Convolutional Neural Network

Hung-yi Lee

CNN常被用在影像處理上，目的在簡化neuron network的架構

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

Why CNN for Image

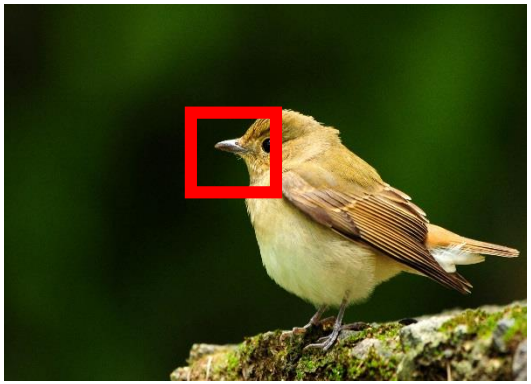
- Some patterns are much smaller than the whole image

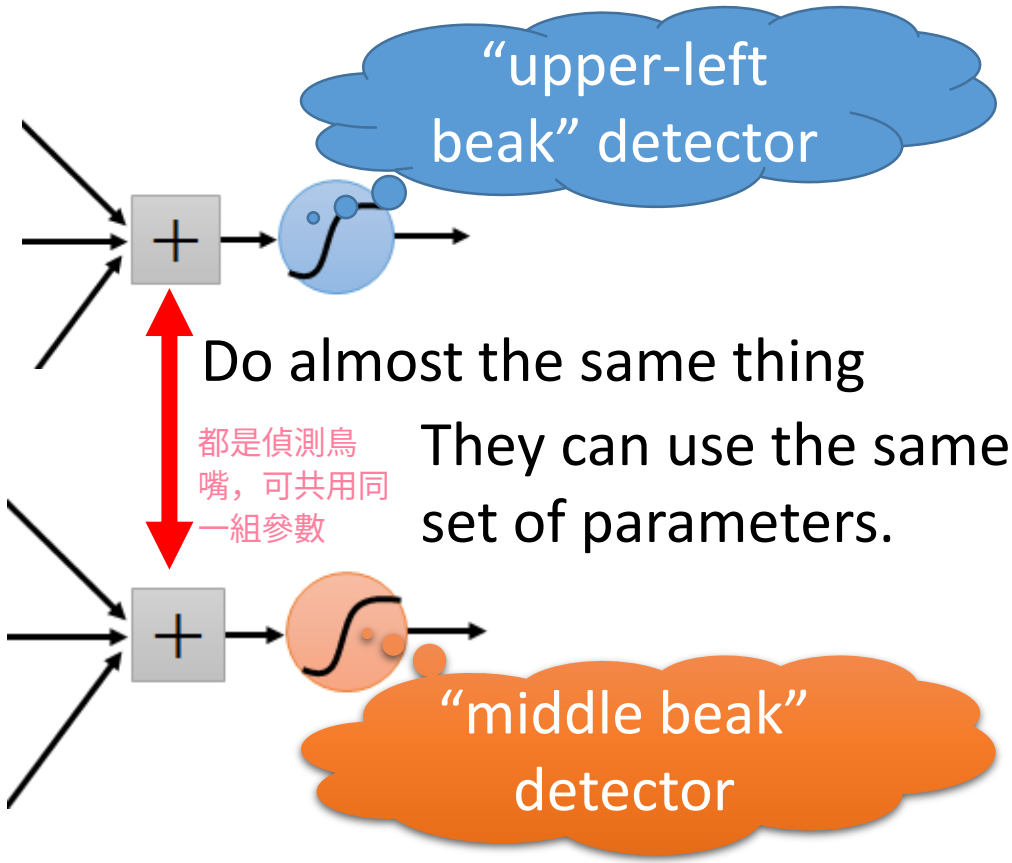
A neuron does not have to see the whole image to discover the pattern. 只需要看一小部分就可决定，不用看全部

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

把圖像變小並沒有影響理解

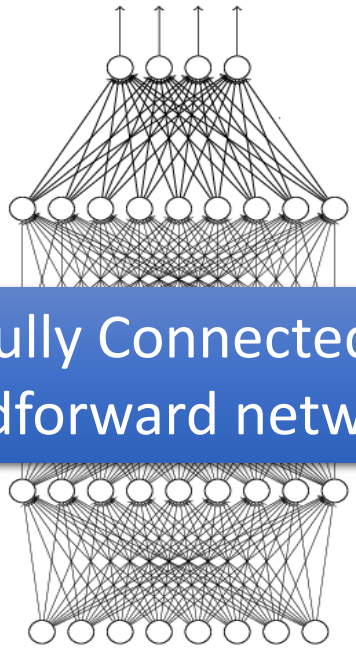
bird

We can subsample the pixels to make image smaller

→ Less parameters for the network to process the image

The whole CNN

cat dog ……



Fully Connected Feedforward network

Convolution

Max Pooling

Convolution

Max Pooling

Can repeat many times

Flatten

The whole CNN

Property 1

- Some patterns are much smaller than the whole image

Property 2

- The same patterns appear in different regions.

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

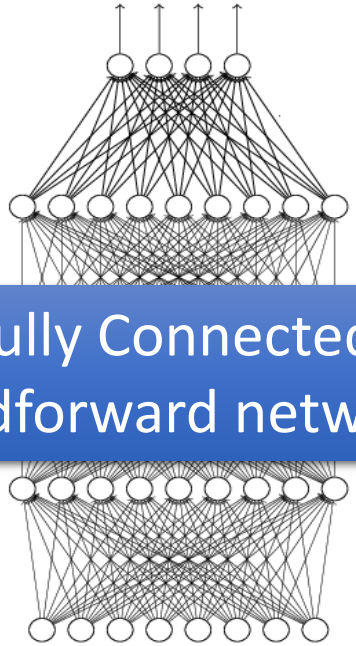
Max Pooling

Convolution

Max Pooling

Can repeat many times

Flatten



Fully Connected Feedforward network

cat dog ……

CNN – Convolution

Those are the network parameters to be learned.

要做什麼是自動學出來的

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

每一個filter裡的每一個elements=neuron的參數

-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

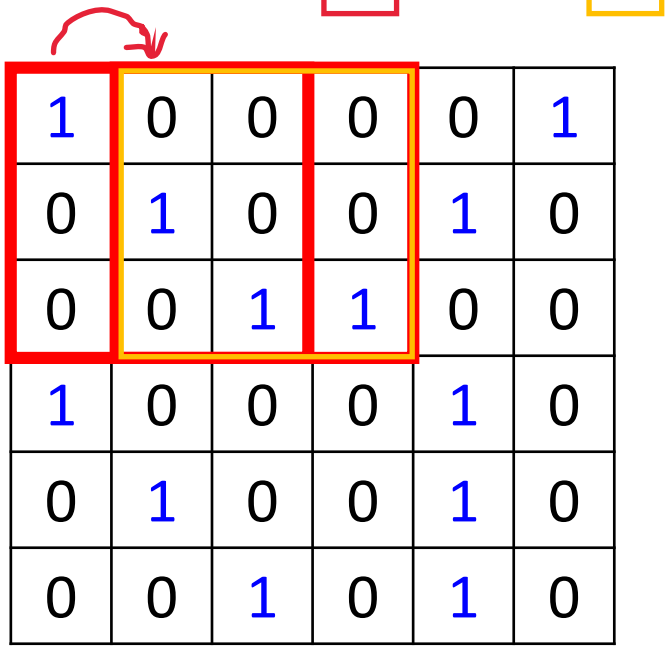
1	-1	-1
-1	1	-1
-1	-1	1

Filter 1

挪動的距離(人自己設)

stride=1

一格



6 x 6 image

filter放左上角，與 內積得3

3

-1

filter和 內積

CNN – Convolution

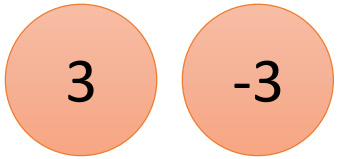
1	-1	-1
-1	1	-1
-1	-1	1

Filter 1

If stride=2

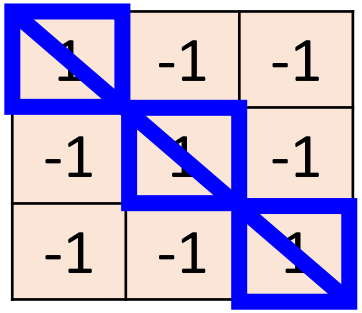
1	0	0	0	0	1
0	1	0	0	1	0
0	0	1	1	0	0
1	0	0	0	1	0
0	1	0	0	1	0
0	0	1	0	1	0

6 x 6 image



We set stride=1 below

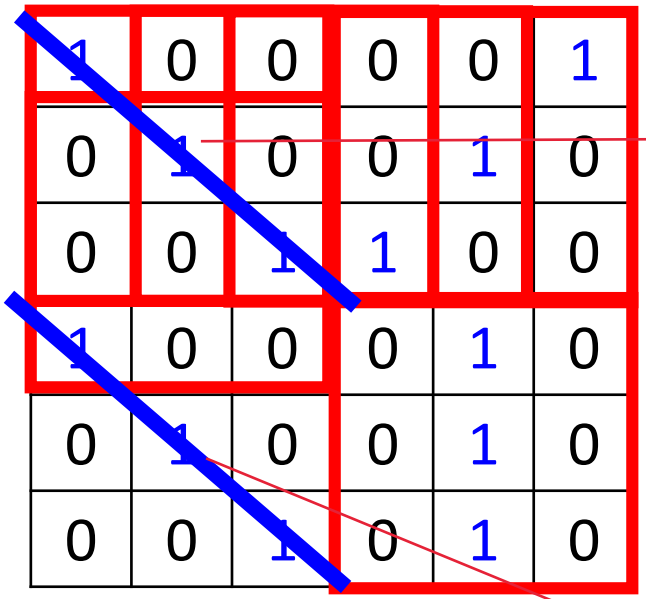
CNN – Convolution



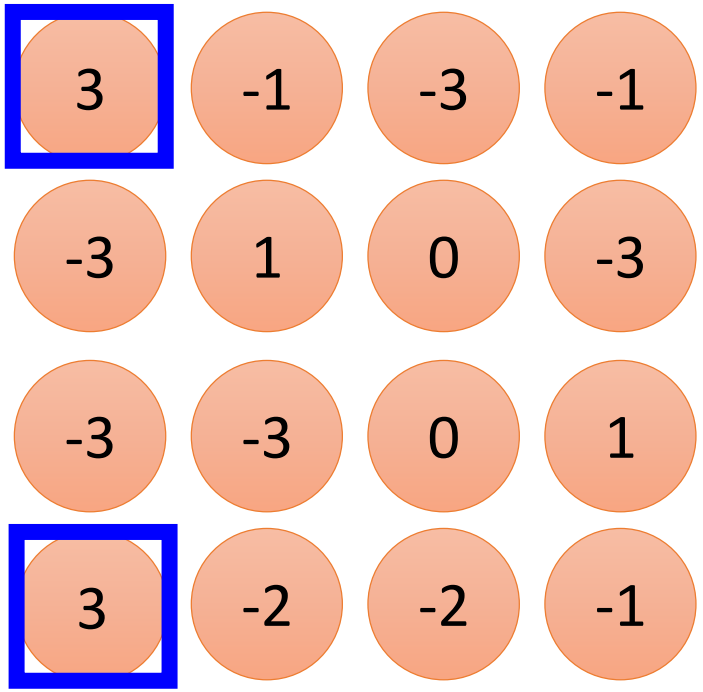
Filter 1

dectect有沒有左上到右下斜的1.1.1

stride=1



6 x 6 image



原本6*6的矩陣經過convolution process後變成4*4的矩陣

Property 2

CNN – Convolution

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

Filter 2

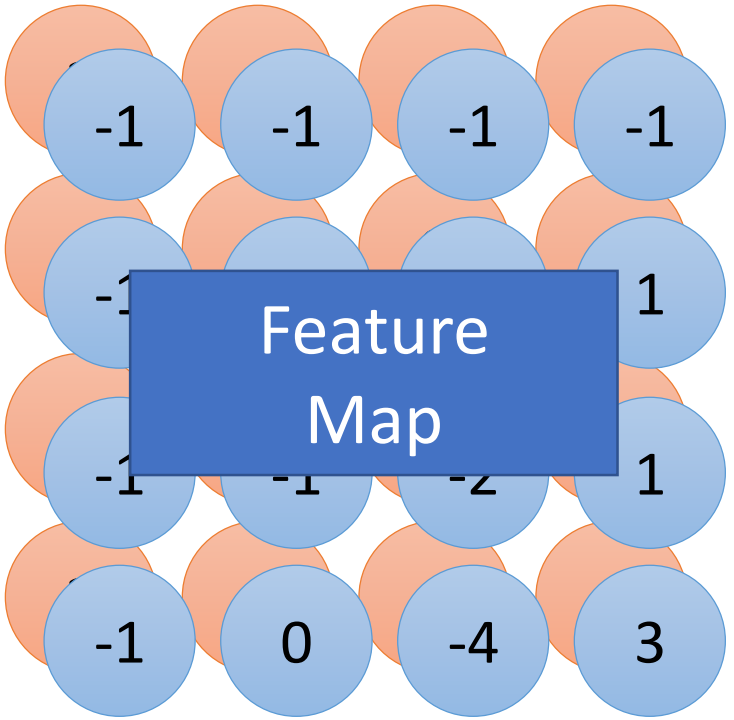
stride=1

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

紅藍矩陣加起來=feature map

Do the same process for every filter

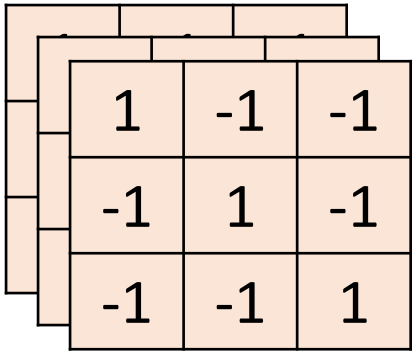


4 x 4 image

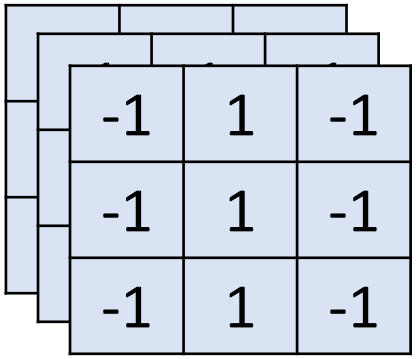
幾個filter就得到幾個image

CNN – Colorful image

彩色是立方體: 3*3*3

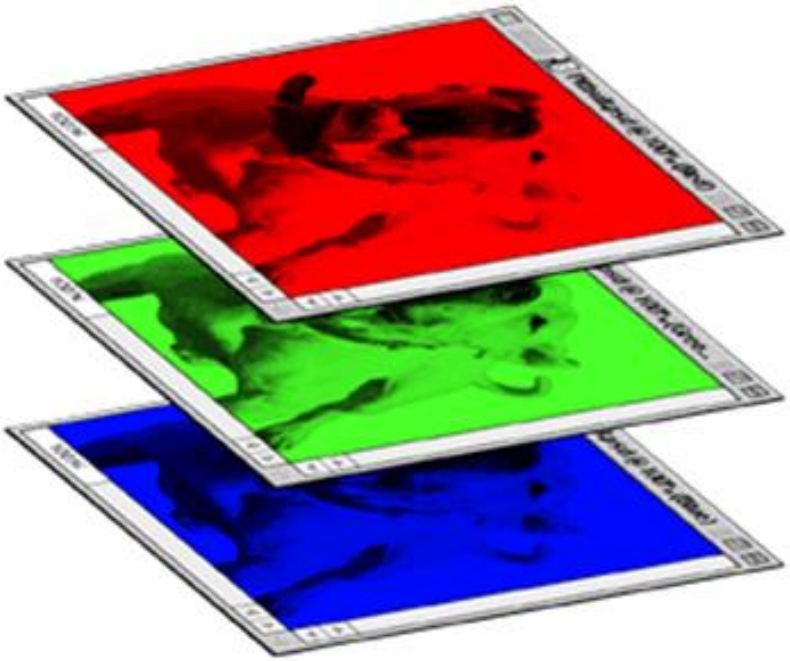


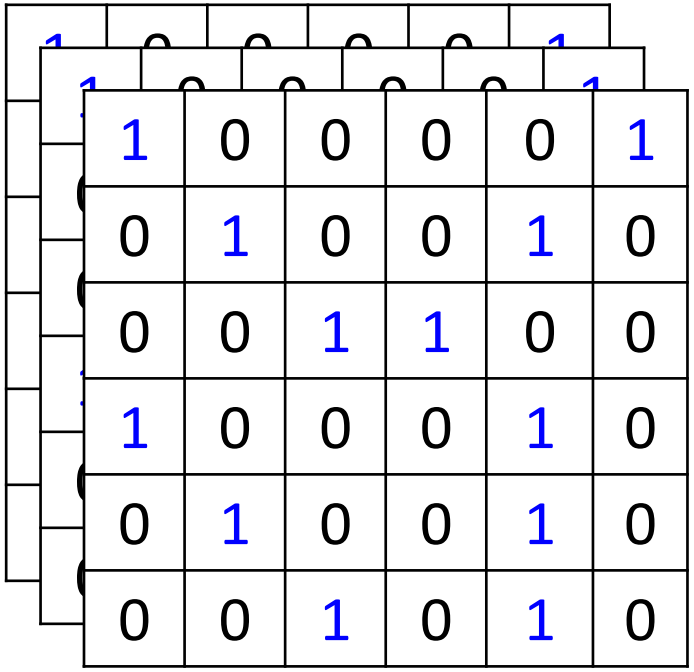
Filter 1



Filter 2

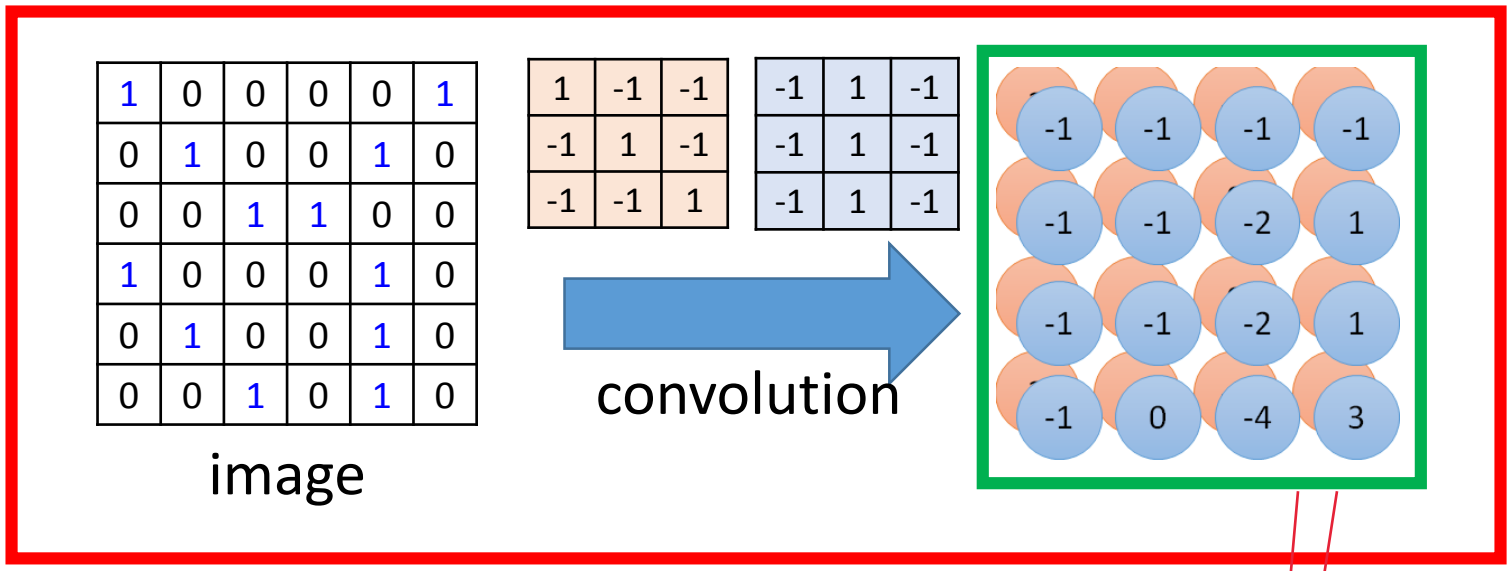
Colorful image



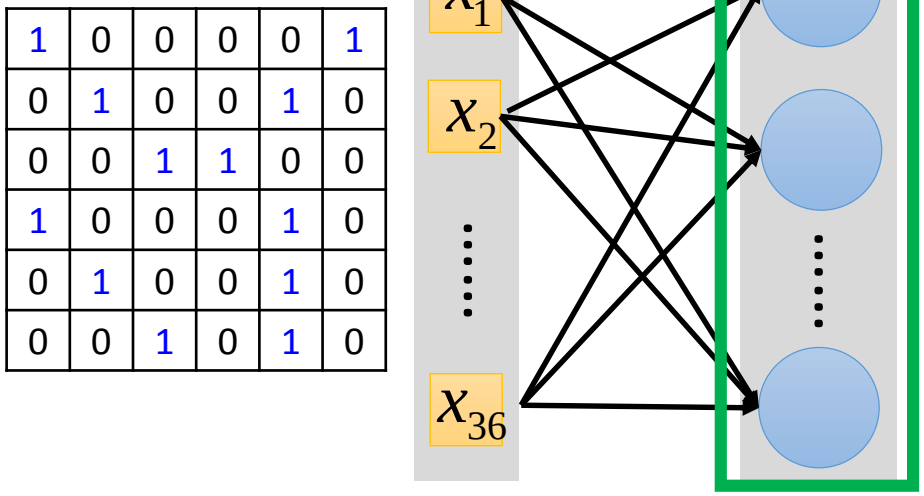


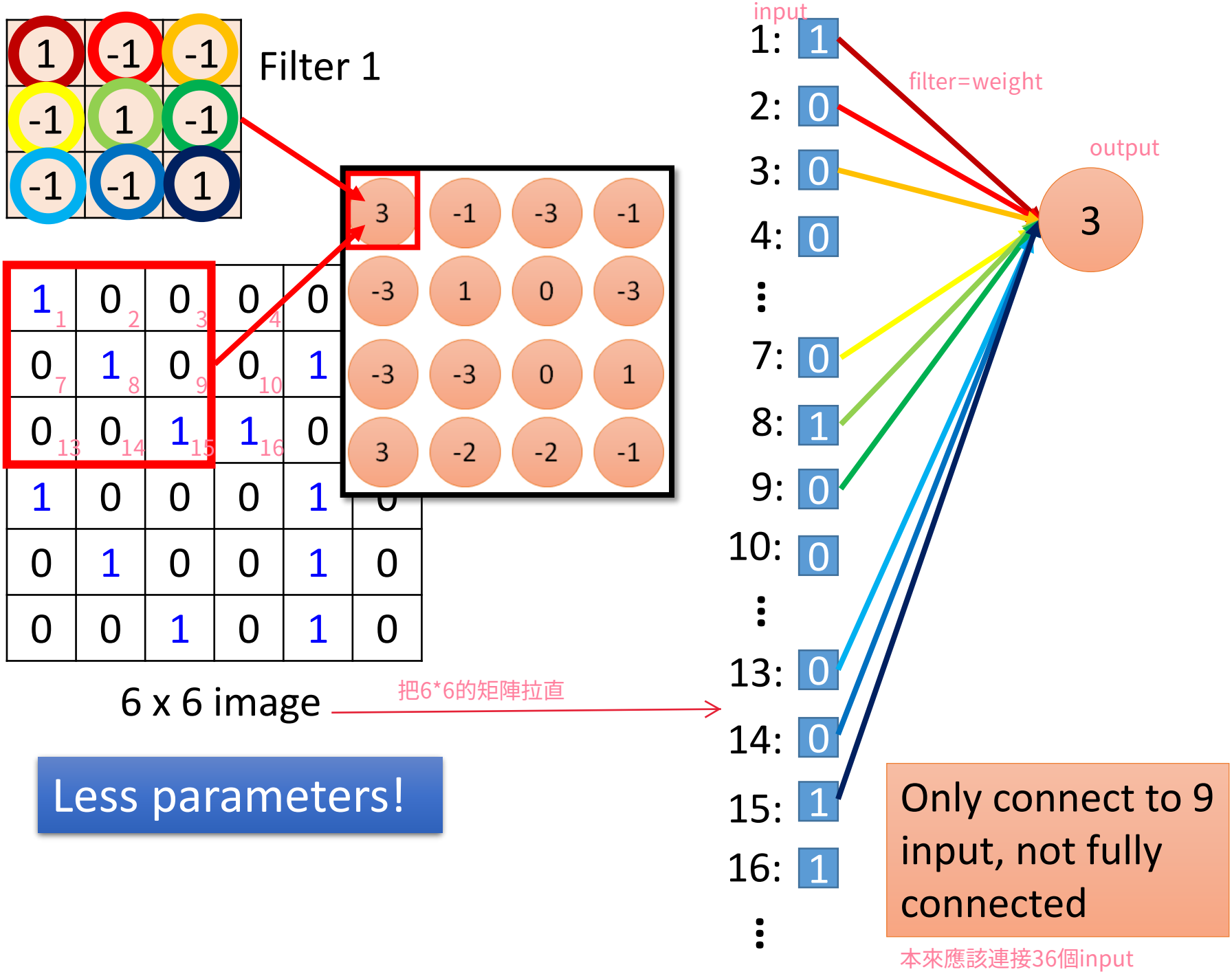
Convolution v.s. Fully Connected

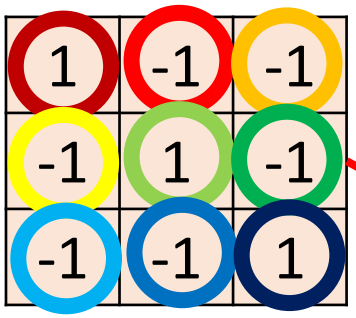
Convolution等於Fully Connected的layer把一些weight拿掉



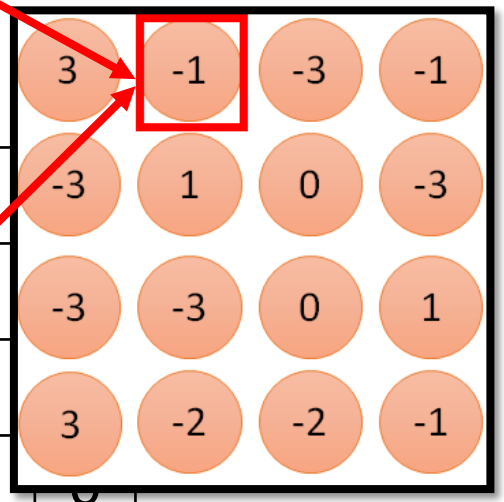
Fully-connected




1 -1 -1
-1 1 -1
-1 -1 1
Filter 1
3 -1 -3 -1
-3 1 0 -3
-3 -3 0 1
3 -2 -2 -1
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
把6*6的矩陣拉直
input
1: 1
2: 0
3: 0
4: 0
7: 0
8: 1
9: 0
10: 0
13: 0
14: 0
15: 1
16: 1
filter=weight
output
3
Less parameters!
Only connect to 9 input, not fully connected
本來應該連接36個input



Filter 1

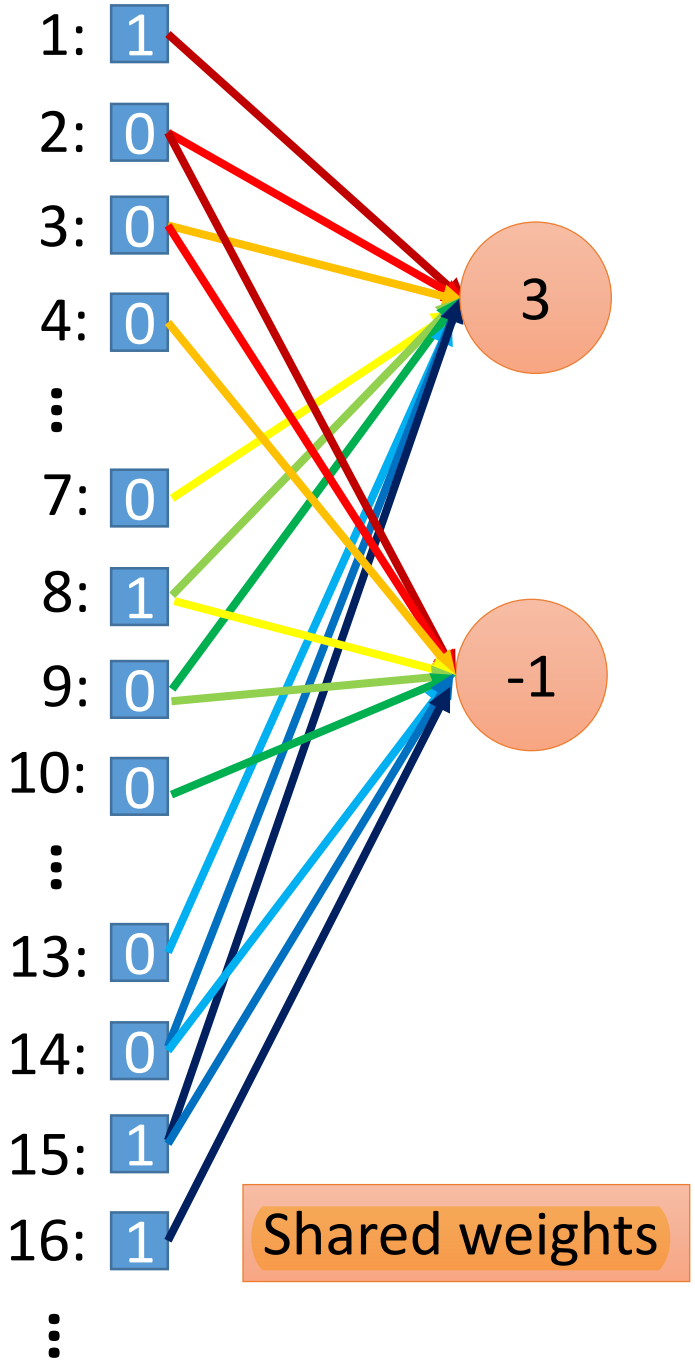


1	0	0	0	0	
0	1	0	0	1	
0	0	1	1	0	
1	0	0	0	1	
0	1	0	0	1	0
0	0	1	0	1	0

6 x 6 image

Less parameters!

Even less parameters!



Shared weights

The whole CNN

Convolution

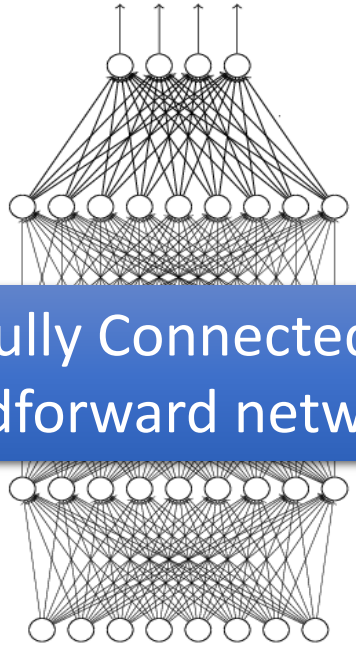
Max Pooling

Convolution

Max Pooling

Can repeat many times

Flatten



Fully Connected Feedforward network

cat dog ……

CNN – Max Pooling

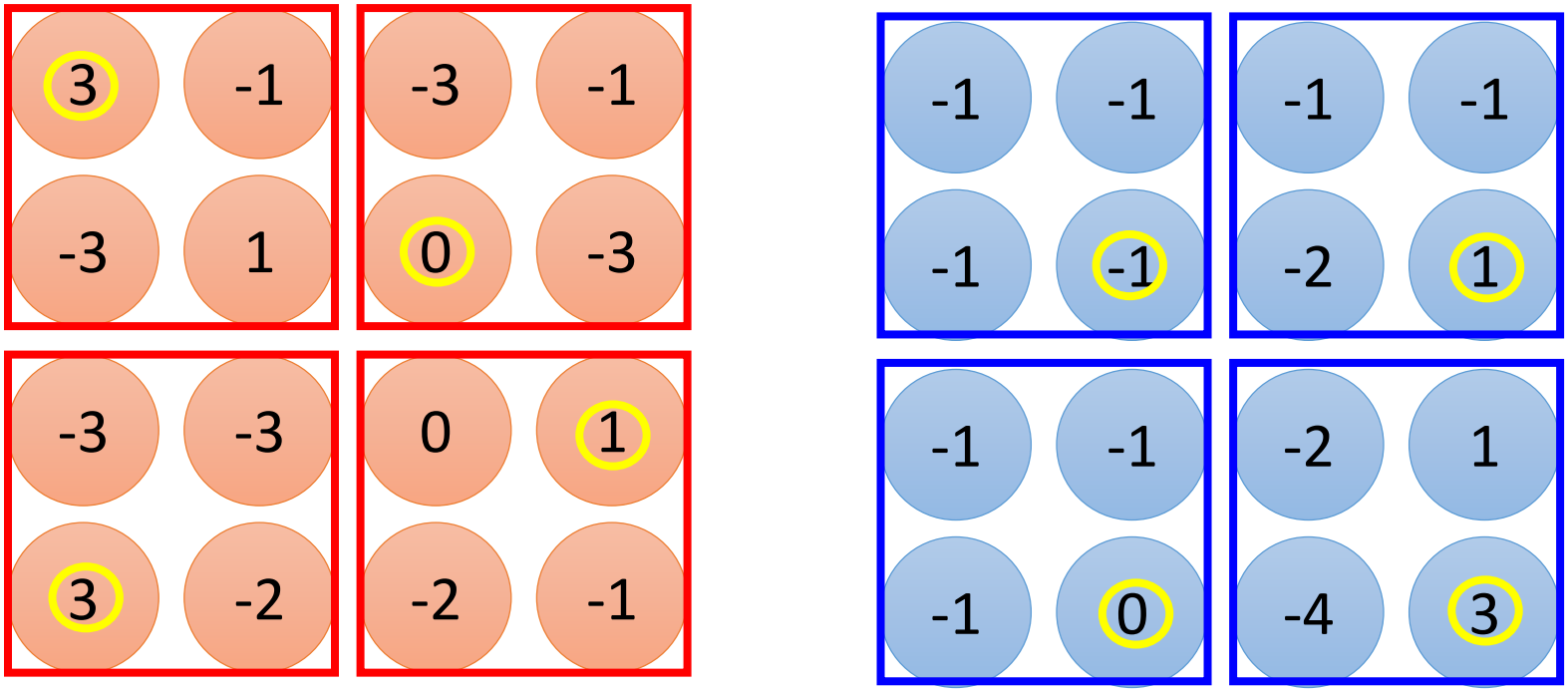
1	-1	-1
-1	1	-1
-1	-1	1

Filter 1

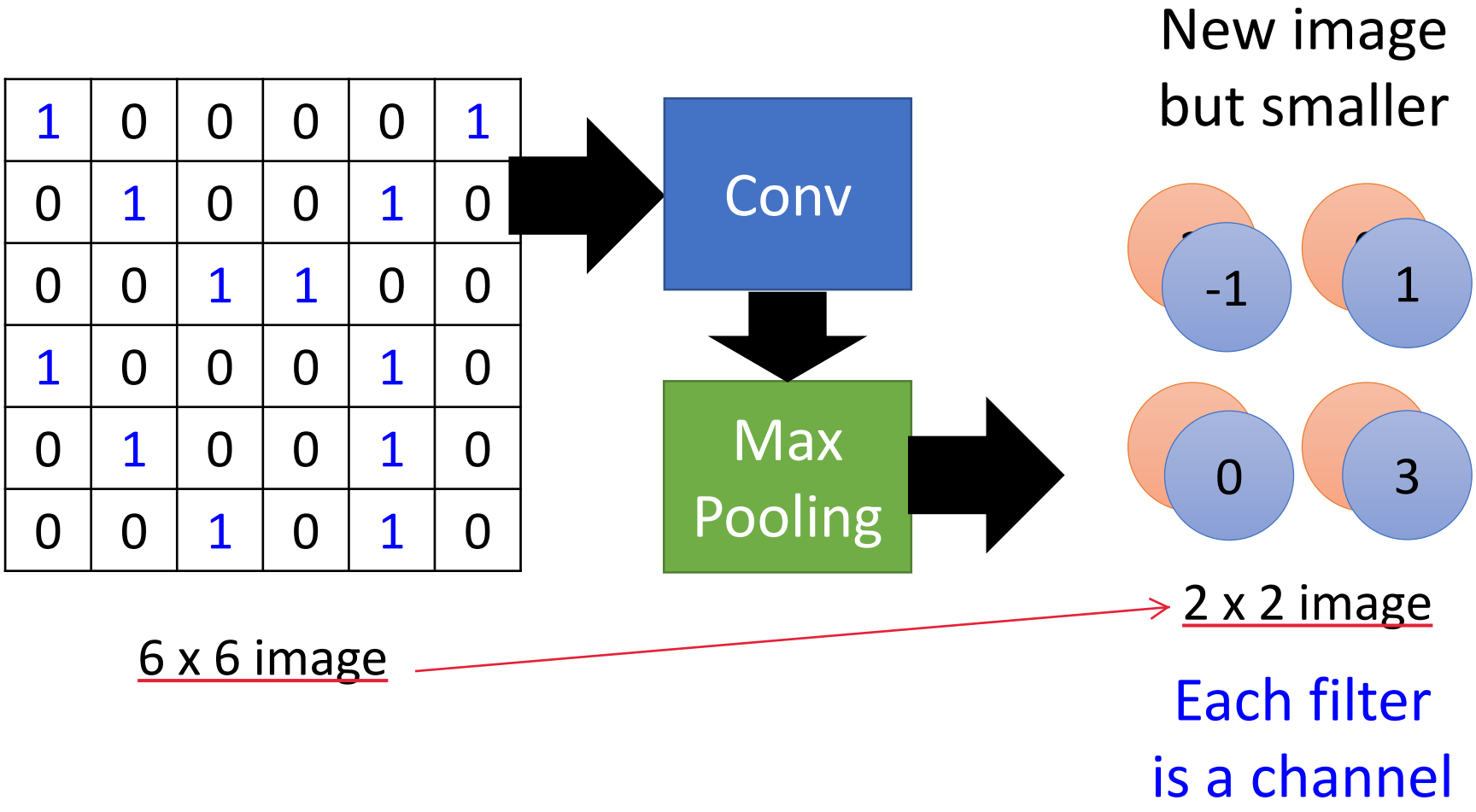
-1	1	-1
-1	1	-1
-1	1	-1

Filter 2

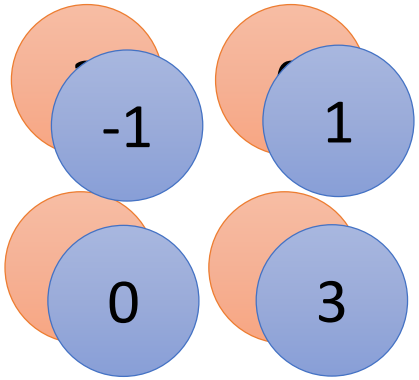
把最大的保留下來



CNN – Max Pooling



The whole CNN



A new image

Convolution

Max Pooling

Convolution

Max Pooling

Can repeat many times

Smaller than the original image

The number of the channel is the number of filters

The whole CNN

cat dog ……

Convolution

Max Pooling

A new image

Convolution

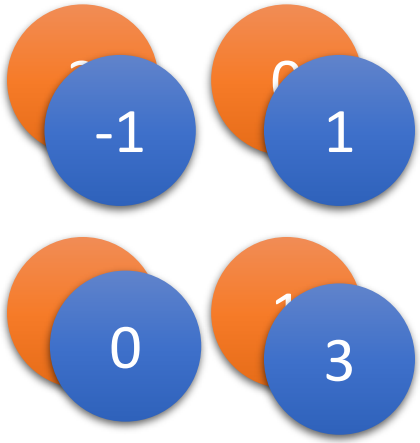
Max Pooling

A new image

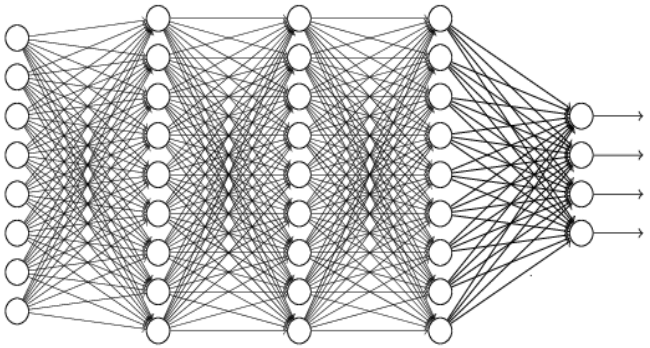
Flatten

Fully Connected Feedforward network

Flatten



Flatten



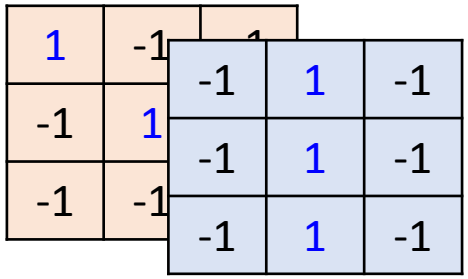
Fully Connected
Feedforward network

CNN in Keras

Only modified the ***network structure*** and ***input format (vector -> 3-D tensor)*** 長*寬*色彩(三維)

幾個filter　filter是幾乘幾的矩陣

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



……

There are 25 3x3 filters.

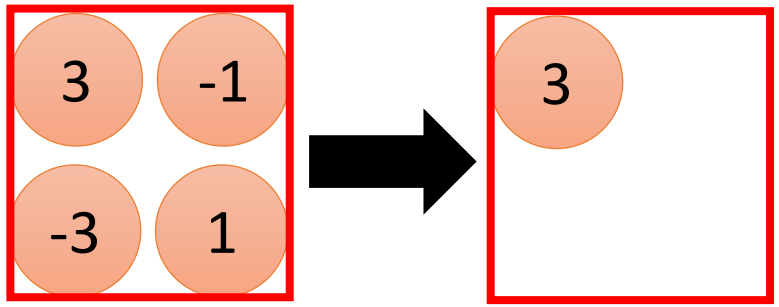
Input_shape = (28 , 28 , 1)

28 x 28 pixels　1: black/white, 3: RGB

黑白　彩色

2*2的矩陣

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



input

Convolution

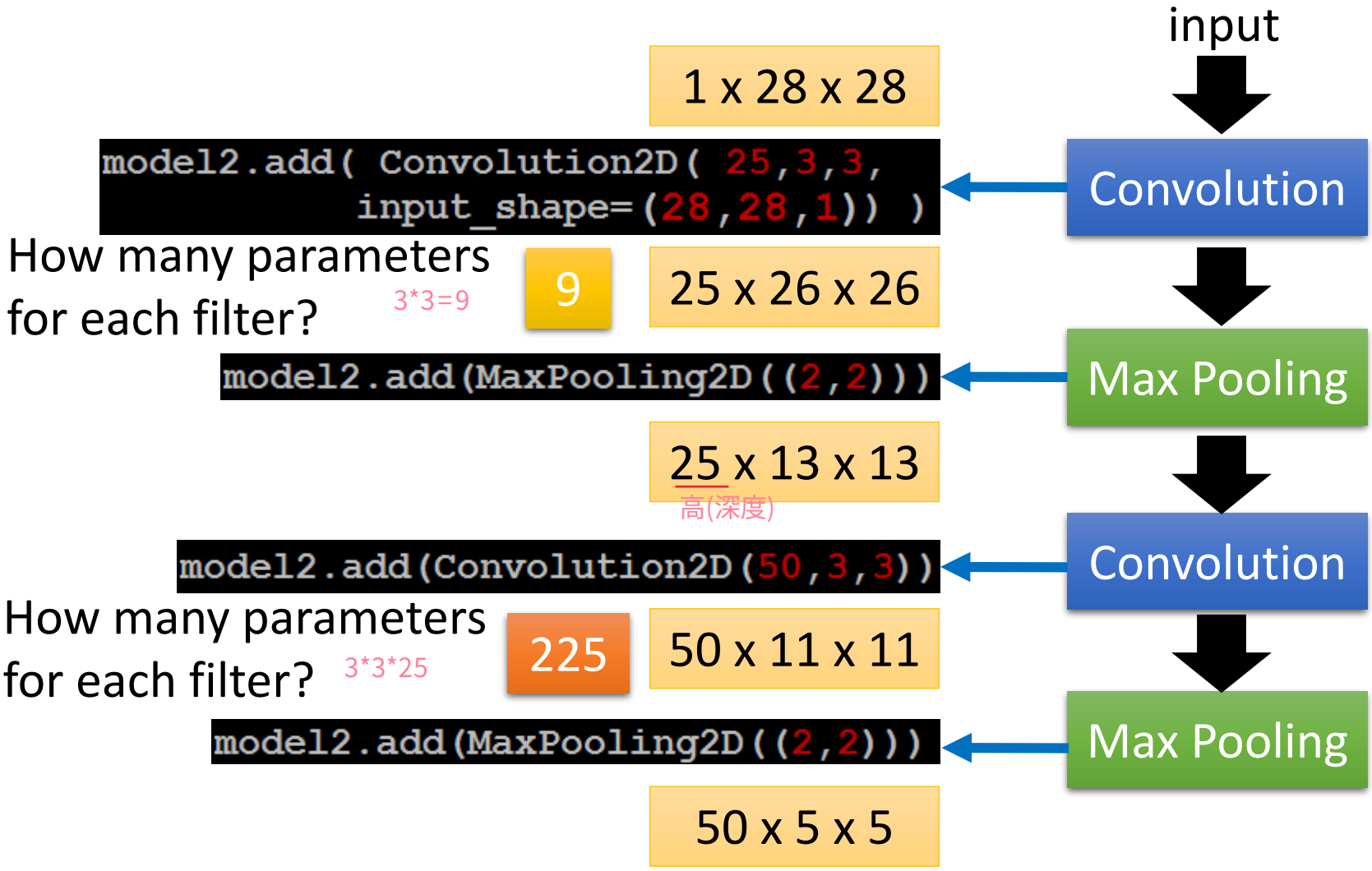
Max Pooling

Convolution

Max Pooling

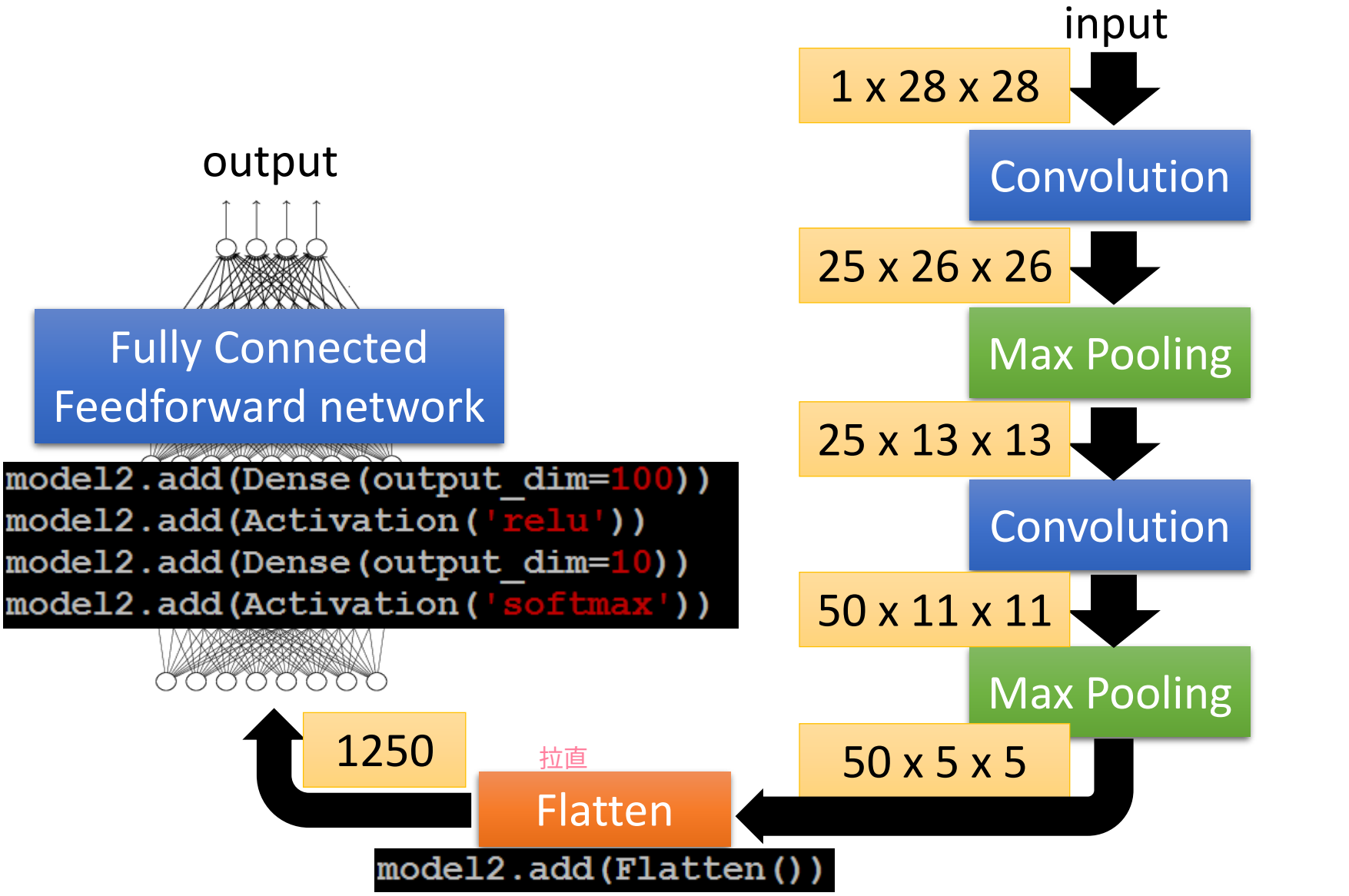
CNN in Keras

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)***



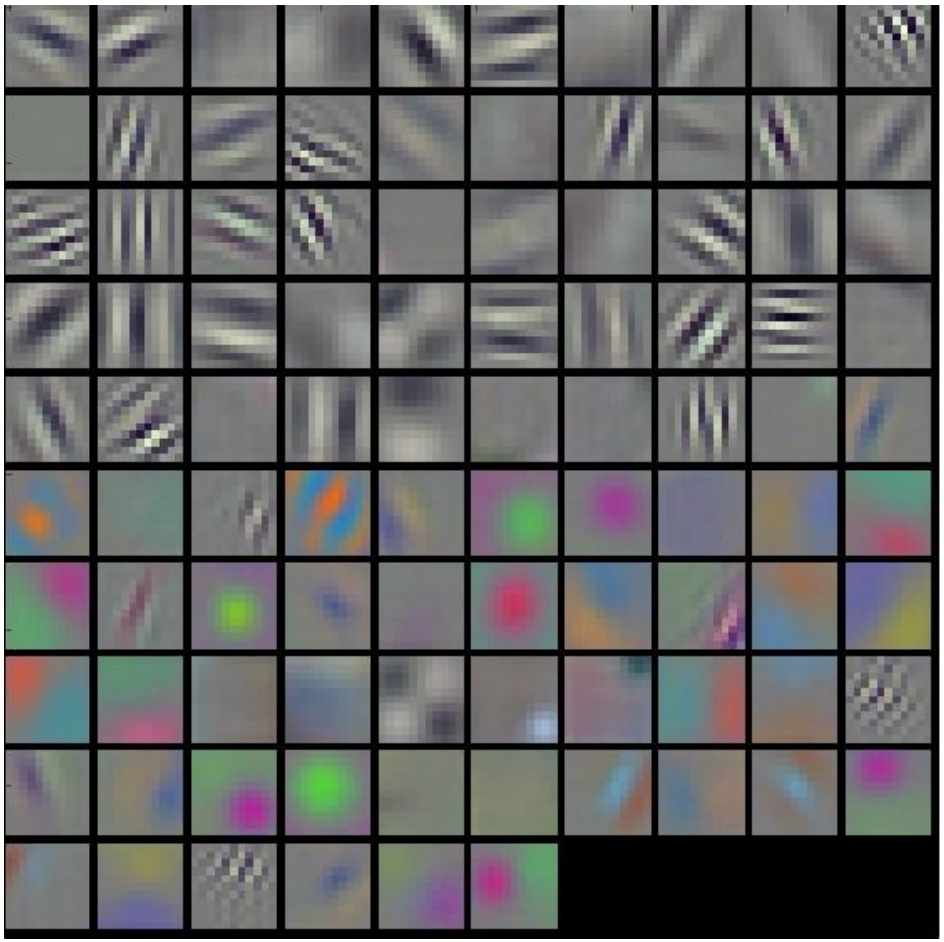
Live Demo

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

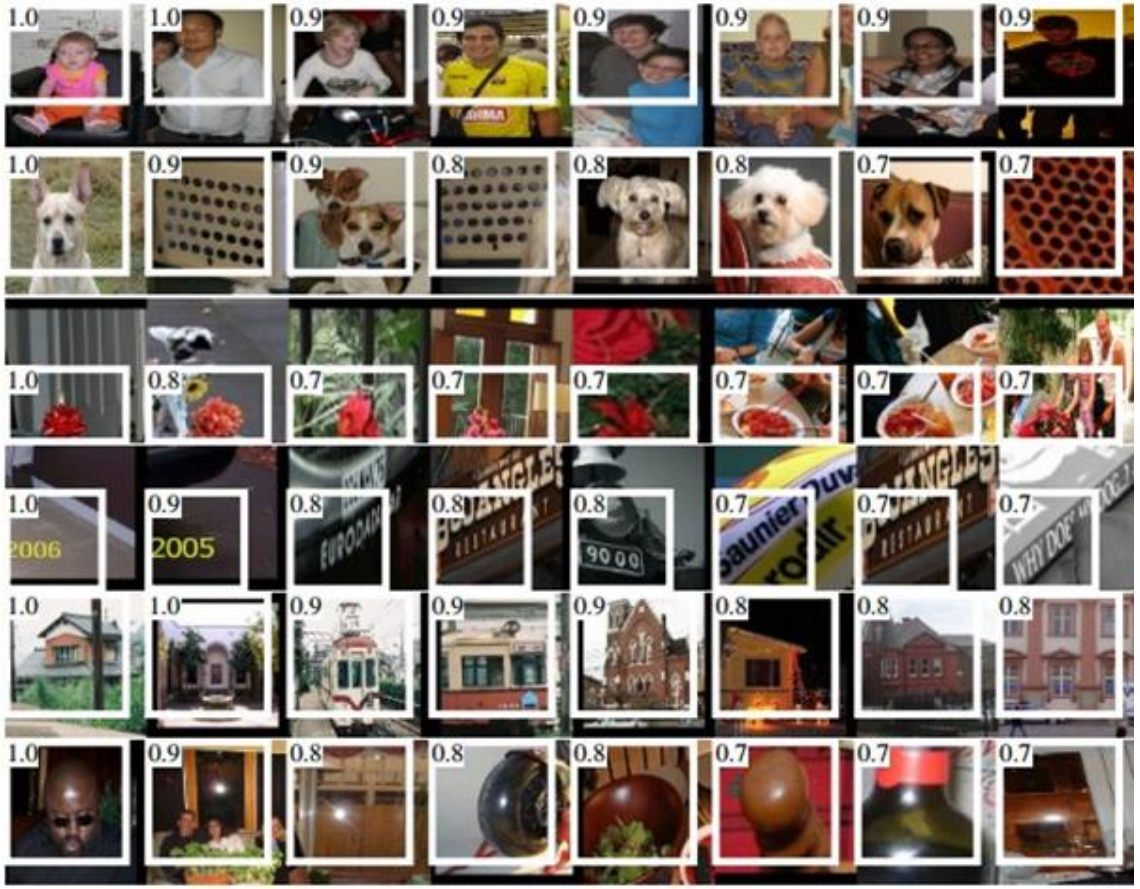


11 x 11
(AlexNet)

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

How about higher layers?

- Which images make a specific neuron activate



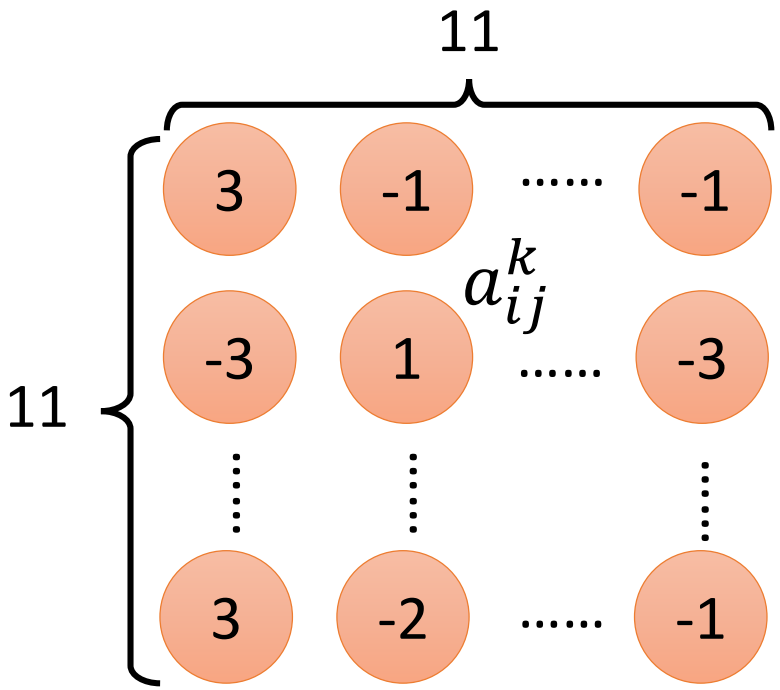
Ross Girshick, Jeff Donahue, Trevor Darrell, Jitendra Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation”, CVPR, 2014

What does CNN learn?

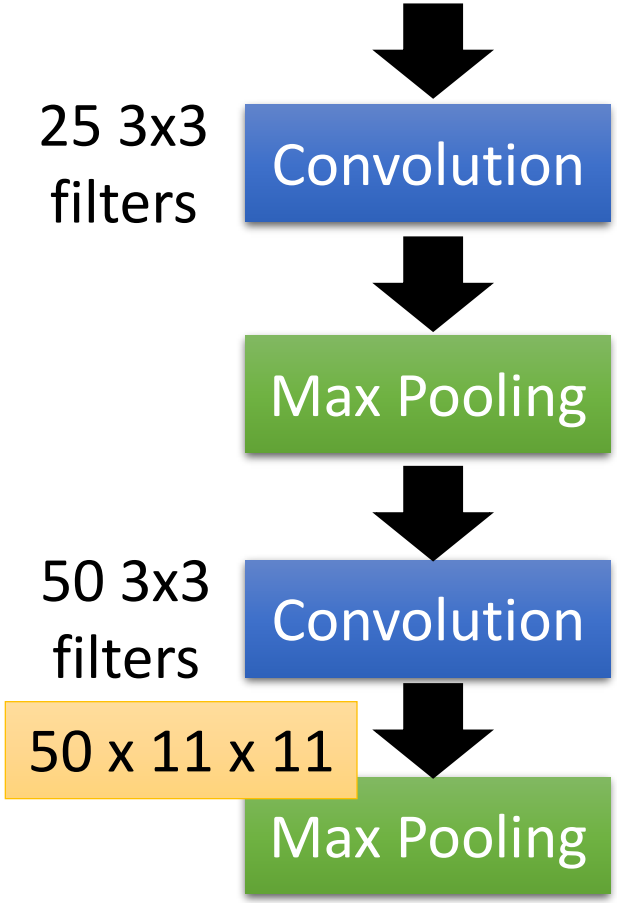
The output of the k-th filter is a 11 x 11 matrix.

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

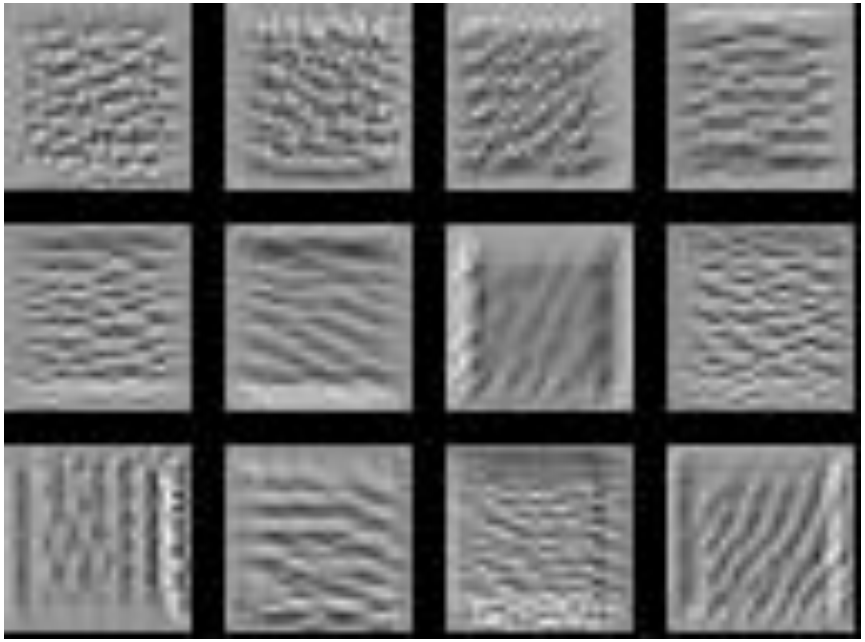


What does CNN learn?

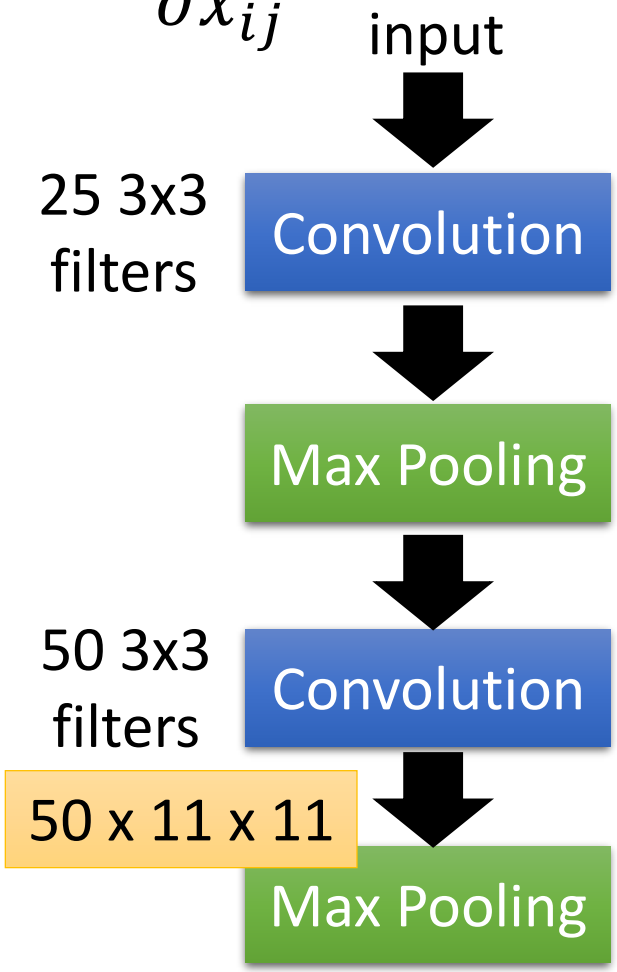
The output of the k-th filter is a 11 x 11 matrix.

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)



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

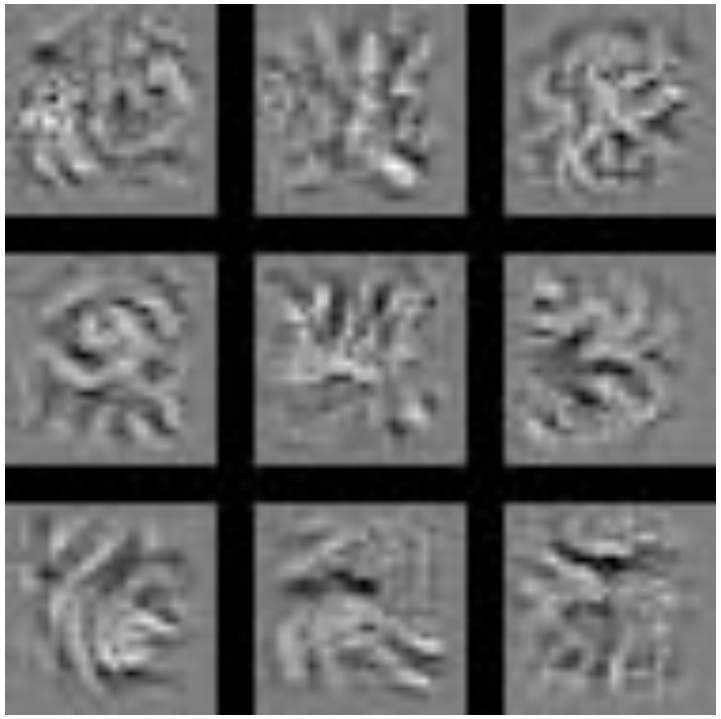


For each filter

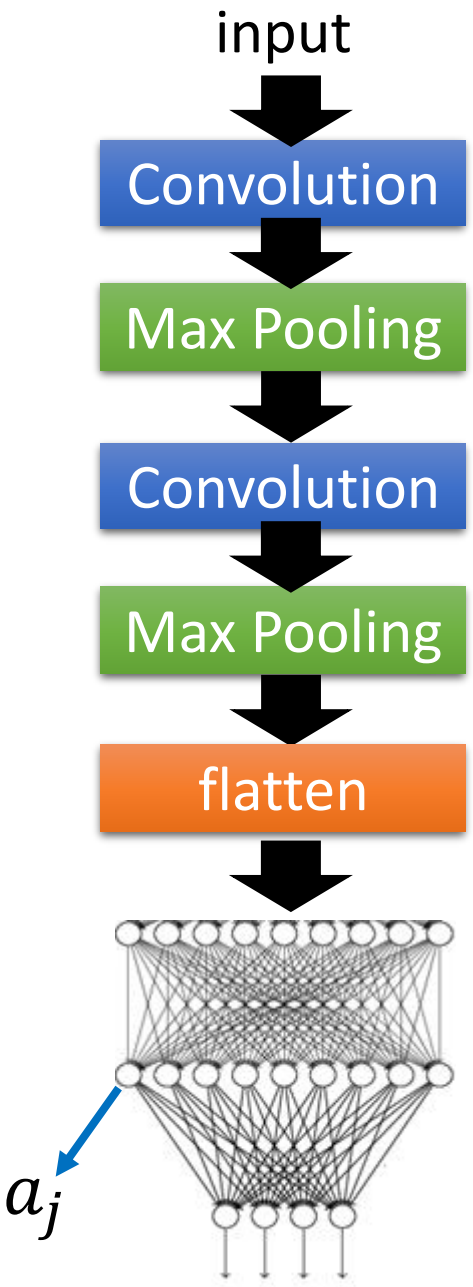
What does CNN learn?

Find an image maximizing the output of neuron:

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



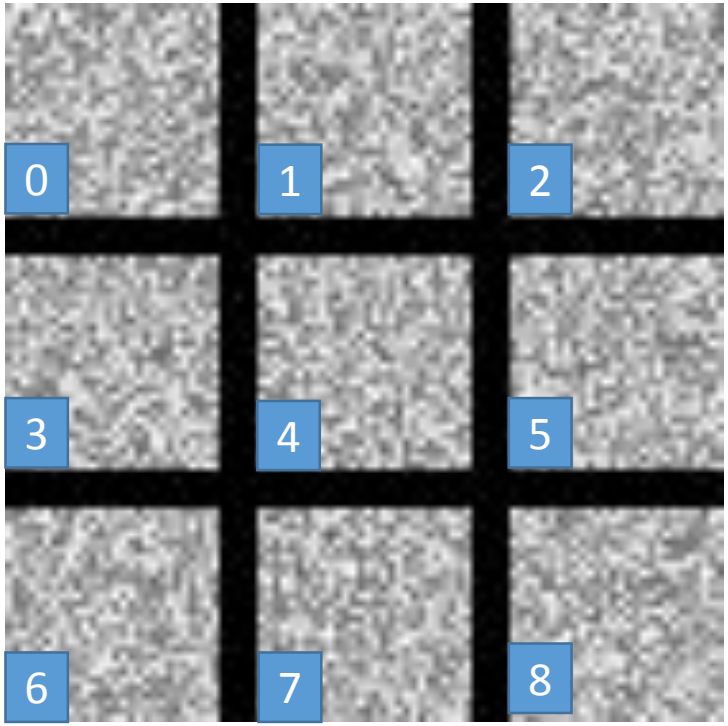
Each figure corresponds to a neuron



What does CNN learn?

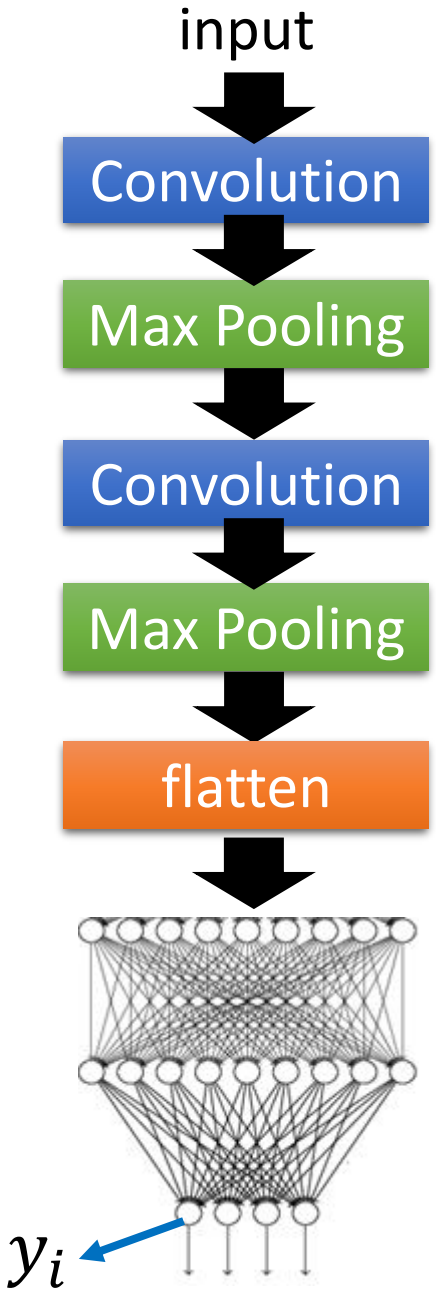
$$x^* = arg\max_x y^i$$

Can we see digits?



Deep Neural Networks are Easily Fooled

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



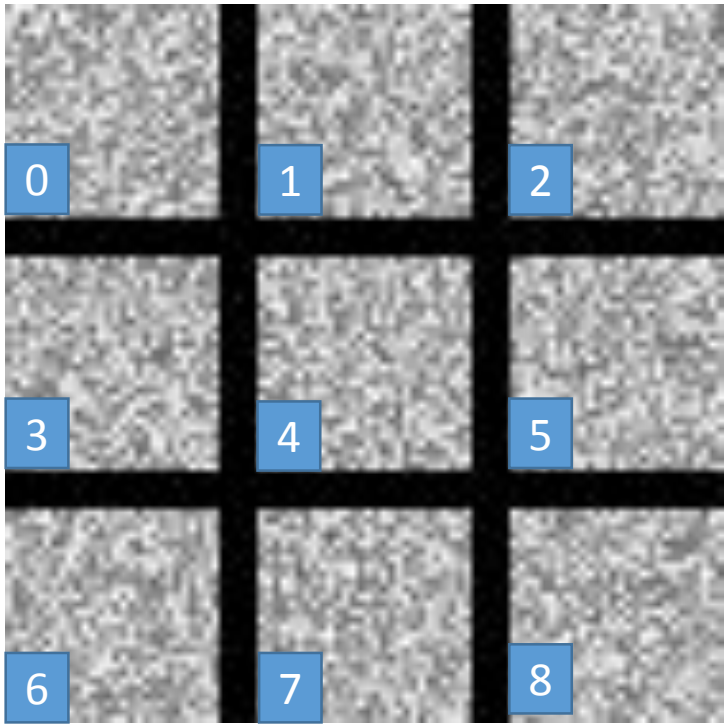
What does CNN learn?

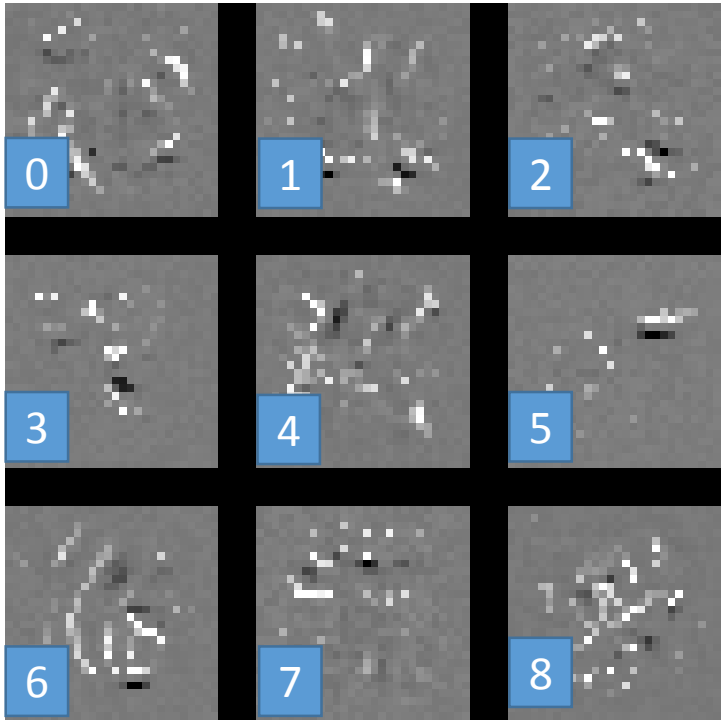
$$x^* = arg\max_x y^i$$

加constraint

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

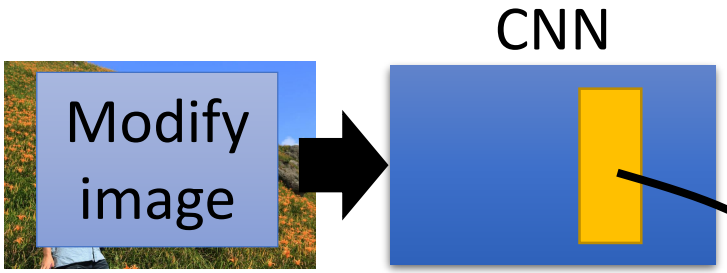
$$\left| \frac{\partial y_k}{\partial x_{ij}} \right|$$

y_k: the predicted class of the model

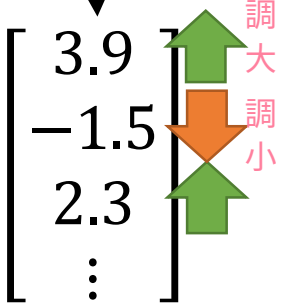
Karen Simonyan, Andrea Vedaldi, Andrew Zisserman, "Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps", ICLR, 2014

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 ……

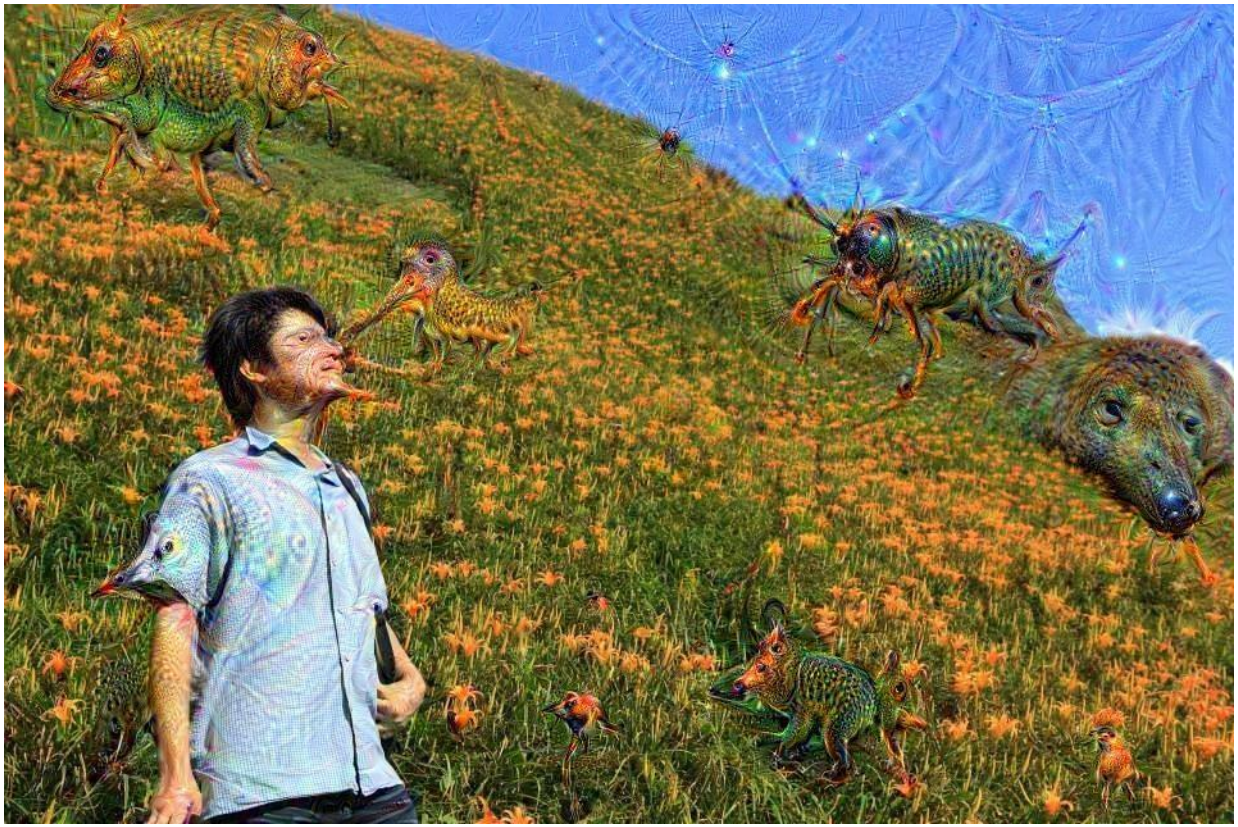


activate的更劇烈，讓CNN誇大他原本看到的東西

http://deepdreamgenerator.com/

Deep Dream

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

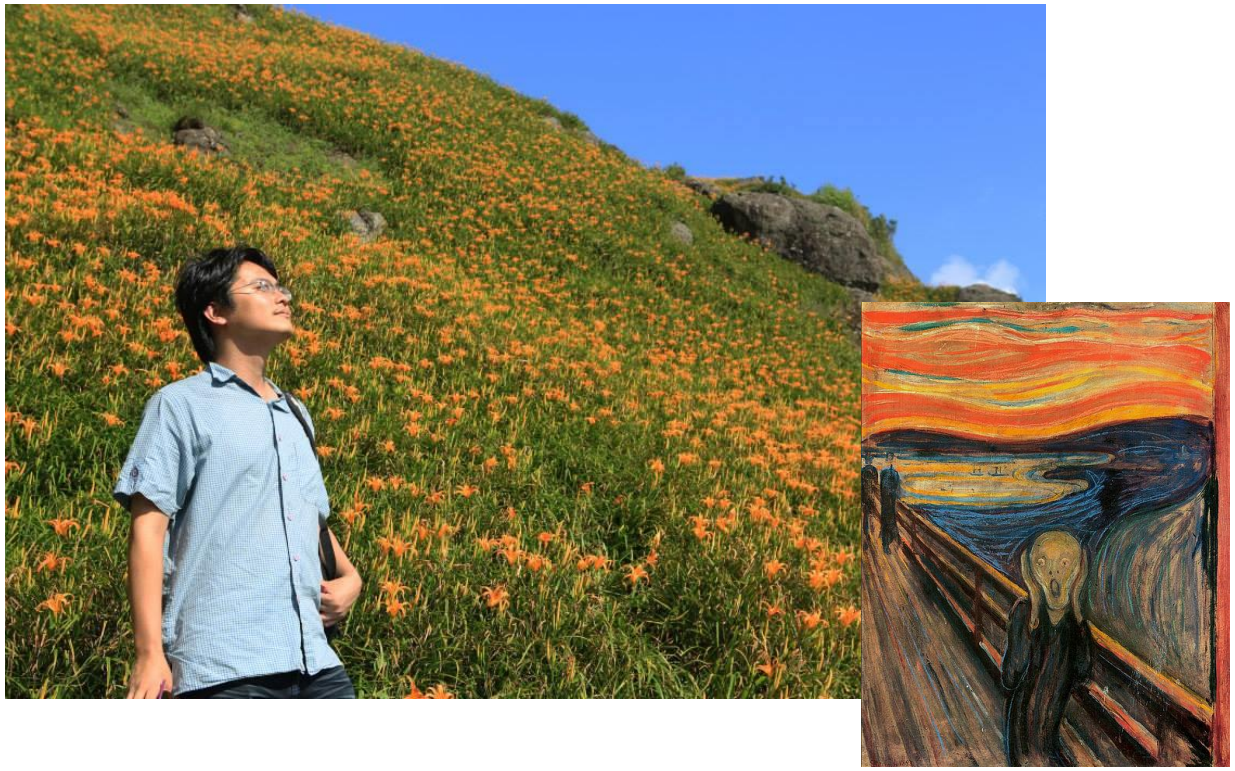


結果看到熊

http://deepdreamgenerator.com/

Deep Style

- Given a photo, make its style like famous paintings



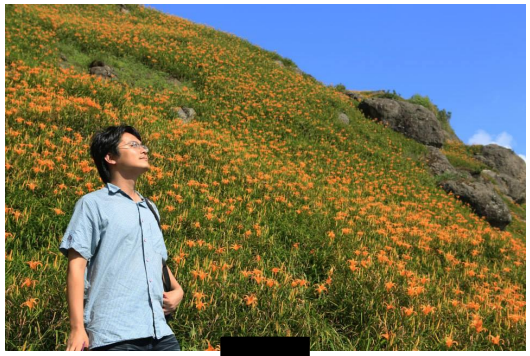
https://dreamscopeapp.com/

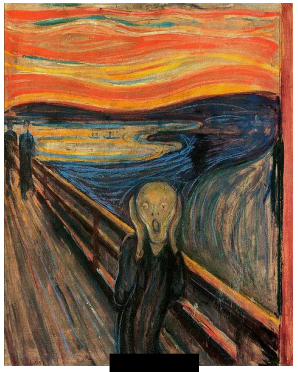
Deep Style

- Given a photo, make its style like famous paintings

https://dreamscopeapp.com/

Deep Style





CNN

CNN

在意兩個filter的output
之間的correlation

output

content

style

A Neural Algorithm of Artistic Style

https://arxiv.org/abs/1508.06576

CNN

?

找一張可以maxmize左右兩邊相關的照片

More Application: Playing Go

19 x 19 matrix (image)

Network

Next move (19 x 19 positions)

19 x 19 vector

Black: 1

white: -1

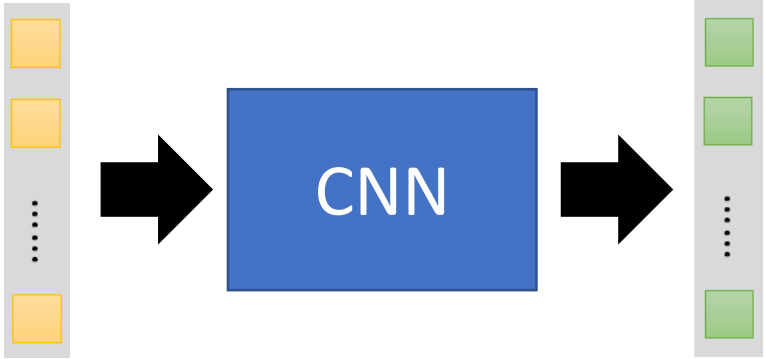
none: 0

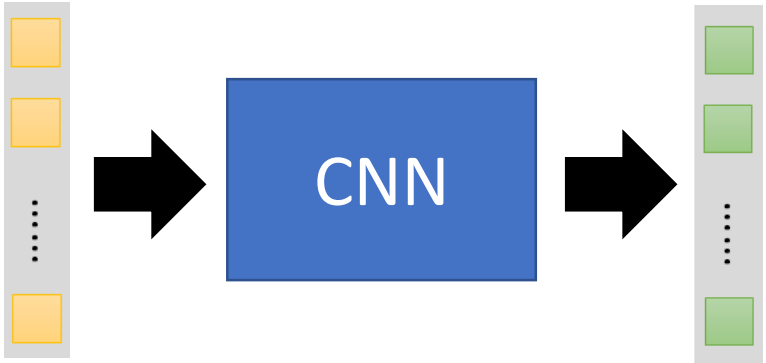
Fully-connected feedforward network can be used

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 …





Target:
“天元” = 1
else = 0

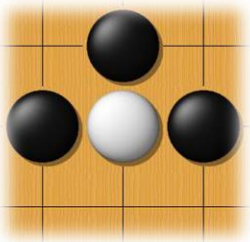
Target:
“五之 5” = 1
else = 0

什麼時候用CNN? 思考最一開始提到的3個特性

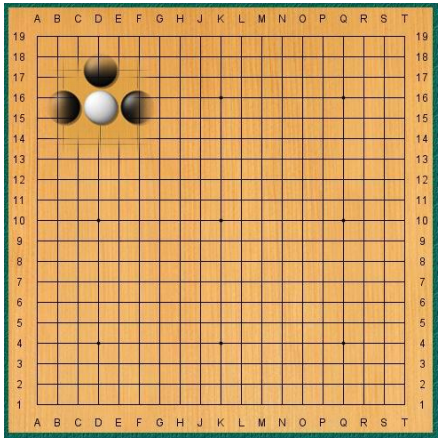
Why CNN for playing Go?

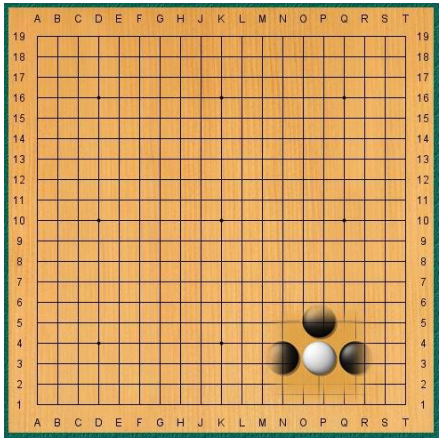
- Some patterns are much smaller than the whole image

  Alpha Go uses 5 x 5 for first layer



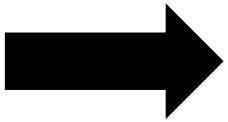
- The same patterns appear in different regions.





Why CNN for playing Go?

- Subsampling the pixels will not change the object



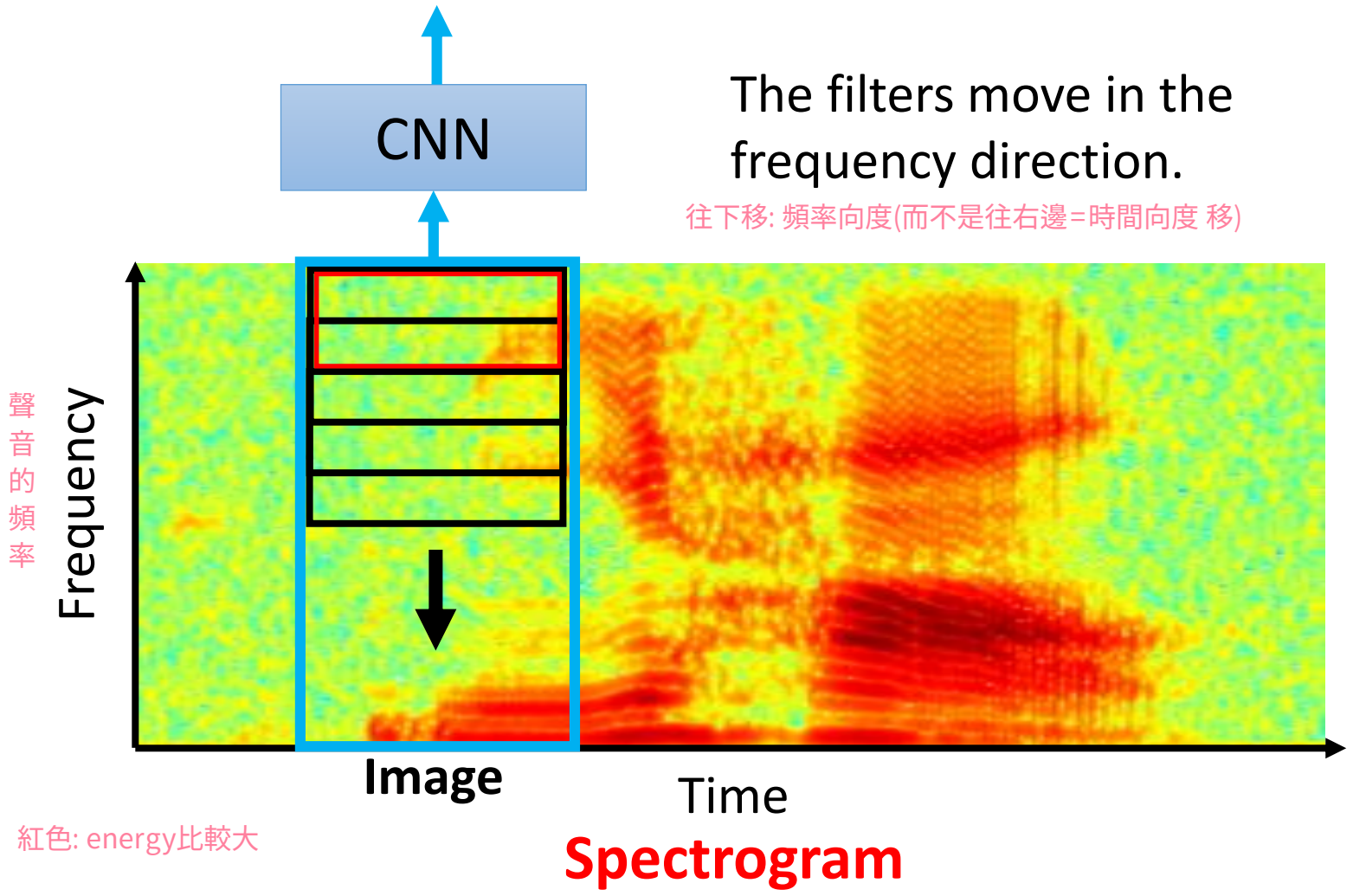
Max Pooling

How to explain this???

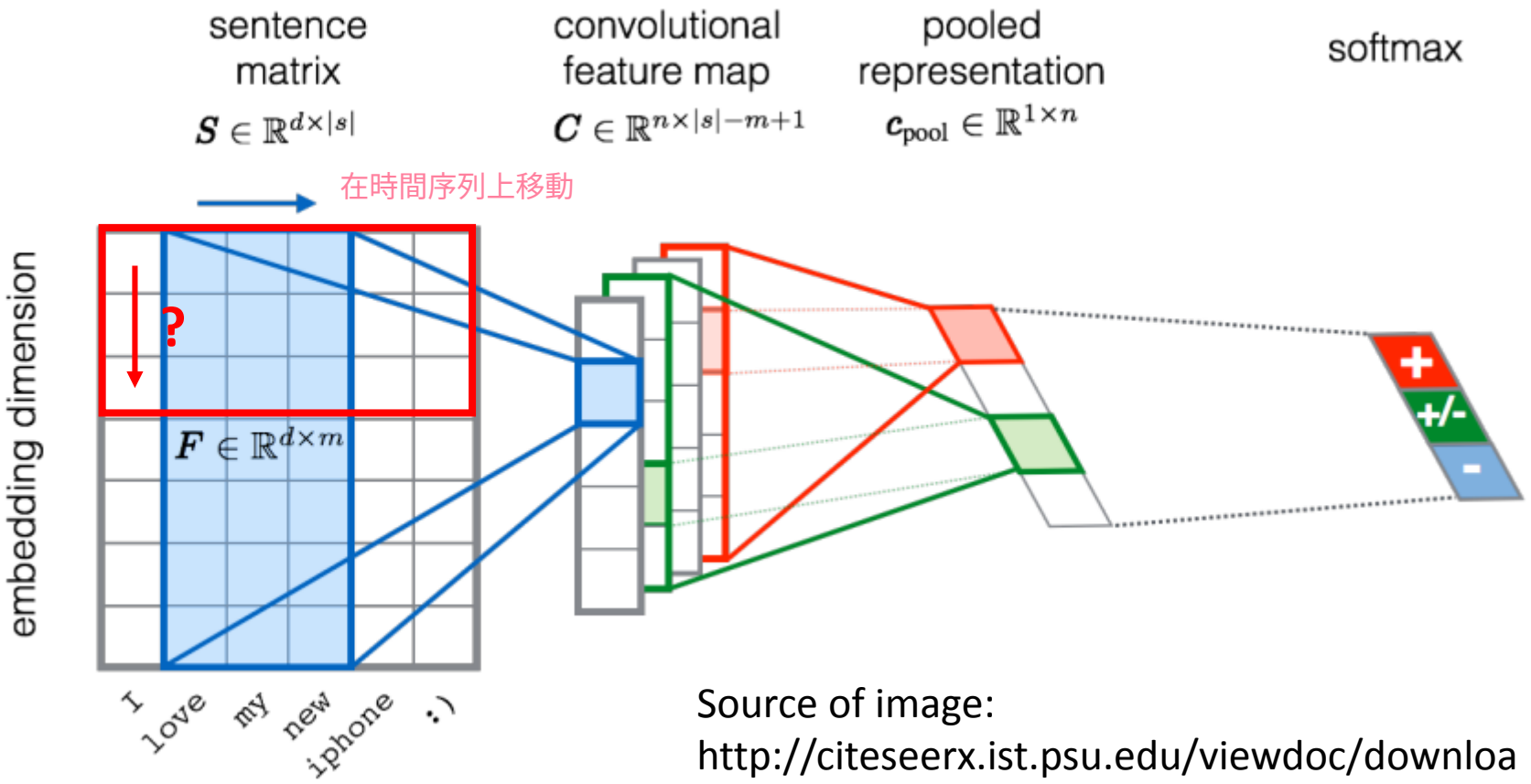
Neural network architecture. The input to the policy network is a $19 \times 19 \times 48$ image stack consisting of 48 feature planes. The first hidden layer zero pads the input into a 23×23 image, then convolves k filters of kernel size 5×5 with stride 1 with the input image and applies a rectifier 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