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

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

Выполнил:
Студент 2 курса
Группы ПО-4(1)
Прокопюк Н.О
Проверил:
Крощенко А.А.

2020

Лабораторная работа №1

Линейная искусственная нейронная сеть. Правило Видроу-Хоффа

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

Вариант 7

Задание:
Написать на любом ЯВУ программу моделирования прогнозирующей линейной ИНС. Для тестирования использовать функцию
y = a*sin(bx) + d
a = 3, b = 7, d = 0.3, кол-во входов ИНС = 5.
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 альфа. 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 альфа.

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 нейросетью – грамотно подобрать коэффициенты к нейронам. Для этого используется обучение – процесс нахождения корректных весов для нейросети. От того, как именно обучат нейросеть, будут зависеть ее решения.

Код программы:

```
#include <iostream>
#include <iomanip>
#include <ctime>

using namespace std;

int main() {
	setlocale(0, "");
	int a = 3,
		b = 7,
		enteries = 5, //входы ИНС
		n = 30, //количество значений для обучения
		values = 15; //количество значений для прогнозирования

 double d = 0.3,
	 Em = 0.05, //минимальная среднеквадратичная ошибка сети
	 E, //суммарная среднеквадратичная ошибка сети
	  T = 1; //порог НС

	double* W = new double[enteries]; //весовые коэффициенты (3)
	//srand(time(NULL)); //для разного рандома
	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;
}
```

C:\Users\uzzzz\source\repos\ConsoleApplication1\Debug\ConsoleApplication1.exe

```
W[0] = 0.00125126
W[1] = 0.563585
W[2] = 0.193304

27 | 16.014
54 | 0.592588
81 | 0.0309533

РЕЗУЛЬТАТЫ ОБУЧЕНИЯ
         Эталонные значения    Полученные значения              Отклонение
y[1] =               3.02154                3.01415               0.0073953
y[2] =               2.12639                2.12834              -0.00194609
y[3] =               0.52336               0.529123              -0.00576308
y[4] =              -1.22756               -1.22484              -0.00272228
y[5] =              -2.51473               -2.52084               0.00611407
y[6] =              -2.88849               -2.90615               0.0176592
y[7] =              -2.21829               -2.24617                 0.02788
y[8] =             -0.738246              -0.771453               0.0332061
y[9] =               1.03462                1.00285                0.031777
y[10] =                2.481                 2.45691               0.0240918
y[11] =              3.09563                3.08279               0.0128352
y[12] =               2.6638                2.66186               0.00193953
y[13] =              1.33636                1.34114              -0.00478912
y[14] =             -0.42298               -0.41798              -0.00500021
y[15] =             -1.99962                 -2.001                 0.00138
y[16] =             -2.84281               -2.85493               0.0121227
y[17] =             -2.65799               -2.68146               0.0234752
y[18] =             -1.50972               -1.54119               0.0314717
y[19] =             0.200869                0.16755               0.0333188
y[20] =              1.87622                1.84785               0.0283713
y[21] =              2.93109                2.91273               0.0183574
y[22] =              2.99697                 2.9902               0.00677538
y[23] =              2.05086                2.05319              -0.00232892
y[24] =             0.423261               0.429036              -0.00577509
y[25] =             -1.31727               -1.31491              -0.00235927
y[26] =              -2.5627               -2.56943               0.00672529
y[27] =             -2.87798               -2.89628               0.0183051
y[28] =             -2.15296                -2.1813                0.028335
y[29] =             -0.640921              -0.674232               0.0333112
y[30] =              1.12994                1.09845               0.0314955

РЕЗУЛЬТАТЫ ПРОГНОЗИРОВАНИЯ
          Эталонные значения    Полученные значения              Отклонение
y[31] =              -2.15296                -2.1813                0.028335
y[32] =             -0.640921              -0.674232               0.0333112
y[33] =              1.12994                1.09845               0.0314955
y[34] =              2.54102                 2.5175                0.023522
y[35] =              3.09938                 3.0872               0.0121762
y[36] =              2.60997                2.60855               0.00142153
y[37] =              1.24375                1.24874              -0.00498519
y[38] =             -0.522009              -0.517203              -0.00480586
y[39] =             -2.07048               -2.07238               0.00189687
y[40] =             -2.86075               -2.87353               0.0127815
y[41] =             -2.61674               -2.64078               0.0240458
y[42] =             -1.42369               -1.45544               0.0317548
y[43] =             0.301624               0.268409               0.0332155
y[44] =              1.95651                1.92859               0.0279176
y[45] =              2.96286                2.94514               0.0177119
Для продолжения нажмите любую клавишу . . .
```

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