1.

by 投影片 P38. $d_{VC} \le d+1$

故 3D perceptron 的 $d_{VC} \le 3+1 = 4$

[a]

$$\begin{matrix} & x_1 & x_2 & x_3 \end{matrix}$$

$$\begin{pmatrix} 1 & 1 & 1 \\ 2 & 4 & 3 \\ 3 & 3 & 3 \\ 4 & 2 & 3 \end{pmatrix}$$

$$\to \begin{pmatrix} 1 & 1 & 1 \\ 0 & 2 & 1 \\ 0 & 0 & 0 \\ 0 & -2 & -1 \end{pmatrix} \to \begin{pmatrix} 1 & 1 & 1 \\ 0 & 2 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

$\to$ L.D

[b]

$$\begin{matrix} & x_1 & x_2 & x_3 & x_4 \end{matrix}$$

$$\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 2 & 4 & 4 \\ 1 & 3 & 3 & 2 \\ 1 & 4 & 2 & 3 \end{pmatrix} \to \begin{pmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 3 & 3 \\ 0 & 2 & 2 & 1 \\ 0 & 3 & 1 & 2 \end{pmatrix}$$

$$\to \begin{pmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 3 & 3 \\ 0 & 0 & -4 & -5 \\ 0 & 0 & -8 & -7 \end{pmatrix} \to \begin{pmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 3 & 3 \\ 0 & 0 & -4 & -5 \\ 0 & 0 & 0 & 3 \end{pmatrix}$$

$\to$ L.I

(C) $X_1\ X_2\ X_3\ X_4$

$$\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 2 & 4 & 2 \\ 1 & 3 & 3 & 2 \\ 1 & 4 & 2 & 2 \end{pmatrix} \to \begin{pmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 3 & 1 \\ 0 & 2 & 2 & 1 \\ 0 & 3 & 1 & 1 \end{pmatrix}$$

$$\to \begin{pmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 3 & 1 \\ 0 & 0 & -4 & -1 \\ 0 & 0 & -8 & -2 \end{pmatrix} \to \begin{pmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 3 & 1 \\ 0 & 0 & -4 & -1 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

$\to L \cdot D$

(D) $d_{vc} \le d+1$, $5 > 4$ 故排除.

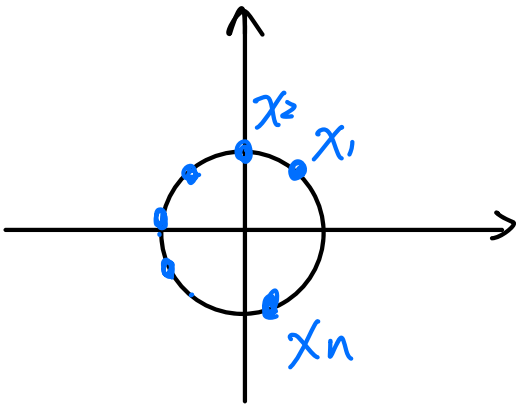
(a)(c) 因綫性相依使得某矣 x_n 只能由其它 $n-1$ 個 x 決定，故無法產生 2^n 個 dichotomy.

(a) $w^T x_3 = w^T x_1 - w^T x_2 > 0$ ○ ✗ $d_{vc} = 2$

(c) $w^T x_4 = w^T x_1 + w^T x_2 - w^T x_3 > 0$ ○ ○ ✗ $d_{vc} = 3$

(d) L.I. $w^T x_4 \neq w^T x_1 + w^T x_2 + w^T x_3$, $d_{vc} = 4$

2. 由於 $W_0 = 0$，故 Data 可轉換至極座標表達，且長度不影響 data 關係，故可以單位圓表示，如下圖



由於 data 只由"角度"來分類，故可將問題等價成

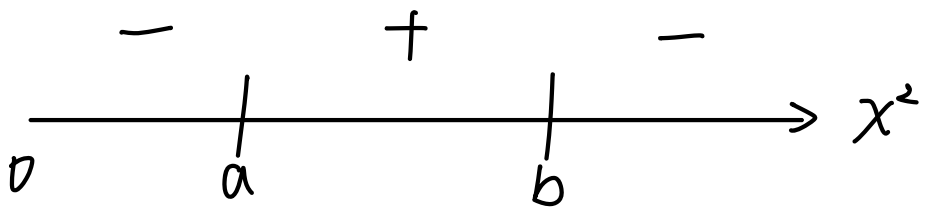
positive negative ray.

故 $m_H(N) = 2N$.

$\Rightarrow$ ans : [C]

3. data 為圓心的距離 x_i^2，決定 "+"、"−"

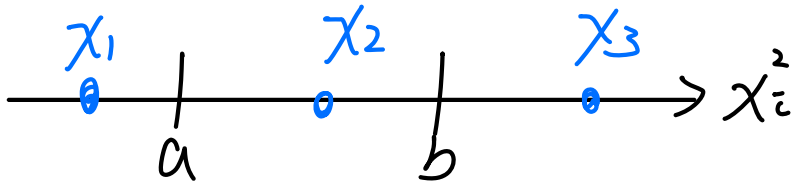
因此問題可等價成一維 positive interval，如下圖



故 $m_H(N) = \binom{N+1}{2} + 1$

⇒ ans: [a]

4. 由於 positive interval 的 $dvc(m_H) = 2$ (by 第3題)



若 $\{x_1\ x_2\ x_3\} = \{+, -, +\}$ 則無法 shatter

故 $dvc(m_H) = 2$.

$\Rightarrow$ ans : [d]

5. [a] degree of freedom = 4, dvc = 4

1. 單一 positive intervals 決定了 "o" 區間後，若能從剩餘X中找出不相鄰 "o" 的 "o"，則無法 shatter

2. union positive intervals 決定了 2 個不相鄰 "o" 區間後，若能從剩餘data找出不相鄰data的組合，則無法shatter

n=4 時 $\left\{\begin{matrix} oxox \\ oxxo \\ xoxo \end{matrix}\right\}$ 為所有 o 不相鄰組合，

此時無論將任意 x 更改為 o，皆會與原來的 o 相鄰

故找不出第3個 o 區間，因此 union positive intervals 能 shatter n=4

n=5 時
1. oxoxx
2. oxxox
3. oxxxo
4. xoxox
5. xoxxo
6. xxoxo

為所有 (不相鄰 o) 且 (o區間長度皆為1) 的組合

此時 1 和 3 號組合可以從 x 區間再選出一不連續 "o"

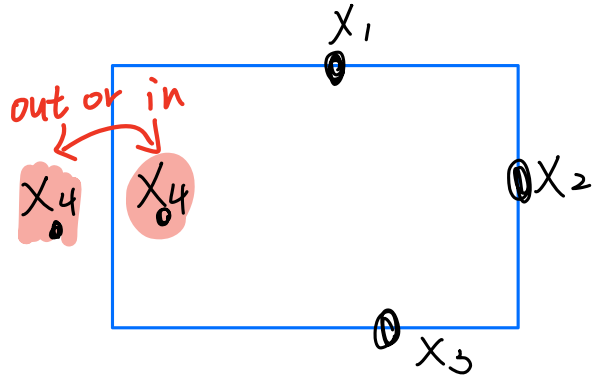
=> oxoxo，因此 n=5 為 break point，dvc = 4

[b] degree of freedom = 4 (長方形的 top-Left
top-right
bottom-Left
bottom-right)

, dvc = 4

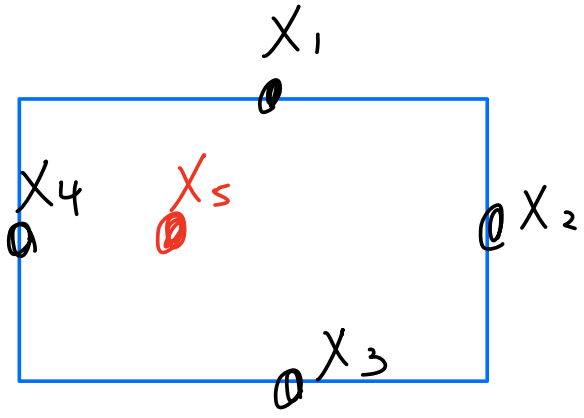
(以上根據 lecture 4. P40 簡略估計, 以下較詳細)

- 在 n=4 時, 將長方形的任3邊以任3点決定, 如下



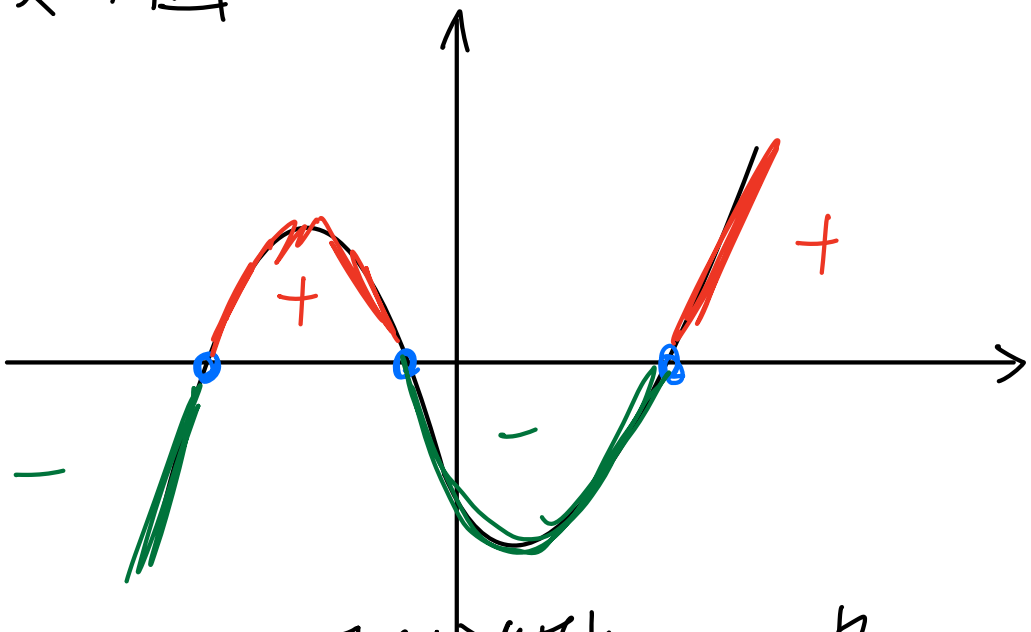
此時 x_4 可在矩形內 or 外, 故在3点決定後, 第4個仍可 shatter, 所以 n=4 可 shatter.

- 在 n=5 時, 將長方形任4邊以 $x_1 \sim x_4$ 決定, 其中 x_1 決定 max y, x_2 決定 max x
x_3 決定 min y, x_4 決定 min x
此時 x_5 必在矩形內, 被 $x_1 \sim x_4$ 決定 (如下圖)
故 x_5 無法被 shatter, 因此 dvc = 4



[d] 3次多項式最多會有3個根

如下圖：



因 $x_1 \sim x_n$ 最多變號 $n-1$ 次

故 $x_1 \sim x_4$ 最多變號 3 次

Ex: $h(x_1), h(x_2), h(x_3), h(x_4) \Rightarrow -, +, -, +$

故 $n=4$ 時可 shatter (變號3次)

而 $n=5$ 時，最多會變號 4 次，

故 $n=5$ 無法 shatter，$d_{VC}=4$.

(c)(1) $X = \begin{bmatrix} -X_1- \\ -X_2- \\ \vdots \\ -X_4- \end{bmatrix}$ when $n=3$.

special $X = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 \end{bmatrix}$ $\Rightarrow$ L.I. - invertible

for any $y = \begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{pmatrix}$, find W, s.t.

$\text{sign}(XW) = y \Leftarrow (XW) = y \Leftrightarrow W = X^{-1}y$

其中 $W_0 > 0$, 故:

$X^{-1} = \begin{bmatrix} X'_{11} & X'_{12} \cdots X'_{14} \\ \vdots \\ X'_{41} & X'_{42} \cdots X'_{44} \end{bmatrix}_{4\times4}$

$\Rightarrow W_0 = y_1 X'_{11} + y_2 X'_{12} + y_3 X'_{13} + y_4 X'_{14}$

when $y_i \cdot X_{1i} > 0$, then $W_0 > 0$

由於 $X'_{11} \sim X'_{14}$ 可為 + or −, 故任意 y 均可在 $W_0 > 0$ 的前提下得到

一組 $x'_{11}\sim x'_{14}$ 分別的內積為正，

ex: $y=\begin{bmatrix}0\\x\\0\\x\end{bmatrix}$ then $x'_{11}>0$, $x'_{12}<0$, $x'_{13}>0$, $x'_{14}<0$

故 $w_0>0$ 不改變 VC 維度，

$\Rightarrow dvc\geq 4$

(2) $X=\begin{pmatrix}-x_1-\\-x_2-\\-x_3-\\-x_4-\\-x_5-\end{pmatrix}$, L.D 不可逆.

special $X=\begin{pmatrix}1&0&0&0\\1&0&0&1\\1&0&1&0\\1&1&0&0\\1&1&1&1\end{pmatrix}_{5\times 4}$

∵(1)已說明 $w_0>0$ 不改變分的 dichotomy 數量

故存在某 w, s.t.

$$w^Tx_5=\underbrace{w^Tx_1}_{0}-\underbrace{w^Tx_2}_{X}-\underbrace{w^Tx_3}_{X}-\underbrace{w^Tx_4}_{X}>0$$

(由於 L.D 故 x_5 為 $x_1\sim x_4$ 的線性組合)

因此 $n=5$ 時 x_5 不可 shatter

$\Rightarrow dvc\leq 4$, by (1), (2) $\Rightarrow dvc=4$

6. 每個 dichotomy 對應 1 個 hypothesis

而 n 個 input 最多有 2^n 個 dichotomy.

因本題的 hypothesis = 1126

$\Rightarrow 2^n < 1126 \quad \Rightarrow n = 10.13$

$d_{vc}(H) \le \lg |H|$

$d_{vc}(H) \le \lg(1126)$

≤ 10.13

then $d_{vc}(H)$ 最大為 10.

7.

$$2M \cdot \exp(-2\varepsilon^2 N) = \delta$$

$$\Rightarrow -2\varepsilon^2 N = \ln\left(\frac{\delta}{2M}\right)$$

$$\Rightarrow \varepsilon^2 N = \frac{1}{2}\ln\left(\frac{2M}{\delta}\right)$$

$$\Rightarrow \varepsilon = \sqrt{\frac{1}{2N}\ln\frac{2M}{\delta}}$$

$$\because P[\,|E_{out}(g) - E_{out}(g^*)| < \varepsilon\,] \geq 1-\delta$$

$1-\delta$ 的 upper bound

$\Rightarrow$ ans is [C]

8. positive ray 的 $m_H(N) = 2N$

$\varepsilon = 0.1, \delta = 0.1$

$\Rightarrow \delta = 8 \cdot N \cdot \exp\left(\frac{-1}{8}(0.1)^2 \cdot N\right) = 0.1 = \delta$

在 [a], $\delta = 0.298$

[b], $\delta = 0.09$

[c], $\delta = 0.029$

[d], $\delta = 0.009$

[e], $\delta = 0.0028$

所以 $N = 11000$ 是滿足 $\delta < 0.1$ 的最少的 N [b]

9. ① $E(w) \simeq E(u) + b_E(u)^T (w-u) + \frac{1}{2}(w-u)^T A_E(u)(w-u)$

② $w = u+v$, $v = w-u$.

$$E(w) \simeq E(u) + b_E(u)^T \cdot v + \frac{1}{2} v^T A_E(u) \cdot v$$

$$\frac{\partial E(w)}{\partial w} = \frac{\partial}{\partial w}\left(E(u) + b_E(u)^T \cdot v + \frac{1}{2} v^T A_E(u) \cdot v\right)$$

$$\frac{\partial E(w)}{\partial w} = b_E(u) + v A_E(u)$$

$$\frac{\partial E(w)}{\partial w} = 0 \quad , \quad v = -b_E(u) A_E^{-1}(u)$$

$\Rightarrow$ ans : [b]

10. $\frac{\partial E_{in}(w)}{\partial w} = \frac{\partial}{\partial w} \frac{1}{N} \sum_{n=1}^{N} \ln\left(1 + \exp(-y_n w^T x_n)\right)$

令 $1 + \exp(-y_n w^T x_n) = \square$

$-y_n w^T x_n = 0$

$= \frac{1}{N} \sum_{n=1}^{N} \left(\frac{1}{\square}\right) \frac{\partial}{\partial w} \square$

$= \frac{1}{N} \sum_{n=1}^{N} \frac{1}{1 + \exp(0)} \cdot (-y_n x_n) \cdot \exp(0)$

$= \frac{1}{N} \sum_{n=1}^{N} \frac{\exp(0)}{1 + \exp(0)} \cdot (-y_n x_n)$

$= \frac{1}{N} \sum_{n=1}^{N} \frac{1}{1 + \exp(-0)} \cdot (-y_n x_n)$

$= \frac{1}{N} \sum_{n=1}^{N} h_t(y_n x_n) \cdot (-y_n x_n) = E_{in}(w)$

$$\frac{\partial E_{in}(w)}{\partial w} = \frac{1}{N}\sum_{n=1}^{N} \frac{\partial}{\partial w}\left(\frac{1}{1+\exp(y_n w^T x_n)}\right)\cdot(-y_n x_n)$$

$$= \frac{1}{N}\sum_{n=1}^{N} \frac{(-y_n x_n)\exp(y_n w^T x_n)}{(1+\exp(y_n w^T x_n))^2}\cdot(-y_n x_n)$$

$$= \frac{1}{N}\sum_{n=1}^{N} \frac{\exp(y_n w^T x_n)}{1+\exp(y_n w^T x_n)}\cdot\frac{1}{1+\exp(y_n w^T x_n)}\cdot(x_n x_n^T)$$

$$= \frac{1}{N}\sum_{n=1}^{N} \frac{1}{1+\exp(-y_n w^T x_n)}\cdot\frac{1}{1+\exp(y_n w^T x_n)}$$

$$\cdot(x_n x_n^T)$$

$$= \frac{1}{N}\sum_{n=1}^{N} h(y_n x_n)\cdot h(-y_n x_n)(x_n x_n^T)$$

$\Rightarrow$ ans : [d]

11. by def $XX^+X = X$ — ①

$X^+XX^+ = X^+$ — ②

$(XX^+)^T = XX^+$ — ③

$(X^+X)^T = X^+X$ — ④.

(a) by ③. 同乘 $X^T \Rightarrow X^T(XX^+)^T = X^TXX^+$

$\Rightarrow (XX^+X)^T = (X^TX)X^+$

by ① $\Rightarrow X = (X^TX)X^+$

X^TX 可逆 $\Rightarrow (X^TX)^{-1}X = X^+$

(b) $(XX^+)^2 = XX^+XX^+$

by def $XX^+X = X$

then $XX^+XX^+ = XX^+$

$(XX^+)^3 = (XX^+)^2(XX^+)$

$= (XX^+)^2 = XX^+$

so. $(XX^+)^K = XX^+$

(C) $U\Sigma V^T V\Sigma^+ U^T$

$= U\Sigma I \Sigma^+ U^T$

$= U\Sigma\Sigma^+ U^T = U I U^T = I$

由於 X 為 N by d+1 matrix

then XX^+ 為 $N \times N$ matrix.

(d) 假設 $X_{6\times5}$ 則 $X^{\dagger}_{5\times6}$

by (c) 的結論.

故 $XX^{\dagger}$ 為 $I_{6\times6}$

而 $X_{6\times5}$ 的 rank 最大為 5 (因第 6 列必為前 5 列的線性組合

$\Rightarrow \text{trace}(I_{6\times6}) = 6 \neq \text{rank}(X) = 5$

$\Rightarrow$ (d) false.

12. $D = \{(X_1, y_1), (X_2, y_2) \dots (X_n, y_n)\}$

$X_i \in R^d$, $y_i \in R$

find $W_* = \text{argmax}\, P(D|W_*)$

$$P(D|W_*) = P(y_1, y_2 \dots y_n \mid X_1, X_2 \dots X_n, W_*)$$

$$= \prod_{i=1}^{n} P(y_i \mid X_i, W_*)$$

$$= \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi a^2}} \exp\left(\frac{1}{-2a^2}(y_i - W^T X_i)^2\right)$$

$$= \left(\frac{1}{\sqrt{2\pi a^2}}\right)^n \exp\left(\frac{1}{-2a^2} \sum_{i=1}^{n} (y_i - W^T X_i)^2\right)$$

改寫 $\sum_{i=1}^{n}(y_i - W^T X_i) \Rightarrow \begin{pmatrix} y_1 - W^T X_1 \\ y_2 - W^T X_2 \\ \vdots \\ y_n - W^T X_n \end{pmatrix} = y - W^T X = y - X^T W$

$$= \left(\frac{1}{\sqrt{2\pi a^2}}\right)^n \exp\left(\frac{1}{-2a^2}(y - X^T W)^T (y - X^T W)\right)$$

$$P(D|W_*) = \left(\frac{1}{\sqrt{2\pi\alpha^2}}\right)^n \exp\left(\frac{1}{-2\alpha^2}(y-X^TW)^T(y-X^TW)\right)$$

what W_* can maximize $P(D|W_*)$?

我們需要找出 $\underset{W}{\operatorname{argmin}}(y-X^TW)^T(y-X^TW)$

假設 $G(W) = (y-X^TW)^T(y-X^TW)$

$$\frac{\partial G(W)}{\partial W} = \left(-X^T(y-X^TW)\right)\times 2$$

$$= X^T(X^TW-y)\times 2$$

$$= 2(XX^TW - X^Ty)$$

令 $\frac{\partial G(W)}{\partial W} = 0 = XX^TW - X^Ty$

$$\Rightarrow X^Ty = XX^TW$$

$$\Rightarrow W = (XX^T)^{-1}X^Ty$$

所以當 $W_* = (XX^T)^{-1}X^Ty$ 時 $P(D|W_*)$ 有最大值.

```python
def generateData(traNUM,valNUM,addOutliers = 0):

    # 產生200筆 training data traVal
    # 產生5000筆 testing data testVal
    traLabel = np.random.normal(0, 1, traNUM).reshape(traNUM,1)
    valLabel = np.random.normal(0, 1, valNUM).reshape(valNUM,1)

    # traData = np.zeros((200,2))
    # valData = np.zeros((5000,2))
    traData = []
    valData = []
    for _ in traLabel:
        if _[0] >= 0:
            #     產生 var = 0.6 ,mean  = [2 , 3]
            x1 = np.random.normal(2, 0.6 ** 0.5,1)[0]
            x2 = np.random.normal(3, 0.6 ** 0.5,1)[0]

        else:
            # 產生var = 0.4 ,mean  = [0 , 4]
            x1 = np.random.normal(0, 0.4 ** 0.5,1)[0]
            x2 = np.random.normal(4, 0.4 ** 0.5,1)[0]
        traData.append([1,x1, x2])
    for _ in valLabel:
        if _[0] >= 0:
            #     產生 var = 0.6 ,mean  = [2 , 3]
            x1 = np.random.normal(2, 0.6 ** 0.5,1)[0]
            x2 = np.random.normal(3, 0.6 ** 0.5,1)[0]
        else:
            # 產生var = 0.4 ,mean  = [0 , 4]
            x1 = np.random.normal(0, 0.4 ** 0.5,1)[0]
            x2 = np.random.normal(4, 0.4 ** 0.5,1)[0]
        valData.append([1,x1, x2])
    traData = np.asarray(traData,dtype=np.float64)
    valData = np.asarray(valData,dtype=np.float64)

    if addOutliers :
        Noise = []
        NoiseLabel = [1 for _ in range(20)]
        for _ in range(20):
            x1 = np.random.normal(6, 0.3 ** 0.5,1)[0]
            x2 = np.random.normal(0, 0.1 ** 0.5,1)[0]
            Noise.append([1,x1,x2])

        Noise = np.asarray(Noise,dtype=np.float64)
        NoiseLabel = np.asarray(NoiseLabel,dtype=np.float64).reshape(20,1)

        traData = np.concatenate((traData,Noise),axis=0)
        traLabel = np.concatenate((traLabel,NoiseLabel),axis=0)

    traLabel[traLabel >= 0] = 1
    traLabel[traLabel < 0] = -1
    valLabel[valLabel >= 0] = 1
    valLabel[valLabel < 0] = -1

    return [[traData,traLabel],[valData,valLabel]]


def meanErr(iterNUM,noiseTF = 0):
    lin_errinBag = []
    lin_errInOutBag = []
    lin_log_errout01Bag = []
    lin_log_errout01Bag_noise = []

    for _ in range(iterNUM):

        DataSET = generateData(200, 5000, 0)
        M_plus = np.linalg.pinv(DataSET[0][0])
        W_lin = np.dot(M_plus, DataSET[0][1])
        errin = np.dot(DataSET[0][0], W_lin) - DataSET[0][1]
        sqrErrMean = np.sum(errin ** 2) / len(errin)
        lin_errinBag.append(sqrErrMean)

        #for Q14
        lin_errin01  = linValidate(W_lin,DataSET[0][0],DataSET[0][1])
        lin_errout01 = linValidate(W_lin,DataSET[1][0],DataSET[1][1])
        lin_errInOutBag.append(abs(lin_errout01-lin_errin01))

        #for Q15
        log_errout01= logisticGrad(500,DataSET[0][0],DataSET[0][1],DataSET[1][0],DataSET[1][1])
        lin_log_errout01Bag.append([lin_errout01,log_errout01])
        #for Q16
        if noiseTF == 1:

            DataSET = generateData(200, 5000, 1)
            M_plus = np.linalg.pinv(DataSET[0][0])
            W_lin = np.dot(M_plus, DataSET[0][1])

            lin_errout01 = linValidate(W_lin, DataSET[1][0], DataSET[1][1])
            log_errout01= logisticGrad(500,DataSET[0][0],DataSET[0][1],DataSET[1][0],DataSET[1][1])
            lin_log_errout01Bag_noise.append([lin_errout01,log_errout01])


    lin_errIn01 = sum(lin_errinBag) / iterNUM
    lin_errInOutDiff01 = sum(lin_errInOutBag)[0] / iterNUM
    lin_log_errout01Bag = np.array(lin_log_errout01Bag)
    lin_log_err = np.mean(lin_log_errout01Bag,axis=0).reshape(-1)

    lin_log_errout01Bag_noise = np.array(lin_log_errout01Bag_noise)
    lin_log_errout01Bag_noise = np.mean(lin_log_errout01Bag_noise,axis=0).reshape(-1)

    return lin_errIn01,lin_errInOutDiff01,lin_log_err,lin_log_errout01Bag_noise



def linValidate(W_lin,valDATA,valLabel):

    pred = np.dot(valDATA,W_lin)
    pred[pred >= 0] = 1
    pred[pred < 0] = -1
    err01 = sum(pred != valLabel) / len(valLabel)
    return err01

def logisticGrad(iterNUM, traData, traLabel,valData,valLabel):

    # traData 200,3
    # traLabel 200,1
    # weight 3,1
    W = np.zeros((3,1),dtype=np.float64)
    theta = lambda x: (1 / (1 + np.exp(-x)))

    for _ in range(iterNUM):

        # s = y_n * W^T * x_n
        s = np.dot(traData, W) * traLabel*(-1)
        grad = ( theta(s)* (  (-1)*traLabel * traData))
        grad = np.mean(grad, axis=0).reshape(3, 1)
        W = W - (0.1* grad)

    s = np.dot(valData, W)
    pred = theta(s)
    pred[pred >= 0.5] = 1
    pred[pred < 0.5] = -1
    res = sum(pred != valLabel) / len(valLabel)
    return res
```