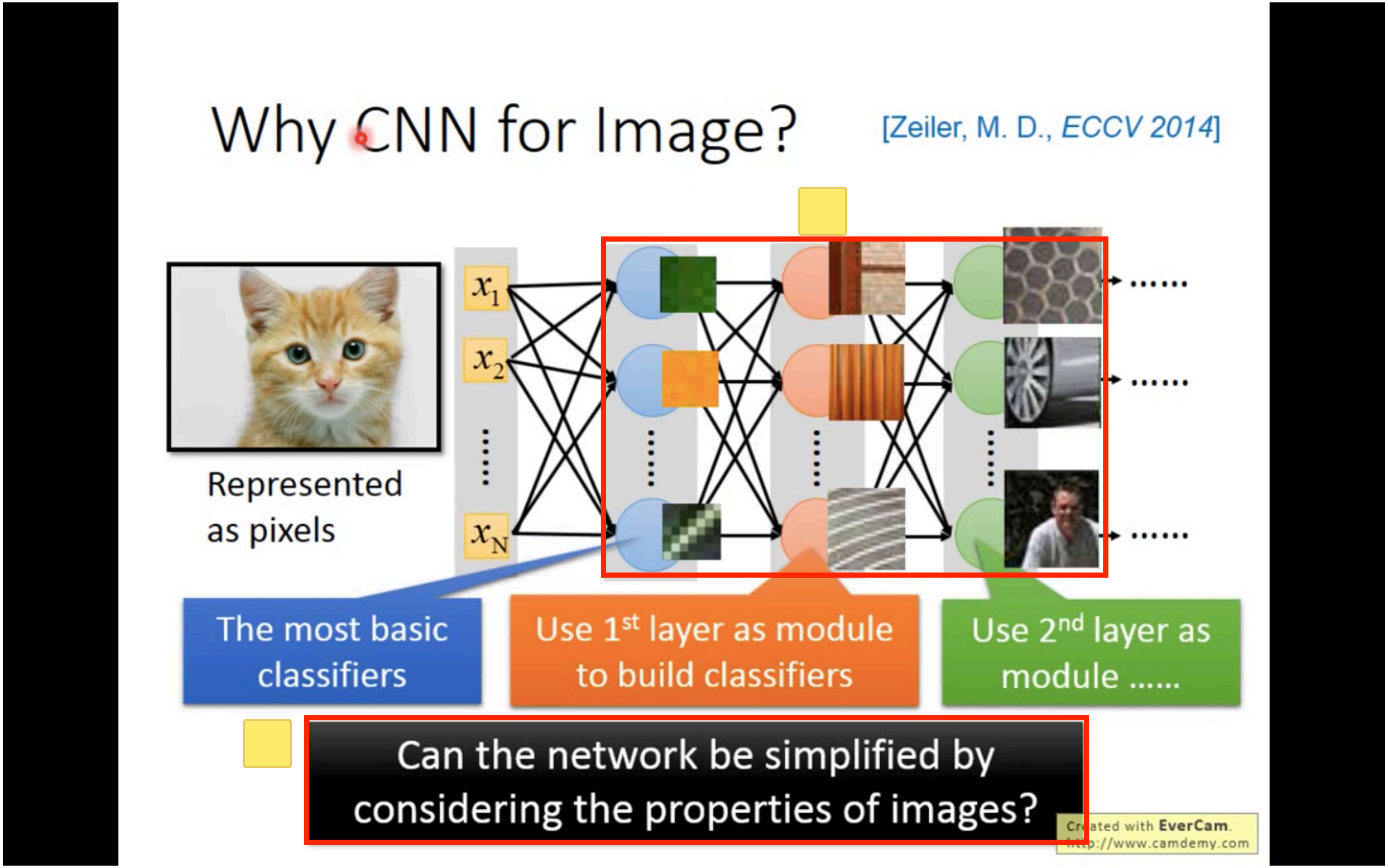
Convolutional Neural Network

Hung-yi Lee

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



Why CNN for Image?

[Zeiler, M. D., *ECCV 2014*]

Represented as pixels

The most basic classifiers

Use 1st layer as module to build classifiers

Use 2nd layer as module ......

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

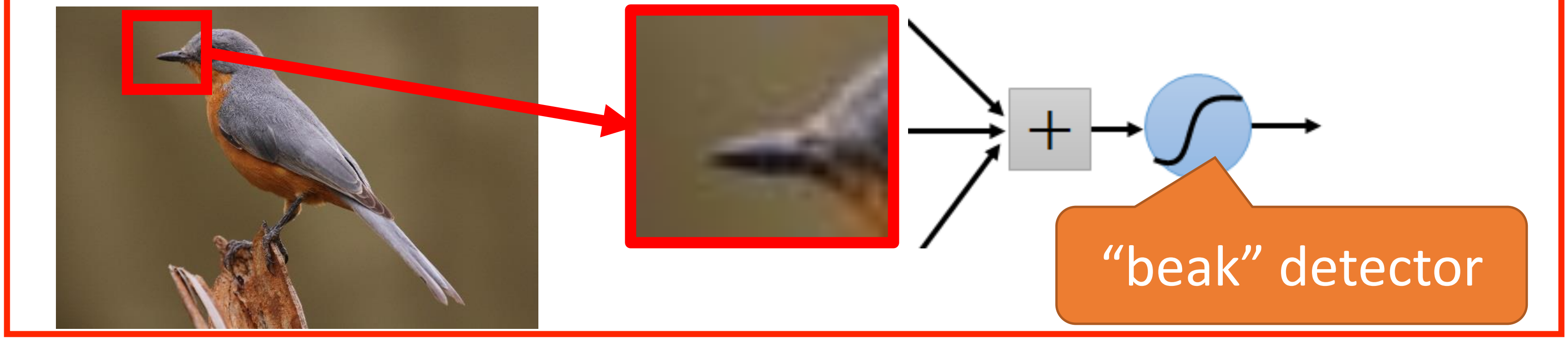
利用 CNN 來處理圖片的第一個好處：辨識一個 Pattern 不需要整張圖片 => 參數量減少

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

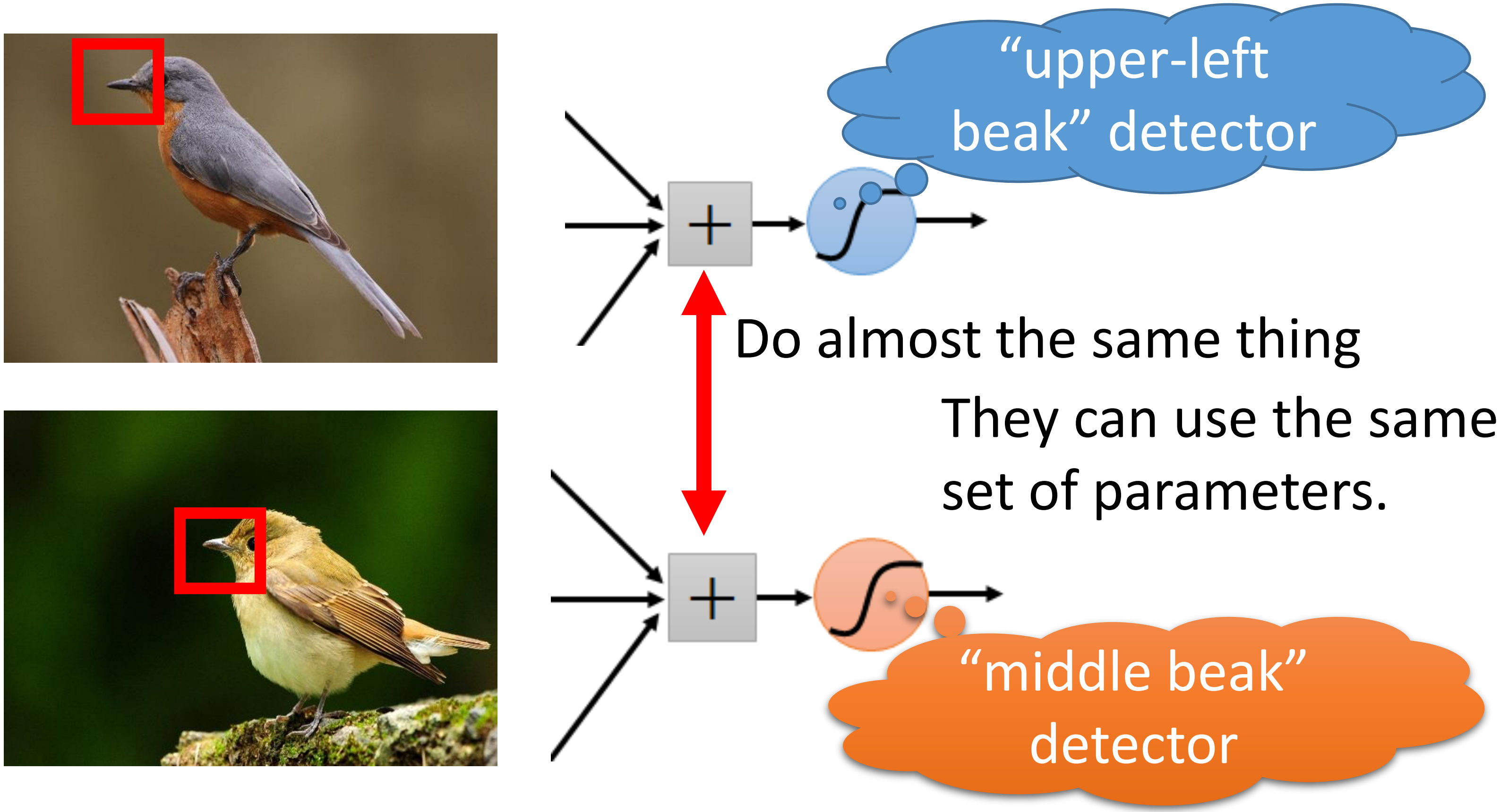


“beak” detector

利用 CNN 來處理圖片的第二個好處：同一個 Classifier 可以辨識不同位置的相同 Pattern => 參數量減少

Why CNN for Image

- The same patterns appear in different regions.



“upper-left beak” detector

Do almost the same thing

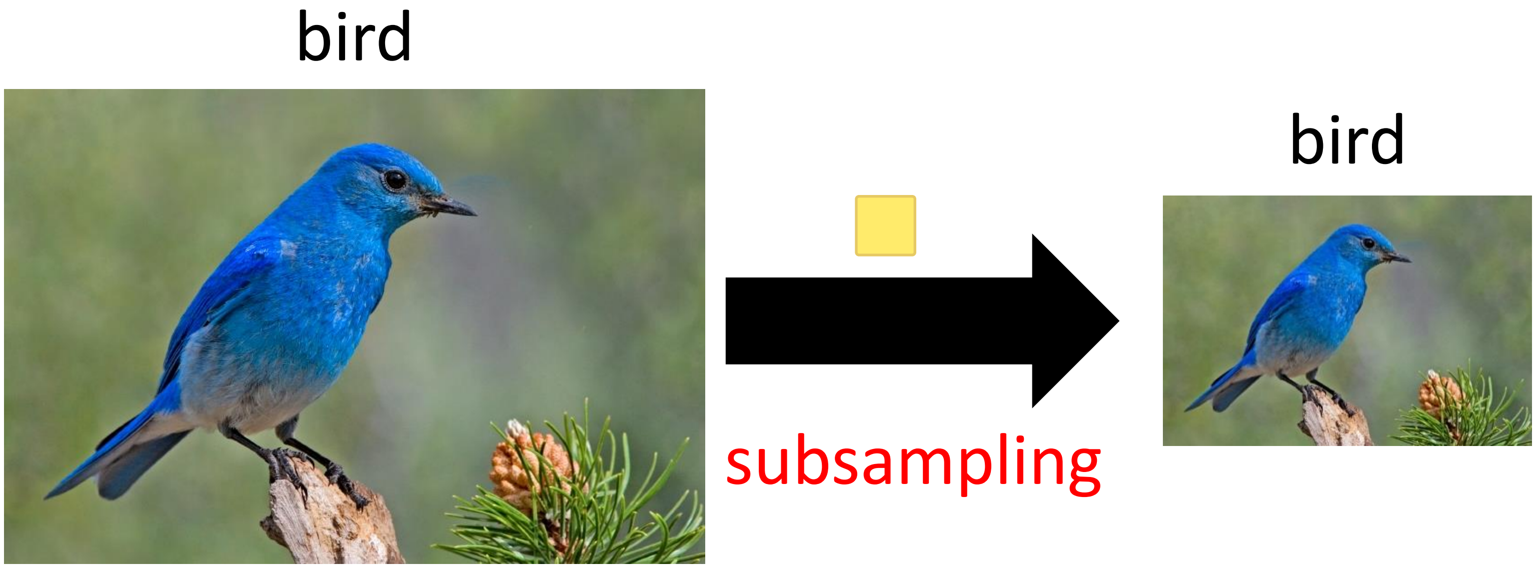
They can use the same set of parameters.

“middle beak” detector

利用 CNN 來處理圖片的第三個好處：將原圖片做 Subsampling 後不影響辨識 => 參數量減少

Why CNN for Image

- Subsampling the pixels will not change the object

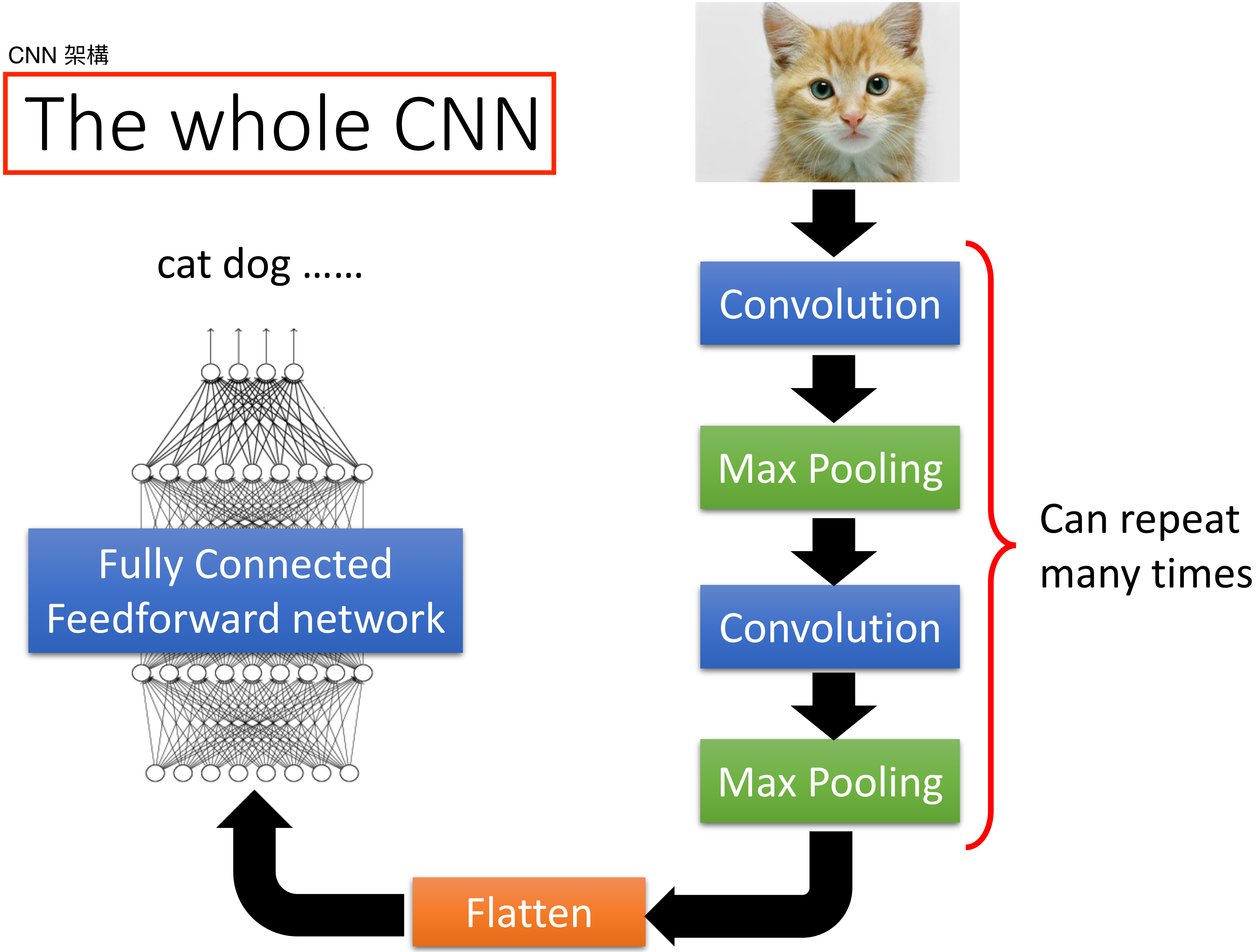


We can subsample the pixels to make image smaller

→ Less parameters for the network to process the image

CNN 架構

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

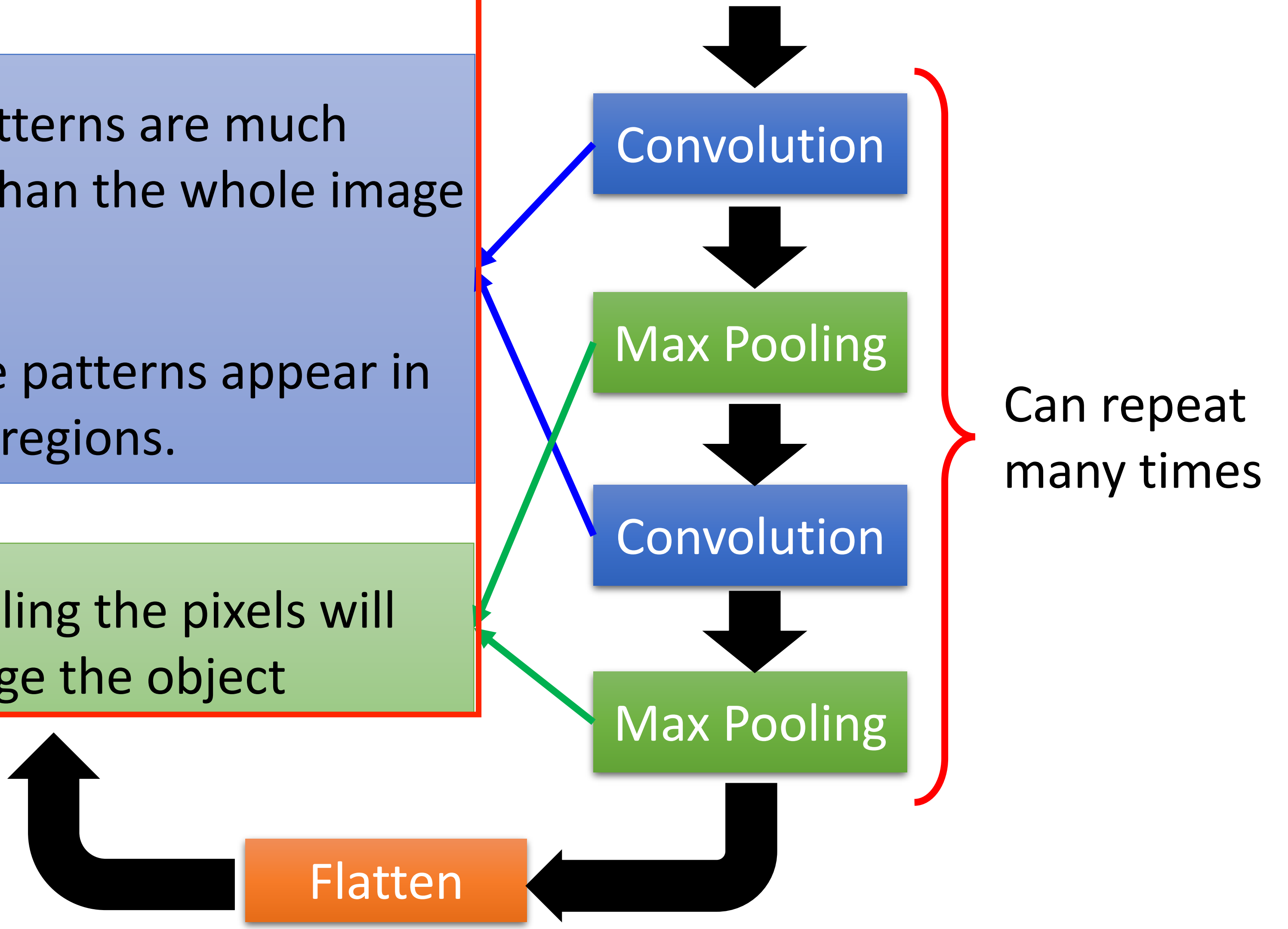
- ➢ Some patterns are much smaller than the whole image

Property 2

- ➢ The same patterns appear in different regions.

Property 3

- ➢ Subsampling the pixels will not change the object



Convolution

Max Pooling

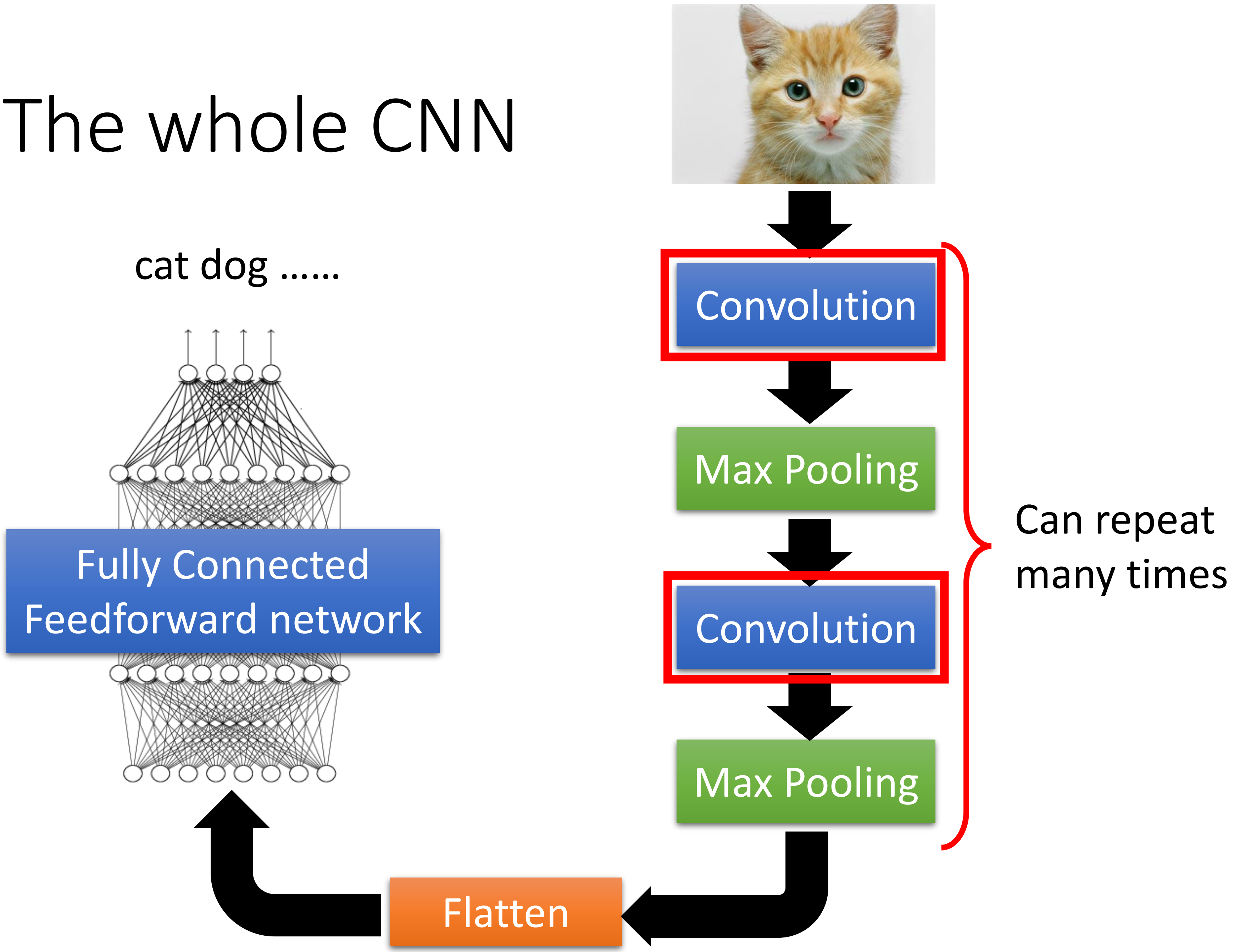
Convolution

Max Pooling

Can repeat many times

Flatten

The whole CNN



cat dog ……

Convolution

Max Pooling

Convolution

Max Pooling

Can repeat many times

Flatten

Fully Connected Feedforward network

Convolution Layer 中包含：一張 Image、很多張 Filter

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

每個 Filter 都是一個 Matrix，Matrix 裡面的數值是 CNN 自己要去學習的！

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

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

Filter 1

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

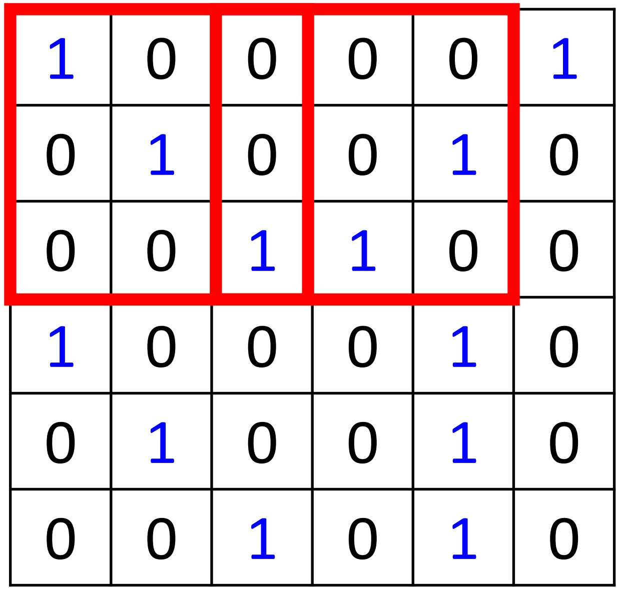
3 -1

CNN – Convolution

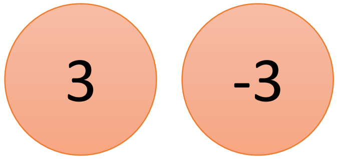
1	-1	-1
-1	1	-1
-1	-1	1

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



We set stride=1 below

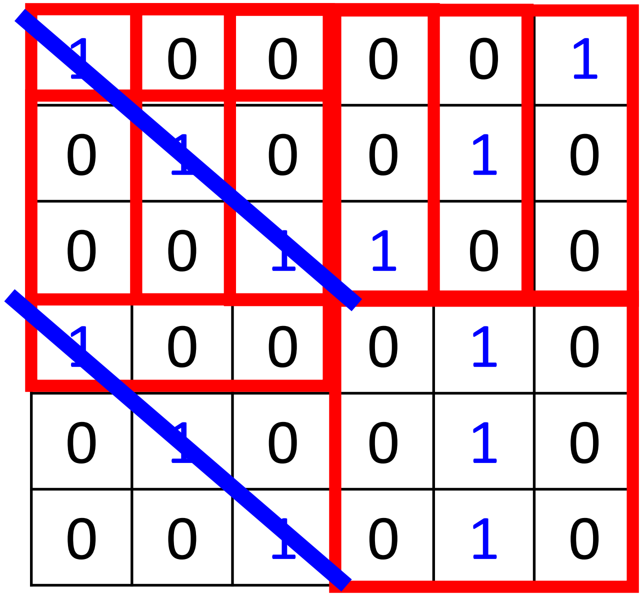
6 x 6 image

CNN – Convolution

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

Filter 1

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

3	-1	-3	-1
-3	1	0	-3
-3	-3	0	1
3	-2	-2	-1

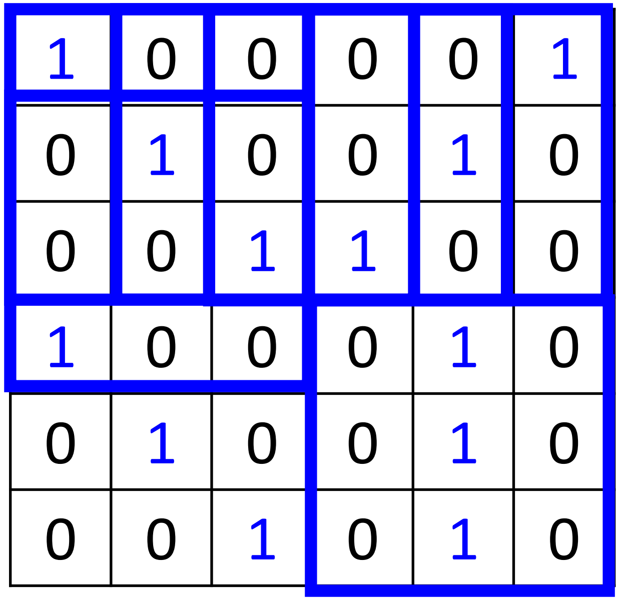
Property 2

CNN – Convolution

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

Filter 2

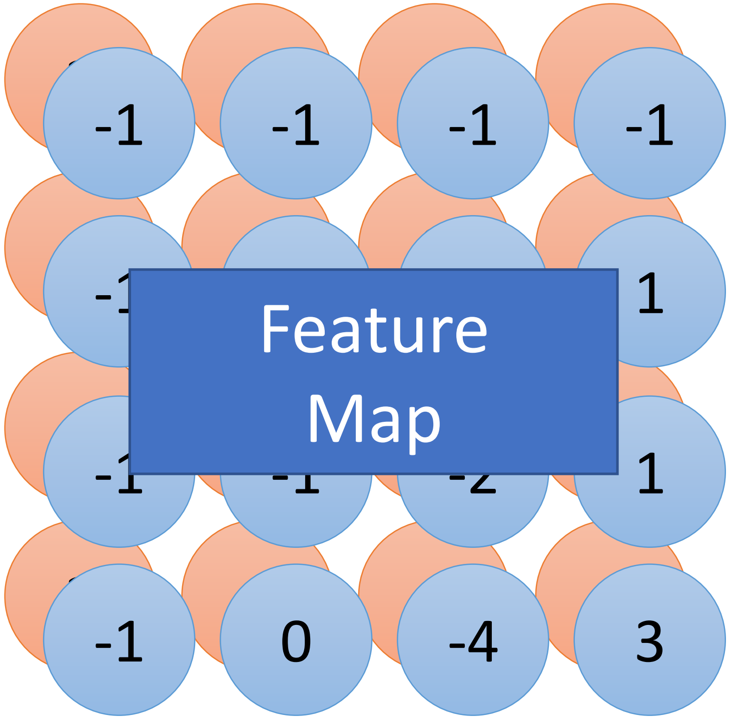
stride=1



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

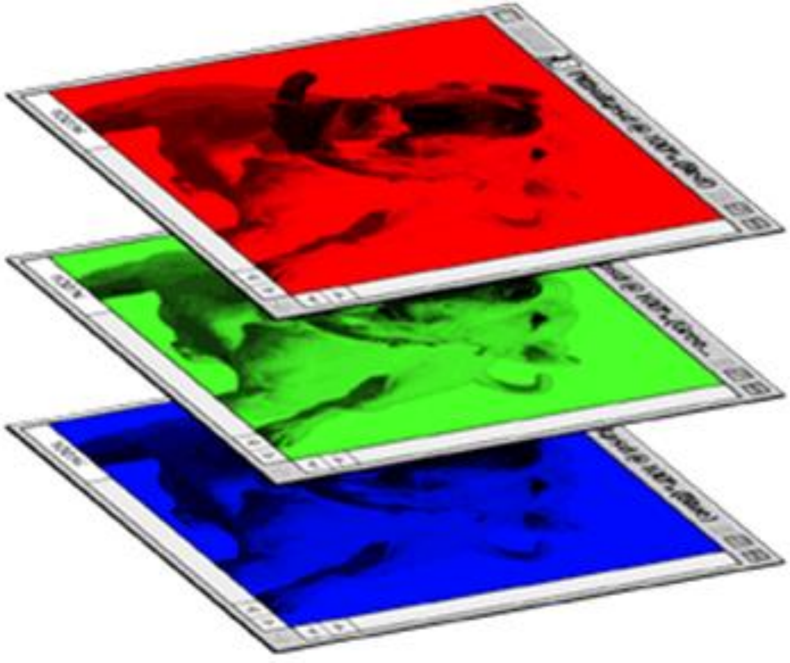
1	-1	-1
-1	1	-1
-1	-1	1

Filter 1

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

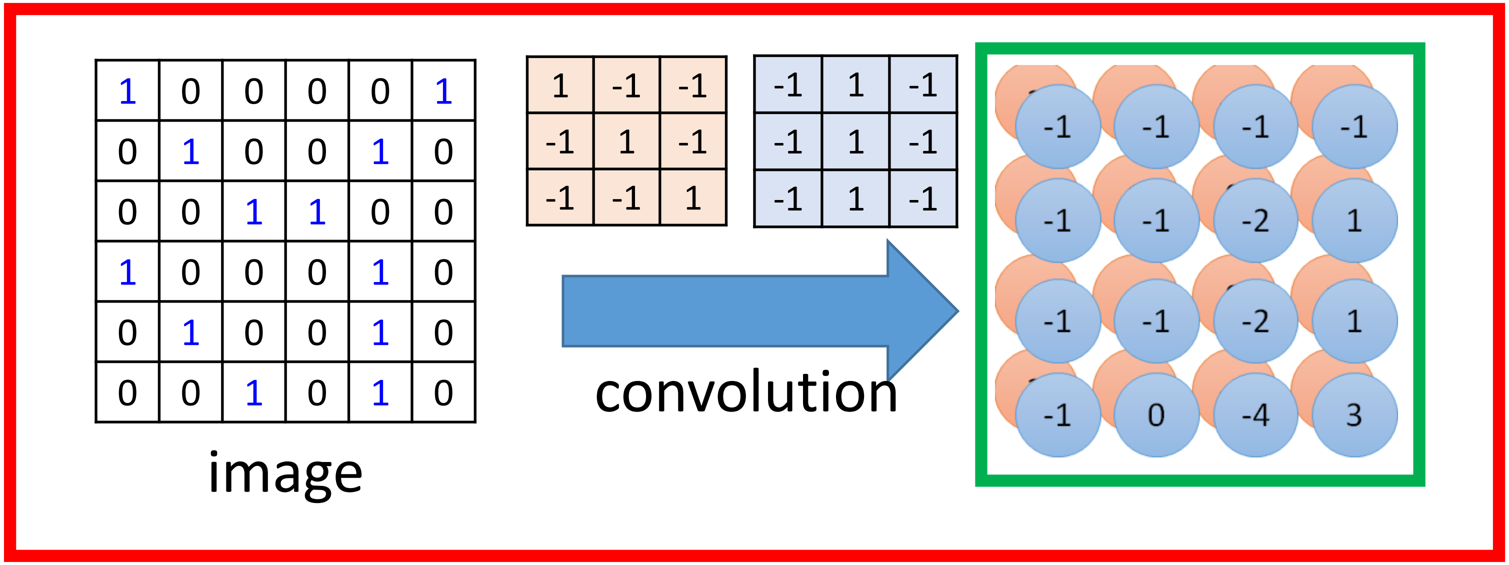
Filter 2

Colorful image



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

Convolution v.s. Fully Connected



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

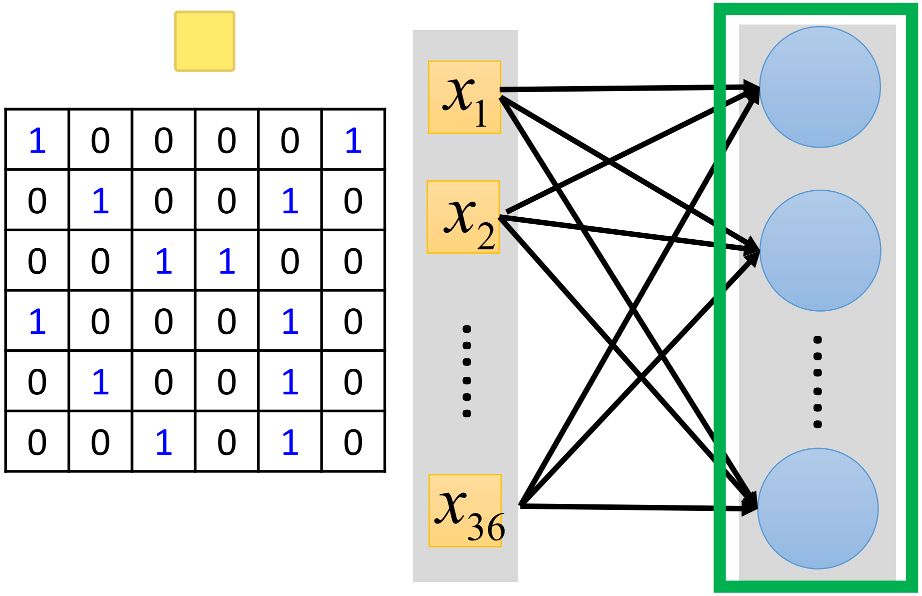
image

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

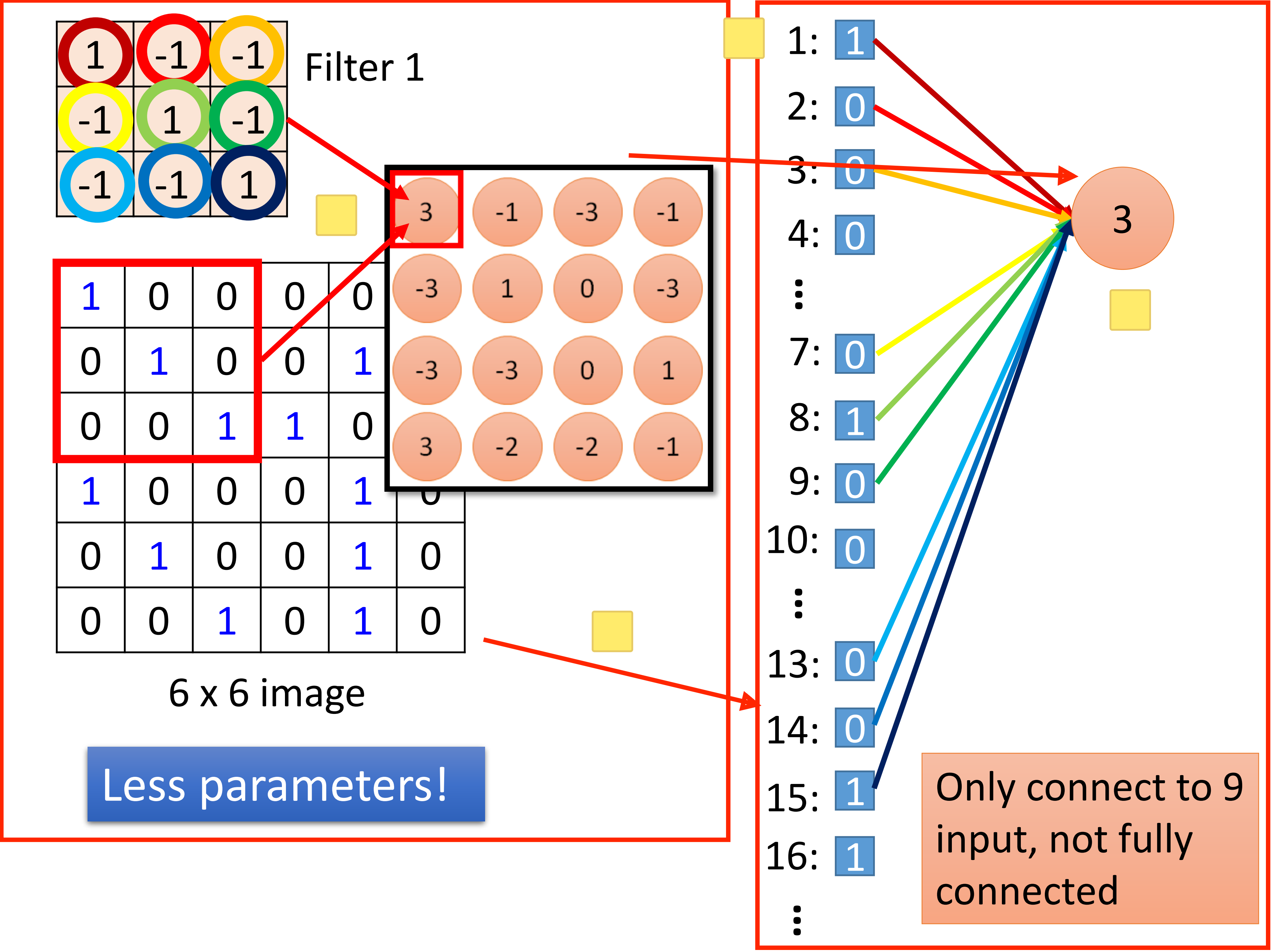
-1	1	-1
-1	1	-1
-1	1	-1

-1	-1	-1	-1
-1	-1	-2	1
-1	-1	-2	1
-1	0	-4	3

Fully-connected



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



Filter 1

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

3	-1	-3	-1
-3	1	0	-3
-3	-3	0	1
3	-2	-2	-1

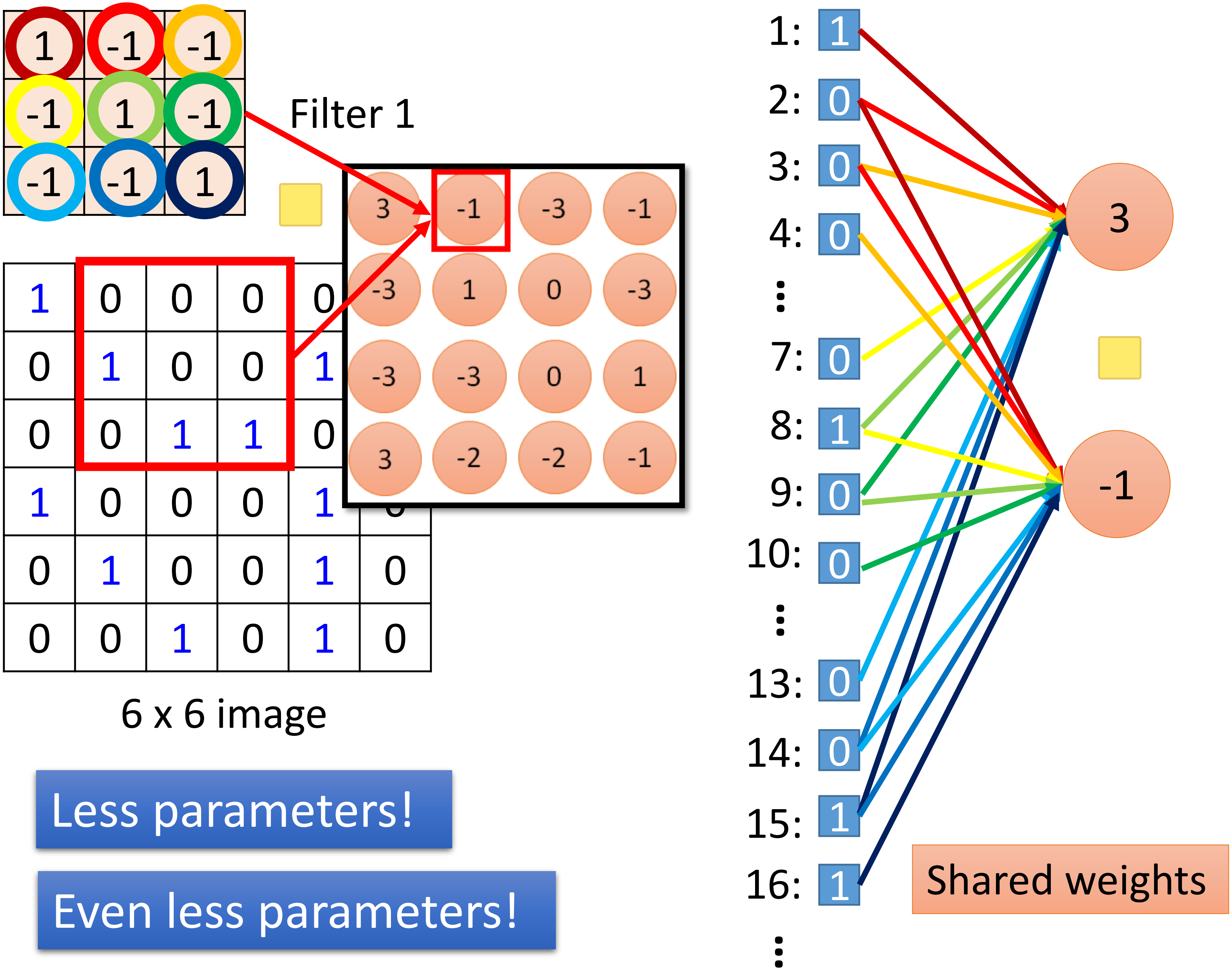
1	0	0	0	0	
0	1	0	0	1	
0	0	1	1	0	
1	0	0	0	1	0
0	1	0	0	1	0
0	0	1	0	1	0

6 x 6 image

Less parameters!

1:	1
2:	0
3:	0
4:	0
⋮	
7:	0
8:	1
9:	0
10:	0
⋮	
13:	0
14:	0
15:	1
16:	1
⋮	

Only connect to 9 input, not 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

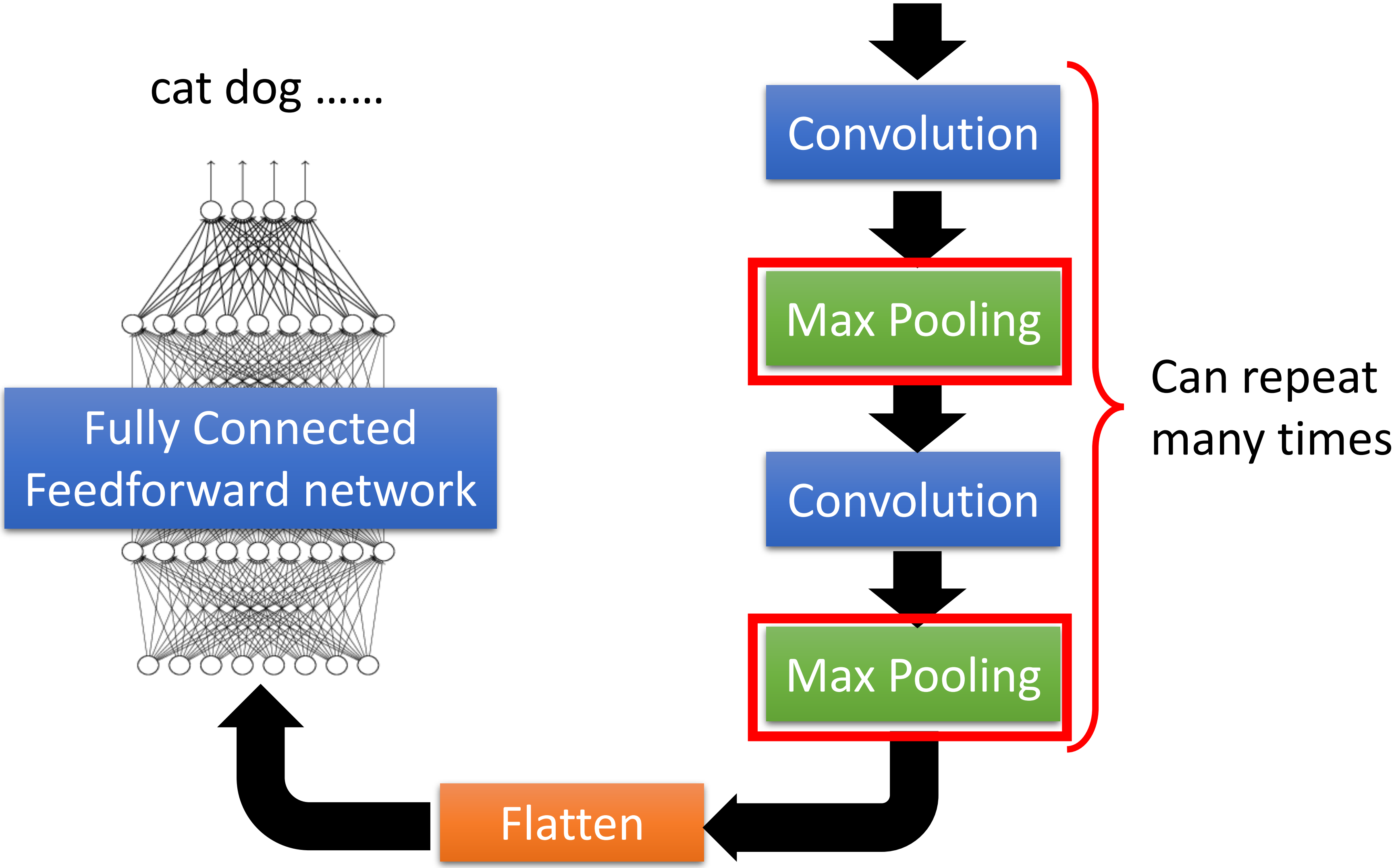
6 x 6 image

Shared weights

Less parameters!

Even less parameters!

The whole CNN



cat dog ……

Convolution

Max Pooling

Convolution

Max Pooling

Can repeat many times

Flatten

Fully Connected Feedforward network

Max Pooling 相當於在把 Image (Feature Map) 做 Subsampling => 使 Image (Feature Map) 的 Size 更小

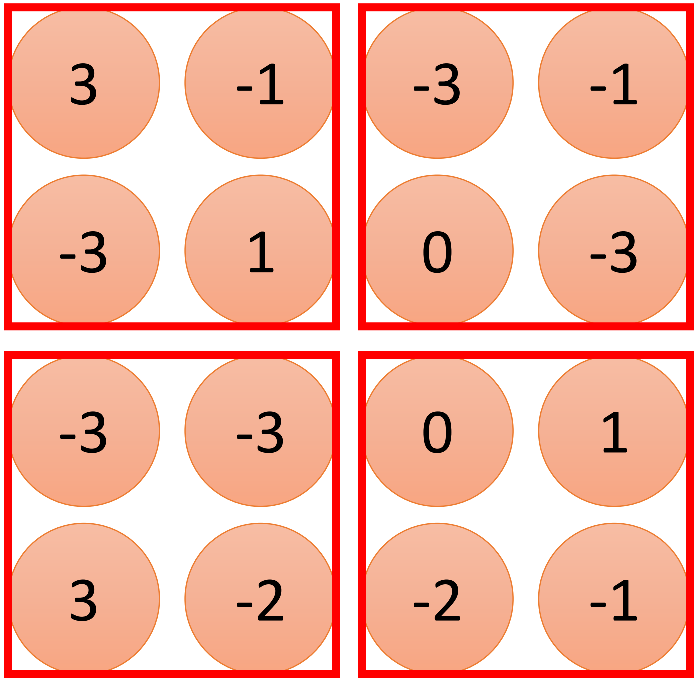
CNN – Max Pooling

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

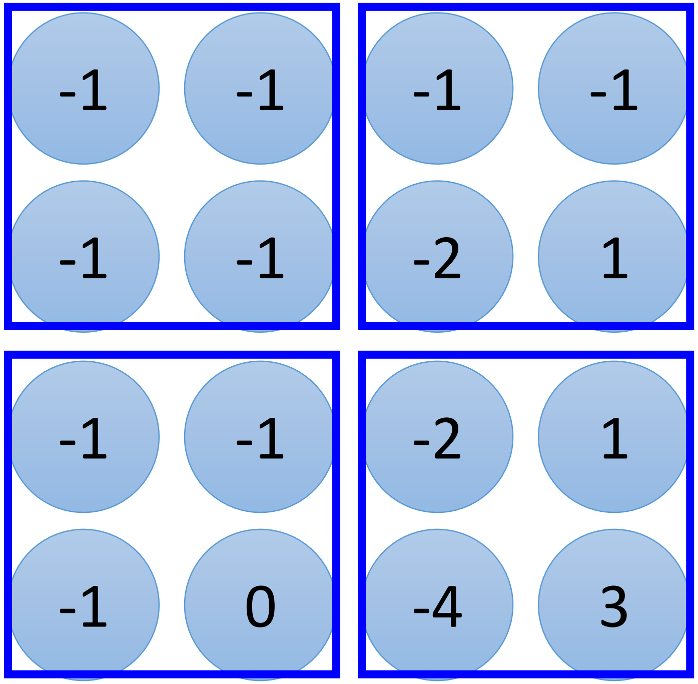
Filter 1

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

Filter 2



3	-1	-3	-1
-3	1	0	-3
-3	-3	0	1
3	-2	-2	-1

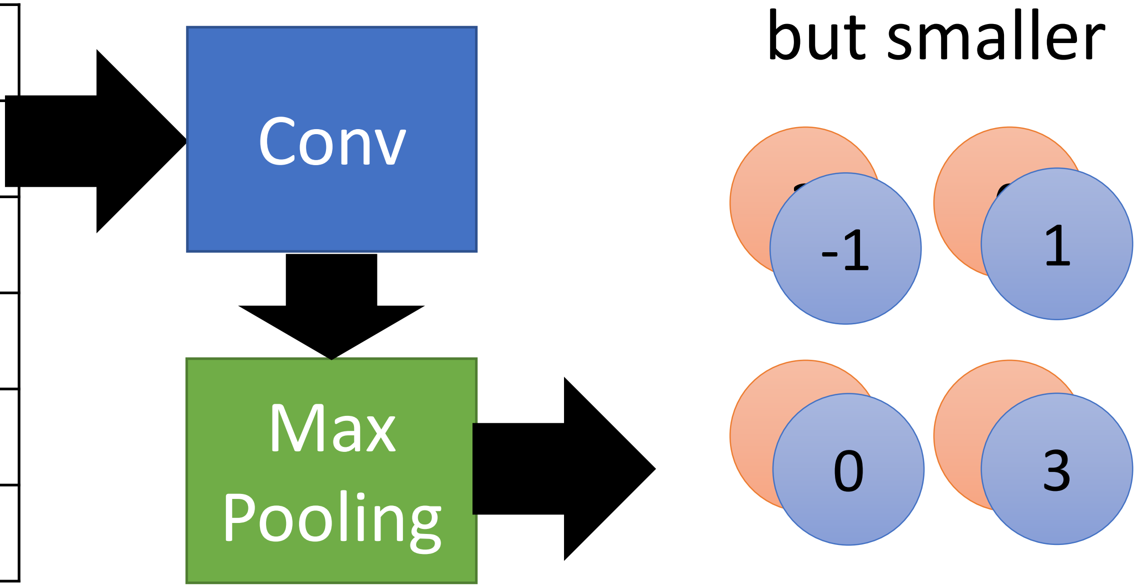


-1	-1	-1	-1
-1	-1	-2	1
-1	-1	-2	1
-1	0	-4	3

CNN – Max Pooling

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

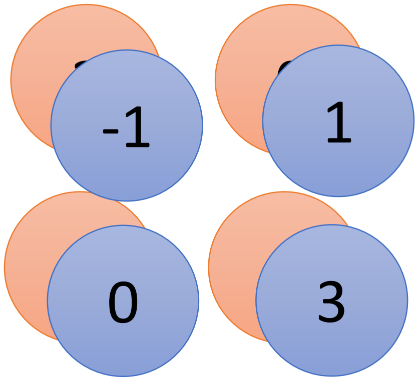


New image but smaller

2 x 2 image

Each filter is a channel

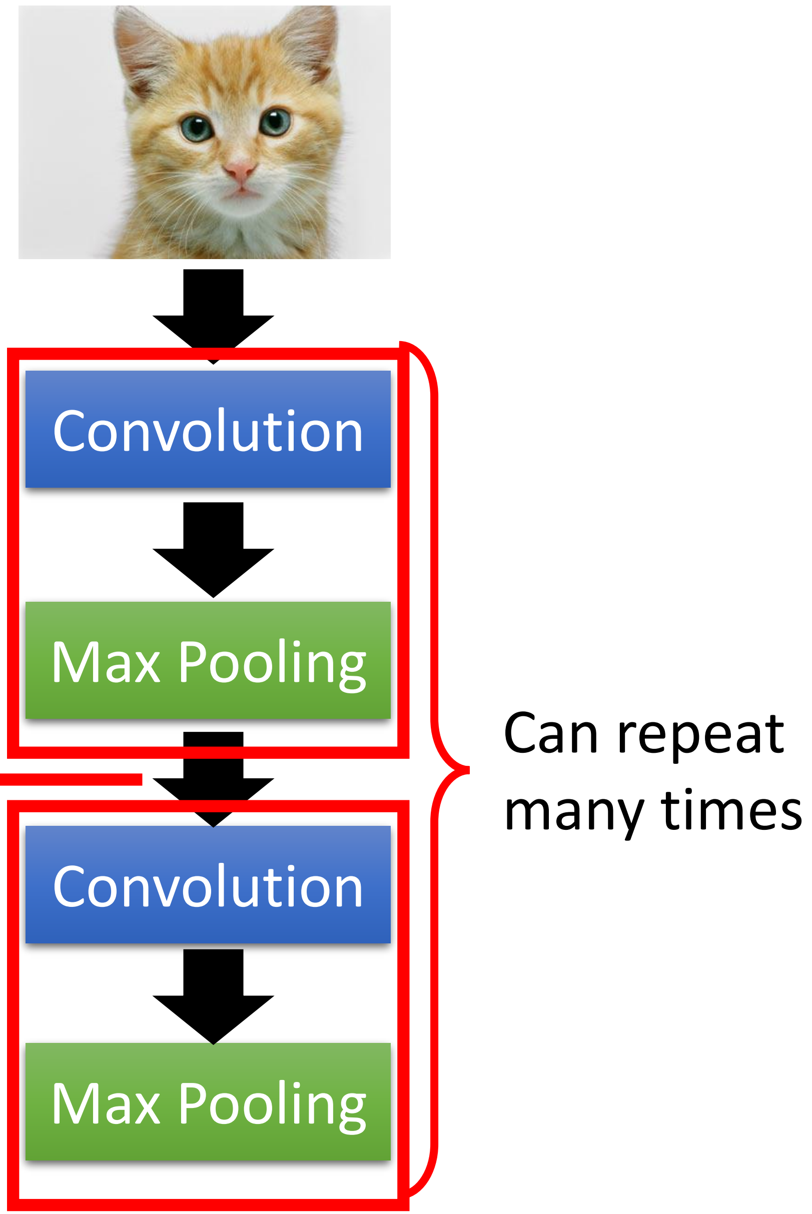
The whole CNN



A new image

Smaller than the original image

The number of the channel is the number of filters



Convolution

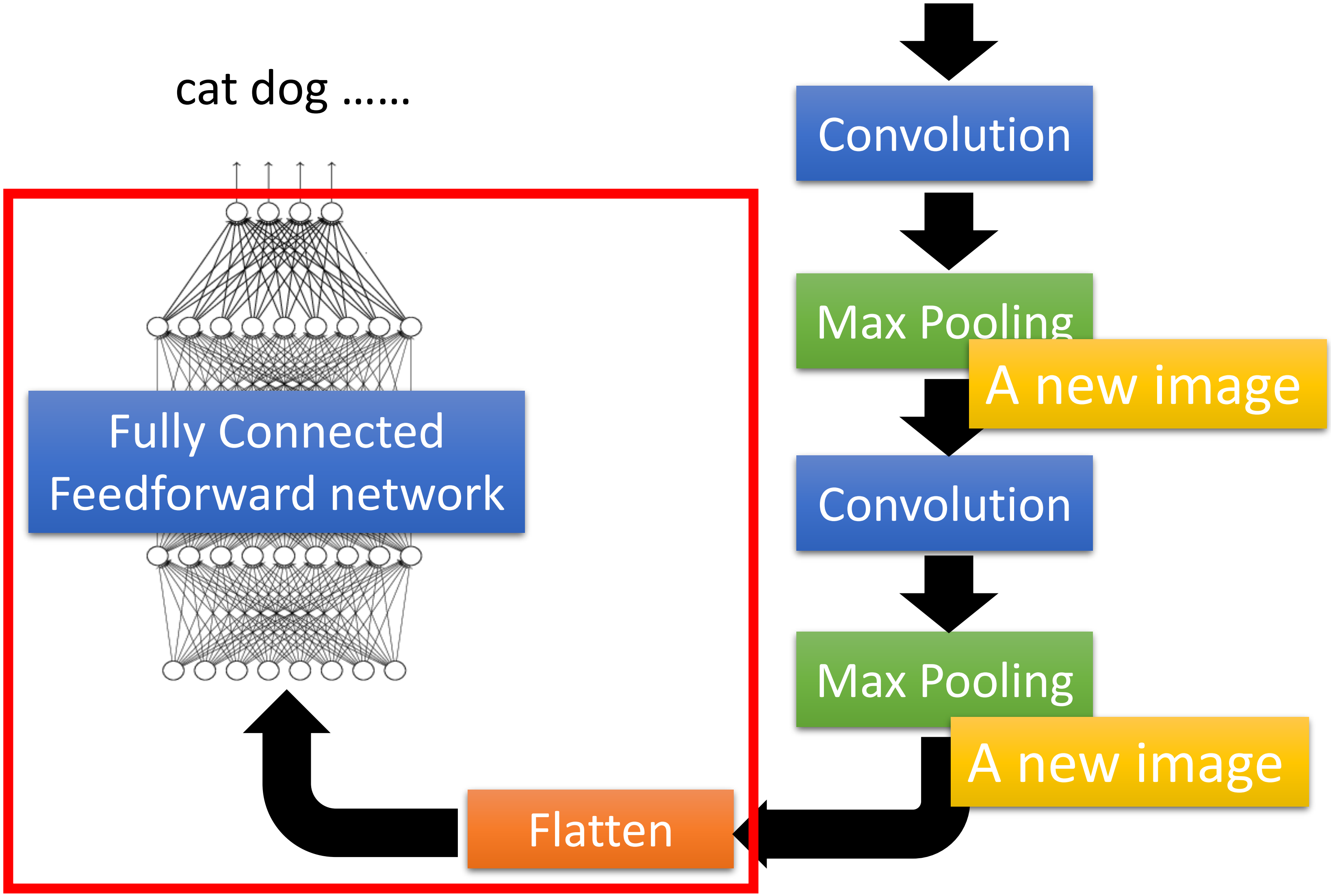
Max Pooling

Convolution

Max Pooling

Can repeat many times

The whole CNN



cat dog ……

Convolution

Max Pooling

A new image

Convolution

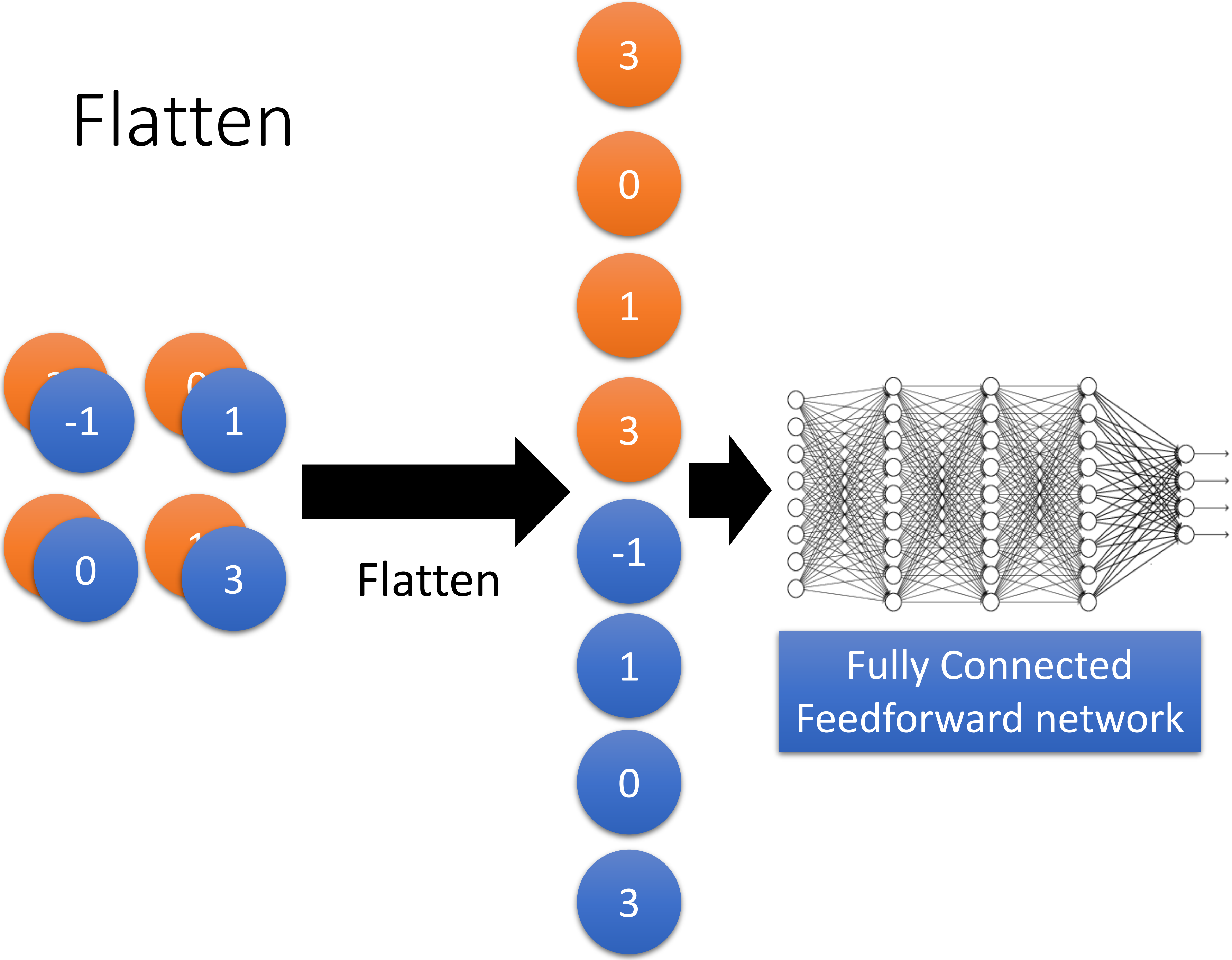
Max Pooling

A new image

Flatten

Fully Connected Feedforward network

Flatten



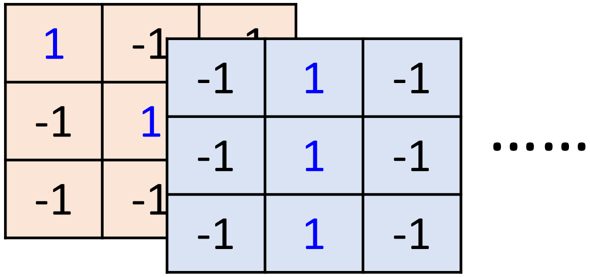
Flatten

Fully Connected Feedforward network

CNN in Keras

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

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



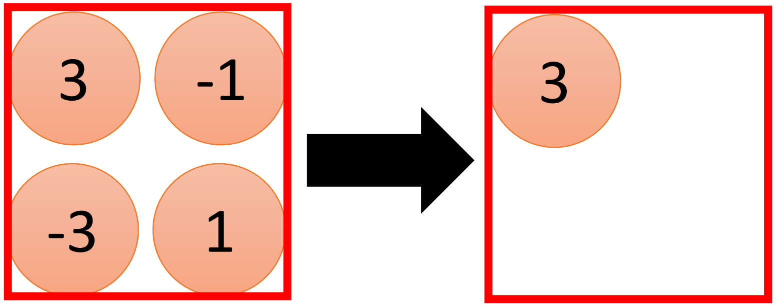
There are 25 3x3 filters.

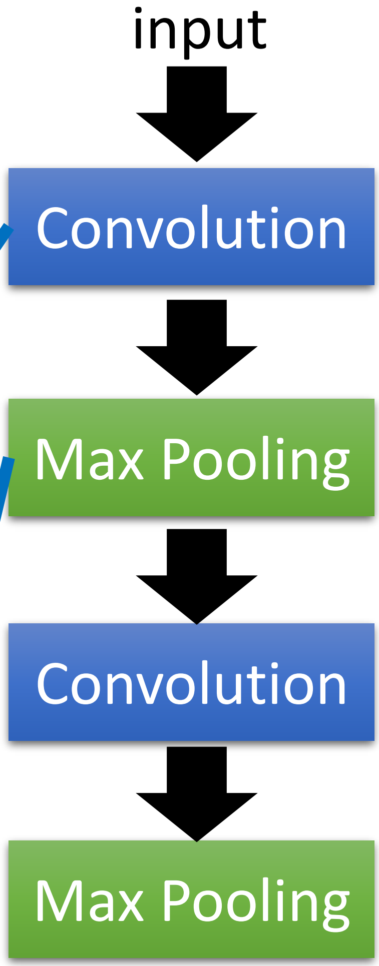
Input_shape = (28 , 28 , 1)

28 x 28 pixels

1: black/white, 3: RGB

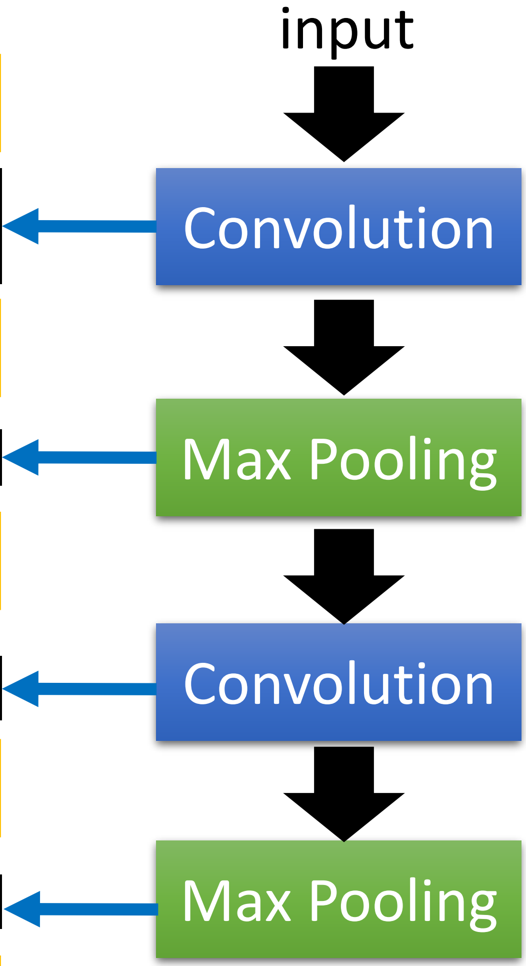
```
model2.add(MaxPooling2D((2,2)))
```



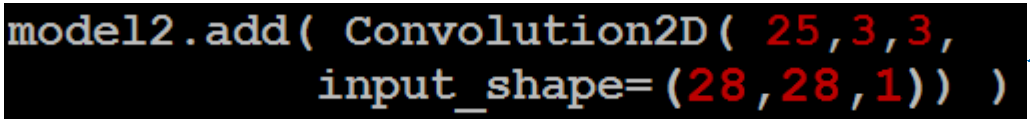


CNN in Keras

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



1 x 28 x 28

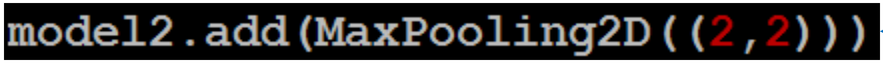


```
model2.add( Convolution2D( 25,3,3,
        input_shape=(28,28,1)) )
```

How many parameters for each filter?

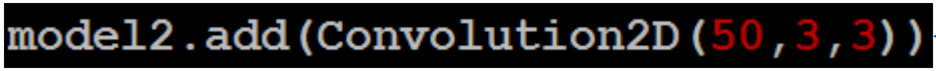
9

25 x 26 x 26



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

25 x 13 x 13

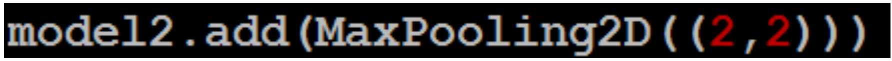


```
model2.add(Convolution2D(50,3,3))
```

How many parameters for each filter?

225

50 x 11 x 11

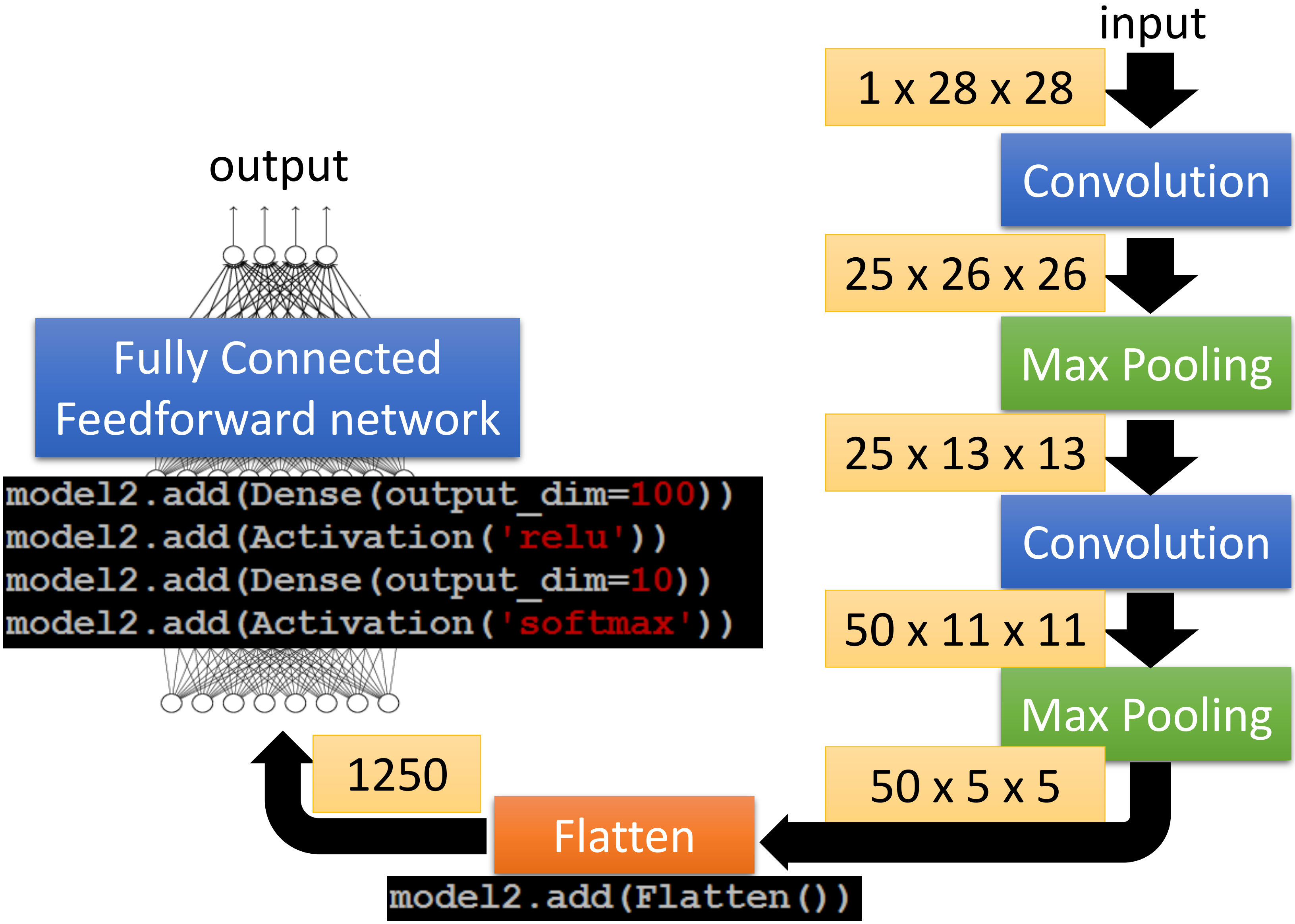


```
model2.add(MaxPooling2D((2,2)))
```

50 x 5 x 5

CNN in Keras

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



```
model2.add(Dense(output_dim=100))
model2.add(Activation('relu'))
model2.add(Dense(output_dim=10))
model2.add(Activation('softmax'))
```

```
model2.add(Flatten())
```