Homework 2 Report

資工所碩二 R06922132 何羿辰

EE5184 - Machine Learning

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？

Layer (type)	Output Shape	Param #
conv2d_1 (Conv2D)	(None, 32, 48, 48)	320
batch_normalization_1 (Batch	(None, 32, 48, 48)	128
activation_1 (Activation)	(None, 32, 48, 48)	0
conv2d_2 (Conv2D)	(None, 32, 48, 48)	9248
batch_normalization_2 (Batch	(None, 32, 48, 48)	128
activation_2 (Activation)	(None, 32, 48, 48)	0
conv2d_3 (Conv2D)	(None, 32, 48, 48)	9248
batch_normalization_3 (Batch	(None, 32, 48, 48)	128
activation_3 (Activation)	(None, 32, 48, 48)	0
max_pooling2d_1 (MaxPooling2	(None, 32, 24, 24)	0
conv2d_4 (Conv2D)	(None, 64, 24, 24)	18496
batch_normalization_4 (Batch	(None, 64, 24, 24)	256
activation_4 (Activation)	(None, 64, 24, 24)	0
conv2d_5 (Conv2D)	(None, 64, 24, 24)	36928
batch_normalization_5 (Batch	(None, 64, 24, 24)	256
activation_5 (Activation)	(None, 64, 24, 24)	0
conv2d_6 (Conv2D)	(None, 64, 24, 24)	36928
batch_normalization_6 (Batch	(None, 64, 24, 24)	256
activation_6 (Activation)	(None, 64, 24, 24)	0
max_pooling2d_2 (MaxPooling2	(None, 64, 12, 12)	0
conv2d_7 (Conv2D)	(None, 128, 12, 12)	73856
batch_normalization_7 (Batch	(None, 128, 12, 12)	512
activation_7 (Activation)	(None, 128, 12, 12)	0
conv2d_8 (Conv2D)	(None, 128, 12, 12)	147584
batch_normalization_8 (Batch	(None, 128, 12, 12)	512
activation_8 (Activation)	(None, 128, 12, 12)	0
conv2d_9 (Conv2D)	(None, 128, 12, 12)	147584
batch_normalization_9 (Batch	(None, 128, 12, 12)	512
activation_9 (Activation)	(None, 128, 12, 12)	0
max_pooling2d_3 (MaxPooling2	(None, 128, 6, 6)	0
flatten_1 (Flatten)	(None, 4608)	0
dense_1 (Dense)	(None, 1024)	4719616
batch_normalization_10 (Batc	(None, 1024)	4096
activation_10 (Activation)	(None, 1024)	0
dropout_1 (Dropout)	(None, 1024)	0
dense_2 (Dense)	(None, 7)	7175

Total params: 5,213,767
Trainable params: 5,210,375
Non-trainable params: 3,392

訓練過程:

固定參數: Batch size 固定為 128，optimizer 固定使用 ADAM

Preprocess:

```
data_generator_train = ImageDataGenerator(
    rotation_range=15, width_shift_range=0.1, height_shift_range=0.1, horizontal_flip=True, data_format='channels_first')
```

變動參數: 前 75 個 epoch 固定使用 lr 為 0.01，之後的 75~300epoch 從 lr=0.001 開始 reduce lr 尋找

最佳解

訓練前有先切出 10%資料作為 validation set，剩下的 90%則為 training set

準確率:最後準確率為 0.68152

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？

Layer (type)	Output Shape	Param #
flatten_1 (Flatten)	(None, 2304)	0
dense_1 (Dense)	(None, 1024)	2360320
batch_normalization_1 (Batch	(None, 1024)	4096
activation_1 (Activation)	(None, 1024)	0
dense_2 (Dense)	(None, 1024)	1049600
batch_normalization_2 (Batch	(None, 1024)	4096
activation_2 (Activation)	(None, 1024)	0
dense_3 (Dense)	(None, 1024)	1049600
batch_normalization_3 (Batch	(None, 1024)	4096
activation_3 (Activation)	(None, 1024)	0
dropout_1 (Dropout)	(None, 1024)	0
dense_4 (Dense)	(None, 512)	524800
batch_normalization_4 (Batch	(None, 512)	2048
activation_4 (Activation)	(None, 512)	0
dense_5 (Dense)	(None, 512)	262656
batch_normalization_5 (Batch	(None, 512)	2048
activation_5 (Activation)	(None, 512)	0
dense_6 (Dense)	(None, 512)	262656
batch_normalization_6 (Batch	(None, 512)	2048
activation_6 (Activation)	(None, 512)	0
dropout_2 (Dropout)	(None, 512)	0
dense_7 (Dense)	(None, 256)	131328
batch_normalization_7 (Batch	(None, 256)	1024
activation_7 (Activation)	(None, 256)	0
dense_8 (Dense)	(None, 256)	65792
batch_normalization_8 (Batch	(None, 256)	1024
activation_8 (Activation)	(None, 256)	0
dense_9 (Dense)	(None, 256)	65792
batch_normalization_9 (Batch	(None, 256)	1024
activation_9 (Activation)	(None, 256)	0
dropout_3 (Dropout)	(None, 256)	0
dense_10 (Dense)	(None, 7)	1799

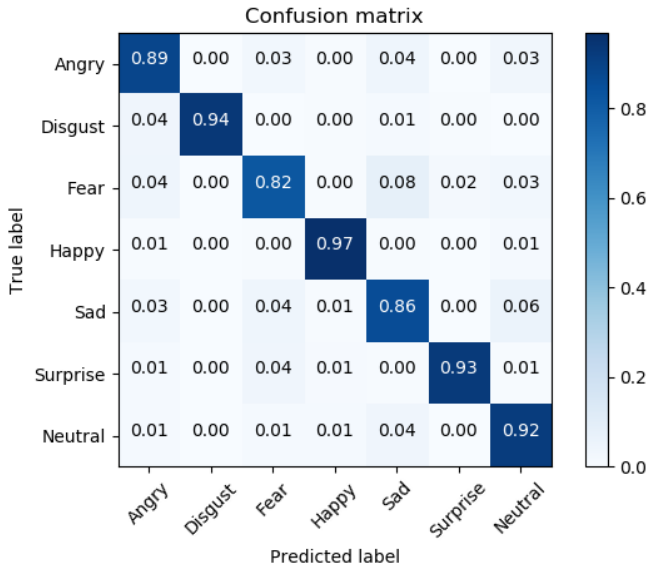
Total params: 5,795,847
Trainable params: 5,785,095
Non-trainable params: 10,752

訓練過程:

為了比較結果，所以訓練過程完全比照 CNN 的作法

準確率: 0.49902，雖然參數量比 CNN 稍微多了一些，但準確率卻差很多，可以判斷 DNN 對於影像辨識的準確度比 CNN 效率要差

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]



從 confusion matrix 中可以看出，Happy 的判斷最準確，Fear 判斷的準確度最差，而 Fear 最容易被誤判為 Sad，也許是 Fear 與 Sad 兩者的情緒表現本來就很相似，另外 Sad 也很容易被誤判為 Neutral

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),
"""Layer A"""    padding ="valid",
                 kernel_size=(2,2),
                 input_shape=(8,8,5),
                 activation='relu'))
model.add(Conv2D(filters=4,
                 strides=(2, 2),
"""Layer B"""    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 of each layer = (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:
   Layer B:
b. How many multiplications/additions are needed for a forward pass(each layer).

   Layer A:
   Layer B:
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 lth and l-1th layer)

(a) 計算公式為: ((kernel_size)*stride+1)*filters)

Layer A: ((2*2)*5+1)*6)=**126#**

Layer B: ((2*2)*6+1)*4)=**100#**

(b)

Layer A:

Additions: (2*2*5-1)*3*3*6=**1026#**

Multiplications: (2*2*5) *3*3*6=**1080#**

Layer B:

Additions: (2*2*6-1)*1*1*4=**92#**

Multiplications: (2*2*6) *1*1*4=**96#**

(c) 所求為所有卷積層的時間複雜度相加，而每個捲積層的複雜度為:

輸出特徵圖面積乘上捲積核面積乘上輸入/輸出通道數(filter size)

假設c_1 為 input data channel size

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

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)
   a. (1) What are the principal axes?
   b. (2) Compute the principal components for each sample.
   c. (3) Reconstruction error if reduced to 2D.(Calculate the L2-norm)

(a) 以下三題皆有先做 centering

Step1:歸一化減去均值

$$\text{mean}_1 = \frac{1+4+3+1+5+7+9+3+11+10}{10} = 5.4$$

$$\text{mean}_2 = \frac{2+8+12+8+14+4+8+8+5+11}{10} = 8.0$$

$$\text{mean}_3 = \frac{3+5+9+5+2+1+9+1+6+7}{10} = 4.8$$

10 samples 減去均值後: (-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)

Step2:斜方差求 eigenvalue 與 eigenvector

斜方差矩陣求得為: $\begin{bmatrix} 13.37 & 0.56 & 3.64 \\ 0.55 & 13.55 & 3.22 \\ 3.64 & 3.22 & 9.06 \end{bmatrix}$

矩陣的 eigenvalue 與 eigenvector 依 eigenvalue 大到小排序為:

Eigenvalue: 16.99715933,12.9228041,6.08

Eigenvector: (-0.61,-0.58,-0.52),(0.67,-0.73,0.02),(0.40,0.33,-0.85) #

Principle axes 即為這些 eigenvector(老師投影片裡的$\mathbf{u}_i$)

(b) 將各個 principle axis 與歸一化後的 x 相乘做投影可得 principle component(老師投影片裡的$\mathrm{u}_i{}^{\mathrm{T}}x$)

1'st principle component:

$[\mathbf{7.18 \quad 0.75 \quad -3.07 \quad 2.6 \quad -1.82 \quad 3.35 \quad -4.41 \quad 3.46 \quad -2.31 \quad -5.75}]$ **#**

2'nd principle component:

$[\mathbf{1.37 \quad -0.94 \quad -4.45 \quad -2.97 \quad -4.75 \quad 3.91 \quad 2.55 \quad -1.73 \quad 6.03 \quad 0.97}]$ **#**

3'rd principle component:

$[\mathbf{-2.25 \quad -0.73 \quad -3.18 \quad -1.92 \quad 4.25 \quad 2.52 \quad -2.13 \quad 2.27 \quad 0.20 \quad 0.97}]$ **#**

(c) Reduce 到 2D ➔ 取前兩個 vector 重建 x ($\check{\mathrm{x}} = \mathrm{u}_i\mathrm{u}_i{}^{\mathrm{T}}x$)

重建後的 x̌=(1.9,2.75,1.08),(4.29,8.24,4.37),(4.27,13.07,6.28)

取 x 與 x̌ 的:

L2 norm error(兩者相減後平方相加) **可得 L2-norm error 為 54.72 #**

若採用 L2 norm(兩者相減後平方相加取開根號) **可得 L2-norm 為 20.47 #**