Министерство образования Республики Беларусь

Учреждение образования

«Брестский государственный технический университет»

Кафедра ИИТ

**Лабораторная работа №7**

**Тема: «Векторный квантователь»**

**Выполнил:**

Студент 4 курса

Факультета ЭИС

Группы АС-55

Черноокий И.В.

**Проверил:**

Савицкий Ю.В.

Брест 2022

**Лабораторная работа №7**

**«Векторный квантователь»**

**Цель работы:** изучить обучение и функционирование искусственной нейронной сети (ИНС) векторного квантователя при решении задач распознавания образов.

**Задание:** написать на любом языке высокого уровня программу моделирования ИНС векторного квантователя для распознавания образов.

**Текст программы:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Lab6

{

class Network

{

private double[][] w = new double[3][];

private int[][] vectors;

private int[] inputVector;

public Network(int[][] vectors)

{

this.vectors = vectors;

Random random = new Random();

for (int i = 0; i < 3; i++)

{

w[i] = new double[20];

for (int j = 0; j < 20; j++)

{

w[i][j] = random.NextDouble();

}

}

}

private void setVector(int vectorNumber)

{

inputVector = new int[20];

for (int i = 0; i < vectors[vectorNumber].Length; i++)

{

inputVector[i] = vectors[vectorNumber][i];

}

}

private double[] calculateVector()

{

double[] output = new double[3];

for (int i = 0; i < 3; i++)

{

for (int j = 0; j < 20; j++)

{

output[i] += Math.Pow(inputVector[j] - w[i][j], 2);

}

output[i] = Math.Sqrt(output[i]);

}

return output;

}

private int findWinner(double[] output)

{

// Поиск минимума

double minimal = output.Min();

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

if (output[i] == minimal)

return i;

return 0;

}

private void recalculateWinner(int winner, int real, int iteration)

{

for (int i = 0; i < 20; i++)

{

w[winner][i] += (winner == real ? 1 : -1) \* ((double)inputVector[i] - w[winner][i]) / (double)(iteration + 1);

}

}

public void Study(int totalIterations)

{

for (int i = 0; i < totalIterations; i++)

{

for (int j = 0; j < 3; j++)

{

this.setVector(j);

double[] output = this.calculateVector();

int winner = this.findWinner(output);

this.recalculateWinner(winner, j, i);

}

}

}

public int Recognize(int[] recognizingVector)

{

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

{

inputVector[i] = recognizingVector[i];

}

double[] output = this.calculateVector();

return findWinner(output);

}

}

class Program

{

static void Main(string[] args)

{

int[][] vectors = {

new int[] { 0, 1, 0, 0, 1, 1, 0, 1, 0, 0, 0, 0, 1, 0, 1, 0, 1, 0, 0, 0 },

new int[] { 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 0, 0 },

new int[] { 1, 1, 1, 0, 0, 0, 1, 1, 1, 0, 0, 0, 1, 1, 1, 0, 0, 0, 1, 1 }

};

Network network = new Network(vectors);

network.Study(400);

Random random = new Random();

for (int i = 2; i < 20; i++)

{

int count = 0;

int vectorNumber = random.Next(1000) % 3;

int[] tempVector = new int[20];

Array.Copy(vectors[vectorNumber], tempVector, tempVector.Length);

Console.WriteLine("Исходный вектор: ");

for (int j = 0; j < vectors[vectorNumber].Length; j++)

Console.Write("{0} ", vectors[vectorNumber][j]);

Console.WriteLine();

for (int j = 0; j < i; j += 2, count++)

{

tempVector[j] = (tempVector[j] == 0) ? 1 : 0;

}

Console.WriteLine("Новый вектор: ");

for (int j = 0; j < tempVector.Length; j++)

Console.Write("{0} ", tempVector[j]);

Console.WriteLine();

Console.WriteLine("Исходный номер: {0}", vectorNumber);

Console.WriteLine("Распознанный номер: {0}", network.Recognize(tempVector));

Console.WriteLine("Измененных бит: {0}", count);

Console.WriteLine();

}

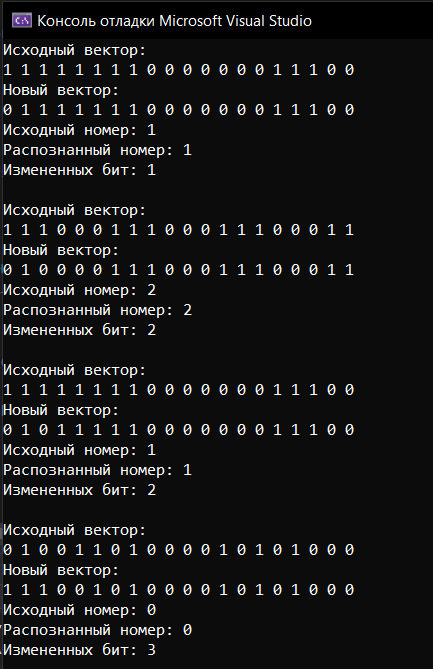
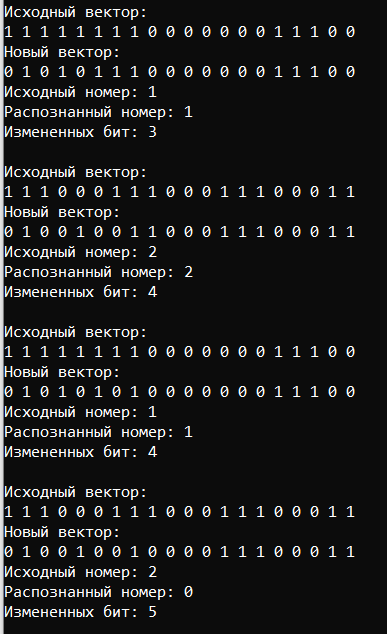
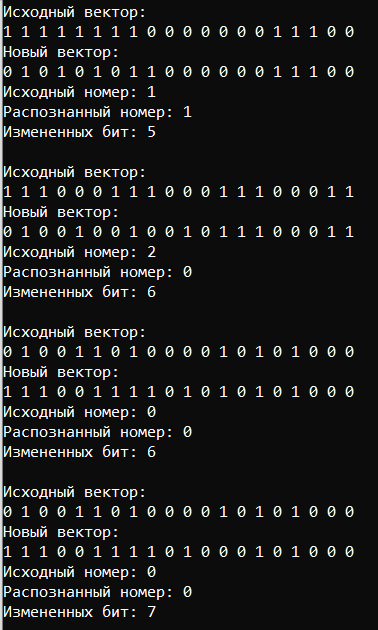
Console.ReadKey(true);

}

}

}

**Результат работы:**

**  **

Видно, что сеть способна распознавать образы до тех пор, пока количество измененных бит не приблизится к половине их длины. Этот параметр также будет зависеть от того, какой вид имеют другие образы.

**Вывод**: изучил обучение и функционирование ИНС Хопфилда при решении задач распознавания образов.