REPRODUCIBILITY REPORT / NETWORK DECONVOLUTION

Runying Jiang, Xinhao Wu, Liang Zhu and Yuanlong Zhang
University of Southampton
{rj1u20,xw6g20,lz3g20,yz6n20}@soton.ac.uk

ABSTRACT

The main purpose of this project is to reproduce the experiment described in the original paper, verifying whether the content of that experiment can be reproduced completely and exploring the validity of the results and the conclusions of the paper. In addition, we have extended the research appropriately by reading the relevant articles.

1 INTRODUCTION

Convolution is the core component of the Convolutional Neural Networks(CNN), which applies the small local function (also called ”kernel”) to the overlapping regions moving on the image. However, in practical real-world applications image data is highly correlated with each other, so the convolution kernel actually makes a lot of redundant data to learn. The original paper(Ye et al., 2020) proposes network deconvolution, a deconvolution method that removes channel correlation from each layer of the pixel network layer by layer. In this report, we will explore and analyse the five main sections of: ”Overview of the target problems in research”, ”Parsing the Network deconvolution method of the original article”, ”Performing experiments and analysis ”, ”Reflection and Discussion”, and ”Conclusion”.

2 TARGET QUESTIONS

We raised the following questions to verify whether the conclusions of the article can be supported: **1)** Does network deconvolution achieve a better accuracy compared with batch normalization? If so, in what ways can BN be completely replaced? **2)** Does network deconvolution lead to a faster convergence, especially the faster decay in training loss? **3)**Is the training speed of network deconvolution similar to that of batch normalization? **4)**To dig deeper, whether network deconvolution can outperform other cutting-edge technology and what are the strengths or weaknesses?

3 METHODOLOGY

Inspired by the centre-surround structure of the visual cortex, the original paper proposes the network deconvolution kernel. For each dataset, we trained the model with BN and DN separately, compare and analyse the performance.**Figure 1** describes the process of experimental methodology.

4 EXPERIMENTATION

The source code uploaded by authors can be found at: **`https://github.com/yechengxi/deconvolution`**. The original experiment in the paper contains classification and segmentation, considering the experimental equipment, we verified the conclusion on the classification with a small dataset instead of ImageNet(All of the models used in our experiment are trained with the following resources: AMD Ryzen 7 4800H 2.90GHz, 6GB NVIDIA GeForce RTX 2060, 16GB RAM). Our experiment is based on the authors’ code to compare the performance of the Batch Normalization(Ioffe & Szegedy, 2015) and Deconvolution(Ye et al., 2020) in CNNs with CIFAR-10

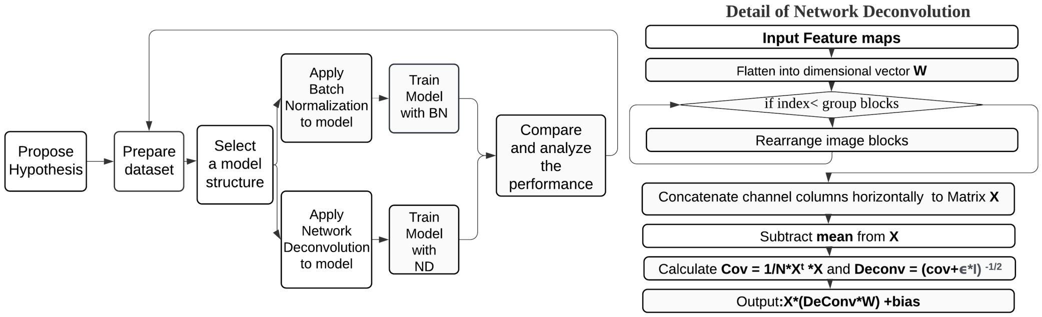
Propose Hypothesis
Prepare dataset
Select a model structure
Apply Batch Normalization to model
Apply Network Deconvolution to model
Train Model with BN
Train Model with ND
Compare and analyze the performance
Detail of Network Deconvolution
Input Feature maps
Flatten into dimensional vector W
if index< group blocks
Rearrange image blocks
Concatenate channel columns horizontally to Matrix X
Subtract mean from X
Calculate Cov = 1/N*X^t *X and Deconv = (cov+ϵ*I) ^-1/2
Output:X*(DeConv*W) +bias

Figure 1: The Flow Chart of methodology and detail of Network Deconvolution

and CIFAR-100. Initially, to simply verify the authors' conclusion, we implemented three simple CNNs(Conv2d, Conv2d with batch norm, Deconvolution) based on the deep learning lab on CIFAR-10(**Table 1**). We ran our structures using the FastDeconv code encapsulated by the authors. The batch size is 128, and the data was preprocessed by $torch.transforms$ using common parameters.

Table 1: Our implement to simply verify the paper's conclusions. The experimental results are in line with them of the paper. The deconvolution has a bit higher accuracy than batch normalization no matter the beginning or the end. Meanwhile, we also verified the important effect of batch normalization in CNNs. It should be noted that the deconvolution needs much more time to calculate.

Model	Accuracy epoch 1	Accuracy epoch 20
Simple CNN 1	26.71%	35.43%
Simple CNN 2	35.07%	38.76%
Simple CNN 3	36.22%	38.89%

4.1 USING MODERN CNN STRUCTURES

The authors provide specific interfaces to run several modern CNNs with deconvolution, according to our hardware condition, we selected suitable models to verify the experimental result from the paper(**Table 2**). According to the same conditions and dataset, although our results are numerically different from the authors', the improvement by deconvolution is consistent.

Table 2: Comparison on CIFAR-10/100 over 4 modern CNN models. They are trained for 1, 20 epochs with SGD(lr = 0.1) using BN and ND. And we also record the time consumed(seconds) by both(BN T and ND T). In general, using the latter can improve the accuracy, but with more computing time.

Model	CIFAR-10						CIFAR-100					
	BN 1	ND 1	BN 20	ND 20	BN T	ND T	BN 1	ND 1	BN 20	ND 20	BN T	ND T
EfficientNet-B0	36.21%	57.54%	84.32%	87.85%	23.5	23.8	11.09%	19.69%	55.41%	61.69%	23.6	24
MonileNet-V2	44.28%	54.21%	90.34%	92.23%	49.3	157	9.16%	19.07%	66.74%	71.3%	49.3	158
VGG-11	30.69%	72.57%	89.2%	91.42%	16.3	42.9	2.75%	36.02%	63.89%	70.36%	16.5	30.8
VGG-16	15.83%	70.95%	90.24%	93.1%	24.5	75.5	1.7%	31.33%	60.47%	72.08%	25.5	75.8

We found that the authors did not test the ResNet-50 which is widely used in the field of image classification, so after modifying the parameters to fit our hardware condition, we tested ResNet-50 and ResNext-29 with smaller batch size and only 10 epochs on CIFAR-10(**Table 3**).

Table 3: The network deconvolution can make improvements in complex CNN models, but it is very time-consuming. The deconvolution takes about twice and 6 times time in ResNet-50 and ResNext-29 respectively. The values in the table are the top1 accuracy.

Model	Batch Norm	Deconvolution	BN Time/s	ND Time/s
ResNet-50	81.93%	90.42%	137.5	261
ResNext-29	87.21%	92.26%	126	732

4.2 Other experimentation

So far, we have verified the ND can play the role of the BN, and by observing 1st epoch and checkpoints, the ND always has higher accuracy and lower loss at the beginning, meaning that its convergence is faster. We also test the influence of other conditions on the verification, such as different optimizers, loss function, and the number of blocks. The experimental baseline is that: using the SGD(lr=0.1) with cross-entropy in 20 epochs, and the number of blocks is 64 which is denoted by the authors as the default value. We want to test the performance of deconvolution by modifying one condition.

Table 4: The influence of different conditions in ND on CIFAR-10.

Model	SGD	Adam	CE	L2(MSE)	B 32	B 64	B 128
VGG-11	91.42%	79.5%	91.42%	90.9%	91.73%	91.42%	91.05%
VGG-16	93.1%	69.92%	93.1%	92.43%	92.9%	93.1%	93.22%
EfficientNet-B0	87.85%	33.21%	87.85%	87.22%	87.78%	87.85%	87.85%

B is the block size in deconvolution, and the authors pointed out B usually equal to 64 in the experiment. According to the verification on several CNN models, it will have a slight impact, so we consider it might be based on experimental experience to set as a middle value. Sometimes, increasing the size of B makes large computing cost but the improvement is very small. According to the experimental result, CE has a little higher accuracy than L2, which is denoted that CE and L2 have similar shapes in the source paper. In such 20 epochs, SGD with a momentum equals 0.9 is better than Adam.

5 Reflection and Discussion

To understand the Network Deconvolution, we first need to figure out where this idea came from. Batch Normalization(Ioffe & Szegedy, 2015), which is widely used in convolutional neural networks in recent years, exploits further by seeking to normalize not only the input to the first layer of the network but also the inputs to each internal layer in the network.

In various applications, different normalization techniques like Instance Normalization, Layer Normalization and group normalization. The difference is shown in **Figure 2**. The normalization techniques generally perform standardization that centres and scales features and the features are still correlated. Among different decorrelation methods, zero-component analysis whitening has drawn attention for its capability of preserving the original axis compared to PCA whitening which spins the axis to the principle direction. Then Decorrelated Batch Normalization(Huang et al., 2018), also known as Batch Whitening, was proposed. It whitens a mini-batch using its covariance matrix, which gives rise to better optimization efficiency than BN in image classification. The latest work is Switchable Whitening (Pan et al., 2019) that generalize different versions of Batch Normalization and Whitening. It shows that using only the whitening techniques performs comparably well compared with using all the normalization methods, indicating whitening can replace normalizations in many cases.

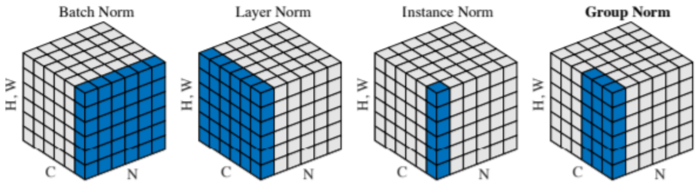

Figure 2: Normalization methods. Each subplot shows a feature map tensor, with N as the batch axis, C as the channel axis, and (H, W) as the spatial axes. The pixels in blue are normalized by the same mean and variance, computed by aggregating the values of these pixels. The Figure is taken with thanks from (Wu & He, 2018)

Network Deconvolution is similar to the Batch Whitening methods. Network deconvolution is mathematically equivalent to batch whitening using ZCA. The reason why it is called deconvolution is that the operation is indeed a generalized deconvolution operation that negates the effects of the convolution using kernel $k_{cov} = Cov^{0.5} \cdot \delta$. However, the Network Deconvolution is not the same as the existing batch whitening techniques. It has the following features:

- Feature 1: Network Deconvolution not only do channel-wise decorrelation but also take into consideration the spatial decorrelation between pixels.
- Feature 2: Network Deconvolution divides channels into groups and computes covariance in groups which introduces sparsity.
- Feature 3: There is no extra parameter to train for Network Deconvolution. Since it does not use parameters to control the extent of whitening adaptively like batch normalization, the capacity of the model is limited.
- Feature 4: It introduces some acceleration operations. Coupled Newton-Schulz iteration is numerically stable and the iteration number effectively controls the extent of whitening. Subsampling techniques handle large kernels. And the covariance matrix is frozen to be the running average after training and makes the testing faster.

In general, this paper makes a clear point of their idea and the experiments are well-organized. There are still some problems to solve: The paper says that when k=1 and B=1, it becomes Batch normalization. However, it is closer to batch whitening rather than normalization. Also, the method is only compared with Batch Normalization in the paper, more variants should be considered.

6 Conclusion

Overall, our experimental results are consistent with the authors' conclusion no matter in simple CNN or complex CNN. Network Deconvolution can at least achieve the effect of Batch Norm, and usually can improve the accuracy of the image classification tasks. Meanwhile, it can use a relatively large learning rate like BN, the ND has a higher convergence speed than BN as well. As the number of network layers increases and the structure becomes more complex, due to considering the pixel correlation(or covariance computation), the ND is much more time-consuming, although the authors implemented the so-called Fast Deconvolution. We noticed that there are still many normalization methods proposed after BN, the authors only compare ND and BN which is relatively narrow. In addition, after reading the relevant articles, our analysis concluded that the original authors were closer to batch whitening rather than normalization. If we can expand on this in the future, we hope to conduct a more in-depth comparative analysis with other variants.

References

Lei Huang, Dawei Yang, Bo Lang, and Jia Deng. Decorrelated batch normalization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 791–800, 2018.

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

Xingang Pan, Xiaohang Zhan, Jianping Shi, Xiaoou Tang, and Ping Luo. Switchable whitening for deep representation learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 1863–1871, 2019.

Yuxin Wu and Kaiming He. Group normalization. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 3–19, 2018.

Chengxi Ye, Matthew Evanusa, Hua He, Anton Mitrokhin, Tom Goldstein, James A. Yorke, Cornelia Fermuller, and Yiannis Aloimonos. Network deconvolution. In *International Conference on Learning Representations*, 2020.