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

Маевский А.В.

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

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| вариант | a | b | с | d | Кол-во входов ИНС | Кол-во НЭ в скрытом слое |
| 6 | 0.2 | 0.6 | 0.05 | 0.6 | 10 | 4 |

Функция 0.2 \* cos(0.6 \* x) + 0.05 \* sin(0.6 \* 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 0.2 \* cos(0.6 \* x) + 0.05 \* sin(0.6 \* x);

}

double\* hidden(double x, double w1[4][10], double T[4])

{

double\* result = new double[4];

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

result[i] = 0;

double Inputs[10];

for (int k = 0; k < 10; k++, x += 0.1)

Inputs[k] = function(x);

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

{

for (int k = 0; k < 10; k++)

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

result[i] -= T[i];

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

}

return result;

}

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

{

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

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

{

A += pow(Error \* w2[i] \* (1 - Hiddens[i]) \* Hiddens[i], 4) \* Hiddens[i] \* (1 - Hiddens[i]);

B += pow(Error \* w2[i] \* (1 - Hiddens[i]) \* Hiddens[i], 4) \* 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[4][10], double w2[4], double T[4 + 1])

{

double Result = 0;

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

for (int j = 0; j < 4; j++) {

Result += hidden\_result[j] \* w2[j];

}

Result -= T[4];

return Result;

}

int main()

{

setlocale(LC\_ALL, "rus");

double w1[4][10], w2[4], T[4 + 1], Reference, E\_min = 0.00004, alpha4 = 0.4, alpha = 0.4, x = 4, current, E = 0;

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

{

for (int k = 0; k < 10; 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 < 400; q++)

{

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

Reference = function(x + 10 \* 0.1);

double error = current - Reference;

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

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

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

T[4] += alpha \* error;

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

{

for (int i = 0; i < 10; i++)

w1[k][i] -= alpha4 \* function(x + i \* 0.1) \* Hiddens[k] \* (1 - Hiddens[k]) \* w2[k] \* error;

T[k] += alpha4 \* Hiddens[k] \* (1 - Hiddens[k]) \* w2[k] \* error;

}

alpha4 = get\_alpha(w2, error, current, Hiddens);

x += 0.1;

E += pow(error, 2);

}

} while (E > E\_min);

cout << "Эталоное значение " << setw(23) << "Полученное значение " << setw(20) << "Отклонение: " << endl;

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

{

double Result = output(x, w1, w2, T), Etalonn = function(x + 10 \* 0.1);

cout << fixed << setprecision(5) << Etalonn << setw(21) << Result << setw(29) << Result - Etalonn << endl;

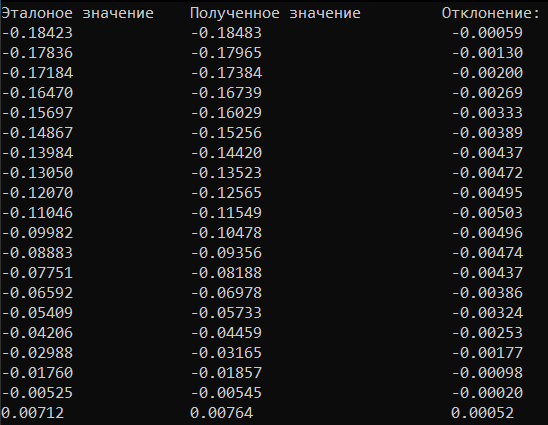
x += 0.1;

}

system("pause");

}

Результат выполнения работы:

****

Вывод: изучил обучение и функционирование нелинейной ИНС при решении задач прогнозирования.