbol{\alpha}_s^{l-1}$ is a transformation matrix and $\boldsymbol{\beta}_s^{l-1}$ is a bias vector for speaker s with $l-1$ layer dimension. These speaker-specific parameters can be effectively reduced using an SVD. Recently, [21] reported an LRPD adaptation, which is a method for applying low-rank matrices and a diagonal matrix together, as shown in Figure 1B. The $\boldsymbol{\alpha}_s$ matrix is decomposed as follows:

$$\boldsymbol{\alpha}_{s,n\times n} \approx \mathbf{D}_{s,n\times n} + \mathbf{P}_{s,n\times k}\mathbf{Q}_{s,k\times n}, \tag{4}$$

where $\mathbf{D}_{s,n\times n}$ is an n by n dimensionality of a diagonal matrix for a speaker transformation. In addition, $\mathbf{P}_{s,n\times k}$ and $\mathbf{Q}_{s,k\times n}$ are low-rank matrices decomposed using an SVD from $\boldsymbol{\alpha}_s$ without $\mathbf{D}_s$ and are truncated into k elements with $k \ll n$. In general, the additional matrix for adaptation shows a diagonal dominant distribution, and thus the use of an SVD with only an off-diagonal matrix can make a low-rank matrix better for reducing the SD parameters.

Alternatively, we tried to apply an additional layer on top of a linear layer to convert all linear parameters using the layer dimensions by performing the following equation:

$$\mathbf{v}^l = f(\boldsymbol{\alpha}_s^l(\mathbf{W}^l\mathbf{v}^l + \mathbf{b}^l) + \boldsymbol{\beta}_s^l). \tag{5}$$

The main difference with the conventional method is that this method can set the transformation matrix, $\boldsymbol{\alpha}_s^l$ to affect the entire linear layer, including the weight and bias, and can even add bias factor $\boldsymbol{\beta}_s^l$, as shown in Figure 1C, whereas the LRPD only affects the weight matrix, $\mathbf{W}^l$. If the previous LRPD method is termed a down-LRPD, this method can be referred to as an up-LRPD.

4 | DNN ADAPTATION USING EXTENDED DIAGONAL LINEAR TRANSFORMATION

As mentioned previously, SVD-based adaptation methods have helped to reduce storage requirements, but are computationally costly when applied to online services. They may even require a graphic processing unit (GPU) computation rather than a central processing unit (CPU) to speed up the operation. Thus, we examined the linear transformation in [14] and [15], and proposed an extended diagonal linear transformation (EDLT) method with an upward linear layer added to the hidden layer, as shown in Figure 1D. This proposed method begins with the idea that the two closely linked nodes, locally between the lower and upper layers, have a relatively more effective connection than the two far-linked nodes. One reason for this is the time difference as the input data expands to

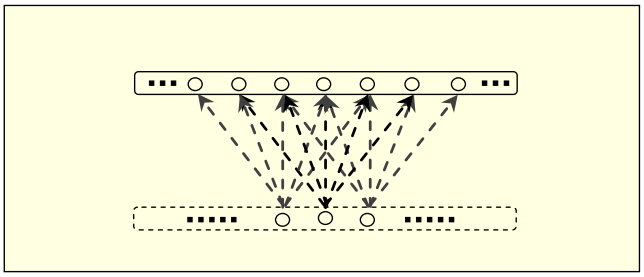

FIGURE 2 Illustration of a limited connection with left-right window-size link

include the left-right frames. Another reason is that linear transformations using square matrices tend to dominate diagonally, as described in [21].

Therefore, the proposed method uses limited connections of a left-right window size instead of having fully connected weights in the additional linear layer, as shown in Figure 2. This is equivalent to using an extended diagonal matrix, which deviates from traditional methods that use only the diagonal elements. The value of $J(\cdot)$ included in (5) is differentiated using the extended diagonal matrix, $ext\_diag(\boldsymbol{\alpha}_k^{L-1})$, below the last L layer as follows:

$$\frac{\partial J(\boldsymbol{\alpha}_s, \boldsymbol{\beta}_s)}{\partial ext\_diag(\boldsymbol{\alpha}_s^{L-1})} = ext\_diag\left(\frac{\partial J(\boldsymbol{\alpha}_s, \boldsymbol{\beta}_s)}{\partial \boldsymbol{\alpha}_s^{L-1}}\right) \tag{6}$$

$$= ext\_diag\left(\frac{\mathbf{y}}{\mathbf{v}^L} \cdot \frac{\partial f(\mathbf{z}^L)}{\partial \mathbf{z}^L} \cdot \mathbf{W}^L \cdot \frac{\partial f(\mathbf{z}^{L-1})}{\partial \mathbf{z}^{L-1}} \cdot \left(\mathbf{W}^{L-1}\mathbf{v}^{L-2} + \mathbf{b}^{L-1}\right)^T\right). \tag{7}$$

Similar to the formula in [14] using Equation (3), this equation finds the gradient, and then uses it to update $\boldsymbol{\alpha}_s$. The value of $\boldsymbol{\beta}_s$ is also obtained in this manner. This method can be applied to both a full-sized DNN and a low-rank DNN [19,21], although we evaluated it only for a full-sized DNN.

5 | EXPERIMENT RESULTS

In our research, we focused on data and adaptation methods to improve the performance of personal speech recognition. In particular, we aimed to improve spontaneous speech recognition using a small amount of data on mobile devices such as smartphones. Therefore, training and evaluation were conducted using conversational data recorded in mobile conditions.

The experiments were performed using a Korean LVCSR system, which includes a trigram language model with a 400K vocabulary for automatic speech transcription, and uses a weighted finite-state transducer as a language network. By collecting over several billion sentences, we have created the LM corpus primarily by combining the daily life with the tourism domain dialogue. To train the AM of a DNN hidden-Markov-model (DNN-HMM), the modeling of the GMM-HMM is processed as the first step. This step generates 7,981 senone states and 256K Gaussians. In the second step, the same training data are force-aligned to obtain senone state labels for DNN training. The DNN input features, which are log filter-banks with 40 coefficients distributed on a mel-scale, have a context window of 15 frames to incorporate acoustic context information of a speech signal. After splicing, the features of the input layer are transformed using linear discriminant analysis (LDA) to reduce the correlated variables of the log filter-bank. On top of this input layer, there are six hidden layers with 2,048 nodes each. The output layer has as many nodes as the number of HMM senone states.

We trained a DNN acoustic model, which includes various Korean datasets for the transcription of conversational Korean speech, using more than 1,500 hour of speech data. Most speech data are spontaneous speech recorded using 16-KHz sampling various smartphones in clean and noisy added environments using various additive noises, and the rest is read speech recorded using the same sampling general condense-microphone. We utilized the Kaldi open-source based toolkit [26] to model the SI DNN in these experiments. Under the same smartphone speech condition as in the office environment, the evaluation data are a Korean travel-guide dataset with 4,000 utterances, which comprises spontaneous conversations about the tourism domain. This data includes an average length adaptation of 2.1 seconds, and a test data length with 3.6 seconds per utterance. With more than 30K words, we used 40 speakers for adaptation and testing with 50 sentences per speaker, and there was no overlap between the training, adaptation, and testing sets under a supervised manner.

5.1 | Hidden-layer transfer adaptation

We first observed the recognition results using an HLT adaptation of 50 utterances for various hidden layers. The base word accuracy rate (WAC) is 94.68%, as shown in Table 1. The word error-reduction rate (WERR) of a one-layer adaptation is relatively good in the middle layer, and using several hidden layers together, such as HLT-345L, is better than using a few layers only. Updating all hidden layers (HLT-All) yields the best performance, and thus it seems to have a combined effect.

Although the HLT uses the basic BP algorithm, such as to retrain the SI DNN while freezing the input and output layers, it shows a maximum WERR of 17.62% with all hidden layers. Nevertheless, the drawback is that too many

TABLE 1 Recognition results and a number of footprint of parameters using the hidden-layer transfer of 50 adaptation utterances

HLT-layer	WAC (%)	WERR (%)	Footprint (#)
No adaptation	94.68	0	37.3M
HLT-3L	95.09	7.67	4.0M
HLT-4L	95.21	9.92	4.0M
HLT-5L	95.08	7.52	4.0M
HLT-6L	95.00	5.95	4.0M
HLT-56L	95.17	9.19	8.0M
HLT-345L	95.52	15.82	12.0M
HLT-456L	95.44	14.33	12.0M
HLT-All	95.62	17.62	21.8M

parameters for each speaker will need to be stored in an online service.

5.2 | SVD-based adaptation

For this experiment, we evaluated the recently proposed LRPD method [21], which uses an SVD with relatively few parameters for adaptation of all hidden layers. This is a down-LRPD (D- LRPD), whereas our proposed method is an up-LRPD (U-LRPD). To reduce the SD parameters, the experiment only conducted various rank dimensions of about 3% or less of the full-rank matrix. The adaptation performs best during three epochs of iterations, computing, and updating the SVD for each run.

We evaluated several rank dimensions of the U-LRPD adaptation for 50 utterances, as shown in Table 2. The performance gradually increases from rank 6 to rank 25, but then decreases at rank 50. The U-LRPD has a maximum WERR of 15.62% at rank 25, which is better than the D-LRPD at the same rank. In addition, at peak performance, the SD parameters are only 1.6% of the total model with 612K parameters.

TABLE 2 Recognition results and the number of footprints of parameters using SVD-based LRPD of 50 adaptation utterances

HLT-layer	WAC (%)	WERR (%)	Footprint (#)
No adaptation	94.68	0	37.3M
Up-LRPD-6	95.23	10.38	156K
Up-LRPD-12	95.39	13.37	300K
Up-LRPD-25	95.54	15.62	612K
Up-LRPD-50	95.10	7.82	1,184K
Down-LRPD-25	95.45	14.44	554K
Down-LRPD-28	95.41	13.70	620K

5.3 | Extended-diagonal linear transformation

The proposed EDLT method was devised as a solution for excessive SD parameters of the HLT and the computational cost of matrix decomposition of the LRPD. This EDLT uses all the hidden linear layers considering the advantages shown through the HLT experiments, and applies the transformation matrix added to the upper position of the hidden layer like an up-LRPD. Experiments were conducted on 50 adaptation utterances in the same supervised manner described above, and the proposed method was shown to perform best during seven epochs of iterations.

Figure 3 shows the relationship between the recognition accuracy and number of SD parameters while increasing the left-right window size of the extended diagonal matrices. Increasing the window size from 7 to 25 gradually improves the performance of the EDLT, recording a maximum WERR of 17.10% for a window size of 21. As the number of SD parameters associated with the window size increases from 200K to 600K, the performance graph gradually shows a convergence curve.

Figure 4 compares the performance of fast adaptation with the incremental adaptation data from five to 50 utterances for each representative adaptation method. The U-LRPD method has a better performance than the D-LRPD method, and as the performance increases, it can be expected to improve with even more data. Although HLT-All using all parameters of the hidden layer achieves the best performance for 50 utterances, the proposed EDLT has a better accuracy in fewer utterances, and it therefore appears to be more appropriate for a fast adaptation. EDLT-17 and EDLT-21 each use 428K and 522K fewer SD parameters than U-LRPD-25.

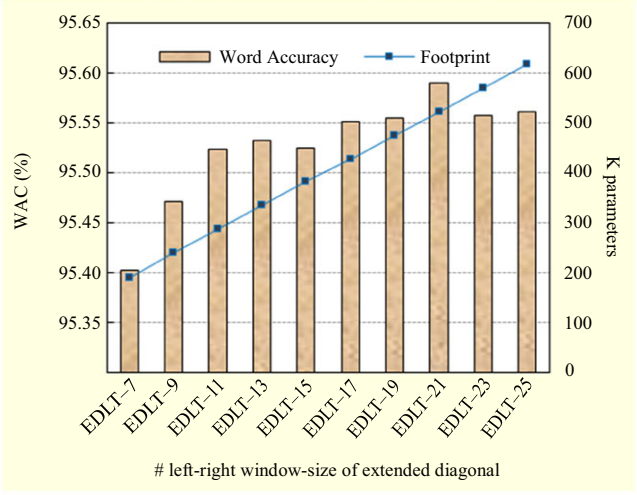

FIGURE 3 Comparison of EDLT performance and footprint variation with increasing left-right window-size

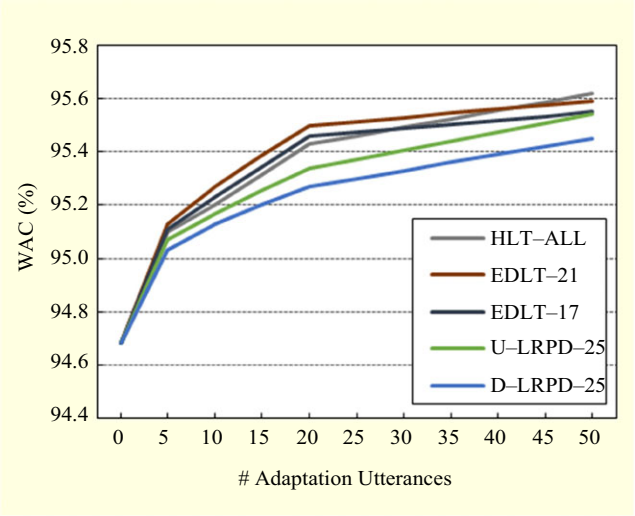

FIGURE 4 Performance comparison of various methods with increase in adaptation data

We also compared the LRPD and EDLT in an unsupervised adaptation manner using 20 and 50 utterances, respectively, for an online speaker adaptation service, as shown in Table 3. Despite the unsupervised adaptation, the performance degradation is relatively small owing to a high recognition rate. The EDLT still shows a better performance than LRPD, and is particularly effective for short adaptation data such as 20 utterances. In addition, it may be suitable for an online service when considering relatively low footprints and fast processing.

5.4 | Combined adaptation using i-vector features

In order to improve the performance of the acoustic model, i-vector based methods have been proposed which add a feature representing the characteristics of the speaker identity to the input data [27]. Adaptation methods that utilize additional features use a different approach than transformation-based adaptations, so they can be combined to reflect each adaptive effect. As shown in [9,10], i-vector extractor and an acoustic model using i-vector features as the input must be trained first to perform the adaptation. Considering the low computational cost for online processing, we used a GMM-based recipe of i-vector learning in the Kaldi toolkit [26,28].

In this experiment, we made a diagonal matrix-based universal background model (UBM) using 512 Gaussians, and trained the i-vector extractor using the UBM's posterior. Then, 100-dimensional i-vector sequences recommended by [9,27] were extracted using 650 hour of Mel-frequency cepstral coefficients (MFCCs) data which are a subset of the total training data with speaker information. The input data of the DNN is converted from 600-dimensional features containing left and right context frames for 40-dimensional log filter-banks to 700-dimensional features with the addition of an i-vector. For comparison with the previous DNN, we first generated an LDA matrix of the input layer that receives the 700-dimensional features, and then used 650 hour of the input data to make an i-vector-added DNN (i-vec DNN) that is re-trained based on the baseline DNN.

Using the i-vector features, we evaluated the utterance adaptation under the supervised manner. As shown in Table 4, the i-vec DNN had an improved performance to 4.51% WERR compared to the baseline DNN. In addition, i-vector adaptation based on the i-vec DNN for 20 and 50 utterances per speaker showed an increased accuracy of 7.87% and 14.17% WERR, respectively. Although this adaptation method has a less WERR than the proposed transformation-based method for the fast adaptation, the added i-vector features contribute to the overall improvement by increasing the accuracy of the baseline model. The combined adaptation of i-vector and the EDLT showed a larger improvement than the i-vector only adaptation and the previous EDLT by only one. Therefore, it may be deduced that the speaker identity feature and the proposed condensed transformation method affect each other in this evaluation.

TABLE 3 Recognition results of 20 and 50 adaptation utterances under unsupervised adaptation conditions

Method (#utter.)	WAC (%)	WERR (%)	Footprint (#)
No adaptation	94.68	0	37.3M
U-LRPD-25 (20)	95.31	11.84	612K
EDLT-21 (20)	95.46	14.66	522K
U-LRPD-25 (50)	95.49	15.23	612K
EDLT-21 (50)	95.55	16.35	522K

TABLE 4 Recognition results of 20 and 50 adaptation utterances using i-vector features under supervised adaptation conditions

Method (#utter.)	WAC (%)	WERR (%)
Baseline DNN	94.68	N/A
Re-trained i-vec DNN	94.92	0
i-vec. adaptation (20)	95.32	7.87
i-vec. + EDLT-21 adaptation (20)	95.70	15.35
i-vec. adaptation (50)	95.64	14.17
i-vec. + EDLT-21 adaptation (50)	95.90	19.29

6 | CONCLUSIONS

In this study, we researched various techniques for the fast adaptation of a DNN for a Korean LVCSR system. First,

the simplest HLT method was investigated to explore the characteristics of each hidden layer. In fact, this method also strongly updates only certain parameters, but has a disadvantage in terms of the storing of the parameters because it uses an intact SI structure. Therefore, we applied the LRPD method using an SVD to reduce the SD parameters. By doing this, we found that the upper position of the LRPD is better than the lower position of the conventional method. This paper proposed an EDLT method, which applies an extended diagonal matrix to the upper position of a linear hidden layer and adapts fewer parameters without using an SVD. The proposed EDLT method exhibits a fast adaptation performance when using five to 50 utterances for 40 speakers, and has a maximum 17.10% WERR with fewer parameters than the LRPD method. In addition, the EDLT performance improved when it used i-vector features. EDLT combined with i-vector adaptation increased the WERR by up to 19.29%.

ACKNOWLEDGEMENTS

This work was supported by the ICT R&D program of MSIP/IITP (R7119-16-1001, Core technology development of real-time simultaneous speech translation based on knowledge enhancement).

ORCID

Donghyun Kim http://orcid.org/0000-0002-2063-5551

REFERENCES

1. F. Seide, G. Li, and D. Yu, Conversational speech transcription using context-dependent deep neural networks, *Annu. Conf. Int. Speech Commun. Assoc.*, Florence, Italy, Aug. 27–31, 2011, 437–440.
2. G. Hinton et al., *Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups*, IEEE Signal Process **29** (2012), no. 6, 82–97.
3. M. Gales, *Maximum likelihood linear transformations for HMM-based speech recognition*, Comput. Speech Language **12** (1998), no. 2, 75–98.
4. F. Seide et al., Feature engineering in context-dependent deep neural networks for conversational speech transcription, *IEEE Workshop Autom. Speech Recogn. Understanding*, Waikoloa, HI, USA, Dec. 11–15, 2011, pp. 24–29.
5. J. Stadermann and G. Rigoll, Two-stage speaker adaptation of hybrid tied-posterior acoustic models, *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, Philadelphia, PA, USA, Mar. 23–25, 2005, pp. 977–980.
6. D. Albesano et al., Adaptation of artificial neural networks avoiding catastrophic forgetting, *IEEE Int. Joint Conf. Neural Network Proc.*, Vancouver, Canada, July 16–21, 2006, pp. 1554–1561.
7. S. Xue et al., *Fast adaptation of deep neural network based on discriminant codes for speech recognition*, IEEE/ACM Trans. Audio, Speech, Language Process **22** (2014) no. 12, 1713–1725.
8. T. Tan, Y. Qian, and K. Yu, *Cluster adaptive training for deep neural network based acoustic model*, IEEE/ACM Trans. Audio, Speech, Language Process **24** (2016) no. 3, 459–468.
9. G. Saon et al., Speaker adaptation of neural network acoustic models using i-vectors, *IEEE Workshop Autom Speech Recogn. Understanding*, Olomouc, Czech Republic, Dec. 8–12, 2013, pp. 55–59.
10. A. Senior and I. Lopez-Moreno, Improving DNN speaker independence with i-vector inputs, *IEEE Int. Conf. Acoust. Speech Signal Process.*, Florence, Italy, May 4–9, 2014, pp. 225–229.
11. J. Trmal, J. Zelinka, and L. Müller, Adaptation of a feedforward artificial neural network using a linear transform, *Int. Conf. Text Speech Dialogue Proc.*, Brno, Czech Republic, Sept. 6–10, 2010, pp. 423–430.
12. B. Li and K.C. Sim, Comparison of discriminative input and output transformations for speaker adaptation in the hybrid NN/HMM systems, *Annu. Conf. Int. Speech Commun. Assoc.*, Makuhari, Japan, Sept. 26–30, 2010, pp. 526–529.
13. R. Gemello et al., Adaptation of hybrid ANN/HMM models using linear hidden transformations and conservative training, *IEEE Int. Conf. Acoust. Speech Signal Process. Proc.*, Toulouse, France, May 14–19, 2006, pp. 1189–1192.
14. K. Yao et al., Adaptation of context-dependent deep neural networks for automatic speech recognition, *IEEE Spoken Language Technol. Workshop*, Miami, FL, USA, Dec. 2–5, 2012, pp. 366–369.
15. Y. Zhao et al., Investigating online low-footprint speaker adaptation using generalized linear regression and click-through data, *IEEE Int. Conf. Acoust. Speech Signal Process.*, Brisbane, Australia, Apr. 19–24, 2015, pp. 4310–4314.
16. X. Li and J. Bilmes, Regularized adaptation of discriminative classifiers, *IEEE Int. Conf. Acoust. Speech Signal Process.*, Toulouse, France, May 14–19, 2006, pp. 237–240.
17. D. Yu et al., KL-divergence regularized deep neural network adaptation improved large vocabulary speech recognition, *IEEE Int. Conf. Acoust. Speech Signal Process.*, Vancouver, Canada, May 26–31, 2013, pp. 7893–7897.
18. P. Swietojanski and S. Renals, Learning hidden unit contributions for unsupervised speaker adaptation of neural network acoustic models, *IEEE Spoken Language Technol. Workshop*, South Lake Tahoe, NV, USA, Dec. 7–10, 2014, pp. 171–176.
19. J. Xue et al., Singular value decomposition based low-footprint speaker adaptation and personalization for deep neural network, *IEEE Int. Conf. Acoust. Speech Signal Process.*, Florence, Italy, May 4–9, 2014, pp. 6359–6363.
20. K. Kumar et al., Intermediate-layer DNN Adaptation for offline and session-based iterative speaker adaptation, *Annu. Conf. Int. Speech Commun. Assoc.*, Dresden, Germany, Sept. 6–10, 2015, pp. 1091–1095.
21. Y. Zhao, J. Li, and Y. Gong, Low-rank plus diagonal adaptation for deep neural networks, *IEEE Int. Conf. Acoust. Speech Signal Process.*, Shanghai, China, Mar. 20–25, 2016, pp. 5005–5009.
22. D. Yu and L. Deng, *Automatic speech recognition: A deep learning approach*, Spinger-Verlag London, UK, 2015, pp. 57–65.
23. I. Sutskever et al., On the importance of initialization and momentum in deep learning, *Proc. Int. Conf. Mach. Learn.*, Atlanta, GA, USA, June 16–21, 2013, pp. 1139–1147.
24. S. Pan and Q. Yang, *A survey on transfer learning*, IEEE Trans. Knowl. Data Eng. **22** (2010), no. 10, 1345–1359.
25. J. Huang et al., Cross-language knowledge transfer using multilingual deep neural network with shared hidden layers, *IEEE Int.*

Conf. Acoust. Speech Signal Process., Vancouver, Canada, May 26–31, 2013, pp. 7304–7308.

26. D. Povey et al., The Kaldi speech recognition toolkit, *IEEE Workshop Autom. Speech Recogn*. Understanding, Waikoloa, HI, USA, Dec. 11–15, 2011.
27. Y. Miao, H. Zhang, and F. Metze, *Speaker adaptive training of deep neural network acoustic models using i-vectors*, IEEE/ACM Trans. Audio, Speech, Language Process. **23** (2015), no. 11, 1938–1949.
28. D. Snyder, D. Garcia-Romero, and D. Povey, Time delay deep neural network-based universal background models for speaker recognition, *IEEE Workshop Autom. Speech Recogn. Understanding*, Scottsdale, AZ, USA, Dec. 13–17, 2015, pp. 92–97.

AUTHOR BIOGRAPHIES

Donghyun Kim received the BS and MS degrees in computer and communication engineering from Korea University, Seoul, Rep. of Korea, in 1999 and 2004, respectively, and the PhD degree in computer science from Korea University in 2008. Since 2009, he has been working for Electronics and Telecommunications Research Institute, Daejeon, Rep. of Korea. His interests are speech recognition, spoken language processing, and machine learning.

Sanghun Kim received the BS degree in electrical engineering from Yonsei University, Seoul, Rep. of Korea in 1990, the MS degree in electrical engineering and electronic engineering from KAIST, Daejon, Rep. of Korea in 1992, and the PhD degree from the Department of Electrical, Electronic, and Information Communication Engineering from the University of Tokyo, Japan in 2003. Since 1992, he has been working for Electronics and Telecommunications Research Institute, Daejeon, Rep. of Korea. His research interests are speech translation, spoken language understanding, and multi-modal information processing.