ICLR REPRODUCIBILITY CHALLENGE / NETWORK DECONVOLUTION

Caterina Sbandati, Elliot Stein & Alex Wang
Department of Electronics and Computer Science
University of Southampton
Southampton, SO17 1BJ , United Kingdom
{cals1e20, es3e20, ahw1n20}@soton.ac.uk

ABSTRACT

This study was set out to reproduce the findings published in Network Deconvolution. Deconvolution operation is presented as an alternative to Batch Normalisation, a widely adopted operation in machine learning to improve performance. It is therefore critical that these findings are reproducible. This study provides supporting evidence for its reproducibility, as well as motivation for further work in assessing the validity of Network Deconvolution.

1 INTRODUCTION

This paper forms a submission to the International Conference on Learning Representations (ICLR) reproducibility challenge. The paper whose reproducibility is being assessed investigates Network Deconvolution (ND): an operation designed to replace Batch Normalisation (BN) (Ioffe & Szegedy (2015)) in deep networks which optimises the efficiency of the convolution operation. It does this by removing pixel and channel-wise correlations before the data is fed into the convolutional layer, so that the convolution operation is not wasting computations by learning redundant information. A number of pertinent claims are made in the paper which are addressed in this study. The authors have published code to GitHub, allowing detailed investigation into the reproducibility of their findings.

2 NETWORK DECONVOLUTION SUMMARY

The authors suggest that natural images captured by eyes and cameras alike, have a high statistical correlation between adjacent pixels as stated by Hyvärinen (2010). From this, they deduce that, in the same way it is possible to remove blur by deconvolving with a specific Gaussian blur filter, it may be possible to improve the image by deconvolving with some unknown filter as shown in Fig.1. Discovering precisely how this image would be improved, as well as finding the optimal deconvolutional kernel, were central aims of the paper. A core assumption made, but not explicitly stated, is that the blurring effect that we wish to remove through deconvolution is non-random. Information is necessarily lost during a random blur therefore cannot be restored by deconvolution. However, the promising results of this study suggest that this assumption holds.

Mathematical Basis: The convolution operation can be expressed directly as a matrix multiplication. The paper demonstrates this in Fig. 2. First, all the pixels in the input image that a given kernel parameter would be multiplied with in the convolution, is converted into a column through the `im2col` operation. For example, in a 3x3 kernel, the first parameter would be multiplied with all pixels except those in the final two columns and final two rows. This is repeated for all kernel parameters, and each column is horizontally concatenated, forming the data matrix $\mathbf{X}$. The kernel can be converted to a vector $\check{\mathbf{w}}$. Then, the vector $\mathbf{X} \cdot \check{\mathbf{w}}$ is exactly equivalent to the image(s) convolved with the kernel, after reshaping.

The deconvolution matrix, $\mathbf{D}$, is calculated by first calculating the covariance matrix of $\mathbf{X}$:

$$Cov(X) = \frac{1}{N}(\mathbf{X} - \mu)^T(\mathbf{X} - \mu) \tag{1}$$

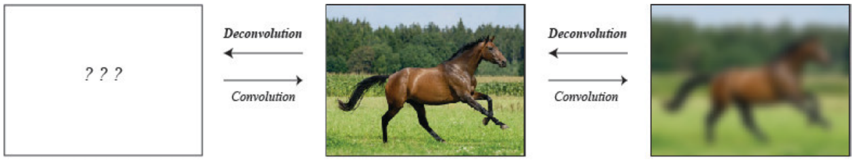


Figure 1: Image from original paper Ye et al. (2020). Visually demonstrates that deconvolution can improve image quality (from right to left) by removing local pixel correlations. The authors suggest that this can be extrapolated to further improve image quality beyond what is present in the original image. It is noteworthy that this does not hold generally for random noise or blur, in this case the right image is blurred because it is convolved with a Gaussian filter.

where N represents the number of samples and μ is the mean element of X. Then, approximating its inverse square root:

$$\mathbf{D} = Cov(X)^{-0.5} \tag{2}$$

Multiplying the data matrix **X** by this deconvolution matrix **D**, results in a new data matrix which has a covariance of the identity **I** (with small deviations due to approximation error in the calculation of the covariance matrix). This new data matrix can be multiplied by the kernel vector $\check{\mathbf{w}}$, to produce an output which benefits from the deconvolution.

While this is the principle behind the operation, many optimisation and approximation techniques are used to allow the deconvolution to run efficiently. All of this is taken into account in the experiments in this reproducibility study, as the same code is used where possible.

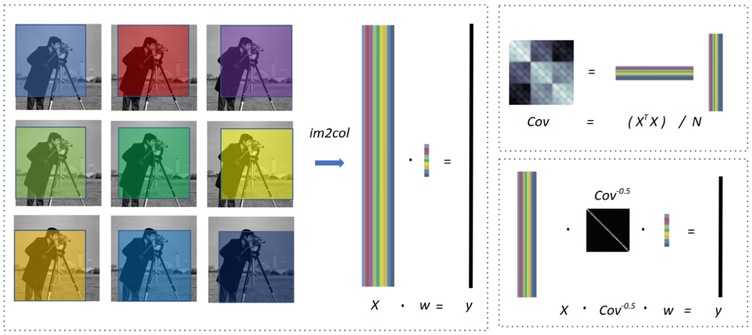


Figure 2: From original paper Ye et al. (2020). Demonstration of the construction of the deconvolution matrix and $D = Cov^{-0.5}$

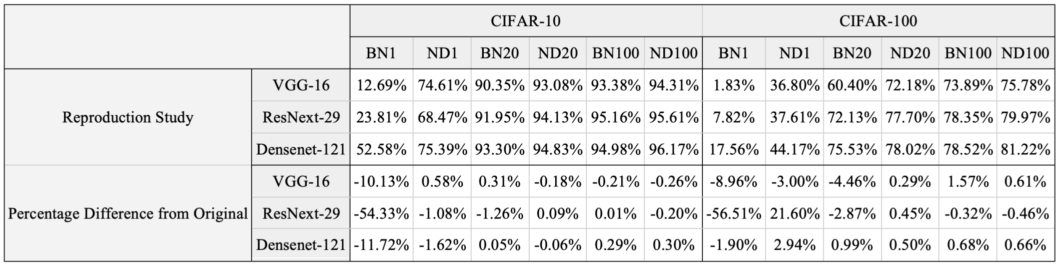
3 Experimental Details

In an effort to prove the superiority of the Deconvolution over widely adopted Batch Normalisation technique, the authors provide a thorough and extensive range of results. They tested the effect of replacing batch normalisation with Deconvolution operations on 10 model architectures on CIFAR-10 and CIFAR-100, three model architectures on ImageNet and one on Cityscape. Each is trained with 1, 20 or 100 epochs to assess convergence. This study focuses on reproducing the results attained on the median and top performing models on CIFAR-10 and CIFAR-100 datasets, to verify the claims made in the paper without requiring an unreasonable amount of computational time. Given the author's claims about the consistent appearance centre-surround structures in the RGB channel deconvolutional kernel, this study will also set out to verify this by inspecting randomly chosen kernels.

4 Results and Discussion

4.1 Experiment 1

The code published by the authors was mainly functional and capable of running on the University of Southampton High Performance Computing Cluster (IRIDIS - Nvidia Tesla V100). Reproduced results are shown in Fig.3 for VGG-16, ResNext-29 and Densenet-121. The levels of accuracy for the selected neural networks that was used for the comparison were further confirmed by Bianco et al. (2018). Due to the use of IRIDIS, there were sufficient computational resources to run all experiments for each dataset.This allowed us to assess not only the reproducibility of the final accuracy scores but also the claim that replacing Batch Normalisation layers with Network Deconvolution layers improved convergence time, leading to higher accuracies in low epoch runs.



		CIFAR-10						CIFAR-100					
		BN1	ND1	BN20	ND20	BN100	ND100	BN1	ND1	BN20	ND20	BN100	ND100
Reproduction Study	VGG-16	12.69%	74.61%	90.35%	93.08%	93.38%	94.31%	1.83%	36.80%	60.40%	72.18%	73.89%	75.78%
	ResNext-29	23.81%	68.47%	91.95%	94.13%	95.16%	95.61%	7.82%	37.61%	72.13%	77.70%	78.35%	79.97%
	Densenet-121	52.58%	75.39%	93.30%	94.83%	94.98%	96.17%	17.56%	44.17%	75.53%	78.02%	78.52%	81.22%
Percentage Difference from Original	VGG-16	-10.13%	0.58%	0.31%	-0.18%	-0.21%	-0.26%	-8.96%	-3.00%	-4.46%	0.29%	1.57%	0.61%
	ResNext-29	-54.33%	-1.08%	-1.26%	0.09%	0.01%	-0.20%	-56.51%	21.60%	-2.87%	0.45%	-0.32%	-0.46%
	Densenet-121	-11.72%	-1.62%	0.05%	-0.06%	0.29%	0.30%	-1.90%	2.94%	0.99%	0.50%	0.68%	0.66%

Figure 3: Table of results and comparison to original paper's results. Note: Difference is measured as a percentage change from the original, not an absolute increase.

The results suggest that the reproducibility of these findings is excellent (Figure 3). In all test runs for more than one epoch, the accuracy varied from the original by less than 3%, with most being less than 1%. Test runs for a single epoch showed significantly more deviation, however this was to be expected as the randomly initialised starting conditions have more of an effect when training for very few epochs. The results also support the claim that the Deconvolution layers increase performance and convergence speed, outperforming the equivalent Batch Normalization networks in every test and doing so by a significantly greater margin in low-epoch tests. Additionally, it is noteworthy that results from architectures implementing Network Deconvolution required more computational resources in comparison to Batch Normalization due to more complex arithmetic calculations.

4.2 Experiment 2

In terms of reproducibility, this experiment was more involved, code was not provided to inspect individual Deconvolutional Kernels. Custom code was created with the GitHub link provided in Sect.6. The "Deconvolution Operation" section from Ye et al. (2020), which is partially summarized in Figure 2, was utilised to inform the construction of this code. We were able to create visualisations of Deconvolutional kernels corresponding to individual channels of individual images, as detailed in Figure 4. However, when it came to comparing to the visualisations in the paper, it became necessary to calculate Deconvolutional kernels for 1024 ImageNet images. The paper did not specify what precisely this means, leaving us to assume that it was the mean average over 1024 iterations of the same process, for different images. Fig 4 displays our results. These did not show the same centre-surround structures that were expected from them. There are a number of possible explanations for this. The smaller sample size used could fail to capture overall trends, there could be an error in our code or a misunderstanding of what exactly is being visualised in the paper. Further investigation, or justification from the authors, would be beneficial in providing a concrete understanding of the ways in which deconvolution improves performance.

4.2.1 Experiment 3

The final experiment in this study applied ND layers to a non-convolutional network. The authors suggested this type of architecture would benefit from this just as the convolutional architectures did. However, in our experiment this was not the case.

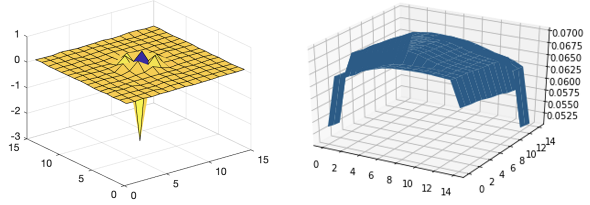


Figure 4: Deconvolution kernels for first channel (red) visualised. (Left) Original for comparison, from 1024 random ImageNet images. (Right) Reproduction from this study, a smaller sample of 128 random ImageNet images were used.

A Multi-Layer Perceptron architecture was trained on simple MNIST and fashion-MNIST datasets, since the other datasets were not as suitable for this type of network. Comparing simple SGD training with Batch Normalization, Deconvolution, or neither, all three methods produced high accuracies of approximately $> 99.5\%$, with similar convergence rates.

5 Conclusion

From Experiment 1, the conclusion that Network Deconvolution is a successful and generalised alternative to Batch Normalisation, is well supported by experimental evidence. However, more work is required to reproduce the remaining experiments that were unable to be reproduced in this study due to resource constraints. Given the widespread use of BN and the potential for ND to displace it in common usage, it is particularly important that these claims are critically reviewed, reproduced and tested on other datasets and network types. Further to this, Experiment 2 did not support the reproducability of the evidence provided for the claim that Network Deconvolution kernels naturally develop centre-surround structure. This implies that further investigation into the biological plausibility would be informative. However sufficient experimental evidence and mathematical justification was provided that biological plausibility should not be seen as necessary. Experiment 3 casts some doubt on the claim that ND layers are a universal improvement over BN. While it has not performed worse than BN, it did not offer significant improvement over BN in the case of simple non-convolutional networks. Further work in this area could include a more rigorous study into the effects of BN on non-convolutional networks, similar in scope to the study reproduced in experiment 1.

6 Code

Full code is available at:

`https://github.com/COMP6248-Reproducability-Challenge`
`/Network-Deconvolution-Reproducibility-Study`

References

Simone Bianco, Remi Cadene, Luigi Celona, and Paolo Napoletano. Benchmark analysis of representative deep neural network architectures. *IEEE Access*, 6:64270–64277, 2018.

Aapo Hyvärinen. Statistical models of natural images and cortical visual representation. *Topics in Cognitive Science*, 2:251 – 264, 04 2010. doi: 10.1111/j.1756-8765.2009.01057.x.

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift, 2015.

Chengxi Ye, Matthew Evanusa, Hua He, Anton Mitrokhin, Tom Goldstein, James A. Yorke, Cornelia Fermüller, and Yiannis Aloimonos. Network deconvolution, 2020.