

## Assignment: Perceptron Training

PRAJAKTA DEOKULE

Batch: A1

Roll no: 3330

```
import java.util.ArrayList;  
import java.util.Scanner;
```

```
public class PerceptronTraining {  
  
    public static void main(String args[]) {  
  
        int and[] = {0,0,0,1};  
        int or[] = {0,1,1,1};  
  
        int arr[][] = {{0,0},{0,1},{1,0},{1,1}};  
  
        Scanner sc = new Scanner(System.in);  
  
        System.out.println("Enter weight 1:");  
        double weight1 = sc.nextDouble();  
        sc.nextLine();  
  
        System.out.println("Enter weight 2:");  
        double weight2 = sc.nextDouble();  
        sc.nextLine();  
  
        System.out.println("Enter threshold value:");  
        double threshold = sc.nextDouble();  
        sc.nextLine();  
  
        System.out.println("Enter learning value:");  
        double learningRate = sc.nextDouble();  
        sc.nextLine();  
  
        int choice;  
        do  
        {  
            System.out.println("*****MENU*****");  
            System.out.println("1.AND Gate");  
            System.out.println("2.OR Gate");  
            System.out.println("3.Exit");  
  
            choice = sc.nextInt();  
  
            switch(choice) {  
  
                case 1: solve(weight1, weight2, threshold, learningRate, and, arr);  
                        break;  
  
                case 2: solve(weight1, weight2, threshold, learningRate, or, arr);  
                        break;  
  
                case 3: System.out.println("Exited!");  

```

```

        break;

        default: System.out.println("Incorrect choice");
        break;
    }

    }while(choice!=3);

}

```

```

public static void solve(double w1,double w2,double n,double alpha,int operatorD[],int arr[][]){

    int count=0;
    int desiredOP=0;
    int actualOP=0;
    double f=0;
    System.out.println("w1="+w1);
    System.out.println("w2="+w2);

    while(count<4){

        desiredOP=operatorD[count];

        System.out.print("Input x1 : "+arr[count][0]+" Input x2 : "+arr[count][1]);
        f=w1*arr[count][0]+w2*arr[count][1];
        if(f<n){
            actualOP=0;
        }
        else {
            actualOP=1;
        }
        System.out.print(" f(x): "+f+" Output : "+actualOP +" Target : "+desiredOP+"\n");

        if(desiredOP!=actualOP){
            double w1new=w1+ (alpha*(desiredOP-actualOP)*arr[count][0]);
            double w2new=w2+(alpha*(desiredOP-actualOP)*arr[count][1]);
            w1=w1new;
            w2=w2new;
            f=0;
            System.out.println("w1new:"+w1new);
            System.out.println("w2new:"+w2new);
            count=0;
        }

        else
            count++;

    } //close outer while loop
}
}
/* Enter weight 1:
1.2
Enter weight 2:
0.6
Enter threshold value:

```

```

1
Enter learning value:
0.5
*****MENU*****
1.AND Gate
2.OR Gate
3.Exit
1
w1=1.2
w2=0.6
Input x1 : 0 Input x2 :0   f(x): 0.0 Output :0 Target :0
Input x1 : 0 Input x2 :1   f(x): 0.6 Output :0 Target :0
Input x1 : 1 Input x2 :0   f(x): 1.2 Output :1 Target :0
w1new:0.7
w2new:0.6
Input x1 : 0 Input x2 :0   f(x): 0.0 Output :0 Target :0
Input x1 : 0 Input x2 :1   f(x): 0.6 Output :0 Target :0
Input x1 : 1 Input x2 :0   f(x): 0.7 Output :0 Target :0
Input x1 : 1 Input x2 :1   f(x): 1.299999999999998 Output :1 Target :1
*****MENU*****
1.AND Gate
2.OR Gate
3.Exit
3
Exited!
Enter weight 1:
0.6
Enter weight 2:
0.6
Enter threshold value:
1
Enter learning value:
0.5
*****MENU*****
1.AND Gate
2.OR Gate
3.Exit
2
w1=0.6
w2=0.6
Input x1 : 0 Input x2 :0   f(x): 0.0 Output :0 Target :0
Input x1 : 0 Input x2 :1   f(x): 0.6 Output :0 Target :1
w1new:0.6
w2new:1.1
Input x1 : 0 Input x2 :0   f(x): 0.0 Output :0 Target :0
Input x1 : 0 Input x2 :1   f(x): 1.1 Output :1 Target :1
Input x1 : 1 Input x2 :0   f(x): 0.6 Output :0 Target :1
w1new:1.1
w2new:1.1
Input x1 : 0 Input x2 :0   f(x): 0.0 Output :0 Target :0
Input x1 : 0 Input x2 :1   f(x): 1.1 Output :1 Target :1
Input x1 : 1 Input x2 :0   f(x): 1.1 Output :1 Target :1
Input x1 : 1 Input x2 :1   f(x): 2.2 Output :1 Target :1
*****MENU*****
1.AND Gate
2.OR Gate
3.Exit
3
Exited!

```