Foundations of Convolutional Neural Networks

Pio

Computer Vision Problems

Image Classification

64x64

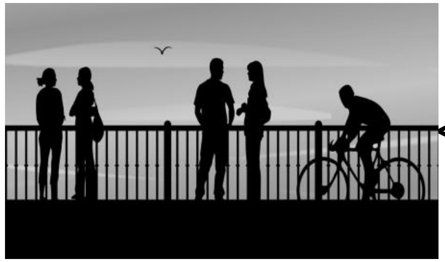
Cat? (0/1)

Object detection

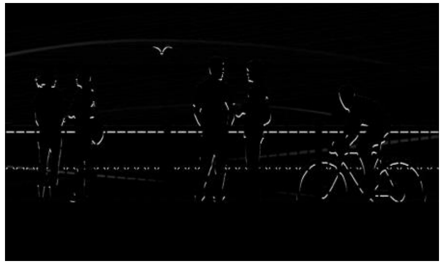
Neural Style Transfer

Andrew Ng

Computer Vision Problem



vertical edges



horizontal edges

Andrew Ng

Vertical edge detection

3	0	1	2	7	4
1	5	8	9	3	1
2	7	2	5	1	3
0	1	3	1	7	8
4	2	1	6	2	8
2	4	5	2	3	9

6 x 6

convolution

*

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

3 x 3

filter / kernel

=

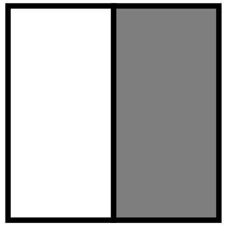
0	-2	-4	-7
-3	-2	-3	-16

4 x 4

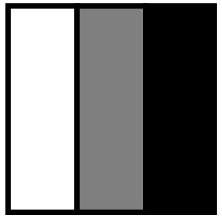
Andrew Ng

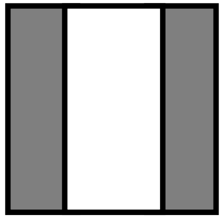
Vertical edge detection

$$\begin{bmatrix} 10 & 10 & 10 & 0 & 0 & 0 \\ 10 & 10 & 10 & 0 & 0 & 0 \\ 10 & 10 & 10 & 0 & 0 & 0 \\ 10 & 10 & 10 & 0 & 0 & 0 \\ 10 & 10 & 10 & 0 & 0 & 0 \\ 10 & 10 & 10 & 0 & 0 & 0 \end{bmatrix} * \begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix} = \begin{bmatrix} 0 & 30 & 30 & 0 \\ 0 & 30 & 30 & 0 \\ 0 & 30 & 30 & 0 \\ 0 & 30 & 30 & 0 \end{bmatrix}$$



$*$

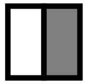
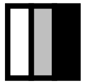




Andrew Ng

Vertical edge detection examples

$$\begin{bmatrix} 10 & 10 & 10 & 0 & 0 & 0 \\ 10 & 10 & 10 & 0 & 0 & 0 \\ 10 & 10 & 10 & 0 & 0 & 0 \\ 10 & 10 & 10 & 0 & 0 & 0 \\ 10 & 10 & 10 & 0 & 0 & 0 \\ 10 & 10 & 10 & 0 & 0 & 0 \end{bmatrix} * \begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix} = \begin{bmatrix} 0 & 30 & 30 & 0 \\ 0 & 30 & 30 & 0 \\ 0 & 30 & 30 & 0 \\ 0 & 30 & 30 & 0 \end{bmatrix}$$


$$\begin{bmatrix} 0 & 0 & 0 & 10 & 10 & 10 \\ 0 & 0 & 0 & 10 & 10 & 10 \\ 0 & 0 & 0 & 10 & 10 & 10 \\ 0 & 0 & 0 & 10 & 10 & 10 \\ 0 & 0 & 0 & 10 & 10 & 10 \\ 0 & 0 & 0 & 10 & 10 & 10 \end{bmatrix} * \begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix} = \begin{bmatrix} 0 & -30 & -30 & 0 \\ 0 & -30 & -30 & 0 \\ 0 & -30 & -30 & 0 \\ 0 & -30 & -30 & 0 \end{bmatrix}$$



Andrew Ng

Vertical and Horizontal Edge Detection

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

Vertical

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

Horizontal

10	10	10	0	0	0
10	10	10	0	0	0
10	10	10	0	0	0
0	0	0	10	10	10
0	0	0	10	10	10
0	0	0	10	10	10

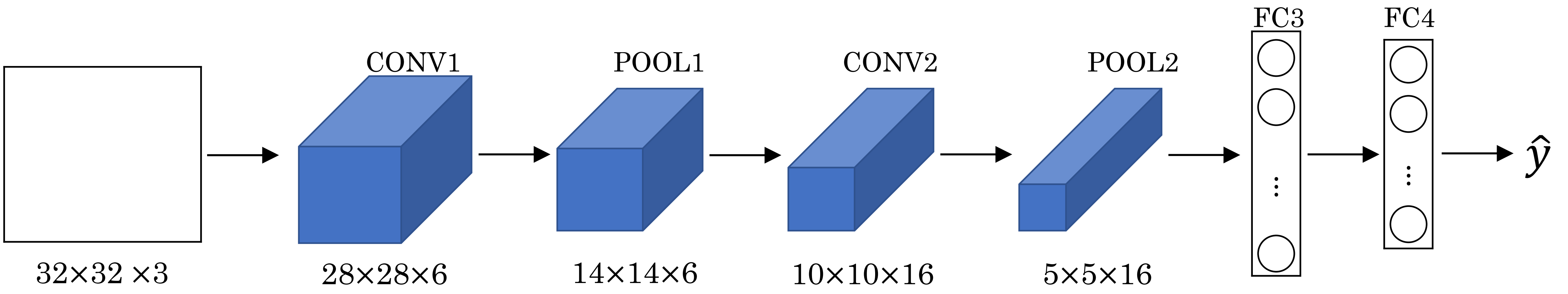
*

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

=

0	0	0	0
30	10	-10	-30
30	10	-10	-30
0	0	0	0

Types of layer in a convolutional network:

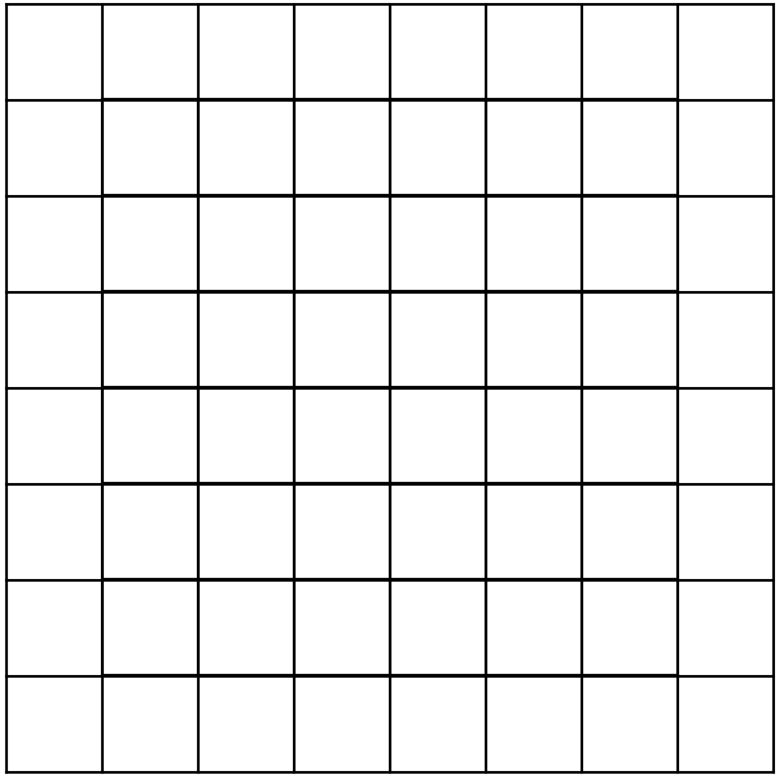


- Convolution (CONV)
- Pooling (POOL)
- Fully connected (FC)

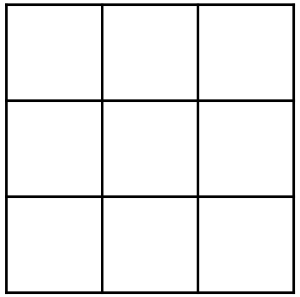
Convolution Parameters

- Padding
- Stride
- Multiple Filters

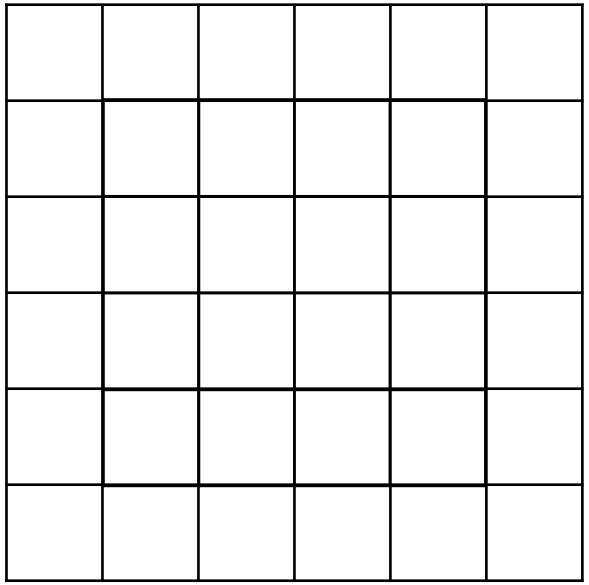
Padding



$*$



$=$



6 x 6 → 8 x 8

Padding p = 1

3 x 3

4 x 4 → 6 x 6

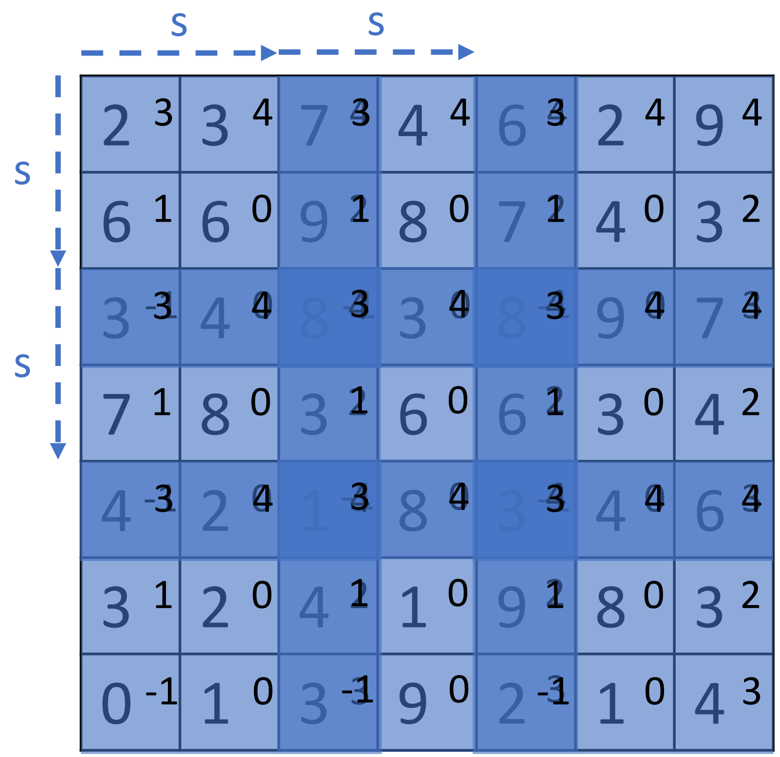
$(n - f + 1)$
$\times (n - f + 1)$

Valid
Convolution

$(n + 2p - f + 1)$
$\times (n + 2p - f + 1)$

Same
Convolution

Strided convolution



$n \times n$ image
padding p
stride $s = 2$

$*$

3	4	4
1	0	2
-1	0	3

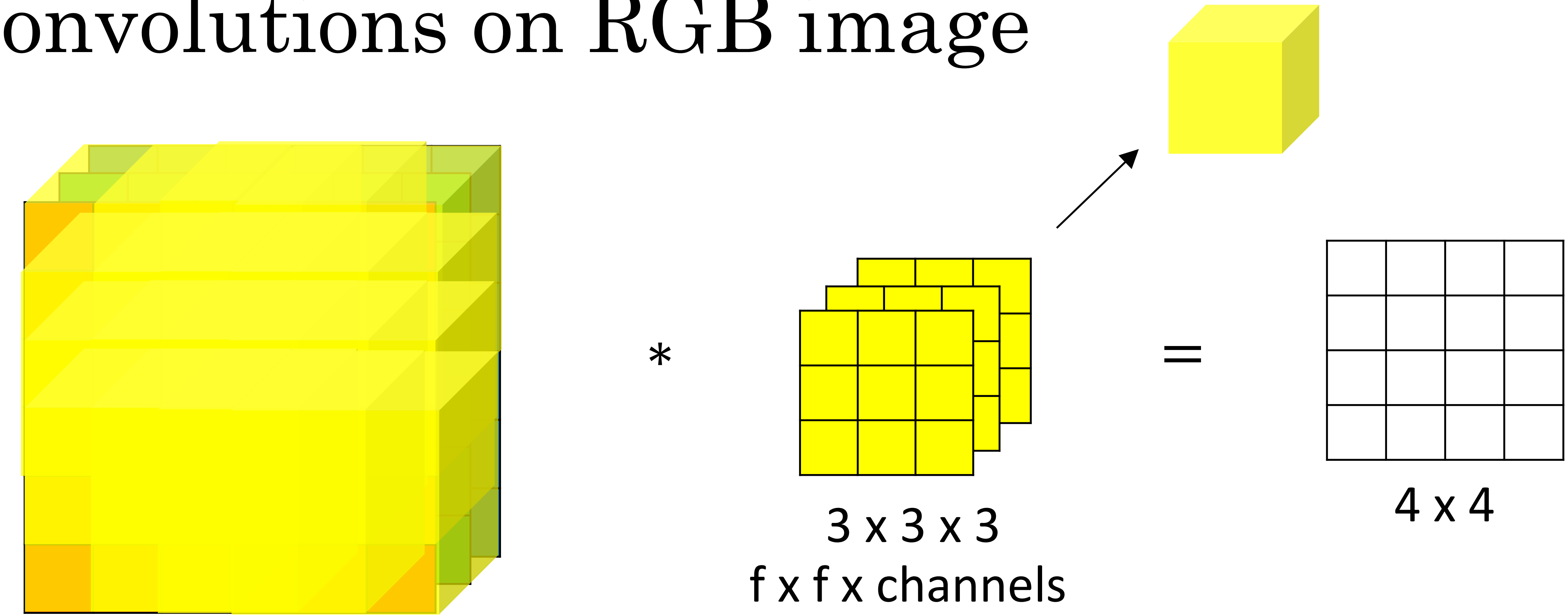
$f \times f$ filter

$=$

$$\left\lfloor \frac{n + 2p - f}{s} + 1 \right\rfloor \times \left\lfloor \frac{n + 2p - f}{s} + 1 \right\rfloor$$

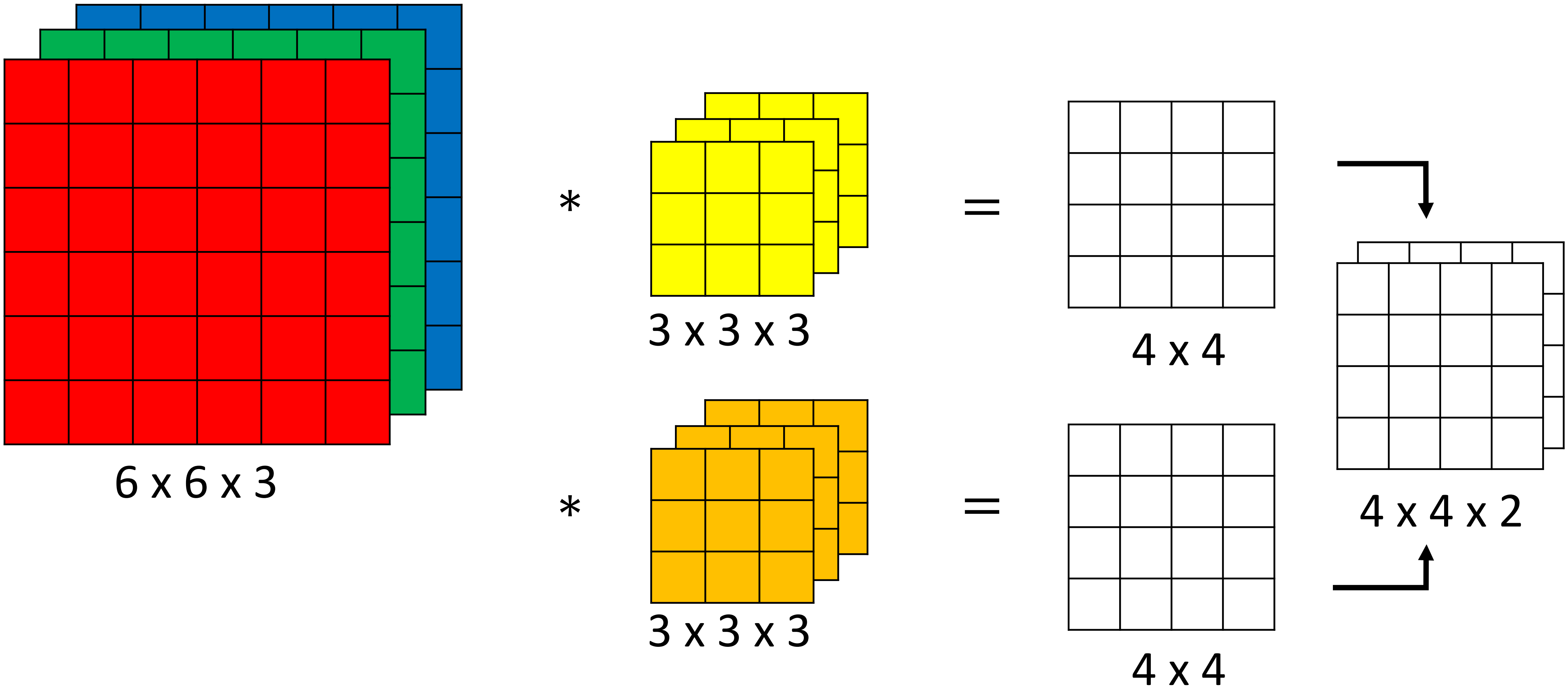
Andrew Ng

Convolutions on RGB image



Andrew Ng

Multiple filters



Andrew Ng

Summary of Convolutions

Hyperparameters:

n_H : image height
n_W : image width
n_C : # of image channels (# of filters)
f : filter size
s : stride
p : padding

$$\left\lfloor \frac{n_H + 2p - f}{s} + 1 \right\rfloor \times \left\lfloor \frac{n_W + 2p - f}{s} + 1 \right\rfloor \times n_C$$

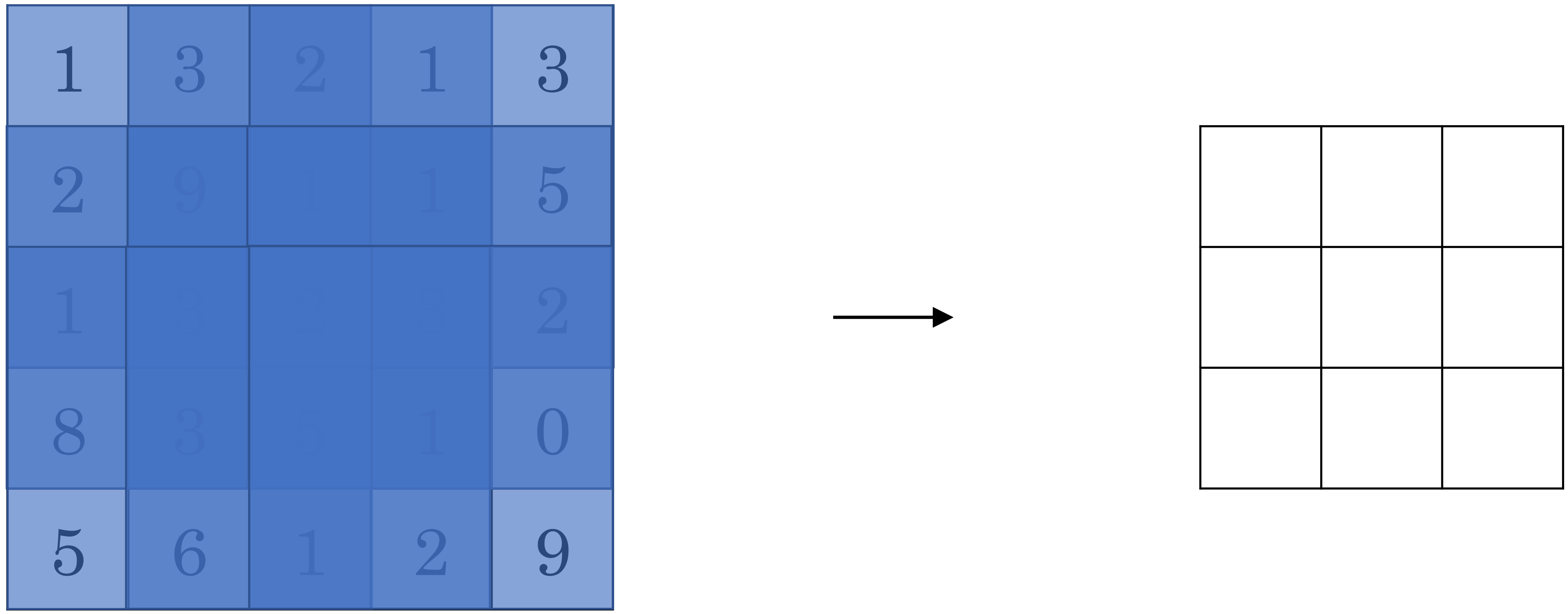
Andrew Ng

Pooling layer: Max pooling

$$\begin{bmatrix} 1 & 3 & 2 & 1 \\ 2 & 9 & 1 & 1 \\ 1 & 3 & 2 & 3 \\ 5 & 6 & 1 & 2 \end{bmatrix} \longrightarrow \begin{bmatrix} 9 & 2 \\ 6 & 3 \end{bmatrix}$$

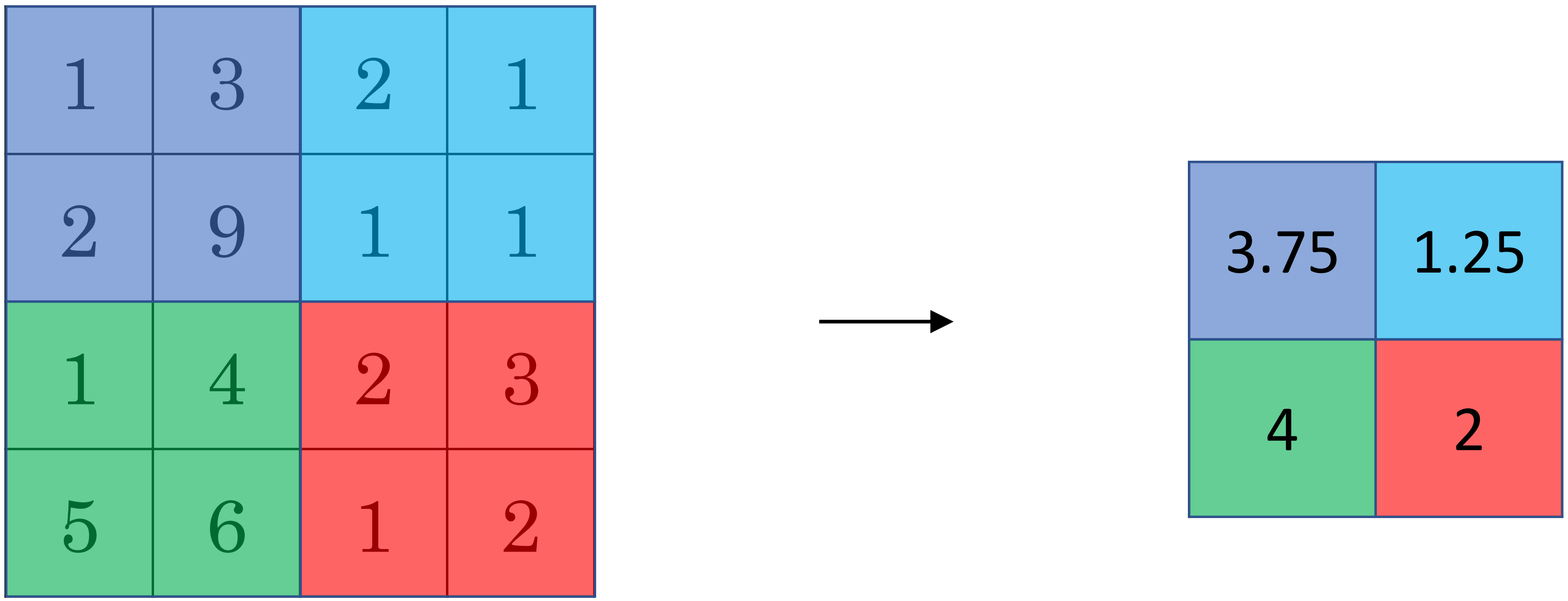
Andrew Ng

Pooling layer: Max pooling



Andrew Ng

Pooling layer: Average pooling



Summary of pooling

Hyperparameters:

f : filter size

s : stride

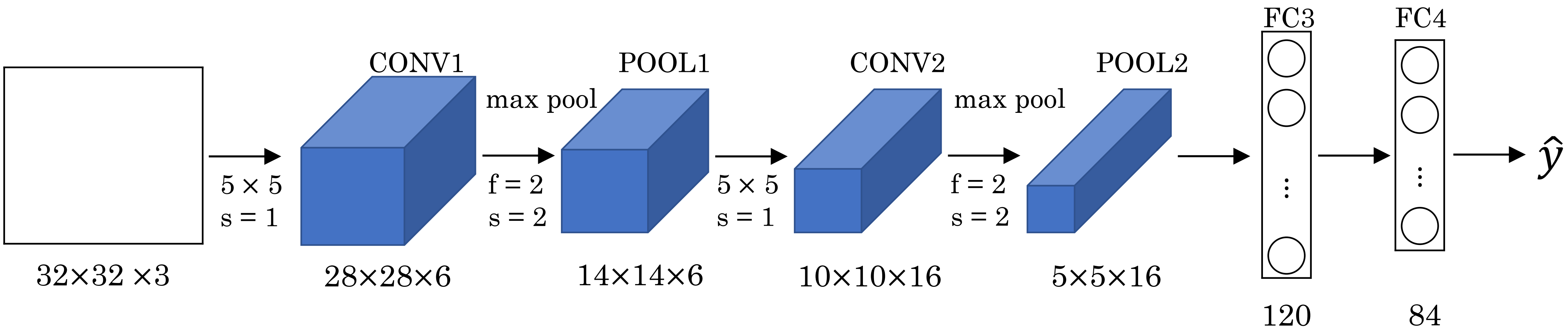
Max or average pooling

~~p : padding~~

$$\left\lfloor \frac{n_H - f}{s} + 1 \right\rfloor \times \left\lfloor \frac{n_W - f}{s} + 1 \right\rfloor \times n_C$$

Andrew Ng

Neural network example (LeNet – 5)

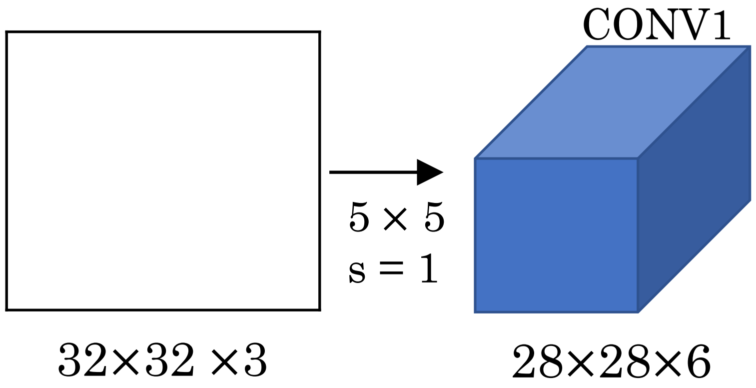


	Activation shape	Activation Size	# parameters
Input:	(32,32,3)	3,072	0
CONV1 (f=5, s=1)	(28,28,6)	4704	156
POOL1	(14,14,6)	1176	0
CONV2 (f=5, s=1)	(10,10,16)	1,600	416
POOL2	(5,5,16)	400	0
FC3	(120,1)	120	48,001
FC4	(84,1)	84	10,081
Softmax	(10,1)	10	841

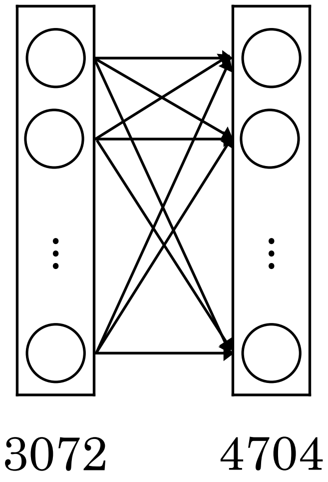
$n_H, n_W \downarrow$
$n_C \uparrow$

Andrew Ng

Why convolutions



	Activation shape	Activation Size	# parameters
Input:	(32,32,3)	3,072	0
CONV1 (f=5, s=1)	(28,28,6)	4,704	156



3072 x 4704 ≈ 14M

Andrew Ng

Why convolutions

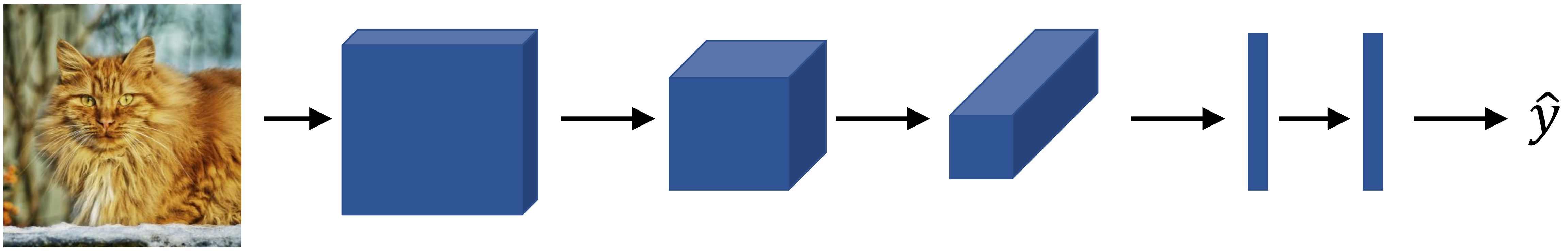
$$
\begin{bmatrix} 10 & 10 & 10 & 0 & 0 & 0 \\ 10 & 10 & 10 & 0 & 0 & 0 \\ 10 & 10 & 10 & 0 & 0 & 0 \\ 10 & 10 & 10 & 0 & 0 & 0 \\ 10 & 10 & 10 & 0 & 0 & 0 \\ 10 & 10 & 10 & 0 & 0 & 0 \end{bmatrix} * \begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix} = \begin{bmatrix} 0 & 30 & 30 & 0 \\ 0 & 30 & 30 & 0 \\ 0 & 30 & 30 & 0 \\ 0 & 30 & 30 & 0 \end{bmatrix}
$$

Parameter sharing: A feature detector (such as a vertical edge detector) that's useful in one part of the image is probably useful in another part of the image.

Sparsity of connections: In each layer, each output value depends only on a small number of inputs.

Putting it together

Training set $(x^{(1)}, y^{(1)}) \ldots (x^{(m)}, y^{(m)})$.



Cost $J = \frac{1}{m} \sum_{i=1}^{m} \mathcal{L}(\hat{y}^{(i)}, y^{(i)})$

Use gradient descent to optimize parameters to reduce J

Quiz

- What do you think applying this filter to a grayscale image will do?

$$\begin{bmatrix} 0 & 1 & -1 & 0 \\ 1 & 3 & -3 & -1 \\ 1 & 3 & -3 & -1 \\ 0 & 1 & -1 & 0 \end{bmatrix}$$

a) Detect vertical edges
b) Detect horizontal edges
c) Detect image contrast
d) Detect 45 degree edges

- Because pooling layers do not have parameters, they do not affect the backpropagation (derivatives) calculation.
  a) True
  b) False

- In lecture we talked about “parameter sharing” as a benefit of using convolutional networks. Which of the following statements about parameter sharing in ConvNets are true? (Check all that apply.)
  - [x] It allows gradient descent to set many of the parameters to zero, thus making the connections sparse.
  - [ ] It allows parameters learned for one task to be shared even for a different task (transfer learning).
  - [ ] It reduces the total number of parameters, thus reducing overfitting.
  - [x] It allows a feature detector to be used in multiple locations throughout the whole input image/input volume.

- In lecture we talked about “sparsity of connections” as a benefit of using convolutional layers. What does this mean?
  a) Regularization causes gradient descent to set many of the parameters to zero.
  b) Each activation in the next layer depends on only a small number of activations from the previous layer.
  c) Each layer in a convolutional network is connected only to two other layers
  d) Each filter is connected to every channel in the previous layer.

Valid and Same convolutions

→ no padding

"Valid": $n \times n \; * \; f \times f \; \longrightarrow \; n-f+1 \times n-f+1$

$6 \times 6 \; * \; 3 \times 3 \; \longrightarrow \; 4 \times 4$

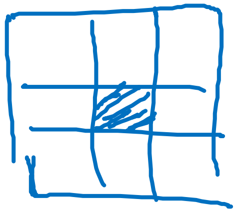
"Same": Pad so that output size is the same as the input size.

$n+2p-f+1 \times n+2p-f+1$

$\cancel{n}+2p-f+1 = \cancel{n} \Rightarrow \boxed{p = \frac{f-1}{2}}$

$3 \times 3 \quad p = \frac{3-1}{2} = 1$ | 5×5, $f=5$ $\quad p=2$

f is usually odd



1×1

3×3

5×5

7×7

Andrew Ng

Technical note on cross-correlation vs. convolution

Convolution in math textbook:

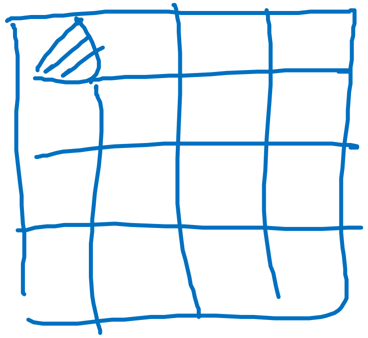
2^{7}	3^{2}	7^{5}	4	6	2
6^{9}	6^{0}	9^{4}	8	7	4
3^{-1}	4^{1}	8^{3}	3	8	9
7	8	3	6	6	3
4	2	1	8	3	4
3	2	4	1	9	8

*

3	4	5
1	0	2
-1	9	7

7	2	5
9	0	4
-1	1	3

=



$$(A * B) * C = A * (B * C)$$

Andrew Ng

Learning to detect edges

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

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

Sobel Filter

3	0	-3
10	0	-10
3	0	-3

Scharr Filter

- You have an input volume that is 32x32x16, and apply max pooling with a stride of 2 and a filter size of 2. What is the output volume?
  a) 16x16x8
  b) 15x15x16
  c) 32x32x8
  d) 16x16x16