學號：B07902143 系級：資訊三 姓名：陳正康

1. (1%) 請使用不同的Autoencoder model，以及不同的降維方式(降到不同維度)，討論其reconstruction loss & public / private accuracy。（因此模型需要兩種，降維方法也需要兩種，但clustrering不用兩種。）

   i. Model 1:
   Encoder用了(Conv-BatchNorm-ReLU-MaxPool)疊了4次，輸出2x2x256=1024維latent code，decoder則是(ConvTranspose-BatchNorm-ReLU)疊了4次。詳細dimension在最後。
   Reconstruction loss：0.02259
   Model 1以不同降維方法降至2維後做K-Means的正確率：PCA+TSNE: 0.80904 / Spectral Embedding: 0.707333 / LLE: 0.69811

   ii. Model 2:
   Encoder用了(Conv-BatchNorm-ReLU-MaxPool)疊了5次，再接一層線性，輸出256維latent code，decoder則是一層線性再用(ConvTranspose-BatchNorm-ReLU)疊了5次。詳細dimension在最後。
   Reconstruction loss: 0.04115
   Model 2以不同降維方法降至2維後做K-Means的正確率：TSNE: 0.80011 / Spectral Embedding: 0.69311 / LLE: 0.73567

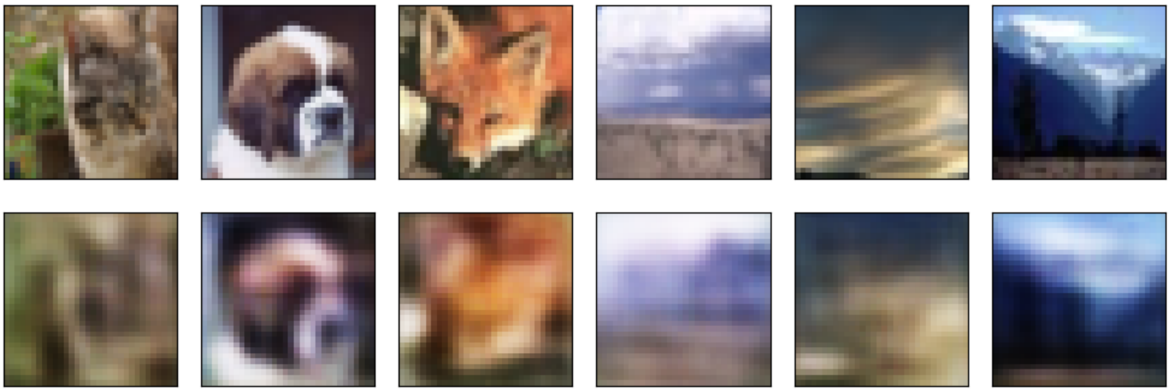
有一個直觀的想法是，autoencoder並不知道要做clustering，因此並沒有特別要讓資料之間的距離拉開，因此latent code的維度太少時，有可能資料擠在一起，造成後續降維的困難，也可以說某些資訊會被autoencoder篩掉，進而使結果變差。

```
Model1(
 (encoder): Sequential(
  (0): Conv2d(3, 32, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
  (1): BatchNorm2d(32, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (2): LeakyReLU(negative_slope=0.01, inplace)
  (3): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
  (4): Conv2d(32, 64, kernel_size=(5, 5), stride=(1, 1), padding=(2, 2))
  (5): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (6): LeakyReLU(negative_slope=0.01, inplace)
  (7): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
  (8): Conv2d(64, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
  (9): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (10): LeakyReLU(negative_slope=0.01, inplace)
  (11): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
  (12): Conv2d(128, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
  (13): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (14): LeakyReLU(negative_slope=0.01, inplace)
  (15): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
 )
 (decoder): Sequential(
  (0): ConvTranspose2d(256, 128, kernel_size=(3, 3), stride=(1, 1))
  (1): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (2): LeakyReLU(negative_slope=0.01, inplace)
  (3): ConvTranspose2d(128, 64, kernel_size=(5, 5), stride=(1, 1))
  (4): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (5): LeakyReLU(negative_slope=0.01, inplace)
  (6): ConvTranspose2d(64, 32, kernel_size=(9, 9), stride=(1, 1))
```

```
    (7): BatchNorm2d(32, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (8): LeakyReLU(negative_slope=0.01, inplace)
    (9): ConvTranspose2d(32, 3, kernel_size=(17, 17), stride=(1, 1))
    (10): BatchNorm2d(3, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (11): Tanh()
  )
)
Model2(
  (encoder): Sequential(
    (0): Conv2d(3, 32, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (1): BatchNorm2d(32, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (2): LeakyReLU(negative_slope=0.01, inplace)
    (3): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
    (4): Conv2d(32, 64, kernel_size=(5, 5), stride=(1, 1), padding=(2, 2))
    (5): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (6): LeakyReLU(negative_slope=0.01, inplace)
    (7): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
    (8): Conv2d(64, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (9): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (10): LeakyReLU(negative_slope=0.01, inplace)
    (11): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
    (12): Conv2d(128, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (13): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (14): LeakyReLU(negative_slope=0.01, inplace)
    (15): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
    (16): Conv2d(256, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (17): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (18): LeakyReLU(negative_slope=0.01, inplace)
    (19): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
)
(linear1): Sequential(
    (0): Linear(in_features=512, out_features=256, bias=True)
    (1): BatchNorm1d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (2): LeakyReLU(negative_slope=0.01, inplace)
  )
  (linear2): Sequential(
    (0): Linear(in_features=256, out_features=512, bias=True)
    (1): BatchNorm1d(512, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (2): LeakyReLU(negative_slope=0.01, inplace)
  )
  (decoder): Sequential(
    (0): ConvTranspose2d(512, 256, kernel_size=(2, 2), stride=(1, 1))
    (1): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (2): LeakyReLU(negative_slope=0.01, inplace)
    (3): ConvTranspose2d(256, 128, kernel_size=(3, 3), stride=(1, 1))
    (4): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (5): LeakyReLU(negative_slope=0.01, inplace)
    (6): ConvTranspose2d(128, 64, kernel_size=(5, 5), stride=(1, 1))
    (7): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (8): LeakyReLU(negative_slope=0.01, inplace)
    (9): ConvTranspose2d(64, 32, kernel_size=(9, 9), stride=(1, 1))
    (10): BatchNorm2d(32, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (11): LeakyReLU(negative_slope=0.01, inplace)
    (12): ConvTranspose2d(32, 3, kernel_size=(17, 17), stride=(1, 1))
    (13): BatchNorm2d(3, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (14): Tanh()
  )
)
```

2. (1%) 從dataset選出2張圖，並貼上原圖以及經過autoencoder後reconstruct的圖片。

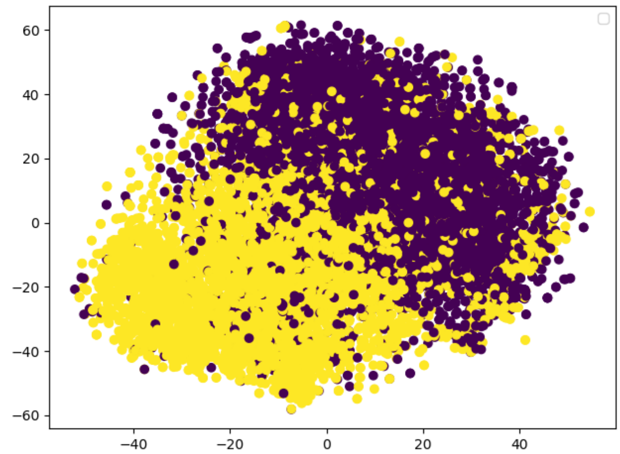
此為model 1的輸出



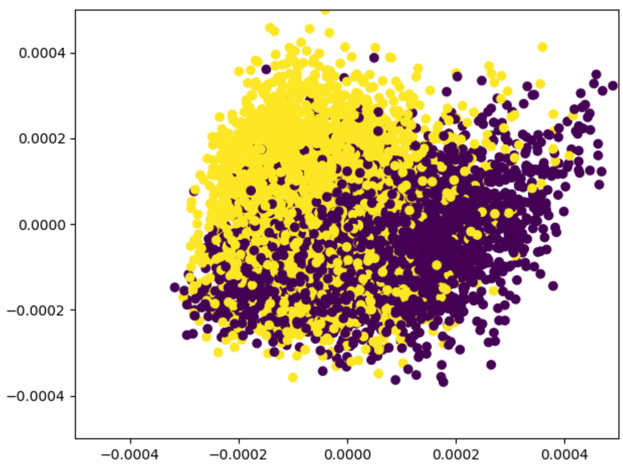
3. (1%) 我們會給你dataset的label。請在二維平面上視覺化label的分佈。

以下使用Model 1

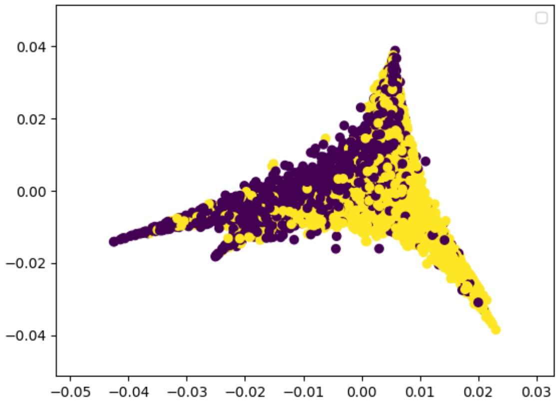
PCA+TSNE:



Spectral Embedding:

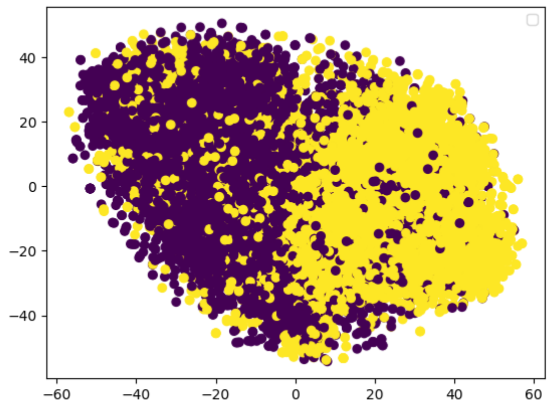


LLE:

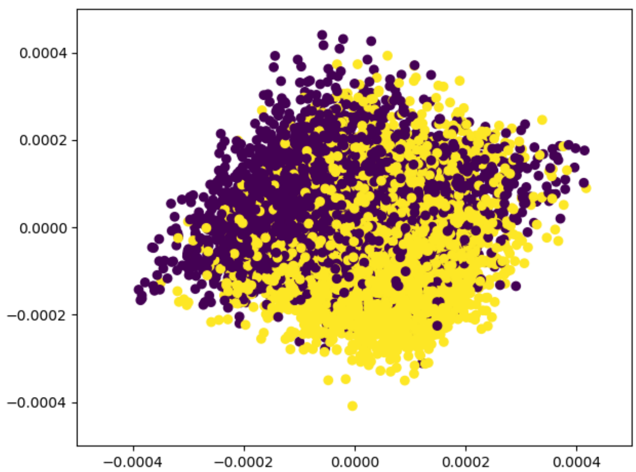


以下使用Model 2

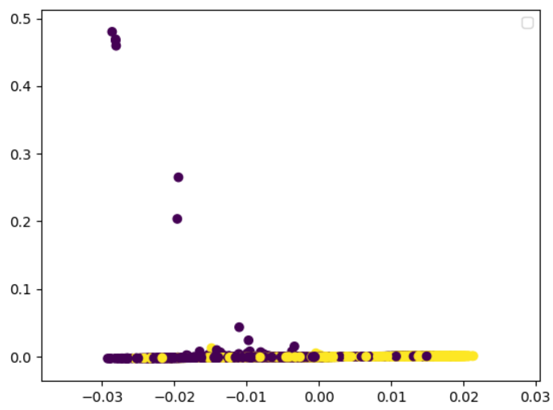
TSNE:



Spectral Embedding:



LLE:



4. (3%) Refer to math problem https://drive.google.com/file/d/1-rmlFaIj_6hEfJGOHLKUxInoKMsKLHLf/view?usp=sharing

4-1.

$$\begin{cases} z = w \cdot x + b \\ z_i = w_i \cdot x + b_i \\ z_f = w_f \cdot x + b_f \\ z_o = w_o \cdot x + b_o \end{cases}$$

$$y = f(z_o)\, h(c')$$

$$c' = f(z_i)\, g(z) + c\, f(z_f)$$

$$f(z) = \frac{1}{1+e^{-z}}$$

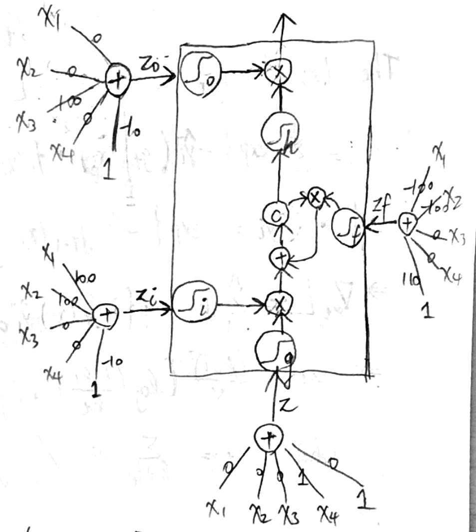
$$g(z) = z$$

$$h(z) = z$$

$$\begin{cases} f(z_i) = x_1 \vee x_2 \\ f(z_o) = x_3 \\ f(z_f) = x_1 \wedge x_2 \to 0 \end{cases}$$

$x_1 \oplus x_2 \to \sigma(10)$

$\sim x_1 \wedge \sim x_2 \to 1$



t	1	2	3	4	5	6	7	8
x_t	0 1 0 3	1 0 1 -2	1 1 1 4	0 1 1 0	0 1 0 2	0 0 1 -4	1 1 1 1	1 0 1 2
$g(z)=z$	3	-2	4	0	2	-4	1	2
$f(z_i)$	1	1	1	1	1	0	1	1
c	0	3	$-2+3\sigma(10)$	4	$4\sigma(10)$	$2+4\sigma^2(10)$	$2+4\sigma^2(10)$	1
$f(z_f)$	$\sigma(10)$	$\sigma(10)$	0	$\sigma(10)$	$\sigma(10)$	1	0	$\sigma(10)$
$h(c')=c'$	3	$-2+3\sigma(10)$	4	$4\sigma(10)$	$2+4\sigma^2(10)$	$2+4\sigma^2(10)$	1	$2+\sigma(10)$
$f(z_o)$	$\sigma(-10)$	1	1	1	$\sigma(-10)$	1	1	1
y_t	$3\sigma(-10)$ ≈ 0.000136	$-2+3\sigma(10)$ ≈ 0.999864	4	$4\sigma(10)$ ≈ 3.999818	≈ 0.000272	$2+4\sigma^2(10)$ ≈ 5.999637	1	$2+\sigma(10)$ ≈ 2.999955

Where $\sigma(x) = \frac{1}{1+e^{-x}}$

4-2.

$$h = W^T x \in R^N \quad (W \in R^{V \times N})$$

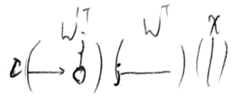
$$u = W'^T h \in R^V \quad (W' \in R^{N \times V})$$

$$y = \text{Softmax}(u) \in R^V$$

$$\text{Loss} = L = -\log \prod_{c \in C} P(W_{out,c}, W_{in}) = -\log \prod_{c \in C} \frac{e^{u_c}}{\sum_{i \in V} e^{u_i}} = \sum_{c \in C} -\log \overset{\text{c-th elem. of } y}{y_c}$$

$$C = \text{Context of input}$$

$$\Rightarrow \frac{\partial L}{\partial W_{ij}^T} = \sum_{c \in C} -\frac{1}{y_c}\Big(\overset{\frac{\partial}{\partial u_c}\text{softmax}(u_c)}{y_c + y_c^2}\Big)\frac{\partial u_c}{\partial W_{ij}^T} = \sum_{c \in C} -(1+y_c)\frac{\partial}{\partial W_{ij}^T}\underset{\text{c-th elem.}}{[W'^T W^T x]_c}$$



$$= \sum_{c \in C} -(1+y_c)\frac{\partial}{\partial W_{ij}^T}\left(W'^T_{ci} W^T_{ij} x_j\right) = \sum_{c \in C} -(1+y_c) W'^T_{ci} x_j$$

$$\frac{\partial L}{\partial W'^T_{ij}} = \sum_{c \in C} -(1+y_c)\frac{\partial}{\partial W'^T_{ij}}\left[W'^T W^T x\right]_c = \sum_{c \in C} -(1+y_c)\frac{\partial}{\partial W'^T_{ij}}\left(W'^T_{cj}\left(W^T_{j1} x_1 + W^T_{j2} x_2 + \cdots + W^T_{jV} x_V\right)\right)$$

$$= \begin{cases} 0 & \text{if } i \notin C \\ -(1+y_i)\left(W^T_{j1} x_1 + \cdots + W^T_{jV} x_V\right) & \text{if } i \in C \end{cases}$$

$$= \begin{cases} -(1+y_i)\left[W^T x\right]_j & \text{if } i \in C \\ 0 & \text{if } i \notin C \end{cases} = \begin{cases} -(1+y_i) h_j & \text{if } i \in C \\ 0 & \text{if } i \notin C \end{cases}$$