Implementing deep expander networks on: ResNet, Wide ResNet, and PyramidNet

Jonathan Healy
University of Victoria
Victoria, BC
jonathan.d.healy@gmail.com

Abstract

In this paper, various residual network inspired Convolutional Neural Network architectures are briefly explored and compared. The intention of this approach is to attempt to determine performance differences especially with respect to efficiency. This could be of interest to people with limited resources. All of the testing was done on the CIFAR-10 dataset. ResNet, Wide ResNet and PyramidNet were trained and tested individually and then also by adding Deep expander graphs to create a sparse and efficient but well connected architecture. The results generally show lower accuracy given equal or slightly less training time per epoch. These results are inconclusive and much more testing would be needed to provide definitive answers.

1. Introduction

Convolutional Networks like ResNet and DenseNet focused on designing efficient networks with increasing connectivity [1]. Adding connectivity provides efficient information flow. This allowed CNNs to have more efficient architecture. Two networks that try to improve on deep residual networks (ResNet) are: 1) Wide residual networks (Wide Resnet) and 2) Deep pyramidal residual networks (PyramidNet). Deep expander networks are aimed at providing sparse but well connected layers to further improve efficiency.

The main focus of this paper is on Deep expander networks, also called X-Nets. X-Nets when implemented with ResNet are reported to provide better performance than the original network. X-Nets should be faster to train than other residual networks while providing similar results. To achieve a sparse and lightweight architecture, deep expander networks utilize a class of graphs known as, Expander graphs.

In this paper X-Nets (expander networks) are implemented inside of ResNet, Wide ResNet, and PyramidNet to explore performance and training efficiency. Code from various sources was assembled and put together to allow others the opportunity to investigate these models further.

2. Contributions of original papers

2.1 ResNet

The ResNet network was first introduced in 2015 in a paper entitled, “Deep Residual Learning for Image Recognition” [3]. ResNet introduced a way to successfully process very deep layers. Previously this had not been done on the same magnitude because of the vanishing gradient problem. The vanishing gradient problem occurs because as a gradient is back propagated to earlier layers, repeated multiplication will make a gradient too small. As a network gets deeper, its performance can start degrading quickly. To deal with this, ResNet introduced an identity shortcut connection that skips one or more layers. ResNet set new standards at the time for Convolutional Neural Networks (CNNs).

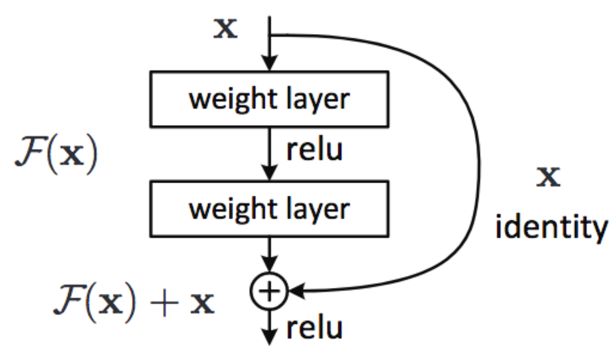

Figure 1: Residual learning: a building block. [3]

2.2 Wide ResNet

Wide ResNet is a network unleashed in 2016 via the paper, “Wide Residual Networks” [2]. The idea behind Wide Residual Networks is that each fraction of a percent in improvement in Deep Residual Networks comes at the cost of nearly doubling the number of layers. This makes Deep Residual Networks very slow to train. Wide ResNet seeks to increase the width of residual networks while decreasing depth at the same time. They showed that even

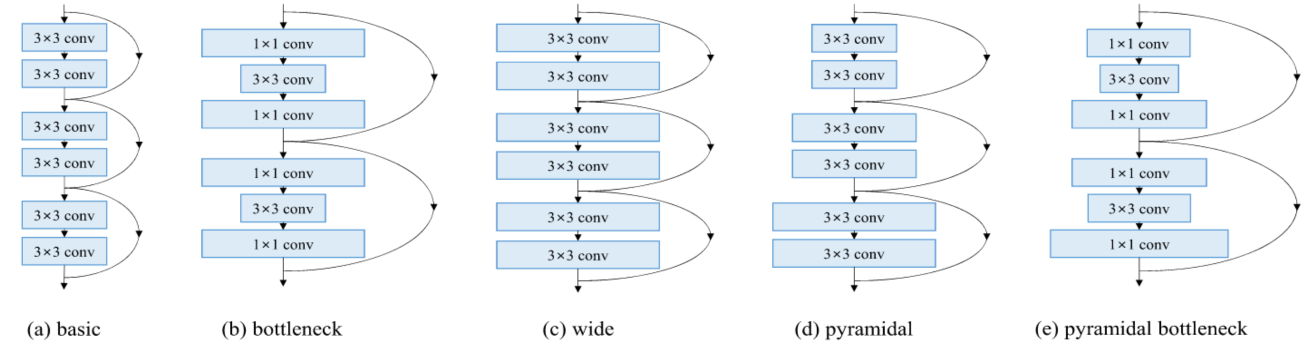

Figure 2: From the paper, "Deep Pyramidal Residual Networks" [4]

a simple 16 layer wide residual network could outperform deep residual networks, even ones with over a thousand layers. This paper also looked at the effects of dropout on residual networks, showing that dropout can lead to lower loss.

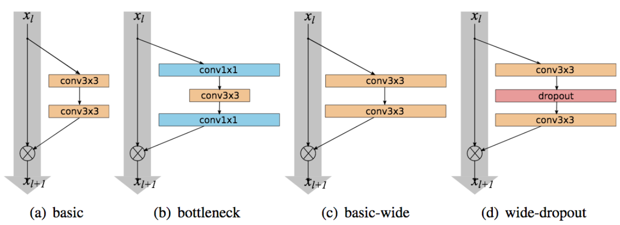

Figure 3: Various residual blocks used in, "Wide Residual Networks" [2].

2.3 PyramidNet

In the paper entitled, "Deep Pyramidal Residual Networks" [4], the authors proposed a new way to improve on deep residual networks like ResNet. They introduce PyramidNet, where the dimensions of the channels gradually increase. In ResNet and Wide ResNet on the other hand, sizes are maintained until a residual unit with down-sampling occurs. In tests using CIFAR-10 and CIFAR-100 PyramidNet reportedly outperforms most other residual networks including Wide-ResNet.

2.4 Deep Expander Networks

The paper, "Deep Expander Networks: Efficient Deep Networks from Graph Theory" [1] proposed modelling connections between filters of a CNN using graphs that are sparse and well connected. Sparse graphs are intended to provide efficiency and well-connectedness is to maintain the power of CNNs. To achieve their goals, the authors employed a class of graphs known as Expander graphs. Connections between filters in CNNs are modelled using Expander graphs and the architecture that this provides is called an X-Net. Two guarantees are provided concerning X-Nets [1]:

1) Each node influences every node in a layer of logarithmic steps.
2) The number of paths between two sets of nodes is proportional to the product of their sizes.

It is claimed that Expander based models give a 4% improvement over similar ideas on MobileNet like grouped convolutions which achieve sparse connections but lose connectivity. It is further asserted that X-Nets give better performance trade-offs than the original ResNet architecture. The authors created model sizes similar to what would be achieved with pruning, without pruning. They seem to have provided the first attempt to improve deep networks using graph theory. By using X-Net architecture one should be able to train deeper and wider CNNs.

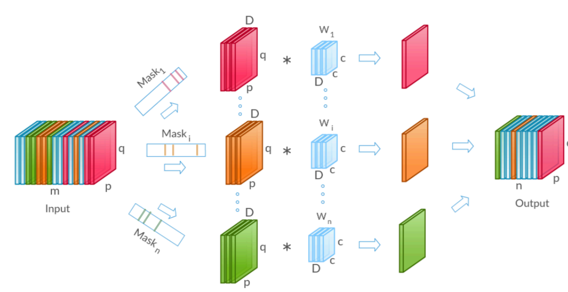

Figure 4: Fast convolution algorithm in deep expander networks [1].

3. Contributions of this paper

This paper seeks to examine and implement three types of residual networks: 1) the original ResNet, 2) a wide residual network or Wide-ResNet and 3) a ResNet version of PyramidNet or a deep pyramidal residual network. Furthermore, Expander graphs presented in the paper, "Deep Expander Networks: Efficient Deep Networks from Graph Theory" are used to compare and contrast all three.

Each network is run for 80 epochs with a dropout rate of 0.3. In each network the learning rate is reduced at the same intervals. Training begins with a learning rate of 0.025 and is divided by 5 to 0.005 at epoch 40 and then again to 0.001 at epoch 70.

The time to complete each epoch was purposefully set to approximately 3 minutes by adjusting the network parameters. All of the training was done on an Nvidia

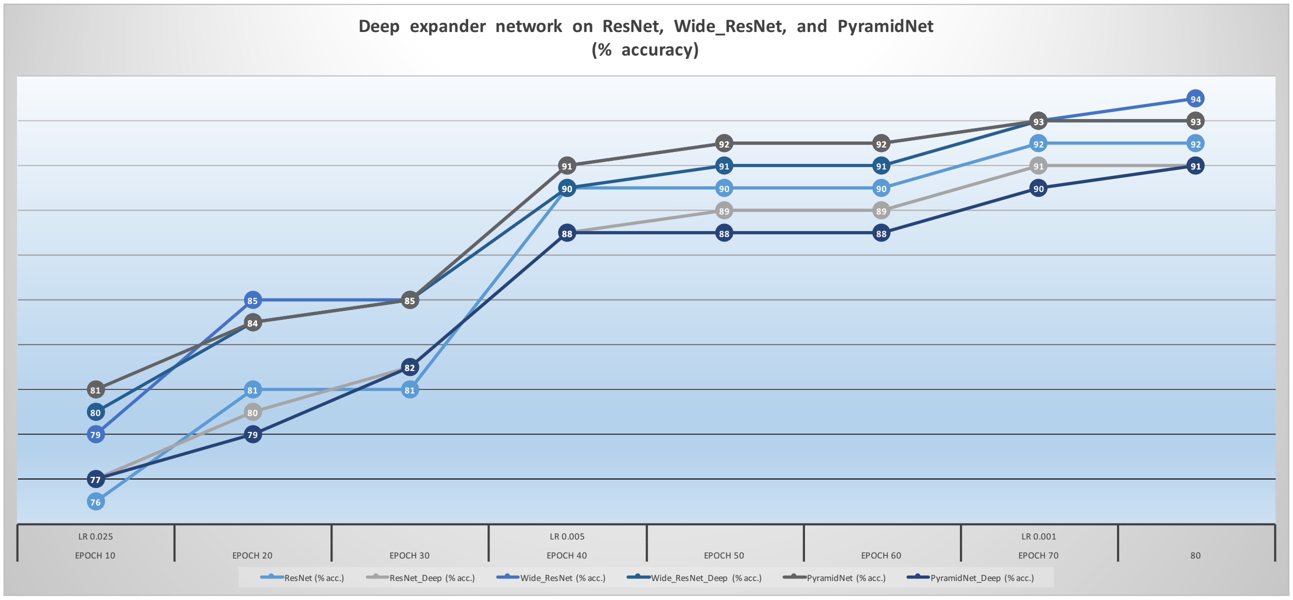

Figure 5: Accuracies graphed per every 10 epochs

GTX 1070ti graphics card with 8 mb. of memory. The memory constraints of this card prevented testing larger networks. All of the training was done on the CIFAR10 dataset.

The idea behind the approach used in this paper is that deep expander networks or X-Nets, being more efficient, should train to higher levels of accuracy in the same time period when applied to and then compared with ResNet, Wide-ResNet, and PyramidNet architecture. Deep expander networks are implemented by changing the convolutional layers, applying a mask to create an Expander graph.

Accuracy (%) CIFAR10	Epoch 20 lr 0.025	Epoch 40 lr 0.005	Epoch 80 lr 0.001	Time/ Epoch (m:s)
ResNet	81	90	92	2:50
ResNet – Deep	80	88	91	2:37
WideResNet	85	91	94	3:17
WideResNet - Deep	84	90	93	3:15
PyramidNet	84	91	93	2:45
PyramidNet - Deep	79	88	91	2:47

Table 1: Comparing accuracy at different epochs.

4. Results

4.1. ResNet, ResNet-Deep

ResNet and ResNet-Deep were trained with a depth of 50. ResNet trained for approximately 2:50 per epoch while ResNet-Deep trained for 2:37 per epoch. ResNet-Deep was set with the expandsize parameter at two which represents the compression ratio, meaning that all connections were compressed by the expander graph by a factor of two.

At the same depth, ResNet showed consistently higher accuracy than ResNet-Deep which uses deep expander graphs to ensure sparse but good connectivity. ResNet-Deep did train for approximately 13 seconds less per epoch at the same depth which is not insignificant. Expandersizes of 4 and 8 were also tested but this led to even lower accuracy.

4.2 WideResNet, WideResNet-Deep

Wide-ResNet and Wide-ResNet-Deep, at accuracies of 94% and 93% respectively showed better performance than any of the other networks except for PyramidNet. This however, is offset by the fact that the Wide-ResNet versions used in this experiment trained for longer at 3:17 and 3:15 per epoch respectively compared to 2:50 and 2:37 for the ResNet versions. This is a considerable time difference and could explain the higher accuracies. Both Wide-ResNet and Wide-ResNet-Deep were trained with a depth of 16 and a widen_factor of 10. The expandsize for Wide-ResNet-Deep was four.

4.3 PyramidNet, PyramidNet-Deep

PyramidNet and PyramidNet-Deep were both trained with depths of 38. Training at this depth means that Bottleneck blocks were employed. Also, PyramidNet-Deep was trained with an expandsize of 2. The accuracies achieved by PyramidNet were impressive considering the lower time spent per epoch at two minutes and 45 seconds. The accuracies are almost identical to WideResNet-Deep with a full 30 seconds less time spent per epoch.

The Deep expander version of PyramidNet performed significantly worse than the normal version with approximately the same amount of time spent. The differences between PyramidNet and PyramidNet-Deep were greater than the differences between the ResNet and Wide-ResNet variations.

5. Conclusion

A lot more testing would be needed to make conclusive statements about any observations made while conducting this study. At first glance, it may seem like deep expander networks do not live up to the hype as they produce lower accuracy on the CIFAR10 results compared to the models that they were tested against. Careful diligence was used in implementing the algorithms properly but this could easily be a real source of error. It is claimed that deep expander networks should allow for deeper and wider CNNs. They should theoretically also lead to networks being trained faster. Based on the results produced here, this is not the case. Much more thought would be needed to make this statement with any certainty.

Comparing ResNet, Wide-ResNet, and PyramidNet, without the implementation of deep expander networks, leads to the conclusion that PyramidNet performs much better than the other two. PyramidNet has produced almost identical accuracy compared with Wide-ResNet over a much lower time period. Making this observation with any real certainty however, would entail much more training and testing.

Interestingly the model sizes for the Deep expander graph versions are all just about double the size of the non-Deep expander graph versions. This could suggest an improper implementation of the X-Net architecture (deep expander graphs). This in turn points to a poor understanding of the knowledge base by the author. In turn, it could possibly point to a need to re-evaluate Deep expander graphs for CNN architecture. This last point is much more unlikely and in any case much more study is needed to fully understand these networks.

References

[1] Prabhu, Ameya, Girish Varma, and Anoop Namboodiri. “Deep expander networks: Efficient deep networks from graph theory.” Proceedings of the the European Conference on Computer Vision (ECCV), pp. 20-35. 2018.

[2] Zagoruyko, Sergey, and Nikos Komodakis. “Wide residual networks.” arXiv preprint arXiv:1605.07146. 2016.

[3] He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. “Deep residual learning for image recognition.” Proceedings of the IEEE conference on computer visions and pattern recognition, pp. 770-778. 2016.

[4] Han, Dongyoon, Jiwhan Kim, and Junmo Kim. "Deep pyramidal residual networks." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 5927-5935. 2017.