INTERSPEECH 2016
September 8–12, 2016, San Francisco, USA

GMM-Free Flat Start Sequence-Discriminative DNN Training

Gábor Gosztolya [1,2], *Tamás Grósz* [2], *László Tóth* [1]

[1] MTA-SZTE Research Group on Artificial Intelligence, Szeged, Hungary
[2] Department of Informatics, University of Szeged, Hungary
{ ggabor, groszt, tothl } @ inf.u-szeged.hu

Abstract

Recently, attempts have been made to remove Gaussian mixture models (GMM) from the training process of deep neural network-based hidden Markov models (HMM/DNN). For the GMM-free training of a HMM/DNN hybrid we have to solve two problems, namely the initial alignment of the frame-level state labels and the creation of context-dependent states. Although flat-start training via iteratively realigning and retraining the DNN using a frame-level error function is viable, it is quite cumbersome. Here, we propose to use a sequence-discriminative training criterion for flat start. While sequence-discriminative training is routinely applied only in the final phase of model training, we show that with proper caution it is also suitable for getting an alignment of context-independent DNN models. For the construction of tied states we apply a recently proposed KL-divergence-based state clustering method, hence our whole training process is GMM-free. In the experimental evaluation we found that the sequence-discriminative flat start training method is not only significantly faster than the straightforward approach of iterative retraining and realignment, but the word error rates attained are slightly better as well.
Index Terms: deep neural networks, flat start, sequence discriminative DNN training

1. Introduction

While deep neural network (DNN) based speech recognizers have recently replaced Gaussian mixture (GMM) based systems as the state-of-the-art in ASR, the training process of HMM/DNN hybrids still relies on the HMM/GMM framework. Conventionally, we start the training of a HMM/DNN by constructing a HMM/GMM system, which is then applied to get an alignment for the frame-level state labels. These labels are then used as the training targets for the DNN. The second task that requires GMMs is the state tying algorithm utilized for the construction of context-dependent (CD) phone models. We proposed a GMM-free solution for state clustering earlier [1], and in this study we will focus on the issue of obtaining the initial state alignment for training the DNN.

The most convenient way of training the DNN component of a HMM/DNN hybrid is by applying a frame-level error criterion, which is usually the cross-entropy (CE) function. This solution, however, requires frame-aligned training labels, while the training dataset contains just orthographic transcripts in most cases. Trivially, one may train a HMM/GMM system to get aligned labels, but this is clearly a waste of resources.

The procedure for training HMM/GMM systems without alignment information is commonly known as 'flat start training' [2]. This consists of initializing all phone models with the same parameters, which would result in a uniform alignment of phone boundaries in the first iteration of Baum-Welch training. It is possible to construct a flat start-like training procedure for CE-trained DNNs as well, by iteratively training and realigning the DNN. For example, Senior et al. randomly initialized their neural network [3], while Zhang et al. trained their first model on equal-sized segments for each state [4]. As these solutions have a slow convergence rate, they require a lot of training-realignment loops.

Although training the DNN at the frame level is straightforward, it is clearly not optimal, as the recognition is performed and evaluated at the sentence level. Within the framework of HMM/GMM systems, several sequence-discriminative training methods have been developed, and these have now been adapted to HMM/DNN hybrids as well [5, 6, 7]. However, most authors apply sequence-discriminative criteria only in the final phase of training, for the refinement of the DNN model. That is, the first step is always CE-based training, either to initialize the DNN (e.g. [8, 9, 10]) or just to provide frame-level state labels (e.g. [5, 6, 11, 12, 13]).

The Connectionist Temporal Classification (CTC) approach has recently become very popular for training DNNs without an initial time alignment being available [14]. Rao et al. proposed a flat start training procedure which is built on CTC [15]. However, CTC has several drawbacks compared to MMI. First, it introduces blank labels, which require special care in the later steps (e.g. CD state tying) of the training process. Second, the CTC algorithm is not a sequence-discriminative training method, so for the best performance it has to be combined with techniques like sMBR training [14, 15].

In contrast with the previous authors, here we propose a training procedure that applies sequence-discriminative training in the flat-start training phase. This requires several small modifications compared to the standard usage of sequence-discriminative training, which will be discussed in detail. In the experimental part we compare the proposed method with the CE-based iterative retraining-realignment procedure of Zhang et al. [4], and we find that our method is faster and gives slightly lower word error rates. Furthermore, we can combine sequence-discriminative flat start training with the Kullback-Leibler divergence-based state clustering method we proposed recently [1]. With this, we eliminate all dependencies from a HMM/GMM system, making the whole training procedure of context-dependent HMM/DNNs GMM-free.

2. Flat-start training of HMM/DNN

Conventionally, the training of a HMM/DNN system is initiated by training a HMM/GMM just to get time-aligned training labels. Here, we compare two approaches that seek to eliminate GMMs from this process. As the baseline method, we apply a simple solution that iterates the loop of CE DNN training and

http://dx.doi.org/10.21437/Interspeech.2016-391

realignment. Afterwards, we propose an approach that creates time-aligned transcriptions for the training data by training a DNN with a sequence training criterion. From the wide variety of sequence training methods, we opted for MMI (Maximum Mutual Information) training [5]. Applying sequence training to flat start requires some slight modifications, which we will now discuss.

2.1. Iterative CE Training and Realignment

For comparison we will also test what is perhaps the most straightforward solution for flat start DNN training, namely just using the CE training criterion and iterating DNN training and realignment. Here, we used the following algorithm that was based on the description of Zhang et al. [4]:

1. Train a DNN using uniformly segmented sound files.
2. Use the current DNN to realign the labels.
3. Train a randomly initialized DNN using the new alignments.
4. Repeat steps 2–3 several times.

The final DNN was utilized to create time-aligned labels for the training set.

The main advantage of this method is that it requires only an implementation of CE training for the DNN, and the realignment step can also be readily performed by using standard ASR toolkits. The drawback is that the procedure of retraining and realignment tends to be rather time-consuming, which was also confirmed by our experiments (see Section 6).

2.2. Sequence-Discriminative Training Using MMI

Several sequence-discriminative training criteria have been developed for HMM/GMMs [16] – and adapted to HMM/DNNs [5, 6, 12, 17] – from which the maximum mutual information (MMI) criterion is the oldest and simplest. The MMI function measures the mutual information between the distribution of the observation and the phoneme sequence. Denoting the sequence of all observations by $O_u = o_{u1}, \ldots, o_{uT_u}$, and the label-sequence for utterance u by W_u, the MMI criterion can be defined by the formula

$$F_{MMI} = \sum_u \log \frac{p(O_u|S_u)^\alpha p(W_u)}{\sum_W p(O_u|S)^\alpha p(W)}, \tag{1}$$

where $S_u = s_{u1}, \ldots, s_{uT_u}$ is the sequence of states corresponding to W_u, and α is the acoustic scaling factor. The sum in the denominator is taken over all phoneme sequences in the decoded speech lattice for u. Differentiating Eq. (1) with respect to the log-likelihood $\log p(o_{ut}|r)$ for state r at time t, we get

$$\begin{aligned}\frac{\partial F_{MMI}}{\partial \log p(o_{ut}|r)} &= \alpha\delta_{r;s_{ut}} - \frac{\alpha \sum_{W:s_t=r} p(O_u|S)^\alpha p(W)}{\sum_W p(O_u|S)^\alpha p(W)} \\ &= \alpha\big(\delta_{r;s_{ut}} - \gamma_{ut}^{DEN}(r)\big),\end{aligned} \tag{2}$$

where $\gamma_{ut}^{DEN}(r)$ is the posterior probability of being in state r at time t, computed over the denominator lattices for utterance u using the forward-backward algorithm, and $\delta_{r;s_{ut}}$ is the Kronecker delta function (the binary frame-level phonetic targets).

3. Performing DNN Flat Start with MMI

Sequence training criteria like the MMI error function are now widely used in DNN training. However, all authors initialize their networks using CE training, and apply the sequence-discriminative criterion only in the final phase of the training procedure, to fine-tune their models [6, 12], which makes it necessary to use some method (HMM/GMM or iterative CE training) to provide frame-level state targets. In contrast with these authors, here we propose to apply MMI training in the flat start phase. In order to be able to perform flat start of randomly initialized DNNs using sequence training, we made some slight changes in the standard MMI process, which we will describe next.

Firstly, we use the numerator occupancies $\gamma_{ut}^{NUM}(r)$ in Eq. (2) instead of the $\delta_{r;s_{ut}}$ values. This way we can work with smoother targets instead of the crude binary ones usually employed during DNN training. Another advantage of eliminating the $\delta_{r;s_{ut}}$ values is that it allows us to skip the preceding (usually GMM-based) label alignment step, responsible for generating the frame-level training targets. We applied the forward-backward algorithm to obtain the $\gamma_{ut}^{NUM}(r)$ values, which solution has been mentioned in some studies (e.g. [6, 17]), but we only found Zhou et al. [8] actually doing this. However, they pre-trained their DNN with the CE criterion first, while we apply MMI training from the beginning, starting with randomly initialized weights.

The second difference is that sequence training is conventionally applied only to refine a fully trained system. Thus, the MMI training criterion is calculated with CD phone models and a word-level language model. This makes the decoding process slow, and hence the numerator and denominator lattices are calculated only once, before starting MMI training. In contrast to this, we execute sequence DNN training using only phone-level transcripts and CI phone models. This allows very fast decoding, so we can recalculate the lattices after each sentence. This difference is crucial for the fast convergence of our procedure. For converting the orthographic transcripts to phone sequences one can follow the strategy of HTK. That is, in the very first step we get the phonetic transcripts from the dictionary, with no silences between the words. Pronunciation alternatives and the optional short pause at word endings can be added later on, performing realignment with a sufficiently well-trained model [2].

A further difference is that we use no state priors or language model, which makes the α scaling factor in Eq. (2) unnecessary as well. Next, to reduce the computational requirements of the algorithm, we estimated $\gamma_{ut}^{DEN}(r)$ using just the most probable decoded path instead of summing over all possible paths in the lattice (denoted by $\hat{\gamma}_{ut}^{DEN}(r)$).

With these modifications, the gradient with respect to the output activations (a_{ut}) of the DNN is found using

$$\begin{aligned}\frac{\partial F_{MMI}}{\partial a_{ut}(s)} &= \sum_r \frac{\partial F_{MMI}}{\partial \log p(o_{ut}|r)} \frac{\partial \log p(o_{ut}|r)}{\partial a_{ut}(s)} \\ &= \gamma_{ut}^{NUM}(s) - \hat{\gamma}_{ut}^{DEN}(s),\end{aligned} \tag{3}$$

which can be applied directly for DNN training. A standard technique in DNN training is to separate a hold-out set from the training data (see e.g. [18]). If the error increases on this hold-out set after a training iteration, then the DNN weights are restored from a backup and the training continues with a smaller learning rate. This strategy can be readily adapted to sequence DNN training [5], and we found it to be essential for the stability of our flat-start MMI DNN training method.

(1) Frame-level phonetic targets ($\gamma_{ut}^{NUM}(r)$) are determined by a forward-backward search.

(2) We use only phoneme-level transcripts and CI phoneme states.

(3) We do not use state priors or language model.

(4) We estimate $\gamma_{ut}^{DEN}(r)$ by just the most probable decoded path ($\hat{\gamma}_{ut}^{DEN}(r)$).

(5) We measure training error on a hold-out set; when the error increases after a training iteration, we restore the weights and lower the learning rate.

Table 1: *Summary of our modifications on MMI training for DNN flat start.*

Flat start method	State tying method	WER %		No. of epochs
		Dev.	Test	
GMM + ANN	GMM	18.83%	17.27%	—
GMM + ANN	KL	17.12%	16.54%	—
Iterative CE	KL	16.81%	16.50%	48
MMI		16.50%	15.96%	13
MMI + CE		16.36%	15.86%	29

Table 2: *Word error rates (WER) for the different flat start and state tying strategies.*

Table 1 summarizes the modifications that we made to make MMI suitable for DNN flat start. Note that steps (1) through (4) seek to simplify the procedure both to speed it up and to make it more robust. Step (2) also helps us to perform sequence-discriminative DNN training before CD state tying, which is essential for applying it in flat start. Step (5), however, is applied in our general DNN training process, but we found it essential to avoid the "runaway silence model" issue which is a common side effect haunting sequence-discriminative DNN training.

4. KL divergence-based CD state tying

Having aligned the CI phone models using flat-start training, the next step is the construction of CD models. Currently, the dominant solution for this is the decision tree-based state tying method [19]. This technique pools all context variants of a state, and then builds a decision tree by successively splitting this set into two, according to one of the pre-defined questions. For each step, it fits Gaussians on the distribution of the states, and chooses the question which leads to the highest likelihood gain. However, modeling the distribution of states with a Gaussian function might be suboptimal when we utilize DNNs in the final acoustic model.

To this end, we decided to first train an auxiliary neural network on the CI target labels and then perform the CD state tying based on the output of this network. Such a frame-level output can be treated as a discrete probability distribution, and a natural distance function for such distributions is the Kullback-Leibler (KL) divergence [20]. Therefore, to control the state tying process, we utilized the KL divergence-based decision criterion introduced by Imseng et al. [21, 22]. We basically replaced the Gaussian-based likelihood function with a KL-divergence based state divergence function; in other respects, the mechanism of the CD state tying process remained the same. With this technique we were not only able to eliminate GMMs from the state tying process, but we also achieved a 4% reduction in WER. For details, see [1].

5. Experimental Setup

Our experimental setup is essentially the same as that of our previous study [1]. We employed a DNN with 5 hidden layers, each containing 1000 rectified neurons [23], while the softmax activation function was applied in the output layer. We used our custom DNN implementation for GPU, which achieved outstanding results on several datasets (e.g. [24, 25, 26, 27, 28]). We used 40 mel filter bank energies as features along with their first and second order derivatives. Decoding and evaluation was performed by a modified version of HTK [2].

The 28 hour-long speech corpus of Hungarian broadcast news [29] was collected from eight TV channels. The training set was about 22 hours long, a small part (2 hours) was used for validation purposes, and a 4-hour part was used for testing. We used a trigram language model and a vocabulary of 500k word forms. The order of utterances was randomized at the beginning of training. We configured the state tying algorithms to get roughly 600, 1200, 1800, 2400, 3000 and 3600 tied states.

We tested four approaches for flat start training (i.e. to get the frame-level phonetic targets for CD state tying and CE DNN training). Firstly, we applied the standard GMM-based flat-start training to produce initial time-aligned labels. To further improve the segmentation, we trained a shallow CI ANN using the CE criterion and re-aligned the frame labels based on the outputs of this ANN (we will refer to this approach as the *"GMM + ANN"* method). (In our previous study we found that using a deep network for this re-alignment setup did not give any significant improvement [1].) After the realignment, we applied both the standard GMM-based and our KL-criterion algorithms for state tying. Then KL-based state tying was performed on the output of the CI ANN.

Besides the standard GMM flat start approach, we evaluated the two algorithms presented in sections 2 and 3 for flat starting with DNNs. In these tests we always used five-hidden-layer CI DNNs. For the flat-start method with iterative CE training (*"Iterative CE"*) we performed four training-aligning iterations, and KL-based CD state tying was performed using the output and the alignments created by the final DNN. For MMI training (*"MMI"*) we also commenced with a randomly initialized CI DNN. After applying the discriminative sequence training method proposed in Section 3, the resulting DNN was used to create forced aligned labels and also to provide the input posterior estimates for KL clustering. In the last flat start approach tested, we first applied the sequence-discriminative flat start method (i.e. "MMI"). Then, based on the alignments of this network, we trained another DNN with the CE criterion to supply both the final frame labels and the likelihoods for KL-based CE state tying (*"MMI + CE"*).

The aim of this study was to compare various flat-start strategies. This is why, after obtaining the CD labels, the final DNN models were trained starting from randomly initialized weights and using just the CE criterion. Of course, it might be possible to extend the training with a final refinement step using sequence-discriminative training.

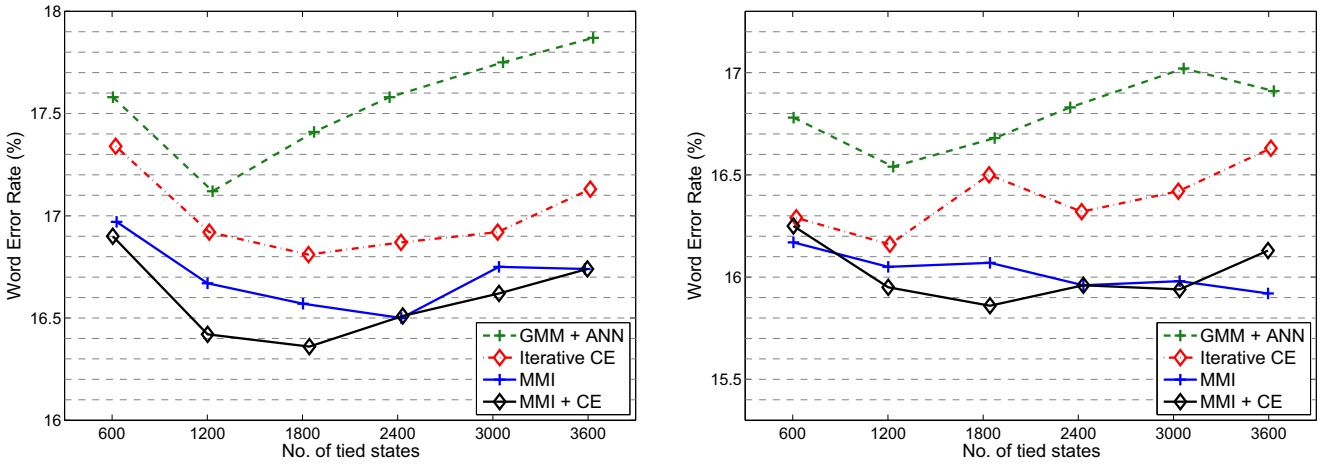

Figure 1: WER as a function of the number of KL-clustered tied states on the development (left) and test (right) sets.

6. Results and Discussion

Figure 1 shows the resulting WER scores as a function of the number of CD tied states. As can be seen, the MMI-based flat start strategy gave slightly better results than the iterative method in every case. We also observed that the final CD models which got their training labels from the MMI-trained DNN were more stable with respect to varying the number of CD states. Fine-tuning the labels of the MMI-trained DNN with a CE-trained DNN ("MMI" vs. "MMI+CE") seems unnecessary, as it was not able to significantly improve the results. This indicates that sequence training yields both fine alignments and good posterior estimates.

Table 2 summarizes the best WER values on the development set, and the corresponding scores on the test set. The KL clustering method clearly outperformed the GMM-based state tying technique. Comparing the alignment methods, we see that relying on the alignments produced by the HMM/GMM resulted in the lowest accuracy score, in spite of the fine-tuning step using an ANN. With the parameter configurations applied, the iterative CE training method performed slightly worse than the MMI-based strategies. Unfortunately, for the iterative CE method the right number of training-aligning steps is hard to tune. For example, Zhang et al. performed 20 such iterations [4], while we employed only 4 iterations. In this respect, it is more informative to compare the training times, which are shown in the rightmost column of Table 2. (We did not indicate the number of epochs for the "GMM + ANN" method, as the training procedure was radically different there.) For our 28-hour dataset, 48 epochs were required by the four iterations of iterative CE flat-start training, while MMI required only one-fourth of it; and although performing the forward-backward search adds a slight overhead to the training process, it is clear that MMI was still much faster, even when the final CE re-alignment step was also involved.

Measuring the training times in CPU/GPU time gives even larger differences in favor of the MMI method (3 hours vs. 16 hours). The reason is that for iterative CE flat-start training we used a mini-batch of 100 frames (which we found optimal previously [1]), while for MMI whole utterances (usually more than 1000 frames) were used to update the weights, and this allowed better parallelization on the GPU.

In our view, two modifications are crucial for the speed and stability of the proposed algorithm. The first one is that we use only CI phone models without phone language model, so we can very quickly update the numerator and denominator lattices after the processing of each sentence. This continuous refinement of the frame-level soft targets obviously leads to a faster convergence. The only study we know of, which does not perform the re-alignment of the frame-level targets strictly after a training iteration, is that of Bacchiani et al. [30]. Their study focuses on describing their massively parallelized online neural network optimization system, where a separate thread is responsible for the alignment of the phonetic targets, while DNN training is performed by the client machines. Besides the fact that in their model there is no guarantee for that the alignment of phonetic targets are up-to-date, it is easy to see that their architecture is quite different from a standard DNN training architecture, making their techniques pretty hard to adapt. In contrast, our slight modifications can be applied relatively easily.

As regards stability, a known drawback of sequence training methods is that the same process is responsible both for aligning and training the DNN, which often leads to the "run-away silence model" issue [31]. That is, after a few iterations, only one model (usually the silence model) dominates most parts of the utterances, which is even reinforced in the next training step. To detect the occurrence of this phenomenon, we monitored the error rate on a hold-out set during training. If the error increased after an iteration, we restored the weights of the network to their previous values and the learning rate was halved. In our experience, restoring the weights to their previous values and continuing the training using a lower learning rate can successfully handle this issue.

7. Conclusions

Here, we showed how to perform flat start with sequence-discriminative DNN training. We applied the standard MMI sequence training method, for which we introduced several minor modifications. Our results showed that, compared to the standard procedure of iterative CE DNN training and re-alignment, not only were we able to reduce the WER scores, but we also achieved a significant reduction in training times. By also utilizing the Kullback-Leibler divergence-based CD state tying method proposed earlier, we made the whole training procedure of context-dependent HMM/DNNs GMM-free.

8. References

[1] G. Gosztolya, T. Grósz, L. Tóth, and D. Imseng, "Building context-dependent DNN acousitc models using Kullback-Leibler divergence-based state tying," in *Proceedings of ICASSP*, 2015, pp. 4570–4574.

[2] S. Young, G. Evermann, M. J. F. Gales, T. Hain, D. Kershaw, G. Moore, J. Odell, D. Ollason, D. Povey, V. Valtchev, and P. Woodland, *The HTK Book*. Cambridge, UK: Cambridge University Engineering Department, 2006.

[3] A. Senior, G. Heigold, M. Bacchiani, and H. Liao, "GMM-free DNN acoustic model training," in *Proceedings of ICASSP*, 2014, pp. 5639–5643.

[4] C. Zhang and P. Woodland, "Standalone training of context-dependent Deep Neural Network acoustic models," in *Proceedings of ICASSP*, 2014, pp. 5634–5638.

[5] B. Kingsbury, "Lattice-based optimization of sequence classification criteria for neural-network acoustic modeling," in *Proceedings of ICASSP*, 2009, pp. 3761–3764.

[6] K. Veselý, A. Ghoshal, L. Burget, and D. Povey, "Sequence-discriminative training of deep neural networks," in *Proceedings of Interspeech*, 2013, pp. 2345–2349.

[7] T. Grósz, G. Gosztolya, and L. Tóth, "A sequence training method for Deep Rectifier Neural Networks in speech recognition." in *Proceedings of SPECOM*, Novi Sad, Serbia, Sep 2014, pp. 81–88.

[8] P. Zhou, L. Dai, and H. Jiang, "Sequence training of multiple Deep Neural Networks for better performance and faster training speed," in *Proceedings of ICASSP*, 2014, pp. 5664–5668.

[9] 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 *Proceedings of Interspeech*, 2014, pp. 1224–1228.

[10] A. Mohamed, D. Yu, and L. Deng, "Investigation of full-sequence training of Deep Belief Networks for speech recognition," in *Proceedings of Interspeech*, 2010, pp. 2846–2849.

[11] G. Saon and H. Soltau, "A comparison of two optimization techniques for sequence discriminative training of Deep Neural Networks," in *Proceedings of ICASSP*, 2014, pp. 5604–5608.

[12] S. Wiesler, P. Golik, R. Schüter, and H. Ney, "Investigations on sequence training of neural networks," in *Proceedings of ICASSP*, 2015, pp. 4565–4569.

[13] D. Chen, B. Mak, and S. Sivadas, "Joint sequence training of phone and grapheme acoustic model based on multi-task learning Deep Neural Networks," in *Proceedings of Interspeech*, 2014, pp. 1083–1087.

[14] A. Graves, A.-R. Mohamed, and G. E. Hinton, "Speech recognition with Deep Recurrent Neural Networks," in *Proceedings of ICASSP*, 2013, pp. 6645–6649.

[15] K. Rao, A. Senior, and H. Sak, "Flat start training of CD-CTC-SMBR LSTM RNN acoustic models," in *Proceedings of ICASSP*, Shanghai, China, 2016, pp. 5405–5409.

[16] X. He and L. Deng, *Discriminative Learning for Speech Recognition*. San Rafael, CA, USA: Morgan & Claypool, 2008.

[17] D. Yu and L. Deng, "Chapter 8: Deep neural network sequence-discriminative training," in *Automatic Speech Recognition — A Deep Learning Approach*. Springer, October 2014.

[18] S. J. Rennie, V. Goel, and S. Thomas, "Annealed dropout training of deep networks," in *Proceedings of SLT*, South Lake Tahoe, NV, USA, 2014, pp. 159–164.

[19] S. J. Young, J. J. Odell, and P. C. Woodland, "Tree-based state tying for high accuracy acoustic modelling," in *Proceedings of HLT*, 1994, pp. 307–312.

[20] S. Kullback and R. Leibler, "On information and sufficiency," *Ann. Math. Statist.*, vol. 22, no. 1, pp. 79–86, 1951.

[21] D. Imseng and J. Dines, "Decision tree clustering for KL-HMM," IDIAP Research Institute, Tech. Rep. Idiap-Com-01-2012, 2012.

[22] D. Imseng, J. Dines, P. Motlicek, P. Garner, and H. Bourlard, "Comparing different acoustic modeling techniques for multilingual boosting," in *Proceedings of Interspeech*, 2012.

[23] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier networks," in *Proceedings of AISTATS*, 2011, pp. 315–323.

[24] L. Tóth, "Convolutional deep maxout networks for phone recognition," in *Proceedings of Interspeech*, 2014, pp. 1078–1082.

[25] T. Grósz, R. Busa-Fekete, G. Gosztolya, and L. Tóth, "Assessing the degree of nativeness and Parkinson's condition using Gaussian Processes and Deep Rectifier Neural Networks," in *Proceedings of Interspeech*, Sep 2015, pp. 1339–1343.

[26] G. Gosztolya, T. Grósz, R. Busa-Fekete, and L. Tóth, "Detecting the intensity of cognitive and physical load using AdaBoost and Deep Rectifier Neural Networks," in *Proceedings of Interspeech*, Singapore, Sep 2014, pp. 452–456.

[27] L. Tóth, G. Gosztolya, V. Vincze, I. Hoffmann, G. Szatlóczki, E. Biró, F. Zsura, M. Pákáski, and J. Kálmán, "Automatic detection of mild cognitive impairment from spontaneous speech using ASR," in *Proceedings of Interspeech*, Dresden, Germany, Sep 2015, pp. 2694–2698.

[28] Gy. Kovács and L. Tóth, "Joint optimization of spectro-temporal features and Deep Neural Nets for robust automatic speech recognition," *Acta Cybernetica*, vol. 22, no. 1, pp. 117–134, 2015.

[29] T. Grósz and L. Tóth, "A comparison of Deep Neural Network training methods for Large Vocabulary Speech Recognition," in *Proceedings of TSD*, Pilsen, Czech Republic, 2013, pp. 36–43.

[30] M. Bacchiani, A. Senior, and G. Heigold, "Asynchronous, online, GMM-free training of a context dependent acoustic model for speech recognition," in *Proceedings of Interspeech*, Singapore, Singapore, Sep 2014, pp. 1900–1904.

[31] H. Su, G. Li, D. Yu, and F. Seide, "Error back propagation for sequence training of context-dependent deep networks for conversational speech transcription," in *Proceedings of ICASSP*, 2013, pp. 6664–6668.