DNN ONLINE ADAPTATION FOR AUTOMATIC SPEECH RECOGNITION

Xinwei Li, Yue Pan, Matthew Gibson, Puming Zhan

Nuance Communications, Inc.
{xinwei.li, yue.pan, matt.gibson, puming.zhan}@nuance.com

Abstract: Although DNN-HMM based ASR systems can provide better accuracy than GMM-HMM based ASR systems in general, their performance still suffers from mismatches between the training and testing conditions. Online adaptation is a very effective way to make an ASR system more robust to a variety of environments and speaker characteristics. However, given large number of DNN parameters and only a limited amount of adaptation data, it is very challenging to perform DNN online adaptation effectively. In this paper, we propose two methods, namely i-vector and KL-Divergence regularized Linear Hidden Network, for performing DNN online adaptation for real-time speech recognition systems. The proposed methods were evaluated on a voice search data set. Over 3% relative word error rate reduction (WERR) was achieved from each of the proposed methods alone. A further relative WERR of over 2% was achieved from combining them.

1 Introduction

Deep Neural Networks (DNN) techniques are widely used for acoustic modeling in the state-of-the-art Automatic Speech Recognition (ASR) systems nowadays [1,2]. Although DNN-HMM based ASR systems can provide better accuracy than GMM-HMM based ASR systems in general, they are still very sensitive to speaker, channel, and background changes. This is reflected by large variance in accuracy across different speakers, channels, and environments that we have observed. Online adaptation is a very effective way to make an ASR system more robust to a variety of environments and speaker characteristics. It has been widely used in HMM-GMM based ASR systems. However, it is very challenging to perform DNN online adaptation effectively, because a DNN acoustic model usually contains millions of parameters and adapting them with a very small amount of data (i.e. a single utterance) in a traditional way is usually not effective for improving accuracy. On the other hand, online adaptation is subject to strict constraints on computational cost because of the latency requirement for real time speech recognition systems. Many approaches for DNN adaptation have been developed over the years, such as linear transformation based approaches in [6,7,14], regularization based approaches in [9,16] and i-vector based approaches in [5,15]. Most of them use offline supervised DNN adaptation when the adaptation data is available beforehand and there's no real-time constraint on the computational cost.

In this paper, we propose two methods, namely i-vector and KL-Divergence (KLD) regularized Linear Hidden Network (LHN), for performing DNN online adaptation for real time speech recognition systems. We demonstrate that they each can provide significant accuracy improvement and also show that combination of the two methods can further improve the performance.

The paper is organized as follows: Section 2 describes the online i-vector adaptation for DNNs. Section 3 summarizes the online KLD regularized LHN adaptation for DNNs. Experimental results are shown in Section 4. We conclude the paper in Section 5.

2 Online i-vector adaptation for DNNs

The i-vector technique is very popular in speaker verification and speaker recognition because they encapsulate the relevant information about a speaker's identity in a low dimensional fixed-length representation [4, 10, 11]. This also makes them an attractive tool for speaker adaptation techniques for ASR [5].

The i-vector extraction has been studied in many works [4, 10, 11]. Suppose the acoustic feature vectors $\boldsymbol{x}_t \in R^D$ are generated from a universal background model (UBM) represented as a GMM with K diagonal covariance Gaussians

$$\boldsymbol{x}_t \sim \sum_{k=1}^{K} c_k N(.; \boldsymbol{\mu}_k(0), \boldsymbol{\Sigma}_k) \tag{1}$$

with mixture coefficients c_k, means $\boldsymbol{\mu}_k$ and diagonal covariances $\boldsymbol{\Sigma}_k$. Moreover, data $\boldsymbol{x}_t(s)$ belonging to speaker s are drawn from the distribution

$$\boldsymbol{x}_t(s) \sim \sum_{k=1}^{K} c_k N(.; \boldsymbol{\mu}_k(s), \boldsymbol{\Sigma}_k) \tag{2}$$

where $\boldsymbol{\mu}_k(s)$ are the means of the GMM adapted to speaker s. The main idea about the i-vector is to assume the speaker-dependent means $\boldsymbol{\mu}_k(s)$ and the speaker-independent means $\boldsymbol{\mu}_k(0)$ can be modeled as

$$\boldsymbol{\mu}_k(s) = \boldsymbol{\mu}_k(0) + \mathbf{T}_k \boldsymbol{w}(s), \qquad k = 1 \cdots K \tag{3}$$

$\mathbf{T}_k$ of size $D \times M$, is called the factor loading submatrix corresponding to component k. $\boldsymbol{w}_s$ is the speaker identity vector (i-vector) corresponding to s. Each $\mathbf{T}_k$ contains M bases which span the subspace with important variability in the component mean vector space.

Given the UBM, the factor loading submatrix $\mathbf{T}_k$ and speaker data$\{\boldsymbol{x}_t(s)\}$, i-vector $\boldsymbol{w}_s$ can be estimated with the eq. (4).

$$\boldsymbol{w}(s) = \mathbf{L}^{-1}(s) \sum_{k=1}^{K} \mathbf{T}_k^T \boldsymbol{\Sigma}_k^{-1} \boldsymbol{\theta}_k(s), \tag{4}$$

where

$$\mathbf{L}(s) = \mathbf{I} + \sum_{k=1}^{K} \gamma_k(s)\, \mathbf{T}_k^T \boldsymbol{\Sigma}_k^{-1} \mathbf{T}_k, \tag{5}$$

$$\gamma_k(s) = \sum_t \gamma_{tk}(s), \tag{6}$$

$$\boldsymbol{\theta}_k(s) = \textstyle\sum_t \gamma_{tk}(s)\, (\boldsymbol{x}_t(s) - \boldsymbol{\mu}_k(0)), \tag{7}$$

where $\gamma_k(s)$ and $\boldsymbol{\theta}_k(s)$ are the zero-order and centered first-order statistics accumulated from speaker data $\{\boldsymbol{x}_t(s)\}$. $\mathbf{L}^{-1}(s)$ is the posterior covariance of $\boldsymbol{w}(s)$ given the observations of speaker s.

Model hyperparameters $\{\mathbf{T}_1, \cdots, \mathbf{T}_k\}$ can be estimated using the EM algorithm to maximize the ML objective function [11]. In the E step, the accumulators in eqs. (8) and (9) are collected over all the speakers.

$$\mathbf{C}_k = \sum_s \boldsymbol{\theta}_k(s)\boldsymbol{w}^T(s), \tag{8}$$

$$\mathbf{A}_k = \sum_s \gamma_k(s)(\mathbf{L}^{-1}(s) + \boldsymbol{w}(s)\boldsymbol{w}^T(s)) \tag{9}$$

The M step update is given as eq. (10):

$$\mathbf{T}_k = \mathbf{C}_k\mathbf{A}_k^{-1} \quad k = 1 \cdots K \tag{10}$$

Using i-vector for DNN adaptation has been studied in [5] where the input features to DNN are augmented with i-vectors computed from the speakers in the training set for training the DNN acoustic model. At recognition time, the i-vector for a given speaker is computed with the adaptation data from the speaker in offline mode and augmented with the speech features in the decoding process. So the i-vector based adaptation described in [5] is supervised and is done in offline batch-mode. Our focus in this paper is to apply such i-vector based DNN adaptation in online mode in which i-vector is computed and updated while the recognition process is ongoing. One major challenge with such approach is how to take advantage of having more and more available data down the session. Another major challenge is how to update i-vector in an efficient way so that the updating process will not cause extra latency to a real time ASR system which usually has strict constraint to latency. Since there is no available data to be used for computing i-vector at the very beginning of a session, we also need to deal with this situation as well.

We proposed to carry out i-vector based online adaptation in a per utterance basis along with an ongoing recognition session. That is to calculate i-vector with the currently available utterances in the session and use it in recognizing the next utterance in the same session. We proposed to carryover either the sufficient statistics or the i-vector estimated based on the previous utterances in a session in the i-vector updating process for taking advantage of more available utterances as the recognition process proceeds. We used a universal i-vector estimated over all the training data as the initial i-vector in recognizing the first utterance in each session. The i-vector is updated immediately after each utterance is recognized. In case of sufficient statistics carryover, the statistics accumulated from the current recognized utterance is first merged with the history statistics accumulated from the previous utterances in the same session. Then a refined i-vector is estimated based on the merged statistics. In case of i-vector carryover, the new i-vector is a weighted average of the previous i-vector and the one estimated with the current recognized utterance. Length normalization of i-vectors has been shown to be effective to boost performance in speaker recognition systems [12]. In our work described in this paper, we also observed that normalizing i-vector to unit length could improve performance.

3 Online KLD regularized LHN adaptation for DNNs

The basic idea of LHN [7] is illustrated in Figure 1. A linear layer is inserted immediately before the output layer to transform the activation of the last hidden layer to better match the adaptation data.

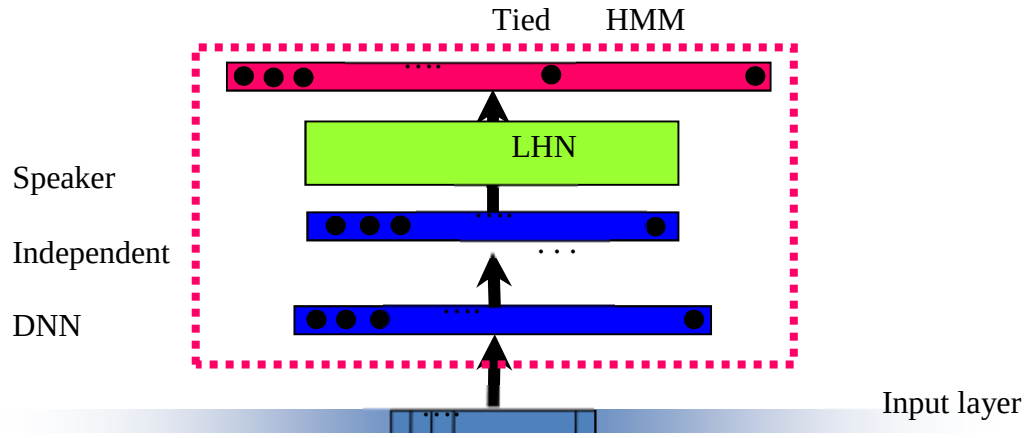

Figure 1 - Illustration for Linear Hidden Network(LHN)

The weights of LHN are initialized with an identity matrix. A standard back-propagation algorithm is used to train the weights of the LHN with the adaptation data while keeping the weights of the original network fixed. However, adapting a DNN with limited amount of data using the standard back-propagation can cause the DNN to overfit the data, hence deteriorate the previously well trained networks. To mitigate the overfitting problem, KLD regularization [9] can be applied to the adaptation criterion. The idea of KLD regularization is to prevent the posterior distribution estimated from the adapted DNN from deviating too far away from that estimated with the original DNN by adding the KLD between the two distributions to the Cross Entropy objective function.

$$\widehat{D} = (1-\rho)\frac{1}{N}\sum_{t=1}^{N}\sum_{y=1}^{S}\tilde{p}(y|x_t)\log p(y|x_t) + \rho\frac{1}{N}\sum_{t=1}^{N}\sum_{y=1}^{S}p^{SI}(y|x_t)\log p(y|x_t) \quad (11)$$

where N is the number of samples in the adaptation data and $\tilde{p}(y|x_t)$ is the target probability. $p^{SI}(y|x_t)$ is the posterior probability estimated from the SI DNN and computed with a forward pass using the SI DNN. ρ is the regularization weight. S is the total number of states in the output layer. Eq. (14) can be reorganized to

$$\widehat{D} = \frac{1}{N}\sum_{t=1}^{N}\sum_{y=1}^{S}[(1-\rho)\tilde{p}(y|x_t) + \rho p^{SI}(y|x_t)]\log p(y|x_t) \quad (12)$$

$$= \frac{1}{N}\sum_{t=1}^{N}\sum_{y=1}^{S}\hat{p}(y|x_t)\log p(y|x_t) \quad (13)$$

where

$$\hat{p}(y|x_t) \triangleq (1-\rho)\tilde{p}(y|x_t) + \rho p^{SI}(y|x_t) \quad (14)$$

From eqs. (11) to (14), we can see that applying the KLD regularization to the cross entropy criterion equals to changing the target probability distribution from $\tilde{p}(y|x_t)$ to $\hat{p}(y|x_t)$, which is a linear interpolation of the distribution estimated from the SI DNN and the ground truth alignment of the adaptation data.

Like online i-vector adaptation, online LHN adaptation is also carried out in a per utterance basis. The weights of LHN are updated iteratively with the data from the utterance that has just been recognized in a session. Therefore, it is unsupervised adaptation. Early stopping can be applied to the process to further mitigate the overfitting problem and reduce the adaptation time.

4 Experimental results

We conducted adaptation experiments based on a voice search data set, which contains over 3000 hours of audio data for training the DNN acoustic models. There are over 1 million speakers in the training data set. A large number of speakers have less than 10 seconds of audio. We evaluated the performance on a test data set which contains about 90 hours of audio data. There are 901 speakers with over 35k utterances and 330k words in the test set. The number of utterances per speaker ranges from 1 to over 100.

The speech data was processed using a 25ms window with window shift 10ms. Each frame of the acoustic features contains 45-dimensional MFCC plus 7-dimensional Fundamental Frequency Variation (FFV) features [13] to form a 52-dimensional feature vector. All dimensions in the acoustic features were quantized to 8-bit integers.

A diagonal UBM for i-vector extraction with 2048 32-dimensional Gaussian mixture components was built on the training data using the maximum likelihood criterion. Only the first 12 dimensions of the 45-dimensional MFCC were used in the UBM training. 9 consecutive frames were spliced together and projected down to 32 dimensions using LDA. The speakers in the training data set were clustered into about 12k speaker groups to speed up the i-vector training. All speakers in the same group shared one i-vector. The dimension of the i-vector was set to 100 in our experiments. After i-vectors were extracted for each speaker group, length normalization was applied to them.

Two feed-forward DNNs were trained from the training data: acoustic features only DNN as the baseline and i-vector augmented DNN. The difference between the two DNNs is whether the i-vectors are included in the input features. Both DNNs share the same topology for the rest of the network. With a context window of 15, the dimension of the input vector for the baseline DNN is 780, while that for the i-vector augmented DNN is 880. There are 7 hidden layers with ReLU activation functions in the networks. The first 2 layers have 2048 nodes each. The next 2 layers have 1024 nodes each. The next layer is a bottleneck layer with 52 nodes for parameter reduction, followed by another layer with 256 nodes. The output layer has 9000 softmax units that correspond to the tied context-dependent HMM states obtained from a phonetic decision tree. Both DNNs were first trained with the stochastic gradient descent (SGD) algorithm using the frame-level Cross Entropy criterion followed by the Hessian-free sequence training algorithm using a state-based minimum Bayes risk criterion as described in [17].

During recognition, the online adaptation was carried out in a per utterance basis. The i-vector and LHN transformation matrix were updated using the utterance that had just been recognized. The LHN transformation matrix was inserted after the bottleneck layer with a dimension of 52 by 52. The i-vector was initialized as the universal i-vector and the LHN transformation matrix was initialized as an identity matrix for the first utterance of each speaker. The recognition output was used as the pseudo-truth for LHN adaptation.

Model	WER (%)	WERR (%)
Baseline	9.29	-
+i-vector	8.98	3.34
+LHN	8.96	3.55
+i-vector+LHN	8.77	5.60

Table 1 - WER and relative WERR for online I-vector and LHN adaptation

In Table 1, we compare the WERs obtained by using different online adaptation methods, i.e. i-vector only, LHN only, and combination of i-vector and LHN. The baseline DNN without any adaptation achieved 9.29% WER. All three online adaptation methods outperformed the baseline. The relative WERR from i-vector only online adaptation is 3.34%. The relative WERR from LHN only online adaptation is 3.55%. A relative WERR of 5.60% is achieved by combining i-vector with LHN which brings down the WER to 8.77%. The sufficient statistics carryover method was used for the i-vector based adaptation in Table 1. We also experimented the i-vector carryover method and observed almost the same accuracy improvement as with the statistics carryover method. Since the i-vector carryover method requires much less memory compared with the statistics carryover method, it could be more appealing for memory limited usage case.

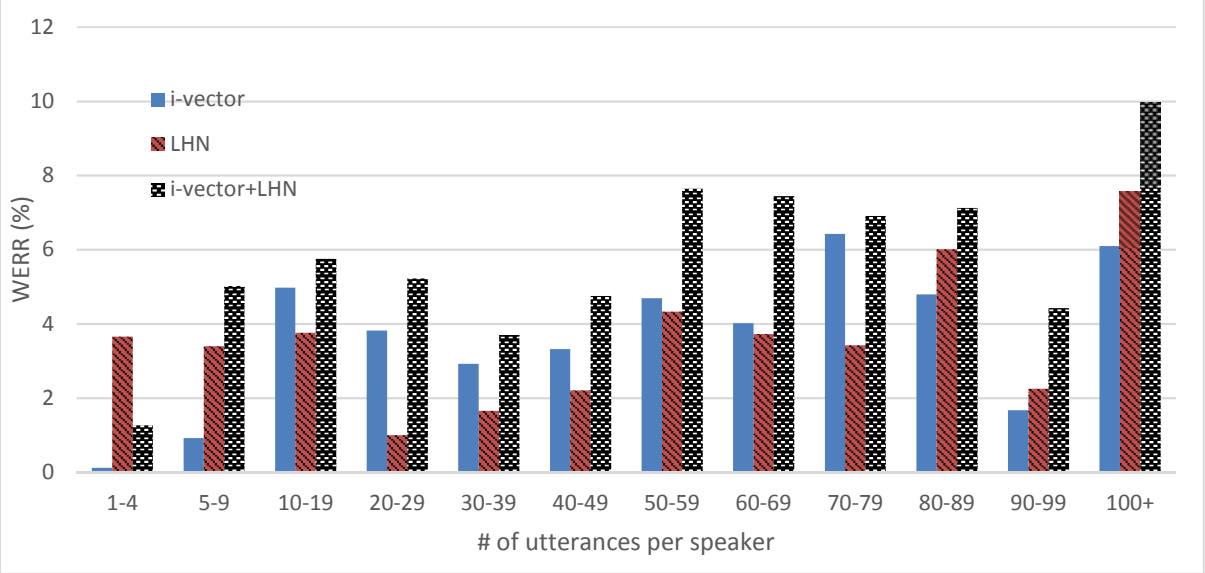

Figure 2 - Relative WERR for speaker groups with different number of utterance

To investigate the impact of the number of utterances for a speaker on the adaptation performance, we divide the speakers into 12 groups based on the number of utterances per speaker as shown in the x-axis in Figure 2. The y-axis presents the relative WERR over the baseline for the speakers in each group. From the figure, we can see that the gain from LHN only and i-vector+LHN adaptation is significant for all speaker groups. The i-vector only adaptation outperforms the baseline significantly for most speaker groups except the one with 1 to 4 utterances. This seems to indicate that the i-vector is not very robust when the amount of adaptation data is very small. However, the KLD regularized LHN works very well even with just a few utterances.

5 Conclusions

In this paper, we proposed to use i-vector and LHN to perform DNN online adaptation in a per utterance basis for real time ASR systems. For i-vector based online adaptation, we proposed to carryover either i-vector sufficient statistics or the previous i-vector itself from utterance to utterance for taking advantage of having more data along a session to improve the performance. We also proposed to use a universal i-vector, which is estimated with all training data, in recognizing the first utterance to improve accuracy for the first utterance in a session. For LHN adaptation, we investigated to apply the KLD regularization to the objective function to overcome the overfitting problem and apply early stopping to the optimization process to reduce the adaptation time. Experiments on a voice search task demonstrated that our proposed approaches made i-vector and LHN adaptation effective and affordable for performing online adaptation for real time speech recognition systems. Over 3% relative word error reduction was achieved from the i-vector only online adaptation and the LHN only online adaptation individually. A further improvement of over 2% was achieved from combining i-vector with LHN online adaptation. Note that our proposed online adaptation methods can be used along with offline batch-mode adaptation. In this case, the initial i-vector and LHN transformation matrix will be estimated with the adaptation data.

6 References

[1] F. SEIDE, G. LI, X. CHEN, AND D. YU, *"Feature engineering in context-dependent deep neural networks for conversational speech transcription,"* in Automatic Speech Recognition and Understanding (ASRU), 2011 IEEE Workshop on, Dec 2011, pp. 24– 29.

[2] G. E. DAHL, D. YU, L. DENG, AND A. ACERO, *"Context-dependent pre-trained deep neural networks for large vocabulary speech recognition,"* IEEE Trans. on Audio, Speech, and Language Processing, vol. 20, no. 1, pp. 30–42, 2012

[4] N. DEHAK, P. KENNY, R. DEHAK, P. DUMOUCHEL, AND P. OUELLET, *"Frontend factor analysis for speaker verification,"* IEEE Trans. Audio, Speech and Language Processing, vol. 19, no. 4, May 2011

[5] G. SAON, H. SOLTAU, D. NAHAMOO, AND M. PICHENY, *"Speaker adaptation of neural network acoustic models using i-vectors,"* in Automatic Speech Recognition and Understanding (ASRU), 2013 IEEE Workshop on, Dec 2013, pp. 55–59.

[6] J. NETO, L. ALMEIDA, M. HOCHBERG, C. MARTINS, L. NUNES, S. RENALS, AND T. ROBINSON, *"Speaker-adaptation for hybrid HMM-ANN continuous speech recognition system,"* in EUROSPEECH, 1995.

[7] R. GEMELLO, F. MANA, S. SCANZIO, P. LAFACE, AND R. D. MORI, *"Adaptation of hybrid*

ANN/HMM models using linear hidden transformations and conservative training," in ICASSP, 2006, pp. 1189–1192.

[8] K. YAO, D. YU, F. SEIDE, H. SU, L. DENG, AND Y. GONG, "*Adaptation of context-dependent deep neural networks for automatic speech recognition*," in Proc. SLT, 2012.

[9] D. YU, K. YAO, H. SU, G. LI, AND F. SEIDE, "*KL-divergence regularized deep neural network adaptation for improved large vocabulary speech recognition*," in ICASSP, May 2013, pp. 7893–7897.

[10] O. GLEMBEK, L. BURGET, P. MATEJKA, M. KARAFIAT, AND P. KENNY, "*Simplification and optimization of i-vector extraction,"* in Proc. ICASSP 2011.

[11] P. KENNY, "*Joint factor analysis of speaker and session variability: Theory and algorithms – technical report CRIM-06/08-13. Montreal, CRIM, 2005,"* 2005.

[12] D. GARCIA-ROMERO, AND C. ESPY-WILSON, "*Analysis of I-vector Length Normalization in Speaker Recognition Systems*," in Proc. Interspeech 2011.

[13] K. LASKOWSKI, M. HELDNER, AND J. EDLUND, "*The fundamental frequency variation spectrum,"* in FONETIK, 2008.

[14] K. KUMAR, C. LIU, K. YAO, AND Y. GONG, "*Intermediate-layer DNN Adaptation for Offline and Session-based Iterative Speaker Adaptation*," in Proc. Interspeech 2015.

[15] H. ARSIKERE, AND S. GARIMELLA, "*Robust online i-vectors for unsupervised adaptation of DNN acoustic models: A study in the context of digital voice assistants*," in Proc. Interspeech 2017.

[16] X. LI, AND J. BILMES, "*Regularized adaptation of discriminative classifiers*," in Proc. ICASSP 2006.

[17] B. KINGSBURY, T. SAINATH, AND H. SOLTAU, "*Scalable minimum Bayes risk training of deep neural network acoustic models using distributed Hessian-free optimization*," in Proc. Interspeech, 2012.