CSE 340 –MACHINE LEARNING 1

LAB ASSIGNMENT 1

17/02/2024

Unique id :E0321037 Sarah Madhavan K

QUESTION :

A neural network consists of layers. The first layer is the input and the last layer is the output. The layers between the input and output are called hidden layers. The nodes of one layer can be connected to the nodes of the next layer. Each connection between two nodes (called an edge) has a weight.

Write a java code which does the following :

It takes the number of layers, and nodes in each layer as input, and creates a class object.
It sets each edge's weight from user input
It should be possible to query the weight of any edge, given the node.

Your program should follow the principles of object oriented program

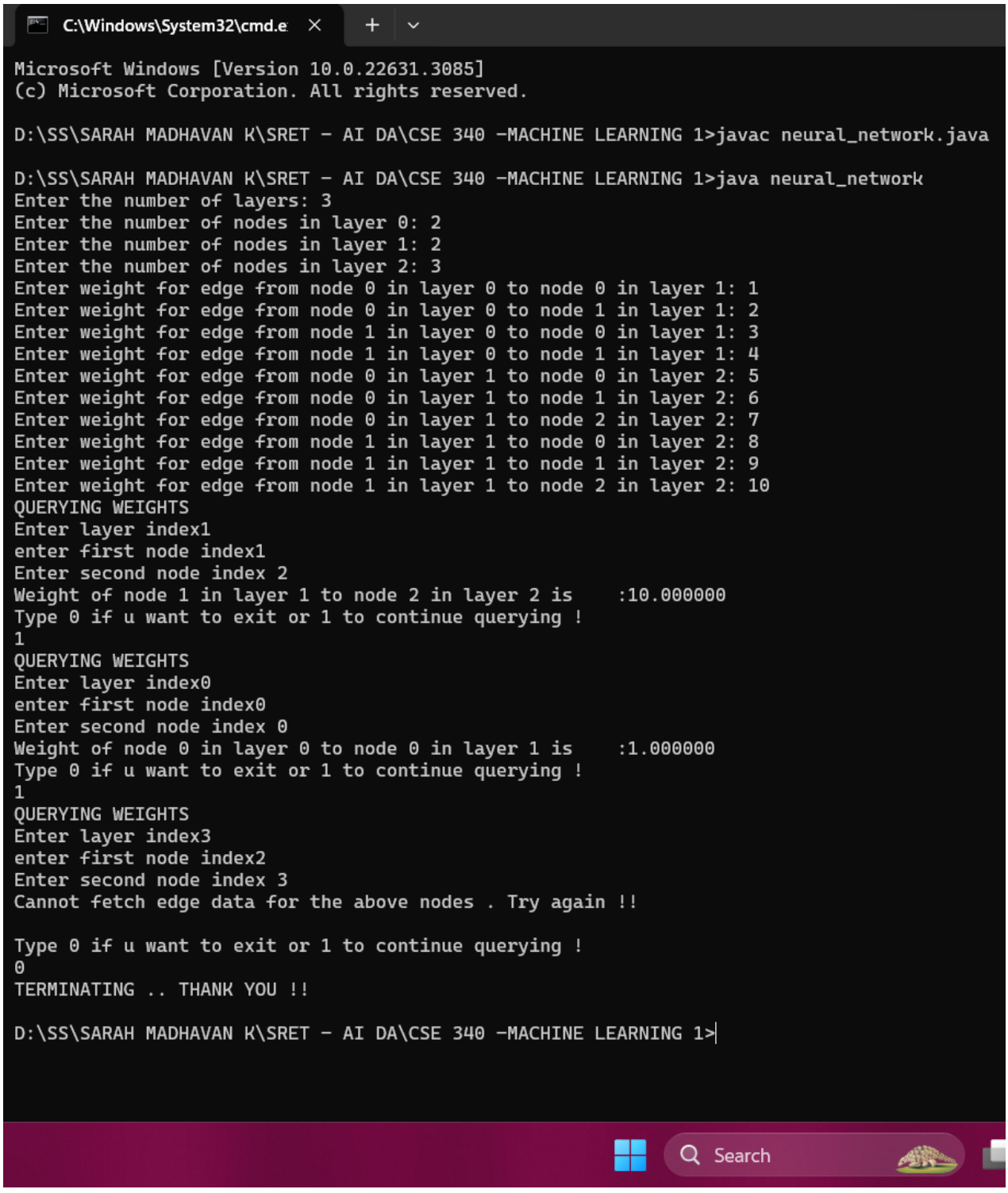
CODE :

```
import java.util.Scanner;


public class neural_network {


    private int no_of_layers;

    private int[] nodes_of_layer;

    private double[][][] edge_weights;


    public neural_network(int no_of_layers, int[] nodes_of_layer) {

        this.no_of_layers = no_of_layers;

        this.nodes_of_layer = nodes_of_layer;


        // Initialize edge_weights with random values
```

```
        edge_weights = new double[no_of_layers - 1][][];


    }


    public void setedge_weights() {
        Scanner scanner = new Scanner(System.in);
        for (int i = 0; i < no_of_layers - 1; i++) {
            edge_weights[i] = new double[nodes_of_layer[i]][nodes_of_layer[i + 1]];
            for (int j = 0; j < nodes_of_layer[i]; j++) {
                for (int k = 0; k < nodes_of_layer[i + 1]; k++) {
                    System.out.printf("Enter weight for edge from node %d in layer %d to node %d in layer
%d: ", j, i, k, i + 1);
                    edge_weights[i][j][k] = scanner.nextDouble();
                }
            }
        }
    }


    public void query(neural_network nn,Scanner scanner){
         System.out.println("QUERYING WEIGHTS ");


        System.out.print("Enter layer index");
        int layer_no = scanner.nextInt();
        System.out.print("enter first node index");
        int node1_no = scanner.nextInt();
        System.out.print("Enter second node index ");
        int node2_no = scanner.nextInt();
        try{
        double weight = nn.getedge_weights(layer_no, node1_no, node2_no);
        System.out.printf("Weight of node %d in layer %d to node %d in layer %d is    :%f
",node1_no,layer_no,node2_no,layer_no+1,weight);}
        catch(Exception e){
            System.out.println("Cannot fetch edge data for the above nodes . Try again !!");
```

```
    }
  }
  public double getedge_weights(int layer_no, int node1_no, int node2_no) {
    return edge_weights[layer_no][node1_no][node2_no];
  }


  public static void main(String[] args) {
    Scanner scanner = new Scanner(System.in);
    System.out.print("Enter the number of layers: ");
    int no_of_layers = scanner.nextInt();
    int[] nodes_of_layer = new int[no_of_layers];
    for (int i = 0; i < no_of_layers; i++) {
      System.out.printf("Enter the number of nodes in layer %d: ", i);
      nodes_of_layer[i] = scanner.nextInt();
    }

    neural_network nn = new neural_network(no_of_layers, nodes_of_layer);
    nn.setedge_weights();

    int do_u_want_to_query_again=1;
    while (do_u_want_to_query_again==1){
      nn.query(nn,scanner);
          System.out.println();
      System.out.println("Type 0 if u want to exit or 1 to continue querying !");
      do_u_want_to_query_again=scanner.nextInt();
    }
    {System.out.println("TERMINATING .. THANK YOU !! ");}


  }
}
```

OUTPUT :


C:\Windows\System32\cmd.e
Microsoft Windows [Version 10.0.22631.3085]
(c) Microsoft Corporation. All rights reserved.
D:\SS\SARAH MADHAVAN K\SRET - AI DA\CSE 340 -MACHINE LEARNING 1>javac neural_network.java
D:\SS\SARAH MADHAVAN K\SRET - AI DA\CSE 340 -MACHINE LEARNING 1>java neural_network
Enter the number of layers: 3
Enter the number of nodes in layer 0: 2
Enter the number of nodes in layer 1: 2
Enter the number of nodes in layer 2: 3
Enter weight for edge from node 0 in layer 0 to node 0 in layer 1: 1
Enter weight for edge from node 0 in layer 0 to node 1 in layer 1: 2
Enter weight for edge from node 1 in layer 0 to node 0 in layer 1: 3
Enter weight for edge from node 1 in layer 0 to node 1 in layer 1: 4
Enter weight for edge from node 0 in layer 1 to node 0 in layer 2: 5
Enter weight for edge from node 0 in layer 1 to node 1 in layer 2: 6
Enter weight for edge from node 0 in layer 1 to node 2 in layer 2: 7
Enter weight for edge from node 1 in layer 1 to node 0 in layer 2: 8
Enter weight for edge from node 1 in layer 1 to node 1 in layer 2: 9
Enter weight for edge from node 1 in layer 1 to node 2 in layer 2: 10
QUERYING WEIGHTS
Enter layer index1
enter first node index1
Enter second node index 2
Weight of node 1 in layer 1 to node 2 in layer 2 is :10.000000
Type 0 if u want to exit or 1 to continue querying !
1
QUERYING WEIGHTS
Enter layer index0
enter first node index0
Enter second node index 0
Weight of node 0 in layer 0 to node 0 in layer 1 is :1.000000
Type 0 if u want to exit or 1 to continue querying !
1
QUERYING WEIGHTS
Enter layer index3
enter first node index2
Enter second node index 3
Cannot fetch edge data for the above nodes . Try again !!
Type 0 if u want to exit or 1 to continue querying !
0
TERMINATING .. THANK YOU !!
D:\SS\SARAH MADHAVAN K\SRET - AI DA\CSE 340 -MACHINE LEARNING 1>
Search