Investigation of Mixture Splitting Concept for Training Linear Bottlenecks of Deep Neural Network Acoustic Models

Muhammad Ali Tahir[1], *Simon Wiesler*[1], *Ralf Schlüter*[1], *Hermann Ney*[1,2]

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

ABSTRACT

A Gaussian or log-linear mixture model trained by maximum likelihood may be trained further using discriminative training. It is desirable that the mixture splitting is also done during the discriminative training, to achieve better mixture density distribution. In previous work such a discriminative splitting approach was presented. Similarly, the resolution of a deep neural network may also be increased by splitting. In this paper, discriminative splitting is applied as a way of initializing a linear bottleneck between two layers of a DNN. Experiments for a single hidden layer and six hidden layer cases show the potential of this approach as an alternative method of pre-training for linear bottlenecks for MLP hidden layers.

Index Terms— mixture splitting, deep neural network, linear bottleneck

1. INTRODUCTION

The task of speech recognition is to convert a given audio stream into text in a particular language. For a long time the standard technique for acoustic modelling has been a hidden Markov model (HMM) based Gaussian mixture model (GMM). This GMM is commonly trained by a maximum likelihood (ML) training procedure. However, ML training makes some assumptions about the distribution which in practice may not be true, because the amount of data is limited. Another training paradigm called discriminative training takes the competition between classes into account, and has shown to outperform the ML-based acoustic model training. Thus, using an initial ML-trained GMM and correspondingly created HMM-to-acoustic-feature alignment, the GMM model can be further trained by a discriminative criterion such as MMI (maximum mutual information) or MPE (minimum phone error) [1] [2]. However, a more direct approach would be to train an acoustic model completely using discriminative training procedure. Some discriminative acoustic models such as maximum entropy Markov models (MEMM) [3] and conditional random fields [4] have been proposed in the literature. These acoustic models can be completely trained with a discriminative criterion, only utilizing the ML training for a feature-to-HMM-state alignment. The log-linear mixture model (LLMM) model also incorporates mixtures like a standard GMM, and due to its equivalence with a GMM [5], it has been initialized from GMM models. In previous work [6] it was shown that an LLMM can be directly estimated without initialization from GMM model. This is done by discriminative splitting of LLMM parameters during the discriminative training.

More recently, multilayer-perceptron and deep neural networks [7] have been increasingly popular to model the emission posterior probabilities of HMM phone states. Since the log-linear models have a somewhat similar structure to an MLP network (it is equivalent to a layer with softmax activation function), therefore we aim to extend the results of log-linear discriminative splitting to deep neural networks. Experiments show that the discriminative splitting can be used to estimate a sparse linear transformation between two layers of a deep MLP, whose performance compares favourably with other procedures. Secondly, this aforementioned transformation can be used to initialize a non-sparse linear transformation, which can be employed as an effective pre-training method.

The remainder of the paper is organized as follows; Section 2 reviews the discriminative splitting of mixture densities for hidden conditional random fields. Section 3 describes the application of this concept to initialization of a deep neural network i.e. initializing a higher resolution model from a lower resolution one. Some results for splitting of deep neural networks are presented in Section 4 and finally a conclusion and outlook are provided in Section 5.

2. DISCRIMINATIVE SPLITTING OF LOG-LINEAR MIXTURE MODEL

In [5] log-linear models were introduced as an alternative to the Gaussian distributions, for representing the emission probabilities of HMM states. Under assumption of a pooled covariance matrix, the posterior probabilities of Gaussian single density HMMs are equivalent to corresponding log-linear models' posterior probabilities. It is therefore possible to convert the Gaussian parameters to log-linear and vice-versa. The posterior probability of a feature vector x with respect to a phonetic class s is defined as

$$p_\theta(s|x) = \frac{\exp(\lambda_s^\top x + \alpha_s)}{\sum_{s'} \exp(\lambda_{s'}^\top x + \alpha_{s'})} \tag{1}$$

Here the state parameters are $\Lambda_s = \{\lambda_s, \alpha_s\}$. The equivalence of Gaussian and log-linear posterior probabilities can also be extended to Gaussian mixture models. Here the hidden variables are the mixture components of the state-emission probability. The corresponding posterior probability is

$$p_\theta(s|x) = \frac{\sum_l \exp(\lambda_{s,l}^\top x + \alpha_{s,l})}{\sum_{s',l} \exp(\lambda_{s',l}^\top x + \alpha_{s',l})} \tag{2}$$

for $l = 1...L_s$ mixture parameters in each class s.

For discriminative training, the frame level objective function is

$$\mathcal{F}^{(frame)}(\Lambda, A) = -\tau_\Lambda ||\Lambda||^2 + \sum_{r=1}^{R} \sum_{t=1}^{T_r} w_s \log p_\Lambda(s_t|x_t) \tag{3}$$

for a fixed feature vector to state alignment s_1^T. τ_Λ is a regularization parameter. w_s are state weights which could be tuned to give less weight to the accumulations of e.g. noise and silence states. $\hat{\alpha}_s = \alpha_s + \log p(s)$, $p(s)$ is the prior probability of state s and R is the total number of sentences in the training corpus. The state priors are added to α_s for training, and later subtracted at recognition time. The objective function is a frame-level MMI, with extra regularization terms. The MMI optimization can also be done at sentence level i.e. the prior probabilities are sentence level language-model probabilities.

2.1. Splitting procedure for LLMM

The log-linear training is only convex for a single density per state s. For mixture density training this presents challenges as the initial guess is very important and can influence the final results for the objective function and WER. Therefore we need a method to specify a better initial guess to the training of mixture densities, so that the WER is at least as good as the word error rate of a similar but less complex model. To solve this problem we adopt an approach similar to the iterative density splitting algorithm used in a maximum likelihood framework, but applied to the log-linear parameters $\lambda_{s,l}$ instead of the means, as in the Gaussian mixtures case. All the $\lambda_{s,l}$ in state s are duplicated and a small offset is added to both new λ's to pull them apart. The log-linear model is covariance normalized, therefore the direction of the offset is not important. Subsequent training of this newly split model causes an increase in the objective function as the new λ's discriminatively adapt themselves to the training data. This successive discriminative training and splitting can be repeated several times until the desired model resolution is achieved.

3. DISCRIMINATIVE SPLITTING FOR DEEP NEURAL NETWORKS

3.1. Deep Neural Networks

Neural networks have become an important tool to estimate HMM state posterior probabilities in acoustic modelling, especially with the application of deep learning concept in the last few years. A neural network for speech recognition is usually a multilayer-perceptron (MLP), with the hidden layers having a sigmoid or another activation function and the output layer having a softmax activation function. There are two ways of applying neural networks for acoustic modelling: tandem and hybrid MLPs. A tandem MLP system has a bottleneck layer as the last output layer of the network and then a regular GMM based maximum likelihood acoustic model is trained on top of it. On the other hand, a hybrid MLP system directly uses the posterior probabilities of MLP network as acoustic model probabilities. In this work the application of discriminative splitting for hidden layers of an MLP is discussed, which implies that these concepts can be used for tandem as well as hybrid approaches.

3.2. Linear bottlenecks for DNNs

Two consecutive layers of an MLP network that are fully connected may have redundancy in the structure. Many of the elements in a layer may have a negligibly small effect on the output of that layer. If the number of elements in a layer can be reduced by removing those redundancies while not compromising the classification performance, it can provide large decreases in time and memory requirements of MLP training. Several methods have been proposed to achieve this compression. [8] have reduced the number of elements in the layers by removing the close to zero elements and converting the matrices to an index-based representation. [9] do a low-rank factorization of the final weight layer and report a decrease of 30-50% in the number of parameters without a significant loss in final recognition accuracy. [10] have factored the weight matrix into a product of two smaller matrices, providing parameter compression. They have reported encouraging results by doing a singular value decomposition (SVD) based factorization between the hidden layers. The error rate degrades at first but after doing a full network training with back-propagation, the classification performance of the MLP network is restored. [11] have proposed a training mechanism whereby a hidden layer and its low-rank factorization can be

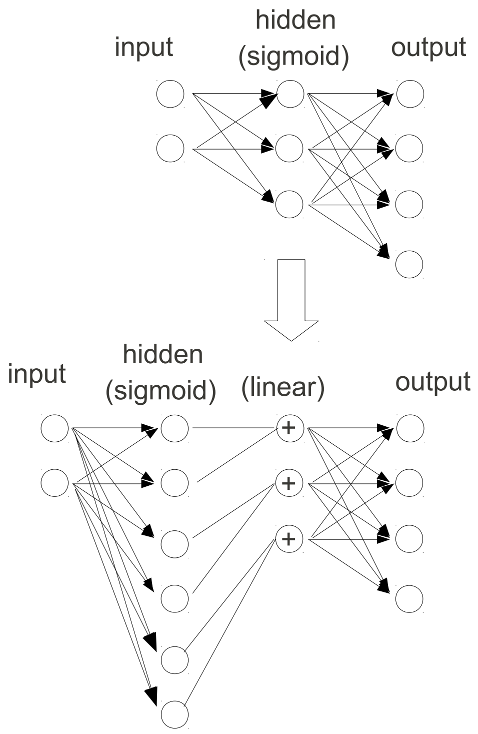

Fig. 1. *Flow diagram of discriminative splitting of MLP hidden layer*

simultaneously trained from scratch. Apart from model parameter reduction, they report an added benefit of regularization from this factorization. Thus the linear bottleneck can reduce over-training of MLP network parameters.

3.3. Discriminative Splitting for DNNs

In this paper a method for training a linear bottleneck between two MLP layers is investigated, which is inspired by mixture density training of GMM acoustic models. For GMM initially a single density is trained for each tied context-dependent state. This density is then split iteratively into a successively larger number of mixture densities until the desired parameter resolution is achieved. The final class conditional probability is a weighted sum of all the respective densities in that mixture. A similar approach has also been employed in Section 2 for discriminative training of log-linear mixture acoustic models, where the state posterior probabilities have been successively split during the discriminative training. Such a method could in principle also be used for a hidden layer in an MLP network; a hidden layer is trained and then all the nodes and their weight parameters are duplicated with some random offset. These duplicated and offset copies of each node are being summed up into the original node, thus converting the original hidden layer into a linear bottleneck with a new larger hidden layer behind it. The feasibility of such a layer splitting method is investigated. This process is illustrated graphically in Figure 1.

4. EXPERIMENTS AND RESULTS

4.1. Speech Corpus and Baseline System

For the performance analysis of discriminative splitting, the large vocabulary continuous speech recognition task QUAERO English is used. It is composed of podcasts, television news broadcasts and debates under a range of clean to noisy conditions. The training corpus is a 50 hour subset of the total QUAERO English data and the development and evaluation corpora are 3 hours each.

The acoustic model of the baseline system uses cross-word triphones. The lexicon contains 54k words and a trigram language model is used. The classes are 4501 triphone CART leaves and a pooled covariance is used.

For the input MFCC features the feature vector length is 29, and 17 consecutive frames are appended together. The MLP network therefore has 493 input features. The number of nodes in the output softmax layer is 4501 (No. of CART states). The hidden layers have a sigmoid activation function. The number of hidden layers and number of nodes in each layer will be varied during the course of these experiments. First, the results for a single hidden layer are presented, and then for the case of six hidden layers (best system configuration).

Table 1. *Comparison of full hidden layer and linear mixture bottleneck, w.r.t. WER and no. of parameters*

size of hidden layer	size of linear (mixture) bottleneck	No. of params.	WER (CER) % dev	eval
1024	-	5.1M	22.7 (14.7)	28.9 (19.0)
2048	1024	5.6M	22.1 (14.4)	28.4 (18.5)
8192	1024	8.7M	21.2 (13.6)	27.7 (17.4)
2048	no BN	10M	21.8 (14.2)	28.4 (18.1)
8192	no BN	41M	20.6 (12.9)	27.1 (16.9)

4.2. MLP with a single hidden layer

For a single hidden layer MLP, initially a network with 1024 hidden (sigmoid) layer nodes is trained. This gives a WER of 22.7% on the development corpus. Then each node in the hidden layer is duplicated such that each two consecutive nodes sum up into one node in the following linear bottleneck layer. A small random offset is added to each node's parameters, therefore the objective function (cross-entropy) remains roughly the same. The network is then further trained so that the increased resolution model adapts itself to the training

Table 2. *Comparison of sparse mixture bottleneck with full linear bottleneck, as well as random initialization vs. splitting*

size of hidden layer	size of linear bottleneck	type of bottleneck	initialization type	No. of params.	WER (CER) % dev	WER (CER) % eval
128	no bottleneck			0.6M	32.1 (21.0)	39.6 (28.8)
2048	no bottleneck			10M	21.8 (14.2)	28.4 (18.1)
2048	128	mixture	splitting	1.6M	24.2 (14.7)	31.0 (21.0)
2048	128	full	splitting	1.9M	21.8 (13.0)	28.0 (18.1)
2048	128	full	pre-training	1.9M	22.1 (14.5)	29.0 (18.5)

data. The WER decreases to 22.1%. Splitting the hidden layer further to 8192 nodes brings the WER down to 21.2%. This is 0.6% better than the WER with 2048 hidden layer size with no linear bottleneck, with only 87% as much parameters. This is shown in Table 1 (CER refers to character error rate).

It is important to note that splitting the hidden layer and adding a mixture layer after it does not increase the number of (non-linear) layers. It is basically a matrix factorization.

The linear mixture bottleneck as described above contains only one matrix (the other matrix is sparse consisting of only zeros and ones in a particular order, hence having no trainable/storable parameters). What if we convert this sparse matrix into a full matrix and train it further using the sparse representation as initial guess? Would it be able to perform better than a linear bottleneck initialized from discriminative pre-training (random initialization)? As shown in Table 2 a full linear bottleneck (two matrices) initialized from discriminative splitting has a 2.4% better WER (dev) than a mixture bottleneck (with only one non-sparse matrix). It is also 0.3% better than a full linear bottleneck initialized from discriminative pre-training. This shows that the discriminative splitting can be used as an effective pre-training method in cases where linear bottlenecks are involved between the layers of a neural network. Furthermore, by using a smaller model as an initial guess, the number of training iterations is also decreased as compared to random initialization as for regular pre-training.

4.3. MLP with six hidden layers

The discriminative splitting method for one hidden layer (as in previous section) can easily be extended to a deep neural network scenario. Table 3 shows the WER for a 6 hidden layer deep neural network for the same task and configuration as in the previous section. The details of this DNN setup can be found in [11]. Each hidden layer has 2048 nodes. There is a linear bottleneck of size 256 between every two hidden layers and between the last hidden layer and output layer. The parameters of these linear bottleneck layers are initialized by discriminative pre-training with random initialization. In our experiment, the network is initialized first by pre-training a small MLP of 256 hidden nodes and then discriminatively splitting it to 2048 nodes. The comparison between these two initialization types is shown in Table 3. It can be seen that the system with splitting based initialization has 0.2% absolute WER improvement over pre-training. This shows that the potential of discriminative splitting applies to deep MLPs too.

Table 3. *Comparison of random initialization pre-training and discriminative splitting, for the same deep MLP network*

initialization type	No. of params.	WER (CER) % dev	WER (CER) % eval
pre-training	4.9M	18.2 (10.9)	24.0 (14.8)
splitting	4.9M	18.0 (10.7)	23.8 (14.8)

5. CONCLUSION AND OUTLOOK

In this paper discriminative splitting is presented as an approach to increasing the model resolution during the discriminative training process. For training a log-linear mixture acoustic model, instead of using ML split mixture models one could perform the splitting during the MMI training. Similarly, the resolution of a deep neural network may also be increased by discriminative splitting method. Experiments for a single hidden layer and six hidden layer cases show the potential of this approach as an alternative method of pre-training for linear bottlenecks for MLP hidden layers.

6. ACKNOWLEDGEMENTS

The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement No. 287658 (EU-Bridge), and under grant agreement 287755 (TransLectures). H. Ney was partially supported by a senior chair award from DIGITEO, a French research cluster in Ile-de-France.

7. REFERENCES

[1] D. Povey and P.C. Woodland, "Minimum phone error and i-smoothing for improved discriminative training," in *Acoustics, Speech, and Signal Processing (ICASSP), 2002 IEEE International Conference on*, May 2002, vol. 1, pp. I–105–I–108.

[2] Xiadong He and Li Deng, *Discriminative learning for speech recognition theory and practice*, [San Rafael, Calif.] : Morgan & Claypool, 2008.

[3] Hong-Kwang Jeff Kuo and Yuqing Gao, "Maximum entropy direct models for speech recognition," *Trans. Audio, Speech and Lang. Proc.*, vol. 14, no. 3, pp. 873–881, Dec. 2006.

[4] Asela Gunawardana, Milind Mahajan, Alex Acero, and John C. Platt, "Hidden conditional random fields for phone classification," in *International Conference on Speech Communication and Technology*. September 2005, International Speech Communication Association.

[5] G. Heigold, R. Schlüter, and H. Ney, "On the equivalence of Gaussian HMM and Gaussian HMM-like hidden conditional random fields," in *Interspeech*, Antwerp, Belgium, Aug. 2007, pp. 1721–1724.

[6] Muhammad Ali Tahir, Ralf Schlüter, and Hermann Ney, "Discriminative splitting of gaussian/log-linear mixture hmms for speech recognition," in *2011 IEEE Workshop on Automatic Speech Recognition & Understanding, ASRU 2011, Waikoloa, HI, USA, December 11-15, 2011*, 2011, pp. 7–11.

[7] Geoffrey E. Hinton, "Learning multiple layers of representation," *Trends in Cognitive Sciences*, vol. 11, no. 10, pp. 428–434, 2007.

[8] Dong Yu, Frank Seide, Gang Li, and Li Deng, "Exploiting sparseness in deep neural networks for large vocabulary speech recognition," in *in Proc. ICASSP*, 2012, pp. 4409–4412.

[9] T.N. Sainath, B. Kingsbury, V. Sindhwani, E. Arisoy, and B. Ramabhadran, "Low-rank matrix factorization for deep neural network training with high-dimensional output targets," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, May 2013, pp. 6655–6659.

[10] Jian Xue, Jinyu Li, and Yifan Gong, "Restructuring of deep neural network acoustic models with singular value decomposition.," in *INTERSPEECH*. 2013, pp. 2365–2369, ISCA.

[11] Simon Wiesler, Alexander Richard, Ralf Schlüter, and Hermann Ney, "Mean-normalized stochastic gradient for large-scale deep learning," in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, Florence, Italy, May 2014, pp. 180–184.