Interspeech 2018
2-6 September 2018, Hyderabad

Speaker adaptive training and mixup regularization for neural network acoustic models in automatic speech recognition

Natalia Tomashenko[1,2], *Yuri Khokhlov*[3], *Yannick Estève*[1]

[1]LIUM, University of Le Mans, France
[2]ITMO University, Saint-Petersburg, Russia
[3]STC-Innovations Ltd, Saint-Petersburg, Russia

natalia.tomashenko@univ-lemans.fr, khokhlov@speechpro.com, yannick.esteve@univ-lemans.fr

Abstract

This work investigates speaker adaptation and regularization techniques for deep neural network acoustic models (AMs) in automatic speech recognition (ASR) systems. In previous works, GMM-derived (GMMD) features have been shown to be an efficient technique for neural network AM adaptation. In this paper, we propose and investigate a novel way to improve speaker adaptive training (SAT) for neural network AMs using GMMD features. The idea is based on using inaccurate transcriptions from ASR for adaptation during neural network training, while keeping the exact transcriptions for targets of neural networks. In addition, we apply a mixup technique, recently proposed for classification tasks, to acoustic models for ASR and investigate the impact of this technique on speaker adapted acoustic models. Experimental results on the TED-LIUM corpus show that the proposed approaches provide an additional gain in speech recognition performance in comparison with the speaker adapted AMs.

Index Terms: speech recognition, acoustic models, data augmentation, mixup, deep neural networks, GMMD features, speaker adaptation, speaker adaptive training, MAP

1. Introduction

Adaptation of neural network acoustic models is a rapidly developing research area. The aim of acoustic model (AM) adaptation is to reduce mismatches between training and testing acoustic conditions and improve the accuracy of the automatic speech recognition (ASR) system for a target speaker or channel using a limited amount of adaptation data from the target acoustic source.

Various adaptation methods have been developed for deep neural network (DNN) hidden Markov model (HMM) AMs. They include *linear transformation*, such as linear input network transformation (LIN) [1, 2], feature-space discriminative linear regression (fDLR) [3, 4], linear hidden network (LHN) [1], linear output network (LON) [2], and output-feature discriminative linear regression [4]. In order to improve generalization during the adaptation *regularization techniques*, such as L2-prior regularization [5], Kullback-Leibler divergence regularization [6], conservative training [7] and others [8] are used. There are also several *model-space adaptation* methods, such as learning speaker-specific hidden unit contributions (LHUC) [9], the adaptation parameters estimation via maximum a posteriori (MAP) linear regression [10] and hierarchical MAP approach [11]. The concept of *multi-task learning* (MTL) has been applied to the task of speaker adaptation in several works [12–14] and has been shown to improve the performance of different model-based DNN adaptation techniques. Using *auxiliary features*, such as i-vectors [15–17], is another widely used approach in which the acoustic feature vectors are augmented with additional speaker-specific or channel-specific features computed for each speaker or utterance at both training and test stages. Alternative methods include adaptation with speaker codes [18], factorized adaptation [19] and multi-factor aware joint DNN training [20]. Another way of DNN AM adaptation is based on combining Gaussian mixture model (GMM) and DNN models [21–27]. In the past, many effective adaptation algorithms that have been developed for GMM-HMM systems, such as maximum a posteriori (MAP) adaptation [28], maximum likelihood linear regression (MLLR) [28], and others [29]. A common way to apply GMM-HMM adaptation algorithms to DNN-HMM models is using GMM-adapted features as input for a DNN. For example, features adapted with fMLLR are used for DNN training in [3, 21, 22, 26]. GMM-derived (GMMD) features [27, 30–33] provide a universal method for transfer of adaptation algorithms from the GMM models to DNN framework. GMMD features are extracted using an auxiliary GMM model and are fed to DNN models as auxiliary or basic features. Adaptation of a DNN model trained on GMMD features is performed through adaptation of an auxiliary GMM model used in GMMD feature extraction.

Methods developed in this paper and their applications mainly focus on the speaker adaptive training (SAT) technique which is based on using GMMD features. The objective of this paper is to propose and investigate various approaches of speaker adaptation performance improvement.

The first contribution consists in a novel improved method of SAT using GMMD features. The idea is based on using inaccurate transcriptions for speaker adaptation during DNN AM training.

The second contribution concerns the *mixup* technique, recently proposed in papers [34, 35] for several tasks, such as image classification [34], recognition of Google commands [34], and sound recognition [35]. Despite the fact that these papers demonstrate the promising results for the classification problem, the question of applying the mixup technique to the continuous speech recognition task has not been studied in the literature. In this paper, we first discuss a way to integrate mixup training for recurrent neural network (RNN) acoustic models, and propose a method to apply mixup for neural network models sequence-trained with *lattice-free maximum mutual information* (LF-MMI) criterion [36]. Then, we investigate if it is possible to improve the performance of SAT models, trained with GMMD features, by applying the mixup technique.

The rest of the paper is organized as follows. Section 2 discusses mixup AM training for ASR and introduces a possible way to apply mixup for DNN AMs trained with LF-MMI cri-

10.21437/Interspeech.2018-2209

terion. A SAT technique, based on the use of GMMD features, and its proposed improved modification are presented in Section 3. Section 4 describes the experimental results for SAT, mixup training and their combination. Finally, the conclusions are given in Section 5.

2. Mixup for acoustic models in ASR

Mixup technique [34], also called *between class learning* in [35], has recently been proposed in the literature for several classification tasks as a form of data augmentation and regularization for deep neural networks. The idea of this method is based on adding during DNN training new synthetic feature vector examples obtained as a linear combination of original feature vectors. Targets for this synthetic vectors are generated as a linear combination of corresponding targets of vectors used in combination. Let $\mathbf{x}^i$ and $\mathbf{x}^j$ denote two feature vectors from the training data set. Then, a synthetic feature vector $\tilde{\mathbf{x}}^{i,j}(\xi)$ is obtained as a linear combination of these vectors as follows:

$$\tilde{\mathbf{x}}^{i,j}(\xi) = \xi\mathbf{x}^i + (1-\xi)\mathbf{x}^j, \tag{1}$$

where $\xi \in [0,1]$ is a random variable representing a mixing weight. In our case, ξ follows continuous uniform distribution $\mathcal{U}(0, 0.5)$. The synthetic target vector $\tilde{\mathbf{y}}^{i,j}(\xi)$ for $\tilde{\mathbf{x}}^{i,j}(\xi)$ is modeled as

$$\tilde{\mathbf{y}}^{i,j}(\xi) = \xi\mathbf{y}^i + (1-\xi)\mathbf{y}^j, \tag{2}$$

where $\mathbf{y}^i$ and $\mathbf{y}^j$ are the targets for $\mathbf{x}^i$ and $\mathbf{x}^j$, respectively.

In this paper, we investigate one modification of mixup for the speech recognition task. More specifically, we are interested in applying the mixup concept to sequence-trained neural network AMs, which nowadays are state-of-the-art of acoustic modeling in ASR.

2.1. Mixup for sequence-trained neural networks on lattice-free MMI

Lets denote a sequence of input vectors in the training corpus $\mathbb{T}$ as $\mathbf{X}^i = \{\mathbf{x}_1^i, \ldots, \mathbf{x}_T^i\}$, and the corresponding targets as $\mathbf{Y}^i = \{\mathbf{y}_1^i, \ldots, \mathbf{y}_T^i\}$, so that $(\mathbf{X}^i, \mathbf{Y}^i) \in \mathbb{T}$. Each epoch of training using the mixup algorithm consists of the following steps:

1. For each $(\mathbf{X}^i, \mathbf{Y}^i) \in \mathbb{T}$:

- **1.1.** Randomly choose from the training corpus another sequence of training vectors $(\mathbf{X}^j, \mathbf{Y}^j) \in \mathbb{T}$.
- **1.2.** Get $\xi \leftarrow \mathcal{U}(0, 0.5)$.
- **1.3.** Generate a new synthetic sequence of input vectors as $\tilde{\mathbf{X}}^{i,j}(\xi) = \{\tilde{\mathbf{x}}_1^{i,j}(\xi), \ldots, \tilde{\mathbf{x}}_T^{i,j}(\xi)\} \in \tilde{\mathbb{T}}$, where $\tilde{\mathbf{x}}_t^{i,j}(\xi)$ is obtained from $\mathbf{x}_t^i$ and $\mathbf{x}_t^j$ as shown in Formula (1).
- **1.4.** Generate a sequence of new synthetic target vectors for $\tilde{\mathbf{X}}^{i,j}(\xi)$ as $\tilde{\mathbf{Y}}^{i,j}(\xi) = \{\tilde{\mathbf{y}}_1^{i,j}(\xi), \ldots, \tilde{\mathbf{y}}_T^{i,j}(\xi)\}$, where $\tilde{\mathbf{y}}_t^{i,j}(\xi)$ is obtained using Formula (2). This new pair of sequences will be a part of the synthetic corpus $\tilde{\mathbb{T}}$: $(\tilde{\mathbf{X}}^{i,j}(\xi), \tilde{\mathbf{Y}}^{i,j}(\xi)) \in \tilde{\mathbb{T}}$.

2. Complete the current epoch of training on the obtained synthetic data $\tilde{\mathbb{T}}$.

In this paper, we focus on applying this algorithm to neural networks trained with LF-MMI [36]. The MMI objective function [37, 38] aims to maximize the posterior probability of the correct utterance, also decreasing the probability of incorrect alternatives, given the model:

$$\mathcal{F}_{MMI}(\mathbf{\Lambda}) = \sum_{r=1}^{R} \log \frac{p_{\mathbf{\Lambda}}(\mathcal{O}_r|\phi_r)^k P(\phi_r)}{\sum_{\phi} p_{\mathbf{\Lambda}}(\mathcal{O}_r|\phi)^k P(\phi)}, \tag{3}$$

where $\mathbf{\Lambda}$ represents the acoustic model parameters; $\boldsymbol{\mathcal{O}} = (\mathcal{O}_1, \ldots, \mathcal{O}_R)$ is the set of training sentences; k is the acoustic scale; $P(\phi)$ is the language model probability for sentence $\mathcal{O}$; $\phi_r = \phi(\mathbf{W}_r)$ is the composite HMM model, corresponding to the (correct) transcription of the training sentence $\mathcal{O}_r$ and $\mathbf{W}_r$ is the sequence of words in this transcription. Here the numerator corresponds to the data given the correct word sequence $\mathbf{W}_r$, and the denominator corresponds to the total likelihood of the data given all possible word sequences $\mathbf{W}$.

To compute Formula (3), *numerator* and *denominator graphs* are used. In [36] for LF-MMI both these graphs are represented in the form of finite state acceptors (FSAs), and a phone-level n-gram language model (LM) is used instead of a word-level LM. Details about these FSAs are provided in [36]. Each arc of the numerator FSA can be associated with a frame index. In order to apply the algorithm described above to the LF-MMI DNN training, step **1.4.** should be modified to operate on the level of numerator FSAs. One way to do this is to perform a weighted combination of two nominator FSAs into a single FSA. Weights for the combination are proportional to ξ^α and $(1-\xi)^\alpha$, where ξ and $(1-\xi)$ are the same weights that are applied for the corresponding feature vectors, and α is a scaling factor[1]. Other ways to apply mixup for ASR can be found in [39].

3. Improved SAT using GMMD features

The use of *GMM-derived* (GMMD) features has been shown to provide an efficient technique of neural network AM adaptation for different adaptation tasks, such as speaker adaptation [27, 30, 40], environment or noise adaptation [31, 32]. In this section, we describe a standard (Section 3.1) and improved (Section 3.2) SAT procedures for neural network AMs using GMMD features.

3.1. Speaker adaptation of neural network AMs using GMMD features

GMMD features are obtained using an auxiliary GMM-HMM model, which transforms acoustic feature vectors into log-likelihoods vectors. For the auxiliary GMM-HMM model, a monophone or triphone GMM-HMM with a low number of states (50–200) can be used. At this step, speaker adaptation of the auxiliary speaker-independent (SI) GMM-HMM model is performed for each speaker in the training corpus using correct transcriptions and a new speaker-adapted GMM-HMM model is created in order to obtain speaker-adapted GMMD features.

For a given acoustic feature vector, a new GMMD feature vector is obtained by calculating log-likelihoods across all the states of the auxiliary GMM model on the given vector. Suppose $\mathbf{o}_t$ is the acoustic feature vector at time t, then the new GMMD feature vector $\mathbf{f}_t$ is calculated as follows:

$$\mathbf{f}_t = [p_t^1, \ldots, p_t^n], \tag{4}$$

where n is the number of states in the auxiliary GMM-HMM model,

[1] In this paper, we empirically chose to use $\alpha = 3$ for experiments.

$$p_t^i = \log\left(P(\mathbf{o}_t \mid s_t = i)\right) \quad (5)$$

is the log-likelihood estimated using the GMM-HMM. Here s_t denotes the state index at time t.

The adapted GMMD feature vector $\mathbf{f}_t$ is concatenated with the original vector $\mathbf{o}_t$ to obtain vector $\mathbf{x}_t$. These features are used as the input for training a SAT neural network AM.

The described SAT training is universal and can be used for any type of neural network topology. For experiments in this paper, we chose one promising architecture, recently proposed in [41] – *interleaved TDNN-LSTM* models. These neural network models combine in their structure temporal convolution with recurrent neural networks and consist of a number of interleaving *time delay neural network* (TDNN) and *long short-term memory* (LSTM) layers.

In this work, we use the MAP adaptation algorithm [42] in order to adapt the auxiliary SI GMM model. Speaker adaptation of a neural network AM model, built on GMMD features, is performed through the MAP adaptation of the auxiliary GMM model which is used for calculating GMMD features.

3.2. Proposed approach to SAT using transcriptions from ASR

In the standard SAT approach at the training stage, adaptation for each speaker is performed using original (exact) transcriptions from the training corpus. However, at the test time, inaccurate transcriptions from the ASR system are used for adaptation. Hence, speaker adaptive training and decoding are performed in different conditions. Also, it is known that MAP approach is sensitive to the quality of the transcriptions used in adaptation. Both these factors can degrade adaptation performance.

Taking these factors into account, we propose to improve the SAT procedure with GMM-based adaptation framework as shown in Figure 1. The main idea is based on using transcriptions from the ASR system for adaptation of the auxiliary GMM model. This is different from the standard approach, where exact transcriptions are used for adaptation. At the same time, in the proposed SAT scheme, the targets and alignment for training are obtained using the exact transcriptions.

The decoding of the training corpus can be done with a speaker-independent (SI) AM and the LM, which is used in the evaluation experiments. From the practical point of view, in order to obtain more realistic transcriptions for SAT, we should exclude from the training corpus for this AM those data, which this AM will decode. To follow this principle, one solution, that we applied in this paper, is to split the training corpus into two parts: $Train_1$ and $Train_2$ and train two AMs, correspondingly, AM_1 and AM_2. Then, we can use AM_1 to decode $Train_2$ and AM_2 to decode $Train_1$.

The motivation for this approach is to make adaptation more robust to overfitting during the training and to transcription errors during adaptation at the test time.

4. Experimental results

4.1. Data sets

The experiments were conducted on the TED-LIUM corpus [43]. We used the last (second) release of this corpus. This publicly available data set contains 1495 TED talks that amount to 207 hours (141 hours of male, 66 hours of female) speech data from 1242 speakers, 16kHz. For experiments with SAT and adaptation we removed from the original corpus data for those speakers, who had less than 5 minutes of data, and from the rest of the corpus we made three data sets: training, development and test. Characteristics of the obtained data sets are given in Table 1. A more detailed description of data can be found in [40].

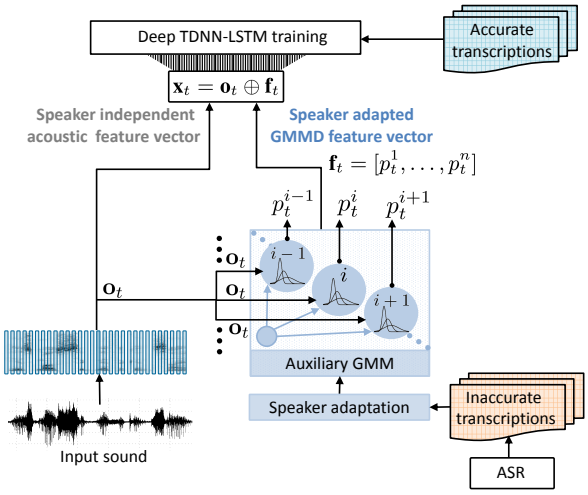

Figure 1: *Speaker adaptive training for a deep TDNN-LSTM AM using MAP-adapted GMMD features and inaccurate transcriptions from ASR for adaptation.*

Table 1: *Data sets statistics*

Characteristic	Train	Dev.	Test
Total duration, hours	171.66	3.49	3.49
Mean duration per speaker, minutes	10	15	15
Number of speakers	1029	14	14
Number of words	-	36672	35555

For evaluation, a 4-gram language model (LM) with 152K word vocabulary was used. The LM is similar to the "small" one, which is currently used in the Kaldi *tedlium s5_r2* recipe. The only difference is that we modified a little a training set, removing from it those data, that present in our test and development sets.

4.2. Baseline system

The open-source Kaldi speech recognition toolkit [44] was used for the experiments presented in this paper. We used TDNN-LSTM model topology described in [41]. 40-dimensional Mel-frequency cepstral coefficients (MFCCs) without cepstral truncation were used as the input into the neural network. Each interleaved TDNN-LSTM model had 9 layers[2] followed by a softmax layer where 3640 triphone states were used as targets. All models were trained using the LF-MMI criterion and with 3-fold reduced frame rate, as in [36].

[2]Following the notation from [41, 45], the model configuration can be described as {-2,-1,0,1,2} {-1,0,1} $\mathcal{L}$ {-3,0,3} {-3,0,3} $\mathcal{L}$ {-3,0,3} {-3,0,3} $\mathcal{L}$, where $\mathcal{L}$ corresponds to a projected LSTM (LSTMP) layer with 512 cells and 128-dimensional recurrent and 128 non-recurrent projections.

Table 2: *Results for SI and SAT TDNN-LSTM AMs trained with LF-MMI (with and without mixup) on the development and test data sets of the TED-LIUM corpus. ΔWER denotes the relative WER reduction with respect to the baseline SI AM. For the improved SAT, the number of AMs in parentheses shows the number of AMs used to decode the training corpus for GMMD feature extraction.*

#	AM	Features	Development		Test	
			WER,%	ΔWER,%	WER,%	ΔWER,%
1	SI	MFCC	11.48	baseline	8.63	baseline
2	SI with mixup	MFCC	10.42	9.2	7.60	11.9
3	SAT	MFCC ⊕ GMMD	10.27	10.5	7.91	8.3
4	Improved SAT (1 AM)	MFCC ⊕ GMMD	9.79	14.7	7.88	8.7
5	Improved SAT (2 AMs)	MFCC ⊕ GMMD	9.71	15.4	7.70	10.8
6	Improved SAT (2 AMs) with mixup	MFCC ⊕ GMMD	**8.70**	**24.2**	**7.26**	**15.9**

4.3. SAT models

4.3.1. Auxiliary GMM

An auxiliary GMM-HMM model was used to calculate GMMD features, as described in Section 3. In this paper, the auxiliary GMM-HMM model had 168 triphone states, with 1-state HMM topology for a triphone, as in [36] and was trained using 3-fold reduced frame rate in order to be consistent with TDNN-LSTM AMs trained with LF-MMI criterion. Adaptation of the auxiliary GMM-HMM model was performed using MAP adaptation algorithm [42].

During the training of SAT TDNN-LSTMs, two scenarios were investigated for speaker adaptation of the auxiliary GMM-HMM models: (1) adaptation using exact transcriptions; and (2) adaptation using inaccurate transcriptions obtained from the decoding of the training corpus. The first scenario is related to the standard SAT approach, and the second one – to the improved SAT approach. In the second scenario, in order to perform decoding, two SI AMs were trained on the subsets of the training corpus, as it is explained in Section 3.2. For comparison purpose, we also performed an additional experiment without splitting the corpus into two parts, when only a single SI baseline AM was used to decode the training corpus and obtain inaccurate transcriptions for adaptation.

4.3.2. SAT TDNN-LSTMs

Three SAT TDNN-LSTM AMs were trained using GMMD features and MAP adaptation, as described in Section 3.

Input features for SAT TDNN-LSTM AMs were 168-dimensional speaker-adapted GMMD features concatenated with conventional 40-dimensional MFCCs without cepstral truncation (the same MFCCs, as were used to train the baseline AM). All SAT TDNN-LSTMs had the same configuration, except for the input layer, as the baseline SI AM, described in Section 4.2, and were trained in the same manner, using LF-MMI criterion and 3-fold reduced frame rate.

The first SAT TDNN-LSTM AM corresponds to the standard SAT approach, described in Section 3.1. The two other SAT TDNN-LSTM AMs correspond to the improved SAT proposed in Section 3.2. They are different from each other in the way inaccurate transcriptions were obtained for adaptation during the SAT training: either with a single baseline AM, or with two AMs trained on two different parts of the training corpus.

4.4. AMs with mixup

Two TDNN-LSTM AMs were trained using the mixup technique described in Section 2. First, mixup was applied to the baseline SI model (Section 4.2). The SI TDNN-LSTM AM with mixup was trained with the same criterion and configuration, as the baseline AM, while applying mixup for LF-MMI, as described in Section 2.1, during the training. Second, to explore the impact of the mixup technique to SAT, we applied mixup during the training of the best SAT TDNN-LSTM AM.

4.5. Analysis of results

The summary of experimental results for TED-LIUM development and test data sets is presented in Table 2 in terms of word error rate (WER). The first two lines of the table correspond to two SI AMs: (#1) the baseline AM described in Section 4.2 and (#2) the AM with mixup. We can see that mixup training provides 9.2–11.9% of relative WER reduction.

The rest of the table is devoted to different SAT AMs. The adaptation experiments were conducted in an unsupervised mode on the test data using transcriptions obtained from the first decoding pass by the SI baseline AM. Line #3 shows the result for the standard SAT, which gives 8.3–10.3% of relative WER reduction with respect to the baseline SI AM. The improved SAT (lines #4 and #5) provides an additional gain in performance in comparison with the standard SAT. Also, as we expected, splitting the training corpus into two parts to obtain inaccurate transcriptions for SAT (as it was described in Section 3.2) gives slightly better results, than using a singe AM to decode all the corpus. Further improvement (about 16-24% of relative WER reduction with respect to the SI baseline) was achieved when mixup was applied during the SAT training.

5. Conclusions

In this work, we have proposed and investigated two different ways to improve SAT for neural network AMs trained using MAP-adapted GMMD features. The first approach is based on using transcriptions from the ASR system for adaptation of the auxiliary GMM model during the SAT. The second approach is related to the mixup training technique. Experiments on the TED-LIUM corpus demonstrated the effectiveness of these methods for state-of-the-art TDNN-LSTM neural network AMs trained with LF-MMI criterion. It was found that SAT of TDNN-LSTM AMs using MAP-adapted GMMD features and the mixup training technique can be complementary to each other, and together provide 16-24% of relative WER reduction with respect to the speaker independent AM.

6. Acknowledgements

This work was partially funded trough the news.bridge project, sponsored as a Google Digital News Innovation project, also it was partially funded by the Government of the Russian Federation (Grant 08-08).

7. References

[1] R. Gemello *et al.*, "Adaptation of hybrid ANN/HMM models using linear hidden transformations and conservative training," in *ICASSP*, 2006, pp. 1189–1192.

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

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

[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 *SLT*. IEEE, 2012, pp. 366–369.

[5] H. Liao, "Speaker adaptation of context dependent deep neural networks," in *ICASSP*. IEEE, 2013, pp. 7947–7951.

[6] D. Yu, K. Yao, H. Su, G. Li, and F. Seide, "KL-divergence regularized deep neural network adaptation for improved large vocabulary speech recognition," in *ICASSP*, 2013, pp. 7893–7897.

[7] D. Albesano, R. Gemello, P. Laface, F. Mana, and S. Scanzio, "Adaptation of artificial neural networks avoiding catastrophic forgetting," in *IJCNN'06*. IEEE, 2006, pp. 1554–1561.

[8] T. Ochiai, S. Matsuda, X. Lu, C. Hori, and S. Katagiri, "Speaker adaptive training using deep neural networks," in *ICASSP*. IEEE, 2014, pp. 6349–6353.

[9] P. Swietojanski and S. Renals, "Learning hidden unit contributions for unsupervised speaker adaptation of neural network acoustic models," in *SLT*. IEEE, 2014, pp. 171–176.

[10] Z. Huang *et al.*, "Feature space maximum a posteriori linear regression for adaptation of deep neural networks," in *INTERSPEECH*, 2014, pp. 2992–2996.

[11] Z. Huang, S. M. Siniscalchi, I.-F. Chen, J. Wu, and C.-H. Lee, "Maximum a posteriori adaptation of network parameters in deep models," in *INTERSPEECH*, 2015, pp. 1076–1080.

[12] S. Li, X. Lu, Y. Akita, and T. Kawahara, "Ensemble speaker modeling using speaker adaptive training deep neural network for speaker adaptation," in *INTERSPEECH*, 2015, pp. 2892–2896.

[13] P. Swietojanski, P. Bell, and S. Renals, "Structured output layer with auxiliary targets for context-dependent acoustic modelling," in *INTERSPEECH*, 2015, pp. 3605–3609.

[14] R. Price, K. i. Iso, and K. Shinoda, "Speaker adaptation of deep neural networks using a hierarchy of output layers," in *SLT*, Dec 2014, pp. 153–158.

[15] P. Karanasou, Y. Wang, M. J. Gales, and P. C. Woodland, "Adaptation of deep neural network acoustic models using factorised i-vectors," in *INTERSPEECH*, 2014, pp. 2180–2184.

[16] V. Gupta, P. Kenny, P. Ouellet, and T. Stafylakis, "I-vector-based speaker adaptation of deep neural networks for French broadcast audio transcription," in *ICASSP*. IEEE, 2014, pp. 6334–6338.

[17] A. Senior and I. Lopez-Moreno, "Improving DNN speaker independence with i-vector inputs," in *ICASSP*, 2014, pp. 225–229.

[18] 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 *ICASSP*, 2013, pp. 7942–7946.

[19] J. Li, J.-T. Huang, and Y. Gong, "Factorized adaptation for deep neural network," in *ICASSP*. IEEE, 2014, pp. 5537–5541.

[20] Y. Qian, T. Tan, and D. Yu, "Neural network based multi-factor aware joint training for robust speech recognition," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 24, no. 12, pp. 2231–2240, Dec 2016.

[21] S. P. Rath *et al.*, "Improved feature processing for deep neural networks." in *INTERSPEECH*, 2013, pp. 109–113.

[22] H. Kanagawa, Y. Tachioka, S. Watanabe, and J. Ishii, "Feature-space structural MAPLR with regression tree-based multiple transformation matrices for DNN," in *APSIPA*, 2015, pp. 86–92.

[23] X. Lei, H. Lin, and G. Heigold, "Deep neural networks with auxiliary Gaussian mixture models for real-time speech recognition," in *ICASSP*. IEEE, 2013, pp. 7634–7638.

[24] S. Liu and K. C. Sim, "On combining DNN and GMM with unsupervised speaker adaptation for robust automatic speech recognition," in *ICASSP*. IEEE, 2014, pp. 195–199.

[25] B. Murali Karthick, P. Kolhar, and S. Umesh, "Speaker adaptation of convolutional neural network using speaker specific subspace vectors of SGMM," in *INTERSPEECH*, 2015, pp. 1096–1100.

[26] S. H. K. Parthasarathi *et al.*, "fMLLR based feature-space speaker adaptation of DNN acoustic models," in *INTERSPEECH*, 2015.

[27] N. Tomashenko and Y. Khokhlov, "Speaker adaptation of context dependent deep neural networks based on MAP-adaptation and GMM-derived feature processing," in *INTERSPEECH*, 2014, pp. 2997–3001.

[28] M. J. Gales, "Maximum likelihood linear transformations for HMM-based speech recognition," *Computer speech and language*, vol. 12, no. 2, pp. 75–98, 1998.

[29] P. C. Woodland, "Speaker adaptation for continuous density HMMs: A review," in *ISCA Tutorial and Research Workshop (ITRW) on Adaptation Methods for Speech Recognition*, 2001.

[30] N. Tomashenko and Y. Khokhlov, "GMM-derived features for effective unsupervised adaptation of deep neural network acoustic models," in *INTERSPEECH*, 2015, pp. 2882–2886.

[31] S. Kundu, K. C. Sim, and M. J. Gales, "Incorporating a generative front-end layer to deep neural network for noise robust automatic speech recognition." in *INTERSPEECH*, 2016, pp. 2359–2363.

[32] K. Nathwani, E. Vincent, and I. Illina, "DNN uncertainty propagation using GMM-derived uncertainty features for noise robust ASR," *IEEE Signal Processing Letters*, 2018.

[33] N. Tomashenko and Y. Estève, "Evaluation of feature-space speaker adaptation for end-to-end acoustic models," in *LREC 2018*, pp. 3163–3170.

[34] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "mixup: Beyond empirical risk minimization," *arXiv preprint arXiv:1710.09412*, 2017.

[35] Y. Tokozume, Y. Ushiku, and T. Harada, "Learning from between-class examples for deep sound recognition," *arXiv preprint arXiv:1711.10282*, 2017.

[36] D. Povey *et al.*, "Purely sequence-trained neural networks for ASR based on lattice-free MMI." in *INTERSPEECH*, 2016.

[37] L. Bahl, P. Brown, P. De Souza, and R. Mercer, "Maximum mutual information estimation of hidden Markov model parameters for speech recognition," in *ICASSP*. IEEE, 1986, pp. 49–52.

[38] D. Povey, D. Kanevsky, B. Kingsbury, B. Ramabhadran, G. Saon, and K. Visweswariah, "Boosted MMI for model and feature-space discriminative training," in *ICASSP*. IEEE, 2008, pp. 4057–4060.

[39] Medennikov *et al.*, "An investigation of mixup training strategies for acoustic models in ASR," *Accepted for INTERSPEECH*, 2018.

[40] N. Tomashenko *et al.*, "On the use of Gaussian mixture model framework to improve speaker adaptation of deep neural network acoustic models," in *INTERSPEECH*, 2016, pp. 3788–3792.

[41] V. Peddinti, Y. Wang, D. Povey, and S. Khudanpur, "Low latency acoustic modeling using temporal convolution and lstms," *IEEE Signal Processing Letters*, vol. 25, no. 3, pp. 373–377, 2018.

[42] J.-L. Gauvain and C.-H. Lee, "Maximum a posteriori estimation for multivariate gaussian mixture observations of Markov chains," *IEEE Trans. Speech and Audio Proc.*, vol. 2, pp. 291–298, 1994.

[43] A. Rousseau, P. Deléglise, and Y. Estève, "Enhancing the TED-LIUM corpus with selected data for language modeling and more TED talks." in *LREC*, 2014, pp. 3935–3939.

[44] D. Povey *et al.*, "The Kaldi speech recognition toolkit," in *ASRU*, 2011.

[45] V. Peddinti, D. Povey, and S. Khudanpur, "A time delay neural network architecture for efficient modeling of long temporal contexts." in *INTERSPEECH*, 2015, pp. 3214–3218.