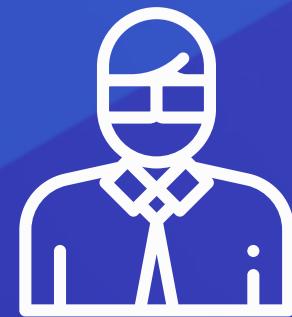


Day100

深度學習應用卷積神經網路

訓練卷積神經網路的細節與技巧

遷移學習 transfer learning



楊証琨

出題教練



知識地圖 卷積網路訓練技巧

遷移學習

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

卷積類神經網路訓練技巧

Training Skill of CNN

處理大量數據

處理小量數據

遷移學習 (Transfer Learning)

本日知識點目標

- 了解遷移學習 (Transfer learning) 的意義
- 了解在 Keras 中如何使用遷移學習來訓練模型

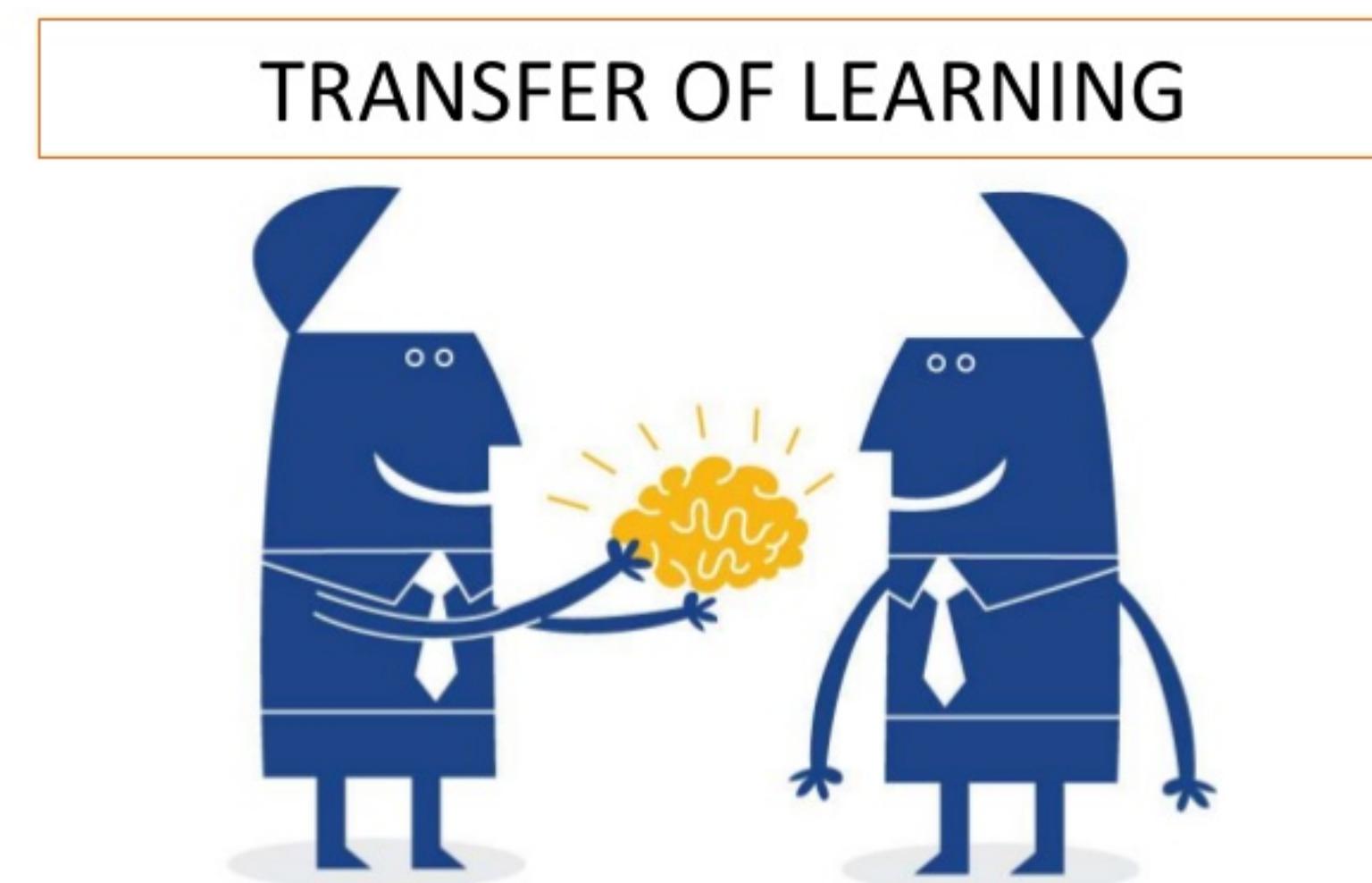
遷移學習，Transfer Learning



- 資料量不足時，遷移學習也是很常見的方法
- 神經網路訓練前的初始參數是隨機產生的，不具備任何意義
- 透過在其他龐大資料集上訓練好的模型參數，我們使用這個參數當成起始點，改用在自己的資料集上訓練！

為何可以用遷移學習？

- 前面 CNN 的課程有提到，CNN 淺層的過濾器 (filter) 是用來偵測線條與顏色等簡單的元素。因此不管圖像是什麼類型，基本的組成應該要是一樣的
- 大型資料集 (如 ImageNet) 訓練好的參數具有完整的顏色、線條 filters，從此參數開始，我們訓練在自己的資料集中，逐步把 filters 修正為適合自己資料集的結果。



The application of skills, knowledge, and/or attitudes that were learned in one situation to another **learning** situation (Perkins, 1992)

圖片來源：[Transfer of learning](#)

參考大神們的網路架構

- 同學們對於要疊幾層 CNN，filters 數量要選擇多少？Stride, Pooling 等參數要設定多少？這些應該都很疑惑。
- 許多學者們研究了許多架構與多次調整超參數，並在大型資料集如 ImageNet 上進行測試得到準確性高並容易泛化的網路架構，我們也可以從這樣的架構開始！



- 以下的程式碼提供給有 GPU 且具有較大影像尺寸資料集的同學參考，若沒有 GPU 的同學可以直接看本日的 jupyter notebook 程式碼學習即可
- Cifar-10 並不適合直接使用 transfer learning 原因是多數模型都是在 ImageNet 上預訓練好的，而 ImageNet 影像大小為 $(224,224,3)$ ，圖像差異極大，硬套用的結果反而不好

Transfer learning in Keras: ResNet-50 (1/3)

```
from keras.applications.resnet50 import ResNet50  
  
resnet_model = ResNet50(input_shape=(224,224,3),  
weights="imagenet", pooling="avg",  
include_top=False)
```

- 我們使用了 ResNet50 網路結構，其中可以看到 weights="imagenet"，代表我們使用從 imagenet 訓練好的參數來初始化，並指定輸入的影像大小為 (224,224,3)
- pooling=avg 代表最後一層使用 Global Average pooling，把 featuremaps 變成一維的向量
- include_top=False 代表將原本的 Dense layer 拔掉，因為原本這個網路是用來做 1000 個分類的模型，我們必須替換成自己的 Dense layer 來符合我們自己資料集的類別數量

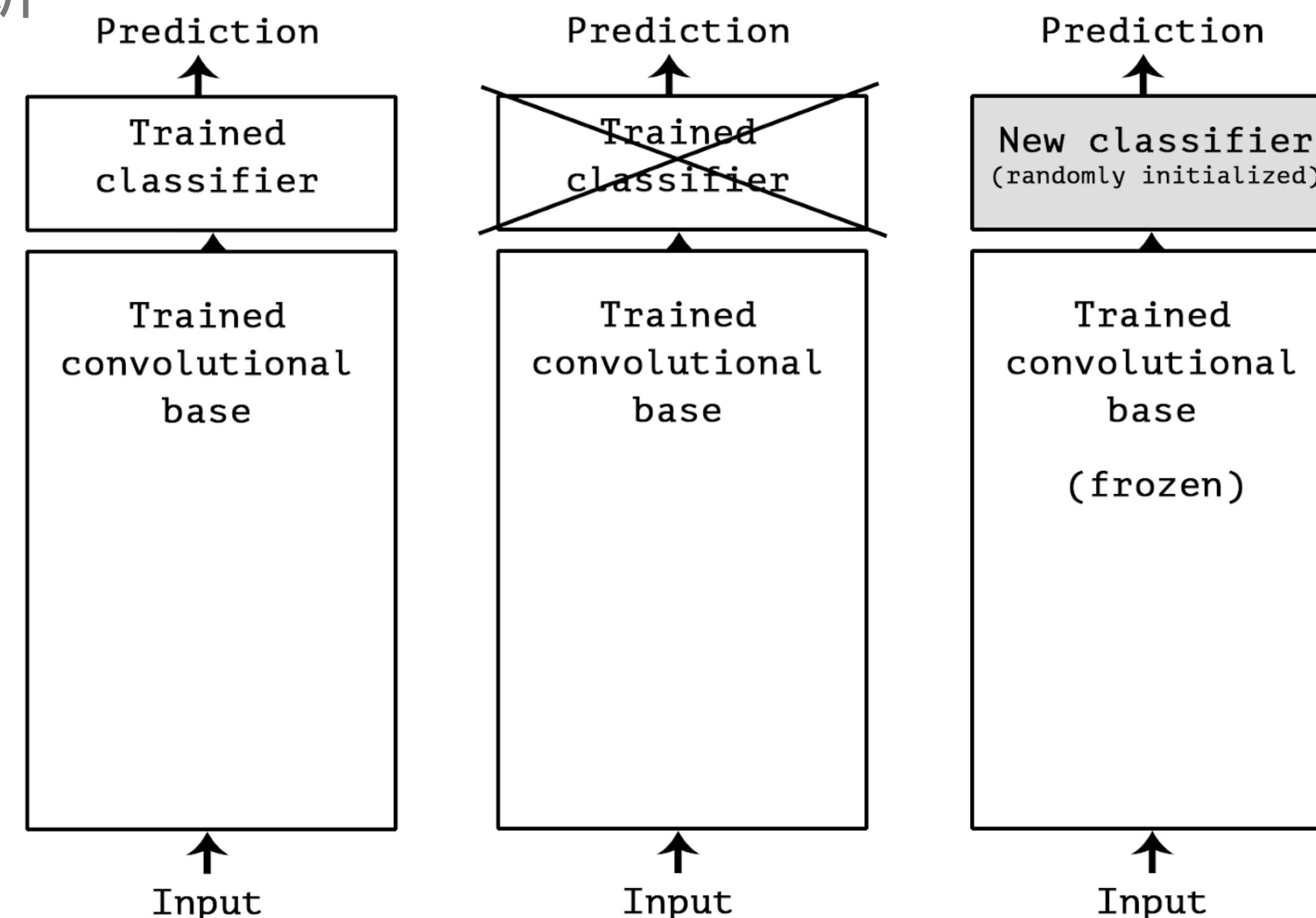
Transfer learning in Keras: ResNet-50 (2/3)

```
last_featuremaps = resnet_model.output  
flatten_featuremap = Flatten()(last_featuremaps)  
output = Dense(num_classes)(flatten_featuremap)  
  
New_resnet_model =  
Model(inputs=resnet_model.input, outputs=output)
```

- 上一頁的模型我們已經設定成沒有 Dense layers，且最後一層做 GAP，使用 `resnet_model.output` 我們就可以取出最後一層的 featuremaps
- 將其使用 `Flatten` 攤平後，再接上我們的 Dense layer，神經元數量與資料集的類別數量一致，重建立模型，就可以得到一個新的 ResNet-50 模型，且參數是根據 ImageNet 大型資料集預訓練好的

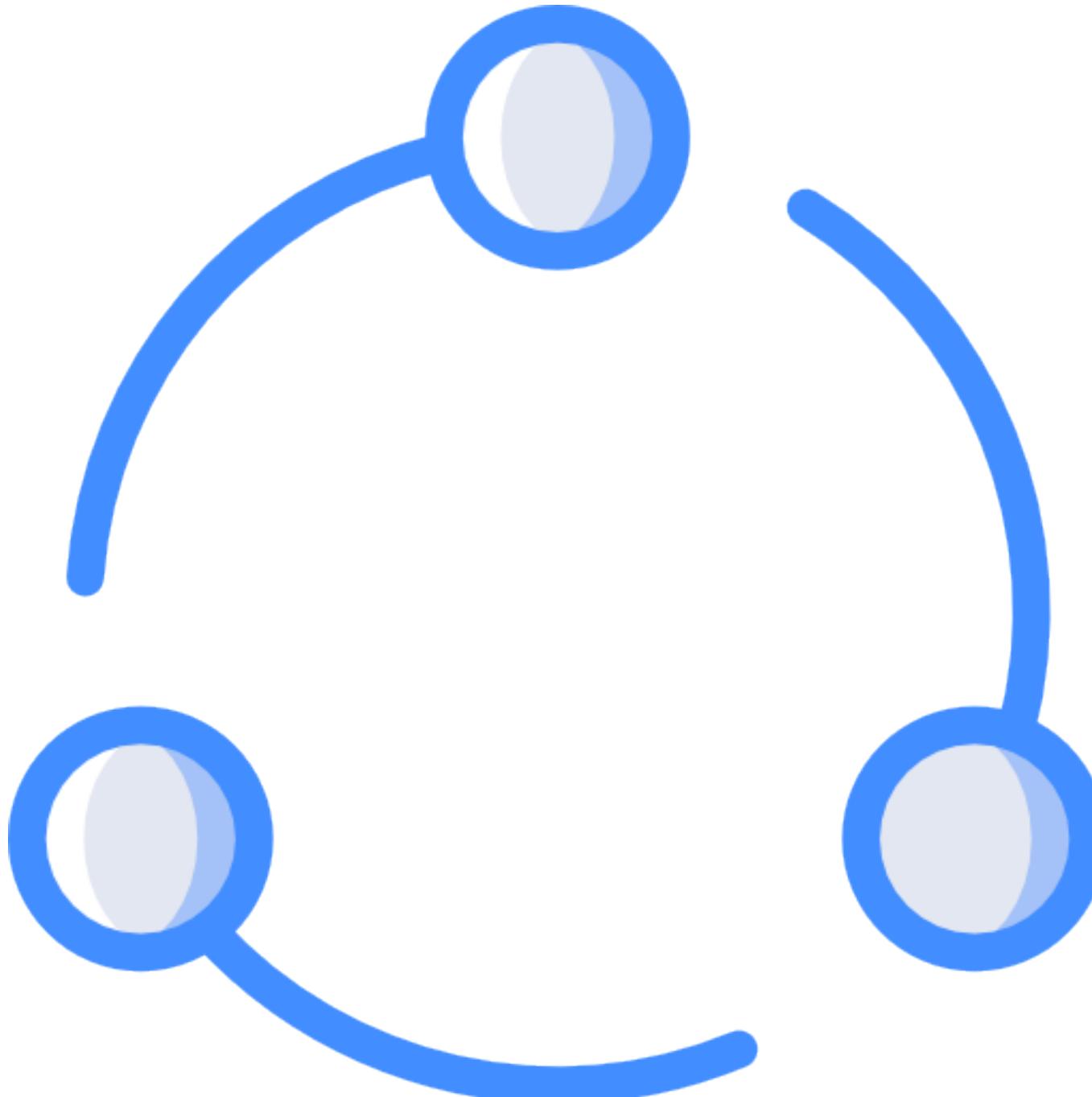
Transfer learning in Keras: ResNet-50 (3/3)

整體流程如下圖，我們保留 Trained convolutional base，並新建 New classifier (Dense 的部分)，最後 convolutional base **是否要 frozen (不訓練)** 則是要看資料集與預訓練的 ImageNet 是否相似，如果差異很大則建議訓練時不要 frozen，讓 CNN 的參數可以繼續更新



圖片來源：[Github](#)

重要知識點複習



- 遷移學習是透過預先在大型資料集訓練好的權重，再根據自己的資料集進行微調 (finetune) 的一種學習方法
- Keras 中的模型，只要指定權重 `weights="imagenet"` 即可使用遷移學習



延伸 閱讀

- 簡單使用 Keras 完成 transfer learning (中文)
- Keras 作者教你用 pre-trained CNN 模型 (英文)



解題時間

It's Your Turn

請跳出PDF至官網Sample Code & 作業
開始解題

