# Deep Residual Learning for Image Recognition

Summary written by Nicholas Hinke
March 28, 2022

**Summary:** As demonstrated in the years prior, the importance of network *depth* had been growing clearer and clearer [9, 13, 16]. However, due to the many challenges of training very deep networks, deeper networks did not always perform better than their shallower counterparts. This was due to a myriad of problems, including overfitting due to high model complexities, vanishing or exploding gradients, and accuracy degradation as a result of excessive depth. In this paper, the authors propose a novel method for constructing networks to combat the last problem of performance degradation, and subsequently demonstrate the ability of their networks to go far deeper without sacrificing performance. Denoted as *deep residual networks* (or "ResNets"), these networks employed what are known as "residual blocks" in an attempt to recapture the meaning of a given layer's input within its output (*i.e.* so that none of the input meaning is lost as the network grows deeper). Inspired by the then-recent work on the vanishing/exploding gradients problem [1, 4, 7, 10], other deep convolutional neural networks (CNNs) [9, 10, 12, 19], shortcut connections in neural networks [2, 8, 11], highway networks and gating functions [6, 14, 15], as well as hierarchical basis preconditioning [3, 17, 18], the authors claim to have developed networks that can far outperform any of the other existing state-of-the-art models. Indeed, this was demonstrated by the authors' victories at several image classification competitions using ResNets in 2015 [5].

**Approach:** As noted above, the purpose of these residual networks was to address the issue of accuracy degradation as network depth increases. This was done directly through the use of residual blocks (which are typically comprised of two or three layers), which could be inserted anywhere within a standard network architecture. For a feed-forward neural network, these residual connections could be implemented using an identity "short circuit connection" from the input of the block to the summation before the last output nonlinearity is applied. Since the identity mapping was used within the short circuit connection, there were no additional parameters to be learned, nor was there any significant added computational cost (since the element-wise addition is essentially negligible). Thus, this method of utilizing residual blocks provides a highly efficient way to ensure that the input data to each block is not forgotten in its output, thereby allowing networks to grow deeper without suffering from the problems of degradation [5].

Within this paper and their competition submissions, the authors primarily focus on residual networks containing anywhere from 34 to 152 layers. However, using this technique, the authors also demonstrated the ability to construct a network containing over *one thousand* layers through the use of so-called "bottleneck layers", without substantially sacrificing performance due to degradation. Indeed, a 1202-layer ResNet was outperformed by just 1.5% in comparison to a 110-layer ResNet on the CIFAR-10 dataset (achieving error rates of 7.93% and 6.43%, respectively), and the authors argue that the decrease in performance is primarily due to overfitting, *not* degradation [5].

In order to decisively demonstrate the superior performance of ResNets, the authors boasted a single model validation error of just 4.49% for a 152-layer ResNet on the ImageNet dataset. Consequently, this single model outperformed *all* of the previous best ensemble results. Moreover, using an ensemble network containing six ResNets of varying depths, the authors won the ImageNet Detection and ImageNet Localization competitions at ILSVRC 2015, and they won then COCO Detection and COCO Segmentation competitions at COCO 2015 [5].

**Strengths:** Most notably, the resulting ResNets constructed by the authors outperformed all other state-of-the-art networks on a variety of datasets, thus pushing the boundary of research in this area. Additionally, this paper demonstrated the potential for further accuracy *improvements* as network depth increases, rather than performance degradation. Finally, the authors effectively displayed the generality of this approach, both by validating its performance on a wide variety of datasets, and by demonstrating the ability to train ResNets using the familiar techniques of SGD and backpropagation and implement them using commonly available libraries such as Caffe [5].

**Weaknesses:** As mentioned by the authors, the motivating hypothesis behind the use of residual blocks is still an open question, and thus desperately requires additional formal study [5]. Moreover, while the authors do briefly mention the ability to apply this approach to other network architectures rather than only feed-forward networks, it would be quite useful for the reader to see a more detailed discussion regarding the use and applicability of residual blocks within other architectures such as CNNs and RNNs.

**Reflections:** Upon reaching the conclusion of the paper, there seem to be two main areas in which further research may prove especially useful. First, it may prove quite beneficial to further study how the number of internal layers

within a residual block affects the performance of the network as a whole. Additionally, while the author's claim that their approach is superior since the residual connections can never be "closed" [5], it may prove useful to examine the effects of utilizing mappings other than the identity mapping within the short circuit connections (in a similar manner to "highway networks" [6, 14, 15]).

## References


[1] Y. Bengio, P. Y. Simard, and P. Frasconi. Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks*, 52:157–66, 1994.

[2] C. M. Bishop. Neural networks for pattern recognition. 1995.

[3] W. L. Briggs. A multigrid tutorial. 1987.

[4] X. Glorot and Y. Bengio. Understanding the difficulty of training deep feedforward neural networks. In *AISTATS*, 2010.

[5] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[6] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Computation*, 9:1735–1780, 1997.

[7] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *ArXiv*, abs/1502.03167, 2015.

[8] K. Kafadar, J. R. Koehler, W. N. Venables, and B. D. Ripley. Modern applied statistics with s-plus. 1999.

[9] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012.

[10] Y. LeCun, B. E. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. E. Hubbard, and L. D. Jackel. Backpropagation applied to handwritten zip code recognition. *Neural Computation*, 1:541–551, 1989.

[11] B. D. Ripley. Pattern recognition and neural networks. 1996.

[12] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun. Overfeat: Integrated recognition, localization and detection using convolutional networks. *CoRR*, abs/1312.6229, 2014.

[13] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2015.

[14] R. K. Srivastava, K. Greff, and J. Schmidhuber. Highway networks. *ArXiv*, abs/1505.00387, 2015.

[15] R. K. Srivastava, K. Greff, and J. Schmidhuber. Training very deep networks. In *NIPS*, 2015.

[16] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. E. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–9, 2015.

[17] R. Szeliski. Fast surface interpolation using hierarchical basis functions. *Proceedings CVPR '89: IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 222–228, 1989.

[18] R. Szeliski. Locally adapted hierarchical basis preconditioning. *ACM Trans. Graph.*, 25:1135–1143, 2006.

[19] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In *ECCV*, 2014.