


National University of Computer and Emerging Sciences, Lahore Campus

	Course:	Artificial Intelligence	Course Code:	AI2002
	Program:	BS(CS)	Semester:	Spring-24
	Exam Type:	Assignment 4	Total Marks:	100
	Due Date:		Weight	TBD
	Section:	Sec-B	Page(s):	2

Instructions:

- Do the work by yourself, this is an individual assignment.
- Plagiarism cases will be dealt with strictly.
- Submit a single python file.

Question 4:

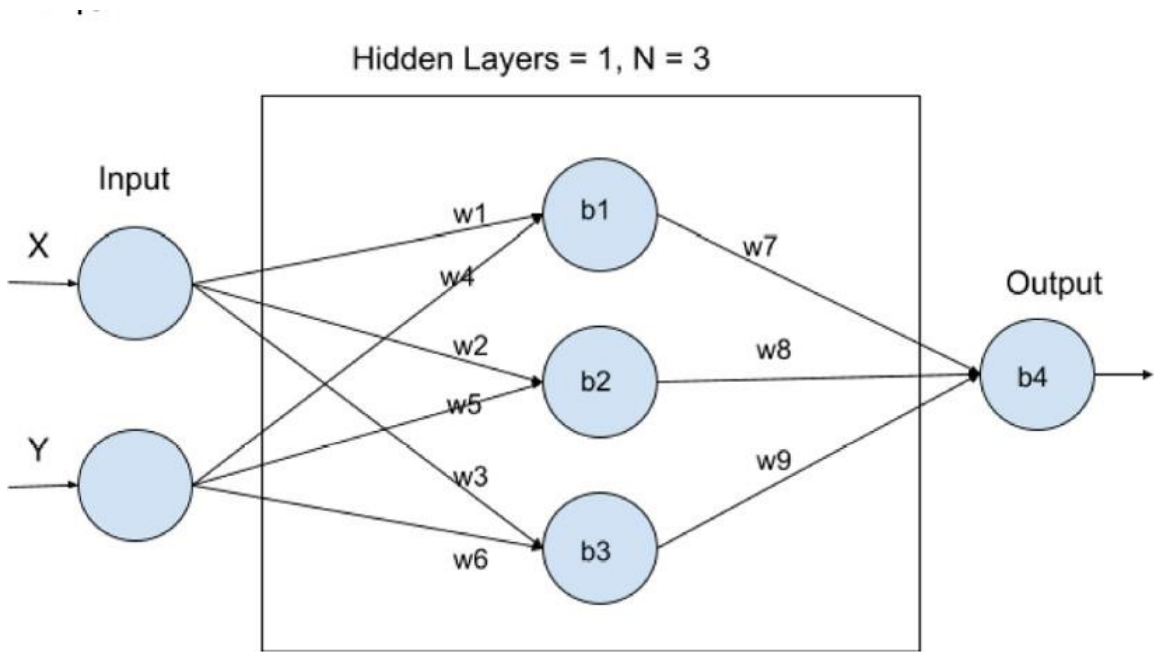
Implement a Customizable Neural Network Training with Backpropagation for XOR. Requirements are as follows:

1. Input size will be 2.
2. Output size will be 1.
3. Initialize the weights randomly.
4. Number of hidden layers and N (number nodes in a hidden layer) will be parameters and will be given by the user during run time your program should be generic to work for any input sequence of the hidden layers' parameters. A user can enter different no of neurons for two different layers.
5. Activation Function will be Sigmoid.
6. Implement the feedforward process to compute the output of each layer.
7. Implement the backpropagation algorithm to update weights and biases.
8. Use a learning rate (eta) for updating weights and biases.
9. Use the XOR problem dataset.
10. Allow user input for the following parameters:
 - Number of hidden layers.
 - Number of nodes in each hidden layer.
 - Number of iterations for training.
11. Print the output, weights, and biases after each iteration during training.

Use this neural network to predict the output of XOR. The following table shows the correct output of XOR:

X	Y	Output
0	0	0
0	1	1
1	0	1
1	1	0

Sample Example when hidden layers =1 and N=3:



Hint:

Weights and biases for the above example can be stored in matrices as:

$W1 =$

w1	w2	w3
w4	w5	w6

$B1 =$

b1
b2
b3

$W2 =$

w7
w8
w9

$B2 =$

b4
