Convolutional Neural Networks

Computer vision

deeplearning.ai

Computer Vision Problems

Image Classification

64x64

→ Cat? (0/1)

Object detection

Neural Style Transfer

Andrew Ng

Deep Learning on large images

64x64 × 3

Cat? (0/1)

12288

$x \in \mathbb{R}^{3M}$

$W^{[1]}$ (1000, 3m)

3 billion

x_1
x_2
$\vdots$
x_n

↰ 3M   ↰ 1000

$\hat{y}$

1000×1000×3
= 3 million

Andrew Ng

deeplearning.ai

Convolutional Neural Networks

Edge detection example

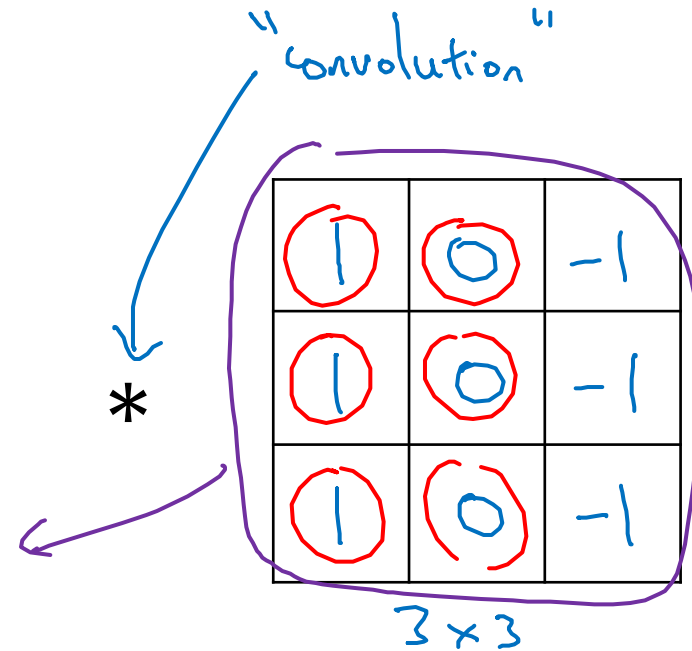# Computer Vision Problem

vertical edges

horizontal edges

Andrew Ng

Vertical edge detection

$3 \times 1 + 1 \times 1 + 2 \times 1 + 0 \times 0 + 5 \times 0 + 7 \times 0 + 1 \times 7 + 8 \times -1 + 2 \times -1 = -5$

6×6

"convolution"

*

3×3

$\rightarrow$ filter
kernel

=

-5	-4	0	8
-10	-2	2	3
0	-2	-4	-7
-3	-2	-3	-16

4×4

Andrew Ng

Vertical edge detection

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

6 x 6

*

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

3 x 3

=

0	30	30	0
0	30	30	0
0	30	30	0
0	30	30	0

4 x 4

Andrew Ng

deeplearning.ai

Convolutional
Neural Networks

More edge
detection

Vertical edge detection examples

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

*

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

=

0	30	30	0
0	30	30	0
0	30	30	0
0	30	30	0

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

*

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

=

0	-30	-30	0
0	-30	-30	0
0	-30	-30	0
0	-30	-30	0

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

6 × 6

*

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

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

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

convolution

w_1	w_2	w_3
w_4	w_5	w_6
w_7	w_8	w_9

3×3

$\begin{bmatrix} 45° \\ 70° \\ 73° \end{bmatrix}$

Andrew Ng

Convolutional
Neural Networks

Padding

deeplearning.ai

Padding

- Shrinks output
- throw away info from edge

3×3
$f \times f$

$= $

6×6

$6 \times 6 \rightarrow 8 \times 8$
$\underline{n \times n}$

$n - f + 1 \times n - f + 1$
$6 - 3 + 1 = 4$

4×4

$P = \text{padding} = \underline{1}$

$n + 2p - f + 1 \times n + 2p - f + 1$
$6 + 2 - 3 + 1 \times \underline{\quad} = 6 \times 6$

$p = 2$

Andrew Ng

Valid and Same convolutions

$\rightarrow$ no padding

"Valid": $n \times n$ * $f \times f$ $\longrightarrow$ $\underline{n - f + 1} \times n - f + 1$

6×6 * 3×3 $\longrightarrow$ 4×4

"Same": Pad so that output size is the <u>same</u> as the input size.

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

f is usually odd

1×1
3×3
5×5
7×7

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

3×3 $p = \frac{3-1}{2} = 1$ | 5×5 $p = 2$

$f = 5$

Andrew Ng

deeplearning.ai

Convolutional
Neural Networks

Strided
convolutions

Strided convolution

| 2³ | 3⁴ | 7³ | 4⁴ | 6³ | 2⁴ | 9⁴ |

$$
\begin{array}{ccccccc}
2^{3} & 3^{4} & 7^{3} & 4^{4} & 6^{3} & 2^{4} & 9^{4} \\
6^{1} & 6^{0} & 9^{1} & 8^{0} & 7^{1} & 4^{0} & 3^{2} \\
3^{3} & 4^{4} & 8^{3} & 3^{4} & 8^{3} & 9^{4} & 7^{4} \\
7^{1} & 8^{0} & 3^{1} & 6^{0} & 6^{1} & 3^{0} & 4^{2} \\
4^{3} & 2^{4} & 1^{3} & 8^{4} & 3^{3} & 4^{4} & 6^{4} \\
3^{1} & 2^{0} & 4^{1} & 1^{0} & 9^{1} & 8^{0} & 3^{2} \\
0^{-1} & 1^{0} & 3^{-1} & 9^{0} & 2^{-1} & 1^{0} & 4^{3}
\end{array}
$$

7×7

$*$

3	4	4
1	0	2
-1	0	3

3×3

Stride = 2

$=$

91	100	83
69	91	127
44	72	74

3×3

$\lfloor z \rfloor = floor(z)$

$n \times n$ * $f \times f$

paddg p stride s

$s = 2$

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

$$\frac{7 + 0 - 3}{2} + 1 = \frac{4}{2} + 1 = 3$$

Andrew Ng

Summary of convolutions

$n \times n$ image $f \times f$ filter

padding p stride s

Output Size:

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

Technical note on cross-correlation vs. convolution

Convolution in math textbook:

2	3	7	4	6	2
6	6	9	8	7	4
3	4	8	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

=

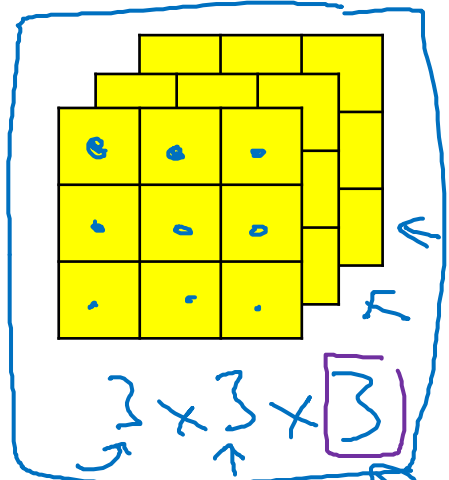$$(A * B) * C = A * (B * C)$$

Convolutional
Neural Networks

Convolutions over
volumes

deeplearning.ai

Convolutions on RGB images

$6 \times 6 \times 3$

height

width

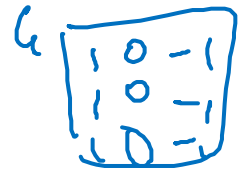#channels

$*$

$3 \times 3 \times 3$

$=$

4×4

Andrew Ng

Convolutions on RGB image

$6 \times 6 \times \boxed{3}$

$*$

$3 \times 3 \times \boxed{3}$

27 numbers

R
1	0	-1
1	0	-1
1	0	-1

G

B
$\rightarrow 3 \times 3 \times 3$

R
1	0	-1
1	0	-1
1	0	-1

G

B
$\rightarrow 3 \times 3 \times 3$

$=$

4 x 4

Andrew Ng

Multiple filters

Vertical edge

horizontal edge

$6 \times 6 \times 3$

channels

$3 \times 3 \times 3$

$3 \times 3 \times 3$

4×4

4×4

$4 \times 4 \times 2$

$4 \times 4 \times 2$

Summary: $n \times n \times \boxed{n_c} \quad * \quad f \times f \times \boxed{n_c} \quad \longrightarrow \quad \dfrac{n-f+1}{4} \times \dfrac{n-f+1}{4} \times n_c'$

$6 \times 6 \times 3 \qquad\qquad 3 \times 3 \times 3 \qquad\qquad 4 \quad\times\quad 4 \quad \times 2 \quad$ #filters

Andrew Ng

Convolutional
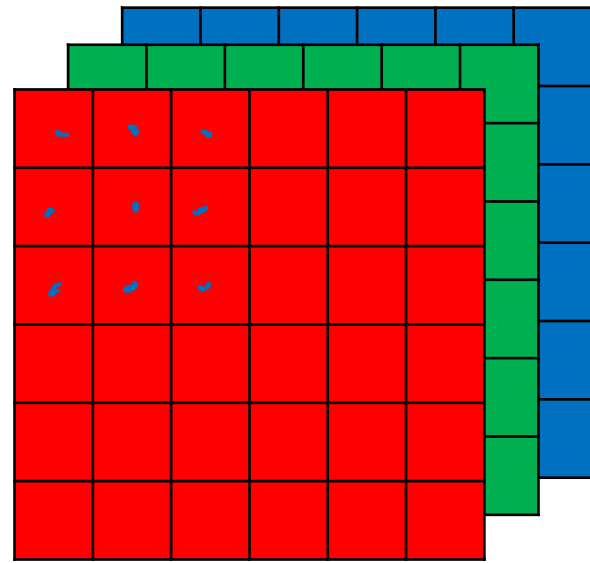Neural Networks

One layer of a
convolutional
network

deeplearning.ai

Example of a layer

$6 \times 6 \times 3$

$a^{[0]}$

$z^{[1]} = W^{[1]} a^{[0]} + b^{[1]}$

$a^{[1]} = g(z^{[1]})$

filters
10

$3 \times 3 \times 3$

$3 \times 3 \times 3$

"$W^{[1]}$"

$6 \times 6 \times 3$ $a^{[0]}$ → $a^{[1]}$ $4 \times 4 \times 2$

"$W^{[1]} a^{[0]}$"   R $z^{[1]}$

→ ReLU $+ b_1$ → 4×4

4×4

R $+ b_2$ → 4×4

→ ReLU 4×4

$4 \times 4 \times 2$ ← $a^{[1]}$ $4 \times 4 \times 10$

Andrew Ng

Number of parameters in one layer

If you have 10 filters that are 3 x 3 x 3 in one layer of a neural network, how many parameters does that layer have?

3×3×3

27 parameters.

+bias

→ 28 parameters.

280 parameters.

Summary of notation

If layer l is a convolution layer:

$f^{[l]}$ = filter size

$p^{[l]}$ = padding

$s^{[l]}$ = stride

$\boxed{n_c^{[l]}}$ = number of filters

$\rightarrow$ Each filter is: $f^{[l]} \times f^{[l]} \times \boxed{n_c^{[l-1]}}$

Activations: $a^{[l]} \rightarrow n_H^{[l]} \times n_W^{[l]} \times n_c^{[l]}$

Weights: $f^{[l]} \times f^{[l]} \times n_c^{[l-1]} \times n_c^{[l]}$

bias: $n_c^{[l]} \quad - \quad (1,1,1,n_c^{[l]})$ $\underset{\text{#filters in layer } l.}{\nearrow}$

Input: $\boxed{n_H^{[l-1]} \times n_W^{[l-1]} \times n_c^{[l-1]}}$ $\leftarrow$

Output: $n_H^{[l]} \times n_W^{[l]} \times n_c^{[l]}$ $\leftarrow$

$n_{HW}^{[l]} = \left\lfloor \dfrac{n_{HW}^{[l-1]} + 2p^{[l]} - f^{[l]}}{s^{[l]}} + 1 \right\rfloor$

$A^{[l]} \rightarrow m \times n_H^{[l]} \times n_W^{[l]} \times n_c^{[l]}$

$n_c \times n_H \times n_W$

Convolutional Neural Networks

deeplearning.ai

A simple convolution network example

Example ConvNet

$a^{[1]}$

$a^{[2]}$

$\to f^{[1]} = 3$

$\to S^{[1]} = 1$

$\to p^{[1]} = 0$

$\to 10$ filters

x

$39 \times 39 \times 3$

$n_H^{[0]} = n_W^{[0]} = 39$

$n_c^{[0]} = 3$

$37 \times 37 \times 10$

$n_H^{[1]} = n_W^{[1]} = 37$

$n_c^{[1]} = 10$

$f^{[2]} = 5$

$S^{[2]} = 2$

$p^{[2]} = 0$

20 filters

$17 \times 17 \times 20$

$n_H^{[2]} = n_W^{[2]} = 17$

$n_c^{[2]} = 20$

$f^{[3]} = 5$

$S^{[3]} = 2$

40 filters

$7 \times 7 \times 40$

$\hat{y}$

logistic / softmax

1960

$$\frac{n + 2p - f}{s} + 1$$

$$\frac{39 + 0 - 3}{1} + 1 = 37$$

Types of layer in a convolutional network:

- Convolution (CONV) ←

- Pooling (POOL) ←

- Fully connected (FC) ←

Convolutional Neural Networks

Pooling layers

deeplearning.ai

Pooling layer: Max pooling

1	3	2	1
2	9	1	1
1	3	2	3
5	6	1	2

Pooling layer: Max pooling

$$5 \times 5 \times n_c$$

$$3 \times 3 \times n_c$$

$$\frac{f = 3}{S = 1}$$

$$\left(\frac{n + 2p - f}{S} + 1 \right)$$

Andrew Ng

Pooling layer: Average pooling

$f = 2$

$s = 2$

$7 \times 7 \times 1000 \longrightarrow 1 \times 1 \times 1000$

Summary of pooling

Hyperparameters:

f : filter size

s : stride

Max or average pooling

$f=2, s=2$

$f=3, s=2$

$\Rightarrow p : \text{padding.}$

No parameters to learn!

$n_H \times n_W \times n_C$

$\downarrow$

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

$\times \; n_C$

deeplearning.ai

Convolutional
Neural Networks

Convolutional neural
network example

Neural network example (LeNet-5)

CONV1 ↓

POOL1 ↓

CONV2

POOL2

$32 \times 32 \times 3$

$f=5$
$s=1$

$28 \times 28 \times 6$

maxpool
$f=2$
$s=2$

$14 \times 14 \times 6$

$f=5$
$s=1$

$10 \times 10 \times 16$

maxpool
$f=2$
$s=2$

$5 \times 5 \times 16$

(400)

Layer 1

Layer 2

FC3

FC4

→ 0 Softmax
(10 outputs)

$0, 1, 2, \ldots 9$

$W^{[3]}$ (120, 400)

$b^{[3]}$ (120)

(120)

84

$n_H, n_W \downarrow$

$n_C \uparrow$

CONV — POOL — CONV — POOL — FC — FC — FC — SOFTMAX

Andrew Ng

Neural network example

	Activation shape	Activation Size	# parameters
Input:	(32,32,3)	3,072 $a^{[0]}$	0
CONV1 (f=5, s=1)	(28,28,8)	6,272	608
POOL1	(14,14,8)	1,568	0
CONV2 (f=5, s=1)	(10,10,16)	1,600	3216
POOL2	(5,5,16)	400	0
FC3	(120,1)	120	48120
FC4	(84,1)	84	10164
Softmax	(10,1)	10	850

Convolutional
Neural Networks

Why convolutions?

deeplearning.ai

Why convolutions

$5 \times 5 \quad - \quad 25$

26

$6 \times 26 = 156$ parameters

$f = 5$

6 filters

$32 \times 32 \times 3$

$28 \times 28 \times 6$

3,072

4,704

$3,072 \times 4,704 \approx 14M$

3072

4704

Why convolutions

translation invariance

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

*

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

3×3

=

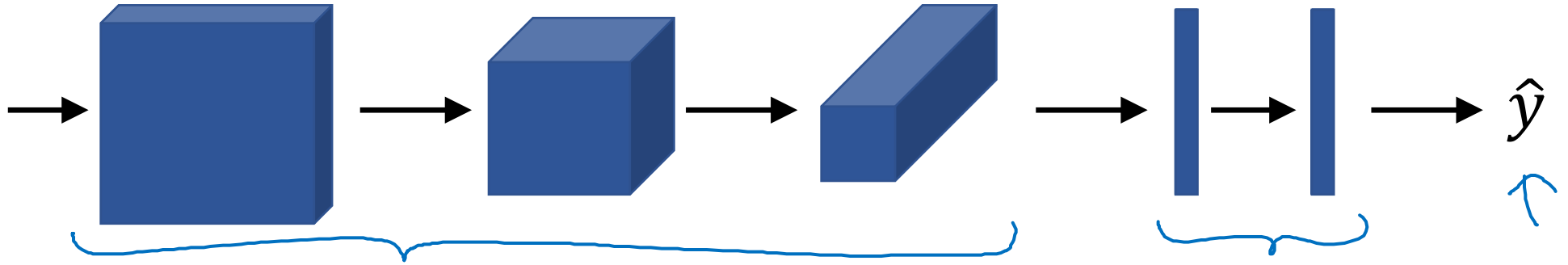| 0	30	30	0
0	30	30	0
0	30	30	0
0	30	30	0

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.

Andrew Ng

Putting it together

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

w, b

$\hat{y}$

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

Use gradient descent to optimize parameters to reduce J

Andrew Ng