Homework3 Report

Professor Pei-Yuan Wu.

EE5184 - Machine Learning

姓名：顏百謙

學號：R07922135

Problem1

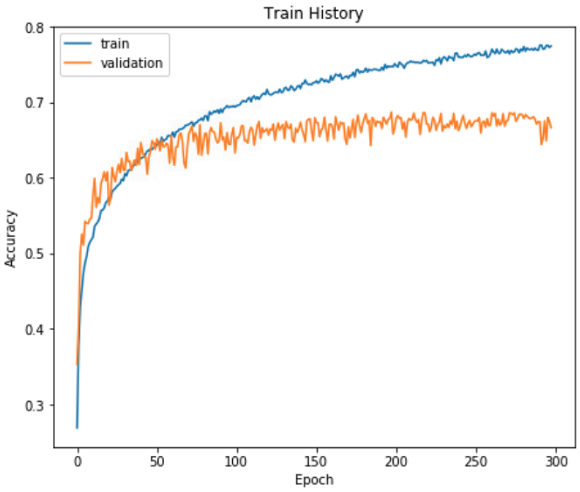
(1%) 請說明你實作的 CNN model，其模型架構、訓練過程和準確率為何？

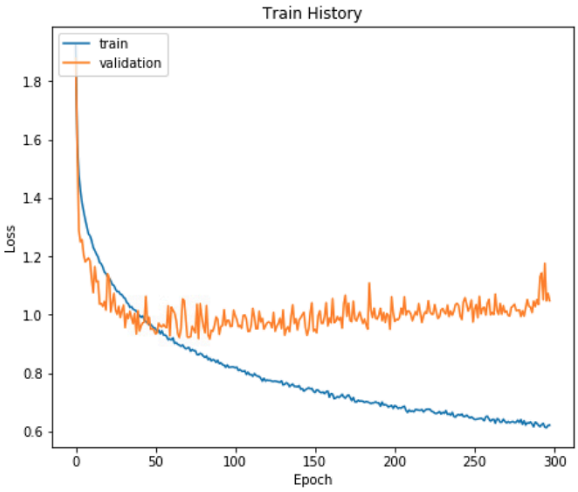
實作利用4層CNN架構(+ MaxPooling)以及兩層DNN的模型，設計如下：

- CNN Layer1
  - filter數：64 || kernel size：(3,3) || BathchNormalization || MaxPooling：(2,2) || Dropout：0.5 || LeakyReLU = (0.3)
- CNN Layer2
  - filter數：128 || kernel size：(3,3) || BathchNormalization || MaxPooling：(2,2) || Dropout：0.4 || LeakyReLU = (0.3)
- CNN Layer3
  - filter數：256 || kernel size：(3,3) || BathchNormalization || MaxPooling：(2,2) || Dropout：0.3 || LeakyReLU = (0.3)
- CNN Layer4
  - filter數：512 || kernel size：(3,3) || BathchNormalization || MaxPooling：(2,2) || Dropout：0.3 || LeakyReLU = (0.3)
- Flatten
- DNN Layer1
  - Kernel = 175 || BatchNormalization || relu || Dropout：0.7
- DNN Layer2
  - Kernel = 7 || softmax

訓練過程

在12/1重新train 1000個Epoch (early stop: 298)得到的結果做圖





準確率

- 最終在Kaggle上得到
  - public：0.67901 ||. private：0.67539

homework3_model15.csv 3 days ago by r07922135_omuraisu model15-2	0.67539	0.67901

Problem2

(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？

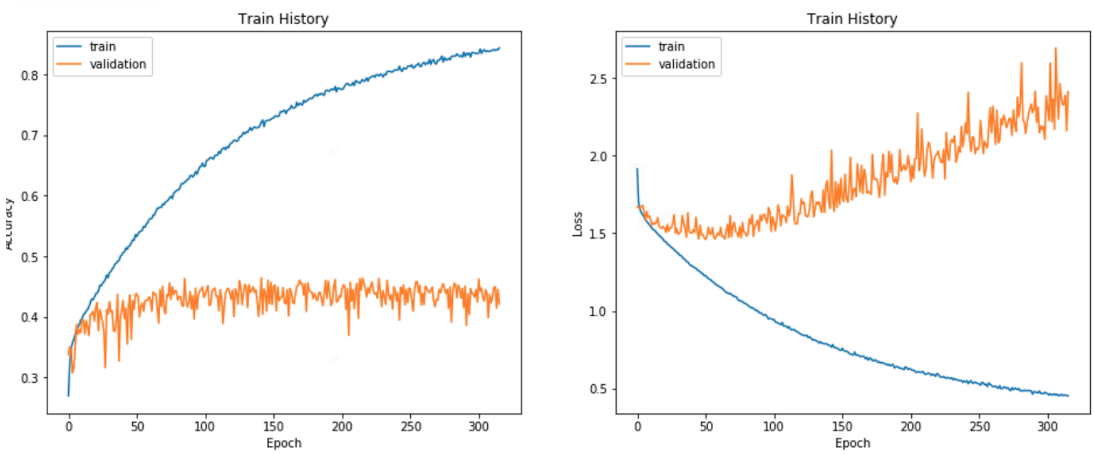
- CNN model的參數量：

```
Total params: 2,362,171
Trainable params: 2,359,901
Non-trainable params: 2,270
```

- DNN model設計：
  - 參數量：

Layer (type)	Output Shape	Param #
dense_5 (Dense)	(None, 1000)	2305000
batch_normalization_6 (Batch	(None, 1000)	4000
activation_2 (Activation)	(None, 1000)	0
dropout_6 (Dropout)	(None, 1000)	0
dense_6 (Dense)	(None, 175)	175175
batch_normalization_7 (Batch	(None, 175)	700
activation_3 (Activation)	(None, 175)	0
dropout_7 (Dropout)	(None, 175)	0
dense_7 (Dense)	(None, 7)	1232

```
Total params: 2,486,107
Trainable params: 2,483,757
Non-trainable params: 2,350
```

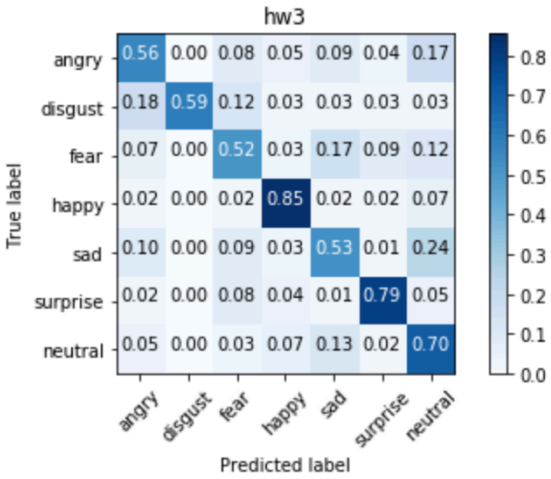


- 觀察結果：
  - DNN model很容易造成overfit，且學習效果較CNN差，原因可能是DNN都是以整張圖來進行判斷，而不像CNN有把圖利用filter切割為小區域進行訓練。

Problem3

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]

- 使用同training時的validation data來畫confusion matrix



- 由得到的結果可以發現在happy和surprise的部分有較高的正確性
- 而disgust的圖片很容易被誤判為angry和fear
- sad的圖片很容易被判為neutral
- fear的圖片容易被判為sad
- feature較不明顯的圖片都會被歸類到neutral

Problem4

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)

(4-a)

由於$input$為$(8,8,5)$故$Layer\ A$的每個$kernel$上會有$(2*2*5)$個參數，

又總共有6個$filter$，每個$filter$加上1個$bias$故可得

$$Layer\ A:(2*2*5)*6+6=126$$

由於$Layer\ B$的$input$為$Layer\ A$的$ouput$（$3,3,6$）故$Layer\ B$的每個$kernel$上會有

$(2*2*6)$個參數，又總共有4個$filter$，每個$filter$加上1個$bias$故可得

$$Layer\ B:(2*2*6)*4+4=100$$

將參數相加可得此$model$總共有226個參數

(4-b)

$Layer\ A$中每個$filter$在一個位置會執行$(2*2*5)$次乘法以及$(2*2*5-1)$次加法，

又每個$filter$在$(8,8,5)$的$input$下共會進行9次運算，

此$Layer$總共有6個$filter$，因此可得知：

$$LayerA(+):(2*2*5-1)*9*6=1026\ (\text{次加法})$$

$$LayerA(*):(2*2*5)*9*6=1080\ (\text{次乘法})$$

$Layer\ B$中每個$filter$在一個位置會執行$(2*2*6)$次乘法以及$(2*2*6-1)$次加法，

又每個$filter$在$(3,3,6)$的$input$下共會進行1次運算（若$padding=same$則為4次），

此$Layer$總共有4個$filter$，因此可得知：

$$LayerB(+):(2*2*6-1)*1*4=92\ (\text{次加法})$$

$$LayerB(*):(2*2*6)*1*4=96\ (\text{次乘法})$$

(4-c)

單層CNN的$time\ complexity$為$O(C_{in}*M^2*K^2*C_{out})$

其中

$C_{in}=$輸入的$layer$數

$K=kernel$邊長

$C_{out}=$輸出的$layer$數$=$下一層的輸入數

又

$$M_i=(X-K+2*Padding)/Stride+1=\frac{(n_i-k_i+2*p_i)}{s_i}+1$$

故整個CNN模型的$time\ complexity$為

$$O(\sum_{i=1}^{l}C_i*M_i^2*K_i^2*C_{i+1})$$

Problem5

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)

(5-a)

$$X = [\,x_0 \quad x_1 \quad x_2 \quad \cdots \quad x_8 \quad x_9\,] = \begin{bmatrix} 1 & 4 & 3 & 1 & 5 & 7 & 9 & 3 & 11 & 10 \\ 2 & 8 & 12 & 8 & 14 & 4 & 8 & 8 & 5 & 11 \\ 3 & 5 & 9 & 5 & 2 & 1 & 9 & 1 & 6 & 7 \end{bmatrix}$$

算得 $\mu = \begin{bmatrix} 5.4 \\ 8 \\ 4.8 \end{bmatrix}$

$$\Sigma = \frac{1}{9}\sum_{i=0}^{9}(x_i - \mu)(x_i - \mu)^T$$

$$= \frac{1}{9}\begin{bmatrix} 120.4 & 5 & 32.8 \\ 5 & 122 & 29 \\ 32.8 & 29 & 81.6 \end{bmatrix}$$

(Σ是利用$numpy$進行計算)

再利用$numpy$計算$eigenvalue$和$eigenvector$得到

$$eigenvalue\ \lambda_1 = 16.99716 \quad eigenvector\ v_1 = \begin{bmatrix} -0.6166 \\ -0.5888 \\ -0.5226 \end{bmatrix}$$

$$eigenvalue\ \lambda_2 = 12.92280 \quad eigenvector\ v_2 = \begin{bmatrix} -0.6782 \\ 0.7344 \\ -0.0273 \end{bmatrix}$$

$$eigenvalue\ \lambda_3 = 6.08004 \quad eigenvector\ v_3 = \begin{bmatrix} 0.3999 \\ 0.3376 \\ -0.8521 \end{bmatrix}$$

取兩個最大的$eigenvalue(\lambda_1, \lambda_2)$和其所對應的$eigenvector(v_1, v_2)$

可得：

$$Principle\ Axes_1 = v_1 = \begin{bmatrix} -0.6166 \\ -0.5888 \\ -0.5226 \end{bmatrix}$$

$$Principle\ Axes_2 = v_2 = \begin{bmatrix} -0.6782 \\ 0.7344 \\ -0.0273 \end{bmatrix}$$

(5-b)

欲取得1^{st}和$2^{nd}$$principle\ component$在各個$data$上的值只要將$x_i \in 10 samples\ (i = 0 \sim 9)$

分別做$v_1^T x_i$ 和 $v_2^T x_i$即可得到

利用$numpy$運算後可得

$$1^{st} Principle\ component = \begin{bmatrix} -3.36201464 \\ -9.78988804 \\ -13.61894165 \\ -7.94010395 \\ -12.37159312 \\ -7.19402383 \\ -14.96324467 \\ -7.0829102 \\ -12.86219784 \\ -16.30109667 \end{bmatrix}$$

$$2^{nd} Principle\ component = \begin{bmatrix} 0.70874446 \\ 3.02597728 \\ 6.53257419 \\ 5.06051399 \\ 6.83599606 \\ -1.83697744 \\ -0.47405978 \\ 3.81329871 \\ -3.95173109 \\ 1.10550298 \end{bmatrix}$$

(5-c)

假設$zero\ mean$我們可以用$dot((v_1, v_2), (1^{st}PC, 2^{nd}PC)^T)$

得到$Reconstruction$的結果：$\widehat{X^s} = \begin{bmatrix} 1.59 & 2.50 & 1.74 \\ 3.98 & 7.99 & 5.03 \\ 3.97 & 12.82 & 6.94 \\ 1.46 & 8.39 & 4.01 \\ 2.99 & 12.30 & 6.28 \\ 5.68 & 2.89 & 3.81 \\ 9.55 & 8.46 & 7.83 \\ 1.78 & 6.97 & 3.60 \\ 10.61 & 4.67 & 6.83 \\ 9.30 & 10.41 & 8.49 \end{bmatrix}^T \quad (s = 10sample)$

和原始的X計算$Reconstruction\ error$可得

$$Error = \begin{bmatrix} 1.48139761 \\ 0.03941652 \\ 2.41865723 \\ 1.16014972 \\ 5.02123906 \\ 3.29720109 \\ 1.36988181 \\ 3.0481365 \\ 0.97349277 \\ 1.74702909 \end{bmatrix}$$

$$Sum = 20.5566$$