Министерство образования Республики Беларусь

Учреждение образования

«Брестский государственный технический университет»

Кафедра ИИТ

Лабораторная работа №4

за 1 семестр

По дисциплине: «МиАПР»

Тема: «Нелинейные ИНС в задачах прогнозирования. Адаптивный шаг»

Выполнил:

Студент 2 курса

Группы ПО-4(1)

Воробей Анастасия Петровна

Проверил:

Крощенко Александр Александрович

Брест 2020

Лабораторная работа №4

Нелинейные ИНС в задачах прогнозирования. Адаптивный шаг

Цель работы: Изучить обучение и функционирование нелинейной ИНС при решении задач прогнозирования с использованием адаптивного шага обучения.

Вариант 4

Задание:

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

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| № варианта | a | b | с | d | Кол-во входов ИНС | Кол-во НЭ в скрытом слое |  |
| 4 | 0.4 | 0.4 | 0.08 | 0.4 | 6 | 2 |  |

Код программы:

#include <iostream>

#include <math.h>

#include <iomanip>

using namespace std;

double Sigmoid(double x) {

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

}

double func(double x) {

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

return a \* cos(b \* x) + c \* sin(d \* x);

}

double\* hidden(double x, double Wes1[2][6], double T[2]) {

double\* Result = new double[2];

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

Result[i] = 0;

}

double entrances[6];

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

entrances[k] = func(x);

}

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

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

Result[i] += entrances[k] \* Wes1[i][k];

}

Result[i] -= T[i];

Result[i] = Sigmoid(Result[i]);

}

return Result;

}

double adapt(double \*Wes2, double Error, double Output, double \*Hiddens) {

double Alpha = 0, A = 0, B = 0;

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

A += pow(Error \* Wes2[i] \* (1 - Hiddens[i]) \* Hiddens[i], 2) \* Hiddens[i] \* (1 - Hiddens[i]);

B += pow(Error \* Wes2[i] \* (1 - Hiddens[i]) \* Hiddens[i], 2) \* Hiddens[i] \* Hiddens[i] \* (1 - Hiddens[i]) \* (1 - Hiddens[i]);

}

Alpha = 4 \* A / (B \* (1 + Output \* Output)); //24 formula

return Alpha;

}

double output(double x, double Wes1[2][6], double Wes2[2], double T[2 + 1]) {

double Resultat = 0;

double\* hidden\_Result = hidden(x, Wes1, T);

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

Resultat += hidden\_Result[j] \* Wes2[j];

}

Resultat -= T[4];

return Resultat;

}

int main() {

setlocale(0, "");

double Wes1[2][6], Wes2[2], T[2 + 1], reference\_value, 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++) {

Wes1[i][k] = ((double)rand() / RAND\_MAX) \* 0.005;

}

Wes2[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 < 200; q++) {

current = output(x, Wes1, Wes2, T);

reference\_value = func(x + 6 \* 0.1);

double error = current - reference\_value;

double\* Hiddens = hidden(x, Wes1, T);

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

Wes2[j] -= Alpha \* error \* Hiddens[j];

}

T[4] += Alpha \* error;

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

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

Wes1[k][i] -= Alpha2 \* func(x + i \* 0.1) \* Hiddens[k] \* (1 - Hiddens[k]) \* Wes2[k] \* error;

}

T[k] += Alpha2 \* Hiddens[k] \* (1 - Hiddens[k]) \* Wes2[k] \* error;

}

Alpha2 = adapt(Wes2, error, current, Hiddens);

x += 0.1;

E += pow(error, 2);

}

E /= 2;

cout << "\rError: " << E;

} while (E > E\_min);

cout << endl;

for (int i = 0; i < 30; i++) {

double result = output(x, Wes1, Wes2, T),

ethelon\_value = func(x + 6 \* 0.1);

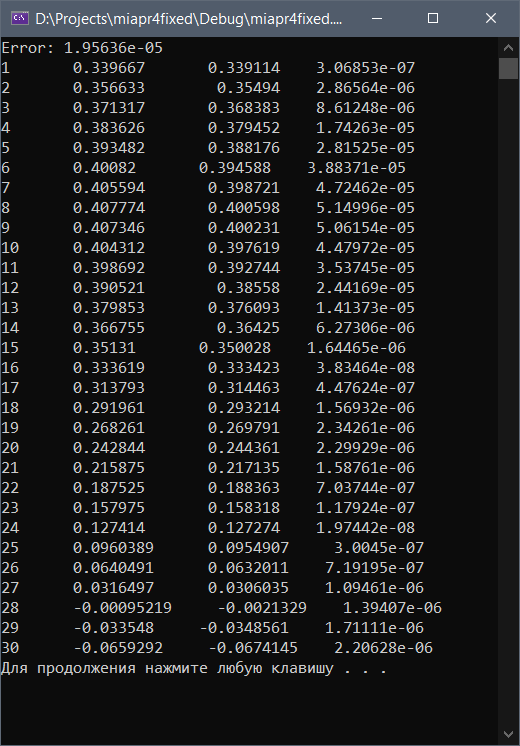
cout << i + 1 << "\t" << ethelon\_value << right << setw(15) << result << right << setw(15) << (result - ethelon\_value)\*(result - ethelon\_value)) << endl;

x += 0.2;

}

system("pause");

}



Вывод: Изучила обучение и функционирование нелинейной ИНС при решении задач прогнозирования с использованием адаптивного шага обучения.