

Assignment #3 Image Classification Using Convolution Neural Network

F64126147 胡瑀真

注意事項 1: 書面報告可參考網路資料，但須理解與整理，自己理解與整理後撰寫報告，
使用整段網路內容所獲的評分不高

注意事項 2: 勿分享報告給課程同學，相似內容的報告所獲評分不高

1. 說明是否參考或使用網路上程式，或全部程式自行撰寫？

(允許參考網路程式資源，在書面報告中須說明參考來源與參考程式的範圍。如程式與模型全部自做，亦在報告中說明，有額外分數)

CNN 部分，原先參考第七章講義，製作 VGG-like model，但因準確度不高、訓練時間過長，推測是因深度過深。為改善前述缺點，改參考[Demo] CNN using keras 中的 CNN 模型，製作深度較淺、簡化版的 VGG-like model。

FCN 部分，參考網路程式資源，再修改神經元數量，使 FCN 模型的未知參數數量和 CNN 模型的未知參數數量相近。

2. a) CNN 待求解未知數為何？

b) 描述對 Convolution Neural Network (CNN)的理解

- a) CNN 的待求解未知數 filter 裡的係數，包括權重(weight)和偏差(bias)。舉例而言，若有 5 個 filter，每個 filter size 是 3×3 ，則一個 filter 的待求解未知數為 $3 \times 3 + 1$ ，表示 9 個權重加上 1 個偏差，又因為有 5 個 filter，故總求解未知數為 $5 \times (3 \times 3 + 1) = 50$ ，共有 50 個待求解未知數。

CNN 中，同一層的神經元會共享一組權重，可達到縮減未知數數量，有效率使用未知數的效果。

- b) CNN 是卷積神經網路，因為可以萃取影像特徵，又在多層萃取下，影像辨識的穩定性、功能性較好，故常用於影像辨識的問題。

一個 CNN 模型由 Convolution 和 Pooling layer 疊加，再和 Flatten、Fully-Connected layer、Output layer 連接組成。

模型中，Convolution 的目的即是萃取影像特徵，運用 Filter 滑行過程中和影像的重疊部分做積分，得到卷積結果、新的 feature map。此外，藉由改變 Filter 係數，可改變萃取的特徵，例如：萃取垂直、水平或 45° 的邊界線等。

Pooling layer 則可用來縮減影像尺寸、保留重要資訊並減少資訊量（如：Max pooling 只保留最大值）。在 Convolution 和 Pooling layer 多層堆疊時，因 Pooling layer 會使資訊量減少，故可透過增加 Convolution 的 Filter 數量增加模型厚度。

Flatten 用以將多維 feature map 展平成 1 維向量，以便與 Fully-Connected layer 連接。而 Fully-Connected layer 用於整合特徵，並做分類或回歸，讓 Output layer 輸出結果。

3. 約略相同的模型大小之下，比較 CNN 與 Fully-Connected Network (FCN) 模型分類正確度

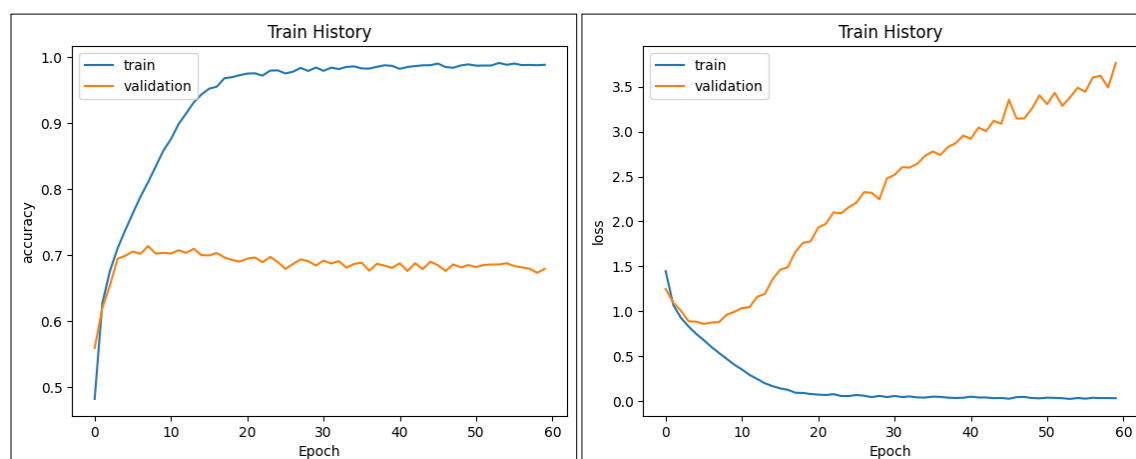
(使用的 FC 與 CNN 模型架構與超參數、FC 與 CNN 模型分類正確度比較)

CNN 架構：

```
1 # Build a neural network
2 #TODO
3 model=Sequential()
4
5 model.add(Conv2D(filters=32,kernel_size=(3,3),padding='same',input_shape=(32,32,3),activation='relu'))
6 model.add(MaxPooling2D(pool_size=(2,2)))
7
8 model.add(Conv2D(filters=64,kernel_size=(3,3),padding='same',activation='relu'))
9 model.add(MaxPooling2D(pool_size=(2,2)))
10
11 model.add(Flatten())
12 model.add(Dense(128,activation='relu'))
13 model.add(Dense(10,activation='softmax'))
14 model.summary()
```

- (3) 運用 Sequential() 建立模型架構。
- (5) 用 add() 語法建立模型的第一層卷積，其中濾波器數量是 32，濾波器尺寸是 3×3，並在 feature map 邊緣補 0，令 feature map 不會愈來愈小，再設定輸入影像大小是 32×32、顏色為 RGB 3 channels 和激勵函數為 ReLU，此層的未知參數量為 $(3 \times 3 \times 3 + 1) \times 32 = 896$ 。
- (6) 加入 pooling layer，將 feature map 壓縮為原本尺寸的一半：16×16，且只保留每 2×2 中的最大數值。
- (8) 類似(5)，加入第二層卷積，並將濾波器數量增加為 64，以彌補在(7)中減少的資訊量，此層的未知參數量為 $(3 \times 3 \times 32 + 1) \times 64 = 18496$ 。
- (9) 類似(6)，feature map 此時被壓縮成 8×8。
- (11) 在模型中加入一層 Flatten，把 feature map 展平成一維向量，以便和之後的 Fully-Connected layer 連接。
- (12) 在模型中加入一層 Fully-Connected layer，並設定有 128 個神經元、激勵函數為 ReLU，此層的未知參數量為 $(8 \times 8 \times 64 + 1) \times 128 = 524416$ 。
- (13) 模型的最後一層 output layer，設定輸出為對應的 10 個類別、激勵函數為 softmax 把輸出轉換成機率，以解決多元分類問題，此層的未知參數量為 $(128 + 1) \times 10 = 1290$ 。
- (14) 用 summary() 列出模型各層的 Output shape 和參數量，可知 CNN 模型的未知參數量為 $896 + 18496 + 524416 + 1290 = 545098$ ，和 FCN 模型的未知參數量(540278)相差不大，兩模型大小相近。

CNN 分類結果：



Accuracy of testing data = 67.6%

CNN 模型的 train accuracy 上升快速，在約 15 epoch 時接近 100%，而 validation accuracy 則快速上升，且穩定在約 70%。此外，train loss 下降快速，同樣在約 15 epoch 時接近 0，而 validation loss 在 0 至 5 epoch 快速下降，但在 5 epoch 後穩定上升，在 60 epoch 時超過 3.5。

由於 train loss 逐漸下降，且 train accuracy 持續上升且近 100%，但 validation loss 卻在 5 epoch 後逐漸上升，表示模型有輕微過擬合的現象，因 validation accuracy 仍穩定在約 70%，沒有發生 accuracy 驟降或 loss 飆升，故不是嚴重的過擬合現象。

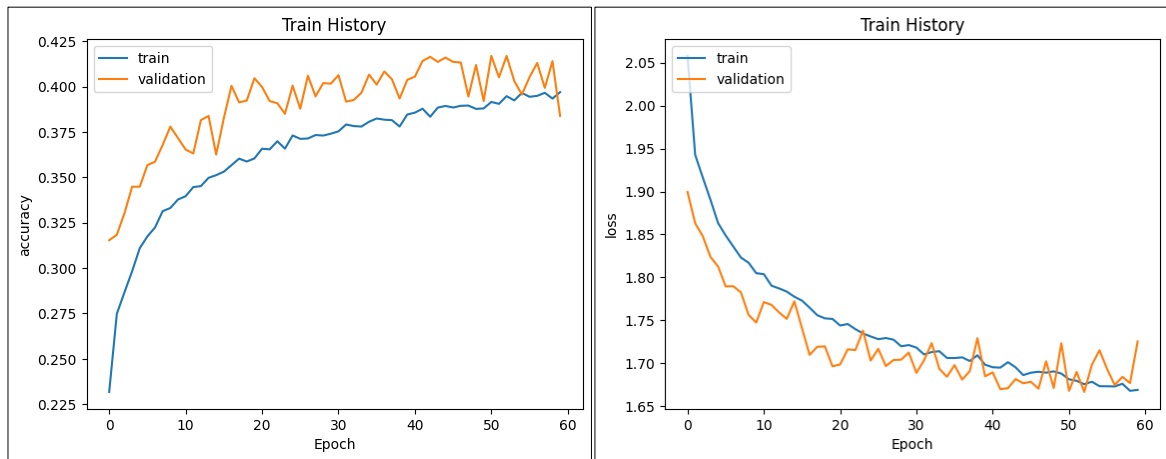
FCN 架構：

```
1 # Build a FCN model
2 fcn_model = Sequential()
3
4 fcn_model.add(Flatten(input_shape=(32, 32, 3)))
5
6 fcn_model.add(Dense(172, activation='relu'))
7 fcn_model.add(Dropout(0.3))
8 fcn_model.add(Dense(64, activation='relu'))
9 fcn_model.add(Dense(10, activation='softmax'))
10
11 fcn_model.summary()
```

- (2) 運用 Sequential() 建立模型架構。
- (4) 用 add() 語法，在模型中加入一層 Flatten，並設定輸入影像尺寸為 32×32，顏色為 RGB 3 channels，使多維 (32×32×3=3072 維) 影像展平成一維向量。
- (6) 在模型中加入一層 Fully-Connected layer，並設定有 172 個神經元、激勵函數為 ReLU，此層的未知參數量為 3072×172+172=528556。
- (7) 隨機忽略 172 個神經元中的 30%。
- (8) 類似(5)，但設定神經元數量為 64，此層的未知參數量為 172×64+64=11072。
- (9) 模型的最後一層 output layer，設定輸出為對應的 10 個類別、激勵函數為 softmax 把輸出轉換成機率，以解決多元分類問題，此層的未知參數量為 64×10+10=650。

- (11) 用 summary() 列出模型各層的 Output shape 和參數量，可知 FCN 模型的未知參數量為 $528556+11072+650=540278$ ，和 CNN 模型的未知參數量(545098)相差不大，兩模型大小相近。

FCN 分類結果：



Accuracy of testing data = 38.2%

FCN 模型的 train accuracy 起初上升快速，在 10 epoch 後上升速度雖減緩，但仍持續上升，在 60 epoch 時約在 40%。而 validation accuracy 明顯略好於 train accuracy，但震盪較 train accuracy 劇烈，在 37.5%至 42.5%間震盪。此外，train loss 和 validation loss 皆在初始有快速下降，即使在 5 epoch 後下降速度減緩，但仍穩定下降，且 validation loss 震盪較 train loss 劇烈。

從 train history 的兩圖可知，兩 loss 曲線皆穩定下降，且兩 accuracy 曲線穩定上升且 accuracy 相近，因此 FCN 模型的訓練沒有過擬合情況。

FC 與 CNN 模型分類準確度比較：

CNN 模型的未知參數數量是 545,098，其分類準確度是 67.6%；而 FCN 模型的未知參數數量是 540,278，其分類準確度是 38.2%。

兩模型的超參數設定，皆為 epoch = 60, batch size = 64, optimizer = Adam, activation function = ReLU + Softmax，可知在超參數設定相同、模型大小相近的情況，CNN 的準確度顯著大於 FCN。

4. 接續第 3 點，CNN 與 FCN 那個模型在影像分類的問題上有較佳準確度？為什麼？

CNN 模型在影像分類問題有明顯較佳的準確度。推測是因為 CNN 有「保留影像 pixel 與 pixel 之間位置關係」的特性，在 Convolution 中，藉由積分 Filter 滑動過程中，與影像重疊部分的面積，製作出 Feature map，例如：透過具有特定權重的 Filter，使影像的特徵銳利化，或模糊化不重要的資訊，萃取出影像特徵。

並且，CNN 模型運用堆疊多層 Convolution 和 Pooling layer 的組合，逐步萃取出

影像特徵，之後使用 Flatten，令多維 Feature map 攤平成一維向量，再與 Fully-Connected layer 連接，將萃取結果整合、分類，最後輸出分類結果。

相較之下，FCN 模型在一開始，便使用 Flatten 將多維影像攤平成一維向量，接著透過多層 Fully-Connected layer 學習影像特徵和分類，導致 FCN 模型的建立沒有考量到影像 pixel 與 pixel 間的位置關係，忽略 pixel 間的相對位置關係。

此外，CNN 模型中，同一層卷積會共用相同的 Filter、共享同一組權重，讓共同的 Filter 依序、逐步地尋找影像的特定特徵，令 CNN 模型有「多層萃取下，影像辨識的穩定性較好」的優點；相對的，FCN 模型因沒有共享權重，導致其泛化能力、訓練速度較差。

綜上所述，CNN 模型因為使用 Filter 做卷積運算，達到保留 pixel 之間的空間關係，並擁有較佳的影像辨識穩定性、泛化能力和訓練速度，令 CNN 在影像分類問題有較 FCN 佳的準確度。

參考資料（書面報告的參考資料）

1. 上課講義.....
2. <http://www.deeplearning.com>