Convolutional Neural Networks

Computer vision

deeplearning.ai

1

Computer Vision Problems

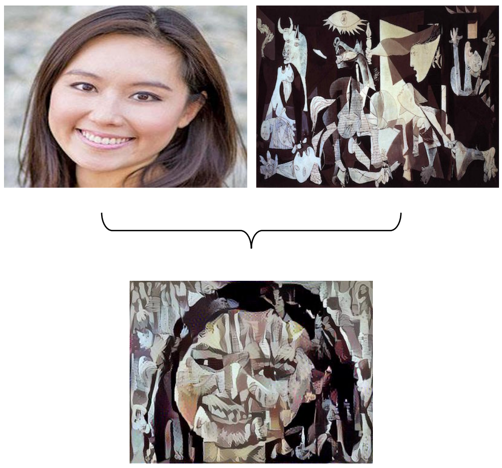
Image Classification

64x64

→ Cat? (0/1)

Object detection

Neural Style Transfer



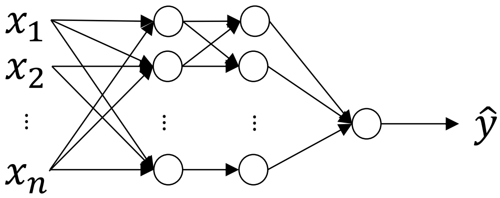
Andrew Ng

2

Deep Learning on large images

64x64

→ Cat? (0/1)



Andrew Ng

3

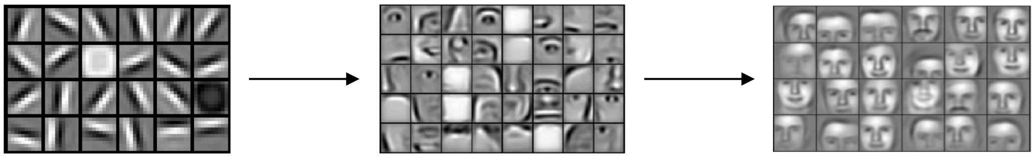
deeplearning.ai

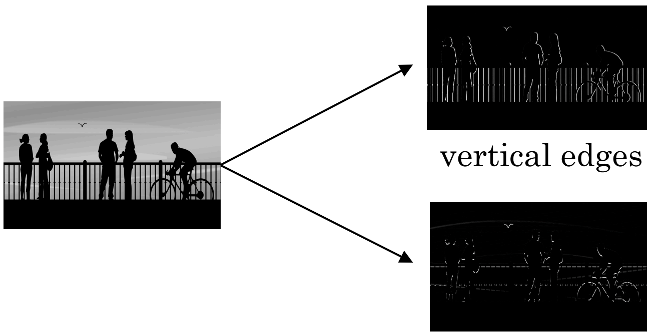
Convolutional Neural Networks

Edge detection example

4

Computer Vision Problem





Andrew Ng

5

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

*

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

=

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

Andrew Ng

6

Vertical edge detection

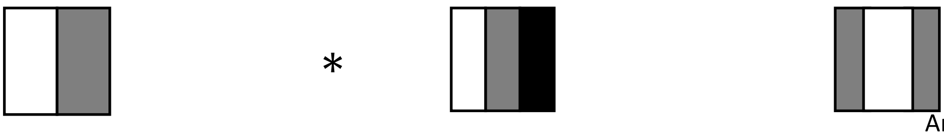
10	10	10	0	0	0
10	10	10	0	0	0
10	10	10	0	0	0
10	10	10	0	0	0
10	10	10	0	0	0
10	10	10	0	0	0

*

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



Andrew Ng

7

Convolutional Neural Networks

deeplearning.ai

More edge detection

8

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

Andrew Ng

9

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

Andrew Ng

10

Learning to detect edges

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

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

w_1	w_2	w_3
w_4	w_5	w_6
w_7	w_8	w_9

Andrew Ng

11

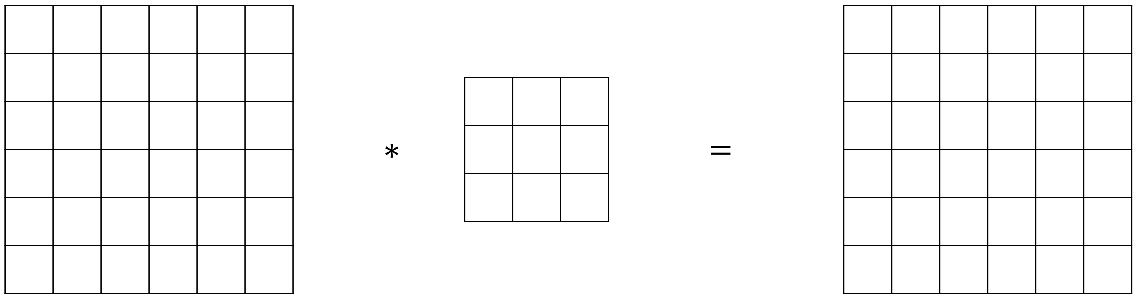
deeplearning.ai

Convolutional Neural Networks

Padding

12

Padding



Valid and Same convolutions

“Valid”:

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

Convolutional Neural Networks

deeplearning.ai

Strided convolutions

15

Strided convolution

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

$*$

3	4	4
1	0	2
-1	0	3

$=$

Andrew Ng

16

Summary of convolutions

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

padding p $\qquad$ stride s

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

Andrew Ng

17

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

Andrew Ng

18

Convolutional
Neural Networks

deeplearning.ai

Convolutions over volumes

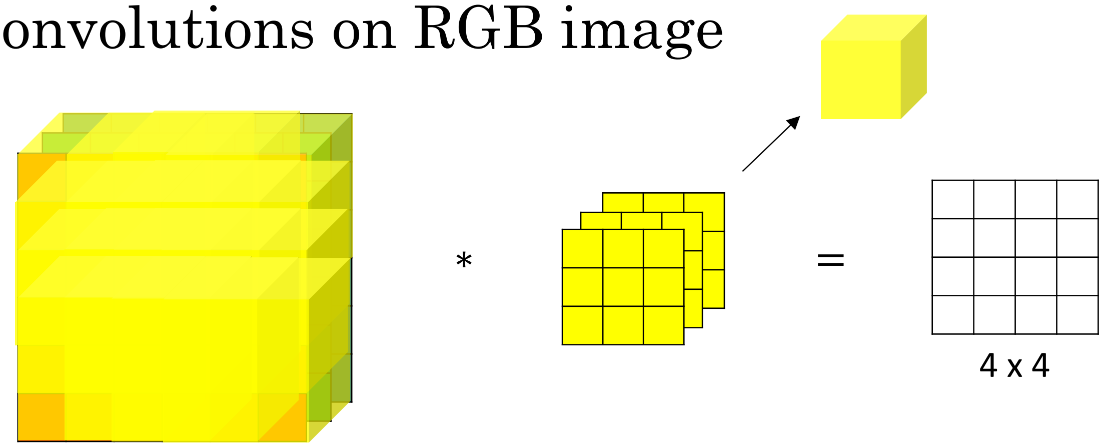
19

Convolutions on RGB images

Andrew Ng

20

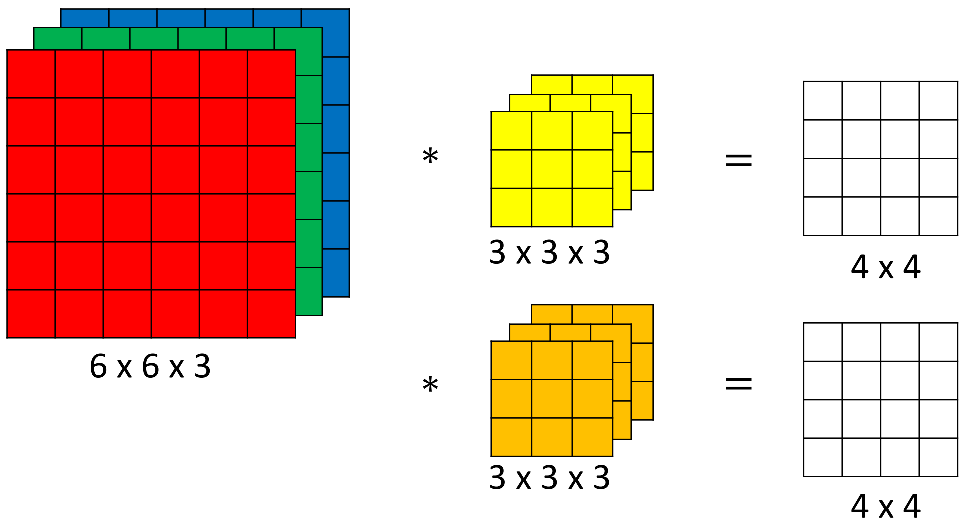
Convolutions on RGB image



Andrew Ng

21

Multiple filters



Andrew Ng

22

deeplearning.ai

Convolutional Neural Networks

A simple convolution network example

23

Example ConvNet

Andrew Ng

24

Types of layer in a convolutional network:

- Convolution
- Pooling
- Fully connected

Andrew Ng

25

deeplearning.ai

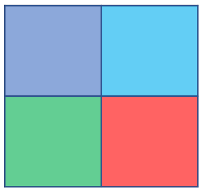
Convolutional Neural Networks

Pooling layers

26

Pooling layer: Max pooling

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

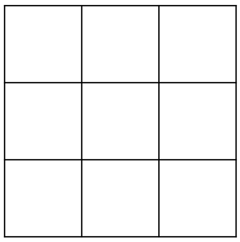


Andrew Ng

27

Pooling layer: Max pooling

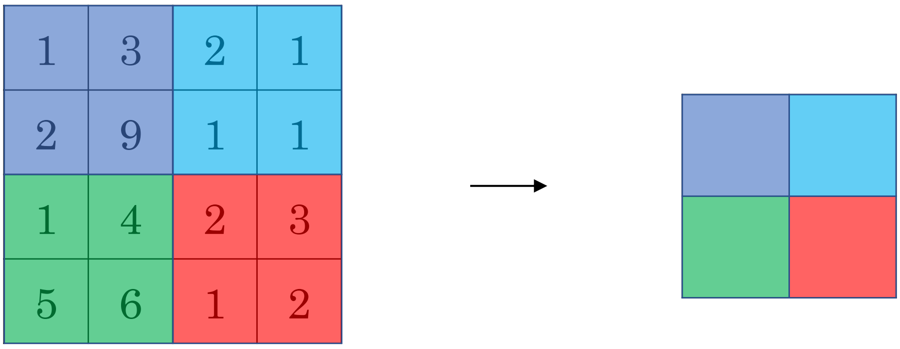
1	3	2	1	3
2	9	1	1	5
1	3	2	3	2
8	3	5	1	0
5	6	1	2	9



Andrew Ng

28

Pooling layer: Average pooling



Andrew Ng

29

Summary of pooling

Hyperparameters:

f : filter size

s : stride

Max or average pooling

Andrew Ng

30

deeplearning.ai

Convolutional
Neural Networks

Convolutional neural network example

31

Neural network example

Andrew Ng

32

Neural network example

	Activation shape	Activation Size	# parameters
Input:	(32,32,3)	3,072	0

Andrew Ng

33

deeplearning.ai

Convolutional Neural Networks

Why convolutions?

34

Why convolutions

Andrew Ng

35

Why convolutions

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

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.

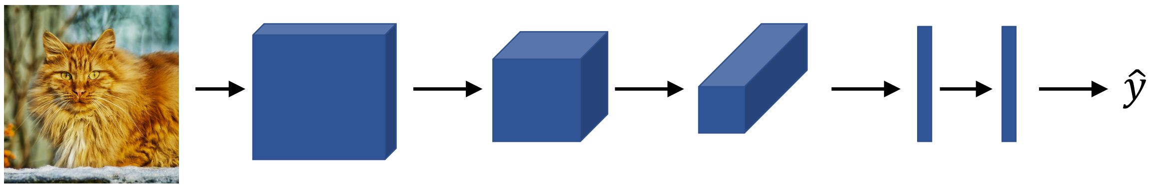
Sparsity of connections: In each layer, each output value depends only on a small number of inputs.

Andrew Ng

36

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

Andrew Ng

37