Comparing Different Ways of Convolution and Normalization to Improve the Performance of Neural Networks

Shujie Deng
University of Toronto
shujie.deng@mail.utoronto.ca

Lingfei Li
University of Toronto
lingfeiii.li@mail.utoronto.ca

Saifei Liao
University of Toronto
saifei.liao@mail.utoronto.ca

Abstract

Deep learning has developed rapidly in recent years, allowing us to see the emergence of many neural nets with astonishing performances. In this paper, we compared the standard convolution structure with separable convolution, batch normalization with group normalization and conducted experiments with AlexNet and VGG-16. Furthermore, we added a BatchNorm layer within separable convolution to evaluate the performance. Empirically, we show that separable convolution perform better than standard convolution, and group normalization perform worse than batch normalization with large batch size.

1 Introduction

Convolution neural network is a most commonly applied architecture in deep learning. In this paper, we studied ways of improving the performance of Convolutional Neural Networks from two perspectives. On one hand, we consider different ways of normalization between layers. On the other hand, we use Separable convolution instead of standard convolution in our models. We first reproduce the experimental results by reconstructing the architecture of AlexNet and VGG16, and perform sensitivity analysis on hyper parameters. Then, we test these two architectures on a new task domain, namely, image classification, and evaluate the performances on CIFAR-10 and STL-10.

2 Related work

To improve convolution layers, researchers tried methods from changing structures of convolution to developing different normalization methods. AdderNet[1] uses l_1 norm between inputs and filters to reduce computational complexity, and Kervolutional neural networks[2] introduces a non-linear kernel to perform convolutions with higher expressivity. However, it is difficult for these architectures to fit into GPU running since GPU are optimized on the assumption of multiplication. Inception Model[3] introduces a 1×1 convolution layer to solve the computational bottleneck.

To improve normalization, Ba et al[4]., proposed layer normalization, which normalizes across channels rather than batches, to improve the training speed and overcome the difficulty of applying normalization to models with small batch sizes. Taking one step further, Lempitsky et al[5], introduced Instance Normalization to reduce computation costs. This further reduces the size to

Preprint. Under review.

perform normalization and puts constraints on the normalization quality. The following are the algorithms for Layer normalization and Instance Normalization:

Layer Normalization

$$\mu^{l}=\frac{1}{H}\sum_{i=1}^{H}a_{i}^{l},\sigma^{l}=\sqrt{\frac{1}{H}\sum_{i=1}^{H}\left(a_{i}^{l}-\mu^{l}\right)^{2}}$$

where H is the number of hidden units in a layer.

Instance Normalization

$$y_{tijk}=\frac{x_{tijk}-\mu_{ti}}{\sqrt{\sigma_{ti}^{2}+\epsilon}},\quad \mu_{ti}=\frac{1}{HW}\sum_{l=1}^{W}\sum_{m=1}^{H}x_{tilm},\quad \sigma_{ti}^{2}=\frac{1}{HW}\sum_{l=1}^{W}\sum_{m=1}^{H}\left(x_{tilm}-mu_{ti}\right)^{2}$$

where H and W corresponds to the width and height within a layer.

3 Method

Inspired by the architecture of Xception[6], Batch Normalization[7], Group Normalization[8], we construct the following 4 structures to test the performance of models with different convolution and normalization.

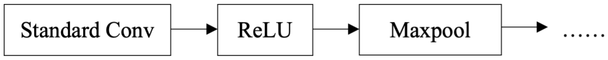

Figure 1: Standard Benchmark

3.1 Structure 1

The standard convolution process is introduced in Figure 1. In Structure 1, we replaced standard convolution with Separable convolution before ReLU, and a BatchNorm layer is added after the activation layer.

Separable convolution is first introduced in Xception and it aims to reduce the computational cost of the convolution layer by using Depthwise convolution followed by Pointwise Convolution.

Depthwise Convolution

The spatial convolution is performed on each channel of the input.

Pointwise Convolution

Use 1×1 convolution to perform the calculation on the output of the depthwise convolution and output to the new channel space.

Batch Norm:

The BatchNorm takes the output from the previously hidden layer and performs normalization process before passing the output to the next hidden layer. This helps to reduce the internal covariate shift and allows the gradient descent to converge better.

$$\mu_{\mathcal{B}}=\frac{1}{m}\sum_{i=1}^{m}x_{i},\sigma_{\mathcal{B}}^{2}=\frac{1}{m}\sum_{i=1}^{m}\left(x_{i}-\mu_{\mathcal{B}}\right)^{2},\hat{x}_{i}=\frac{x_{i}-\mu_{\mathcal{B}}}{\sqrt{\sigma_{\mathcal{B}}^{2}+\epsilon}},y_{i}=\gamma\hat{x}_{i}+\beta=\mathrm{BN}_{\gamma,\beta}\left(x_{i}\right)$$

where m is the batch size, γ and β are learnable parameters.

3.2 Structure 2

Compared to Structure 1, in Structure 2, BatchNorm layer is replaced with Group Normalization Layer, introduced by He et al. Group Normalization groups different features together, and performs normalization across groups rather than batches.

Group Norm:

$$\mu = \frac{1}{UG}\sum_{i=1}^{G}\sum_{j=1}^{F} x_{ij}, \sigma 2 = \frac{1}{UG}\sum_{i=1}^{U}\sum_{j=1}^{H}(x_{ij} - \mu)^2$$

where U denotes the number of units per feature, and G denotes the number of features in a group.

3.3 Structure 3/4

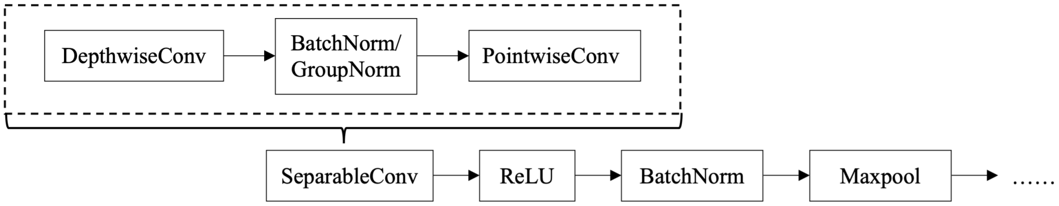

Figure 2: Structure 3/4: Structure 2 with additional BatchNorm/GroupNorm after Depthwise Convolution

We add an additional BatchNorm/GroupNorm Layer within the Separable Convolution Layer after the Depthwise Convolution Layer. The intention is to stabilize optimization within Separable Convolution.

4 Experiments and Discussions

The datasets to tested on are CIFAR-10 and STL-10. STL-10 is an image recognition dataset inspired by CIFAR-10, which has fewer labelled training examples but higher resolution images. The main two architectures used in our experiments are AlexNet and VGG-16. Our experiments can be divided into 3 categories: comparison of convolution layer (Benchmark and Structure 1), comparison of normalization(Structure 1 and Structure 2), and comparison of normalization within SeparableConv2D(Structure 3 and 4).

4.1 Comparison of Convolution Layer(Benchmark and Structure 1)

Table 1: Comparison of Standard Conv2D and SeparableConv2D

	Model	AlexNet	AlexNet-1*	VGG16	VGG16-1*
CIFAR-10	Accuracy(%)	84.15	85.19	80.93	82.78
	Training Loss	0.0235	0.0088	0.0704	0.0362
	Test Loss	0.8009	0.7514	0.8723	0.6998
STL-10	Accuracy(%)	57.23	67.17	46.11	59.01
	Training Loss	0.9653	0.0135	1.2378	0.0837
	Test Loss	1.1782	1.3903	1.4453	1.4274

*Note that AlexNet-1 is AlexNet with separable convolution layers + BatchNorm and VGG16-1 is VGG-16 with separable convolution layers + BatchNorm.

In this section, we replaced the standard Conv2D layers in AlexNet and VGG16 with separable Conv2D layers. Overall, the test accuracy of *Alex-Net-1* is higher than *Alex-Net* on both datasets and on both models. It is also worth mentioning that not only the training loss but also the test loss is reduced with the implementation of separable convolution layers and bath norm layers.
The improvement can be attributed to 2 reasons. First and foremost, separate filters are applied to separate channels, so the cross-channel correlation are decoupled. Second, the number of trainable

parameters of *Alex-Net-1* is larger than that of *Alex-Net*, which contributed to greater expressivity. In this way, we can gain more information after the training process, resulting in better performance.

4.2 Comparison of Normalization(Structure 1 and Structure 2)

Table 2: Comparison of BatchNorm and GroupNorm

	Model	AlexNet-1	AlexNet-2*	VGG16-1	VGG16-2*
CIFAR-10	Accuracy(%)	85.19	84.41	82.78	83.27
	Training Loss	0.0088	0.0042	0.0362	0.0179
	Test Loss	0.7514	0.7380	0.6998	0.7058
STL-10	Accuracy(%)	67.17	64.49	59.01	58.65
	Training Loss	0.0135	0.0087	0.0837	0.1206
	Test Loss	1.3903	1.4621	1.4274	1.3883

*Note that AlexNet-2 is AlexNet with separable convolution layers + GroupNorm and VGG16-2 is VGG-16 with separable convolution layers + GroupNorm.

In this section, we used different ways of normalization, namely batch normalization and group normalization, on the two models, *AlexNet-1* and *VGG16-1*. From Table 2 above, we observe that accuracy on both datasets are slightly lower when we use group normalization.
Small batch size might lead to inaccurate estimation of the batch statistics, that is, mini-batch mean and variance. Since our batch size is 64, which is a relative large number, the model error will be smaller and thus the model will perform better compared to the one with group normalization.

4.3 Comparison of Normalization within SeparableConv2D(Structure 3 and 4)

Table 3: Comparison of BatchNorm and GroupNorm within SeparableConv2D

	Model	AlexNet-1	AlexNet-1-1*	AlexNet-1-2*
STL-10	Accuracy(%)	67.17	66.77	64.58
	Training Loss	0.0135	0.0448	0.0129
	Test Loss	1.3903	1.1953	1.3009

*Note that AlexNet-1-1 is AlexNet-1 applying batch normalization after DepthwiseConv2D within SeparableConv2D and AlexNet-1-2 is AlexNet-1 applying group normalization.

In this section, we added a BatchNorm and a GroupNorm layer within SeparableConv2D layers. The accuracy of *AlexNet-1* is a little bit higher than both modified models. This is because separable convolution aims to uncouple features of different channels, but the addition of normalization re-adds the coupling between channels. Furthermore, since the architecture already adds normalization, adding a second convolution does not provide additional functions.

5 Conclusion and Limitations

In this paper, we proposed several convolution structures using Separable Convolution, Batch Normalization and Group Normalization and conducted several experiments to compare the performance of the AlexNet and VGG-16 on different datasets. In conclusion, we discover that there is a significant improvement using Structure 1; Structure 1 performs better than Structure 2; and Structure 3/4 performs slightly worse than the benchmark.
There are several limitations in our experiments. First of all, we used the same batch size throughout the whole experiement. We may try different batch sizes to see the effects on the nomalization layers. Moreover, we did not tune the in_features and out_features in the convolution layer for AlexNet and VGG16. We may consider tuning the hyperparameters. Lastly, CIFAR-10 may not be so representative for AlexNet since AlexNet performs better on larger dataset such as ImageNet. The conclusion will be more convincing if we use diverse datasets.

6 Contribution

- Implementation of AlexNet and VGG16: Saifei Liao, Shujie Deng
- Implementation of AlexNet-1, AlexNet-2, VGG16-1 and VGG16-2: Saifei Liao
- Implementation of AlexNet-1-1 and AlexNet-1-2: Shujie Deng, Lingfei Li
- Training and Visualization: Lingfei Li
- Report: Shujie Deng, Lingfei Li, Saifei Liao

References

[1] Chen, H., Wang, Y., Xu, C., Shi, B., Xu, C., Tian, Q., & Xu, C. (2020). AdderNet: Do we really need multiplications in deep learning?. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 1468-1477).

[2] Wang, C., Yang, J., Xie, L., Yuan, J. (2019). Kervolutional neural networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 31-40).

[3] Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z. (2016). Rethinking the inception architecture for computer vision. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 2818-2826).

[4]Ba, J. L., Kiros, J. R., Hinton, G. E. (2016). Layer normalization. arXiv preprint arXiv:1607.06450.

[5]Ulyanov, D., Vedaldi, A., Lempitsky, V. (2016). Instance normalization: The missing ingredient for fast stylization. arXiv preprint arXiv:1607.08022.

[6] Chollet, F. (2017). Xception: Deep learning with depthwise separable convolutions. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 1251-1258).

[7] Ioffe, S., Szegedy, C. (2015, June). Batch normalization: Accelerating deep network training by reducing internal covariate shift. In International conference on machine learning (pp. 448-456). PMLR.

[8] Wu, Y., He, K. (2018). Group normalization. In Proceedings of the European conference on computer vision (ECCV) (pp. 3-19).

Appendix

Below are some plots from our experiments.

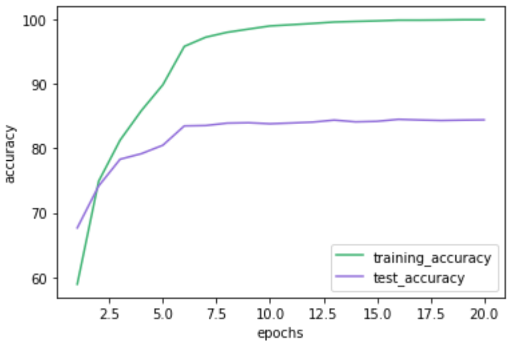

Figure 3: Accuracy of AlexNet with Seperable Convolution and GroupNorm on CIFAR-10

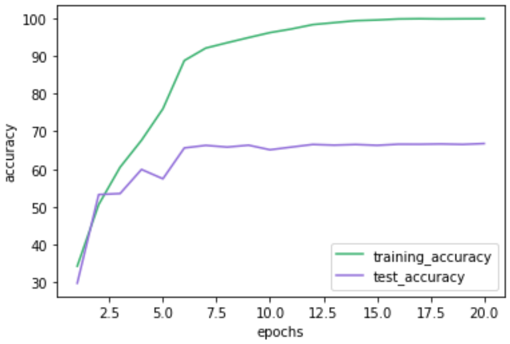

Figure 4: Accuracy of AlexNet with Structure 3 and GroupNorm on CIFAR-10

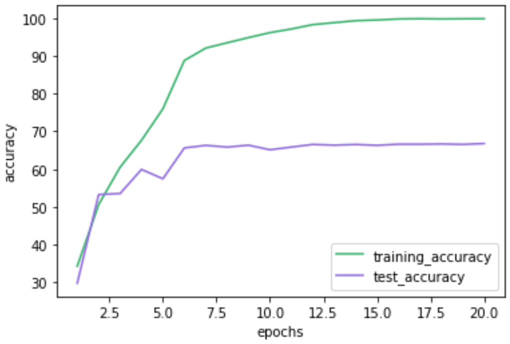

Figure 5: AlexNet with Structure 3 on STL-10

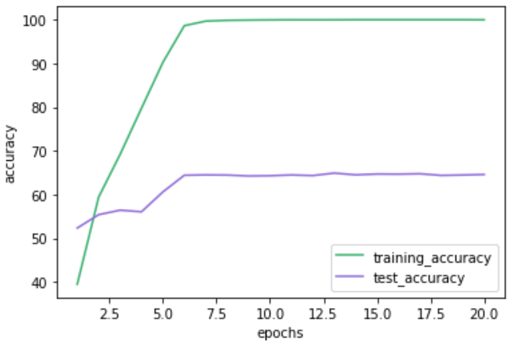

Figure 6: AlexNet with Structure 4 on STL-10