

Day 95

# 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

## 池化(Pooling)層與參數調整

深度神經網路  
Supervised LearningDeep 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

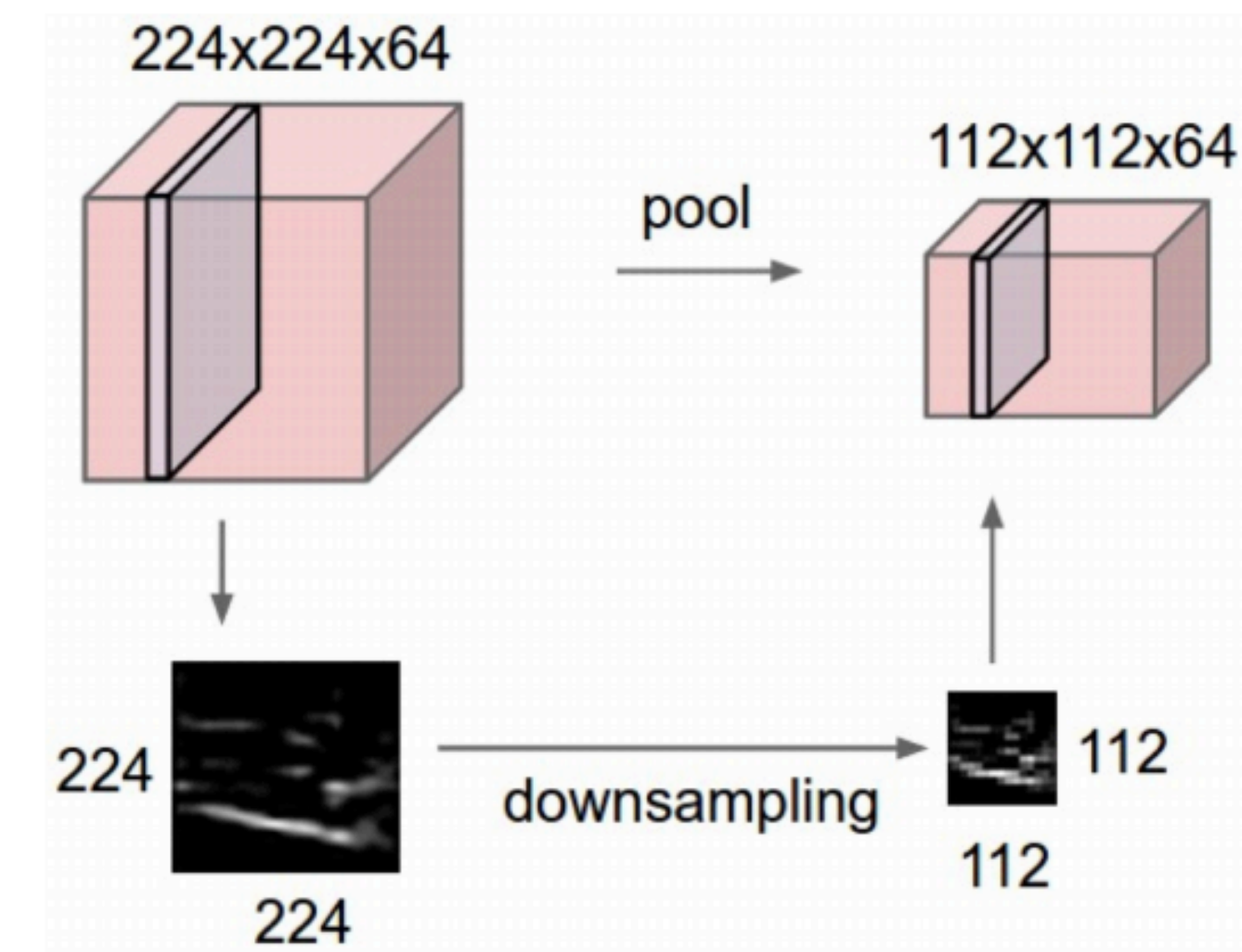
- 了解 CNN Flow
- 池化層超參數的調適

# 池化層(Pooling Layer) 如何調用

- 以 Keras 為例
- `keras.layers.MaxPooling2D(pool_size=(2, 2), strides=None, padding='valid', data_format=None)`
  - **pool\_size**：整數，沿（垂直，水平）方向縮小比例的因數。
    - (2, 2)會把輸入張量的兩個維度都縮小一半。
  - **strides**：整數，2 個整數表示的元組，或者是"None"。表示步長值。
    - 如果是 None，那麼默認值是 pool\_size。
  - **padding**："valid"或者"same"（區分大小寫）。
  - **data\_format**：channels\_last(默認)或 channels\_first 之一。表示輸入各維度的順序
    - channels\_last 代表尺寸是(batch, height, width, channels)的輸入張量。
    - channels\_first 代表尺寸是(batch, channels, height, width)的輸入張量。

# 池化層(Pooling Layer) 超參數

- 前端輸入feature map 維度： $W1 \times H1 \times D1$
- 有兩個hyperparameters：
  - Pooling filter 的維度-  $F$ ,
  - 移動的步數  $S$ ,
- 所以預計生成的輸出是  $W2 \times H2 \times D2$ :
  - $W2 = (W1 - F) / S + 1$
  - $H2 = (H1 - F) / S + 1$
  - $D2 = D1$



圖片來源：cnblogs

# 池化層(Pooling Layer) 常用的類型

- Pooling Layer 常用的類型:
- Max pooling (最大池化)
- Average pooling (平均池化)

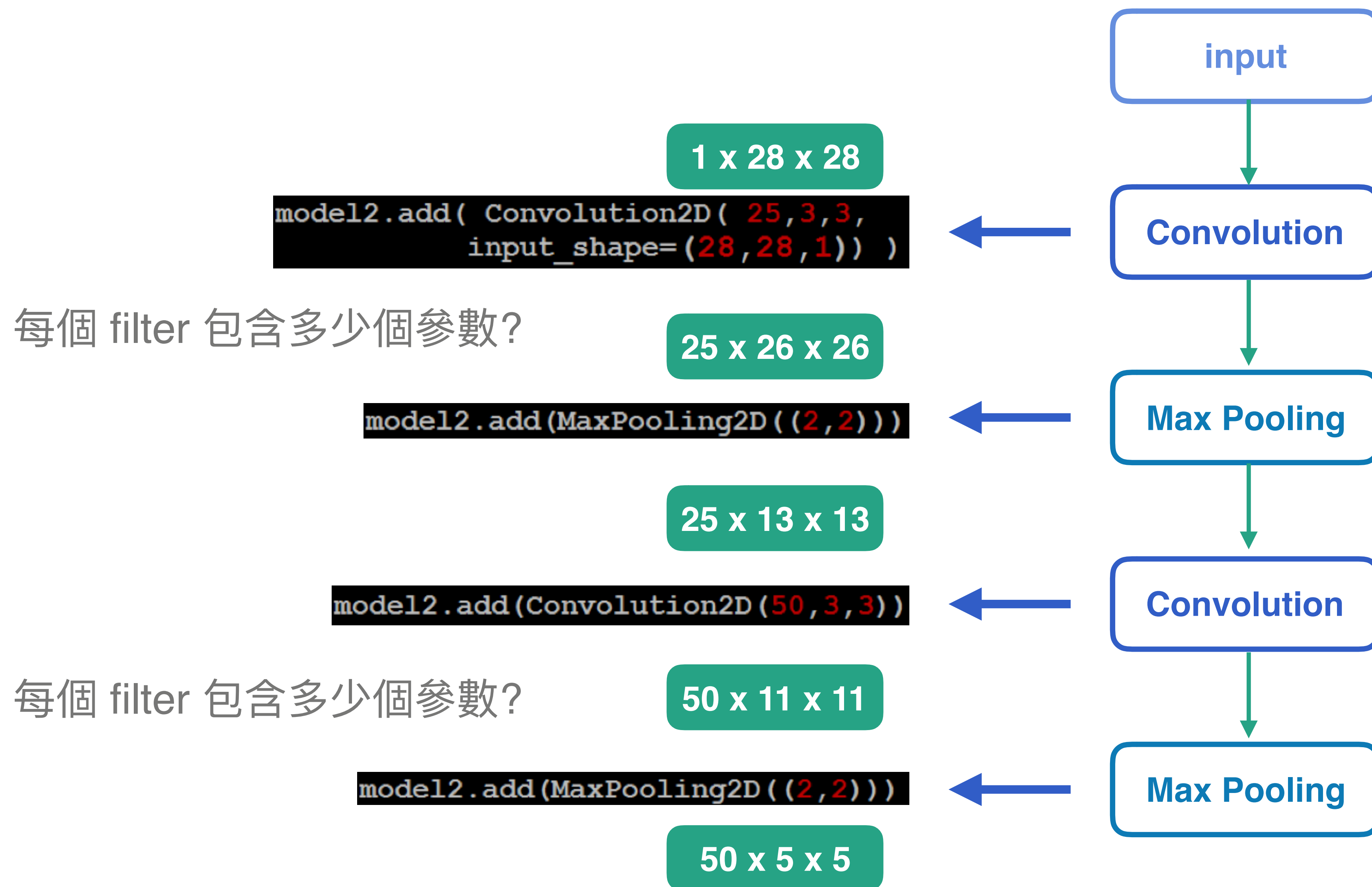
2	3	1	9
4	7	3	5
8	2	2	2
1	3	4	5

(pool\_size=(2, 2),  
strides=None)

7	9
8	5

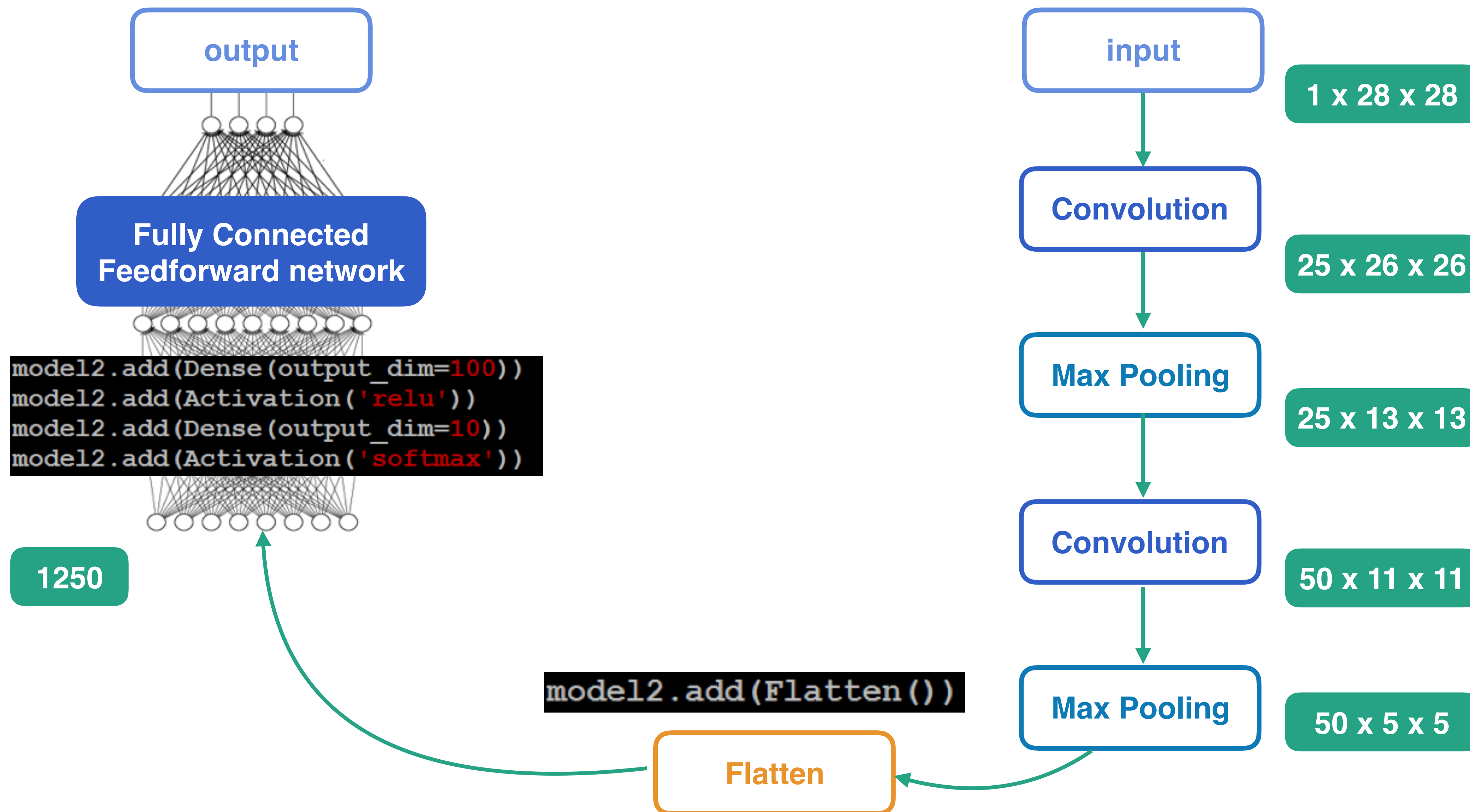
4	4.5
3.5	3.25

# 建立CNN Model by Keras (I)



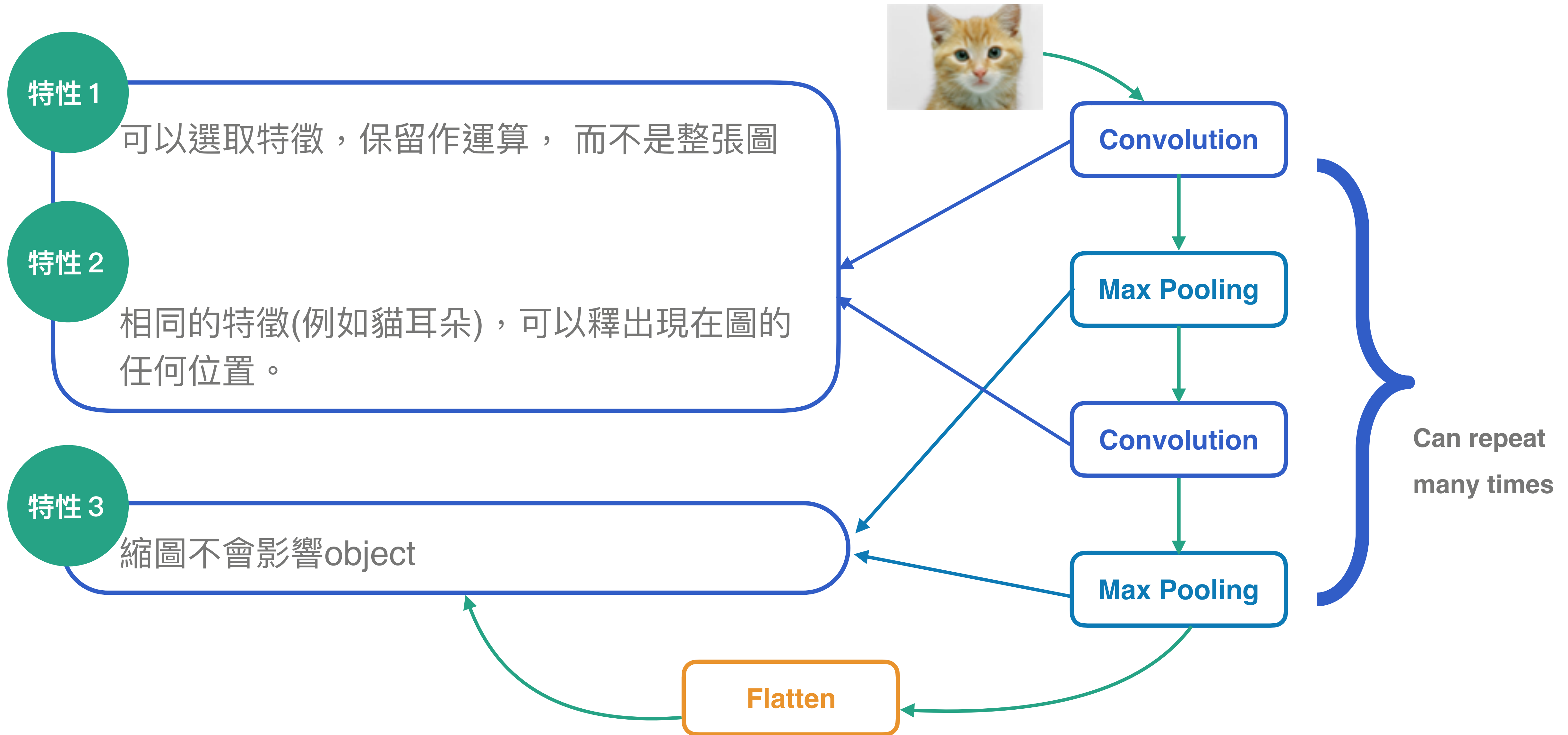


# 建立CNN Model by Keras (II)





# 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 (全連接層) 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(過度擬合)
  - CNN 針對影像辨識的特性，特別設計過，來減少參數
  - Convolution(卷積)：學出 filter 比對原始圖片，產生出 feature map (特徵圖, 也當成image)
  - Max Pooling (最大池化)：將 feature map 縮小
  - Flatten (平坦層)：將每個像素的 channels (有多少個filters) 展開成 fully connected feedforward network (全連接的前行網路)
- AlphaGo 也用了 CNN，但是沒有用 Max Pooling (所以不同問題需要不同 model)

# Pooling Layer (池化層) 適用的場景

---

- 特徵提取的誤差主要來自兩個方面：
  - (1) 鄰域大小受限造成的估計值方差增大；
  - (2) 卷積層超參數與內核造成估計均值的偏移。
- 一般來說，
  - average-pooling 能減小第一種誤差，更多的保留圖像的背景信息
  - max-pooling 能減小第二種誤差，更多的保留紋理信息

# 解題時間 It's Your Turn

請跳出PDF至官網Sample Code & 作業  
開始解題

