

CS550 - Machine Learning and Business Intelligence

Designing XOR Gate

By Soe Wunna (19651)

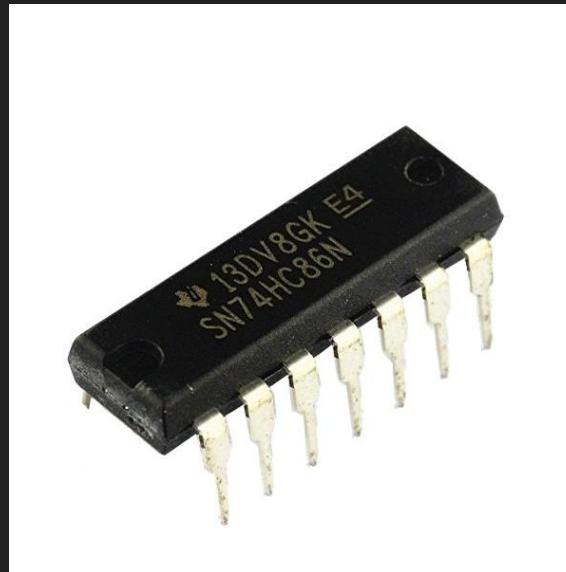
Instructor: Dr. Chang, Henry

Table of Content

1. Introduction
2. XOR Gate Truth Table
3. Calculation
4. Conclusion
5. References

1. Introduction

An XOR gate (Exclusive OR gate) is a digital logic gate that takes two binary inputs and produces a single binary output. The output is "1" if the inputs are different (i.e., one input is "1" and the other input is "0"), and "0" if the inputs are the same (i.e., both inputs are either "1" or "0").



2. XOR Gate Truth Table

The truth table for an XOR gate is as follows:

Input A	Input B	Output
0	0	0
0	1	1
1	0	1
1	1	0

XOR gates are widely used in digital electronics and computer science for a variety of purposes, including error detection and correction, encryption and decryption, and in the design of adders, subtractors, and other arithmetic circuits.

3. Calculation

The forward/backward process

- Forward process
Calculate the output Z for the given input (X,Y) .
- Backward process
Adjust weights
 - + If the output Z is too low, increase the weights by 0.5 which had inputs that were "1".
 - + If the output Z is too high, decrease the weights by 0.5 which had inputs that were "1".

Using step activation function

```
Z := ( W0 * C + W1 * X + W2 * Y >= T )
where T := 1.0
```

```
if ( W0 * C + W1 * X + W2 * Y >= T )
then ouput is 1
else output = 0
```

3. Calculation

Loop 1:

$$W1 = W2 = 1.0$$

X Y | Z1

0	0		0	Z ok
0	1		1	Z ok
1	0		1	Z ok
1	1		1	Z ok

Desired Function for “OR”:

X Y | Z

0	0		0
0	1		1
1	0		0
1	1		1

W1 = 1.0, W2 = 1.0 for “OR” Gate

“Nand” Gate Formula:

$$Z := (W0 * C + W1 * X + W2 * Y >= T)$$

where $T := 1.0$

3. Calculation

Desired Function for “NAND”:

C X Y | Z2

1 0 0 | 1
1 0 1 | 1
1 1 0 | 1
1 1 1 | 0

Loop 1:

W0 = 0.0

W1 = W2 = 1.0

C X Y | Z2

1 0 0 | 0 Z too low, increase W0 by 0.5, W0=0.5
1 0 1 | 1 Z ok
1 1 0 | 1 Z ok
1 1 1 | 1 Z too high, decrease all by 0.5, W0=0.0, W1=0.5, W2=0.5

3. Calculation

Loop 2:

W0 = 0.0

W1 = W2 = 0.5

C X Y | Z2

1 0 0 | 0 Z too low, increase W0 by 0.5, W0=0.5

1 0 1 | 0 Z too low, increase W0, W2 by 0.5, W0=1.0, W2=1.0

1 1 0 | 0 Z too low, increase W0, W1 by 0.5, W0=1.5, W1=1.0

1 1 1 | 1 Z too high, decrease all by 0.5, W0=1.0, W1=0.5, W2=0.5

Loop 3:

W0 = 1.0, W1 = W2 = 0.5

C X Y | Z2

1 0 0 | 1 Z ok

1 0 1 | 1 Z ok

1 1 0 | 1 Z ok

1 1 1 | 1 Z too high, decrease all by 0.5, W0=0.5, W1=0.0, W2=0.0

3. Calculation

Loop 4:

W0 = 0.5, W1 = W2 = 0.0

C X Y | Z2

1 0 0 | 0 Z too low, increase W0 by 0.5, W0=1.0
1 0 1 | 0 Z too low, increase W0, W2 by 0.5, W0=1.5, W2=0.5
1 1 0 | 0 Z too low, increase W0, W1 by 0.5, W0=2.0, W1=0.5
1 1 1 | 0 Z ok

Loop 5:

W0 = 0.5, W1 = W2 = 0.0

C X Y | Z2

1 0 0 | 1 Z ok
1 0 1 | 1 Z ok
1 1 0 | 1 Z ok
1 1 1 | 1 Z too high, decrease all by 0.5, W0=1.5, W1=0.0, W2=0.0

3. Calculation

Loop 6:

W0 = 1.5, W1 = W2 = 0.0

C X Y | Z2

1 0 0 | 1 Z ok
1 0 1 | 1 Z ok
1 1 0 | 1 Z ok
1 1 1 | 1 Z too high, decrease all by 0.5, W1=1.0, W2=-0.5, W3=-0.5

Loop 7:

W0 = 1.0, W1 = W2 = -0.5

C X Y | Z2

1 0 0 | 1 Z ok
1 0 1 | 1 Z too low, increase W0, W2 by 0.5, W0=1.5, W2=0.0
1 1 0 | 1 Z too low, increase W0, W2 by 0.5, W0=1.5, W2=0.0
1 1 1 | 1 Z ok

3. Calculation

Loop 8:

$W_0 = 2.0, W_1 = W_2 = 0.0$

C X Y | Z2

1 0 0 | 1 Z ok
1 0 1 | 1 Z ok
1 1 0 | 1 Z ok
1 1 1 | 1 Z too high, decrease all by 0.5, $W_1=1.5, W_2=-0.5, W_3=-0.5$

Loop 9:

$W_0 = 1.5, W_1 = W_2 = -0.5$

C X Y | Z2

1 0 0 | 1 Z ok
1 0 1 | 1 z ok
1 1 0 | 1 Z ok
1 1 1 | 0 Z ok

W0 = 1.5, W1 = -0.5, W2 = -0.5 for “NAND” Gate

3. Calculation

“AND” Gate Formula:

$$Z := (W1 * X + W2 * Y \geq T)$$

where $T := 1.0$

Desired Function for “AND”:

X	Y		Z3

0	0		0
0	1		0
1	0		0
1	1		1

Loop 1:

$$W1 = W2 = 1.0$$

X	Y		Z3

0	0		0	Z ok
0	1		1	Z too high, decrease W2 by 0.5, W2=0.5
1	0		1	Z too high, decrease W1 by 0.5, W1=0.5
1	1		1	Z ok

3. Calculation

Loop 2:

$$W1 = W2 = 0.5$$

X Y | Z3

0	0		0	Z ok
0	1		1	Z ok
1	0		1	Z ok
1	1		1	Z ok

W1 = 0.5, W2 = 0.5 for “AND” Gate

3. Calculation

“XOR” Gate formula (plugging W0, W1, and W2s from previous gates):

Z1 := X "Or" Y

Z2 := X "NAND" Y

Z := Z3 := Z1 "AND" Z2

Z := (X "Or" Y) "AND" (X "NAND" Y)

Z := (1.0 * X + 1.0 * Y >= 1.0) "AND" (1.5 * 1.0 + -0.5 * X + -0.5 * Y >= 1.0)

Z := (0.5 * (1.0 * X + 1.0 * Y >= 1.0) + 0.5 * (1.5 + -0.5 * X + -0.5 * Y >= 1.0) >= 1.0)

Desired Function for “XOR”:

X Y | Z3

0 0 | 0

0 1 | 1

1 0 | 1

1 1 | 0

3. Calculation

When X=0, Y=0:

$$\begin{aligned} Z3 &= (0.5 * (1.0 * 0 + 1.0 * 0 \geq 1.0) + 0.5 * (1.5 + -0.5 * 0 + -0.5 * 0 \geq 1.0) \geq 1.0) \\ &= (0.5 * (0.0 \geq 1.0) + 0.5 * (1.5 \geq 1.0) \geq 1.0) \\ &= (0.5 * (\text{false}) + 0.5 * (\text{true}) \geq 1.0) \\ &= 0.5 * 0 + 0.5 * 1 \geq 1.0 \\ &= 0.5 \geq 1.0 \\ &= \text{false} \\ &= 0 \rightarrow \text{same as desired} \end{aligned}$$

When X=0, Y=1:

$$\begin{aligned} Z3 &= (0.5 * (1.0 * 0 + 1.0 * 1 \geq 1.0) + 0.5 * (1.5 + -0.5 * 0 + -0.5 * 1 \geq 1.0) \geq 1.0) \\ &= (0.5 * (1.0 \geq 1.0) + 0.5 * (1.0 \geq 1.0) \geq 1.0) \\ &= (0.5 * (\text{true}) + 0.5 * (\text{true}) \geq 1.0) \\ &= 0.5 * 1 + 0.5 * 1 \geq 1.0 = 1.0 \geq 1.0 \\ &= \text{true} \\ &= 1 \rightarrow \text{same as desired} \end{aligned}$$

3. Calculation

When X=1, Y=0:

$$\begin{aligned} Z3 &= (0.5 * (1.0 * 1 + 1.0 * 0 \geq 1.0) + 0.5 * (1.5 + -0.5 * 1 + -0.5 * 0 \geq 1.0) \geq 1.0) \\ &= (0.5 * (1.0 \geq 1.0) + 0.5 * (1.0 \geq 1.0) \geq 1.0) \\ &= (0.5 * (\text{true}) + 0.5 * (\text{true}) \geq 1.0) \\ &= 0.5 * 1 + 0.5 * 1 \geq 1.0 = 1.0 \geq 1.0 \\ &= \text{true} \\ &= 1 \rightarrow \text{same as desired} \end{aligned}$$

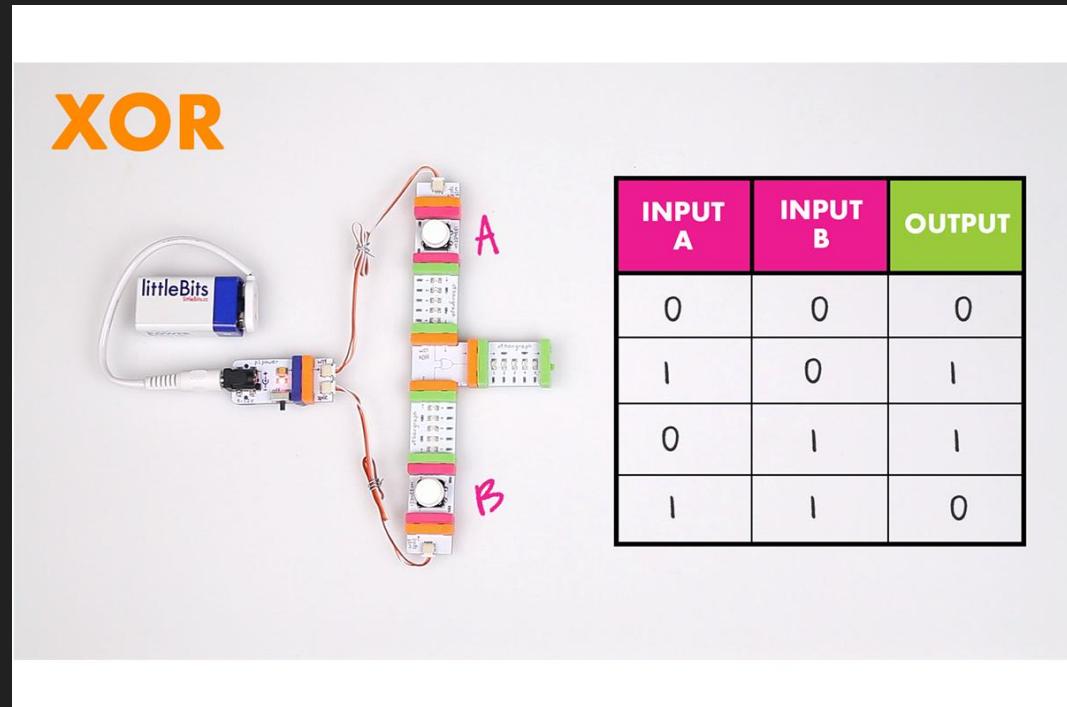
When X=1, Y=1:

$$\begin{aligned} Z3 &= (0.5 * (1.0 * 1 + 1.0 * 1 \geq 1.0) + 0.5 * (1.5 + -0.5 * 1 + -0.5 * 1 \geq 1.0) \geq 1.0) \\ &= (0.5 * (2.0 \geq 1.0) + 0.5 * (0.5 \geq 1.0) \geq 1.0) \\ &= (0.5 * (\text{true}) + 0.5 * (\text{false}) \geq 1.0) \\ &= 0.5 * 1 + 0.5 * 0 \geq 1.0 = 0.5 \geq 1.0 \\ &= \text{false} \\ &= 0 \rightarrow \text{same as desired} \end{aligned}$$

All Z3 same as desired in “XOR” Gate!

4. Conclusion

In conclusion, XOR gate is a fundamental digital logic gate that performs exclusive or operation on two binary inputs and produces a single binary output. It is widely used in various applications of digital electronics and computer science, such as error detection and correction, encryption and decryption, and in the design of arithmetic circuits. The truth table of XOR gate shows that its output is "1" only when its inputs are different, making it a crucial component in the design of many electronic devices and circuits.



5. References

1. https://hc.labnet.sfbu.edu/~henry/sfbu/course/machine_learning/deep_learning/slides/exercise_deep_learning.html
2. https://www.electronics-tutorials.ws/logic/logic_7.html
3. <https://www.analog.com/en/design-center/glossary/xor-gate.html>