Self-attentional Convolution for Neural Networks

Qingyu Li\textsuperscript{a} and Zhenjiang Miao\textsuperscript{b}

Institute of Information Science, Beijing Jiaotong University, Beijing, P. R. China. 100044

E-mail: \textsuperscript{a}15112090@bjtu.edu.cn; \textsuperscript{b}zjmiao@bjtu.edu.cn

Abstract. Convolutional neural networks (CNNs) have proven to be effective models for tackling a variety of visual tasks. For each convolutional layer, a set of filters are learned to express local spatial connectivity patterns along inputs, but with the complexity of the structure growing, the network training becoming harder and harder, one of the reasons is over-fitting. Many methods are proposed recently in order to reduce this factor by network regularization or normalization, but few of them put the thought of structural control into consideration. In this paper, we propose a new method called Self-attentional Convolution for learning and regularizing the structure of convolutional layers by using the thoughts of attention models, which is more efficient than the original convolutional layers. For calculable and reasonable, we divide the attention weight factors into two parts, channels and shape of the kernels, as the structural constraints for each convolution layer from different views, and multiply them as the global attentional factors for the weights of the convolutional kernel. At last, several experiments are designed, and the final improvement is about 1% on 20, 32, 44, 56 and 110 layers of ResNets on CIFAR-10, and 0.8% of AlexNet on ImageNet dataset, which can prove the effectiveness of our method.

1. Introduction

Convolutional neural networks (CNNs) have proven to be effective models for tackling a variety of visual tasks. For each convolutional layer, a set of filters are learned to express local spatial connectivity patterns along input channels. In other words, convolutional filters are expected to be informative combinations by fusing spatial and channel-wise information together within local receptive fields. By stacking a series of convolutional layers interleaved with non-linearity functions and down-samplings, CNNs are capable of capturing hierarchical patterns with global receptive fields as powerful image descriptions.

But with the complexity of the structure growing, the network training becoming harder and harder, one of the reasons is over-fitting. In order to reduce this factor, many methods for simplifying the models are appearing.

For each convolutional layer, the inner weight can be described as a form of $[\cdot]_{K\cdotC\cdotH\cdotW}$, in which $K, C, H, W$ means amount of the kernels, the number of input channels, height and width of each kernel separately. All the methods based on directing the structure of the kernel could be considered as multiplying a weight tensor with the same shape as the original inner weight parameters.

In this paper, we proposed a method which used two attention models on weights of convolution layers, assisting the structure training by considering the kernels’ structure by channels and plain shapes. The product of the outputs builds up the shape-considered weight tensor.
In order to evaluate the proposed method, the experiments based on a modified ResNet with CIFAR-10 dataset and a modified AlexNet with ImageNet ILSVRC 2012 dataset are deployed.

The rest of this paper is structured as follows. In Section 2, we introduce the related methods of attention models and neural network structures learning; in Section 3, we highlight our intuition and introduce its implementation details. Section 4 experimentally analyses our method and Section 5 draws the conclusions.

2. Related Work

2.1. Neural Network Structures Learning

In order to make the network robust, many methods are deployed recently. Normalizations are used to regularize the weights in a global view, the mostly common used L2 normalization is also called weight decay. Dropout strategy [1] is randomly considering some of the edges in the network during the training period for learning a stable structure, and use the averaged result as the prediction. Pruning methods view the whole network as a dense graph, which could be pruned to a sparse one with a similar or even better performance because of over-fitting reduced. Han [2, 3] considered the distribution of the weights, deleting the edges with small weights which are below a threshold for edges cutting and structures learning. Guo [4] gives a splicing policy on Han’s method to restore some weights during training. Liu [5] considers the structure of Batch Normalization, in which the scaling factors could be used as the intensity weights for each channel. Both of them get a smaller model with an ever higher accuracy. But these researches only consider the numerical distribution, but the relationship involved in the inner relationship of the elements in kernels are ignored.

Group Lasso methods [6] lead a consideration on shape of the kernels, as a locally regularization, considers not only numerical, but also structural, but is hard to train due to its huge amount of hyper parameters.

2.2. Attention Models

Attention models could be viewed as a tool for giving the input information a weight factor regarding to itself, which could make the important information obviously and weaken the others. It is implemented by a combination of a part of neural network and a gating function (e.g. Softmax or Sigmoid).

The normally usage is use the visual attention to determine the patches of an image to focus on, and then classifying based on the areas focused, used in image captioning [7], visual question answering [8], optical characters recognition [9] and image classification [10].

These method based on regarding to the input feature, selecting the most meaningful information and providing a result. In this paper, we use attention model for elements relationship analysing, in which the network parameters as inputs, scaling factors are learned while training. In order for trainable and reasonable, the factors are calculated from both channel-wise and shape-wise.

3. Method

3.1. Motivation

![Figure 1](image-url)  
**Figure 1.** Attention model description for a specific channel of convolution kernels
For each convolutional layer, the inner weight can be described as a form of \(W_{k,c,h,w}\), in which \(K,C,H,W)\) means amount of the kernels, the number of input channels, height and width of each kernel separately. In the training period, the parameters are adjusted by the gradients in order to minimize the loss function. But for each parameter, this kind of adjusting only considers a single point, ignoring the influences between parameters, which could be thought as a kind of locally relationship in shape.

In order to import a limitation for the constraint between parameters, the network structure transforming could be show in a form as follows for each element.

\[
W'_{i,j,k,l} = W_{i,j,k,l} \cdot M_{i,j,k,l}
\]  

(1)

In the formulation, the original weight parameter \(W_{(i,j,k,l)}\), which at the location of \((k,l)\) in the \(jth\) channel of \(ith\) kernel, is eager to be multiplied by another value \(M_{(i,j,k,l)}\) which used as a limitation calculated among all other parameters, then \(W'_{(i,j,k,l)}\) is calculated and used for convolutional computation.

We propose to use attention model on the convolution weights, paying attention to the structure of convolutional layers and providing an importance weight for each element. The final weight could be expressed as following.

\[
W^a_{(i,j,k,l)} = W_{(i,j,k,l)} \cdot att_{(i,j,k,l)}
\]  

(2)

In which \(att\) is the output of the attention model where original weight \(W\) is used as the input. As shown in Figure 1, due to a consideration of the relationship within the kernel shape as well as the limitation of computation, it could be divided into two parts, channel wise attention and shape wise attention.

\[
W^a_{(i,j,k,l)} = W_{(i,j,k,l)} \cdot att^c_{(j)} \cdot att^s_{(i,k,l)}
\]  

(3)

3.2. Channel Attention

Channels are used for describing the information in different views, when a filter collecting the input channels to a specific output, each input should be supposed to have different importance. Channel attention implements this idea.

For each input channel, the mean value of the weights in this slice can be viewed as intensity of it, which is used as representation

\[
avg^c_{(i,j)} = mean(W_{(i,j,:)}) = \frac{1}{WH} \sum_{k=0}^{H-1} \sum_{l=0}^{W-1} W_{(i,j,k,l)}
\]  

(4)

Then, the channel-wise means are putted into a fully connected layer, analysing the inner structure information, and a sigmoid activation as a gate.

\[
att^c = sigmoid(fc^c(avg^c))
\]  

(5)

3.3. Shape Attention

The shape of a kernel is used for picking up information at a specific location with different importance. The representations for each location are extracted by a channel-wise mean.

\[
avg^s_{(i,j,k,l)} = mean(W_{(i,j,:)}), = \frac{1}{C} \sum_{j=0}^{C-1} W_{(i,j,k,l)}
\]  

(6)

The following procedure is similar as above.

\[
att^s = sigmoid(fc^s(avg^s))
\]  

(7)
Table 1. Training progress of the self-attention convolution.

| Input: Original weights $W$ | Output: Attentional weights $W^a$ |
|-----------------------------|----------------------------------|
| $avg^c = [\cdot]_{K \cdot C}$ | $avg^c = [\cdot]_{K \cdot C}$ |
| $for$ $i$ $in$ $\{0, 1, ..., K - 1\}$ | $for$ $i$ $in$ $\{0, 1, ..., K - 1\}$ |
| $for$ $j$ $in$ $\{0, 1, ..., C - 1\}$ | $for$ $j$ $in$ $\{0, 1, ..., C - 1\}$ |
| $avg_{i,j}^c = mean(W_{(i,j,:)})$ | $end$ |
| $end$ | $end$ |
| $att^c = sigmoid(fc^c(avg^c))$ | $avg^s = [\cdot]_{K \cdot H \cdot W}$ |
| $for$ $i$ $in$ $\{0, 1, ..., K - 1\}$ | $for$ $i$ $in$ $\{0, 1, ..., K - 1\}$ |
| $for$ $j$ $in$ $\{0, 1, ..., H - 1\}$ | $for$ $j$ $in$ $\{0, 1, ..., H - 1\}$ |
| $for$ $k$ $in$ $\{0, 1, ..., W - 1\}$ | $for$ $k$ $in$ $\{0, 1, ..., W - 1\}$ |
| $avg_{i,j,k}^s = mean(W_{(i,j,k,:})$ | $end$ |
| $end$ | $end$ |
| $att^s = sigmoid(fc^s(avg^s))$ | $W^a = [\cdot]_{K \cdot C \cdot H \cdot W}$ |
| $for$ $i$ $in$ $\{0, 1, ..., K - 1\}$ | $for$ $i$ $in$ $\{0, 1, ..., K - 1\}$ |
| $for$ $j$ $in$ $\{0, 1, ..., C - 1\}$ | $for$ $j$ $in$ $\{0, 1, ..., C - 1\}$ |
| $for$ $k$ $in$ $\{0, 1, ..., H - 1\}$ | $for$ $k$ $in$ $\{0, 1, ..., H - 1\}$ |
| $for$ $l$ $in$ $\{0, 1, ..., W - 1\}$ | $for$ $l$ $in$ $\{0, 1, ..., W - 1\}$ |
| $W_{(i,j,k,l)}^a = W_{(i,j,k,l)} \cdot att_{(i,j)}^c \cdot att_{(i,k,l)}^s$ | $end$ |
| $end$ | $end$ |
| $end$ | $end$ |

3.4. Overall Procedure

Table 1 is the training procedure described above.

When deploying, we only use $W^a$ instead of $W$ for calculation, as the shape and channel weights are in the structure and no need for re-calculation.

4. Experiments

4.1. ResNet and CIFAR-10

Deep residual network (ResNet) [11] is a commonly used network recently, it argued that residual connections are of inherent importance for training very deep architectures. We select this network as a base model for evaluation.

CIFAR-10 is a commonly used dataset which consists of 50k training images and 10k testing images in 10 classes. We present experiments trained on the training set and evaluated on the test set of CIFAR-10.

Table 2. CIFAR 10 validation accuracy results with different settings of ResNet.

| ResNet layers | Validation Accuracy |
|---------------|---------------------|
| 20            | Original 91.25      |
|               | Self-Attentional 92.78 |
We deploy the same settings as original ResNet, in which a weight decay of 0.0001 and momentum of 0.9 are used, and adopt the weight initialization of MSRA [12], use Batch Normalization [13] but no dropout. These models are trained with a mini-batch size of 128 on two GPUs. We start with a learning rate of 0.1, divide it by 10 at 32k and 48k iterations, and terminate training at 64k iterations.

We follow the same simple data augmentation for training: 4 pixels are padded on each side, and a 32×32 crop is randomly sampled from the padded image or its horizontal flip. For testing, we only evaluate the single view of the original 32×32 image.

As in Table 2, we can see that all the settings are improved by about 1%.

### 4.2. AlexNet and ImageNet

We evaluate our method on the ImageNet 2012 dataset which consists of 1000 classes. The models are trained on the 1.28 million training images, and evaluated on 50k validation images. The top-1 accuracy rates is in consideration of the evaluating.

For the base model, we choose AlexNet [1]. All the data preparation and training strategies are the same as the original paper, except for the learning rate of the parameters in self-attentional convolutions, we set it as 1e-4 for all the iterations.

Because of the huge parameters in this network, we use bottleneck structures to reduce the memory usage, in which we use two fully-connected layers instead of the model used in CIFAR-10 dataset, because the intermediate bottleneck is much smaller than the original size, these two fully-connected layers have less amount of computation than one.

Table 3 shows the parameter setting details of the convolutional layers.

We achieve a 57.9% validation top-1 accuracy in this setting, which original AlexNet is 57.1%.

### 5. Conclusion

In this paper, the main contribution is provide a method for constraining the convolutional layers by calculating scale factors for the weights, which adjusts the importance of each elements in channel-wise and shape-wise, weaken the effect of the noise, and improves the extensiveness. The improvement is about 1% on 20, 32, 44, 56 and 110 layers of ResNet on CIFAR-10, and 0.8% of AlexNet on ImageNet dataset.

Table 3. The modified structure of AlexNet. For layers of Conv2~Conv5, we replace the convolution by our Self-Attention Convolution layer, with a bottleneck factor.

| Convolution layer | Type               | Bottleneck factor |
|-------------------|--------------------|-------------------|
| Conv1             | Conv               | -                 |
| Conv2             | Self-Attentional Conv | 1/16              |
| Conv3             | Self-Attentional Conv | 1/32              |
| Conv4             | Self-Attentional Conv | 1/64              |
| Conv5             | Self-Attentional Conv | 1/32              |

In the future, we will explore and research the weights distribution of Self-Attention Convolutional layers, find a further way for improvement.
Acknowledgements
This work is supported by the NSFC 61672089, 61273274, 61572064, and National Key Technology R&D Program of China 2012BAH01F03.

References
[1] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In Advances in neural information processing systems (pp. 1097-1105).
[2] S. Han, H. Mao, W. J. Dally (2016). Deep Compression: Compressing Deep Neural Networks with Pruning, Trained Quantization and Huffman Coding, International Conference on Learning Representations.
[3] S. Han, J. Pool, S. Narang, H. Mao, S. Tang, E. Elsen, B. Catanzaro, J. Tran, W. J. Dally (2017) DSD: Regularizing Deep Neural Networks with Dense-Sparse-Dense Training Flow, International Conference on Learning Representations.
[4] Guo, Y., Yao, A., & Chen, Y. (2016). Dynamic network surgery for efficient dnns. In Advances In Neural Information Processing Systems (pp. 1379-1387).
[5] Liu, Z., Li, J., Shen, Z., Huang, G., Yan, S., & Zhang, C. (2017). Learning efficient convolutional networks through network slimming. In Computer Vision (ICCV), 2017 IEEE International Conference on (pp. 2755-2763).
[6] Wen, W., Wu, C., Wang, Y., Chen, Y., & Li, H. (2016). Learning structured sparsity in deep neural networks. In Advances in Neural Information Processing Systems (pp. 2074-2082).
[7] Lu, J., Xiong, C., Parikh, D., & Socher, R. (2017, July). Knowing when to look: Adaptive attention via a visual sentinel for image captioning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (Vol. 6, p. 2).
[8] Yang, Z., He, X., Gao, J., Deng, L., & Smola, A. (2016). Stacked attention networks for image question answering. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 21-29).
[9] Lee, C. Y., & Osindero, S. (2016). Recursive recurrent nets with attention modeling for ocr in the wild. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 2231-2239).
[10] Wang, F., Jiang, M., Qian, C., Yang, S., Li, C., Zhang, H. & Tang, X. (2017). Residual Attention Network for Image Classification. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 3156-3164).
[11] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778).
[12] He, K., Zhang, X., Ren, S., & Sun, J. (2015). Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In Proceedings of the IEEE international conference on computer vision (pp. 1026-1034).
[13] Ioffe, S. and Szegedy, C. (2015) Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. Proceedings of the 32nd International Conference on Machine Learning.