ML HW3 Report

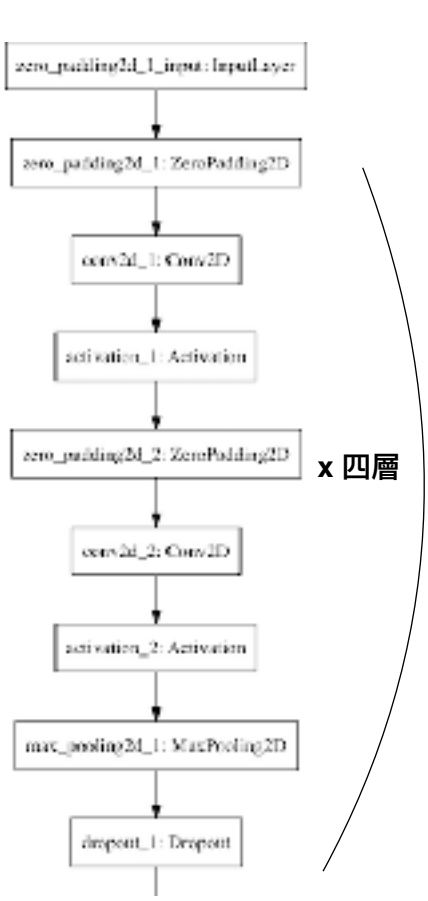
B02901093 吳岳

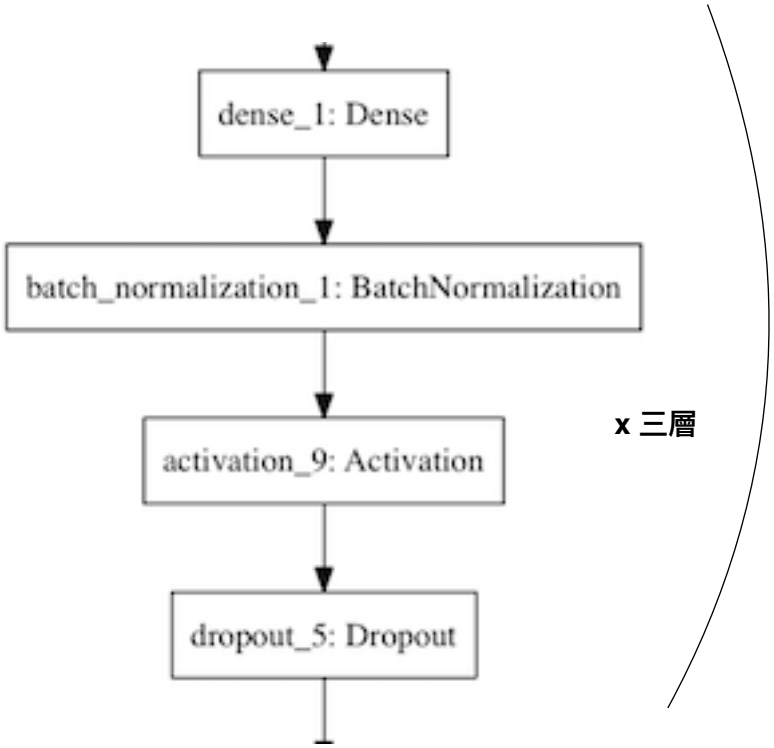
1. (1%) 請說明你實作的 CNN model，其模型架構、訓練過程和準確率為何？

模型架構

如左下圖所示，前段CNN的架構為(Zero Padding + 兩層convolution + maxPooling + Dropout)*四層，再加上後面的flatten。

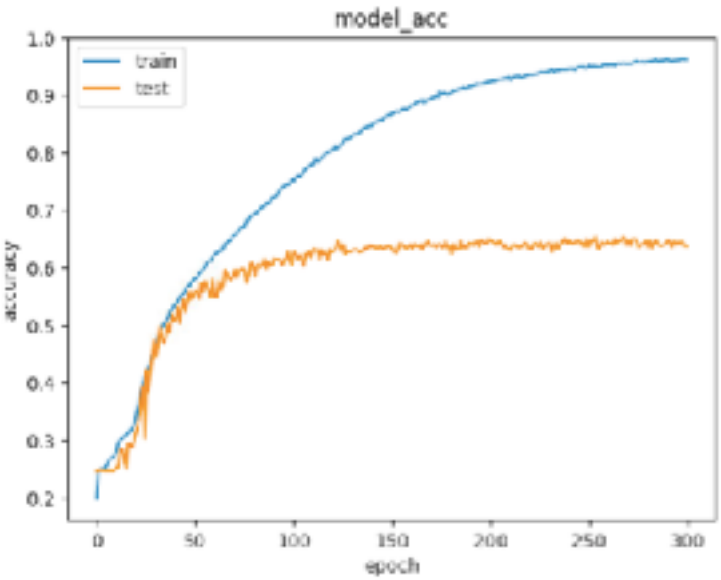
後段的Dense是三層的架構，如右下圖所示，這邊我有用Batch Normalization去調整每層NN進入Activation Function前的大小，再加上Dropout，Activation是用SGD，learning rate為 0.01，並加入 momentum 為 0.9，加入 weight decay。





訓練過程

我將原90%的training set當成訓練的training，剩下10%的原training set當成testing set，每次epoch我都會判斷新training set跟testing set的準確率。從右圖可以發現在趨近於第150 epoch時testing的準確率近乎converge了。而training set的準確率還持續增加。



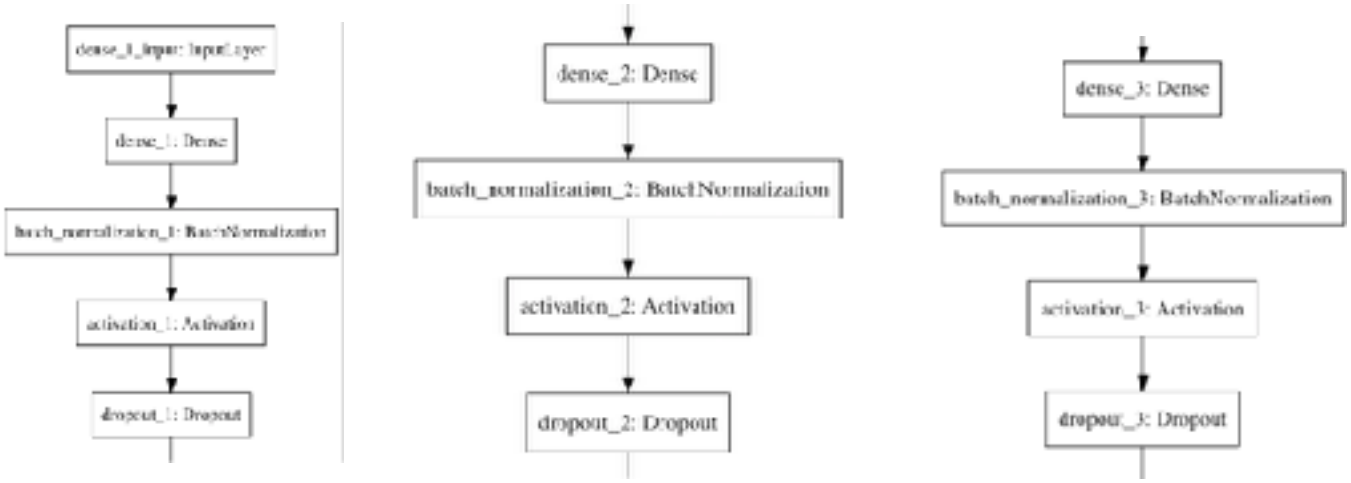
準確率

最後的準確率為65.17%，在kaggle上面public的準確率約莫為67.2%。

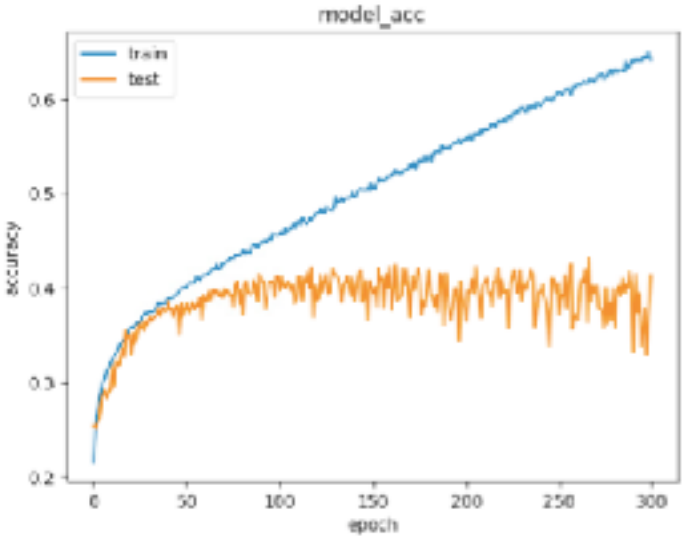
2. (1%) 承上題，請用與上述 CNN 接近的參數量，實做簡單的 DNN model。其模型架構、訓練過程和準確率為何？試與上題結果做比較，並說明你觀察到了什麼？

模型架構

扣掉CNN之後，剩下的是三層的Dense架構，這邊我還是有用Batch Normalization去調整每層NN進入Activation Function前的大小，再加上Dropout，Activation是用SGD，參數跟上題都一樣。



訓練過程



training set跟testing set的分配如第一題。training set尚未converge，若用更多的epoch的話準確率可以更高，然而，testing set的準確率在50個epoch左右就無法再上升，在38%左右擺動。

準確率

DNN的做法準確率training set約莫為64%，testing set為40.5%。

觀察

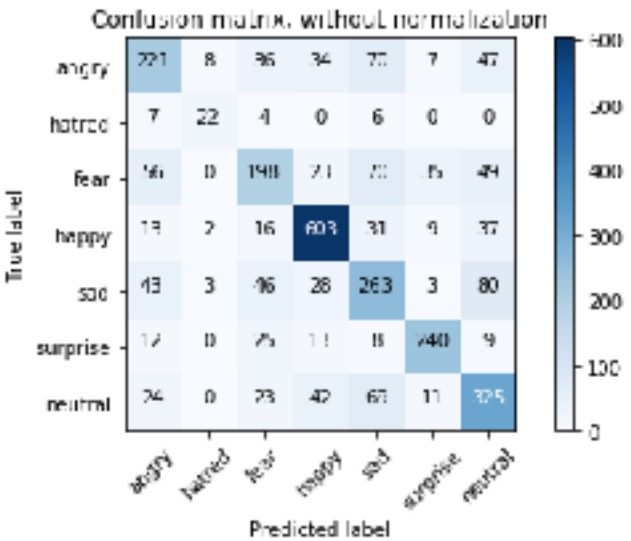
除了準確率明顯降低之外，testing set converge的時間也明顯提早了，再者，testing set converge之後擺盪的幅度也比CNN還高，CNN在2%以內，DNN介於4-5%之間。

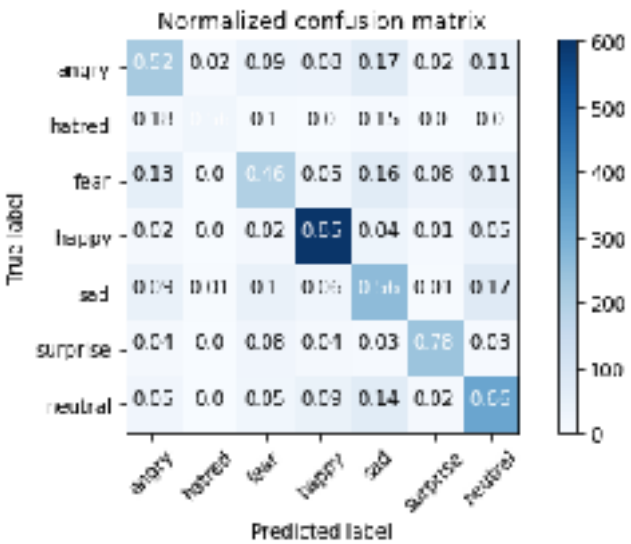
當然，DNN的做法明顯比CNN快，原本每個epoch CNN 要跑18s，DNN 2秒內就可以跑完。

3. (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

A:左圖是取1/10個training data對應7種表情的數量，右圖則是比例，看對角線的validation的準確率由上到下為0.52, 0.66, 0.46, 0.85, 0.56, 0.78, 0.66。

根據右圖，容易搞混的表情如下表：

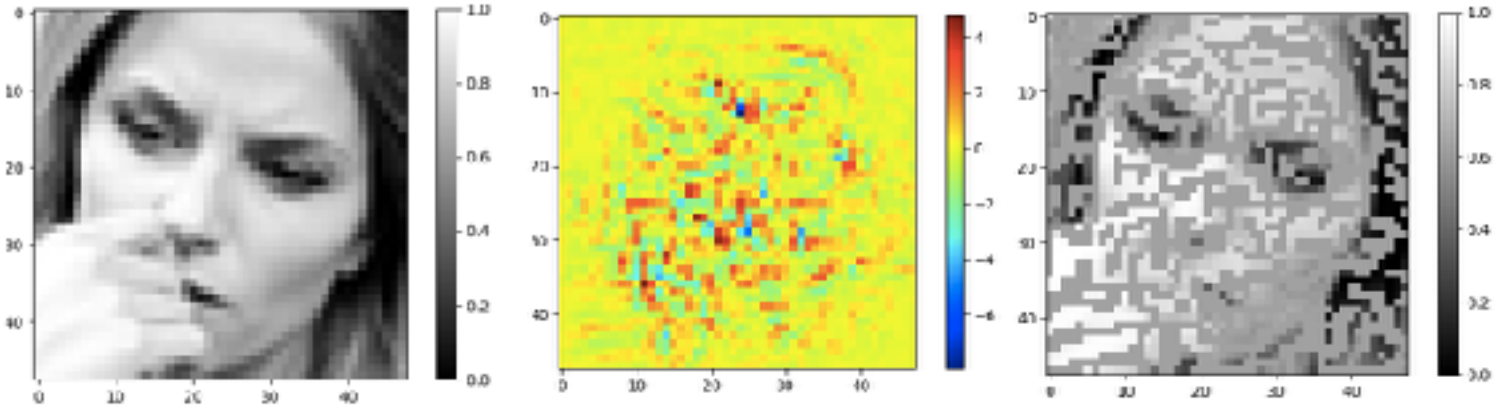




正確表情	錯誤預測表情	錯誤率	推測錯誤率較高的原因
angry	sad	0.17	angry時皺眉、憋嘴的特徵可能誤判成sad
hatred	angry	0.18	hatred誇張的臉部表情(尤其在眼睛周圍)特別容易跟angry搞混
sad	nuetral	0.17	部分的憂傷來自於眼神而非面部表情，此特徵可能無法訓練出來
neutral	sad	0.14	同上

4. **(1%) 從(1)(2)可以發現，使用 CNN 的確有些好處，試繪出其 saliency maps，觀察模型在做 classification 時，是 focus 在圖片的哪些部份？**

原圖(Fear)　　Silence Map　　After Heat



以此系列圖形來說，silency map在眉毛、鼻子、嘴巴、跟拇指的位置數值較高(眼睛比較不明顯)。而在mask heat掉小部分之後，特徵較突出的點則為眼睛、眉毛、鼻子、嘴巴、跟拇指。

上述特徵除了手之外都符合我最初對恐懼表情特徵的猜測，而我推測程式之所以會將「手」作為恐懼的特徵，是因為緊張或害怕時，手可能會放在嘴巴附近(咬手指、手磨蹭嘴唇)。

5. **(1%) 承(1)(2)，利用上課所提到的 gradient ascent 方法，觀察特定層的filter最容易被哪種圖片 activate。**

下面第一張圖是id=17的圖在convolution layer第二層的filter輸出，第二張圖則是對應的filter 數值，我們可以發現隨著數值的變大變小，會影響此圖像filter之後的activation。視覺上最明顯的就是人像的表情特徵。

以下用紅色框框標示出數值最高跟數值最低的兩個filter跟對應的人像，可以發現，最高值的輪廓比最低值明顯很多，害怕的特徵(手部動作、嘴巴、眼睛)亦有很大的差異。

[Bonus] (1%) 從 training data 中移除部份 label，實做 semi-supervised learning

A:利用第一題準確率最高對應參數來建立supervise learning CNN 架構，我分別嘗試過 5000, 8000, 10000 筆 data當作label來建model，將剩餘的training data當作unlabelled data，預測 unlabeled data 屬於每個 class 個別的機率，如果一筆 unlabeled data 屬於某個 class 的機率 大於0.9，就推斷此 model 預測是正確的，並將那個data加入 labeled data set， 得到新的 labeled training model， 並利用新的model繼續做訓練，又可以得到新的 mode，不斷重複此項動作直到幾乎將所有的 unlabeled data 加入 label training set。

而每次每個新的 model 都會 train 30 個 epoch 數，得到最後的 training model，最後使用此 model 來預測 testing set 的結果，正確率如下表：

# of label data	Accuracy
5000	54.6%
8000	56.9%
10000	60.2%

由上表可以發現，正確率不會因semi-supervised而提升，畢竟將原本既有label的data當成unlabelled是自斷手腳的行為。於是，我也另外做的一個實驗，training data維持labeled，

另外7000多個testing data則當成unlabelled，作完之後kaggle的準確率破了我原本supervised的67.177%，到達67.68%，(後來時間不夠就沒有再往上衝了)

[Bonus] (1%) 在Problem 5 中，提供了3個 hint，可以嘗試實作及觀察 (但也可以不限於 hint 所提到的方向，也可以自己去研究更多關於 CNN 細節的資料)，並說明你做了些什麼？ [完成1個: +0.4%, 完成2個: +0.7%, 完成3個: +1%]

[hint2]

這裡我拿原本的filter跟其他的fear class(id = 21, 33)去比對，結果跟第五題契合，在紅色框框數值最高的地方都有極高的activation