Robust i-vector based Adaptation of DNN Acoustic Model for Speech Recognition

Sri Garimella[1], *Arindam Mandal*[2], *Nikko Strom*[2], *Bjorn Hoffmeister*[2]
Spyros Matsoukas[2], *Sree Hari Krishnan Parthasarathi*[2]

[1]Amazon.com, India
[2]Amazon.com, USA
{srigar,arindamm,nikko,bjornh,matsouka,sparta}@amazon.com

Abstract

In the past, conventional i-vectors based on a Universal Background Model (UBM) have been successfully used as input features to adapt a Deep Neural Network (DNN) Acoustic Model (AM) for Automatic Speech Recognition (ASR). In contrast, this paper introduces Hidden Markov Model (HMM) based i-vectors that use HMM state alignment information from an ASR system for estimating i-vectors. Further, we propose passing these HMM based i-vectors though an explicit non-linear hidden layer of a DNN before combining them with standard acoustic features, such as log filter bank energies (LFBEs). To improve robustness to mismatched adaptation data, we also propose estimating i-vectors in a causal fashion for training the DNN, restricting the connectivity among hidden nodes in the DNN and applying a max-pool non-linearity at selected hidden nodes. In our experiments, these techniques yield about 5-7% relative word error rate (WER) improvement over the baseline speaker independent system in matched condition, and a substantial WER reduction for mismatched adaptation data.

Index Terms: speech recognition, adaptation of DNN acoustic model, i-vector, robustness

1. Introduction

A DNN AM is prevalent in many state-of-the-art ASR systems [1]. It is typically trained using the LFBE features as input to estimate the posterior probabilities of senones (or tied context dependent HMM states) at output. One of the challenging problems is to adapt such a DNN using limited amount of speaker and/or channel data, currently an area of active research. Several techniques have been proposed to address this problem. They are broadly classified as feature space adaptation [2], [3], model space adaptation [4] and training with an additional speaker and channel information [5], [6], [7]. This paper falls under the last category.

Recently, improvements in ASR performance have been reported by training DNNs that use input feature vectors formed by concatenation of LFBEs and Gaussian Mixture Model (GMM) based UBM i-vectors [7], [8]. In this case, an i-vector specifies how the UBM-GMM should be adapted in an affine subspace of the parameter space in order to capture the variability of the underlying speech segment [9], [10]. In a different work [11], it is observed that using the posterior probability estimates from a DNN AM for accumulating i-vector sufficient statistics has resulted in significant gain for speaker verification. This paper extends this idea by introducing HMM-GMM i-vectors which use HMM states alignment information from an ASR system for accumulating the sufficient statistics. Instead of using the posterior estimates from a DNN, the alignments from a HMM-DNN ASR system are used for accumulating the i-vector sufficient statistics. Thus it allows us to exploit the reference transcripts of training data or the rich LM prior information when determining state alignments.

In an earlier work on using utterance i-vectors as features [8], it is empirically observed that regularization of a DNN is necessary for obtaining better generalization and to avoid overfitting. In this paper, we propose an alternative approach for achieving generalization by passing the i-vectors through an explicit non-linear hidden layer and then combine with the LFBEs. The i-vector specific non-linear hidden layer is crucial for obtaining improvements as removing it did not provide any gain over the baseline system.

Another aspect of the i-vector based DNN speaker adaptation that is not studied earlier is the effect of i-vector mismatch during decoding. This situation arises in a supervised adaptation setting where some amount of adaptation data is available for each test speaker. We study the effect of using an i-vector estimated on two minutes each of matched, multi-user and mismatched adaptation datasets in our experiments. Consequently, several techniques are introduced in this paper to improve robustness of the DNN ASR system to mismatched adaptation data. First, the training data for each speaker is modified by randomly picking utterances from the total speaker pool and inserting them between the current speaker's utterances, thus simulating sudden speaker changes. Then we propose to use causal i-vectors, which are estimated for each utterance of a speaker taking into account the past utterances, for training the DNN. This is in contrast to speaker i-vectors [7] and utterance i-vectors [8] that have been reported in the literature. Secondly, the connectivity among hidden nodes of a DNN is restricted to make certain hidden nodes independent of the input i-vector and thus making them robust to any i-vector mismatch. Finally, a max-pool component is used to sample from hidden representation of a DNN.

2. System Description

We propose to introduce a non-linear hidden layer to transform i-vectors and then combine with LFBEs for better generalization as depicted in Figure 1. Two types of i-vectors are used in our experiments - the standard UBM-GMM i-vectors and the proposed HMM-GMM i-vectors. The remainder of this section describes our experimental setup and the results.

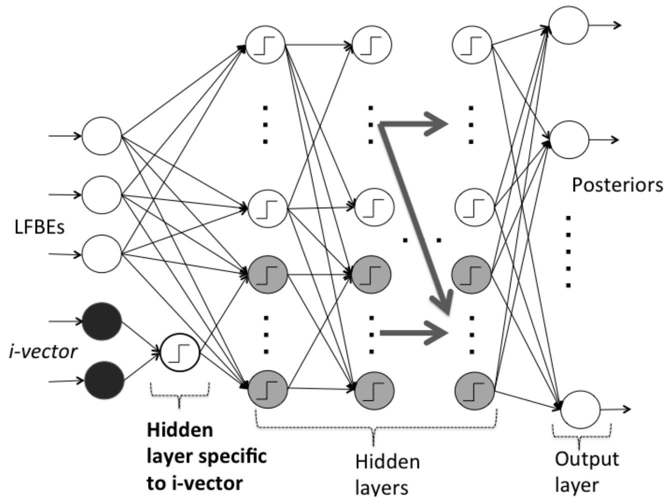

Figure 1: A DNN depicting a non-linear hidden layer specific to i-vector and restricted connectivity among hidden nodes. We use multiple hidden nodes in i-vector specific hidden layer, and optionally use restricted connectivity in our experiments.

2.1. Corpora

In our experiments, we use a training dataset of roughly 200 hours of speech data from an in-house collection sampled at 16 kHz; this is a subset of the data set used for training our production models. From the same data collection we carve out a cross-validation set of 20 hours, used to optimize meta-parameters, and a test set of 10 hours on which we report WER improvements. A randomly chosen 10 hours of training data is used for DNN pre-training. Training, validation, and test set do not have any speaker overlap.

I-vector corresponding to each test speaker is estimated on a separate two minutes of supervised (i.e., underlying word sequence is known) adaptation data, and three different types of adaptation datasets are used - matched, multi-user and mismatched. As names suggest, matched and mismatched adaptation datasets contain utterances from a matched and random mismatched speaker respectively for each test speaker. Where as in multi-speaker adaptation dataset, half of the utterances match and rest do not match each test speaker. The only exception being UBM-GMM i-vector results in Table 1, where utterance i-vectors are extracted on test set to show the effectiveness of the i-vector specific non-linear hidden layer.

2.2. LFBE features

The LFBE feature vectors are extracted for every 10 ms using an analysis window of length 25 ms. A discrete fourier transform (DFT) is applied on each windowed signal and then power spectrum is computed by taking the absolute square values of the DFT output. The resultant power spectrum is integrated using 20 mel windows, and finally a natural logarithm is applied to obtain the LFBE features. They are mean normalized by subtracting the running mean estimate of LFBEs. For training a DNN, both the mean and variance in each dimension are normalized across the training data.

2.3. UBM-GMM i-vectors

An UBM-GMM with 1024 diagonal Gaussian components is trained on the 40 dimensional features obtained from an ASR system. The features are obtained by applying the block diagonal discrete cosine transform (DCT), the linear discriminant analysis (LDA) and the maximum likelihood linear transform (MLLT) [12] respectively on the concatenated LFBE features of context nine frames. Both LDA and MLLT transforms are obtained from an ASR system. The $\mathbf{T}$ matrix for estimating the i-vectors is randomly initialized and trained using the expectation maximization (EM) algorithm [9]. As in LFBE features, global mean and variance normalization is applied to i-vectors for training a DNN.

2.4. HMM-GMM i-vectors

In this case, a single Gaussian per HMM state replaces the UBM-GMM for estimating i-vectors. Specifically, each tied context dependent phone state specified by the acoustic decision tree is modeled using a single diagonal Gaussian. Both, inference and training of an i-vector, require the information about which Gaussian components of the model have generated the observed feature vectors of a segment or utterance. This information is obtained from the state alignment of an ASR system that uses the same decision tree and the same HMM structure. Notice that one can use either an HMM-DNN or an HMM-GMM AM based ASR system. During training the HMM state sequence is derived by doing a forced-alignment against the reference transcript, and at runtime the state sequence from the ASR decoding result is used. The HMM state sequence is used for accumulating the sufficient statistics for estimating an i-vector. In an earlier work [11], authors have already proposed to use a single Gaussian per state HMM-GMM AM for obtaining i-vectors. However, the main difference of their approach is that they are using frame level estimates of posterior probabilities of senones from a DNN for accumulating the sufficient statistics, which do not take either reference transcript or LM prior information into account.

The HMM-GMM AM for extracting i-vectors is trained on the 40 dimensional features described in section 2.3. The AM has 3052 tied HMM states. As in UBM-GMM case, the $\mathbf{T}$ matrix for estimating i-vectors is randomly initialized and trained using the EM algorithm. Both, training and adaptation (or test) i-vectors, are computed using the baseline LFBE-DNN ASR model. The feature vectors corresponding to the silence and non-speech phones are dropped when computing an i-vector.

2.5. DNN training

A DNN is discriminatively pre-trained one layer at a time by minimizing the cross-entropy cost function between the targets and its outputs on 10 hours of pre-training data described in section 2.1. After the pre-training step, it is trained to minimize the cross-entropy cost of the full training set. The targets for training the DNN are obtained by force-aligning the reference transcripts using a HMM-GMM system. The parameters of the DNN are updated using the stochastic gradient descent with a mini-batch size of 256 and an initial learning rate of 0.008. The learning rate is exponentially reduced as a function of epoch for 12 epochs.

The baseline LFBE-DNN is trained on input features obtained by concatenating 11 frames of LFBEs. It estimates 3052 posterior probabilities of senones corresponding to the tied context dependent HMM states at the output layer. The DNN has four non-linear sigmoid hidden layers with 1024 nodes per layer, and an output softmax layer. We report results as the relative reduction or relative improvement in WER compared to this baseline system. Furthermore, all experiments use the same acoustic decision tree and differ only in the DNN and the DNN input.

2.6. Results

In the first set of experiments, speaker adapted DNN models are trained by augmenting the baseline LFBEs with the 32 dimensional utterance UBM-GMM i-vectors. Three fully connected feed-forward DNNs are trained - one without an i-vector specific hidden layer and others with an i-vector specific hidden layer of sizes 16 and 32 nodes respectively. For decoding the test set, LFBEs are appended with the corresponding utterance UBM-GMM i-vector. Table 1 lists the relative WER improvements[1] of these systems over the baseline LFBE-DNN system described in Section 2.5. The results indicate that both speaker adapted systems with an i-vector specific hidden layer yield 5.3% relative WER improvement on the test set over the baseline system. The system augmenting the LFBEs directly with the i-vectors shows no improvement, showing the importance of passing i-vectors through an explicit non-linear hidden layer.

Table 1: *Importance of i-vector specific non-linear hidden layer.*

Training i-vector type	i-vector specific hidden layer size	Relative WER improvement (%)
utterance, UBM	N/A	0
utterance, UBM	16	**5.3**
utterance, UBM	32	5.3
speaker, HMM	N/A	-0.4
speaker, HMM	16	6.1
speaker, HMM	32	**7.0**

The next set of experiments use a 32 dimensional speaker HMM-GMM i-vectors. For each speaker in our training and cross-validation set, speaker HMM-GMM i-vector is obtained by accumulating statistics from all utterances of a speaker and then computing an i-vector. During test, HMM-GMM i-vector of each test speaker is estimated on a two minute matched adaptation set described in Section 2.1 and then it is appended to the LFBEs of test utterances corresponding to the speaker. Three speaker adapted fully connected DNNs are trained on concatenated LFBEs and HMM-GMM i-vectors. As in earlier case, two DNNs are trained with i-vector specific hidden layer and one without it. The relative WER improvement of these systems over the baseline LFBE-DNN system is summarized in Table 1. These results also confirm the benefits of adding an i-vector specific non-layer hidden layer for better generalization of the DNN AM.

All experiments conducted in the reminder of the paper use 32 dimensional i-vectors and 16-dimensional i-vector specific hidden layer. We study the effect of i-vector mismatch on WER and propose techniques to improve robustness to mismatched adaptation data.

3. Causal i-vectors

We propose to extract the causal i-vector (applicable to both UBM-GMM and HMM-GMM based i-vectors) of each utterance for training the DNN. A causal i-vector of an utterance of a speaker uses all previous utterances of that speaker for accumulating i-vector sufficient statistics. To make the causal i-vector encode recently encountered speaker and channel characteristics, exponentially decaying weights are applied on previous speech frames to give more importance to the recent audio.

Let $\mathbf{X}_u = \left[\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_{n_{u-1}}\right]$ be the set of all frames from previous utterances of u, and $\mathbf{Z}_u = \left[\mathbf{z}_0, \mathbf{z}_1, \ldots, \mathbf{z}_{n_{u-1}}\right]$ be the corresponding Gaussian alignments. Here $\mathbf{z}_t$ is a vector of all zeros except a one at an index corresponding to a Gaussian alignment, and its i^{th} entry is denoted as z_{ti}. Therefore, its dimensionality is same as the number of Gaussians in either UBM-GMM or HMM-GMM model. The zeroth and first order statistics corresponding to an i^{th} Gaussian and u^{th} utterance for estimating a causal i-vector are defined as follows:

$$\begin{aligned} \gamma_{u,i} &= \sum_{t:z_{ti}=1, 0 \le t \le n_{u-1}} e^{-(n_{u-1}-t)\tau} \\ \mathbf{f}_{u,i} &= \sum_{t:z_{ti}=1, 0 \le t \le n_{u-1}} e^{-(n_{u-1}-t)\tau} \left(\mathbf{x}_t - \mu_i\right) \end{aligned} \quad (1)$$

Where μ_i denotes the mean of i^{th} Gaussian, τ is a positive exponential decaying factor. The causal i-vector of an utterance u is estimated using the i-vector sufficient statistics defined in (1).

3.1. Results

Speaker adapted DNNs are trained using the speaker HMM-GMM i-vectors and the causal HMM-GMM i-vectors. Causal i-vectors are extracted after mixing utterances from random speakers into the sequence of utterances of a current speaker for training the DNN. Table 2 lists the relative WER improvements of these systems over the baseline on the test set. HMM-GMM i-vector corresponding to each test speaker is estimated on three different two minute adaptation datasets described in Section 2.1. The results clearly indicate usefulness of the causal i-vectors in training for improving robustness of the speaker adapted DNN system, especially in multi-speaker and mismatched speaker test i-vectors.

Table 2: *Robustness of training with HMM-GMM based causal i-vectors.*

Training i-vector type	matched	multi	mismatched
speaker	6.1	-14.9	-88.6
causal, 50% mixed	**7.0**	**-4.8**	**-52.2**

In the next set of experiments, speaker adapted DNN models are trained using the utterance UBM-GMM i-vectors and the causal UBM-GMM i-vectors. During test, a two minute matched adaptation dataset described in Section 2.1 is used for estimating the UBM-GMM i-vector corresponding to each test speaker. The relative WER improvement of these systems over the baseline LFBE-DNN model is summarized in Table 3. Notice that first row of the table corresponds to a case where utterance i-vectors are used in training and i-vector estimated on matched adaptation data is used for decoding. The two-minute adaptation data for each test speaker exceeds the typical length of a training utterance, resulting in a mismatch. Using the causal i-vectors in training enables the model to be robust to such length mismatch as varying lengths of data segments[2] are

[1] Positive indicates an improvement (or decrease) in WER and negative indicates an increase in WER over the baseline.

[2] Consider a sequence of utterances of a speaker. For the first ut-

used for extracting causal i-vectors. This is ascertained by results in Table 3, where systems trained on causal i-vectors yield better results than the one trained using utterance i-vectors.

Table 3: *Robustness of UBM-GMM based causal i-vectors to segment length mismatch.*

Features	Training i-vector type	Relative WER improvement (%)
LFBEs, i-vector	utterance	1.8
LFBEs, i-vector	causal	6.1
LFBEs, i-vector	causal, 50% mixed	**6.6**

4. DNN with Restricted Connectivity and Max-pool Non-linearity

This section introduces modifications to the DNN architecture to further improve robustness against speaker mismatch of the speaker adapted system.

4.1. Restricted Connectivity

A fraction of hidden nodes in each hidden layer are made independent of the input i-vector and depend only on the input LFBEs. The remaining hidden nodes are made dependent on both the feature streams as in a fully connected DNN. This is achieved by removing connections from the hidden nodes in previous layer that depend on the input i-vector. This configuration would make part of the DNN hidden representation unaffected by any i-vector mismatch, and thus providing robustness. Figure 1 depicts this idea where only non-shaded hidden nodes in each hidden layer are independent of the input i-vector.

4.2. Max-pool Non-linearity

In the DNN configuration described in Section 4.1 and shown in Figure 1, the number of i-vector-dependent hidden nodes is fixed. However, to make the ASR system more robust, it would be desirable to vary the i-vector-dependent hidden nodes depending on whether decoding with a mismatched or a matched i-vector. Whether the adaptation was mismatched is not known, but we use a max-pool non-linearity to enable the network to learn to sample from hidden representations that are either based on pure LFBEs or a combination of LFBEs and i-vectors. This is shown in Figure 2. The max-pool component outputs element-wise maximum of pairs of input vectors.

The network in Figure 2 with a tanh non-linearity is trained in four phases. In phase one, an LFBE-DNN with the restricted connectivity architecture, obtained by removing affine transforms 2 and 3, bottleneck layer and max-pool, is pre-trained and further trained for three epochs. In phase two, affine transform 2 is initialized with zeroes, affine transform 3 with random weights but its bias with zeroes and bottleneck layer parameters with random weights. In phase three, only bottleneck parameters and affine transform 3 are pre-trained, and then they along with affine transform 2 are trained for three epochs. In the last phase, all network parameters are trained for several epochs. The role of innovation affine transform is to adjust the LFBE hidden representation that is combined with the i-vector hidden representation to be independent of the LFBE hidden representation which directly goes into the max-pool non-linearity.

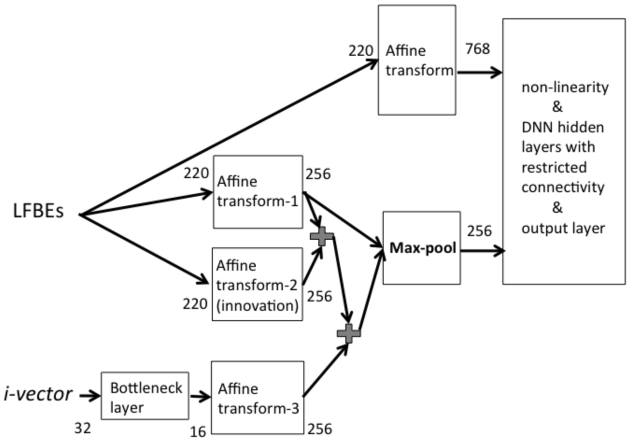

Figure 2: Block diagram of DNN AM with max-pool component and restricted connectivity.

4.3. Results

The relative WER improvement of several speaker adapted DNN systems over the baseline LFBE-DNN system is summarized in Table 4. The first row in this table corresponds to the system used in last row of Table 2. An additional robustness to i-vector mismatch is obtained by restricting the connectivity among hidden nodes by forcing 768 out of 1024 nodes in each hidden layer to be independent of the input i-vector as described in Section 4.1. Further robustness is obtained by adding a 256 dimensional max-pool non-linearity as described in Section 4.2. The results in Table 4 confirm the robustness of the final speaker adapted DNN system to mismatched and multi-user conditions without losing much in matched condition.

Table 4: *Robustness of restricting connectivity of a DNN and using a max-pool non-linearity. The relative WER improvements (in %) of speaker adapted DNN systems over a baseline LFBE-DNN system are summarized below.*

DNN configuration	matched	multi	mismatched
Fully connected	**7.0**	-4.8	-52.2
+ Restricted connectivity	5.7	-2.6	-36.4
+ Max-pool non-linearity	4.8	**-1.3**	**-16.7**

5. Conclusions

This paper has proposed a novel HMM-GMM based i-vectors that use more accurate HMM state alignments of an ASR system. Next, it has introduced an explicit non-linear hidden layer on i-vectors to achieve better generalization of a DNN trained on concatenated LFBEs and i-vectors. Subsequently, the robustness of such a DNN AM to i-vector mismatch is improved by using causal i-vectors in training, restricting connectivity among hidden nodes and applying a max-pool non-linearity to sample hidden representations. Our experiments have clearly showed the effectiveness of these techniques for improving WER in matched condition and providing robustness in mismatched condition.

terance in sequence, there are no previous utterances leading to a zero causal i-vector. Second utterance uses first utterance, third utterances uses first two utterances and so on for obtaining causal i-vectors.

6. References

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

[2] P. Swietojanski, A. Ghoshal, and S. Renals, "Revisiting hybrid and gmm-hmm system combination techniques," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2013.

[3] S. H. K. Parthasarathi, B. Hoffmeister, S. Matsoukas, A. Mandal, N. Strom, and S. Garimella, "fmllr based feature-space speaker adaptation of dnn acoustic models." in *INTERSPEECH*, 2015.

[4] H. Liao, "Speaker adaptation of context dependent deep neural networks," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2013.

[5] N. Strom, "Speaker adaptation by modeling the speaker variation in a continuous speech recognition system," in *Proceedings of Fourth International Conference on Spoken Language (ICSLP)*, 1996.

[6] M. L. Seltzer, D. Yu, and Y. Wang, "An investigation of deep neural networks for noise robust speech recognition," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2013.

[7] G. Saon, H. Soltau, D. Nahamoo, and M. Picheny, "Speaker adaptation of neural network acoustic models using i-vectors," in *IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, 2013.

[8] A. Senior and I. Lopez-Moreno, "Improving dnn speaker independence with i-vector inputs," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2014.

[9] N. Dehak, P. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 4, pp. 788–798, 2011.

[10] P. Kenny, "Joint factor analysis of speaker and session variability: Theory and algorithms," *CRIM, Montreal,(Report) CRIM-06/08-13*, 2005.

[11] Y. Lei, N. Scheffer, L. Ferrer, and M. McLaren, "A novel scheme for speaker recognition using a phonetically-aware deep neural network," in *In Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2014.

[12] M. J. Gales, "Semi-tied covariance matrices for hidden markov models," *IEEE Transactions on Speech and Audio Processing*, vol. 7, no. 3, pp. 272–281, 1999.