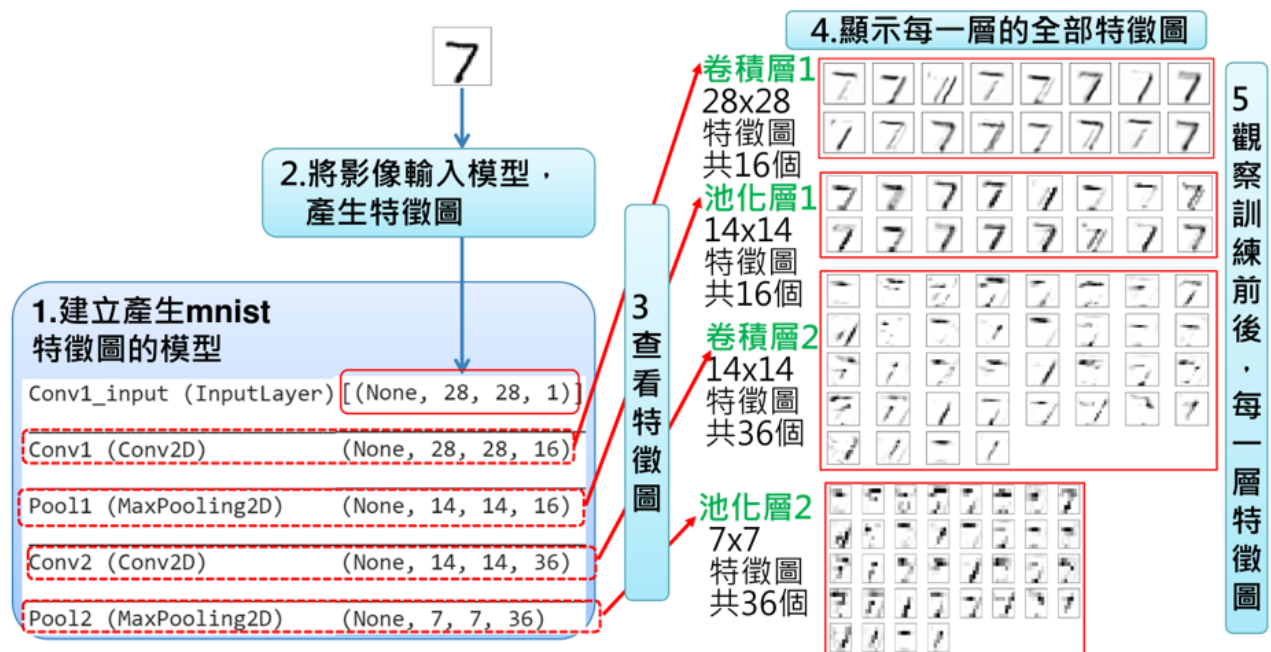


視覺化顯示卷積神經網路(CNN)特徵圖，理解 CNN 如何提取特徵

林大貴 6月15, 2022

人們常說深度學習模型是「黑箱模型」，我們只知道深度學習模型能進行預測，卻不知道模型如何預測。例如多層感知器模型，由多個神經層組成，每個神經層由很多神經元組成，確實難以知道內部如何運作。不過對於卷積神經網路，我們卻可以呈現每一層特徵圖的輸出，讓我們知道模型如何透過濾鏡提取特徵。我們將介紹，如何建立輸出 CNN 特徵圖(Feature Map)的模型，呈現每一層特徵圖的結果。

目標：產生 mnist CNN 模型的特徵圖

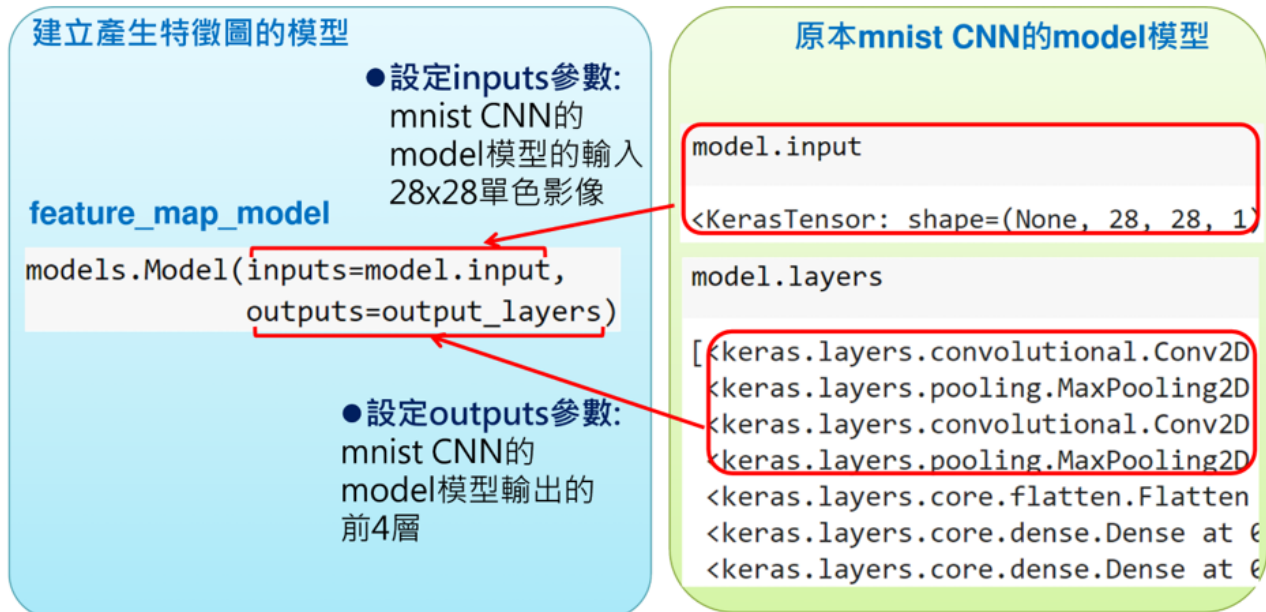


如上圖，產生 mnist 模型的特徵圖的步驟如下：

- 1.建立產生 mnist 特徵圖的模型：我們將介紹如何建立，產生 mnist 特徵圖的模型的程式碼。
- 2.將影像輸入模型，產生特徵圖：影像進行預處理後，輸入模型進行預測，就能產生特徵圖。因為要觀察訓練前後產生特徵圖的差異。所以模型訓練前先產生特徵圖，訓練後再次產生特徵圖。
- 3.查看特徵圖：介紹查看與顯示特徵圖的程式碼。
- 4.顯示每一層的全部特徵圖：定義 `display_feature_maps()` 函數，能顯示每一層的全部特徵圖。
- 5.觀察訓練前後，每一層特徵圖：讓我們可以理解卷積神經網路，如何提取特徵。

建立產生 mnist 特徵圖的模型

關於建立產生 mnist 特徵圖的模型的詳細說明請參考本書，以下僅介紹其概念。



如上圖，產生 mnist 特徵圖的模型，是由原本 mnist CNN 的 model 模型，擷取部分元件所建立的。我們將使用 `models.Model` 模組，建立產生特徵圖的模型 `feature_map_model`，須設定以下參數：

- **inputs 參數:** 設定為 mnist CNN 的 model 模型的輸入 `model.input` (28x28 單色影像)
- **outputs 參數:** 設定為 mnist CNN 的 model 模型輸出的前 4 層

建立產生特徵圖的模型程式碼

建立 `feature_map_model` 模型：主要是使用 `models.Model` 模組，設定 `inputs` 參數與 `outputs` 參數，說明如下：

```
[ ] output_layers=[layer.output
    for layer in model.layers[:DISPLAY_LAYER]]
    output_layers -> cnn模型的前4層卷積與池化層特徵圖
[<KerasTensor: shape=(None, 28, 28, 16) dtype=float32
 <KerasTensor: shape=(None, 14, 14, 16) dtype=float32
 <KerasTensor: shape=(None, 14, 14, 36) dtype=float32
 <KerasTensor: shape=(None, 7, 7, 36) dtype=float32 (c
[ ] model.input -> mnist cnn模型的輸入28x28單色影像
    <KerasTensor: shape=(None, 28, 28, 1) dtype=float32 (
[ ] from tensorflow.keras import models
    feature_map_model=models.Model(inputs=model.input,
    outputs=output_layers)
```

inputs 參數

outputs 參數

2.回傳：已建立產生特徵圖的模型

1.建立模型

將影像輸入模型，產生特徵圖

建立 `feature_map_model` 模型後，依照下列步驟產生特徵圖：

- 1.讀取影像並進行預處理：詳細請參考本書說明
- 2.將預處理後的影像，輸入模型進行預測：以下會介紹
- 3.查看預測結果：詳細請參考本書說明

將預處理後的影像，輸入模型進行預測

我們只需要將影像預處理後的 4 維張量，輸入模型進行預測，就可以產生每一層特徵圖。

2.回傳：訓練前每一層的特徵圖，
儲存於feature_maps變數

1. 訓練前：使用
feature_map_model模型進行預測

●輸入影像：
的4維張量

```
[ ] feature_maps = feature_map_model.predict(input_4d_img)
```

```
[ ] train_history=model.fit(  
    x=x_train_normalize,y=y_train_onehot,  
    validation_split=0.2,batch_size=200,epochs=10,verbose=2)
```

```
Epoch 1/10  
200/200 - 4s - loss: 1.5507 - acc: 0.4471 - val_loss: 1.2435 - val_acc: 0.5667  
... 執行10 epochs訓練，過程省略  
Epoch 10/10  
200/200 - 2s - loss: 0.3024 - acc: 0.9024 - val_loss: 0.9361 - val_acc: 0.7219
```

3
執行
訓練

```
[ ] feature_maps_trained = feature_map_model.predict(input_4d_img)
```

5.回傳：訓練後每一層的特徵圖，
儲存於feature_maps_trained
變數

4.訓練後：再次使用
feature_map_model模型
進行預測

●輸入影像：
的4維張量

以上產生的訓練前後特徵圖變數：

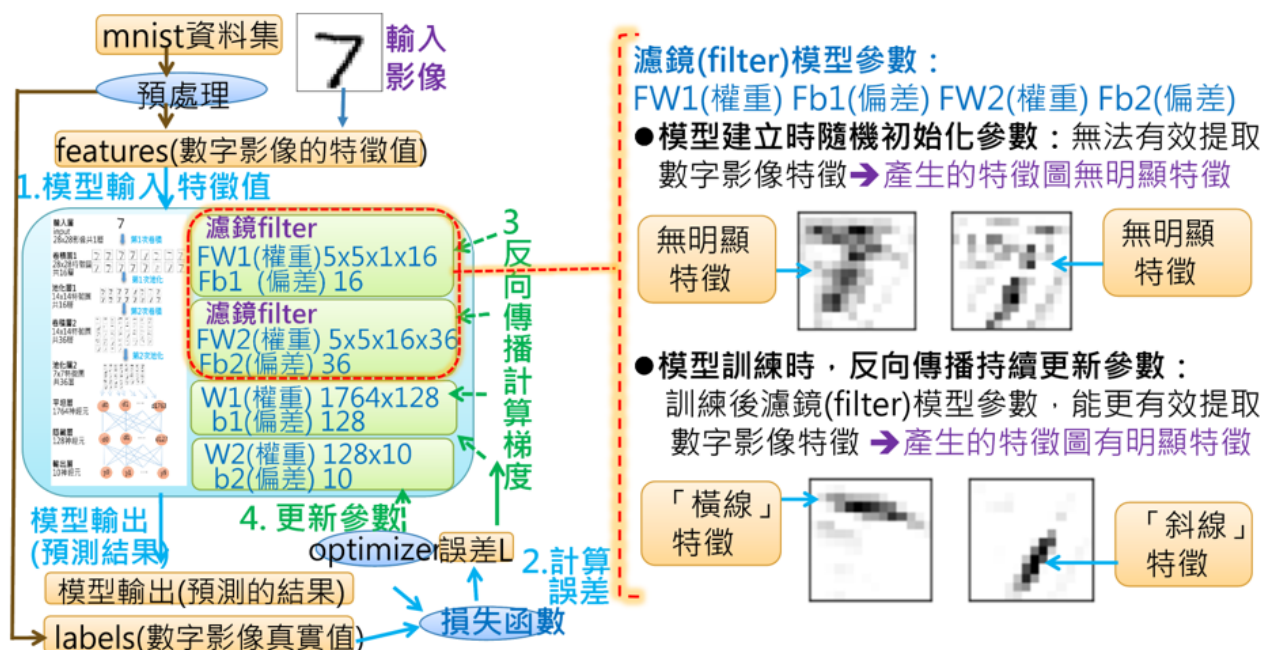
- 訓練前：特徵圖變數 feature_maps
- 訓練後：特徵圖變數 feature_maps_trained

顯示模型訓練前後特徵圖

我們將定義 `display_feature_maps()` 函數(詳細請參考本書)，顯示卷積層或池化層每一個特徵圖。

模型訓練前後，產生特徵圖差異說明

後續我們將顯示使用反向傳播演算法訓練模型前後，產生特徵圖差異，說明如下：



卷積層 1：提取淺層特徵

以下執行函數，輸入參數 `idx=0`，顯示卷積層 1 的訓練前後特徵圖。



如上圖，卷積層 1：提取淺層特徵(輪廓、邊緣、形狀)，你可以看到訓練前後，提取特徵的差異。

Step2.卷積層 2：提取中層特徵

以下執行函數，輸入參數 `idx=2`，顯示卷積層 2 的訓練前後特徵圖。由於卷積層 2 所產生的特徵圖數量 36 個比較多，所以我們參數 `fig_width` 寬度為 16，參數 `n_cols` 列數為 12。

```
display_feature_maps(feature_maps, layer_names, idx=2, fig_width=16, n_cols=12)
```

layer 2 conv2d_3 (1, 14, 14, 36)

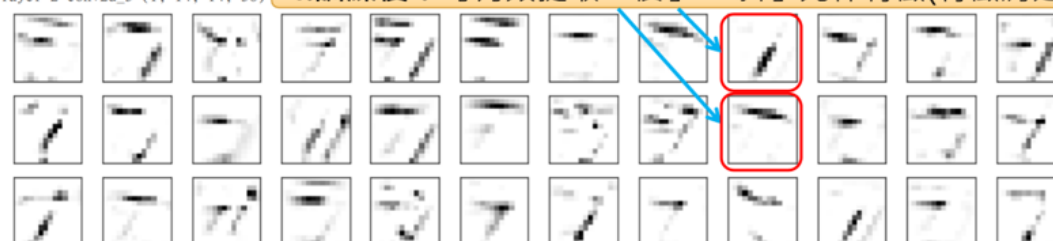


2.訓練前：無法有效提取「橫」「斜」元件特徵(特徵模糊)

1.顯示
訓練前
卷積層2
特徵圖

```
display_feature_maps(feature_maps_trained, layer_names, idx=2, fig_width=16, n_cols=12)
```

layer 2 conv2d_3 (1, 14, 14, 36)



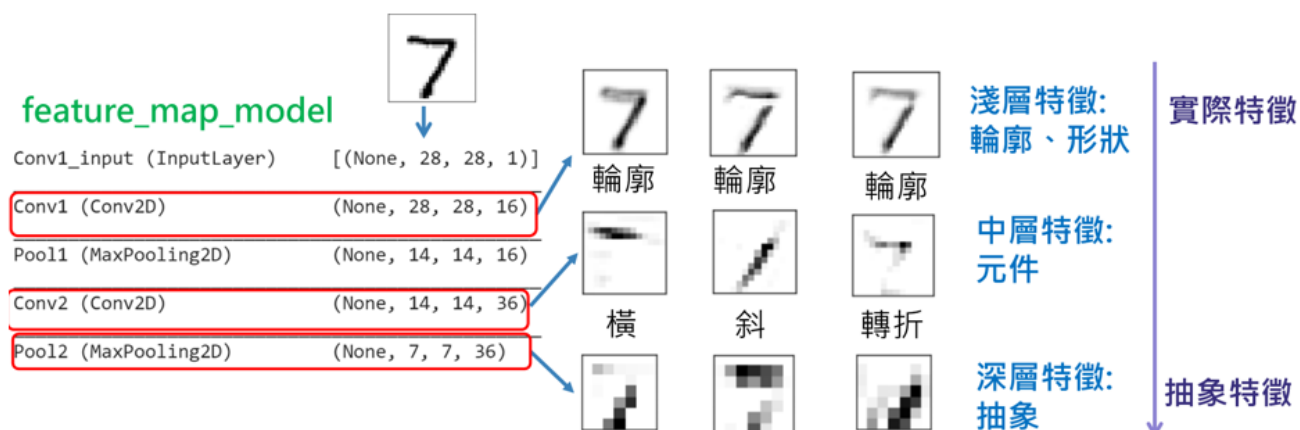
4.訓練後：可有效提取「橫」「斜」元件特徵(特徵清楚)

3.顯示
訓練後
卷積層2
特徵圖

如上圖，卷積層 2：提取中層特徵(例如「橫」、「斜」元件特徵)。你可以看到訓練前後，提取特徵的差異。

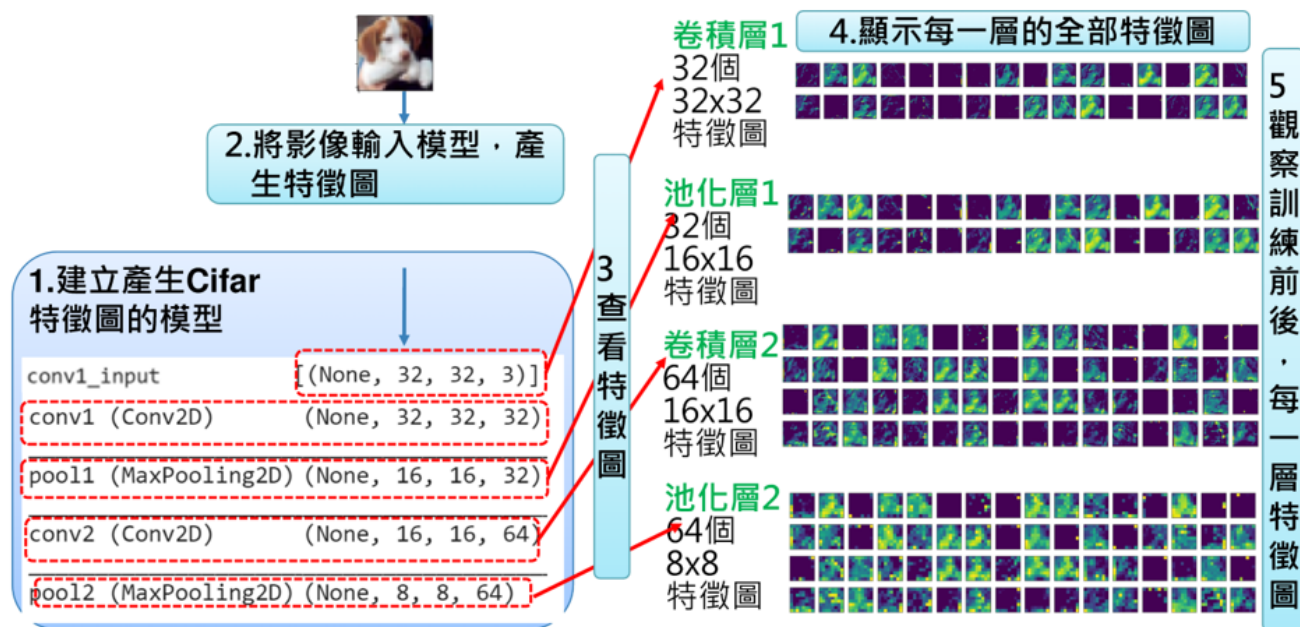
每一層特徵圖的特色

訓練後濾鏡提取每一層特徵圖的特色：由實際特徵到抽象特徵。



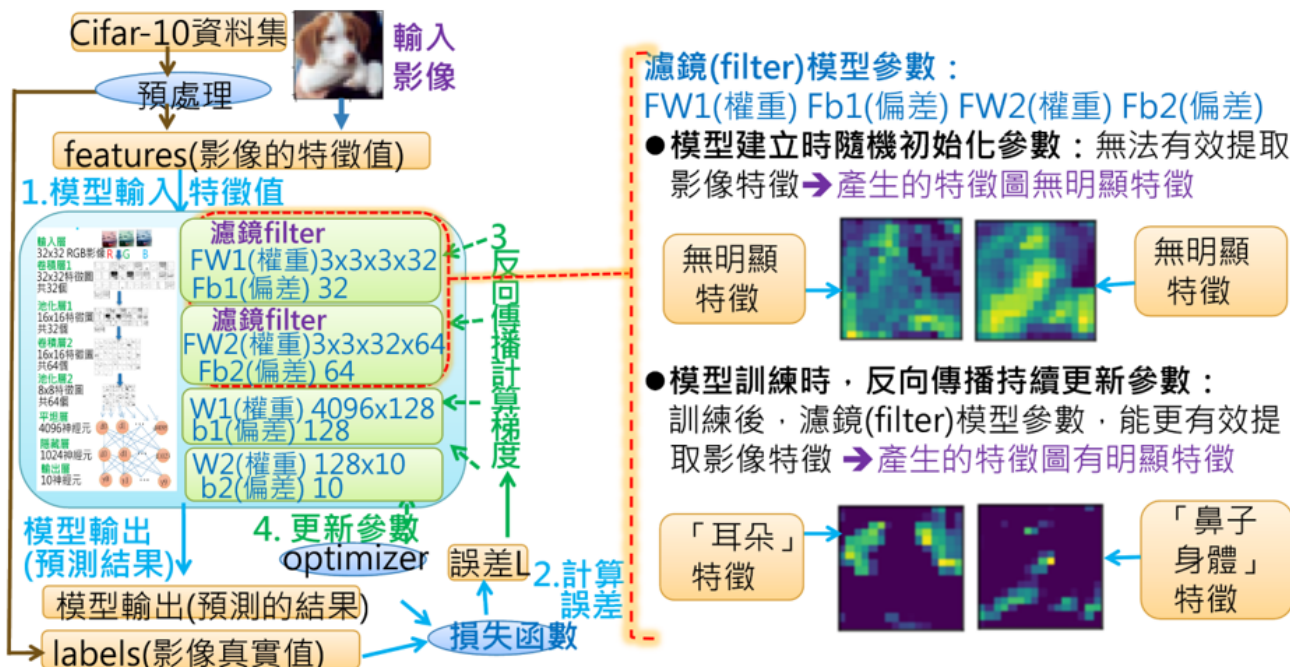
目標：產生 Cifar CNN 模型的特徵圖

依照相同的步驟我們也可以，產生 Cifar CNN 模型的特徵圖，詳細請參考本書說明。



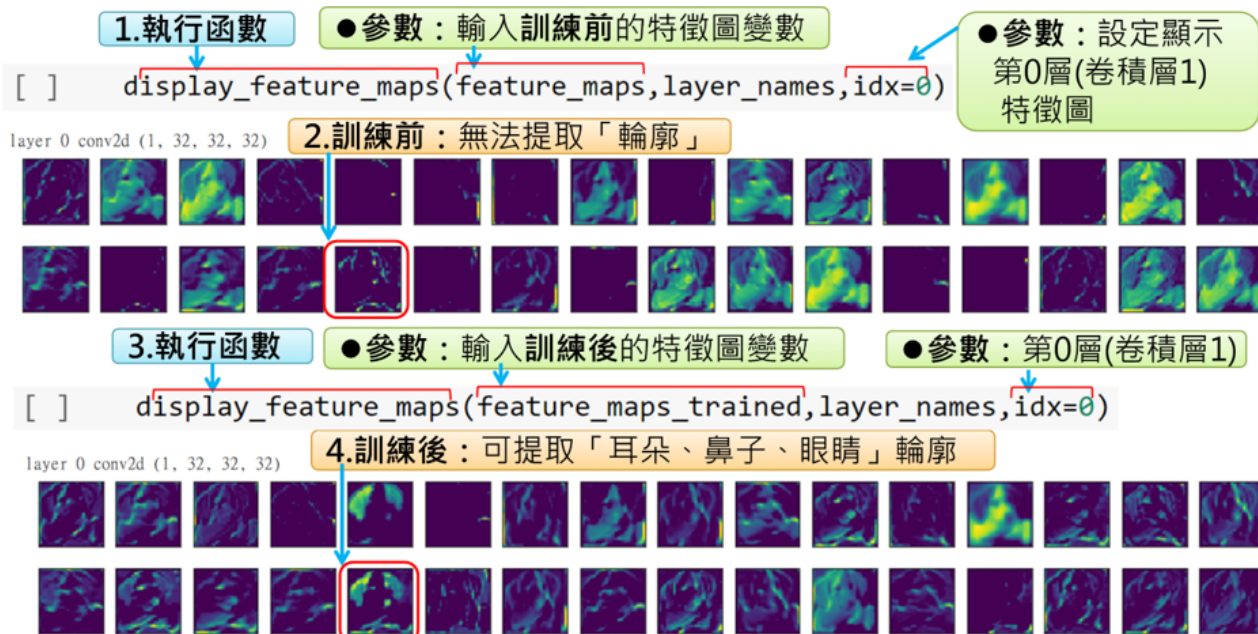
顯示模型訓練前後特徵圖

後續我們將使用，反向傳播演算法訓練模型，模型訓練前後，產生特徵圖差異，說明如下：



Step1.卷積層 1：提取淺層特徵

以下執行函數，輸入參數 `idx=0`，顯示卷積層 1 的訓練前後特徵圖。



如上圖，卷積層 1：提取淺層特徵(輪廓、邊緣、形狀)，如下圖，你可以看到訓練前後，提取特徵的差異。

Step2.卷積層 2：提取中層特徵

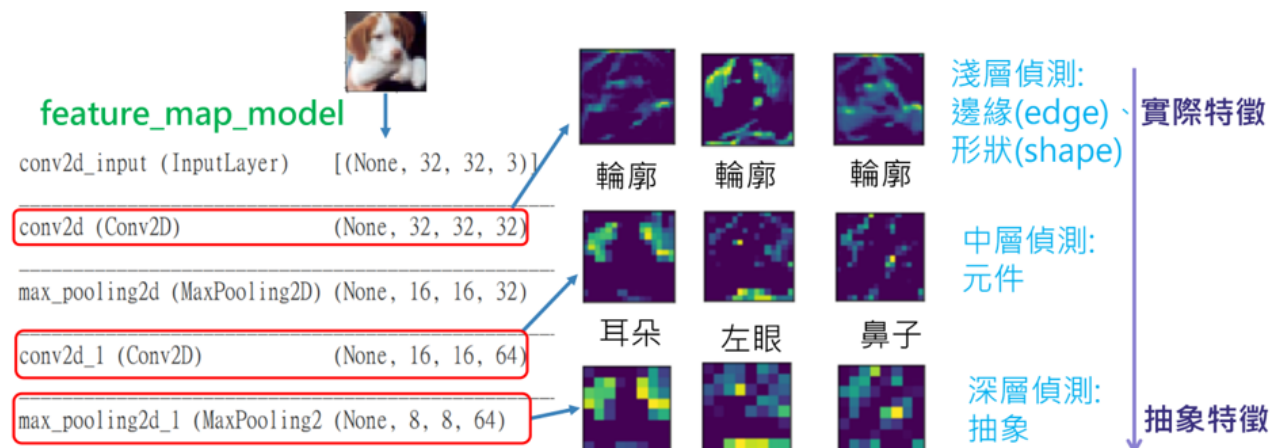
以下執行函數，輸入參數 `idx=2`，顯示卷積層 2 的訓練前後特徵圖。



如上圖，池化層 2 進一步提取特徵，讓特徵更明顯，將低訓練參數。

每一層特徵圖的特色

訓練後，濾鏡提取每一層特徵圖的特色：由實際特徵到抽象特徵。



結論

透過建立輸出 Cifar CNN 特徵圖的模型，呈現每一層特徵圖的結果，能讓我們了解模型如何透過濾鏡提取影像特徵。未來在本系列 Tensorflow 2.0 進階書籍中，還會有詳細介紹 Grad-CAM：這是可解釋性 AI (Explainable AI ，XAI) 的一種方式，讓我們可以了解 CNN 在圖像分類過程中，關注照片中的哪一個區域，做為它判斷分類的依據。

以上內容節錄自這本書，很適合 Deep Learning 深度學習入門初學者:

[圖解 TensorFlow 2 初學篇：實作 tf.keras + Colab 雲端、深度學習、人工智慧、](#)

[影像辨識](#)

[購買本書 限時特價專區》](#)

[天瓏網路書店](#)

[博客來網路書店](#)