IMPROVEMENTS TO SPEAKER ADAPTIVE TRAINING OF DEEP NEURAL NETWORKS

Yajie Miao, Lu Jiang, Hao Zhang, Florian Metze

Language Technologies Institute, School of Computer Science, Carnegie Mellon University
{ymiao,lujiang,haoz1,fmetze}@cs.cmu.edu

ABSTRACT

Speaker adaptive training (SAT) is a well studied technique for Gaussian mixture acoustic models (GMMs). Recently we proposed to perform SAT for deep neural networks (DNNs), with speaker i-vectors applied in feature learning. The resulting *SAT-DNN* models significantly outperform DNNs on word error rates (WERs). In this paper, we present different methods to further improve and extend SAT-DNN. First, we conduct detailed analysis to investigate i-vector extractor training and flexible feature fusion. Second, the SAT-DNN approach is extended to improve tasks including bottleneck feature (BNF) generation, convolutional neural network (CNN) acoustic modeling and multilingual DNN-based feature extraction. Third, for transcribing multimedia data, we enrich the i-vector representation with global speaker attributes (age, gender, etc.) obtained automatically from video signals. On a collection of instructional videos, incorporation of the additional visual features is observed to boost the recognition accuracy of SAT-DNN.

Index Terms— Deep neural networks, speaker adaptive training, speech recognition

1. INTRODUCTION

DNNs have been applied widely to automatic speech recognition (ASR), showing superior performance over the traditional GMM-HMM models [1, 2]. Like GMM models, DNNs also face the challenge of potential mismatch between training and testing conditions. Various methods have been proposed for speaker adaptation of DNN models. Examples of the solutions include augmenting the speaker-independent DNN with additional layers [3, 4], adapting the activation function [6] and using speaker-adapted feature space [2, 7, 8]. To further resolve this issue, our recent study [9] ported the concept of SAT to DNNs. Training of SAT-DNN models starts from an initial DNN which has been trained over all the speakers. Then, a smaller neural network, referred to as *iVecNN*, is learned to convert speaker i-vectors [10] into linear feature shifts. These shifts are added to the original DNN inputs and the resulting feature space becomes more speaker-normalized. Finally, we update the initial DNN in the new feature space, which generates the canonical DNN model. On hybrid systems, SAT-DNN models have shown significant WER improvement over DNNs [9], regardless of whether the inputs are speaker-independent (e.g., filterbank) or speaker-adapted (fMLLR) features. The goal of this paper is to analyze appropriate settings for the SAT-DNN architecture and explore possible improvements to it.

First of all, we examine two critical variations in the configuration of SAT-DNN. The first variation lies in the training of i-vector extractors [10], and we study the impact of i-vector training data on the performance of SAT-DNN. Also, in the existing SAT-DNN, feature shifts from the iVecNN network are fused with the original DNN inputs via a simple `sum` operation. For more flexible feature fusion, we explore two other fusion functions: the `product` and the more complicated `weighted sum`.

Second, hybrid systems have been shown to benefit from the SAT-DNN approach [9]. Apart from hybrid systems, popular applications of deep learning also include BNF generation [11] and CNN-based acoustic modeling [12, 13]. We investigate the utility of the SAT idea in improving both tasks. The introduced SAT-BNF and SAT-CNN models are found to perform better than their baselines relatively by 4-8%. Moreover, DNNs trained with multilingual data have served successfully as deep feature extractors on a new language [14]. For more invariant feature representations, we extend SAT-DNN to the learning of multilingual feature extractors and develop two strategies to train iVecNN over multiple languages. Cross-language experiments show that on the new language, feature extractors trained with SAT-DNN achieve better WERs than DNN-based extractors.

Third, we investigate enrichment of i-vectors with visual features when transcribing video data. Specifically, speaker attributes (age, gender and race) are extracted from video frames which show the images of the speakers. These attributes, related with acoustic characteristics, are then appended to i-vectors as additional descriptors. Experiments show that incorporating the additional visual features brings modest improvement to SAT-DNN models. This way of leveraging visual features is applicable in real-world scenarios because it requires global visual features at the video level. In comparison, previous work [15, 16, 17] on audio-visual ASR relies on frame-level lip/mouth features which can be accessed only in highly constrained conditions.

2. REVIEW OF SAT-DNN

The architecture of the SAT-DNN model [9] is illustrated in Figure 1. The starting point of SAT-DNN is an initial DNN which has been fully trained for hybrid system building.

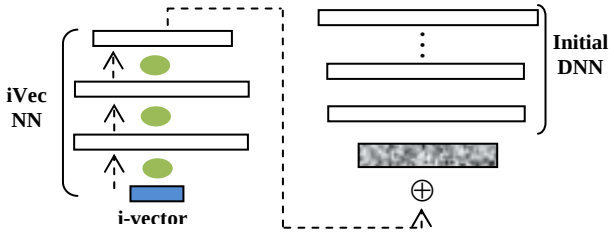

Figure 1. *The SAT-DNN model. Green circles depict the connection parameters for the iVecNN network.*

Training of SAT-DNN consists of two major steps. First, with the initial DNN fixed, we learn the smaller network iVecNN (on the left of Figure 1) whose inputs are i-vectors. Originating from speaker identification [10], i-vectors represent compactly the acoustic characteristics of speakers and have been exploited in ASR for speaker adaptation [8, 18]. The outputs of iVecNN are linear feature shifts which can be formulated as:

$$\mathbf{a}_t = \mathbf{o}_t + f(\mathbf{i}_s) \quad (1)$$

where $\mathbf{i}_s$ denotes the i-vector for speaker s, $\mathbf{o}_t$ is an original feature vector from speaker s, f denotes iVecNN which maps the i-vector into a feature shift. After adding this shift to $\mathbf{o}_t$, we get a speaker normalized feature vector $\mathbf{a}_t$. In the second step, we fix the trained iVecNN and update the parameters of the initial DNN in the new feature space $\mathbf{a}_t$. This finally generates the canonical DNN more independent of specific speakers. Training of iVecNN and updating of the DNN can be done via the standard error back-propagation (BP) algorithm.

During decoding, we extract i-vectors for testing speakers and feed the i-vectors to the architecture in Figure 1. This will adapt the SAT-DNN model to each testing speaker without any DNN fine-tuning on the adaptation data. Also, since i-vector extraction is totally unsupervised, no initial decoding pass is needed prior to adaptation. Therefore, SAT-DNN enables us to perform unsupervised adaptation in a very efficient manner.

3. ANALYSIS AND EXTENSIONS OF SAT-DNN

This section explores the behaviors of SAT-DNN with respect to variations in its configuration. With the same experimental setup, we also present two extensions to it.

3.1. Experimental Setup and Baseline Results

Our experiments inherit the setup used in [9]. We select 100k utterances from the Switchboard-1 pack and create a training set with 110 hours of conversational telephone speech. The testing set is the Switchboard part of Hub5'00 and contains 20 conversations. Decoding uses a trigram language model trained from the entire Switchboard-1 transcripts. A GMM-HMM system is built with the standard Kaldi recipe [19]. This gives us the SAT-GMM model which has 4287 context-dependent triphone states.

Table 1. *WERs(%) of baseline DNN and SAT-DNN with the two feature types.*

Models	filterbanks	fMLLRs
DNN	21.7	19.2
SAT-DNN	19.3	17.9

DNN models are constructed with the Kaldi+PDNN framework[1] [20]. We first build a DNN model on speaker-independent features, i.e., 11 neighboring frames of 40-dimensional log-scale filterbank coefficients. The second DNN is built on 11 frames of speaker-adapted fMLLR features. On both feature types, the class labels for speech frames are generated by the SAT-GMM model through forced alignment. DNN fine-tuning optimizes the cross-entropy objective with mini-batch based stochastic gradient descent (SGD) and using the Newbob learning rate schedule.

For SAT-DNN, the iVecNN network contains 3 hidden layers each of which has 512 units. The output layer of iVecNN has the same dimension as the original features (440 for the two feature types) and uses the linear activation function. The other layers in iVecNN adopt the sigmoid activation function. Table 1 shows the results of DNN and SAT-DNN on the Hub5'00-SWB testing set. The i-vector extractor is trained on the entire 318 hours of Switchboard-1 speech. A 100-dimensional i-vector is generated for each training and testing speaker.

Note that WERs of both DNN and SAT-DNN in Table 1 differ from the numbers reported in [9]. This is because the DNN in Table 1 has 6 hidden layers and is pre-trained with Stacked Denoising Autoencoders (SDAs) [21], while the DNN in [9] has 5 layers and is randomly initialized. For SAT-DNN, besides the differences in initial DNNs, this study turns to Kaldi's in-built i-vector extractor, while [9] uses the external open-source ALIZE toolkit [22]. Due to these factors, we are getting better WERs than [9] and comparable numbers to [7], which means that we are working with a strong baseline. The results of the baseline DNN and SAT-DNN, as well the BNF and CNN models, can be replicated with our publicly released Kaldi+PDNN.

3.2. Analysis about SAT-DNN Configuration

Our first investigation focuses on improvement to i-vector extraction. Training of i-vector extractors uses no transcripts and is unsupervised in nature. Therefore, the training data can be enlarged easily by pooling more untranscribed speech. We add 2000 hours of Fisher telephone speech into the training set, while keeping all the other i-vector configuration (e.g., the i-vector dimension, the number of Gaussians in UBM) unchanged. From Table 2, we can see that no gains are obtained from augmenting the data for i-vector extractor training. When DNN inputs are fMLLRs, we also attempt to train the i-vector extractor over fMLLR

[1] http://www.cs.cmu.edu/~ymiao/kaldipdnn.html

Table 2. *WERs(%) of SAT-DNN with filterbank features when different sets of training data are used for i-vector extractors.*

I-Vector Training Data	WER%
Switchboard-1	19.3
Fisher + Switchboard-1	19.3

features instead of the raw MFCCs. In this case, SAT-DNN gives the WER of 17.8%, which is only 0.1% absolute improvement over the baseline (17.9%). In general, the SAT-DNN method performs robustly to the training of i-vector extractors.

Second, the existing SAT-DNN fuses the feature shifts and DNN inputs via a sum operation. A natural alternative is the product function. That is, the iVecNN outputs and the original inputs are multiplied in an element-wise fashion. Then, iVecNN is learned to generate feature weights rather than feature shifts. Another more complicated function is weighted sum which can be formally written as

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

where the vectors **m** and **n** contain weights for the original features and linear shifts respectively, **b** is a bias vector, and $\otimes$ represents element-wise product. We do not define the values of **m**, **n** and **b** in advance. Instead, these three vectors are learned together with iVecNN through BP. Table 3 shows the performance of these two functions when SAT-DNN takes filterbanks as inputs. We observe that weighted sum gives slight improvement due to more flexibility in feature fusion, while product deteriorates the WER.

3.3. SAT for Bottleneck Feature Extraction

BNF features can be extracted from a narrow bottleneck hidden layer in DNNs and used to construct GMM-HMM tandem systems. In this subsection, we improve the quality of the BNF front-end by applying the SAT-DNN approach. The initial DNN is a Deep BNF (DBNF) network described in [11, 23]. It has 6 hidden layers, in which the 5-th layer is a bottleneck with only 42 units. When the DBNF network, either a DNN or SAT-DNN, has been trained, we build a LDA+MLLT tandem system with the BNF features. Specifically, 9 consecutive BNF frames are spliced and then projected down to 40 dimensions with LDA. On top of the LDA+MLLT system, discriminative training is further performed with the boosted maximum mutual information (BMMI) objective [24].

Table 4 compares the BMMI models when different architectures are employed for BNF generation. Again, the DBNF network can take the speaker-independent filterbanks

Table 3. *WERs(%) of SAT-DNN with fusion function variants.*

Fusion Function	WER%
product	19.5
weighted sum	19.1

Table 4. *WERs(%) of BMMI tandem systems.*

Front-end	filterbanks	fMLLRs
DBNF with DNN	19.6	18.0
DBNF with SAT-DNN	18.0	17.5

and the speaker-adapted fMLLRs as inputs. In each case, we can see that application of the SAT technique results in superior bottleneck features and improves the recognition performance of the BMMI tandem systems.

3.4. SAT for Convolutional Neural Networks

Deep convolutional neural networks (CNNs) have been exploited as an alternative to DNNs for acoustic modeling [12, 13]. Instead of using fully-connected parameter matrices, CNNs are characterized by parameter sharing and local feature filtering. The local filters help to capture locality along the frequency bands. On the convolution layer, a max-pooling layer is added for feature normalization and dimension reduction. CNNs reduce spectral variation in the speech signal, and have been experimentally confirmed to generate better WERs compared to DNNs.

Our CNN architecture follows [26], consisting of 2 convolution *stages* and 4 fully-connected layers. A key difference is that we are now applying 2-dimensional convolution over both time and frequency, while [26] uses convolution only on the frequency axis. The CNN inputs are 29 neighboring frames each of which has 29-dimensional log-scale filterbanks. The first convolution layer filters the image-like 29x29 inputs using 64 kernels with the size of 4x4x1. The second convolution layer takes as inputs the outputs from the first convolution stage and filters them with 64 kernels of 4x4x64. Every convolution layer is followed by a max-pooling layer with the pooling size of 2x2. The convolution operation has the stride of 1 and the max-pooling operation is non-overlapping. Each of the 4 fully-connected layers contains 1024 hidden units and uses the sigmoid activation function.

Motivated by SAT-DNN, implementation of SAT for CNNs is straightforward to accomplish. After getting the initial CNN, we learn the iVecNN network which has the output dimension of 29x29. Then, the CNN model is updated in the newly-estimated feature space (Equation 1). Apart from SI filterbanks, we also turn to speaker-adapted filterbanks with VTLN for complete evaluations. From Table 5, we can see that the improvement of SAT-CNN over CNN becomes less significant in comparison to the improvement of SAT-DNN over DNN. This is because in general, CNNs normalize the speech features more effectively than DNNs, which decreases the efficacy of SAT.

Table 5. *WERs(%) of CNN and SAT-CNN with two feature types.*

Models	filterbanks	VTLN-filterbanks
CNN	19.9	19.0
SAT-CNN	19.2	18.6

Also, since speaker variability has been partly modeled by VTLN transforms, SAT-CNN achieves marginal gains over the CNN baseline when the inputs are VTLN-filterbanks.

4. SAT-LUFE FOR LANGUAGE UNIVERSAL FEATURE EXTRACTION

This section introduces SAT of multilingual DNNs in order for more effective *language-universal feature extraction* (LUFE) [14, 25, 26, 27]. LUFE aims to generate high-level, language-independent feature representations from DNNs which have been trained collectively over a group of languages. The hidden layers of the multilingual DNN are shared across all the languages, while each language has its own output layer, speech data and class labels. Fine-tuning is carried out using the standard SGD with one critical difference: each epoch traverses data from all the *source languages* instead of a single language. Parameters of the shared layers are updated with gradients accumulated from multiple languages. Interested readers can refer to [14, 26] for more details regarding multilingual DNN training.

After the multilingual DNN is trained, the shared hidden layers act as a deep feature extractor. Given a *new language*, DNN hybrid models can be built using features generated from this extractor, instead of the raw acoustic features (e.g., MFCCs). Cross-language acoustic modeling in this fashion enables knowledge transfer across languages and thus improves ASR on the new language, especially when the new language has limited transcribed speech.

4.1. LUFE with Speaker Adaptive Training

The key to the success of SAT-DNN is the learning of the speaker-normalized feature space with iVecNN and i-vectors. This motivates us to combine SAT and LUFE, which potentially enhances the feature representations. We perform SAT for the multilingual DNN after the feature extractor has been fully trained as described by [26]. Similarly, the iVecNN network is learned to convert i-vectors into linear feature shifts. Then parameters of the multilingual DNN are updated on the new features with iVecNN applied. We propose two strategies to train iVecNN over multiple languages. The iVecNN network can be shared by all the source languages and trained in the same manner as the multilingual DNN. Alternatively, each source language can have its iVecNN separately. These two methods are referred to as *Share* and *Unshare* respectively.

When switching to the new language, we input the i-vectors, together with the speech features, to the SAT architecture which generates feature representations from the highest layer. If the iVecNN network has been trained with Share, then this iVecNN can be ported to the new language directly. Otherwise, we have to retrain the iVecNN network on the new language from scratch. More comparison between Share and Unshare will be conducted in Section 4.2.

4.2. Experiments and Analysis

The quality of the feature extractors is evaluated on a cross-language acoustic modeling task. Our experiments use the multilingual corpus collected under the BABEL program [11, 23, 26, 27]. This corpus covers a wide range of languages including Cantonese, Tagalog, Pashto, etc. Each language contains around 80 hours of conversational telephone speech for training and 10 hours for decoding. Additionally, there is also a low-resource 10-hour condition under which only 10 hours of transcribed speech are allowed to be used for system building. We take Tagalog (IARPA-babel106-v0.2f) as the new language. The source languages include the 80-hour sets of Cantonese (IARPA-babel101-v0.4c), Turkish (IARPA-babel105b-v0.4) and Pashto (IARPA-babel104b-v0.4aY).

On the new language, hybrid systems are built with the outputs from the feature extractors. We select 2 hours of speech from the 10-hour decoding data as the testing set. It is worth noting that i-vector extraction is always performed within each language. We are not training a joint i-vector extractor over all the multilingual speech. Table 6 presents WERs of the new-language hybrid systems under the 80-hour and 10-hour conditions. *No-LUFE* denotes the purely monolingual case without using any feature extractors.

In comparison to No-LUFE, applying LUFE brings significant gains especially under the 10-hour condition. SAT-LUFE, which uses the SAT-trained feature extractor, outperforms the normal LUFE consistently. Comparing the two iVecNN training strategies reveals that Share performs better than Unshare under the 10-hour condition. We think the reason is that Unshare requires re-estimation of iVecNN on the new language. This re-estimation may not be reliable under limited training data (10 hours). When the training data are increased to 80 hours, the iVecNN network can be trained more robustly with Unshare. At the same time, the iVecNN re-estimation helps to adapt the feature extractor to the new language. That is why we observe slightly better WERs achieved by Unshare under the 80-hour condition.

5. ENRICHING I-VECTORS WITH VISUAL FEATURES

Transcribing multimedia data has become an active research area in ASR [28, 29]. In addition to the audio track, the video signal provides rich information which can potentially benefit ASR. Previous work [15, 16, 17] has successfully combined audio and supplemental visual features (e.g., lip

Table 6. *Results of feature extractors under both new-language conditions. WERs (%) are reported on the 2-hour testing set.*

Feature Extractor	80-hour	10-hour
No-LUFE	49.3	65.8
LUFE	46.7	59.6
SAT-LUFE iVecNN-Share	46.1	57.8
SAT-LUFE iVecNN-Unshare	45.7	58.3

contours, mouth shapes, facial expressions, etc.) to improve ASR. However, the applicability of these proposals is limited by the availability of frame-level visual features which are usually not obtainable from open-domain data (e.g., YouTube videos). Also, since the video and audio have different sampling rates, aligning frames from these two streams poses another challenge.

In contrast, some video-level features can be easily extracted even from real-world videos, which to some extent globally characterize the acoustic conditions. Examples of these features are scenes (office, street, etc.) of the conversations and attributes (age, gender, race, etc.) of the speakers. This section studies the utility of SAT-DNN acting as a flexible framework for incorporating global visual features. Although we only deal with speaker attributes in this work, SAT-DNN is capable of using other feature types, ranging from segment-level actions/concepts to video-level scene/event attributes.

5.1. Dataset

We download a collection of around 4k English videos from online archives such as Youku.com, Tudou.com, YouTube.com and CreativeCommons.org. These videos are intended for expertise sharing on specific tasks (e.g., oil change and sandwich making), and have an average duration of 90 seconds. For each video, the raw closed captions are available. We take several steps to convert the collected data in an applicable ASR training corpus. These steps include cleaning and normalizing transcripts, down-sampling the audio track, and adding new words into the dictionary. Time markers for each utterance are obtained via forced alignment with the raw closed captions and our existing broadcast news recognizer. This finally gives us 94 hours of speech data, out of which 90 hours are selected for training and 4 hours for testing.

5.2. Visual Feature Extraction

In this paper, we focus on speaker attributes that can be deduced automatically from the videos. We observe that in each of these instructional videos, the (principal) speaker tends to appear at the beginning for a brief introduction. Based on this observation, we extract only the frame at the position which is immediately after the first utterance starts. Then, this image, which is assumed to show the speaker, is submitted to the Face++ API[2] that returns 3 attributes: `age`, `gender` and `race`. The value of age is continuous, while gender and race have categorical values. We categorize the age value into 6 bins: < 20, 20-30, 30-40, 40-50, 50-60, >60. These bins are represented by a 6-dimensional vector. Each of the 6 elements is a binary variable indicating whether the speaker's age falls into the corresponding bin. The gender classification result is converted into a 2-dimensional vector whose binary elements denote male and female respectively. Similarly, a 3-dimensional vector is employed to represent the 3 possible values of race: White, Black and Asian. The final attribute vector is assembled by concatenating these three sub-vectors. For example, the attribute vector for a 58-year-old, male and white speaker is [0 0 0 0 1 0 | 1 0 | 1 0 0]. For some videos, no speaker attributes can be generated due to image resolution, illumination condition or timing of the speakers' show-ups. In this case, we set the elements in each of the sub-vectors uniformly, e.g., [0.5, 0.5] for gender and [0.33 0.33 0.33] for race.

[2] www.faceplusplus.com

5.3. Method and Results

With the training corpus, GMM, DNN and SAT-DNN acoustic models are constructed by following the procedures described in Section 3. We only experiment with filterbank features as DNN and SAT-DNN inputs. Training data for the i-vector extractor consist of the defined 90-hour training set, as well as the additional 400 hours of videos collected from the same sources. Our first set of experiments are with the DNN model. We append the 11-dimensional attribute vector to the filterbank features on each speech frame. Table 7 shows 0.5% absolute WER improvement (22.2% vs. 22.7%) achieved by adding the attributes. This verifies that these speaker attributes are helpful for acoustic modeling. When switching to SAT-DNN, we append the speaker attributes to the speaker i-vectors rather than to speech frames. Training of SAT-DNN with the enriched i-vectors follows the same protocol as adopted by the baseline SAT-DNN. The only difference is that the i-vector dimension is enlarged from 100 to 111. We can see from Table 7 that the incorporation of the speaker attributes reduces the WER of SAT-DNN by 0.4% absolutely (21.0% vs. 21.4%).

6. CONCLUSIONS AND FUTURE WORK

This paper has studied improvements and extensions to the SAT-DNN approach from the following aspects. First, we analyze the impact of i-vector extractor training and alternative feature fusion. Second, SAT-DNN is applied to various tasks including BNF generation, CNN acoustic modeling and multilingual DNN feature extraction. Last, when transcribing multimedia data, we explore the enrichment of i-vectors with additional visual features. For our future work, we are interested to further study the portability of the iVecNN network across domains and languages. Also, we would like to improve SAT-DNN by enriching i-vectors with more visual features such as scene and action classification results.

Table 7. *DNN and SAT-DNN without and with the speaker attributes. WERs (%) are reported on the 4-hour testing set.*

Model	DNN	SAT-DNN
Baseline	22.7	21.4
+ Speaker Attributes	22.2	21.0

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 and Department of Interior National Business Center contract number D11PC20068. 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, DoI/NBC, or the U.S. Government.

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

[3] 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.

[4] K. Yao, D. Yu, F. Seide, H. Su, L. Deng, and Y. Gong, "Adaptation of context-dependent deep neural networks for automatic speech recognition," in *Proc. SLT*, pp. 366–369, 2012.

[5] 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.

[6] 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.

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

[8] 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.

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

[10] 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.

[11] J. Gehring, Y. Miao, F. Metze, and A. Waibel, "Extracting deep bottleneck features using stacked auto-encoders," in *Proc. ICASSP*, pp. 3377-3381, 2013.

[12] 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.

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

[14] J. Huang, J. Li, D. Yu, L. Deng, and Y. Gong, "Cross-language knowledge transfer using multilingual deep neural network with shared hidden layers," in *Proc. ICASSP*, pp. 7304-7308, 2013.

[15] S. Dupont, and J. Luettin, "Audio-visual speech modeling for continuous speech recognition," *IEEE Transactions on Multimedia*, vol. 2(3), pp. 141-151, 2000.

[16] G. Gravier, G. Potamianos, and C. Neti, "Asynchrony modeling for audio-visual speech recognition," in *Proc. HLT*, pp. 149-152, 2012., pp. 1-6, 2002.

[17] Y. Kashiwagi, M. Suzuki, N. Minematsu, and K. Hirose, "Audio-visual feature integration based on piecewise linear transformation for noise robust automatic speech recognition," in *Proc. SLT*, pp. 149-152, 2012.

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

[20] Y. Miao, "Kaldi+PDNN: building DNN-based ASR systems with Kaldi and PDNN," arXiv:1401.6984, 2014.

[21] 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.

[22] 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*.

[23] 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.

[24] D. Povey, "Discriminative training for large vocabulary speech recognition," Ph.D. dissertation, Cambridge Univ., 2003.

[25] Y. Miao, H. Zhang, and F. Metze, "Distributed learning of multilingual DNN feature extractors using GPUs," to appear in *Proc. Interspeech*, 2014.

[26] Y. Miao, and F. Metze, "Improving language-universal feature extraction with deep maxout and convolutional neural networks," to appear in *Proc. Interspeech*, 2014.

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

[28] N. Jaitly, P. Nguyen, A. Senior, and V. Vanhoucke, "Application of pretrained deep neural networks to large vocabulary speech recognition," in *Proc. Interspeech*, 2012.

[29] H. Liao, E. McDermott, and A. Senior, "Large scale deep neural network acoustic modeling with semi-supervised training data for YouTube video transcription," in *Proc. ASRU*, pp. 368-373, 2013.