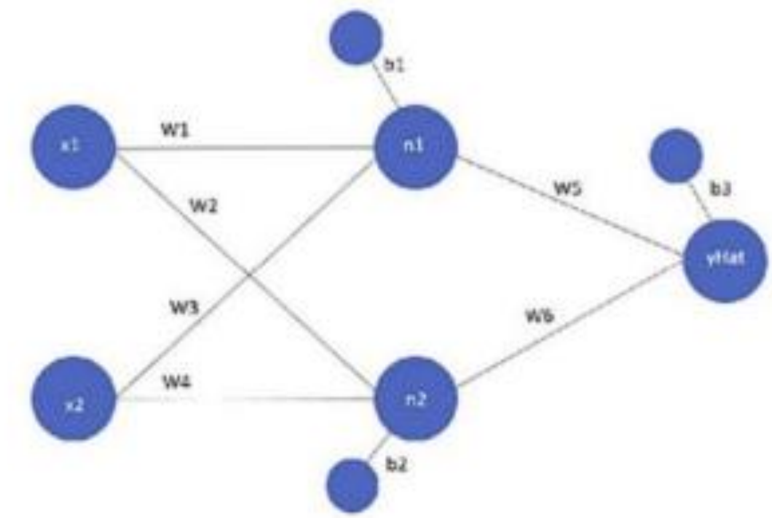
AI 2002

Artificial Intelligence

Assignment No 07

Question 1:

Consider the neural network architecture for XAND function given below:



1. Assume that the activation function is the sigmoid function. Initialize all the weights with 0.1. Write down the values of the weights after the first and the second iteration of Backpropagation algorithm run with the following examples:

X1	**X2**	**Y**
0	0	1
1	0	0
0	1	0
1	1	1

2. Consider a neural network that uses a function tanh instead of the sigmoid function. What will be the values of the weights after the first and second iteration in such a case?

Question 2:

1. Derive a gradient descent based learning rule for a perceptron that uses tanh as activation function. Assume that the perceptron has D inputs and a bias term and we have N training examples (Xi, Yi) I = 1, 2, ... , N to learn weights of the perceptron. Please note that the derivative of tanh(x) is 1 – tanh2(x)

2. Show the mathematical working of Artificial Neural Network by taking the case in figure below. First two columns are the input values for X1 and X2 and the third column is the desired output. Show only 3 iterations.

   Learning rate = 0.2

   Threshold = 0.5

   Actual output = W1X1+W2X2

   Next weight adjustment = Wn+ΔWn

   Change in weight = ΔWn = learning rate * (desired output- actual output) * Xn

   Show two complete iterations for acquiring the desired output.

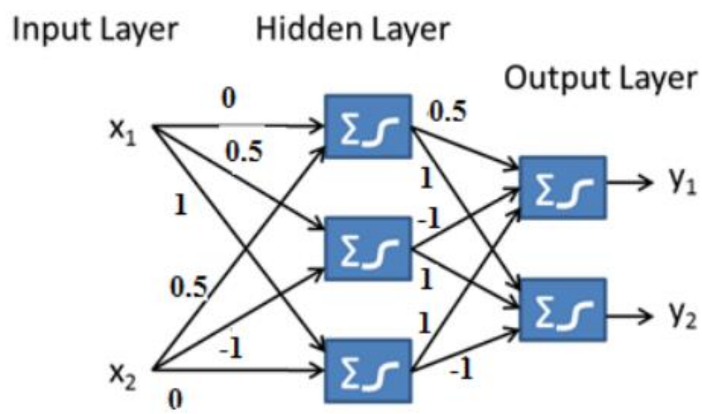
X1	X2	W1	W2	D	Y	ΔW1	ΔW2

X1	**X2**	**Output**
0	0	0
0	1	1
1	0	0
1	1	1

Question 3:

Consider the neural network architecture with two inputs (x1, x2) three hidden layer neurons and two output neurons (Y1, Y2). Assume that each neuron has a bias term set equal to -1 and each neuron uses the sigmoid activation function given as

$$f(x) = \frac{1}{1+e^{-x}}$$



Weights of all neurons as shown on the figure are:

Hidden Neurons

Top Neuron	**0**	**0.5**
Middle Neuron	**0.5**	**-1**
Bottom Neuron	**1**	**0**

Output Neurons

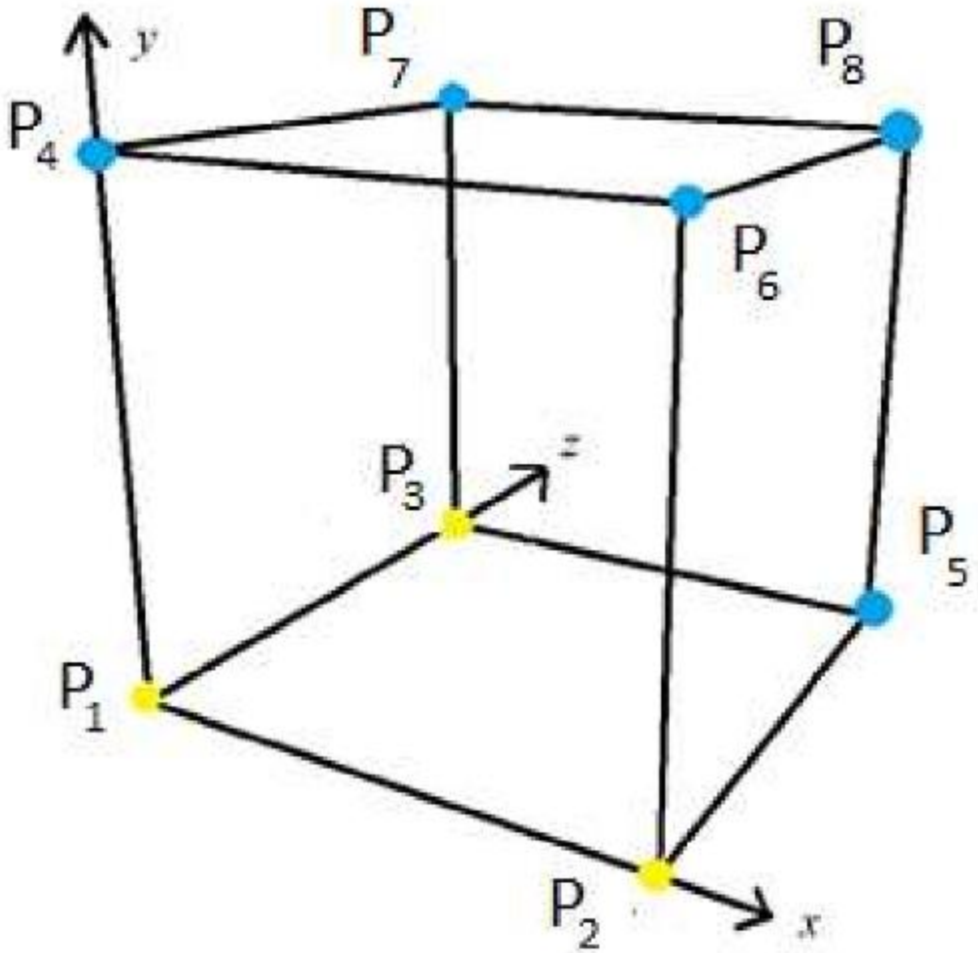
Top Neuron	**0.5**	**-1**	**1**
Bottom Neuron	**1**	**1**	**-1**

For the training example with input X = [1, 1] and Target Y = [0, 1]

1. **Compute Output**
   Compute the neural network output where $\alpha = 1$
2. **Error Calculation and Back Propagation**
   - For each of the output neurons, compute the error term δk used by the back propagation algorithm to update the weights.
   - Compute the error term δh for each of the hidde n neurons.
3. **Weight Update**
   Use the errors computed above to compute the updated weights of the neural network. Take the value of learning rate $\eta = 0.5$.

Question 4:

You are given a 2-class, 3-dimensional data set. The data is plotted in the graph below. Yellow and blue dots represent negative class (-1) and positive class (1) respectively. Basically, the samples/points represent the corners of a unit cube. The point P1 is at the centre of the coordinate plane i.e. (0, 0, 0).



Using a single Perceptron, learn a linear model for this data that correctly classifies all the samples/points. You must start with the initial weights [-2 2 3 2] and use the points in order i.e. P1 then P2 ... P8 during the learning process. Further, the first weight i.e. -2 is that of the bias term the bias term is always 1 and learning rate is also 1. Show all the working that you did to calculate the values of the weights. The activation function g(x) is given below:

$$g(x) = \begin{cases} 1, & x \geq 0 \\ -1, & x < 0 \end{cases}$$

Question 5:

Consider a single perceptron unit with a bias term and n inputs **[x0, x1, ... xn].** The perceptron uses weights **[w0, w1, ... wn]** and the given sigmoid activation function to compute the output.

$$\text{(i.e. } f(x) = \frac{1}{1+e^{-x}} \text{)}$$

Derive the weight update equation for such a perceptron using the following definition of

$$\textbf{Error} = \sum_{j}^{m} (y_j - y_j^*)^2 + \sum_{i=0}^{n} w_i^2$$

where yj is the target output and yj* is the output computed using weights and the activation function of neuron.

Hints:

- Derivative of sigmoid activation function can be written as

$$f'(x) = f(x) \,.\, (1 - f(x))$$

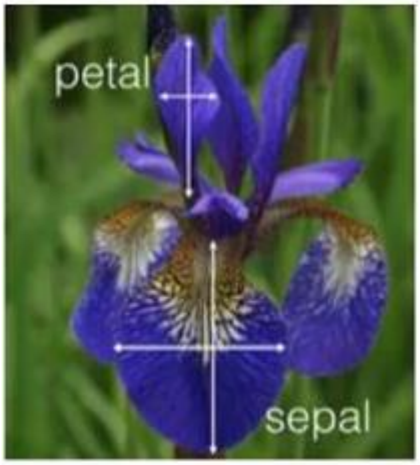
- Weights are updated using the equation

$$\boldsymbol{W_i = W_i - \eta \Delta E_i}$$

Therefore, to derive the final equation you need to compute the derivative of Error with respect to each of the weights components wi and then use the weight update equation given above to derive the final equation.

Question 6:

Iris dataset has 50 samples for each of three different species of Iris flower (total number of samples is 150). For each data sample, you have sepal length, sepal width, petal length and petal width and a species name (class/label). Figure below shows Iris flower and features in dataset:




```
5.1,3.8,1.9,0.4,Iris-setosa
4.8,3.0,1.4,0.3,Iris-setosa
5.1,3.8,1.6,0.2,Iris-setosa
4.6,3.2,1.4,0.2,Iris-setosa
5.3,3.7,1.5,0.2,Iris-setosa
5.0,3.3,1.4,0.2,Iris-setosa
7.0,3.2,4.7,1.4,Iris-versicolor
6.4,3.2,4.5,1.5,Iris-versicolor
6.9,3.1,4.9,1.5,Iris-versicolor
5.5,2.3,4.0,1.3,Iris-versicolor
6.5,2.8,4.6,1.5,Iris-versicolor
5.7.2.8.4.5.1.3.Iris-versicolor
```

Perform the following tasks:

- Load the given Iris dataset
- Split it into a training set and a test set
- Preprocess the data
- Build a Decision Tree classifier
- Then train a Random Forest classifier
- How much better does it perform compared to the Decision Tree classifier?