МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Курский государственный университет»

Кафедра программного обеспечения и администрирования

информационных систем

Направление подготовки «Математическое обеспечение и

администрирование информационных систем»

Форма обучения – очная

**Отчёт**

**о выполнении лабораторной работы № 7**

«Временные ряды»

###### Дисциплина «Основы теории нейронных сетей»

|  |  |
| --- | --- |
| Выполнил: | студент группы 413  Мусонда Салиму |
| Проверил: | проф.  кафедры ПОиАИС  Добрица В.П. |

Курск, 2021

**Цель работы:** освоить методы обучения однослойных нейронных сетей для прогнозирования временных рядов.

**Задача № 1.** Составить обучающую программу однослойной нейронной сети, интерпретирующей временной ряд, с 6 входными и одним выходным нейронами. Параметры обучающей программы: шаг обучения , предполагаемая точность , временной шаг , начальный момент , длина обучающей серии . Значения временного ряда вычисляются по данной временной функции.

|  |  |
| --- | --- |
| 4 |  |

**Задача №2.** Провести обучение нейросети для того же временного ряда с адаптивным шагом обучения. Сравнить время обучения сетей в этих задачах с постоянным шагом обучения и с адаптивным шагом обучения.

**Выполнение работы**

Для реализации алгоритма понадобятся следующие формулы:

– выходное значение нейросети, где – весовые коэффициенты, T – значение пороговой функции

– весовые коэффициенты, где– предыдущие весовые коэффициенты, – скорость обучения, и – эталонное и выходное значение нейросети, - входное значение нейросети.

– значение пороговой функции, где и – эталонное и выходное значение нейросети.

– среднеквадратичная ошибка нейросети для входного образа L. Для данного примера, возьмем .

Для реализации адаптивного шага понадобятся формула нахождения адаптивного шага обучения:

Функция нейронной сети вычисляется по формуле где . Для того, чтобы значения функции находились в интервале [0,1] надо провести преобразование по формуле

**Блок-схема алгоритма**



Рисунок 1 – Блок-схема алгоритма обучения

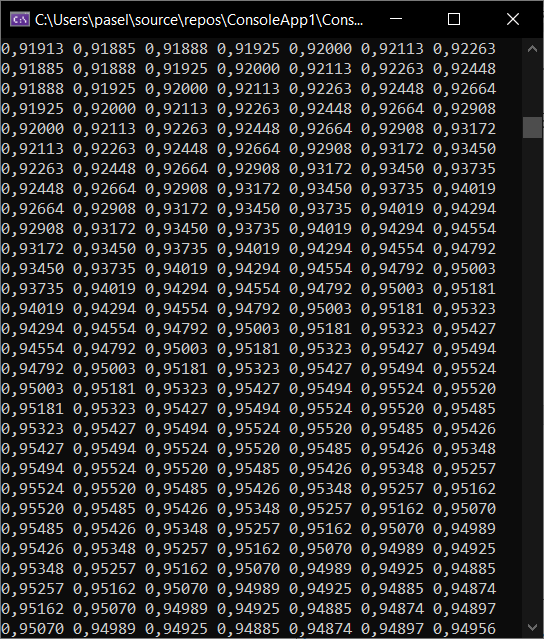


Рисунок 2 – Фрагмент обучающих данных для нейронной сети

**Результат**

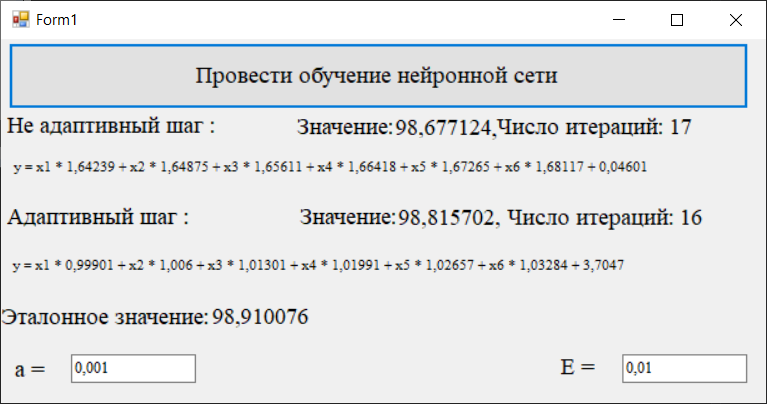
****

Рисунок 3 – результат обучения нейросети

А что же Вы вывод не сделали? Не привели схему и формулу нейронной сети, а так же проверку ее работы на последнем примере. А так все нормально. Подправьте и работа будет зачтена.

Схема, полученной нейронной сети, приведена на рисунке 4.

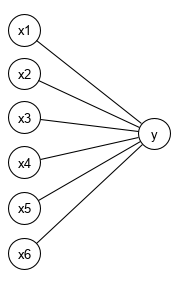


Рисунок 4 – Схема НС

**Код программы:**

using System.Linq;

using static System.Math;

namespace Lab7NN

{

public class Result

{

public double E { get; set; }

public double Y { get; set; }

}

public class NN

{

private double A;

private double[] W;private int Training;

private int Lenght;

public NN(int lengthTraining, int lenght, double a, double[] w)

{

Training = lengthTraining;

Lenght = lenght;

A = a;

W = w;

}

public Result Obuchenie()

{

double e = 0;

double t = 0;

double ttime = 0.1;//т по времени

double planY = 0;

for (int i = 0; i < Training - Lenght; i++)

{

double[] rowY = CalculateRowY(t, ttime, Lenght);

planY = rowY.Multiple(W);//Прогнозируемое Y

double calcY = Function(t + Lenght \* ttime);

A = GetNextA(rowY);

W = NextW(W, planY, calcY, rowY);

e += CalculateE(planY, calcY);

t += ttime;

}

return new Result

{

E = e,

Y = planY

};

}

protected virtual double GetNextA(double[] rowY)

{

return A;

}

private double[] CalculateRowY(double t, double ttime, int count)

{

double[] row = new double[count];

for (int i = 0; i < count; i++)

{

row[i] = Function(t + i \* ttime);

}

return row;

}

public static double Function(double t)

{

return t + Sin(2 \* t);//функиця

}

private double[] NextW(double[] W, double Y, double calculatedY, double[] rowY)

{

return W.Select((w, index) => w - A \* (Y - calculatedY) \* rowY[index]).ToArray();

}

private double CalculateE(double projectedY, double calculatedY)

{

return Pow(projectedY - calculatedY, 2) / 2; ;

}

}

}

**Нормализация**

label2.Text = $"Значение: {(Math.Round(((Adaptive.result.Y)\*105)-1,6))}, Число итераций: {Adaptive.countIteration}";

var nonAdaptive = Learn(false);

if (nonAdaptive.result.Y > 0)

{

label1.Text = $"Значение: {(Math.Round(((nonAdaptive.result.Y)\*105)-1, 6))}, Число итераций: {nonAdaptive.countIteration}";

}

else

{

label1.Text = $"Нейросеть не обучена, Число итераций: {nonAdaptive.countIteration}";

}

Label.Text = "Эталонное значение: " + (Math.Round(NN.Function(99.9),6)).ToString();