

## Kaggle Competition

1. (1%) 請附上你在 kaggle 競賽上表現最好的降維以及分群方式，並條列五種不同降維維度的設定對應到的表現(public / private accuracy)

註 1: auto-encoder 和 PCA 只要任一維度不一樣即可算是一種組合。

註 2: 不限於以上方法，同學也可以使用任何其他 embedding algorithm 實現降維。

Public Score	Private Score	Reduced Method	Latent Dimension	Reduced Dimension
0.78355	0.77733	PCA	32	8
0.78333	0.77755	tSNE	32	8
0.77822	0.78244	PCA	32	16
0.77800	0.78377	PCA	32	64
0.77222	0.77133	PCA	128	8
0.77044	0.77244	PCA	64	8

2. (1%) 從 train.npy 選出不同類別的 2 張圖，貼上原圖以及你的 autoencoder reconstruct 的圖片。用 Mean Square Error 計算這兩張圖的 reconstruction error，並說明該 error 與 kaggle score 的關係。

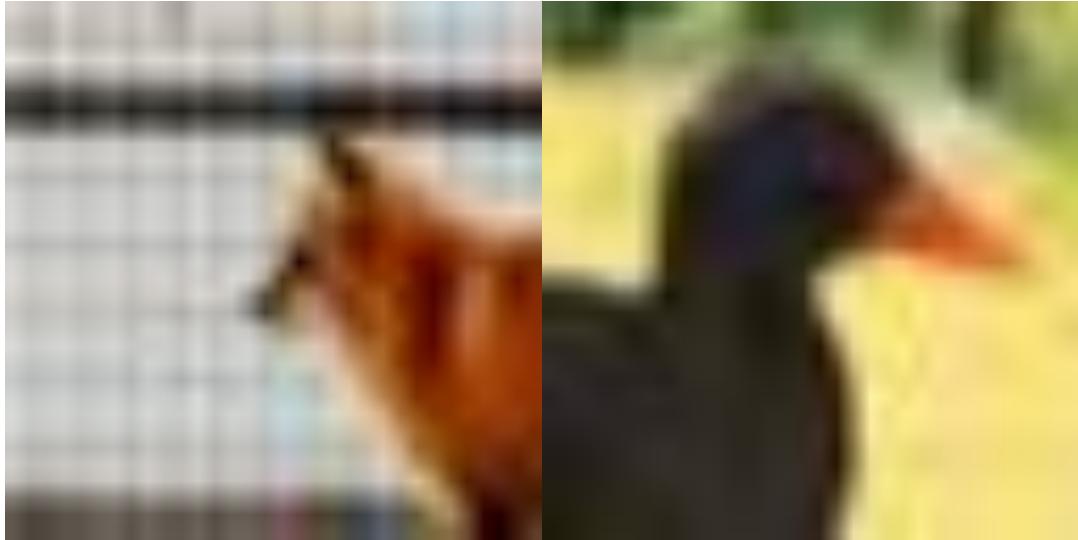
註 1: 所以一共要貼上 4 張圖片。

註 2: 原圖請貼上做 augmentation 之前的版本。

Original Image



Reconstruct Image



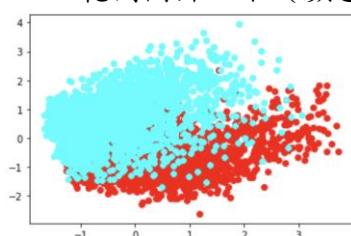
The error loss made by MSELoss is 0.0162 and 0.0055 respectively. And the checkpoint is assigned by the highest score version on Kaggle. When you assigned different checkpoint parameter on Kaggle, you'll find that they have positive correlation.

3. (2%) 請使用 pca 以及 tsne 兩種方法，將 visualization.npy 的圖片經過 autoencoder 降維後得到之 latent vector，進一步降維至二維平面並作圖。並說明兩張圖之差異。

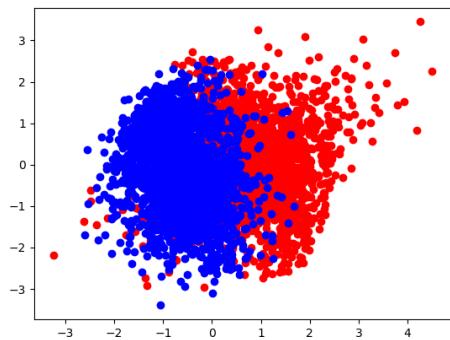
註 1: visualization.npy 前 2500 張 label 為 0；後 2500 張 label 為 1

註 2: 一共要貼上 2 張圖片。

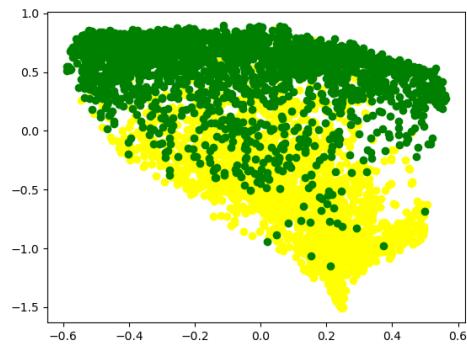
註 3: 範例圖片如下（顏色、分佈不用完全一樣）



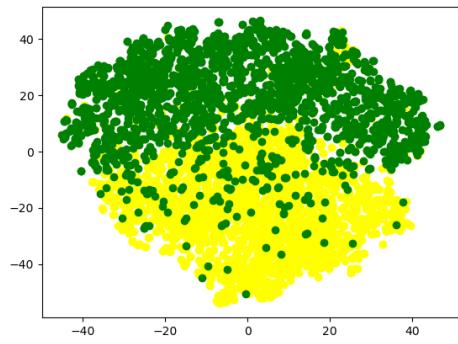
The outcome of the testing is as below. I used PCA with default configuration and tSNE with 250 iterations, 500 iterations and 1000 respectively. And the model is used that same as the previous question, that is, the best score on Kaggle. You can see that the result generated from PCA is crowded that can not split the data clearly. But, tSNE can split the data as much as better when iteration getting higher. You can see that the result from tSNE with 500 iterations, 750 iterations, and 1000 iterations are similar.



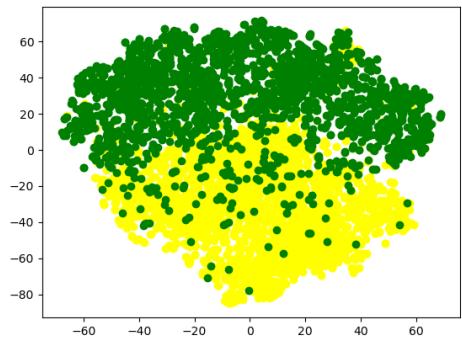
▲ PCA



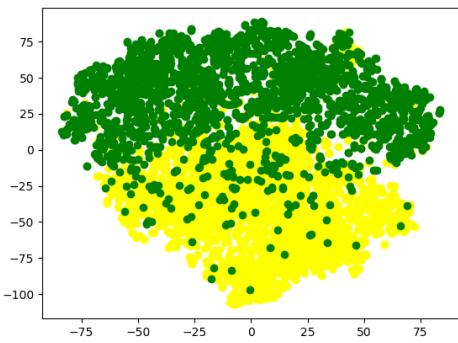
▲ tSNE with iter=250



▲ tSNE with iter=500



▲ tSNE with iter=750



▲ tSNE with iter=1000

4. (4%) Refer to math problem :

<https://hackmd.io/@g4HRMJCzQL2hzLedRcbVPQ/SyCBoc1qt>

### MLHW3.

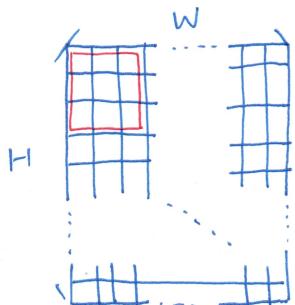
Q1. Given batch of image data with shape  $(B, W, H, C_{\text{Input}})$

$\Rightarrow \text{Conv2D}(C_{\text{in}}, C_{\text{out}}, \text{kernel-size}=(k_1, k_2), \text{stride}=(s_1, s_2), \text{padding}=(P_1, P_2))$

(Solution) As the reference of pytorch document

$$H_{\text{out}} = \left\lfloor \frac{H_{\text{in}} + 2P_1 - (k_1 - 1) - 1}{s_1} + 1 \right\rfloor$$

$$W_{\text{out}} = \left\lfloor \frac{W_{\text{in}} + 2P_2 - (k_2 - 1)}{s_2} + 1 \right\rfloor$$



if  $k=(3 \times 3)$ ,  
stride  $(1,1)$ ,  
padding  $(0,0)$   $\Rightarrow H_{\text{new}} = H - 2$   
 $H_{\text{new}} = H - (k_2 - 1)$

if  $k=(5 \times 5)$ ,  
stride  $(1,1)$ ,  
padding  $(0,0)$   $\Rightarrow W_{\text{new}} = W - 4$   
 $H_{\text{new}} = H - 4$

$$W_{\text{new}} = W - (k_1 - 1)$$

$$H_{\text{new}} = H - (k_2 - 1)$$

if  $k = k_1 \times k_2$   
stride  $(1 \times 1)$   $\Rightarrow W_{\text{new}} = W + 2P_1 - (k_1 - 1)$   
padding  $(P_1, P_2)$   $H_{\text{new}} = H + 2P_2 - (k_2 - 1)$

if  $k=1 \times 1$   
padding  $(0,0)$   $\Rightarrow W_{\text{new}} = W$   
stride  $(1 \times 1)$   $H_{\text{new}} = H$

$k=1 \times 1$ padding $(0,0)$ $\Rightarrow W_{\text{new}} = \left\lfloor \frac{W}{2} + 1 \right\rfloor$ stride $(2 \times 2)$ $H_{\text{new}} = \left\lfloor \frac{H}{2} + 1 \right\rfloor$	$k=1 \times 1$ padding $(0,0)$ $\Rightarrow W_{\text{new}} = \left\lfloor \frac{W}{3} + 1 \right\rfloor$ stride $(3,3)$ $H_{\text{new}} = \left\lfloor \frac{H}{3} + 1 \right\rfloor$
--	---

$W_{\text{new}} = \left\lfloor \frac{W + 2P_1 - (k_1 - 1)}{s_1} + 1 \right\rfloor$

$H_{\text{new}} = \left\lfloor \frac{H + 2P_2 - (k_2 - 1)}{s_2} + 1 \right\rfloor$

if  $W=16$   
 $k=1$   
padding  $= 0$

$W_{\text{new}} = \begin{cases} 8 & , \text{if } s=2 \\ 6 & , \text{if } s=3 \\ 4 & , \text{if } s=4 \\ 4 & , \text{if } s=5 \end{cases}$

$\left\lfloor \frac{16-1}{2} + 1 \right\rfloor = 8$   
 $\left\lfloor \frac{16-1}{3} + 1 \right\rfloor = 6$   
 $\left\lfloor \frac{16-1}{4} + 1 \right\rfloor = 4$   
 $\left\lfloor \frac{16-1}{5} + 1 \right\rfloor = 4$

$\Rightarrow W_{\text{new}} = \left\lfloor \frac{W_{\text{in}} + 2P_1 - (k_1 - 1) - 1}{s_1} + 1 \right\rfloor$

$H_{\text{new}} = \left\lfloor \frac{H_{\text{in}} + 2P_2 - (k_2 - 1) - 1}{s_2} + 1 \right\rfloor$

Q.E.D.

Q2 Given Input: value of  $X$  over a mini-batch:  $B = \{x_1, \dots, x_m\}$

Output:  $y_i = BN_{\gamma, \beta}(x_i)$

$$\mu_B = \frac{1}{m} \sum_{i=1}^m x_i$$

$$\sigma_B^2 = \frac{1}{m} \sum_{i=1}^m (x_i - \mu_B)^2$$

$$\hat{x}_i = \frac{x_i - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}}$$

$$y_i = \gamma \hat{x}_i + \beta \equiv BN_{\gamma, \beta}(x_i)$$

Before solution, we can review the differentiation w.r.t multi-variable function first.

Given  $u(x, y)$  where  $x(r, t)$  and  $y(r, t)$ , then to determine the value of  $\frac{\partial u}{\partial r}, \frac{\partial u}{\partial t}$   
we need to use multi-variable chain rules that is  $\frac{\partial u}{\partial r} = \frac{\partial u}{\partial x} \frac{\partial x}{\partial r} + \frac{\partial u}{\partial y} \frac{\partial y}{\partial r}$

<sol> We have 5 functions

$$\textcircled{1} l(y) \quad \textcircled{2} y(\hat{x}, \gamma, \beta) \quad \textcircled{3} \hat{x}(x, \mu_B, \sigma_B^2) \quad \textcircled{4} \underline{\mu_B}(x) \quad \textcircled{5} \underline{\sigma_B^2}(x, \mu_B)$$

To determine  $\frac{\partial l}{\partial x_i} = \frac{\partial l}{\partial \hat{x}_i} \frac{\partial \hat{x}_i}{\partial x_i} + \frac{\partial l}{\partial \mu_B} \frac{\partial \mu_B}{\partial x_i} + \frac{\partial l}{\partial \sigma_B^2} \cdot \frac{\partial \sigma_B^2}{\partial x_i}$

$$= \frac{\partial l}{\partial \hat{x}_i} \cdot \frac{\partial \hat{x}_i}{\partial x_i} + \left( \underbrace{\frac{\partial l}{\partial \hat{x}} \cdot \frac{\partial \hat{x}}{\partial \mu_B}}_{\text{to } \{x_1, \dots, x_m\}} + \frac{\partial l}{\partial \sigma_B^2} \frac{\partial \sigma_B^2}{\partial \mu_B} \right) \frac{\partial \mu_B}{\partial x_i} + \underbrace{\frac{\partial l}{\partial \hat{x}} \frac{\partial \hat{x}}{\partial \sigma_B^2} \frac{\partial \sigma_B^2}{\partial x_i}}$$

$$= \frac{\partial l}{\partial \hat{x}_i} \frac{1}{\sqrt{\sigma_B^2 + \epsilon}} + \left( \sum_{j=1}^m \frac{\partial l}{\partial \hat{x}_j} \cdot \frac{-1}{\sqrt{\sigma_B^2 + \epsilon}} + \frac{\partial l}{\partial \sigma_B^2} \cdot \left( \frac{-2}{m} \sum_{i=1}^m (x_i - \mu_B) \right) \frac{1}{m} + \sum_{j=1}^m \frac{\partial l}{\partial \hat{x}_j} \cdot \left( \frac{-1}{2} \right) \sum_{j=1}^m \frac{x_j - \mu_B}{(\sigma_B^2 + \epsilon)^{3/2}} \right) \frac{2}{m} (x_i - \mu_B)$$

↓  
cont. to next page

$$\begin{aligned}
&= \frac{\partial \ell}{\partial y_i} \cdot r \cdot \frac{1}{\sqrt{\sigma_B^2 + \varepsilon}} - \frac{\sum_{j=1}^m \frac{\partial \ell}{\partial x_j}}{m \sqrt{\sigma_B^2 + \varepsilon}} + \frac{\partial \ell}{\partial \sigma_B} \cdot \underbrace{\frac{1}{m} \sum_{i=1}^m (x_i - \mu_B)}_{\frac{1}{m} \sum_{i=1}^m x_i - \frac{1}{m} \sum_{i=1}^m \mu_B} - \frac{1}{m} \sum_{j=1}^m \frac{(x_j - \mu_B)}{(\sigma_B^2 + \varepsilon)^{3/2}} \frac{\partial \ell}{\partial x_j} (x_i - \mu_B) \\
&= \mu_B - \frac{m \mu_B}{m} = 0
\end{aligned}$$

$$\begin{aligned}
&= \frac{\partial \ell}{\partial y_i} \cdot r \cdot \frac{1}{\sqrt{\sigma_B^2 + \varepsilon}} - \frac{\sum_{j=1}^m \frac{\partial \ell}{\partial x_j}}{m \sqrt{\sigma_B^2 + \varepsilon}} - \frac{1}{m} \sum_{j=1}^m \frac{(x_j - \mu_B) \frac{\partial \ell}{\partial x_j}}{(\sigma_B^2 + \varepsilon)^{3/2}} (x_i - \mu_B) \\
&= \frac{1}{m \sqrt{\sigma_B^2 + \varepsilon}} \left( \frac{\partial \ell}{\partial y_i} r m \frac{\partial \ell}{\partial y} - \sum_{j=1}^m \frac{\partial \ell}{\partial x_j} - \sum_{j=1}^m (x_j - \mu_B) \frac{\partial \ell}{\partial x_j} (x_i - \mu_B) (\sigma_B^2 + \varepsilon)^{-1} \right) \\
&= \frac{1}{m \sqrt{\sigma_B^2 + \varepsilon}} \left( m \frac{\partial \ell}{\partial y_i} r - \sum_{j=1}^m \frac{\partial \ell}{\partial x_j} - \sum_{j=1}^m \frac{x_j - \mu_B}{\sqrt{\sigma_B^2 + \varepsilon}} \cdot \frac{\partial \ell}{\partial x_j} \cdot \frac{(x_i - \mu_B)}{\sqrt{\sigma_B^2 + \varepsilon}} \right) \\
&= \boxed{\frac{1}{m \sqrt{\sigma_B^2 + \varepsilon}} \left( m \frac{\partial \ell}{\partial y_i} r - \sum_{j=1}^m \frac{\partial \ell}{\partial x_j} - \hat{x}_i \sum_{j=1}^m \hat{x}_j \frac{\partial \ell}{\partial x_j} \right)} \quad \text{Q.E.D.}
\end{aligned}$$

$$\frac{\partial \ell}{\partial \gamma} = \frac{\partial \ell}{\partial y} \frac{\partial y}{\partial \gamma} = \boxed{\sum_{i=1}^m \frac{\partial \ell}{\partial y_i} r} \quad \text{Q.E.D.}$$

$$\frac{\partial \ell}{\partial \beta} = \frac{\partial \ell}{\partial y} \frac{\partial y}{\partial \beta} = \boxed{\sum_{i=1}^m \frac{\partial \ell}{\partial y_i}} \quad \text{Q.E.D.}$$

Q3 Given N-Dimensional vector of real number

$$\text{Softmax}(\vec{z}) = S(\vec{z}) : \begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_N \end{bmatrix} \rightarrow \begin{bmatrix} s_1 \\ s_2 \\ \vdots \\ s_N \end{bmatrix}$$

$$s_j = \frac{e^{z_j}}{\sum_{k=1}^N e^{z_k}} \quad \forall j \in 1 \dots N \Rightarrow s(z_j) = s_j = \frac{e^{z_j}}{\sum_{k=1}^N e^{z_k}}$$

$$L(y_i, \hat{y}_i) = -\sum_i y_i \log(\hat{y}_i) \quad \text{where } y_i \text{ is G.T. and } \hat{y}_i = S(\vec{z}_i)$$

$$(a) \text{ If } i=j, \text{ then } \frac{\partial s(z_i)}{\partial z_j} = \frac{\partial}{\partial z_j} \frac{e^{z_i}}{\sum_{k=1}^N e^{z_k}} = \frac{e^{z_i} \sum_{k=1}^N e^{z_k} - (e^{z_i})^2}{(\sum_{k=1}^N e^{z_k})^2}$$

$$= \frac{e^{z_i}}{\sum_{k=1}^N e^{z_k}} - \left( \frac{e^{z_i}}{\sum_{k=1}^N e^{z_k}} \right)^2 = s(z_i)(1 - s(z_j))$$

$$\text{If } i \neq j, \text{ then } \frac{\partial}{\partial z_j} \frac{e^{z_i}}{\sum_{k=1}^N e^{z_k}} = -e^{z_i} \left( \sum_{k=1}^N e^{z_k} \right)^{-2} e^{z_j} = -\frac{e^{z_i}}{\sum_{k=1}^N e^{z_k}} \cdot \frac{e^{z_j}}{\sum_{k=1}^N e^{z_k}}$$

$$= -s(z_i)s(z_j)$$

With respect to these 2 cases, we can create a general form

$$\boxed{\frac{\partial s(z_i)}{\partial z_j} = s(z_i)(s(z_j) - s(z_i)) \text{ where } s_{ij} = \begin{cases} 0, & \text{if } i \neq j \\ 1, & \text{if } i = j \end{cases}}$$

(b) Derive that  $\frac{\partial L}{\partial z_i} = \hat{y}_i - y_i$

$$\frac{\partial L}{\partial z_i} = \frac{\partial}{\partial z_i} - \sum_i y_i \log(s(z_i)) = -\sum_i y_i \frac{\partial \log(s(z_i))}{\partial z_i} = -\sum_i \frac{y_i}{\hat{y}_i} \frac{\partial s(z_i)}{\partial z_i}$$

$$= -\frac{y_j}{\hat{y}_j} \frac{\partial s(z_j)}{\partial z_i} - \sum_{i \neq j} \frac{y_i}{\hat{y}_i} \frac{\partial s(z_i)}{\partial z_i} = -\frac{y_j}{\hat{y}_j} s(z_j)(1 - s(z_j)) - \sum_{i \neq j} \frac{y_i}{\hat{y}_i} \cdot (-s(z_i)s(z_j))$$

$$= -y_j + y_j \hat{y}_j + \sum_{i \neq j} y_i \hat{y}_j = -y_j + \sum_i y_i \hat{y}_j = -y_j + \hat{y}_j \times \boxed{\sum_i y_i}$$

$$\boxed{\hat{y}_j - y_j}$$

Q.E.D.

It's a probability distribution

Q4. Given  $\Sigma = \frac{1}{n} \sum_{i=1}^n (x_i - \mu)(x_i - \mu)^T$  which is  $\in \mathbb{R}^{m \times m}$  is a symmetric semi-definite matrix  
 $\mu = \frac{1}{n} \sum_{i=1}^n x_i$  which is  $\in \mathbb{R}^m$

(a) Construct a set of points  $\vec{x}_1 \dots \vec{x}_n \in \mathbb{R}^m$  and find the relation between set of points and  $(\mu, \Sigma)$  and  $(\mu, \Sigma)$  is known.  $\vec{x}_1 \dots \vec{x}_n$  are generated identically independent distributed

<sol>

$\because \Sigma$  is a symmetric and semi-definite matrix

$\therefore$  We can do eigen decomposition on it

$$\text{That is } \Sigma = U \Lambda U^T = [u_1 \dots u_n] \begin{bmatrix} \lambda_1 & 0 & 0 & \cdots \\ 0 & \lambda_2 & 0 & \\ 0 & 0 & \lambda_3 & \\ \vdots & & & \ddots \\ 0 & 0 & 0 & \lambda_n \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ \vdots \\ u_n \end{bmatrix} = \sum_{i=1}^n u_i \lambda_i u_i^T$$

$$= \sum_{i=1}^n u_i \sqrt{\lambda_i} \sqrt{\lambda_i}^T u_i^T = \sum_{i=1}^n (u_i \sqrt{\lambda_i})(u_i \sqrt{\lambda_i})^T = \frac{1}{n} \sum_{i=1}^n (x_i - \mu)(x_i - \mu)^T$$

$$\Rightarrow \frac{1}{\sqrt{n}}(x_i - \mu) = u_i \sqrt{\lambda_i} \Rightarrow \boxed{x_i = \mu + \sqrt{n} u_i \sqrt{\lambda_i}}$$

Q.E.D.

(b). Let  $1 \leq k \leq m$ , and objective minimize  $\text{Trace}(\Phi^T \Sigma \Phi)$  subject to  $\Phi^T \Phi = I_k$ , variables  $\Phi \in \mathbb{R}^{m \times k}$

<sol>  $\text{Trace}(\Phi^T \frac{1}{n} X X^T \Phi) = \frac{1}{n} \text{Trace}(\Phi^T X X^T \Phi) = \frac{1}{n} \text{Trace}((\Phi^T X)(\Phi^T X)^T) = \frac{1}{n} \|\Phi^T X\|_F^2$

which is

$$\begin{bmatrix} \phi_1^T \\ \phi_2^T \\ \vdots \\ \phi_m^T \end{bmatrix} \begin{bmatrix} x_1 & \dots & x_n \end{bmatrix}_{m \times n} = \begin{bmatrix} \phi_1^T x_1 & \phi_1^T x_2 & \dots & \phi_1^T x_n \\ \vdots & & & \\ \phi_m^T x_1 & & & \phi_m^T x_n \end{bmatrix}$$

$\downarrow$   
 $\phi_1 \geq \phi_2 \geq \phi_3 \geq \dots \geq \phi_m$   
it's a increasing relationship

the physical meaning of this column is  
project  $x_i$  to  $\phi_1 \dots \phi_m$ .

But now we want to minimize  $\text{Trace}(\Phi^T \Sigma \Phi)$  by choosing the value from  $\phi_1$  to  $\phi_N$

so, the  $\Phi_{opt} = [\phi_N, \phi_{N-1} \dots \phi_{N-m+1}]$  which is the opposite concept with PCA

$\downarrow$   
begin from smallest

$\downarrow \phi_N \leq \phi_{N-1} \leq \phi_{N-2} \leq \dots \leq \phi_{N-m+1}$  which are decreasing relationship

$$\Rightarrow \boxed{\frac{1}{n} \sum_{i=1}^n \|\Phi^T x_i\|^2} = \frac{1}{n} \sum_{i=1}^n \|\text{project } x_i \text{ to span}(\phi_N \dots \phi_{N-m+1})\|$$

\* which is a problem  
similar with laplacian eigenmap.