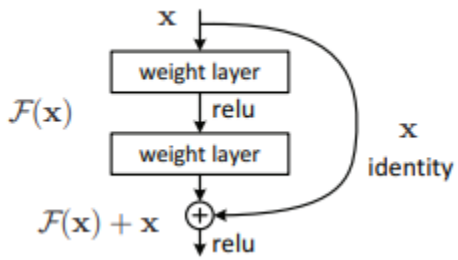
# ResNet

## Introduction:-

ResNet addresses the issue of optimisation of very deep neural networks and the problem of vanishing gradients by introduction of residual connections. It is inspired by VGG. Instead of simply stacking up layers and making them learn the desired mapping, it proposed explicitly making the layers learn a residual mapping.

It was hypothesized that it's easier to optimize the residual mapping than to optimize the original, unreferenced mapping. 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. If the added layers can learn identity mapping, then the model shouldn't have a higher error than its shallower versions. If the optimal function is closer to an identity mapping than to a zero mapping, it should be easier for the solver to find the perturbations with reference to an identity mapping, than to learn the function as a new one.

## Identity Mapping and Shortcut Connections:-




In the above image, x denotes the activations of a layer and they are added to the output of the subsequent two stacked layers. This connection was referred to as 'shortcut connection' and they neither add any extra parameter nor computational complexity.

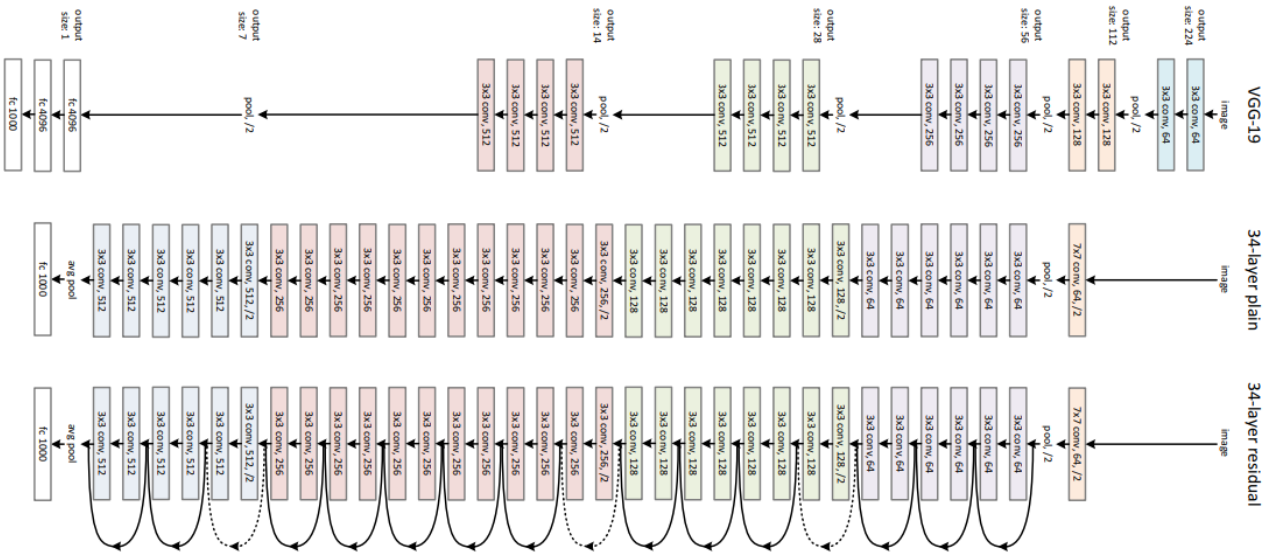
Let's assume x to be the activations of a layer 'l'. Hence the the activations of the layer l+1 will be σ(W[l+1] * x) (omitting biases for simplicity) where W[l+1] denote the associated weights and σ denotes the activation function i.e. ReLU. For the next layer, if they were simply stacked up, the activations would have been σ(W[l+2](σ(W[l+1] * x))). But since we have 'shortcut connections', the actual activations are σ(W[l+2](σ(W[l+1] * x)) + x). The assumption here is that the activations of the layer l and of simply stacked up layer l+2 have the same shape. In case, if they don't, x is linearly projected by multiplication with a Ws matrix to have the same shape for element wise addition. There are two ways of matching the dimensions:-

a. Adding zero padding (no extra parameters)
b. Using 1x1 convolutions (extra parameters)

Note: Channel wise element addition takes place in these residual connections

## Network Architectures:-

For getting inferences on improvement due to residual connections, the researchers made their observations on a plain network and then on a residual network. The architecture for VGG, 34-layer plain network and 34-layer residual network is given below.




A striking feature was that instead of using fully connected layers, a global average pooling layer was used which was followed by 1000 classes of softmax layer.

Following are the architectures of deeper versions of ResNet along with their shallow versions:-

| layer name | output size | 18-layer | 34-layer | 50-layer | 101-layer | 152-layer |
|---|---|---|---|---|---|---|
| conv1 | 112×112 | 7×7, 64, stride 2 | | | | |
| conv2_x | 56×56 | 3×3 max pool, stride 2 | | | | |
| | | $\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times2$ | $\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times3$ | $\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$ | $\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$ | $\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$ |
| conv3_x | 28×28 | $\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times2$ | $\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times4$ | $\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$ | $\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$ | $\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times8$ |
| conv4_x | 14×14 | $\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times2$ | $\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times6$ | $\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times6$ | $\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times23$ | $\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times36$ |
| conv5_x | 7×7 | $\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times2$ | $\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times3$ | $\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$ | $\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$ | $\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$ |
| | 1×1 | average pool, 1000-d fc, softmax | | | | |
| FLOPs | | $1.8\times10^9$ | $3.6\times10^9$ | $3.8\times10^9$ | $7.6\times10^9$ | $11.3\times10^9$ |

## Implementation:-

For ImageNet, the image was resized with a shorter side randomly sampled in [256,480] for scale augmentation. A 224×224 crop is randomly sampled from an image or its horizontal flip, with the per-pixel mean subtracted. The standard color augmentation, similar to AlexNet, is

used. Batch normalization was applied after the convolution layer and then the resulting values were passed to the activation function. SGD with a mini batch size of 256 was used for updating the weights.

Hyperparameters: LR = 0.1 (divided by 10 when error plateaus), Iterations = 6*10e5, Weight Decay = 10e-4, Momentum = 0.9

## Experiments:-

| | plain | ResNet |
|---|---|---|
| 18 layers | 27.94 | 27.88 |
| 34 layers | 28.54 | **25.03** |

An interesting fact is that even though the 18-layer Plain Network and ResNet have similar accuracy but the ResNet version converges faster indicating that ResNet eases the optimization by providing convergence at an earlier stage.

## Deeper Bottleneck Architectures:-

Due to training time limitations, we use the following bottleneck design. Instead of using a stack of 2 layers, we use a stack of 3 layers comprising of 1x1, 3x3, and 1x1 convolutions where 1x1 convolutions are responsible for reducing and increasing the dimensions respectively, leaving the 3x3 layer as bottleneck.

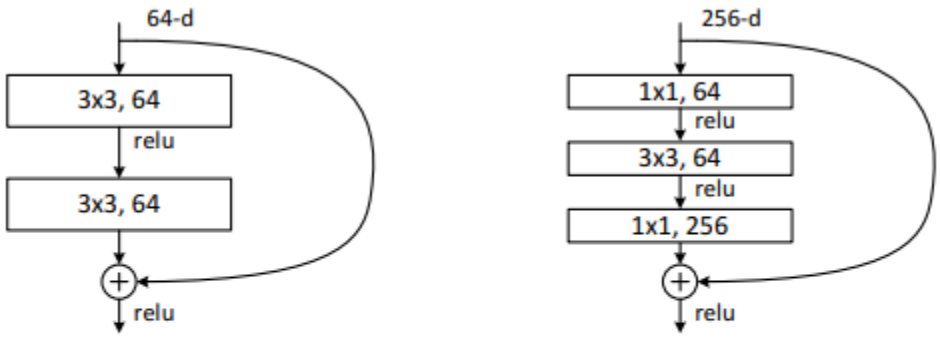



Figure 5. A deeper residual function $\mathcal{F}$ for ImageNet. Left: a building block (on 56×56 feature maps) as in Fig. 3 for ResNet-34. Right: a "bottleneck" building block for ResNet-50/101/152.

## Results Obtained:-

1st place on the tasks of ImageNet detection, ImageNet localization, COCO detection, and COCO segmentation.

1st place in ILSVRC 2015

| training data | COCO train | | COCO trainval | |
|---|---|---|---|---|
| test data | COCO val | | COCO test-dev | |
| mAP | @.5 | @[.5, .95] | @.5 | @[.5, .95] |
| baseline Faster R-CNN (VGG-16) | 41.5 | 21.2 | | |
| baseline Faster R-CNN (ResNet-101) | 48.4 | 27.2 | | |
| +box refinement | 49.9 | 29.9 | | |
| +context | 51.1 | 30.0 | 53.3 | 32.2 |
| +multi-scale testing | 53.8 | 32.5 | **55.7** | **34.9** |
| ensemble | | | **59.0** | **37.4** |

Table 9. Object detection improvements on MS COCO using Faster R-CNN and ResNet-101.

| system | net | data | mAP | areo | bike | bird | boat | bottle | bus | car | cat | chair | cow | table | dog | horse | mbike | person | plant | sheep | sofa | train | tv |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| baseline | VGG-16 | 07+12 | 73.2 | 76.5 | 79.0 | 70.9 | 65.5 | 52.1 | 83.1 | 84.7 | 86.4 | 52.0 | 81.9 | 65.7 | 84.8 | 84.6 | 77.5 | 76.7 | 38.8 | 73.6 | 73.9 | 83.0 | 72.6 |
| baseline | ResNet-101 | 07+12 | 76.4 | 79.8 | 80.7 | 76.2 | 68.3 | 55.9 | 85.1 | 85.3 | **89.8** | 56.7 | 87.8 | 69.4 | 88.3 | 88.9 | 80.9 | 78.4 | 41.7 | 78.6 | 79.8 | 85.3 | 72.0 |
| baseline+++ | ResNet-101 | COCO+07+12 | **85.6** | **90.0** | **89.6** | **87.8** | **80.8** | **76.1** | **89.9** | **89.9** | 89.6 | **75.5** | **90.0** | **80.7** | **89.6** | **90.3** | **89.1** | **88.7** | **65.4** | **88.1** | **85.6** | **89.0** | **86.8** |

Table 10. Detection results on the PASCAL VOC 2007 test set. The baseline is the Faster R-CNN system. The system "baseline+++" include box refinement, context, and multi-scale testing in Table 9.

| system | net | data | mAP | areo | bike | bird | boat | bottle | bus | car | cat | chair | cow | table | dog | horse | mbike | person | plant | sheep | sofa | train | tv |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| baseline | VGG-16 | 07++12 | 70.4 | 84.9 | 79.8 | 74.3 | 53.9 | 49.8 | 77.5 | 75.9 | 88.5 | 45.6 | 77.1 | 55.3 | 86.9 | 81.7 | 80.9 | 79.6 | 40.1 | 72.6 | 60.9 | 81.2 | 61.5 |
| baseline | ResNet-101 | 07++12 | 73.8 | 86.5 | 81.6 | 77.2 | 58.0 | 51.0 | 78.6 | 76.6 | 93.2 | 48.6 | 80.4 | 59.0 | 92.1 | 85.3 | 84.8 | 80.7 | 48.1 | 77.3 | 66.5 | 84.7 | 65.6 |
| baseline+++ | ResNet-101 | COCO+07++12 | **83.8** | **92.1** | **88.4** | **84.8** | **75.9** | **71.4** | **86.3** | **87.8** | **94.2** | **66.8** | **89.4** | **69.2** | **93.9** | **91.9** | **90.9** | **89.6** | **67.9** | **88.2** | **76.8** | **90.3** | **80.0** |

Table 11. Detection results on the PASCAL VOC 2012 test set (`http://host.robots.ox.ac.uk:8080/leaderboard/displaylb.php?challengeid=11&compid=4`). The baseline is the Faster R-CNN system. The system "baseline+++" include box refinement, context, and multi-scale testing in Table 9.