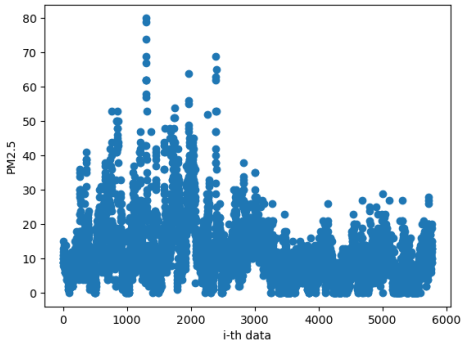
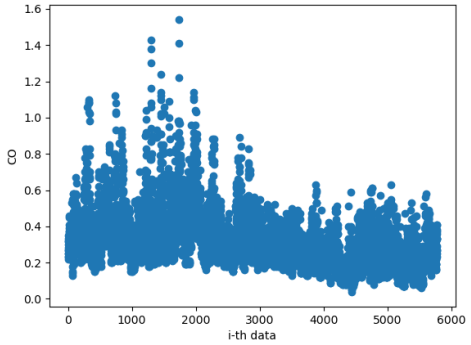
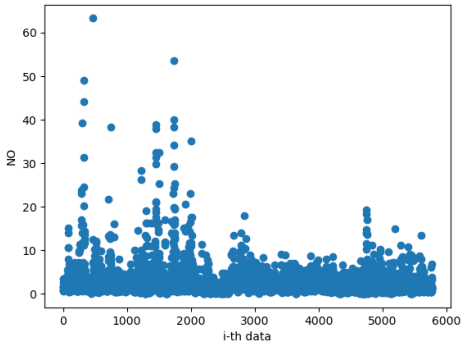
學號：R11921A16 系級： 電機碩一 姓名：何秉學

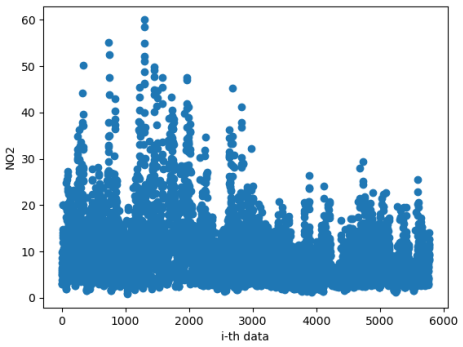
1. (1%) 解釋什麼樣的 data preprocessing 可以 improve 你的 training/testing accuracy，e.g., 你怎麼挑掉你覺得不適合的 data points。請提供數據(例如 kaggle public score RMSE)以佐證你的想法。

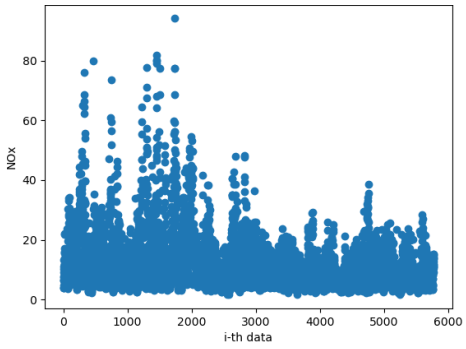
   After observing the training data visualized image, you can be aware of the relationship between the PM2.5 feature and the others.
   For instance, the CO image, NO image, NO2 image, and NOx image are much more correlated with PM2.5.



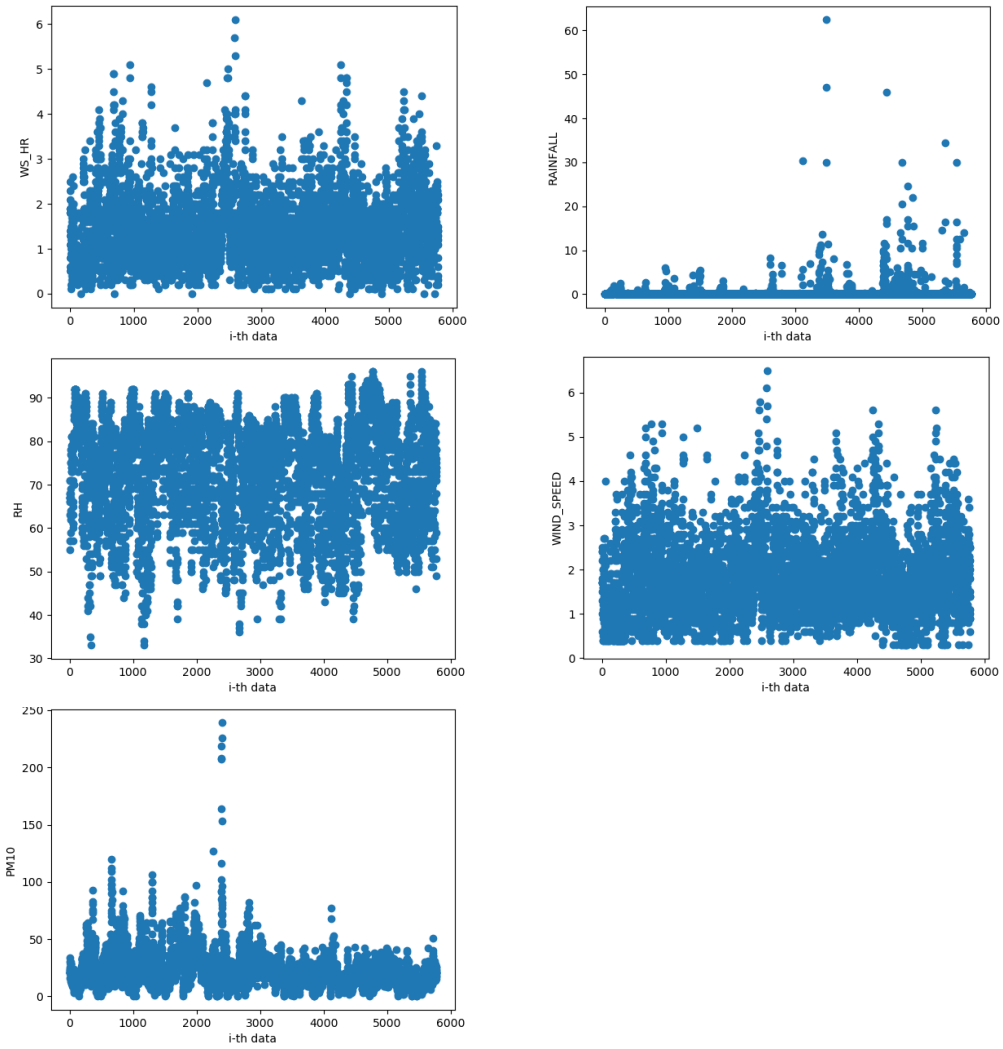








   I also choose PM10, WS_HR, RAINFALL, RH, WIND_SPEED which are also correlated to PM2.5 but not that much as above 5 features.



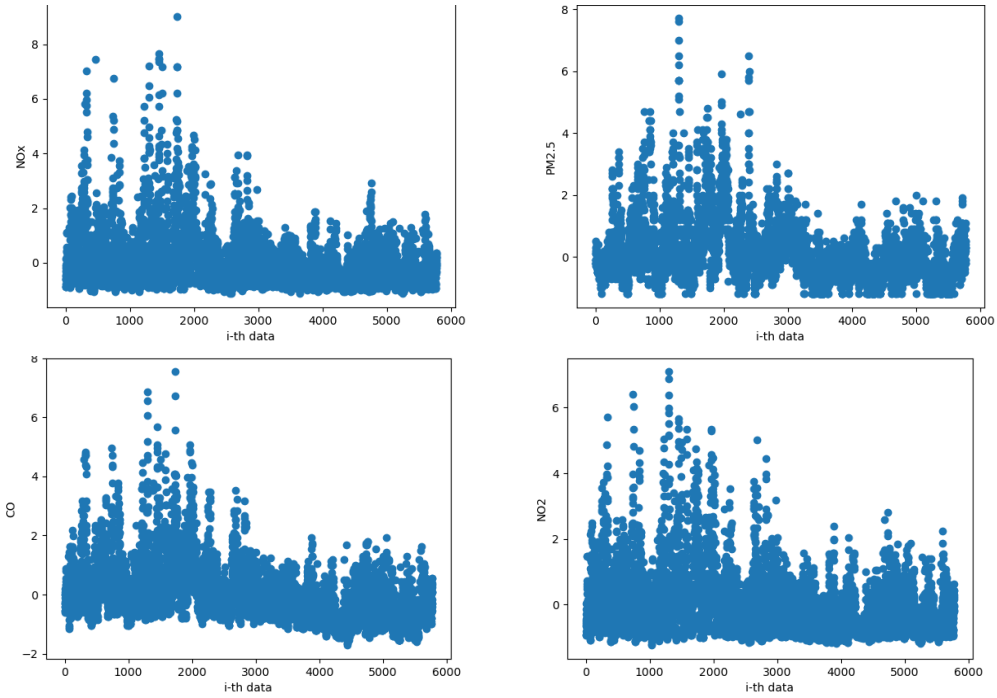
Problem: And in my experience, normalizing data can help to gather all data to a specific area that the model can converge much more rapidly. But using normalization is not like what I thought. In this case, the result is worse and also appear negative value of the PM2.5 result. According to this page, I thought maybe the normalization method is not suitable in my case.

Problem: I also figured that using the stored weight and bias by my pretrained model is not the right way. I used pickle to store the dump parameters during the training and used the best one as my pretrained parameter. But it's still not that good enough.

The better way in this project to enhance your accuracy is tuning your training config and select good features.

Solve: After discussing with my friend, I figured out the problem and tried to solve it successfully by fitting numpy random seed. Then, the parameter will truly fix but normalization is still not working to help model converging.

I used Zscore normalization to implement in my project and can see as below



You can see the different result of using or unusing normalization with the same config.

Regression	Learning Rate	Feats	Batch Size	Loss Function	Optimizer	RMSE(Lower is Better)	Filter Data	Norm. Data
1st_order	1.50E-02	[1-4, 6-9, 13, 14]	1024	MSE	Adam	2.44623	O	O
						2.13897	O	X
1st_order	1.50E-02	[1-4, 6-9, 13, 14]	512	MSE	Adam	2.49801	O	O
						2.13411	O	X

The experience with unnormalized data has lower RMSE.

2. (1%) 請實作 2nd-order polynomial regression model (不用考慮交互項)。

(a) 貼上 polynomial regression 版本的 Gradient descent code 內容

```
def minibatch(x, y, config):

    # Randomize the data in minibatch
    index = np.arange(x.shape[0])
    np.random.shuffle(index)
    x = x[index]
    y = y[index]

    # Initialization
    batch_size = config.batch_size
    lr = config.lr
    lam = config.lam
    epoch = config.epoch

    beta_1 = np.full(x[0].shape, 0.9).reshape(-1, 1)
```

```
beta_2 = np.full(x[0].shape, 0.99).reshape(-1, 1)
# Linear regression: only contains two parameters (w, b).
w = np.full(x[0].shape, 0.1).reshape(-1, 1)
w2 = np.full(x[0].shape, 0.1).reshape(-1, 1)    # Implement 2-nd polynomial regression
bias = 0.1
m_t = np.full(x[0].shape, 0).reshape(-1, 1)
v_t = np.full(x[0].shape, 0).reshape(-1, 1)
m_t_2 = np.full(x[0].shape, 0).reshape(-1, 1)    # Implement 2-nd polynomial regression
v_t_2 = np.full(x[0].shape, 0).reshape(-1, 1)    # Implement 2-nd polynomial regression
m_t_b = 0.0
v_t_b = 0.0
t = 0
epsilon = 1e-8

# Training loop
total_loss = np.zeros(epoch)
for num in range(epoch):
    for b in range(int(x.shape[0]/batch_size)):
        t+=1
        x_batch = x[b * batch_size:(b+1) * batch_size]
        y_batch = y[b * batch_size:(b+1) * batch_size].reshape(-1,1)

        # Implement 2-nd polynomial regression
        pred = np.dot(x_batch, w) + np.dot(x_batch**2, w2) + bias

        # loss(In this project, we use MSE Loss function.)
        loss = y_batch - pred   # This loss is just a variable, that actually loss function.

        # Compute w gradient
        g_t = np.dot(x_batch.transpose(), loss) * (-2)
        m_t = beta_1 * m_t + (1-beta_1) * g_t
        v_t = beta_2 * v_t + (1-beta_2) * np.multiply(g_t, g_t)
        m_cap = m_t / (1-(beta_1**t))
        v_cap = v_t / (1 - (beta_2**t))

        # Compute w2 gradient
        g_t_2 = np.dot((x_batch**2).transpose(), loss) * (-2)
        m_t_2 = beta_1 * m_t_2 + (1-beta_1) * g_t_2
        v_t_2 = beta_2 * v_t_2 + (1-beta_2) * np.multiply(g_t_2, g_t_2)
        m_cap_2 = m_t_2 / (1-(beta_1**t))
        v_cap_2 = v_t_2 / (1 - (beta_2**t))

        # Compute bias gradient
        g_t_b = loss.sum(axis=0) * (-2)
        m_t_b = 0.9 * m_t_b + (1 - 0.9) * g_t_b
        v_t_b = 0.99 * v_t_b + (1 - 0.99) * (g_t_b * g_t_b)
        m_cap_b = m_t_b / (1 - (0.9**t))
        v_cap_b = v_t_b / (1 - (0.99**t))

        w_0 = np.copy(w)

        # Update weight & bias
        w -= ((lr * m_cap) / (np.sqrt(v_cap) + epsilon)).reshape(-1, 1)
        w2 -= ((lr * m_cap_2) / (np.sqrt(v_cap_2) + epsilon)).reshape(-1, 1)
        bias -= (lr * m_cap_b) / (math.sqrt(v_cap_b) + epsilon)
```

return w**,** bias

(b) 在只使用 NO 數值作為 feature 的情況下，紀錄該 model 所訓練出的 parameter 數值（w2, w1, b）以及 kaggle public score.
Weight1:
[[0.46354605][0.22520665][0.11621199][0.21244262][0.24963454][0.19956132][0.00622001][0.67153817]]

Weight2:[[-0.01296799][-0.00587196][-0.00271931][-0.00365357][-0.00333787][-0.00294193][0.00257786][-0.01050794]]

Bias: [6.8461746]

Kaggle score: 5.79415 with config→2nd-order polynomial/epoch=200/lr=1.5e-2/batch_size=1024

3. (4%) Refer to math problem: https://hackmd.io/@lH2AB7kCSAS3NPw2FffsGg/Sk1n8xPWo?fbclid=IwAR0LiCps2fhIZFJT-gYP8kr7KlvLaRvS9-ftLIaPQY5DVgye1AuHM-RW3Yg

ML HW

Q1 : (a) Symmetric matrix $M \in \mathbb{R}^n$ is a positive semi-definite if $\forall X \in \mathbb{R}^n$ $X^T M X \geq 0$, given matrix $A \in \mathbb{R}^{n \times n}$. Show that AA^T is a positive semi-definite

$\Rightarrow$ Assume a matrix $V \in \mathbb{R}^n$

$V^T A A^T V \geq 0$

$= (A^T V)^T (A^T V) = \|A^T V\|^2$

Because of the power of the matrix, $\|A^T V\|$ is alway ≥ 0 $\forall V$

Then AA^T is positive semi-definite.

(b) If $f(x_1, x_2) = x_1 \sin(x_2) e^{-x_1 x_2}$, $\nabla f = \begin{bmatrix} \frac{\partial f}{\partial x_1} \\ \frac{\partial f}{\partial x_2} \end{bmatrix} = ?$

$\frac{\partial f}{\partial x_1} = \sin(x_2)\left[e^{-x_1 x_2} + x_1 \cdot e^{-x_1 x_2} \cdot (-x_2) \right]$

$= \boxed{e^{-x_1 x_2} \cdot \sin(x_2) \cdot (1 - x_1 x_2)}$ ※

$\frac{\partial f}{\partial x_2} = x_1 \left[\cos(x_2) e^{-x_1 x_2} + \sin(x_2) e^{-x_1 x_2} \cdot (-x_1) \right]$

$= \boxed{x_1 e^{-x_1 x_2} \left[\cos(x_2) - x_1 \sin(x_2) \right]}$ ※

(c) Given $f(x; p) = p^x (1-p)^{1-x}$ for $x \in \{0, 1\}$

The MLE of p is:

(1) construct $L(p) = f(x_1; p) f(x_2; p) \cdots f(x_n; p) = p^{x_1}(1-p)^{1-x_1} p^{x_2}(1-p)^{1-x_2} \cdots p^{x_n}(1-p)^{1-x_n}$

$= p^{\sum_{i=1}^n x_i} (1-p)^{n - \sum_{i=1}^n x_i}$

(2) $\ln(L(p)) = \ln(p^{\sum_{i=1}^n x_i}) + \ln((1-p)^{n - \sum_{i=1}^n x_i}) = \sum_{i=1}^n x_i \ln(p) + (n - \sum_{i=1}^n x_i) \ln(1-p)$

given $\sum_{i=1}^n x_i = n\bar{x}$ where $\bar{x}$ is average of all sample

(3) $\frac{d}{dp} \ln(L(p)) = n\bar{x} \cdot \frac{1}{p} + (n - n\bar{x}) \cdot \frac{-1}{1-p}$, suppose $\frac{d}{dp} \ln(L(p)) = 0$

$\Rightarrow \frac{n\bar{x}}{p} = \frac{n - n\bar{x}}{1-p} \Rightarrow \boxed{p = \bar{x}}$ ※

make this correct

examine the 2nd derivative

(4) $\frac{d^2}{dp^2} \ln(L(p)) = \frac{-n\bar{x}}{p^2} + \frac{-(n - n\bar{x})}{(1-p)^2}$, substitude $p = \bar{x} \Rightarrow \frac{-n}{\bar{x}} - \frac{n(1-\bar{x})}{(1-\bar{x})^2} = \frac{-n}{\bar{x}} - \frac{n}{1-\bar{x}}$

because of $x \in \{0, 1\}$, $0 \leq \bar{x} \leq 1$ and $1 - \bar{x} \geq 0 \Rightarrow \frac{d^2}{dp^2} \ln(L(p)) < 0$

ML HW

Q2, Linear Regression model $\vec{y} = \vec{X}\vec{\theta} + \epsilon$ where $\vec{y} \in \mathbb{R}^n$, $\vec{X} \in \mathbb{R}^{n \times d}$, $\vec{\theta} \in \mathbb{R}^d$, $\epsilon \in \mathbb{R}^n$

(a) Find general form of θ^* that minimize the weighted MSE

Given: $L(\theta) = (y - X\theta)^T \Omega (y - X\theta)$

$= (y^T - \theta^T X^T)\Omega(y - X\theta) = (y^T\Omega - \theta^T X^T \Omega)(y - X\theta)$

$= y^T\Omega y - \underbrace{\theta^T X^T \Omega y}_{③} - \underbrace{y^T \Omega X \theta}_{②} + \underbrace{\theta^T X^T \Omega X \theta}_{①}$ → 2nd order term

$= (\theta - \phi)^T X^T \Omega X (\theta - \phi) +$ complementary term ↖ shorted as CT

$= (\theta^T - \phi^T)(X^T \Omega X)(\theta - \phi) +$ CT

$= (\theta^T X^T \Omega X - \phi^T X^T \Omega X)(\theta - \phi) + CT$

$= \underbrace{\theta^T X^T \Omega X \theta}_{△} - \underbrace{\phi^T X^T \Omega X \theta}_{△} - \underbrace{\theta^T X^T \Omega X \phi}_{△} + \phi^T X^T \Omega X \phi + CT$

the same as the last term ← $△$

△ = ①
△ = ②
△ = ③

$\theta^T X^T \Omega y = \theta^T X^T \Omega X \phi \Rightarrow \phi = (X^T \Omega X)^{-1} X^T \Omega y$

put into △ = ② $\Rightarrow \theta^T X^T \Omega X \theta = \theta^T X^T \Omega X (X^T \Omega X)^{-1} X^T \Omega y = \theta^T X^T \Omega y$

$CT = y^T \Omega y - \phi^T X^T \Omega X \phi = y^T \Omega y - y^T \Omega^T X \left[(X^T \Omega X)^{-1}\right]^T X^T \Omega X (X^T \Omega X)^{-1} X^T \Omega y$ (the same with ②)

$= y^T \Omega y - y^T \Omega^T X \left[(X^T \Omega X)^{-1}\right]^T X^T \Omega y$

$$\boxed{L(\theta) = (\theta - \phi)^T X^T \Omega X (\theta - \phi) + y^T \Omega y - y^T \Omega^T X \left[(X^T \Omega X)^{-1}\right]^T X^T \Omega y \quad \text{where } \phi = (X^T \Omega X)^{-1} X^T \Omega y} \ *$$

(b) $L(\theta) = (y_i - X_i\theta)^T (y_i - X_i\theta) + \lambda w w^T \quad \because w^T = \theta$

$\therefore = (y_i - X_i\theta)^T (y_i - X_i\theta) + \lambda \theta^T \theta$

$\frac{\partial L(\theta)}{\partial \theta} \overset{①}{=} \frac{\partial (y_i - X_i\theta)}{\partial \theta}(y_i - X_i\theta) + \frac{\partial (y_i - X_i\theta)}{\partial \theta}(y_i - X_i\theta) + \frac{\partial \lambda \theta^T \theta}{\partial \theta}$

$= 2\left[\overset{②}{-X_i^T}(y_i - X_i\theta)\right] + \overset{③}{2\lambda\theta} \Rightarrow$ when $\frac{\partial L(\theta)}{\partial \theta} = 0$, it has extreme value

$\Rightarrow -2X_i^T y_i + 2X_i^T X_i \theta + 2\lambda\theta = 0 \Rightarrow X_i^T X_i \theta + \lambda\theta = X_i^T y_i = (X_i^T X_i + \lambda I)\theta$

$$\boxed{\theta = (X_i^T X_i + \lambda I)^{-1} X_i^T y_i} \ *$$

Cont. for Q2.b

① $\frac{\partial \vec{u}^T \vec{v}}{\partial \vec{x}} = \frac{\partial \vec{u}}{\partial \vec{x}} \vec{v} + \frac{\partial \vec{v}}{\partial \vec{x}} \vec{u} = \frac{\partial \sum_k u_k v_k}{\partial \vec{x}} = \sum_k \frac{\partial u_k v_k}{\partial \vec{x}} = \sum_k \left(u_k \frac{\partial v_k}{\partial \vec{x}} + v_k \frac{\partial u_k}{\partial \vec{x}} \right)$

$= \sum_k \frac{\partial v_k}{\partial \vec{x}} u_k + \sum_k \frac{\partial u_k}{\partial \vec{x}} v_k = \frac{\partial \vec{v}}{\partial \vec{x}} \vec{u} + \frac{\partial \vec{u}}{\partial \vec{x}} \vec{v}$

② $\frac{\partial A\vec{u}}{\partial \vec{x}} = \frac{\partial \vec{u}}{\partial \vec{x}} A^T$

$\left(\frac{\partial A\vec{u}}{\partial \vec{x}} \right)_{ij} = \frac{\partial \sum_k a_{jk} u_k}{\partial x_i} = \sum_k a_{jk} \frac{\partial u_k}{\partial x_i} = \sum_k \left(\frac{\partial \vec{u}}{\partial \vec{x}} \right)_{ik} (A^T)_{kj} = \left(\frac{\partial \vec{u}}{\partial \vec{x}} A^T \right)_{ij}$

③ $\frac{\partial \vec{x}^T \vec{x}}{\partial \vec{x}} = 2X$, Use $\frac{\partial X^T A X}{\partial X} = (A + A^T) X$ and replace A to I.

$\frac{\partial \vec{x}^T I \vec{x}}{\partial X} = \frac{\partial X^T I X}{\partial X} = \frac{\partial X}{\partial X} IX + \frac{\partial X}{\partial X} I^T X = IIX + II^T X$

$= X + X = 2X$

What is the extreme value that $\frac{\partial L(\theta)}{\partial \theta}$ is

$\frac{\partial^2}{\partial \theta^2} \left(-2X_i^T y_i + 2X_i^T X_i \theta + 2\lambda\theta \right) = 2\left(X_i^T X\right)^T + 2\lambda = 2\left(\|X\|^2\right)^T + 2\lambda > 0$

curved open up so $\frac{\partial L(\theta)}{\partial \theta} = 0$ has minimum value. ✱

ML HW

Q.3 : Logistic Sigmoid Function and Hyperbolic Tangent Function

Given: $\sigma(a) = \frac{1}{1+e^{-a}}$, $\tanh(a) = \frac{e^{a}-e^{-a}}{e^{a}+e^{-a}}$

(1) Show $\tanh(a) = 2\sigma(2a) - 1$

$$\Rightarrow \tanh(a) = \frac{e^{a}-e^{-a}}{e^{a}+e^{-a}} = \frac{1-e^{-2a}}{1+e^{-2a}} = \frac{1}{1+e^{-2a}} - \frac{e^{-2a}}{1+e^{-2a}}$$

$$= \frac{1}{1+e^{-2a}} - \frac{1}{e^{2a}+1} = \sigma(2a) - \sigma(-2a)$$

$$\because \sigma(z) = 1 - \sigma(-z) \quad \therefore \tanh(a) = \sigma(2a) - 1 + \sigma(2a)$$

$$= 2\sigma(2a) - 1 \quad \text{※}$$

(2) Given $\begin{cases} y(x, \vec{w}) = w_0 + \sum_{j=1}^{M} w_j \sigma\left(\frac{x-\mu_j}{s}\right) \\ y(x, \vec{u}) = u_0 + \sum_{j=1}^{M} u_j \tanh\left(\frac{x-\mu_j}{2s}\right) \end{cases}$

Assume $a = \frac{x-\mu_j}{2s}$

$$\Rightarrow w_j \sigma(a) = u_j \tanh(a) = u_j\left[2\sigma(2a) - 1\right] = u_j\left[2\frac{1}{1+e^{-2a}} - 1\right]$$

$$= w_j \frac{1}{1+e^{-a}} \Rightarrow \frac{w_j}{u_j} = 2\frac{(1+e^{-a})}{1+e^{-2a}} - (1+e^{-a}) = \frac{2(1+e^{-a}) - (1+e^{-a})(1+e^{-2a})}{1+e^{-2a}}$$

$$= \frac{2 + 2e^{-a} - 1 - e^{-a} - e^{-2a} - e^{-3a}}{1+e^{-2a}} = \frac{1+e^{-a} - e^{-2a} - e^{-3a}}{1+e^{-2a}}$$

$$= \frac{1-e^{-2a}}{1+e^{-2a}} + \frac{e^{-a} - e^{-3a}}{1+e^{-2a}} = \frac{e^{a}-e^{-a}}{e^{a}+e^{-a}} + e^{-a}\left(\frac{1-e^{-2a}}{1+e^{-2a}}\right)$$

$$= \tanh(a) + e^{-a}\left(\frac{e^{a}-e^{-a}}{e^{a}+e^{-a}}\right) = \tanh(a) + e^{-a}\tanh(a) = \boxed{(1+e^{-a})\tanh(a)}$$

Next Page for Q3.2

$$w_j \sigma(2a) = u_j \tanh(a) = u_j\left[2\sigma(2a) - 1\right] \Rightarrow \frac{w_j}{u_j} = \frac{2\sigma(2a)-1}{\sigma(2a)} = 2 - \frac{1}{\sigma(2a)}$$

$$= \boxed{2 - \frac{1}{\sigma\left(\frac{x-\mu_j}{s}\right)}} \quad \text{※}$$

Q3.2

Given $\begin{cases} y(x,\vec{w}) = w_0 + \sum_{j=1}^{M} w_j \sigma\left(\frac{x-\mu_j}{s}\right) \\ y(x,\vec{u}) = u_0 + \sum_{j=1}^{M} u_j \tanh\left(\frac{x-\mu_j}{2s}\right) \end{cases}$

Show these 2 expression are the same.

$$w_0 + \sum_{j=1}^{M} w_j \sigma\left(\frac{x-\mu_j}{s}\right) = u_0 + \sum_{j=1}^{M} u_j \tanh\left(\frac{x-\mu_j}{2s}\right) \quad \text{Assume } a = \frac{x-\mu_j}{2s}$$

$$\Rightarrow w_0 + \sum_{j=1}^{M} w_j \sigma(2a) = u_0 + \sum_{j=1}^{M} u_j \tanh(a) = u_0 + \sum_{j=1}^{M} u_j (2\sigma(2a)-1)$$

$$\Rightarrow w_0 + \frac{w_1}{1+\exp\left(\frac{-x+\mu_1}{s}\right)} + \frac{w_2}{1+\exp\left(\frac{-x+\mu_2}{s}\right)} + \cdots\cdots + \frac{w_M}{1+\exp\left(\frac{-x+\mu_M}{s}\right)}$$

$$= u_0 + u_1 \frac{2-\left(1+\exp\left(\frac{-x+\mu_1}{s}\right)\right)}{1+\exp\left(\frac{-x+\mu_1}{s}\right)} + \frac{u_2\left(2-\left(1+\exp\left(\frac{-x+\mu_2}{s}\right)\right)\right)}{1+\exp\left(\frac{-x+\mu_2}{s}\right)} + \cdots + \frac{w_M}{1+\exp\left(\frac{-x+\mu_M}{s}\right)}$$

$$\frac{[w_1, w_2, w_3 \ldots w_M]}{[u_1, u_2, u_3 \ldots u_M]} = 1-\exp\left(\frac{-x+\mu_j}{s}\right), \quad j = \{1, 2, \ldots M\}$$

※

Refer to 黃浩然

$$w_0 + \sum_{j=1}^{M} w_j \sigma(2a) = u_0 + \sum_{j=1}^{M} u_j (2\sigma(2a)-1) = u_0 + \sum_{j=1}^{M} 2u_j \sigma(2a) - \sum_{j=1}^{M} u_j$$

$$= u_0 - \sum_{j=1}^{M} u_j + \sum_{j=1}^{M} 2u_j \sigma(2a)$$

Because $w_0 \neq u_0$, cannot use the method above (red cross area)

$$w_0 = u_0 - \sum_{j=1}^{M} u_j$$

$$\sum_{j=1}^{M} w_j \sigma(2a) = \sum_{j=1}^{M} 2u_j \sigma(2a) \Rightarrow \boxed{\frac{u_j}{w_j} = \frac{1}{2}} \quad \text{for } j \in \{1, 2 \ldots M\}$$

MLHW

Q4. Given $f_{w,b}(x) = w^T x + b$ where $w \in \mathbb{R}^k$, $b \in \mathbb{R}$, $x_i \in \mathbb{R}^k$, $\eta_i \in \mathbb{R}^k$

$$\tilde{L}_{ss}(w,b) = \mathbb{E}\left[\frac{1}{2N}\sum_{i=1}^{N}\left(f_{w,b}(x_i+\eta_i) - y_i\right)^2\right]$$

$$= \frac{1}{2N}\sum_{i=1}^{N}\left[\mathbb{E}\left[\left(w^T(x_i+\eta_i)+b-y_i\right)^2\right]\right]$$

$$= \frac{1}{2N}\sum_{i=1}^{N}\left[\mathbb{E}\left[\left(w^Tx_i + w^T\eta_i + b - y_i\right)^2\right]\right]$$

$$= \frac{1}{2N}\sum_{i=1}^{N}\left[\mathbb{E}\left[\left(f_{w,b}(x_i) - y_i + w^T\eta_i\right)^2\right]\right]$$

$$= \frac{1}{2N}\sum_{i=1}^{N}\left[\mathbb{E}\left[\underbrace{\left(f_{w,b}(x_i)-y_i\right)^2}_{a^2} + \underbrace{2}_{2}\underbrace{\left(f_{w,b}(x_i)-y_i\right)}_{a}\underbrace{\left(w^T\eta_i\right)}_{b} + \underbrace{\left(w^T\eta_i\right)^2}_{b^2}\right]\right]$$

(Independent)

$$= \frac{1}{2N}\sum_{i=1}^{N}\left[\mathbb{E}\left[\left(f_{w,b}(x_i)-y_i\right)^2\right] + 2\mathbb{E}\left[f_{w,b}(x_i)-y_i\right]\mathbb{E}\left[w^T\eta_i\right] + \mathbb{E}\left[\left(w^T\eta_i\right)^2\right]\right]$$

($\left(f_{w,b}(x_i)-y_i\right)^2$: constant to $\mathbb{E}$)

$$= \frac{1}{2N}\sum_{i=1}^{N}\left[\left(f_{w,b}(x_i)-y_i\right)^2 + 2\left(f_{w,b}(x_i)-y_i\right)w^T\underbrace{\mathbb{E}\left[\eta_i\right]} + \left(w^T\right)^2\mathbb{E}\left[\eta_i^2\right]\right]$$

($\mathbb{E}[\eta_i] = 0$ where $\mathbb{E}[\eta_{i,j}] = 0$)

$$= \frac{1}{2N}\sum_{i=1}^{N}\left(f_{w,b}(x_i)-y_i\right)^2 + \frac{\left(w^T\right)^2}{2N}\sum_{i=1}^{N}\mathbb{E}\left[(\eta_i)^2\right]$$

$$= \frac{1}{2N}\sum_{i=1}^{N}\left(f_{w,b}(x_i)-y_i\right)^2 + \frac{\|w\|^2}{2N}\sum_{i=1}^{N}\delta_{ii'}\delta_{ii'}\sigma^2 \quad \text{when } \delta_{i,i'} = \begin{cases}1, & \text{if } i=i'\\ 0, & \text{otherwise}\end{cases}$$

$$= \frac{1}{2N}\sum_{i=1}^{N}\left(f_{w,b}(x_i)-y_i\right)^2 + \frac{\|w\|^2\cdot N\sigma^2}{2N} =$$

$$= \boxed{\frac{1}{2N}\sum_{i=1}^{N}\left(f_{w,b}(x_i)-y_i\right)^2 + \frac{\|w\|^2\sigma^2}{2}} \ ※$$

Subject : ML HW

No. :

Date : ...../...../.....

Q5: Given linear function of the feature vector $\vec{x} = [x_1, x_2, x_3 \dots x_n]^T \in \mathbb{R}^n$ and $\vec{w} = [w_1, w_2, w_3 \dots w_n]^T \in \mathbb{R}^n$, so that

$$f_{\vec{w},b}(\vec{x}) = p(C_1|\vec{x}) = \sigma\left(\sum_i w_i x_i + b\right) = \sigma(\vec{w}^T\vec{x} + b)$$

with $p(C_2|\vec{x}) = 1 - p(C_1|\vec{x}) = 1 - f_{\vec{w},b}(\vec{x})$

(a) Suppose $\vec{w} = [-1, 2, -1, 5]^T$, $\vec{x} = [7, 0, 3, 10]^T$ and $b = 3$

$\Rightarrow f_{\vec{w},b}(\vec{x}) = \sigma(\vec{w}^T\vec{x} + b) = \sigma([-1, 2, -1, 5][7, 0, 3, 10]^T + 3) = \sigma(-7 + 0 - 3 + 50 + 3)$

$= \sigma(43) = \frac{1}{1 + e^{-43}} = 1$ ※

(b) Given training dataset $\{x_i, y_i\}_{i=1}^{N}$ ($N \rightarrow$ quantity of dataset), where $y_i \in \{0, 1\}$

$\Rightarrow$ likelihood function : $L(\theta) = f(x_1;\theta) f(x_2;\theta) f(x_3;\theta) \cdots f(x_n;\theta) = \prod_{i=1}^{n} f(x_i;\theta)$

$\Rightarrow L(w,b) = f_{w,b}(x^1) f_{w,b}(x^2) \dots f_{w,b}(x^N)$

$\Rightarrow$ Let $L(w,b)$ likelihood function maximize means minimize $-\ln L(w,b)$

$\Rightarrow -\ln L(w,b) = -\ln f_{w,b}(x^1) \cdot -\ln f_{w,b}(x^2) \cdot -\ln f_{w,b}(x^3) \cdots -\ln f_{w,b}(x^N)$

$= -[y_1 \ln f_{w,b}(x_1) + (1 - y_1)\ln(1 - f_{w,b}(x_1)] \cdot -[y_2 \ln f_{w,b}(x_2) + (1 - y_2)\ln(1 - f_{w,b}(x_2)] \cdots$

$= \prod_{i=1}^{N} -[y_i \ln f_{w,b}(x_i) + (1 - y_i)\ln(1 - f_{w,b}(x_i)]$ ※

(c) Observe $(x^i, \hat{y}^i)_{i=1}^{N}$ where $\hat{y}^i \in \{0, 1\}$, $x^i \in \mathbb{R}^d$

Given: $P_\theta(C_1|X) = \sigma(w^T x + b)$ is equivalent to minimize $\sum_{i=1}^{N} -[y_i \ln(\sigma(w^T x + b) + (1 - y_i)\ln(1 - \sigma(w^T x + b))]$

$\Rightarrow$ Apply Gradient Descent $\frac{\partial L}{\partial w_i}(w,b)$ 、 $\frac{\partial L}{\partial b_i}(w,b)$

$\Rightarrow \frac{\partial L}{\partial w_i}(w,b) = -\sum_{i=1}^{N}\left[y^i \frac{\partial}{\partial w_i}\ln(\sigma(w^T x + b)) + (1 - y^i)\frac{\partial}{\partial w_i}\ln(1 - \sigma(w^T x + b))\right]$

$\frac{\partial}{\partial z}\sigma(z) = \sigma(z)\sigma(-z) = \sigma(z)(1 - \sigma(z))$

$= -\sum_{n=1}^{N}\left[y^i \cdot \frac{\sigma(w^T x^n + b)(1 - \sigma(w^T x^n + b))}{\sigma(w^T x^n + b)} \cdot x_i^n + (1 - y^i)\frac{(-1)\sigma(w^T x^n + b)(1 - \sigma(w^T x^n + b))}{1 - \sigma(w^T x^n + b)} \cdot x_i^n\right]$

($w_i x_i + w_2 x_2 \cdots w_i x_i + \cdots w_N x_N$)

$= -\sum_{n=1}^{N} x_i^n\left[y_i(1 - \sigma(w^T x^n + b)) + (-1 + y_i)\sigma(w^T x^n + b)\right]$

$= -\sum_{n=1}^{N} x_i^n\left[y_i - y_i\sigma(w^T x^n + b) + -\sigma(w^T x + b) + y_i\sigma(w^T x^n + b)\right]$

$= -\sum_{n=1}^{N} x_i^n\left[y_i - \sigma(w^T x^n + b)\right] = -\sum_{n=1}^{N} x_i^n\left[y_i - f_{w,b}(x_i^n)\right]$

$\Rightarrow w_i^{(t+1)} = w_i^{(t)} - \eta \sum_n -(y^i - f_{w,b}(x_i^n))\, x_i^n$ ※