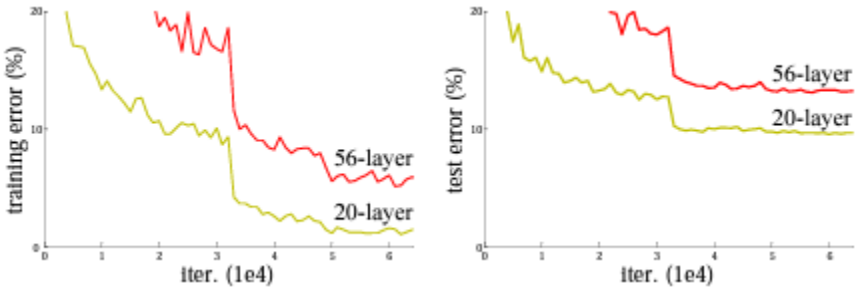
Deep Residual Learning for Image Recognition

Deep Residual Learning for Image Recognition (ResNet)[1] is one of the most influential neural network designs ever published, with over 20000 citations. It allows us to train very deep neural networks with increasing performance. Previous models of ImageNet had depths of 16 to 30 layers. ResNets increased this to as many as 152 layers (in ResNet-152).



Deep Neural networks happen to be difficult to train due to well-known problems known as the *"vanishing" gradient*[2] and the *"exploding" gradient*[3]. The authors have shown on the CIFAR-10 image classification dataset that deepening a convolutional neural network just by stacking more convolution layers on top of activations and batch normalizations[4] decreases performance (increases %error) instead of increasing it. This is known as the *degradation* problem. This problem was not caused due to overfitting of parameters (which is implied by when the test error goes up while the training error is low). The ResNet aims to solve precisely this problem.

ResNet reformulates the layers as learning residual functions with reference to the layer inputs (x), instead of learning unreferenced functions. These Residual networks use the concept of "skip connections". Instead of stacking convolutional layers in a traditional way, the original input is also added as a component to the output of the convolutional block.

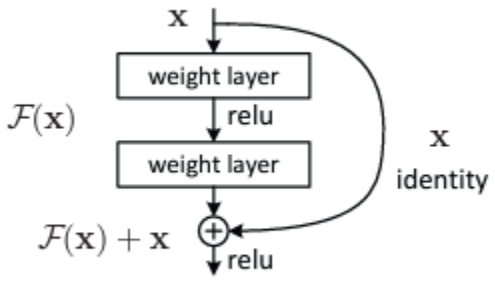


Figure 2. Residual learning: a building block.

In ResNet, we have two kinds of blocks - convolutional blocks and identity blocks. Identity block is used when the input size and output size of the image is the same. That is, if the sizes of the

input (x) and the output (F(x)) are the same, the input can be directly added to the output through the skip connection line (as depicted by solid lines in the architecture diagram in the paper). However, when the sizes are different, they have to be added in a different way (depicted by dotted lines in the architecture). The paper proposes different schemes for upsampling the previous input layer's dimensions through this identity skip connection.

$$\mathbf{y} = \mathcal{F}(\mathbf{x}, \{W_i\}) + W_s\mathbf{x}.$$

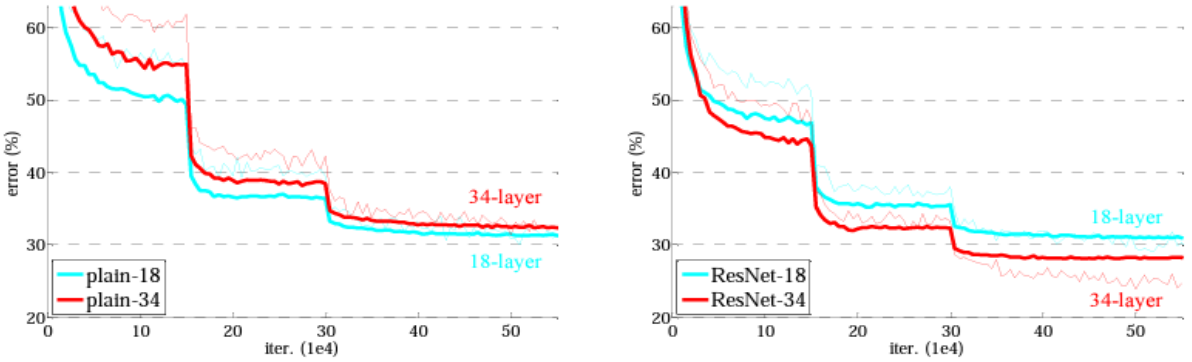
The first proposed way is to zero-pad the outsize of the dimensions - the shortcut path still performs the identity mapping, with extra zero entries padded for increasing dimensions. The benefit of this is that it is very quick as there is no extra parameter involved while doing this. In the second method, we expand the dimensions using a 1*1 convolution (filter size =1). That is, we have to add a convolution layer in the skip connection line. The task of the convolution layer is to make the input and output sizes compatible to be added.

The paper compares how ResNet looks in comparison to a 34-layer plain network (a series of convolution layers stacked followed by activations, followed by batch normalization, as mentioned in the beginning) and also in comparison to another really popular model, the VGG-19.

The ResNet experiments tested 152-layer ResNet on ImageNet, which was 8 times deeper than the VGG nets. A highlight of this was the ResNet having less parameters, leading to less computational expense in terms of FLOPs (Floating Point Operations per second). The ResNet-152 took 11.3 billion FLOPs as compared to VGG-16, which took 15.3 billion FLOPs, and also VGG-19, which took 19.6 billion FLOPs. The ensemble of ResNets were able to achieve 3.57% error on the ImageNet test set, earning it the 1st place in ILSVRC 2015. ResNet was also used on COCO object detection dataset, showing a whooping improvement of 28%, and thus also earning it the 1st place in COCO detection and segmentation.

Other miscellaneous hyperparameter details about the ResNet experiment are:

1. Batch Normalization[4] is used after every convolution and before every activation.
2. It uses a batch size of 256.
3. It uses He initialization[5] invented by the author of the paper, Kaiming He.
4. The Learning rate starts from 0.1 and is divided by 10 when the error plateaus.
5. It uses weight decay of 0.0001
6. It uses momentum of 0.9
7. It doesn't use dropouts[6].



Thus, the paper shows how ResNet performs increasingly better with the increase in number of layers (above figure showing the experimental result on increasing the number of layers from 18 to 32) as compared to the plain concatenated convolutional layers.

REFERENCES

1. He, K. (2015b, December 10). *Deep Residual Learning for Image Recognition*. arXiv.Org. https://arxiv.org/abs/1512.03385
2. Wang, C. (2021, December 7). *The Vanishing Gradient Problem - Towards Data Science*. Medium. https://towardsdatascience.com/the-vanishing-gradient-problem-69bf08b15484
3. DeepAI. (2020, June 25). *Exploding Gradient Problem*. https://deepai.org/machine-learning-glossary-and-terms/exploding-gradient-problem#:%7E:text=Exploding%20gradients%20are%20a%20problem,updates%20are%20small%20and%20controlled.
4. Ioffe, S. (2015, February 11). *Batch Normalization: Accelerating Deep Network Training by. . .* arXiv.Org. https://arxiv.org/abs/1502.03167
5. He, K. (2015a, February 6). *Delving Deep into Rectifiers: Surpassing Human-Level Performance*. . . arXiv.Org. https://arxiv.org/abs/1502.01852
6. Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. (January 2014). *Dropout: a simple way to prevent neural networks from overfitting*. J. Mach. Learn. Res. https://dl.acm.org/doi/10.5555/2627435.2670313