BUILDING CONTEXT-DEPENDENT DNN ACOUSTIC MODELS USING KULLBACK-LEIBLER DIVERGENCE-BASED STATE TYING

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

[1]MTA-SZTE Research Group on Artificial Intelligence, Szeged, Hungary
[2]Idiap Research Institute, Martigny, Switzerland
{ ggabor, groszt, tothl } @ inf.u-szeged.hu, dimseng@idiap.ch

ABSTRACT

Deep neural network (DNN) based speech recognizers have recently replaced Gaussian mixture (GMM) based systems as the state-of-the-art. HMM/DNN systems have kept many refinements of the HMM/GMM framework, even though some of these may be suboptimal for them. One such example is the creation of context-dependent tied states, for which an efficient decision tree state tying method exists. The tied states used to train DNNs are usually obtained using the same tying algorithm, even though it is based on likelihoods of Gaussians. In this paper, we investigate an alternative state clustering method that uses the Kullback-Leibler (KL) divergence of DNN output vectors to build the decision tree. It has already been successfully applied within the framework of KL-HMM systems, and here we show that it is also beneficial for HMM/DNN hybrids. In a large vocabulary recognition task we report a 4% relative word error rate reduction using this state clustering method.

***Index Terms*—** Speech recognition, deep neural networks, state tying, Kullback-Leibler divergence

1. INTRODUCTION

Deep neural network (DNN) based hybrid speech recognizers are nowadays regarded as the state-of-the-art and have replaced conventional Gaussian mixture modeling (GMM) based hidden Markov models (HMMs). Since the introduction of HMMs, the speech community developed many techniques to optimize the process of the training of GMM-based acoustic models. HMM/DNN hybrid systems have inherited most of these methods, even though some of these may be inappropriate for them. Two such examples are the flat start training scheme and the creation of context-dependent (CD) phone models, which are vital components of conventional HMM/GMM systems.

More specifically, HMM/GMM systems are usually trained by an iterative re-estimation and re-aligment of the models, also known as 'flat start' training. Since it is not obvious how to perform such a flat start training with HMM/DNN-based acoustic models, most HMM/DNN systems are trained on frame-level labels that were obtained from a previously trained HMM/GMM system using forced alignment. Although sequence-based training strategies have begun to emerge, these still give better results when initialized with frame-level training [1]. Quite recently, it was shown by several researchers that, if done with proper caution, flat start training can also be performed with DNNs [2, 3].

While hybrid models applied only context-independent (CI) phone models for a long time [4], there is now common agreement that HMM/DNN systems also greatly benefit from using context-dependent tied states [5, 6]. Thus, it is necessary to find an approach for efficiently creating context-dependent tied states in DNN systems. However, this seems to be a more challenging task than flat start training.

Currently, the dominant solution is the decision tree-based state tying method [7]. This technique fits Gaussians on the distribution of the states, and uses the likelihood gain to govern the state-splitting process. Thanks to the Gaussian assumption and the decision tree representation, this approach is computationally very efficient. However, as already mentioned, sometimes it may be inappropriate to just impose the common HMM/GMM-based techniques on the HMM/DNN training procedure. For several reasons, such as the usage of different features and the fact that the objective functions during training are completely different, this may be so for the state tying approach.

GMM-based methods assume that the Gaussian components have diagonal covariance matrices, and thus require decorrelated features like cepstral coefficients (MFCCs). However, it was observed that HMM/DNN hybrids work better on more primitive features like mel filter bank energies [8]. Since conventional HMM/GMM systems cannot be efficiently trained on these features, one would have to train a HMM/GMM system on a standard feature set like MFCCs, create the tied state inventory and alignment, and then discard

This publication is supported by the European Union and co-funded by the European Social Fund. Project title: Telemedicine-oriented research activities in the fields of mathematics, informatics and medical sciences. Project number: TÁMOP-4.2.2.A-11/1/KONV-2012-0073

the feature set. Therefore, it may be better to perform the state clustering not on the raw features, but on the output of a DNN. This approach was investigated by Senior et al [2]. A simple modification of this method is to use the activations on the last hidden layer of the DNN instead of the outputs of the final softmax layer [9]. In a similar study, Zhang et al. derived formulas for converting the output of the DNN softmax layer or a hidden network layer into class-conditional Gaussian distributions [3]. Note that all these studies manipulated only the input of the clustering algorithm, but for the clustering they used the same standard Gaussian-based decision tree clustering method.

A second argument is that, intuitively, the state clustering algorithm should split those states where the splitting would be beneficial for the respective classifier. Since the objective functions during GMM and DNN training fundamentally differ, measuring how a Gaussian models a given class may be unrelated to the difficulty of modeling that class by a DNN that is able to represent much more complex decision boundaries. This suggests that some metric other than the likelihood of Gaussians should be used by the state clustering process.

Recently, a variant of the decision tree clustering algorithm was proposed that also works on DNN output vectors [10]. In contrast to the earlier cited studies which converted the DNN outputs into class conditional distributions and fit Gaussians on these, this algorithm exploits the fact that the DNN output vectors form discrete probability distributions. A natural distance function for such distributions is the Kullback-Leibler divergence [11]. Hence, it is reasonable to modify the state clustering algorithm so that it works with the Kullback-Leibler divergence instead of Gaussian likelihoods. Imseng et al. successfully used this KL-divergence based state tying routine in the framework of Kullback-Leibler divergence-based HMMs (KL-HMM) [12].

In this paper, we investigate the applicability of this algorithm for creating tied states in a HMM/DNN hybrid. The evaluation will be carried out on a large vocabulary speech recognition task of 28 hours of Hungarian broadcast news data. As a baseline, a context dependent HMM/DNN hybrid that applies conventional GMM-based state tying is used. Then we repeat the same experiments by training a context independent (CI) auxiliary neural network, and then create context-dependent (CD) states by applying the modified, KL-divergence based clustering code on the network output.

2. DECISION TREE BASED STATE TYING

The decision tree-based state tying algorithm was introduced by Young et al. [7], and evolved into a vital component of training large vocabulary speech recognizers. The main idea is to pool all context variants of a state, and then build a decision tree by successively splitting this set into two. For each step, the algorithm chooses one of the pre-defined questions in such a way that the resulting two non-overlapping sub-sets of the original state set $\mathcal{S}$ differs maximally. The algorithm measures this difference by using a likelihood-based decision criterion. Although minor improvements to the algorithm like the automatic generation of the questions via clustering were proposed [13], the main scheme of the method proved so successful that it has remained unaltered ever since.

2.1. Likelihood based decision criterion

Odell formulated a maximum likelihood-based decision criteria [14] and proposed a computationally efficient algorithm by approximating the splitting criterion as

$$L(\mathcal{S}) \simeq -\frac{1}{2}\big(\log[(2\pi)^K|\Sigma(\mathcal{S})|] + K\big)\sum_{s\in\mathcal{S}} N(s), \quad (1)$$

where $s \in \mathcal{S}$ are the individual states, $\Sigma(\mathcal{S})$ is the variance of data in $\mathcal{S}$, and $N(s)$ is the number of examples (frames) in the training data which belong to state s. Using this formula, we should choose the question q which maximizes the likelihood difference $\Delta L(q|\mathcal{S})$

$$\Delta L(q|\mathcal{S}) = \big(L(\mathcal{S}_y(q)) + L(\mathcal{S}_n(q))\big) - L(\mathcal{S}), \quad (2)$$

where $\mathcal{S}_y(q)$ and $\mathcal{S}_n(q)$ are the two subsets of $\mathcal{S}$ formed based on the answer to the question q. It can be seen that the likelihood values do not depend on the training observations themselves, but only on the variance over training data corresponding to the states, and the raw number of frames belonging to each state. Although this assumption (regarding the variance of the feature vectors) fits well to a system employing GMMs, in a HMM/DNN hybrid speech recognizer framework some other decision criterion might result in a more suitable set of tied states.

2.2. Kullback-Leibler divergence based decision criterion

This decision criterion was introduced by Imseng et al., who successfully applied it in their KL-HMM framework [15]. Next, we will give a brief description of this algorithm, based on articles [10] and [12].

Although the Kullback-Leibler divergence is known to be asymmetric, unfortunately there is no closed form of the symmetric KL-divergence based cost function. Therefore we will apply the asymmetric KL-divergence between two posterior vectors z_t and y_s, defined as

$$D_{KL}(y_s||z_t) = \sum_{k=1}^{K} y_s(k)\log\frac{y_s(k)}{z_t(k)}, \quad (3)$$

where $k \in \{1, \ldots, K\}$ is the dimensionality index of the posterior distribution vector [11]. The KL-divergence is always non-negative and zero if and only if the two posterior vectors are equal. So instead of maximizing the likelihood, we will minimize the KL-divergence

$$D_{KL}(\mathcal{S}) = \sum_{s \in S} \sum_{f \in F(s)} \sum_{k=1}^{K} y_{\mathcal{S}}(k) \log \frac{y_{\mathcal{S}}(k)}{z_f(k)}, \quad (4)$$

where $\mathcal{S}$ is a set of states s, and $F(s)$ is the set of training vectors corresponding to state s. The posterior vector associated with the set $\mathcal{S}$ ($y_{\mathcal{S}}$) can be calculated as the normalized geometrical mean of the example vectors belonging to the elements of $\mathcal{S}$, i.e.

$$y_{\mathcal{S}}(k) = \frac{\left(\prod_{s \in \mathcal{S}} \prod_{f \in F(s)} z_f(k)\right)^{\frac{1}{N(\mathcal{S})}}}{\sum_{k=1}^{K} \tilde{y}_{\mathcal{S}}(k)}. \quad (5)$$

After expanding and simplifying, we get [10]

$$D_{KL}(\mathcal{S}) = -\sum_{s \in \mathcal{S}} N(s) \log \sum_{k=1}^{K} \tilde{y}_{\mathcal{S}}(k), \quad (6)$$

so the KL divergence of a set of states $\mathcal{S}$ can be calculated based on the statistics y_s and $N(s)$ of the individual states.

For the splitting of a set of states $\mathcal{S}$, the straightforward option is to choose the question that maximizes the KL-divergence difference $\Delta D_{KL}(q|\mathcal{S})$:

$$\Delta D_{KL}(q|\mathcal{S}) = D_{KL}(\mathcal{S}) - \left(D_{KL}(\mathcal{S}_y(q)) + D_{KL}(\mathcal{S}_n(q))\right).$$

3. APPLYING KL-BASED STATE TYING FOR HMM/DNN HYBRIDS

In our baseline system context-dependent HMM/GMM phone models are trained first, which are then used in force aligment mode to generate CD training labels for the DNN. This system operates on MFCC features, and was implemented in HTK [16]. It applies the standard, Gaussian-based state tying process as part of the training process of the HMM/GMM CD phone models. Having obtained the clustered states using the HMM/GMM, a DNN is trained using these tied states as the training labels. This DNN is used during the decoding process, which is preformed by applying a modified version of the HTK Hdecode routine [16].

The KL divergence-based clustering algorithm requires CI state label posterior estimates as its input. To get these, we trained an auxiliary neural network with one hidden layer (ANN) on the CI labels got from the HMM/GMM system. Next, we applied the KL-divergence based clustering algorithm on the output of this ANN. Then, having obtained the clustered states, we trained the DNN using these tied states as the training labels. Similar to the baseline system, the decoder used this DNN during recognition.

4. EXPERIMENTAL SETUP

As the DNN component of our hybrid recognizer, we applied a deep network consisting of rectified linear units as hidden neurons [17]. The main advantage of deep rectifier nets is that they can be efficiently trained with the standard backpropagation algorithm, without any tedious pre-training [18]. We used our custom implementation, which achieved the best accuracy known to us on the TIMIT database with a phone error rate of 16.7% on the core test set [19].

For the actual task we employed a DNN with 5 hidden layers, each containing 1000 rectified neurons. In the output layer, we applied the softmax function. We used 40 mel filter bank energies as features along with their first and second order derivatives; following the HTK notation, we will refer to this feature set as the FBANK feature set. Decoding and evaluation was performed by a modified version of HTK [16].

The speech corpus of Hungarian broadcast news was collected from eight TV channels. From the 28 hours of recordings, 22 hours were used as the train set, 2 hours for development and 4 hours for testing. The total number of different triphone occurrences was 13,467, resulting in 40,401 initial CD phone models. We built a trigram language model from a corpus of about 50 million words taken from the www.origo.hu news portal, using the language modelling tools of HTK [16]. As Hungarian is an agglutinative language with a lot of word forms, the recognition dictionary consisted of 486,982 words.

Though the DNN was always trained on the FBANK feature set, we investigated two variants of the auxiliary ANN. First, we trained it on the MFCC feature set that was also used by the HMM/GMM system. The second version was trained on the FBANK feature set that was utilized by the DNN. Note that for the baseline system we had to use different feature sets to construct the tied states and for learning them by the DNN (MFCCs vs. FBANK), as training GMMs on FBANK features would have produced unusable results. As the auxiliary ANN was thrown away after state tying, it contained only one hidden layer. We will examine the relevance of the size of this network later on.

For both clustering algorithms, we varied the state tying stopping threshold to get roughly 600, 1200, 1800, 2400, 3000 and 3600 tied states.

5. RESULTS

As shown in figures 1 and 2, the KL divergence-based clustering method performed consistently and significantly better than the conventional GMM/HMM clustering on both sets. The standard algorithm gives the best performance with 600 tied states, though the results are roughly the same across all state values. The KL-divergence based system has a clear optimum at 1200 states, and it yielded a 4% relative error rate reduction compared to the best score of the conventional system. Among the two variants of the auxiliary ANN, the one trained on FBANK features – that is, the same feature set that the final DNN was trained on – led to somewhat better scores, though the difference was not significant.

An obvious drawback of the KL state tying approach over

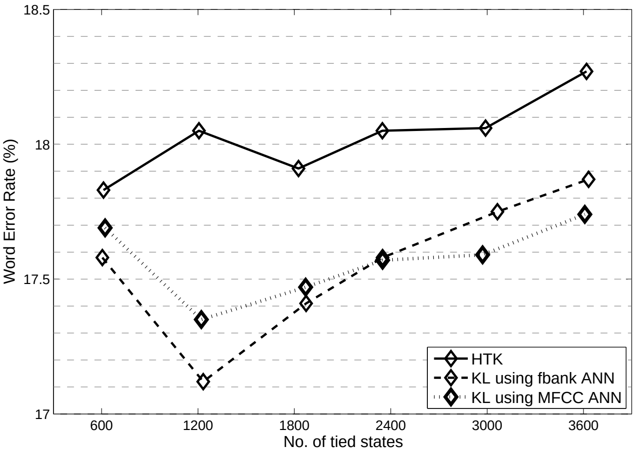

Fig. 1. Word error rates as a function of the number of tied states on the development set.

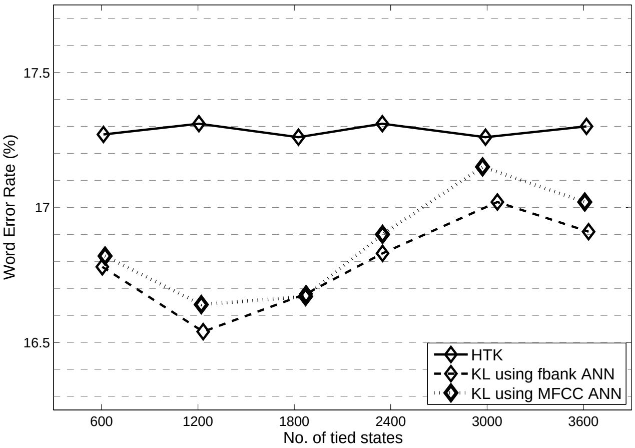

Fig. 2. Word error rates as a function of the number of tied states on the test set.

the conventional algorithm is that we first need to train the auxiliary ANN to obtain the input of clustering. The GMM-based method uses Gaussians for the same purpose, and fitting these on the data is much faster than training an ANN. While it is obvious that the result of the KL divergence-based method depends on the auxiliary ANN, it is not at all clear how accurate this network should be. Perhaps a much smaller network could also lead to similar results, while the training time could be considerably reduced. To discover if this is so, we repeated the experiment by varying the size of the hidden layer of the auxiliary ANN. The clustering step and the training of the DNN was the same as in the previous experiments. The state clustering algorithm was configured so as to get roughly 1200 tied states, as this value gave the best performance earlier.

No. of hidden neurons	WER %	
	Dev. set	Test set
500	17.38%	16.76%
1000	17.12%	16.54%
2000	17.43%	16.44%

Table 1. *Word error rates as a function of hidden layer size in the auxiliary ANN.*

The word error rates for different network sizes are given in Table 1. Although the size of the hidden layer of the ANN affected the WER scores, the difference is minimal, and even the worst scores are much better than the ones obtained via the GMM-based state tying method. This fact tells us that the KL-clustering algorithm can give good results even when the auxiliary network is much smaller than the final DNN.

Besides keeping the auxiliary ANN as small as possible, there is a further option available to reduce the training time. Here the idea is to keep the weights of the auxiliary ANN, and use them to initialize the lowest hidden layer of the DNN. This might reduce the training time on one hand, and yield slightly better results on the other. Of course, in this case the ANN must have the same number of hidden units as the final DNN, which in our case was set to 1000 units. The results are listed in Table 2 below. Unfortunately, the accuracy of the system trained this way was no better than the previous scores. Further studies are required to see whether this could be improved if the auxiliary ANN contained more hidden layers.

State tying method	WER %	
	Dev. set	Test set
KL with MFCC ANN	17.35%	16.64%
KL with fbank ANN	17.12%	16.54%
KL with fbank ANN + ANN init.	17.38%	16.79%
GMM/HMM clustering	17.83%	17.26%

Table 2. *Word error rates for the different training strategies.*

6. CONCLUSIONS

We evaluated a state clustering algorithm that is based on the KL-divergence of posterior probability distributions. Compared to the standard method that uses the likelihood of Gaussians, this algorithm seems to be more plausible and appropriate when the input data to be clustered are ANN output vectors. Indeed, there is experimental evidence that the KL-based algorithm to create the CD targets of a HMM/DNN hybrid yields slightly better recognition scores with the same number of tied states. In a large vocabulary recognition task we reported a 4% relative word error rate reduction compared to that for the standard state clustering method.

7. REFERENCES

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

[2] A. Senior, G. Heigold, M. Bacchiani, and H. Liao, "GMM-free DNN training," in *Proceedings of ICASSP*, 2014.

[3] C. Zhang and P. Woodland, "Standalone training of context-dependent Deep Neural Network acoustic models," in *Proceedings of ICASSP*, 2014, pp. 5597–5601.

[4] H. Bourlard and N. Morgan, *Connectionist Speech Recognition – A Hybrid Approach*, Kluwer Academic, 1994.

[5] D. Yu, L. Deng, and G. Dahl, "Roles of pretraining and fine-tuning in context-dependent DNN-HMMs for real-world speech recognition," in *Proceedings of NIPS Workshop on Deep Learning and Unsupervised Feature Learning*, 2010.

[6] G. Dahl, D. Yu, L. Deng, and A. Acero, "Context-dependent pre-trained Deep Neural Networks for large vocabulary speech recognition," *IEEE Trans. ASLP*, vol. 20, no. 1, pp. 30–42, 2012.

[7] 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.

[8] A. Mohamed, G. E. Dahl, and G. Hinton, "Acoustic modeling using deep belief networks," *IEEE Trans. ASLP*, vol. 20, no. 1, pp. 14–22, 2012.

[9] M. Bacchiani and D. Rybach, "Context dependent state tying for speech recognition using deep neural network acoustic models," in *Proceedings of ICASSP*, 2014, pp. 230–234.

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

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

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

[13] K. Beulen and H. Ney, "Automatic question generation for decision tree based state tying," in *Proceedings of ICASSP*, 1998, pp. 805–808.

[14] J.J. Odell, *The Use of Context in Large Vocabulary Speech Recognition*, Ph.D. thesis, University of Cambridge, 1995.

[15] M. Razavi, R. Rasipuram, and M. Magimai-Doss, "On modeling context-dependent clustered states: Comparing HMM/GMM, hybrid HMM/ANN and KL-HMM approaches," in *Proceedings of ICASSP*, 2014.

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

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

[18] L. Tóth, "Phone recognition with deep sparse rectifier neural networks," in *Proceedings of ICASSP*, 2013, pp. 6985–6989.

[19] L. Tóth, "Combining time- and frequency-domain convolution in convolutional neural network-based phone recognition," in *Proceedings of ICASSP*, 2014, pp. 190–194.