Homework3 Report Template

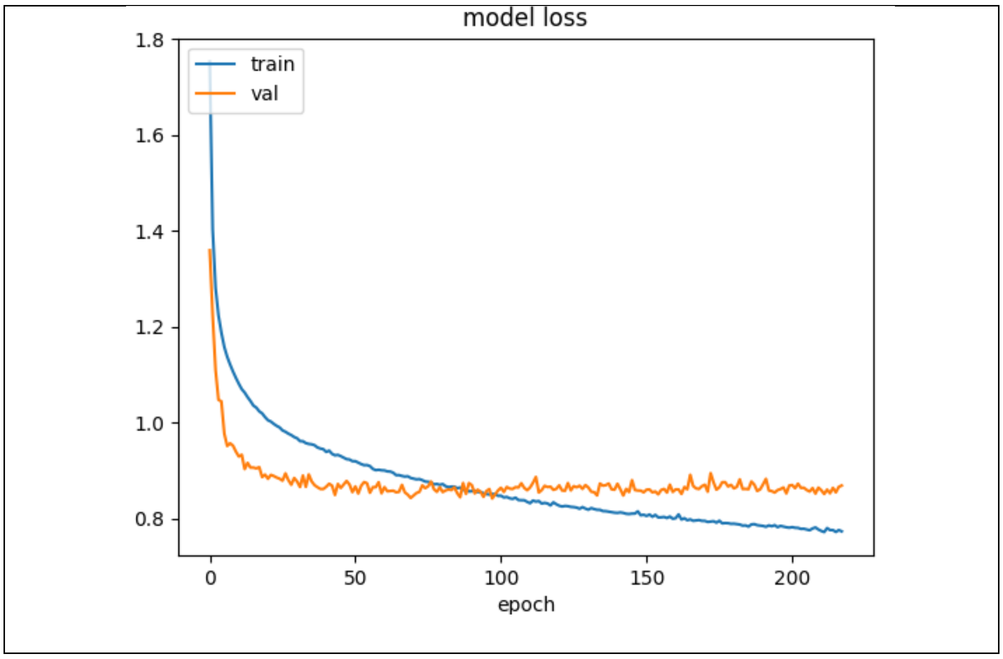
Professor Pei-Yuan Wu
EE5184 - Machine Learning

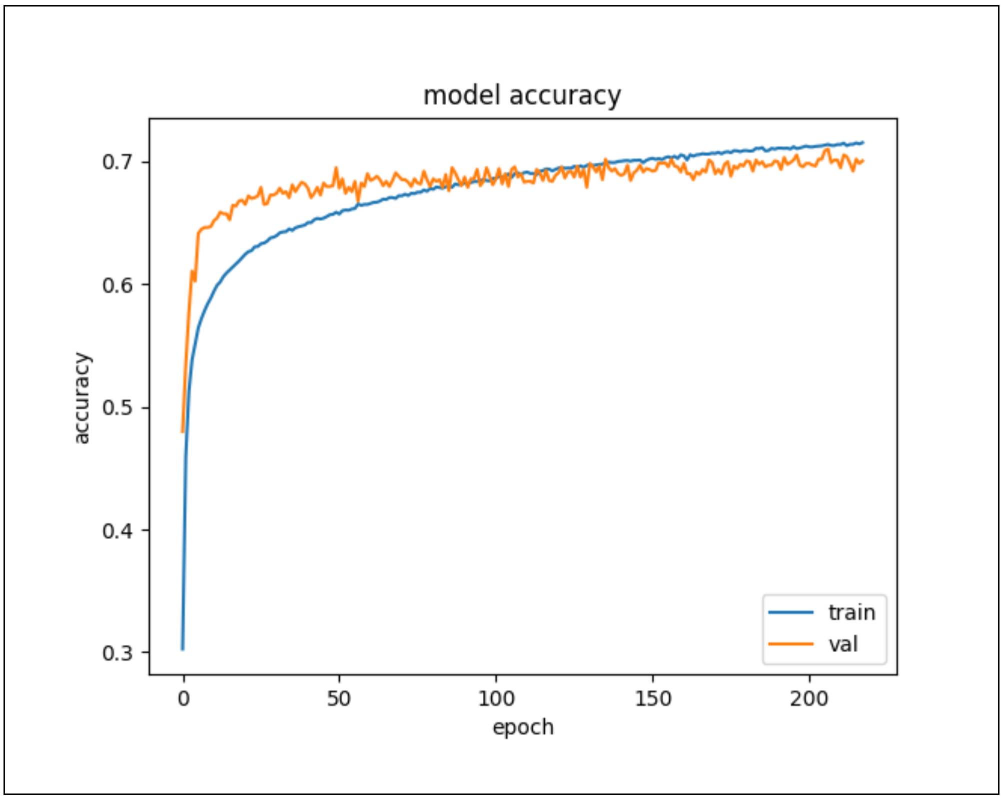
姓名： 詹鈞皓
學號： R06942141

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？

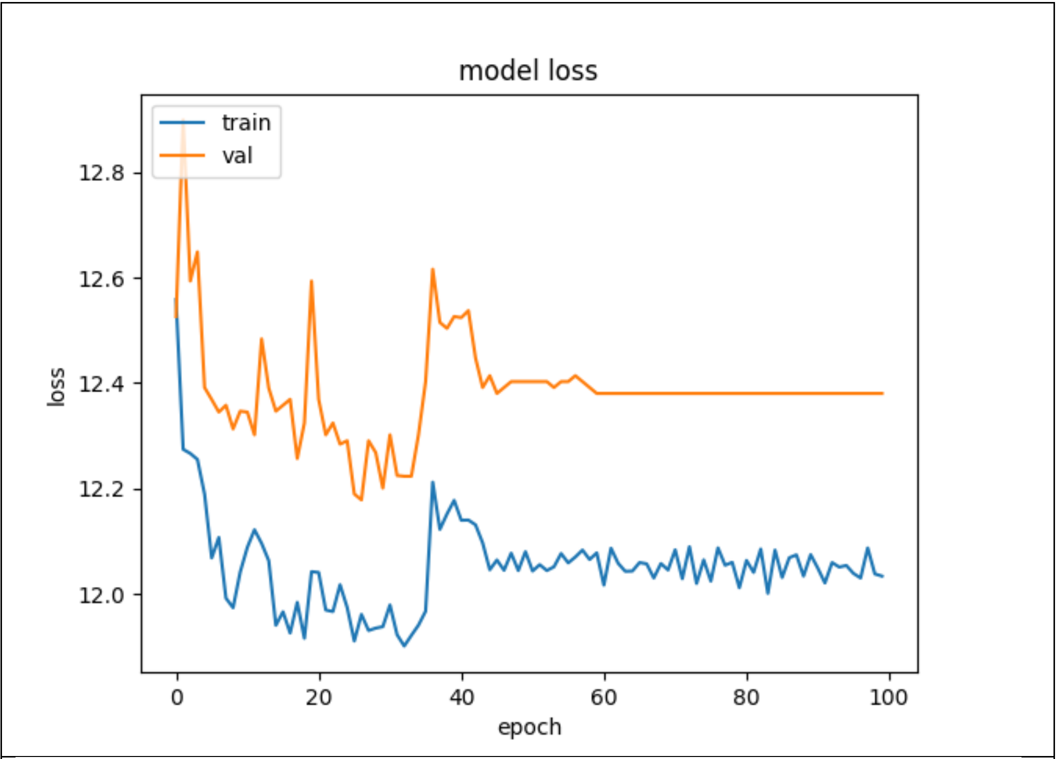
這次作業所實作的 CNN 模型，我參考 Alexnet 的架構所建立出來的，因為有趣參考文獻，發現 Alexnet 相較於 VGG 系列會表現的較好，所以就採用了 Alexnet 的模型。我總共建立 4 層 convolution layer，然後 flatten 之後再經過 2 層 Dense layer，就進入 outpu layer 進行預測，如上圖所示。我的每層 convolution layer 都是使用 leaklyrelu 作為 activation function，dense layer 的部分就只是一般的 relu，然後我每層 layer 都有加上 Batch normalization 和 Dropout，其中 Dropout 的參數是採遞增制，越後面的 layer，Dropout 設的越大，在 model 建立之外，我還有做 data augmentation，增加資料量，並儲存表現較好的 model，使用這些 model 進行 ensemble 再做預測，我的 learn curve 如下圖所示，都有不錯的表現。

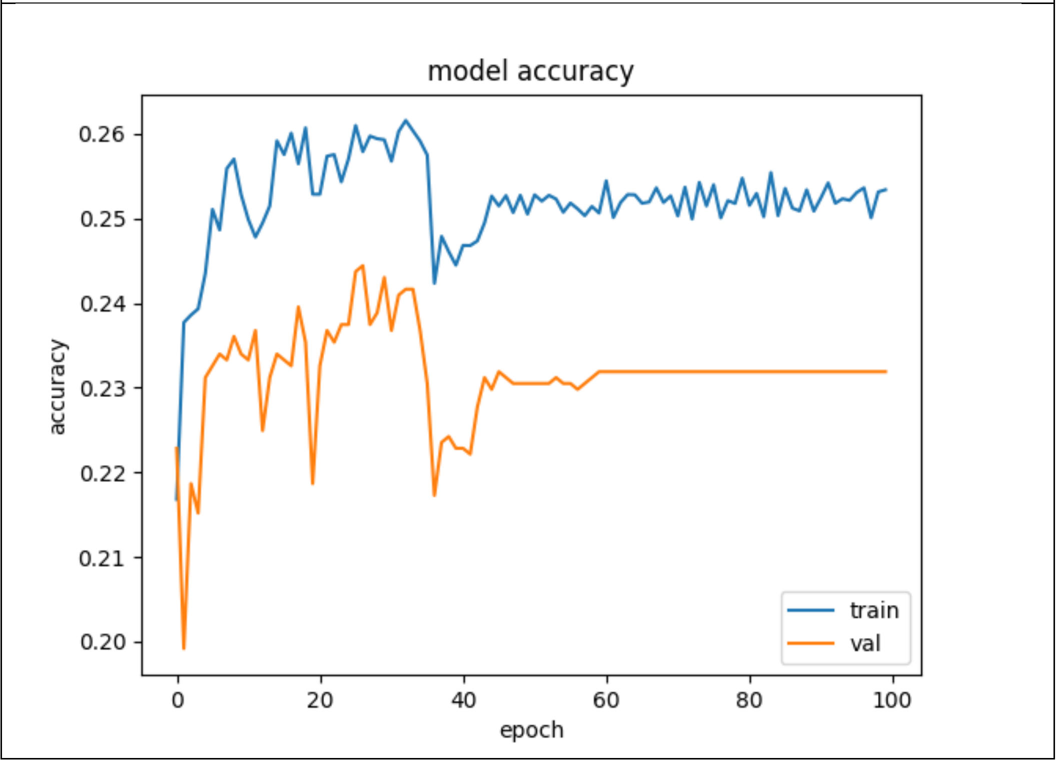




2. **(1%)**承上題，請用與上述 **CNN** 接近的參數量，實做簡單的 **DNN model**，其模型架構、訓練過程和準確率為何？試與上題結果做比較，並說明你觀察到了什麼？

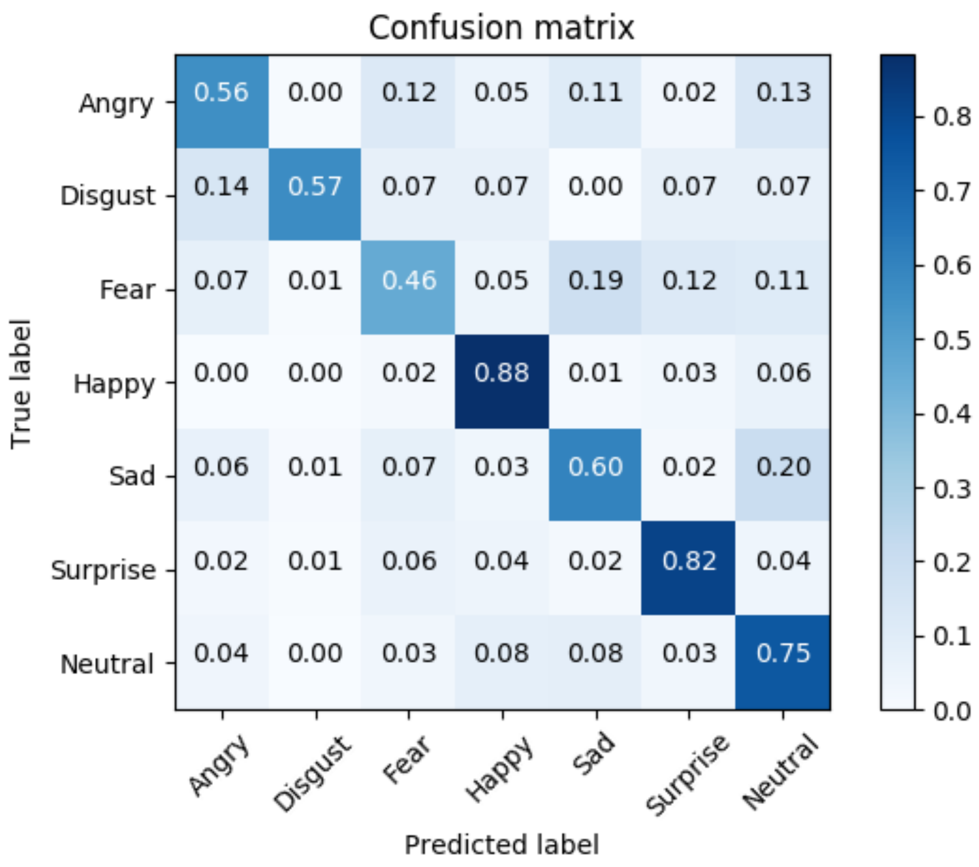
在這一題中，我實作了 3 層的 dense layer，參數數量大約 400 萬個，和第一題的 CNN 差不多數量，每一層的 activation function 都使用 relu，並且會加上 batchnormalization 和 dropout。但是跑出來的結果何用 CNN 跑出來的差了非常多，loss 和 accuracy 如下圖：





我們可以看到 accuracy 才到 0.24 左右而已，效果非常差。

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]

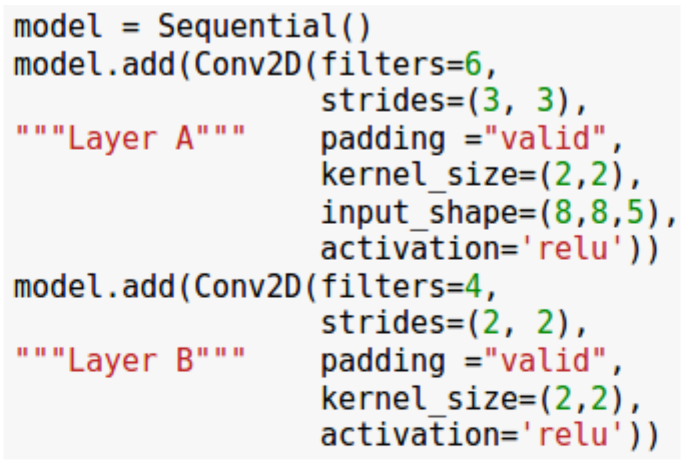


根據上圖 confusion matrix 所示，最容易被分錯的類別是 Fear，其中原因有二，一是因為屬於 Fear class 的資料本來就比較少，所以訓練量不足，導致容易分錯，二是因為 Fear 這個情緒，以人臉的表情來表示時，通常會有很多種不一樣的樣子，每個人在遭受恐懼時會表現出不一樣的反應，反之來說，人在開心時的表情就像對明顯，通常在開心時，臉線就會浮出笑容，所以在分類結果看來，準確率是最高的。而在傷心這個類別中，最容易與中性情緒做搞混，因為傷心的情緒可能不容易表現在臉上，除了很明顯在哭，或很悲傷，不然你臉面無表情，眼神空洞，面容呆滯，都足以代表心中悲傷，所以容易跟中性表情搞混。

----------------Handwritten question----------------

4. (1.5%,each 0.5%)CNN time/space complexity:

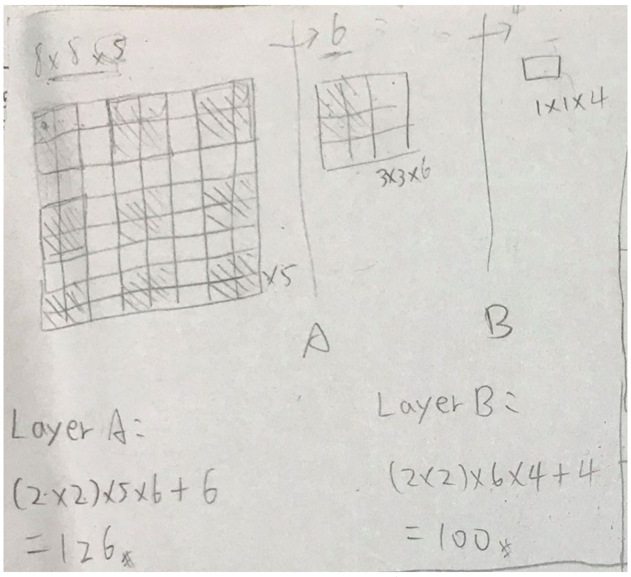
For a. b. Given a CNN model as



```
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);

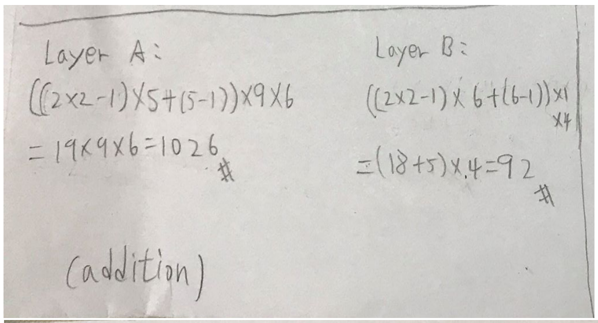
a. How many parameters are there in each layer(Hint: you may consider whether the number of parameter is related with)
Layer A:
Layer B:



Layer A:
$(2\times2)\times5\times6+6$
$=126$*

Layer B:
$(2\times2)\times6\times4+4$
$=100$*

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


Layer A:

$((2\times2-1)\times5+(5-1))\times9\times6$

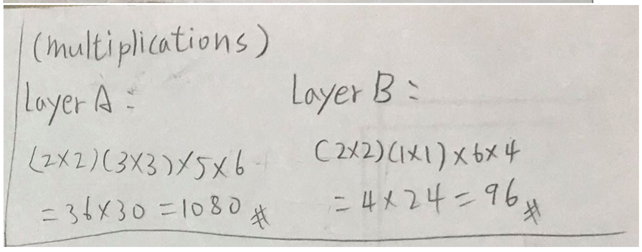
$=19\times9\times6=1026$ #

Layer B:

$((2\times2-1)\times6+(6-1))\times1\times4$

$=(18+5)\times4=92$ #

(addition)


(multiplications)

Layer A:

$(2\times2)(3\times3)\times5\times6$

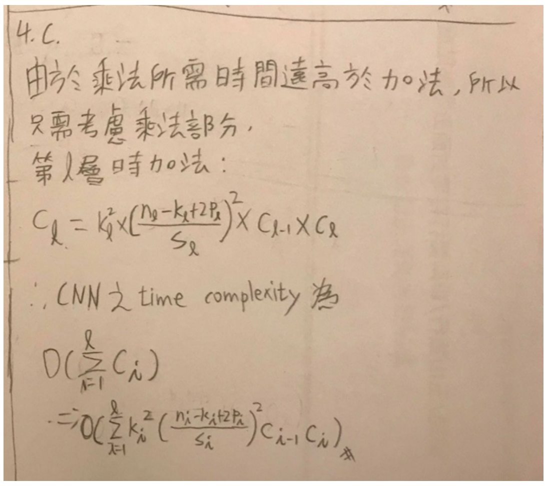
$=36\times30=1080$ #

Layer B:

$(2\times2)(1\times1)\times6\times4$

$=4\times24=96$ #

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 □ □,□ □−1 as lth and l-1th layer)


4.C.

由於乘法所需時間遠高於加法，所以只需考慮乘法部分，

第l層時加法：

$C_l = k_l^2 \times \left(\frac{n_l-k_l+2P_l}{S_l}\right)^2 \times C_{l-1} \times C_l$

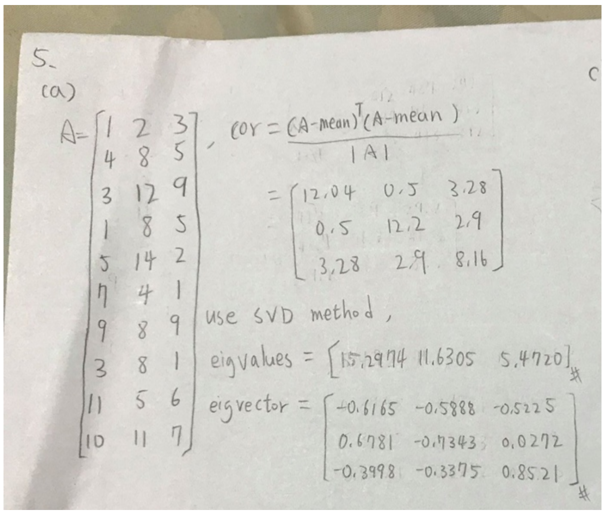
∴ CNN之time complexity為

$O\left(\sum_{i=1}^{l} C_i\right)$

$= O\left(\sum_{i=1}^{l} k_i^2 \left(\frac{n_i-k_i+2P_i}{S_i}\right)^2 C_{i-1} C_i\right)$ #

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?



5.

(a)

$$A=\begin{bmatrix}1&2&3\\4&8&5\\3&12&9\\1&8&5\\5&14&2\\7&4&1\\9&8&9\\3&8&1\\11&5&6\\10&11&7\end{bmatrix},\quad cor=\frac{(A-mean)^T(A-mean)}{|A|}$$

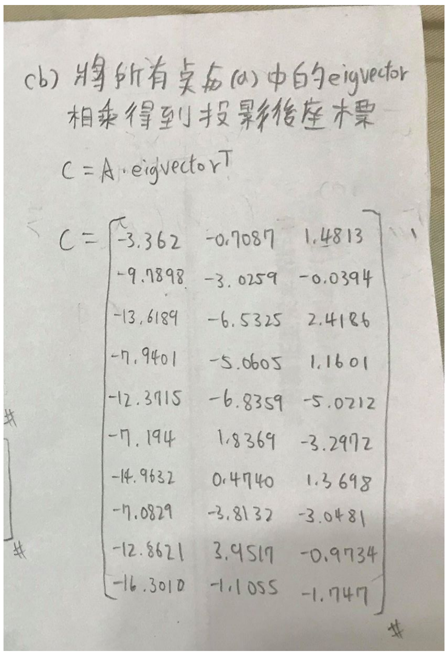
$$=\begin{bmatrix}12.04&0.5&3.28\\0.5&12.2&2.9\\3.28&2.9&8.16\end{bmatrix}$$

use SVD method,

$$\text{eigvalues}=[15.2974\ \ 11.6305\ \ 5.4720]\ \#$$

$$\text{eigvector}=\begin{bmatrix}-0.6165&-0.5888&-0.5225\\0.6781&-0.7343&0.0272\\-0.3998&-0.3375&0.8521\end{bmatrix}\ \#$$

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



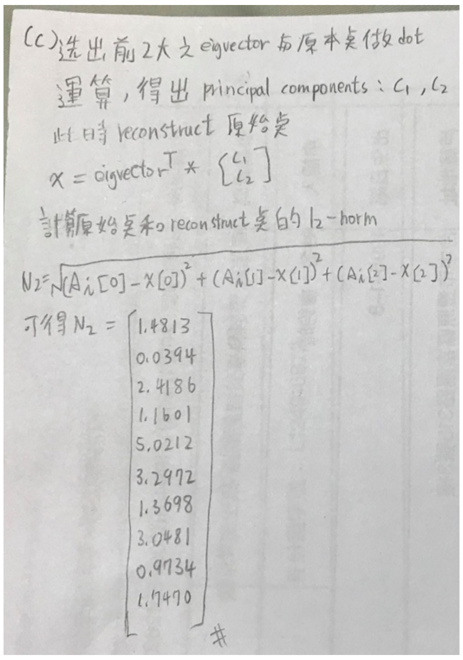
(b) 將所有点与(a)中的eigvector
相乘得到投影後座標

$C=A\cdot \text{eigvector}^T$

$$C=\begin{bmatrix}-3.362&-0.7087&1.4813\\-9.7898&-3.0259&-0.0394\\-13.6189&-6.5325&2.4186\\-7.9401&-5.0605&1.1601\\-12.3715&-6.8359&-5.0212\\-7.194&1.8369&-3.2972\\-14.9632&0.4740&1.3698\\-7.0829&-3.8132&-3.0481\\-12.8621&3.9517&-0.9734\\-16.3010&-1.1055&-1.747\end{bmatrix}\ \#$$

6.

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



(c) 選出前2大之 eigvector 與原本點做 dot 運算，得出 principal components：c_1，c_2

此時 reconstruct 原始點

$$x = \text{eigvector}^T * \begin{bmatrix} c_1 \\ c_2 \end{bmatrix}$$

計算原始點和 reconstruct 點的 l_2-norm

$$N_2 = \sqrt{(A_i[0]-X[0])^2 + (A_i[1]-X[1])^2 + (A_i[2]-X[2])^2}$$

可得 $N_2 = \begin{bmatrix} 1.4813 \\ 0.0394 \\ 2.4186 \\ 1.1601 \\ 5.0212 \\ 3.2972 \\ 1.3698 \\ 3.0481 \\ 0.9734 \\ 1.7470 \end{bmatrix}$ #