

學號：R08946006 系級：資料科學學程碩一 姓名：周逸平

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。

嘗試了兩種不同的 autoencoder 架構，以及改變了 clustering 的算法。

由於發現將 MiniBatchKMeans 改為 KMeans 後效果較佳，故以下表格只有改動 autoencoder 並做比較。關於降維與分群參數如下：

Sklearn_seed = 0

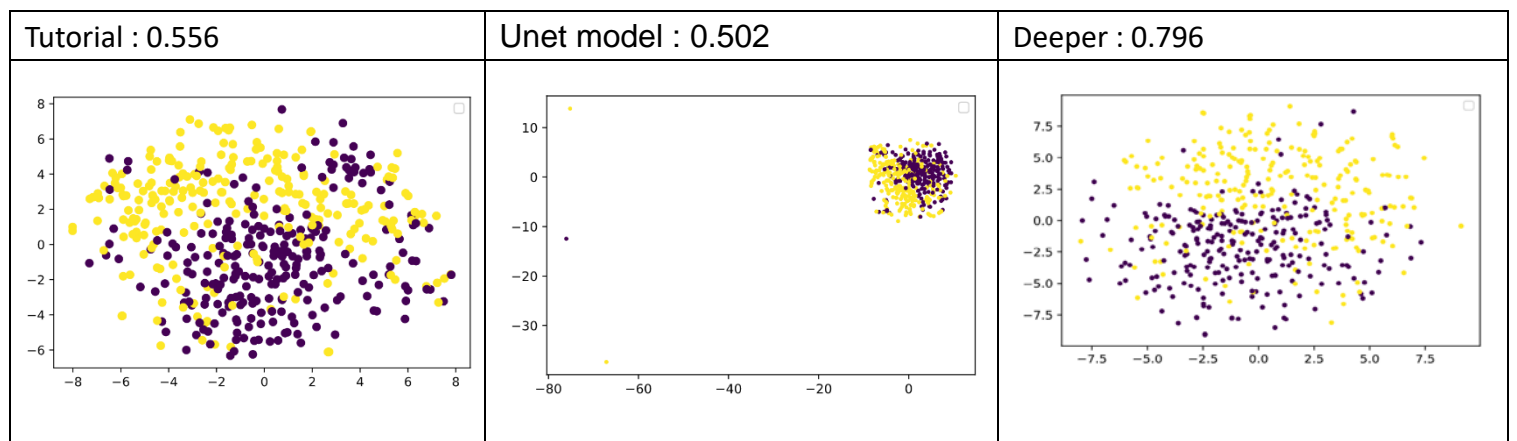
TSNE_PERPLEXITY = 100

TSNE_ITER = 1000

TSNE_LR = 200

KPCA_components = 200

	Tutorial model	Unet model	Deeper tutorial model
Kaggle acc	0.74918	0.73789	0.78942



發現其實 valid 由於只有 500 個點，相較於 train 共 8500 個點其實參考價值不大，tutorial 以及 unet 在 valid 都只有 5 成左右的準確度，但在 kaggle 上都可以得到 7 成以上。

而可以看到 unet model 居然分數比較低，儘管他的還原度十分好，見圖：



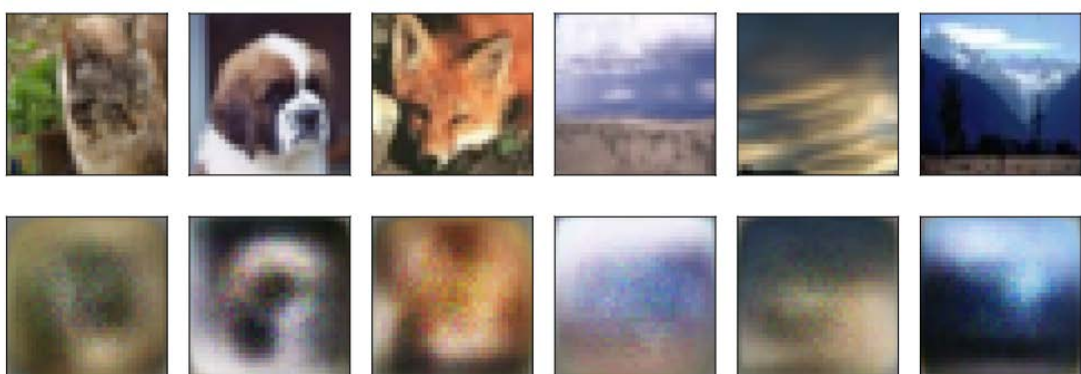
上圖為 unet 使用 problem 2 同樣做圖法後得到的結果，可以看到還原的相當漂亮。但 unet 卻得不到相對好的分群效果，理由猜測為 unet 中 decoder 的部分效能太強大，以及 8500 張圖片可能數量不足，以至於 encoder 的部分 (最後維度 $256 * 4 * 4$) 儲存的資訊其實不足以勝任分群工作。

上述 unet encoder 部分意指 down conv 部分(U 的左半邊) 而 decoder 則是 upsample 部分 (U 的右半邊)。

最後嘗試的便是 t-SNE 的 perplexity 大小，從 sklearn tutorial 以及老師上課影片中可知道 perplexity 對於不同形狀維度的輸入有著很大的差異，經過嘗試後決定使用 100 做為 t-SNE 的取值。

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。

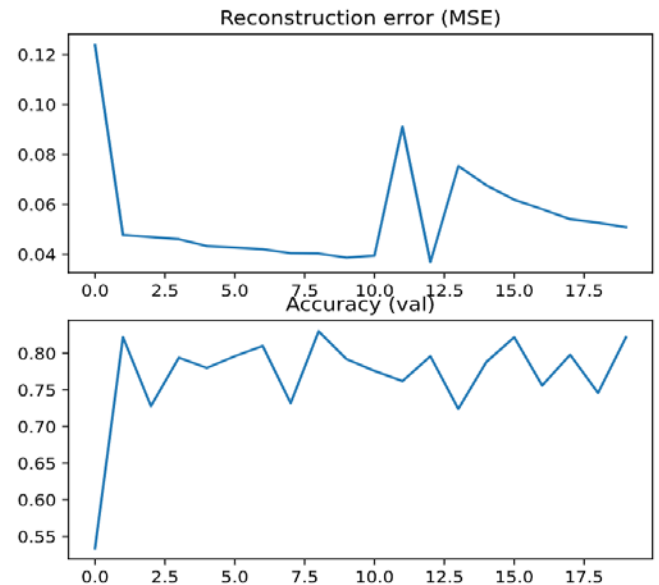


3. (2%) 在 **autoencoder** 的訓練過程中，至少挑選 10 個 **checkpoints**
- a. 請用 **model** 的 **train reconstruction error** (用所有的 **trainX** 計算 **MSE**) 和 **val accuracy** 對那些 **checkpoints** 作圖。
- b. 簡單說明你觀察到的現象。

跑了 20 個 **epoch**，故有 20 個點。

可以看到該 **case** 期訓練速度較快，在約第 2 個 **epoch** 就已經降到不錯的 **loss** 並緩慢開始收斂，而至 **epoch11** 時可能開始有 **over fitting** 情況，**train** 中可能帶有與 **valid** 樣本分布中差異較大的樣本，但震盪後依舊繼續下降。

而在 **Acc** 的部分其走向也與 **MSE** 相去不遠，準確度約在 **epoch 2** 前迅速上升後便開始震盪，甚至在 2 的時候就已經達到了超過 0.8 的 **acc**。而後續震盪可以合理推斷為 **train** 與 **valid** 之樣本分布差異。



透過這樣的比較我們知道其實 **autoencoder** 訓練上可以在少量的 **epoch** 便達到不錯的效果，故其實對 **model** 做更多調整可帶來的效益其實不大，然而在降維與分群，尤其是 **Kernel PCA** 的部分，使用不同的 **sklearn seed** 其實帶來的結果將會天差地遠，同樣的 **model** 不同的 **seed** 有時會帶來 **valid acc** 到 0.2 的差距。其原因猜測為 **encoder** 每次的維度約落在 4096 左右，而要將其降為 200 維其實是相對困難的，而兩次的降維中也難保其資料損失程度，但若是降維維度過少又會造成 **t-SNE** 在計算上的負荷過重，故我認為如何 **fine tune** 降維過程是個 **unsupervised** 中相較重要的一步。