Assignment-3

CLID: sxb8547

Email-id:saisreebhagi@gmail.com

1. Partition a data file (in ASCII) into training and test sets.

Solution:

Source code:

```
#include<stdio.h>
#include<conio.h>
#include<math.h>
#include<string.h>
#include<stdlib.h>
#define BLOCK    1000
int main(int argc, char* argv[])
{
	FILE *fp;
	char ch;
	fp = fopen("data.txt","r");
	if ( fp == NULL )
	{
		puts ( "Cannot open source file");
	}

		char c;
	    while((c = fgetc(fp)) != EOF)
	{
	      printf("%c",c);
	    }
	/*reopen the file to reset the pointer*/
	fclose(fp);
	fp = fopen("data.txt","r");
	int m[0][0];
	int n[0][0];
	/*Load file into array*/
	int i,j;
	for(i=0;i<10;i++)
	{
		for(j=0;j<10;j++)
		{
			fscanf( fp, "%d", &(m[i][j]));
```

```
            }
            fscanf(fp, "\n", NULL);
        }

    /*print out stored matrix*/
        for(i=0;i<1;i++)
    {
            for(j=0;j<1;j++)
            {
               printf("%d ",m[i][j]);
            }
            printf("\n");
    }
    int k,l;
    for(k=0;k<10;k++)
    {
            for(l=0;l<10;l++)
            {
                  fscanf( fp, "%d", &(n[k][l]));
            }
            fscanf(fp, "\n", NULL);
        }

    /*print out stored matrix*/

        for(k=0;k<1;k++)
    {
            for(l=0;l<1;l++)
            {
               printf("%d ",n[k][l]);
            }
            printf("\n");
    }
}
```

I partitioned 50% of samples as for training data set and remaining 50% as for testing data set. So, after partitioning I displayed 600 samples in training data set and remaining 600 data samples in the testing data set on the console.

Tool or software used: I implemented the program by using Dev C++ tool.

Operating systems: Windows 8

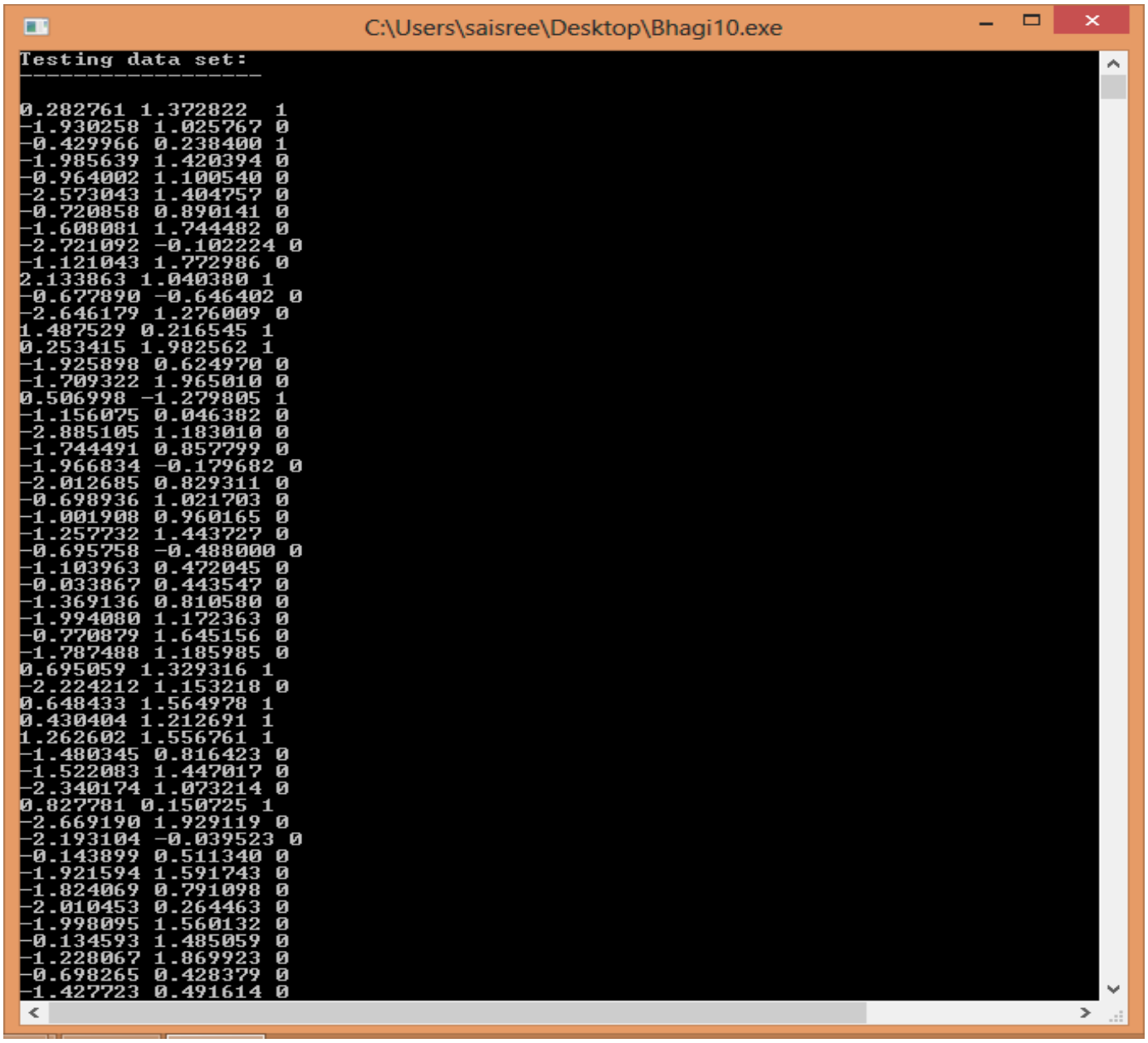
OUTPUT:

600 samples in training data set

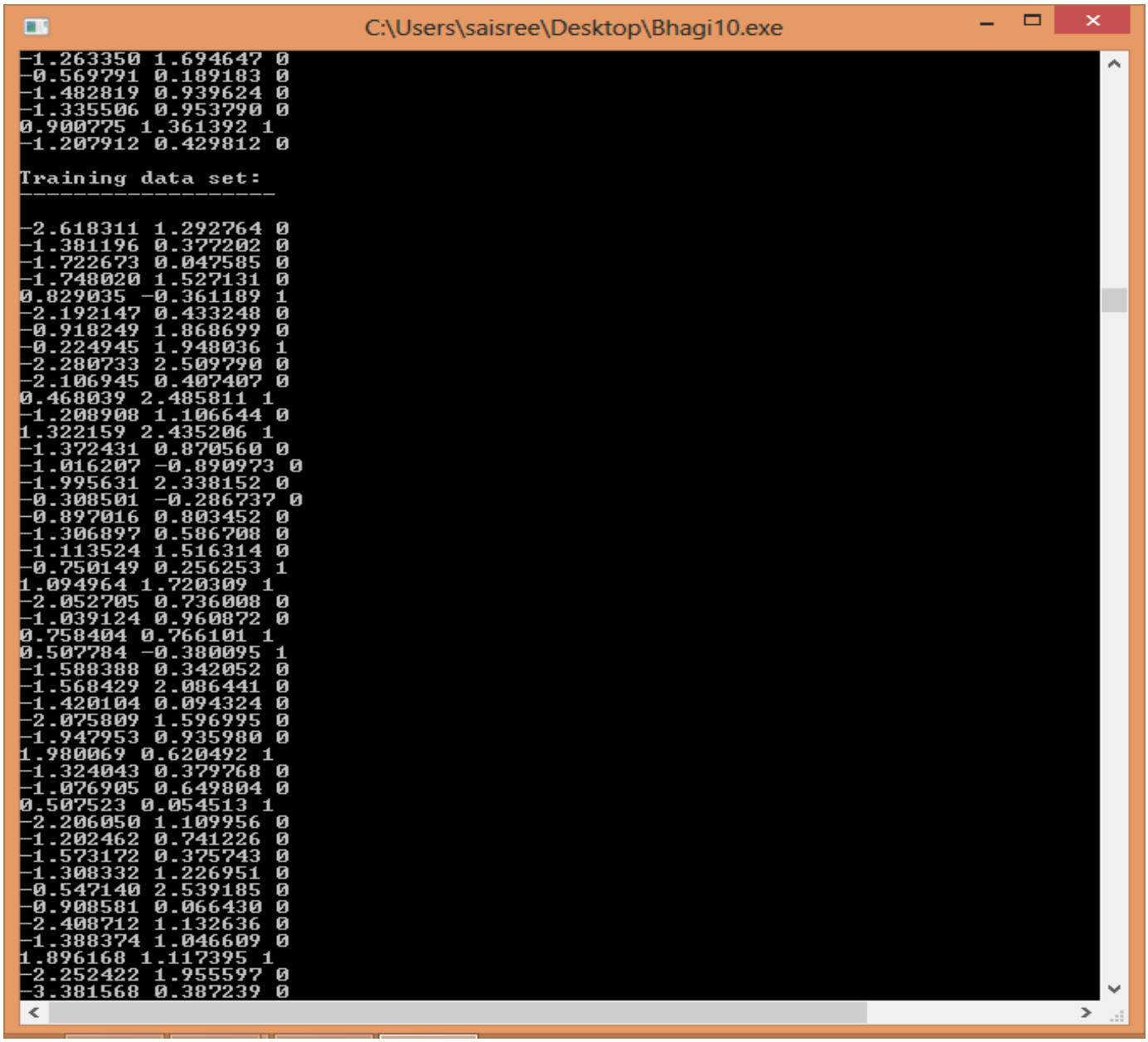
600 samples in testing data set.

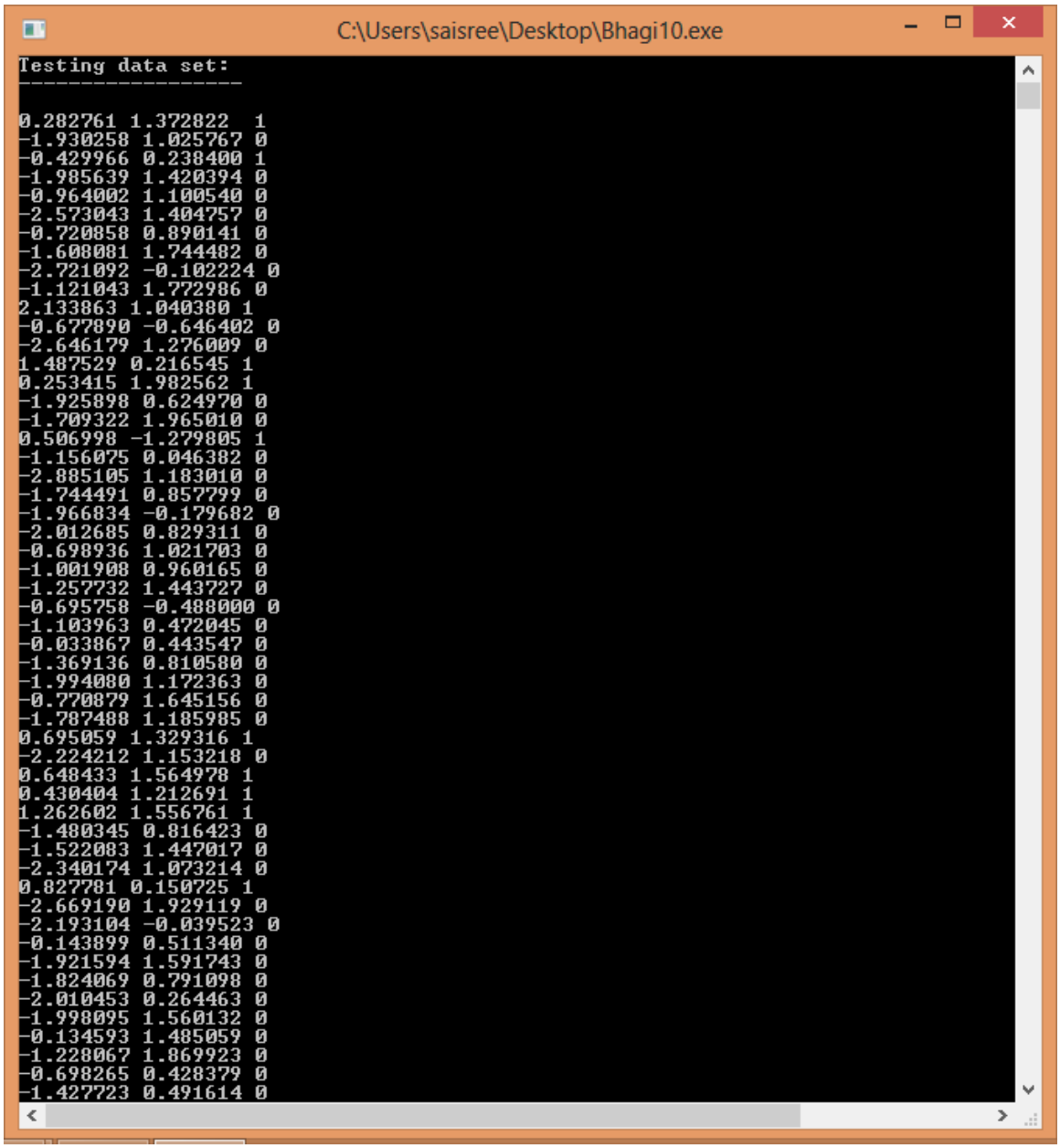
Output screenshot:

Testing data set:



Training data set:


C:\Users\saisree\Desktop\Bhagi10.exe
-1.263350 1.694647 0
-0.569791 0.189183 0
-1.482819 0.939624 0
-1.335506 0.953790 0
0.900775 1.361392 1
-1.207912 0.429812 0
Training data set:
-2.618311 1.292764 0
-1.381196 0.377202 0
-1.722673 0.047585 0
-1.748020 1.527131 0
0.829035 -0.361189 1
-2.192147 0.433248 0
-0.918249 1.868699 0
-0.224945 1.948036 1
-2.280733 2.509790 0
-2.106945 0.407407 0
0.468039 2.485811 1
-1.208908 1.106644 0
1.322159 2.435206 1
-1.372431 0.870560 0
-1.016207 -0.890973 0
-1.995631 2.338152 0
-0.308501 -0.286737 0
-0.897016 0.803452 0
-1.306897 0.586708 0
-1.113524 1.516314 0
-0.750149 0.256253 1
1.094964 1.720309 1
-2.052705 0.736008 0
-1.039124 0.960872 0
0.758404 0.766101 1
0.507784 -0.380095 1
-1.588388 0.342052 0
-1.568429 2.086441 0
-1.420104 0.094324 0
-2.075809 1.596995 0
-1.947953 0.935980 0
1.980069 0.620492 1
-1.324043 0.379768 0
-1.076905 0.649804 0
0.507523 0.054513 1
-2.206050 1.109956 0
-1.202462 0.741226 0
-1.573172 0.375743 0
-1.308332 1.226951 0
-0.547140 2.539185 0
-0.908581 0.066430 0
-2.408712 1.132636 0
-1.388374 1.046609 0
1.896168 1.117395 1
-2.252422 1.955597 0
-3.381568 0.387239 0



2. Train a linear classifier using Widrow-Hoff algorithm. Output the linear classifier weights and the training accuracy.

Solution:

Source code

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <math.h>
int main()
```

```
{
  int j, i, k, diff;
  float w1=-1.5f,w2=1.0f,w3=-1.0,ip,dwt[2],eta=0.2;
  float count1 = 0.0f ,count2 = 0.0f ,count3 = 0.0f;
  float data[1200][3];
  int label[800];
  float num;
    FILE *fp = fopen("data.txt", "r");
       if (fp == NULL)
       {
             printf("\n Can't open the file\n");
              exit;
       }
       fclose(fp);
{
       for(k=1;k<=100;k++)    //iterations
    {
       eta = eta/(float)k;
       for(i=0;i<600;i++)
    {
                              if((ip < 0) && (label[i] == -1) )
                                    {
                                      count1++; // class0
                                    }
                              else if( (ip > 0) && (label[i] == 1) )
                                    {
                                      count2++;    /// class 1
                                    }
                         ip=-1.5f*data[i][0]+1.0f*data[i][1]+(-1.0f);
                         diff = data[i][2] - ip;
                         data[0][0]=eta*diff*w1;
                         data[0][1]=eta*diff*w2;
                         data[1][0]=eta*diff*w3;
                       w1+=dwt[0];
                       w2+=dwt[1];
                       w3+=dwt[2];
       printf(" Enter the weights \n");
       printf("weight 1=");
       scanf("%f" ,&w1);
       printf("weight 2=");
       scanf("%f" ,&w2);
       printf("weight 3=");
       scanf("%f" ,&w3);
       printf(" The updated weights are:"\n);
       printf("w1=%f \n w2=%f \n w3=%f",w1,w2,w3);
       exit(1);
    }
}
}
```

```
        for (i=0; i< 600; i++)
            {
            for(j = 0; j < 3; j++)
                        {
                                        fscanf(fp, "%lf", &num);
                                                                data[i][j] = num;
                        }
            }
printf("\n----------------------------------------------------------------------------------------------------------------------------------------
---------------");
printf("\n Training set: \n");
printf("\n----------------------------------------------------------------------------------------------------------------------------------------
---------------\n");
        for (i=0; i< 600; i++)
            {
             printf("\n");
             for(j = 0; j < 3; j++)
                                {
                                        num = data[i][j];
                                printf("%f\t",data[i][j]);
                        }
                  }
printf("\n----------------------------------------------------------------------------------------------------------------------------------------
---------------");
printf("\n Test set: \n ");
printf("\n----------------------------------------------------------------------------------------------------------------------------------------
---------------\n");
for(i = 600; i < 1200; i++)
   {
              printf("\n");
        for (j=0; j < 3; j++)
                    {
                    num = data[i][j];
                    printf("%f\t", data[i][0]);
                    }
   }
}
```

Report:

OUTPUT:

In this,initially I am declaring the weights to '0'.In next case,I am taking the weights in the range [-1, +1]. So,after updating the data samples with the declared weights in the program by using the widrow-hoff algorithm, I got the weights as follows:

Enter the weights

weight 1=-1.5

weight 2=-1.0

weight 3=1.0

The updated weights are:

w1=-1.500000

w2=-1.000000

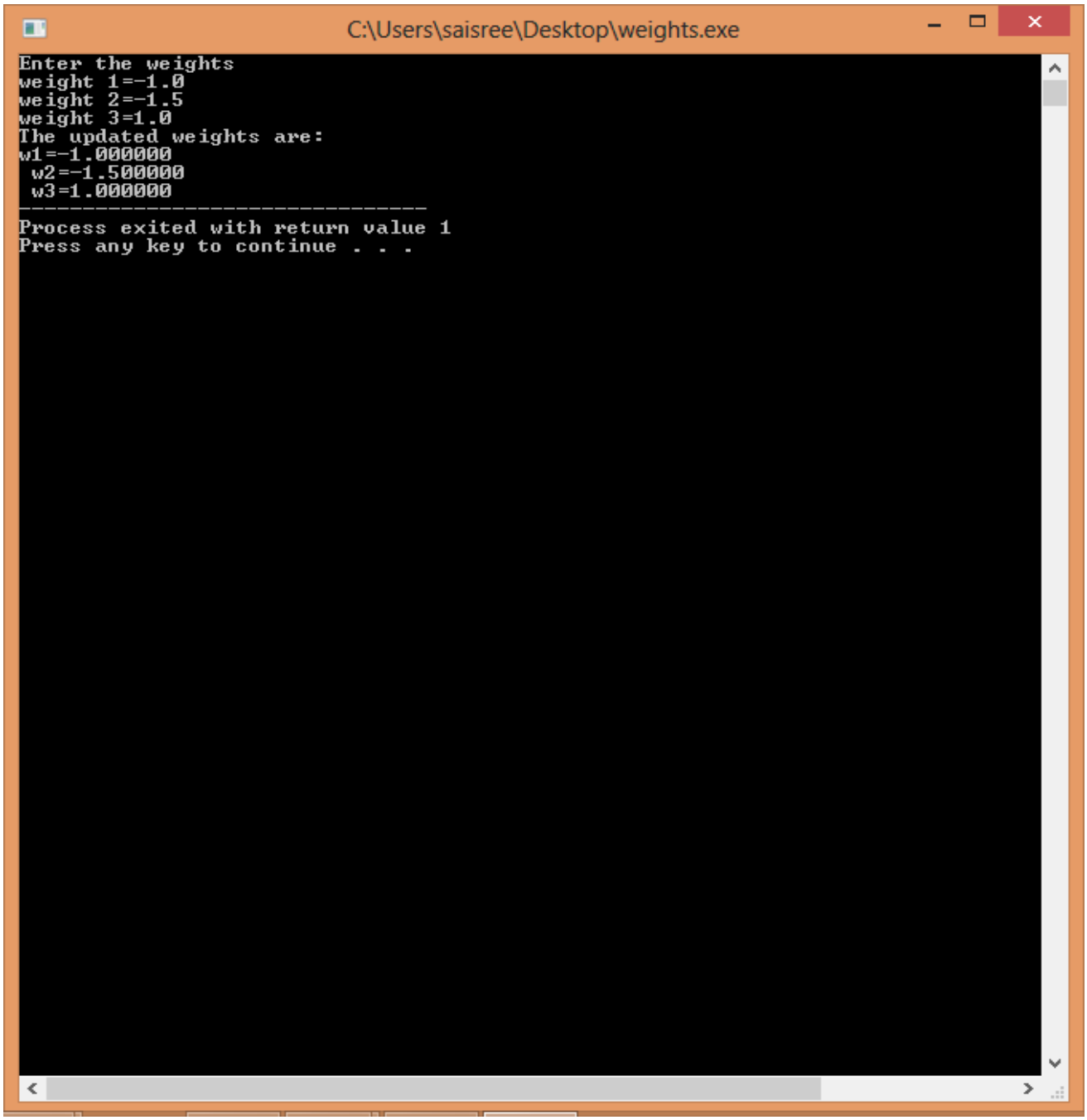
w3=1.000000

Process exited with return value 1

Press any key to continue . . .

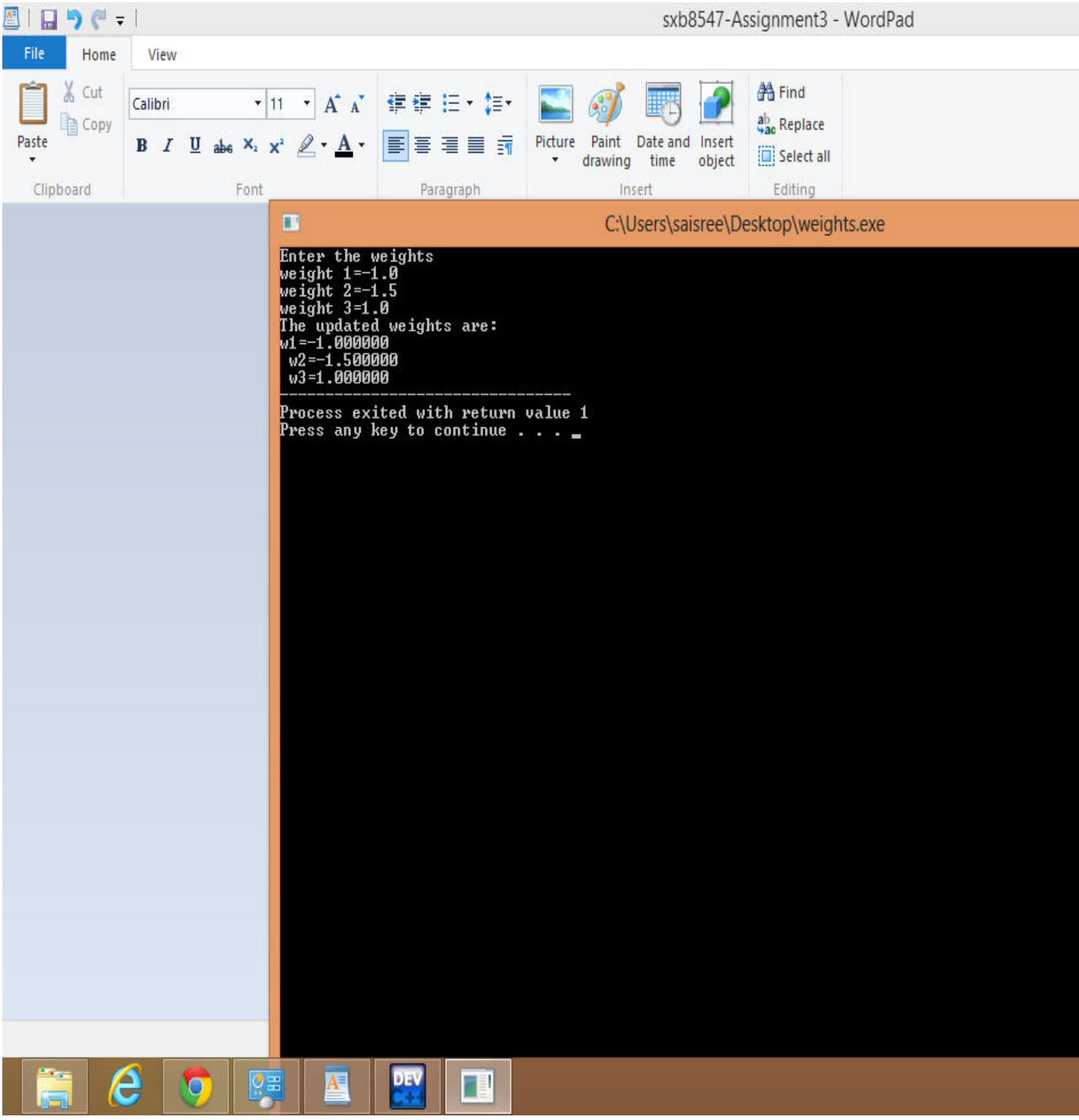
Inorder to train the linear classifier,I replaced weights '0' with ' -1'.

output screenshot:


C:\Users\saisree\Desktop\weights.exe
Enter the weights
weight 1=-1.0
weight 2=-1.5
weight 3=1.0
The updated weights are:
w1=-1.000000
w2=-1.500000
w3=1.000000

Process exited with return value 1
Press any key to continue . . .

(or)



Accuracy:

A Data set data.txt which needs to be partitioned in to two sets namely training set and testing set .

The optimum design rule states that the linear combination of inputs i.e. the weighted sum

$$w^T * x = 0$$

Where w = the weights adapted , the weights are arbitrarily chosen .

x = the observed vectors (the data sample)

Here each and every data value from the input file is multiplied with an arbitrary weight , which on summation becomes the weighted sum.

Widrow-Hoff Linear Classifier

A data set data.txt consisting of data samples or the observation vectors .

Widrow-hoff linear classifier decision rule is obtained as

$W_k = W_{(k+1)} + eta(t-w^t X_k) *X_k$

x is the observed vectors

t is the target vector

w is the weight vectors

where eta=learning rate.

Here the target vector t takes a value 1 if the label is 1 else -1 if the class label is -1 in the Input file

CLASSIFICATION OF DATA:

If ($w^T * x < 0$)

{

Choose class0

}

Else if ($w^T * x > 0$)

{

Choose class1

}

Else

{

misclassified

}

Here the data set or the observation vectors have been divided into training set and testing sets. Initially known weight values are considered during the training of the data set and later the weight vectors are computed and is associated with each data sample during the testing of the data set.

To determine accuracy,Using the testing set we have to calculate the misclassification percentage to determine it's accuracy and by using the training data set,we have to calculate the misclassification rate.

Source code:

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include<conio.h>
```

```
// Opens the file and reads in the stream //

int main(void) //Main Function //
{

double feature_1[1200],feature_2[1200],feature_3[1200];//declare array (random number)
int j , i;
double num;

FILE *fp = fopen("data.txt", "r");

if (fp==NULL) //checks for the file
{
printf("\n File not found \n");
exit;
}
//reads in the data to an array, analyzes, and the prints the array

for (i=0; i< 1200; i++)
{
fscanf(fp,"%lf %lf %lf \n",&feature_1[i],&feature_2[i],&feature_3[i]);

}
for(i=0;i<600;i++)
{
if(feature_3[i]==0)
{
feature_3[i]=-1;
}
}
double weight1=0.25,weight2=0.95,weight3=-.6,internal_result=0;

double difference=0;
int count_one=0,count_two=0;
float accuracy ;

for(i=0;i<600;i++)
{
internal_result=feature_1[i]*weight1+feature_2[i]*weight2+1*weight3;
if(internal_result<0 && feature_3[i] == -1 )
{
count_one++;
}
else
{
count_two++;

}
```

```
}
accuracy=(float)(count_one)/ (float )600; // accuracy =
printf (" the Accuracy of the Linear Classifer if Tested is :%f \n",accuracy);

fclose(fp);

getch();

}
```

Output:

The output of the Linear classifier if tested is: 1.000000

Tool or software used: I implemented the widrow-hoff algorithm by using Dev C++ tool.

Operating systems: Windows 8

3. Run the linear classifier trained in (2) using the test set. Output the test set performance using as many measures as you know from this course (use a minimum set of accuracy and precision).

Solution:

In this,the updated output weights which I got for the training data set are taken as an input to the testing data set.
The former w1,w2,w3 values in training data set are taken as an initial input weights in the test set.
Using these updated weights of training data set, I updated the weights for the test set.

Source code

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <math.h>
int main()
{
  int j, i, k, diff;
  float w1=-1.5f,w2=1.0f,w3=-1.0,ip,dwt[2],eta=0.2;
  float count1 = 0.0f ,count2 = 0.0f ,count3 = 0.0f;
  float data[1200][3];
  int label[800];
  float num;
    FILE *fp = fopen("data.txt", "r");
       if (fp == NULL)
       {
             printf("\n Can't open the file\n");
              exit;
```

```
        }
        fclose(fp);
{
        for(k=1;k<=100;k++)    //iterations
    {
        eta = eta/(float)k;
        for(i=600;i<1200;i++)
    {
                            if((ip < 0) && (label[i] == -1) )
                                {
                                  count1++; // class0
                                }
                            else if( (ip > 0) && (label[i] == 1) )
                                {
                                  count2++;    /// class 1
                                }
                         ip=-1.5f*data[i][0]+1.0f*data[i][1]+(-1.0f);
                         diff = data[i][2] - ip;
                         data[0][0]=eta*diff*w1;
                         data[0][1]=eta*diff*w2;
                         data[1][0]=eta*diff*w3;
                       w1+=dwt[0];
                       w2+=dwt[1];
                       w3+=dwt[2];
        printf(" Enter the weights \n");
        printf("weight 1=");
        scanf("%f" ,&w1);
        printf("weight 2=");
        scanf("%f" ,&w2);
        printf("weight 3=");
        scanf("%f" ,&w3);
        printf(" The updated weights for the test set are:"\n);
        printf("w1=%f \n w2=%f \n w3=%f",w1,w2,w3);
        exit(1);
    }
}
}
        for (i=600; i< 1200; i++)
            {
            for(j = 0; j < 3; j++)
                        {
                                        fscanf(fp, "%lf", &num);
                                                                data[i][j] = num;
                        }
            }
printf("\n----------------------------------------------------------------------------------------------------------------------------------
----------------");
printf("\n Test set: \n ");
printf("\n----------------------------------------------------------------------------------------------------------------------------------
```

```
----------\n");
for(i = 600; i < 1200; i++)
   {
            printf("\n");
      for (j=0; j < 3; j++)
                {
                num = data[i][j];
                printf("%f\t", data[i][0]);
                }
   }
}
```

OUTPUT:

Enter the weights

weight 1= -1.5

weight 2= -1.0

weight 3= 1.0

The updated weights for the test set are:

w1=-1.500000

w2=-1.000000

w3=1.000000

Process exited with return value 1

Press any key to continue . . .

Accuracy:

To determine accuracy,Using the testing set we have to calculate the misclassification percentage to determine it's accuracy and by using the training data set,we have to calculate the misclassification rate.

Source code:

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include<conio.h>

// Opens the file and reads in the stream //
```

```
int main(void) //Main Function //
{

double feature_1[1200],feature_2[1200],feature_3[1200];//declare array (random number)
int j , i;
double num;

FILE *fp = fopen("data.txt", "r");

if (fp==NULL) //checks for the file
{
printf("\n File not found \n");
exit;
}
//reads in the data to an array, analyzes, and the prints the array

for (i=0; i< 1200; i++)
{
fscanf(fp,"%lf %lf %lf \n",&feature_1[i],&feature_2[i],&feature_3[i]);

}
for(i=0;i<600;i++)
{
if(feature_3[i]==0)
{
feature_3[i]=-1;
}
}
double weight1=0.25,weight2=0.95,weight3=-.6,internal_result=0;

double difference=0;
int count_one=0,count_two=0;
float accuracy ;

for(i=0;i<600;i++)
{
internal_result=feature_1[i]*weight1+feature_2[i]*weight2+1*weight3;
if(internal_result<0 && feature_3[i] == -1 )
{
count_one++;
}
else
{
count_two++;

}

}
```

```
accuracy=(float)(count_one)/ (float )600; // accuracy =
printf (" the Accuracy of the Linear Classifer if Tested is :%f \n",accuracy);

fclose(fp);

getch();

}
```

Output:

The output of the Linear classifier if tested is: 1.000000

Tool or software used: I implemented the widrow-hoff algorithm and it's accuracy by using Dev C++ tool. i compiled the source code by using F9 and I run it by using F10 command.

Operating systems: Windows 8