Министерство образования Республики Беларусь

Учреждение образования

“Брестский государственный технический университет”

Кафедра ИИТ

Лабораторная работа №2

Линейная искусственная нейронная сеть.   
Правило обучения Видроу-Хоффа.

Выполнил:

Студент 4 курса

Факультета ЭИС

Группы АС – 56

Пунько Г. А.

Проверил:

Савицкий Ю. В.

Брест 2022

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 **

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.

Варианты заданий приведены в следующей таблице:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № варианта | а | b | d | Количество входов ИНС |
| 11 | 3 | 5 | 0.5 | 4 |

Код программы:

Файл Program.cs

using System.Diagnostics;

Stopwatch stopwatch = new();

stopwatch.Start();

int n\_in = 4, // количество входов

n\_ob = 30, // размер выборки для обучения

n\_pr = 15, // размер прогнозируемой выборки

it = 0; // счетчик итераций

double T, // пороговое значение

E, E\_min, // ошибка обучения и мин.ошибка

a, Y; // скорость обучения

double[] w = new double[4]; // весовые коэффициенты

double[] t = new double[60]; // массив эталонных значений

double[] y = new double[60]; // массив выходных значений сети

Random random = new();

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

w[i] = Math.Abs(1 + random.NextInt64() % 9) / (double)10; // инициализация весовых коэффициентов

T = Math.Abs(1 + random.NextInt64() % 9) / (double)10; // инициализация порогового значения

for (int j = 0; j < 60; j++) // заполнение массива выборки, на которой проходит обучение, т.е. эталонными значениями

t[j] = 3 \* Math.Sin(0.5 \* (j + 1)) + 0.5;

Console.Write("Enter a: ");

a = Convert.ToDouble(Console.ReadLine());

Console.Write("Enter Emin: ");

E\_min = Convert.ToDouble(Console.ReadLine());

Console.WriteLine("------------------------------------------------------------------");

Console.Write("t |\t\t\t\t\t Y |\t\t\t\t\t E |\t\t\t\t\t");

//обучение

do

{

E = 0;

Console.WriteLine("------------------------------------------------------------------");

it++;

Console.WriteLine($"it = {it}");

for (int i = 0; i < (n\_ob - n\_in); i++)

{

Y = 0;

for (int j = 0; j < n\_in; j++)

Y += w[j] \* t[i + j];

Y -= T; // рассчитываем выходную активность

for (int j = 0; j < n\_in; j++)

w[j] -= a \* (Y - t[i + n\_in]) \* t[i + j]; // изменяем весовые коэффициенты

T += a \* (Y - t[i + n\_in]); // меняем пороговое значение

}

for (int i = 0; i < (n\_ob - n\_in); i++)

{

Y = 0;

for (int j = 0; j < n\_in; j++)

Y += w[j] \* t[i + j];

Y -= T; // расчитываем выходную активность

E += 0.5 \* Math.Pow((Y - t[i + n\_in]), 2); // находим суммарную среднеквадратичную ошибку

Console.Write($"{t[i + n\_in]}\t\t |");//эталонные значения

Console.Write($"{Y}\t\t |");//выходная активность

Console.WriteLine($"{E}\t\t |");// сред. квадр. ошибка

}

} while (E >= E\_min);

// прогнозирование

Console.WriteLine("------------------------------------------------------------------");

Console.WriteLine(" Prediction ");

Console.WriteLine("t |\t\t Y |\t\t E |\t\t");

for (int i = (n\_ob - n\_in); i < (n\_ob + n\_pr); i++)

{

Y = 0;

for (int j = 0; j < n\_in; j++)

Y += w[j] \* t[i + j];

Y -= T;

E = 0.5 \* Math.Pow((Y - t[i + n\_in]), 2);

Console.Write($"{t[i + n\_in]}\t\t |");

Console.Write($"{Y}\t\t |");

Console.WriteLine($"{E}\t\t |");

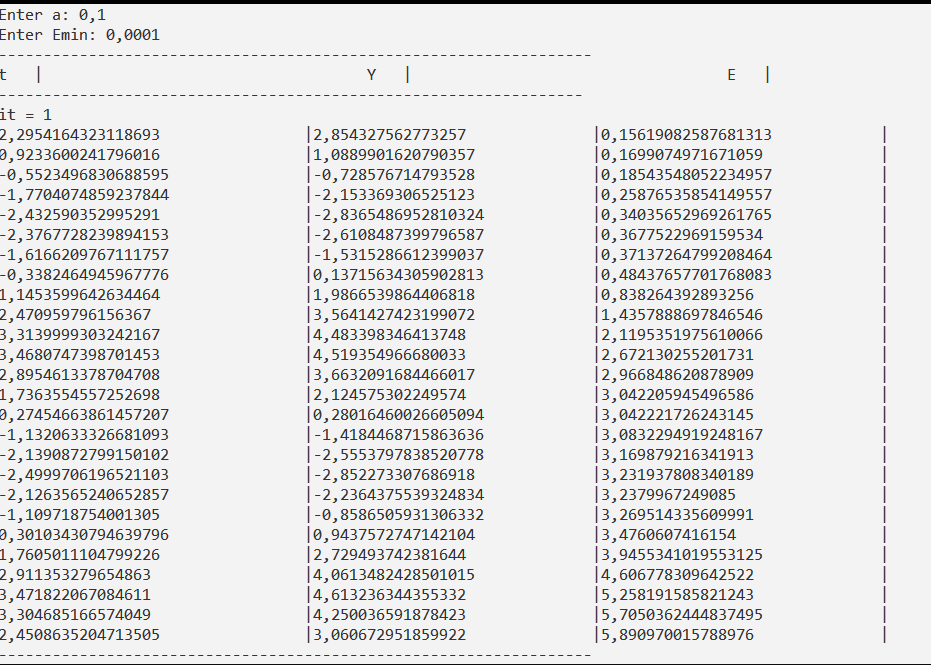
t[i + n\_in] = Y;

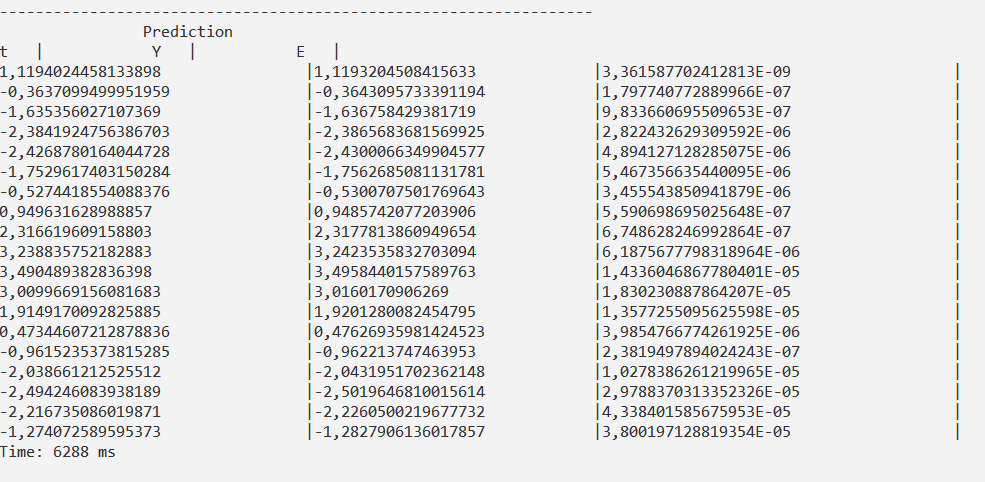
}

stopwatch.Stop();

Console.WriteLine($"Time: {stopwatch.ElapsedMilliseconds} ms");

Результат работы программы:





При E=0.0001 получили следующие значение с различными альфа.

|  |  |  |  |
| --- | --- | --- | --- |
| alpha | Error | Time, ms | Количество итераций |
| 0,05 | 2,39552428e-5 | 8674 | 8 |
| 0.1 | 3,03618e-5 | 6288 | 4 |
| 0,15 | 1,132043e-5 | 7778 | 6 |

Отобразим значения для альфа, равного 0.1. Т.к. скорость обучения при данном шаге обучения минимальна, ошибка достаточно мала.

График разности результатов между эталонным и спрогнозированным значением

График изменения ошибки в зависимости от итерации

Вывод**:** в ходе лабораторной работы изучил прогнозирование и

функционирование линейной ИНС при решении задачи прогнозирования. Определил оптимальную альфа, при которой обучение происходит с максимальной скоростью при достаточно малой ошибке. Спрогнозировал значения и убедился, что нейронная сеть работает правильно.