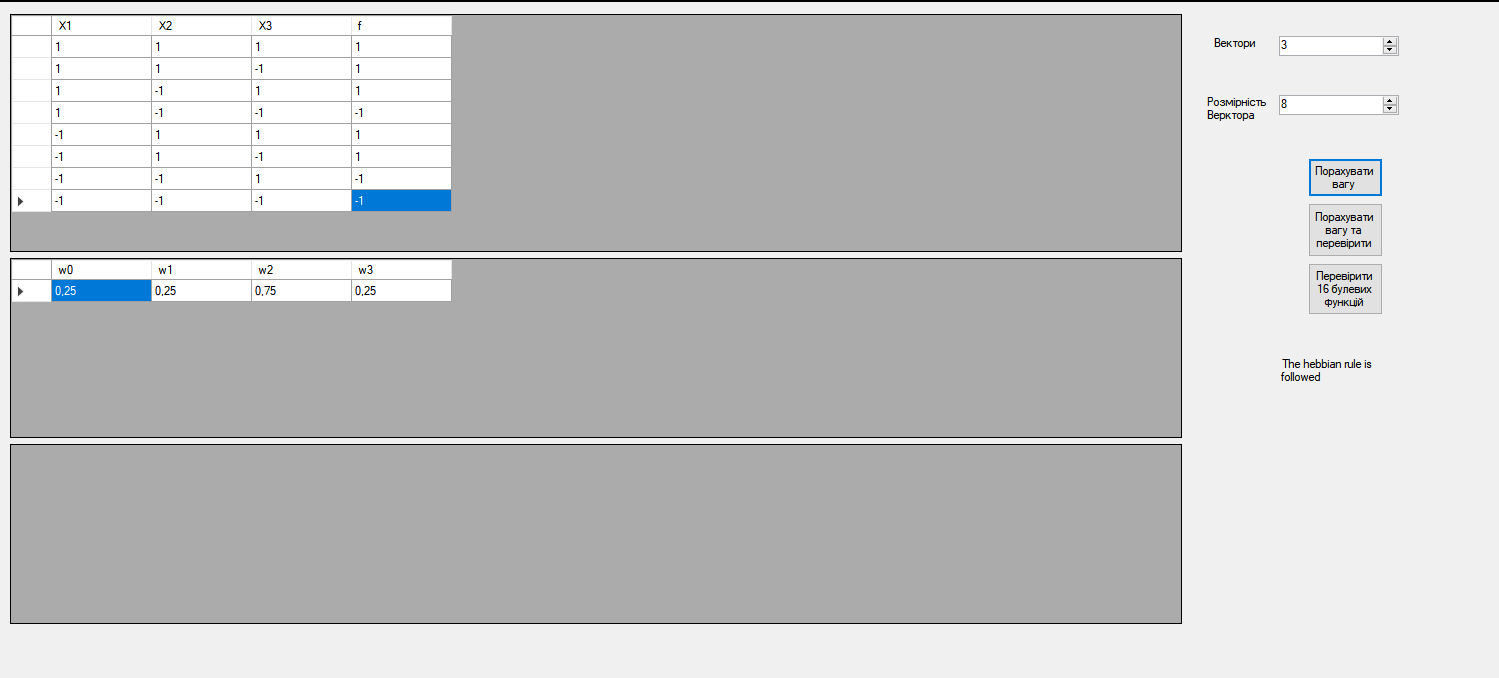
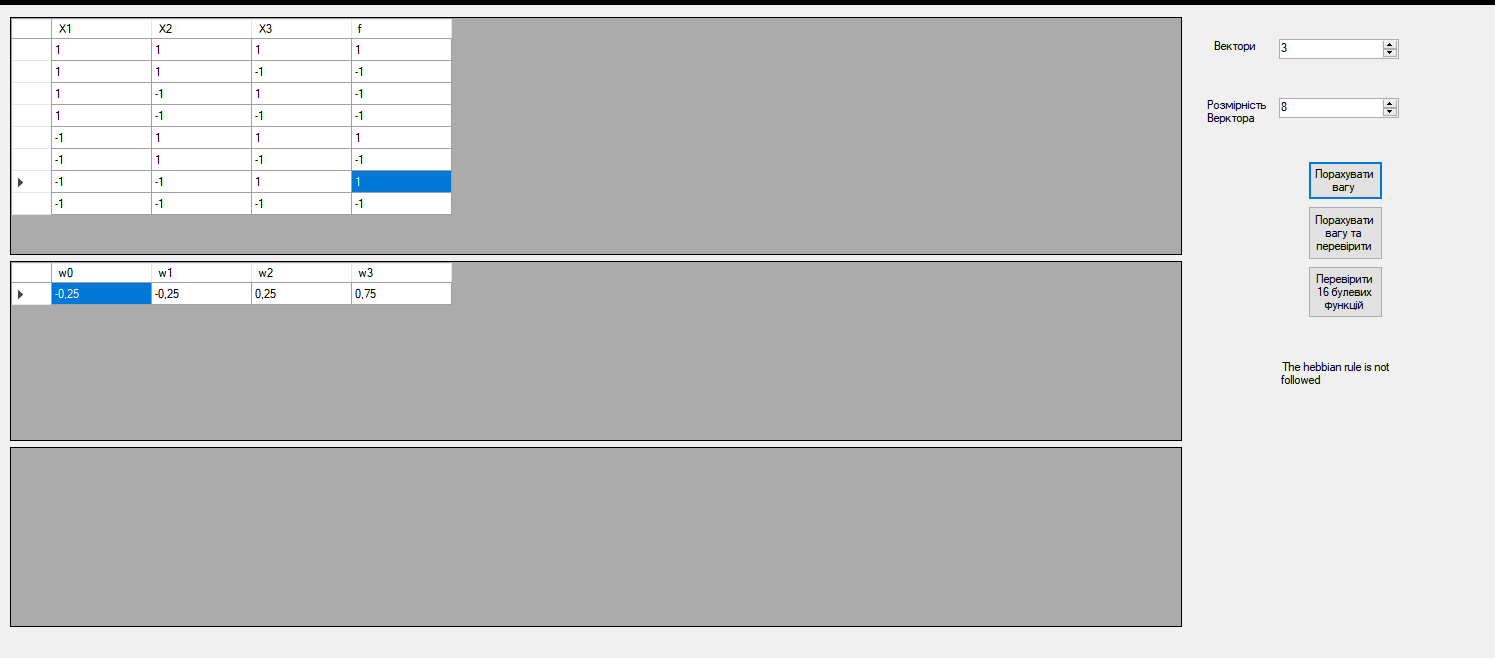
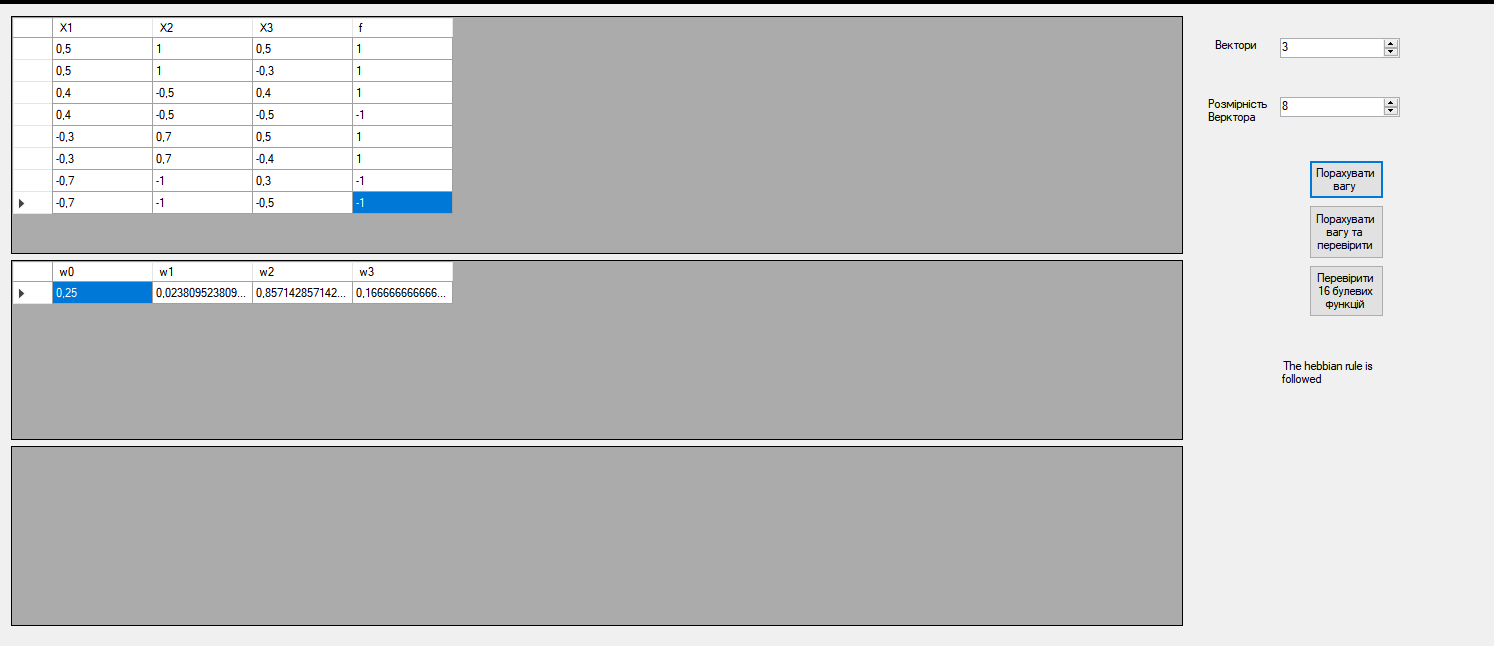
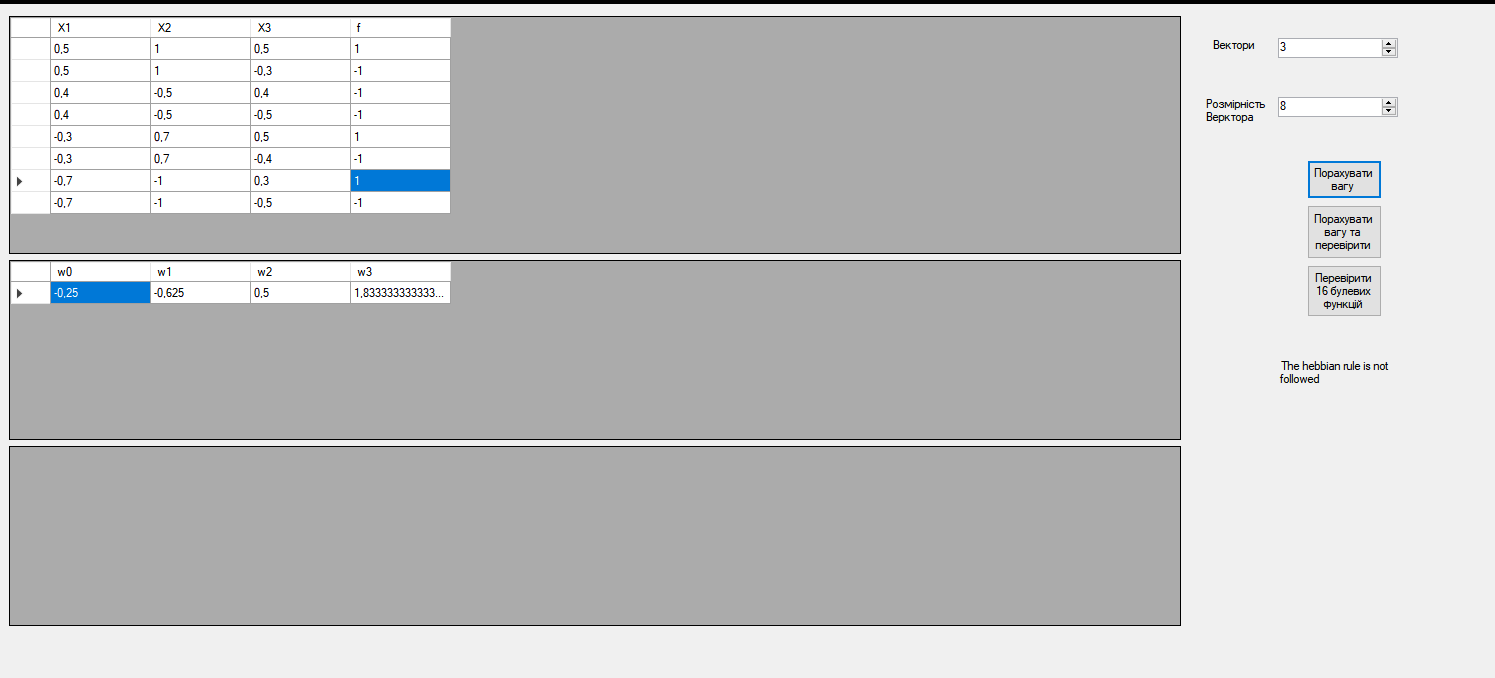
1.





2.





4.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace HebbianLearning

{

class HebbianLearningClass

{

private double[][] InputArr;

private double[] OutputArr;

private int N;

public HebbianLearningClass(double[][] inputArr, double[] outputArr)

{

double[] X0 = new double[inputArr[1].Length];

for (int i = 0; i < X0.Length; i++)

{

X0[i] = 1;

}

InputArr = new double[inputArr.Length + 1][];

InputArr[0] = X0;

inputArr.CopyTo(InputArr, 1);

OutputArr = outputArr;

N = InputArr[1].Length;

}

public double Scalar(double[] a1, double[] a2)

{

double sum = 0;

for (int i = 0; i < a1.Length; i++)

{

sum += a1[i] \* a2[i];

}

return sum;

}

public double[] WeightArr()

{

double[] output = new double[InputArr.Length];

for (int i = 0; i < output.Length; i++)

{

output[i] = Scalar(Transp(InputArr[i]),Transp(OutputArr)) / N;

}

return output;

}

public double[] WeightSum()

{

double[] WeightList = WeightArr();

double[][] transp = ArrTransp();

double[] output = new double[InputArr[1].Length];

for (int i = 0; i < output.Length; i++)

{

output[i] = Scalar(transp[i], WeightList);

}

return output;

}

public double[] Transp(double[] arr)

{

for (int i = 0; i < arr.Length; i++)

{

arr[i] = 1 / arr[i];

}

return arr;

}

public double[][] ArrTransp()

{

int k = 0;

double[][] output = new double[InputArr[1].Length][];

for (int i = 0; i < InputArr[1].Length; i++)

{

output[i] = new double[InputArr.Length];

for (int j = 1; j < InputArr.Length; j++)

{

k++;

output[i][j] = InputArr[j][i];

}

output[i][0] = 1;

}

return output;

}

public int[] ConvertToThreshold()

{

double[] input = WeightSum();

int[] output = new int[input.Length];

for (int i = 0; i < input.Length; i++)

{

output[i] = (input[i] < 0) ? -1 : 1;

}

return output;

}

}

}