Beyond Cross-entropy: Towards Better Frame-level Objective Functions For Deep Neural Network Training In Automatic Speech Recognition

Zhen Huang[1], Jinyu Li[2], Chao Weng[1], Chin-Hui Lee[1]

[1] School of ECE, Georgia Institute of Technology, Atlanta, GA. USA
[2] Microsoft Corporation, One Microsoft Way, Redmond, WA. USA

Abstract

We propose two approaches for improving the objective function for the deep neural network (DNN) frame-level training in large vocabulary continuous speech recognition (LVCSR). The DNNs used in LVCSR are often constructed with an output layer with softmax activation and the cross-entropy objective function is always employed in the frame-leveling training of DNNs. The pairing of softmax activation and cross-entropy objective function contributes much in the success of DNN. The first approach developed in this paper improves the cross-entropy objective function by boosting the importance of the frames for which the DNN model has low target predictions (low target posterior probabilities) and the second one considers jointly minimizing the cross-entropy and maximizing the log posterior ratio between the target senone (tied-triphone states) and the most competing one. Experiments on Switchboard task demonstrate that the two proposed methods can provide 3.1% and 1.5% relative word error rate (WER) reduction , respectively, against the already very strong conventional cross-entropy trained DNN system.

Index Terms: deep neural network, cross-entropy, boosting difficult samples, log posterior ratio

1. Introduction

The recent success of context dependent deep neural network hidden Markov models (CD-DNN-HMMs) in automatic speech recognition (ASR) [1] draws a lot of attention. It has been shown that the CD-DNN-HMMs outperform traditional Gaussian mixture model (GMM) HMMs with more than 10% relative error reduction in various tasks and data sets [2, 3, 4, 5, 6, 7, 8]. Comparing to traditional artificial neural networks (ANNs) used in ASR [9, 10], recent DNNs have much wider and deeper hidden layers, often are constructed with a large softmax output layer to directly model senones [11] and always employ cross-entropy as the objective function in frame-leveling training. These differences make the foundation of DNN's success.

Various aspects of DNN are explored by researchers. In [6, 12] the second-order optimization method called "Hessian-free" for DNN training is investigated; in [13, 14, 15] different non-linear activation function in hidden layers are employed to for better DNN formulation; sequence-level discriminative training of DNN [7] also shows nice ability in improving the DNN's discriminative power. One thing worth noticing is that the frame-level training is an essential part of recent DNN systems, and even sequence-level discriminative training needs a good frame-level trained DNN to start with, however, there is rare research done for improving this important part.

In this paper, to improve the frame-level objective function of DNN training, we proposed two frameworks. Inspired by the intuitive idea that people always learn more from difficult problems in which they are prone to make mistakes, the first approach emphasizes the difficult frames with low target posterior probabilities and deemphasizes the importance of those frames already well predicted by the DNN. The second method jointly minimizes the cross-entropy and maximizes the log posterior ratio between the target senone (tied-triphone states) [11, 4] and the most competing one. Maximizing the probability ratio between the target and the most competing senone enlarges the margin between the right and the most competing prediction, thus helps to increase DNN model's generalization power and prediction robustness. The experiments on Switchboard task demonstrate that the proposed two methods can provide 3.1% and 1.5% relative word error reduction (WER), respectively, against the already very strong conventional cross-entropy based DNN system. Those WER reductions seem not that significant, but as reported in [16], Switchboard task is a very challenging one, even with speaker adaptation, only 2.7% relative WER reduction is obtained. Therefore, the proposed frameworks do improve the conventional cross-entropy based frame-level training of DNN in a non-neglectable manner.

The rest of the paper is organized as follows. In Section 2, we review the frame-level DNN training with the cross-entropy objective function. In Section 3, the proposed objective functions for frame-level DNN training are described. The experimental results are reported in Section 4. The conclusion and future plan are given in Section 5.

2. Frame-level Cross-entropy Objective Function for DNN training

In this work, the input of the DNN was a splice of a central frame (whose label is that for the splice) and its n context frames on both left and right sides, *e.g.,* $n = 10$. The hidden layers were constructed by sigmoid units and output layer is a softmax layer. The basic structure of a deep neural network is shown in Fig. 1. Specifically, the values of the nodes can be expressed as,

$$\mathbf{x}^i = \begin{cases} W_1\mathbf{o}^t + \mathbf{b}_1, & i = 1 \\ W_i\mathbf{y}^i + \mathbf{b}_i, & i > 1 \end{cases}, \quad (1)$$

$$\mathbf{y}^i = \begin{cases} \mathrm{sigmoid}(\mathbf{x}^i), & i < n \\ \mathrm{softmax}(\mathbf{x}^i), & i = n \end{cases}, \quad (2)$$

where W_1, W_i are the weight matrices and $\mathbf{b_1}, \mathbf{b_i}$ are the bias vectors; n is the total number of the hidden layers and both the $\mathrm{sigmoid}$, and $\mathrm{softmax}$ functions are element-wise operations. The vector $\mathbf{x}^i$ corresponds to pre-nonlinearity activations, and $\mathbf{y}^i$ and $\mathbf{y}^n$ are the neuron vectors at the i^{th} hidden layer and the

output layer, respectively. The softmax outputs were considered as an estimate of the senone posterior probability:,

$$p(C_j|\mathbf{o}_t) = \mathbf{y}_t^n(j) = \frac{\exp(\mathbf{x}_t^n(j))}{\sum_i \exp(\mathbf{x}_t^n(i))}, \tag{3}$$

where C_j represents the j^{th} senone and $\mathbf{y}^n(j)$ is the j^{th} element of $\mathbf{y}^n$ in Fig. 1.

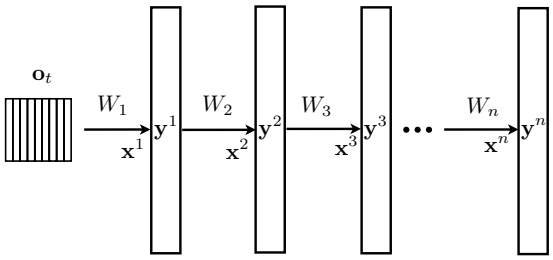

Figure 1: Structure of a deep neural network: W_i is weight matrix at the i^{th} layer, note that the bias terms are omitted for simplicity.

DNN is trained by maximizing the log posterior probability over the training frames. This is equivalent to minimizing the cross-entropy objective function. Let $\mathcal{X}$ be the whole training set which contains N frames, *i.e.* $\mathbf{o}_{1:N} \in \mathcal{X}$, then the loss with respect to. $\mathcal{X}$ is given by,

$$\mathcal{L}_{1:N} = -\sum_{t=1}^{N}\sum_{j=1}^{J} \mathbf{d}_t(j) \log p(C_j|\mathbf{o}_t), \tag{4}$$

where $p(C_j|\mathbf{o}_t)$ is defined in Eq. (3); $\mathbf{d}_t$ is the label vector of frame t. In real practices of DNN systems, the label vector $\mathbf{d}_t$ is often obtained by a forced alignment by an existing system resulting in only the target entry that is equal to 1. Thus, a simplified loss function is obtained:

$$\begin{aligned} \mathcal{L}_{1:N} &= -\sum_{t=1}^{N} \log p(C_l|\mathbf{o}_t) \\ &= -\sum_{t=1}^{N} \log \mathbf{y}_t^n(l), \end{aligned} \tag{5}$$

Where C_l is target senone indicated by the forced alignment.

The objective function is minimized by using error back propagation [17] which is a gradient-descent based optimization method developed for neural networks. Specifically, taking partial derivatives of the objective function with respect to the pre-nonlinearity activations of output layer $\mathbf{x}^n$, the error vector to be backpropagated to the previous hidden layers is generated:

$$\boldsymbol{\epsilon}_t^n = \frac{\partial \mathcal{L}_{1:N}}{\partial \mathbf{x}^n} = \mathbf{y}_t^n - \mathbf{d}_t, \tag{6}$$

the backpropagated error vector at previous hidden layer is,

$$\boldsymbol{\epsilon}_t^i = W_{i+1}^T \boldsymbol{\epsilon}_t^{i+1} * \mathbf{y}^i * \left(\mathbf{1} - \mathbf{y}^i\right), i < n \tag{7}$$

where $*$ denotes element-wise multiplication. With the error vectors at certain hidden layers, the gradient over the whole training set with respect to the weight matrix W_i is given by

$$\frac{\partial \mathcal{L}_{1:N}}{\partial W_i} = \mathbf{y}_{1:N}^{i-1} (\boldsymbol{\epsilon}_{1:N}^i)^T, \tag{8}$$

Note that in above equation, both $\mathbf{y}_{1:N}^{i-1}$ and $\boldsymbol{\epsilon}_{1:N}^i$ are matrices, which is formed by concatenating vectors corresponding to all the training frames from frame 1 to N, *i.e.* $\boldsymbol{\epsilon}_{1:N}^i = [\boldsymbol{\epsilon}_1^i, \ldots, \boldsymbol{\epsilon}_t^i, \ldots, \boldsymbol{\epsilon}_N^i]$. The batch gradient descent updates the parameters with the gradient in Eq. (8) only once after each sweep through the whole training set and in this way parallelization can be easily conducted. However, stochastic gradient descent (SGD) [18] usually works better in practice where the true gradient is approximated by the gradient at a single frame t, *i.e.* $\mathbf{y}_t^{i-1}(\boldsymbol{\epsilon}_t^i)^T$, and the parameters are updated right after seeing each frame. The compromise between the two, the mini-batch SGD [19], is more widely used, as the reasonable size of mini-batches makes all the matrices fit into GPU memory, which leads to a more computationally efficient learning process. In this work, we use mini-batch SGD to update the parameters.

Training a neural network directly from the randomly initialized parameters usually results in a poor local optimum when performing error back propagation, especially when the neural network is deep [20]. To cope with this, pre-training methods have been proposed for a better initialization of the parameters [21]. Pre-training grows the neural network layer by layer without using the label information. Treating each pair of layers in the network as a restricted Boltzmann machine (RBM), layers of the neural network can then be trained using an objective criterion called contrastive divergence [21].

3. Improving Cross-entropy Objective Function

3.1. Boosted cross-entropy

Inspired by the intuitive idea that people always learn more from difficult problems in which they are prone to make mistakes, we want our DNN to learn more from the "difficult frames" whose label the DNN model has difficulty in predicting correctly (that is indicated by a low target posterior probabilities $p(C_l|\mathbf{o}_t)$). To achieve this, we formulate the new objective function by adding a weighting term to the cross-entropy objective function to boost the difficult frames as in Eq. (9).

$$\begin{aligned} \mathcal{L}_{1:N}^{boosted} &= -\sum_{t=1}^{N} (1 - p(C_l|\mathbf{o}_t)) \log p(C_l|\mathbf{o}_t) \\ &= -\sum_{t=1}^{N} (1 - \mathbf{y}_t^n(l)) \log \mathbf{y}_t^n(l), \end{aligned} \tag{9}$$

By adding the weighting term $(1 - \mathbf{y}_t^n(l))$, the frames with smaller $\mathbf{y}_t^n(l)$ (which can be interpreted as prediction correctness of the target senone) are emphasized.

Taking partial derivatives of the objective function with respect to the pre-nonlinearity activations of output layer $\mathbf{x}^n$, the error vector to be backpropagated to the previous hidden layers is,

$$\begin{aligned} \boldsymbol{\epsilon}_t^n = \frac{\partial \mathcal{L}_{1:N}^{boosted}}{\partial \mathbf{x}^n} &= (1 - \mathbf{y}_t^n(l) - \mathbf{y}_t^n(l) \log \mathbf{y}_t^n(l))(\mathbf{y}_t^n - \mathbf{d}_t) \\ &= f_t(\mathbf{y}_t^n - \mathbf{d}_t), \end{aligned} \tag{10}$$

where $f_t = 1 - \mathbf{y}_t^n(l) - \mathbf{y}_t^n(l) \log \mathbf{y}_t^n(l)$.

Comparing Eq. (10) with Eq. (7), there is a scaling factor f_t in the new formulation. From the viewpoint of backpropagated error vectors, the importance of each frame is scaled by the factor f_t referred as importance factor in this paper.

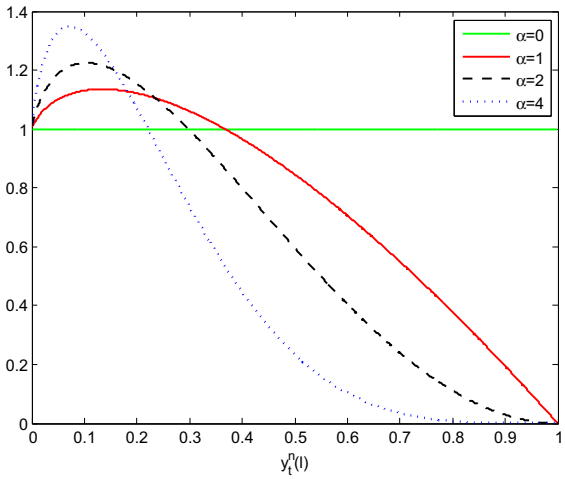

Figure 2: Shape of the importance factor f_t with respect to $\mathbf{y}_t^n(l)$ $(0 \leq \mathbf{y}_t^n(l) \leq 1)$ with different boosting order α.

To extend this framework, we want to control the degree of emphasis that is laid on the difficult frames. This is done by letting the order of the weighting term $(1 - \mathbf{y}_t^n(l))$ to be a variable parameter α called boosting order in this paper,

$$\mathcal{L}_{1:N}^{boosted} = -\sum_{t=1}^{N}(1 - \mathbf{y}_t^n(l))^{\alpha} \log \mathbf{y}_t^n(l), \tag{11}$$

Correspondingly, the error vector to be backpropagated to the previous hidden layers becomes,

$$\begin{aligned}\boldsymbol{\epsilon}_t^n &= \frac{\partial \mathcal{L}_{1:N}^{boosted}}{\partial \mathbf{x}^n} \\ &= (1 - \mathbf{y}_t^n(l))^{\alpha-1}(1 - \mathbf{y}_t^n(l) - \alpha \mathbf{y}_t^n(l) \log \mathbf{y}_t^n(l))(\mathbf{y}_t^n - \mathbf{d}_t)\end{aligned} \tag{12}$$

The importance factor f_t becomes,

$$f_t = (1 - \mathbf{y}_t^n(l))^{\alpha-1}(1 - \mathbf{y}_t^n(l) - \alpha \mathbf{y}_t^n(l) \log \mathbf{y}_t^n(l)) \tag{13}$$

Fig. 2 shows the shape of the importance factor f_t with respect to $\mathbf{y}_t^n(l)$ $(0 \leq \mathbf{y}_t^n(l) \leq 1)$ with different boosting order α. From the figure, we can find that when $\mathbf{y}_t^n(l)$ is approaching 1, f_t approaches 0, which means the importance of the corresponding frame falls when the prediction correctness grows up; when $\mathbf{y}_t^n(l)$ becomes smaller, which indicates DNN has difficulty in predicting the associated frame correctly, that particular frame's importance trends to increase. It can also be found that generally the higher α is, the more emphasis laid on the difficult frames. When $\alpha = 0$, the importance factor f_t becomes a constant as in Fig. 2, and the proposed objective function reduces to the conventional cross-entropy.

3.2. Cross-entropy with Log Posterior Ratio

When a statistical model is trained to discriminate data samples in different classes, it is always wanted that prediction probability of the target class are larger than the competing classes, especially the most competing class, in a significant way, thus, increase the prediction robustness of the model. This concept is widely used in GMM discriminative training, *e.g.,* minimum classification error (MCE) [22] and soft margin estimation (SME) [23, 24]. In the case of DNN acoustic models for LVCSR, we want the log posterior ratio between the target and the most competing senone to be large. In order to achieve this, we design the objective function to jointly minimize the cross-entropy and maximize the log posterior ratio between the target senone and the most competing one as in Eq. (14),

$$\begin{aligned}\mathcal{L}_{1:N}^{ratio} &= -\sum_{t=1}^{N}(\lambda(\log p(C_l|\mathbf{o}_t) - \log p(C_m|\mathbf{o}_t)) + \log p(C_l|\mathbf{o}_t)) \\ &= -\sum_{t=1}^{N}(\lambda(\log \mathbf{y}_t^n(l) - \log \mathbf{y}_t^n(m)) + \log \mathbf{y}_t^n(l)),\end{aligned} \tag{14}$$

Where, C_l is the target senone and C_m is the most competing senone defined as the senone with the largest posterior $p(C_m|\mathbf{o}_t)$ besides the target senone. $(\log p(C_m|\mathbf{o}_t) - \log p(C_l|\mathbf{o}_t))$ is the log posterior ratio term and $\log \mathbf{y}_t^n(l)$ is the negative cross-entropy term. $\lambda \geq 0$ here is a factor controlling the balance between the log posterior ratio and cross-entropy.

Taking partial derivatives of the objective function with respect to the pre-nonlinearity activations of output layer $\mathbf{x}^n$, the error vector to be backpropagated to the previous hidden layers is,

$$\boldsymbol{\epsilon}_t^n = \frac{\partial \mathcal{L}_{1:N}^{ratio}}{\partial \mathbf{x}^n} = \mathbf{y}_t^n - \mathbf{r}_t, \tag{15}$$

Where $\mathbf{r}_t$ is the revised label vector of frame t. All the entries of $\mathbf{r}_t$ are zero except the entry for the target senone $\mathbf{r}_t(l) = 1 + \lambda$ and the entry for the most competing senone $\mathbf{r}_t(m) = -\lambda$.

Comparing Eq. (15) with Eq. (7), the difference occurs between $\mathbf{d}_t$ and $\mathbf{r}_t$. From the viewpoint of the backpropagated error vector $\boldsymbol{\epsilon}_t^n$, the extra error $-\lambda$ in the target entry and the extra error λ in the most competing entry, enlarge the margin between the right and the most competing prediction, thus helps to increase DNN model's generalization power and prediction robustness. When $\lambda = 0$, there is no extra error to be backpropagated and the proposed objective function reduces to the conventional cross-entropy.

4. Experimental Results and Analysis

In this paper, we conduct the experiments on the 109 hour subset of the Switchboard conversational telephone speech task in the Kaldi environment [25]. The first 4k utterances from the 300 hour Switchboard-1 Release 2 (LDC97S62) [26] data form our development set, and the training set is consisted of the next 100K utterances. The Switchboard part of Hub5 '00 (LDC2002S09) [27] data serves as our testing set. The trigram language model (LM) is trained on 3M words of the Switchboard conversational telephone speech transcripts.

The feature used to train the DNN is the Kaldi's standard 40 dimensional "$linear\ discriminant\ analysis(LDA) + single\ semitied\ covariance(STC) + fMLLR$" feature with zero-mean and unit-variance normalization which is same with [7]. The final input to the DNN is an 11 frame (5 frames on each side of the current frame) context window of the 40 dimensional features. The DNN is constructed with 5 hidden layers. Each hidden layer has 2048 neurons. The output softmax layer has 2979 output units (according to 2979 senones).

The following scheme is used for training the DNN: the DNN is initialized with stacked restricted Boltzmann machines

Table 1: Testing WER for boosted cross-entropy objective function with different boosting order α

α	0	1	2	4
WER	19.2%	18.7%	18.6%	18.8%

Table 2: Testing WER for the joint cross-entropy/log-posterior-ratio objective function with different controlling factor λ

λ	0	1e-03	2e-03	4e-03
WER	19.2%	18.9%	18.9%	19.1%

(RBMs) by using layer by layer generative pre-training [21]. An initial learning rate of 0.01 is used to train the Gaussian-Bernoulli RBM and a learning rate of 0.4 is applied to the Bernoulli-Bernoulli RBMs. Then the network is discriminatively trained with different frame-level objective functions using backpropagation. All the utterances and frames are randomized before being fed in the DNN. The mini-batch size is set to 256 and the initial learning rate is set to 0.008. After each training epoch, we validate the frame accuracy on the development set, if the improvements is less than 0.5%, we shrink the learning rate by the factor of 0.5. The training process is stopped after the frame accuracy improvement is less than 0.1%. General purpose graphics processing units (GPGPUs) are utilized to accelerate both the training and pre-training processes.

Table 1 shows the testing WER results for the proposed boosted cross-entropy objective function with different boosting order α. $\alpha = 0$ is the case that the plain cross-entropy objective function is used. With a boosting order $\alpha = 2$, a relative WER reduction of 3.1% is achieved against the plain cross-entropy trained DNN. According to Fig. 3 and Table 1, it can be found that the proposed boosted cross-entropy objective function gets better testing WER with lower training and development frame-accuracy when comparing with the plain cross-entropy. This phenomenon is reasonable because the plain cross-entropy which treats each frame equally is more directly related to overall frame accuracy, while the proposed boosted cross-entropy objective function might not do well in enhancing the overall training frame accuracy but does help make some difficult and crucial frames be predicted correctly, thus, reduces the testing WER.

Table 2 shows the WER results for the proposed joint cross-entropy/log-posterior-ratio objective function with different controlling factor λ. Here $\lambda = 0$ means the joint cross-entropy/log-posterior-ratio objective function reduces to plain cross-entropy objective function. With λ=1e-03 or 2e-03, a relative WER reduction of 1.5% is achieved against the plain cross-entropy. From Fig. 3, it can be observed that the proposed joint cross-entropy/log-posterior-ratio objective function achieve higher frame accuracy in the development set with a lower training frame accuracy against the plain cross-entropy. Usually, this phenomenon means better dealing with the overfitting problem and better robustness of a model. The enlarged log posterior ratio between the target and most competing senone by the proposed objective function brings stronger generalization power to DNN model and helps to improve the testing WER in the end.

5. Conclusion and Future Work

In this paper, we have proposed two frameworks to improve the objective function of DNN frame-level training in LVCSR. The

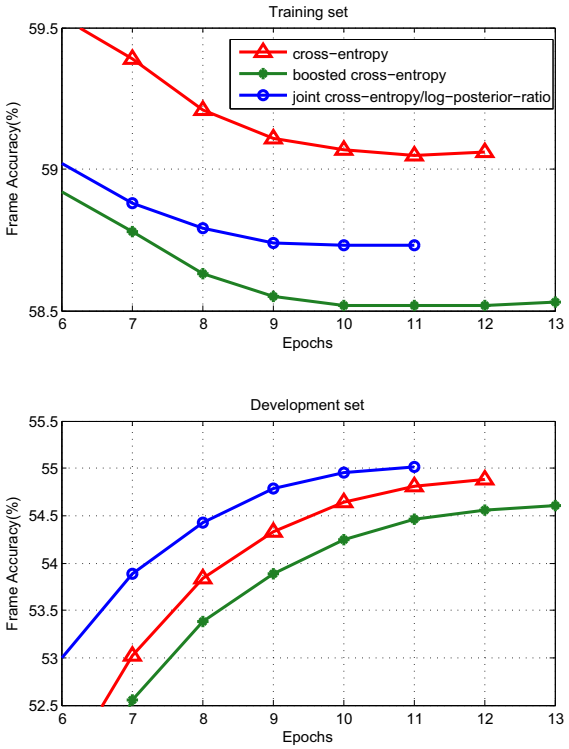

Figure 3: Frame Accuracy on Training and Development set with 3 different objective functions in the last 5-7 training epochs. For boosted cross-entropy, the boosting order $\alpha = 2$; for the joint cross-entropy/log-posterior-ratio, λ=1e-03. The specific value of α and λ is chosen to make sure the frame accuracy curve for each objective function is corresponding to the best case in term of testing WER.

basic idea behind the first approach, "boosted cross-entropy", is to "learn more from the difficult frames which the DNN model can't correctly predict"; the second "joint cross-entropy/log-posterior-ratio" approach enlarges the margin between right and most competing prediction so as to improve DNN model's generalization power and prediction robustness. Experiments on Switchboard task demonstrate that the proposed two frameworks can provide 3% and 1.5% relative word error reduction (WER), respectively, against the already very strong conventional cross-entropy trained DNN system.

The potential of the frame-level objective function is still far from being exploited. We believe deeper investigation on the proposed two frameworks will yield more interesting results, and the fusion of these two frameworks is worth trying. At the same time, designing more discriminative frame-level objective function through different ways other than the proposed ones can also be promising and we also want to incorporate the idea in this paper into sequence-level discriminative training.

6. Acknowledgment

The authors would like to thank our colleague You-Chi Cheng, Kehuang Li of Georgia Institute of Technology and Professor Ji Wu of Tsinghua University for valuable discussions and suggestions. We also want to thank Professor Bo Hong of Georgia Institute of Technology and his PhD student Jiadong Wu for helping us set up and utilize GPGPUs in DNN training.

7. References

[1] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, and B. Kingsbury, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 82–97, 2012.

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

[3] T. N. Sainath, B. Kingsbury, B. Ramabhadran, P. Fousek, P. Novak, and A. Mohamed, "Making deep belief networks effective for large vocabulary continuous speech recognition," in *Proc. ASRU*, 2011, pp. 30–35.

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

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

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

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

[8] Z. Yan, Q. Huo, and J. Xu, "A scalable approach to using DNN-derived features in GMM-HMM based acoustic modeling for LVCSR," in *Proc. Interspeech*, 2013.

[9] R. P. Lippmann, "Review of neural networks for speech recognition," *Neural computation*, vol. 1, no. 1, pp. 1–38, 1989.

[10] H. A. Bourlard and N. Morgan, *Connectionist speech recognition: a hybrid approach*. Springer, 1994, vol. 247.

[11] M.-Y. Hwang and X. Huang, "Shared-distribution hidden Markov models for speech recognition," *IEEE Trans. Speech and Audio Processing*, vol. 1, no. 4, pp. 414–420, 1993.

[12] S. Wiesler, J. Li, and J. Xue, "Investigations on Hessian-free optimization for cross-entropy training of deep neural networks," in *Proc. Interspeech*, 2013.

[13] A. L. Maas, A. Y. Hannun, and A. Y. Ng, "Rectifier nonlinearities improve neural network acoustic models," in *Proc. ICML*, 2013.

[14] M. Cai, Y. Shi, and J. Liu, "Deep maxout neural networks for speech recognition," in *Proc. ASRU*, 2013, pp. 291–296.

[15] P. Swietojanski, J. Li, and J. T. Huang, "Investigation of maxout networks for speech recognition," in *Proc. ICASSP*, 2014.

[16] D. Yu, K. Yao, H. Su, G. Li, and F. Seide, "KL-divergence regularized deep neural network adaptation for improved large vocabulary speech recognition," in *Proc. ICASSP*, 2013, pp. 7893–7897.

[17] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, *Learning representations by back-propagating errors*. MIT Press, Cambridge, MA, USA, 1988.

[18] T. Zhang, "Solving large scale linear prediction problems using stochastic gradient descent algorithms," in *Proc. ICML*, 2004, pp. 919–926.

[19] O. Dekel, R. Gilad-Bachrach, O. Shamir, and L. Xiao, "Optimal distributed online prediction," in *Proc. ICML*, 2011, pp. 713–720.

[20] D. Erhan, P.-A. Manzagol, Y. Bengio, S. Bengio, and P. Vincent, "The difficulty of training deep architectures and the effect of unsupervised pre-training," in *Proc. AISTATS*, 2009, pp. 153–160.

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

[22] B. H. Juang, W. Chou, and C.-H. Lee, "Minimum classification error rate methods for speech recognition," *IEEE Trans. Speech and Audio Processing*, vol. 5, no. 3, pp. 257–265, 1997.

[23] J. Li, M. Yuan, and C.-H. Lee, "Soft margin estimation of hidden Markov model parameters." in *Proc. Interspeech*, 2006.

[24] ——, "Approximate test risk bound minimization through soft margin estimation," *IEEE Trans. Audio, Speech, and Language Processing*, vol. 15, no. 8, pp. 2393–2404, 2007.

[25] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, J. Silovskỳ, G. Stemmer, and K. Veselỳ, "The Kaldi speech recognition toolkit," in *Proc. ASRU*, 2011.

[26] J. J. Godfrey and E. Holliman, "Switchboard-1 release 2," *Linguistic Data Consortium, Philadelphia*, 1997.

[27] J. Fiscus, W. M. Fisher, A. F. Martin, M. A. Przybocki, and D. S. Pallett, "2000 NIST evaluation of conversational speech recognition over the telephone: English and mandarin performance results," in *Proc. Speech Transcription Workshop*, 2000.