學號：R07941023 系級： 光電碩二 姓名：呂彥穎

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。）

Sol:

decomposition	clustrering	latens dim	Loss(-1~1)	public accuracy	private accuracy
AE+PCA	K-means	128	980.792	0.80	0.808
VAE+ PCA	K-means	64	995.319	0.6865.	0.6803

Original size: 32x32

這裡採用 AE 以及 VAE

(model 越簡單越好, 最後是靠 AE+PCA+kmeans 分 10 類(每次結果不同, 需要多傳幾次)才能過 strong baseline)

AE model:

------enconder

1. Convolution – 16x30x30
2. BatchNorm – 16x30x30
3. ReLU – 16x30x30
4. Pooling– 16x15x15
5. Convolution – 8x8x8
6. BatchNorm – 8x 8x8
7. ReLU – 8x8x8
8. Pooling– 8x4x4

------decoder

9. ConvolutionTranspose – 16x9x9
10. BatchNorm – 16x9x9
11. ReLU – 16x9x9
12. ConvolutionTranspose – 8x17x17
13. BatchNorm – 8x17x17
14. ReLU – 8x17x17
15. ConvolutionTranspose – 1x32x32
16. Tanh – 1x32x32

VAE model:

------enconder

1. Convolution – 16x30x30
2. BatchNorm – 16x30x30
3. ReLU – 16x30x30
4. Pooling– 16x15x15
5. Convolution – 16x8x8

6. BatchNorm – 16x8x8
7. ReLU – 8x8x8
8. Pooling– 8x4x4
9. linear – 100
10. linear – 64

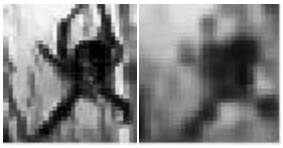
------decoder

11. linear – 128
12. ReLU – 128
13. Dropout - 128
14. ConvolutionTranspose – 16x9x9
15. BatchNorm – 16x9x9
16. ReLU – 16x9x9
17. ConvolutionTranspose – 8x17x17
18. BatchNorm – 8x17x17
19. ReLU – 8x17x17
20. ConvolutionTranspose – 1x32x32
21. Tanh – 1x32x32

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

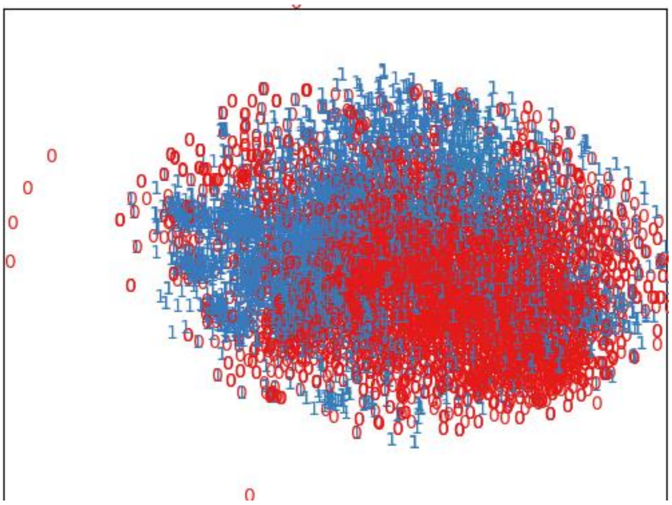
Sol:

ID=0: 原圖/AE reconstruct



ID=10: 原圖/AE reconstruct

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



4. (3%)Refer to math problem
https://drive.google.com/file/d/1e_IDAV2yv0YEhIuVWpDdaH4Pzz5s1p2P/view?fbclid=IwAR0tO9NRxK9JZeUDNdawNuSbGTvqI7niuMX3Kkk9arauC8O6p6iJc7oMz84

P1

(a) $\mu = \frac{1}{N}\sum_{i=1}^{N} x_i = \frac{1}{10}\begin{bmatrix} 1+4+3+1+5+7+9+3+11+10 \\ 2+8+12+8+14+4+8+8+5+11 \\ 3+5+9+5+2+1+9+1+6+7 \end{bmatrix} = \begin{bmatrix} 5.4 \\ 8 \\ 4.8 \end{bmatrix}$ — (1)

$\Sigma = \frac{1}{N}\sum_{i=1}^{N}(x_i-\mu)(x_i-\mu)^T$, $\mu = \begin{bmatrix} 5.4 \\ 8 \\ 4.8 \end{bmatrix}$, $N = 10$

$= \begin{bmatrix} 12.04 & 0.5 & 3.28 \\ 0.5 & 12.2 & 2.9 \\ 3.28 & 2.9 & 8.16 \end{bmatrix}$ — (2)

$\Sigma x = \lambda x$ — (3)

$p(\lambda) = \det(\Sigma - \lambda I) = 0$

$= \begin{bmatrix} 12.04-\lambda & 0.5 & 3.28 \\ 0.5 & 12.2-\lambda & 2.9 \\ 3.28 & 2.9 & 8.16-\lambda \end{bmatrix}$

$= -\lambda^3 + 32.4\lambda^2 - 325.268\lambda + 973.5692$

$= -(\lambda - 15.2974434)(\lambda - 11.63052369)(\lambda - 5.472032913)$

If $\lambda = \lambda_1 = 15.2974434$, (3) $\Rightarrow \begin{cases} (12.04-\lambda_1)x_1 + 0.5x_2 + 3.28x_3 = \lambda_1 x_1 \\ 0.5x_1 + (12.2-\lambda_1)x_2 + 2.9x_3 = \lambda_1 x_2 \\ 3.28x_1 + 2.9x_2 + (8.16-\lambda_1)x_3 = \lambda_1 x_3 \end{cases}$, $\Rightarrow \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} -0.61659 \\ -0.58882 \\ -0.52259 \end{bmatrix}$ — (4)

If $\lambda = \lambda_2 = 11.63$, 同上. 由(3), $\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} -0.67818 \\ 0.73439 \\ -0.02728 \end{bmatrix}$ — (5)

If $\lambda = \lambda_3 = 5.47$ 同上, 由(3), $\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} -0.39985 \\ -0.33758 \\ 0.85214 \end{bmatrix}$ — (6)

(b) sample 1 $d_1 = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}$ principal component. 由(4)(5)(6) $V = \begin{bmatrix} -0.61659 & -0.67818 & -0.39985 \\ -0.58882 & 0.73439 & -0.33758 \\ -0.52259 & -0.02728 & 0.85214 \end{bmatrix}$

$\Rightarrow Vd_1 = [-3.362, 0.7087, 1.481]$, $Vd_{10} = [-16.3011, 1.1055, -1.747]$

$Vd_2 = [-9.789, 3.0259, -0.0394]$

$Vd_3 = [-13.618, 6.532, 2.4186]$

$Vd_4 = [-7.94, 5.0605, 1.1601]$

$Vd_5 = [-12.371, 6.836, -5.0212]$

$Vd_6 = [-7.194, -1.837, -3.297]$

$Vd_7 = [-14.963, -0.474, 1.3698]$

$Vd_8 = [-7.0829, 3.8132, -3.0481]$

$Vd_9 = [-12.862, -3.9517, -0.9735]$

(c)

from ④.⑤ , 2D . $V = \begin{bmatrix} -0.6166 & -0.6782 \\ -0.5888 & 0.7344 \\ -0.5226 & -0.0213 \end{bmatrix}$

Reconstruction $R = VV^T$

for data 1. $d_1 = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}$. $d_1' = Rd_1 = \begin{bmatrix} 1.5923 \\ 2.5001 \\ 1.7316 \end{bmatrix}$, $loss = (d_1 - d_1')^2 = 1.48$

同理, $\begin{cases} d_2 .\ loss = 0.039 \\ d_3 .\ loss = 2.418 \\ d_4 .\ loss = 1.16 \\ d_5 .\ loss = 5.02 \\ d_6 ,\ loss = 3.297 \\ d_7 ,\ loss = 1.36988 \\ d_8 ,\ loss = 3.048 \\ d_9 ,\ loss = 0.97349 \\ d_{10} .\ loss = 1.747 \end{cases}$

$L = \frac{1}{N} \sum_{i=1}^{N} loss\, d_i = \frac{1}{10}(16.05237) = \underline{1.605237}$ //

P2

(a) $\overset{m\times n\ n\times m}{AA^T}$, $\overset{n\times m\ m\times n}{A^TA}$, $AA^T = \overset{m\times m}{\Sigma}$ (以下證明同等 $A^TA = \overset{n\times n}{\Sigma}$)

given $w = \begin{bmatrix} w_1 \\ \vdots \\ w_m \end{bmatrix}$ $\Rightarrow W^T \Sigma(A) W \geq 0$

$\Rightarrow \Sigma(W^TA) \geq 0$

$\Rightarrow \mathrm{var}\left[\sum_{i=1}^{m} w_i a_i\right] \geq 0$

$\Rightarrow \sum_{i=1}^{m} \mathrm{var}(w_i a_i) \geq 0$ #

$\Phi^T = \Phi^{-1}$

(b) $\because \overset{m\times m}{\Sigma}$ is a positive semi-definite $\therefore \Sigma = \overset{m\times n\ n\times m}{XX^T}$,

$= \frac{1}{n}\sum_{i=1}^{n} x_i x_i^T$. 則 $\frac{1}{n}\sum_{i=1}^{n} x_i = \mu$ #

$A = \phi^T$. $B = \Sigma$. $C = I$

(c) $\frac{\partial(Tr(\phi^T\Sigma\phi))}{\partial\Phi} = \nabla_\phi Tr(\phi^T\Sigma\phi)$. let $\phi^T\Sigma = \mu(\phi^T)$. $\phi = V(\phi)$

$= \nabla_{\phi:\mu(\phi^T)} Tr(\mu(\phi^T)V(\phi)) + \nabla_{\phi:V(\phi)} Tr(\mu(\phi^T)V(\phi))$, $tr(\phi)$

$= \nabla_\phi \mu(\phi) Tr(\mu(\phi)V(\phi^T)) + \nabla_\phi \cdot V(\phi^T) Tr(\mu(\phi)V(\phi^T))$.

$= (V(\phi^T))^T \nabla_\phi \mu(\phi) + (\nabla_{\phi^T} \cdot (\phi^T) Tr(\mu(\phi)V(\phi^T)))^T$

$= \phi\Sigma^T + ((\mu(\phi)))^T \nabla_{\phi^T} V(\phi^T))^T$

$= \phi\Sigma + \phi\Sigma^{-1}$

$\Phi = \Phi - \phi\Sigma + \phi\Sigma^{-1}$ #

P3

gradient boosting

goal $\min_{g} L(g) = \sum_{n=1}^{N} \exp(-\hat{y}^n g(x^n))$

$g_t(x) = g_{t-1}(x) + \alpha_t f_t(x) \Rightarrow L(g_t) = \sum_{n=1}^{N} - e^{-\hat{y} g_t(x)} = \sum_{n=1}^{N} e^{-\hat{y}^n (g_{t-1}(x^n) + \alpha_t f_t(x^n))}$

$= \sum_{n=1}^{N} e^{\hat{y}^n g_{t-1}(x^n) - \hat{y}^n \alpha_t f_t(x^n)}$

$= \underbrace{\sum_{n=1}^{N} e^{-\hat{y}^n g_{t-1}(x^n)}}_{L(g_{t-1})} + \sum_{n=1}^{N} \underbrace{e^{-\hat{y} g_{t-1}(x^n)}}_{u_t^n} [(-\hat{y}^n) \alpha_t f_t(x^n)]$

To $\min \sum_{n=1}^{N} u_t^n [(-\hat{y}^n) \alpha_t f_t(x^n)] = \min \sum_{n=1}^{N} u_t^n \alpha_t [\delta(f_t(x^n) \neq \hat{y}^n) - \underbrace{\delta(f_t(x^n) = \hat{y})}_{1 - \delta(f_t(x^n) \neq \hat{y})}]$

$= \min \sum_{n=1}^{N} u_t^n \alpha_t [2\delta(f_t(x^n) \neq \hat{y}^n) - 1] = \min_{f_t} \alpha_t \sum_{n: f_t(x^n) \neq \hat{y}^n} u_t^n$

$L(g_t) = \sum_{n: f_t(x^n) \neq \hat{y}^n} u_t^n e^{\alpha_t} + \sum_{n: f_t(x^n) = \hat{y}^n} u_t^n e^{-\alpha_t} = Z_t \epsilon_t e^{\alpha_t} + Z_t (1 - \epsilon_t) e^{-\alpha_t}$

则

$\Rightarrow L(g_T^1, \ldots, g_T^k) = \sum_{i=1}^{n} e^{\frac{1}{k-1} \sum_{k \neq \hat{y}_i} g_T^k(x_i) - g_T^{\hat{y}_i}(x_i)}$