Convolutional Neural Network

Hung-yi Lee

Can the network be simplified by considering the properties of images?

提出背景

图像分类可以用普通的 Fully-Connect 神经网络来完成，即 DNN

但是一张图所有 pixel 输入太多，导致每个 pixel 都会有一个 w 被训练方维数

解决什么？

CNN 可以简化这件事，类似于一种预处理，大大减少了 w 个数

所以 CNN 的 model 比 DNN 更简单

Why CNN for Image

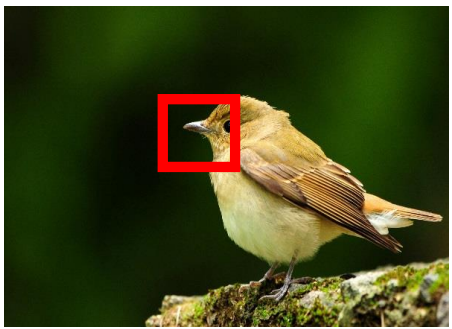
- Some patterns are much smaller than the whole image

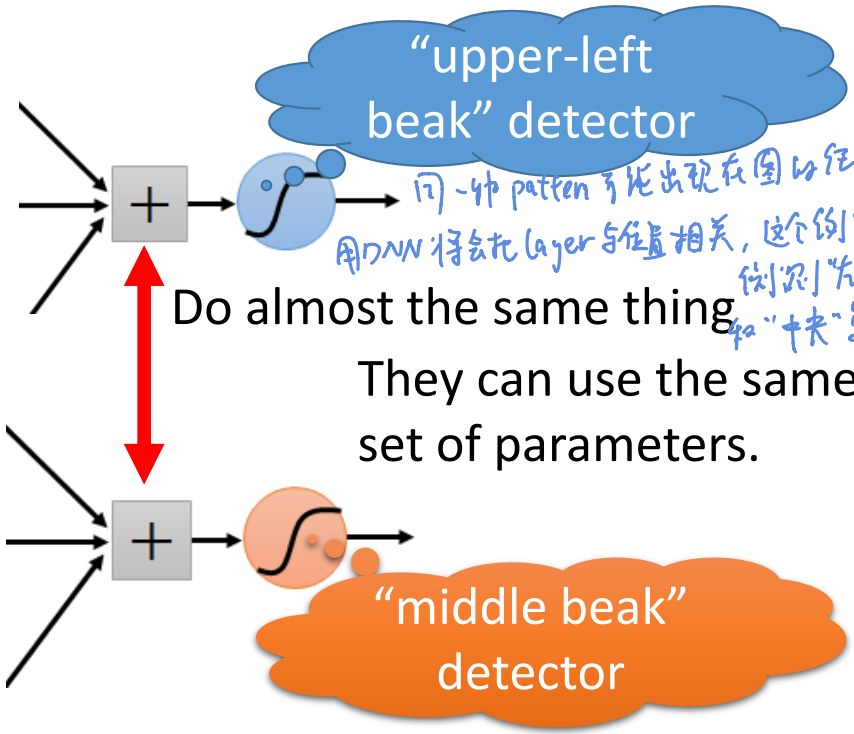
A neuron does not have to see the whole image to discover the pattern.

Connecting to small region with less parameters

Why CNN for Image

- The same patterns appear in different regions.





Why CNN for Image

- Subsampling the pixels will not change the object

bird

subsampling

采样，resize图片，减少 input

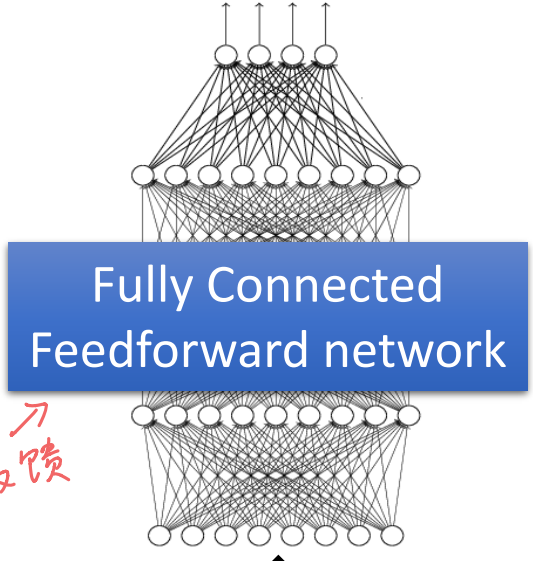
bird

We can subsample the pixels to make image smaller

→ Less parameters for the network to process the image

The whole CNN

cat dog ……



卷积

Convolution

池化

Max Pooling

Convolution

Max Pooling

Can repeat many times

Flatten

The whole CNN

Property 1

- Some patterns are much smaller than the whole image

取图中的部分

Property 2

- The same patterns appear in different regions.

patten在不同区域

Property 3

- Subsampling the pixels will not change the object

Convolution

Max Pooling

Convolution

Max Pooling

Can repeat many times

Flatten

The whole CNN

Convolution

Max Pooling

Convolution

Max Pooling

Can repeat many times

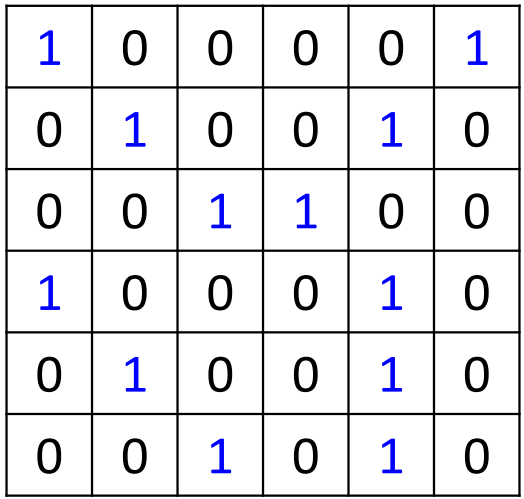
Flatten

Fully Connected Feedforward network

cat dog ……

CNN – Convolution

Those are the network parameters to be learned.



1	0	0	0	0	1
0	1	0	0	1	0
0	0	1	1	0	0
1	0	0	0	1	0
0	1	0	0	1	0
0	0	1	0	1	0

6 x 6 image

1	-1	-1
-1	1	-1
-1	-1	1

Filter 1

Matrix

-1	1	-1
-1	1	-1
-1	1	-1

Filter 2

Matrix

Property 1

Each filter detects a small pattern (3 x 3).

CNN – Convolution

Filter

1	-1	-1
-1	1	-1
-1	-1	1

Filter 1

matrix B

stride=1

matrix A

1	0	0	0	0	1
0	1	0	0	1	0
0	0	1	1	0	0
1	0	0	0	1	0
0	1	0	0	1	0
0	0	1	0	1	0

6 x 6 image

A·B 内积 (对应位置元素相乘，加和)

=

3

-1

=

C·B 内积

CNN – Convolution

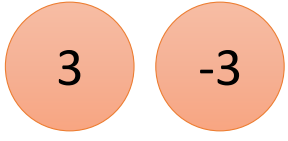
1	-1	-1
-1	1	-1
-1	-1	1

Filter 1

If stride=2 ⇐每次移2格

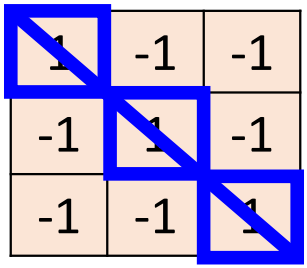
1	0	0	0	0	1
0	1	0	0	1	0
0	0	1	1	0	0
1	0	0	0	1	0
0	1	0	0	1	0
0	0	1	0	1	0

6 x 6 image



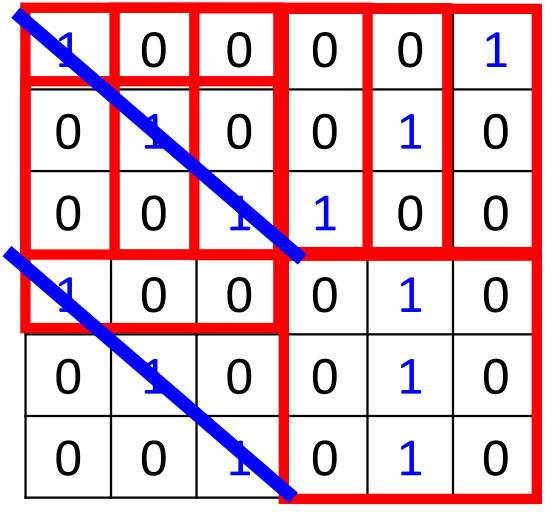
We set stride=1 below

CNN – Convolution



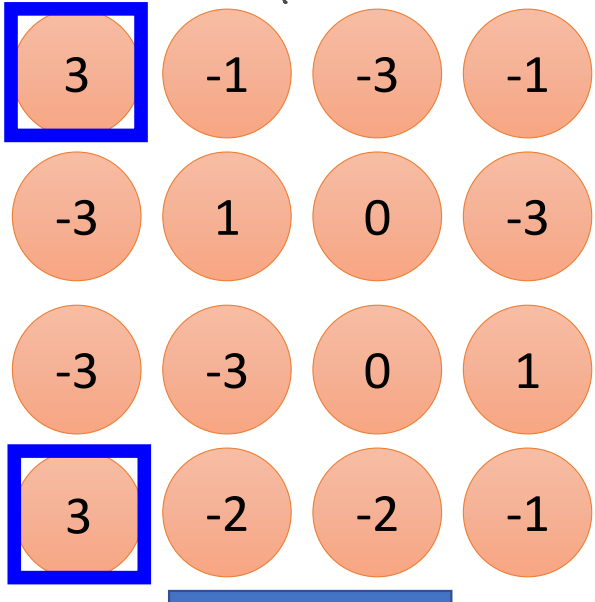
Filter 1

stride=1



6 x 6 image

值越大，与 Filter 匹配度越高 (patten)



Property 2

CNN – Convolution

-1	1	-1
-1	1	-1
-1	1	-1

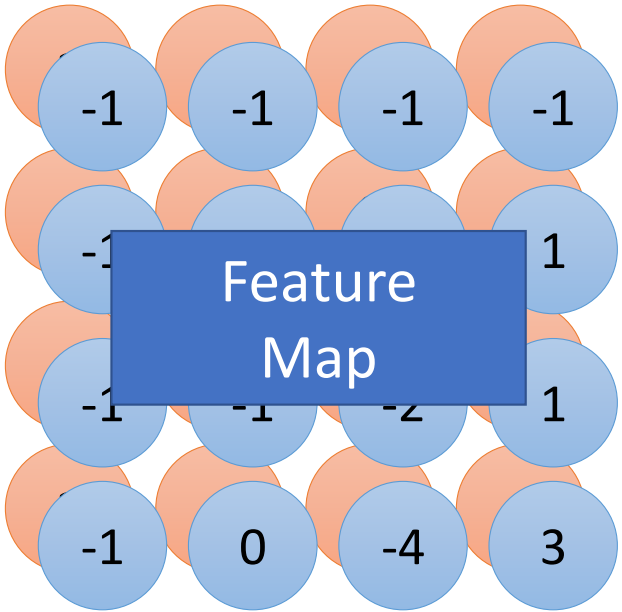
Filter 2

stride=1

1	0	0	0	0	1
0	1	0	0	1	0
0	0	1	1	0	0
1	0	0	0	1	0
0	1	0	0	1	0
0	0	1	0	1	0

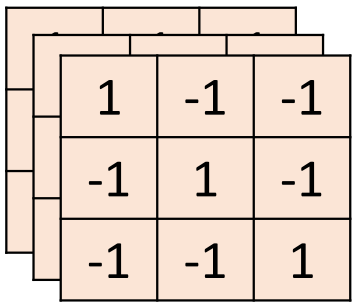
6 x 6 image

Do the same process for every filter

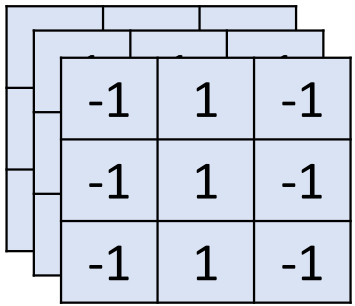


4 x 4 image

CNN – Colorful image

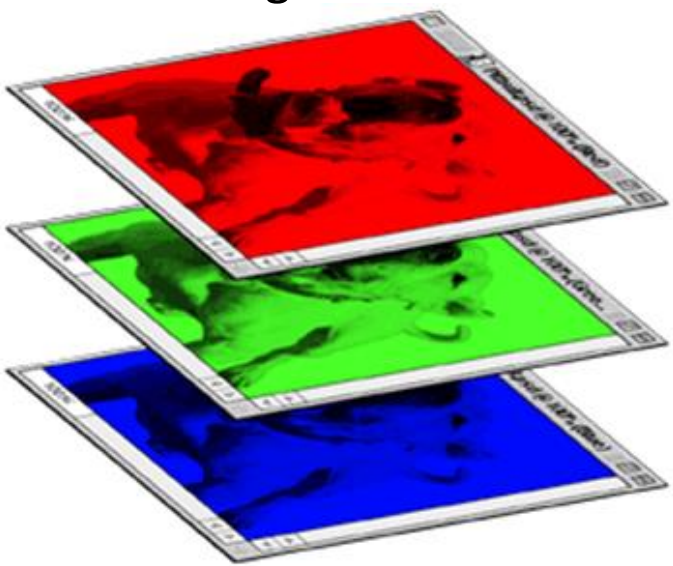


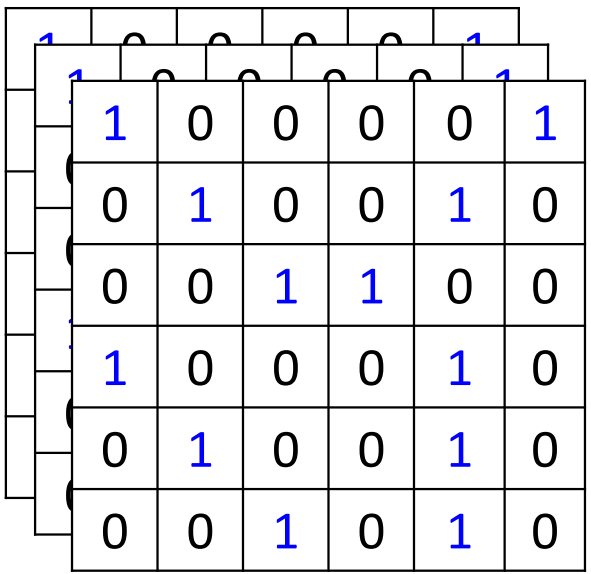
Filter 1



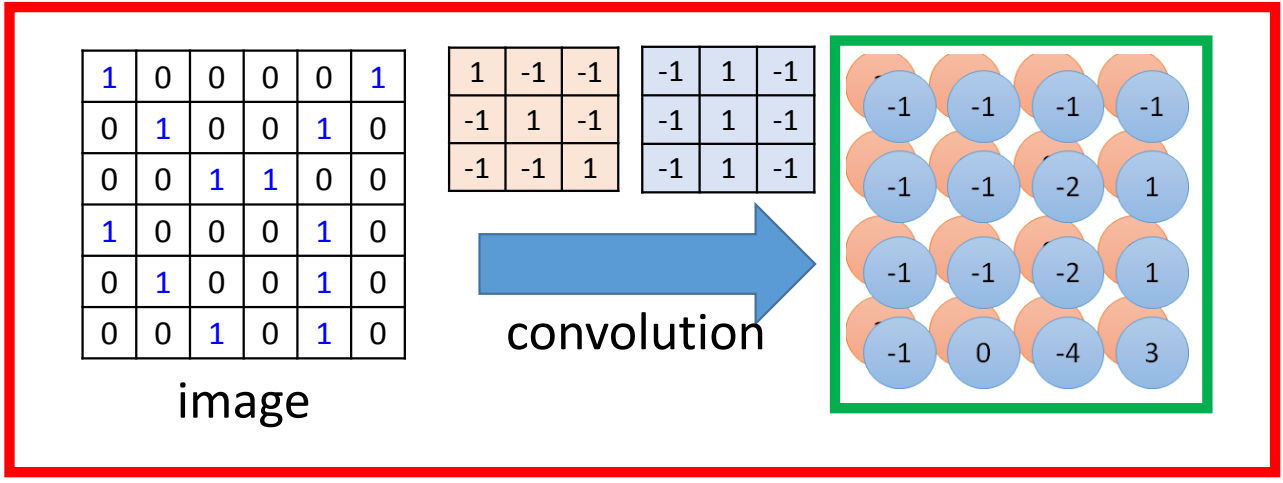
Filter 2

Colorful image

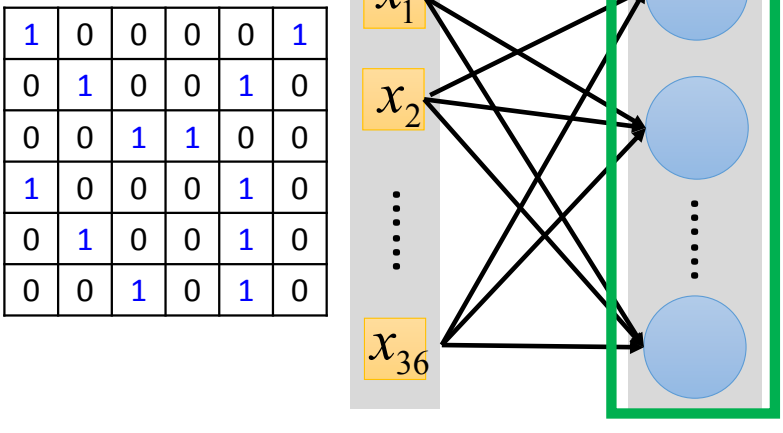


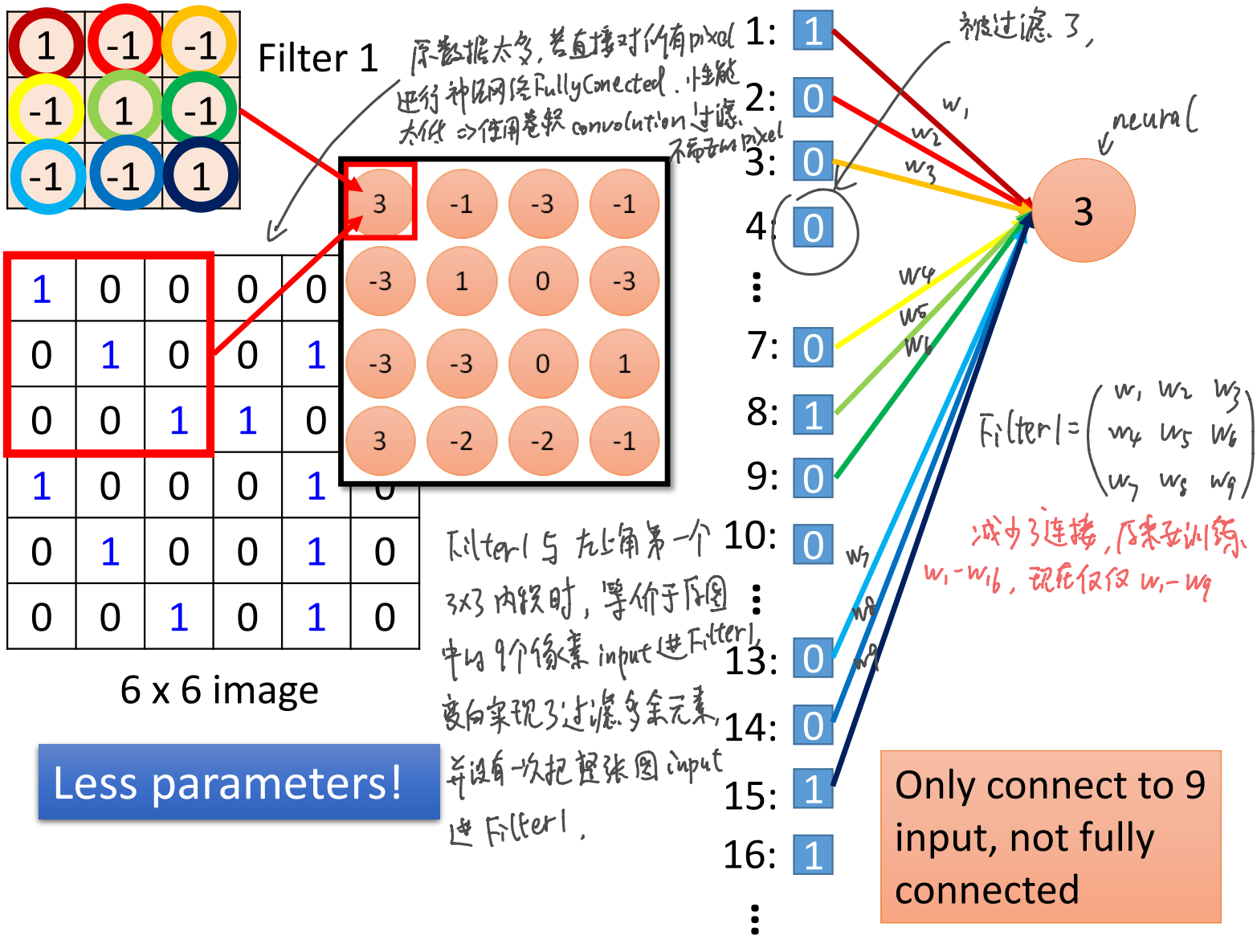


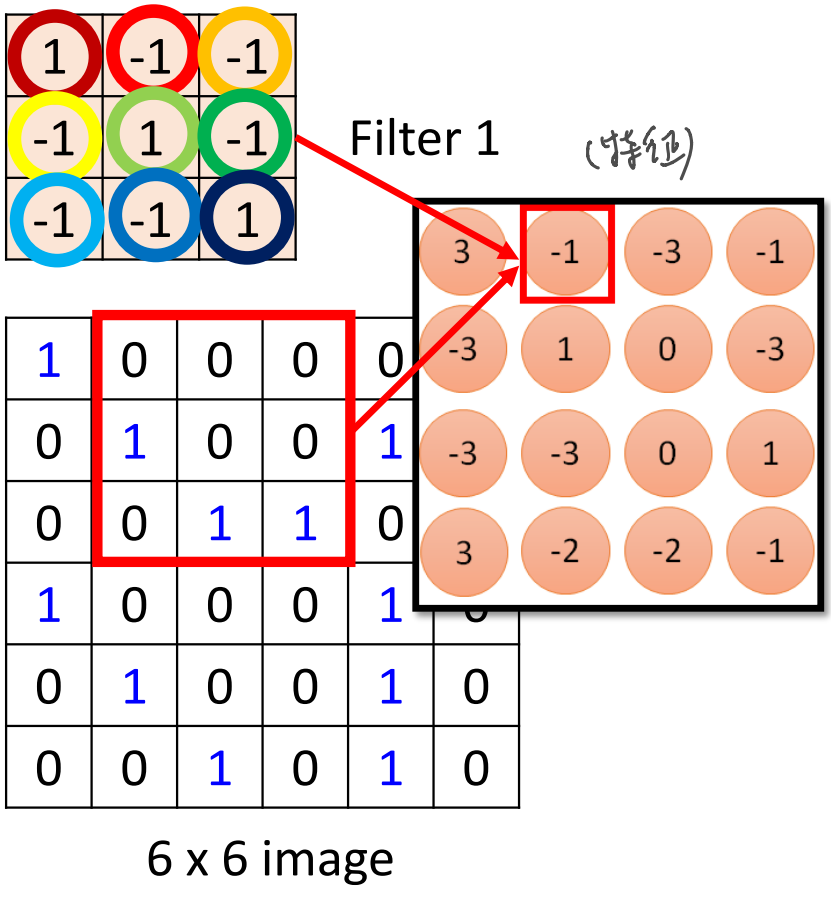
Convolution v.s. Fully Connected



Fully-connected

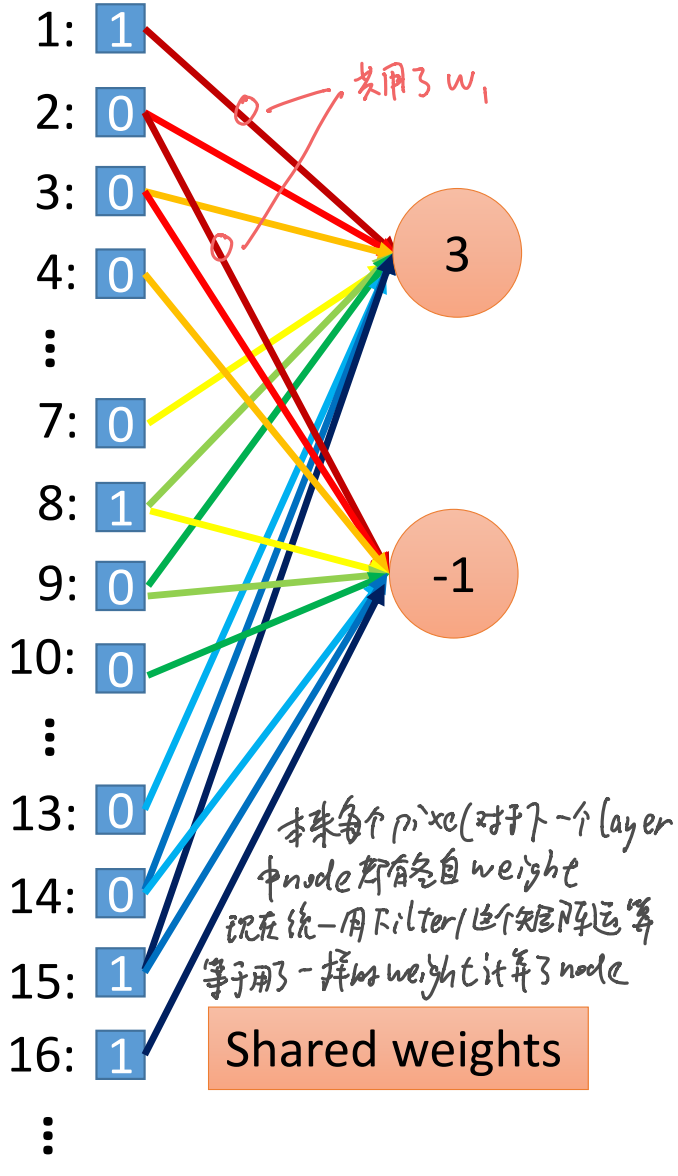



1 -1 -1
-1 1 -1
-1 -1 1
Filter 1
原数据太多，若直接对所有pixel进行神经网络FullyConected，计算能力太低 => 使用卷积 convolution 过滤不重要pixel
1 0 0 0 0
0 1 0 0 1
0 0 1 1 0
1 0 0 0 1 0
0 1 0 0 1 0
0 0 1 0 1 0
6 x 6 image
3 -1 -3 -1
-3 1 0 -3
-3 -3 0 1
3 -2 -2 -1
Less parameters!
Filter1 与 左上角第一个 3x3 内积时，等价于原图中的9个像素 input 进 Filter1，变向实现了过滤多余元素，并没有一次把整张图 input 进 Filter1.
1:
2:
3:
4:
7:
8:
9:
10:
13:
14:
15:
16:
1
0
0
0
0
1
0
0
1
0
被过滤了，
w1
w2
w3
neuron
3
w4
w5
w6
w7
w8
w9
Filter1 = w1 w2 w3 / w4 w5 w6 / w7 w8 w9
减少了连接，原来要训练 w1-w16，现在仅仅 w1-w9
Only connect to 9 input, not fully connected



Less parameters!

Even less parameters!



The whole CNN

Convolution

Max Pooling

Convolution

Max Pooling

Can repeat many times

Flatten

Fully Connected Feedforward network

cat dog ……

(进一步提取特征)

CNN – Max Pooling 池化

进一步缩小图片

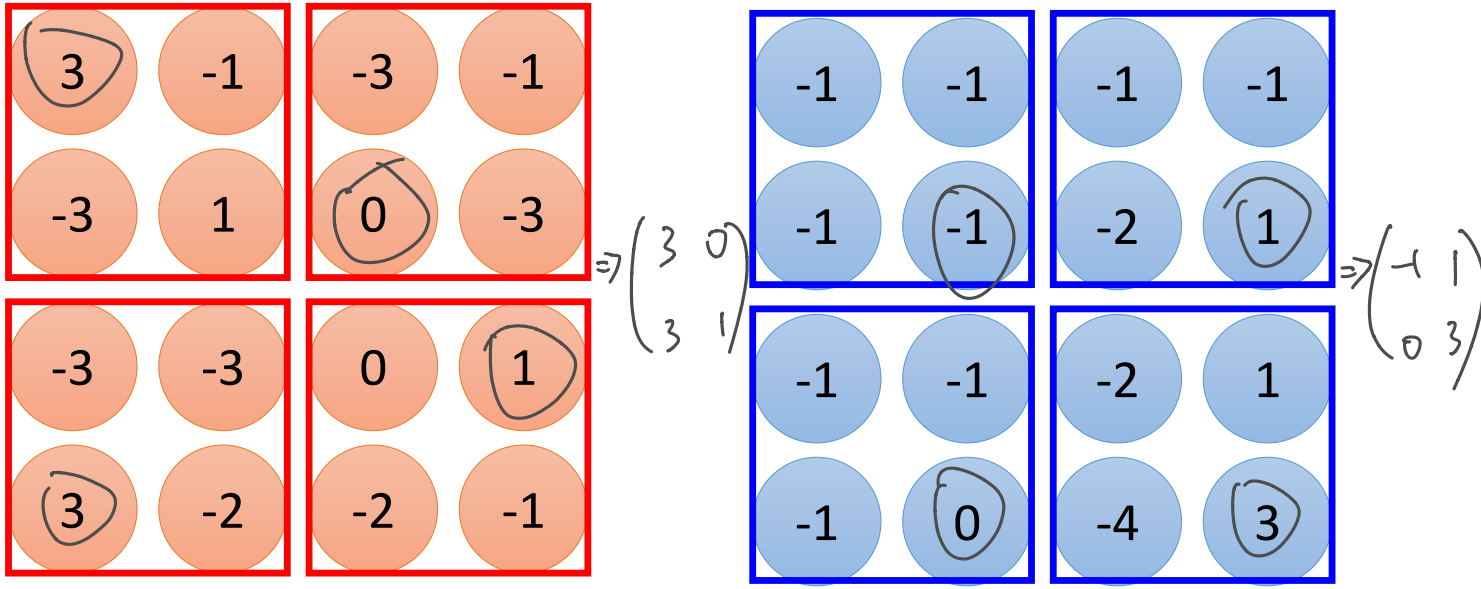
1	-1	-1
-1	1	-1
-1	-1	1

Filter 1

-1	1	-1
-1	1	-1
-1	1	-1

Filter 2

就四，取组内最大

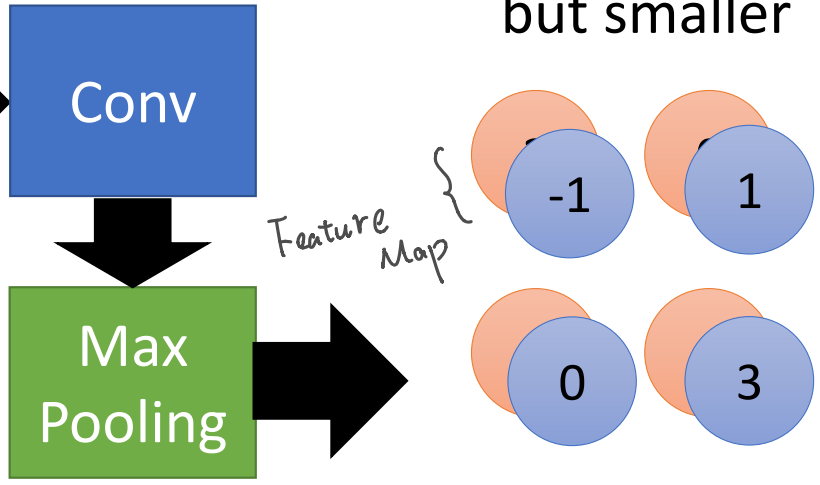


CNN – Max Pooling

New image but smaller

1	0	0	0	0	1
0	1	0	0	1	0
0	0	1	1	0	0
1	0	0	0	1	0
0	1	0	0	1	0
0	0	1	0	1	0

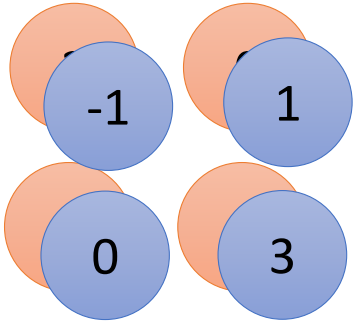
6 x 6 image



2 x 2 image

Each filter is a channel

The whole CNN



A new image

Smaller than the original image

The number of the channel is the number of filters

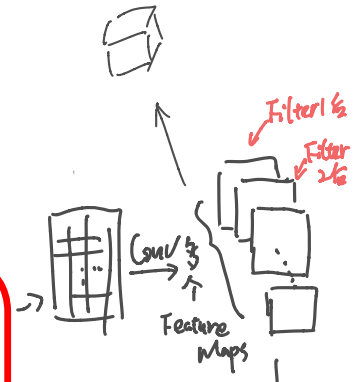
Convolution

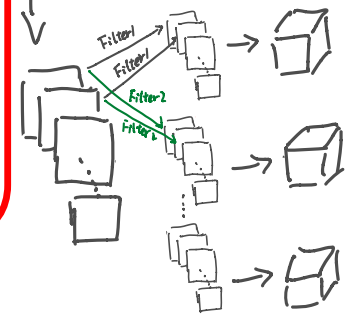
Max Pooling

Convolution

Max Pooling

Can repeat many times





The whole CNN

Convolution

Max Pooling

A new image

Convolution

Max Pooling

A new image

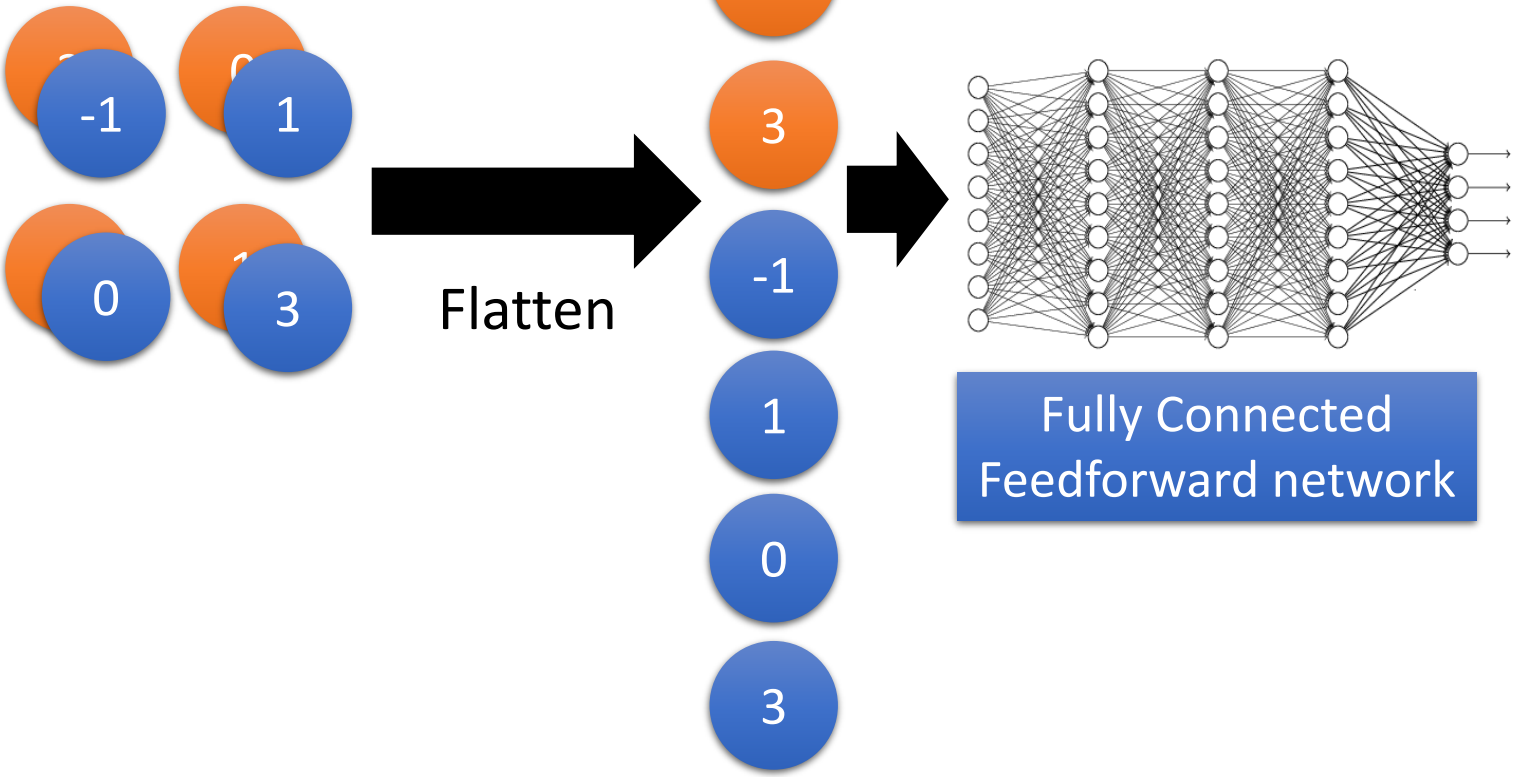
Flatten

Fully Connected Feedforward network

cat dog ……

Flatten

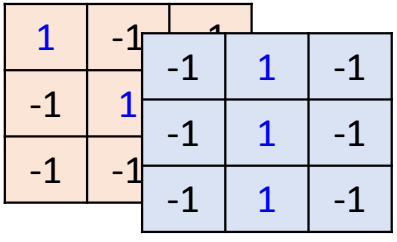
卷积后，拉直



CNN in Keras

Only modified the ***network structure*** and ***input format (vector -> 3-D tensor)***

```
model2.add( Convolution2D( 25,3,3,
            input_shape=(28,28,1)) )
```



.....

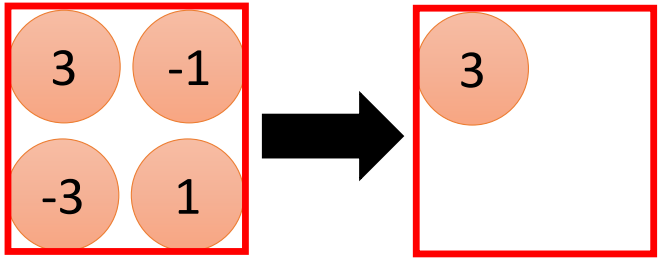
There are 25 3x3 filters.

Input_shape = (28 , 28 , 1)

28 x 28 pixels

1: black/white, 3: RGB

```
model2.add(MaxPooling2D((2,2)))
```



input

Convolution

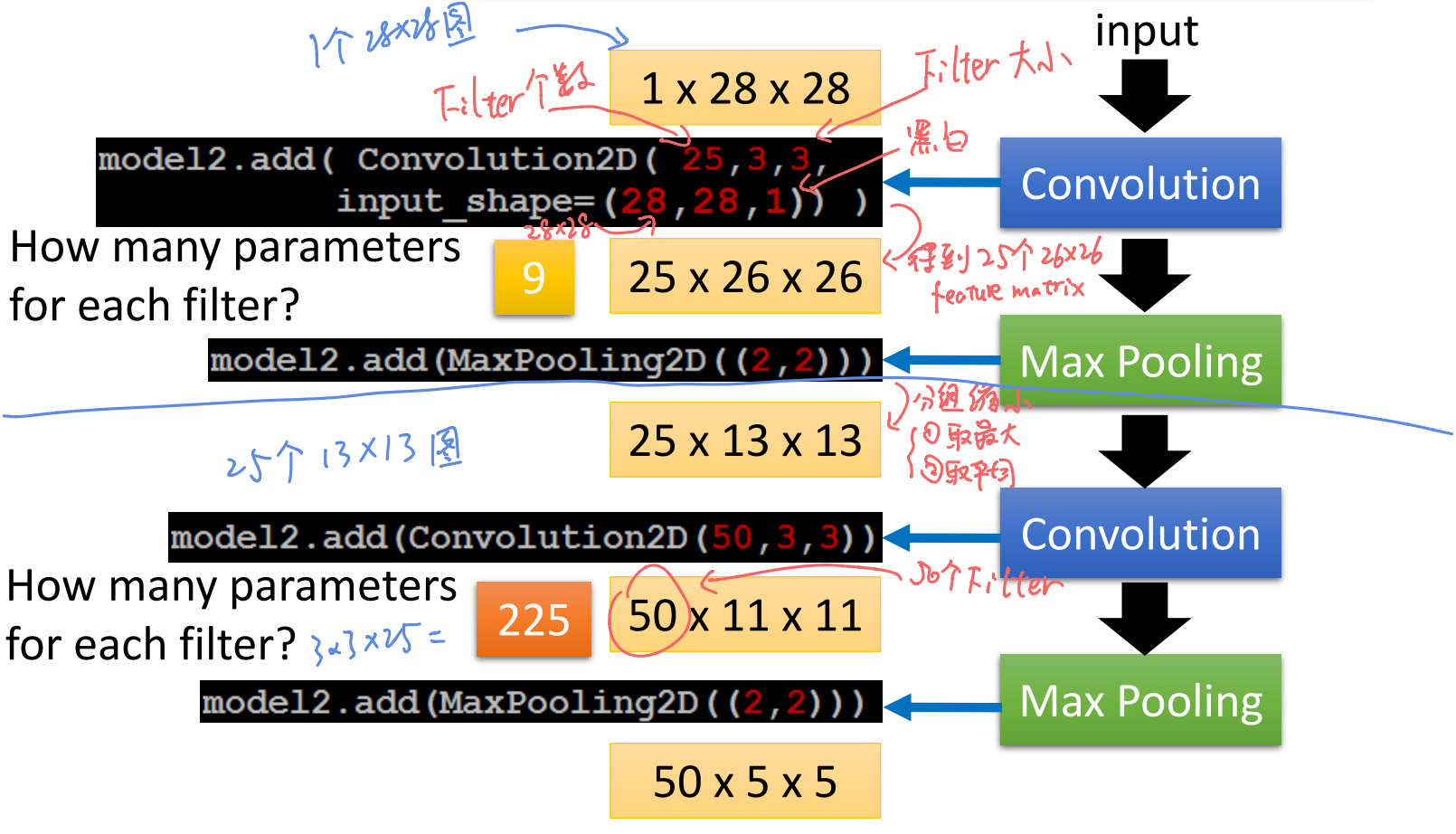
Max Pooling

Convolution

Max Pooling

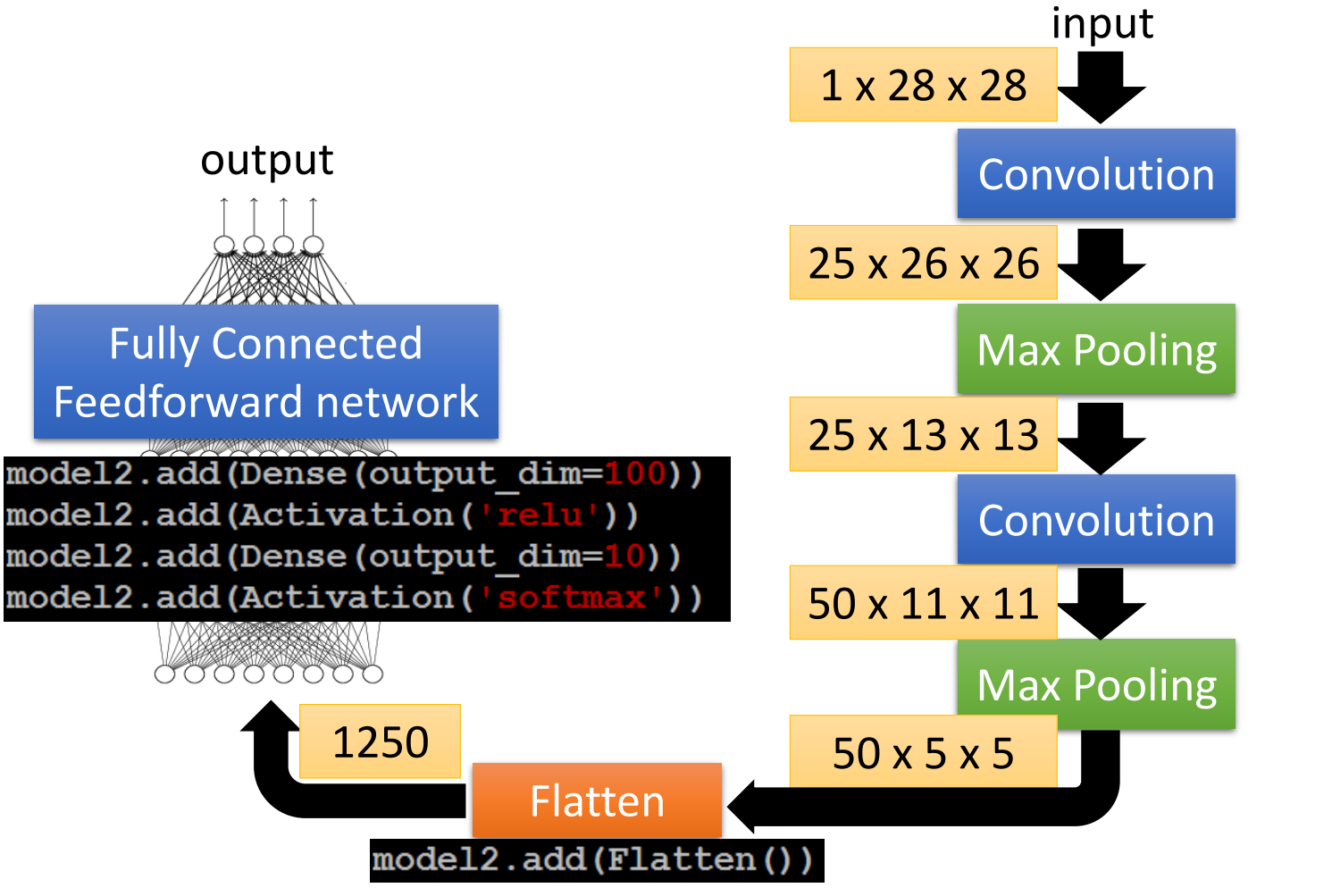
CNN in Keras

Only modified the ***network structure*** and ***input format (vector -> 3-D tensor)***



CNN in Keras

Only modified the ***network structure*** and ***input format (vector -> 3-D tensor)***



Live Demo

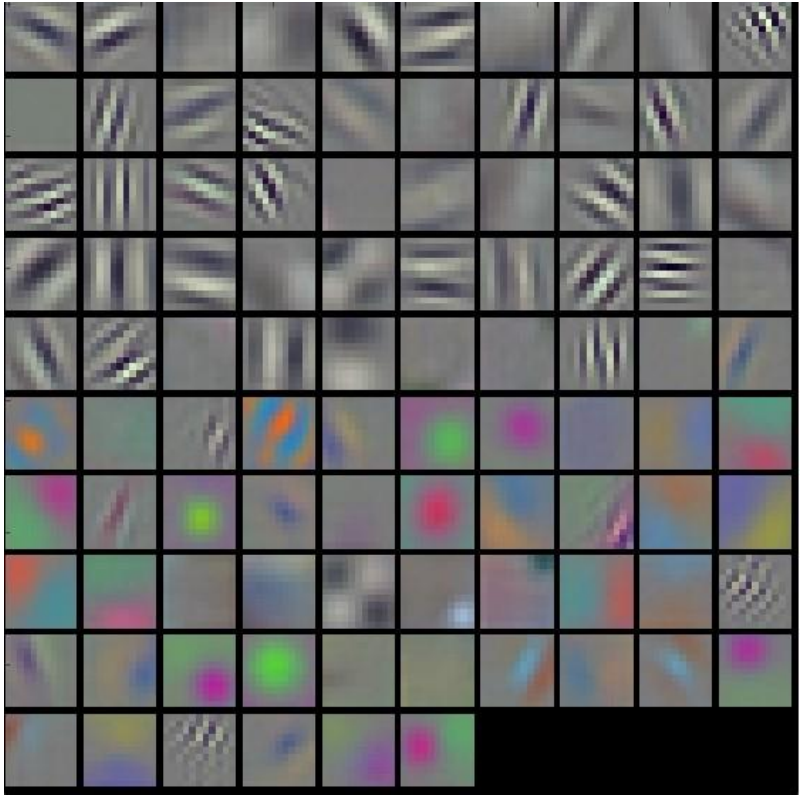
What does machine learn?

http://newsneakernews.wpengine.netdna-cdn.com/wp-content/uploads/2016/11/rihanna-puma-creeper-velvet-release-date-02.jpg

First Convolution Layer

- Typical-looking filters on the trained first layer

使用训练好的Filter反向了解似，每个Filter最想看的特征

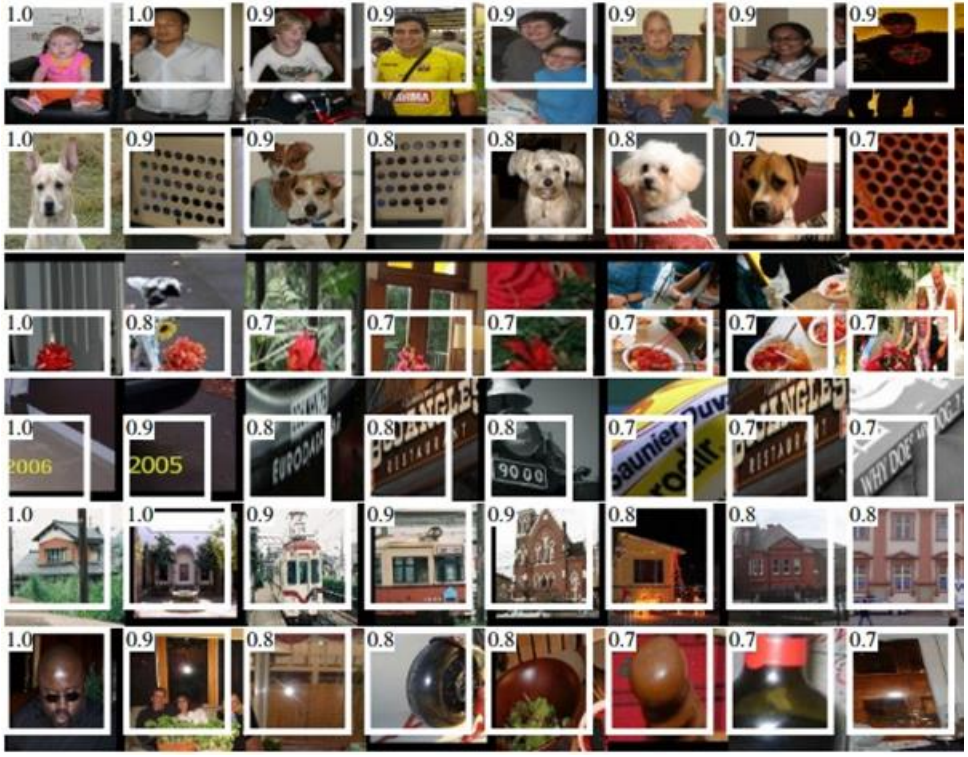


11 x 11
(AlexNet)

http://cs231n.github.io/understanding-cnn/

How about higher layers?

- Which images make a specific neuron activate



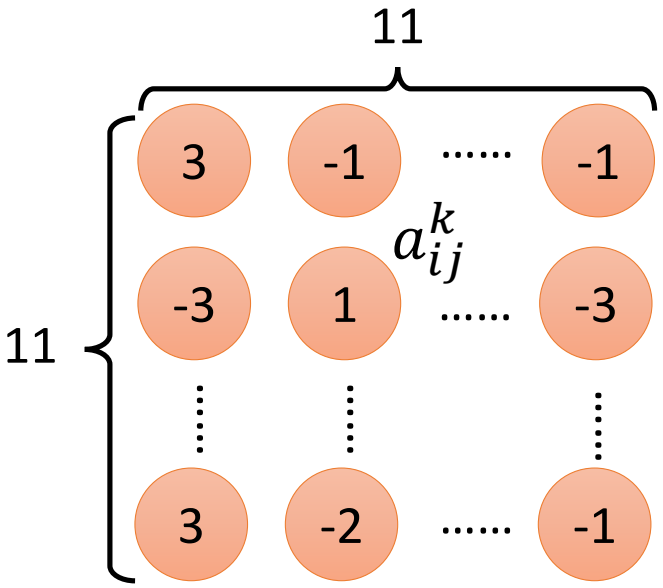
Ross Girshick, Jeff Donahue, Trevor Darrell, Jitendra Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation”, CVPR, 2014

What does CNN learn?

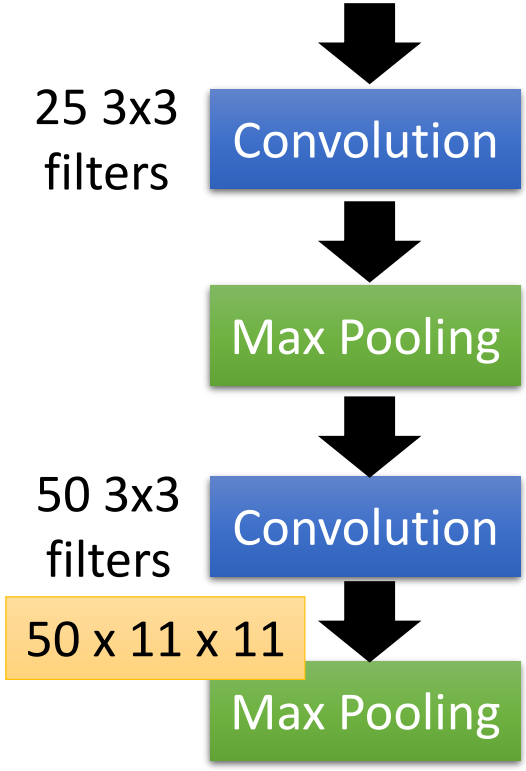
The output of the k-th filter is a 11 x 11 matrix.

Degree of the activation of the k-th filter: $a^k = \sum_{i=1}^{11} \sum_{j=1}^{11} a_{ij}^k$

$x^* = arg \max_x a^k$ (gradient ascent)



$\frac{\partial a^k}{\partial x_{ij}}$ x input

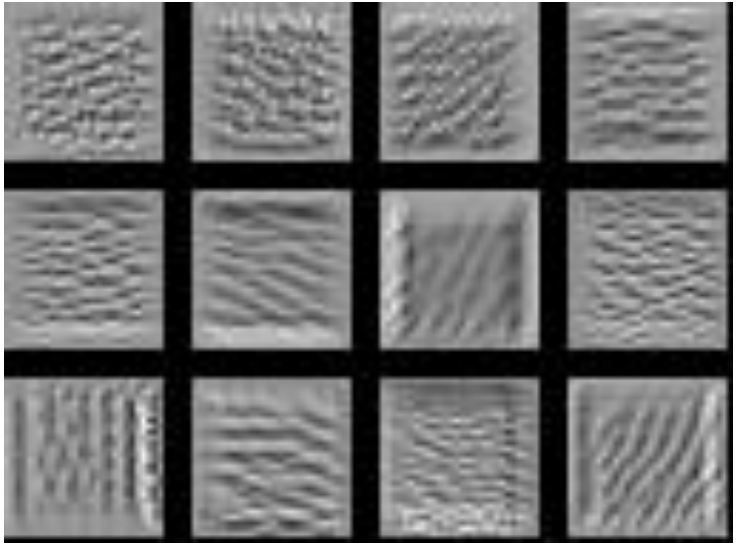


What does CNN learn?

The output of the k-th filter is a 11 x 11 matrix.

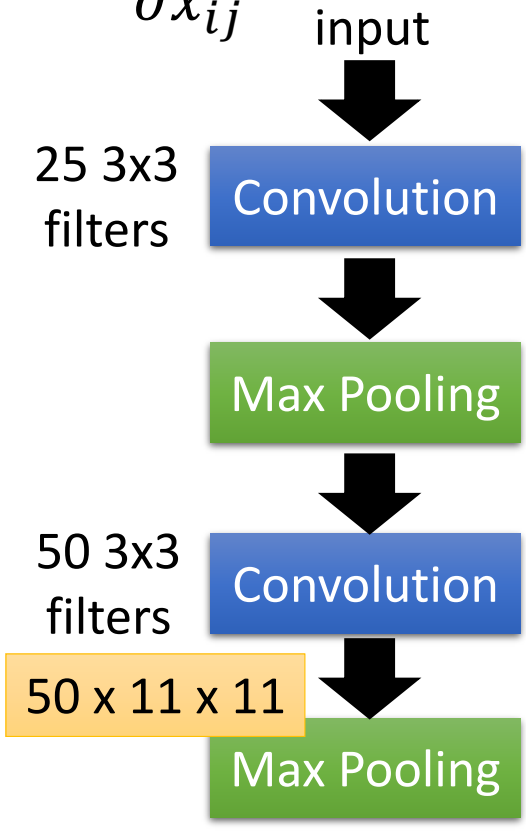
Degree of the activation of the k-th filter: $a^k = \sum_{i=1}^{11} \sum_{j=1}^{11} a_{ij}^k$

$x^* = arg \max_x a^k$ (gradient ascent)



For each filter

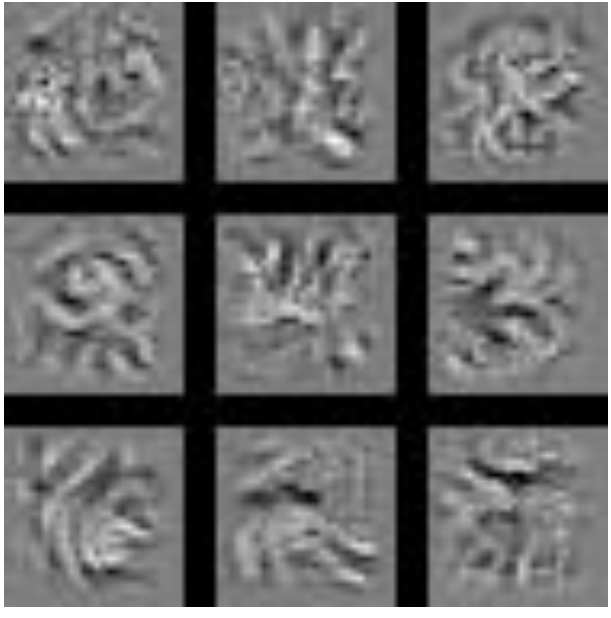
$$\frac{\partial a^k}{\partial x_{ij}}$$



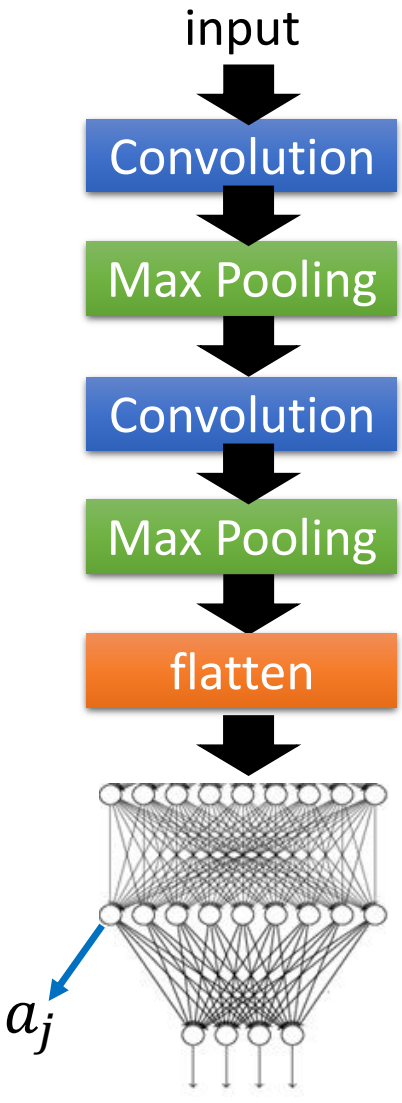
What does CNN learn?

Find an image maximizing the output of neuron:

$$x^* = arg \max_x a^j$$



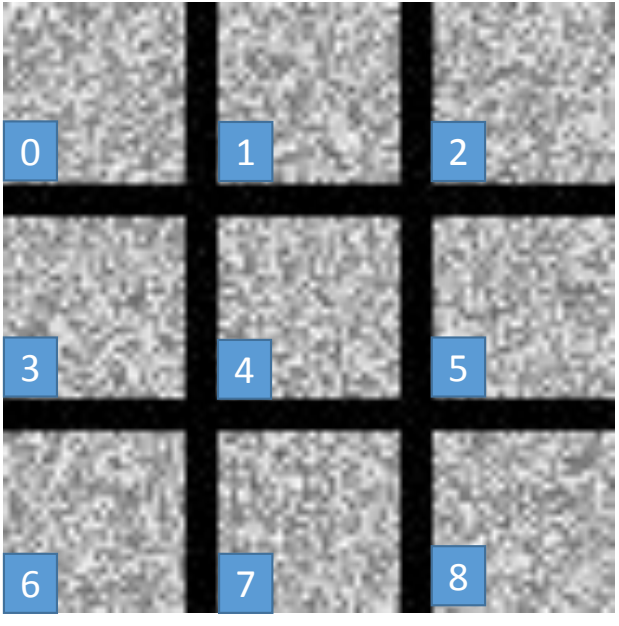
Each figure corresponds to a neuron



What does CNN learn?

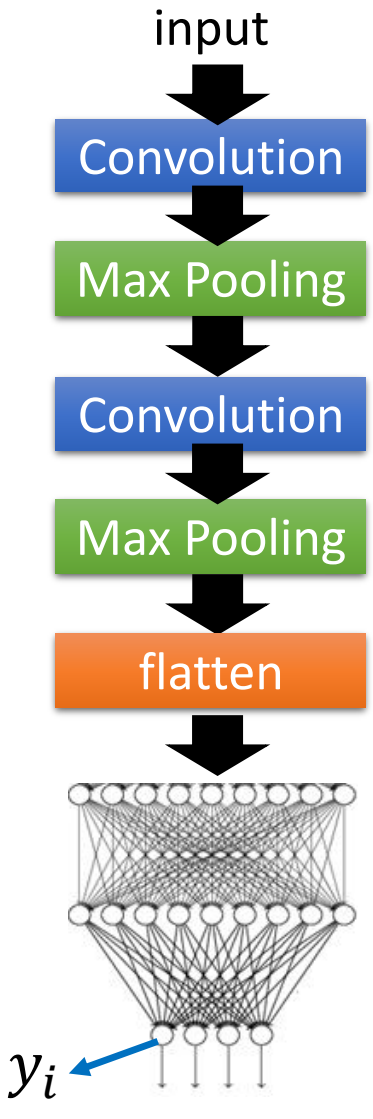
$$x^* = arg\max_x y^i$$

Can we see digits?



Deep Neural Networks are Easily Fooled

https://www.youtube.com/watch?v=M2IebCN9Ht4

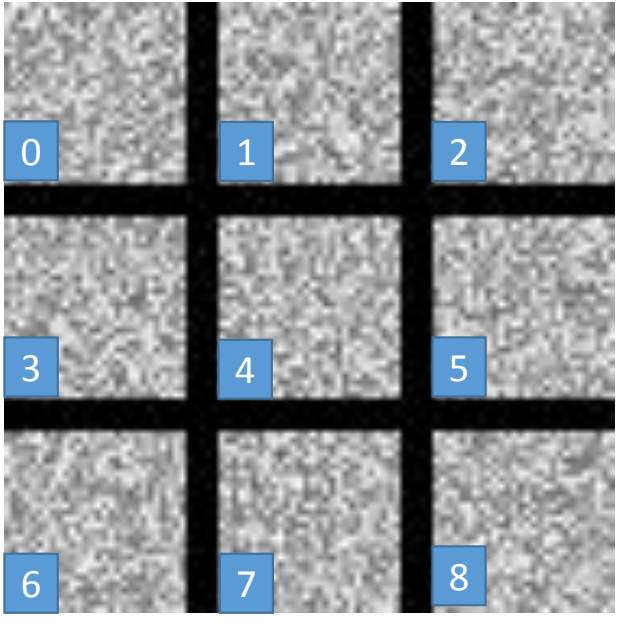


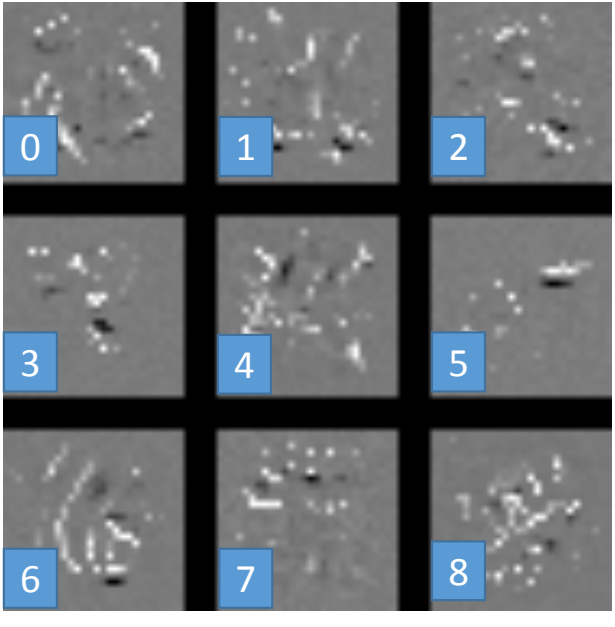
What does CNN learn?

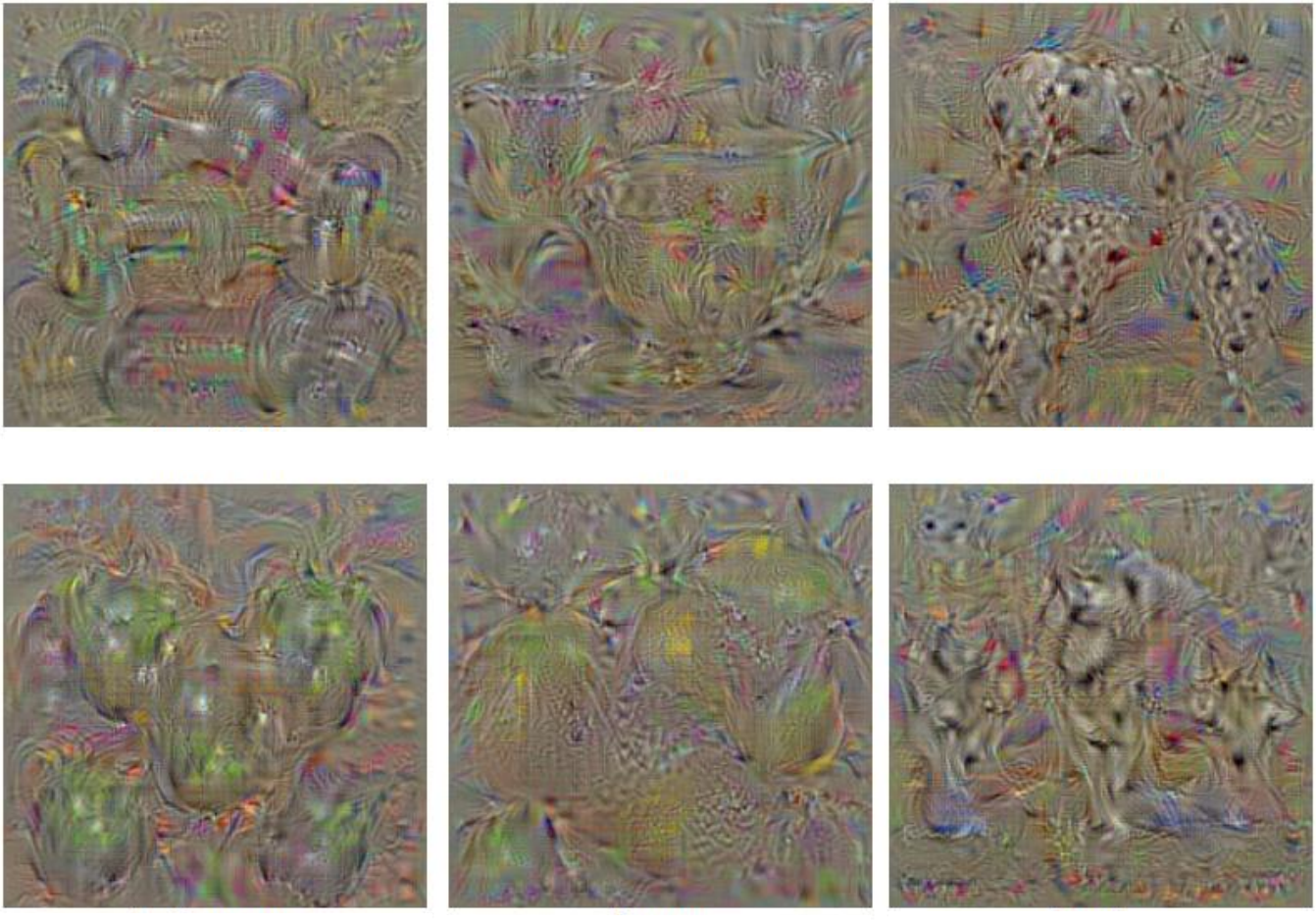
$$x^* = arg\max_x y^i$$

Over all pixel values

$$x^* = arg\max_x \left(y^i - \sum_{i,j} |x_{ij}| \right)$$







Karen Simonyan, Andrea Vedaldi, Andrew Zisserman, “Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps”, ICLR, 2014

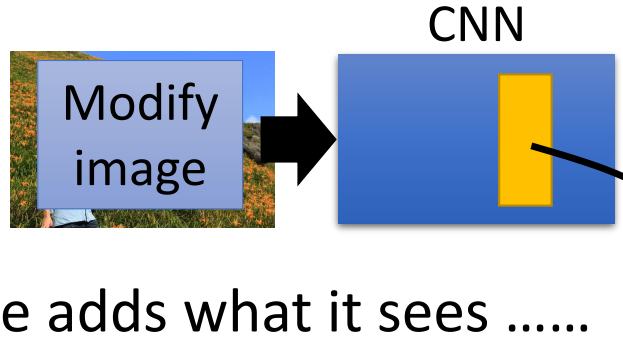
$$\left|\frac{\partial y_k}{\partial x_{ij}}\right|$$

y_k: the predicted class of the model

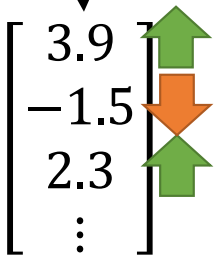
Karen Simonyan, Andrea Vedaldi, Andrew Zisserman, "Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps", ICLR, 2014

Reference: Zeiler, M. D., & Fergus, R. (2014). Visualizing and understanding convolutional networks. In *Computer Vision–ECCV 2014* (pp. 818-833)

Deep Dream



- Given a photo, machine adds what it sees ……



http://deepdreamgenerator.com/

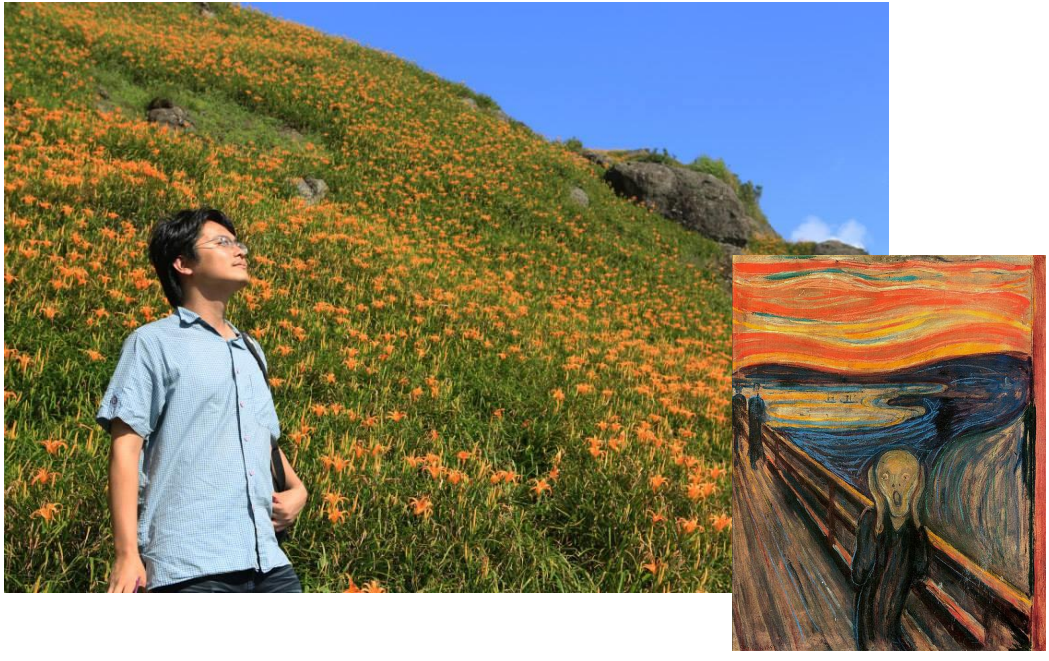
Deep Dream

- Given a photo, machine adds what it sees ……

http://deepdreamgenerator.com/

Deep Style

- Given a photo, make its style like famous paintings



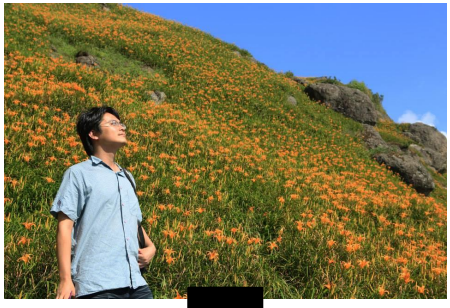
https://dreamscopeapp.com/

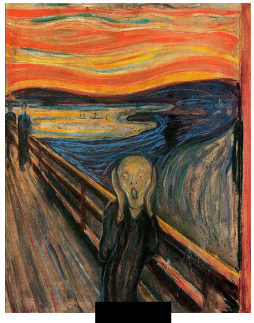
Deep Style

- Given a photo, make its style like famous paintings

https://dreamscopeapp.com/

Deep Style





CNN

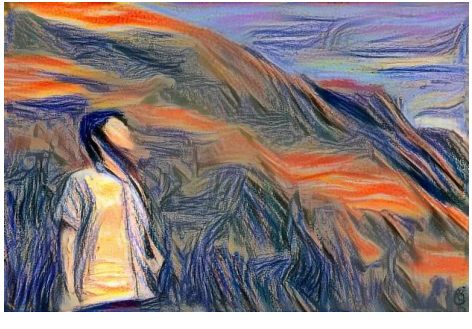
CNN

content

style

A Neural Algorithm of Artistic Style

https://arxiv.org/abs/1508.06576



CNN

?

More Application: Playing Go

19 x 19 matrix (image)

Network

Next move (19 x 19 positions)

19 x 19 vector

Black: 1

white: -1

none: 0

Fully-connected feedforward network can be used

But CNN performs much better.

More Application: Playing Go

Training: record of previous plays 黑: 5之五 → 白: 天元 → 黑: 五之5 …

CNN

CNN

Target:
“天元” = 1
else = 0

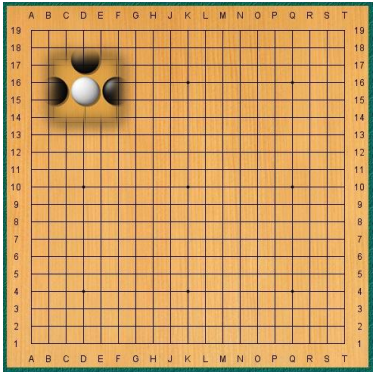
Target:
“五之 5” = 1
else = 0

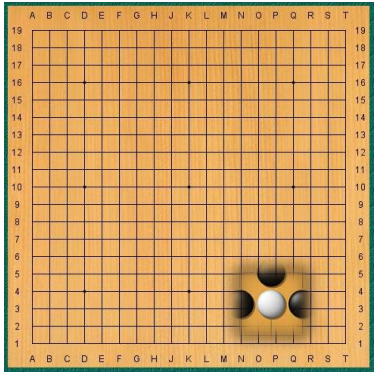
Why CNN for playing Go?

- Some patterns are much smaller than the whole image

  Alpha Go uses 5 x 5 for first layer

- The same patterns appear in different regions.





Why CNN for playing Go?

- Subsampling the pixels will not change the object

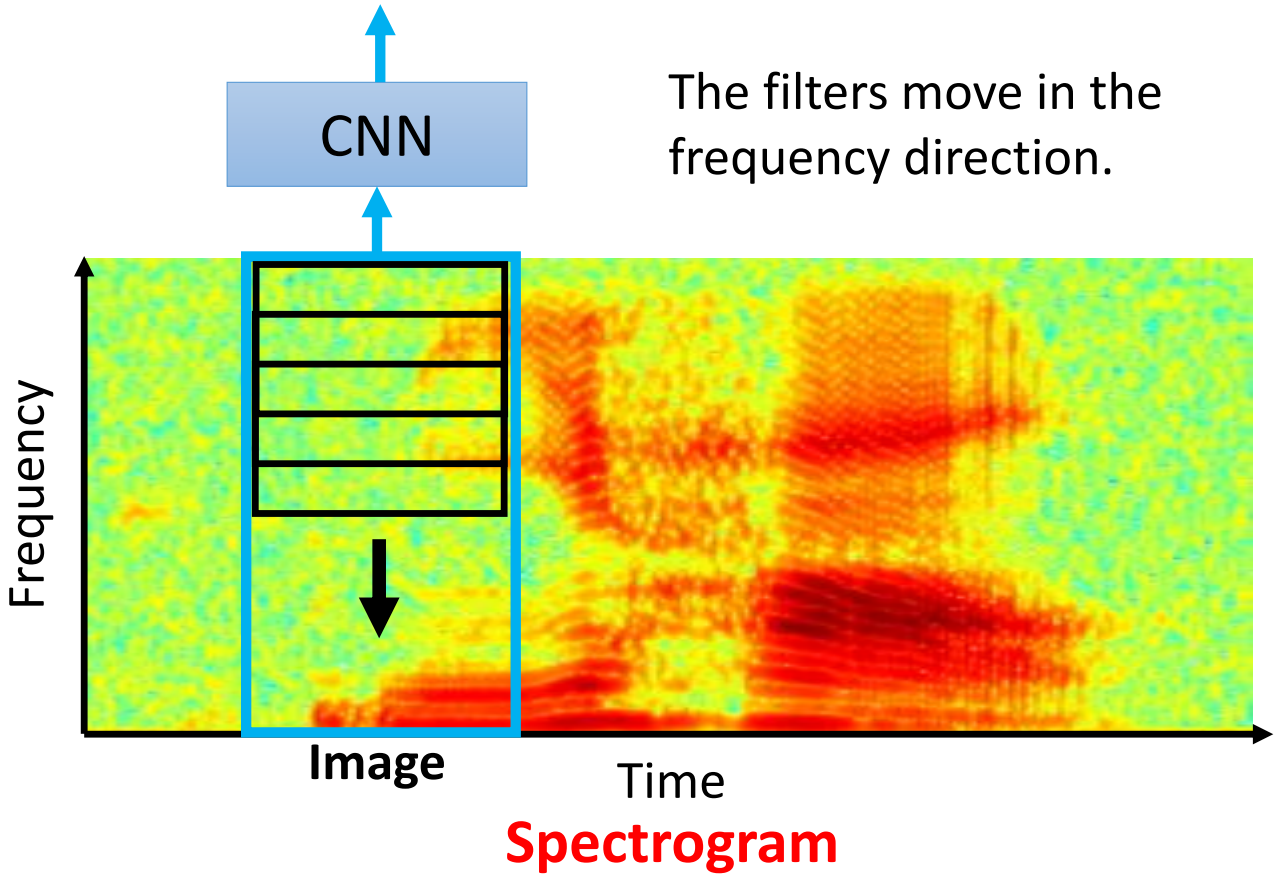
Max Pooling

How to explain this???

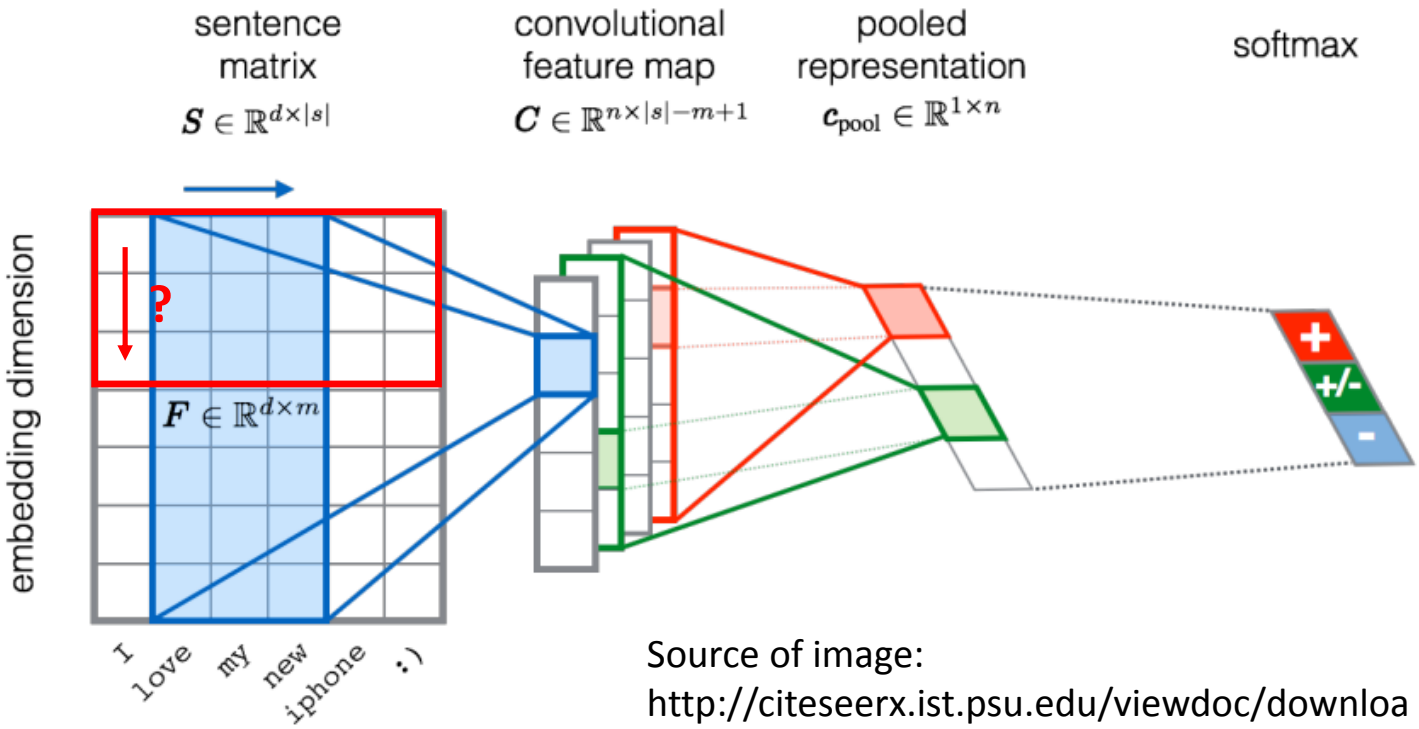
Neural network architecture. The input to the policy network is a $19 \times 19 \times 48$ image stack consisting of 48 feature planes. The first hidden layer zero pads the input into a 23×23 image, then convolves k filters of kernel size 5×5 with stride 1 with the input image and applies a rectifier nonlinearity. Each of the subsequent hidden layers 2 to 12 zero pads the respective previous hidden layer into a 21×21 image, then convolves k filters of kernel size 3×3 with stride 1, again followed by a rectifier nonlinearity. The final layer convolves 1 filter of kernel size 1×1 with stride 1, with a different bias for each position, and applies a softmax function. The Extended Data Table 3 additionally show the results of training with $k = 128$, 256 and 384 filters.

Alpha Go does not use Max Pooling ……

More Application: Speech



More Application: Text



Source of image: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.703.6858&rep=rep1&type=pdf

Acknowledgment

- 感謝 Guobiao Mo 發現投影片上的打字錯誤