Monday, September 11, 2023 10:11 PM

Q1

Designing a two layer Neural Network to implement the given logic gate

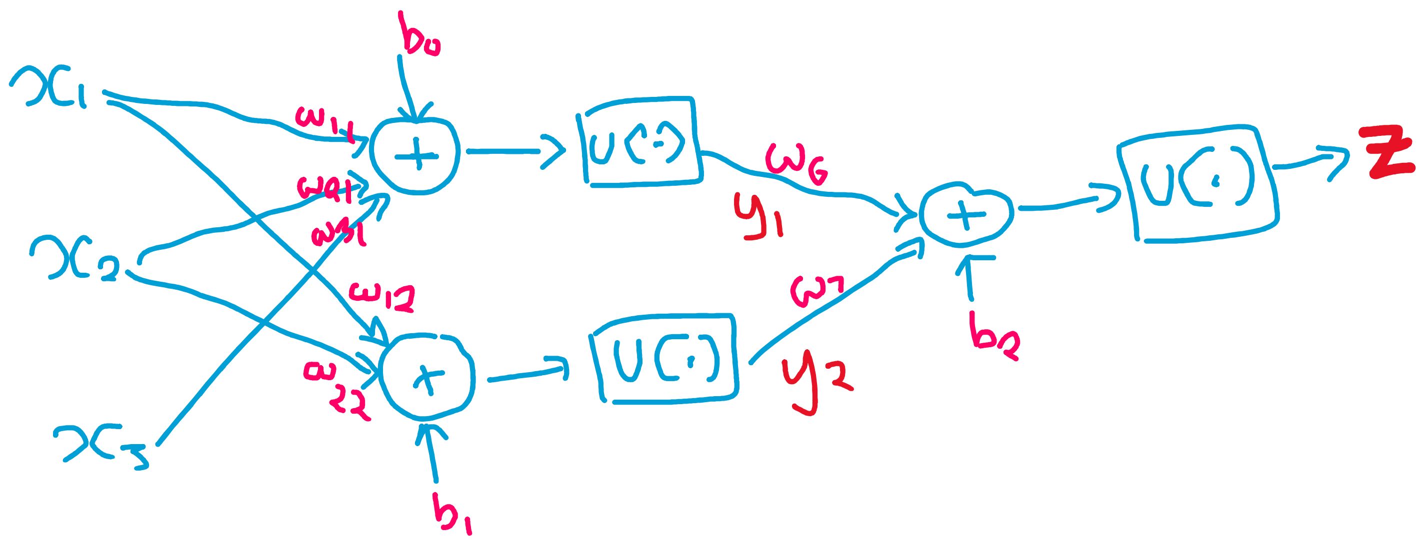
f(x1,x2,x3) = (not x1)x2x3 + x1(not x2)

Or

f(x1,x2,x3) = ((NOT x1) AND x2 AND x3) OR (x1 AND (NOT x2))

Since there are two terms separated by an OR we can design the network to be a 3-2-1 network, that is 3 source nodes (inputs, here x1,x2 and x3) and 2 neurons in the first hidden layer and 1 neuron in the output layer.

Since the first term has all 3 terms, the first neuron will take all 3 as input. Similarly, the second neuron will have only 2 terms as its input. The output layer neuron has the output of these two neurons as its input. Each neuron has its own bias term and uses the signum activation function.



Thus, we obtain the following equations

y1 = sgn(b0 + w11x1 + w21x2 + w31x3) -- eq1

y2 = sgn(b1 + w12x1 + w22x2) -- eq2

z = sgn(b2 + w6y1 + w7y2) -- eq3

Solving for variables in eq1 by substituting values for x1, x2, x3 and desired output y1 = (not x1)x2x3

for x1 = 1, x2 = 1, x3 = 1 => y1 = -1

-1 = sgn(b0 + w11*1 + w21*1 + w31*1)

=> b0 + w11 + w21 + w31 < 0

Solving similarly for all x1,x2,x3 values gives us.

b0 - w11 - w21 - w31 < 0 (x1 = -1, x2 = -1, x3 = -1)
b0 - w11 - w21 + w31 < 0 (x1 = -1, x2 = -1, x3 = 1)
b0 - w11 + w21 - w31 < 0 (x1 = -1, x2 = 1, x3 = -1)
b0 - w11 + w21 + w31 > 0 (x1 = -1, x2 = 1, x3 = 1)
b0 + w11 - w21 - w31 < 0 (x1 = 1, x2 = -1, x3 = -1)
b0 + w11 + w21 - w31 < 0 (x1 = 1, x2 = 1, x3 = -1)
b0 + w11 + w21 + w31 < 0 (x1 = 1, x2 = 1, x3 = 1)
b0 + w11 - w21 + w31 < 0 (x1 = 1, x2 = -1, x3 = 1)

And the truth table for the same would be

x1	x2	x3	y1	y2
-1	-1	-1	-1	-1
-1	-1	1	-1	-1
-1	1	-1	-1	-1
-1	1	1	1	-1
1	-1	-1	-1	1
1	1	-1	-1	-1
1	1	1	-1	-1
1	-1	1	-1	1

Following the truth table and the set of inequalities, we can solve for the values of b0,w11,w21,w31

Adding the first and 7th inequality is adding two negative numbers, which gives us

2b0 < 0 => b0 < 0

Using value substitution/trial, one possible set of weights and bias for neuron 1 is:

b0 = -3, w11 = -2, w21 = 0.75, w31 = 0.75

Similarly for y2, using the eq2 by substituting values for x1, x2 and desired output y2 = (not x2)x1, we get the following inequalities

b1 - w12 - w22 < 0
b1 - w12 + w22 < 0
b1 + w12 - w22 > 0
b1 + w12 + w22 < 0 (even though both y1 and y2 cannot be true at the same time because of x2 in y1 and not x2 in y2)

adding first and the last inequality gives us
b1 + b1 - w12 + w12 - w22 + w22 < 0
=> 2b1 < 0 => b1 < 0

Using the inequalities and truth table along with value substitution/trial we get the following possible biases and weights for neuron 2

b1 = -1, w12 = 1, w22 = -1

Similarly for neuron 3, building a truth table and the deriving set of inequalities

y1	y2	z
-1	-1	-1
-1	1	1
1	-1	1
1	1	1

z = sgn(b2 + w6y1 + w7y2) -- eq3

for y1 = -1 and y2 = -1, z = -1 as both are false and thus false or false = false

-1 = sgn(b2 - w6 - w7)
=> b2 - w6 - w7 < 0

Similarly for all y1,y2 values

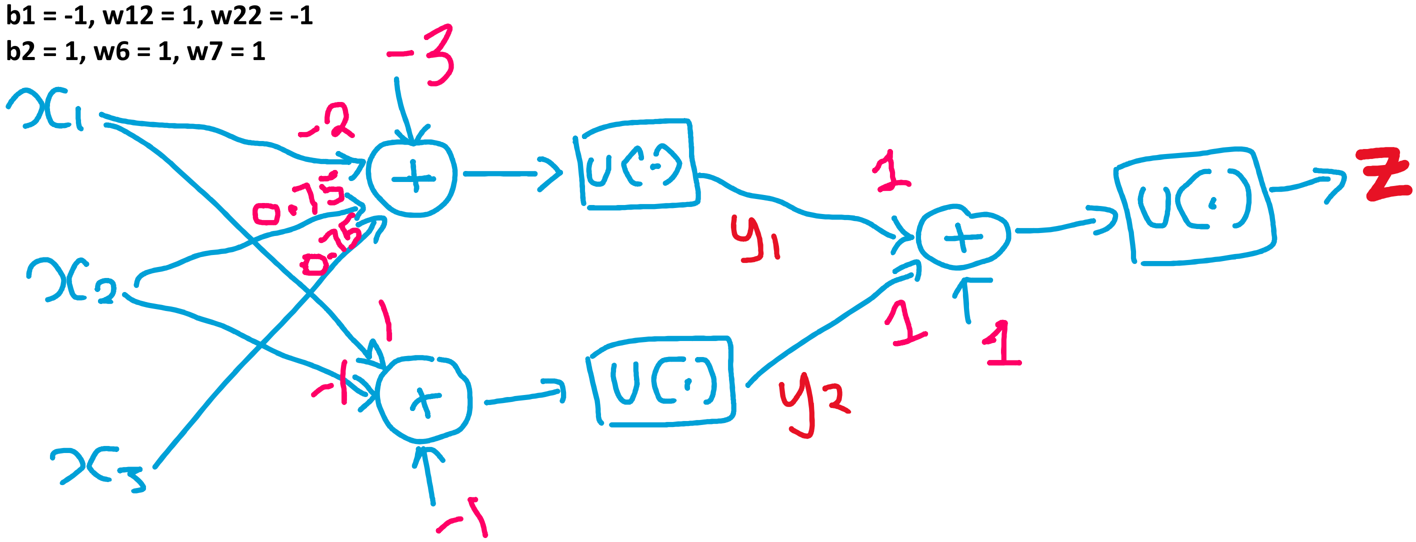
b2 - w6 - w7 < 0
b2 - w6 + w7 > 0
b2 + w6 - w7 > 0
b2 + w6 + w7 > 0

Using the inequalities and truth table along with value substitution/trial we get the following possible biases and weights for neuron 3

b2 = 1, w6 = 1, w7 = 1

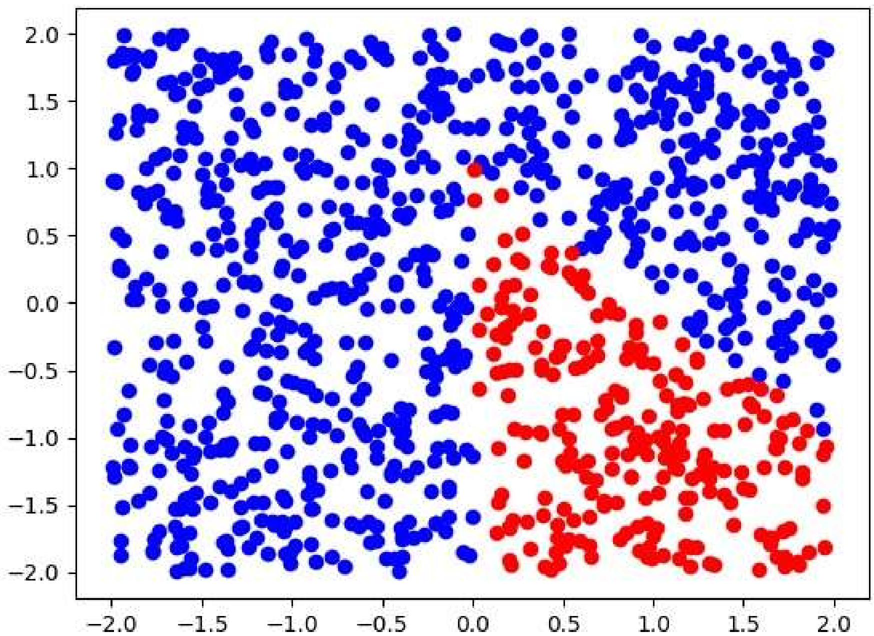
That gives us the following neuron.

b0 = -3, w11 = -2, w21 = 0.75, w31 = 0.75
b1 = -1, w12 = 1, w22 = -1
b2 = 1, w6 = 1, w7 = 1



Q2

The code included in the code folder on the box folder, the plot and decision boundary explained below



Estimate of the decision region:

The decision region is comprised of these two equations.
Combined they form the decision boundary, if the points meet the criteria they are 1 (red) else 0.

We can clearly see in the plot that there are no red points if x < 0
When x > 0 we see a downward slope line starting at (0,1) till (2,-1)

Using the two points we can get the equation of the line

slope = (2-0)/(-1-1) = -1

y = mx + b => y = -1*x + b

substituting the point (0,1) in this gives us

1 = -0 * b
=> b = 1

thus the equation of the line is

y = -x + 1

If we see the output of the neural network on point (1,0) will tell us if the points on the line are part of z = 1 or not

z = step(-1.5 + step(1+x-y) + step(1-x-y) - step(-x))
=> step(-1.5 + step(1+1-0) + step(1-1-0) - step(-1))
=> step(-1.5 + 1 + 1 - 0)
=> step(0.5)
=> 1

thus the points on y = -x + 1 are part of z = 1

Similarly for (0,1) to see if x = 0 is part of the z = 1

z = step(-1.5 + step(1+x-y) + step(1-x-y) - step(-x))
=> step(-1.5 + step(1+0-1) + step(1-0-1) - step(-0))
=> step(-1.5 + 1 + 1 - 1)
=> step(-0.5)
=> 0

Therefore the decision region is represented by the following

z = 1 if x > 0 and y <= -x+1; else z = 0