Министерство образования Республики Беларусь

Учреждение образования

“Брестский государственный технический университет”

**Лабораторная работа №1**

**По дисциплине МиАПР за 3 семестр**  
**Тема: «Линейная искусственная нейронная сеть.**

**Правило обучения Видроу-Хоффа»**

**Выполнил:**

Студент группы ПО-6(1)  
 2-го курса

Мартынович Даниил

**Проверил:**

Михняев А. Л.

Брест 2021

Цель работы: изучить обучение и функционирование линейной ИНС при решении задач прогнозирования.

Вариант 2

Задание:

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

y = a\*sin(bx) + d

a = 2, b = 6, d = 0.2, кол-во входов ИНС = 4.

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.

Заголовочный файл Header.h

#pragma once

#ifndef LAB

#define LAB

#include <iostream>

#include<time.h>

#include <iomanip>

using namespace std;

void Print();

void set\_Weight(double\* W);

void set\_Y(double\* y);

int a = 2,b=6,num\_enter = 4,num\_study = 30,num\_forecast = 15;//Кол-во входов, кол-во элементов обучения, кол-во прогнозтрование

double d = 0.1;

double E;

double Emin = 0.01, T = 1;//минимальная погрешность, порог

#endif

Cpp – файл main.cpp

#include "Header.h"

int main()

{

srand(time(0));

setlocale(LC\_ALL, "RU");

double\* W = new double[num\_enter];//весовые коэффициенты

set\_Weight(W);

double\* y = new double[num\_study];//эталонные(подсчет функции)

set\_Y(y);

do {//обучение нейронки

double y1, //выходное значение нейронной сети

A = 0.00015; //скорость обучения

E = 0;

for (int i = 0; i < num\_study - num\_enter; i++) {

y1 = 0;

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

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

}

y1 =y1-T;

for (int j = 0; j < num\_enter; j++) { //изменение весовых коэффициентов

W[j] =W[j]- A \* (y1 - y[i + num\_enter]) \* y[i + j];

}

T =T + A \* (y1 - y[i + num\_enter]); //изменение порога нейронной сети

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

}

} while (E > Emin);

Print();

double\* forecast\_values = new double[num\_study + num\_forecast];//прогнозирование нейронки

for (int i = 0; i < num\_study; i++) {

forecast\_values[i] = 0;

for (int j = 0; j < num\_enter; j++) {

forecast\_values[i] =forecast\_values[i] + W[j] \* y[j + i - num\_enter];

}

forecast\_values[i] =forecast\_values[i] - T;

cout << "y[" << i << "] = " << setw(20) << left << y[i] << setw(23) << left;

cout << forecast\_values[i] << y[i] - forecast\_values[i] << endl;

}

Print();

for (int i = 0; i < num\_forecast; i++)

{

forecast\_values[i + num\_study] = 0;

for (int j = 0; j < num\_enter; j++) {

//прогнозируемые значения

forecast\_values[i + num\_study] = forecast\_values[i + num\_study] + W[j] \* y[num\_study - num\_enter + j + i];

}

forecast\_values[i + num\_study] = forecast\_values[i + num\_study] + T;

cout << "y[" << num\_study + i << "] = " << setw(20) << left << y[i + num\_study] << setw(23) << left;

cout << forecast\_values[i + num\_study] << y[i + num\_study] - forecast\_values[i + num\_study] << endl;

}

delete[]y;

delete[]forecast\_values;

delete[]W;

system("pause");

return 0;

}

void Print()

{

cout << endl;

cout << setw(28) << left << "Эталонные значения" << setw(23) << left << "Полученные значения" << "Отклонение" << endl;

}

void set\_Weight(double \*W)

{

for (size\_t i = 0; i < num\_enter; i++)

{

W[i] = (rand()) / (RAND\_MAX / 10); //весовые значение

}

}

void set\_Y(double\* y)

{

for (size\_t i = 0; i < num\_study ; i++)

{

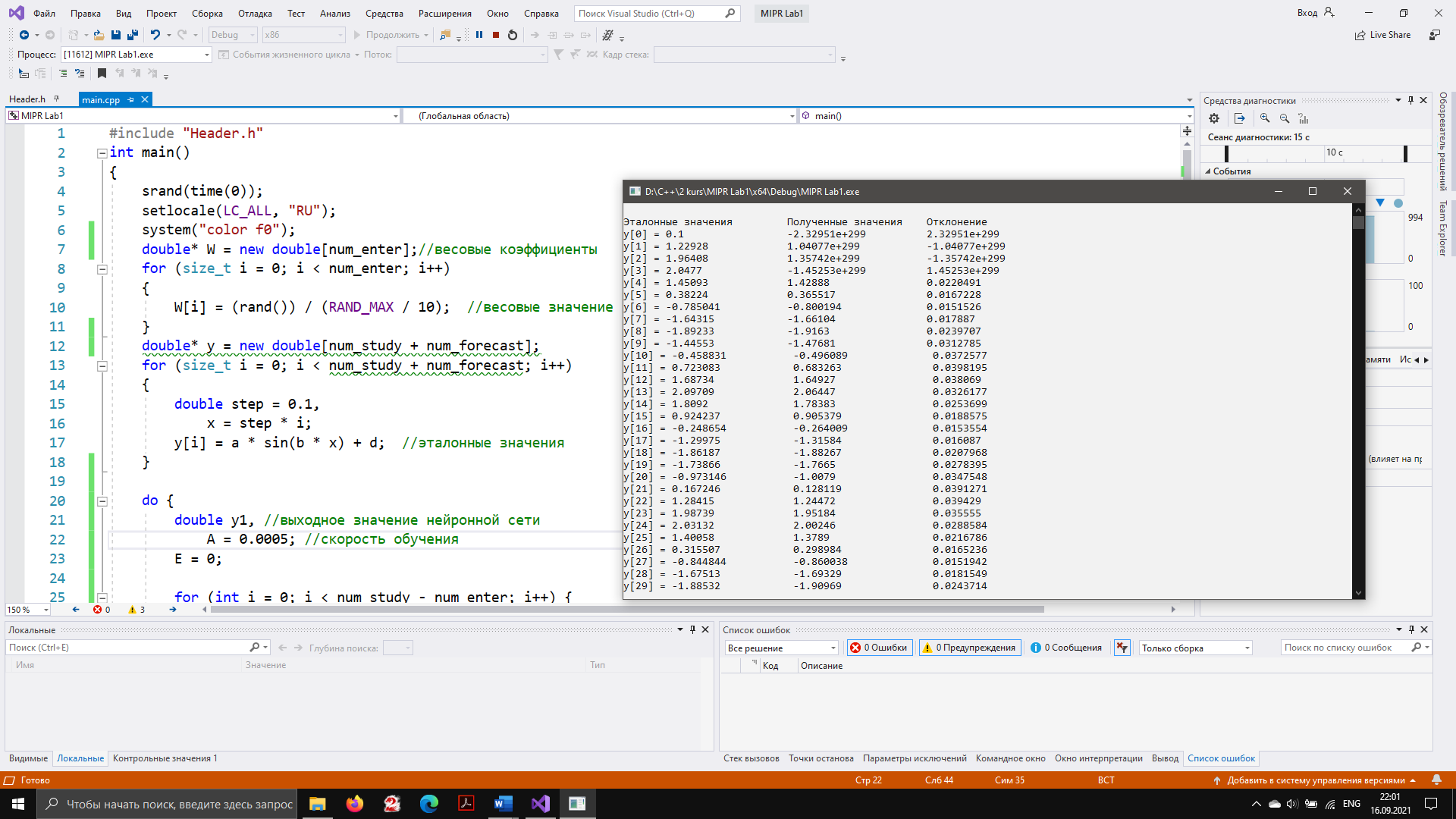
double step = 0.1,

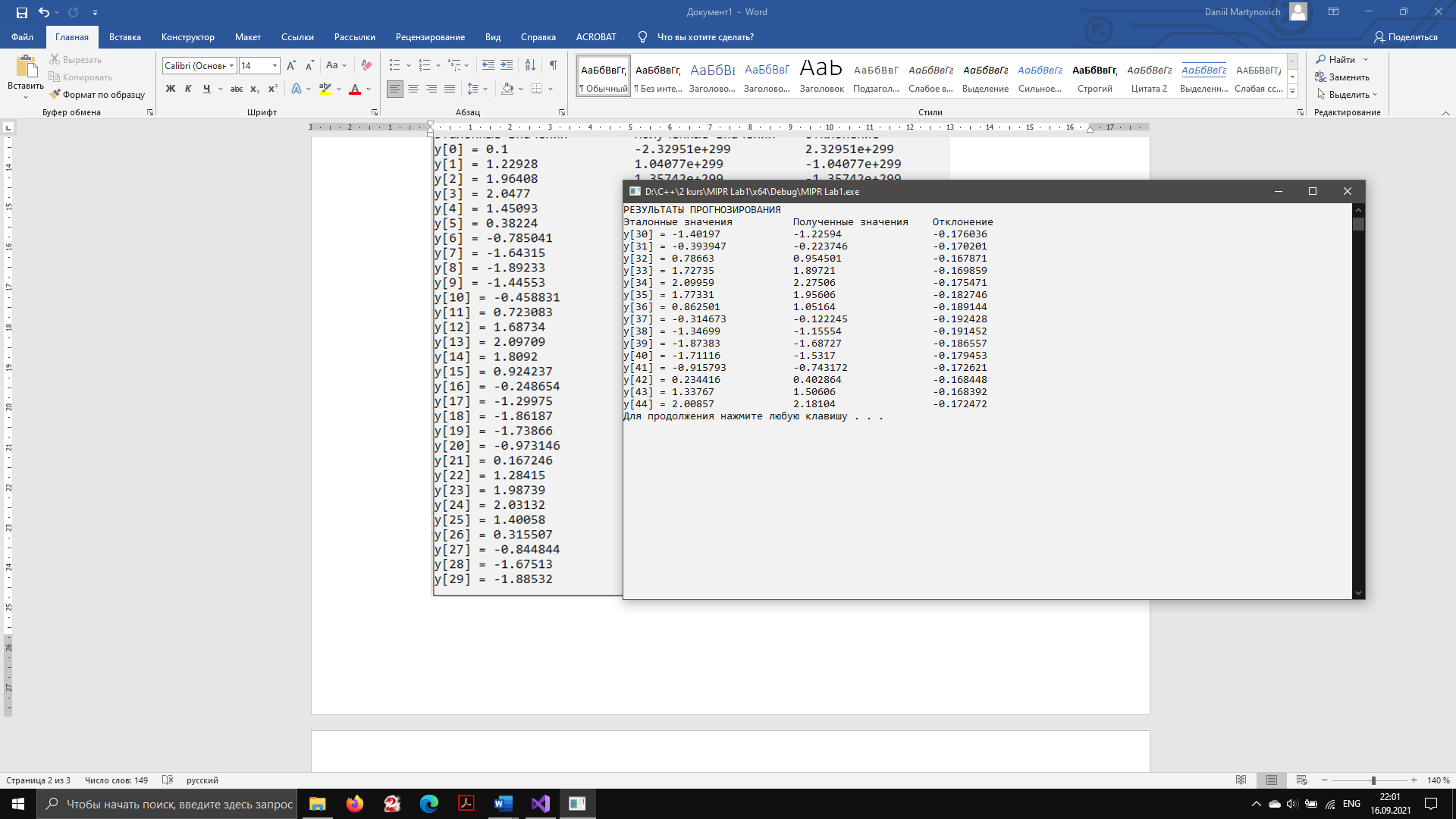
x = step \* i;

y[i] = a \* sin(b \* x) + d; //эталонные значения

}

}





Вывод: изучил обучение и функционирование линейной ИНС при решении задач прогнозирования