卷積神經網路 (Convolutional Neural Networks, CNN)

Create at 2022/06/22

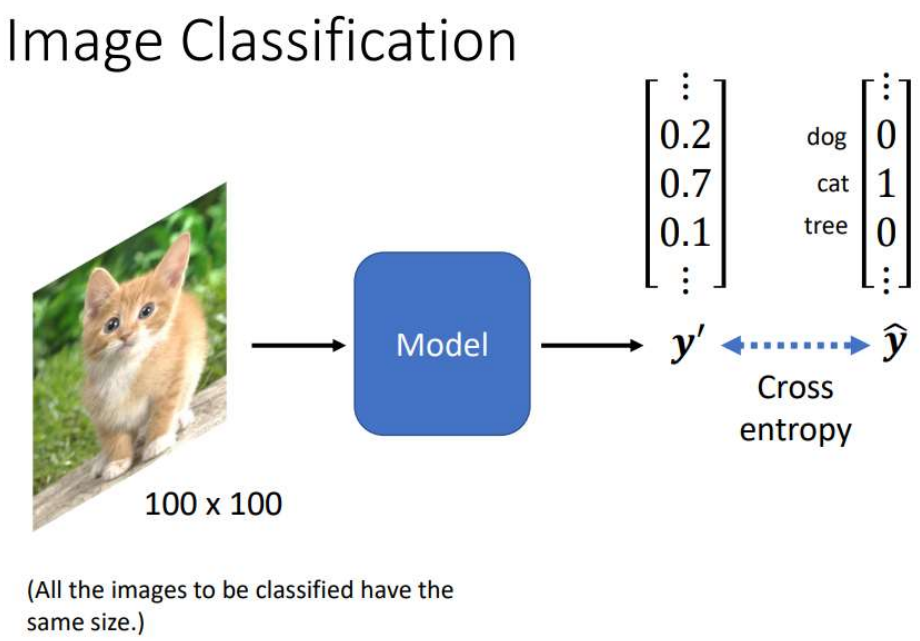
- 上課資源：
  1. 卷積神經網路 (Convolutional Neural Networks, CNN) (https://www.youtube.com/watch?v=OP5HcXJg2Aw)
- 參考資料：
  1. Spatial Transformer Layer (https://www.youtube.com/watch?v=SoCywZ1hZak)

CNN

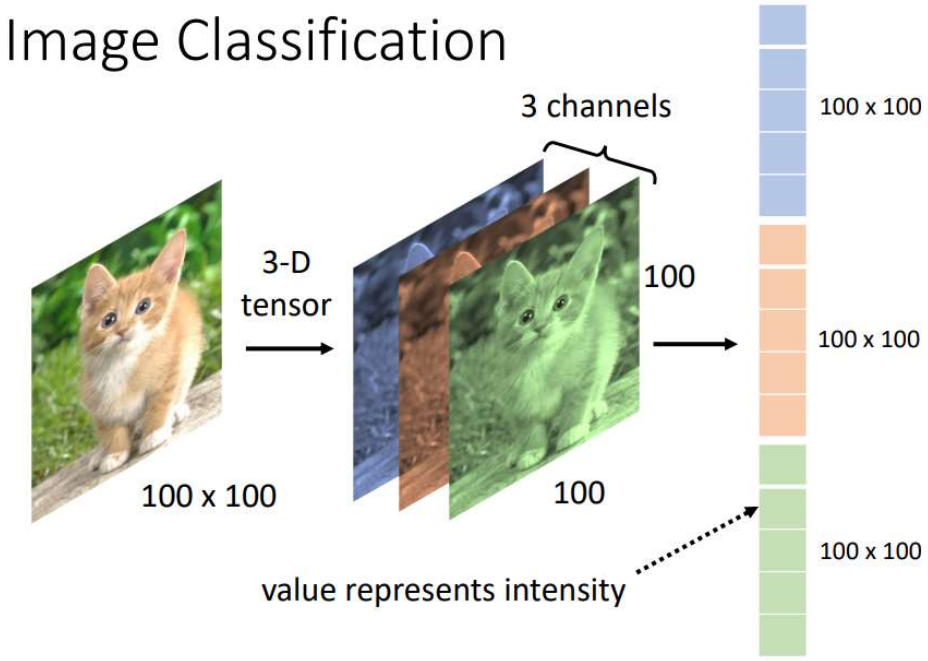
- 專門被用在影像上
- 希望透過 CNN 的例子，了解它的架構與想法
- 怎麼設計 Network 架構，可以讓 Network 結果做得更好

CNN 的第一種介紹方式

Image Classification

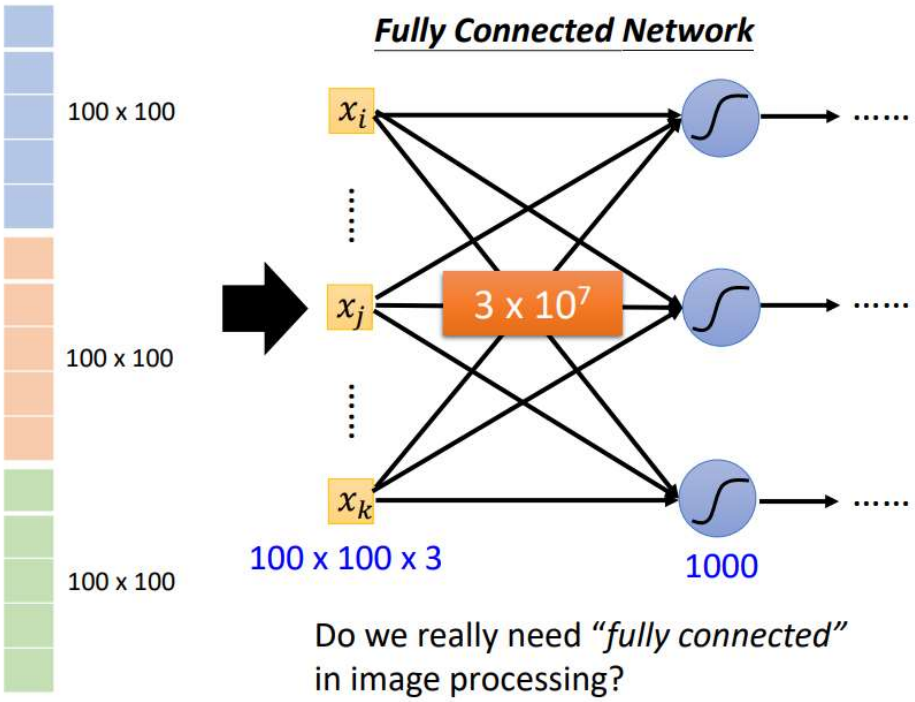


- 給機器一張圖片，會去判斷這張圖片裡面是什麼東西
- 以下討論，都假設我們模型輸入的圖片大小是固定的
- 把所有圖片都先 rescale 成一樣的大小，再丟到影像辨識系統
- 目標是分類，會把每個類別表示成一個 One-Hot vector $\hat{y}$
- 目標的長度表示現在的模型可以辨識出多少不同種類的東西
- 模型的輸出經過 softmax 之後就是 y'
- 希望 y' 跟 $\hat{y}$ 之間的 Cross Entropy 越小越好



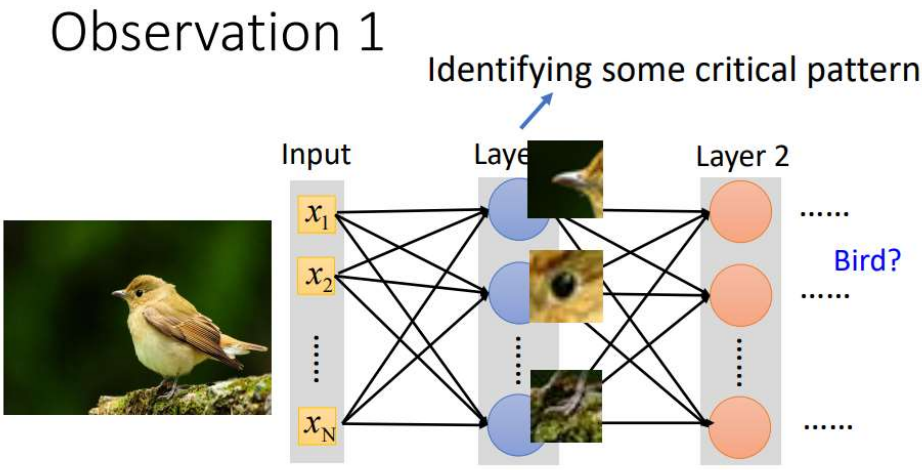
- 對電腦來說，一張圖片是一個三維的 tensor
  - tensor : 想像成是維度大於 2 的矩陣
  - 三維 :
    1. 圖片的寬
    2. 圖片的高
    3. 圖片的 channel 數目
       - R、G、B
- 把三維的 tensor 拉直，之後即可丟到 network 裡面

- 這張圖片是由 100 * 100 * 3 個數字 (pixel、intensity) 所組成
- 把數字拿出來排成一排，就是一個巨大的向量，可以做為 network 的輸入



- input 的 feature vector 長度就是 100 * 100 * 3
- 假設第一層 neuron 的數目有 1000 個，第一層總共有 $3 * 10^7$ 個 weight
- 雖然參數的增加可以增加模型的彈性跟能力，但是也增加了 overfitting 的風險

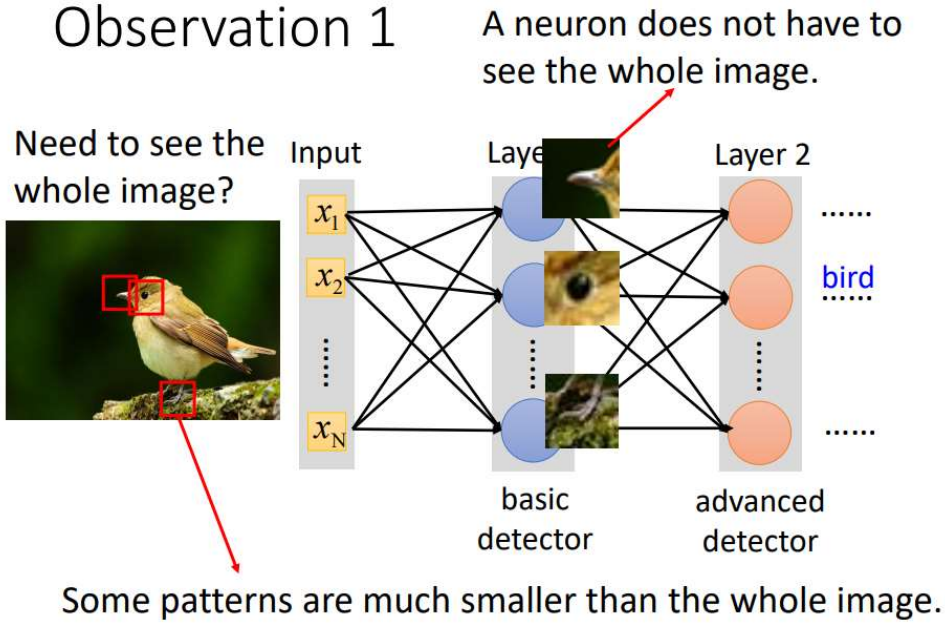
Observation 1



- 影像辨識這個問題
- 對影像辨識系統、影像辨識的類神經網路裡面的神經元而言
  - 要做的就是偵測圖片裡面有沒有特別重要的 pattern
  - 而這些 pattern 代表了某種物件

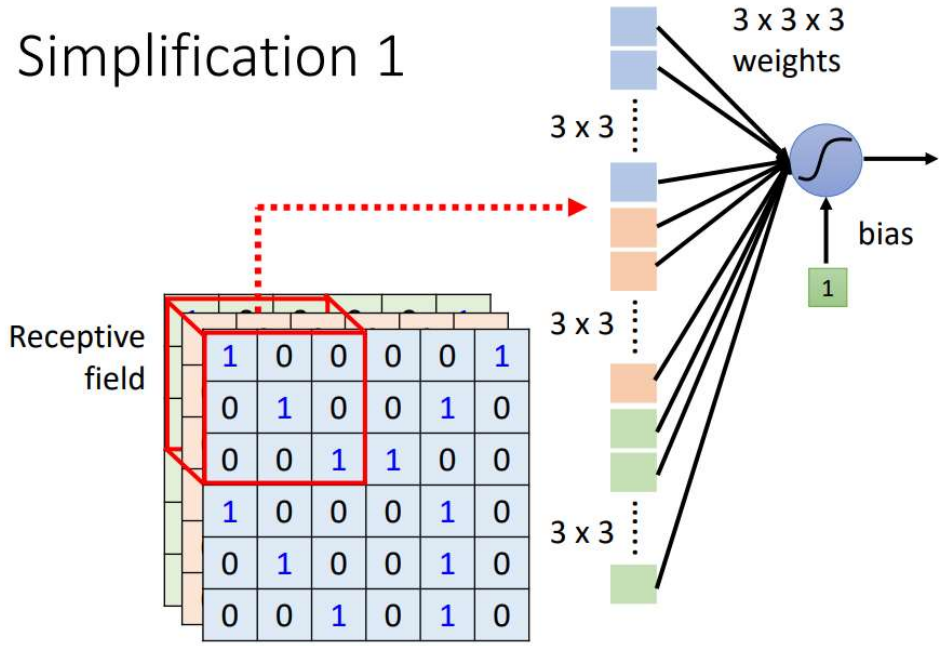
https://www.dcard.tw/f/funny/p/233833012

- 就算是人，在判斷物件的時候往往也是抓最重要的特徵

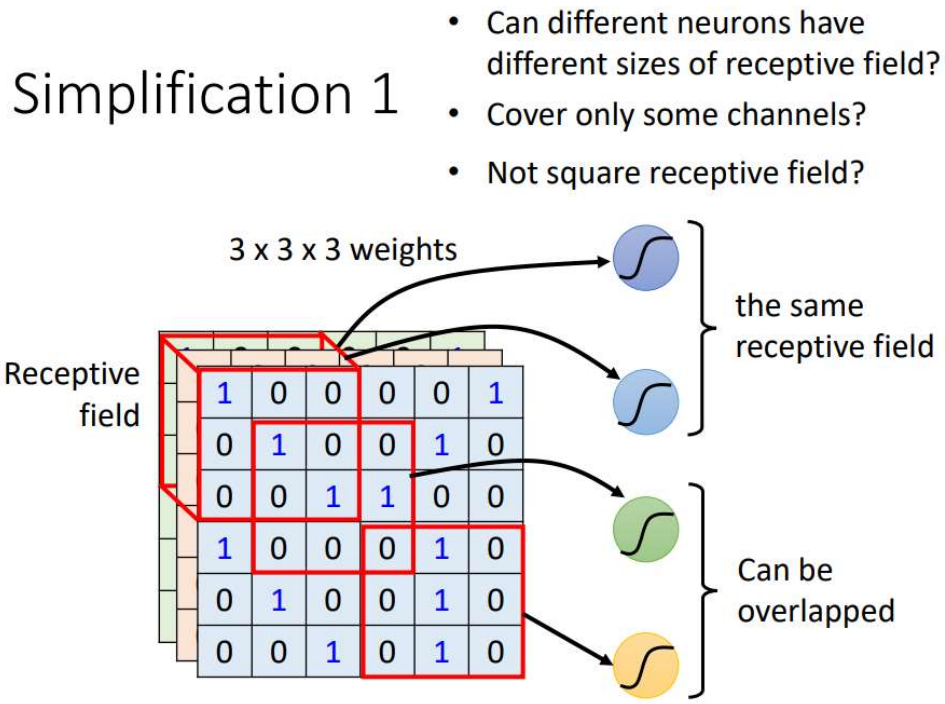


- 假設現在用 neuron 做的是判斷有沒有某種 pattern 出現
- 不一定需要每個 neuron 都去看一張完整的圖片
- 因為例如鳥嘴、眼睛、鳥爪，並不需要看整張完整的圖片才能夠得到這些資訊
- 要知道有沒有鳥嘴，只要看很小的範圍就知道了
- 根據這個觀察即可做一個簡化

Simplification 1



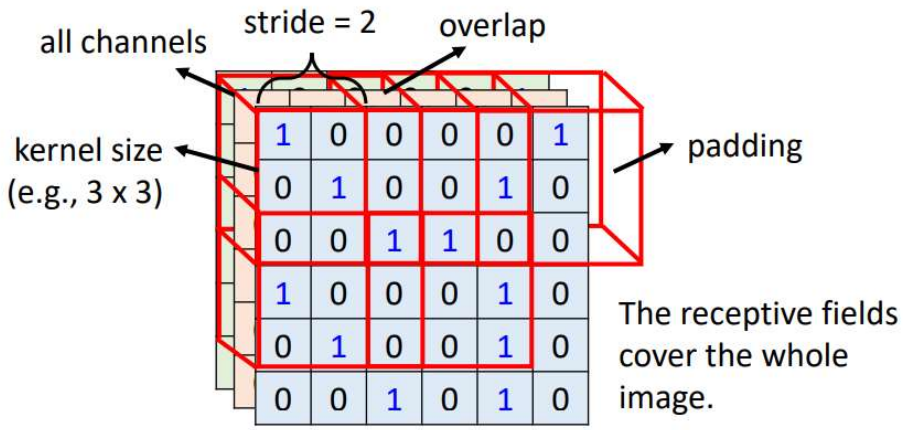
- 本來每個 neuron 要看完整的圖片，圖片的每個 pixel 有 3 個數字，把一張圖片裡面所有的資訊都丟給一個 neuron，這是 fully connected network
- 簡化：
  - 設定一個 Receptive field，每個 neuron 都只關心自己的 Receptive field 就好
  - 把 $3 * 3 * 3$ 的數值拉直，變成一個 27 維的向量
  - 再把這個向量作為 neuron 的輸入
  - neuron 會給每一個 dimension 一個 weight
  - dimension value * weight + bias 得到輸出
  - 這個輸出再送給下一層的 neuron 當作輸入



- Receptive field 彼此之間可以是重複的，也可以不相連
- Receptive field 可以有大有小
- 可以只考慮某一個 channel
- Receptive field 可以是長方形的

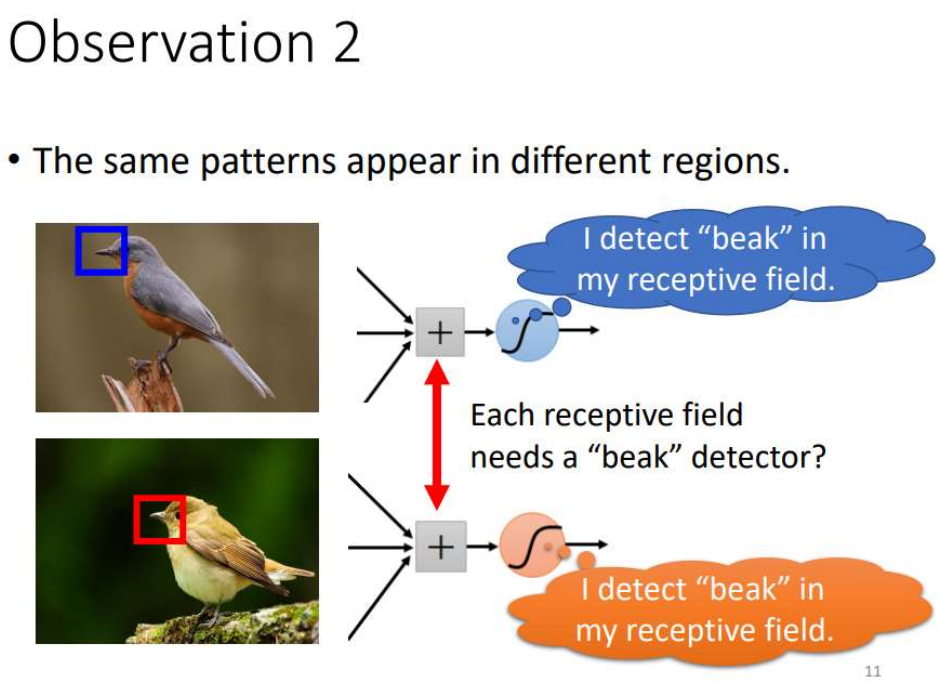
Simplification 1 – Typical Setting

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



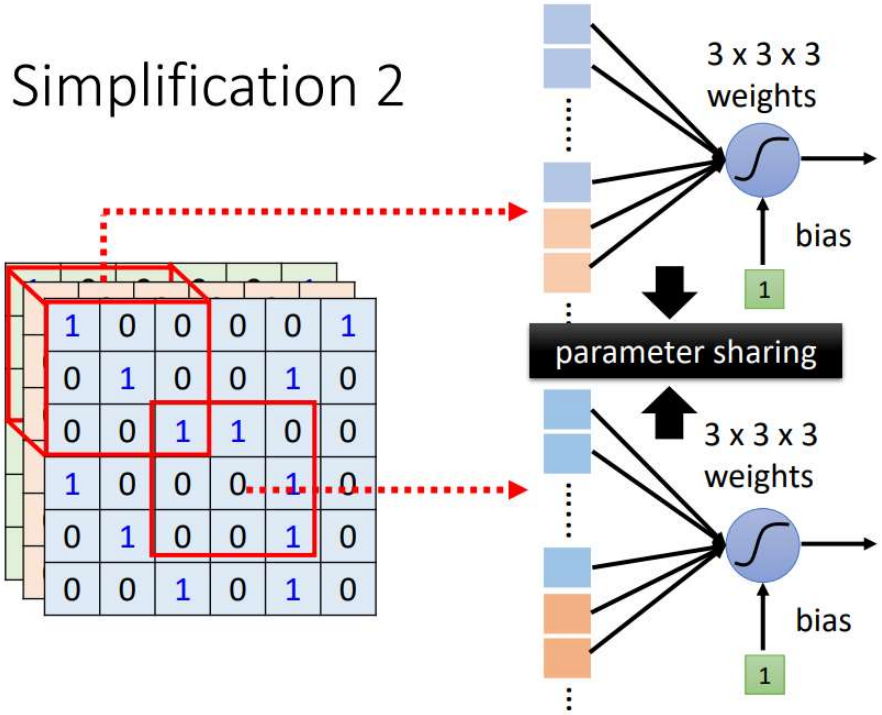
- 經典的 Receptive Field 設計
  - 會看所有的 channel，所以在描述一個 receptive field 的時候只要講高 & 寬即可，kernel size (高 * 寬)
  - kernel size 往往都不會設太大，3 * 3 就夠了，7 * 7、9 * 9 算是大的 kernel size
  - 一般來說，同一個 receptive field 會有一組 neuron 去看這個範圍，不會只有一個 neuron 去看它
- 各個不同 Receptive Field 之間的關係
  - 會把左上的 receptive field 右移一點形成一個新的 receptive field，移動的量稱為 stride
  - stride 是一個 hyperparameter 的參數，往往不會設太大，設 1 or 2 即可
  - 因為會希望 receptive field 之間是有重疊的
  - 超出影像範圍怎麼辦？
    - 做 padding (補 0、整張圖片的平均、延續邊邊)

Observation 2

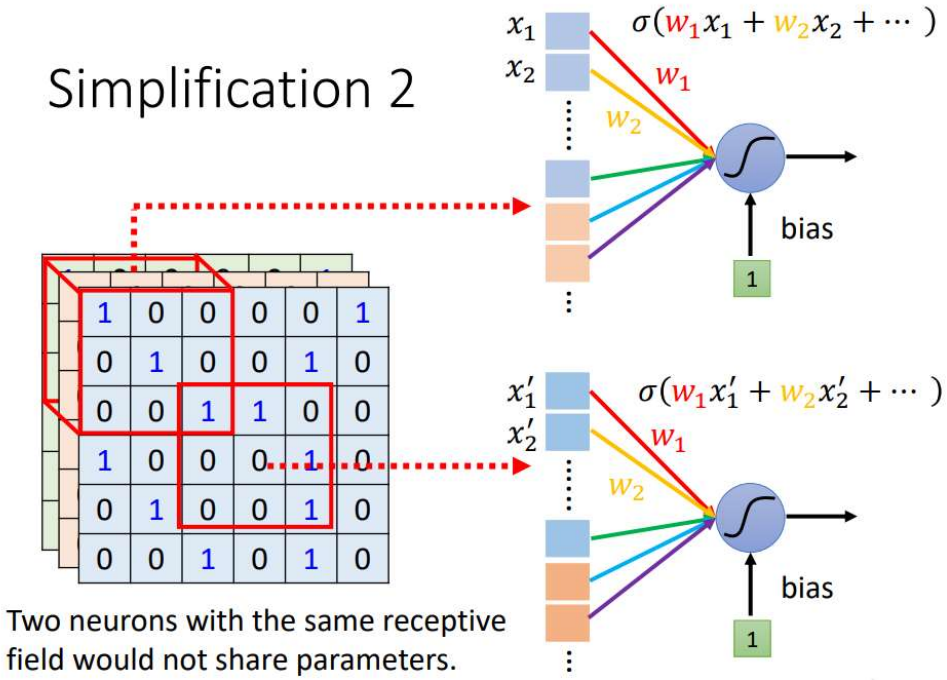


- 同樣的 pattern 可能會出現在圖片的不同區域裡面
- 任何區域都一定在某個 receptive field 裡面，receptive field 一定有一組 neuron 在查看
- 偵測鳥嘴的 neuron 其實他們做的事情是一樣的，只是他們查看的範圍不一樣

Simplification 2



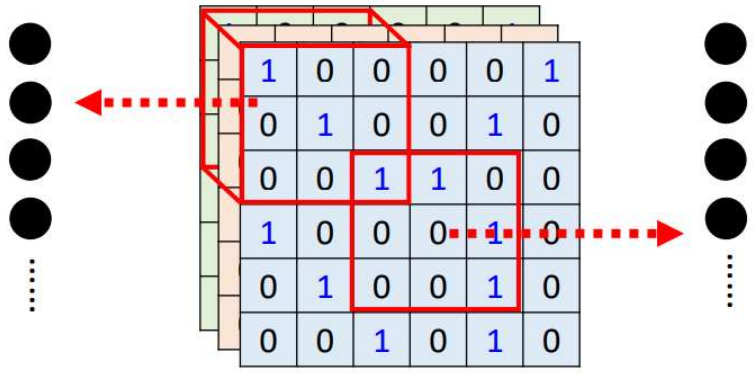
- 能不能進行簡化？
  - 因為他們做的事情都是一樣的，真的需要那麼多的參數嗎？

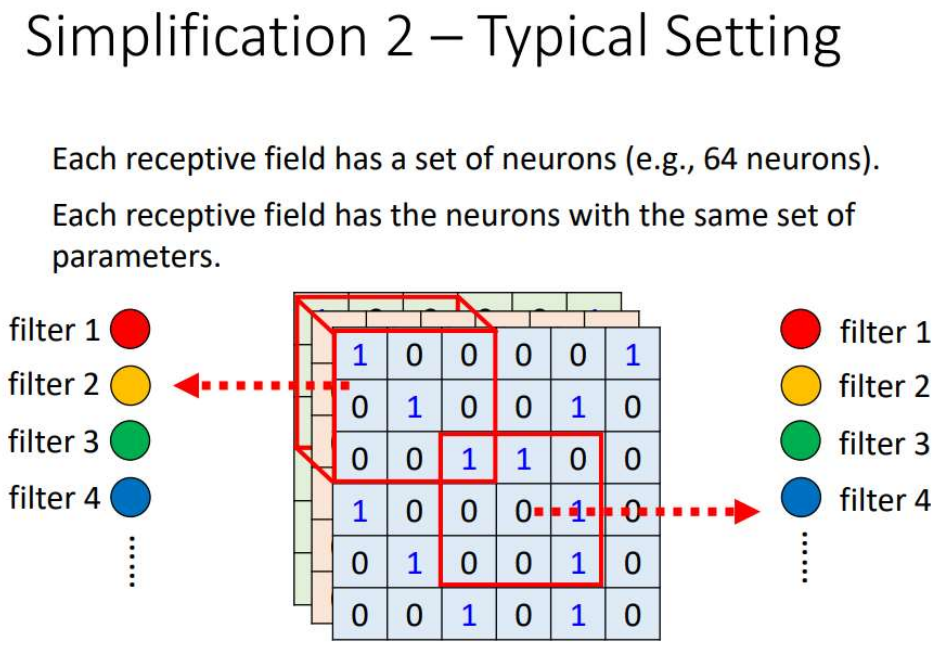


- 是否能讓 receptive field 的 neuron 共享參數 (parameter sharing)
  - 意思是：兩個 neuron 的 weight 相同
  - 因為他們的輸入是不一樣的，所以就算兩個 neuron 共用參數，輸出也不會一樣

Simplification 2 – Typical Setting

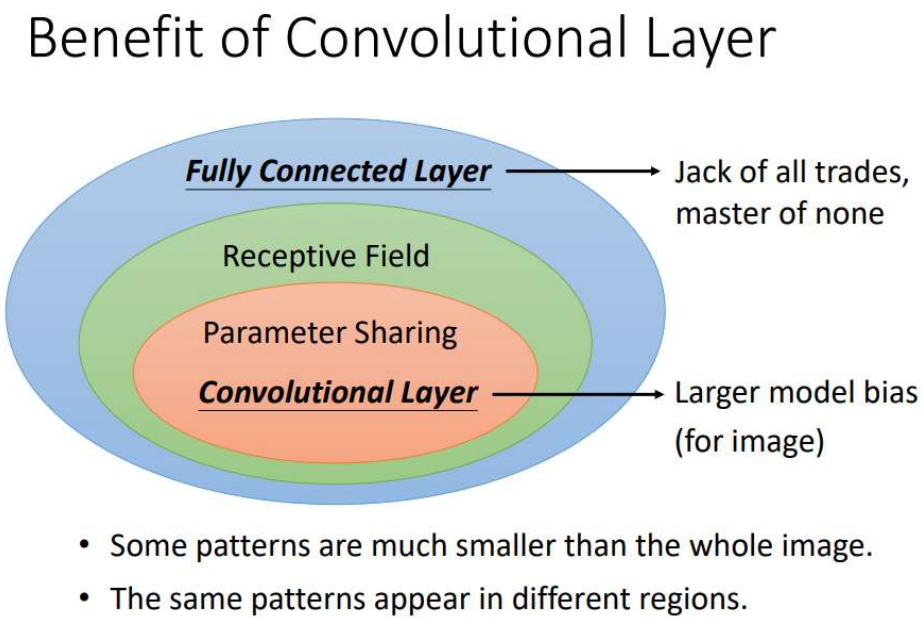
Each receptive field has a set of neurons (e.g., 64 neurons).





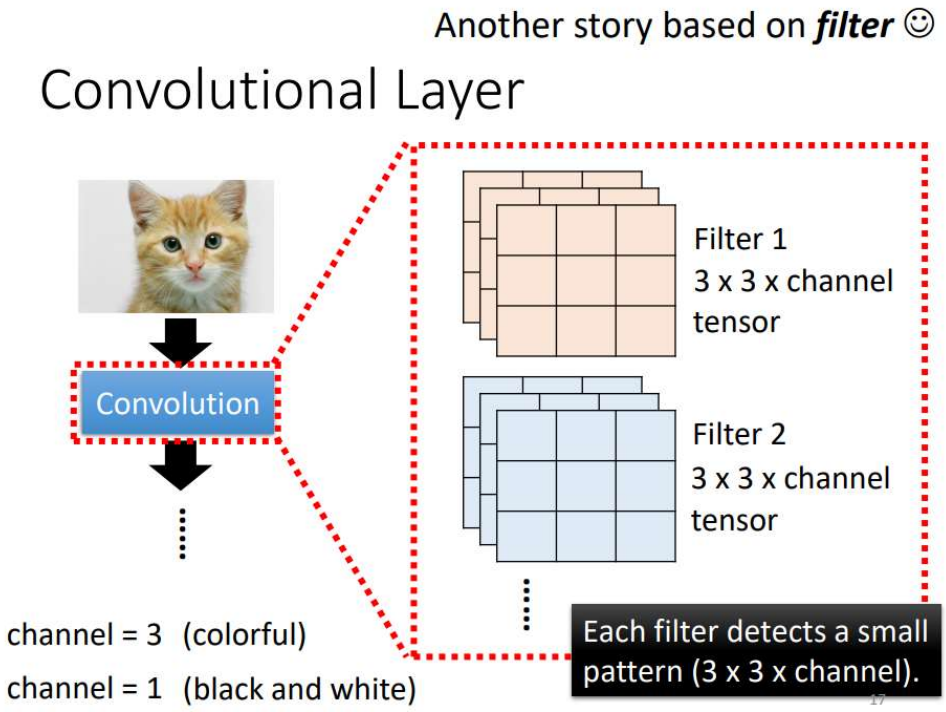
- 在影像辨識上常見參數共享的方法
  - 每一個 receptive field 都有一組 neuron 進行查看 (例如 64 個 neurons)
  - 每個 receptive field 都有 64 個 neuron
  - 所以每一組 receptive field 都只有一組參數
  - 這些參數稱為 **Filter**

Benefit of Convolutional Layer

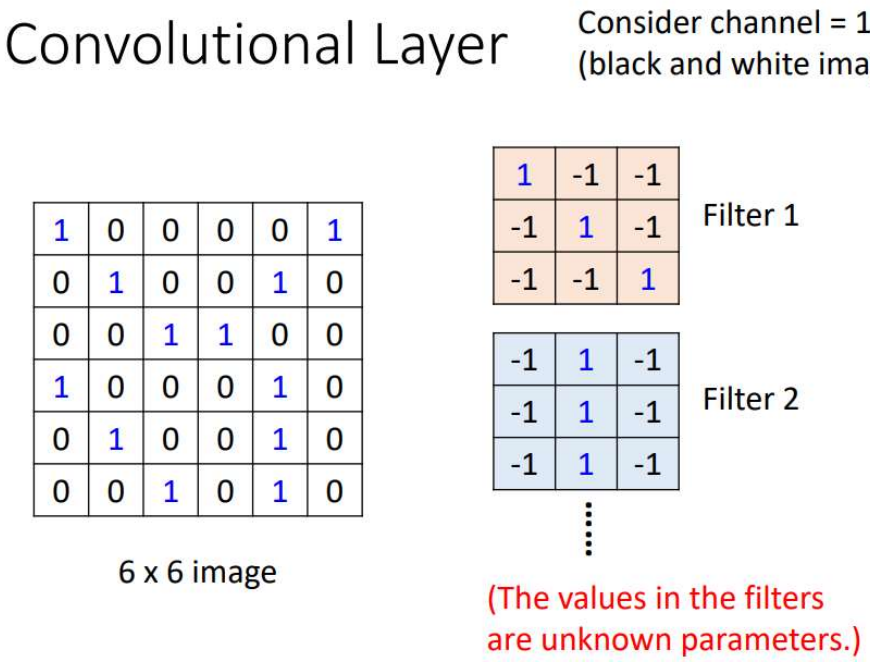


- Receptive Field 加上 Parameter Sharing 就是 Convolutional Layer
- 有用 Convolutional Layer 的 network 就叫 **Convoutional Neural Network (CNN)**
- CNN 的 model bias 比較大，但 bias 大不一定是壞事
  - 因為當 model bias 小，model 的 flexibility 很高的時候，比較容易 overfitting
- Fully Connected Layer 可以做各式各樣的事情和變化，但是它沒辦法在任何特定的任務上做好
- Convolutional Layer 是專門為影像設計的，所以在影像上仍然可以做得好，雖然 bias 大，但這個在影像上不是問題

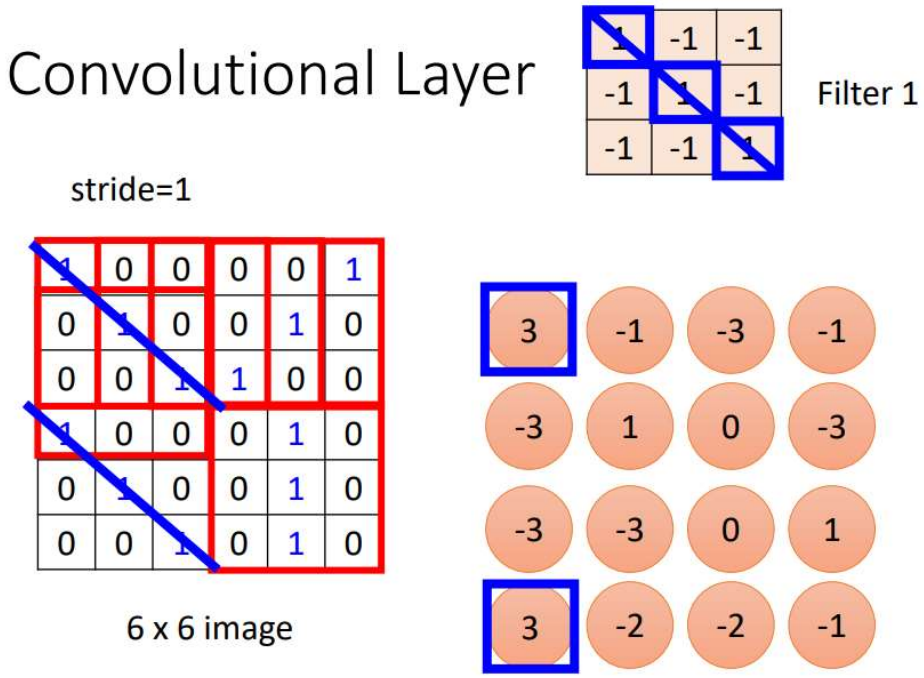
CNN 的第二種介紹方式



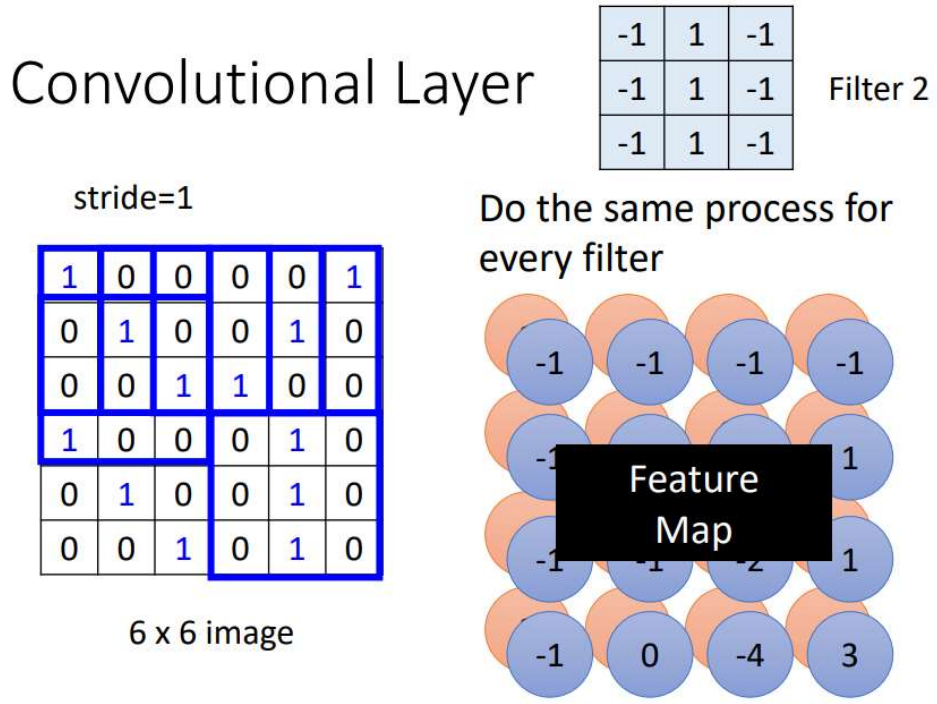
- Convolutional Layer 裡面有一排的 filter
- 這些 filter 的大小是 3 * 3 的 tensor，再 * channel
  - channel
    - 彩色圖片：3
    - 黑白圖片：1
- 每一個 filter 的作用，是去圖片抓取某一個 pattern



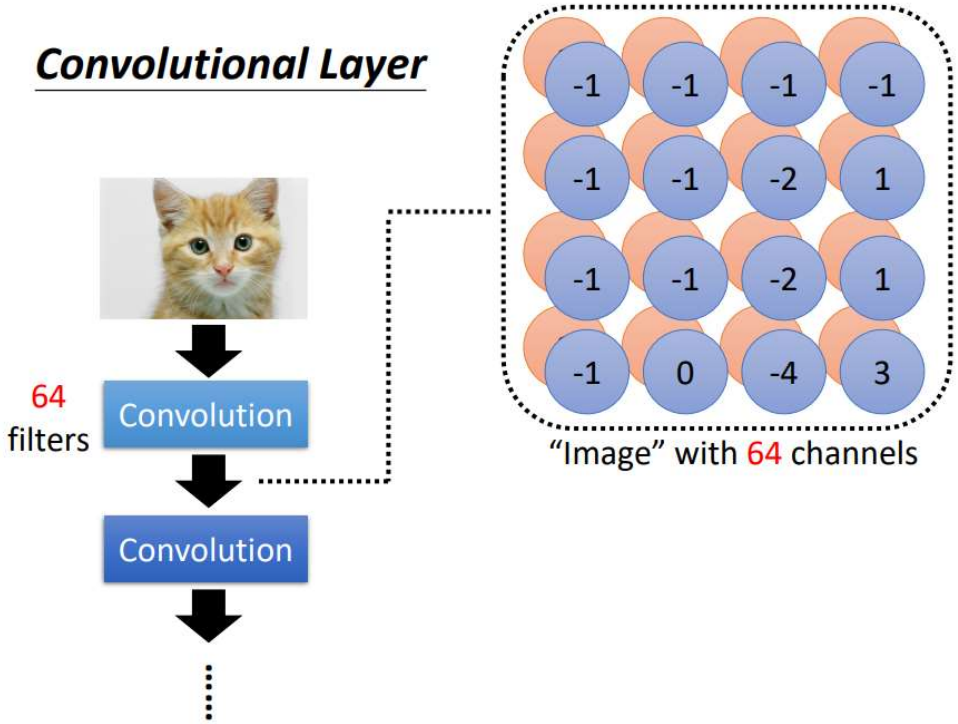
- filter 怎麼去圖片抓 pattern 呢？
  - 這邊 channel 假設是 1，代表是黑白的圖片
  - 假設 filter 的參數是已知的
  - 實際上 tensor 裡面的數值是 model 裡面的 parameter 其實是未知的，需要透過 gradient decent 去找出來



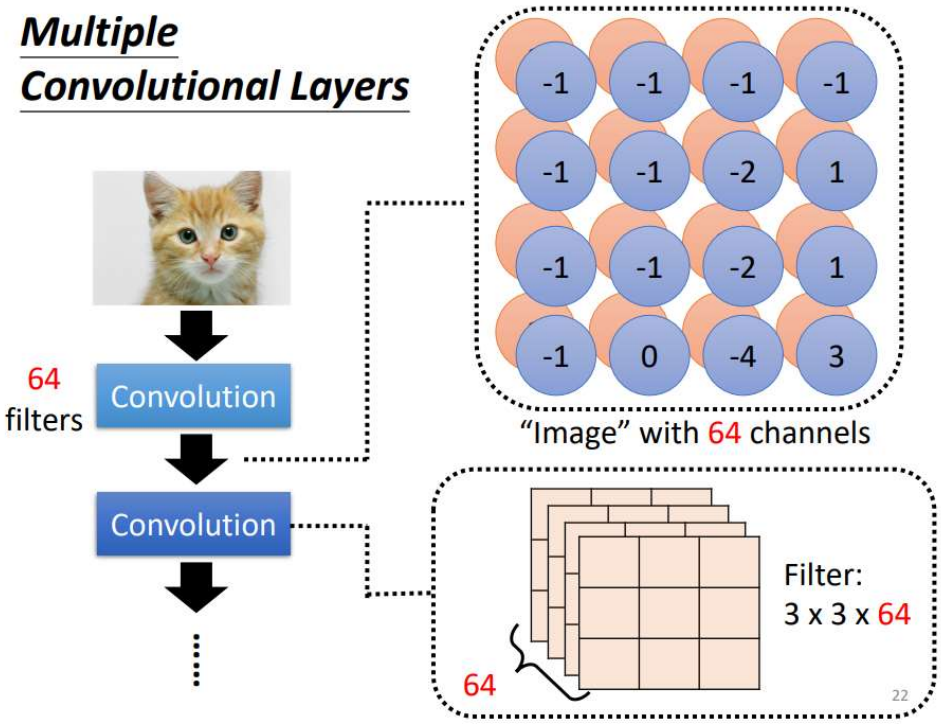
- stride 設為 1，九宮格個別相乘後相加
- 可以觀察特徵



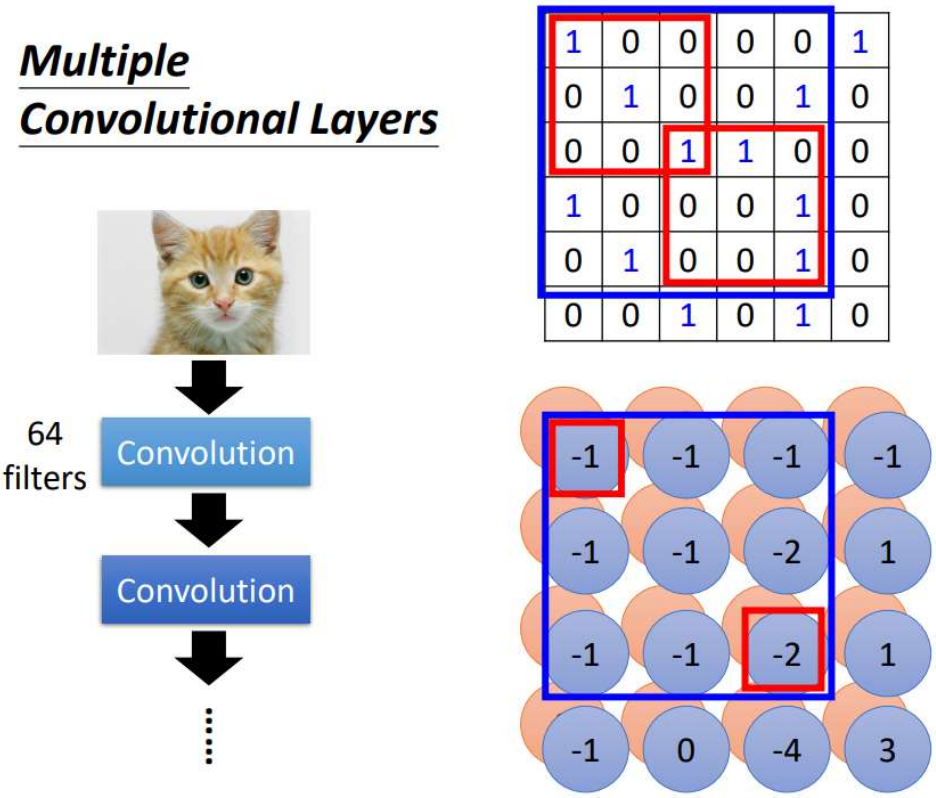
- 把每一個 filter 都做重複的 process
- 每個 filter 都會得到一群數值
- 這一大群數值稱為 Feature Map



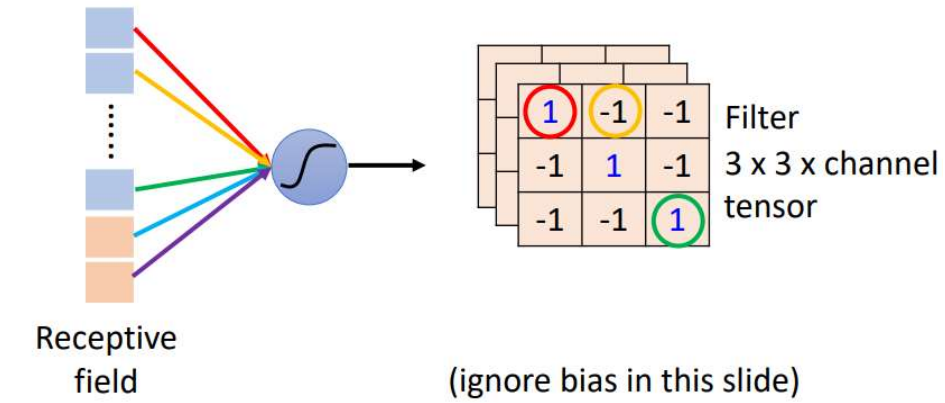
- 當把一張圖片通過一個 Convolutional Layer，裡面有一堆 filter 的時候，產生出來了一個 feature map
- 假設這個 convolution layer 裡面有 64 個 filter
  - 產生出來的 feature map 就有 64 組數字
  - 每一組在這個例子是 4 * 4
- Feature map 可以看成是另外一張新的圖片，只是這個圖片的 channel 不是 RGB 圖片的 channel，它有 64 個 channel



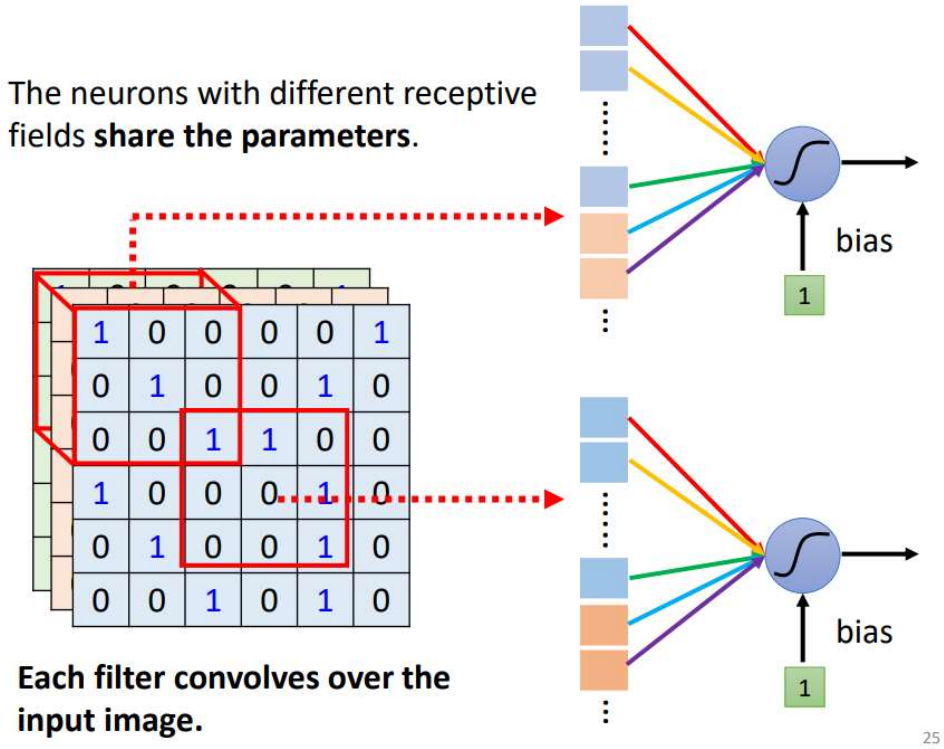
- Convolutional Layer 可以疊很多層



- 如果 filter 的大小一直設 3 * 3，會不會讓 network 沒辦法看比較大範圍的 pattern
- 所以 network 疊的越深，同樣是 3 * 3 大小的 filter，它看的範圍就會越大，所以不用怕偵測不到比較大的 pattern



- 兩個版本的故事相同
  - 第一個版本說到 neural 會共用參數，這些共用的參數就是第二個版本故事裡面的 filter



- 第一個版本說不同的 neuron 可以 share weight，然後去查看不同的 receptive field
  - shared weight 其實就是把 filter 掃過一張圖片
  - filter 掃過一張圖片，其實就是 Convolution

Convolutional Layer

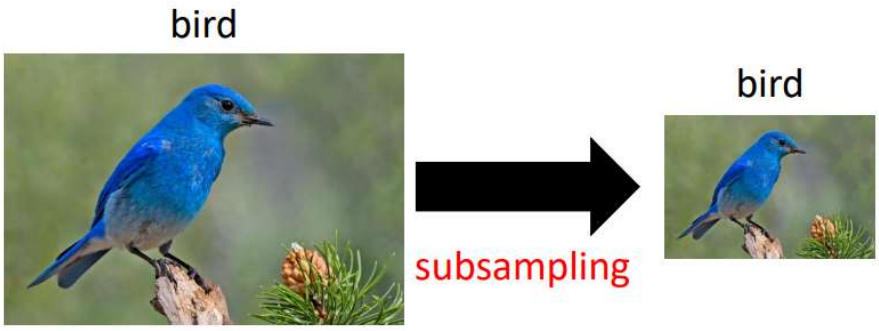
Neuron Version Story	***Filter Version Story***
Each neuron only considers a receptive field.	There are a set of filters detecting small patterns.
The neurons with different receptive fields share the parameters.	Each filter convolves over the input image.

They are the same story.

Observation 3

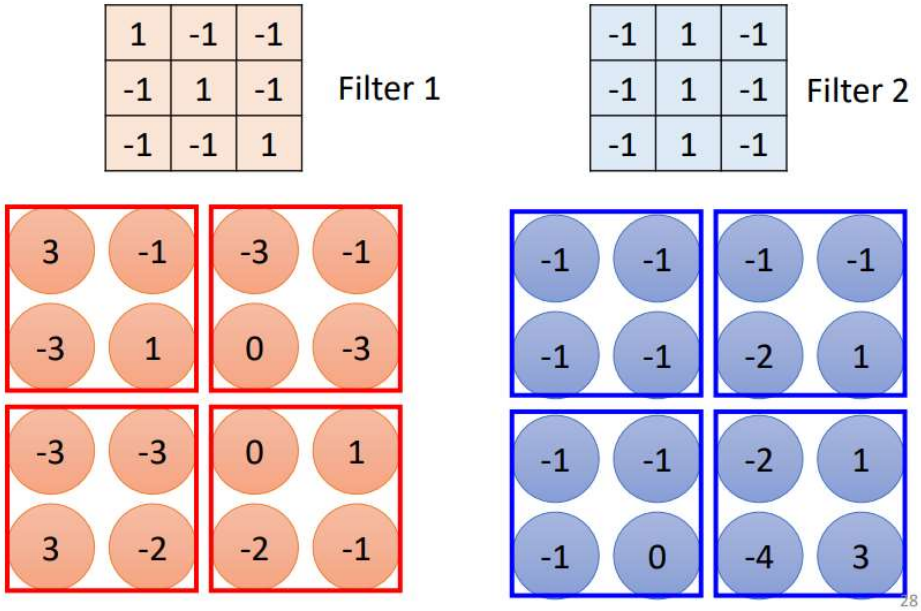
Observation 3

- Subsampling the pixels will not change the object

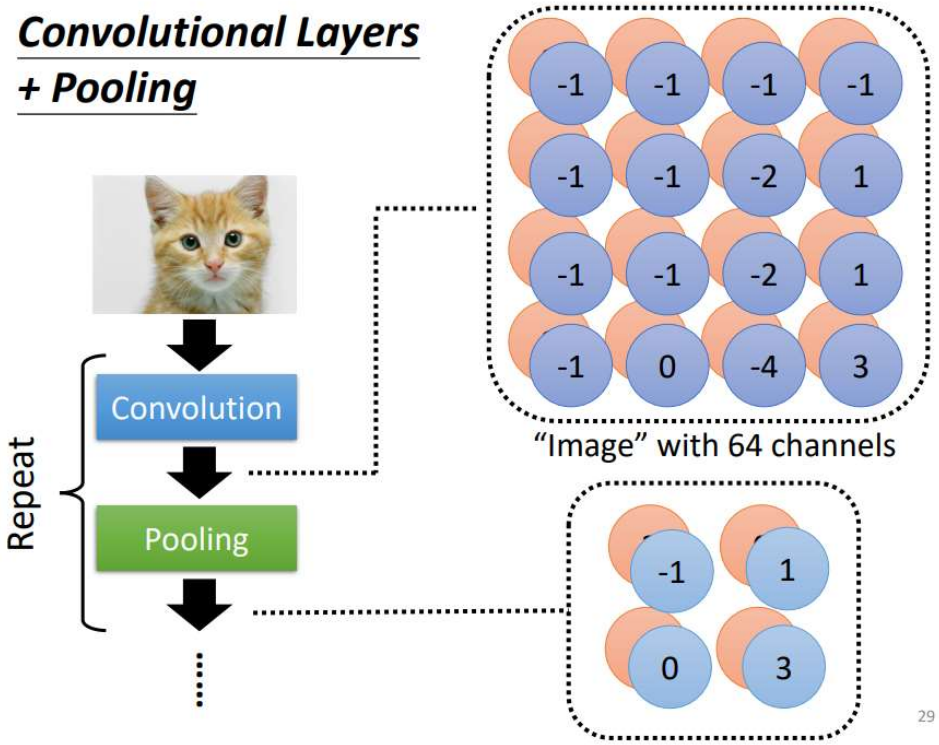


- 把一張比較大的圖片做 subsampling
  - ex. 把偶數的 column 都拿掉，奇數的 row 都拿掉，圖片變成原來的 1/4，但是不會影響圖片是什麼東西

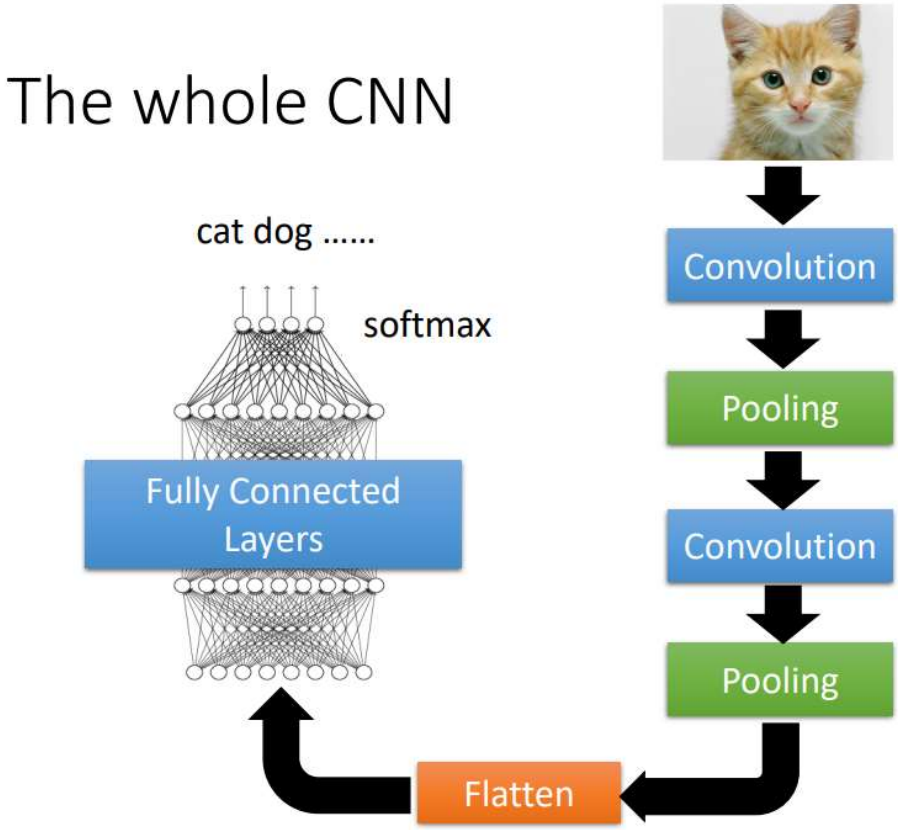
Pooling – Max Pooling



- Pooling
  - 本身沒有參數
  - 比較像一個 activation function (Sigmoid、ReLU 那些)
  - 每一個 filter 都產生一組數字
  - 要做 pooling 的時候，把這些數字幾個組成一組
  - 每一組裡面選一個代表
  - 在 Max Pooling 裡面，選的代表就是最大的那個

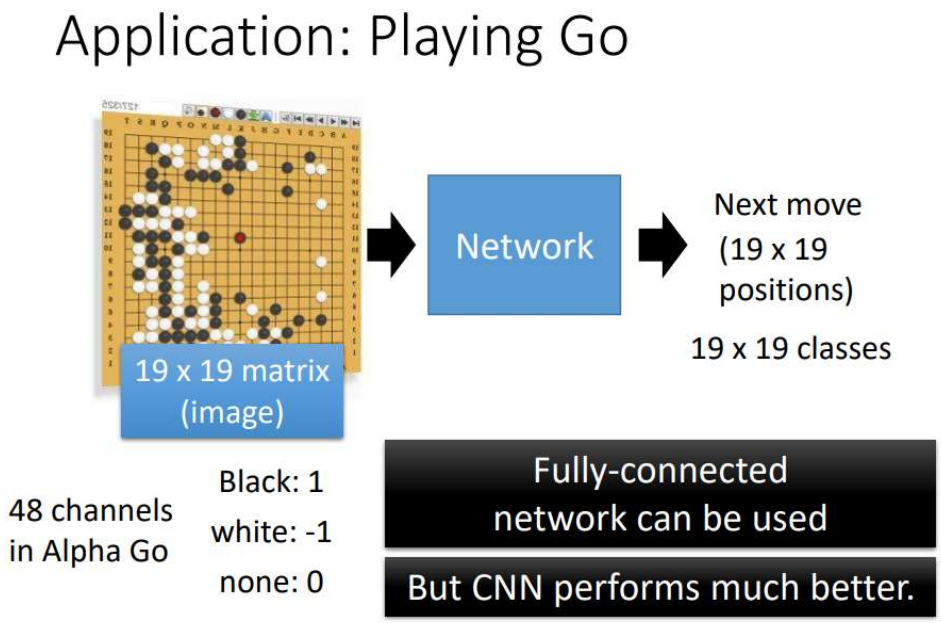


- 做完 Convolution 之後會得到一張圖片，這張圖片有很多的 channel，往往會搭配 Pooling
- Pooling 做的就是把圖片變小，但 channel 不變
- 在實作上往往是 convolution 跟 pooling 交替使用
- 但 pooling 對於 performance 有可能帶來傷害
  - 如果要偵測的是非常細微的東西
  - 隨便做 subsampling，performance 就會比較差



- pooling 可有可無
- 經典的影像辨識 network
  - 最後會把 pooling 的 output 做 flatten
  - 在把這個向量丟進 fully connected layer
  - 最後可能再過個 softmax
  - 得到影像辨識的結果

應用



- 下圍棋是一個分類的問題
  - 輸入：棋盤上黑子和白子的位置
  - 輸出：下一步應該要落子的位置
- network 的輸入是一個向量，如何把棋盤表示成一個向量？
  - 黑：1
  - 白：-1
  - X：0
- 下圍棋是一個 19 * 19 個類別的問題

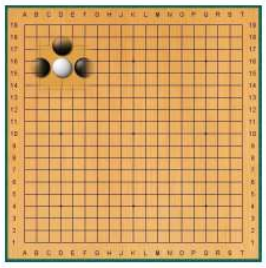
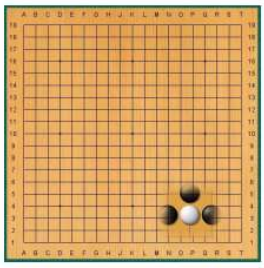
- 可以用 fully connected network 解決
- 但是用 CNN 的效果更好
  - 可以把棋盤看成是一張 19 * 19 的圖片
  - 棋盤上每個 pixel 有 48 個 channel

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.

- 圍棋跟影像有共同的特性
  - observe 1：很多重要的 pattern，只需要看小範圍就知道
  - observe 2：同樣的 pattern 可能會出現在不同的位置

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 $19 \times 19 \times 48$ 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 match version of AlphaGo used $k = 192$ filters; Fig. 2b and Extended Data Tab... 256 and 384 filters

Alpha Go does not use Pooling .....

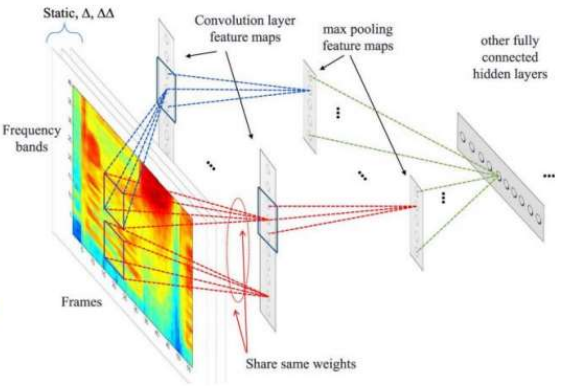
33

- Alpha Go 的 network 沒有用 pooling

More Applications

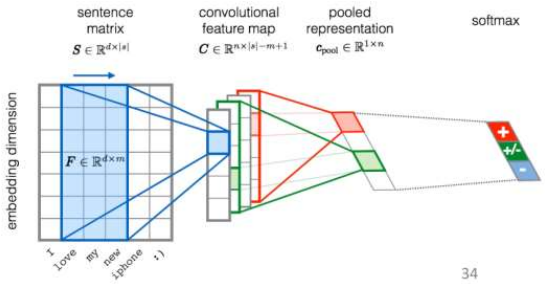
Speech

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



Natural Language Processing

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



34

CNN 也用在語音、文字處理

To learn more ...

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

Spatial Transformer Layer

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

35

- CNN 無法處理影像放大、縮小、旋轉的問題
- 所以在座影像辨識的時候往往要做 data augmentation
  - data augmentation : 把訓練資料，美張圖片都截一小塊出來放大，讓 CNN 看過不同大小的 pattern，然後把圖片旋轉，讓它有看過某一個物件旋轉以後的樣子，CNN 才會做到好的結果
- Spatial Transformer Layer 架構可以處理 scaling、rotation 的問題

tags: 2022 李宏毅_機器學習