

Day 94

深度學習應用卷積神經網路

卷積神經網路 - 卷積(Convolution)層與參數調整



出題教練

陳宇春



知識地圖 卷積網路套件練習

卷積(Convolution)層與參數調整

深度神經網路
Supervised Learning Deep Neural Network (DNN)

簡介 Introduction

套件介紹 Tools: Keras

組成概念 Concept

訓練技巧 Training Skill

應用案例 Application

卷積神經網路
Convolutional Neural Network (CNN)

簡介 introduction

套件練習 Practice with Keras

訓練技巧 Training Skill

電腦視覺 Computer Vision

卷積類神經網路套件練習
Practice CNN with Keras

建立 CNN 模型

Keras 中的 CNN Layers

使用 CNN 完成 CIFAR-10 預測

本日知識點目標

- 了解CNN Flow
- 卷積 (Convolution) 的 超參數(Hyper parameter)設定與應用

卷積 (Convolution) 的 超參數(Hyper parameter)

- 卷積 (Convolution) 的 超參數(Hyper parameter)
 - 卷積內核 (kernel)
 - Depth (kernels的總數)
 - Padding (是否加一圈0值的pixel)
 - Stride(選框每次移動的步數)

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

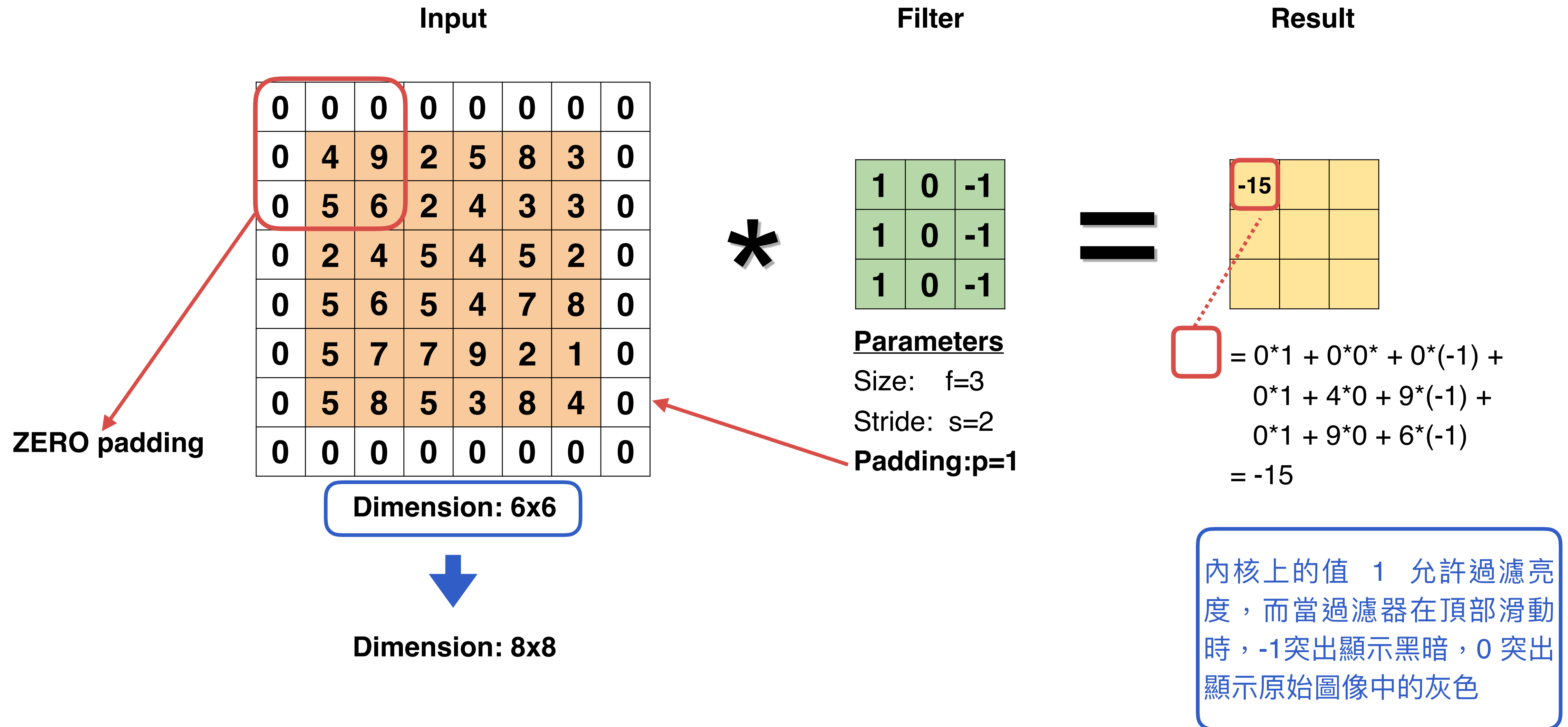
(卷積引數：filter數量，filter長，filter寬，輸入影象的三維 (RGB，長，寬))

給定了 kernels 的 深度與維度

填充或移動步數(Padding/Stride)的用途

- RUN 過 CNN，兩個問題
 - 是不是卷積計算後，卷積後的圖是不是就一定只能變小？
 - 可以選擇維持一樣大
 - 卷積計算是不是一次只能移動一格？
- 控制卷積計算的圖大小 - Valid and Same convolutions
 - **padding = 'VALID'** 等於最一開始敘述的卷積計算，圖根據 **filter** 大小和 **stride** 大小而變小
 - $\text{new_height} = \text{new_width} = (W - F + 1) / S$
 - **padding = 'Same'** 的意思就是要讓輸入和輸出的大小是一樣的
 - pad=1，表示圖外圈額外加 1 圈 0，假設 pad=2，圖外圈額外加 2 圈 0，以此類推

加了填充(padding) 之後

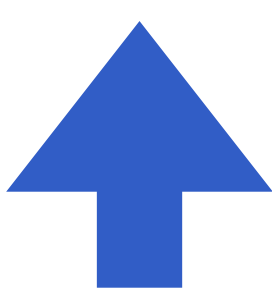


舉例

```
Model.add(Convolution2D(32, 3, 3), input_shape=(1, 28, 28), strides=2,  
padding='valid')
```

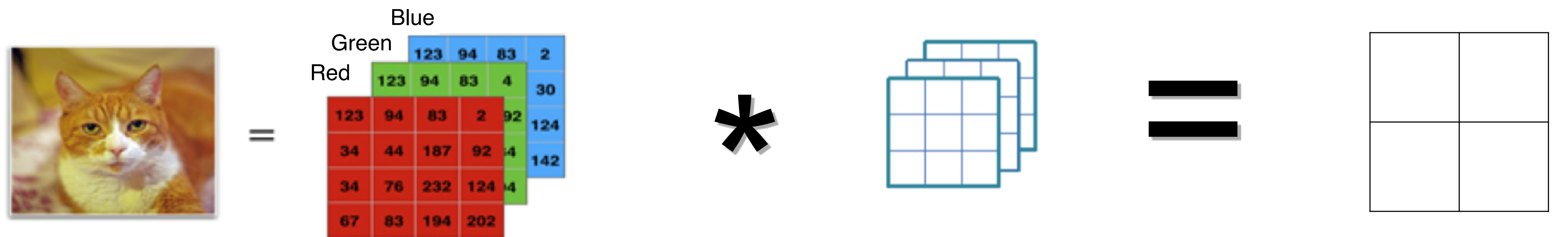
- 這代表卷積層 filter 數設定為 32，filter 的 kernel size 是 3，步伐 stride 是 2，pad 是 1。
 - pad = 1，表示圖外圈額外加 1 圈 0，假設 pad = 2，圖外圈額外加 2 圈 0，以此類
 - (1)kernel size是 3 的時候，卷積後圖的寬高不要變，pad 就要設定為 1
 - (2)kernel size是 5 的時候，卷積後圖的寬高不要變，pad 就要設定為 2

Input = 8x8
Kernel =32x3x3
Pad = p
Output data?


$$(New)^n = \frac{n+2p-f}{S}$$

多個通道(channels)的卷積作法

- 考慮多種顏色- 針對 RGB
- 會有3個對應的 kernel



輸入的原圖 $4 \times 4 \times 3$

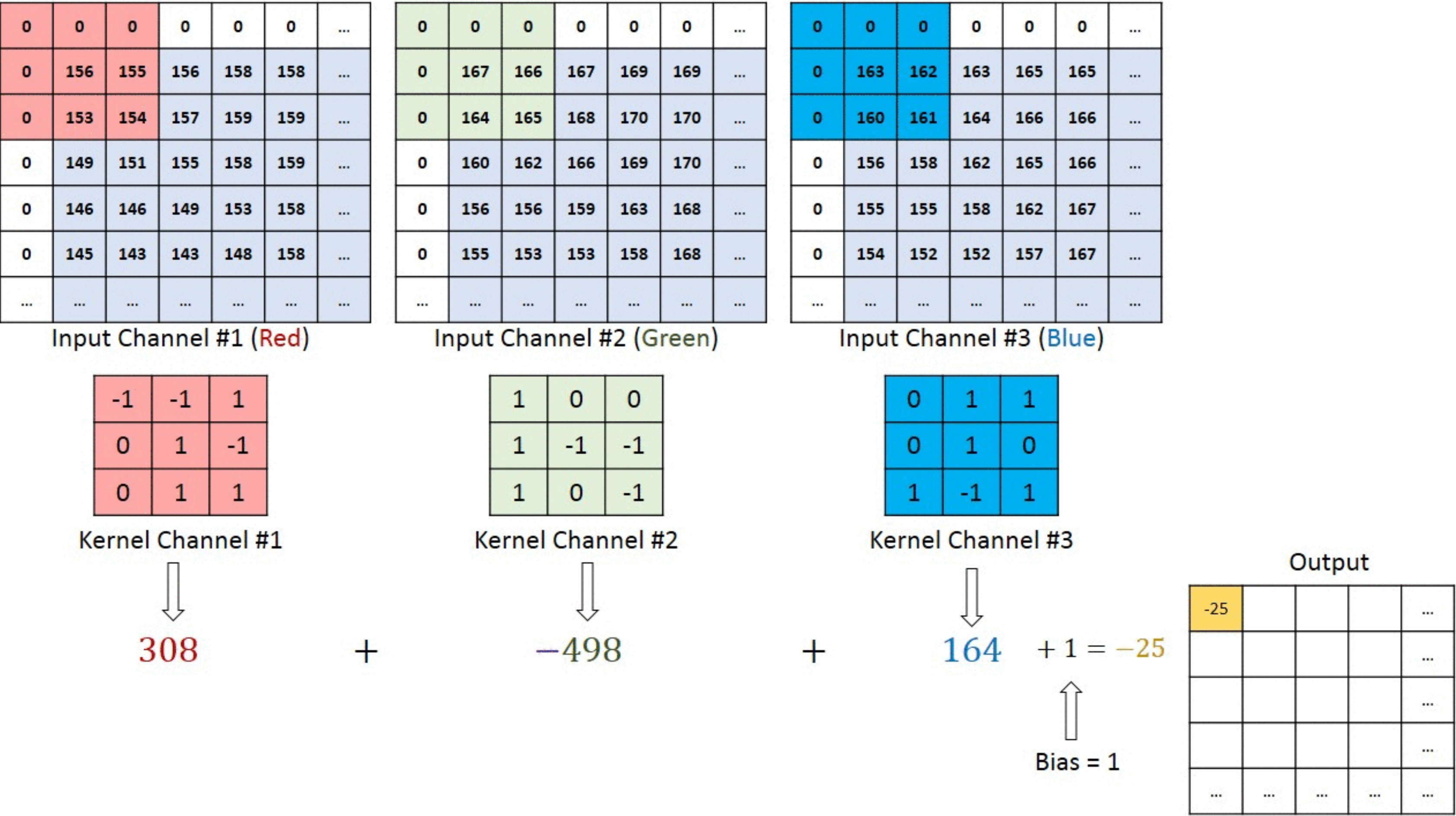
內核 $3 \times 3 \times 3$

特徵圖 $2 \times 2 \times 1$

長 x 寬 x 通道 (深度)

要一樣

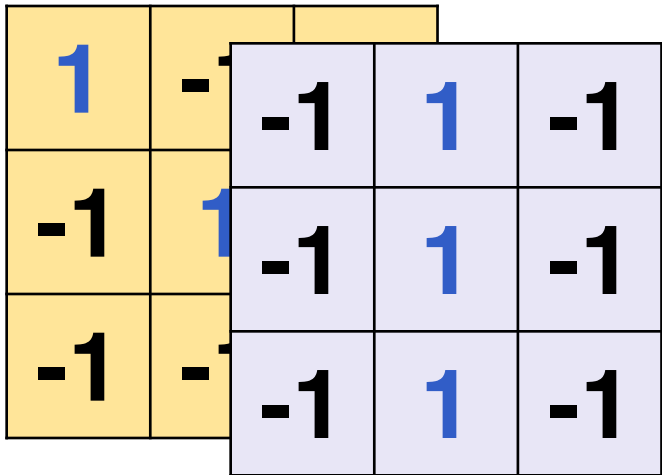
多個通道(channels)的卷積作法



前述流程 / python程式 對照

Filter number 由少
input side到多

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

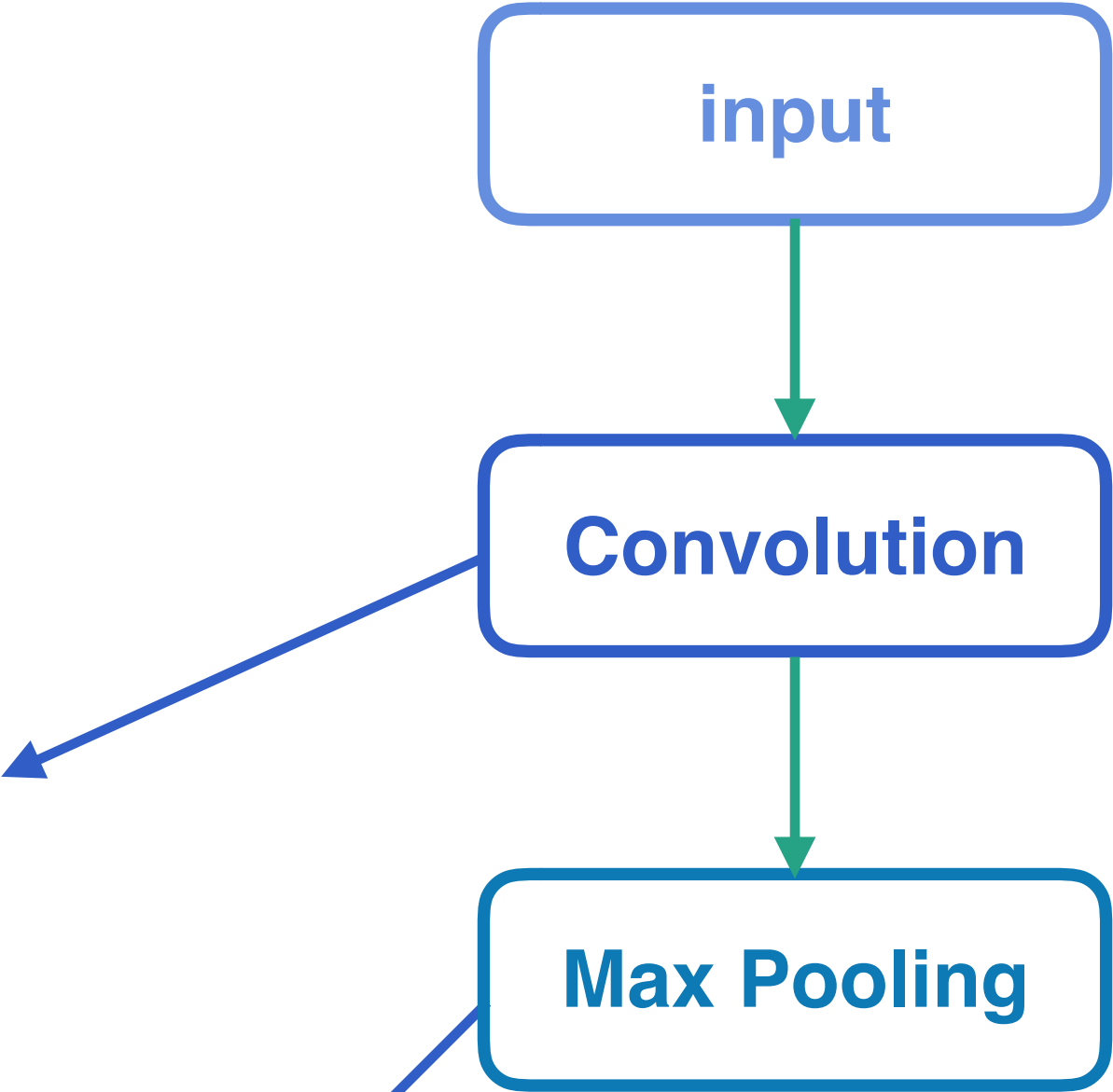
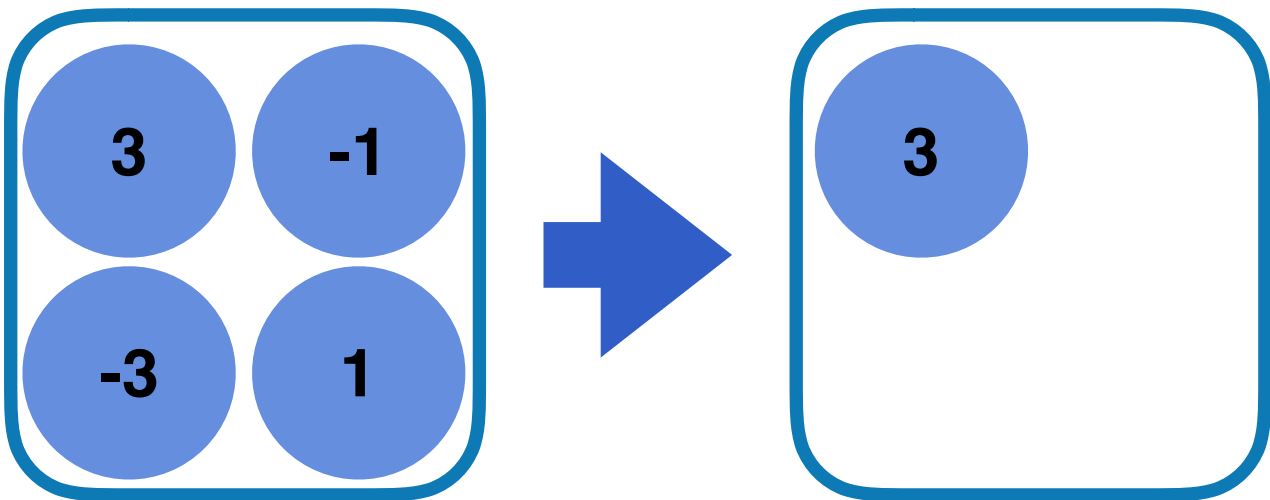


.....There are 25 3x3 filters.

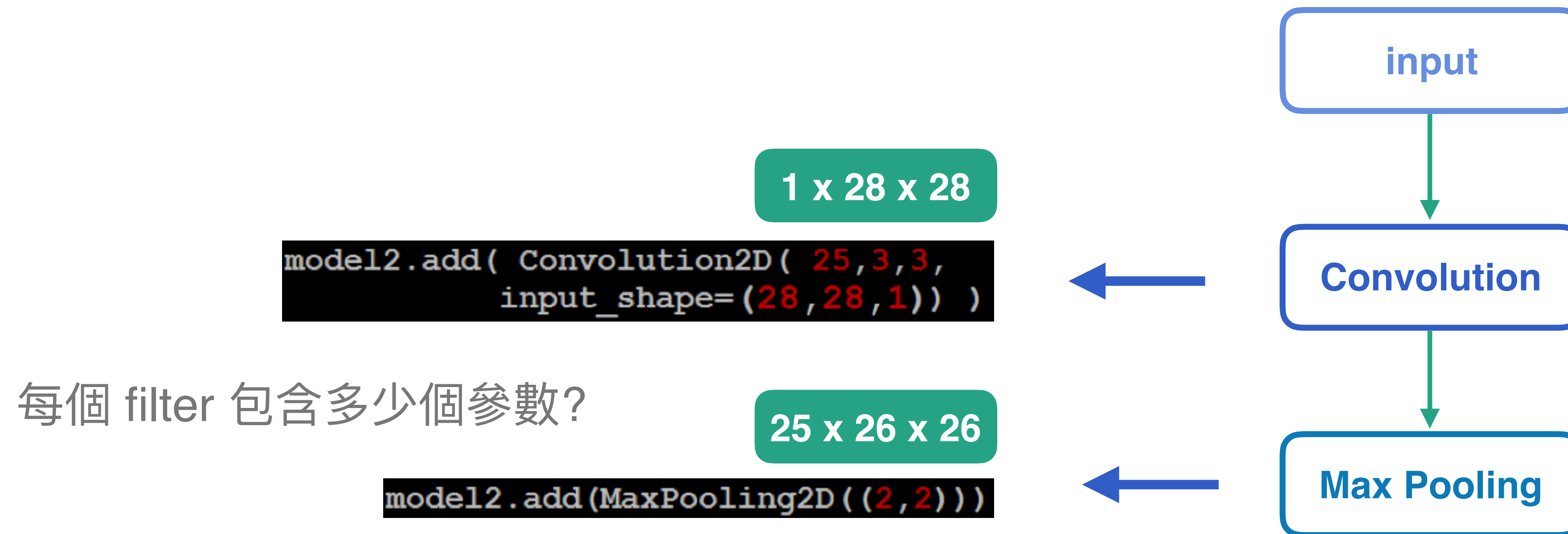
Input_shape = (28 , 28 , 1)

28 x 28 pixels 1: black/white, 3 : RGB

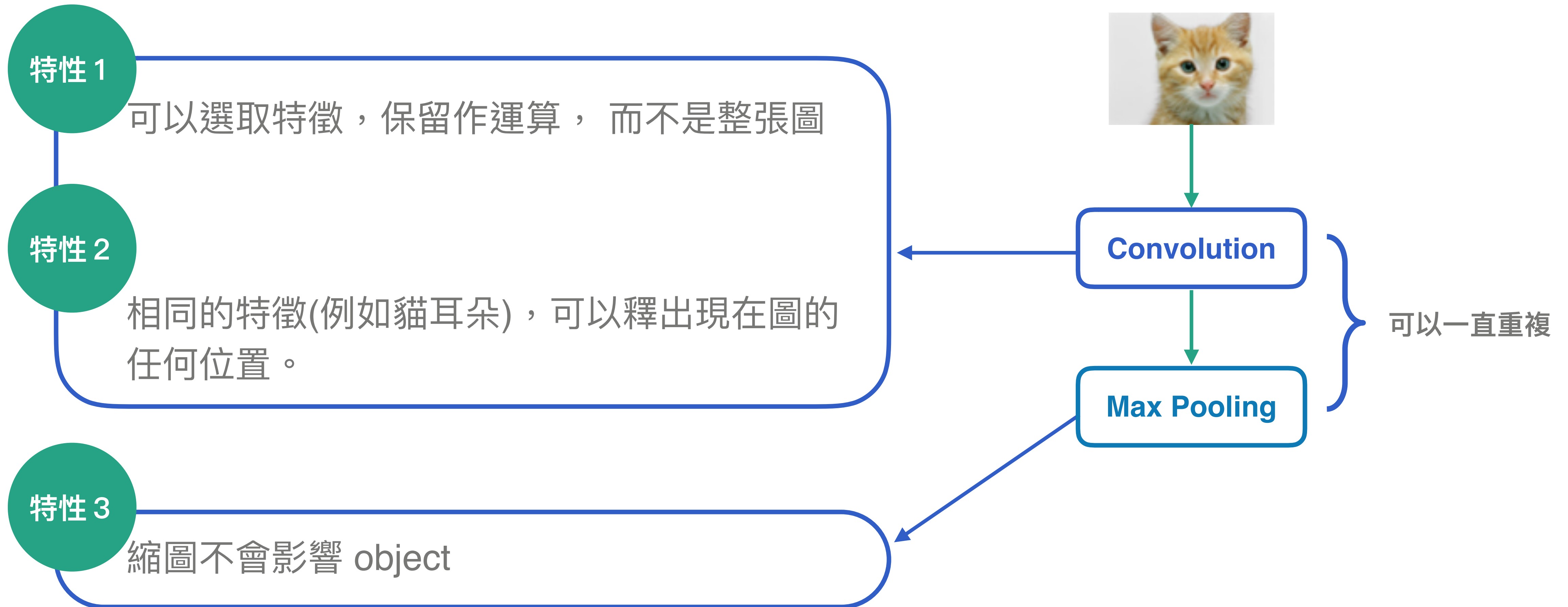
```
model2.add(MaxPooling2D( (2,2) ))
```



前述流程 / python程式 對照



重要知識點複習：卷積(Convolution) 跟池化(Pooling)



卷積神經網路(CNN)特性

- 適合用在影像上
 - 因為 fully-connected networking 如果用在影像辨識上，會導致參數過多 (因為像素很多)，導致 over-fitting
 - CNN 針對影像辨識的特性，特別設計過，來減少參數
 - Convolution：學出 filter 比對原始圖片，產生出 feature map (也當成 image)
 - Max Pooling：將 feature map 縮小
 - Flatten：將每個像素的 channels (有多少個 filters) 展開成 fully connected feedforward network

解題時間 It's Your Turn

請跳出PDF至官網Sample Code & 作業
開始解題

