

Московский энергетический институт

Кафедра управления и информатики

Лабораторные работы

по курсу

“Нейрокомпьютеры и их применение”

Группа: А-1-00

Студент: Курапов О.В.

2005 г.

Цель работы

- моделирование нейронных сетей с использованием пакета Statistica Neural Networks
- обучение тестовых нейронных сетей выполнению логических операций “Импликация” и “Исключающее ИЛИ”, решению задач “Сократ”, “Энигма”, а также создание системы, оценивающей стоимость недвижимости (“Real Estate Tutorial”)

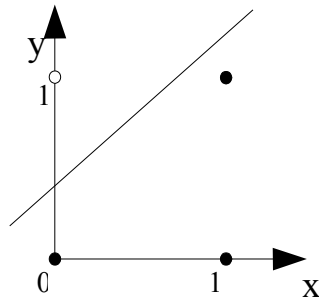
Задача 1: “Импликация”

Постановка задачи:

Распознавание логической функции “Импликация”, заданной таблицей истинности.

Решение задачи:

Данная задача является линейно разделимой, как показано на графике:

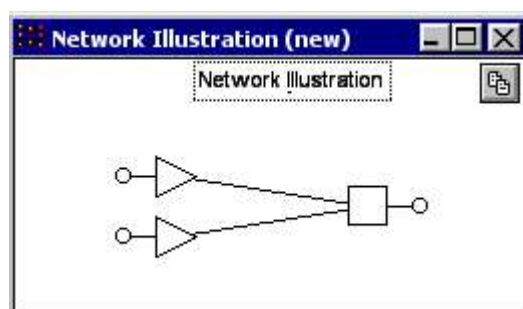


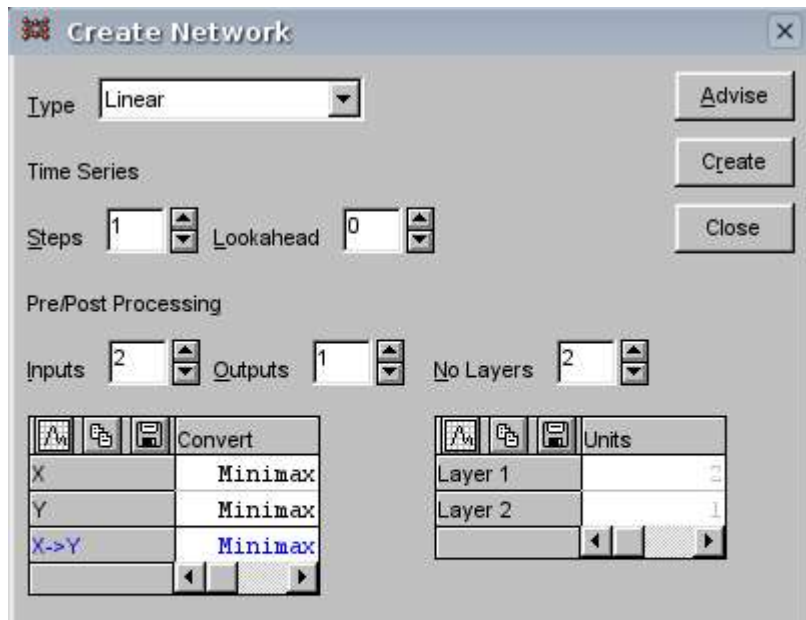
Следовательно, она может быть решена с помощью классического персептрона Розенблатта.

1. При помощи диалога “Create data set” (File -> New -> Data Set..) создадим новую таблицу. Заполним ее набором исходных данных:

	X	Y	X->Y
01	0	0	1
02	1	0	1
03	0	1	0
04	1	1	1

2. Создадим новую однослойную сеть (File -> New -> Network) с двумя нейронами во входном слое и одним – в выходном:





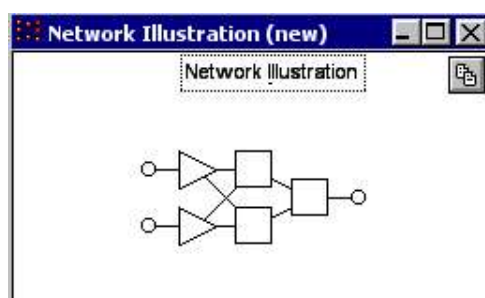
3. Проведем обучение сети (Train -> Linear).

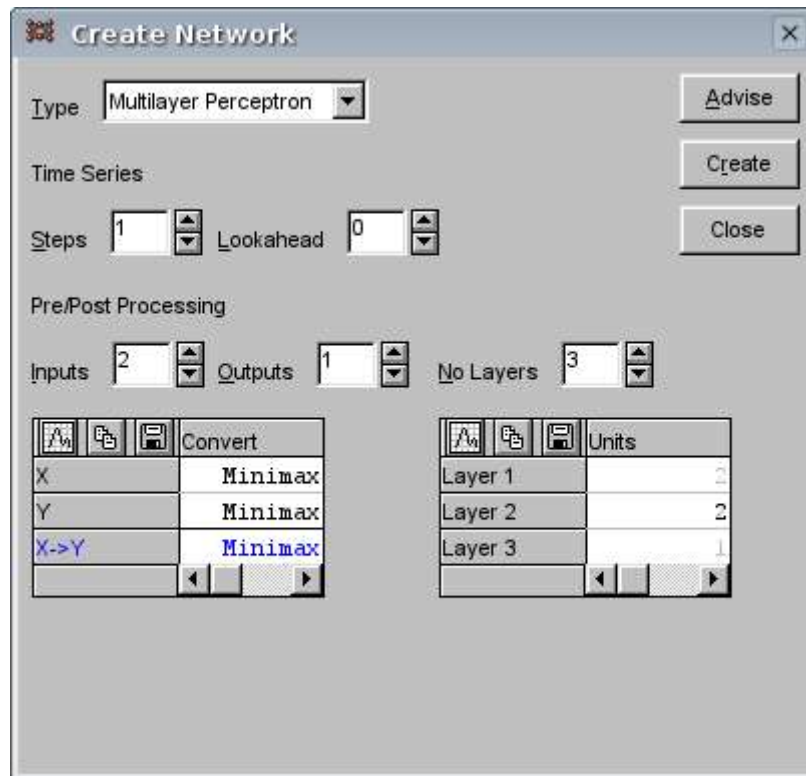
4. Проверим результат обучения запуском на выполнение набора исходных данных (Run -> Data Set...):

	X->Y	T. X->Y	E. X->Y	Error
01	0.75	1	-0.25	0.25
02	1.25	1	0.25	0.25
03	0.25	0	0.25	0.25
04	0.75	1	-0.25	0.25

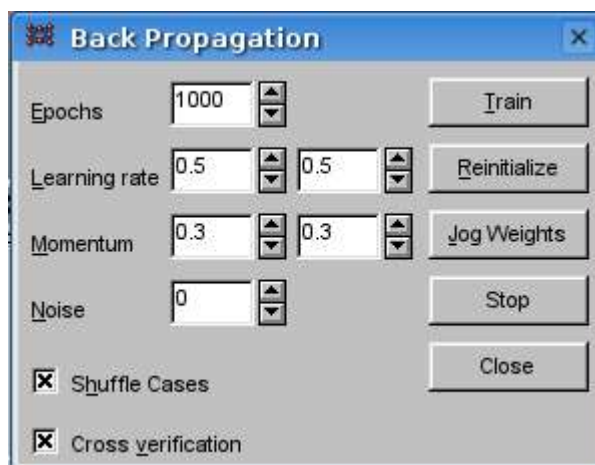
После обучения сеть смогла выполнить логическую операцию “Импликация” с ошибкой равной 0.25. Попробуем добиться лучших значений изменив структуру сети.

1. Создадим новую сеть типа “многослойный персептрон” (на том же наборе исходных данных):





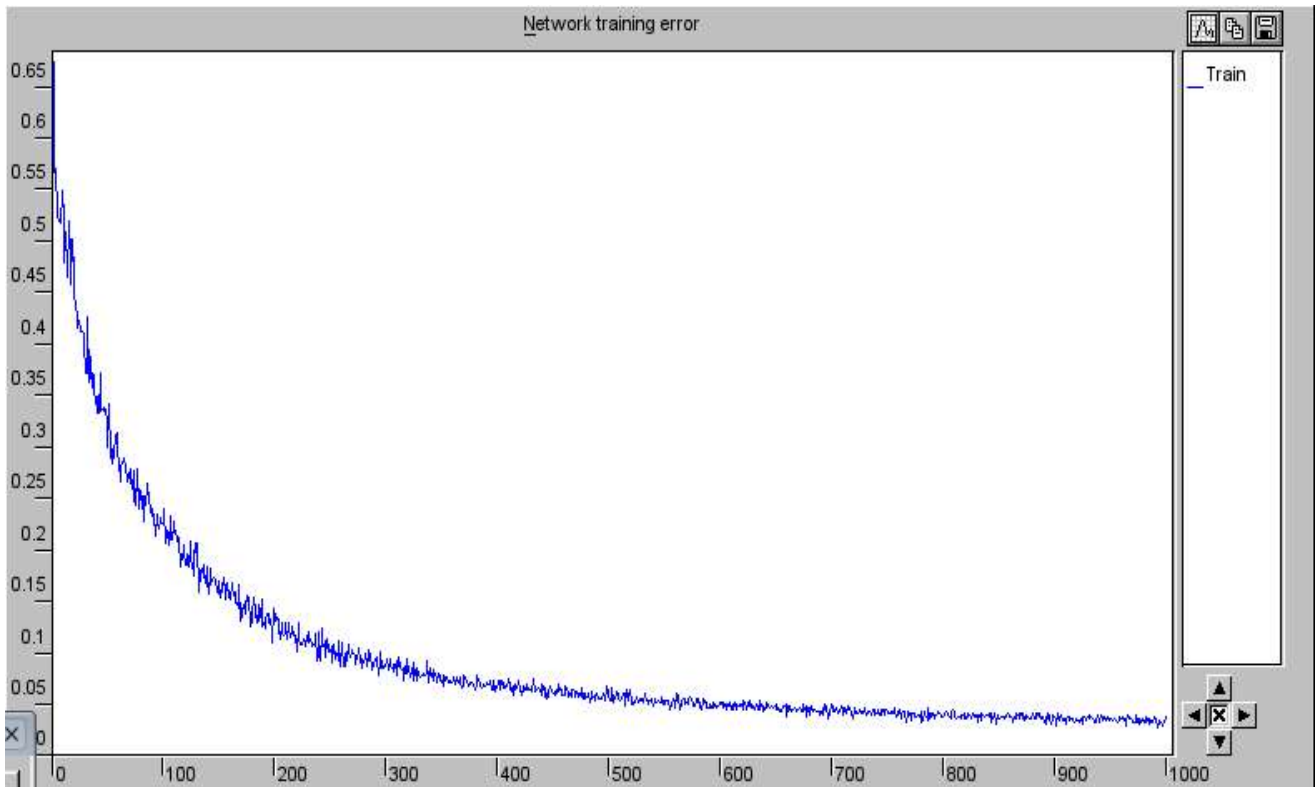
2. Проведем обучение нейронной сети методом обратного распространения (Train -> Multilayer Perceptrons -> Back propagation) с параметрами:



Здесь:

- Epochs – количество эпох обучения;
- Learning rate – степень (уровень) обучения;
- Momentum – показатель темпа обучения;
- Noise – шум;
- Shuffle Cases – перемешивание образов.

Полученный график ошибок обучения (Statistics -> Training Graph...):



3. Проверим результат обучения запуском на выполнение набора исходных данных (Run -> Data Set...):

The figure shows a "Run Data Set" dialog box. It includes a dropdown menu for "Outputs shown" set to "Variables", a "Run" button, and a "<-Data Set" button. Below these, the "RMS Error" is displayed as "Train 0.03301", "Verify 0", and "Test 0". A table with 5 columns and 5 rows is shown below. The first row is the header, and the following four rows contain data for samples 01 through 04.

	X*Y	T: X*Y	E: X*Y	Error
01	0.9995867	1	-0.0004133	0.0004133
02	1.060826	1	0.06083	0.06083
03	0.0162	0	0.0162	0.0162
04	0.9800858	1	-0.01991	0.01991

Здесь: первый столбец содержит результаты, полученные на выходе; второй столбец – значения, получаемые при выполнении логической операции “Импликация”; третий – отклонение полученного от искомого; четвертый – абсолютную величину ошибки.

Вывод: после обучения двухслойного персептрона в течение 1000 эпох ошибка составила 0,033, т.е. стала почти на порядок меньше, чем при использовании однослойного персептрона Розенблатта.

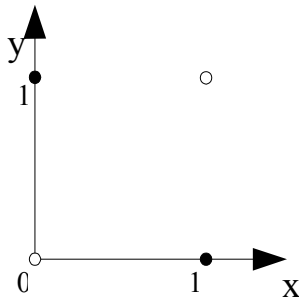
Задача 2: “Исключающее ИЛИ”

Постановка задачи:

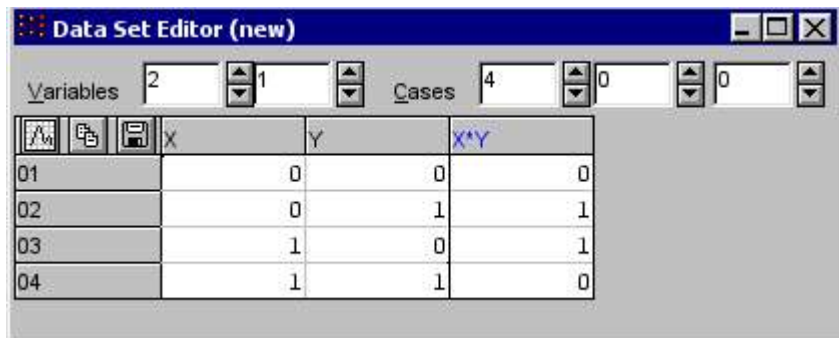
Распознавание логической операции “Исключающее ИЛИ” (XOR), заданной таблицей истинности.

Решение задачи:

При решении этой задачи встает проблема линейной неразделимости операции XOR: если значения входов нанести на оси графика, то станет ясно, что между этими точками невозможно провести линию, разделяющую точки одного образа - класса множества от точек другого класса.

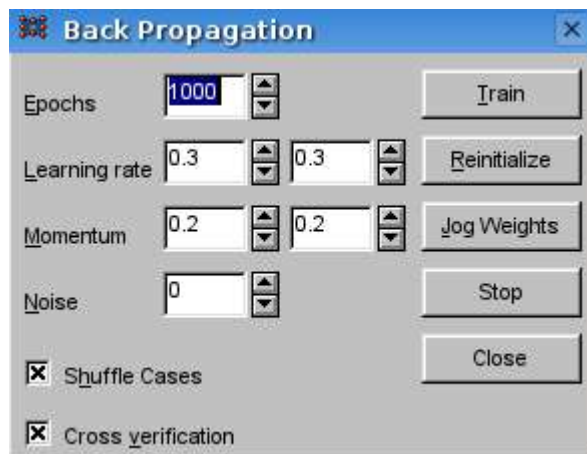


1. Создадим новый набор данных с помощью диалога “Data Set Editor”:

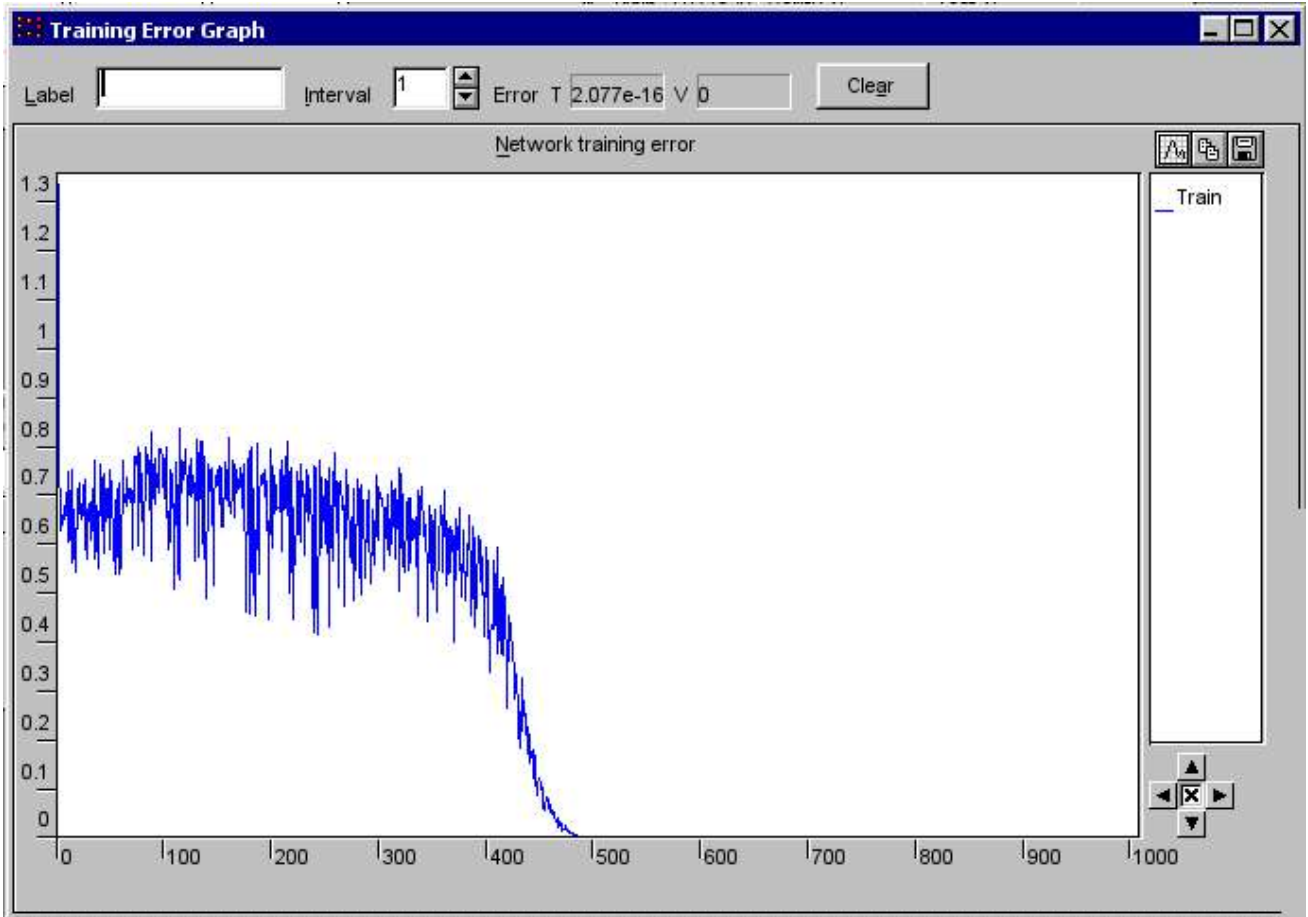


2. Создадим новую нейронную сеть, такую же, как и в предыдущем случае, т.е. двухслойный персептрон с двумя входами, двумя скрытыми нейронами и одним выходом.

3. Запустим обучение по методу обратного распространения со следующими параметрами:



В результате получен график ошибки обучения такого вида:



4. Проверим результат обучения:

The figure is a window titled "Run Data Set". It shows "Outputs shown" as "Variables", a "Run" button, and a "<-Data Set" button. Below this, it displays "RMS Error Train $2.077e-16$ Verify 0 Test 0". A table below shows the following data:

	X*Y	T. X*Y	E. X*Y	Error
01	$2.22e-16$	0	$2.22e-16$	$2.22e-16$
02	1	1	$-1.11e-16$	$1.11e-16$
03	1	1	$-3.331e-16$	$3.331e-16$
04	0	0	0	0

Вывод: Ошибка составила $2 \cdot 10^{-16}$ – стремится к нулю, сеть успешно обучена.

Задача 3: “Сократ”

Постановка задачи:

Обучение нейронной сети с целью обработки номинальных данных, построения логических выводов и формирования понятий на основе примеров:

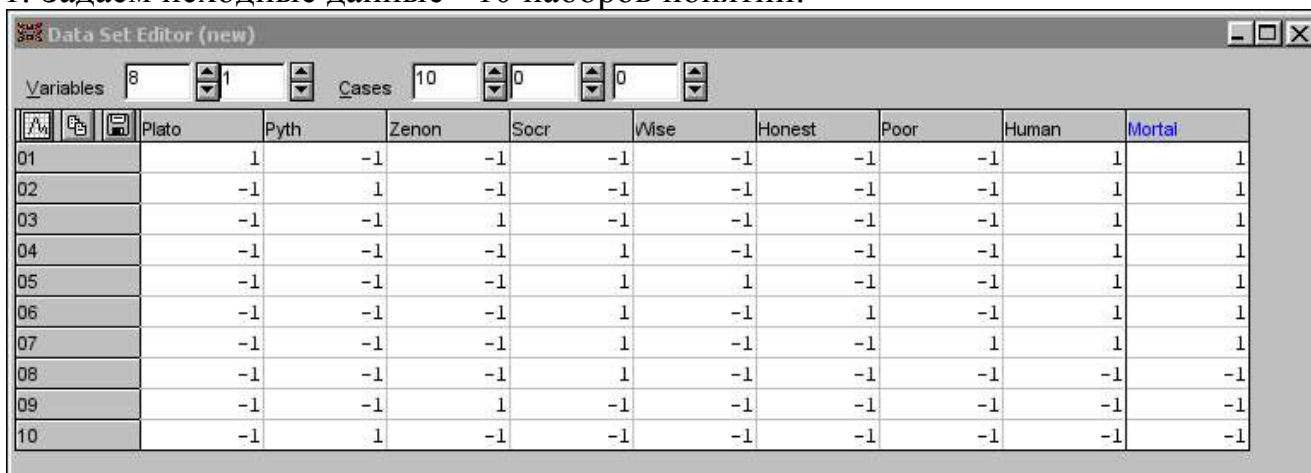
Решение задачи:

Требуется создать и обучить нейронную сеть на следующих примерах:

Платон – человек – смертен;
Пифагор – человек – смертен;
Зенон – человек – смертен;
Сократ – человек – мудрый;
Сократ – человек – честный;
Сократ – человек – бедный.

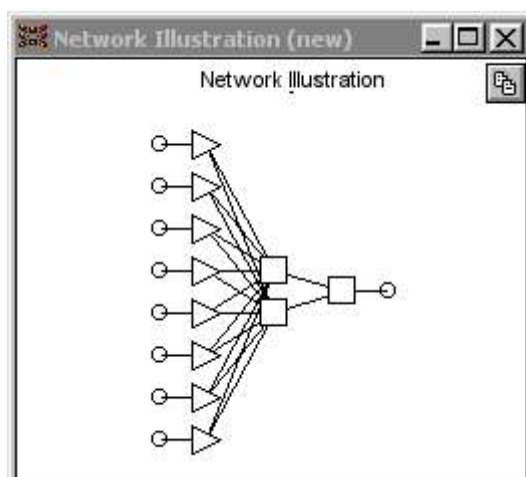
В результате на основе посылки “Сократ – человек” сеть должна выдать решение: “смертен”.

1. Задаем исходные данные - 10 наборов понятий:



	Plato	Pyth	Zenon	Socr	Wlse	Honest	Poor	Human	Mortal
01	1	-1	-1	-1	-1	-1	-1	1	1
02	-1	1	-1	-1	-1	-1	-1	-1	1
03	-1	-1	1	-1	-1	-1	-1	-1	1
04	-1	-1	-1	1	-1	-1	-1	1	1
05	-1	-1	-1	1	1	-1	-1	1	1
06	-1	-1	-1	1	-1	1	-1	1	1
07	-1	-1	-1	1	-1	-1	1	1	1
08	-1	-1	-1	1	-1	-1	-1	-1	-1
09	-1	-1	1	-1	-1	-1	-1	-1	-1
10	-1	1	-1	-1	-1	-1	-1	-1	-1

2. Создаем новую сеть типа “многослойный персептрон”, состоящую из 8 входных нейронов, одного выходного и одного скрытого слоя с двумя нейронами:



3. Проводим обучение по методу обратного распространения параметрами:

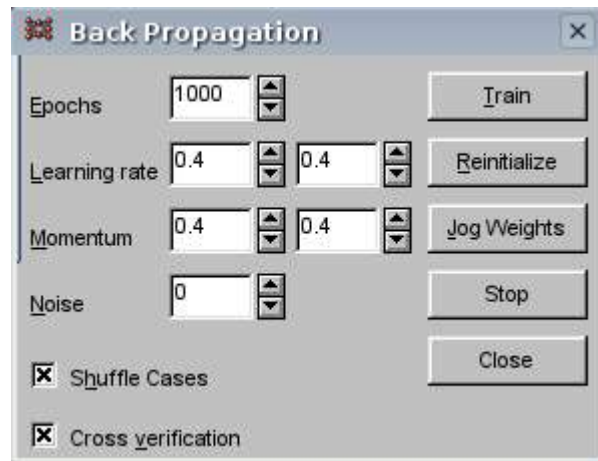
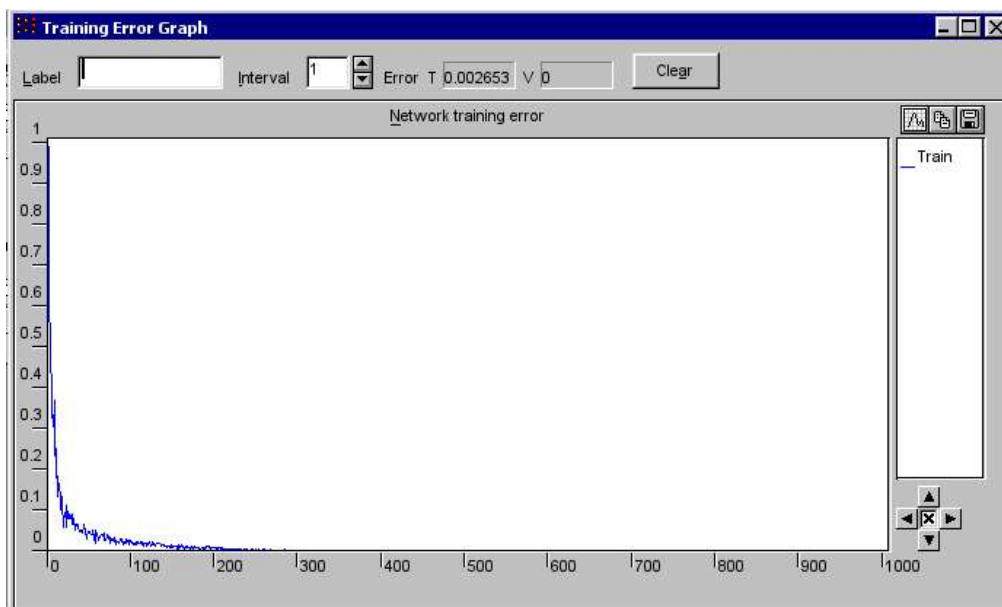
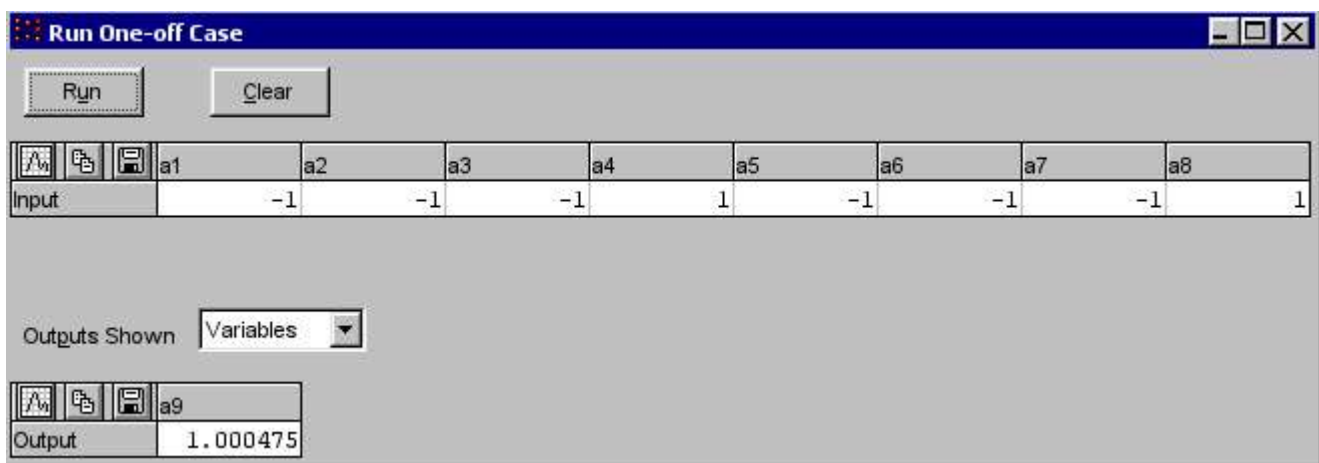


График ошибки обучения:



4. Проверим эффективность обучения, подавая на вход единичную посылку (Run-> One-off...) “Сократ – человек”:



Вывод: Ошибка составила примерно $5 \cdot 10^{-4}$ – сеть успешно обучена.

Задача 4: “Realty Estate” (Управление недвижимостью)

Постановка задачи:

Задача оценки объектов недвижимости.

Решение задачи:

Объекты недвижимости оценивались по 17 входным параметрам:

DWLUN – число

RDOS – обратная дата продажи (число месяцев с момента продажи)

YRBLT – год постройки

TOTFIXT – число точек водоснабжения и канализации

HEATING – система отопления

WBFPSKTS – камины

BMNTGAR – подвал/гараж

ATTFRGAR – смежный гараж

TOTLIVAR – общая жилая площадь

DECK/OFP – закрытая/открытая площадь участка

ENCLPOR – площадь примыкающего участка

NBHDGRP – соседи (код)

RECROOM – площадь комнаты отдыха

FINBSMT – полная площадь подвала

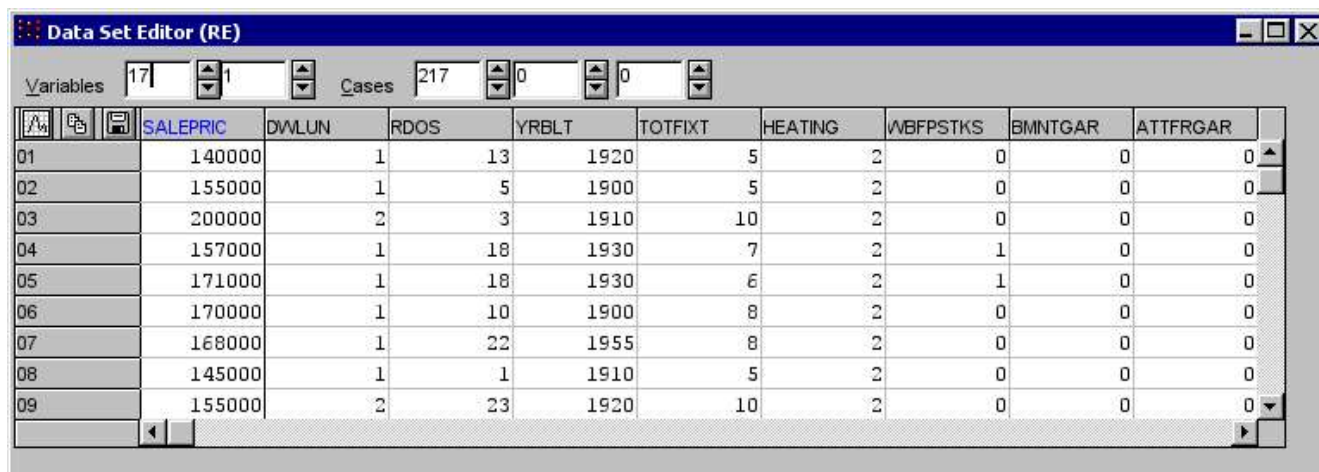
GRADE% - поправочные коэффициенты

CDU – состояние/привлекательность/удобство

TOTOBY – общая добавочная стоимость (здание и участок)

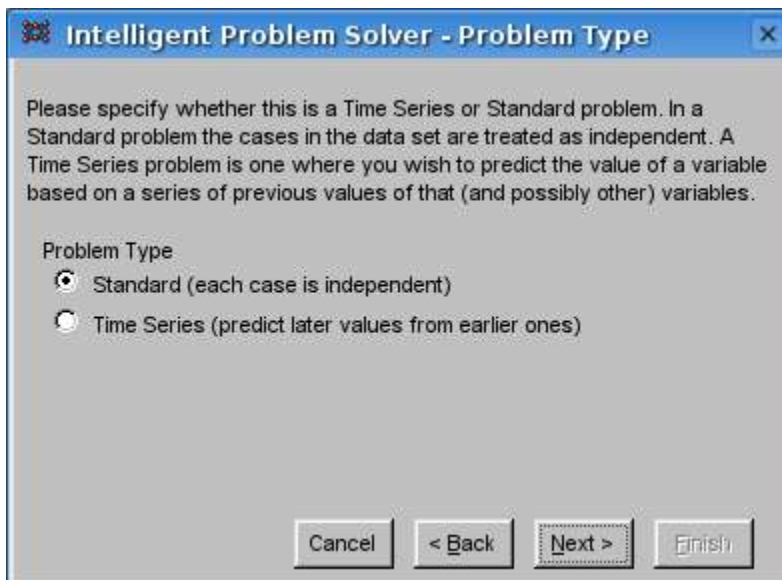
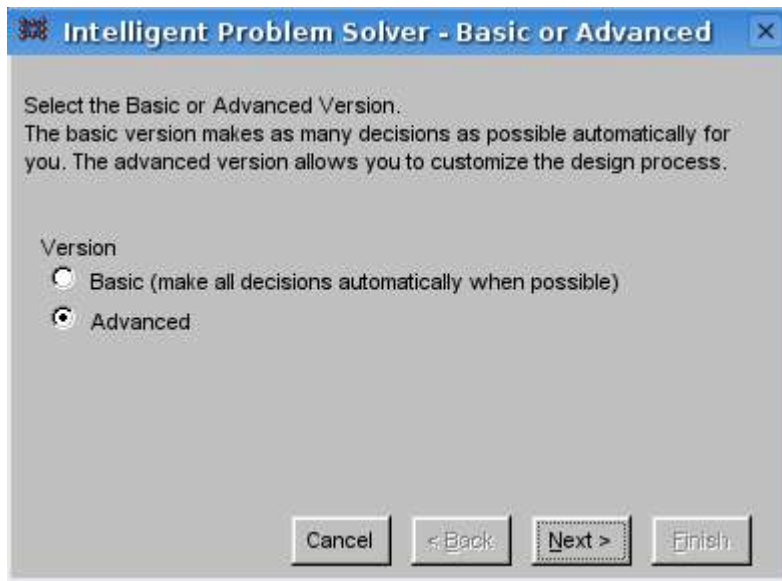
Выходной параметр SALEPRIC – цена, за которую был продан дом.

1. Открываем файл исходных данных RE.PAT (File -> Open):

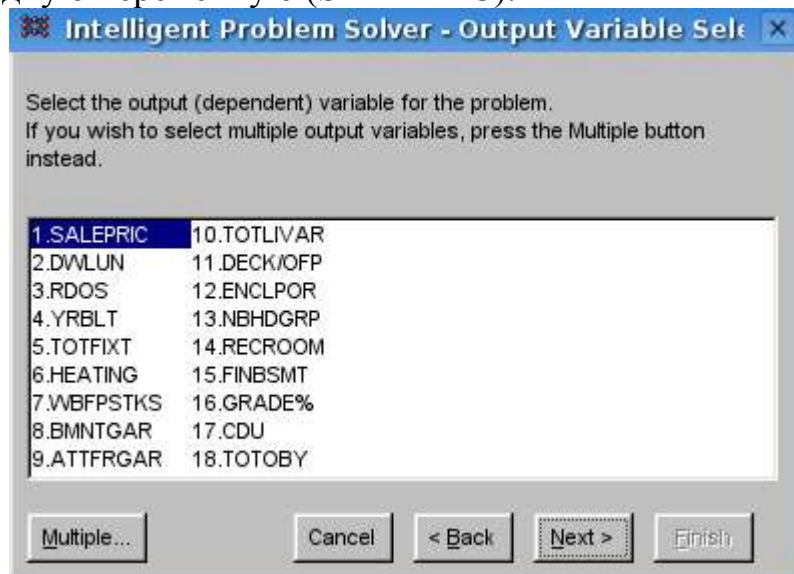


	SALEPRIC	DWLUN	RDOS	YRBLT	TOTFIXT	HEATING	WBFPSKTS	BMNTGAR	ATTFRGAR
01	140000	1	13	1920	5	2	0	0	0
02	155000	1	5	1900	5	2	0	0	0
03	200000	2	3	1910	10	2	0	0	0
04	157000	1	18	1930	7	2	1	0	0
05	171000	1	18	1930	6	2	1	0	0
06	170000	1	10	1900	8	2	0	0	0
07	168000	1	22	1955	8	2	0	0	0
08	145000	1	1	1910	5	2	0	0	0
09	155000	2	23	1920	10	2	0	0	0

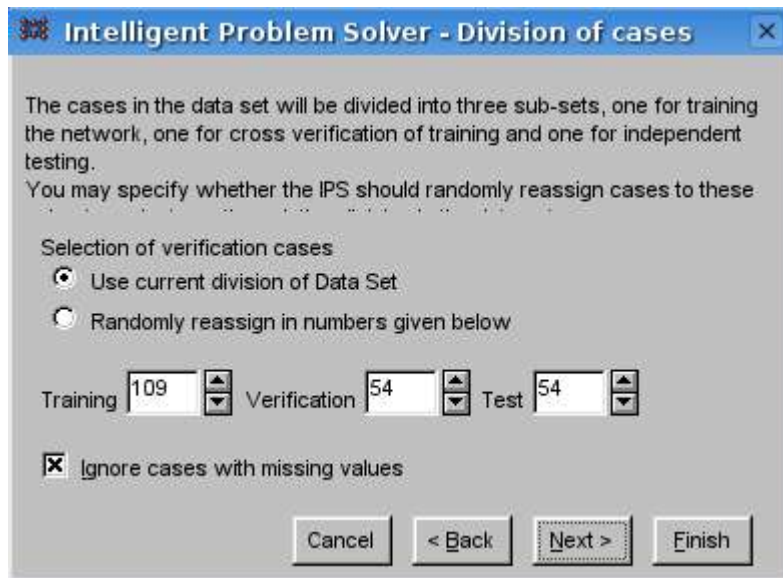
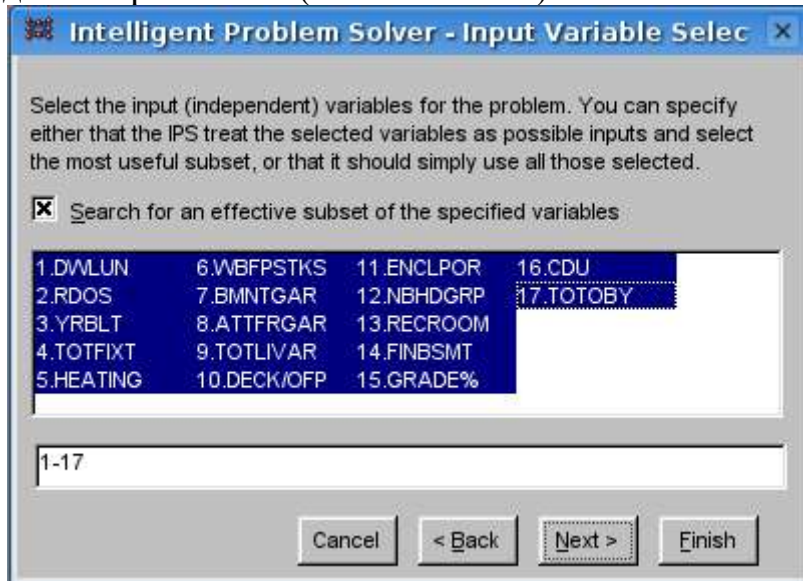
2. Запускаем мастер нахождения решения (File -> New -> Intelligent Problem Solver...):



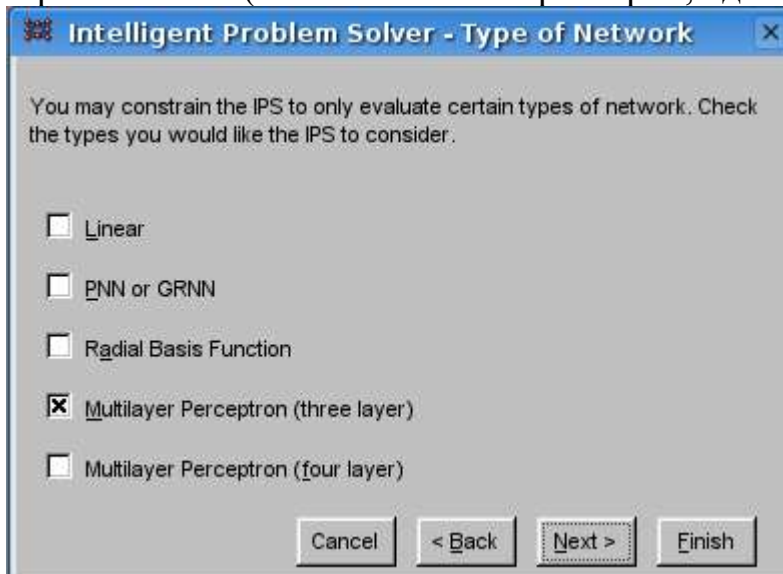
Выбираем выходную переменную (SALEPRIC):



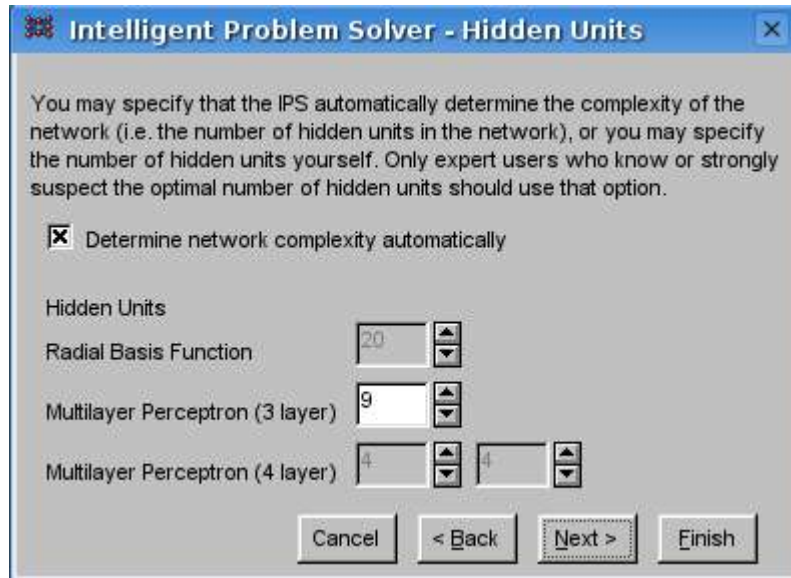
Выбираем выходные переменные (все остальные):



Выбираем тип нейронной сети (многослойный перцептрон, один скрытый слой):



Запускаем процесс поиска лучшей структуры сети:



3. Мастер поиска нашел наилучшую структуру.

Результат работы мастера:

```
Intelligent Problem Solver
(select Options/Detailed Feedback for more detailed, technical feedback)
Input variables will be selected automatically
Network complexity will be determined automatically
The best 10 networks found will be retained
Starting the search now...
0:00:17 An improved network has been found
0:00:56 An improved network has been found
0:02:16 An improved network has been found
...search has finished
16 networks were tested, 10 retained
The best network found had O.K. performance (regression ratio 0.391815,
correlation 0.920044)
```

Получена сеть следующего вида: 9 входов, 3 скрытых нейрона, 1 выход:

Type	Error	Inputs	Hidden	Performance
01	MLP 15944.45	1	1	0.5385741
02	MLP 15929.03	1	1	0.5380544
03	MLP 15402	2	5	0.5202493
04	MLP 15387.11	2	5	0.519749
05	MLP 14225.81	6	4	0.4805227
06	MLP 13448.87	6	4	0.4542789
07	MLP 13071.07	5	6	0.4415175
08	MLP 12721.2	5	6	0.4296995
09	MLP 11692.64	9	3	0.3949563
10*	MLP 11599.63	9	3	0.3918151

4. Оценим с помощью этой сети цену дома, данные о котором приведены в файле REIN.PAT (Run -> One-off...):

The screenshot shows the 'Run One-off Case' dialog box. It has a 'Run' button and a 'Clear' button. Below them is a table of input variables:

	DWLUN	YRBLT	TOTFIXT	HEATING	ATTFRGAR	TOTLIVAR	NBHDGRP	FINBSMT
Input	2	1950	5	2	0	1300	1	200

Below the table, there is a section for 'Outputs Shown' with a dropdown menu set to 'Variables'. Underneath, there is a table showing the output:

	SALEPRIC
Output	171077.3

Оценка ошибки при обработке исходных данных $1.967 \cdot 10^4$:

The screenshot shows the 'Run Data Set' dialog box. It has a dropdown menu for 'Outputs shown' set to 'Variables', a 'Run' button, and a '<-Data Set' button. Below these, there are fields for 'RMS Error Train' (1.967e+04), 'Verify D', and 'Test D'. The main part of the dialog is a table with the following data:

	SALEPRIC	T. SALEPRIC	E. SALEPRIC	Error
01	136071.7	140000	-3928.341	0.02672
02	127328.1	155000	-27671.86	0.1882439
03	179992.6	200000	-20007.4	0.1361047
04	136629.5	157000	-20370.54	0.1385751
05	141972.1	171000	-29027.94	0.197469
06	140413.3	170000	-29586.71	0.2012701
07	150135.4	168000	-17864.64	0.1215282
08	135736.5	145000	-9263.508	0.06302
09	157196.8	155000	2196.82	0.01494
10	169719.4	167000	2719.438	0.0185
11	178093	172000	6093.023	0.04145
12	173569.9	172000	1569.919	0.01068
13	135954.9	155000	-19045.11	0.1295586
14	201848.3	232000	-30151.66	0.2051133
15	206133.9	240000	-33866.14	0.2303819
16	161686.8	170000	-8313.216	0.05655
17	137382.8	139900	-2517.163	0.01712

5. Обучим сеть по методу обратного распространения со следующими параметрами:

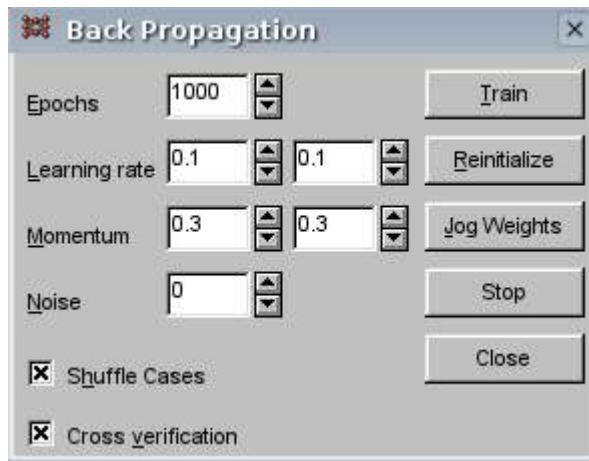
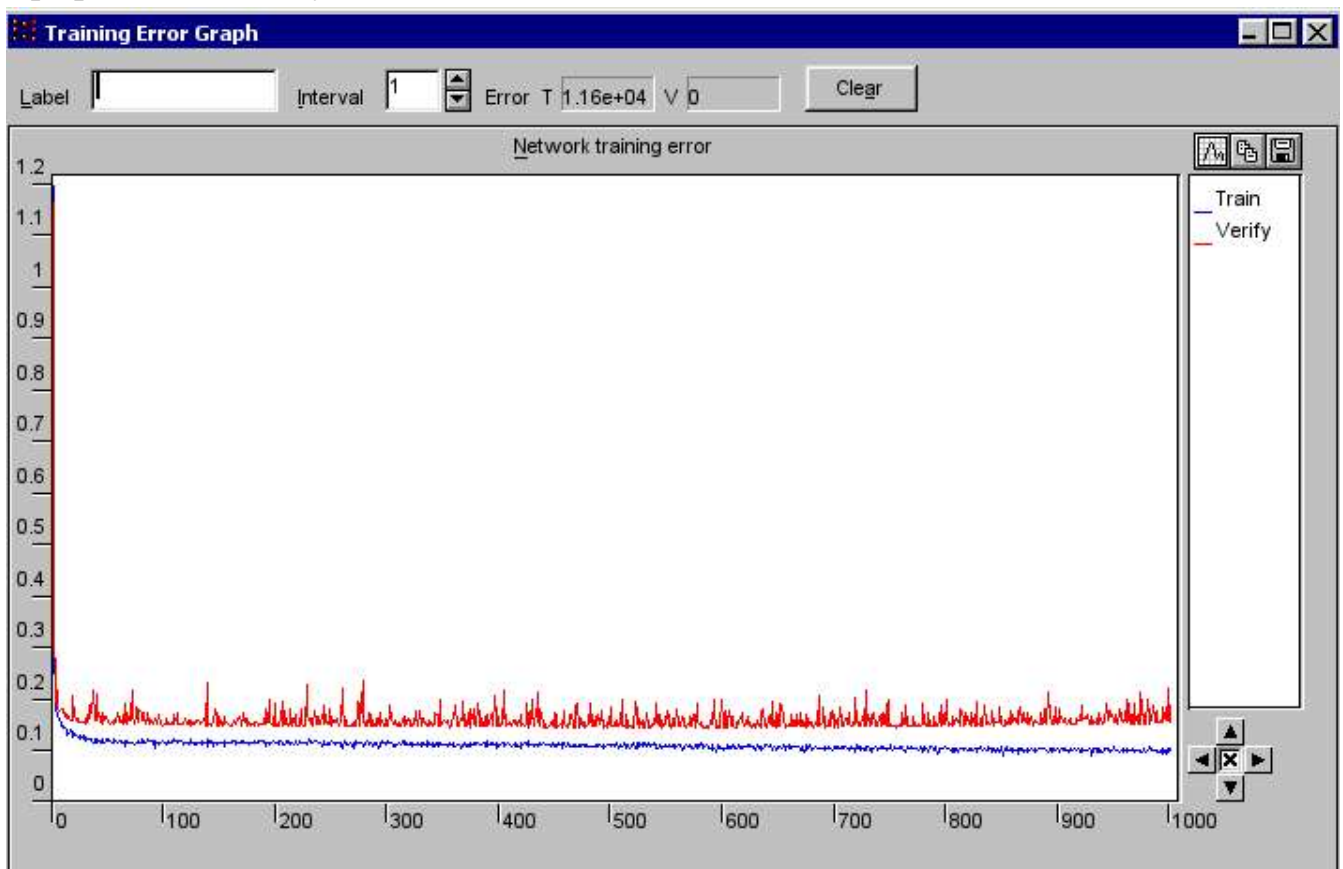
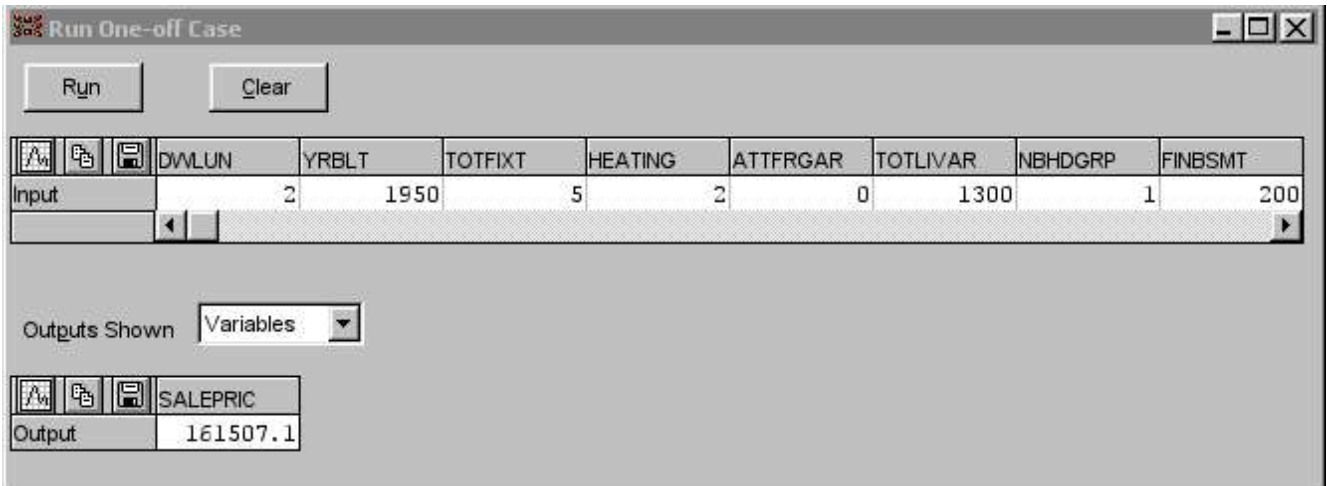


График ошибки обучения:



6. Запустим повторную проверку на данных примера:



Ошибка при обработке исходных данных $1,373 \cdot 10^4$, т.е. примерно в полтора раза меньше:

Run Data Set

Outputs shown Variables Run <-Data Set

RMS Error Train 1.373e+04 Verify 0 Test 0

	SALEPRIC	T. SALEPRIC	E. SALEPRIC	Error
01	155154.1	140000	15154.14	0.1030894
02	145577.9	155000	-9422.068	0.0640957
03	203106.1	200000	3106.094	0.02113
04	157529.4	157000	529.4339	0.003602
05	163577.9	171000	-7422.059	0.0504902
06	160847.4	170000	-9152.551	0.06226
07	171812.6	168000	3812.577	0.0259359
08	154401	145000	9401.02	0.06395
09	180194.5	155000	25194.52	0.1713913
10	191252.1	167000	24252.09	0.1649802
11	201551.5	172000	29551.46	0.2010303
12	195538.9	172000	23538.89	0.1601285
13	155038.4	155000	38.36824	0.000261
14	226254.6	232000	-5745.402	0.03908
15	232614.9	240000	-7385.071	0.05024
16	184406.2	170000	14406.23	0.098
17	156073.3	139900	16173.28	0.1100223

Вывод: Нейронная сеть обучена, ошибка составила приблизительно 10%