

學號：B03705028 系級：資管三 姓名：陳星宇

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？
答：

Validation set 為 train.csv 前十分之一的 data。

● 模型架構

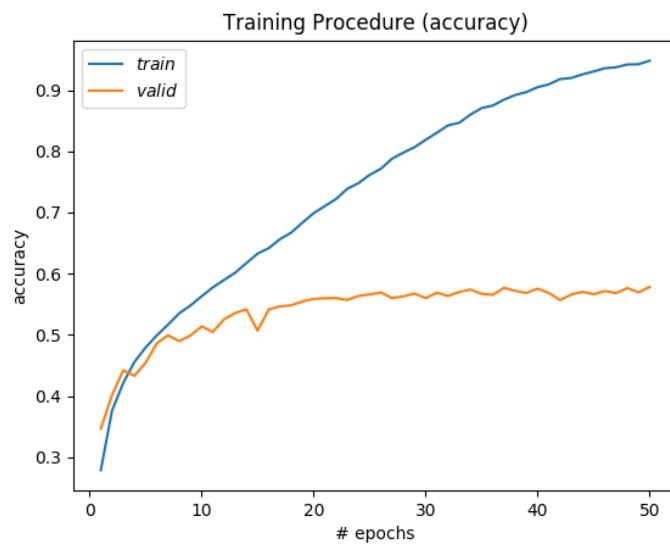
Layer (type)	Output Shape	Param #
conv2d_1 (Conv2D)	(None, 44, 44, 64)	1664
p_re_lu_1 (PReLU)	(None, 44, 44, 64)	123904
zero_padding2d_1 (ZeroPaddin	(None, 48, 48, 64)	0
max_pooling2d_1 (MaxPooling2	(None, 22, 22, 64)	0
zero_padding2d_2 (ZeroPaddin	(None, 24, 24, 64)	0
conv2d_2 (Conv2D)	(None, 22, 22, 64)	36928
p_re_lu_2 (PReLU)	(None, 22, 22, 64)	30976
zero_padding2d_3 (ZeroPaddin	(None, 24, 24, 64)	0
conv2d_3 (Conv2D)	(None, 22, 22, 64)	36928
p_re_lu_3 (PReLU)	(None, 22, 22, 64)	30976
average_pooling2d_1 (Average	(None, 10, 10, 64)	0
zero_padding2d_4 (ZeroPaddin	(None, 12, 12, 64)	0
conv2d_4 (Conv2D)	(None, 10, 10, 128)	73856
p_re_lu_4 (PReLU)	(None, 10, 10, 128)	12800

zero_padding2d_5 (ZeroPaddin	(None, 12, 12, 128)	0
conv2d_5 (Conv2D)	(None, 10, 10, 128)	147584
p_re_lu_5 (PReLU)	(None, 10, 10, 128)	12800
zero_padding2d_6 (ZeroPaddin	(None, 12, 12, 128)	0
average_pooling2d_2 (Average	(None, 5, 5, 128)	0
flatten_1 (Flatten)	(None, 3200)	0
dense_1 (Dense)	(None, 1024)	3277824
p_re_lu_6 (PReLU)	(None, 1024)	1024
dropout_1 (Dropout)	(None, 1024)	0
dense_2 (Dense)	(None, 1024)	1049600
p_re_lu_7 (PReLU)	(None, 1024)	1024
dropout_2 (Dropout)	(None, 1024)	0
dense_3 (Dense)	(None, 7)	7175
=====		
Total params: 4,845,063		
Trainable params: 4,845,063		
Non-trainable params: 0		

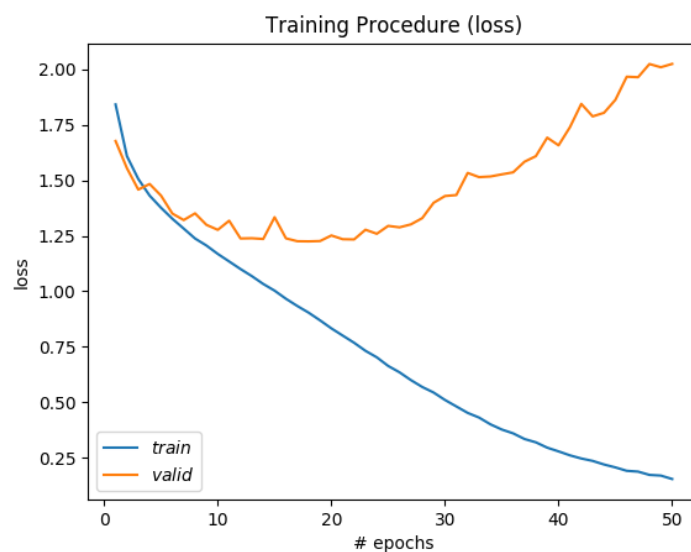
- 訓練過程

Batch_size = 64, epochs = 50

Training accuracy 穩定上升，但 validation accuracy 在十幾個 epochs 之後，就開始在震盪中以極緩慢的速度上升 (幾乎停滯)。



Training loss 穩定下降，但 validation loss 在十幾個 epochs 之後，就開始慢慢上升。



- 準確率

在第 50 個 epoch 後， validation accuracy 為 0.5782

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？

答：

Validation set 為 train.csv 前十分之一的 data。(與第一題相同)

- 模型架構

Layer (type)	Output Shape	Param #
max_pooling2d_1 (MaxPooling2)	(None, 22, 22, 1)	0
average_pooling2d_1 (Average)	(None, 10, 10, 1)	0
flatten_1 (Flatten)	(None, 100)	0
dense_1 (Dense)	(None, 256)	25856
p_re_lu_1 (PReLU)	(None, 256)	256
dropout_1 (Dropout)	(None, 256)	0
dense_2 (Dense)	(None, 469)	120533
p_re_lu_2 (PReLU)	(None, 469)	469
dropout_2 (Dropout)	(None, 469)	0
dense_3 (Dense)	(None, 1024)	481280
p_re_lu_3 (PReLU)	(None, 1024)	1024

dropout_3 (Dropout)	(None, 1024)	0
dense_4 (Dense)	(None, 1024)	1049600
p_re_lu_4 (PReLU)	(None, 1024)	1024
dropout_4 (Dropout)	(None, 1024)	0
dense_5 (Dense)	(None, 1024)	1049600
p_re_lu_5 (PReLU)	(None, 1024)	1024
dropout_5 (Dropout)	(None, 1024)	0
dense_6 (Dense)	(None, 2048)	2099200
p_re_lu_6 (PReLU)	(None, 2048)	2048
dropout_6 (Dropout)	(None, 2048)	0
dense_7 (Dense)	(None, 7)	14343
Total params: 4,846,257		
Trainable params: 4,846,257		
Non-trainable params: 0		

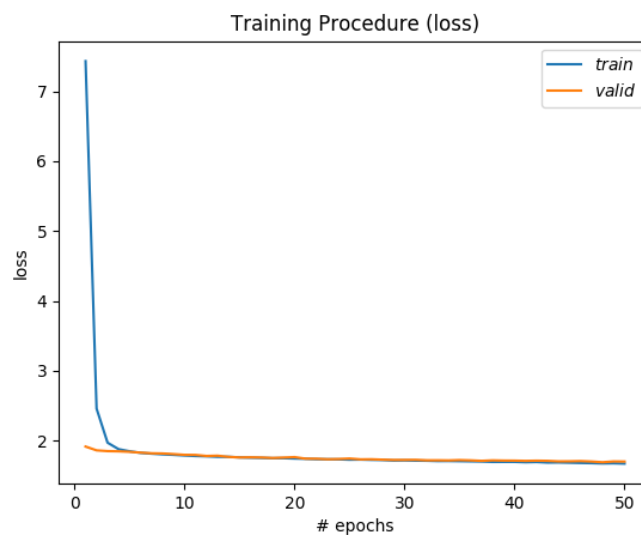
- 訓練過程

Batch_size = 64, epochs = 50 (與第一題相同)

Training accuracy 與 validation accuracy 都持續緩慢上升 (與第一題 accuracy 上升的速率相比)。



Training loss 與 validation loss 都持續以極緩慢的速率下降。



- 準確率

在第 50 個 epoch 後， validation accuracy 為 0.3170

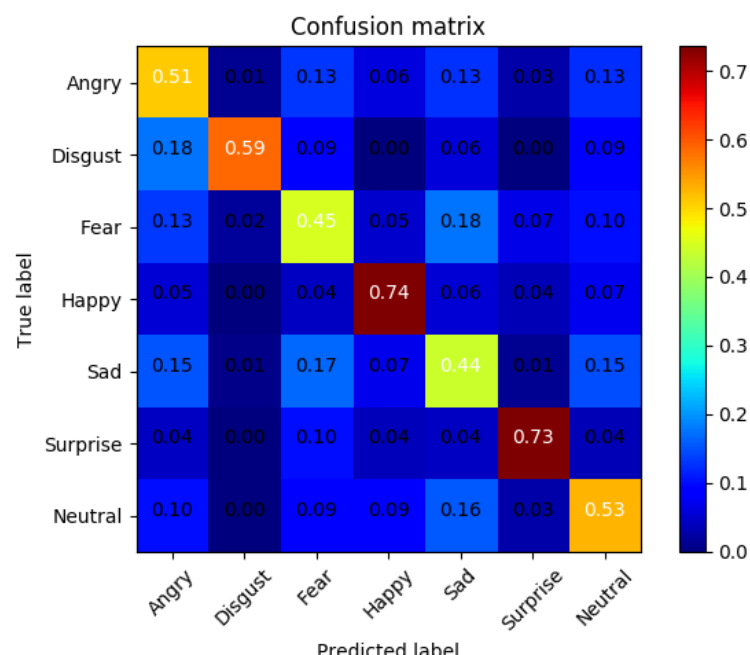
- 與上題結果做比較，觀察到了什麼

DNN 的訓練速度明顯比 CNN 緩慢許多，準確率也比 CNN 低許多。而且在第 50 個 epoch 過後，DNN 的 validation accuracy 甚至有一點下降的趨勢，因此我推論就算讓 DNN 訓練足夠的 epoch，它的準確率可能也無法與 CNN 相提並論。

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]

答：

- Confusion Matrix



由上圖可以發現，我的 CNN model 最容易將 Fear 與 Sad 弄混 (將 Fear 誤判為 Sad : 0.18, 將 Sad 誤判為 Fear : 0.17)，因此我決定從這兩個 class 之中，挑選幾張我的 model 判斷錯誤的圖片，並分析 model 在判斷它們屬於哪個 class 時的 probability distributions。

- Probability Distributions

	Angry	Disgust	Fear	Happy	Sad	Surprise	Neutral
圖 1	0.00005	0.00000	0.99138	0.00355	0.00476	0.00001	0.00025
圖 2	0.00001	0.00000	0.00071	0.00000	0.99920	0.00000	0.00008
圖 3	0.00002	0.00000	0.04944	0.05692	0.50478	0.00009	0.38874
圖 4	0.01200	0.00844	0.49331	0.01345	0.45220	0.00006	0.02054

圖 1~圖 4 (由左至右)



我特別挑了兩張 **Fear** 跟 **Sad** 的機率相差懸殊的，與兩張兩個 **class** 的機率相差無幾的。經人眼觀察後發現，在四張圖中，圖 1、圖 3、圖 4 都蠻難判斷應該要屬於哪一個 **class**，但這也沒辦法，答案就是只能給一個。但圖 2 我個人認為她看起來蠻 **Sad**，我的 **model** 也說她 99% 的機率是 **Sad**，但答案是給 **Fear**，這是其中比較奇怪的地方。

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：

我在 validation set 中挑了 model 預測正確、且機率皆高於 99% 的七張照片，並附上他們的原始影像與相對應的 Saliency Map。(由左至右分別為 Sad, Happy, Fear, Angry, Disgust, Neutral, Surprise)

- Original Images



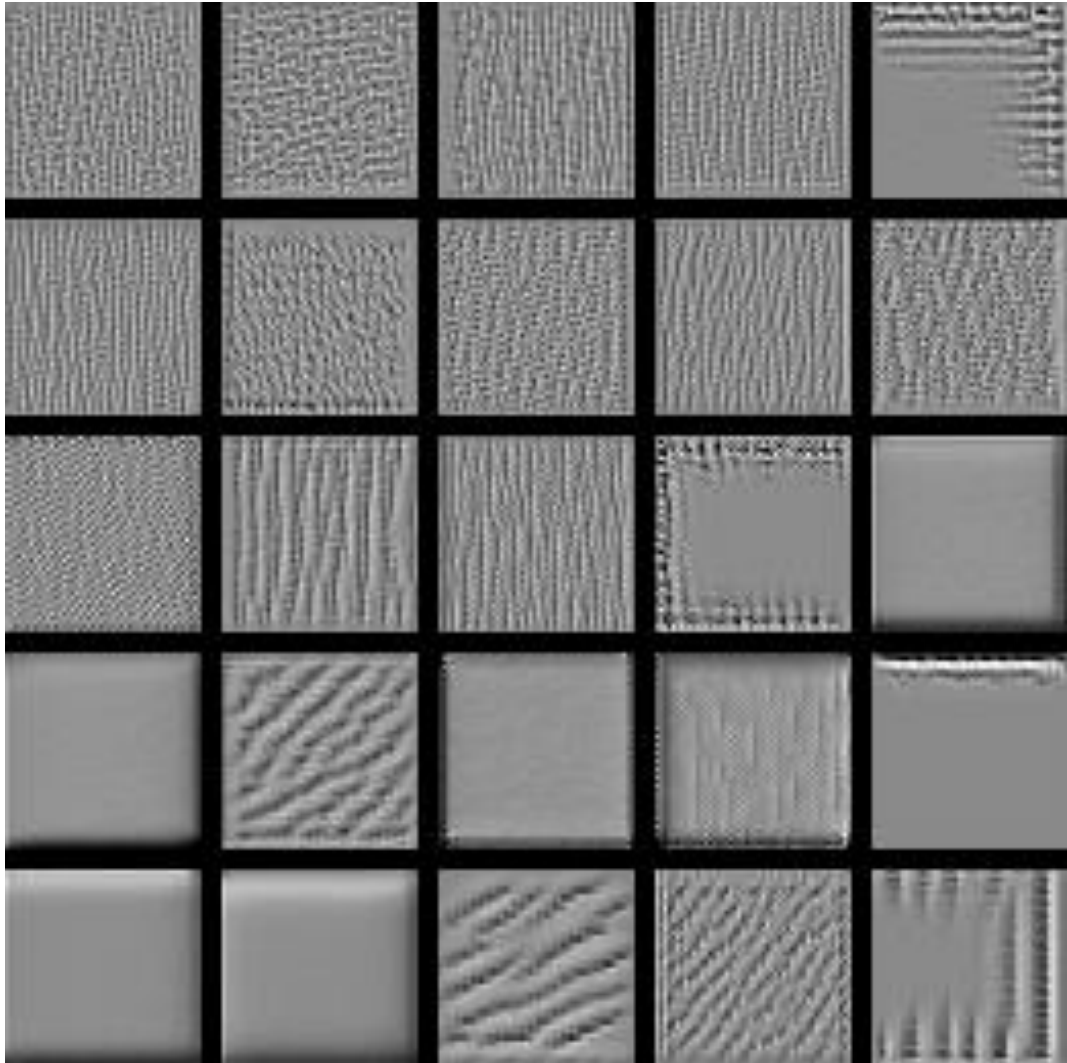
- Saliency Maps



七張圖片中，眼睛的部分都有被 focus，而這也和我們區分一個人的表情時，會觀察他的眼睛這個事實相符合。此外，Happy 的圖片會 focus 在她的上揚的嘴角、Disgust 的圖片會 focus 在因為兩側嘴角下拉而產生的臉頰的皺紋、Surprise 的圖片會 focus 在因為驚訝而張開的嘴巴，也符合人類區分這些表情時的判斷依據。然而，可以明顯發現有些圖片上 focus 的點蠻分散的，這應該也是我的 model 的 accuracy 不高的原因。

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**。

答：



分別找出最可以 **activate conv2d_3** 的 64 個 **filter** 的 64 張圖片之後，再根據每一個 **filter** 被 **maximized** 後的 **output** 排序，篩選出前 25 張圖片。由上圖可以觀察出一些幾何上的 **pattern**，像是一些類似迷宮的圖(第一排左邊四張)、直線(第三排第二、三張)、斜線(第四排第二張、第五排第三章)。然而，也有一些人眼較難以判斷的 **pattern** (第三排第五張、第四排第一、三、五張、第五排第一、二張)。

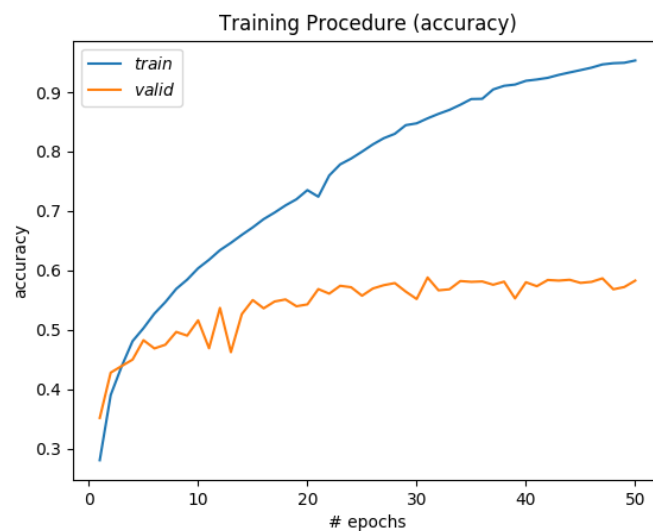
[Bonus] (1%) 從 training data 中移除部份 label，實做 semi-supervised learning

- 模型架構

同第一題

- 訓練過程

將 train.csv 前十分之一的 data 當作 validation set，剩下的作為 labeled data，並將 test.csv 裡的 data 作為 unlabeled data，並且在每一個 epoch 後，將 model 認為屬於某個 class 的機率大於 0.75 的 unlabeled data 加入 labeled data。batch_size = 64, epochs = 50



- 準確率

在第 50 個 epoch 後， validation accuracy 為 0.5827

- 觀察

可以發現 accuracy 與 loss 的圖都跟第一題的 CNN 差不多，推測是因為 unlabeled data 的量與 labeled data 相比太少，不符合 $U \gg R$ 的假設，導致在訓練結果上沒有太大的幫助。

[Bonus] (1%) 在 Problem 5 中，提供了 3 個 hint，可以嘗試實作及觀察 (但也可以不限於 hint 所提到的方向，也可以自己去研究更多關於 CNN 細節的資料)，並說明你做了些什麼？ [完成 1 個: +0.4%, 完成 2 個: +0.7%, 完成 3 個: +1%]

- 沒寫