學號：b10705025　系級：資管三 姓名：彭鈞道

我因為最好的model跟訓練參數忘記存下來，所以Kaggle競賽上最後選的是用public score 0.774的模型，檢查過了對分數沒有影響，再請助教檢查，謝謝。造成困擾非常抱歉。模型的連結在 https://drive.google.com/file/d/1Kh6EKQSnrutE4Cf7MrROMqTXgbTsK15e/view?usp=sharing

Kaggle Competetion

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。

   Auto-encoder dimension: 32, PCA: 12

   Auto-encoder dimension: 32, PCA: 8

   Auto-encoder dimension: 24, PCA: 8

   Auto-encoder dimension: 24, PCA: 6

   Auto-encoder dimension: 32 PCA: 6

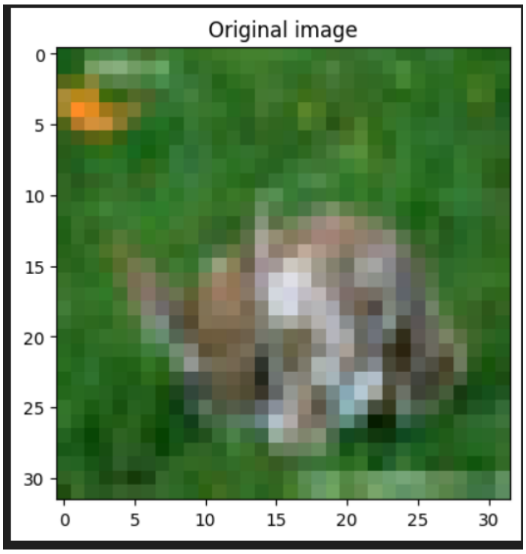
   最好的當初是用Auto-encoder dimension: 24, PCA: 6，但之後寫report才發現Auto-encoder dimension: 24, PCA: 8比較好。

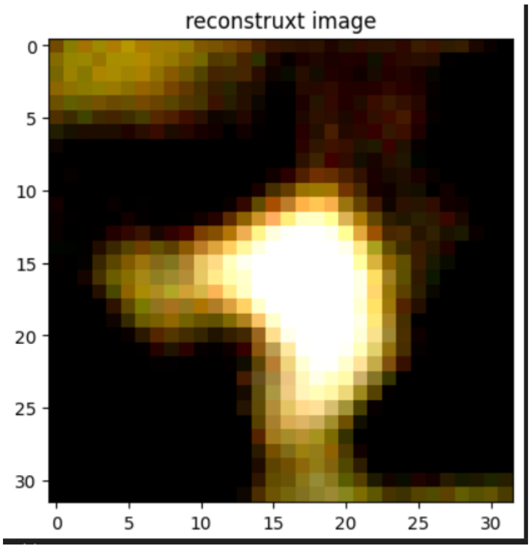
2. (1%) 從 trainX.npy 選出不同類別的 2 張圖，貼上原圖以及你的 autoencoder reconstruct 的圖片。用 Mean Square Error 計算這兩張圖的 reconstruction error, 並說明該 error 與 kaggle score 的關係。

註1: 所以一共要貼上4張圖片。

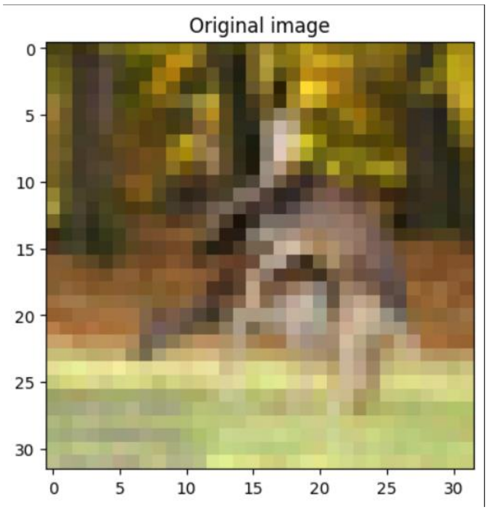
註2: 原圖請貼上做 augmentation 之前的版本。

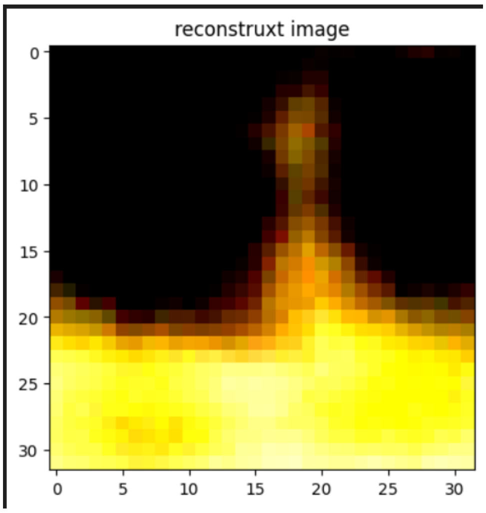
id = 0, label =0, error : 0.5516





id = 32, label =1, error : 0.2425





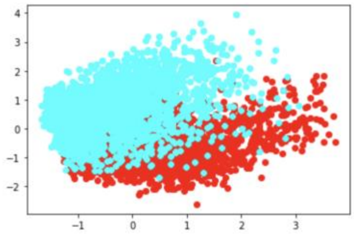
其中error是將圖片標準化後的結果，可以發現error較低的被分到label 1，error較高的被分到label 0，代表kaggle score取決於model對每張圖片的error。

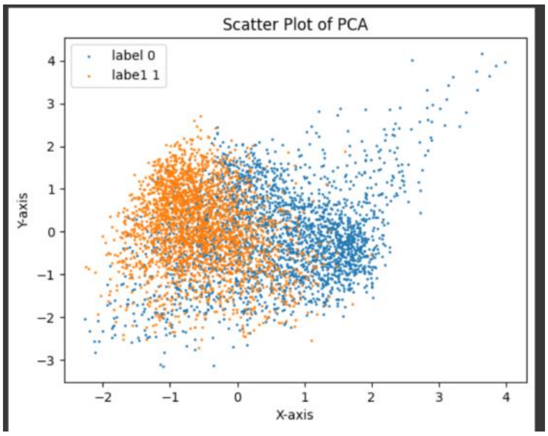
3. (2%) 請使用 pca 以及 tsne **兩種**方法, 將 visualization.npy 的圖片經過 autoencoder 降維後得到之 latent vector，進一步降維至二維平面並作圖。並說明兩張圖之差異。

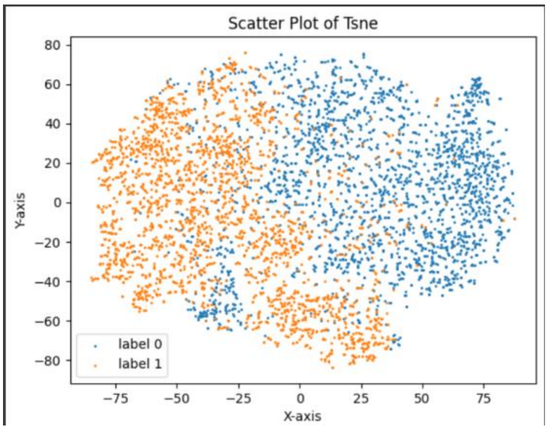
註1: visualization.npy 前 2500 張 label 為 0 ; 後 2500 張 label 為 1

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）







可以發現PCA的降維可以找到軸線來分割，且降維的分布可能就是由兩個PCA AXIS 來決定。TSNE的降維結果也可以大致用一條直線分割，但是分布沒有PCA來的分散，圖片也較近似圓形(獲曲面)。

4. (6%) Refer to math problem：

Math 1

hw3 1.

$$z = W \cdot x + b \qquad z^f = W_f \cdot x + b_f$$
$$z^i = W_i \cdot x + b_i \qquad z^o = W_o \cdot x + b_o$$
$$f(z) = \frac{1}{1+e^{-z}}, \quad g(z) = z, \quad h(z) = z$$

$$W = [0,0,1,0], \quad b = 0$$
$$W_i = [50,50,0,0], \quad b_i = -5$$
$$W_f = [-50,-50,0,0], \quad b_f = 120$$
$$W_o = [0,0,200,0], \quad b_o = -30$$

$$x^1 = [0,0,1,3] \Rightarrow z = 1+0 = 1, \quad z^i = -5$$
$$z^f = 120, \quad z^o = 200-30 = 170$$
$$c' = f(-5) \cdot g(1) + c f(z^f)$$
initial $c = 0$
$$= f(-5) \cdot 1 = \frac{1}{1+e^5}$$
$$y_1 = f(z^o) \cdot h(c')$$
$$= \frac{1}{1+e^{-170}} \cdot \frac{1}{1+e^5}$$

$$x^2 = [0,1,-1,2] \Rightarrow z = -1, \quad z^i = 45$$
$$z^f = 70, \quad z^o = -230$$
$$c' = f(z^i) g(z) + c f(z^f)$$
$$= \frac{1}{1+e^{-45}} \cdot -1 + \frac{1}{1+e^5} \cdot \frac{1}{1+e^{-70}}$$
$$= \frac{1}{1+e^5} \cdot \frac{1}{1+e^{-70}} - \frac{1}{1+e^{-45}}$$
$$y_2 = \frac{1}{1+e^{230}} \cdot \left(\frac{1}{1+e^5} \cdot \frac{1}{1+e^{-70}} - \frac{1}{1+e^{-45}} \right)$$

Math 1

hw1.

$x^3 = [2,1,3,4] \Rightarrow z = 3, \ z^i = 145$

$z^f = -30, \ z_0 = 570$

$c' = f(z^i)g(z) + cf(z^f)$

$= \frac{1}{1+e^{-145}} \cdot 3 + \left(\frac{1}{1+e^5} \cdot \frac{1}{1+e^{-70}} - \frac{1}{1+e^{-45}}\right) \cdot \frac{1}{1+e^{30}}$

$y_3 = \frac{1}{1+e^{-570}} \cdot \left(\frac{3}{1+e^{-145}} + \left(\frac{1}{1+e^{30}} \left(\frac{1}{1+e^5} \cdot \frac{1}{1+e^{-70}} - \frac{1}{1+e^{-45}}\right)\right)\right)$

$x_4 = [0,1,0,0] \Rightarrow z = 0, \ z^i = 45$

$z^f = 70, \ z_0 = -30$

$c' = cf(z^f)$

$= \left(\frac{3}{1+e^{-145}} + \frac{1}{1+e^{30}} \left(\frac{1}{1+e^5} \cdot \frac{1}{1+e^{-70}} - \frac{1}{1+e^{-45}}\right)\right) \frac{1}{1+e^{-70}}$

$y_4 = \frac{1}{1+e^{30}} \cdot \frac{1}{1+e^{-70}} \left(\frac{3}{1+e^{-145}} + \frac{1}{1+e^{30}} \left(\frac{1}{1+e^5} \cdot \frac{1}{1+e^{-70}} - \frac{1}{1+e^{-45}}\right)\right) \cdot \frac{1}{1+e^{-70}}$

Math 2.1, 2.2, 2.5

hw3

2.1

$$W = \begin{bmatrix} 0&1&1&1&0&0&0&0&0&0 \\ 1&0&0&1&0&0&0&1&0&0 \\ 1&0&0&0&0&0&0&1&0&0 \\ 1&1&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&1&1&0&0&0 \\ 0&0&0&0&1&0&0&0&0&0 \\ 0&0&0&0&1&0&0&0&0&1 \\ 0&1&1&0&0&0&0&0&1&0 \\ 0&0&0&0&0&0&0&1&0&1 \\ 0&0&0&0&0&0&1&0&1&0 \end{bmatrix}$$

2.2 $D = \mathrm{diag}(3, 3, 2, 2, 2, 1, 2, 3, 2, 2)$

$$L = \begin{bmatrix} 3&-1&-1&-1&0&0&0&0&0&0 \\ -1&3&0&-1&0&0&0&-1&0&0 \\ -1&0&2&0&0&0&0&-1&0&0 \\ -1&-1&0&2&0&0&0&0&0&0 \\ 0&0&0&0&2&-1&-1&0&0&0 \\ 0&0&0&0&-1&1&0&0&0&0 \\ 0&0&0&0&-1&0&2&0&0&-1 \\ 0&-1&-1&0&0&0&0&3&-1&0 \\ 0&0&0&0&0&0&0&-1&2&-1 \\ 0&0&0&0&0&0&-1&0&-1&2 \end{bmatrix}$$

2.5 $L = (D - W)$

$$\Rightarrow L_{ij} = \frac{\sum_{i=1}^{n} W_{ij} + W_{ji}}{2} \text{ if } i = j \ (W_{ij} = 0)$$

$$= -W_{ij} \text{ if } i \neq j$$

since $W^T = W$, $L_{ij} = \sum_{i=1}^{n} W_{ij}$ if $i \neq j$

the sum of a row $\sum_{j=1}^{n} L_{ij} = \sum_{j=1}^{n} W_{ij} - \sum_{j=1}^{n} W_{ij} = 0$

since $L^T = L$, sum of a culumn $= 0$

so vector $\begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix}$ is obvious an eigenvector with 0 eigenvalue 0 (add on column's and row's sum will result in $\begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix}$, so $\begin{bmatrix} c \\ \vdots \\ c \end{bmatrix}$ is the eigenvector .

Math 2.3

先利用PCA算出降維的axis

```
from sklearn.decomposition import PCA
num_components = 10
pca = PCA(n_components=num_components)
pca.fit(L)
transformed_matrix = pca.transform(L*np.linalg.inv(D))
```

```
transformed_matrix
```

```
array([[ 0.44057589,  0.57969109, -0.3618034 , -0.16008427,  0.22250433,
         0.08679345,  0.1381966 , -0.200428  , -0.31146667, -0.31622777],
       [-0.54221265,  0.03268234, -0.3618034 ,  0.5059795 , -0.24971223,
        -0.16861949,  0.1381966 , -0.14826893, -0.29351261, -0.31622777],
       [-0.3458209 , -0.17045596,  0.1381966 , -0.18271555,  0.30114772,
         0.62882139, -0.3618034 , -0.06919254, -0.27762167, -0.31622777],
       [ 0.03460475, -0.33673606,  0.4472136 , -0.29675578,  0.05000608,
        -0.35330707,  0.4472136 , -0.255985  , -0.32325305, -0.31622777],
       [ 0.0128974 , -0.23004797, -0.3618034 , -0.42163549, -0.482972  ,
         0.23046679,  0.1381966 , -0.18298997,  0.44639991, -0.31622777],
       [-0.00327588,  0.08161909,  0.1381966 ,  0.19469785,  0.31278713,
        -0.29993088, -0.3618034 , -0.50524978,  0.51219985, -0.31622777],
       [-0.03460475,  0.33673606,  0.4472136 ,  0.29675578, -0.05000608,
         0.35330707,  0.4472136 ,  0.255985  ,  0.32325305, -0.31622777],
       [ 0.57512624, -0.26970735,  0.1381966 ,  0.37305549, -0.3863563 ,
         0.05884181, -0.3618034 ,  0.1061755 , -0.20811191, -0.31622777],
       [-0.22602946,  0.35854423,  0.1381966 , -0.38503778, -0.22757855,
        -0.38773232, -0.3618034 ,  0.46826682, -0.02646627, -0.31622777],
       [ 0.08873936, -0.38232545, -0.3618034 ,  0.07574026,  0.5101799 ,
        -0.14864075,  0.1381966 ,  0.53168691,  0.15857937, -0.31622777]])
```

挑出最後面的三個維度，並算出ψ，和 Z，跟最小化的答案(0.52156)

```
eigen_last10= transformed_matrix[:,9]
eigen_last9= transformed_matrix[:,8]
eigen_last8= transformed_matrix[:,7]
a_inv = np.sqrt(np.linalg.inv(D))
```

```
posi = np.array([eigen_last10, eigen_last9, eigen_last8]).T
posi = np.matmul(a_inv, posi)
```

```
print(np.trace(np.matmul(np.matmul(posi.T,L),posi)))
```

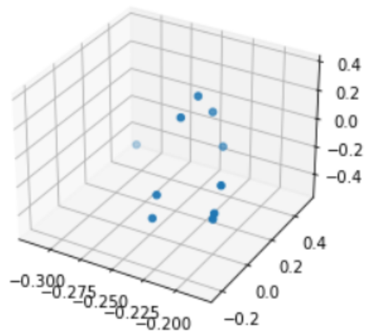
```
0.5215585365092121
```

```
z = posi.T
z
```

```
array([[-0.18257419, -0.18257419, -0.2236068 , -0.2236068 , -0.2236068 ,
        -0.31622777, -0.2236068 , -0.18257419, -0.2236068 , -0.2236068 ],
       [-0.17982536, -0.16945958, -0.19630817, -0.22857442,  0.3156524 ,
         0.51219985,  0.22857442, -0.12015347, -0.01871448,  0.11213255],
       [-0.11571716, -0.08560311, -0.04892652, -0.18100873, -0.12939345,
        -0.50524978,  0.18100873,  0.06130045,  0.33111464,  0.37595942]])
```

作圖並檢查限制式

```
import matplotlib.pyplot as plt
fig = plt.figure()
ax = fig.add_subplot(projection='3d')
ax.scatter(z[0], z[1], z[2])
from numpy.linalg import matrix_rank
plt.show()
```




```
print(np.matmul(np.matmul(posi.T,D),posi))
```

```
[[ 1.00000000e+00 -1.76077312e-16 -2.07824327e-16]
 [-1.76077312e-16  1.00000000e+00  3.84196992e-16]
 [-2.07824327e-16  3.84196992e-16  1.00000000e+00]]
```

Math 2.4

把選取的維度改成第六第七第八，並重複動作

```
eigen_last10= transformed_matrix[:,6]
eigen_last9= transformed_matrix[:,7]
eigen_last8= transformed_matrix[:,8]
a_inv = np.sqrt(np.linalg.inv(D))
```

```
posi = np.array([eigen_last10, eigen_last9, eigen_last8]).T
posi = np.matmul(a_inv, posi)
```

```
print(np.trace(np.matmul(np.matmul(posi.T,L),posi)))
```

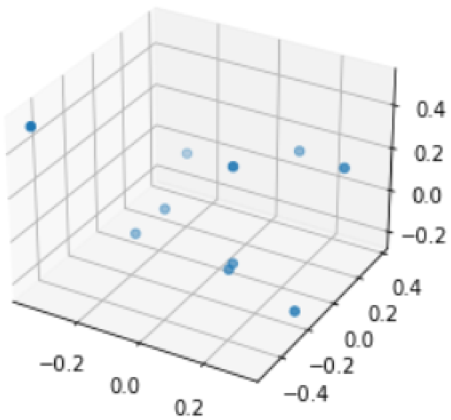
```
1.2484052511181407
```

```
z = posi.T
z
```

```
array([[ 0.07978784,  0.07978784, -0.25583364,  0.31622777,  0.09771975,
        -0.3618034 ,  0.31622777, -0.20888729, -0.25583364,  0.09771975],
       [-0.11571716, -0.08560311, -0.04892652, -0.18100873, -0.12939345,
        -0.50524978,  0.18100873,  0.06130045,  0.33111464,  0.37595942],
       [-0.17982536, -0.16945958, -0.19630817, -0.22857442,  0.3156524 ,
         0.51219985,  0.22857442, -0.12015347, -0.01871448,  0.11213255]])
```

算出來結果是1.2484不是1.098，接著作圖跟檢查限制

```python
import matplotlib.pyplot as plt
fig = plt.figure()
ax = fig.add_subplot(projection='3d')
ax.scatter(z[0], z[1], z[2])
from numpy.linalg import matrix_rank
plt.show()
```




```python
print(np.matmul(np.matmul(posi.T,D),posi))
```

```
[[ 1.00000000e+00 -3.29080263e-16  2.45306846e-16]
 [-3.29080263e-16  1.00000000e+00  3.84196992e-16]
 [ 2.38367952e-16  3.84196992e-16  1.00000000e+00]]
```

符合限制式

Math 2.6 2.7 2.8

2.6 $\frac{1}{2}\sum_{ij}(f_i-f_j)^2 W_{ij}$

$$= \frac{1}{2}\sum_{i,j}(f_i^2+f_j^2-2f_if_j)W_{ij}$$

$$= \frac{1}{2}\Big(\sum_i f_i^2 D_{ii} + \sum_j f_j^2 D_{jj} - 2\sum_{ij} f_if_jW_{ij}\Big)$$

$$= \frac{1}{2}\left(f^TDf + f^TDf - 2f^TLf\right)$$

$$= \frac{1}{2}\times 2\left(f^T(D-L)f\right) = f^TLf$$

2.7 if f is an eigenvector of L eigenvalue $=0$

$$f^TLf = f^T\cdot 0 f = 0$$

2.8 G is connected, x is the eigenvector of L of eigenvalue $=0$.

$$Lx = 0$$

$$x^TLx = \sum_{i,j\in \text{Graph}}(x(i)-x(j))^2 = 0$$

$\Rightarrow x(i)=x(j)$, since every pair i,j is connected, so x should be a constant vector. $\Rightarrow$ only one eigenvector for eigenvalue $=0$

Math 3 (ft)

hw3

3. $g_{t+1} = g_t + \alpha f_t$

$$f_t = \underset{f_t}{\operatorname{argmin}} \frac{\partial}{\partial \alpha} L(g_{t+1})$$

$$\frac{\partial}{\partial \alpha} L(g_{t+1})$$

$$= \frac{\partial}{\partial \alpha} \sum_{i=1}^{m} e^{\left(\frac{1}{K-1}\sum_{k \neq \hat{y}_i} g_t^k(x_i) - g_t^{\hat{y}_i}(x_i) + \alpha f_t^k(x_i) - \alpha f_t^{\hat{y}_i}(x_i)\right)}$$

$$= \sum_{i=1}^{m} e^{\frac{1}{K-1}\sum_{k \neq \hat{y}_i} g_t^k(x_i) - g_t^{\hat{y}_i}(x_i) + \alpha f_t^k(x_i) - \alpha f_t^{\hat{y}_i}(x_i)} \cdot \frac{1}{K-1}\sum_{k \neq \hat{y}_i} \left(f_t^k(x_i) - f_t^{\hat{y}_i}(x_i)\right)$$

when $\alpha \to 0$

let $Z_t = \sum_{i=1}^{m} e^{\frac{1}{K-1}\sum_{k \neq \hat{y}_i} g_t^k(x_i) - g_t^{\hat{y}_i}(x_i)}$

$$= Z_t \sum_{i=1}^{m} \boxed{\frac{\exp\left(\frac{1}{K-1}\sum_{k \neq \hat{y}_i} g_t^k(x_i) - g_t^{\hat{y}_i}(x_i)\right)}{Z_t}} \frac{1}{K-1}\sum_{k \neq \hat{y}_i} f_t^k(x_i) - f_t^{\hat{y}_i}(x_i)$$

→ can be views an a propability distribution D_t

$$= Z_t \mathop{\mathbb{E}}_{i \sim D_t}\left[\frac{1}{K-1}\sum_{k \neq \hat{y}_i} f_t^k(x_i) - f_t^{\hat{y}_i}(x_i)\right]$$

$$= Z_t \times \frac{1}{K-1} \mathop{\mathbb{E}}_{i \sim D_t}\left[K-1_{\text{ if } f_t(x_i) \neq y_i} - K-1_{f_t(x_i) = y_i}\right]$$

$$= Z_t \mathop{\mathbb{E}}_{i \sim D_t}\left[2_{f_t(x_i) \neq y_i} - 1\right]$$

to minimize it, $f_t(x_i) \neq y_i$'s probability must be minimized ⇒ f_t must be the classifier minimizes error rate

Math 3(alpha)

hw3-3. α_t is the minimal solution to

$$\sum_{i=1}^{m} \exp\left(\frac{1}{K-1}\sum_{k\neq y_i} g_t^k(x_i) - g_t^{\hat{y}_i}(x_i) + \alpha f_t^k(x_i) - \alpha f_t^{\hat{y}_i}(x_i)\right)$$

$$= Z_t \sum_{i=1}^{m} \underset{i\sim D_t}{D_t(i)} \cdot \exp\left(\frac{\alpha}{K-1}\sum_{k\neq y_i} f_t^k(x_i) - f_t^{\hat{y}_i}(x_i)\right)$$

$$= Z_t \mathop{E}_{i\sim D_t}\left[e^{\frac{\alpha}{K-1}\sum_{k\neq y_i} f_t^k(x_i) - f_t^{\hat{y}_i}(x_i)} \right]$$

$$= Z_t \mathop{E}_{i\sim D_t}\left[1_{f_t(x_i)\neq y_i} \cdot e^{\frac{\alpha}{K-1}\cdot K-1} + 1_{f_t(x_i)=y_i} e^{\frac{\alpha}{K-1}\cdot -(K-1)} \right]$$

$$= Z_t\left[\varepsilon_t e^{\alpha} + (1-\varepsilon_t) e^{-\alpha} \right]$$

take $\frac{\partial}{\partial\alpha}$

$$\frac{\partial}{\partial\alpha} Z_t\left[\varepsilon_t e^{\alpha} + (1-\varepsilon_t) e^{-\alpha} \right] = 0$$

$$Z_t\left[\varepsilon_t e^{\alpha} - (1-\varepsilon_t) e^{-\alpha} \right] = 0$$

$$Z_t \cdot e^{-\alpha}\left[\varepsilon_t e^{2\alpha} - (1-\varepsilon_t) \right] = 0$$

$$e^{\alpha} = \sqrt{\frac{1-\varepsilon_t}{\varepsilon_t}} \quad \alpha = \log\sqrt{\frac{1-\varepsilon_t}{\varepsilon_t}}$$

the same in 2 class adaboost

Math 4

hw 3

4.

$$h_1 = \tanh(W_x x_1 + W_h \overset{=0}{h_0}) = \tanh(W_x x_1)$$

$$h_2 = \tanh(W_x x_2 + W_h h_1)$$

$$= \tanh(W_x x_2 + W_h \cdot \tanh(W_x x_1))$$

$$\hat{y} = \frac{1}{1+e^{-W_o h_2}}$$

$$\frac{\partial L(y,\hat{y})}{\partial W_o} = \frac{\partial L(y,\hat{y})}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial W_o} = \frac{\partial L(y,\hat{y})}{\partial \hat{y}} \cdot \frac{\sigma(W_o h_2)}{\partial W_o}$$

$$= \frac{\partial L(y,\hat{y})}{\partial \hat{y}} \cdot \sigma(W_o h_2) \cdot (1-\sigma(W_o h_2)) \cdot h_2$$

$$\frac{\partial L(y,\hat{y})}{\partial h_2} = \frac{\partial L(y,\hat{y})}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial h_2} = \frac{\partial L(y,\hat{y})}{\partial \hat{y}} \cdot \frac{\sigma(W_o h_2)}{\partial h_2}$$

$$= \frac{\partial L(y,\hat{y})}{\partial \hat{y}} \cdot \sigma(W_o h_2)(1-\sigma(W_o h_2)) \cdot W_o$$

$$\frac{\partial h_2}{\partial W_h} = \frac{\partial(\tanh(W_x x_2 + W_h \cdot h_1))}{\partial W_h}$$

$$= (1-\tanh^2(W_x x_2 + W_h h_1)) \cdot h_1$$

$$\frac{\partial h_2}{\partial W_x} = \frac{\partial \tanh(W_x x_2 + W_h \cdot \tanh(W_x x_1))}{\partial W_x}$$

$$= (1-h_2^2) \cdot \left(x_2 + \frac{\partial W_h \cdot \tanh(W_x \cdot x_1)}{\partial W_x}\right)$$

Math 4

$$= (1-h_2^2) \cdot (x_2 + w_h (1-h_1^2) \cdot x_1)$$

$$\frac{\partial L}{\partial w_o} = \frac{\partial L(y,\hat{y})}{\partial \hat{y}} \cdot \sigma(w_o h_2)(1-\sigma(w_o h_2)) \cdot h_2$$

$$\frac{\partial L}{\partial w_h} = \frac{\partial L}{\partial h_2} \cdot \frac{\partial h_2}{\partial w_h} = \frac{\partial L(y,\hat{y})}{\partial \hat{y}} \cdot \sigma(w_o h_2)$$

$$\cdot (1-\sigma(w_o h_2)) \cdot w_o \cdot (1-h_2^2) \cdot h_{21}$$

$$\frac{\partial L}{\partial w_x} = \frac{\partial L}{\partial h_2} \cdot \frac{\partial h_2}{\partial w_x} = \frac{\partial L(y,\hat{y})}{\partial \hat{y}} \cdot \sigma(w_o h_2)(1-\sigma(w_o h_2))$$

$$\cdot w_o \cdot (1-h_2^2)(x_2 + w_h(1-h_1^2) \cdot x_1)$$

Math 5.a

hw3

5.(a)

(i)

$N_{left}=50,\ P^1_{left}=0.8,\ P^2_{left}=0.2,\ N_{right}=50$

$P^1_{right}=0.75,\ P^2_{right}=0.25$

$$\text{Gini} = \frac{1}{2}\left(1-\left((0.8)^2+(0.2)^2\right)\right) + \frac{1}{2}\left(1-(0.75)^2-0.25^2\right) = 0.3475$$

$$\text{Shannon information gain} = \frac{1}{2}\left(-0.8\cdot\log_2 0.8 - 0.2\log_2 0.2\right) + \frac{1}{2}\left(-0.75\log_2 0.75 - 0.25\log_2 0.25\right)$$

$$= 0.7666$$

(ii)

$N_{left}=80,\ P^1_{left}=0,\ P^2_{left}=100\%$

$N_{right}=20,\ P^1_{right}=0.9\ \ P^2_{right}=0.1$

$$\text{Gini} = 0.8\times(1-1^2)+0.2\left(1-(0.1)^2-(0.9)^2\right) = 0.036$$

$$\text{Shannon} = 0.8\left(-1\log 1-0\log 0\right)+0.2\left(-0.1\log_2 0.1 - 0.9\log_2 0.9\right)$$

$$= 0.09379$$

0.008638

0.02736

Math 5.b

(iii) $N_{left} = 90$, $p^1_{left} = 0.01$, $p^2_{left} = 0.99$
$N_{right} = 10$, $p^1_{right} = 100\%$

Gini : $0.9 \times (1 - (0.01)^2 - (0.99^2))$
$= 0.01782$

Shannon: $0.9(-0.01\log_2 0.01 - 0.99\log_2 0.99) + 0$
$= 0.07271$

(b) Assume for binary classification
choose left for negative and right part for positive
Only care about for false negative
Let Loss function
$= \frac{N_{left}}{N}(1 - P^{negative}_{left})$

Loss of (i) $= \frac{1}{2}(1 - 0.2) = 0.4$
Loss of (ii) $= 0$
Loss of (iii) $= 0.9(1 - 0.99) = 0.009$