

Homework3 Report

資工三 B05902001 廖彥綸

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？

CNN model：

input_shape = (48,48,1)

第一層：Conv kernel_size = (5,5)，channel=64，input_shape = (48,48,1)

第二層：MaxPooling = (2,2)

第三層：Conv kernel_size = (3,3)，channel=512

第四層：MaxPooling = (2,2)

第五層：Conv kernel_size = (3,3)，channel=512

第六層：MaxPooling = (2,2)

第七層：Flatten

第八層：Dense(512)

第九層：Dense(512)

第十層(output)：Dense(7)

每層中間皆放入 PReLU、BatchNormalizaton 和 Dropout

訓練過程：weight_decay=0.00001；optimizers=Adam(lr=0.00025,decay=0.000001)；
利用 ImageDataGenerator 增加資料，共使用水平平移 0.2、垂直平移 0.2 和水平
翻轉，model 的 batch_size = 128，跑 75 個 epochs。

結果：

約 28000 筆訓練資料中切前 2000 筆為 valid_data；valid_data 有 0.644 的準確率。
kaggle 上的 public data 有 0.669 的準確率。

2. 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？

上述的 CNN model 用了約 4100 萬個參數

實作的 DNN model 如下：

第一層：Flatten，input_shape = (48,48,1)

第二層：Dense(512)

第三層：Dense(512)

第四層：Dense(2048)
第五層：Dense(8192)
第六層：Dense(2048)
第七層：Dense(2048)
第八層：Dense(128)
第九層：Dense(128)
第十層(output)：Dense(7)

每層中間皆放入 PReLU、BatchNormalizaton 和 Dropout

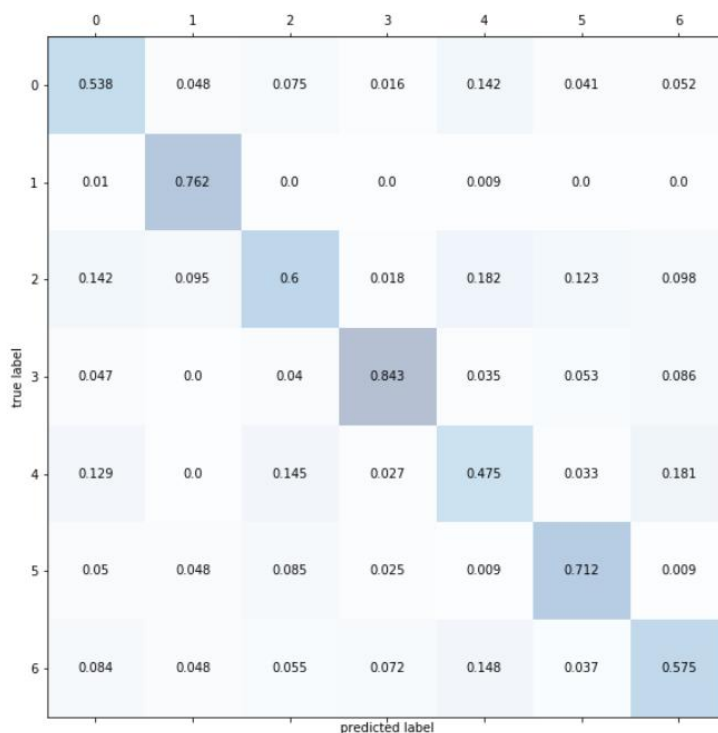
訓練過程：總共使用了約 4000 萬個參數和上述 CNN model 相近。除此之外使用相同的 ImageDataGenerator、weight_decay、optimizers、batch_size 和 epochs 以降低其他變因的影響。

結果：

約 28000 筆訓練資料中切前 2000 筆為 valid_data；valid_data 只有 0.442 的準確率。而 kaggle 上的 public data 只有 0.423 的準確率。

DNN model 相較 CNN model 由於不需要做 convolution 縮短了大量的訓練時間，但正確率也大幅降低，loss function 的下降速度也更緩慢。推論 DNN 較不利於取個圖片的 feature。

3. 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？



0(生氣)

1(厭惡)

2(恐懼)

3(高興)

4(難過)

5(驚訝)

6(中立)

生氣和難過常被互相誤判；同時，難過也容易和恐懼混淆；恐懼常被判斷為生氣或難過；中立也會被誤判為難過。

這次需要判斷的 7 種情緒中有四種偏負面的情緒，包括生氣、厭惡、恐懼、難過，這四種情緒可能有所相關性所以容易互相混淆；唯一的正面情緒：高興，則有最高的辨識率。

4.

(a)

LayerA:

$$kernelsize * inputchannel + 1 = 2 * 2 * 5 + 1 = 21$$

LayerB:

$$kernelsize * inputchannel + 1 = 2 * 2 * 6 + 1 = 25$$

(b)

LayerA: 乘法次數 = $2 * 2 * 5 * 9 = 180$; 加法次數 = $(2 * 2 * 5 - 1) * 9 = 171$

LayerB: 乘法次數 = kernel 涵蓋數 4, 2, 2, 1 = $(2 * 2 + 2 + 2 + 1) * 6 = 54$;

加法次數 = $2 * 2 * 6 - 1 + 2 * 6 - 1 + 2 * 6 - 1 + 6 - 1 = 50$

由於進行 zero-padding 的部分不列入計算。

(c)

考慮做完一個 kernel 的大小時：

至少要做 $k^2 * c_{l-1}$ 次乘法和 $k^2 * c_l - 1 - 1$ 次加法，複雜度約 $O(c_{l-1}k^2)$

需要進行的次數 $\lceil \frac{n-k}{s} \rceil + 1$ 複雜度約 $O(\frac{n-k}{s})$ ，padding 進行時間相對少，可忽略。

最後擴張至 c_l 個 channel。總複雜度為 $O(c_{l-1}c_l k^2 * \frac{n-k}{s})$

5.

$$X = \begin{pmatrix} 1 & 4 & 3 & 1 & 5 & 7 & 9 & 3 & 11 & 10 \\ 2 & 8 & 12 & 8 & 14 & 4 & 8 & 8 & 5 & 11 \\ 3 & 5 & 9 & 5 & 2 & 1 & 9 & 1 & 6 & 7 \end{pmatrix}$$

$$E[X] = \begin{pmatrix} 5.4 \\ 8 \\ 4.8 \end{pmatrix}$$

$$Conv[X] = E[(x - \mu)(x - \mu)^T] =$$

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

$$S = Conv[X]/(n-1) = \begin{pmatrix} 13.3778 & 0.5556 & 3.6444 \\ 0.5556 & 13.5556 & 3.2222 \\ 3.6444 & 3.2222 & 9.0667 \end{pmatrix}$$

$$eigenvalue = (6.0800 \quad 12.9228 \quad 16.9972)$$

對應的列向量：

$$\begin{pmatrix} -0.4111 & -0.6644 & -0.6242 \\ -0.3308 & 0.7468 & -0.5770 \\ 0.8494 & -0.0307 & -0.5268 \end{pmatrix}$$

(a) 使所有點最分散的投影軸，列向量 $-0.6166, -0.5888, -0.5226$

(b)

$$\begin{pmatrix} -0.4111 & -0.6644 & -0.6242 \\ -0.3308 & 0.7468 & -0.5770 \\ 0.8494 & -0.0307 & -0.5268 \end{pmatrix}$$

(c) 投影到平面後損失 $\sqrt{6.0800^2 / (6.0800^2 + 12.9228^2 + 16.9972^2)} = 0.2739$