

Homework3 Report

Professor Pei-Yuan Wu

EE5184 - Machine Learning

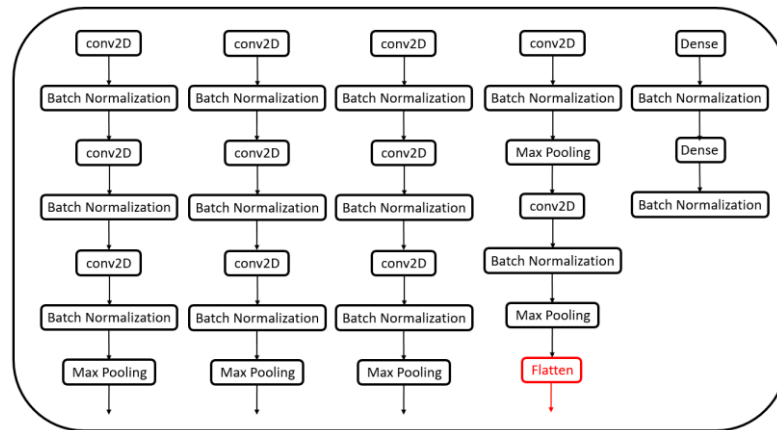
姓名：陳禹達

系級：資工碩一

學號：R07922120

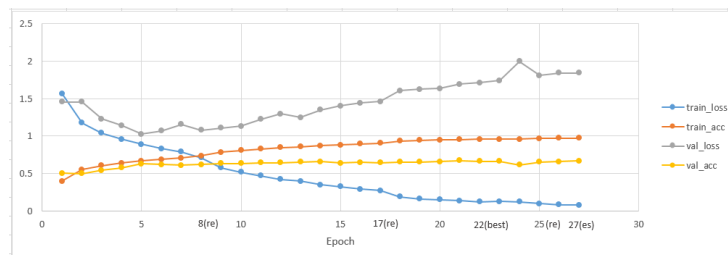
1. (1%) 請說明你實作的 CNN model，其模型架構、訓練過程和準確率為何？

CNN model :



上面是我的 CNN 架構，我的 Convolution 共用了 11 層，Max Pooling Layer5 層，Dense Layer2 層，為了避免 Overfitting，在每個 Dense Layers 和 Convolution Layers 加上一層 Batch Normalization。

訓練過程：



除了上述的架構之外，我另外用了 Image data generator，將圖片作些許的平移、旋轉等方法產生新的 Data，讓可以 train 的資料量變大，並且在每三次 validation accuracy 未提升時將 learning rate 減半，此方法讓我將 validation accuracy 從 0.61 左右提升至 0.68 左右，另外也加上了 early stopping，出現六次 validation accuracy 未提升時，則終止訓練，原本是為了避免 train 的太久造成 overfitting，但在加上

Model check point 後，因為可以存下最好的 validation accuracy 時的 model，因此 Early stopping 變成了我省下時間的方法。

準確率：

利用上述的方法後，我的 model 的 public score 為 0.66035 而 private score 為???, 但因為每一次 training 的過程我都會將 0.1 的資料切出來做為 validation set，因此可以 train 的資料會少很多，為了解決這樣的問題，我 train 了十個 model，每一次的 validation set 都是隨機切割，利用這個方法後最終我的 public score 為 0.71189，而 private score 為???。

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？

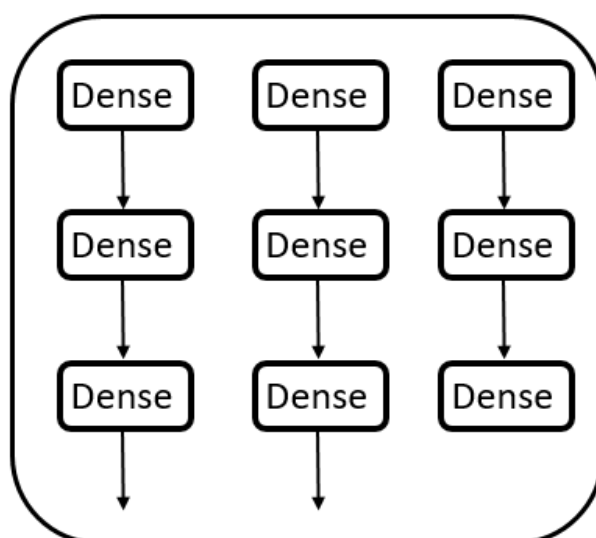
模型架構：

| | |
|---|---|
| Total params: 2,494,279 Trainable params: 2,489,351 Non-trainable params: 4,928 | Total params: 2,430,727 Trainable params: 2,430,727 Non-trainable params: 0 |
|---|---|

CNN 參數量

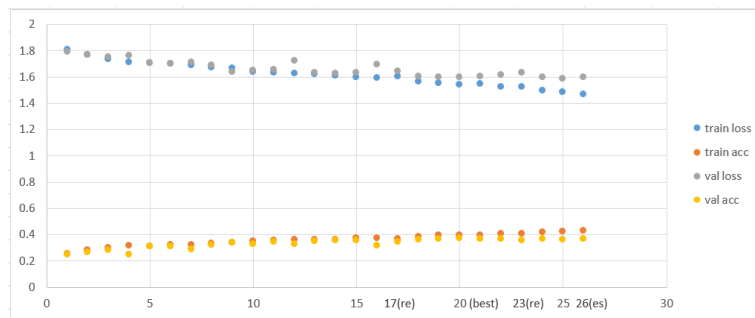
DNN 參數量

DNN 的模型架構為



共用了九層的 Dense Layers

訓練過程：同樣使用了 Model Check Point、Early Stopping、Reduce Learning Rate，但這次沒有使用 Image Data Generator，可以看出 training accuracy 上升十分緩慢，而 validation accuracy 也非常低大約 0.37 左右。



準確率：

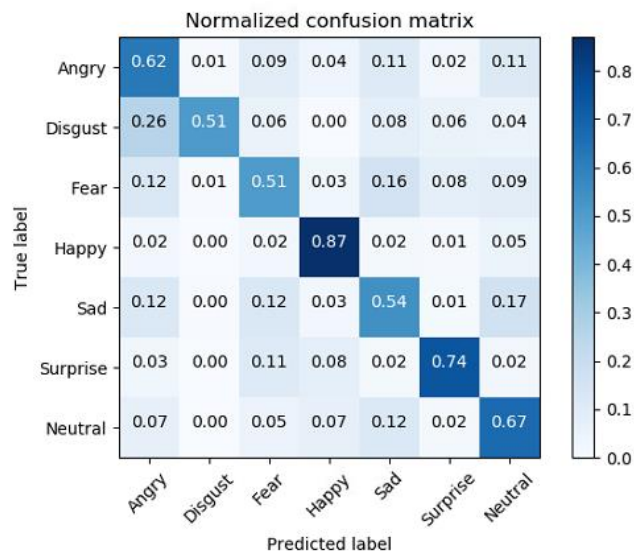
最終我的 Dense model 的 validation accuracy 為 0.3682, public score 為 0.36751, private score 為 ???。

CNN 與 DNN 比較：

在訓練過程中，DNN 較 CNN 無法快速訓練，train 到 train accuracy 約 0.4 左右就無法繼續往上升了，且 validation accuracy 以及 test accuracy 都明顯比 CNN 還要低，幾乎是一半了，但在每一個 epoch，DNN 卻較為快速訓練完畢，我的 CNN model 每一個 Epoch 大約要 26 秒完成，但 DNN model 卻只要一秒左右即可完成一個 Epoch，整整比 CNN model 快了二十幾倍。

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]

Confusion matrix：



上面是我繪製的 Confusion Matrix，將誤認大於 0.1 的 class 拿出來做分析，容易混淆的 class 為：

1. 將 Angry 誤判為 Sad, Neutral
2. 將 Disgust 誤判為 Angry

3. 將 Fear 誤判為 Angry, Sad
4. 將 Fear 誤判為 Sad
5. 將 Surprise 誤判為 Fear
6. 將 Neutral 誤判為 Sad

上述混淆的 classes 大致都為負面情緒，我認為造成此結果的原因為，人在表達負面情緒或者正面情緒時，表情大多都是相似的，而在這些 classes 中大多都是負面情緒因此容易產生混淆，反而正面情緒只有 Happy 這個 class 因此不容易產生混淆，另外雖然有一個中立表情，但很多人在情緒不佳時也是面無表情的，因此容易造成與負面情緒混淆，而恐懼常常伴隨著驚訝，兩者的表情也極為相似，因此兩者造成混淆我認為也很合理。

-----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 = (n,n);
padding = p;
strides = (s,s);
```

- a. How many parameters are there in each layer(Hint: you may consider whether the number of parameter is related with)

Layer A: $(2*2*5+1)*6=126$ (個)

Layer B: $(2*2*6+1)*4=100$ (個)

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

Layer A:

加法： $(2*2*5-1)*9*6=1026$ (次)

乘法： $2*2*5*9*6=1080$ (次)

Layer B:

加法： $(2*2*6-1)*1*4=92$ (次)

乘法： $2*2*6*1*4=96$ (次)

c. 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)

前一層的 feature map C_{i-1} 為 $N_{i-1} \times N_{i-1}$
經過 padding 後為 $(N_{i-1}+P_i) \times (N_{i-1}+P_i)$
令 kernel 在一個 channel 的 feature map 上花了 t_i 的時間
則 $k_i \times t_i + (S_i - k_i)(t_i - 1) = (N_{i-1} + P_i)$
 $\Rightarrow S_i t_i - S_i + k_i = N_{i-1} + P_i$
 $\Rightarrow t_i = \frac{N_{i-1} + P_i - k_i}{S_i} + 1$
 $T_i = t_i \times C_{i-1} \times C_i \times k_i^2$
 $= O(C_{i-1} \times C_i \times k_i^2 \times (\frac{N_{i-1} + P_i - k_i}{S_i} + 1)^2)$
 $T = O(\sum_{i=1}^L C_{i-1} \times C_i \times k_i^2 \times (\frac{N_{i-1} + P_i - k_i}{S_i} + 1)^2)$

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) What are the principal axes?
 - (2) Compute the principal components for each sample.
 - (3) Reconstruction error if reduced to 2D. (Calculate the L2-norm)

此題小數點部分四捨五入取至第二位

Matrix=

Mean=[5.4 8 4.8]

Matrix-Mean(for every row)=

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 8 & 5 \\ 3 & 12 & 9 \\ 1 & 8 & 5 \\ 5 & 14 & 2 \\ 7 & 4 & 1 \\ 9 & 8 & 9 \\ 3 & 8 & 1 \\ 11 & 5 & 6 \\ 10 & 11 & 7 \end{bmatrix}$$

$$\begin{bmatrix} -4.4 & -6 & -1.8 \\ -1.4 & 0 & 0.2 \\ -2.4 & 4 & 4.2 \\ -4.4 & 0 & 0.2 \\ -0.4 & 6 & -2.8 \\ 1.6 & -4 & -3.8 \\ 3.6 & 0 & 4.2 \\ -2.4 & 0 & -3.8 \\ 5.6 & -3 & 1.2 \\ 4.6 & 3 & 2.2 \end{bmatrix}$$

Covariance matrix=

$$\begin{bmatrix} 13.38 & 0.56 & 3.64 \\ 0.56 & 13.56 & 3.22 \\ 3.64 & 3.22 & 9.07 \end{bmatrix}$$

a.

Principal axes :

$$\begin{bmatrix} 0.62 & -0.68 & 0.40 \\ 0.59 & 0.73 & 0.38 \\ 0.52 & -0.03 & -0.85 \end{bmatrix}$$

b.

Principal components :

$$\begin{bmatrix} -7.19 & -1.37 & -2.25 \\ -0.76 & 0.94 & -0.73 \\ 3.07 & 4.45 & -3.19 \\ -2.61 & 2.98 & -1.93 \\ 1.82 & 4.75 & 4.25 \\ -3.35 & -3.92 & 2.53 \\ 4.41 & -2.56 & -2.14 \\ -3.47 & 1.73 & 2.28 \\ 2.31 & -6.03 & 0.2 \\ 5.75 & -0.78 & 0.98 \end{bmatrix}$$

c.

Reconstruct :

$$\begin{bmatrix} 1.90 & 2.76 & 1.08 \\ 4.29 & 8.25 & 4.38 \\ 4.27 & 13.08 & 6.28 \\ 1.77 & 8.65 & 3.36 \\ 3.30 & 12.56 & 5.62 \\ 5.99 & 3.15 & 3.15 \\ 9.86 & 8.72 & 7.18 \\ 2.09 & 7.23 & 2.94 \\ 10.92 & 4.93 & 6.17 \\ 9.61 & 10.67 & 7.83 \end{bmatrix}$$

Error : 2.34