學號：R10921019 系級： 電機所碩二 姓名：鄭皓方

1. (1%) 解釋什麼樣的**data preprocessing**可以**improve**你的**training/testing accuracy**?

我總共用了三種方法來處理資料，第一種是選擇 feature，我計算input的各種特徵和output的相關係數，選擇相關係數最高的前六個當作我的feature；第二種方式是設定output 的 threshold，因為linear regresion 容易受到極端資料的影響，最後我將大於20的output全部拿掉；第三種方法是我將trainning data 的 input做 normalize，如下方的公式。

$$\frac{train\ x - train\ x.min}{train\ x.max - train\ x.min}$$

然後testing data一樣套用training data的值做normalize。這個方法在kaggle的結果沒有比較好，然而我在自己local端的測試有好一點點，於是我決定最後繳交的兩個檔案一個是有normalize一個沒有，看看最後的結果。

特徵名稱	相關係數
PM2.5	1
CO	0.659
NO2	0.554
NOx	0.513
PM10	0.818
SO2	0.361

階數	threshold	有無normalize	kaggle public score
1	80	無	4.11
1	50	無	2.37
1	35	無	2.36
1	24	無	2.23
1	15	無	2.29
1	24	有	2.30
2	24	有	2.38

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

```
# Prediction of linear regression
x_square = np.power(x_batch, 2)
pred = np.dot(x_square, w_1)+np.dot(x_batch, w_0) + bias
# loss
loss = y_batch - pred
# Compute gradient
g_t = np.dot(x_batch.transpose(), loss) * (-2) + 2 * lam * np.sum(w_0)
g_t_2 = np.dot(x_square.transpose(), loss) * (-2) + 2 * lam * np.sum(w_1)
g_t_b = loss.sum(axis=0) * (-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))
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))
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))

# Update weight & bias
w_0 -= ((lr*m_cap)/(np.sqrt(v_cap)+epsilon)).reshape(-1, 1)
w_1 -= ((lr*m_cap_2)/(np.sqrt(v_cap_2)+epsilon)).reshape(-1, 1)
bias -= (lr*m_cap_b)/(math.sqrt(v_cap_b)+epsilon)
```

(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.

	1	2	3	4	5	6	7	8
w1	0.104	0.100	0.095	0.187	0.070	0.054	-0.020	0.214
w2	-0.026	-0.023	-0.016	0.002	-0.013	-0.056	-0.043	-0.011
bias	7.489	7.489	7.489	7.489	7.489	7.489	7.489	7.489

kaggle public score: 4.15023

3.(4%) Refer to math problem:

Machine learning HW1

R10921019
鄭皓方

1、Mathematic Background

(a.) let $M = AA^T$

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

s.t $M = M^T$, M is a symmetric matrix

therefore, AA^T is a semidefinite matrix by the given definition.

(b)、 $f(x_1, x_2) = x_1 \sin(x_2) \exp(-x_1 x_2)$

$$\nabla f(x) = \begin{bmatrix} \frac{\partial f}{\partial x_1} \\ \frac{\partial f}{\partial x_2} \end{bmatrix} = \begin{bmatrix} \sin(x_2)\exp(-x_1x_2) - x_1x_2\sin(x_2)\exp(-x_1x_2) \\ x_1\cos(x_2)\exp(-x_1x_2) - x_1^2\sin(x_2)\exp(-x_1x_2) \end{bmatrix}$$

(c)、 $L(p,x) = f_p(x_1) f_p(x_2) \cdots f_p(x_n)$

$$L(p,x) = p^{\sum_1^n x}(1-p)^{n-\sum_1^n x}$$

$$\log L(p,x) = \sum_1^n x \log p + (n - \sum_1^n x)\log(1-p)$$

because $p, (1-p) > 0$

let $\sum_1^n x = Y$

$$\frac{\partial \log L(p,x)}{\partial p} = \frac{Y}{p} - \frac{n-Y}{1-p} = \frac{Y - pY - np + pY}{p(1-p)}$$

$\Leftarrow$ maximize at same value as $L(p,x)$

equilibrium point is at $p = \frac{Y}{n} = \frac{\sum_1^n x_i}{n}$ #

2.

(a) $L(\theta) = (y_i - X_i\theta)^T \Omega (y_i - X_i\theta)$

$= y^T\Omega y - \theta^T X^T \Omega y - y^T \Omega X\theta + \theta^T X^T \Omega X \theta$

$\nabla_\theta L(\theta) = 2X^T\Omega X\theta - 2X^T\Omega y$

$\theta^* = (X^T\Omega X)^{-1} X^T \Omega y$ ※

(b) $L(\theta) = \|y - X\theta\|_2^2 + \lambda\|w\|_2^2$ no bias term $\rightarrow w = \theta$

$L(\theta) = (y - X\theta)^T(y - X\theta) + \lambda I \theta\theta^T$

$\nabla_\theta L(\theta) = 2X^TX\theta - 2X^Ty + 2\lambda I\theta$

$= 2(X^TX + \lambda I)\theta - 2X^Ty$

$\theta^* = (X^TX + \lambda I)^{-1} X^Ty$

(c) let $X = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1m} & \cdots & 1 \\ x_{21} & x_{22} & \cdots & x_{2m} & \cdots & 1 \\ & & \vdots & & & \\ x_{n1} & x_{n2} & \cdots & x_{nm} & & 1 \end{bmatrix}$ $\theta = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_d \\ b \end{bmatrix}$ $I' = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & & \vdots \\ \vdots & & \ddots & 1 \\ 0 & \cdots & & 0 \end{bmatrix}$

$L(\theta) = (y - X\theta)^T(y - X\theta) + \lambda I'\theta\theta^T$

$\nabla_\theta L(\theta) = 2X^TX\theta - 2X^Ty + 2\lambda I'\theta$

$\theta^* = (X^TX + \lambda I')^{-1} X^Ty$ if $(X^TX + \lambda I')$ is invertible

3. (a)

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

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

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

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

(b)

$$y(x,w) = w_0 + \sum_{j=1}^{M} w_j \sigma\left(\frac{x-\mu_j}{s}\right)$$

$$= w_0 + \sum_{j=1}^{M} \frac{w_j}{2}(2\sigma)\left(2\left(\frac{x-\mu_j}{2s}\right)\right)$$

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

$$= w_0 + \sum_{j=1}^{M} \frac{w_j}{2}\left[\tanh\left(\frac{x-\mu_j}{2s}\right) + 1\right]$$

$$= w_0 + \sum_{j=1}^{M} \frac{w_j}{2} + \sum_{j=1}^{M} \frac{w_j}{2}\left[\tanh\left(\frac{x-\mu_j}{2s}\right)\right]$$

$$[u_1, u_2, \dots, u_M] = \left[\frac{w_1}{2}, \frac{w_2}{2}, \dots, \frac{w_M}{2}\right] \#$$

4、

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

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

$$= E\left[\frac{1}{2N} \sum_{1}^{N} \left(f_{w,b}(x_i) - y_i\right)^2 + \underbrace{\cancel{2w\, x_i^T w^T \eta_i}}_{\to 0} \underbrace{\cancel{- 2w\, \eta_i^T y_i}}_{\to 0} + w\, \eta_i^T w^T \eta_i \right]$$

Two terms equal to zero because $E[\eta_{i,j}] = 0$

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

$$= \frac{1}{2N} \sum_{1}^{N} \left[\left(f_{w,b}(x_i) - y_i\right)^2 + \|w\|^2 \delta_{ii'} \delta_{jj'} \sigma^2 \right]$$

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

5、

(a)

$$f_{w,b}(x) = \sigma(w^T x + b)$$

$$= \sigma\left([-1, 2, -1, 5] \begin{bmatrix} 7 \\ 0 \\ 3 \\ 10 \end{bmatrix} + 3 \right) = \sigma(43)$$

$$= 1$$

(b) $L(w,b) = \Pi_i \; y_i f_{w,b}(X) + (1-y_i)(1-f_{w,b}(X))$

$$-\ln L(w,b) = \sum_{i=1}^{n} -\left[y_i \ln f_{w,b}(X) + (1-y_i)\ln(1-f_{w,b}(X)) \right]$$

(c) $\dfrac{-\ln L(w,b)}{\partial w_i} = \sum -\left[y_i \dfrac{\partial \ln f_{w,b}(X)}{\partial w_i} + (1-y_i)\dfrac{\partial \ln(1-f_{w,b}(X))}{\partial w_i} \right]$

$$\frac{\partial \ln f_{w,b}(X)}{\partial w_i} = \frac{\partial \ln f_{w,b}(X)}{\partial z}\frac{\partial z}{\partial w_i} = \frac{\partial \ln f_{w,b}(X)}{\partial z} x_i \quad \text{where } z = w_i x_i + b$$

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

$$\therefore \frac{\partial \ln f_{w,b}(X)}{\partial w_i} = [1-\sigma(z)]x_i = (1-f_{w,b}(X))x_i$$

then rewrite the second part as $\dfrac{\partial \ln(1-\sigma(z))}{\partial z}\dfrac{\partial z}{\partial w}$

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

$$\frac{-\ln L(w,b)}{\partial w_i} = \sum -\left[y_i(1-f_{w,b}(X))x_i - (1-y_i)f_{w,b}(X)\,x_i \right]$$

$$= \sum -\left[y_i - \cancel{y_i f_{w,b}(X)} - f_{w,b}(X) + \cancel{y_i f_{w,b}(X)} \right] x_i$$

$$= \sum -\left[y_i - f_{w,b}(X) \right] x_i$$

$$w^{(t+1)} = w^{(t)} - \eta \sum_i -(y_i - f_{w,b}(X))\, x_i$$ ※