Paper: *Deep Residual Learning for Image Recognition*

Nirajan Koirala

CSC 9010_001

Summary

The authors present a novel way of training deeper networks using residual learning framework in this paper. Using this framework on a network with 152 layers, they delivered an astounding top-5 error rate of 3.57% on the ImageNet test set. They key to their network's success is that they were able to train it using shortcut connections where the signal feeding into a layer is also added to the output of a layer located higher up the stack of networks.

When training very deep networks, it seems obvious that the more layers we provide, the more abstract features from the data can be learned from the networks. However, as the networks get very deep, strange results are obtained where both the training and test set error is higher than the network's shallow counterparts. Even though the vanishing/exploding gradient problem is addressed using normalization the accuracy of the network still gets saturated and degrades rapidly with increasing layers. Such a phenomenon is not caused by overfitting as adding more layers leads to higher training error. A new architecture with identity mapping into the layers is suggested and according to this solution deeper models should produce no higher training error than its shallower counterpart. But experiments show that these solvers are unable to find any better solutions.

To address the degradation problem, instead of directly fitting a desired underlying mapping, authors let the layers fit a residual mapping. In this way they added the input x to the output of the network and then the network is forced to model $f(x) = h(x) - x$ rather than $h(x)$. They hypothesize that it is easier to optimize the residual mapping than to optimize the original, unreferenced mapping. Also, if an identity mapping were optimal, it would be easier to push the residual to zero than to fit an identity mapping by a stack of nonlinear layers. They have also mentioned that identity shortcut connections add no extra parameter or computational load with training. Adding many shortcut connections, the network can start making progress even if the several layers have not started learning yet since the signal can easily make its way across the whole network.

In the later sections, they have provided the equations for the residual learning and also suggested that if the dimensions of x and F are not equal, a linear projection W_s can be performed by the shortcut connections to match the dimensions. In the paper they have used 2 or 3 layers for the residual function and each residual unit is composed of two convolutional layers, with Batch Normalization and ReLU activation, using a 3 x 3 kernel and preserving spatial dimensions. Testing of various plain/residual networks is done and the residual models have fewer filters as well lower complexity and lower FLOPS than its plain counterparts. To address the problem of unmatching dimensions when adding shortcuts, authors have mentioned two solutions, one with extra zero entries padded and the other is the projection shortcut done by 1 x 1 convolutions. In further sections, they discuss how the networks were implemented for the ImageNet and the methods used for their experiments. First they compare the deep and shallow plain networks and make a conjecture that the

reason for the higher error rates of deep plain networks is exponentially low convergence rates. Then, they evaluate the deep and shallow residual nets and observe that the deeper (34 layer) network have lower error rate than the shallow (18 layer) network. They suggest that using ResNets they have addressed the degradation problem and managed to obtain accuracy gains from increased depth. They also mention that if the depth of network is kept the same, ResNets converges faster than plain nets and eases the optimization. Next, they investigate the shortcuts in networks and after evaluating various options, they conclude that projection shortcuts are not essential for addressing the degradation problem and identity shortcuts are important as they do not increase the complexity of the very deep networks in which bottleneck design is used. For deeper networks like ResNet50 and ResNet152, they have used the bottleneck design to decrease the training time. In this design, for each residual function F, 3 layers are stacked one over the other and the three layers are 1 x 1, 3 x 3, and 1 x 1 convolutions. The 1 x 1 convolution layers are responsible for reducing and then restoring the dimensions. The 3 x 3 layer is left as a bottleneck with smaller input/output dimensions. After increasing the depth considerably, they still do not observe the degradation problem and produce significantly higher accuracies. They then combine six models of different depths to form an ensemble leading to 3.57% top-5 error on the test set which helped them won 1st place in ILSVRC 2015. Later in the paper, they discuss how applying similar residual learning framework on other datasets and problems helped them obtain better results than similar plain networks.

Strengths

- The paper goes in very detail about all the methods used and provide sufficient figures and tables.
- The experiment section is very thorough and well-supported. It talks about plain, deep and ensemble networks and helps in understanding reason behind various approaches they take.
- Authors have tested their learning framework on other datasets like CIFAR-10 as well to provide substantial basis for their claims.

Weaknesses

- The introduction section is a little lengthy than necessary. They already start discussing about the formal underlyings of residual network even though they have done it fully in the 3.1 section which is all about residual learning method.
- The organization of the paper needs a little improvement since many topics like degradation problem is discussed multiple times similarly in the paper.

Points of Confusion

- I am confused about how the different layers described in the bottleneck architecture help reduce the time complexity.

- Authors mention about the reasonable preconditioning that identity mappings provide. I am not sure how this leads to small responses.
- When the networks are learning using backpropagation, wouldn't there be two pathways for the gradients in residual blocks. How would it decide which path to follow?

Discussion Questions

- On very deep layered networks like the 1202 ResNet, the errors level is found to be higher than the same network with 110 layers. Authors argue that this may be occurring due to overfitting. In general, is there a way to decide if a certain number of layers are necessary for a particular dataset or we just have to find the right amount of layers by experiments and validation?
- In the paper, ReLU is applied after the addition of f(x) and x. Would it make any difference to the network if ReLU is applied before that or even at the beginning of each residual block? In particular how do we find this ReLU activation location for best performance?
- If the degradation problem is not caused by overfitting and deep neural networks are not able to learn identity functions, does it mean that deep networks are suffering from the curse of dimensionality?