INTEGRATING GAUSSIAN MIXTURES INTO DEEP NEURAL NETWORKS: SOFTMAX LAYER WITH HIDDEN VARIABLES

Zoltán Tüske[a], Muhammad Ali Tahir[a], Ralf Schlüter[a], Hermann Ney[a,b]

[a] Human Language Technology and Pattern Recognition, Computer Science Department, RWTH Aachen University, 52056 Aachen, Germany
[b]Spoken Language Processing Group, LIMSI CNRS, Paris, France
{tuske, tahir, schlueter, ney}@cs.rwth-aachen.de

ABSTRACT

In the hybrid approach, neural network output directly serves as hidden Markov model (HMM) state posterior probability estimates. In contrast to this, in the tandem approach neural network output is used as input features to improve classic Gaussian mixture model (GMM) based emission probability estimates. This paper shows that GMM can be easily integrated into the deep neural network framework. By exploiting its equivalence with the log-linear mixture model (LMM), GMM can be transformed to a large softmax layer followed by a summation pooling layer. Theoretical and experimental results indicate that the jointly trained and optimally chosen GMM and bottleneck tandem features cannot perform worse than a hybrid model. Thus, the question „hybrid vs. tandem" simplifies to optimizing the output layer of a neural network. Speech recognition experiments are carried out on a broadcast news and conversations task using up to 12 feed-forward hidden layers with sigmoid and rectified linear unit activation functions. The evaluation of the LMM layer shows recognition gains over the classic softmax output.

Index Terms— Log-linear, mixture model, LMM, GMM, DNN, bottleneck, neural network, hybrid, tandem, ASR

1. INTRODUCTION

Deep neural networks (DNN) have become an essential part of the acoustic model of recent automatic speech recognition (ASR) systems. Estimating state posterior probabilities directly, the hybrid approach has shown enormous gains over old-fashioned GMMs trained only on cepstral features. GMMs can be improved when trained on the output or a hidden layer of a neural network (NN) which effectively results in two acoustic models in tandem. In state-of-the-art systems the two approaches perform head-to-head in large vocabulary ASR tasks. So far only few works addressed the joint training of the tandem models. Therefore, this paper focuses on a consistent tandem approach. As will be shown, the GMM can be easily integrated into the DNN framework through a generalized softmax layer, which could also indicate why the two approaches can achieve similar performance.

The NN based HMMs were proposed in the early 90's [1]. It has been also discovered that training of the hybrid models with tens of thousands of allophone targets is still feasible but only with low-rank factorization of the last layer [2],[3]. This can be achieved by a linear bottleneck (BN) layer. The probabilistic tandem approach was introduced in [4], and was extended by the bottleneck concept proposed by [5]. Linear BN features for tandem approach were investigated in [6]. Previous work in [7] also addressed the joint training of GMM and shallow BN features using the (sequence) MMI criterion. The author calculated the derivatives of the error function w.r.t. GMM parameters directly and applied the chain rule to update the GMM simultaneously with the BN features. In this paper, we follow a different approach to integrate the GMM into DNN. The work of [8] and [9] showed that log-linear model with quadratic features corresponds with Gaussian model. Similarly, GMM and log-linear mixture model (LMM) are also equivalent [10]. Since the softmax layer of the NN is a log-linear model, its substitution with an LMM is a natural way of integrating GMM into the DNNs.

Therefore, in this work we exploit that GMMs with pooled covariance matrix can be easily transformed to LMM by already well-known neural network elements: linear, softmax, and sum- or max-pooling layer. The joint training of BN and GMM is then addressed through a generalized softmax layer and compared to various hybrid models.

The paper is organized as follows. In Section 2 we give a short overview of the log-linear mixture modeling and its relation to GMM, further, the softmax layer with hidden variables is also introduced. Section 3 gives details about our experimental setup. Experimental results are presented in Section 4. The paper closes with conclusions in Section 5.

2. THE LOG-LINEAR MIXTURE LAYER

2.1. The log-linear models

Log-linear models are discriminative models directly estimating posterior probabilities of class s given the feature vector $x \in \mathbb{R}^{\mathrm{M}}$:

$$p_\theta(s|x) = \frac{\exp\left(w_s^T f\left(x\right) + b_s\right)}{\sum\limits_{s'} \exp\left(w_{s'}^T f\left(x\right) + b_{s'}\right)} \tag{1}$$

with model parameters $\theta = \{w_s, b_s\}$, where $w_s \in \mathbb{R}^{\mathrm{N}}$ and $b_s \in \mathbb{R}$ are state specific parameters. The $f(x) : \mathbb{R}^{\mathrm{M}} \rightarrow \mathbb{R}^{\mathrm{N}}$ corresponds to the feature function such as linear, polynomial or any non-linear feature mapping, e.g. another tandem model [11, 12, 13, 14, 15]. Within the neural network framework Eq. 1 corresponds to the softmax output layer: w_s, b_s form the last weight matrix and bias vector, the rest of the network up to the output of the last hidden layer forms the feature function f. In HMM-based ASR the estimated posterior probabilities are transformed to likelihood via the Bayes-rule (hybrid approach) [1].

The model parameters are often trained by maximizing the empirical conditional likelihood (also known as cross entropy (CE) or maximum mutual information). In this paper our targets s are tied-triphone states and we use a frame-wise form of the criterion. The

alignment between the feature vectors and the HMM states is kept fixed.

A major advantage of log-linear models is that the above criterion is convex. By applying the chain rule deep neural network basically trains the classification layer and feature function jointly, making the optimization non-convex.

As has been shown, the posterior form of Gaussian model results in a log-linear model with linear and quadratic feature functions and vice versa [9]. In a special case, if classes share the same covariance matrix Σ only linear features remain. Thus, in the NN framework the softmax layer is equivalent to a discriminatively trained single Gauss model with pooled covariance matrix:

$$p_\theta(s|y) = \frac{p(s)\mathcal{N}(y|\mu_s, \Sigma)}{\sum_{s'} p(s')\mathcal{N}(y|\mu_{s'}, \Sigma)} = \frac{\exp(w_s^T y + b_s)}{\sum_{s'} \exp(w_{s'}^T y + b_{s'})} \quad (2)$$

where y corresponds to the observation vector e.g. LDA transformed MFCC, $\mathcal{N}(y|\mu_s, \Sigma)$ denotes the normal distribution of y in class s with mean vector μ_s, and the pooled full covariance matrix is Σ. For the conversion of the generative model parameters to the log-linear parameters we refer to [10], and see the generalized case below.

2.2. The log-linear mixture models

Log-linear model with hidden variables is also called log-linear mixture model (LMM):

$$p_\theta(s|x) = \sum_i p_\theta(s, i|x) = \frac{\sum_i \exp\left(w_{si}^T f(x) + b_{si}\right)}{\sum_{s',i'} \exp\left(w_{s'i'}^T f(x) + b_{s'i'}\right)} \quad (3)$$

where i is the hidden variable, w_{si} and b_{si} denotes the hidden state and class dependent parameters. It has also been shown that the posterior form of GMM with a given class prior distribution corresponds to this model [16, 10]. Again, a covariance matrix Σ globally shared between all mixture components results only in linear features:

$$\frac{p(s)\sum_i p(i|s)\mathcal{N}(y|\mu_{si}, \Sigma)}{\sum_{s'} p(s')\sum_i p(i|s')\mathcal{N}(y|\mu_{s'i}, \Sigma)} = \frac{\sum_i \exp(w_{si}^T y + b_{si})}{\sum_{s',i} \exp(w_{s'i}^T y + b_{s'i})} \quad (4)$$

where $p(i|s)$ denotes the mixture component weights and μ_{si} is the mean vector of the ith Gaussian mixture component of state s. The conversion from GMM to LMM can be performed with the following equations:

$$\begin{aligned} b_{si} &= -\frac{1}{2}\mu_{si}^T \Sigma \mu_{si} + \ln\ p(s) + \ln\ p(i|s) \\ w_{si} &= \Sigma^{-1}\mu_{si} \end{aligned} \quad (5)$$

During the ASR decoding process the logarithm of the acoustic scores is accumulated. Using maximum approximation enables the fast computation of the GMM log-likelihood [17]. Because the hidden variable in Eq. 3 simply corresponds to the mixture index, the approximation is equal to finding the maximum term of the numerator.

2.3. Softmax layer with hidden variable

Grouping the parameters of a state, Eq. 3 can be realized by already existing NN building blocks as a softmax layer followed by a sum-pooling over a region. In the case of maximum approximation the last layer becomes a max-pooling. The softmax layer with hidden variables is more general than the classic output layer employed in current NNs. In the tandem approach, the feature function $f(x)$ is equal to the BN feature extraction and the GMM are trained on $y \doteq f(x)$. Transforming a GMM (pretrained with e.g. maximum-likelihood criterion) to a LMM allows a natural way to train the acoustic model and BN features jointly. Figure 1 shows the proposed NN hierarchy for a consistent tandem approach. As can be seen, a linear BN tandem GMM can be easily converted to a very similar structure proposed in [2].

Using the maximum approximation, it is possible to train the model with the Expectation-Maximization (EM) algorithm: fixing the mixture index i for each observation x the maximization step simplifies to a classic neural network training. During the expectation step the observation is realigned to mixture index i. The EM steps of NN with hidden variables can also be combined with the EM steps of the HMM models. NNs are usually trained by stochastic gradient (SGD) methods even if the activation functions are non-differentiable on a finite set of points, like rectified linear units (ReLU), or max-pooling [18, 19]. Therefore, we limited our investigation only to direct training of the model with SGD. The number of hidden states and the initial parameters were determined by maximum likelihood (ML) estimated GMM according to Eq. 5. Thus, the training of a bottleneck Multi Layer Perceptron (BN-MLP) and the GMM can be considered as pretraining steps of a more complex MLP. The training of the generalized softmax-layer from scratch e.g. with discriminative splitting algorithm [20] was not investigated.

In preliminary experiments we observed that the state posterior distribution obtained by ML-GMM, especially by models with higher number of densities, are much sharper than the output of CE trained models. In order to fit the generative model parameters to the CE criterion the posteriors should be smoothed with $\alpha < 1$ before DNN-training:

$$w_{si} \rightarrow \alpha \cdot w_{si} \qquad b_{si} \rightarrow \alpha \cdot b_{si} \quad (6)$$

In the case of LMM with maximum approximation this is equivalent to $p_\theta(s|x)^\alpha$ and does not influence the classification accuracy (see Figure 2).

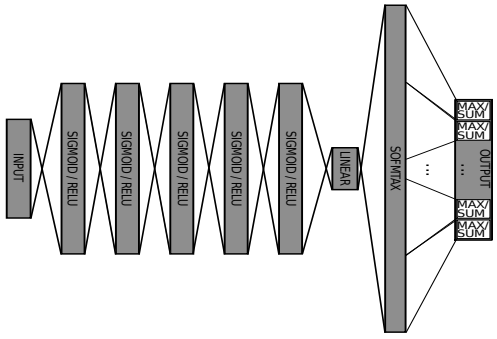

Fig. 1. *The proposed neural network structure for a consistent tandem BN and GMM training: softmax layer with hidden variables.*

It should be mentioned that the softmax layer is much larger due to the hidden variable, e.g. a GMM trained on 4500 tied states results in over 1 million (!) nodes after 8 splits. Therefore, an efficient GPU implementation of the softmax function is crucial. Furthermore, due to limited available memory on GPU, the usual mini-batch was also processed in sub-batches wherever it was possible. Because of the huge softmax layer, the low-rank factorization of the last weight ma-

trix through linear BN layer is inevitable as it was proposed also for NN with more than 10k outputs [2].

For decoding, the last layer can be transformed back to a GMM. And beside the maximum approximation, further speed-up in acoustic score calculation can be expected after applying density preselection methods.

3. EXPERIMENTAL SETUP

The proposed way of training BN features and GMM jointly was investigated on a broadcast news and conversation task. Similar to [21], the training of the GMM and DNN acoustic models was carried out on a Viterbi alignment generated by our best performing evaluation system. For our research purpose we defined a 50-hour subset of the full corpus. The corpus statistics can be found in Table 1. The recognition lexicon contained 150k words. For further details about the task and language model estimation we refer to [22] and [23].

Unless otherwise stated, fast-VTLN and segmentwise mean-and-variance normalized 16-dimensional MFCCs are extracted [24]. The GMMs used pooled, diagonal covariance matrix and were trained with ML criterion using maximum approximation. The number of parameters roughly doubled after each split. The MLP and GMM model 4500 tied context-dependent triphone states. As input, 17 frames of MFCC are fed into the MLP. The MLP consists of 6 non-BN hidden-layers with 2000 nodes each.

During the optimization of framewise cross-entropy (CE) objective function by SGD, the mini-batch size was fixed to 512 frames. The networks were initialized by discriminative pretraining according to [25]. Momentum term and l_2 regularization were applied only with ReLU MLPs. In order to prevent overfitting and adjust the learning rate, 10% of the training corpus was selected for cross-validation (CV). To control the training procedure, we used a slightly modified newbob learning rate scheduling strategy. The learning rate was kept fixed as long as the frame error rate (FER) improved by at least 0.1%. In the subsequent epochs the learning rate was halved until the FER improvement was less than 0.1%. In addition, the ramping state was reset if the improvement was over 0.15%. Since the GMM model was later merged into the DNN through the LMM conversion, the CV set was discarded during the ML-GMM training.

The softmax layer with hidden variables was always initialized by our GMM. For this purpose hybrid models with linear BN right before the output were also trained. Then, the GMMs were trained only on this BN features without any further processing (PCA or LDA, windowing) or concatenation with cepstral features to keep the structure of acoustic models consistent. The ultimative comparison of the hybrid and the proposed consistent tandem approach after sequence discriminative training is not addressed here.

4. EXPERIMENTAL RESULTS

Before the joint training of the GMM and BN features the model conversion was studied. According to our observation α is inversely proportional to the BN size. In addition to the smoothing (Eq. 6),

Table 1. *Corpus statistics*

corpus		words [K]	frames [M]	hours	LM ppl.	OOV rate[%]
train	small	500	15.3	50		
	large	2700	75.2	250		
dev		41	1.3	3.7	123	0.4
eval		35	1.2	3.3	136	0.4

Table 2. *Baseline sigmoid and ReLU 6-hidden-layer hybrid systems with and without linear BN. Results are in WER [%].*

		Sigmoid					ReLU				
Train	BN size	64	128	256	512	-	64	128	256	512	-
small	dev	19.2	**18.9**	19.0	19.4	19.4	18.2	18.1	17.9	17.8	**17.7**
	eval	24.9	**24.6**	24.7	25.3	25.3	24.5	23.8	23.8	23.6	**23.5**
large	dev	15.8	15.6	**15.4**	15.5	15.6	15.9	15.6	15.7	**15.4**	15.7
	eval	20.7	20.8	**20.6**	20.7	20.8	21.2	20.8	21.2	**20.7**	20.9

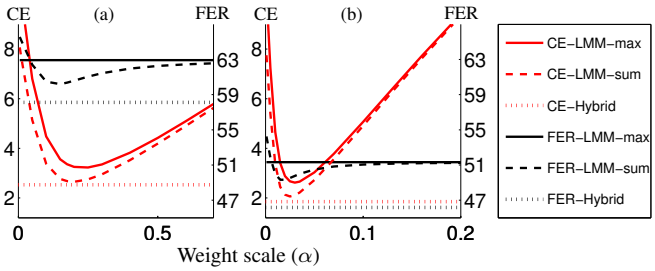

Fig. 2. *The effect of scaling the weight of the ML-GMM initialized LMM. Frame error rate (FER [%]) and cross-entropy (CE) were measured on the CV set. The GMM with 16 components/state was trained on BN-MLP features. (a) 64-dimensional sigmoid BN features (small scale); (b) 512-dimensional ReLU bottleneck features (large scale).*

the state and mixture priors should also be rescaled to better fit the GMM-initialized LMM to the CE criterion. As can be seen in Fig. 2, the ML trained GMM can achieve a similar objective function value, but performs few percents relative worse than the discriminative (single Gauss) hybrid model. The gap increased with more data and complex models but could be reduced with more splits. Furthermore, the exact model (sum-pooling) achieved clearly better frame accuracy and objective function value than the model with maximum approximation. Although the smoothing was not always necessary – could be learned by the network within the first epochs –, without it we often ended up in convergence problem with larger GMMs.

Table 2 shows the baseline hybrid systems with different degree of low-rank factorization of the last matrix. Besides the parameter reduction, the insertion of linear BN before the softmax layer always resulted in a gain with sigmoid NN. Applying low-rank factorization with ReLU, we observed degradation on the small training set.

4.1. Small scale experiments

In the first experiments we investigated the joint training of BN-GMM/LMM with different feature size and complexity (Table 3). First experiments were carried out with usual BN size (64 dim) (rows 1-9). The joint training of BN-GMM was tested with single Gaussian (split-0) due to its model equivalence to our baseline (column 1 in Table 2). As should be expected, the CE training resulted in almost the same performance (row 2 of Table 3). Increasing the GMM complexity, we measured the best result after ML training with split-8 model (row 7). However, after the discriminative training much less components per state were sufficient, split-4 model showed the lowest WER (row 6). Table 3 also shows that CE training of the exact model always resulted in better WER than using maximum approximation (row 4↔5 or 8↔9). We obtained 1% absolute WER improvement over the best ML model after the CE based joint model training. Training only the LMM layer, the best result was achieved again by a split-4 model and reached 19.1% WER (row 4).

Table 3. *Recognition results with classic and jointly trained tandem GMM and BN features (50h). Maximum approximation (+) applied optionally during the training or recognition.*

	BN size	BN-GMM/LMM split	Joint training	Training criterion	Max. approx. train	Max. approx. recog.	WER [%] dev	WER [%] eval
Sigmoid	64	0	no	ML			22.4	28.7
			yes	CE			19.2	24.8
		4	no	ML	+	+	20.2	26.3
				CE			19.1	24.7
			yes		+	+	18.9	24.8
							18.5	24.2
		8	no	ML	+	+	19.5	25.4
			yes	CE	+	+	19.2	25.2
							19.0	24.9
	128	4	no	ML	+	+	20.8	26.6
			yes	CE	+	+	18.9	24.7
							18.3	24.0
						+	**18.1**	**23.9**
		8	no	ML	+	+	20.2	25.8
ReLU	256	2	no	ML	+	+	19.6	25.8
			yes	CE	+	+	18.1	24.0
							17.5	23.0
		5	no	ML	+	+	18.5	24.4
			yes	CE			17.4	23.3
						+	**17.2**	**22.9**
		8	no	ML	+	+	18.7	24.3

This result indicates that the joint training of BN and GMM/LMM is necessary. Further experiments were carried out with larger, 128-dimensional BN (rows 10-14) features. In contrast to the ML, after the joint training we obtained a better performance (row 12). Application of the maximum approximation only during the recognition led to some recognition gains (row 13). Compared to the best sigmoid baseline result in Table 2, our MLP with the generalized softmax layer outperformed our baseline by 0.7% absolute in WER.

The experiments with ReLU were carried out only with BN of 256 nodes, and confirmed the observations made with sigmoid MLPs (rows 15-21). The consistent CE training of ReLU BN-MLP reached about 0.5% absolute lower WER compared to the corresponding result in Table 2. For further comparison, we also optimized the output of the hybrid systems using 256 dimensional BN: a MLP with 12k outputs resulted in 17.5% WER on the development set.

4.2. Large scale experiments

Using 250 hours of training data, the joint BN-GMM training was investigated with two types of MLP. In order to measure the effect of more data, first a 6-hidden-layer sigmoid MLP was trained.

Table 4. *Recognition results with classic and jointly trained tandem GMM and BN features (250h).*

	BN size	BN-GMM/LMM split	Joint training	Training criterion	Max. approx. train	Max. approx. recog.	WER [%] dev	WER [%] eval
Sigmoid	64	5	no	ML	+	+	17.3	22.8
			yes	CE			15.3	20.6
						+	15.4	20.4
		10	no	ML	+	+	16.4	21.3
	256	5	no	ML	+	+	17.9	23.6
			yes	CE			15.0	20.2
						+	**14.9**	**20.2**
		8	no	ML	+	+	16.8	22.1

Table 5. *Comparison of hybrid and jointly trained BN-GMM systems using 12-layer ReLU MLP.*

System	#output	split	criterion	WER [%] dev	WER [%] eval
Hybrid	4.5k	-	CE	13.3	18.1
+low-rank output				13.5	18.2
	12.0k			**13.0**	**17.7**
BN tandem	4.5k	8	ML	14.2	19.0
+joint training		4	CE	**13.1**	**17.8**

According to Table 2, the sigmoid MLP performed best with 256-dimensional BN layer. Although the ML trained high dimensional BN tandem performed significantly worse than a lower dimensional features (row 1↔5, 4↔8 in Table 4), we focused on the results after the CE based joint training. Similar to small scale, the larger BN pays off after CE training (row 2 and 6), and maximum approximation does not hurt the recognition performance (row 3 and 7).

Aiming at the best WER, the second set of large scale experiments was carried out with a 12-layer ReLU MLP trained on 19 frames of 50-dimensional GT features [26]. About 2% absolute improvement is attributable to these modification steps. Again, hybrid systems with optimized output size (up to 25k) were also compared to our jointly trained BN-GMM system. The results are summarized in Table 5. The generalized softmax (jointly trained BN-GMM) outperformed our very strong hybrid baseline. The hybrid system achieved slightly better results with only optimized output size (12k). However, the increased output size could be also applied on the BN tandem system, and also the split size was not optimized at this level of NN complexity. Although on large scale the consistent tandem and the hybrid models performed equally, the hidden variables within the model allowed less target states with tandem approach, thus, could result in a smaller search space.

5. CONCLUSIONS

We have shown that the tandem approach can be considered as a softmax layer with hidden variables. The integration of the GMM into the DNN framework results in a deep and wide structure. As has been demonstrated, the BN-MLP training and tandem approach can simply be considered as an initialization step of this more complex model. On small scale, the joint training of tandem BN-GMM through generalized softmax layer always resulted in better recognition performance than any of our hybrid baselines. Furthermore, large scale experiments verified that the proposed BN-LMM model with hidden variables could achieve similar performance with fewer output targets than a classic hybrid system.

Since our current best tandem systems are built on hierarchical BN structures [27], in the future work our research will be extended to more complex BN features. The effect of sequence discriminative training will be also investigated.

Acknowledgement: The study 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-0012. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. Disclaimer: The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of IARPA, DoD/ARL, or the U.S. Government. H. Ney was partially supported by a senior chair award from DIGITEO.

6. REFERENCES

[1] H. A. Bourlard and N. Morgan, *Connectionist Speech Recognition: A Hybrid Approach*. Norwell, MA, USA: Kluwer Academic Publishers, 1993.

[2] T. N. Sainath, B. Kingsbury, V. Sindhwani, E. Arisoy, and B. Ramabhadran, "Low-rank matrix factorization for deep neural network training with high-dimensional output targets," in *ICASSP*, 2013, pp. 6655–6659.

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

[4] H. Hermansky, D. P. Ellis, and S. Sharma, "Tandem connectionist feature extraction for conventional HMM systems," in *ASRU*, vol. 3, 2000, pp. 1635–1638.

[5] F. Grézl, M. Karafiát, S. Kontár, and J. Černocký, "Probabilistic and bottle-neck features for LVCSR of meetings," in *ICASSP*, 2007, pp. 757–760.

[6] K. Veselý, M. Karafiát, and F. Grézl, "Convolutive Bottleneck Network Features for LVCSR," in *ASRU*, 2011, pp. 42–47.

[7] M. Paulik, "Lattice-based training of bottleneck feature extraction neural networks," in *Interspeech*, 2013, pp. 89–93.

[8] J. Anderson, "Logistic discrimination," in *Handbook of statistics*, vol. 2. North-Holland, 1982, pp. 169–191.

[9] G. Heigold and R. Schlüter, "On the Equivalence of Gaussian HMM and Gaussian HMM-like Hidden Conditional Random Fields," in *Interspeech*, 2007, pp. 1721–1724.

[10] G. Heigold, "A Log-Linear Discriminative Modeling Framework for Speech Recognition," Ph.D. dissertation, RWTH Aachen University, Aachen, 2010.

[11] S. Wiesler, M. Nußbaum-Thom, G. Heigold, R. Schlüter, and H. Ney, "Investigations on features for log-linear acoustic models in continuous speech recognition," in *ASRU*, 2009, pp. 52–57.

[12] G. Heigold, G. Zweig, X. Li, and P. Nguyen, "A flat direct model for speech recognition," in *ICASSP*, 2009, pp. 3861–3864.

[13] Y. Kubo, S. Wiesler, R. Schlüter, H. Ney, S. Watanabe, A. Nakamura, and T. Kobayashi, "Subspace pursuit method for kernel-log-linear models," in *ICASSP*, 2011, pp. 4500–4503.

[14] M. Tahir, H. Huang, R. Schlüter, H. Ney, L. ten Bosch, B. Cranen, and L. Boves, "Training log-linear acoustic models in higher-order polynomial feature space for speech recognition." in *Interspeech*, 2013, pp. 3352—-3355.

[15] K. Demuynck and F. Triefenbach, "Porting concepts from DNNs back to GMMs," in *ASRU*, 2013, pp. 356–361.

[16] L. Saul and D. Lee, "Multiplicative updates for classification by mixture models," in *NIPS*, vol. 2, 2001, pp. 897–904.

[17] S. Kanthak, K. Schütz, and H. Ney, "Using SIMD instructions for fast likelihood calculation in LVCSR," in *ICASSP*, 2000.

[18] X. Glorot, A. Bordes, and Y. Bengio, "Deep Sparse Rectifier Neural Networks," in *AISTATS*, 2011, pp. 315–323.

[19] I. J. Goodfellow, D. Warde-Farley, M. Mirza, A. Courville, and Y. Bengio, "Maxout networks," in *ICML*, 2013.

[20] M. Tahir, R. Schlüter, and H. Ney, "Discriminative splitting of Gaussian/log-linear mixture HMMs for speech recognition," in *ASRU*, 2011, pp. 7–11.

[21] S. Wiesler, A. Richard, P. Golik, R. Schlüter, and H. Ney, "RASR/NN: The RWTH neural network toolkit for speech recognition," in *ICASSP*, 2014, pp. 3305–3309.

[22] M. Nußbaum-Thom, S. Wiesler, M. Sundermeyer, C. Plahl, S. Hahn, R. Schlüter, and H. Ney, "The RWTH 2009 QUAERO ASR evaluation system for English and German," in *Interspeech*, 2010, pp. 1517–1520.

[23] M. Sundermeyer, M. Nußbaum-Thom, S. Wiesler, C. Plahl, A.-D. Mousa, S. Hahn, D. Nolden, R. Schlüter, and H. Ney, "The RWTH 2010 QUAERO ASR Evaluation System for English, French, and German," in *ICASSP*, 2011, pp. 2212–2215.

[24] L. Welling, H. Ney, and S. Kanthak, "Speaker adaptive modeling by vocal tract normalization," *IEEE Trans. on Speech and Audio Processing*, vol. 10, no. 6, pp. 415–426, Sep. 2002.

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

[26] R. Schlüter, I. Bezrukov, H. Wagner, and H. Ney, "Gammatone features and feature combination for large vocabulary speech recognition," in *ICASSP*, 2007, pp. 649–652.

[27] Z. Tüske, R. Schlüter, and H. Ney, "Deep hierarchical bottleneck MRASTA features for LVCSR," in *ICASSP*, 2013, pp. 6970–6974.