Towards Speaker Adaptive Training of Deep Neural Network Acoustic Models

Yajie Miao, Hao Zhang, Florian Metze

Language Technologies Institute, School of Computer Science, Carnegie Mellon University
Pittsburgh, PA, USA

{ymiao,haoz1,fmetze}@cs.cmu.edu

Abstract

We investigate the concept of speaker adaptive training (SAT) in the context of deep neural network (DNN) acoustic models. Previous studies have shown success of performing speaker adaptation for DNNs in speech recognition. In this paper, we apply SAT to DNNs by learning two types of feature mapping neural networks. Given an initial DNN model, these networks take speaker i-vectors as additional information and project DNN inputs into a speaker-normalized space. The final SAT model is obtained by updating the canonical DNN in the normalized feature space. Experiments on a Switchboard 110-hour setup show that compared with the baseline DNN, the SAT-DNN model brings 7.5% and 6.0% relative improvement when DNN inputs are speaker-independent and speaker-adapted features respectively. Further evaluations on the more challenging BABEL datasets reveal significant word error rate reduction achieved by SAT-DNN.

Index Terms: Deep neural networks, speaker adaptive training, automatic speech recognition

1. Introduction

In recent years, DNNs have been used widely for automatic speech recognition (ASR), showing superior performance than the state-of-the-art GMM-HMM systems [1, 2, 3]. GMM models take advantage of speaker adaptation to reduce mismatch between training and testing conditions. Speaker adaptation applies affine transforms, such as maximum likelihood linear regression (MLLR) [4], either to GMM model parameters or to speech features. However, this idea of linear transformation is not applicable for DNN adaptation in the sense that input features of DNNs normally have very high dimensions. Also, DNNs are trained discriminatively with the back-propagation (BP) algorithm rather than maximum likelihood estimation (MLE). How to effectively adapt DNN acoustic models therefore becomes an active research area.

The first group of methods perform adaptation by augmenting the speaker-independent DNN (SI-DNN) with an additional layer. The parameters of such a layer are learned via BP on the adaptation data. This layer can be inserted between the input layer and features [3, 5], acting as the new input layer. Alternatively, it can be placed immediately after the last hidden layer, which is equivalent to modifying the parameters of the softmax classification layer [5, 6]. Given insufficient adaptation data, [6] only updates the bias vector of the softmax layer for robust adaptation. Competitors of these approaches are [7, 8] in which no changes are made to the DNN structure. Instead, the shape of the activation function is adjusted to fit SI-DNN to the testing condition. Meanwhile, various efforts have been made to train DNNs on speaker-adapted features. For example, [3, 9] evaluate the effectiveness of applying GMM-derived vocal tract length normalization (VTLN) and feature-space MLLR (fMLLR) [4] transforms to DNN inputs. Speaker-adapted features can also be obtained by explicitly incorporating speaker information into DNN training. In [10], the authors use i-vectors [11, 12, 13] as low-dimensional representations of speaker characteristics, and concatenate i-vectors with raw acoustic frames such as MFCCs.

Another key technique to boost GMMs is speaker adaptive training (SAT) [14]. This paper ports the concept of SAT to DNN acoustic models. Following the similar steps adopted by SAT-GMM, SAT-DNN starts from an initial DNN[1] which has been trained over all the speakers. A feature mapping function, analogous to fMLLR transforms in GMM, is learned to incorporate i-vectors as extra information and project the original features into a speaker-normalized space. Finally, the canonical DNN model is re-finetuned in the new feature space with the feature mapping applied.

We represent this feature mapping function as a complex neural network and propose two implementations for it. The first method *AdaptNN* inserts multiple adaptation layers above the input layer of the initial DNN. This idea is related to [15] with one important difference: we append i-vectors, instead of the trained speaker codes [15, 16], to adaptation layer outputs. Benefits of using i-vectors will be discussed in Section 3. The second method involves training a smaller network which takes i-vectors as input and produces a linear feature shift at the output. This shift is added to the original DNN inputs and the resulting feature space becomes more speaker-normalized.

In the training stage, the two types of feature mappings, as well as the canonical model, can be learned with the standard BP. During decoding, the SAT-DNN model is adapted simply by extracting the i-vector for each testing speaker and feed the i-vector to the architecture. Speaker adaptation in this manner is efficient because no initial decoding pass is required. In contrast, the existing DNN adaptation methods rely on first-pass decoding hypotheses to get the supervision targets. Since DNNs are not re-finetuned on the adaptation data, this approach is less sensitive to hypotheses errors and thus more suitable for unsupervised adaptation. Experiments with the Switchboard dataset show that the proposed SAT-DNN achieves significant improvement over the baseline initial DNN, regardless of whether the baseline model has been trained on speaker-independent or speaker-adapted features. Moreover, we demonstrate the advantage of SAT-DNN on the more challenging BABEL corpus.

2. Extraction of I-Vectors

The introduction of i-vectors has resulted in state-of-the-art results in speaker recognition and verification [11, 12, 13]. The i-vector approach differs from the earlier joint factor analysis (JFA) [17] in that it has a single variability subspace, rather than separate speaker and channel subspaces. A speaker independent GMM model, also referred to as universal

[1] This initial DNN can be either SI-DNN or a DNN trained over speaker-adapted features such as fMLLR.

background model (UBM), can be trained on the speech segments from a group of speakers S. Then, we adapt the UBM to a specific speaker s and concatenate means of the speaker-dependent GMM into a supervector which is further decomposed as

$$\mathbf{v}_s = \mathbf{m} + \mathbf{T}\mathbf{i}_s \qquad (1)$$

where $\mathbf{m}$ is the supervector of the UBM means, and $\mathbf{T}$ is the total variability matrix subsuming principal components of variability in the supervector space. Training of the $\mathbf{T}$ matrix is entirely unsupervised, following the similar procedures used to train the speaker subspace in JFA [17]. The low-dimensional i-vector $\mathbf{i}_s$ contains factors on each principal component. We assume a standard normal distribution $N(0, 1)$ over i-vectors. Then, $\mathbf{i}_s$ can be obtained by maximum a posterior (MAP) estimation given the speech segments from speaker s. Previously, i-vectors have been applied successfully for discriminative adaptation of GMM models [18] and speaker adaptation of DNNs [10]. In this paper, we exploit i-vectors to realize SAT of DNNs.

3. Speaker Adaptive Training of DNNs

SAT starts from an initial DNN model built for hybrid systems. This DNN is trained to classify the input acoustic features into context-dependent HMM states. DNN outputs are the estimate of posterior probabilities of the states given each feature vector. This section first presents two methods for transforming input features and then elaborates on SAT procedures.

3.1. AdaptNN: Bottom Adaptation Layers

The first method AdaptNN, as shown on the right of Figure 1, inserts multiple fully-connected adaptation layers under the initial DNN but above the input features. Given an input feature vector $\mathbf{o}_t$ from speaker s, each adaptation layer, except the highest one, appends the corresponding i-vector $\mathbf{i}_s$ to its hidden activation. The combined outputs are propagated to the next layer as inputs. Suppose that $\mathbf{W}_m$ is the weight matrix connecting the m-1-th and m-th adaptation layer. Then, the size of $\mathbf{W}_m$ is $N_m \times (N_{m\text{-}1} + d)$, with N_m denoting the number of units at the m-th adaptation layer and d denoting the dimension of i-vectors.

The intuition behind AdaptNN is that by incorporating i-vectors, the adaptation layers convert the original DNN inputs into more speaker-independent features. Parameters of the AdaptNN network, marked with green circles in Figure 1, can be estimated with BP on the training data by keeping the initial DNN fixed. The highest adaptation layer generates the new features and must have the same dimension as the original feature vectors. Also, this highest layer uses the linear activation function, while the other adaptation layers use the sigmoid function.

Although having the similar architecture as [15], AdaptNN differs on two aspects. First, in [15], representations of speaker characteristics, also called speaker codes, have to be learned on both training and testing sets. In contrast, i-vectors can be extracted in a completely unsupervised way. Therefore, AdaptNN requires no finetuning over the adaptation data. Second, speaker codes are appended both to the adaptation layers and also to the original features [15]. However, we observe optimal recognition performance when AdaptNN appends i-vectors *only* to the adaptation layers.

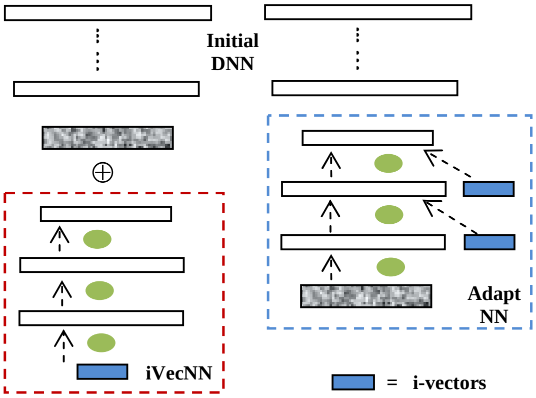

Figure 1: *Illustration of the iVecNN and AdaptNN methods for input feature mapping. The green circles mark the connection parameters for the feature mapping networks.*

3.2. iVecNN: Linear Feature Shift

For GMM models, linear feature shift has been exploited extensively for speaker adaptation and feature-space discriminative training (such as fMPE [19]). A bias vector is estimated and added to the original features, making the resulting feature space either speaker independent or discriminative. Previous work [18] has also attempted to learn feature shift from i-vectors in order for GMM adaptation. The idea can be formulated as follows:

$$\mathbf{a}_t = \mathbf{o}_t + f(\mathbf{i}_s) \qquad (2)$$

where $\mathbf{o}_t$ is the original feature vector from speaker s, and f is the function which maps the i-vector to a bias vector. After adding this bias, we can get a speaker independent feature vector $\mathbf{a}_t$. In [18], the mapping function f is formulated into region dependent linear transforms (RDLT) [20].

For DNN acoustic models, we use an i-vector neural network as f. Our method is depicted on the left of Figure 1. The i-vector network iVecNN takes i-vectors as input and generates the feature shift as output. Its output layer has the same dimension as $\mathbf{o}_t$ and uses the linear activation function. The other layers in iVecNN adopt the sigmoid activation function. This smaller iVecNN network is combined with the initial DNN via feature addition to form an even deeper network. We keep the parameters of the initial DNN unchanged. Parameters of iVecNN are updated through BP from the top softmax layer of the initial DNN. Note that inputs to this combined DNN include i-vectors together with the speech features. Thus, the number of training examples equals the number of training speech frames, not the number of i-vectors (i.e., training speakers). A notable advantage of iVecNN lies in its applicability to convolutional neural network (CNN) acoustic models [21, 22, 23]. In comparison, appending i-vectors to convolution layers is difficult because convolution outputs are generally organized in the form of feature maps. As a result, the AdaptNN method is not applicable to CNNs.

3.3. Speaker Adaptive Training

After the feature mappings are learned, speaker adaptive training is straightforward to accomplish. We apply the feature mapping, either AdaptNN or iVecNN, to input features. The upper initial DNN is further updated in the transformed feature space, while parameters of AdaptNN or iVecNN are kept fixed. This generates the canonical DNN model more independent of specific speakers. The procedures for building SAT-DNN can be summarized as follows.

1) Train the baseline initial DNN over the training data
2) Extract i-vectors for training speakers
3) Learn the feature mapping using i-vectors and based on the AdaptNN or iVecNN method
4) Update the canonical DNN model in the transformed feature space

During decoding, we extract i-vectors for testing speakers and feed the i-vectors to the architecture in Figure 1. This will adapt SAT-DNN to each testing speaker, without any finetuning on the adaptation data. Therefore, unlike SAT-GMM, SAT-DNN only needs to decode the testing data once, even if unsupervised adaptation is performed.

4. Experiments on Switchboard

4.1. Experimental Setup

The first set of experiments are on the Switchboard conversational telephone speech. For faster turnaround of tuning experiments, we select 100k utterances from the entire Switchboard-1 Phase 2 pack and create a smaller training set with 110 hours of speech, as described in [9, 24]. Evaluation is conducted on the eval2000 (Hub5'00) testing set. This testing set consists of 20 conversations from Switchboard and 20 conversations from CallHome English. We report results on the Hub5'00-SWB part. All decoding runs use a trigram language model trained solely from the Switchboard-1 transcripts. During DNN training, a 5-hour validation set, independent of the 110-hour training set, is employed for parameter tuning.

The GMM-HMM systems are built with the standard Kaldi Switchboard recipe [25]. We first train the initial ML model based on 39-dimensional MFCC+delta+acceleration features with per-speaker cepstral mean normalization. Then 9 frames of MFCCs are spliced together and projected down to 40 dimensions with linear discriminant analysis (LDA). A maximum likelihood linear transform (MLLT) is applied on the LDA features and generates the LDA+MLLT model. Finally, to deal with speaker variability, SAT is performed based on fMLLR. The SAT model has 4287 context-dependent triphone states and an average of 20 Gaussian components per state.

We turn to the open-source ALIZE toolkit [26] for i-vector extraction. The i-vector extractor uses 19-dimensional MFCCs and log-energy as the features, with the frame length of 25 ms and shift of 10 ms. Computing deltas and accelerations finally gives a 60-dimensional feature vector on each frame. Both the UBM model and the total variability matrix are trained on the entire 318 hours of Switchboard-1 speech. A 100-dimensional i-vector is generated for each training and testing speaker. One may argue that we are making use of extra data beyond the defined 110 hours. However, training of i-vector extractors is unsupervised and uses no transcripts. In practice, large amounts of untranscribed speech are easily accessible. Thus, we think that system comparison in our experiments remains fair and valid.

4.2. Baseline DNN Systems

On the 110-hour training set, we construct two DNN models on top of different feature types. The inputs of the first DNN are 11 neighboring frames of 30-dimensional log-scale filterbank coefficients with per-speaker cepstral mean and variance normalization. The second DNN is trained over 9 neighboring fMLLR frames. In both cases, the class labels for speech frames are generated by the SAT-GMM model through forced alignment. DNNs have 5 hidden layers, each of which contains 1024 units. Finetuning of the networks is to optimize the cross-entropy objective with an exponentially decaying learning rate schedule. Specifically, the learning rate starts from an initial value and remains unchanged for 15 epochs. Then the learning rate is halved at each epoch until the frame accuracy on the validation set stops to improve. A momentum of 0.5 is adopted for fast convergence, and we use the mini-batch size of 256 for stochastic gradient descent (SGD).

It's worth pointing out that DNN parameters are initialized randomly to rule out impact of pre-training on system comparison. We achieve the best WER of 19.9% on Hub5'00-SWB, while the authors of [9] report 19.7% under the similar setting. This 0.2% gap may come from differences in language model pruning, number of targets, scoring configuration, etc. After doing pre-training with Stacked Denoising Autoencoders (SDAs) [27], we are able to bring the WER down to 19.3%, a slightly better number than [9]. This means that if only acoustic modeling is concerned, we are working with a reasonable baseline. A more competitive baseline is the approach proposed in [10]. However, we experiment with this approach on our setups and fail to get gains out of it. We will continue to work on replicating the numbers reported by [10].

4.3. Results of SAT-DNN

When SAT-DNN is deployed, AdaptNN and iVecNN have the following configurations: their output layers have the same dimension as the original features (330 for filterbanks and 360 for fMLLR); each of the other layers has 512 units. Figure 2 shows SAT-DNN results as we vary the number of layers in AdaptNN and iVecNN. In general, DNNs with speaker-adapted fMLLR features get better performance than DNNs with speaker-unadapted filterbank features. For each feature type, the SAT-DNN model consistently outperforms the baseline DNN (see Table 1), even when the baseline is trained on fMLLR features. On both feature types, SAT-DNN achieves its optimal WER when AdaptNN has 3 layers (including the output layer). When switching to iVecNN, SAT-DNN performs best if iVecNN has 4 layers.

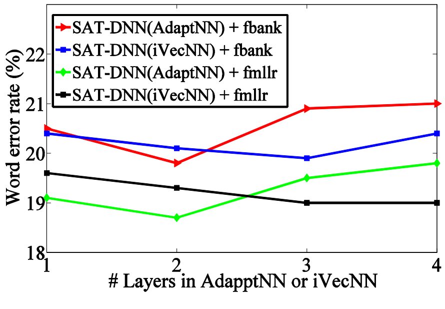

Figure 2: *Performance of SAT-DNN as the number of layers in AdaptNN and iVecNN increases. WER(%) is measured on the HUB'00-SWB set.*

Table 1 presents results corresponding to the best configuration from Figure 2. The AdaptNN method performs better than iVecNN in terms of WER of the final SAT-DNN. We think this is because the linear shift from iVecNN is not powerful enough to transform the original features sophisticatedly. With the filterbank features, SAT-DNN with AdpatNN achieves the WER of 19.8%, i.e., 7.5% relative improvement over the baseline model. When fMLLR features are used, the improvement of SAT-DNN becomes less significant (6.0% relative), simply because speaker variability has been partly modeled by fMLLR transforms.

The last two rows in Table 1 show the results when we don't update the canonical DNN model after estimating the feature mappings. In this case, we get better WER than the baseline, demonstrating the benefits of AdaptNN and iVecNN for feature normalization. However, the numbers are worse than the complete SAT-DNN model. Another natural question is: how does SAT-DNN perform without adding the i-vectors? We build SAT-DNN following the same steps (Section 3.3) but without appending i-vectors to the AdaptNN network. On the filterbank (fMLLR) features, SAT-DNN gives the WER of 20.9% (19.7%). This is marginally better than the baseline DNN but significantly worse than SAT-DNN with i-vectors.

5. Experiments on BABEL

We further evaluate SAT-DNN on the BABEL corpus that has been collected under the IARPA BABEL research program. The corpus consists of a variety of languages including Cantonese, Tagalog, Turkish, etc. Each language contains around 80 hours of conversational telephone speech for training and 10 hours for system development. The data collection covers a variety of acoustic conditions, speaking styles and dialects. Also, a large portion of the audio data are either non-speech events (e.g., breath, laugh, cough) or non-lexical speech (e.g., hesitations, fragments and foreign words). Due to all these factors, speech recognition on the BABEL corpus is a very difficult task [28, 29, 30].

In this paper, we conduct our experiments on Tagalog (IARPA-babel106-v0.2f) and Turkish (IARPA-babel105b-v0.4). We follow the similar setups as on Switchboard to build the GMM and DNN models. The SAT models of the two languages have 3890 and 3880 triphone tied states respectively. On each language, the i-vector extractor is trained only on its training data, without utilizing any external speech. During decoding, we select approximately 2 hours of speech from the entire 10-hour development data as the testing set. The trigram language model is built from training transcripts.

Table 1. *WER(%) of various DNN models on HUB'00-SWB. Results are reported with filterbank and fMLLR features respectively. Numbers in brackets are relative improvement over the baseline, which holds for Table 2 and 3.*

Models	Filterbank	fMLLR
Initial DNN (Baseline)	21.4	19.9
SAT-DNN (AdaptNN)	19.8 (7.5%)	18.7 (6.0%)
SAT-DNN (iVecNN)	19.9 (7.0%)	19.0 (4.8%)
AdaptNN + Initial DNN	20.8 (2.8%)	19.2 (3.5%)
iVecNN + Initial DNN	21.2 (0.9%)	19.7 (1.0%)

We observe that on BABEL data, the fMLLR front-end does not hold a clear advantage over the filterbank features. Therefore, we only present the results of SAT-DNN with filterbanks in Table 3. On both Tagalog and Turkish, the SAT-DNN model displays better recognition performance compared with the corresponding baseline DNN. For example, on Tagalog, SAT-DNN with AdaptNN achieves 2.2% absolute (or 4.5% relative) improvement in terms of WER. On Turkish, the improvement is enlarged to 2.7% absolute or equivalently 5.3% relative.

Table 3. *WER(%) of SAT-DNN on the BABEL Tagalog and Turkish datasets. The features are filterbanks.*

Models	Tagalog	Turkish
Initial DNN (Baseline)	49.3	51.3
SAT-DNN (AdaptNN)	47.1 (4.5%)	48.6 (5.3%)
SAT-DNN (iVecNN)	47.3 (4.1%)	49.3 (3.9%)

6. Conclusions and Future Work

In this paper, we present an effective framework to perform SAT for DNN acoustic models. Two types of neural networks, AdaptNN and iVecNN, are proposed in order for feature transformation. These networks take speaker i-vectors as additional information and are trained to map speech features into a speaker-normalized space. The canonical DNN is further updated in the new feature space to generate the final SAT-DNN model. Experiments show that SAT-DNN achieves nice gains when the initial DNN has been trained over both speaker-independent and speaker-adapted features.

The SAT-DNN model is likely to be more advantageous with improved i-vector exaction. In our future work, we will explore training of the i-vector extractor on more external data [18]. Also, as discussed in Section 3.2, the iVecNN method is applicable to CNNs. We will extend SAT to CNN acoustic models [21, 22, 23] and examine the effectiveness of the resulting SAT-CNN model.

7. Acknowledgments

This work was supported by the Intelligence Advanced Research Projects Activity (IARPA) via Department of Defense U.S. Army Research Laboratory (DoD / ARL) contract number W911NF-12-C-0015. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. Disclaimer: The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of IARPA, DoD/ARL, or the U.S. Government. This work used the Extreme Science and Engineering Discovery Environment (XSEDE), which is supported by National Science Foundation grant number OCI-1053575.

8. References

[1] G. Dahl, D. Yu, L. Deng, and A. Acero, "Context-dependent pre-trained deep neural networks for large vocabulary speech recognition," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 20(1), pp. 30-42, 2012.

[2] F. Seide, G. Li, and D. Yu, "Conversational speech transcription using context-dependent deep neural networks," in *Proc. Interspeech*, pp. 437–440, 2011.

[3] F. Seide, G. Li, X. Chen, and D. Yu, "Feature engineering in context-dependent deep neural networks for conversational speech transcription," in *Proc. ASRU*, pp. 24–29, 2011.

[4] M. Gales, "Maximum likelihood linear transformations for HMM-based speech recognition," *Computer Speech and Language*, vol. 12, pp. 75–98, 1998.

[5] B. Li, and K. C. Sim, "Comparison of discriminative input and output transformations for speaker adaptation in the hybrid NN/HMM systems," in *Proc. Interspeech*, pp. 526–529, 2010.

[6] 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. IEEE Spoken Language Technology Workshop*, pp. 366–369, 2012.

[7] S. M. Siniscalchi, J. Li, and C.-H. Lee, "Hermitian-based hidden activation functions for adaptation of hybrid HMM/ANN models," in *Proc. Interspeech*, pp. 526–529, 2012.

[8] S. M. Siniscalchi, J. Li, and C.-H. Lee, "Hermitian polynomial for speaker adaptation of connectionist speech recognition systems," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 21, no. 10, pp. 2152-2161, 2013.

[9] S. P. Rath, D. Povey, K. Vesely, and J. Cernocky, "Improved feature processing for deep neural networks," in *Proc. Interspeech*, 2013.

[10] G. Saon, H. Soltau, D. Nahamoo, and M. Picheny, "Speaker adaptation of neural network acoustic models using i-vectors," in *Proc. ASRU*, pp. 55-59, 2013.

[11] N. Dehak, R. Dehak, P. Kenny, N. Brummer, P. Ouellet, and P. Dumouchel, "Support vector machines versus fast scoring in the low-dimensional total variability space for speaker verification," in *Proc. Interspeech*, pp. 1559–1562, 2009.

[12] N. Dehak, P. J. 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.

[13] O. Glembek, L. Burget, P. Matejka, M. Karafiat, and P. Kenny, "Simplification and optimization of i-vector extraction," in *Proc. ICASSP*, pp. 4516-4519, 2011.

[14] T. Anastasakos, J. McDonough, and J. Makhoul, "Speaker adaptive training: a maximum likelihood approach to speaker normalization," in *Proc. ICASSP*, pp. 1043-1046, 1997.

[15] O. Abdel-Hamid, and H. Jiang, "Fast speaker adaptation of hybrid NN/HMM model for speech recognition based on discriminative learning of speaker code," in *Proc. ICASSP*, pp. 7942-7946, 2013.

[16] O. Abdel-Hamid, and H. Jiang, "Rapid and effective speaker adaptation of convolutional neural network based models for speech recognition," in *Proc. Interspeech*, 2013.

[17] P. Kenny, P. Ouellet, N. Dehak, V. Gupta, and P. Dumouchel, "A study of interspeaker variability in speaker verification," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 16, no. 5, pp. 980-988, 2008.

[18] M. Karafiat, L. Burget, P. Matejka, O. Glembek, and J. Cernocky, "iVector-based discriminative adaptation for automatic speech recognition," in *Proc. ASRU*, pp. 152-157, 2011.

[19] D. Povey, B. Kingsbury, L. Mangu, G. Saon, H. Soltau, and G. Zweig, "fMPE: discriminatively trained features for speech recognition," in *Proc. ICASSP*, pp. 961-964, 2005.

[20] B. Zhang, S. Matsoukas, and R. Schwartz, "Recent progress on the discriminative region-dependent transform for speech feature extraction," in *Proc. Interspeech*, 2006.

[21] O. Abdel-Hamid, A. Mohamed, H. Jiang, and G. Penn, "Applying convolutional neural networks concepts to hybrid NN-HMM model for speech recognition," in *Proc. ICASSP*, pp. 4277-4280, 2012.

[22] T. N. Sainath, A. Mohamed, B. Kingsbury, and B. Ramabhadran, "Deep convolutional neural networks for LVCSR," in *Proc. ICASSP*, pp. 8614-8618, 2013.

[23] T. N. Sainath, B. Kingsbury, A. Mohamed, G. Dahl, G. Saon, H. Soltau, T. Beran, A. Aravkin, and B. Ramabhadran, "Improvements to deep convolutional neural networks for LVCSR," in *Proc. ASRU*, 2013.

[24] K. Vesely, A. Ghoshal, L. Burget, and D. Povey, "Sequence-discriminative training of deep neural networks," in *Proc. Interspeech*, 2013.

[25] D. Povey, A. Ghoshal, et al., "The Kaldi speech recognition toolkit," in *Proc. ASRU*, 2011.

[26] J.-F. Bonastre, N. Scheffer, D. Matrouf, C. Fredouille, A. Larcher, A. Preti, G. Pouchoulin, N. Evans, B. Fauve, and J. Mason, "ALIZE/SpkDet: a state-of-the-art open source software for speaker recognition," in *Proc. ISCA/IEEE Speaker Odyssey 2008*.

[27] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P. Manzagol, "Stacked denoising autoencoders: learning useful representations in a deep network with a local denoising criterion," *Journal of Machine Learning Research*, vol. 11, pp. 3371-3408, 2010.

[28] J. Gehring, Y. Miao, F. Metze, and A. Waibel, "Extracting deep bottleneck features using stacked auto-encoders," in *Proc. ICASSP*, 2013.

[29] J. Gehring, W. Lee, K. Kilgour, I. Lane, Y. Miao, and A. Waibel, "Modular Combination of Deep Neural Networks for Acoustic Modeling," in *Proc. Interspeech*, pp. 94-98, 2013.

[30] Y. Miao, F. Metze, and S. Rawat, "Deep maxout networks for low-resource speech recognition," in *Proc. ASRU*, 2013.