學號：R07943107 系級：電子所碩一　姓名：徐晨皓

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: None)

答：

- CNN model

（1）模型架構：

本次作業使用 keras 實作，model 分為兩部分，CNN 與 DNN。圖 1 為本次作業 CNN 模型架構。首先，由**五層 convolution layers** 與**三層 max pooling layers** 交錯組成 CNN 的部分，每層 convolution layer 都有使用 **zero padding** 的技術，讓圖片不會損失原有大小。接著再以**三層 fully connected layers** 組成 DNN 部分，最後再接上 output layer。除了 output layer 的 activation function 為 **softmax**，convolution layer 與 dense layer 都使用 **relu** 作為 activation function。此模型有使用 **dropout** 來避免 overfitting。

（有使用 data augmentation）

（2）訓練參數：

(a) Dropout rate: 0.25
(b) Optimizer: Adam (default)
(c) Epochs: 100
(d) Batch size: 100
(e) **Number of parameters: 7,255,815**

（3）準確率：

Private score: 0.70103
Public score: 0.71217

- DNN model

（1）模型架構：

圖 2 為 DNN 模型架構。使用了共**九層的 dense layers**（含 output layer）。除了 output layer 的 activation function 為 **softmax**，convolution layer 與 dense layer 都使用 **relu** 作為 activation function。此模型有使用 **dropout**。

（2）訓練參數：

(a) Dropout rate: 0.25
(b) Optimizer: Adam (default)
(c) Epochs: 100

(d) Batch size: 100

(e) **Number of parameters: 7,268,615**

（3）準確率：

Private score: 0.38032

Public score: 0.36639

- 觀察結果

  雖然 CNN model 與 DNN model 的參數量接近，但由 public score 與 private score 來看，CNN model 的預測結果比 DNN model 的預測結果相當多。

Layer (type)	Output Shape	Param #
conv2d_1 (Conv2D)	(None, 48, 48, 256)	2560
max_pooling2d_1 (MaxPooling2	(None, 24, 24, 256)	0
batch_normalization_1 (Batch	(None, 24, 24, 256)	1024
conv2d_2 (Conv2D)	(None, 24, 24, 256)	590080
batch_normalization_2 (Batch	(None, 24, 24, 256)	1024
conv2d_3 (Conv2D)	(None, 24, 24, 256)	590080
batch_normalization_3 (Batch	(None, 24, 24, 256)	1024
max_pooling2d_2 (MaxPooling2	(None, 12, 12, 256)	0
dropout_1 (Dropout)	(None, 12, 12, 256)	0
conv2d_4 (Conv2D)	(None, 12, 12, 256)	590080
batch_normalization_4 (Batch	(None, 12, 12, 256)	1024
conv2d_5 (Conv2D)	(None, 12, 12, 256)	590080
batch_normalization_5 (Batch	(None, 12, 12, 256)	1024
max_pooling2d_3 (MaxPooling2	(None, 6, 6, 256)	0
dropout_2 (Dropout)	(None, 6, 6, 256)	0
flatten_1 (Flatten)	(None, 9216)	0
dense_1 (Dense)	(None, 512)	4719104
batch_normalization_6 (Batch	(None, 512)	2048
dropout_3 (Dropout)	(None, 512)	0
dense_2 (Dense)	(None, 256)	131328
batch_normalization_7 (Batch	(None, 256)	1024
dropout_4 (Dropout)	(None, 256)	0
dense_3 (Dense)	(None, 128)	32896
batch_normalization_8 (Batch	(None, 128)	512
dropout_5 (Dropout)	(None, 128)	0
dense_4 (Dense)	(None, 7)	903

Total params: 7,255,815
Trainable params: 7,251,463
Non-trainable params: 4,352

圖 1. CNN model

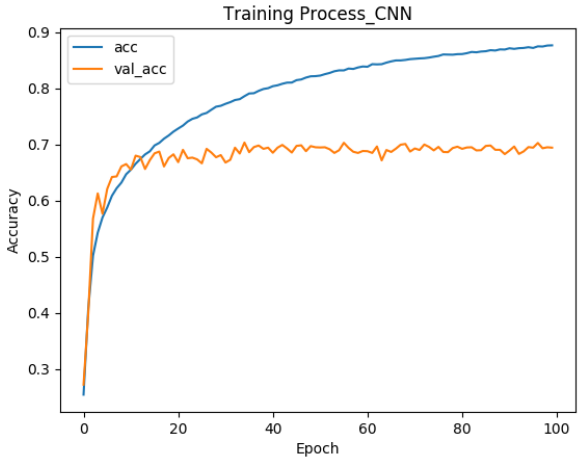
Layer (type)	Output Shape	Param #
dense_1 (Dense)	(None, 1024)	2360320
dense_2 (Dense)	(None, 1024)	1049600
batch_normalization_1 (Batch	(None, 1024)	4096
dropout_1 (Dropout)	(None, 1024)	0
dense_3 (Dense)	(None, 1024)	1049600
batch_normalization_2 (Batch	(None, 1024)	4096
dropout_2 (Dropout)	(None, 1024)	0
dense_4 (Dense)	(None, 1024)	1049600
batch_normalization_3 (Batch	(None, 1024)	4096
dropout_3 (Dropout)	(None, 1024)	0
dense_5 (Dense)	(None, 1024)	1049600
batch_normalization_4 (Batch	(None, 1024)	4096
dropout_4 (Dropout)	(None, 1024)	0
dense_6 (Dense)	(None, 512)	524800
batch_normalization_5 (Batch	(None, 512)	2048
dropout_5 (Dropout)	(None, 512)	0
dense_7 (Dense)	(None, 256)	131328
batch_normalization_6 (Batch	(None, 256)	1024
dropout_6 (Dropout)	(None, 256)	0
dense_8 (Dense)	(None, 128)	32896
batch_normalization_7 (Batch	(None, 128)	512
dropout_7 (Dropout)	(None, 128)	0
dense_9 (Dense)	(None, 7)	903

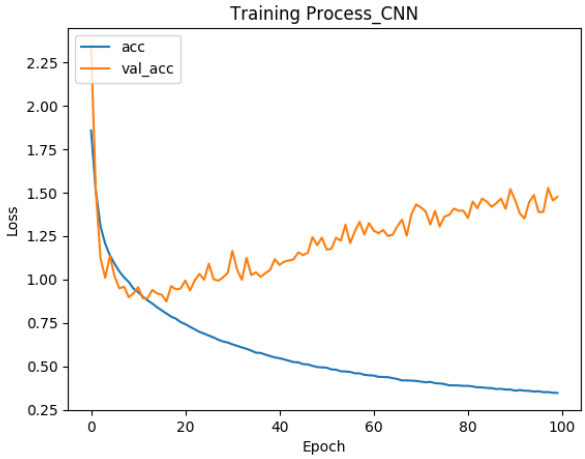
Total params: 7,268,615
Trainable params: 7,258,631
Non-trainable params: 9,984

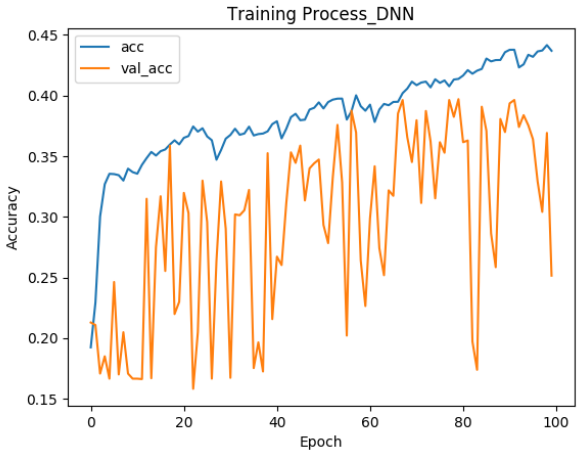
圖 2. DNN model

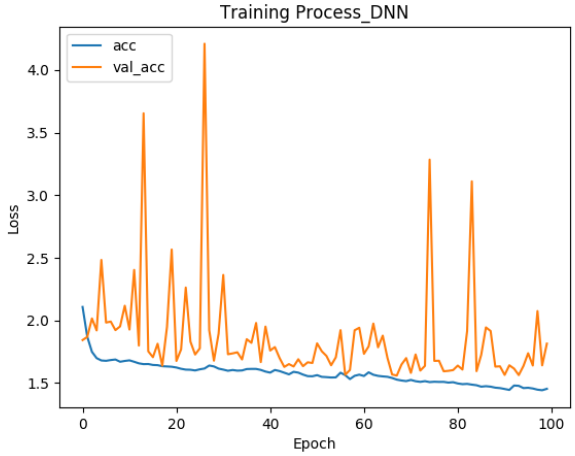
2. (1%) 承上題，請分別畫出這兩個 models 的訓練過程 (i.e., loss/accuracy v.s. epoch)
(Collaborators: None)

答：









（取 10%的 training data 當作 validation set）

由上四圖可以觀察到在訓練過程中：

（1）CNN model 的 training accuracy 可以達到約 0.90，但 DNN model 的 training accuracy 只能達到約 0.45。

（2）CNN model 的 validation accuracy 穩定在約 0.70，而 DNN model 的 validation accuracy 卻在 0.16 和 0.40 之間劇烈震盪。

（3）CNN model 的 training loss 下降至約 0.30，但 DNN model 的 training loss 只能達到約 1.50。

（4）CNN model 的 validation loss 雖然逐漸上升但沒有很大的震盪，可能是為了讓 model 更 fit training data 造成 validation loss 增加。而 DNN model 的 validation loss 卻在 1.6 和 4.0 之間劇烈震盪。

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: None)

答：

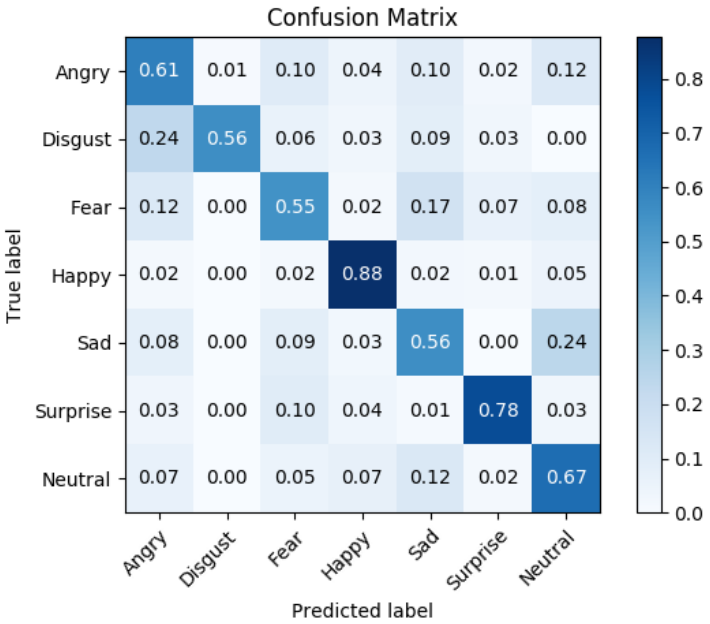
- 實作方法
  (1) Data normalization：由於圖片的 pixel 是 0~255 的灰階，因此直接將每個 pixel 的值除以 255，即可達成 data normalization。
  （2）Data augmentation：此作業使用的是 keras 的 ImageDataGenerator，來增加 training data。使用的參數為
  width_shift_range = 0.2,
  height_shift_range=0.2,
  rotation_flip = 12,
  zoom_range=0.5,
  fill_mode='nearest'。

- 實驗結果
  （1）兩者都不使用：
  Private score: 0.59765
  Public score: 0.60573
  （2）只使用 data normalization：
  Private score: 0.62412
  Public score: 0.63220
  （3）同時做 data normalization 與 data augmentation：
  Private score: 0.69908
  Public score: 0.69657

- 實驗結果分析
  由實驗結果來看，使用 data normalization 與 data augmentation 都能使準確率上升。其中以 data augmentation 的效果最為顯著。

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: None)

答：



根據 confusion matrix 我們可以觀察到以下兩點：

（1）Disgust 的圖片容易被辨識成 angry。由於 disgust 與 angry 的圖片都有皺眉與嘴角向下的特徵，容易造成 model 誤判。

（2）Sad 的圖片容易被辨識成 neutral。因為有些 sad 圖片的臉部肌肉沒有太大的改變，因此 model 會將 sad 辨認成 neutral。但 neutral 卻相對不易被誤判成 sad。