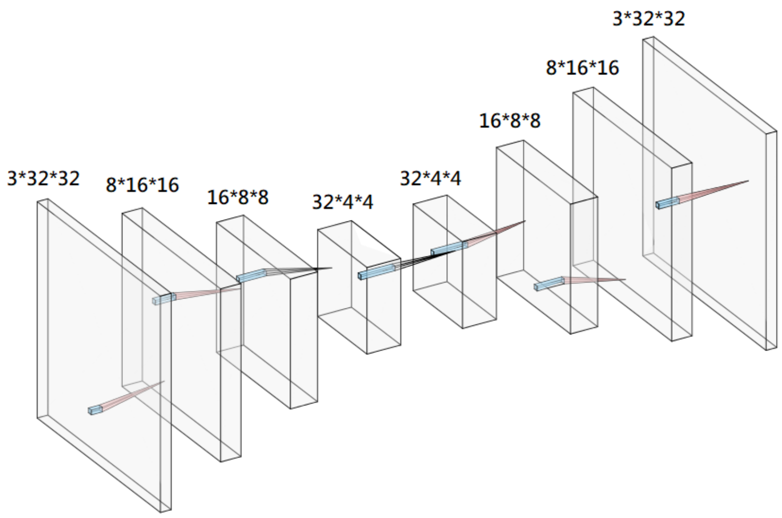
Machine Learning 2019 Fall
HW4 Report

---

b05902105 余友竹

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。

Model 1:



圖中的數字分別代表: $(C,\ H,\ W)$。

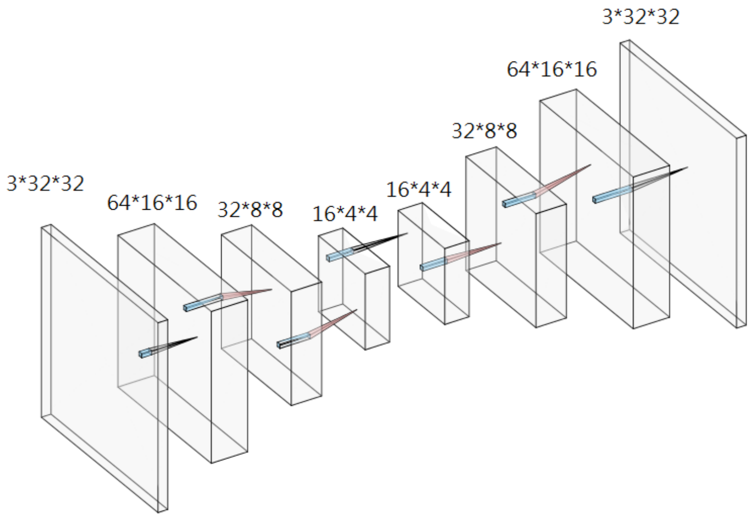
第一種模型我採用了對稱的架構， Filter數量由少至多，這也是最主流的做法，我認為可以從Activation Maximization的結果來解釋這樣的選擇。

從上次的作業中可以觀察到，隨著Hidden Layer數量增加，Weight的變化會越來越小，越難activate每個filter。所以在越深層的layer，給予比較多的filter，也就比較有機會activate更多的filter。

到中間的latent，攤平後的vector共有 $32 \times 4 \times 4$的維度。

由於中間層的維度夠多(512維)，reconstruction的loss不高，約為0.0568。

Model 2:



圖中的數字分別代表: $(C,\ H,\ W)$。

第二種模型同樣是對稱的架構，但不同的點是，我在第一層convolution layer就娶了64個filter，隨著layer越深，反而取了比較少的filter。

出乎意料之外的是，這樣的model也能work，而且效果不會差太多，我認為這樣的model可以用以下的說法解釋：

第一層的data即是最原始的data，沒有經過任何hidden layer做轉換，當我們對第一層的data取的資訊量太少(filter數量不夠)，即使後面增加filter的數量，得到的data都無法忠實呈現原始的data。

反之，在第一層就取了足夠多的data(filter數量夠多)，就得到了夠多的資訊可以呈現原始data，後面取的filter就有比較多的選擇，也能有比較正確的資訊。

透過這樣的model運作，中間的latent共有$16 \times 4 \times 4$的維度。

由於壓縮的維度比較多，reconstruct的loss比較高，約為0.0763。

接著我讓兩個model分別apply不同的降維方式，分別是PCA(whiten = True)以及T-SNE，下表紀錄兩個models通過不同的降維方法所得到的public/private accuracy。

PCA (whiten = True)

	PUBLIC	PRIVATE
Model 1	0.80000	0.79412
Model 2	0.80333	0.79269

T-SNE

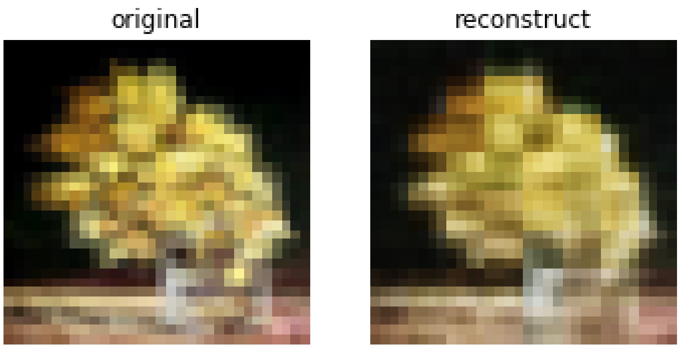
	PUBLIC	PRIVATE
Model 1	0.66095	0.65777
Model 2	0.68297	0.69270

有點意外的是T-SNE的表現普遍比較差，不過PCA在沒有加上whiten = True的選項時表現又會更差，而T-SNE隨機性比較高，有時很高，有時很低。

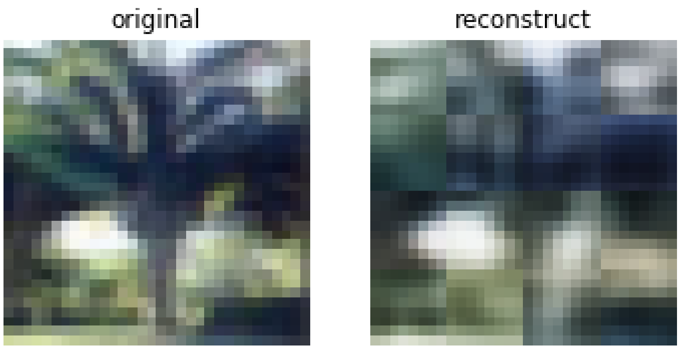
所以我認為這樣並不能代表T-SNE做的一定都會比較差。

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

Model 1



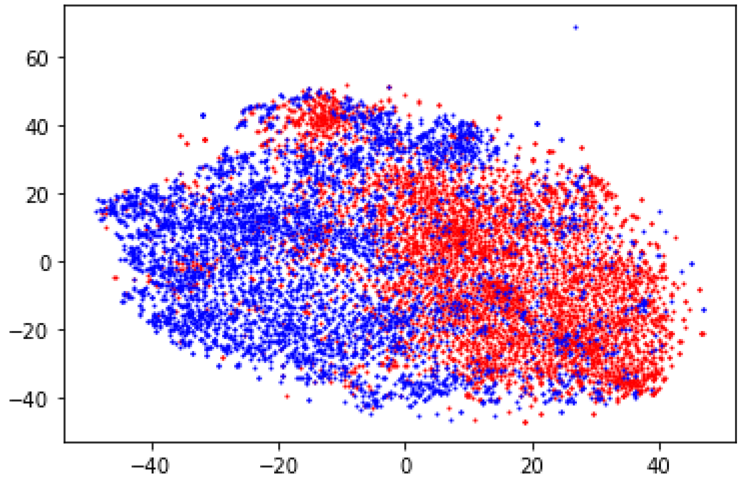
Model 2



明顯的第一個model reconstruct的效果較好，這是因為他的latent維度比較高，保有比較多的資訊，事實上model 1的loss也比較少，圖片中忠實反映了這兩個model的差別。

但事實上reconstruct的比較好完全不代表後續的clustering效果也會比較好，由第一題的結果我們可以看到，雖然相距不遠，但model 2的表現普遍優於model 1。

3. 請在二維平面上視覺化label的分佈。



上圖是使用了model1做第一次降維，再經過T-SNE壓縮至2維得到的結果。

可以看到，兩種照片的分布相當接近，很難真的區分出像MNIST手寫辨識那樣，分群成一塊一塊的樣子。

4. Math Problem

Principle Component Analysis

首先輸入數據

```
X = np.array([[1, 2, 3], [4, 8, 5], [3, 12, 9],
              [1, 8, 5], [5, 14, 2], [7, 4, 1],
              [9, 8, 9], [3, 8, 1], [11, 5, 6], [10, 11,
7]])
```

(a) principal axes

計算mean: $\mu = \frac{1}{N}\sum_{i=1}^{N} x_i$

```
mu = X.mean(axis=0)
print(mu)
```

```
[5.4 8.  4.8]
```

計算covariance matrix: $\Sigma = \frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)(x_i - \mu)^T$

```
sigma = (X - mu).T.dot(X - mu) / 10
print(sigma)
```

```
[[12.04  0.5   3.28]
 [ 0.5  12.2   2.9 ]
 [ 3.28  2.9   8.16]]
```

對Σ做正交對角化: $\Sigma = Q\Lambda Q^T$

```
L, Q = np.linalg.eigh(sigma)
```

重新排列順序

```
L = L[::-1]
Q = Q[:, [2, 1, 0]]
```

L為Σ的eigenvalue，所以Λ即為:

```
print(np.diag(L))
```

```
[[15.2974434   0.          0.        ]
 [ 0.         11.63052369  0.        ]
 [ 0.          0.          5.47203291]]
```

而Q的行向量為Σ的eigenvector

```
print(Q)
```

```
[[ 0.6165947  -0.67817891  0.39985541]
 [ 0.58881629  0.73439013  0.33758926]
 [ 0.52259579 -0.02728563 -0.85214385]]
```

可得principal axes分別為:

```
print('q1 = ', Q[:, 0], '\nq2 = ', Q[:, 1], '\nq3 = ',
Q[:, 2])
```

```
q1 =  [0.6165947  0.58881629 0.52259579]
q2 =  [-0.67817891  0.73439013 -0.02728563]
q3 =  [ 0.39985541  0.33758926 -0.85214385]
```

(b) principal components

將10個向量與Q相乘即可得到每個向量的principal components

```
print(X.dot(Q))
```

```
[[ 3.36201464  0.70874446 -1.48139761]
 [ 9.78988804  3.02597728  0.03941652]
 [13.61894165  6.53257419 -2.41865723]
 [ 7.94010395  5.06051399 -1.16014972]
 [12.37159312  6.83599606  5.02123906]
 [ 7.19402383 -1.83697744  3.29720109]
 [14.96324467 -0.47405978 -1.36988181]
 [ 7.0829102   3.81329871  3.0481365 ]
 [12.86219784 -3.95173109  0.97349277]
 [16.30109667  1.10550298  1.74702909]]
```

(c) reconstruction error

q_1, q_2為前2重要的principal axes，令$\hat{Q} = [q_1 \ \ q_2]^T$

將X透過$\hat{Q}$降維，再乘$\hat{Q}^T$反推回來，計算原本的x與reconstruct後的x的方均差。也就是：

$\text{err} = \frac{1}{N}\sum_{i=1}^{N} ||x_i - \hat{Q}^T\hat{Q}x_i)||^2$

```
P = Q[:, :2]
print(((X - X.dot(P).dot(P.T))**2).sum()/10)
```

```
6.064383051974751
```

Constrained Mahalanobis Distance Minimization Problem

(a)

1. Symmetric:

   $(AA^T)^T = AA^T$
   $(A^TA)^T = A^TA$

   Thus, both AA^T and A^TA are symmetric.

2. Positive Semi-Definite:

   $\forall\ \mathbf{y} \neq \mathbf{0},\ \mathbf{y}^TAA^T\mathbf{y} = (A^T\mathbf{y})^T(A^T\mathbf{y}) = ||A^T\mathbf{y}||^2 \geq 0$
   $\forall\ \mathbf{y} \neq \mathbf{0},\ \mathbf{y}A^TA\mathbf{y} = (A\mathbf{y})^T(A\mathbf{y}) = ||A\mathbf{y}||^2 \geq 0$

3. Shared Non-Zero Eigenvalues

   Theorem: 對於所有$A \in \mathbb{R}^{m\times n},\ B \in \mathbb{R}^{n\times m}$, AB與BA擁有相同的非0 eigenvalues

   Proof:

   Without Loss of Generality, 假設A, B不為O (若為O, $AB = BA = O$不存在非0 eigenvalues)。

   $$\begin{aligned} &\forall\lambda \in \lambda(AB) \neq 0,\ \exists\mathbf{v} \neq \mathbf{0} \\ \text{s.t. }& AB\mathbf{v} = \lambda\mathbf{v} \\ \Rightarrow\ & BAB\mathbf{v} = \lambda B\mathbf{v} \\ \Rightarrow\ & BA(B\mathbf{v}) = \lambda(B\mathbf{v}) \end{aligned}$$

   因為$B \neq O$, 且$\mathbf{v} \neq \mathbf{0}$，所以$B\mathbf{v} \neq \mathbf{0}$

因此$B\mathbf{v}$為BA相對於λ的一個eigenvector

$\Rightarrow \lambda \in \lambda(BA)$

同理可證，$\forall\, \lambda \in \lambda(BA) \neq 0,\ \lambda \in \lambda(AB)$

根據上述定理，AA^T與A^TA具有相同的非0特徵值。

(b)

令：

$\mathbf{z_1} = [\sqrt{m}\; 0\; \cdots\; 0]^T,$
$\mathbf{z_2} = [-\sqrt{m}\; 0\; \cdots\; 0]^T,\ \cdots$
$\mathbf{z_{2m-1}} = [0\; \cdots\; \sqrt{m}]^T,$
$\mathbf{z_{2m}} = [0\; \cdots\; -\sqrt{m}]^T,\ \mathbf{z_i} \in \mathbb{R}^m,\ \forall\, 1 \leq i \leq m$

則 $\mathbf{z_1},\ \cdots,\ \mathbf{z_{2m}}$的mean為: $\frac{1}{2m}\sum_{i=1}^{2m}\mathbf{z_i} = \mathbf{0}$

covariance matrix 為: $\frac{1}{2m}\sum_{i=1}^{2m}(\mathbf{z_i}-\mathbf{0})(\mathbf{z_i}-\mathbf{0})^T = I_m$

$\because \Sigma$ is positive semi-definite, $\therefore \exists A \in \mathbb{R}^{m\times m}$ s. t. $\Sigma = AA^T$

取$\mathbf{x_k} = A\mathbf{z_k} + \mu$ ，則, $\mathbf{x_1},\ \cdots,\ \mathbf{x_{2m}}$的mean為:

$$\begin{aligned}\frac{1}{2m}\sum_{i=1}^{2m}\mathbf{x_i} &= \frac{1}{2m}\sum_{i=1}^{2m}(A\mathbf{z_i}+\mu)\\ &= A(\frac{1}{2m}\sum_{i=1}^{2m}\mathbf{z_i}) + \mu\\ &= A\cdot\mathbf{0} + \mu = \mu\end{aligned}$$

covariance matrix為:

$$\begin{aligned}\frac{1}{2m}\sum_{i=1}^{2m}(\mathbf{x_i}-\mu)(\mathbf{x_i}-\mu)^T &= \frac{1}{2m}\sum_{i=1}^{2m}(A\mathbf{z_i}+\mu-\mu)(A\mathbf{z_i}+\mu-\mu)^T\\ &= \frac{1}{2m}\sum_{i=1}^{2m}(A\mathbf{z_i})(A\mathbf{z_i})^T\\ &= \frac{1}{2m}\sum_{i=1}^{2m}A\mathbf{z_i}\mathbf{z_i}^TA^T\\ &= A\frac{1}{2m}\sum_{i=1}^{2m}\mathbf{z_i}\mathbf{z_i}^TA^T\\ &= A\cdot I_m\cdot A^T = AA^T = \Sigma\end{aligned}$$

3. multiclass AdaBoost.

根據老師提供的方法.

Init: $g_0^k = 0$, $\forall k = 1, \dots, K$.

Iteration:

for $l = 1, \dots, \frac{T}{K} + 1$

for $k = 1, \dots, K$.

$t = l \cdot k + k$.

$f_t = \underset{f}{\operatorname{argmin}} \frac{\partial}{\partial \alpha} L(g_{t-1}^1, \dots, g_{t-1}^k + \alpha f, \dots, g_{t-1}^K)$.

$\alpha_t = \underset{\alpha}{\operatorname{argmin}} \frac{\partial}{\partial \alpha} L(g_{t-1}^1, \dots, g_{t-1}^k + \alpha f_t, \dots, g_{t-1}^K)$.

Return: $g_T^k = \sum_{t=1}^{T} \alpha_t^k f_t$, $\forall k = 1, \dots, K$.

$h(x) = \underset{1 \le k \le K}{\operatorname{argmax}} g_T^k(x)$.

因 F is finite, 可將所有 $\in F$ 代入上式.

求出 $f_t = \underset{f \in F}{\operatorname{argmin}} \frac{\partial}{\partial \alpha} L(g_{t-1}^1, \dots, g_{t-1}^k + \alpha f, \dots, g_{t-1}^K)$

再令 $\frac{\partial}{\partial \alpha} L(g_{t-1}^1, \dots, g_{t-1}^k + \alpha f_t, \dots, g_{t-1}^K) = 0$.

求解 α, 即可最小化 $L(g_T^1, \dots, g_T^K)$ #.