fMLLR based feature-space speaker adaptation of DNN acoustic models

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

[1]Amazon.com, USA.
[2]Amazon.com, India.

{sparta,bjornh,matsouka,arindamm,nikko,srigar}@amazon.com

Abstract

We investigate the problem of speaker adaptation of DNN acoustic models in two settings: the traditional unsupervised adaptation and a supervised adaptation (SuA) where a few minutes of transcribed speech is available. SuA presents additional difficulties when a test speaker's adaptation information does not match the registered speaker's information. Employing feature-space maximum likelihood linear regression (fMLLR) transformed features as side-information to the DNN, we reintroduce some classical ideas for combining adapted and unadapted features: early and late fusion methods, as well as the estimation of the fMLLR transforms using simple target models (STM). Results show that early fusion helps DNNs generalize better when features are combined after a non-linear bottleneck layer, while late fusion improves robustness, specifically in mismatched cases. STM give consistent improvements in both settings.

Index Terms: Speech recognition, feature-space speaker adaptation, DNN acoustic models.

1. Introduction

Neural networks have become the state-of-the-art in acoustic modeling (AM) for large vocabulary continuous speech recognition systems (LVCSR) [1]; specifically neural networks with multiple layers of affine transforms followed by nonlinear activations are trained to estimate the posterior probability of clustered triphone states. Two common approaches to use the posteriors are: (a) the hybrid approach [2], where the neural network outputs are divided by the state priors to estimate emission likelihoods in hidden Markov model (HMM) (b) the TANDEM approach [3], where the posteriors are further processed and used as additional features to estimate HMM likelihoods using Gaussian mixture models (GMM).

Employing the hybrid approach, this paper describes our work on speaker adaptation of deep neural network (DNN) acoustic models in two settings: the traditional unsupervised 2-pass adaptation and a supervised adaptation (SuA). Our work in both settings happens in the context of spontaneous speech recognition; in the SuA setting, additionally, we have two minutes of transcribed speech available per speaker.

Whereas speaker adaptation of GMM-HMM AM is a well-established problem, with a number of approaches, adaptation of DNN-HMM AM is an area of active research, exploring feature and model space methods [4, 5, 6, 7, 8, 9, 10, 11, 12]. Furthermore speaker adaptation in the SuA setting presents a set of additional challenges: although the SuA setting provides us with per speaker adaptation data which can be used to perform supervised DNN-HMM AM adaptation, there are cases of varying difficulties that can occur when a test speaker's utterance is presented to the system. For instance in our definition of the problem for the SuA setting, a speaker whose data was used to estimate adaptation material need not necessarily match the speaker presented at test time: this leads to two cases namely, *matched* and a *mismatched speakers*.

Despite a number of recent papers on DNN adaptation, [13] observes that in comparison to GMM adaptation, DNN adaptation yields much less gains over an unadapted, speaker independent (SI) model. They speculate that retraining a SI DNN for each speaker on adaptation data results in overfitting, showing improvements by retraining speaker specific DNNs with regularization. Another approach is to adapt a smaller set of parameters, such as activations in a layer [11]. These approaches [13, 11] use more adaptation data per speaker than this work. For instance, [13] uses SuA similar to our setup, but with 10 minutes per speaker instead of only 2 minutes.

Consequently, we investigate a feature space adaptation method using a single adaptively trained DNN for all speakers; specifically, we employ per-speaker fMLLR transformed features as input to the DNN. Additionally, in SuA setting, mismatched speakers can make the AM fragile. To this end, we successfully reintroduce some classical ideas to combine features and also to increase robustness of the DNN against sudden speaker changes: early fusion performed through a bottleneck nonlinearity and a late-stage fusion of DNNs yield gains. Furthermore, we introduce the idea of estimating fMLLR transforms using simple target models (STM), and find significant gains over using more complex models for fMLLR estimation.

The rest of the paper is organized as follows: feature-space adaptation, its motivation, some challenges, and proposed approaches are presented in Section 2; datasets used for training, cross-validation, and testing are described in Section 3. The baseline acoustic model, the proposed acoustic modeling methods, and the rest of the ASR system used in our experiments are discussed in Sections 4 and 5. Results are discussed in Section 6. Concluding remarks are drawn in Section 7.

2. DNN adaptation using fMLLR

We provide a brief background on feature-space adaptation of DNN-HMM AM. We also discuss connections to other approaches; an interpretation of feature-space transform as an adaptation of the first layer is presented.

DNNs are cascades of multiple layers of affine transforms and non-linear activation functions, the latter being typically sigmoid for the hidden layers and softmax for the output layer. After the first layer of affine transform and non-linearity, the output of the activation at the first hidden layer can be written:

$$h_1 = \phi_1(A_1 X + b_1) \quad (1)$$

where X is the input, and A_1 and b_1 are the linear transform and bias for the first layer, respectively. Let $\{A_i, b_i\}_{i=1:N}$ be the set of all parameters of the DNN. Retraining the DNN for each speaker is a solution to the problem of estimating speaker-specific model; however, in most LVCSR tasks the amount of data per speaker is only of the order of a few minutes. More robust estimates of DNN parameters can be made when an update is made to a small set of parameters or in a subspace of the parameters. A recent work [11] adapts the activations specific to a layer by re-scaling them. Transfer learning approaches to adapting DNN freeze the parameter estimates for first few layers and adapt or retrain the last few layers [14]. Other recent methods adopt a *feature-space adaptation* technique.

2.1. Feature-space adaptation

Since adaptation has a long history in GMM-HMM AM, and a number of methods have been developed for GMM AM[1], DNN adaptation can be performed by providing speaker information as an input; for instance [5, 6, 7] use i-vectors, fMLLR transformed features, and VTLN respectively. These can be seen as adapting the bias in first layer of the DNN:

$$h_1^{corr} = \phi_1(A_1 X + b_1 + b_1^{corr}) = \phi_1(A_1 X + b_1 + T b_1^{spkr}) \quad (2)$$

where b_1^{spkr} is a speaker-specific side information and T is an affine transform learnt by back-propagation. Of course b_1^{spkr} can also be learnt by back-propagation [8].

2.2. fMLLR transformed features

Although originally proposed as a per-speaker model-space transformation for GMM AM [16], constrained MLLR is frequently interpreted as a feature-space transformation; it is straightforward to show that the feature transform is an inverse of the model transform. The transform estimation is usually formulated in the maximum likelihood sense: the actual estimation is done using the Expectation Maximization (EM) algorithm. We provide fMLLR transformed features as input to the DNN, making it a feature normalization technique. Adaptation of DNN can therefore be interpreted as speaker adaptive training; that is a different motivation than behind the original MLLR and fMLLR adaptation of GMMs.

There are many advantages to performing DNN AM adaptation with fMLLR: (a) the parameter estimation is quick – a few iterations of EM usually suffice; (b) a few minutes of audio data is usually sufficient for a robust estimation of the parameters – in case of further data sparsity, the number of parameters can be further reduced by considering a “diagonal” or a “bias” only transform; (c) MLLR transforms can compensate to some extent for acoustic mismatches [19]. (d) it is less sensitive to transcription errors than optimizing a discriminative criterion (e) parameter estimation can be done in a number of settings: (i) unsupervised 2-pass estimation; (ii) supervised estimation with reference transcripts; and (iii) online estimation of the transform parameters. This paper focuses on the unsupervised and supervised estimations (i.e. (i) and (ii)).

2.3. Feature streams, fusion, and bottleneck layer

We remarked earlier that in comparison to GMM AM adaptation, speaker adaptation of DNN AMs yields much less gains with respect to the corresponding SI model. While it is possible that DNNs are relatively more invariant to speaker variations, it is natural to ask if feature streams themselves, or their combinations can be made more effective.

While using fMLLR transformed features by themselves as input to the DNN does provide some feature normalization, DNN AM would be sensitive to noise in fMLLR estimation. Another possibility is to combine the fMLLR features with SI features to provide robustness. However a direct combination of SI and speaker dependent (SD) features (say, like Equation 2) forces the DNN to learn the weights in the first layer using feature streams drawn from different distributions. Our hypothesis is that this has been one of the reasons why an L2 regularization with a weight decay was required in [13]. To address this we introduce late and early fusion (Section 5.2). Early-fusion with a bottleneck can act as a powerful regularizer, while late-fusion can provide significant robustness in cases where fMLLR estimation is noisy.

2.4. Complex and simple target models

When using fMLLR transforms for recognition with GMM-HMM AM a distinction can be made between target and recognition models [20]; i.e. the model used for the estimation of feature transforms, the target model, can be distinct from the recognition model. It was shown that this separation yields gains in recognition [20]. Furthermore it was argued that using simpler target models (STM), such as using a single Gaussian per HMM state, can yield bigger gains than using more complex target models (CTM). While the separation of target and recognition models occurs naturally when using fMLLR transformed features with a DNN-HMM AM, it is interesting to investigate whether gains can be obtained with STM.

3. Data sets

In our experiments, we use a dataset of roughly 200 hours of speech data from an in-house collection; this is a subset of the data set used for training our production models. From the same data collection we carve out a development set of 20 hours, used to optimize meta-parameters like the language model scale, and an evaluation set of 10 hours on which we report WER improvements. Training, development, and evaluation set do not have any speaker overlap. An additional two minutes of transcribed speech is available for each test speaker to be used in the SuA setting to compute fMLLR transforms (unsupervised two-pass adaptation does not use this data). Mismatched SuA presents a random speaker’s fMLLR transform (computed on 2 min) along with a test utterance.

We report results as relative reduction in WER compared to a strong baseline system. All decoding experiments use the same language model with the language model scale being tuned separately for each experiment for minimum WER on the development set. Furthermore, all experiments use the same acoustic decision tree and differ only in the DNN and its input. The baseline DNN model is described in Section 4.

4. Baseline System

Features for the SI DNN consist of log-energies computed on the audio signals every 10 ms (25 ms analysis window), from a bank of 20 filters placed on mel-warped spectrum. A causal mean estimate is computed and subtracted; we refer to these features as Log Filter Bank Energies (LFBE). The input to the DNN is a temporal window over LFBE features, splicing 5 frames of left and right temporal context (total input size $11 * 20 = 220$), which is normalized by applying a global mean

[1]Common methods include constrained and unconstrained MLLR [15, 16] and vocal tract length normalization (VTLN) [17, 18].

and variance normalization, computed from the training data.

The baseline acoustic model is an SI DNN-HMM trained on the LFBE features. It has 4 hidden layers each consisting of 1,536 logistic sigmoid units; the input to the DNN is the 220 dimensional spliced and whitened LFBE features, the output is a softmax layer consisting of 3,052 units, corresponding to the leaves of a phonetic decision tree. The architecture is summarized as $220\text{x}1536^4\text{x}3052$. To denote the same DNN with an emphasis on the input and output layers, we use $220 \oplus 3052$. Figure 1(a) shows this DNN (about 12M parameters).

The DNN trainer is an internal tool; DNN AM is pretrained by growing layer-by-layer and optimizing its parameters by minimizing the cross-entropy loss function. The model parameters are learnt on the training set by doing several epochs of cross entropy training on a single GPU; during all training runs the frame accuracy saturates by 12 epochs. Although sequence training has been reported to yield gains over cross-entropy training (indeed we observe fairly large gains as well) this article is restricted to cross entropy training for quicker turnaround of experiments. The frame-level targets for cross-entropy training are derived using a GMM AM trained on the same set. We refer to this DNN AM as LFBE DNN.

5. Proposed Methods and Systems

This section describes methods and systems for fMLLR estimation; then we discuss late and early fusion.

5.1. Estimation of fMLLR transforms

For estimating the training-time speaker transforms we build a GMM AM on the same training set as follows: beginning with the LFBE features and with the application of DCT, the parameters of the GMM AM are estimated using the EM algorithm, with E-step obtained from LFBE DNN using reference transcripts. After estimating LDA and MLLT, and keeping the E-step from the LFBE DNN fixed, fMLLR transforms are estimated by performing 2 iterations of Gaussian posterior computation and 5 inner row-iterations. The convergence is quick and a couple iterations of EM suffice. We keep the phonetic decision tree fixed, i.e. 3,052 states; the GMM AM has 80K Gaussians.

fMLLR features are derived by applying the DCT, LDA, MLLT, and fMLLR transforms on the LFBE features to obtain 40-dimensional features. A context window of 11 frames is applied to obtain an fMLLR stream input size of 440.

5.1.1. Unsupervised and supervised fMLLR transforms

The classical method of estimating fMLLR transforms on the test set is the unsupervised 2-pass estimation [16]; we follow the same procedure as the training-time estimation except that, instead of reference transcripts, decoded-hypotheses are used to perform the E-step with the SI LFBE DNN. Using the LDA and MLLT from training-time, the rest of the steps in the estimation of the fMLLR transforms are identical to the description in Section 5.1.

We also study estimating fMLLR using in the SuA setting, where transcription for two minutes of speech per speaker is available. Reference transcripts corresponding to utterances are used instead of decoded-hypotheses to do the E-step of fMLLR estimation. Two complexities with the SuA setting are: (a) amount of transcribed speech is limited – we study one and two minutes per speaker; (b) a potential mismatch between registered and test speakers.

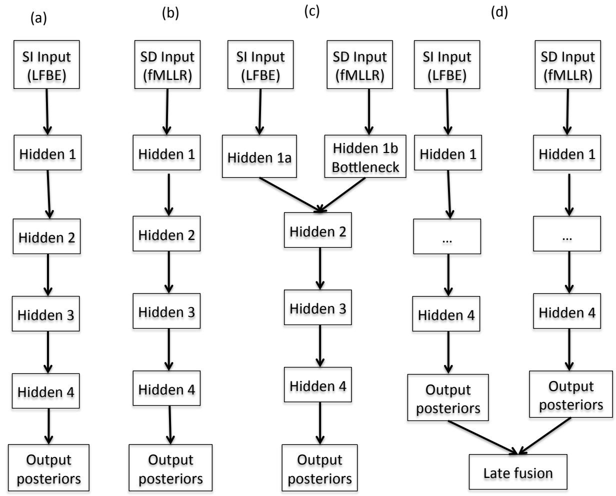

Figure 1: DNN architectures: (a) linear chain speaker independent LFBE DNN (b) linear chain speaker dependent fMLLR DNN (c) Early fusion DNN (d) Late fusion DNN.

5.1.2. Complex vs simple target models

The notion of simple and complex target models was discussed in Section 2.4. CTM and STM used for fMLLR estimation are: (a) CTM: GMM AM with 80k Gaussians (3,052 states), and (b) STM: GMM AM with 3k Gaussians (one per HMM state).

5.2. Late and early fusion

Adapted features can be used alone (Figure 1(b)); more robust models can likely be built when SI and SD features are combined. We begin by studying late fusion (i.e. combining posteriors); while having a layer of non-linearity can make the SI and SD features more compatible, it is natural to ask if features can be combined earlier.

<u>*Late Fusion*</u> (LF): Combination techniques typically assign a weight to each classifier's evidence [21]. We use static weighting with equal weights; this would yield better posterior estimates when errors by the two DNNs are uncorrelated. Figure 1(d) illustrates this, represented as $\binom{220\oplus3052}{440\oplus3052}\text{x}3052$.

<u>*Early Fusion*</u> (EF): Combining LFBE and fMLLR at the input would yield a 660-dimensional feature ($= 11*(20+40)$). This DNN can be represented as: $660 \oplus 3052$; we call it input fusion (IF). Combining after a layer of non-linearity as two Bernoulli vectors would yield a representation: $\binom{220\text{x}1536}{440\text{x}hidden} \oplus 3052$, illustrated in Figure 1(c). While the size of the introduced hidden layer is empirical, when the speaker information is viewed as a bias correction in Equation 2, it is reasonable to expect it to be a bottleneck layer, especially since it acts as a regularizer.

6. Results and Discussion

Results are presented on unsupervised and SuA settings, with standalone, LF, and EF methods. We also discuss our results with simple target models.

6.1. Unsupervised adaptation

Table 1 (rows 1 to 3) presents the results for unsupervised 2-pass adaptation in terms of relative Word Error Rate (WER) improvement with respect to the baseline SI LFBE system[2]. These

[2] Positive numbers imply a WER improvement/reduction.

Table 1: *Unsupervised adaptation with fMLLR: SI and SD represent standalone LFBE and fMLLR features. IF refers to input fusion of SI and SD features without a non-linearity. LF refers to late fusion, while EF1 and EF2 refer to fusion after 1 and 2 non-linear layers, respectively. EF1 BN refers to fusion after layer 1 with a bottleneck.*

Sno	Sys	Architecture	WER reduction %
CTM: GMM based fMLLR estimation			
1	SI	$220 \oplus 3052$	0
2	SD	$440 \oplus 3052$	0
3	IF	$660 \oplus 3052$	-15.2%
4	LF	$\binom{220 \oplus 3052}{440 \oplus 3052}$x3052	**+9.9%**
5	EF1	$\binom{220\text{x}1536}{440\text{x}1536} \oplus 3052$	-7.8%
6	EF2	$\binom{220\text{x}1536\text{x}1536}{440\text{x}1536\text{x}1536} \oplus 3052$	$+4.8\%$
7	EF1 BN	$\binom{220\text{x}1536}{440\text{x}32} \oplus 3052$	**+7.4%**
STM: single Gaussian fMLLR estimation			
8	EF1 BN	$\binom{220\text{x}1536}{440\text{x}32} \oplus 3052$	**+10.4%**
9	LF	$\binom{220 \oplus 3052}{440 \oplus 3052}$x3052	**+14.7%**

results suggest that although the features themselves (LFBE and fMLLR, rows 1 and 2) are good, they may not be compatible in the input space – for instance, one is a correlated set of features (LFBE), while the other uncorrelated (fMLLR).

These trends are further confirmed by measures such as frame classification accuracy (FA) and cross-entropy (Xent) loss. FA on the test set for SI, SD, and IF DNNs are (in %): 48.2, 49.6, and 40.7 respectively; similarly the Xent loss for the DNNs are (in bits): 2.2, 2.1, and 2.6 respectively. Clearly the SD features are good, if not slightly better than SI; nevertheless they yield no gains when combined with the SI features. [13] reports a similar behavior, requiring L2 regularization.

Let us now study the results with LF (Table 1 row 4): over an SI DNN we get a 9.9% relative reduction in WER. These results confirm that we need at least one layer of non-linearity to combine LFBE and fMLLR features. These gains are confirmed by other measures: LF DNN yields 54.1% FA on the test set.

Let us consider earlier fusions: EF1 and EF2 in Table 1 refer to fusion after 1 and 2 non-linear layers respectively. While there is a gain with fusion after the second layer (EF2), there is a slight loss with fusion after the first layer (EF1). Although the exact size of the EF1 is largely empirical, we expect it to be a bottleneck (BN) layer performing regularization. With a bottleneck of 32 units (it is a parameter that can be empirically studied), a significant WER reduction can be obtained (-7.4%).

With LF and EF BN, DNNs yield gains between 7 to 10%. Interestingly, these relative gains are in the range of what can be obtained using fMLLR with GMM-HMM AM. Although not presented here, combining SD and SI DNNs (rows 1 and 2) through LF (row 4) is as good as combining SI and EF1 BN DNN (rows 1 and 7) via LF.

6.2. Supervised adaptation

In the previous section we reported fMLLR adaptation results in the classical 2-pass framework exploiting early and late fusion. This section reports results in the SuA setting, where the speaker transforms were trained with 1 min and 2 min.

Table 2: *Supervised adaptation: Relative WER reductions with fMLLR transforms estimated using CTM (WER_{ctm}) and a STM (WER_{stm}).*

Sno	Sys	Architecture	WER_{ctm} reduction %	WER_{stm} reduction %
1	SI	$220 \oplus 3052$	0	0
Matched speakers 1 min SuA				
2	EF1 BN	$\binom{220\text{x}1536}{440\text{x}32} \oplus 3052$	$+7.4\%$	**+12.1%**
3	LF	$\binom{220 \oplus 3052}{440 \oplus 3052}$x3052	$+7.8\%$	**+13.4%**
Matched speakers 2 min SuA				
4	EF1 BN	$\binom{220\text{x}1536}{440\text{x}32} \oplus 3052$	$+7.4\%$	**+13.0%**
5	LF	$\binom{220 \oplus 3052}{440 \oplus 3052}$x3052	$+8.2\%$	**+13.9%**
Mismatched speakers 2 min SuA				
6	EF1 BN	$\binom{220\text{x}1536}{440\text{x}32} \oplus 3052$	-45.0%	-38.9%
7	LF	$\binom{220 \oplus 3052}{440 \oplus 3052}$x3052	-6.6%	-1.2%

Matched speakers: When registered and test speakers are matched, with Table 2 (rows 2 to 5, and column WER_{ctm}), we confirm the gains using SD features employing early and late fusion. For the CTM case and an 1-min of supervised data, we get a 7.8% relative reduction in WER. Interestingly, in contrast to the unsupervised fMLLR estimation, with the SuA setting, LF does not yield as big a gain over EF (BN). This is perhaps due to having smaller amount of data for fMLLR estimation. Also, the gains due to EF (BN) seem to stabilize even with 1-min of adaptation data – this could be explained by the SD DNN being able to combine LFBE stream more effectively to make decisions despite reduced data.
Mismatched speakers: This is a hard case especially with EF; however, LF recovers most of the loss due to mismatched speakers. We incur about 6.6% relative loss in WER in comparison to the SI DNN.

6.3. Effect of single Gaussian fMLLR estimation

The results for the STM for the unsupervised fMLLR estimation is presented in Table 1(rows 8 and 9), while the results for STM with supervised adaptation is presented in Table 2 (column WER_{stm}). In both cases, there is a gain in estimating per-speaker fMLLR matrices using STM (almost 5% relative). This gain is not only consistent across the unsupervised and SuA scenarios, but we also almost completely recover the loss due to the mismatched speaker case using LF.

7. Conclusions

We investigated the problem of speaker adaptation of DNN AM in two settings: unsupervised and supervised adaptations. Employing the fMLLR transformed features, this paper successfully reintroduces some classical ideas for increasing the robustness of DNN AM: early and late fusion, as well as the estimation of the fMLLR transforms using STM. Our results show that early fusion helps adapted systems generalize better when feature streams are combined after a layer of bottleneck non-linearity, while late fusion improves the robustness of DNNs, specifically in mismatched cases. STM alone gives about 5% relative improvements in both settings. Overall we get 10 to 14 % relative WER gain in unsupervised adaptation and the matched supervised speaker adaptation; we suffer almost no loss with mismatched speakers.

8. References

[1] G. Hinton, L. Deng, D. Yu, G. Dahl, A. R. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, and B. Kingsbury, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal Processing Magazine*, vol. 29, pp. 82–97, 2012.

[2] H. Bourlard and N. Morgan, *Connectionist speech recognition - a hybrid approach*. Kluwer Academic Publishers, 1994.

[3] H. Hermansky, D. Ellis, and S. Sharma, "Tandem connectionist feature extraction for conventional HMM systems," in *In Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2000.

[4] F. Seide, G. Li, X. Chen, and D. Yu, "Feature engineering in context dependent deep neural networks for conversational speech transcription," in *In Proceedings of IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, 2011.

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

[6] T. Sainath, B. Kingsbury, H. Soltau, and B. Ramabhadran, "Optimization techniques to improve training speed of deep neural networks for large speech tasks," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 21, pp. 2267 – 2276, 2013.

[7] R. Serizel and D. Giuliani, "Vocal tract length normalisation approaches to DNN-based children's and adults' speech recognition," in *In Proceedings of IEEE Spoken Language Technology (SLT) Workshop*, 2014.

[8] S. Xue, O. Abdel-Hamid, H. Jiang, L. Dai, and Q. Liu, "Fast adaptation of deep neural network based on discriminant codes for speech recognition," *ACM/IEEE Transactions on Audio, Speech, and Language Processing*, vol. 22, pp. 1713 – 1725, 2014.

[9] 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 *In Proceedings of IEEE Spoken Language Technology (SLT) Workshop*, 2012.

[10] H. Liao, "Speaker adaptation of context dependent deep neural networks," in *In Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2013.

[11] P. Swietojanski and S. Renals, "Learning hidden unit contributions for unsupervised speaker adaptation of neural network acoustic models," in *In Proceedings of IEEE Spoken Language Technology (SLT) Workshop*, 2014.

[12] S. Garimella, A. Mandal, N. Strom, B. Hoffmeister, S. Matsoukas, and S. H. K. Parthasarathi, "Robust i-vector based adaptation of DNN acoustic model for speech recognition," in *In Proceedings of Interspeech*, 2015.

[13] A. Senior and I. Lopez-Moreno, "Improving DNN speaker independence with i-vector inputs," in *In Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2014.

[14] Y. J, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?" in *In Proceedings of Advances in Neural Information Processing Systems 27*, 2014.

[15] C. J. Leggetter and P. C. Woodland, "Maximum likelihood linear regression for speaker adaptation of continuous density hidden markov models," *Computer Speech and Language*, no. 9, 1995.

[16] M. J. F. Gales and P. C. Woodland, "Mean and variance adaptation within the MLLR framework," *Computer Speech and Language*, vol. 10, pp. 249–264, 1996.

[17] L. Lee and R. Rose, "A frequency warping approach to speaker normalization," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 6, pp. 49 – 60, 1998.

[18] S. Wegmann, D. McAllaster, J. Orloff, and B. Peskin, "Speaker normalization on conversational telephone speech," in *In Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1996.

[19] S. H. K. Parthasarathi, S. Y. Chang, J. Cohen, N. Morgan, and S. Wegmann, "The blame game in meeting room ASR: An analysis of feature versus model errors in noisy and mismatched conditions," in *In Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2013.

[20] G. Stemmer, F. Brugnara, and D. Giuliani, "Adaptive training using simple target models," in *In Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2005.

[21] J. Kittler, M. Hatef, R. P. W. Duin, and J. Matas, "On combining classifiers." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, pp. 226–239, 1998.