Федеральное государственное бюджетное образовательное учреждение высшего образования   
«Саратовский государственный технический университет имени Ю. А. Гагарина»

Кафедра прикладных информационных технологий

**Практическая работа по курсу**

**Компьютерные методы моделирования искусственных нейронных сетей**

**«Разработка нейроэмулятора ассоциативной памяти»**

Выполнил студент б1-ИФСТ-31,

Кошелева Алина Денисовна

Проверил преподаватель КММИНС,

Кузьмин Алексей Константинович

Саратов, 2024

**Цель работы**

Цель данной практической работы заключается в разработке нейроэмулятора ассоциативной памяти, который способен выполнять восстановление и распознавание образов. В ходе работы необходимо изучить алгоритмы и методы обучения для сети Хемминга. Для данной сети необходимо выстроить соответствующую архитектуру, создать программное обеспечение для обучения нейроэмулятора на обучающей выборке и провести эксперименты для оценки качества работы.

**Сеть Хемминга**

**Предметная область**

В качестве объектов для распознавания возьмем записи с информацией о характеристиках стран мира:

1. **Место в мире по площади выращивания винограда**;
2. **Место в мире по численности населения винодельцев** (это предложение подчеркивает количество людей, занимающихся виноделием в данном месте);
3. **Форма организации винодельческого сообщества** (это предложение указывает на специфическую структуру и организацию винодельческого сообщества, включая правила, нормы и процессы, применяемые в данной области);
4. **Организация винодельческих регионов** (это предложение подразумевает, что территории, где выращивается виноград и производится вино, организованы в регионы с определенными характеристиками и правилами);
5. **Географическое положение винодельческой территории** (это предложение подчеркивает местоположение конкретной области, где происходит выращивание винограда и производство вина);
6. **Уровень развития винодельческой инфраструктуры** (это предложение указывает на степень развития инфраструктуры, связанной с виноделием, в данной области);
7. **Индекс винодельческого развития** (это предложение подразумевает оценку и измерение уровня развития винодельческой отрасли в данной области);
8. **Место в мире по объему винодельческой экономики** (это предложение указывает на позицию данной области виноделия в мировой экономике, основываясь на объеме производства и экономической активности винодельческой отрасли);

**Реализация**

**Класс, который выполняет необходимую логику по вычислениям.**

namespace HammingNetwork

{

delegate double ActiveFunction(double x);

public class HammingNetwork

{

//число входных параметров

public static int M;

//число эталонов, которые умеем распознавать

public static int K;

// = M/2

public static double T;

// меньше 1/K

public static double E;

//до сколько обучаться

public static double Emin;

//первый слой

double[,] ws;

//второй слой

double[,] es;

//активационная функция

ActiveFunction f;

public HammingNetwork(double[,] xs)

{

ws = new double[K, M];

es = new double[K, K];

for (int i = 0; i < K; i++) {

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

{

ws[i,j] = xs[i, j] / 2;

}

}

f = (x) =>

{

if (x <= 0) return 0;

else if (x <= T) return x;

return T;

};

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

{

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

{

if (i == j) es[i, j] = 1;

else es[i, j] = -E;

}

}

}

public double[] Find(double[] x)

{

//выход первого слоя

double[] s = new double[K];

//умножение матрицы весов на вектор входных параметров

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

{

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

{

s[i] += ws[i,j] \* x[j];

}

//прибавление вектора T = {T, ..., T}

s[i] += T;

}

//выход второго слоя на первом шаге это выход первого слоя

double[] y = new double[K];

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

{

y[i] = s[i];

}

//вспомогательный массив

double[] z;

while (true)

{

z = new double[K];

//выход второго слоя до применения активационной функции

s = new double[K];

//умножение матрицы второго слоя на выход второго слоя

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

{

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

{

s[i] += es[i, j] \* y[j];

}

}

//применяем активационную функцию

double d = 0;

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

{

z[i] = f(s[i]);

//считаем расхождение между выходами

d += (z[i] - y[i]) \* (z[i] - y[i]);

}

//если расхождение минимально - конец

if (d < Emin)

{

break;

}

//иначе продолжаем

y = z;

}

return z;

}

}

}

**Класс для проверки входных параметров и проверки по корректности входных параметров.**

namespace HammingNetwork

{

/// <summary>

/// Логика взаимодействия для MainWindow.xaml

/// </summary>

public partial class MainWindow : Window

{

HammingNetwork net;

List<string> names = new List<string>();

public MainWindow()

{

InitializeComponent();

}

//Нормализация входных параметров

private double[] MyPars(string[] ps)

{

double[] res = new double[HammingNetwork.M];

int a = 0;

//Место в мире по площади выращивания винограда

names.Add(ps[0]);

//Место в мире по численности населения винодельцев

byte p1 = byte.Parse(ps[1]);

for (int j = 0; j < 8; j++, a++)

{

res[a] = (p1 & 1) == 1 ? 1 : -1; // true это 1 - false это -1

p1 >>= 1;

}

byte p2 = byte.Parse(ps[2]);

for (int j = 0; j < 8; j++, a++)

{

res[a] = (p2 & 1) == 1 ? 1 : -1;

p2 >>= 1;

}

//Форма организации винодельческого сообщества

res[a] = "Корпорации и семейные винодельни".Equals(ps[3]) ? 1 : -1;

a++;

//Организация винодельческих регионов

res[a] = "Контроль происхождения и качества вина".Equals(ps[4]) ? 1 : -1;

a++;

//Географическое положение винодельческой территории

res[a] = "Европа".Equals(ps[5]) || "Евразия".Equals(ps[5]) ? 1 : -1; a++;

res[a] = "Aзия".Equals(ps[5]) || "Евразия".Equals(ps[5]) ? 1 : -1; a++;

res[a] = "Северная Америка".Equals(ps[5]) ? 1 : -1; a++;

res[a] = "Южная Америка".Equals(ps[5]) ? 1 : -1; a++;

res[a] = "Африка".Equals(ps[5]) ? 1 : -1; a++;

res[a] = "Австралия".Equals(ps[5]) ? 1 : -1; a++;

//Уровень развития винодельческой инфраструктуры

res[a] = "низкий".Equals(ps[6]) ? 1 : -1; a++;

res[a] = "средний".Equals(ps[6]) ? 1 : -1; a++;

res[a] = "высокий".Equals(ps[6]) ? 1 : -1; a++;

res[a] = "очень высокий".Equals(ps[6]) ? 1 : -1; a++;

//Индекс винодельческого развития

res[a] = "низкий".Equals(ps[7]) ? 1 : -1; a++;

res[a] = "средний".Equals(ps[7]) ? 1 : -1; a++;

res[a] = "высокий".Equals(ps[7]) ? 1 : -1; a++;

res[a] = "очень высокий".Equals(ps[7]) ? 1 : -1; a++;

//Место в мире по объему винодельческой экономики

byte p8 = byte.Parse(ps[8]);

for (int j = 0; j < 8; j++, a++)

{

res[a] = (p8 & 1) == 1 ? 1 : -1;

p8 >>= 1;

}

return res;

}

private double[,] GetData(string path)

{

double[,] xs = new double[HammingNetwork.K, HammingNetwork.M];

using(StreamReader r = new StreamReader(path))

{

for (int i = 0; i < HammingNetwork.K; i++)

{

string[] line = r.ReadLine().Split('\t');

double[] res = MyPars(line);

for (int j = 0; j < HammingNetwork.M; j++) xs[i, j] = res[j];

}

}

return xs;

}

private void Button\_Click(object sender, RoutedEventArgs e)

{

HammingNetwork.M = int.Parse(M.Text);

HammingNetwork.K = int.Parse(K.Text);

HammingNetwork.E = double.Parse(E.Text);

HammingNetwork.Emin = double.Parse(Emin.Text);

HammingNetwork.T = double.Parse(M.Text) / 2;

string path = Path.Text;

double[,] xs = GetData(path);

net = new HammingNetwork(xs);

MessageBox.Show("Сеть создана!");

}

private void Button\_Click\_1(object sender, RoutedEventArgs e)

{

double[] xs = MyPars(new string[] {

What.Text,

Size.Text,

People.Text,

Type1.Text,

Type2.Text,

Place.Text,

Urban.Text,

IPD.Text,

GDP.Text

});

double[] res = net.Find(xs);

string name = "";

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

{

if (res[i] > 0) name += this.names[i] + " ";

}

Out.Text = name;

}

private void Button\_Click\_2(object sender, RoutedEventArgs e)

{

string path = Path.Text;

int n = int.Parse(N.Text);

string r = File.ReadLines(path).Skip(n - 1).First();

string[] rs = r.Split('\t');

What.Text = "Введенный образ: " + rs[0];

Size.Text = rs[1];

People.Text = rs[2];

Type1.Text = rs[3];

Type2.Text = rs[4];

Place.Text = rs[5];

Urban.Text = rs[6];

IPD.Text = rs[7];

GDP.Text = rs[8];

}

private void Path\_TextChanged(object sender, TextChangedEventArgs e)

{

}

}

}

**Форма представления параметров**

Количественные признаки (1, 2, 8) будем представлять 8 входными параметрами, так как количество стран и регионов, которые производят вино в мире меньше 256. Это удобно, так как можно считать число в формате byte, затем все единицы представить, как 1, а нули как -1.

Порядковые признаки (6, 7) представляются 4 входными параметрами: низкий, средний, высокий, очень высокий. Если, например, в записи “высокий”, то в входной вектор будут записаны -1, -1, 1, -1.

Номинальный признак (5) представляется 6 входными параметрами: Азия, Европа, Африка, Северная Америка, Южная Америка, Австралия - и представляется во входном векторе, как и порядковые. Для входа “Евразия” будем устанавливать 1 и для “Европа”, и для “Азия” (1, 1, -1, -1, -1, -1).

Бинарные признаки (3, 4) представляются 1 параметром.

Всего получается 3\*8 + 2\*4 + 1\*6 + 2\*1 = 40 параметров.

**Обучающая выборка**

Испания(Риоха) 1 1 Корпорации и семейные винодельни Контроль происхождения и качества вина Европа высокий высокий 3

Италия(Тоскана) 2 2 Семейные винодельни Контроль происхождения и качества вина Европа очень высокий высокий 2

Франция(Бордо) 3 3 Корпорации и семейные винодельни Контроль происхождения и качества вина Европа средний средний 1

США(Калифорния) 4 4 Корпорации и семейные винодельни Географическая индикация Северная Америка высокий очень высокий 4

Аргентина(Мендоса) 5 5 Семейные винодельни Географическая индикация Южная Америка низкий средний 9

Австралия(Южная Австралия) 6 6 Корпорации и семейные винодельни Географическая индикация Австралия высокий высокий 6

Чили(Долина Майпо) 7 7 Корпорации и семейные винодельни Географическая индикация Южная Америка очень высокий высокий 14

Германия(Рейнгау) 8 9 Корпорации и семейные винодельни Географическая индикация Европа высокий очень высокий 8

Португалия(Дору) 9 8 Корпорации и семейные винодельни Контроль происхождения и качества вина Европа высокий очень высокий 10

ЮАР(Кейптаун) 10 10 Корпорации и семейные винодельни Географическая индикация Африка средний средний 7

Новая Зеландия(Марлборо) 11 11 Семейные винодельни Географическая индикация Австралия высокий средний 13

Греция(Санторини) 12 12 Корпорации и семейные винодельни Контроль происхождения и качества вина Европа низкий низкий 19

Хорватия(Истрия) 13 13 Семейные винодельни Географическая индикация Европа средний высокий 25

Мексика(Баха Калифорния) 14 14 Семейные винодельни Контроль происхождения и качества вина Северная Америка средний высокий 38

Бразилия(Серра Гаушо) 15 15 Семейные винодельни Контроль происхождения и качества вина Южная Америка высокий высокий 22

Нидерланды(Лимбург) 16 16 Корпорации и семейные винодельни Географическая индикация Европа высокий высокий 28

Канада(Оканаган) 17 17 Семейные винодельни Контроль происхождения и качества вина Северная Америка средний средний 25

Австрия(Вена) 18 18 Корпорации и семейные винодельни Географическая индикация Европа высокий высокий 15

Швейцария(Вале) 19 19 Корпорации и семейные винодельни Географическая индикация Европа высокий высокий 33

Южная Корея(Чхунчхун) 20 20 Корпорации и семейные винодельни Контроль происхождения и качества вина Азия средний высокий 17

Уругвай(Канэлонес) 21 21 Корпорации и семейные винодельни Географическая индикация Южная Америка высокий низкий 24

Китай(Шандун) 22 22 Корпорации и семейные винодельни Географическая индикация Азия высокий очень высокий 11

Болгария(Тракия) 23 23 Корпорации и семейные винодельни Контроль происхождения и качества вина Европа средний очень высокий 40

Турция(Анталья) 24 24 Корпорации и семейные винодельни Географическая индикация Евразия средний средний 53

Ливан(Бекаа) 25 25 Семейные винодельни Контроль происхождения и качества вина Азия низкий низкий 56

Израиль(Галилея) 26 26 Семейные винодельни Географическая индикация Азия высокий высокий 48

Венгрия(Эгер) 27 27 Корпорации и семейные винодельни Географическая индикация Европа высокий высокий 41

Словения(Горишка Бранда) 28 28 Корпорации и семейные винодельни Корпорации и семейные винодельни Австралия очень высокий очень высокий 42

Румыния(Мурфатлар) 29 29 Корпорации и семейные винодельни Корпорации и семейные винодельни Европа средний средний 12

Молдова(Кодру) 30 30 Корпорации и семейные винодельни Корпорации и семейные винодельни Европа средний средний 67

**Тестирование**

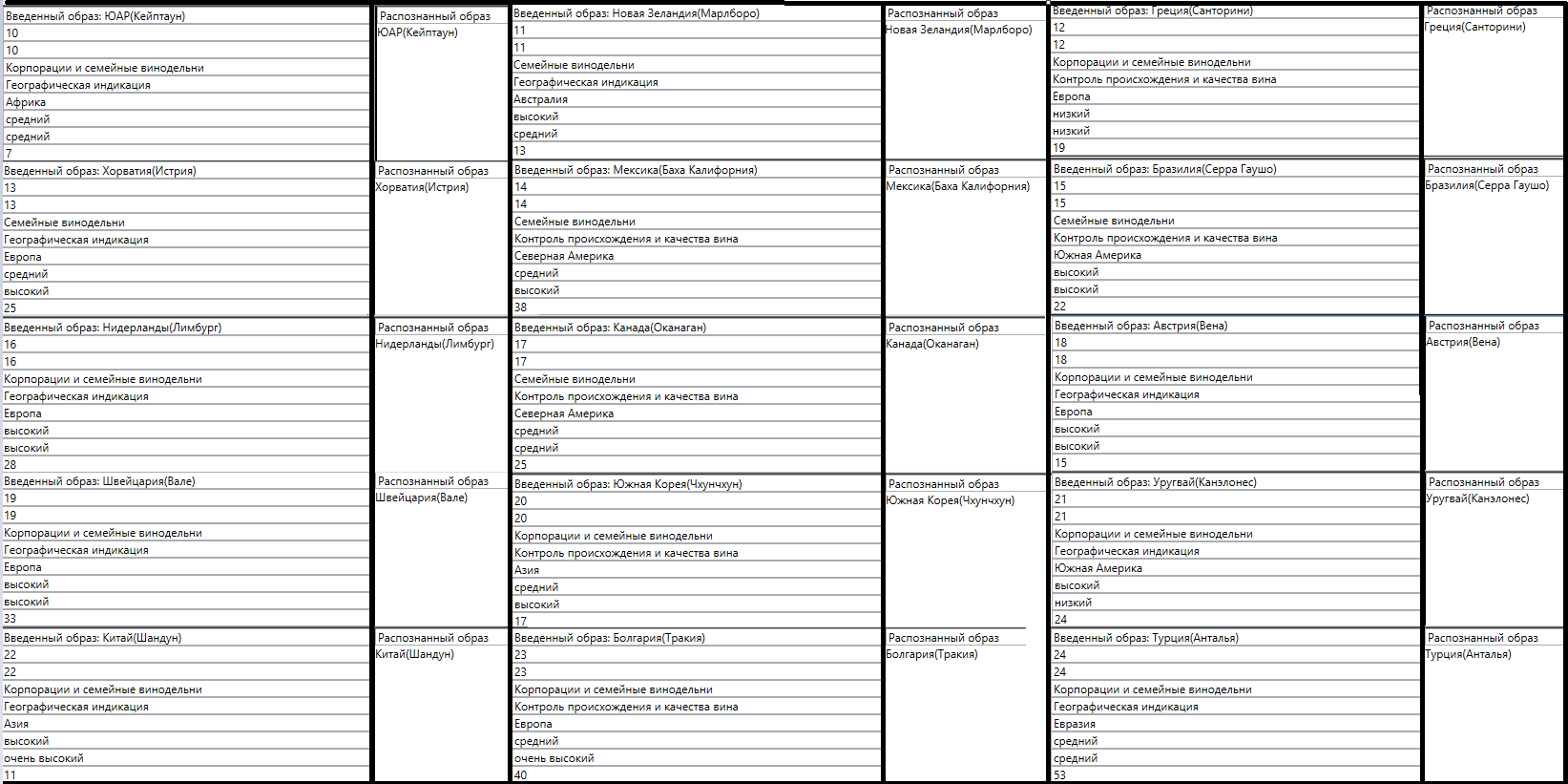
Нейроэмулятор для распознавания текстовых записей был несколько изменен. Во входной вектор преобразуется текстовая строка обучающего файла.

Все 30 образов сеть запомнила (Рисунок 2).

С тестовыми образами, содержащими искажения, сеть хорошо проявила себя (Рисунок 1). Для удобства искажения отмечены красным маркером.



Рисунок 1 - Результат распознавания тестовых текстовых образов



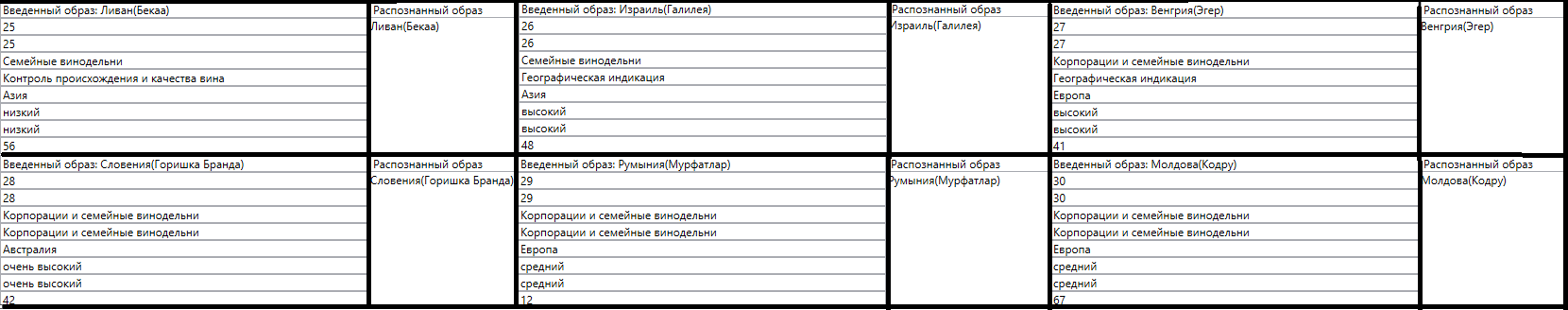


Рисунок 2 - Результат распознавания текстовых образов

**Вывод**

Таким образом, была рассмотрена более совершенная в запоминании сеть Хемминга, которая хорошо показала себя в ходе тестирования в текстовых записях.