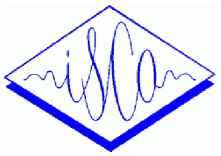

INTERSPEECH 2012
ISCA's 13th Annual Conference
Portland, OR, USA
September 9-13, 2012

Discriminative feature-space transforms using deep neural networks

George Saon and Brian Kingsbury

IBM T. J. Watson Research Center, Yorktown Heights, NY, 10598
e-mail: {gsaon,bedk}@us.ibm.com

Abstract

We present a deep neural network (DNN) architecture which learns time-dependent offsets to acoustic feature vectors according to a discriminative objective function such as maximum mutual information (MMI) between the reference words and the transformed acoustic observation sequence. A key ingredient in this technique is a greedy layer-wise pretraining of the network based on minimum squared error between the DNN outputs and the offsets provided by a linear feature-space MMI (FMMI) transform. Next, the weights of the pretrained network are updated with stochastic gradient ascent by backpropagating the MMI gradient through the DNN layers. Experiments on a 50 hour English broadcast news transcription task show a 4% relative improvement using a 6-layer DNN transform over a state-of-the-art speaker-adapted system with FMMI and model-space discriminative training.

Index Terms: speech recognition, deep neural networks

10.21437/Interspeech.2012-4

1. Introduction

For the past 30 years or so, the de facto standard in acoustic modeling has been hidden Markov models (HMMs) with state-dependent Gaussian mixture models (GMMs) for expressing the distributions of the acoustic feature vectors within each state. Traditionally, the estimation of the GMM parameters (means, variances, mixture weights) is done with maximum likelihood via the EM algorithm. At the turn of the century, there was a significant advance in the estimation of GMM-HMMs through the advent of powerful discriminative training techniques such as maximum mutual information and minimum phone error (MPE) training. Discriminative training can be done either in model space or, as shown more recently, in feature space as in FMPE [1] where the goal is to estimate a transform which maps high-dimensional vectors of Gaussian posteriors to time-dependent offsets which are added to the regular acoustic feature vectors. The projection is trained to enhance the discrimination between correct and incorrect word sequences.

The dominance of GMM-HMMs in acoustic modeling has led over time to an entire "ecosystem" of front-end processing and speaker-adaptation techniques specifically tailored to maximize the recognition performance under this model. Linear transforms such as the semi-tied covariance (STC) transform and maximum likelihood linear regression (MLLR) are prime examples of such techniques that were developed in the context of diagonal-covariance GMMs. Because of this, the status quo was hard to challenge with competing acoustic modeling approaches until very recently.

Following a recipe outlined in [2], the authors of [3] present a 30% relative improvement in word error rate over a discriminatively-trained GMM-HMM on a 300-hour English conversational telephone speech transcription task by using a deep neural network acoustic model. This is the first successful challenge of the status quo on a respectable LVCSR task, and it was made possible by a specific way of training DNNs which was popularized by Hinton [4] in the context of deep belief networks. The approach uses greedy, layer-wise pretraining of the network with either a supervised or unsupervised criterion. This leads to a better starting point in weight space for the optimization, and prevents the supervised training of the final network from being trapped in a poor local optimum.

Another application of neural networks is discriminative feature extraction. In [5], the authors use multi-layer perceptrons to estimate phone posteriors which are transformed and then modeled with conventional GMMs. A refinement to this technique is proposed in [6] where bottleneck features are introduced for LVCSR and are derived from a 5-layer neural network with a constriction in the middle (hidden layer with few units).

A third way of applying neural networks to speech recognition that, to our knowledge, has not been previously considered is to replace the various linear transformations that are present in an ASR system with non-linear counterparts computed by multi-layer perceptrons. This has the advantage that most of the existing ASR system is left unchanged (front-end, back-end) except for the linear transformation module that is being replaced. This minimal change to an existing ASR infrastructure is desirable for a rapid deployment of this technique. Additionally, when using a GMM-HMM acoustic model, all other front-end processing, speaker adaptation and discriminative training techniques remain applicable. This is not the case for neural network acoustic models where speaker adaptation is an ongoing area of research.

In this paper we propose an alternative to the feature-space discriminative transform known as FMMI (or FMPE depending on the training criterion) that consists in a non-linear projection computed by a deep neural network. Like FMMI, the neural network transform takes as input a block of consecutive features and produces a time-dependent offset that is added to the central frame. Unlike FMMI however, there is no need to have a secondary GMM to compute posterior features; the network does the relevant feature extraction. Another difference with FMMI is that the transformation is not factored into separate transforms for posterior projection and temporal context aggregation. These differences result in a transform that is conceptually simpler.

The paper is organized as follows: in section 2 we describe the neural network transform formulation; in section 3 we present some experimental evidence of its utility, and in section 4 we summarize our findings and propose future directions.

2. Neural network FMMI formulation

The acoustic feature vectors $\mathbf{x}_t \in \mathbb{R}^D$ are transformed to

$$\mathbf{y}_t = \mathbf{x}_t + \mathbf{f}(\mathbf{x}_{t-\tau}, \ldots, \mathbf{x}_t, \ldots, \mathbf{x}_{t+\tau}; W) \quad (1)$$

where $\mathbf{f}$ is a vector function of dimension D computed by an L-layer perceptron which is parameterized by the set of weight matrices $W = \{W_1, \ldots, W_L\}$ with W_i of size $n_i \times (n_{i-1}+1)$ where n_i denotes the number of units for layer i. $\mathbf{f}$ is computed by alternating matrix-vector multiplications for the weight matrices and component-wise non-linear activation functions. More precisely, we define $\mathbf{u}_0 = [\mathbf{x}_{t-\tau}^T, \ldots, \mathbf{x}_t^T, \ldots, \mathbf{x}_{t+\tau}^T]^T$ the input to the neural network of dimension $(2\tau+1)D$ and consider feed-forward networks with the recursion

$$\begin{aligned} \mathbf{v}_i &= W_i \begin{bmatrix} \mathbf{u}_{i-1} \\ 1 \end{bmatrix} \\ \mathbf{u}_i &= \sigma(\mathbf{v}_i), \qquad i = 1 \ldots L \end{aligned} \quad (2)$$

where $\mathbf{v}_i$ is the vector of total inputs, $\mathbf{u}_i$ is the output vector, and σ is the component-wise nonlinear activation function. Popular choices for σ are the sigmoid or hyperbolic tangent. With this notation, the function computed by the network is given by the values of the output layer, i.e. $\mathbf{f}(\mathbf{u}_0; W) \triangleq \mathbf{u}_L$.

2.1. Discriminative objective function

The objective function that we consider in this paper is given by the mutual information between the transformed acoustic observation sequence[1] $Y = \mathbf{y}_1 \ldots \mathbf{y}_T$ and the sequence of reference words W^{ref}

$$\begin{aligned} \mathcal{F}(\lambda) &\triangleq \log \frac{P_\lambda(Y, W^{ref})}{P_\lambda(Y)P(W^{ref})} \\ &= \log P_\lambda(Y|W^{ref}) - \log \sum_W P_\lambda(Y|W)P(W) \\ &\triangleq \mathcal{F}^{num}(\lambda) - \mathcal{F}^{den}(\lambda) \end{aligned} \quad (3)$$

which is expressed as the difference of log likelihood functions between the reference word sequence W^{ref} and all possible word sequences W.[2] The functions $\mathcal{F}^{num}(\lambda)$ and $\mathcal{F}^{den}(\lambda)$ correspond to the numerator and denominator terms in the objective function, respectively. Each word sequence is weighted by the language model probability $P(W)$.
$\lambda = \{\pi_i, a_{ij}, \omega_{ij}, \mu_{ij}, \Sigma_{ij}\}$ denotes an N-state, M-mixture component per state GMM-HMM consisting of initial state probabilities $\{\pi_i\}$, state transition probabilities $\{a_{ij}\}$, mixture component weights $\{\omega_{ij}\}$, Gaussian mean vectors $\{\mu_{ij}\}$, and Gaussian covariance matrices $\{\Sigma_{ij}\}$.

2.2. Gradient backpropagation

Since $\mathcal{F}(\lambda)$ is a function of the transformed feature vectors $Y = \mathbf{y}_1 \ldots \mathbf{y}_t$, we compute the derivative with respect to the set of network weights W using the chain rule

$$\frac{\partial}{\partial W}\mathcal{F}(Y;\lambda) = \sum_{t=1}^{T}\sum_{d=1}^{D} \frac{\partial}{\partial y_{td}}\mathcal{F}(Y;\lambda)\frac{\partial y_{td}}{\partial W} \quad (4)$$

[1] We assume a single observation sequence for simplicity.
[2] Not to be confused with the set of network weights.

Additionally, the gradient of the objective function with respect to the transformed feature vectors has a direct and an indirect component because the model parameters are also a function of the transformed feature vectors [1]

$$\frac{\partial}{\partial \mathbf{y}_t}\mathcal{F}(Y;\lambda(Y)) = \underbrace{\frac{\partial \mathcal{F}}{\partial \mathbf{y}_t}}_{\text{direct}} + \underbrace{\frac{\partial \mathcal{F}}{\partial \lambda}\frac{\partial \lambda}{\partial \mathbf{y}_t}}_{\text{indirect}} \quad (5)$$

Furthermore, it can be shown that the direct derivative is

$$\frac{\partial \mathcal{F}}{\partial \mathbf{y}_t}^{\text{direct}} = -\sum_{i=1}^{N}\sum_{j=1}^{M}[\gamma_t^{num}(i,j) - \gamma_t^{den}(i,j)]\Sigma_{ij}^{-1}(\mathbf{y}_t - \mu_{ij}) \quad (6)$$

where $\gamma_t^{num}(i,j)$, $\gamma_t^{den}(i,j)$ represent the posterior probabilities of being in state i, mixture component j at time t given the acoustics according to the numerator and denominator models, respectively. The indirect derivative is obtained by differentiating the objective function with respect to the Gaussian means and variances, which are assumed to be estimated with maximum likelihood on the transformed features. The derivation can be found in [1] (the main difference being that MMI state occupancies are used instead of MPE arc occupancies) and will not be detailed here. The backpropagation recursion at time t can be written as

$$\begin{aligned} \frac{\partial \mathcal{F}}{\partial \mathbf{u}_L} &= \frac{\partial}{\partial \mathbf{y}_t}\mathcal{F}(Y;\lambda(Y)) \\ \frac{\partial \mathcal{F}}{\partial \mathbf{v}_i} &= \sigma'(\mathbf{v}_i) * \frac{\partial \mathcal{F}}{\partial \mathbf{u}_i} \\ \frac{\partial \mathcal{F}}{\partial W_i} &= \frac{\partial \mathcal{F}}{\partial \mathbf{v}_i}[\mathbf{u}_{i-1}^T; 1] \\ \frac{\partial \mathcal{F}}{\partial \mathbf{u}_{i-1}} &= W_i^T \frac{\partial \mathcal{F}}{\partial \mathbf{v}_i}, \qquad i = L \ldots 1 \end{aligned} \quad (7)$$

where "*" denotes component-wise multiplication. The weight matrices are updated with stochastic gradient ascent with learning rate η

$$W_i^{(k+1)} = W_i^{(k)} + \eta\frac{\partial \mathcal{F}}{\partial W_i}, \qquad i = 1 \ldots L \quad (8)$$

3. Experiments

3.1. Experimental setup

We experimented with neural network FMMI transforms on an English broadcast news transcription task. The training data has 50 hours of transcribed audio collected from a variety of sources. Results are reported on the DEV04f test set which contains 22.6K words and approximately 2 hours of audio.

3.1.1. Front-end processing

Speech is parameterized by extracting 13-dimensional VTL-warped PLP cepstral frames every 10ms with mean and variance normalization for every speaker. Every 9 consecutive frames are concatenated and projected down to 40 dimensions by a linear discriminant analysis (LDA) transform. The range of the LDA projection is decorrelated with a global STC transform [7].

Training	WER	BMMI objfun.
ML	23.6%	0.16
FMMI	20.3%	0.18
FMMI-BMMI	18.7%	0.20

Table 1: Word error rates and BMMI objective function values for baseline acoustic models.

Architecture	MSE	BMMI
17x40/1000/40	21.5%	–
17x40/2048/40	21.1%	20.7%
17x40/4096/40	21.1%	–

Table 2: Comparison of word error rates for models with NN-FMMI transforms after MSE pretraining.

3.1.2. Acoustic models

Words in the recognition lexicon are represented as sequences of phones, and phones are modeled with a 3-state left-to-right HMM topology that does not allow state skipping. Context-dependent states are obtained by growing phonetic decision trees which can ask questions within a $\pm$2-phone cross-word context window. The acoustic models have 2200 HMM states and 64 Gaussians/state and are speaker adaptively trained with feature-space MLLR (FMLLR) [7]. At test time, speaker adaptation is performed with VTLN, FMLLR and multiple regression tree-based MLLR transforms. The recognition vocabulary contains 90K words and the decoding is done with a 4-gram language model containing 4M ngrams trained with modified Kneser-Ney smoothing.

3.1.3. Baseline discriminative training

Discriminative training is performed in both feature and model-space using a variant of the MMI criterion called boosted (or margin-based) MMI (BMMI) [8]. The objective function used is a modification of BMMI proposed in [9] which uses a frame-based, state-based loss function instead of a phone-based accuracy measure. For the baseline model, we train two-level FMMI transforms with offset features following the recipe in [10]. The Gaussian posteriors and offset features that are used as input for the first-level transform are provided by 512 diagonal covariance Gaussians. The second-level (or context) transform spans $\pm$8 frames. We used 4 iterations of BMMI for training the transform followed by an additional 4 BMMI iterations for estimating the Gaussian parameters in the resulting FMMI space. In Table 1 we summarize the word error rates and BMMI objective function values for the baseline acoustic models after the different training steps (ML, FMMI, FMMI with model-space BMMI).

3.2. Neural network FMMI experiments

3.2.1. Architecture

The networks that were trained have the following configuration:

- An input layer of dimension 17×40 corresponding to $\pm$8 input frames (same input as regular FMMI)
- A variable number of hidden layers with a hyperbolic tangent non-linearity
- An output layer with 40 units with a hyperbolic tangent non-linearity

3.2.2. Direct training

In an initial experiment we trained two NN-FMMI transforms each with one hidden layer (1000 and 2048 units, respectively) directly with BMMI starting with weights that were initialized to have a uniform distribution over $[-0.01, 0.01]$. Training was done with stochastic updates after each utterance with a fixed learning rate η =1e-5. The BMMI objective function improved from 0.160 to 0.162 for both networks after 20 passes through the training data (epochs). The word error rate obtained with the smaller network is 23.1%. Compared to the values in Table 1, the BMMI objective function improved very little which suggests that the BMMI criterion is hard to optimize and the training gets stuck in a poor local optimum very early.

3.2.3. MSE pretraining using linear FMMI offsets

In a subsequent experiment, we trained the NN-FMMI transforms with minimum squared error between the outputs and the offsets produced by the best linear FMMI transform. The hope was that squared error is easier to optimize and that learning the linear FMMI offsets provides a good starting point for further optimization. We confirm this conjecture in Figure 1 where we plot the BMMI objective function for a 3-layer network (17x40/2048/40) with and without pretraining.

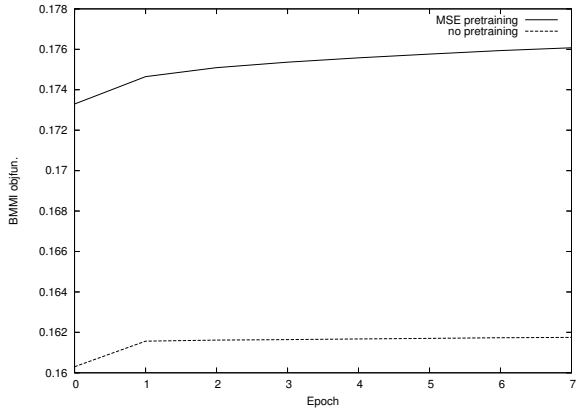

Figure 1: BMMI objective function for NN-FMMI with one hidden layer with and without MSE pretraining.

In Table 2 we indicate the word error rates for three NN-FMMI transforms which differ in the size of the hidden layer after MSE pretraining. We also show the BMMI refinement result for the middle configuration. As can be seen, the performance improvement saturates as more units are added to the hidden layer. This prompted us to look at stacking the layers to create a deep neural network architecture.

3.2.4. Layer stacking

Here, the training proceeds as follows. The network is grown one layer at the time and initialized with the MSE pretrained weights from the previous network. MSE pretraining is done for all the layers, not just for the newly added weights. Once a desired configuration is reached, all the weights are trained

according to the BMMI objective function.

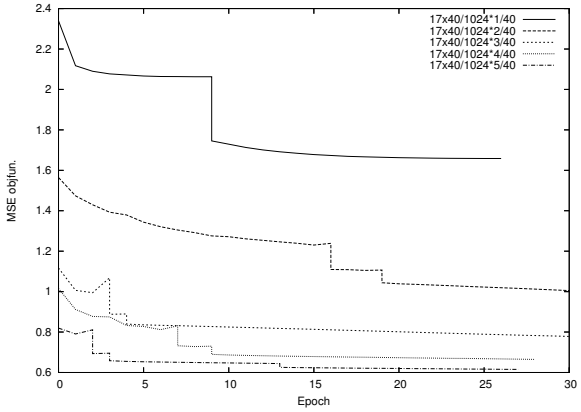

Figure 2: MSE objective functions for NN-FMMIs with increasing number of hidden layers.

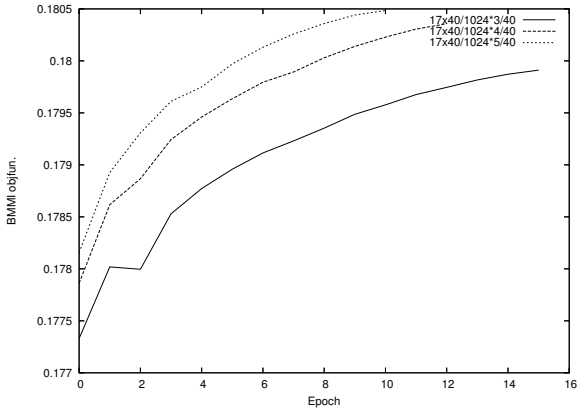

Figure 3: BMMI objective functions for NN-FMMIs with 3, 4 and 5 hidden layers.

In Figure 2 we show the MSE objective functions on held-out data (1/10th of the training data). The jumps correspond to the learning rate annealing steps. Correspondingly, we plot the BMMI objective function for the networks with 3, 4 and 5 hidden layers in Figure 3. Unlike the results from 3.2.2, here the BMMI criterion improves for networks with more parameters because of the pretraining step. Increased BMMI values also lead to improved word error rates as shown in Table 3. The best result of 17.9% is obtained after model-space training for a 6-layer NN-FMMI transform.

4. Conclusion

We have presented a discriminative feature transform formulation using deep neural networks which are trained through boosted MMI gradient backpropagation. Pretraining is done one layer at a time by learning the offsets of a linear FMMI transform. Experimental results on English broadcast news transcription show the superiority of NN-FMMI over regular FMMI after model-space discriminative training. Because of the pretraining step, NN-FMMI requires the estimation of a linear FMMI transform first. Future work will address ways

Architecture	MSE	BMMI	Model-space
17x40/1024*1/40	21.5%	–	–
17x40/1024*2/40	20.8%	–	–
17x40/1024*3/40	20.8%	20.1%	–
17x40/1024*4/40	20.8%	19.8%	17.9%
17x40/1024*5/40	20.7%	19.9%	–

Table 3: Comparison of word error rates for NN-FMMI transforms after MSE pretraining and BMMI refinement.

of removing this dependency by directly training the network with a regularized BMMI objective function and by employing second-order optimization methods [11] which have proven successful for neural network acoustic modeling [12].

5. Acknowledgments

This work was supported in part by DARPA under Grant HR0011-12-C-0015[3].

6. References

[1] D. Povey, B. Kingsbury, L. Mangu, G. Saon, H. Soltau, and G. Zweig, "fMPE: Discriminatively trained features for speech recognition," in *Proc. of ICASSP*, 2005, pp. 961–964.

[2] G.E. Dahl, D. Yu, L. Deng, and A. Acero, "Context-dependent pre-trained deep neural networks for large vocabulary speech recognition," *IEEE Transactions on Speech and Audio Processing*, vol. 20, no. 1, pp. 30–42, 2012.

[3] F. Seide, G. Li, X. Chen, and D. Yu, "Conversational speech transcription using context-dependent deep neural networks," in *Proc. Interspeech*, 2011.

[4] G. E. Hinton, S. Osindero, and Y. Teh, "A fast learning algorithm for deep belief nets," *Neural Computation*, vol. 18, pp. 1527–1554, 2006.

[5] H. Hermansky, D.P.W. Ellis, and S. Sharma, "Tandem connectionist feature extraction for conventional HMM systems," in *Proc. ICASSP*, 2000.

[6] František Grézl, Martin Karafiát, Stanislav Kontár, and Jan Černocký, "Probabilistic and bottle-neck features for LVCSR of meetings," in *Proc. ICASSP*, 2007.

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

[8] D. Povey, D. Kanevsky, B. Kingsbury, B. Ramabhadran, G. Saon, and K. Visweswariah, "Boosted MMI for model and feature-space discriminative training," in *Proc. of ICASSP*, 2008, pp. 4057–4060.

[9] G. Saon and D. Povey, "Penalty function maximization for large margin HMM training," in *Proc. Interspeech*, 2008, pp. 920–923.

[10] D. Povey, "Improvements to fMPE for discriminative training of features," in *Proc. Interspeech*, 2005, pp. 2977–2980.

[11] J. Martens, "Deep learning via Hessian-free optimization," in *Proc. ICML*, 2010.

[12] B. Kingsbury, T. N. Sainath, and H. Soltau, "Scalable minimum Bayes risk training of neural network acoustic models using distributed Hessian-free optimization," in *Proc. Interspeech*, 2012, Submitted.

[3] Approved for Public Release, Distribution Unlimited. The views, opinions, and/or findings contained in this article/presentation are those of the author/presenter and should not be interpreted as representing the official views or policies, either expressed or implied, of the Defense Advanced Research Projects Agency or the Department of Defense.