ML-HW9 Image Clustering Report

學號：b06901066　系級：電機三　姓名：孟妍

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。
分別記錄改進前、後的 test accuracy 為多少。
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。
盡量詳細說明你做了哪些改進。

1.(a)

・Baseline model:

用原本 colab 的 sample code，只改 lr=8e-6 跟 epoch=250，上傳 kaggle 的準確率為 0.75576，差一點點可以過 strong baseline。

・Improved model:

improve 的部分我一開始先把 lr 調成 7e-6，epoch 改成 200，然後在 encoder 的部分每層都加 batch normalization，就可以看到 training 時的 loss 變小，reconstruct 的圖片也有變清晰一些，kaggle 的準確率提升到 0.78705。

```
self.encoder = nn.Sequential(
    nn.Conv2d(3, 64, 3, stride=1, padding=1),
    nn.BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True),
    nn.ReLU(True),
    nn.MaxPool2d(2),
    nn.Conv2d(64, 128, 3, stride=1, padding=1),
    nn.BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True),
    nn.ReLU(True),
    nn.MaxPool2d(2),
    nn.Conv2d(128, 256, 3, stride=1, padding=1),
    nn.BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True),
    nn.ReLU(True),
    nn.MaxPool2d(2)
)
```

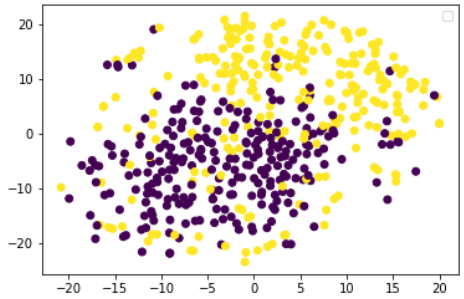
再來我把 encoder 的部分加到 7 層，用類似 vgg 的架構。在 colab 的環境下跑，kaggle 的準確率就提升到了 0.84752，架構如下：

```
self.encoder = nn.Sequential(
    nn.Conv2d(3, 64, 3, 1, 1),
    nn.BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True),
    nn.ReLU(True),
    nn.Conv2d(64, 64, 3, 1, 1),
    nn.BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True),
    nn.ReLU(True),
    nn.MaxPool2d(2, 2, 0),

    nn.Conv2d(64, 128, 3, 1, 1),
    nn.BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True),
    nn.ReLU(True),
    nn.Conv2d(128, 128, 3, 1, 1),
    nn.BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True),
    nn.ReLU(True),
    nn.MaxPool2d(2, 2, 0),

    nn.Conv2d(128, 256, 3, 1, 1),
    nn.BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True),
    nn.ReLU(True),
    nn.Conv2d(256, 256, 3, 1, 1),
    nn.BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True),
    nn.ReLU(True),
    nn.Conv2d(256, 256, 3, 1, 1),
    nn.BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True),
    nn.ReLU(True),
    nn.MaxPool2d(2, 2, 0),
)
```

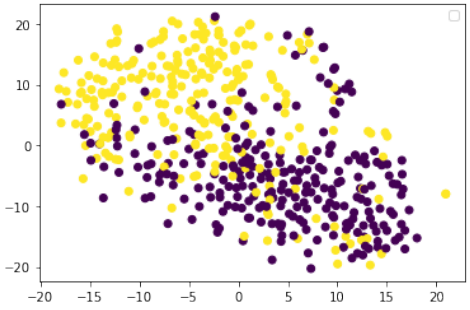
1.(b)

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:

(1) baseline model



(2) improved model

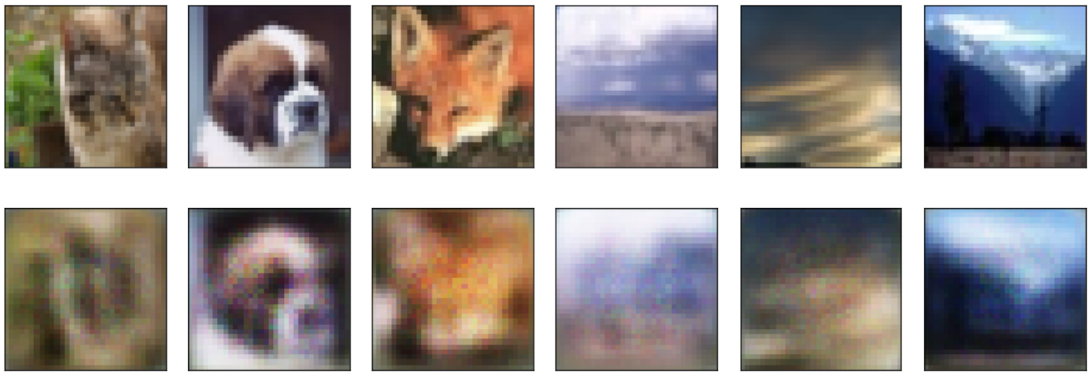


* 有用 same_seeds 和固定 sklearn function 裡的 random_state，但換到另一個環境跑結果就差很多 QQ

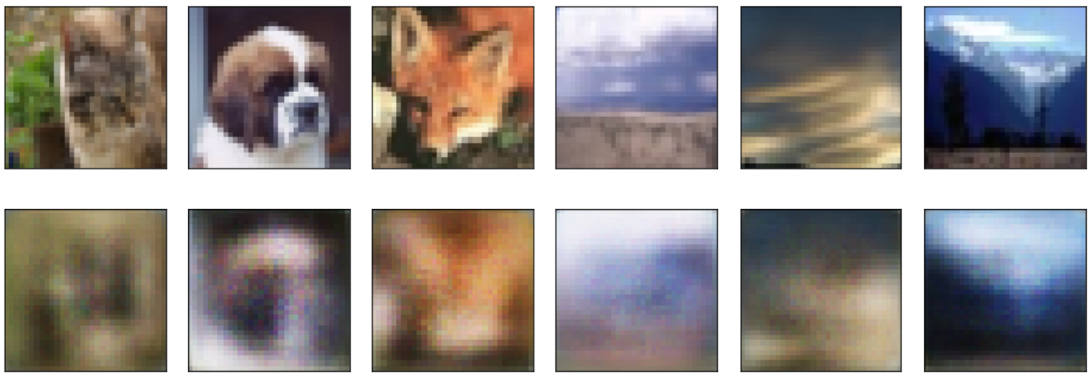
2. (1%) 使用你 test accuracy 最高的 autoencoder，從 trainX 中，取出 index 1, 2, 3, 6, 7, 9 這 6 張圖片

畫出他們的原圖以及 reconstruct 之後的圖片。

· best model:

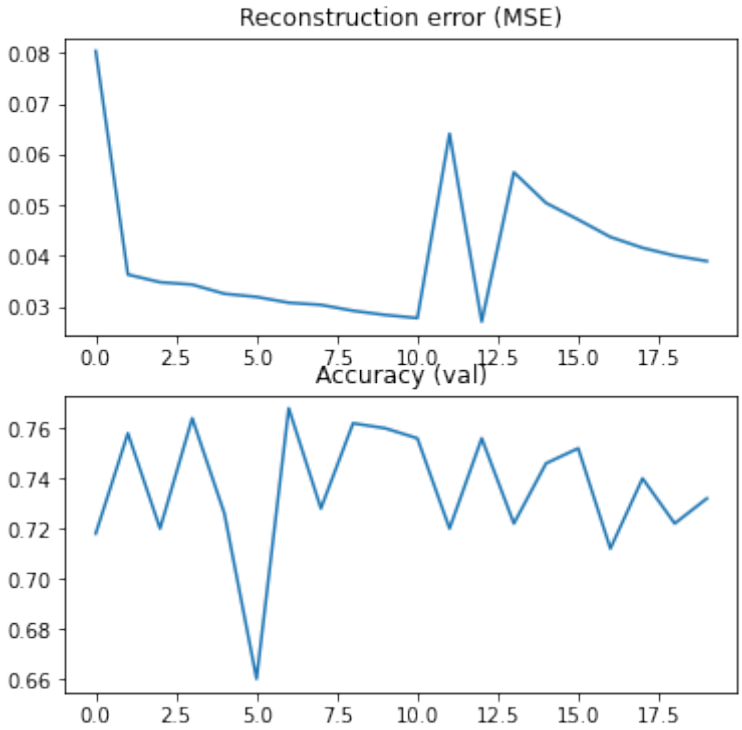


・baseline model



雖然用了 best model reconstruct 的 image 還是有點糊，但跟 baseline model 比還是有變清楚很多。

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。



MSE error 基本上是有隨著訓練 epoch 增加而下降，雖然中間是會有些震盪，但最後的 error 一定是比一開始小很多的。Accuracy 的部分就比較沒有規律，整體而言是明顯有比 baseline model 的 accuracy 高，但 training 的過程 accuracy 一直上下震盪，之前有用 validation 高低決定最後 prediction 要用的 model，但發現結果沒有比較好，且 validation

accuracy 較高的 model 最後上傳 kaggle 的結果也沒有比較好，因此後來都直接拿最後一個 checkpoint 來 predict。