DESIGN XOR GATE

BY ADELEYE ADEBODUN 19747

Question

Design your own AND Gate, OR Gate, and NAND Gate

Note
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 output is 1
else output = 0

-
- The bias C for NAND is 1.0

OR Gate
Desired output

X	Y	Z
0	0	0
0	1	1
1	0	1
1	1	1

W1= W2 = 0

X	Y	Z
0	0	0
0	1	0
1	0	0
1	1	0

W1= W2 = 0.5

X	Y	Z
0	0	0
0	1	0
1	0	0
1	1	1

W1= W2 = 1.0

X	Y	Z
0	0	0
0	1	1
1	0	1
1	1	1

If the weight is increased by 0.5

The values for (0,1) and (1,0) will be 0

And the weight is doubled to 1.0

```
Z1 = (1.0 * X + 1.0 * Y >= 1.0)
Z1 := (Y > -X + 1.0)
```

AND Gate

Desired result

X	Y	Z1
0	0	0
0	1	0
1	0	0
1	1	1

W1= W2 = 0

X	Y	Z
0	0	0
0	1	0
1	0	0
1	1	0

$W1 = W2 = 0.5$

X	Y	Z1
0	0	0
0	1	0
1	0	0
1	1	1

Z = (0.5 * X + 0.5 * Y >= 1.0)

Z := (Y > -X + 2.0)

Design NAND Gate

X	Y	Z2
0	0	1
0	1	1
1	0	1
1	1	0

Using the formula from AND gate

NOT AND = NAND

Z2 = (0.5 * X + 0.5 * Y <= 1.0)

Z2:= (-0.5 * X + -0.5 * Y >= -1.0)

Z2:= (1.5 * 1 + -0.5 * X + -0.5 * Y >= +1.0)

Z2 := (+1.5 * 1.0 + -0.5 * X + -0.5 * Y >= +1.0)

==> Z2 := (Y <= -0.5 * X + 2.0)

```
      +------------------------------------------------------------+
      |    "NAnd"  +---------+                 "XOr" Function  |
      | 1.0-->1.5--->|         |                                   |
      |           |     |                               |
X --->|----> -0.5-->| Neuron |-->Z2--                              |
      |\          |     |       \                         |
Y---->|-\--> -0.5-->|         |     \ "And"  +---------+     |
      |\ \        +---------+       ->0.5-->|         |     |
      | \ \                               | Neuron |-->Z3-->|-->Z
      |  \ \  "Or"  +---------+     -->0.5-->|         |     |
      |   \ -->1.0-->|         |     /          +---------+        |
      |     \     | Neuron |-->Z1--                           |
      |     -->1.0-->|         |                                  |
      |           +---------+                                |
      +------------------------------------------------------------+
```

The blackbox for the XOR function

Formula for Z

Z1: = X “Or"   Y

Z2: = X “NAND" Y

Z: = Z3: = Z1 "AND" Z2

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

Z: = (1 * X + 1 * Y >= 1.0) "AND"
(1.5 + -0.5 * X + -0.5 * Y >= 1.0)

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

Prove that the XOR Gate works for the following

- X=1, Y=1
- X=1, Y=0
- X=0, Y=1
- X=0, Y=0

For X=1, Y=1

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

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

Z := (0.5 * (2.0 >= 1.0) +
0.5 * (0.5 >= 1.0) >= 1.0)

Z := (0.5 * (true) +
0.5 * (false) >= 1.0)

Z := (0.5 * 1 + 0.5 * 0 >= 1.0)

Z := (0.5 + 0.0 >= 1.0)

Z := (false)

Z := 0

For X=1, Y=0

Z := (0.5 * (1.0 * 1.0 + 1.0 * 0 >= 1.0) +

0.5 * (1.5 + -0.5 * 1.0 + -0.5 * 0 >= 1.0) >= 1.0)

Z := (0.5 * (1.0 + 0 >= 1.0) +

0.5 * (1.5 + -0.5 + -0 >= 1.0) >= 1.0)

Z := (0.5 * (1.0 >= 1.0) +

0.5 * (1.0 >= 1.0) >= 1.0)

Z := (0.5 * (true) +

0.5 * (true) >= 1.0)

Z := (0.5 * 1 + 0.5 * 1 >= 1.0)

Z := (0.5 + 0.5 >= 1.0)

Z := (true)

Z := 1

For X=0, Y=1

Z := (0.5 * (1.0 * 0 + 1.0 * 1.0 >= 1.0) +

0.5 * (1.5 + -0.5 * 0 + -0.5 * 1.0 >= 1.0) >= 1.0)

Z := (0.5 * (0 + 1.0 >= 1.0) +

0.5 * (1.5 + -0 + -0.5 >= 1.0) >= 1.0)

Z := (0.5 * (1.0 >= 1.0) +

0.5 * (1.0 >= 1.0) >= 1.0)

Z := (0.5 * (true) +

0.5 * (true) >= 1.0)

Z := (0.5 * 1 + 0.5 * 1 >= 1.0)

Z := (0.5 + 0.5 >= 1.0)

Z := (true)

Z := 1

For X=0, Y=0

Z := (0.5 * (1.0 * 0 + 1.0 * 0 >= 1.0) +
0.5 * (1.5 + -0.5 * 0 + -0.5 * 0 >= 1.0) >= 1.0)

Z := (0.5 * (0 + 0 >= 1.0) +
0.5 * (1.5 + -0 + -0 >= 1.0) >= 1.0)

Z := (0.5 * (0 >= 1.0) +
0.5 * (1.5 >= 1.0) >= 1.0)

Z := (0.5 * (false) +
0.5 * (true) >= 1.0)

Z := (0.5 * 0 + 0.5 * 1 >= 1.0)

Z := (0 + 0.5 >= 1.0)

Z := (false)

Z := 0