# Residual Networks and its Variations

**Bonggun Shin** BONGGUN.SHIN@EMORY.EDU
Emory University

## Abstract

Deep networks have shown state of the art performance in image recognition problems in past few years. However, there exists practical limitations of depth of layers, because deeper networks are difficult to train. Recently, a reformulation method for the layers has been introduced in the literature, which is called residual networks, and it shows not only good performance but also nice properties. In this paper, we review residual networks and its several variations. In addition, we investigate experimental results with CIFAR-10/100 and Imagenet dataset.

## 1. Introduction

In the past few years, deep learning has proved itself excellent at many domains, such as object detection (Girshick et al., 2014), segmentation (Long et al., 2015), human pose estimation (Toshev & Szegedy, 2014), video classification (Karpathy et al., 2014), object tracking (Wang & Yeung, 2013), superresolution (Dong et al., 2014), and sentiment analysis (Severyn & Moschitti, 2015). Among those application domains, computer vision has been a leader of deep learning, because the major breakthroughs of deep learning have come from a computer vision competition, called ImageNet. The first successful deep learning approaches in ImageNet Cometition was AlexNet which was proposed in 2012. Since then many deep learning models have been introduced in the literature, such as GoogLenet, VGG, and ResNet. ResNet is currently by far state of the art convolutional neural network models.

### 1.1. Previous models before ResNets

The first model that revived convolutional neural networks in computer vision was AlexNet (Krizhevsky et al., 2012), which is proposed by Alex Krizhevsky in 2012. It was renowned for significantly outperforming the second runner-up in the ImageNet ILSVRC challenge in 2012. The top-5 error was 16% while the second place model got 26%. The model is basically the same with LeNet (LeCun et al., 1995), which is the first model of neural Networks for vision, and it is proposed by Yann LeCun in 1990s. The difference comes from the fact that AlexNet has deeper layers in that each layer stacks on top of each other layers. In the ImageNet competition in 2013, a new algorithm named ZF Net (Zeiler & Fergus, 2014) was developed by Matthew Zeiler. It is just an extension of AlexNet by tweaking hyperparameters especially the number of hidden layers, the stride, and the filter size.

In the competition in 2014, two important models are recognized. One is the winner's model, named GoogLeNet (Szegedy et al., 2015a), and the other one is the runner-up algorithm called VGGNet (Simonyan & Zisserman, 2014). The major contribution of GoogLeNet is that it greatly curtailed the number of parameters in the network (less than 7% of the size). The large portion of reduction of the number of parameters attributes to using an inception module and average pooling helps cuts down more parameters that do not matter much. Since it is using inception module, it has another name InceptionNet. Since then, there are follow up versions of InceptionNet, from Inception-v1 to Inception-v4. The second place model was VGGNet by Karen Simonyan. Although the model didn't achieve the best performance, its contribution was in showing the depth of the network is critical to the performance. They stack 16 homogeneous components, which are consist of convolution and pooling. They achieved good performance by just stacking the same components deeply.

However, VGGNet suffers from the degradation problem. If the depth of networks is very deep, the accuracy gets saturated and starts to degrade swiftly. To handle this problem, Kaiming He et al developed ResNets (He et al., 2015) and this model won the 1st place on the competition in 2015. They proposed special skip connections and used batch normalization heavily.

All the models discussed in this review are presented with experimental results in computer vision applications.

Proposal of a class Project of CS572(IR) at Emory University

## 2. Deep Residual Learning for Image Recognition

### 2.1. Motivation

In traditional neural networks, deep models are not popularly used, because there is an obstacle to make them converge. This is caused by the notorious problem of vanishing/exploding gradients (Bengio et al., 1994; Glorot & Bengio, 2010). Recently, many kinds of normalization methods is introduced to relive this problem, and it enables networks with deep layers converges well. But as the network depth is increased more and more, the accuracy becomes saturated and then degrades quickly. The authors claim that overfitting is not the reason for the degradation, and they thoroughly verify this with empirical results. As a solution of the degradation problem, they introduce a residual block.

### 2.2. Model Description

Deep neural networks stack a building block repetitively. Before ResNet, many deep neural networks architectures use layers to fit a desired underlying mapping. However, the layer of ResNet tries to fit a residual mapping. Let $H(x)$ denotes the desired underlying mapping with the inputs $x$. They define residual function as $H(x)-x$, and call it $F(x)$. Therefore the original function becomes $F(x)+x$. With or without the residual block, we can fit an original function. But once we use the residual block, learning becomes easier than using just a standard layer. This reformulation is actually helpful in practical learning, because if the optimal function is similar to an identity mapping, then the solver lead the weights of the nonlinear layers to zero to make it closer to identity mappings.

### 2.3. Experiments

They show empirical results on the ImageNet 2012 classification dataset (Russakovsky et al., 2015) There are 1.28 million training images and 50k validation images. The performance is measured on 100k test images. The Evaluation metrics are top-1 and top-5 error rates. To contrast the performance of the proposed model, they evaluate 18-layer and 34-layer plain nets. And the proposed models are 18-layer and 34-layer ResNets.

The 34-layer plain networks have higher errors than 18-layer networks for both train and validation dataset. And this is called degradation problem. However this is not reasonable, because the solution space of the 34-layer plain networks is a superspace of that of the 18-layer one. The authors tried to find the reason experimentally, by looking at forward propagated signals and backward propagated gradients, but they were unable to find it.

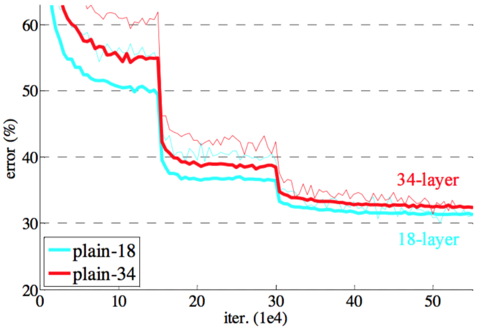


*Figure 1.* Errors for plain networks of 18 and 34 layers. Thin curves denote training error and bold ones denote validation errors.

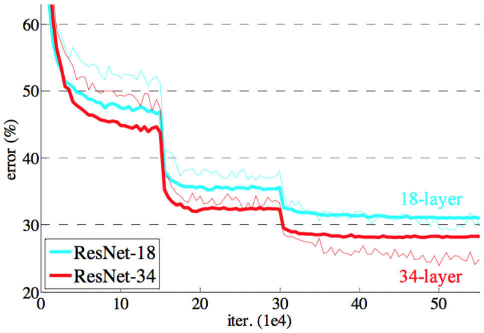


*Figure 2.* Errors for residual networks of 18 and 34 layers. Thin curves denote training error and bold ones denote validation errors.

In the contrary, residual network based models don't have the degradation problem. The fact that 34-layer ResNet is better than 18-layer ResNet is meaningful in terms of deep learning, because we can stack as many layers as we want. In addition, they outperform the plain networks, in both accuracy and speed. ResNet reduces the top-1 error by 3.5% compared with a plain counterpart(Figure 1, 2). Not only that, ResNets converge faster than plain nets.

They compare identity shortcut with projection shortcut, bottleneck and non-bottleneck architectures, and more deeper ResNets up to 152 layers. And lastly, they compare ResNets with state of the art methods, such as VGG, GoogLeNet, and BN-inception.

### 2.4. Conclusion

This paper propose reformulation of layers with a residual block. They show comprehensive experimental results showing that the proposed residual block is practically helpful for deep learning. ResNets don't have the degradation problem, outperform state of the art in terms of both accuracy and speed.

# 3. Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning

## 3.1. Motivation

The motivation of this paper is an attempt to combine two famous models, inception-v3 (Szegedy et al., 2015b) and ResNets (He et al., 2015). The two models are famous for not only performance but good characteristics. Inception-v3 is computationally efficient and ResNet enables a network have deeper layers. In this paper the authors design Inception-v4, a technical variation of Inception-v3 by taking advantage of Tensorflow to mitigating technical difficulties of Inveption-v3, such as the partitioning problem when distributing the algorithm. Next they combine Inception-v4 and ResNet. The proposed model outperforms state of the art models in the ImageNet competition dataset.

## 3.2. Model description

They use simpler Inception blocks than the original Inception. Since this Inception causes reduced dimensions, they use filter-expansion layer to compensate the reduced dimensions.

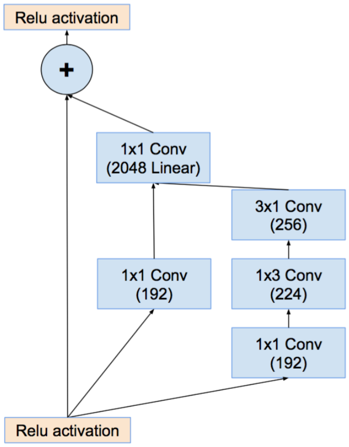


*Figure 3.* Inception-ResNet-v2 module

Although they try a couple of versions of the residual-inception module, they present details of only two of those in the paper, v1 and v2(Figure 3). V1 is designed based on Inception-v3, while v2 matches Inception-v4 network.

## 3.3. Experiments

They show empirical results on the ImageNet 2012 classification dataset (Russakovsky et al., 2015) The Evaluation metrics are top-1 and top-5 error rates. To contrast the performance of the proposed model, they evaluate Inception-v3 and Inception-v4 models. And the proposed models are residual-inception-v1 and residual-inception-v2.

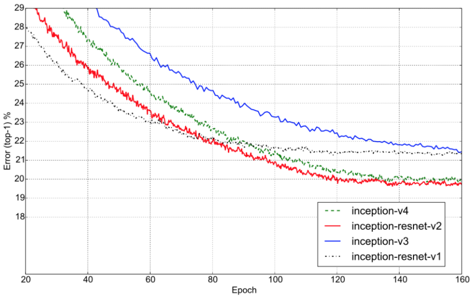


*Figure 4.* Top-1 error of four models

The combined models show better results in terms of accuracy. In addition, an Inception variant without residual connections outperforms the original Inception model v3.

## 3.4. Conclusion

This paper tries to combine Inception models and ResNets. They firstly design a technical variation for Inception and then integrate it to a residual module. The merged model outperforms state of the art.

One of the interesting points of this paper was that they present a new and different claim about the very deep networks. He et al. argue in their paper (He et al., 2015) that residual connections are necessary for very deep convolutional networks to be converged. But, this paper does not support this view, although they restrict this claim for only image recognition problems.

# 4. Identity Mappings in Deep Residual Networks

## 4.1. Motivation

The advent of residual networks has motivated rigorous research on extremely deep networks. And this leads the author to focus on the characteristics of the flow of information through both shortcut connections and residual units and to find the best structures. They experiment various structures such as identity mappings as the skip connections and after-adding activation.

## 4.2. Model Description

The original residual networks performs the following computations:

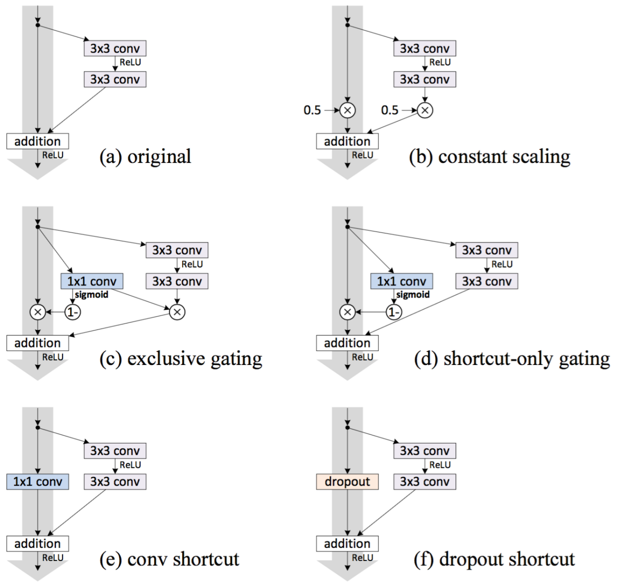
$$y_l = h(x_l) + F(x_l, W_l)$$

$$x_{l+1} = f(y_l)$$

This paper consider various types of mappings, $h(x_l)$ and activation functions, $F$.

### 4.2.1. Identity Skip Connections

The authors compare various types of shortcut connections as shown in Figure 5.



(a) original

(b) constant scaling

(c) exclusive gating

(d) shortcut-only gating

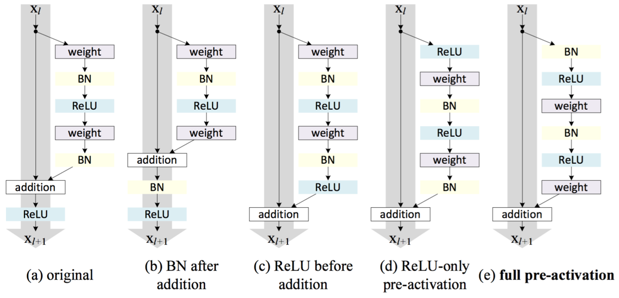
(e) conv shortcut

(f) dropout shortcut

*Figure 5.* Various types of shortcut connections

The original model is using identity mapping, $h(x_l) = x_l$. Other models have a shortcut of constant scaling, gatings, convolusion, or dropout. Among those models, the original model, identity mapping, shows the best performance and good convergence characteristics. This is because non-identity mappings on the shortcuts hamper information propagation and lead to optimization problems.

### 4.2.2. Activation Functions



(a) original

(b) BN after addition

(c) ReLU before addition

(d) ReLU-only pre-activation

(e) **full pre-activation**

*Figure 6.* Various types of activation

The authors try various combinations of residual submodules as shown in Figure 6.

A residual module consists of sub modules such as weight, batch normalization, and ReLu activation. A different combination shows different characteristics because of particular ways of controlling of information propagation. For example, if batch normalization(BN) is followed by addition, BN layer changes the signal that traverses through the shortcut and interfere information propagation.

The best performing combination of sub modules is full pre-activation (Figure 6.(e)). The advantage of pre-activation is twofold. Since it makes $f$ as an identity mapping, the optimization becomes easy. In addition, if we use BN as pre-activation, it regularizes the model efficiently.

## 4.3. Conclusion

This paper examines two components of a residual module, and finds the best form of a residual unit. For the shortcut, identity mapping is the best choice, and the full pre-activation should be used for the inside structure of a residual module.

## References

Bengio, Yoshua, Simard, Patrice, and Frasconi, Paolo. Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks*, 5(2): 157–166, 1994.

Dong, Chao, Loy, Chen Change, He, Kaiming, and Tang, Xiaoou. Learning a deep convolutional network for image super-resolution. In *European Conference on Computer Vision*, pp. 184–199. Springer, 2014.

Girshick, Ross, Donahue, Jeff, Darrell, Trevor, and Malik, Jitendra. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 580–587, 2014.

Glorot, Xavier and Bengio, Yoshua. Understanding the difficulty of training deep feedforward neural networks. In *Aistats*, volume 9, pp. 249–256, 2010.

He, Kaiming, Zhang, Xiangyu, Ren, Shaoqing, and Sun, Jian. Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385*, 2015.

Karpathy, Andrej, Toderici, George, Shetty, Sanketh, Leung, Thomas, Sukthankar, Rahul, and Fei-Fei, Li. Large-scale video classification with convolutional neural networks. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pp. 1725–1732, 2014.

Krizhevsky, Alex, Sutskever, Ilya, and Hinton, Geoffrey E. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pp. 1097–1105, 2012.

LeCun, Yann, Jackel, LD, Bottou, Leon, Brunot, A, Cortes, Corinna, Denker, JS, Drucker, Harris, Guyon, I, Muller, UA, Sackinger, Eduard, et al. Comparison of learning algorithms for handwritten digit recognition. In *International conference on artificial neural networks*, volume 60, pp. 53–60, 1995.

Long, Jonathan, Shelhamer, Evan, and Darrell, Trevor. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3431–3440, 2015.

Russakovsky, Olga, Deng, Jia, Su, Hao, Krause, Jonathan, Satheesh, Sanjeev, Ma, Sean, Huang, Zhiheng, Karpathy, Andrej, Khosla, Aditya, Bernstein, Michael, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015.

Severyn, Aliaksei and Moschitti, Alessandro. Twitter sentiment analysis with deep convolutional neural networks. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 959–962. ACM, 2015.

Simonyan, Karen and Zisserman, Andrew. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

Szegedy, Christian, Liu, Wei, Jia, Yangqing, Sermanet, Pierre, Reed, Scott, Anguelov, Dragomir, Erhan, Dumitru, Vanhoucke, Vincent, and Rabinovich, Andrew. Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–9, 2015a.

Szegedy, Christian, Vanhoucke, Vincent, Ioffe, Sergey, Shlens, Jonathon, and Wojna, Zbigniew. Rethinking the inception architecture for computer vision. *arXiv preprint arXiv:1512.00567*, 2015b.

Toshev, Alexander and Szegedy, Christian. Deeppose: Human pose estimation via deep neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1653–1660, 2014.

Wang, Naiyan and Yeung, Dit-Yan. Learning a deep compact image representation for visual tracking. In *Advances in neural information processing systems*, pp. 809–817, 2013.

Zeiler, Matthew D and Fergus, Rob. Visualizing and understanding convolutional networks. In *European Conference on Computer Vision*, pp. 818–833. Springer, 2014.