Министерство образования Республики Беларусь

Учреждение образования

«Брестский государственный технический университет»

Кафедра ИИТ

**Лабораторная работа №1**

По дисциплине: “МиАПР”

Тема: “Линейная искусственная нейронная сеть. Правило обучения Видроу-Хоффа”

Вариант 4

**Выполнил:**

Студент 2 курса

Группы ПО-7

Пищанюк В.Ю.

**Проверил:**

Крощенко А.А.

**Брест, 2021**

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

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

y = a\*sin(bx) + d

a = 4, b = 8, d = 0.4, кол-во входов ИНС = 3.

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.

**Код программы:**

#include <iostream>

#include <iomanip>

#include <Windows.h>

using namespace std;

int main() {

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

int a = 4,

b = 8,

inputs\_number = 3,

n\_obychenie = 30,

n\_prognoz = 15,

era = 0;

double d = 0.4,

x = 0,

h = 0.1,

min\_error = 0.001,

sum\_error,

T = 1;

double\* w = new double[inputs\_number];

for (int i = 0; i < inputs\_number; i++) {

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

}

double\* main = new double[n\_obychenie + n\_prognoz];

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

x += h;

main[i] = a \* sin(b \* x) + d;

}

double y,

V = 0.03;

do {

sum\_error = 0;

for (int i = 0; i < n\_obychenie - inputs\_number; i++) {

y = 0;

for (int j = 0; j < inputs\_number; j++) {

y += w[j] \* main[i + j];

}

y -= T;

for (int j = 0; j < inputs\_number; j++) {

w[j] -= V \* (y - main[i + inputs\_number]) \* main[i + j];

}

sum\_error += 0.5 \* pow((y - main[i + inputs\_number]), 2);

T += V \* (y - main[i + inputs\_number]);

}

era++;

} while (sum\_error > min\_error);

cout << "Number of eras: " << era << endl;

cout << "Results" << endl;

cout << setw(27) << left << "Reference values" << setw(29) << left << "The resulting values" << setw(30) << left << "Deviation" << endl;

double\* prognoz = new double[n\_obychenie + n\_prognoz];

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

prognoz[i] = 0;

for (int j = 0; j < inputs\_number; j++) {

prognoz[i] += w[j] \* main[i + j];

}

prognoz[i] -= T;

cout << "y[" << i << "] = " << setw(25) << left << main[i + inputs\_number] << setw(25) << left;

cout << prognoz[i] << setw(30) << left << pow(main[i + inputs\_number] - prognoz[i], 2) << endl;

}

cout << "Forecasting Results" << endl;

cout << setw(27) << left << "Reference values" << setw(29) << left << "The resulting values" << setw(30) << left << "Deviation" << endl;

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

prognoz[i + n\_obychenie] = 0;

for (int j = 0; j < inputs\_number; j++) {

prognoz[i + n\_obychenie] += w[j] \* main[n\_obychenie + j + i - inputs\_number];

}

prognoz[i + n\_obychenie] -= T;

cout << "y[" << n\_obychenie + i << "] = " << setw(25) << left << main[i + n\_obychenie] << setw(25) << left;

cout << prognoz[i + n\_obychenie] << setw(30) << left << pow(main[i + n\_obychenie] - prognoz[i + n\_obychenie], 2) << endl;

}

delete[] w;

delete[]main;

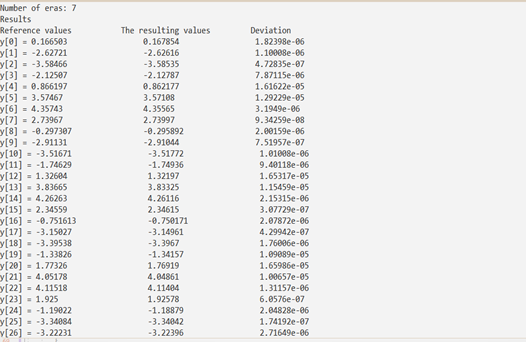
delete[]prognoz;

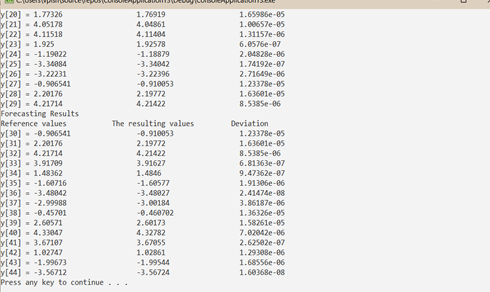
system("pause");

return 0;

}

**Результаты работы программы:**





**Вывод:** В данной лабораторной работе я изучил обучение и функционирование линейной ИНС при решении задач прогнозирования, а так же решил поставленные мне задачи.