Homework3 Report Template

Professor Pei-Yuan Wu
EE5184 - Machine Learning

姓名：蘇軒
學號：b04203058

Note:1~3 題建議不要超過三頁

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？

   Input →
   Convolution(64, (3, 3)) → batch_normalization → Average pooling(2, 2) → Dropout(0.2) →
   Convolution(128, (3, 3)) → batch_normalization → Average pooling(2, 2) → Dropout(0.25) →
   Convolution(256, (3, 3)) → batch_normalization → Average pooling(2, 2) → Dropout(0.3) →
   Convolution(512, (3, 3)) → batch_normalization → Average pooling(2, 2) → Dropout(0.4) →
   Flattern() →
   Dense(1024) → batch_normalization → Dropout(0.5) →
   Dense(512) → batch_normalization → Dropout(0.5) →
   Dense(256) → batch_normalization → Dropout(0.5) →
   Dense(7)
   Softmax()

   將 data 切出約 1/5 的數量拿來 validate
   加上 keras 內建的套件 ImageDataGenerator 將圖片進行旋轉還有歪斜來增加 data 量
   訓練的 optimizer 使用 adam 各項參數為預設
   batch size: 128
   steps_per_epoch: 10*data length//batch_size
   epoch: 300
   並且設立 model_checkpoint 在每一個 epoch 後只要有 improve validation 就會將 model save 下來，最後 validate 準確率約為 0.71323
   上傳 kaggle 分數為 0.71352。

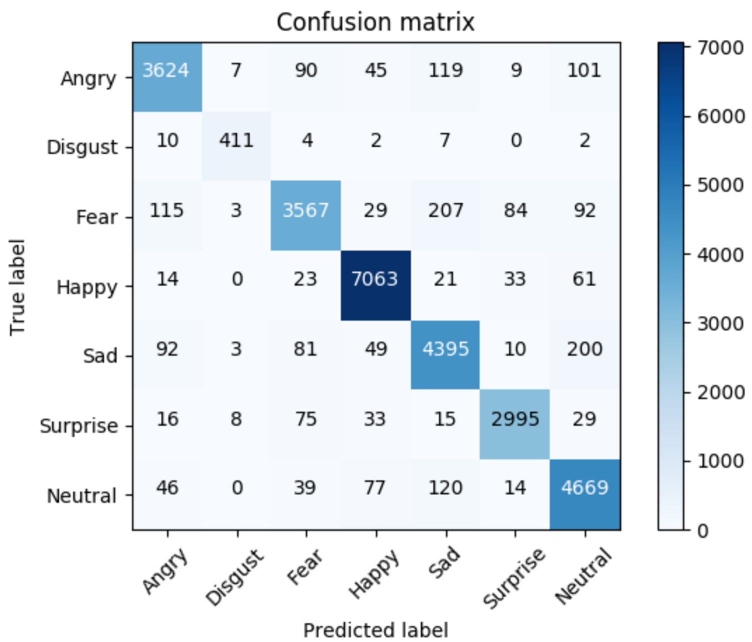
2. (1%) 承上題，請用與上述 CNN 接近的參數量，實做簡單的 DNN model，其模型架構、訓練過程和準確率為何？試與上題結果做比較，並說明你觀察到了什麼？

   Input →
   Flattern() →
   Dense(512) → batch_normalization → Dropout(0.5) →
   Dense(512) → batch_normalization → Dropout(0.5) →
   Dense(1024) → batch_normalization → Dropout(0.5) →
   Dense(1024) → batch_normalization → Dropout(0.5) →

```
Dense(1024) → batch_normalization → Dropout(0.5) →
Dense(1024) → batch_normalization → Dropout(0.5) →
Dense(1024) → batch_normalization → Dropout(0.5) →
Dense(512) → batch_normalization → Dropout(0.5) →
Dense(256) → batch_normalization → Dropout(0.5) →
Dense(7) →
Softmax()
```

將 data 切出約 1/5 的數量拿來 validate
加上 keras 內建的套件 ImageDataGenerator 將圖片進行旋轉還有歪斜來增加 data 量
訓練的 optimizer 使用 adam 各項參數為預設
batch size: 128
steps_per_epoch: 10*data length//batch_size
epoch: 300
並且設立 model_checkpoint 在每一個 epoch 後只要有 improve validation 就會將 model save 下來，最後 validate 準確率約為 0.41912
上傳 kaggle 分數為 0.40234。

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]



我們可以很明顯的發現 Fear 和 Sad 還有 Sad 和 Neutral 非常容易搞混，除此之外還可以發現就算是自己和自己的 class 也會有相似度非常低的可能發生像是 Disgust，儘管如此從 confusion matrix 我們還是可以知道這次的 model 將正面以及負面的表情區分開來了。

-----------------Handwritten question----------------
Collaborator : b04504042 劉家豪

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: (2*2*5)weights + 1 bias → 6 filters have 6*(2*2*5 +1) = 126 parameters

Layer B: (2*2*6)weights + 1 bias → 4 filters have 4*(2*2*6 +1) = 100 parameters

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

Each kernel has 2*2*5 multiplications and 2*2*5 - 1 additions

Layer A:

multiplications: 2*2*5*9*6 = 1080

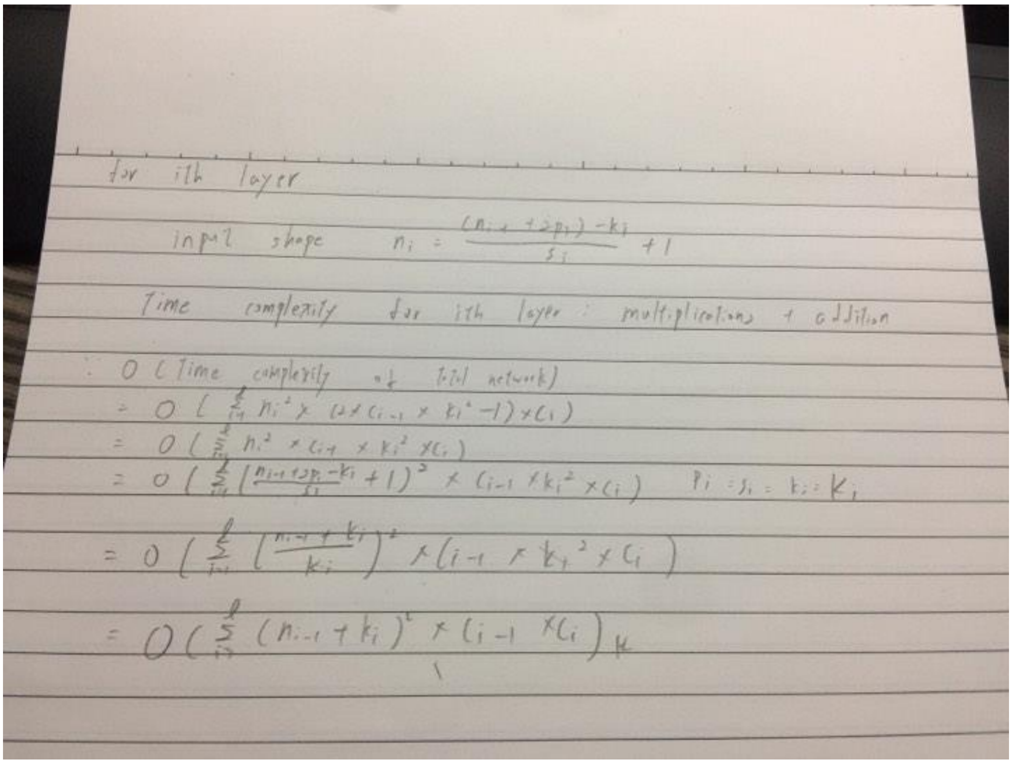
additions: (2*2*5-1)*9*6 = 1026

Layer B:

multiplications: 2*2*6*1*4 = 96

additions: (2*2*6-1)*4 = 92

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 □ □,□ □-1as lth and l-1th layer)



for ith layer

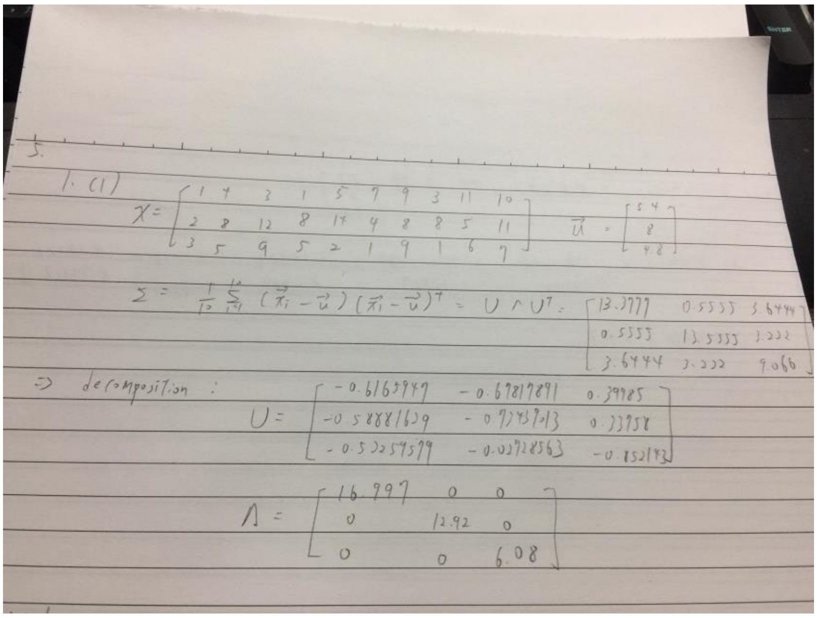
input shape $n_i = \frac{(n_{i-1} + 2p_i) - k_i}{s_i} + 1$

Time complexity for ith layer : multiplications + addition

$\therefore O(\text{Time complexity of total network})$
$= O\left(\sum_{i=1}^{l} n_i^2 \times (2 \times C_{i-1} \times k_i^2 - 1) \times C_i\right)$
$= O\left(\sum_{i=1}^{l} n_i^2 \times C_{i-1} \times k_i^2 \times C_i\right)$
$= O\left(\sum_{i=1}^{l} \left(\frac{n_{i-1} + 2p_i - k_i}{s_i} + 1\right)^2 \times C_{i-1} \times k_i^2 \times C_i\right)$ $p_i = s_i = k_i = K_i$
$= O\left(\sum_{i=1}^{l} \left(\frac{n_{i-1} + k_i}{k_i}\right)^2 \times C_{i-1} \times k_i^2 \times C_i\right)$
$= O\left(\sum_{i=1}^{l} (n_{i-1} + k_i)^2 \times C_{i-1} \times C_i\right)$ $_k$

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?



5.

1.(1)

$$X = \begin{bmatrix} 1 & 4 & 3 & 1 & 5 & 7 & 9 & 3 & 11 & 10 \\ 2 & 8 & 12 & 8 & 14 & 4 & 8 & 8 & 5 & 11 \\ 3 & 5 & 9 & 5 & 2 & 1 & 9 & 1 & 6 & 7 \end{bmatrix} \quad \vec{u} = \begin{bmatrix} 5.4 \\ 8 \\ 4.8 \end{bmatrix}$$

$$\Sigma = \frac{1}{10} \sum_{i=1}^{10} (\vec{x}_i - \vec{u})(\vec{x}_i - \vec{u})^T = U \Lambda U^T = \begin{bmatrix} 13.3777 & 0.5555 & 3.6444 \\ 0.5555 & 13.5555 & 3.222 \\ 3.6444 & 3.222 & 9.066 \end{bmatrix}$$

$\Rightarrow$ decomposition :

$$U = \begin{bmatrix} -0.6165947 & -0.67817891 & 0.39985 \\ -0.58881629 & -0.77439013 & 0.33958 \\ -0.52254579 & -0.02728563 & -0.852143 \end{bmatrix}$$

$$\Lambda = \begin{bmatrix} 16.997 & 0 & 0 \\ 0 & 12.92 & 0 \\ 0 & 0 & 6.08 \end{bmatrix}$$

b. (2) Compute the principal components for each sample.

$$P = \begin{bmatrix} -0.6165947 & -0.6481789 \\ -0.5888127 & -0.73439013 \\ -0.52259579 & -0.02718563 \end{bmatrix}$$

(2)

principle Component $\Rightarrow P^T \hat{x}$

$$\hat{x} = \begin{bmatrix} -4.4 & -1.4 & -2.4 & -4.4 & -0.4 & 1.6 & 3.6 & -2.4 & 5.6 & 4.6 \\ -6 & 0 & 4 & 0 & 6 & -4 & 0 & 0 & -3 & 3 \\ -1.8 & 2 & 4.2 & 0.2 & -2.8 & -3.8 & 4.2 & -3.8 & 1.2 & 2.2 \end{bmatrix}$$

Answer =

```
[[ 7.18658682  1.37323947]
 [ 0.75871342 -0.94399334]
 [-3.07034019 -4.45059025]
 [ 2.60849751 -2.97853006]
 [-1.82299166 -4.75401212]
 [ 3.35457763  3.91896138]
 [-4.41464321  2.55604371]
 [ 3.46569126 -1.73131477]
 [-2.31359638  6.03371503]
 [-5.75249521  0.97648096]]
```

c. (3) Reconstruction error if reduced to 2D.(Calculate the L2-norm)

(3) Reconstruction $= PP^T(\hat{x})$ and 平移 $\begin{bmatrix} 54 \\ 8 \\ 48 \end{bmatrix}$

l_2-norm = [2.25104047 , 0.73022597 , 3.18826 , 1.92996 , 4.25088
2.52755 2.13919 2.2384 0.26383 0.99938]

average = 2.04

```
[[ 1.90009072  2.75992709  1.08178971]
 [ 4.29198496  8.24651657  4.37774211]
 [ 4.27485905 13.07633588  6.28310968]
 [ 1.77163801  8.65147726  3.35553912]
 [ 3.29997625 12.56470677  5.62297154]
 [ 5.98934216  3.14672348  3.1538432 ]
 [ 9.85550052  8.72228056  7.17681721]
 [ 2.08893199  7.23080501  2.94160433]
 [10.91848951  4.93118246  6.17370944]
 [ 9.60918683 10.6700449   7.83287366]]
```