IMPROVING LONG SHORT-TERM MEMORY NETWORKS USING MAXOUT UNITS FOR LARGE VOCABULARY SPEECH RECOGNITION

Xiangang Li, Xihong Wu

Speech and Hearing Research Center,
Key Laboratory of Machine Perception (Ministry of Education),
Peking University, Beijing, 100871
{lixg, wxh}@cis.pku.edu.cn

ABSTRACT

Long short-term memory (LSTM) recurrent neural networks have been shown to give state-of-the-art performance on many speech recognition tasks. To achieve a further performance improvement, in this paper, maxout units are proposed to be integrated with the LSTM cells, considering those units have brought significant improvements to deep feed-forward neural networks. A novel architecture was constructed by replacing the input activation units (generally tanh) in the LSTM networks with maxout units. We implemented the LSTM network training on multi-GPU devices with truncated BPTT, and empirically evaluated the proposed designs on a large vocabulary Mandarin conversational telephone speech recognition task. The experimental results support our claim that the performance of LSTM based acoustic models can be further improved using the maxout units.

Index Terms— long short-term memory, maxout, deep neural network, acoustic modeling, large vocabulary speech recognition

1. INTRODUCTION

Last decade has witnessed significant progress in automatic speech recognition (ASR), and most advances are triggered by the developing of new machine learning algorithms. As major advances have been made in deep neural networks (DNNs), almost all of the state-of-the-art ASR systems adopt DNNs as the acoustic modeling method (e.g. [1][2][3][4][5]).

Recently, in the researches of DNNs based acoustic modeling, long short-term memory (LSTM) based deep networks have been shown to give the state-of-the-art performance on some speech recognition tasks. In the seminal work, Graves et al. [6] proposed to use stacked bidirectional LSTM networks for phoneme recognition, which operate on the input sequence in both direction to make a decision for the current input. For the robust speech recognition, LSTM networks have been proved to be more efficient [7]. For the large vocabulary speech recognition, literature [8][9] has shown that LSTM networks can obtain notable performance improvement with thousands of context dependent (CD) states. Subsequently, the sequence discriminative training of LSTM networks is investigated in [10], and a significant gain was obtained.

However, in the literatures, various methods have been proposed to enhance the feed-forward DNNs in acoustic modeling. The importance of well-designed nonlinear activation functions has become more apparent recently. Novel nonlinear activation functions that are unbounded and often piecewise linear but not continuous such as rectified linear units (ReLU) [11] and maxout units [12] have been found to be more suited for deep networks. ReLU units simply choose the units output as $y = max(x, 0)$, and have been proved to outperform the standard sigmoid activations for acoustic modeling [13][14]. The maxout nonlinearity can be regarded as a generalization of ReLU, has given state-of-the-art performance in various computer vision tasks [12], and also achieved improvements in speech recognition tasks [15][16].

Inspired from the developing of nonlinear activation functions for DNNs, we attempt to discuss the activations in LSTMs. Besides, we also find that, [17] proposed to use maxout-like units in the deep recurrent neural networks, and obtained notable improvements on polyphonic music prediction task. Applying the maxout units in the LSTM networks is thus a natural choice. In this paper, a novel LSTM architecture is proposed to be constructed using maxout units for acoustic modeling. Experiments were conducted on a large vocabulary speech recognition task, and results show the performance of LSTM networks can be further improved using the maxout units, just like the deep maxout networks [12].

2. THE CONVENTIONAL LSTM ARCHITECTURE

Given an input sequence $x = (x_1, x_2, \ldots, x_T)$, an conventional recurrent neural network (RNN) computes the hidden vector sequence $h = (h_1, h_2, \ldots, h_T)$ and output vector sequence $y = (y_1, y_2, \ldots, y_T)$ from $t = 1$ to T as

$$h_t = \mathcal{H}(W_{xh}x_t + W_{hh}h_{t-1} + b_h) \quad (1)$$

$$y_t = W_{hy}h_t + b_y \quad (2)$$

Where the W denotes weight matrices, the b denotes bias vectors and $\mathcal{H}$ denotes the hidden layer function.

However, in practice, RNNs are hard to train properly due to the vanishing gradient and exploding gradient problems as described in [18]. To address these problems, long short-term memory (LSTM) is proposed [19]. The LSTM architecture

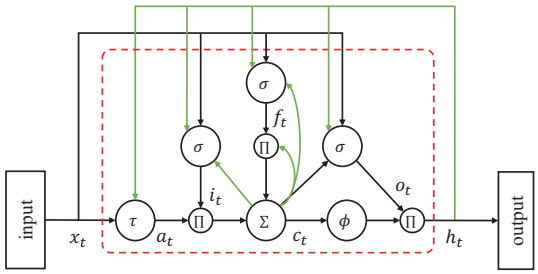

Fig. 1. The architecture of LSTM networks with one memory block, where green lines are time-delayed connections.

consists of a set of recurrently connected subnets known as "memory blocks". Each memory block contains one or more self-connected memory cells and three multiplicative gates to control the flow of information. In each LSTM cell, the flow of information into and out of the cell is guarded by the learned input and output gates. Later, in order to provide a way for the cells to reset themselves, the forget gate was added [20]. In addition, the modern LSTM architecture contains peephole weights connecting the gates to the memory cell, which improve the LSTM's ability to learn tasks that require precise timing and counting of the internal states [21]. As illustrated in Fig. 1, the equations of the LSTM memory blocks are as follows:

$$i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci}c_{t-1} + b_i) \quad (3)$$

$$f_t = \sigma(W_{xf}x_t + W_{hf}h_{t-1} + W_{cf}c_{t-1} + b_f) \quad (4)$$

$$a_t = \tau(W_{xc}x_t + W_{hc}h_{t-1} + b_c) \quad (5)$$

$$c_t = f_t c_{t-1} + i_t a_t \quad (6)$$

$$o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + W_{co}c_t + b_o) \quad (7)$$

$$h_t = o_t \phi(c_t) \quad (8)$$

Where, σ is the logistic sigmoid function, and i, f, o, a and c are respectively the input gate, forget gate, output gate, cell input activation, and cell state vectors, all of which are the same size as the hidden vector h. W_{ci}, W_{cf}, W_{co} are diagonal weight matrices for peephole connections, so element m in each gate vector only receives input from element m of the cell vector. τ and ϕ are the cell input and cell output activation functions, generally in the conventional LSTM $tanh$.

Besides, literature [8] proposed a novel LSTM architecture, called LSTM Projected (LSTMP), which has a separate linear projection layer after the LSTM layer. This LSTMP network has been applied on a large vocabulary speech recognition task, and yielded the state-of-the-art performance.

3. THE MAXOUT LSTM ARCHITECTURE

3.1. Maxout Units

The maxout units are proposed in the feed-forward DNNs [12], where the hidden units are divided into disjunct groups. Fig. 2 illustrates one hidden layer in a deep maxout network (DMN). We denote the number of units groups as K and the group size as G. The maxout nonlinearity would reduce the dimension from $K \times G$ to K. For each group of G neurons, the output would be the maximum of all the inputs:

$$h_i = max_{i=1}^{G} z_{ij} \quad (9)$$

where $z_{ij} = x^T W_{...ij} + b_{ij}$ is obtained by forward propagation from the layer below. In a deep maxout network (DMN), x is the lower maxout layer output vector or the whole network input vector. We can see that the maxout function applies a max pooling operation on z. Just like the pooling operator used in convolutional neural networks, this operator which summarizes a group of spatially neighboring neurons in a lower layer is able to achieve the property of local translation invariance. The difference from traditional nonlinearities is that the pooling operator is not applied element-wise on the lower layer, but rather on groups of hidden units, which leads to a dimension reduction of hidden units.

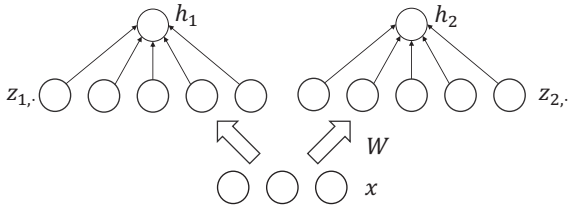

Fig. 2. Illustration of maxout layer with group size of 5.

3.2. Application to LSTMs

Maxout units have been found to be well suited for feed-forward DNNs, and we attempt to use the maxout units to improve the performance of LSTM networks.

First of all, in a conventional RNN, which mostly uses saturating nonlinear activations such as $tanh$ to compute the hidden state at each time step, it is hard to use the non-saturating activation functions such as maxout. However, the authors of [17] pointed out that the non-saturating activation function can be used in a deep RNN without causing the instability of the model when a saturating non-linearity ($tanh$) is also applied at the same time.

When we go back to the LSTM architecture, it can be easily found out that, beside the activations of the three gates, there are two non-linear functions, τ and ϕ as illustrated in Fig. 1, which are generally saturating non-linearity $tanh$. However, the output vector of ϕ is multiplied with the output

gate o_t, which means that there cannot be a dimension reduction in this non-linear function ϕ. Thus, a straight-forward proposal is that using the maxout units in the place of τ, which leads to the architecture illustrated in Fig. 3, and called as maxout LSTM (mLSTM for short) in this paper.

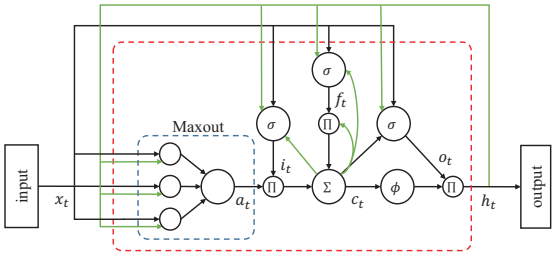

Fig. 3. The architecture of maxout LSTM networks with one memory block, where green lines are time-delayed connections and the group size of maxout unit is 3.

In the proposed mLSTM, the output activation ϕ in equation (8) is still the saturating nonlinearity $tanh$, thus the hidden states h_t of the memory blocks are bounded. This allows us to use any potentially non-saturating nonlinear function for τ, and here we use the maxout non-linearity. The equation for the cell input activation a_t in equation (5) is changed as:

$$a_t = max_{i=1}^{G}(W_{xci}x_t + W_{hci}h_{t-1} + b_{ci}) \qquad (10)$$

where G is the group size.

However, the maxout units can also been used in the LSTMP networks, which leads to a novel architecture called as maxout LSTMP (mLSTMP for short) in this paper.

4. EXPERIMENTS

We evaluate these LSTM networks on a large vocabulary speech recognition task - the HKUST Mandarin Chinese conversational telephone speech recognition [22]. The corpus (LDC2005S15, LDC2005T32) is collected and transcribed by Hong Kong University of Science and Technology (HKUST), which contains 150-hour speech, and 873 calls in the training set and 24 calls in the development set, respectively. In our experiments, around 10-hour speech was randomly selected from the training set, used as the validate set for network training, and the original development set in the corpus was used as speech recognition test set, which is not used in the training or the hyper-parameters determination procedures.

4.1. Experimental setup

The speech in the dataset is represented with 25ms frames of Mel-scale log-filterbank coefficients (including the energy value), along with their first and second temporal derivatives. In the experiments, the feed-forward DNNs used the concatenated features, which were produced by concatenating the current frame with 5 frames in its left and right context. However, for the inputs of LSTM networks, only current features (no context) were used.

A trigram language model was used in all the experiments, which was estimated using all the acoustic model training transcriptions. The hybrid approach [4][8] is used for acoustic modeling with LSTM networks or DNNs, in which the neural networks' outputs are converted as pseudo likelihood as the state output probability in hidden Markov model (HMM) framework. All the networks were trained based on the alignments generated by a well-trained GMM-HMM systems with 3304 tied context dependent HMM states (realignments by DNNs were not performed), and only the cross-entropy objective function was used for all networks.

We implemented the LSTM network training on multi-GPU devices. In the training, the truncated back-propagation though time (BPTT) learning algorithm [23] is adopted. Each sentence in the training set is split into subsequences with equal length (15 frames in the experiments), and two adjacent subsequences have overlapping frames (5 frames in the experiments). For computational efficiency, one GPU operates in parallel on 20 subsequences from different utterances at a time. In order to train these networks on multi-GPU devices, asynchronous stochastic gradient descent (ASGD) [24][25] is adopted. The strategy introduced in [26] was applied to scale down the gradients. Since the information from the future frames helps making better decisions for current frame, we also delayed the output HMM state labels by 3 frames.

In the experiments, the learning rate for training each network was decreased exponentially. We tried to set the initial and final learning rates specific to a network architecture for stable convergence of training. The initial learning rates ranged from 0.0005 to 0.002, and each final learning rate was always set as one-tenth of the corresponding initial one.

4.2. Experimental results

Firstly, the character error rates (CER) of the baseline systems are summarized in Table 1. For training the Subspace GMM [27], KALDI toolkit [28] was used. All the DNNs in the experiments had 4 hidden layers. Each layer in the "ReLU DNN" model had 2000 ReLU units. Each layer in the "Maxout DNN"(or the DMN) model had 800 maxout units, where the group size is 3. Each layer in the "PNorm DNN" model had 800 pnorm units [29], where the hyper-parameter p is set to 2, and the group size is set to 8. It can be found out that, the performance of baseline GMM-HMM and DNN-HMM is comparable with that reported in [30][31][32].

For the proposed mLSTM architecture, we firstly explored the effects of different group size of the maxout units. Experiments were conducted based on the 1-layer shallow LSTM network. The number of LSTM cells is fixed on 750, but we varied the group size to 2, 3, 4 and 5 for the mLST-

Table 1. Recognition results of the baseline systems on the HKUST speech recognition task.

Model Descriptions	CER(%)
GMM	48.68
Subspace GMM	44.29
ReLU DNN	38.42
Maxout DNN	38.09
PNorm DNN	**38.01**

M networks. From the experimental results in Table 2, we can find out that, although the performance of the shallow mLSTM network is still worse than the baseline DMN, the performance of shallow LSTM network was improved by using the maxout units. A 4.37% relatively CER reduction can be obtained by setting the group size to 4.

Table 2. Recognition results about maxout LSTM networks. Each network had one LSTM hidden layer.

Model Descriptions	Group Size	CER(%)
LSTM	-	40.28
Maxout LSTM	2	39.33
Maxout LSTM	3	39.05
Maxout LSTM	4	**38.53**
Maxout LSTM	5	38.96

Then, we compared the proposed mLSTMP with LSTMP, and the results are listed in Table 3. In these networks, there were only one recurrent layer with 2000 LSTM memory cells, and 750 nodes in the linear projection layer. The the group size of mLSTMP network is set to 4. The experimental results show that the LSTMP network can also been improved using the maxout units.

Table 3. Recognition results about LSTMP networks. Each network had one hidden layer, and the group size is 4.

Model Descriptions	CER(%)
LSTMP	35.92
Maxout LSTMP	**35.07**

Deep LSTM networks have been shown to be more expressive models, thus, some further experiments were conducted for the deep LSTM networks. We constructed deep LSTM networks by simply stacking three LSTM or LSTMP layers, in which, each layer had the same configurations as those in the experiments described above. Compared the results showed in Table 4 with the one-layer LSTM networks, the stacked LSTMs networks indeed yielded better performances, and the performance of stacked LSTM and stacked LSTMPs networks can also been improved using the maxout units. The best performance can be obtained by the stacked mLSTMPs network, which reduced the CER from 34.84% using stacked LSTMPs network to 33.89% using stacked mLSTMPs network.

Table 4. Recognition results about deep LSTM networks. Each network had 3 hidden layers, and the group size is 4.

Model Descriptions	CER(%)
Stacked LSTMs	35.91
Stacked LSTMPs	34.84
Stacked maxout LSTMs	34.81
Stacked maxout LSTMPs	**33.89**

From these results, we can find out that, importantly, using maxout units can improve these LSTM networks in all the cases, where the relatively CER reductions ranged from 2.3% to 4.4%. Compared with the feed-forward DNNs, the stacked mLSTMPs network can reduce the CER from 38.01% to 33.89%, which is a 10.84% relatively CER reduction.

5. DISCUSSION AND CONCLUSIONS

Maxout units have brought significant improvements to feed-forward DNNs on various speech recognition and computer vision tasks. We investigate approaches to use the maxout units to improve the LSTMs by studying the non-linearity functions in LSTMs. The proposed architectures came out by replacing the input activation units (generally $tanh$) in LSTM or LSTMP networks with the maxout units, and the hidden states h_t of the LSTM memory blocks are still bounded by the saturating nonlinearity $tanh$ in the output activation units.

We empirically evaluated the proposed designs against the conventional LSTM and LSTMP networks on a large vocabulary Mandarin conversational telephone speech recognition task. The experimental results revealed that these LSTM networks can be improved using the maxout units. Although, the LSTM networks have reached the state-of-the-art performance on some speech recognition tasks, these experiments suggested that the performance can be further improved using the maxout units, just like the DMNs.

However, in the literatures, there are some generalized maxout units proposed, such as the soft-Maxout and p-norm units [29] and L_p units [17], which will been explored for the LSTM based acoustic modeling in our future work.

6. ACKNOWLEDGEMENTS

The work was supported in part by the National Basic Research Program (2013CB329304), the research special fund for public welfare industry of health(201202001), and National Natural Science Foundation (No.61121002, No.91120001).

7. REFERENCES

[1] A. Mohamed, G. Dahl, and G. Hinton, "Acoustic modeling using deep belief networks," *IEEE Trans. Audio Speech Lang. Processing*, vol. 20, pp. 14–22, 2012.

[2] G. Hinton, L. Deng, D. Yu, and et al., "Deep neural networks for acoustic modeling in speech recognition: the shared views of four research groups," *IEEE Signal Processing Mag*, vol. 29, pp. 82–97, 2012.

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

[4] G. Dahl, D. Yu, L. Deng, and A. Acero, "Context-dependent pretrained deep neural networks for large-vocabulary speech recognition," *IEEE Trans. Audio Speech Lang. Processing*, vol. 20, pp. 30–42, 2012.

[5] N. Jaitly, P. Nguyen, A. Senior, and V. Vanhoucke, "Application of pretrained deep neural networks to large vocabulary speech recognition," in *Interspeech*, 2012, pp. 2577–2580.

[6] A. Graves, A. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *ICASSP*, 2013, pp. 6645–6649.

[7] J. Geiger, X. Zhang, F. Weninger, and et al., "Robust speech recognition using long short-term memory recurrent neural networks for hybrid acoustic modelling," in *Interspeech*, 2014, pp. 631–635.

[8] H. Sak, A. Senior, and F. Beaufays, "Long short-term memory recurrent neural network architectures for large scale acoustic modeling," in *Interspeech*, 2014, pp. 338–342.

[9] H. Sak, A. Senior, and F. Beaufays, "Long short-term memory based recurrent neural network architectures for large vocabulary speech recognition," arXiv:1402.1128, 2014.

[10] H. Sak, O. Vinyals, G. Heigold, and et al., "Sequence discriminative distributed training of long short-term memory recurrent neural networks," in *Interspeech*, 2014, pp. 1209–1213.

[11] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," *Journal of Machine Learning Research - Proceedings Track*, vol. 15, pp. 315–323, 2011.

[12] I. Goodfellow, D. Warde-Farley, M. Mirza, A. Courville, and Y. Bengio, "Maxout networks," arXiv:1302.4289, 2013.

[13] M. Zeiler, M. Ranzato, R. Monga, and et al., "On rectified linear units for speech processing," in *ICASSP*, 2013, pp. 3517–3521.

[14] G. Dahl, T. Sainath, and G. Hinton, "Improving deep neural networks for lvcsr using rectified linear units and dropout," in *ICASSP*, 2013, pp. 8609–8613.

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

[16] Y. Miao, S. Rawat, and F. Metze, "Deep maxout networks for low resource speech recognition," in *ASRU*, 2013, pp. 398–403.

[17] C. Gulcehre, K. Cho, R. Pascanu, and Y. Bengio, "Learned-norm pooling for deep feedforward and recurrent neural networks," arXiv:1311.1780, 2014.

[18] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," *IEEE Trans. Neural Networks*, vol. 5, pp. 157–166, 1994.

[19] S. Hochreiter and J. Schimidhuber, "Long short-term memory," *Neural Computation*, vol. 9, pp. 1735–1780, 1997.

[20] F. Gers, J. Schmidhuber, and F. Cummins, "Learning to forget: Continual prediction with lstm," *Neural Computation*, vol. 12, pp. 2451–2471, 2000.

[21] F. Gers, N. Schraudolph, and J. Schmidhuber, "Learning precise timing with lstm recurrent networks," *Journal of Machine Learning Research*, vol. 3, pp. 115–143, 2003.

[22] Y. Liu, P. Fung, Y. Yang, and et al., "Hkust/mts: A very large scale mandarin telephone speech corpus," in *ISCSLP*, 2006, pp. 724–735.

[23] R. Williams and J. Peng, "An efficient gradient-based algorithm for online training of recurrent neural network trajectories," *Neural Computation*, vol. 2, pp. 490–501, 1990.

[24] R. Ormándi, I. Hegedüs, and M. Jelasity, "Asynchronous peer-to-peer data mining with stochastic gradient descent," *Lecture Notes in Computer Science*, pp. 528–540, 2011.

[25] S. Zhang, C. Zhang, Z. You, R. Zheng, and B. Xu, "Asynchronous stochastic gradient descent for dnn training," in *ICASSP*, 2013, pp. 6660–6663.

[26] R. Pascanu and Y. Bengio, "On the difficulty of training recurrent neural networks," arXiv:1211.5063, 2012.

[27] D. Povey, L. Burget, M. Agarwal, and et al., "The subspace gaussian mixture model - a structured model for speech recognition," *Computer Speech & Language*, vol. 25, pp. 404–439, 2011.

[28] D. Povey, A. Ghoshal, L. Burget, and et al., "The kaldi speech recognition toolkit," in *ASRU*, 2011, pp. 1–4.

[29] X. Zhang, J. Trmal, D. Povey, and S. Khudanpur, "Improving deep neural network acoustic models using generalized maxout networks," in *ICASSP*, 2014, pp. 215–219.

[30] C. Weng and B. Juang, "Adaptive boosted non-uniform mce for keyword spotting on spontaneous speech," in *ICASSP*, 2013, pp. 6960–6964.

[31] C. Ni, N. Chen, and B. Ma, "Multiple time-span feature fusion for deep neural network modeling," in *ISCSLP*, 2014, pp. 138–142.

[32] Y. Liu, X. Li, and X. Wu, "Margin-based discriminative pronunciation modeling for large vocabulary mandarin speech recognition," in *SLT*, 2014.