EE 5904 Homework 1

Date No.

Q1. 1) $\varphi(v) = av + b$

$v(n) = w_1 x_1 + w_2 x_2 + \cdots + w_m x_m + b_k$

$\varphi = av + b = \xi$

$\Rightarrow a(w_1 x_1 + w_2 x_2 + \cdots + w_m x_m) + ab_k + b - \xi = 0$

It is a hyper-plane on m-dimension.

2) $\varphi(v) = \frac{1}{1+e^{-2v}} = \xi$

$1 = \xi + \xi e^{-2v}$

$\ln 1 = 0 = \ln\xi + \ln\xi - 2v$

$v = \ln\xi \qquad \xi \neq 0$

$\Rightarrow w_1 x_1 + w_2 x_2 + \cdots + w_m x_m + (b_k - \ln\xi) = 0$

It is a hyper-plane on m-dimension if $\xi \neq 0$

3) $\varphi(v) = e^{-\frac{v^2}{2}} = \xi$

$-\frac{v^2}{2} = \ln\xi$

$v^2 = -2\ln\xi$

$v = (-2\ln\xi)^{\frac{1}{2}} \qquad 0 < \xi < 1$

$w_1 x_1 + w_2 x_2 + \cdots + w_m x_m + \left(b_k - (-2\ln\xi)^{\frac{1}{2}}\right) = 0$

It is a hyper-plane on m-dimension if $0 < \xi < 1$

Date No.

Q2. Truth Table of XOR

x_1	0	1	0	1
x_2	0	0	1	1
y	0	1	1	0

Assume there is a line that separate the two classes $y=0$ and $y=1$

$w_1x_1 + w_2x_2 + b > 0$ for class $y=1$
$w_1x_1 + w_2x_2 + b < 0$ for class $y=0$

For $y=1$: $x_1=1$, $x_2=0$
or $x_1=0$, $x_2=1$

$\Rightarrow$ $w_1+b>0$
$w_2+b>0$ $\Rightarrow$ $w_1+w_2+2b>0$ ①

For $y=0$: $x_1=0$, $x_2=0$
or $x_1=1$, $x_2=1$
$\Rightarrow$ $b<0$ ③
$w_1+w_2+b<0$ $\Rightarrow$ $0>w_1+w_2+b$ ②

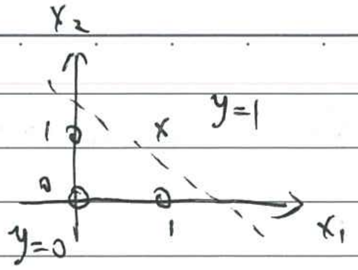
Summing up ① and ②
$\Rightarrow$ $2b>b$
which contradicts since $b<0$ ③

So XOR is not linearly separable.

Date No.

Q3 a. AND

x_1	0	0	1	1
x_2	0	1	0	1
y	0	0	0	1

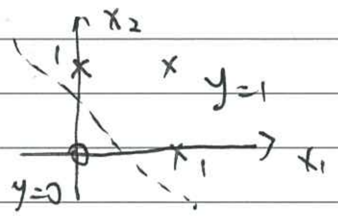


$$v = x_1 + x_2 - 1.5 \begin{cases} >0, \text{ class } y=1 \\ <0, \text{ class } y=0 \end{cases}$$

$$w^T = (-1.5, 1, 1)$$

OR

x_1	0	0	1	1
x_2	0	1	0	1
y	0	1	1	1

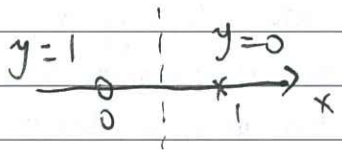


$$v = x_1 + x_2 - 0.5 \begin{cases} >0, \text{ class } y=1 \\ <0, \text{ class } y=0 \end{cases}$$

$$w^T = (-0.5, 1, 1)$$

COMPLEMENT

x	0	1
y	1	0

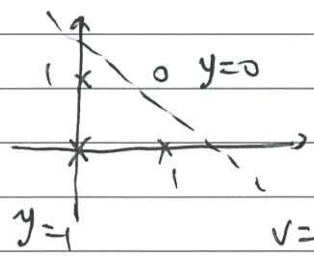


$$v = -x + 0.5 \begin{cases} >0, \text{ class } y=1 \\ <0, \text{ class } y=0 \end{cases}$$

$$w^T = (0.5, -1)$$

NAND

x_1	0	0	1	1
x_2	0	1	0	1
y	1	1	1	0



$$v = -x_1 - x_2 + 1.5 \begin{cases} >0, \text{ class } y=1 \\ <0, \text{ class } y=0 \end{cases}$$

$$w^T = (1.5, -1, -1)$$

Q3 b

Comparing the weights obtained in Q3 a and obtained by computer simulation (Appendix Q3 b (i))

AND

off line : $w^T = (-1.5, 1, 1)$

simulation : $w^T = (-2.2, 1.8, 0.8)$

OR

off line : $w^T = (-0.5, 1, 1)$

simulation : $w^T = (-0.7, 1.5, 0.9)$

COMPLEMENT

off line : $w^T = (0.5, -1)$

simulation : $w^T = (0.8, -1.2)$

NAND

off line : $w^T = (1.5, -1, -1)$

simulation : $w^T = (2.8, -2.2, -1.2)$

Refer to Appendix Q3 b (ii),
we can see that when learning rate increases from 0.01 to 1, the iterations that get stable w become less and less, about 14 iterations. However, when learning rate keeps increasing from 1 to 10, the iterations that get stable weight become more and stable, always 18 iterations.

Q3. c. Please refer to Appendix Q3 c.

We can see that the weights keep changing and will never get the stable values.

Also, there is not a line that can separate the 2 classes $y=1$ and $y=0$.

Q4. a, The regression matrix

$$X = \begin{bmatrix} (1, 0) \\ (1, 0.8) \\ (1, 1.6) \\ (1, 3) \\ (1, 4) \\ (1, 5) \end{bmatrix}$$

$$X^T = \begin{pmatrix} 1, 1, 1, 1, 1, 1 \\ 0, 0.8, 1.6, 3, 4, 5 \end{pmatrix} \qquad d = \begin{pmatrix} 0.5 \\ 1 \\ 4 \\ 5 \\ 6 \\ 9 \end{pmatrix}$$

$$X^T X = \begin{pmatrix} 6, & 14.4 \\ 14.4, & 53.2 \end{pmatrix}$$

$$(X^T X)^{-1} = \begin{pmatrix} \frac{665}{1398}, & \frac{-30}{233} \\ \frac{-30}{233}, & \frac{25}{466} \end{pmatrix}$$

$$(X^T X)^{-1} X^T = \begin{pmatrix} \frac{655}{1398}, & \frac{521}{1398}, & \frac{377}{1398}, & \frac{125}{1398}, & \frac{-55}{1398}, & \frac{-225}{1398} \\ \frac{-30}{233}, & \frac{-20}{233}, & \frac{-10}{233}, & \frac{15}{466}, & \frac{20}{233}, & \frac{65}{466} \end{pmatrix}$$

$$W = (X^T X)^{-1} X^T d = \begin{pmatrix} \frac{361}{932} \\ \frac{375}{233} \end{pmatrix} = \begin{pmatrix} 0.387 \\ 1.609 \end{pmatrix} \qquad \begin{matrix} b = 0.387 \\ w = 1.609 \end{matrix}$$

Please refer to Appendix Q4 a for the fitting line.

Date No.

Q4 b, learning rate $\eta = 0.01$

Please refer to Appendix Q4 b.

$$W^T = \begin{pmatrix} 0.46376698 \\ 1.6033277 \end{pmatrix} \qquad \begin{matrix} b = 0.46376698 \\ w = 1.6033277 \end{matrix}$$

We can see that the weights will finally converge.

Q4 c. Please refer to Appendix Q4 c.

LLS method	LMS method
$b = 0.387339$	$b = 0.46376698$
$W = 1.60944$	$w = 1.6033277$

We can see that the results obtained from LLS method and LMS method are quite similar.

Q4 d. Please refer to Appendix Q4 d for the results.

if learning rate η increases,
the weights will take LESS epochs to get stable.

If learning rate η decreases.
the weights will take MORE epochs to get stable.

Date No.

Q5. Input output pairs : $((x(1), d(1)), (x(2), d(2)), \cdots (x(n), d(n)))$

$$x = (x_1, x_2, \cdots, x_m)^T$$

$$y(x) = w_1 x_1 + w_2 x_2 + \cdots + w_m x_m = w^T x$$

$$e(i) = d(i) - y(i) \Rightarrow e = d - y$$

$$y = (y(1), y(2), \cdots y(n))^T \quad d = (d(1), d(2), \cdots d(n))^T$$

$$y(i) = w^T x(i) = x(i)^T w$$

$$y = \begin{pmatrix} y(1) \\ y(2) \\ \vdots \\ y(n) \end{pmatrix} = \begin{pmatrix} x(1)^T w \\ x(2)^T w \\ \vdots \\ x(n)^T w \end{pmatrix} = \begin{pmatrix} x(1)^T \\ x(2)^T \\ \vdots \\ x(n)^T \end{pmatrix} = Xw \quad X = \begin{pmatrix} x(1)^T \\ x(2)^T \\ \vdots \\ x(n)^T \end{pmatrix}$$

$$e = d - Xw \qquad \frac{de}{dw} = -X$$

$$J(w) = \sum_{i=1}^{n} r(i) e(i)^2 = \sum_{i=1}^{n} r(i) (d(i) - y(x(i)))^2$$

let $R = \begin{pmatrix} r(1) & 0 & 0 \\ 0 & r(2) & 0 \\ 0 & 0 & \ddots r(n) \end{pmatrix}$

$$J(w) = e^T R e \qquad \frac{dJ}{de} = 2e^T R$$

By chain rule: $\frac{dJ}{dw} = \frac{dJ}{de}\frac{de}{dw} = -2e^T RX = 0$

$$\Rightarrow e^T RX = 0$$
$$\Rightarrow (d - Xw)^T RX = 0$$
$$\Rightarrow (d^T - w^T X^T) RX = 0$$
$$\Rightarrow d^T RX - w^T X^T RX = 0$$
$$d^T RX = w^T X^T RX$$
$$(w^T X^T RX)^T = (d^T RX)^T$$
$$(RX)^T \cdot (w^T X^T)^T = (RX)^T \cdot d$$
$$X^T R^T X w = X^T R^T d$$
$$\underline{\underline{w = (X^T R^T X)^{-1} X^T R^T d}}$$

APPENDIX

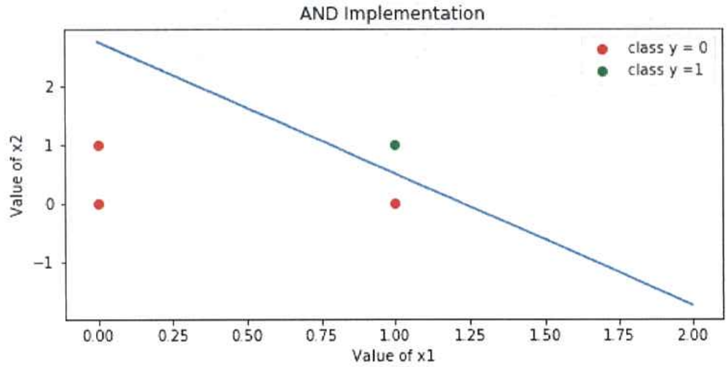
Q3 b (i).

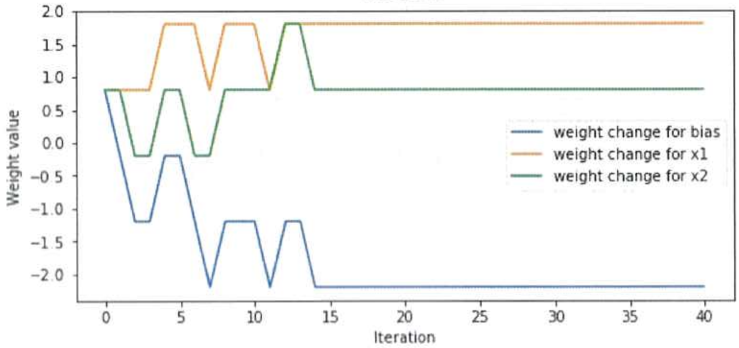
AND Implementation

n = 1

Initial weights: [0.8, 0.8, 0.8]

Final weights: [-2.2, 1.8, 0.8]



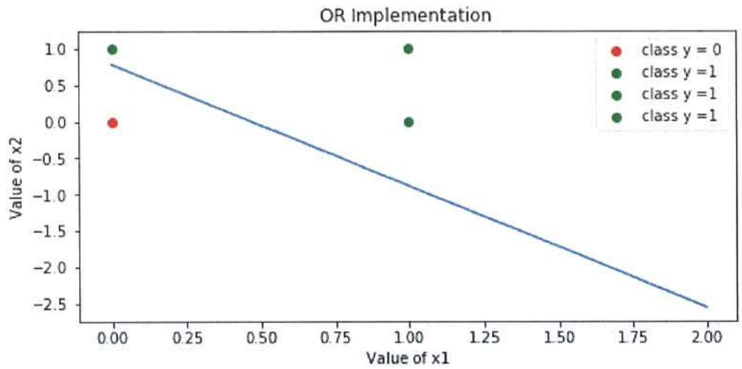


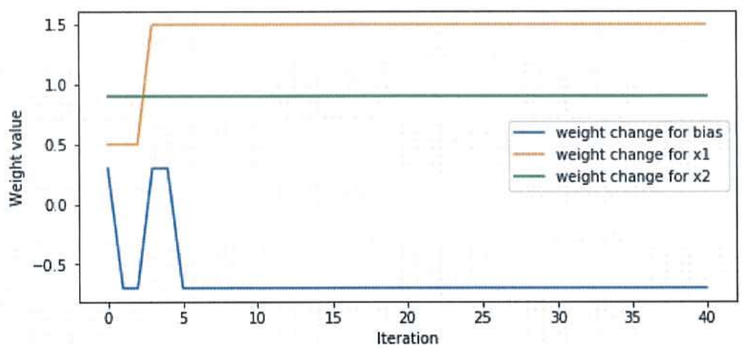
OR Implementation

n = 1

Initial weights: [0.3, 0.5, 0.9]

Final weights: [-0.7, 1.5, 0.9]



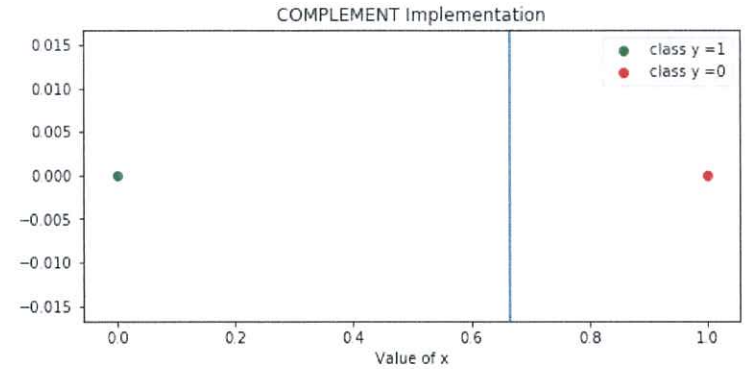


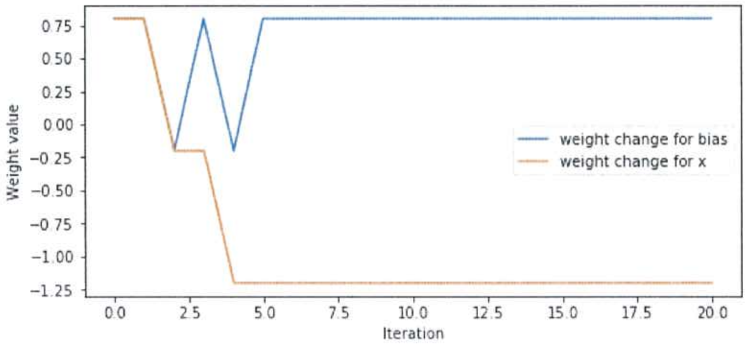
COMPLEMENT Implementation

n = 1

Initial weights: [0.8, 0.8]

Final weights: [0.8, -1.2]



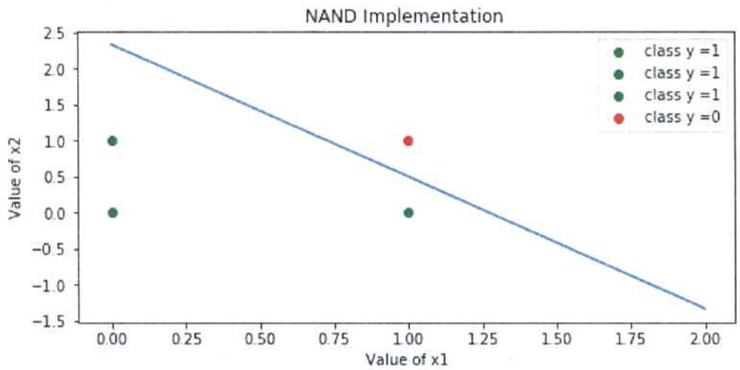


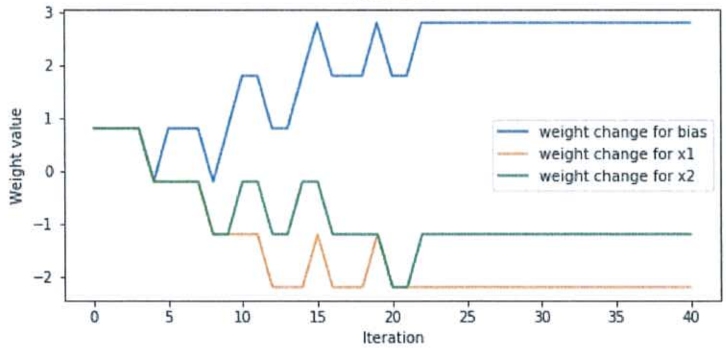
NAND Implementation

n = 1

Initial weights: [0.8, 0.8, 0.8]

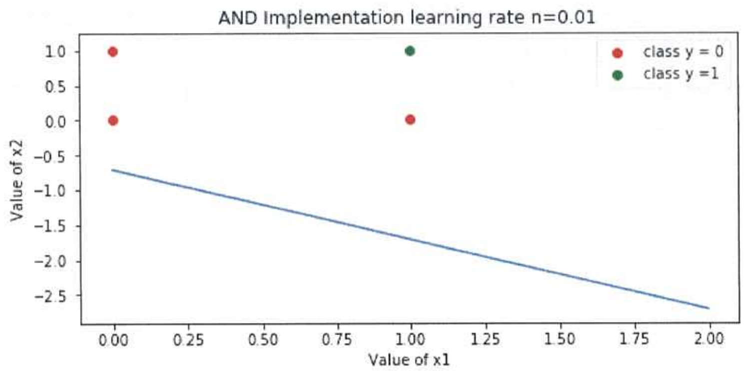
Final weights: [2.8, -2.2, -1.2]

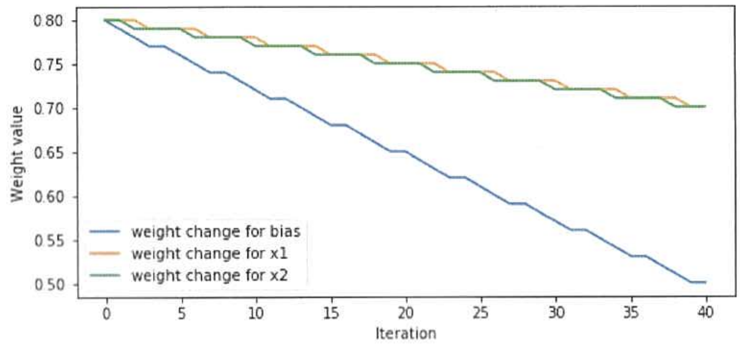


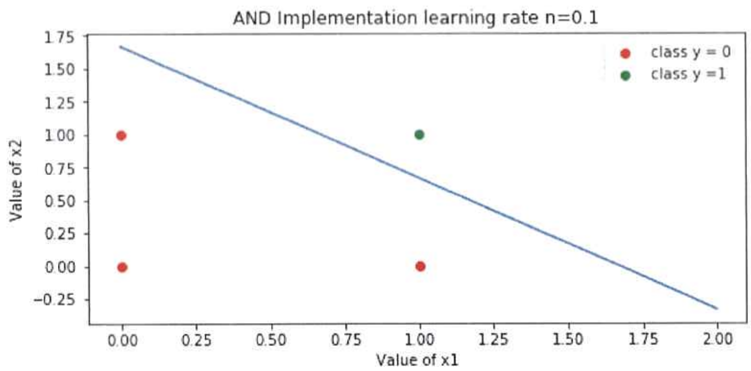


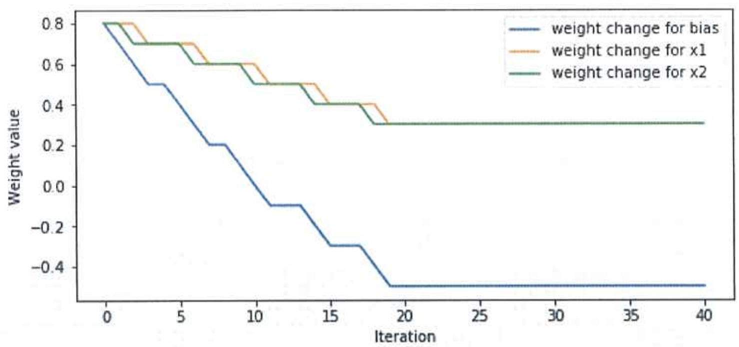
Q3 b (ii).

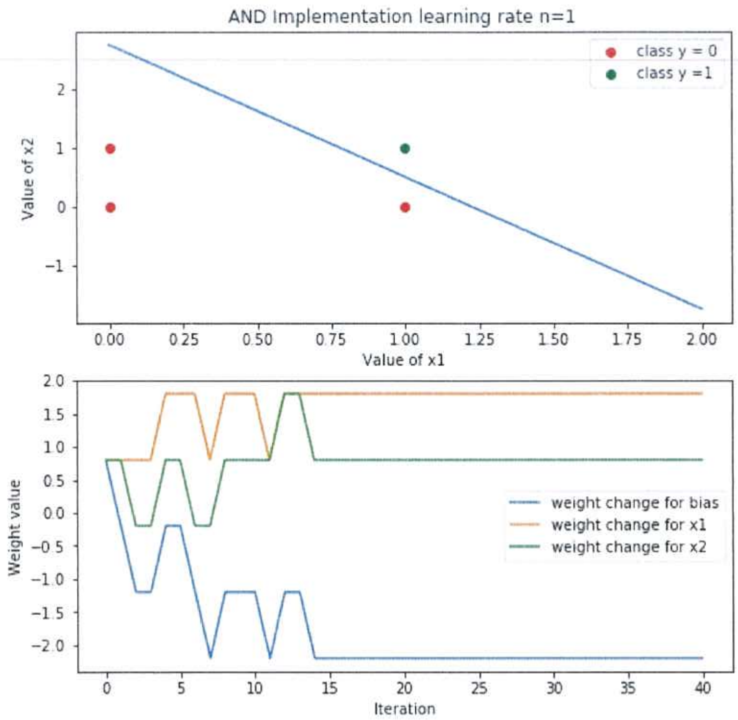
For AND implementation with different learning rate n=0.01, n=0.1, n=1, n=5, n=10

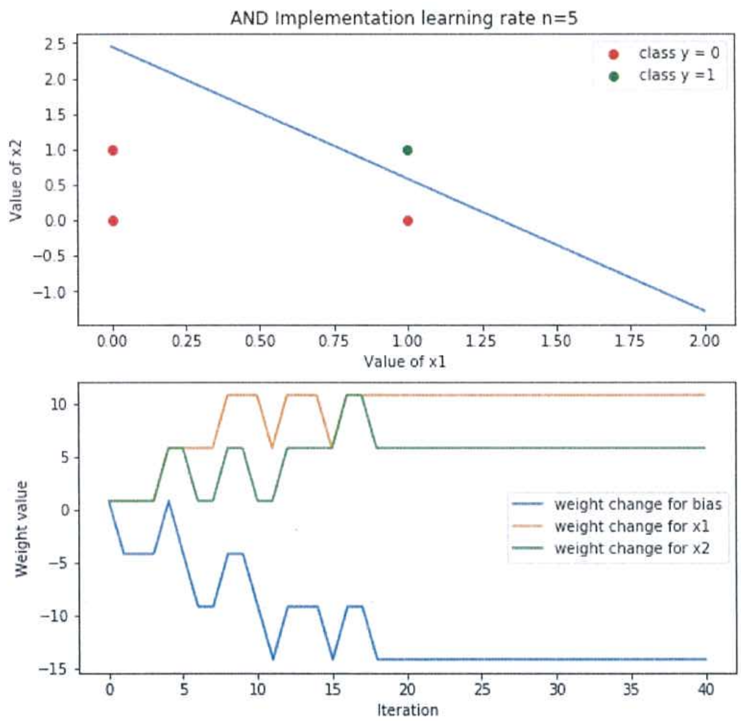


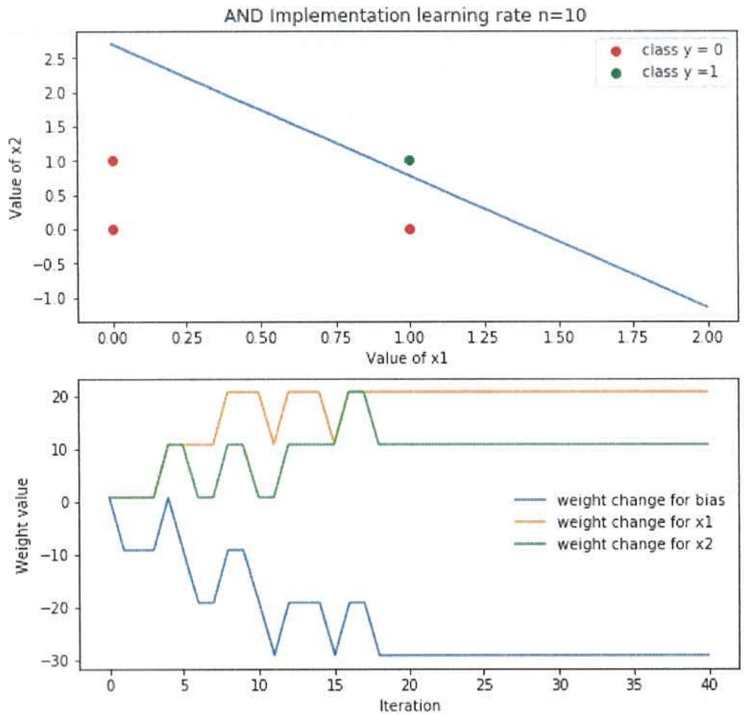







AND Implementation learning rate n=1
class y = 0
class y =1
Value of x2
Value of x1
weight change for bias
weight change for x1
weight change for x2
Weight value
Iteration


AND Implementation learning rate n=5
class y = 0
class y =1
Value of x2
Value of x1
weight change for bias
weight change for x1
weight change for x2
Weight value
Iteration



We can see from above comparison of AND implementation with different learning rates that when learning rate increases to 1, the weights values will take less iterations to converge about 14 iterations.

However, when learning rate keeps increasing to larger value, the weights will still converge, but a bit more iterations about 18 epochs to converge.

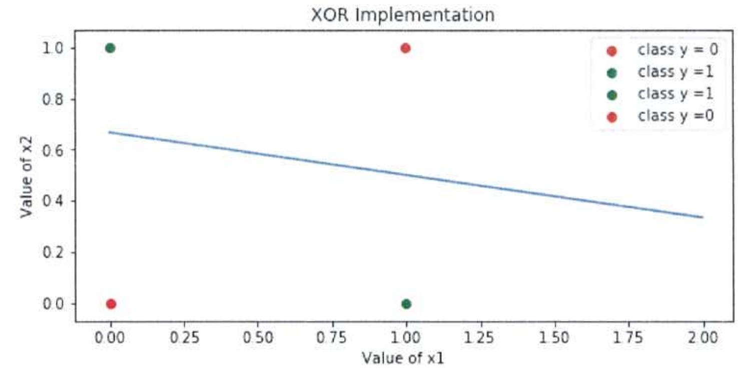
Then the number of iterations will keep the same when learning rate keeps increasing.

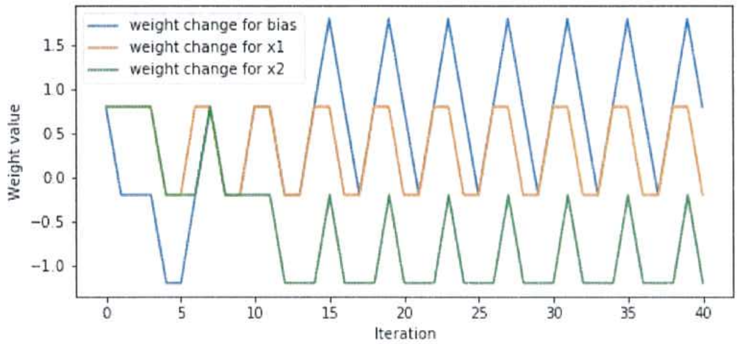
Q3 c.

XOR Implementation

n = 1

Initial weights: [0.8, 0.8, 0.8]





We can see that for XOR implementation, the weights keeps changing and will never get a stable state. And there is not a line that can separate the class y=1 and y=0.

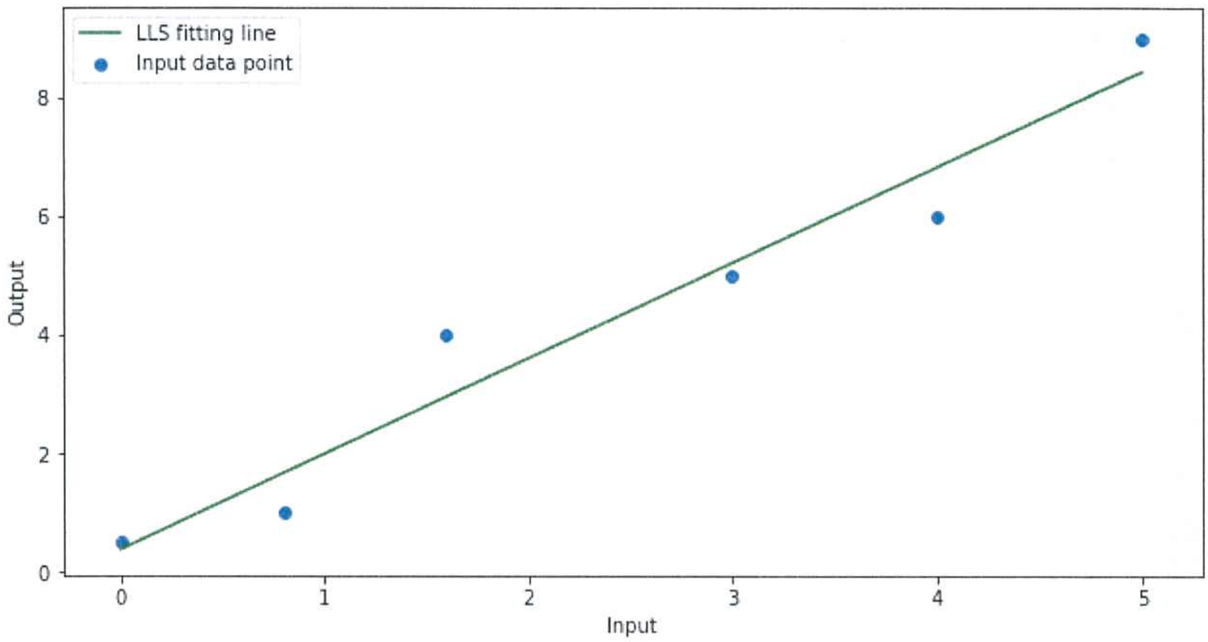
Q4 a.

(LLS) method fitting line

Final weight: [0.387339, 1.60944]

b = 0.387339

w = 1.60944



Q4 b.

LMS method for 100 epochs fitting line and weights change

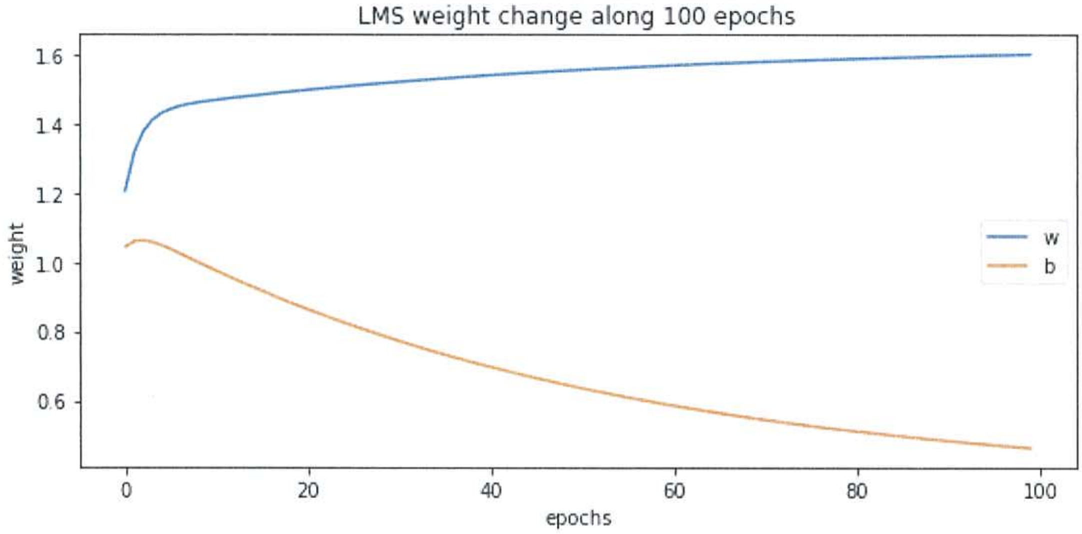
n = 0.01

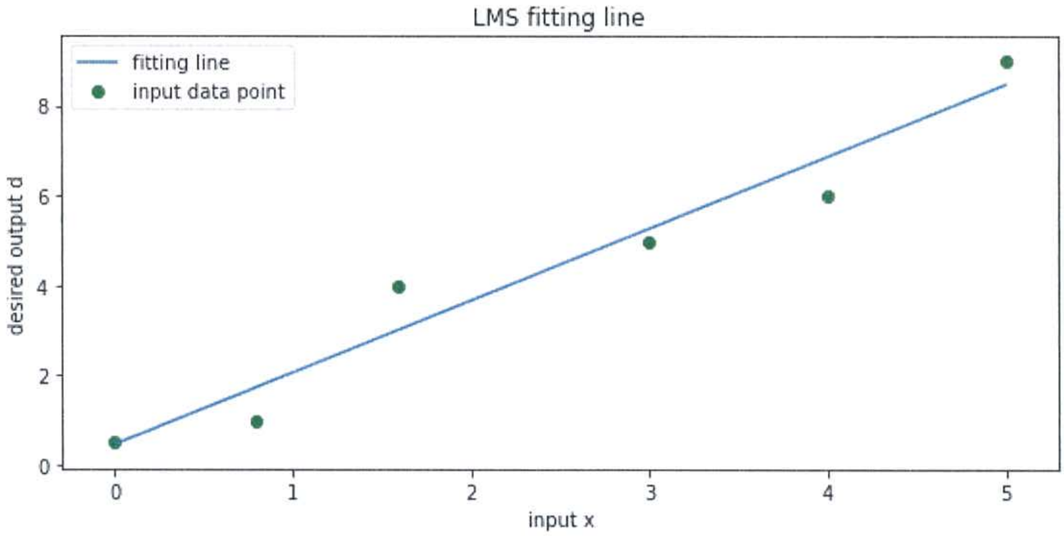
Initial weight array: [1, 1]

Final weight array after 100 epochs: [0.46376698, 1.60333277]

b = 0.46376698

w = 1.60333277





We can see from above graph for learning rate n = 0.01, the w and b will converge to a stable values.

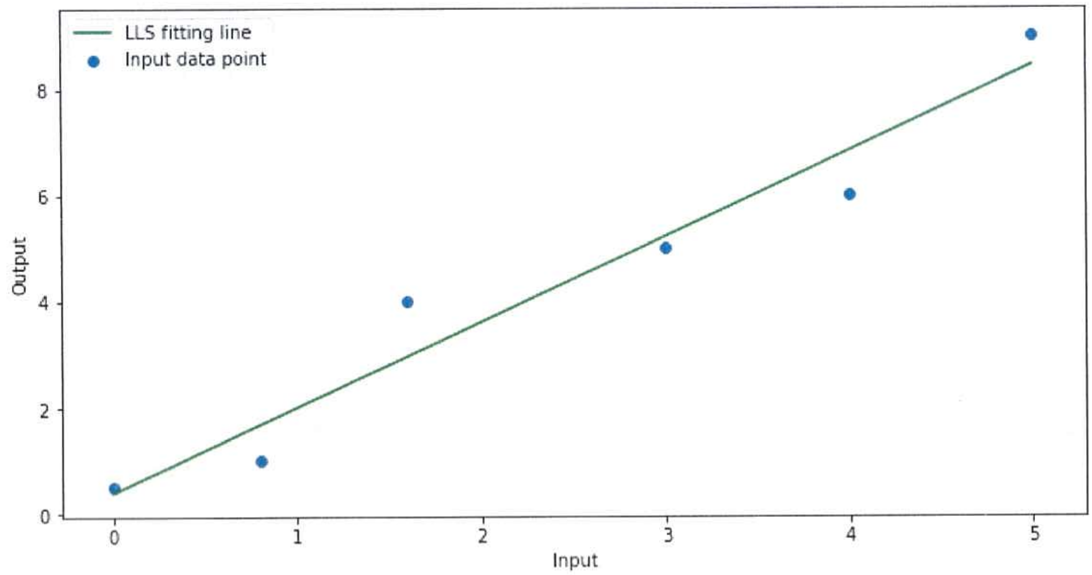
Q4 c.

(LLS) method fitting line

Final weight: [0.387339, 1.60944]

b = 0.387339

w = 1.60944



Comparing LLS method and LMS method after 100 epochs.

LLS

b = 0.387339

w = 1.60944

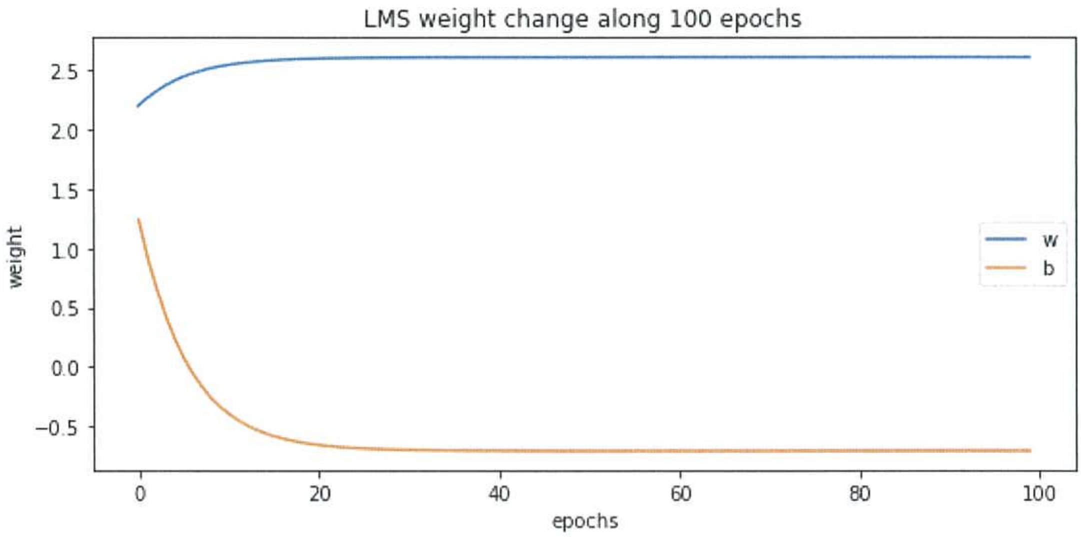
LMS

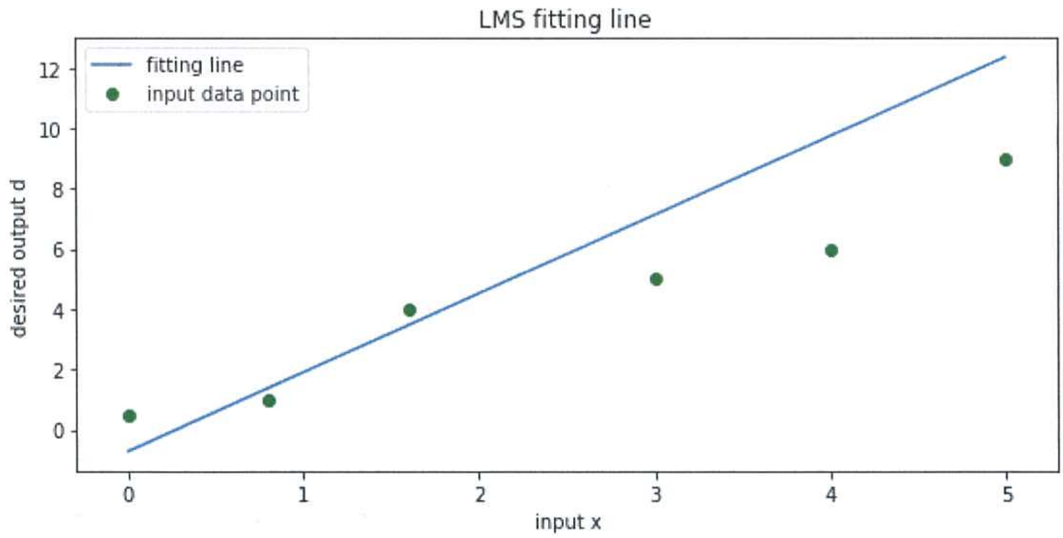
b = 0.46376698

w = 1.60333277

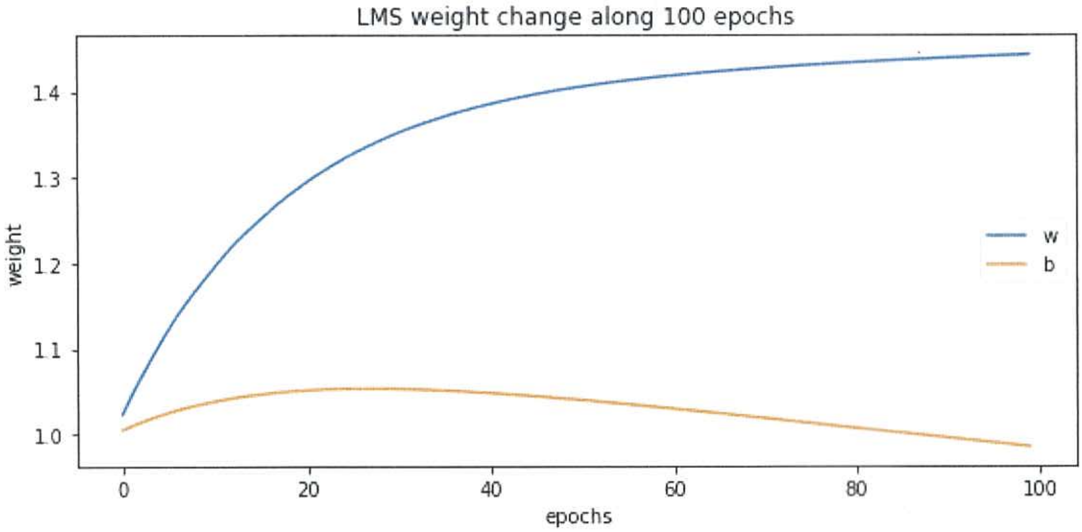
The results of b and w obtained by LLS method and LMS method are quite similar.

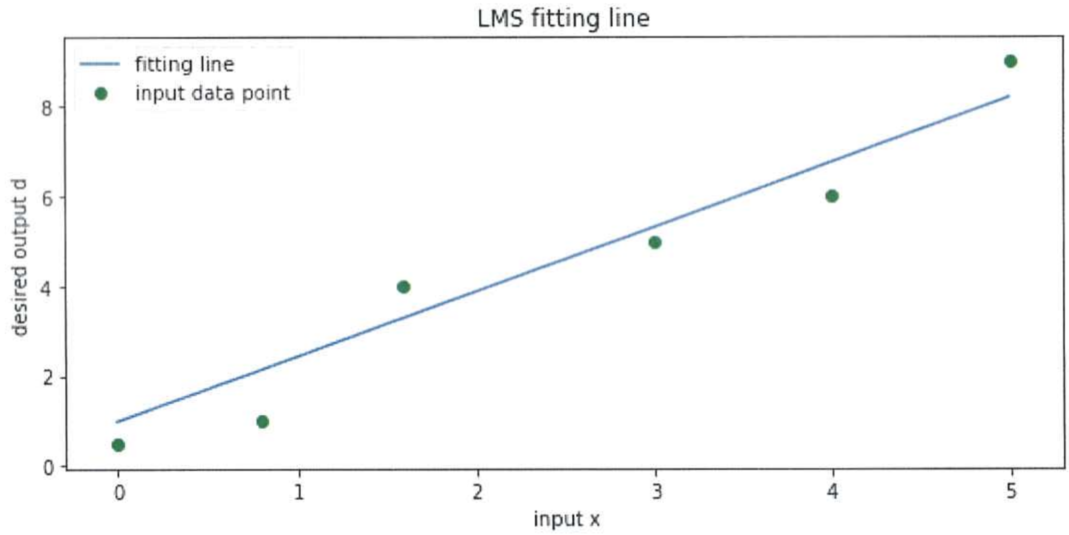
Q4 d.
n = 0.1
Initial weight array: [1, 1]
Final weight array after 100 epochs: [-0.70721958, 2.61155105]
b = -0.70721958
w = 2.61155105





n = 0.001
Initial weight array: [1, 1]
Final weight array after 100 epochs: [0.98491712, 1.44256221]
b = 0.98491712
w = 1.44256221





We can concluded from above images that when learning rate n decrease to 0.001, the w and b will take more epochs to converage stable, even more than 100 epochs to converage stable. And when learning rate n increase to 0.1, the w and b will converage to stable very fast.