INTERSPEECH 2016
September 8–12, 2016, San Francisco, USA

Semi-Supervised Speaker Adaptation for In-Vehicle Speech Recognition with Deep Neural Networks

Wonkyum Lee[1], *Kyu J. Han*[2], *Ian Lane*[1]

[1]Carnegie Mellon University, 5000 Forbes Avenue, Pittsburgh, PA 15213
[2]Ford Motor Company, 3200 Hillview Avenue, Palo Alto, CA 94304
{wonkyuml,lane}@cmu.edu, khan5@ford.com

Abstract

In this paper, we present a new i-vector based speaker adaptation method for automatic speech recognition with deep neural networks, focusing on in-vehicle scenarios. Our proposed method is, rather than augmenting i-vectors to acoustic feature vectors to form concatenated input vectors for adapting neural network acoustic model parameters, is to perform feature-space transformation with smaller *transformation neural networks* dedicated to acoustic feature vectors and i-vectors, respectively, followed by a layer of *linear combination* of the network outputs. This feature-space transformation is learned via semi-supervised learning without any parameter change in the original deep neural network acoustic model. Experimental results show that our proposed method achieves 18.3% relative improvement in terms of word error rate compared to the speaker independent performance, and verify that it has a potential to replace well-known feature-space Maximum Likelihood Linear Regression (fMLLR) in in-vehicle speech recognition with deep neural networks.

Index Terms: Speaker adaptation, deep neural network, i-vector, linear combination layer, semi-supervised learning

1. Introduction

Deep Neural Networks (DNNs) have taken over the dominance of Gaussian Mixture Models (GMMs) in a Hidden Markov Model (HMM) framework for Automatic Speech Recognition (ASR), since deep learning based neural networks showed significant improvements in modeling context-dependent phonetic events [1]. More recently, there are research activities to replace even the HMM framework with recurrent neural networks, moving towards an end-to-end neural network system pipeline for ASR [2, 3, 4]. This paradigm shift is not only due to continuing development in the research field of neural networks, e.g., overcoming vanishing gradient problems during multi-layer network training [5, 6, 7], but thanks to the recent advent of powerful parallel computing architectures that can efficiently handle vast amount of data to train large size DNNs.

Speaker adaptation is critical to achieve consistent ASR performance across different speakers. The acoustic feature vectors captured from spoken utterances, e.g., Mel-Frequency Cepstral Coefficient (MFCC), contain not only phonetic contents but also idiosyncratic attributes, such as speaker-specific traits caused by gender, dialect or nativeness. In general, acoustic models for ASR are delicate to such speaker-dependent variations. For this reason, there have been a significant number of research efforts on speaker adaptation that minimizes the influence of speaker-dependent variations on ASR performance.

In DNN-HMM based ASR systems, i-vectors [8] have gained popularity for speaker adaptation. They are a sub-space representation for Gaussian Super Vectors (GSVs) [9], and are extracted by a simplified joint factor analysis that projects GSVs to a total variability sub-space. In a DNN-HMM based ASR framework, i-vectors are augmented to acoustic feature vectors to form concatenated input vectors for a DNN [10, 11]. Due to i-vectors' speaker-specific information, the parameters of the DNN acoustic model are adapted to be insensitive to speaker-dependent variations.

In this paper, we propose a new i-vector based speaker adaptation method for a DNN-HMM based ASR system, focusing on *in-vehicle* ASR. There are two practical restrictions that we considered for our proposal; 1) the DNN acoustic model of an in-vehicle ASR system is not directly updated or entirely retrained through the speaker adaptation procedure, and 2) the in-vehicle ASR system should work in a hybrid fashion, regardless of whether i-vectors are available for vehicle drivers[1]. The proposed method, rather than adapting the DNN parameters, is to transform input feature vectors for the DNN acoustic model to a new feature space where speaker-dependent variations are normalized, like feature-space Maximum Likelihood Linear Regression (fMLLR) [12] does for GMM-based acoustic models as compared to normal MLLR for adapting GMM means and variances. This feature-space transformation is performed through two smaller *transformation neural networks* for acoustic feature vectors and i-vectors, respectively, followed by a layer of *linear combination* of the network outputs prior to the DNN acoustic model. The transformed feature vectors after the linear combination layer have the same dimension of the original feature vectors, suitable for the DNN acoustic model to work in a hybrid fashion. Conventional i-vector augmentation for speaker adaptation in DNN-HMM based ASR, e.g., [10, 11], assumes the availability of i-vectors, and in a case where i-vector is unavailable for a certain driver (or speaker) it would not be working. All the parameters of the transformation neural networks as well as the linear combination weights are trained via semi-supervised learning [13] where the first-path decoding hypotheses of adaptation data are used as references to compute error signals for back-propagation. The error signals do not affect the parameters of the DNN acoustic model but just by-pass them, only updating the parameters relating to feature-space transformation.

[1]One might argue that i-vectors are always available since they can be generated on the fly per frame for given acoustic feature vectors, but such frame-level i-vectors are limited in terms of representing speaker-specific information. In this paper, i-vector implies a voice profile being extracted from a *collection* of spoken data for a given speaker. In practice, enough data to generate reliable i-vectors is not guaranteed for some speakers.

http://dx.doi.org/10.21437/Interspeech.2016-1625

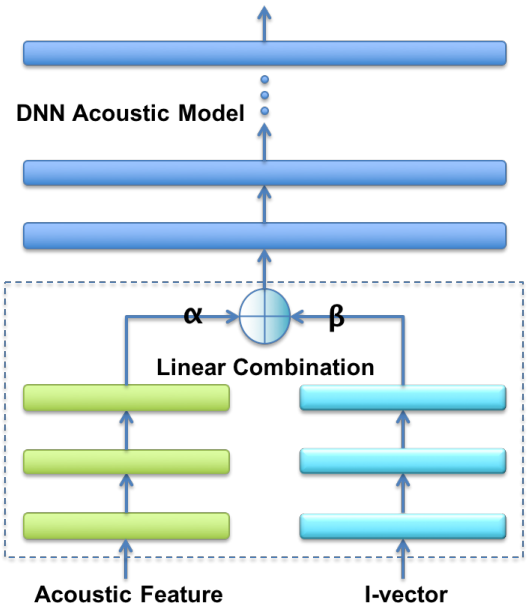

Figure 1: Proposed system architecture for transformation neural network based speaker adaptation (inside the dotted box) in a DNN-HMM ASR framework. The parameters of the two transformation neural networks (bottom left and right) as well as α and β are trained via semi-supervised learning while the parameters of the DNN acoustic model are not updated during back propagation.

This paper is organized as follows. In Section 2, we describe our proposed speaker adaptation method in more detail. The data set used for system evaluation, which is Ford Motor Company's proprietary corpus of noisy voice commands/utterances recorded in Ford-branded vehicles in various ambient noise scenarios, is presented in Section 3, along with experimental results and discussions. We summarize our findings and present future directions in Section 4.

2. Feature-Space Transformation for Speaker Adaptation

Figure 1 shows an illustrative system architecture for our proposed feature-space transformation method (inside the dotted box, in particular) using two transformation neural networks prior to the original DNN acoustic model. The transformation neural networks are relatively smaller networks as compared to the DNN acoustic model, with the purpose of transforming acoustic feature vectors and i-vectors to a new feature space where speaker-dependent variations are normalized. This approach is analogous to constrained MLLR or fMLLR [12] in a GMM-HMM based ASR framework. Consider $\bar{\mathbf{x}}$ is a per-frame acoustic feature vector. fMLLR finds an affine transform to project $\bar{\mathbf{x}}$ onto a new feature space,

$$\bar{\mathbf{x}}_a^{\text{GMM}} = \mathbf{A}\bar{\mathbf{x}} + \bar{\mathbf{b}} \tag{1}$$

where $\mathbf{A}$ and $\bar{\mathbf{b}}$ are optimized with the following loss function $\mathcal{L}$:

$$\log \mathcal{L}(\bar{\mathbf{x}}|\bar{\mu}, \mathbf{\Sigma}, \mathbf{A}, \bar{\mathbf{b}}) = \log \mathcal{N}(\mathbf{A}\bar{\mathbf{x}} + \bar{\mathbf{b}}; \bar{\mu}, \mathbf{\Sigma})) + \frac{1}{2}\log|\mathbf{A}|^2. \tag{2}$$

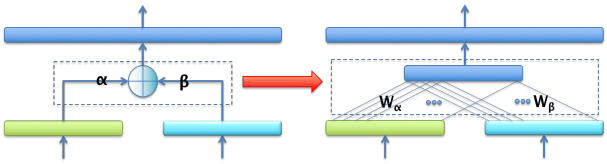

Figure 2: Detailed illustration of a layer of linear combination in the proposed system architecture. $\mathbf{W}_\alpha$ and $\mathbf{W}_\beta$ are the weight matrices having α and β for $W_{\alpha,ii}$ and $W_{\beta,ii}$, respectively, and 0 for $W_{\alpha,ij}$ and $W_{\beta,ij}$, where $i \neq j$.

Here, $\bar{\mu}$ and $\mathbf{\Sigma}$ are the mean vector and the covariance matrix of one Gaussian distribution $\mathcal{N}$ of the GMM acoustic model. The proposed method replaces fMLLR's affine transform. Note that we consider in-vehicle ASR with DNNs, where fMLLR is not available in practice. In in-vehicle ASR scenarios, the DNN acoustic model of an ASR engine built in a car has no ability to support fMLLR, which require GMMs instead. Given that we focus on a DNN-HMM based ASR framework, it would be natural to think of alternatives to fMLLR. Our practical solution is to do function approximation using two transformation neural networks rather than finding a linear transform. Let's consider transformation neural networks with N layers with an activation function σ_n for each layer n. The transformed feature vectors using acoustic feature vectors $\bar{\mathbf{x}}$ and i-vectors $\bar{\mathbf{i}}$ from the proposed method would be

$$\begin{aligned}\bar{\mathbf{x}}_a^{\text{DNN}} = \quad & \alpha \cdot \sigma_N(\mathbf{W}_N^l \bar{\mathbf{a}}_{N-1}^l + \bar{\mathbf{b}}_N^l) + \\ & \beta \cdot \sigma_N(\mathbf{W}_N^r \bar{\mathbf{a}}_{N-1}^r + \bar{\mathbf{b}}_N^r)\end{aligned} \tag{3}$$

where l and r mean the parameters belonging to the transformation neural network for acoustic feature vectors (left) or i-vectors (right) in Fig. 1, respectively. $\mathbf{W}_N$ is the weight matrix in the N^{th}-layer while $\mathbf{b}_N$ is the bias vector. In addition,

$$\bar{\mathbf{a}}_n = \sigma_n(\mathbf{W}_n \bar{\mathbf{a}}_{n-1} + \bar{\mathbf{b}}_n) \tag{4}$$

where $\bar{\mathbf{a}}_1^l = \sigma_1(\mathbf{W}_1^l \bar{\mathbf{x}} + \bar{\mathbf{b}}_1^l)$ and $\bar{\mathbf{a}}_1^r = \sigma_1(\mathbf{W}_1^r \bar{\mathbf{i}} + \bar{\mathbf{b}}_1^r)$. α and β as well as the other neural network parameters ($\mathbf{W}$ and $\bar{\mathbf{b}}$) are trained with the objective function of cross entropy via semi-supervised learning [13], where the first-path decoding hypotheses of adaptation data are used as references to compute error signals for back-propagation. Thanks to this semi-supervised training strategy, our proposed system does not require manual transcriptions for adaptation data, which is another practical advantage for in-vehicle ASR.

In in-vehicle ASR scenarios, it is not feasible to update the parameters in the DNN acoustic model of a built-in ASR engine in a car. The error signal vector (or error terms) $\bar{\mathbf{d}}_M$ generated by cross entropy criterion at the output layer of the DNN acoustic model is hence just by-passed down to the linear combination layer, where α and β are updated as follows:

$$\alpha := \alpha - \lambda(\tfrac{1}{k}\Delta\alpha) = \alpha - \lambda(\tfrac{1}{k}\Delta W_{\alpha,11}) \tag{5}$$

$$\beta := \beta - \lambda(\tfrac{1}{k}\Delta\beta) = \beta - \lambda(\tfrac{1}{k}\Delta W_{\beta,11}). \tag{6}$$

λ is a learning rate while k is a mini-batch size, and $\Delta W_{\alpha,ij}$ and $\Delta W_{\beta,ij}$ are the elements at the i^{th} row and the j^{th} column of $\Delta\mathbf{W}_\alpha$ and $\Delta\mathbf{W}_\beta$, respectively. To compute $\Delta\mathbf{W}_\alpha$ and $\Delta\mathbf{W}_\beta$, let us first refer to Figure 2, which shows a detailed illustration of the linear combination layer in terms of $\mathbf{W}_\alpha$ and $\mathbf{W}_\beta$, where

$$W_{\alpha,ij}\ (\text{or } W_{\beta,ij}) = \begin{cases} \alpha\ (\text{or } \beta\,), & i = j \\ 0, & i \neq j \end{cases}. \tag{7}$$

Using $\mathbf{W}_\alpha$ and $\mathbf{W}_\beta$, Eq. (3) is re-written as

$$\begin{aligned}\bar{\mathbf{x}}_a^{\text{DNN}} = \bar{\mathbf{x}}^{\text{AM}} &= \mathbf{W}_\alpha \cdot \sigma_N(\mathbf{W}_N^l \bar{\mathbf{a}}_{N-1}^l + \bar{\mathbf{b}}_N^l) + \\ &\quad \mathbf{W}_\beta \cdot \sigma_N(\mathbf{W}_N^r \bar{\mathbf{a}}_{N-1}^r + \bar{\mathbf{b}}_N^r) \quad (8)\\ &= \mathbf{W}_\alpha \cdot \sigma_N(\bar{\mathbf{z}}_N^l) + \mathbf{W}_\beta \cdot \sigma_N(\bar{\mathbf{z}}_N^r) \quad (9)\end{aligned}$$

where $\bar{\mathbf{x}}^{\text{AM}}$ is the input to the DNN Acoustic Model (AM). As a result, we can write

$$\bar{\mathbf{a}}_m^{\text{AM}} = \sigma_n^{\text{AM}}(\mathbf{W}_m^{\text{AM}} \bar{\mathbf{a}}_{m-1}^{\text{AM}} + \bar{\mathbf{b}}_m^{\text{AM}}) = \sigma_n^{\text{AM}}(\bar{\mathbf{z}}_m^{\text{AM}}) \quad (10)$$

where the DNN acoustic model has M layers and $\bar{\mathbf{a}}_1^{\text{AM}} = \sigma_1^{\text{AM}}(\mathbf{W}_1^{\text{AM}} \bar{\mathbf{x}}^{\text{AM}} + \bar{\mathbf{b}}_1^{\text{AM}})$. In a back-propagation mode, the error signal vector $\bar{\mathbf{d}}_M$ at the output layer of the DNN acoustic model is propagated as follows

$$\bar{\mathbf{d}}_m = ((\mathbf{W}_m^{\text{AM}})^T \bar{\mathbf{d}}_{m+1}) \bullet \frac{\partial}{\partial \bar{\mathbf{z}}_m^{\text{AM}}} \bar{\mathbf{a}}_m^{\text{AM}}, \quad (11)$$

where $\bullet$ is an element-wise dot product, coming down to $\bar{\mathbf{d}}_1$, error signal vector at the linear combination layer. Thus,

$$\Delta \mathbf{W}_\alpha = \bar{\mathbf{d}}_1 \cdot (\bar{\mathbf{a}}_{N-1}^l)^T. \quad (12)$$

Similarly,

$$\Delta \mathbf{W}_\beta = \bar{\mathbf{d}}_1 \cdot (\bar{\mathbf{a}}_{N-1}^r)^T. \quad (13)$$

The parameters of the transformation neural networks are also computed in a similar way as error signal vectors keep being propagated down through the two networks.

One interesting aspect of the proposed method is a linear combination of the outputs of the two transformation neural networks to keep the dimension of the transformed vectors unchanged for the DNN acoustic model. In other words, $|\bar{\mathbf{x}}_a^{\text{DNN}}| = |\bar{\mathbf{x}}|$ where $|\cdot|$ is the cardinality of a vector. This is to accommodate the practical need that in-vehicle ASR with i-vector based speaker adaptation should work in a hybrid fashion, regardless of whether i-vectors are available or not for drivers (or speakers). The similar approaches to utilize a linear combination for network outputs prior to another DNN can be found in [14, 15], which however are not applicable to in-vehicle ASR scenarios.

3. Experiments and Discussions

3.1. Data

The main data set we used in our experiments is Ford Motor Company's noisy data corpus (2K vocabulary and 20 hours in total) collected in actual driving conditions in Ford-branded vehicles. The utterances were recorded in reverberate and noisy vehicle cabins of varying body styles[2] and different ambient noise conditions[3]. We partitioned this data set into two sets with non-overlapping speakers. The training set contains 15,610 utterances from 82 speakers and the evaluation set contains 1,573 utterances from 8 speakers. We directly used each audio signal without any special noise suppression techniques.

[2]They are categorized to small, medium, large car, SUV and pickup truck.

[3]They are blower on/off, road surface, rough/smooth, 0-65 MPH speed, windshield wipers on/off, windows open/closed, etc.

Table 1: Baseline system performances for both WSJ and Ford's noisy corpus in terms of WER (%). For the Ford corpus, we trained the systems (GMM-HMM and Speaker Independent DNN-HMM) only using the Ford training set.

Systems	dev93	eval92	Ford
GMM-HMM	9.4	5.4	12.5
SI DNN-HMM	N/A	N/A	12.6

3.2. Baselines

The baseline GMM-HMM system was trained with Speaker Adaptive Training (SAT) [16] using conventional 39-dimensional MFCC vectors with 25ms Hamming windowing and 10ms frame shift. The MFCC vectors are spliced over 9 frames and Linear Discriminant Analysis (LDA) is applied to project the spliced vectors onto a 40-dimensional sub-space. Then, Maximum Likelihood Linear Transform (MLLT) is performed for better orthogonality in the represented features. The trained GMM acoustic model contains 3,400 tied tri-phone states and 20,000 Gaussians. The baseline Speaker Independent (SI) DNN-HMM system accepts filterbank features being spliced with 15 frames, resulting in 345 dimensions. The DNN acoustic model consists of six hidden layers with sigmoid activation functions, and one output layer with softmax activation. Each hidden layer has 1,024 neurons. The number of output units matches the number of tri-phone states in the baseline GMM acoustic model. The DNN was initialized with stacked Restricted Boltzmann Machines (RBMs) and then fine-tuned using Stochastic Gradient Descent (SGD) to minimize cross entropy. The baseline performances for both Wall Street Journal (WSJ) and Ford's noisy corpus are shown in Table 1. For the WSJ experiments, we followed the Kaldi recipe for training and testing while for the Ford corpus we trained the baseline systems only using the Ford training set. There is a clear discrepancy in Word Error Rate (WER) between these two corpora since the Ford corpus is noisy as compared to the clean WSJ. Note that the DNN-HMM based ASR performance without any speaker adaptation is comparable with the GMM-HMM baseline with SAT, which verifies a great potential in DNN-HMM based ASR for noise robustness.

3.3. Experimental Setup

The configuration of the proposed transformation neural network architecture is as follows. It accepts 100-dimensional i-vectors, which are drawn per speaker by projecting GSVs that concatenate 2,048 MAP-adapted Gaussian mean vectors of 40 dimensions from a Universal Background Model (UBM) onto a total variability sub-space. The UBM was also trained with the Ford training set using the Expectation-Maximization (EM) algorithm. The generated i-vectors were shared across adaptation data from the same speakers. for the same speakers. We used the adaptation data set held out from the Ford corpus of around 1 hour in the total length, not being used in either training and testing. The two transformation neural networks have the same configuration with 3 layers of 512 neurons each with a sigmoid activation, whose output vectors are 345-dimensional, consistent with the dimension of the original filterbank acoustic feature vectors. To train the parameters of the transformation neural networks as well as the linear combination weights α and β, we employed a semi-supervised learning strategy mentioned in Section 2. We generated alignments between the tri-phone

Table 2: Comparison of WER (%) with the proposed method with different setups as well as the baseline SI DNN-HMM system. $\bar{\mathbf{x}}_a^{\text{AM}}$ is the input vector for the DNN acoustic model whereas $\bar{\mathbf{x}}_a^{\text{DNN}}$ is the transformed feature vector after the transformation neural networks followed by the linear combination layer in Eq. (3).

Systems	WER
SI DNN-HMM	12.6
$\bar{\mathbf{x}}_a^{\text{AM}} = \sigma_N(\mathbf{W}_N^l \bar{\mathbf{a}}_{N-1}^l + \bar{\mathbf{b}}_N^l)$	12.6
$\bar{\mathbf{x}}_a^{\text{AM}} = \bar{\mathbf{x}}_a^{\text{DNN}}$ in Eq. (3)	10.4
$\bar{\mathbf{x}}_a^{\text{AM}} = \bar{\mathbf{x}}_a^{\text{DNN}} + \gamma\bar{\mathbf{x}}$	10.2

states of the SI DNN-HMM system and the first-path decoding hypotheses of adaptation utterances for the purpose of cross entropy training. Then, we employed the back-propagation of log likelihood error signals from the output layer of the DNN acoustic model while freezing its parameters.

3.4. Results and Discussions

In Table 2, we compare the WERs of the proposed method with different cases as well as the baseline SI DNN-HMM system. In the first case, where $\bar{\mathbf{x}}_a^{\text{AM}} = \sigma_N(\mathbf{W}_N^l \bar{\mathbf{a}}_{N-1}^l + \bar{\mathbf{b}}_N^l)$, only the transformation neural network accepting acoustic feature vectors is enabled while the other neural network for i-vectors is disabled. This case represents a scenario where i-vectors are not available. Even such a case, we can observe the ASR system with the proposed architecture works reasonably, providing the same WER with the baseline SI DNN-HMM system. This is the perfect example of our proposed method enabling DNN-HMM based ASR with speaker adaptation to function in a hybrid fashion. In the second case (4^{th} row in the table), where $\bar{\mathbf{x}}_a^{\text{AM}} = \bar{\mathbf{x}}_a^{\text{DNN}}$, both of the transformation neural networks are enabled, as shown in Eq. (3). This is the ideal case where the proposed method for speaker adaptation is fully functional. The relative performance improvement as compared to the SI baseline is 18.3%, showing a statistically meaningful enhancement by speaker adaptation from 12.6% to 10.4% in WER. In the last case, we consider the extra input for the linear combination layer in addition to the outputs of the two transformation neural networks, i.e., original acoustic feature vector $\bar{\mathbf{x}}$. Like α and β, the linear combination weight γ is updated through back-propagation as follows:

$$\gamma := \gamma - \lambda(\frac{1}{k}\Delta\gamma) = \gamma - \lambda(\frac{1}{k}\Delta W_{\gamma,11}) \tag{14}$$

and

$$\Delta\mathbf{W}_\gamma = \bar{\mathbf{d}}_1 \cdot \bar{\mathbf{x}}^T. \tag{15}$$

The reasoning behind consideration of this case in our experiments is to see if acoustic features themselves without any transformation would have impact on speaker adaptation. Even though there is a slight improvement in WER as compared to the previous case (10.4% vs. 10.2%), it is hard to tell adding $\bar{\mathbf{x}}$ in the linear combination layer helps speaker adaptation. Rather it implies i-vectors are more important to normalize speaker-dependent variations, aligned with what is reported in [10, 11, 14, 15].

4. Conclusions

We have presented a new i-vector based speaker adaptation method for a DNN-HMM based ASR framework, focusing on in-vehicle ASR scenarios. Considering two practical restrictions in in-vehicle ASR systems with DNNs, we proposed a feature-space transform using two transformation neural networks for acoustic feature vectors and i-vectors, followed by a layer of linear combination. The parameters of the networks as well as the linear combination weights were trained via a semi-supervised learning strategy, while the parameters of the DNN acoustic model were not updated. In this way, we were able to separate the original SI DNN acoustic model from the speaker adaptation procedure, which is more practical for in-vehicle ASR systems than updating the entire DNN acoustic model parameters with a (relatively) smaller size of adaptation data.

The relative improvement in WER of 18.3% shows the proposed method has a potential to become an alternative feature-space transform to fMLLR for speaker adaptation purposes in DNN-HMM based ASR. Still, fMLLR is powerful at normalizing speaker-specific variations but in situations like in-vehicle ASR with DNNs where GMMs are not available, our proposed system architecture seems promising. In addition, a hybrid way of performing speaker adaptation in the proposed system depending upon the availability of i-vectors could be easily adopted in various voice interface applications where speaker adaptation is critical but i-vectors are not always ready. We plan to extend this work towards more practical use cases of using a much smaller size of adaptation data to normalize speaker-dependent variations in in-vehicle ASR.

5. Acknowledgements

The authors would like to thank Gint Puskorius, Francois Charette, Lakshmi Krishnan and Raju Nallapa for their support and valuable comments throughout the entire period of this work.

6. References

[1] G. Hinton, L. Deng, D. Yu, G. Dahl, A. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. Sainath, and B. Kingsbury, "Deep neural networks for acoustic modeling in speech recognition," *IEEE Signal Process. Mag.*, vol. 29, no. 6, pp. 82–97, November 2012.

[2] A. Graves and N. Jaitly, "Towards end-to-end speech recognition with recurrent neural networks," in *International Conference on Machine Learning*, 2014, pp. 1764–1772.

[3] A. Hannun, C. Case, J. Casper, B. Catanzaro, G. Diamos, E. Elsen, R. Prenger, S. Satheesh, S. Sengupta, A. Coates, and A. Ng, "Deep Speech: Scaling up end-to-end speech recognition," in *arXiv:1412.5567*, 2014.

[4] D. Amodei, R. Anubhai, E. Battenberg, C. Case, J. Casper, B. Catanzaro, J. Chen, M. Chrzanowski, A. Coates, G. Diamos, E. Elsen, J. Engel, L. Fan, C. Fougner, T. Han, A. Hannun, B. Jun, P. LeGresley, L. Lin, S. Narang, A. Ng, S. Ozair, R. Prenger, J. Raiman, S. Satheesh, D. Seetapun, S. Sengupta, Y. Wang, Z. Wang, C. Wang, B. Xiao, D. Yogatama, J. Zhan, Z. Zhu, "Deep Speech 2: End-to-end speech recognition in English and Mandarin," in *arXiv:1512.02595*, 2015.

[5] J. Schmidhuber, "Learning complex, extended sequences using the principle of history compression," *Neural Comp.*, vol. 4, pp. 234–242, 1992.

[6] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comp.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[7] G. Hinton, S. Osindero, and Y. Teh, "A fast learning algorithm for deep belief nets," *Neural Comp.*, vol. 18, no. 7, pp. 1527–1554, 2006.

[8] N. Dehak, P. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *IEEE Trans. Audio, Speech and Lang. Process.*, vol. 19, no. 4, pp. 788–797, 2011.

[9] W. Campbell, D. Sturim, and D. Reynolds, "Support vector machines using GMM supervectors for speaker verification," *IEEE Signal Process. Lett.*, vol. 13, no. 5, pp. 308–311, 2006.

[10] G. Saon, H. Soltau, D. Namahoo, and M. Picheny, "Speaker adaptation of neural network acoustic models using i-vectors," in *Workhop on Automatic Speech Recognition and Understanding*, 2013, pp. 55–59.

[11] V. Gupta, P. Kenny, P. Ouellet, and T. Stafylakis, "I-vector-based speaker adaptation of deep neural networks for french broadcast audio transcription," in *International Conference on Acoustics, Speech, and Signal Processing*, 2014, pp. 6334–6338.

[12] M. Gales, "Maximum likelihood linear transformations for HMM-based speech recognition," *Comp. Speech and Lang.*, vol. 12, pp. 75–98, 1997.

[13] F. Metze, A. Gandhe, Y. Miao, Z. Sheikh, Y. Wang, D. Xu, H. Zhang, J. Kim, I. Lane, W. Lee, S. Stuker, and M. Muller, "Semi-supervised training in low-resource ASR and KWS," in *International Conference on Acoustics, Speech, and Signal Processing*, 2015, pp. 4699–4703.

[14] Y. Miao, H. Zhang, and F. Metze, "Towards speaker adaptive training of deep neural network acoustic models," in *Interspeech*, 2014.

[15] Y. Miao, L. Jiang, H. Zhang, and F. Metze, "Improvements to speaker adaptive training of deep neural networks," in *Workshop on Spoken Language Technology*, 2014.

[16] T. Anastasakos, J. McDonough, and J. Makhoul, "Speaker adaptive training: A maximum likelihood approach to speaker normalization," in *International Conference on Acoustics, Speech, and Signal Processing*, 1997, pp. 1043–1046.