Министерство образования и науки Российской Федерации

Федеральное Государственное Бюджетное Образовательное Учреждение

Высшего Профессионального Образования

Тамбовский Государственный Технический Университет

Кафедра

Лабораторная работа №3

дисциплина «Базы знаний»

Вариант D7

Выполнил студент гр. –41 qwinmen

Проверила: И. Л.

Тамбов

**Задание:**

Создать нейронную сеть для аппроксимации таблично заданной функции. Определить оптимальные параметры сети при заданном количестве нейронов в скрытом слое.

Рисунок Таблица вариантов задания

**Решение:**

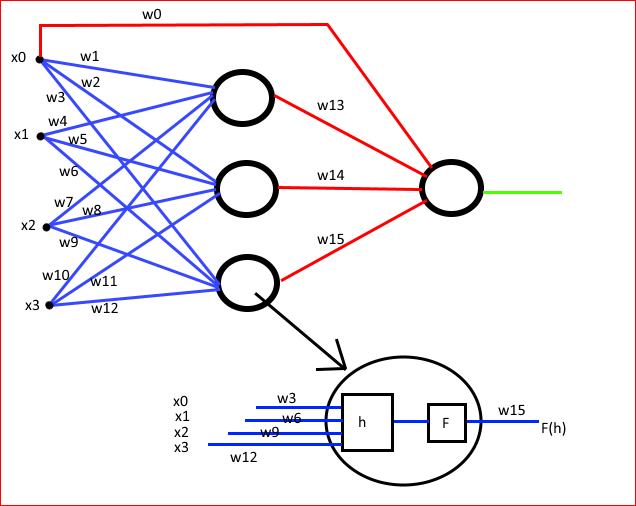
Начальные веса получаем случайным образом, дальше действуем по алгоритму:

Рисунок Сеть

λ = 1, const



Если , то задача решена, переходим к следующему примеру (x и d из следующей строки в таблице рисунок 1).

A

Если , то:

Ошибка для второго слоя

Ошибка для первого слоя (Рисунок 3)

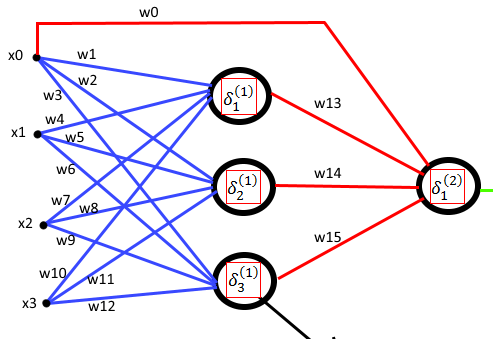


Рисунок Ошибка слоя

, const

Пересчитываем весы: (первый слой):

(второй слой):

Теперь w

[второй слой]:

Вычислить и сравнить с . [Шаг A]

A

Если , то на текущем примере пересчитываем

Если , то берем следующий пример обучения (x и ) и пересчитываем

Пишем глобальную ошибку из достигнутой на i-уровне

Если все примеры пройдены, проверяем , при

Если , то переход на начало и взять первый пример обучения для пересчета

Если , то закончить обучение, вывод статистики.

Koд:

//Вариант D7

**namespace** KiloMozg

**{**

class clNeiro

**{**

**private** const double \_c **=** .1**;**//[0.1..10]

**private** const double \_ε **=** .001**;**

**private** const double \_λ **=** 1.0**;**

**private** const double \_x0 **=** 1**;**

**private** **readonly** double**[]** \_x1 **=** **new** double**[]** **{** 0**,** 0**,** 0**,** 0**,** 1**,** 1**,** 1**,** 1 **};**

**private** **readonly** double**[]** \_x2 **=** **new** double**[]** **{** 0**,** 0**,** 1**,** 1**,** 0**,** 0**,** 1**,** 1 **};**

**private** **readonly** double**[]** \_x3 **=** **new** double**[]** **{** 0**,** 1**,** 0**,** 1**,** 0**,** 1**,** 0**,** 1 **};**

**private** **readonly** double**[]** \_d **=** **new** double**[]** **{** 0**,** 1**,** 0**,** 1**,** 1**,** 0**,** 1**,** 0 **};**

**private** **readonly** Random \_rnd **=** **new** Random**();**

**private** double \_E **=** 0.0**;**

**public** StringBuilder \_отчетНаФорму **=** **new** StringBuilder**();**

**public** double**[]** \_wForm **=** **new** double**[**16**];**

**public** int \_SelectIndexInTable**;**

**private** Thread \_thread**;**

**public** clNeiro**(){}**

**public** clNeiro**(**AdvTree advTree**)**

**{**

Log **=** advTree**;**

//Запуск отдельного фонового потока для расчетов//Независим от текущего примера

\_thread **=** **new** Thread**(new** ThreadStart**(**ShowMemory**))** **{** IsBackground **=** **true** **};**

\_thread**.**Start**();**

**}**

**private** AdvTree Log**;**

**public** PointPairList \_list **=** **new** PointPairList**();**

**delegate** void WriteToLogDelegate**(**int message**);**

**private** void WriteToLog**(**int message**)**

**{**

**if** **(**Log**.**InvokeRequired**)**

**{**

var writeToLog **=** **new** WriteToLogDelegate**(**WriteToLog**);**

Log**.**Invoke**(**writeToLog**,** message**);**

**}**

**else**

**{**

Log**.**SelectedIndex **=** message**;**

**}**

**}**

**delegate** void WriteToZedGraphDelegate**(**PointPair pair**);**

**private** void WriteToZedGraph**(**PointPair pair**)**

**{**

\_list**.**Add**(**pair**);**

**}**

**public** int threadd**;**

**private** int \_eIndex**;**

**public** void ShowMemory**()**

**{**

var индексОбучения **=** 0**;**//[0..7]

//Сгенерировать рандомные веса

var w **=** **new** double**[**16**];**

**for** **(**int i **=** 0**;** i **<** w**.**Length**;** i**++)**

**{**

w**[**i**]** **=** \_rnd**.**NextDouble**();**

**if** **(**i **==** 8**)** \_отчетНаФорму**.**AppendFormat**(**"\r\n"**);**

\_отчетНаФорму**.**AppendFormat**(**"w{0}{1}={2:0.00}; "**,** **(**i **==** 0 **||** i **==** 13 **||** i **==** 14 **||** i **==** 15**)** **?** '²' **:** '¹'**,** i**,** w**[**i**]);**

**}**

\_отчетНаФорму**.**AppendFormat**(**"\r\n"**);**

nextStep**:**

threadd**++;**

**if** **(**threadd **>** 20999**)** **throw** **new** Exception**(**"Зациклился!"**);**

WriteToLog**(**индексОбучения**);**

//Рассчитать h для первого слоя

var h1\_1 **=** h**(**1**,** **new** double**[]** **{** w**[**1**],** w**[**4**],** w**[**7**],** w**[**10**]** **},** \_x0**,** \_x1**[**индексОбучения**],** \_x2**[**индексОбучения**],** \_x3**[**индексОбучения**]);**

var h1\_2 **=** h**(**1**,** **new** double**[]** **{** w**[**2**],** w**[**5**],** w**[**8**],** w**[**11**]** **},** \_x0**,** \_x1**[**индексОбучения**],** \_x2**[**индексОбучения**],** \_x3**[**индексОбучения**]);**

var h1\_3 **=** h**(**1**,** **new** double**[]** **{** w**[**3**],** w**[**6**],** w**[**9**],** w**[**12**]** **},** \_x0**,** \_x1**[**индексОбучения**],** \_x2**[**индексОбучения**],** \_x3**[**индексОбучения**]);**

\_отчетНаФорму**.**AppendFormat**(**"h¹1={0:0.00}; h¹2={1:0.00}; h¹3={2:0.00}; \r\n"**,** h1\_1**,** h1\_2**,** h1\_3**);**

//Рассчитать F(h) для первого слоя

var Fh1\_1 **=** Fh**(**h1\_1**);**

var Fh1\_2 **=** Fh**(**h1\_2**);**

var Fh1\_3 **=** Fh**(**h1\_3**);**

\_отчетНаФорму**.**AppendFormat**(**"F(h¹1)={0:0.00}; F(h¹2)={1:0.00}; F(h¹3)={2:0.00}; \r\n"**,** Fh1\_1**,** Fh1\_2**,** Fh1\_3**);**

//Рассчитать h для второго слоя

var h2\_1 **=** h**(**2**,** **new** double**[]** **{**w**[**0**],** w**[**13**],** w**[**14**],** w**[**15**]},** \_x0**,** Fh1\_1**,** Fh1\_2**,** Fh1\_3**);**

\_отчетНаФорму**.**AppendFormat**(**"h²1={0:0.00}; \r\n"**,** h2\_1**);**

//Рассчитать F(h) для второго слоя

var Fh2\_1 **=** Fh**(**h2\_1**);**

\_отчетНаФорму**.**AppendFormat**(**"F(h²1)={0:0.00}; \r\n"**,** Fh2\_1**);**

//Рассчитать ошибку уровня

var ep1 **=** ep**(**Fh2\_1**,** \_d**[**индексОбучения**]);**

\_отчетНаФорму**.**AppendFormat**(**"ep{0}={1:0.0000}; \r\n"**,** индексОбучения**,** ep1**);**

**if(**ep1 **<=** \_ε**)**

**{**

\_E **+=** ep1**;**

\_eIndex**++;**

WriteToZedGraph**(new** PointPair**(**\_eIndex**,** \_E**));**

**if** **(**\_E **>** \_ε **+** 0.011**)**

**{**

\_отчетНаФорму**.**AppendFormat**(**"-------------------END-----------------\r\n"**);**

//Закончить обучение

**return;**

**}**

//то задача решена, переходим на новый уровень

индексОбучения **+=** индексОбучения**+**1 **<** \_d**.**Length **?** 1 **:** **-**7**;**

**goto** nextStep**;**

**}**

//иначе рассчитать ошибку для второго слоя

var δ2\_1 **=** δ**(**2**,** \_d**[**индексОбучения**],** Fh2\_1**,** h2\_1**);**

\_отчетНаФорму**.**AppendFormat**(**"δ²1={0:0.00}; \r\n"**,** δ2\_1**);**

//рассчитать ошибку для первого слоя

var прицеп **=** Прицеп**(new** double**[]** **{**w**[**0**],** w**[**13**],** w**[**14**],** w**[**15**]},** δ2\_1**);**

var δ1\_1 **=** δ**(**1**,** \_d**[**индексОбучения**],** Fh2\_1**,** h1\_1**,** прицеп**);**

var δ1\_2 **=** δ**(**1**,** \_d**[**индексОбучения**],** Fh2\_1**,** h1\_2**,** прицеп**);**

var δ1\_3 **=** δ**(**1**,** \_d**[**индексОбучения**],** Fh2\_1**,** h1\_3**,** прицеп**);**

\_отчетНаФорму**.**AppendFormat**(**"δ¹1={0:0.00}; δ¹2={1:0.00}; δ¹3={2:0.00}; \r\n"**,** δ1\_1**,** δ1\_2**,** δ1\_3**);**

//пересчитаем весы

var Δw **=** **new** double**[**w**.**Length**];**

**for** **(**int i **=** 0**;** i **<** Δw**.**Length**;** i**++)**

**{**//просчитываем все дельта от 0 до 15

**if** **(**i **==** 0 **||** i **==** 13 **||** i **==** 14 **||** i **==** 15**)**

Δw**[**i**]** **=** **this.**Δw**(**2**,** δ2\_1**,** **-**1**,** i **==** 0 **?** \_x0 **:** i **==** 13 **?** Fh1\_1 **:** i **==** 14 **?** Fh1\_2 **:** i **==** 15 **?** Fh1\_3 **:** **-**1**);**

**else**

Δw**[**i**]** **=** **this.**Δw**(**1**,** i **==** 1 **||** i **==** 4 **||** i **==** 7 **||** i **==** 10 **?** δ1\_1 **:** i **==** 2 **||** i **==** 5 **||** i **==** 8 **||** i **==** 11 **?** δ1\_2 **:** i **==** 3 **||** i **==** 6 **||** i **==** 9 **||** i **==** 12 **?** δ1\_3 **:** **-**1**,** i **==** 1 **||** i **==** 2 **||** i **==** 3 **?** \_x0 **:** i **==** 4 **||** i **==** 5 **||** i **==** 6 **?** \_x1**[**индексОбучения**]** **:** i **==** 7 **||** i **==** 8 **||** i **==** 9 **?** \_x2**[**индексОбучения**]** **:** i **==** 10 **||** i **==** 11 **||** i **==** 12 **?** \_x3**[**индексОбучения**]** **:** **-**1**);**

**}**

**{**//вывести дельта в отчет

var i **=** 0**;**

**foreach** **(**var dbl **in** Δw**)**

\_отчетНаФорму**.**AppendFormat**(**"Δw{0}{1}={2}; \r\n"**,** **(**i **==** 0 **||** i **==** 13 **||** i **==** 14 **||** i **==** 15**)** **?** '²' **:** '¹'**,** i**++,** dbl**);**

**}**

//пересчитать весы

**for** **(**int i **=** 0**;** i **<** w**.**Length**;** i**++)**

**{**

w**[**i**]** **-=** Δw**[**i**];**

**if** **(**i **==** 8**)** \_отчетНаФорму**.**AppendFormat**(**"\r\n"**);**

\_отчетНаФорму**.**AppendFormat**(**"w{0}{1}={2:0.00}; "**,** **(**i **==** 0 **||** i **==** 13 **||** i **==** 14 **||** i **==** 15**)** **?** '²' **:** '¹'**,** i**,** w**[**i**]);**

\_wForm**[**i**]** **=** w**[**i**];**//для вывода на форму

**}**

\_отчетНаФорму**.**AppendFormat**(**"\r\n"**);**

**goto** nextStep**;**

**}**

/// <summary>

/// Пробный пуск системы

/// </summary>

**public** void Tester**(**double \_x1**,** double \_x2**,** double \_x3**,** int индексОбучения**)**

**{**

\_отчетНаФорму**.**AppendFormat**(**"-------------------TEST\_d{0}={1}----------------\r\n"**,** индексОбучения**,** \_d**[**индексОбучения**]);**

var w **=** \_wForm**;**

**for** **(**int i **=** 0**;** i **<** w**.**Length**;** i**++)**

**{**

**if** **(**i **==** 8**)** \_отчетНаФорму**.**AppendFormat**(**"\r\n"**);**

\_отчетНаФорму**.**AppendFormat**(**"w{0}{1}={2:0.00}; "**,** **(**i **==** 0 **||** i **==** 13 **||** i **==** 14 **||** i **==** 15**)** **?** '²' **:** '¹'**,** i**,** w**[**i**]);**

**}**

\_отчетНаФорму**.**AppendFormat**(**"\r\n"**);**

//Рассчитать h для первого слоя

var h1\_1 **=** h**(**1**,** **new** double**[]** **{** w**[**1**],** w**[**4**],** w**[**7**],** w**[**10**]** **},** \_x0**,** \_x1**,** \_x2**,** \_x3**);**

var h1\_2 **=** h**(**1**,** **new** double**[]** **{** w**[**2**],** w**[**5**],** w**[**8**],** w**[**11**]** **},** \_x0**,** \_x1**,** \_x2**,** \_x3**);**

var h1\_3 **=** h**(**1**,** **new** double**[]** **{** w**[**3**],** w**[**6**],** w**[**9**],** w**[**12**]** **},** \_x0**,** \_x1**,** \_x2**,** \_x3**);**

\_отчетНаФорму**.**AppendFormat**(**"h¹1={0:0.00}; h¹2={1:0.00}; h¹3={2:0.00}; \r\n"**,** h1\_1**,** h1\_2**,** h1\_3**);**

//Рассчитать F(h) для первого слоя

var Fh1\_1 **=** Fh**(**h1\_1**);**

var Fh1\_2 **=** Fh**(**h1\_2**);**

var Fh1\_3 **=** Fh**(**h1\_3**);**

\_отчетНаФорму**.**AppendFormat**(**"F(h¹1)={0:0.00}; F(h¹2)={1:0.00}; F(h¹3)={2:0.00}; \r\n"**,** Fh1\_1**,** Fh1\_2**,** Fh1\_3**);**

//Рассчитать h для второго слоя

var h2\_1 **=** h**(**2**,** **new** double**[]** **{** w**[**0**],** w**[**13**],** w**[**14**],** w**[**15**]** **},** \_x0**,** Fh1\_1**,** Fh1\_2**,** Fh1\_3**);**

\_отчетНаФорму**.**AppendFormat**(**"h²1={0:0.00}; \r\n"**,** h2\_1**);**

//Рассчитать F(h) для второго слоя

var Fh2\_1 **=** Fh**(**h2\_1**);**

\_отчетНаФорму**.**AppendFormat**(**"F(h²1)={0:0.00}; \r\n"**,** Fh2\_1**);**

//Рассчитать ошибку уровня

var ep1 **=** ep**(**Fh2\_1**,** \_d**[**индексОбучения**]);**

\_отчетНаФорму**.**AppendFormat**(**"ep{0}={1:0.0000}; \r\n"**,** индексОбучения**,** ep1**);**

\_отчетНаФорму**.**AppendFormat**(**"-------------------END----------------------\r\n"**);**

**}**

/// <summary>

/// Вычислить h

/// </summary>

/// <param name="номерСлоя">1\2 слой</param>

/// <param name="w">значение весов</param>

/// <param name="вход">x0..3 или F(h)</param>

/// <returns>Вернет просчитаное число</returns>

**private** double h**(**int номерСлоя**,** double**[]** w**,** **params** double**[]** вход**)**

**{**

**switch** **(**номерСлоя**)**

**{**

**case** 1**:**

**return** вход**[**0**]\***w**[**0**]** **+** вход**[**1**]\***w**[**1**]** **+** вход**[**2**]\***w**[**2**]** **+** вход**[**3**]\***w**[**3**];**

**case** 2**:**

**return** вход**[**0**]\***w**[**0**]** **+** вход**[**1**]\***w**[**1**]** **+** вход**[**2**]\***w**[**2**]** **+** вход**[**3**]\***w**[**3**];**

**}**

**return** **-**1**;**

**}**

**private** double Fh**(**double h**)**

**{**

**return** 1.0 **/** **(**1.0 **+** Math**.**Exp**(-**\_λ **\*** h**));**

**}**

**private** double ep**(**double Fh**,** double d**)**

**{**

**return** Math**.**Pow**(**Fh **-** d**,** 2**);**

**}**

**private** double δ**(**int номерСлоя**,** double d**,** double Fh**,** double h**,** **params** double**[]** прицеп**)**

**{**

**switch** **(**номерСлоя**)**

**{**

**case** 1**:**

**return** **(**1.0**\***\_λ**\***Math**.**Exp**(-**\_λ**\***h**))/(**Math**.**Pow**(**1.0 **+** Math**.**Exp**(-**\_λ**\***h**),** 2**))\***прицеп**[**0**];**

**case** 2**:**

**return** **-**2.0**\*(**d **-** Fh**)\*(**1.0**\***\_λ**\***Math**.**Exp**(-**\_λ**\***h**))/(**Math**.**Pow**(**1.0 **+** Math**.**Exp**(-**\_λ**\***h**),** 2**));**

**}**

**return** **-**1**;**

**}**

**private** double Прицеп**(**double**[]** w**,** double δ**)**

**{**

**return** δ**\***w**[**0**]** **+** δ**\***w**[**1**]** **+** δ**\***w**[**2**]** **+** δ**\***w**[**3**];**

**}**

**private** double Δw**(**int номерСлоя**,** double δ**,** double x**,** **params** double**[]** Fh**)**

**{**

**switch** **(**номерСлоя**)**

**{**

**case** 1**:**

**return** \_c **\*** δ **\*** x**;**

**case** 2**:**

**return** \_c **\*** δ **\*** Fh**[**0**];**

**}**

**return** **-**1**;**

**}**

**}**

Тестирование программы:

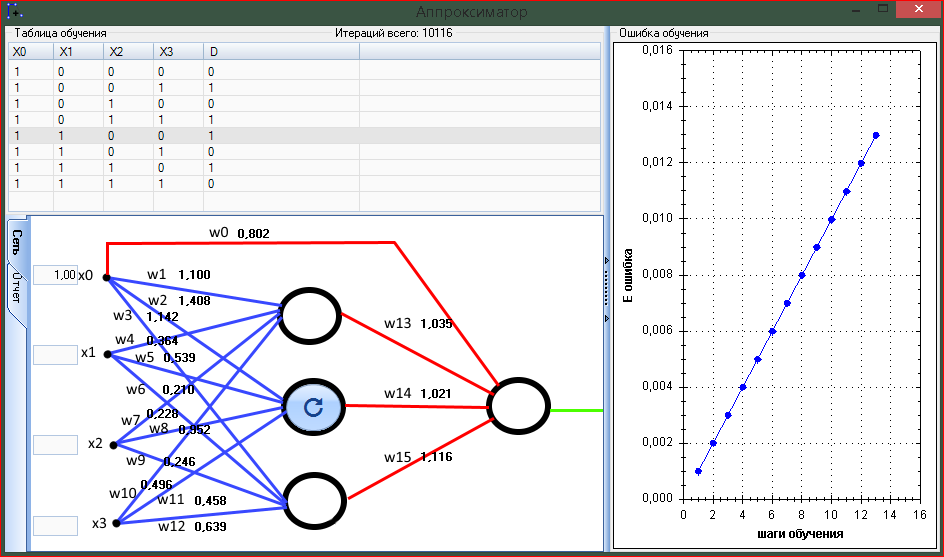


Рисунок Устоявшаяся система

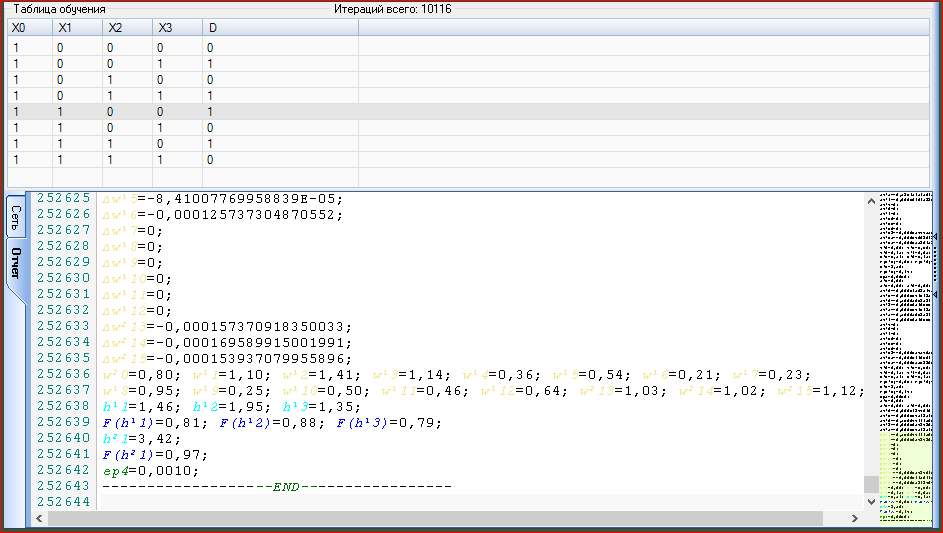
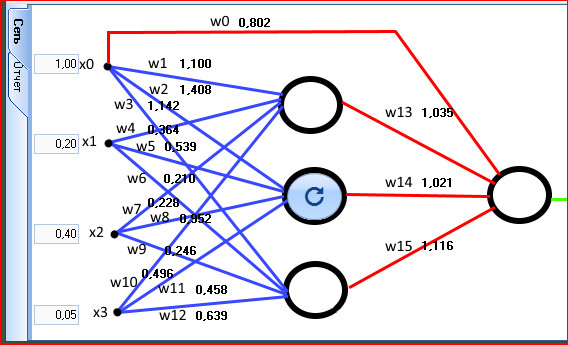


Рисунок Отчет для значений каждого шага итерации

Рисунок Тестирование системы

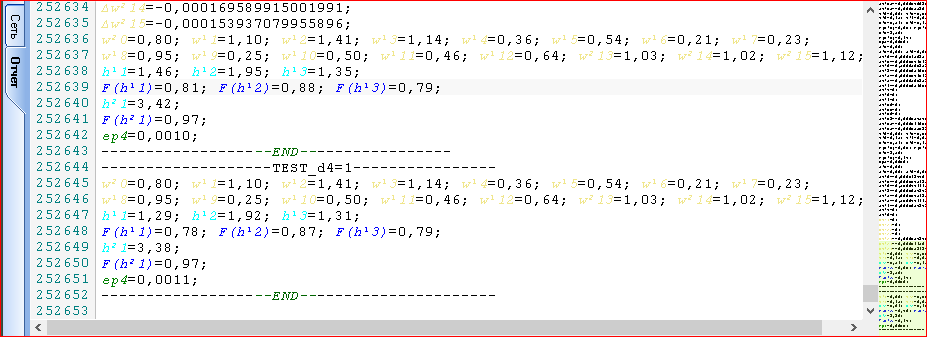


Рисунок Отчет теста