Министерство образования Республики Беларусь

Учреждение образования

«Брестский государственный технический университет»

Кафедра ИИТ

Лабораторная работа №5

По дисциплине: «Нелинейные ИНС в задачах распознавания образов»

Выполнил:

Студент 2-го курса

Группы ПО-7

Гункевич И.А.

Проверил:

Крощенко А.А.

Брест 2021

**Цель работы:** Изучить обучение и функционирование нелинейной ИНС при решении задач распознавания образов.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 3 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |
| 2 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| 8 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 |

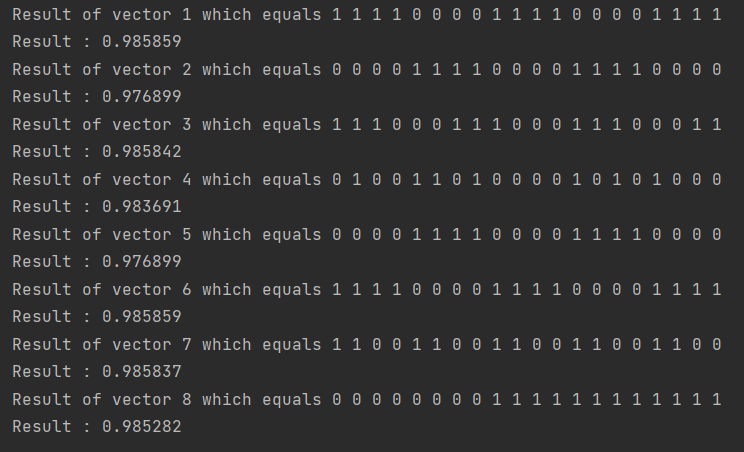
**Задание**

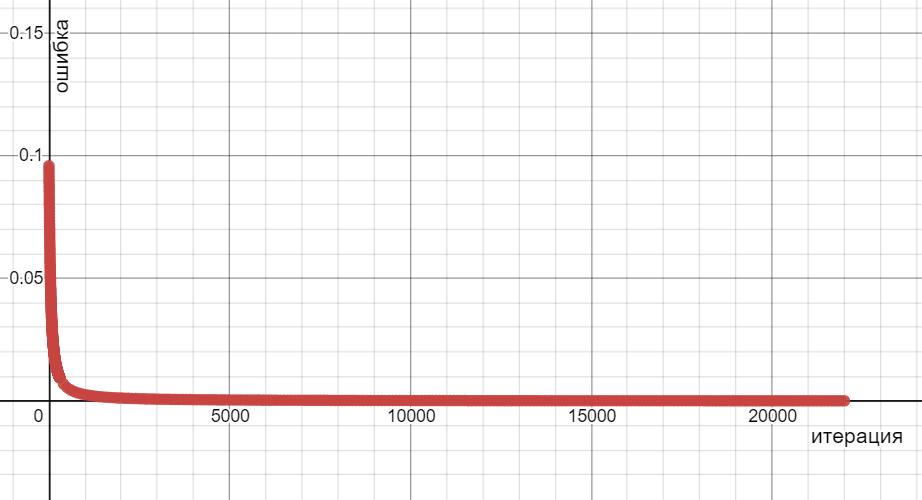
**Код программы:**

#include <iostream>  
#include <math.h>  
#define NUMBER\_IN 10  
#define NUMBER\_HID 4  
#define NUMBER\_OUT 1  
#define e 2.71828  
using namespace std;  
  
double sigmoid(double x) { //Сигмоид  
 return 1 / (1 + pow(e, -x));  
}  
  
double\* get\_hiddens(bool\* Inputs, double w12[NUMBER\_IN][NUMBER\_HID], double T\_Hid[]) {  
 double\* Hiddens = new double[NUMBER\_HID];  
 for (int i = 0; i < NUMBER\_HID; i++) Hiddens[i] = 0;  
 for (int i = 0; i < NUMBER\_HID; i++) {  
 for (int j = 0; j < NUMBER\_IN; j++) {  
 Hiddens[i] += w12[j][i] \* Inputs[j];  
 }  
 Hiddens[i] -= T\_Hid[i];  
 Hiddens[i] = sigmoid(Hiddens[i]);  
 }  
 return Hiddens;  
}  
  
double\* get\_result(bool\* Inputs, double w12[NUMBER\_IN][NUMBER\_HID], double T\_Hid[], double w23[NUMBER\_HID][NUMBER\_OUT], double T\_Out[], double Hiddens[NUMBER\_HID]) {  
 double\* Results = new double[NUMBER\_OUT];  
 for (int i = 0; i < NUMBER\_OUT; i++)  
 Results[i] = 0;  
 for (int j = 0; j < NUMBER\_OUT; j++) {  
 for (int i = 0; i < NUMBER\_HID; i++) {  
 Results[j] += Hiddens[i] \* w23[i][j];  
 }  
 Results[j] -= T\_Out[j];  
 Results[j] = sigmoid(Results[j]);  
 }  
 return Results;  
}  
  
  
int main() {  
 bool Vect1[] = { 1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1};  
 bool Vect2[] = { 0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0 };  
 bool Vect3[] = { 1,1,1,0,0,0,1,1,1,0,0,0,1,1,1,0,0,0,1,1 };  
 bool\* Inputs = new bool[NUMBER\_IN];  
 for (int i = 0; i < NUMBER\_IN; i++) Inputs[i] = 0;  
 bool\*\* Vectors = new bool\*[8];  
 Vectors[0] = Vect1;  
 Vectors[1] = Vect2;  
 Vectors[2] = Vect3;  
 double w12[NUMBER\_IN][NUMBER\_HID], w23[NUMBER\_HID][NUMBER\_OUT], T\_Hid[NUMBER\_HID], T\_Out[NUMBER\_OUT], E\_min = 0.0001, alpha = 0.04, Ethalon, E = 0, Outputs[NUMBER\_OUT] = { 0 };  
 double\* Currents = new double[NUMBER\_OUT];  
 double\* Hiddens = new double[NUMBER\_HID];  
 double Mistakes[NUMBER\_OUT] = { 0 };  
 double Ethalons[NUMBER\_OUT] = { 0 };  
 double MistakesHid[NUMBER\_HID] = { 0 };  
 int Iter = 1; //KOLVO ITERACII POKA 4to 1  
 for (int i = 0; i < NUMBER\_IN; i++) {  
 for (int j = 0; j < NUMBER\_HID; j++) {  
 w12[i][j] = ((double)rand() / (RAND\_MAX)) - 0.5;  
 for (int k = 0; k < NUMBER\_OUT; k++) {  
 w23[j][k] = ((double)rand() / (RAND\_MAX)) - 0.5;  
 T\_Out[k] = ((double)rand() / (RAND\_MAX)) - 0.5;  
 }  
 T\_Hid[j] = ((double)rand() / (RAND\_MAX)) - 0.5;  
 }  
 }  
 int H = 0;  
 do {  
 E = 0;  
 for (int N = 0; N < NUMBER\_OUT; N++) {  
 Ethalons[0] = 0;  
 Ethalons[N] = 1;  
 for (int q = 0; q < Iter; q++) { //1 iteracia poka 4to  
 Inputs = Vectors[N];  
 Hiddens = get\_hiddens(Inputs, w12, T\_Hid);  
 Currents = get\_result(Inputs, w12, T\_Hid, w23, T\_Out, Hiddens);  
 for (int i = 0; i < NUMBER\_OUT; i++)  
 Mistakes[i] = Currents[i] - Ethalons[i];  
 for (int j = 0; j < NUMBER\_HID; j++) {  
 for (int m = 0; m < NUMBER\_OUT; m++) {  
 MistakesHid[j] += Mistakes[m] \* Currents[m] \* (1 -Currents[m]) \* w23[j][m];  
 }  
 }  
 for (int j = 0; j < NUMBER\_OUT; j++) {  
 for (int i = 0; i < NUMBER\_HID; i++) {  
 w23[i][j] -= alpha \* Mistakes[j] \* Currents[j] \* (1 - Currents[j]) \* Hiddens[i];  
 }  
 T\_Out[j] += alpha \* Mistakes[j] \* Currents[j] \* (1 - Currents[j]);  
 }  
 for (int j = 0; j < NUMBER\_HID; j++) {  
 for (int i = 0; i < NUMBER\_IN; i++) {  
 w12[i][j] -= alpha \* MistakesHid[j] \* Hiddens[j] \* (1 - Hiddens[j]) \* Inputs[i];  
 }  
 T\_Hid[j] += alpha \* MistakesHid[j] \* Hiddens[j] \* (1 - Hiddens[j]);  
 }  
 E += pow(Mistakes[N], 2);  
 }  
 }  
 E /= 2;  
 if (H % 100 == 0 || H<300) {  
 cout <<H<<";"<< E << endl;  
 }  
 H++;  
  
 } while (E > E\_min);  
  
 //Predictions:  
 double\* HiddenPred;  
 double\* Values;  
 bool Vectors3[] = { 0,1,0,0,1,1,0,1,0,0,0,0,1,0,1,0,1,0,0,0 };  
bool Vectors4[] = { 0,0,0,0,1,1,1,1,0,0,0,0,1,1,1,1,0,0,0,0 };  
bool Vectors5[] = { 1,1,1,1,0,0,0,0,1,1,1,1,0,0,0,0,1,1,1,1 };  
bool Vectors6[] = { 1,1,0,0,1,1,0,0,1,1,0,0,1,1,0,0,1,1,0,0 };  
bool Vectors7[] = { 0,0,0,0,0,0,0,0,1,1,1,1,1,1,1,1,1,1,1,1 };  
Vectors[3] = Vectors3;  
Vectors[4] = Vectors4;  
Vectors[5] = Vectors5;  
Vectors[6] = Vectors6;  
Vectors[7] = Vectors7;  
  
for (int i = 0; i < 8; i++) {  
Inputs = Vectors[i];  
cout << "Result of vector " << i + 1 << " which equals ";  
for (int j = 0; j < 20; j++) {  
cout << Inputs[j] << ' ';  
}  
cout << endl << "Result : ";  
HiddenPred = get\_hiddens(Inputs, w12, T\_Hid);  
Values = get\_result(Inputs, w12, T\_Hid, w23, T\_Out, HiddenPred);  
cout << Values[0] << endl;  
}

}

**Результат:**





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