EE5904 Neural Networks: Homework 1

Jin Lexuan A0232696W

February 7, 2022

Q1.

Considering the bias term,observation vector:

$$x = [1, x_1, x_2, \cdots, x_m]^T$$

Weight:

$$w = [b, w_1, w_2, \cdots, w_m]^T$$

So the induced local field can be written as:

$$v = w^T x = \sum_{i=1}^{m} x_i w_i + b$$

Consider the following activation function:

1)

For logistic function: $\varphi(v) = \frac{1}{1+e^{-v}}$
Let $\xi = \varphi(v)$, then get $v = -\ln(\frac{1}{\xi} - 1)$
when $0 < \xi < 1, v$ is a constant. So the decision boundary is a hyper-plane.

2)

For Bell-shaped Gaussian function: $\varphi(v) = e^{-\frac{(v-m)^2}{2}}$
Let $\xi = \varphi(v)$, then get $v = \pm\sqrt{-2\ln\xi} + m$
when $0 < \xi < 1, v$ is not a constant. So the decision boundary is not a hyper-plane.

3)

For Softsign function: $\varphi(v) = \frac{v}{1+|v|}$
Let $\xi = \varphi(v)$, then get:

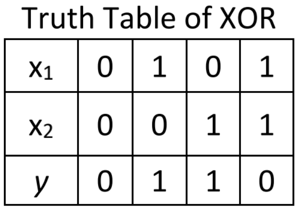
$$\begin{cases} v = \frac{\xi}{1-\xi}, v < 0 \\ v = \frac{\xi}{1+\xi}, v \geq 0 \end{cases}$$

When $\xi = 0$, v is a constant. So the decision boundary is a hyper-plane.

Q2.

Using proof by contradiction.Assuming XOR problem is linearly separable.Then existing b, x_1, x_2 that:

$$\begin{cases} b + w_1 x_1 + w_2 x_2 > 0,\ y = 1 \\ b + w_1 x_1 + w_2 x_2 < 0,\ y = 0 \end{cases}$$



Truth Table of XOR

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

Figure 1: XOR truth table

According to the truth table of XOR(shown in Figure 1).It can be deduced that:

$$\begin{cases} b < 0, \\ b + w_1 > 0 \\ b + w_2 > 0 \\ b + w_1 + w_2 < 0 \end{cases}$$

Then we get $2b + w_1 + w_2 > 0$ and b $+ w_1 + w_2 < 0$. From this two inequality, it can be derived thatb > 0, which conflicts with b < 0, Since contradiction happens, the assumption is wrong, so XOR problem is not linearly separable.

Q3.

a).

Weights by offline calculation:
AND:

$$\mathbf{w} = [-1.5, 1, 1]^T, v = x_1 + x_2 - 1.5$$

OR:

$$\mathbf{w} = [-0.5, 1, 1]^T, v = x_1 + x_2 - 0.5$$

COMPLEMENT:

$$\mathbf{w} = [0.5, -1]^T, v = -x_1 + 0.5$$

NAND:

$$\mathbf{w} = [1.5, -1, -1]^T, v = -x_1 - x_2 + 1.5$$

For each logic function, when v $\geq 0, y = 1, otherwise\ y = 0$. The implementation of logic functions will be shown in b).

b).

Learning procedure results, initial weights are chosen randomly and learning rate is 1.0, trajectories are shown from Figure2 to Figure5. From these four figures, it shows that weights change at initial epochs, then keep stable(converge). The reason is that Weights only change when misclassification happens, once the decision boundary can fit all the points, weights will not change.

The comparison of off-line calculation and learning procedure is shown in Table1. The parameter of each weight is a little different, since the learning procedure starts with randomly chosen weight. However, they can do the correct classification. The decision boundary figures of off-line calculation and learning procedure are shown from Figure 6 to Figure 9. The result shows that there is difference between two kinds of decision boundaries, but both of them can classify situation correctly.

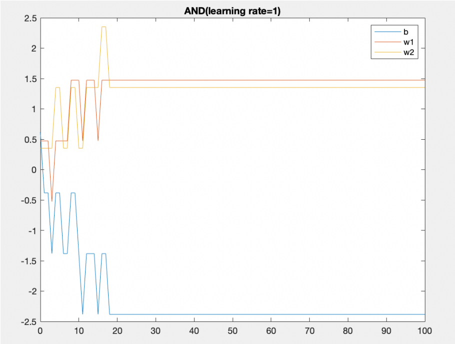


Figure 2: AND weights(learning rate=1)

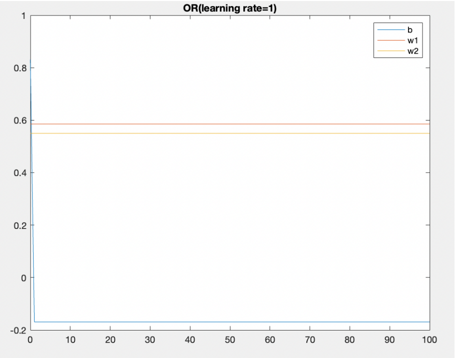


Figure 3: OR weights(learning rate=1)

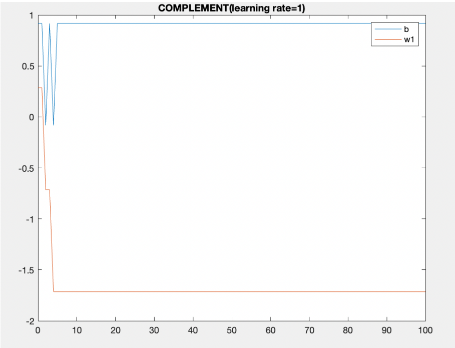


Figure 4: COMPLEMENT(learning rate=1)

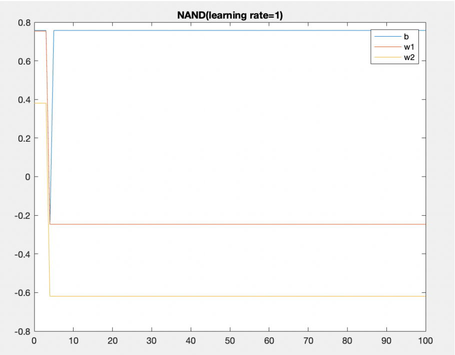


Figure 5: NAND weights(learning rate=1)

	b	w_1	w_2
AND(off-line)	-1.5	1	1
AND(learning)	-2.3840	1.4733	1.3517
OR(off-line)	-0.5	1	1
OR(learning)	-0.1692	0.5853	0.5497
COMPLEMENT(off-line)	0.5	-1	NA
COMPLEMENT(learning)	0.9172	-1.7142	NA
NAND(off-line)	1.5	-1	-1
NAND(learning)	0.7572	-0.2463	-0.6196

Table 1: Comparison of off-line calculation and learning procedure

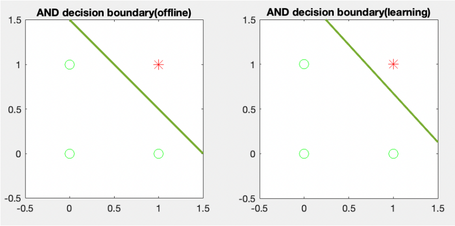


Figure 6: AND decision boundary

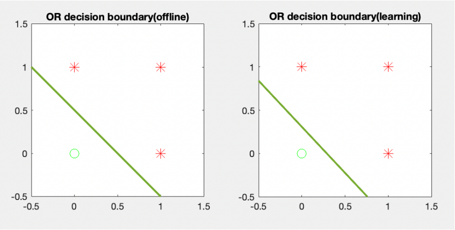


Figure 7: OR decision boundary

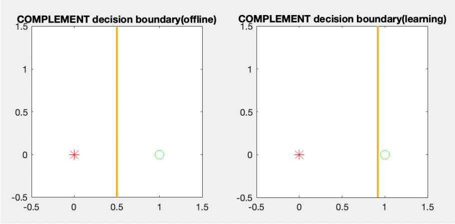


Figure 8: COMPLEMENT decision boundary

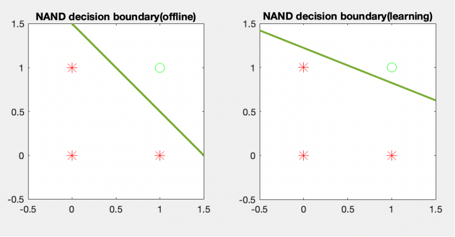


Figure 9: NAND decision boundary

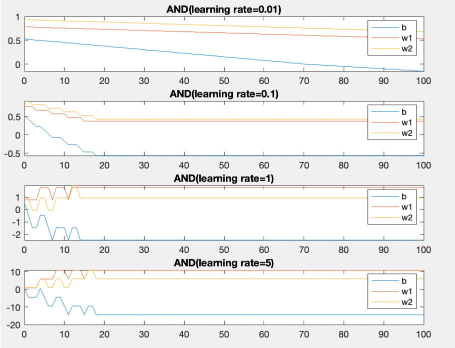


Figure 10: AND different learning rates

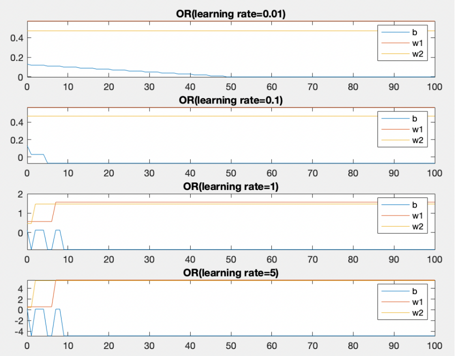


Figure 11: OR different learning rates

When different learning rates, including 0.01, 0.1, 1 and 5, are applied to the learning procedure, the weights trajectories show differently.(Shown from Figure 10 to Figure 13). Generally, when learning rate is high, weights will get convergence at a relatively high speed, for learning rate equals to 0.01 situation, weights did not stay stable after 100 epochs. Additionally, under the circumstance of high learning rate, the final results of each weight have a larger absolute value, this situation results from high learning rate causing larger change of weight for each epoch, the weight will stay stable at a higher absolute value.

c).

When the same experiment is implemented on the EXCLUSIVE OR function. The trajectory of weights is shown Figure 14 and Figure 15. They show that whatever the learning rate(0.01, 0.1, 1, 5) is, XOR weights trajectory will not get convergence. Trajectories keep fluctuating at each epoch. Higher learning rate situation fluctuates with a higher amplitude. The reason is that XOR function is not linear separable, no possible solution to separate XOR function with only one line. Wrong

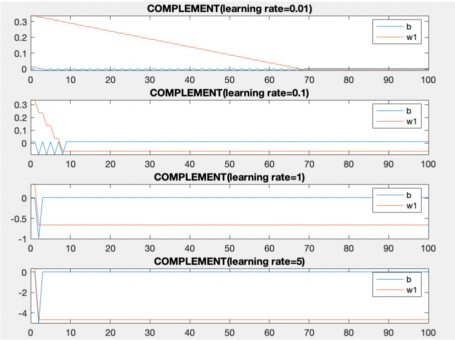


Figure 12: COMPLEMENT different learning rates

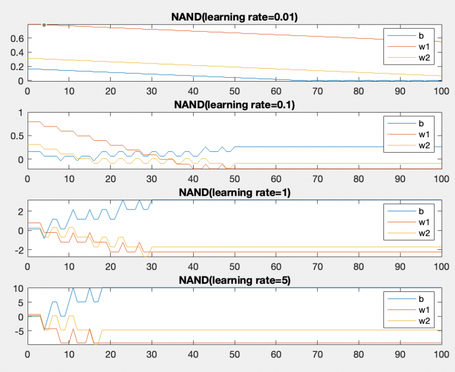


Figure 13: NAND different learning rates

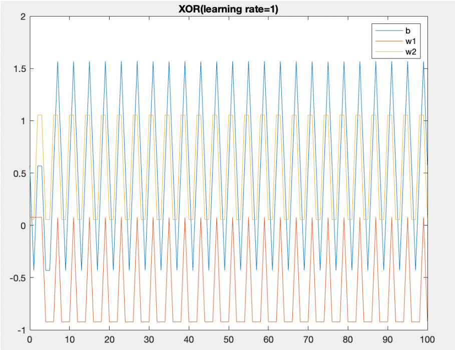


Figure 14: XOR weights(learning rate=1)

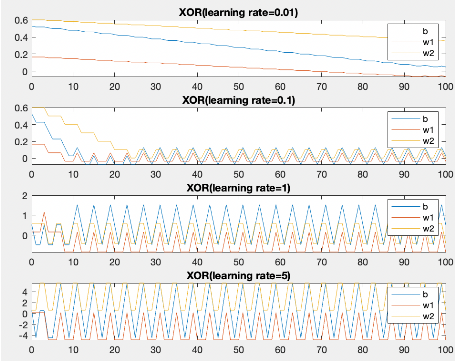


Figure 15: XOR different learning rates

classification will always happen during epochs.

Q4.

a).

According to the LLS method formula in textbook, the value of w and b can be calculated, the result is shown in Table 2.The regression result is drawn in Figure 16. The line fits dots very well.

b).

Figure 17 shows the result of LMS with randomly chosen weight and learning rate of 0.01. The line also fits the dots very well. Figure 18 presents the trajectory of w and b during epochs. Weights get convergence.

c).

The results of LLS and LMS are shown in Table 2 as mentioned in subquestion a). LMS learning procedure obtains similar weights with LLS calculation. From Figure 16 and Figure 17, two fitting lines are also similar.

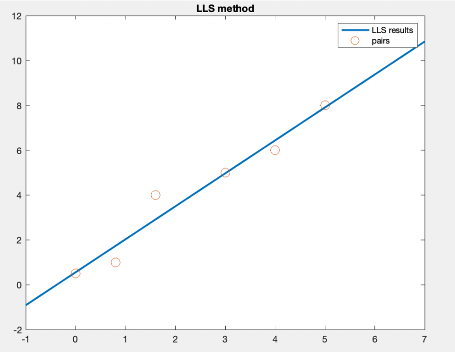


Figure 16: LLS result

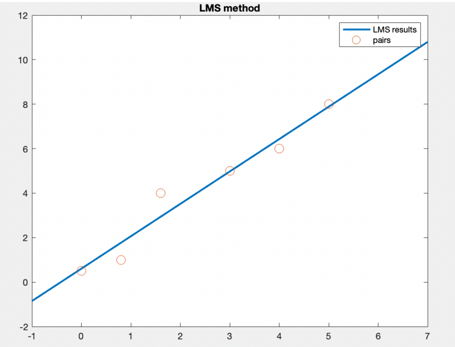


Figure 17: LMS result

	b	w
LLS	0.5554	1.4700
LMS	0.6502	1.4433

Table 2: Comparison of off-line calculation and learning procedure

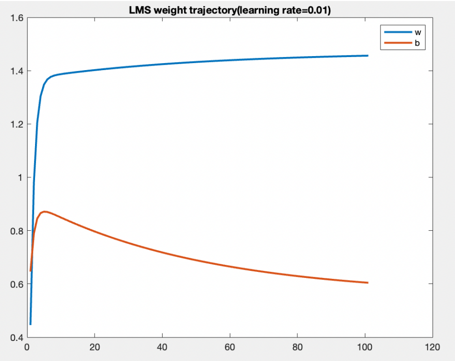


Figure 18: LMS weight(learning rate=0.01)

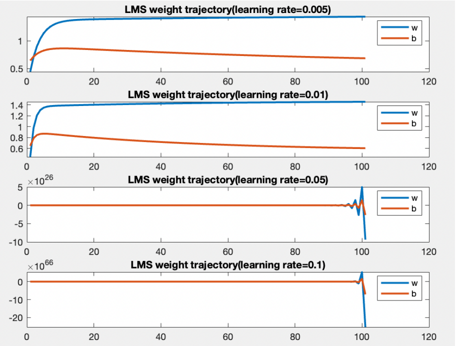


Figure 19: LMS different learning rates

d).

When different learning rate is applied, it shows different results for low learning rate and high learning rate. From Figure 19, for low learning rate, weights initially keep changing and then converging. But for high learning rate, weights remain unchanged initially and fluctuate with a high amplitude at last epochs and cannot get the correct answers. High learning rate is not suitable for LMS questions.

Q5.

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

Since

$$Y = w^T X$$

$$e = d - Y$$

$$e = d - w^T X$$

Assuming J$(w) = J_1(w) + J_2(w)$, where J$_1(w) = \frac{1}{2}\sum_{i=1}^{n} r^2(i)e(i)^2$ and $J_2(w) = \frac{1}{2}\lambda||w||^2$.
For $J_1(w)$, using diagonal matrix $R = diag(r^2(1), r^2(2), \cdots r^2(n))$ to present,get $J_1(w) = \frac{1}{2}e^T we$
Using chain rule,

$$\frac{\partial J_1(w)}{\partial w} = \frac{\partial J_1(w)}{\partial e}\frac{\partial e}{\partial w}$$

It is also known that $\mathbf{e} = d - \mathbf{w}^T X$,So

$$\frac{\partial J_1(w)}{\partial w} = \frac{\partial(\frac{1}{2}e^T\mathbf{Re})}{\partial e}\frac{\partial(d - w^T X)}{\partial w} = -X^T Re$$

$$\frac{\partial J_2(w)}{\partial w} = \lambda w$$

Let $\frac{\partial J_1(w)}{\partial w} = 0$

$$-X^T R(d - Xw) + \lambda w = 0$$

Solving this equation

$$(X^T RX + \lambda\mathbf{I})w = X^T Rd$$

$$w = (X^T RX + \lambda\mathbf{I})^{-1}X^T Rd$$

($\mathbf{I}$ is the unit diagonal matrix)