



Deep learning with keras

C H 8

CONTENT

- 01** 理解卷積神經網路
- 02** 以少量資料訓練 CNN
- 03** 利用預先訓練好的模型



8 - 1

理解捲積神經網路

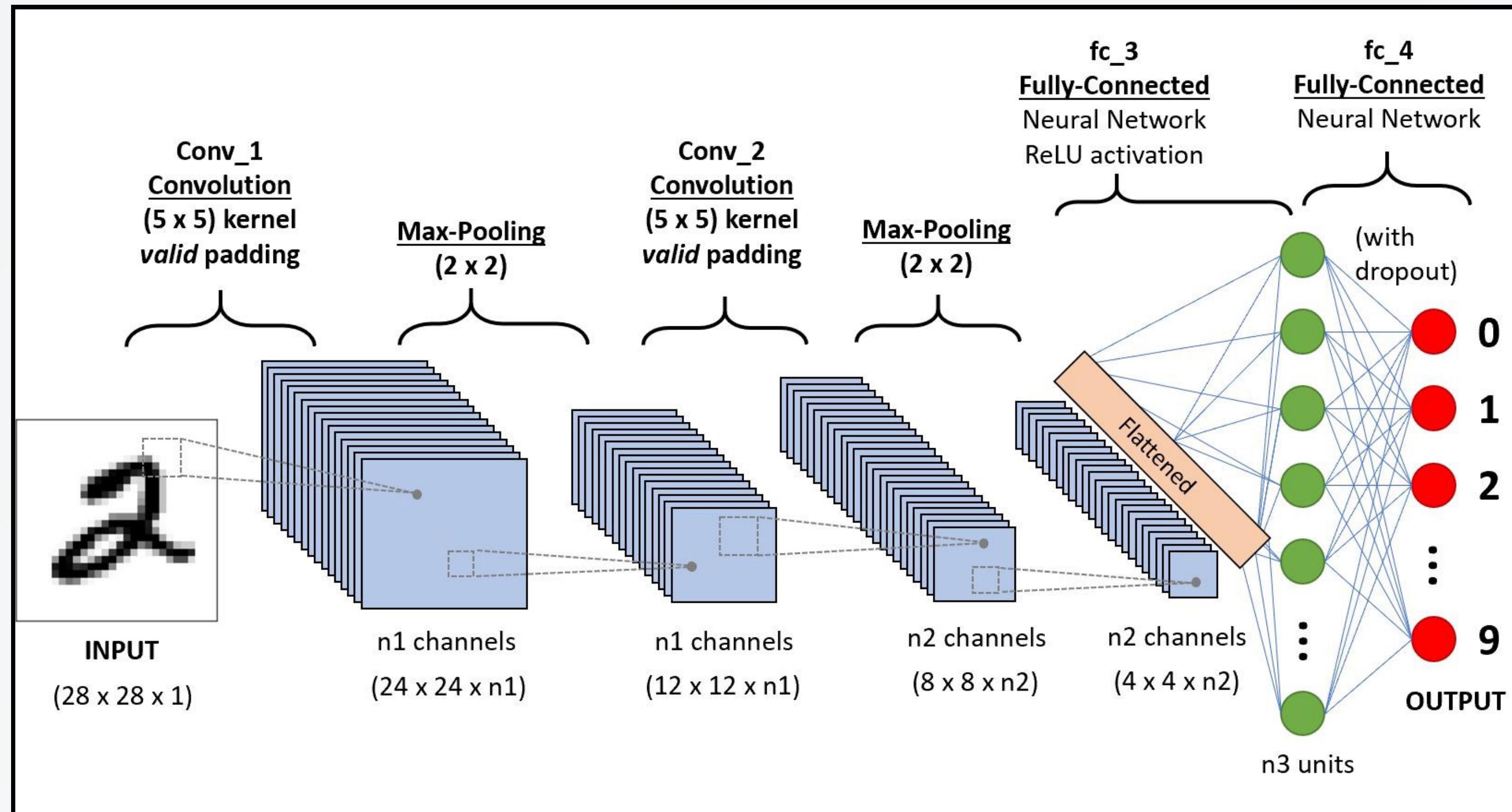
卷積神經網路(CNN)

- 準確度超越密集層連接模型
- 包含卷基層和池化層
- 適用於影像辨識和分類

look code

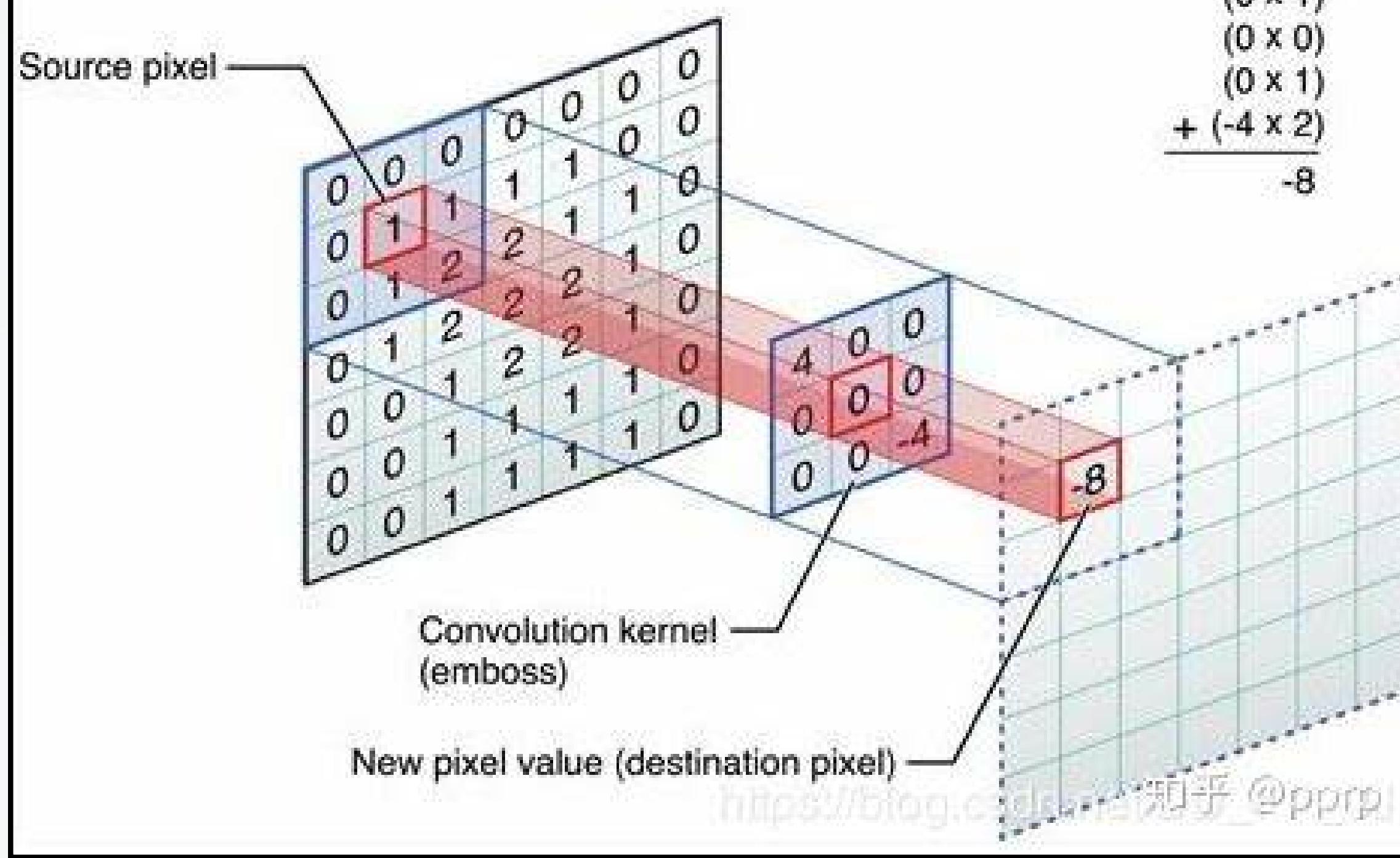
卷積層操作

- 抓取資料的特徵建構 pattern
- 學習到的 pattern 具平移不便性
- 能學習到 pattern 的空間階層架構



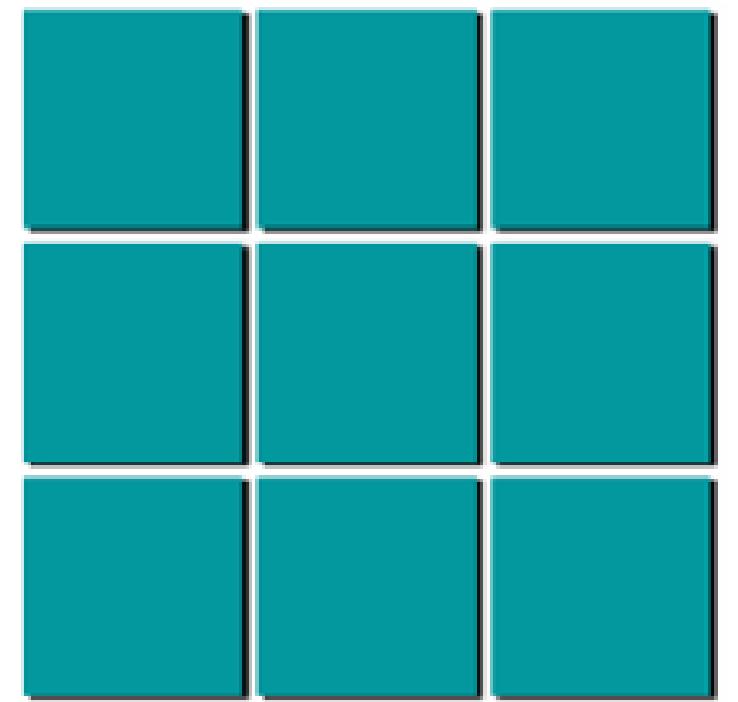
Center element of the kernel is placed over the source pixel. The source pixel is then replaced with a weighted sum of itself and nearby pixels.

$$\begin{array}{r} (4 \times 0) \\ (0 \times 0) \\ (0 \times 0) \\ (0 \times 0) \\ (0 \times 1) \\ (0 \times 1) \\ (0 \times 0) \\ (0 \times 1) \\ + (-4 \times 2) \\ \hline -8 \end{array}$$



邊界效應和填補(padding)

- 特徵圖經過卷積層處理後 size 會變小
- 可以使用填補(padding)讓輸入前後 size 不變
- 填補(padding)：特徵圖輸入前先進行擴張



Input Image

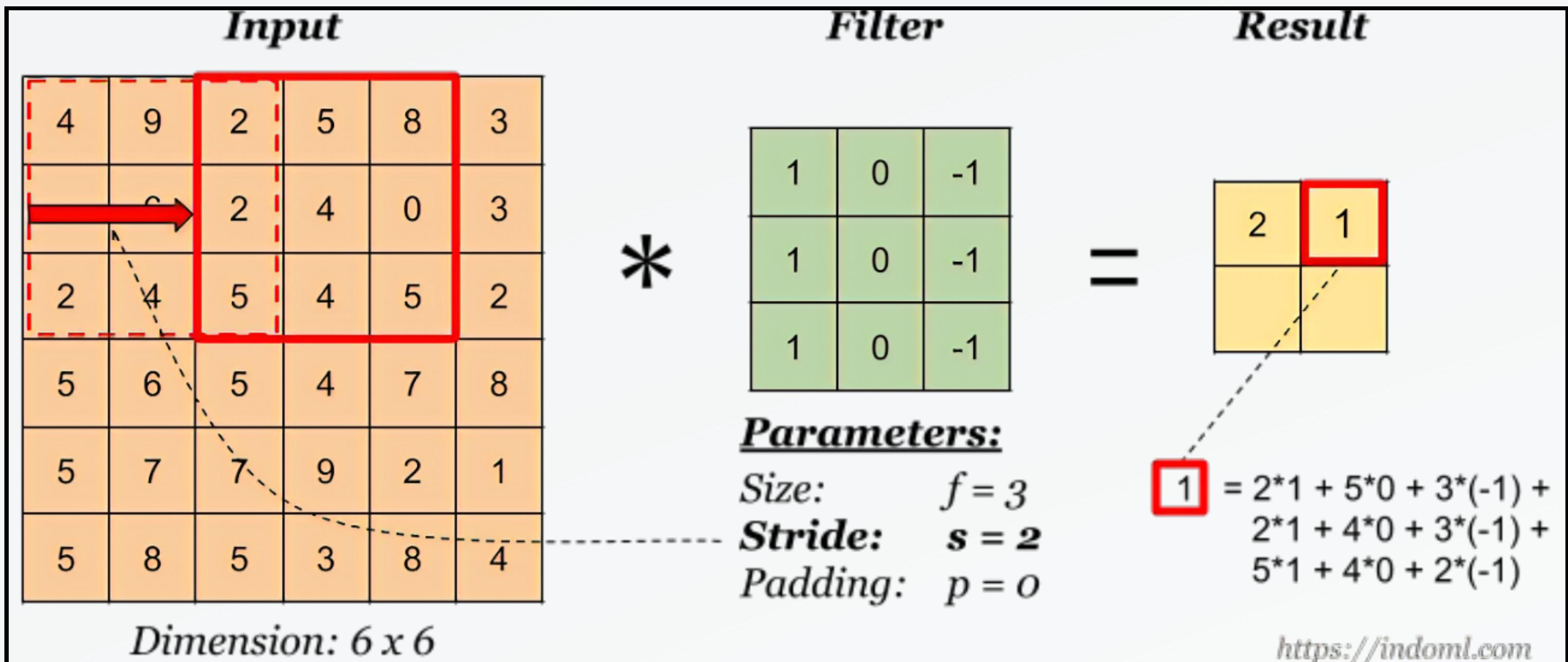
Applying padding
of 1 on 3x3

0	0	0	0	0
0				0
0				0
0				0
0	0	0	0	0

Padded Image

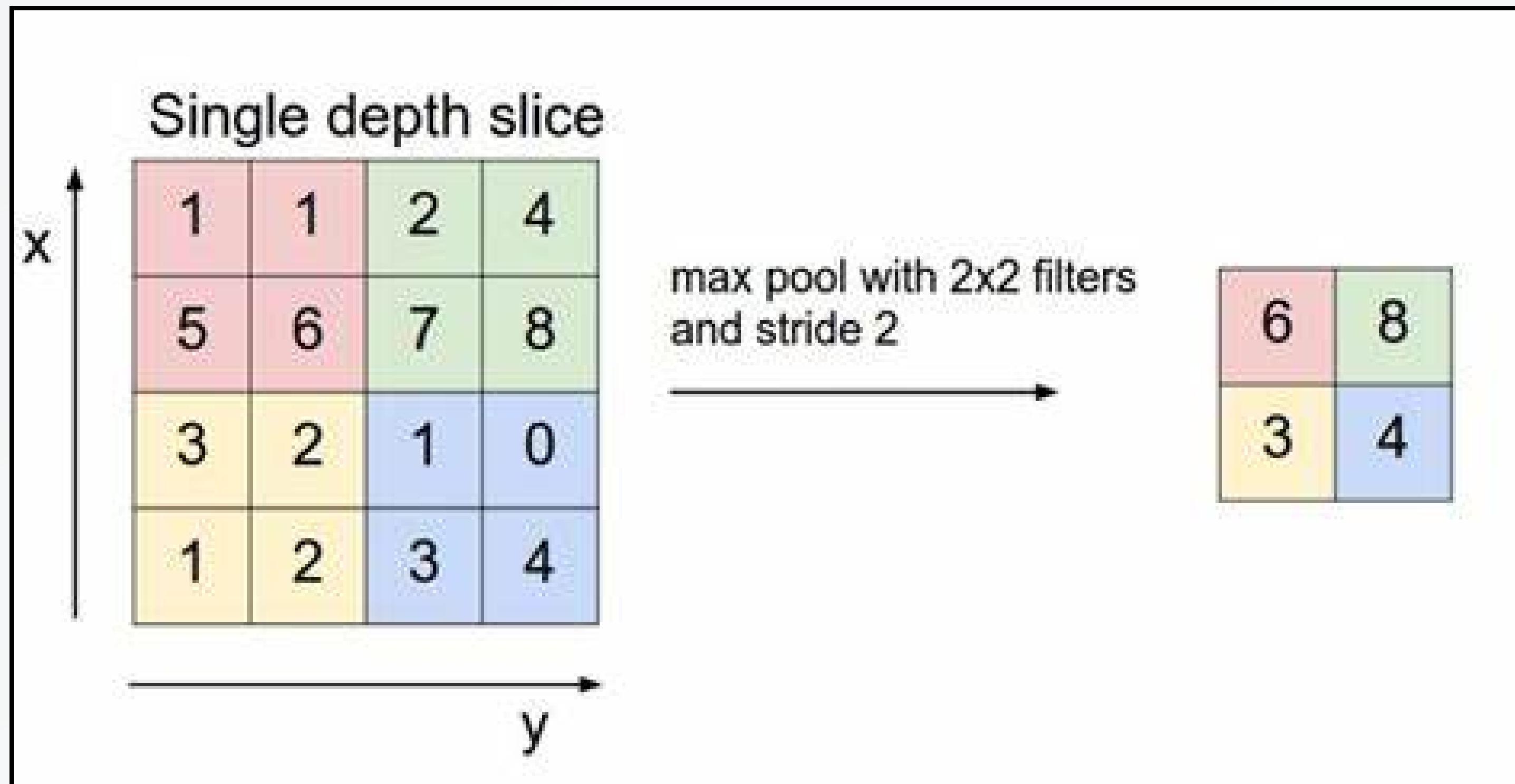
步長(stride)

- 特徵圖經過卷積層處理後 size 會變小
- 可以使用填補(padding)讓輸入前後 size 不變
- 填補(padding)：特徵圖書入前先進行擴張



<https://indoml.com>

最大化池(MaxPooling)



不使用池化層會怎樣？

無法快速提取特徵，訓練 pattern 的進展很小



8 - 2

以少量資料訓練 CNN

減低過度適配的三種技術

- 資料擴增法
- 套用其他預訓練模型的特徵萃取能力
- 套用並微調預訓練模型

深度學習與少量資料間的相關性

- 深度學習只有在大量訓練樣本時才有效？
- 卷積神經網路具有學習力強的性能
- 深度學習模型可高度再利用

建立 CNN

- 下載 kaggle 資料集 (dogs-vs-cats)
- 建立 model
- 配置 model

look code

資料預處理

- 讀取影像檔
- 將 JPEG 解碼為 RGB 像素網格
- 將 RGB 像素轉換成浮點數張量
- 調整大小為 180*180
- 將樣本打包成批次 (batch_size=32)

look code

Dataset (Tensorflow的物件)

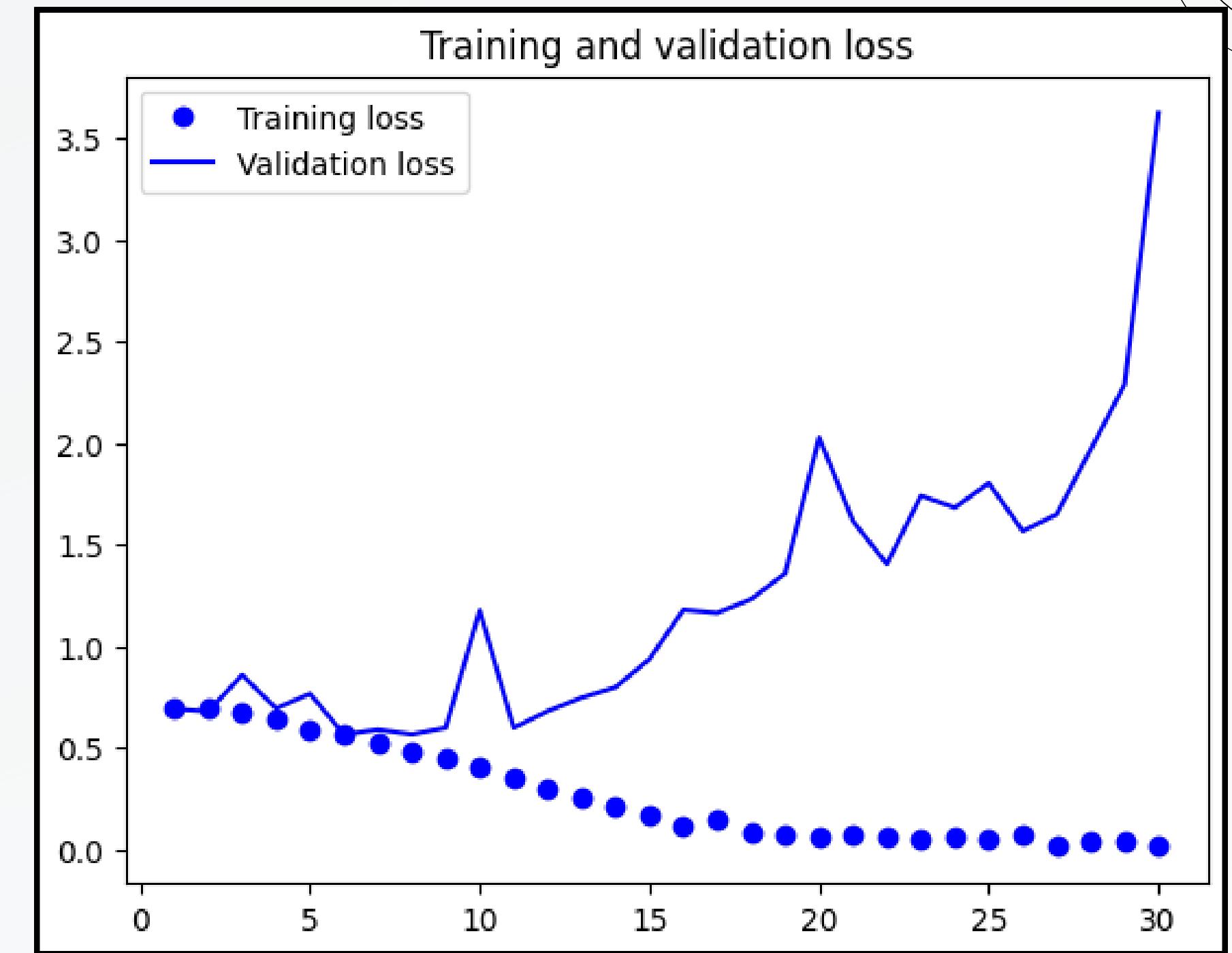
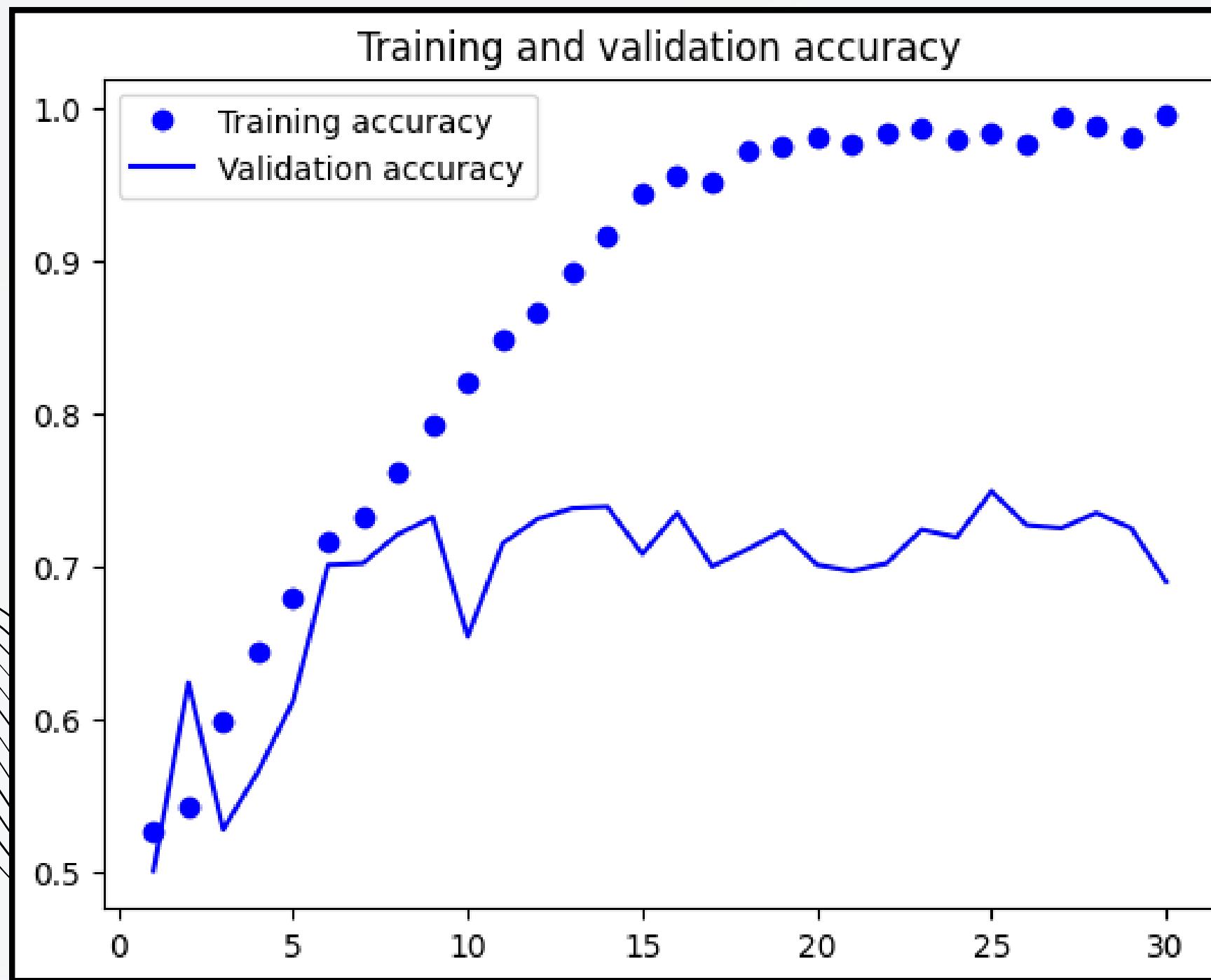
- Dataset 是一種 iterator(迭代器)
- 傳回包含輸入與標籤的批次資料
- 考直接傳入 Keras 模型的 fit() 使用
- 提供非同步資料擷取

look code

Dataset method

- shuffle(buffer_size) : 打亂元素順序
- prefetch(buffer_size) : 預取 buffer_size 個樣本到 GPU, 以更好利用硬體
- map(callable) : 對資料集中的元素進行轉換
ex : reshape

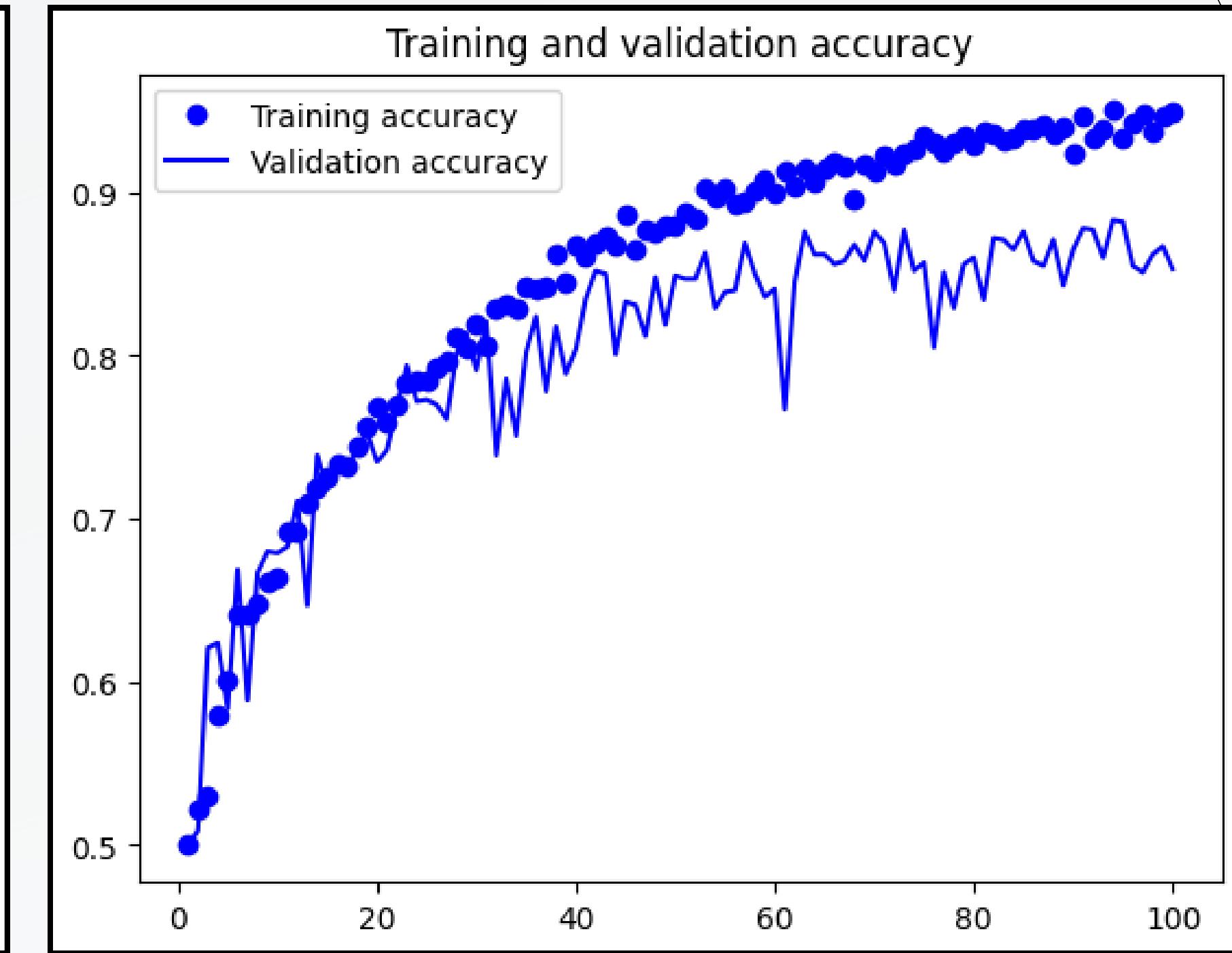
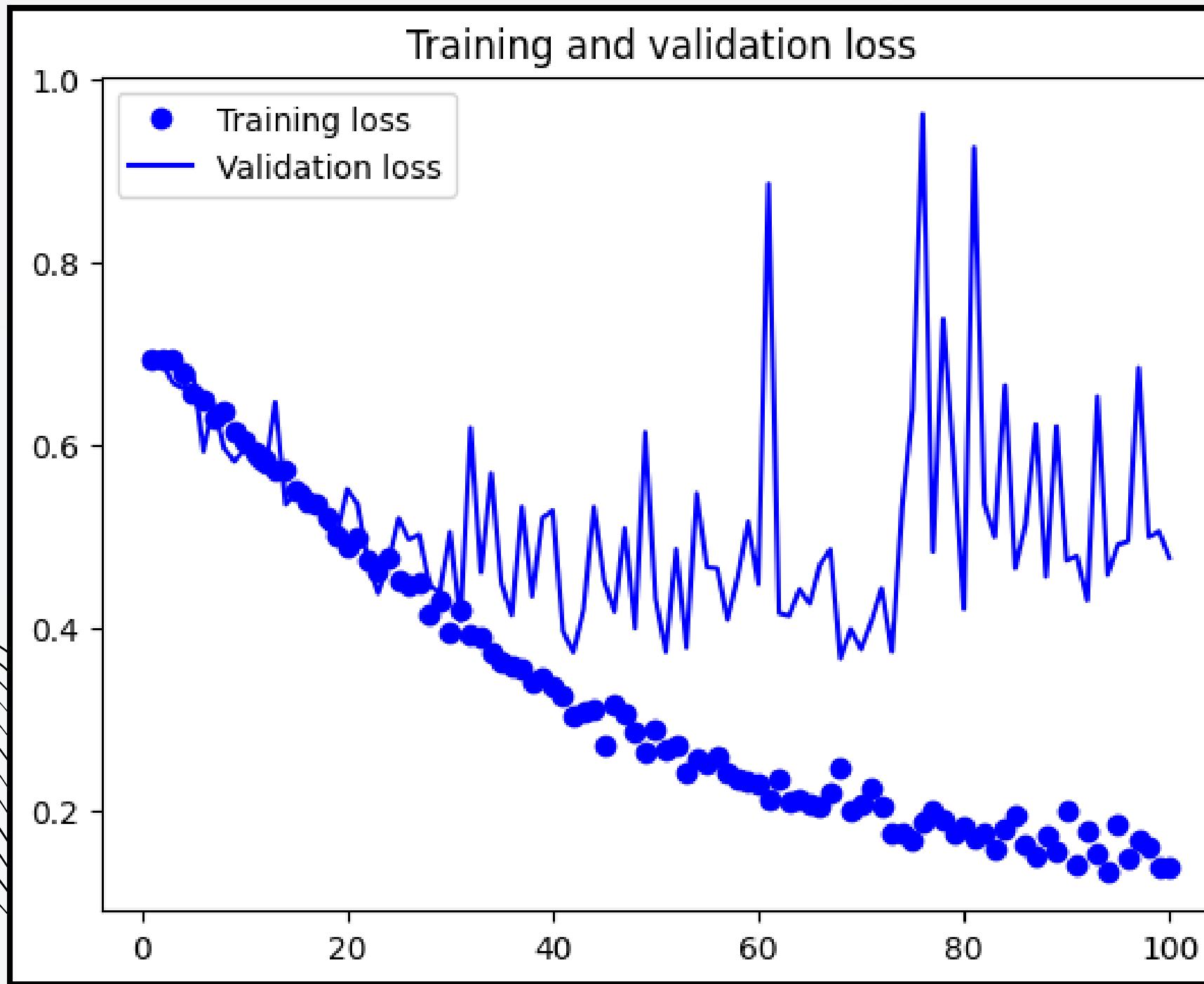
損失和準確度曲線



使用資料擴增法

- 用現有樣本生成更多訓練資料
- 重新混和的訓練資料仍無法完全擺脫過度配適
- 加入 Dropout 層減緩過度適的發生

使用資料擴增法



在資料很少的情況下，很難進一步提升準確度，因此，可以使用預訓練模型解決這個問題

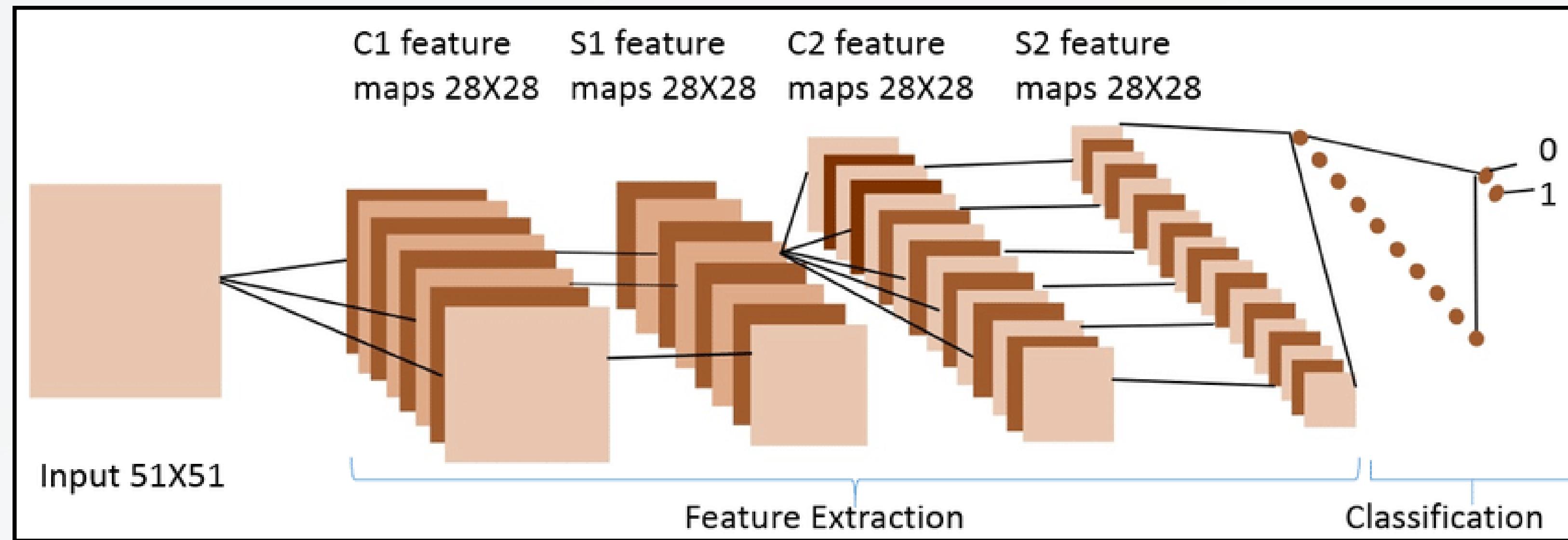
8 - 3

利用預先訓練好的模型

使用預訓練模型

- deep learning 可將學習到特徵移植到不同問題上
- 使用大量資料訓練後所保存下來的優秀模型
- 使用預訓練模型的兩種方法：**特徵萃取、微調**

特徵萃取



卷積層不變

更換分類器

look code

為何只重複使用卷積層，分類器 是否也能重複使用？

- 卷積神經網路的特徵圖代表的是通用概念圖
- 分類器所學習到的表示法只適用於目前模型所要分類的類別

補充

- 卷積層包含物件出現的位置訊息
- CNN 較前面的層會萃取局部、通用性高的特徵圖 (邊緣、顏色、紋理)
- CNN 較深層會萃取更抽象、細節的特徵 (貓耳朵、眼睛)
- 如果新資料與原始模型差別較大，最好只使用前幾層進行特徵萃取

建立 VGG16 模型

- 從 keras.applications 模組中載入
- 傳遞三個參數給 VGG16 建構子
- 特徵萃取 (有2個方法)
- 接上密集連接分類器

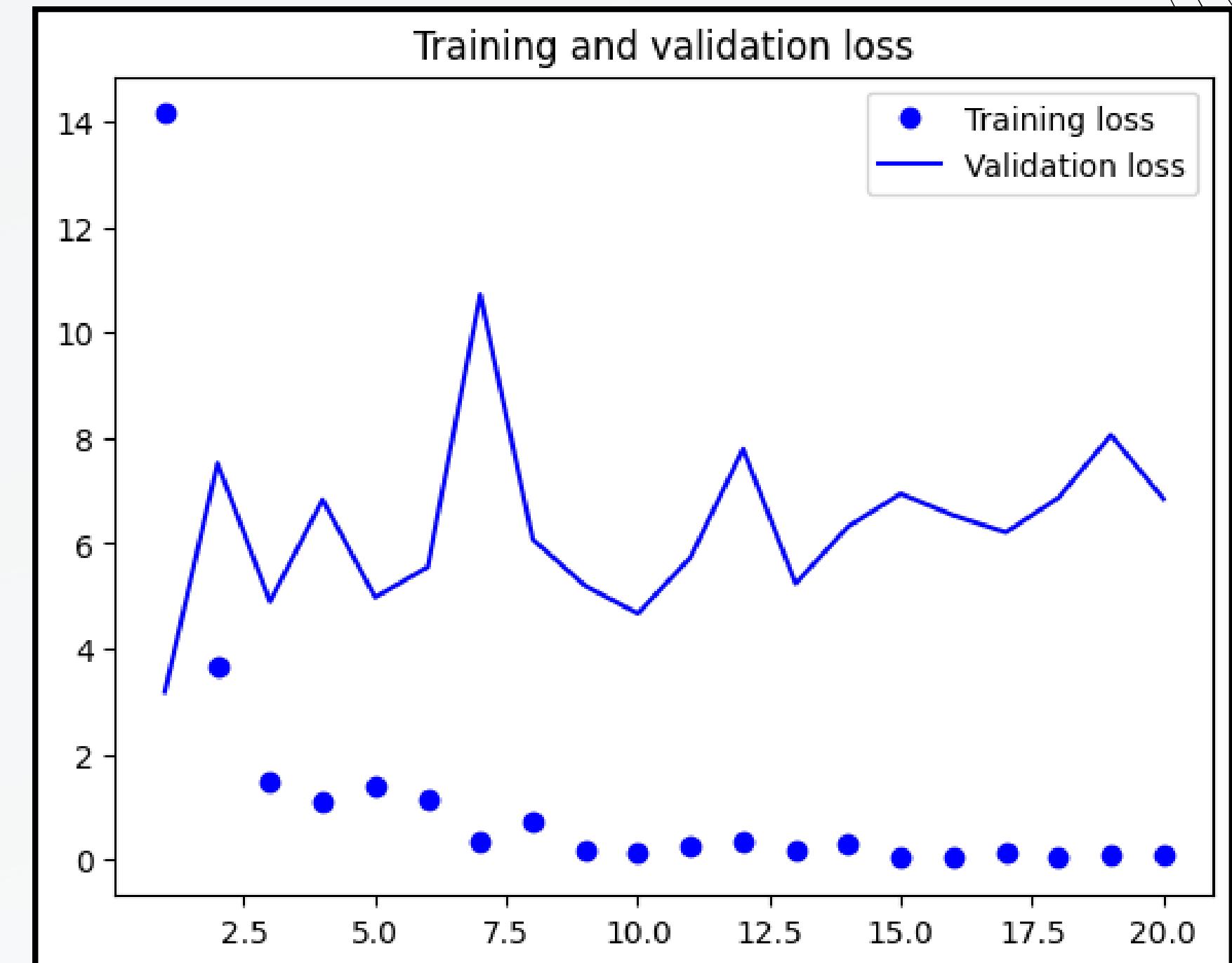
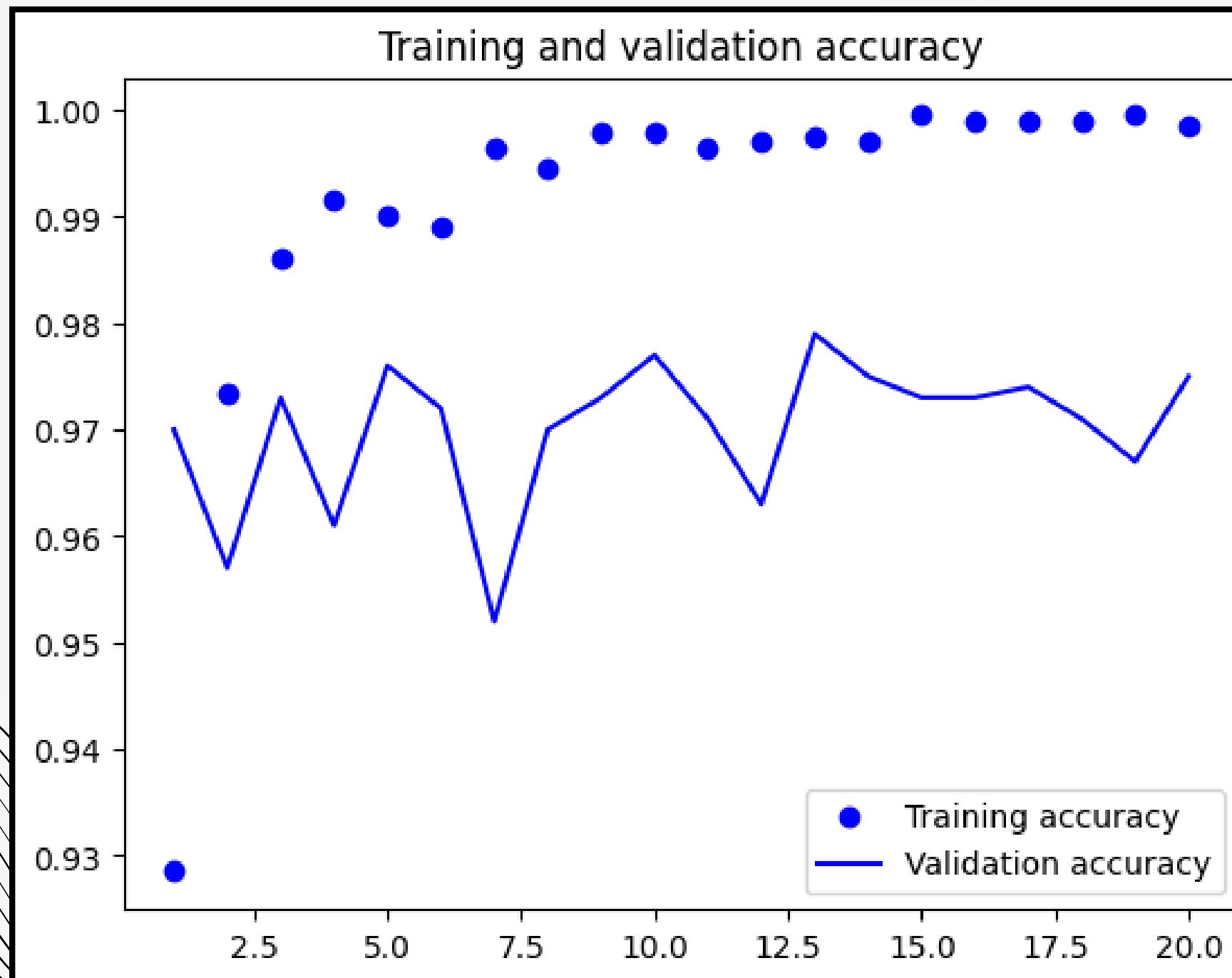
look code

特徵萃取方法 1

- 在資料集上執行 Convolutional base, 結果儲存至 NumPy 陣列, 輸入密集層分類器
- 缺點：不可使用資料擴增法
- 優點：執行速度快, low overhead

look code

特徵萃取方法 1



look code

特徵萃取方法 2

- 在最後端增加 Dense 層擴展模型, 使用輸入資料, 從頭到尾執行整個處理過程
- 缺點：可使用資料擴增法
- 優點：執行速度慢, high overhead

model 須包含：

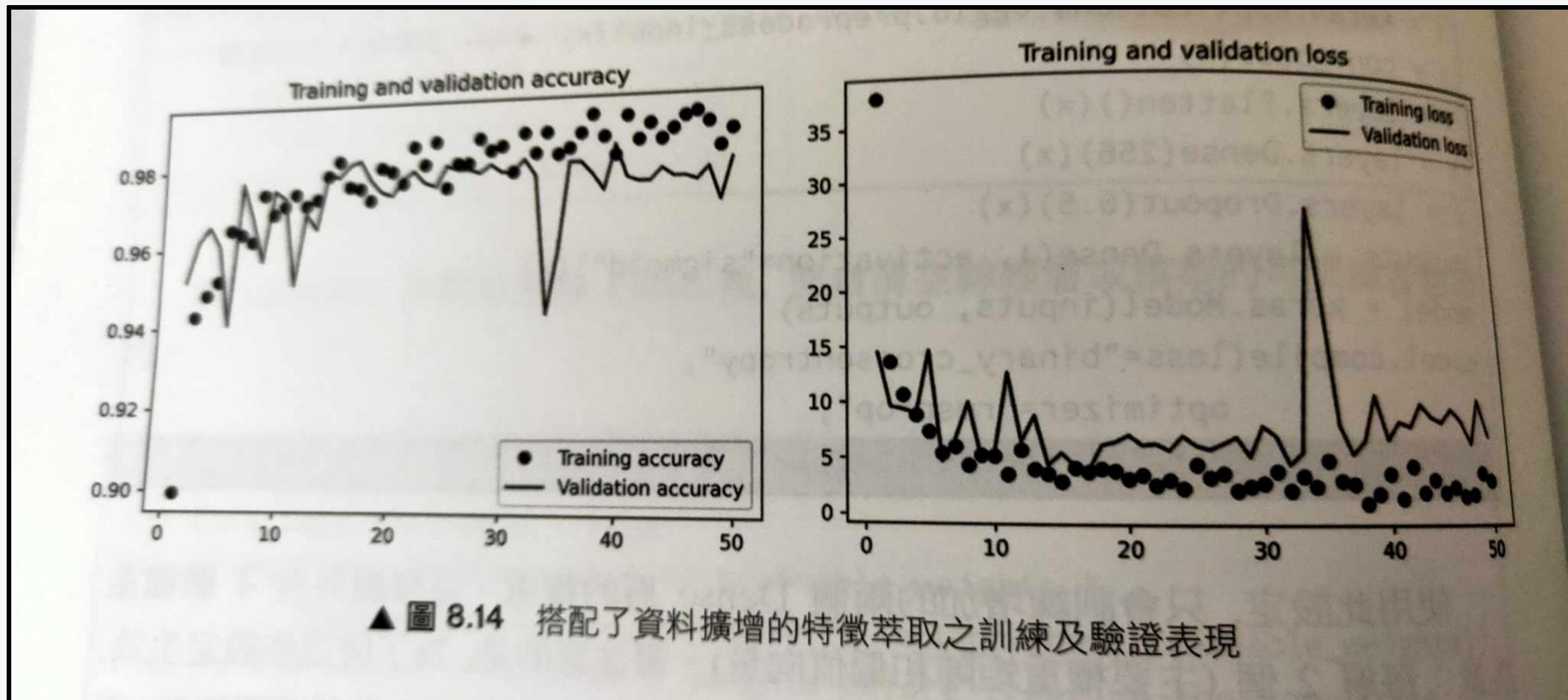
- 數個資料擴增的神經層
- 凍結 convolutional base
- 一個密集分類器

特徵萃取方法 2

- 數個資料擴增的神經層
- 凍結 convolutional base
- 一個密集分類器

look code

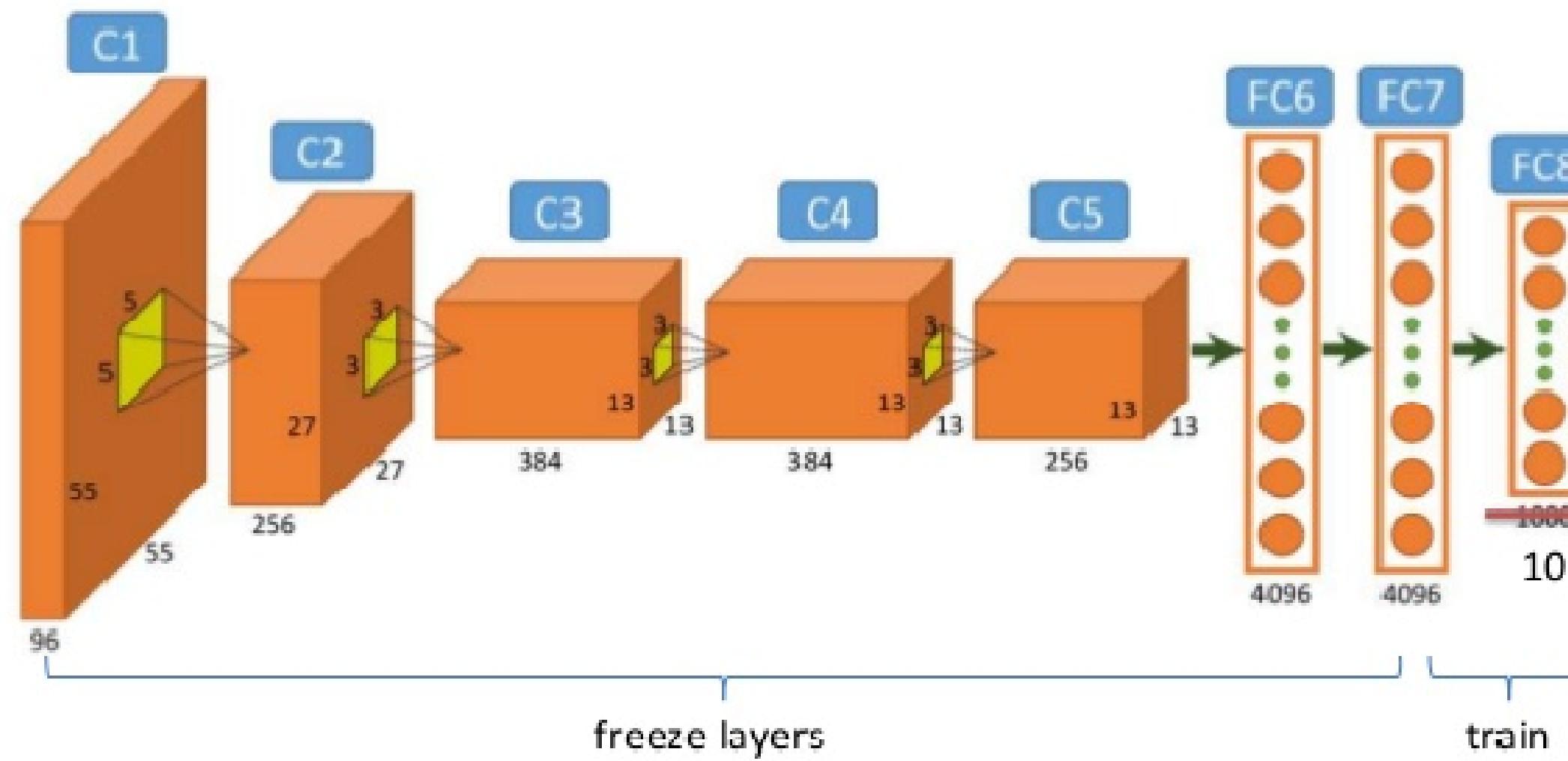
特徵萃取方法 2



look code

微調

Fine-tuning Pretrained Network



look code

微調神經網路 step

- 在訓練過的 convolutional base 上新增分類器
- 凍結 convolutional base
- 訓練分類器
- 解凍卷積基底中某幾層 (批次正規化層不能解凍)
- 共同訓練解凍的這幾層和分類器

為什麼不微調整個卷積基底

- 卷積基底的前幾層具有通用性
- 卷積基底深層表現的是特定的特徵
- 微調卷積基底深層效果最好
- 訓練參數越多，越可能過度適配
- VGG16 卷積基底有1500萬個參數，嘗試在少量資料集上訓練，很可能凸顯過度適配

ENDING

