Министерство образования Республики Беларусь
Учреждение образования
Брестский государственный технический университет
Кафедра ИИТ

Лабораторная работа №1
За 3 семестр
По дисциплине: **«МиАПР»**
Тема: **«Линейная искусственная нейронная сеть. Правило обучения Видроу-Хоффа.»**

Выполнил: студент 2 курса
Группы ПО-4(2)
Юрьев В. А.
Проверил: Крощенко А.А.

Брест 2020

Цель работы:
Изучить обучение и функционирование линейной ИНС при решении задач прогнозирования.
Задание:
Написать на любом ЯВУ программу моделирования прогнозирующей линейной ИНС. Для тестирования использовать функцию: y = a sin (bx) +d.

№ варианта	a	b	d	Кол-во входов ИНС
11	3	5	0,5	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.
Код программы:

```
#include <iostream>
#include <iomanip>
#include <ctime>
using namespace std;
int main()
{
	setlocale(0, ""); srand(time(NULL));
	int a = 3, b = 5, enteries = 4 /*входы ИНС*/, n = 30/*количество значений для обучения*/, values = 15/*количество значений для прогнозирования*/;
	double d = 0.5, Em = 0.05/*минимальная среднеквадратичная ошибка сети*/, E/*суммарная среднеквадратичная ошибка сети*/, T = 1 /*порог НС*/;
	double* W = new double[enteries]; //весовые коэффициенты (3)

	for (int i = 0; i < enteries; i++)
	{ //генерирует весовые коэффициенты
		W[i] = (double)(rand()) / RAND_MAX; //от 0 до 1
		cout << "W[" << i << "] = " << W[i] << endl; //вывод весовых коэффициентов
	}
	cout << endl;

	double* etalon_values = new double[n + values]; //эталонные значения y

	for (int i = 0; i < n + values; i++)
	{ //вычисляем эталонные значения
		double step = 0.1; //шаг
		double x = step * i;
		etalon_values[i] = a * sin(b * x) + d; //формула для проверки
	}

	int era = 0; //для индексов

	while (1)
	{
		double y1; //выходное значение нейронной сети
		double Alpha = 0.05; //скорость обучения

		E = 0; //ошибка

		for (int i = 0; i < n - enteries; i++)
		{
			y1 = 0;
			for (int j = 0; j < enteries; j++)
				//векторы выходной активности сети
```

```
                        y1 += W[j] * etalon_values[j + i];

                y1 -= T;
                for (int j = 0; j < enteries; j++)
                        //изменение весовых коэффициентов
                        W[j] -= Alpha * (y1 - etalon_values[i + enteries]) *
etalon_values[i + j];

                T += Alpha * (y1 - etalon_values[i + enteries]); //изменение порога
нейронной сети
                E += 0.5 * pow(y1 - etalon_values[i + enteries], 2); //расчет суммарной
среднеквадратичной ошибки

                era++;
            }
            cout << era << " | " << E << endl;
            if (E < Em) break;
        } //далее сеть обучена
        cout << endl;
        cout << "РЕЗУЛЬТАТЫ ОБУЧЕНИЯ" << endl;
        cout << setw(27) << right << "Эталонные значения" << setw(23) << right << "Полученные
значения";
        cout << setw(23) << right << "Отклонение" << endl;
        double* prognoz_values = new double[n + values];
        for (int i = 0; i < n; i++)
        {
            prognoz_values[i] = 0;

            for (int j = 0; j < enteries; j++)
                prognoz_values[i] += W[j] * etalon_values[j + i]; //получаемые значения
в результате обучения

            prognoz_values[i] -= T;

            cout << "y[" << i + 1 << "] = " << setw(20) << right << etalon_values[i +
enteries] << setw(23) << right;
            cout << prognoz_values[i] << setw(23) << right << etalon_values[i + enteries]
- prognoz_values[i] << endl;
        }
        cout << endl << "РЕЗУЛЬТАТЫ ПРОГНОЗИРОВАНИЯ" << endl;
        cout << setw(28) << right << "Эталонные значения" << setw(23) << right << "Полученные
значения" << setw(23) << right << "Отклонение" << endl;
        for (int i = 0; i < values; i++)
        {
            prognoz_values[i + n] = 0;

            for (int j = 0; j < enteries; j++)
                //прогнозируемые значения
                prognoz_values[i + n] += W[j] * etalon_values[n - enteries + j + i];

            prognoz_values[i + n] -= T;

            cout << "y[" << n + i + 1 << "] = " << setw(20) << right << etalon_values[i +
n] << setw(23) << right;
            cout << prognoz_values[i + n] << setw(23) << right << etalon_values[i + n] -
prognoz_values[i + n] << endl;
        }
        delete[]etalon_values;
        delete[]prognoz_values;
        delete[]W;
        system("pause");
        return 0;
}
```

Результат работы программы:

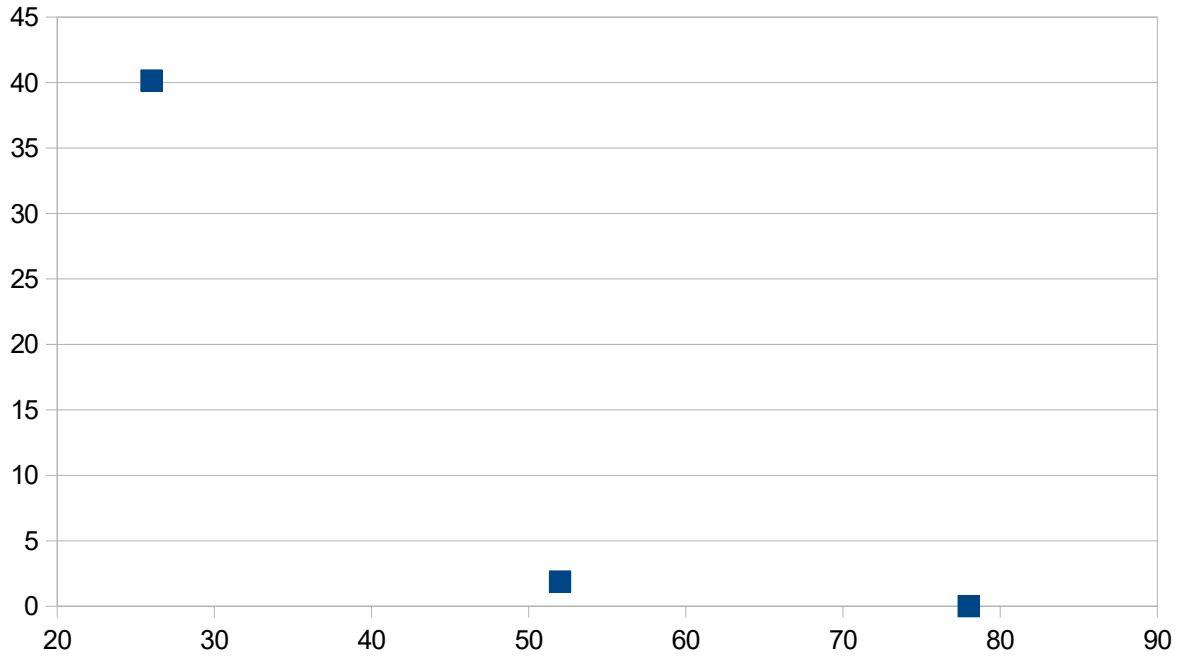
```
W[0] = 0.9494
W[1] = 0.752312
W[2] = 0.708731
W[3] = 0.476089

26 | 40.1452
52 | 1.88319
78 | 0.0158383

РЕЗУЛЬТАТЫ ОБУЧЕНИЯ
          Эталонные значения    Полученные значения              Отклонение
y[1] =               3.22789                3.22978            -0.00188745
y[2] =               2.29542                2.29462            0.000792874
y[3] =               0.92336               0.919271              0.0040895
y[4] =              -0.55235              -0.559545             0.00719529
y[5] =              -1.77041               -1.77976             0.00934985
y[6] =              -2.43259               -2.44262              0.0100257
y[7] =              -2.37677               -2.38583             0.00905726
y[8] =              -1.61662                -1.6233             0.00668174
y[9] =             -0.338246              -0.341727             0.00348073
y[10] =               1.14536                1.14512            0.000237933
y[11] =               2.47096                2.47321             -0.0022527
y[12] =                 3.314                3.31738            -0.00338137
y[13] =               3.46807                3.47095            -0.00287174
y[14] =               2.89546                2.89631           -0.000848589
y[15] =               1.73636                1.73416             0.00219275
y[16] =              0.274547               0.269039             0.00550764
y[17] =              -1.13206               -1.14035              0.0082845
y[18] =              -2.13909               -2.14893             0.00984344
y[19] =              -2.49997               -2.50977             0.00980278
y[20] =              -2.12636               -2.13453             0.00817249
y[21] =              -1.10972               -1.11507              0.0053517
y[22] =              0.301034               0.299003             0.00203105
y[23] =                1.7605                1.76148           -0.000976446
y[24] =               2.91135                2.91429            -0.00293446
y[25] =               3.47182                3.47519            -0.00336359
y[26] =               3.30469                3.30684            -0.00215878
y[27] =               2.45086                2.45048            0.000385002
y[28] =                1.1194                1.11576             0.00364494
y[29] =              -0.36371              -0.370533             0.00682289
y[30] =              -1.63536                -1.6445             0.00914078

РЕЗУЛЬТАТЫ ПРОГНОЗИРОВАНИЯ
          Эталонные значения    Полученные значения              Отклонение
y[31] =               2.45086                2.45048            0.000385002
y[32] =                1.1194                1.11576             0.00364494
y[33] =              -0.36371              -0.370533             0.00682289
y[34] =              -1.63536                -1.6445             0.00914078
y[35] =              -2.38419               -2.39422              0.0100311
y[36] =              -2.42688               -2.43615             0.00927589
y[37] =              -1.75296               -1.76002             0.00706004
y[38] =             -0.527442              -0.531368             0.00392606
y[39] =              0.949632                0.94899            0.000641267
y[40] =               2.31662                2.31861            -0.00199011
y[41] =               3.23884                3.24216            -0.00332382
y[42] =               3.49049                3.49352            -0.00303332
y[43] =               3.00997                3.01116            -0.00118973
y[44] =               1.91492                1.91316             0.00175556
y[45] =              0.473446               0.468365             0.00508145
Для продолжения нажмите любую клавишу . . .
```

График:



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