Министерство образования Республики Беларусь

Учреждение образования

«Брестский государственный технический университет»

Кафедра ИИТ

Лабораторная работа №2

за 3 семестр

По дисциплине: «МиАПР»

Тема: «Линейная искусственная нейронная сеть.

Адаптивный шаг обучения.»

Выполнил:

Студент 2 курса

Группы ПО-4(2)

Луд А.С.

Проверил:

Крощенко А.А.

2020

Лабораторная работа №2

Линейная искусственная нейронная сеть. Адаптивный шаг обучения.

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

**Условие:**

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

1. Выбор адаптивного шага обучения

Для ускорения процедуры обучения градиентного спуска, вместо постоянного шага обучения можно использовать адаптивный шаг обучения α(t). Назовем адаптивным шагом обучения такой шаг, который целенаправленно выбирается на каждом этапе алгоритма таким образом, чтобы минимизировать среднеквадратичную ошибку сети.

Вариант 4

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.

**Текст 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, // входы ИНС

n\_obychenie = 30, // кол зн на кот проводится обучение

n\_prognoz = 15, // кол зн на кот проводится прогнозирование

era = 0;

double d = 0.4,

x = 0,

h = 0.1, // шаг

min\_error = 0.001, //минимальная ошибка

sum\_error, //суммарная ошибка

T = 1; // пороговое значение ИНС

double\* w = new double[inputs\_number]; // весовые коэффициенты

for (int i = 0; i < inputs\_number; i++) { // задаём их с помощью рандома

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

}

double\* etalon = new double[n\_obychenie + n\_prognoz]; // эланонные зн для у

for (int i = 0; i < n\_obychenie + n\_prognoz; i++) { // вычисление эталонных значений

x += h;

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

}

double y, // вых зн ИНС

V = 0.0003; // скорость

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] \* etalon[i + j];

}

y -= T;

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

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

}

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

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

double t = 0.0;

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

t += pow(etalon[i + j], 2);

}

V = 1 / (1 + t); // адаптивный шаг

}

era++;

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

cout << "Количество эпох: " << era << endl;

cout << "||Результаты Обучения||" << endl;

cout << setw(27) << left << "|Эталонные значения|" << setw(29) << left << "|Полученные значения|" << setw(30) << left << "|Отклонение|" << endl;

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

for (int i = 0; i < n\_obychenie; i++) { //значения в результате обучения

prognoz[i] = 0;

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

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

}

prognoz[i] -= T;

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

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

}

cout << "||Результаты Прогнозирования||" << endl;

cout << setw(27) << left << "|Эталонные значения|" << setw(29) << left << "|Полученные значения|" << setw(30) << left << "|Отклонение|" << endl;

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

prognoz[i + n\_obychenie] = 0;

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

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

}

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

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

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

}

delete[] w;

delete[]etalon;

delete[]prognoz;

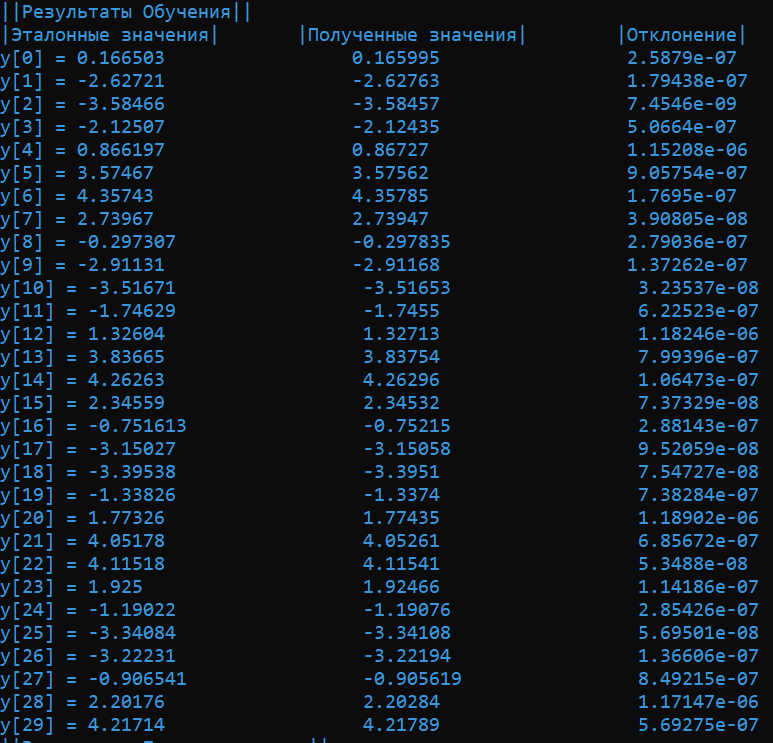
system("pause");

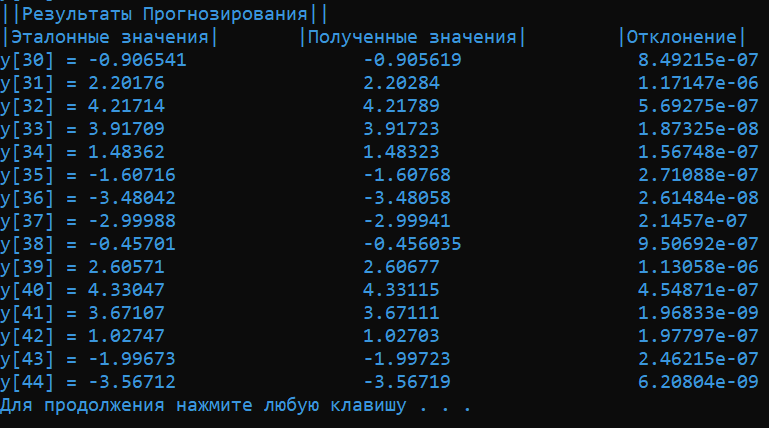
return 0;

}

**Результаты программы:**







**Вывод:** Изучил обучение и функционирование линейной ИНС с применением адаптивного шага