ResNet-34

Analytics Vidhya

ResNet-34

- Proposed in 2015

Deep Residual Learning for Image Recognition

Kaiming He Xiangyu Zhang Shaoqing Ren Jian Sun
Microsoft Research
{kahe, v-xiangz, v-shren, jiansun}@microsoft.com

Abstract

Deeper neural networks are more difficult to train. We present a residual learning framework to ease the training of networks that are substantially deeper than those used previously. We explicitly reformulate the layers as learning residual functions with reference to the layer inputs, instead of learning unreferenced functions. We provide comprehensive empirical evidence showing that these residual networks are easier to optimize, and can gain accuracy from considerably increased depth. On the ImageNet dataset we

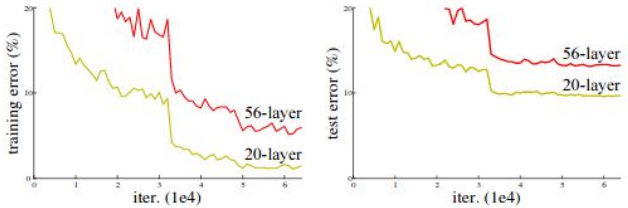


Figure 1. Training error (left) and test error (right) on CIFAR-10 with 20-layer and 56-layer "plain" networks. The deeper network has higher training error, and thus test error. Similar phenomena on ImageNet is presented in Fig. 4.

Analytics Vidhya

ResNet-34

- Proposed in 2015
- It has 34 layers with learnable parameters

ResNet-34

- Proposed in 2015
- It has 34 layers with learnable parameters
- Architecture details:
  - 16 residual blocks

ResNet-34

- Proposed in 2015
- It has 34 layers with learnable parameters
- Architecture details:
  - 16 residual blocks
  - Mostly 3 X 3 filters are used

ResNet-34

- Proposed in 2015
- It has 34 layers with learnable parameters
- Architecture details:
  - 16 residual blocks
  - Mostly 3 X 3 filters are used
  - Batch normalization layer is used after each convolution operation

Analytics Vidhya

ResNet-34

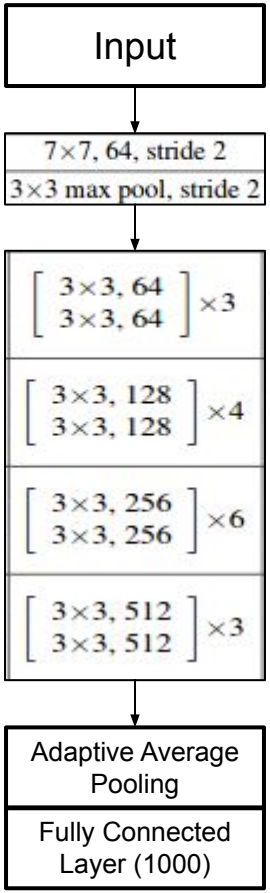
- Proposed in 2015
- It has 34 layers with learnable parameters
- Architecture details:
  - 16 residual blocks
  - Mostly 3 X 3 filters are used
  - Batch normalization layer is used after each convolution operation
  - Activation function: ReLU

ResNet-34

- Proposed in 2015
- It has 34 layers with learnable parameters
- Architecture details:
  - 16 residual blocks
  - Mostly 3 X 3 filters are used
  - Batch normalization layer is used after each convolution operation
  - Activation function: ReLU
- Trained on imagenet dataset

Analytics Vidhya

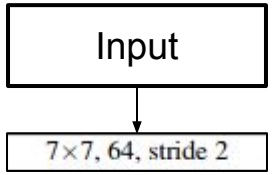
Architecture: ResNet-34



Architecture: ResNet-34

224, 224, 3

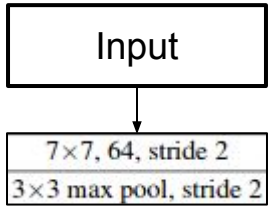
Architecture: ResNet-34



224, 224, 3

112, 112, 64

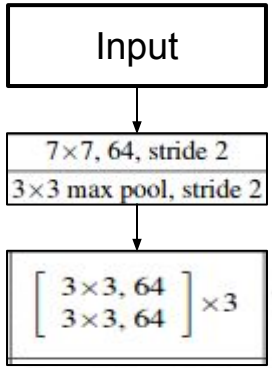
Architecture: ResNet-34



224, 224, 3

112, 112, 64
56, 56, 64

Architecture: ResNet-34



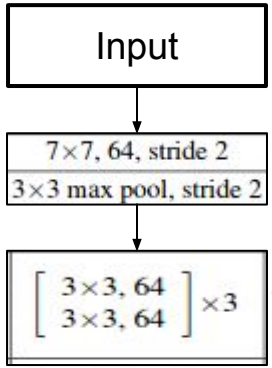
224, 224, 3

112, 112, 64
56, 56, 64

56, 56, 64

Architecture: ResNet-34

Input
(56, 56, 64)



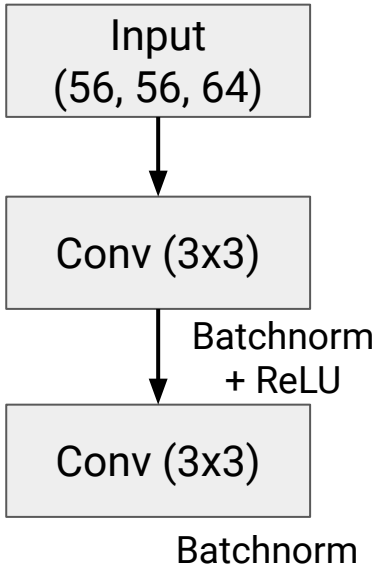
224, 224, 3

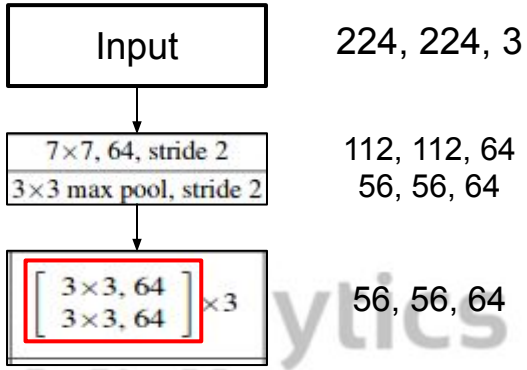
112, 112, 64

56, 56, 64

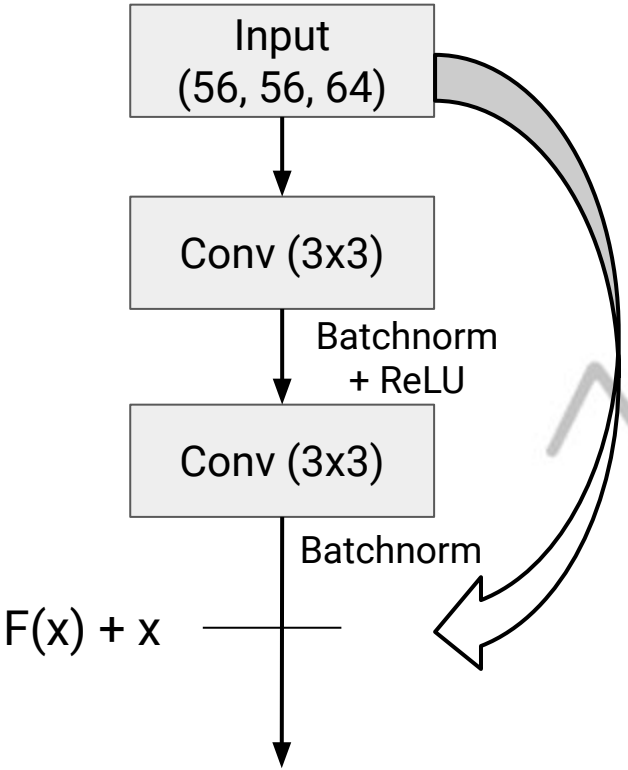
56, 56, 64

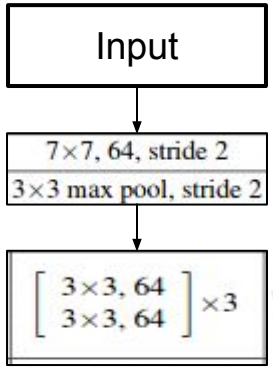
Architecture: ResNet-34





Architecture: ResNet-34



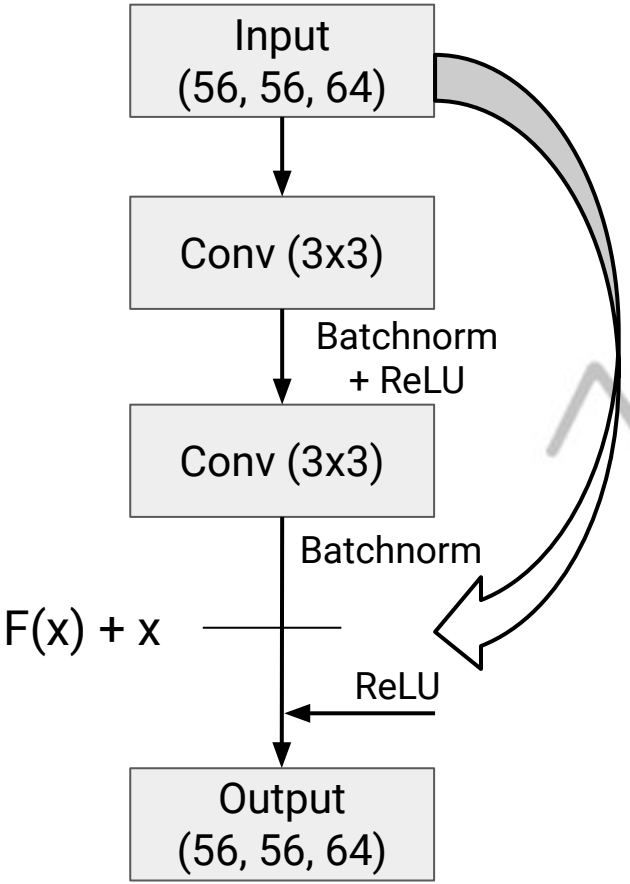


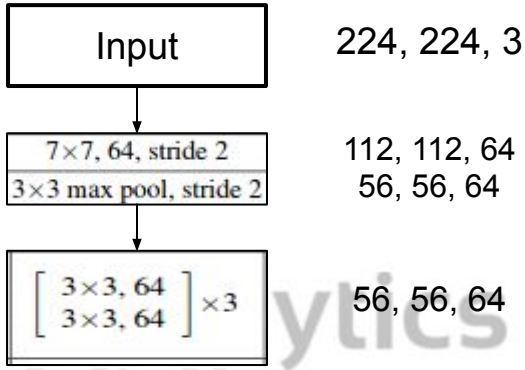
224, 224, 3

112, 112, 64
56, 56, 64

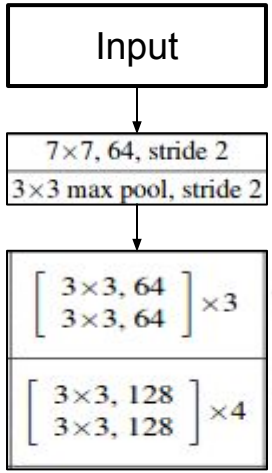
56, 56, 64

Architecture: ResNet-34





Architecture: ResNet-34



224, 224, 3

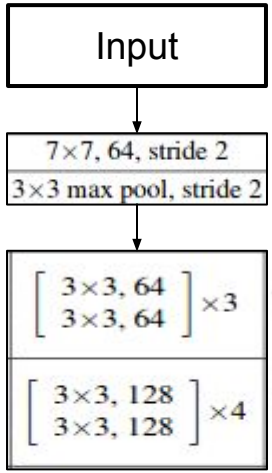
112, 112, 64
56, 56, 64

56, 56, 64

28, 28, 128

Architecture: ResNet-34

Input
(56, 56, 64)



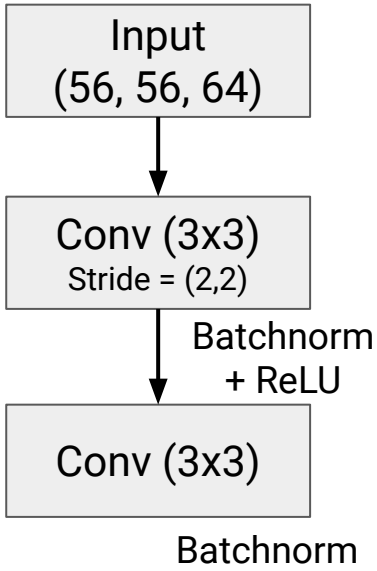
224, 224, 3

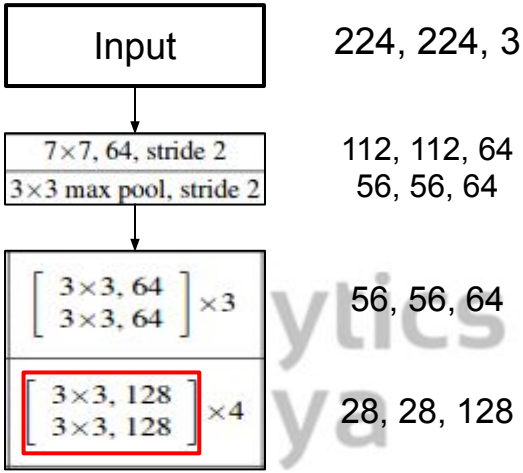
112, 112, 64
56, 56, 64

56, 56, 64

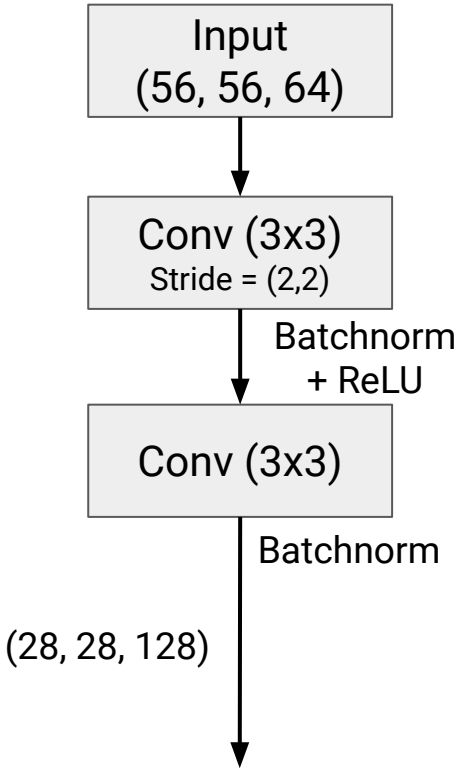
28, 28, 128

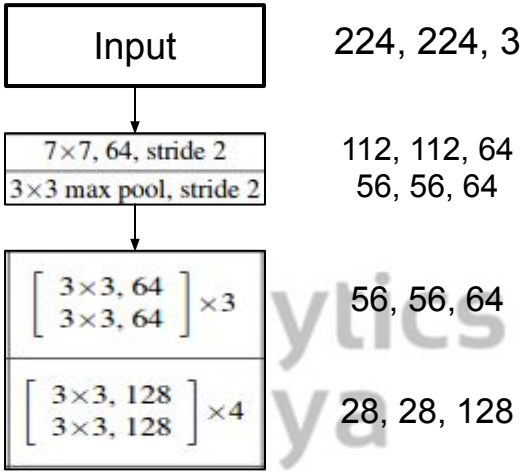
Architecture: ResNet-34



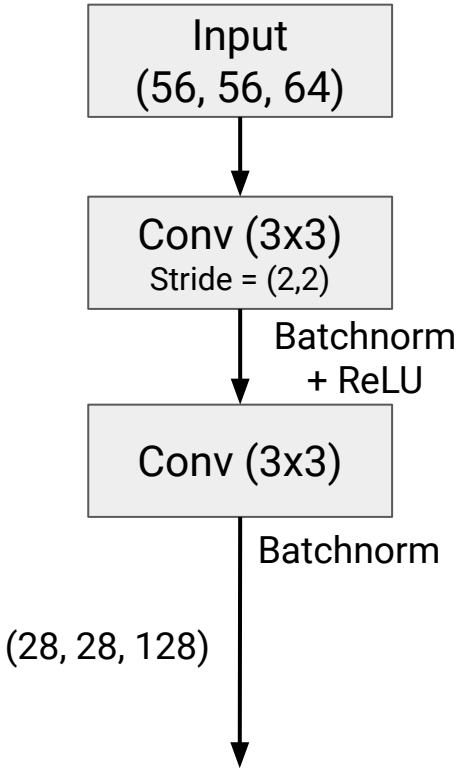


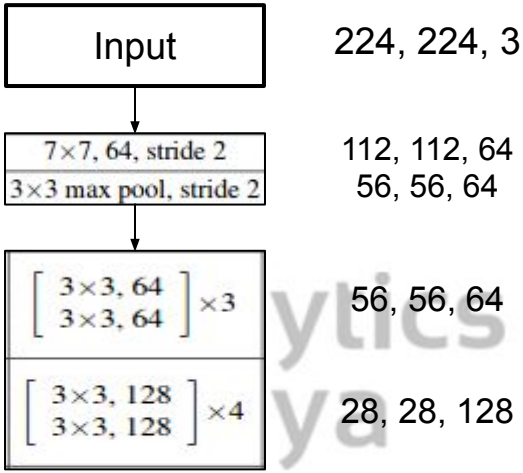
Architecture: ResNet-34





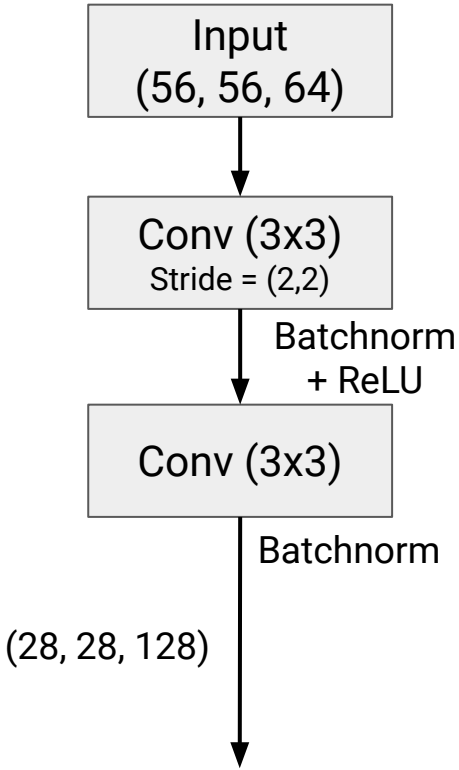
Architecture: ResNet-34

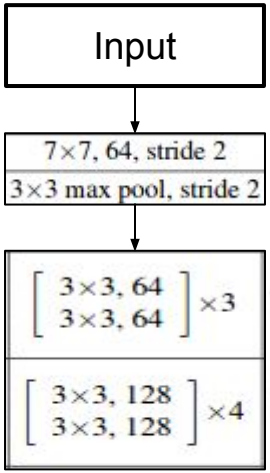




F(x) + x

Architecture: ResNet-34





224, 224, 3

112, 112, 64
56, 56, 64

56, 56, 64

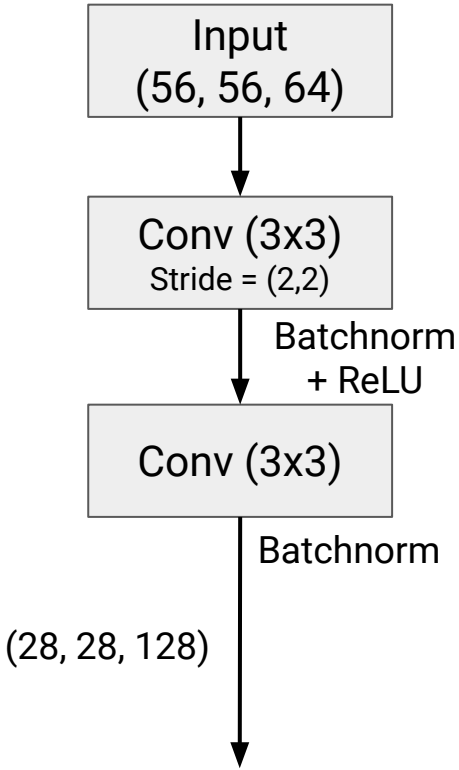
28, 28, 128

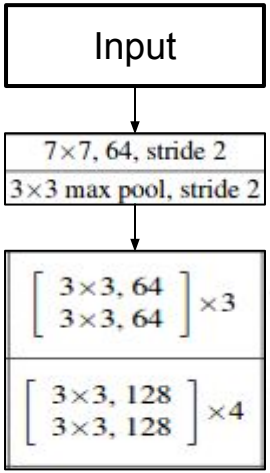
F(x) + x

F(x): (28, 28, 128)

x: (56, 56, 64)

Architecture: ResNet-34





224, 224, 3

112, 112, 64
56, 56, 64

56, 56, 64

28, 28, 128

$F(x) + W_s x$

F(x): (28, 28, 128)

x: (56, 56, 64)

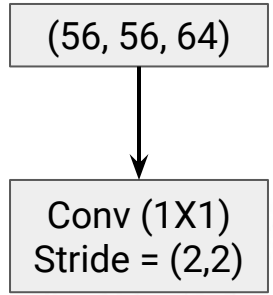
Architecture: ResNet-34

(56, 56, 64)

$F(x) + W_s x$

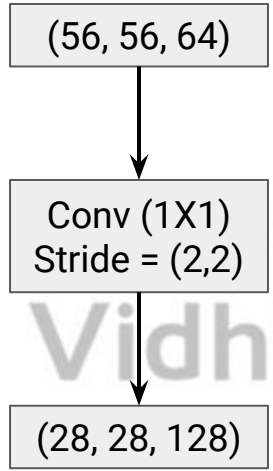
Architecture: ResNet-34

$F(x) + W_s x$

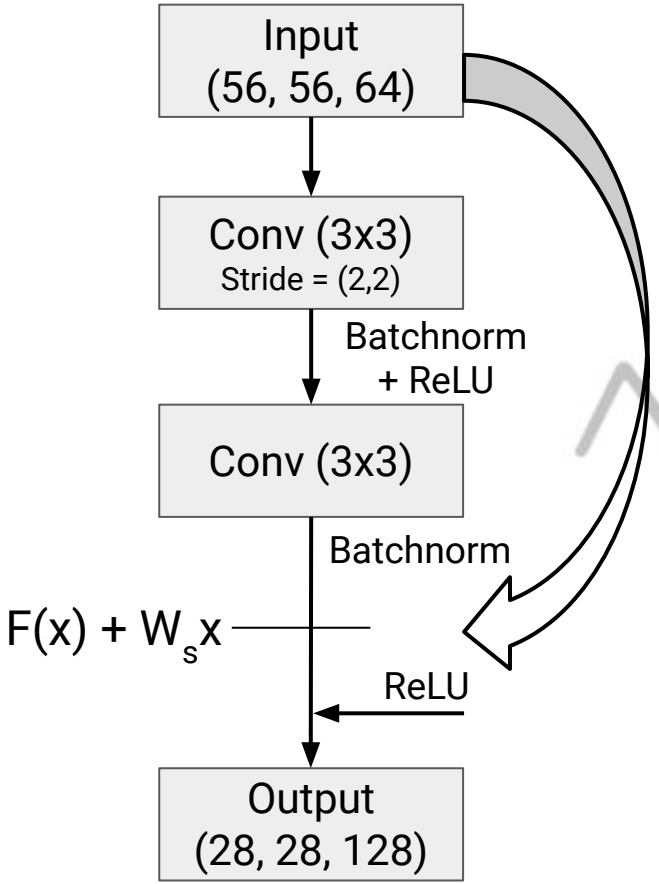


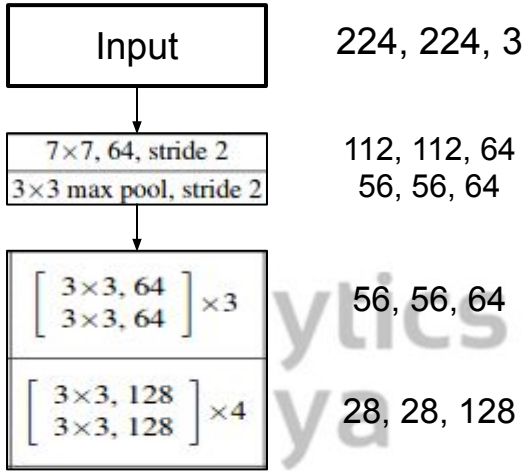
Architecture: ResNet-34

$F(x) + W_s x$

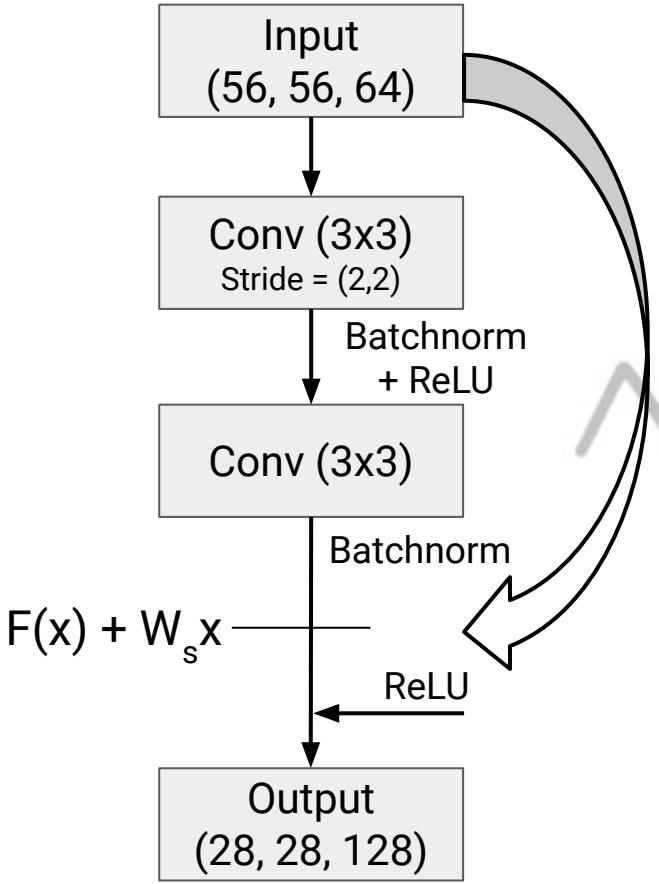


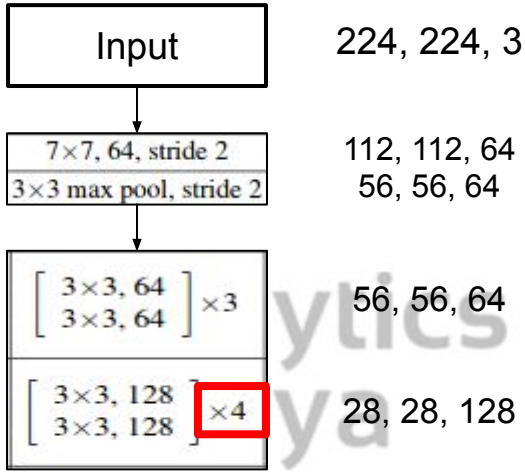
Architecture: ResNet-34





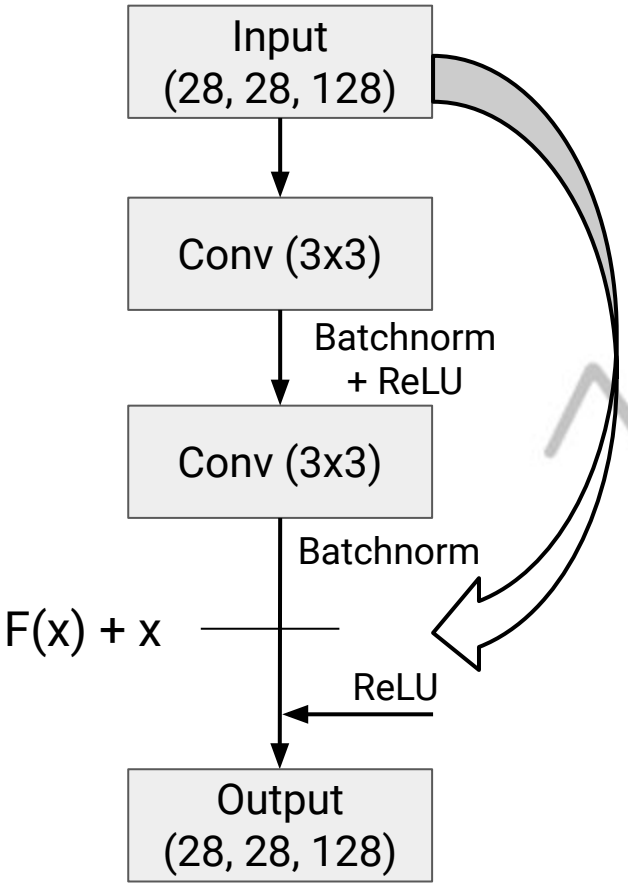
Architecture: ResNet-34

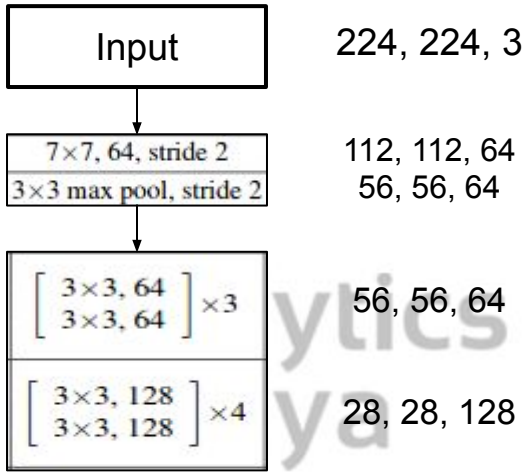




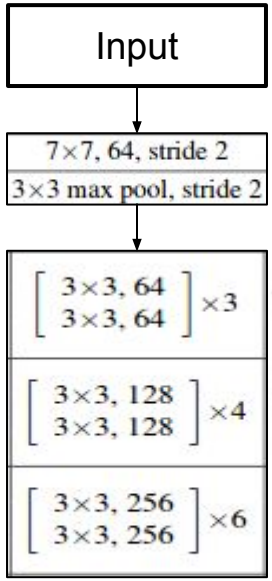
Analytics Vidhya

Architecture: ResNet-34





Architecture: ResNet-34



224, 224, 3

112, 112, 64

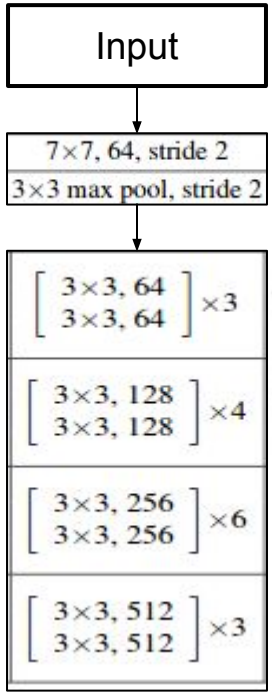
56, 56, 64

56, 56, 64

28, 28, 128

14, 14, 256

Architecture: ResNet-34



224, 224, 3

112, 112, 64
56, 56, 64

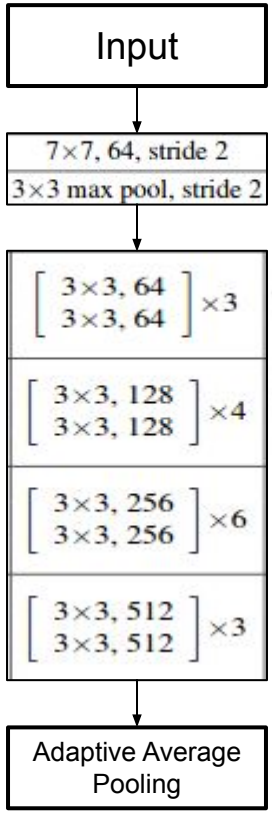
56, 56, 64

28, 28, 128

14, 14, 256

7, 7, 512

Architecture: ResNet-34



224, 224, 3

112, 112, 64
56, 56, 64

56, 56, 64

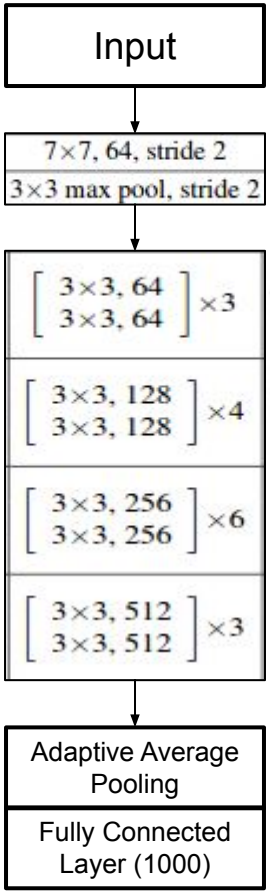
28, 28, 128

14, 14, 256

7, 7, 512

1, 1, 512

Architecture: ResNet-34



224, 224, 3

112, 112, 64
56, 56, 64

56, 56, 64

28, 28, 128

14, 14, 256

7, 7, 512

1, 1, 512

Summary: ResNet-34

Summary: ResNet-34

- It has 34 layers with learnable parameters

Summary: ResNet-34

- It has 34 layers with learnable parameters
- Takes RGB image as input

Summary: ResNet-34

- It has 34 layers with learnable parameters
- Takes RGB image as input
- Architecture details:
  - 16 residual blocks
  - Mostly 3 X 3 filters are used
  - Batch normalization layer is used after each convolution operation

Thank You

Analytics
Vidhya