STOCHASTIC POOLING MAXOUT NETWORKS FOR LOW-RESOURCE 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

Maxout network is a powerful alternate to traditional sigmoid neural networks and is showing success in speech recognition. However, maxout network is prone to overfitting thus regularization methods such as dropout are often needed. In this paper, a stochastic pooling regularization method for maxout networks is proposed to control overfitting. In stochastic pooling, a distribution is produced for each pooling region by the softmax normalization of the piece values. The active piece is selected based on the distribution during training, and an effective probability weighting is conducted during testing. We apply the stochastic pooling maxout (SPM) networks within the DNN-HMM framework and evaluate its effectiveness under a low-resource speech recognition condition. On benchmark test sets, the SPM network yields 4.7-8.6% relative improvements over the baseline maxout network. Further evaluations show the superiority of stochastic pooling over dropout for low-resource speech recognition.

Index Terms— stochastic pooling, maxout network, speech recognition, low-resource, deep learning

1. INTRODUCTION

In recent years, deep learning methods have achieved great success in automatic speech recognition (ASR). From small scale TIMIT phoneme recognition task [1] to large vocabulary continuous speech recognition (LVCSR) [2, 3, 4, 5], deep neural network (DNN) based systems surpass Gaussian mixture model-hidden Markov model (GMM-HMM) systems by a large scale. Because of the advantages of deep learning methods, more and more researchers in both speech recognition and machine learning communities are keen to explore more powerful models [6] and many new types of deep learning methods are emerging for speech recognition [7].

Maxout network is a new type of deep neural network that sets the state-of-the-art in many machine learning tasks [8]. Recently it shows good performance in speech recognition [9, 10]. In maxout network, the neurons select their activations by max pooling across a group of linear pieces. The activations are then directly passed to the next layer without any further nonlinear transformations. Despite its simplicity, the maxout network has several attractive features compared to traditional sigmoid networks. First, the maxout network yields better optimization performance. During back-propagation (BP) training, the gradients always equal to one for the selected piece, which avoids the saturation problem described in [11]. Second, the maxout network has fewer activations compared to the sigmoid network with the same size. The architecture of maxout network will naturally result in sparse gradient matrix during BP training, which also facilitates optimization. Thanks to the good optimization performance of the maxout network, very deep model can be trained without pre-training. The maxout network can effectively deal with the problem of underfitting, but is sometimes prone to overfitting. Because of this, one has to reduce the model size, use relatively smaller learning rate, or use regularization methods such as dropout [12] to control overfitting.

In this paper, we propose a novel regularization for maxout networks. Instead of using max pooling, a stochastic pooling operation is applied to the piece selection process in maxout networks. For the stochastic pooling maxout (SPM) network training, a distribution is assigned to each piece group with respect to the corresponding piece values, and the piece selection takes place according to the probability distribution. For the SPM network testing, the activation value is a weighted sum of the pieces in the corresponding piece group. Our experimental results under a low-source speech recognition condition show that the proposed SPM network makes more robust parameter estimation with limited training data, and it reduces the word error rate (WER) by 4.7-8.6% relatively compared to the maxout network baseline.

The remainder of this paper is organized as follows: In Section 2, we briefly review the maxout network and the dropout regularization. In Section 3, we propose the stochastic pooling maxout network and analyze its properties. We report our experiments in detail in Section 4 and finally draw the conclusions in Section 5.

This work is supported by National Natural Science Foundation of China under Grant No. 61370034, No. 61273268, No. 61005019 and No. 61005017.

978-1-4799-2893-4/14/$31.00 ©2014 IEEE

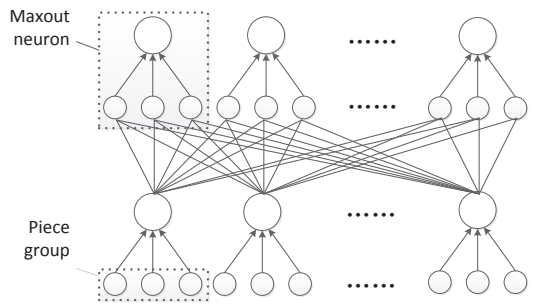

Fig. 1. Illustration of the maxout network.

2. MAXOUT AND DROPOUT

Maxout network [8] is a feed-forward neural network. A brief illustration is shown in Fig. 1. A maxout neuron consists of a group of linear pieces, typically 2 to 5. Let z_l^{ij} be the jth piece of piece group i at layer l, the corresponding activation h_l^i is the maximum value of the pieces in the piece group, i.e.,

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

where k is the number of pieces. This temporal maximum value selection process is also referred to as max pooling, which has been studied in accompany with the convolutional neural network (CNN) [13] for a long time. Max pooling operation has the potential benefit to select useful features and make robust estimations, but there are also discoveries that CNNs with max pooling may overfit the training set easily [14]. For maxout networks, this overfitting problem may be more serious because the maxout network involves a lot of distinct max pooling groups across all hidden layers, without weight sharing [13, 15] or sigmoid nonlinearity.

To control overfitting, regularization methods are often needed, among which dropout [12] is a good choice. Dropout avoids complex co-adaptations of neural network parameters by randomly omitting a fraction of neurons for every training case. So for each training case, a different sub-network is updated. Let r be the dropout probability, for the lth layer, the activation $\mathbf{h}_l$ becomes

$$\mathbf{h}_l = \mathbf{m}_l * \sigma(\mathbf{W}_{l-1}^T \cdot \mathbf{h}_{l-1} + \mathbf{b}_{l-1}) \tag{2}$$

where $\mathbf{W}_{l-1}$ and $\mathbf{b}_{l-1}$ are weight matrix and bias vector, the σ is a nonlinear function such as sigmoid function or max pooling function as in Eq. (1), the $*$ denotes element-wise multiplication, the $\mathbf{m}_l$ is the dropout mask, whose elements m_l^i obeys the distribution $Bernoulli(1-r)$. For dropout testing, all the network parameters are used, but the activations are scaled by a normalization factor $(1-r)$. This can be viewed as a model averaging process [16], performing an effective fusion of many sub-models in a single forward pass.

3. STOCHASTIC POOLING MAXOUT NETWORK

In this section, the stochastic pooling maxout (SPM) network is presented. We give the model description first and then analyze its properties.

3.1. Model description

The SPM network shares the same topology with maxout network. But instead of using max pooling during training and testing, a stochastic pooling method is used, which is inspired by the successful application with CNN [14].

In stochastic pooling, the pieces are randomly selected according to a distribution P determined by the values of the pieces. Let $\mathcal{M}_l^i = \{z_l^{ij} | j \in 1...k\}$ be piece group i of layer l. For each z_l^{ij} in $\mathcal{M}_l^i$, a probability p_l^{ij} is computed. Although any monotonically non-negative function can be used to map the piece value z_l^{ij} into probability p_l^{ij}, in practice we find the softmax function works just fine[1]:

$$p_l^{ij} = \frac{\exp(z_l^{ij})}{\sum_h \exp(z_l^{ih})}. \tag{3}$$

Let n be the index of the selected piece. After obtaining the distribution $P(p_l^{i1}, p_l^{i2}, ...p_l^{ik})$, the value of n obeys:

$$n \sim P(p_l^{i1}, p_l^{i2}, ...p_l^{ik}) \tag{4}$$

which makes the pooling process stochastic.

For the SPM network training, stochastic pooling is applied to the piece groups $\mathcal{M}_l^i$ during the forward-propagation. During the backward-propagation, gradients are propagated through the selected pieces. The model parameters corresponding to the selected pieces are updated, while parameters corresponding to the unselected pieces remain unchanged. This training process also results in sparse gradient matrix as in the maxout network.

For the SPM network testing, the activation h_l^i of group $\mathcal{M}_l^i$ is a weighted sum of the piece values rather than stochastic:

$$h_l^i = \sum_j p_l^{ij} \cdot z_l^{ij}. \tag{5}$$

This probability weighting scheme makes use of all piece values, which also has the effect of model averaging and fusion.

3.2. Model analysis

The stochastic pooling controls overfitting by making the piece selection a stochastic process. We argue that the SPM network is effective due to two main reasons.

First, the randomness introduced by stochastic pooling gives the model better chances to escape from local maxima

[1]We actually also tried the normalized sigmoid function $p_l^{ij} = \text{sigmoid}(z_l^{ij}) \Big/ \sum_h \text{sigmoid}(z_l^{ih})$, but the results are not as good as those obtained with the softmax function, so we choose not to report them.

during training. For maxout network, max pooling will enforce the pieces with large activation values while ignore other pieces. But when training data is limited, the pieces with large activation values will likely to be updated again and again. Though the model will converge fast on the training set, it may not generalize well on the test set. The stochastic pooling, on the other hand, will force the whole parameter space to be thoroughly searched by giving every piece a chance. Dropout also controls overfitting by introducing randomness, but it is not particularly designed for neural networks with pooling regions.

Second, the probability term controls the balance between the model's confusion and certainty. The max pooling is a winner-take-all action. It makes a very certain decision, which is effective when a strong model with much training data is used. But with limited training data, the model's confusion needs to be taken into account. The probability weighting scheme is a flexible way to consider all piece values for the final decision making.

4. EXPERIMENTAL RESULTS

4.1. Experiment setup

Since the stochastic pooling is a regularization to control overfitting, we are particularly interested in the performance evaluation of the SPM network under low-resource speech recognition conditions. We choose a 24-hour subset of the Switchboard phone-call corpus for DNN training to simulate the low-resource speech recognition condition. The SWB part of Hub5'00 set acts as development set and the FSH part of RT03S set acts as the test set.

The GMM-HMM uses all 300 hours of training data. We first extract 13 dimensional PLP features with mean-covariance normalization and then concatenate the basic features with their first, second and third order derivatives. The features dimension are further reduce to 39 by HLDA. The GMM-HMM model contains 9308 states with 40 Gaussian mixtures per state. A trigram language model is trained using the transcription of the 2000-hour fisher corpus and is interpolated with a more general trigram.

For the DNN-HMM training, a 24-hour subset of training data is used as in [9]. We extract 40 dimensional filter-bank features along with their first and second order derivatives. The features are normalized to have zero mean and unit variance based on speaker-side information. A context window of 11 frames is used for the input to DNN.

4.2. Influence of group size

The first thing we investigate is the influence of group sizes. For the SPM networks, the number of hidden layers is fixed to 7 and the number of units is fixed to 720, while the numbers of pieces are varied among 2, 3 and 4. A maxout network with 7 hidden layers, 720 units per hidden layer and 2 pieces per unit

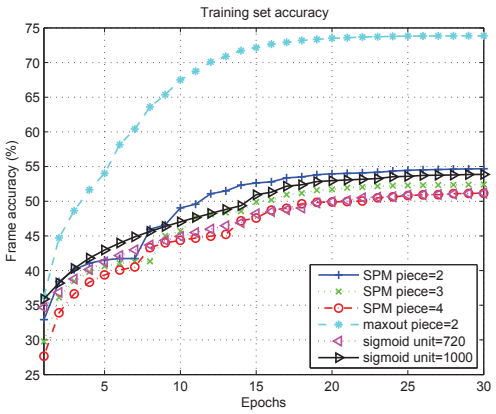

Fig. 2. Frame accuracy on the training set.

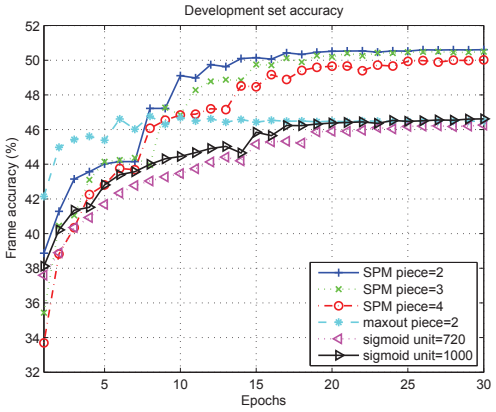

Fig. 3. Frame accuracy on the development set.

is trained as the baseline[2]. To make comparisons, two sigmoid networks are also trained, one contains 7 hidden layers and 720 units per hidden layer, the other contains 7 hidden layer and 1000 units per hidden layer (which has roughly the same size as the maxout network).

The neural network training strategies are as follows: The parameters are randomly initialized with a normalized uniform distribution as proposed in [11]. The learning rates are tuned on the development set, which is 0.08 for the SPM networks and the sigmoid networks, and 0.02 for the maxout network. At the end of every epoch, the learning rate is reduced by a factor of 2 if the frame accuracy on the development set drops. The momentum value starts off with 0, and then increases to 0.5 linearly after 20 epochs. For the sigmoid network, we also perform discriminative pre-training as proposed in [17], while the SPM and maxout networks are trained without pre-training. Our implementations are based on the CUDAMat library [18] with many extended functions.

[2]Experimental results in [9] show that maxout network with 2 pieces works best for this task.

Table 1. Speech recognition results on 24 hours of Switchboard training data. Performances are measured in WER given in %.

Model	Hub5'00-SWB	RT03S-FSH
sigmoid 720 units	27.9	31.2
sigmoid 1000 units	25.3	29.0
maxout 2 pieces	23.2	27.9
SPM 2 pieces	**22.1**	**25.5**
SPM 3 pieces	22.6	25.8
SPM 4 pieces	23.2	26.2

The results in Fig. 2 and Fig. 3 illustrates the frame accuracy on the training and test set as learning progresses. From the figures we see the distinct behaviours between the SPM network and the maxout network. On the training set, the maxout network quickly converges to achieve a high accuracy, while on the development set, the SPM networks achieve better results. This suggests that stochastic pooling controls overfitting and improves model generalization. Results also show that the SPM network with 2 pieces have the best performance. Speech recognition results in Table 1 further verify the effectiveness of the SPM networks.

4.3. Influence of model depth

The depth of DNN has a large impact on the its performance. In [9], the WER falls steadily as the number of hidden layers of the maxout network increases from 5 to 9. It's interesting to explore the effects of model depths to SPM networks.

For the model depth evaluations, the number of units is fixed to 720 and the number of pieces is fixed to 2 for both the SPM and the maxout network. We try to train the networks with 5, 7, 9 and 11 hidden layers and the results are shown in Fig. 4. These results show that the SPM network also benefits from the growth of model depth. The optimal number of hidden layers for the SPM network is 9 for this task, while for the maxout network, the optimal number is even larger. For different hidden layer numbers, the performance gap between the SPM network and the maxout network is larger on the test set than on the development set, suggesting that the SPM network has better generalization abilities.

4.4. Influence of dropout

Like stochastic pooling, dropout is another way to control overfitting. Moreover, dropout could even be combined with stochastic pooling, at least intuitively. We study the influence of dropout to the SPM network and the maxout network, respectively.

For the dropout training, the networks are fixed to have 7 hidden layers, 720 units and 2 pieces. The dropout probability r is searched over 0.05, 0.1, 0.2 and 0.3 using the development set. We also use larger learning rate as suggested in [12], which is 0.12 for the SPM network and 0.04 for the maxout network. Table 2 compares the results with and without dropout training. The optimal r is 0.2 for the maxout network and 0.05 for the SPM network. These results suggests that dropout is effective for the maxout network, but is unnecessary for the SPM network, as stochastic pooling already controls overfitting effectively. Also, the stochastic pooling regularization is more effective for this task than dropout.

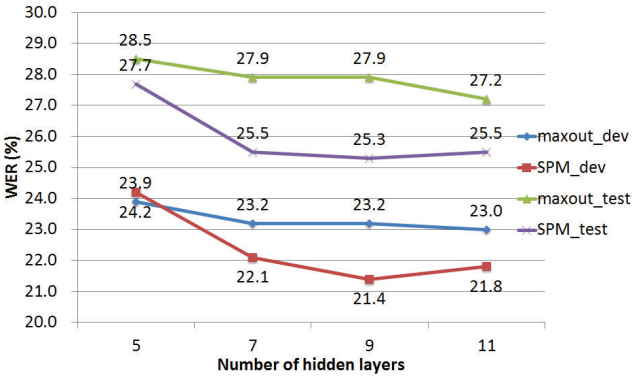

Fig. 4. Speech Recognition results with SPM networks and maxout networks with different depths. Performances are measured in WER given in %.

Table 2. Evaluation of dropout training for maxout and SPM networks. Performances are measured in WER given in %.

Model	Hub5'00-SWB	RT03S-FSH
maxout	23.9	28.5
maxout + dropout	22.7	26.2
SPM	22.1	25.5
SPM + dropout	22.2	25.6

5. CONCLUSIONS

In this paper, we have proposed the stochastic pooling maxout (SPM) networks for low-resource speech recognition. During the SPM network training, the selection of pieces is based on a distribution determined by the piece values, while for the SPM network testing, an effective model averaging schemes is made by probability weighting of the piece values. Our experiments on a 24-hour subset of the Switchboard training corpus verify that the stochastic pooling method controls overfitting effectively. Results show that SPM network with 2 pieces works best. Moreover, SPM network benefits from model depth and it shows better generalization ability on the test set. The evaluations with dropout suggests dropout is not necessary for the SPM network and stochastic pooling works better for this low-resource speech recognition task.

6. REFERENCES

[1] 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.

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

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

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

[6] L. Deng and X. Li, "Machine learning paradigms for speech recognition: An overview," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 21, no. 5, pp. 1060–1089, 2013.

[7] L. Deng, G. Hinton, and B Kingsbury, "New types of deep neural network learning for speech recognition and related applications: an overview," in *Proc. ICASSP*. IEEE, 2013, pp. 8599–8603.

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

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

[10] Y. Miao, F. Metze, and S. Rawat, "Deep maxout networks for low-resource speech recogntion," in *Proc. ASRU*. IEEE, 2013, pp. 398–403.

[11] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proc. AISTATS*, 2010, pp. 249–256.

[12] 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.

[13] 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.

[14] M.D. Zeiler and R. Fergus, "Stochastic pooling for regularization of deep convolutional neural networks," in *Proc. ICLR*, 2013.

[15] O. Abdel-Hamid, A. Mohamed, H. Jiang, and G. Penn, "Applying convolutional neural networks concept to hybrid NN-HMM model for speech recogntion," in *Proc. ICASSP*. IEEE SPS, 2012, pp. 4277–4280.

[16] J. Li, X. Wang, and B. Xu, "Understanding the dropout strategy and analyzing its effectiveness on LVCSR," in *Proc. ICASSP*. IEEE SPS, 2013, pp. 7614–7618.

[17] 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.

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