

1. (3%) 請至少使用兩種方法 (autoencoder 架構、optimizer、data preprocessing、後續降維方法、clustering 算法等等) 來改進 baseline code 的 accuracy。

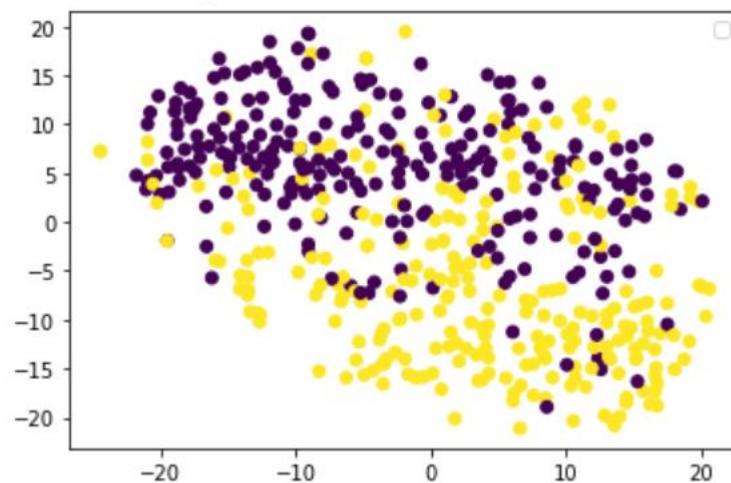
- a. 分別記錄改進前、後的 test accuracy 為多少。

改進前 test accuracy : 0.74941

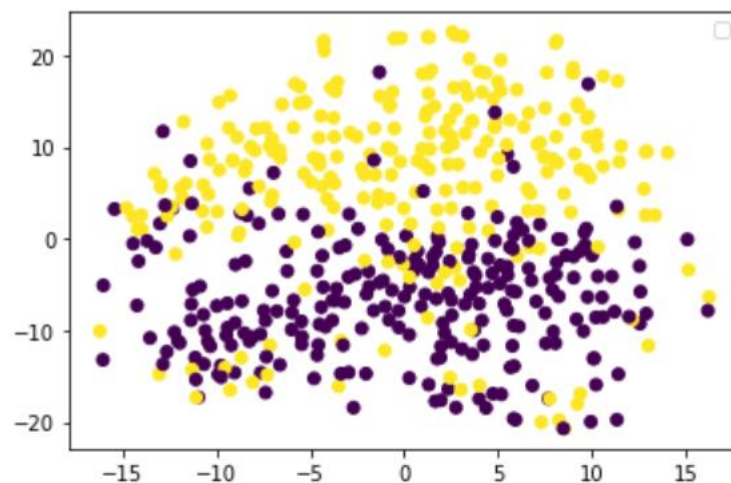
改進後 test accuracy : 0.79105

- b. 分別使用改進前、後的方法，將 **val data** 的降維結果 (embedding) 與他們對應的 label 畫出來。

改進前：(Validation accuracy : 0.714)



改進後：(Validation accuracy : 0.84)



- c. 盡量詳細說明你做了哪些改進。

Auto encoder：在原先的 model 裡使用多一層 CNN，並且在 convolution 後都有做 batch normalization，另外使用 Leaky ReLU 當作 activation function。一張圖片在經過 CNN encode 成(256, 2, 2)的 vector 後，我將 vector 拉平再通過一個 Linear 層將其轉為 512 維的 vector 當作 latent vector，decode 過程則是將 latent vector 通過 Linear 層轉為 1024 維，最後將 convolution 反過來做一次。

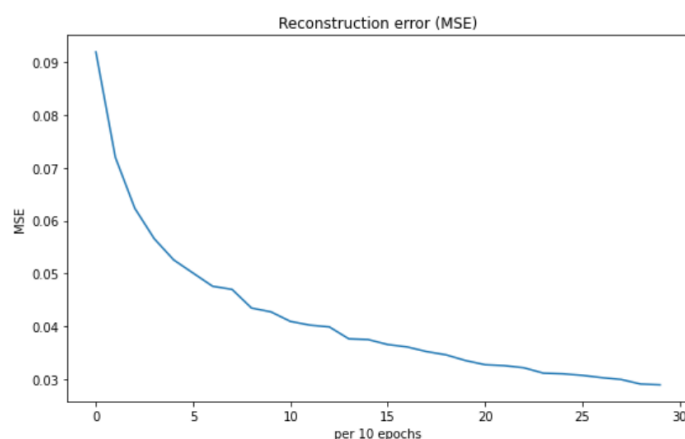
Training 過程：在一開始 training 時有發現 loss 基本上會一直上上下下震盪，但是感覺整體有在慢慢變低，因此將 epoch 數調高至 300，learning rate 則維持 0.00001，並使用 Adam 當作 optimizer。

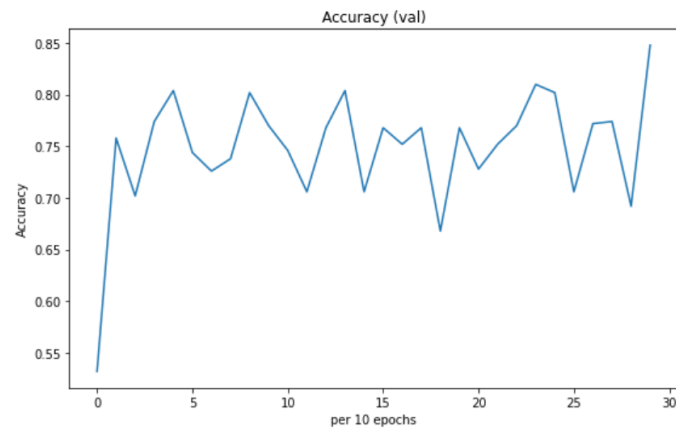
Dimension reduction：因為 auto encoder 最後將圖片 encode 成 512 維的 vector，因此我第一次降維使用 Kernel PCA 搭配 rbf 的 kernel 將其降到 100 維，接著透過 TSNE 降成可視化的 2 維。

2. (1%) 使用你 test accuracy 最高的 autoencoder，從 trainX 中，取出 index 1, 2, 3, 6, 7, 9 這 6 張圖片
- a. 畫出他們的原圖以及 reconstruct 之後的圖片。



3. (2%) 在 autoencoder 的訓練過程中，至少挑選 10 個 checkpoints
- a. 請用 model 的 train reconstruction error (用所有的 trainX 計算 MSE) 和 **val accuracy** 對那些 checkpoints 作圖。





b. 簡單說明你觀察到的現象。

從以上可以發現 **MSE** 隨著 **epoch** 越大有持續的變小，也符合我前面的改進，必須將 **epoch** 的數量提高，才能讓 **auto encoder** 的 **performance** 更好，另外根據 **validation accuracy** 來看會發現會一直上下震盪，不過好的 **performance** 甚至會到 0.8 以上，因此在選取 **checkpoint** 必須注意不要挑到剛好 **accuracy** 比較低的參數。