Investigating Factor Analysis Features for Deep Neural Networks In Noisy Speech Recognition

Sriram Ganapathy, Samuel Thomas, Dimitrios Dimitriadis, Steven Rennie

IBM T.J. Watson Research Center, Yorktown Heights, NY, USA.
{ganapath,sthomas,dbdimitr,srennie}@us.ibm.com

Abstract

The problem of speaker and channel adaptation in deep neural network (DNN) based automatic speech recognition (ASR) systems is of substantial interest in advancing the performance of these systems. Recently, the speaker identity vectors (i-vectors) have shown improvements for ASR systems in matched conditions. In this paper, we propose the application of the general factor analysis framework for noisy speech recognition tasks. Several methods for deriving speaker and channel factors are explored including joint factor analysis (JFA) and i-vectors derived from DNN posteriors instead of the traditional Universal background model (UBM) approach. We also experiment with the late fusion of i-vector features with bottleneck (BN) features obtained from a previously trained convolutional neural network (CNN) system. The ASR experiments are performed on the Aspire challenge test data which contains noisy far-field speech while the acoustic models are trained with conversational telephone speech (CTS) data from the Fisher corpus. In these experiments, we show that the factor analysis based methods provide significant improvements in the word error rate (relative improvements of about 11% compared to the baseline DNN system trained with speaker adapted features).

Index Terms: Factor analysis, Speaker and Channel Adaptation, Deep Neural Networks, Automatic Speech Recognition

1. Introduction

Deep neural networks (DNNs) have shown promising performance for tasks like automatic speech recognition (ASR) [1, 2] and in the recent years have increasingly become the defacto method for acoustic modeling replacing the Gaussian mixture models (GMMs). In the context of GMM based ASR systems, the problem of speaker adaptation has been widely studied and transformation techniques like maximum likelihood linear regression (MLLR) have been successfully applied. However, for discriminative models like DNNs, speaker and channel adaptation from a small amount of data is not straightforward. While adaptation of a subset of parameters have been tried in the past [3], these methods require some form of regularization to avoid issues of overfitting [4].

For speaker and language recognition, the concept of identity vectors (i-vectors) is widely used for summarizing the statistics from a single recording with a fixed dimensional vector [5, 6]. Recently, the i-vectors have been explored for ASR tasks by concatenating the i-vectors along with acoustic features for training DNN models [7, 8, 9]. This approach attempts to learn the weights of the DNN in a manner which reduces the speaker variability in phoneme classification by exploiting the speaker characteristics embedded in the i-vectors. In other words, the speaker i-vector features represent the nuisance directions for the phoneme classification task and the network is trained to ignore these variabilities. In another related work [10], the authors argue that i-vector features may encompass much more than speaker specific information.

In this paper, we propose to use the i-vector approach for ASR to address channel and noise related variabilities in the speech signal in addition to the speaker variability. Joint factor analysis (JFA) [11] provides a decomposition scheme which separates the projection model into separate speaker and channel/session sub-spaces. Using this procedure, we derive speaker and channel factors which can be used with acoustic features for DNN training. To our knowledge, this is the first work using JFA framework with DNN based posteriors instead of the GMM-UBM approach [12, 13]. In this case, the mixture components correspond to phonetic classes and the GMM based posteriors used in the conventional i-vector estimation are replaced with phonetic posteriors.

The other scenario of interest here is the use of the i-vectors along with a previously trained DNN/CNN acoustic model. The goal here is to improve ASR performance with minimal retraining. We develop a scheme of using hidden layer activation outputs (BN features) from the trained DNN model with the i-vectors to train a shallow neural network.

The ASR experiments are performed on the Aspire challenge data [14] which consists of a scenario of mis-matched acoustic training and testing conditions. The acoustic models are trained on 600 hours of conversational telephone speech (CTS) from the Fisher corpus. The test data is collected in far-field microphone conditions and it includes significant room noise and reverberation. Our experiments in this task show that factor analysis features provide significant improvements in the WER compared to baseline speaker adapted acoustic features.

The rest of the paper is organized as follows. Sec. 2 describes the general factor analysis framework and highlights the different schemes used in this paper The experimental setup and the training procedure are discussed in Sec. 3. The results for various ASR evaluations are reported in Sec. 4 followed by a brief discussion. Sec. 5 provides a summary of the techniques proposed in this work.

2. Factor Analysis Framework

The techniques outlined here are derived from the previous work on joint factor analysis (JFA) and i-vectors [5, 11, 15]. We follow the notations used in [5]. The training data from all the speakers is used to train a GMM with model parameters $\lambda = \{\pi_c, \boldsymbol{\mu_c}, \boldsymbol{\Sigma_c}\}$ where π_c, $\boldsymbol{\mu}_c$ and $\boldsymbol{\Sigma}_c$ denote the mixture component weights, mean vectors and covariance matrices respectively for $c = 1, .., C$ mixture components. Here, $\boldsymbol{\mu}_c$ is a vector of dimension F and $\boldsymbol{\Sigma}_c$ is of assumed to be diagonal matrix of dimension $F \times F$.

2.1. I-vector Representations

Let $\boldsymbol{\mathcal{M}_0}$ denote the UBM supervector which is the concatenation of $\boldsymbol{\mu}_c$ for $c = 1, .., C$ and is of dimension of $CF \times 1$. Let $\boldsymbol{\Sigma}$ denote the block diagonal matrix of size $CF \times CF$ whose diagonal blocks are $\boldsymbol{\Sigma}_c$. Let $\mathcal{X}(s) = \{\boldsymbol{x}_i^s, i = 1, ..., H(s)\}$ denote the low-level feature sequence for input recording s where i denotes the frame index. Here $H(s)$ denotes the number of frames in the recording. Each $\boldsymbol{x}_i^s$ is of dimension $F \times 1$.

Let $\boldsymbol{\mathcal{M}(s)}$ denote the recording supervector which is the concatenation of speaker adapted GMM means $\boldsymbol{\mu}_c(s)$ for $c = 1, .., C$ for the speaker s. Then, the i-vector model is,

$$\boldsymbol{\mathcal{M}}(s) = \boldsymbol{\mathcal{M}_0} + \boldsymbol{V}\boldsymbol{y}(s) \tag{1}$$

where $\boldsymbol{V}$ denotes the total variability matrix of dimension $CF \times M$ and $\boldsymbol{y}(s)$ denotes the i-vector of dimension M. The i-vector is assumed to be distributed as $\mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$.

In order to estimate the i-vectors, the iterative EM algorithm is used. We begin with random initialization for the total variability matrix $\boldsymbol{V}$. Let $p_\lambda(c|\boldsymbol{x}_i^s)$ denote the alignment probability of assigning the feature vector $\boldsymbol{x}_i^s$ to mixture component c. The sufficient statistics are then computed as,

$$\begin{aligned} N_c(s) &= \sum_{i=1}^{H(s)} p_\lambda(c|\boldsymbol{x}_i^s) \\ \boldsymbol{S}_{X,c}(s) &= \sum_{i=1}^{H(s)} p_\lambda(c|\boldsymbol{x}_i^s)(\boldsymbol{x}_i^s - \boldsymbol{\mu}_c) \end{aligned} \tag{2}$$

Let $\boldsymbol{N(s)}$ denote the $CF \times CF$ block diagonal matrix with diagonal blocks $N_1(s)\boldsymbol{I}$, $N_2(s)\boldsymbol{I}$,..,$N_C(s)\boldsymbol{I}$ where $\boldsymbol{I}$ is the $F \times F$ identity matrix. Let $\boldsymbol{S}_X(s)$ denote the $CF \times 1$ vector obtained by splicing $\boldsymbol{S}_{X,1}(s)$,..,$\boldsymbol{S}_{X,C}(s)$.

It can be easily shown [5] that the posterior distribution of the i-vector $p_\lambda(\boldsymbol{y}(s)|\mathcal{X}(s))$ is Gaussian with covariance $\boldsymbol{l}^{-1}(s)$ and mean $\boldsymbol{l}^{-1}(s)\boldsymbol{V}^*\boldsymbol{\Sigma}^{-1}\boldsymbol{S}_X(s)$, where

$$\boldsymbol{l}(s) = \boldsymbol{I} + \boldsymbol{V}^*\boldsymbol{\Sigma}^{-1}\boldsymbol{N}(s)\boldsymbol{V} \tag{3}$$

The optimal estimate for the i-vector $\boldsymbol{y}(s)$ obtained as $argmax_y\big[p_\lambda(\boldsymbol{y}(s)|\mathcal{X}(s))\big]$ is given by the mean of the posterior distribution.

For re-estimating the $\boldsymbol{V}$ matrix, the maximization of the expected value of the log-likelihood function (EM algorithm), gives the following relation [5],

$$\sum_{s=1}^{S} \boldsymbol{N}(s)\ \boldsymbol{V}\ \mathrm{E}\big[\boldsymbol{y}(s)\boldsymbol{y}^*(s)\big] = \sum_{s=1}^{S} \boldsymbol{S}_X(s)\mathrm{E}\big[\boldsymbol{y}^*(s)\big] \tag{4}$$

where E[.] denotes the posterior expectation operator. The solution for Eq. (4) can be computed for each row of $\boldsymbol{V}$. Thus, the i-vector estimation is performed by iterating between the estimation of posterior distribution and the update of the total variability matrix (Eq. (4)).

2.2. Joint Factor Analysis

The JFA approach attempts to capture the additional channel factors that represent intraspeaker variability [11]. These factors represent the variability in the recording environment for different segments from the same speaker. For this case, we assume that for speaker s, there are $q = 1, .., Q(s)$ sessions, each with $H_q(s)$ frames. The JFA model is

$$\begin{aligned} \boldsymbol{\mathcal{M}}(s) &= \boldsymbol{\mathcal{M}_0} + \boldsymbol{V}\boldsymbol{y}(s) + \boldsymbol{D}\boldsymbol{z}(s), \\ \boldsymbol{\mathcal{M}}_q(s) &= \boldsymbol{\mathcal{M}}(s) + \boldsymbol{U}\boldsymbol{x}_q(s), \end{aligned} \tag{5}$$

where $\boldsymbol{V}$ denotes the speaker variability matrix of size $CF \times M$, $\boldsymbol{U}$ denotes the channel/session variability matrix of size $CF \times N$ and $\boldsymbol{D}$ is a diagonal matrix of size $CF \times CF$ capturing the residual space. Here, $\boldsymbol{\mathcal{M}}(s)$ and $\boldsymbol{\mathcal{M}}_q(s)$ represent supervectors for the entire data from speaker s and for the session q from speaker s respectively. The factors $\boldsymbol{y}(s)$, $\boldsymbol{x}_q(s)$ and $\boldsymbol{z}(s)$ are speaker factors, channel factors and residual factors of dimension M, N and CF respectively. The sub-space $\boldsymbol{V}\boldsymbol{V}^*$ captures the interspeaker variability while the sub-space $\boldsymbol{U}\boldsymbol{U}^*$ captures the intraspeaker channel variability.

In order to estimate the parameters in the JFA model, let $\underline{\boldsymbol{Y}}(s)$ denote the collection of factors for each speaker s. $\underline{\boldsymbol{Y}}(s) = [\boldsymbol{x}_1^*(s)\ \boldsymbol{x}_2^*(s)\ ...\ \boldsymbol{x}_{Q(s)}^*(s)\ \boldsymbol{y}^*(s)\ \boldsymbol{z}^*(s)]^*$. Also, let

$$\underline{\boldsymbol{V}} = \begin{bmatrix} \boldsymbol{U} & & & \boldsymbol{V} & \boldsymbol{D} \\ & \ddots & & \vdots & \vdots \\ & & \boldsymbol{U} & \boldsymbol{V} & \boldsymbol{D} \end{bmatrix} \tag{6}$$

where $\underline{\boldsymbol{V}}$ is of dimension $[Q(s)CF \times (Q(s)N + M + CF)]$. If we also have $\underline{\boldsymbol{\mathcal{M}}}(s)$ as the concatenation of all $\boldsymbol{\mathcal{M}}_q(s)$ for $q = 1, .., Q(s)$ and $\underline{\boldsymbol{\mathcal{M}}}_0$ as the concatenation of the same vector $\boldsymbol{\mathcal{M}}_0$ $Q(s)$ times, then we can rewrite Eq. (5) as

$$\underline{\boldsymbol{\mathcal{M}}}(s) = \underline{\boldsymbol{\mathcal{M}}}_0 + \underline{\boldsymbol{V}}\ \underline{\boldsymbol{Y}}(s) \tag{7}$$

which is similar to Eq. (1). Thus, the parameters of the JFA model can be computed in a very similar fashion to the EM formulation described in Sec. 2.1. In the ASR experiments, we group together speech segments from a speaker so as to form at least 5 sessions per speaker. In our experiments, we use $M = 150$ and $N = 150$. For each speech utterance, these features (one feature per speaker) are replicated to match the frame length of the acoustic features for the utterance and appended at the input of the DNN/CNN acoustic model.

2.3. DNN i-vectors

Instead of using a GMM-UBM based computation of i-vectors, we can also use DNN context dependent state (senone) posteriors to generate the sufficient statistics used in the i-vector computation [12, 13]. The GMM mixture components will be replaced with the senone classes present at the output of the DNN. Specifically, $p_\lambda(c|\boldsymbol{x}_i^s)$ used in Eq. (2) is replaced with the DNN posterior probability estimate of the senone c given the input acoustic feature vector $\boldsymbol{x}_i^s$ and the number of senones is the parameter C. The other parameters of the UBM model $\lambda = \{\pi_c, \boldsymbol{\mu_c}, \boldsymbol{\Sigma_c}\}$ are computed as

$$\begin{aligned} \pi_c &= \frac{\sum_{s=1}^{S}\sum_{i=1}^{H(s)} p(c|\boldsymbol{x}_i^s)}{\sum_{c=1}^{C}\sum_{s=1}^{S}\sum_{i=1}^{H(s)} p(c|\boldsymbol{x}_i^s)} \\ \boldsymbol{\mu}_c &= \frac{\sum_{s=1}^{S}\sum_{i=1}^{H(s)} p(c|\boldsymbol{x}_i^s)\boldsymbol{x}_i^s}{\sum_{c=1}^{C}\sum_{s=1}^{S}\sum_{i=1}^{H(s)} p(c|\boldsymbol{x}_i^s)} \\ \boldsymbol{\Sigma}_c &= \frac{\sum_{s=1}^{S}\sum_{i=1}^{H(s)} p(c|\boldsymbol{x}_i^s)(\boldsymbol{x}_i^s - \boldsymbol{\mu}_c)(\boldsymbol{x}_i^s - \boldsymbol{\mu}_c)^*}{\sum_{c=1}^{C}\sum_{s=1}^{S}\sum_{i=1}^{H(s)} p(c|\boldsymbol{x}_i^s)} \end{aligned} \tag{8}$$

Using these estimates for the UBM parameters, the rest of the i-vector formulation discussed in Sec. 2.1 is followed to derive the DNN i-vectors. For the DNN i-vectors, we use a reduced set of senones (1088 obtained by merging the 10000 triphone states using a decision tree).

3. Experimental setup

3.1. ASR System

The ASR system is similar to the setup described in [16]. The first step in the acoustic modeling involved the training of traditional HMM-GMM based acoustic models. The GMM models are trained on 13 dimensional PLP features estimated in 25 ms windows of speech. The cepstral features from 9 consecutive frames are then spliced after speaker based cepstral mean-variance and vocal tract length normalizations (VTLN). A LDA transform is applied to reduce the final feature dimensionality to 40. The ML training of the GMM models is also interleaved with the estimation of a global semi-tied covariance (STC) transform. Speaker-space feature maximum likelihood regression (FMLLR) is finally applied to train speaker adapted models. The training is done with 900 hours of speech from the Fisher corpus [17]. The Aspire test data [14] contains 30 recordings each of duration 10 minutes long amounting to 5 hours of test data.

3.1.1. Deep Neural Network Models

The DNN models are fully connected multilayer perceptrons with several non-linear hidden layers that are discriminatively trained to estimate posterior probabilities of context-dependent states. Using the standard error back-propagation and cross-entropy objective function, the DNNs are trained on speaker adapted FMLLR features using alignments produced from the HMM-GMM acoustic model described earlier. The DNNs are pretrained by growing them layer-wise to 7 hidden layers. Except for the penultimate bottleneck (BN) layer with 512 units all the other hidden layers have 2048 units. In all the experiments reported in this paper, the DNNs are trained on 600 hours of audio data from the Fisher corpus.

3.1.2. Convolutional Neural Network Models

Convolutional neural networks (CNN) [18] use additional feature extracting layers based on $2-D$ convolution before a DNN. We train CNN models on 40 dimensional log-mel spectra augmented with Δ and $\Delta\Delta$s. Each frame of speech is also appended temporally with a fixed set of 11 frames. All of the 128 nodes in the first feature extracting layer are attached with 9×9 filters while the second feature extracting layer with 256 nodes has a similar set of 4×3 filters. The non-linear outputs from the second feature extracting layer are then passed onto the following DNN layers.

3.1.3. Language models

The ASR system uses a 4-gram model containing 18M n-grams derived from the entire Fisher training corpus.

3.1.4. Late-fusion Acoustic Models

These networks are much shallower networks with 4 hidden layers with 1024 units each. The input to these networks are 512 dimensional BN features from DNN/CNN models, concatenated with i-vector features.

3.2. Denoising

The training and testing sets of the Aspire task [14] are highly mismatched. While the training data is derived from conversational telephone speech (CTS), the test data is recorded in noisy conditions using a far-field microphone. Thus, the test data contains significant amounts of noise and reverberation artifacts. In order to decrease the effects of these two types of distortions, we first suppress the additive noise using a variation of the MMSE algorithm [19]. Then, we subtract the late reverberation component of the signal employing the MSLP ("long-term Multi-Step Linear Prediction") algorithm [20]. The denoising process is applied only on the test set, while the audio of the training set is left unprocessed.

Table 1: Performance in terms of word error rate (WER %) for the baseline ASR system trained with 600 hours of CTS data and tested on the Aspire challenge data. The fmllr−v2 stands for two pass fmllr using transcripts from first pass fmllr.

System	WER (%)
Logmel feat.	51.2
vtln+lda+fmllr	47.6
vtln+lda+fmllr + Denoising	43.7
vtln+lda+fmllr−v2 + Denoising (baseline)	**43.2**

Table 2: Performance in terms of word error rate (WER %) for different variants of plp based i-vectors.

System	WER (%)
Baseline	43.2
+ ivec-plpfmllr-unorm	41.0
+ ivec-plp2048	41.3
+ ivec-plp4096	40.8
+ ivec-plp1024-cmvn	40.1
+ ivec-plp1024-vtln-lda-fmllr	39.5
+ ivec-unorm-plp1024-vtln-lda-fmllr	**38.9**

4. Results

4.1. Baseline System

We explore the usefulness of speaker specific (VTLN-LDA-FMLLR) transforms for the Aspire data as well as the benefits of denoising the test data. These results are reported in Table 1. In these experiments the DNN input layer is of dimension 360 (9 frame of 40 dimensional features). As shown here, the application of speaker transforms provides significant improvements over the log Mel features. The denoising procedure described in Sec. 3 gives further improvements of about 9 % relative compared to the speaker specific features. The last row of this table (FMLLR-v2) corresponds to scenario of retraining the FMLLR transform using the lattice generated from the first pass speaker specific FMLLR features with denoising. This system will be used as the baseline for investigating the usefulness of factor analysis features.

4.2. I-vector variants

The next set of experiments compare the different variants of GMM-UBM based i-vectors (Sec. 2.1). These results reported in Table 2 compare the performance of i-vectors obtained from different number of Gaussian mixture components (namely 1024,2048 and 4096) which were trained using 39 dimensional PLP cepstral coefficients with delta and acceleration coefficients. The i-vectors for all these experiments are of $M = 150$ dimensions, which would make the DNN input layer of 510 dimensions. As seen here, the i-vector features improve the baseline ASR performance for all the cases considered here. The results suggest that increasing the number of Gaussians has a relatively minor effect on performance of

Table 3: Performance in terms of word error rate (WER %) for various factor analysis features.

System	WER (%)
Baseline	43.2
+ ivec-plp-fmllr-unorm	38.9
+ ivec-dnn-fmllr-unorm	38.8
+ jfa-dnn-fmllr-unorm	**38.6**

Table 4: Performance in terms of word error rate (WER %) for various late-fusion experiments

System	WER (%)
Baseline DNN	43.2
DNN-BN + ivec-plp1024	42.7
CNN-logmel-vtln	44.0
CNN-logmel-vtln-stc-fmllr	43.0
CNN-logmel-vtln-stc-fmllr-BN + ivec-fdlp1024	**40.4**

the system. Although the i-vectors with 4096 Gaussian mixture components is slightly better than those with 1024 components, the computational burden is significantly high in training the i-vector model as well as the extraction of these features for the test data. The last two rows of Table 2 report the performance of transformations applied to the PLP coefficients before GMM-UBM and i-vector training. The use of variance normalized (CVN) features improve the performance by about 0.9 % in absolute WER. We observe additional gains by using speaker transformed features (VTLN-LDA-FMLLR) even in the i-vector training process. The speaker transformed features used in i-vector extraction provide absolute WER improvements of about 1.5 % over the i-vector based system without any normalization. Further, the unit length normalization [21] of i-vectors also provides additional improvements in the ASR performance and provides a relative improvement of 10 % over the baseline system.

4.3. General Factor Analysis Features

The experiments reported in Table 3 compare the performance of i-vector features with other factor analysis features namely the JFA method (Sec. 2.2) and the DNN i-vector method (Sec. 2.3). The DNN i-vectors provide similar results compared to GMM-UBM based i-vectors. The JFA method of modeling intra-speaker variability provides a relatively moderate improvement in the ASR performance. The overall improvement of the JFA framework with the DNN based statistics is about 11% relative to the baseline.

4.4. Late Fusion

The final set of experiments reported in Table 4 explore the performance of late-fusion approaches where a previously trained DNN without i-vectors is used to generate hidden layer activations. The BN activations are used in conjunction with i-vectors to train a shallow NN. The results indicate that while the early fusion approaches are more beneficial (Table 2), the late fusion techniques improve the performance of a previously trained DNN with a relatively minor computational effort in retraining.

The late fusion approaches can also be used to improve the adaptation performance of convolutional neural network (CNN) based ASR systems, as shown in Table 4. Here, a baseline CNN system is improved by feature adaptation using dynamic noise adaptation with clean Detection (DNA-CD) [22] followed by FMLLR in Mel-semi-tied-covariance (Mel-STC) space [23]. As described in [23], the (speaker-dependent) FMLLR transformation F estimated in the STC space must be multiplied by inverse of the STC matrix to reconstruct adapted log Mel features of each frame. This strategy allows one to utilize a diagonal covariance GMM as a basis for adapting the highly correlated log Mel features in an unsupervised manner, while maintaining the CNN's ability to exploit correlation patterns seen during training. However, this adaptation approach is single-frame-based, has limited adaptation capacity. The adapted CNN system is significantly outperformed by the corresponding late-fused system, which incorporates 100 dimensional i-vectors derived from frequency domain linear prediction (FDLP) features [24]. The i-vectors express a higher dimensional synopsis of the acoustic mismatch and provide relevant summary of the entire recording. Thus, the inclusion of i-vectors to the feature adapted CNN acoustic model is highly effective, even when they are fused after several layers of non-linearities.

4.5. Discussion

The various ASR experiments reported in this section indicate that factor analysis features provide useful information for ASR tasks. The approach of using all the segments from the same speaker to generate a single i-vector (similar to the one proposed in [7]) is slightly inferior to the approach using session level factor analysis features (JFA). The application of speaker normalization transforms like VTLN and FMLLR to acoustic features are beneficial even for the i-vector extraction. The i-vectors based on these transformed features improve the ASR performance and further underline the questions addressed in [10]. Specifically, the i-vectors based on these transformed features have low speaker specific information but however improve the ASR performance. This would mean that the i-vectors normalize other variabilities in the speech signal beyond speaker.

The use of DNN based JFA features improves the ASR performance compared to GMM-UBM based i-vectors. The use of i-vectors in late fusion scenario enables the application in CNN based ASR systems and it improves the performance of a previously trained ASR system with minimal retraining effort. The next logical step in this pipeline is to combine all individual approaches - training DNN i-vectors on speaker transformed diverse acoustic frontend. Furthermore, there is a need for more scientific and experimentation analysis to explore the information conveyed by i-vectors for DNN acoustic models.

5. Summary

In this paper, we have analyzed the use of factor analysis features for ASR tasks in noisy speech. The various factor analysis schemes explored in this work include - conventional i-vectors, joint factor analysis and DNN based i-vectors. Several ASR experiments using these features indicate that the factor analysis features improves the performance of ASR systems by a considerable margin.

6. Acknowledgements

The authors would like to thank Brian Kingsbury, Hong-Kwang Kuo and Lidia Mangu for their help in building the ASR system and Jason Pelecanos for his help with the i-vector setup.

7. References

[1] G. Hinton and et al., "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *Signal Processing Magazine, IEEE*, vol. 29, no. 6, pp. 82–97, 2012.

[2] George E Dahl, Dong Yu, Li Deng, and Alex Acero, "Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 1, pp. 30–42, 2012.

[3] Kaisheng Yao, Dong Yu, Frank Seide, Hang Su, Li Deng, and Yifan Gong, "Adaptation of context-dependent deep neural networks for automatic speech recognition.," in *SLT*, 2012, pp. 366–369.

[4] Dong Yu, Kaisheng Yao, Hang Su, Gang Li, and Frank Seide, "KL-divergence regularized deep neural network adaptation for improved large vocabulary speech recognition," in *Proc of ICASSP*. IEEE, 2013, pp. 7893–7897.

[5] Patrick Kenny, Gilles Boulianne, and Pierre Dumouchel, "Eigenvoice modeling with sparse training data," *IEEE Transactions on Speech and Audio Processing*, vol. 13, no. 3, pp. 345–354, 2005.

[6] Najim Dehak, Pedro A Torres-Carrasquillo, Douglas A Reynolds, and Reda Dehak, "Language recognition via i-vectors and dimensionality reduction.," in *Proc of INTERSPEECH*, 2011, pp. 857–860.

[7] George Saon, Hagen Soltau, David Nahamoo, and Michael Picheny, "Speaker adaptation of neural network acoustic models using i-vectors," in *IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, 2013, pp. 55–59.

[8] Andrew Senior and Ignacio Lopez-Moreno, "Improving DNN speaker independence with i-vector inputs," in *Proc. of ICASSP*, 2014.

[9] Penny Karanasou, Yongqiang Wang, Mark JF Gales, and Philip C Woodland, "Adaptation of deep neural network acoustic models using factorised i-vectors," in *Proc. of INTERSPEECH*, 2014.

[10] Mickael Rouvier and Benoit Favre, "Speaker adaptation of DNN-based asr with i-vectors: Does it actually adapt models to speakers?," in *Proc. of INTERSPEECH*, 2014.

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

[12] Mitchell McLaren, Yun Lei, Nicolas Scheffer, and Luciana Ferrer, "Application of convolutional neural networks to speaker recognition in noisy conditions," in *Proc. of INTERSPEECH*, 2014.

[13] Yun Lei, Luciana Ferrer, Aaron Lawson, Mitchell McLaren, and Nicolas Scheffer, "Application of convolutional neural networks to language identification in noisy conditions," in *Proc. Speaker Odyssey Workshop*, 2014.

[14] "The IARPA ASPIRE challenge," in *http://www.iarpa.gov/index.php/working-with-iarpa/prize-challenges/306-automatic-speech-in-reverberant-environments-aspire-challenge/*.

[15] Najim Dehak, Patrick Kenny, Réda Dehak, Pierre Dumouchel, and Pierre Ouellet, "Front-end factor analysis for speaker verification," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 4, pp. 788–798, 2011.

[16] H. Soltau, H.K. Kuo, L. Mangu, G. Saon, and T. Beran, "Neural Network Acoustic Models for the DARPA RATS Program," in *Proc. of INTERSPEECH*, 2013.

[17] C. Cieri, D. Miller, and K. Walker, "The Fisher Corpus: a Resource for the Next Generations of Speech-to-Text," in *LREC*, 2004.

[18] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient based Learning applied to Document Recognition," *Proceedings of the IEEE*, 1998.

[19] J. S. Erkelens and R. Heusdens, "Tracking of nonstationary noise based on data-driven recursive noise power estimation," *IEEE Trans. on Audio, Speech and Language Process.*, vol. 16, no. 6, pp. 1112–1123, Aug. 2008.

[20] K. Kinoshita, M. Delcroix, T. Nakatani, and M. Miyoshi, "Suppression of Late Reverberation Effect on Speech Signal Using Long-Term Multiple-step Linear Prediction," *IEEE Trans. on Audio, Speech and Language Process.*, vol. 17, no. 4, May 2000.

[21] Daniel Garcia-Romero and Carol Y Espy-Wilson, "Analysis of i-vector length normalization in speaker recognition systems.," .

[22] Steven J Rennie, Pierre L Dognin, and Petr Fousek, "Matched-condition robust dynamic noise adaptation," in *IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*. IEEE, 2011, pp. 137–140.

[23] Tara N Sainath and et al., "Improvements to deep convolutional neural networks for lvcsr," in *IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, 2013, pp. 315–320.

[24] S. Thomas, S. Ganapathy, and H. Hermansky, "Recognition of reverberant speech using frequency domain linear prediction," *IEEE Signal Processing Letters*, vol. 15, pp. 681–684, 2008.