

學號：B06502149 系級：資工二 姓名：張琦琛

1. (2%) 請說明你實作的 CNN model，其模型架構、訓練參數和準確率為何？並請用與上述 CNN 接近的參數量，實做簡單的 DNN model，同時也說明其模型架構、訓練參數和準確率為何？並說明你觀察到了什麼？

(Collaborators:) 答:

Layer (type)	Output Shape	Param #
conv2d_9 (Conv2D)	(None, 48, 48, 64)	1664
leaky_re_lu_9 (LeakyReLU)	(None, 48, 48, 64)	0
batch_normalization_13 (Batch Normalization)	(None, 48, 48, 64)	256
max_pooling2d_9 (MaxPooling2D)	(None, 24, 24, 64)	0
dropout_13 (Dropout)	(None, 24, 24, 64)	0
conv2d_10 (Conv2D)	(None, 24, 24, 128)	73856
leaky_re_lu_10 (LeakyReLU)	(None, 24, 24, 128)	0
batch_normalization_14 (Batch Normalization)	(None, 24, 24, 128)	512
max_pooling2d_10 (MaxPooling2D)	(None, 12, 12, 128)	0
dropout_14 (Dropout)	(None, 12, 12, 128)	0
conv2d_11 (Conv2D)	(None, 12, 12, 512)	590336
leaky_re_lu_11 (LeakyReLU)	(None, 12, 12, 512)	0
batch_normalization_15 (Batch Normalization)	(None, 12, 12, 512)	2048
max_pooling2d_11 (MaxPooling2D)	(None, 6, 6, 512)	0
dropout_15 (Dropout)	(None, 6, 6, 512)	0
conv2d_12 (Conv2D)	(None, 6, 6, 512)	2359808
leaky_re_lu_12 (LeakyReLU)	(None, 6, 6, 512)	0
batch_normalization_16 (Batch Normalization)	(None, 6, 6, 512)	2048
max_pooling2d_12 (MaxPooling2D)	(None, 3, 3, 512)	0
dropout_16 (Dropout)	(None, 3, 3, 512)	0
flatten_3 (Flatten)	(None, 4608)	0
dense_7 (Dense)	(None, 512)	2359808
batch_normalization_17 (Batch Normalization)	(None, 512)	2048
dropout_17 (Dropout)	(None, 512)	0
dense_8 (Dense)	(None, 512)	262656
batch_normalization_18 (Batch Normalization)	(None, 512)	2048
dropout_18 (Dropout)	(None, 512)	0
dense_9 (Dense)	(None, 7)	3591
Total params: 5,660,679		
Trainable params: 5,656,199		

CNN 模型中我使用了 4 層 Convolution layer，在每一組都用了 LeakyReLU，BatchNormalization，DropOut，Filter 數量是越來越多個，為了避免 OverFitting，我的 DropOut rate 也有慢慢增加，分別是 0.25、0.3、0.35、0.35。最後再將圖壓平，進入 2 層的 Fully-connected-network，DropOut rate 設為

0.5，再利用 softmax 當作 Activation function，產生 7 種分類。在訓練的過程中使用 Adam 當作 optimizer，整個模型大致用了 5700000 個參數。而 Single model 最後在 Kaggle 的到的分數為 Public:0.69239 Private:0.69462。但最後我是使用 Ensemble 的方式，將 4 個準確率差不多的 Model 的預測值取平均，得到的分數為 **Public:0.70938 Private:0.70075**。

Layer (type)	Output Shape	Param #
dense_1 (Dense)	(None, 2304)	5310720
batch_normalization_1 (Batch Normalization)	(None, 2304)	9216
dropout_1 (Dropout)	(None, 2304)	0
dense_2 (Dense)	(None, 256)	590080
batch_normalization_2 (Batch Normalization)	(None, 256)	1024
dropout_2 (Dropout)	(None, 256)	0
dense_3 (Dense)	(None, 128)	32896
dense_4 (Dense)	(None, 7)	903
Total params: 5,944,839		
Trainable params: 5,939,719		

DNN 模型中我用了 3 層 Fully connected network, 每層都有用 BatchNormalization 和 Dropout, 總共參數量約為 600 萬。但是 DNN 在 Kaggle 上的分數為 **Public:0.38200 Private:0.39398**，明顯低了 CNN 不少。

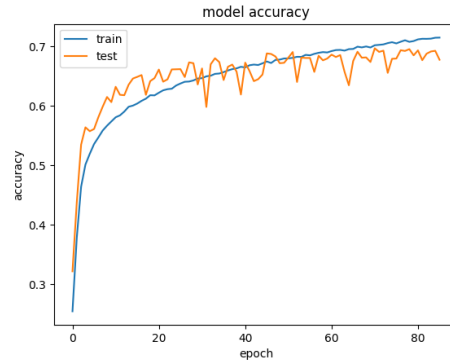
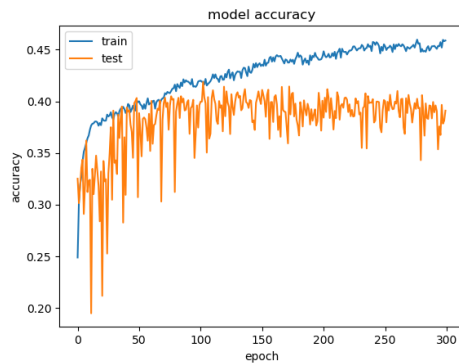
在經過相同的 Epoch，CNN 的準確率已經遠高於 DNN，但是雖然用的參數量差不多，CNN 的訓練速度卻比 DNN 慢許多，可能是因為 CNN 在用卷積層取出圖片特徵時的運算量較大。而我認為我做了 Data augmentation 也是讓 DNN 準確率不高的因素之一，因為 DNN 並沒有像 CNN 能過濾圖片特徵，可能是經過 augmentation 後產生的 noise 影響 DNN 的判斷。

2. (1%) 承上題，請分別畫出這兩個 model 的訓練過程 (i.e., loss/accuracy v.s. epoch)
(Collaborators:)

答:

DNN MODEL

CNN MODEL



3. (1%) 請嘗試 **data normalization**, **data augmentation**,說明實作方法並且說明實行前後對準確率有什麼樣的影響？

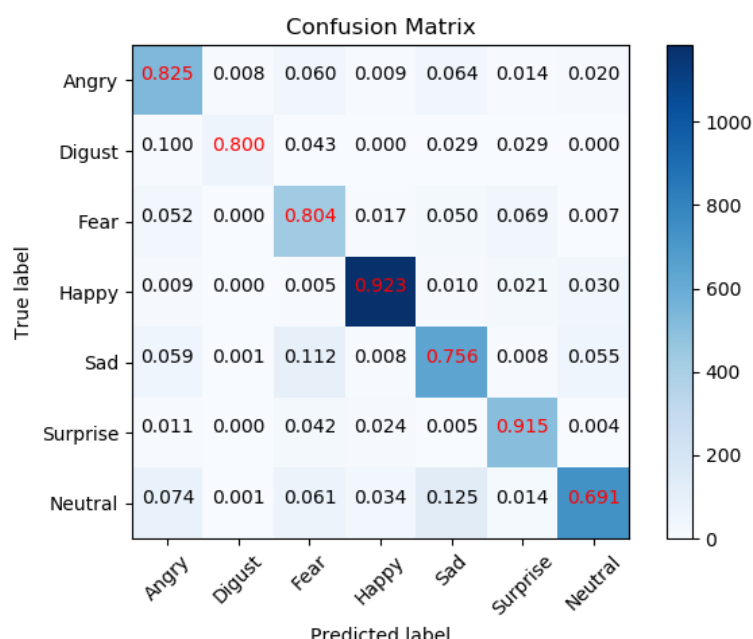
(Collaborators:)

由於圖片中的每個元素皆為 0~255 的數值，因此我將每個元素同除以 255 做 Normalization，如此可以避免某些偏差太大的元素影響結果，在沒有經過標準化前的分數為 **Public:0.68709 Private:0.67372**，經過標準化之後為 **Public:0.69239 Private:0.69462**，可以發現未標準化結果較差，尤其在 **Private** 的分數也有下降。至於 data augmentation 我是使用 `keras.preprocessing.image` 中的 `ImageDataGenerator`，將圖片進行旋轉、縮放、平移、翻轉、推移，產生更多的 data 來訓練 CNN 模型，如此可以增加抗躁性，並可以避免 overfitting。由於使用 `ImageDataGenerator`，需要用 `fit_generator`，而我將 generator 一次產生的資料量恰為 `BatchSize = 128`, `steps_per_epoch` 設為 $5 * (\text{train_data_num}) / \text{BatchSize} = 953$, 因此一次 epoch 實際上訓練的資料量為 $128 * 953 = 121984$ 。在沒有經過 data augmentation 的分數為 **Public:0.68877 Private:0.68514**，在經過 data augmentation 的分數為 **Public:0.69239 Private:0.69462**。

4. (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:)

答:



由圖可看出 Fear、Disgust、Sad 和 Neutral 的準確率較低，尤其從 Sad 所在的 row 和 column 來看，Sad 最容易誤判成 Angry 和 Neutral; Neutral 最容易誤判成 Sad 和 Angry。我認為是因為在 Training Data 中有些照片的情緒不夠強烈，容易讓 Sad 誤判成 Neutral 或者是 Neutral 誤判成 Sad。但我認為對於這樣的情況連人類也很難判斷正確的結果。