

# Homework3 Report

Professor Pei-Yuan Wu  
EE5184 - Machine Learning

姓名：劉家豪

學號：B04504042

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？

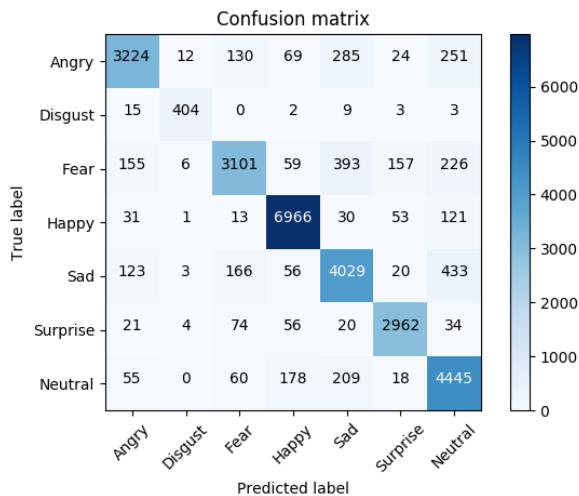
使用四層convolution layer，filter 數量依序為64, 128, 256, 512，kernel size 皆為(3,3)，且皆使用batch\_normalization 和 average\_pooling(2,2)，drop out rate 依序為0.2, 0.25, 0.3, 0.4。flatten 後丟入兩層nn，output size 分別為512, 256, drop out rate 都是0.5。訓練上optimizer 用adam，且使用data generator 為input，能翻轉圖片或是旋轉角度以增加資料量。batch size = 128, epoch = 400，最後保留的model為validation acc 最高的參數。最後 validation acc 跟上傳kaggle的分數基本上差不多。  
result : public score : 0.69657，private score : 0.70855。

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？

使用8層layer，output size 依序 512, 512, 1024, 1024, 1024, 1024, 512, 256 皆使用batch\_normalization 且drop out rate皆為0.5。訓練上跟cnn一樣，optimizer 用adam，且使用data generator 為input。batch size = 128, epoch = 400，最後保留的model為validation acc 最高的參數。validation acc 跟上傳kaggle的分數基本上差不多。  
result : public score : 0.37252，private score : 0.39648。

可以發現沒有使用convolution layer效果差非常多，且我觀察到訓練過程中acc 進步得很慢

3. (1%) 觀察答錯的圖片中，哪些 class 彼此間容易用混？並說明你觀察到了什麼？[繪出 confusion matrix 分析]



由圖片可以知道 (Fear , Sad) , (Angry ,Sad) , (Neutral , Sad) , (Fear , Surprise) , (Fear , Neutral) 這幾組比較容易混淆

觀察到Happy 的準確率最高，達到96%  
而Fear 準確率最低，僅75%

-----Handwritten question-----

4. (1.5%,each 0.5%)CNN time/space complexity:

For a. b. Given a CNN model as

```
model = Sequential()
model.add(Conv2D(filters=6,
                  strides=(3, 3),
                  padding = "valid",
                  kernel_size=(2,2),
                  input_shape=(8,8,5),
                  activation='relu'))
model.add(Conv2D(filters=4,
                  strides=(2, 2),
                  padding = "valid",
                  kernel_size=(2,2),
                  activation='relu'))
```

And for the c. given the parameter as:

kernel size = (k,k);

channel size = c;

input shape of each layer = (n,n);

padding = p;

strides = (s,s);

1. How many parameters are there in each layer(Hint: you may consider whether the number of parameter is related with)  
 $\text{param\_num} = [\text{kernel\_size} * \text{input\_layer\_num} + 1(\text{bias})] * \text{filter\_num}$

Layer A:  $(2*2*5+1)*6 = 126$        $\text{input\_shape} = (3,3,6)$

Layer B:  $(2*2*6+1)*4 = 100$

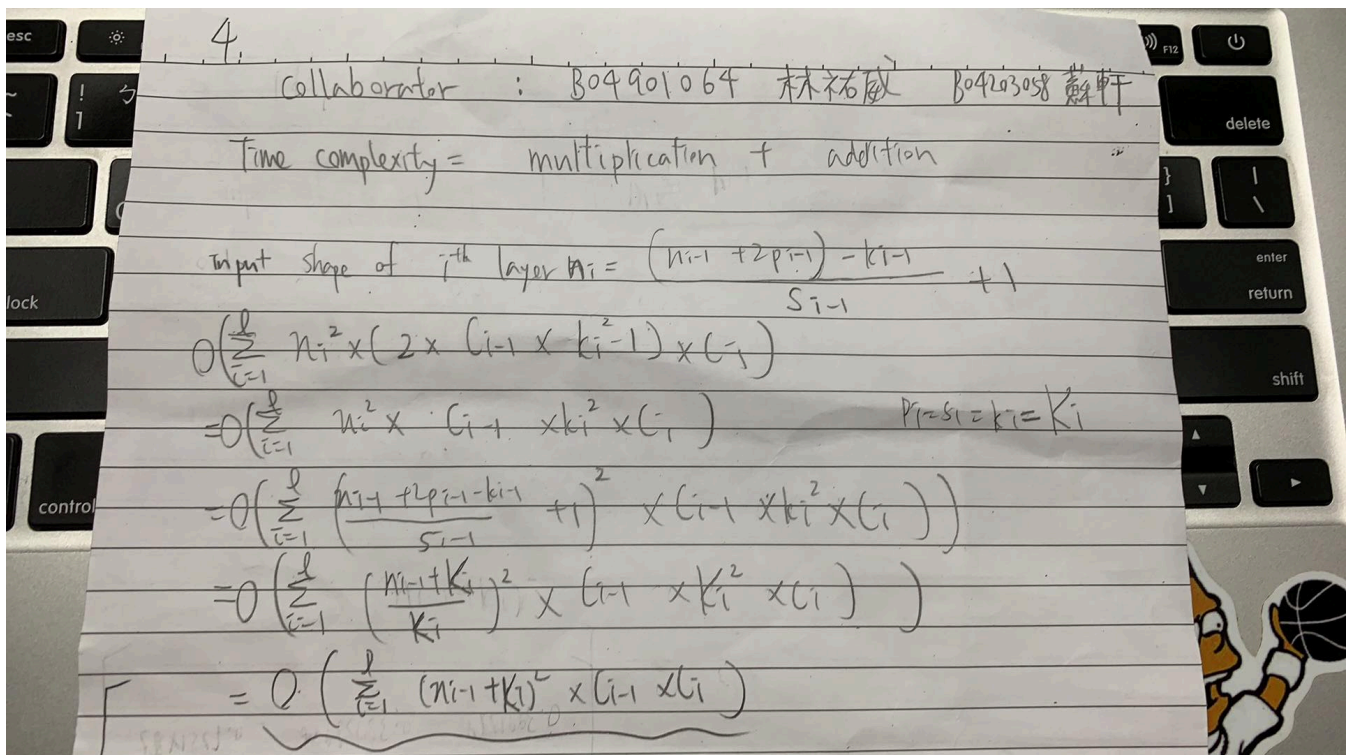
2. How many multiplications/additions are needed for a forward pass(each layer).

multiplication      /      addition

Layer A:  $2*2*5*6*9 = 1080$       /       $(2*2*5-1)*6*9 = 1026$

Layer B:  $2*2*6*4*1 = 96$       /       $(2*2*6-1)*4*1 = 92$

3. What is the time complexity of convolutional neural networks?  
 (note: you must use big-O upper bound, and there are  $l$ (lower case of  $L$ ) layer, you can use  $C_l, C_{l-1}$  as  $l$ th and  $l-1$ th layer)



5. (1.5%,each 0.5%)PCA practice:Problem statement: Given 10 samples in 3D space.(1,2,3),(4,8,5),(3,12,9),(1,8,5),(5,14,2),(7,4,1),(9,8,9),(3,8,1),(11,5,6),(10,11,7)

1. (1) What are the principal axes?

5

1. (1)  $X = \begin{bmatrix} 1 & 4 & 3 & 1 & 5 & 7 & 9 & 3 & 11 & 10 \\ 2 & 8 & 12 & 8 & 14 & 4 & 8 & 8 & 5 & 11 \\ 3 & 5 & 9 & 5 & 2 & 1 & 9 & 1 & 6 & 7 \end{bmatrix}$   $\vec{u} = \begin{bmatrix} 5.4 \\ 8 \\ 4.8 \end{bmatrix}$

$\Sigma = \frac{1}{10} \sum_{i=1}^{10} (\vec{x}_i - \vec{u})(\vec{x}_i - \vec{u})^T = U \Lambda U^T = \begin{bmatrix} 13.3777 & 0.5555 & 3.6444 \\ 0.5555 & 13.5555 & 3.2222 \\ 3.6444 & 3.2222 & 9.0666 \end{bmatrix}$

$\Rightarrow$  decomposition  $\therefore U = \begin{bmatrix} -0.6165947 & -0.67817891 & 0.39985 \\ -0.58881629 & -0.7343903 & 0.33758 \\ -0.52259579 & -0.0228563 & -0.852143 \end{bmatrix}$

$\Lambda = \begin{bmatrix} 16.927 & 0 & 0 \\ 0 & 12.92 & 0 \\ 0 & 0 & 6.08 \end{bmatrix}$

Principle Axes (2-dimension)

$P = \begin{bmatrix} -0.6165947 & -0.67817891 \\ -0.58881629 & -0.7343903 \\ -0.52259579 & -0.0228563 \end{bmatrix}$

2. (2) Compute the principal components for each sample.

(2) Principle Components  $\Rightarrow P^T \hat{X}$

$\hat{X} = \begin{bmatrix} -4.4 & -1.4 & -2.4 & -4.4 & -0.4 & 1.6 & 3.6 & -2.4 & 5.4 \\ -6 & 0 & 4 & 0 & 6 & -4 & 0 & 0 & -3 \\ -1.8 & 0.2 & 4.2 & 0.2 & -2.8 & -3.8 & 4.2 & -3.8 & 6.2 \end{bmatrix}$

answer =

$\begin{bmatrix} 7.18658682 & 1.37323947 \\ 0.75871342 & -0.94399334 \\ -3.07034019 & -4.45059025 \\ 2.60849751 & -2.97853006 \\ -1.82299166 & -4.75401212 \\ 3.35457763 & 3.91896138 \\ -4.41464321 & 2.55604371 \\ 3.46569126 & -1.73131477 \\ -2.31359638 & 6.03371503 \\ -5.75249521 & 0.97648096 \end{bmatrix}$

3. (3) Reconstruction error if reduced to 2D. (Calculate the L2-norm)

(3) Reconstruction =  $P P^T (\hat{X})$  再平移  $\begin{bmatrix} 5.4 \\ 8 \\ 4.8 \end{bmatrix}$

$\begin{bmatrix} 1.90009072 & 2.75992709 & 1.08178971 \\ 4.29198496 & 8.24651657 & 4.37774211 \\ 4.27485905 & 13.07633588 & 6.28310968 \\ 1.77163801 & 8.65147726 & 3.35553912 \\ 3.29997625 & 12.56470677 & 5.62297154 \\ 5.98934216 & 3.14672348 & 3.1538432 \\ 9.85550052 & 8.72228056 & 7.17681721 \\ 2.08893199 & 7.23080501 & 2.94160433 \\ 10.91848951 & 4.93118246 & 6.17370944 \\ 9.60918683 & 10.6700449 & 7.83287366 \end{bmatrix}$

l2-norm =

$\begin{bmatrix} 2.25104047, & 0.73022599, & 3.18825971, & 1.92976683, & 4.25088226, \\ 2.52755813, & 2.13939244, & 2.27848634, & 0.20383817, & 0.97738426 \end{bmatrix}$

average = 2.04