

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。

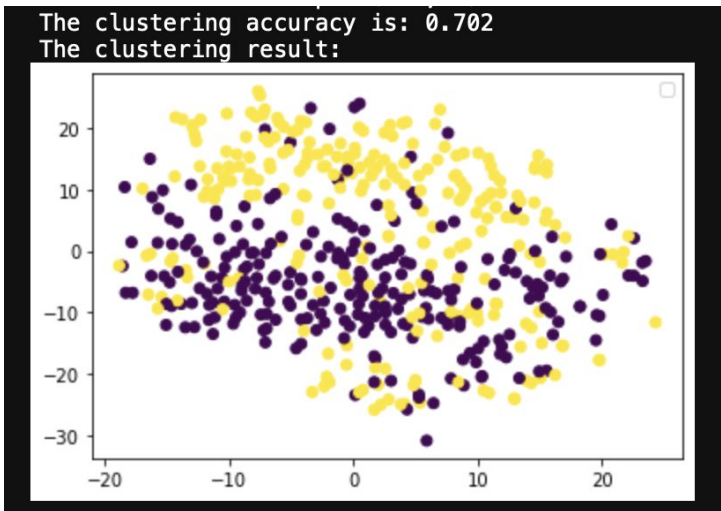
a. 分別記錄改進前、後的 test accuracy 為多少。

改進前在 kaggle 的 public score 為 0.75364。

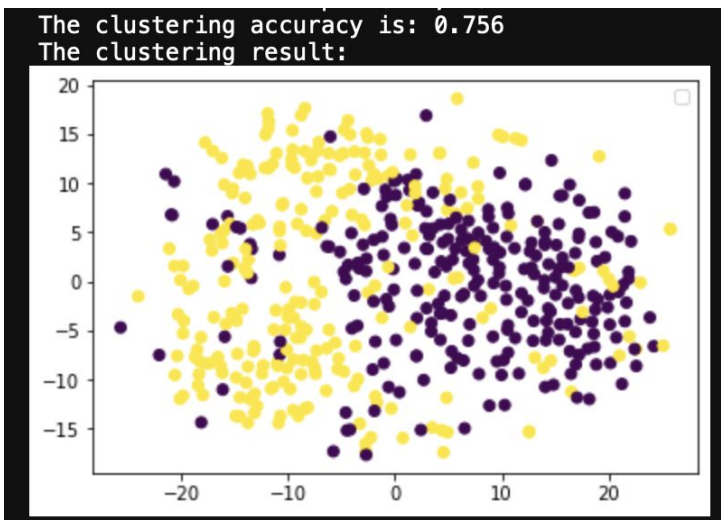
改進後在 kaggle 的 public score 為 0.76682。

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：



改進後：



c. 盡量詳細說明你做了哪些改進。

首先我將 epoch 設為 300，並且將 optimizer 的 learning rate 調成 0.000008，接著再將架構改成下圖所示。

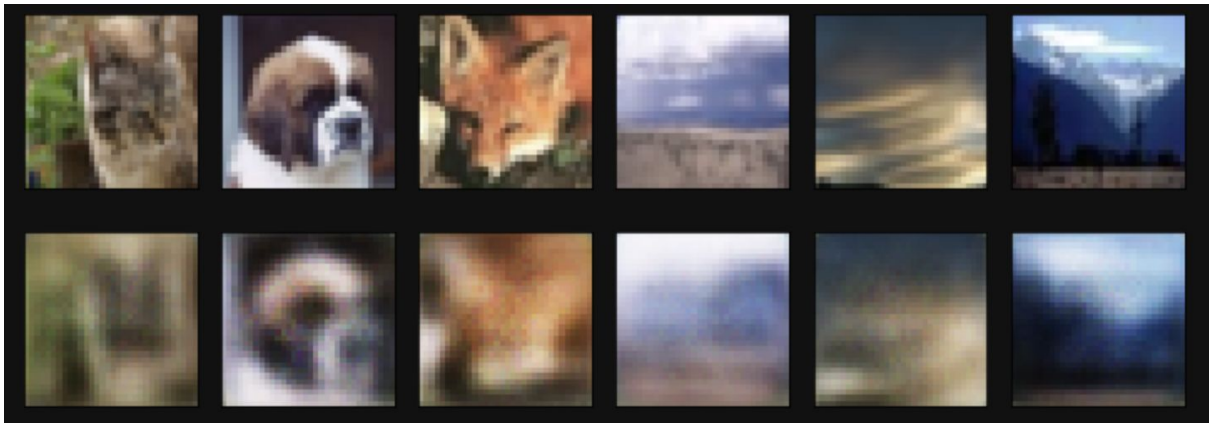
```

self.encoder = nn.Sequential(
    nn.Conv2d(3, 128, 3, stride=1, padding=1),
    nn.ReLU(True),
    nn.MaxPool2d(2),
    nn.Conv2d(128, 256, 3, stride=1, padding=1),
    nn.ReLU(True),
    nn.MaxPool2d(2),
    nn.Conv2d(256, 512, 3, stride=1, padding=1),
    nn.ReLU(True),
    nn.MaxPool2d(2)
)

self.decoder = nn.Sequential(
    nn.ConvTranspose2d(512, 256, 5, stride=1),
    nn.ReLU(True),
    nn.ConvTranspose2d(256, 128, 9, stride=1),
    nn.ReLU(True),
    nn.ConvTranspose2d(128, 3, 17, stride=1),
    nn.Tanh()
)

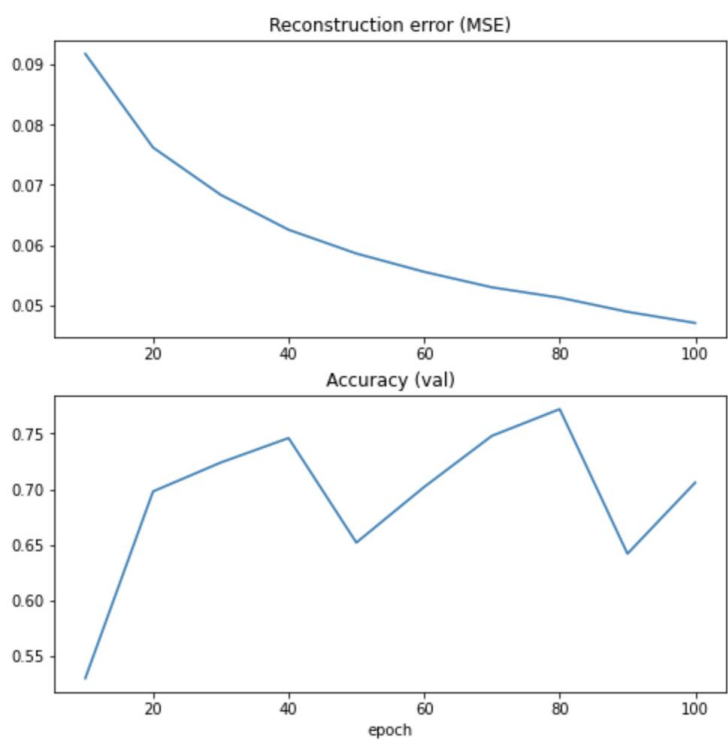
```

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。

下圖的 x 座標為 epoch 當前次數所存的 model，y 座標則是 MSE 以及 val accuracy。



b. 簡單說明你觀察到的現象。

可發現 MSE 會隨著 epoch 增加而減少，這是理所當然的因為我們在 train 的時候，MSE 就是我們的 loss function。val accuracy 則是在前 20 筆 epoch 會有明顯的提升，而之後則是會有些起伏，因為今天就算 encoder 和 decoder 的參數都 train 到很完美，但是在做 classification 的時候也並不能保證在對降維後的 vector 分類時能分得完善，頂多只能保證可以區分大部分而已，也因此只有在前半的 epoch 會造成 val accuracy 明顯的提升。