PROBLEM 3 (10pts) NEURAL NETWORKS

Part 1 (7pts)

1. Give a short general description what is a Neural Network

2. Give a short general description how Neural Network learns

3. Given a classification data **D** with **attributes a1, a2, ... an** and classes **c1, c2, .. ck**

Which is the number of INPUT nodes of any NN for **D**?

Which is the number of OUTPUT nodes of any NN for full classification for **D**?

Which is he number of hidden layers?

Which is the number of nodes in the hidden layers?

4. Given the following

CLASSIFICATION DATA **D**

a1	a2	C
1	0	c1
0	1	c2
1	1	c1
1	0	c2
0	0	c3
0	0	c3

Design 3 NEURAL NETWORKS for **D**

One for full classification

Two for contrast learning (for your chosen classes)

Draw pictures and **explain** correctness of your topology

FOR SOLUTION USE your additional page

Part 2 (3pts)

1. Give a general description of the following STEPS of the **Backpropagation Algorithm**

Step 1: initialize

Step 2: feed

Step 3: propagate

Step 4: backpropagate

Step 5: backpropagate

Step 6: repeat

Step 7; terminate when

PART - 2

Question - 3

PART - 1

(1) Neural Network is basically a set of nodes which are connected and mimics the functionality of a human brain. There nodes start with the input nodes, internal nodes & finally with the output nodes. The nodes which are at present at the same depth from the input nodes form layers. So there are generally three layers Input, hidden & output. The inputs are the attributes of the data tuple which are fed simultaneously through the input nodes. These layers are connected with each other through nodes & these connections are called edges & each edge is given a weight that gets adjusted as the learning proceeds. There is also a non-linear / activation function which takes the weighted summation

output from a layer & non-linearizes it before sending it to the next layer. As a Neural Network tends to mimic a human brain, it's also called as a Perceptron and a Neural Network with multiple layers is called as the Multi-Layered Perceptron. It is a supervised learning.

(2) Neural Network learning is referred as "Connectionist Learning" as the learning parameters are the weights of the edges or connections between the layers. So basically a Neural Network tend to adjust it's weights in order to correctly classify the input through the output layer. This in general a time taking process & possible where we can compromise on the learning phase. They also require few parameters to move through the learning phase like learning rate, momentum e.t.c. these parameters together called as the Network Topology. But these algorithms have high tolerance to noisy & incomplete data.

(3) Given the number of attributes as $a_1, a_2, \ldots a_n$ & classes $C_1, C_2, \ldots C_k$ for D.

(a) Number of input nodes of any NN for D:

This would be same for all Neural Networks & it is the number of attributes i.e n.

(b) Number of output nodes of any NN for full classification:

As it was given for full classification, the number of output nodes would the number of classes i.e k.

(c) Number of Hidden Layers:

This totally depends on the implementation & can't be determined by the number of attributes & classes but atleast need to be 1.

(d) Number of nodes in the hidden layers:

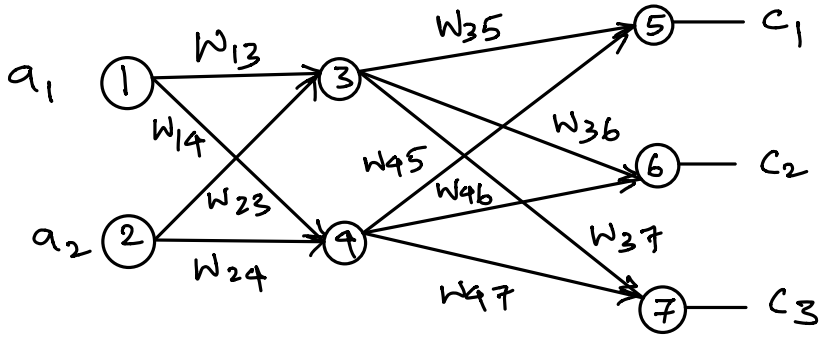
This also totally depends on the implementation & can't be determined by the number of attributes & classes but atleast need to be 1.

(4) Given classification data D.

a_1	a_2	C
1	0	c_1
0	1	c_2
1	1	c_1
1	0	c_2
0	0	c_3
0	0	c_3

We are asked to design 3 neural networks.

(1) Full classification.

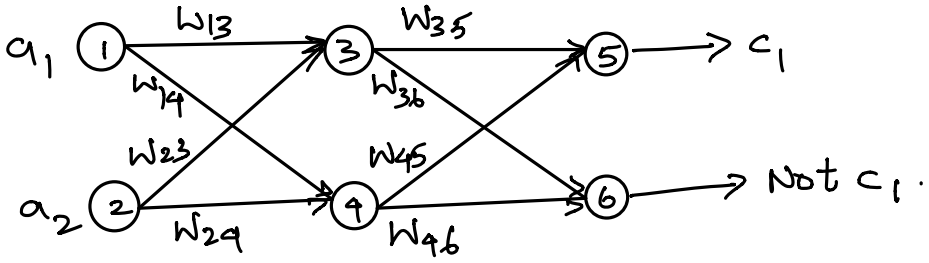


We have designed a Neural Network with 1 hidden layer containing 2 hidden nodes.

Since there are two attributes, there are 2 units (1,2) in the input layer. there are 3 classes to be classified, so there are 3 (5,6,7)

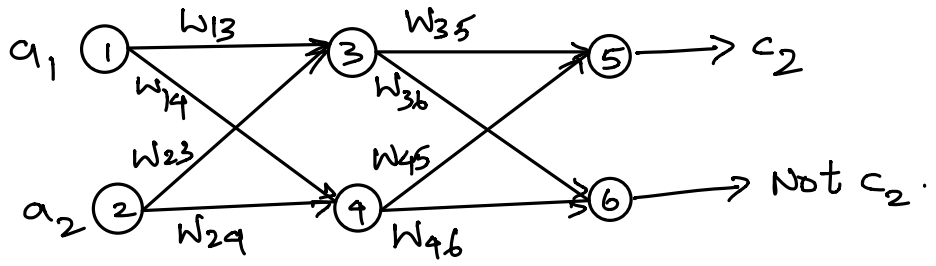
units in the output layer. And, we require atleast 1 hidden layer, we had one with 2 hidden nodes (3, 4). Hence the topology is valid.

(2) Contrast classification w.r.t c_1 :



We again used a hidden layer with 2 internal nodes (3,4) as there atleast need to be 1. As there are 2 attributes, there are 2 input nodes (1,2) & since there are only 2 classes to classify as this being contrast classification (c_1 & not c_1). We decided to go with 2 output nodes.

(3) Contrast classification w.r.t C_2:



We again used a hidden layer with 2 internal nodes (3,4) as there atleast need to be 1. As there are 2 attributes, there are 2 input nodes (1,2) & since there are only 2 classes to classify as this being contrast classifications (C_2 & not C_2). We decided to go with 2 output nodes.

PART-2

PROBLEM-3

PART-2

(1) General description of the following steps of the Backpropagation algorithm.

Step 1 : Initialize :

The weights of all the edges present in the Neural Network are initialized to some small random numbers & also each node is also initialized with a bias. These bias values are also initialized to some small random numbers.

Step 2 : Feed :

Here the training tuple is fed into the input layer where each input node gets the value of the corresponding attribute of the input training tuple.

Step 3 : Propogate :

After feeding the training tuple to the input layer, we need to propogate the inputs forward. The inputs to any unit/node are based on the outputs of the previous layer. It's basically an output of a non-linear activation function applied on the weighted sum of outputs along with the bias of the current node/unit. So basically each unit in the hidden & output layers takes in the net input & applies an activation function on it. Sigmoid or Logistic functions are the most commonly used activation functions.

Step 4 : Back propagate :

In this step, we back propagate the error. After propogating the inputs till the output layer we calculate the error w.r.t to the true output & update the weights & biases by propogating the error backwards.

Step 5 : Back propagate :

This is part of the above back propagation step where we are updating the weights & biases. This back propagation method learns using gradient descent to search for the optimum weights & biases. Our main agenda behind this updation is to decrease the error between the class prediction & assigned true class values. This gradient descent occurs based on a learning rate & this updation of weights & biases is called as case updation. There's another method where these changes are accumulated until all the training data is feeded into the Neural Network. This is called epoch updating, where one iteration through the training data is called an epoch.

Step 6 : Repeat :

We repeat the steps 1-5 again till we terminate to reach a global minimum. Many of the times this process might lead to a local minimum. so the learning rate, number of epochs are an important criteria for learning.

step 7 : Terminate :

We terminate this learning process in three conditions.

(1) When the updation value of the weights in the previous epoch were below some threshold

(2) No. of misclassified tuples are below some threshold.

(3) The prespecified number of epochs are completed.