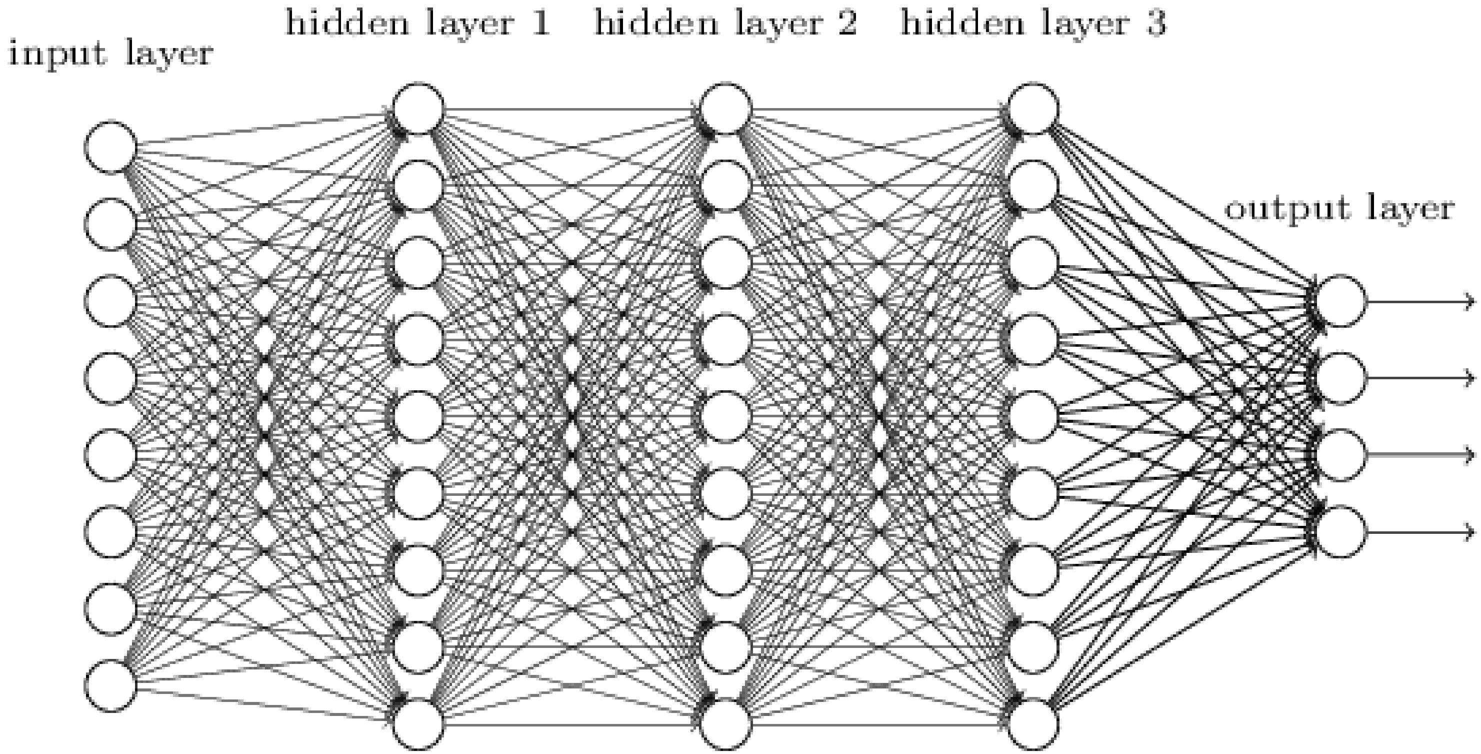
Convolutional Networks

Mohsen Afsharchi

Fully Connected Multilayer NN



The Origin of Convolutional Networks

- The origins to the 1970s.

- Modern subject of convolutional networks was a 1998 paper, "Gradient-based learning applied to document recognition", by Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner

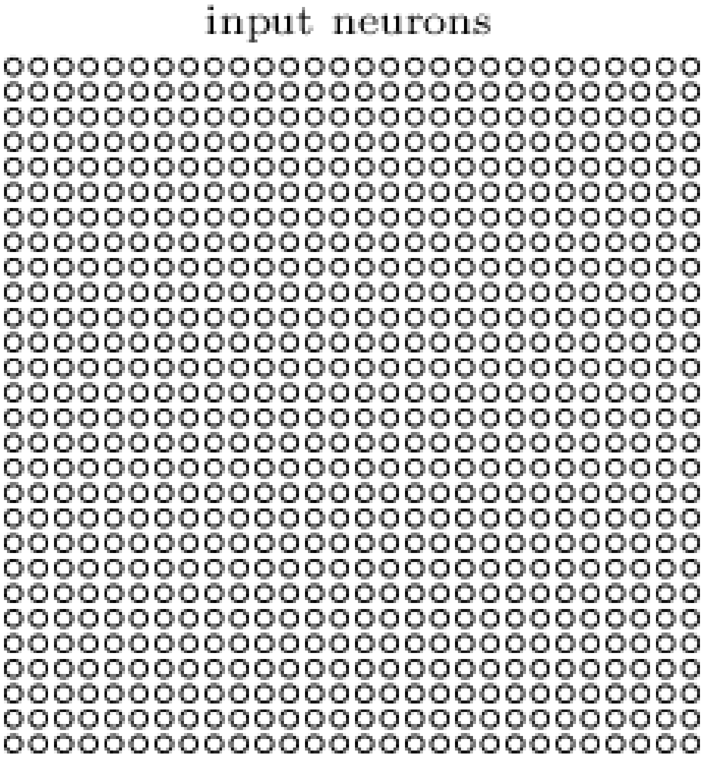
- “The [biological] neural inspiration in models like convolutional nets is very tenuous. That's why I call them 'convolutional nets' not 'convolutional neural nets' ”
- Convolutional nets use many of the same ideas as the neural networks we've studied up to now

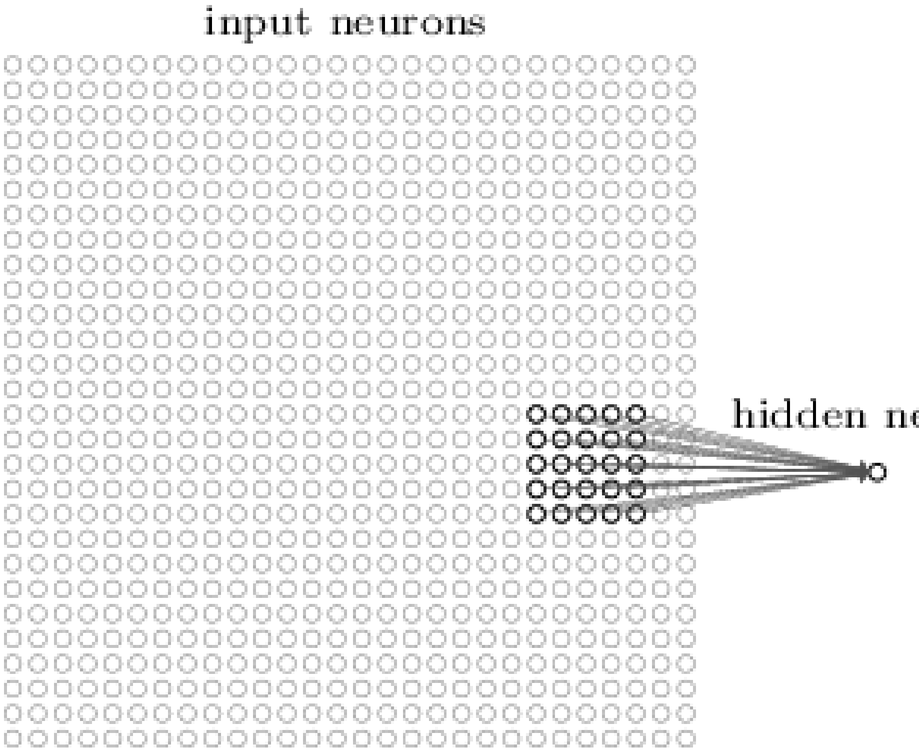
Basic Ideas

- Sparse Interactions
  - Spatial Arrangement

- Parameter Sharing
  - Shared Weights

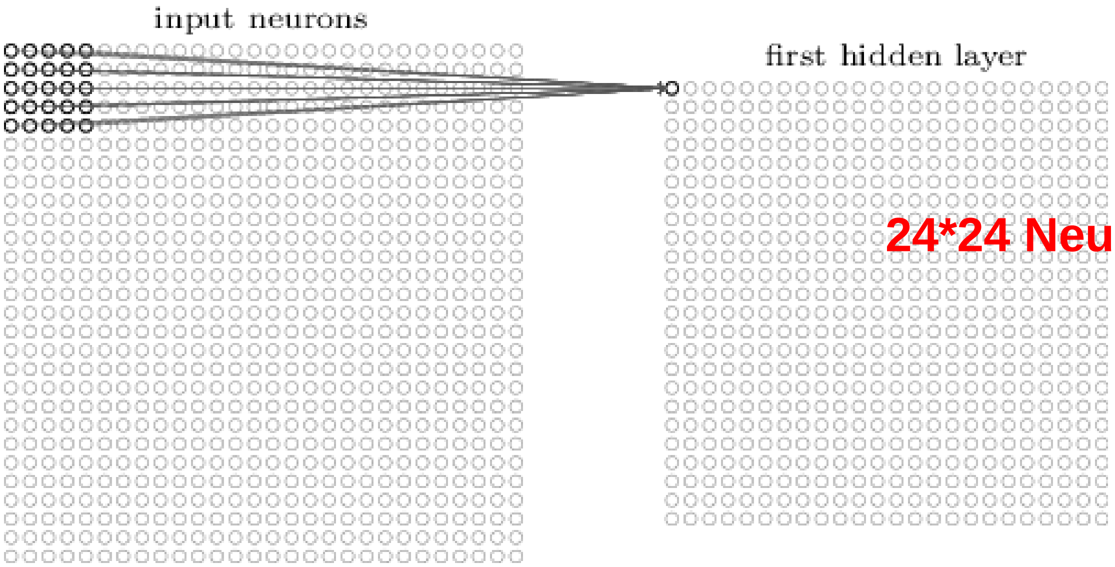
Local Receptive Fields





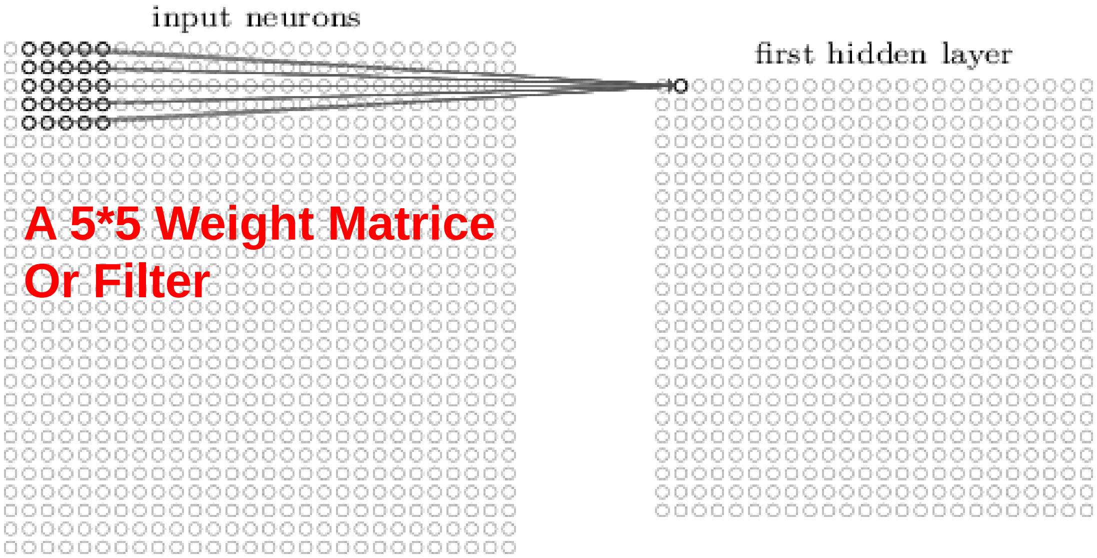
A 28*28 image

Convolution Layer



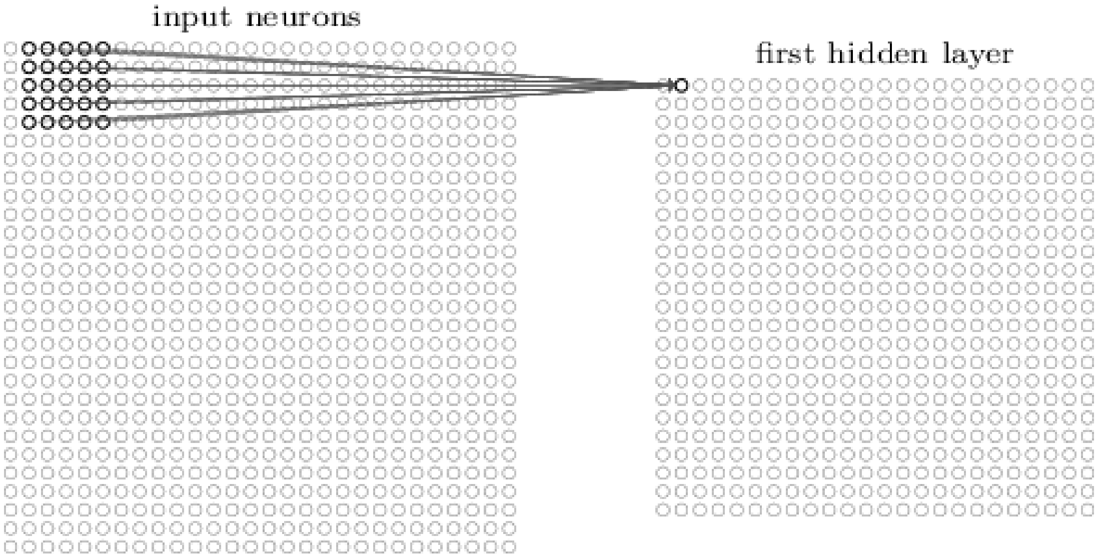
24*24 Neurons in Hidden Layer

A 28*28 image



A 5*5 Weight Matrice
Or Filter

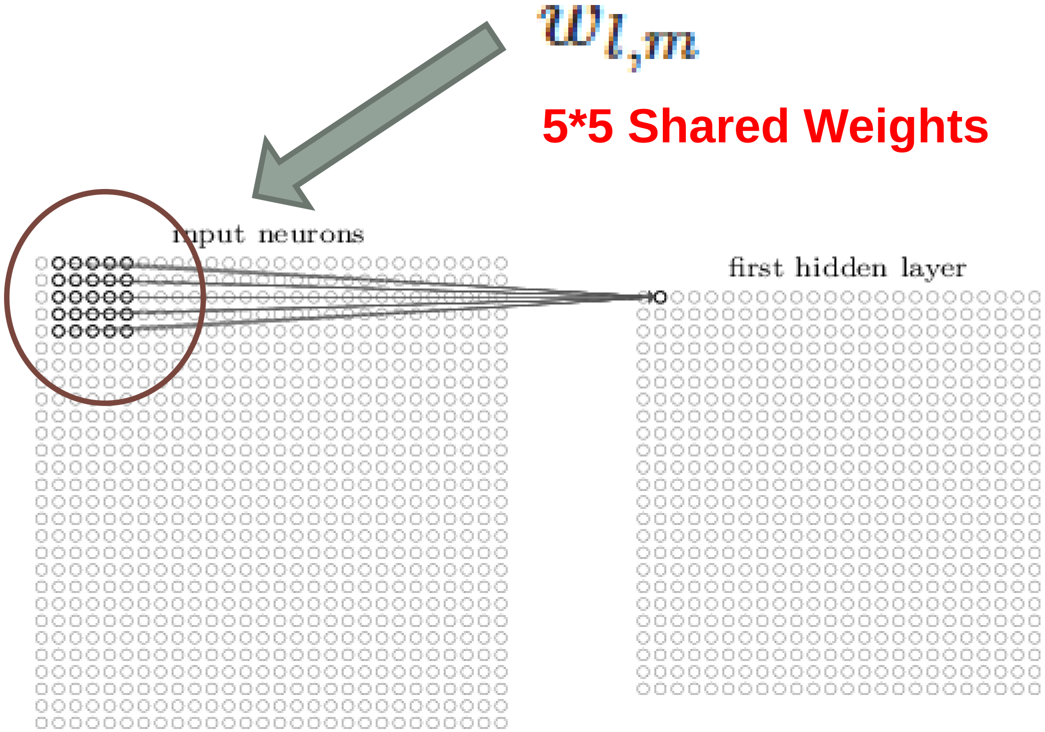
Convolution



$$\sigma\left(b + \sum_{l=0}^{4}\sum_{m=0}^{4} w_{l,m} a_{j+l,k+m}\right)$$

Could be a Sigmoid Function

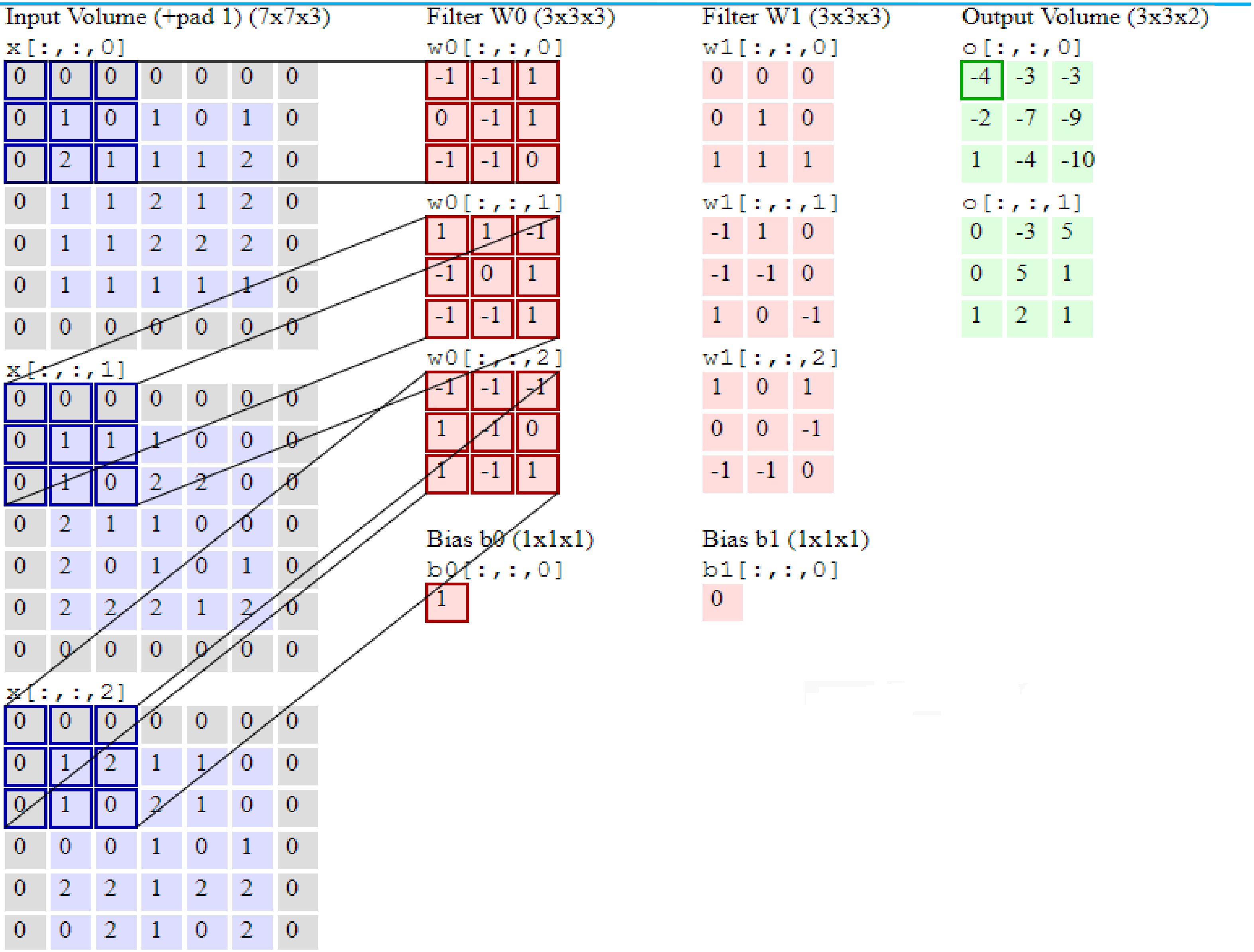
Shared Weights and Bias



$w_{l,m}$

5*5 Shared Weights

Example



Input Volume (+pad 1) (7x7x3)

x[:,:,0]

0	0	0	0	0	0	0
0	1	0	1	0	1	0
0	2	1	1	1	2	0
0	1	1	2	1	2	0
0	1	1	2	2	2	0
0	1	1	1	1	1	0
0	0	0	0	0	0	0

x[:,:,1]

0	0	0	0	0	0	0
0	1	1	1	0	0	0
0	1	0	2	2	0	0
0	2	1	1	0	0	0
0	2	0	1	0	1	0
0	2	2	2	1	2	0
0	0	0	0	0	0	0

x[:,:,2]

0	0	0	0	0	0	0
0	1	2	1	1	0	0
0	1	0	2	1	0	0
0	0	0	1	0	1	0
0	2	2	1	2	2	0
0	0	2	1	0	2	0

Filter W0 (3x3x3)

w0[:,:,0]

-1	-1	1
0	-1	1
-1	-1	0

w0[:,:,1]

1	1	-1
-1	0	1
-1	-1	1

w0[:,:,2]

-1	-1	-1
1	-1	0
1	-1	1

Bias b0 (1x1x1)

b0[:,:,0]

1

Filter W1 (3x3x3)

w1[:,:,0]

0	0	0
0	1	0
1	1	1

w1[:,:,1]

-1	1	0
-1	-1	0
1	0	-1

w1[:,:,2]

1	0	1
0	0	-1
-1	-1	0

Bias b1 (1x1x1)

b1[:,:,0]

0

Output Volume (3x3x2)

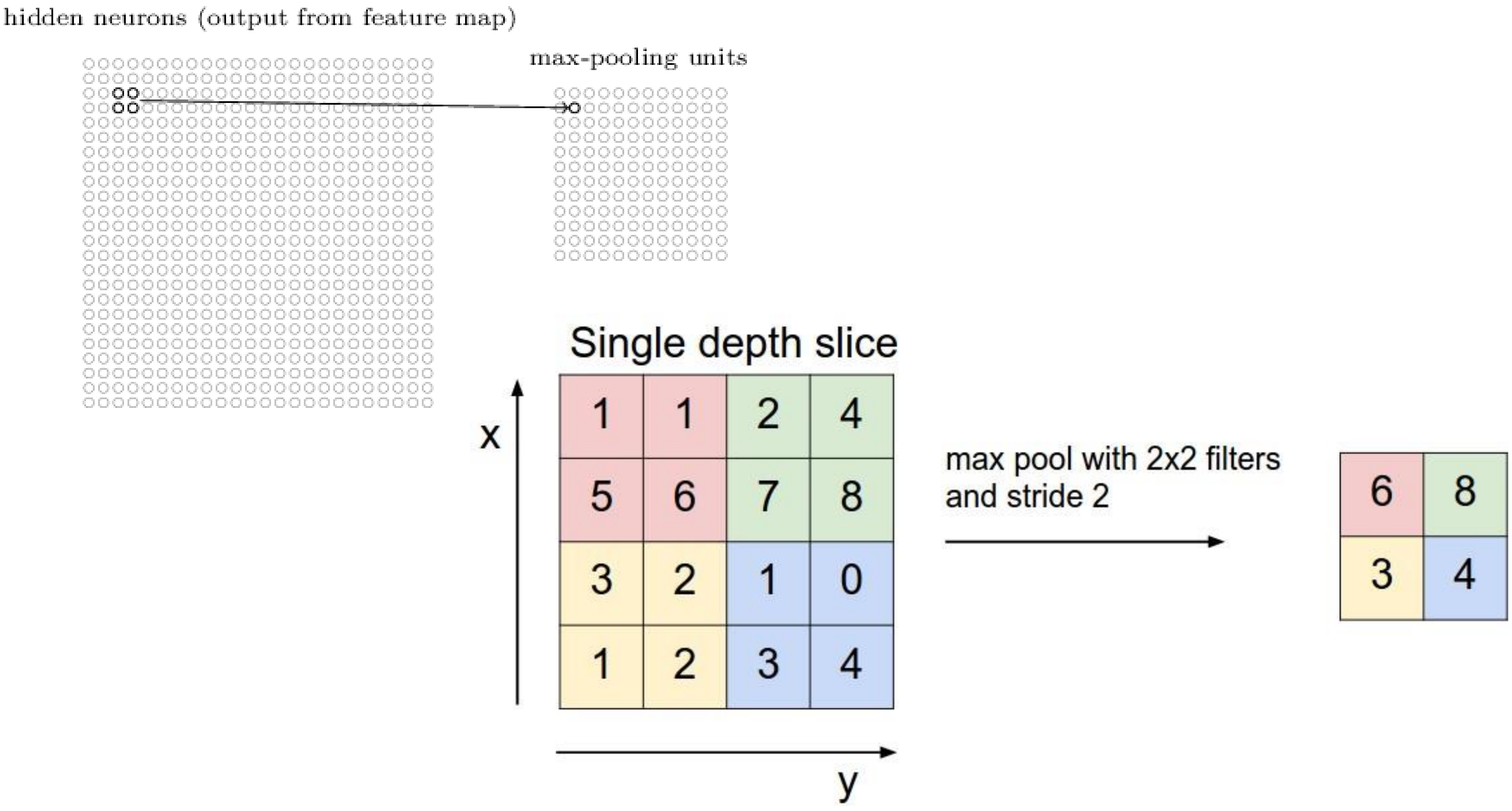
o[:,:,0]

-4	-3	-3
-2	-7	-9
1	-4	-10

o[:,:,1]

0	-3	5
0	5	1
1	2	1

Pooling


hidden neurons (output from feature map)
max-pooling units
Single depth slice
x
y
max pool with 2x2 filters and stride 2

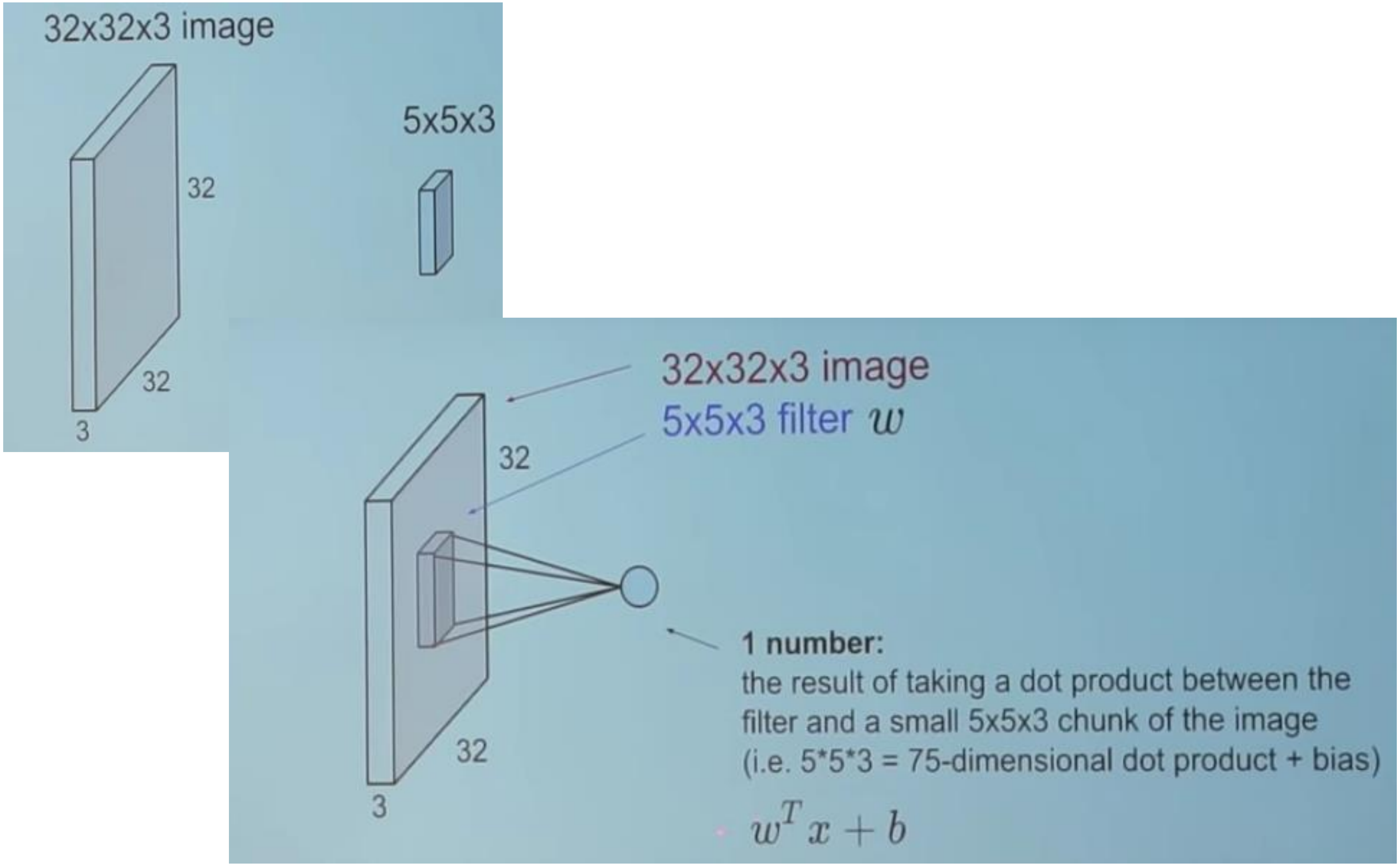
1	1	2	4
5	6	7	8
3	2	1	0
1	2	3	4

6	8
3	4

Layers

- Convolutional Layer
- Pooling Layer
- Fully-Connected Layer

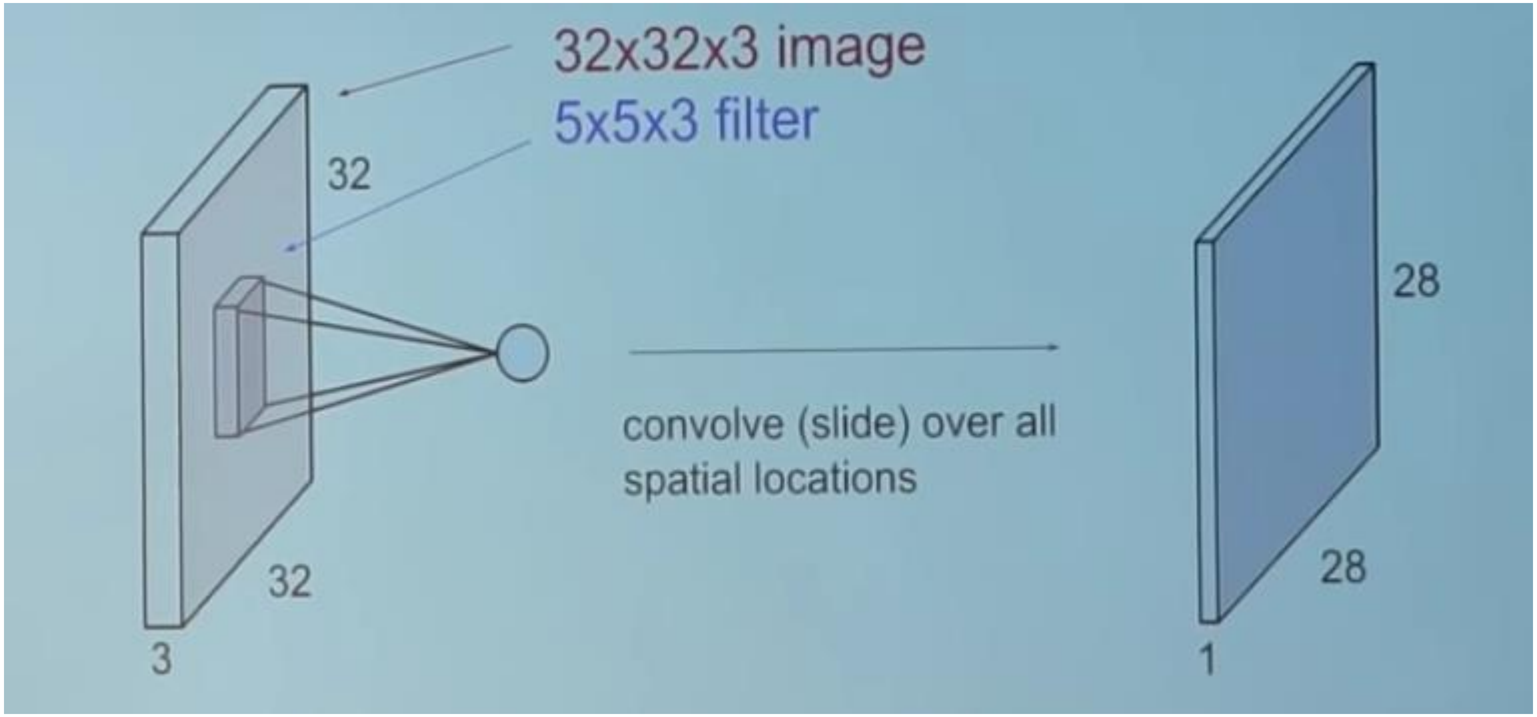
Real Networks



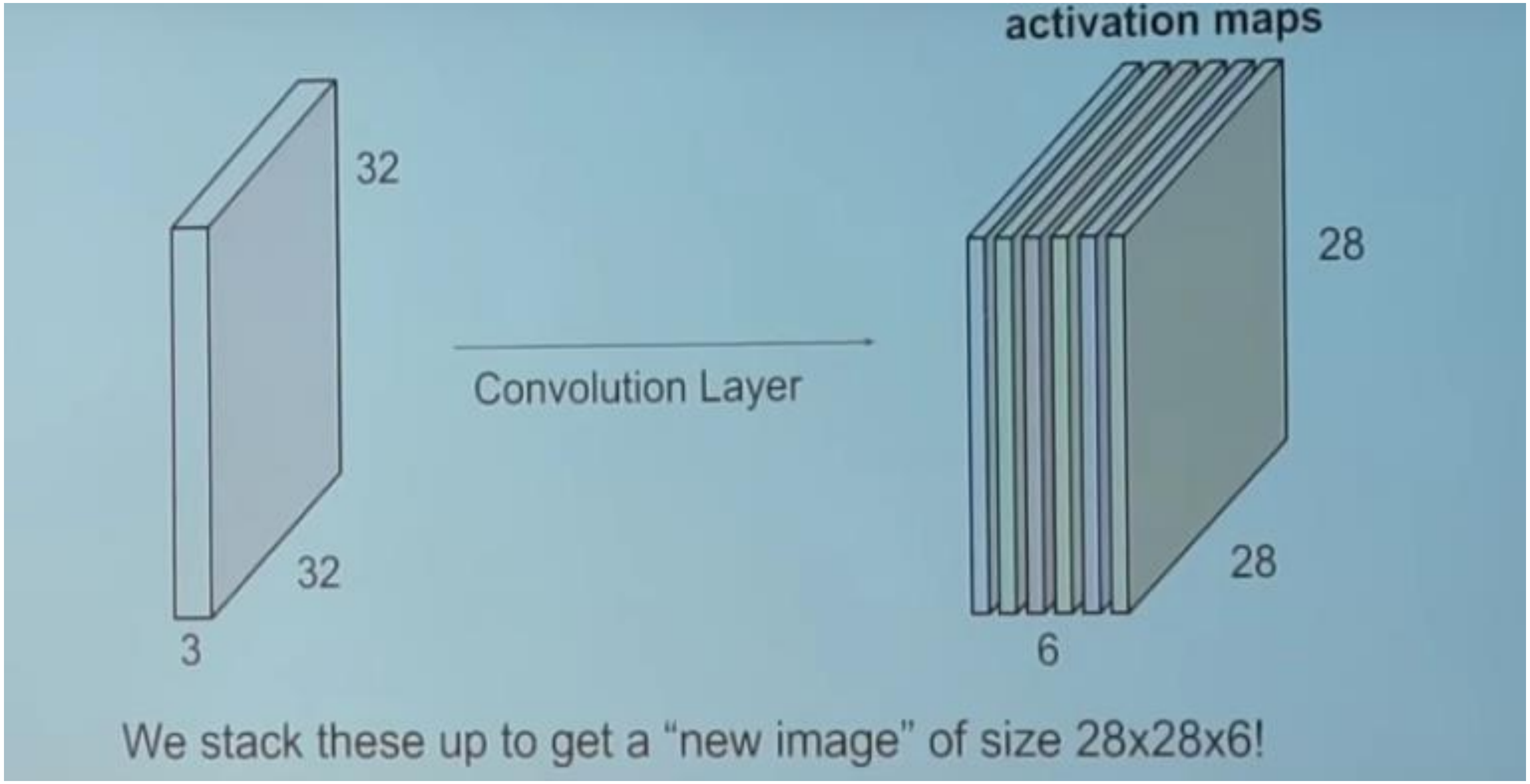
1 number:
the result of taking a dot product between the filter and a small 5x5x3 chunk of the image
(i.e. 5*5*3 = 75-dimensional dot product + bias)

$$w^T x + b$$

Real Networks

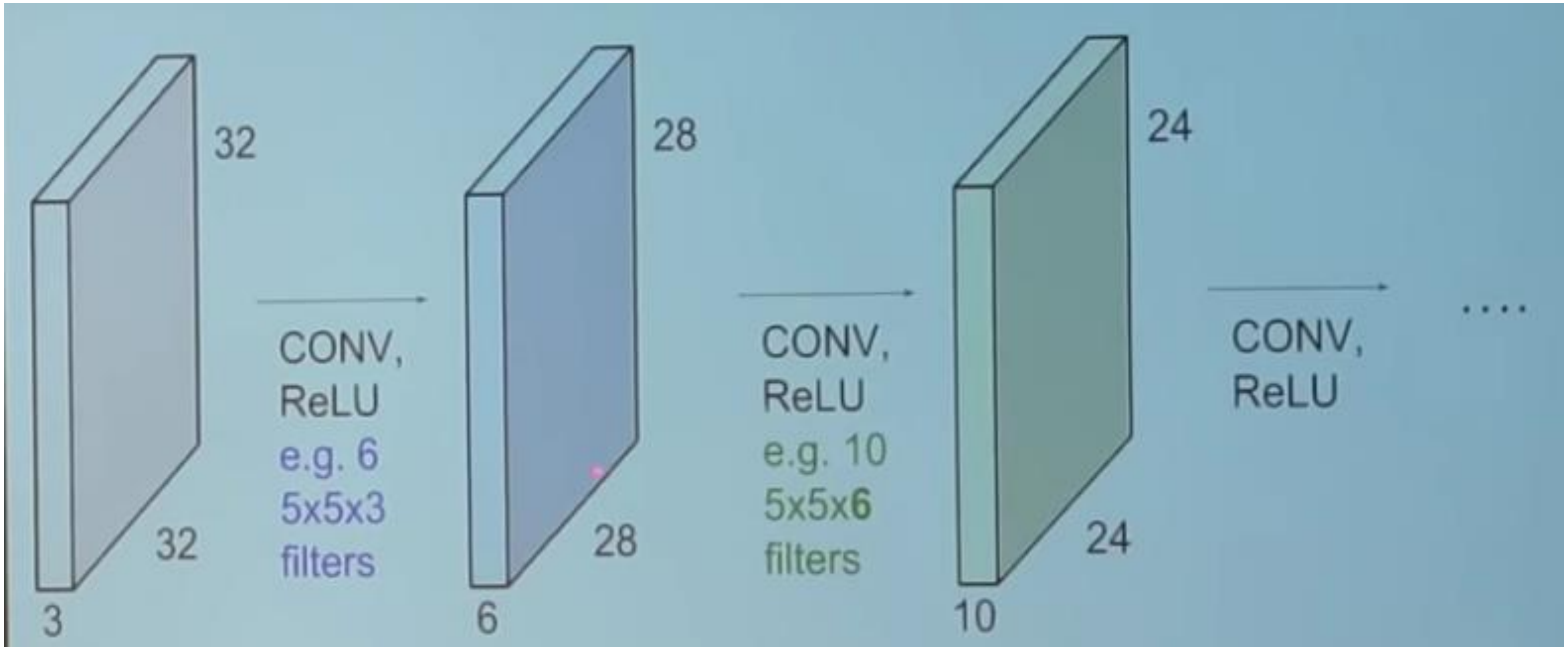


Real Networks

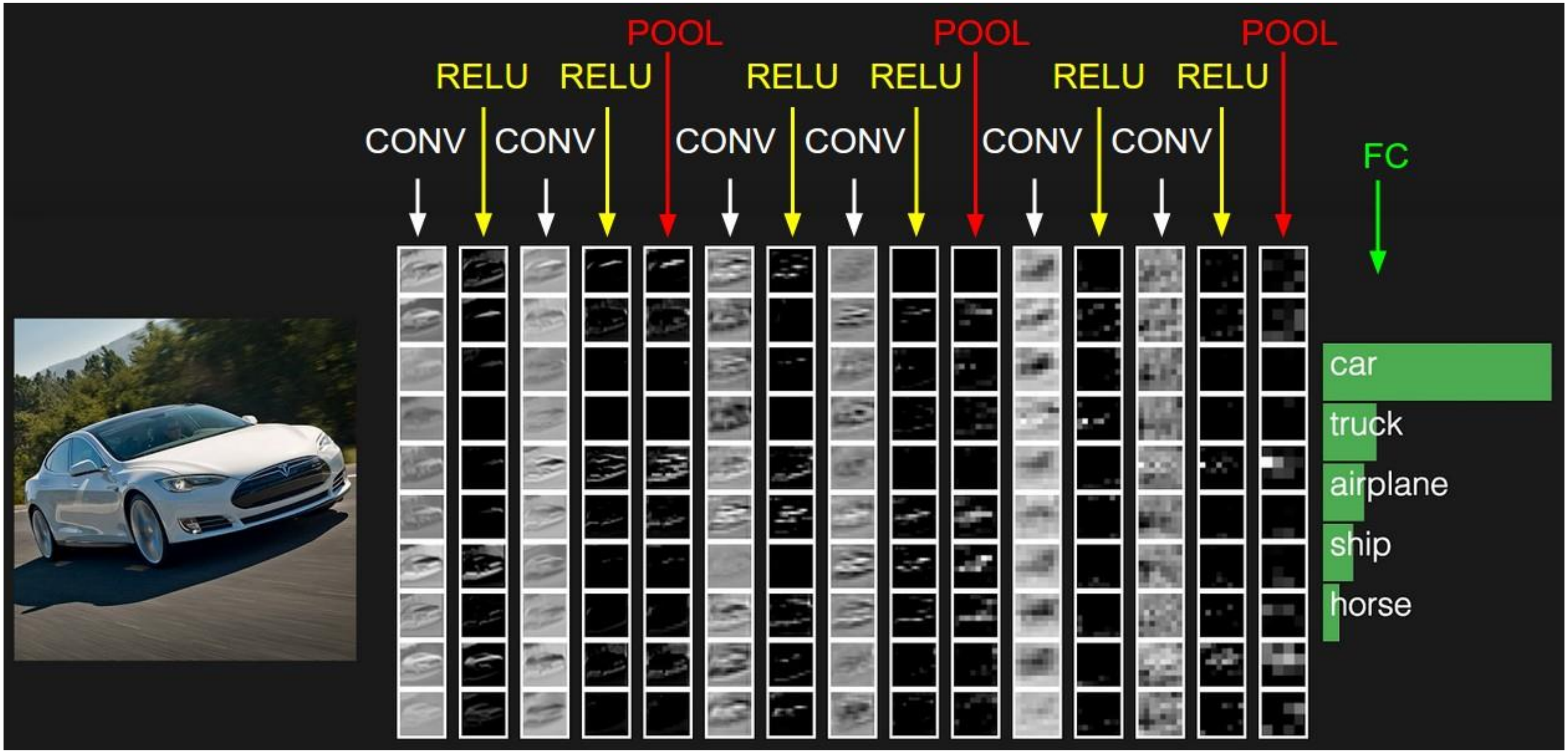


We stack these up to get a “new image” of size 28x28x6!

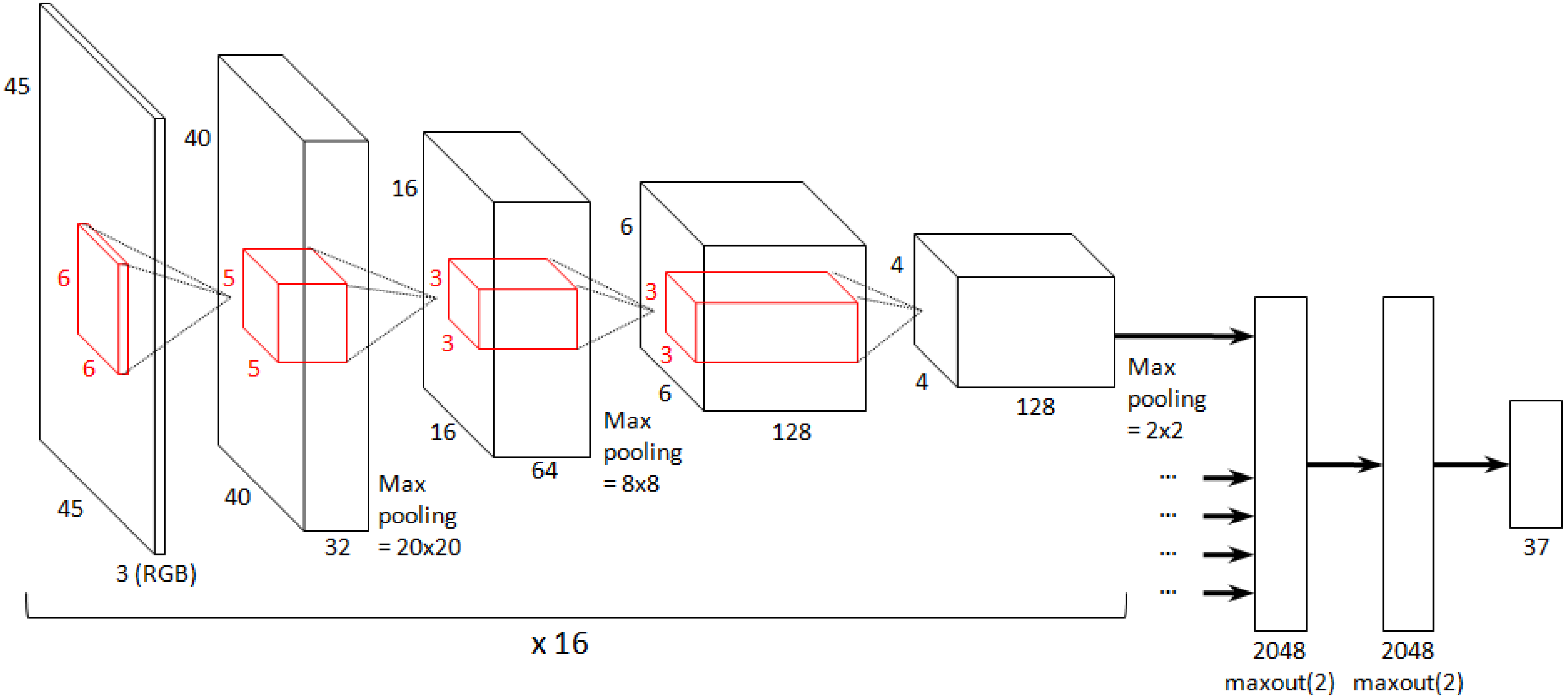
Real Networks



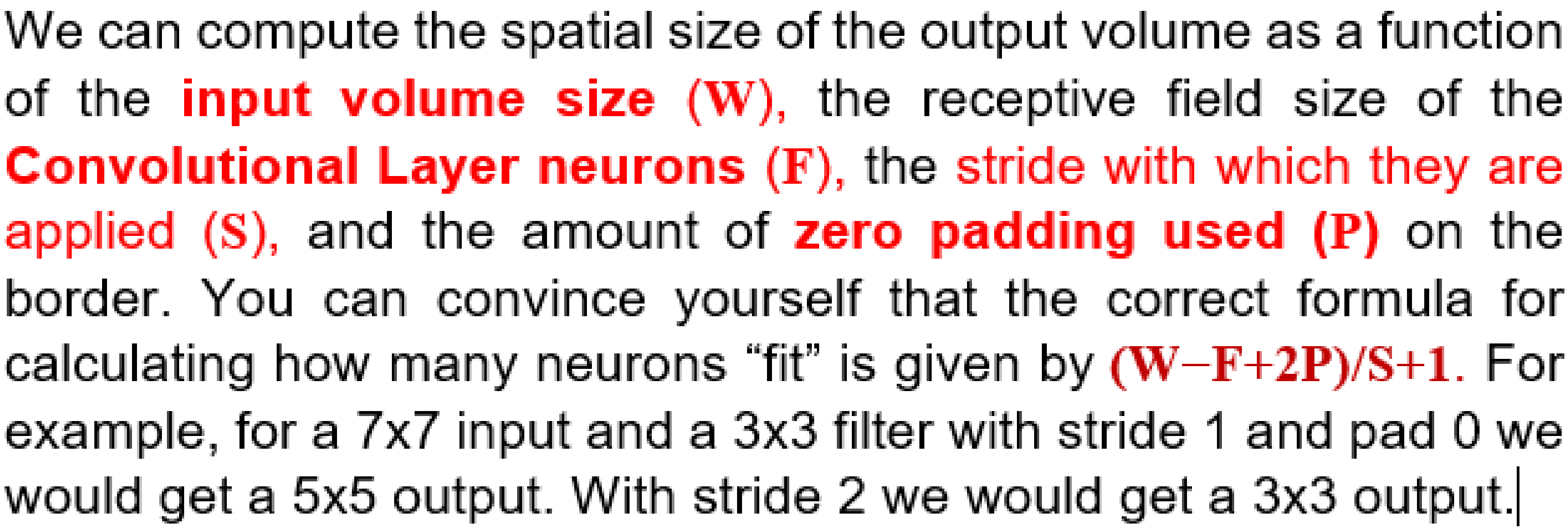
How it Works


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

Example Network

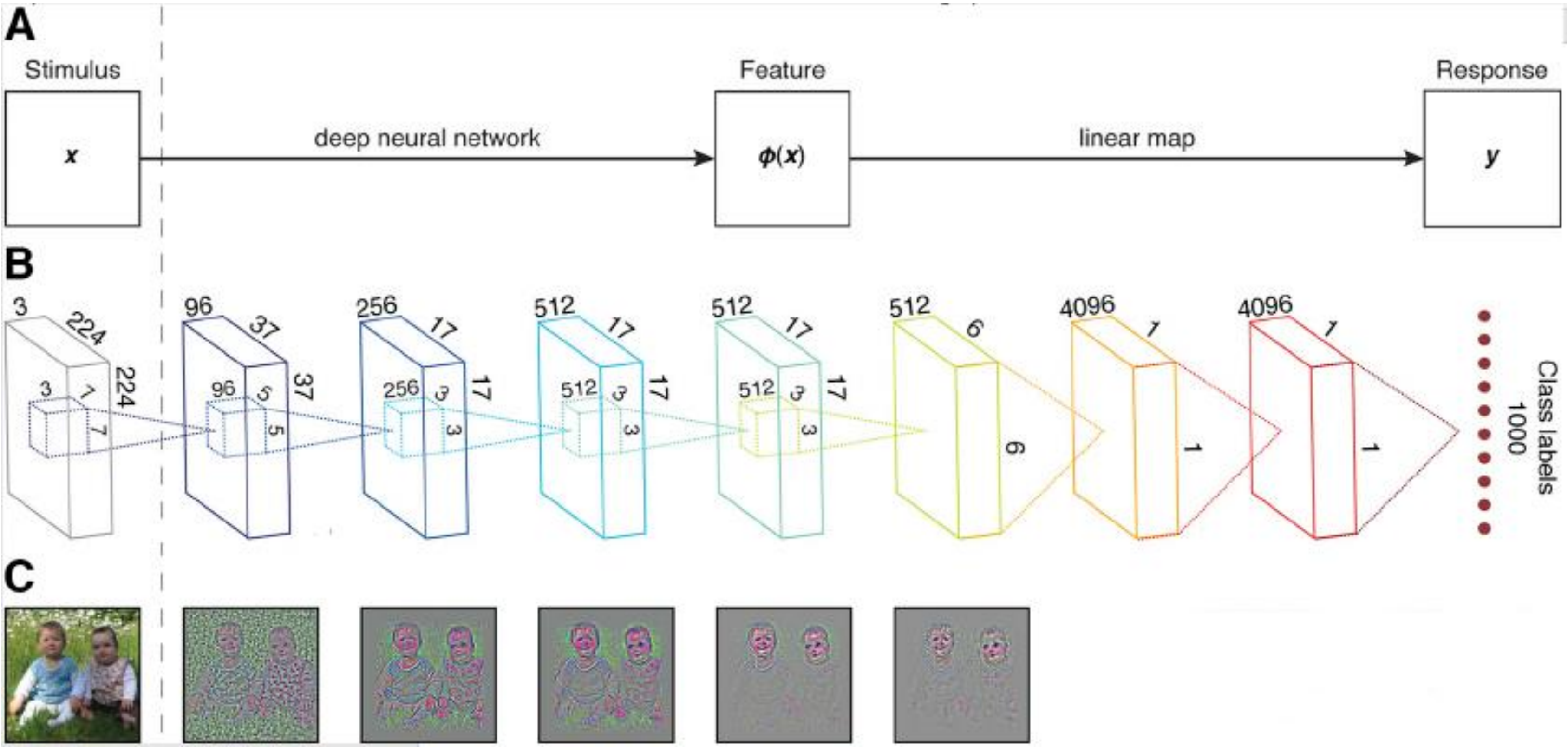


Parameters



We can compute the spatial size of the output volume as a function of the **input volume size (W)**, the receptive field size of the **Convolutional Layer neurons (F)**, the stride with which they are applied (S), and the amount of **zero padding used (P)** on the border. You can convince yourself that the correct formula for calculating how many neurons "fit" is given by **$(W-F+2P)/S+1$**. For example, for a 7x7 input and a 3x3 filter with stride 1 and pad 0 we would get a 5x5 output. With stride 2 we would get a 3x3 output.

Features


A
Stimulus
x
deep neural network
Feature
φ(x)
linear map
Response
y
B
3
224
224
3
7
7
96
37
37
96
5
5
256
17
17
256
3
3
512
17
17
512
3
3
512
17
17
512
3
3
512
6
6
4096
1
1
4096
1
1
Class labels
1000
C