

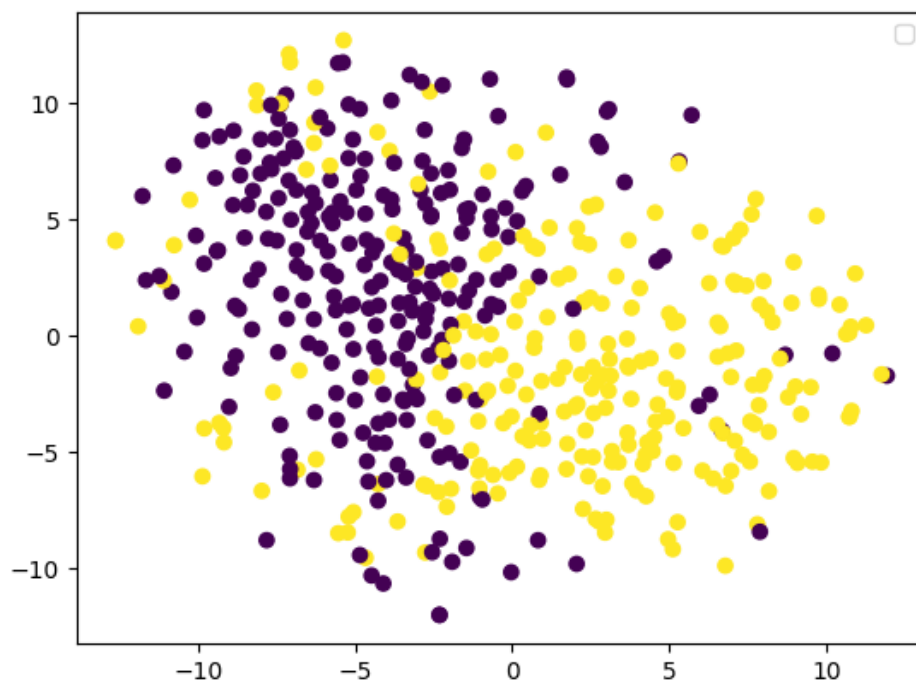
Machine Learning 2020 Spring - HW9 Report

學號: b07902064 系級: 資工二 姓名: 蔡銘軒

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。
 - 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。
 - c. 盡量詳細說明你做了哪些改進。

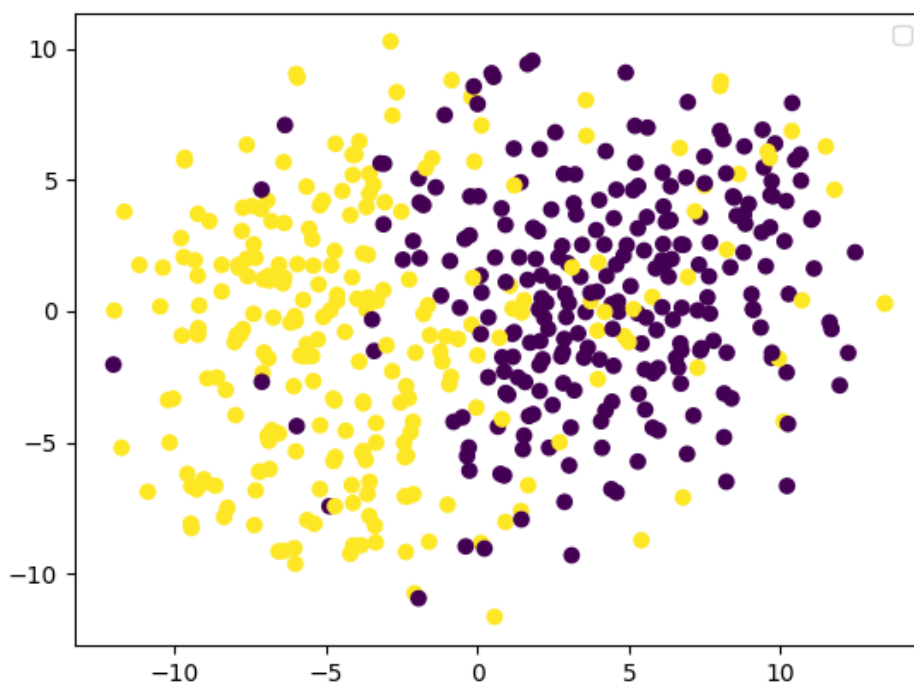
改進前：

validation accuracy: 0.78



改進後：

validation accuracy: 0.84



所做的更改：

我首先從助教提供的Sample code出發，主要針對兩個部分做修改。

1. autoencoder: 我做了許多不同的嘗試。比較小的例如修改Sample code裡convolutional layer的channel數量(增加或減少皆有嘗試)，或者是在convolutional layer之後加上1~2層FC，增加encoder/decoder的複雜度以及能力；比較大的修改有重新設計自己的架構，疊出更深的convolutional layer以及FC，使用stride = 2來取代MaxPool等。前前後後嘗試了許多修改，表現卻始終沒有比Sample code進步多少(比較好的也只有0.78左右)，因此這部分最後還是回到最初的狀態，與Sample code相同。
2. cluster: 這部分也有許多可以調整的。我先嘗試不同的組合，例如 encoder的輸出直接做Kmeans、PCA+Kmeans、PCA+TSNE+Kmeans、TSNE+Kmeans。這邊我是隨著autoencoder的架構跟著調整，有些架構下沒有做PCA直接做TSNE+Kmeans表現會更好，有些情況下直接做Kmeans表現最好等等。另外在不同組合下，調整 $n_components$ 也對結果有滿大的影響。在PCA的部分Sample code的設定是200，我把它降低為40後在Baseline code裡accuracy提高到0.79。在Best model裡則設為100，達到0.84的validation accuracy。我也嘗試了很多其他更大或更小的數值，都沒有比設為100時更好。TSNE的這個參數因為不能設為4以上，我就沒有做其他嘗試，直接設為2。另外Kmeans的 n_init 參數也有小幅影響結果，我實驗幾個數值後發現設為64有使accuracy上升約0.5%。

其餘的調整包含:

1. optimizer: 我嘗試過SGD，結果收斂效果不佳，因此最後決定使用Adam。但即使都是使用Adam，learning rate的影響也滿大的。在Baseline model我使用 10^{-3} ，而在Best model則使用 $2 \cdot 10^{-4}$ 。我認為這裡用小一點的learning rate表現比較好。
2. Normalization/Loss function: 我有觀察到有些autoencoder會使用binary cross entropy 來作為loss function。為了實驗這個作法，我把normalization的方式改為直接除以255，使數值落在0-1之間，然後將decoder的最後一層從Tanh改成Sigmoid，同樣是為了將數值壓到0-1區間。但這樣的更改沒有帶給我明顯的進步，因此最後沒有採用。
3. epoch: 訓練autoencoder時，在某些架構下，我發現多train幾輪會得到更好的結果。例如有些model在epoch=120時給我最好的表現，稍微多一點或少一點都使結果大幅降低。在Best model

裡，我實驗的結果是100個epoch最佳。

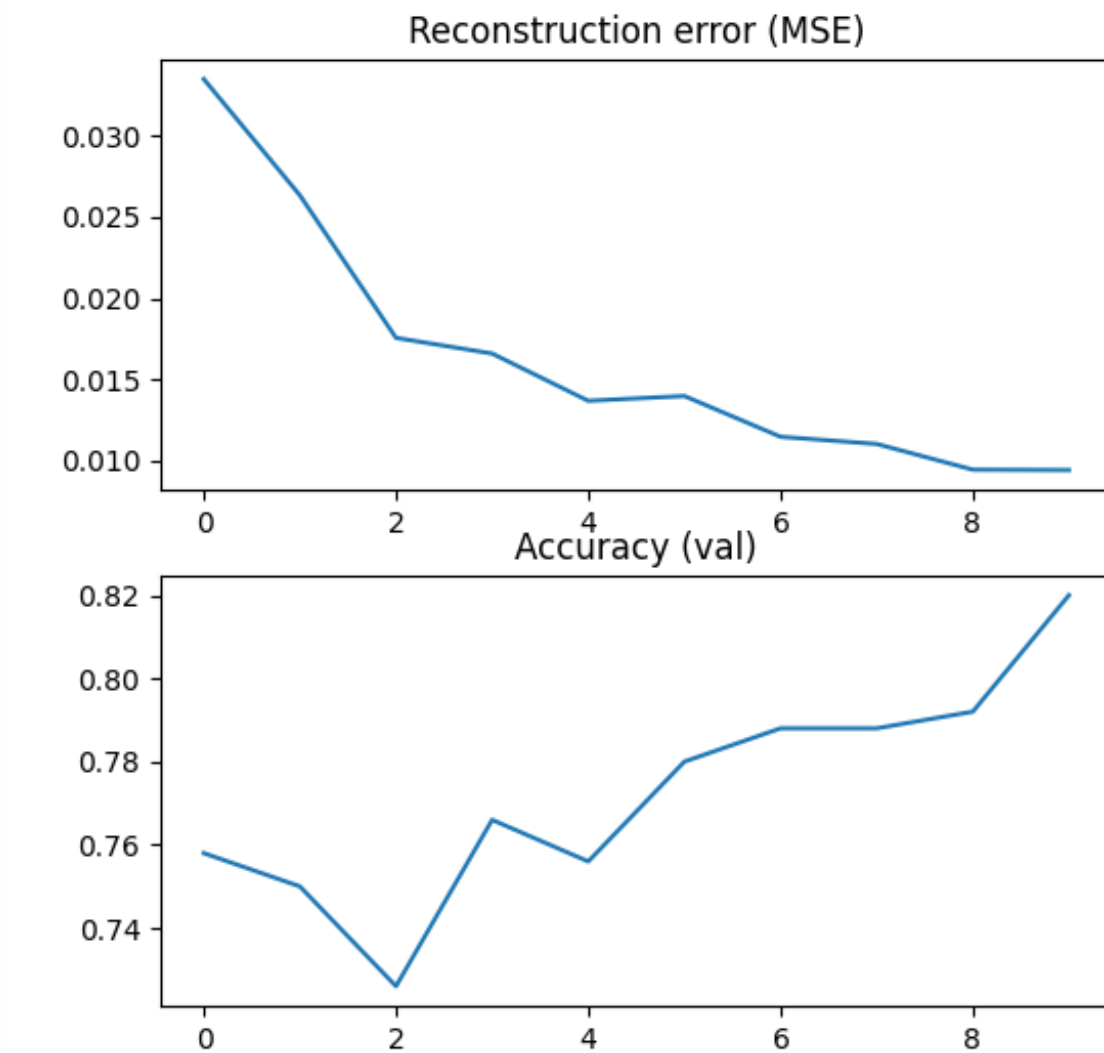
整體而言，Baseline跟Best的差別在於optimizer的learning rate以及PCA與Kmeans的參數修改。我認為這一次的作業對我來說特別難處理，任何參數做一點小修改，例如多train一點epoch，正確率都可以有非常大的變化(5%~10%不等)。前前後後做了幾十次不同修改與測試，還是沒有明確的結論應該怎麼做會最好。而最後Best model其實也跟Sample code相差不多。

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 張圖片畫出他們的原圖以及 reconstruct 之後的圖片。

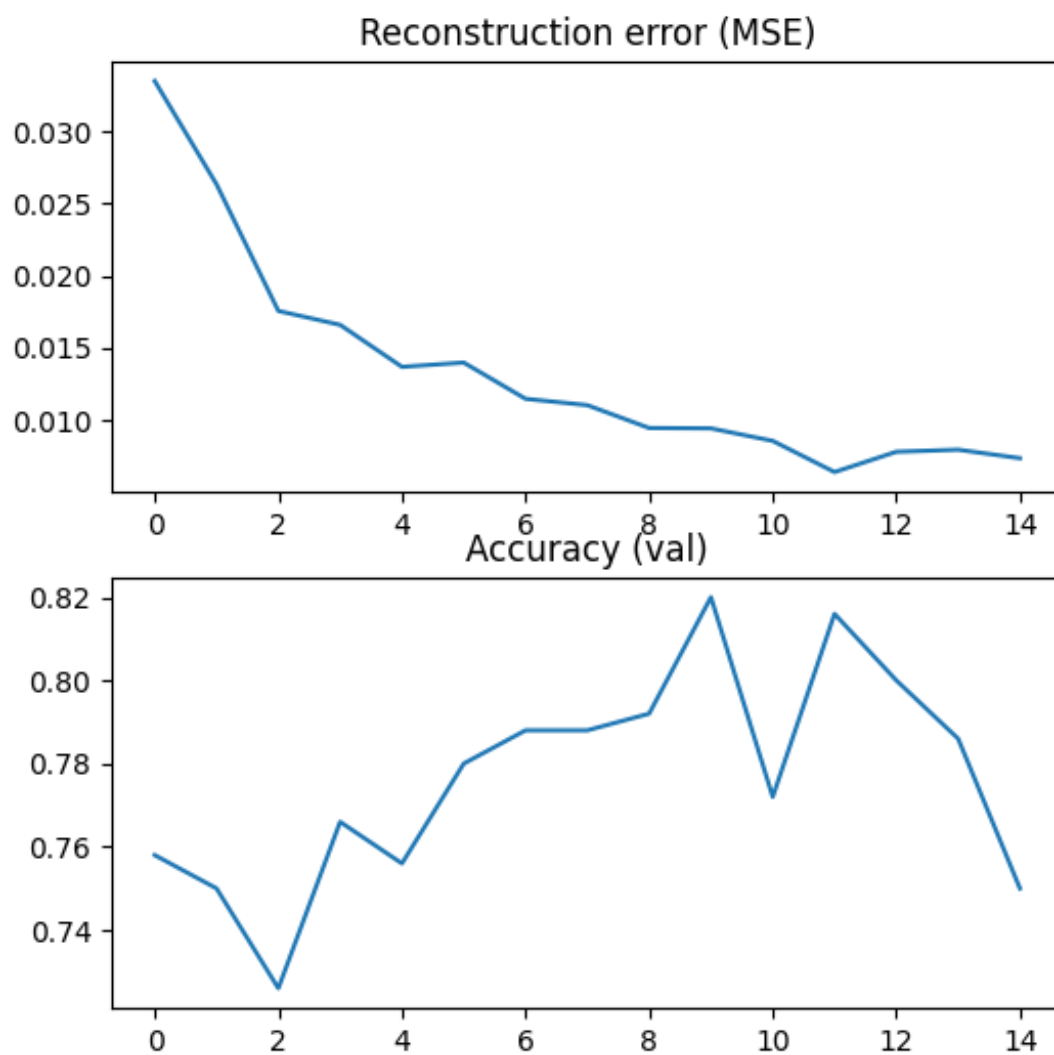


可以看到還原的圖片比較模糊，細節的部分也會失真，但整體而言還是跟原圖有不錯的相似度，還是可以辨認出圖中的物體。

3. (2%) 在 autoencoder 的訓練過程中，至少挑選 10 個 checkpoints
- 請用 model 的 train reconstruction error (用所有的 trainX 計算 MSE) 和 **val accuracy** 對那些 checkpoints 作圖。
 - 簡單說明你觀察到的現象。



圖片裡每10的epoch紀錄一次，大致可以看出隨著Loss的降低，Accuracy呈現上升的趨勢。在epoch=100時，Accuracy來到最高點。但我認為分類的Accuracy跟Loss沒有完全正相關。我Best model裡使用的autoencoder，Loss大概降到0.07左右就無法更低了。我有使用其他結構比較深，能力比較強的autoencoder，Loss可以做到0.002以下，但分類的正確率卻只有60%左右。另外我也嘗試與圖片裡相同的model進行150個epoch，結果正確率在epoch=100後就開始下降了，結果如下圖。



我認為除了autoencoder以外，PCA、TSNE、kmeans怎麼配合也影響非常大。在這次的例子裡，我運氣不錯有找到一個Accuracy突然很高的點。