學號：B04505028 系級： 工海四 姓名：林秀銓

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：

CNN:

- 架構

  Total params: 4,746,471, Trainable params: 4,741,415, Non-trainable params: 5,056

```
def build_model():
    filt_size = (3, 3)
    model = Sequential()
    model.add(Convolution2D(32, filt_size, input_shape=(48,48,1), activation='relu', padding='same'))
    model.add(Convolution2D(32, filt_size, activation='relu', padding='same'))
    model.add(BatchNormalization())
    model.add(MaxPooling2D((2,2)))
    model.add(Dropout(0.5))

    model.add(Convolution2D(64, filt_size, activation='relu', padding='same'))
    model.add(Convolution2D(64, filt_size, activation='relu', padding='same'))
    model.add(BatchNormalization())
    model.add(MaxPooling2D((2,2)))
    model.add(Dropout(0.5))

    model.add(Convolution2D(128, filt_size, activation='relu', padding='same'))
    model.add(Convolution2D(128, filt_size, activation='relu', padding='same'))
    model.add(Convolution2D(128, filt_size, activation='relu', padding='same'))
    model.add(BatchNormalization())
    model.add(MaxPooling2D((2,2)))
    model.add(Dropout(0.5))

    model.add(Convolution2D(256, filt_size, activation='relu', padding='same'))
    model.add(Convolution2D(256, filt_size, activation='relu', padding='same'))
    #model.add(Convolution2D(256, filt_size, activation='relu', padding='same'))
    model.add(BatchNormalization())
    model.add(MaxPooling2D((2,2)))
    model.add(Dropout(0.5))


    model.add(Flatten())
    model.add(Dense(1024, activation='relu'))
    model.add(BatchNormalization())
    model.add(Dropout(0.5))
    model.add(Dense(1024, activation='relu'))
    model.add(BatchNormalization())
    model.add(Dropout(0.5))
    model.add(Dense(7))
    model.add(Activation('softmax'))
```

- 參數： batch size = 128, epoch = 100
- public score: 0.66815, private score: 0.65310

DNN:

- 架構

  Total params: 4,995,719, Trainable params: 4,985,095, Non-trainable params: 10,624

```
def build_model():
    model = Sequential()
    model.add(Flatten(input_shape=(48, 48, 1)))

    model.add(Dense(64, activation='relu'))
    model.add(Dense(64, activation='relu'))
    model.add(BatchNormalization())
    model.add(Dropout(0.5))

    model.add(Dense(128, activation='relu'))
    model.add(Dense(128, activation='relu'))
    model.add(BatchNormalization())
    model.add(Dropout(0.5))

    model.add(Dense(512, activation='relu'))
    model.add(Dense(512, activation='relu'))
    model.add(Dense(512, activation='relu'))
    model.add(BatchNormalization())
    model.add(Dropout(0.5))

    model.add(Dense(512, activation='relu'))
    model.add(Dense(512, activation='relu'))
    #model.add(Convolution2D(256, filt_size, activation='relu', padding='same'))
    model.add(BatchNormalization())
    model.add(Dropout(0.5))

    model.add(Dense(1024, activation='relu'))
    model.add(BatchNormalization())
    model.add(Dropout(0.5))

    model.add(Dense(1024, activation='relu'))
    model.add(BatchNormalization())
    model.add(Dropout(0.5))

    model.add(Dense(1024, activation='relu'))
    model.add(BatchNormalization())
    model.add(Dropout(0.5))
    model.add(Dense(1024, activation='relu'))
    model.add(BatchNormalization())
    model.add(Dropout(0.5))
    model.add(Dense(7))
    model.add(Activation('softmax'))
```

- 參數： batch size = 128, epoch = 100
- public score: 0.34104, private score: 0.33797

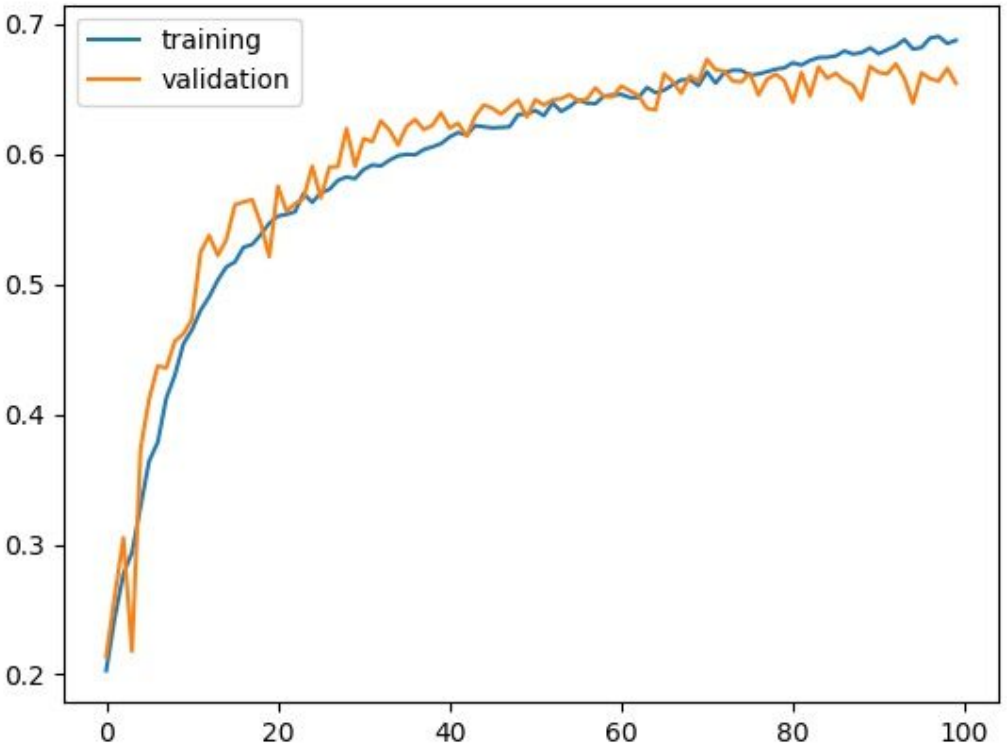
雖然DNN架構未做太多的調整，但是其預測結果非常差，而且由第二題的圖可以發現結果

波動很大且並未達到收斂。

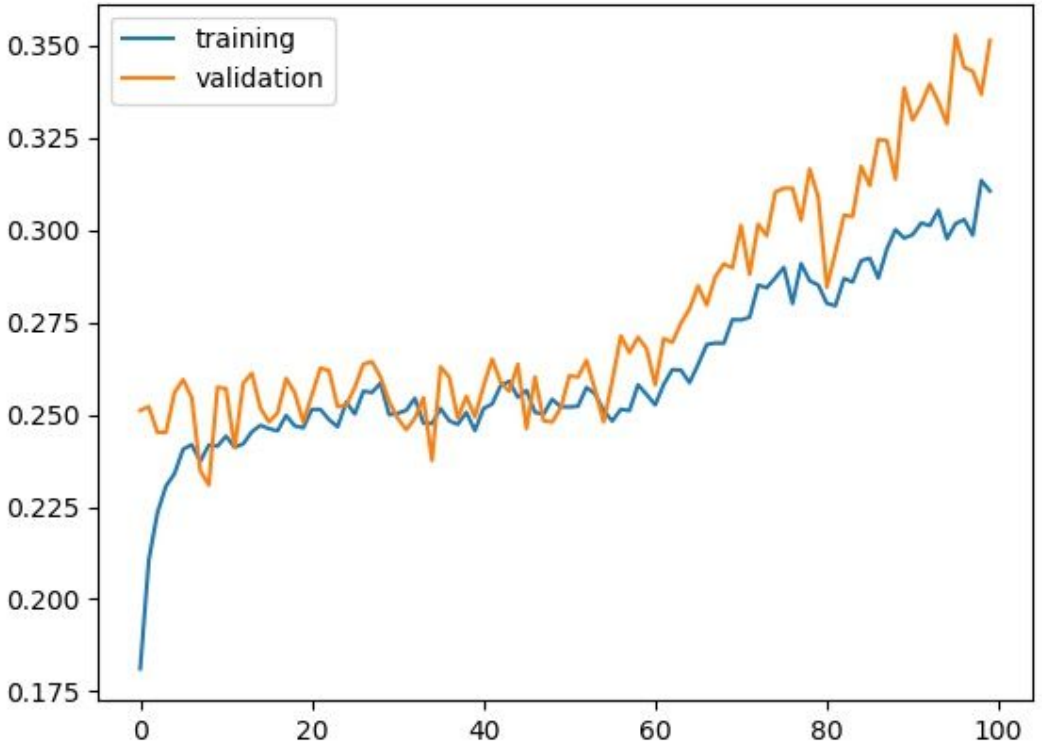
2. (1%) 承上題，請分別畫出這兩個model的訓練過程 (i.e., loss/accuracy v.s. epoch) (Collaborators:)

答：

CNN



DNN



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:)

答：

在本題實驗中，我使用CNN的架構執行40個epoch來做比較， normalization 的實驗中是使用 (x - mean) / (std*225)， 進行normalization後準確率有略為上升一些，且波動也比較小，收斂的速度似乎較快。而 data augmentation 我則是使用 Keras 的 image generator， 並給圖片進行部分的 shift 及 rotate，但準確率反而下降不少，可能是image generator的參數沒有調整好。

(a) data normalization (X) , data augmentation (X)

public score:0.63360 private score:0.61827

(b) data normalization (O) , data augmentation (X)

public score:0.63555 private score:0.62886

(c) data normalization (X) , data augmentation (O)

public score:0.47868 private score:0.47673

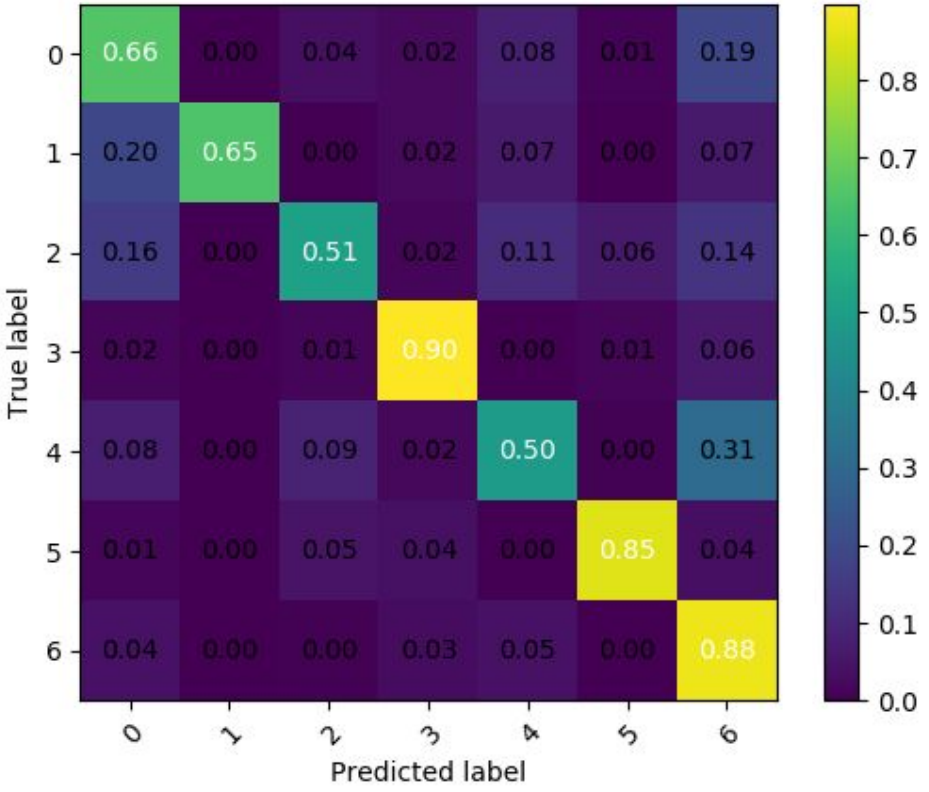
(d) data normalization (O) , data augmentation (O)

同第一題結果

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：

從 confusion matrix 中可以觀察到，最準確的是高興這個分類。而容易混淆的則是難過 vs 中立。


0.66 0.00 0.04 0.02 0.08 0.01 0.19
0.20 0.65 0.00 0.02 0.07 0.00 0.07
0.16 0.00 0.51 0.02 0.11 0.06 0.14
0.02 0.00 0.01 0.90 0.00 0.01 0.06
0.08 0.00 0.09 0.02 0.50 0.00 0.31
0.01 0.00 0.05 0.04 0.00 0.85 0.04
0.04 0.00 0.00 0.03 0.05 0.00 0.88
0
1
2
3
4
5
6
True label
Predicted label
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0