Министерство образования Республики Беларусь

Учреждение образования

«Брестский государственный технический университет»

Кафедра ИИТ

**Лабораторная работа №4**

По дисциплине: «Методы и алгоритмы принятия решений»

Тема: «Нелинейные ИНС в задачах прогнозирования»

Выполнил:

студент 2 курса

группы ПО-4

Синяк Д.А

Проверил:

Крощенко А.А.

2020

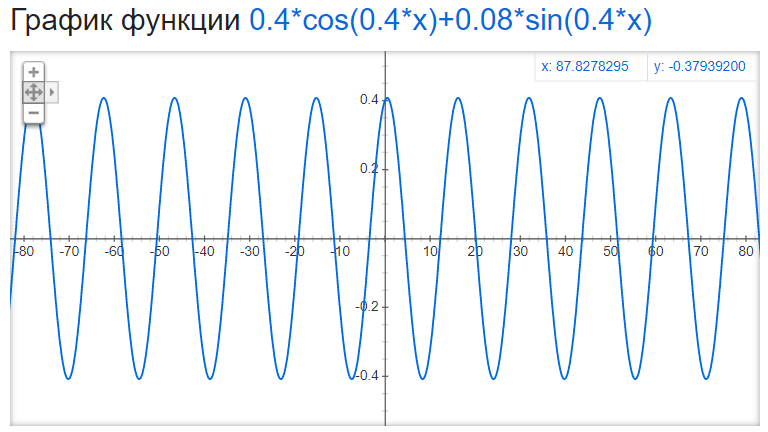
**Лабораторная работа №4**

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

**Задание:** Спрогнозировать нелинейный временной ряд, применяя параметры лабораторной работы №3. При этом необходимо использовать алгоритм обучения многослойной ИНС с адаптивным шагом

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| № Варианта | a | b | c | d | Кол-во входов ИНС | Кол-во НЭ в скрытом слое |
| 8 | 0.4 | 0.4 | 0.08 | 0.4 | 6 | 2 |

Функция 0.4 \* cos(0.4 \* x) + 0.08 \* sin(0.4 \* x)



Код программы:

#include <iostream>

#include <math.h>

#include <iomanip>

using namespace std;

double sigmoid(double x)

{

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

}

double function(double x)

{

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

}

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

{

double\* Resultatik = new double[2];

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

Resultatik[i] = 0;

double Inputs[6];

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

Inputs[k] = function(x);

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

{

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

Resultatik[i] += Inputs[k] \* w1[i][k];

Resultatik[i] -= T[i];

Resultatik[i] = sigmoid(Resultatik[i]);

}

return Resultatik;

}

double get\_alpha(double w2[], double Error, double Output, double Hiddens[])

{

double alpha = 0, A = 0, B = 0;

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

{

A += pow(Error \* w2[i] \* (1 - Hiddens[i]) \* Hiddens[i], 2) \* Hiddens[i] \* (1 - Hiddens[i]);

B += pow(Error \* w2[i] \* (1 - Hiddens[i]) \* Hiddens[i], 2) \* Hiddens[i] \* Hiddens[i] \* (1 - Hiddens[i]) \* (1 - Hiddens[i]);

}

alpha = 4 \* A / (B \* (1 + Output \* Output));

return alpha;

}

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

{

double Resultat = 0;

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

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

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

}

Resultat -= T[4];

return Resultat;

}

int main()

{

int epoch = 0;

setlocale(LC\_ALL, "rus");

double w1[2][6], w2[2], T[2 + 1], Reference, E\_min = 0.00002, alpha2 = 0.4, alpha = 0.4, x = 4, current, E = 0;

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

{

E = 0;

for (int q = 0; q < 600; q++) /// Обучающая выборка, изменял от 100 до 1000, самый минимум эпох в 600, начиная с 700 начинают эпохи расти

{

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

Reference = function(x + 6 \* 0.1);

double oshibka = current - Reference;

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

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

w2[j] -= alpha \* oshibka \* Hiddens[j];

T[4] += alpha \* oshibka;

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

{

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

w1[k][i] -= alpha2 \* function(x + i \* 0.1) \* Hiddens[k] \* (1 - Hiddens[k]) \* w2[k] \* oshibka;

T[k] += alpha2 \* Hiddens[k] \* (1 - Hiddens[k]) \* w2[k] \* oshibka;

}

alpha2 = get\_alpha(w2, oshibka, current, Hiddens);

x += 0.1;

E += pow(oshibka, 2);

}

E /= 2;

cout << "Oshibka: " << E << endl;

epoch++;

} while (E > E\_min);

cout << "Количество эпох " << epoch << "\n";

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

{

double Resultat = output(x, w1, w2, T), Etalonn = function(x + 6 \* 0.1);

cout << "Etalonnoe znachenie: " << fixed << setprecision(5) << Etalonn << " Poluchennoe znachenie: " << Resultat << " Otklonenie: " << Resultat - Etalonn << endl;

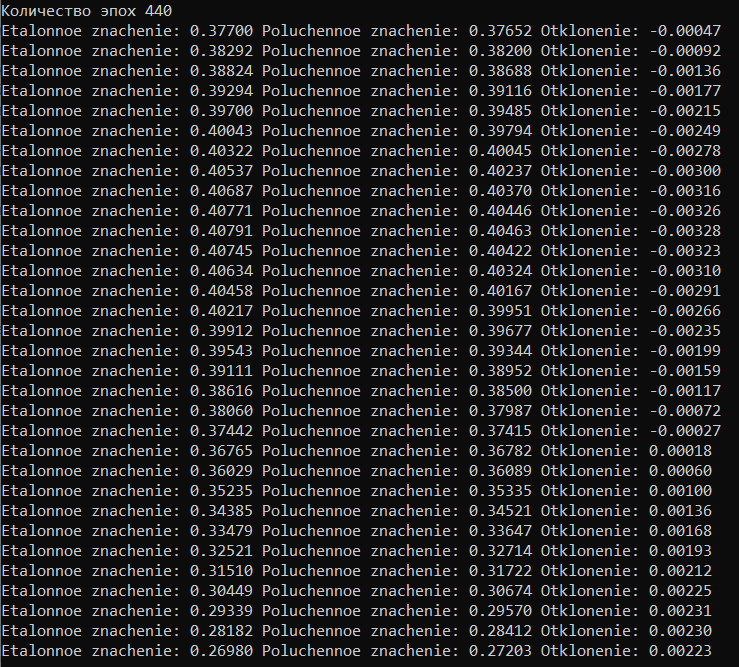
x += 0.1;

}

system("pause");

}

Результат выполнения работы:

****

