ARTIFICIAL INTELLIGENCE

Name :- Shreyash Tekade PRN No. :- 12110840 Roll No. : 71

Problem Statement : - Implement a single perceptron

Code :-

```
# include <bits/stdc++.h>
using namespace std;

int signum(double value){
   return value > 0 ? 1 : -1;
}

double calculateNet(vector<double> w, vector<double> x){
   double sum = 0;
   for(int i=0;i<w.size();i++){
      sum += w[i] * x[i];
      // cout << "The sum is ::";
      // cout << sum << endl;
   }
   // cout << "The net is ::" << sum << endl;

   return sum;

}

int main(){

   double c = 1;
   int size = 0, dimension = 0;
```

```
// cout << "Enter the number of inputs::";
cin >> size;
// cout << "Enter the dimesion of the data::";
cin >> dimension;
// vector<vector<int>> v = take_input(size, dimension);
vector<vector<double>> v(size, vector<double>(dimension, 0));
double data;
for(int i=0;i<size;i++){
   for(int j=0;j<dimension;j++){
      cin >> data;
      v[i][j] = data;
   }
}

for(int i=0;i<size;i++){
   for(int j=0;j<dimension;j++){
      cout << v[i][j] << " ";
   }
   cout << endl;
}

// cout << "Enter the initial w_i::";
vector<double> w_i(dimension, 0);
for(int i=0;i<dimension;i++){
   cin >> data;
   w_i[i] = data;
}

vector<double> y(size, 0);

// cout << "Enter the actual labels::";
```

```
double ans;
for(int i=0;i<size;i++){
   cin >> ans;
   y[i] = ans;
}

int epoch = 1000;
double error = DBL_MAX;

for(auto x : y){
   cout << x << " ";
}



vector<double> _error;

int cycle = 0;
while(error != 0){
   double e = 0;
   double row = 0;
   for(int j=0;j<size;j++){
      vector<double> input(dimension, 0);
      for(int j=0;j<dimension;j++){
         input[j] = v[row][j];
      }
      cout << endl;
      cout << "This is the input::";
      for(auto x : input){
         cout << x << " ";
      }
      double net = calculateNet(w_i, input);
```

```
int o = signum(net);
cout << "o is ::" << o <<  endl;
cout << "y is ::" << y[row] << endl;
e += (y[row] - o);

cout << "e is " << e << endl;

vector <double> w(dimension, 0);

double constant = c * (y[row] - o);
for(int k=0;k<dimension;k++){
   w[k] = constant * input[k];
   cout << w[k];
}
vector<double> _w_i(dimension, 0);
cout << "This is w_i::";
for(int p=0;p<dimension;p++){
   _w_i[p] = w[p] + w_i[p];

   cout << _w_i[p] << " ";
}

cout << "This is the constant::" << constant << endl;
cout << "This is w_i::";
for(auto x : _w_i){
   cout << x << " ";
}
w_i = _w_i;
row ++;
```

```
        }
        cycle ++;
        cout << endl;
        cout << "-------------------------------------------------"<< endl;

        error = e;
        _error.push_back(error);
        cout << error << endl;
        cout << "-------------------------------------------------"<< endl;


    }

    cout << "The error is " << error << endl;

    for(auto x : w_i){
        cout << x << " ";
    }

    cout << endl;

    for(auto x : _error){
        cout << x << endl;
    }



    return 0;
}
```

Output:

```
D:\AI\Lab4>a < input.txt
1 -2 1.5 0
1 -0.5 -2 -1.5
0 1 -1 1.5
1 -1 1
This is the input::1 -2 1.5 0 o is ::1
y is ::1
e is 0
0-000This is w_i::1 -1 0 0.5 This is the constant::0
This is w_i::1 -1 0 0.5
This is the input::1 -0.5 -2 -1.5 o is ::1
y is ::-1
e is -2
-2143This is w_i::-1 0 4 3.5 This is the constant::-2
This is w_i::-1 0 4 3.5
This is the input::0 1 -1 1.5 o is ::1
y is ::1
e is -2
00-00This is w_i::-1 0 4 3.5 This is the constant::0
This is w_i::-1 0 4 3.5
------------------------------------------------
-2
------------------------------------------------

This is the input::1 -2 1.5 0 o is ::1
y is ::1
e is 0
0-000This is w_i::-1 0 4 3.5 This is the constant::0
This is w_i::-1 0 4 3.5
This is the input::1 -0.5 -2 -1.5 o is ::-1
y is ::-1
e is 0
0-0-0-0This is w_i::-1 0 4 3.5 This is the constant::0
This is w_i::-1 0 4 3.5
This is the input::0 1 -1 1.5 o is ::1
y is ::1
e is 0
00-00This is w_i::-1 0 4 3.5 This is the constant::0
This is w_i::-1 0 4 3.5
------------------------------------------------
0
------------------------------------------------
The error is 0
-1 0 4 3.5
-2
```

Graph :-

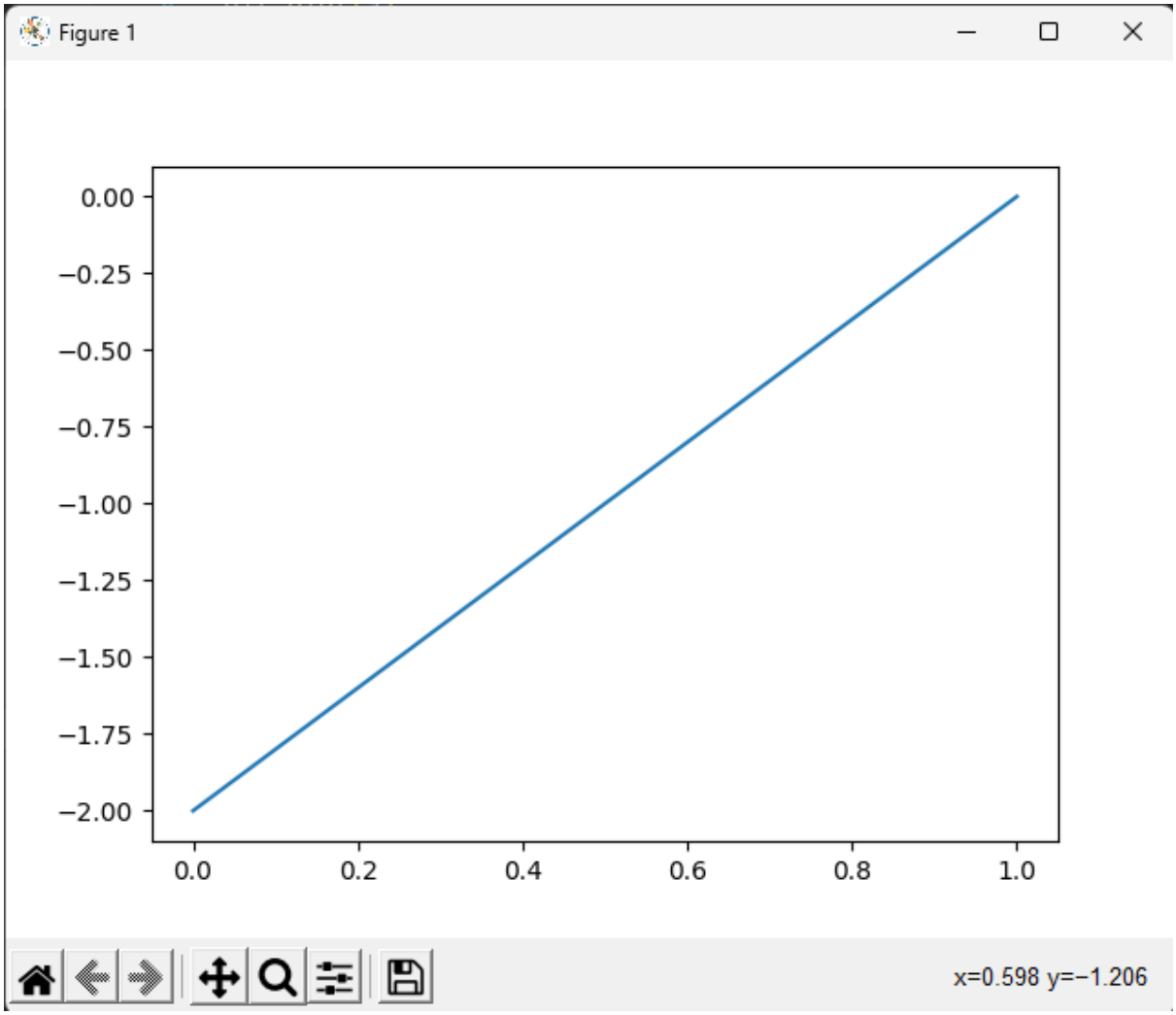
Figure 1
0.00
−0.25
−0.50
−0.75
−1.00
−1.25
−1.50
−1.75
−2.00
0.0
0.2
0.4
0.6
0.8
1.0
x=0.598 y=−1.206