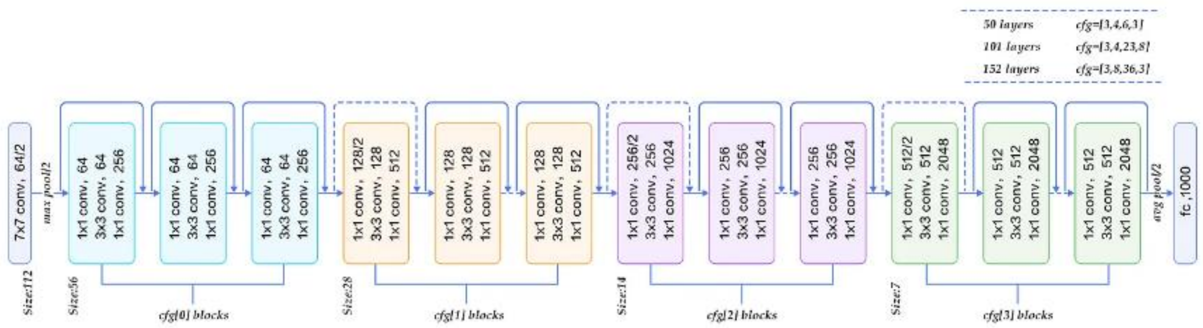
Resnet50 Architecture

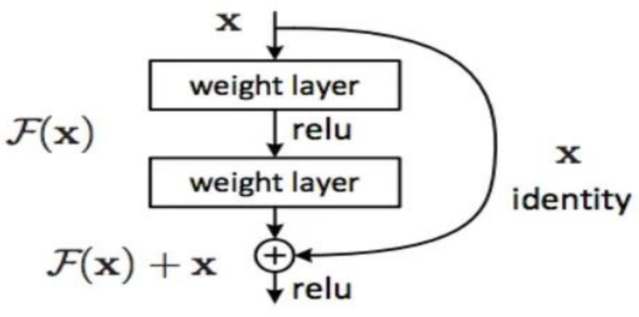
ResNet-50 is a convolutional neural network that is 50 layers deep. ResNet, short for Residual Networks is a classic neural network used as a backbone for many computer vision tasks. The fundamental breakthrough with ResNet was it allowed us to train extremely deep neural networks with 150+layers. It is an innovative neural network that was first introduced by **Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun** in their 2015 computer vision research paper titled **'Deep Residual Learning for Image Recognition'**.

Convolutional Neural Networks have a major disadvantage — '**Vanishing Gradient Problem**'. During backpropagation, the value of gradient decreases significantly, thus hardly any change comes to weights. To overcome this, ResNet is used. It make use of **"SKIP CONNECTION".**

ResNet-50 Architecture



Skip Connection — *Adding the original input to the output of the convolutional block.*


x
weight layer
relu
weight layer
$\mathcal{F}(\mathbf{x})$
x
identity
$\mathcal{F}(\mathbf{x}) + \mathbf{x}$
relu

Skip Connection

All algorithms train on the output 'Y' but, ResNet trains on F(X). In simpler words, ResNet tries to make F(X)=0 so that Y=X.

All algorithms train on the output ‘Y’ but, ResNet trains on F(X). In simpler words, ResNet tries to make F(X)=0 so that Y=X.

SKIP CONNECTION is a direct connection that skips over some layers of the model. The output is not the same due to this skip connection. Without the skip connection, input ‘X gets multiplied by the weights of the layer followed by adding a bias term.

Then comes the activation function, F() and we get the output as :

F(w*x + b) (=F(X))

But with skip connection technique, the output is :

F(X)+x

In ResNet-50, there are two kinds of blocks –

- Identity Block
- Convolutional Block

The value of ‘x’ is added to the output layer if and only if the

$$input\ size == output\ size$$

If this is not the case, we add a ‘convolutional block’ in the shortcut path to make the input size equal to output size.

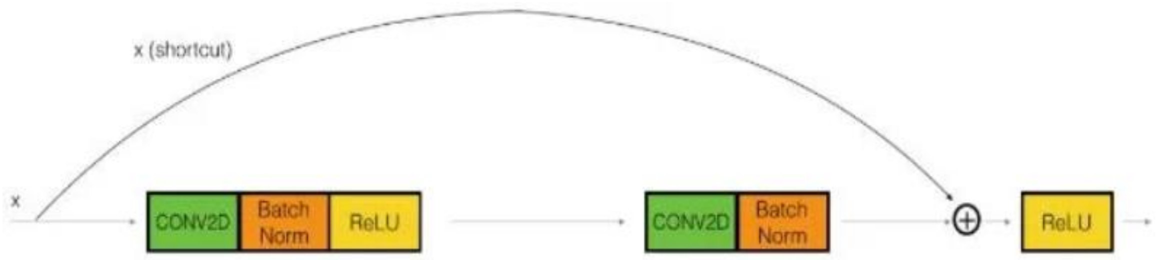


Figure 3 : **Identity block.** Skip connection "skips over" 2 layers.

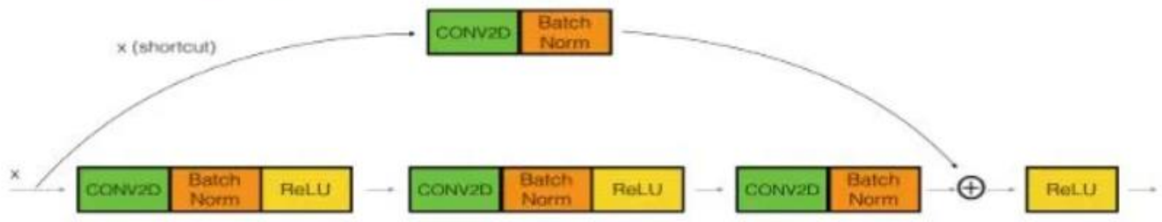


Figure 4 : **Convolutional block**

There are 2 ways to make the input size equal to the output size -

- Padding the input volume
- Performing 1*1 convolutions.

Size of output layer is calculated using —

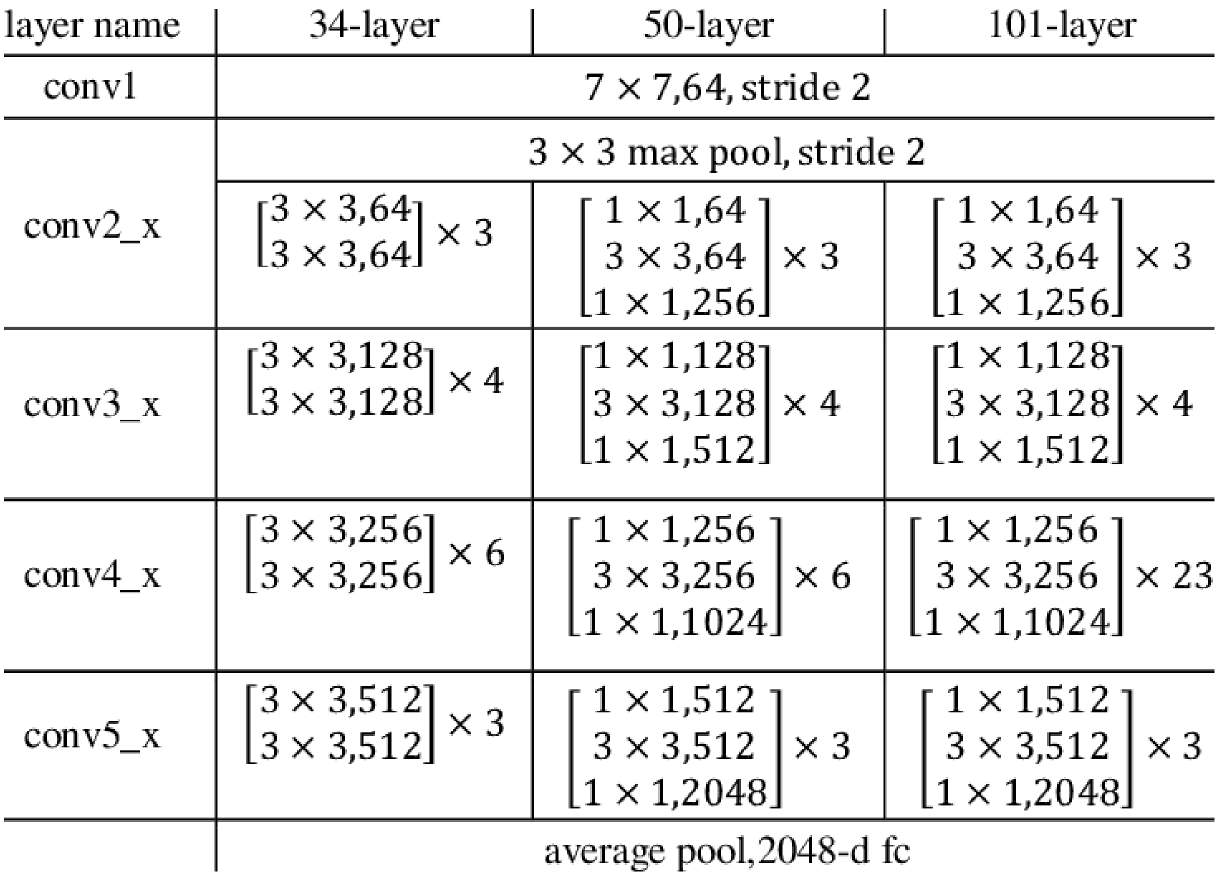
$$[\{(n+2p-f)/s\}+1]^2$$

where,
n= input image size,
p=padding,
s=stride,
f=number of filters.
For, 1*1 convolutional layers, size of output layer =

$$(n/2)*(n/2)$$

given the input size is ‘n’.

In CNNs, to reduce the size of the image, pooling is used. Here, we make use of stride=2 instead.



layer name	34-layer	50-layer	101-layer
conv1	$7 \times 7, 64$, stride 2		
conv2_x	3×3 max pool, stride 2		
	$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$
conv3_x	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$
conv4_x	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 23$
conv5_x	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$
	average pool,2048-d fc		

References

1-) https://blog.devgenius.io/resnet50-6b42934db431