Министерство образования Республики Беларусь

Учреждение образования

«Брестский государственный технический университет»

Кафедра ИИТ

**Лабораторная работа №3**

По дисциплине: “МиАПР”

Тема: “Нелинейные ИНС в задачах прогнозирования”

Вариант 4

**Выполнил:**

Студент 2 курса

Группы ПО-7

Пищанюк В.Ю.

**Проверил:**

Крощенко А.А.

**Брест, 2021**

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

**Задача:** Написать на любом ЯВУ программу моделирования прогнозирующей нелинейной ИНС. Для тестирования использовать функцию

a = 0.4; b = 0.4; c = 0.08; d = 0.4;

Количество входов ИНС = 6;

Количество нейронных элементов в скрытом слое = 2;

y= a\*cos(bx) + c\*sin(dx)

Для прогнозирования использовать многослойную ИНС с одним скрытым слоем. В качестве функций активации для скрытого слоя использовать сигмоидную функцию, для выходного - линейную.

**Код программы:**

#include <iostream>

#include <math.h>

#include <iomanip>

using namespace std;

double Func(double x);

double Sigm\_Func(double x);

double\* Hidden(double x, double w1[2][6], double T[2]);

double Print(double x, double w1[2][6], double w2[2], double T[2 + 1]);

int main()

{

setlocale(LC\_ALL, "rus");

double w1[2][6],

w2[2],

T[2 + 1],

main,

current,

V = 0.4,

x = 4,

Emin = 0.00002,

Emax = 0;

int era = 0,

max\_era = 1200;

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

{

for (int k = 0; k < 6; k++)

{

w1[i][k] = ((double)rand() / RAND\_MAX) \* 0.005;

}

w2[i] = ((double)rand() / RAND\_MAX) \* 0.005;

T[i] = ((double)rand() / RAND\_MAX) \* 0.005;

}

T[4] = ((double)rand() / RAND\_MAX) \* 0.005;

do

{

Emax = 0;

for (int q = 0; q < 300; q++)

{

current = Print(x, w1, w2, T);

main = Func(x + 6 \* 0.1);

double err = current - main;

double\* hiddens = Hidden(x, w1, T);

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

w2[j] -= V \* err \* hiddens[j];

T[4] += V \* err;

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

{

for (int i = 0; i < 6; i++)

w1[k][i] -= V \* Func(x + i \* 0.1) \* hiddens[k] \* (1 - hiddens[k]) \* w2[k] \* err;

T[k] += V \* hiddens[k] \* (1 - hiddens[k]) \* w2[k] \* err;

}

x += 0.1;

Emax += pow(err, 2);

}

Emax /= 2;

era++;

} while (Emax > Emin);

cout << "Emax= : " << Emax << endl;

cout << "Number of epochs: " << era << endl;

cout << setw(27) << left << "Reference values" << setw(29) << left << "The resulting values" << setw(30) << left << "Deviation" << endl;

for (int i = 0; i < 100; i++)

{

double Resultat = Print(x, w1, w2, T), Etalon = Func(x + 6 \* 0.1);

cout << setw(27) << left << Etalon << setw(27) << left << Resultat << setw(30) << Resultat - Etalon << endl;

x += 0.1;

}

system("pause");

return 0;

}

double Func(double x) {

return 0.4 \* cos(0.4 \* x) + 0.08 \* sin(0.4 \* x);

}

double Sigm\_Func(double x) {

return 1 / (1 + pow(2, -x));

}

double\* Hidden(double x, double w1[2][6], double T[2]) {

double\* polychennoe = new double[2];

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

polychennoe[i] = 0;

double input\_neuron[6];

for (int k = 0; k < 6; k++, x += 0.1)

input\_neuron[k] = Func(x);

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

{

for (int k = 0; k < 6; k++)

polychennoe[i] += input\_neuron[k] \* w1[i][k];

polychennoe[i] -= T[i];

polychennoe[i] = Sigm\_Func(polychennoe[i]);

}

return polychennoe;

}

double Print(double x, double w1[2][6], double w2[2], double T[2 + 1])

{

double Resultat = 0;

double\* hidden\_neuron = Hidden(x, w1, T);

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

Resultat += hidden\_neuron[j] \* w2[j];

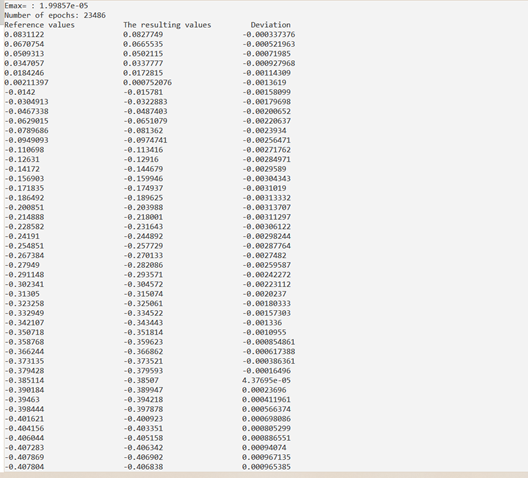
}

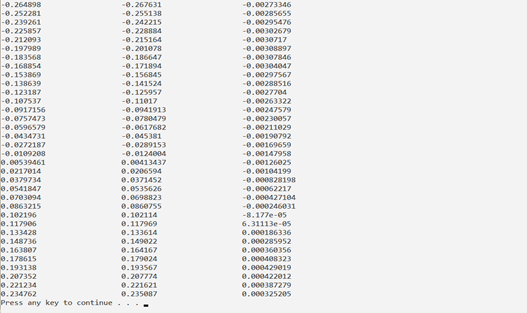
Resultat -= T[4];

return Resultat;

}

**Результат работы программы:**





**Вывод:** В данной лабораторной работе я изучил обучение и функционирование нелинейной ИНС при решении задач прогнозирования.