Applied Machine Learning with Big Data "EE 6973"

Topic:
Convolution Neural Networks

Paul Rad, Ph.D.

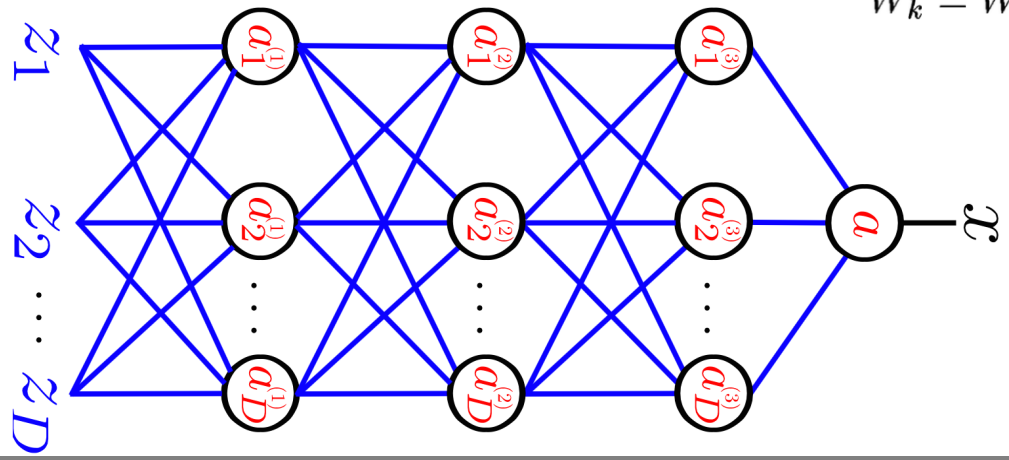
Chief Research Officer
UTSA Open Cloud Institute(OCI)
University of Texas at San Antonio

Stochastic Gradient Descent (SGD)

Loop:

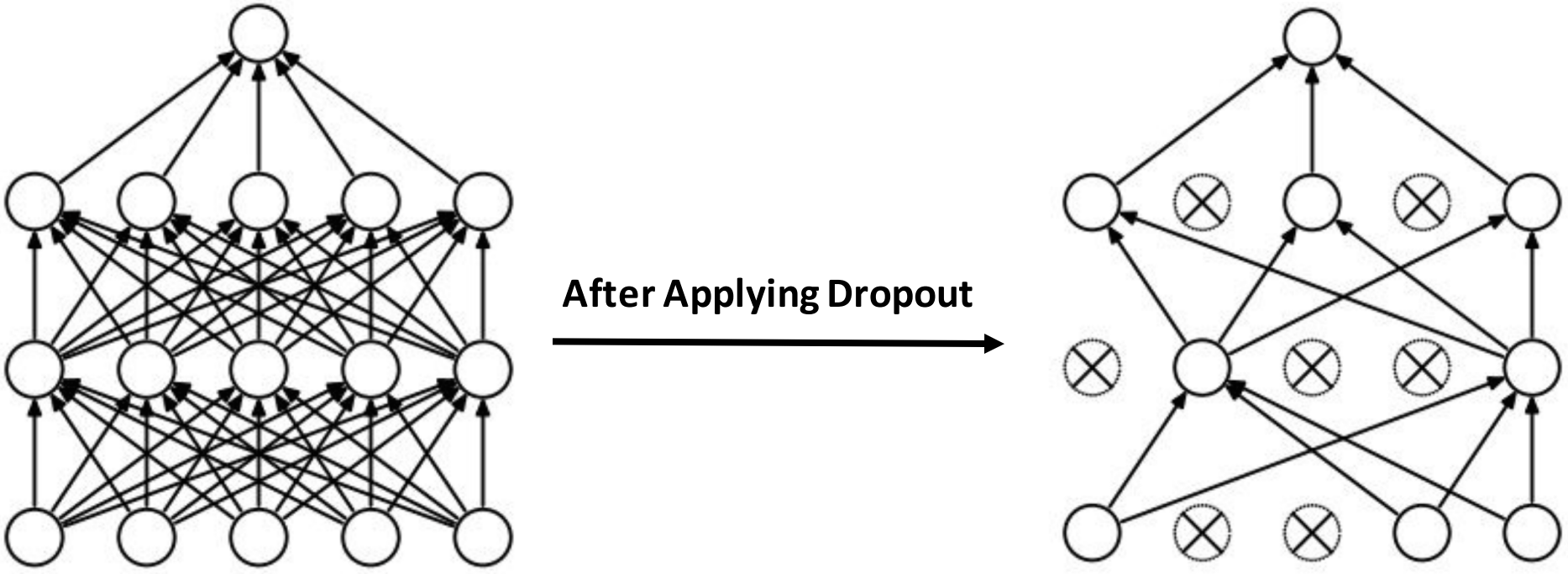
1. **Sample a batch of data with their labels**
2. **Forward Propagate it through the graph, calculate the error**
3. **Backpropagate to calculate the gradients**
4. **Update the parameters (weights) using the gradient**

$$W_k = W_{k-1} - \epsilon \frac{\partial E(W)}{\partial W}$$



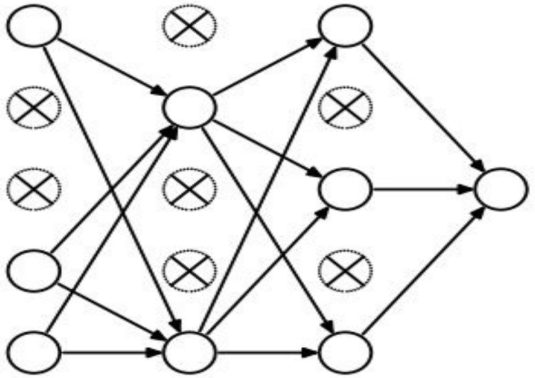
Dropout

Randomly set some neurons to zero in the forward pass

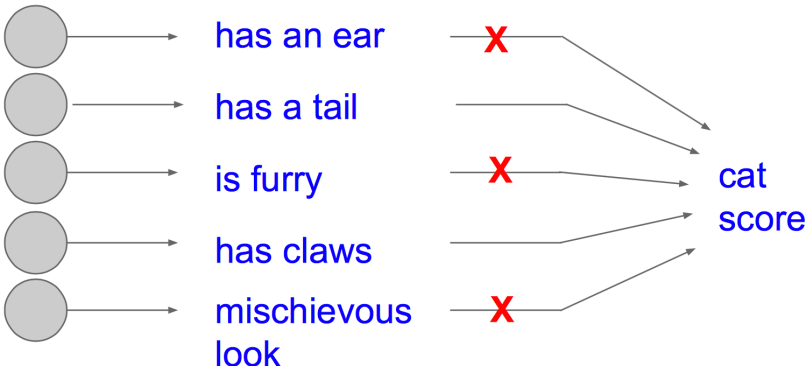


Srivastava et al., 2014, https://www.cs.toronto.edu/~hinton/absps/JMLRdropout.pdf

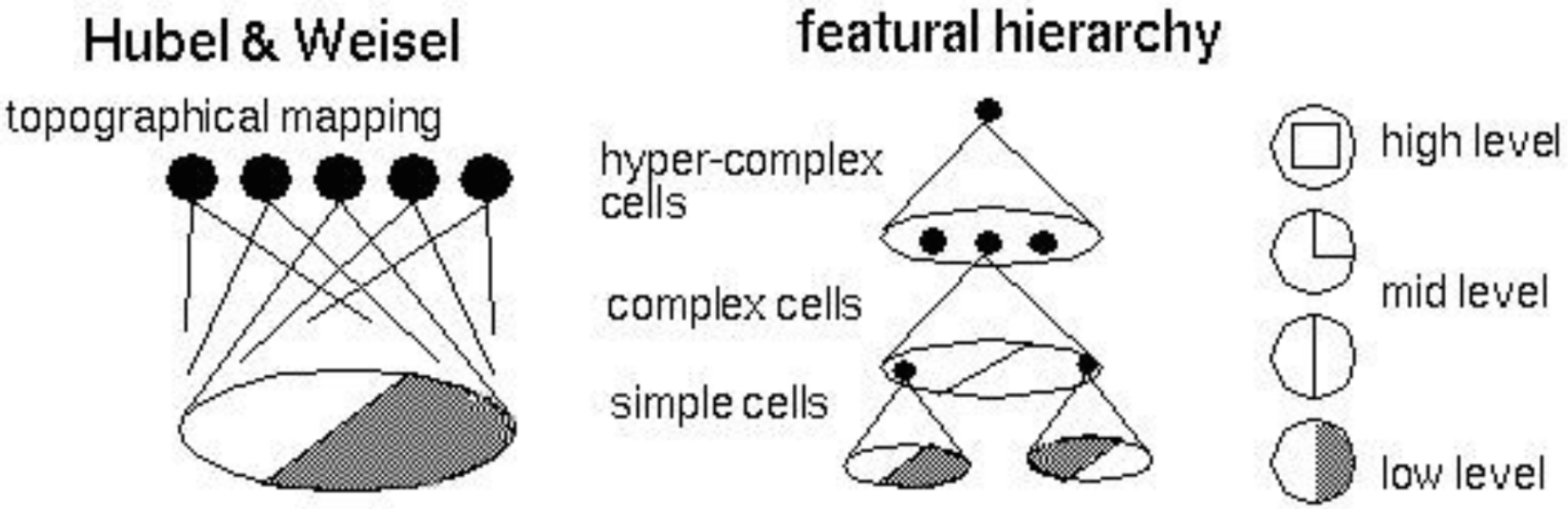
Dropout



Forces the network to have a redundant representation

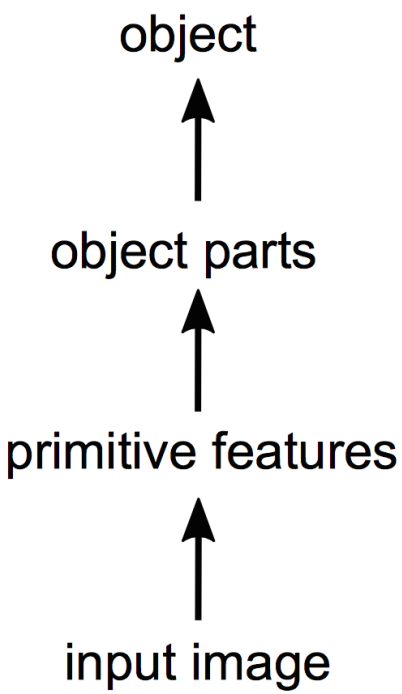


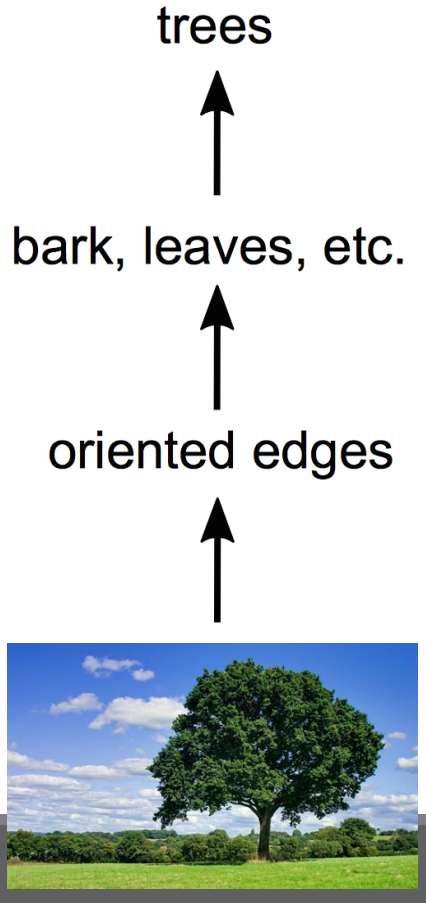
Vision: Hierarchical Organization (Year: 1962)

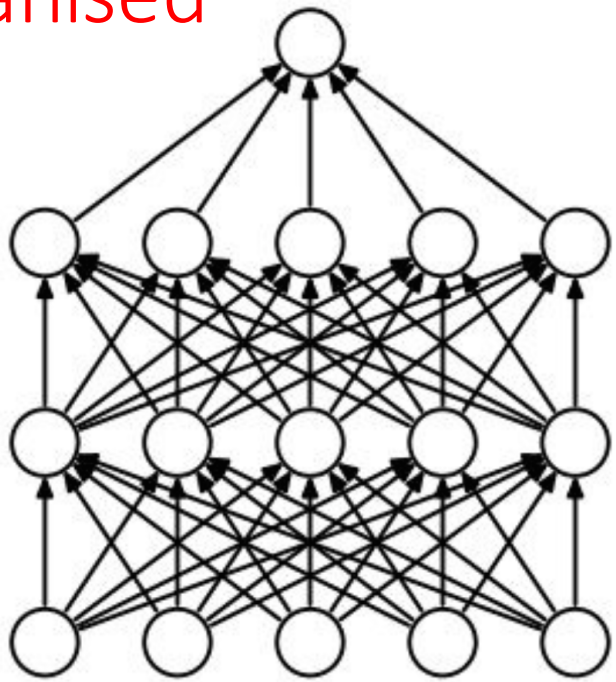


Why use hierarchical multi-layered models?

Biological vision is hierachically organised







What's wrong with standard neural networks?

Hard to Train

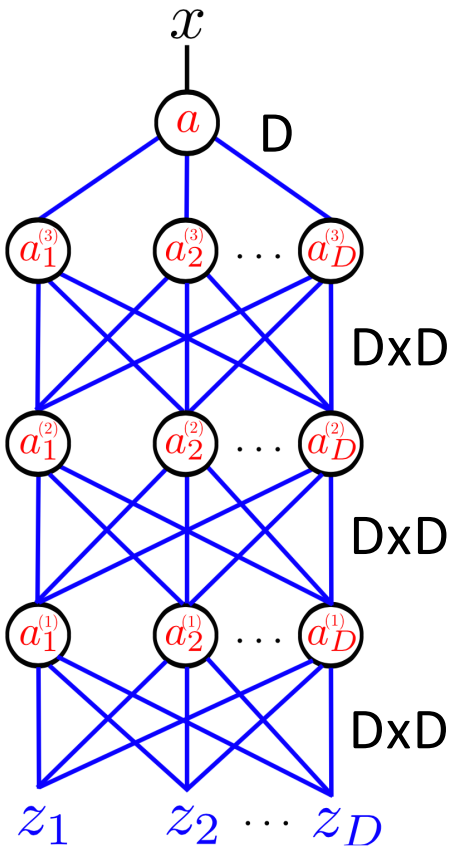
How many parameters does this network have?

Number of Parameters = 3 x (DxD) + D

For a small D = 32 x 32 = 1024 MNIST image:

Number of Parameters = 3 x (1024x1024) + 1024

~ 3×10^6



Architecture of LeNet-5, Convolution Neural Network

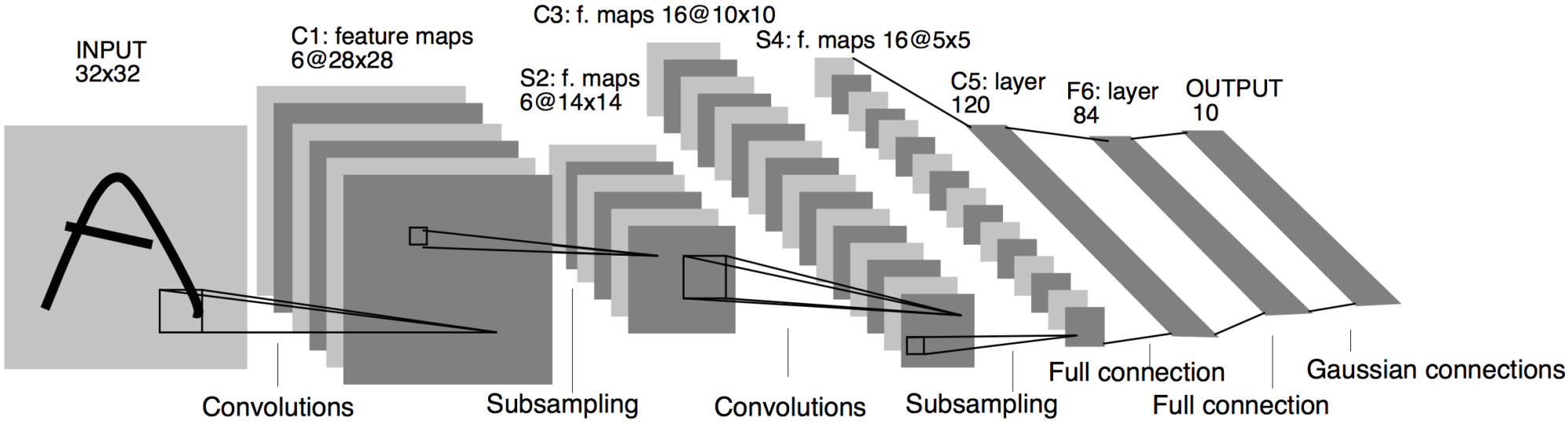
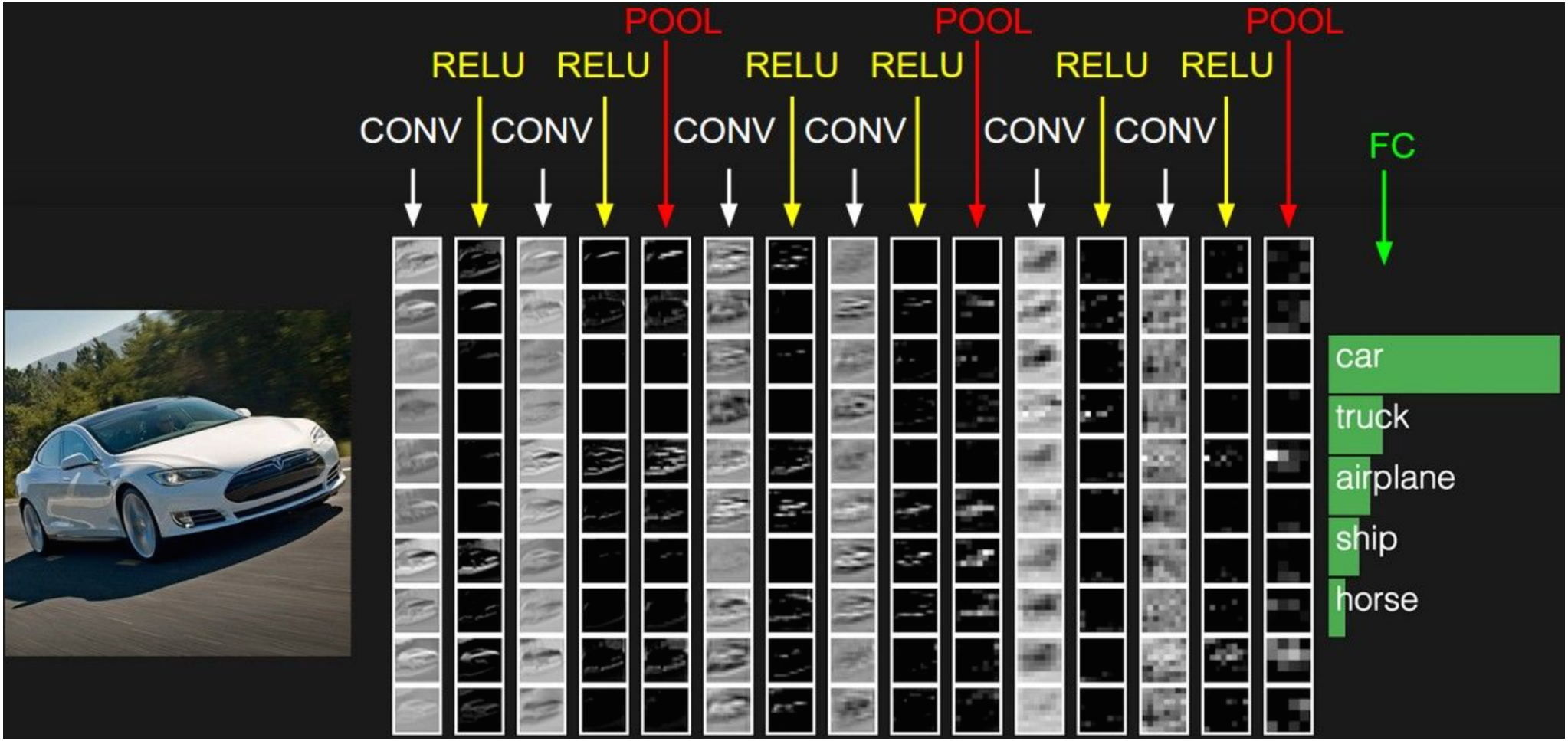


Fig. 2. Architecture of LeNet-5, a Convolutional Neural Network, here for digits recognition. Each plane is a feature map, i.e. a set of units whose weights are constrained to be identical.

Proc. Of the IEEE, November 1998, "Gradient-Based Learning Applied to Document Recognition"


POOL
POOL
POOL
RELU
RELU
RELU
RELU
RELU
RELU
CONV
CONV
CONV
CONV
CONV
CONV
FC
car
truck
airplane
ship
horse

Review: What is convolution?

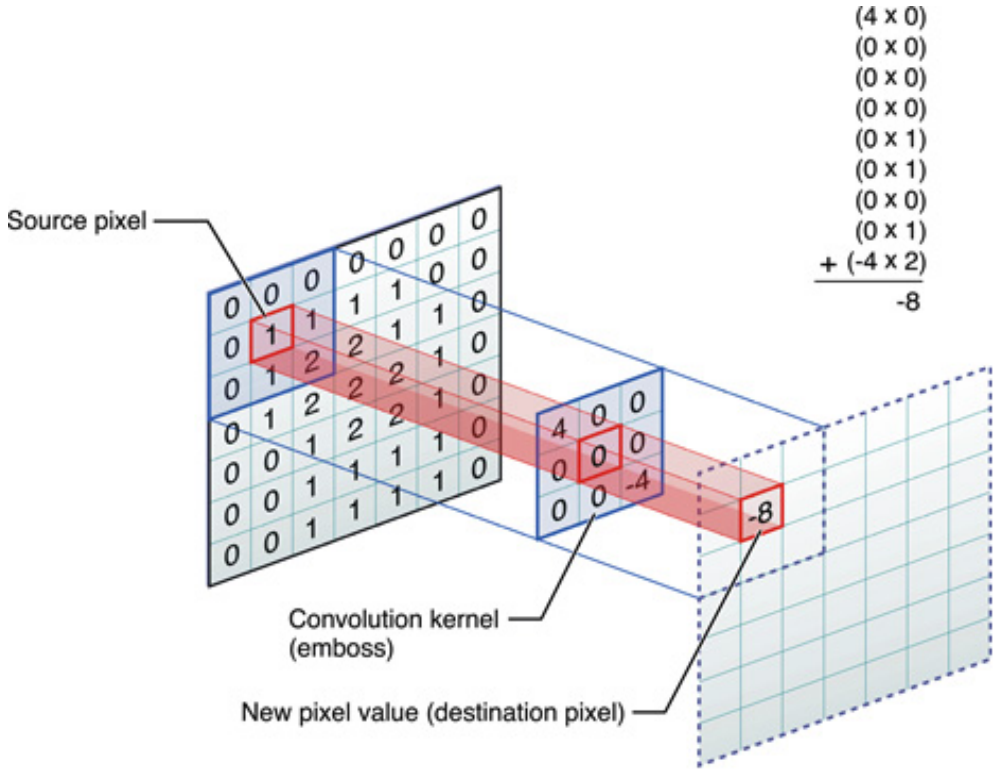
- Convolution is an important operation from signal processing
- A convolution is an integral that expresses the amount of overlap of one function as it is shifted over another function .

$$f * g = \int_{-\infty}^{\infty} f(\tau)\, g(t-\tau)\, d\tau = \int_{-\infty}^{\infty} g(\tau)\, f(t-\tau)\, d\tau$$

- 2 Dimensional Discrete Function (Image)

$$f[x,y] * g[x,y] = \sum_{n_1=-\infty}^{\infty} \sum_{n_2=-\infty}^{\infty} f[n_1,n_2] \cdot g[x-n_1, y-n_2]$$

2-Dimensional Convolution



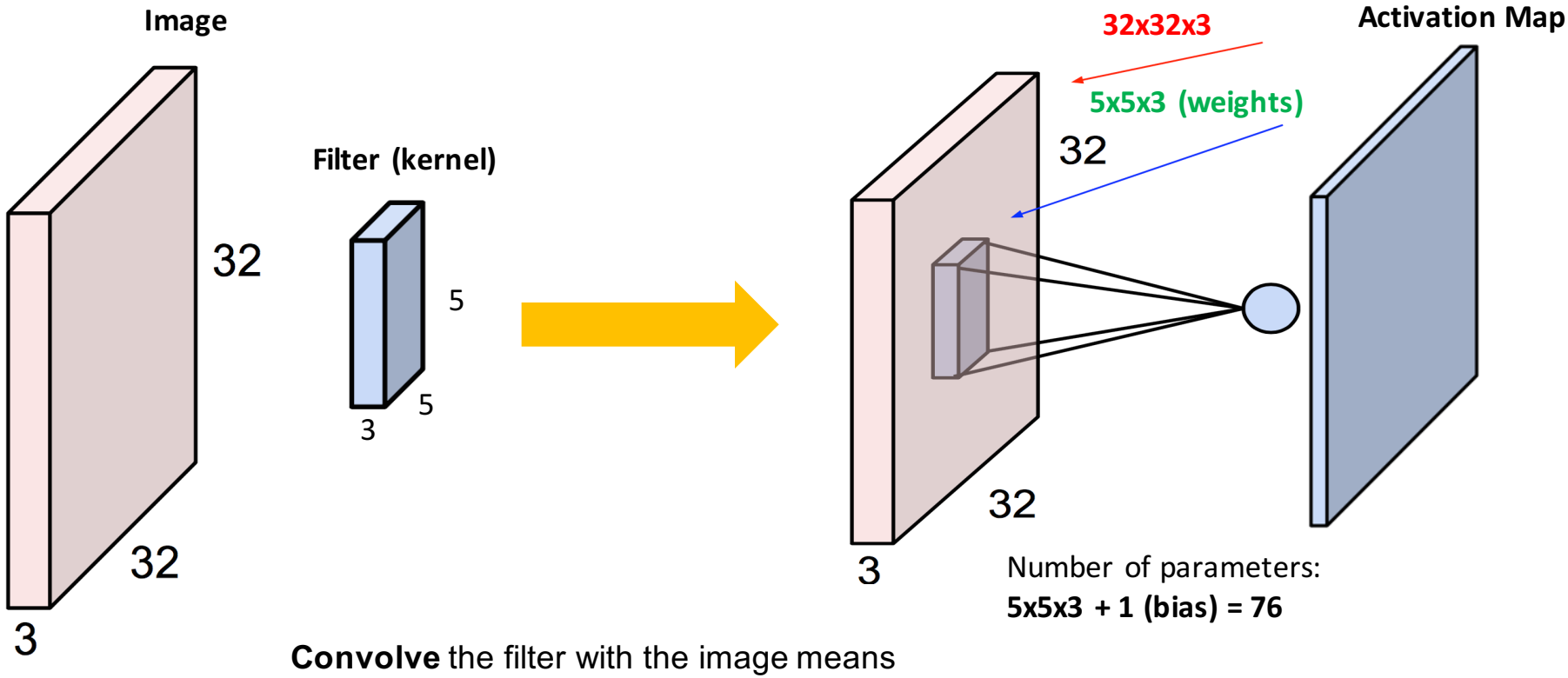
Example: 2-Dimensional Convolution

A convolution is an integral (discrete signals :Matrix Dot Product) that expresses the amount of overlap of one function as it is shifted over another function

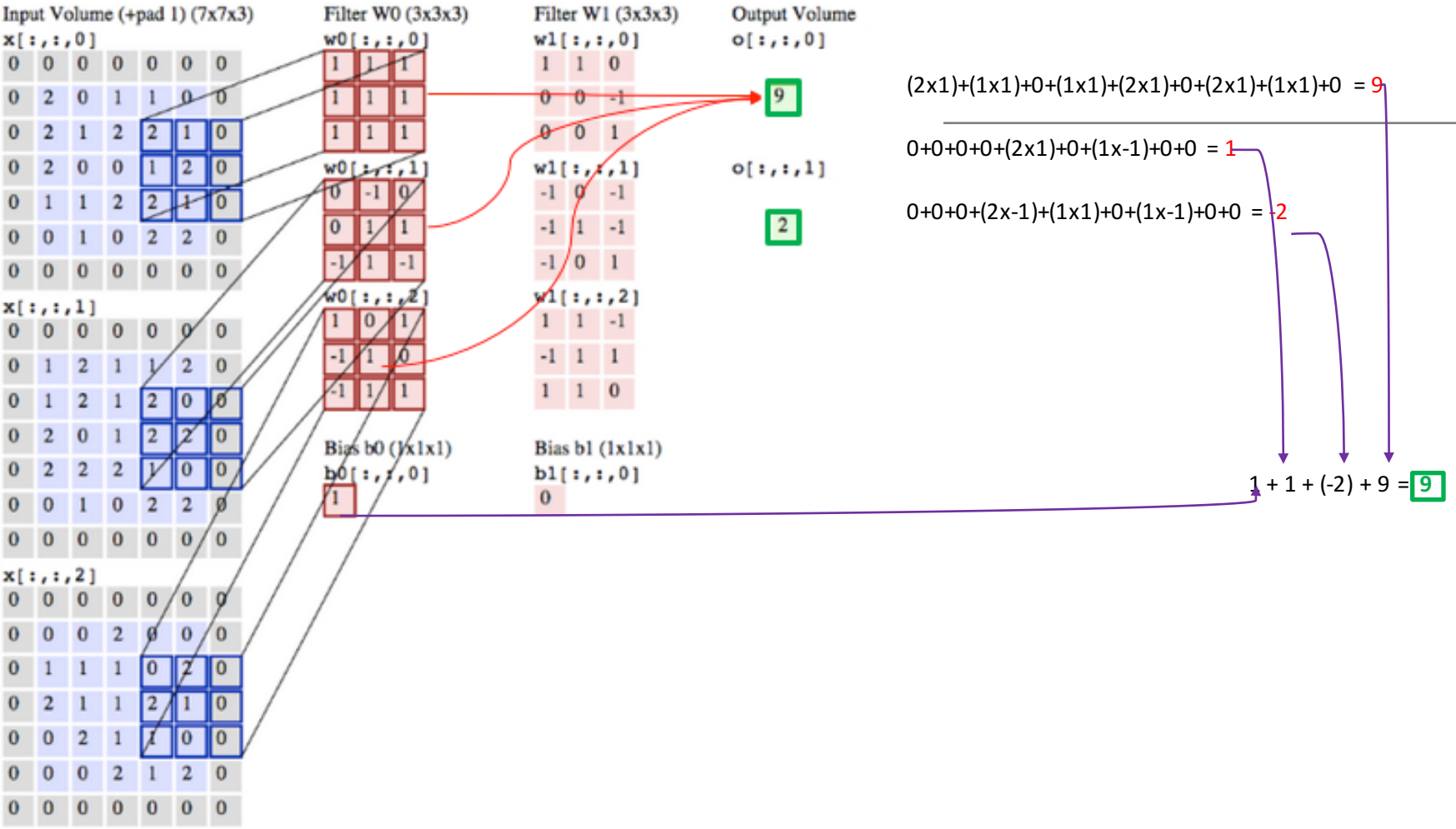
-1	-2	-1
0	0	0
1	2	1

-1	0	1
-2	0	2
-1	0	1

Convolution Layer



Convolve the filter with the image means
"slide over the image spatially, compute dot products"


Input Volume (+pad 1) (7x7x3)
x[:,:,0]
0 0 0 0 0 0 0
0 2 0 1 1 0 0
0 2 1 2 2 1 0
0 2 0 0 1 2 0
0 1 1 2 2 1 0
0 0 1 0 2 2 0
0 0 0 0 0 0 0
x[:,:,1]
0 0 0 0 0 0 0
0 1 2 1 1 2 0
0 1 2 1 2 0 0
0 2 0 1 2 2 0
0 2 2 2 1 0 0
0 0 1 0 2 2 0
0 0 0 0 0 0 0
x[:,:,2]
0 0 0 0 0 0 0
0 0 0 2 0 0 0
0 1 1 1 0 2 0
0 2 1 1 2 1 0
0 0 2 1 1 0 0
0 0 0 2 1 2 0
0 0 0 0 0 0 0
Filter W0 (3x3x3)
w0[:,:,0]
1 1 1
1 1 1
1 1 1
w0[:,:,1]
0 -1 0
0 1 1
-1 1 -1
w0[:,:,2]
1 0 1
-1 1 0
-1 1 1
Bias b0 (1x1x1)
b0[:,:,0]
1
Filter W1 (3x3x3)
w1[:,:,0]
1 1 0
0 0 -1
0 0 1
w1[:,:,1]
-1 0 -1
-1 1 -1
-1 0 1
w1[:,:,2]
1 1 -1
-1 1 1
1 1 0
Bias b1 (1x1x1)
b1[:,:,0]
0
Output Volume
o[:,:,0]
9
o[:,:,1]
2
(2x1)+(1x1)+0+(1x1)+(2x1)+0+(2x1)+(1x1)+0 = 9
0+0+0+0+(2x1)+0+(1x-1)+0+0 = 1
0+0+0+(2x-1)+(1x1)+0+(1x-1)+0+0 = -2
1 + 1 + (-2) + 9 = 9

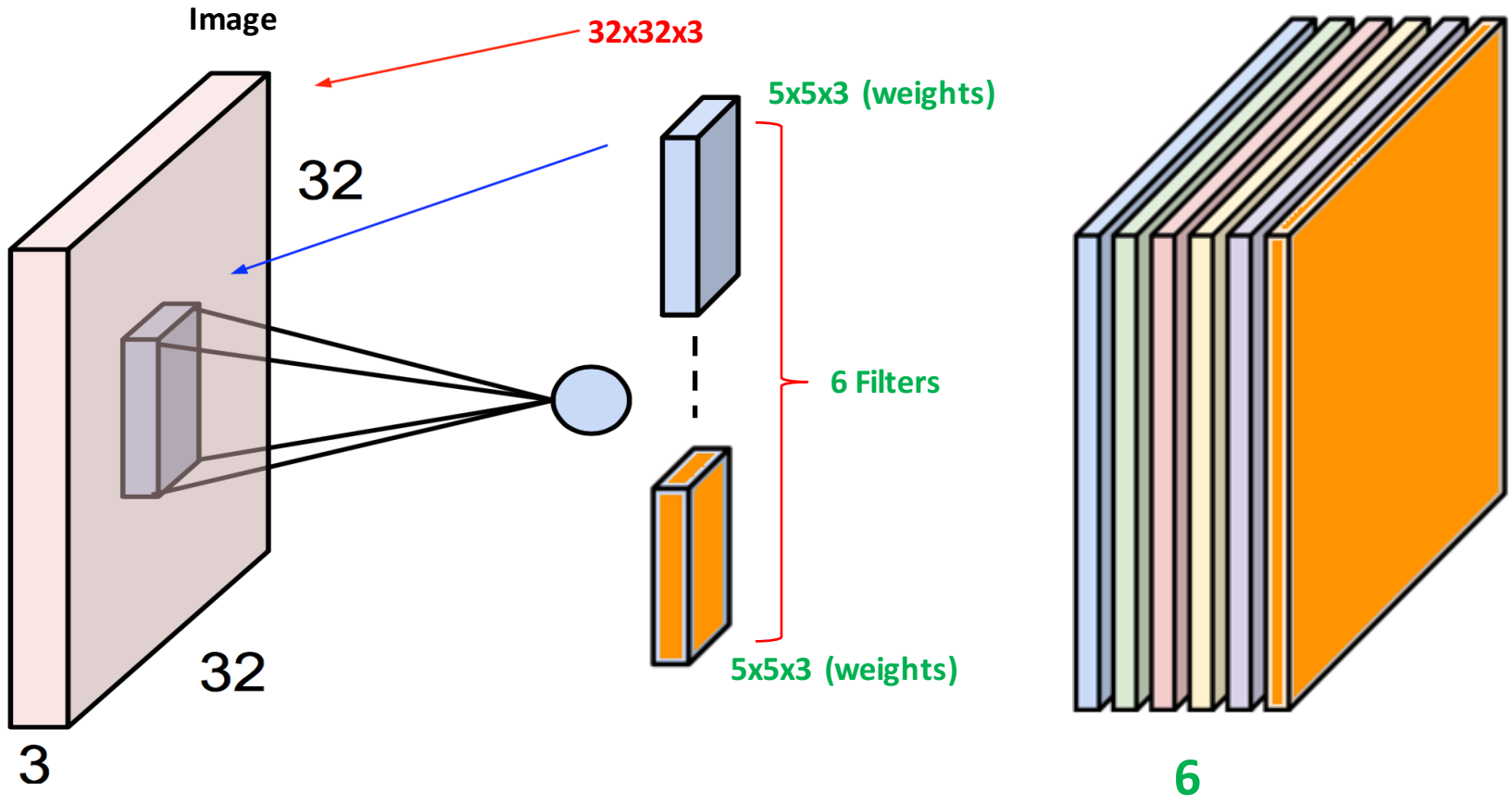
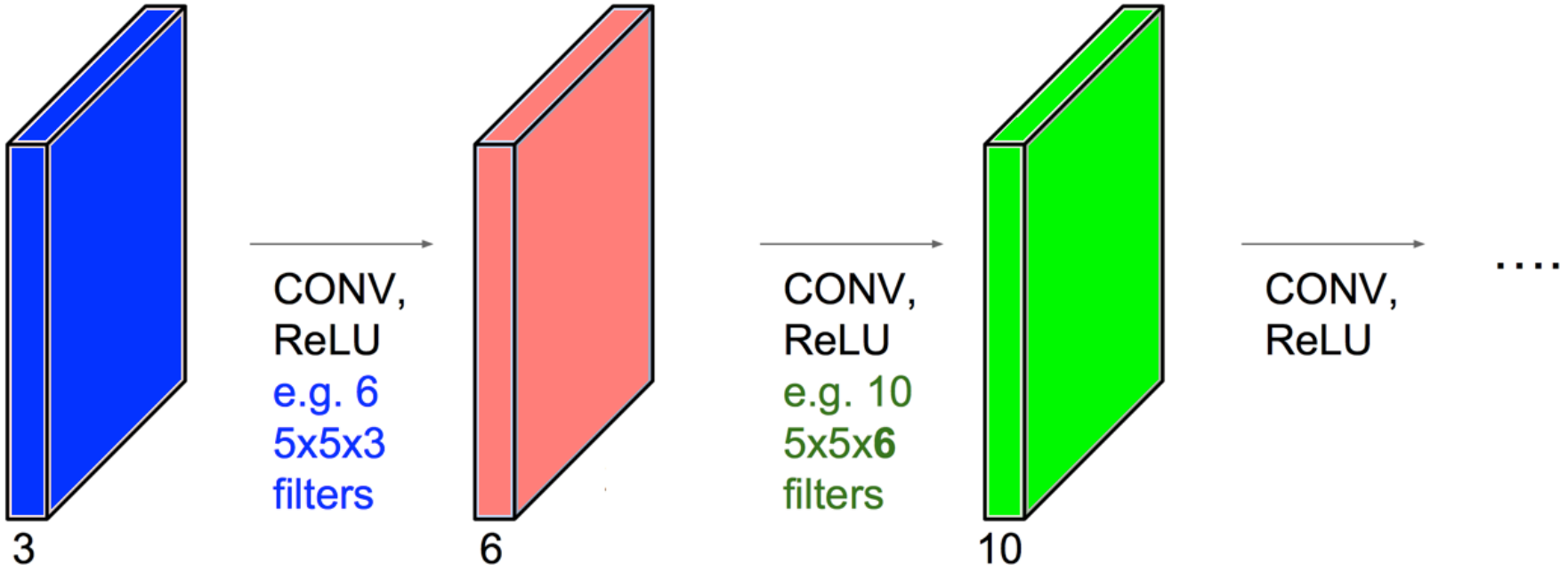
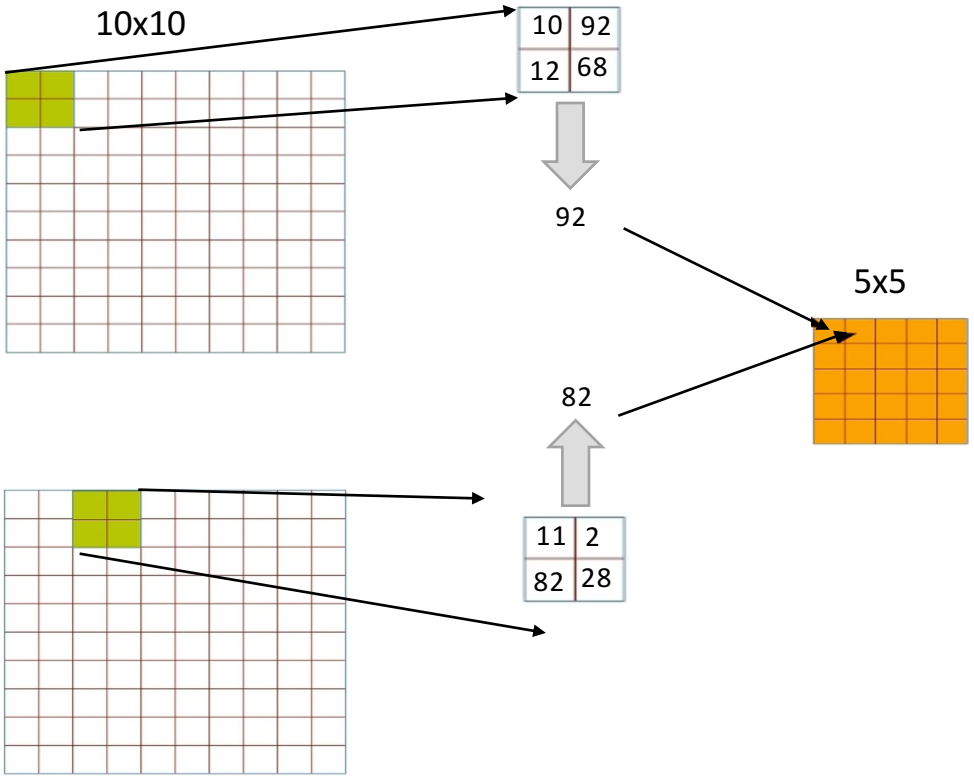


Image
32x32x3
5x5x3 (weights)
32
6 Filters
32
5x5x3 (weights)
3
6

Convolution Network



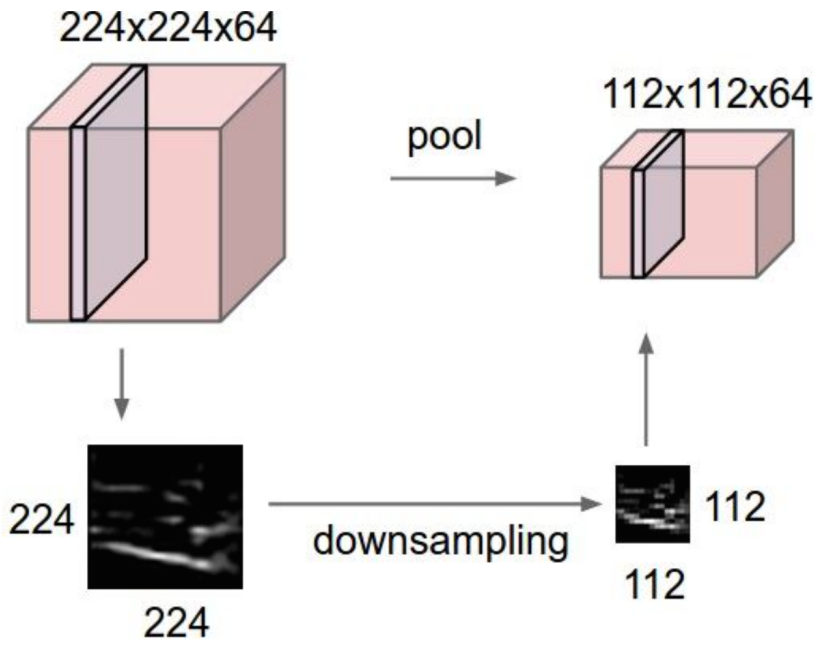
Downsampling = Max Pooling

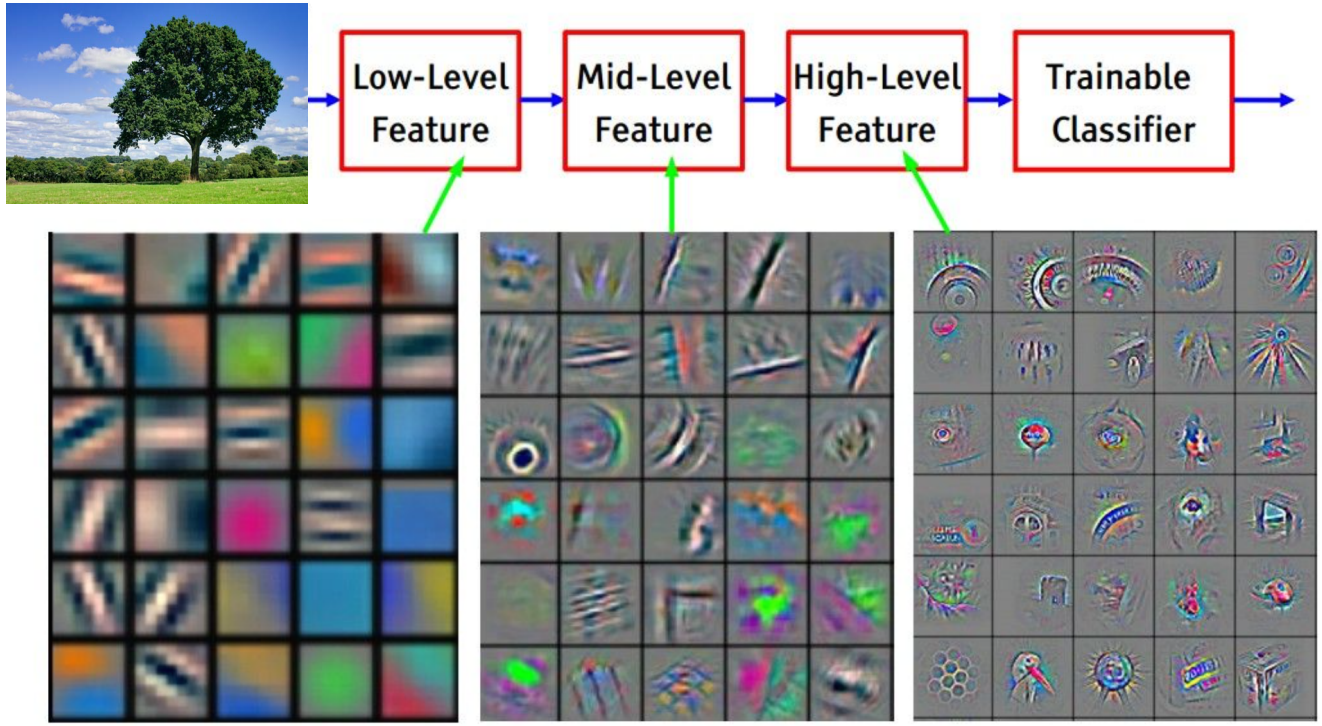


“Max Pooling” Layers to extract the “best” local feature

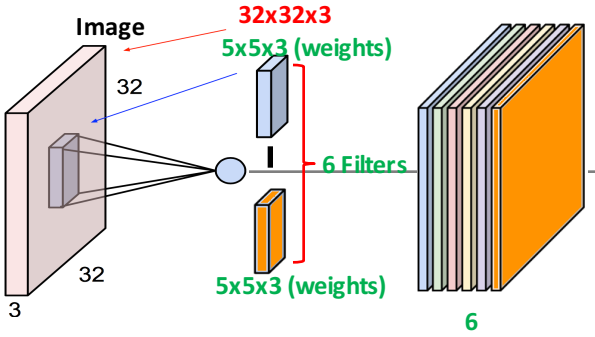
Pooling Layer

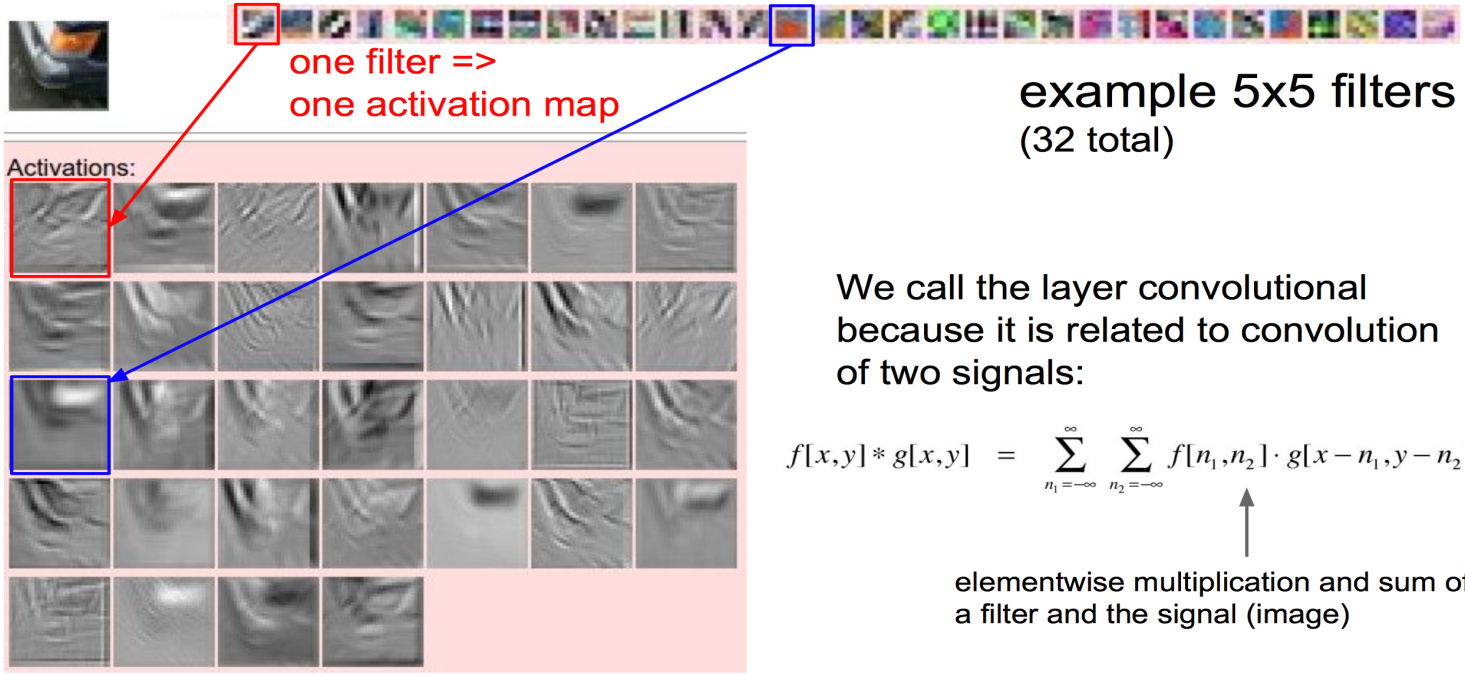
- Makes the representations smaller and more manageable
- Operates over each activation map independently





Feature visualization of convolution net trained on ImageNet from [Zeiler & Fergus 2013]





example 5x5 filters
(32 total)

We call the layer convolutional because it is related to convolution of two signals:

$$f[x,y] * g[x,y] = \sum_{n_1=-\infty}^{\infty} \sum_{n_2=-\infty}^{\infty} f[n_1,n_2] \cdot g[x-n_1, y-n_2]$$

elementwise multiplication and sum of a filter and the signal (image)

Architecture of LeNet-5, Convolution Neural Network

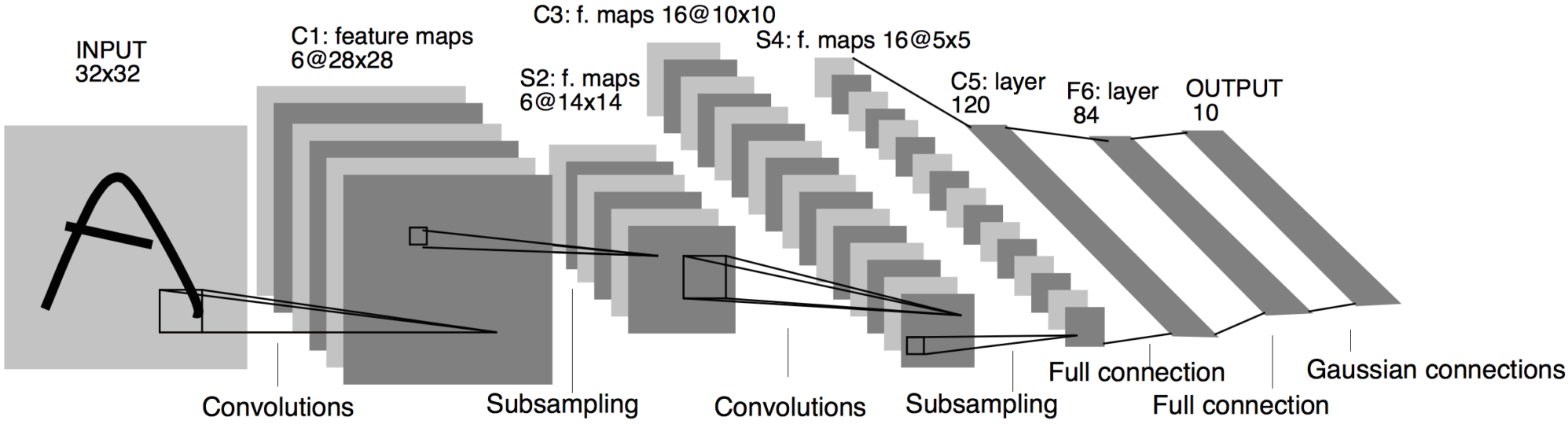
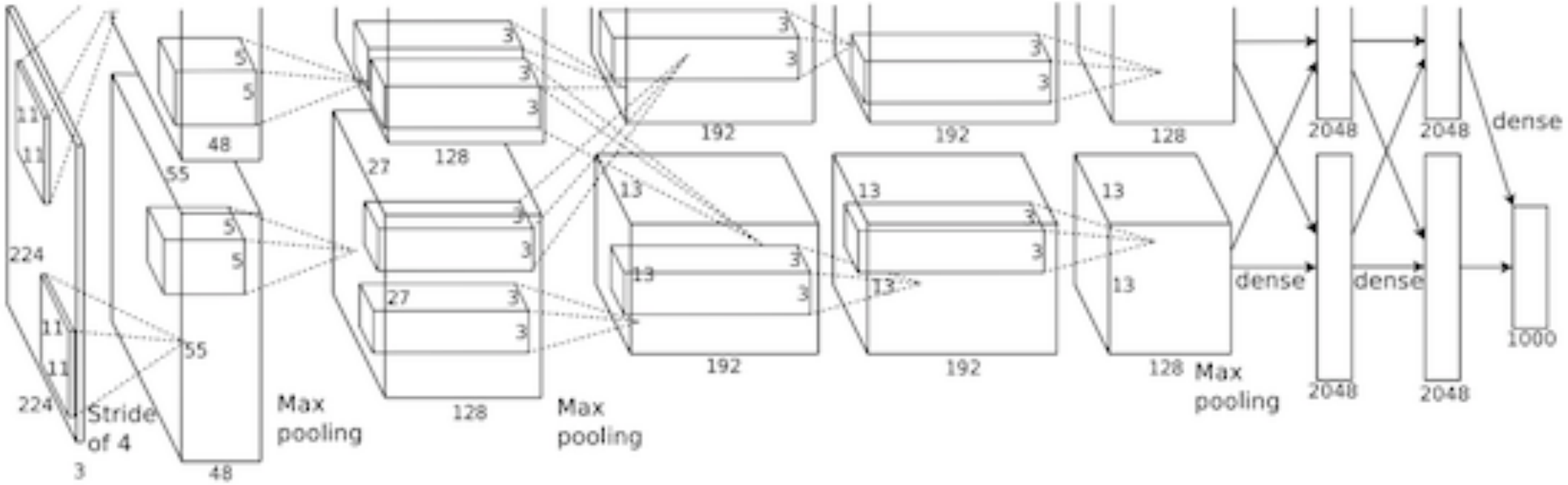


Fig. 2. Architecture of LeNet-5, a Convolutional Neural Network, here for digits recognition. Each plane is a feature map, i.e. a set of units whose weights are constrained to be identical.

Proc. Of the IEEE, November 1998, "Gradient-Based Learning Applied to Document Recognition"

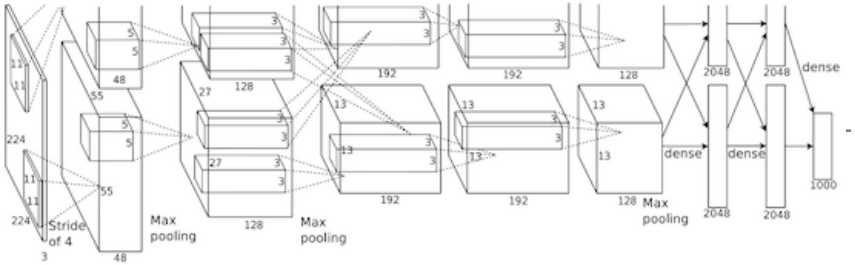
Case Study: AlexNet



Input 227x227x3 images
First Layer (Conv1): 96 11x11 filters applied at stride 4
What is the output volume size? (N-F)/stride+1=55
Output volume : [55x55x96]
Parameters: (11*11*3)*96 = 35k

https://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf

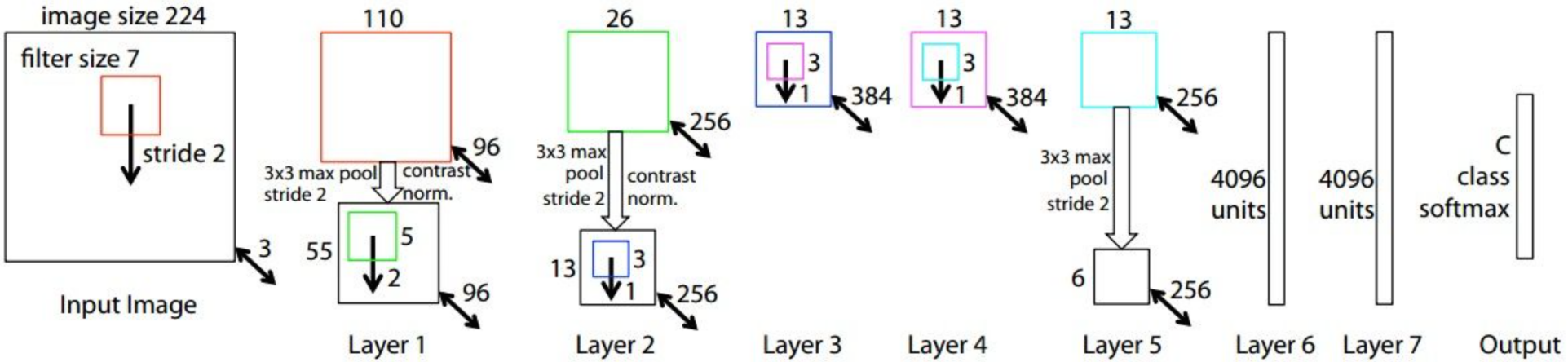
AlexNet



Full (simplified) AlexNet architecture:
[227x227x3] INPUT
[55x55x96] CONV1: 96 11x11 filters at stride 4, pad 0
[27x27x96] MAX POOL1: 3x3 filters at stride 2
[27x27x96] NORM1: Normalization layer
[27x27x256] CONV2: 256 5x5 filters at stride 1, pad 2
[13x13x256] MAX POOL2: 3x3 filters at stride 2
[13x13x256] NORM2: Normalization layer
[13x13x384] CONV3: 384 3x3 filters at stride 1, pad 1
[13x13x384] CONV4: 384 3x3 filters at stride 1, pad 1
[13x13x256] CONV5: 256 3x3 filters at stride 1, pad 1
[6x6x256] MAX POOL3: 3x3 filters at stride 2
[4096] FC6: 4096 neurons
[4096] FC7: 4096 neurons
[1000] FC8: 1000 neurons (class scores)

Details/Retrospectives:
- first use of ReLU
- used Norm layers (not common anymore)
- heavy data augmentation
- dropout 0.5
- batch size 128
- SGD Momentum 0.9
- Learning rate 1e-2, reduced by 10 manually when val accuracy plateaus
- L2 weight decay 5e-4
- 7 CNN ensemble: 18.2% -> 15.4%

ZFNet



Changes to Alex Net:
Conv1: change from (11x11 stride 4) to (7x7 stride 2)
Conv 3,4,5: instead of 384, 384, 256 filters use 512, 1025,512

ImageNet (top 5) error: 15.4% → 14.8%

Case Study :VGGNet

Only 3x3 Conv stride 1, pad 1
and 2x2 Max Pool stride 2

Best Model

11.2% top 5 error in 2013

7.3% top 5 error

ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224 × 224 RGB image)					
conv3-64	conv3-64 **LRN**	conv3-64 **conv3-64**	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 **conv3-128**	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 **conv1-256**	conv3-256 conv3-256 **conv3-256**	conv3-256 conv3-256 conv3-256 **conv3-256**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

Karen Simonyan, Andrew Zisserman, Very Deep Convolutional Networks for Large-Scale Image Recognition, 2014

VGGNet

(not counting biases)

INPUT: [224x224x3] memory: 224*224*3=150K params: 0
CONV3-64: [224x224x64] memory: 224*224*64=3.2M params: (3*3*3)*64 = 1,728
CONV3-64: [224x224x64] memory: 224*224*64=3.2M params: (3*3*64)*64 = 36,864
POOL2: [112x112x64] memory: 112*112*64=800K params: 0
CONV3-128: [112x112x128] memory: 112*112*128=1.6M params: (3*3*64)*128 = 73,728
CONV3-128: [112x112x128] memory: 112*112*128=1.6M params: (3*3*128)*128 = 147,456
POOL2: [56x56x128] memory: 56*56*128=400K params: 0
CONV3-256: [56x56x256] memory: 56*56*256=800K params: (3*3*128)*256 = 294,912
CONV3-256: [56x56x256] memory: 56*56*256=800K params: (3*3*256)*256 = 589,824
CONV3-256: [56x56x256] memory: 56*56*256=800K params: (3*3*256)*256 = 589,824
POOL2: [28x28x256] memory: 28*28*256=200K params: 0
CONV3-512: [28x28x512] memory: 28*28*512=400K params: (3*3*256)*512 = 1,179,648
CONV3-512: [28x28x512] memory: 28*28*512=400K params: (3*3*512)*512 = 2,359,296
CONV3-512: [28x28x512] memory: 28*28*512=400K params: (3*3*512)*512 = 2,359,296
POOL2: [14x14x512] memory: 14*14*512=100K params: 0
CONV3-512: [14x14x512] memory: 14*14*512=100K params: (3*3*512)*512 = 2,359,296
CONV3-512: [14x14x512] memory: 14*14*512=100K params: (3*3*512)*512 = 2,359,296
CONV3-512: [14x14x512] memory: 14*14*512=100K params: (3*3*512)*512 = 2,359,296
POOL2: [7x7x512] memory: 7*7*512=25K params: 0
FC: [1x1x4096] memory: 4096 params: 7*7*512*4096 = 102,760,448
FC: [1x1x4096] memory: 4096 params: 4096*4096 = 16,777,216
FC: [1x1x1000] memory: 1000 params: 4096*1000 = 4,096,000

ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224 × 224 RGB image)					
conv3-64	conv3-64 **LRN**	conv3-64 **conv3-64**	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 **conv3-128**	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 **conv1-256**	conv3-256 **conv3-256**	conv3-256 conv3-256 **conv3-256**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 **conv3-512**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 **conv3-512**
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

Revolution of Depth