# Homework 3 - Image Sentiment Classification

資工系博士班二年級 D06922023 顏志軒

2018 年 12 月 1 日

**Problem 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?

我的 CNN model 使用多層 Conv2D+PReLu，以及兩層 fully-connected layer，最後是 softmax。loss 使用 cross entropy。詳細 model 架構如表 1所示。訓練的過程我使用 ReduceLROnPlateau，動態的藉由 validation set 的 loss 來調整 learning。最後得到的準確度為 0.6445。

**Problem 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?

我的 CNN model 中有 4845063 個參數。我實做如表 2的 DNN，有 4868871 個參數。訓練過程中，調整 learning rate 的方式與原本的 CNN model 相同。從第 30 個 epoch 開始，就一直維持在 training accuracy 大約在 25%，而 validation accuracy 大約在 21%。我判斷這是 under fitting，原因可能是 DNN 無法像 CNN 藉由圖片的局部特徵來判斷，因此準確度差。

**Problem 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]

我用 validation set 做 confusion matrix，結果為:

[ 62 0 14 1 9 8 15] [ 3 4 1 0 0 1 0] [ 10 0 51 1 17 15 8] [ 0 0 2 136 3 4 8] [ 13 1 10 3 63 2 21] [ 5 0 6 2 0 72 3] [ 4 0 8 7 17 6 93]

可發現生氣與恐懼、難過與中立容易搞混，可能是這兩類表情彼此較像的緣故。

**Problem 4.** (1.5%, each 0.5%) CNN time/space complexity:
For a. b. Given a CNN model as

表 1: CNN model 詳細架構

| Layer (type) | Output Shape | Param # |
|---|---|---|
| conv2d_1 (Conv2D) | (None, 44, 44, 64) | 1664 |
| p_re_lu_1 (PReLU) | (None, 44, 44, 64) | 123904 |
| zero_padding2d_1 (ZeroPaddin | (None, 48, 48, 64) | 0 |
| max_pooling2d_1 (MaxPooling2 | (None, 22, 22, 64) | 0 |
| zero_padding2d_2 (ZeroPaddin | (None, 24, 24, 64) | 0 |
| conv2d_2 (Conv2D) | (None, 22, 22, 64) | 36928 |
| p_re_lu_2 (PReLU) | (None, 22, 22, 64) | 30976 |
| zero_padding2d_3 (ZeroPaddin | (None, 24, 24, 64) | 0 |
| conv2d_3 (Conv2D) | (None, 22, 22, 64) | 36928 |
| p_re_lu_3 (PReLU) | (None, 22, 22, 64) | 30976 |
| average_pooling2d_1 (Average | (None, 10, 10, 64) | 0 |
| zero_padding2d_4 (ZeroPaddin | (None, 12, 12, 64) | 0 |
| conv2d_4 (Conv2D) | (None, 10, 10, 128) | 73856 |
| p_re_lu_4 (PReLU) | (None, 10, 10, 128) | 12800 |
| zero_padding2d_5 (ZeroPaddin | (None, 12, 12, 128) | 0 |
| conv2d_5 (Conv2D) | (None, 10, 10, 128) | 147584 |
| p_re_lu_5 (PReLU) | (None, 10, 10, 128) | 12800 |
| zero_padding2d_6 (ZeroPaddin | (None, 12, 12, 128) | 0 |
| average_pooling2d_2 (Average | (None, 5, 5, 128) | 0 |
| flatten_1 (Flatten) | (None, 3200) | 0 |
| dense_1 (Dense) | (None, 1024) | 3277824 |
| p_re_lu_6 (PReLU) | (None, 1024) | 1024 |
| dropout_1 (Dropout) | (None, 1024) | 0 |
| dense_2 (Dense) | (None, 1024) | 1049600 |
| p_re_lu_7 (PReLU) | (None, 1024) | 1024 |
| dropout_2 (Dropout) | (None, 1024) | 0 |
| dense_3 (Dense) | (None, 7) | 7175 |
| activation_1 (Activation) | (None, 7) | 0 |

表 2: DNN model 詳細架構

| Layer (type) | Output Shape | Param # |
|---|---|---|
| dense_1 (Dense) | (None, 48, 48, 64) | 128 |
| p_re_lu_1 (PReLU) | (None, 48, 48, 64) | 147456 |
| dropout_1 (Dropout) | (None, 48, 48, 64) | 0 |
| flatten_1 (Flatten) | (None, 147456) | 0 |
| dense_2 (Dense) | (None, 32) | 4718624 |
| p_re_lu_2 (PReLU) | (None, 32) | 32 |
| dropout_2 (Dropout) | (None, 32) | 0 |
| dense_3 (Dense) | (None, 64) | 2112 |
| p_re_lu_3 (PReLU) | (None, 64) | 64 |
| dropout_3 (Dropout) | (None, 64) | 0 |
| dense_4 (Dense) | (None, 7) | 455 |
| activation_1 (Activation) | (None, 7) | 0 |

```
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: 6*(2*2)*5 = 120

Layer B: Layer A 的 output 有 6 個 channel，故此 layer 有 4*(2*2)*6=96

個參數

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

Layer A: 一個 filter 在 input feature map 上直向和橫向都計算了 3 次 (0-1, 3-4, 6-7)，因此總乘法數為 6*(2*2)*(3*3)*5 = 1080。總加法數則為 6*(3*3)*(2*2*5-1) = 1026。

Layer B: Layer A 的 output 為 3*3*6，因此 Layer B 的一個 filter 在橫向和直向上都只計算了一次 (0-1)，總乘法數為 4*(2*2)*(1*1)*6 = 96，總加法數為 4*(1*1)*(2*2*6-1) = 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 $C_l$, $C_{l-1}$ as lth and l-1th layer)

由於乘法的計算成本遠高於加法，因此 complexity 我只考慮乘法。因為第 i 層的 channel 數為 i-1 層的 filter 數，因此 i 層的 filter 數為 i+1 層的 channel 數。l 層的總乘法數為:

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

**Problem 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) What are the principal axes?

令這 10 個點為 $\mathbf{x_1}, \mathbf{x_2}, \ldots, \mathbf{x_1 0}$。這 10 個點的平均為:

$$\mu = \frac{1}{10}\sum_{n=1}^{10}\mathbf{x_n} = \begin{bmatrix}5.4\\8\\4.8\end{bmatrix} \tag{2}$$

其 covariance matrix 為:

$$\Sigma = \frac{1}{10}\sum_{n=1}^{10}(\mathbf{x_i}-\mu)(\mathbf{x_i}-\mu)^{\mathbf{T}} = \begin{bmatrix}12.04 & 0.5 & 3.28\\0.5 & 12.2 & 2.9\\3.28 & 2.9 & 8.16\end{bmatrix} \tag{3}$$

此矩陣的三個 eigenvalue 由大到小分別為：$\lambda_1 = 15.30, \lambda_2 = 11.63, \lambda_3 = 5.47$。對應的 eigenvector 為

$$\mathbf{u_1} = \begin{bmatrix} -0.62 \\ -0.59 \\ -0.52 \end{bmatrix}, \mathbf{u_2} = \begin{bmatrix} -0.68 \\ 0.73 \\ -0.03 \end{bmatrix}, \mathbf{u_3} = \begin{bmatrix} 0.40 \\ 0.34 \\ -0.85 \end{bmatrix} \tag{4}$$

(b) Compute the principal components for each sample.

每個點的 principal components 即為投影到 eigenvector 的分量，即：

$$(-3.36, 0.71, -1.48)$$
$$(-9.79, 3.03, 0.04)$$
$$(-13.62, 6.53, -2.42)$$
$$(-7.94, 5.06, -1.16)$$
$$(-12.37, 6.84, 5.02)$$
$$(-7.19, -1.84, 3.3)$$
$$(-14.96, -0.47, -1.37)$$
$$(-7.08, 3.81, 3.05)$$
$$(-12.86, -3.95, 0.97)$$
$$(-16.3, 1.11, 1.75)$$

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