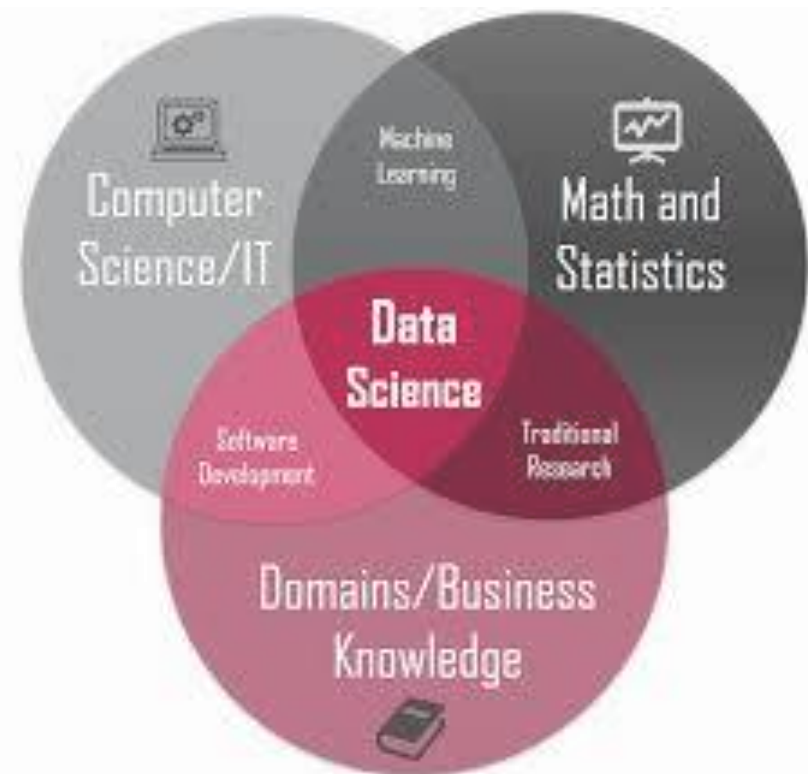


AI & DS



U11-深度學習_卷積神經網路CNN

2023.11_V1.1

Data
Science

Artificial
Intelligence

Machine
Learning

Deep
Learning

Statistics

單元大綱

- 認識卷積神經網路(CNN)
- 建立與訓練CNN相關語法

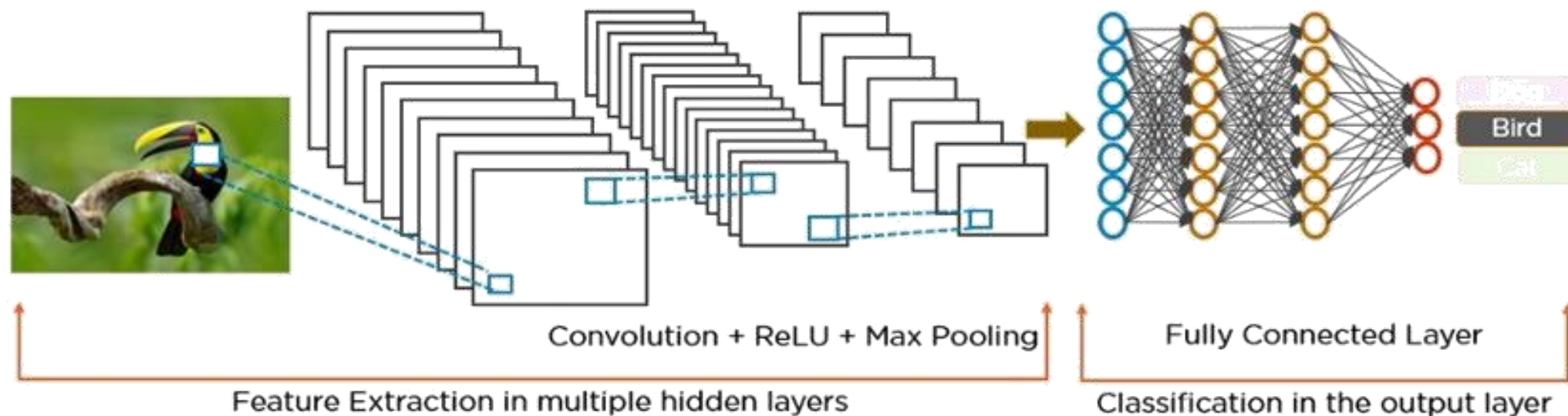
Part 1

認識卷積神經網路 (CNN)



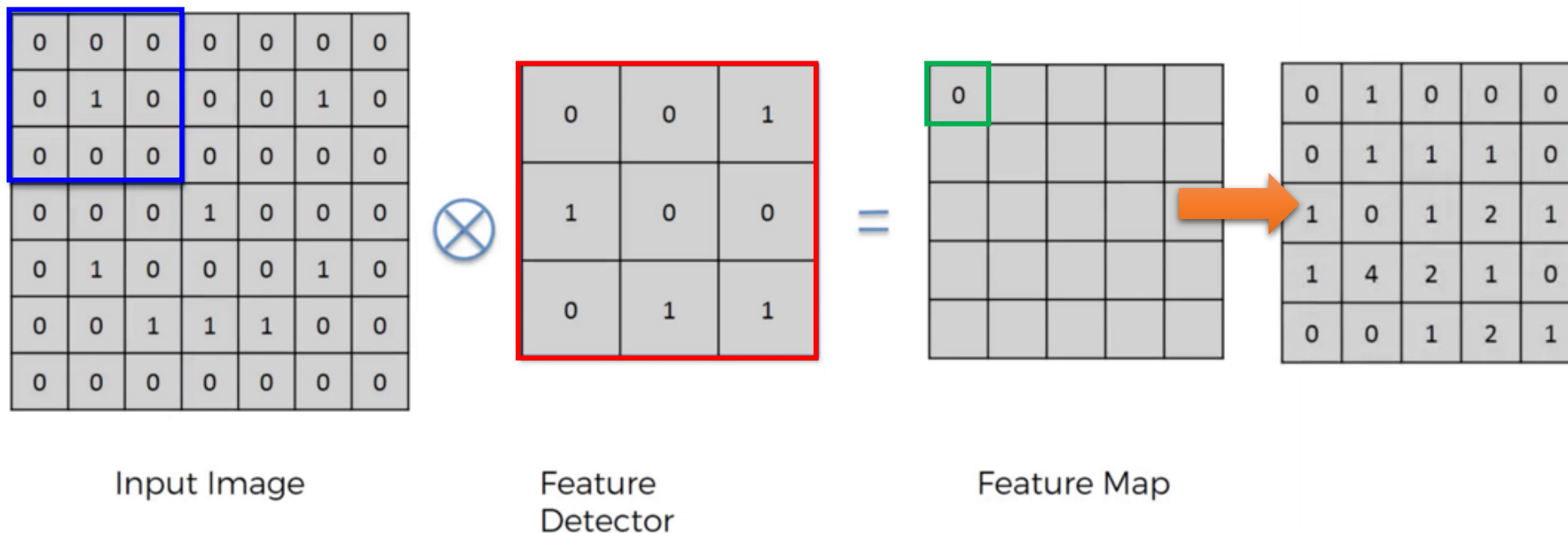
卷積神經網路(Convolutional Neural Network)

- 卷積神經網路(CNN)是一種**前饋神經網路**，其靈感來自於動物視覺皮層組織的神經連接方式。單個神經元只對有限區域內的刺激作出反應，不同神經元的感知區域相互重疊從而覆蓋整個視野。CNN能夠利用輸入資料的**二維結構**，與其他深度學習結構相比，在圖像和語音辨識方面能夠給出更好的結果。
- CNN包括了3個層次：
 - 卷積層(Convolutional layer)**：處理多數的計算、檢查圖像特徵。
 - 池化層(Pooling layer)**：圖像掃描及過濾，降低了許多參數，提高了效率降低了複雜性。
 - 全連接層(Fully connected layer)**：根據前幾層處理程序來提取特徵進行圖像分類的地方，層層相扣每一層都有相連節點。
- 簡單來說，圖片經過**各兩次**的Convolution→Pooling→Fully Connected就是CNN的架構了



Convolution Layer 卷積層

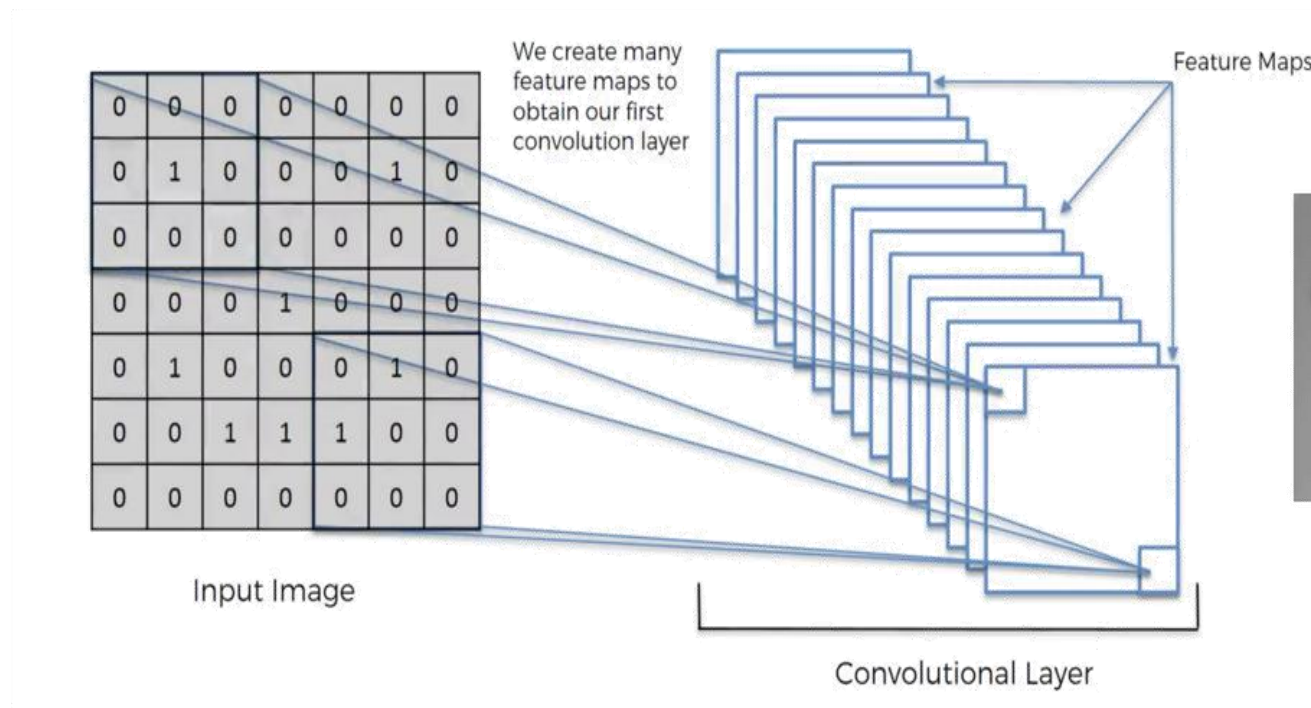
- 卷積運算：從原始圖片中，取出卷積核(Kernel)與特定的Feature Detector(filter)做卷積運算(符號 \otimes)。
- 卷積核的大小：通常為： 3×3 , 5×5 或 7×7 區塊。
- 移動幅度(stride)：積卷核會以向右或向下移動1或2個像素，進行特徵提取，
- 卷積運算就是將下圖兩個 3×3 的矩陣作相乘後再相加，以下圖為例 $0*0 + 0*0 + 0*1 + 0*1 + 1*0 + 0*0 + 0*0 + 0*1 + 0*1 = 0$ 。依序做完整張表，整理成下右圖



[圖文引用] <https://pse.is/4yzkw9>

特徵篩選 Feature Detector(filter)

- Feature Detector(Filter)會隨機產生好幾種(ex:16種)
- Feature Detector的目的就是幫助我們萃取出圖片當中的一些特徵(ex:形狀)，就像人的大腦在判斷這個圖片是什麼東西也是根據形狀來推測。



*

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



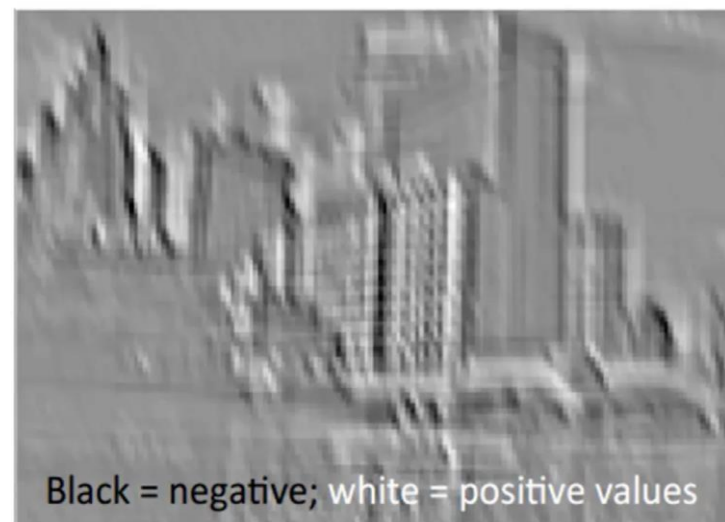
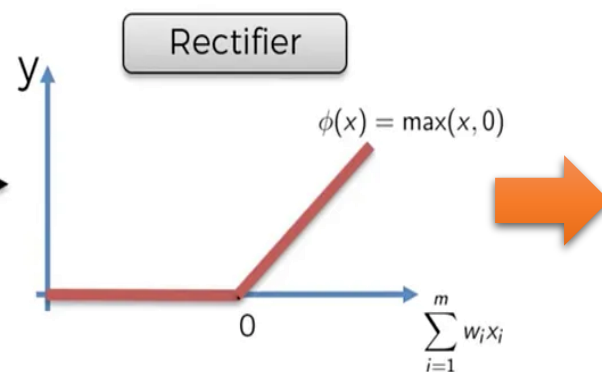
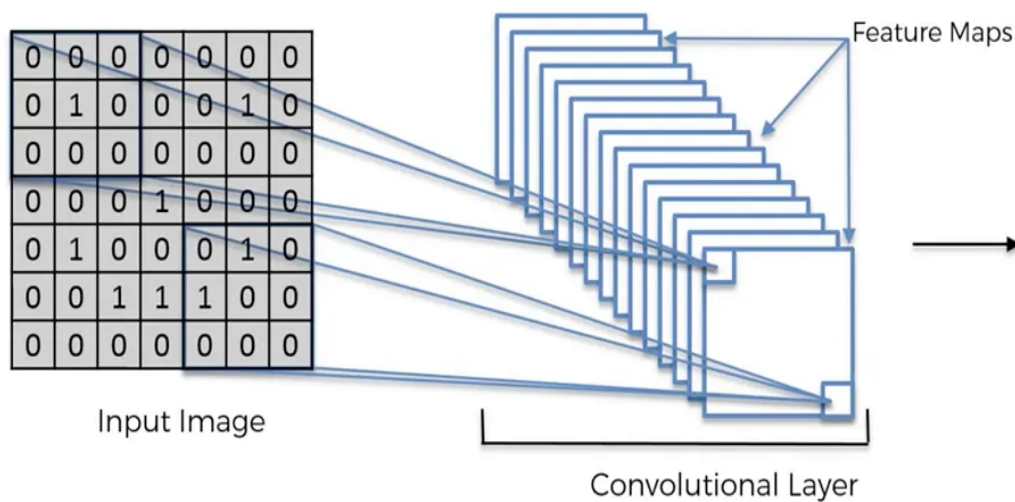
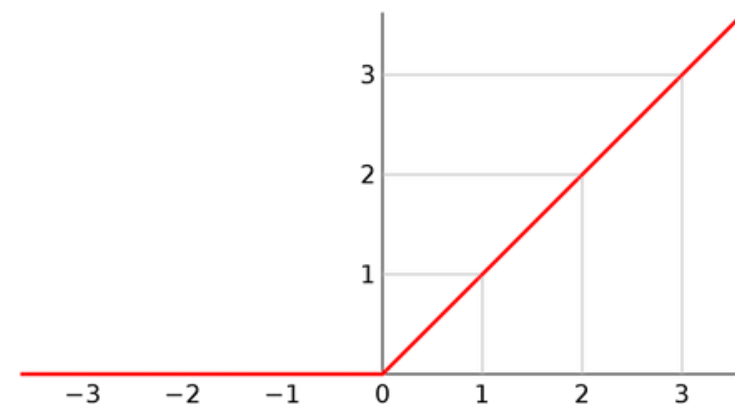
利用Feature Detector萃取出物體的邊界

激勵函數 ReLU (Rectified Linear Unit)

- ReLU為目前最常被使用的激勵函數，數學式型態可表示為：

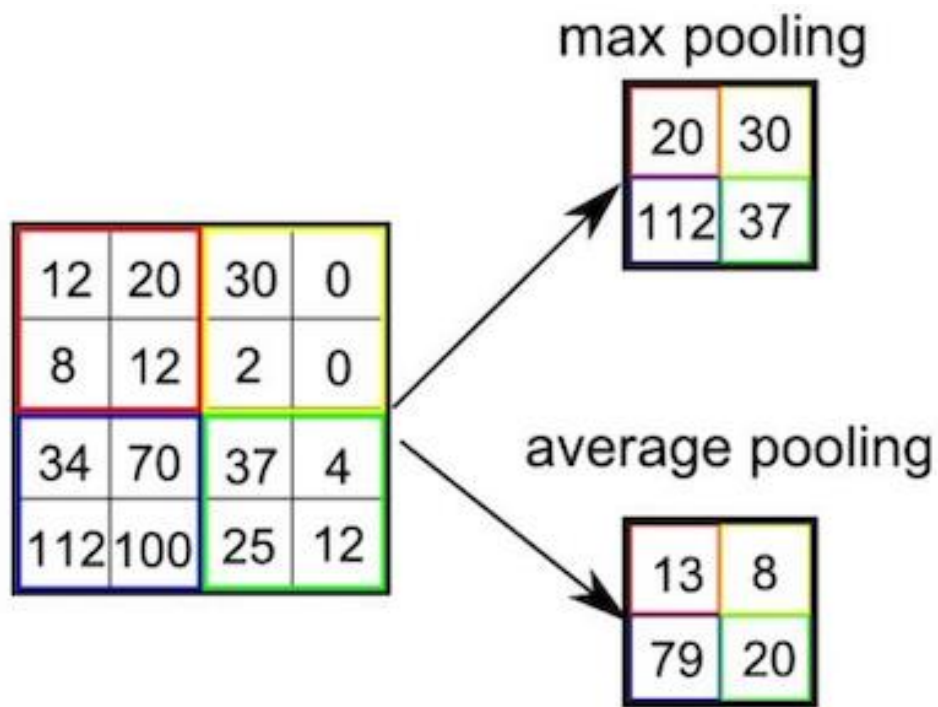
$$f(x) = \max(0, x)$$

- 若輸出值為正值，該輸出值經過ReLU函數轉換仍可正常輸出，若輸出值為負值，則經過ReLU函數轉換後，輸出值為0。
- ReLU可增加類神經網路的稀疏性，可有效減緩過擬合(Overfitting)的狀況發生，也可解決梯度爆炸或梯度消失等問題。
- 使用Relu函數去掉負值，更能淬煉出物體的形狀。



Pooling Layer 池化層

- Pooling Layer主要是採用Max Pooling，只挑出矩陣當中的最大值就好。Max Pooling主要的好處是當圖片整個平移幾個Pixel的話對判斷上完全不會造成影響，以及有很好的抗雜訊功能。
- 池化(Pooling)為減少資料量但同時保存特徵地圖中重要資訊的方法，常見的池化法包含**最大池化法(Max Pooling)**或**平均池化法(Average Pooling)**，其作法顧名思義，便是取一特徵篩選器特定範圍內的最大值或平均值，成為新的特徵矩陣，作法如下所示：



0	1	0	0	0
0	1	1	1	0
1	0	1	2	1
1	4	2	1	0
0	0	1	2	1

Feature Map

Max Pooling

1		

Pooled Feature Map

0	1	0	0	0
0	1	1	1	0
1	0	1	2	1
1	4	2	1	0
0	0	1	2	1

Feature Map

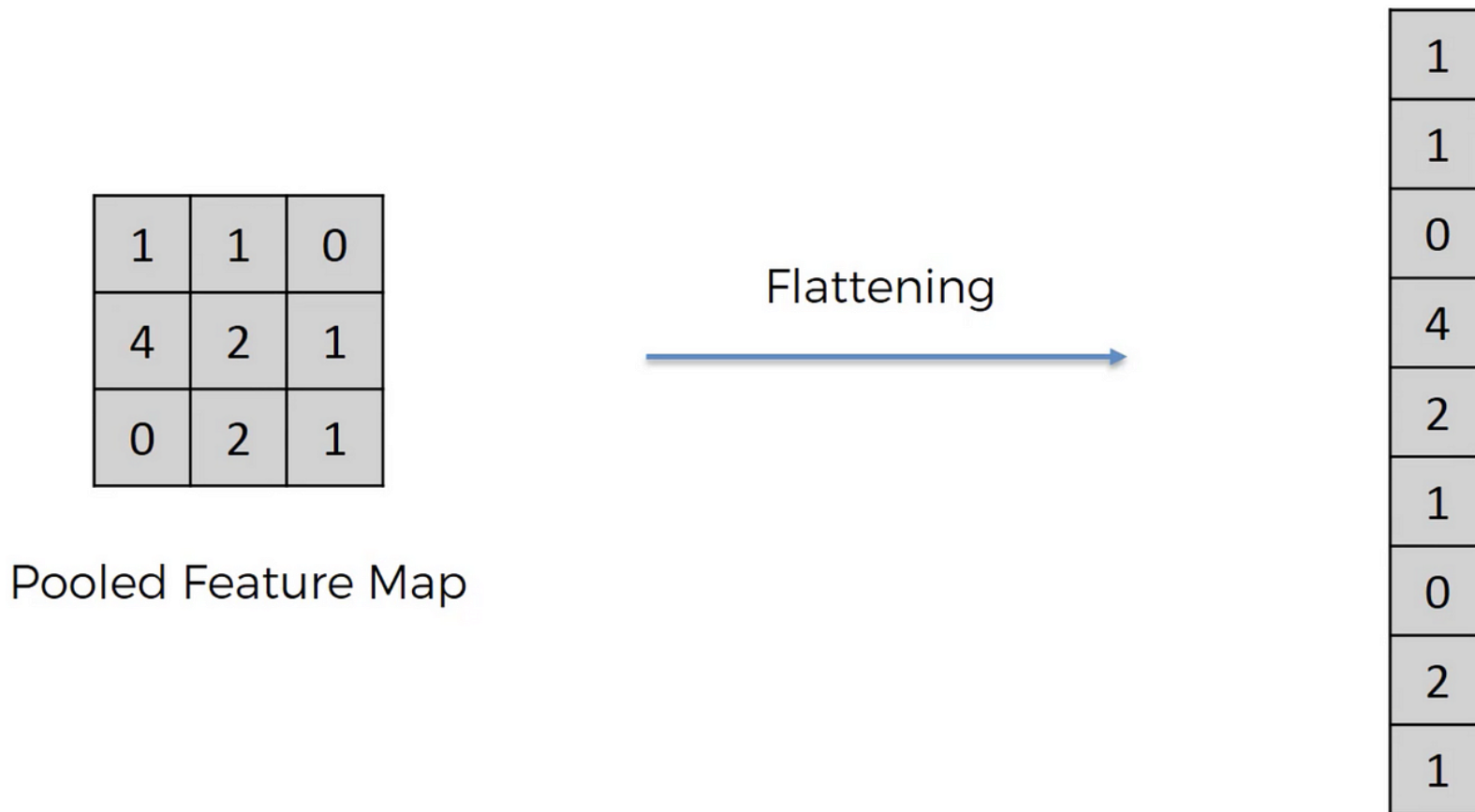
Max Pooling

1	1	0
4	2	1
0	2	1

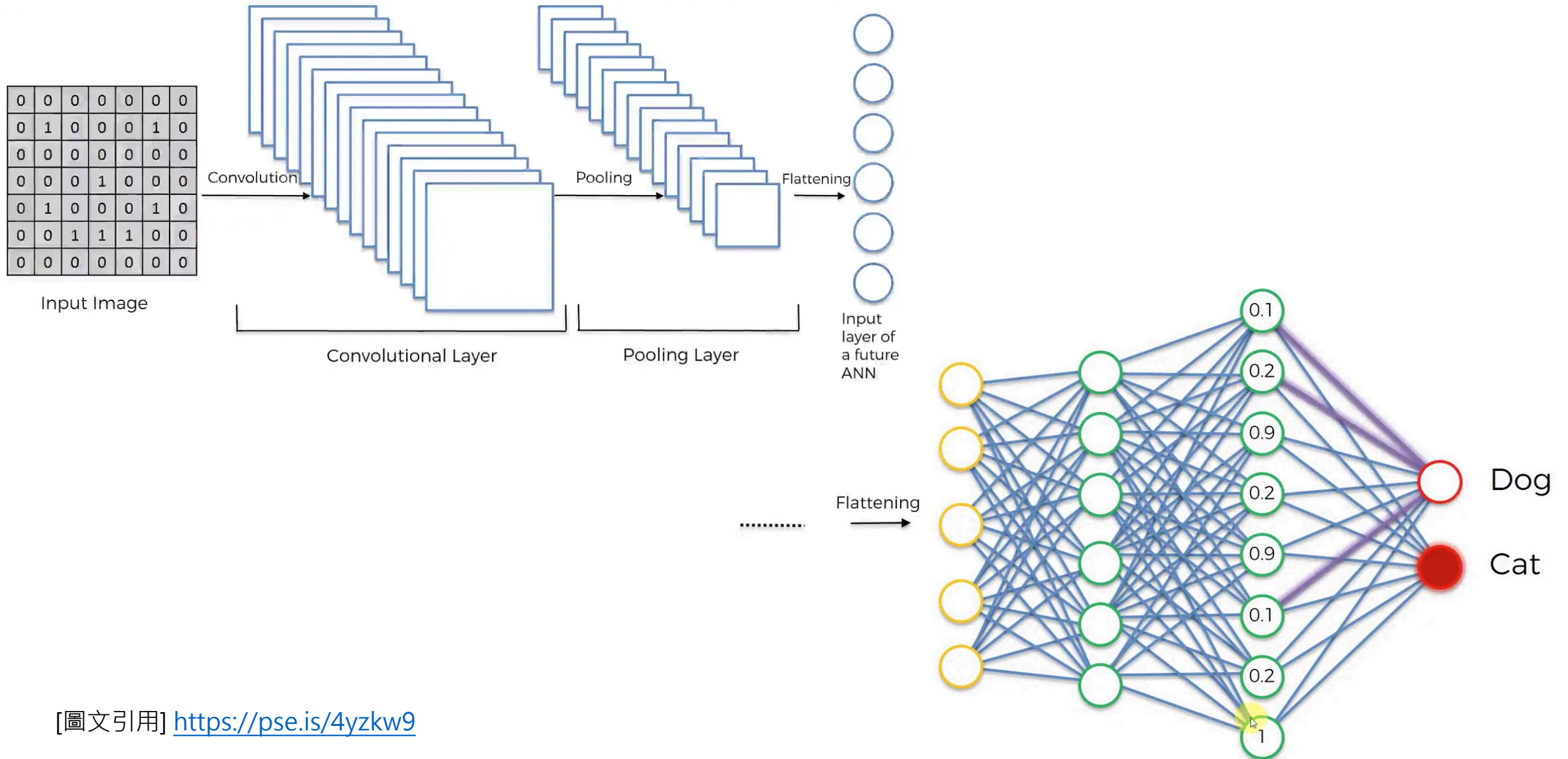
Pooled Feature Map

Fully Connected Layer 全連接層

- 全連接層的部分就是將之前的結果平坦化之後接到最基本的神經網絡。



CNN整體步驟



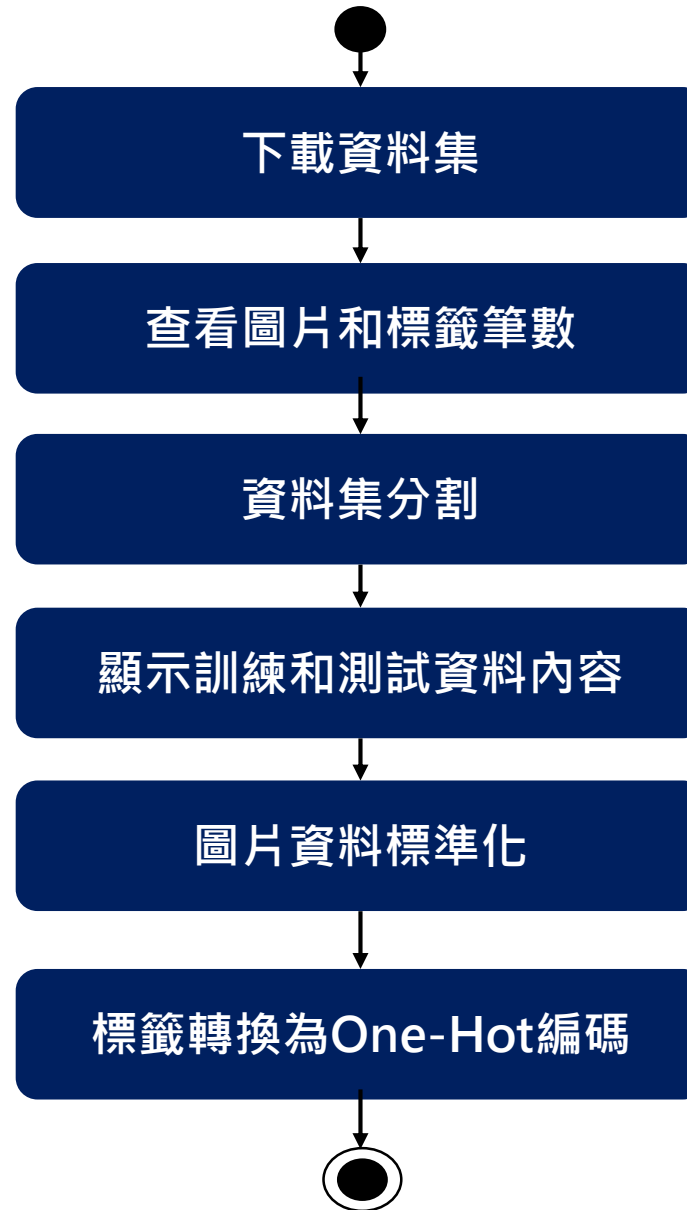
[圖文引用] <https://pse.is/4yzkw9>

Part 2

建立及訓練CNN 相關語法



圖片辨識步驟



載入CNN模型模組

1.載入CNN模型相關模組：

```
from keras.models import Sequential  
from keras.layers import Conv2D, MaxPooling2D, Dropout  
Flatten, Dense
```

2. 建立Sequential物件

```
模型變數 = Sequential()
```

建立第一層卷積層

- 建立卷積層語法：

```
模型變數.add(Conv2D(filters=數值, kernel_size=二維元組,  
padding=填充方式, strides=數值, input_shape=三維元組,  
activation=激勵函式))
```

- **filters**：設定濾鏡個數，每一個 filter 會使用特定卷積核來擷取局部特徵。
- **kernel_size**：設定卷積核尺寸，一般為 3x3 或 5x5。
- **padding**：此參數非必填。設定卷積運算前圖片周圍填充方式，可能值有兩個：
 - 「**same**」表示在圖片周圍填充 0，卷積後圖片與原始圖片相同大小。
 - 「**valid**」表示不填充，卷積後圖片會比原始圖片小。**預設值為 valid**。
- **strides**：此參數非必填。設定移動幅度。**預設值為 1**。
- **input shape**：此參數非必填。若設定此參數，表示建立輸入層及第一層卷積層，參數值為原始圖片的大小；若未設定此參數，表示建立卷積層。
- **activation**：設定激勵函式。

建立第一層池化層和第一層拋棄層

- 建立卷積層語法：

```
模型變數.add(MaxPooling2D(pool_size=二維元組))
```

- **pool_size**：設定池化區大小，一般使用(2,2)、(3,3)或(4,4)。

- 建立第一層拋棄層：

```
model.add(Dropout(0.2))
```

建立平坦層、全連結隱藏層和輸出層

- 建立平坦層及拋棄層語法：

```
model.add(Flatten())  
model.add(Dropout(0.2))
```

- 建立全連結隱藏層語法：

```
model.add(Dense(units=128, activation= 'relu' ))
```

- 建立輸出層：

```
model.add(Dense(units=2, activation= 'softmax' ))
```

- 激勵函數: **Softmax** 會計算所有輸出值的機率，輸出值最大者就是模型預測的類別。

*查看權重

- 查看模型權重數量的語法：

模型變數.summary()

Model: "sequential"

Layer (type)	Output Shape	Param #	
conv2d (Conv2D)	(None, 80, 80, 8)	608	$5 \times 5 \times 3 \times 8 + 16 = 608$
max_pooling2d (MaxPooling2D)	(None, 40, 40, 8)	0	
dropout (Dropout)	(None, 40, 40, 8)	0	
conv2d_1 (Conv2D)	(None, 40, 40, 16)	3216	$5 \times 5 \times 8 \times 16 + 16 = 3216$
max_pooling2d_1 (MaxPooling2D)	(None, 20, 20, 16)	0	
dropout_1 (Dropout)	(None, 20, 20, 16)	0	
conv2d_2 (Conv2D)	(None, 20, 20, 32)	12832	$5 \times 5 \times 16 \times 32 + 32 = 12832$
max_pooling2d_2 (MaxPooling2D)	(None, 10, 10, 32)	0	
dropout_2 (Dropout)	(None, 10, 10, 32)	0	
flatten (Flatten)	(None, 3200)	0	
dropout_3 (Dropout)	(None, 3200)	0	
dense (Dense)	(None, 128)	409728	$3200 \times 128 + 128 = 409728$
dense_1 (Dense)	(None, 2)	258	$128 \times 2 + 2 = 258$
=====			
Total params: 426,642	權重總數		
Trainable params: 426,642			
Non-trainable params: 0			