**Name: Mohammad Zain Abbas**

**Reg #: 6865**

**DE-36 (CE), Syndicate: A**

**LAB 14 JOURNEL**

**Equipment Used:** Notebook Computer, Matlab , Visual Studio(C#)

**Lab Tasks:**

Develop a C# console application to implement perceptron for the following dataset. Use

learning rate of 0.1 with zero biasing factor and initialize all the weights with 0:

**SOLUTION CODE:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace LAB14

{

public class FeatureVector

{

public int x0;

public int x1;

public int x2;

public FeatureVector(int age, int loan, int defaulter)

{

this.x0 = x0;

this.x1 = x1;

this.x2 = x2;

}

}

class Program

{

static void Main(string[] args)

{

double learning\_rate = 0.1;

List<FeatureVector> Train\_Dataset = new List<FeatureVector>();

Train\_Dataset.Add(new FeatureVector(1, 0, 0));

Train\_Dataset.Add(new FeatureVector(1, 0, 1));

Train\_Dataset.Add(new FeatureVector(1, 1, 0));

Train\_Dataset.Add(new FeatureVector(1, 1, 1));

List <double>Actual\_Classes = new List<double>();

Actual\_Classes.Add(1);

Actual\_Classes.Add(1);

Actual\_Classes.Add(1);

Actual\_Classes.Add(0);

List<double> Weights = new List<double>();

Weights.Add(0);

Weights.Add(0);

Weights.Add(0);

double Threshold = 0;

double output;

double count = 0;

int Iterations = 0;

while(true)

{

List<double> Predicted\_Classes = new List<double>();

for (int i=0;i<Train\_Dataset.Count;i++)

{

double value = Weights[0] \* Train\_Dataset[i].x0 + Weights[1] \* Train\_Dataset[i].x1 + Weights[2] \* Train\_Dataset[i].x2;

if(value > Threshold)

{

output = 1;

}

else

{

output = 0;

}

Predicted\_Classes.Add(output);

double Error = learning\_rate \* (output - value);

Weights[0] = Weights[0] + Train\_Dataset[i].x0 \* Error;

Weights[1] = Weights[1] + Train\_Dataset[i].x1 \* Error;

Weights[2] = Weights[2] + Train\_Dataset[i].x2 \* Error;

}

for (int j = 0; j < Predicted\_Classes.Count; j++)

{

if(Predicted\_Classes[j]==Actual\_Classes[j])

{

count = count + 1;

}

}

Iterations = Iterations + 1;

Console.WriteLine("No of Iterations = " + Iterations);

double Accuracy = (count / Actual\_Classes.Count())\*100;

Console.WriteLine("Accuracy = " + Accuracy);

if(Accuracy==100)

{

break;

}

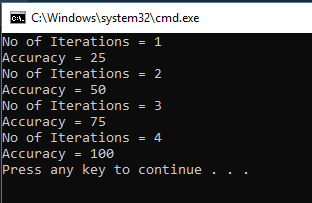
}

}

}

}

**OUTPUT:**



2) Use the given cancer dataset and classify it using pceptron:

a) First create a MATLAB script and load ‘cancer’ mat file.

b) Identify features and classes from the loaded dataset.

c) Perform 2-fold cross validation on the dataset by splitting it into testing and

training parts.

d) Implement the perceptron using the above algorithm and use tining dataset to

classify each of e sample within testing dataset. Use learning rate of 0.1 and 0

biasing factor with 0 initial weights.

e) Repeat the above process until the accuracy of 80% or above is not achieved

**SOLUTION CODE:**

warning off

clear all;

load ('cancer.mat')

[rows,cols]=size(dataset);

Training\_Dataset=dataset(1:rows/2,1:9);

Training\_Labels=dataset(1:rows/2,10);

Testing\_Dataset=dataset((rows/2):end-1,1:9);

Testing\_Dataset(1,1:9)=ones(1,9);

Testing\_Labels=dataset(rows/2:end-1,10);

Testing\_Labels=Testing\_Labels';

learning\_rate=0.1;

weights=zeros(1,9);

Predicted\_labels=zeros(1,length(Testing\_Labels));

threshold=0;

j=0;

while(1)

for i=1:length(Testing\_Dataset)

value=sum(weights.\*Testing\_Dataset(i,1:9));

if(value>threshold)

output=1;

else

output=0;

end

Predicted\_labels(i)=output;

Error=learning\_rate\*(Testing\_Labels(i)-output);

weights(1:9)=weights(1:9)+Testing\_Dataset(i,1:9).\*Error;

end

Accuracy=((sum(Predicted\_labels==Testing\_Labels))/length(Testing\_Labels))\*100;

j=j+1;

if(Accuracy>=80)

break;

end

end

**OUTPUT:**

