SPEAKER ADAPTIVE JOINT TRAINING OF GAUSSIAN MIXTURE MODELS AND BOTTLENECK FEATURES

Zoltán Tüske, Pavel Golik, Ralf Schlüter, Hermann Ney

Human Language Technology and Pattern Recognition, Computer Science Department,
RWTH Aachen University, 52056 Aachen, Germany
{tuske, golik, schlueter, ney}@cs.rwth-aachen.de

ABSTRACT

In the tandem approach, the output of a neural network (NN) serves as input features to a Gaussian mixture model (GMM) aiming to improve the emission probability estimates. As has been shown in our previous work, GMM with pooled covariance matrix can be integrated into a neural network framework as a softmax layer with hidden variables, which allows for joint estimation of both neural network and Gaussian mixture parameters. Here, this approach is extended to include speaker adaptive training (SAT) by introducing a speaker dependent neural network layer. Error backpropagation beyond this speaker dependent layer realizes the adaptive training of the Gaussian parameters as well as the optimization of the bottleneck (BN) tandem features of the underlying acoustic model, simultaneously. In this study, after the initialization by constrained maximum likelihood linear regression (CMLLR) the speaker dependent layer itself is kept constant during the joint training. Experiments show that the deeper backpropagation through the speaker dependent layer is necessary for improved recognition performance. The speaker adaptively and jointly trained BN-GMM results in 5% relative improvement over very strong speaker-independent hybrid baseline on the Quaero English broadcast news and conversations task, and on the 300-hour Switchboard task.

Index Terms— MLP, GMM, log-linear mixture model, joint-training, unsupervised adaptation, CMLLR, SAT

1. INTRODUCTION

Neural networks estimating directly the context-dependent phone posterior probabilities have become the state-of-the-art acoustic models in the field of automatic speech recognition (ASR) [1, 2]. In the tandem approach, being another advanced modeling technique, classic Gaussian mixture models are trained on the output or bottleneck (BN) hidden layer activation of a neural network [3, 4]. On large vocabulary ASR tasks, the hybrid and tandem approaches usually perform head-to-head [5].

Although the latter technique ends up in two acoustic models working in tandem, a large variety of well established adaptation methods exist for GMMs. They aim at decreasing the variance between speakers and coping with acoustic mismatch between training and test conditions. For instance, to compensate for the frequency shifts of formants related to differing vocal tract sizes of different speakers, the amplitude spectrum is often calculated using a warped frequency scale. The warping factor is estimated utterance-wise by maximizing the likelihood either under a text-dependent or a universal Gaussian acoustic model [6, 7]. If more adaptation data are available, an affine transformation of the speaker independent (SI) GMM parameters can be estimated, a method known as maximum likelihood linear regression (MLLR) [8, 9]. A constrained version of the MLLR transform (CMLLR) can be carried out in the feature space [10]. The CMLLR matrices allow the speaker adaptive training (SAT) of the underlying model. Besides that they can already be estimated robustly on few minutes of data in an *unsupervised* way. For large amounts of data the maximum a posteriori estimation of the model parameters can improve the adapted models [11].

Although the inherent robustness of NNs against unwanted variability – e.g. speaker variation – is well known in the literature [12, 13, 14], the recognition performance can still drop significantly on acoustically mismatched data [15]. Since discriminative approaches are more sensitive to recognition errors and state-of-the-art models have a large amount of trainable parameters, a lot of (mostly supervised) data is required [16, 17] to directly adapt the input [18], output [19], hidden [20, 21], or all layers. The deviation of the weights from the original SI model [16] or the difference in the posterior output [22] needs to be controlled by a careful regularization. This is especially important for adaptation with limited amount of data, even in case of supervised adaptation. Therefore limiting the update to a low number of parameters, such as linear transformation of a narrow BN layer [23] or sharing the transformation between time frames [24, 25], can make the adaptation more effective. Thus, the robust maximum likelihood (ML) feature-space adaptation techniques developed for GMMs are still very often used with NN based acoustic models [26, 25]. For instance it is possible to perform the CMLLR adaptation of each BN layer, when training hierarchical BN structures [27, 28, 29].

Recent adaptation methods for NNs also include speaker aware training, where speaker related information is directly fed as a supplementary feature vector into the network. This results in a speaker dependent bias in each layer the speaker vector is presented to. The speaker dependent input can either be learned by the network (e.g. speaker code [30]) or derived from auxiliary models, like i-vector [31, 32, 33].

In the tandem acoustic modeling the final GMM is trained in a natural way on CMLLR transformed BN features. Therefore this

978-1-4799-7291-3/15/$31.00 ©2015 IEEE

paper investigates the possibility of training the speaker adapted BN features and the GMM *jointly*. Several previous studies addressed the simultaneous training of the tandem models [34, 35]. Our recent investigation in [36] demonstrated that a GMM with a pooled covariance matrix can be easily implemented in the NN framework as a softmax layer with hidden variables, a more general hybrid output layer. Results also indicate that a properly tuned and jointly trained tandem model can achieve equal or better error rates compared to a standard hybrid models using the same number of output classes. Following a similar principle as in [37], in this study we extend our previous work by a speaker adaptive layer which enables the joint training of the adapted BN-GMM. We will show that the CMLLR adaptation of a tandem BN-GMM system can be interpreted as BN layer adaptation of a low-rank factorized output layer similar to [23]. The initialization of the speaker dependent layer with CMLLR matrices leads then to a robust unsupervised adaptation scheme. By backpropagating the error signal through this layer without updating it, the network is forced to use the CMLLR adaptation matrices. It is also forced to adjust the rest of the parameters shared between all speakers to minimize the discriminative error criterion, effectively training both the BN feature extraction and the "GMM layer". During the recognition we estimate the CMLLR transformations as usual on the first-pass output obtained with an SI model. The improved results demonstrate that neither does the speaker adaptively and jointly trained BN-GMM overfit the training matrices, nor does it suffer too much from recognition errors in the CMLLR estimation.

The paper is organized as follows. Section 2 presents how the speaker adapted GMM can be integrated into the deep neural network (DNN) framework. It shortly overviews the relation between a GMM and a softmax layer with hidden variables. After introducing a speaker adaptive layer we address the joint training of the proposed model. Section 3 and Section 4 give details about our experimental setups and present recognition results on two different tasks. The paper closes with conclusions in Section 5.

2. INTEGRATION OF SPEAKER ADAPTIVE GMM INTO DNN

2.1. The log-linear mixture models

It has been shown that the posterior form of a Gaussian mixture model is a log-linear model with hidden variables, also referred to as log-linear mixture model (LMM), where the feature functions are linear and quadratic [38, 39]. A GMM with a globally shared covariance matrix Σ results in a simplified LMM which has only linear feature functions:

$$p_\theta(s|x) = \sum_i p_\theta(s,i|x) = \tag{1}$$

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

where $p_\theta(s|x)$ corresponds to the estimated posterior probability of state s given the observation x and model parameters $\theta = \{w_{si}, b_{si}\}$. We assume that the input feature x is transformed e.g. by a BN multi-layer perceptron (MLP) such that $f(x) = y$. Index i denotes the hidden variable, w_{si} and b_{si} are the hidden state and output class dependent parameters. $\mathcal{N}(y|\cdot,\Sigma)$ stands for the normal distribution. The mixture component weights are denoted as $p(i|s)$, and μ_{si} is the mean vector of the ith Gaussian mixture component of state s. The conversion from the GMM to the LMM parameters is defined by the following equations:

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

If we move the summation out of the numerator in Eq. 1, the model can be represented by well-known building blocks of a neural network, such as linear, softmax, and non-overlapping sum-pooling layers. The LMM estimates class posterior probabilities directly, and in a special case without hidden variables the model corresponds to a conventional softmax layer. Due to hidden variables the softmax layer is much larger than in a common hybrid acoustic model. E.g. a model with 9000 HMM states and 32 Gaussians per state results in 144k nodes in this layer. The sum in the numerator of Eq. 1 can be approximated by the maximum term (max-pooling). This enables a faster and more stable log-likelihood acoustic score computation, because the successive application of exponential and logarithmic functions cancel out. Our previous study already demonstrated the importance of the exact modeling during the training. However, using maximum approximation only in decoding did not hurt the recognition performance [36].

2.2. Feature space adaptation of Gaussian mixture models

In the constrained maximum likelihood linear regression (CMLLR) approach a global transformation is applied to each Gaussian parameter per speaker [10]. This is equivalent to a linear feature-space transformation, hence the goal is to select a matrix which maximizes the log-likelihood of the transformed input data under the given acoustic model and usually a Viterbi alignment to the given transcription:

$$\mathcal{N}(y|\mu_{s,r},\Sigma_{s,r}) = |A_r|\mathcal{N}(A_r y + a_r|\mu_s,\Sigma) \tag{3}$$

where r denotes the speaker cluster, $\mu_{s,r}$ and $\Sigma_{s,r}$ correspond to the speaker dependent Gaussian parameters of state s, obtained by a linear transformation of the non-adapted acoustic model parameters μ_s and Σ. A_r and a_r refer to the affine feature-space transform for the speaker r. The adaptation is not performed to the SI recognition model but to a simpler target model [40]. We use a single Gaussian model, thus the index i of the hidden variable was dropped.

2.3. Speaker dependent layer and joint training of speaker adapted BN-GMM

In this section, the speaker adaptive layer is introduced in a very similar way as in [37]. We define the speaker adapted layer in its general form, however, in order to implement the simultaneous training of the BN features and the adapted GMM, only the last hidden layer should be speaker dependent. Denoting the output of the lth layer as $z^{(l)}(x)$, the forward rule for a speaker dependent linear layer is defined as:

$$z^{(l)}(x_r) = A_r z^{(l-1)}(x_r) + a_r \tag{4}$$

where x_r corresponds to an input vector belonging to speaker r. The layer contains a set of linear transformations similar to a tensor layer [41]. However, only one transformation is selected by the hard speaker label. The backpropagation through the speaker dependent layer is very similar to a conventional linear layer, except that the speaker ID of the input vector is also considered.

$$\frac{\partial E}{\partial z^{(l-1)}} = \sum_j \frac{\partial E}{\partial z_j^{(l)}} \frac{\partial z_j^{(l)}}{\partial z^{(l-1)}} = A_r^{(l)T} \frac{\partial E}{\partial z^{(l)}} \tag{5}$$

where $\partial E / \partial z^{(l)}$ denotes the backpropagated error signal vector at the output of the lth layer w.r.t. the objective function E, and $z_j^{(l)}$ is the jth element of the vector $z^{(l)}$.

The speaker dependent layer can be considered an extension of a regular linear layer. The layer has a special input stream synchronized with the input features, which contains the speaker label of the current observation vector, and selects the proper transformation matrix and the bias. As a consequence of the frame-level shuffling of the training corpus, each frame in a mini-batch might have a different speaker ID. Thus the usual matrix-matrix multiplication needs to be carried out as a series of matrix-vector multiplications. The vectors within the mini-batch can be processed concurrently either by exploiting multiple cores of a modern CPU or by running parallel computation streams on a GPU. In case of the GPU implementation, keeping all speaker dependent matrices in the GPU memory might be impossible due to the limited available memory. This can be overcome if only matrices are copied from host to device that are necessary for processing the current mini-batch. This data transfer can overlap with the computation of the preceding speaker independent layers. At the recognition time a new matrix for each new speaker needs to be loaded. Their robust initialization on limited and unreliably transcribed data is one of the biggest challenges in NN adaptation with linear layers.

A speaker adapted tandem BN-GMM acoustic model can now be constructed from the previously introduced building blocks and is depicted in Fig. 1. The classic tandem feature extractor is followed by a speaker dependent linear layer initialized by CMLLR matrices. The speaker adaptively trained GMM is converted to an LMM and added to the network topology as the final layer. The full error backpropagation to the input layer then performs an additional adaptive training step. The Gaussian parameters are thus optimized simultaneously with the bottleneck (BN) features that were used to train the underlying acoustic model. The CMLLR addresses the robust initialization issue for a new speaker. Because the network training is based on CMLLR, both unsupervised and online adaptation methods can also be applied. Although the joint training of the speaker adapted BN-GMM results in a second set of tandem features for the second recognition pass, it can be extracted up to the BN immediately, i.e. in parallel with the first pass, because it does not include the speaker dependent layer.

2.4. Initialization of softmax layer with hidden variables

The training of the bottleneck MLP, the estimation of the single Gaussian target model, and the speaker adaptive training of the GMM can be interpreted as pretraining steps of our final joint model. We observed that the state posterior probability distribution of an ML GMM, especially with a high number of densities, is much sharper than the output of cross-entropy (CE) trained DNN models. This seems not to fit well to the discriminative criterion. We therefore applied the smoothing steps in Eq. 6 iteratively before integrating the GMM into the NN framework. The parameters α, β and γ were optimized in each step by a grid search minimizing the cross-entropy.

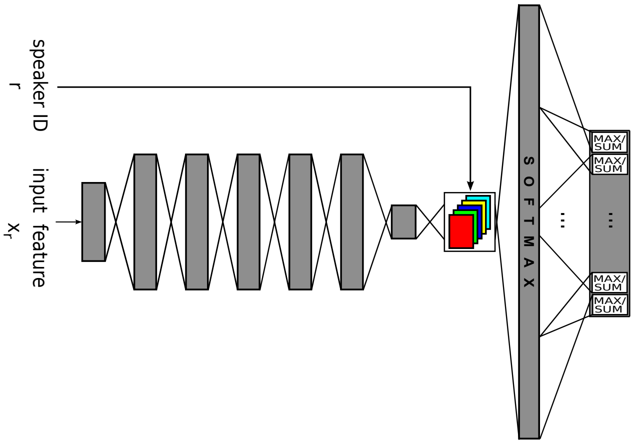

Fig. 1. *Joint model for tandem BN features and feature space adapted GMM within the DNN framework.*

$$\begin{aligned} w_{si} \rightarrow \alpha \cdot w_{si} \qquad b_{si} &\rightarrow \alpha \cdot b_{si} \\ p(s) &\rightarrow p^{\beta}(s) \\ p(i|s) &\rightarrow p^{\gamma}(i|s) \end{aligned} \tag{6}$$

In case of the maximum approximation in Eq. 1, the first step corresponds to the state posterior probability smoothing $p^{\alpha}(s|y)$, and does not affect the frame classification error. The other two steps „smooth” the class prior and the mixture component weights, often reversing the effect of the first step. According to our observation, the most crucial parameter is α, which we found to be roughly inverse proportional to the feature space dimension. Usually a single run over these three steps already reaches the minimum, and the three parameters can already be estimated robustly on a small subset of the cross-validation (CV) set, e.g. 5 hours of speech. For instance, on the Switchboard task the optimization ended up with values $(\alpha, \beta, \gamma) = (0.1, 9, 6)$. Measuring the cross-entropy on a subset of CV, the effect of weight smoothing, and the joint training of speaker independent or adapted BN-GMM is demonstrated in Fig. 2.

Since the softmax function is invariant under additive transformations, the parameters of the LMM can be centered around 0 by subtracting the mean value from each column of the weight matrix and from the bias vector:

$$\frac{\exp(w_{si}^T y + b_{si})}{Z(y)} = \frac{\exp((w_{si} - v)^T y + b_{si} - d)}{\hat{Z}(y)} \tag{7}$$

where $Z(y)$ denotes the denominator of the log-linear model, v and d correspond to the offset vector and scalar, respectively. The denominator $\hat{Z}(y)$ is defined using the centered parameters.

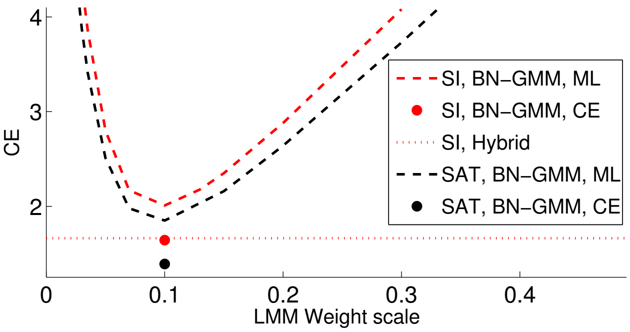

Fig. 2. *The effect of scaling the weights of an LMM initialized by speaker independent (SI) or speaker adaptively trained (SAT) maximum likelihood (ML) GMMs. The GMM contains 16 densities per state. For hybrid and joint BN-GMM training the cross-entropy (CE) training criterion was used.*

3. EXPERIMENTS ON BROADCAST NEWS AND CONVERSATIONS TASK

3.1. Experimental setup

The first set of experiments was carried out on a smaller set of the Quaero English broadcast news and conversations corpus [42]. For our research purpose, we defined a 50 hour subset of the full corpus as the training set. 10% of the data was held out to adjust the learning rate during the CE model training using the newbob scheduling. For a fair comparison of tandem and hybrid models, the CV was excluded from ML GMM training as well. The test data were automatically segmented by LIUM segmenter [43] and clustered by RASR [44]. For further details on the corpus we also refer to [45].

In every case the DNNs are trained on fast vocal tract length normalized input features [46]. The extraction of critical band energies and the discrete cosine transformation were followed by utterance-wise mean and variance normalization. Similar to our previous results reported on this small task [47, 48], the GMM and the DNN acoustic models were trained on the Viterbi alignment generated by the best performing evaluation system of the previous years. In this work we only use rectified linear unit (ReLU) based neural networks with L_2 regularization [49]. The NNs model 4500 tied triphone states and the weights are initialized by the discriminative pretraining [25]. The CMLLR matrices for the test data were estimated in an unsupervised manner on the first pass output using the simple target model approach [40]. The distribution of the cluster lengths can be seen in Fig. 3. The average cluster length of training, development and evaluation corpora are 208, 109, and 138 seconds.

3.2. Speaker independent results: improved baseline

In the first set of experiments we optimized our feature extraction pipeline. As shown in Table 1, increasing the resolution of the usual Mel-cepstral feature extraction pipeline from 16 to 40, we observed a significant word error rate (WER) reduction. An additional gain was measured once we switched from the MFCC pipeline to the Gammatone (GT) feature extraction [50].

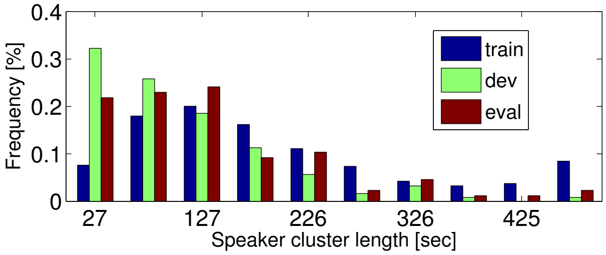

Fig. 3. *Speaker cluster length distribution on Quaero English.*

We also reconsidered the way the ±8 frames context of the cepstral features should be presented to the neural network. As can be seen in Table 2, the application of a non-dimension-reducing PCA to the 850-dimensional spliced input vector (17×50) outperformed the usual preprocessing steps where the features are concatenated with derivatives, followed by a global mean and variance normalization ($+\Delta, \Delta\Delta$; GMVN). Switching to a square LDA preprocessing of the spliced input, better results could be observed if the transformation ended up in identity within-scatter matrix. In summary, LDA finished slightly behind the PCA transformation.

Table 2 also shows that inserting 3 additional hidden layers improved the results. Further increase in the depth of the MLP did not lead to improvement.

Table 1. *Effect of different input features on a 6-layer MLP hybrid acoustic model. Word error rates (WER [%]) reported on Quaero English task using speaker independent systems.*

Features		WER [%]	
Type	Dimension	dev	eval
MFCC	16	17.4	23.0
	40	16.5	22.5
	50	16.8	22.4
GT	15	17.5	23.2
	50	**16.5**	**21.9**

3.3. Speaker adaptive and joint training of BN-MLP and GMM

In order to investigate the joint training of adapted BN-GMMs, we also trained tandem systems. Our previous study has already shown that a larger BN can be more beneficial after the CE training of a speaker independent BN-GMM [36]. Therefore we trained a hy-

Table 2. *Effect of different input feature transformations and more hidden layers using 50-dimensional GT features. Speaker independent results are reported on Quaero English task.*

Transformation	hidden layers	WER [%]	
		dev	eval
$+\Delta, \Delta\Delta$; GMVN	6	16.5	21.9
PCA		15.8	21.2
	9	**15.4**	**20.6**
LDA		15.6	20.7

brid system which introduced a 256-dimensional linear BN layer as the 10th hidden layer directly before the output. A single Gaussian model with a pooled covariance matrix is known to be mathematically equivalent to a hybrid model with a low-rank factorized last layer. Thus we include the result in the 2nd row of Table 3. As can be seen, the hybrid model with a BN achieved a slightly worse performance than the best system in Table 2.

After the extraction of the BN features the GMMs were trained according to the maximum likelihood criterion. Table 3 shows that the performance of ML trained GMMs saturates after 7 splits and lags behind the hybrid results (rows 2 and 5). However, after the joint training of the BN and the GMM the lost performance can be gained back. The CE trained BN-GMM (row 4) performed better than the hybrid with low-rank factorized output (row 2). The comparison with the best hybrid system (Table 2) does not allow a clear conclusion.

In the next step we include the CMLLR adaptation of the BN features and perform speaker adaptive training of the GMMs. All ML GMMs benefited from the transformed BN features (cf. rows 5 and 10). However, the CE training of the joint and adapted model showed less satisfying results. We measured a slight improvement over the corresponding SI model (rows 4 and 9), but compared to the best hybrid only the evaluation result showed a reduction of the error rate. Therefore, in further experiments we re-investigated whether the BN size, which was optimized in our previous study for SI systems, is still an optimal choice for CMLLR adaptation. Instead of training new hybrid models with smaller BN layers, the BN was reduced by PCA and then the two matrices were multiplied into a single BN layer resulting in the desired tandem model. The error rates in rows 9 and 13 of Table 3 clearly indicate that a narrower BN is crucial for the speaker adaptive joint training of BN-GMMs. In contrast, in speaker independent systems the larger BN pays off after the CE training [36]. In summary, the best CMLLR adapted and jointly trained BN-GMM showed 2% and 5% relative improvement over the best hybrid model on the development and evaluation sets.

4. EXPERIMENTS ON SWITCHBOARD

4.1. Experimental setup

Further, we perform evaluation on the Switchboard task. The training set consists of Switchboard-1 Release 2 (LDC97S62), and the Hub5'00 evaluation data (LDC2002S09) is used for testing. We report the word error rates for the Switchboard (SWB) and CallHome (CH) parts separately. The training transcripts of 2003 and the pronunciation lexicon have been downloaded from the ISIP website.[1] The recognition lexicon of size 30k has been derived from the training lexicon by removing word fragments, which are not modeled well by the language model, and converting all words to lower case. Thus there are no OOV words. The audio data segmentation is based on the manual transcription by ISIP and the Hub5'00 reference STM file. The speaker clusters have also been defined manually corresponding to audio files after separating the channels. For the estimation of the tied triphone states we used English phonetic questions from our Quaero setup. The resulting decision tree has 9000 leaves.

Table 3. *Effect of joint training on speaker independent (SI) and speaker adaptively (SA) trained tandem BN-GMM/LMM systems (Quaero English).*

AM	BN size	BN-GMM/LMM			WER		
		split	Joint training	Training criterion	[%]		
					dev	eval	
SI	256	0	no	ML	18.9	24.8	1.)
			yes	CE	15.9	21.5	2.)
		5	no	ML	16.7	22.5	3.)
			yes	CE	15.7	20.5	4.)
		7	no	ML	16.8	22.2	5.)
SA	256	0	no	ML	17.2	22.4	6.)
			yes	CE	16.0	20.9	7.)
		5	no	ML	16.4	21.6	8.)
			yes	CE	15.5	20.4	9.)
		7	no	ML	16.2	21.4	10.)
SA	128	0	no	ML	17.1	22.3	11.)
		5	no	ML	16.2	21.1	12.)
			yes	CE	15.1	**19.5**	13.)
		8	no	ML	16.0	21.1	14.)
SA	64	5	yes	CE	**14.9**	19.7	15.)
		9	no	ML	15.8	20.7	16.)

A 4-gram language model has been estimated by interpolating LMs trained on two data sets: the transcripts of the acoustic training data (3M running words) and the merged transcripts of Fisher English corpus part 1 (LDC2004T19) and 2 (LDC2005T19), that amount for 22M running words. The Kneser-Ney discount parameters and the interpolation weights have been optimized on 10k sentences held out from the Switchboard transcripts. The perplexity of the final LM on Hub5'00 is 71.

Our Switchboard model building is still in progress. The following experiments were carried out on an alignment obtained with a ML-SAT tandem GMM acoustic model that achieved 16.2% WER on the SWB part of the test set. In contrast to the Quaero data, Switchboard consists of narrow band telephone conversations, such that the feature extraction pipeline needs to be modified, resulting in 15 Mel and 40 Gammatone filters. We did not adjust the lower and upper filter bank frequencies to the telephone bandwidth (300 to 3400 Hz). Similar to the Quaero task we excluded 10% of 300h training data for CV such that there is no speaker overlap. The average speaker cluster length of training data is 230 seconds, and 185 and 140 seconds for the Switchboard and CallHome parts of Hub5'00, respecitvelly (Fig. 4). Due to the larger output layer, the size of the BN layer of the hybrid model with low-rank factorized output layer was increased from 256 to 512.

4.2. Baseline

Table 4 shows the speaker independent baseline results with two different features. The higher resolution of cepstral features again led to a significant improvement of the WER. Due to the larger training set, a 12-layer MLP became our current best speaker independent model. According to our knowledge this is one of the best speaker

[1] http://www.isip.piconepress.com/projects/switchboard/

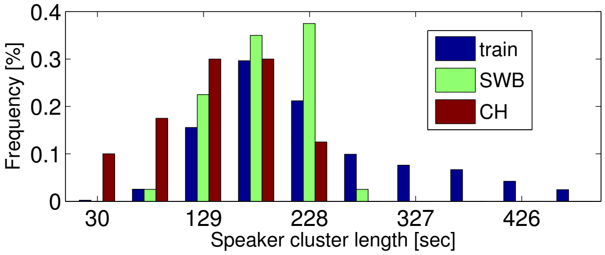

Fig. 4. *Speaker cluster length distribution on Switchboard.*

independent results ever published on the 300h Switchboard task after frame-wise training using only cepstral features. Surprisingly, CMLLR transformation of the input features did not boost our improved SI system.

Table 4. *Effect of different input features and number of hidden layers on hybrid acoustic models (Switchboard).*

Features		MLP		WER [%]		
Type	Dim.	#hidden	BN	SWB	CH	Total
MFCC	15	7	-	15.8	29.5	22.7
GT	40			15.1	27.7	21.4
		12	512	**13.7**	26.9	**20.3**
GT$_{\text{CMLLR}}$				13.9	**26.6**	20.3

4.3. Speaker adapted systems

After setting up the baseline hybrid models, we trained tandem models on the 128-dimensional PCA reduced BN output. Table 5 presents the results of our speaker independent and adapted, ML and CE trained BN-GMM systems. The CE objective function before and after the joint training of a SI and SA tandem BN-GMM is shown in Fig. 2. As can be seen, the jointly trained SI BN-GMM achieved the same recognition performance as the best hybrid (row 3). Table 5 also shows that the best speaker adapted ML GMM (row 11) already outperforms our best hybrid model and the gap increased further after speaker adaptive joint training of the model with CE criterion (row 10). The unsupervised adaptation of BN-GMM leads to around 5% relative improvement over the best hybrid model on both the Switchboard and the CallHome part of the test data. This improvement is confirmed after minimum phone error (MPE) sequence level discriminative training of the acoustic models [51]. After MPE, we measured 12.6% WER for the best hybrid model and 11.5% WER for the speaker adapted BN-GMM on the Switchboard test set.

The results in rows 7 and 10 indicate the importance of the mixture layer, the CMLLR aware adaptation of a standard low-rank factorized hybrid output is 0.5% absolute behind the adapted BN-GMM. Further experiments (rows 6-7 and 9-10) also show that the CE training solely of the GMM is not enough, and that deeper back-propagation is necessary to obtain the best speaker adapted results.

Table 5. *Effect of joint training on speaker independent (SI) and speaker adaptively (SA) trained tandem BN-GMM/LMM systems (Switchboard).*

	BN size	BN-GMM/LMM			WER [%]			
		split	Joint training	Training criterion	SWB	CH	Total	
SI	128	0	no	ML	18.5	35.0	26.8	1.)
		4	no	ML	15.4	29.7	22.6	2.)
			yes	CE	13.6	27.0	20.3	3.)
		8	no	ML	14.2	27.8	21.0	4.)
SA	128	0	no	ML	16.0	29.8	22.9	5.)
				CE	14.1	26.4	20.3	6.)
			yes		13.4	25.4	19.4	7.)
		4	no	ML	13.9	26.9	20.7	8.)
				CE	13.7	26.0	19.9	9.)
			yes		**12.9**	**24.9**	**18.9**	10.)
		8	no	ML	13.3	26.4	19.9	11.)

5. CONCLUSIONS

It has been shown that a feature-space adapted tandem BN-GMM can be easily expressed in the neural network framework after introducing a speaker dependent linear layer. The speaker adaptation of a tandem MLP-GMM is closely related to linear BN layer adaptation of a low-rank factorized output layer in a hybrid system. We also demonstrated that the neural networks can be trained CMLLR transformation aware. During the recognition the CMLLR matrices can be estimated in an unsupervised manner, fitting well to the model. On two different English speech recognition tasks – broadcast news and conversations, and conversational telephone speech – we measured 5% relative WER improvement over a very strong speaker independent hybrid baseline after cross-entropy training. Our results on the Switchboard part of Hub5'00 improved from 12.6% to 11.5% word error rate by the proposed model after sequence level discriminative training.

Some obvious extensions of the proposed method include the application of multiple BN and CMLLR layers as well as the discriminative fine-tuning of the speaker dependent layer initialized by CMLLR. Further improvements are expected by using i-vectors or more complex neural network structures [32]. Our research will extend in these directions.

6. 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.

7. REFERENCES

[1] H. Soltau, G. Saon, and T. N. Sainath, "Joint training of convolutional and non-convolutional neural networks," in *ICASSP*, 2014, pp. 5609–5613.

[2] E. McDermott, G. Heigold, P. Moreno, A. Senior, and M. Bacchiani, "Asynchronous Stochastic Optimization for Sequence Training of Deep Neural Networks : Towards Big Data," in *Interspeech*, 2014, pp. 1224–1228.

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

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

[5] Z.-J. Yan, Q. Huo, and J. Xu, "A Scalable Approach to Using DNN-Derived Features in GMM-HMM Based Acoustic Modeling For LVCSR," in *Interspeech*, 2013, pp. 104–108.

[6] S. Wegmann, D. McAllaster, J. Orloff, and B. Peskin, "Speaker normalization on conversational telephone speech," in *ICASSP*, vol. 1, 1996, pp. 339–341.

[7] M. Pitz and H. Ney, "Vocal tract normalization as linear transformation of MFCC," in *Eurospeech*, 2003, pp. 1445–1448.

[8] C. Leggetter and P. Woodland, "Maximum likelihood linear regression for speaker adaptation of continuous density hidden Markov models," *Computer Speech & Language*, vol. 9, no. 2, pp. 171–185, Apr. 1995.

[9] V. V. Digalakis, D. Rtischev, and L. G. Neumeyer, "Speaker adaptation using Constrained estimation of Gaussian mixtures," *IEEE Transactions on Speech and Audio Processing*, vol. 3, no. 5, pp. 357–366, 1995.

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

[11] J.-L. Gauvain and C.-H. Lee, "Maximum a posteriori estimation for multivariate Gaussian mixture observations of Markov chains," *IEEE Transactions on Speech and Audio Processing*, vol. 2, no. 2, pp. 291–298, 1994.

[12] Q. Zhu, B. Chen, N. Morgan, and A. Stolcke, "On using MLP features in LVCSR," in *Interspeech*, 2004, pp. 921–924.

[13] T. Schaaf and F. Metze, "Analysis of Gender Normalization Using MLP and VTLN Features," in *Interspeech*, 2010, pp. 306–309.

[14] Z. Tüske, C. Plahl, and R. Schlüter, "A study on speaker normalized MLP features in LVCSR," in *Interspeech*, 2011, pp. 1089–1092.

[15] Y. Huang, D. Yu, C. Liu, and Y. Gong, "A Comparative Analytic Study on the Gaussian Mixture and Context Dependent Deep Neural Network Hidden Markov Models," in *Interspeech*, 2014, pp. 1895–1899.

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

[17] R. Doddipatla, M. Hasan, and T. Hain, "Speaker Dependent Bottleneck Layer Training for Speaker Adaptation in Automatic Speech Recognition," in *Interspeech*, 2014, pp. 2199–2203.

[18] J. P. Neto, C. Martins, and L. B. Almeida, "Speaker-adaptation in a hybrid HMM-MLP recognizer," in *ICASSP*, 1996, pp. 3382–3385.

[19] 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.

[20] J. Neto, L. Almeida, M. Hochberg, C. Martins, L. Nunes, S. Renals, and T. Robinson, "Speaker-Adaptation for Hybrid HMM-ANN Continuous Speech Recognition System," in *Eurospeech*, 1995, pp. 2171–2174.

[21] R. Gemello, F. Mana, S. Scanzio, P. Laface, and R. De Mori, "Linear hidden transformations for adaptation of hybrid ANN/HMM models," *Speech Communication*, vol. 49, no. 10-11, pp. 827–835, 2007.

[22] 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.

[23] J. Xue, J. Li, D. Yu, M. Seltzer, and Y. Gong, "Singular value decomposition based low-footprint speaker adaptation and personalization for deep neural network," in *ICASSP*, 2014, pp. 6409–6413.

[24] V. Abrash, H. Franco, A. Sankar, and M. Cohen, "Connectionist speaker normalization and adaptation," in *Eurospeech*, 1995, pp. 2183–2186.

[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] T. N. Sainath, B. Kingsbury, B. Ramabhadran, P. Fousek, P. Novak, and A.-r. Mohamed, "Making Deep Belief Networks Effective for Large Vocabulary Continuous Speech Recognition," in *ASRU*, 2011, pp. 30–35.

[27] M. Karafiát, F. Grézl, M. Hannemann, and J. Černocký, "BUT neural network features for spontaneous vietnamese in Babel," in *ICASSP*, 2014, pp. 5622–5626.

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

[29] Y.-Q. Wang and M. J. F. Gales, "Tandem system adaptation using multiple linear feature transforms," in *ICASSP*, 2013, pp. 7932–7936.

[30] 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.

[31] N. Dehak, P. J. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 19, no. 4, pp. 788–798, 2011.

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

[33] Y. Miao, H. Zhang, and F. Metze, "Towards Speaker Adaptive Training of Deep Neural Network Acoustic Models," in *Interspeech*, 2014, pp. 2189–2193.

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

[35] E. Variani, E. McDermott, and G. Heigold, “A Gaussian mixture model layer jointly optimized with discriminative features within a deep neural network architecture,” in *ICASSP*, 2015, pp. 4270–4274.

[36] Z. Tüske, M. A. Tahir, R. Schlüter, and H. Ney, “Integrating Gaussian mixtures into deep neural networks: softmax layer with hidden variables,” in *ICASSP*, 2015, pp. 4285–4289.

[37] T. Ochiai, S. Matsuda, X. Lu, C. Hori, and S. Katagiri, “Speaker adaptive training using deep neural networks,” in *ICASSP*, 2014, pp. 6399–6403.

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

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

[40] G. Stemmer, F. Brugnara, and D. Giuliani, “Adaptive training using simple target models,” in *ICASSP*, vol. 1, 2005, pp. 997–1000.

[41] D. Yu, X. Chen, and L. Deng, “Factorized deep neural networks for adaptive speech recognition,” in *Int. Workshop on Statistical Machine Learning for Speech Processing*, 2012.

[42] 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.

[43] M. Rouvier, G. Dupuy, P. Gay, E. Khoury, T. Merlin, and S. Meignier, “An open-source state-of-the-art toolbox for broadcast news diarization,” in *Interspeech*, 2013, pp. 1477–1481.

[44] D. Rybach, S. Hahn, P. Lehnen, D. Nolden, M. Sundermeyer, Z. Tüske, S. Wiesler, R. Schlüter, and H. Ney, “RASR - The RWTH Aachen University Open Source Speech Recognition Toolkit,” in *ASRU*, 2011.

[45] 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.

[46] 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.

[47] S. Wiesler, A. Richard, R. Schlüter, and H. Ney, “Mean-normalized stochastic gradient for large-scale deep learning,” in *ICASSP*, 2014, pp. 180–184.

[48] Z. Tüske, P. Golik, R. Schlüter, and H. Ney, “Acoustic Modeling with Deep Neural Networks Using Raw Time Signal for LVCSR,” in *Interspeech*, 2014, pp. 890–894.

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

[50] 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.

[51] D. Povey and P. Woodland, “Minimum phone error and i-smoothing for improved discriminative training,” in *ICASSP*, Orlando, FL, USA, May 2002, pp. I–105–I–108.