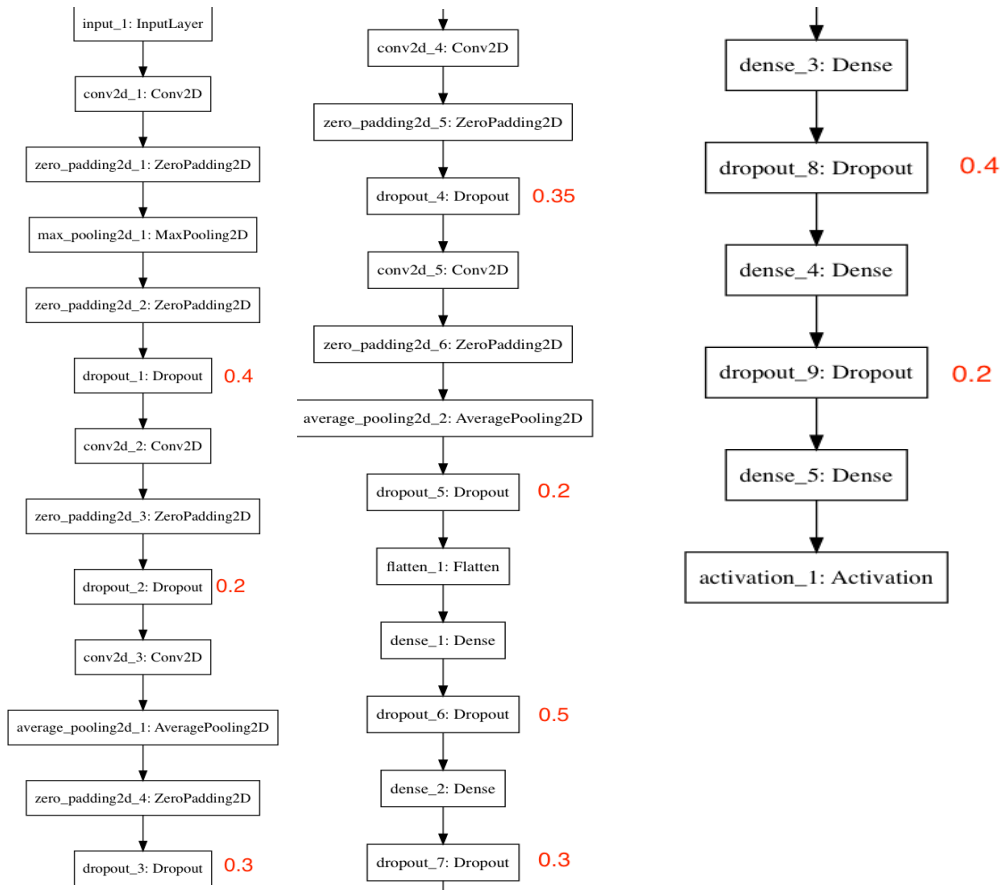


學號：B04902092 系級：資工三 姓名：張均銘

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？
(Collaborators: b04902082 鍾偉傑)



模型非常長，所以把他切成三段貼上來，由左而右，簡單而言就是 5 層 Conv2d layer 加上 4 層的 Dense layer, 每層都有做一些 Dropout，圖上紅字就是 Dropout 的比例。

Total Parameter : 1232519

Training:

Conv2D activation : relu, Activation : Softmax, Optimizer : Adam,

準確率:

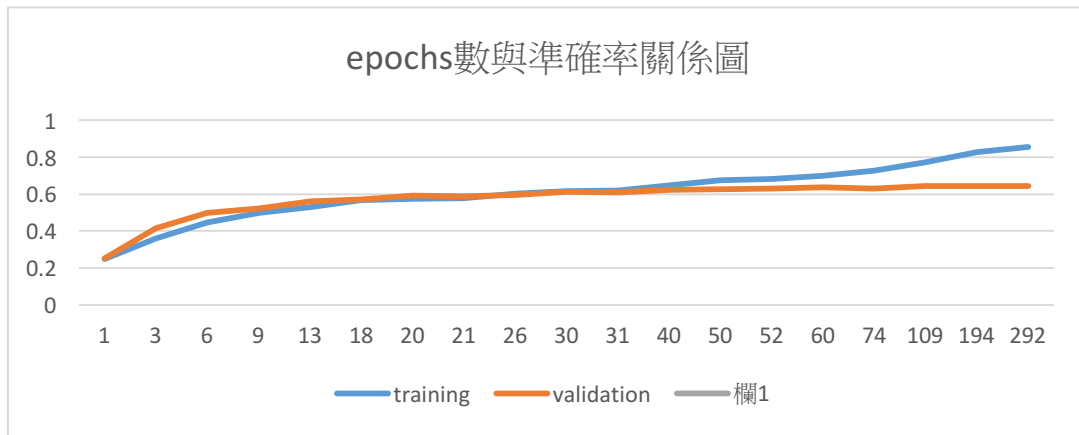
下圖為 epochs=300 時使用 callback:checkpoint 以 val_acc 為基準來記錄在不同 epochs 時的 model 的 training_acc 跟 valid_acc，下表格主要顯示有較大變化的 epoch 而已，而不是畫出所有 epochs。

Kaggle 準確率: private : 0.64140 public : 0.63722

發現: 雖然在 Conv2D 的階段 Dropout 非常奇怪，但卻很神奇的會提高準確率。

而且助教的 code 其實再加一兩層 Dense 會有更好的效果，但是三層

Dense 後就沒有明顯進步，epochs 也是，跑 epochs=1000 時可能到大概 400 之內就沒有再產生新的更好的 model，應該是卡在 local minimum。



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？

答：

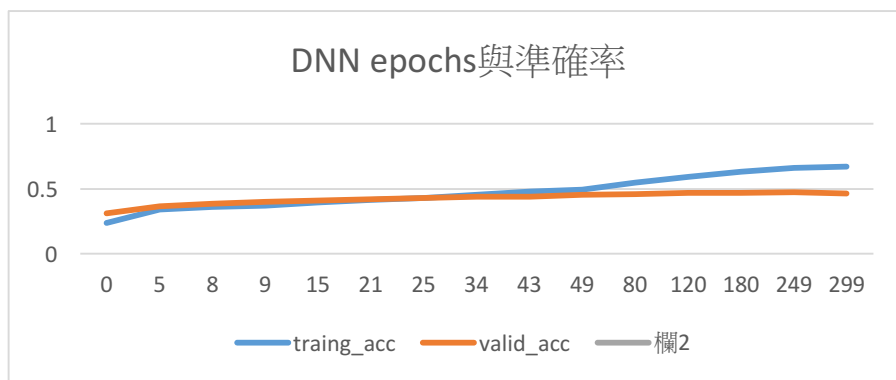
Total Parameter : 1279751

Kaggle 分數: public:0.47394 private:0.48174

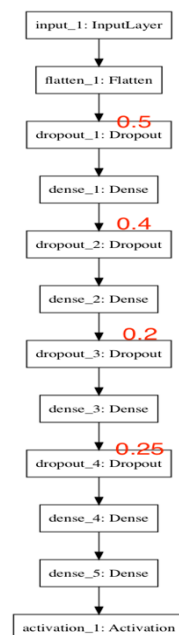
Training:

4 層 Dense Layer，一樣每層都有做 Dropout，左圖紅字就是 Dropout 的 rate，Activation,optimizer 都同第一題。

下圖折線圖是拿 model train epoch300 次，並用 Checkpoint 來記錄有哪些較好的 model 並作紀錄，明顯看到，valid_acc 在約 25 以後就沒有再上升。

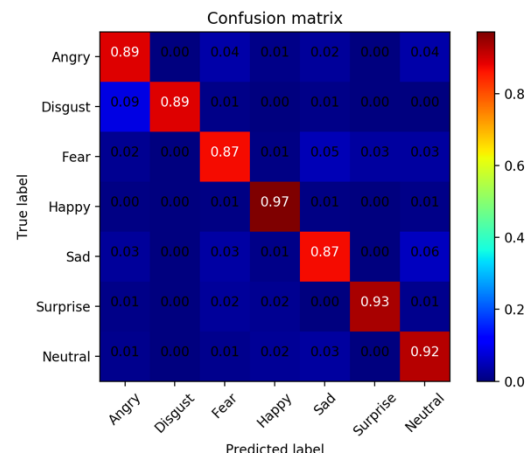


與 CNN 差別:



- 1) 兩者初始差不多，但 CNN 不但 acc 上升速度較快、但兩者大約都是 epochs = 50 多時 valid_acc 停止上升。
- 2) cnn 的準確度明顯比 dnn 還要高，原因：因為 CNN 是把原圖切成一小塊一小塊的圖做判斷，但 dnn 可以說是每個 pixel 來做判斷，很明顯地就會不准很多，例如每個圖都會有空白的地方，那這樣 conv2d 比較不會受此影響，但 dnn 受到的影響就會比較大
- 3) dnn 速度遠比 cnn 快，因為他只要做一次 dot 就好，cnn 要做多次矩陣相乘

發現：另外我試了一個 dnn model 只有 3 個 layer 的(比這個還少一個 layer)，總參數是 1183495, dropout 少 drop 一個 0.4(因為少一個 layer)，其他維持依樣，發現準確度比原本 DNN model 還要高一些, train_acc 到 0.68, valid_acc 到 0.48，但 kaggle 上成績差不多，看起來是差不多爛，不過 3 個 layer 的感覺還有機會 train 下去



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]

validation set 切法：

tmp = np.random.permutation(28709)

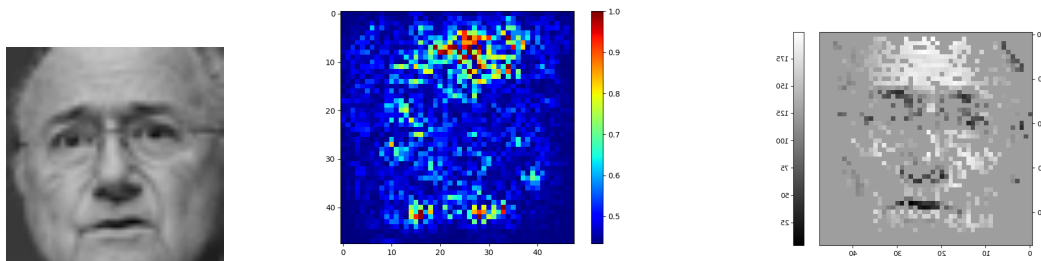
tmp = tmp[:5740]的方式來切 validate

發現：

Angry 跟 disgusting 因為有種相依關係，例如 disgusting 的人就有種會討厭這種東西的感覺，所以把 Disgust 誤認為 Angry 的比例有點高，意外的是我以為 Disgust 和 Fear 應該會有點關係(厭惡通常會帶有害怕的情緒，反而幾乎沒有誤判，估計是 testdata 的 Fear 的表情都是那種看恐怖片或者看到鬼的表情，不是那種看到噁心東西的表情)。而且竟然沒有任何表情會被誤判成 Disgusting，有點神奇。

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？(Collaborators:b04902082 鍾偉傑)

選用 train.csv 第五張圖，由產生的 saliency map 可知，主要判斷方式是透過嘴巴方式來判別此圖



(original image)

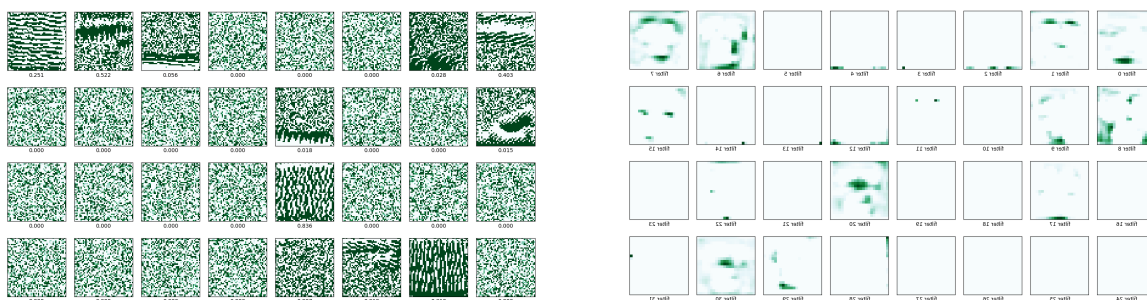
(heat map)

(saliency map)

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**。

(Collaborators: b04902082 鍾偉傑)

依樣用第五張(跟第四題一樣)那張圖來做圖，採用 conv2d_2(第二個 layer 的 conv2d)



左圖是 conv2d_2 裡面前 32 個 filter 所學到的圖，右圖是該層前 32 個 filter 輸入第五張圖片後，過濾什麼圖形。右圖到後面幾乎空白，很迷顯應該是 train 壞了。