

Homework3 Report

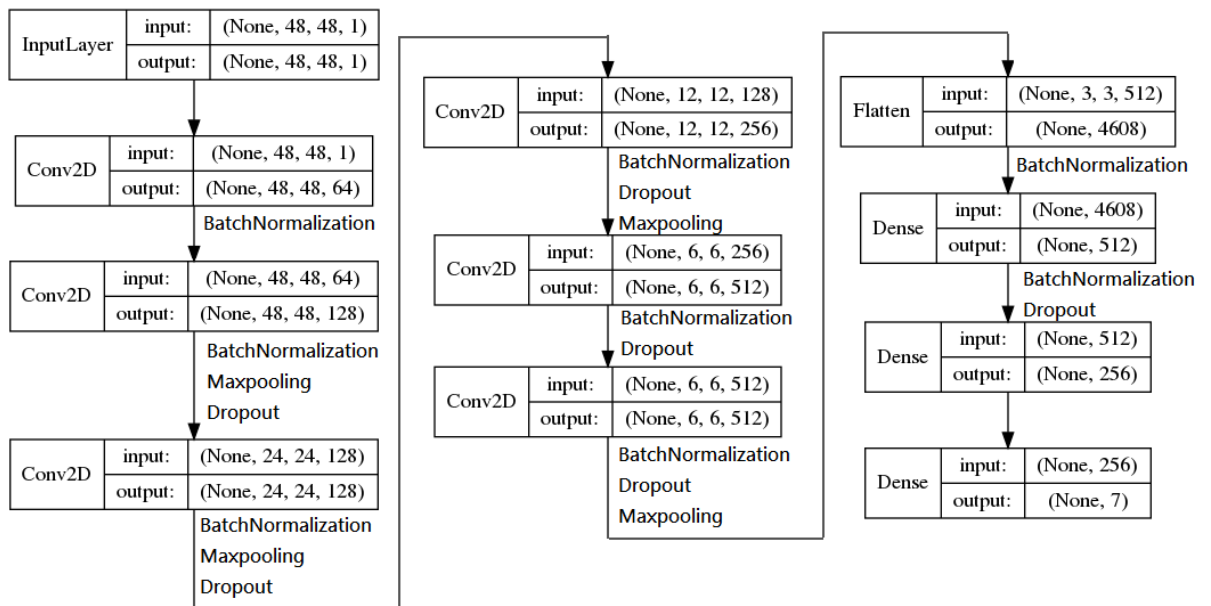
Professor Pei-Yuan Wu
EE5184 - Machine Learning

姓名：洪正皇

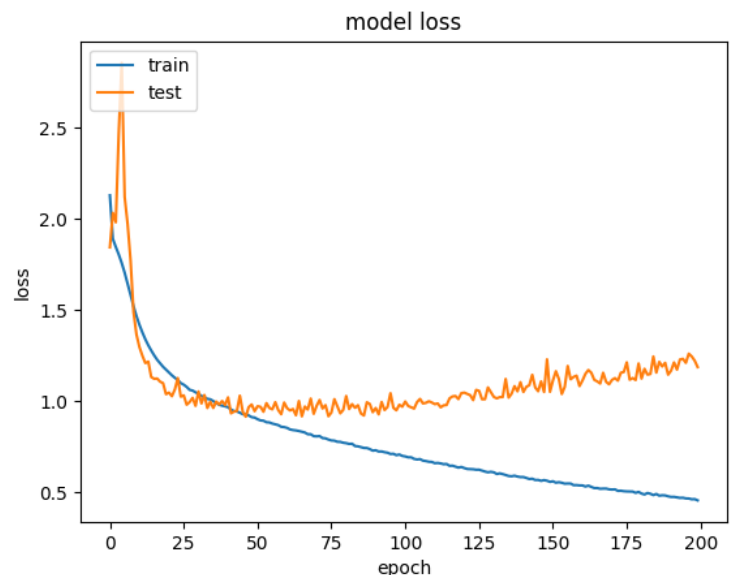
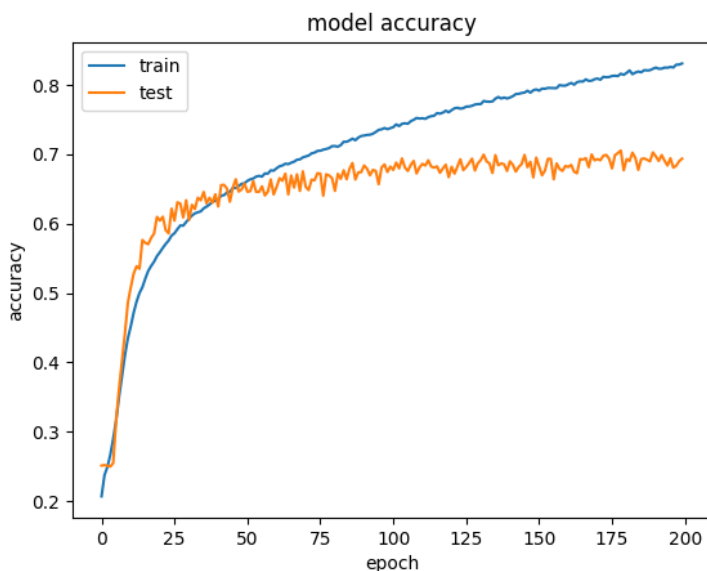
學號：R07922050

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？

在每層 convolution2D 中加上一些 BatchNormalization、Dropout 及 Maxpooling，第一層 convolution2D 的部分我擔心離 output 太遠，參數比較難 train 到，因此就沒有加上 dropout，而 Flatten 之前也疊了兩層 convolution layer 才做最後一次 maxpooling，為了使 convolution layer 總共為偶數層，Flatten 後也刻意少掉一些 dropout。



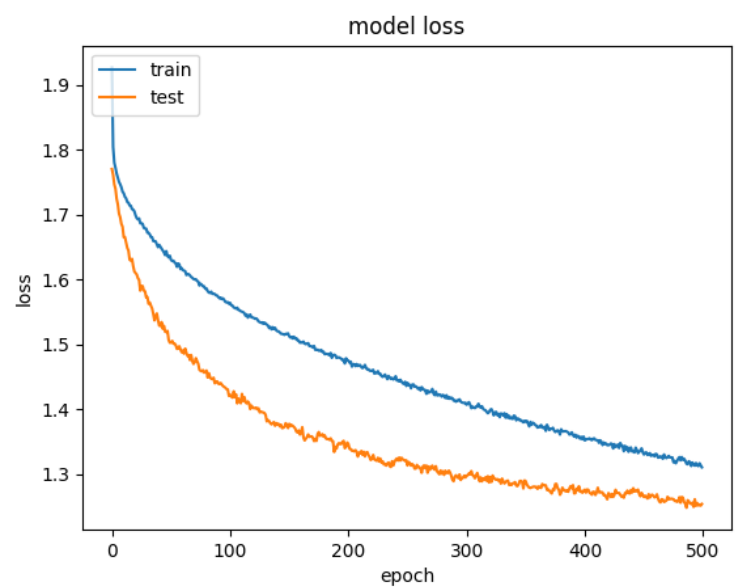
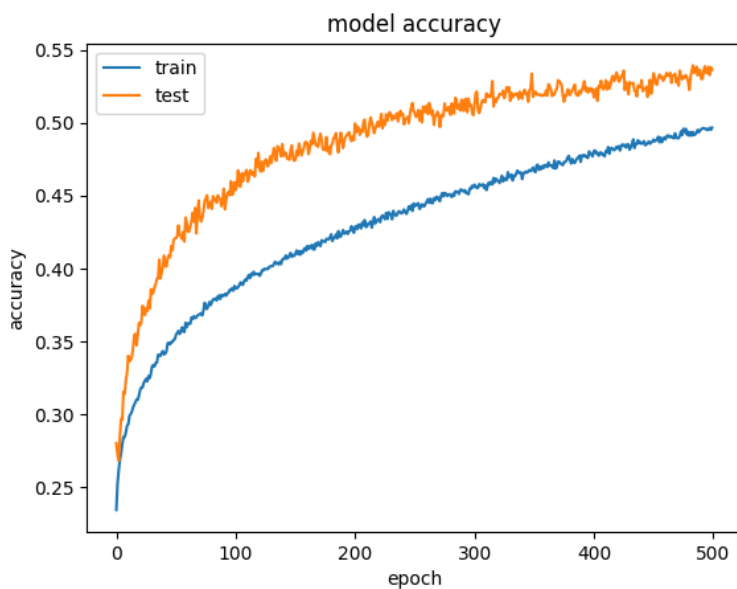
原本我只有使用四層 convolution layers，val_acc 始終卡在 0.65~0.68 之間，以為是我的 model 不夠複雜，因此多加了兩層，並增加 epoch 數量，結果效能還是沒有多大的提升，但可以看到 model 在 validation data 上的準確率浮動滿大的，我想運氣不錯的話可以取到滿意的結果，我 scoreboard 的名次也前進了 20 名次。



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？

在我的 DNN model 中，比 CNN 多了 5~8% 左右的參數，原本期待如果資料量夠多的話，說不定也能得到還不錯的結果，但世事不盡如人意，即便跑了 500 個 epochs，準確率也只有在 0.5 上下，CNN 的一些特性果然能更準確的提取出所需要的特徵，而不是參數多的 NN 就會贏。

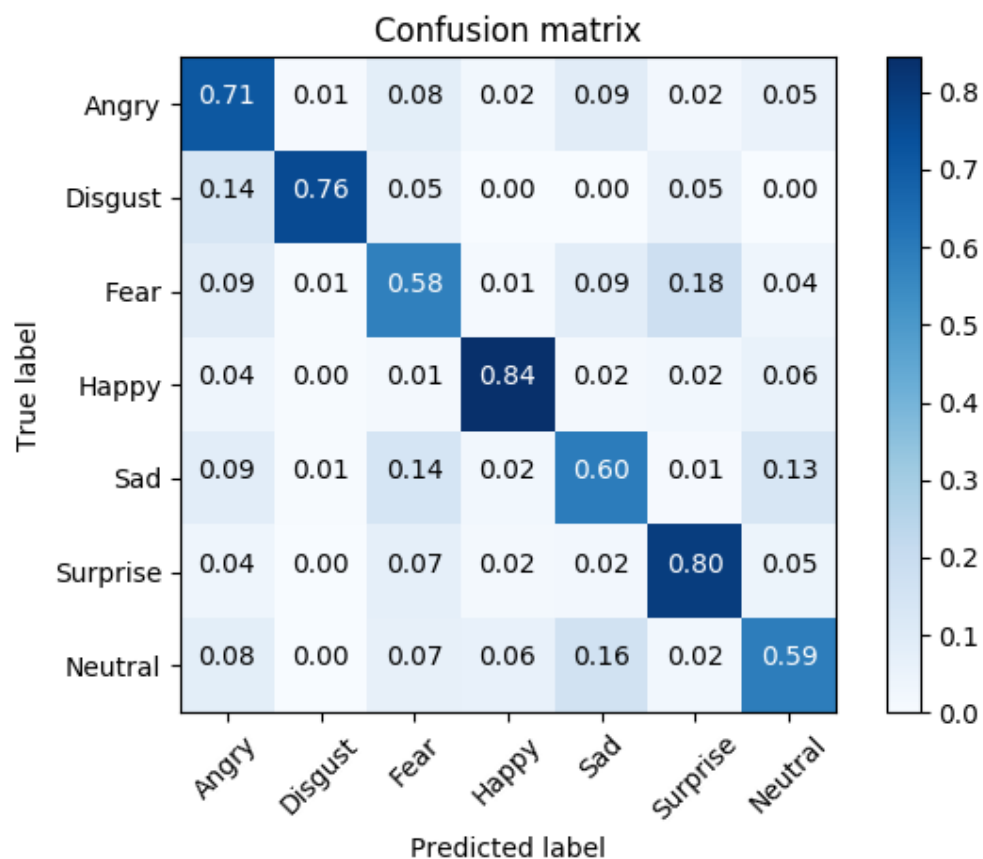
另外一個比較有趣的現象是，validation data 的 training loss 與 accuracy 一直比 training data 的表現較好，我將 epochs 數量提高後仍然是這樣的結果，如果能 train 起來一定也能有不錯的結果。



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]

在我的模型中，Sad 與 Neutral 是較為容易彼此搞混的，我想現代人大多偏向於不將悲傷的情緒表達出來，因此 Sad 與 Neutral 這兩種情況中間有滿大的模糊地帶，我想這是非常難以避免的。

而一個有趣的現象是，Disgust 容易被預測為 Angry，而 Angry 反而較少被預測為 Disgust，這兩者都是負面的情緒，彼此間本來就有些關聯。Disgust 中可能有滿多 Angry 有的特徵，而 Angry 中有較多 Disgust 所沒有的特徵，因此 Angry 不容易被誤判成 Disgust。



4. a. Layer A: $(2 \times 2 \times 5 + 1) \times 6 = 126$
 Layer B: $(2 \times 2 \times 6 + 1) \times 4 = 100$

b. Layer A $\begin{cases} \text{加法: } (2 \times 2 \times 5 - 1) \times (3 \times 3) \times 6 = 1026 \\ \text{乘法: } (2 \times 2 \times 5) \times (3 \times 3) \times 6 = 1080 \end{cases}$

Layer B $\begin{cases} \text{加法: } (2 \times 2 \times 6 - 1) \times (1 \times 1) \times 4 = 92 \\ \text{乘法: } (2 \times 2 \times 6) \times (1 \times 1) \times 4 = 96 \end{cases}$

c. $O \left(\sum_{k=2}^2 (k^2 \times C_{k1}) \times \left(\left\lfloor \frac{n_k + k_k + p_k}{s_k} \right\rfloor + 1 \right)^2 \times C_k \right)$

5. b. by Matlab
 $\text{COV}(X) = \begin{bmatrix} 13.3778 & 0.5556 & 3.6444 \\ 0.5556 & 13.5556 & 3.2222 \\ 3.6444 & 3.2222 & 9.0667 \end{bmatrix}$

\Rightarrow principle components = eigenvectors = $\begin{cases} (-0.6166, -0.5888, -0.5226) \\ (-0.6782, 0.7344, -0.0273) \\ (-0.3999, -0.3376, 0.8521) \end{cases}$

c. with principle components $(-0.6166, -0.5888, -0.5226)$

The projection samples are: $(2.0730, 1.9796, 1.7570)$

$(6.0364, 5.7644, 5.1162), (8.3974, 8.0191, 7.1172)$

$(4.8958, 4.6753, 4.1495), (7.6283, 7.2846, 6.4653), (4.4358, 4.2360, 3.7596)$

$(9.2263, 8.8106, 7.8197), (4.3673, 4.1705, 3.7015), (7.9308, 7.5735, 6.6721)$

$(10.0512, 9.5984, 8.5189)$

The L_2 -norm are: $1.6422, 3.0262, 6.9659, 5.1918, 8.4820, 3.7744,$
 $1.4496, 4.8818, 4.0699, 2.0674$

Total L_2 -norm sum: 41.5512

a. 一組 variance 由高至低的 axis, 且任兩個 principle axes 間的 Covariance 皆為零