

Homework 3 Report

學號：B05902052 系級：資工二 姓名：劉家維

1. (1%) 請說明你實作的 CNN model，其模型架構、訓練參數和準確率為何？

Model 1: public/private: 0.57286 / 0.57397

Architecture:

```
Conv2D(filter 大小=(3, 3), filter 個數=32, activation=relu)
Conv2D(filter 大小=(3, 3), filter 個數=32, activation=relu)
MaxPooling2D(Pooling 大小=(2, 2))
Conv2D(filter 大小=(3, 3), filter 個數=32, activation=relu)
Conv2D(filter 大小=(3, 3), filter 個數=32, activation=relu)
MaxPooling2D(Pooling 大小=(2, 2))
Conv2D(filter 大小=(3, 3), filter 個數=32, activation=relu)
Conv2D(filter 大小=(3, 3), filter 個數=32, activation=relu)
MaxPooling2D(Pooling 大小=(2, 2))
Flatten()
Dense(hidden size=64, activation=relu)
Dense(hidden size= 7, activation=softmax)
Optimizer = RMSprop, lr=0.001, rho=0.9
```

Model 2: public/private: 0.62245 / 0.61103

Architecture:

```
Conv2D(filter 大小=(3, 3), filter 個數=32, activation=relu)
Conv2D(filter 大小=(3, 3), filter 個數=32, activation=relu)
MaxPooling2D(Pooling 大小=(2, 2))
Conv2D(filter 大小=(3, 3), filter 個數=64, activation=relu)
Conv2D(filter 大小=(3, 3), filter 個數=64, activation=relu)
MaxPooling2D(Pooling 大小=(2, 2))
Conv2D(filter 大小=(3, 3), filter 個數=128, activation=relu)
Conv2D(filter 大小=(3, 3), filter 個數=128, activation=relu)
MaxPooling2D(Pooling 大小=(2, 2))
Flatten()
Dense(hidden size=256, activation=relu)
Dense(hidden size=128, activation=relu)
Dense(hidden size= 64, activation=relu)
Dense(hidden size= 7, activation=softmax)
Optimizer = RMSprop, lr=0.001, rho=0.9
```

Model 3: public/private: 0.68208 / 0.67706

```

定義一個 ResLayer(num, size, strides) 為：
Conv0 = Conv2D(filter大小=size, filter個數=num, 間隔大小=strides)(Input)
Conv0_bn = BatchNormalization(Conv0)
Conv0_rl = ReLU(conv0_bn)
Conv1 = Conv2D(filter大小=size, filter個數=num, 間隔大小=1)(Conv0_rl)
Conv1_bn = BatchNormalization(Conv1)
Conv1_rl = ReLU(conv1_bn)
Output = Input + Conv1_rl
(Output = Input + 兩次(Conv2D + BatchNormalization + ReLU)後的 Input)

```

Architecture:
(34-layer ResNet)

```

Conv2D(filter大小=(5, 5), filter個數=64, activation=relu)
BatchNormalization()
ReLU()
ResLayer(num=64, size=(3, 3), strides=1)
ResLayer(num=64, size=(3, 3), strides=1)
ResLayer(num=64, size=(3, 3), strides=1)
ResLayer(num=128, size=(3, 3), strides=2)
ResLayer(num=128, size=(3, 3), strides=1)
ResLayer(num=128, size=(3, 3), strides=1)
ResLayer(num=128, size=(3, 3), strides=1)
ResLayer(num=256, size=(3, 3), strides=2)
ResLayer(num=256, size=(3, 3), strides=1)
ResLayer(num=256, size=(3, 3), strides=1)
ResLayer(num=256, size=(3, 3), strides=1)
ResLayer(num=256, size=(3, 3), strides=1)
ResLayer(num=256, size=(3, 3), strides=1)
ResLayer(num=512, size=(3, 3), strides=2)
ResLayer(num=512, size=(3, 3), strides=1)
ResLayer(num=512, size=(3, 3), strides=1)
AveragePooling2D(pool_size=(1,1))
Flatten()
Dense(hidden size=1024, activation=relu)
Dense(hidden size=7, activation=softmax)
Optimizer = RMSprop, lr=0.001, rho=0.9

```

2. (1%) 請嘗試 data normalization, data augmentation, 說明實行方法並且說明對準確率有什麼樣的影響？

Data Normalization: 用所有圖片的 pixel (28709 * 48 * 48) 求出平均值和標準差之後，將所有 pixel 減去平均值再除以標準差。模型使用第一題的 Model 1。

有做 Data Normalization: public/private: 0.57286 / 0.57397

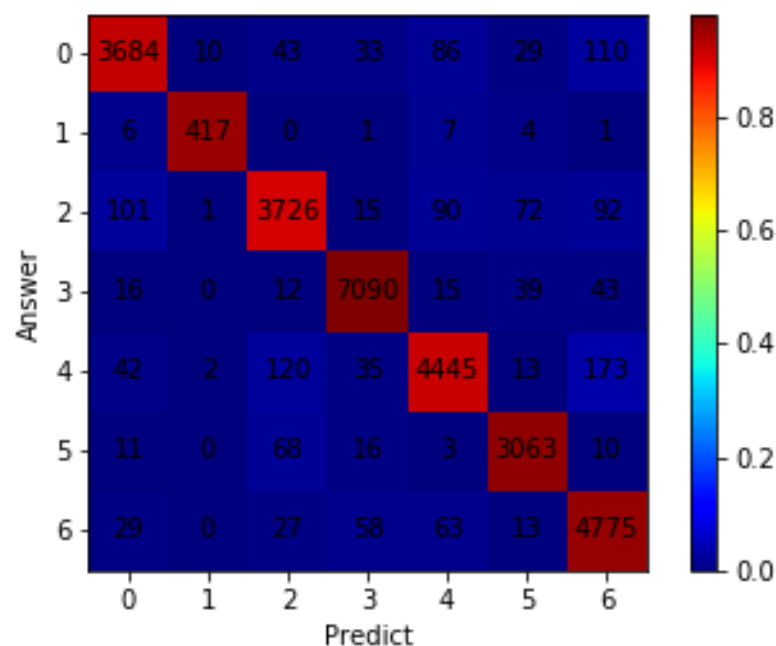
沒做 Data Normalization: public/private: 0.55502 / 0.55586

Data Augmentation: 將 training data 分割成真正的 training set 和額外的 validation set, 然後訓練時每個 epoch 都將 training set 的每張 image 隨機轉 -10~10 度, 隨機水平/垂直移動 -5~5 個 pixel, 隨機縮放 -0.1~0.1 倍, 隨機水平翻轉。模型使用第一題的 Model 1。

有做 Data Augmentation: public/private: 0.57286 / 0.57397

沒做 Data Augmentation: public/private: 0.40178 / 0.41515

3. (1%) 觀察答錯的圖片中, 哪些 class 彼此間容易用混? [繪出 confusion matrix 分析]



因為沒有 test data 的資料, 所以我用 training data 進行 confusion matrix 的分析。

(1: 厭惡, 2: 恐懼, 3: 高興, 4: 難過, 5: 驚訝, 6: 中立)

可以發現, model 最容易將 2 誤認為 4, 6, 將 4 誤認為 2, 6

4. (1%) 從(1)(2)可以發現, 使用 CNN 的確有些好處, 試繪出其 saliency maps, 觀察模型在做 classification 時, 是 focus 在圖片的哪些部份?

答: 合理說明 test 的圖片和觀察到的東西 -> 0.5 分

貼出 saliency 圖片 -> 0.5 分

5. (1%) 承(4) 利用上課所提到的 gradient ascent 方法, 觀察特定層的 filter 最容易被哪種圖片 activate 與觀察 filter 的 output。(Collaborators:)

答: 合理說明 test 的層數和觀察到的東西 -> 0.5 分

貼出 filter input and output 的圖片 -> 0.5 分

