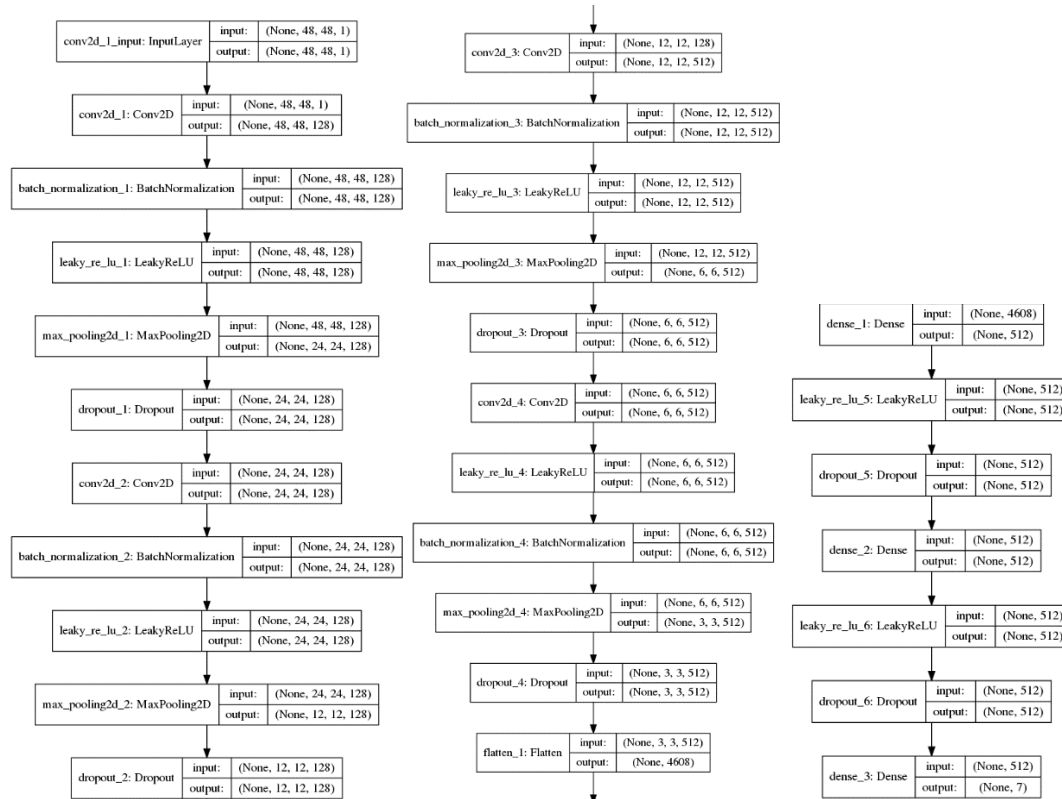


Homework 3 Report

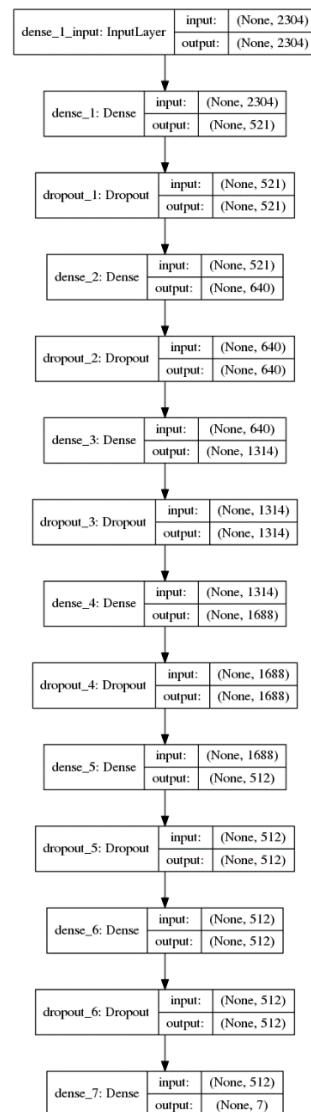
學號：r06631035 系級：生機所碩二 姓名：王凱陞

1.



上圖為我所使用的 CNN 架構，因為過長所以分為三段，單一使用這個模型 public score 可以獲得 0.67901，private 則可以達到 0.66759，再和助教手把手所使用的 VGG 縮減版本模型做 ensemble 便可達到 public 為 0.70075，private 則為 0.69267，訓練的過程中我是由小到大以二的倍數去增加 layer 的 node 數，Layer 量則是觀察 validation 的狀況進行微調，其中這個模型如果沒做任何 data augmentation 會完全訓練不起來，在測試中也發現將 Relu 改為 LeakyRelu 可以幫助在模型加深的时候比較可以訓練起來狀況，其餘參數則幾乎都是由小到大測試得到的較佳結果，在訓練中原本都是用 cross validation 的方式並取最好的 validation epoch 再用完整的資料再跑一次 testing 資料，但是最後發現在 train 這種較為龐大架構的模型時，這樣的做法實在太耗時，所以只將 validation 設的很小，直接把 training 的 model 拿去做 testing，也因為這次的資料似乎切很小的 validation 就可以達到很接近 public 的結果，所以才選擇用這樣的做法進行 training。

2.

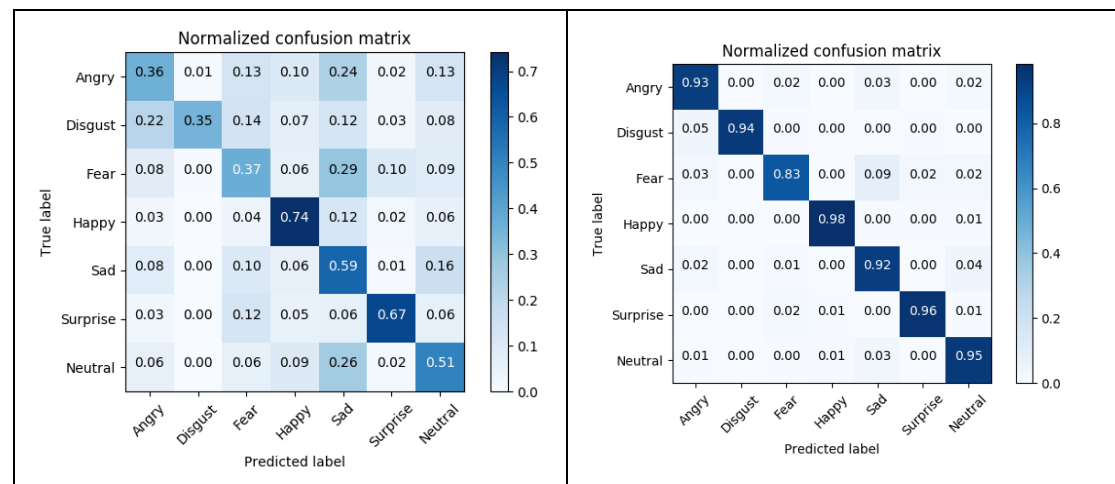


上圖為使用 DNN 進行接近 CNN 的 MODEL，訓練中 training data accuracy 上升速度明顯比 CNN 慢，且 validation 也最高只到 0.3 左右，丟到 kaggle 的分數更糟，其中 DNN 的總參數為 5727994，而 CNN 則為 5994375，而 DNN 的訓練速度明顯較快，但是效果極差，private 最高只到 0.249 最高，基本上都在 0.15 左右。

3. 此題以過 simple baseline 的模型進行作答，左圖為使用 validation set 所跑出的 confusion matrix，可以看到訓練出的結果生氣、厭惡、恐懼做的相對較差，其中較容易搞錯的幾個 class 由預測的角度來看(從 columns 看)有：

(生氣、厭惡)、(生氣、恐懼、難過、中立)

再由 training set 的結果(右圖)來觀察，這幾項容易搞錯的 class 在 training set 時也有較於其他幾項 class 容易搞錯一些，因此這幾項即為此資料中較為容易搞混的 class



4.

a. 依照題意可透過 $\frac{\text{input length} - \text{kernel length} + 2 \times \text{padding}}{\text{strides length}} + 1$ 計算 output length，帶

入題目所給的 Layer A 參數，則可算出 $\text{output length} = \frac{8 - 2 + 2 \times 0}{3} + 1 = 3$ ，則單一

channel 的 convolution layer 對 input 的五個 channel 進行運算，加上 bias，則會有 $2 \times 2 \times 5 + 1$ 個 parameters，Layer A 中共有六個 channel，故有

$$\text{Layer A's parameters} = (2 \times 2 \times 5 + 1) \times 6 = 126$$

Layer A 得到的 output shape 為 (3, 3, 6)，則可透過上述公式計算出 Layer B 的

$\text{output length} = \frac{3 - 2 + 2 \times 0}{2} + 1 = 1$ ，則可得單一 channel 的 parameters 為

$2 \times 2 \times 6 + 1$ ，共有四個 channel，故可得

$$\text{Layer B's parameters} = (2 \times 2 \times 6 + 1) \times 4 = 100$$

b. 承 a 所計算的 output length

Layer A:

乘法: 則為 filter size \times output size \times input channel \times filter channel，即 filter 內的 elements 所做的乘法，則為 $2 \times 2 \times 3 \times 3 \times 5 \times 6 = 1080$ 。

加法: 一個 kernel 大小對應到一個 Input channel 的運算會是 $(2 \times 2 - 1)$, 每個 input channel 則為 $(2 \times 2 - 1) \times 5$, 並相加則為 $(2 \times 2 - 1) \times 5 + 4$, 計算加法的總數考量, 還需乘上 output size \times filter channel 才為總計算加法次數, 故可得為 $((2 \times 2 - 1) \times 5 + 4) \times 9 \times 6 = 1026$

Layer B:

乘法: 承上述討論的方式, 則為 $2 \times 2 \times 1 \times 6 \times 4 = 96$ 。

加法: 一個 kernel 大小對應到一個 Input channel 的運算會是 $(2 \times 2 - 1)$, 每個 input channel 則為 $(2 \times 2 - 1) \times 6$, 並相加則為 $(2 \times 2 - 1) \times 6 + 5$, 計算加法的總數考量, 還需乘上 output size \times filter channel 才為總計算加法次數, 故可得為 $((2 \times 2 - 1) \times 6 + 5) \times 4 = 92$

c. 根據 output length 公式可計算出 output shape, 則 C_l 層的 output shape 為

$$\left(\frac{n_{l-1} - k_{l-1} + 2 \times p_{l-1}}{s_{l-1}}, \frac{n_{l-1} - k_{l-1} + 2 \times p_{l-1}}{s_{l-1}}, c_{l-1} \right), \text{ 而每一層的 big-O 為即為 b 小題所計算}$$

的乘法運算(加法為線性運算), 即為 filter size \times output size \times input channel \times filter channel, 將每層的運算累加則可得在 C_l 的時間複雜度為

$$O\left(\sum_{i=1}^l k_i^2 \times \left(\frac{n_{l-1} - k_{l-1} + 2 \times p_{l-1}}{s_{l-1}}\right)^2 \times c_l \times c_{l-1}\right)$$

5.

a. 欲計算 principal axes, 先計算出資料的 covariance matrix 為

$$\begin{bmatrix} 13.378 & 0.556 & 3.644 \\ 0.556 & 13.566 & 3.222 \\ 3.644 & 3.222 & 9.067 \end{bmatrix} \text{ (計算時考量樣本變異數則除上 } N-1 = 9 \text{), 計算其將其}$$

eigenvalue 由大到小排列並以對角矩陣表示為 $\begin{bmatrix} 16.997 & 0 & 0 \\ 0 & 12.923 & 0 \\ 0 & 0 & 6.080 \end{bmatrix}$, 則其

對應的 eigenvector 矩陣則為 $\begin{bmatrix} -0.617 & -0.678 & 0.399 \\ -0.589 & 0.734 & 0.338 \\ -0.523 & -0.027 & -0.852 \end{bmatrix}$, 則 principal axes 為

$$\begin{bmatrix} -0.617 \\ -0.589 \\ -0.523 \end{bmatrix}, \begin{bmatrix} -0.678 \\ 0.734 \\ -0.027 \end{bmatrix}, \begin{bmatrix} 0.399 \\ 0.338 \\ -0.852 \end{bmatrix}$$

b. 欲求 principle component, 即資料投影至 principle axes 上, 透過即資料所排列之矩陣 $\mathbf{X}(10 \times 3)$ 和 eigenvector 矩陣 $\mathbf{U}(3 \times 3)$ 內積則可得到, 資料分別投影至 principle axes 的矩陣, 則各資料對應的 principle component 如下:

Data	Component 1	Component 2	Component 3
(1, 2, 3)	-3.36201464	0.70874446	-1.48139761
(4, 8, 5)	-9.78988804	3.02597728	0.03941652
(3, 12, 9)	-13.61894165	6.53257419	-2.41865723
(1, 8, 5)	-7.94010395	5.06051399	-1.16014972
(5, 14, 2)	-12.37159312	6.83599606	5.02123906
(7, 4, 1)	-7.19402383	-1.83697744	3.29720109
(9, 8, 9)	-14.96324467	-0.47405978	-1.36988181
(3, 8, 1)	-7.0829102	3.81329871	3.0481365
(11, 5, 6)	-12.86219784	-3.95173109	0.97349277
(10, 11, 7)	-16.30109667	1.10550298	1.74702909

c. 依照題意，若降維至二維各資料取前兩個 component，則可得降維後的資料 $\hat{\mathbf{X}}(10 \times 2)$ ，則 reconstruction 則可透過 $\hat{\mathbf{X}} \cdot \hat{\mathbf{U}}^T$ 計算，其中 $\hat{\mathbf{U}}$ 為取前兩個對應的 eigenvalue 較大的 eigenvector 所組成的矩陣，再透過 L2 Norm 可得每個資料的 ERROR:

Data	Error
(1, 2, 3)	1.48139761
(4, 8, 5)	0.03941652
(3, 12, 9)	2.41865723
(1, 8, 5)	1.16014972
(5, 14, 2)	5.02123906
(7, 4, 1)	3.29720109
(9, 8, 9)	1.36988181
(3, 8, 1)	3.0481365
(11, 5, 6)	0.97349277
(10, 11, 7)	1.74702909