Response to He et al. (2016)

Matthew Evans

March 26, 2025

Overview

Adding layers to a network's architecture increases its complexity and potentially its predictive ability. Yet, beyond a certain layer depth, accuracy degrades. In [1], He et al. propose a novel approach to overcome this problem, unlocking the potential of deep networks (i.e., with hundreds of layers) for image classification tasks.

Approach

The layers of a network can be thought of as functions with the output of one function being the input of the next. The network as a whole can thus be viewed as the function composition

$$Y = H_n \circ H_{n-1} \circ H_{n-2} \circ \cdots \circ H_1(x)$$

with Y as the final output, $H_1, H_2, \ldots, H_n$ as the hidden layers, and x as the input.

Thus, the goal of training is to learn sufficiently close approximations of each layer's function so as to yield the correct final output. In some task domains, such as image classification, each of these functions can be thought of as extracting increasingly sophisticated features (e.g., edges, then textures, then objects) from the input data to aid in the classification task. Yet in networks with many layers, the optimal adjustment made by each layer may be small, or even no adjustment at all. In other words, if a object is already sufficient identifiable at the ith layer, there is no need to extract additional features in the remaining $n - i$ layers. In fact, learning the remaining $n - i$ functions is a waste of computation at best and, at worst, can be detrimental to the network's accuracy, an issue known as the *degradation problem*. The authors introduce the strategy of using *shortcut connections* to overcome the degradation problem.

Residual Functions and Shortcut Connections

In *plain networks* (i.e., those not using the strategies introduced in [1]), for each hidden layer, the network aims to learn that layer's function,

$$H_i(x_i).$$

As noted above, in deep architectures for image classification, difference between a layer's input and output is often small. That is,

$$H_i(x_i) = F(x_i) + x_i$$

where $F(x_{i-1})$ is a *residual function* capturing this small difference between input and output. The authors' key insight is that since $H_i(x_i) \approx x_i$, the network's attention should be focused on the more tractable problem of computing $F(x_i)$. To force this learning behavior, the authors' proposed *residual network* architecture introduces a *shortcut connection*, supplying a given layer's input as supplemental *output* of a subsequent layer's activation function (Fig. 1).

Additional Benefits

The authors demonstrate the following benefits of this strategy.

- Using shortcut layers does not introduce any additional parameters, and the introduction of the sum adds minimal computational overhead.

- Shortcut layers can be used with variety of layer types (e.g., fully connected, convolutional, etc.).
- Shortcut layers can be effectively added to existing network architectures, yielding faster convergence, and in the case of deeper networks, dramatically improve accuracy.

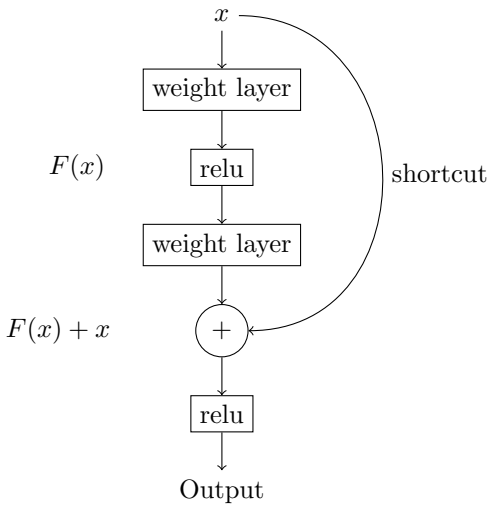

Figure 1: A residual learning building block as described by [1].

Measures of Success

The authors validate their approach through experiments on the ImageNet 2012 dataset, showing that ResNets significantly outperform plain networks in both accuracy and convergence. For instance, a 34-layer ResNet achieves a top-1 error rate of 25.03% compared to 28.54% for a plain network, with even greater gains for deeper networks (e.g., 19.38% for a 152-layer ResNet). ResNets are also shown to converge faster during training, easing optimization.

On the CIFAR-10 dataset, ResNets achieve state-of-the-art results with fewer parameters and faster convergence, further demonstrating their effectiveness in addressing the degradation problem and enabling the training of very deep networks.

Considerations

- While the authors offer conjecture for the exact root cause of the degradation problem, suggesting that perhaps deep plain networks suffer from exponentially low convergence rates, they offer no sure answer. While this doesn't detract from their demonstrated performance gains, it can't make sure claims on other problem domains.
- The authors suggest that their experiments with a variety of vision related tasks indicate that residual learning is a generic principle, and thus expect that it is applicable in non-vision problems, but offer no proof of this.

Impact

Since ResNet [1], residual functions and shortcut connections have been adopted in a variety of problem domains and deep network architectures including, but also well beyond that of image classification and object detection. Of particular note are *transformer* network architectures which use a similar concept ("add & norm" layers). Popular examples include BERT with 12+ layers [2], and GPT-3 with 96 layers[3].

References

[1] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[2] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018.

[3] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. *CoRR*, abs/2005.14165, 2020.