DEEP MAXOUT NEURAL NETWORKS FOR SPEECH RECOGNITION

Meng Cai, Yongzhe Shi and Jia Liu

Tsinghua National Laboratory for Information Science and Technology
Department of Electronic Engineering, Tsinghua University, Beijing 100084, China
cai-m10@mails.tsinghua.edu.cn, shiyz09@gmail.com, liuj@tsinghua.edu.cn

ABSTRACT

A recently introduced type of neural network called maxout has worked well in many domains. In this paper, we propose to apply maxout for acoustic models in speech recognition. The maxout neuron picks the maximum value within a group of linear pieces as its activation. This nonlinearity is a generalization to the rectified nonlinearity and has the ability to approximate any form of activation functions. We apply maxout networks to the Switchboard phone-call transcription task and evaluate the performances under both a 24-hour low-resource condition and a 300-hour core condition. Experimental results demonstrate that maxout networks converge faster, generalize better and are easier to optimize than rectified linear networks and sigmoid networks. Furthermore, experiments show that maxout networks reduce underfitting and are able to achieve good results without dropout training. Under both conditions, maxout networks yield relative improvements of 1.1-5.1% over rectified linear networks and 2.6-14.5% over sigmoid networks on benchmark test sets.

Index Terms— Maxout networks, acoustic modeling, neuron nonlinearity, speech recognition

1. INTRODUCTION

The combination of deep neural networks and hidden Markov models, also referred to as DNN-HMM hybrid approach, is quickly becoming the dominant acoustic modeling technology for speech recognition in recent years [1, 2]. The key idea is to use a neural network with many hidden layers to model the HMM state posterior probabilities. This method has several advantages over traditional GMM-HMM approaches. Firstly, neural networks with many nonlinear hidden layers yield stronger modeling power than GMMs. Secondly, with minimal assumptions about the data distribution, DNNs are able to extract informative features with less front-end processing (e.g. filter-bank features) [3] and discover the relationships between neighboring frames [4]. Thirdly, the hierarchical structures of DNNs enable parameter sharing within the hidden layers, which is more powerful and efficient than having many disjoint parameters for every target.

Since the initial successful attempt of the DNN-HMMs in phoneme recognition [5] and large vocabulary phone-call transcription tasks [6], extensive explorations have been focused on deep learning methods for speech recognition, and state-of-the-art results have been updated many times. Successful models include pre-trained sigmoidal networks [1, 2], rectified linear networks [7, 8, 9], etc.

For pre-trained sigmoidal networks, the logistic sigmoidal nonlinearity is chosen. Although this form of nonlinearity is widely used in neural networks due to its smoothness and its simplicity of gradient computation, some drawbacks still exist. One obvious drawback is that the sigmoid function only has large gradient values when its input is near zero. If the input magnitudes of some neurons are large, the corresponding gradient values tend to be small and standard back-propagation (BP) training is less effective. This problem is especially severe when the network is deep, making the network training procedures sensitive to the hyper-parameter tuning. This issue is partially solved by an efficient pre-training algorithm proposed by Hinton et. al. to pre-train each hidden layer generatively as an RBM [10]. Later, Seide et. al. argue that when enough training data is available, pre-training is not necessary [11]. This discovery inspires people to find better ways to train DNNs without pre-training.

The recently proposed rectified linear units (ReLU) [12] for deep neural networks are showing some benefits over sigmoid units. This form of nonlinearity chooses $y = \max(x, 0)$ as the activation function, the resulting gradients for each neuron are either 0 (when $x < 0$) or 1 (when $x > 0$). This simple method prevents the gradients from vanishing, so that the ReLU networks are easier to train with BP and deeper models can be applied. The ReLU networks are proved to perform well for acoustic modeling. Impressive results reported in [8] show that ReLU networks with up to 12 hidden layers can be successfully trained using several hundred hours of speech data. And in [9], ReLU networks have shown superior performance in the well-known Switchboard phone-call transcription task over tanh networks. In both [8] and [9], the ReLU networks are trained only with BP, i.e. without pre-training.

The success of the ReLU networks motivates us to recon-

This work is supported by National Natural Science Foundation of China under Grant No. 61370034, No. 61273268 and No. 61005019.

sider the choice of the nonlinearities for deep neural networks. Inspired by the recent work in many machine learning tasks [13], we introduce *maxout* nonlinearity to deep neural network acoustic modeling. The maxout network is so named because each maxout unit chooses the maximum value within a group of linear pieces as the activation. So that the network is **linear** almost everywhere (except for the output softmax nonlinearity), which resembles the ReLU network. However, the maxout units compare values of a group of candidate pieces, while the ReLUs only compare the value of a single piece with 0. In this opinion, the maxout units are generalizations to ReLUs. In [13], the authors further demonstrate that maxout is a universal approximator and maxout networks perform better than previous methods in many machine learning benchmark tasks.

In this paper, we study deep maxout networks for acoustic modeling and apply them to the Switchboard phone-call transcription tasks. We find that maxout networks have good performance when applied to speech recognition systems. A series of experimental results show that maxout networks perform better than both pre-trained sigmoidal networks and ReLU networks. In [13], maxout is applied along with dropout [14], but we discover maxout networks can be successfully trained without dropout and reduce underfitting.

The remainder of this paper is organized as follows: In Section 2, we briefly review DNN-HMMs. In Section 3, we propose the maxout network acoustic models, including its training and testing strategies. We report our experiments in detail in Section 4 and conclude this paper in Section 5.

2. REVIEW OF DNN-HMMS

In the DNN-HMM hybrid approach, acoustic events are modeled by a deep neural network, whose inputs are concatenated acoustic features, and whose outputs are softmax posterior probabilities $p(s|\mathbf{o})$ with respect to each HMM state s. The DNN training process optimizes the cross entropy function:

$$D = -\sum_{s} d_s \log p(s|\mathbf{o}) \tag{1}$$

where d_s is 1 for the target state and 0 for non-target states. The state-level transcription is generated by a forced alignment using a baseline system. During test process, the acoustic score $\log p(\mathbf{o}|s)$ for the observation $\mathbf{o}$ is computed as

$$\log p(\mathbf{o}|s) = \log p(s|\mathbf{o}) + \log p(\mathbf{o}) - \log p(s) \tag{2}$$

where $p(s)$ is the state prior probability estimated from the training set. The $p(\mathbf{o})$ is the observation probability, which can be omitted as it's a constant for each input observation.

Though there exist many nonlinear functions for DNNs, the traditional sigmoid function is most often used. The recently proposed ReLU nonlinearity has shown some benefits over sigmoid nonlinearity and often yields better results [9]. Both these nonlinearities are illustrated in Fig. 1.

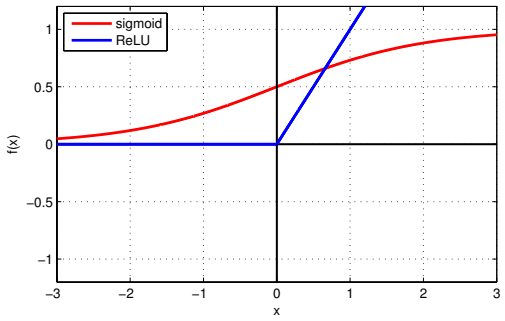


Fig. 1. Illustration of sigmoid and ReLU nonlinearity.

3. MAXOUT NETWORKS AND THEIR APPLICATIONS TO ACOUSTIC MODELING

In this section, we present the maxout networks and analyze their properties. We also consider some practical issues when the maxout networks are applied to acoustic modeling.

3.1. Model description

In maxout neural network, each neuron has a group consisting of k candidate pieces. The maximum value across the k pieces is chosen as the neuron activation. Denote the ith node of the lth hidden layer as h_l^i and its corresponding pieces as z_l^{ij}, the relationship between them satisfies:

$$h_l^i = \max_{j \in 1...k} z_l^{ij} \tag{3}$$

and z_l^{ij} is obtained by forward propagation from the layer below, i.e.:

$$\mathbf{z}_l = \mathbf{W}_{l-1}^T \mathbf{h}_{l-1} + \mathbf{b}_l \tag{4}$$

where $\mathbf{z}_l \in \mathbb{R}^Z$ stands for the vector of the lth layer to be max-pooled, whose elements are z_l^{ij}. The $\mathbf{h}_{l-1} \in \mathbb{R}^H$ is the maxout activation vector of the $l-1$th layer. The $\mathbf{W}_{l-1} \in \mathbb{R}^{H \times Z}$ denotes the weight matrix of the $l-1$th layer and $\mathbf{b}_l \in \mathbb{R}^Z$ denotes the bias vector of the lth layer. An example of the maxout network with 3 hidden layers and a group of 3 pieces for each neuron is illustrated in Fig. 2.

The forward-propagation process of the maxout network is the same as other feed-forward neural networks except that the activation computation follows equation (3) and (4). For the back-propagation process during training, the gradient for each maxout neuron is always 1, but only the weights corresponding to the piece with the maximum activation within each group $\{z_l^{ij} | j \in 1...k\}$ for $l \in [1, L]$ and $i \in [1, N^l]$ are updated. This is similar to max-pooling in convolutional networks [15]. However, in convolutional networks max-pooling happens across spacial locations while in maxout networks we do max-pooling across k pieces, which can be viewed as different features across the same spacial region.

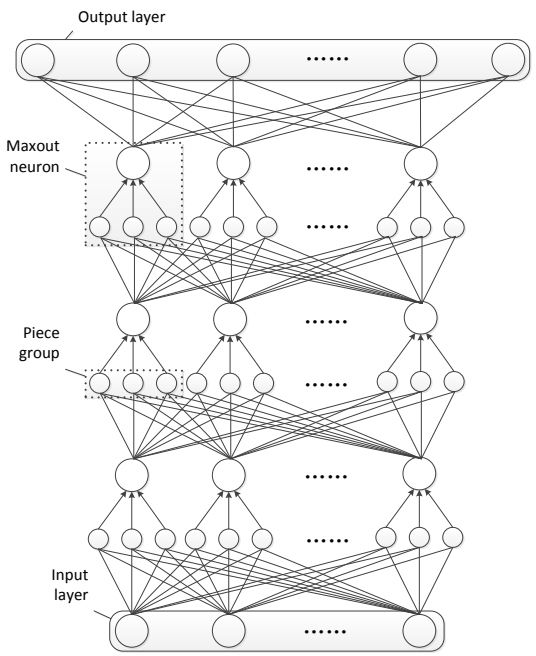


Fig. 2. Example of a fully-connected maxout network with 3 hidden layers and a group of 3 pieces for each neuron.

3.2. Model analysis

The maxout unit achieves its nonlinearity just by max-pooling across k pieces, which may seem surprisingly simple compared with traditional neural network nonlinear functions such as logistic sigmoid or hyperbolic tangent. It is interesting to analyze how and why it works. We come up with three reasons for the maxout network's competitiveness: The max-pooling operation enables robust feature selection; the maxout neuron is a generalization to the ReLU neuron and the maxout nonlinearity is a universal approximator.

The max-pooling operation is like a winner-take-all action, which is first applied to convolutional networks [15]. Given a region consisting of k candidate neurons, the most active neuron is selected as a representation of that region, while other candidates are eliminated. Each candidate neuron in maxout network can be viewed as a different feature map, representing a different aspect of information from the layers below. The maxout neuron, if well-trained, will hopefully select the most useful feature and make the classifier robust.

The maxout neuron is also a generalization of the ReLU neuron, their relationship is illustrated in Fig. 3. For the ReLU nonlinearity, max-pooling happens between a single feature map and 0, which behaves just like a maxout neuron with 2 pieces but one piece is always 0. By comparison, maxout units perform feature selections in a wise way, while ReLU just throws information away.

In [13], the authors have proved that just two maxout units

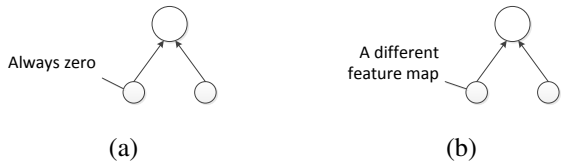


Fig. 3. A unified view of (a) ReLU neuron and (b) maxout neuron with 2 pieces. ReLU neuron performs max-pooling between a single feature map and 0, while maxout neuron performs max-pooling between different feature maps.

can approximate any continuous function arbitrarily closely if there are enough pieces for each unit. So during the maxout network training, the neurons learn the activation functions automatically themselves and can implement any form of nonlinear functions in theory. This makes maxout network powerful and flexible when applied to real-world problems.

3.3. Maxout networks for acoustic modeling

When applied to acoustic modeling, the maxout network estimates HMM state posterior probabilities in the same fashion as traditional sigmoidal network does. However, it is necessary to explore the effects of using different numbers of pieces and different numbers of hidden layers. It is also interesting to compare maxout networks with ReLU networks and pre-trained sigmoidal networks on benchmarks test sets.

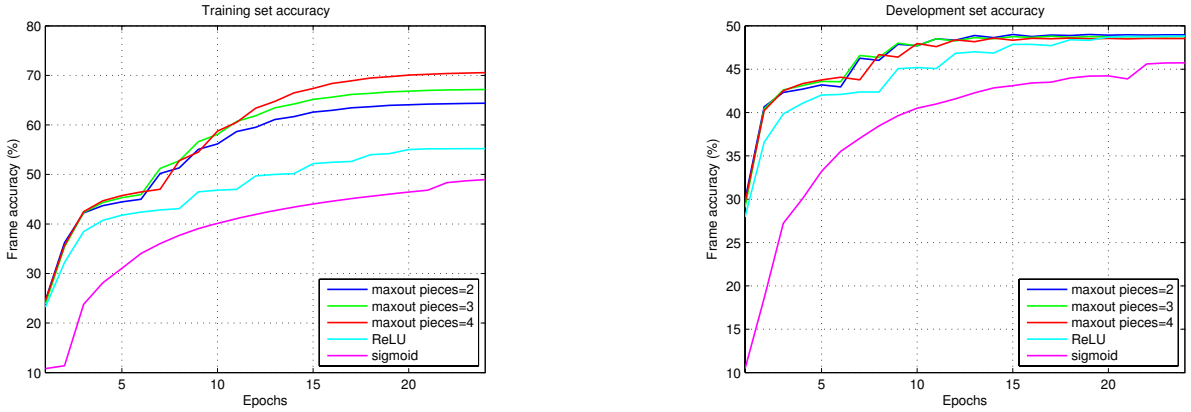
4. EXPERIMENTAL RESULTS

In this section, we present our experimental settings and results of maxout networks on the Switchboard phone-call transcription task. For all our experiments, we use the SWB part of Hub5'00 set as development set and the FSH part of RT03S set as test set. To our knowledge, this is the first time maxout networks are applied to speech recognition.

4.1. Experiment setup and baseline results

In the paper, we conduct our experiments on two training conditions: the core condition, under which we use all 300 hours of training data, and the low-resource condition, under which we select 24 hours of training data. We try different network settings under the low-resource condition and then report results under the core condition.

The baseline setup is based on our previous work [16], which is briefly described below. The raw 13-dimensional PLP features are concatenated with their first, second and third order derivatives and reduced to 39 dimensions using HLDA. For the core condition, the GMM-HMM contains 9308 states with 40 Gaussians each. The model is first trained using maximum likelihood and then refined using MPE criterion. The ML-trained model is used to generate state-aligned



(a) (b)

Fig. 4. Frame classification accuracy on (a) 24 hours of Switchboard training set and (b) Hub5'00 SWB development set as learning progresses. Results of maxout networks with different pieces, an ReLU network and a pre-trained sigmoidal network are compared. All the networks have 7 hidden layers with 480 units each.

Table 1. Baseline GMM-HMM results. Models are trained using 300 hours of Switchboard corpus. Performances are measured in word-error rate (WER) given in %.

Model	Hub5'00-SWB	RT03S-FSH
GMM-HMM ML	26.4	29.6
GMM-HMM MPE	23.4	26.8

transcriptions for DNN training. A trigram language model is trained using the transcription of the 2000h Fisher corpus and interpolated with a more general trigram. The baseline results are listed in Table 1.

4.2. Effects of piece groups

For the maxout network, the first thing we explored is the effects of different numbers of pieces for the piece groups under the low-resource condition. The input features to DNNs are 13-dimensional PLP features plus their first order and second order derivatives. The features are normalized to have zero mean and unit variance based on conversation-side information. A context window of 11 frames (5 frames on each side) is used.

To investigate the effects of different piece numbers, we fix maxout networks to have 7 hidden layers with 480 units each, but vary the piece numbers to 2, 3 and 4. A sigmoidal network and an ReLU network with the same number of hidden layers and units are also trained under the same condition for comparison. The learning procedures for the maxout and ReLU networks are the same. The initial learning rate is set to 0.01. At the end of every epoch, we evaluate frame accuracy of the development set and reduce the learning rate by

Table 2. Speech recognition results on 24 hours of Switchboard training data. Networks have 7 hidden layers with 480 units each. Performances are measured in WER given in %.

Model	Hub5'00-SWB	RT03S-FSH
GMM-HMM ML	35.3	39.2
sigmoid	26.2	29.2
ReLU	23.6	26.9
maxout pieces=2	**22.4**	**26.0**
maxout pieces=3	22.4	26.4
maxout pieces=4	22.4	26.3

a factor of 2 if the accuracy decreases. For the first epoch, we use a momentum of 0.5 and increase it to 0.9 afterwards. To prevent the weight vectors from growing too large, a norm constraint is also applied to limit the norm of the incoming weight vectors corresponding to each hidden unit to 0.8. For the sigmoidal network, DBN pre-training is first applied following the process in [11]. At the fine-tuning stage, the initial learning rate is set to 0.08. No norm constraint is used. Our implementations of the networks are based on an extended version of CUDAMat library [17].

For different networks, the frame accuracies are shown in Fig. 4. Results show that maxout networks converge faster than both sigmoidal and ReLU networks. On the training set, the maxout networks show better abilities to fit the data. While on the test set, the maxout networks and the ReLU network achieve almost the same accuracy, but the pre-trained sigmoidal network performs less well. As better classification accuracy doesn't always lead to lower speech recognition error rate [18], we also evaluate the speech recognition performance and the results are shown in Table 2. A GMM-HMM

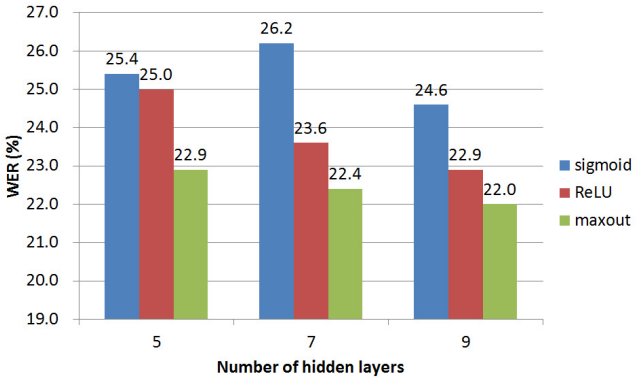


Fig. 5. Speech recognition results on Hub5'00-SWB corpus. Networks with 480 units for each hidden layer are trained using 24 hours of Switchboard data.

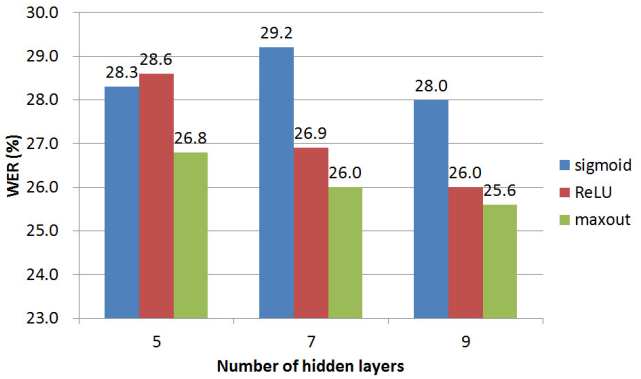


Fig. 6. Speech recognition results on RT03S-FSH corpus. Networks with 480 units for each hidden layer are trained using 24 hours of Switchboard data.

acoustic model with 1832 states and HLDA transformation is trained using the 24 hours of data for comparison. The speech recognition experiments show that maxout networks outperform both ReLU and sigmoidal networks. The maxout network with 2 pieces yields the best generalization ability and achieves relative error reductions of 14.5% and 5.1% on the development set and 10.9% and 3.3% on the test set compared with sigmoidal and ReLU networks. Therefore, maxout networks with **2 pieces** are used for all the following experiments.

4.3. Effects of network layers

DNNs often perform better with more hidden layers, but as networks grow deeper, their optimizations become harder. We evaluate the effects of different numbers of hidden layers to maxout networks, ReLU networks and sigmoidal networks using the 24 hours of Switchboard training data. For all the networks, we use the same learning procedure as described in the previous subsection. In these experiments, the number of units for the networks are fixed to 480 but the number of hidden layers are varied to 5, 7 and 9. Speech recognition results on Hub5'00-SWB development set and RT03S-FSH test set are shown in Fig. 5 and Fig. 6.

Experiments show that maxout networks bring consistent improvements over ReLU and sigmoidal networks for different numbers of hidden layers. Results also show that maxout and ReLU networks with up to 9 hidden layers can be successfully trained with back-propagation alone. In addition, lower error rates can be achieved for maxout and ReLU networks when more hidden layers are used. But for the sigmoidal networks, the performance improves less obviously with more hidden layers, sometimes even get worse. This suggests that maxout networks and ReLU networks are easier to optimize than sigmoidal networks.

Table 3. Speech recognition results using 300 hours of Switchboard training data. All networks have 7 hidden layers with 2048 units each. Performances are measured in WER given in %.

Model	Hub5'00-SWB	RT03S-FSH
sigmoid	15.7	19.0
ReLU	15.3	18.7
maxout	**15.1**	**18.5**

4.4. Core results

After exploring the effects of different network settings, we present our results under the core condition using all the 300 hours of Switchboard training data. We extract 40-dimensional filter-bank features plus energy, along with first and second order temporal derivatives. The features are then normalized to have zero mean and unit variance. We compare results of a maxout network, an ReLU network and a pre-trained sigmoidal network. All of them have 7 hidden layers and 2048 units per hidden layer. For the maxout network, we apply a norm constraint of 1.0 and a final momentum of 0.7. For the ReLU network, we apply a norm constraint of 1.0 and a final momentum of 0.9. Training stops when the learning rate is reduced 5 times. Other learning procedures are the same as that described in Subsection 4.2.

The core results are shown in Table 3. The maxout network offers a relative improvement of 2.6-3.8% over the sigmoidal network, and a relative improvement of 1.1-1.3% over the ReLU network. Though the improvements are less significant than those under the low-resource condition, the maxout network needs fewer epochs to converge and we believe larger performance gap may be obtained when the networks have even more hidden layers.

5. CONCLUSIONS

In this paper, we have introduced and analyzed maxout networks for acoustic modeling in speech recognition. The maxout unit selects the maximum value within a group of different feature maps, and is a generalization to the rectified linear unit. We observe that maxout networks converge faster and generalize better than ReLU networks and pre-trained sigmoidal networks, and maxout networks with 2 pieces perform best for speech recognition. Moreover, experiments also show that deep maxout networks are easy to optimize without the need for pre-training and work well without dropout. Finally, on the Switchboard phone-call transcription task, the maxout network achieves a 2.6-14.5% relative improvement over the pre-trained sigmoidal network and a 1.1-5.1% relative improvement over the ReLU network.

In the future, we will explore deeper maxout neural networks with more training data. We will also explore the combination of maxout neural networks and dropout for speech recognition under low-resource conditions and for minority languages.

6. ACKNOWLEDGEMENTS

We thank Ian Goodfellow from Université de Montréal for helpful discussions and suggestions about maxout networks.

7. REFERENCES

[1] G.E. 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, pp. 82–97, November 2012.

[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 Audio, Speech and Language Processing*, vol. 20, pp. 30–42, January 2012.

[3] A. Mohamed, G. Hinton, and G. Penn, "Understanding how deep belief networks perform acoustic modeling," in *Proc. ICASSP*. IEEE SPS, 2012, pp. 4273–4276.

[4] J. Pan, C. Liu, Z. 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*. IEEE, 2012, pp. 301–305.

[5] A. Mohamed, G.E. Dahl, and G.E. Hinton, "Acoustic modeling using deep belief networks," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 20, pp. 14–22, January 2012.

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

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

[8] M.D. Zeiler, M. Ranzato, R. Monga, M. Mao, K. Yang, Q.V. Le, P. Nguyen, A. Senior, V. Vanhoucke, J. Dean, and G.E. Hinton, "On rectified linear units for speech processing," in *Proc. ICASSP*. IEEE, 2013, pp. 3517–3521.

[9] A. Maas, A. Hannun, and A. Ng, "Rectifier nonlinearities improve neural network acoustic models," in *Proc. ICML*. International Machine Learning Society, 2013.

[10] G.E. Hinton, S. Osindero, and Y. Teh, "A fast learning algorithm for deep belief nets," *Neural Computation*, vol. 18, no. 7, pp. 1527–1554, 2006.

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

[12] V. Nair and G. Hinton, "Rectified linear units improve restricted Boltzmann machines," in *Proc. ICML*. International Machine Learning Society, 2010.

[13] I. Goodfellow, D. Warde-Farley, M. Mirza, A. Courville, and Y. Bengio, "Maxout networks," in *Proc. ICML*. International Machine Learning Society, 2013.

[14] G.E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R.R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," *arXiv:1207.0580*, 2012.

[15] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, November 1998.

[16] M. Cai, W.Q. Zhang, and J. Liu, "Improving deep neural network acoustic models using unlabeled data," in *Proc. ChinaSIP*. IEEE, 2013.

[17] V. Mnih, "CUDAMat: a CUDA-based matrix class for python," Tech. Rep. UTML TR 2009-004, Department of Computer Science, University of Toronto, 2009.

[18] R. Prabhavalkar, T. Sainath, D. Nahamoo, B. Ramabhadran, and D. Kanevsky, "An evaluation of posterior modeling techniques for phonetic recogntion," in *Proc. ICASSP*. IEEE, 2013, pp. 7165–7169.