

# Machine Learning HW3 Report

電機三 B05901043 莊鎧爾

1.

Model 如下

Layer type	Kernel size	Output shape	Activation
Conv2D	(3, 3)	(46, 46, 128)	Relu
Conv2D	(3, 3)	(44, 44, 128)	Relu
Max Pooling	----	(22, 22, 128)	----
dropout	----	(22, 22, 128)	----
Conv2D	(3, 3)	(20, 20, 256)	Relu
Conv2D	(3, 3)	(18, 18, 256)	Relu
Max Pooling	----	(9, 9, 256)	----
dropout	----	(9, 9, 256)	----
Conv2D	(3, 3)	(7, 7, 512)	Relu
Conv2D	(2, 2)	(6, 6, 512)	Relu
Max Pooling	----	(3, 3, 512)	----
Dropout	----	(3, 3, 512)	----
Flatten	----	4608	----
Dense	----	2048	Relu
Dropout	----	2048	----
Dense	----	1024	Relu
Dropout	----	1024	----
Dense	----	512	Relu
Dense	----	7	Softmax

Total params: 15,329,159

一開始直接用 training data 訓練，其中有 20% 的 training data 用來當作 validation data，此時在 Kaggle 可以得到 public score = 0.64391。

之後加入 ImageDataGenerator，使他自己產生一些新的 training data，而 validation data 仍維持原本的 20% training data，得到 Kaggle public score = 0.65561。

2.

DNN Model

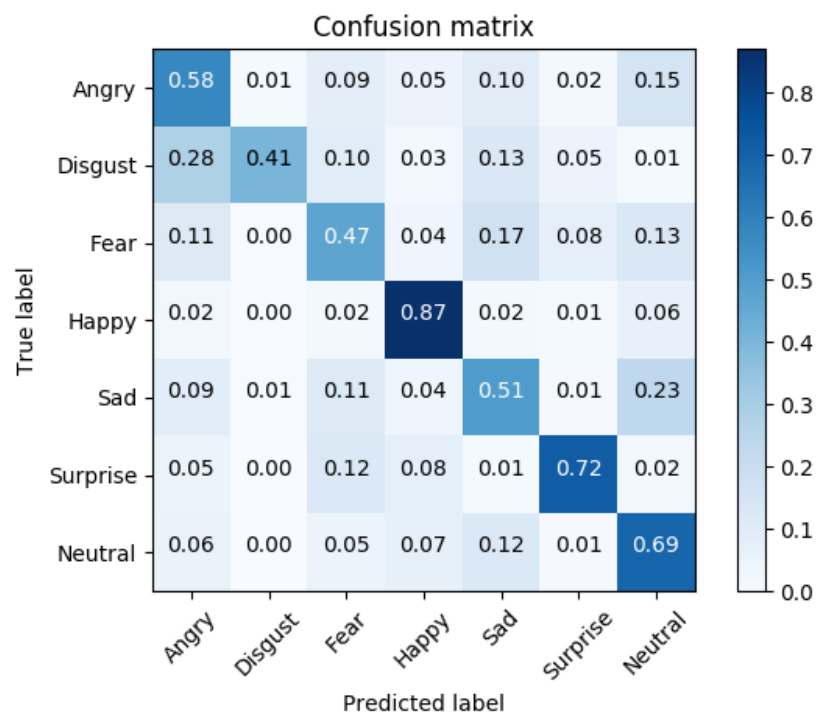
Layer type	Kernel size	Output shape	Activation
Flatten	----	2304	----
Dense	----	1537	Relu
Dense	----	1534	Relu
Dropout	----	1534	----
Dense	----	1534	Relu

Dropout	-----	1534	-----
Dense	-----	1534	Relu
Dropout	-----	1534	-----
Dense	-----	1534	Relu
Dropout	-----	1534	-----
Dense	-----	1534	Relu
Dropout	-----	1534	-----
Dense	-----	7	Softmax

Total params: 15,331,582

使用的 validation set 和 CNN 是一樣的，訓練過程也都一樣，用 `sgd`，得到 Kaggle public score = 0.48314，遠遠低於 CNN 可以達到的準確率。

3.



‘Disgust’ 容易被辨識成 ‘Angry’

‘Sad’ 容易被辨識成 ‘Neutral’

‘Fear’ 容易被辨識成 ‘Sad’

4.

```
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'))
```

a. 如果 layer 中考慮 bias

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

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

如果 layer 中不考慮 bias

$$\text{Layer A: } 6 \times (2 \times 2 \times 5) = 120$$

$$\text{Layer B: } 4 \times (2 \times 2 \times 6) = 96$$

b. 根據 Report hint，這題不考慮 bias 的計算

Layer A:

$$\text{每一個 kernel 的乘法次數} = 2 \times 2 \times 5 = 20$$

$$\text{每一個 kernel 的加法次數} = 2 \times 2 \times 5 - 1 = 19$$

$$\text{一個 kernel 的計算次數} = \left(\frac{8-2}{3} + 1\right)^2 = 9$$

$$\text{Layer A 的乘法次數} = 6 \times 9 \times 20 = 1080$$

$$\text{Layer A 的加法次數} = 6 \times 9 \times 19 = 1026$$

Layer B:

$$\text{每一個 kernel 的乘法次數} = 2 \times 2 \times 6 = 24$$

$$\text{每一個 kernel 的加法次數} = 2 \times 2 \times 6 - 1 = 23$$

$$\text{一個 kernel 的計算次數} = \left(\left\lfloor \frac{3-2}{2} \right\rfloor + 1\right)^2 = 1$$

$$\text{Layer B 的乘法次數} = 4 \times 1 \times 24 = 96$$

$$\text{Layer B 的加法次數} = 4 \times 1 \times 23 = 92$$

c. 對 Layer l:  $\text{kernel size} = (k_i \times k_i)$

$$\text{channel size} = c_i$$

$$\text{input shape} = (n_i \times n_i)$$

$$\text{padding} = p_i$$

$$\text{strides} = (s_i, s_i)$$

每一個 *kernel* 的乘法次數 =  $k_i \times k_i \times c_{i-1} = k_i^2 c_{i-1}$

每一個 *kernel* 的加法次數 =  $k_i \times k_i \times c_{i-1} - 1 = k_i^2 c_{i-1} - 1$

$$n_i = \frac{n_{i-1} + p_{i-1} - k_{i-1}}{s_{i-1}} + 1$$

$$\text{一個 } kernel \text{ 的計算次數} = \left( \frac{n_i + p_i - k_i}{s_i} + 1 \right)^2$$

$$Layer B \text{ 的乘法次數} = c_i \times \left( \frac{n_i + p_i - k_i}{s_i} + 1 \right)^2 \times k_i^2 c_{i-1}$$

$$Layer B \text{ 的加法次數} = c_i \times \left( \frac{n_i + p_i - k_i}{s_i} + 1 \right)^2 \times (k_i^2 c_{i-1} - 1)$$

Time complexity of the CNN is:

$$O\left(\sum_{i=1}^l c_i \times \left( \frac{n_i + p_i - k_i}{s_i} + 1 \right)^2 \times k_i^2 c_{i-1}\right)$$

where  $c_0 = 1$

5.

a.

$$\mu = (5.4, 8.0, 4.8)$$

$$\Sigma = \frac{1}{N} \sum_{i=1}^N (x_i - \mu)(x_i - \mu)^T = \begin{pmatrix} 12.04 & 0.5 & 3.28 \\ 0.5 & 12.2 & 2.9 \\ 3.28 & 2.9 & 8.16 \end{pmatrix}$$

$$= \begin{pmatrix} 0.40 & -0.68 & -0.62 \\ 0.34 & 0.73 & -0.59 \\ -0.85 & -0.027 & -0.52 \end{pmatrix} \begin{pmatrix} 5.47 & 0 & 0 \\ 0 & 11.6 & 0 \\ 0 & 0 & 15.3 \end{pmatrix} \begin{pmatrix} 0.40 & -0.68 & -0.62 \\ 0.34 & 0.73 & -0.59 \\ -0.85 & -0.027 & -0.52 \end{pmatrix}^T$$

可以得到 principle axes

$$u_1 = \begin{pmatrix} -0.62 \\ -0.59 \\ -0.52 \end{pmatrix}^T, u_2 = \begin{pmatrix} -0.68 \\ 0.73 \\ -0.027 \end{pmatrix}^T, u_3 = \begin{pmatrix} 0.40 \\ 0.34 \\ -0.85 \end{pmatrix}^T$$

b.

現在把每一個向量  $v$  表示成  $(a, b, c)$ ，其中  $a, b, c$  滿足  $v = au_1 + bu_2 + cu_3$ ，所以，十個向量的座標轉換如下：

原座標	(1, 2, 3)	(4, 8, 5)	(3, 12, 9)	(1, 8, 5)	(5, 14, 2)
轉換後座標	(-3.36, 0.71, -1.48)	(-9.79, 3.03, 0.039)	(-13.6, 6.53, -2.42)	(-7.94, 5.06, -1.16)	(-12.4, 6.84, 5.02)

原座標	(7, 4, 1)	(9, 8, 9)	(3, 8, 1)	(11, 5, 6)	(10, 11, 7)
轉換後座標	(-7.19, -1.84, 3.30)	(-15.0, -0.47, -1.37)	(-7.08, 3.81, 3.05)	(-12.9, -3.95, 0.97)	(-16.3, 1.11, 1.75)

c.

去掉  $u_3$ ，並重建這些向量

原座標	(1, 2, 3)	(4, 8, 5)	(3, 12, 9)	(1, 8, 5)	(5, 14, 2)
轉換後座標	(-3.36, 0.71)	(-9.79, 3.03)	(-13.6, 6.53)	(-7.94, 5.06)	(-12.4, 6.84)
重建向量	(1.59, 2.50, 1.74)	(3.98, 7.99, 5.03)	(3.97, 12.8, 6.94)	(1.46, 8.39, 4.01)	(2.99, 12.3, 6.28)

原座標	(7, 4, 1)	(9, 8, 9)	(3, 8, 1)	(11, 5, 6)	(10, 11, 7)
轉換後座標	(-3.36, 0.71)	(-9.79, 3.03)	(-13.6, 6.53)	(-7.94, 5.06)	(-12.4, 6.84)
重建向量	(5.68, 2.89, 3.81)	(9.55, 8.46, 7.83)	(1.78, 6.97, 3.60)	(10.6, 4.67, 6.83)	(9.30, 10.4, 8.49)

$$L2\ norm = 2.46$$