

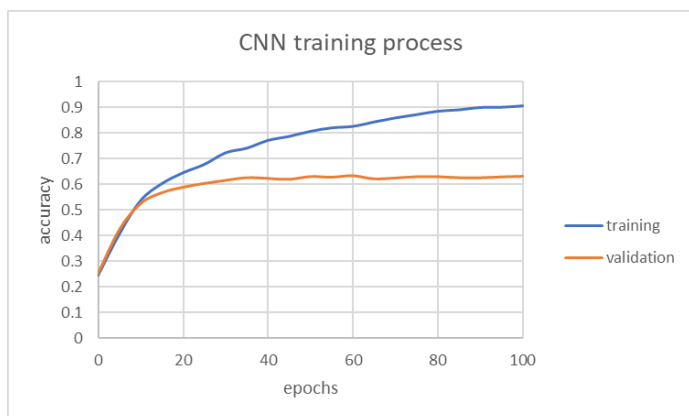
# ML2017FALL-HW3-Report

學號：B03801039 系級：電機四 姓名：楊福恩

1. (1%) 請說明你實作的 CNN model，其模型架構、訓練過程和準確率為何？  
(Collaborators: 陳學平 B03901103，廖宜倫 B03901001)

答：

Layer (type)	Output Shape	Param #
conv2d_1 (Conv2D)	(None, 48, 48, 64)	640
dropout_1 (Dropout)	(None, 48, 48, 64)	0
conv2d_2 (Conv2D)	(None, 48, 48, 64)	36928
dropout_2 (Dropout)	(None, 48, 48, 64)	0
max_pooling2d_1 (MaxPooling2D)	(None, 24, 24, 64)	0
conv2d_3 (Conv2D)	(None, 24, 24, 128)	73856
dropout_3 (Dropout)	(None, 24, 24, 128)	0
conv2d_4 (Conv2D)	(None, 24, 24, 128)	147584
dropout_4 (Dropout)	(None, 24, 24, 128)	0
max_pooling2d_2 (MaxPooling2D)	(None, 12, 12, 128)	0
conv2d_5 (Conv2D)	(None, 12, 12, 256)	295168
dropout_5 (Dropout)	(None, 12, 12, 256)	0
conv2d_6 (Conv2D)	(None, 12, 12, 256)	590880
dropout_6 (Dropout)	(None, 12, 12, 256)	0
max_pooling2d_3 (MaxPooling2D)	(None, 6, 6, 256)	0
conv2d_7 (Conv2D)	(None, 6, 6, 512)	1180160
dropout_7 (Dropout)	(None, 6, 6, 512)	0
conv2d_8 (Conv2D)	(None, 6, 6, 512)	2359808
dropout_8 (Dropout)	(None, 6, 6, 512)	0
max_pooling2d_4 (MaxPooling2D)	(None, 3, 3, 512)	0
flatten_1 (Flatten)	(None, 4608)	0
dropout_9 (Dropout)	(None, 4608)	0
dense_1 (Dense)	(None, 4096)	18878464
dropout_10 (Dropout)	(None, 4096)	0
dense_2 (Dense)	(None, 4096)	16781312
dropout_11 (Dropout)	(None, 4096)	0
dense_3 (Dense)	(None, 7)	28679
Total params: 40,372,679		
Trainable params: 40,372,679		
Non-trainable params: 0		



模型描述與訓練過程：

先參考了 VGG16 的架構，每一層的 filter 個數皆以 8 的倍數作為基礎，並且將 filter 大小皆選為 (3, 3)，在經過 50 多個不同模型的嘗試且考量預測時間後，決定了如右圖所示的架構，並且每兩層 Conv2D layer 後都做 max pooling，參數選擇為 pool size=(2, 2)，strides=(2, 2)。另外因為 training accuracy 在每個 model 中皆大於 validation accuracy，所以除了輸出層外每一層 NN 後皆做 dropout，Conv2D 後的 dropout 參數取 0.3，Dense 後的參數取 0.5，但 overfitting 依然十分明顯；optimizer 則是選擇 Adamax；訓練時取 batch size=128；中間層 activation function 為 relu，輸出層為 softmax。此外對於輸入數據有做 normalize，將平均設為 0，標準差設為 1。

Accuracy:

Training: 0.9050

Validation: 0.63888

Public: 0.65979

Private: 0.66508

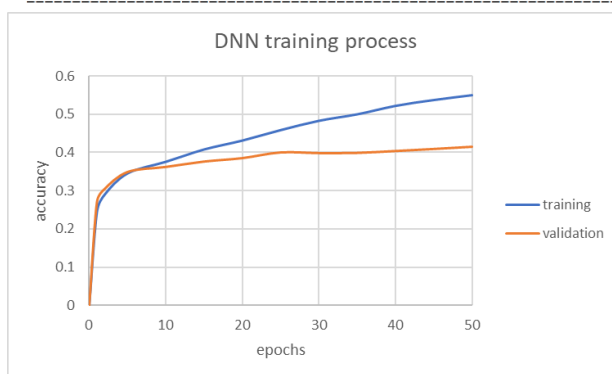
Average(pub. & pri.): 0.66244

2. (1%) 承上題，請用與上述 CNN 接近的參數量，實做簡單的 DNN model。其模型架構、訓練過程和準確率為何？試與上題結果做比較，並說明你觀察到了什麼？

(Collaborators: None )

答：

Layer (type)	Output Shape	Param #
dense_1 (Dense)	(None, 256)	590080
dropout_1 (Dropout)	(None, 256)	0
dense_2 (Dense)	(None, 256)	65792
dropout_2 (Dropout)	(None, 256)	0
dense_3 (Dense)	(None, 512)	131584
dropout_3 (Dropout)	(None, 512)	0
dense_4 (Dense)	(None, 512)	262656
dropout_4 (Dropout)	(None, 512)	0
dense_5 (Dense)	(None, 1024)	525312
dropout_5 (Dropout)	(None, 1024)	0
dense_6 (Dense)	(None, 2048)	2099200
dropout_6 (Dropout)	(None, 2048)	0
dense_7 (Dense)	(None, 2048)	4196352
dropout_7 (Dropout)	(None, 2048)	0
dense_8 (Dense)	(None, 4096)	8392704
dropout_8 (Dropout)	(None, 4096)	0
dense_9 (Dense)	(None, 4096)	16781312
dropout_9 (Dropout)	(None, 4096)	0
dense_10 (Dense)	(None, 7)	28679
Total params: 33,073,671		
Trainable params: 33,073,671		
Non-trainable params: 0		



模型描述與訓練過程：

選用與 problem 1 的 CNN model 同個數量級的參數量，明顯可見正確率低於 CNN model，且在訓練過程中可發現 CNN 每個 epoch 訓練時間 (82 秒) 約為 DNN 每個 epoch 訓練時間 (12 秒) 的 7 倍左右。此外每一層也都有加上 dropout，前 7 層 dropout 參數取 0.3，後兩層 dropout 參數取 0.5，中間層的 activation function 皆選擇 relu，輸出層為 softmax。故由正確率可知，處理影像方面 CNN model 優於 DNN，但由運算時間看 CNN 等待結果的虐心程度遠大於 DNN。

Accuracy:

Training: 0.55

Validation: 0.42837

Public: 0.43156

Private: 0.43354

Average(pub. & pri.): 0.43255

註 1: problem 1 和 2 training process 的圖，繪製方式為將 DeepQ 平台上 train 完模型後的 console log 檔內的 training accuracy 與 validation accuracy 記錄下來，用 excel 繪製。

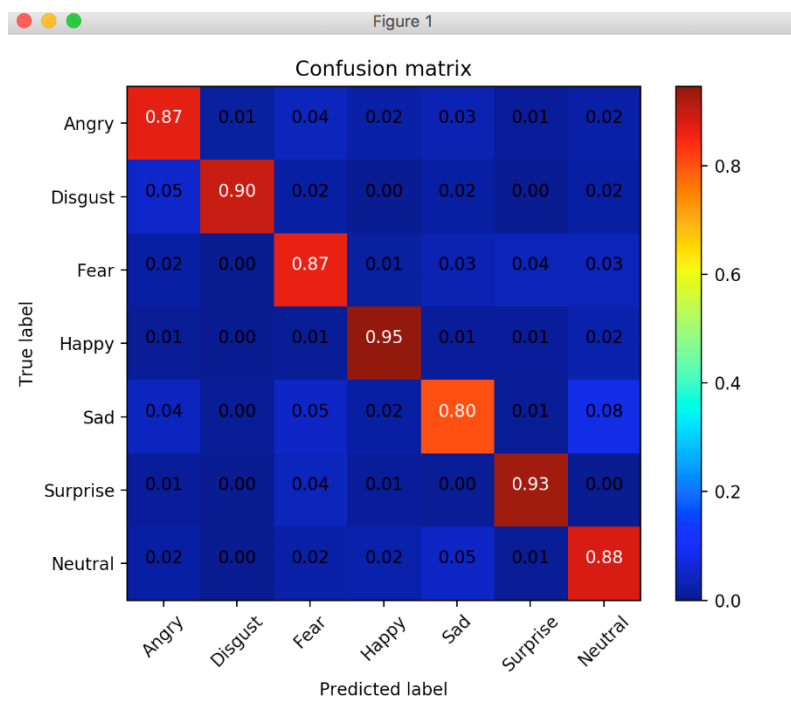
註 2: 訓練以上模型時皆使用全部 training data，validation accuracy 為使用 Keras 在每一個 epoch 中隨機切 15% 當 validation。

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]

(Collaborators: None )

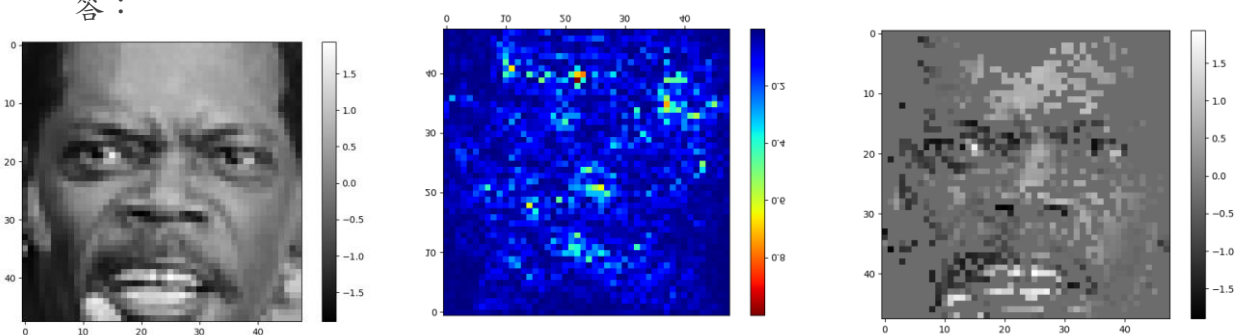
答：

因為根據以上的實驗結果發現 training data 的數據量愈多時，正確率愈高，故在 kaggle 上較高分的 model 都是用全部 training data 下去 train，為了較貼近 kaggle 上的模型，所以用於此題的 model 所切的 validation data 並不多，故 confusion matrix 所呈現的可能比較能看出 training 過程中對於各個 class 的判斷準確度，由下圖可看出 Happy 的判斷準確率最高，推測可能原因應為高興時會有嘴角上揚等較明確會出現的表情特徵，讓機器較容易經過大量數據的學習來判別。而 Sad 的準確率則最低，推測可能原因為這 7 個 class 中，屬於負面情緒的 class 較多，而 sad 屬於表情特徵較不明顯的，故容易被判斷成其他情緒。



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？  
(Collaborators: None)

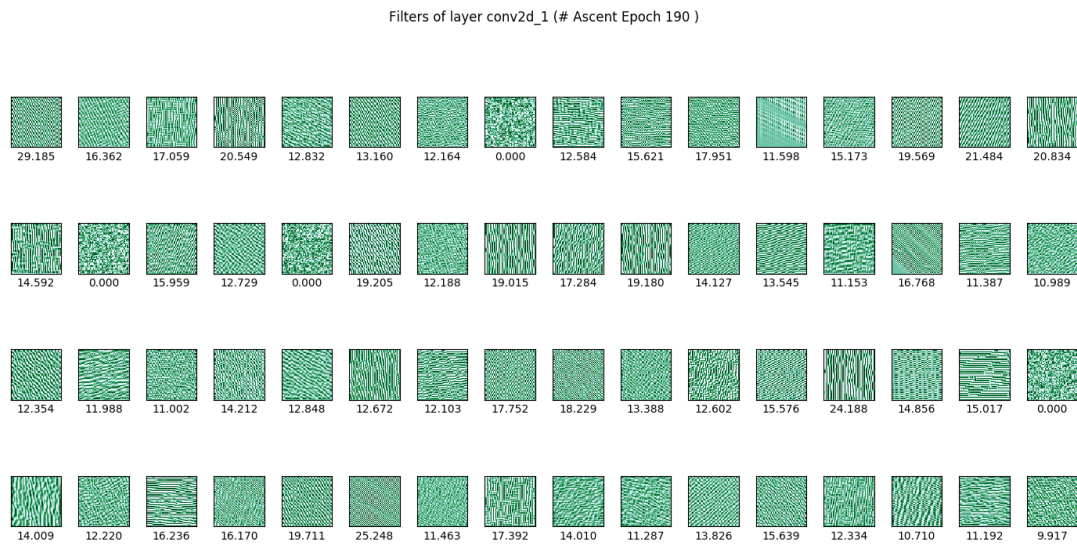
答：



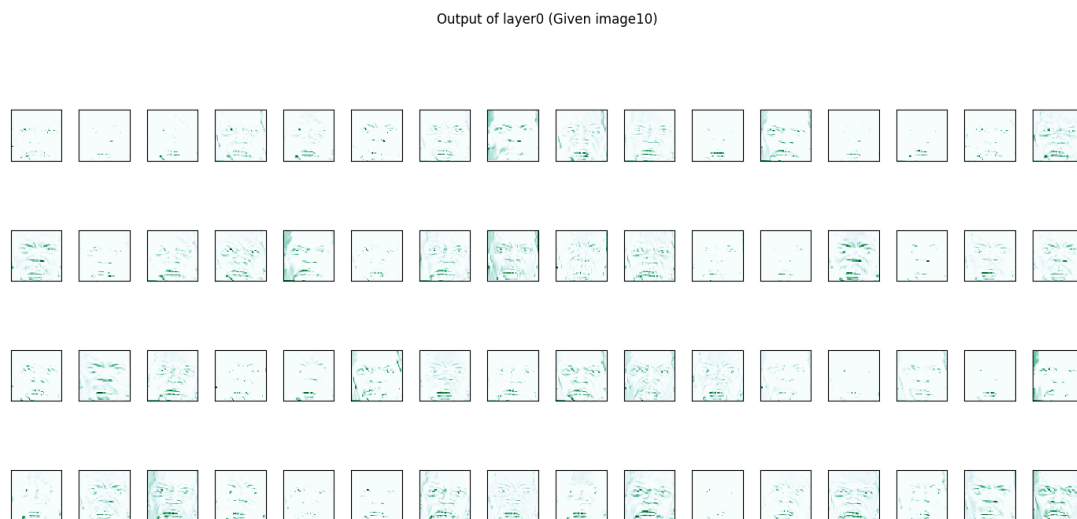
此表情的情緒為 angry，可以看出模型 focus 主要在眼睛與嘴巴附近的位置。

5. (1%) 承(1)(2)，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。  
(Collaborators: None)

答：



上圖為能使得第一層 convolution layer 有最大 activate 的圖片，可看出能最大激發第一層 filter 的圖案大多為基本的紋理



上圖為第一層 convolution layer 的 output。