Министерство образования Республики Беларусь

Учреждение образования

“Брестский государственный технический университет”

Кафедра ИИТ

**Лабораторная работа №1**

Однослойная искусственная нейронная сеть.

Процедура обучения Розенблатта.

**Выполнил:**

Студент 4 курса

Факультета ЭИС

Группы АС-55

Черноокий И.В.

**Проверил:**

Савицкий Ю.В.

**Брест 2022**

**Лабораторная работа №1**

**Однослойная искусственная нейронная сеть.**

**Процедура обучения Розенблатта**

**Вариант 3**

**Цель работы:** Исследование принципа построения, обучения и функционирования нейронной сети, реализующей однослойный перcептрон Розенблатта, для решения функции AND и OR.

**Задание:**

1. Написать на любом ЯВУ программу моделирования персептрона Розенблатта для линейного разбиения множества точек на два класса.

Варианты заданий приведены в следующей таблице:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № варианта | Точка 1 - (1, 1) | Точка 2 (-1, 1) | Точка 3 (-1, -1) | Точка 4 (1, -1) |
| 3 | -1 | -1 | 1 | -1 |

**Код программы:**

#include <iostream>

#include <iomanip>

using namespace std;

class network

{

private:

double input[4][2]; //входные данные

double e[4]; //эталонные значения

const double a = 0.2; //шаг обучения

double w[2]; //веса

double y = 0; //выходное значение

double T = 0, S = 0; //порог и взв.сумма

public:

network(double in[][2], double et[])

{

for (int i = 0; i < 4; i++)

{

for (int j = 0; j < 2; j++)

input[i][j] = in[i][j];

e[i] = et[i];

}

w[1] = w[0] = double(rand() % 10) / 10; //рандомные веса

}

void learn()

{

int flag[2] = { 0,0 }; //подсчёт эпох и счётчик соответствия эталонам

do {

++flag[0]; //номер эпохи

flag[1] = 0; //сброс соответствия

for (int i = 0; i < 4; i++)

{

S = w[0] \* input[i][0] + w[1] \* input[i][1] - T; //подсчёт взв.суммы

if (S < 0)

y = -1;

else

y = 1;

w[0] -= a \* input[i][0] \* (y - e[i]);

w[1] -= a \* input[i][1] \* (y - e[i]);

T += a \* (y - e[i]);

if (y == e[i])

++flag[1];

cout << setw(2) << right << input[i][0] << setw(2) << input[i][1] << '\t' << "S = " << setw(10) << left << S << "y = " << setw(10) << y << "e = " << setw(10) << e[i] << "w1 = " << setw(10) << w[0] << "w2 = " << setw(10) << w[1] << "T = " << setw(10) << T << endl;

}

cout << endl;

} while (flag[1] != 4);

cout << "END - " << flag[0] << "st" << endl << endl;

}

void func(int x0, int x1)

{

S = w[0] \* x0 + w[1] \* x1 - T;

if (S < 0)

y = -1;

else

y = 1;

cout << setw(2) << right << x0 << setw(2) << x1 << "\ty = " << y << endl;

}

};

int main()

{

double etalon[] = { -1, -1, 1, -1 }; //эталонные значения

double x[4][2] = { {1, 1},{-1,1},{-1,-1},{1,-1} }; //входные данные

network one(x, etalon);

one.learn();

//проверка

one.func(1, 1);

one.func(-1, -1);

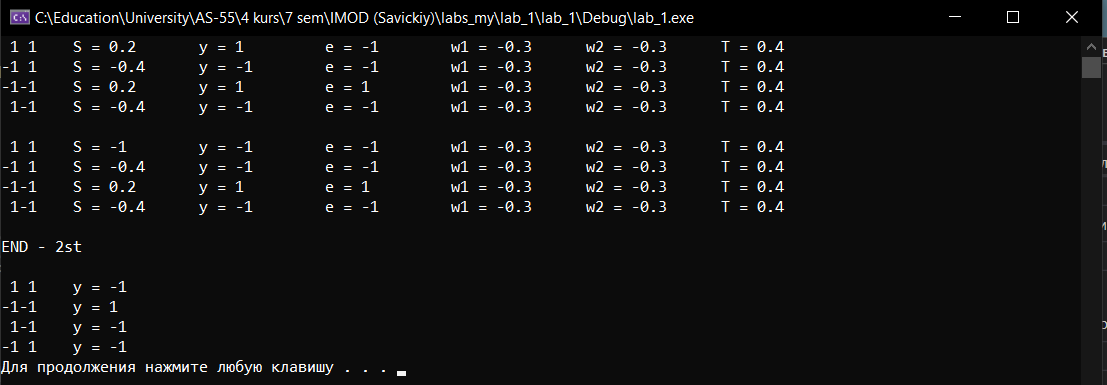
one.func(1, -1);

one.func(-1, 1);

system("pause");

}

**Результат работы программы:**



**Вывод:** был исследован принцип построения, обучения и функционирования нейронной сети, реализующей однослойный перcептрон Розенблатта, для решения функции AND и OR.