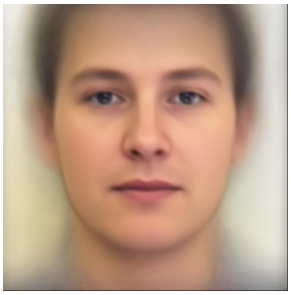
ML Homework Report 6

學號：B04902092 系級： 資工三 姓名：張均銘

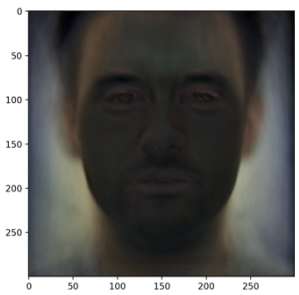
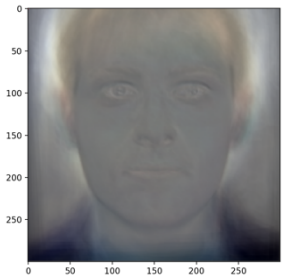
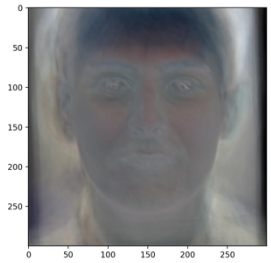
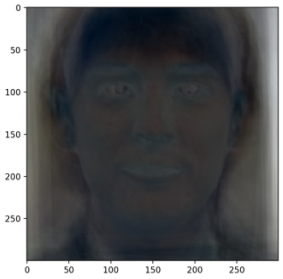
A. PCA of colored faces

請畫出所有臉的平均。



畫出前四個 Eigenfaces 就是對應前四大 Eigenvalues 的 Eigenvectors。

因電腦 RAM 不夠，所以 resize 成(300,300,3)

請從數據集中挑出任意四個圖片，並用前四大 Eigenfaces 進行 reconstruction，並畫出結果。

取圖 50,100,150,200 四張圖來 reconstruct

請寫出前四大 Eigenfaces 各自所佔的比重 (explained variance ratio)，請四捨五入到小數點後一位。

個別值：[0.04193883, 0.02975112, 0.02402014, 0.0222198]

第一個:4.2%, 第二個:3.0%

第三個：2.2% 第四個:2.4%

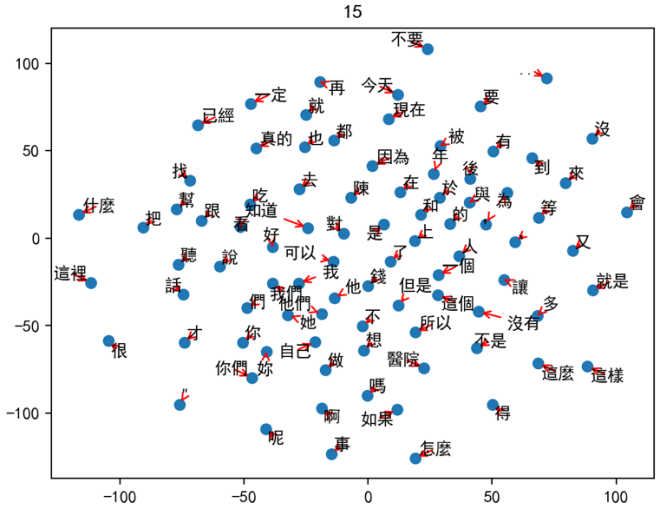
B. Visualization of Chinese word embedding

使用套件:

使用 gensim 的 word2vec model 來建造 model，wordCount = 6000，代表在這個 data 中，我只對出現次數超過 6000 次的 data 訓練，

vector size =100，即每個出來的單字都是 100 維的向量，並用 TSNE 降到 2 維，兩個維度分別就是這個詞的 X,Y 軸，再把他 plot 到圖上會使用 wordCount=6000 的原因是 3000 的話出現字詞太多，在 plot 的結果上效果非常不好，所以改用 wordCount6000 來做圖。

Word embedding 圖:



觀察:

從圖中可以發現，“你們” “我們” 這些代名詞被歸類在同一區，“今天” “現在”也在附近、“這麼” ” 這樣”也是在同一區。因為性質相近，但我覺得使用 skipgram 的方式 train 這些效果會更加明顯(genism default 是 CBOW)，因為 data 本身就比較像是要從一段話推出上下文，所以比　較適合 skipgram 的 train 法，

C. Image clustering

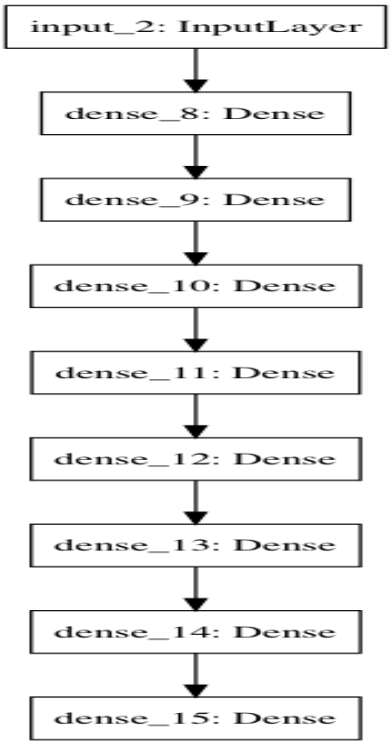
Autoencode 和 TruncatedSVD 的比較:

在做分類的過程兩個 model 最後再降完維後都用 Kmeans 來進行分類，兩者在降維部分都是降到 40 維在做 Kmeans。

以下是 autoencoder 的 model 架構

PCA 的部分則是用 TruncatedSVD 的方式降維，他是 sklearn 裡面的一個套件，做的事情和 PCA 差不多，只是他花較少的記憶體做到，他是直接在矩陣上運算，不用像 PCA 求出 covariance matrix 所以花費記憶體較少。

Autoencode model:



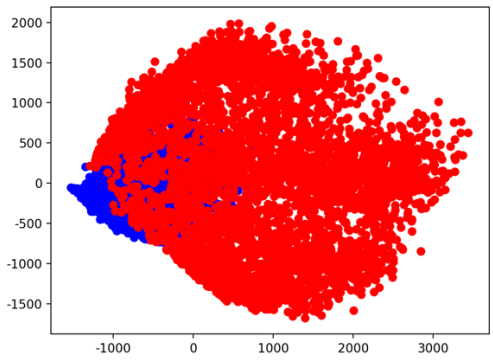
兩者在 **kaggle** 上分數比較**:**

TruncatedSVD public: 0.02945 private:0.02925

Autoencode public: 0.98796 private:0.98815

visualization.npy 原 data 二維圖

紅色表示前 5000 張圖片，藍色表示後 5000 張圖片



預測與實際 label 差距:

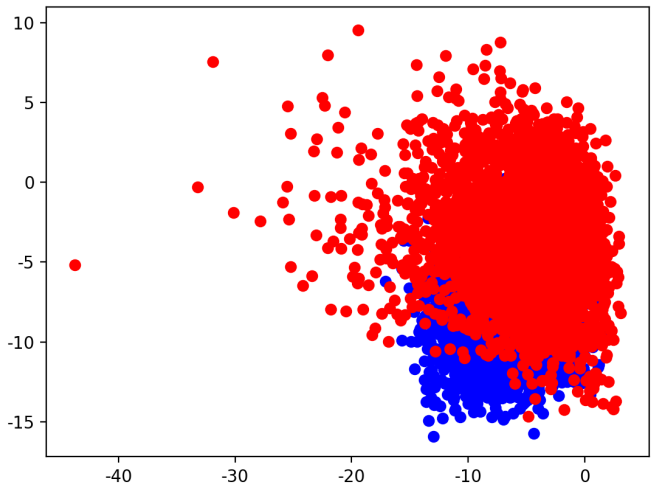
預測錯誤數量：

前五千筆：0

5001~10000:46

Autoencode 預測 label 二維圖：

紅色代表預測維前 5000 張，藍色預測為後 5000 張

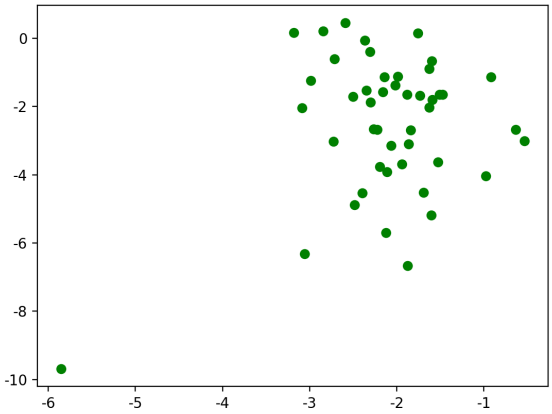


預測錯誤原因：

由未 train 過及 train 過的 data 放在二維圖上的比較，明顯看出有部分的 data 在 二維圖上是有重疊的，估計就是這些重疊的部分造成分類的錯誤，因為無法正確確定他在哪一類所以導致判斷錯誤。實際上把錯誤的點標注在二維圖上，確實也這些點是在重疊的地方。

PCA 預測錯誤的圖:

以下是 PCA 預測錯誤的點的位置



發現：

明顯發現，經過 normalized 及 autoencode 出來的圖較集中，而原本 data 分布非常廣可以到 5000 多。