CS60010: Deep Learning

Spring 2023

Sudeshna Sarkar

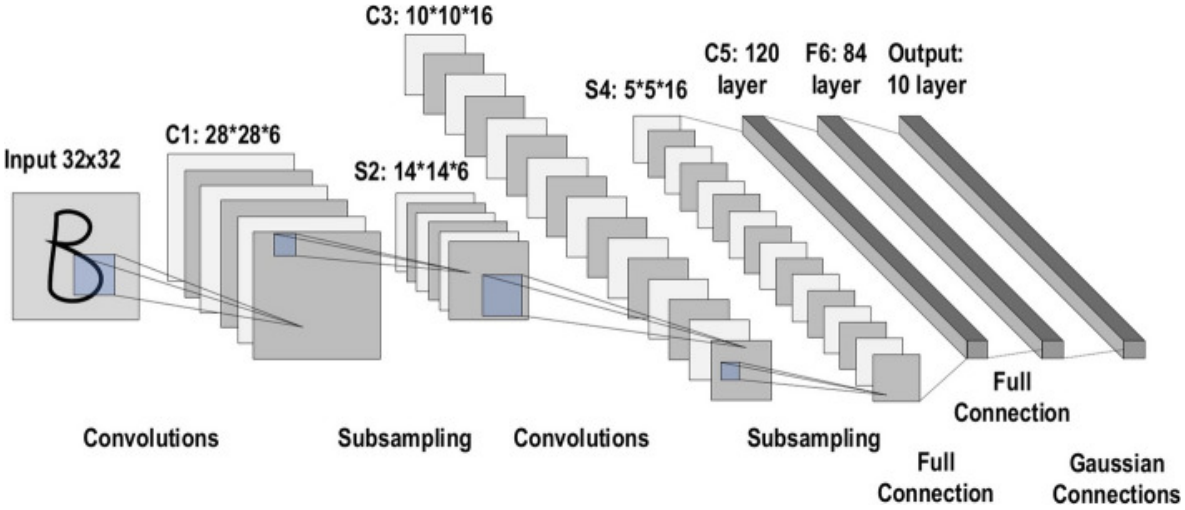
CNN Architectures

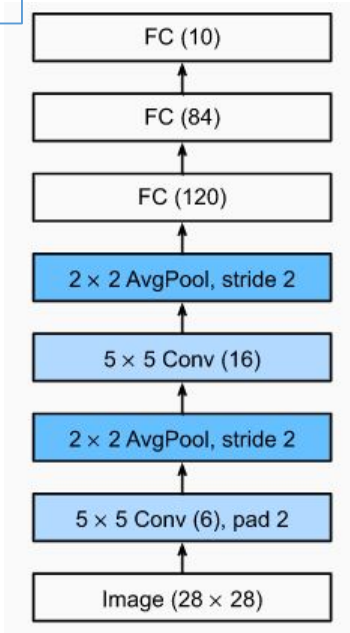
9 Feb 2023

LeNet

The history of deep CNNs began with the appearance of LeNet (handwritten character recognition)

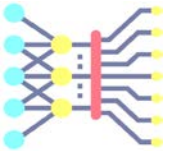
Trained on MNIST digit dataset with 60K training examples





LeCun Y, Jackel LD, Bottou L, Cortes C, Denker JS, Drucker H, Guyon I, Muller UA, Sackinger E, Simard P, et al. Learning algorithms for classification: a comparison on handwritten digit recognition. *Neural Netw Stat Mech Perspect.* 1995;261:276. [

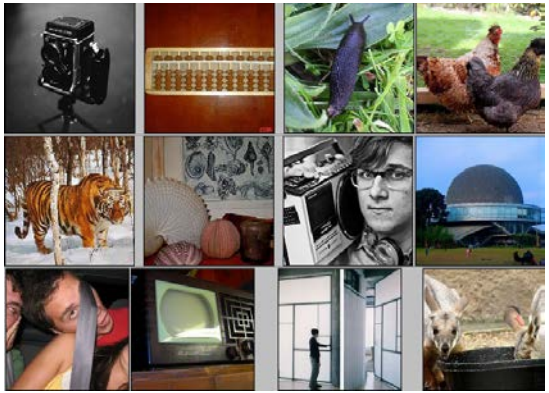
A tour of modern CNN architectures



- AlexNet: the first large-scale network deployed to beat conventional computer vision methods on a large-scale vision challenge
- VGG network, which makes use of a number of repeating blocks of elements
- network in network (NiN) which convolves whole neural networks patch-wise over inputs;
- GoogLeNet, which uses networks with parallel concatenations;
- residual networks (ResNet), which remain the most popular off-the-shelf architecture in computer vision;
- densely connected networks (DenseNet), which are expensive to compute but have set some recent benchmarks.

Each of these networks was briefly a dominant architecture and many were winners or runners-up in the ImageNet competition, which has served as a barometer of progress on supervised learning in computer vision since 2010.

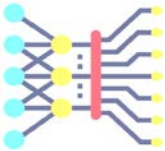
IMAGENET



- ~14 million labeled images, 20k classes
- Images gathered from Internet
- Human labels via Amazon MTurk
- ImageNet Large-Scale Visual Recognition Challenge (ILSVRC):
- 1.2 million training images, 1000 classes
- 100k test

ILSVRC: Imagenet Large Scale Visual Recognition Challenge

www.image-net.org/challenges/LSVRC/



ILSVRC:

Imagenet Large Scale Visual Recognition Challenge

[Russakovsky et al 2014]

The Problem: Classification

Classify an image into 1000 possible classes:

e.g. Abyssinian cat, Bulldog, French Terrier, Cormorant, Chickadee, red fox, banjo, barbell, hourglass, knot, maze, viaduct, etc.

cat, tabby cat (0.71)
Egyptian cat (0.22)
red fox (0.11)
…..

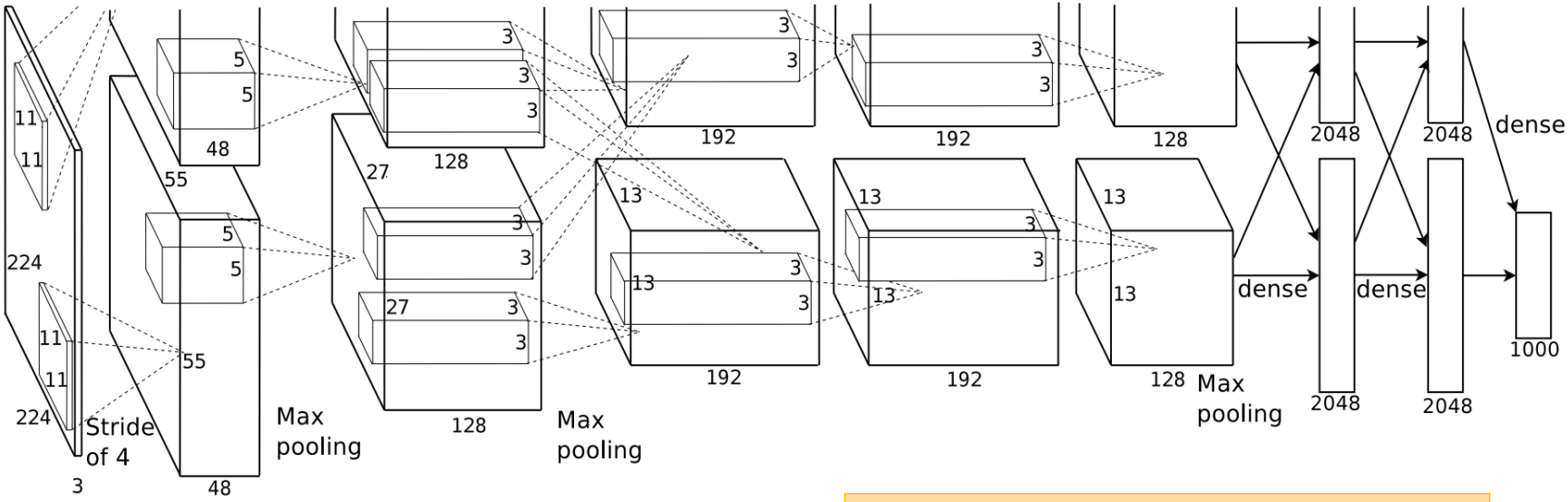
The Evaluation Metric: Top K-error

True label: Abyssinian cat

Top-1 error: 1.0	Top-1 accuracy: 0.0
Top-2 error: 1.0	Top-2 accuracy: 0.0
Top-3 error: 1.0	Top-3 accuracy: 0.0
Top-4 error: 0.0	Top-4 accuracy: 1.0
Top-5 error: 0.0	Top-5 accuracy: 1.0

cat, tabby cat (0.61)
Egyptian cat (0.22)
red fox (0.11)
Abyssinian cat (0.10)
French terrier (0.03)
.....

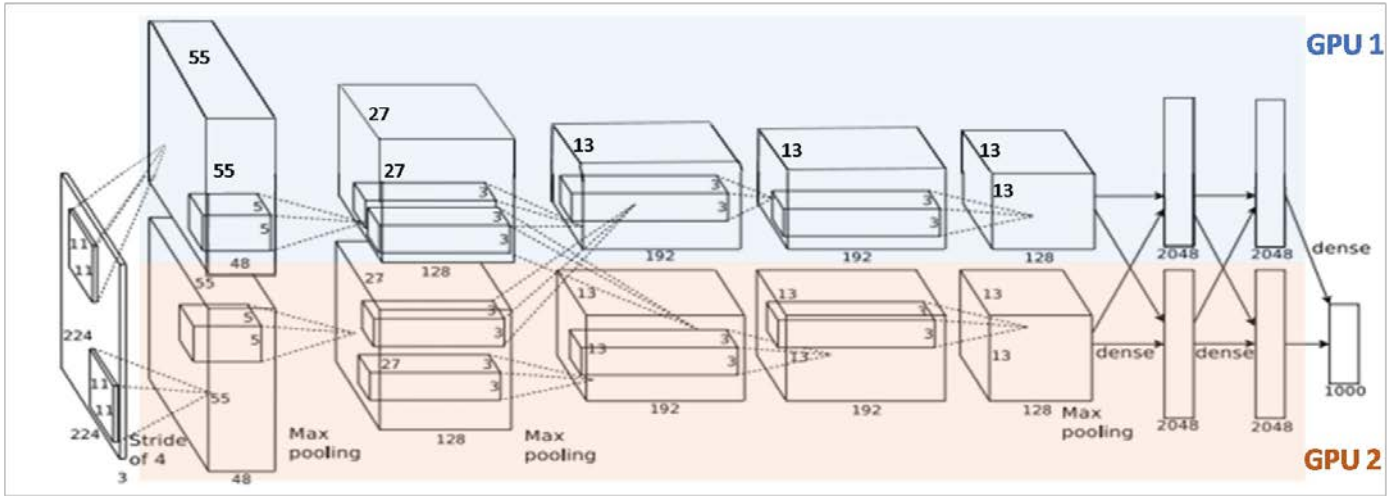
Alexnet: ILSVRC 2012 winner

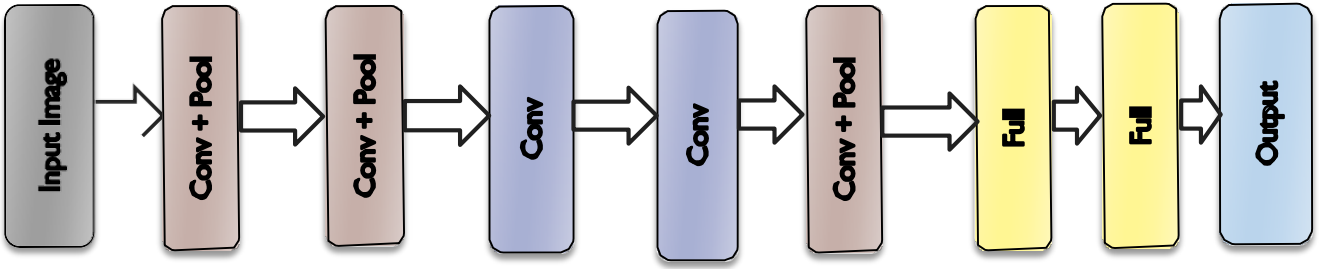


ImageNet Classification Task:

Previous Best : ~25% (CVPR-2011)
AlexNet : ~15 % (NIPS-2012)

Figure 2: An illustration of the architecture of our CNN, e ilities between the two GPUs. One GPU runs the layer-parts at the -parts at the bottom. The GPUs communicate only at certain layer l, and the number of neurons in the network's remaining layers is given by 253,440–186,624–64,896–64,896–43,264–4096–4096–1000.

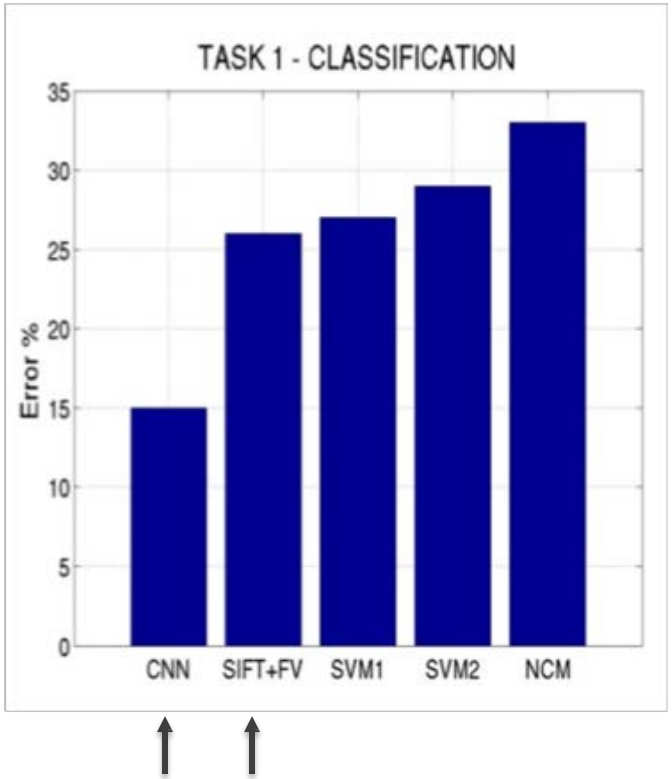

GPU 1
GPU 2
55
55
27
27
13
13
13
13
48
128
192
192
128
2048
2048
dense
224
224
Stride of 4
3
48
128
192
192
128
2048
2048
1000
Max pooling
Max pooling
Max pooling
dense
dense


Input Image
Conv + Pool
Conv + Pool
Conv
Conv
Conv + Pool
Full
Full
Output

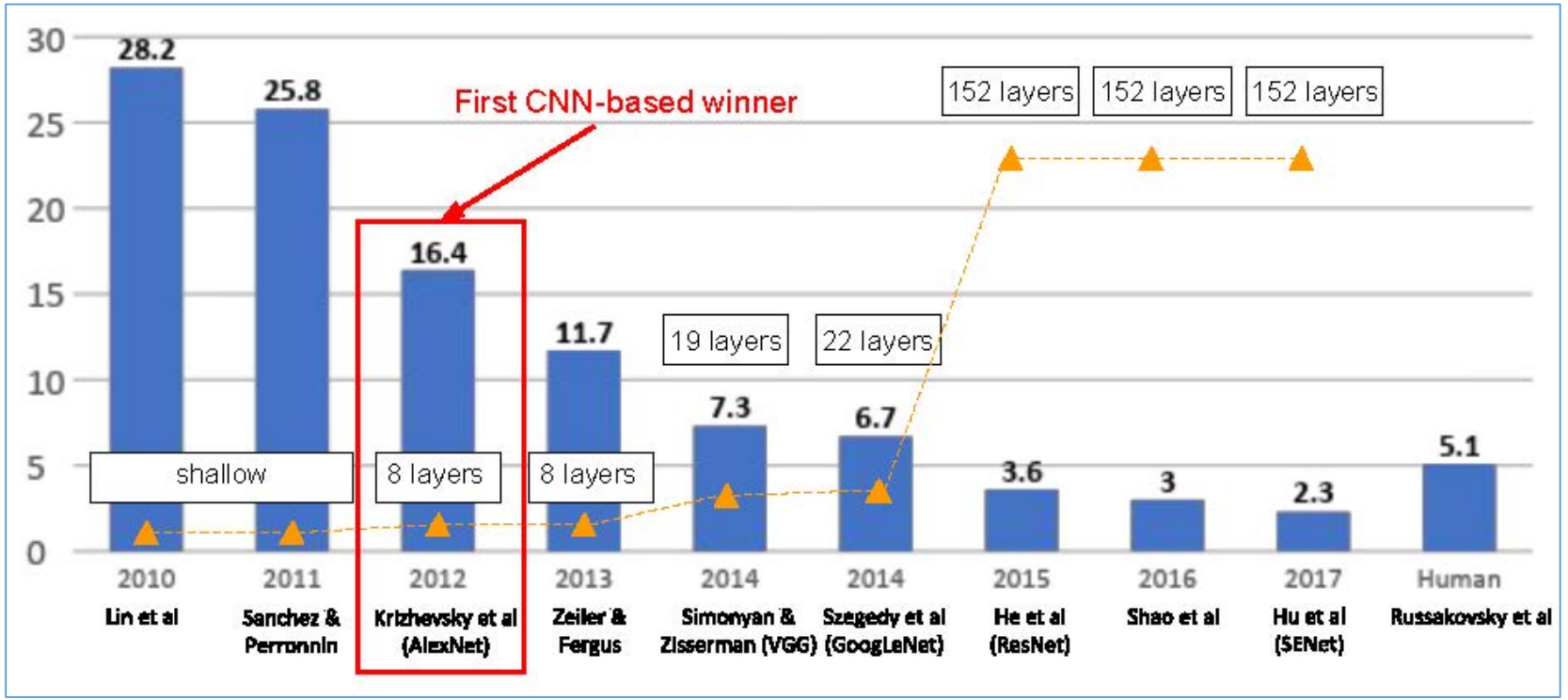
AlexNet Network - Structural Details												
Input			Output			Layer	Stride	Pad	Kernel size		in	out	# of Param
227	227	3	55	55	96	conv1	4	0	11	11	3	96	34944
55	55	96	27	27	96	maxpool1	2	0	3	3	96	96	0
27	27	96	27	27	256	conv2	1	2	5	5	96	256	614656
27	27	256	13	13	256	maxpool2	2	0	3	3	256	256	0
13	13	256	13	13	384	conv3	1	1	3	3	256	384	885120
13	13	384	13	13	384	conv4	1	1	3	3	384	384	1327488
13	13	384	13	13	256	conv5	1	1	3	3	384	256	884992
13	13	256	6	6	256	maxpool5	2	0	3	3	256	256	0
						fc6			1	1	9216	4096	37752832
						fc7			1	1	4096	4096	16781312
						fc8			1	1	4096	1000	4097000
Total													**62,378,344**

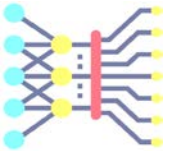
About 57M parameters are in the FC layers (about 95% of parameters)

Top-5 error on this competition (2012)



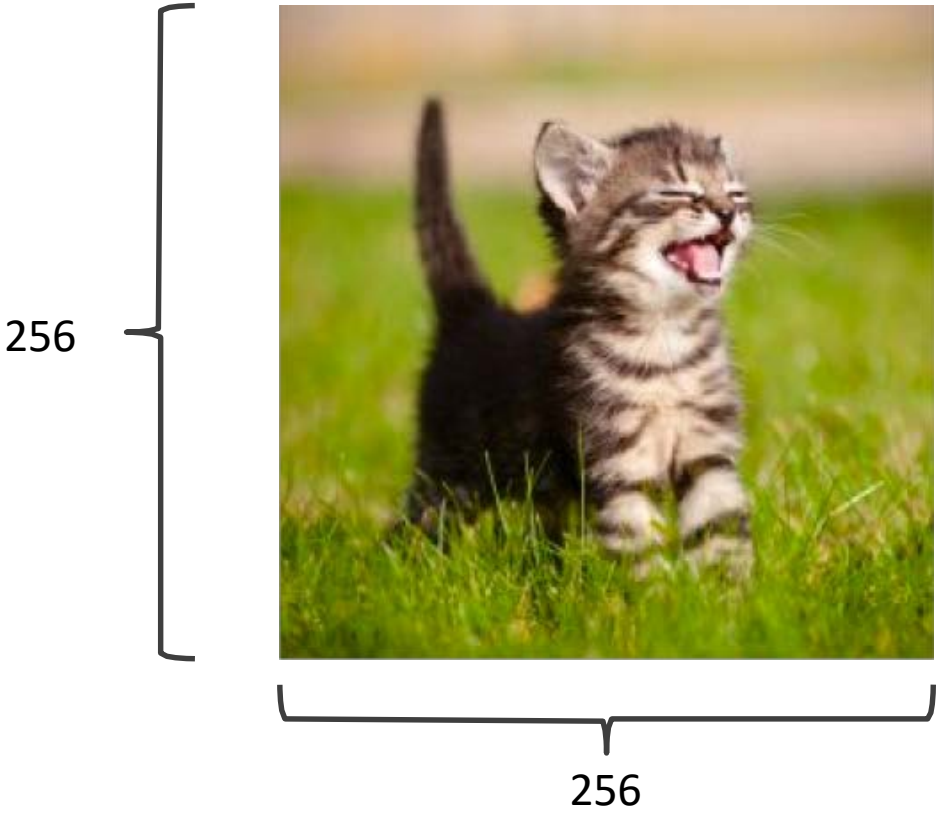
Imagenet Leaderboard





- Dropout
- Preprocessing and Data Augmentation
  - image translations and horizontal reflections,
  - performed Principle Component Analysis (PCA) on the RGB pixel values to change the intensities of RGB channels

Preprocessing and Data Augmentation



Preprocessing and Data Augmentation

224x224

Preprocessing and Data Augmentation

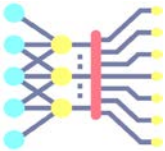
224x224





True label: Abyssinian cat

Alexnet

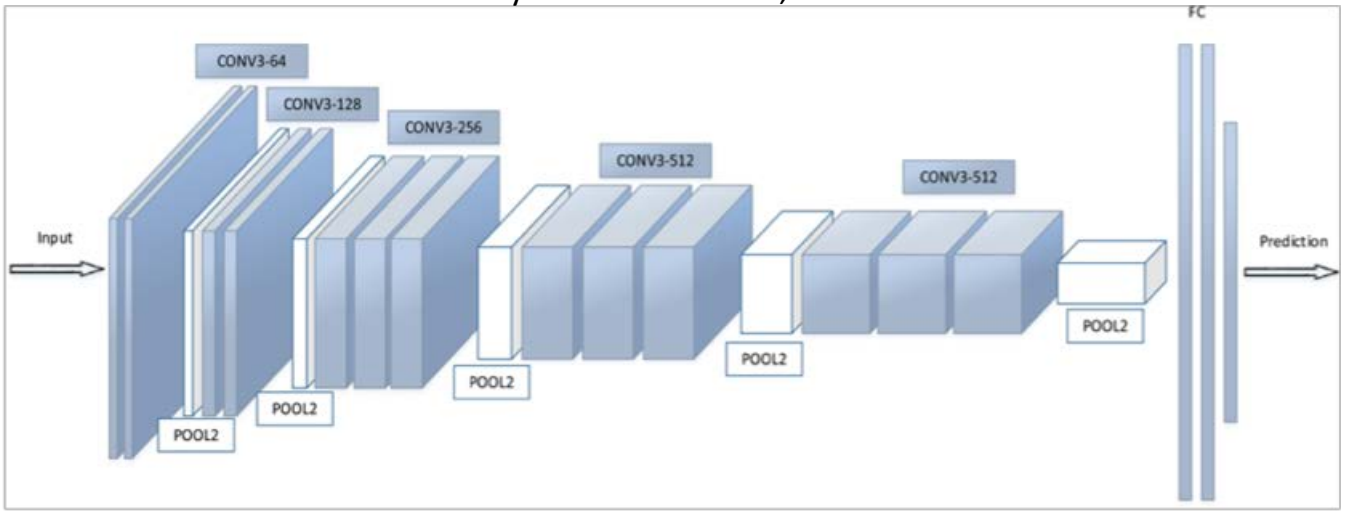


- Preprocessing and Data Augmentation

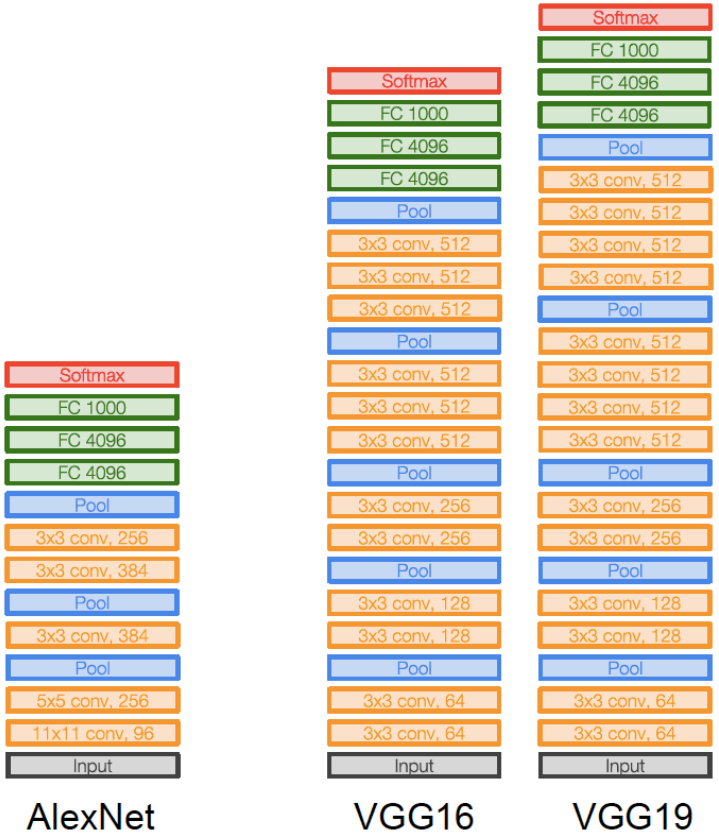
- Using ReLUs instead of Sigmoid or Tanh
- Momentum + Weight Decay
- Dropout (Randomly sets Unit outputs to zero during training)
- GPU Computation!

VGG Network: ILSVRC 2014 2nd place

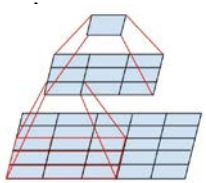
Simonyan and Zisserman, 2014.



VGGnet

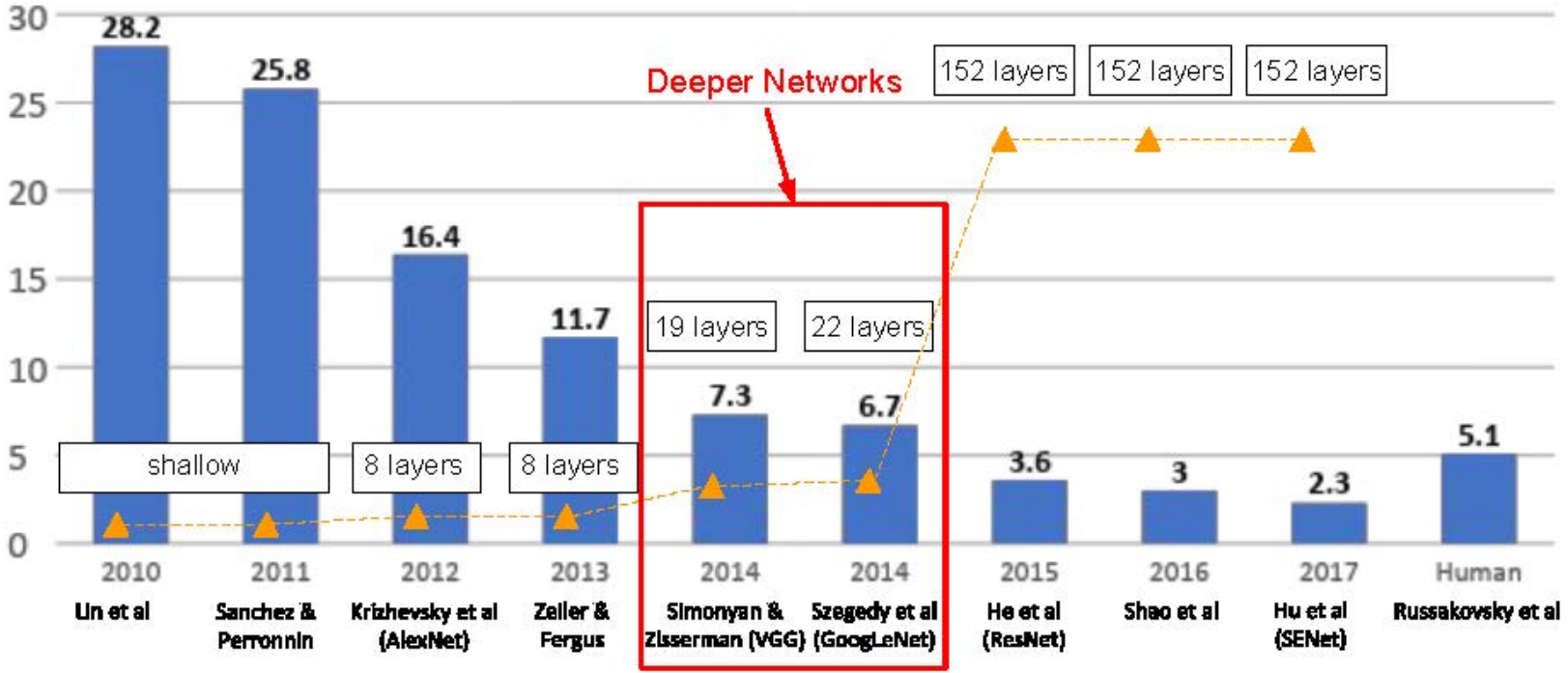


- Sequence of deeper networks trained progressively
- Large receptive fields replaced by successive layers of 3×3 convolutions (with ReLU in betwee

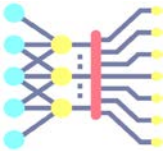


- How many weights does one 7×7 conv layer with K input and output channels have?
  - $49K^2$
- What about three 3×3 conv layers with K input and output channels?
  - $27K^2$
- All maxpool layers are 2×2 stride 2, followed by doubling of the number of channels (keeping cost of convolutions the same)

Imagenet Leaderboard



GoogLeNet: ILSVRC 2014 winner

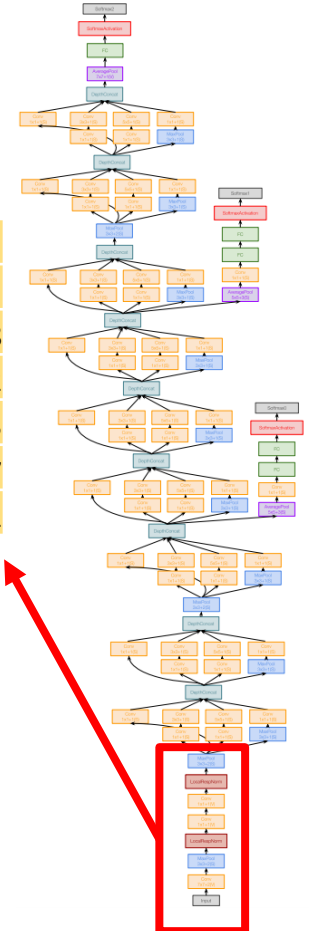


- Deeper networks with computational efficiency
  - 22 layers
- Efficient Inception modules
  - Networks with Parallel Concatenations
  - Employ a combination of variously-sized kernels.

GoogLeNet: Aggressive stem

Stem network at the start aggressively downsamples input

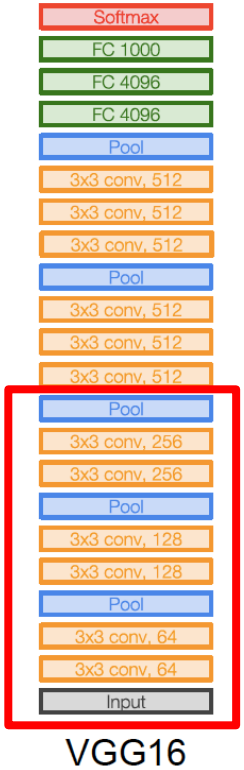
	Input size		Layer				Output size				
Layer	C	H / W	filters	kernel	stride	pad	C	H/W	memory (KB)	params (K)	flop (M)
conv	3	224	64	7	2	3	64	112	3136	9	118
max-pool	64	112		3	2	1	64	56	784	0	2
conv	64	56	64	1	1	0	64	56	784	4	13
conv	64	56	192	3	1	1	192	56	2352	111	347
max-pool	192	56		3	2	1	192	28	588	0	1



Total from 224 to 28 resolution:
Memory: 7.5 MB
Params: 124K
MFLOP: 418

Source: J. Johnson

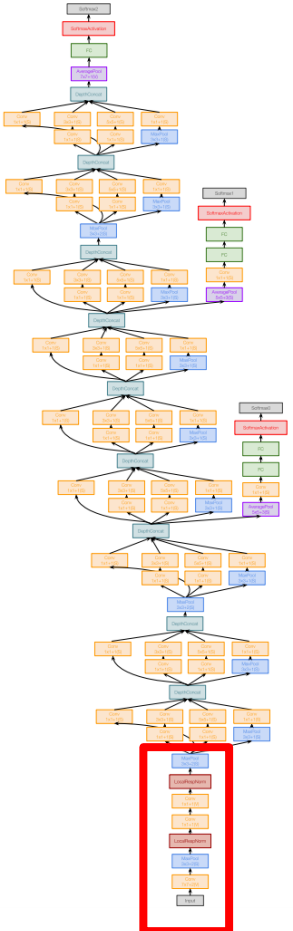
GoogLeNet: Aggressive stem



VGG16

Compare VGG-16:
Memory: 42.9 MB (5.7x)
Params: 1.1M (8.9x)
MFLOP: 7485 (17.8x)

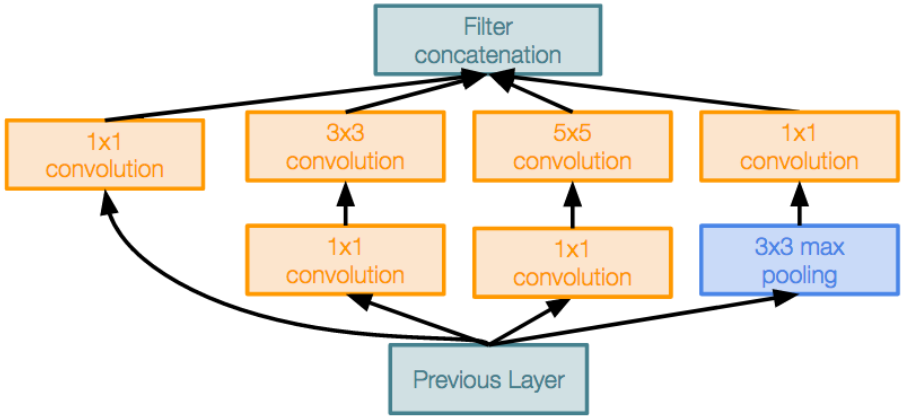
Total from 224 to 28 resolution:
Memory: 7.5 MB
Params: 124K
MFLOP: 418

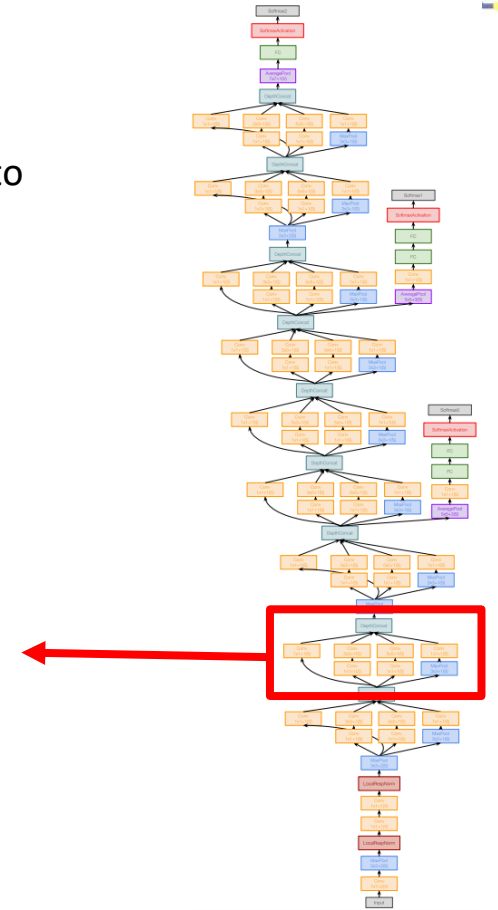


Source: J. Johnson

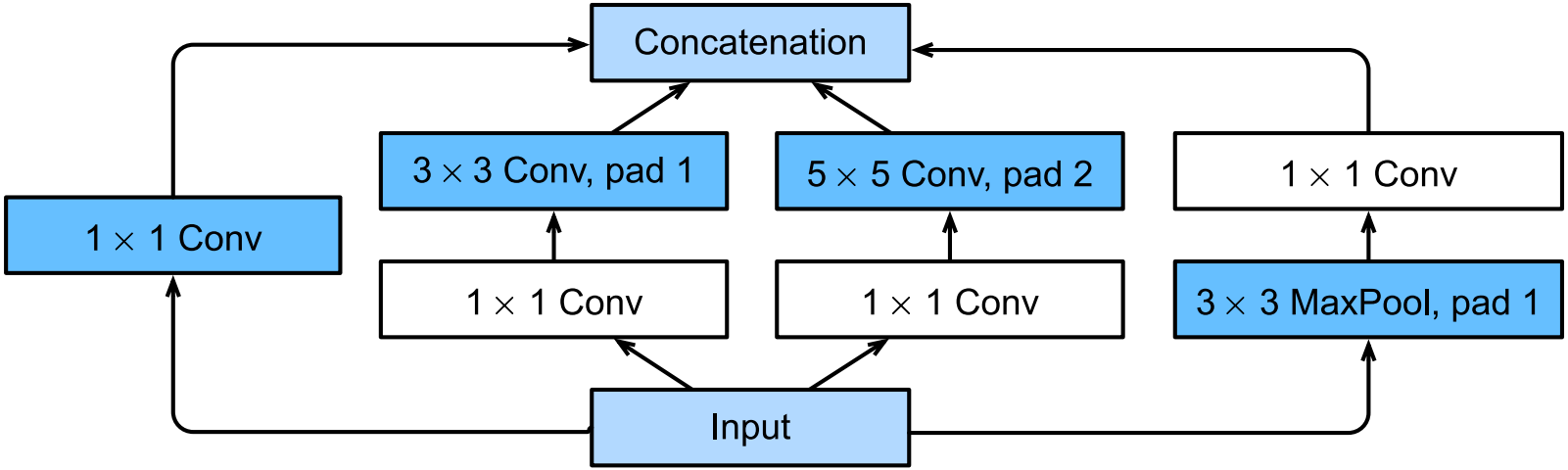
GoogLeNet: Inception module

- Design a good network topology (network within network) and stack these modules
- Parallel paths with different receptive field sizes and operations are meant to capture sparse patterns of correlations in the stack of feature maps
- Use 1x1 convolutions for dimensionality reduction before expensive convolutions



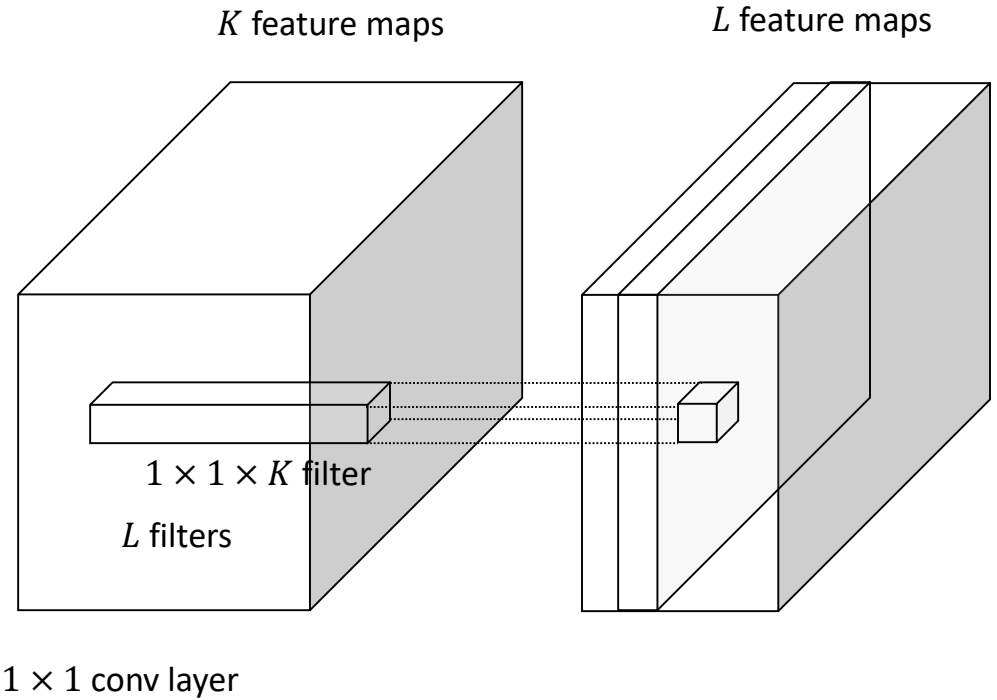


Source: J. Johnson

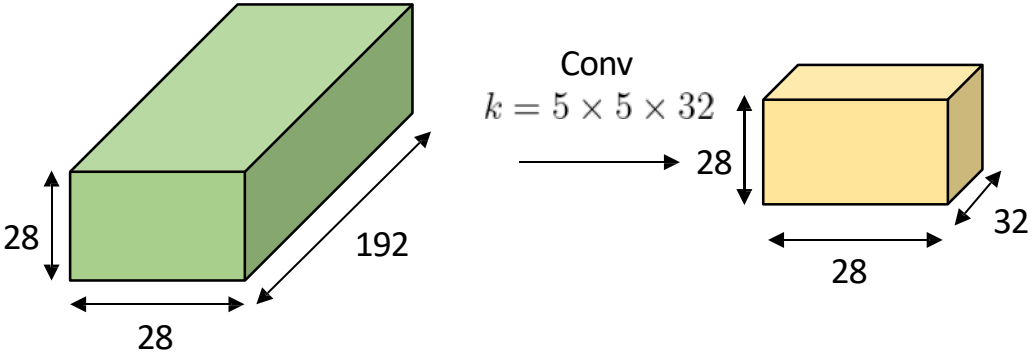


The first three paths use convolutional layers with window sizes of 1×1, 3×3, and 5×5 to extract information from different spatial sizes.
The middle two paths perform a 1×1 convolution on the input to reduce the number of channels, reducing the model's complexity.
The fourth path uses a 3×3 maximum pooling layer, followed by a 1×1 convolutional layer to change the number of channels.
The four paths all use appropriate padding to give the input and output the same height and width.
Finally, the outputs along each path are concatenated along the channel dimension and comprise the block's output.

1x1 convolution



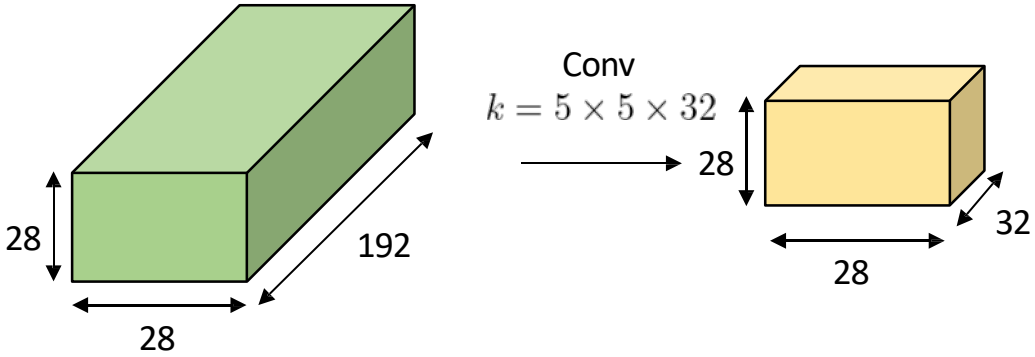
GoogLeNet: Motivation of using a 1x1 Convolutional Layer



Total number of operations:

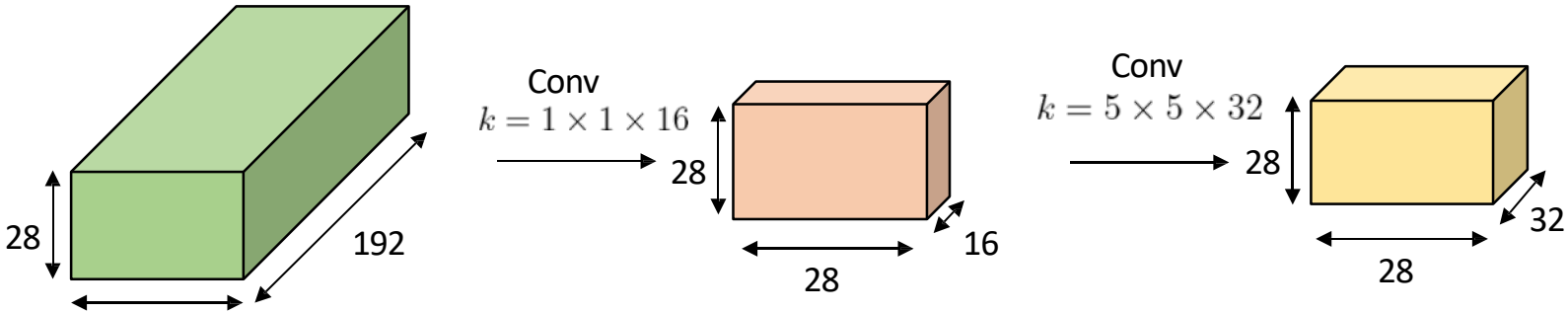
(28 x 28 x 32) x (5 x 5 x 192) = 120M

GoogLeNet: Motivation of using a 1x1 Convolutional Layer



Total number of operations:

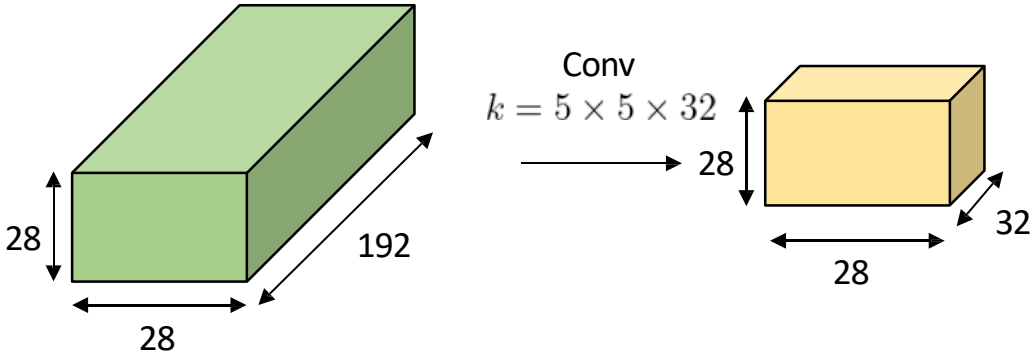
(28 x 28 x 32) x (5 x 5 x 192) = 120M



28**Total number of operations:**

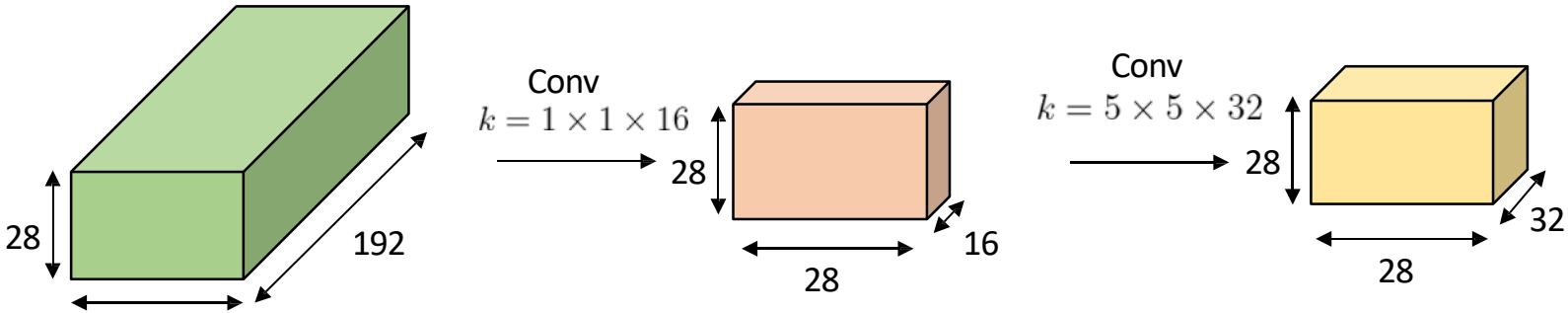
(28 x 28 x 16) x (1 x 1 x 192) + (28 x 28 x 32) x (5 x 5 x 16) = 12.4M

GoogLeNet: Motivation of using a 1x1 Convolutional Layer



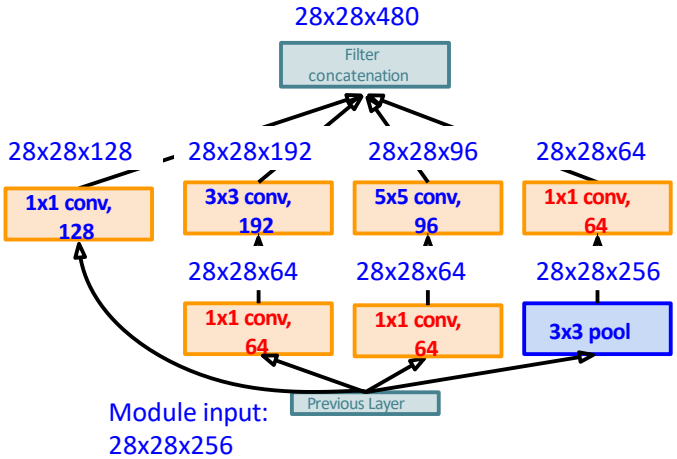
Total number of operations:

(28 x 28 x 32) x (5 x 5 x 192) = 120M



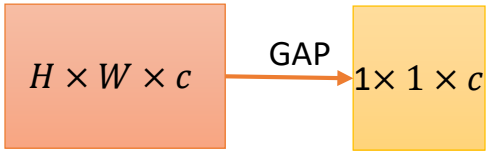
28**Total number of operations:**

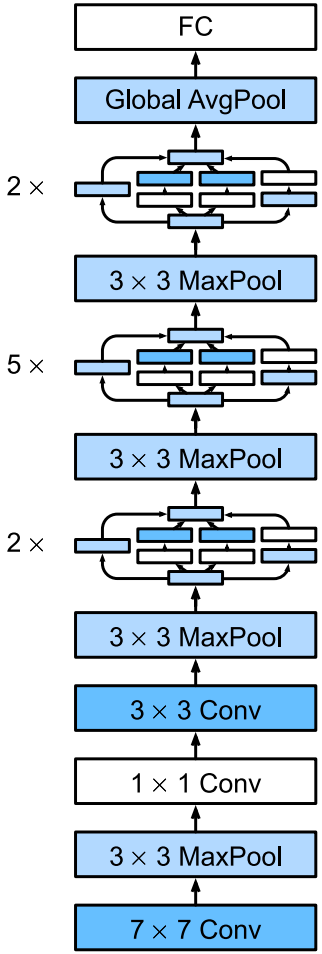
(28 x 28 x 16) x (1 x 1 x 192) + (28 x 28 x 32) x (5 x 5 x 16) = 12.4M


28x28x480
Filter concatenation
28x28x128
28x28x192
28x28x96
28x28x64
1x1 conv, 128
3x3 conv, 192
5x5 conv, 96
1x1 conv, 64
28x28x64
28x28x64
28x28x256
1x1 conv, 64
1x1 conv, 64
3x3 pool
Previous Layer
Module input: 28x28x256

GoogLeNet Model

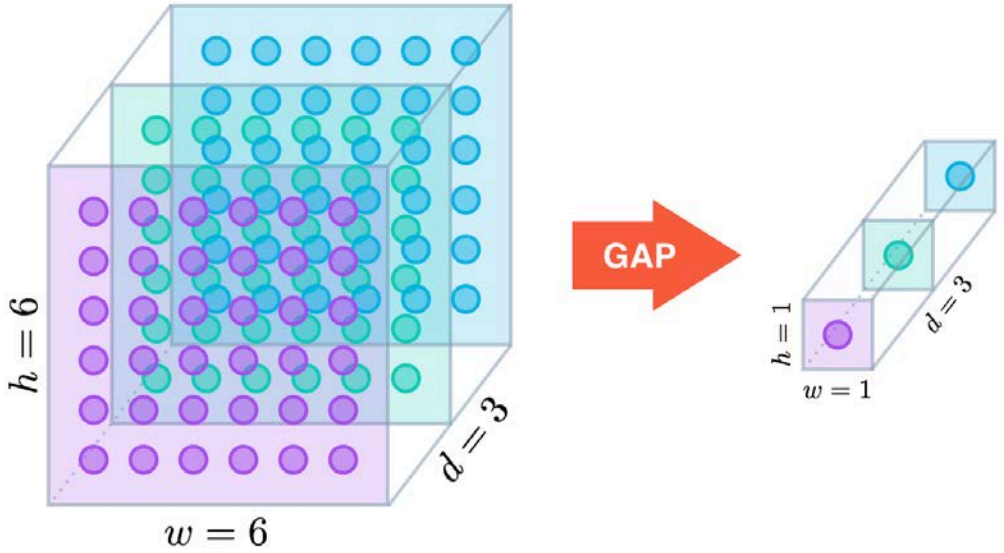
- Uses a stack of 9 inception blocks
- 22 total layers with weights
- Maximum pooling between inception blocks reduces the dimensionality.
- After the last conv layer, a global average pooling layer is used that spatially averages across each feature map before final FC layer.



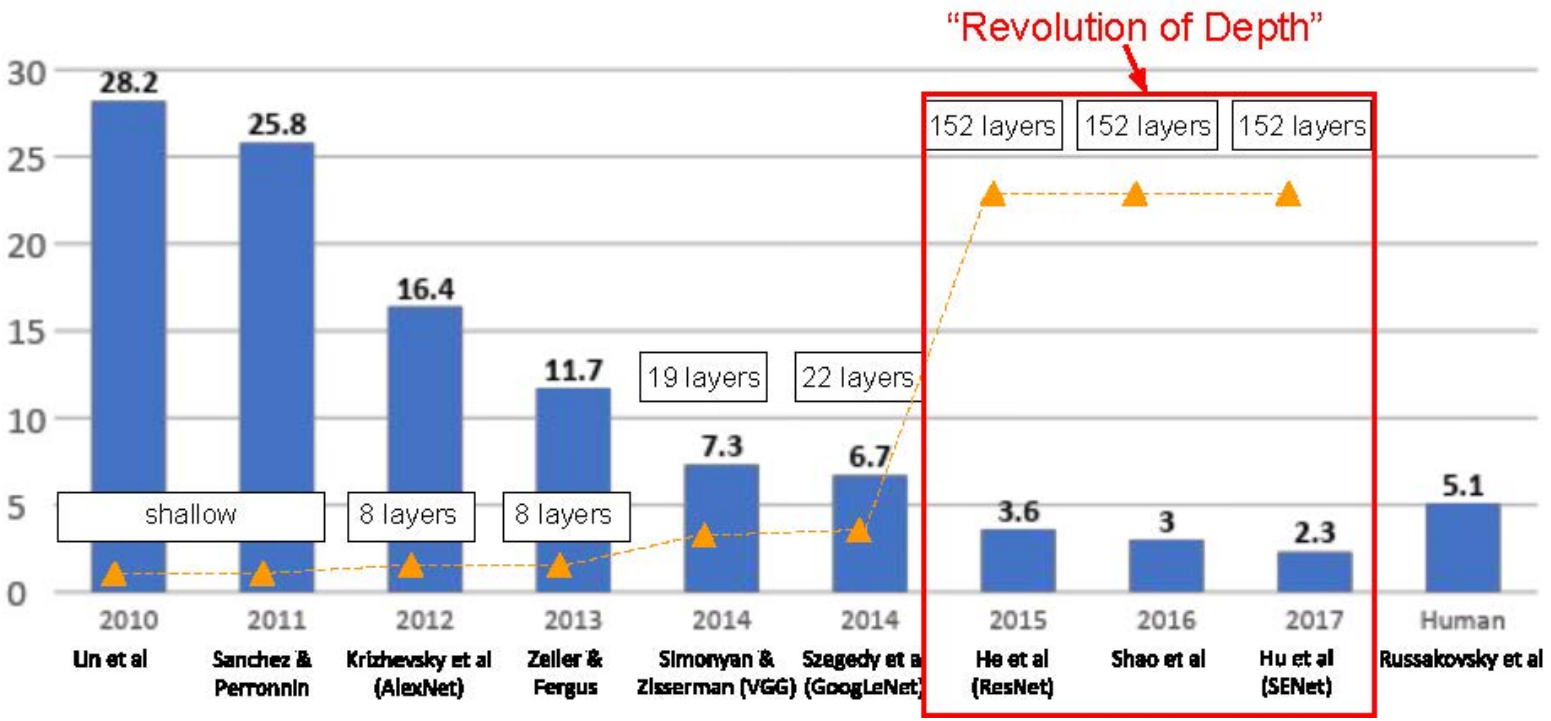


Global Average Pooling (GAP)

- Reduce the spatial dimensions of a tensor.
- A tensor with dimensions h×w×d is reduced in size to have dimensions 1×1×d.
- GAP layers reduce each h×w feature map to a single number by simply taking the average of all hw values.

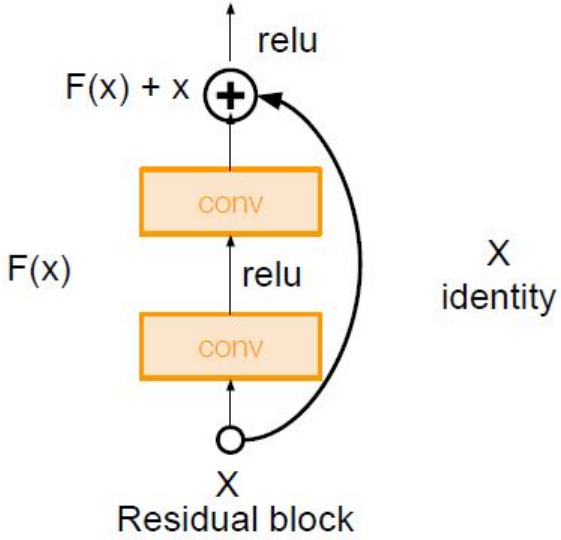


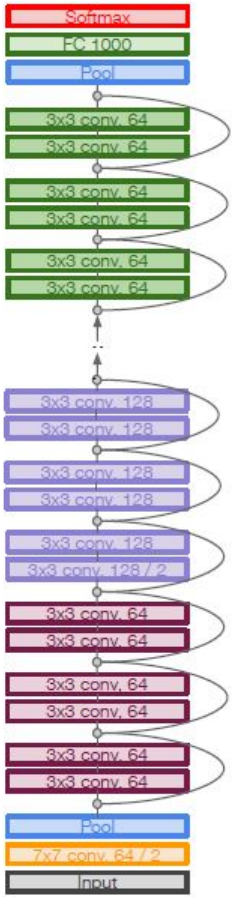
ResNet


"Revolution of Depth"
30
25
20
15
10
5
0
28.2
25.8
16.4
11.7
7.3
6.7
3.6
3
2.3
5.1
shallow
8 layers
8 layers
19 layers
22 layers
152 layers
152 layers
152 layers
2010
2011
2012
2013
2014
2014
2015
2016
2017
Human
Lin et al
Sanchez & Perronnin
Krizhevsky et al (AlexNet)
Zeiler & Fergus
Simonyan & Zisserman (VGG)
Szegedy et al (GoogLeNet)
He et al (ResNet)
Shao et al
Hu et al (SENet)
Russakovsky et al

Resnet

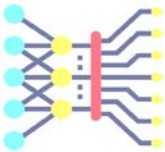
- Naïve solution
  - If extra layers are an identity mapping, then training errors do not increase





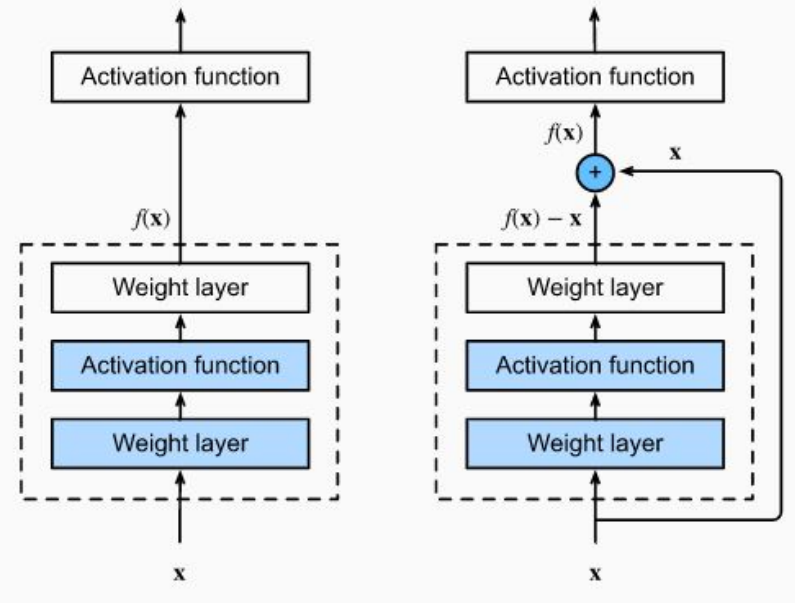
Very deep networks using residual connections

- 152 layer for ImageNet
- ILSVRC 2015 classification winner



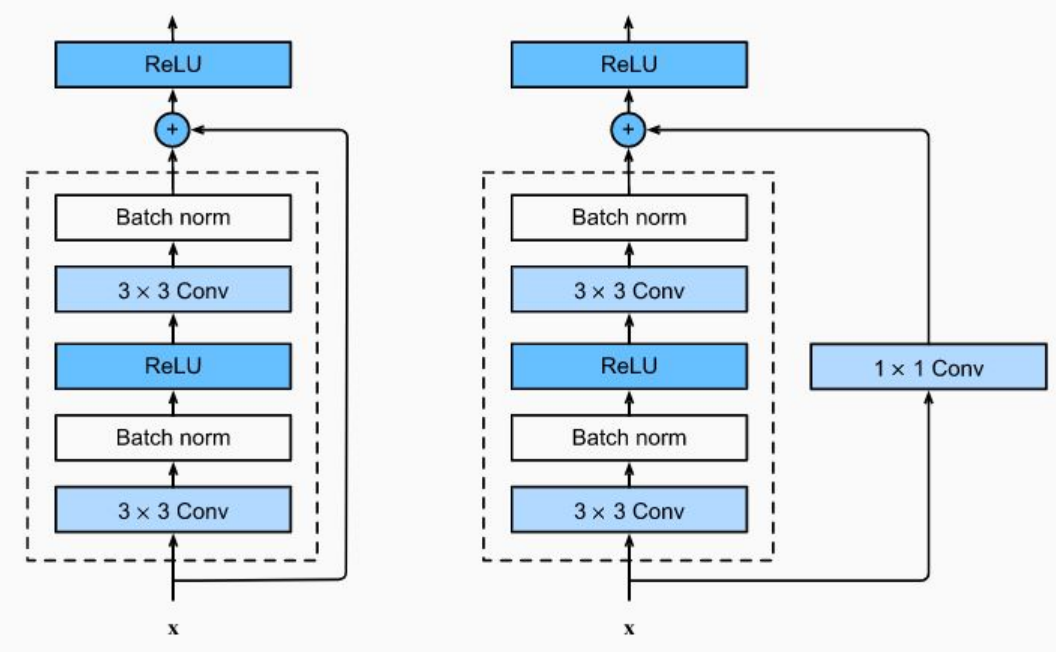
- Deep models have more representation power (more parameters) than shallower models.
- But deeper models are harder to optimize
- What should the deeper model learn to be at least as good as the shallower model?
- A solution by construction is copying the learned layers from the shallower model and setting additional layers to identity mapping.

Residual Blocks

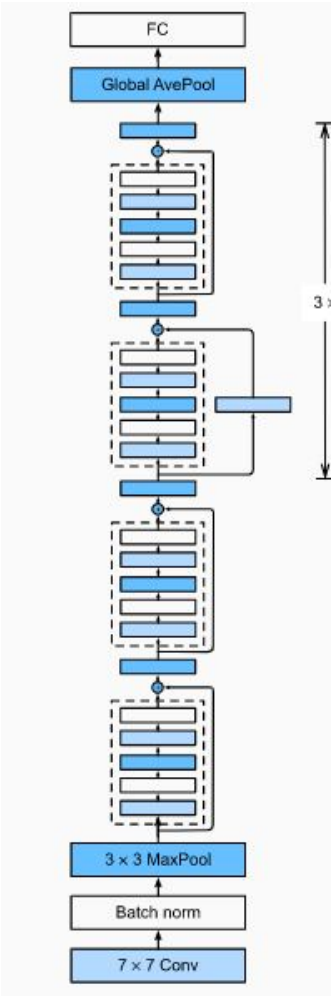


- The portion within the dotted-line box needs to learn the residual mapping.
- The solid line carrying the layer input x to the addition operator is called a *residual connection* (or *shortcut connection*).
- Inputs can forward propagate faster through the residual connections across layers.

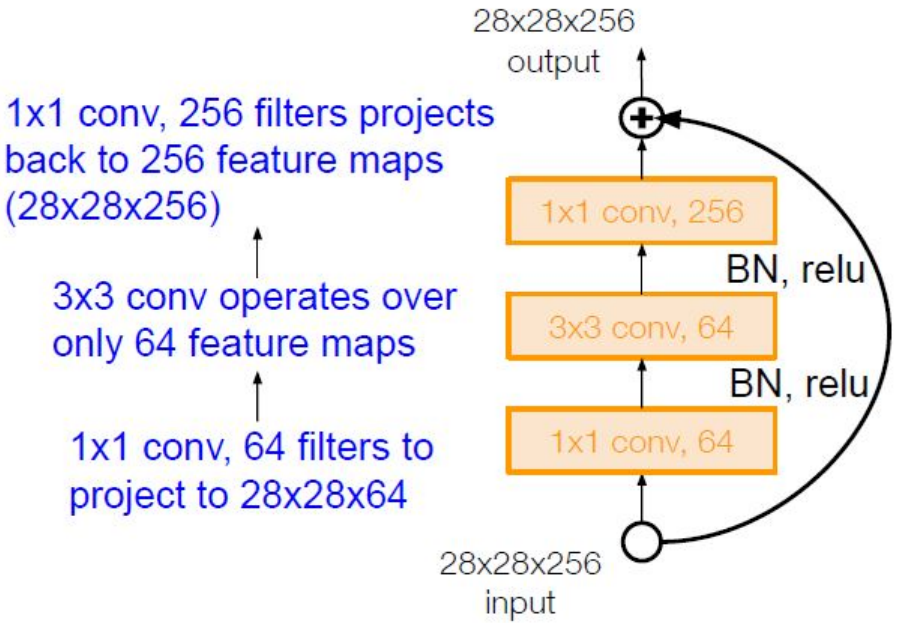
Resnet Blocks



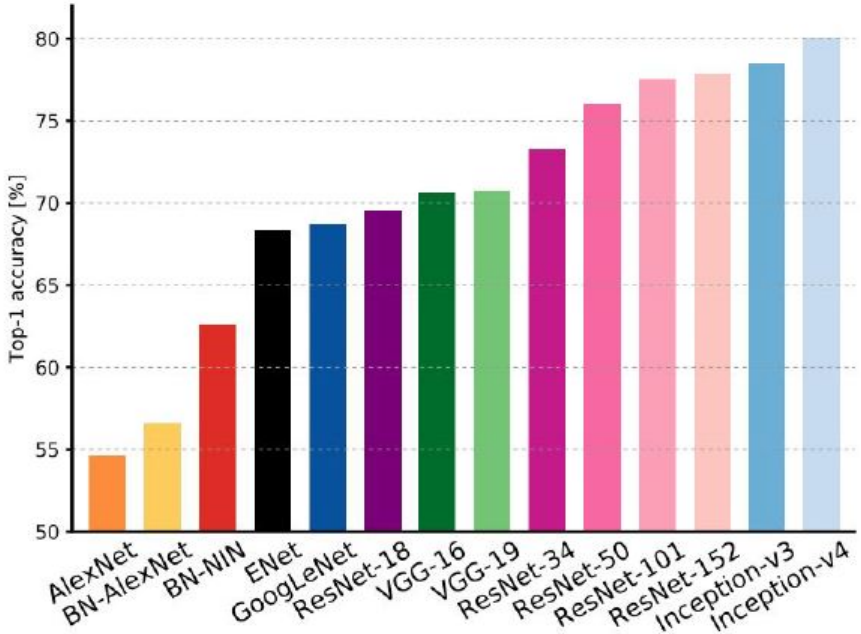
Resnet Model

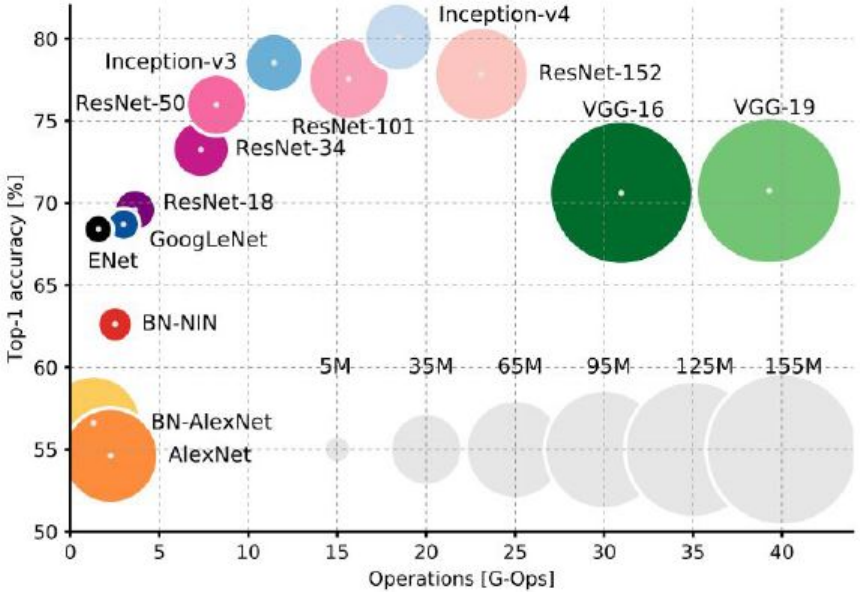


For deeper networks (ResNet-50+), use “bottleneck” layer to improve efficiency (similar to GoogLeNet)

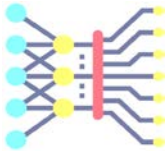


Comparing complexity...





An Analysis of Deep Neural Network Models for Practical Applications, 2017.



Other Ideas: Densely Connected Convolutional Networks (DenseNet)

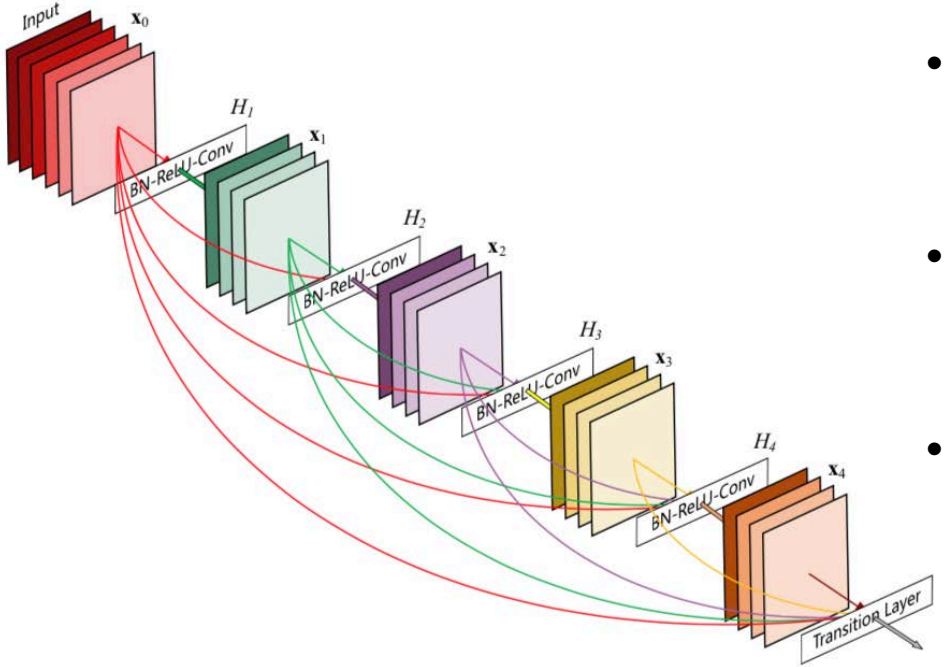


Figure 1: A 5-layer dense block with a growth rate of $k = 4$. Each layer takes all preceding feature-maps as input.

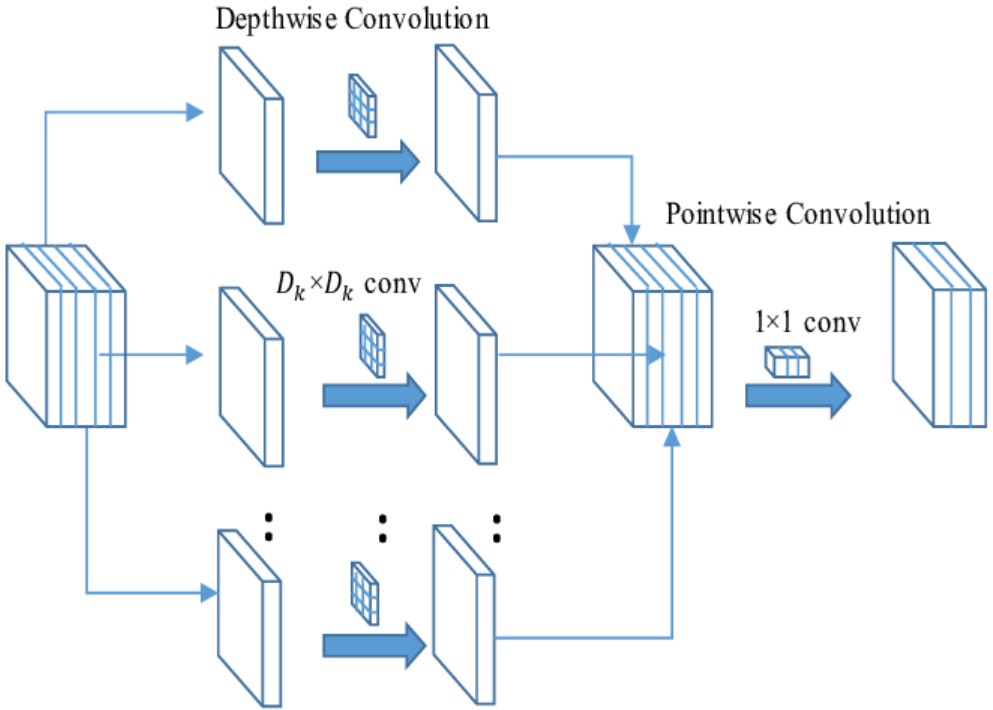
- Dense blocks where each layer is connected to every other layer in feedforward fashion
- Alleviates vanishing gradient, strengthens feature propagation, encourages feature reuse
- Showed that shallow 50-layer network can outperform deeper 152 layer ResNet

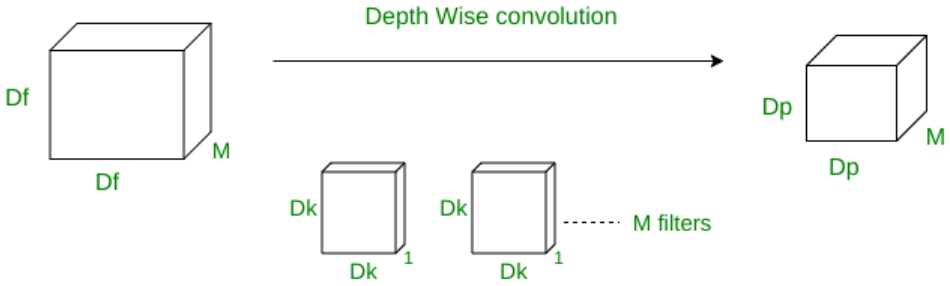
MobileNets: Efficient Convolutional Neural Networks for Mobile Applications

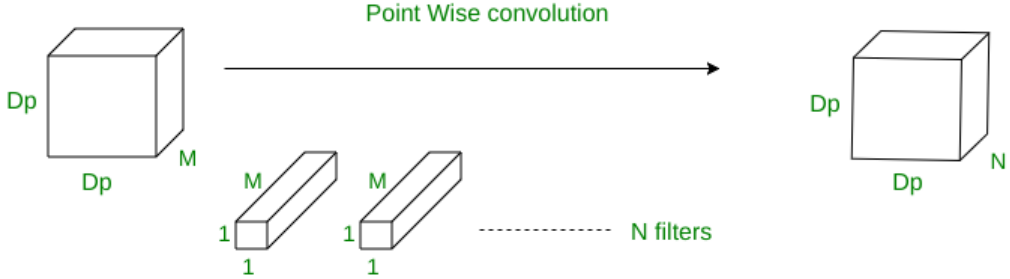
Depthwise separable convolutions replace standard convolutions by factorizing them into

- a depthwise convolution and
- a 1x1 convolution (pointwise convolution)

- Much more efficient.
- Little loss in accuracy




Depth Wise convolution
Df
M
Df
Dk
1
Dk
Dk
1
Dk
M filters
Dp
M
Dp


Point Wise convolution
Dp
M
Dp
M
1
1
M
1
1
N filters
Dp
N
Dp

Depthwise separable convolution