

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.
Правило адаптивного шага обучения.**»

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 .

Нейронная сеть представляет собой последовательность связанных нейронов. К нейрону поступают входящие сигналы, каждому из которых присвоен определенный вес. Сигнал умножается на свой вес, значения суммируются, и получается единое число, которое получает активационная функция. На выходе она принимает решение, транслировать ли сигнал дальше. Самая сложная задача в работе с нейронной сетью – грамотно подобрать коэффициенты к нейронам. Для этого используется обучение – процесс нахождения корректных весов для нейронной сети. От того, как именно обучают нейронную сеть, будут зависеть ее решения.

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] = 0.0876186
w[1] = 0.709555
w[2] = 0.861232
w[3] = 0.0251473

```

```

26 | 28.2617
52 | 1.41063
78 | 0.0118494

```

РЕЗУЛЬТАТЫ ОБУЧЕНИЯ

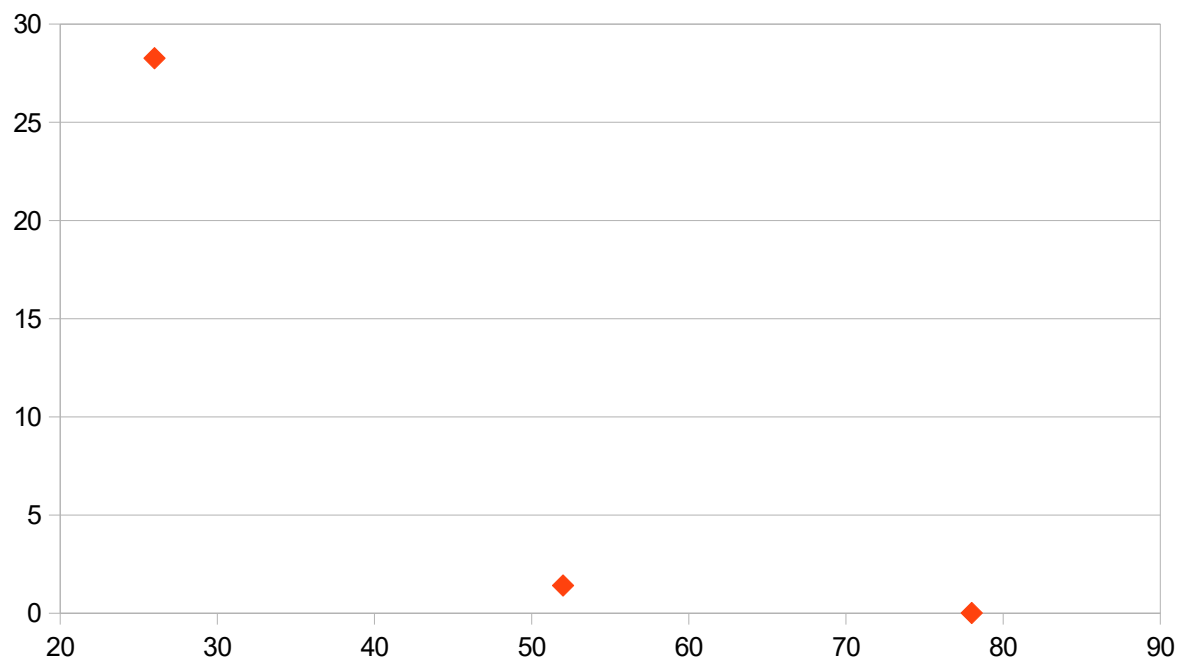
	Эталонные значения	Полученные значения	Отклонение
y[1] =	3.22789	3.22952	-0.00163255
y[2] =	2.29542	2.29473	0.000685796
y[3] =	0.92336	0.919823	0.00353721
y[4] =	-0.55235	-0.558573	0.00622357
y[5] =	-1.77041	-1.77849	0.00808715
y[6] =	-2.43259	-2.44126	0.00867169
y[7] =	-2.37677	-2.38461	0.00783407
y[8] =	-1.61662	-1.6224	0.00577938
y[9] =	-0.338246	-0.341257	0.00301066
y[10] =	1.14536	1.14515	0.0002058
y[11] =	2.47096	2.47291	-0.00194847
y[12] =	3.314	3.31692	-0.00292471
y[13] =	3.46807	3.47056	-0.00248391
y[14] =	2.89546	2.8962	-0.000733988
y[15] =	1.73636	1.73446	0.00189662
y[16] =	0.274547	0.269783	0.00476384
y[17] =	-1.13206	-1.13923	0.00716568
y[18] =	-2.13909	-2.1476	0.00851408
y[19] =	-2.49997	-2.50845	0.00847892
y[20] =	-2.12636	-2.13343	0.00706879
y[21] =	-1.10972	-1.11435	0.00462895
y[22] =	0.301034	0.299278	0.00175676
y[23] =	1.7605	1.76135	-0.000844577
y[24] =	2.91135	2.91389	-0.00253816
y[25] =	3.47182	3.47473	-0.00290934
y[26] =	3.30469	3.30655	-0.00186723
y[27] =	2.45086	2.45053	0.000333007
y[28] =	1.1194	1.11625	0.00315269
y[29] =	-0.36371	-0.369611	0.00590146
y[30] =	-1.63536	-1.64326	0.00790632

РЕЗУЛЬТАТЫ ПРОГНОЗИРОВАНИЯ

	Эталонные значения	Полученные значения	Отклонение
y[31] =	2.45086	2.45053	0.000333007
y[32] =	1.1194	1.11625	0.00315269
y[33] =	-0.36371	-0.369611	0.00590146
y[34] =	-1.63536	-1.64326	0.00790632
y[35] =	-2.38419	-2.39287	0.00867641
y[36] =	-2.42688	-2.4349	0.00802318
y[37] =	-1.75296	-1.75907	0.00610658
y[38] =	-0.527442	-0.530838	0.00339585
y[39] =	0.949632	0.949077	0.000554664
y[40] =	2.31662	2.31834	-0.00172135
y[41] =	3.23884	3.24171	-0.00287494
y[42] =	3.49049	3.49311	-0.00262367
y[43] =	3.00997	3.011	-0.00102906
y[44] =	1.91492	1.9134	0.00151847
y[45] =	0.473446	0.469051	0.0043952

Для продолжения нажмите любую клавишу . . . ■

График:



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