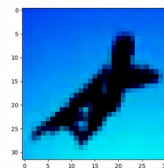
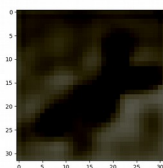
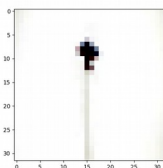
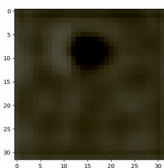


1. (2%) 任取一個 baseline model (sample code 裡定義的 fcn, cnn, vae) 與你在 kaggle leaderboard 上表現最好的單純 autoencoder 架構的 model (如果表現最好的 model 就是 sample code 裡定義的 model 的話就再任選一個, e.g. 如果 cnn 最好那就再選 fcn), 對各自重建的 testing data 的 image 中選出與原圖 mse 最大的兩張加上最小的兩張並畫出來。(假設有五張圖, 每張圖經由 autoencoder A 重建的圖片與原圖的 MSE 分別為 [25.4, 33.6, 15, 39, 54.8], 則 MSE 最大的兩張是圖 4、5 而最小的是圖 1、3)。須同時附上原圖與經 autoencoder 重建的圖片。(圖片總數: (原圖 + 重建)\*(兩顆 model)\*(mse 最大兩張 + mse 最小兩張) = 16 張)

1. 以下照片左側為 mse 後, 右側為 origin:best model(cnn)

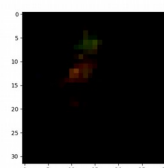
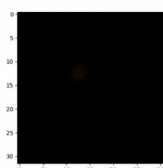
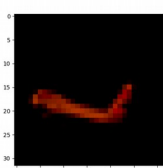
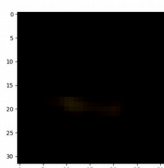
1. mse=46.734

2. mse=45.848



3. mse=5.688

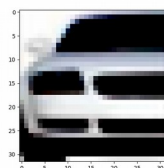
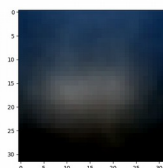
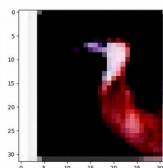
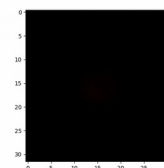
4. mse = 6.292



2. 以下照片左側為 mse 後, 右側為 origin:another model(fcn)

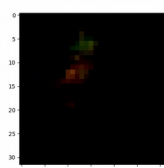
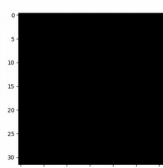
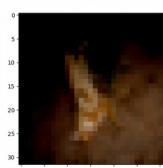
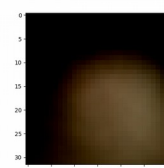
1. mse=40.687

2. mse=38.763



3. mse=5.445

4. mse = 5.457



2. (1%) 嘗試把 sample code 中的 K-means 與 PCA 分別做在 autoencoder 的 encoder output 上，並回報兩者的 auc score 以及本來 model 的 auc。autoencoder 不限。不論分數與本來的 model 相比有上升還是下降，請同學簡述原因。

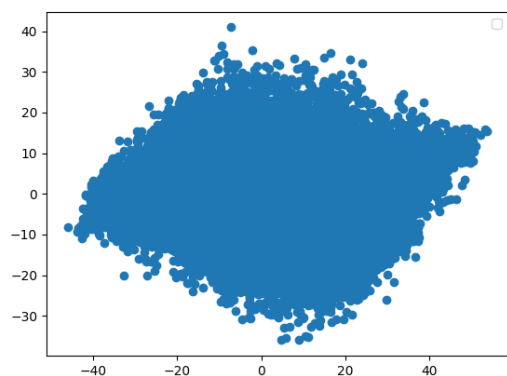
<a href="#">prediction_cnn_knn2.csv</a> an hour ago by <a href="#">b06502158_</a> <a href="#">add submission details</a>	0.55004	<input type="checkbox"/>
<a href="#">prediction_nothing.csv</a> 2 hours ago by <a href="#">b06502158_</a> <a href="#">add submission details</a>	0.62655	<input checked="" type="checkbox"/>

將 best model (cnn) 的 output 直接丟入 kaggle 後，得到的分數為 0.62。將 best model (cnn) 的 output 經過 knn=3(上面顯示 2 是因為從 0 開始排序)丟入 kaggle，得到的分數為 0.55。會下降如此多，猜測原因可能是因為，cluster 的數字和 cnn 所分類出的數字不相同。

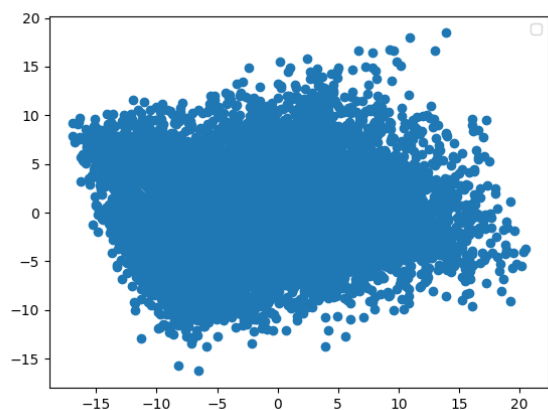
<a href="#">prediction.csv</a> a day ago by <a href="#">b06502158_</a> <a href="#">add submission details</a>	0.59127	<input type="checkbox"/>
<a href="#">prediction2.csv</a> a day ago by <a href="#">b06502158_</a> <a href="#">add submission details</a>	0.60645	<input type="checkbox"/>

將 another model (fcf) 的 output 直接丟入 kaggle 後，得到的分數為 0.59。將 best model (cnn) 的 output 經過 cluster n=3(上面顯示 2 是因為從 0 開始排序)丟入 kaggle，得到的分數為 0.60。這次上升，讓 knn 進行 fit transform，確實應該讓 prediction 上升，加上 cluster n=3 恰巧符合 fcf 的 output，所以有明顯的提升。

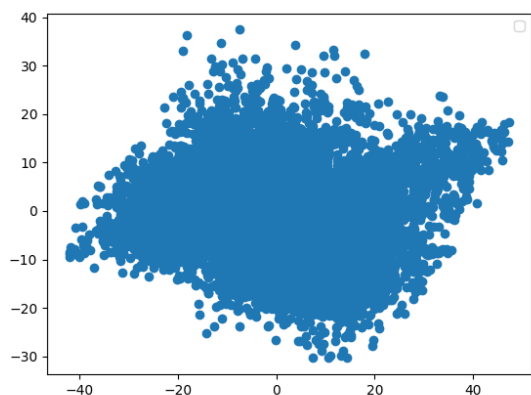
3. (1%) 如 hw9，使用 PCA 或 T-sne 將 testing data 投影在 2 維平面上，並將 testing data 經第 1 題的兩顆 model 的 encoder 降維後的 output 投影在 2 維平面上，觀察經 encoder 降維後是否分成兩群的情況更明顯。（因未給定 testing label，所以點不須著色）



左圖為 origin，也就是直接將 testing data 投影到二維，經由 PCA 降至二維後，基本上無法分出兩群。



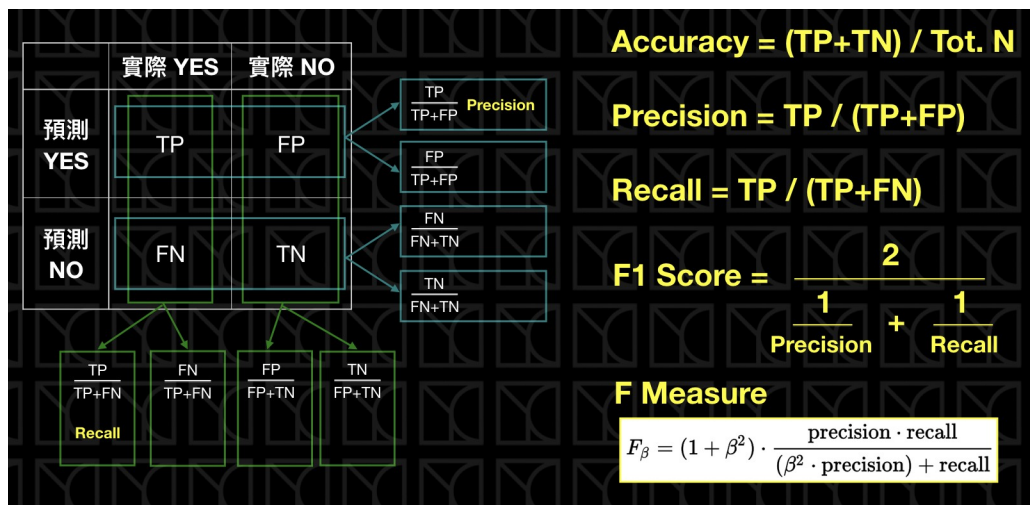
左圖為 cnn model，也就是將 testing data 丟入 cnn 的 autoencoder 再投影到二維，經由 PCA 降至二維後，基本上仍無法分出兩群。但觀察值的發散度後，發現相較之前，分的更近了。這也可能就是我一直無法突破 strong base line 的原因，autoencoder 或許太強了。



左圖為 fcn model，也就是將 testing data 丟入 fcn 的 autoencoder 再投影到二維，經由 PCA 降至二維後，基本上仍無法分出兩群。但觀察斜率約等於一的地方，有些許的分成兩邊。

4. (2%) 說明為何使用 auc score 來衡量而非 binary classification 常用的 f1 score。如果使用 f1 score 會有什麼不便之處？

因為 f1 score 無法衡量 Precision（準確率）和 Recall（召回率）誰帶來的影響較大，因其 Precision 和 Recall 都同等重要。Precision 和 Recall 同時關注的都是 True Positive（都在分子），但是角度不一樣，Precision 看的是在預測正向的情形下，實際的「精準度」是多少，而 Recall 則是看在實際情形為正向的狀況下，預測「能召回多少」實際正向的答案。



而如果加入了 auc score，則我們就能自定義，你所在意的角度，並增加其 penalty/cost，其運算出的 score 越貼近你所期待的結果。