Министерство образование Республики Беларусь

Учреждение образования

"Брестский государственный технический университет"  
Кафедра ИИТ

Лабораторная работа №2

За 1 семестр  
по дисциплине: «МиАПР»

Тема: «Линейная искусственная нейронная сеть. Адаптивный шаг обучения»

Подготовил:

Студент 1 курса

Группы ПО-4(2)

Тупик Д. Л.

Проверил:

Крощенко А. А

Брест, 2020

**Лабораторная работа №2**

**Тема:** «Линейная искусственная нейронная сеть. Адаптивный шаг обучения»

**Цель:** изучить обучение и функционирование линейной ИНС с применением адаптивного шага.

**Ход работы:**

**Вариант 9**

Модифицировать программу из лабораторной работы №1, используя правило адаптивного шага обучения. Произвести исследование получившейся модели ИНС на задачах прогнозирования, согласно варианту лабораторной работы №1.

**Условие:** Написать на любом ЯВУ программу моделирования прогнозирующей линейной ИНС. Для тестирования использовать функцию

y = a\*sin(bx) + d

a = 1, b = 8, d = 0.3, кол-во входов ИНС = 5.

Обучение и прогнозирование производить на 30 и 15 значениях соответственно табулируя функцию с шагом 0.1. Скорость обучения выбирается студентом самостоятельно, для чего моделирование проводится несколько раз для разных α. Результаты оцениваются по двум критериям - скорости обучения и минимальной достигнутой ошибке. Необходимо заметить, что эти критерии в общем случае являются взаимоисключающими, и оптимальные значения для каждого критерия достигаются при разных α.

**Решение:** Составим код программы

//вариант 9

#include <iostream>

#include <Windows.h>

#include <iomanip>

#include <ctime>

#include <cmath>

using namespace std;

float function(int a, int b, float x, float d);

void print\_result(int n, float T, float\* Y, int n\_learn, int N, int n\_protected, float\* W);

int main() {

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

system("color f0");

srand(time(0));

int a = 1, b = 8, n = 5, n\_learn = 30, n\_predicted = 15;

float d = 0.3, step = 0.1, x = 0, E, Ee = 0.01, T = 1;

float\* W = new float[n];

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

W[i] = rand() % 100 \* 0.1;

}

int N = n\_learn + n\_predicted;

float\* Y = new float[n\_learn + n\_predicted];

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

x = x + step;

Y[i] = function(a, b, x, d);

}

int eras = 0;

do {

double y1, //выходное значение нейронной сети

alpha = 0.3; //скорость обучения

E = 0;

for (int i = 0; i < n\_learn - n; i++) {

double temp = 0;

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

temp += pow(Y[i + j], 2);

}

alpha = 1 / (1 + temp); //адаптивный шаг

y1 = 0;

for (int j = 0; j < n; j++) { //векторы выходной активности сети

y1 += W[j] \* Y[i + j];

}

y1 -= T;

for (int j = 0; j < n; j++) { //изменение весовых коэффициентов

W[j] -= alpha \* (y1 - Y[i + n]) \* Y[i + j];

}

T += alpha \* (y1 - Y[i + n]); //изменение порога нейронной сети

E += 0.5 \* pow(y1 - Y[i + n], 2); //расчет суммарной среднеквадратичной ошибки

}

cout << eras << "\t" << E << endl;

eras++;

} while (E > Ee);

print\_result(n, T, Y, n\_learn, N, n\_predicted, W);

delete[] Y;

delete[] W;

system("pause");

return 0;

}

float function(int a, int b, float x, float d) {

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

}

void print\_result(int n, float T, float\* Y, int n\_learn, int N, int n\_predicted, float\* W) {

cout << "РЕЗУЛЬТАТЫ:" << endl;

cout << "1) Обучение:" << endl;

cout << setw(30) << left << "Эталонные значения" << setw(30) << left << "Полученные значения" << "Отклонение" << endl;

float\* prediction = new float[N];

for (int i = 0; i < n\_learn; i++) {

prediction[i] = 0;

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

prediction[i] += W[j] \* Y[j + i];

}

prediction[i] -= T;

cout << "y[" << i << "] = " << setw(30) << left << Y[i + n] << setw(30) << left << prediction[i] << pow((Y[i + n] - prediction[i]), 2) << endl;

}

cout << "2) Прогнозирование:" << endl;

cout << setw(30) << left << "Эталонные значения" << setw(30) << left << "Полученные значения" << "Отклонение" << endl;

for (int i = 0; i < n\_predicted; i++) {

prediction[i + n\_learn] = 0;

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

prediction[i + n\_learn] += W[j] \* Y[i + j + n\_learn - n];

}

prediction[i + n\_learn] -= T;

cout << "y[" << i + n\_learn << "] = " << setw(30) << left << Y[i + n\_learn] << setw(30) << left << prediction[i + n\_learn] <<

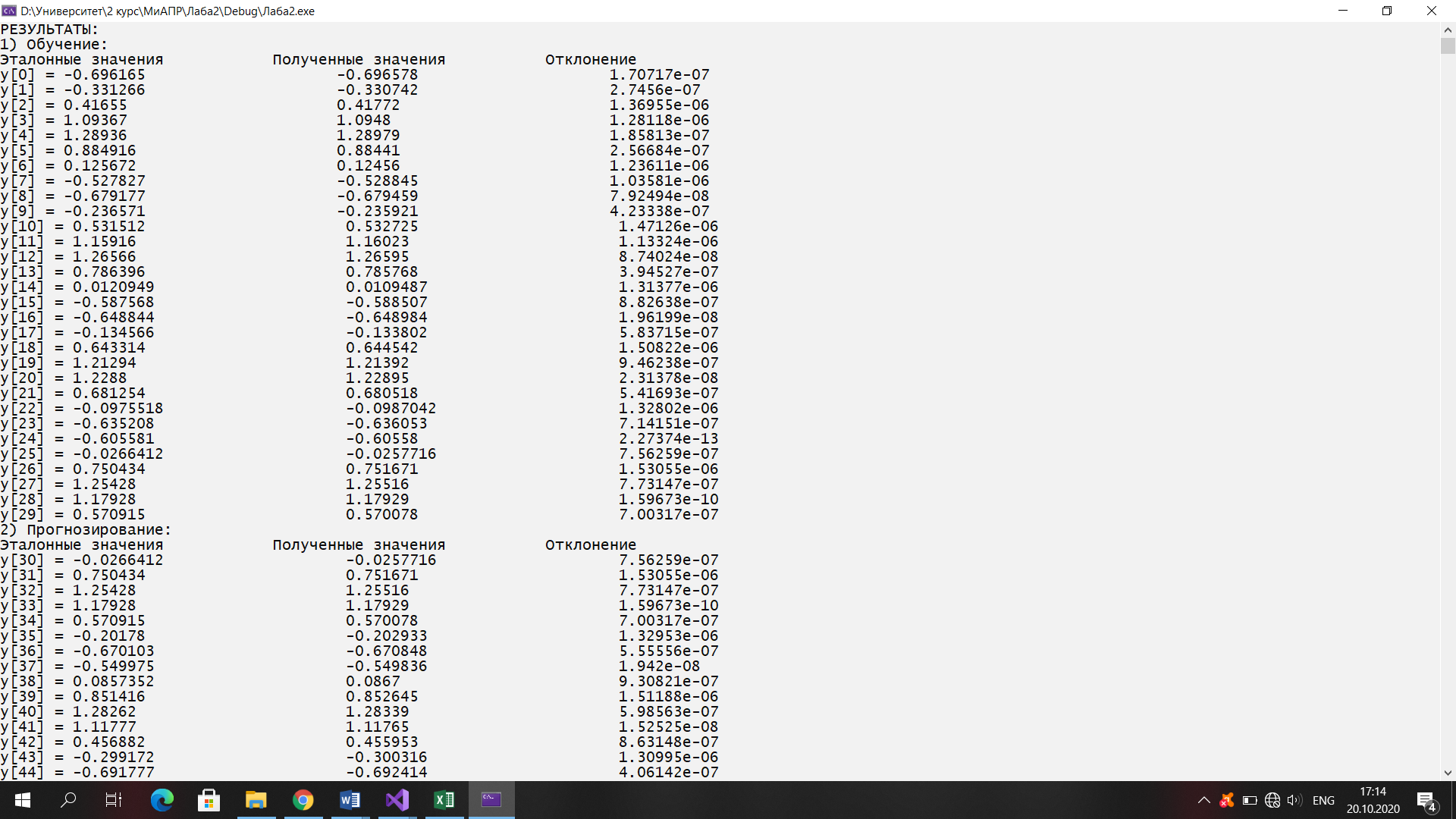
pow((Y[i + n\_learn] - prediction[i + n\_learn]), 2) << endl;

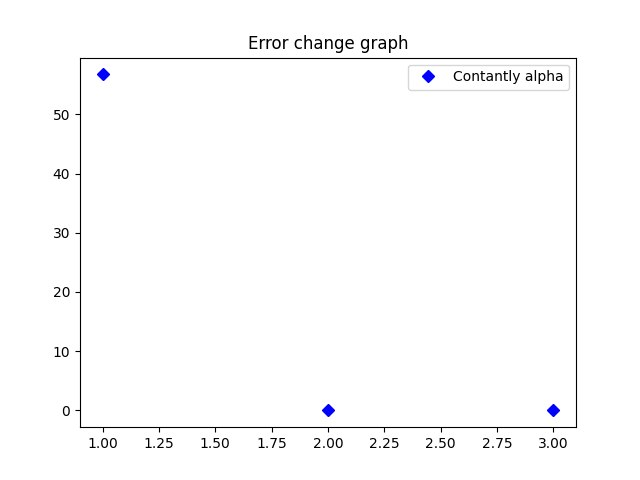
}

delete[] prediction;

}

5. Выполнение программы





**Вывод:** в ходе лабораторной работы, изучил обучение и функционирование линейной ИНС с применением адаптивного шага.