INCOHERENT TRAINING OF DEEP NEURAL NETWORKS TO DE-CORRELATE BOTTLENECK FEATURES FOR SPEECH RECOGNITION

Yebo Bao[1] *Hui Jiang*[2] *Lirong Dai*[1] *Cong Liu*[3]

[1]Department of Electronic Engineering and Information Science
University of Science and Technology of China, Hefei, Anhui, P. R. China
[2]Department of Computer Science and Engineering, York University, Toronto, Canada
[3]iFlytek Research, Anhui USTC iFlytek Co., Ltd., Hefei, Anhui, P. R. China
Email: bybillow@mail.ustc.edu.cn, hj@cse.yorku.ca, lrdai@ustc.edu.cn, congliu2@iflytek.com

ABSTRACT

Recently, the hybrid model combining deep neural network (DNN) with context-dependent HMMs has achieved some dramatic gains over the conventional GMM/HMM method in many speech recognition tasks. In this paper, we study how to compete with the state-of-the-art DNN/HMM method under the traditional GMM/HMM framework. Instead of using DNN as acoustic model, we use DNN as a front-end bottleneck (BN) feature extraction method to de-correlate long feature vectors concatenated from several consecutive speech frames. More importantly, we have proposed two novel incoherent training methods to explicitly de-correlate BN features in learning of DNN. The first method relies on minimizing coherence of weight matrices in DNN while the second one attempts to minimize correlation coefficients of BN features calculated in each mini-batch data in DNN training. Experimental results on a 70-hr Mandarin transcription task and the 309-hr Switchboard task have shown that the traditional GMM/HMMs using BN features can yield comparable performance as DNN/HMM. The proposed incoherent training can produce 2-3% additional gain over the baseline BN features. At last, the discriminatively trained GMM/HMMs using incoherently trained BN features have consistently surpassed the state-of-the-art DNN/HMMs in all evaluated tasks.

***Index Terms*—** Deep neural networks (DNN), nonlinear dimensionality reduction, bottleneck features, incoherent training, large vocabulary continuous speech recognition (LVCSR)

1. INTRODUCTION

Recently, the hybrid acoustic model using deep neural network (DNN) and hidden Markov models (HMMs) has been receiving more and more research attention in speech recognition because it has achieved some significant performance gain over the conventional acoustic models using continuous density HMMs based on Gaussian mixture models (GMM) in a variety of challenging large vocabulary continuous speech recognition (LVCSR) tasks [1, 2, 3]. Comparing with the early neural network HMMs (NN/HMMs) in 1990's, today's DNN/HMMs make use of a much larger neural network: (i) NN depth is largely increased by adding more hidden layers; (ii) NN output layer is significantly expanded to directly associate with a large number of tied states of context-dependent triphone HMMs rather than a small number of monophone states. Thanks to Hinton's unsupervised generative pre-training based on Restricted Boltzmann Machines (RBM) [4, 5, 6] and widely available general purpose GPUs, it is now possible to reliably learn such a huge NN efficiently from a large amount of training data. With these changes, it has been reported that DNN/HMM yields an unprecedented error reduction (over 20-30%) in the well-known Switchboard task [1]. Moreover, a recent experimental study in [7] has suggested that this impressive performance gain of DNN/HMM is almost entirely attributed to DNN's input vectors that are concatenated from several consecutive speech frames within a relatively long context window. Therefore, it becomes an interesting topic to study how to take advantage of these concatenated long feature vectors in the conventional GMM/HMM framework. However, unlike NNs, GMMs are unable to directly model these long concatenated feature vectors because these consecutive frames are highly correlated. We need to use some dimensionality reduction methods to de-correlate these long feature vectors before we can successfully apply the conventional GMM/HMMs to modeling them. In [8], a set of linear dimensionality reduction methods, such as PCA and LDA, have been explored to de-correlate these long concatenated feature vectors with no success, which suggests that other more advanced methods should be utilized for this purpose.

In [9], it has proposed a quite flexible nonlinear dimensionality reduction method based on neural networks by taking advantage of DNN's excellent capability of deeply mining high-dimensional data. The method has been used for speech recognition under the so-called bottleneck (BN) features [10, 11], where bottleneck features are initially extracted using some shallow NNs. Recently, the pre-trained DNNs are also used to extract bottleneck (BN) features for speech recognition [12, 13]. The idea of bottleneck (BN) features is to use a small hidden layer (the so-called bottleneck layer) in the middle of a DNN. After DNN is well-trained, the activation signals in the bottleneck layer can be used as a compact representation of the original high-dimensional inputs fed to the input layer of DNN. The reason to make the bottleneck layer small is to ensure that the activation signals in this layer are independent or uncorrelated. As a result, the low-dimensional activation signals can be used as new features to train the traditional GMM-based HMMs for acoustic modeling. It has been demonstrated that bottleneck (BN) features are effective in improving accuracy of speech recognition systems. One more advantage of using bottleneck features is that many successful techniques developed in the GMM-HMM framework can be easily applied to further improve recognition performance, such as discriminative training [14], fast adaptation and adaptive training.

In this paper, we first investigate how to use pre-trained DNNs to extract bottleneck (BN) features for large vocabulary continuous speech recognition (LVCSR). Contrary to some previous work in [12], we have found that discriminatively trained GMM/HMMs us-

978-1-4799-0356-6/13/$31.00 ©2013 IEEE

ing the bottleneck features extracted by DNNs have yielded comparable or even better recognition performance than the popular hybrid DNN/HMM models in several LVCSR tasks, including the well-known Switchboard task. Moreover, along this line of research, we have proposed two novel incoherent learning algorithms to train neural networks for extracting even better bottleneck features. The basic idea of our incoherent learning algorithm is to introduce a regularization term to the original objective function of DNN training. The regularization term aims to directly measure correlation among all activation signals in the bottleneck layer. In the first method, the regularization term is defined based on coherence of weight matrices in DNN. In the second method, we take advantage of the fact that the bottleneck layer is linear so that the regularization term can be directly formulated based on correlation coefficients computed in each mini-batch of training data. In this way, by optimizing a regularized objective function, we may directly control correlation in the bottleneck layer to derive better de-correlated BN features, which are even more suitable for GMM/HMMs with diagonal covariance matrices. Experimental results have shown that the proposed incoherent training methods have produced 2-3% further gain over the baseline BN features and the discriminatively trained GMM/HMM models using incoherently trained BN features have consistently surpassed the popular hybrid DNN-HMM methods in all evaluated LVCSR tasks.

2. BOTTLENECK FEATURES EXTRACTION

In this section, we briefly review the traditional way to use DNNs to extract bottleneck (BN) features as in [9, 10, 12]. As shown in Fig. 1, it uses a specially structured DNN, which includes a small bottleneck layer in the middle to control information flowing from the input layer of DNN to the output layer. Since the number of hidden nodes in the bottleneck layer is normally much smaller than the other layers, DNN training will force activation signals in the bottleneck layers to form as a low-dimensional compact representation of the original inputs as long as DNN is well trained to generate low frame classification error rate in the output layer. This can be viewed as an effective nonlinear dimensionality reduction to de-correlate the concatenated long input features to DNNs so that they can be directly used as new features to train GMM-HMMs as usual.

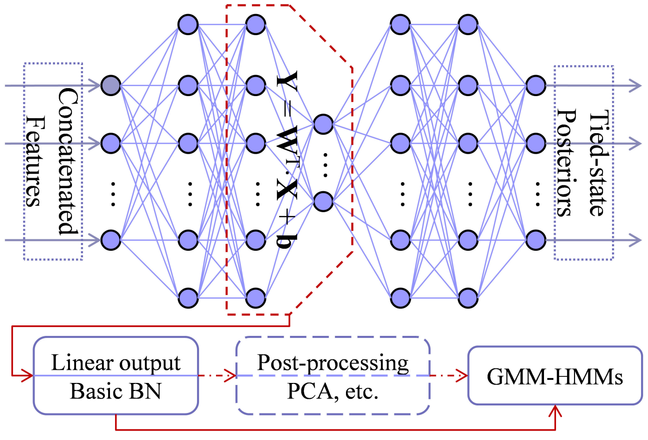
Concatenated Features
$\mathbf{Y} = \mathbf{W}^{\top}\mathbf{X} + \mathbf{b}$
Tied-state Posteriors
Linear output Basic BN
Post-processing PCA, etc.
GMM-HMMs

Fig. 1. Bottleneck features extraction architecture.

The training of the bottleneck DNN starts from RBM-based pre-training (see [9] for details). After pre-training, the bottleneck DNN is fine-tuned with the standard error back-propagation (BP) procedure to optimize an objective function. Normally, the objective function, $\mathcal{D}^{(0)}$, is defined as the total negative log posterior probability of all T training samples $\mathbf{O} = \{\mathbf{o}(t)\}$ given their ground-truth state labels $\mathbf{S} = \{s(t)\}$, i.e.

$$\mathcal{D}^{(0)} = -\sum_{t=1}^{T} \log P(s(t)|\mathbf{o}(t)) \,. \tag{1}$$

The derivatives of $\mathcal{D}^{(0)}$ can be easily derived as in [2, 3, 15].

3. INCOHERENT TRAINING OF DEEP NEURAL NETWORKS

In the above training process based on eq.(1), BN features are implicitly de-correlated by making the BN layer small. However, a small BN layer typically leads to lower frame classification rate in the output layer of the bottleneck DNN than regular DNNs. This indicates certain information has been lost in the narrow BN layer. On the other hand, as we increase the size of the bottleneck layer, the frame classification rate in the output layer typically improves accordingly but the activation signals in the BN layers may become more and more correlated with each other. As a result, they may not be good anymore for the following GMM/HMMs.

For the purpose of more effective bottleneck features extraction, in this paper, we propose two novel incoherent training methods for deep neural networks, where correlation between activation signals in the bottleneck layer is explicitly modeled and optimized when the bottleneck DNN is learned. The basic idea is to introduce a new regularization term into the original objective function in eq.(1) to derive a new regularized objective function for DNN training as follows:

$$\mathcal{D}^{(1)} = \mathcal{D}^{(0)} + \alpha \cdot \mathcal{D}^{(reg)} \,, \tag{2}$$

where $\mathcal{D}^{(0)}$ is the same as in eq.(1) and $\mathcal{D}^{(reg)}$ denotes a new regularization term to measure correlation in the bottleneck layer, and α is a control parameter to balance $\mathcal{D}^{(0)}$ and $\mathcal{D}^{(reg)}$ in learning. To some extent, the regularization term can be considered as a kind of constraint for DNN parameters estimation in the BP procedure so that the resultant BN features can be explicitly de-correlated during the BP training of DNNs. This becomes even more important when we use a slightly bigger bottleneck layer. In the following, we consider two different methods to construct the regularization term for our incoherent training by leveraging coherence of weight matrix and correlation coefficients computed with one mini-batch of training data.

3.1. Minimizing coherence of weight matrix in DNN

Coherence of a matrix $\mathbf{M}$ is defined as the maximum cosine value of angles between all normalized column vectors of $\mathbf{M}$. If we let $\mathbf{m}_i$ and $\mathbf{m}_j$ be any two column vectors of $\mathbf{M}$, coherence can be written as $\max_{i \neq j} \left| \frac{\mathbf{m}_i \cdot \mathbf{m}_j}{\|\mathbf{m}_i\| \cdot \|\mathbf{m}_j\|} \right|$. The concept of coherence has been widely used for dictionary learning in sparse representations of signals [16]. Intuitively, a matrix with smaller coherent value indicates all of its column vectors are less similar.

Since we can view the activation signals in the bottleneck layer as transformation from input vectors through several sigmoid layers with different weight matrices. If we can manage to reduce coherence of all these weight matrices, we may indirectly de-correlate signals in the BN layer. Therefore, we propose to construct the regularization term, $\mathcal{D}^{(reg)}$, as maximum coherence of these weight matrices. In practice, we use *softmax* to replace the above *max* to derive a differentiable $\mathcal{D}^{(reg)}$ only for the weight matrix right before the BN layer (assumed to contain N columns, $\mathbf{w}_i$, $1 \le i \le N$) as follows:

$$\mathcal{D}^{(reg)} \;=\; \log\left(\frac{1}{M}\sum_{i=1}^{N}\sum_{j=i+1}^{N}\exp\{\beta\cdot g_{ij}\}\right)^{\frac{1}{\beta}} \tag{3}$$

where $\beta > 1$ is a control parameter used for *softmax*, N is total number of hidden nodes in BN layer, $M = \frac{N(N-1)}{2}$, and g_{ij} denotes the absolute value of cosine of angle between any two column vectors in weight matrix:

$$g_{ij} \;=\; \frac{|\mathbf{w}_i\cdot\mathbf{w}_j|}{\|\mathbf{w}_i\|\cdot\|\mathbf{w}_j\|}\,. \tag{4}$$

Therefore, the gradients of $\mathcal{D}^{(reg)}$ with respect to each column of weight matrix, $\mathbf{w}_k$, $(k = 1,\cdots,N)$ can be calculated as:

$$\begin{aligned}\frac{\partial\mathcal{D}^{(reg)}}{\partial\mathbf{w}_k} \;&=\; \frac{\sum_{j\neq k}^{N}\left[\exp\{\beta\cdot g_{kj}\}\cdot\dfrac{\partial g_{kj}}{\partial\mathbf{w}_k}\right]}{\sum_{i=1}^{N}\sum_{j=i+1}^{N}\exp\{\beta\cdot g_{ij}\}} \\ &= \sum_{j=1}^{N}\gamma_{kj}g_{kj}\left[\frac{\mathbf{w}_j}{\mathbf{w}_k\cdot\mathbf{w}_j}-\frac{\mathbf{w}_k}{\mathbf{w}_k\cdot\mathbf{w}_k}\right] \;=\; \hat{\mathbf{W}}\cdot\hat{\mathbf{g}}_k\end{aligned} \tag{5}$$

where $\hat{\mathbf{W}}$ denotes a normalized weight matrix composed of column vectors, $\left[\frac{\mathbf{w}_j}{\mathbf{w}_k\cdot\mathbf{w}_j}-\frac{\mathbf{w}_k}{\mathbf{w}_k\cdot\mathbf{w}_k}\right]$ $(j = 1,\cdots,N)$, and $\hat{\mathbf{g}}_k$ is a scaled vector consisting of elements $\gamma_{kj}\cdot g_{kj}$ $(j = 1,\cdots,N)$ with $\gamma_{kj} = \frac{\exp\{\beta\cdot g_{kj}\}}{\sum_{i=1}^{N}\sum_{j=i+1}^{N}\exp\{\beta\cdot g_{ij}\}}$.

In BP learning, the α-scaled gradients as computed from eq.(2) are used to update weight matrices. Note the above $\mathcal{D}^{(reg)}$ can be easily generalized to include all weight matrices in the whole DNN not just the one right before the BN layer.

3.2. Minimizing correlation coefficients of data in each mini-batch

Here we consider to construct the regularization term, $\mathcal{D}^{(reg)}$, based on correlation coefficients of data. It is possible because the BN layer can be viewed as a linear layer since outputs of hidden nodes in the BN layer are directly used as BN features without using the sigmoid function. As shown in Fig. 1, the BN features, $\mathbf{Y}$, can be viewed as a linear transformation of output signals, $\mathbf{X}$, from the previous hidden layer as: $\mathbf{Y} = \mathbf{W}^\top\mathbf{X} + \mathbf{b}$, where $\mathbf{W}$ is the bottleneck layer weight matrix and $\mathbf{b}$ is a bias vector. Obviously, the relationship between the covariance matrix of $\mathbf{X}$ and that of $\mathbf{Y}$ can be expressed as: $\mathbf{C_Y} = \mathbf{W}^\top\mathbf{C_X}\mathbf{W}$. Therefore, we can define a new regularization term using correlation coefficients of $\mathbf{Y}$ calculated within each mini-batch of training data in the BP procedure. In this case, $\mathcal{D}^{(reg)}$ has the exactly same form as in eq.(3) except g_{ij} is computed as a correlation coefficient:

$$g_{ij} \;=\; \frac{\left|(\mathbf{w}_i)^\top\mathbf{C_X}\mathbf{w}_j\right|}{\sqrt{(\mathbf{w}_i)^\top\mathbf{C_X}\mathbf{w}_i}\cdot\sqrt{(\mathbf{w}_j)^\top\mathbf{C_X}\mathbf{w}_j}}\,. \tag{6}$$

Similarly, the gradients of $\mathcal{D}^{(reg)}$ with respect to each column of weight matrix can be computed as follows:

$$\frac{\partial\mathcal{D}^{(reg)}}{\partial\mathbf{w}_k} \;=\; \frac{\sum_{j\neq k}^{N}\left[\exp\{\beta\cdot g_{kj}\}\cdot\dfrac{\partial g_{kj}}{\partial\mathbf{w}_k}\right]}{\sum_{i=1}^{N}\sum_{j=i+1}^{N}\exp\{\beta\cdot g_{ij}\}}\,, \tag{7}$$

where $\dfrac{\partial g_{kj}}{\partial\mathbf{w}_k} = g_{kj}\left[\dfrac{\mathbf{C_X}\mathbf{w}_j}{(\mathbf{w}_k)^\top\mathbf{C_X}\mathbf{w}_j}-\dfrac{\mathbf{C_X}\mathbf{w}_k}{(\mathbf{w}_k)^\top\mathbf{C_X}\mathbf{w}_k}\right]$ for $k = 1,\cdots,N$. In the BP-based learning, $\mathbf{C_X}$ is first computed based on each mini-batch of training data and then the above derivatives are calculated accordingly to update the weight matrix before the BN layer. Note that this method can only be used for a linear layer where $\mathbf{C_Y}$ can be simply derived from $\mathbf{C_X}$.

4. EXPERIMENTS

In this work, we have evaluated the proposed incoherent training methods on two LVCSR tasks, namely an in-house 70-hour Mandarin transcription task and the 309-hour Switchboard task.

4.1. Mandarin transcription task

For the Mandarin transcription task, the training set contains 76,858 utterances (about 70 hours) from 1,539 speakers. The recognition performance is evaluated on a separate 3-hour test set, consisting of 3,720 utterances from other 50 speakers. In this task, we use a large trigram language model (LM) that is separately trained from a large Chinese text corpus. We evaluate recognition performance in the form of character error rate (CER) for this Chinese task.

4.1.1. Baseline systems

First of all, we build the baseline system based on the standard tied-state cross-word tri-phone GMM/HMMs using the regular 43-dimension features, that includes 39-dimension MFCC features (static, first and second derivatives) and 4-dimension pitch features. Prior to model training, feature vectors are pre-processed with cepstral mean normalization (CMN). The baseline models are first trained based on maximum likelihood estimation (MLE), including 3,978 tied states and 30 Gaussian components per state. After that, GMM-HMMs are discriminatively learned based on the minimum phone error (MPE) criterion. In the first row of Table 1, we give recognition performance of these two baseline GMM/HMMs using MFCCs. Next, based on the best DNN training configuration in [8], we have trained a 5-hidden-layer CD-DNN-HMMs with 2,048 nodes in each hidden layer to compute the posterior probability of all tied states used in the above baseline GMM-HMMs. In DNN-HMM, we use long concatenated feature vectors, stacking from all consecutive frames within a context window (5+1+5), as DNN's inputs. In BP, the mini-batch size is set to 1,024 samples for the stochastic gradient descend. The recognition performance is listed in the second row of Table 1 for comparison. We can see that CD-DNN-HMM yields a significant performance improvement, i.e., 21.6% relative error reductions, over the best discriminatively trained GMM-HMMs.

For the baseline BN features, we have trained a 5-hidden-layer bottleneck DNN. We place a bottleneck layer in the middle, which contains 43 hidden nodes for this task. After BP learning, the activation signals in the bottleneck layer are directly used as BN features to train another set of GMM/HMM without any post-processing. The recognition performance of BN-based GMM/HMM is listed

Table 1. Performance comparison (CER in %) of various models in the Mandarin task. (BN-GMM-HMM: GMM-HMMs using bottleneck features extracted by DNN)

	MLE	MPE
GMM-HMM	18.2	**16.7**
CD-DNN-HMM	**13.1**	
BN-GMM-HMM	13.6	**12.2**

Table 2. Performance comparison (CER in %) of GMM/HMMs using different bottleneck features in Mandarin task.

	MLE	MPE
Baseline BN	13.6	**12.2**
Weight-matrix Incoherent BN	**13.3**	-
Mini-batch-data Incoherent BN	**13.2**	**12.0**

in the last row of Table 1. The results show that performance of MLE-trained GMM/HMMs using BN features (BN-GMM-HMMs) is comparable to that of the state-of-the-art context-dependent DNN/HMM method. After MPE-based discriminative training, it even surpasses DNN/HMM (12.2% over 13.1% in CER).

4.1.2. Incoherently trained BN features

The performance comparison between the baseline BN features and incoherently trained BN features based on matrix coherence (as in section 3.1) is given in Table 2. The results show that the bottleneck features can benefit from weight matrix based incoherent training. For MLE-based GMM-HMM, further performance improvement is observed, i.e. 2.2% relative error reductions over baseline BN features. Similarly, we have evaluated another incoherent training based on mini-batch data (as in section 3.2). Results in Table 2 show that the MLE-trained GMM-HMMs using incoherent training based on mini-batch data give 3.0% relative error reductions over baseline BN features, which is slightly better than the first one using weight matrix based incoherent training. At last, we have discriminatively trained GMM-HMMs based on MPE using incoherently-trained BN features based on mini-batch data. The recognition performance in Table 2 indicates that it has yielded 12.0% in CER, about 1.7% relative error reduction over the same MPE-trained GMM-HMM using the baseline BN features.

4.2. Switchboard task

For the Switchboard task, the training data consists of 300-hour Switchboard-I training set and 20-hour Call Home English data. We use NIST 1998 and 2001 Hub5 evaluation sets, denoted as Hub5e98 and Hub5e01 respectively, to evaluate recognition performance.

4.2.1. Baseline systems

The baseline system is a standard tied-state cross-word triphone GMM/HMMs trained with both MLE and MPE criterion using 39-dimension PLP features (static, first and second derivatives) that are pre-processed with cepstral mean and variance normalization (CMVN) per conversation side. GMM/HMM consists of 8,991 tied states and 40 Gaussians per state. In decoding, we use a tri-gram LM trained with all training transcripts. In the first row of Table 3, we give recognition performance of two baseline GMM/HMMs, which is comparable with the best single-pass performance reported under the same training condition in [17].

Table 3. Performance Comparison (WER in %) of various baseline models in Switchboard task.

Acoustic Models	**Hub5e98**		**Hub5e01**	
	MLE	MPE	MLE	MPE
GMM-HMMs	46.6	**43.4**	35.4	**32.8**
CD-DNN-HMMs	**31.2**		**23.7**	
BN-GMM-HMMs	34.3	**31.3**	26.0	**23.2**

Next, we have trained a 5-hidden-layer CD-DNN-HMMs with 2,048 nodes in each hidden layer. The recognition performance is listed in the second row of Table 3. We can see context-dependent DNN/HMM yields 28.1% and 27.7% relative error reductions over the discriminatively trained GMM/HMMs on Hub5e98 and Hub5e01 test sets respectively. These results are similar to [1, 3]. Furthermore, recognition performance of GMM/HMM using 39-dimension baseline BN features is also listed in the last row of Table 3, which show that MPE-trained GMM/HMM using BN features yield comparable performance as the state-of-the-art CD-DNN-HMMs. These results are not the same as [12]. We have found that adjusting acoustic score factor in Viterbi decoding helps to boost performance for all BN-based systems.

4.2.2. Incoherently trained BN features for SWD

In Table 4, we have listed performance comparison of GMM/HMMs using different bottleneck features, including the baseline BN and two incoherently trained BN. Results have shown that incoherent training can consistently lead to about 2% further performance improvement over the baseline BN on two different evaluation sets.

Table 4. Performance (WER in %) comparison of GMM-HMMs using different BN features in Switchboard task.

BN Feature Systems	**Hub5e98**		**Hub5e01**	
	MLE	MPE	MLE	MPE
Baseline BN	34.3	31.3	26.0	23.2
Weight-matrix Incoherent BN	**34.0**	-	**25.7**	-
Mini-batch-data Incoherent BN	**33.8**	**31.0**	**25.6**	**22.8**

5. CONCLUSIONS

In this paper, we have proposed two new incoherent training methods for deep neural networks to de-correlate bottleneck features. Experimental results have shown that the incoherent training methods yield consistent performance gain over the traditional bottleneck feature extraction methods. More importantly, we have demonstrated that discriminatively trained GMM/HMMs using the incoherently trained bottleneck features have consistently outperformed the state-of-the-art DNN/HMM method in all evaluated speech recognition tasks, including the challenging Switchboard task.

Acknowledgement

This work was partially funded by the National Nature Science Foundation of China (Grant No. 61273264) and the National 973 program of China (Grant No. 2012CB326405).

6. REFERENCES

[1] F. Seide, G. Li, and D. Yu, "Conversational speech transcription using context-dependent deep neural networks," in *Proc. Interspeech*, 2011, pp. 437–440.

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

[3] F. Seide, G. Li, X. Chen, and D. Yu, "Feature engineering in context-dependent deep neural networks for conversational speech transcription," in *Proc. ASRU*, 2011, pp. 24–29.

[4] A. Mohamed, G.E. Dahl, and G.E. Hinton, "Deep belief networks for phone recognition," in *Proc. NIPS Workshop on Deep Learning for Speech Recognition and Related Applications*, 2009.

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

[6] A. Mohamed, G.E. Dahl, and G.E. Hinton, "Acoustic modeling using deep belief networks," *IEEE Trans. on Audio, Speech and Language Processing*, vol. 20, no. 1, pp. 14–22, January 2012.

[7] J. Pan, C. Liu, Z.G. Wang, Y. Hu, and H. Jiang, "Investigation of deep neural networks (DNN) for large vocabulary continuous speech recognition: Why DNN surpasses GMMs in acoustic modeling," in *Proc. ISCSLP*, 2012, (in press).

[8] Y.B. Bao, H. Jiang, C. Liu, Y. Hu, and L.R. Dai, "Investigation on dimensionality reduction of concatenated features with deep neural network for LVCSR systems," in *Proc. ICSP*, 2012, vol. 1, pp. 562–566.

[9] G.E. Hinton and R.R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.

[10] F. Grezl, M. Karafiat, S. Kontar, and J. Cernocky, "Probabilistic and bottle-neck features for LVCSR of meetings," in *Proc. ICASSP*, 2007, pp. 757–760.

[11] H.B. Hu and S.A. Zahorian, "A neural network based nonlinear feature transformation for speech recognition," in *Proc. Interspeech*, 2008, pp. 1533–1536.

[12] D. Yu and M.L. Seltzer, "Improved bottleneck features using pretrained deep neural networks," in *Proc. Interspeech*, 2011, pp. 237–240.

[13] T.N. Sainath, B. Kingsbury, and B. Ramabhadran, "Auto-encoder bottleneck features using deep belief networks," in *Proc. ICASSP*, 2012, pp. 4153–4156.

[14] H. Jiang, "Discriminative training for automatic speech recognition: A survey," *Computer and Speech, Language*, vol. 24, no. 4, pp. 589–608, October 2010.

[15] C.M. Bishop, *Neural Networks for Pattern Recognition*, Oxford University Press, 1995.

[16] C.D. Sigg, T. Dikk, and J.M. Buhmann, "Learning dictionaries with bounded self-coherence," *IEEE Signal Processing Letters*, vol. 19, no. 12, pp. 861–864, December 2012.

[17] P.C. Woodland, T. Hain, G. Evermann, and D. Povey, "CU-HTK March 2001 Hub5 system," in *Proc. DARPA Hub5E Conversational Speech Recognition Workshop*, 2001.