



Day 95

深度學習應用卷積神經網路

卷積神經網路 - 池化(Pooling)層與參數調整



陳宇春

出題教練



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 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

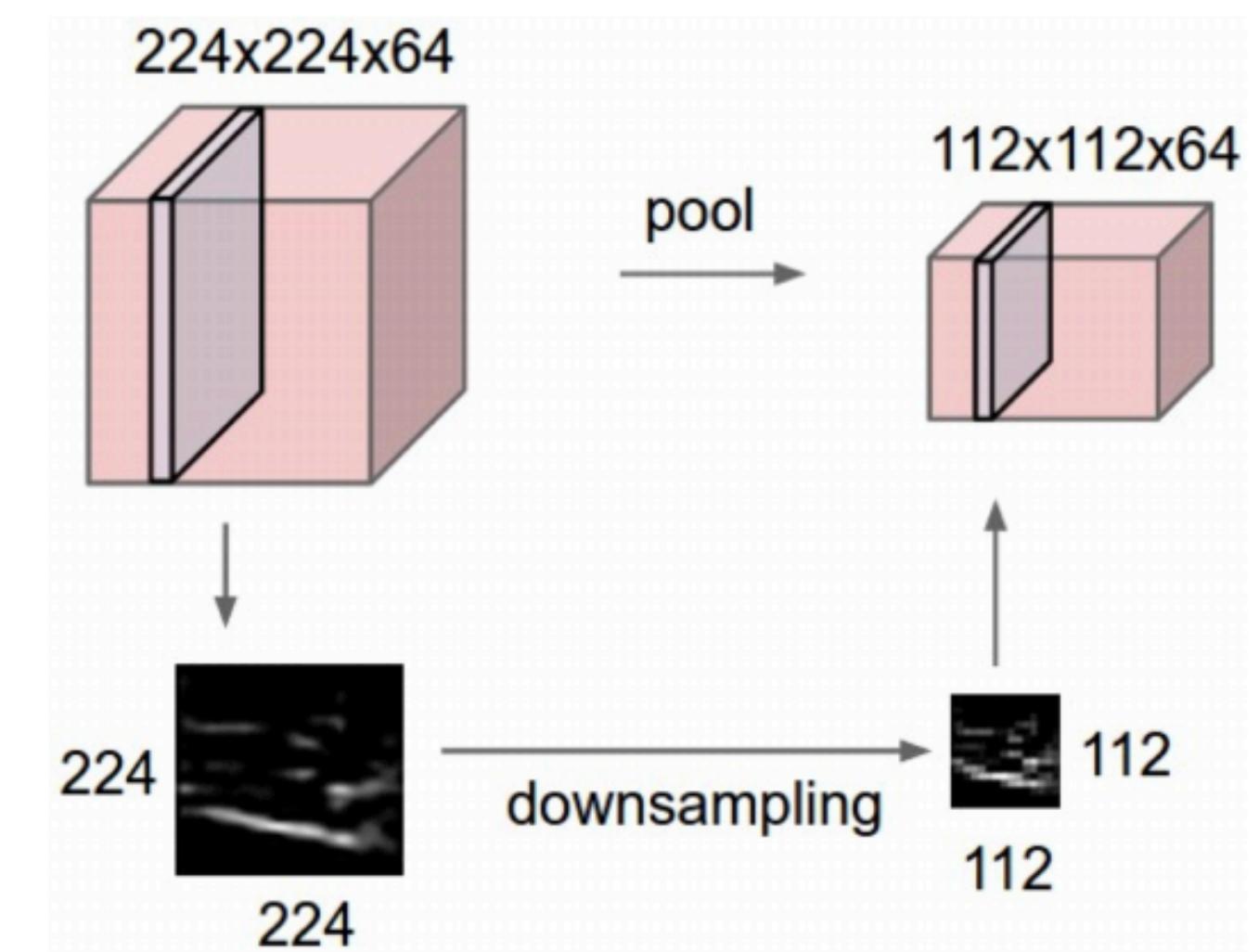
- 了解 CNN Flow
- 池化層超參數的調適

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： $W_1 \times H_1 \times D_1$
- 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 $W_2 \times H_2 \times D_2$:
 - $W_2 = (W_1 - F)/S + 1$
 - $H_2 = (H_1 - F)/S + 1$
 - $D_2 = D_1$



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)

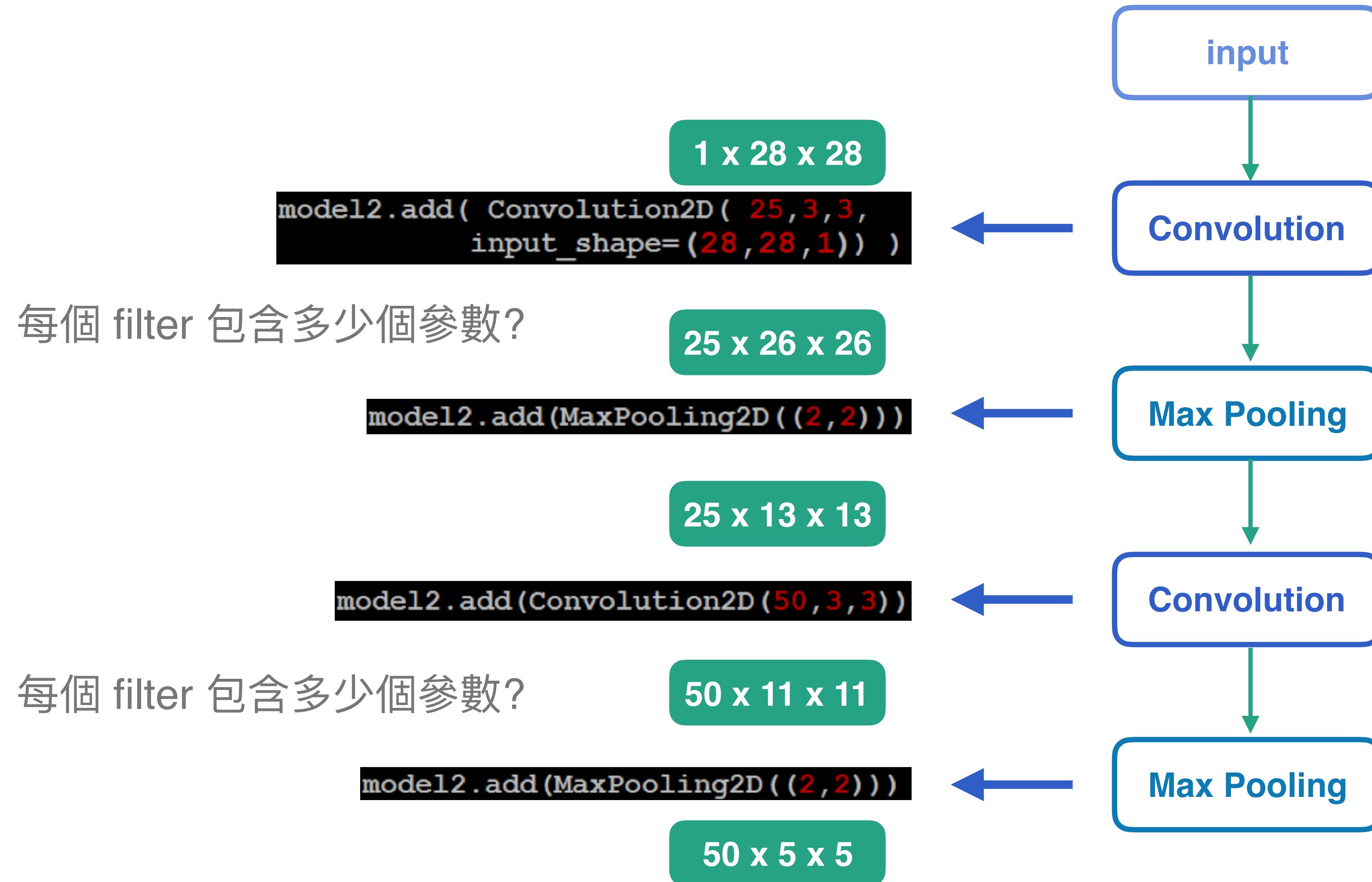
2	3	1	9
4	7	3	5
8	2	2	2
1	3	4	5

7	9
8	5

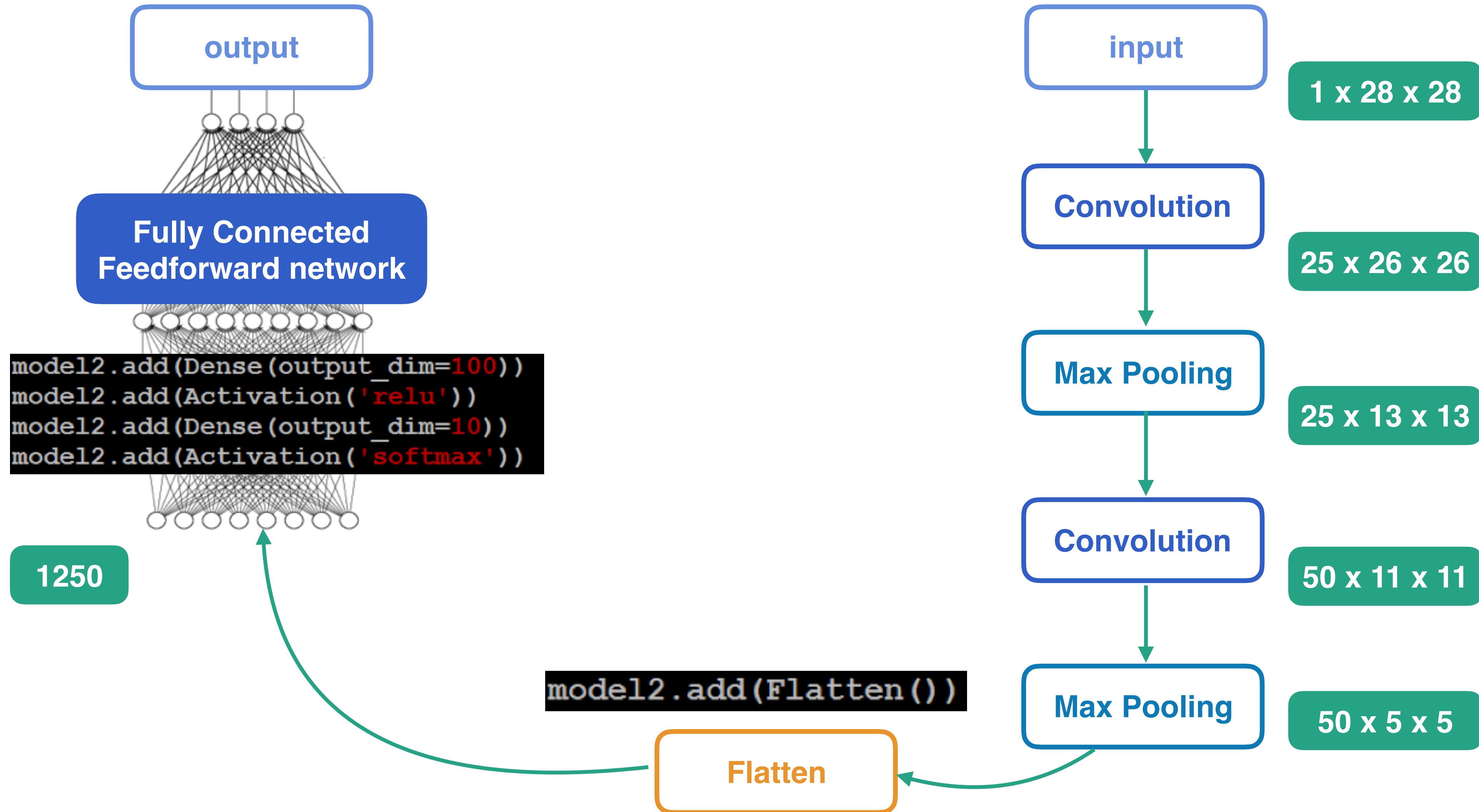
(pool_size=(2, 2),
strides=None)

4	4.5
3.5	3.25

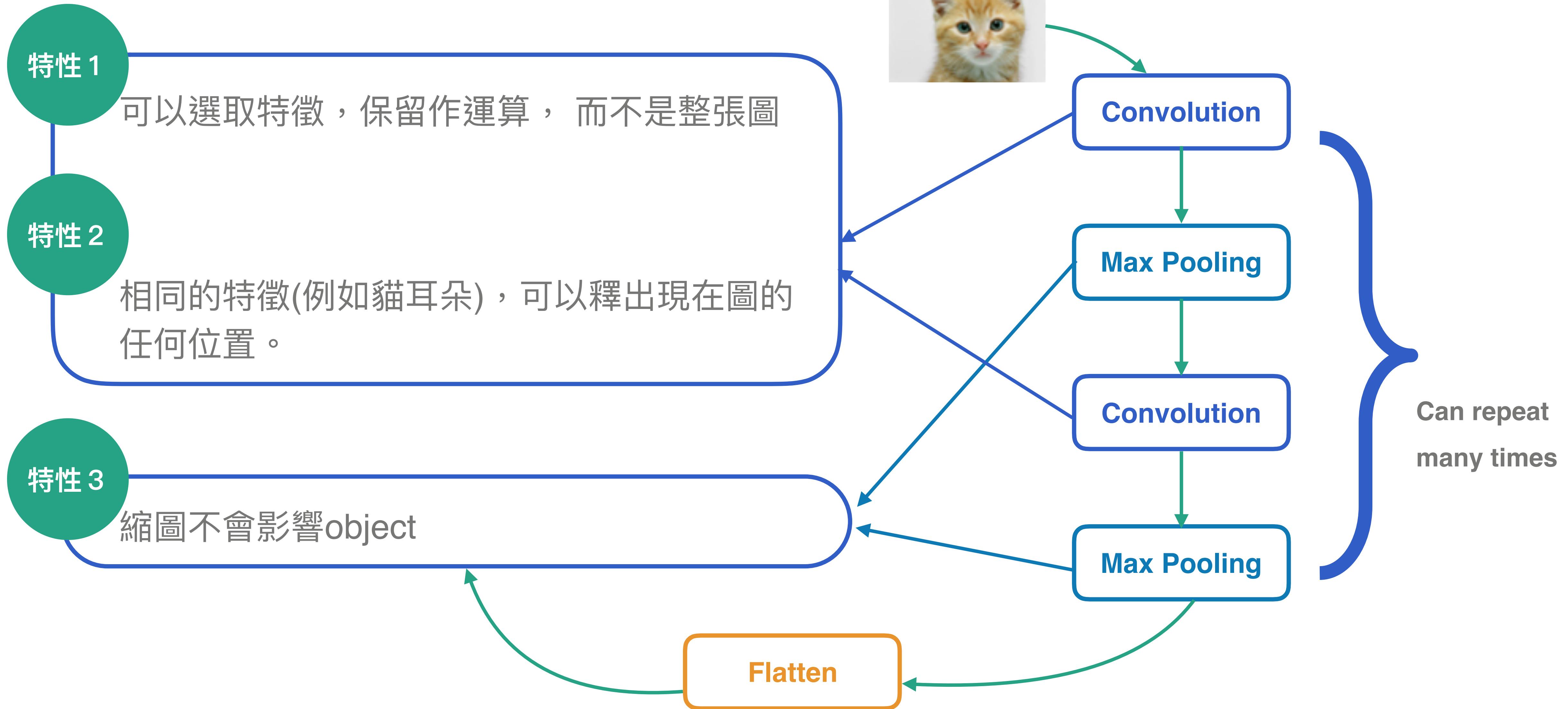
建立CNN Model by Keras (I)



建立CNN Model by Keras (II)



重要知識點複習：卷積Convolution 跟 池化Pooling

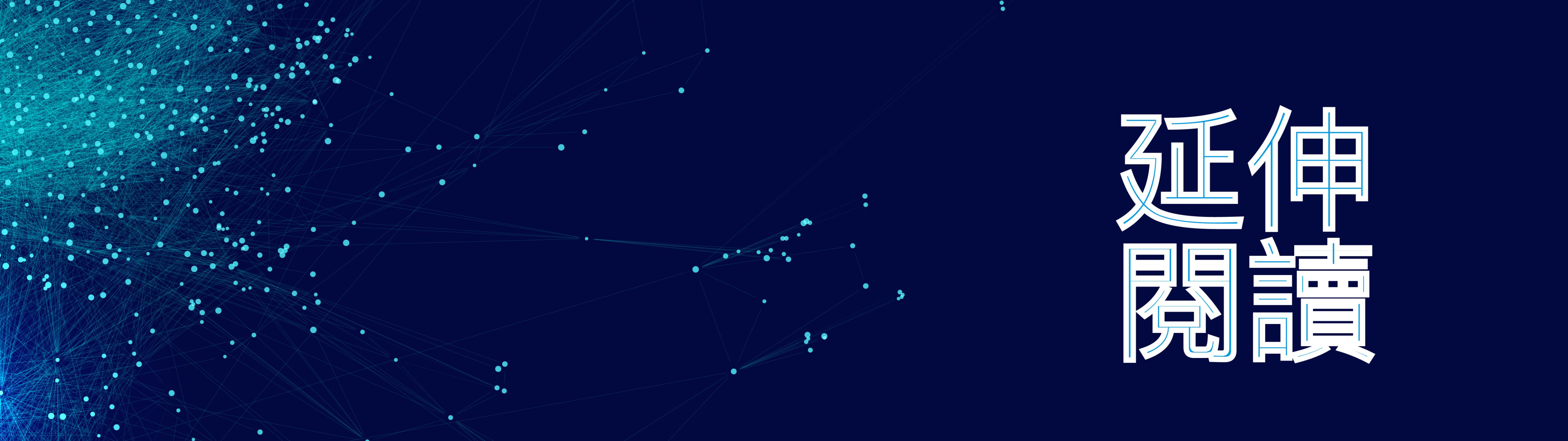


卷積神經網路(CNN)特性

- 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



延伸 閱讀

除了每日知識點的基礎之外，推薦的延伸閱讀能補足學員們對該知識點的了解程度，建議您解完每日題目後，若有
多餘時間，可再補充延伸閱讀文章內容。

推薦延伸閱讀

以 Tensorflow 為例

```
# 池化
def max_pool_2x2(x):
    return tf.nn.max_pool(x, ksize=[1, 2, 2, 1], strides=[1, 2, 2, 1], padding='SAME')

# 池化跟卷積的情況有點類似
# x 是卷積後,有經過非線性啟動後的圖像,
# ksize 是池化滑動張量
# ksize 的維度[batch, height, width, channels],跟 x 張量相同
# strides [1, 2, 2, 1],與上面對應維度的移動步長
# padding與卷積函數相同,padding='VALID',對原圖像不進行0填充
```



推薦延伸閱讀

基於Keras的卷積神經網路 (CNN) 視覺化

本文整理自Deep Learning with Python，書本上完整的代碼在這裡的5.4節，介紹三種視覺化方法：

- 卷積核輸出的視覺化(Visualizing intermediate convnet outputs (intermediate activations))，即視覺化卷積核經過啟動之後的結果。能夠看到圖像經過卷積之後結果，幫助理解卷積核的作用
- 卷積核的視覺化(Visualizing convnets filters)，說明我們理解卷積核是如何感受圖像的
- 熱度圖視覺化(Visualizing heatmaps of class activation in an image)，通過熱度圖，瞭解圖像分類問題中圖像哪些部分起到了關鍵作用，同時可以定位圖像中物體的位置。

卷积神经网络可视化

- 本文整理自Deep Learning with Python，书本上完整的代码在[这里的5.4节](#)，并附有详细的注释。
- 深度学习一直被人们称为“黑盒子”，即内部算法不可见。但是，卷积神经网络(CNN)却能够被可视化，通过可视化，人们能够了解CNN识别图像的过程。
- 介绍三种可视化方法
 1. 卷积核输出的可视化(Visualizing intermediate convnet outputs (intermediate activations))，即可视化卷积核经过激活之后的结果。能够看到图像经过卷积之后结果，帮助理解卷积核的作用
 2. 卷积核的可视化(Visualizing convnets filters)，帮助我们理解卷积核是如何感受图像的
 3. 热度图可视化(Visualizing heatmaps of class activation in an image)，通过热度图，了解图像分类问题中图像哪些部分起到了关键作用，同时可以定位图像中物体的位置。

卷积核输出的可视化(Visualizing intermediate convnet outputs (intermediate activations))

- 想法很简单：向CNN输入一张图像，获得某些卷积层的输出，可视化该输出
- 代码中，使用到了cats_and_dogs_small_2.h5模型，这是在原书5.2节训练好的模型，当然你完全可以使用keras.applications中的模型，例如VGG16等。
- 可视化结果如下图。

推薦延伸閱讀

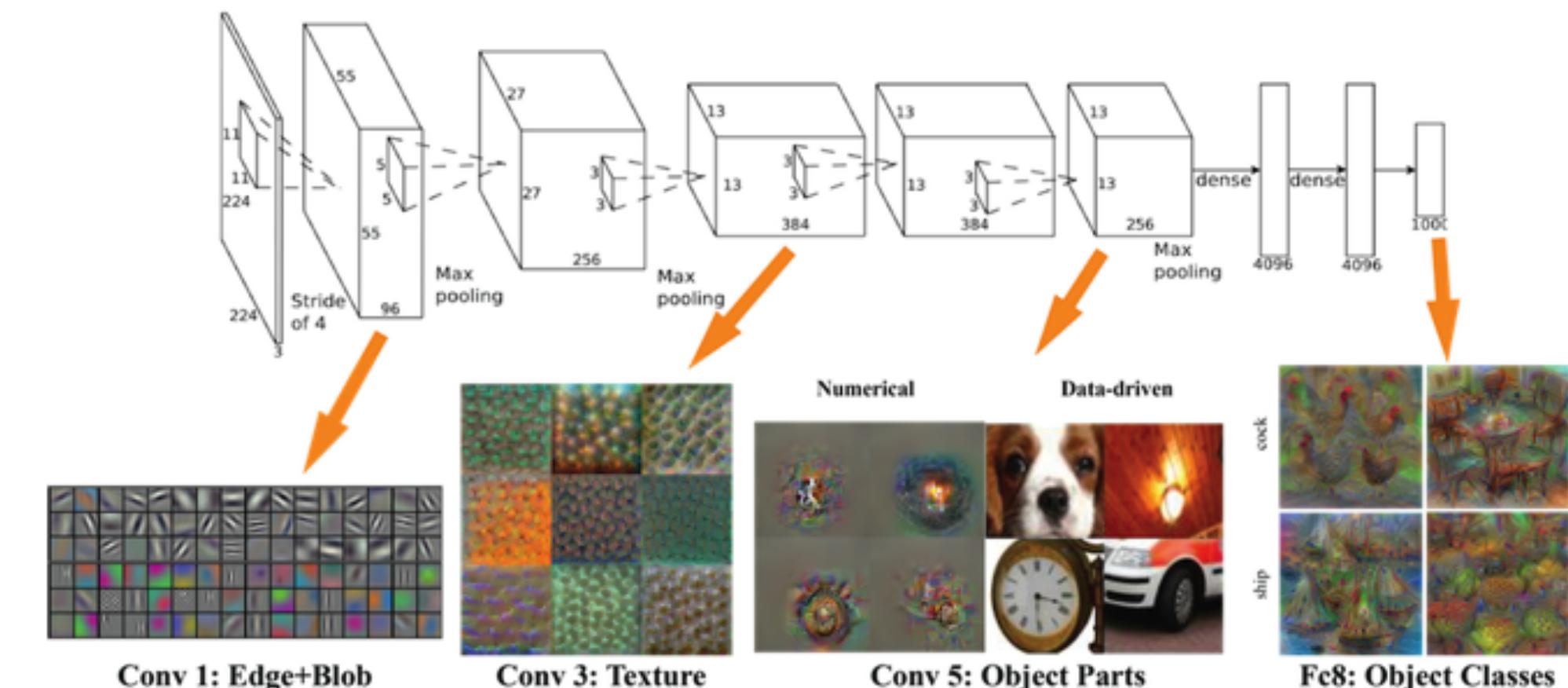
- CNN for Image 過程說明

將一個卷積核作用於一張圖像時，卷積核就像檢測時的觀察窗口

其從圖像的左上角逐步滑動到右下角，其滑動的每個位置對應於一個輸出節點

這個節點只和窗口內的輸入節點（圖像上每一個點都對應於一個輸入節點）相連接

而不同輸出節點和對應輸入節點連接的權值是相同的。



圖片來源：computervisionblog



解題時間

It's Your Turn

請跳出PDF至官網Sample Code & 作業
開始解題

