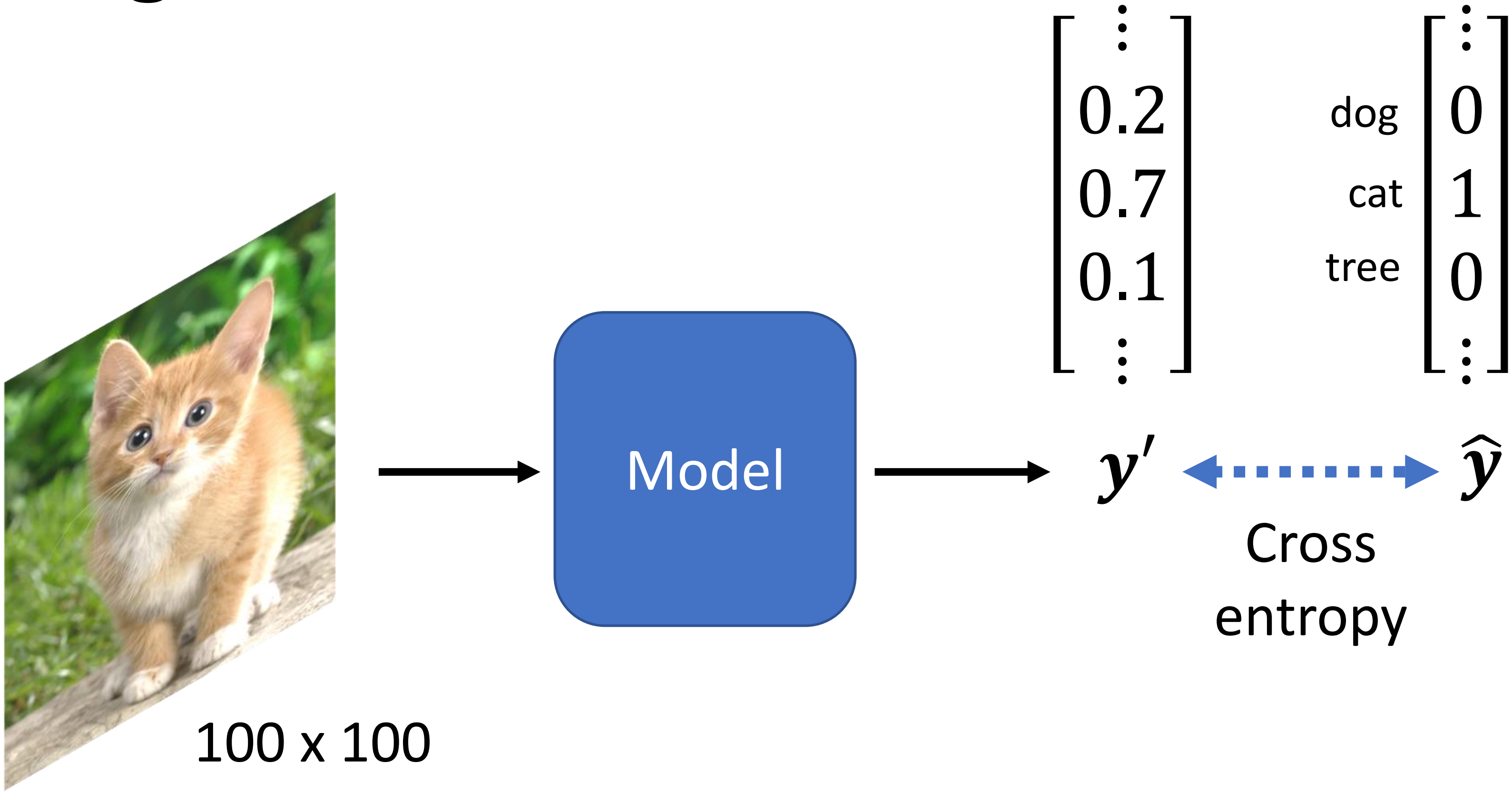
Convolutional Neural Network (CNN)

Network Architecture designed for Image

Image Classification



100 x 100

Model

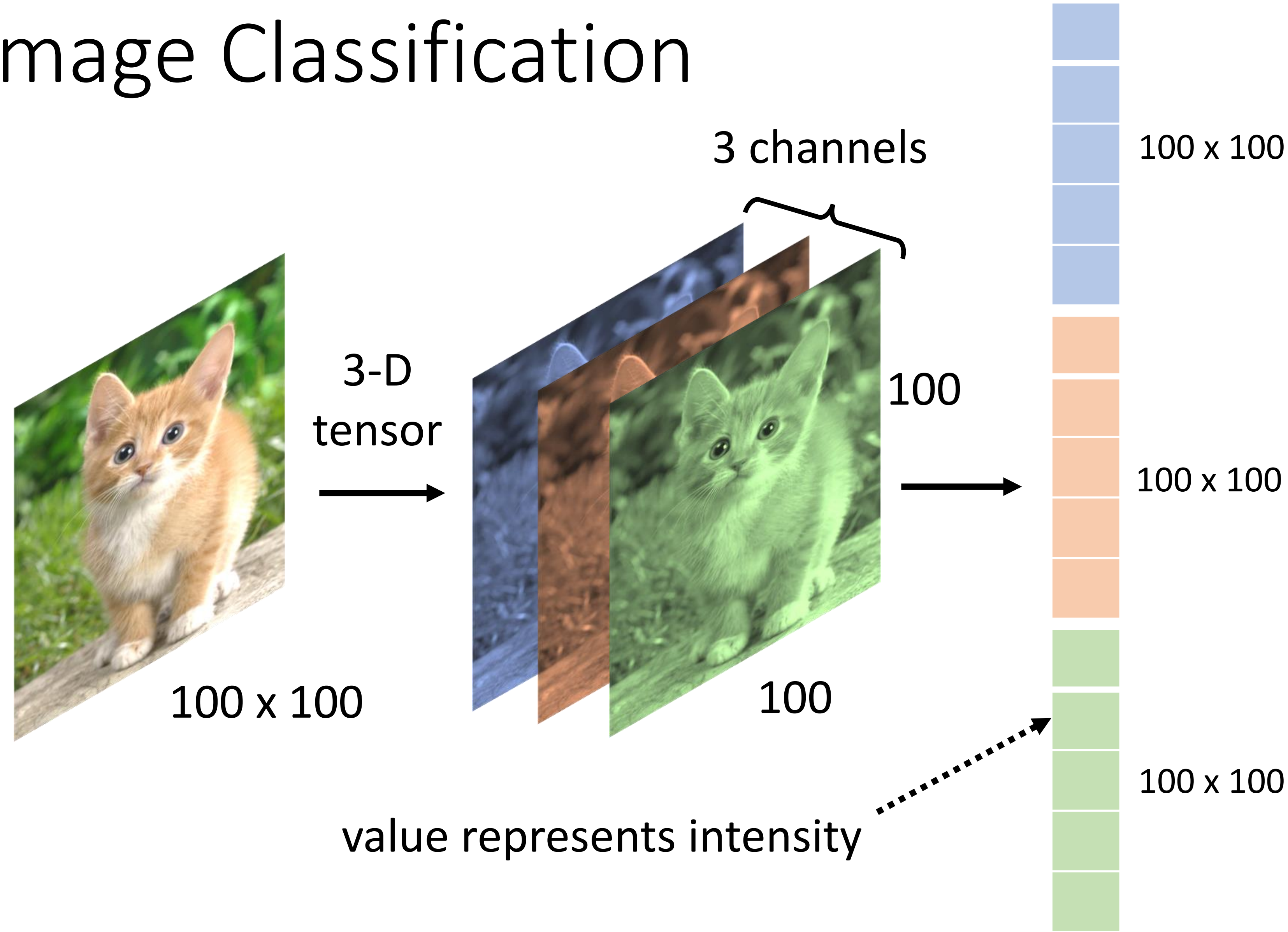
$$\boldsymbol{y}' = \begin{bmatrix} \vdots \\ 0.2 \\ 0.7 \\ 0.1 \\ \vdots \end{bmatrix} \qquad \hat{\boldsymbol{y}} = \begin{array}{r} \\ \text{dog} \\ \text{cat} \\ \text{tree} \\ \\ \end{array} \begin{bmatrix} \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \end{bmatrix}$$

$\boldsymbol{y}' \longleftrightarrow \hat{\boldsymbol{y}}$

Cross entropy

(All the images to be classified have the same size.)

Image Classification



3 channels

3-D tensor

100 x 100

100

100

100 x 100

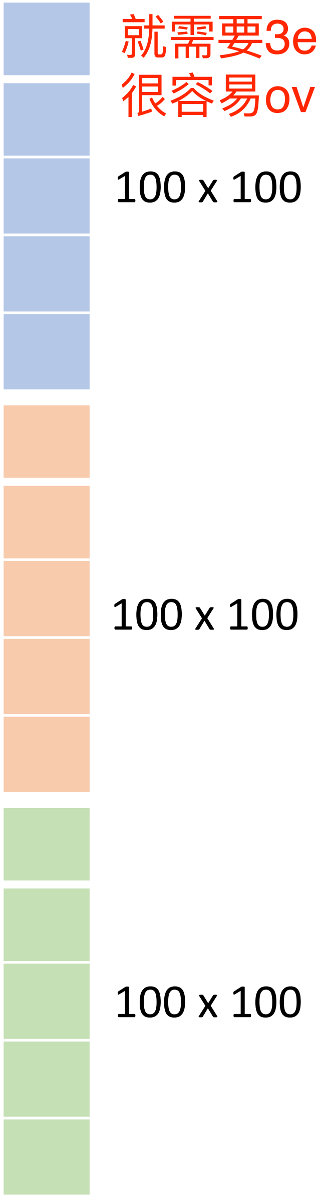
100 x 100

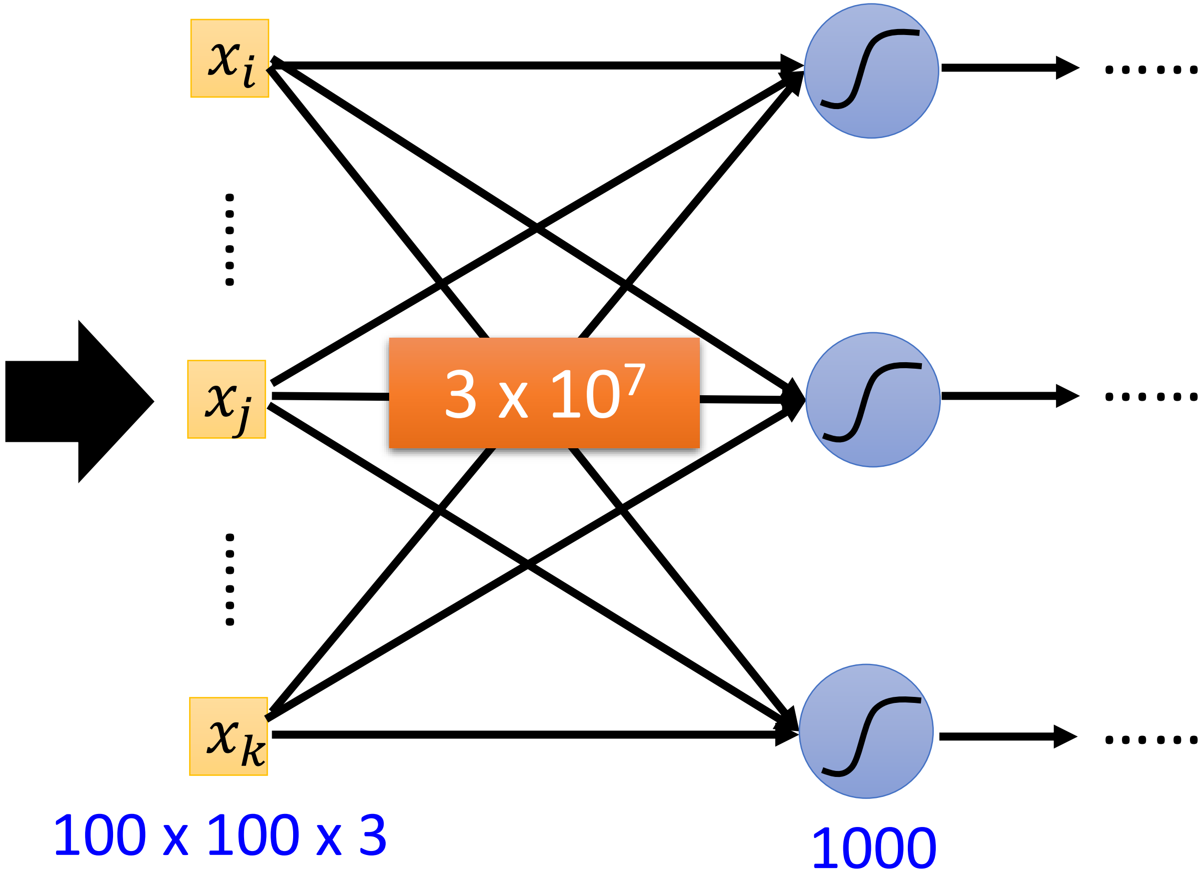
100 x 100

value represents intensity

若一層有1000個neuron，
就需要3e7個參數
很容易overfitting

Fully Connected Network

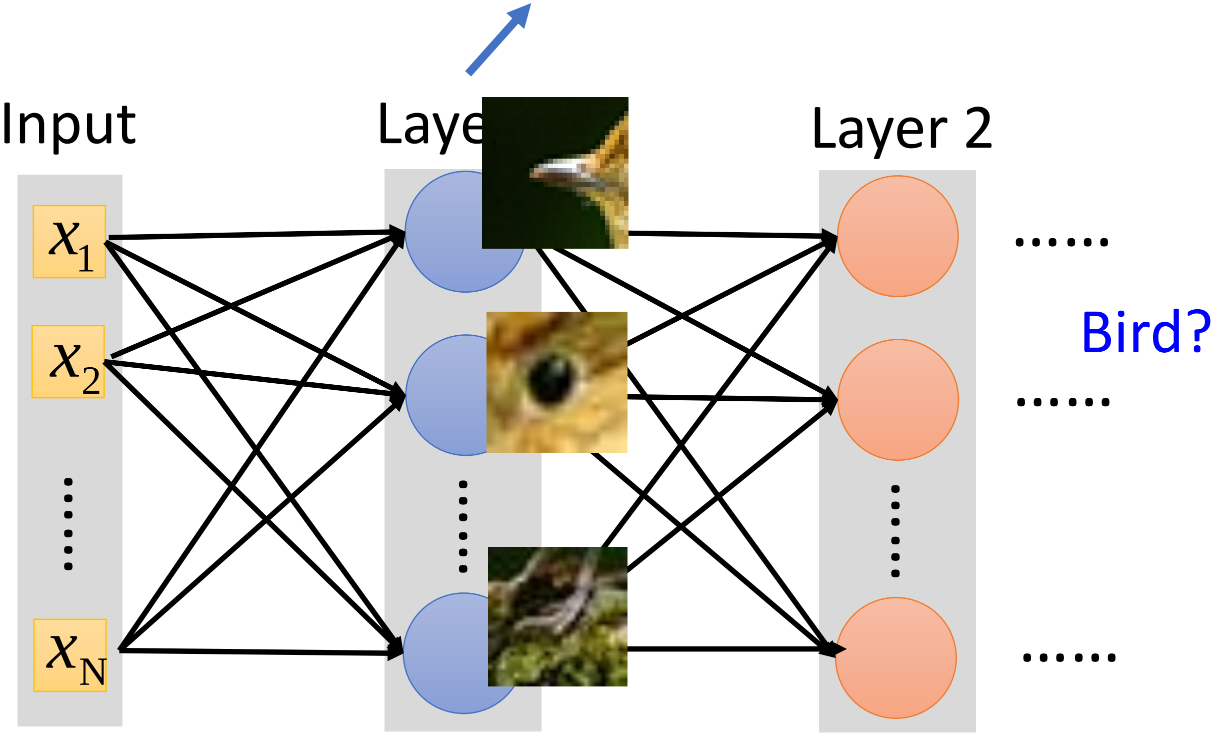




Do we really need "*fully connected*" in image processing?

Observation 1

Identifying some critical patterns



Perhaps human also identify birds in a similar way … ☺

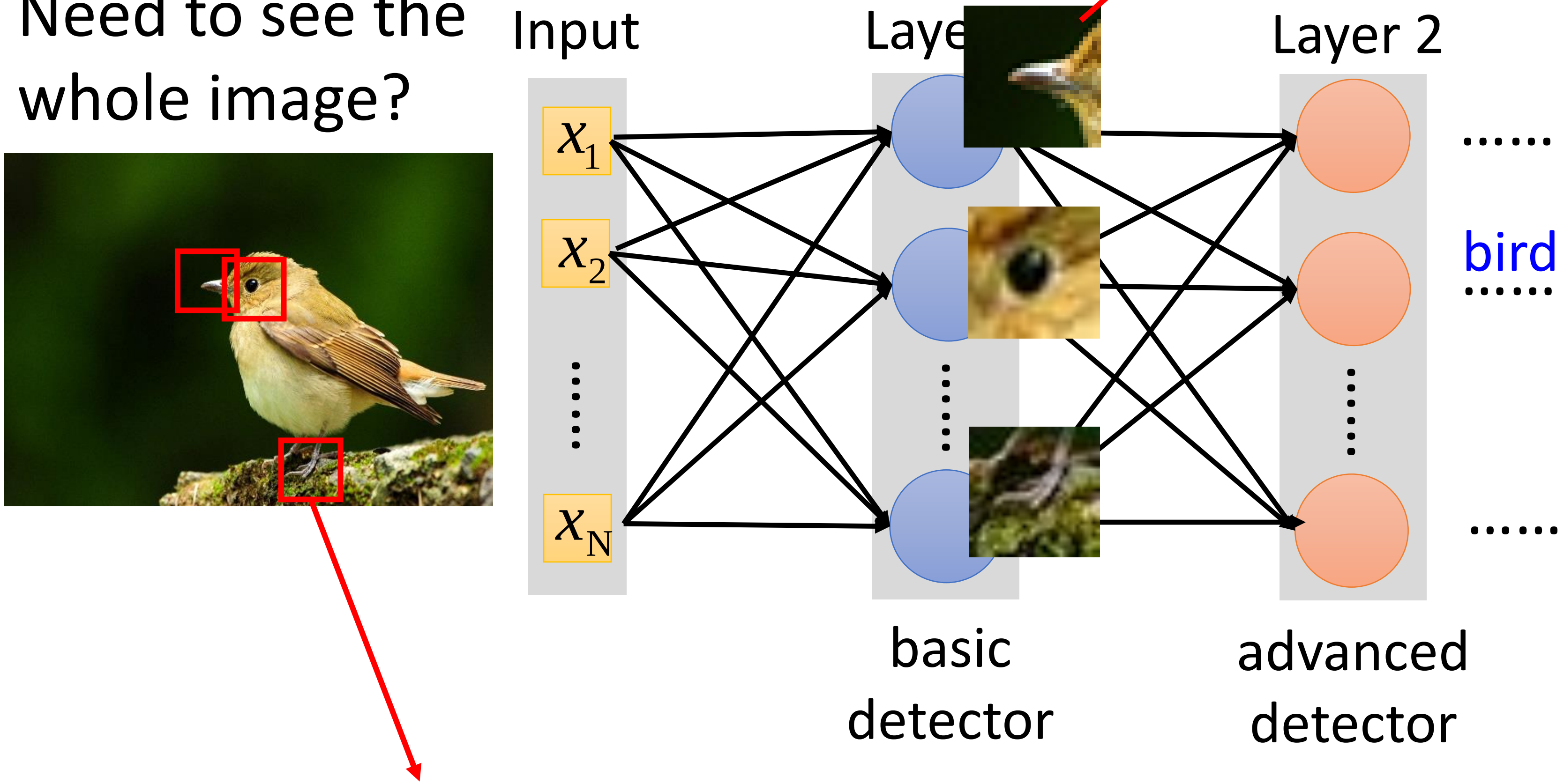
如果你看到這隻鳥，你就應該放下
酒杯了！因為這是隻貓！

https://www.dcard.tw/f/funny/p/233833012

Observation 1

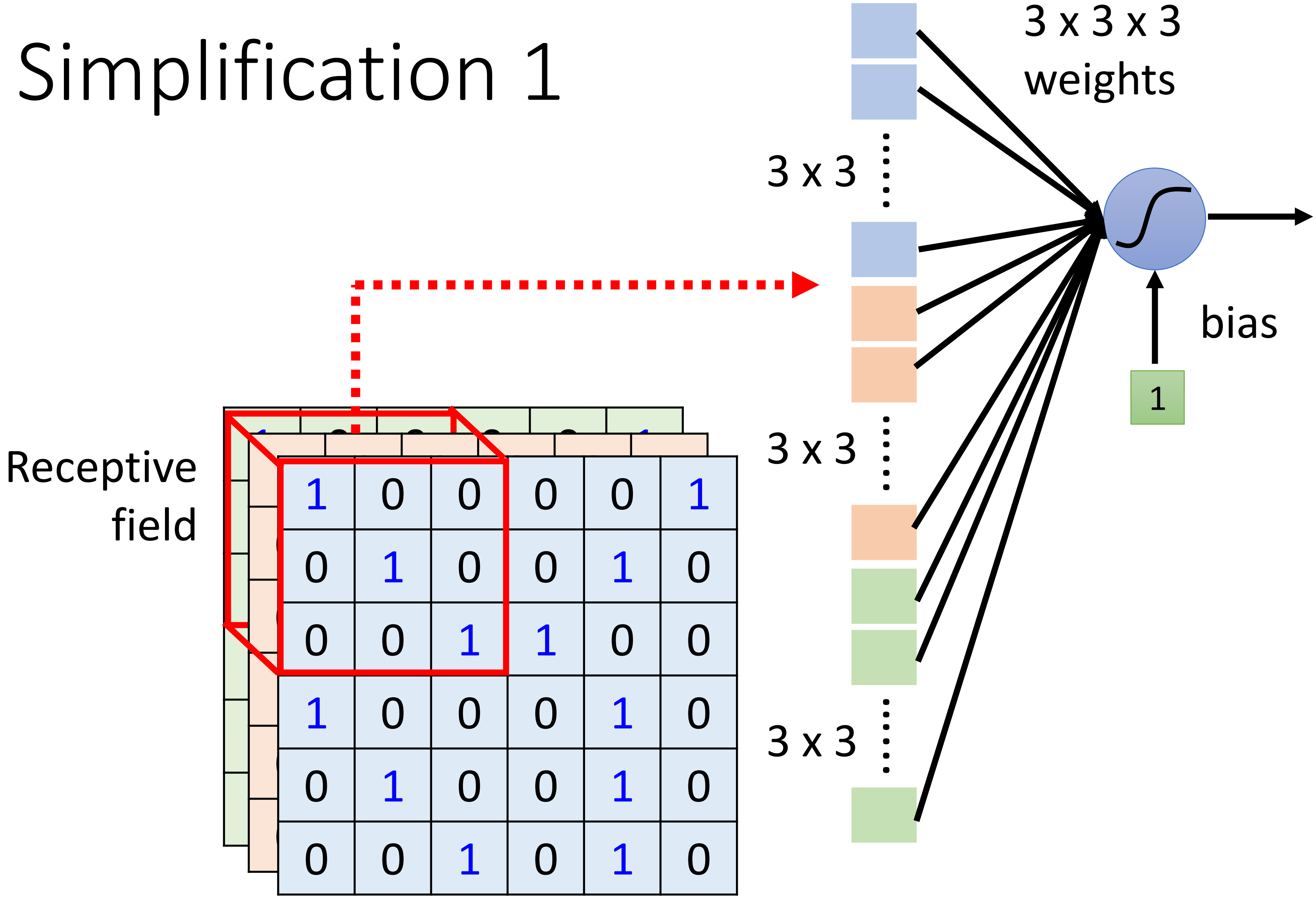
A neuron does not have to see the whole image.

Need to see the whole image?



Some patterns are much smaller than the whole image.

Simplification 1

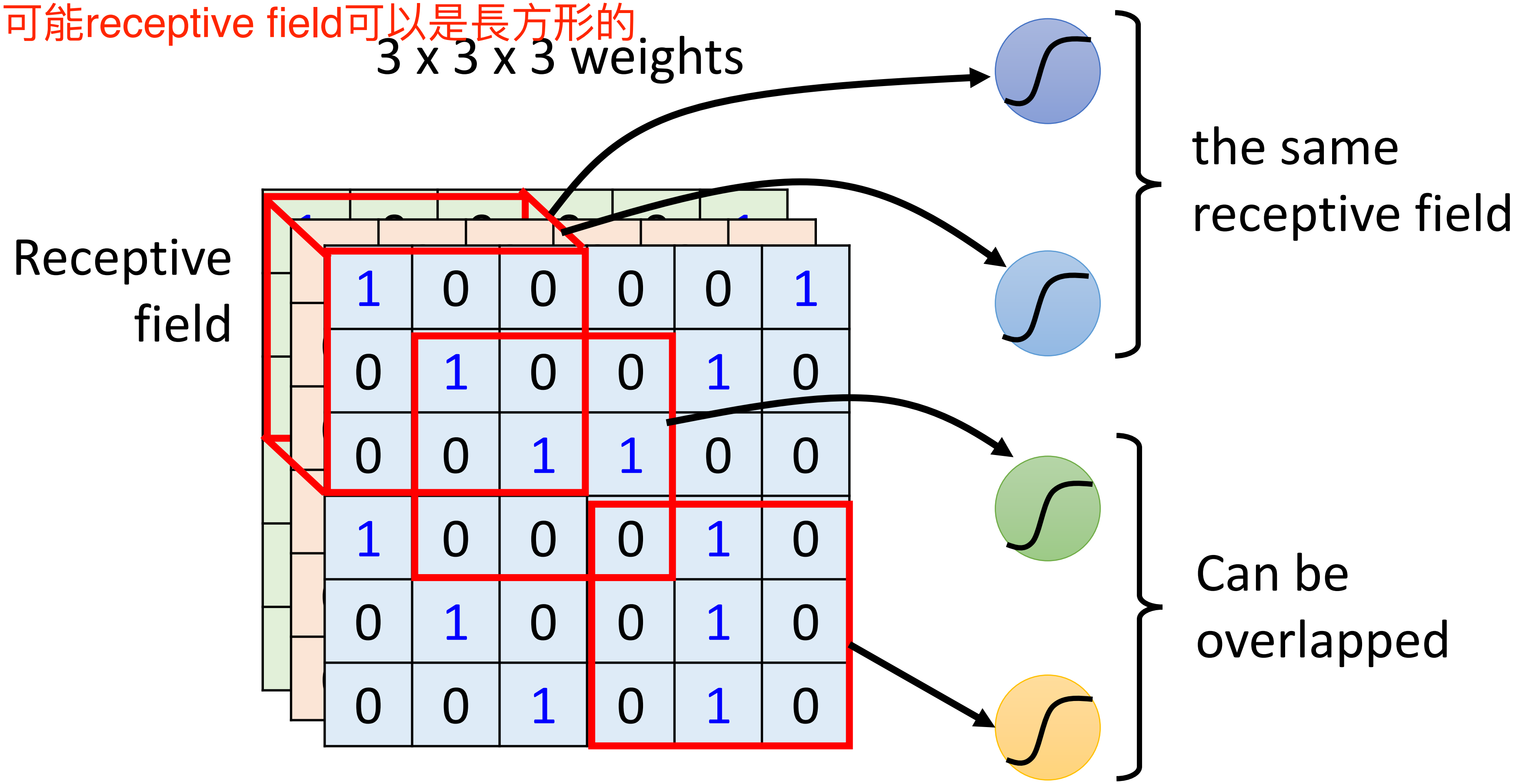


1	0	0	0	0	1
0	1	0	0	1	0
0	0	1	1	0	0
1	0	0	0	1	0
0	1	0	0	1	0
0	0	1	0	1	0

Simplification 1

架構的設計跟問題本身有關
有可能不一定要把rbg都計算
也有可能同一個receptive field可以接兩個以上的neuron
也有可能receptive field可以是長方形的

- Can different neurons have different sizes of receptive field?
- Cover only some channels?
- Not square receptive field?

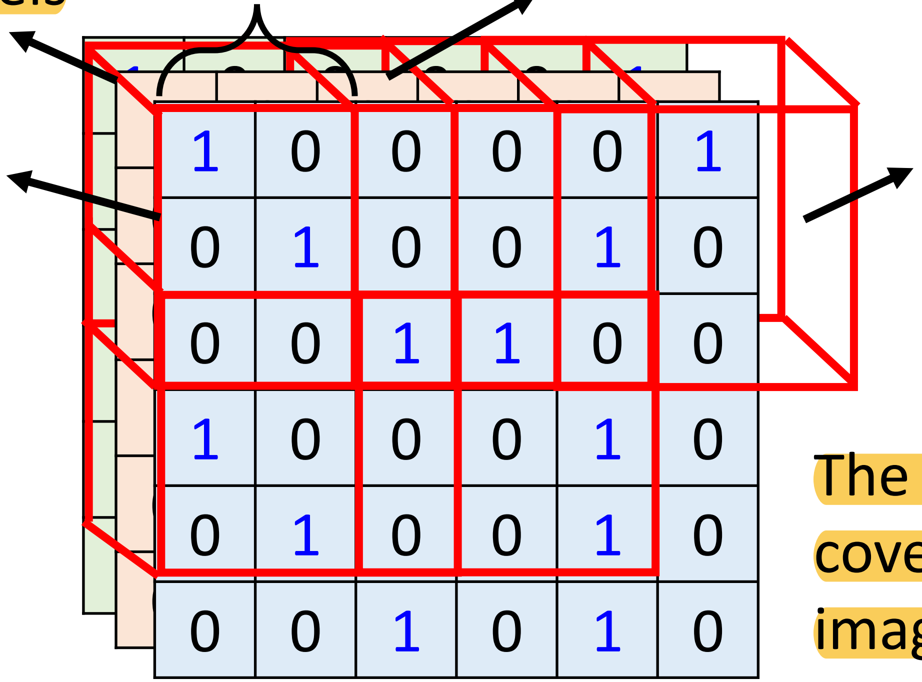


1	0	0	0	0	1
0	1	0	0	1	0
0	0	1	1	0	0
1	0	0	0	1	0
0	1	0	0	1	0
0	0	1	0	1	0

Simplification 1 – Typical Setting

經典的CNN架構

Each receptive field has a set of neurons (e.g., 64 neurons).



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

all channels

stride = 2

overlap

kernel size

(e.g., 3 x 3)

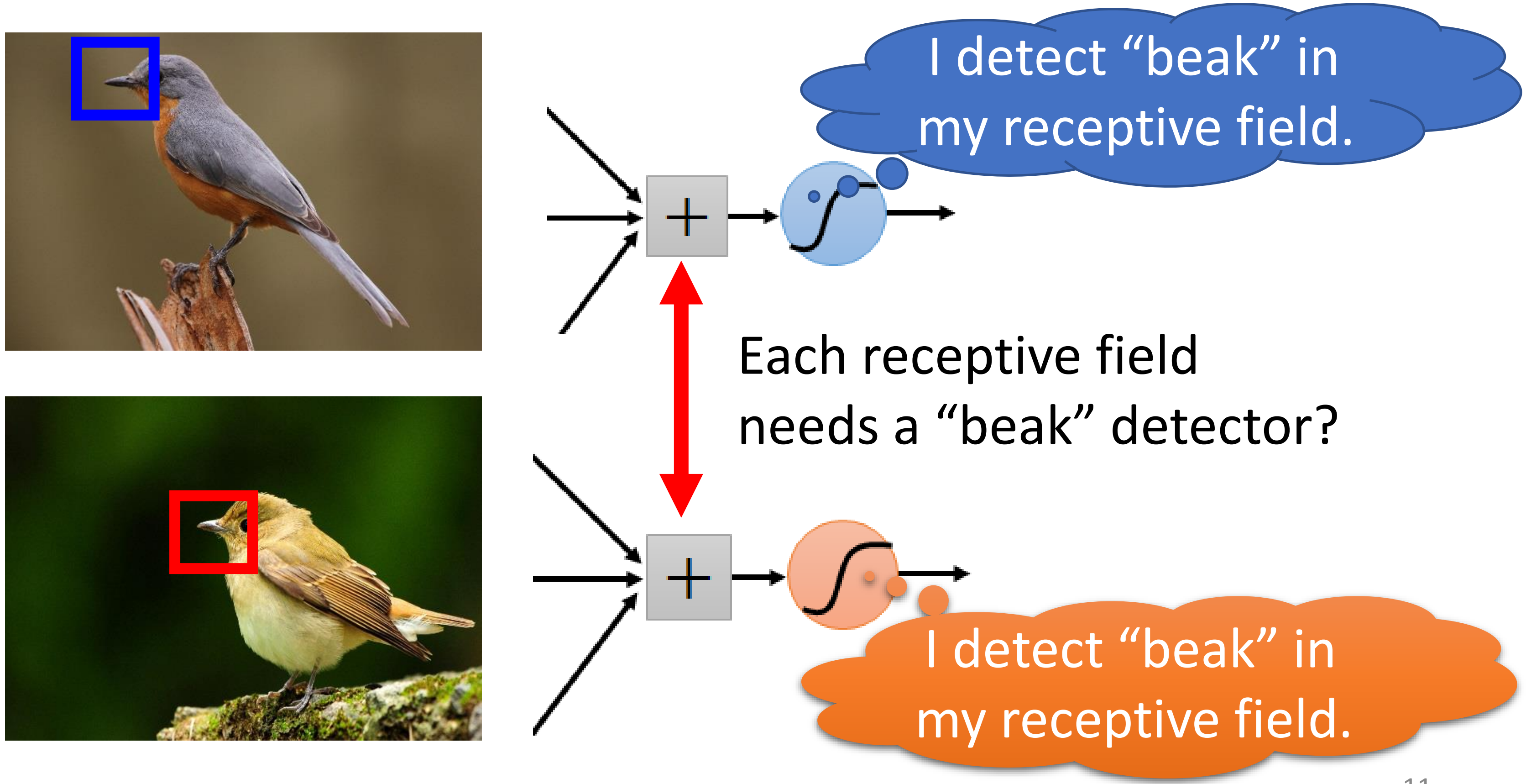
padding

The receptive fields cover the whole image.

Observation 2

每一個receptive field在做的事情都很相近

- The same patterns appear in different regions.

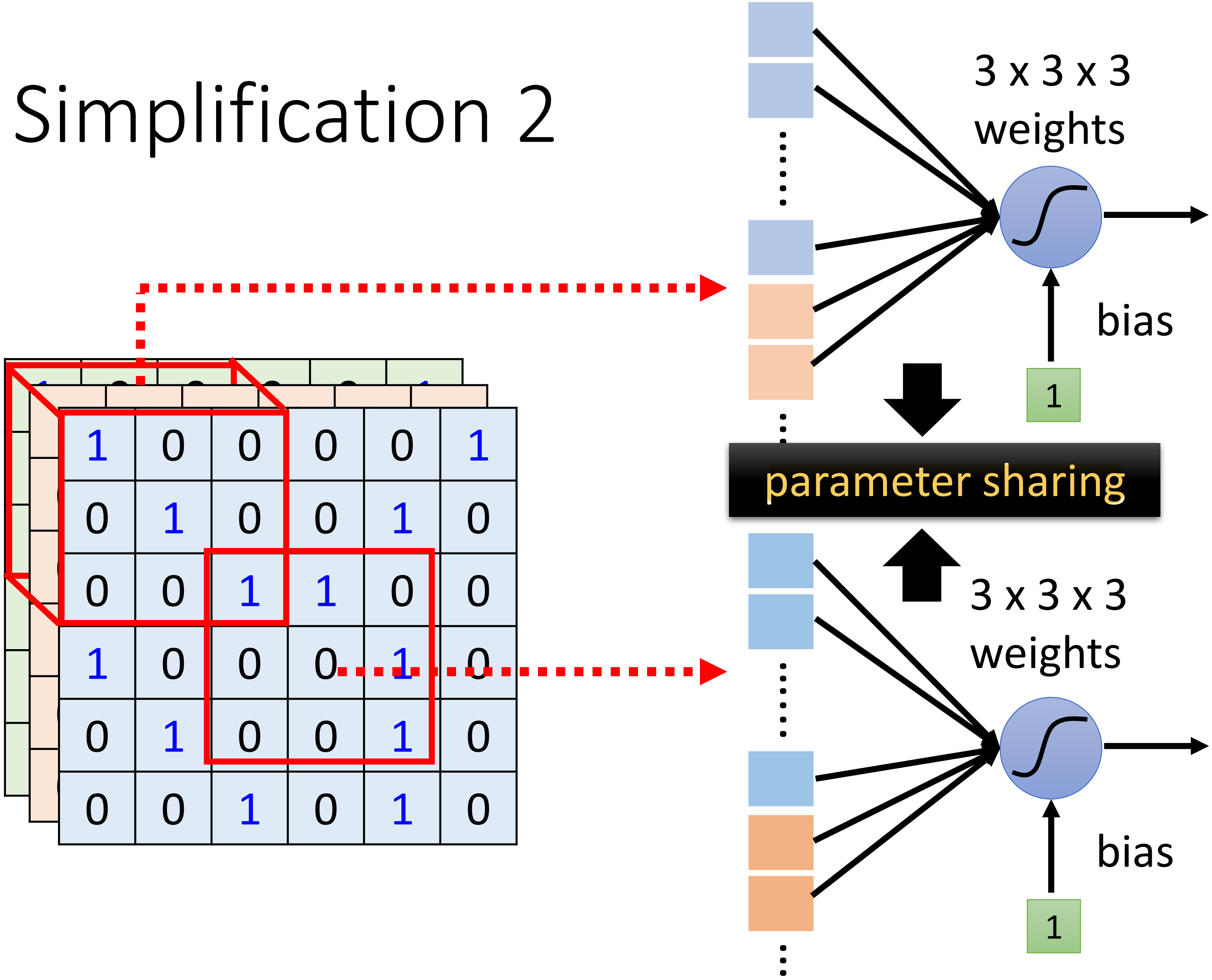


I detect “beak” in my receptive field.

Each receptive field needs a “beak” detector?

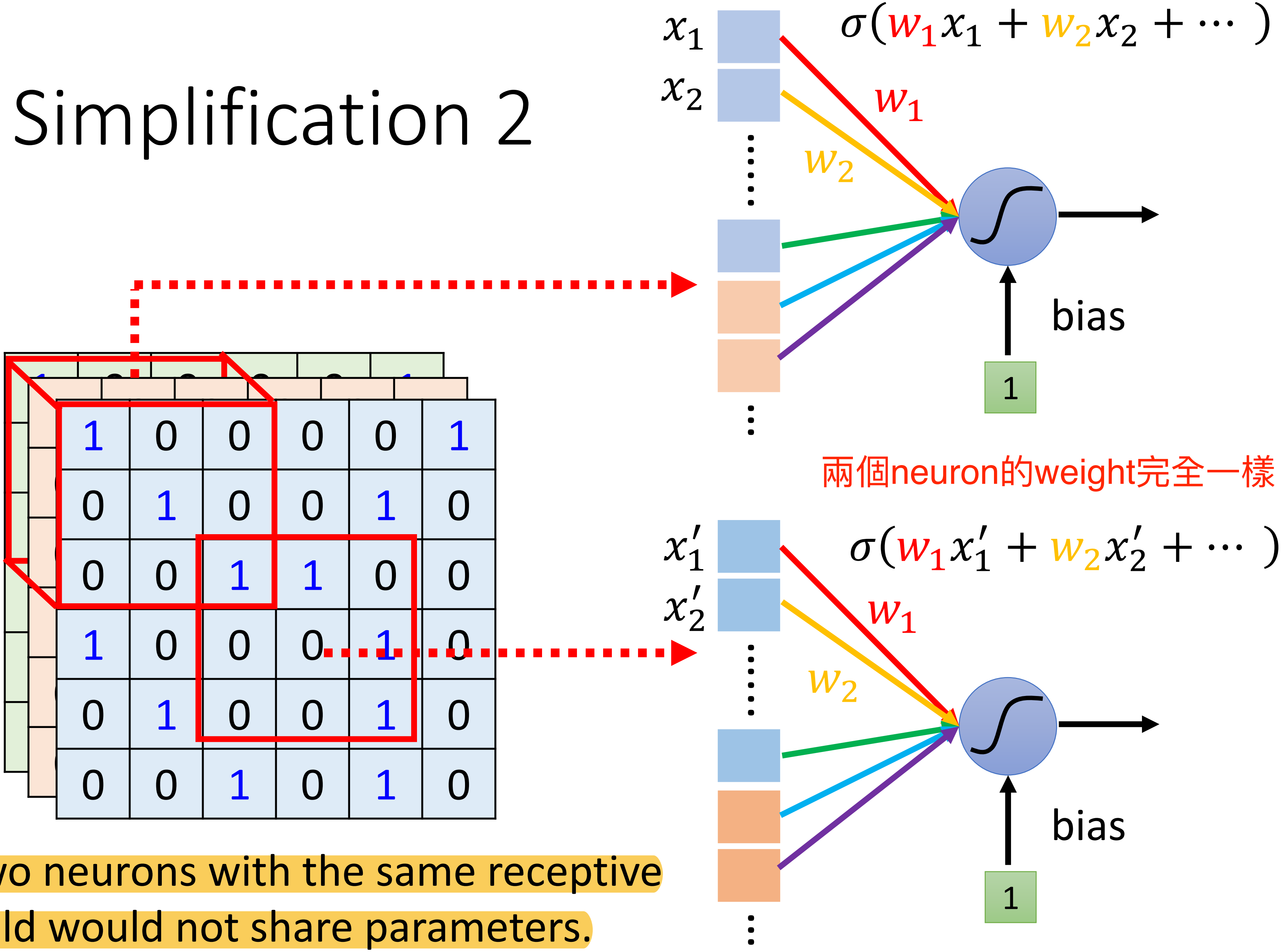
I detect “beak” in my receptive field.

Simplification 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

Simplification 2


x_1
x_2
$\sigma(w_1 x_1 + w_2 x_2 + \cdots)$
w_1
w_2
bias
1
x_1'
x_2'
$\sigma(w_1 x_1' + w_2 x_2' + \cdots)$
w_1
w_2
bias
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

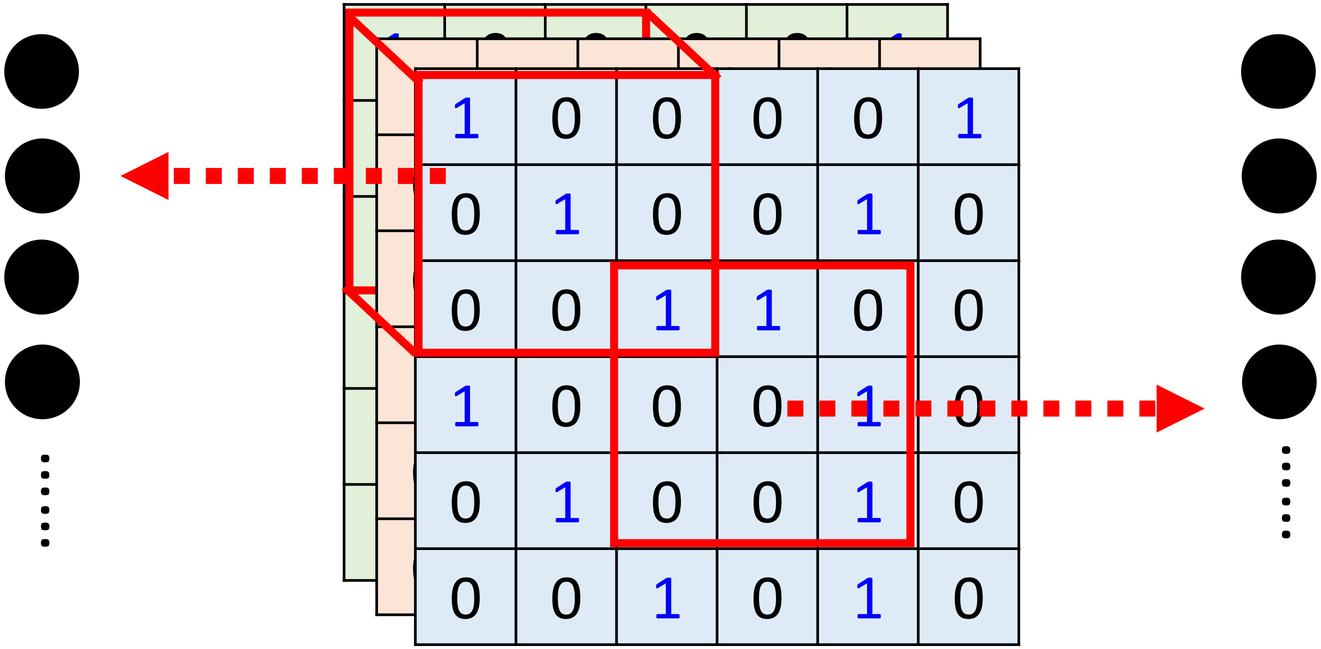
兩個neuron的weight完全一樣

Two neurons with the same receptive field would not share parameters.

因為這樣輸出就都會是一樣的，沒有必要

Simplification 2 – Typical Setting

Each receptive field has a set of neurons (e.g., 64 neurons).

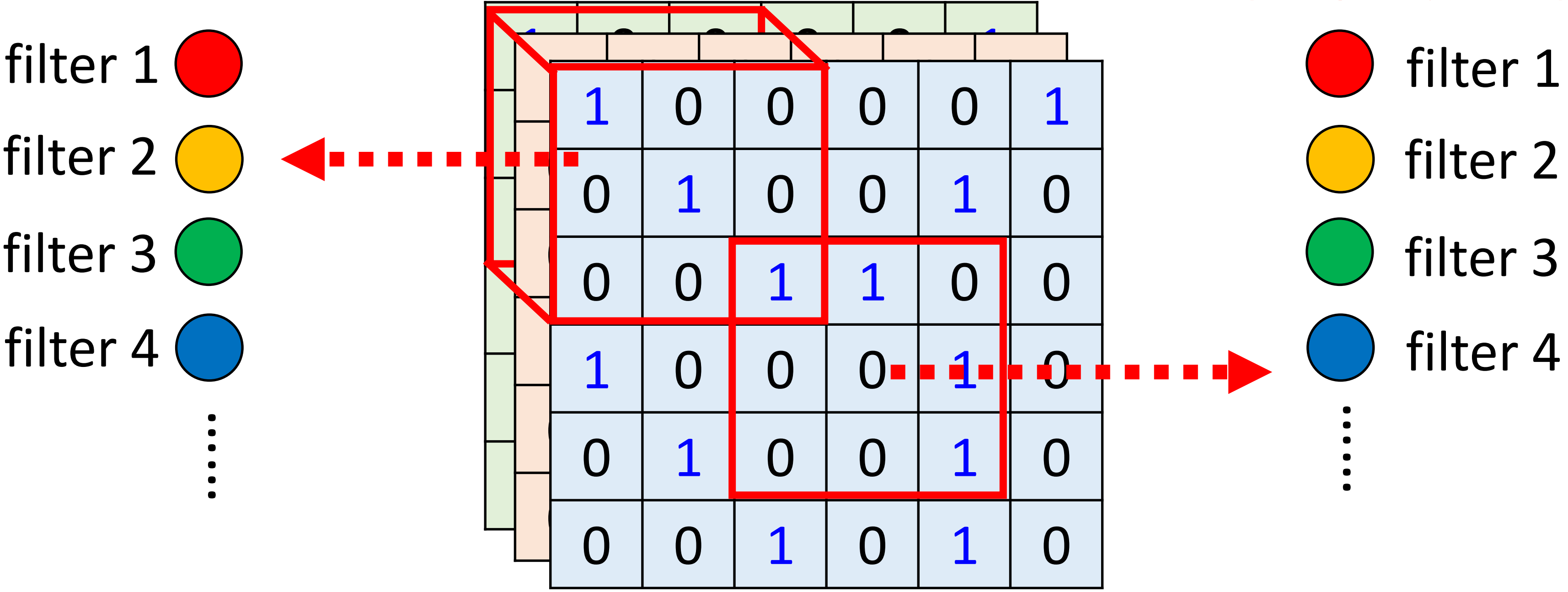


1	0	0	0	0	1
0	1	0	0	1	0
0	0	1	1	0	0
1	0	0	0	1	0
0	1	0	0	1	0
0	0	1	0	1	0

Simplification 2 – Typical Setting

Each receptive field has a set of neurons (e.g., 64 neurons).

Each receptive field has the neurons with the same set of parameters.

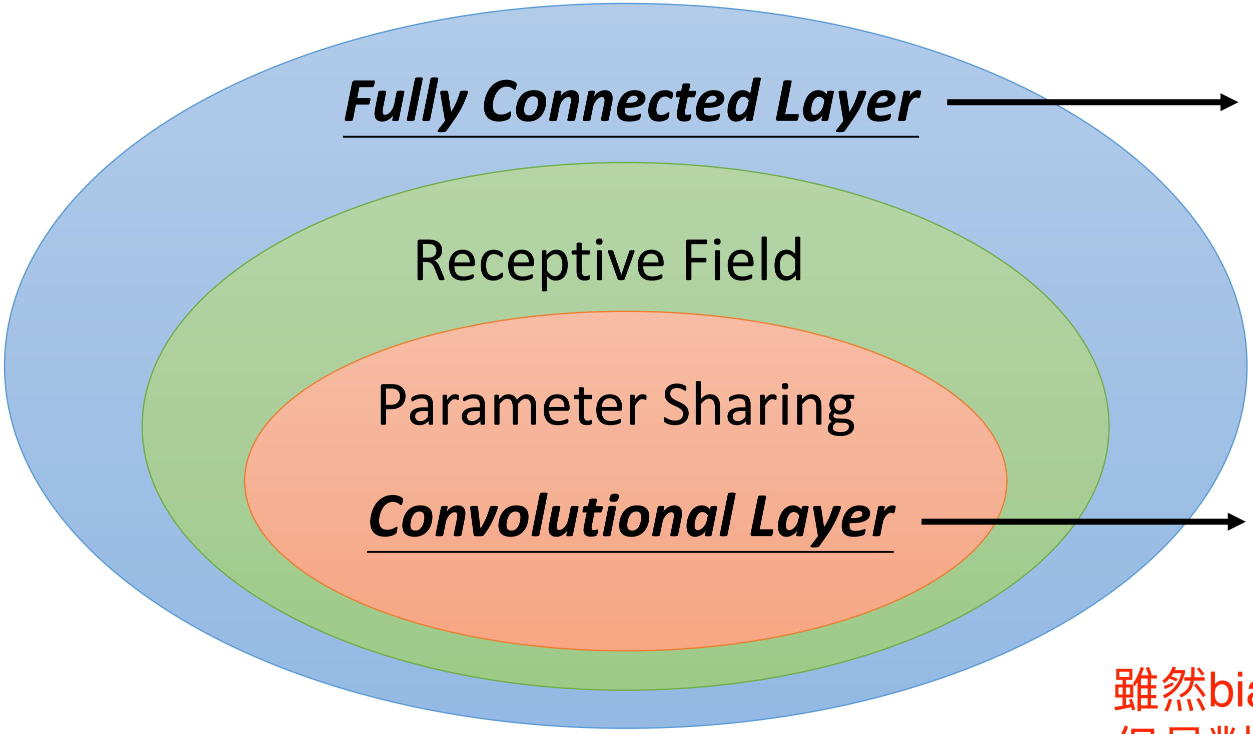


顏色一樣就代表參數一樣

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

Benefit of Convolutional Layer

三種function set是包含關係



Jack of all trades, master of none

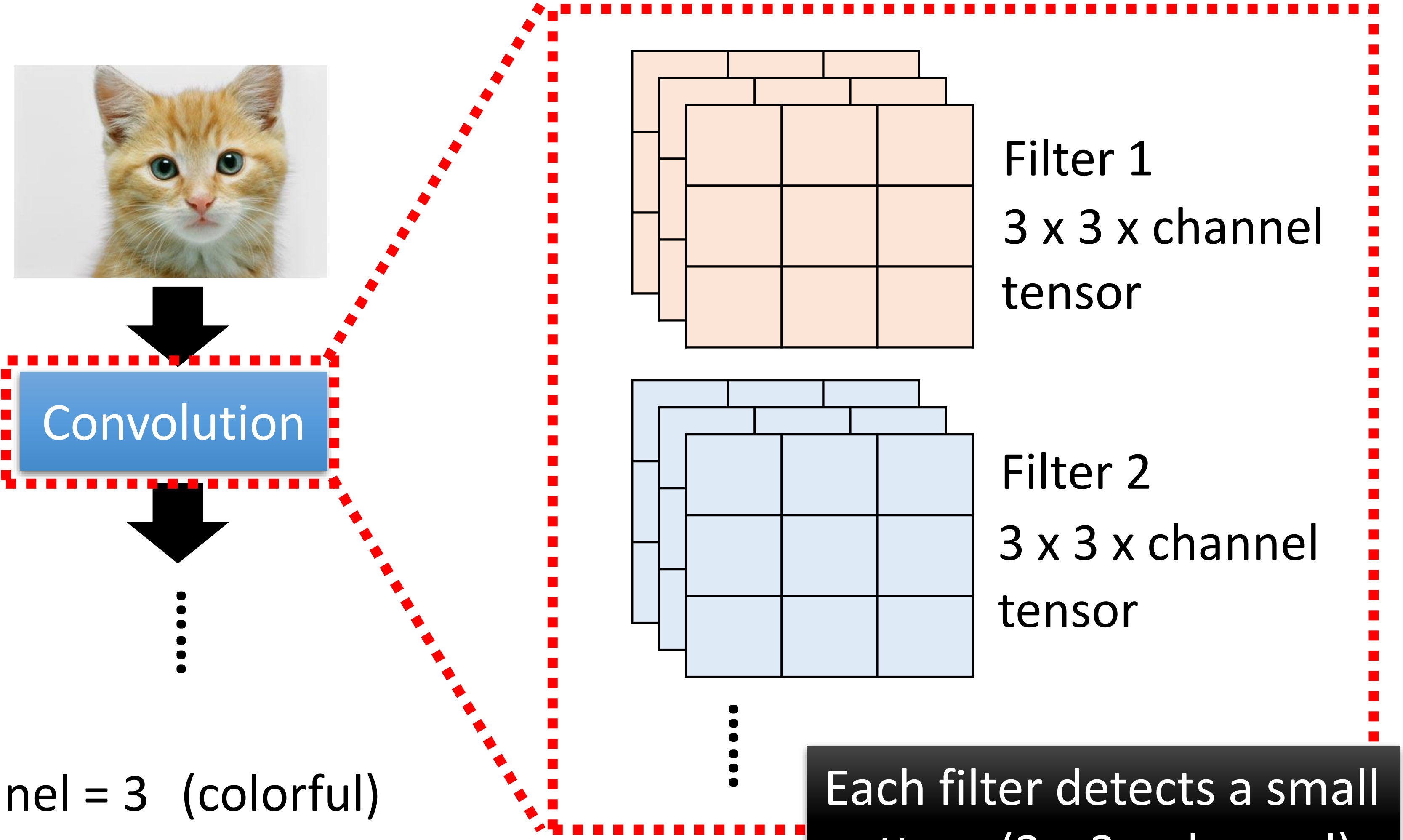
Larger model bias (for image)

雖然bias比較大
但是對於image的處理並無影響

- Some patterns are much smaller than the whole image.
- The same patterns appear in different regions.

Another story based on ***filter*** ☺

Convolutional Layer



channel = 3 (colorful)

channel = 1 (black and white)

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

Convolutional Layer

Consider channel = 1
(black and white 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

6 x 6 image

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

Filter 1

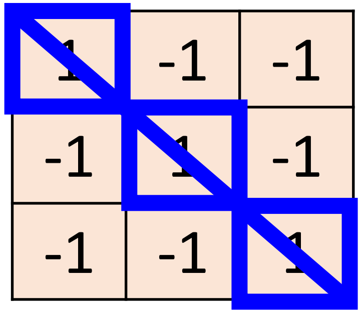
-1	1	-1
-1	1	-1
-1	1	-1

Filter 2

⋮

(The values in the filters
are unknown parameters.)

Convolutional Layer

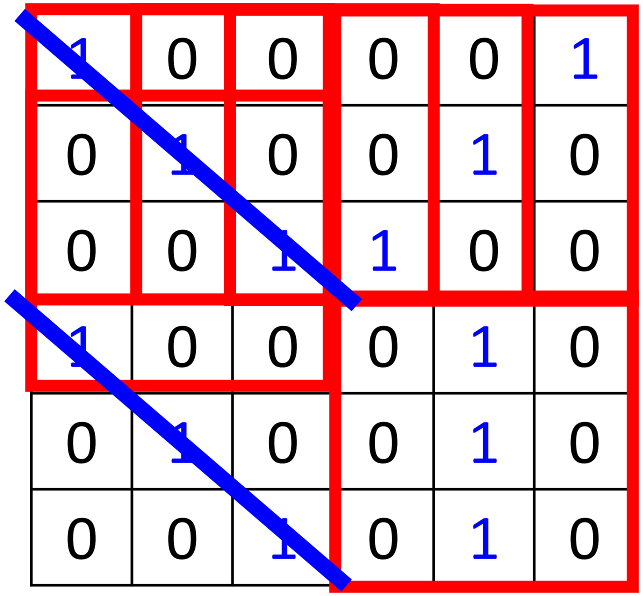


1	-1	-1
-1	1	-1
-1	-1	1

Filter 1

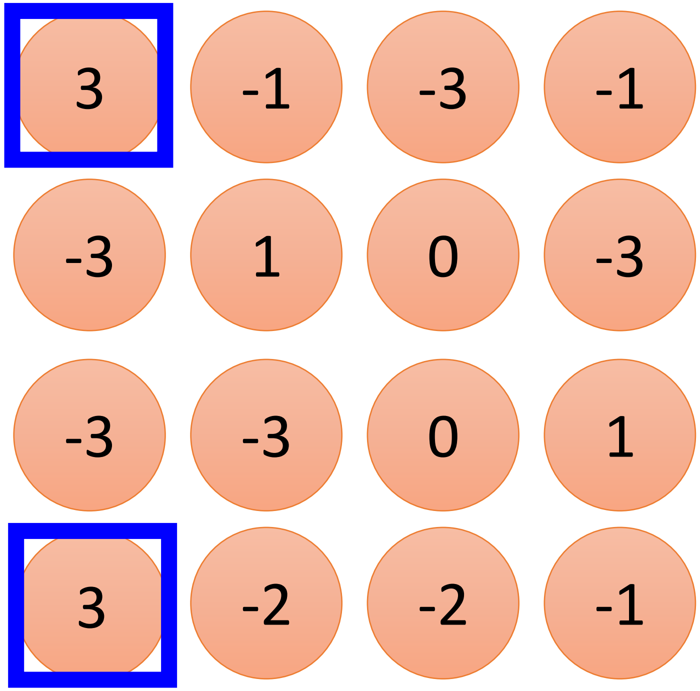
偵測左上到右下的斜線

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

Convolutional Layer

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

Filter 2

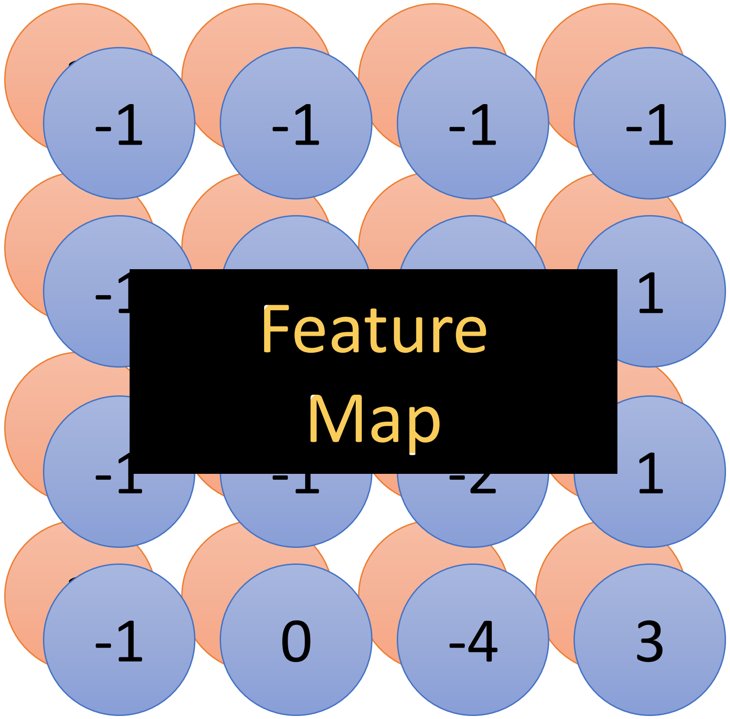
偵測垂直線

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

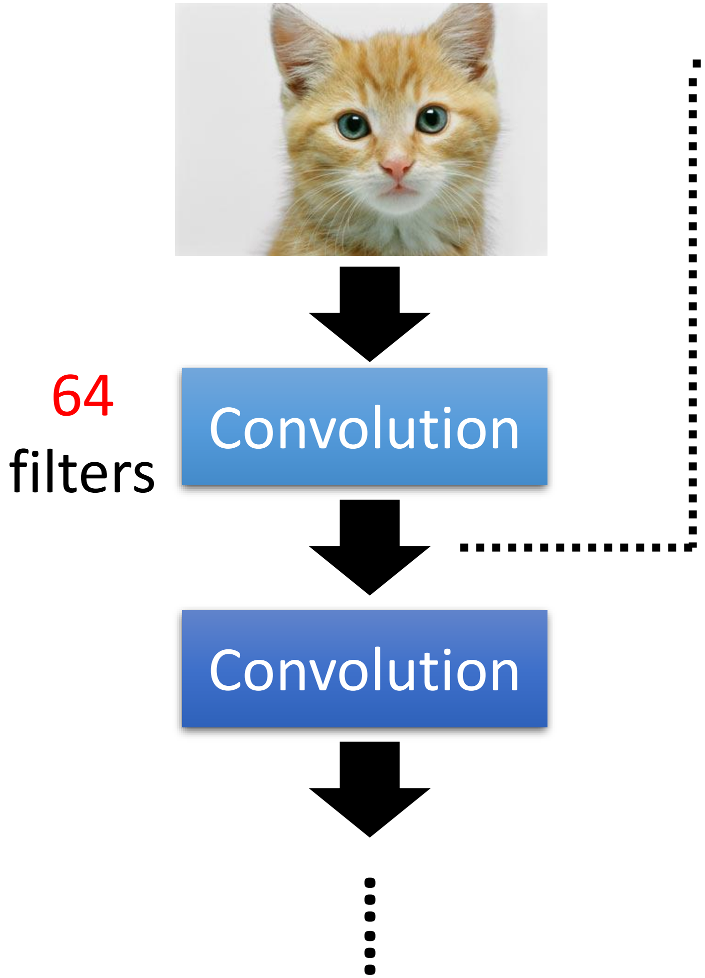
6 x 6 image

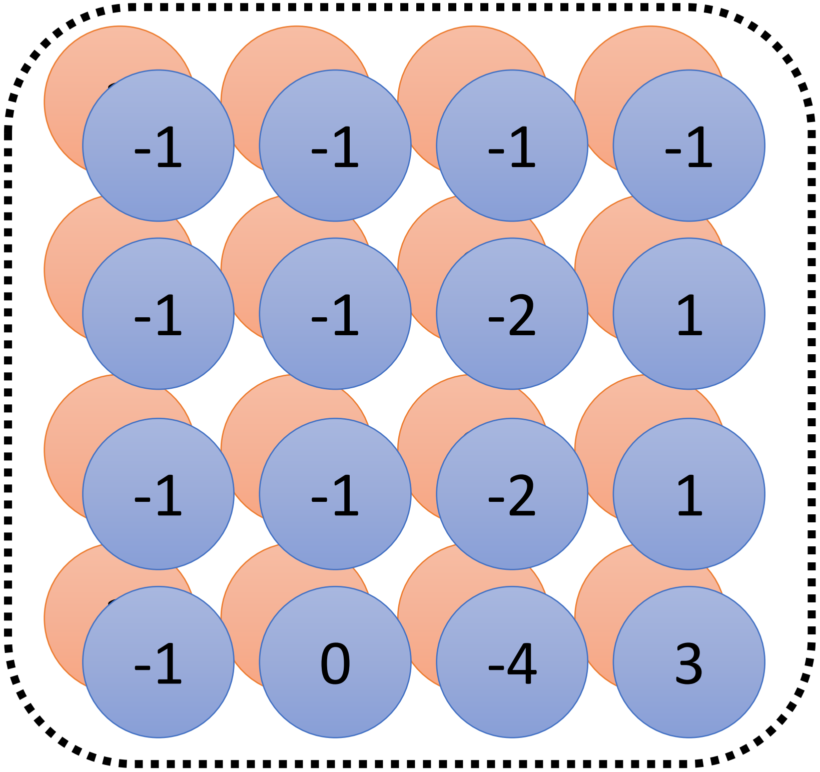
Do the same process for every filter



所有filter偵測出來的值
就稱為feature map

Convolutional Layer





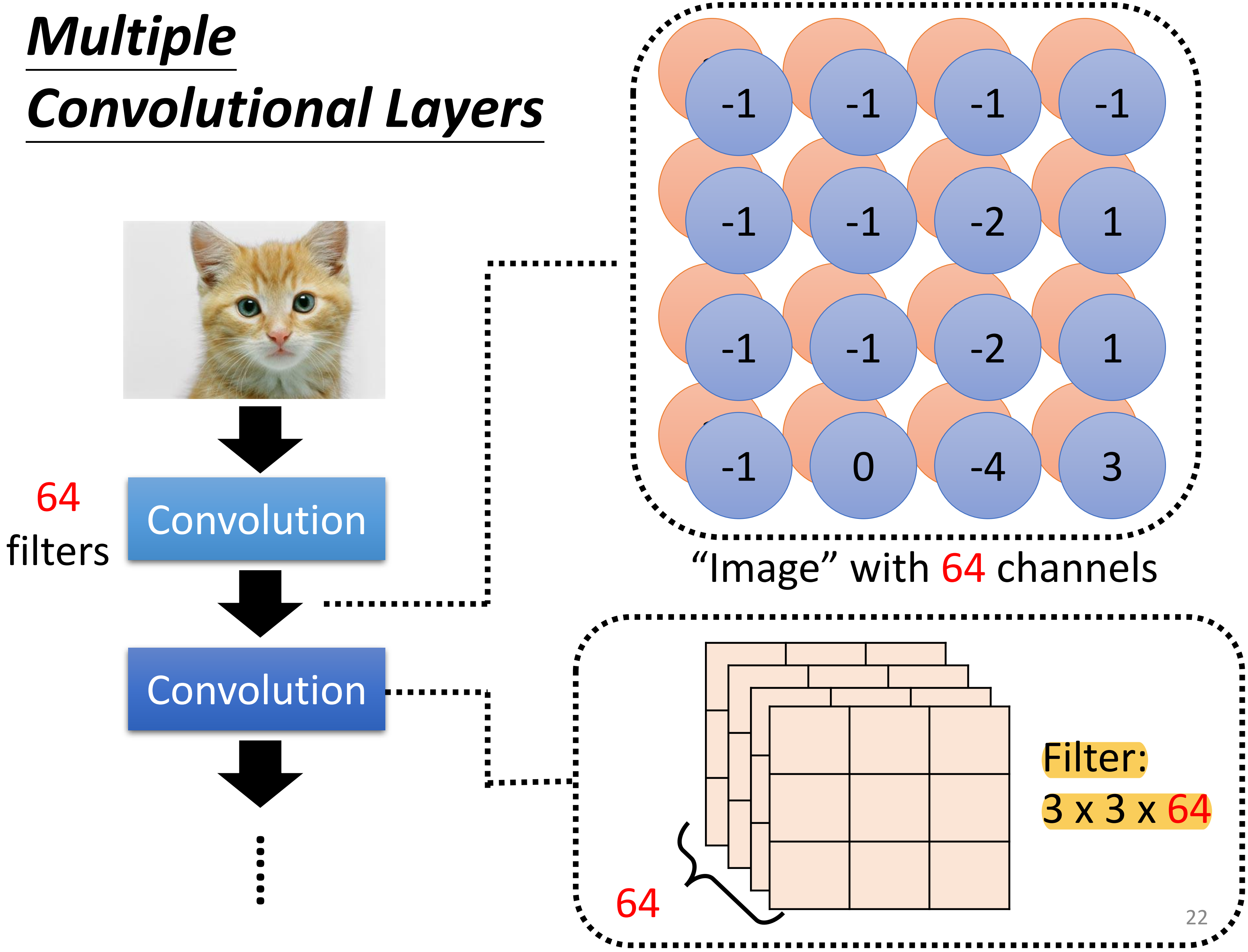
-1	-1	-1	-1
-1	-1	-2	1
-1	-1	-2	1
-1	0	-4	3

“Image” with 64 channels

可以看做是一張新的圖片
而且有64個channels
（如果有64個filters）
（原本的圖片為3個channel：red, green, blue）

而且圖片會變小，接著又可以繼續再做convolution

Multiple Convolutional Layers



64 filters

Convolution

Convolution

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

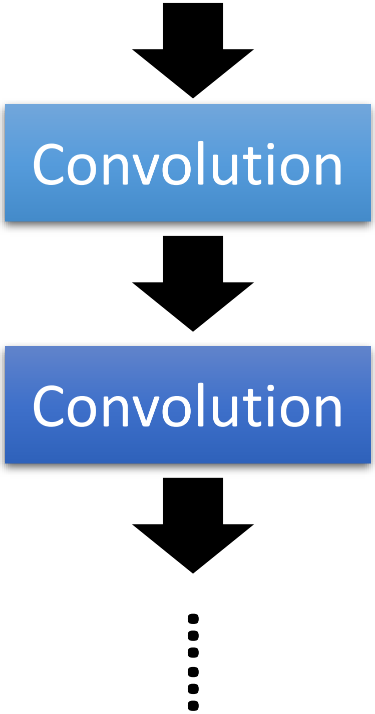
"Image" with 64 channels

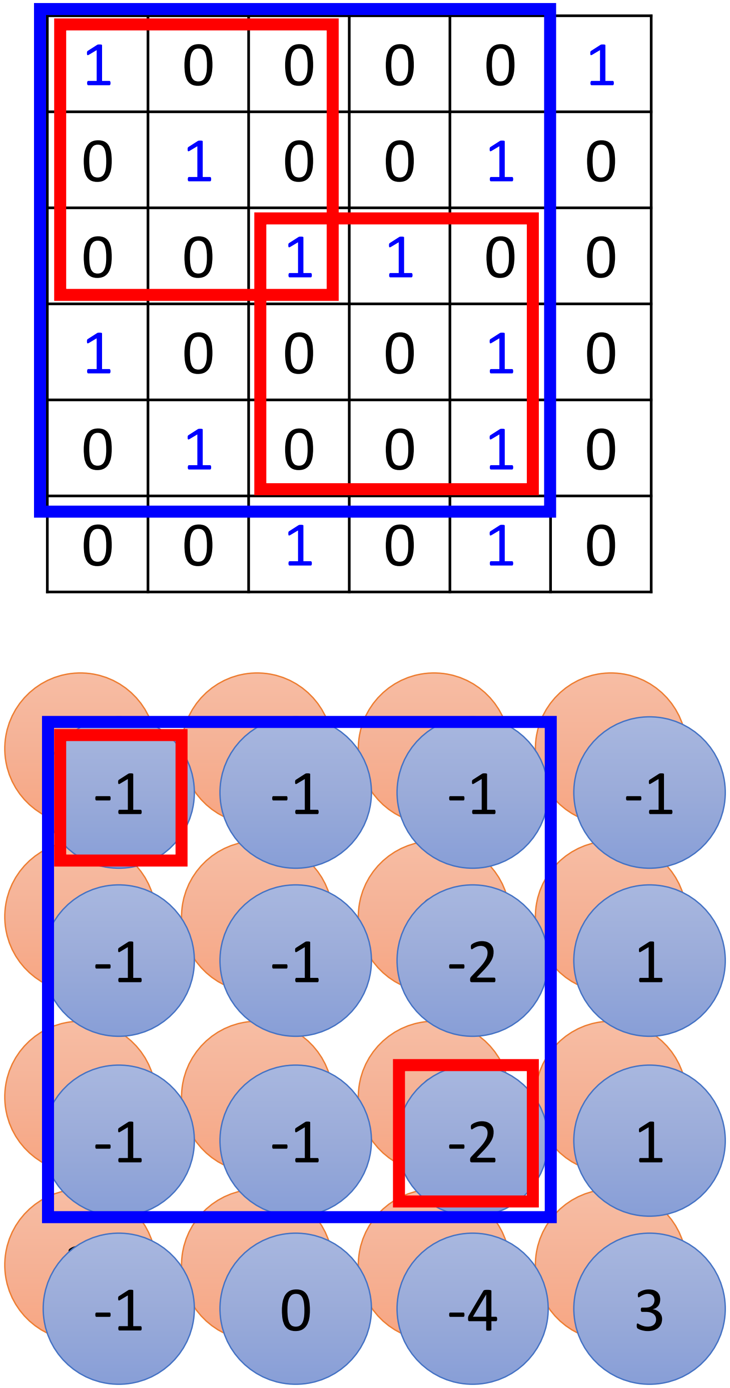
Filter:
3 x 3 x 64

64

Multiple Convolutional Layers

若第二層filter僅有3x3
但是對應於原本的圖片
已經涵蓋5x5了

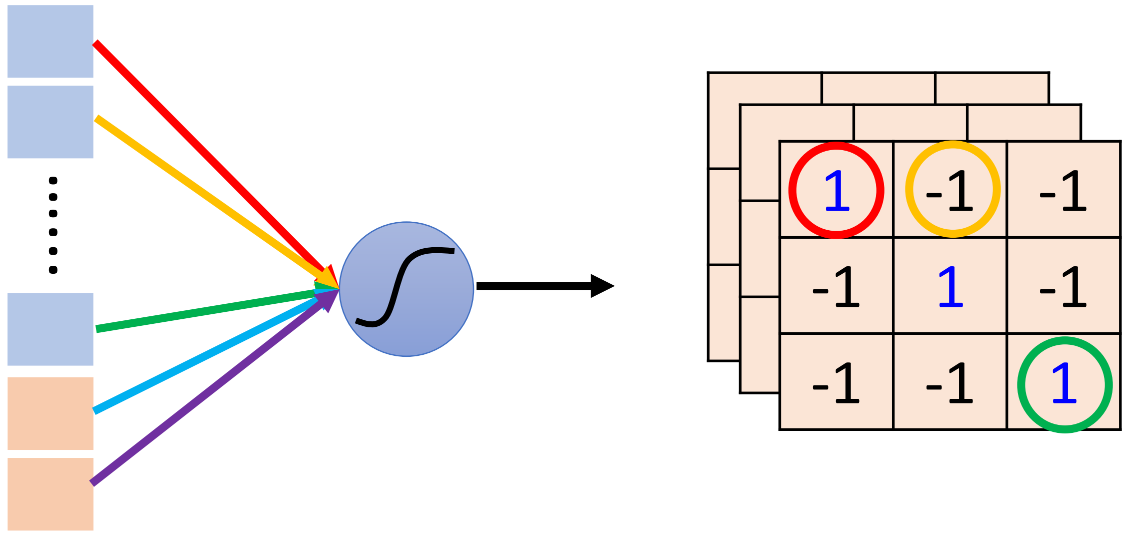

64 filters



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

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

Comparison of Two Stories



Receptive field

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

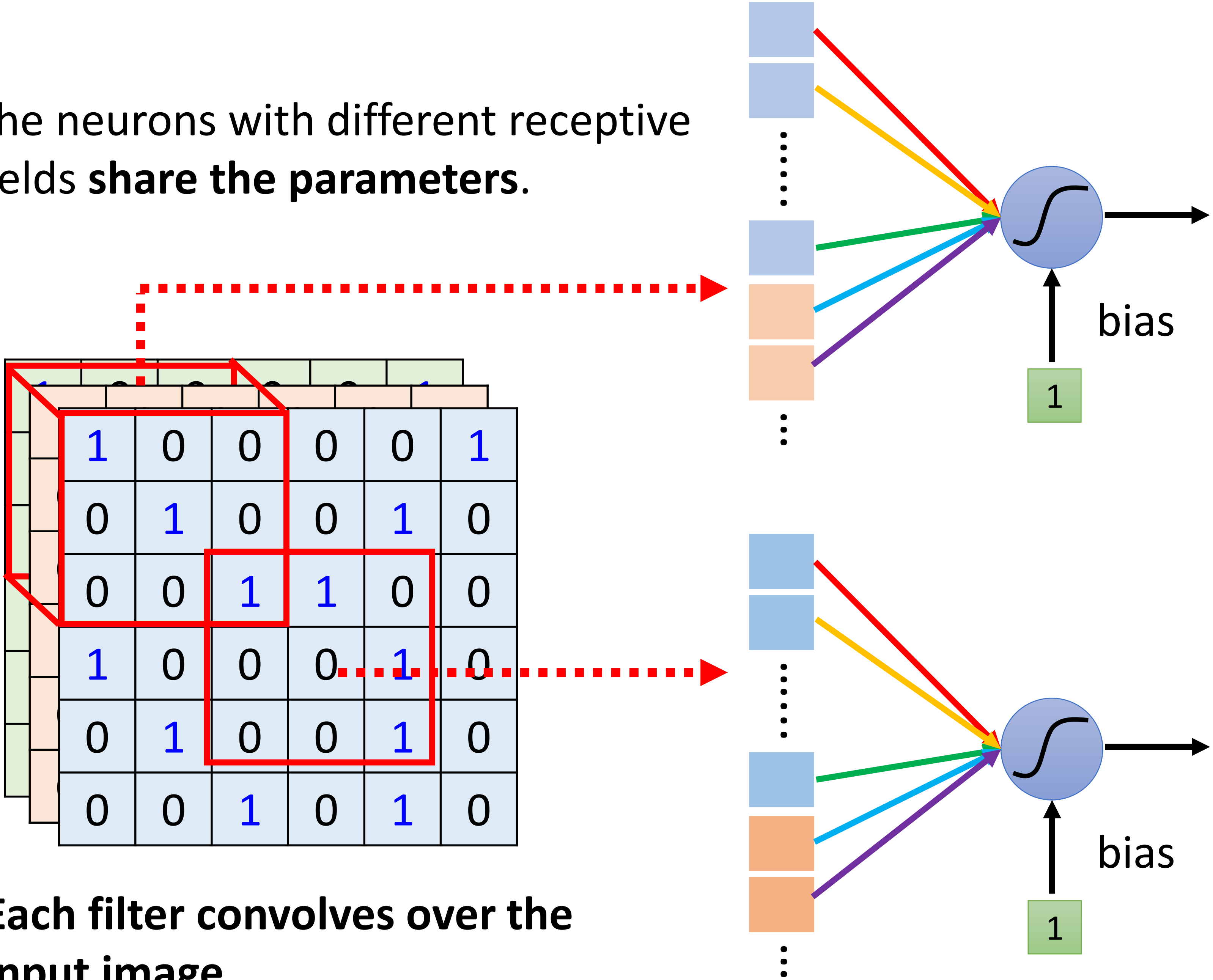
Filter

3 x 3 x channel

tensor

(ignore bias in this slide)

The neurons with different receptive fields **share the parameters**.



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

Each filter convolves over the input image.

Convolutional Layer

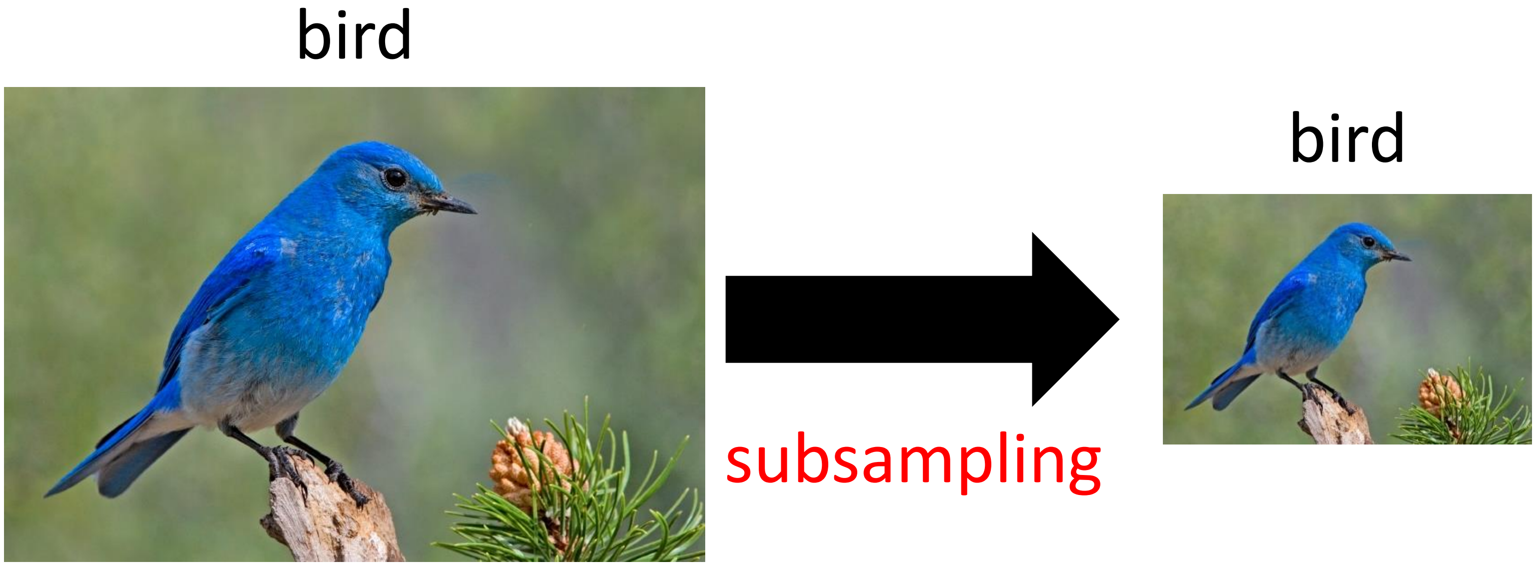
Neuron Version Story	***Filter Version Story***
Each neuron only considers a receptive field.	There are a set of filters detecting small patterns.
The neurons with different receptive fields share the parameters.	Each filter convolves over the input image.

They are the same story.

Observation 3

- Subsampling the pixels will not change the object

例如將原先圖片的偶數列與偶數行拿掉
看起來還是會很像



bird

subsampling

bird

pooling有很多版本，這裡提的是max pooling

Pooling – Max Pooling

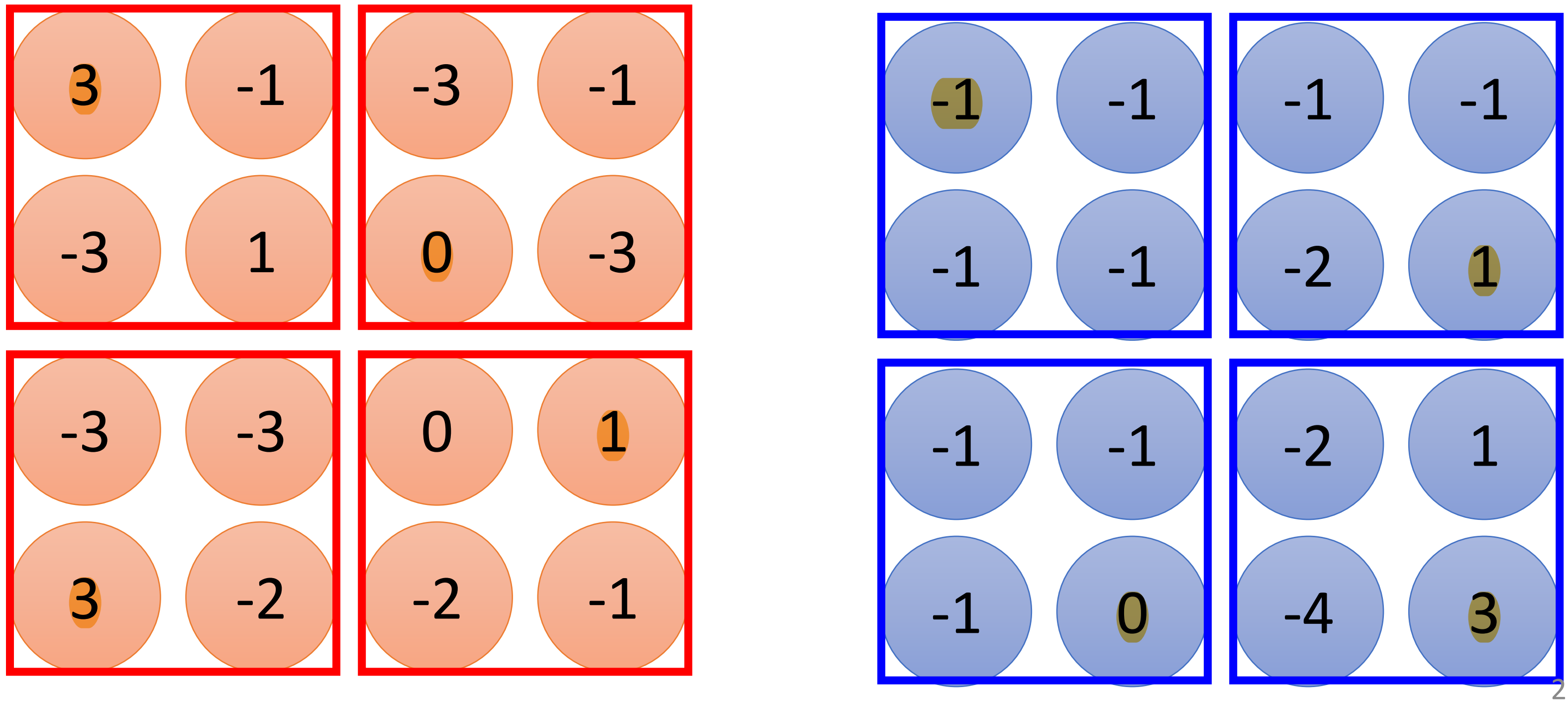
1	-1	-1
-1	1	-1
-1	-1	1

Filter 1

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

Filter 2

max pooling，
就是在kernel size上取最大的當代表

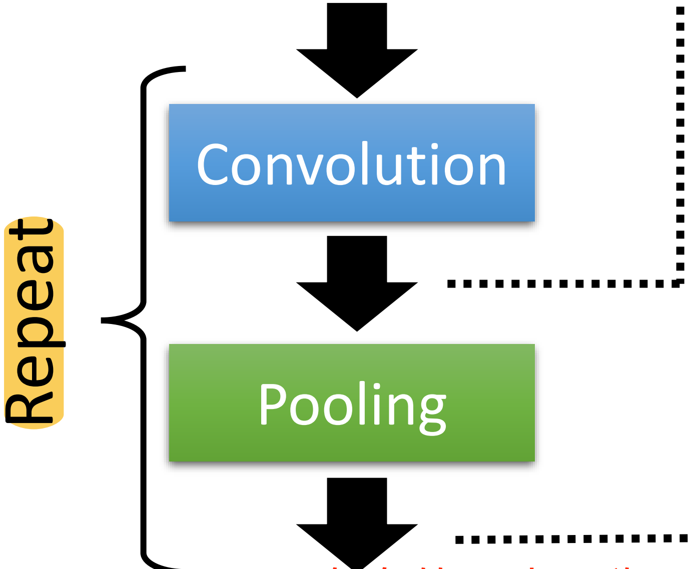


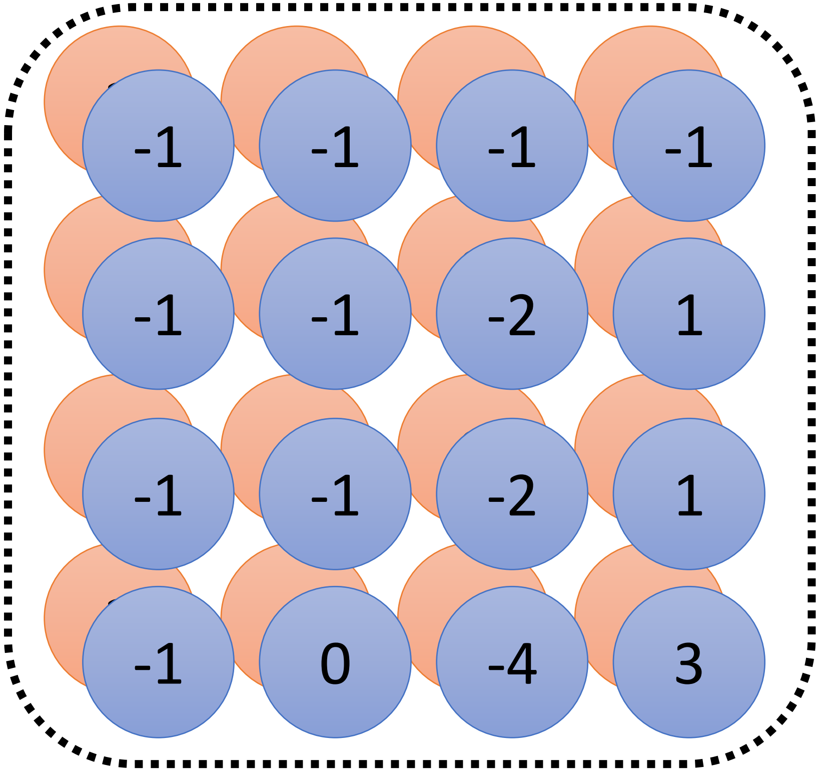
3	-1	-3	-1
-3	1	0	-3
-3	-3	0	1
3	-2	-2	-1

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

Convolutional Layers + Pooling

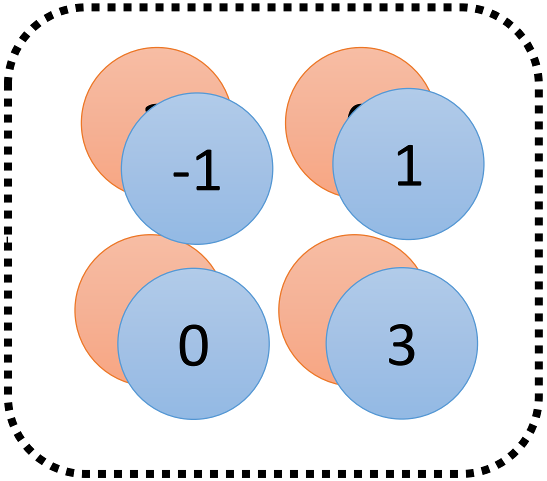
通常架構就是convolution和pooling
在交替使用





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

“Image” with 64 channels

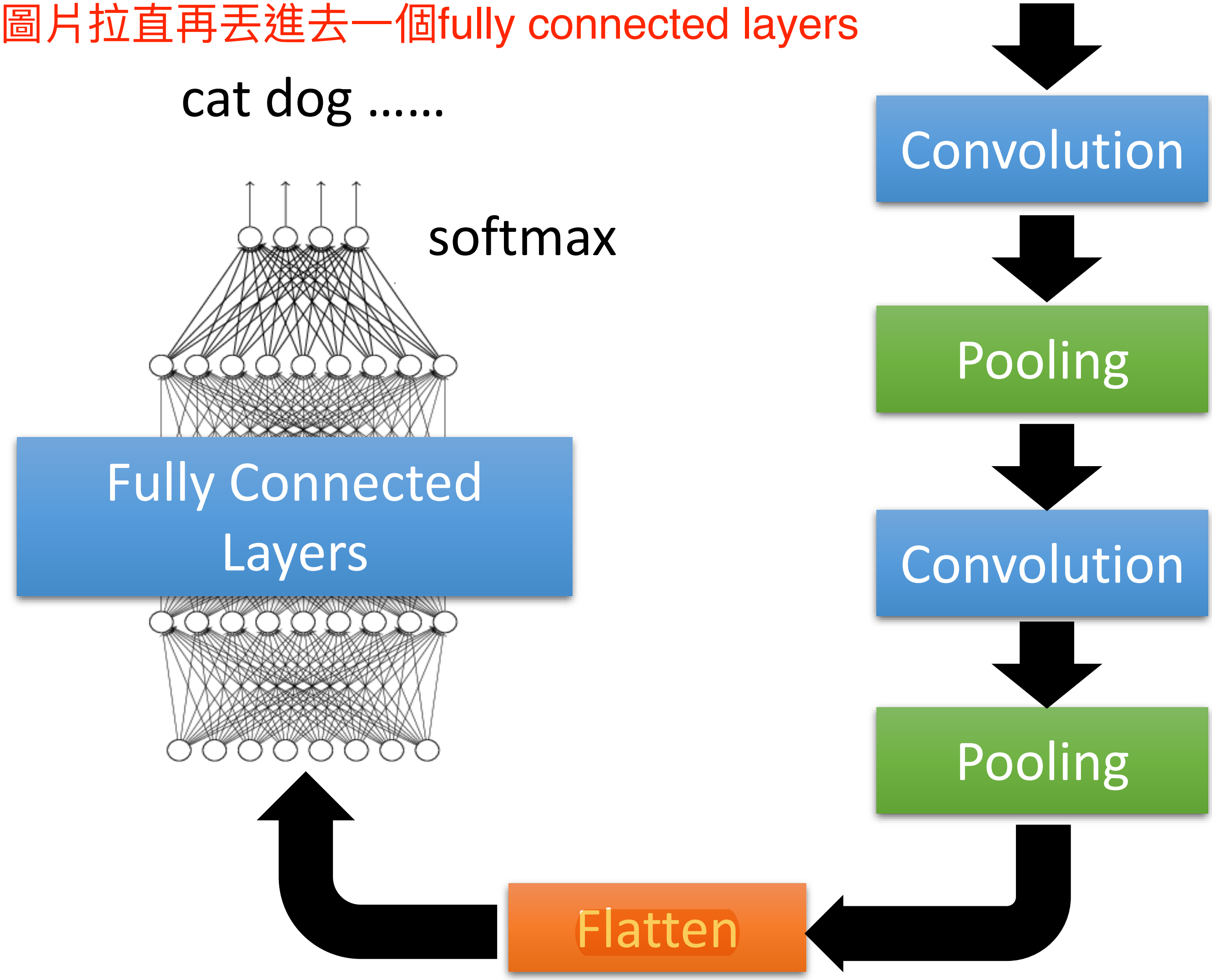


-1	1
0	3

pooling存在的理由是為了降低運算
但可能對資料有所傷害
因次近年來，運算能力變強
越來越多人把pooling拿掉

The whole CNN

最後把圖片拉直再丟進去一個fully connected layers



cat dog ……

softmax

Fully Connected Layers

Convolution

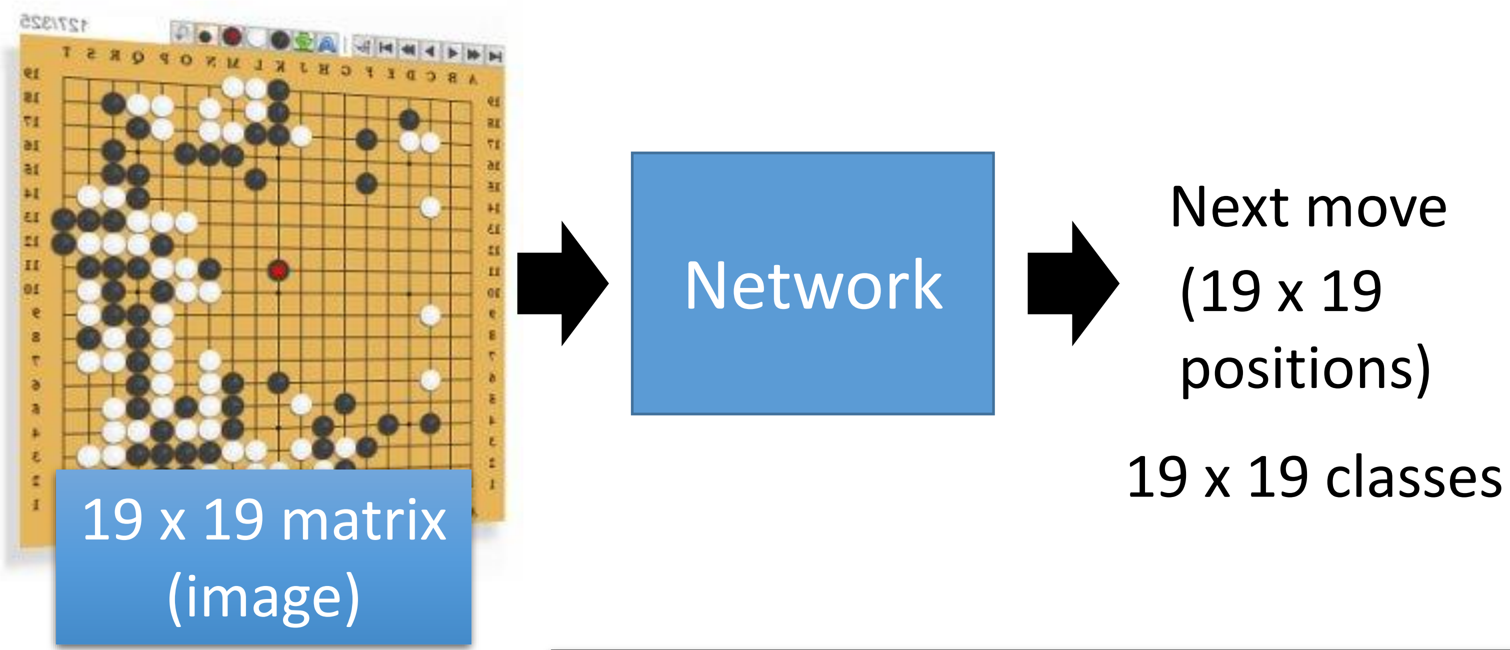
Pooling

Convolution

Pooling

Flatten

Application: Playing Go



19 x 19 matrix (image)

Network

Next move (19 x 19 positions)

19 x 19 classes

48 channels in Alpha Go

Black: 1

white: -1

none: 0

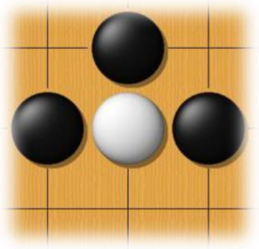
Fully-connected network can be used

But CNN performs much better.

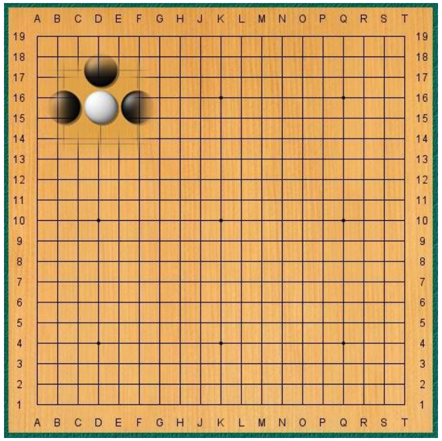
Why CNN for Go playing?

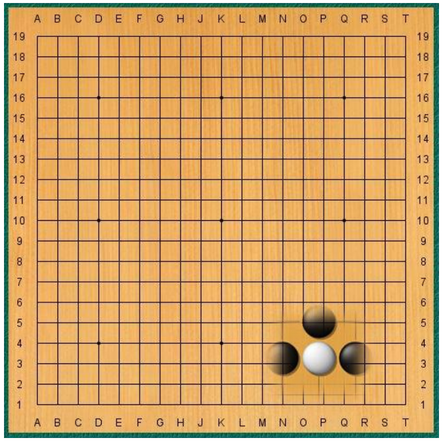
- Some patterns are much smaller than the whole image

Alpha Go uses 5 x 5 for first layer



- The same patterns appear in different regions.



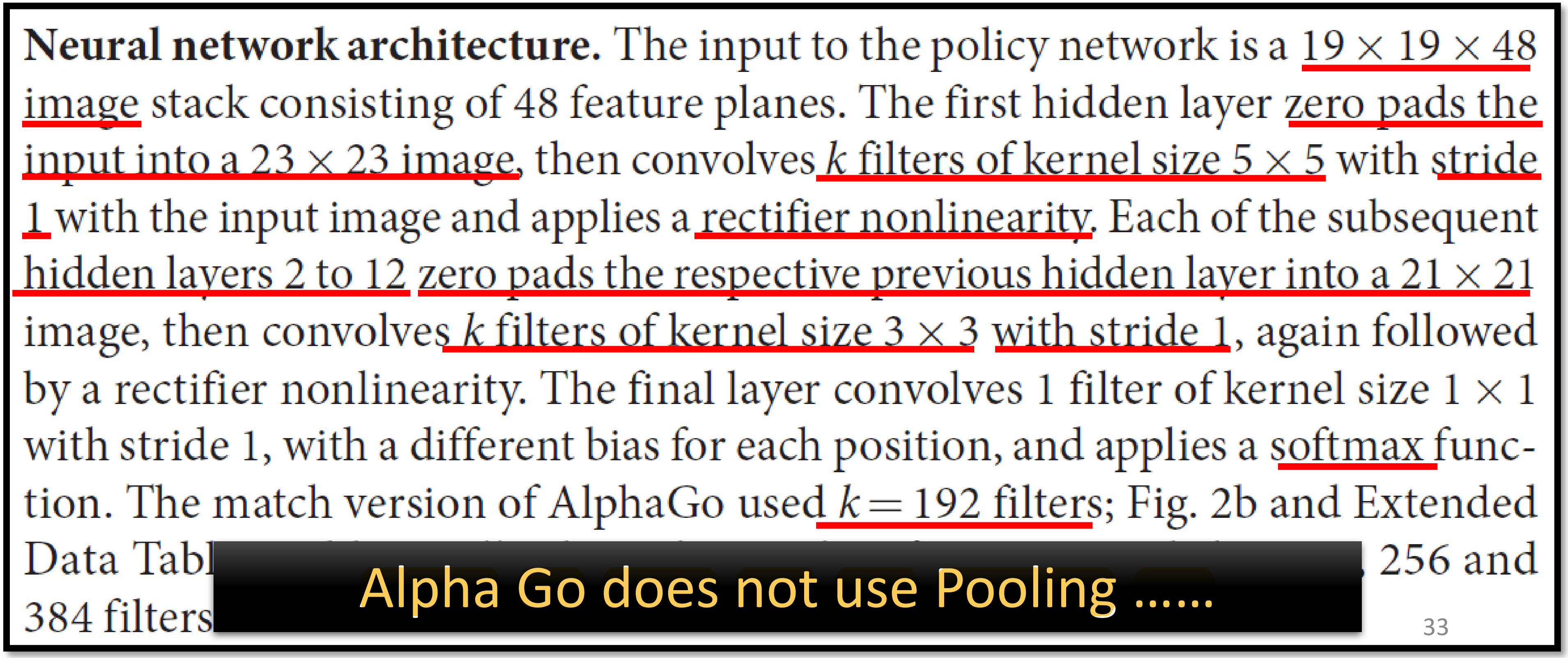


Why CNN for Go playing?

- Subsampling the pixels will not change the object

Pooling

How to explain this???



Neural network architecture. The input to the policy network is a 19 × 19 × 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 match version of AlphaGo used $k = 192$ filters; Fig. 2b and Extended Data Tabl

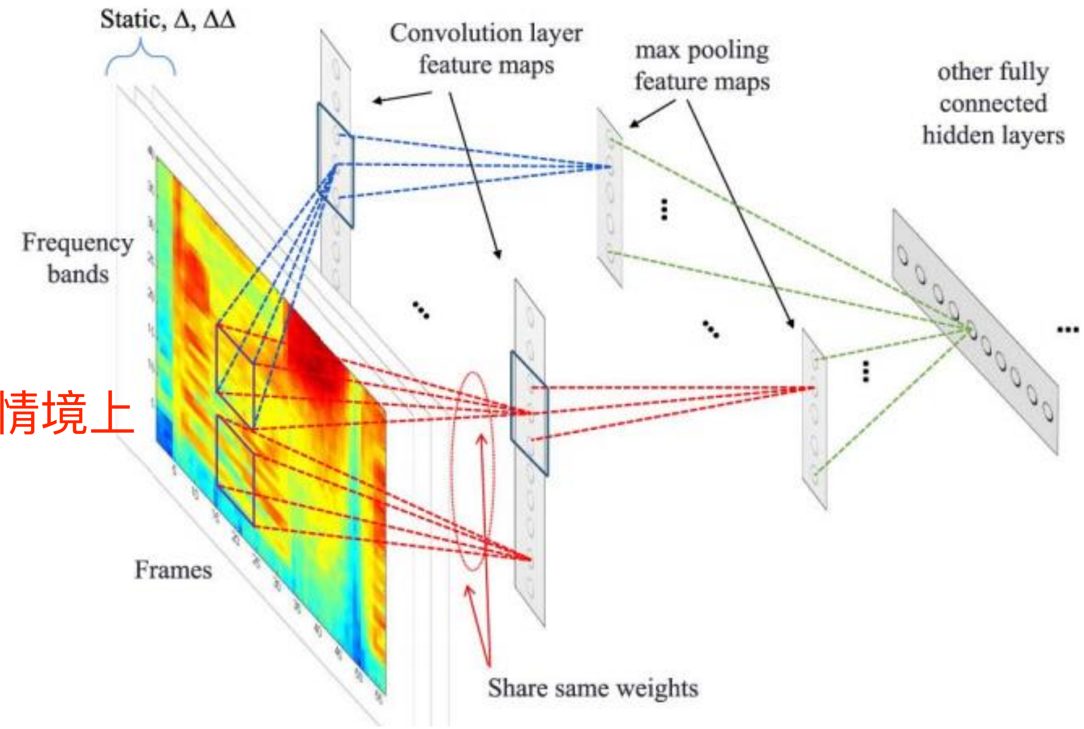
256 and

384 filters

Alpha Go does not use Pooling ……

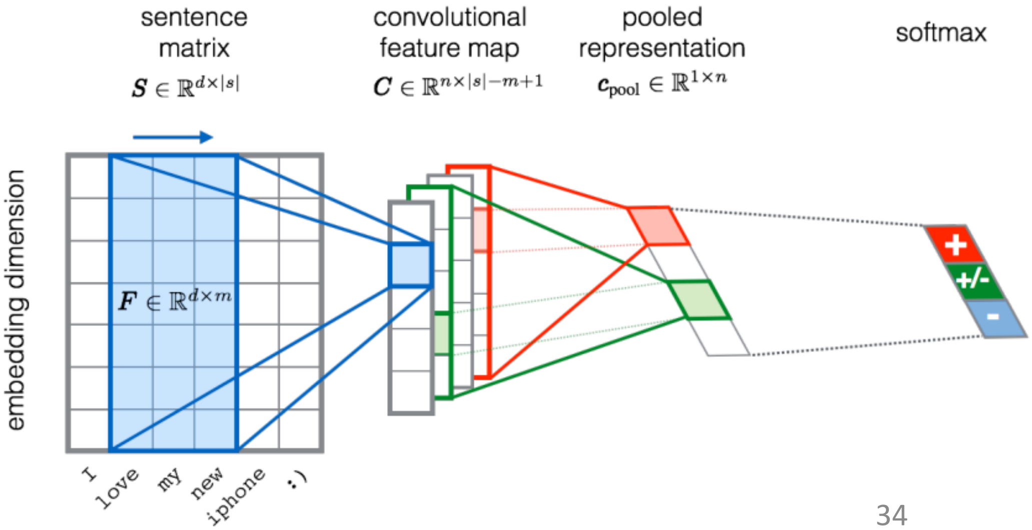
More Applications

每一種cnn的應用
都會針對問題去設計架構
並不是圖片的cnn就能套用在任意的情境上



Speech

https://dl.acm.org/doi/10.1109/TASLP.2014.2339736


sentence matrix
$S \in \mathbb{R}^{d \times |s|}$
convolutional feature map
$C \in \mathbb{R}^{n \times |s|-m+1}$
pooled representation
$c_{\text{pool}} \in \mathbb{R}^{1 \times n}$
softmax
embedding dimension
$F \in \mathbb{R}^{d \times m}$
I
love
my
new
iphone
:)
+
+/-
-

Natural Language Processing

https://www.aclweb.org/anthology/S15-2079/

cnn辨識不出來這兩張圖片
雖然data augmentation可以處理這種問題

To learn more ...

但是data augementation也不是所有放大或旋轉的角度都有包含
因此若能讓cnn自己學會scaling和rotation是最好的（就是spatial transformer layer）

- CNN is not invariant to scaling and rotation (we need data augmentation ☺).

Spatial Transformer Layer

https://youtu.be/SoCywZ1hZak
(in Mandarin)