LECTURE 11

CNN

Sung Kim <hunkim+ml@gmail.com>
http://hunkim.github.io/ml

LECTURE 11-1

CNN INTRODUCTION

Sung Kim <hunkim+ml@gmail.com>
http://hunkim.github.io/ml

Convolutional Neural Networks

A Bit of History

Hubel & Wiesel

1959

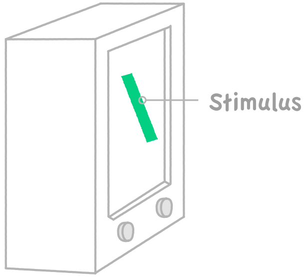
Receptive Fields of Single Neurones in the Cat's Striate Cortex

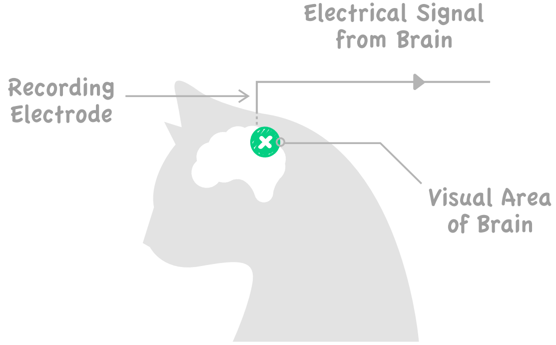
1962

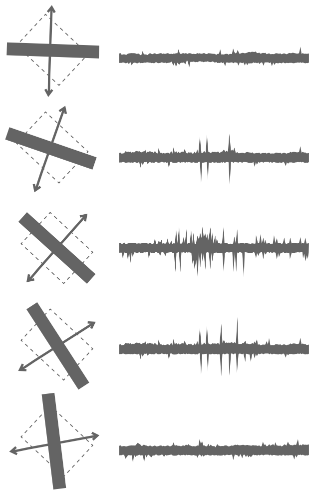
Receptive Fields, Binocular Interaction and Fuctional Architecture in the Cat's Visual Cortex

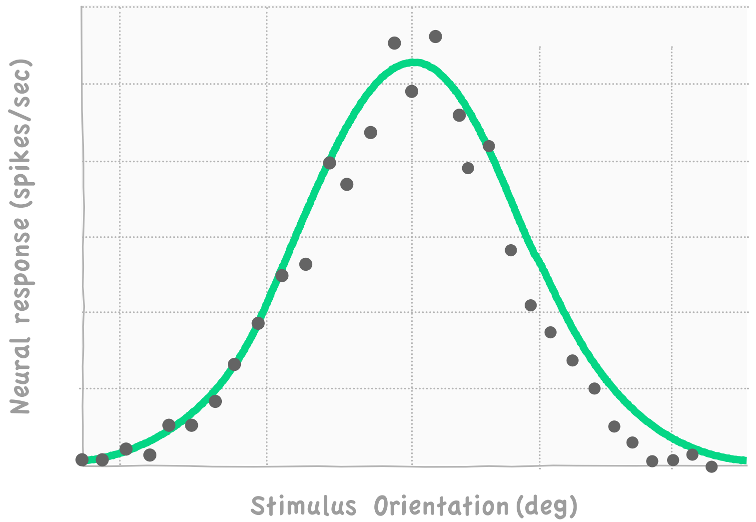
1968

...

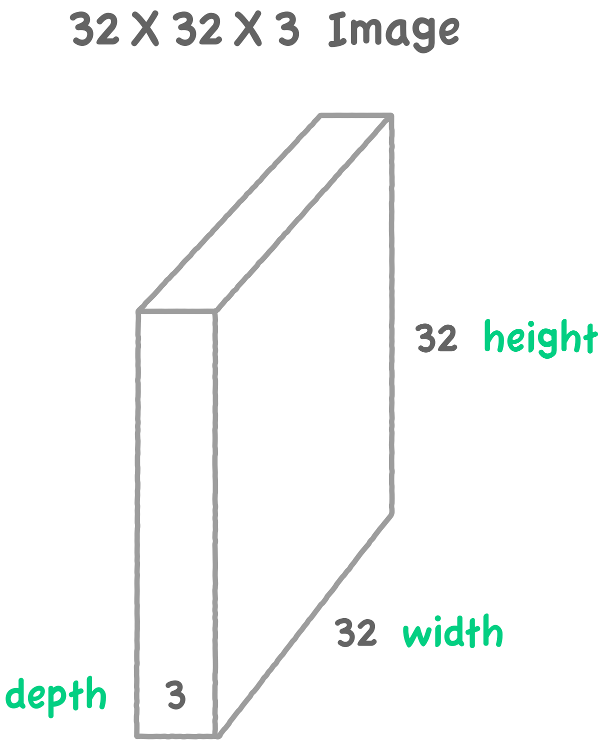






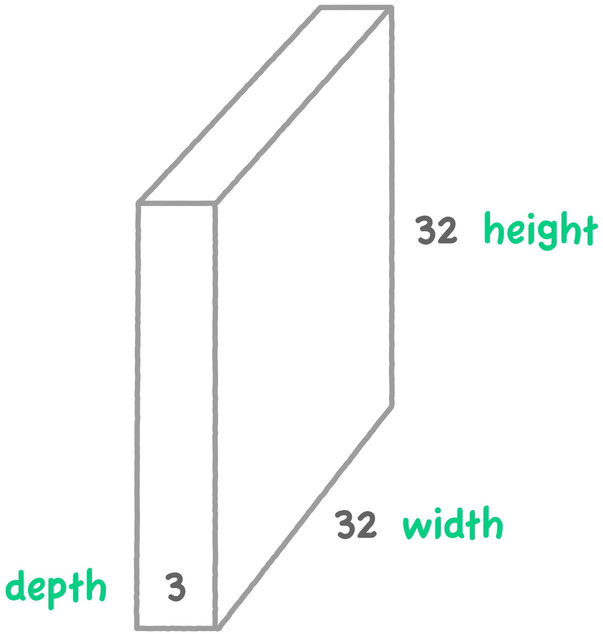


Convolution Layer

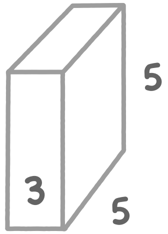


Convolution Layer

32 X 32 X 3 Image



5 X 5 X 3 Filter

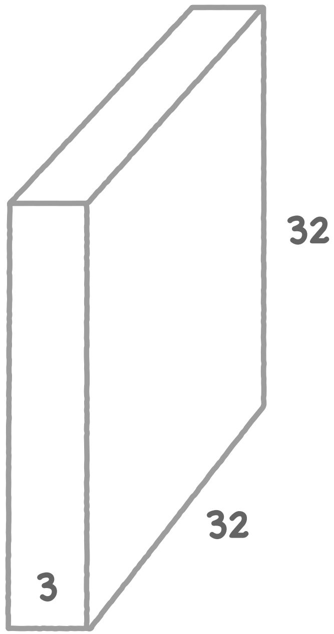


Convolve the Filter with the Image

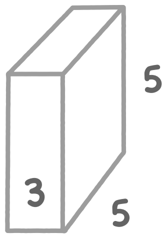
i.e.
"Slide over the Image Spatially,
Computing Dot Products"

Convolution Layer

32 X 32 X 3 Image



5 X 5 X 3 Filter



Filters always
Extend the Full Depth
of the Input Volume

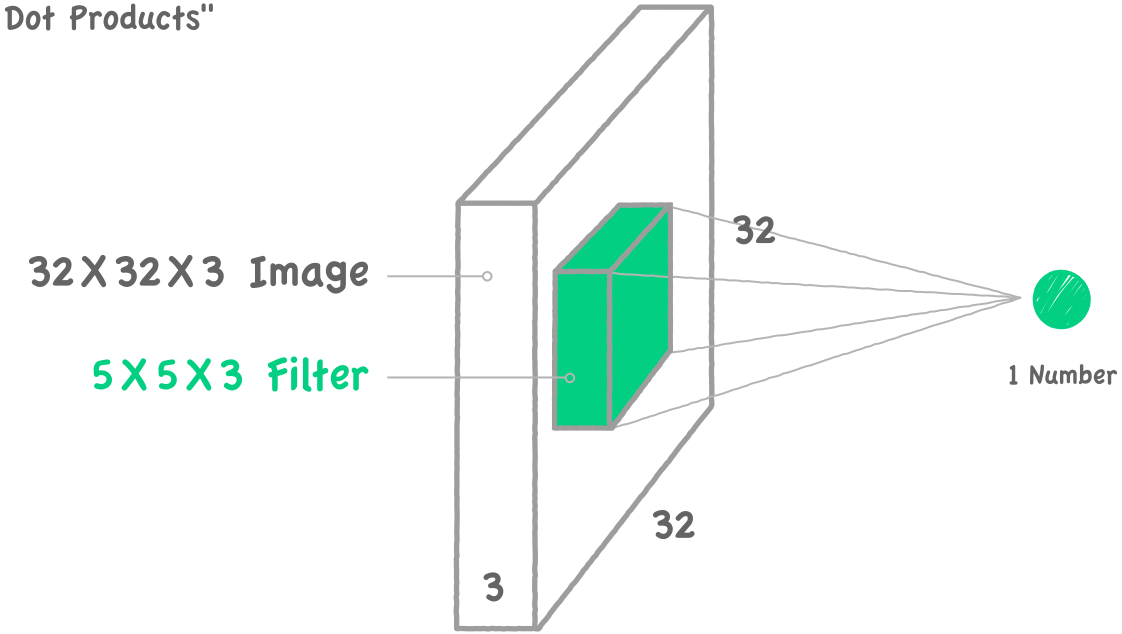
Convolve the Filter with the Image

i.e.
"Slide over the Image Spatially,
Computing Dot Products"

Convolution Layer

Convolve the Filter with the Image

i.e.
"Slide over the Image Spatially,
Computing Dot Products"



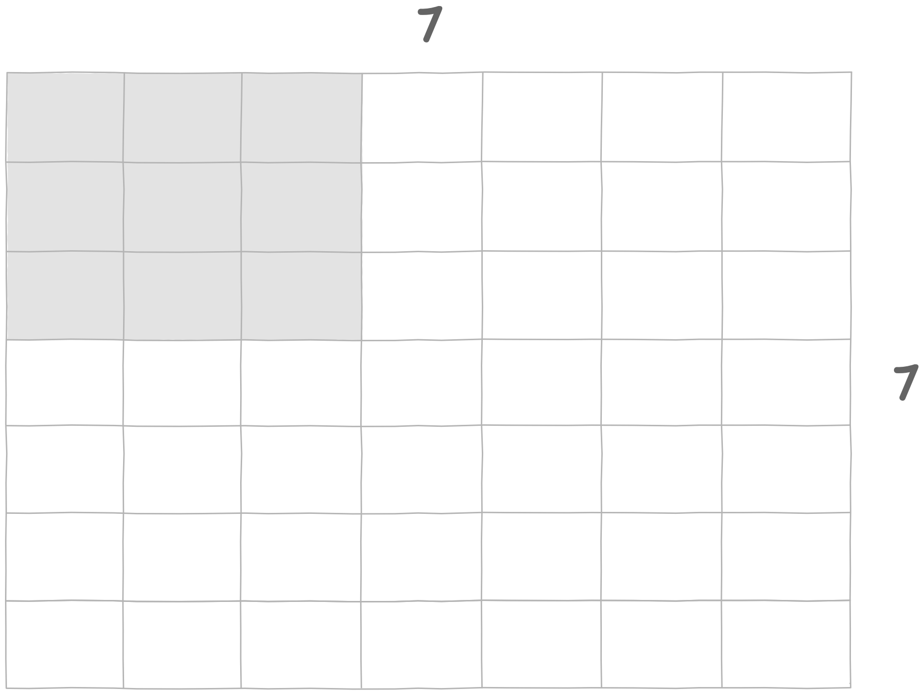
1 number :
The result of taking a dot product between the filter and a small 5x5x3 Chunk of the Image

i.e.
5 x 5 x 3 =
75 - dimensional dot product + bias

$w^T x + b$

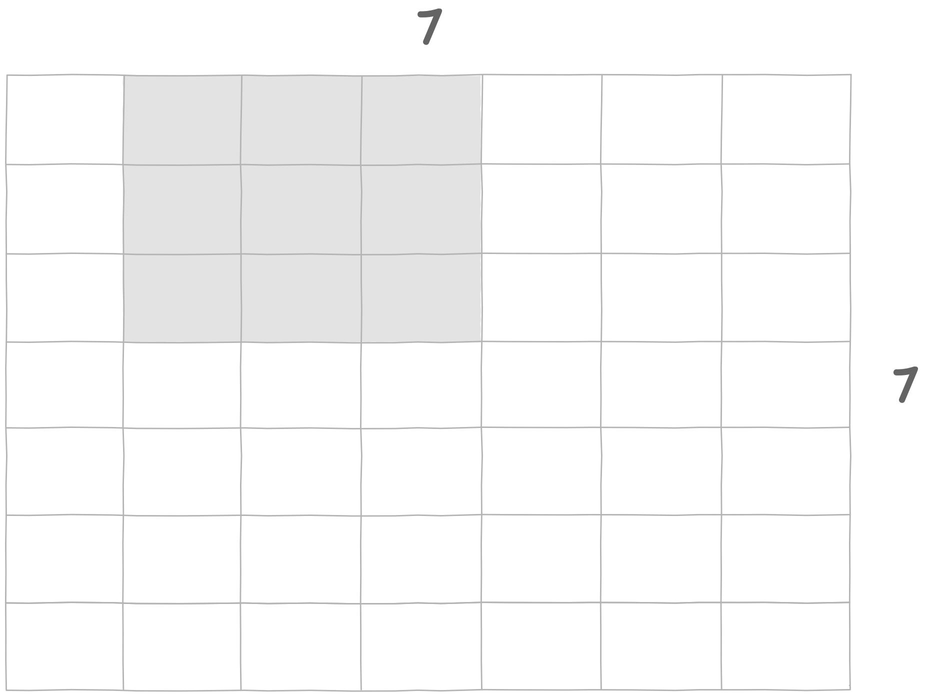
A Closer Look at Spatial Dimensions

7x7 Input (Spatially)
assume 3x3 Filter



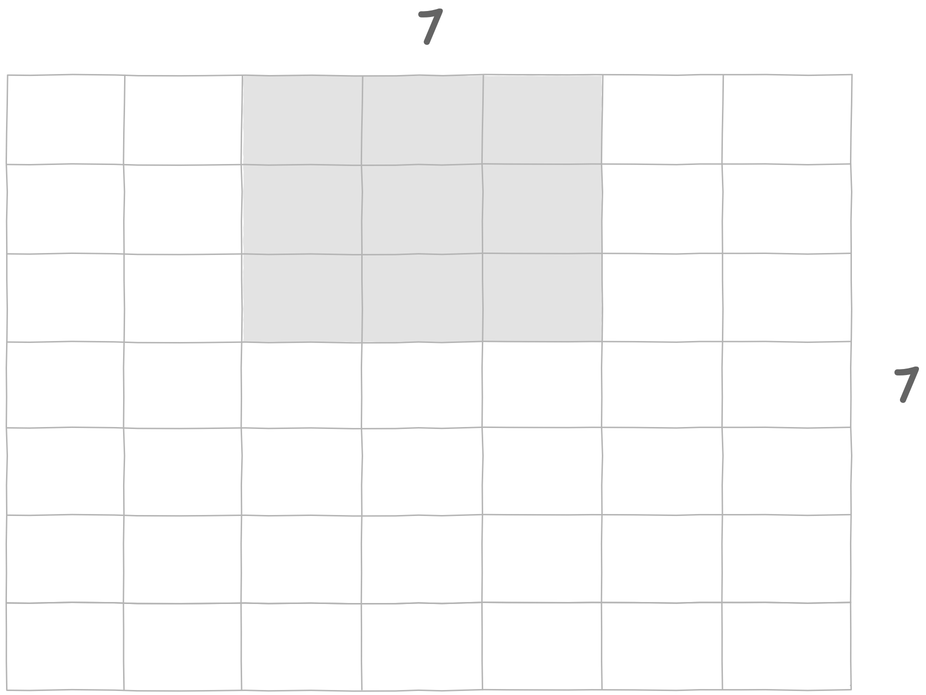
A Closer Look at Spatial Dimensions

7x7 Input (Spatially)
assume 3x3 Filter



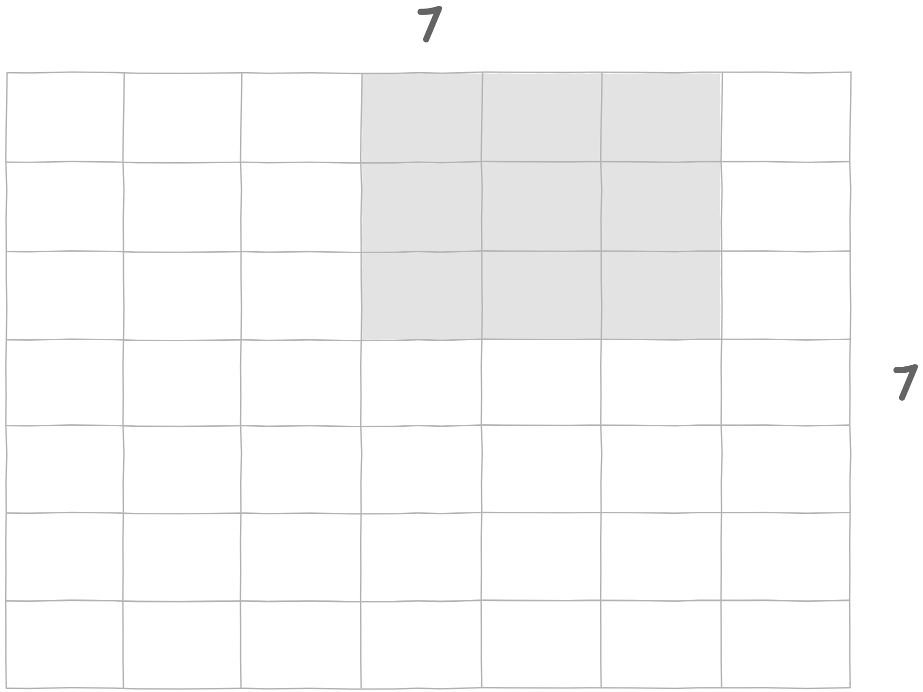
A Closer Look at Spatial Dimensions

7x7 Input (Spatially)
assume 3x3 Filter



A Closer Look at Spatial Dimensions

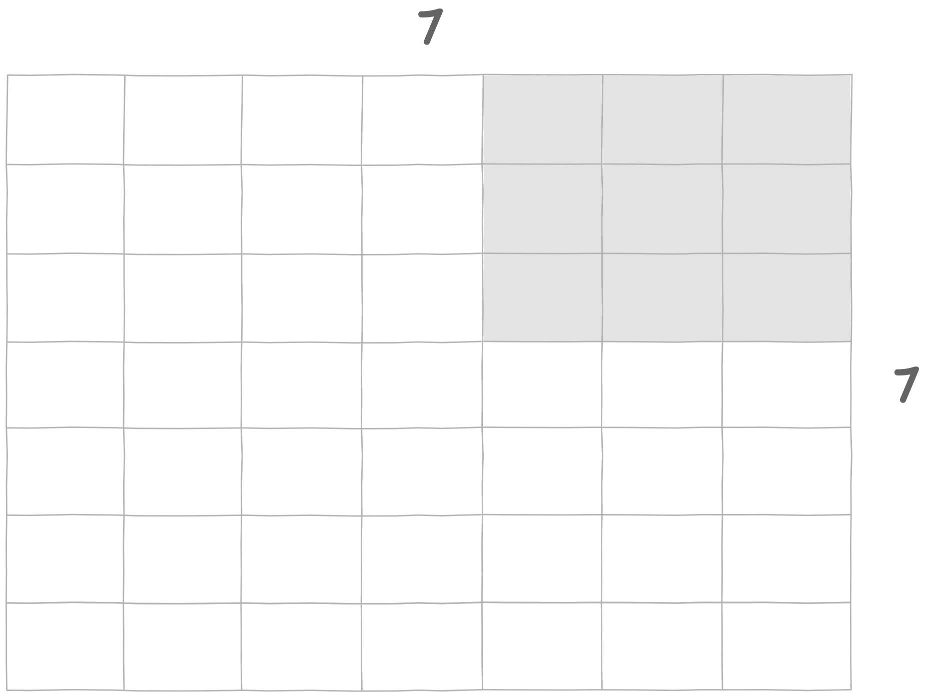
7x7 Input (Spatially)
assume 3x3 Filter



A Closer Look at Spatial Dimensions

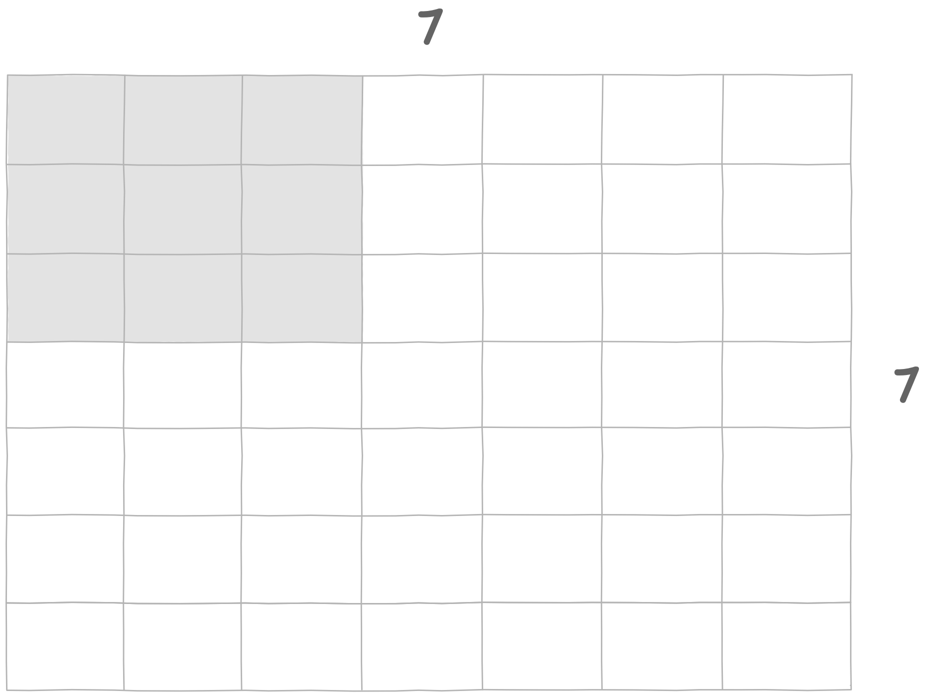
7x7 Input (Spatially)
assume 3x3 Filter

= 5x5 Output



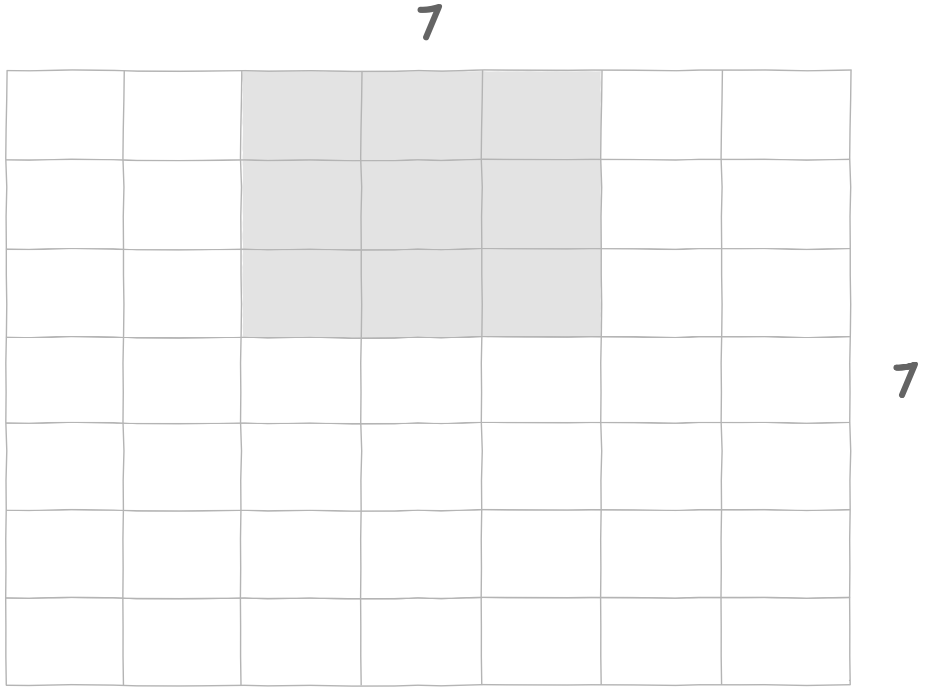
A Closer Look at Spatial Dimensions

7x7 Input (Spatially)
assume 3x3 filter
applied with Stride 2



A Closer Look at Spatial Dimensions

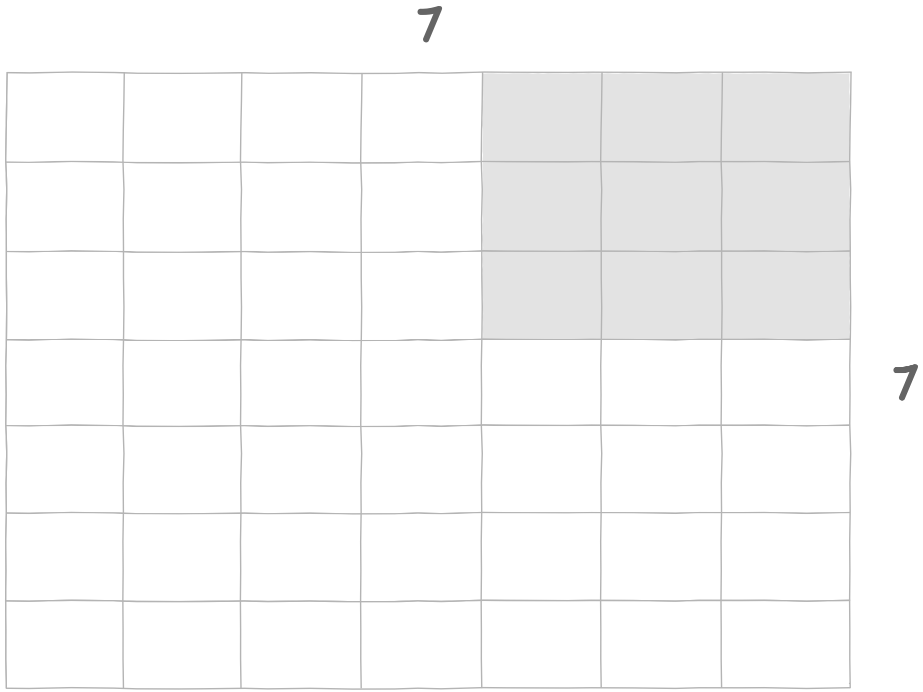
7x7 Input (Spatially)
assume 3x3 filter
applied with Stride 2



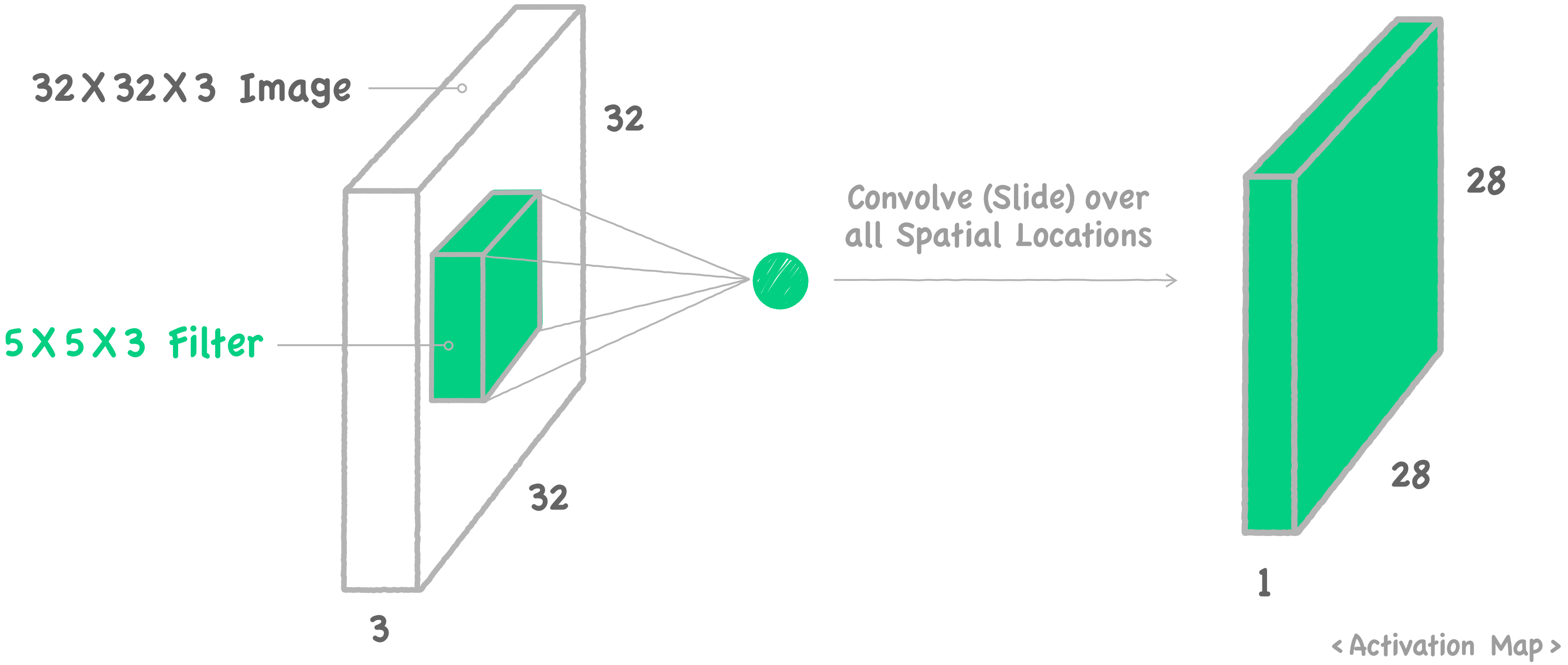
A Closer Look at Spatial Dimensions

7x7 Input (Spatially)
assume 3x3 filter
applied with Stride 2

= 3x3 Output

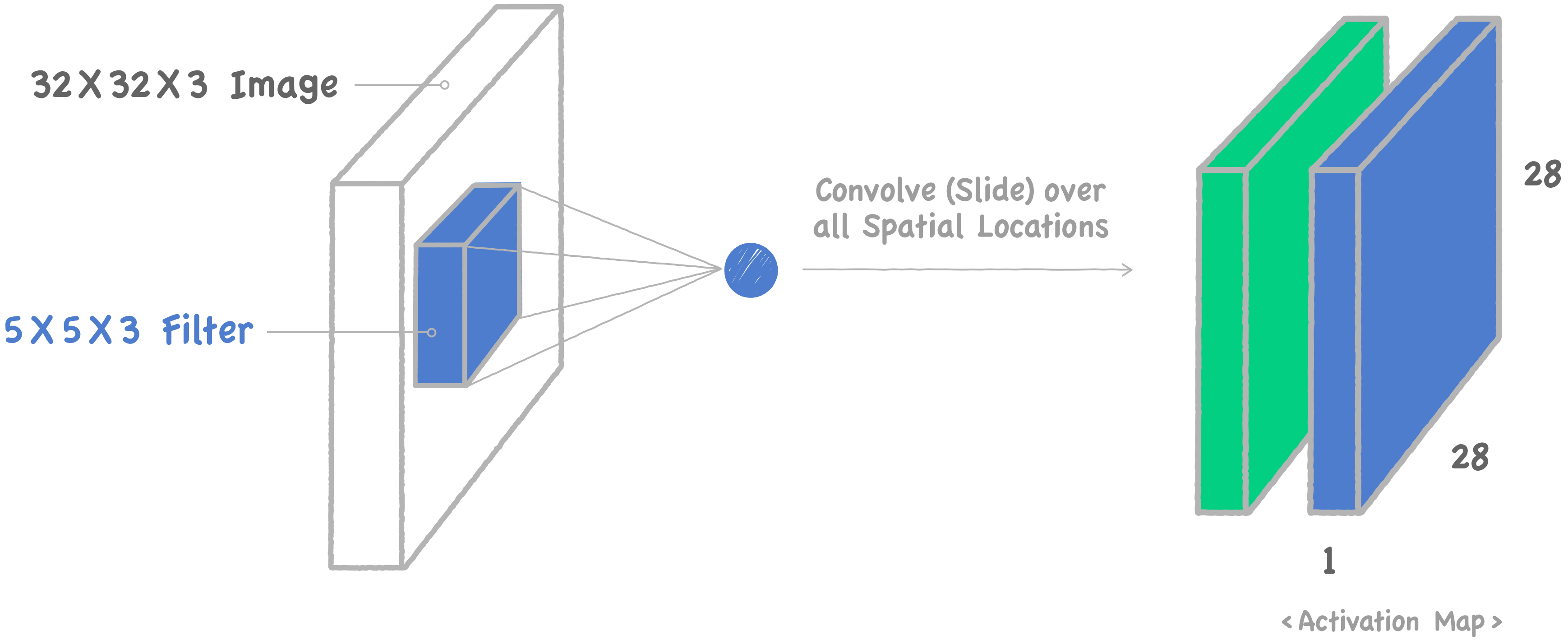


Convolution Layer



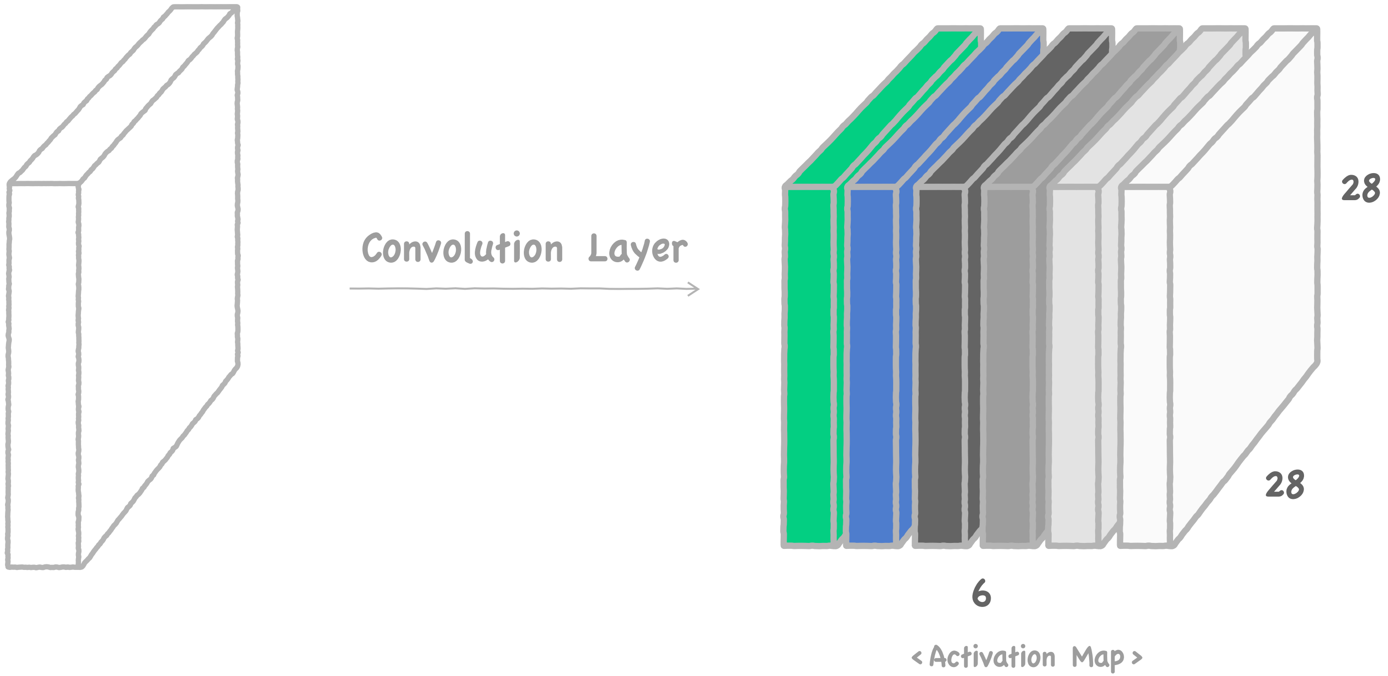
Convolution Layer

Consider a Second, Blue Filter



Convolution Layer

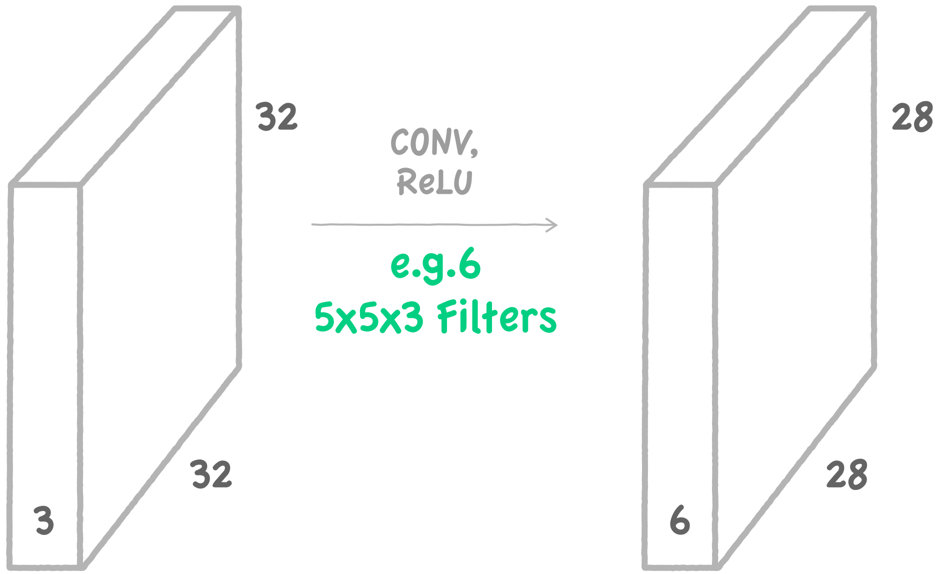
For example, if we had 6 5X5 filters,
we'll get 6 seperate activation maps:



We stack these up to get a "New Image" of size 28X28X6!

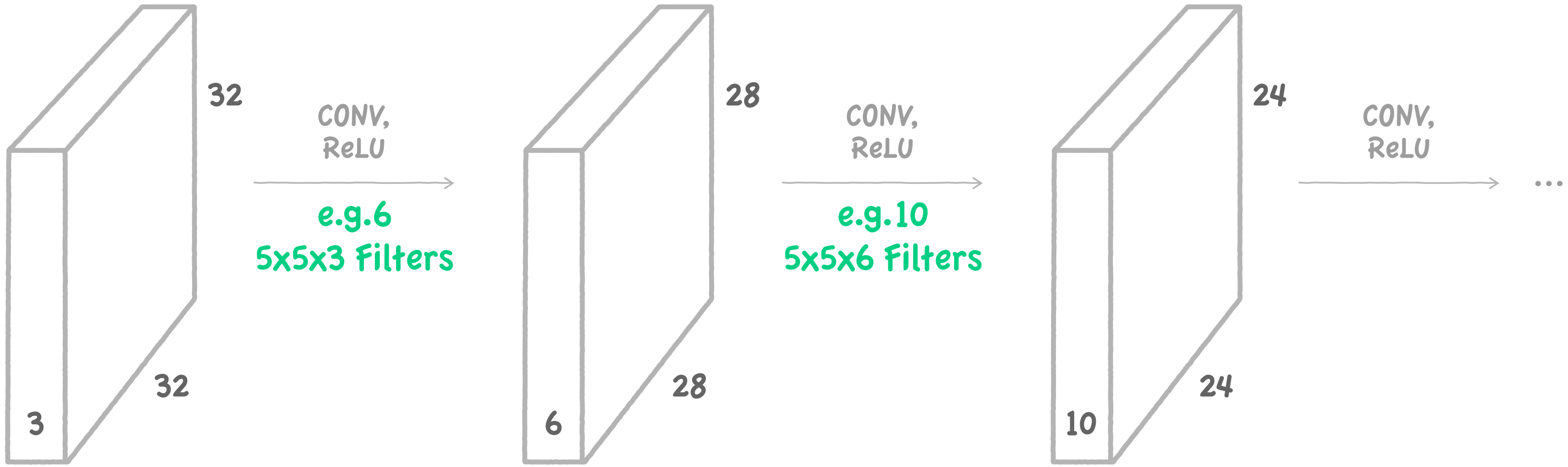
Convolution Layer

Preview:
ConvNet is a sequence of convolutional layers, interspersed with activation functions



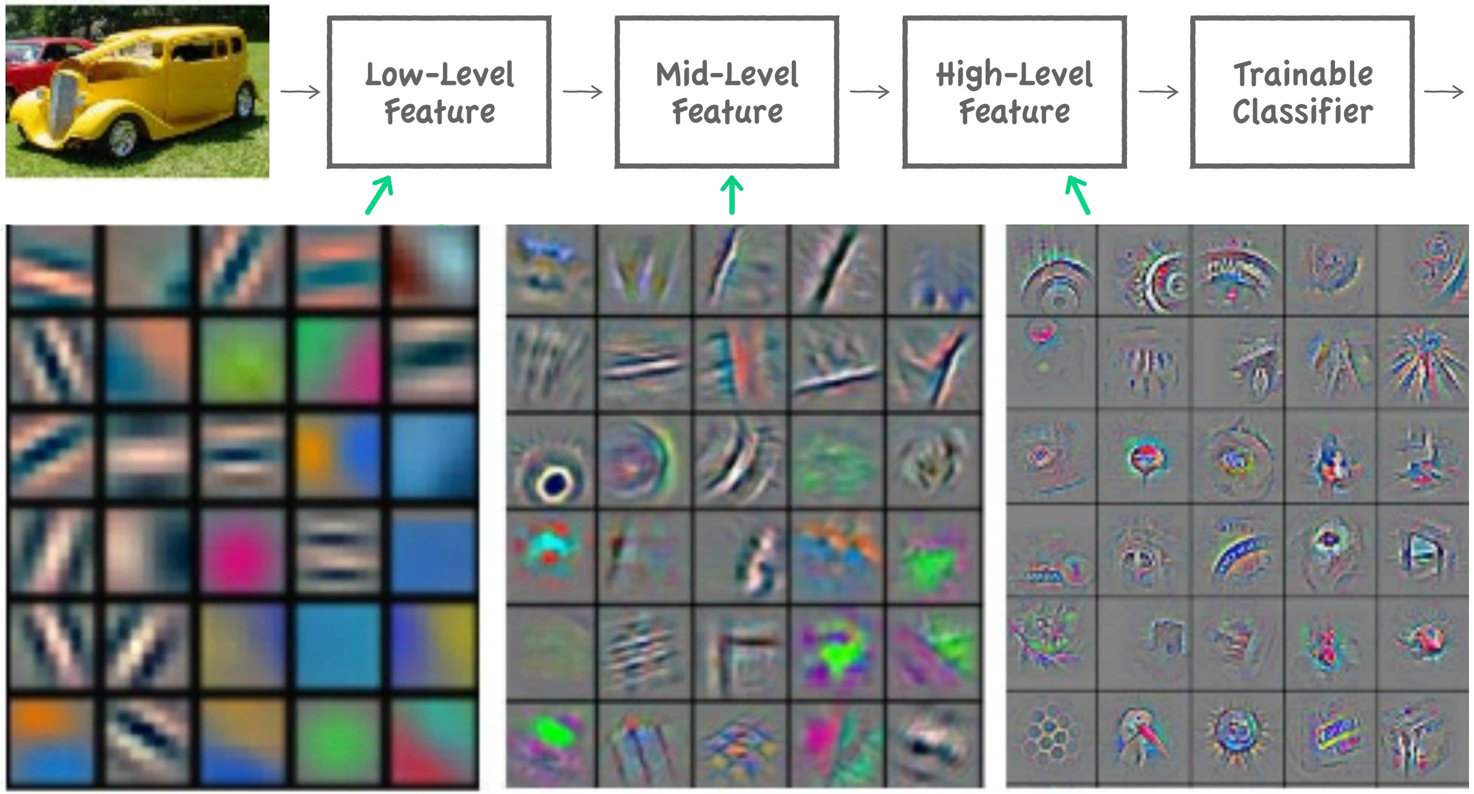
Convolution Layer

Preview:
ConvNet is a sequence of convolutional layers, interspersed with activation functions



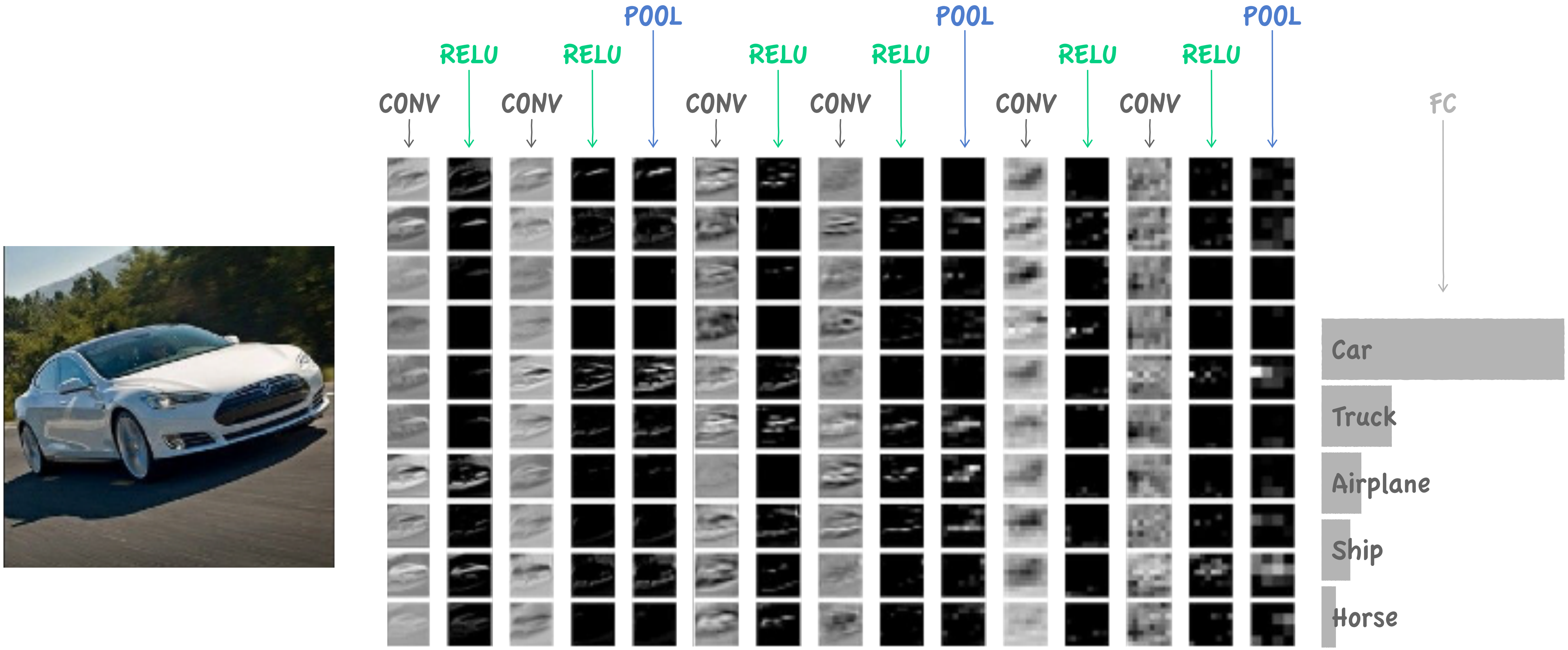
Preview

From recent Yann LeCun slides



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

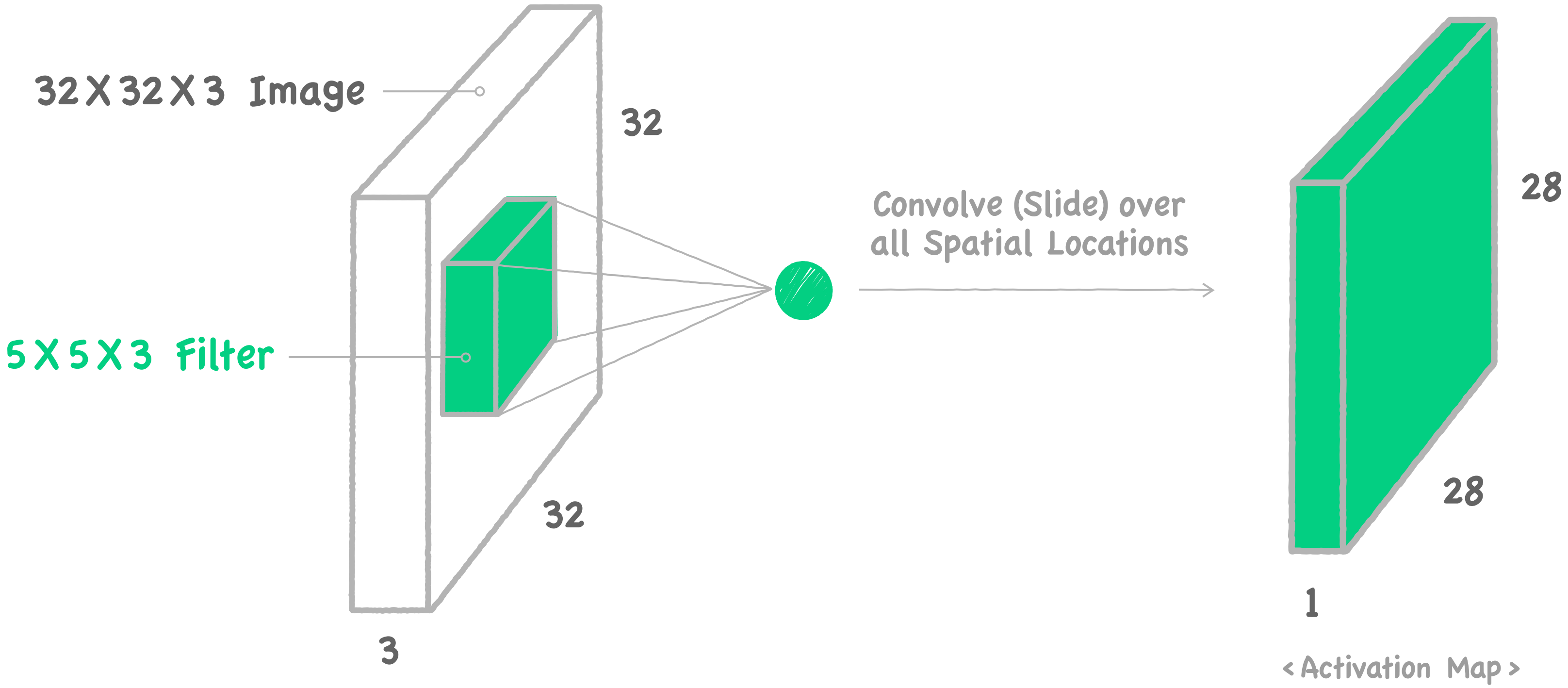
Preview



POOL
RELU
RELU
POOL
RELU
RELU
POOL
RELU
RELU
POOL
CONV
CONV
CONV
CONV
CONV
CONV
FC
Car
Truck
Airplane
Ship
Horse

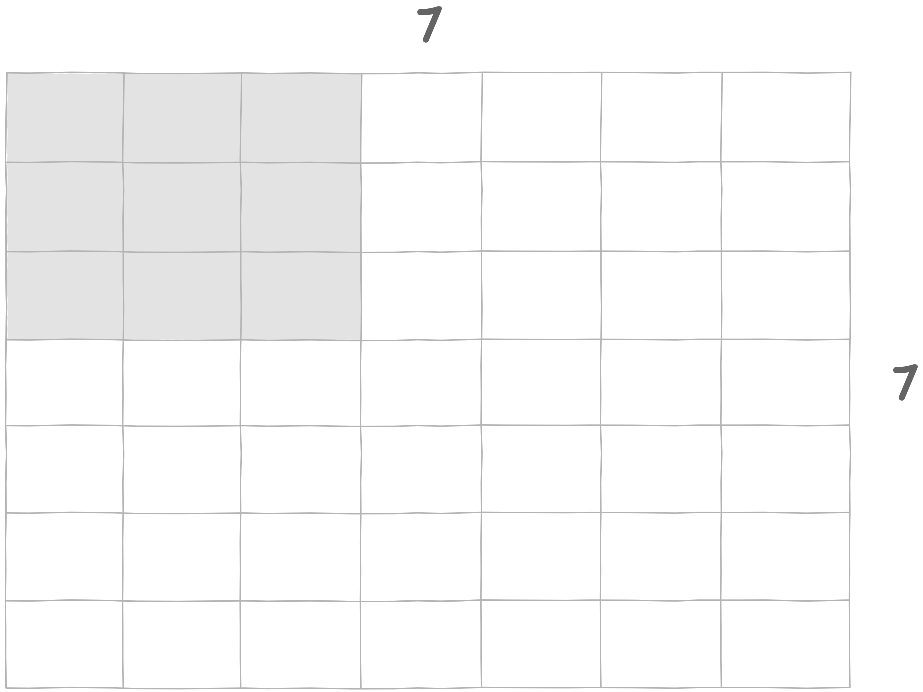
Convolution Layer

A closer look at spatial dimensions



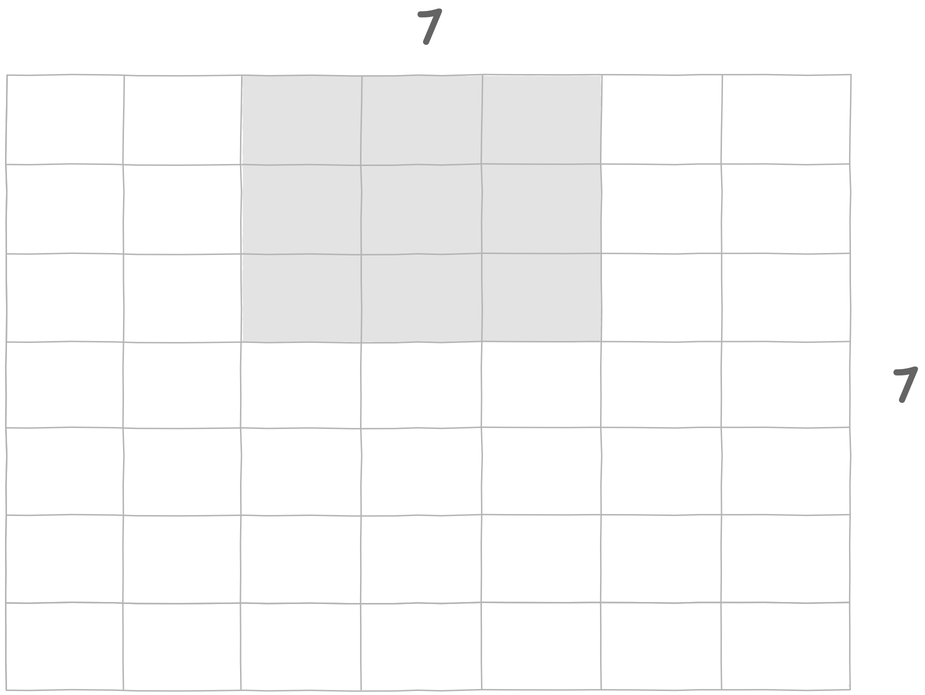
A Closer Look at Spatial Dimensions

7x7 Input (Spatially)
assume 3x3 filter
applied with Stride 2



A Closer Look at Spatial Dimensions

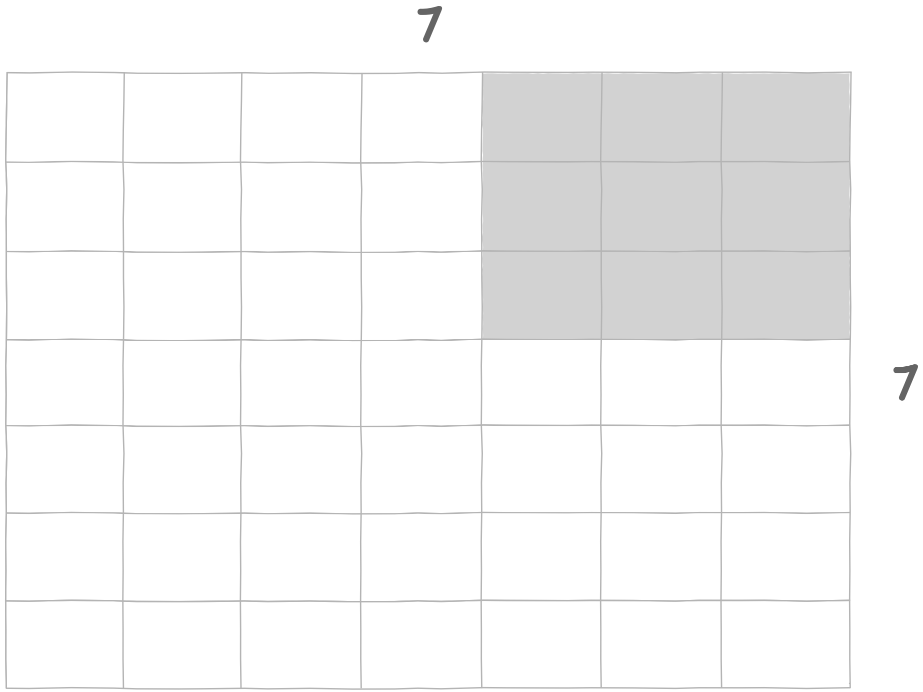
7x7 Input (Spatially)
assume 3x3 filter
applied with Stride 2



A Closer Look at Spatial Dimensions

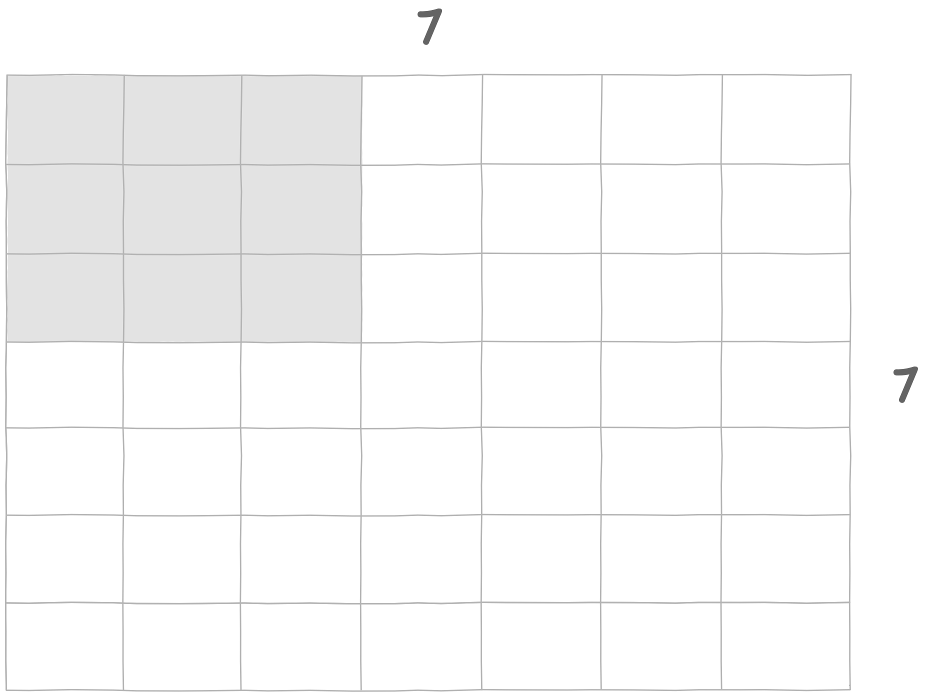
7x7 Input (Spatially)
assume 3x3 filter
applied with Stride 2

= 3x3 Output



A Closer Look at Spatial Dimensions

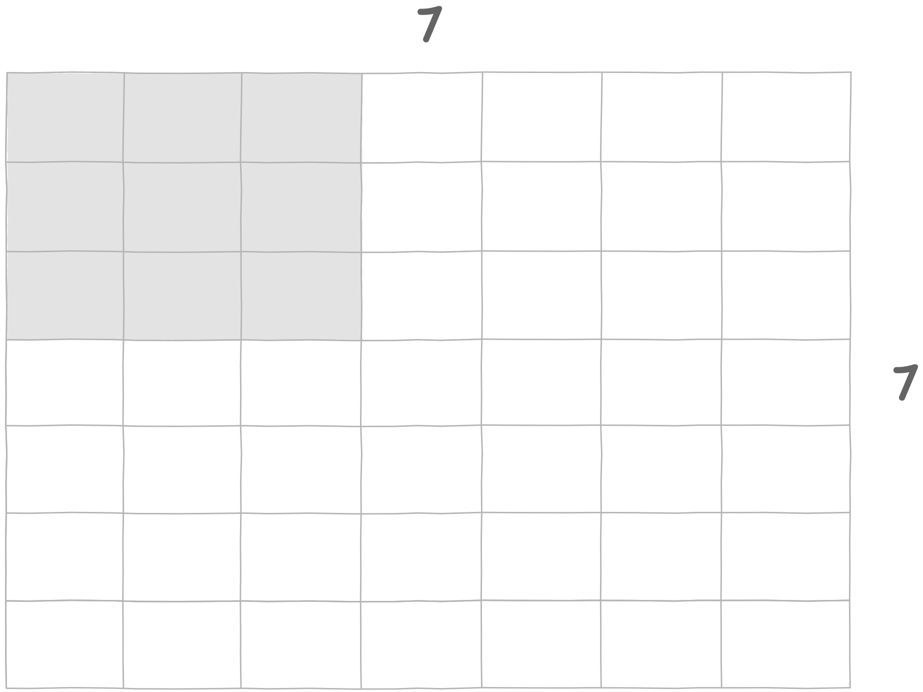
7x7 Input (Spatially)
assume 3x3 filter
applied with Stride 2



A Closer Look at Spatial Dimensions

7x7 Input (Spatially)
assume 3x3 filter
applied with Stride 3?

"Doesn't fit!"
Cannot apply 3x3 filter on
7x7 input with stride 3

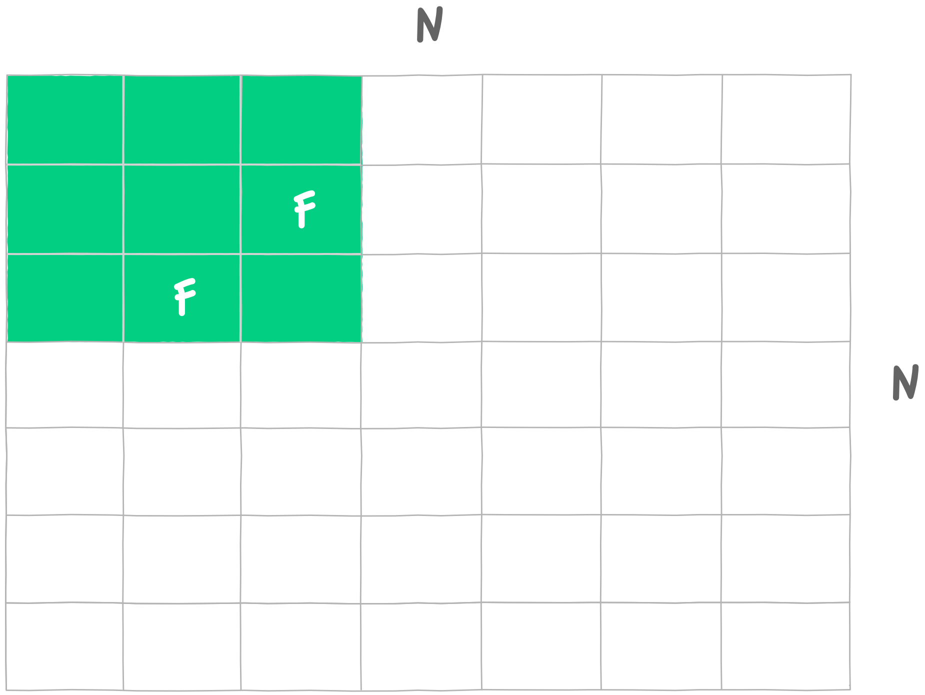


A Closer Look at Spatial Dimensions

Output Size:
(N - F) / Stride + 1

e.g. N=7, F=3

Stride 1: (7-3)/1+1 = 5
Stride 2: (7-3)/2+1 = 3
Stride 3: (7-3)/3+1 = 2.33 :\



In Practice : Common to Zero Pad the Border

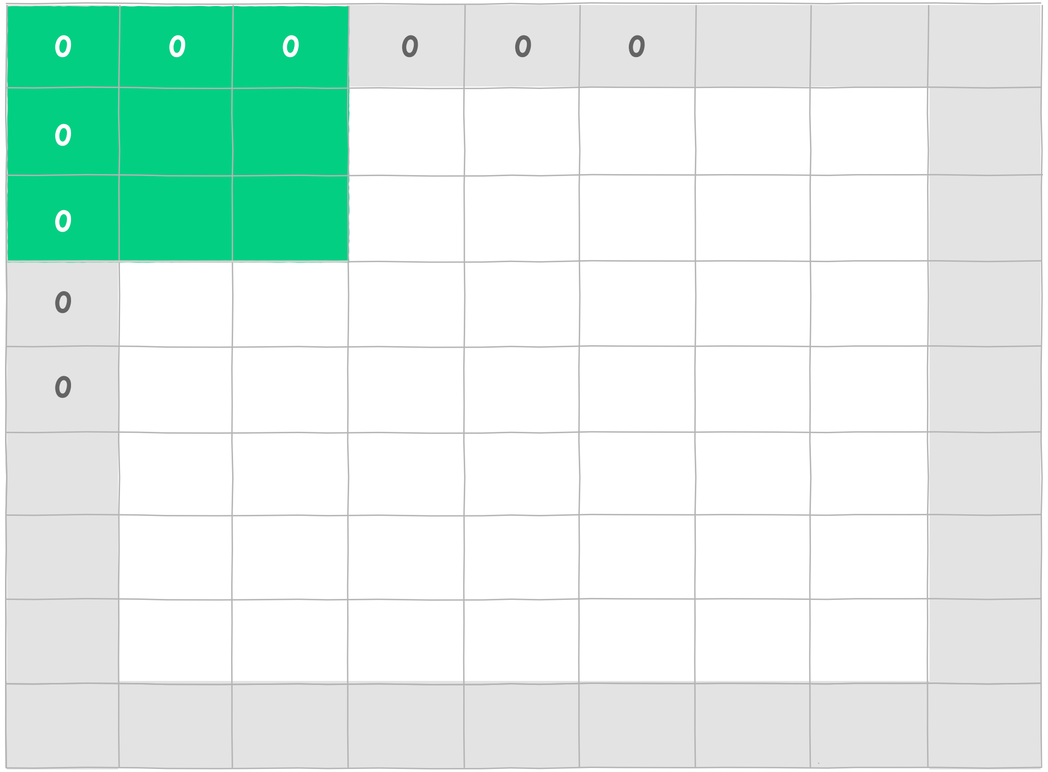
e.g. input 7X7

3X3 filter, applied with stride 1
pad with 1 pixel border

-> What is the output?

(recall :)

(N-F)/Stride+1



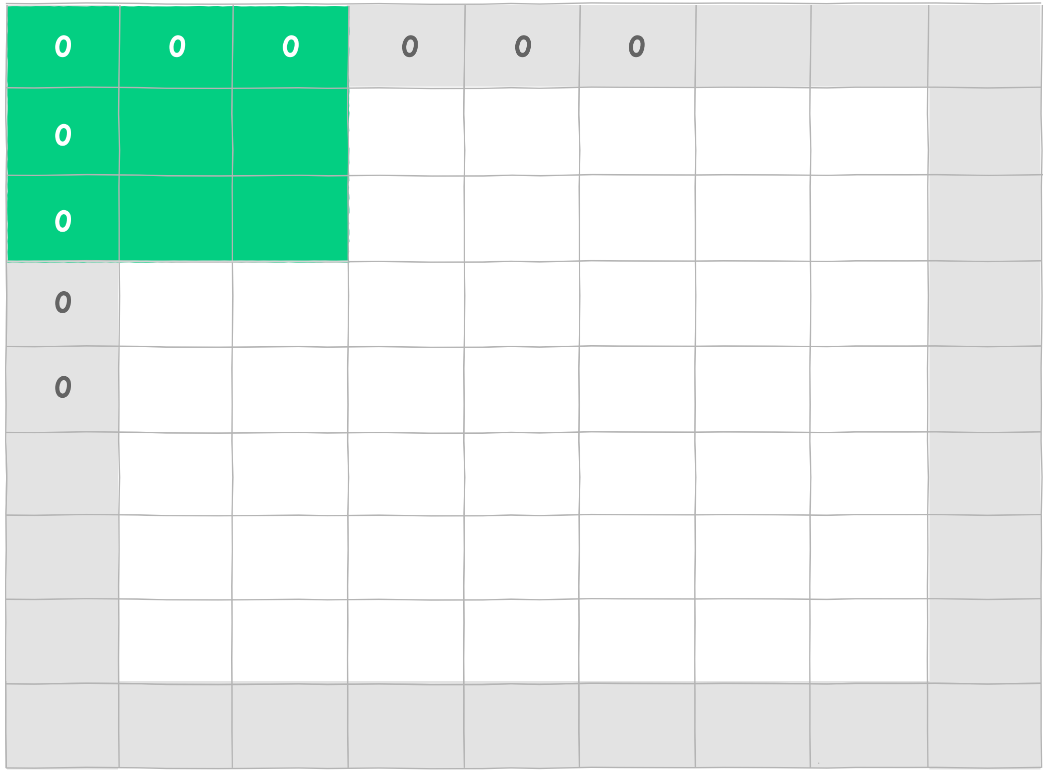
In Practice : Common to Zero Pad the Border

e.g. input 7X7

3X3 filter, applied with stride 1
pad with 1 pixel border

-> What is the output?

= 7x7 Output!



In Practice : Common to Zero Pad the Border

e.g. input 7X7

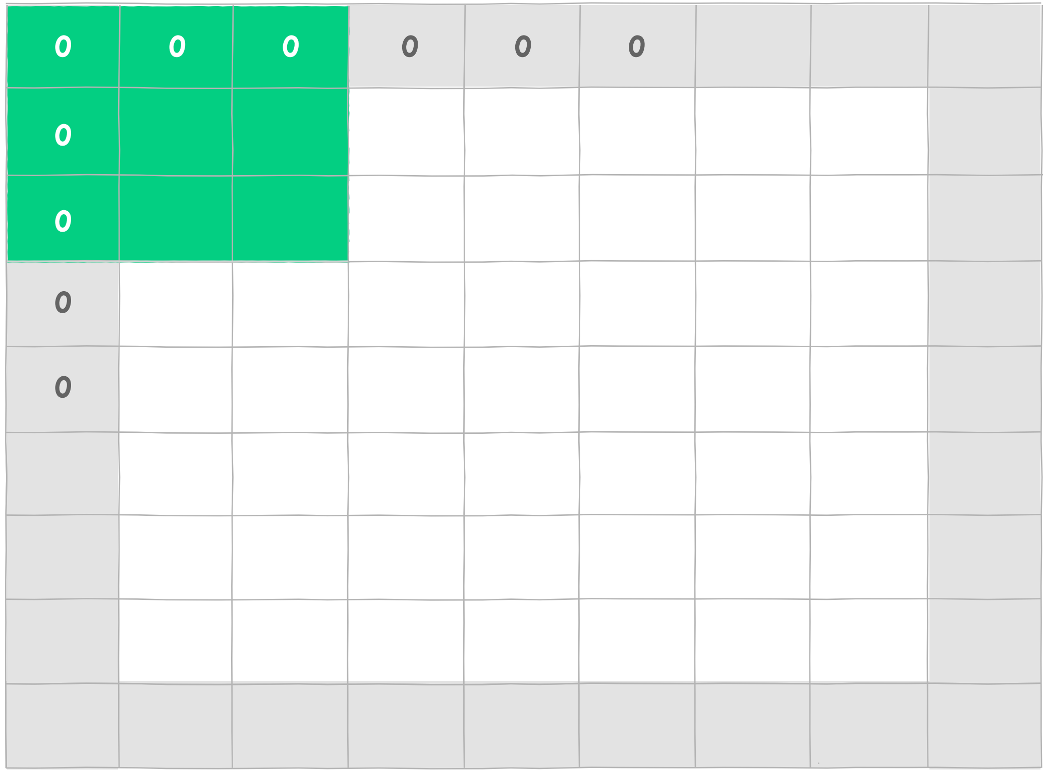
3X3 filter, applied with stride 1
pad with 1 pixel border

-> What is the output?

= 7x7 Output!

in general, common to see CONV layers
with stride 1, filters of size FXF,
and zero-padding with (F-1)/2.
(will preserve size saptially)

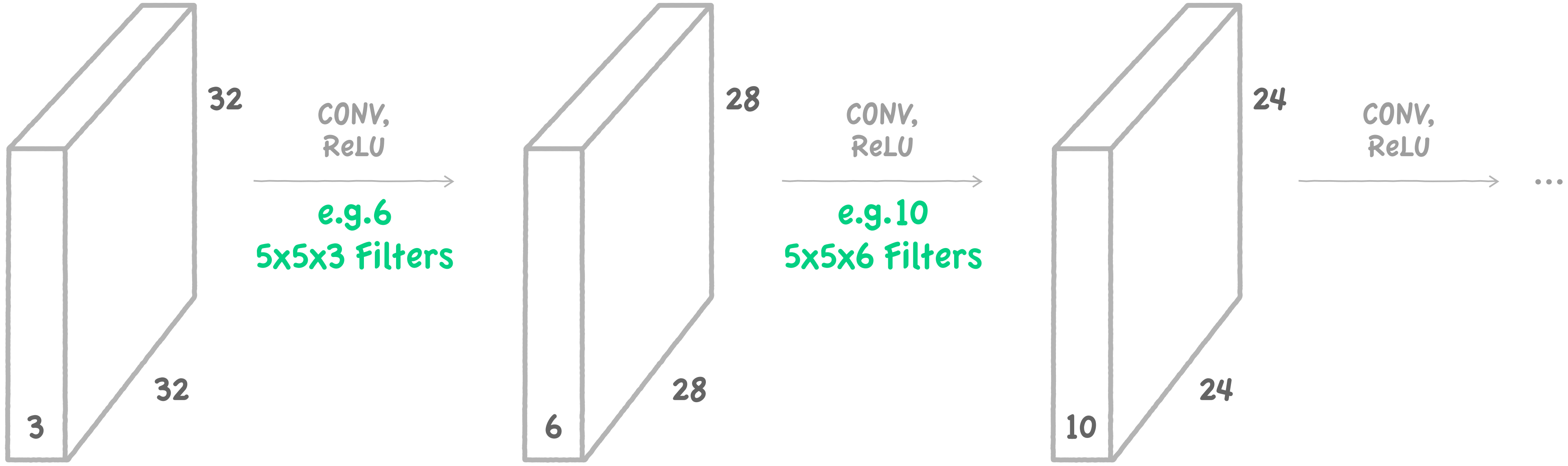
e.g F=3 -> zero pad with 1
F=5 -> zero pad with 2
F=7 -> zero pad with 3



Convolution Layer

Remember back to...

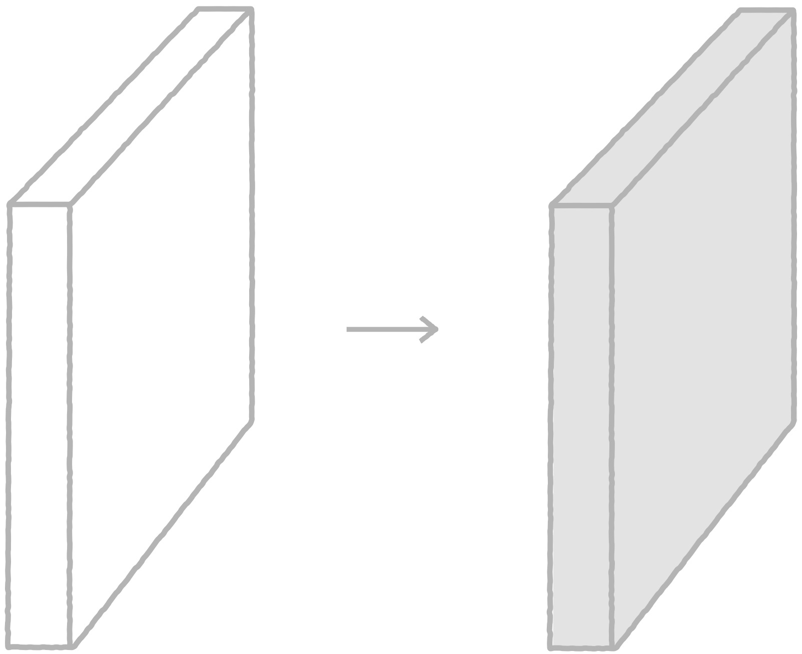
E.g.
32x32 input convolved repeatedly with 5x5 filters
shrinks volumes spatially! (32 -> 28 -> 24 ...).
Shrinking too fast is not good, doesn't work well.



Examples Time

Input Volume:
32 X 32 X 3
10 5X5 filters with stride 1, pad 2

Output Volume Size: ?



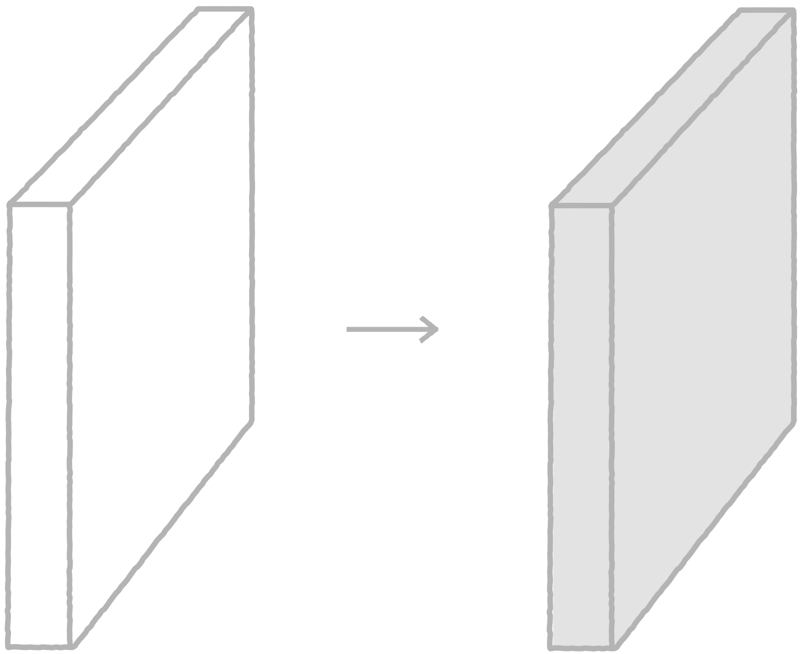
Examples Time

Input Volume:

32 X 32 X 3
10 5X5 filters with stride 1, pad 2

Output Volume Size:

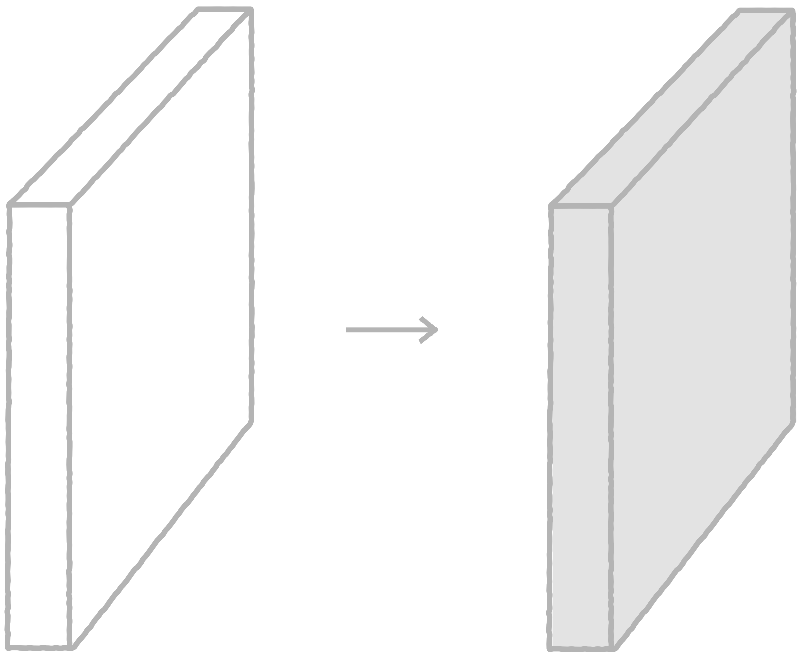
(32+2x2-5)/1+1=32 spatially, so
32X32X10



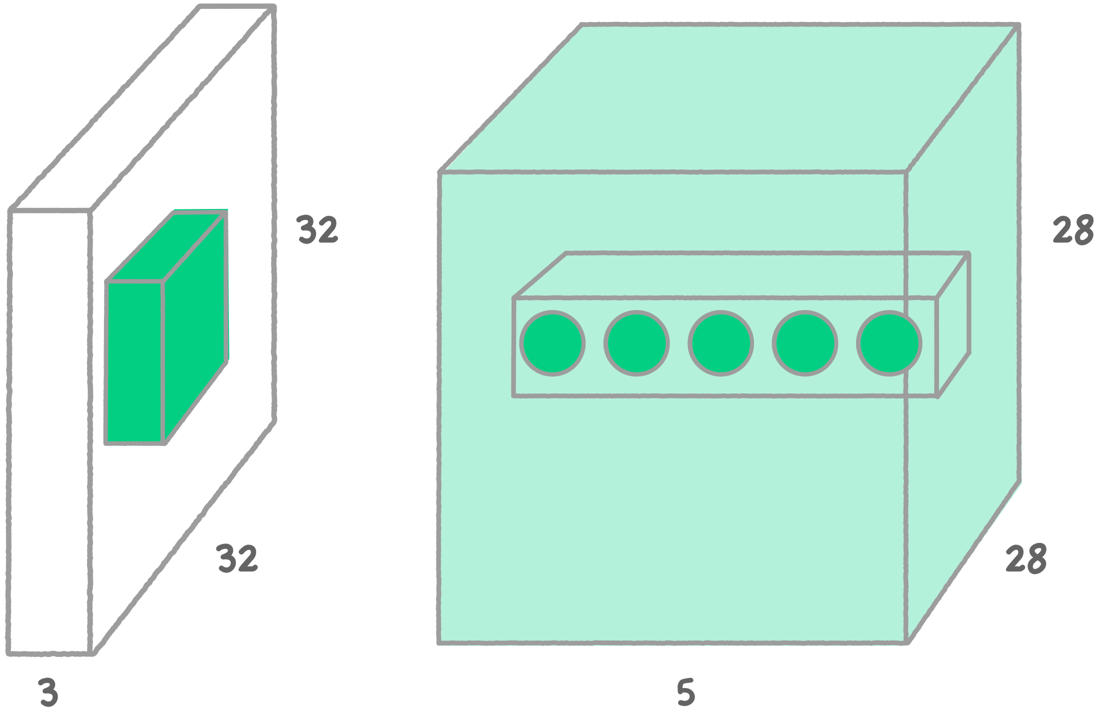
Examples Time

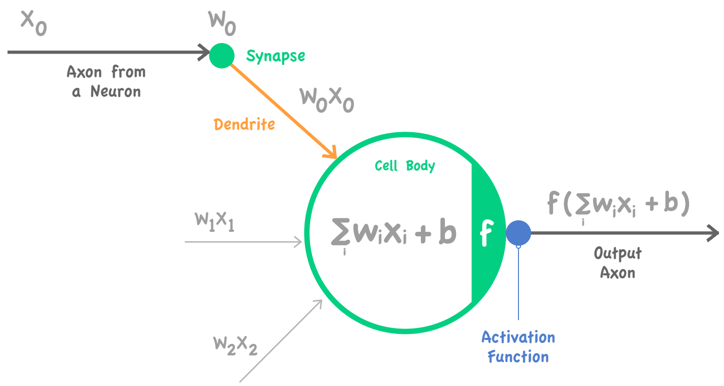
Input Volume:
32 X 32 X 3
10 5X5 filters with stride 1, pad 2

Output Volume Size: ?



The Brain/Neuron View of CONV Layer

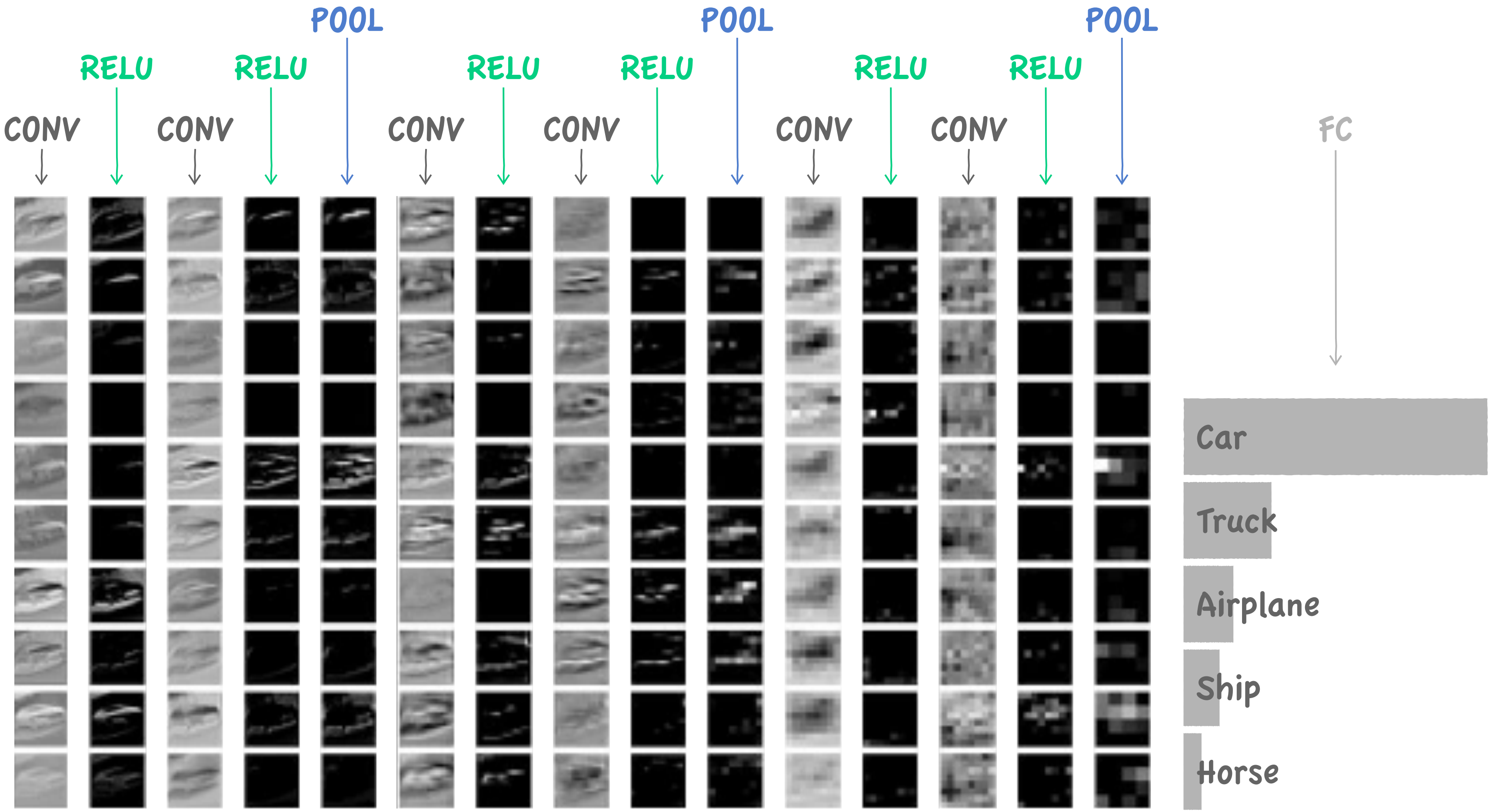




e.g. with 5 filters,
CONV layer consists of
neurons arranged in a 3D grid
(28x28x5)

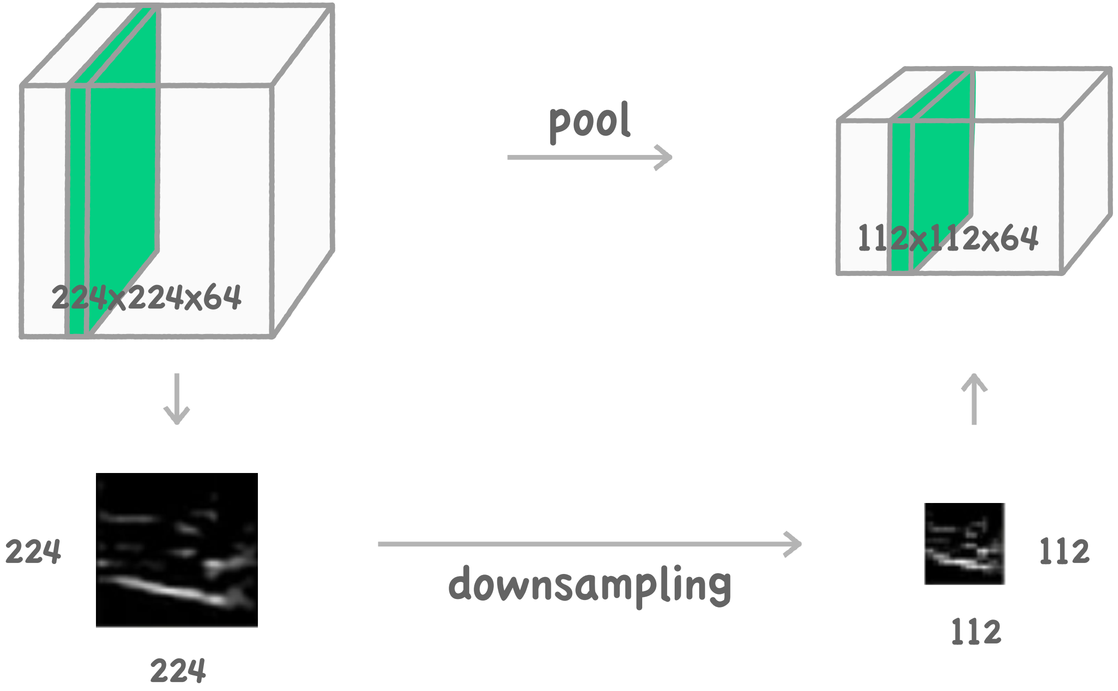
There will be 5 different neurons
all looking at the same region
in the input volume

Two More Layers to Go : POOL / FC

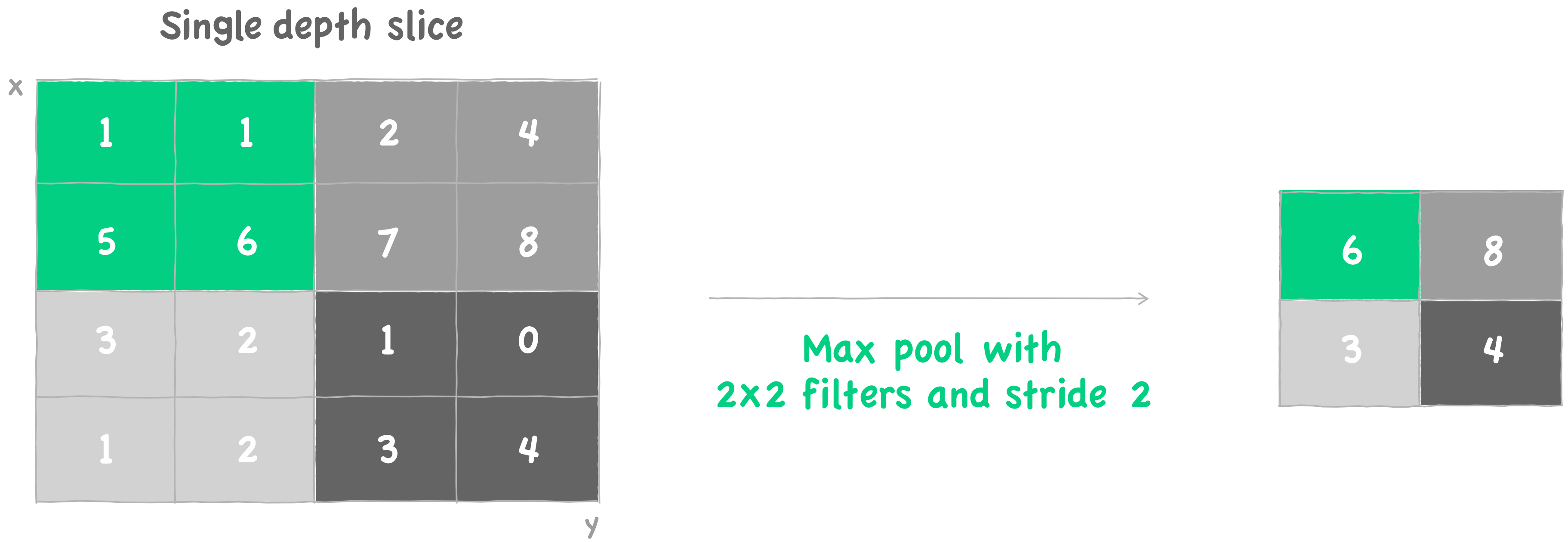


Pooling Layer

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



Max Pooling


Single depth slice
x
1 1 2 4
5 6 7 8
3 2 1 0
1 2 3 4
y
Max pool with
2x2 filters and stride 2
6 8
3 4

Max Pooling

01. Accepts a volume of size $W_1 \times H_1 \times D_1$
02. Requires three hyperparameters:
    - their spatial extent F
    - the stride S
03. Produces a volume of size $W_2 \times H_2 \times D_2$ where:
    - $W_2 = (W_1 - F) / S + 1$
    - $H_2 = (H_1 - F) / S + 1$
    - $D_2 = D_1$
04. Introduces zero parameters since it computes a fixed function of the input
05. Note that it is not common to use zero-padding for pooling layers

Max Pooling

01. Accepts a volume of size $W_1 \times H_1 \times D_1$
02. Requires three hyperparameters:
    - their spatial extent F
    - the stride S
03. Produces a volume of size $W_2 \times H_2 \times D_2$ where:
    - $W_2 = (W_1 - F)/S + 1$
    - $H_2 = (H_1 - F)/S + 1$
    - $D_2 = D_1$
04. Introduces zero parameters since it computes a fixed function of the input
05. Note that it is not common to use zero-padding for pooling layers

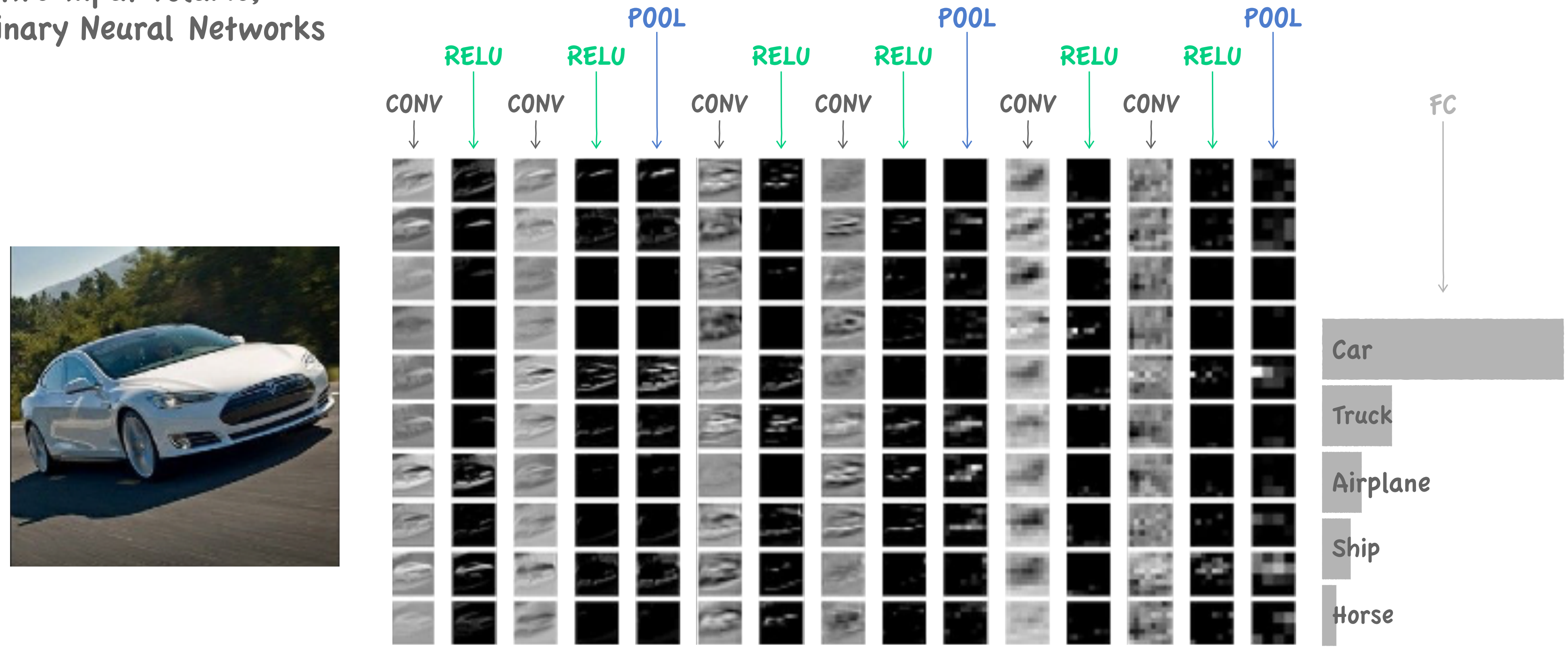
Common Settings

$F=2, \ S=2$

$F=3, \ S=2$

Fully Connected Layer (FC Layer)

Contains neurons that connect to the entire input volume, as in ordinary Neural Networks



ConvNetJs Demo : Training on CIFAR-10

http://cs.stanford.edu/people/karpathy/convnetjs/demo/cifar10.html

NEXT LECTURE

CNN
CASE STUDY