A Deep Analysis of Grouped Convolution Schemes for Improving Deep Learning Performance

Jianhao Gong, Hengyi Li, Qi Li, and Lin Meng

Dept.of Electronic and Computer Engineering, Ritsumeikan University,
Kusatsu, Shiga, Japan,
{gr0468kx@ed,menglin@fc.}ritsumei.ac.jp

Abstract

With the widespread utilization of deep learning, reducing the model parameters and the inference time for applying various hardware environments becomes an important issue. This paper aims to improve deep learning performance by reducing model parameters and inference time through a deep analysis of the grouped convolution. The technique is equipped on VGG model which is one of the major deep learning architecture and extended to new deep learning models. The experimental results show that the inference time is reduced to 55% only with slight accuracy deterioration.

1 Introduction

In modern society, deep learning has been widely used in various areas and makes our daily living more convenient. Since the emergence of AlexNet proposed by Alex Krizhevsky in 2012[1], deep learning develops rapidly and has achieved great successes. Multiple excellent deep learning architectures have been designed, such as VGG, ResNet-50[2], Inception-v3 [3] and so on. These deep neural networks (DNNs) also have been applied in various domains, such as data analysis and mining, pattern recognition, bioinformatics, voice recognition, natural language processing, cultural heritage protection [4, 5], etc.

However, there are also crucial problems to be solved for DNNs: the huge size and computation intensity of DNNs make them a heavy burden to be deployed in lightweight devices which are resource-constrained. At the same, DNNs are also overparameterized with a large number of redundant computations [6]. Thus, the optimization and speed-up of DNNs become more and more important, and has been a major research area.

This paper aims to improve deep learning performance by reducing model parameters and inference time through a deep analysis of the grouped convolution. The study is equipped on VGGBN model[7], which is one of the major DNN architectures. The main contributions of this paper are as follows: First, Implementing the grouped convolution on the VGGBN model and making an obvious improvement on the performance; Second, Evaluating and comparing various kinds of grouped convolution schemes, making a thorough analysis of grouped convolution over DNNs.

The rest of the paper is organized as follows. Section 2 introduces the related work. Section 3 details our research proposal and section 4 shows our schemes, process of our experiment and data from the experiment. The conclusion is reached in section5.

2 Related Works

Grouped convolution is first applied in AlexNet[1]to distribute the model into two GPUs in 2012 for the lack of memory of GTX580. Later, there has been a lot of research about grouped convolution.

ResNeXt [8]makes further efforts using grouped convolution to implement a set of transformations and certificating effectiveness of the model. A large number of researchers apply grouped convolution to design computation-efficient DNNs[9][10][11][12][13][14]. On the basis of grouped convolution, some researchers have also proposed depthwise convolution[9][11][15]. Depthwise convolution is a more extreme case of grouped convolution, which refers to a grouped convolution scheme with the number of groups equal to the quantity of input feature maps. In addition, researchers in Condensenet[16] proposed a learnable grouped convolution to automatically select the input channel of each group , which is far more efficient than modern convolutional networks such as ShuffleNet.

3 Preliminary

3.1 Research objective

VGGBN architecture is taken to make the research, which improves VGG architecture by introducing Batch Normalization [17] operation. The details of the network are shown in Table 1, and for simplicity, the experiments are taken on VGG13BN. VGG architecture is proposed by the Visual Geometry Group of Oxford University[7]. An improvement of VGG model compared with AlexNet is to use several smaller convolution kernels to replace the larger convolution kernels in AlexNet. To be specific, in VGG, three 3×3 convolution kernels are used to replace 7×7 convolution kernel, two 3×3 convolution kernels are used to replace 5×5 convolution kernel. VGG architecture reveals that the depth of CNN neural network which has several stacked kernel filters with small size is a significant factor for the network performance. For the reason that a given receptive field, using a stacked small convolution kernel is better than using a large convolution kernel. And in terms of multiple nonlinear layers with more depth of the network, it ensures the network with the ability of learning complex patterns. As for Batch Normalization[17], the method is proposed by Sergey Ioffe and Christian Szegedy mainly to improve training speed and prevent overfitting.

3.2 Dataset

In this paper, Dataset of Kuzushiji is taken to make the study. Kuzushiji is a dataset that consists of over 65000 labeled high-resolution images of ancient Asian characters, which are classified into 1120 categories. In the experiments, we divide the dataset into a training set, a validation set and a testing set in a ratio of about 16:4:5, of which there are 41770 pieces of data in the training set, 10586 pieces of data in the validation set and 13,439 pieces of data in the testing set. And the inputs are unified as RGB channels and reshaped with the centered [224×224] to maintain the size for the network.

Figure 1: Kuzushiji Dataset

Table 1: Architectures of VGG

layer name	output size	VGG13BN	VGG16BN	VGG19BN
conv-1	224×224	[3×3,64]×2	[3×3,64]×2	[3×3,64]×2
	112×112	2×2 maxpool,stride 2		
conv-2	112×112	[3×3,128]×2	[3×3,128]×2	[3×3,128]×2
	56×56	2×2 maxpool,stride 2		
conv-3	56×56	[3×3,256]×2	[3×3,256]×3	[3×3,256]×4
	28×28	2×2 maxpool,stride 2		
conv-4	28×28	[3×3,512]×2	[3×3,512]×3	[3×3,512]×4
	14×14	2×2 maxpool,stride 2		
conv-5	14×14	[3×3,512]×2	[3×3,512]×3	[3×3,512]×4
	7×7	2×2 maxpool,stride 2		
	7×7	AdaptiveAvgPool		
	1×1	FC-4096		
	1×1	ReLU,DropoutFC-4096		
	1×1	ReLU,DropoutFC-4096		

[1] $[x \times x, z]$:convolution kernel size $x \times x$, y channels.
[2] Each convolution layer follows with a BN and ReLU layer.
[3] Output size: $width \times height$

3.3 Grouped Convolution

The grouped convolution means a group of convolutions for the layer, with multiple groups of kernels and corresponding multiple groups of output channels. The method was originally proposed by AlexNet[1] to distribute the model on two GPUs for the lack of graphics card memory. And then, the model MobileNets[9] proves that grouped convolution can reduce the parameters of the neural network.

The diagram of grouped convolution is shown in Fig.2. The figure a shows the original convolution with eight output channels. And the right figure b convolution exhibits the grouped convolution with four groups of eight output channels.

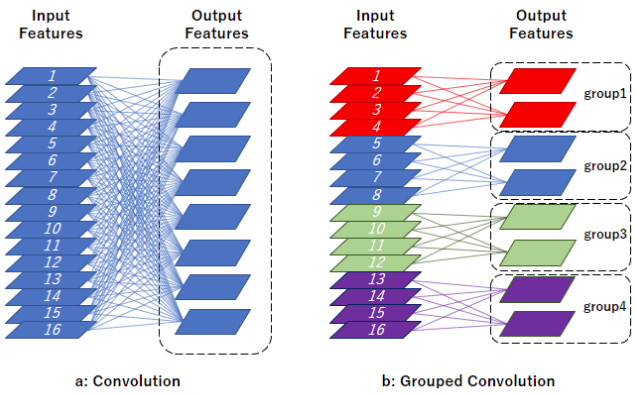

Figure 2: grouped convolution

3.4 Training Algorithm

SGD is taken as the optimization algorithms for training phase in the experiments. The stochastic gradient descent (SGD)[18] has become one of the most commonly used training algorithms for deep neural networks. Although SGD is simple, it performs well in a variety of applications and has a strong theoretical foundation.

4 Research Proposal

In terms of grouped convolution, to make a thorough analysis of the method, We take different kinds of schemes shown as follows to make experiments.

- Scheme A: convolutional layers of conv-3 that with the output channels of 256 are grouped into 64 groups.
- Scheme B: convolutional layers of conv-4 and conv-5 that with the output channel of 512 are grouped into 64 groups.
- Scheme C: convolutional layers of conv-3, conv-4 and conv-5 with the output channel of 256 and 512 respectively are grouped into 64 groups.
- Scheme D: convolutional layers of conv-2, conv-3, conv-4 and conv-5 with the output channel of 128, 256 and 512 respectively are grouped into 64 groups.
- Scheme E: convolutional layers of conv-2, conv-3, conv-4 and conv-5 with the output channel of 128, 256 and 512 respectively are grouped into 32 groups.

5 Experiment

5.1 Experimental Results

In the process of training, we set the batch size to be 20 and take the epoch of 25. The experimental results are shown in the following figures, and the baseline corresponds to the original model.

Fig.3-Fig.8 show the accuracy curve and loss curve of training and validation during the training process. The training loss for the baseline is 0.064, and the values for grouped convolutions vary from 0.062 to 0.086. The validation loss of the baseline is 1.139, and the values for grouped convolutions vary from 1.199 to 1.4. It can be seen that the grouped convolution has little impact on the training loss and validation loss compared with the baseline. In terms of the training accuracy, the grouped convolutions have little change compared with the baseline. As for the validation accuracy, the grouped convolutions have a little more decrease. Such as the validation accuracy and test accuracy for scheme E decreases by 4.04% and 2.49% respectively compared with the baseline. Fig.9 shows the test accuracy of the baseline and the grouped convolution network. The test accuracy follows the same laws as the validation accuracy. However, the decrease is still within the acceptable range.

Fig.10 shows the Macs and the inference time for processing one input of the model. As can be seen, with acceptable loss of accuracy, the two indexes the models which adopt grouped convolution have a significant decrease compared with that of the original network. For example, the Macs of scheme E decreased by about 80% compared with the baseline. The inference time of scheme E is decreased by 54.73%.

5.2 Loss and accuracy

As can be seen from Fig.3-Fig.8, the grouped convolution has little impact on the performance of the model. Even for scheme D, the test accuracy decreases only by 3.03% compared with the baseline.

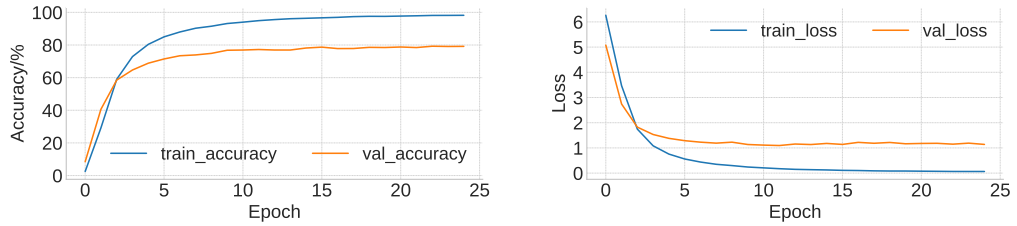

Figure 3: Baseline

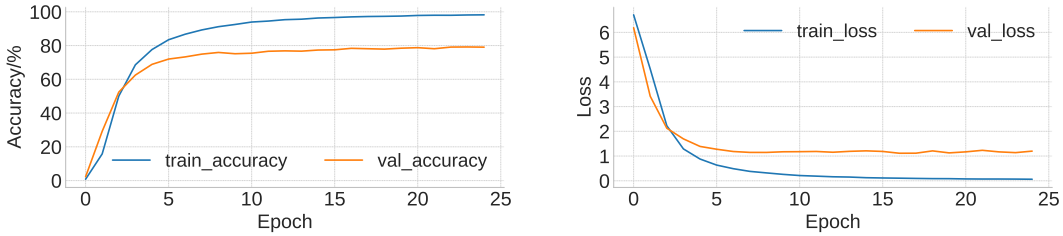

Figure 4: Scheme A

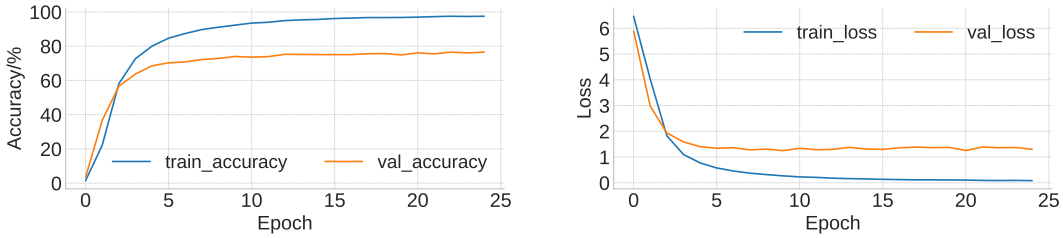

Figure 5: Scheme B

In this section, we take the multiply-accumulate operations (Macs) and the inference time to make the evaluation. According to Fig.10, it can be seen that the multiply-accumulate operations(Macs) of the models which adopt grouped convolution are lower than that of the original network. For example, the Macs of scheme E decreased by about 80% compared with the baseline. The inference time of scheme E is also significantly decreased by 54.73%. From the above data, the more layers that convolved with groups the model has, the faster its training speed is, and the lower the amount of parameters and Macs is.

easychair: Running title head is undefined.

easychair: Running author head is undefined.

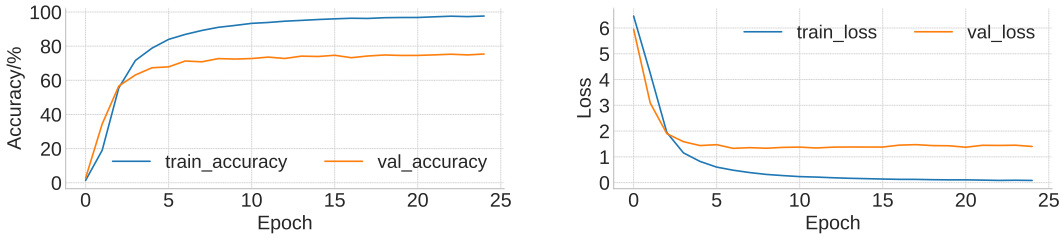

Figure 6: Scheme C

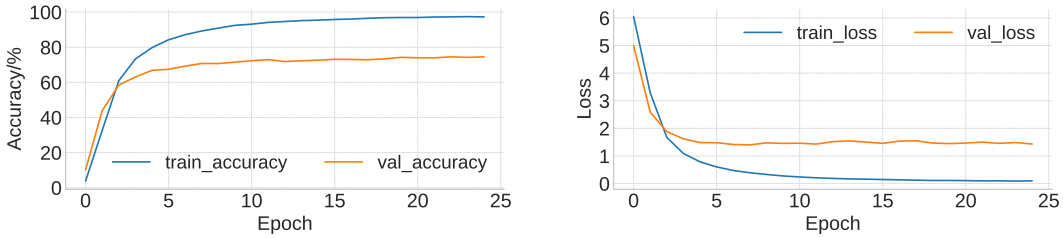

Figure 7: Scheme D

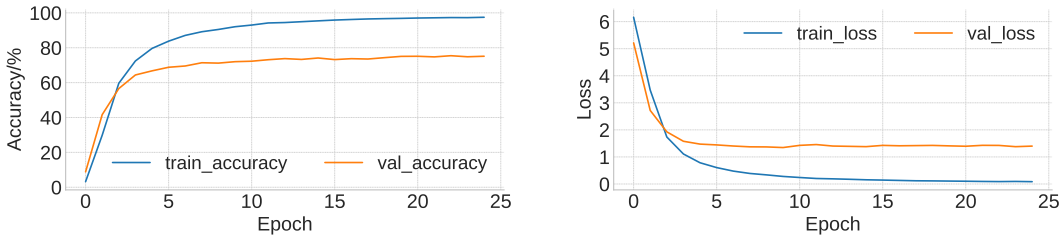

Figure 8: Scheme E

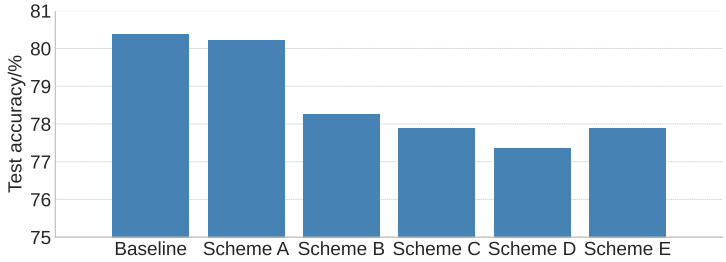

Figure 9: Test accuracy

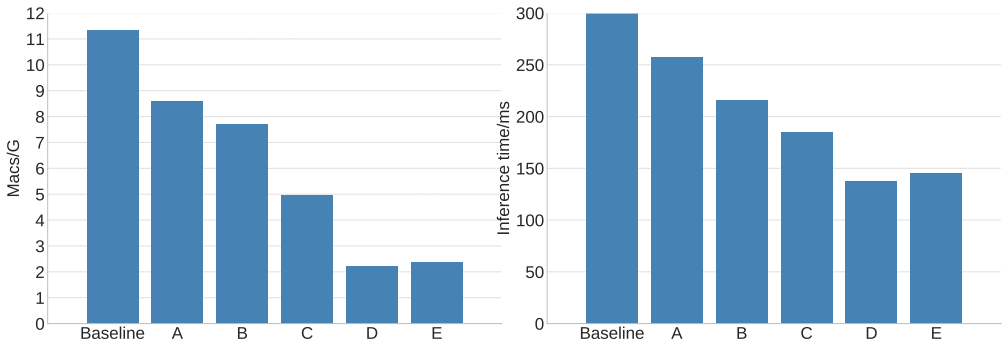

Figure 10: Macs and Inference time

6 Conclusion

The paper makes an in-depth study on the grouped convolution and proves its effectiveness. The grouped convolution makes the input layer into groups and convolves them separately. By this way, the method can greatly reduce the Macs by up to 80% and decrease the inference time by up to 50% for neural network with a little impact on the accuracy. According to the research, conclusions can be drawn that the more grouped convolution makes more decrease on the Macs and the inference time of the model, however, along with relatively more loss on the accuracy. In practice, we need to weigh these factors to make the best scheme. In the follow-up research, we plan to make further study on more datasets such as ImageNet and CIFAR100 to analysis the difference among various datasets. And combining the shuffle operation with the grouped convolution to make further improvements for neural networks by exploiting the potential of the cooperation, with expectations to achieve a more simplified neural network with high accuracy.

References

[1] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25:1097–1105, 2012.

[2] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[3] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016.

[4] Lyu Bing, Hiroyuki Tomiyama, and Lin Meng. Frame detection and text line segmentation for early japanese books understanding. In *Proceedings of the 9th International Conference on Pattern Recognition Applications and Methods - ICPRAM,*, pages 600–606. INSTICC, SciTePress, 2020.

[5] Lin Meng, Bing Lyu, Zhiyu Zhang, C.V.Aravinda, Naoto Kamitoku, and Katsuhiro Yamazaki. Oracle bone inscription detector based on ssd. *ICIAP2019*, pages 126–136, 2019.

[6] Hengyi Li, Zhichen Wang, Xuebin Yue, Wenwen Wang, Hiroyuki Tomiyama, and Lin Meng. A comprehensive analysis of low-impact computations in deep learning workloads. In *in Proceedings of the Great Lakes Symposium on VLSI 2021 (the 31st GLSVLSI)*, 2021.

[7] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[8] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1492–1500, 2017.

[9] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

[10] Ningning Ma, Xiangyu Zhang, Hai-Tao Zheng, and Jian Sun. Shufflenet v2: Practical guidelines for efficient cnn architecture design. In *Proceedings of the European conference on computer vision (ECCV)*, pages 116–131, 2018.

[11] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4510–4520, 2018.

[12] Ke Sun, Mingjie Li, Dong Liu, and Jingdong Wang. Igcv3: Interleaved low-rank group convolutions for efficient deep neural networks. *arXiv preprint arXiv:1806.00178*, 2018.

[13] Ting Zhang, Guo-Jun Qi, Bin Xiao, and Jingdong Wang. Interleaved group convolutions. In *Proceedings of the IEEE international conference on computer vision*, pages 4373–4382, 2017.

[14] Xiangyu Zhang, Xinyu Zhou, Mengxiao Lin, and Jian Sun. Shufflenet: An extremely efficient convolutional neural network for mobile devices. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6848–6856, 2018.

[15] François Chollet. Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1251–1258, 2017.

[16] Gao Huang, Shichen Liu, Laurens Van der Maaten, and Kilian Q Weinberger. Condensenet: An efficient densenet using learned group convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2752–2761, 2018.

[17] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. PMLR, 2015.

[18] Herbert Robbins and Sutton Monro. A stochastic approximation method. *The annals of mathematical statistics*, pages 400–407, 1951.