

Кириченко А.А.

Нейропакеты

– современный

интеллектуальный

инструмент

исследователя

2013

Кириченко А.А.

«Нейропакеты – современный интеллектуальный инструмент исследователя», 2013.
Сетевое электронное издание учебного пособия. 297 страниц, 450 рисунков, формат PDF.

ISBN 978-5-9904911-1-3

Всё, что связано с использованием нейронных сетей получило название нейросетевых технологий (нейрокомпьютинг). Они не требуют программирования, а предусматривают работу по обучению нейронной сети на специально подобранных примерах. На этапе обучения формируются основные отношения между входными параметрами и оформляются в незримые таблицы (образы), которые впоследствии используются при решении задач на сети.

Нейрокомпьютинг предоставляет единую методологию решения очень широкого круга практически интересных задач, как правило - ускоряющих и удешевляющих разработку приложений. В число таких задач входят прогнозирование цен, оценка кредитоспособности, оптическое распознавание, обработка изображений, диагностика, лингвистический анализ, и др.

Использование нейросетей для решения перечисленных задач предусматривает выполнение типовой последовательности действий с помощью нейрокомпьютеров, или нейропакетов.

В книге рассматриваются возможности нейропакетов, использование их для моделирования разнообразных нейронных систем, настройка и обучение универсальных нейропакетов на решение задач с использованием трёх наиболее доступных пакетов: BrainMaker, Deductor Academic, IBM SPSS Neural Network.

Книга предназначена для учащихся старших классов, студентов, бакалавров, магистров, аспирантов.

Кириченко А.А. профессор кафедры архитектуры программных систем Федерального государственного автономного образовательного учреждения высшего профессионального образования "Национальный исследовательский университет "Высшая школа экономики" при правительстве РФ".

ISBN 978-5-9904911-1-3

© Кириченко А.А., 2013

Содержание.

Лекция 1. Назначение и возможности нейросетей. Задачи факультатива.....	3
Лекция 2. НейроЭВМ и нейропакеты.....	10
Лекция 3. Концепция нейросетевого исследования.....	24
Лекция 4. Deductor - флагманский продукт компании BaseGroup Labs.	38
Лекция 5. Deductor. Нейронная сеть. Настройка и обучение.....	46
Лекция 6. Самоорганизующиеся карты Кохонена.....	73
Лекция 7. Нейросетевое исследование в статистическом пакете SPSS.....	95
Практикум.....	117
ПЗ1. Нейросетевая декомпозиция решаемой задачи.....	117
ПЗ 2. Работа с нейропакетом BrainMaker.....	128
ПЗ 3. Подготовка табличных данных.....	154
ПЗ 4. Ирисы Фишера.....	178
ПЗ 5 Прогнозирование стоимости дома в пакете Deductor.....	181
ПЗ 6. Автоматическая классификация данных об ирисах Фишера с помощью карт Кохонена.....	201
ПЗ 7. Использование радиальной основной функции для классификации телекоммуникационных клиентов.....	212
Приложение 1. Описание пакета “BrainMaker 3.11 Pro”.....	228
Приложение 2. Использование Excel для подготовки данных к исследованию.....	262
Приложение 3. Поиск информации в Интернет.....	284
Приложение 4. Исходные данные для задачи «Ирисы Фишера».....	291
Литература.....	296

Лекция 1. Назначение и возможности нейросетей. Задачи факультатива.

Введение.

Проведение научного исследования чаще всего заключается в выявлении скрытых правил и закономерностей в наборах данных, формулировке гипотез и выявлении типовых структур. Для этого приходится использовать различные методы обнаружения (добычи) знаний: абстрагирование, ассоциативное объединение, классификацию, кластеризацию, анализ временных рядов, прогнозирование и др.

Человеческий разум не приспособлен для восприятия больших массивов разнородной информации. В среднем человек не способен улавливать более двух-трех взаимосвязей даже в небольших выборках.

Для расширения аналитических возможностей человека можно использовать методы традиционной статистики, эвристические решающие устройства на основе экспертных систем, семантический дифференциал, теорию решения изобретательских задач (ТРИЗ), нейронные сети.

Традиционная статистика решает аналогичные задачи, но она оперирует усредненными характеристиками выборки, которые часто являются фиктивными величинами, например - средней платежеспособностью клиента, в то время, как необходимо уметь прогнозировать состоятельность и намерения конкретного клиента с учётом функции риска или функции потерь.

Методы математической статистики, эвристические решающие устройства, семантический дифференциал, ТРИЗ, так же, как и статистика относятся к дискретным методам. Для человека же несвойственно использовать при решении жизненных проблем дискретные методы.

Естественным для человека является использование основных принципов мозга — ассоциативное мышление, использование принципов обучения (самообучения) и адаптации, связей «если - то», «посылка - следствие», лежащих в основе распознавания, движения, управления, принятия решений.

Поэтому из различных способов расширения аналитических возможностей человека наиболее эффективными при исследовании задач, не имеющих общепризнанного алгоритма решения, является использование нейронных сетей.

Всё, что связано с использованием нейронных сетей получило название нейросетевых технологий.

Нейросетевые технологии не требуют программирования, а предусматривают работу по обучению нейронной сети на специально подобранных примерах.

Основной функцией обучения нейросети, воспроизводящей работу мозга и ассоциативное мышление является узнавание, умение определять сходство и различия.

На этапе обучения формируются основные отношения между входными параметрами и оформляются в незримые таблицы (образы), которые впоследствии будут использоваться при решении задач на сети.

Какие задачи решают нейросети.

Нейросети наиболее приспособлены к решению широкого круга задач, так или иначе связанных с обработкой образов. Вот список типичных математических постановок задач для нейросетей:

- Аппроксимация функций по набору точек (регрессия).
- Узнавание, классификация данных по заданному набору классов.
- Кластеризация данных с выявлением заранее неизвестных классов-прототипов.
- Сжатие информации.
- Восстановление утраченных данных.
- Ассоциативное преобразование информации.

Этот список можно было бы продолжить и дальше. За этими задачами просматривается некий единый прототип, позволяющий при известной доле воображения сводить их друг к другу. Чаще всего - это типичные примеры некорректных задач, т.е. задач не имеющих единственного решения, алгоритмы которых неизвестны или не имеют строгого обоснования.

Нейрокомпьютинг предоставляет единую методологию решения очень широкого круга практически интересных задач. Это, как правило, ускоряет и удешевляет разработку приложений.

В число таких задач входят прогнозирование цен, оценка кредитоспособности, оптическое распознавание (например, подписи), обработка изображений, диагностика, лингвистический анализ, и др.

Наверное, в каждой предметной области при ближайшем рассмотрении можно найти постановки нейросетевых задач. Список областей, где решение такого рода задач имеет практическое значение уже сейчас, очень широк:

- Экономика и бизнес: предсказание рынков, автоматический дилинг, оценка риска невозврата кредитов, предсказание банкротств, оценка стоимости недвижимости, выявление пере- и недооцененных компаний, автоматическое рейтингование, оптимизация портфелей, оптимизация товарных и денежных потоков, автоматическое считывание чеков и форм, безопасность транзакций по пластиковым карточкам.

- Политические технологии: анализ и обобщение социологических опросов, предсказание динамики рейтингов, выявление значимых факторов, объективная кластеризация электората, визуализация социальной динамики населения.
- Безопасность и охранные системы: системы идентификации личности, распознавание голоса, лиц в толпе, распознавание автомобильных номеров, анализ аэрокосмических снимков, мониторинг информационных потоков, обнаружение подделок.
- Ввод и обработка информации: Обработка рукописных чеков, распознавание подписей, отпечатков пальцев и голоса. Ввод в компьютер финансовых и налоговых документов.

Нейросети - это не что иное, как новый инструмент анализа данных. И лучше других им может воспользоваться именно специалист в своей предметной области.

Основные трудности на пути еще более широкого распространения нейротехнологий - в неумении широкого круга профессионалов формулировать свои проблемы в терминах, допускающих простое нейросетевое решение, т.е. неумение проводить нейросетевую декомпозицию решаемой задачи.

Основные классы задач, решаемых с помощью нейропакетов.

1. Узнавание (Классификация)
2. Кластеризация
3. Регрессия (прогнозирование, предсказание)
4. Понижение размерности

В задачах регрессии целью является оценка значения числовой (принимаящей непрерывный диапазон значений) выходной переменной по значениям входных переменных.

В задачах анализа временных рядов целью является **прогноз** будущих значений переменной, зависящей от времени, на основе предыдущих значений ее и/или других переменных.

Как правило, прогнозируемая переменная является числовой, поэтому прогнозирование временных рядов - это частный случай регрессии. Однако такое ограничение часто в пакет не закладывается, так что в нем можно прогнозировать и временные ряды номинальных (т.е. классифицирующих) переменных.

В задаче узнавания сеть должна отнести каждое наблюдение к одному из нескольких классов (или, в более общем случае, оценить вероятность принадлежности наблюдения к каждому из классов).

Задачи классификации (кластеризации) заключаются в группировке похожих образов в классы (кластеры). В нейропакетах они решаются с помощью сети СОКК (самоорганизующаяся карта Кохонена).

Самоорганизующаяся карта Кохонена (СОКК или сеть Кохонена) принципиально отличается от всех других типов сетей. В то время как все остальные сети предназначены для задач с управляемым обучением (т.е. обучением с учителем), сети Кохонена главным образом рассчитаны на неуправляемое обучение (без учителя).

В алгоритмах неуправляемого обучения значения весов и/или порогов меняются на основании только входных обучающих данных (выходные значения не требуются и, если они присутствуют, игнорируются).

Наиболее общая возможность *понижения размерности* - это анализ главных компонент, который часто в нейропакетах реализован в виде оператора «Что – если?» - он может помочь извлечь небольшое число самых основных компонент из исходных данных довольно большой размерности, сохранив при этом неизменной структуру результатов.

Нейросетевая технология исследований.

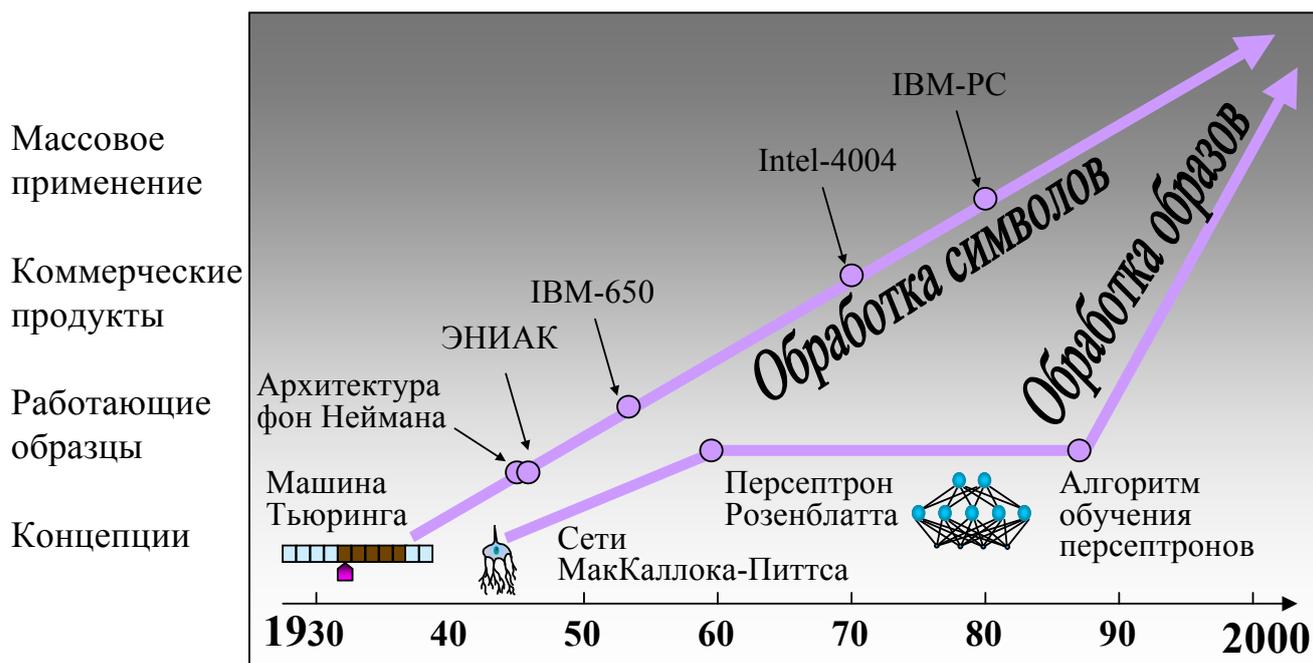
Использование нейросетей для решения перечисленных задач предусматривает выполнение типовой последовательности действий:

1. Получение исходных данных.
2. Отбор входных данных и понижение размерности.
3. Оцифровка данных (шкалирование, преобразование номинальных значений, и др.)
4. Выбор подходящей архитектуры сети.
5. Обучение нейронной сети.
6. Тестирование нейронной сети.
7. Получение готового решения

Что собой представляет нейрокомпьютинг.

Исторически электронные вычислительные машины (компьютеры) развивались в виде трёх разновидностей: Цифровые, Аналоговые и Адаптивные (самонастраивающиеся, дообучающиеся).

Выпуск аналоговых ВМ полностью прекратился к 1980 году. Развитие двух оставшихся видов ВМ иллюстрируется следующим графиком:



При конструировании ВМ основное внимание уделялось цифровым вычислительным машинам.

Адаптивные ВМ представляли собой альтернативную, полностью параллельную архитектуру обработки информации -- «по образу и подобию» биологических нервных систем – *нейрокомпьютинг*. Датой рождения этой науки принято считать 1943 год, в котором появилась статья МакКаллока и Питтса о вычислениях в сетях формальных нейронов, хотя работы по созданию адаптивных систем велись ещё в 30-е годы прошлого столетия.

Вначале развития нейрокомпьютинга [1] шло в направлении создания и использования нейросетевых программных продуктов, рынок которых включал в себя

нейропакеты общего назначения, системы разработки нейроприложений, готовые решения на основе нейросетей, нейроконсалтинг. Впоследствии к этому рынку добавилась работа по созданию нейроЭВМ. Характеризовать секторы рынка нейросетевых программных продуктов можно следующим образом:

Сегмент рынка нейро-продуктов	Преимущества продукции	Недостатки продукции
Нейро-пакеты общего назначения	Не требуют самостоятельного программирования, легко осваиваются, инструмент быстрого и дешевого решения прикладных задач	Не способны к расширению, не способны генерировать отчуждаемые приложения, не могут использоваться для разработки сложных систем или их подсистем
Системы разработки нейроприложений	Могут использоваться для создания сложных систем обработки данных в реальном времени (или их подсистем)	Требуют навыков программирования, более глубокого знания нейросетей
Готовые решения на основе нейросетей	Не предполагают знакомства пользователя с нейросетями, предоставляют комплексное решение проблемы	Как правило - дорогое удовольствие
Нейро-консалтинг	Не предполагает участия пользователя в получении прогнозов, потенциальная дешевизна услуг	Нет возможности дополнить предсказания своим know how. Доступность конфиденциальной информации

Нейропакеты общего назначения нацелены на решение информационных задач в диалоговом режиме - при непосредственном участии пользователя. Они не применимы в условиях потоковой обработки данных. Кроме того, они не приспособлены для разработки сложных систем обработки данных, состоящих из многих блоков, содержащих, скажем, сотни нейросетей,

К ним относятся такие пакеты, как BrainMaker Professional, NeuroForecaster, Лора-IQSOO, Штутгартский симулятор для UNIX-машин.

Коммерческие пакеты отличаются от свободно распространяемых большим набором средств импорта и предобработки данных, дополнительными возможностями по анализу значимости входов и оптимизации структуры сети. Стоимость коммерческих эмуляторов - масштаба \$1000. Как правило, такие пакеты (BrainMaker Professional, NeuroForecaster, Лора-IQ300) имеют собственный встроенный блок предобработки данных, хотя иногда для этой цели удобнее использовать стандартные программные средства типа электронных таблиц.

Так, нейро-продукты группы нейрокомпьютинга ФИАН встраиваются непосредственно в Microsoft Excel в качестве специализированных функций обработки данных. При этом всю предобработку данных и визуализацию результатов можно проводить стандартными средствами Excel, который, кроме того, имеет богатый и расширяемый набор конверторов для импорта и экспорта данных. Пакет прикладных программ Excel Neural Package, использовал в качестве функций активизации гиперболический тангенс, а в качестве алгоритма обучения — алгоритм Rprop.

Удобным инструментом разработки сложных нейросистем является MATLAB с прилагающимся к нему нейросетевым инструментарием, органично вписавшимся в матричную идеологию этой системы. MATLAB предоставляет удобную среду для синтеза нейросетевых методик с прочими методами обработки данных (wavelet-анализ, статистика, финансовый анализ и т.д.).

Разработанные в системе MATLAB приложения могут быть затем перетранслированы в C++.

Полностью переведен на русский язык нейросетевой программный продукт *STATISTICA Neural Networks*. Он является средой анализа нейросетевых моделей и соответствует самым современным инструментам.

Инструменты разработки нейроприложений - главное, что отличает этот класс программного обеспечения - способность генерировать "отчуждаемые" нейросетевые продукты, т.е. генерировать программный код, использующий обученные нейросети для обработки данных. Такой код может быть встроен в качестве подсистемы в любые сколь угодно сложные информационные комплексы.

Примерами подобных систем, способных генерировать исходные тексты программ являются NeuralWorks Professional II Plus (стоимостью от \$3000) фирмы NeuralWare и отечественный Neural Bench (нейро-верстак). Последний интересен, кроме прочего, тем, что может генерировать коды на многих языках, включая Java.

Готовые решения на основе нейросетей - это - конечный результат. Здесь нейросети спрятаны от пользователя в недрах готовых автоматизированных комплексов, предназначенных для решения конкретных производственных задач.

Например, такой продукт, как Falcon встраивается в банковскую автоматизированную систему обслуживания платежей по пластиковым карточкам.

В другом случае такая система может выполнять функции автоматизированной системы управления заводом или реактором. Конечного пользователя, как правило, не интересует способ достижения результата, ему важно лишь качество продукта

Поскольку многие такие готовые решения обладают уникальными возможностями (пока специалисты по нейрокомпьютерингу еще в дефиците) и обеспечивают реальные конкурентные преимущества, их цена может быть довольно высока - гораздо выше, чем стоимость нейроЭВМ (neuro-hardware).

Тем не менее, возможность перевода алгоритмов на нейросетевую основу является очень ценной, поскольку в случае реального применения алгоритмов на практике, нейросетевой подход позволяет получать высокоэффективные параллельные архитектуры при аппаратной реализации алгоритмов в устройствах.

Нейросетевой консалтинг используется вместо того, чтобы продавать готовые программы либо инструменты для их разработки. Некоторые задачи, например такие, как предсказание рыночных временных рядов, являются настолько сложными, что доступны лишь настоящим профессионалам. Не каждая компания может позволить себе издержки, ассоциируемые с передовыми научными разработками (например, постоянное участие в международных конференциях). Поэтому приобретают популярность фирмы, единственной продукцией которых являются исследования рынков. При большом числе клиентов цена таких предсказаний может быть весьма умеренной.

Примером здесь может служить Prediction Company, основанная в 1991 году физиками Дойном Фармером и Норманом Паккардом. Продукция компании пользуется большим успехом среди Швейцарских банков, скупающих прогнозы "на корню" для игры на фондовых и валютных рынках.

Фирма Richard Borst, торгующая недвижимостью, применяет предельно дешевый ("университетский") нейропакет для уточнения оценки выставляемых на продажу домов и квартир. Как свидетельствуют старожилы фирмы, внедрение нейропакета (стоимостью всего \$300) увеличило оборот фирмы в Нью-Йорке и Пенсильвании на 6%.

В статье «Нейронная сеть – оружие финансиста» Андрей Масалович пишет: «Мой приятель Джим - журналист-аналитик, работающий в "информационной империи" McGraw-Hill. Весь его офис стоит менее \$2000, помещается в кармане и состоит из портативного palmtop-компьютера, факс-модема и нейропакета Brain Maker. Ежедневно Джим подключается к необъятным базам данных McGraw-Hill и при помощи специальной утилиты DataMaker "просеивает" через свой нейропакет мегабайты финансовой, экономической и прочей информации. После нескольких минут яростного перемалывания тысяч и тысяч разнородных цифровых параметров, нейропакет выдает

(кстати, в виде изящных таблиц Excel) прогноз ряда макроэкономических индикаторов - на завтра, на неделю и на месяц вперед. Попытка проделать такой объем аналитической работы вручную (даже с использованием вспомогательных программ корреляционного анализа), наверное, привела бы беднягу на больничную койку в первый же день».

Зачем эта дисциплина нужна студентам исследовательского университета?

Статус национального исследовательского университета обязывает иметь среди преподаваемых дисциплин методики проведения научных исследований – разнообразные, современные, новые, может быть даже – необычные.

К таким методикам относятся, например, построенные на использовании нейроЭВМ, нейронных сетей ЭВМ, нейроматематики, в простейших случаях (и наиболее доступных) – на нейропакетах.

Такие методики позволяют решать интеллектуальные задачи в различных областях человеческих знаний. Например, специалисты – политологи (политобозреватели) заняты решением задачи, связанной с просмотром огромного количества материалов (новостей) из разных источников за последние сутки. Читать эти новости можно непрерывно. Но нужно среди них отобрать самые важные. Для этой цели разработаны нейропакеты, просматривающие огромный поток сообщений с электронной скоростью и отбирающие только те, которые представляют интерес для данного специалиста.

Поскольку основные задачи, решаемые с помощью нейропакетов, связаны с поиском скрытых закономерностей, классификацией, прогнозированием, сокращением размерности, область их применения охватывает все направления человеческих знаний – и технических, и гуманитарных.

Изучение возможностей нейропакетов, моделирование разнообразных нейронных систем, разработка программ для автоматизации выполнения интеллектуальных операций, настройка и обучение универсальных нейропакетов на решение определённых задач – вот далеко не полный перечень задач, стоящих перед бакалаврами и магистрами национального исследовательского университета.

В результате освоения дисциплины студент должен:

Знать

- Отличительные особенности задач, эффективно решаемых на нейросистемах, принципы обучения нейросетей. Устройство и принцип действия нейрокомпьютеров, нейронных компьютерных сетей, нейропакетов.

Уметь

- создавать нейронные сети и обучающие выборки для них, обучать нейропакеты решению задач классификации, прогнозирования, снижения размерности данных

Иметь навыки

- (приобрести опыт) проведения аналитических исследований на доступных нейропакетах.

Для освоения учебной дисциплины, студенты должны владеть следующими знаниями и компетенциями:

- Знания основ математической статистики и теории вероятностей;
- Знания в объёме курса «Информатика» бакалаврской подготовки;
- Знание английского языка (умение читать и переводить).

Тематический план учебной дисциплины:

№	Название раздела	Всего часов	Аудиторные часы			Самостоятельная работа
			Лекции	Семинары	Практические занятия	
1	Архитектура нейронных сетей, нейроЭВМ и нейропакетов	22	2		4	16
2	Задачи, решаемые нейронными сетями и архитектура сетей для этого	30	2		8	20
3	Алгоритмы обучения нейронных сетей	26	2		4	20
4	Моделирование нейронных сетей и технология проведения исследований	30	2		8	20
	Итого:	108	8		24	76

Критерии оценки знаний, навыков

- На текущем контроле студент должен продемонстрировать навыки использования пакета BrainMaker при проведении исследований.
- На итоговом контроле студент должен продемонстрировать навыки самостоятельного поиска исходных данных в Интернет и примененного в пакете Deductor или SPSS метода анализа для решения поставленной задачи, интерпретации и представления результатов анализа, формулировки выводов на основе проведенного анализа данных.

Тематика заданий текущего контроля

- Настройка пакета BrainMaker на распознавание десятичных цифр, текста, пространственного расположения объекта;
- Настройка интерфейса пакета BrainMaker;
- Формирование обучающей выборки (текст, графика, звук);
- Прогнозирование в IBM SPSS Neural Networks;
- Классификация и узнавание в нейронных сетях;
- Технология снижения размерности данных. Задача политобозревателя;
- Моделирование ассоциативной памяти в нейронных сетях;
- Формирование отчетности в нейропакетах.

Программные средства

Для успешного освоения дисциплины, студент использует статистические пакеты BrainMaker, Deductor и IBM SPSS Neural Networks

Лекция 2. НейроЭВМ и нейропакеты.

НейроЭВМ.

Универсальные ЭВМ неэффективны при решении трудноформализуемых задач – задач, алгоритм решения которых неизвестен. Они неспособны обучаться решению какой-либо задачи «на примерах». А человек и животные такой способностью обладают! Разница объясняется, видимо, тем, что принципиально отличаются методы решения задач, характерные для живых организмов и ЭВМ.

К числу задач, для решения которых ЭВМ не приспособлены, относятся:

- Задачи предсказания (прогнозирования);
- Нахождение скрытых закономерностей, причинно-следственных связей;
- Распознавание изображений, звуковой информации и иных образов (автоматическая классификация);
- Ассоциативная память;
- Сжатие и кодирование информации в реальном масштабе времени;

- И др.

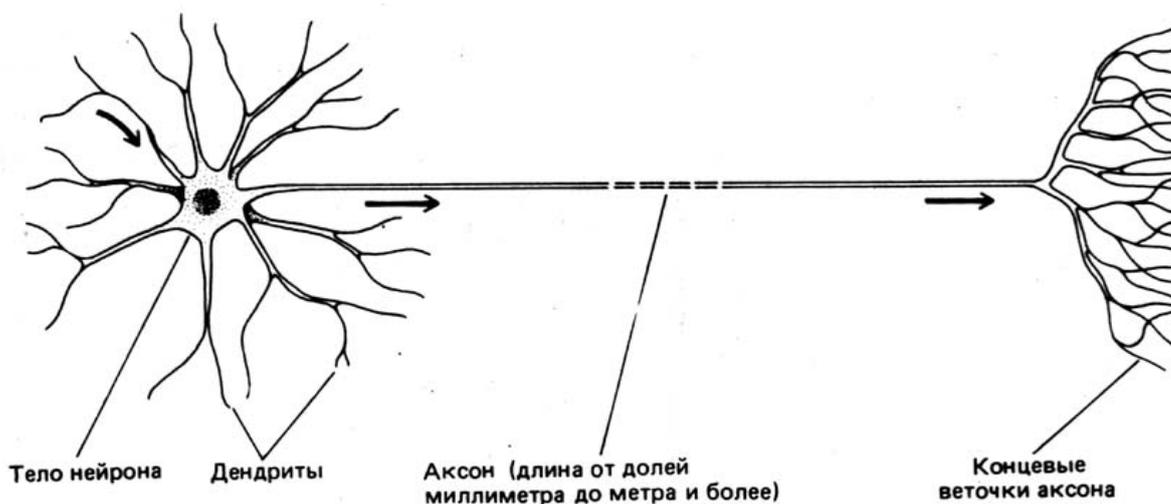
Точные алгоритмы решения таких задач чаще всего отсутствуют. А следовательно, невозможно составить программу для решения такой задачи.

Одновременно с разработкой методов распознавания, реализуемых на универсальных ЭВМ фон-Неймановской архитектуры, ведется поиск методов решения таких задач (в том числе – задач распознавания образов), основанных на иных принципах. Наиболее интересные результаты в этом направлении научных исследований получены при попытках создания распределенных самоорганизующихся систем, например, перцептрона Розенблатта, пандемониума Селфриджа, нейронных ЭВМ (сокращенно – нейроЭВМ), моделирующих работу нервных сетей живых организмов. На их основе в настоящее время разработаны и практически используются параллельные, волновые, матричные и др. системы распознавания.

Элементы, из которых состоит нейросеть, не отличаются разнообразием. Все они имеют одинаковую структуру. Каждый такой элемент суммирует приходящие на него сигналы, и если полученная сумма превышает пороговый уровень, на выходе элемента появляется выходной сигнал. В противном случае элемент не реагирует на входные сигналы. Такие элементы названы *нейронами*, а ЭВМ, построенные на основе таких элементов – нейрокомпьютерами.

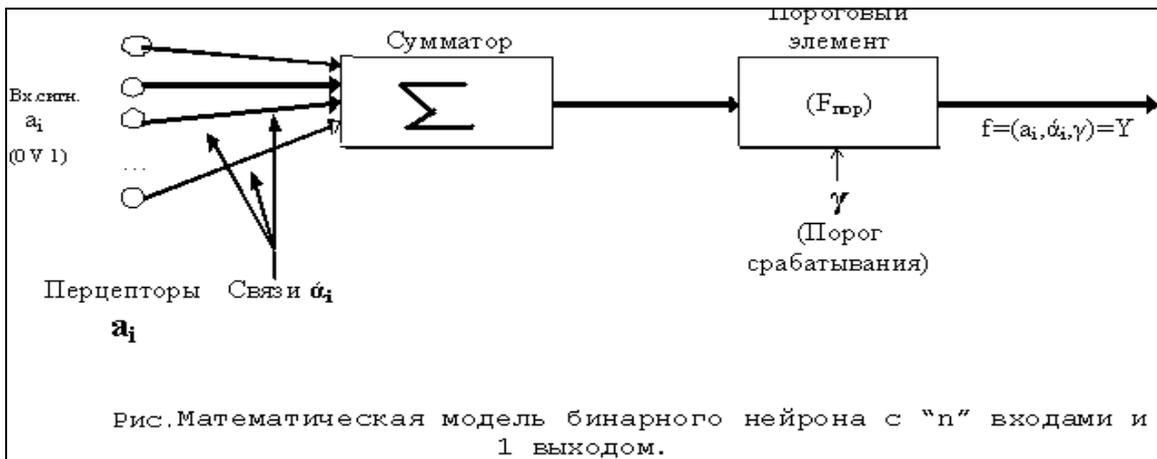
Для нейрокомпьютеров программы не нужны, не нужно и знание того, как решается задача. Их конструкция подсмотрена у природы: элементы, из которых состоит мозг человека и животных; и принципы обработки информации сильно отличаются от элементов и принципов, используемых в ЭВМ.

Нейрон (т.е. нервная клетка) живого организма представляет собой клетку, состоящую из тела клетки, ядра, находящегося в плазме, коротких отростков (дендритов), одного длинного отростка – аксона. Дендриты и аксоны имеют разветвления (синапсы), с помощью которых нейроны связаны друг с другом. Входами нейрона являются дендриты, выходом – аксон.



Работает нейрон так: входная информация поступает в него через дендриты и накапливается в теле нейрона. Если сумма поступивших сигналов превышает порог срабатывания, то в аксон выдается импульс, который является выходным сигналом нейрона. Такой нейрон называется *бинарным*.

Математическая модель бинарного нейрона представлена на рис.:



Пороговый элемент реализует следующую функцию:

$$F_{\text{пор}} = \begin{cases} 0, & \text{если } \Sigma \leq \gamma \\ 1, & \text{если } \Sigma > \gamma \end{cases}$$

Тогда уравнение бинарного нейрона (математическая модель) может быть представлена в следующем виде:

$$Y = F_{\text{пор}}(\sum a_i * \alpha_i, \gamma)$$

Необученный нейрон имеет одинаковые или случайно распределённые веса связей перцепторов с сумматором. Процесс обучения нейрона заключается в том, что на вход подается распознаваемый образ в виде комбинации сигналов a_i , где каждый a_i является двоичным сигналом и может принимать одно из двух значений: 0 или 1.

На выход нейрона подается эталонный сигнал Y_0 (обозначающий, что должно содержаться на выходе обученного нейрона).

Если нейрон выдает другой сигнал, отличный от эталонного, начинается процесс обучения, который может, например, проводиться по Дельта-правилу коррекции весов при обучении:

$$\alpha_i = \alpha_i + (Y_0 - Y) * a_i,$$

где:

- α_i – вес каждой связи перцептора с сумматором;
- a_i – входные сигналы;
- Y_0 – требуемый результат;
- Y – фактический результат.

Допустим, что $Y_0 = 1$, а $Y = 0$. Тогда корректируется вес связи α_1 :

$\alpha_1 = \alpha_1 + a_1$ (поощрение связи), после чего опять проверяется соотношение сигналов Y_0 и Y .

Если $Y_0 \neq Y$, то выполняется $i=i+1$ и корректируется вес связи α_i . Такая коррекция весов связей будет происходить, пока нейрон не станет адекватно реагировать на входной образ.

Для случая, когда $Y_0 = 0$, а $Y = 1$ коррекция производится в сторону уменьшения весов связей (наказание): $\alpha_1 = \alpha_1 - a_1$.

После обучения распознаванию данного образа на вход нейрона подается другой аналогичный образ, и обучение продолжается.

Весь интеллект нейрона заключен в весах связей перцепторов с сумматором.

Реальный нейрон на выход выдает «пачки» импульсов. Размер пачки зависит от степени превышения порога срабатывания: при очень сильном превышении порога нейрон генерирует не одиночный импульс, а их последовательность.

Количество распознаваемых нейронной сетью образов зависит от количества бинарных нейронов в сети – n , и определяется по формуле: 2^n .

После обучения один бинарный нейрон способен распознать два образа (1 образ + 1 нераспознанная ситуация).

Два нейрона – 4 образа ($2^2 = 4$).

Три нейрона – 8 образов ($2^3 = 8$).

Четыре нейрона – 16 образов ($2^4 = 16$).

А в мозгу человека насчитывается от 10^{10} до $1.4 \cdot 10^{10}$ нейронов.

Для распознавания большого количества образов нейроны группируются в **нейронные сети** (нс).

На рис. приведен пример простейшей нейронной сети:

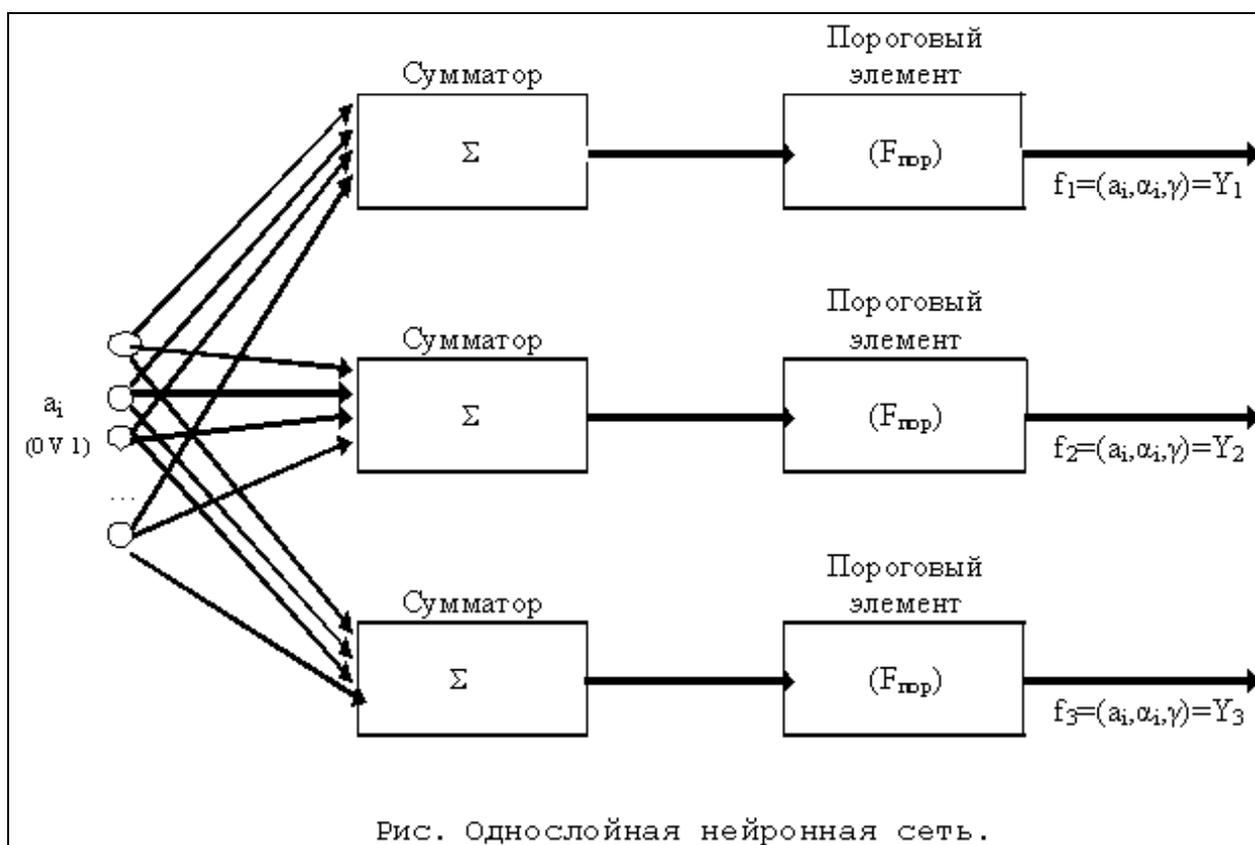
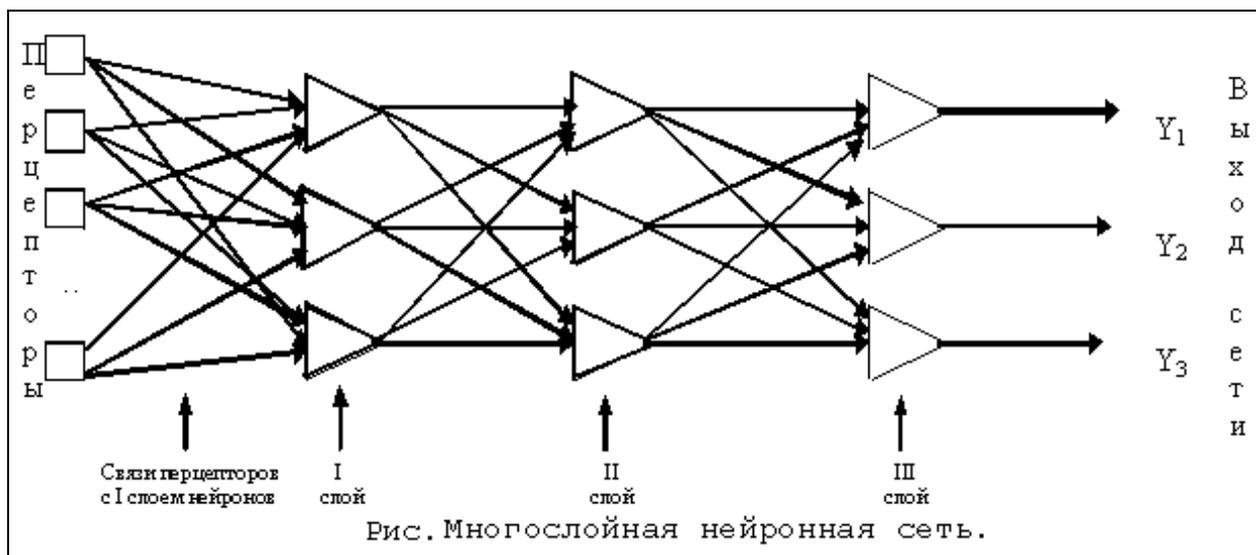


Рис. Однослойная нейронная сеть.

Такая нейронная сеть называется *однослойной*.

Многослойная нейронная сеть изображена на следующем рисунке (на этом рисунке каждый элементарный нейрон обозначен треугольником, так как интерес представляет не внутренняя структура нейрона, а связи между нейронами сети). Поскольку слои легко различимы, такая сеть называется *слоистой*.



В отличие от слоистой сети, в *полносвязной* сети выход каждого нейрона подается на входы всех остальных нейронов (в том числе – и на свой собственный вход). Из нейронов образуется клубок, в котором различить слои практически невозможно.

Нейронные сети, содержащие одинаковые нейроны, называются *гомогенными*, или однородными.

Сети, построенные из разных нейронов, называются *гетерогенными*, или неоднородными.

Интеллектуальные возможности нейронной сети зависят от количества нейронов и от количества связей между ними. Причем количество связей в сложных нейросетях значительно больше, чем количество нейронов (в мозгу человека приблизительно в 1000 раз). Поэтому, машины, построенные на принципах нейросетей, называются *коннекционистскими* (connection machines).

На основе бинарных моделей нейронов построены элементы, из которых создаются либо отдельные устройства, либо целиком ЭВМ – нейрокомпьютеры. Их особенность заключается в отсутствии необходимости составлять программы для решения задач. Для нейрокомпьютеров необходимы сборники тестовых примеров с решениями, на которых проводится обучение нейрокомпьютера.

При обучении нейросети, сеть НЕ запоминает поступившие на вход образы, не формирует библиотеку эталонных образов – у нейросети НЕТ для этого памяти. При обучении изменяются веса связей перцепторов каждого нейрона с его сумматором. Именно в весах связей сосредоточен интеллект нейросети. Если взять из обученной нейросети веса связей и сообщить эти веса необученной сети, необученная сеть сразу становится обученной, способной решать данную задачу.

Таким образом, нейроЭВМ отличаются от универсальных ЭВМ:

- принципом действия: это однородная система, состоящая из большого числа очень простых элементов;
- нейроЭВМ не требуют программирования; нужно обучение на специально подобранных примерах, составляющих обучающую выборку;
- способностью к обучению и самообучению (способность к самообучению начинает появляться при наличии около 100 нейронов);
- способностью решать задачи на примерах, отсутствовавших в обучающей выборке – при этом если нейросистема обучена распознавать образы, то полученный навык она сохраняет при предъявлении ей части объекта или при показе ей объекта, повернутого под другим углом;
- мощность нейроЭВМ определяется количеством нейронных слоев, количеством нейронов в каждом слое и количеством связей между слоями, между нейронами и перцепторами;

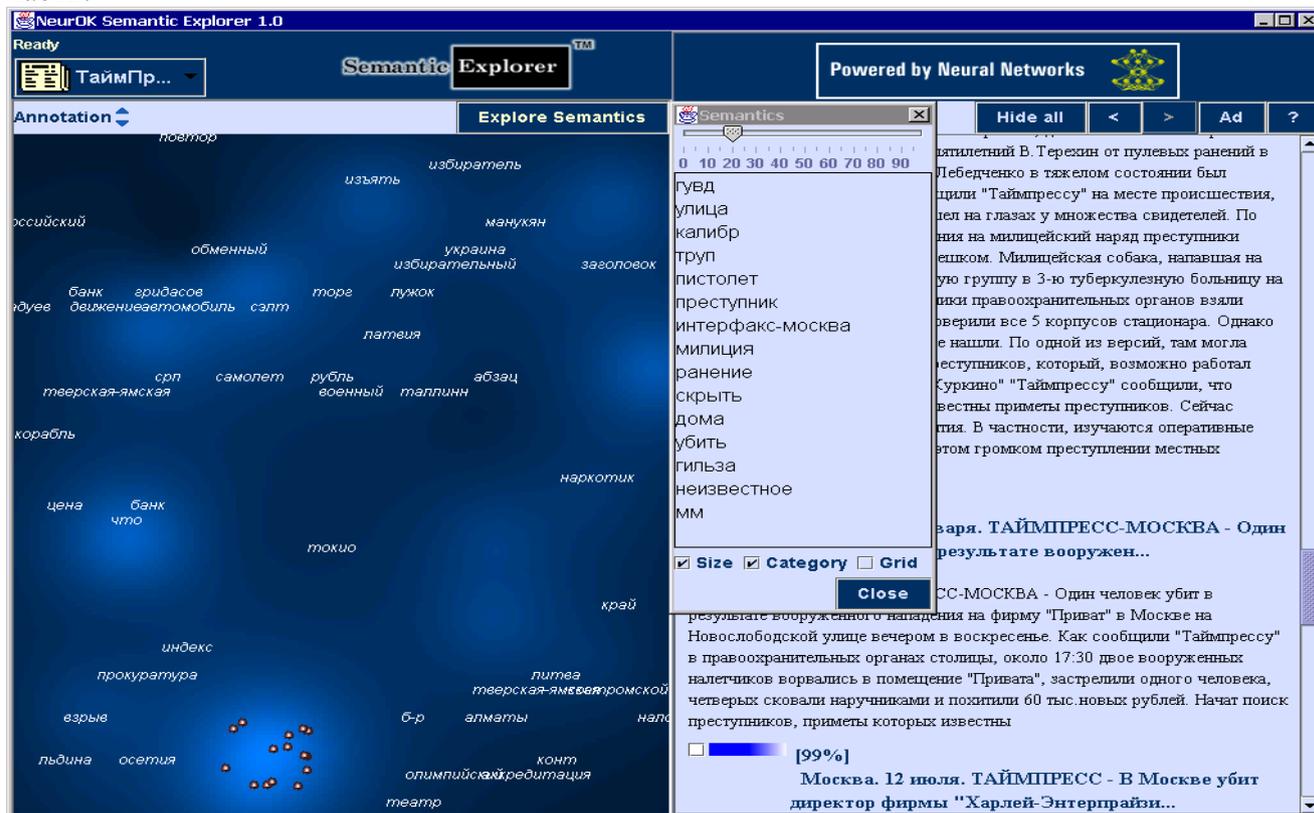
- нейросистема сохраняет работоспособность при разрушении до 15% нейронов;
- в отличие от универсальных ЭВМ, нейроЭВМ не умеют выполнять арифметические вычисления.

НейроЭВМ имеют высокую стоимость. Поэтому предпринимаются попытки реализовать алгоритмы работы нейросети на универсальных ЭВМ. Такие программы называются нейроимитаторами, когнитронами или нейропакетами.

Каждый **нейропакет** представляет собой комплекс программ, моделирующих работу нейроЭВМ. Обычно в пакет входят: **нейроконструктор** – программа, конструирующая нейронную сеть; **обучающая программа** и **решающая программа**. Кроме того, в комплект могут входить программы для исследования эффективности нейронной сети.

Широкое распространение получили такие пакеты, как NeuroSolution – профессиональный пакет для решения практических задач, BrainMaker – более простой пакет, пригодный для первоначального обучения (русифицированная демоверсия этого пакета была доступна в Интернет: ftp://ftp.parnas.spb.ru/pub/WinWinNT/neuro//brainmaker).

Хорошей иллюстрацией возможностей нейропакета является специализированный нейропакет Semantic Explorer, ориентированный на хранение и поиск данных в информационной системе (www.neurok.ru). Он имеет основное окно, разделенное на две части.



В левой половине расположены сгруппированные по смыслу названия находящихся в базе рефератов. Щелчок мышью по названию реферата приводит к его появлению на правой половине экрана. Искать аналогичные по смыслу рефераты просто, так как они уже находятся в левом поле на небольшом расстоянии от избранного – определение смысловой близости имеющихся информационных материалов пакет производит автоматически.

Для ускорения работы универсальной ЭВМ при решении задач с помощью нейроматематики используются нейронные ускорители, которые изготавливаются в виде плат и вставляются в стандартные слоты расширения универсальной ЭВМ.

Использование нейроЭВМ в экономике

ведётся по следующим направлениям:

1. Управление кредитными рисками (оценка надёжности заёмщика, определение вероятности убытков от несвоевременного возврата финансовых средств, оценка стабильности финансовых показателей предприятий – заёмщиков). С помощью нейроЭВМ удаётся определить 90% потенциальных неплательщиков (по данным 100тыс. случаев).
2. Предсказание ситуации на фондовом рынке с учётом постоянно изменяющихся условий торгов и краткости времени, отводимого на принятие решения.
3. Оценка недвижимости с учётом вида собственности, качества постройки, окружающей обстановки и других параметров. Для решения этой задачи на рецепторы нейросети подаются характеристики недвижимости (площадь, высота потолков, качество полов, покрытие стен, состояние сантехники и электрооборудования, наличие телефона, удалённость от метро, куда выходят окна – на проспект, или на лесо-парковую зону, тип здания, и т.д.), а на выходе нейросети появляется стоимость квартиры. Тестовые примеры для обучения берутся из опыта работы риелтерских фирм (показатели, характеризующие недвижимость – и стоимость её на рынке). Нейросеть можно использовать, как имитационную систему для определения, при каких условиях цена будет максимальной или минимальной. Для этого надо посмотреть, что будет, если пол не паркетный, а ковролиновый, если окна выходят на реку, а не на проспект, и т.д.
4. Исследование стратегии и тактики экономического управления предприятием (например, минимизация налогов с предприятия). На вход подать экономические показатели деятельности предприятия – на выходе получить сумму налоговых отчислений. Затем менять экономические показатели, наблюдая за изменением суммы налоговых отчислений.
5. Прогнозирование динамики биржевых курсов.
6. Моделирование и прогнозирование банкротства.
7. Автоматическое распознавание чеков.
8. Прогнозирование состояния безработицы в стране.
9. Прогнозирование уровня риска при страховании.

Для примера приведём некоторые особенности, реализованные в пакете Статистика.

Виды сетей:

- Многослойный перцептрон
- Радиальная базисная функция
- Вероятностная нейронная сеть
- Обобщенная регрессионная нейронная сеть
- Линейная сеть
- Самоорганизующаяся карта Кохонена (СОКК)

Алгоритмы обучения

- Обратное распространение
- Быстрое распространение
- Метод сопряженных градиентов
- Алгоритм Левенберга-Маркара
- Квази-ньютоновский алгоритм
- Дельта-дельта с чертой
- Алгоритм Кохонена
- ОВК (обучающийся векторный квантователь)

- Псевдообратный метод (сингулярное разложение)
- Метод К-средних
- Алгоритмы задания отклонений

Основные классы задач

1. Регрессия
2. Классификация
3. Прогнозирование
4. Кластеризация
5. Понижение размерности

Особые возможности Statistica NN

- Мастер решений
- Генетический алгоритм
- Ансамбли
- Повторный выбор
- Анализ чувствительности
- Восстановление наилучшей сети
- Матрица потерь

Нейропакеты.

Практическое знакомство с нейропакетами и принципом действия нейроЭВМ можно осуществить на примере пакета **Brainmaker**.

Пакет Brainmaker предназначен для построения нейронных сетей обратного распространения. Пакет включает в себя программу подготовки и анализа исходных данных (netmaker.exe), программу обучения и запуска нейросетей (brainmak.exe), а также набор утилит широкого назначения. Программный пакет ориентирован на широкий круг задач от создания прогностических приложений до организации систем распознавания образов и нейросетевой памяти.

Входные и выходные данные могут быть представлены в программе в числовом и символьном видах, а также в виде матричной графики.

Нейропакет использует 3 основных типа файлов: *файлы определений*, *файлы фактов* и *файлы сети*. Все эти файлы имеют текстовый формат и могут быть созданы и отредактированы в текстовом редакторе типа «Блокнот».

Файл определений содержит всю необходимую информацию об интерфейсе пользователя, типах входных и выходных данных, параметрах обучения и т. п. По умолчанию файл имеет расширение **def**.

Файл фактов содержит обучающие, тестирующие и рабочие факты, которые будут использоваться построенной сетью. По умолчанию они имеют расширения соответственно **fct**, **tst**, **in**.

Файл сети создается программой в процессе обучения и содержит текущие параметры, такие как конфигурацию и веса связей в сети. По умолчанию файл имеет расширение **net**.

Файлы определений и фактов создаются программой netmaker, или вручную, за исключением случаев, когда входом или выходом является растр точек, в этом случае файл создается только вручную.

Помимо перечисленных основных файлов программный пакет генерирует множество *других типов файлов*. Эти файлы могут содержать выходные данные сети и сопровождающую информацию (out), статистику обучения (sts) и тестирования (sta), файлы отчетов по исследованиям зависимостей (rpt), отдельные параметры созданной сети (ext) и др. Эти файлы порождаются программой brainmak.

Для полноценного и быстрого обучения сети наиболее значимым является качество обучающих данных. Практически во всех случаях, использование плохо подготовленной обучающей выборки приводит к значительным ошибкам в результатах. Более того, сеть может вообще никогда не научиться правильно решать требуемую задачу.

Состав пакета BrainMaker следующий:

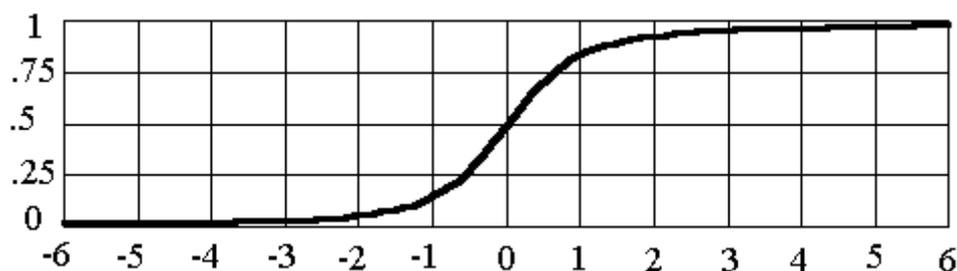
- NetMaker – анализ данных и подготовка обучающей выборки;
- BrainMaker – обучение НС, анализ обученной сети, работа с обученной сетью;
- Utilities – анализ файлов пакета и другой сервис.

Сетевая организация пакета “BrainMaker 3.11 Pro”

Данный нейропакет предназначен для создания прикладных нейронных сетей на основе многослойного перцептрона. Нейронная сеть может содержать один входной слой, один выходной, и от 0 до 5 скрытых (промежуточных) слоёв.

В качестве функции возбуждения нейроны каждого слоя могут реализовать следующие функции:

Sigmoid – сигмоидальная функция. Ее графическое представление:



Для нейронной сети (НС), включающей нейроны с сигмоидальной функцией нелинейности, при возбуждении на выходе нейрона появляется сигнал:

$$f(\alpha) = \frac{1}{1 + e^{-(\alpha - \Theta)}}$$

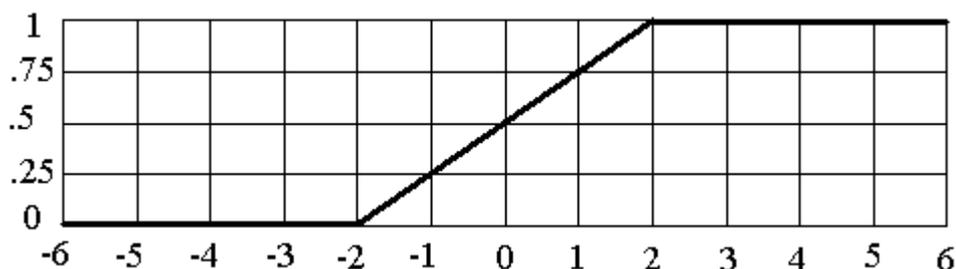
где

$f(\alpha)$ – сигнал на выходе нейрона;

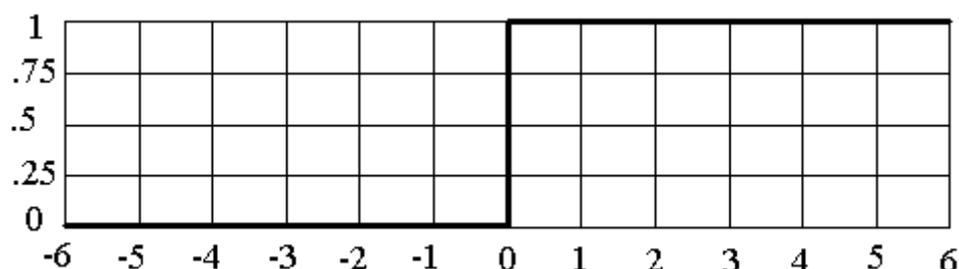
α - взвешенная сумма возбуждений на входах нейрона;

Θ - фиксированное значение порога.

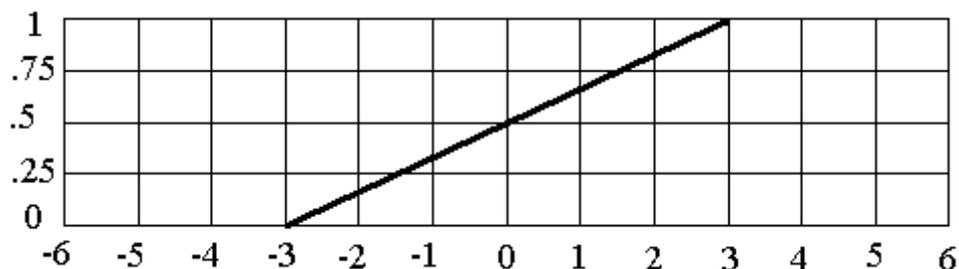
Threshold – линейная пороговая функция. Ее графическое представление:



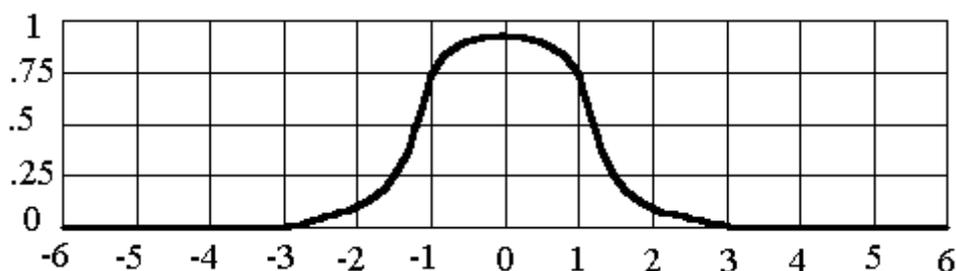
Step – бинарная функция возбуждения. Ее графическое представление:



Linear – линейная функция возбуждения. Ее графическое представление:



Gaussian – гауссовая. Ее графическое представление:



Алгоритм обучения.

В качестве алгоритма обучения пакет Brain Maker использует метод с обратным распространением ошибки.

Этот метод предусматривает передачу информации об ошибочном срабатывании нейронов в направлении, обратном направлению распространения возбуждения. Метод получил широкое распространение для организации обучения в нейронных сетях различного типа – импульсных, статических, дискретных и аналоговых.

Алгоритм модификации весов синапсов включает ряд шагов:

ШАГ 1. Инициализация весов и порогов.

Устанавливаем малые случайные величины для весов синапсов и порогов нейронов.

ШАГ 2. Предъявление входной реализации и эталона на выходе сети.

Вводятся непрерывные величины компонент входного вектора x_0, x_1, \dots, x_{N-1} и соответствующего эталона выходного вектора y_0, y_1, \dots, y_{M-1} . Если сеть используется как классификатор, тогда все каналы выхода устанавливаются равными 0 за исключением канала, соответствующего требуемому классу, которому присваивается значение 1. Входные реализации предъявляются циклически, пока не стабилизируются веса синапсов.

ШАГ 3. Вычисление действительных выходов.

Используя сигмоидальную нелинейность, у которой

$$\alpha \cdot \Theta = \sum_{i=0}^{N-1} w_{ij} \cdot x_i - \Theta_i,$$

вычисляем y_0, y_1, \dots, y_{M-1} .

ШАГ 4. Адаптация весов.

Используя рекурсивный алгоритм и стартуя от выхода нейронов, движемся к предыдущему слою нейронов. Модифицируем веса в соответствии с правилом:

$$w_{ij}(t+1) = w_{ij}(t) + \eta \cdot \delta_j \cdot x_j^i.$$

В этом уравнении w_{ij} - вес синапса для связи, идущей от входа i к нейрону j в момент времени t ; x_j^i - выход i -го нейрона предыдущего слоя, или, что то же – i вход j -го нейрона, η - значение шага адаптации, а δ_j - есть значение ошибки для j -го нейрона:

$$\delta_j = y_j \cdot (1 - y_j) \cdot (\alpha_j - y_j),$$

где y_j - эталон, т.е. желаемый выход j -го нейрона, y_j – действительный выход в момент t .
Если рассматривается j -й нейрон промежуточного слоя, то

$$\delta_j = x_j^i \cdot (1 - x_j^i) \cdot \sum_k \delta_w \cdot w_{ik},$$

где k – номер всех нейронов уровня, стоящего над j -м нейроном.

ШАГ 5. Возврат к шагу 2.

Данная итеративная процедура может продолжаться до достижения стабилизации выходов, наступающей при совпадении отклика сети на входное возбуждение с эталоном данного класса образов.

В заключение отметим, что метод обратного распространения предусматривает обучение НС с учителем, причем роль учителя состоит в предъявлении эталона запоминаемого класса образов.

Программа “NetMaker”

Данная программа предназначена для подготовки обучающей выборки.

Она предоставляет не только нейросетевые методы анализа и обработки данных, но и программы для статистической обработки данных. Это делает данный пакет мощным средством анализа информации.

Пакет BrainMaker 3.11 Pro, в который включена программа NetMaker, предоставляет возможность работать с такими популярными форматами, как Excel, Lotus, dBase, а также некоторыми другими текстовыми и двоичными форматами файлов.

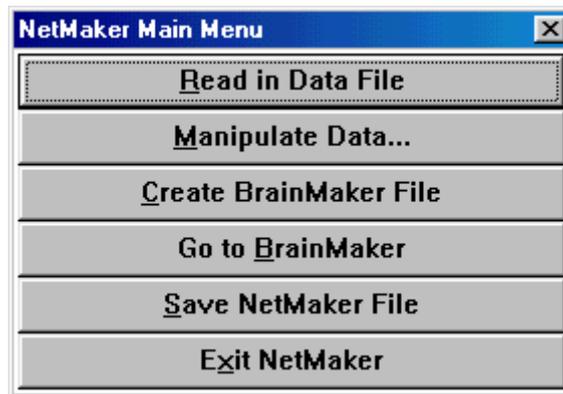
Данная программа обладает следующими развитыми средствами предварительной обработки и подготовки данных:

- статистические функции – помощь в предварительном анализе входных данных;
- операция хронологического сдвига – специальная подготовка данных для анализа изменения в данных во времени;
- арифметические операции с колонками данных;
- анализ диапазонов данных – помощь в поиске данных, выбивающихся из общей закономерности;
- циклический анализ – нахождение периодических закономерностей во входных данных;
- анализ корреляции данных – помощь в обнаружении взаимосвязей между входными переменными и наличия временного сдвига между ними.

По своим возможностям эта программа аналогична пакету Excel.

Программа NetMaker приводится в активное состояние путем запуска исполняемого файла **netmaker.exe**.

После запуска программы NetMaker пользователь видит главное меню:



Окно главного меню программы NetMaker

Функции, выполняемые при нажатии на соответствующие кнопки главного меню, следующие:

- Read in Data File
- Manipulate Data...
- Create BrainMaker File
- Go to BrainMaker
- Save NetMaker File
- Exit NetMaker
- Manipulate Data...

Программа “BrainMaker”

Данная программа нейрокета “BrainMaker 3.11 Pro” предназначена для обучения НС. При этом используется алгоритм обучения НС с обратным распространением ошибки (описанный выше). После обучения сети в BrainMaker имеется возможность эксплуатации такой сети в рабочем режиме.

В программе BrainMaker имеются мощные средства для выполнения анализа обученной НС:

- Анализ чувствительности – определение степени влияния отдельных входов НС на ее ответ.
- Контурный анализ – графическая визуализация зависимости влияния пар входных переменных на ответ НС.

Запустить программу BrainMaker можно несколькими способами:

1. Запуск исполняемого файла **brainmak.exe**.
2. Переход из программы NetMaker в данную программу, используя пункт главного меню NetMaker **Go to BrainMaker**.

После запуска BrainMaker и загрузки файла с обучающей выборкой или с параметрами обученной сети пользователь видит следующее окно:



Рис. Главное окно программы BrainMaker

В левом верхнем углу BrainMaker сообщает пользователю о том, какую операцию он сейчас выполняет:

- Waiting – ожидает действий от пользователя (закончил обучение),

- Training – обучение сети,
- Testing – тестирование сети.

В двух верхних текстовых строках отображены значения следующих параметров:

- Facts – имя файла обучающей выборки.
- Tolerance – значение ошибки при обучении.
- Fact – номер текущего факта.
- Total – общее количество обработанных фактов.
- Bad – текущее количество неправильно распознанных фактов.
- Last – количество неправильно распознанных фактов при последнем проходе в процессе обучения НС.
- Good – текущее количество правильно распознанных фактов.
- Last – количество правильно распознанных фактов при последнем проходе в процессе обучения НС.
- Run – общее количество проходов НС.

Операции, которые пользователь может использовать, представлены пунктами соответствующих меню.

Управление программой производится с помощью следующих пунктов меню.

- Меню Parameters
- Меню Connections
- Меню Display
- Меню Analyze
- Меню Operate
- Меню File
- Меню Edit

Меню Parameters

Позволяет устанавливать необходимые параметры обучения – допустимая вероятность ошибки, условия останова обучения и т.д, параметры шума, смазывания входной информации и симметрии входной и образцовой информации, использовать в нейронах разные функции возбуждения.

Позволяет получить информацию о типе и количестве входной информации и информации об образе для обучения.

Позволяет задать условие изменения размера сети (добавление нейронов и скрытых слоев НС), удалить необходимое количество нейронов во внутренних слоях.

Меню Connections

Позволяет изменить количество внутренних слоев НС, вносить определенные помехи в веса соединений между нейронами НС при обучении для улучшения качества работы НС в рабочем режиме.

Присваивает весам всех соединений случайное значение, т.е. возвращает НС в исходное необученное состояние.

Удаляет (обнуляет веса) соединения нейронов с весом меньше заданного.

Меню Display

Выводит на экран (или скрывает)

- отображение входных и выходных слоев НС,
- значение указания учителя для обучения НС.

- окно гистограммы распределения весов соединений нейронов НС.
- окно прогресса обучения сети (Нижний график отображает тенденцию изменения ошибки обучения. Верхний график отображает распределение по величине ошибки весов соединений).
- строку параметров данной сети,
- строку статистики обучения НС,

Обновляет окно программы,

Позволяет задать, в каком виде будут отображаться слои НС (**Input** – входной, **1..6** – внутренние, **Output** – выходной, **Pattern** – значение указания учителя для обучения, **Error** – величина ошибки обучения).

Меню Analyze

Позволяет проанализировать влияние изменения

- одного входа на выход сети,
- двух входов на выход сети,

Определяет чувствительность всех входов сети.

Анализирует изменения выхода при подаче на вход сети фактов.

Меню Operate

Позволяет производить

- обучение НС,
- тестирование НС с использованием тестовых фактов, определенных в файле обучающей выборки,
- запуск обученной НС в рабочем режиме.

Реализовать

- обучение сети в пошаговом режиме,
- пошаговый режим с остановкой на каждом факте,
- пошаговый режим с остановкой на каждом плохом факте (неправильно распознанном нейронной сетью),
- выбор по номеру факта для следующего шага обучения.
- режим очень быстрого обучения с потерей качества.

Меню File

Позволяет

- загрузить файл с подготовленной в NetMaker обучающей выборкой или с параметрами модели НС, обученной в BrainMaker,
- сохранить параметры (веса, опции обучения) модели НС, обученной в BrainMaker,
- задать условия автоматического сохранения параметров НС в процессе ее обучения,
- выбрать файл с входными векторами (фактами) обучающей выборки или фактами для их обработки обученной НС в рабочем режиме,
- редактировать с использованием мыши факты, представленные в его окне, что даёт возможность пользователю создавать или редактировать обучающую выборку средствами BrainMaker.
- выводить окно диалога с информацией о программе.

Определяет файлы для сохранения:

- статистик обучения,
- результатов каждого шага обучения или тестирования
- результатов работы обученной НС в рабочем режиме (в том числе – параметры слоёв НС).
- информации об изменении выхода при изменении одного входа в заданном интервале

Меню Edit

Позволяет:

- редактировать
 - веса нейронов в обученной или обучаемой сети,
 - текстовую информацию, используемую для формирования фона окна программы.
 - входные значения и указания учителя, используя клавиатуру – клавиши управления курсором, пробела, enter и цифровые клавиши,
 - копировать в буфер обмена информацию о входе и выходе,
 - изменять:
 - интервал, в котором может изменяться величина веса необходимого нейрона входного или выходного слоя,
 - значения элементов входного словаря или словаря указаний учителя для обучения,
 - вставлять информацию из буфера обмена,
 - создавать DDE - соединение с другим приложением для использования в нем информации, полученной в данной программе.

Обнуляет информацию на входе сети, что удобно использовать при формировании новых фактов.

Лекция 3. Концепция нейросетевого исследования.

Архитектура нейросетевых исследований.

Что собой представляет нейросетевое исследование?

От ответа на этот вопрос зависит очень многое. Начиная от того, какие задачи решаются с помощью нейронных сетей, как для такого исследования готовится информация, как должна выглядеть нейронная сеть, и вплоть до того, как должен выглядеть нейропакет – программа, с помощью которой проводится нейросетевое исследование, какие возможности предоставляет эта программа исследователю.

Неправильное представление об архитектуре нейросетевого исследования приводит к тому, что в нейропакет включаются несвойственные ему функции, чаще всего – заимствованные из дискретных методов математической статистики, сам пакет обычно содержит хорошо продуманный блок обучения нейросети и не имеет средств для решения задачи после окончания обучения, необходимых для эксплуатации обученной сети в рабочем режиме; не говоря о том, что нейропакет должен содержать средства для анализа эффективности обучения, для исследования качества получаемого решения, для оценки степени влияния различных исходных данных на результат решения.

Рассмотрим основные элементы архитектуры нейросетевых исследований на примере нейропакета Brain Maker фирмы California Scientific Software [2]. Этот пакет ориентирован на проведение нейросетевых исследований с помощью многослойного перцептрона.

Нейросетевое исследование предусматривает выполнение следующих этапов:

1. Подготовка исходных данных.
2. Разметка исходных данных.
3. Формирование нейронной сети.
4. Предварительное обучение сети.
5. Анализ результатов обучения.
6. Оптимизация обучения сети.
7. Анализ подготовленных для исследования данных.
8. Проведение нейросетевого исследования.

Подготовка данных для исследования с помощью нейронной сети.

Данные в нейронной сети используются прежде всего – для обучения. С этой целью они оформляются в виде обучающей выборки, или обучающего набора данных.

Например, для определения пола посетителя Web-сайта можно использовать нейронную сеть, которую необходимо обучить классификации на два класса по косвенным признакам. Обучающий набор данных в этом случае может представлять собой таблицу вида:

Информационные (inf)	Исходные показатели (input)				Результирующий (pattern)
	№ п/п	Готовите ли вы дома пищу	Как часто вы убираете квартиру	Сколько времени в неделю вы тратите на ремонт автомобиля	
1	нет	редко	3 часа	да	м
2	Да	всегда	0	нет	ж
...

Таблица содержит 3 вида данных: информационные (inf) – например, «№ п/п»; исходные данные (input); результирующий показатель (pattern).

Когда сеть обучена, для определения пола посетителя ему нужно задать только информационные вопросы, по ответам на которые сеть принимает решение, кто зашёл на сайт.

Разметка исходных данных.

При разметке исходных данных создаются тренировочный, тестовый и экзаменационный наборы данных. Тренировочный набор данных содержит обучающую выборку, он используется для обучения нейронной сети.

Тестирование необходимо для проверки создания обобщённых образов, которые были представлены в обучающей выборке данных.

Для адекватной оценки качества полученной сети используется ещё один совершенно независимый набор данных, называемый обычно экзаменационным. В процессе обучения сети, определения момента остановки обучения, определения дополнительных параметров и др. этот набор вообще никак не используется, т.е. при последующем применении сеть "видит" его впервые. Поэтому качество результатов, показанных сетью на экзаменационном наборе, является хорошей оценкой качества результатов, которые сеть сможет показать на новых, неизвестных ей данных. Разумеется, экзаменационный набор также должен быть представительным.

Программы пакета, предназначенные для решения задач классификации, позволяют задавать все три набора данных, как по отдельности, так и путём установления

доли каждого из наборов в общем количестве данных. При применении сети статистика рассчитывается отдельно для каждого набора.

Формирование нейронной сети.

Формирование нейронной сети предусматривает: определение типа сети, количества слоёв, количества нейронов во входном, выходном и промежуточных слоях, и др.

Установка параметров нейронной сети.

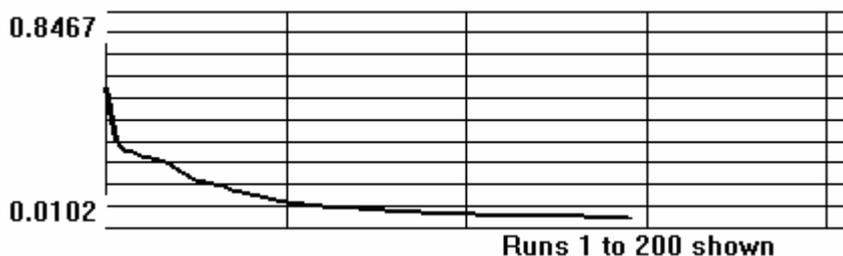
Состав параметров определяется в зависимости от используемого вида нейронной сети. Так, для многослойного перцептрона могут быть определены:

- допустимая погрешность обучения (training tolerance);
- допустимая погрешность тестирования (testing tolerance);
- условия остановки обучения:
 1. после окончания N эпохи;
 2. при отсутствии ошибок во время выполнения всей тестирующей выборки;
 3. при достижении M% правильных результатов обучающей выборки;
 4. при уменьшении ошибки сети ниже K%;
- тестирование сети после прохождения каждых k эпох; или каждых p обучающих примеров;
- сохранение результатов после завершения каждых r эпох;
- и т.д.

При установке параметров нейронной сети создаётся и сохраняется в файле начальная матрица коэффициентов связи перцепторов с сумматором.

Предварительное обучение, анализ результатов и оптимизация обучения сети

В процессе обучения сети ей на вход подаётся один из примеров обучающей выборки и фиксируется результат (output) на выходе сети. Поскольку сеть ещё не обучена, этот результат отличается от желаемого (target). Величина отличия называется ошибкой сети. Сначала она имеет большую величину, по мере обучения – уменьшается:



Общая ошибка нейросети вычисляется как RMS Error (Root Mean Squared) по следующей формуле:

$$RMSError = \frac{\sqrt{\sum ((O - P)^2)}}{N}$$

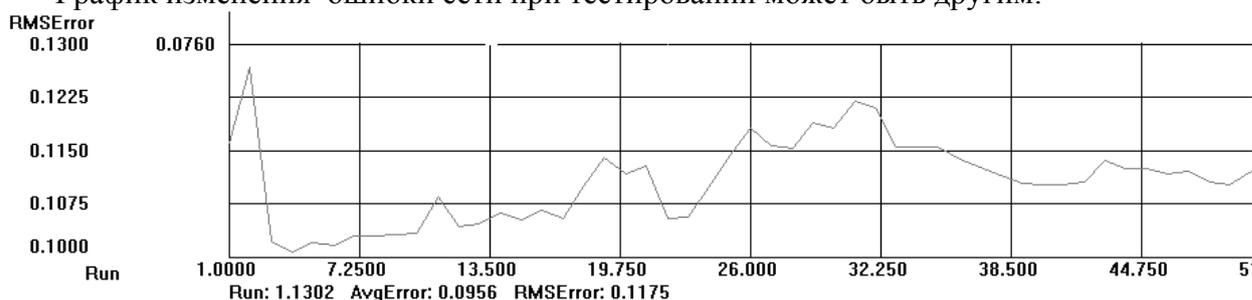
где O output, P pattern, N количество фактов. [5]

Процесс обучения сети заключается в изменении весов связей перцепторов с сумматором таким образом, чтобы ошибка сети уменьшилась.

Когда весь запас обучающих примеров исчерпан, считается, что закончилась одна эпоха. Подсчитывается средняя ошибка за эпоху и примеры из обучающей выборки снова используются для обучения той же слегка обученной сети – это вторая эпоха.

На приведенном выше рисунке приведен график, отражающий результаты 200 эпох.

После каждых k эпох обучения фиксируются результаты, и вместо обучающей выборки на вход сети подаются примеры из тестирующей выборки. По ним так же определяется ошибка сети, но обучение сети не проводится. Когда примеры из тестирующей выборки исчерпаны, определяется средняя ошибка тестирования. График изменения ошибки сети при тестировании может быть другим:



Он демонстрирует явление, которое называется «переучиванием»: т.е. достижение такого состояния сети, в котором сеть со 100% точностью распознаёт все примеры тренировочного набора данных, однако на других данных, не входивших в этот набор, может давать большую погрешность. Можно сказать, что такая сеть плохо обобщает. Из графика видно, что где-то в районе 4 эпохи есть точка, после которой ошибка сети резко возрастает. Это значит, что после эпохи, в которой получена эта точка, продолжать обучение по той же обучающей выборке бесполезно. Прекращение тренировки при минимуме ошибки на тестовом наборе позволяет зафиксировать сеть в том состоянии, когда она уже усвоила наиболее существенную информацию, содержащуюся в данных, и неплохо обобщает, однако ещё не успела переучиться.

Можно уточнить полученную информацию по результатам, сохранённым в файле при обучении. Читаем значения Run и RMSError и заносим их в таблицу:

Run	RMSError
3,2135	0,1019
3,9947	0,1008
4,1250	0,1009
4,9062	0,1019
6,0781	0,1018
7,0546	0,1030
8,0312	0,1031
9,0072	0,1033
9,9192	0,1034
10,049	0,1037
11,026	0,1083

Судя по графику RMSError, улучшение работы сети продолжается до 4 эпохи. В это время квадрат средней ошибки минимален. Возрастание её в дальнейшем связано с «переучиванием» сети. Желательно обучение сети на 4 эпохе закончить.

В некоторых источниках для определения момента переучивания используется средняя ошибка (AvgError).

Оптимизация обучения сети.

Оптимизация обучения сети заключается в том, что полученные результаты обучения стираются, восстанавливается сохраненная в файле начальная матрица коэффициентов связи перцепторов с сумматором, устанавливается ограничение условия остановки обучения: после окончания 4 эпохи, и снова проводится обучение нейронной сети. Заново обученную сеть можно считать оптимальной, она сохраняется в файле и в дальнейшем может быть использована для решения задач данного типа в производственном режиме.

Подготовка данных для исследования на обученной сети.

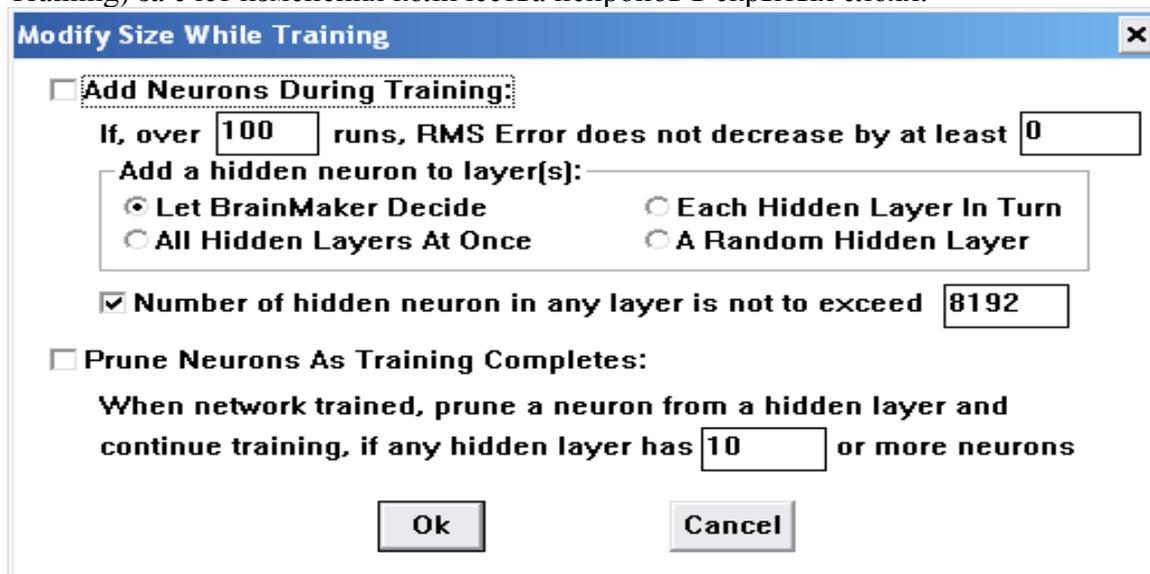
Исследование лучше проводить на заранее подготовленном файле, содержащем исходные данные. Такой файл не должен содержать колонку pattern, результат будет отражаться в колонке out.

Проведение нейросетевого исследования.

В составе нейросетевых пакетов можно использовать имеющиеся средства для наблюдения за работой (например, за процессом обучения).

а. В пакете Brain Maker для наблюдения за ошибкой распознавания можно использовать пункты меню Display и Parameters. Пункт “Network Progress Display” в меню “Display” в открывшемся окне позволяет наблюдать как уменьшается значение ошибки распознавания в процессе обучения.

б. Можно включить модификацию сети во время обучения (Modify Size While Training) за счёт изменения количества нейронов в скрытых слоях:



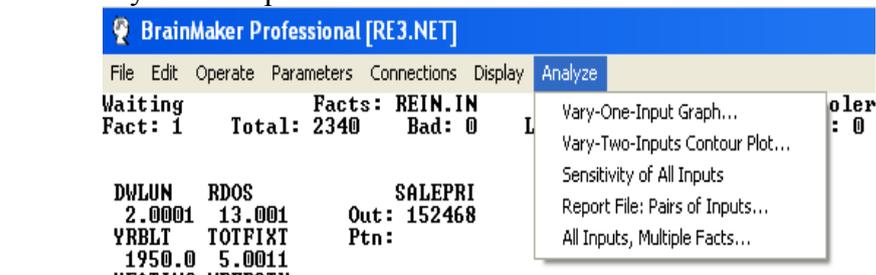
в. Есть ещё одно средство наблюдения - пункт “Show Histograms” в меню “Display” покажет, как нейронная сеть меняет свои внутренние свойства, в частности, теряет, по мере накопления знаний, *способность* обучаться.

Если после обучения нейросеть эффективно обрабатывает данные обучающего множества, важным становится исследование эффективности работы с данными, которые не использовались для обучения.

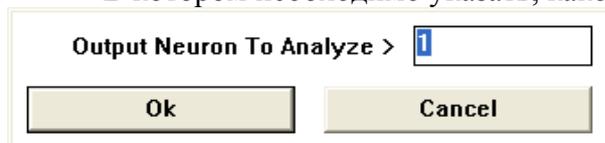
В блок анализа нейропакетов обычно входит функция оценки чувствительности для одного входа. Она демонстрирует, насколько данный вход оказывает влияние на

выход. Предполагается, что с помощью этой функции можно исследовать поведение нейронной сети с целью выявления мало чувствительных входов или поддиапазонов внутри множества принимаемых входов значений для их последующего исключения в момент проектирования нейронной сети.

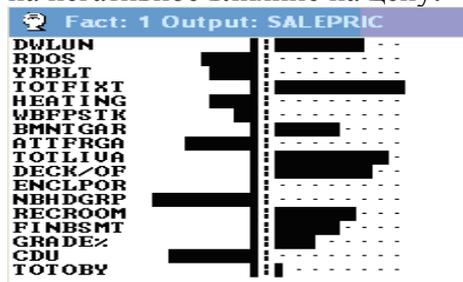
Проверка чувствительности всех входов производится через меню Analyze → Sensitivity of All Inputs.



В котором необходимо указать, какой именно нейрон будет исследоваться:



В пакете BrainMaker указывается чувствительность входов с точностью +/-10%. Результат виден на графике (термометр) через меню Analyzing. Термометр влево укажет на негативное влияние на цену.



Таким образом, нейросетевое исследование по меньшей мере характеризуется следующими факторами, определяющими минимальный состав функций, необходимый для проведения нейросетевого исследования данных:

1. Какая задача должна быть решена:
 - узнавание
 - кластеризация
 - прогнозирование
 - сокращение размерности
 - ...
2. Какая сеть будет использована для решения задачи?
 - простая
 - однослойная
 - многослойная (с каким количеством слоёв и нейронов в них)
 - с какой пороговой функцией нейронов в каждом слое
 - составная
3. Технология обучения
 - обычная для многослойного перцептрона (формируется матрица межслойных связей перцептронов с сумматором)
 - изменением количества или типов нейронов в определённых слоях
4. Дополнительная настройка сети:
 - как определить момент окончания обучения
 - способ получения оптимально обученной сети

- какие наборы данных необходимы для нейросетевого исследования: обучающий, контролирующий, экзаменационный

5. Критерии работы нейропакетов

- tolerance (терпимость сети к ошибкам)
- количество эпох
- допустимое количество (bad) ошибок при обучении или тестировании
- % правильного распознавания
- надо ли производить запись промежуточных результатов, и с какой частотой.

Пример нейросетевого исследования: Прогнозирование стоимости акций в пакете BrainMaker.

Постановка задачи.

Цель исследования: Прогнозирование изменения цен на акции некоторой компании Bart-Davis-100 (BD100).

Исходный файл Price1.dat находится в демонстрационных файлах пакета BrainMaker и представляет собой текстовую таблицу (185 строк), в которой находятся упорядоченные по строкам и столбцам технические индикаторы, индексы и цены:

day	BD100	price1	price2	index1D	index2D	line	streng	utilD	transpD
1	2877	38.77	46.77	14.68	-5.4	114.37	97.62	26.22	3.23
2	2789	39.07	46.88	20.69	-3.55	126.19	91.92	26.84	2.6
3	2622	39.47	47.11	23.66	0.15	140.34	99.76	28.64	2.92

Каждой строке соответствуют данные одного дня. В столбцах, обозначенных day, BD100, price1, price2, index1D, index2D, line, streng, utilD, transpD, содержатся данные, характеризующие с разных сторон стоимость акций.

Первый столбец (day) порядковый номер дня, выполняет чисто информативные функции.

Столбец BD100 содержит значения, которые одновременно являются как входными данными, так и результатом работы нейросети, прогнозом.

Остальные величины могут быть ценами на сырьевые товары (нефть, золото) на мировом рынке, значениями фондовых индексов, отражающих движение цен на биржах (Dow-Jones, Nikkei и др.), индикаторами технического анализа и т.д.

Предварительный анализ этих данных показал, что более точную экономическую картину можно получить, преобразовав их и получив несколько изменённый состав данных. После препроцессинга [5] образовался файл данных – Price4.dat – который изменился и стал содержать несколько новых показателей. Часть старых показателей потеряла своё значение (они помечены, как Not Used). Если исключить неиспользуемые показатели, получается, что для анализа подготовлен файл Price4z.dat, имеющий следующую структуру:

№	Имя переменной	Свойство	Содержание	Примечание
1	day	Annote	порядковый номер дня	
2	index1D	Input	индикатор технического анализа	См. [5]
3	index2D	Input	индикатор технического анализа	См. [5]
4	line	Input	индикатор технического анализа	См. [5]
5	streng	Input	индикатор технического анализа	См. [5]
6	utilD	Input	индикатор технического анализа	См. [5]
7	transpD	Input	индикатор технического анализа	См. [5]

8	BD100D	Input	Изменение стоимости акции	
9	price1D	Input	Изменение цены накануне	
10	price2D	Input	Изменение цены 2 дня назад	
11	BDAvg2	Input	Средняя стоимость акции за 2 дня	
12	ind1-1	Input	вчерашние значения index1D	
13	BD+4	Pattern	Прогноз стоимости акций через 4 дня	

Требуется: по приведенным в таблице Price4z1.dat данным (после исключения лишних данных) обучить сеть прогнозированию стоимости акций.

Подготовка сети.

1. Вводим файл с исходными данными: NetMaker -> File -> Read In Data File -> Выбор Price4z1.dat -> Manipulate Data:

day	index1D	index2D	line	streng	utilD	transpD	BD100D	price1D	price2D	BDAvg2	ind1-1	BD+4
1	-12.89	9.61	44.94	92.45	3.38	0.1	147	0.38	0.08	139	28.02	62
2	-17.9	5.91	47.67	89	4.62	-0.31	-125	-0.11	0.07	11	-12.89	120
3	-18.01	-1.93	54.76	72.32	5.21	-0.21	-118	-0.06	0.06	-121.5	-17.9	-101
4	-11.84	0.15	63.16	77.6	3.7	-1.78	55	-0.03	0.15	-31.5	-18.01	-96
5	-10.88	0.15	57.1	61.49	-1.81	-2.3	62	-0.03	0.14	58.5	-11.84	72
6	-4.08	-2.07	61.39	54.07	-1.22	-1.77	120	-0.01	0.14	91	-10.88	68
7	-2.06	-4.29	68.96	51.03	0.29	-1.57	-101	-0.02	0.31	9.5	-4.08	115
8	-0.84	-6.51	74.01	55.63	3.38	-0.63	-96	0.06	0.4	-98.5	-2.06	-119
9	0.63	-6.51	83.32	66.24	13.87	-0.42	72	0.08	0.37	-12	-0.84	-121
10	-0.07	-6.51	71.37	56.4	6.71	-0.21	68	0.05	0.18	70	0.63	41
11	-1.22	-4.29	57.02	49.12	2.73	0.63	115	0.01	0.07	91.5	-0.07	38
12	-3.59	-4.45	45.91	39.49	-6.11	1.05	-119	0.05	-0.05	-2	-1.22	92
13	-3.85	-6.99	45.47	41.9	-5.8	0.84	-121	0.05	0.17	-120	-3.59	-134
14	-11.5	-11.75	23.07	13.18	-13.97	1.78	41	-0.05	0.09	-40	-3.85	-118

В таблице содержится 175 строк.

По умолчанию 10% всех данных будут использоваться для контроля (testing), в связи с чем в обучающую выборку (training) попадет 157 фактов. Поскольку 175 примеров – это выборка маленькая, для контроля можно взять и 5% данных.

2. Начинаем разметку данных для нейронной сети.

Колонку BD+4 помечаем как выходную - pattern или output обучающей выборки. Колонку day помечаем, как Annote. Все остальные колонки помечаем, как inputs.

3. Сохраняем размеченный файл под именем Price4z2.dat.

File	Column	Row	Label	Number	Symbol	Operate	Indicators								
	Annote	Input	Input	Input	Input	Input	Input	Input	Input	Input	Input	Input	Input	Pattern	
	day	index1D	index2D	line	streng	utilD	transpD	BD100D	price1D	price2D	BDAvg2	ind1-1	BD+4		
1	6	-12.89	9.61	44.94	92.45	3.38	0.1	147	0.38	0.08	139	28.02	62		
2	7	-17.9	5.91	47.67	89	4.62	-0.31	-125	-0.11	0.07	11	-12.89	120		
3	8	-18.01	-1.93	54.76	72.32	5.21	-0.21	-118	-0.06	0.06	-121.5	-17.9	-101		
4	9	-11.84	0.15	63.16	77.6	3.7	-1.78	55	-0.03	0.15	-31.5	-18.01	-96		
5	10	-10.88	0.15	57.1	61.49	-1.81	-2.3	62	-0.03	0.14	58.5	-11.84	72		
6	11	-4.08	-2.07	61.39	54.07	-1.22	-1.77	120	-0.01	0.14	91	-10.88	68		
7	12	-2.06	-4.29	68.96	51.03	0.29	-1.57	-101	-0.02	0.31	9.5	-4.08	115		
8	13	-0.84	-6.51	74.01	55.63	3.38	-0.63	-96	0.06	0.4	-98.5	-2.06	-119		
9	14	0.63	-6.51	83.32	66.24	13.87	-0.42	72	0.08	0.37	-12	-0.84	-121		
10	15	-0.07	-6.51	71.37	56.4	6.71	-0.21	68	0.05	0.18	70	0.63	41		
11	16	-1.22	-4.29	57.02	49.12	2.73	0.63	115	0.01	0.07	91.5	-0.07	38		
12	17	-3.59	-4.45	45.91	39.49	-6.11	1.05	-119	0.05	-0.05	-2	-1.22	92		
13	18	-3.85	-6.99	45.47	41.9	-5.8	0.84	-121	0.05	0.17	-120	-3.59	-134		
14	19	-11.5	-11.75	23.07	13.18	-13.97	1.78	41	-0.05	0.09	-40	-3.85	-118		

В VM будут использоваться такие файлы:

Обучающий: training.fct <- training.dat
 Тестирующий testing.tst <- testing.dat
 Исполнимый (running) rein.dat
 Результирующий re.out

4. Создаём файлы для BrainMaker. В NetMaker набираем: File -> Create_BrainMaker_Files



При этом создаются файлы описания сети (Price4z2.def), обучающая выборка (Price4z2.fct) и файл для тестирования сети (Price4z2.tst).

5. Покидаем NetMaker и переходим в BrainMaker: File -> Exit. Запускаем BrainMaker.



6. Через меню File читаем файл с описанием сети (Price4z2.def):
 File -> Read Network:

При загрузке сети изменяется внешний вид окна программы:

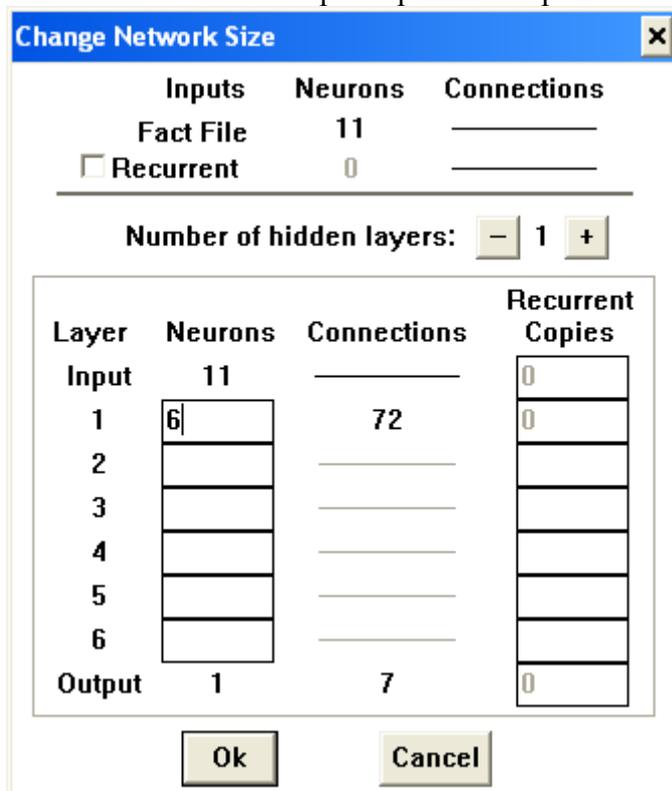


7. Начинаем конструировать сеть.

В данном приложении имеется настолько малое количество фактов, что нужно особое внимание обратить на то, чтобы не создавалось слишком большое количество скрытых нейронов.

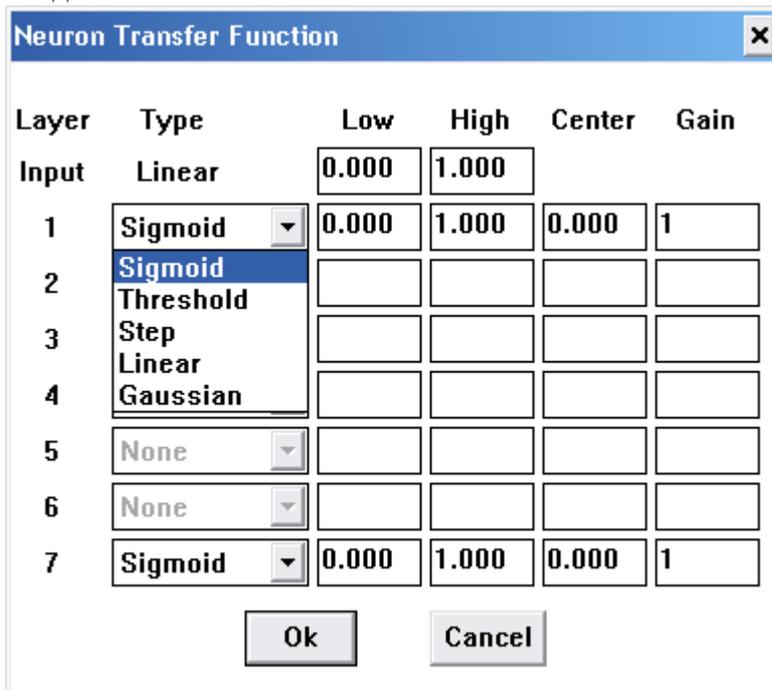
Обычно количество скрытых нейронов выбирается равным 1 слою, в редких случаях 2. BrainMaker допускает создание кроме входного и выходного ещё 6 слоёв. В данном примере во входном слое необходимо 11 нейронов (по одному нейрону на каждый входной параметр). В выходном слое необходим 1 нейрон (так как в таблице исходных данных указан 1 выходной параметр). При 1 промежуточном слое в нём может содержаться либо столько же нейронов, как и во входном слое, либо среднее количество от входного и выходного слоёв $(11+1)/2=6$. Кроме того, в BrainMaker разрешается изменять количество нейронов в скрытом слое во время обучения.

Устанавливаем размеры сети через меню Connections -> Change Network Size:



Итак, входных нейронов 11, один промежуточный (скрытый) слой с 6 нейронами (по умолчанию BrainMaker хотел в скрытом слое установить 11 нейронов) и один выходной слой с 1 нейроном.

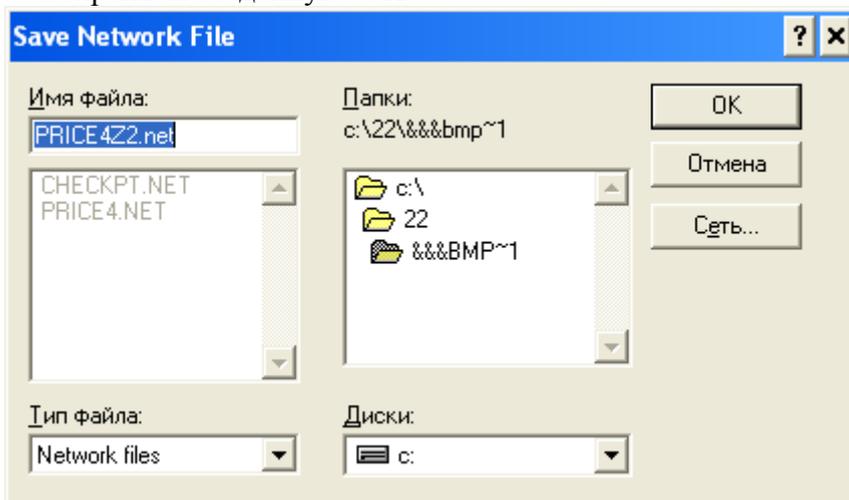
8. При необходимости можно установить типы пороговых функций в каждом слое, кроме входного:



На этом рисунке видны допустимые пороговые функции, открытые для промежуточного слоя: Sigmoid, Treshold, Step, Linear, Gayssian.

В данном примере оставляем выбранные по умолчанию пороговые функции.

9. Сохраняем созданную сеть:



10. Обучаем сеть. Нажимаем Operate -> Get next fact:

```

BrainMaker Professional [RE2.NET]
File Edit Operate Parameters Connections Display Analyze
Waiting Fact: 1 Total: 0 Facts: RE2.fct Bad: 0 Last: 0 Learn: 1.000 Tolerance: 0.200
Good: 0 Last: 0 Run: 0
DVLUN RDOS SALEPRI
1.0001 13.001 Out: 236734
YRBLI TOTFIXT Ptn: 140014
1920.0 5.0011
HEATING WBPFSIX
2.0000 0.0000
BPMIGAR HITPRGA
0.0001 0.0208
TOILIVA DECK/OF
1440.5 24.030
ENCLPOR NBWDGRP
0.0413 1.0000
RECROOM FINBSMI
0.0615 0.0741
GRADE: CDU
1.0000 4.0001
TOTOBV
2792.2

```

На экране появляются результаты **первого шага** обучения. Результат работы сети: out=236734, тогда, как сети предлагалось получить Ptn=140014.

Факты проскакивают через сеть быстро – за одну эпоху сеть обучает 157 фактов, после чего 16 фактов (т.е. 10%) предлагаются сети для тестирования.

11. После обучения на экране появляются результаты:

```

BrainMaker Professional [PRICE4Z2.NET]
File Edit Operate Parameters Connections Display Analyze
Waiting 00:04:12 Facts: PRICE4Z2.fct Learn: 1.000 Tolerance: 0.100
Fact: 158 Total: 419490 Bad: 0 Last: 3 Good: 158 Last: 155 Run: 2655
index1D index2D BD+4
line streng Out:
utilD transpD Ptn:
BD100D price1D
price2D BDaug2
ind1-1

```

Здесь уже out и pattern имеют более близкие значения.

В первой строке экрана появляется слово waiting, во второй - Bad=0, Good=158.

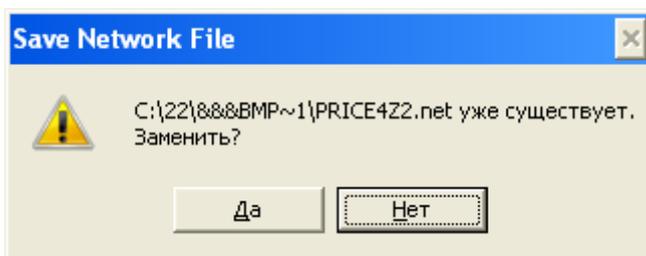
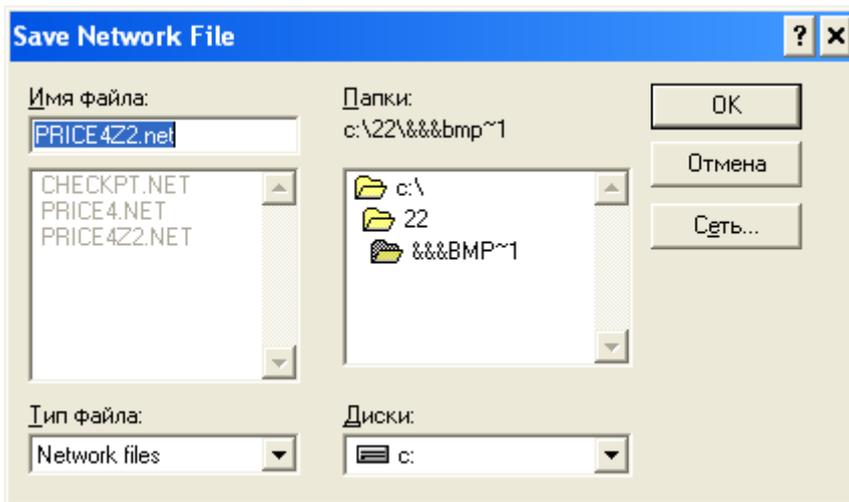
12. Очистка экрана. Display -> Enable Display.

```

BrainMaker Professional [PRICE4Z2.NET]
File Edit Operate Parameters Connections Display Analyze
Waiting Facts: PRICE4Z2.fct Learn: 1.000 Tolerance: 0.100
Fact: 158 Total: 419490 Bad: 0 Last: 3 Good: 158 Last: 155 Run: 2655

```

13. Сохранение сети: File -> Save Network File



Заменяем.

Выходим из BrainMaker: File -> Exit.

Подготовка файла для запуска.

14. Создаём файл PriceIN.dat, содержащий данные для нейросетевого анализа: Для проверки качества прогноза предложим сети для прогнозирования данные о стоимости акций для последних двух строк обучающей выборки (по ним известно, сколько они стоили).

15. Дополняем файл REIN.dat разметкой и сохраняем как Running Fact File –PriceIN.in

The screenshot shows the NetMaker Professional interface with a table titled 'PRICES.TXT'. The table has columns for 'Annote', 'Input', and 'Pattern'. The data is as follows:

	Annote	Input	Input	Input	Input	Input	Input	Input	Input	Input	Input	Input	Input	Pattern
	A	B	C	D	E	F	G	H	I	J	K	L	M	
1	182	6.44	-19.61	63	76.25	1.18	0.83	67	-0.01	0.03	65	6.62	0	
2	183	3.23	-13.84	60.42	56.5	3.46	0.72	122	-0.11	-0.08	94.5	6.44	0	

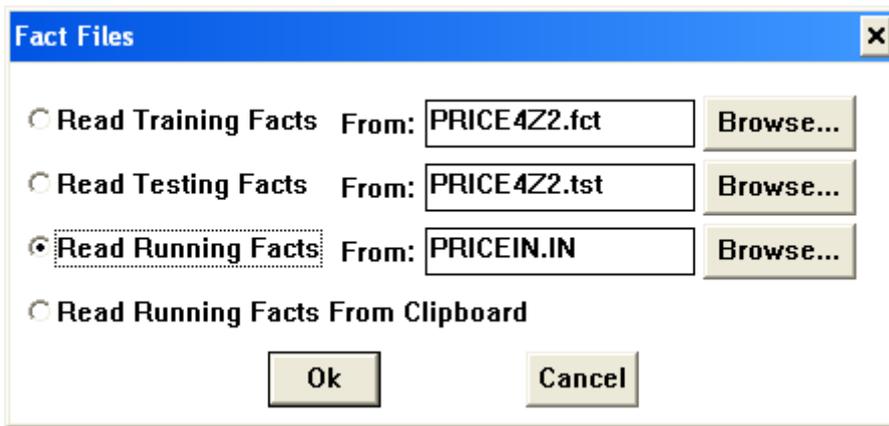
16. Покидаем NetMaker: File -> Exit.

Прогнозирование стоимости акций.

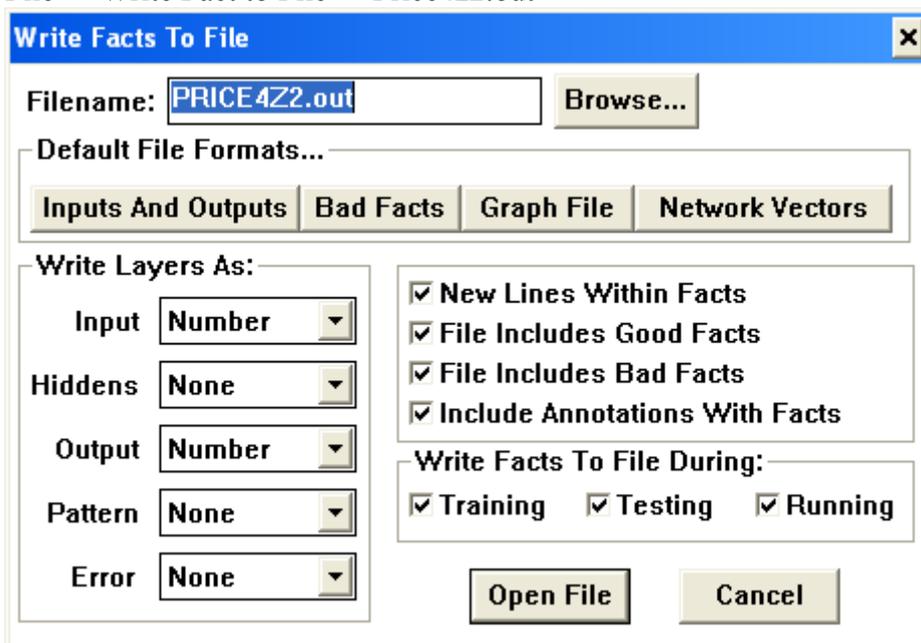
Запускаем BrainMaker.

Читаем сеть из PRICE4Z2.NET.

Читаем Pricein.in (то же, что и Pricein.dat, с указанием, какие данные являются входными).



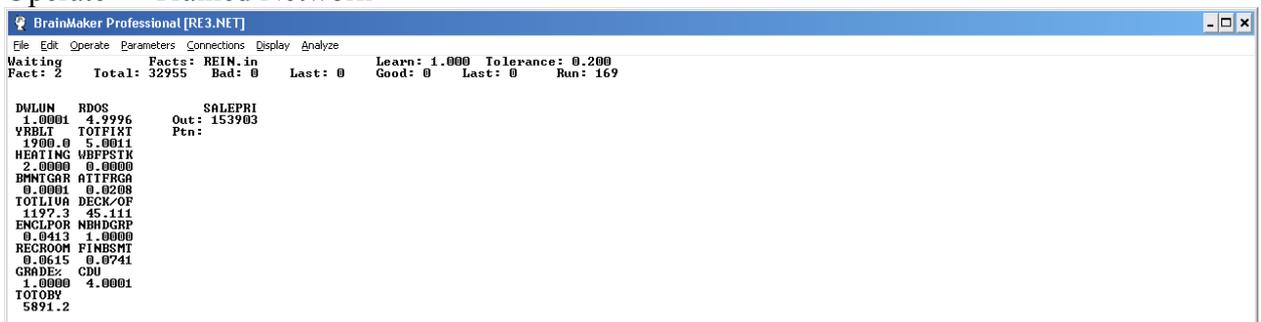
Создаём результирующий файл:
File -> Write Fact to File -> Price4z2.out



Устанавливаем формат файла по умолчанию – Inputs Fyl Outputs.

Запускаем сеть для анализа:

Operate -> Trained Network



Результат анализа читаем в Price4z2.out.

Вначале (до исследования) в файле RE.dat были данные, что цена акций была: 86\$ и 60\$

Прогноз полученный в Price4z2.out:

99.614\$

77.830\$

Этот прогноз содержится в строках 179:

6.6306 -23.175 52.033 72.512 -7.1054 0.3113 63.016 0.0003 -0.0499 -27.475 4.6859
-99.614
и в 180:
6.4403 -19.605 63.028 76.254 1.1823 0.8308 67.071 -0.0096 0.0300 65.039 6.6264
77.830

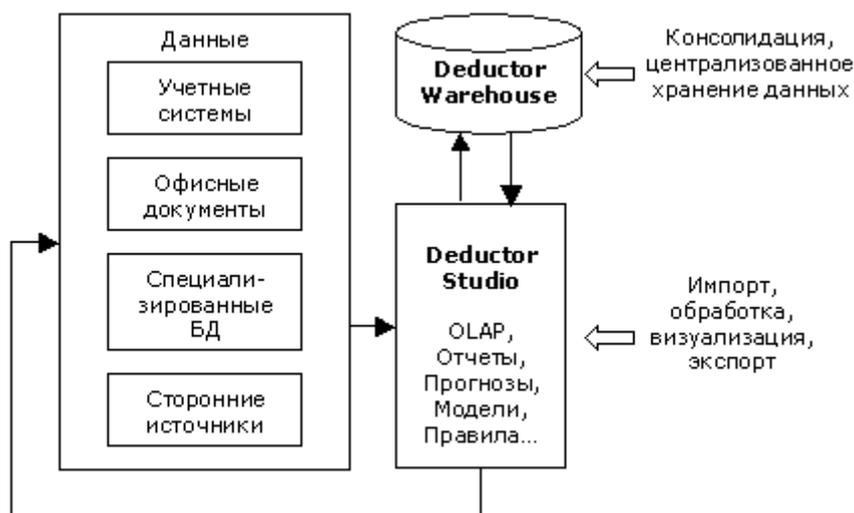
Лекция 4. Deductor - флагманский продукт компании BaseGroup Labs.

О системе

Deductor является аналитической платформой - основой для создания законченных прикладных решений в области анализа данных. Реализованные в Deductor технологии позволяют на базе единой архитектуры пройти все этапы построения аналитической системы от создания хранилища данных до автоматического подбора моделей и визуализации полученных результатов.

Deductor состоит из таких частей:

- [Deductor Warehouse](#) - многомерное хранилище данных, аккумулирующее всю необходимую для анализа предметной области информацию.
- [Deductor Studio](#) - программа, реализующая функции импорта, обработки, визуализации и экспорта данных.
- [Deductor Viewer](#) - программа, ориентированная на конечного пользователя и предназначенная для просмотра подготовленных при помощи Deductor Studio отчетов.



Применение системы

Система Deductor предназначена для решения широкого спектра задач, связанных с обработкой структурированных и представленных в виде таблиц данных. При этом область приложения системы может быть практически любой - механизмы, реализованные в системе, с успехом применяются на финансовых рынках, в страховании, торговле, телекоммуникациях, промышленности, медицине, в логистических и маркетинговых задачах и множестве других.

Кратко рассмотрим наиболее распространенные классы задач, на решение которых рассчитан Deductor:

Системы отчетности

В Deductor включено многомерное хранилище данных Deductor Warehouse, позволяющее быстро создавать системы корпоративной отчетности - финансовой, аналитической, маркетинговой и т.д.

Благодаря наличию большого набора визуализаторов хранящиеся в Deductor Warehouse данные можно просматривать различными способами: в виде OLAP кубов, таблиц, диаграмм, гистограмм и т.д. Дополнительную ценность системе придает возможность проведения обработки и анализа не "сырых" данных, а соответствующим образом подготовленных - сглаженных, очищенных от шумов, пропусков и аномальных значений.

Data Mining проекты

Data Mining - это процесс извлечения из исходных данных ранее неизвестных, нетривиальных и практически ценных зависимостей (знаний). Среди множества задач, решаемых при помощи методов Data Mining, можно выделить такие:

- Анализ и управление рисками;
- Оценка кредитоспособности физических и юридических лиц;
- Определение профилей клиентов, т. е. наиболее характерных особенностей их поведения;
- Промышленная диагностика, обнаружение источников и причин возникновения дефектов;
- Идентификация критических ситуаций;
- Стимулирование продаж;
- Сегментация клиентов, продуктов, услуг;
- Оценка факторов, влияющих на лояльность клиентов
- и множество других...

Очистка данных

Очистка данных даже без последующей обработки сама по себе представляет значительный интерес. На практике исходные данные чаще всего бывают плохого качества. Это связано с объективными причинами, и избавиться от этого, применяя какие-либо административные или организационные методы, невозможно. Поэтому необходимо использовать методы очистки данных. Очищенные данные содержат наиболее ценную для анализа информацию, из которой исключены противоречивые и дублирующиеся данные, устранены аномальные выбросы и шумы.

Прогнозирование

Прогнозирование - одна из наиболее востребованных задач анализа. В Deductor включено несколько механизмов построения прогностических моделей, в том числе с использованием самообучающихся алгоритмов.

Кроме того, имеются механизмы, позволяющие автоматически строить прогнозы на основе временного ряда на несколько шагов вперед. Достаточно построить модель, прогнозирующую изменение на 1 шаг, и автоматически использовать ее на произвольное количество отсчетов вперед.

Моделирование

Построение моделей - наиболее универсальный способ анализа. В действительности при анализе мы в том или ином виде всегда имеем некую модель исследуемого процесса или объекта. Но не всегда эта модель формализована, т.е. описана таким образом, чтобы ей мог воспользоваться кто-то в режиме "черного ящика": подали данные на вход, а на выходе получили результат.

Способов построения моделей существует множество, но в Deductor основной акцент сделан на самообучающиеся методы и машинное обучение. Данные алгоритмы являются универсальными, решающими большой спектр задач и при этом просты в применении.

Анализ "что-если"

При помощи Deductor можно не только строить модели, но и провести анализ по принципу "что-если", т.е. оценить, как может измениться тот или иной показатель при изменении любого влияющего фактора. Для реализации этого простого в использовании и одновременно мощного механизма предназначен специальный визуализатор.

Deductor Studio/Viewer - аналитическое приложение

Deductor Studio - аналитическое ядро платформы Deductor. Deductor Studio содержит полный набор механизмов импорта, обработки, визуализации и экспорта данных для быстрого и эффективного анализа информации.

В Deductor Studio включен полный набор механизмов, позволяющий получить информацию из произвольного источника данных, провести весь цикл обработки (очистку, трансформацию данных, построение моделей), отобразить полученные результаты наиболее удобным образом (OLAP, таблицы, диаграммы, деревья решений...) и экспортировать результаты.

Вся работа по анализу данных в Deductor Studio базируется на выполнении следующих действий: Импорт – Обработка – Визуализация – Экспорт. Управление этими действиями ведут соответствующие мастера.

Мастер импорта

Мастер импорта поможет в интерактивном пошаговом режиме выбрать тип источника данных и настроить соответствующие параметры.

Доступными являются подключения:

- **Хранилища данных:**

[Virtual Warehouse](#) - импорт данных из Virtual Warehouse;
Deductor Warehouse - импорт данных из Deductor Warehouse.

- **Базы данных:**

База данных - импорт данных из баз данных различных типов.

- **Прямой доступ к файлам:**

Текстовый файл с разделителями - текстовый файл в формате, в котором столбцы данных разделены однотипными символами-разделителями;
Импорт из DBF - прямой доступ к файлам плоских баз данных типа DBF, который поддерживается такими приложениями, как dBase, FoxBase, FoxPro.

- **Механизм MS ADO:**

Microsoft Excel - книга Microsoft Excel (*.xls);
Microsoft Access - файл СУБД Microsoft Access (*.mdb);
Импорт из DBF (ADO) - доступ через ADO к файлам плоских баз данных типа DBF, который поддерживается такими приложениями, как dBase, FoxBase, FoxPro;
Текстовый файл с доступом через ADO - текстовый файл с разделителями, доступ к которому производится через механизм ADO.

В Deductor может быть добавлено Бизнес-приложение:

- **Бизнес-приложение:**

1С:Предприятие 7.7 - импорт данных из учетной системы 1С:Предприятие версии 7.7;

Мастер обработки

Мастер обработки поможет в интерактивном пошаговом режиме настроить все необходимые этапы обработки данных.

В Deductor доступны следующие обработчики:

• Очистка данных:

Парциальная обработка - включает в себя алгоритмы восстановления, сглаживания и редактирования аномальных данных.

Факторный анализ - используется для понижения размерности входных факторов.

Корреляционный анализ - устранение незначущих факторов.

Дубликаты и противоречия - выявляет дубликаты и противоречивые записи в исходном наборе данных.

Фильтрация - выполняет фильтрацию записей выборки по заданным условиям.

• Трансформация данных:

Настройка набора данных - выполняет настройку параметров полей.

Скользящее окно - позволяет осуществлять преобразование данных методом скользящего окна.

Дата и время - производит обработку данных в формате "дата" и "время".

Квантование - выполняет квантование значений выборки.

Сортировка - выполняет сортировку записей в исходной выборке данных.

Слияние - объединение данных из двух таблиц.

Замена - замена значений по таблице подстановки.

Группировка данных - производит группировку данных.

Разгруппировка данных - восстанавливает выборку, к которой была применена операция группировки.

• Data Mining:

Прогнозирование - выполняет прогнозирование временного ряда.

Автокорреляция - выполняет автокорреляционный анализ данных.

Линейная регрессия - строит модель данных в виде набора коэффициентов линейного преобразования.

Логистическая регрессия - строит бинарную логистическую регрессионную модель.

Нейросеть - выполняет обработку данных с помощью многослойной нейронной сети.

В этом режиме Мастер обработки позволяет сконструировать нейронную сеть с заданной структурой, определить ее параметры и обучить с помощью одного из доступных в системе алгоритмов обучения. В результате будет получен эмулятор нейронной сети, который может быть использован для решения задач прогнозирования, классификации, поиска скрытых закономерностей, сжатия данных и многих других приложений. Подробнее с нейронными сетями можно познакомиться в руководстве пользователя, расположенном на сайте www.basegroup.ru.

Настройка и обучение нейронной сети состоит из следующих шагов:

1. Настройка назначений полей.
2. Настройка нормализации полей.
3. Настройка обучающей выборки.
4. Настройка структуры нейронной сети.
5. Выбор алгоритма и настройка параметров обучения.
6. Настройка условий остановки обучения.
7. Запуск процесса обучения.
8. Выбор способа отображения данных.

9. Сведения об узле.

Дерево решений - выполняет обработку данных с помощью деревьев решений.

Самоорганизующиеся карты - выполняет кластеризацию данных.

Алгоритм функционирования самоорганизующихся карт (Self Organizing Maps - SOM) представляет собой один из вариантов кластеризации многомерных векторов - алгоритм проецирования с сохранением топологического подобия.

Примером таких алгоритмов может служить алгоритм k -ближайших средних (k -means). Важным отличием алгоритма SOM является то, что в нем все нейроны (узлы, центры классов) упорядочены в некоторую структуру (обычно двумерную сетку). При этом, в ходе обучения модифицируется не только нейрон-победитель (нейрон карты, который в наибольшей степени соответствует вектору входов и определяет к какому классу относится пример), но и его соседи, хотя и в меньшей степени. За счет этого SOM можно считать одним из методов проецирования многомерного пространства в пространство с более низкой размерностью. При использовании этого алгоритма, вектора, схожие в исходном пространстве, оказываются рядом и на полученной карте.

SOM подразумевает использование упорядоченной структуры нейронов. Обычно используются одно- и двумерные сетки. При этом, каждый нейрон представляет собой n -мерный вектор-столбец, где n определяется размерностью исходного пространства (размерностью входных векторов). Применение одно- и двумерных сеток связано с тем, что возникают проблемы при отображении пространственных структур большей размерности (при этом опять возникают проблемы с понижением размерности до двумерной, представимой на мониторе).

Обычно, нейроны располагаются в узлах двумерной сетки с прямоугольными или шестиугольными ячейками. При этом, как было сказано выше, нейроны также взаимодействуют друг с другом. Величина этого взаимодействия определяется расстоянием между нейронами на карте.

При реализации алгоритма SOM заранее задается конфигурация сетки (прямоугольная или шестиугольная), а также количество нейронов в сети. Некоторые источники рекомендуют использовать максимально возможное количество нейронов в карте. При этом начальный радиус обучения (*neighborhood* в англоязычной литературе) в значительной степени влияет на способность обобщения при помощи полученной карты. В случае, когда количество узлов карты превышает количество примеров в обучающей выборке, успех использования алгоритма в большой степени зависит от подходящего выбора начального радиуса обучения. Однако, в случае, когда размер карты составляет десятки тысяч нейронов, время, требуемое на обучение карты, обычно бывает слишком велико для решения практических задач. Таким образом, необходимо достигать допустимый компромисс при выборе количества узлов.

Перед началом обучения карты необходимо проинициализировать весовые коэффициенты нейронов. Удачно выбранный способ инициализации может существенно ускорить обучение и привести к получению более качественных результатов.

Существуют три способа инициирования начальных весов.

- *Инициализация случайными значениями*, когда всем весам даются малые случайные величины.
- *Инициализация примерами*, когда в качестве начальных значений задаются значения случайно выбранных примеров из обучающей выборки.
- *Линейная инициализация*, в этом случае веса иницируются значениями векторов, линейно упорядоченных вдоль линейного подпространства, проходящего между двумя главными собственными векторами исходного набора данных.

Обучение карты заключается в последовательности коррекции векторов, представляющих собой нейроны. На каждом шаге обучения, из исходного набора данных случайно выбирается один из векторов, а затем производится поиск наиболее похожего на него вектора коэффициентов нейронов. При этом выбирается нейрон-победитель, который

наиболее похож на вектор входов. Под похожестью в данной задаче понимается расстояние между векторами, обычно вычисляемое в евклидовом пространстве.

После того, как найден нейрон-победитель, производится корректировка весов карты. При этом вектор, описывающий нейрон-победитель и векторы, описывающие его соседей в сетке, перемещаются в направлении входного вектора.

Обучение состоит из двух основных фаз: на первоначальном этапе выбирается достаточно большое значение скорости обучения и радиуса обучения, что позволяет расположить вектора нейронов в соответствии с распределением примеров в выборке, а затем производится точная подстройка весов, когда значения параметров скорости обучения много меньше начальных. В случае использования линейной инициализации, первоначальный этап грубой подстройки может быть пропущен.

Самоорганизующиеся карты могут использоваться для решения таких задач как моделирование, прогнозирование, поиск закономерностей в больших массивах данных, выявление наборов независимых признаков и сжатие информации.

В результате обучения самоорганизующейся карты в исходную выборку данных будут добавлены следующие поля:

- *<ИМЯ ПОЛЯ>_OUT* - содержат значения выходных полей, рассчитанные картой.
- *Номер ячейки* - содержит номер ячейки карты в которую попала данная запись.
- *Расстояние до центра ячейки* - содержит значение расстояния от данной записи до центра ячейки в которую эта запись попала.
- *Номер кластера* - указывается номер кластера, где расположена ячейка в которую попала данная запись исходной выборки.
- *Расстояние до центра кластера* - указывается значение расстояния от ячейки, куда попала данная запись исходной выборки, до центра кластера.
- *<ИМЯ ПОЛЯ>_ERR* - содержит среднеквадратичную ошибку рассогласования реального значения поля и значения, рассчитанного картой.

Процесс построения и обучения самоорганизующейся карты содержит следующие этапы:

1. Выбор назначений полей.
2. Настройка нормализации полей.
3. Настройка обучающей выборки.
4. Настройка параметров обучения.
5. Настройка условий остановки обучения.
6. Настройка параметров обучения.
7. Запуск процесса обучения.
8. Выбор способа отображения данных.

Ассоциативные правила - обнаружение зависимостей между связанными событиями.

Пользовательская модель - задание модели вручную по формулам.

• **Прочее:**

Скрипт - применение модели к новым данным.

Калькулятор - позволяет сформировать новое поле выборки как результат вычислений над данными из других полей.

Условие - позволяет организовать условное выполнение сценария обработки данных.

Команда ОС - компонент обеспечивает формирование и запуск различных команд операционной системы.

Мастер визуализации

Мастер визуализации поможет вам в интерактивном пошаговом режиме выбрать и настроить наиболее удобный способ представления данных.

Мастер экспорта

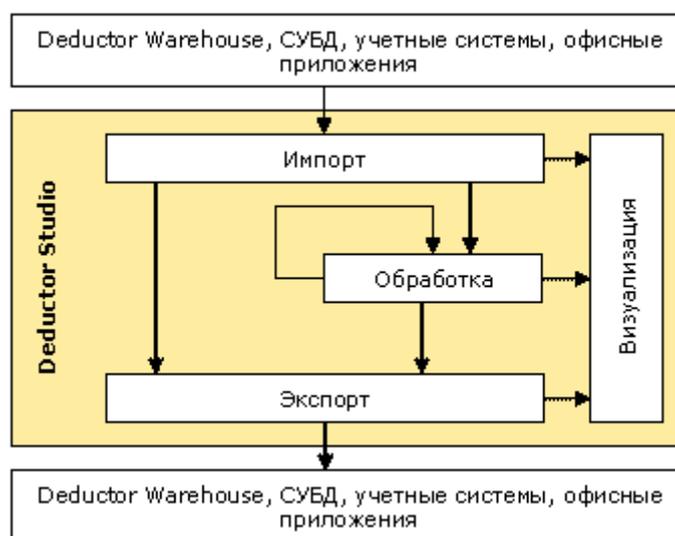
Мастер экспорта позволяет в интерактивном пошаговом режиме выполнить экспорт данных в файлы наиболее распространенных форматов.

Доступные приёмники данных мастера экспорта:

- Хранилища данных:
 - o Deductor Warehouse - экспорт в хранилище данных.
- Базы данных:
 - o База данных - экспорт данных в базы данных различных типов.
- Файлы:
 - o Microsoft Excel - книга Microsoft Excel (*.xls).
 - o Microsoft Word - документ Microsoft Word в формате RTF (*.rtf).
 - o Экспорт в HTML - экспорт в файлы формата HTML.
 - o Экспорт в текстовый файл - экспорт в текстовый файл с разделителями (табуляция).
 - o Экспорт в DBF - экспорт в плоские таблицы DBase III+.
 - o Экспорт в XML - экспорт в файлы формата XML (расширяемый язык разметки).
- Web - серверы
 - o Экспорт через SMTP сервер - отправка сообщения через настроенный SMTP сервер.
- Прочие:
 - o Экспорт в буфер обмена - экспорт в стандартный буфер обмена Windows.

Отправной точкой для анализа всегда является процедура импорта данных. Полученный набор данных может быть обработан любым доступным способом. Результатом обработки также является набор данных, который в свою очередь опять может быть обработан. Результаты обработки можно просмотреть множеством способов и экспортировать в наиболее популярные форматы. Последовательность действий, которые необходимо провести для анализа данных, является сценарием, который можно автоматически выполнять на любых данных.

Укрупнённая структурная схема проекта:



Варианты поставки

Имеются 3 варианта поставки Deductor: Academic, Professional и Enterprise:

- **Academic** – бесплатная версия предназначена только для образовательных целей. Использование данной версии в коммерческих целях запрещено. В ней ограничены

возможности интеграции и автоматической обработки. Поддерживается только 2 источника и приемника данных: Deductor Warehouse и текстовые файлы с разделителями.

- **Professional** – версия ориентирована на профессиональный анализ данных и рассчитана на использование в рамках рабочих групп. В этой версии отсутствуют ограничения на количество обрабатываемых записей, поддерживается работа с множеством источников данных, хранилищем данных на базе бесплатной СУБД Firebird, пакетное выполнение сценариев, все механизмы обработки и визуализации данных.
- **Enterprise** – версия ориентирована на промышленное корпоративное применение. В ней реализован весь функционал, имеющийся в версии Professional. Кроме того, в поставку данной версии входят Deductor Server и Deductor Client для удаленной работы с системой, поддержка хранилищ данных на платформах Oracle и MS SQL (помимо Firebird), поддержка концепции виртуального хранилища данных, реализация OLE сервера и прочие механизмы, необходимые для корпоративного использования аналитической платформы.

Прайс-лист:

Для принятия обоснованного решения воспользуйтесь сравнительной характеристикой вариантов поставки.

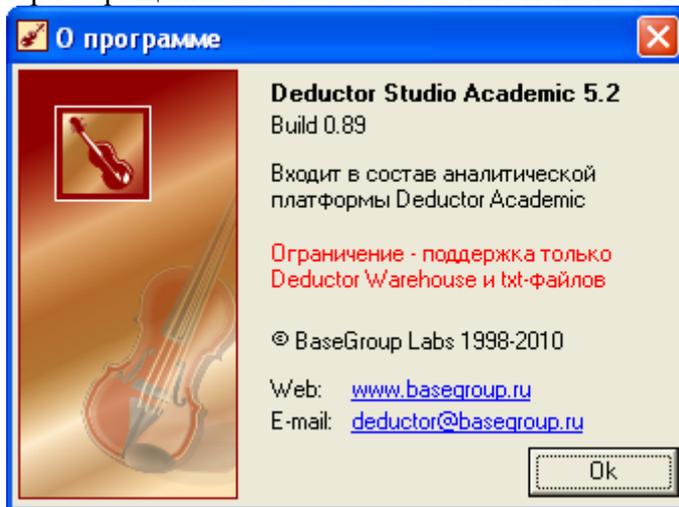
Модуль	Academic (руб.)	Professional (руб.)	Enterprise (руб.)
Studio	бесплатно	29 000	звоните
Viewer	-	8 700	звоните
Warehouse	бесплатно	бесплатно	бесплатно
Server	-	-	звоните
Client	-	-	бесплатно

Лекция 5. Deductor. Нейронная сеть. Настройка и обучение.

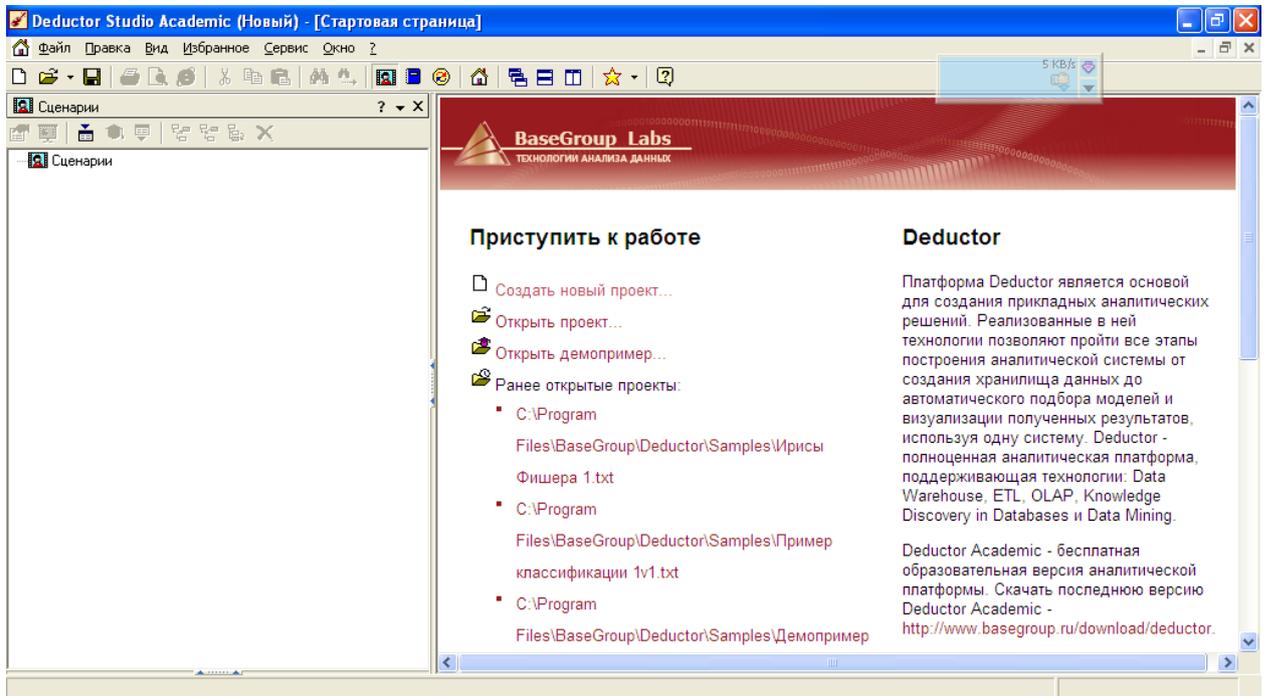
Технология работы с Deductor Academic предусматривает выполнение следующих операций: Импорт – Обработка – Визуализация – Экспорт.

Импорт.

При обращении к Deductor Academic появляется окно:



После нажатия ОК:

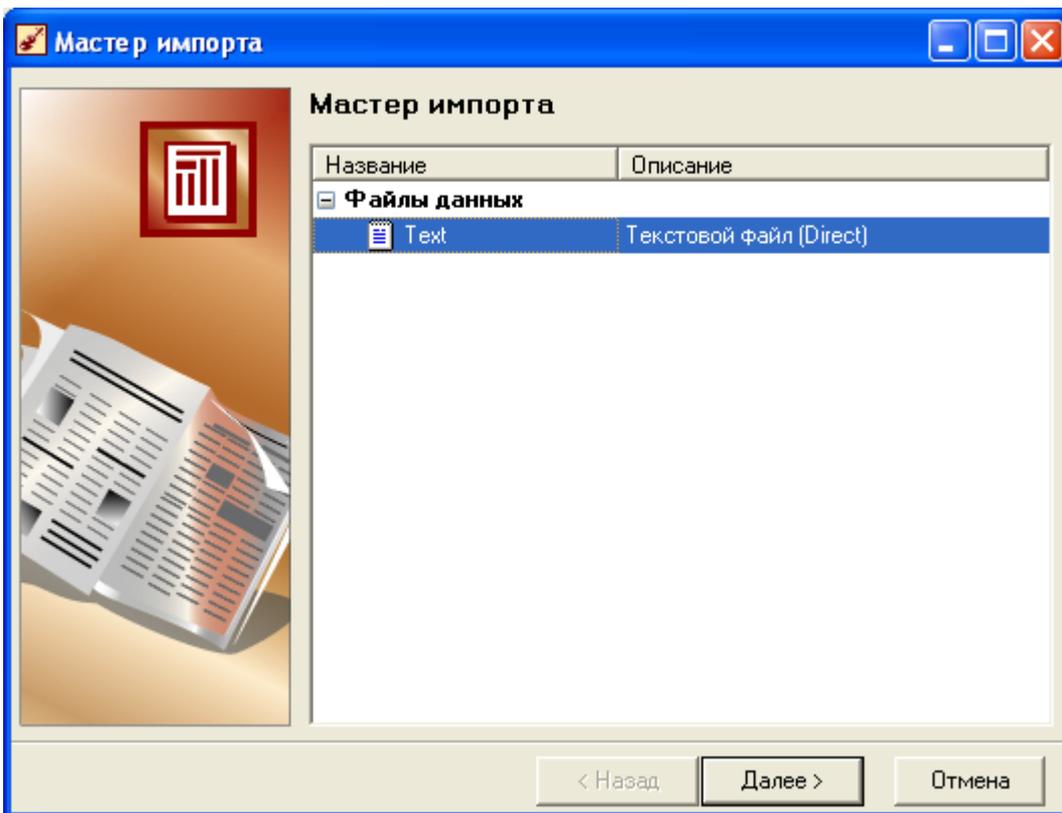


Появляется первое окно, предоставляющее очень мало возможностей.

После загрузки пакета доступен только один элемент управления - Мастер импорта:

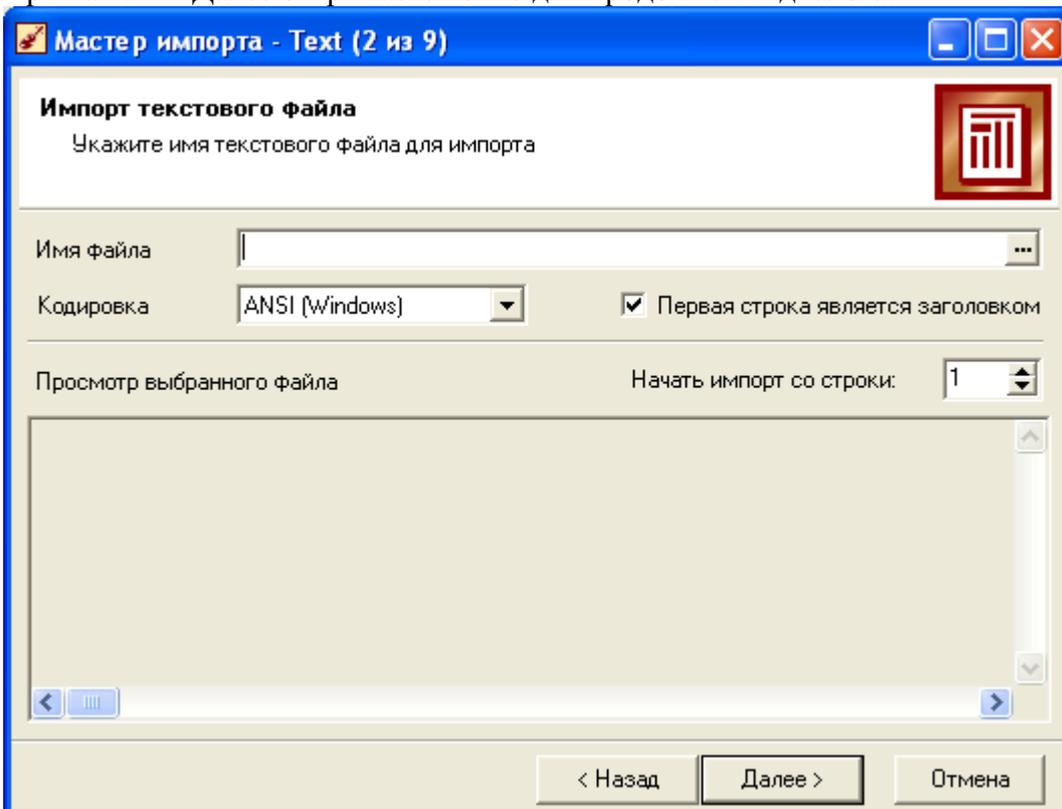


С помощью мастера импорта можно только загрузить в Deductor текстовый файл.

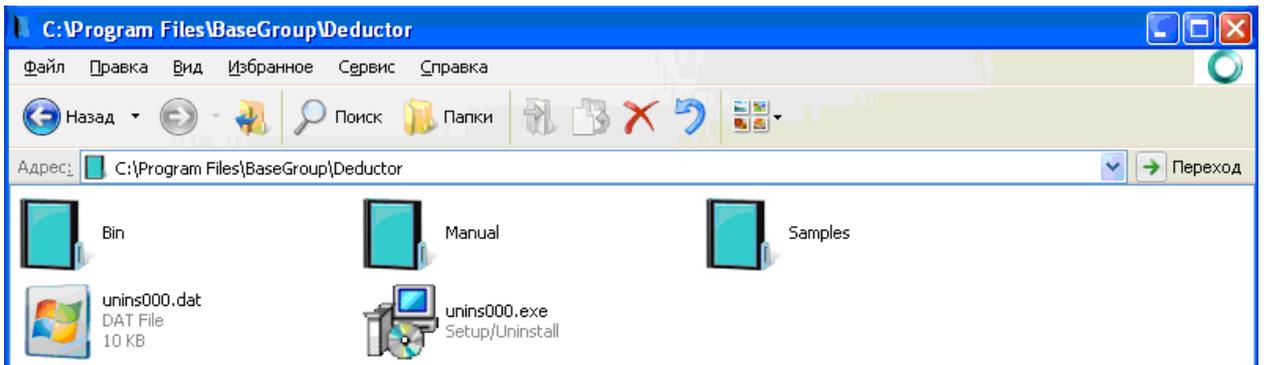


Мастер готов принять текстовый набор данных.

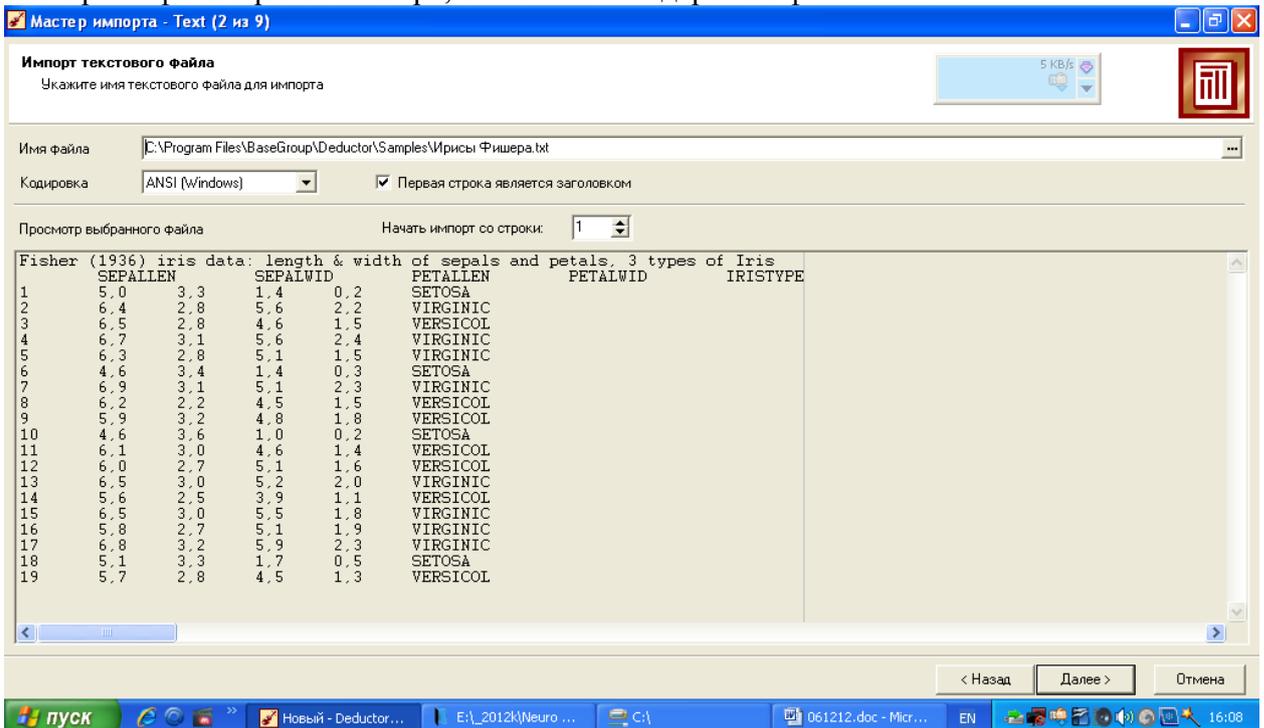
При нажатии Далее открывается окно для продолжения диалога:



При нажатии на многоточие:  открывается переход средствами Windows к данным пакета Deductor:

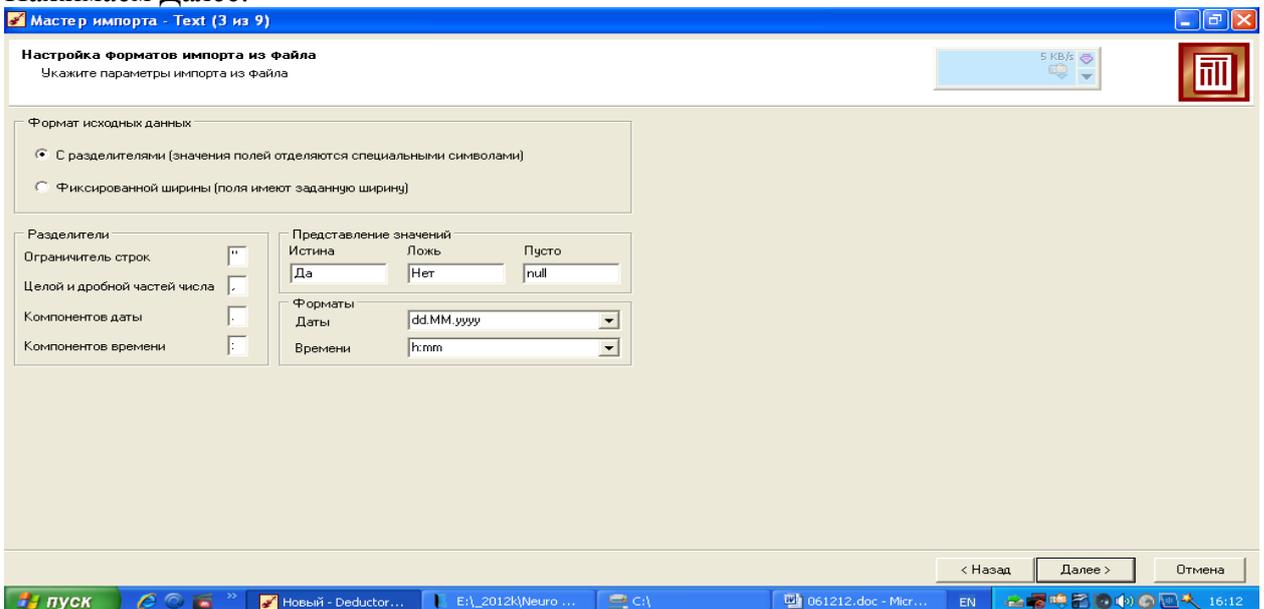


Среди этих данных в папке Samples хранятся файлы с различными наборами данных. В этот же файл можно поместить свои файлы, подготовленные для исследования. Выбираем файл Ирисы Фишера, взятый из стандартных файлов пакета Статистика:



Теперь можно начинать работать с выбранным файлом.

Нажимаем Далее:



Работаем по умолчанию:

Мастер импорта - Текст (4 из 9)

Параметры импорта файла с разделителями

Укажите символ-разделитель столбцов и другие вспомогательные параметры импорта

Символом-разделителем является

Символ табуляции Пробел Точка
 Точка с запятой Запятая Другой

Считать последовательные разделители одним

	SEPALLEN	SEPALWID	PETALLEN	PETALWID	IRISTYPE
▶	1	5	3,3	1,4	0,2
	2	6,4	2,8	5,6	2,2
	3	6,5	2,8	4,6	1,5
	4	6,7	3,1	5,6	2,4
	5	6,3	2,8	5,1	1,5
	6	4,6	3,4	1,4	0,3
	7	6,9	3,1	5,1	2,3
	8	6,2	2,2	4,5	1,5
	9	5,9	3,2	4,8	1,8
	10	4,6	3,6	1	0,2

< Назад Далее > Отмена

пуск D:\2013\2... C:\Program ... Лекция 6. D... Новый - Де... Безымяны... EN | 11:07

Далее:

Мастер импорта - Текст (6 из 9)

Импорт текстового файла

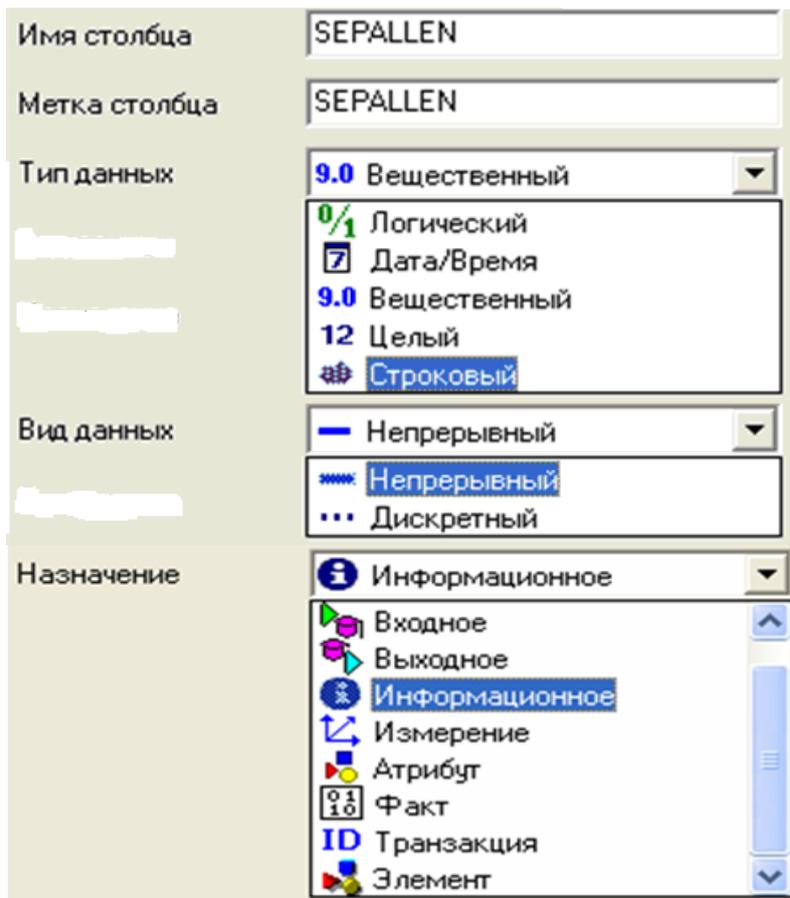
Укажите параметры столбцов

Имя столбца	Метка столбца	Тип данных	Вид данных	Назначение
SEPALLEN	SEPALLEN	9.0 Вещественный	Непрерывный	Информационное
SEPALWID				
PETALLEN				
PETALWID				
IRISTYPE				

< Назад Далее > Отмена

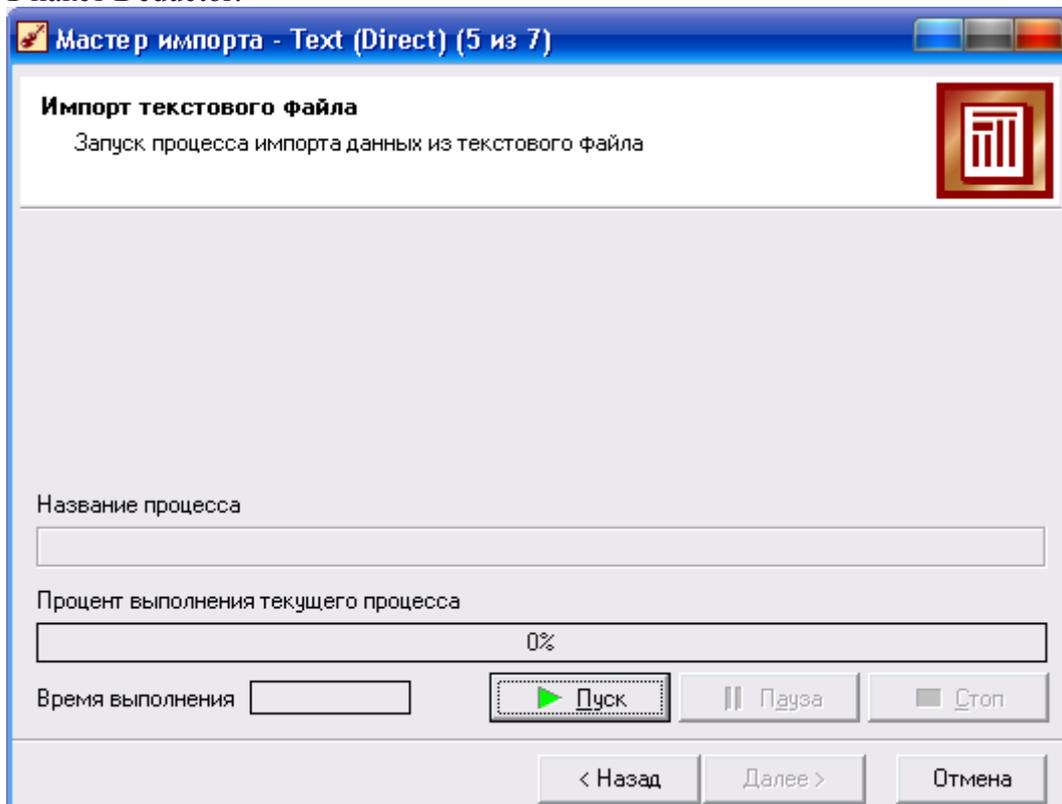
пуск D:\2013\2... C:\Program ... Лекция 6. D... Новый - Де... Безымяны... EN | 11:07

Для импорта текстового файла необходимо указать параметры столбцов: имя столбца, метка столбца, тип данных, вид данных:

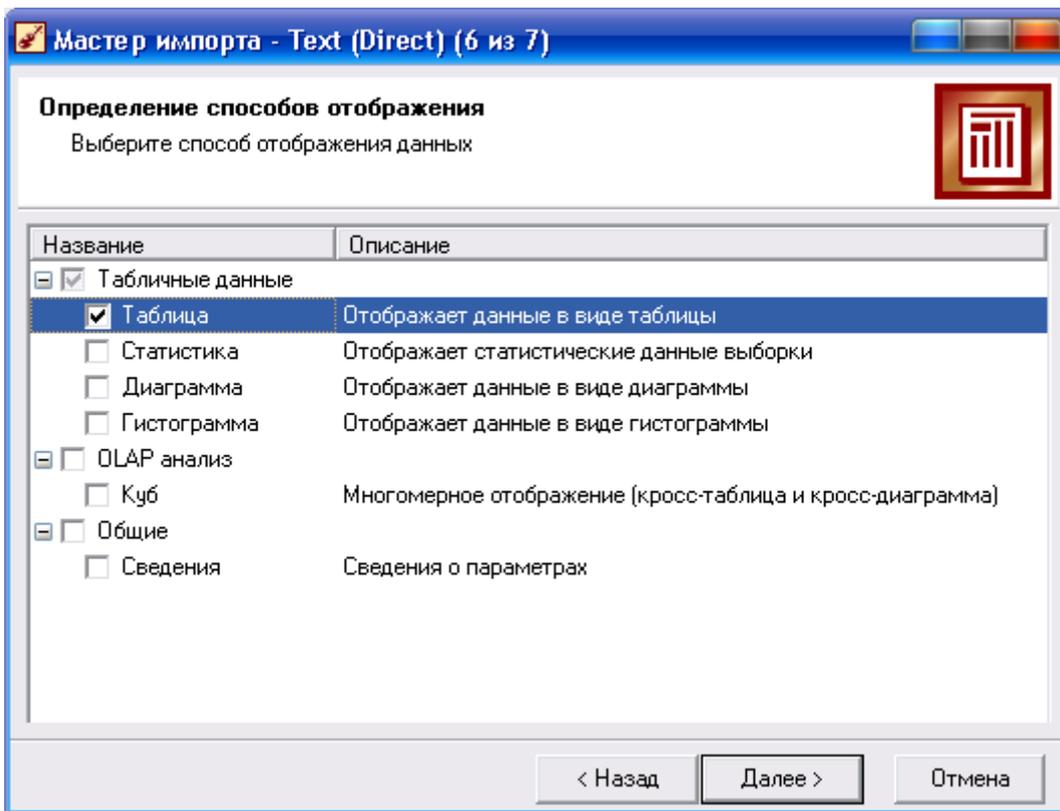


Для ввода характеристик какой-нибудь колонки надо активизировать её имя, щёлкнув по нему.

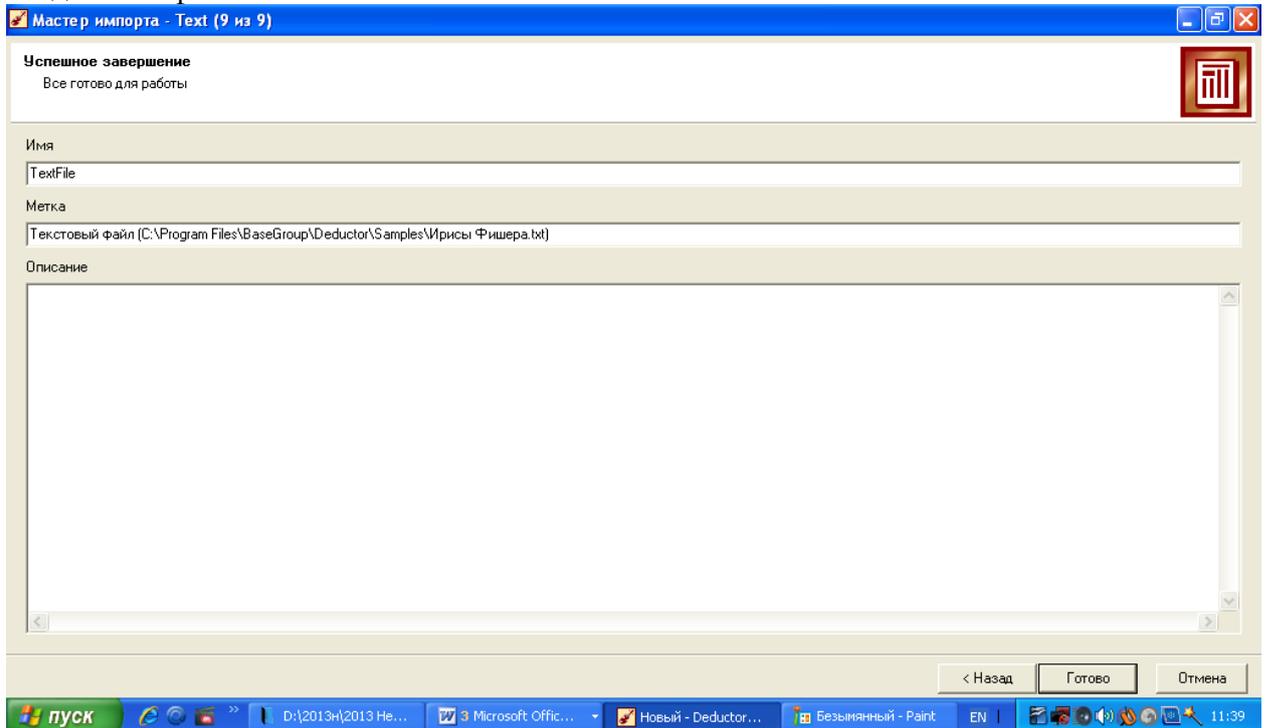
На данном этапе характеризовать колонки не надо. Можно ограничиться импортом файла в пакет Deductor:



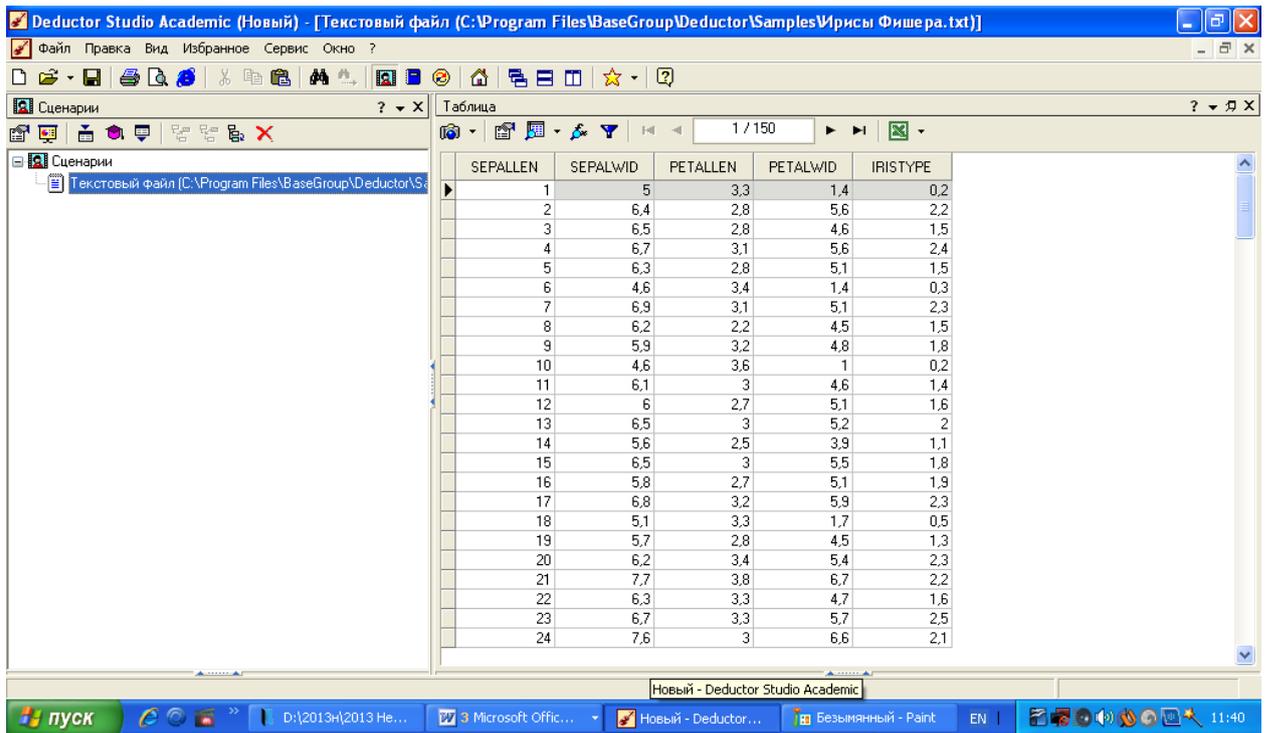
После завершения импорта определяется способ отображения введённых данных:



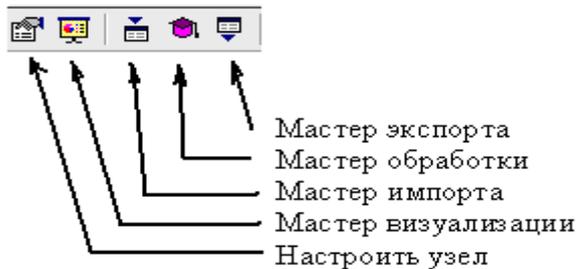
При успешном завершении загрузки данных система сообщает имя и местоположение введённого файла:



Изменяется вид главного окна пакета:



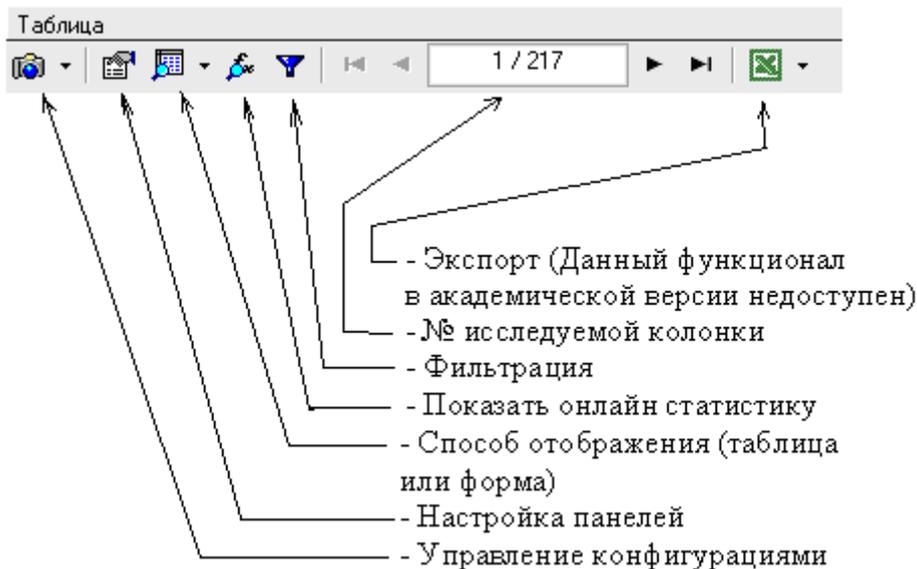
Открываются следующие элементы управления, которые ранее были недоступными:



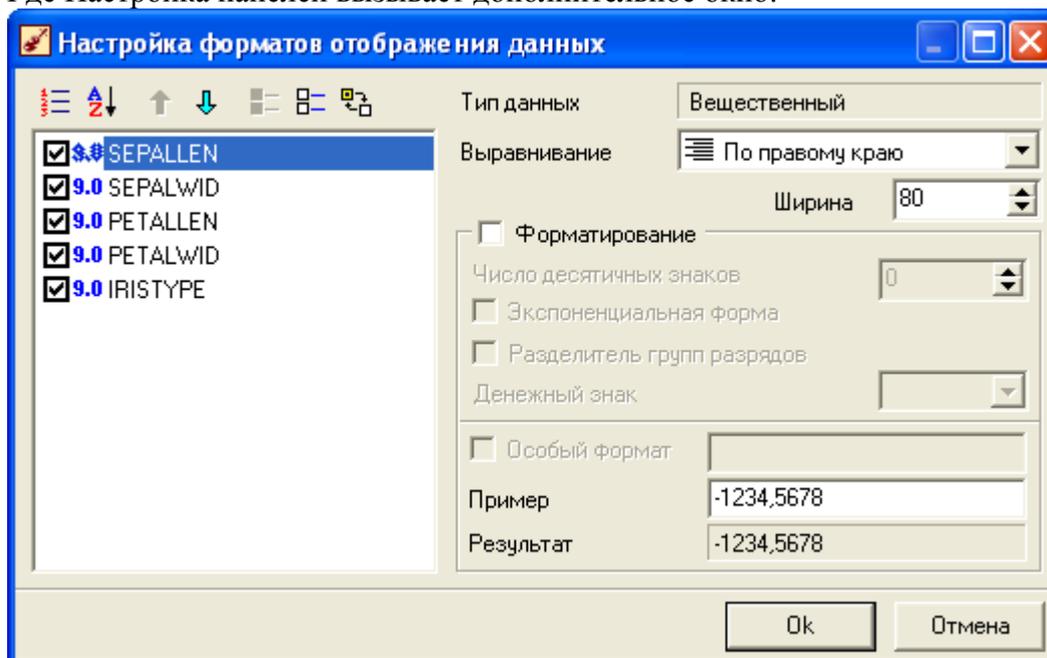
С их помощью можно продолжить работу с нейронной сетью.

Визуализация 1.

Кроме того, сформированная таблица может быть изучена и преобразована с помощью следующего меню:



Где Настройка панелей вызывает дополнительное окно:



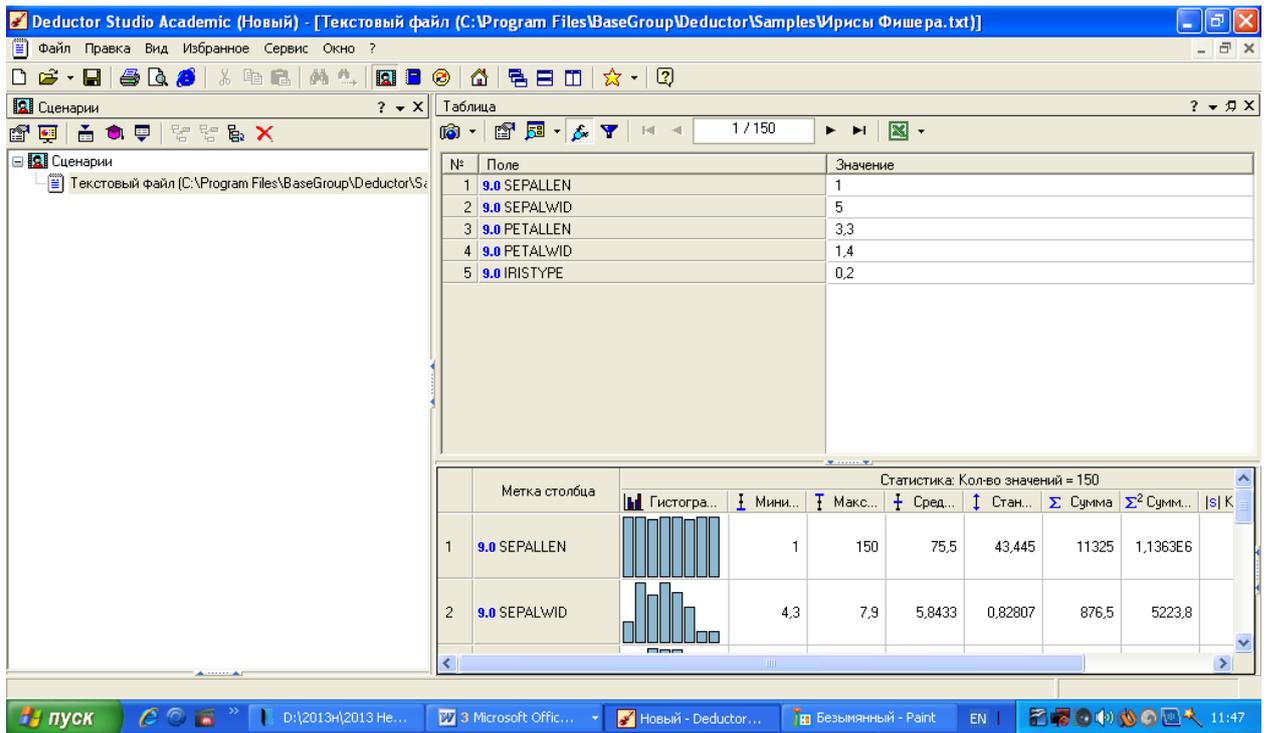
“Форма” вместо полной таблицы выводит на экран одну её строку

№	Поле	Значение
1	9.0 SEPALLEN	1

“Показать online статистику”:

№	Поле	Значение
217	9.0 SEPALLEN	1

В общем окне появляется нижняя вставка в виде таблицы:



Онлайн-статистика из нижней вставки выводится в виде следующей таблицы:

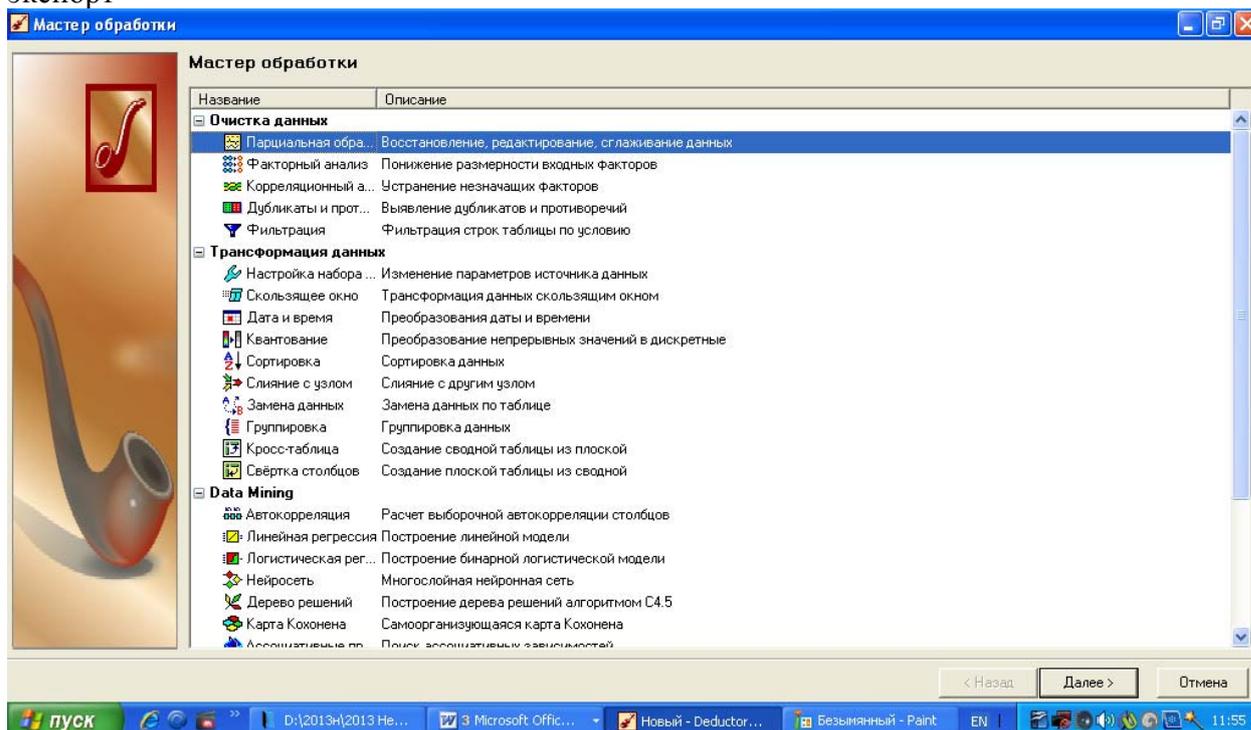
Метка столбца	Гистогра...	Мини...	Макс...	Сред...	Стан...	Сумма	Σ ² Сумм...	s K
1 9.0 SEPALLEN		1	150	75,5	43,445	11325	1,1363E6	
2 9.0 SEPALWID		4,3	7,9	5,8433	0,82807	876,5	5223,8	
3 9.0 PETALLEN		2	4,4	3,0573	0,43587	458,6	1430,4	
4 9.0 PETALWID		1	6,9	3,758	1,7653	563,7	2582,7	
5 9.0 IRISTYPE		0,1	2,5	1,1993	0,76224	179,9	302,33	

Обработка.

Если покинуть мастер импорта, то можно перейти к мастеру обработки:



Ещё раз о последовательности работы с пакетом: Импорт – обработка – визуализация - экспорт



Мастер обработки нейронной сети

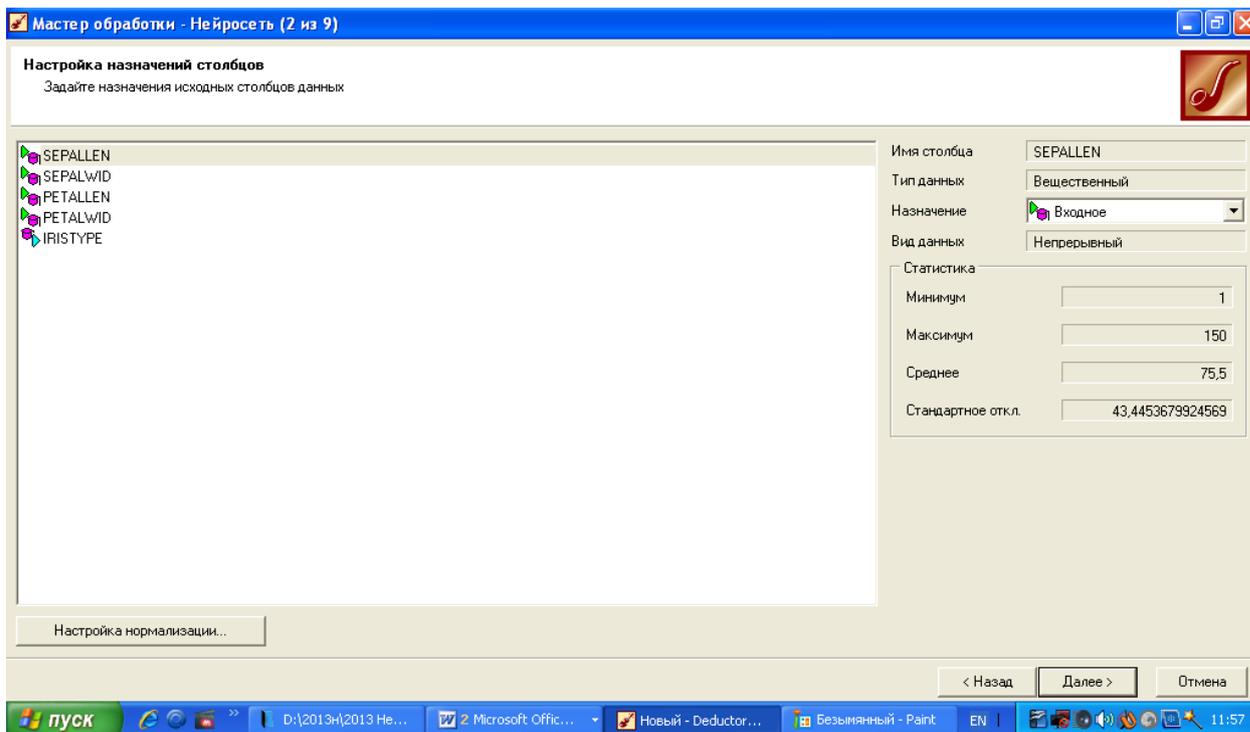
В этом режиме Мастер обработки позволяет сконструировать нейронную сеть с заданной структурой, определить ее параметры и обучить с помощью одного из доступных в системе алгоритмов обучения. В результате будет получен эмулятор нейронной сети, который может быть использован для решения задач прогнозирования, классификации, поиска скрытых закономерностей, сжатия данных и многих других приложений. Подробнее с нейронными сетями можно познакомиться в руководстве пользователя, расположенном на сайте www.basegroup.ru.

Настройка и обучение нейронной сети состоит из следующих шагов:

1. Настройка назначений полей.
2. Настройка нормализации полей.
3. Настройка обучающей выборки.
4. Настройка структуры нейронной сети.
5. Выбор алгоритма и настройка параметров обучения.
6. Настройка условий остановки обучения.
7. Запуск процесса обучения.
8. Выбор способа отображения данных.

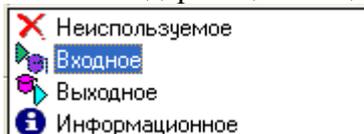
Настройка назначения полей

Здесь необходимо определить, как будут использоваться поля исходного набора данных при обучении нейронной сети и практической работе с ней.



В левой части окна представлен список всех полей исходного набора данных. Для настройки поля следует выделить его в списке, при этом в правой части окна будут отображены его параметры:

- *Имя поля* - идентификатор поля, определенный для него в источнике данных. Изменить его здесь нельзя.
- *Тип данных* - тип данных, содержащихся в поле (вещественный, строковый, дата). Он также задается в источнике данных и здесь изменен быть не может.
- *Назначение* - здесь необходимо выбрать порядок использования данного поля при обучении и работе нейронной сети. Выбор производится с помощью списка, открываемого кнопкой и содержащего следующие варианты:



- Входное - значения поля будут подаваться на соответствующие входы нейронной сети, т.е. являться исходными данными для обучения и дальнейшей практической работы обученной нейронной сети.
- Выходное - значения данного поля будут являться эталонными при обучении нейронной сети, а при практической работе с ней будут содержать результаты обработки входных полей.
- Информационное - поле не будет использоваться при обучении сети, но будет помещено в результирующий набор в исходном состоянии.
- Неиспользуемое - поле не будет использоваться при обучении и работе нейронной сети и будет исключено из результирующей выборки. В отличие от непригодного такое поле может быть использовано, если в этом возникнет необходимость.
- *Вид данных* - указывает на характер данных, содержащихся в поле (непрерывный или дискретный). Изменить это свойство здесь нельзя.

Замечание: к выбору входных и выходных полей следует относиться особенно внимательно, принимая во внимание следующие соображения:

- *Входные и выходные поля определяются характером решаемой задачи: входные поля содержат исходные данные, подлежащие обработке, а Выходные - результаты обработки.*

- *Использование полей, данные в которых не имеют отношения к решаемой задаче, крайне нежелательно, поскольку они только загромождают исходную и результирующую выборки и замедляют работу сети.*

В случае, если текущее поле содержит непрерывные (числовые) данные, в правой нижней части окна появляется секция "Статистика",

Статистика	
Минимум	1
Максимум	150
Среднее	75,5
Стандартное откл.	43,4453679924569

в которой отображаются максимальное и минимальное значения поля, его среднее значение и стандартное отклонение. Если выделенное поле содержит дискретные (строковые) данные, то для него открывается секция "Уникальные значения", в которой отображается общее число уникальных значений в поле, а также список самих уникальных значений.

Настройка нормализации полей

С помощью кнопки "Настройка нормализации" можно вызвать окно настройки нормализации полей для указания параметров нормализации исходных данных.

Метка столбца	Мин.	Макс.
Входные столбцы		
SEPALLEN	1	150
SEPALWID	4,3	7,9
PETALLEN	2	4,4
PETALWID	1	6,9
Выходные столбцы		
IRISTYPE	0,1	2,5

Целью нормализации значений полей является преобразование данных к виду, наиболее подходящему для обработки средствами пакета Deductor. Для таких обработчиков, как нейронная сеть, линейная модель прогнозирования,

самоорганизующиеся карты, данные, поступающие на вход, должны иметь числовой тип (т.е. иметь непрерывный характер), а их значения должны быть распределены в определенном диапазоне. В этом случае нормализатор может преобразовать дискретные данные к набору уникальных индексов или непрерывным значениям.

В окне настройки нормализации полей слева приведен полный список входных и выходных полей. При этом каждое поле помечено значком, обозначающим вид нормализации поля:

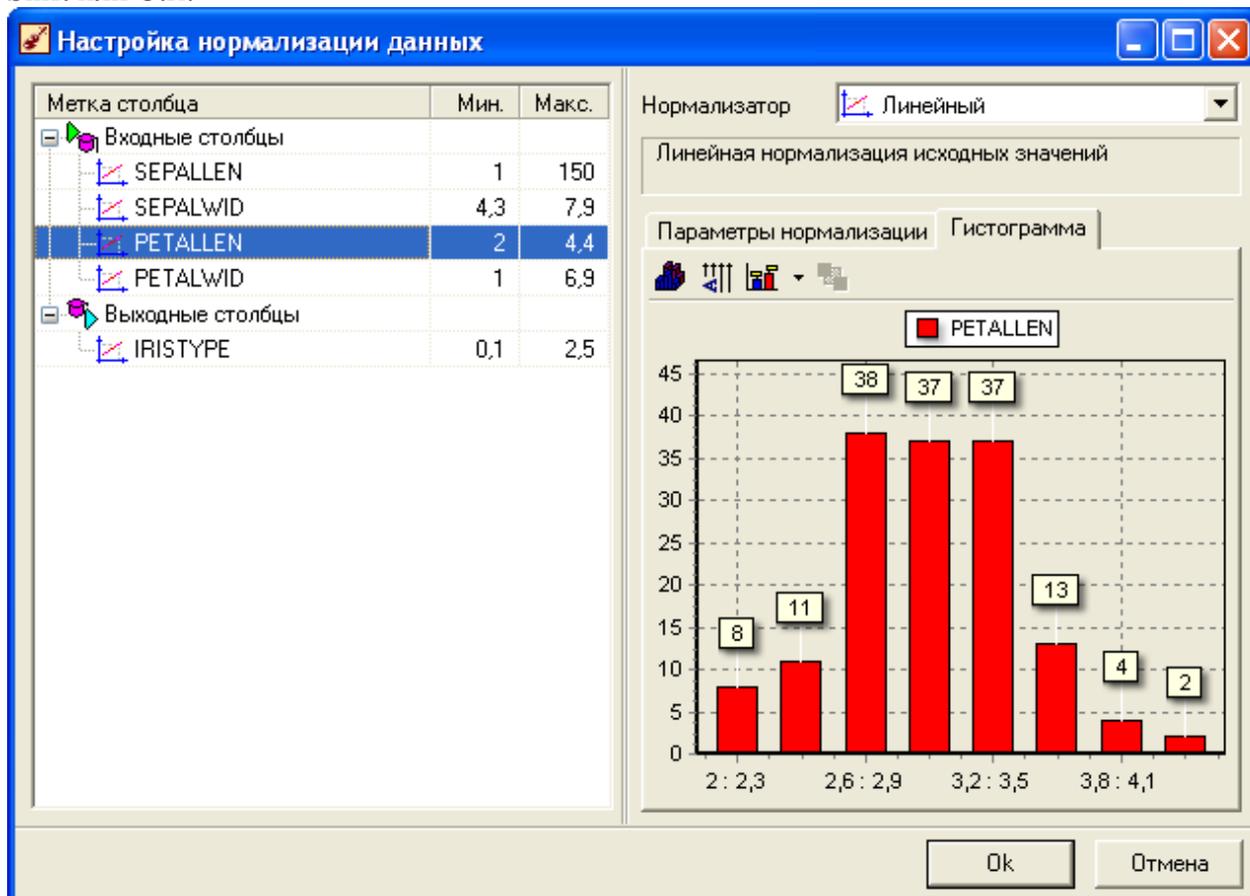
- Линейная - линейная нормализация исходных значений.
- Уникальные значения - преобразование уникальных значений в их индексы.
- Битовая маска - преобразование дискретных значений в битовую маску.

Параметры нормализации

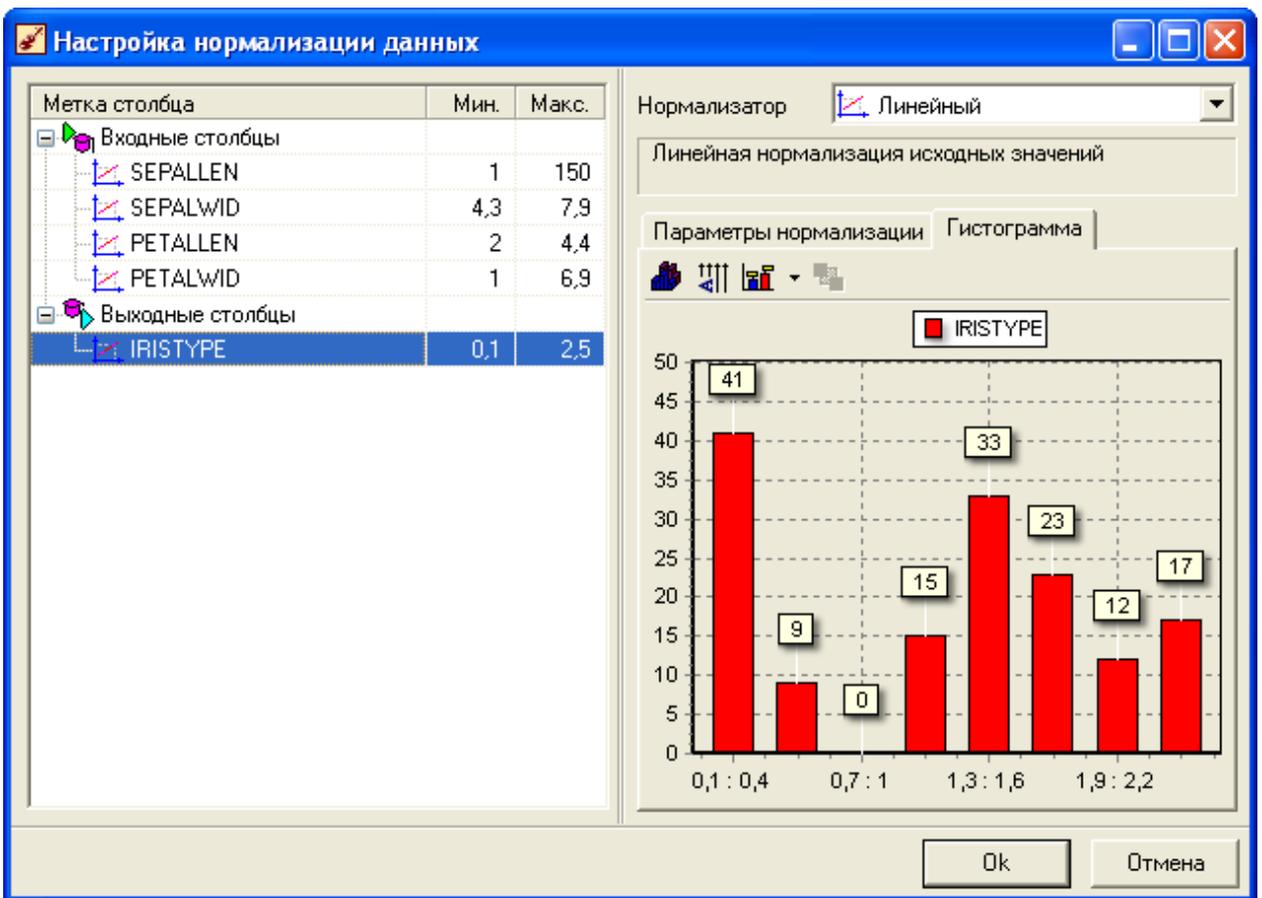
В правой части окна для выделенного поля отображаются параметры нормализации. Правила нормализации и представляемые пакетом Deductor значения отражены в технической документации [4].

Гистограмма

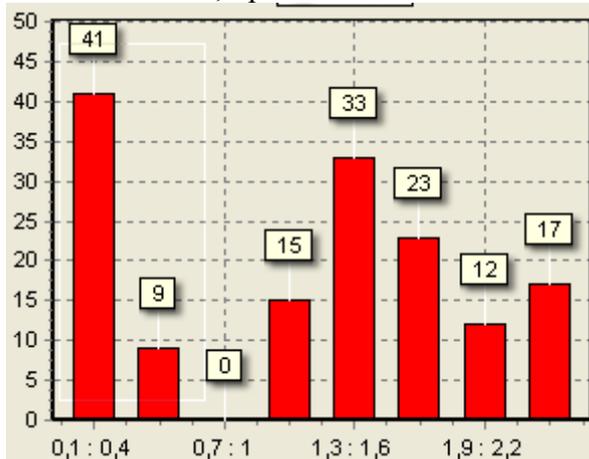
На закладке "Гистограмма" отображается гистограмма значений выделенного поля независимо от способа нормализации. Высота столбца показывает число значений, попадающих в определенный интервал. Границы интервала подписаны под столбцами гистограммы. Для отображения на гистограмме значений из нескольких полей следует выделить в списке доступных полей нужные с помощью мыши при нажатых клавишах Shift или Ctrl.



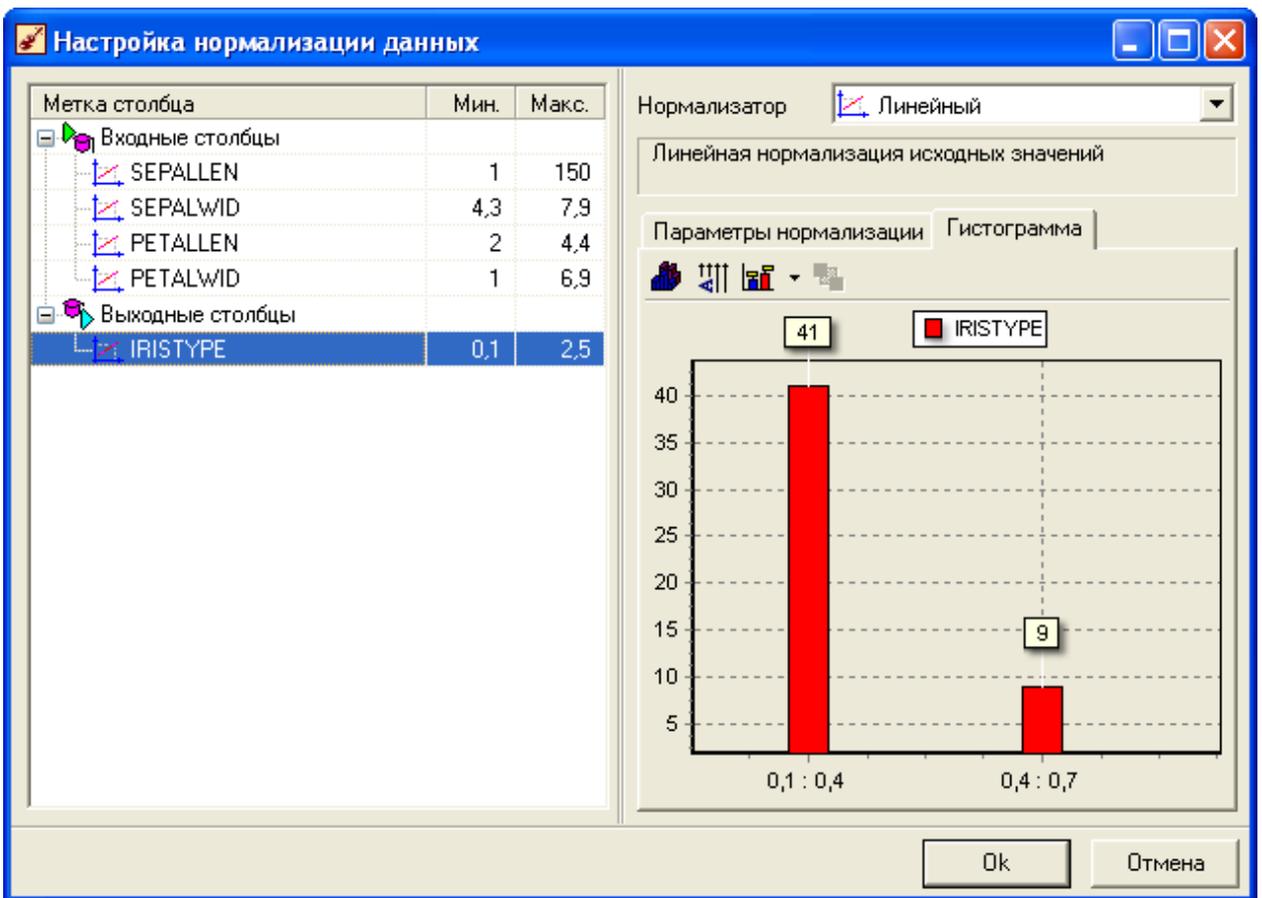
ИЛИ



При работе с гистограммой предусмотрена возможность увеличения масштаба просмотра произвольной области гистограммы. Для этого следует, удерживая левую кнопку мыши нажатой, выделить ту область гистограммы, которую нужно просмотреть более детально, при этом двигая мышью слева направо.

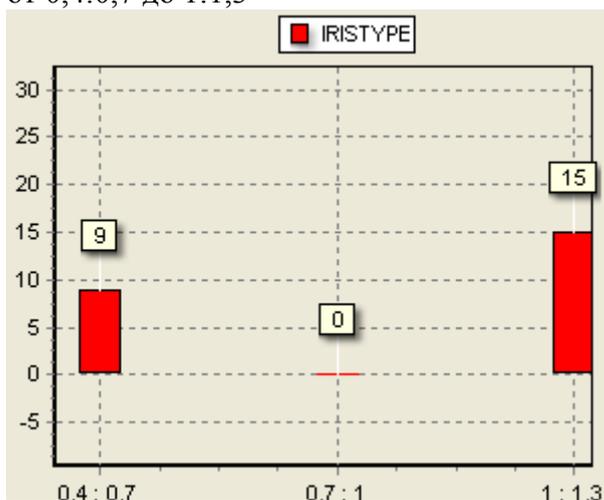


Выделение левой части гистограммы отмечается белым прямоугольником. Как только кнопка мыши будет отпущена, масштаб просмотра выделенной области будет увеличен.



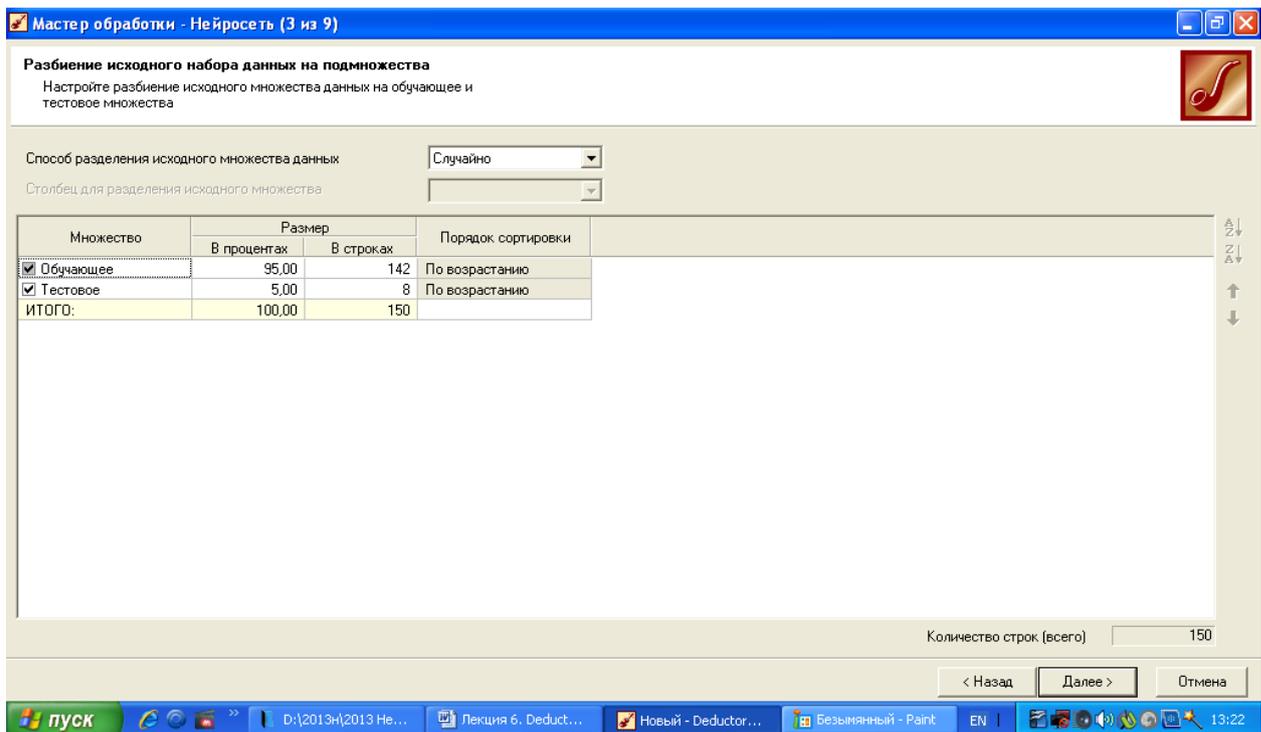
Для дальнейшего увеличения масштаба данную процедуру можно повторить. При выделении области гистограммы движением мыши справа налево масштаб просмотра гистограммы будет возвращен к исходному независимо от размера выделенной области.

Направив указатель мыши в произвольную точку гистограммы и перемещая ее с нажатой правой кнопкой, можно перемещать диаграмму по экрану, делая доступными для просмотра различные ее части. Например, выделим для рассмотрения часть гистограммы от 0,4:0,7 до 1:1,3



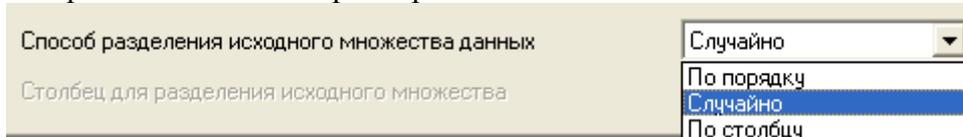
Настройка обучающей выборки

Здесь можно разбить обучающую выборку для на два множества - обучающее и тестовое.



- *Обучающее множество* - включает записи (примеры), которые будут использоваться в качестве входных данных, а также соответствующие желаемые выходные значения.
- *Тестовое множество* - также включает записи, содержащие входные и желаемые выходные значения, но используемое не для обучения модели, а для проверки его результатов.

Для разбиения исходного множества на обучающее и тестовое необходимо настроить несколько параметров:



- Из списка "Способ разделения исходного множества" выбирается порядок отбора записей во все множества. Если выбран вариант "по порядку", то порядок следования записей при их разделении не меняется. Множества последовательно формируются в соответствии с определенным для них числом записей. Если выбран вариант "случайно", то отбор записей происходит случайным образом.

- Затем необходимо указать, какие множества будут использоваться. Для того чтобы множество было сформировано, нужно установить флажок слева от его названия. Если флажок сброшен, то множество использовано не будет. Обучающее множество используется всегда, поэтому сбросить флажок для него нельзя.

- Для каждого из используемых множеств необходимо задать его размер. Размер может быть задан непосредственно количеством записей или в процентах от объема исходной выборки. Для этого достаточно дважды щелкнуть мышью в соответствующей клетке и ввести нужное значение с клавиатуры. При этом размер, введенный в процентах, автоматически пересчитывается в количество строк и наоборот.

В поле "Количество строк (всего)" отображается общее количество записей в исходной выборке данных, которое может быть задействовано для формирования множеств.

Если суммарное число строк для всех используемых множеств меньше полного числа строк исходной выборки, то размеры множеств можно задавать произвольно. Можно, например, использовать не все записи, а только часть из них. Если же суммарное указанное число строк превышает максимальное для данной исходной выборки, то

автоматически включается баланс множеств. Т.е. при указании для одного из множеств размера, в результате которого будет превышено максимальное число записей в исходной выборке, размер остальных множеств будет автоматически уменьшен таким образом, чтобы суммарный размер множеств не превышал доступного числа записей.

В строке "Итого" указывается количество записей, задействованных во всех используемых множествах, а также процент от полного числа записей исходной выборки, который они составляют.

В столбце "Порядок сортировки" можно определить порядок следования записей внутри каждого множества. Для этого дважды щелкнуть мышью в столбце "Порядок сортировки" для соответствующего множества и с помощью появившейся кнопки выбора открыть список, в котором выбрать один из возможных вариантов:

- По возрастанию - записи в данном множестве будут следовать в порядке возрастания.
- По убыванию - записи в данном множестве будут следовать в порядке убывания.
- Случайно - записи в данном множестве будут следовать в случайном порядке.

В правой части окна расположены кнопки, которые позволяют изменять порядок применения множеств в процессе обучения:

- По возрастанию - располагает уникальные значения по возрастанию (по алфавиту).
- По убыванию - располагает уникальные значения по убыванию (против алфавита).
- Вверх - перемещает выделенное значение на одну позицию вверх по списку.
- Вниз - перемещает выделенное значение на одну позицию вниз по списку.

Важно: Для того чтобы обучающее множество было репрезентативным необходимо, чтобы все уникальные значения всех дискретных столбцов содержались в данном наборе данных.

Настройка структуры сети

На этом шаге задаются параметры, определяющие структуру нейронной сети, - количество скрытых слоев и нейронов в них, а также активационная функция нейронов.

Скриншот окна "Структура нейронной сети" (Structure of the neural network) в программе "Мастер обработки - Нейросеть (4 из 9)".

Настройка параметров:

- Нейроны в слое:
 - входном: 4
 - скрытых слоев: 1
 - выходном: 1
- Активационная функция:
 - Тип функции: Сигмоида
 - Круглизна: 1,000

График функции Сигмоида:

x	y
-10	0.0
-8	0.0
-6	0.0
-4	0.0
-2	0.0
0	0.5
2	0.8
4	0.9
6	0.95
8	0.98
10	1.0

В секции "Нейроны в слоях" необходимо указать количество скрытых слоев, т.е. слоев нейронной сети, расположенных между входным и выходным слоями. Для этого в поле "скрытых слоев" указать соответствующее значение. Это можно сделать с клавиатуры, предварительно вызвав щелчком мыши курсор, или с помощью стрелочек в правой части поля. Число нейронов во входном и выходном слоях автоматически устанавливается в соответствии с числом входных и выходных полей обучающей выборки и здесь изменить его нельзя.

Замечание: К выбору количества скрытых слоев и количества нейронов для каждого скрытого слоя нужно подходить осторожно. Хотя до сих пор не выработаны четкие критерии выбора, дать некоторые общие рекомендации все же возможно. Считается, что задачу любой сложности можно решить при помощи двухслойной нейросети, поэтому конфигурация с количеством скрытых слоев, превышающих 2, вряд ли оправдана. Для решения многих задач вполне подойдет однослойная нейронная сеть. При выборе количества нейронов следует руководствоваться следующим правилом: "количество связей между нейронами должно быть примерно на порядок меньше количества примеров в обучающем множестве". Количество связей рассчитывается как связь каждого нейрона со всеми нейронами соседних слоев, включая связи на входном и выходном слоях. Слишком большое количество нейронов может привести к так называемому "переобучению" сети, когда она выдает хорошие результаты на примерах, входящих в обучающую выборку, но практически не работает на других примерах.

В секции "Активационная функция" необходимо определить тип функции активации нейронов и ее крутизну. Для этого в списке "Тип функции" следует выбрать нужную функцию активации,



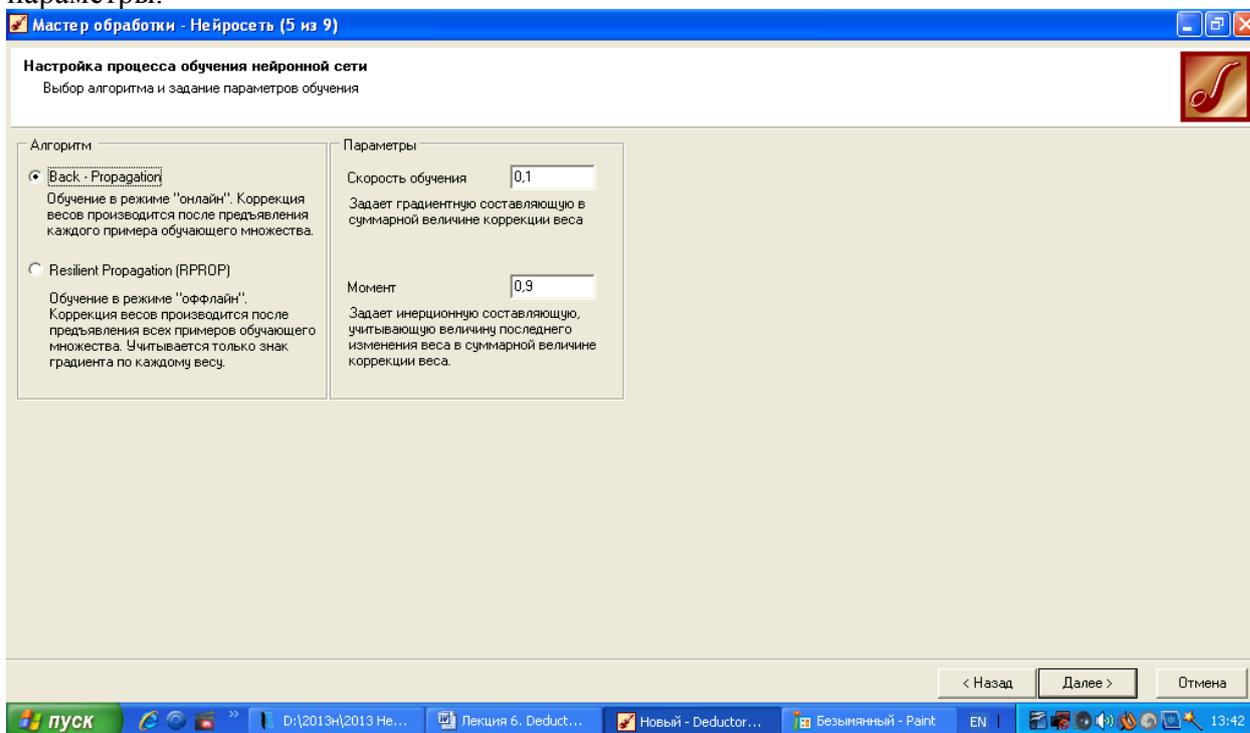
а в поле "Крутизна" - задать ее крутизну (также крутизну можно задать с помощью ползунка, расположенного ниже). В нижней части окна отображается график выбранной функции в соответствии с установленной крутизной:



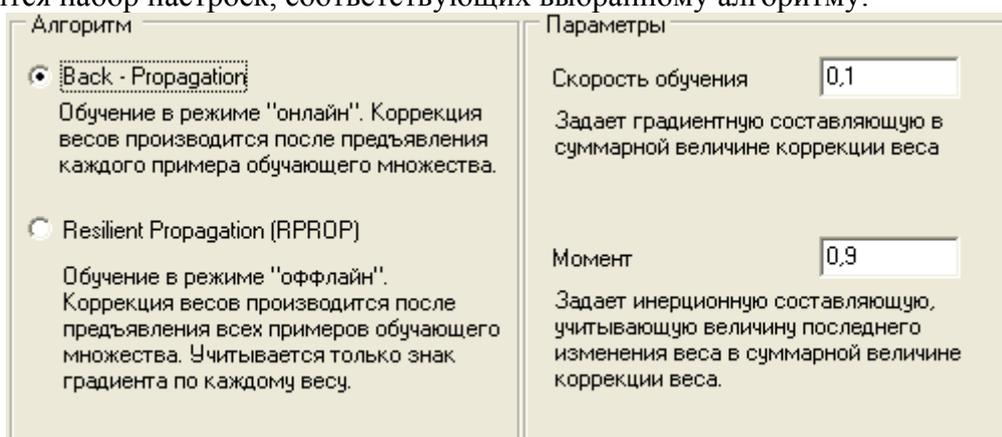


Выбор алгоритма и настройка параметров обучения

На данном шаге можно выбрать алгоритм обучения нейронной сети и задать его параметры.



Чтобы выбрать один из доступных в системе алгоритмов, нужно активизировать соответствующий пункт в секции "Алгоритм". При этом в секции "Параметры" отобразится набор настроек, соответствующих выбранному алгоритму.



Алгоритм	Параметры
<input type="radio"/> Back - Propagation Обучение в режиме "онлайн". Коррекция весов производится после предъявления каждого примера обучающего множества.	Шаг спуска <input type="text" value="0,5"/> В случае изменения знака градиентной составляющей ошибки для данного веса задает величину следующей коррекции веса.
<input checked="" type="radio"/> Resilient Propagation (RPROP) Обучение в режиме "оффлайн". Коррекция весов производится после предъявления всех примеров обучающего множества. Учитывается только знак градиента по каждому весу.	Шаг подъема <input type="text" value="1,2"/> В случае сохранения знака градиентной составляющей ошибки для данного веса задает величину следующей коррекции веса.

Для алгоритма обратного распространения ошибки (Back-Propagation) задаются два параметра:

- *Скорость обучения* - определяет величину шага при итерационной коррекции весов в нейронной сети (рекомендуется задавать в интервале 0...1).
- *Момент* - учитывает величину последнего изменения веса при коррекции весов (задается в интервале 0...1).

Для алгоритма Resilient Propagation указываются следующие параметры:

- *Шаг спуска* - коэффициент увеличения скорости обучения, который определяет шаг увеличения скорости обучения при не достижении алгоритмом оптимального результата.
- *Шаг подъема* - коэффициент уменьшения скорости обучения. Задается шаг уменьшения скорости обучения в случае пропуска алгоритмом оптимального результата.

Настройка условий остановки обучения

На данном шаге необходимо задать условия, при выполнении которых обучение будет прекращено.

Мастер обработки - Нейросеть (6 из 9)

Настройка параметров остановки обучения
 Укажите условия прекращения обучения. Обучение будет остановлено при выполнении одного из условий.

Считать пример распознанным, если ошибка меньше

По достижению эпохи

Обучающее множество

Средняя ошибка меньше

Максимальная ошибка меньше

Распознано примеров (%)

Тестовое множество

Средняя ошибка меньше

Максимальная ошибка меньше

Распознано примеров (%)

Новый - Deductor Studio Academic

< Назад Далее > Отмена

пуск D:\2013\2013 Не... Лекция 6, Deduct... Новый - Deductor... Безымянный - Paint EN | 13:48

"Считать пример распознанным, если ошибка меньше" - критерием останова в данном случае является условие, что рассогласование между эталонным и реальным выходом сети становится меньше заданного значения.

"По достижении эпохи" - установка данного флажка позволяет задать число эпох (циклов обучения) по достижении которого обучение останавливается независимо от величины ошибки. Если флажок сброшен, то обучение будет продолжаться, пока ошибка не станет меньше заданного значения.

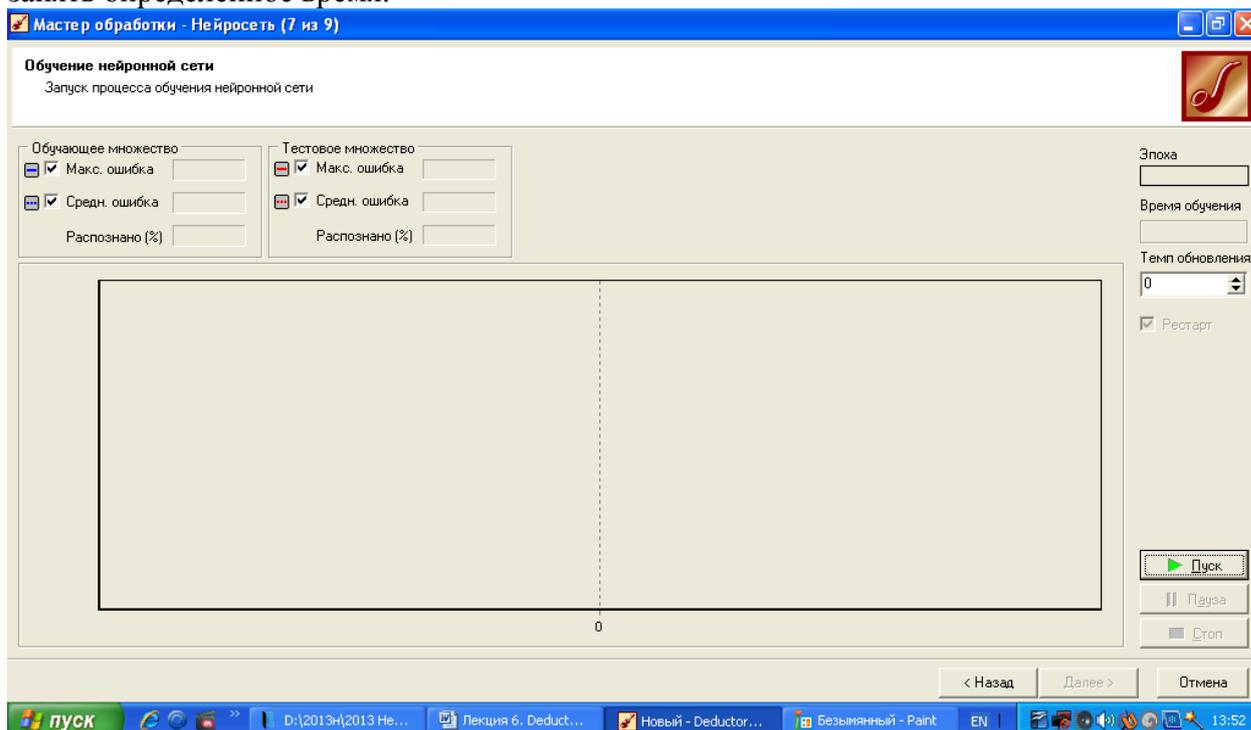
Кроме этого для обучающего и тестового множества в соответствующих секциях окна могут независимо устанавливаться следующие критерии останова обучения:

- *"Средняя ошибка меньше"* - средняя квадратичная ошибка на обучающем множестве или тестовом множестве меньше заданного значения.
- *"Максимальная ошибка меньше"* - максимальная квадратичная ошибка на обучающем множестве или тестовом множестве меньше заданного значения.
- *"Распознано примеров (%)"* - количество распознанных примеров на обучающем множестве или тестовом множестве больше заданного процента.

При выборе нескольких условий останова процесса обучения происходит по достижении хотя бы одного из них.

Запуск процесса обучения

На данном шаге производится собственно процесс обучения нейронной сети. В зависимости от объема обрабатываемых данных и быстродействия компьютера, он может занять определенное время.



Остановить процесс обучения можно с помощью кнопки "Стоп". После этого появляется возможность вернуться на предыдущие шаги Мастера обработки для проверки и изменения параметров. Далее процесс обучения может быть запущен заново, продолжен или отменен. Если обучение было завершено успешно, то в секции "Название текущего процесса" появится сообщение "Успешное завершение".

Для управления процессом обучения предусмотрены следующие кнопки:

- "Пуск" - запускает процесс в первый раз или возобновляет после паузы.
- "Пауза" - временно приостанавливает обработку. Временная приостановка процесса имеет смысл в двух случаях:

- для оценки текущих результатов процесса обучения, например, просмотра графиков динамики ошибок обучения.
- при необходимости освободить ресурсы процессора для других приложений.
- "Стоп" - останавливает процесс без возможности его продолжения. Практически, останавливать процесс обучения модели принудительно имеет смысл если:
 - значения ошибок длительное время не уменьшаются.
 - процент распознанных примеров не увеличивается.

В процессе обучения в секциях "Обучающее множество" и "Тестовое множество" отображаются максимальная квадратичная ошибка и средняя квадратичная ошибка на обучающем множестве и тестовом множестве соответственно, а также процент распознанных примеров.

Обучение может с большой долей вероятности считаться успешным, если процент распознанных примеров на обучающем и тестовом множествах достаточно велик (близок к 100%).

В процессе обучения в окне отображаются следующие графики хода обучения для обучающего (синяя линия) и тестового (красная линия) множеств:

- Максимальная квадратичная ошибка на обучающем множестве и тестовом множестве (сплошная линия).
- Средняя квадратичная ошибка на обучающем множестве и тестовом множестве (пунктирная линия).

Любой из графиков можно скрыть (показать). Для этого достаточно сбросить (установить) соответствующий флажок.

Если какой-либо участок графика требуется рассмотреть более подробно, можно увеличить масштаб просмотра данного участка. Для этого следует выделить прямоугольную область с нужным участком, двигая указатель с нажатой левой кнопкой мыши от левого верхнего к правому нижнему углу прямоугольника. Повторение данного действия приведет к дальнейшему увеличению масштаба. Чтобы вернуть исходный масштаб графика, достаточно выделить любой участок графика, двигая при этом мышью с нажатой левой кнопкой от правого нижнего угла прямоугольника к левому верхнему.

В правой части окна постоянно отображаются номер текущей эпохи и время, прошедшее с начала обучения.

В поле "Темп обновления" можно задать число эпох обучения сети, через которое будет происходить обновление графика.

Флажок "Рестарт" позволяет включить режим инициализации начальных весов сети случайными значениями. Если флажок сброшен, то при повторном запуске обучения после остановки будет иметь место так называемое "дообучение сети", когда обучение будет начато с текущими весами.

Мастер обработки - Нейросеть (7 из 9)

Обучение нейронной сети
Запуск процесса обучения нейронной сети

Обучающее множество: Макс. ошибка: 6,14E-02; Средн. ошибка: 5,20E-03; Распознано (%): 98,59
Тестовое множество: Макс. ошибка: 4,36E-02; Средн. ошибка: 7,39E-03; Распознано (%): 100,00

Эпоха: 10000
Время обучения: 00:00:13
Темп обновления: 0
 Рестарт

Пуск, Пауза, Стоп

Подключение по локальной сети
Сетевой кабель не подключен

На красной линии видно, что в районе 300 эпохи ошибка сети перестаёт уменьшаться. Вернёмся назад и уменьшим количество эпох:

Мастер обработки - Нейросеть (6 из 9)

Настройка параметров остановки обучения
Укажите условия прекращения обучения. Обучение будет остановлено при выполнении одного из условий.

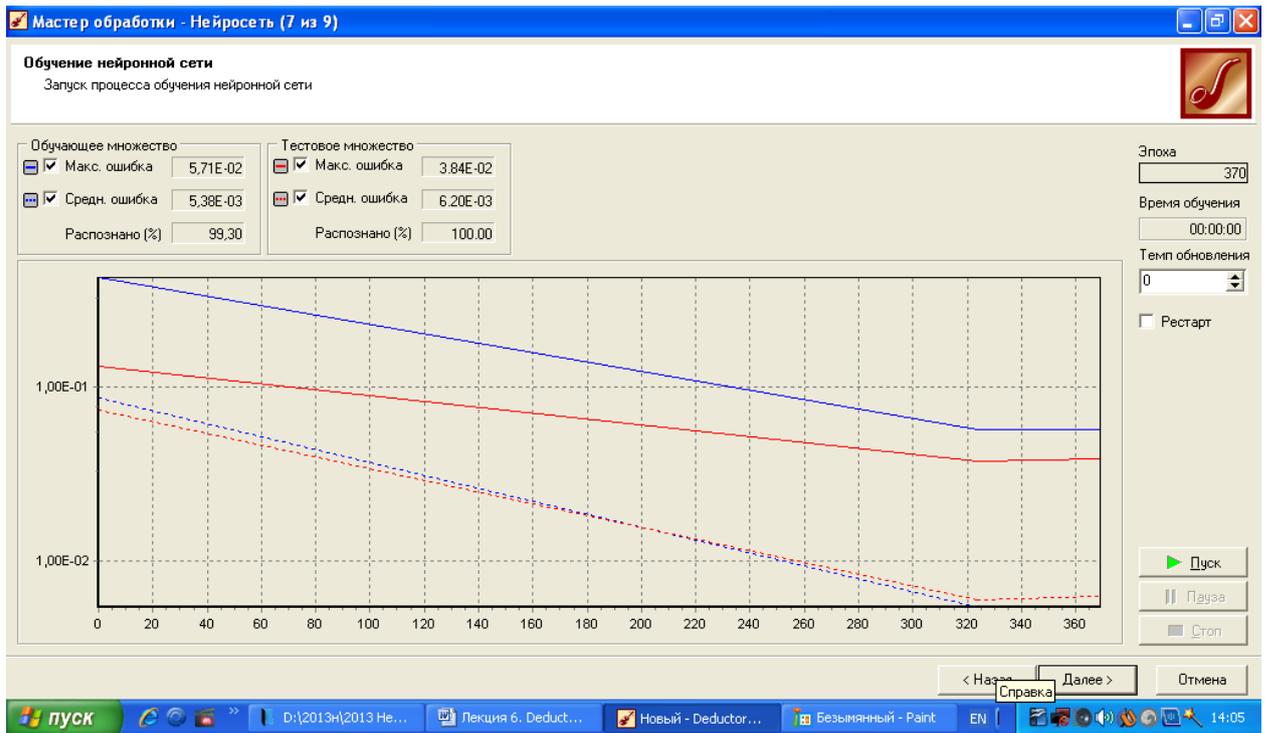
Считать пример распознанным, если ошибка меньше: 0,05
 По достижению эпохи: 370

Обучающее множество:
 Средняя ошибка меньше
 Максимальная ошибка меньше
 Распознано примеров (%): 0

Тестовое множество:
 Средняя ошибка меньше
 Максимальная ошибка меньше
 Распознано примеров (%): 0

Назад, Далее >, Отмена

Проведём новое обучение (с рестартом):

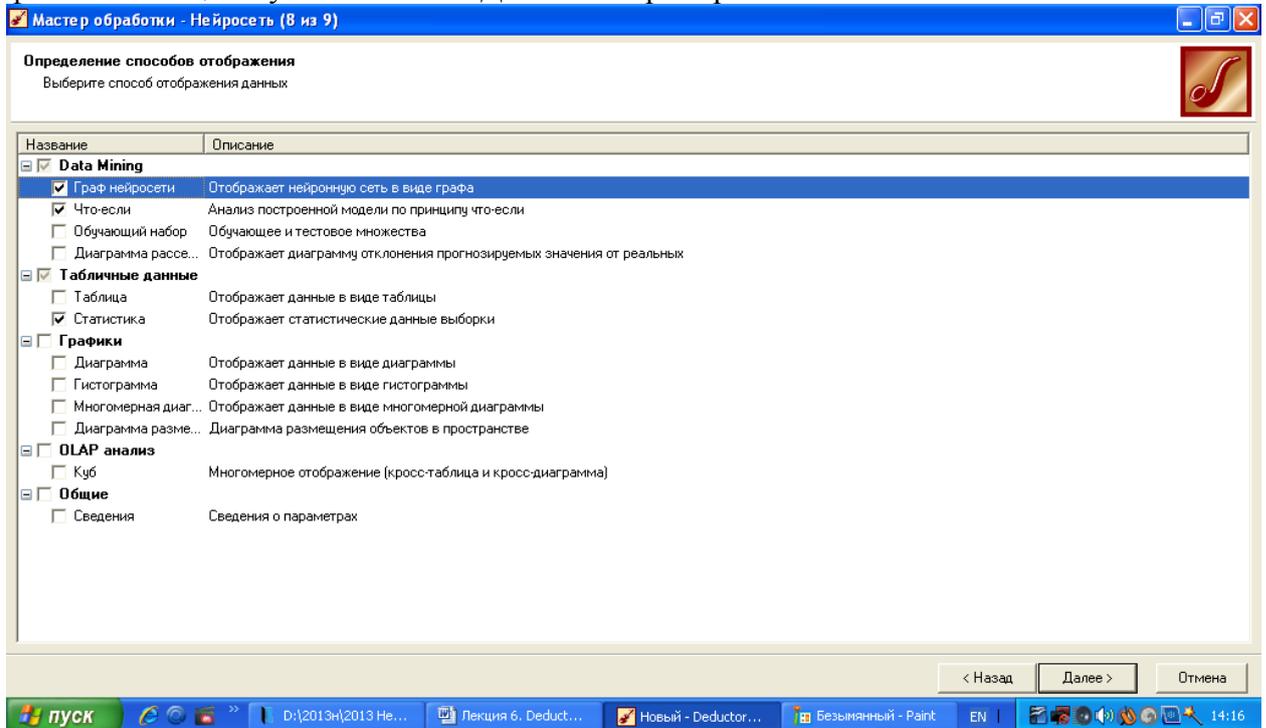


Примерно в этом месте достигнуто оптимальное обучение сети.

Визуализация 2.

Выбор способа отображения

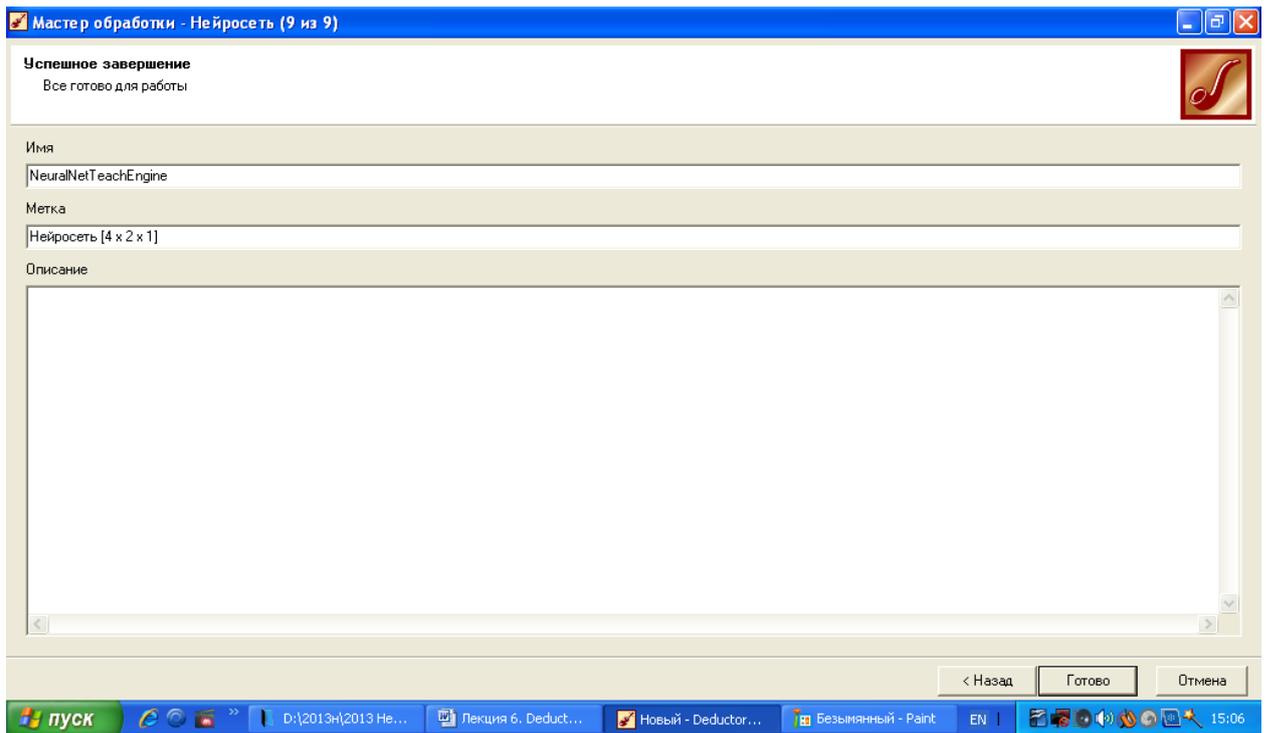
На данном шаге пользователь должен выбрать, в каком виде будут отображены результаты обработки данных. Для этого достаточно пометить нужные виды отображения флажками и щелкнуть по кнопке "Далее". Например:



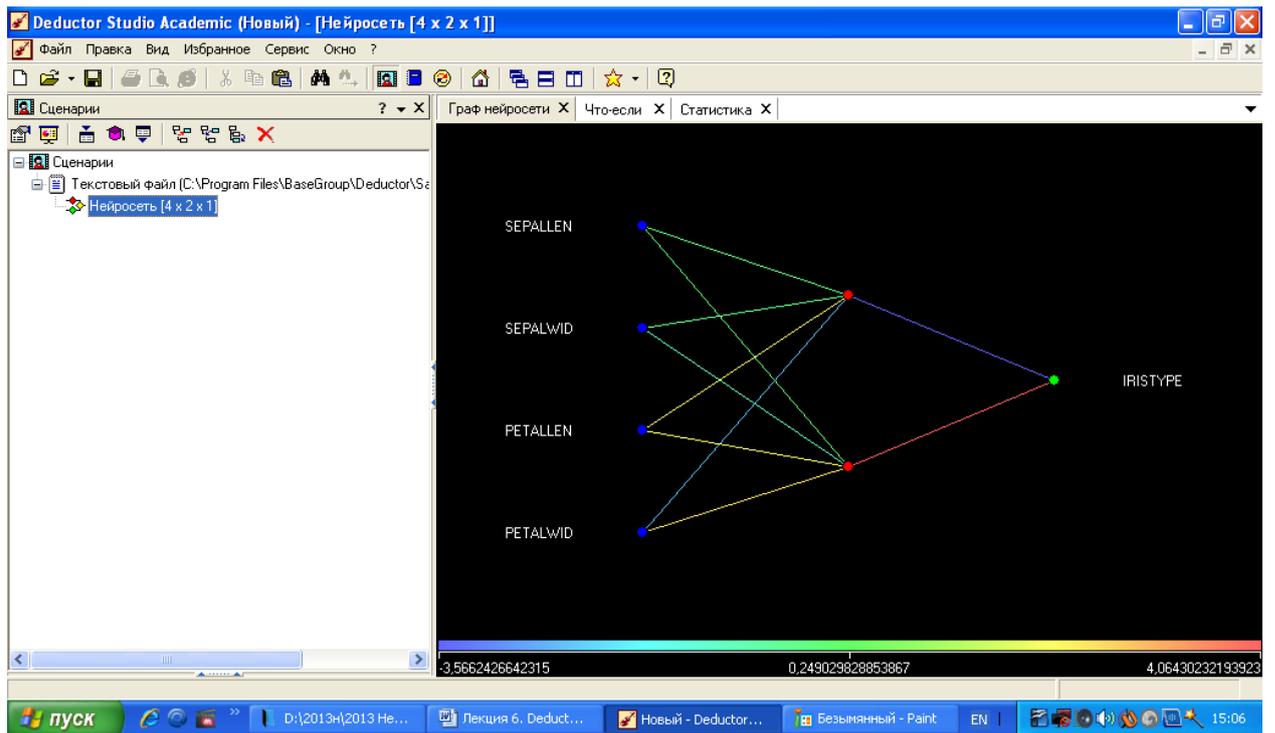
Для выборки данных, полученных в результате обработки данных с помощью нейронной сети, доступны следующие виды отображения:

- Обучающий набор;

- Диаграмма рассеяния;
- Граф нейросети;
- Что-если;
- Таблица сопряженности.
- Таблица;
- Статистика;
- Диаграмма;
- Гистограмма;
- Куб;
- Сведения.



Граф нейросети.

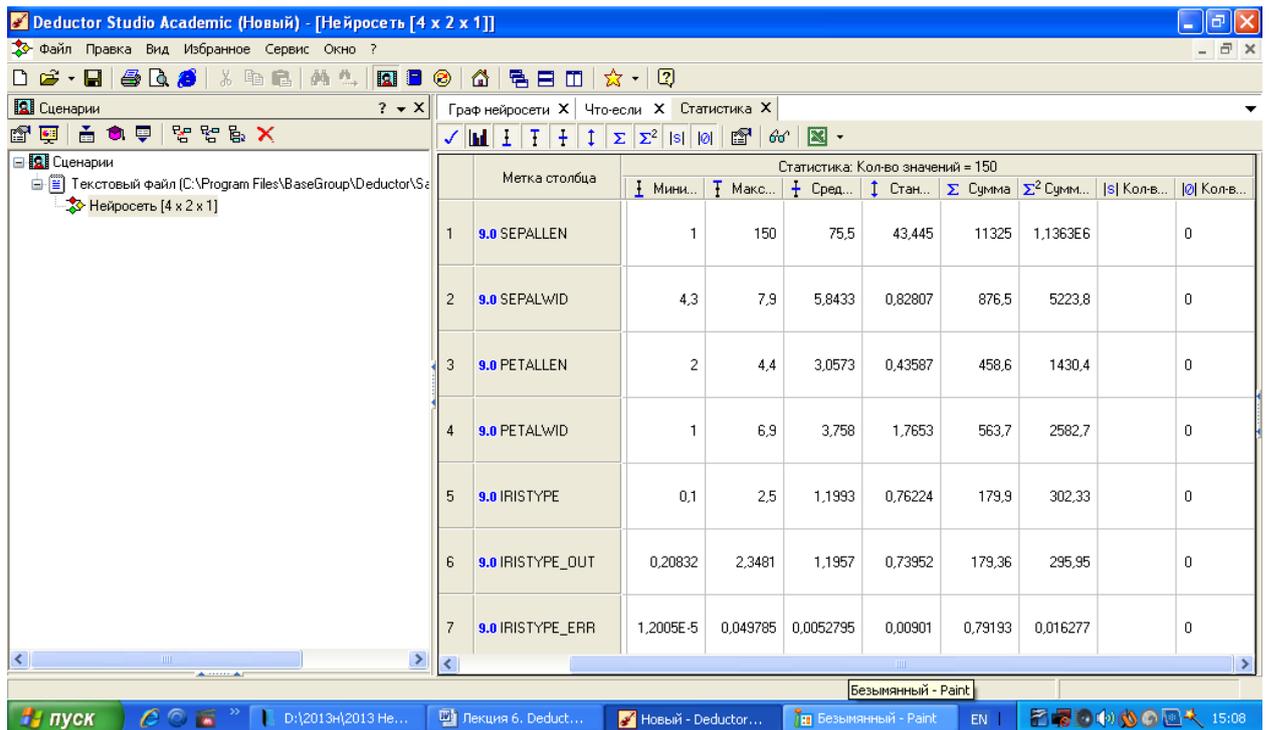
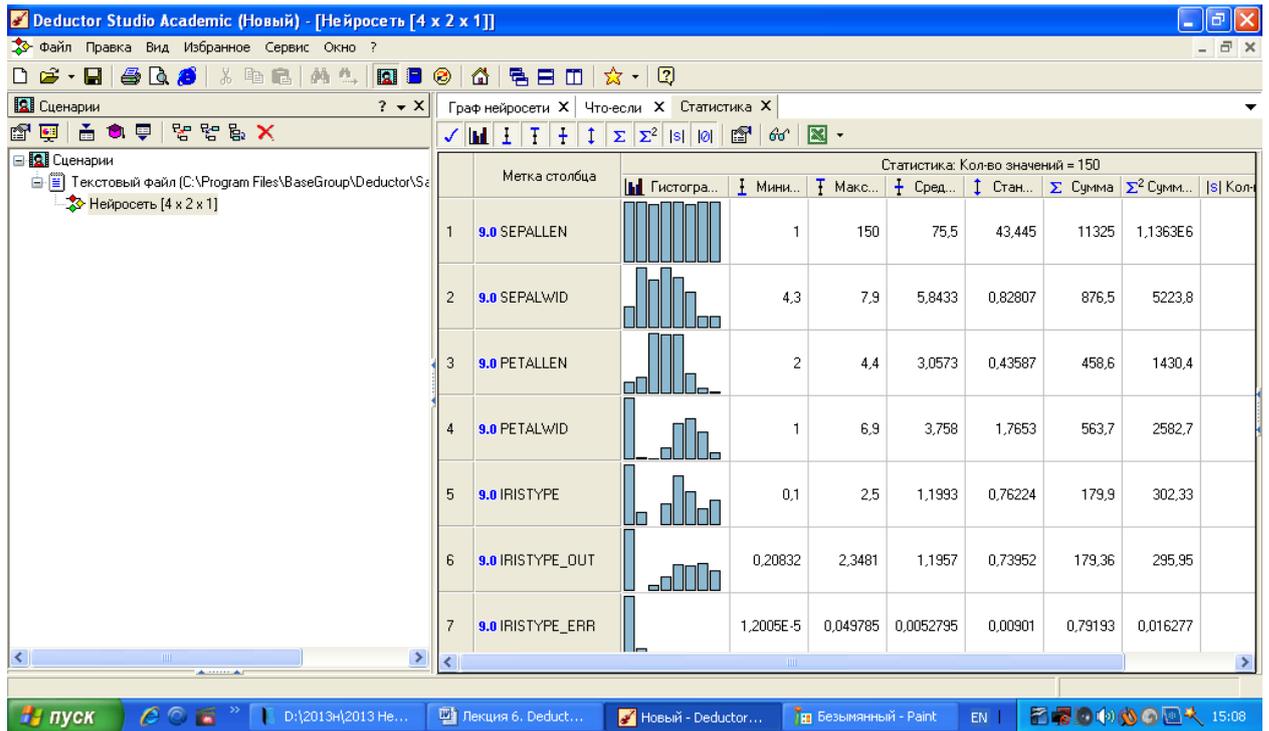


Что – если.

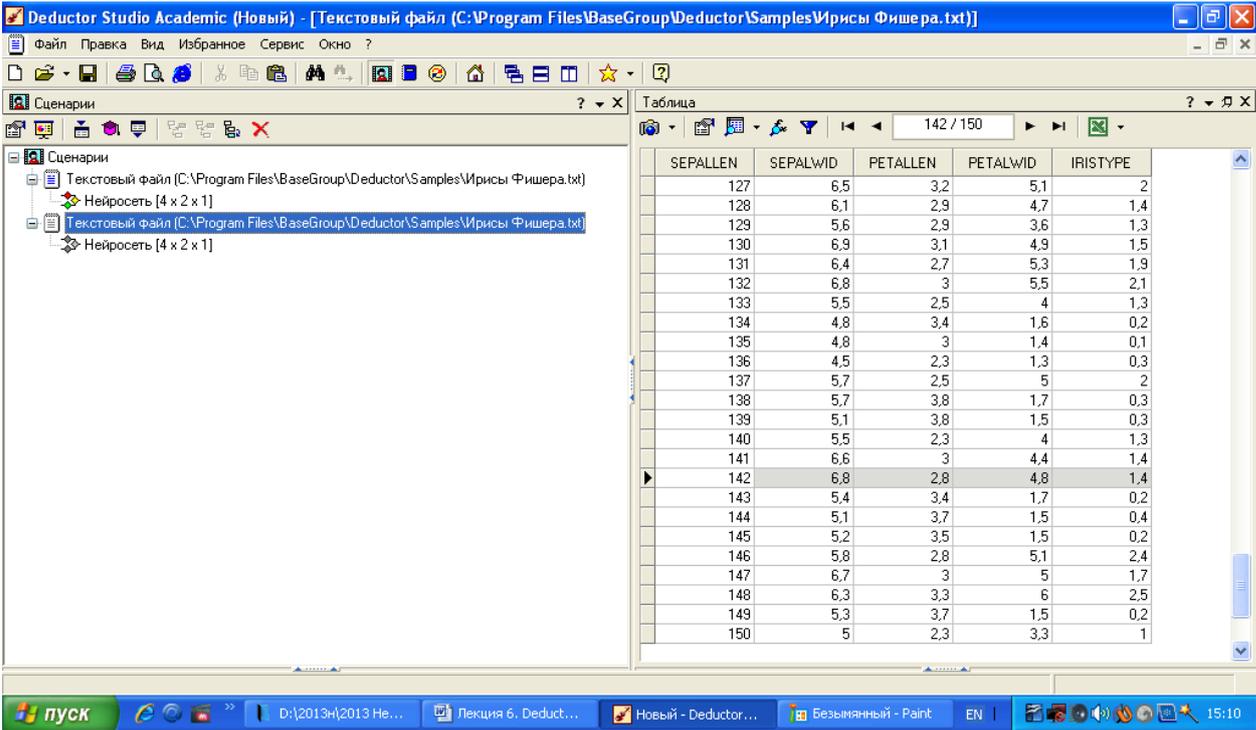
The screenshot shows the 'Что-если' (What-if) analysis window. It displays a table with input and output values for the neural network model.

Поле	Значение
Входные	
9.0 SEPALLEN	1
9.0 SEPALWID	5
9.0 PETALLEN	3,3
9.0 PETALWID	1,4
Выходные	
9.0 IRISTYPE	0,219827537219992

Статистика.



Текстовый файл.



The screenshot shows the Deductor Studio Academic interface. The main window displays a table with 150 rows of data. The columns are labeled SEPALLEN, SEPALWID, PETALLEN, PETALWID, and IRISTYPE. The data represents the Iris dataset, with each row corresponding to a specific flower specimen. The table is currently showing rows 142 through 150.

SEPALLEN	SEPALWID	PETALLEN	PETALWID	IRISTYPE
127	6,5	3,2	5,1	2
128	6,1	2,9	4,7	1,4
129	5,6	2,9	3,6	1,3
130	6,9	3,1	4,9	1,5
131	6,4	2,7	5,3	1,9
132	6,8	3	5,5	2,1
133	5,5	2,5	4	1,3
134	4,8	3,4	1,6	0,2
135	4,8	3	1,4	0,1
136	4,5	2,3	1,3	0,3
137	5,7	2,5	5	2
138	5,7	3,8	1,7	0,3
139	5,1	3,8	1,5	0,3
140	5,5	2,3	4	1,3
141	6,6	3	4,4	1,4
142	6,8	2,8	4,8	1,4
143	5,4	3,4	1,7	0,2
144	5,1	3,7	1,5	0,4
145	5,2	3,5	1,5	0,2
146	5,8	2,8	5,1	2,4
147	6,7	3	5	1,7
148	6,3	3,3	6	2,5
149	5,3	3,7	1,5	0,2
150	5	2,3	3,3	1

Лекция 6. Самоорганизующиеся карты Кохонена

Количество нейронов – это количество переменных в таблице исходных данных.

Объём обучающей выборки – это количество строк в таблице, предъявляемых для обучения. По-другому это называется количеством реализаций.

На карту Кохонена наносятся реализации в виде шестиугольных элементов, причём, расположение их учитывает их степень сходства.

Нейронные сети могут обучаться с учителем или без него.

При **обучении с учителем** для каждого обучающего входного примера требуется знание правильного ответа или функции оценки качества ответа.

Такое обучение называют управляемым.

Нейронной сети предъявляются значения входных и выходных сигналов, а она по определенному алгоритму подстраивает веса синаптических связей.

В процессе обучения производится *корректировка весов* сети по результатам сравнения фактических выходных значений с входными, известными заранее.

При **обучении без учителя** раскрывается внутренняя структура данных или корреляции между образцами в наборе данных.

Выходы нейронной сети формируются самостоятельно, а веса изменяются по алгоритму, учитывающему только входные и производные от них сигналы.

Это обучение называют также неуправляемым.

В результате такого обучения объекты или примеры распределяются по категориям, сами категории и их количество могут быть заранее неизвестны.

Сети, называемые картами Кохонена, - это одна из разновидностей нейронных сетей, однако они принципиально отличаются от рассмотренных ранее перцептронов, поскольку используют неконтролируемое обучение.

При таком обучении обучающее множество состоит лишь из значений входных переменных, в процессе обучения нет сравнения выходов нейронов с эталонными значениями.

Можно сказать, что такая сеть учится понимать структуру данных.

Идея сети Кохонена принадлежит финскому ученому Тойво Кохонену (1982 год).

Основной принцип работы таких сетей - введение в правило обучения нейрона информации относительно его расположения.

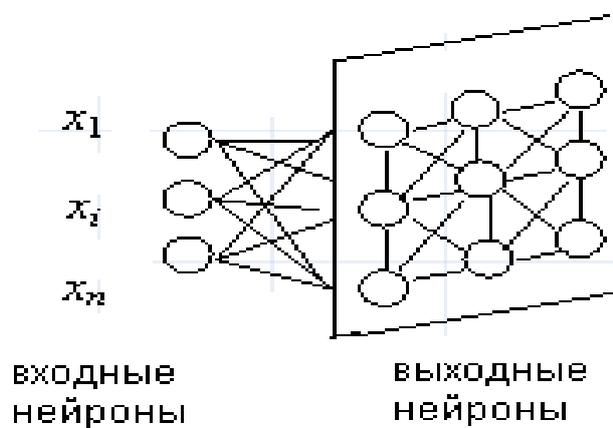
При такой постановке задачи дан набор объектов, каждому из которых сопоставлена строка таблицы (вектор значений признаков).

Требуется разбить исходное множество на классы, т.е. для каждого объекта найти класс, к которому он принадлежит.

Сеть Кохонена, в отличие от *многослойной нейронной сети*, очень проста; она представляет собой два слоя: входной и выходной.

Ее также называют *самоорганизующей картой*.

Элементы карты располагаются в некотором пространстве, как правило, двумерном.



Сеть Кохонена обучается методом последовательных приближений.

В процессе обучения таких сетей на входы подаются данные, но сеть при этом подстраивается не под эталонное значение выхода, а под закономерности во входных данных.

Начинается обучение с выбранного случайным образом выходного расположения центров.

В процессе последовательной подачи на вход сети обучающих примеров определяется нейрон, у которого скалярное произведение весов и поданного на вход вектора минимально. Этот нейрон объявляется победителем и является центром при подстройке весов у соседних нейронов.

Такое правило обучения предполагает "соревновательное" обучение с учетом расстояния нейронов от "нейрона-победителя".

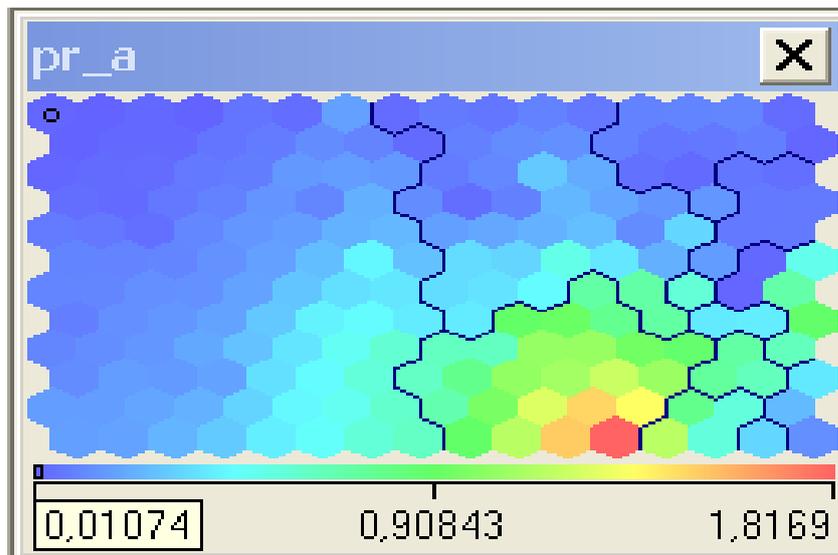
Обучение при этом заключается не в минимизации ошибки, а в подстройке весов (*внутренних параметров* нейронной сети) для наибольшего совпадения с входными данными.

Полученную карту можно использовать как средство визуализации при анализе данных.

В результате обучения карта Кохонена классифицирует входные примеры на кластеры (группы схожих примеров) и визуально отображает многомерные входные данные на плоскости нейронов.

Нейроны карты Кохонена располагают в виде двухмерной матрицы, раскрашивают эту матрицу в зависимости от анализируемых параметров нейронов.

Пример карты Кохонена:



Группа объектов, обозначенная красным цветом, имеет наибольшие значения рассматриваемого показателя, а группа объектов, обозначенная синим цветом - наименьшие.

Перед началом обучения карты необходимо проинициализировать весовые коэффициенты нейронов. Удачно выбранный способ инициализации может существенно ускорить обучение и привести к получению более качественных результатов.

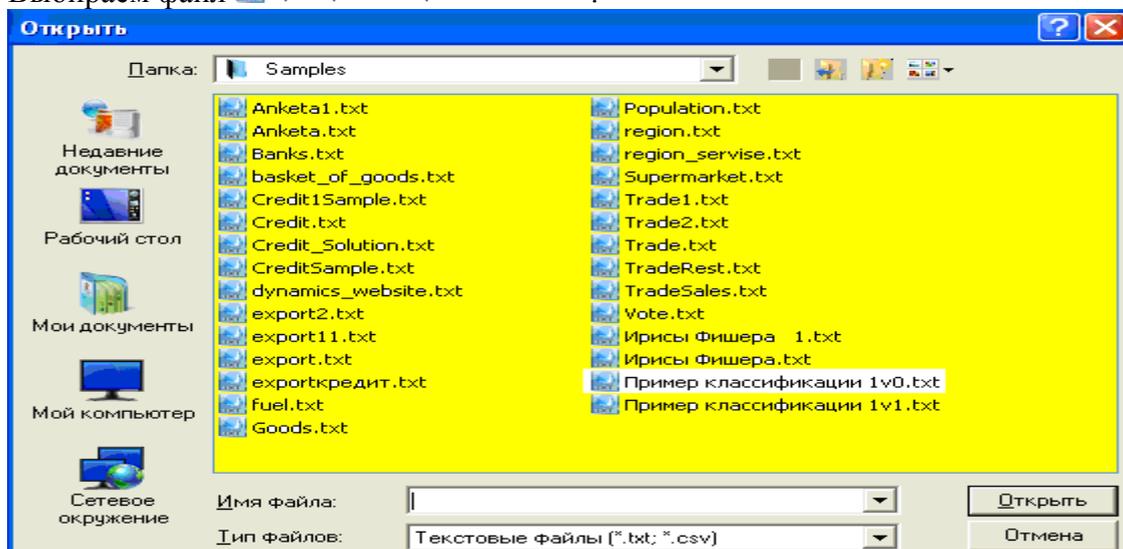
Процесс построения и обучения самоорганизующейся карты содержит следующие этапы:

1. Выбор назначений полей.
2. Настройка нормализации полей.
3. Настройка обучающей выборки.
4. Настройка параметров обучения.
5. Настройка условий остановки обучения.
6. Запуск процесса обучения.
7. Выбор способа отображения данных.

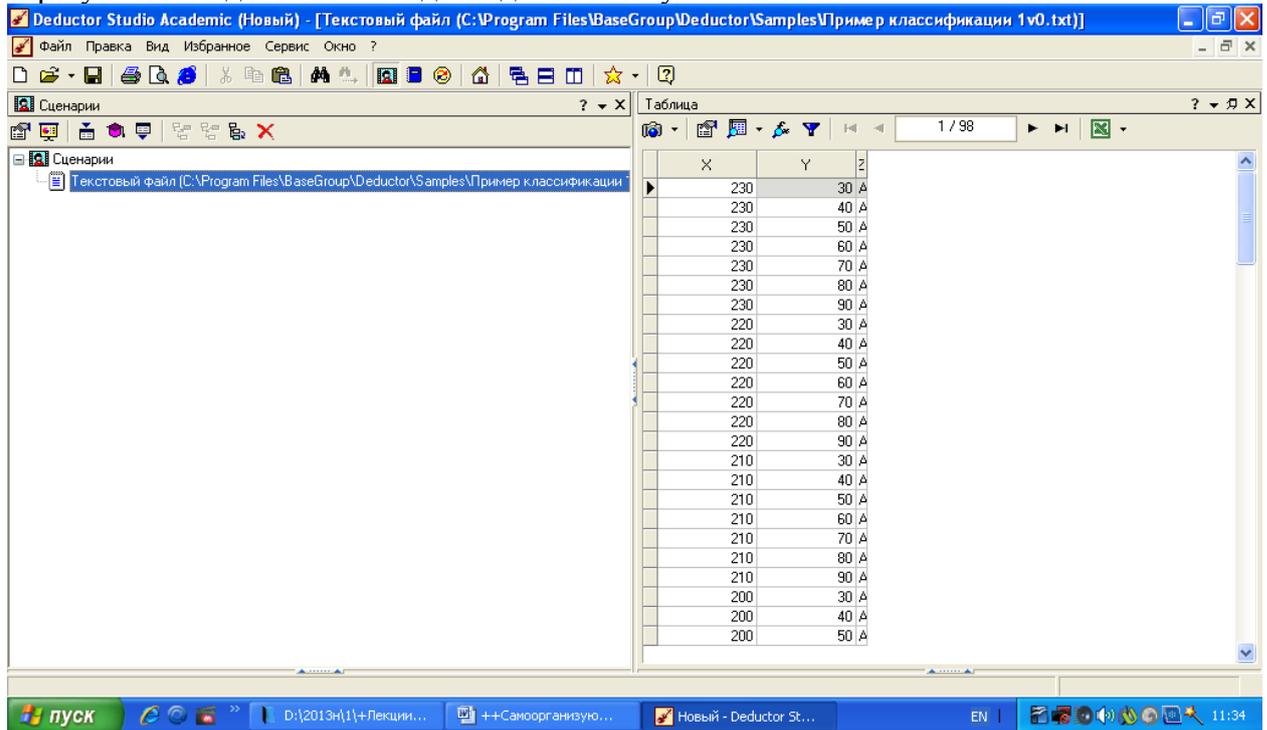
Настройка назначений полей, нормализации, и настройка обучающей выборки.

Настройка назначений полей, нормализации и обучающей выборки были рассмотрены в разделе «Мастер обработки нейронной сети».

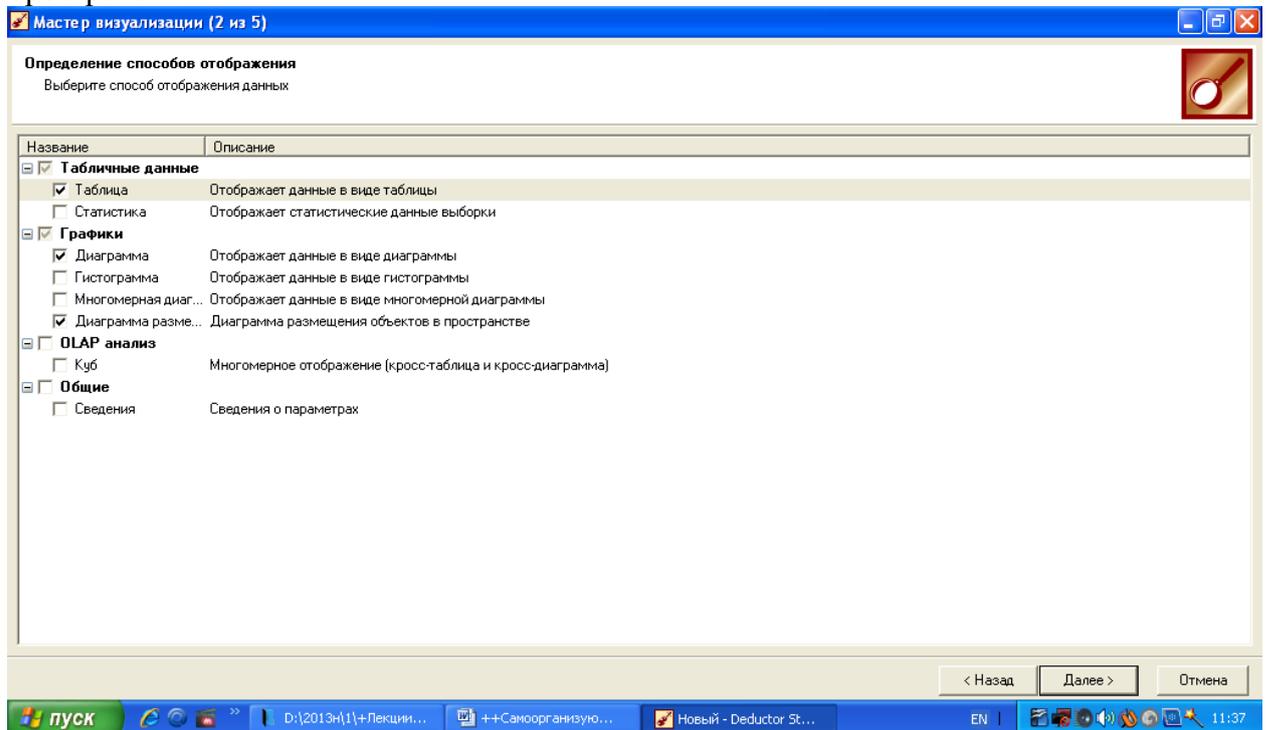
Выбираем файл Пример классификации 1v0.txt :



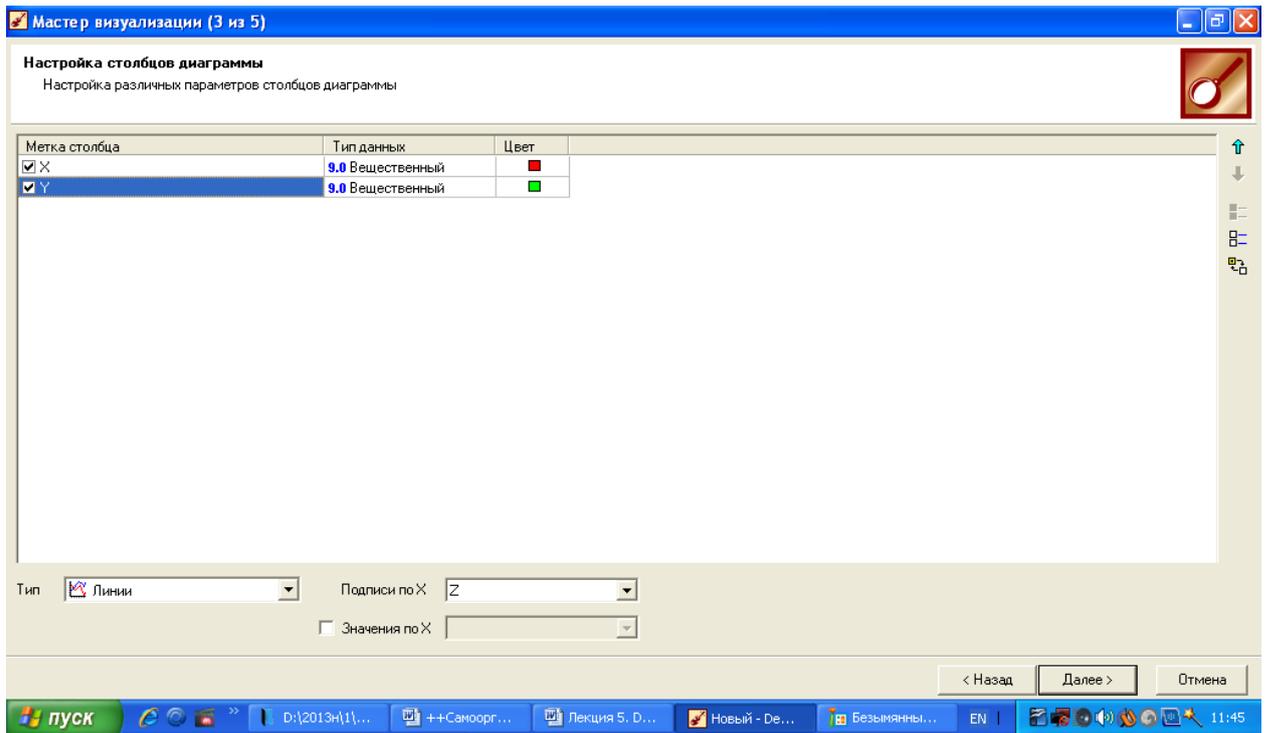
В результате подготовки исходных данных получим:



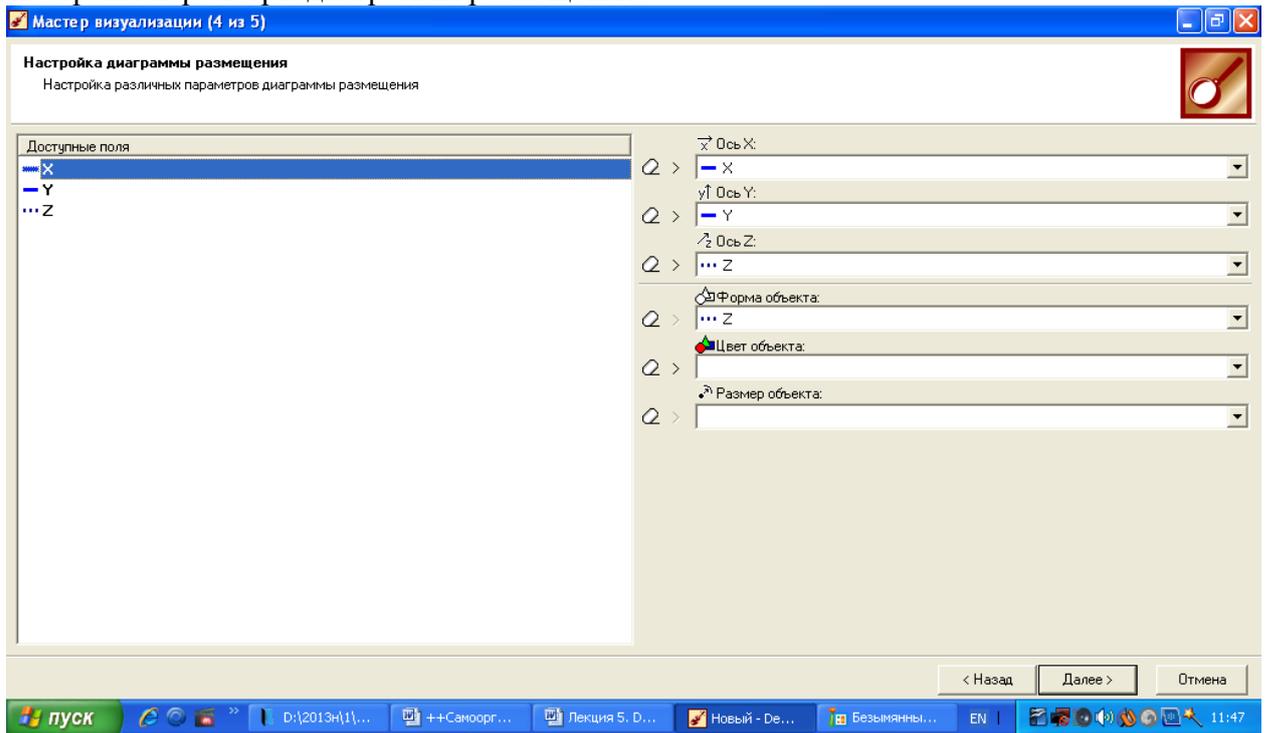
Для того, чтобы узнать, что представляет собой исходный набор данных, запросим через мастер визуализации () простую диаграмму и диаграмму размещения объектов в пространстве.



Произведём настройку столбцов:
Определим тип данных переменных.



Настроим параметры диаграммы размещения:



При готовности сообщается место нахождения файла:

Мастер визуализации (5 из 5)

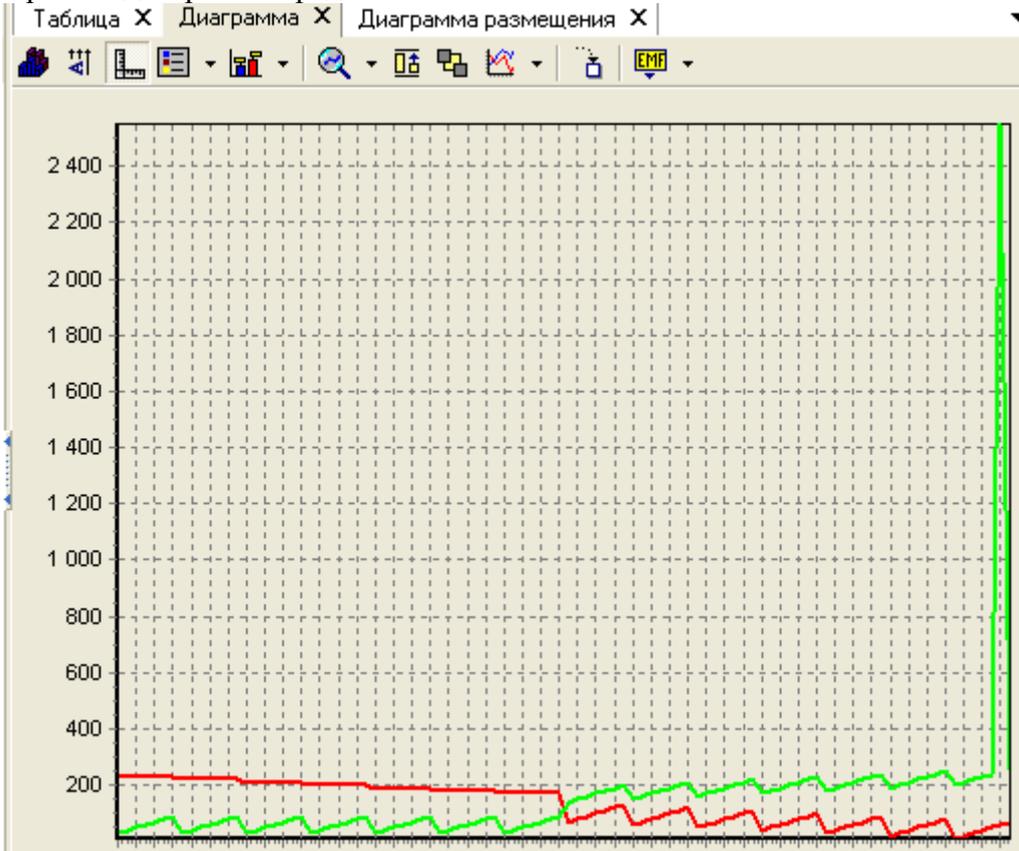
Успешное завершение
Все готово для работы

Имя
TextFile

Метка
Текстовый файл (C:\Program Files\BaseGroup\Deductor\Samples\Пример классификации 1v0.txt)

Описание

Простая диаграмма представляет собой:



Необходимо объяснить странный всплеск зелёной линии справа – резко откланившаяся точка, отнесённая к классу В (предпоследняя точка таблицы). Для этого смотрим исходный файл 1v0.txt. В его конце находится точка с резким отклонением от остальных значений – $x=60$; $Y=2550$:

10	200 В
20	210 В
30	220 В
40	230 В
50	240 В
60	2550 В
70	260 В

Диаграмма размещения X представляет собой:

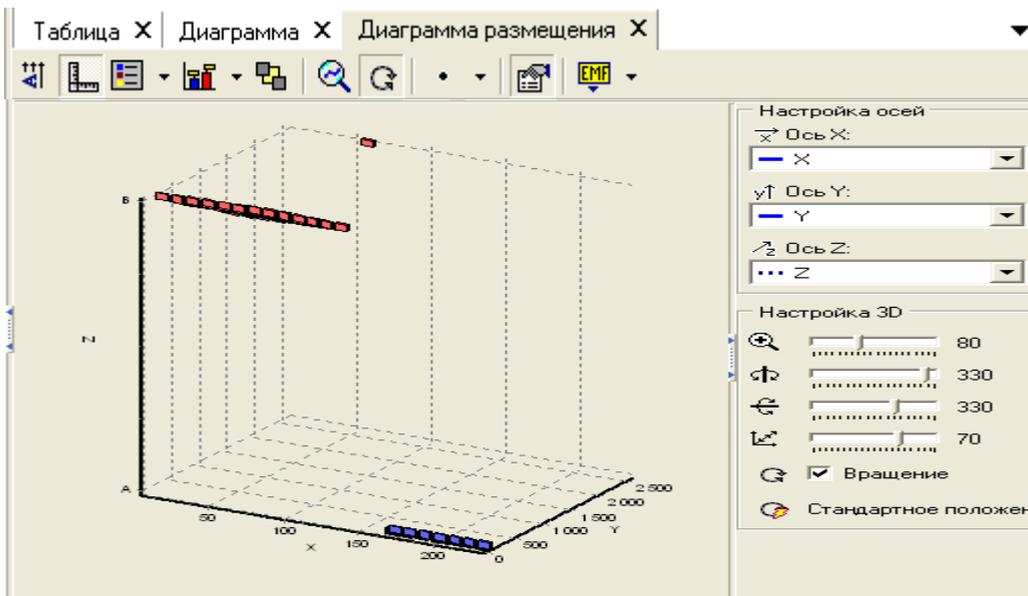
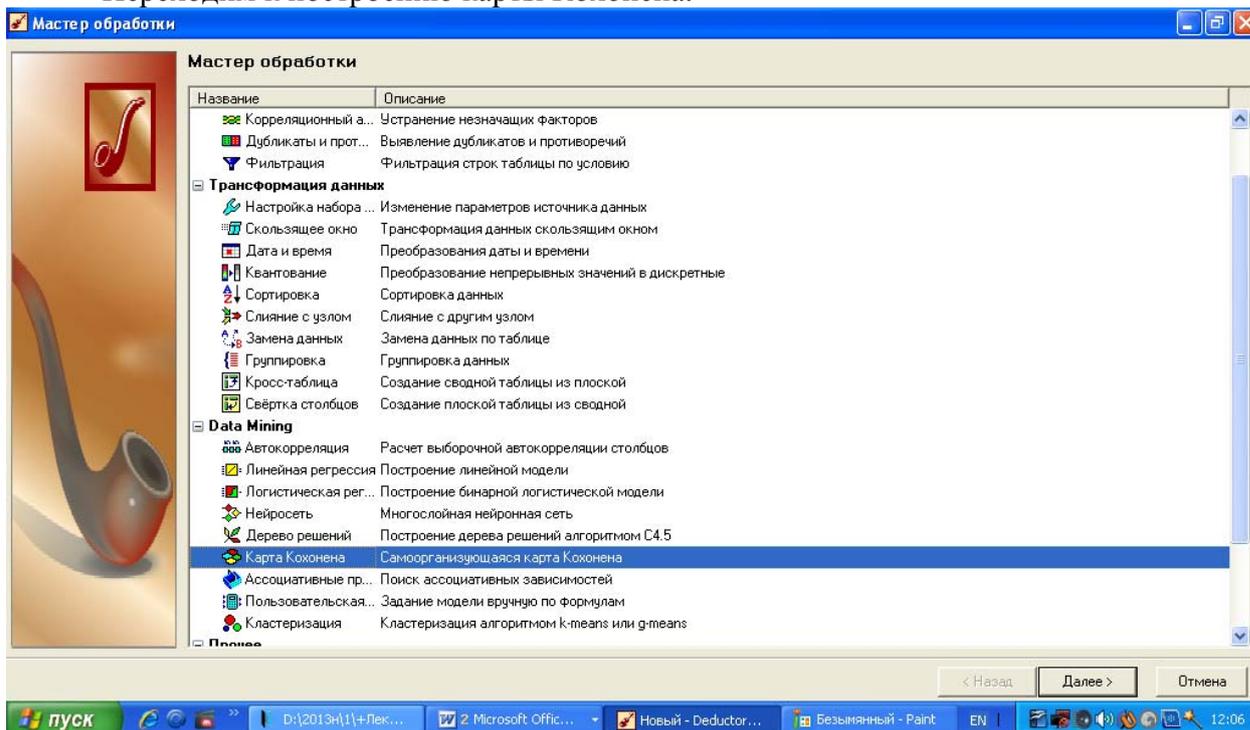


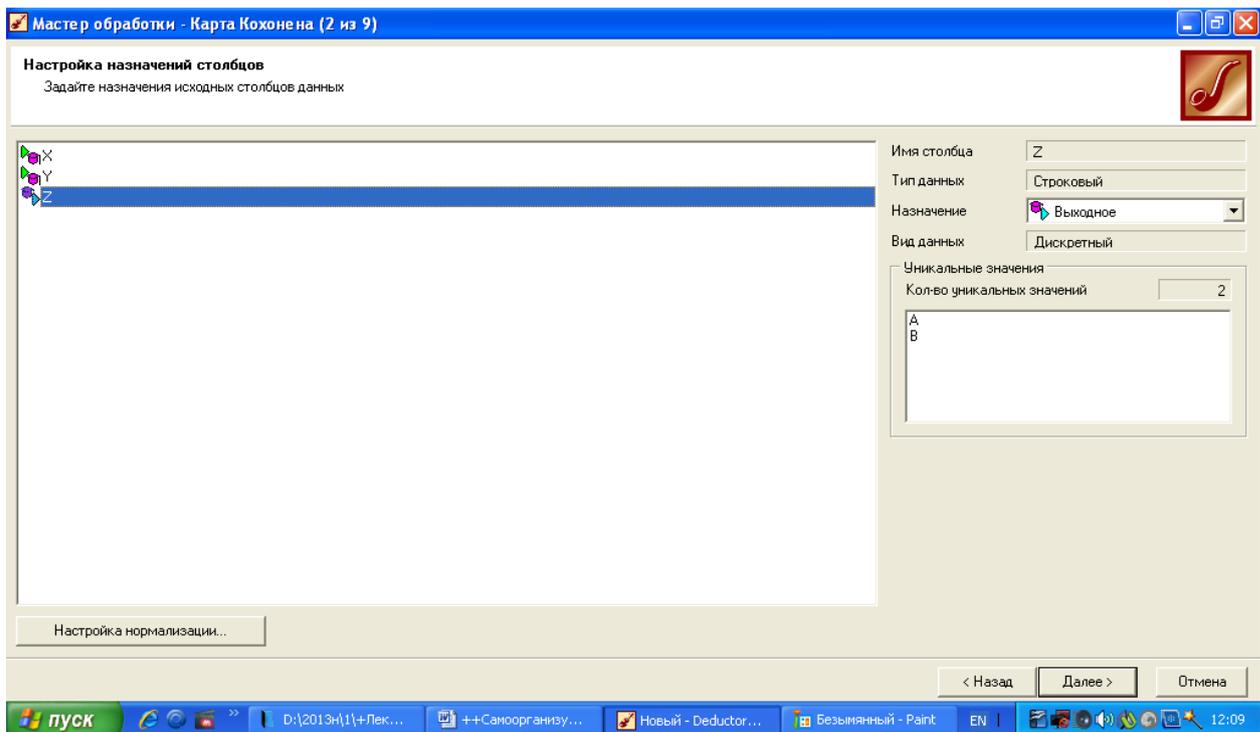
Диаграмма размещения показывает, что объекты размещены в трёх различных зонах пространства.

Настройка параметров карты Кохонена

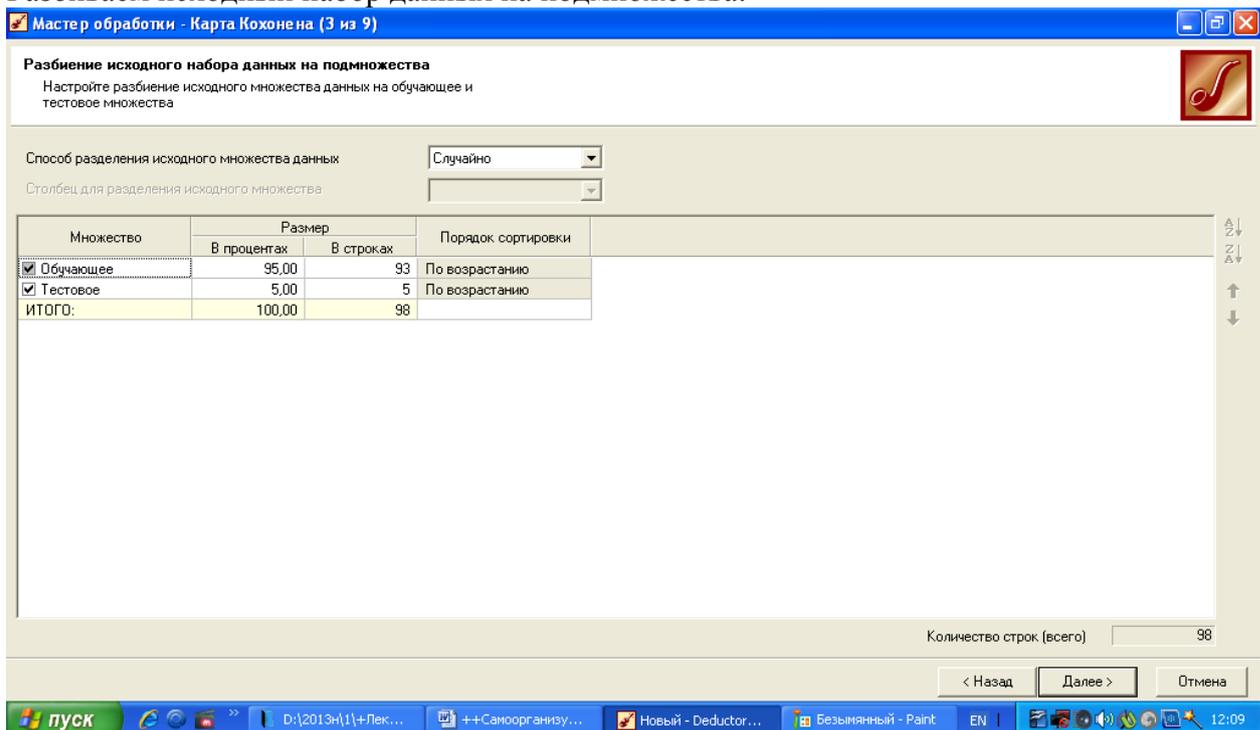
Переходим к построению карты Кохонена.



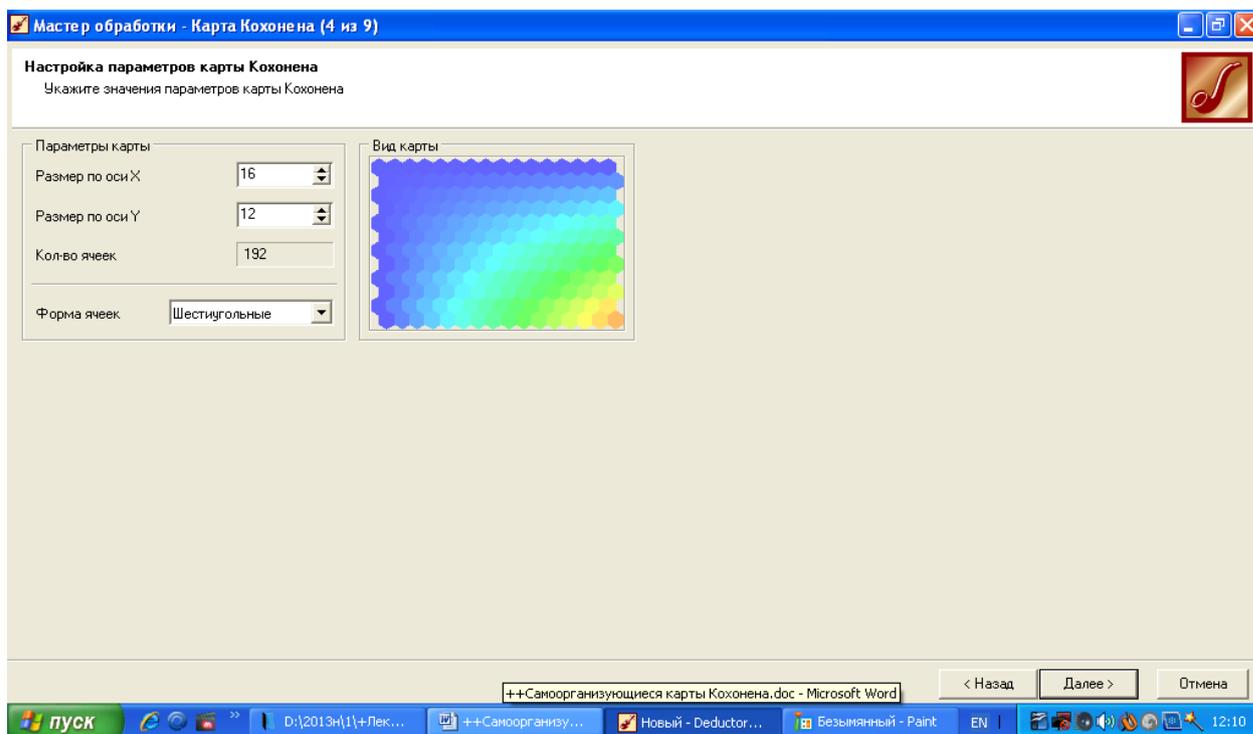
Настраиваем назначения столбцов:



Разбиваем исходный набор данных на подмножества:



В секции "Параметры карты" задается размер карты, т.е. количество ячеек из которых она будет состоять. Для этого в полях "Размер по оси X" и "Размер по оси Y" следует указать количество ячеек по соответствующим координатам.

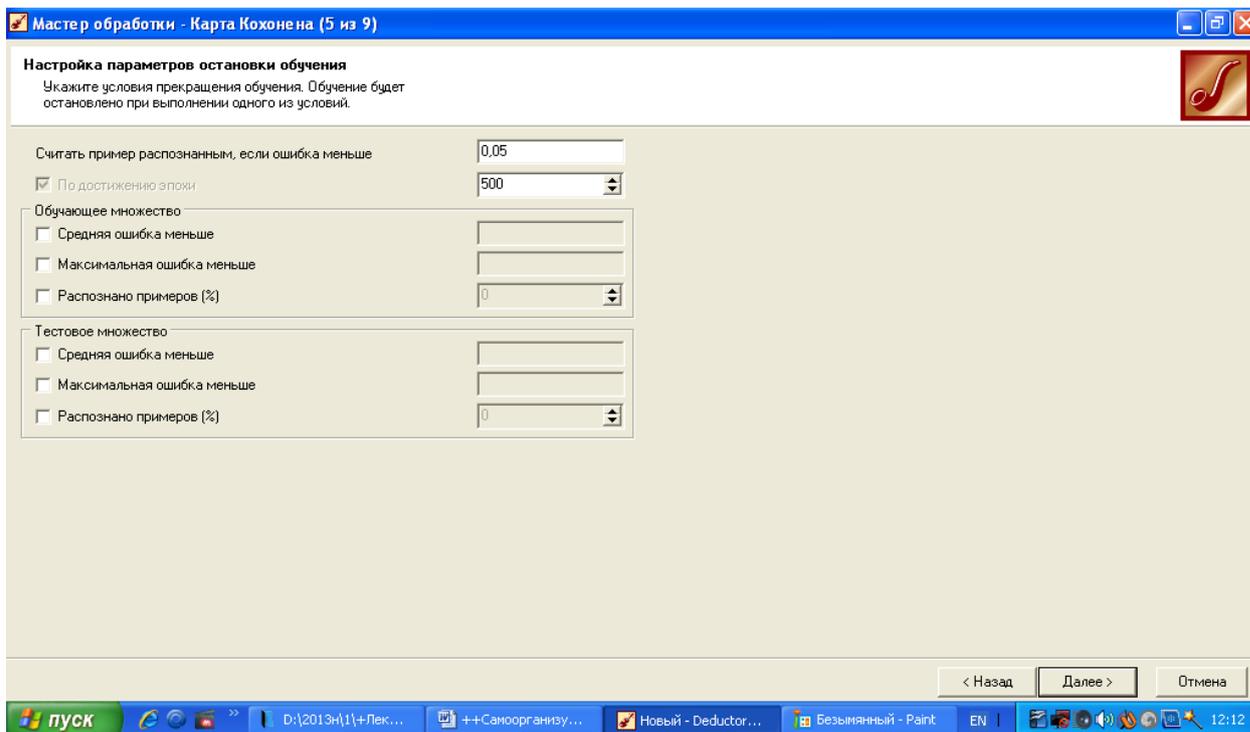


В поле "Количество ячеек" отображается общее число ячеек карты. Оно определяется как произведение значений полей "Размер по оси X" и "Размер по оси Y" и меняется только при их изменении.

В списке "Форма ячеек" выбирается один из вариантов конфигурации ячейки - прямоугольная или шестиугольная. При задании формы ячеек нужно учитывать, что шестиугольники дают более корректные результаты, т.к. расстояние между центрами ячеек ближе к евклидову, чем между центрами прямоугольников. Скорость обучения выше для прямоугольной формы ячеек.

Настройка параметров остановки обучения

На данном шаге необходимо задать условие, при выполнении которого обучение карты будет прекращено.



Под ошибкой сети понимается расстояние выбранного нейрона от «нейрона – победителя», т.е. нейрона, у которого скалярное произведение веса на вектор минимально. В процессе обучения вместе с весом нейрона изменяется и его положение в слое Кохонена.

"Считать пример распознанным, если ошибка меньше" - критерием останова в данном случае является условие, что рассогласование между эталонным и реальным выходом карты становится меньше заданного значения.

"По достижению эпохи" - указывается количество эпох, по достижении которого процесс обучения будет остановлен, даже, если не достигнута заданная ошибка. Позволяет избежать "зацикливания" в ситуациях, когда ошибка не достижима.

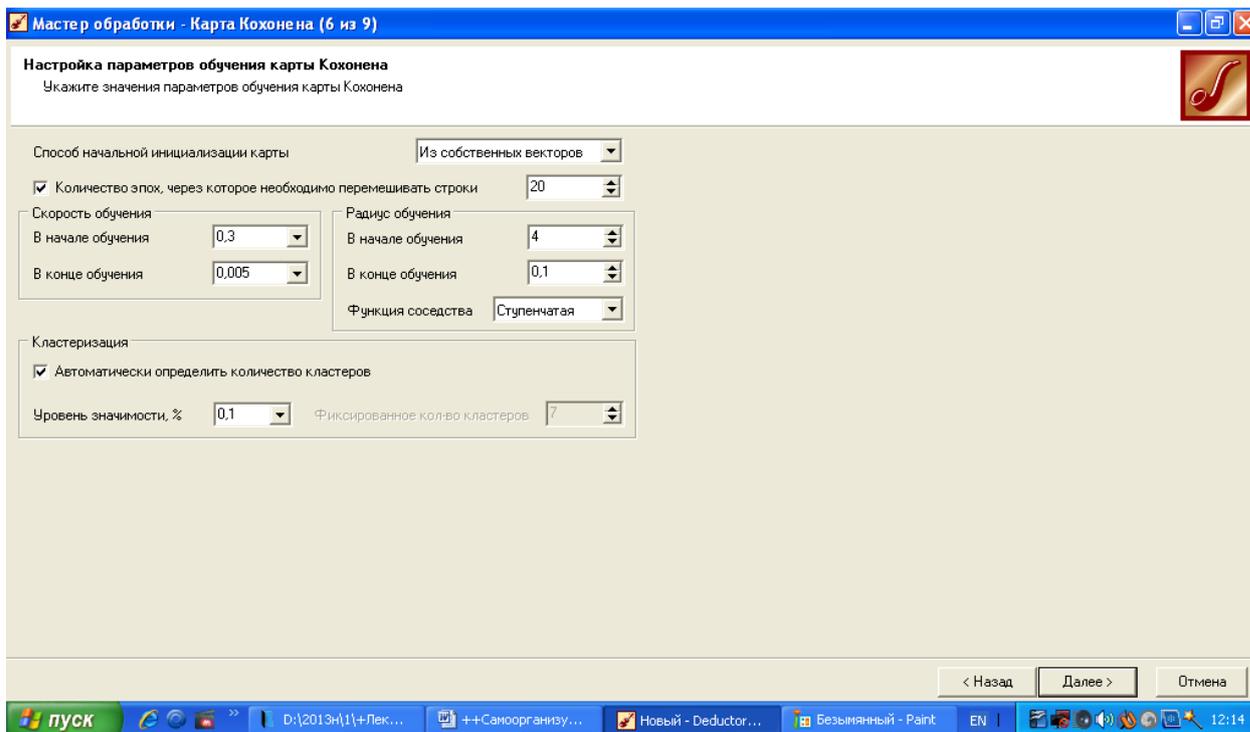
Кроме этого для обучающего и тестового множества в соответствующих секциях окна могут независимо устанавливаться следующие критерии останова обучения:

- *Средняя ошибка меньше* - средняя квадратичная ошибка на обучающем множестве или тестовом множестве меньше заданного значения.
- *Максимальная ошибка меньше* - максимальная квадратичная ошибка на обучающем множестве и тестовом множестве меньше заданного значения.
- *Распознано примеров, (%)* - количество распознанных примеров на обучающем множестве и тестовом множестве больше заданного процента.

При выборе нескольких условий останова процесса обучения происходит по достижении хотя бы одного из них.

Настройка параметров обучения карты Кохонена

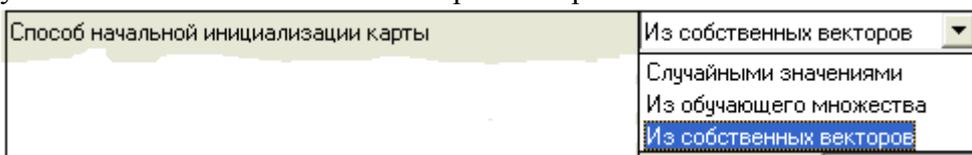
Обучение карты производится итерационными циклами, каждый из которых называется эпохой.



Во время каждой эпохи происходит подстройка весов нейронов карты Кохонена. Подстройка весов во время одной эпохи происходит следующим образом: каждый входной вектор (строка таблицы) обучающей выборки "подтягивает" к себе ближайший по расстоянию нейрон (нейрон-победитель) карты Кохонена с определенной силой (скорость обучения). Вместе с нейроном-победителем подтягиваются и его соседи. Соседство определяется положением нейронов на 2-х мерной четырехугольной или шестиугольной сетке. Здесь расстояние – это обычное Евклидово расстояние между 2-мя точками в многомерном пространстве (входной вектор и веса нейрона победителя).

Способ начальной инициализации

Способ начальной инициализации карты позволяет определить как будут установлены начальные веса нейронов карты.



Возможны три варианта:

- случайными значениями – начальные веса нейронов будут случайными значениями.
- из обучающего множества – в качестве начальных весов будут использоваться случайные примеры из обучающего множества.
- из собственных векторов – начальные веса нейронов карты будут проинициализированы значениями подмножества гиперплоскости, через которую проходят два главных собственных вектора матрицы ковариации входных значений обучающей выборки.

При выборе способа начальной инициализации следует руководствоваться следующей информацией:

1. объемом обучающей выборки;
2. количеством эпох, отведенных для обучения;

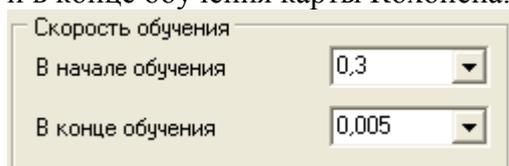
3. размерами обучаемой карты.

Между указанными параметрами и способом начальной инициализации существует много зависимостей. Однако можно выделить несколько главных:

1. Если объем обучающей выборки значительно (раз в 100) превышает количество нейронов карты и время обучения не играет первоочередную роль, то лучше выбрать инициализацию случайными значениями, т.к. это даст меньшую вероятность попадания в локальный минимум ошибки кластеризации.
2. Если объем обучающей выборки не очень велик или ограничено время обучения или необходимо уменьшить вероятность появления после обучения "пустых" нейронов (в которые не попало ни одного экземпляра обучающей выборки), то следует использовать инициализацию примерами из обучающего множества.
3. Инициализацию из собственных векторов можно использовать при любом стечении обстоятельств. Единственное: вероятность появления после обучения "пустых" нейронов выше чем если бы была бы использована инициализация примерами из обучающего множества. Именно этот способ необходимо выбирать при первом ознакомлении с данными.

Скорость обучения

Элементы группы Скорость обучения позволяют задать скорость обучения в начале и в конце обучения карты Кохонена.



Значения можно выбрать из списка или ввести вручную с клавиатуры.

Процесс обучения можно условно разделить на две фазы – грубую подстройку и точную подстройку.

- Для этапа грубой подстройки характерна достаточно большая коррекция весов нейронов сети по прохождении каждой эпохи.
- На этапе точной подстройки величина коррекции значительно уменьшается.

При этом коэффициент (скорость обучения) с которым многомерные координаты (веса) нейрона победителя и его соседей будут двигаться в сторону очередного экземпляра данных изменяется в зависимости от текущей эпохи обучения по правилу, определяемому следующей функцией:

$$v = v_{\text{начало}} * (v_{\text{конец}} / v_{\text{начало}})^{(T / T_{\text{max}})},$$

где

- v – текущий радиус обучения.
- $v_{\text{начало}}$ – начальная скорость обучения.
- $v_{\text{конец}}$ – конечная скорость обучения.
- T_{max} – количество эпох обучения (задается на предыдущем шаге).
- T – текущая эпоха обучения.
- $^{\wedge}$ - показатель степени.

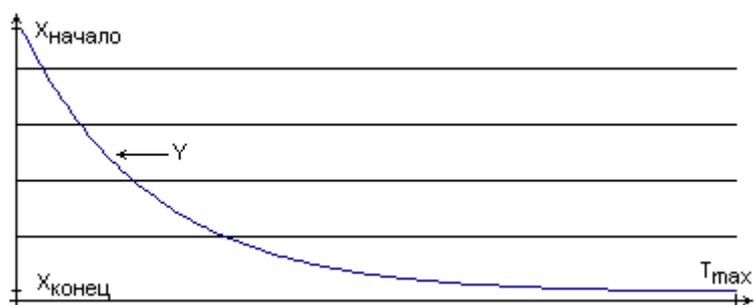


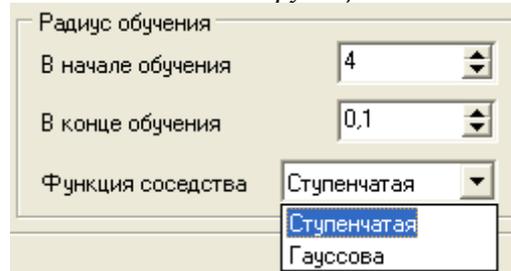
Рис. 1 Примерный вид функции $Y = X_{\text{начало}} * (X_{\text{конец}} / X_{\text{начало}})^{(T/T_{\text{max}})}$

Рекомендуемые значения для скорости обучения:

- в начале обучения 0.1-0.3,
- в конце обучения 0.05-0.005.

Радиус обучения

Элементы группы Радиус обучения позволяют задать *радиус обучения* в начале и в конце обучения карты Кохонена а также *тип функции соседства*.



Радиус обучения - это параметр, который определяет сколько нейронов кроме нейрона-победителя участвуют в обучении. В процессе обучения радиус обучения обычно должен постепенно уменьшаться так, что на заключительных этапах в обучении участвует только нейрон-победитель. При этом радиус обучения изменяется в зависимости от текущей эпохи обучения по правилу, определяемому следующей функцией (примерный вид функции показан на рис. 1):

$$r = r_{\text{начало}} * (r_{\text{конец}} / r_{\text{начало}})^{(T/T_{\text{max}})},$$

где

- r – текущий радиус обучения.
- $r_{\text{начало}}$ – начальный радиус обучения.
- $r_{\text{конец}}$ – конечный радиус обучения.
- T_{max} – количество эпох обучения (задается на предыдущем шаге).
- T – текущая эпоха обучения.

Радиус обучения в начале должен быть достаточно большой – примерно половина или меньше размера карты (максимальное линейное расстояние от любого нейрона до другого любого нейрона). а в конце должен быть достаточно малым – примерно 1 или меньше. Чем больше текущий радиус обучения, тем более грубо подстраивается карта, т.к. приходится корректировать большое количество весов нейронов, и чем меньше текущий радиус обучения, тем более точно подстраивается карта.

Примечание: Начальный радиус обучения подбирается автоматически в зависимости от размера карты. Автоматически подобранный радиус – это всего лишь рекомендуемое значение.

Параметр *Функция соседства* определяет какие нейроны и в какой степени будут считаться соседними по отношению к нейрону-победителю. Этот параметр может принимать два значения: "Ступенчатая", "Гауссова".

Если функция соседства ступенчатая, то "соседями" для нейрона-победителя будут считаться все нейроны, линейное расстояние до которых не больше текущего радиуса обучения. При этом варианте функции соседства процесс обучения происходит немного быстрее, но качество результата может быть немного хуже, чем если бы использовалась Гауссова функция соседства.

Если используется Гауссова функция соседства, то "соседями" для нейрона-победителя будут считаться все нейроны карты, но в разной степени полноты. При этом степень соседства определяется следующей функцией:

$$h = \exp(-d*d)/(2*r),$$

где

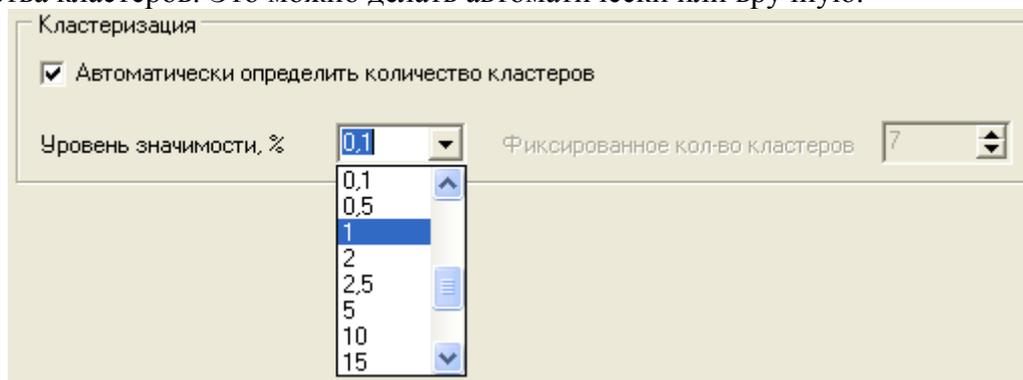
- h – значение, определяющее степень соседства.
- d – линейное расстояние от нейрона победителя до нейрона "соседа",
- r – текущий радиус обучения.

После определения степени соседства очередного нейрона, его веса будут изменены не с текущей скоростью обучения, а со скоростью равной текущей скорости обучения умноженной на коэффициент, определяющий степень соседства h .

При использовании Гауссовой функции соседства обучение проходит более плавно и равномерно, т.к. одновременно изменяются веса всех нейронов, что может дать немного лучший результат, чем если бы использовалась ступенчатая функция. Однако время, необходимое на обучения требуется немного больше по причине того, что на всех эпохах корректируются все нейроны.

Кластеризация

Секция *Кластеризация* предназначена для указания способа определения количества кластеров. Это можно делать автоматически или вручную.



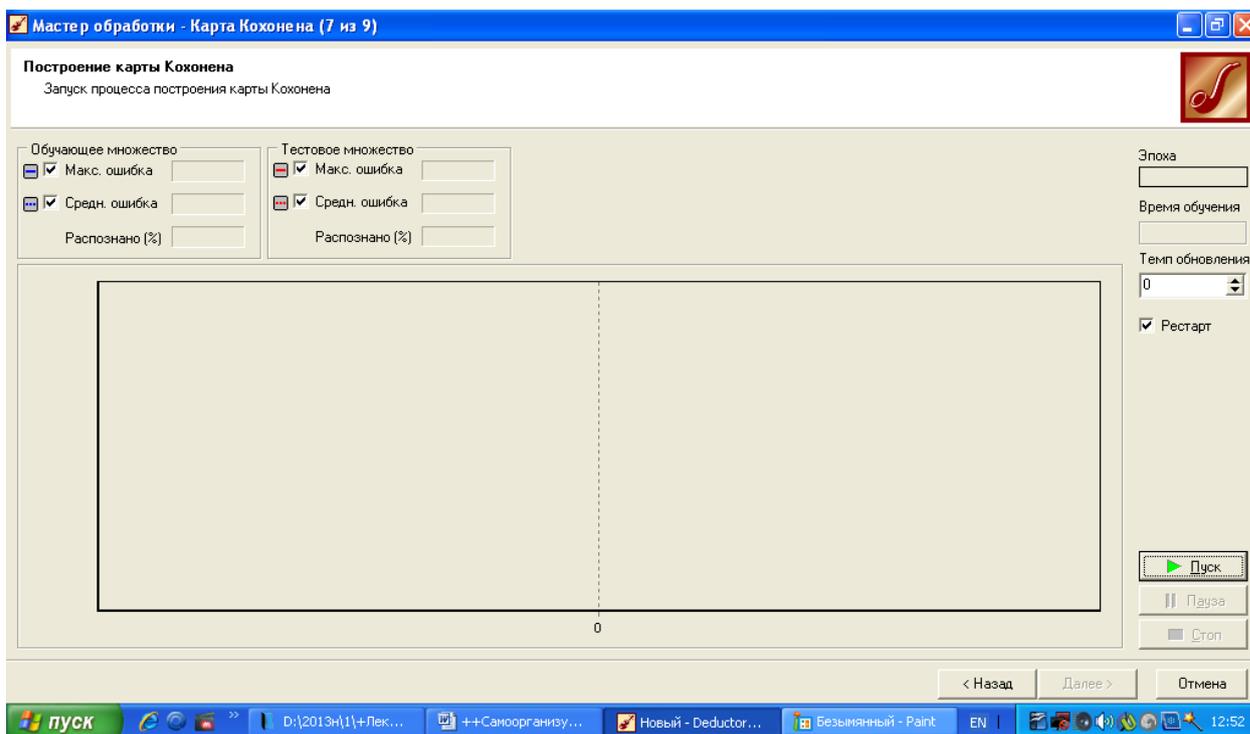
Автоматически определять количество кластеров – после установки этого флажка программа будет автоматически определять количество кластеров.

Уровень значимости – параметр автоматического определения кластеров. Чем больше этот параметр, тем большее количество кластеров будет получено.

Фиксированное количество кластеров – параметр доступный при ручном определении количества кластеров. Собственно задает желаемое количество кластеров, на которое будут разбиты нейроны карты Кохонена.

Построение карты Кохонена

На данном шаге производится собственно обучение карты с заданными параметрами.



В зависимости от объема обрабатываемых данных и быстродействия компьютера, оно может занять определенное время.

Остановить процесс обучения можно с помощью кнопки "Стоп". После этого появляется возможность вернуться на предыдущие шаги Мастера обработки для проверки и изменения параметров. Далее процесс обучения может быть запущен заново, продолжен или отменен. Если обучение было завершено успешно, то в секции "Название текущего процесса" появится сообщение "Успешное завершение".

Для управления процессом обучения предусмотрены следующие кнопки:

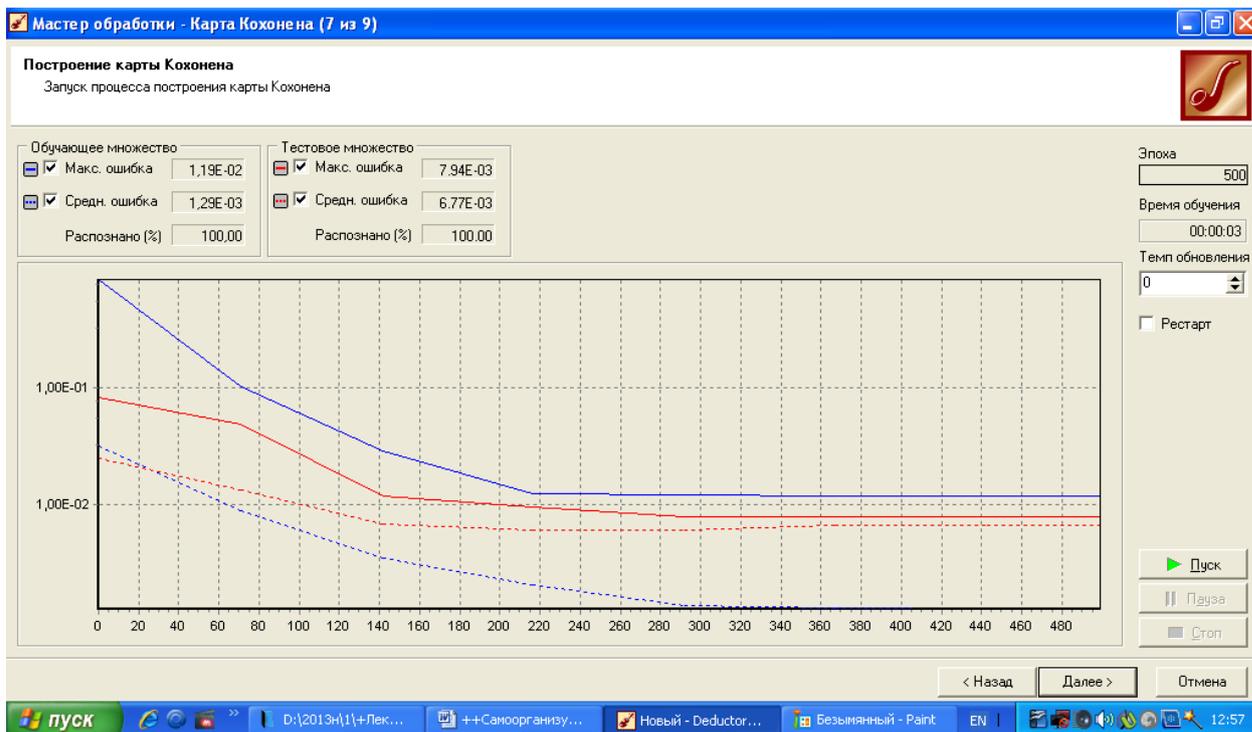
- "Пуск" - запускает процесс в первый раз или возобновляет после паузы.
- "Пауза" - временно приостанавливает обработку. Временная приостановка процесса имеет смысл в двух случаях:
 - для оценки текущих результатов процесса обучения, например, просмотра графиков динамики ошибок обучения.
 - при необходимости освободить ресурсы процессора для других приложений.
- "Стоп" - останавливает процесс без возможности его продолжения.

Практически, останавливать процесс обучения модели принудительно имеет смысл если:

- значения ошибок длительное время не уменьшаются.
- процент распознанных примеров не увеличивается.

В процессе обучения в секциях "Обучающее множество" и "Тестовое множество" отображаются максимальная квадратичная ошибка и средняя квадратичная ошибка на обучающем множестве и тестовом множестве соответственно, а также процент распознанных примеров.

Обучение может с большой долей вероятности считаться успешным, если процент распознанных примеров на обучающем и тестовом множествах достаточно велик (близок к 100%).

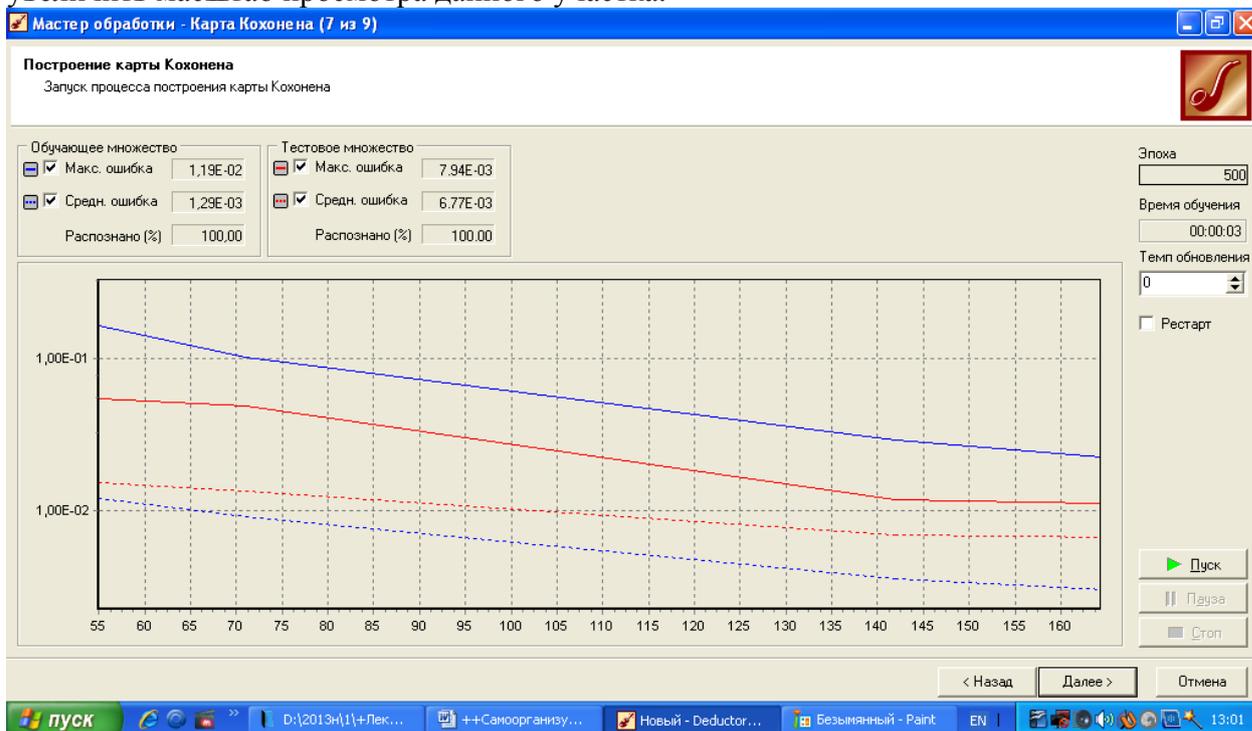


В процессе обучения в окне отображаются следующие графики хода обучения для обучающего (синяя линия) и тестового (красная линия) множеств:

- максимальная квадратичная ошибка на обучающем множестве и тестовом множестве (сплошная линия).
- средняя квадратичная ошибка на обучающем множестве и тестовом множестве (пунктирная линия).

Любой из графиков можно скрыть (показать). Для этого достаточно сбросить (установить) соответствующий флажок.

Если какой-либо участок графика требуется рассмотреть более подробно, можно увеличить масштаб просмотра данного участка.



Для этого следует выделить прямоугольную область с нужным участком, двигая указатель с нажатой левой кнопкой мыши от левого верхнего к правому нижнему углу прямоугольника. Повторение данного действия приведет к дальнейшему увеличению масштаба. Чтобы вернуть исходный масштаб графика, достаточно выделить любой участок графика, двигая при этом мышью с нажатой левой кнопкой от правого нижнего угла прямоугольника к левому верхнему.

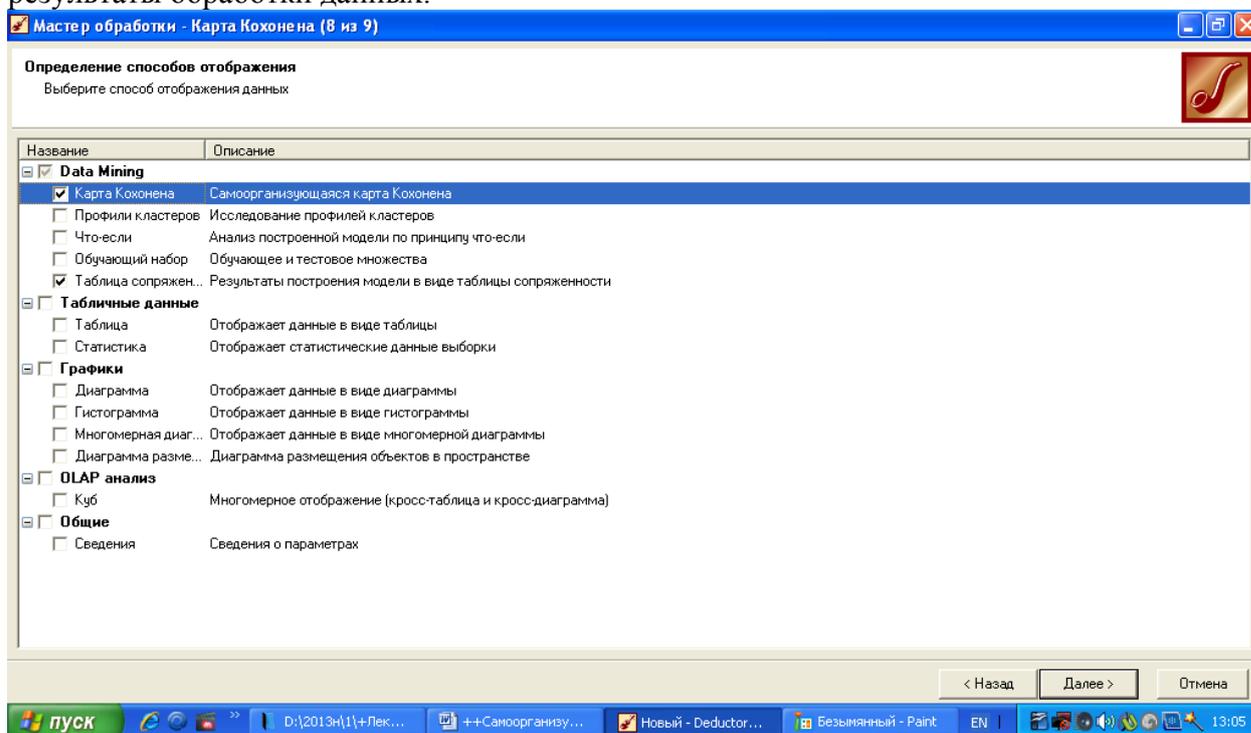
В правой части окна постоянно отображаются номер текущей эпохи и время, прошедшее с начала обучения.

В поле "Темп обновления" можно задать число эпох обучения, через которое будет происходить обновление графика.

Флажок "Рестарт" позволяет включить режим инициализации начальных весов сети перед началом обучения. Если флажок сброшен, то при повторном запуске обучения после остановки будет иметь место так называемое "дообучение сети", когда обучение будет начато с текущими весами. Режим "Переобучить" соответствует установленному флажку "Рестарт", режим "Дообучить" - сброшенному флажку.

Выбор способа отображения

На данном шаге пользователь должен выбрать, в каком виде будут отображены результаты обработки данных.



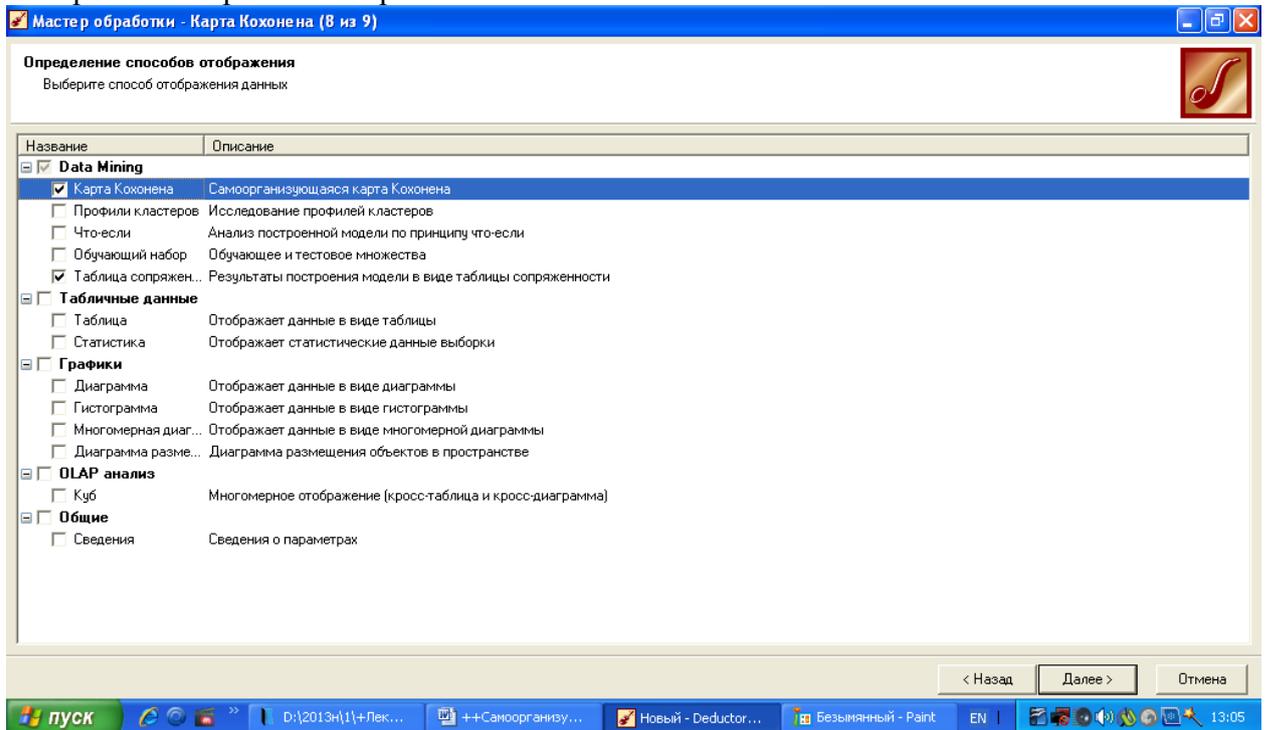
Для этого достаточно пометить нужные виды отображения флажками и щелкнуть по кнопке "Далее".

Для выборки данных, полученных в результате обработки данных с помощью карт Кохонена, доступны следующие виды отображения:

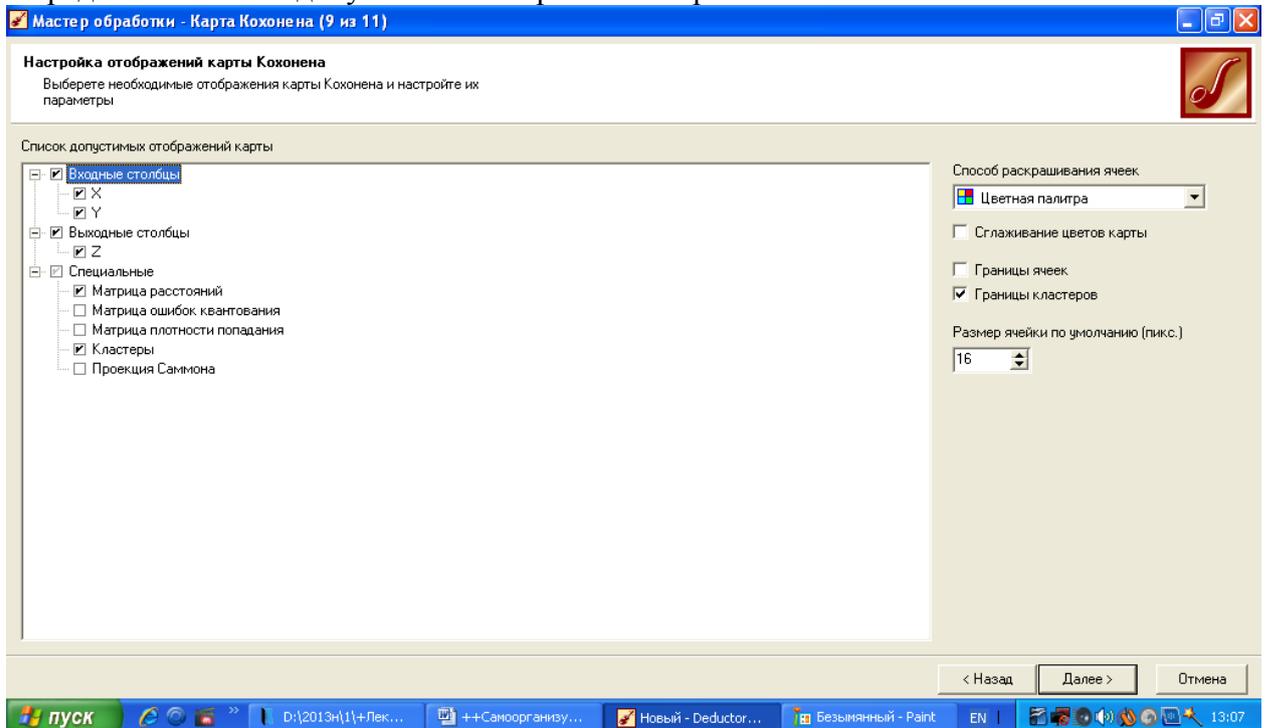
- Обучающий набор;
- Диаграмма рассеяния;
- Самоорганизующаяся карта;
- Что-если;
- Таблица сопряженности.
- Таблица;
- Статистика;

- Диаграмма;
- Гистограмма;
- Куб;
- Сведения.

Настройка отображений карты Кохонена:



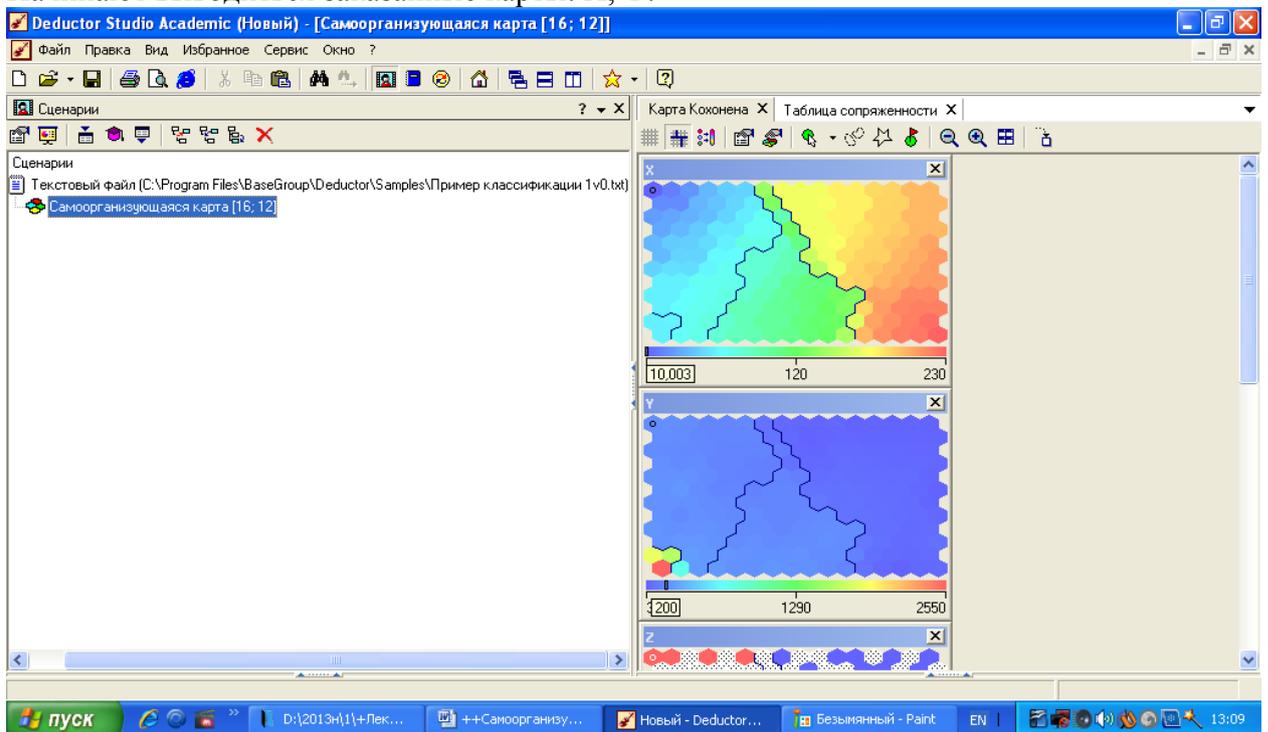
Определение списка допустимых отображений карты:



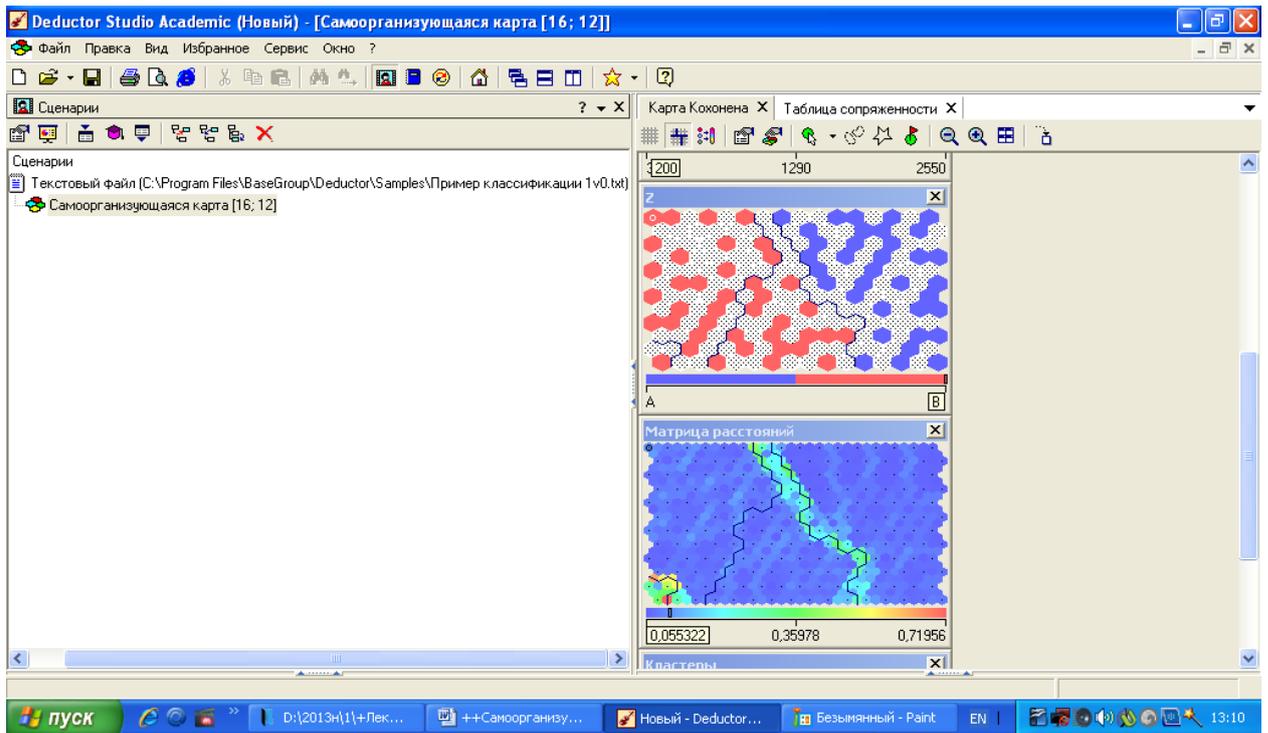
При успешном завершении сообщаются имя и метка карты:



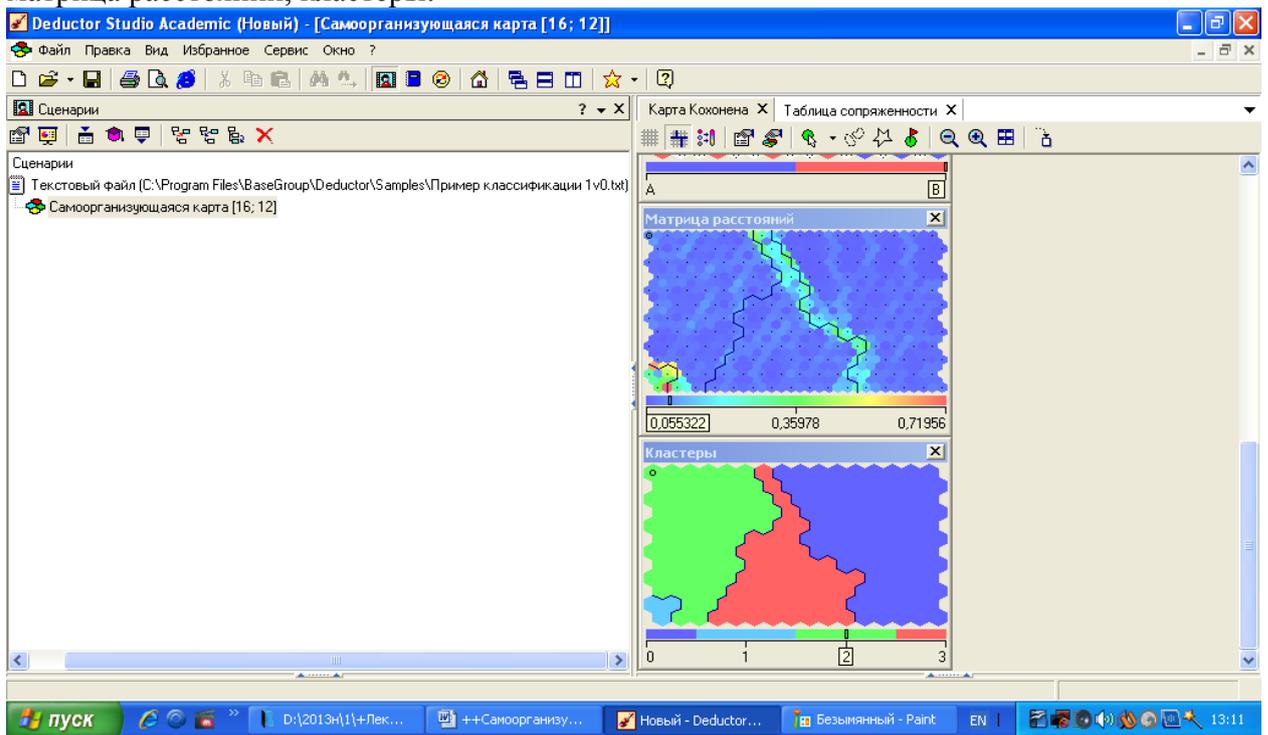
Начинают выводиться заказанные карты: X, Y:



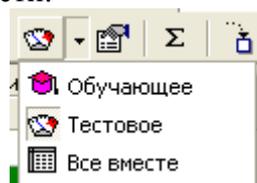
Затем выводятся Z, A:



матрица расстояний, кластеры:



Возможности таблицы сопряжённости:



Обучающее + тестирующее

Классифицировано			
Фактически	A	B	Итого
A	49		49
B		49	49
Итого	49	49	98

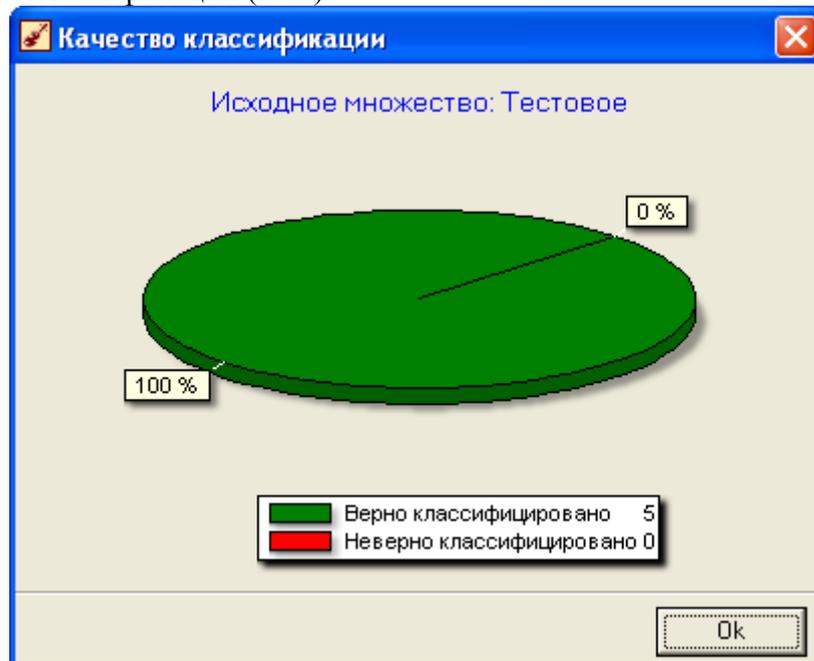
Только обучающее:

Классифицировано			
Фактически	A	B	Итого
A	47		47
B		46	46
Итого	47	46	93

Только тестирующее:

Классифицировано			
Фактически	A	B	Итого
A	2		2
B		3	3
Итого	2	3	5

Оценка качества классификации (Σ):



Показать/скрыть таблицу (F12)

X	Y	Z	Z_OUT	Номер ячейки	Расстояние до центра ячейки
180	90	A	A	58	0,007939460
180	80	A	A	58	0,003971200

Или точнее:

X	Y	Z	Z_OUT	Номер ячейки	Расстояние до центра ячейки
180	90	A	A	58	0,007939460
180	80	A	A	58	0,003971200



Экспорт в MS Excel... - в данной версии не работает.

Сведения об узле

На данном шаге Мастера обработки пользователь может задать "Имя", "Метку" и "Описание" для данного узла обработки данных. "Имя" должно быть уникальным в пределах сценария. Параметр "Метка" отображает заданную метку в дереве сценариев. В параметре "Описание" пользователь может задать дополнительное описание, поясняющее действия, выполняемые этим узлом обработки данных. Для завершения настройки параметров узла обработки данных необходимо нажать кнопку "Готово".

Лекция 7. Нейросетевое исследование в статистическом пакете SPSS.

Введение.

Норман Най, Хедли Халл и Дейл Бент разработали первую версию системы SPSS в 1968 году, затем этот пакет развивался в рамках Чикагского университета. С 1975 года проект выделился в отдельную компанию SPSS Inc. Первая версия пакета под Microsoft Windows вышла в 1992 году. На данный момент также существуют версии под MacOS X и Linux.

В 2009 году компания SPSS произвела ребрендинг своего статистического пакета, который теперь стал называться PASW Statistics (Predictive Analytics SoftWare). 29 июля 2009 года компания SPSS объявила о том, что она приобретает фирмой IBM.

Пакет содержит большое количество программ математической статистики и является одним из наиболее используемых пакетов этого профиля в России. Кроме этого пакета широко используются в России Statistica, SAS, BMDP и другие. Среди программ математической статистики в SPSS предусмотрен модуль «Нейронные сети».

В SPSS считается, что прогнозирующие нейронные сети особенно полезны в случаях, когда основной процесс сложен, например, таков как:

1. Прогноз потребительского спроса.
2. Предсказание вероятности ответа на маркетинг продажи товаров по почте, чтобы определить, каким домашним хозяйствам из списка рассылки должны быть посланы предложения.
3. Выигрыш претендента и определение риска выдаваемого кредита претенденту.
4. Обнаружение мошеннических сделок в базе данных страховых исков.

Для решения таких задач в SPSS используются два типа нейронных сетей - многослойный перцептрон (Multilayer Perceptron - MLP) и сети радиальной основной функции (Radial Basis Function - RBF).

Среди многочисленных видов пакетов SPSS фирма разработала IBM SPSS Modeling Nodes 479, в состав которого входит большее количество типов нейронных сетей, в том числе и сеть Кохонена.

По мнению специалистов SPSS нейронная сеть представляет собой параллельный распределенный процессор, у которого есть естественная склонность к хранению основанного на опыте знания и предоставлению доступа к нему для использования.

Эта посылка имеет сходство с биологическими системами в двух отношениях:

1. Знание приобретено сетью посредством процесса обучения.
2. Силы связи между нейронами, известные как синаптические веса, используются, чтобы сохранить знание.
3. Наложение различных допущений допускается только при извлечении информации из данных, но не до него.

Нейронные сети отличаются от традиционных статистических методов, но могут иметь определённое сходство с ними. Например, традиционная линейная модель регрессии может приобрести знание через метод наименьших квадратов и выразить это знание в коэффициентах регрессии. В этом смысле регрессионную модель можно рассматривать, как нейронную сеть. Тогда можно утверждать, что линейная регрессия - особый случай нейронных сетей определенного типа. Однако, линейная регрессия имеет ряд допущений, которые наложены прежде, чем извлекается информация из данных - выдвигается гипотеза предварительного определения отношений между зависимыми и независимыми переменными. Вместо этого в нейронных сетях форма отношений определяется во время процесса обучения.

Рассмотрим пример использования многослойного перцептрона для решения задачи о классификации набора данных.

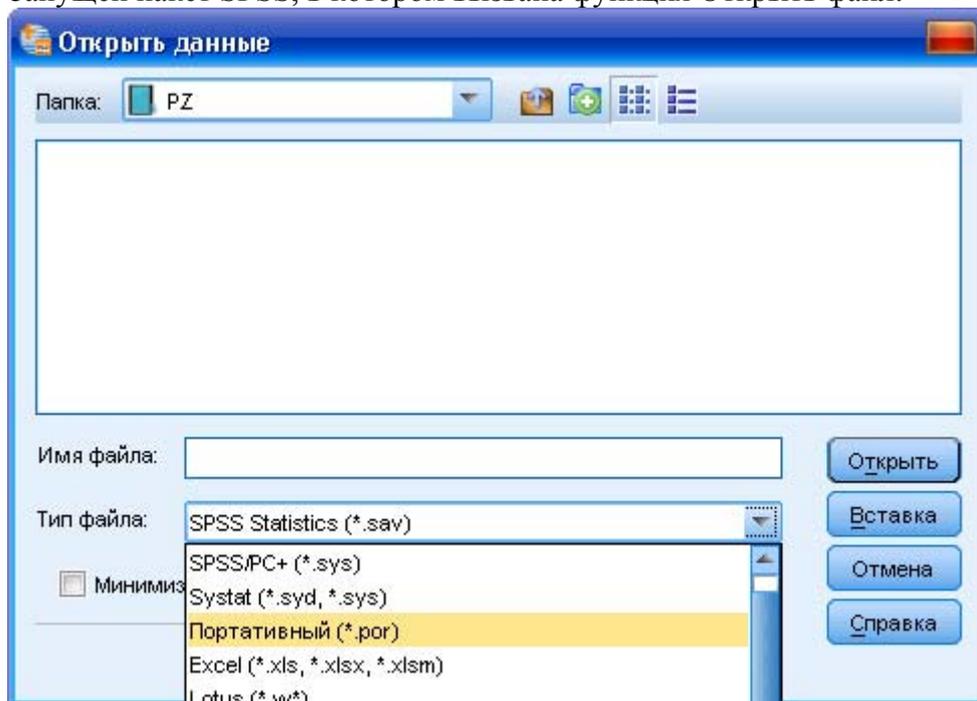
Многослойный перцептрон (MLP) представляет собой прогнозирующую модель для одной или более зависимых (целевых) переменных, на основе характеризующих процесс переменных.

Подготовка данных для исследования.

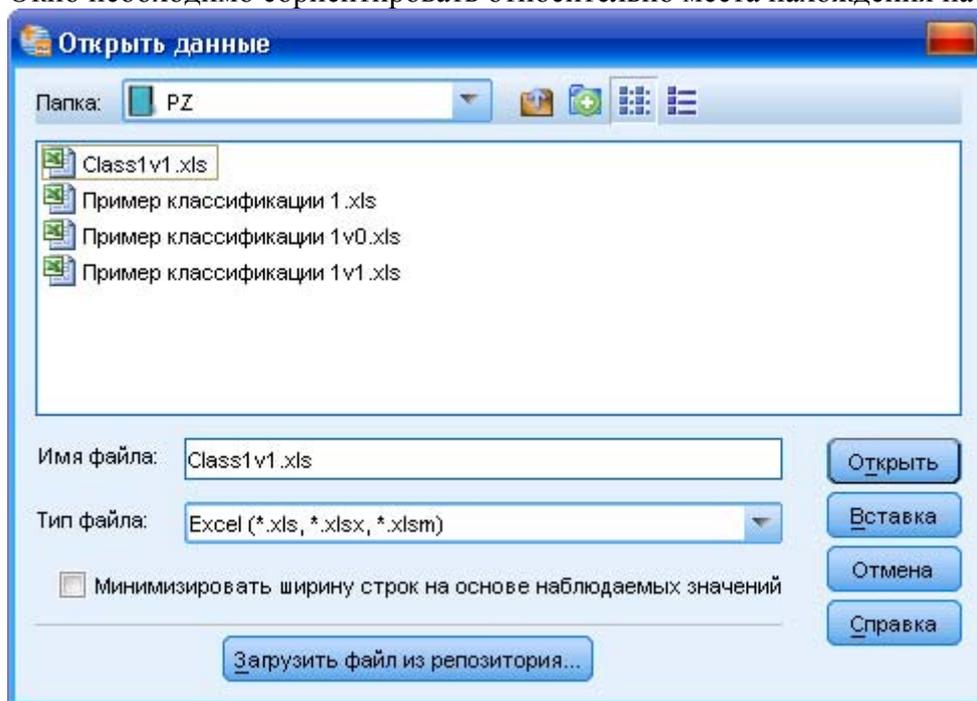
Создан в Excel файл данных «Пример классификации 1.xls».

Он продублирован в «Пример классификации 1v0.xls» и «Пример классификации 1v1.xls», по которым для SPSS создан файл «Class1v1.xls».

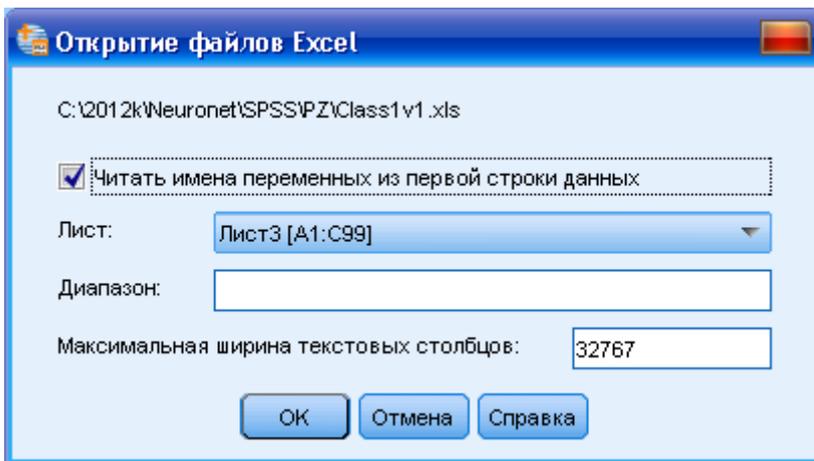
Запущен пакет SPSS, в котором вызвана функция Открыть файл:



Окно необходимо сориентировать относительно места нахождения набора данных:



Если файл создан в Excel,



После чтения файла окно заполняется информацией:

	Имя	Тип	Ширина	Десятич...	Метка	Значения	Пропущенн...	Ширина ...	Выравнивание	Шкала
1	X	Числовая	11	0		Нет	Нет	11	По право...	Количество...
2	Y	Числовая	11	0		Нет	Нет	11	По право...	Количество...
3	Z	Текстовая	1	0		Нет	Нет	1	По левом...	Номинальная
4										

Причём, файл может быть представлен либо в виде описании переменных, либо в виде представления фактического их значения:

	X	Y	Z	пер											
1	230	30	A												
2	230	40	A												
3	230	50	A												
4	230	60	A												
5	230	70	A												
6	230	80	A												
7	230	90	A												
8	220	30	A												
9	220	40	A												
10	220	50	A												
11	220	60	A												
12	220	70	A												
13	220	80	A												
14	220	90	A												
15	210	30	A												
16	210	40	A												
17	210	50	A												
18	210	60	A												
19	210	70	A												
20	210	80	A												
21	210	90	A												
22	200	30	A												
23	200	40	A												
24	200	50	A												
25	200	60	A												

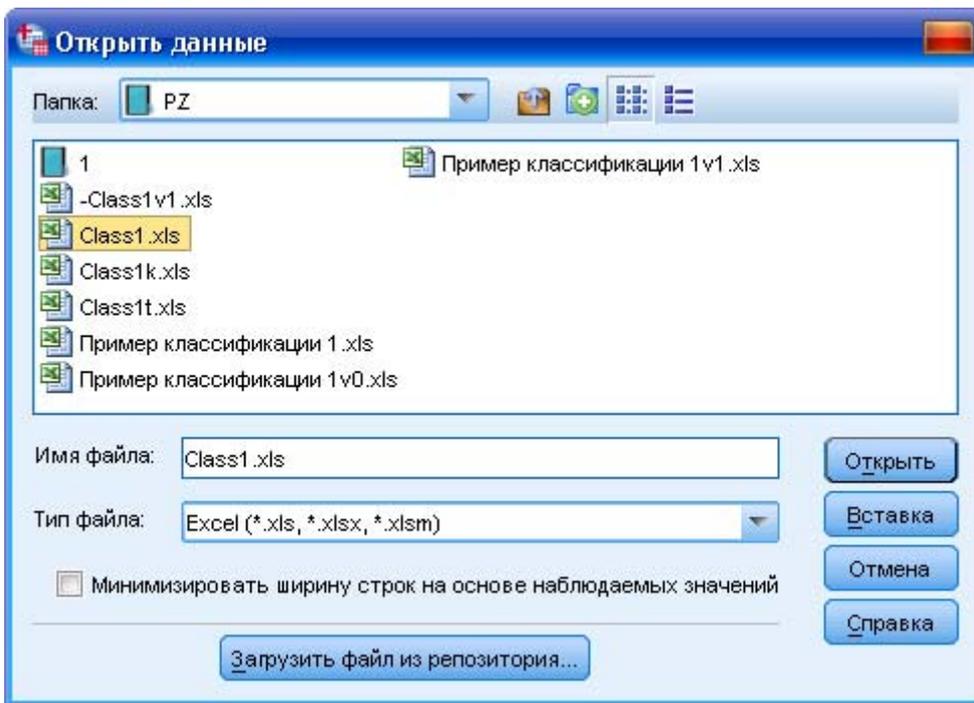
Данные нужно перемешать, распределить их по файлу случайным образом. Это можно сделать средствами Excel или в SPSS. Образовался файл Class1v1-0.csv.

	A	B	C	D	E
1	X	Y	Z		
2	210	50	A		
3	50	160	B		
4	200	70	A		
5	120	190	B		
6	220	80	A		
7	100	190	B		
8	230	40	A		
9	190	90	A		
10	80	150	B		
11	90	240	B		
12	80	170	B		
13	230	80	A		
14	100	170	B		
15	120	210	B		
16	60	170	B		
17	200	40	A		
18	70	220	B		
19	220	60	A		
20	210	30	A		
21	220	70	A		
22	130	200	B		

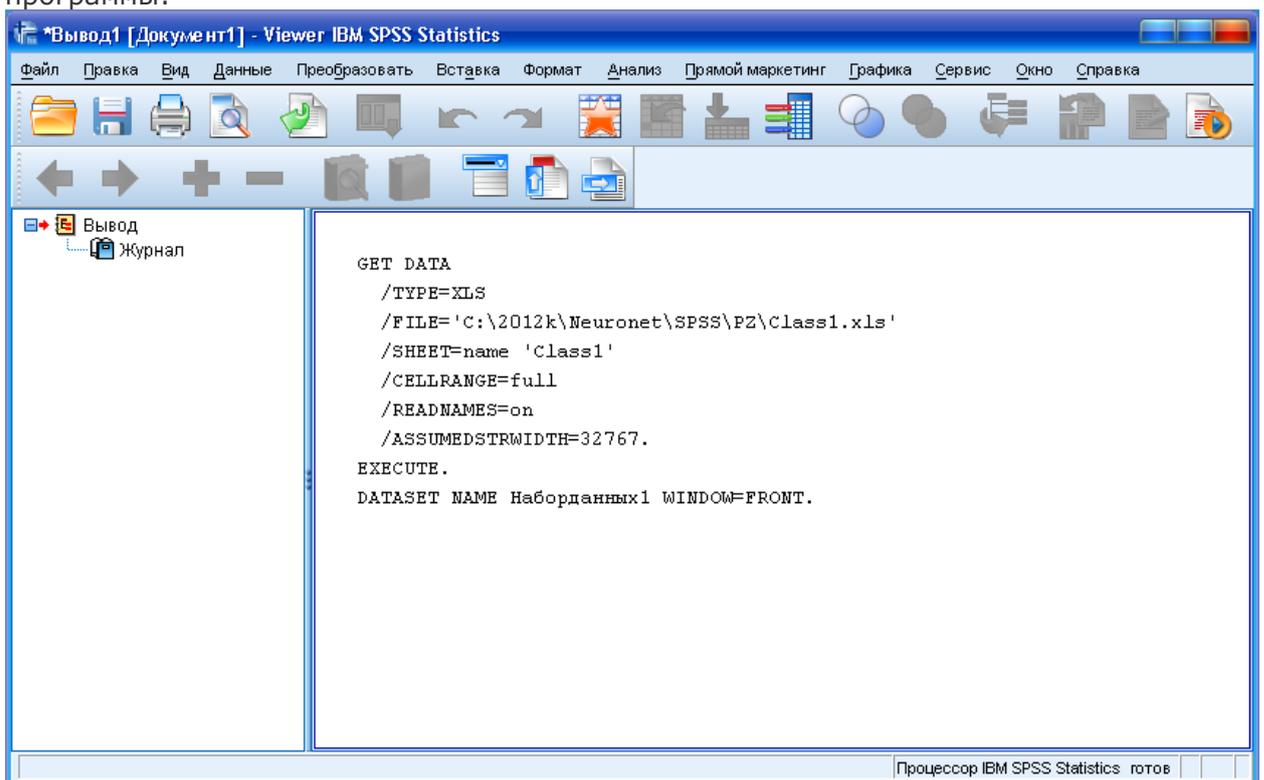
Файл содержит 98 строк + строка с названиями столбцов.

Настройка пакета SPSS.

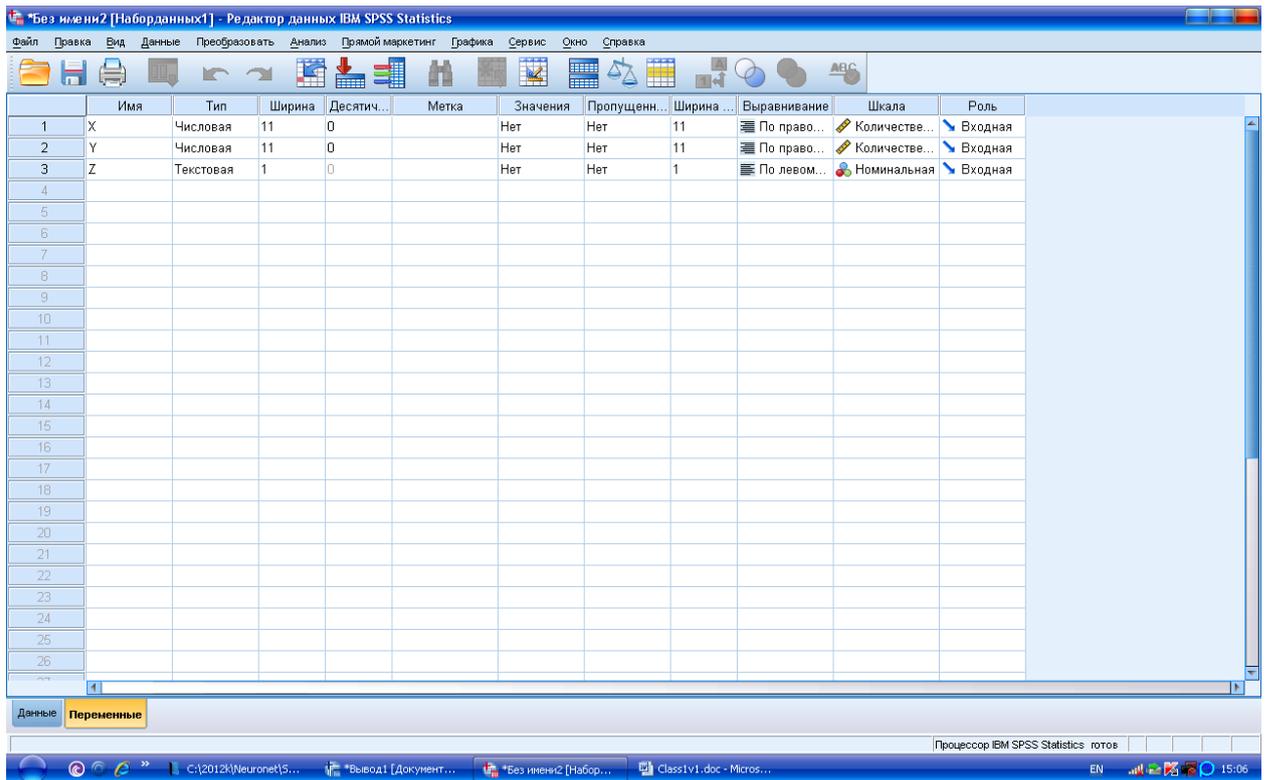
Пакет SPSS открывается в редакторе данных:



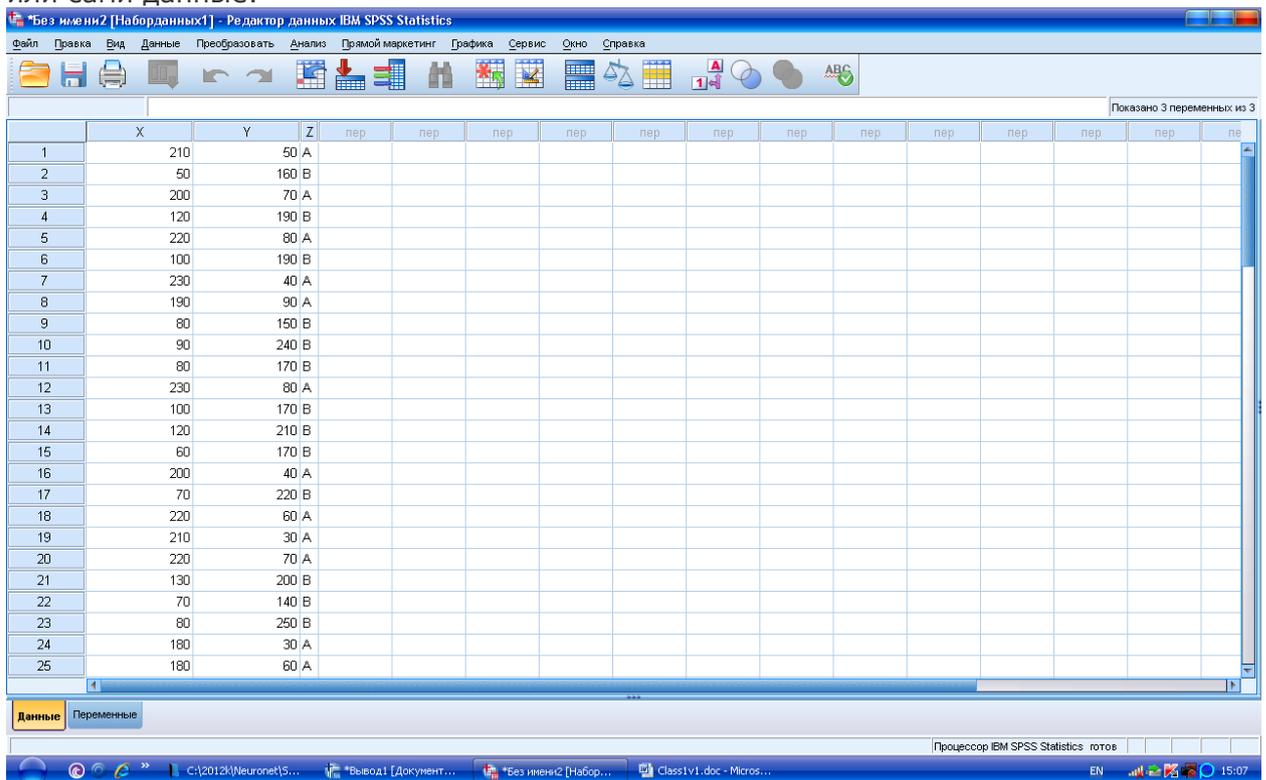
Сразу формируется отчёт, содержащий фрагмент подготовленной к исследованию программы:



При исполнении этой программы вводятся исходные данные. Можно посмотреть их описание:

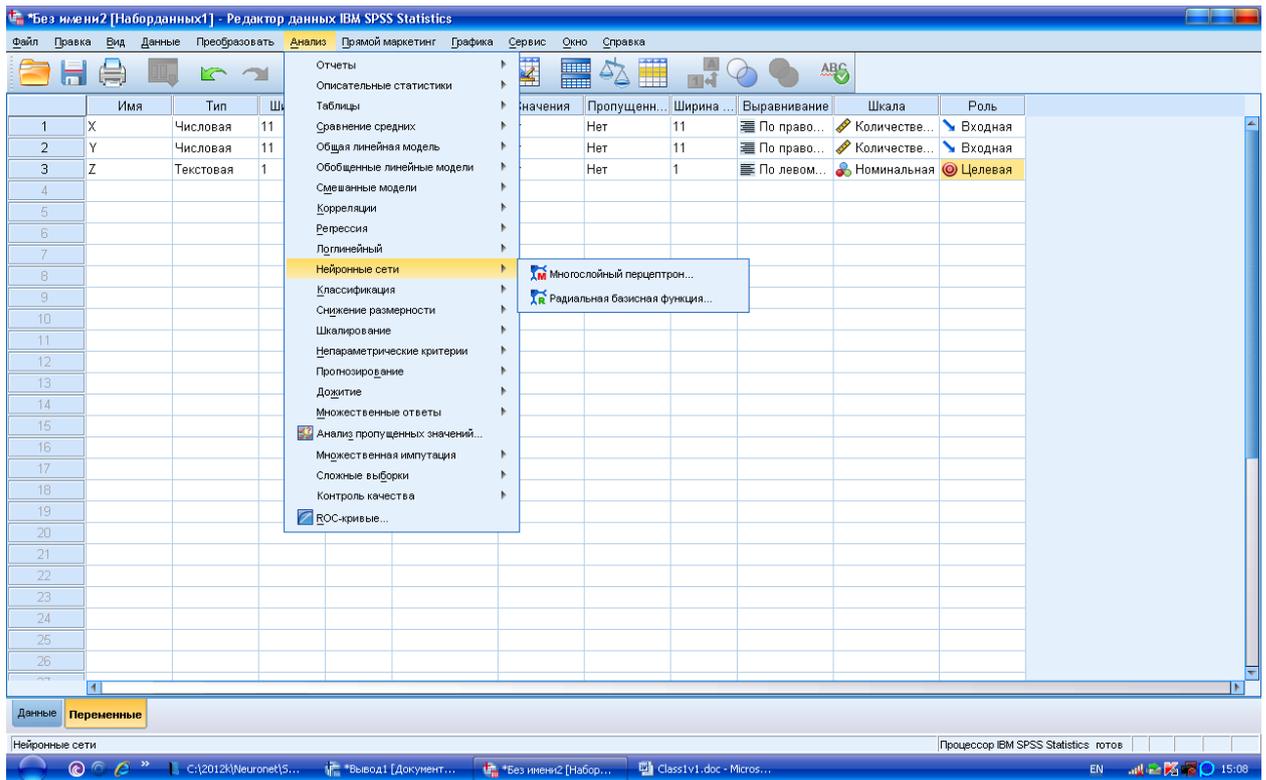


или сами данные:

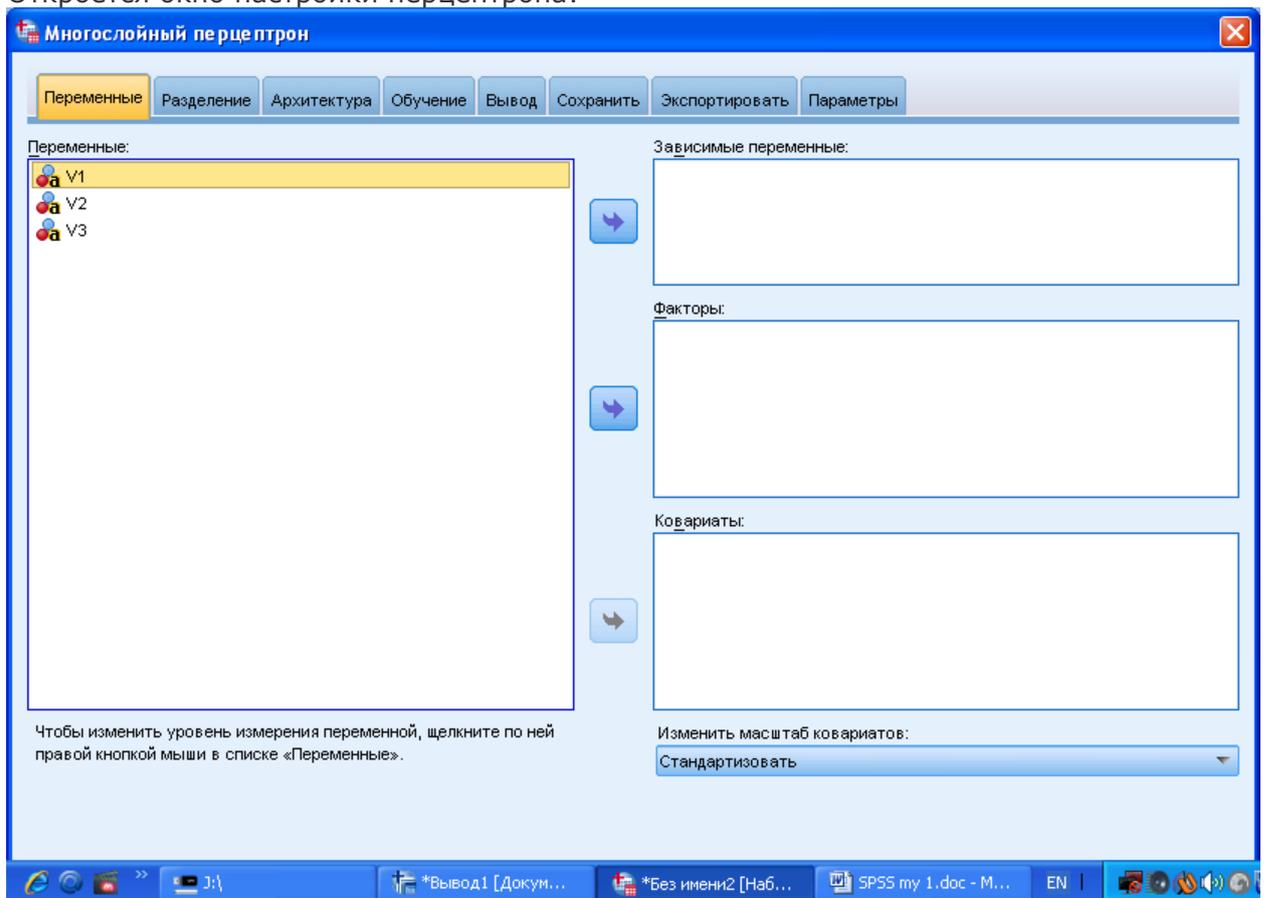


Настройка сети.

Для настройки нейросетевого исследования необходимо запустить: Анализ – Нейронные сети – Многослойный перцептрон:



Откроется окно настройки перцептрона:



Для настройки перцептрона необходимо произвести 8 настроек:

- Переменные
- Разделение
- Архитектура

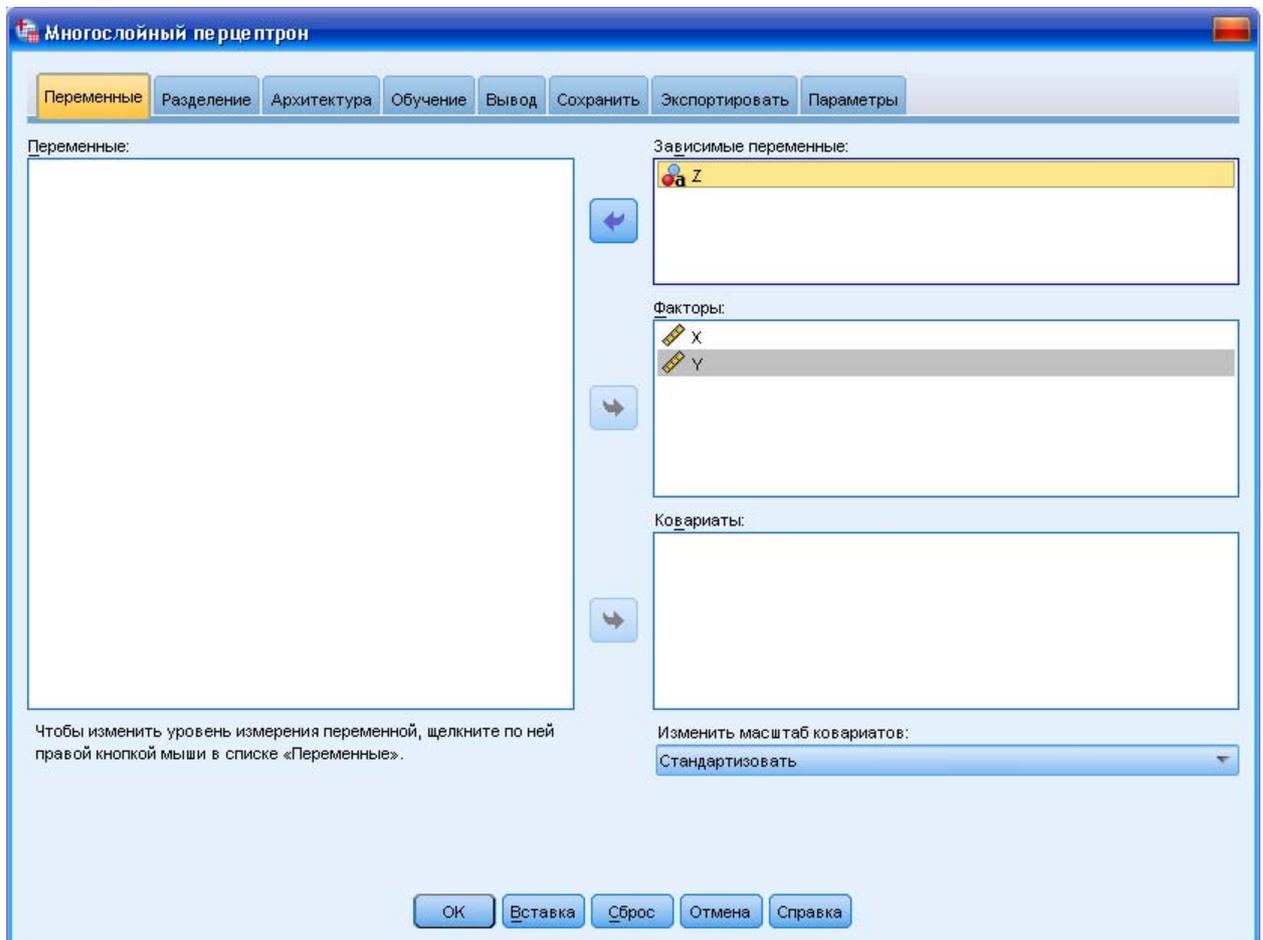
- Обучение
- Вывод
- Сохранение
- Экспорт
- Настройка параметров.

Настройка переменных.

При настройке переменных необходимо определить, какие из них относятся к зависимым переменным, какие – к факторам, и какие – к ковариатам, и как работать с масштабом ковариатов.

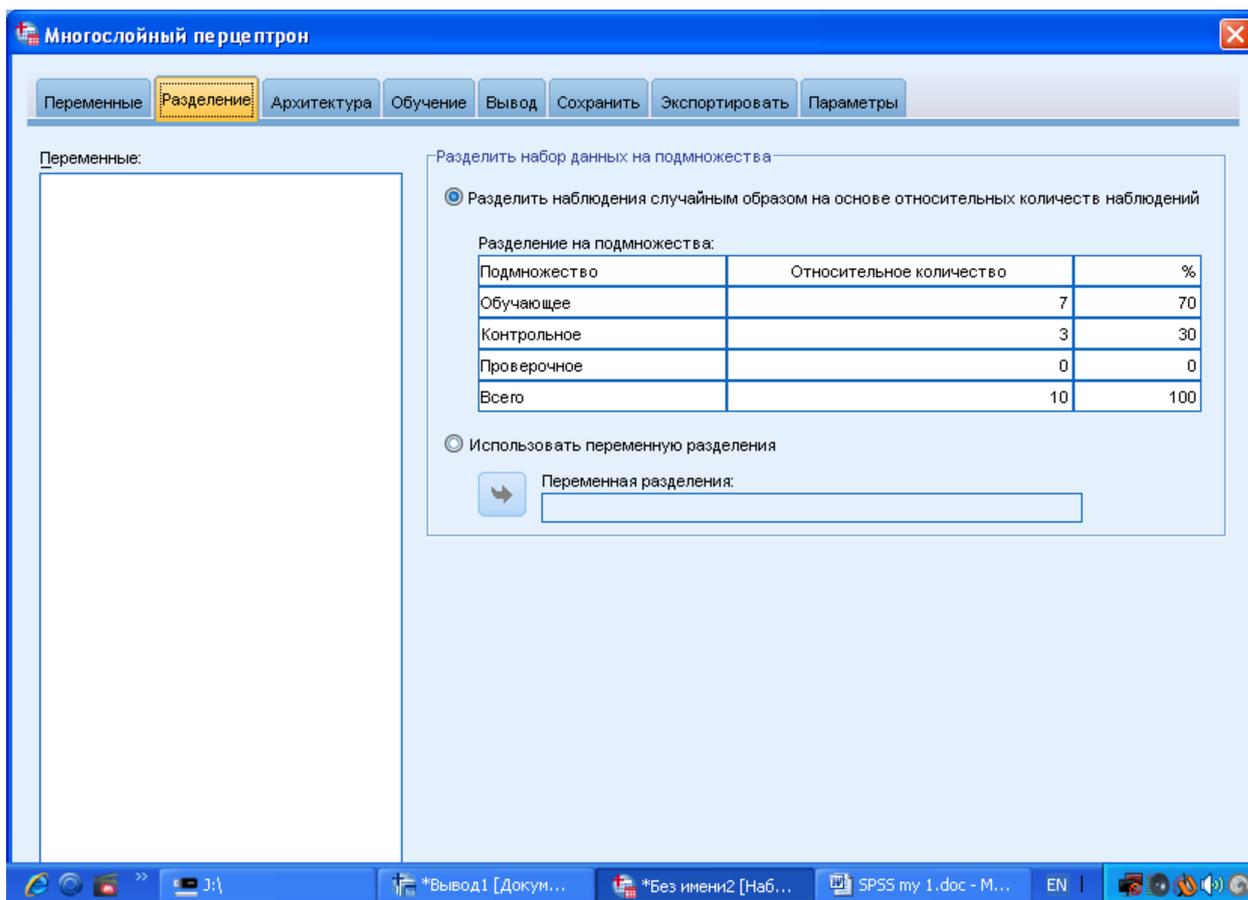
Зависимые и независимые переменные. Переменные, значения которых определяется с помощью измерений в ходе эксперимента (например, балл, набранный при тестировании), называются *зависимыми* переменными. Переменные, которыми можно управлять при проведении эксперимента (например, методы обучения или другие критерии, позволяющие разделить наблюдения на группы или классифицировать) называются *факторами* или *независимыми* переменными.

Разместим переменные так: Z-зависимая; X и Y – факторы.



Разделение переменных.

Все имеющиеся наблюдения разделяются на подмножества: обучающее, контрольное и проверочное. Это разделение можно произвести либо определив относительное количество каждого из них – тогда случайным образом произойдёт разделение набора данных на три части в соответствии с установленным количеством или процентным составом; - либо с помощью специальной переменной разделения.

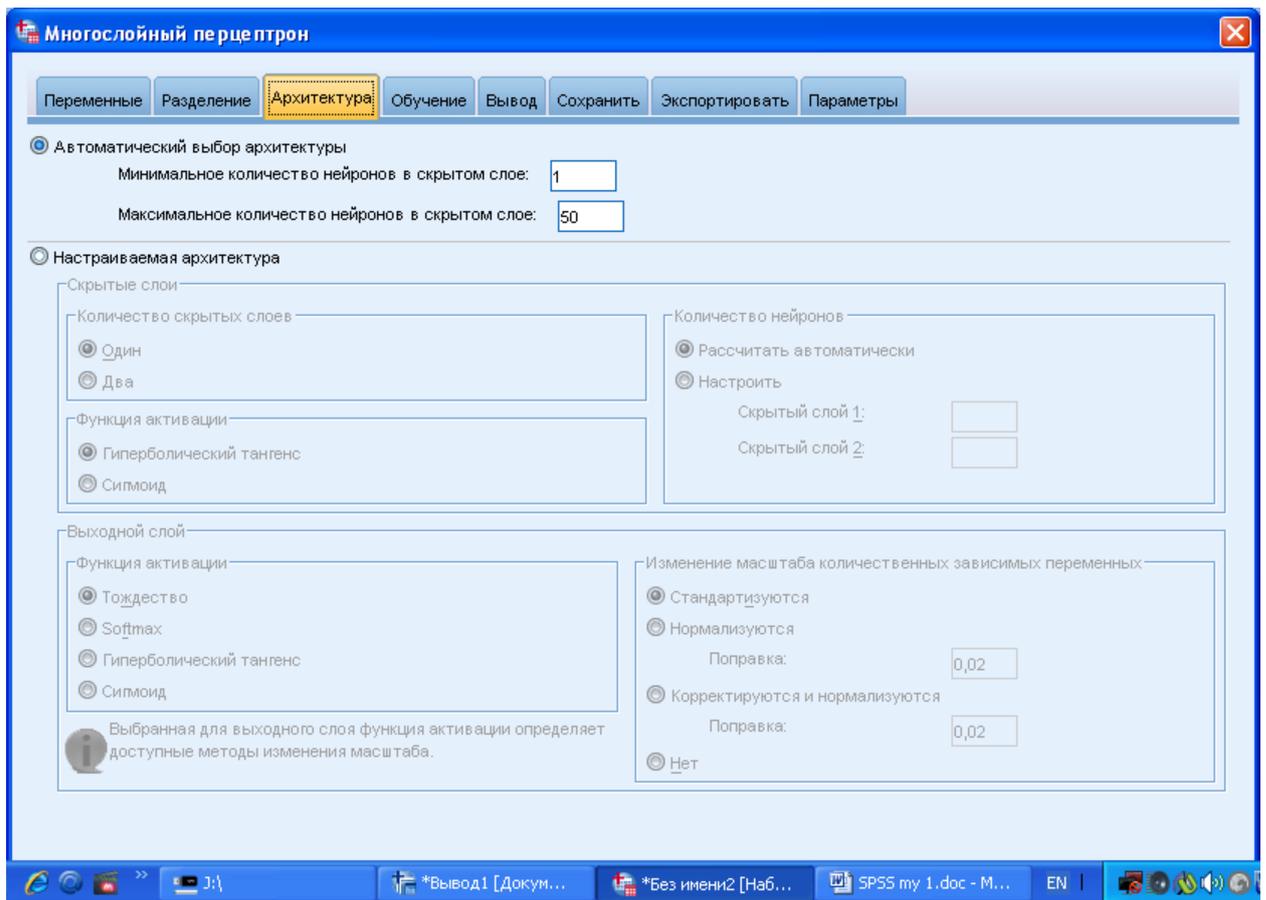


Архитектура нейросети.

Архитектура нейросети может быть сформирована автоматически, с определением только двух параметров: минимального и максимального количества нейронов в скрытом слое; или выбрана путём ручной настройки.

Во втором случае определяется количество скрытых слоёв, количество нейронов в каждом скрытом слое (этот параметр может быть определён и автоматически), функции активации: гиперболический тангенс, или сигмоид.

При определении выходного слоя нейронов функций активации можно использовать больше: тождество, Sogtmax, гиперболический тангенс, или сигмоид. При этом выбранная для выходного слоя функция активации определяет доступные методы изменения масштаба количественных зависимых переменных: масштабы стандартизуются, нормализуются (при нормализации может вводиться поправка), корректируются и нормализуются (возможно – так же с поправкой), или не изменяются.



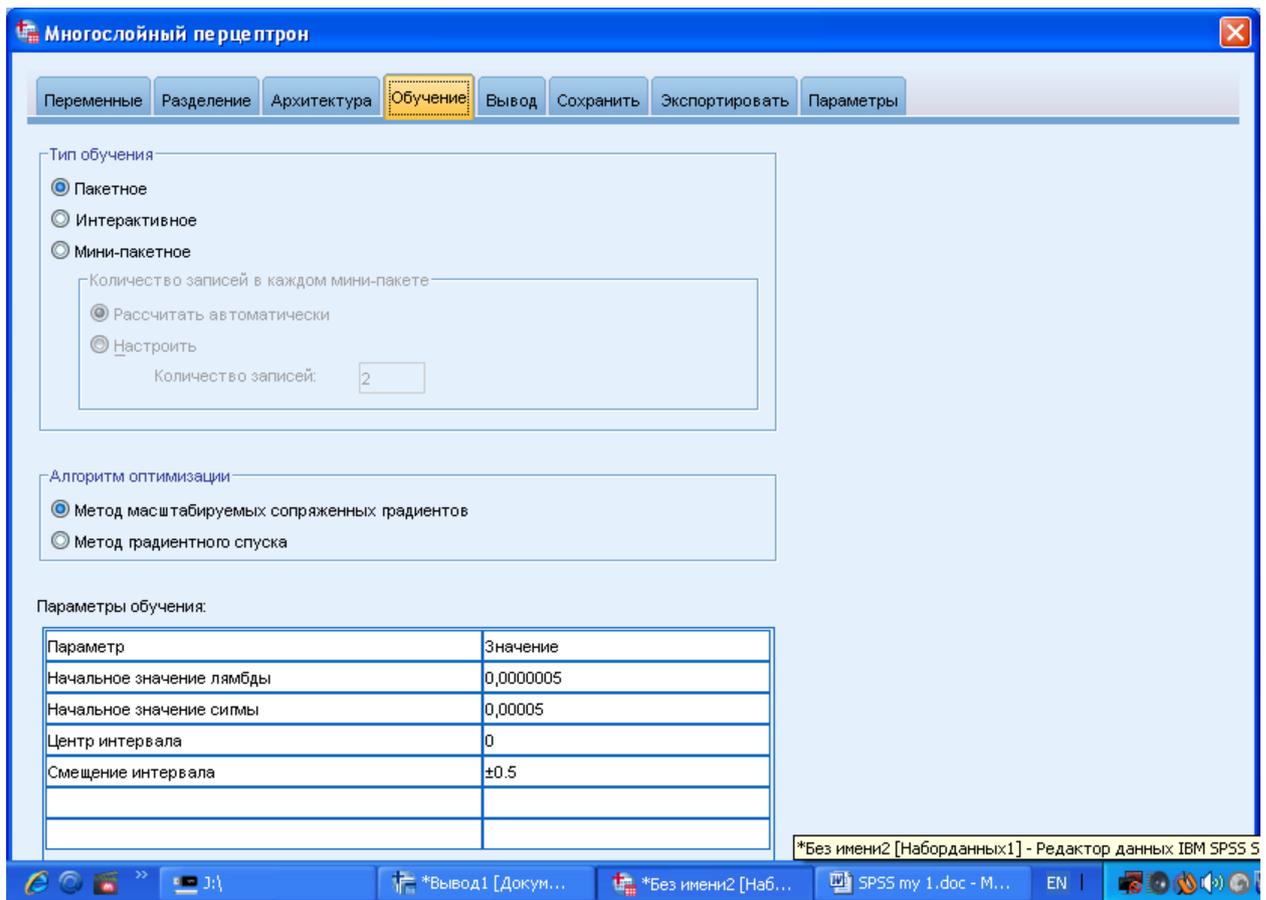
Обучение нейросети.

В этом режиме настройки затрагивают: выбор типа обучения; алгоритма оптимизации и установку параметров обучения.

В пакете доступно три вида обучения: пакетное, интерактивное и минипакетное (которое можно рассчитать автоматически или настроить, определив количество записей).

В качестве алгоритма оптимизации может быть выбран метод масштабируемых сопряжённых градиентов или метод градиентного спуска.

Регулировке поддаются следующие параметры обучения: начальные значения лямбды и сигмы, центр интервала или его смещение.

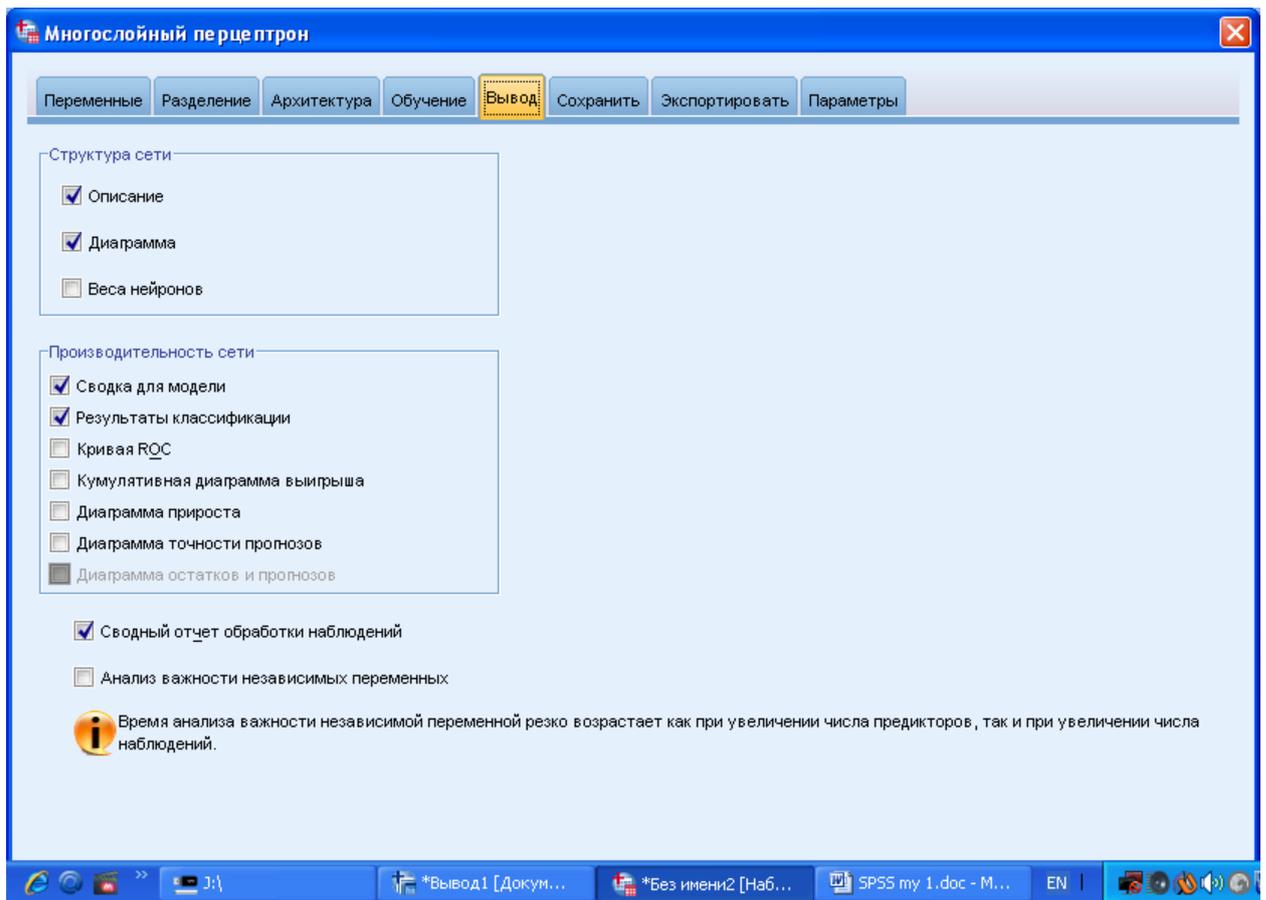


Настройка вывода информации.

При выводе определяется, нужно ли описание, требуется диаграмма, надо ли выводить веса нейронов.

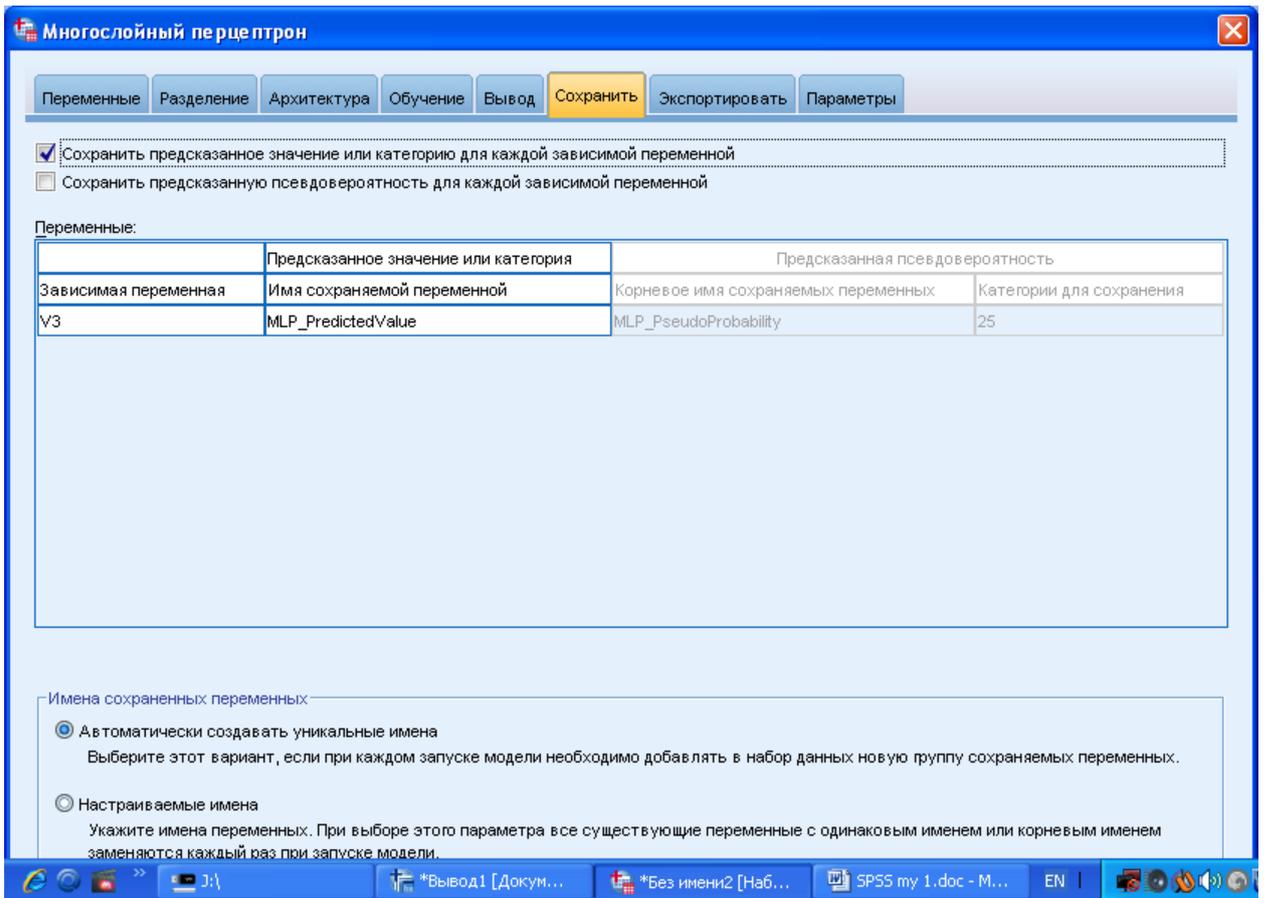
Кроме структуры сети определяется, чем будет характеризоваться производительность сети: нужна ли сводная информация о модели, есть необходимость в результатах классификации, должны быть выведены кривая ROC, кумулятивная диаграмма выигрыша, диаграммы прироста и точности прогнозов, выводится ли диаграмма остатков и прогнозов.

Должен ли вывод информации содержать сводный отчет обработки наблюдений и анализ важности независимых переменных. При этом делается замечание, что время анализа важности независимых переменных резко возрастает при увеличении числа предикторов и при увеличении числа наблюдений.



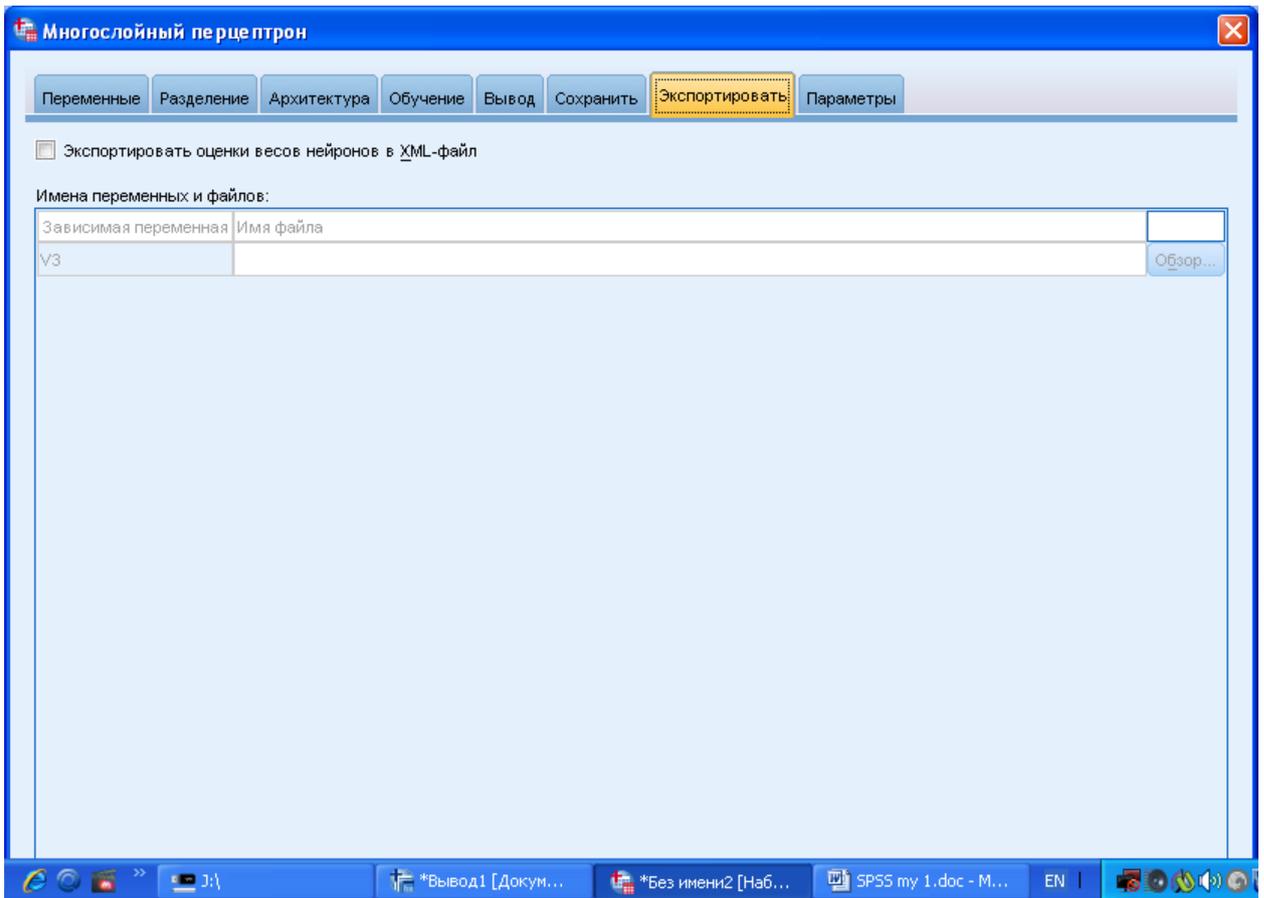
Настройка сохранения результатов.

Сохранить можно предсказанное значение или категорию каждой зависимой переменной, предсказанную псевдовероятность для каждой зависимой переменной, указав имя зависимой переменной и имена сохраняемых переменных. Есть возможность автоматически создавать уникальные имена независимых переменных (если при каждом запуске модели необходимо добавлять в набор данных новую группу сохраняемых переменных) или указывать настраиваемые имена (в этом случае все существующие переменные с одинаковым именем или корневым именем заменяются каждый раз при запуске модели).



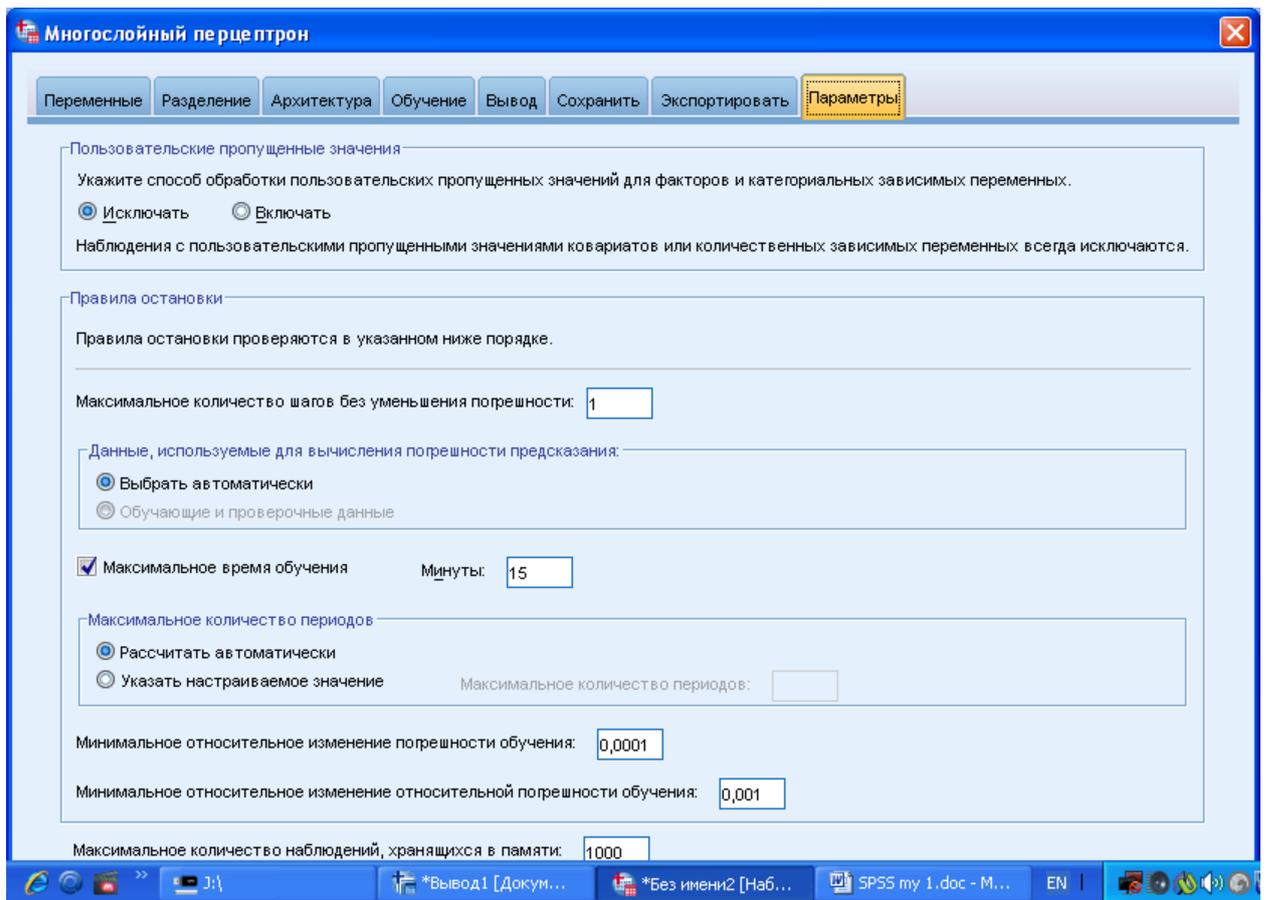
Экспорт результатов.

Экспортировать можно оценки весов нейронов в XML-файл, и имена зависимых переменных в соответствующие файлы.



Настройка дополнительных параметров.

К дополнительным параметрам относятся: особенности обработки пользовательских пропущенных значений; правила остановки (при соблюдении максимального количества шагов без уменьшения погрешности, максимального времени обучения, по достижении максимального количества эпох, при достижении минимальной абсолютной или относительной погрешности обучения, при превышении максимально допустимого количества наблюдений, хранящихся в памяти).



Результаты нейросетевого исследования.

После установки всех необходимых параметров нажатием клавиши Enter запускается нейросетевое исследование. Результаты его оформляются в виде отчёта.

Программа исследования.

```

GET DATA
  /TYPE=TEXT
  /FILE="D:\2013н\1\+Лекции 4 v3 220513\Занятие 6 060613\ПЗ6\Примеры для
исследований\Пример классификации 1v0.txt"
  /DELCASE=LINE
  /DELIMITERS="\t"
  /ARRANGEMENT=DELIMITED
  /FIRSTCASE=1
  /IMPORTCASE=ALL
  /VARIABLES=
  V1 A3
  V2 A4
  V3 A1.
CACHE.
EXECUTE.
DATASET NAME Наборданных1 WINDOW=FRONT.
CODEBOOK V1 [n] V2 [n] V3 [n]
  /VARINFO POSITION LABEL TYPE FORMAT MEASURE ROLE VALUELABELS
MISSING ATTRIBUTES

```

/OPTIONS VARORDER=VARLIST SORT=ASCENDING MAXCATS=200
 /STATISTICS COUNT PERCENT MEAN STDDEV QUARTILES.

Кодовая книга

Примечания

Вывод создан	06-JUN-2013 21:49:50	
Комментарии	D:\2013н\1\+Лекции 4 v3 220513\Занятие 6 060613\П36\Примеры для исследований\Пример классификации 1v0.txt	
Данные	Наборданных1	
Входная	Активный набор данных	<нет>
	Фильтр	<нет>
	Толщина	<нет>
	Расщепить файл	<нет>
	Кол-во строк в рабочем файле данных	99
Редактор синтаксиса	CODEBOOK V1 [n] V2 [n] V3 [n] /VARINFO POSITION LABEL TYPE FORMAT MEASURE ROLE VALUELABELS MISSING ATTRIBUTES /OPTIONS VARORDER=VARLIST SORT=ASCENDING MAXCATS=200 /STATISTICS COUNT PERCENT MEAN STDDEV QUARTILES.	
Ресурсы	Процессорное время	00:00:00,08
	Время вычислений	00:00:00,20

[Наборданных1]

V1

		Значения	Частоты	Процент
Стандартные атрибуты	Положение	1		
	Метки	<нет>		
	Тип	Строка		
	Формат	A3		

	Измерение	Номинальные		
	Роль	Входная		
	10		1	1,0%
	100		4	4,0%
	110		3	3,0%
	120		2	2,0%
	130		1	1,0%
	170		7	7,1%
	180		7	7,1%
	190		7	7,1%
	20		2	2,0%
	200		7	7,1%
Допустимые значения	210		7	7,1%
	220		7	7,1%
	230		7	7,1%
	30		3	3,0%
	40		4	4,0%
	50		5	5,1%
	60		6	6,1%
	70		7	7,1%
	80		6	6,1%
	90		5	5,1%
	X		1	1,0%

V2

		Значения	Частоты	Процент
	Положение	2		
	Метки	<нет>		
Стандартные атрибуты	Тип	Строка		
	Формат	A4		
	Измерение	Номинальные		
	Роль	Входная		
Допустимые значения	140		1	1,0%
	150		2	2,0%
	160		3	3,0%
	170		4	4,0%
	180		5	5,1%
	190		6	6,1%
	200		7	7,1%

210		6	6,1%
220		5	5,1%
230		4	4,0%
240		3	3,0%
250		1	1,0%
2550		1	1,0%
260		1	1,0%
30		7	7,1%
40		7	7,1%
50		7	7,1%
60		7	7,1%
70		7	7,1%
80		7	7,1%
90		7	7,1%
Y		1	1,0%

V3

		Значения	Частоты	Процент
Стандартные атрибуты	Положение	3		
	Метки	<нет>		
	Тип	Строка		
	Формат	A1		
	Измерение	Номинальные		
	Роль	Входная		
	A		49	49,5%
Допустимые значения	B		49	49,5%
	Z		1	1,0%

SUMMARIZE

/TABLES=V1 V2 BY V3

/FORMAT=VALIDLIST NOCASENUM TOTAL LIMIT=100

/TITLE='Сводка для наблюдений'

/MISSING=VARIABLE

/CELLS=COUNT.

Сводка для наблюдений.

Примечания

Вывод создан	06-JUN-2013 21:53:04
Комментарии	

		D:\2013н\1\+Лекции 4 v3 220513\Занятие 6 060613\ПЗ6\Примеры для исследований\Пример классификации 1v0.txt	
Ввод	Данные	Набор данных 1	
	Активный набор данных	<нет>	
	Фильтр	<нет>	
	Толщина	<нет>	
	Расщепить файл	<нет>	
	Кол-во строк в рабочем файле данных		99
Пропущенные значения	Определение пропущенных	Для каждой зависимой переменной в таблице, определенные пользователем пропущенные значения зависимой и всех группирующих переменных трактуются как пропущенные. Наблюдения, использованные для каждой таблицы, не содержат пропущенных значений ни в одной из независимых переменных, и не все зависимые переменные содержат пропущенные значения.	
	Используемые наблюдения	SUMMARIZE /TABLES=V1 V2 BY V3 /FORMAT=VALIDLIST NOCASENUM TOTAL LIMIT=100 /TITLE='Сводка для наблюдений' /MISSING=VARIABLE /CELLS=COUNT.	
Редактор синтаксиса			
Ресурсы	Процессорное время		00:00:00,05
	Время вычислений		00:00:00,05

Сводка обработки наблюдений^a

Наблюдения					
Включенные		Исключенные		Итого	
N	Процент	N	Процент	N	Процент

V1 * V3	99	100,0%	0	0,0%	99	100,0%
V2 * V3	99	100,0%	0	0,0%	99	100,0%

а. Ограничено первыми 100 наблюдениями

Сводка для наблюдений^а

				V1	V2
V3	A	Итого	N	49	49
		1		230	30
		2		230	40
		3		230	50
		4		230	60
		5		230	70
		6		230	80
		7		230	90
		8		220	30
		9		220	40
		10		220	50
		11		220	60
		12		220	70
		13		220	80
		14		220	90
		15		210	30
		16		210	40
		17		210	50
		18		210	60
		19		210	70
		20		210	80
		21		210	90
		22		200	30
		23		200	40
		24		200	50
		25		200	60
		26		200	70
		27		200	80
		28		200	90
		29		190	30
		30		190	40
		31		190	50
		32		190	60

		33	190	70
		34	190	80
		35	190	90
V3	A	36	180	30
		37	180	40
		38	180	50
		39	180	60
		40	180	70
		41	180	80
		42	180	90
		43	170	30
		44	170	40
		45	170	50
		46	170	60
		47	170	70
		48	170	80
		49	170	90
		Итого	N	49
		1	70	140
		2	80	150
		3	90	160
		4	100	170
		5	110	180
		6	120	190
		7	130	200
		8	60	150
		9	70	160
	B	10	80	170
		11	90	180
		12	100	190
		13	110	200
		14	120	210
		15	50	160
		16	60	170
		17	70	180
		18	80	190
		19	90	200
		20	100	210
		21	110	220
V3	B	22	40	170

23	50	180		
24	60	190		
25	70	200		
26	80	210		
27	90	220		
28	100	230		
29	30	180		
30	40	190		
31	50	200		
32	60	210		
33	70	220		
34	80	230		
35	90	240		
36	20	190		
37	30	200		
38	40	210		
39	50	220		
40	60	230		
41	70	240		
42	80	250		
43	10	200		
44	20	210		
45	30	220		
46	40	230		
47	50	240		
48	60	2550		
49	70	260		
Z	Итого	N	1	1
	1	X	Y	
Итого	N		99	99

а. Ограничено первыми 100 наблюдениями

Практикум.

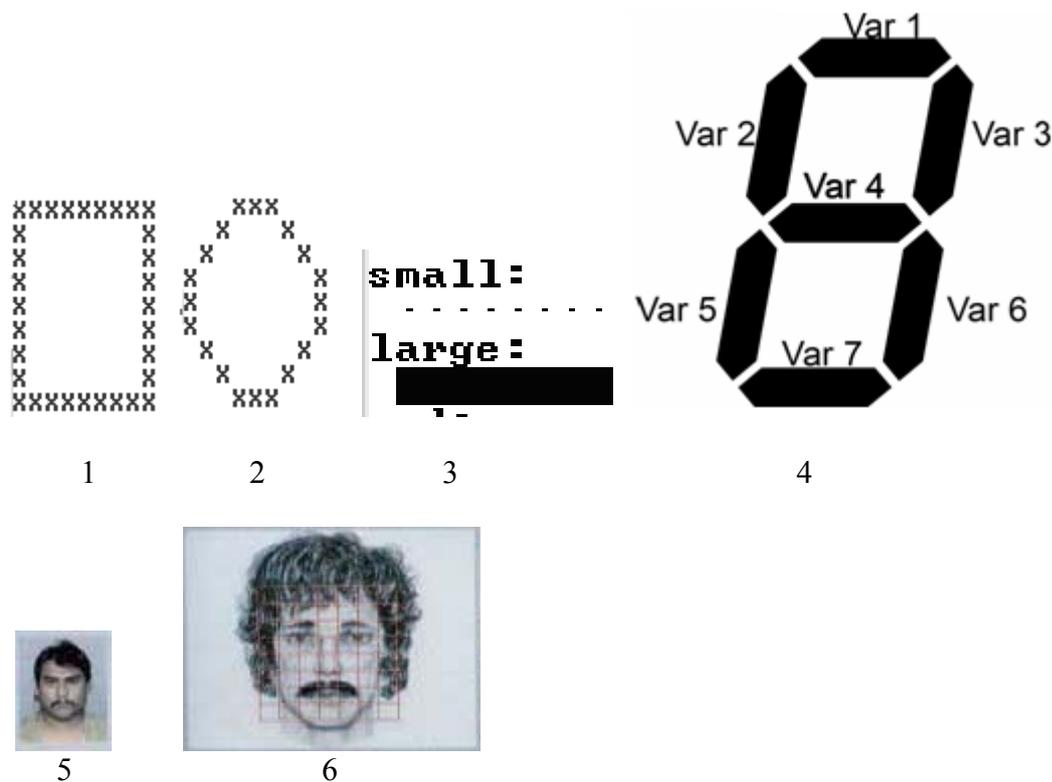
ПЗ1. Нейросетевая декомпозиция решаемой задачи.

Типовое представление данных в нейронных сетях.

Нейросети наиболее приспособлены к решению широкого круга задач, так или иначе связанных с обработкой образов.

Для решения задач, связанных с обработкой образов, могут использоваться либо графические, либо – псевдографические способы отображения.

Графические способы отображения информации.



Где 1, 2, 4 – псевдографика;
3 – термометр;
5, 6 – графика.

Табличный способ отображения информации.

Этот способ более универсальный, чаще других используется для решения различных задач. Информация в этом случае представляется в виде таблицы, в которой каждая колонка отводится для фиксации значений определённой переменной, а каждая строка характеризует реализацию – взаимосвязанную группу переменных, определяющих отдельные наблюдения. Первая строка такой таблицы в пакете BrainMaker отводится для разметки переменных. Во второй строке размещаются названия переменных:

Input Gold	Input Yen	Input DM	Input Gold1	Input Yen1	Input DM1	Input Gold2	Input Yen2	Input DM2	Pattern Yen_F
380	158	3.66	390	150	4.02	375	150	3.88	175
390	150	4.02	375	150	3.88	370	150	3.92	158
390	157	3.54	378	150	4.18	381	150	3.78	170
378	150	4.18	381	150	3.78	365	150	3.89	157
392	158	3.67	375	150	4.11	380	150	3.68	180
375	150	4.11	380	150	3.68	372	147	3.78	158
395	170	3.78	374	152	3.88	384	150	3.72	180
374	152	3.88	384	150	3.72	366	146	4.32	170
401	176	3.66	376	159	3.76	389	150	3.87	180
376	159	3.76	389	150	3.87	373	148	4.08	176

Разметка переменных обычно производится в процессе подготовки нейронной сети.

Подготовка данных для обработки в нейронной сети.

Перед тем, как данные будут использованы в сети, они должны быть определенным образом подготовлены, при этом могут быть выполнены такие операции, как:

- отбраковка входных данных
- поиск оптимальных комбинаций входных переменных
- определение цикличности, корреляции и нелинейных зависимостей
- подавление незначимых переменных
- понижение размерности исходных данных
- масштабирование входных и выходных данных (в том числе шкалирование по минимальному/максимальному значениям и по среднему/стандартному отклонению);
- перекодирование переменных с номинальными значениями (например, Пол={Муж,Жен}), в том числе по методу 1-из-N кодирования.
- проведение работы с пропущенными данными
- оценка чувствительности данных

Эти операции не являются обязательными для исследования с помощью нейронных сетей, и называются «препроцессорная подготовка данных». Аналогичная работа с результатами проведенного нейросетевого исследования называется «постпроцессинг». Препроцессинг и постпроцессинг могут выполняться либо с помощью специальных программ нейропакета, либо на посторонних программных средствах, например таких, как Excel.

Пример препроцессорной подготовки данных представлен в работе [5]:

«Первый шаг при разработке нейросети: определить, что же она должна делать. Этот вопрос важнее, чем кажется на первый взгляд. Предположим, наша нейросеть предсказывает наступление какого-либо события с погрешностью 20%. Сегодня акции стоят 95 долл., а завтра будут стоить 100. Что может быть спрогнозировано с большей точностью: абсолютная цена акций, изменение цены или направление движения рынка? В первом случае прогноз будет в пределах 80-120 долл. При прогнозе изменения цены результат будет находиться в интервале 4-6 долл., что соответствует абсолютной цене 99-101 долл. Ну, а предугадать направление движения рынка вообще можно безошибочно. Таким образом, результат зависит уже от того, как сформулирована цель разработки нейросети...

...Чтобы прогноз был более точным, имеет смысл использовать не абсолютные значения BD100, а только их изменения по сравнению с предыдущим днем. То же относится и к величинам price1, price2. Итак, добавим три новых столбца BD100D, price1D, price2D с приращениями абсолютных значений. Поскольку данные в столбцах с

изменениями цен получаются в результате вычитания значений предыдущего дня из значений текущего, первая строка будет содержать некорректные значения. Поэтому в дальнейшем мы ее удалим. Значения в столбцах index1D, index2D, utilD, transpD уже записаны в виде приращений.

... Для уменьшения случайного шума сгладим BD100D с помощью скользящего среднего (Moving Average). Для сглаживания рекомендуется использовать интервал, равный половине цикла... и т.д.».

Примеры обучающих наборов данных.

Данные в нейронной сети используются прежде всего – для обучения. С этой целью они оформляются в виде обучающей выборки, или обучающего набора данных.

Например, сеть необходимо обучить классификации на два класса по косвенным признакам или обучить прогнозированию.

Примерами таких задач могут служить следующие:

- «Мужчина/женщина», или «Студент/преподаватель», или «Студенты живущие дома/в общежитии»
- «Возможность тренировки парашютистов»
- «Ирисы Фишера»
- «Как выбирают американских президентов»
- «Выбор оптических линз»
- «Ассоциативный поиск текстовой информации»
- «Оценка кредитоспособности»
- «Уровень развития демократии в разных странах»
- «Прогнозирование значений инфляции»
- «Экономико-политические процессы в исламском мире»

Мужчина/женщина.

В задаче «Мужчина/женщина» в общем виде обучающий набор данных может представлять собой таблицу вида:

Информационные (inf)	Исходные показатели (input)				Результирующий (pattern)
	№ п/п	Готовите ли вы дома пищу	Как часто вы убираете квартиру	Сколько времени в неделю вы тратите на ремонт автомобиля	
1	нет	редко	3 часа	да	м
2	Да	всегда	0	нет	ж
...

В качестве исходных показателей лучше использовать переменные, получающие своё значение при ответе на косвенные вопросы.

Примером косвенного вопроса в задаче «Мужчина/женщина» может служить вопрос «Носите ли Вы дома халат», однако вопросы «Носите ли вы дома юбку» или «Приходится ли Вам по утрам бриться» косвенными считаться не могут.

Таблица содержит 3 вида данных: информационные (inf) – например, «№ п/п»; исходные данные (input); результирующий показатель (pattern).

Когда сеть обучена, для определения пола посетителя нужно задать только информационные вопросы, по ответам на которые сеть принимает решение о поле ответившего на вопросы.

Возможность тренировки парашютистов [9].

Принимая во внимание опасность данного вида спорта, тренировка возможна не при любой погоде.

Температура	Влажность	Ветер	Облачность	Тренировка
Жарко	Высокая	Нет	Солнце	Нет
Жарко	Высокая	Да	Солнце	Нет
Жарко	Высокая	Нет	Облачность	Да
Норма	Высокая	Нет	Дождь	Да
Холодно	Норма	Нет	Дождь	Да
Холодно	Норма	Да	Дождь	Нет
Холодно	Норма	Да	Облачность	Да
Норма	Высокая	Нет	Солнце	Нет
Холодно	Норма	Нет	Солнце	Да
Норма	Норма	Нет	Дождь	Да
Норма	Норма	Да	Солнце	Да
Норма	Высокая	Да	Облачность	Да
Жарко	Норма	Нет	Облачность	Да
Норма	Высокая	Да	Дождь	Нет

В этой таблице Температура, Влажность, Ветер, Облачность – это исходные (объективные) показатели, на основе которых принимается решение, можно ли проводить тренировку парашютистов (прыжки) – так называемые «входные переменные».

А колонка «Тренировка» – это решение руководителя (тренера), «выходные, или целевые (target)» переменные.

Пример ирисов Фишера [12].

Господин Фишер вырастил 3 разновидности ирисов: Setosa, Virginic, Versicolor.

Цветки этих ирисов отличаются размерами: (длина (length), ширина (wide)) листьев (petal) и чашелистников (sepal) – всего 4 параметра. При снятии характеристик с 20 случайно отобранных цветков образуется таблица:

№	SEPL	SEPW	PETL	PETW	IRISTYPE
1	5,0	3,3	1,4	0,2	SETOSA
2	6,4	2,8	5,6	2,2	VIRGINIC
4	6,7	3,1	5,6	2,4	VIRGINIC
5	6,3	2,8	5,1	1,5	VIRGINIC
6	4,6	3,4	1,4	0,3	SETOSA
7	6,9	3,1	5,1	2,3	VIRGINIC
10	4,6	3,6	1,0	0,2	SETOSA
13	6,5	3,0	5,2	2,0	VIRGINIC
15	6,5	3,0	5,5	1,8	VIRGINIC
16	5,8	2,7	5,1	1,9	VIRGINIC
17	6,8	3,2	5,9	2,3	VIRGINIC
18	5,1	3,3	1,7	0,5	SETOSA
20	6,2	3,4	5,4	2,3	VIRGINIC

Необходимо провести классификацию отобранных наблюдений и определить, к какому типу относится цветок, обладающий следующими характеристиками: 6,0 3,5 5,4 2,7.

Данные об ирисах имеют несколько интересных особенностей:

- Один из классов (Iris Setosa) линейно отделим от других. Однако, другие два класса нельзя разделить линейно.
- Классы Versicolor и Virginica пересекаются, поэтому важно достигнуть наилучшего уровня классификации.

- Среди четырех переменных существует некоторая избыточность, поэтому можно достигнуть хорошего решения, пользуясь только тремя или двумя переменными, однако неизвестно, какие именно переменные нужно выбрать.

Здесь таблица составлена неправильно – она содержит характеристики только двух разновидностей. Желательно, чтобы таблица содержала примерно одинаковое количество строк, характеризующих различные типы цветков.

№	SEPALLEN	SEPALWID	PETALLEN	PETALWID	IRISTYPE
1	5,0	3,3	1,4	0,2	SETOSA
2	6,4	2,8	5,6	2,2	VIRGINIC
3	6,5	2,8	4,6	1,5	VERSCOL
4	6,7	3,1	5,6	2,4	VIRGINIC
5	6,3	2,8	5,1	1,5	VIRGINIC
6	4,6	3,4	1,4	0,3	SETOSA
7	6,9	3,1	5,1	2,3	VIRGINIC
8	6,2	2,2	4,5	1,5	VERSCOL
9	5,9	3,2	4,8	1,8	VERSCOL

Как выбирают американских президентов [11].

Имеется таблица данных с результатами 31-ой предвыборной ситуации (с 1860 по 1980 г.). Для каждого выбора в таблице содержатся данные по 12-ти бинарным признакам:

1. Правящая партия была у власти более одного срока?
2. Правящая партия получила более 50% голосов на прошлых выборах?
3. В год выборов была активна третья партия?
4. Была серьезная конкуренция при выдвижении кандидата от правящей партии?
5. Кандидат от правящей партии был президентом в год выборов?
6. Был ли год выборов временем спада или депрессии?
7. Был ли рост среднего национального валового продукта на душу населения более 2,1%?
8. Произвел ли правящий президент существенные изменения в политике?
9. Во время правления были существенные социальные волнения?
10. Администрация правящей партии виновна в серьезной ошибке или скандале?
11. Кандидат от правящей партии – национальный герой?
12. Кандидат от оппозиционной партии – национальный герой?

Также в таблице содержится информация о результатах выборов (победе правящей или оппозиционной партии).

Значения бинарных признаков равны -1 (ответ "нет" для входного признака или победа правящей партии) и 1 (ответ "да" для входного признака или победа оппозиции).

Нейронные сети, обученные на этой таблице данных, уверенно предсказывали результаты вторых выборов Рейгана, победу Буша над Дукакисом, обе победы Клинтона.

Рекомендации по выбору оптических линз [11].

Если на основе этой таблицы провести обучение нейронной сети, то обученная сеть может при предоставлении ей информации о Возрасте, Диагнозе, Астигматизме, Индексе чувствительности дать рекомендацию о предпочтительном типе оптических линз для данного человека.

<u>Возраст</u>	<u>Диагноз</u>	<u>Астигматизм</u>	<u>Индекс чувствительности</u>	<u>Рекомендация</u>
Молодой	Близорукость	Нет	Пониженный	Нет
Молодой	Близорукость	Нет	Нормальный	Мягкие
Молодой	Близорукость	Да	Пониженный	Нет
Молодой	Близорукость	Да	Нормальный	Жесткие
Молодой	Дальнозоркость	Нет	Пониженный	Нет
Молодой	Дальнозоркость	Нет	Нормальный	Мягкие
Молодой	Дальнозоркость	Да	Пониженный	Нет
Молодой	Дальнозоркость	Да	Нормальный	Жесткие
Средний	Близорукость	Нет	Пониженный	Нет
Средний	Близорукость	Нет	Нормальный	Мягкие
Средний	Близорукость	Да	Пониженный	Нет
Средний	Близорукость	Да	Нормальный	Жесткие
Средний	Дальнозоркость	Нет	Пониженный	Нет
Средний	Дальнозоркость	Нет	Нормальный	Мягкие
Средний	Дальнозоркость	Да	Пониженный	Нет
Средний	Дальнозоркость	Да	Нормальный	Нет
Пожилой	Близорукость	Нет	Пониженный	Нет
Пожилой	Близорукость	Нет	Нормальный	Нет
Пожилой	Близорукость	Да	Пониженный	Нет
Пожилой	Близорукость	Да	Нормальный	Жесткие
Пожилой	Дальнозоркость	Нет	Пониженный	Нет
Пожилой	Дальнозоркость	Нет	Нормальный	Мягкие
Пожилой	Дальнозоркость	Да	Пониженный	Нет
Пожилой	Дальнозоркость	Да	Нормальный	Нет

Ассоциативный поиск текстовой информации [13].

Традиционные методы поиска и фильтрации документов были разработаны для библиотечных баз данных ограниченного объема и заранее известной структуры. Создание глобальной сети привело к тому, что число поставщиков информации стало стремительно расти, при том, что публикуемая ими информация не имеет однородной структуры.

Последовавший информационный взрыв стал вызовом стандартным информационным технологиям. Новые масштабы с одной стороны сделали аутсайдерами некоторые ранее конкурентоспособные интеллектуальные технологии, а с другой - стимулировали интенсивные исследования в области статистических методов обработки текстовой информации и новых способов навигации в информационном море.

Нейросети являются перспективным инструментом извлечения статистических закономерностей в текстах, и использования этих закономерностей для прецизионной фильтрации документов.

Примером может служить система категоризации текстов, выделяющая в больших массивах документов базовые *тематические категории*.

Таким образом можно представить содержание любого документа вектором в автоматически формируемом нейросетью пространстве категорий, поиск в котором уже не зависит от конкретных ключевых слов, а определяется именно содержанием документов.

Частным случаем документов являются отдельные слова. Табл. 1 иллюстрирует какие ассоциации у обученной нейросети вызывают различные термины.

Табл. 1 Нейро-ассоциации, полученные при обучении на 8000 аннотаций статей научных конференций общества SPIE

Термин	Слова, ассоциируемые с термином (в порядке убывания степени близости)
THE	OF IN FROM THAT ARE ON TO WHICH FOR AND AS TWO IS BOTH A BY IT WITH BETWEEN ALSO AN RESULTS HAS WITHIN INTO BE TIME USED OR
NEURAL	LEARNING CLASSIFIERS UNSUPERVISED TRAINED BACK-PROPAGATION SUPERVISED NEURONS WEIGHTS TRAINING HIDDEN HOPFIELD BACKPROPAGATION NETWORK IMPULSIVE NETS FEEDFORWARD PREDICTOR NETWORKS TEXTURAL SPEAKER TELEPHONE PERCEPTRON LEARN AMBIGUITIES DIGITS MULTIDIMENSIONAL BP MLP CLASSIFIER
CANCER	ORGANS LESIONS THERAPY TUMOR VIVO CAM PHOTSENSITIZERS TUMORS RAT MOUSE PATIENTS AUTOFLUORESCENCE ADMINISTRATION NECROSIS SENSITIZERS VASCULAR RESECTION ADMINISTERED VITRO CLEARANCE INCUBATION PP ACUTE DRUG BALLOON PROSTATE SKIN DISORDERS EPITHELIAL

Как видно из этих примеров, к артиклю «the» ближайшими оказываются служебные слова: артикли, союзы и т.д. К словам же «neural» и «cancer» ближайшими являются слова из той же предметной области.

Этот пример иллюстрирует эффективное нейросетевое **сжатие информации** – текст характеризуется уже не частотами всех входящих в него слов, которых могут быть многие тысячи, а относительно небольшим числом смысловых категорий.

Векторное представление смысла текстов позволяет количественно определить их тематическую близость, построить системы ассоциативной выборки информации из больших архивов текстовых документов.

Такие системы могли бы существенно облегчить поиск юридических документов, патентной информации и безусловно окажутся востребованы в Internet-приложениях.

Построение модели поведенческого скоринга [5].

В данном примере рассмотрена схема построения нейросетевой модели для задачи поведенческого скоринга. Поведенческий скоринг (behavior scoring) используется для принятия решений по уже выданным кредитам.

Основные решения, принимаемые с использованием поведенческого скоринга, можно сформулировать следующим образом:

- Предложение новых услуг и улучшение уже предоставляемых услуг.
- Решение, выдавать ли кредитную карту заново после истечения срока действия, или нет.
- Меньший стартовый кредитный лимит или максимальное значение кредита на кредитной карточке.
- Более строгий сбор платежей с нарушителей или отправка данных о них в агентства сбора платежей.
- Повышение кредитного лимита.
- Помещение под наблюдение ввиду потенциальных мошеннических действий и т.д.

В данном примере необходимо оценить кредитоспособность существующих заёмщиков на основании данных о графике погашения кредитов и динамики движения средств на счетах клиента.

Структура данных

Каждого клиента будем характеризовать 22 признаками. 20 переменных относятся к анкетным данным, которые заполняются в анкете для получения кредита. К этим переменным относятся:

- Текущий баланс счета
 - Продолжительность в мес
 - Назначение кредита
 - Сумма кредита
 - Объем сбережений
 - Время работы на данном рабочем месте
 - Семейное положение/пол
 - Длительность проживания по текущему адресу
 - Возраст в годах
 - Число предыдущих кредитов в банке
 - Должность
- и другие.

На основании перечисленных факторов все клиенты подразделяются на “хороших” и “плохих”. Разбиение на эти группы записано в результирующей переменной - *Кредитоспособность* (Creditability).

Всего имеются данные по 1000 клиентов. При этом, 30% относятся к “плохим”, а остальные 70% - к “хорошим”. Процент невыплат по всей совокупности данных около 3% (данная величина относится к одному месяцу). Элемент таблицы данных показан на рис.1.

	1 Кредитоспособность	2 Текущий баланс счета	3 Продолжи- тельность в мес	4 Выплаты по предыдущим кредитам	5 Назначение кредита сумма кредита
1	плохой	нет текущего счета	36	нет проблем	переподготовка
2	хороший	нулевой баланс	48	неуверенное	переподготовка
3	плохой	>=200	36	не было кредитов	подержанная м.
4	хороший	нет текущего счета	24	выплачены	новая машина
5	хороший	>=200	24	не было кредитов	переподготовка
6	хороший	нулевой баланс	12	не было кредитов	переподготовка
7	плохой	нет текущего счета	30	не было кредитов	подержанная м.
8	хороший	нулевой баланс	15	выплачены	предметы мебе.
9	хороший	>=200	15	выплачены	предметы мебе.
10	плохой	нулевой баланс	27	выплачены	предметы мебе.
11	плохой	нулевой баланс	24	не было кредитов	подержанная м.
12	плохой	нет текущего счета	18	не было кредитов	ремонт
13	плохой	нулевой баланс	36	нет проблем	переподготовка
14	хороший	>=200	6	нет проблем	переподготовка
15	хороший	нет текущего счета	12	не было кредитов	другой
16	хороший	>=200	42	не было кредитов	предметы мебе.
17	хороший	нет текущего счета	48	не было кредитов	новая машина
18	плохой	нулевой баланс	24	с проблемами	ремонт
19	хороший	нулевой баланс	24	выплачены	другой
20	хороший	нулевой баланс	48	не было кредитов	бизнес

Фрагмент исходной таблицы данных.

Классификация политических режимов стран на основе индексов развития новых медиа

Поставлена задача классификации политических режимов стран на основе данных о развитии интернета и локальных условий регламентации медиа.

Предполагается, что обученная сеть способна определять тип политического режима (уровень развития демократии) на основе анализа того, насколько недовольные пользователи интернета могут выражать свои мысли в сети

При проведении исследования в выборку были включены данные об уровне развития демократии, количестве пользователей интернета, уровне удовлетворённости жизнью и индексы свободы слова.

Данные были взяты из статистических БД, созданных для Secondary Data Analysis.

	A	B	C	D	E	F	G	H	I	J
	Country Name	NNSET	Democr acy Index 2011	Democr acy Index 2011 Category	IT.NET.USER_2010	IT.NET.USER_2008	Life Satisfaction 2009	Happy Planet Index 2009	Free Press 2010	Free Press 2009
1										
2	Albania	Select	5,81	Hybrid regime	1441927,80	759081,32	5,47	47,91	21,50	21,75
3	Algeria	Select	3,44	Authoritarian regime	4433526,00	3504773,25	5,59	51,23	47,33	49,56
4	Angola	Train	3,32	Authoritarian regime	1908191,20	829746,34	4,27	26,78	28,50	36,50
5	Argentina	Select	6,84	Flawed democracy	14548455,36	11164731,06	7,14	58,95	16,35	11,33
6	Armenia	Train	4,09	Hybrid regime	1360511,68	191211,30	5,03	48,28	27,50	31,13
7	Australia	Train	9,22	Full democracy	16923971,84	15418952,64	7,88	36,64	5,38	3,13
8	Austria	Train	8,49	Full democracy	6102179,19	6078674,03	7,80	47,69	0,50	3,00
9	Azerbaijan	Select	3,15	Authoritarian regime	4226380,18	1527566,88	5,28	41,21	56,38	53,50
10	Bangladesh	Select	5,86	Hybrid regime	5501608,85	3636957,50	5,25	54,09	42,50	37,33
11	Belarus	Select	3,16	Authoritarian regime	3041748,46	2226558,70	5,83	35,67	57,00	59,50
12	Belgium	Train	8,05	Full democracy	8034049,50	6997430,22	7,61	45,36	4,00	2,50
13	Benin	Train	6,06	Flawed democracy	277001,62	154585,63	3,02	24,58	19,00	16,00
14	Bhutan	Train	4,57	Hybrid regime	98727,84	45939,28	6,13	58,50	17,75	15,75
15	Bolivia	Train	5,84	Hybrid regime	1985969,80	1041679,87	6,50	49,35	28,13	24,17
16	Bosnia and Herzegovina	Train	5,24	Hybrid regime	1955277,48	1308125,24	5,90	44,96	13,50	10,50
17	Botswana	Select	7,63	Flawed democracy	120416,70	122176,38	4,70	20,85	17,50	15,50
18	Brazil	Select	7,12	Flawed democracy	79245740,06	64799077,08	7,57	61,01	16,60	15,88
19	Russia	Train	6,78	Flawed democracy	3464679,68	3011477,44	5,47	47,04	19,00	15,61

В качестве выходных категорий была использована переменная «Тип политического режима», зависимыми переменными считаются данные по пользователям, удовлетворенность жизнью и благополучием, индексы свободы слова.

Обученная сеть способна определять тип политического режима (уровень развития демократии) на основе анализа того, насколько недовольные граждане страны могут выражать свои мысли в интернет.

Прогнозирование инфляции.

Целью данной работы являлось прогнозирование значений инфляции с помощью нейронных сетей, основываясь на прошлых данных по инфляции. Методы построения инфляционных ожиданий – одна из важнейших тем в современной макроэкономике. От того, как экономические агенты прогнозируют инфляцию, зависят её действительные значения.

Инфляционные ожидания входят в значительное количество моделей монетарной экономики. Существует множество методов и подходов для моделирования прогнозирования агентами инфляционной динамики, например, learning-модели, различные поведенческие способы и т.д.

Использование нейронных сетей в прогнозировании будущих значений инфляции в перспективе, возможно, позволит более точно моделировать поведение людей, поскольку сами нейронные сети имеют некоторые похожие на человеческое восприятие и мышление способности.

Для данного исследования в качестве рассматриваемых данных была взята статистика из базы МВФ по годовой инфляции (по ценам потребителей) в США, Боливии и России за различные периоды времени.

С помощью нейронных сетей делались и проверялись прогнозы для различных случаев инфляционной динамики, в том числе, и гиперинфляции в Боливии. На основе

полученных и проверенных прогнозов делались выводы о качестве работы нейронных сетей и их пригодности для прогнозирования.

Россия и исламский мир

Цель - рассмотреть политико-экономические аспекты современного исламского мира и произвести сопоставление с Россией.

Задачи исследования:

На основе нейросетевого исследования представить качественные оценки исламского мира, определяемые количественными данными социально-экономических показателей.

Анализ выполнен на основании данных национальных сайтов статистических служб и Всемирного банка.

В России постоянно проживает, по разным оценкам от 15 до 30 млн. мусульман и примерно 5 млн. трудовых мигрантов, исповедующих ислам.

Мусульмане играют значительную роль, их влияние заметно уже сегодня и будет увеличиваться. По прогнозам SIEMS к 2050г. население г. России сократится до 100 млн. человек. Исламская же умма к этому времени может составить более чем половину всего населения России.

Подготовка обучающего набора данных.

Прежде, чем начинать готовить обучающий набор данных, необходимо осуществить постановку задачи и провести её нейросетевую декомпозицию. Задача должна относиться к одному из допустимых для данного нейросетевого пакета типов:

1. Узнавание (Классификация)
2. Кластеризация
3. Регрессия (прогнозирование, предсказание)
4. Понижение размерности

Обучающий набор данных представляет собой набор *наблюдений*, для которых указаны значения входных и выходных *переменных*.

Первый вопрос, который нужно решить, - какие переменные использовать и сколько (и каких) наблюдений собрать. Выбор переменных (по крайней мере первоначальный) осуществляется интуитивно. Ваш опыт работы в данной предметной области поможет определить, какие переменные являются важными. Вы можете произвольно выбирать переменные и отменять предыдущий выбор. Для начала имеет смысл включить все переменные, которые, по Вашему мнению, могут влиять на результат - на последующих этапах можно сократить это множество.

Нейронные сети могут работать с числовыми данными, лежащими в определенном ограниченном диапазоне. Это создает проблемы в случаях, когда данные имеют нестандартный масштаб, когда в них имеются пропущенные значения, и когда данные являются нечисловыми. Числовые данные масштабируются в подходящий для сети диапазон, а пропущенные значения можно заменить на среднее значение (или на другую статистику) этой переменной по всем имеющимся обучающим примерам.

Более трудной задачей является работа с данными нечислового характера. Чаще всего нечисловые данные бывают представлены в виде номинальных переменных типа $Пол = \{Муж, Жен\}$. Переменные с номинальными значениями можно представить в числовом виде. Однако, нейронные сети не дают хороших результатов при работе с номинальными переменными, которые могут принимать много разных значений.

Одним из методов сбора необходимых данных является анкетирование. Для анкетирования необходимо составить вопросник из косвенных вопросов, по ответам на которые, с точки зрения испытуемого, возможно получить решение. При анкетировании необходимо получить ответы в достаточном количестве (обычно опросив не менее 30 человек) по составленному вопроснику.

Во многих реальных задачах приходится иметь дело с не вполне достоверными или недоопределёнными данными. Такие данные могут сопровождаться коэффициентом доверия. Иногда (как в BrainMaker) коэффициент доверия выражается в виде термометра и может существенно уточнять исходные данные.

Значения некоторых переменных могут быть искажены шумом или частично отсутствовать. Если имеется не так много данных, можно включить в рассмотрение случаи с пропущенными значениями (хотя, конечно, лучше этого избегать).

Нейронные сети в целом устойчивы к шумам. Однако у этой устойчивости есть предел. Например, выбросы, т.е. значения, лежащие очень далеко от области нормальных значений некоторой переменной, могут исказить результат обучения. В таких случаях лучше всего постараться обнаружить и удалить эти выбросы (либо удалив соответствующие наблюдения, либо преобразовав выбросы в пропущенные значения).

Задание: Нейросетевая декомпозиция решаемой задачи.

1. По аналогии с приведенными выше постановками задач выберите тему исследования.
2. Сформулируйте цель
3. Определите необходимый характер исходных данных.
4. Начните формирование обучающей выборки.

При необходимости используйте Excel (см. приложение 2) или рекомендации из приложения 3 по поиску информации в Интернет.

ПЗ 2. Работа с нейропакетом BrainMaker.

Практическое знакомство с нейропакетами и принципом действия нейроЭВМ можно осуществить на примере пакета **Brainmaker**.

Пакет Brainmaker предназначен для построения нейронных сетей обратного распространения. Пакет включает в себя программу подготовки и анализа исходных данных (netmaker.exe), программу обучения и запуска нейросетей (brainmak.exe), а также набор утилит широкого назначения. Программный пакет ориентирован на широкий круг задач от создания прогностических приложений до организации систем распознавания образов и нейросетевой памяти.

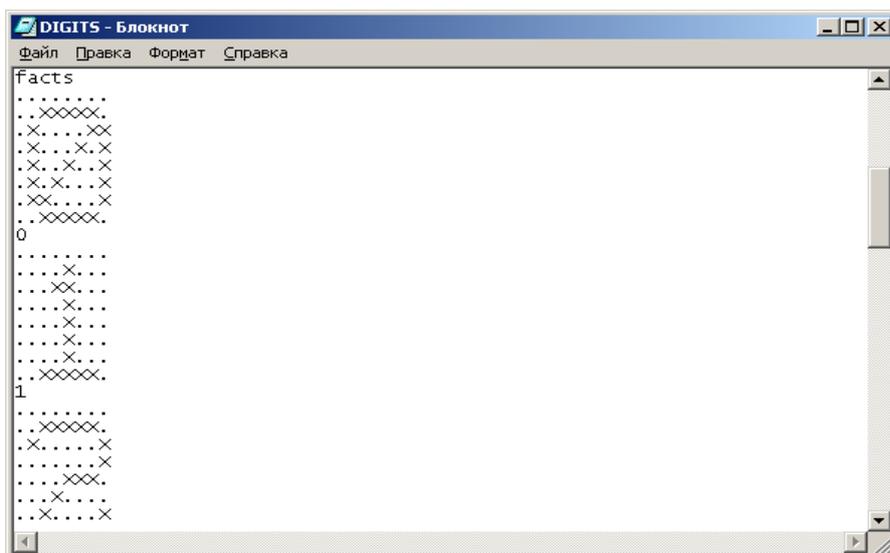
Входные и выходные данные могут быть представлены в программе в числовом и символьном видах, а также в виде матричной графики.

Нейропакет использует 3 основных типа файлов: *файлы определений*, *файлы фактов* и *файлы сети*. Все эти файлы имеют текстовый формат и могут быть созданы и отредактированы в текстовом редакторе типа «Блокнот».

Файл определений содержит всю необходимую информацию об интерфейсе пользователя, типах входных и выходных данных, параметрах обучения и т. п. По умолчанию файл имеет расширение **def**.

Файл фактов содержит обучающие, тестирующие и рабочие факты, которые будут использоваться построенной сетью. По умолчанию они имеют расширения соответственно **fct**, **tst**, **in**.

Файл сети создается программой в процессе обучения и содержит текущие параметры, такие как конфигурацию и веса связей в сети. По умолчанию файл имеет расширение **net**.



Начало обучающей выборки для распознавания десятичных цифр.

Каждый обучающий пример содержит графический образ одного из десятичных символов, выполненных крестиками («x») в поле 8x8 точек (эта часть примера поступает на вход сети), и цифрового обозначения этого символа (т.е. выходного значения Y_0).

Обучающая выборка может быть продолжена – для этого в нее добавляются новые начертания символов. После сохранения отредактированного файла, на нем можно провести обучение сети.

Приведенный в составе нейропакета файл digits.def содержит только по одному графическому образу каждой цифры – этого очень мало, поэтому, нейропакет правильно распознает только достаточно близкие к эталонным начертания цифр.

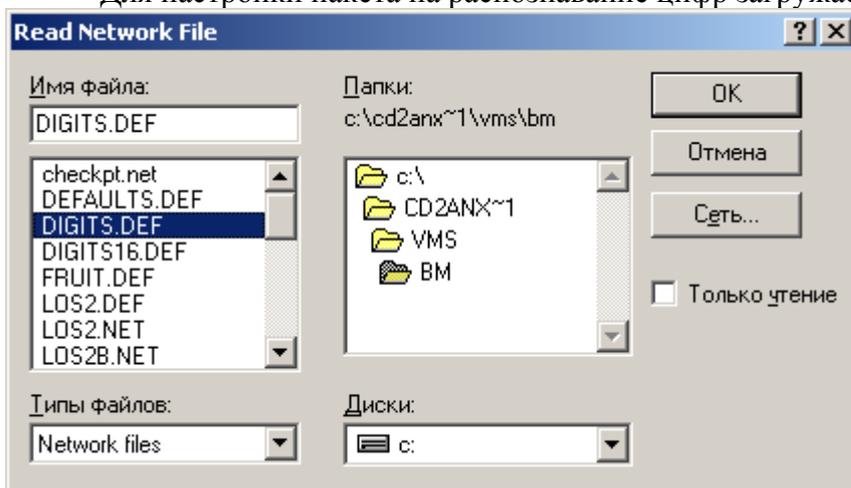
После загрузки в основную память программы brainmak.exe на экран выводится начальное окно пакета:



Начальное окно пакета Brainmaker.

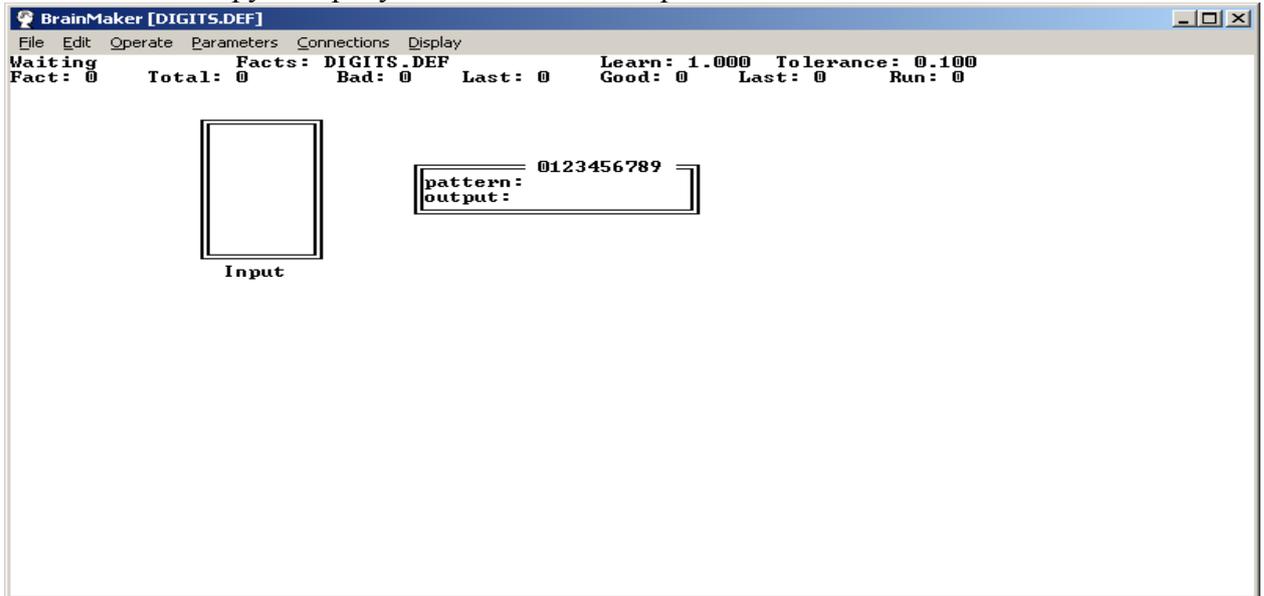
В строке меню окно содержит только опцию “file”, с помощью которой можно либо загрузить в нейропакет сеть, либо выйти из него.

Для настройки пакета на распознавание цифр загружаем файл digits.def.



Окно для выбора загружаемой в пакет нейросети.

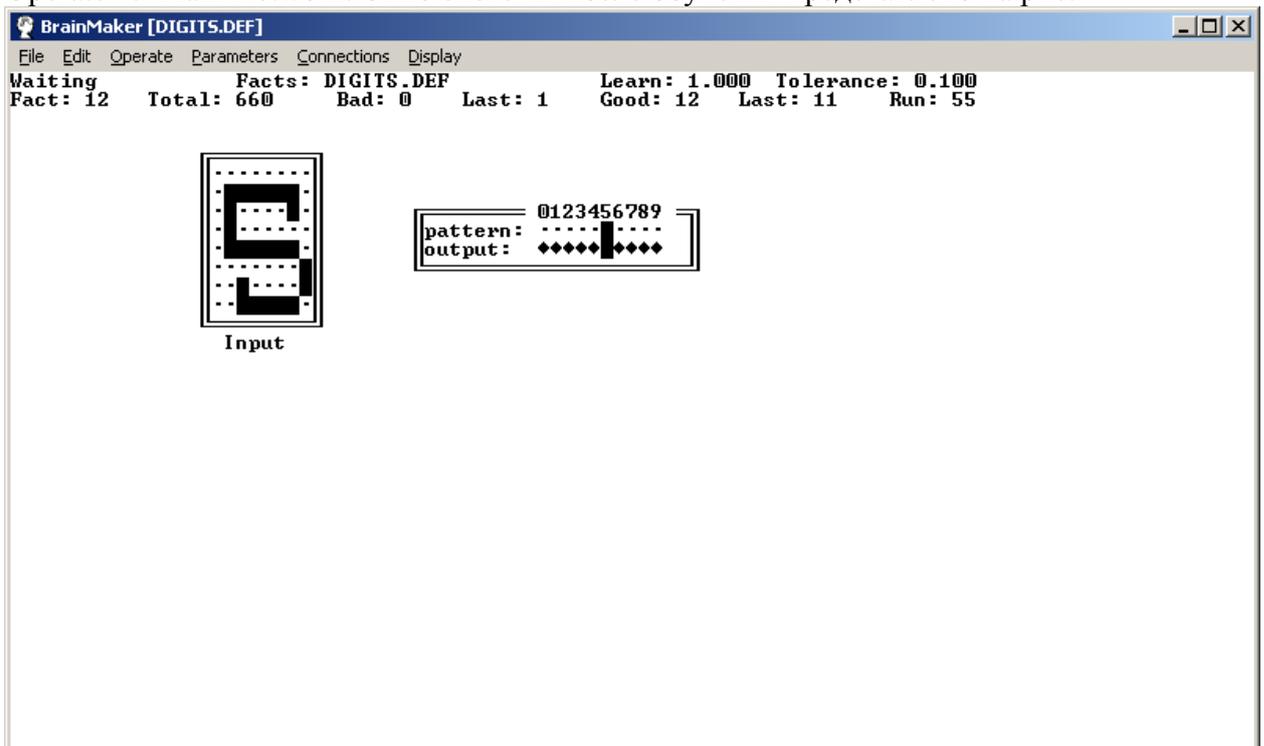
После загрузки требуемой сети окно нейропакета изменяет свой вид:



Окно нейропакета с загруженной сетью.

В рабочей зоне окна появляются такие элементы, как входная матрица (input) и индикатор распознавания (output).

Для работы с пакетом, его надо обучить распознавать образы десятичных цифр: Operate → Train Network. Окно системы после обучения представлено на рис.



Окно обученного нейропакета.

Теперь пакет обучен и готов к работе. В основном меню пакета выбираем: Edit → Edit Network Input или нажимаем клавиши Ctrl+J. Открывается возможность корректировать образ во входной матрице (input) – это заметно по появившемуся в матрице курсору.

Одновременно с коррекцией входного образа, на индикаторе распознавания видно, как воспринимает этот образ нейропакет. Он может считать, что входной образ похож сразу на несколько известных нейропакету цифр. Какой из них нейропакет отдает

предпочтение – видно по насыщенности меток, появляющихся на индикаторе распознавания.

2. Задание по самостоятельному изучению возможностей нейропакета BrainMaker.

BrainMaker - это нейропакет, который может быть использован для решения трудноформализуемых задач, т.е. таких задач, алгоритм решения которых неизвестен.

Нейропакет работает на основе case-based - подхода (т.е. подхода, основанного на примерах). Для решения задачи не требуется знания алгоритма ее решения. Необходимо лишь иметь достаточное количество примеров правильного решения задачи (т.е. исходных данных (input) - и получаемого при этом результата (pattern)). Эти примеры используются для обучения нейронной сети, после чего обученной сети можно предъявлять новые данные, и получать на выходе (output) результат решения задачи.

В общем случае процесс решения задачи с помощью нейропакета состоит из следующих этапов:

1. Формулируется постановка задачи и выделяется набор ключевых параметров, характеризующих предметную область.

Подготовка данных для обучения нейронной сети - процесс достаточно трудоемкий. Необходимо из имеющейся информации выявить только ту, которая каким-то образом (пусть даже неизвестно, как) связана с результирующим показателем. На основе этих данных формируются обучающие примеры (чем больше их количество, тем быстрее будет обучаться нейросеть).

Обучающие (тестовые) примеры оформляются в виде таблицы, каждая строка которой соответствует одному примеру. Колонки таблицы помечаются маркерами: Input (исходный параметр), Pattern (известный (эталонный) результат решения задачи), Annote (пояснения), Not Used (не используется). Исходных параметров может быть несколько.

Таблица записывается в файл.

2. Выбираются параметры нейросети, наиболее подходящие для решения данного класса задач. Обычно нейропакеты имеют проверенные конструкции сетей (нейросетевые парадигмы), которые можно использовать. Но можно сконструировать и свою нейросеть.

3. Проводится обучение нейронной сети - подготовленные тестовые примеры по очереди предъявляются сети, фиксируется полученный сетью результат и сопоставляется с эталонным результатом. Оценивается в % ошибка сети. Если она превышает допустимое значение, проводится корректировка весов связей. Обычно для этого используется алгоритм обучения методом “обратного распространения” (Back-propagation). Но могут быть использованы и другие алгоритмы обучения, часто - значительно более эффективные. Процесс обучения заканчивается, когда ошибка нейросети не превышает заданного уровня.

4. Обученная сеть готова к решению задачи. Если ей предъявить исходные данные, она определит значение результата (output).

Нейропакет BrainMaker содержит следующие компоненты:

brainmak.exe - пакет BrainMaker;

install.exe - программа инсталляции пакета;

netmaker.exe - нейроконструктор;

utility.exe - пакет утилит;

и большое количество тестовых файлов.

При первом обращении к пакету в его окне доступен только пункт меню File с одной функцией: Read Network (чтение сети). Поэтому работу с пакетом необходимо начинать с конструирования тестовых примеров и сети.

BrainMaker содержит подготовленные тестовые примеры, на которых можно ознакомиться с управлением нейропакетом.

Пример 1.

Загружаем в ОП ЭВМ пакет BrainMaker. Для этого устанавливаем курсор мыши на файл brainMak.exe и дважды щелкаем левой кнопкой.

В строке меню выбираем File -> Read Network.

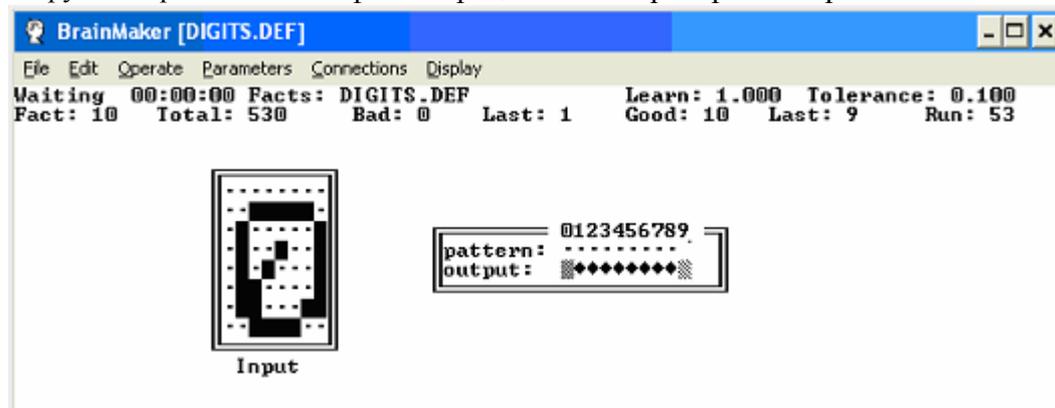
Появляется окно, в котором надо выбрать digits.def. В основном окне пакета появляются новые позиции в строке меню. Выбираем Operate -> Train Network. Начинается обучение сети распознаванию цифр.

В рабочем поле основного окна пакета появляется прямоугольник с названием Input и две горизонтальные шкалы: pattern и output (шкалы имеют 10 позиций, пронумерованные от 0 до 9).

После окончания тренировки нейропакет готов к использованию. Нажимаем клавиши Ctrl+E - очищается прямоугольник Input. После нажатия клавиш Ctrl+J в прямоугольнике Input можно, щелкая мышкой, рисовать любую цифру из диапазона от 0 до 9. На горизонтальной шкале Output при этом оттенками серого отмечаются те позиции, к которым нейросеть склонна отнести создаваемую цифру. Если цифра имеет четкие очертания, схожие с цифрой из эталонного набора, то после ее прорисовки на шкале output остается одна метка, соответствующая распознанной цифре.

Пример работы с нейронной сетью.

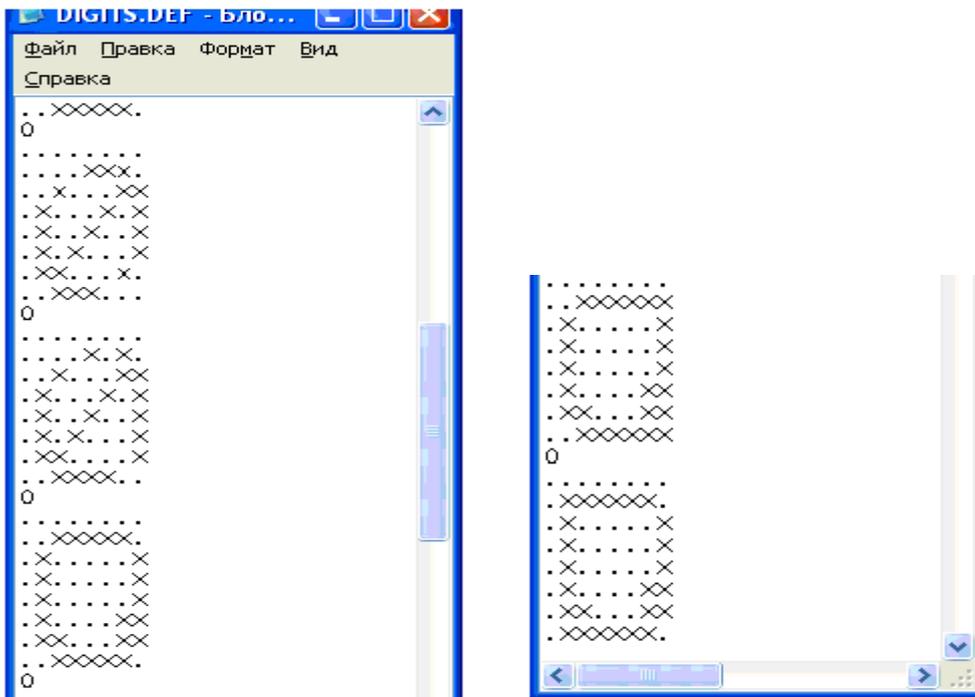
Загрузим первоначальный файл определений и проверим его работоспособность.



Как видим, введенный образ очень похож на цифру ноль, однако неоднозначно определяется программой.

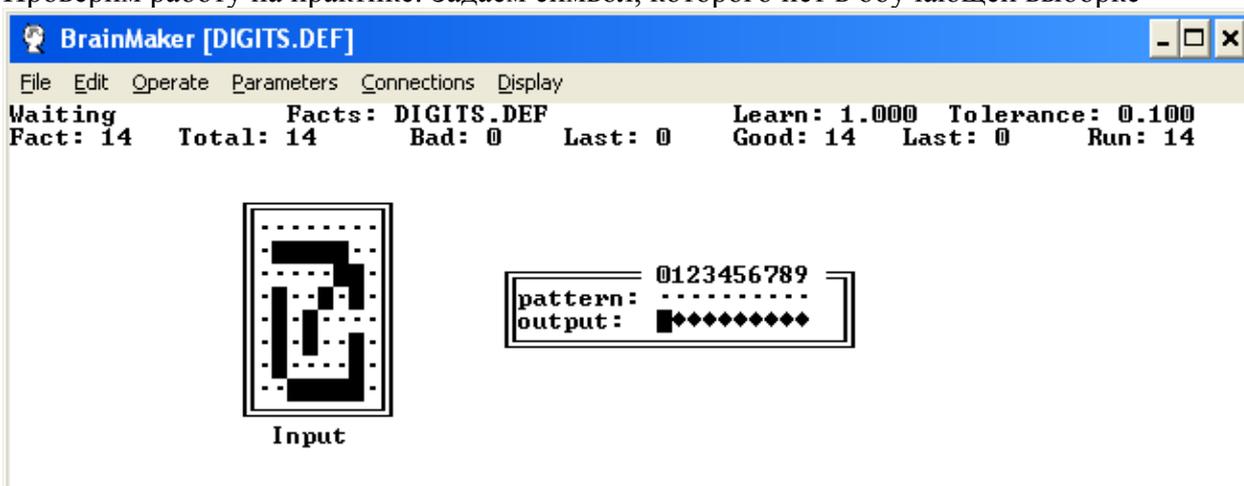
Цифра 0 в обучающей выборке перечёркнута. Если пакету предложить распознать 0 без перечёркивания, он колеблется, 0 это или 9. Это значит, что обучающую выборку надо развить, добавить в неё другие начертания цифр. Для этого нужно с помощью блокнота добавить в конец файла digits.def ещё несколько образов. Затем надо снова обучить пакет на новой обучающей выборке и проверить качество распознавания (проверку надо производить на цифрах, форма которых отсутствует в обучающей выборке).

Добавим в файл определений digits.def различные образы цифры ноль. Обучим программу и проверим на практике. Для этого откроем файл DIGITS.DEF в блокноте и добавим образы цифры ноль:



Обучим программу

Проверим работу на практике. Задаём символ, которого нет в обучающей выборке

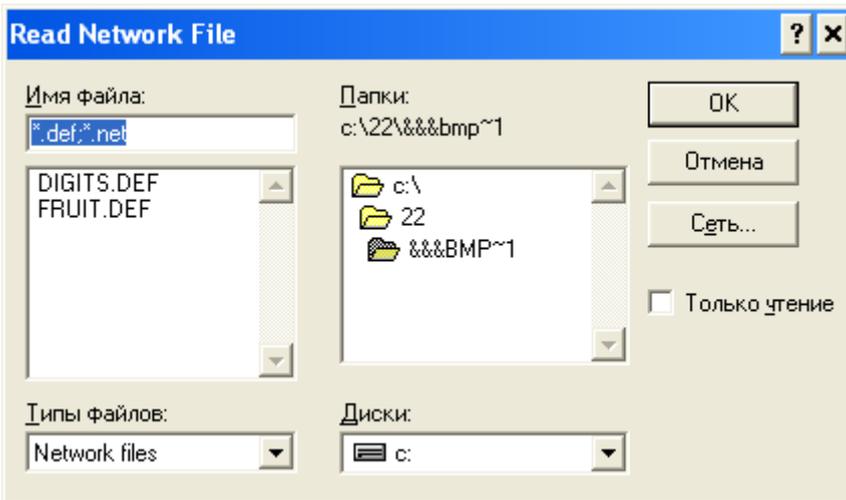


Даже при наличии искажений символ распознан верно. А для этого пришлось всего лишь добавить 5 новых очертаний в обучающую выборку.

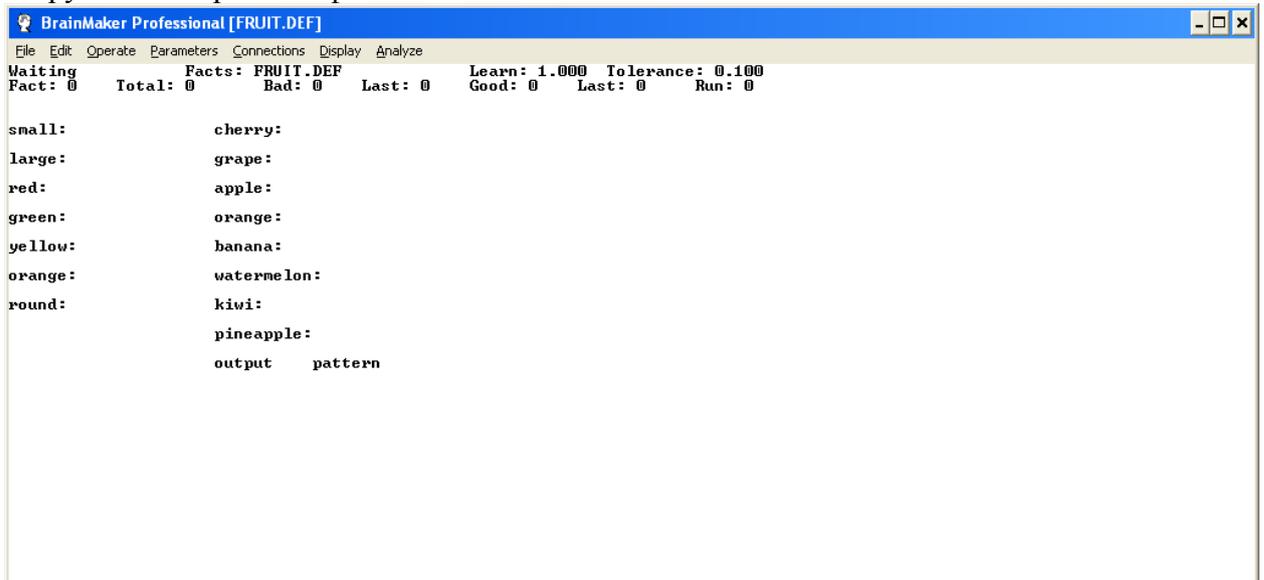
Задание: научите пакет распознавать неизвестную ему цифру (т.е. цифру, начертание которой отсутствует в обучающей выборке). Проверьте, как пакет будет реагировать на искажение её формы, появление разрывов, наростов, и др. помех.

Пример 2.

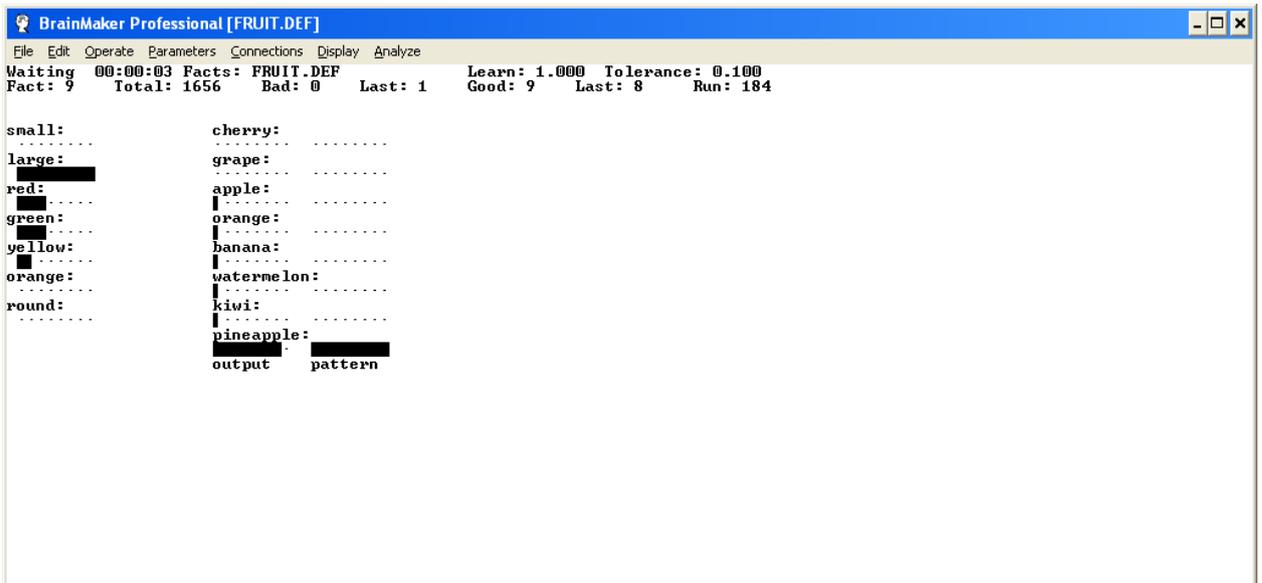
В строке меню выбираем File -> Read Network. В появившемся окне выбираем файл fruit.def .



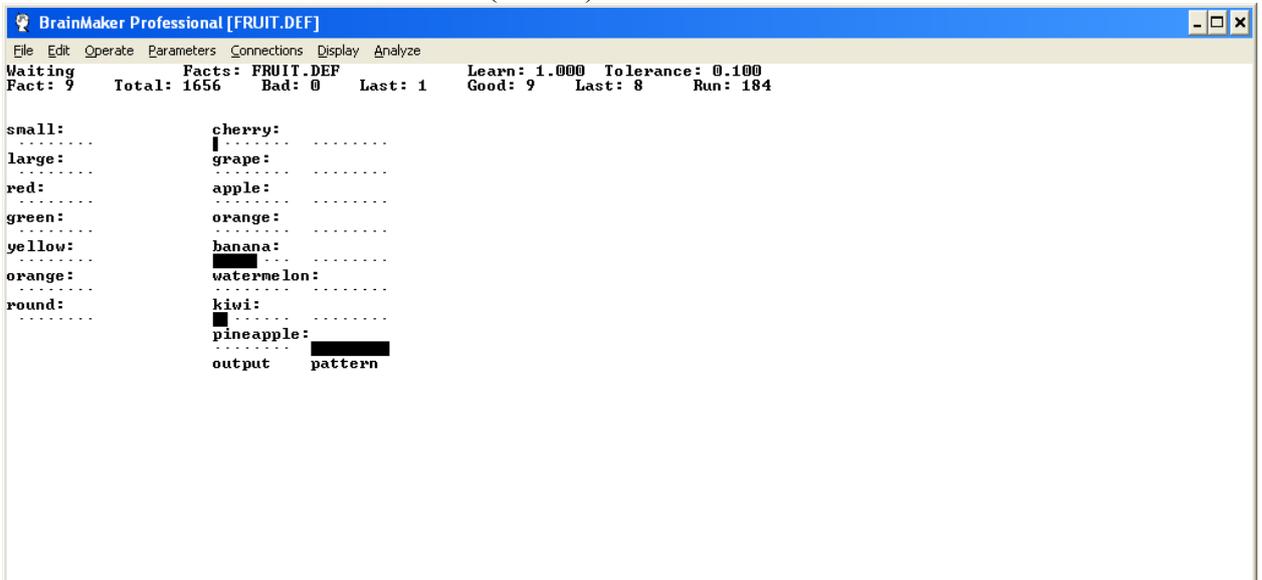
Загружаем выбранный файл:



На экране слева появляется столбец признаков:
 small (маленький),
 large (длинный),
 red (красный),
 green (зеленый),
 yellow (желтый),
 orange (оранжевый),
 round (круглый),
 а справа - столбец с названиями фруктов.
 Обучаем сеть (Ctrl+T).



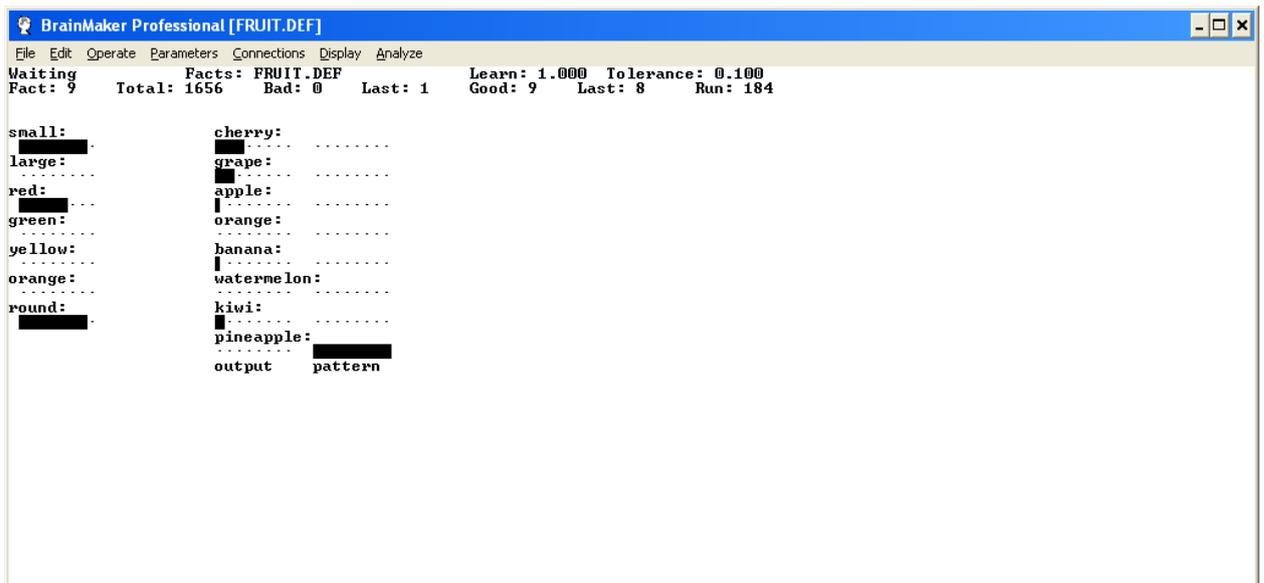
Очищаем исходные данные (Ctrl+E).



Включаем режим распознавания (Ctrl+J).

После этого задаем свойства фрукта, например: очень маленький, очень красный, круглый. Нейросеть распознает:

```
cherry      !-----!
grape       !----!
apple       !--!
banana      !--!
kiwi        !---!
```



Из этих данных видно, что больше всего этот фрукт похож на вишню. После аналогичной корректировки этого файла можно значительно расширить возможности распознавания фруктов.

Классификация графических объектов по их ориентации на плоскости.

Основная задача этой части работы - научить нейросеть различать объекты, имеющие вертикальную ориентацию от горизонтально ориентированных и равномерно распределенных на плоскости объектов.

1. Для этого файл `digit.def` сохраните под другим именем: `digit1.def`, сотрите в нём всё содержимое после слова `facts`. Создайте в матрицах рецепторов (полях 8x8 после слова `facts`) новую обучающую выборку:

Цифрой 1 обозначьте - горизонтальные линии разных размеров (длины и ширины) в разных частях экрана (их нужно создать несколько штук, и каждую пометить цифрой 1);

Цифрой 2 обозначьте - вертикальные линии разных размеров в разных частях экрана;

Цифрой 3 обозначьте - фигуры, не имеющие четкой ориентации по вертикали и горизонтали (квадрат, круг, симметричные по вертикали и горизонтали многоугольники,...) разных размеров в разных частях экрана.

Такая обучающая выборка позволит нейросети определять ориентацию на плоскости графических объектов: код "1" будут иметь горизонтально ориентированные объекты, код "2" – вертикально ориентированные объекты, код "3" – равномерно распределённые по плоскости.

2. Обучите нейросеть распознавать эти фигуры.

3. Предъявите для распознавания их ориентации новые фигуры, не задававшиеся в обучающей выборке и оцените качество распознавания.

4. Проверьте, как действуют на результаты распознавания "помехи" (разрывы линий, утолщения, наросты, сдвиги,...).

5. Проверьте, как будет реагировать программа на наклонные фигуры и при каком наклоне возникнет отказ в распознавании (появляется ошибка).

6. Поинтересуйтесь, к какому классу объектов нейросеть отнесет буквы А, М, срезанную по диагонали часть квадрата, знаки $>$, $<$, $+$, $=$.

7. Оформите отчёт и отразите в нём результаты, полученные в заданиях 1-6.

Пример:

Сохраним файл `Digits.def` под другим именем: `rep_5.def`. Затем откроем созданный файл и удалим все содержимое после слова `facts`.

Приступим к созданию обучающей выборки. В матрицах рецепторов 8x8 создадим по три примера вертикальных, горизонтальных линий и равномерно распределенных фигур расположенных в разных частях экрана, имеющих разную длину и ширину. (Таблица №1).

горизонтальные линии			равномерно распределенные фигуры		
..... ..XXXXX.
1	1	1	3	3	3
вертикальные линии					
.....X.X.X.X.X.X..... ..X..... ..X..... ..X..... ..X.....XX.. ...XX.. ...XX.. ...XX.. ...XX.. ...XX..
2	2	2	3	3	3

Таблица Обучающая выборка

Цифрой 1 пометим горизонтальные линии, цифрой 2 – вертикальные, цифрой 3 – равномерно распределенные фигуры. Данная обучающая выборка позволит нейросети определять ориентацию на плоскости графических объектов. Результаты будут представлены на индикаторе распознавания (*output*) метками с различной степенью насыщенности.

код “1” будут означать, что объект горизонтально ориентирован.

код “2” будут означать, что объект вертикально ориентирован.

код “3” будут означать, что объект равномерно распределён по плоскости.

Сохраним созданную выборку и загрузим её в программу VbrainMaker.exe. Затем обучим нейросеть созданной выборке. Программа готова к работе.

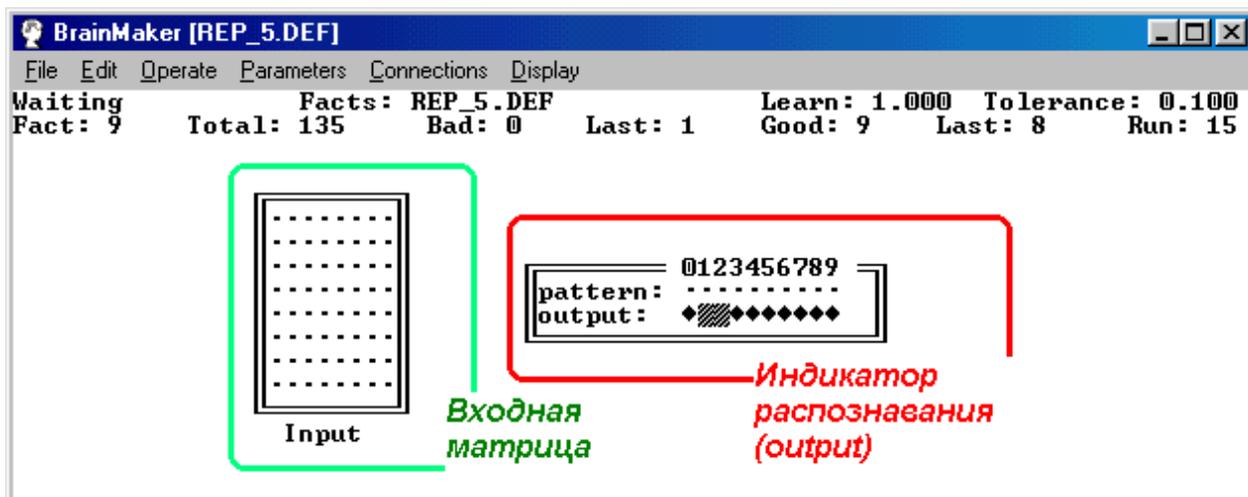


Рисунок 1 Окно нейросети, обученной выборке rep_5.def

Проверим работу программы. Для этого покажем ей несколько образов похожих на образы в обучающей выборке, но не задававшиеся в ней.

Сначала проверим распознавание горизонтальных линий.

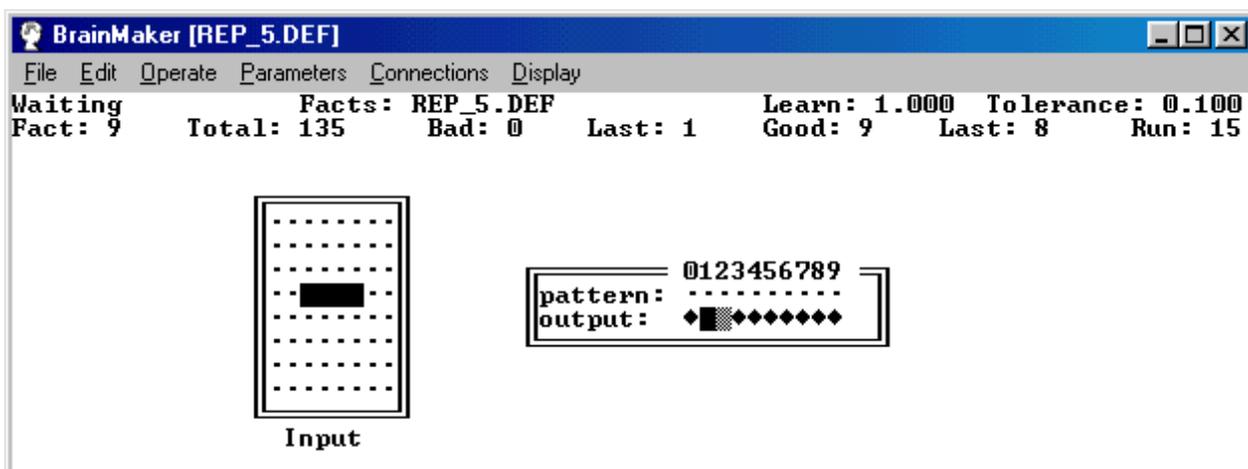


Рисунок 2

Линия распознается как горизонтальная и качество распознавания хорошее.

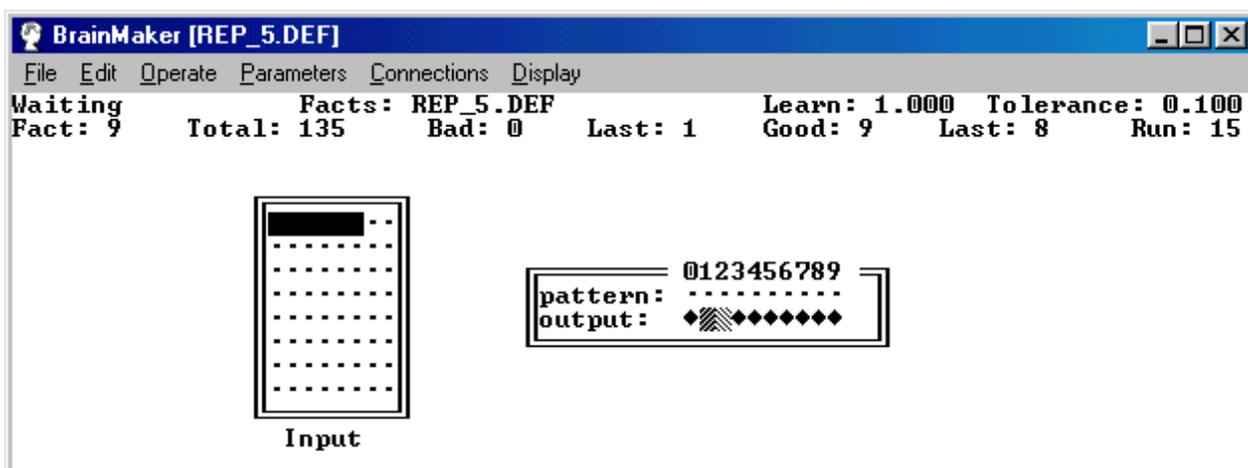


Рисунок 3

Линия распознается как горизонтальная и качество распознавания удовлетворительное.

Теперь проверим распознавание вертикальных линий.

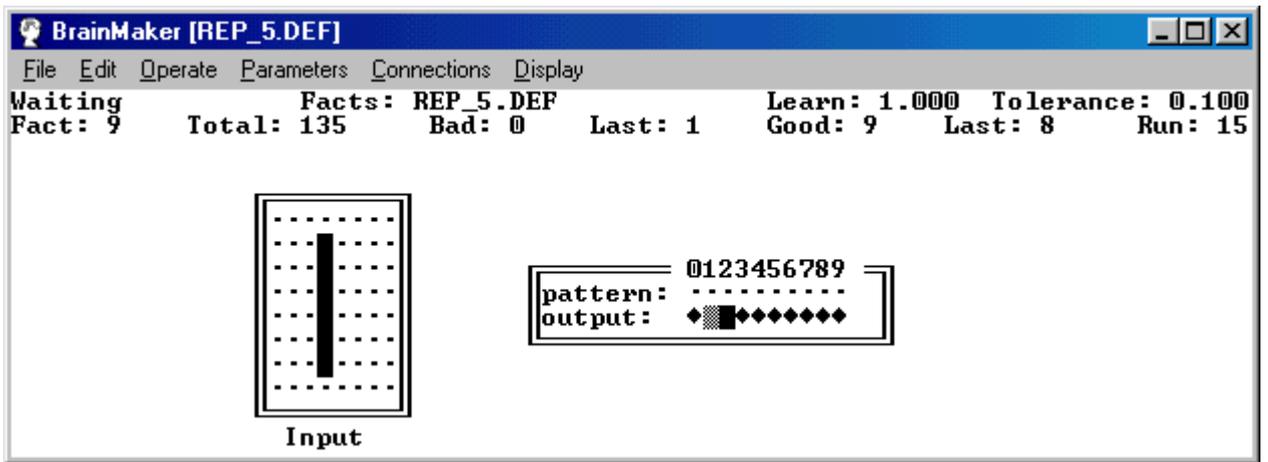


Рисунок 4

Линия распознается как вертикальная, качество распознавания хорошее.

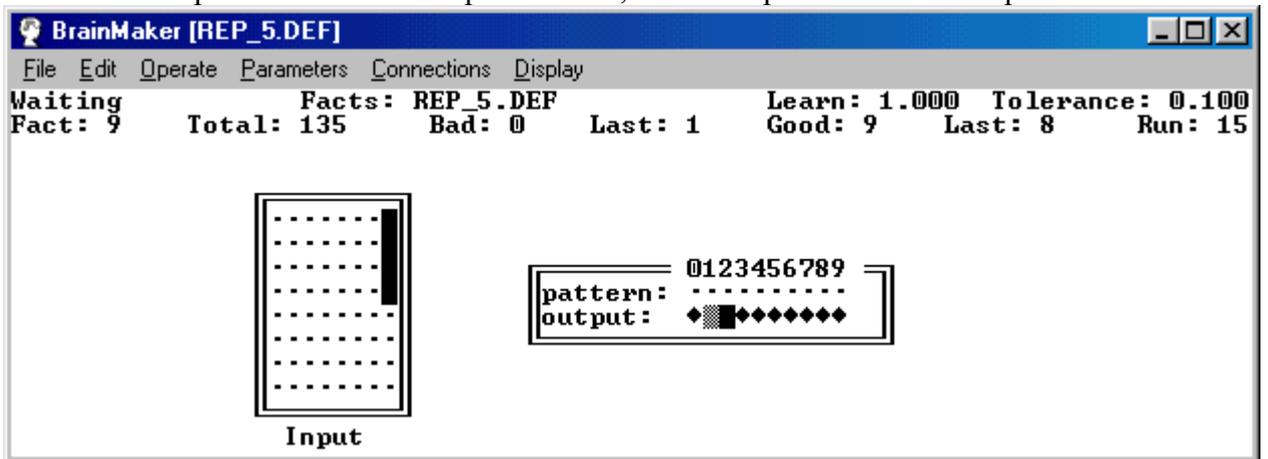
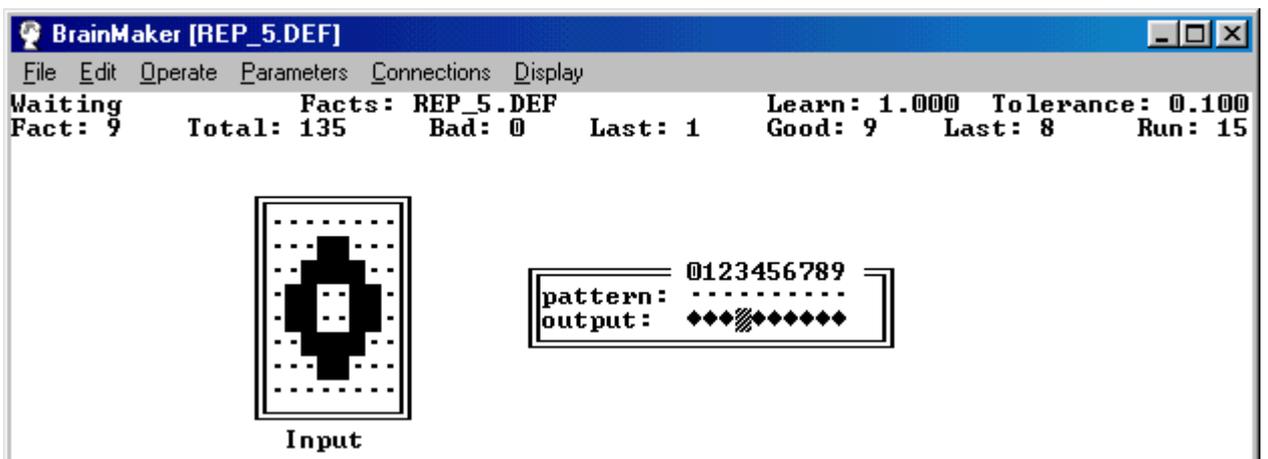


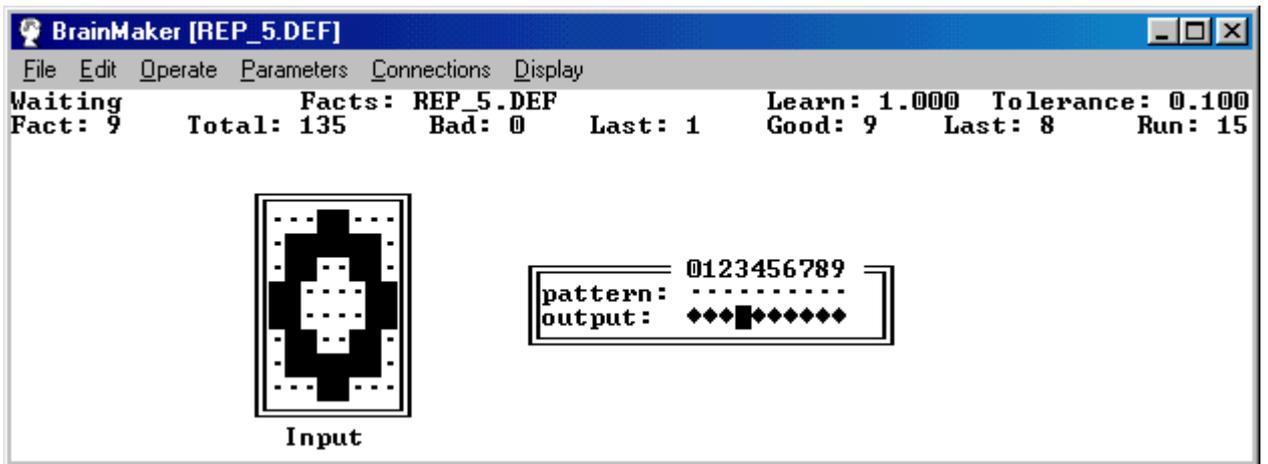
Рисунок 5

Линия распознается как вертикальная, качество распознавания хорошее.

Теперь посмотрим, как программа распознает равномерно распределенные объекты (следующие 2 рисунка).



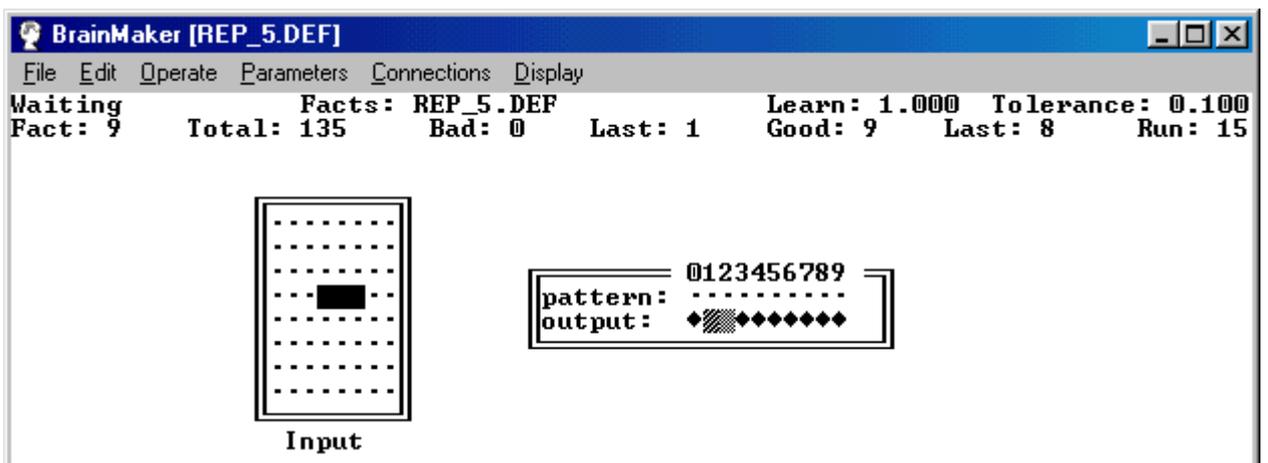
Объект однозначно распознается как равномерно распределенный. Качество распознавания хорошее.



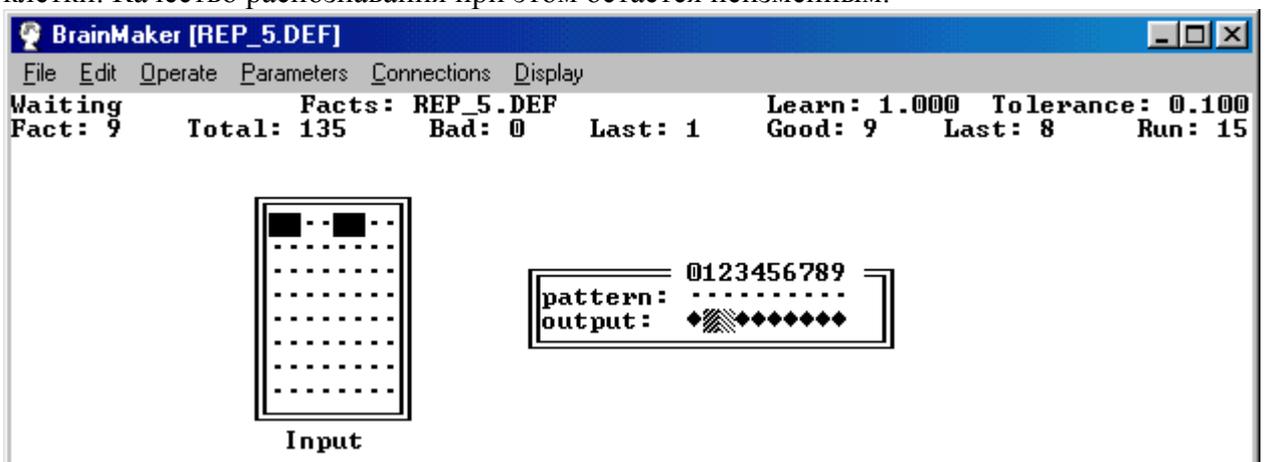
Объект однозначно распознается как равномерно распределенный. Качество распознавания очень хорошее.

Теперь проверим, как на распознавании фигур отразятся “помехи” такие как разрывы линий, утолщения, наросты, сдвиги...

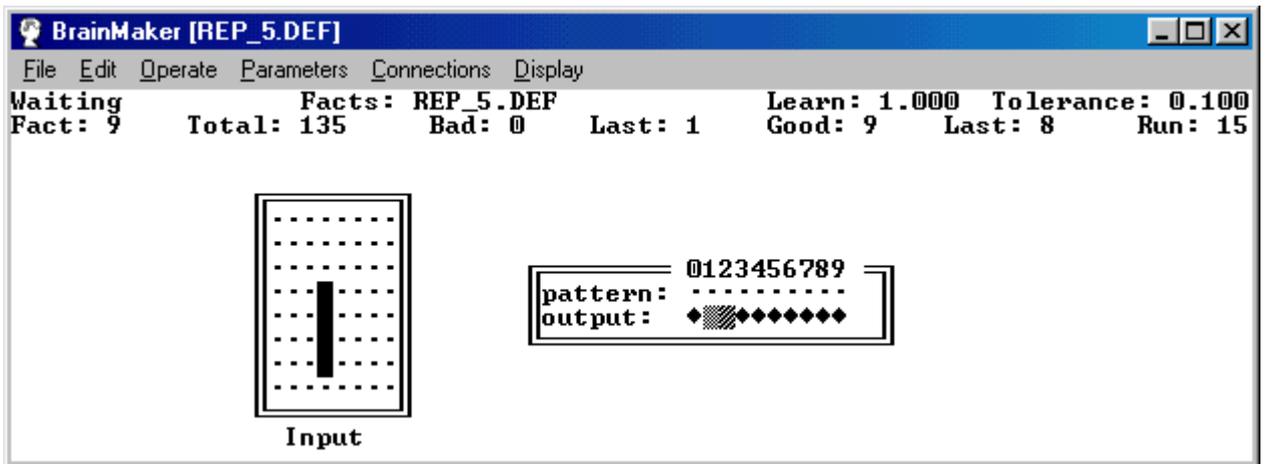
Возьмем линию, как на Рисунке № 2 и уменьшим её длину справа на 1 клетку. Качество распознавания при этом ухудшается.



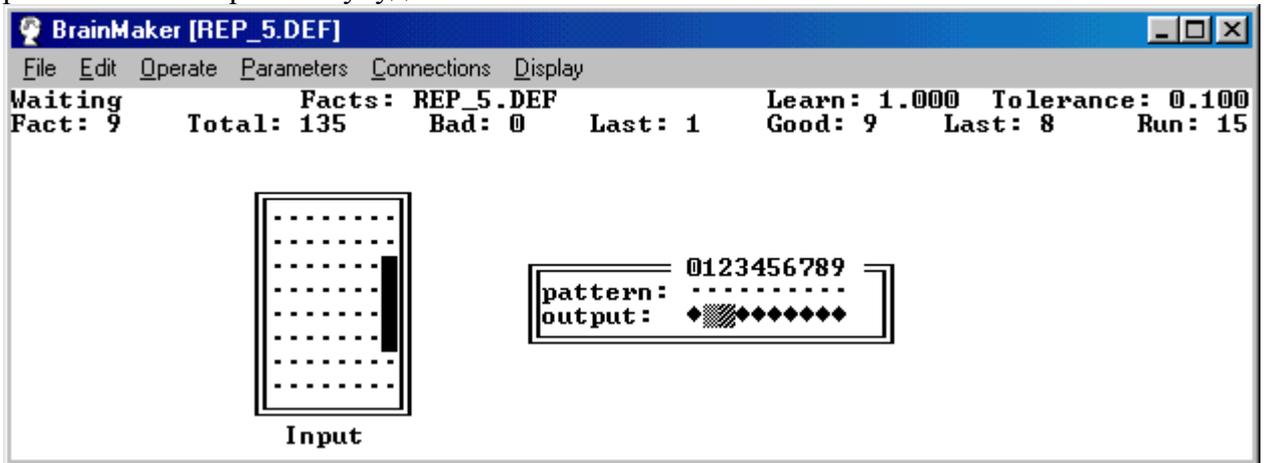
Возьмем линию, как на Рисунке № 3 и сделаем у неё разрыв посередине на 2 клетки. Качество распознавания при этом остается неизменным.



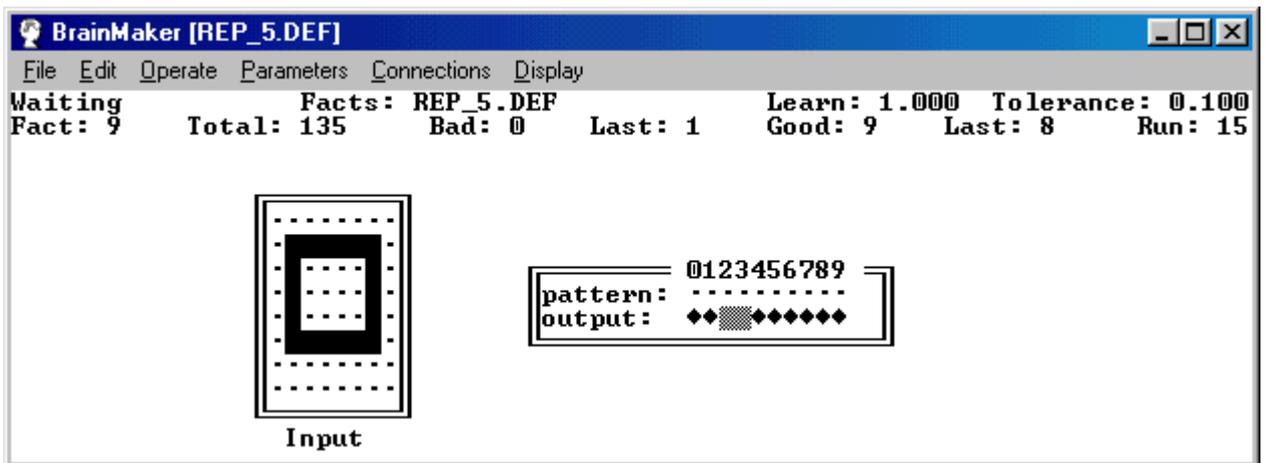
Возьмем линию, как на Рисунке № 4 и уменьшим её длину сверху на 2 клетки. Качество распознавания при этом ухудшается.



Возьмем линию, как на Рисунке № 5 и сдвинем её вниз на 2 клетки. Качество распознавания при этом ухудшается.

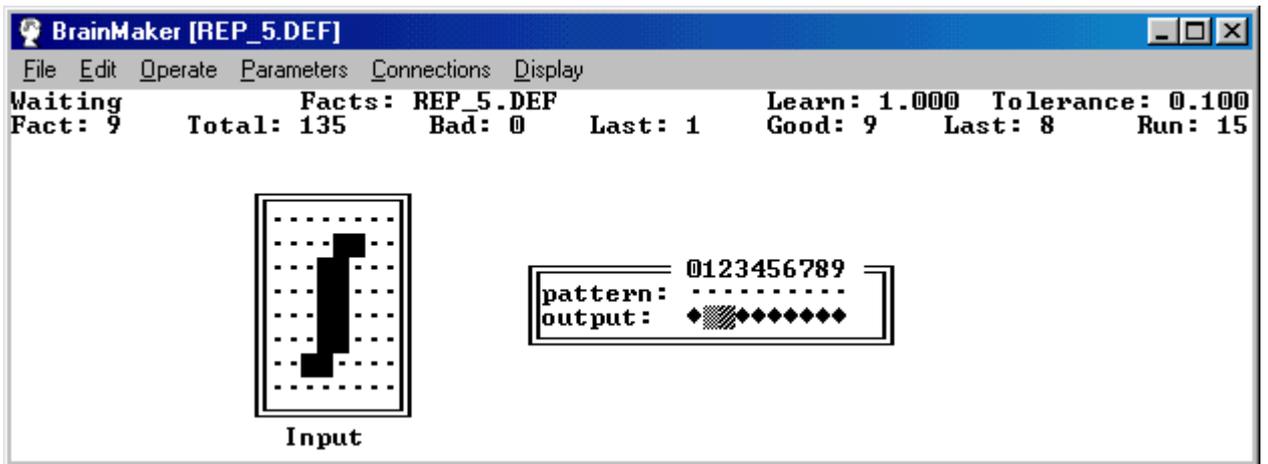


Возьмем квадрат из обучающей выборки (Таблица № 1 слева) и уменьшим его длину снизу на 1 клетку. Качество распознавания при этом ухудшается. Нейросеть опознает её одинаково как вертикальную линию и равномерно распределенную фигуру.

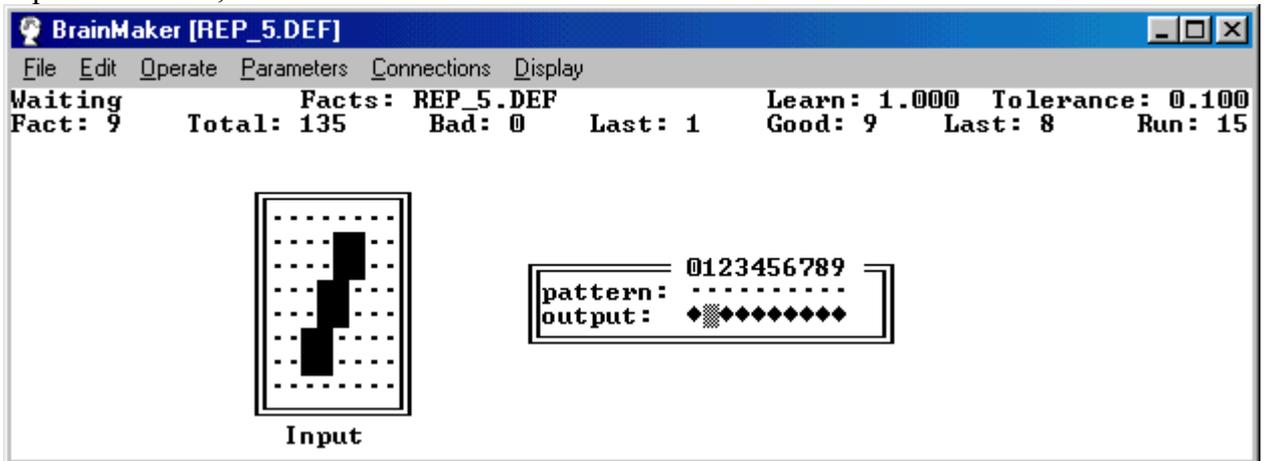


Теперь проверим, как программа будет реагировать на наклонные фигуры, и при каком наклоне возникнет отказ в распознавании.

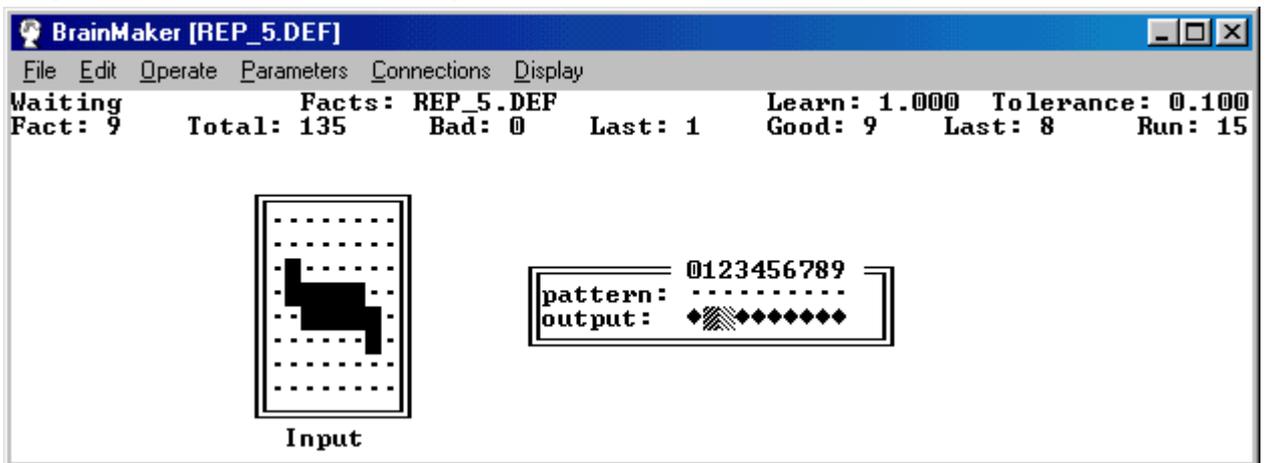
Возьмем вертикальную линию из обучающей выборки (Таблица № 1 справа) и сдвинем один её конец вправо, а другой влево на 1 клетку. Качество распознавания ухудшается: нейросеть находит сходство с горизонтальной линией.



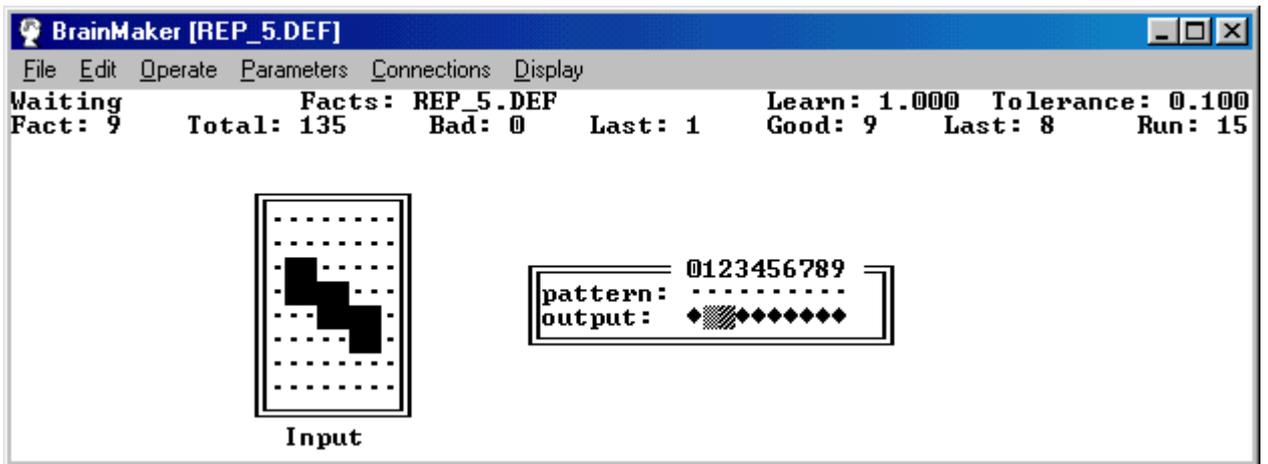
Теперь сделаем наклон сильнее. Произошла ошибка. Линия распознается как горизонтальная, хотя и не совсем точно.



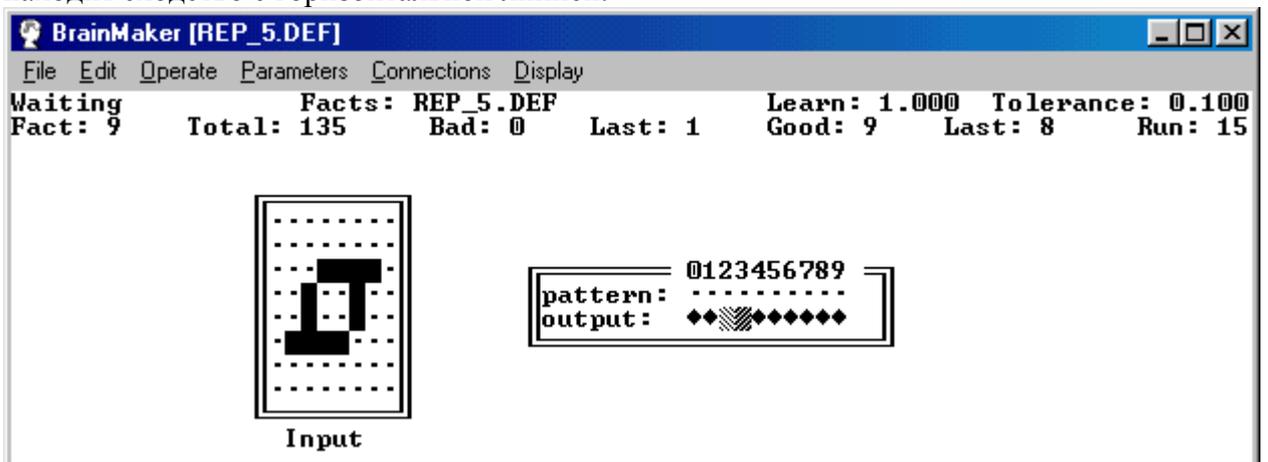
Возьмем горизонтальную линию из обучающей выборки (Таблица № 1) и сдвинем один её конец вправо, а другой влево на 1 клетку. Качество распознавания ухудшается: нейросеть находит сходство с вертикальной линией.



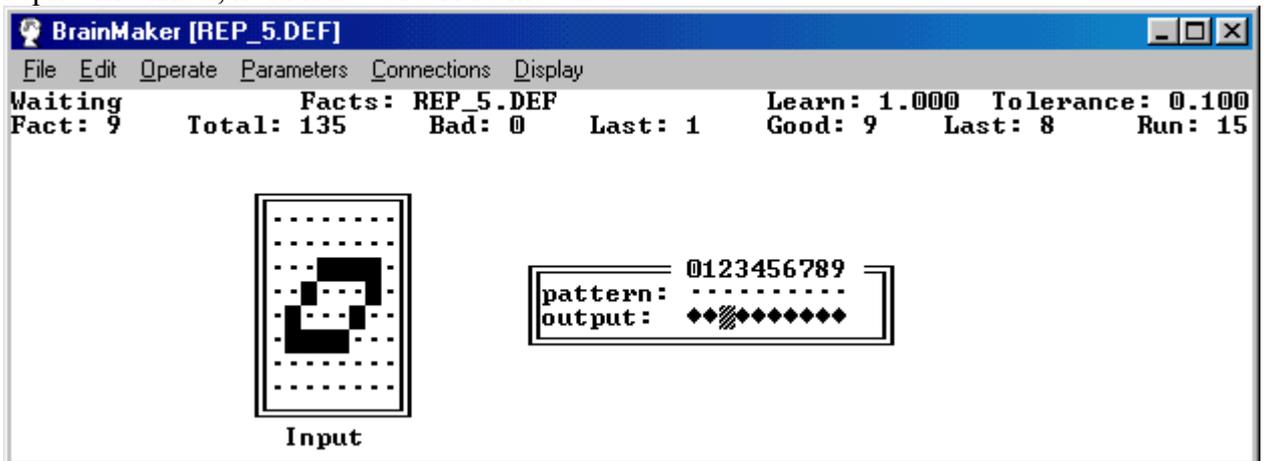
Теперь сделаем наклон сильнее. Произошла ошибка. Линия распознается как вертикальная и как горизонтальная, но, судя по индексу распознавания, больше как горизонтальная.



Возьмем квадрат из обучающей выборки (Таблица № 1) и сдвинем один её конец вправо, а другой влево на 1 клетку. Качество распознавания ухудшается: нейросеть находит сходство с горизонтальной линией.

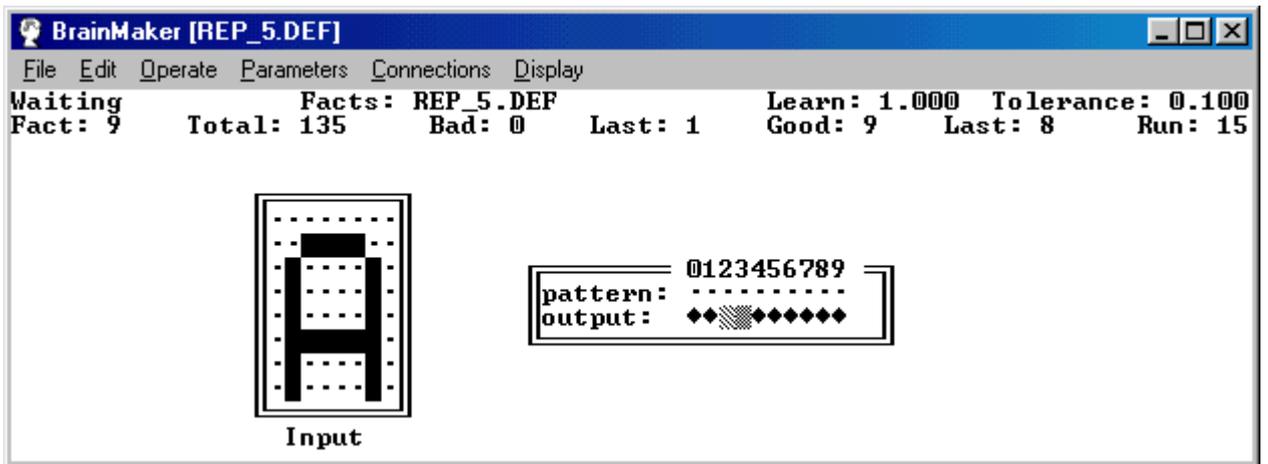


Теперь сделаем наклон сильнее. Произошла ошибка. Линия распознается как горизонтальная, хотя и не совсем точно.

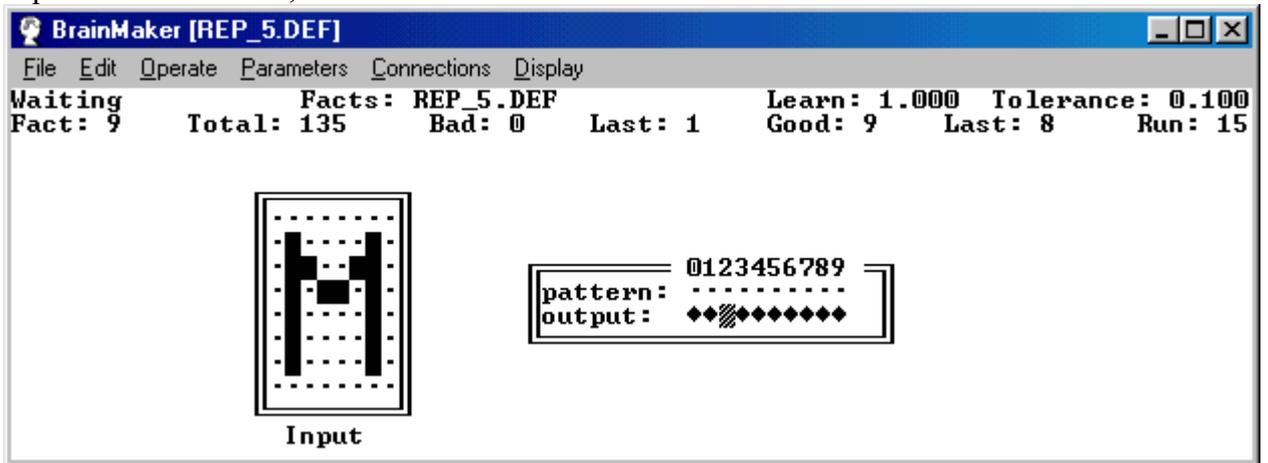


Интересно, к какому классу объектов нейросеть отнесет буквы: А, М, срезанную по диагонали часть квадрата, знаки >, <, +, =.

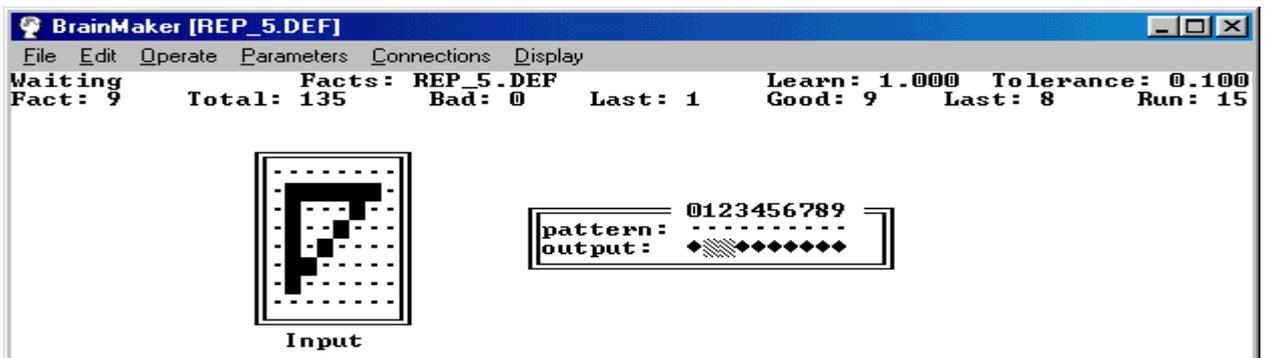
Нарисуем во входной матрице символ «А». Нейросеть относит его в основном к классу равномерно распределенных фигур, но также находит сходство с классом вертикальных фигур.



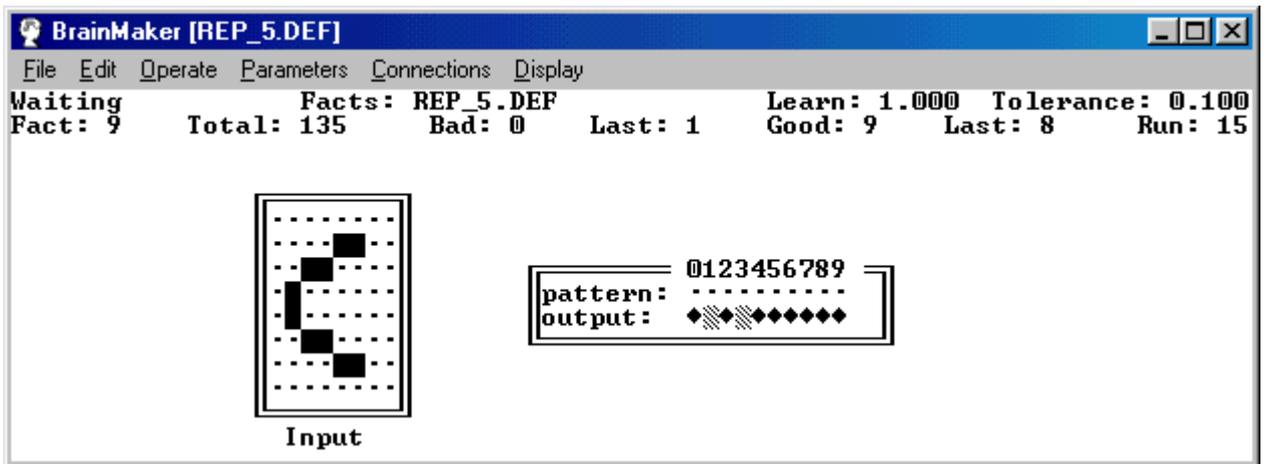
Нарисуем во входной матрице символ «М». Нейросеть относит его к классу вертикальных линий, но не очень точно.



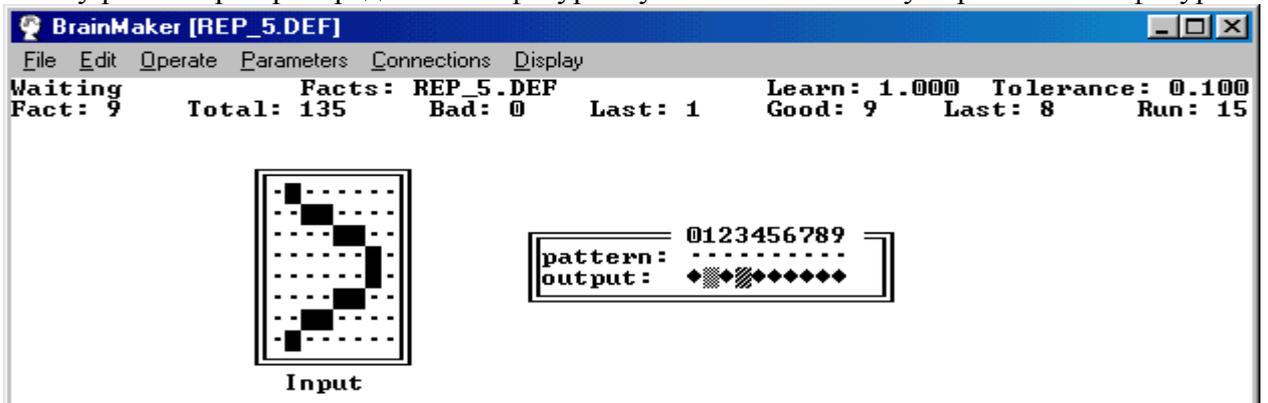
Нарисуем во входной матрице квадрат, а затем как бы отрезем по диагонали нижнюю часть. Нейросеть относит его одинаково как к классу вертикальных фигур, так и к классу горизонтальных фигур.



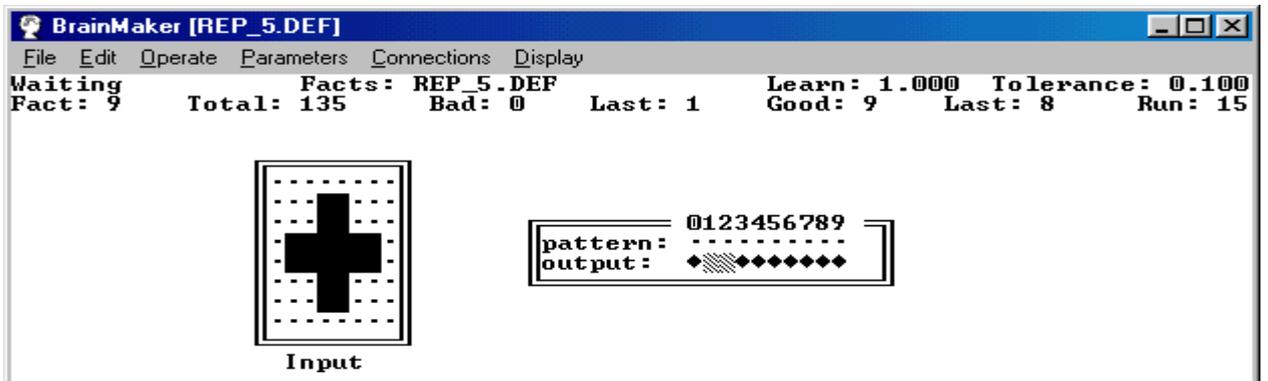
Нарисуем во входной матрице символ «<<». Нейросеть относит его одинаково как к классу равномерно распределенных фигур, так и к классу горизонтальных фигур



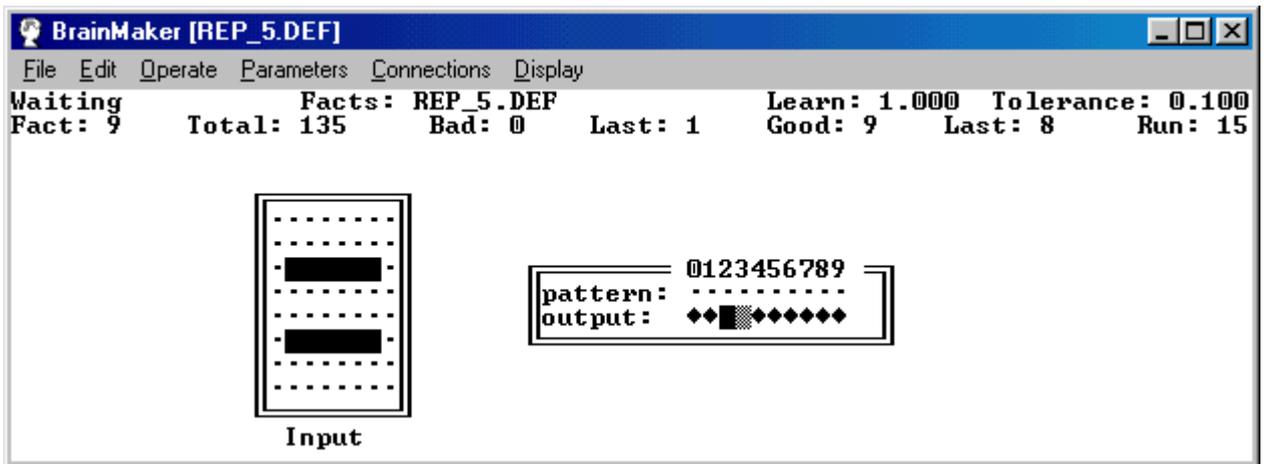
Нарисуем во входной матрице символ «>». Нейросеть относит его в основном к классу равномерно распределенных фигур и чуть меньше к классу вертикальных фигур.



Нарисуем во входной матрице символ «+». Нейросеть относит его одинаково как к классу вертикальных, так и к классу горизонтальных фигур



Нарисуем во входной матрице символ «=». Нейросеть относит его в основном к классу вертикальных фигур и чуть меньше к классу равномерно распределенных фигур.



Выходит, что нейроЭВМ способна распознавать объекты по их соответствию с изначально предъявленными примерами. Нейросеть способна узнавать соответствие с эталоном у фигур с различными видами деформации, вплоть до разрыва. Небольшие отклонения от эталона, однако, приводят к ухудшению распознавания, а иногда и к ошибкам. Поэтому огромное значение имеет правильно подобранная обучающая выборка.

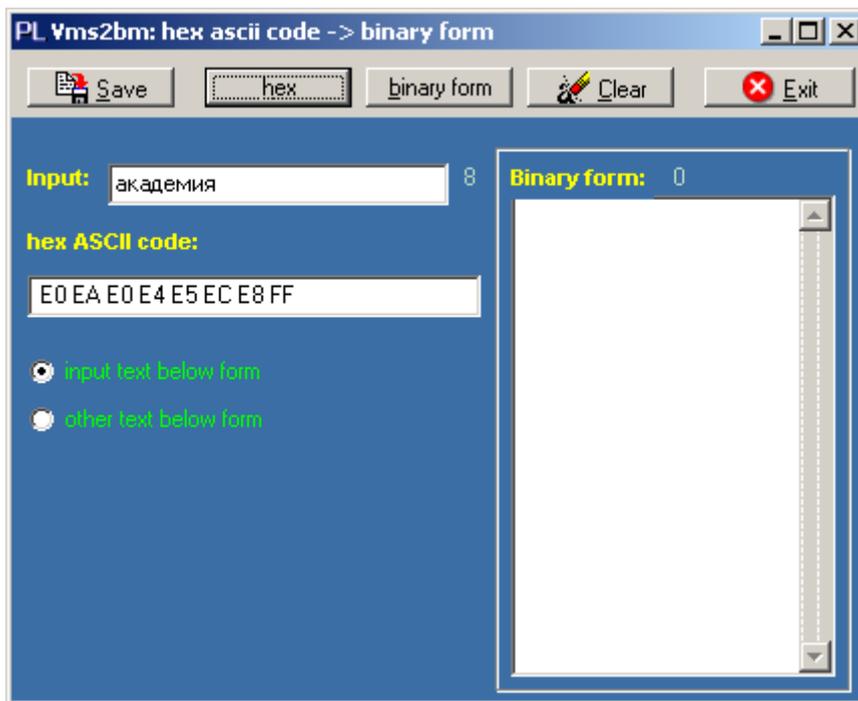
Распознавание текстовой информации.

В матрицу 8x8 обучающей выборки (матрицу рецепторов) можно заносить не только графические объекты, но и различные коды. Например, используя таблицу ASCII, можно научить перцептрон отличать кошку от собаки и от зайца. Кодирование символов можно производить либо по любой таблице, например - таблице ASCII, либо – с помощью

Таблица 1 Альтернативная кодировочная таблица кода ASCII

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
0																	0
1																	1
2	пробел	!	“	#	\$	%	&	'	()	*	+	,	-	.	/	2
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?	3
4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	4
5	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_	5
6	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	6
7	p	q	r	s	t	u	v	w	x	y	z	{		}			7
8	А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П	8
9	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я	9
A	а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п	A
B	р	с	т	у	ф	х	ц	ч	ш	щ	ъ	ы	ь	э	ю	я	B
C																	C
D																	D
E	р	с	т	у	ф	х	ц	ч	ш	щ	ъ	ы	ь	э	ю	я	E
F	Е	е											№				F
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	

специальной программы, как например - «Программа для кодирования слов в коде ASCII - Vms2bm.exe», внешний вид которой приведен на рисунке:



Запишите слово кошка в кодах ASCII - каждая буква в этом коде занимает один байт. Условимся каждой букве отводить одну строку (8 рецепторов), единицы кода заносить в матрицу, зажигая соответствующий рецептор. Тогда, буква “а” (ее код ASCII - шестнадцатиричное число “A0”) займет целую строку и будет находиться в пятой строке матрицы рецепторов. Поскольку шестнадцатиричное число A0 в двоичном виде выглядит, как 10100000, пятая строка матрицы примет вид:

10100000.

В первой и четвёртой строках нужно поместить код ASCII буквы “к”, во второй – буквы “о”, в третьей – буквы “ш”

Набрав слово “кошка”, научим нейросеть воспринимать его, как символ номер 1. Затем сообщим нейросети, что это слово, набранное в других строках, обозначает то же самое. Затем скажем нейросети, что “котик” и “кот” - это тоже кошки (т.е. тоже соответствуют символу с номером 1).

Аналогично научим нейросеть понятиям “собака” (собачка, песик,...) - понятие номер 2, и “заяц” (зайчик, зайныка,...) - понятие номер 3. В результате этой работы будет сформирована обучающая выборка.

После обучения поинтересуйтесь у нейросети, как она считает, кто из названных зверей больше подходит к понятию “кусачка”? На кого больше похож “медведь”?

Результат оформите в виде отчёта, проиллюстрированного соответствующими экранами.

Пример :

Сохраним файл Digits.def под другим именем: гер_6.def. Затем откроем созданный файл и удалим все содержимое после слова *facts*.

Приступим к созданию обучающей выборки. В матрицах рецепторов 8x8 создадим примеры.

Так как для выполнения работ нам не нужна вся таблица ASCII, а нужны только строки с прописными русскими символами, сделаем для себя вспомогательную табличку № 3.

Таблица 2

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
A	а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п	A
B	р	с	т	у	ф	х	ц	ч	ш	щ	ъ	ы	ь	э	ю	я	B

Научим нейросеть различать «кошку», «собаку» и «зайца». Для этого подготовим соответствующую обучающую выборку.

Для начала закодируем слово «кошка» в ASCII, а затем переведем полученные коды в двоичные и запишем в виде матрицы 8x8. В эту же таблицу занесем формы слов «киска» и «кот». В обучающей выборке слова, принадлежащие к классу кошек, будут иметь код 1.

кошка		киска		кот	
10101010	AA	10101010	AA	10101010	AA
10101110	AE	10101000	A8	10101110	AE
10111000	B8	10110001	B1	10110010	B2
10101010	AA	10101010	AA	
10100000	A0	10100000	A0	
.....		
.....		
.....		
1		1		1	

Проделаем то же для класса собак и зайцев.

Закодируем формы слов «собака», «собачка», «песик».

В обучающей выборке слова, принадлежащие к классу собак, будут иметь код 2.

собака		собачка		песик	
10110001	B1	10110001	B1	10101111	AF
10101110	AE	10101110	AE	10100101	A5
10100001	A1	10100001	A1	10110001	B1
10100000	A0	10100000	A0	10101000	A8
10101010	AA	10110111	B7	10101010	AA
10100000	A0	10101010	AA	
.....		10100000	A0	
.....		
2		2		2	

Закодируем формы слов «заяц», «зайчик», «зайка».

В обучающей выборке слова, принадлежащие к классу зайцев, будут иметь код 3.

заяц		зайчик		зайка	
10100111	A7	10100111	A7	10100111	A7
10100000	A0	10100000	A0	10100000	A0
10111111	BF	10101001	A9	10101000	A8
10110110	B6	10110111	B7	10101101	AD
.....		10101000	A8	10111100	BC
.....		10101010	AA	10101010	AA
.....			10100000	A0
.....		
3		3		3	

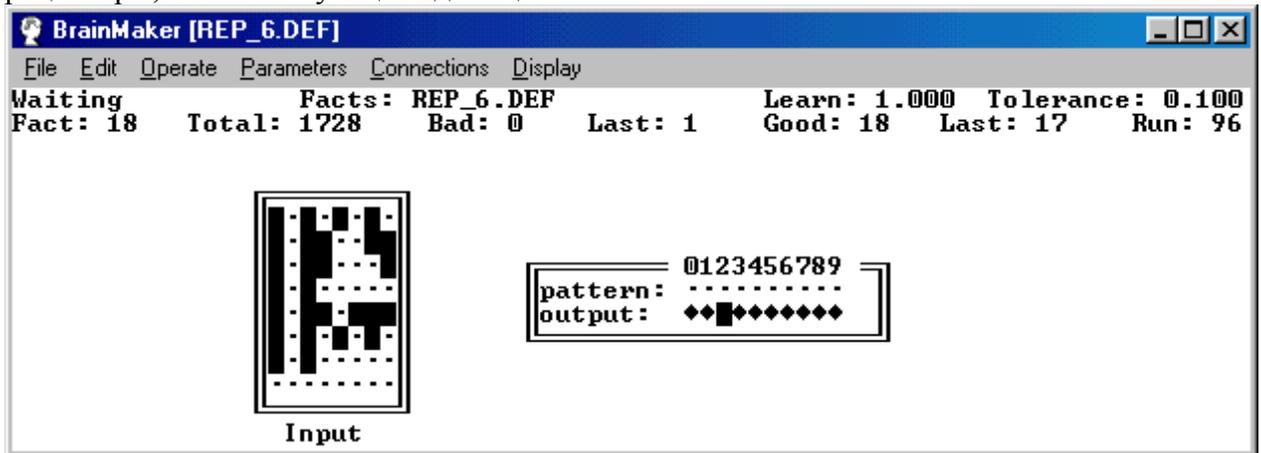
Занесем в обучающую выборку полученные результаты, причем покажем нейросети, что, например, форма слова «кошка» может располагаться как посередине, так и внизу матрицы, то есть как бы мы его не перемещали вверх и вниз оно не меняет своего значения.

Сохраним полученную выборку и загрузим её в BrainMaker. Обучим программу данной выборке.

Теперь проверим, к какому классу программа отнесет слова «кусочка» и «медведь»? Для этого закодируем формы соответствующих слов.

кусачка		медведь	
10101010	AA	10101100	AC
10110011	B3	10100101	A5
10110001	B1	10100100	A4
10100000	A0	10100010	A2
10110111	B7	10100101	A5
10101010	AA	10100100	A4
10100000	A0	10111100	BC
.....		

Занесем полученную форму «кусачка» в рецепторную матрицу нейросети, зажигая рецепторы, соответствующие единице.



Нейросеть определяет принадлежность данной формы к классу собак. Теперь занесем форму «кусачка» в рецепторную матрицу еще раз, только со сдвигом на 1 вниз.

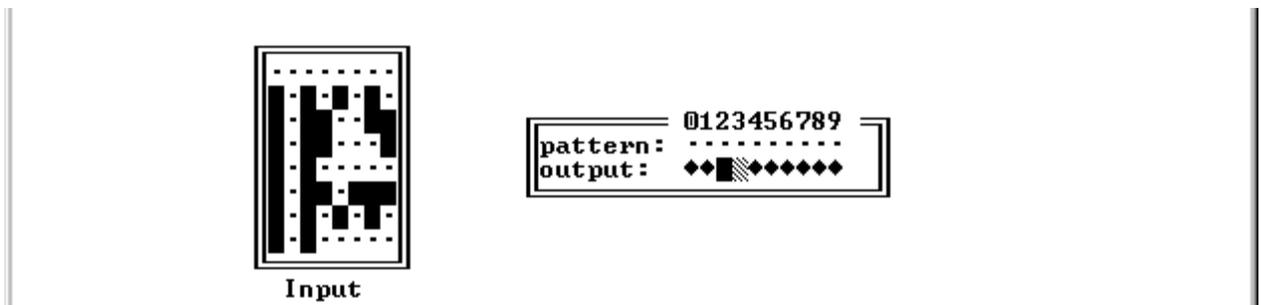
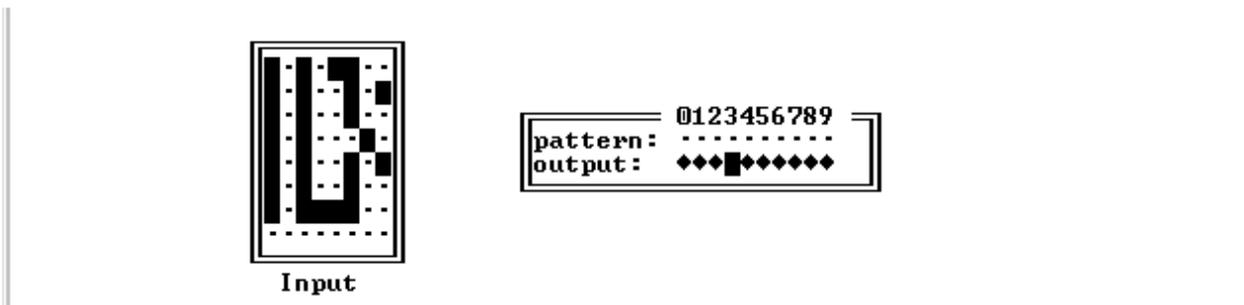


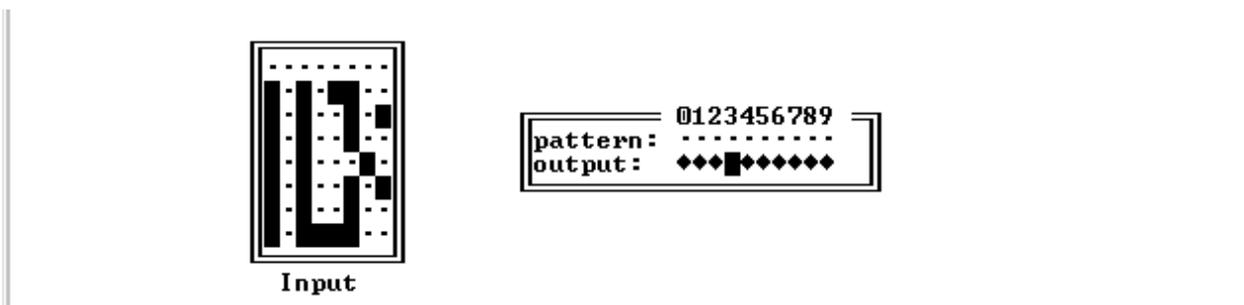
Рисунок 6

Теперь нейросеть определяет принадлежность введенной формы классам собак и зайцев, но больше к классу собак, чем зайцев.

Занесем полученную форму «медведь» в рецепторную матрицу нейросети, зажигая рецепторы, соответствующие единице.



Нейросеть однозначно определяет принадлежность формы «медведь» к классу зайцев. Сдвиг формы на 1 вниз приводит к такому же результату.



НейроЭВМ способна распознавать объекты по их соответствию с изначально предъявленными примерами. Также нейроЭВМ способна находить сходства между эталонами и новыми формами, присваивая новым элементам принадлежность эталонных классов.

Проверка возможности распознавания эмоций.

Произношение слова отличается от его написания. Например, слово Москва произносится, как “мас кваа”. Одно и то же слово можно произнести с разной интонацией. Именно в произношении слова, в интонации проявляется эмоциональная составляющая информации.

Ваша задача: выберите несколько слов и запишите их произношение (транскрипцию) русскими буквами для трех различных эмоций (например, ласково, равнодушно и с удивлением). Обозначьте каждую эмоцию цифрой 1, 2 или 3. Обратите внимание на то, что эмоциональная окраска проявляется в удлиннении звуков, появлении разрывов (пауз) между ними, изменении ударения (которое можно указывать в транскрипции любым редко используемым символом ASCII), что произношение некоторых букв меняется (например, мягкое “я” жестко звучит, как “йаа”), и т.д.

Обучите нейросеть распознаванию эмоций.

Например, ласковое произношение обозначим 1, равнодушное - 2, удивленное - 3. Тогда для слова “кошка”: коошкаа-1, кошка-2, “коо? шка” - 3; для слова “собака”: “сабаака” - 1, “сабака” - 2, “сабаа? ка” - 3; для слова “заяц”: “заец” - 1, “заец” - 2, “заа?йец” - 3; и др. Можно выбрать и другие эмоции: иронию, сарказм, радость,... Количество слов для обучающей выборки не ограничивается (программа сама подскажет, когда обучение можно заканчивать - она перестанет делать ошибки в распознавании). Ограничивается возможностями программы только количество эмоций. Для проверки возможностей системы подберите другие слова, в произношении которых наиболее отчетливо проявляется эмоциональная составляющая.

Фиксируя результаты эксперимента в таблице, докажите, возможно или нет обучить перцептрон распознаванию эмоций.

Оформите иллюстрированный отчет.

Пример отчёта:

Выберем несколько слов русского языка (например, собака, кошка, заяц) запишем их произношение (транскрипцию) русскими буквами для трех различных эмоций:

- Цифрой 1 обозначим ласковое произношение (данная эмоциональная окраска проявляется в удлинении звуков);
- Цифрой 2 – равнодушное (краткость звуков);
- Цифрой 3 – удивленное (усиление ударения, интонация вверх).

с	E1	<i>1</i>	<i>1</i>	<i>1</i>	0	0	0	0	<i>1</i>	XXX...X
а	A0	<i>1</i>	0	<i>1</i>	0	0	0	0	0	X.X.....
б	A1	<i>1</i>	0	<i>1</i>	0	0	0	0	<i>1</i>	X.X...X
а	A0	<i>1</i>	0	<i>1</i>	0	0	0	0	0	X.X.....
а	A0	<i>1</i>	0	<i>1</i>	0	0	0	0	0	X.X.....
а	A0	<i>1</i>	0	<i>1</i>	0	0	0	0	0	X.X.....
к	AA	<i>1</i>	0	<i>1</i>	0	<i>1</i>	0	<i>1</i>	0	X.X.X.X.
а	A0	<i>1</i>	0	<i>1</i>	0	0	0	0	0	X.X.....

к	AA	<i>1</i>	0	<i>1</i>	0	<i>1</i>	0	<i>1</i>	0	X.X.X.X.
о	AE	<i>1</i>	0	<i>1</i>	0	<i>1</i>	<i>1</i>	<i>1</i>	0	X.X.XXX.
о	AE	<i>1</i>	0	<i>1</i>	0	<i>1</i>	<i>1</i>	<i>1</i>	0	X.X.XXX.
ш	E8	<i>1</i>	<i>1</i>	<i>1</i>	0	<i>1</i>	0	0	0	XXX.X...
к	AA	<i>1</i>	0	<i>1</i>	0	<i>1</i>	0	<i>1</i>	0	X.X.X.X.
а	A0	<i>1</i>	0	<i>1</i>	0	0	0	0	0	X.X.....
а	A0	<i>1</i>	0	<i>1</i>	0	0	0	0	0	X.X.....
	00	0	0	0	0	0	0	0	0

з	A7	<i>1</i>	0	<i>1</i>	0	0	<i>1</i>	<i>1</i>	<i>1</i>	X.X...XXX
а	A0	<i>1</i>	0	<i>1</i>	0	0	0	0	0	X.X.....
а	A0	<i>1</i>	0	<i>1</i>	0	0	0	0	0	X.X.....
а	A0	<i>1</i>	0	<i>1</i>	0	0	0	0	0	X.X.....
и	A8	<i>1</i>	0	<i>1</i>	0	<i>1</i>	0	0	0	X.X.X...
ц	E6	<i>1</i>	<i>1</i>	<i>1</i>	0	0	<i>1</i>	<i>1</i>	0	XXX...XX.
	00	0	0	0	0	0	0	0	0
	00	0	0	0	0	0	0	0	0

с	E1	<i>1</i>	<i>1</i>	<i>1</i>	0	0	0	0	<i>1</i>	XXX...X
а	A0	<i>1</i>	0	<i>1</i>	0	0	0	0	0	X.X.....
б	A1	<i>1</i>	0	<i>1</i>	0	0	0	0	<i>1</i>	X.X...X
а	A0	<i>1</i>	0	<i>1</i>	0	0	0	0	0	X.X.....
к	AA	<i>1</i>	0	<i>1</i>	0	<i>1</i>	0	<i>1</i>	0	X.X.X.X.
а	A0	<i>1</i>	0	<i>1</i>	0	0	0	0	0	X.X.....
	00	0	0	0	0	0	0	0	0
	00	0	0	0	0	0	0	0	0

к	AA	<i>1</i>	0	<i>1</i>	0	<i>1</i>	0	<i>1</i>	0	X.X.X.X.
о	AE	<i>1</i>	0	<i>1</i>	0	<i>1</i>	<i>1</i>	<i>1</i>	0	X.X.XXX.
ш	E8	<i>1</i>	<i>1</i>	<i>1</i>	0	<i>1</i>	0	0	0	XXX.X...
к	AA	<i>1</i>	0	<i>1</i>	0	<i>1</i>	0	<i>1</i>	0	X.X.X.X.
а	A0	<i>1</i>	0	<i>1</i>	0	0	0	0	0	X.X.....
	00	0	0	0	0	0	0	0	0
	00	0	0	0	0	0	0	0	0
	00	0	0	0	0	0	0	0	0

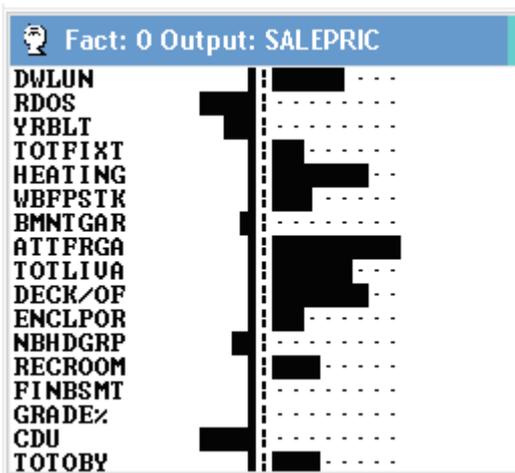
з	A7	1	0	1	0	0	1	1	1	X.X..XXX
а	A0	1	0	1	0	0	0	0	0	X.X.....
и	A8	1	0	1	0	1	0	0	0	X.X.X...
ц	E6	1	1	1	0	0	1	1	0	XXX..XX.
	00	0	0	0	0	0	0	0	0
	00	0	0	0	0	0	0	0	0
	00	0	0	0	0	0	0	0	0
	00	0	0	0	0	0	0	0	0

с	E1	1	1	1	0	0	0	0	1	XXX....X
а	A0	1	0	1	0	0	0	0	0	X.X.....
б	A1	1	0	1	0	0	0	0	1	X.X....X
а	A0	1	0	1	0	0	0	0	0	X.X.....
а	A0	1	0	1	0	0	0	0	0	X.X.....
?	3F	0	0	1	1	1	1	1	1	..XXXXXX
к	AA	1	0	1	0	1	0	1	0	X.X.X.X.
а	A0	1	0	1	0	0	0	0	0	X.X.....

к	AA	1	0	1	0	1	0	1	0	X.X.X.X.
о	AE	1	0	1	0	1	1	1	0	X.X.XXX.
о	AE	1	0	1	0	1	1	1	0	X.X.XXX.
?	3F	0	0	1	1	1	1	1	1	..XXXXXX
ш	E8	1	1	1	0	1	0	0	0	XXX.X...
к	AA	1	0	1	0	1	0	1	0	X.X.X.X.
а	A0	1	0	1	0	0	0	0	0	X.X.....
	00	0	0	0	0	0	0	0	0

з	A7	1	0	1	0	0	1	1	1	X.X..XXX
а	A0	1	0	1	0	0	0	0	0	X.X.....
а	A0	1	0	1	0	0	0	0	0	X.X.....
?	3F	0	0	1	1	1	1	1	1	..XXXXXX
и	A8	1	0	1	0	1	0	0	0	X.X.X...
ц	E6	1	1	1	0	0	1	1	0	XXX..XX.
	00	0	0	0	0	0	0	0	0
	00	0	0	0	0	0	0	0	0

Сохраним файл DIGITS.DEF под именем DIGITS_4.DEF, сотрем в нем все содержимое после слова facts.



Некоторые параметры (такие, как RDOS) оказывают отрицательное влияние на цену. Но в начале исследования это неизвестно.

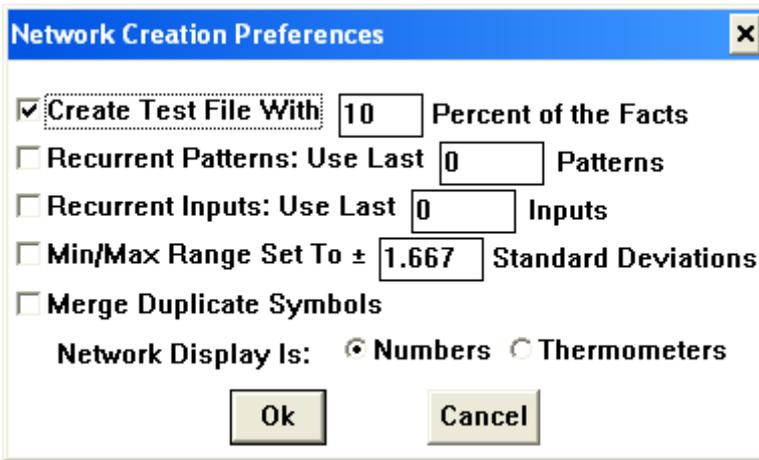
Требуется: по приведенным в таблице RE.dat данным обучить сеть прогнозированию стоимости дома, и определить, по какой цене могут быть проданы ещё 2 дома.

1. Подготовка данных для исследования с помощью нейронной сети.

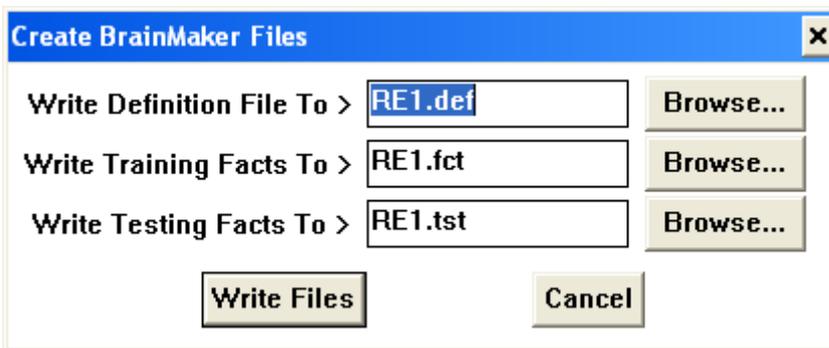
- В NetMaker считывается файл RE.Dat
- Колонка под названием SalePrice помечается, как pattern
- Все остальные колонки помечаются, как input
- После окончания разметки файла с помощью меню Save NetMaker File создаётся файл RE1.Dat

Pattern	Input	Input	Input	Input	Input	Input	Input	Input	Input	Input	Input	Input	Input	Input	Input
SALEPRIC	DWLUN	RDOS	YRBLT	TOTFIXT	HEATING	WBFPSK	BMNTGAR	ATIFRGA	TOTLIVA	DECK/OF	ENCLPOR	NBHDGRP	RECROOM	FINBSMT	
1	140000	1	13	1920	5	2	0	0	0	1440	24	0	1	0	0
2	155000	1	5	1900	5	2	0	0	0	1197	45	0	1	0	0
3	200000	2	3	1910	10	2	0	0	0	2844	293	0	1	0	0
4	157000	1	18	1930	7	2	1	0	0	1428	8	144	1	0	0
5	171000	1	18	1930	6	2	1	0	0	1248	35	84	1	0	0
6	170000	1	10	1900	8	2	0	0	0	1808	0	0	1	0	0
7	168000	1	22	1955	8	2	0	0	0	1430	180	0	1	0	0
8	145000	1	1	1910	5	2	0	0	0	1492	49	0	1	0	0
9	155000	2	23	1920	10	2	0	0	0	1760	220	0	1	0	0
10	167000	2	14	1926	10	2	0	0	0	2254	80	0	1	0	0
11	172000	3	16	1900	15	2	0	0	0	2997	250	140	1	0	0
12	172000	2	22	1920	9	2	0	0	0	2430	90	0	1	0	0
13	155000	1	13	1920	5	2	0	0	0	1435	0	162	1	200	0

- Корректируем тип данных: NetMaker по умолчанию создаёт файлы для BrainMaker, помечая данные, как thermometers. Если изменить сейчас тип данных на numbers, это отразится в BrainMaker – файле, а сохранённого в NetMaker – файле данных это не коснётся. В File – меню находим Preferences. Изменяем в Network Display тип данных на numbers.
- По умолчанию 10% всех данных будут использоваться для контроля (testing), в связи с чем в обучающую выборку (training) попадёт 195 фактов. Поскольку 217 примеров – это выборка маленькая, для контроля можно взять и 5% данных.



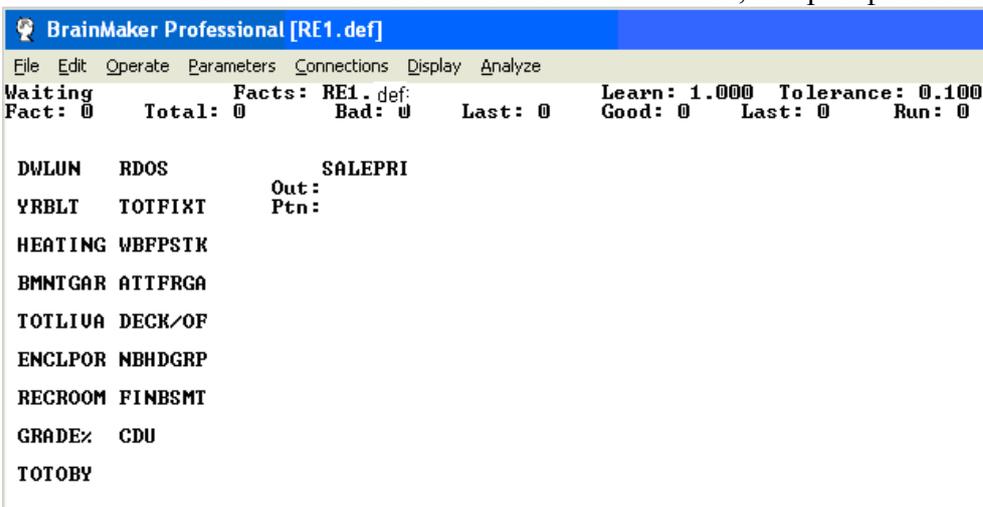
- В File – меню с помощью Create BrainMaker Files создаём BM-файл (содержащий факты для обучения и для тестирования под тем же именем).
- При этом создаются файлы описания сети (Re1.def), обучающая выборка (Re1.fct) и файл для тестирования сети (Re1.tst).



- Покидаем NetMaker и переходим в BrainMaker.

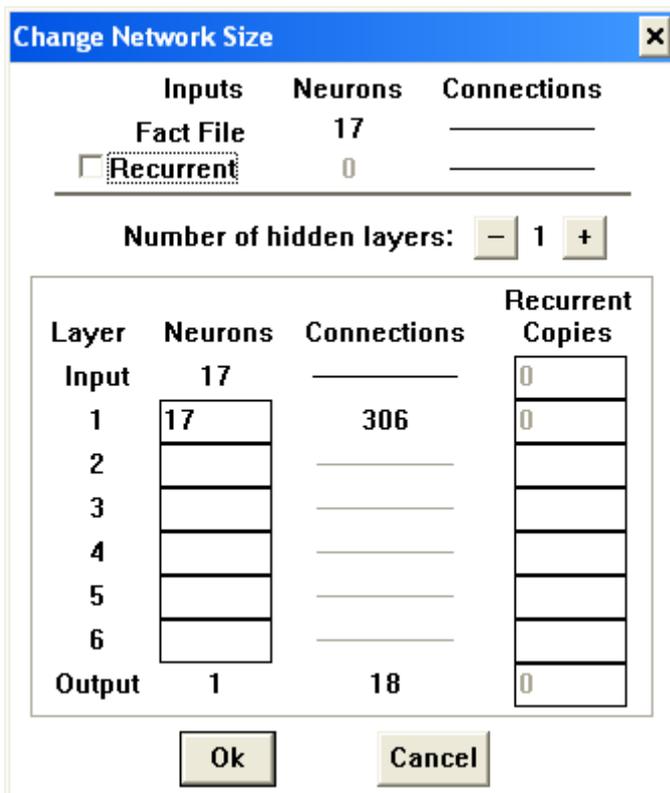
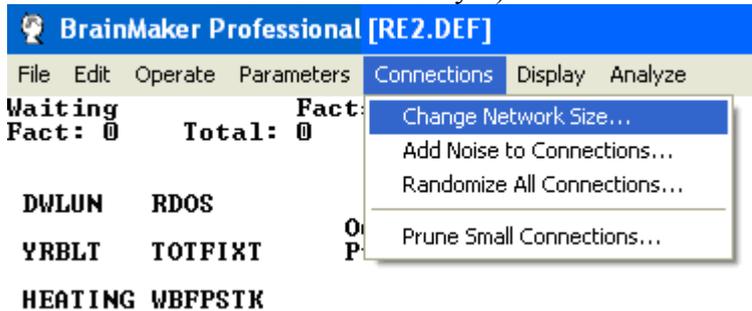
2. Подготовка сети к работе.

- В File – меню выполняем Read Network, выбрав файл RE1.Def



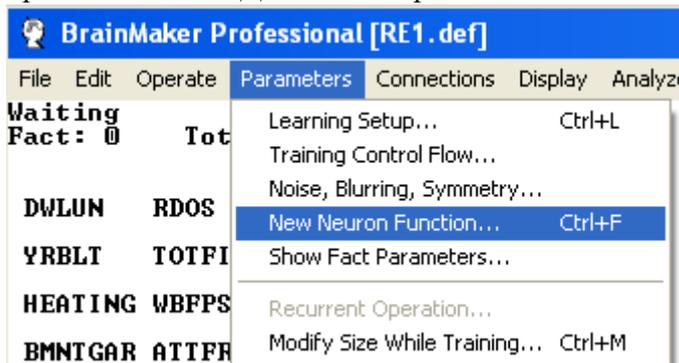
- Определяем размер сети. В этом проекте используется малое количество фактов. Необходимо быть внимательным, чтобы не создать слишком много скрытых нейронов. В Connection – меню устанавливаем размер сети (Change

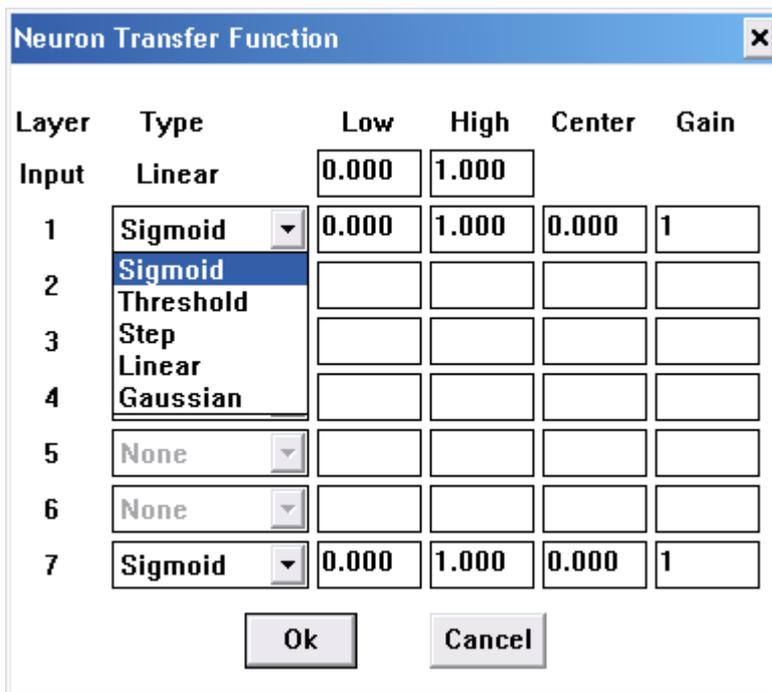
Network Size): 17 входных нейронов (input) и 1 выходной (output).
 Количество скрытых нейронов определяется равным количеству входных нейронов. Скрытые нейроны размещаем на одном уровне (устанавливается в колонке Hidden.Layer).



Здесь указывается, что количество уровней со скрытыми нейронами =1. Количество скрытых нейронов автоматически принимается равным 17.

При необходимости можно установить типы пороговых функций в каждом слое, кроме входного. Для этого переходим в Parameters -> New Neuron Function:

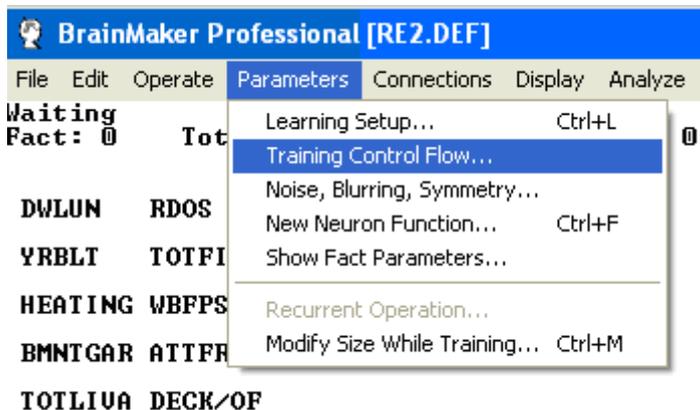




На этом рисунке видны допустимые пороговые функции, открытые для промежуточного слоя: Sigmoid, Treshold, Step, Linear, Gaussian.

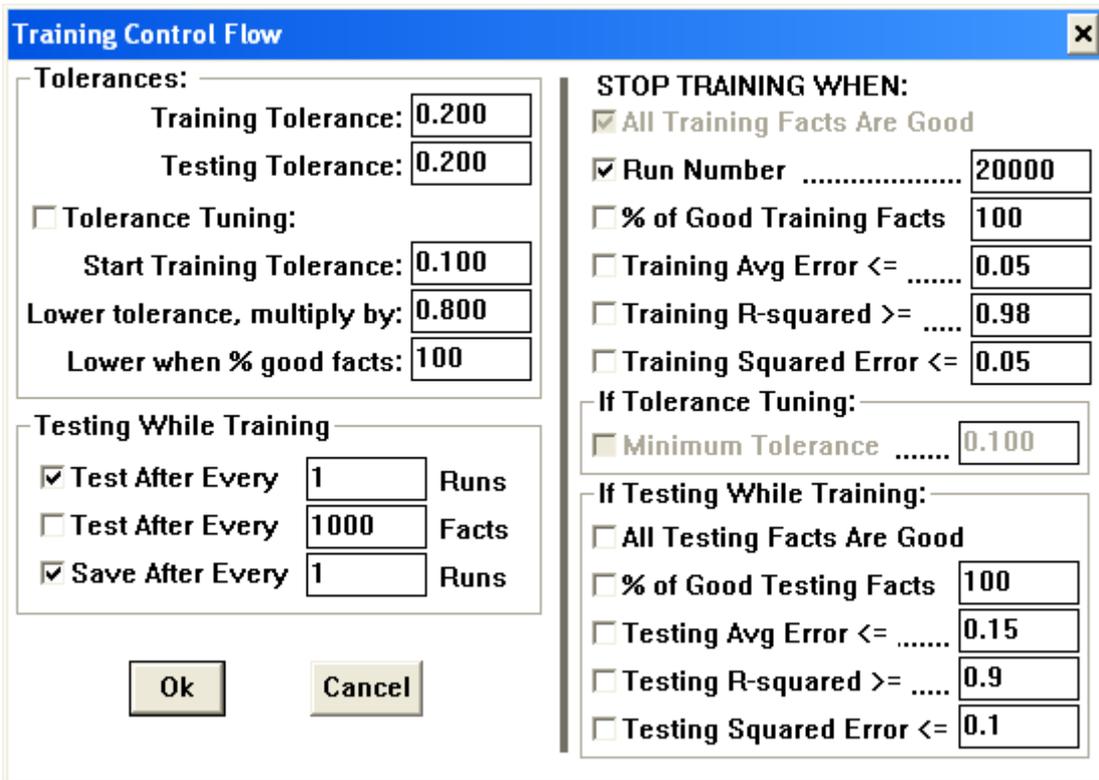
В данном примере оставляем выбранные по умолчанию пороговые функции.

- Устанавливаем допустимые погрешности (tolerance) для обучения и тестирования. В меню Parameters выбираем Training Control Flow. По умолчанию Training Tolerance = .1, что приводит к очень медленной (для IBM PC XT) работе. Чтобы уменьшить время обучения, надо установить «.2».

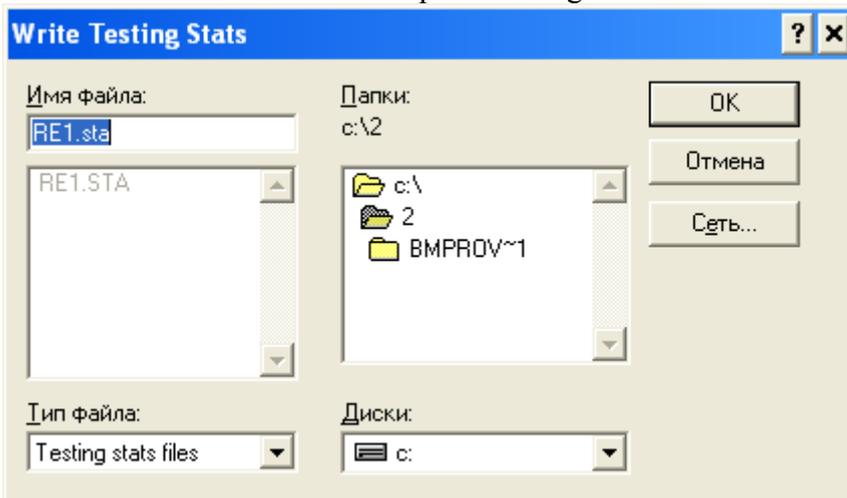


Изменяем так же Testing Tolerance на 0.2,

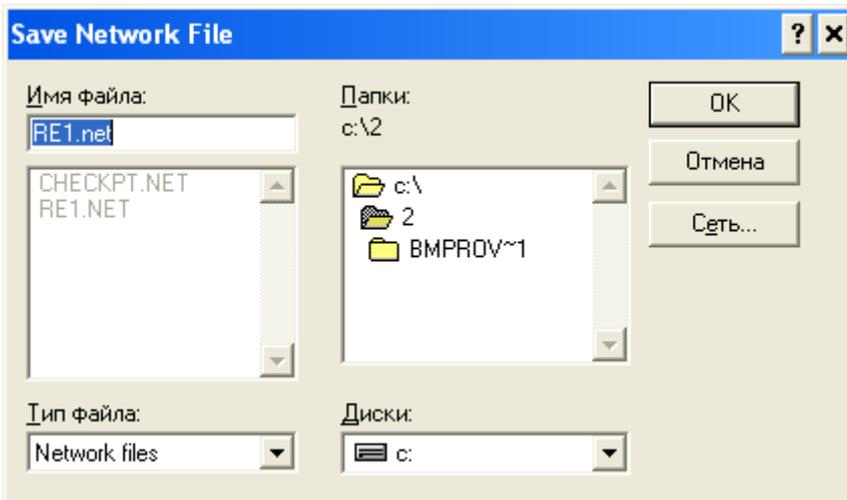
- Test After Every 1 Runs,
- Save After Every 1 Runs,
- Stop Training When: Run Number 20000



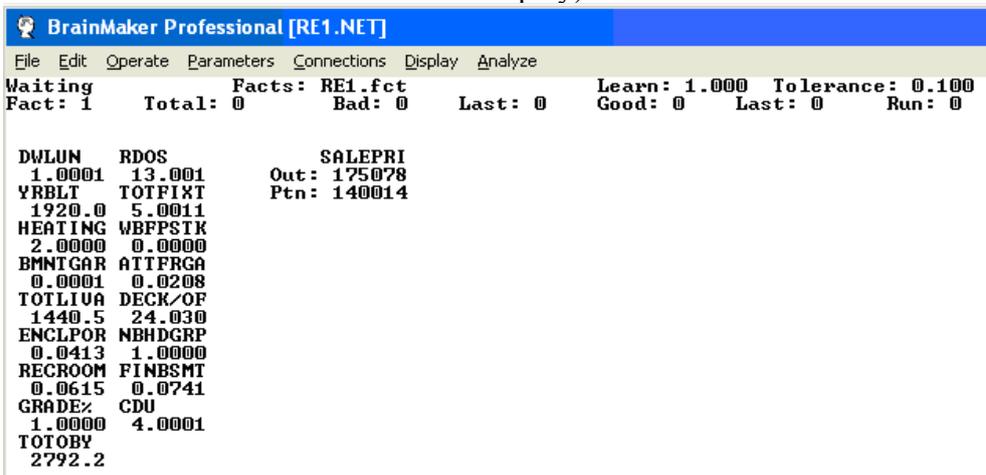
- В меню File выбрать Testing Statistics. Имя оставить по умолчанию.



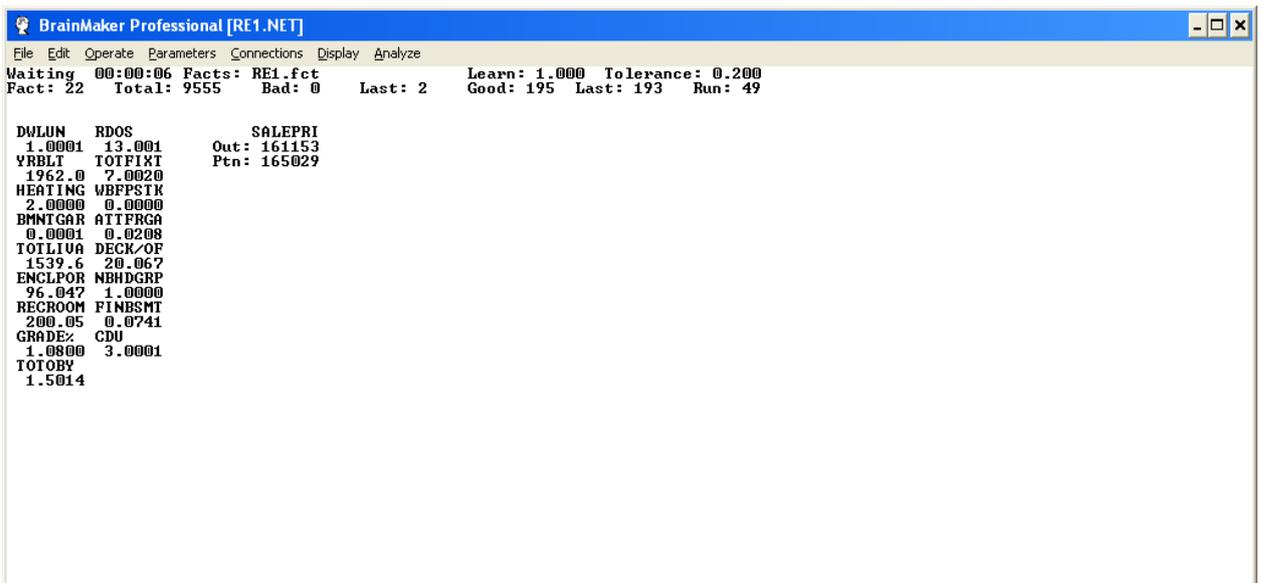
- Сохранить созданную сеть. Из File → Save Network сохранить в файле RE1.Net. (Это очень правильно – периодически сохранять работу сети. При этом облегчается восстановление из-за возникших ошибок, и легче обучать другую версию сети).



- Обучение нейронной сети и чтение экрана BrainMaker.
 - Задать первый обучающий факт: Operate → Get Next Fact. (можно изменить представление данных на термометр через Display Menu → Edit Network Display).



- Стоимость покупки характеризуют два числа: первое – это выход нейросети (output), и второе – желаемый результат обучения (pattern).
- Факты проходят через сеть быстро. В каждой эпохе (run) 195 фактов. Из 217 фактов, прочитанных в NetMaker, 10% - (или 22 факта) будет проверено при тестировании.
- Третья строка экрана показывает состояние (Status): что BrainMaker делает и какой файл использует. Tolerance показывает ошибку сети в каждом режиме (training – testing).
- Четвёртая линия – это линия статистики. Показывает прогресс обучения или тестирования для каждой эпохи.
- Старт обучения: Operate → Train Network. В процессе обучения проверяется, уменьшается ли число плохих фактов.
- Когда сеть заканчивает обучение, на 3 строке экрана появляется слово waiting, а на 4 появляется индикация Bad и Good.

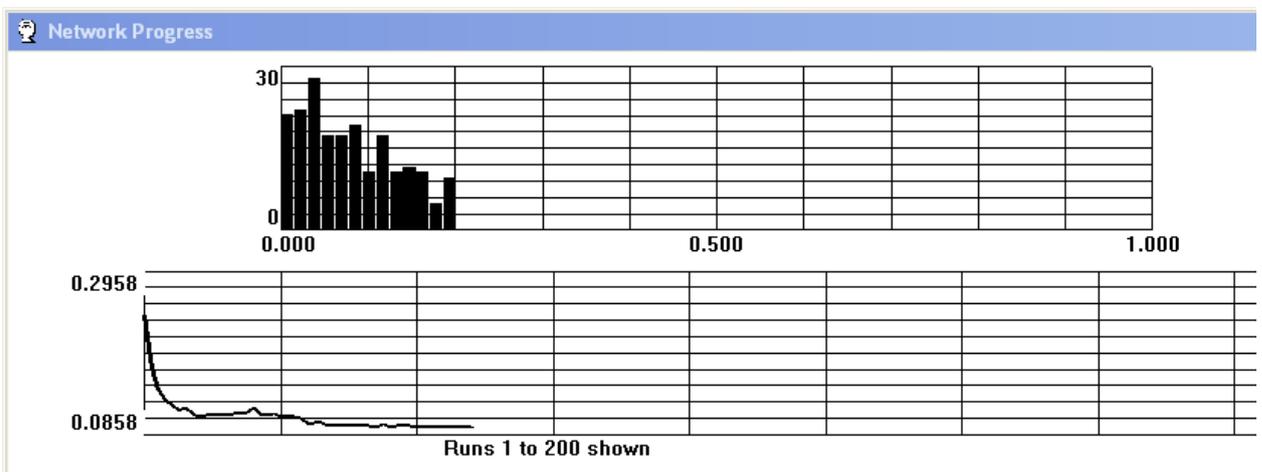
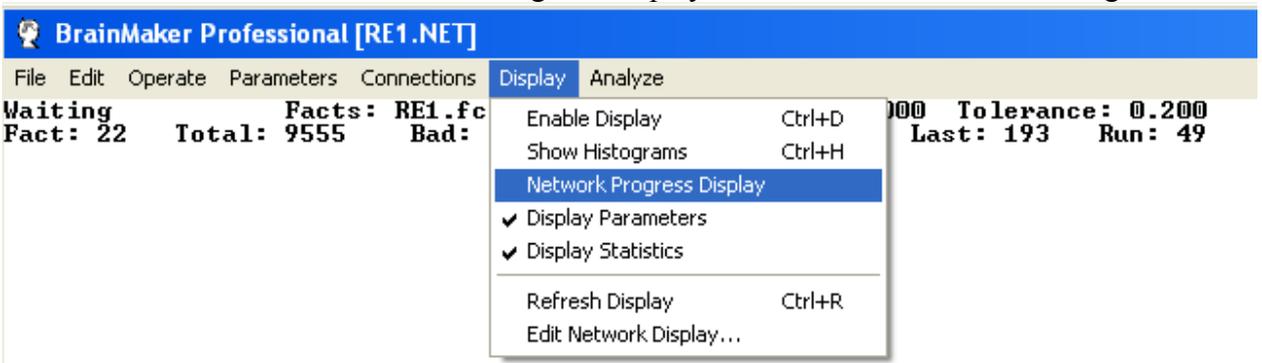


Обучение сети закончилось на 49 эпохе, так как к этому времени выполнилось условие All Training Facts Are Good.

- Отключение демонстрации фактов и включение демонстрации хода обучения. Переключатель Display → Enable Display отключает демонстрацию каждого факта.

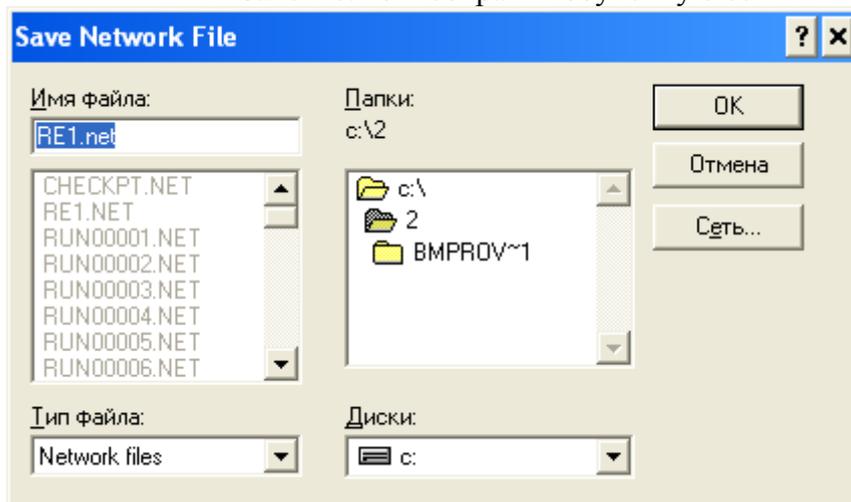


- С помощью Network Progress Display выводится окно Network Progress.



В верхней части окна выводится гистограмма, показывающая распределение ошибок обучения. На горизонтальной оси откладывается уровень ошибок, а на вертикальной оси – количество выходных значений этого уровня.

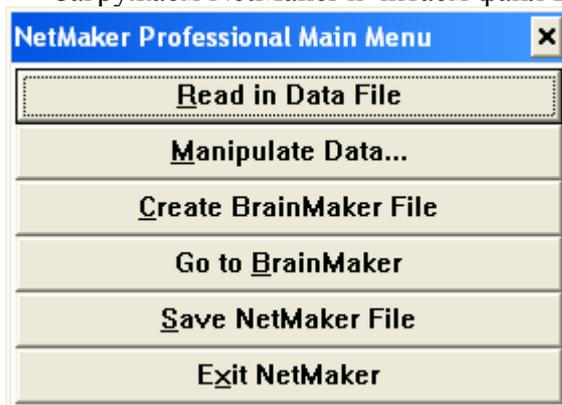
- В конце обучения количество ошибок уменьшается до уровня .2 . Может оказаться, что уровень ошибок снижается не до уровня .2, а до .225.
- Нижний график демонстрирует прогресс при обучении сети. По горизонтальной оси отложено количество запусков (эпох). По вертикальной оси откладывается количество RMS – ошибок, которые превышают среднее значение при малом количестве фактов.
- При увеличении номера эпохи количество ошибок уменьшается.
- Сохранение сети и переход к NetMaker. Когда сеть закончит обучение, File → Save Network сохранит обученную сеть в RE1.Net.



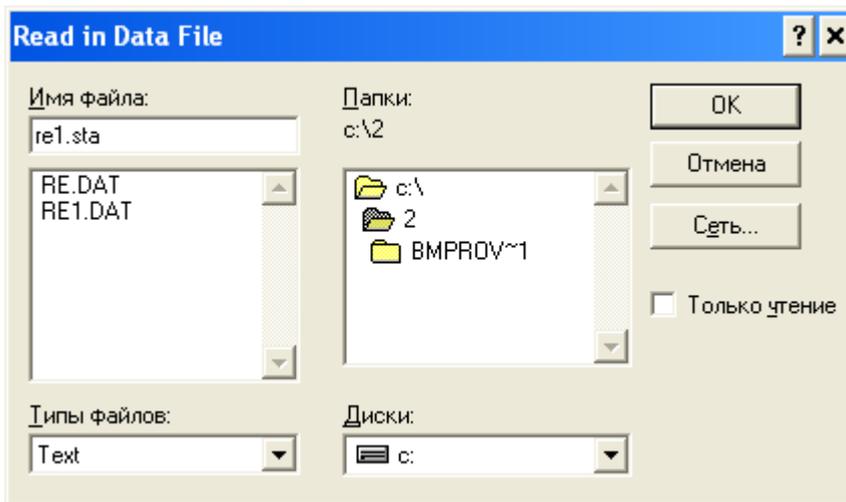
- На этом обучение заканчивается. Выходим из BrainMaker: File → Exit.

3. Анализ процесса обучения с помощью статистики.

- Загружаем NetMaker и читаем файл RE1.Sta. Это – текстовый файл.



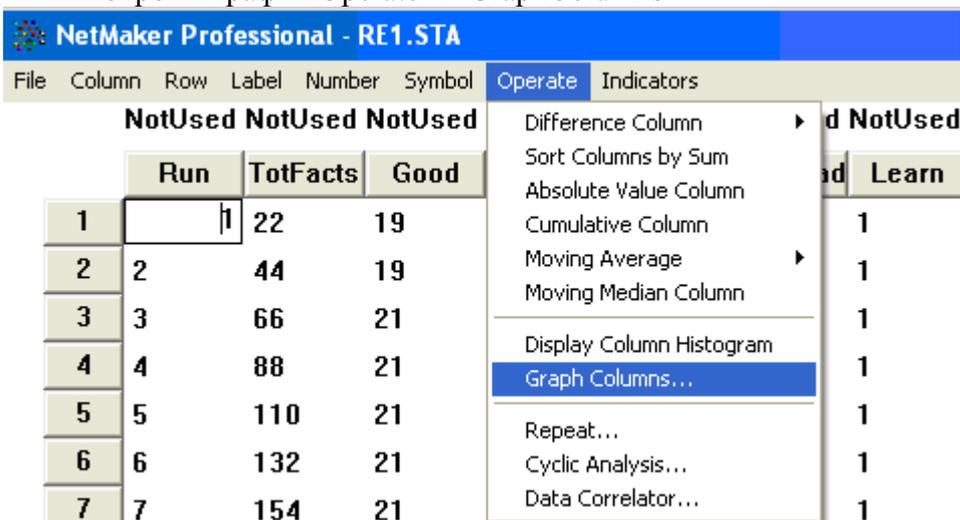
Набираем имя нужного файла:



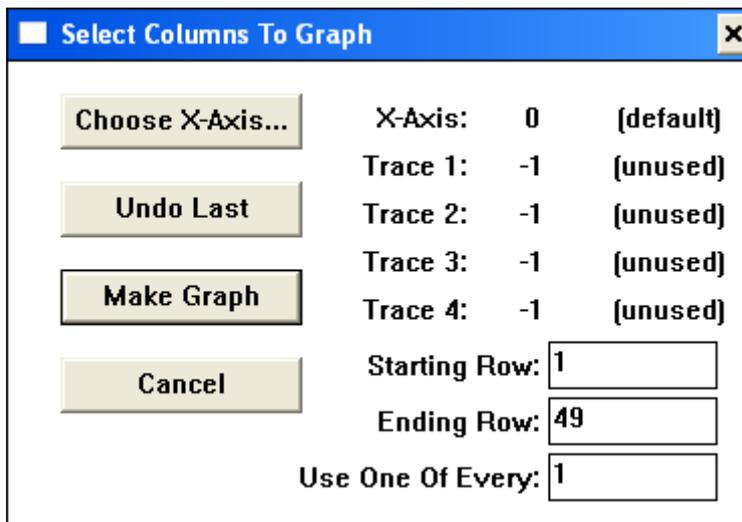
Происходит его загрузка. Переходим к Manipulate Data:

	Run	TotFacts	Good	Bad	adOutput	TotalBad	Learn	tolerance	AvgError	RMSErr	hh:mm:ss	SALEP	d:SALEP
1	1	22	13	9	9	9	1	0.2	0.1798	0.2009	00:00:00	0.426	9
2	2	44	17	5	5	14	1	0.2	0.1236	0.1493	00:00:00	0.4996	5
3	3	66	20	2	2	16	1	0.2	0.1132	0.1322	00:00:00	0.5192	2
4	4	88	20	2	2	18	1	0.2	0.0982	0.1182	00:00:00	0.5393	2
5	5	110	20	2	2	20	1	0.2	0.0939	0.1151	00:00:01	0.5504	2
6	6	132	18	4	4	24	1	0.2	0.1097	0.1411	00:00:01	0.5188	4
7	7	154	18	4	4	28	1	0.2	0.108	0.1399	00:00:01	0.5096	4
8	8	176	19	3	3	31	1	0.2	0.098	0.1252	00:00:01	0.5261	3
9	9	198	18	4	4	35	1	0.2	0.1032	0.1322	00:00:01	0.5186	4
10	10	220	18	4	4	39	1	0.2	0.1042	0.1341	00:00:01	0.517	4
11	11	242	19	3	3	42	1	0.2	0.1034	0.1337	00:00:01	0.506	3
12	12	264	19	3	3	45	1	0.2	0.1061	0.1383	00:00:01	0.4897	3
13	13	286	19	3	3	48	1	0.2	0.1066	0.1369	00:00:01	0.4922	3
14	14	308	19	3	3	51	1	0.2	0.108	0.1378	00:00:01	0.4966	3

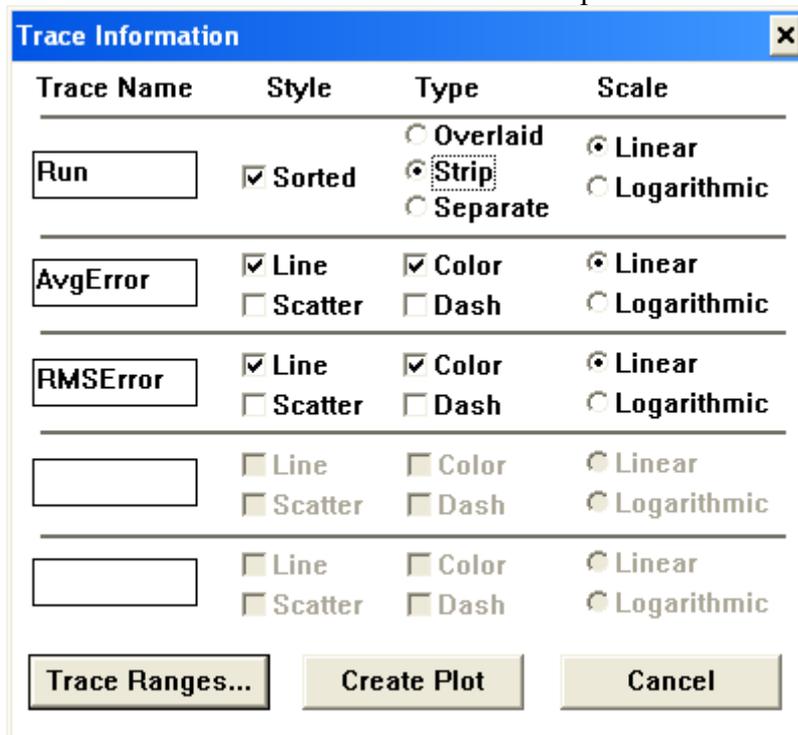
- Строим график Operate → GraphColumns



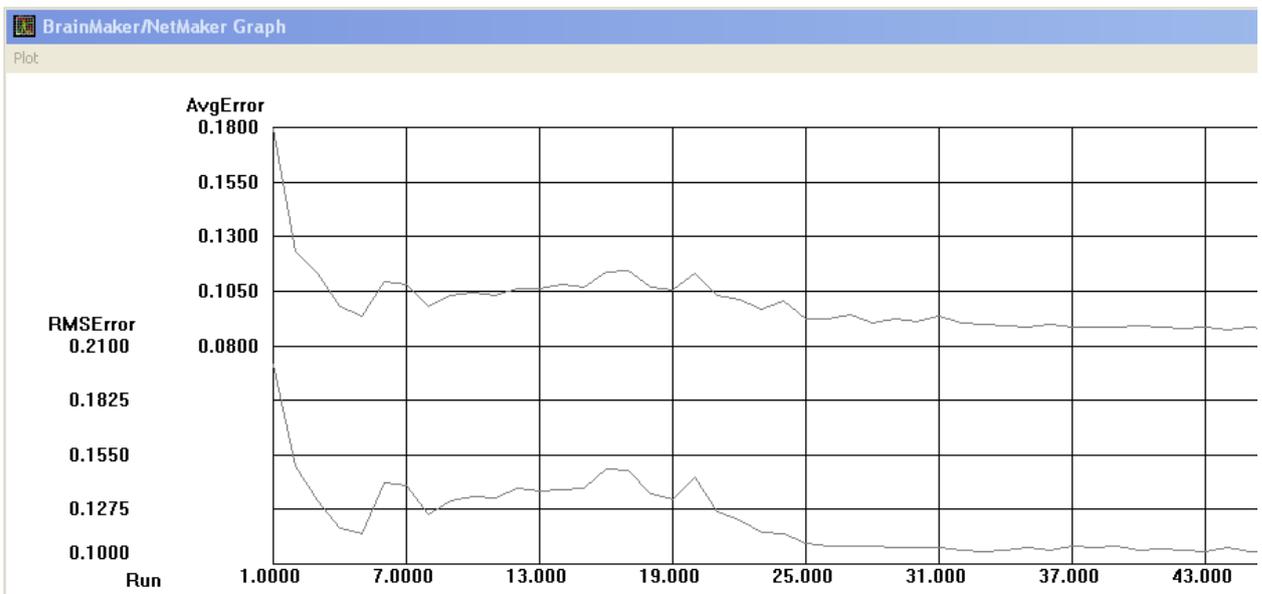
Появляется окно для настройки:



- Выделяем Choose X-Axis (ось X) и в основном окне NetMaker отмечаем нужную колонку - Run.
- Трогаем левой кнопкой мыши надпись Trace 1, после чего трогаем в NetMaker AvgError.
- Затем трогаем надпись Trace 2, после чего трогаем RMSError. В результате этих действий произведена разметка графика: по оси X откладывается Run (ось X показывает эпоху); по оси Y в первом графике откладывается AvgError (Верхний график демонстрирует (Average) среднее значение ошибки), во втором графике – RMSError (Нижний демонстрирует RMS-ошибку).
- Нажимаем Make Graph. Появляется окно для уточнения информации:

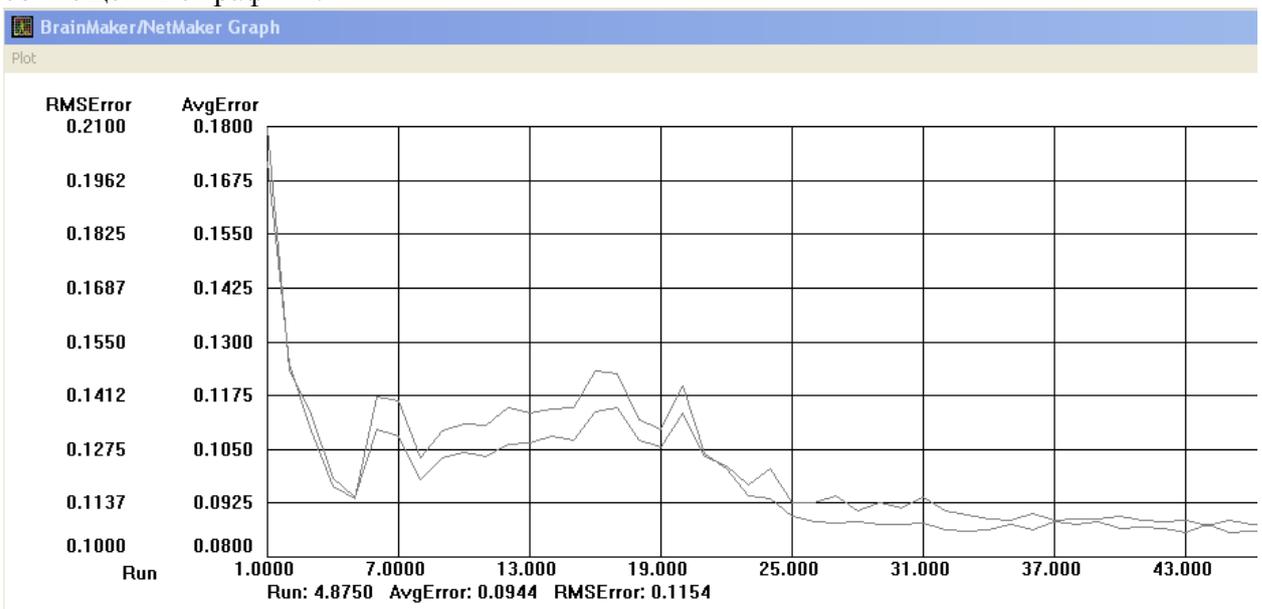


Заменяем Overlaid на Strip и → Create Plot.

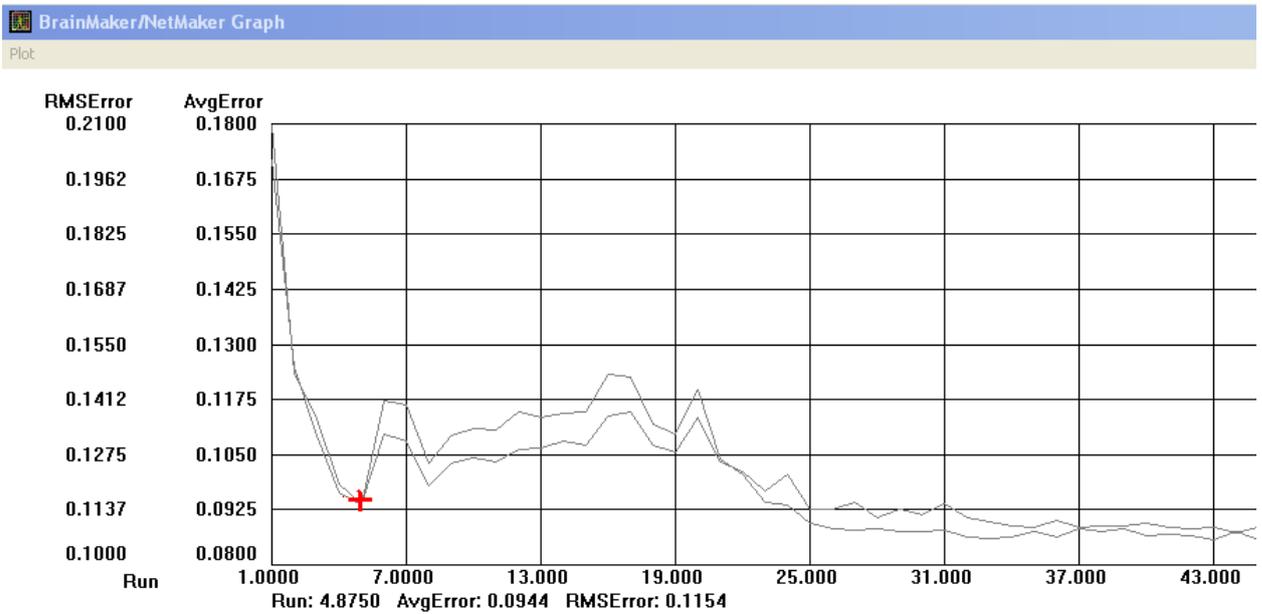


- Обе ошибки снижаются. Наилучший результат теста достигнут между 4 и 6 эпохами, и после 37 эпохи.

Если в окне настройки после Make Graph выбрать не Strip, а Overlaid, получим совмещённые графики:



На этом графике так же видно, что наилучший результат получен где-то в районе 4 -6 эпохи. Устанавливаем крестик в самую нижнюю точку на графике RMSError и в самой нижней строке читаем результат:



Вернёмся к окну для уточнения информации:

Trace Name	Style	Type	Scale
Run	<input checked="" type="checkbox"/> Sorted	<input type="radio"/> Overlaid <input checked="" type="radio"/> Strip <input type="radio"/> Separate	<input checked="" type="radio"/> Linear <input type="radio"/> Logarithmic
AvgError	<input checked="" type="checkbox"/> Line <input type="checkbox"/> Scatter	<input checked="" type="checkbox"/> Color <input type="checkbox"/> Dash	<input checked="" type="radio"/> Linear <input type="radio"/> Logarithmic
RMSError	<input checked="" type="checkbox"/> Line <input type="checkbox"/> Scatter	<input checked="" type="checkbox"/> Color <input type="checkbox"/> Dash	<input checked="" type="radio"/> Linear <input type="radio"/> Logarithmic
	<input type="checkbox"/> Line <input type="checkbox"/> Scatter	<input type="checkbox"/> Color <input type="checkbox"/> Dash	<input type="radio"/> Linear <input type="radio"/> Logarithmic
	<input type="checkbox"/> Line <input type="checkbox"/> Scatter	<input type="checkbox"/> Color <input type="checkbox"/> Dash	<input type="radio"/> Linear <input type="radio"/> Logarithmic

Trace Ranges... Create Plot Cancel

Выбираем Trace Ranges:

Trace Ranges ✕

	Plot Min	Plot Max	Units/Div	Data Min	Data Max
Run	1	49	6	1	49
AvgError	0.08	0.18	0.025	0.087	0.1798
RMSError	0.1	0.21	0.0275	0.1062	0.2009
XAxis			1.2500	1.2500	1.2500
XAxis			1.2500	1.2500	1.2500

Plot Range Functions

Set rounded ranges

Set equal to exact data range

Set all traces to same range

Trace Info...

Create Plot

Cancel

Нажимаем Set all traces to same range

Trace Ranges ✕

	Plot Min	Plot Max	Units/Div	Data Min	Data Max
Run	1	49	6	1	49
AvgError	0.08	0.21	0.0325	0.087	0.1798
RMSError	0.08	0.21	0.0325	0.1062	0.2009
XAxis			1.2500	1.2500	1.2500
XAxis			1.2500	1.2500	1.2500

Plot Range Functions

Set rounded ranges

Set equal to exact data range

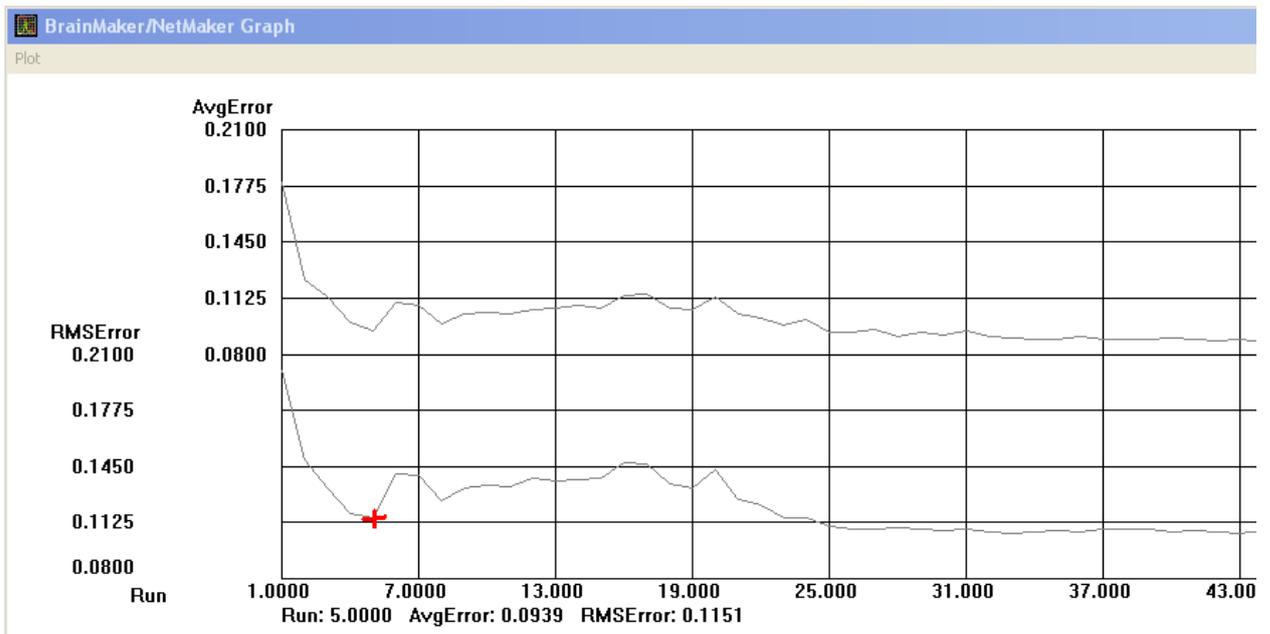
Set all traces to same range

Trace Info...

Create Plot

Cancel

Нажимаем Create Plot для построения графика и устанавливаем крестик в минимальное значение RMSError:

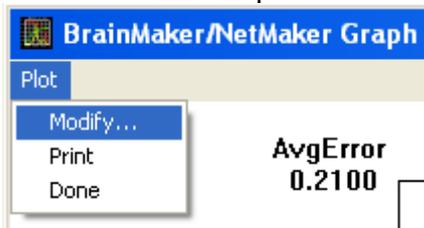


Читаем значения Run, AvgError и RMSError и заносим их в таблицу:

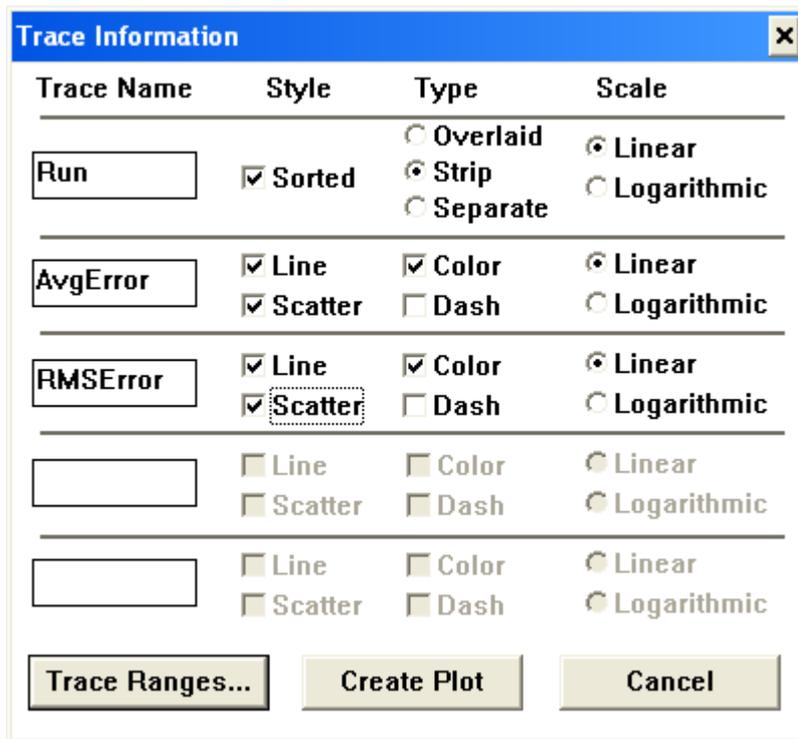
Run	AvgError	RMSError
5.0000	0,0939	0,1151

На RMSError минимальное значение достигнуто уже на 5 эпохе. Возрастание ошибки в дальнейшем может быть связано с «переучиванием» сети. Желательно обучение сети на 5 эпохе закончить.

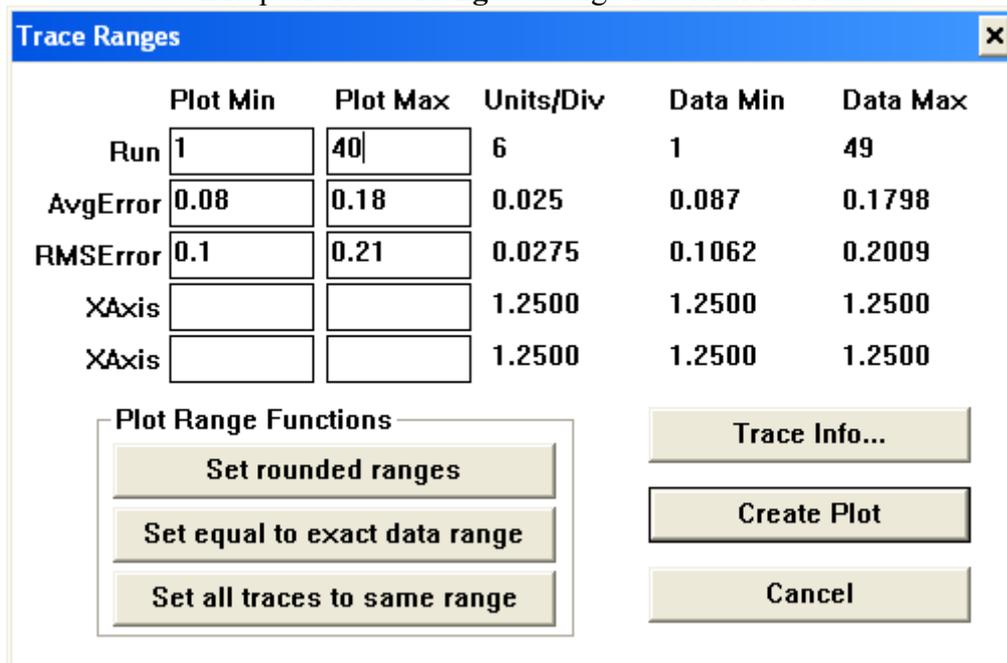
- Улучшение графика.
 - Выбрать Plot → Modify.



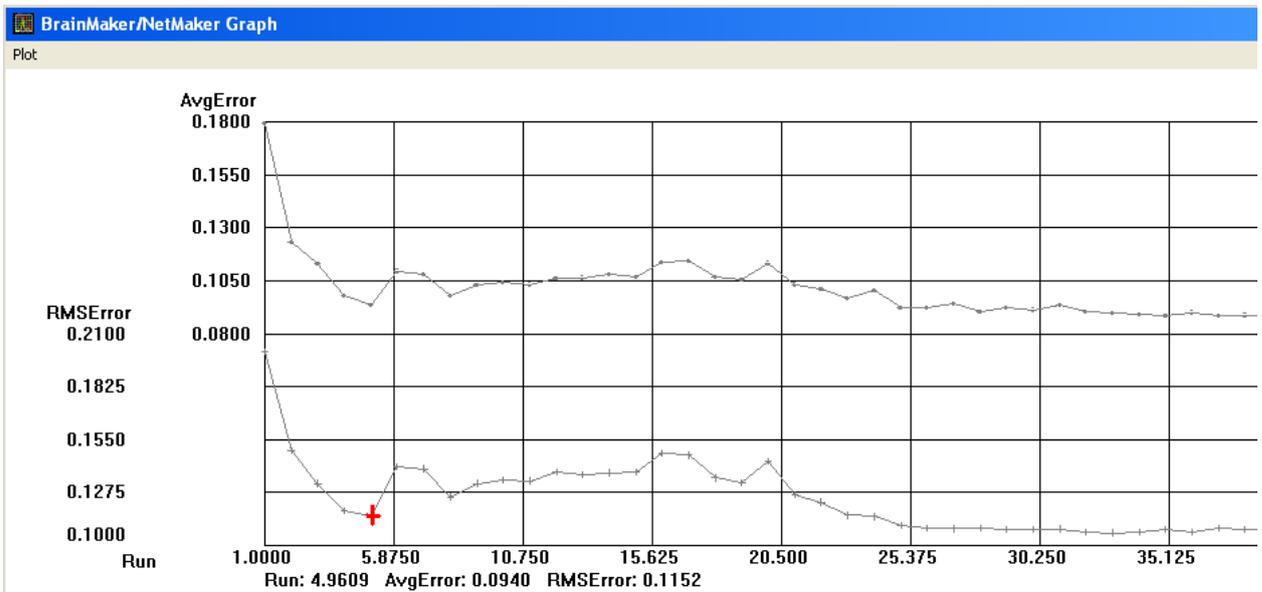
- Для того, чтобы идентифицировать требуемую эпоху, выбираем Scatter для обеих трасс. Это добавит маркеры вдоль каждой трассы в требуемом диапазоне.



- Выбрать **Trace Ranges**. Change Plot Max For Run to 40



Переходим к → Create Plot.



- Перемещаем курсор мыши в нижнюю точку трассы.
- Данные об эпохе, такие, как номер эпохи, среднюю ошибку и RMS – ошибку выдаёт пункт the cursor readout. Перемещая курсор по горизонтали с одним из scatter маркером увидим уточнённые результаты для каждой эпохи.
- Находим наилучший результат

Run	AvgError	RMSError
5	.0940	.1154
33	.0898	.1066

- Закрываем окно.
- Теперь посмотрим статистику обучения.
 - Через File → Read in Data File читаем файл статистики обучения RE1.Sta. Это текстовый файл. Если получение этого файла не было заказано, то создаются файлы RunN.net, где N – номер эпохи.
 - Смотрим на AVG и RMS – ошибки для эпох 5 и 33. Статистика обучения лучше на эпохе 33.

5 0.0939 0.1151
 33 0.0899 0.1066

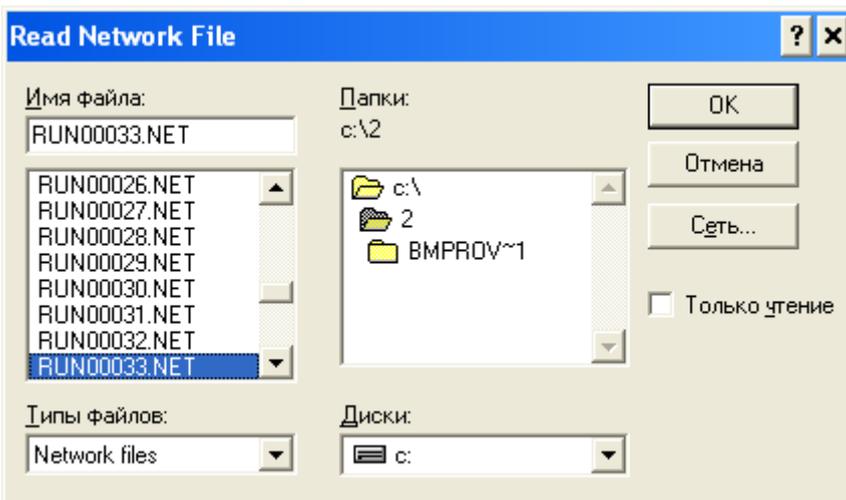
После 33 эпохи роста ошибки сети не наблюдается. Вполне возможно, что рост ошибки после 5 эпохи объясняется не переобучением сети, а другими причинами. Поэтому будем считать, что обучение можно закончить на 33 эпохе.

- На этом заканчиваем работу с сетью.
- Покидаем NetMaker.

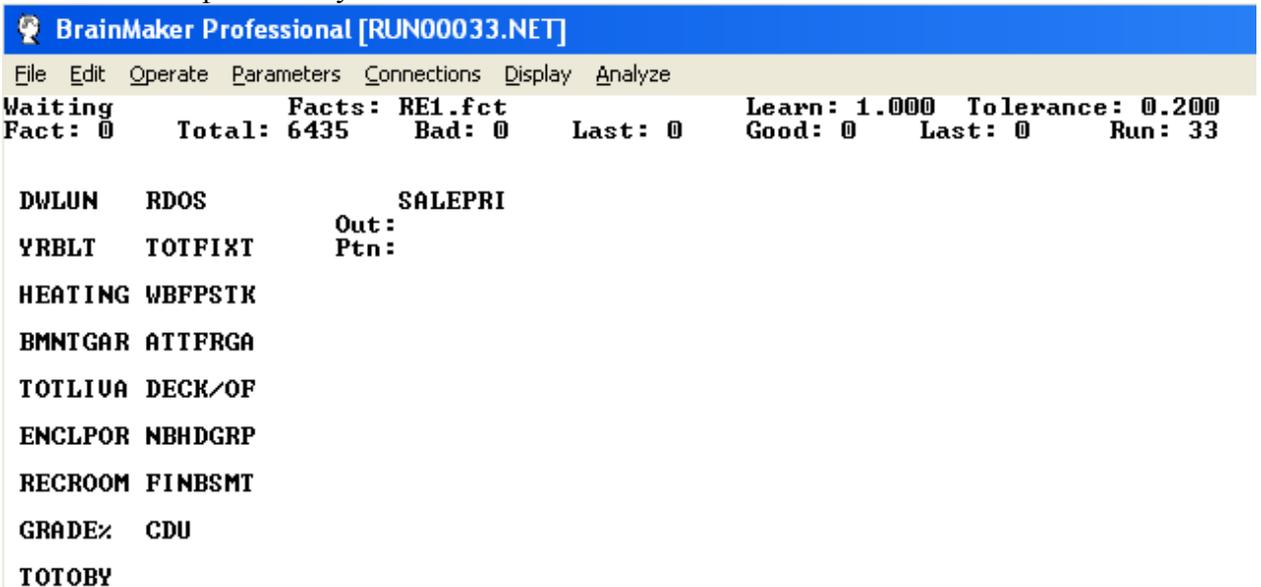
4. Получение идеальной сети.

Возвращаемся к закрытой сети. Продолжаем обучение до идеальной эпохи.

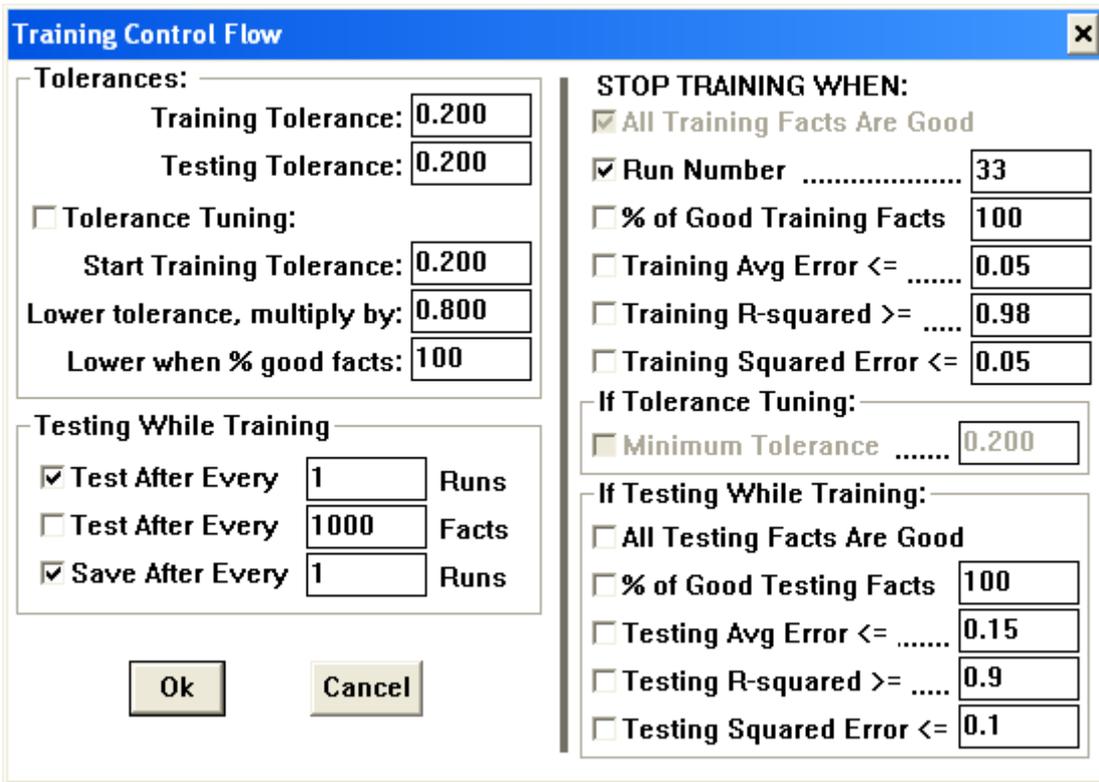
- Вызываем VM.
 - Через File → Read Network. Останавливаемся на имени Run00033.Net (33 эпоха).



После чтения файла получаем:



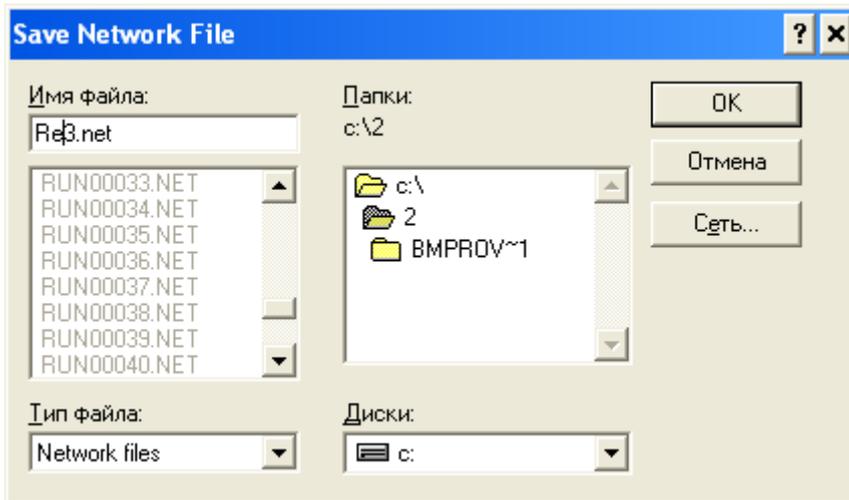
- Выбираем Parameters → Training Control Flow.
- Изменяем номер эпохи на 33. Для этого включаем Stop Training When... (остановить обучение на значении...), устанавливаем Run Number 33.



- Повторяем обучение сети: Operate → Train Network.



- После окончания обучения сохраняем сеть: File → Save Network в файле RE3.Net.



- Покидаем VM: File → Exit.

Теперь сеть можно использовать для предсказания цены дома, выставленного на продажу. Данные помещаются в файл в том же формате, что и исходный файл, использованный для создания сети, но не содержат указаний учителя (Training Pattern) в виде SilesPrice.

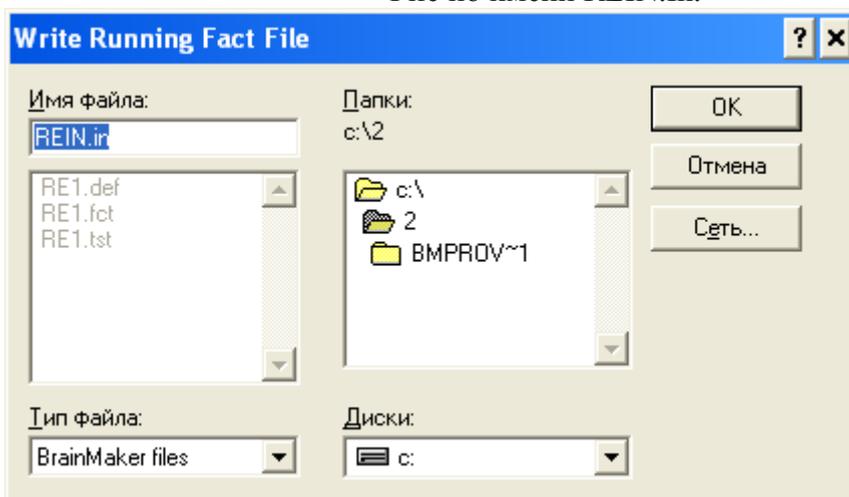
Такой файл необходимо создать с помощью NetMaker. Затем перейти к исследованию с помощью нейронной сети через VM.

5. Создание файла для исследования.

- Создаём файл REIN.dat, содержащий данные для нейросетевого анализа. Входим в NM и из файла RE.dat создаём REIN.dat. Для проверки качества прогноза предложим сети для прогнозирования данные о продаже первых двух домов (по ним известно, за сколько они были проданы):

File	Column	Row	Label	Number	Symbol	Operate	Indicators	NotUsed														
			SALEPRIC	DWLUN	RDOS	YRBLT	TOTFIXT	HEATING	VBFPSI	S	IMNTGAJ	ATTFRGA	R	TOTLIVA	R	DECKJOF	P	INCLPOF	IBHDGR	ICROON	FINBSMT	GRADE%
1			140000	1	13	1920	5	2	0	0	0	1440	24	0	1	0	0	1	4	2790	0	0
2			155000	1	5	1900	5	2	0	0	0	1197	45	0	1	0	0	1	4	5890	0	0

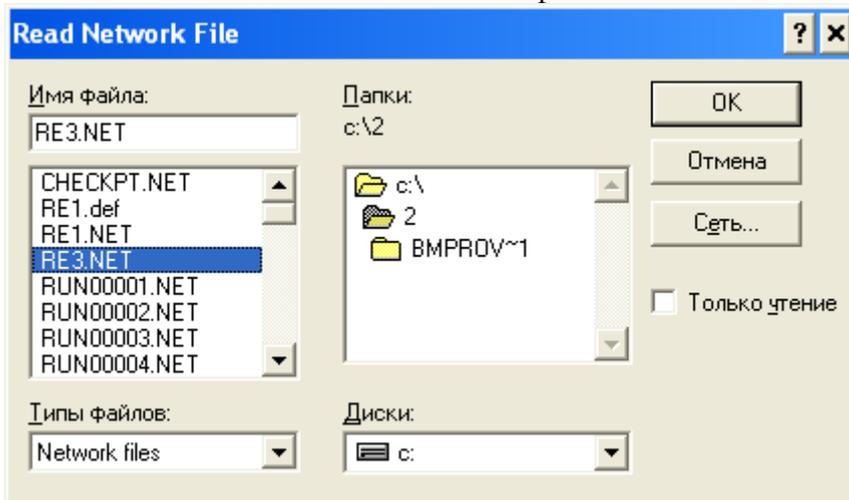
- читаем файл REIN.Dat .
- Переходим к Manipulate Data.
- Помечаем все колонки, как Input.
- Создаём файл фактов для запуска File --> Create Running Fact File по имени REIN.In.



- Покидаем NM и переходим к VM.

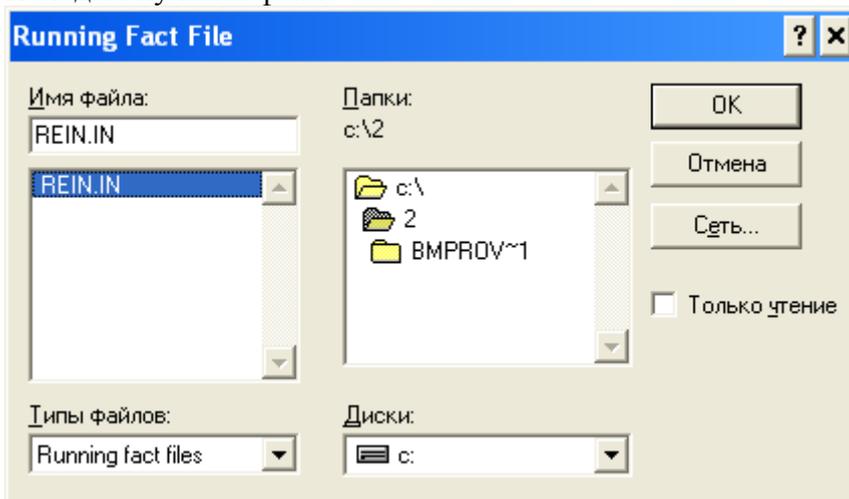
6. Запуск обученной сети. Проведение исследования.

- Входим в VM и читаем файл RE3.Net.

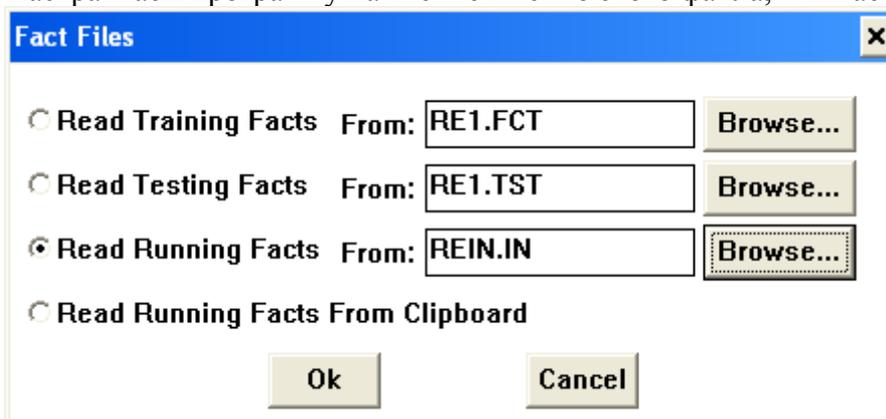


- Читаем файл с исследуемыми данными: File → Select Fact File (REIN.In).

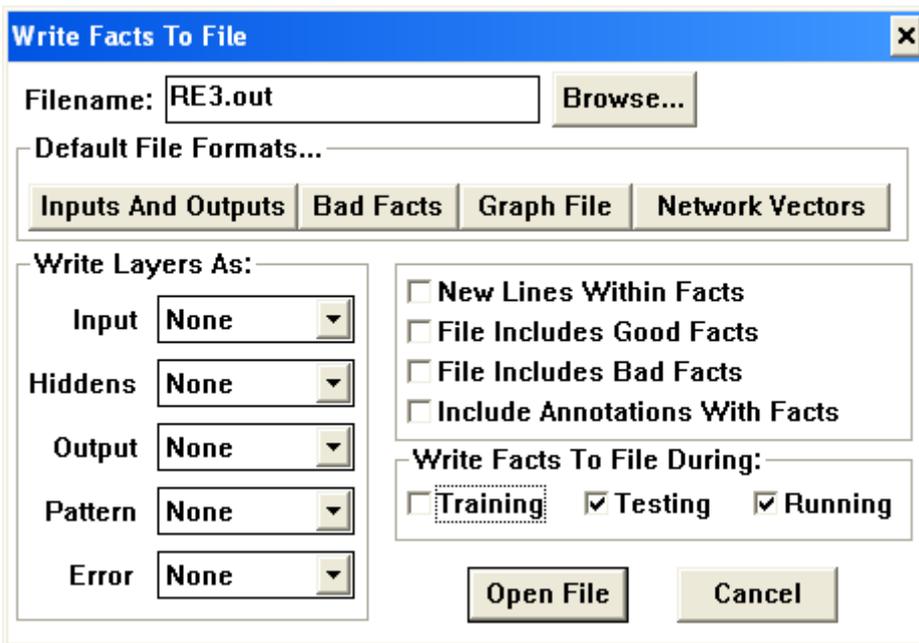
Находим нужный файл:



Настраиваем программу на чтение именно этого файла, и читаем:



- Готовим файл протокола исследования RE3.Out. File → Facts to File.



Нажимаем Inputs And Outputs, затем – Open File, Overwrite.

- Почти все параметры проходят по умолчанию. Нужно, чтобы Write Facts to File During box – был включён.
- Запускаем исследование: Operate → Run Trained Network.



- На экране виден результат прогнозирования: выход сети равен 134523. Это есть вторая прогнозируемая цена дома. Реальная цена была зафиксирована 155000.
- В файле RE3.out находится и первый прогноз. Фактическая цена была зафиксирована 140000.

Содержание файла Re3.out:

```

L ----- 1
1.0001 13.001 1920.0 5.0011 2.0000 0.0000 0.0001 0.0208 1440.5 24.030 0.0413 1.0000
0.0615 0.0741 1.0000 4.0001 2792.2
140337
L ----- 2
1.0001 4.9996 1900.0 5.0011 2.0000 0.0000 0.0001 0.0208 1197.3 45.111 0.0413 1.0000
0.0615 0.0741 1.0000 4.0001 5891.2
134523

```

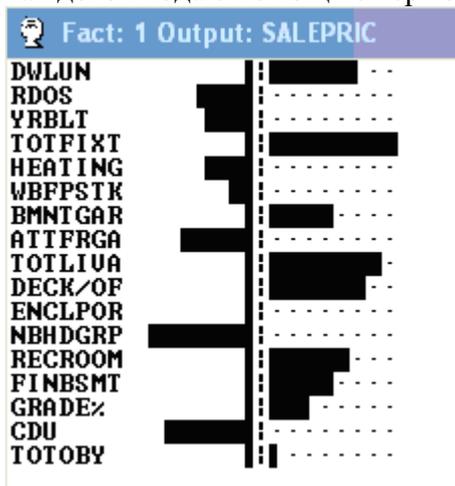
7. Анализ работы сети.

ВМ может сообщить, какие входы были наиболее эффективны. На этом основании можно изменить исследуемую модель, исключить факторы, не влияющие на цену, и т.д.

- Проверка чувствительности всех входов производится через меню Analyze → Sensitivity of All Inputs. Для уточнения надо указать, чувствительность какого входа исследуется:

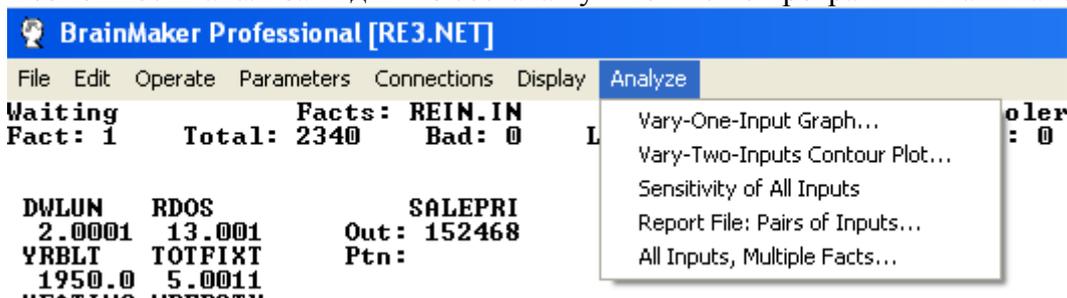


Результат исследования представляется в виде графика с указанием степени влияния каждого входа с помощью термометра:



- ВМ укажет чувствительность входов с точностью +/-10%. Результат виден на графике (термометр) через меню Analyzing. Термометр влево укажет на негативное влияние на цену.
- Закрываем окно, когда граф становится ненужным.

Возможности анализа видны из состава пунктов меню программы BrainMaker:



Активизация пункта Report File: Pairs of Input приводит к выводу таблицы, показывающей, в каких пределах изменяется прогноз при соответствующих изменениях входов:

Fact Number: 1				
Output #1 (SALEPRIC) value: 152468				
		-10%	+10%	Avg Change
Input #4:	(TOTFIXT)	-2619.8	2799.3	2709.5
Input #9:	(TOTLIVAR)	-2440.4	2440.4	2440.4
Input #10:	(DECK/OFP)	-1973.8	2117.4	2045.6
Input #12:	(NBHDGRP)	1973.8	-1937.9	-1955.9
Input #1:	(DWLUN)	-1794.4	1830.3	1812.3
Input #13:	(RECROOM)	-1866.2	1650.8	1758.5

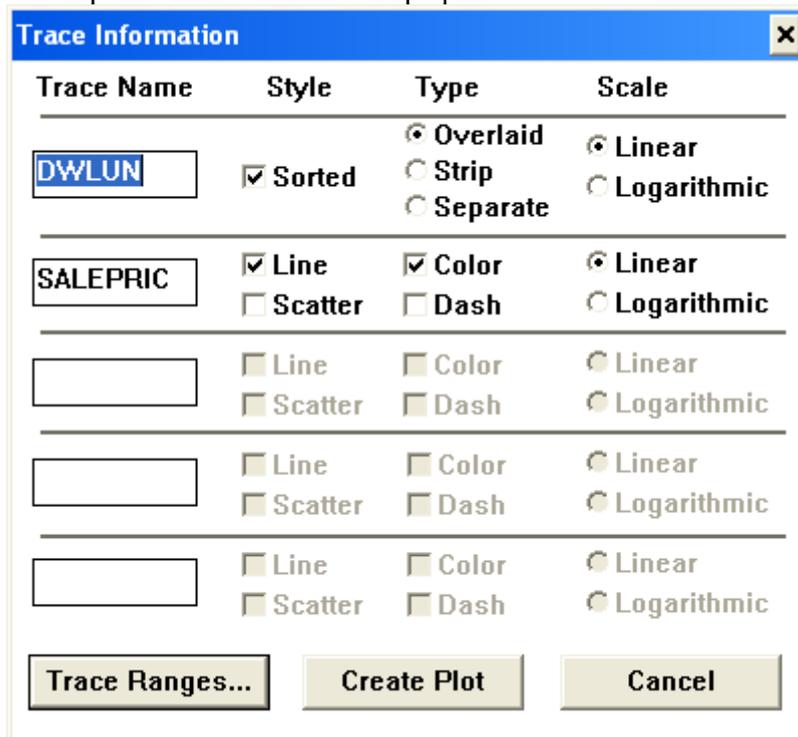
				Changed By	
Inputs	+#4	+#9	+(TOTFIXT)	+(TOTLIVAR)	5275.6
Inputs	+#4	+#10	+(TOTFIXT)	+(DECK/OFP)	5024.4
Inputs	-#4	-#9	-(TOTFIXT)	-(TOTLIVAR)	-4988.5
Inputs	+#4	-#12	+(TOTFIXT)	-(NBHDGRP)	4880.8
Inputs	+#4	-#16	+(TOTFIXT)	-(CDU)	4844.9
Inputs	+#1	+#4	+(DWLUN)	+(TOTFIXT)	4773.1

Активизация пункта All Inputs, Multiple Facts приводит к выводу отчёта о чувствительности к изменению указанного входного множества:

Активизация пункта Vary-Two-Inputs предоставит график влияния двух входных величин на выход:

Активизация пункта Vary-One-Inputs предоставит график влияния указанного входного значения на значения выходных нейронов:

с настройкой особенностей графика:



ПЗ 4. Ирисы Фишера.

Постановка задачи.

Рассматриваются три вида цветков ириса: *Setosa*, *Versicolor* и *Virginica*. Всего имеется по 50 экземпляров каждого вида и для каждого из них измерены четыре величины: длина и ширина чашелистика, длина и ширина лепестка.

Цель состоит в том, чтобы научиться определять тип нового цветка по данным таких измерений.

Эта задача интересна сразу по нескольким причинам. Во-первых, один из классов (*Setosa*) линейно отделим от двух других. Оставшиеся два разделить гораздо труднее, кроме того, имеется несколько экземпляров, выбивающихся из общей картины, которые легко отнеести к другому классу. Такие выбросы являются хорошей проверкой надежности работы сети на сложных данных.

Полный текст исходных данных можно найти в пакете Statistica фирмы StatSoft в файле: ...\Examples\Datasets\ Irisdat.sta и IrisSNN.sta:

Данные: Irisdat.sta (5v * 150с)					
Fisher (1936) iris data: length & width of sepals and petals, 3 types of Iris					
	1	2	3	4	5
	SEPALLEN	SEPALWID	PETALLEN	PETALWID	IRISTYPE
1	5,0	3,3	1,4	0,2	SETOSA
2	6,4	2,8	5,6	2,2	VIRGINIC
3	6,5	2,8	4,6	1,5	VERSICOL
4	6,7	3,1	5,6	2,4	VIRGINIC
5	6,3	2,8	5,1	1,5	VIRGINIC
6	4,6	3,4	1,4	0,3	SETOSA
7	6,9	3,1	5,1	2,3	VIRGINIC
8	6,2	2,2	4,5	1,5	VERSICOL
9	5,9	3,2	4,8	1,8	VERSICOL
10	4,6	3,6	1,0	0,2	SETOSA
11	6,1	3,0	4,6	1,4	VERSICOL

Поскольку выходная переменная - номинальная с тремя состояниями, некоторые программы (например, *Statistica Neural Network*) будут представлять выходную переменную в виде трёх выходных элементов. Каждое из трех возможных номинальных значений выражается высоким выходом одного из этих элементов и низким - двух оставшихся (например, типу *Setosa* может соответствовать выходной набор {0,97; 0,02; 0,01}).

Как и в случае номинальной переменной с двумя состояниями, точная интерпретация такого результата зависит от значений порогов принятия (*Accept*) и отвержения (*Reject*). Конкретнее, если наибольшее из выходных значений превышает порог принятия, а остальные два - ниже порога отвержения, то номинальной переменной присваивается значение. В противном случае выход считается «неясным» (в таблице с предсказанными значениями ему соответствует пропущенное данное).

Всего файл содержит 150 реализаций, имеющих структуру:

№	SEPALLEN	SEPALWID	PETALLEN	PETALWID	IRISTYPE
	- Длина чашел	- Ширина чашел	- Длина лепестка	- Ширина лепестка	- Тип ириса

При решении поставленной задачи:

1. 130 реализаций из файла исходных данных, перемешанных случайным образом, предъявить сети в качестве обучающего (training) набора

	SEPALLEN	SEPALWID	PETALLEN	PETALWID	IRISTYPE
1	5,0	3,3	1,4	0,2	SETOSA
2	6,4	2,8	5,6	2,2	VIRGINIC
3	6,5	2,8	4,6	1,5	VERSICOL
4	6,7	3,1	5,6	2,4	VIRGINIC
5	6,3	2,8	5,1	1,5	VIRGINIC
6	4,6	3,4	1,4	0,3	SETOSA
...
127	6,5	3,2	5,1	2,0	VIRGINIC
128	6,1	2,9	4,7	1,4	VERSICOL
129	5,6	2,9	3,6	1,3	VERSICOL
130	5,0	2,3	3,3	1,0	VERSICOL

2. 10 случайно отобранных реализаций, не включённых в обучающий набор, предъявить сети в качестве тестового (testing) набора. Например, такие:

1	6,9	3,1	4,9	1,5	VERSICOL
2	6,4	2,7	5,3	1,9	VIRGINIC
3	6,8	3,0	5,5	2,1	VIRGINIC
4	5,5	2,5	4,0	1,3	VERSICOL
5	4,8	3,4	1,6	0,2	SETOSA
6	4,8	3,0	1,4	0,1	SETOSA
7	4,5	2,3	1,3	0,3	SETOSA
8	5,7	2,5	5,0	2,0	VIRGINIC
9	5,7	3,8	1,7	0,3	SETOSA
10	5,1	3,8	1,5	0,3	SETOSA

3. 10 случайно отобранных реализаций, не включённых в обучающий набор, предъявить обученной сети для распознавания. Например, такие:

1	5,5	2,3	4,0	1,3	VERSICOL
2	6,6	3,0	4,4	1,4	VERSICOL
3	6,8	2,8	4,8	1,4	VERSICOL
4	5,4	3,4	1,7	0,2	SETOSA
5	5,1	3,7	1,5	0,4	SETOSA
6	5,2	3,5	1,5	0,2	SETOSA
7	5,8	2,8	5,1	2,4	VIRGINIC
8	6,7	3,0	5,0	1,7	VERSICOL
9	6,3	3,3	6,0	2,5	VIRGINIC
10	5,3	3,7	1,5	0,2	SETOSA

Задание 1.

В постановке задачи задано 4 входных параметра. Проверить и объективно доказать, что размерность входного набора данных можно сократить без ухудшения результатов распознавания.

Задание 2.

Используя исходные данные, изложенные в постановке задачи, создать и сохранить в файле оптимальную обученную сеть. Показать, каким образом можно запустить созданную сеть для распознавания информации из набора данных 3. Сравнить результаты с результатами решения 1 задачи.

Задание 3.

Раньше мы рассматривали задачи классификации - выходная переменная являлась номинальной (каждое наблюдение принадлежало одному из дискретных классов).

Нейронные сети можно использовать и для решения задач регрессии, в которых выход является непрерывной числовой переменной.

В качестве примера предскажем значение *PETALWID* в зависимости от переменных *SEPALLEN*, *SEPALWID* и *PETALLEN* и *Iristype*. Иными словами, выходной переменной определим *PETALWID*, а входными - *SEPALLEN*, *SEPALWID* и *PETALLEN* и *Iristype*. В качестве исходных данных использовать данные, изложенные в постановке задачи.

ПЗ 5 Прогнозирование стоимости дома в пакете Deductor.

Постановка задачи.

Таблица RE.dat из стандартного набора данных BrainMaker содержит 217 примеров продажи домов. Каждый продаваемый дом характеризуется следующими параметрами:

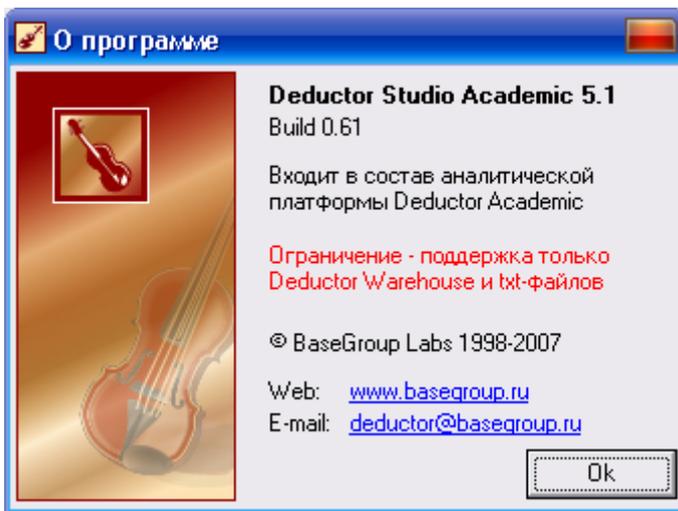
№п/п	Имя	Описание	Ранг
1	SALEPRIC	Действующая цена дома при продаже	103тыс – 250тыс \$
2	DWLUN	Количество построек дома	1 - 3
3	RDOS	Продолжительность продажи (мес.)	0 - 23
4	YRBLT	Год постройки	1859 - 1986
5	TOTFIXT	Количество вертикальных	5 - 17
6	HEATING	Код системы отопления	2 или 3
7	WBFSTKS	Количество брандмауэров	0 - 1
8	BMNTGAR	Количество подвалов/гаражей	0 - 2
9	ATTFRGAR	Полезная площадь гаража	0 - 228
10	TOTLIVAR	Общая жилая площадь	714 - 4185
11	DECK/OFP	Площадь настила или открытой веранды	0 - 738
12	ENCLPOR	Площадь закрытой веранды	0 - 452
13	NBHDGRP	Количество соседей (код)	1 или 2
14	RECROOM	Площадь гостиных (recreation room)	0 - 672
15	FINBSMT	Площадь основных комнат	0 - 810
16	GRADE%	Классность постройки	0,85 – 1,08
17	CDU	Степень удобства	3 - 5
18	TOTOBY	Комплекс других значений (постройки и двора)	0 - 16400

Вся информация о состоявшихся продажах собрана в таблицу. Каждая колонка этой таблицы представляет тип данных, а каждая строка – реализацию (состоявшуюся продажу).

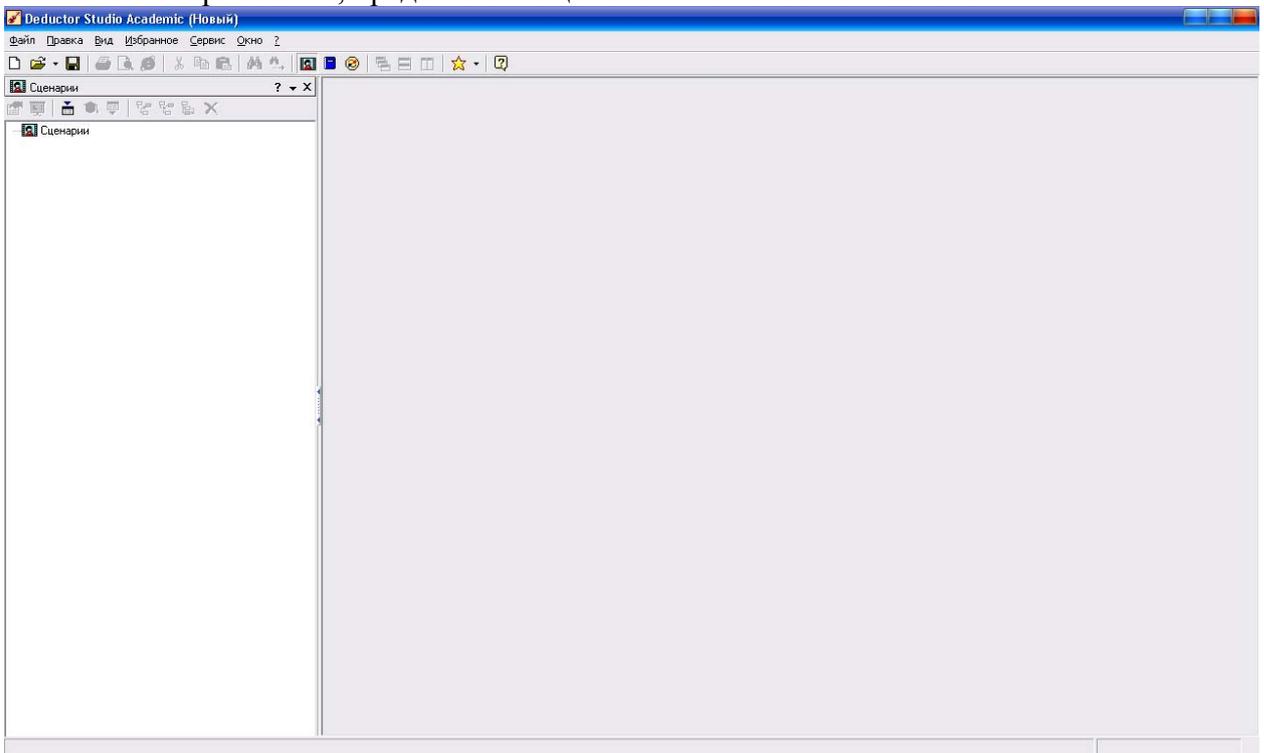
Требуется: по приведенным в таблице RE.dat данным обучить сеть прогнозирования стоимости дома, и определить, по какой цене могут быть проданы ещё 2 дома.

Подготовка текстового файла для нейросетевого исследования.

Запускаем пакет:



Появляется первое окно, предоставляющее очень мало возможностей:

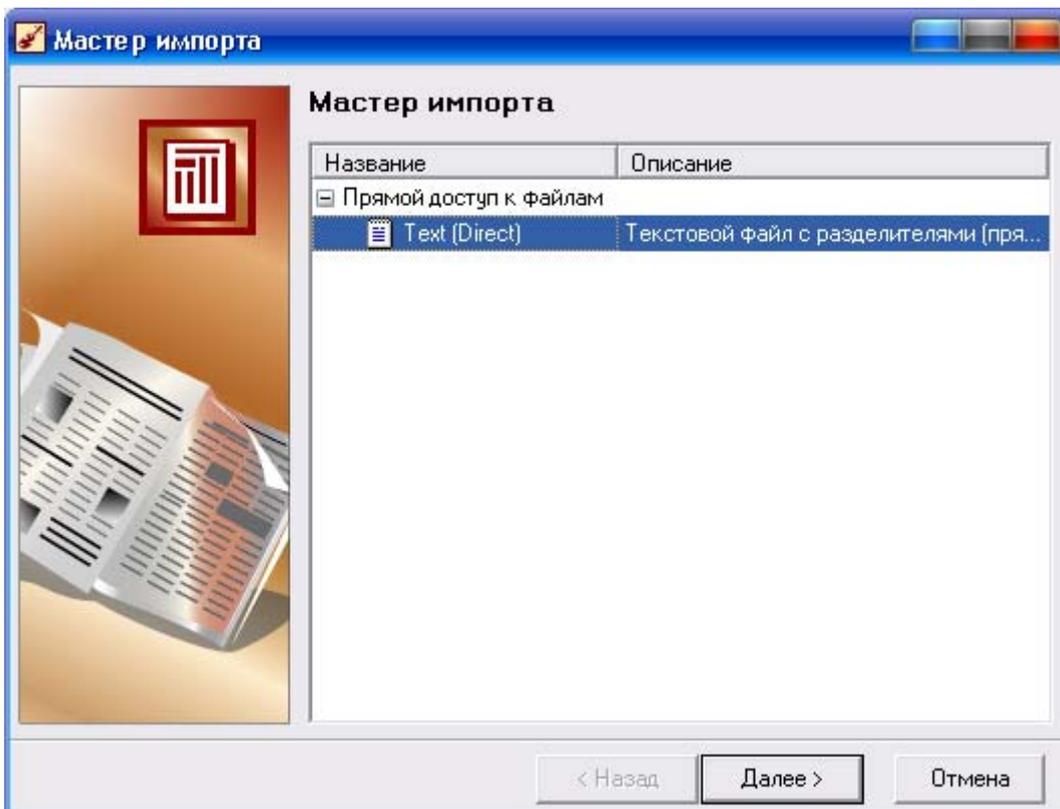


После загрузки пакета доступен только один элемент управления - Мастер импорта:

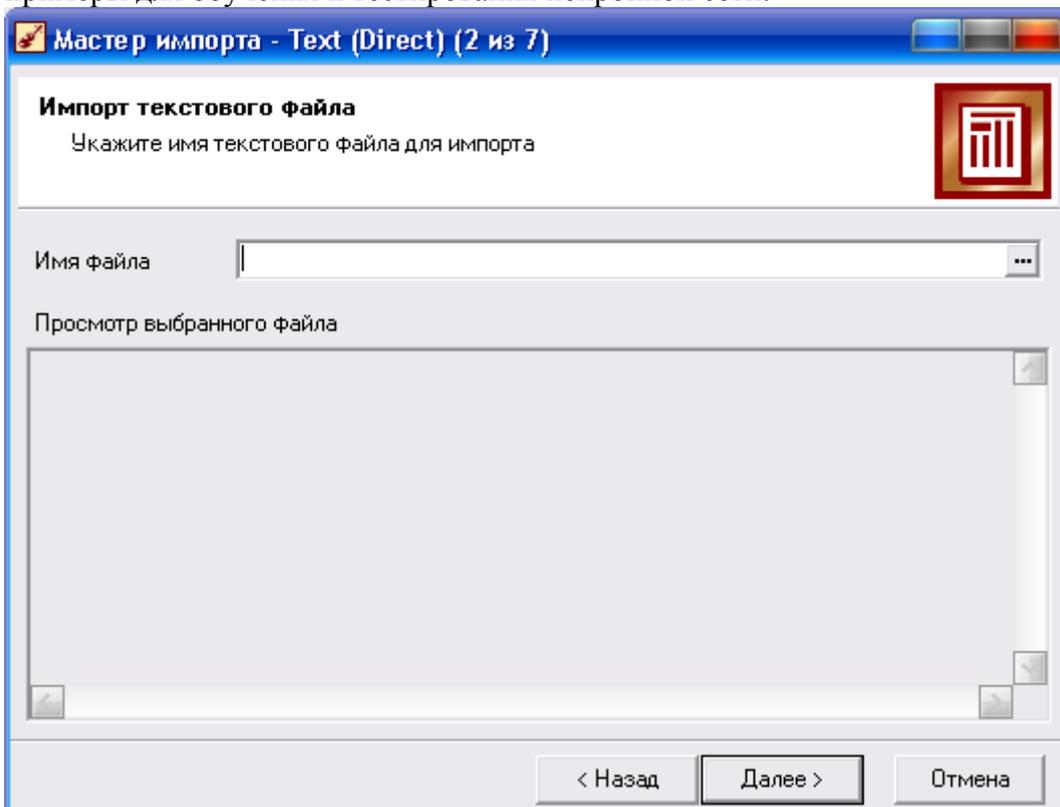


Загрузка в Deductor текстового файла.

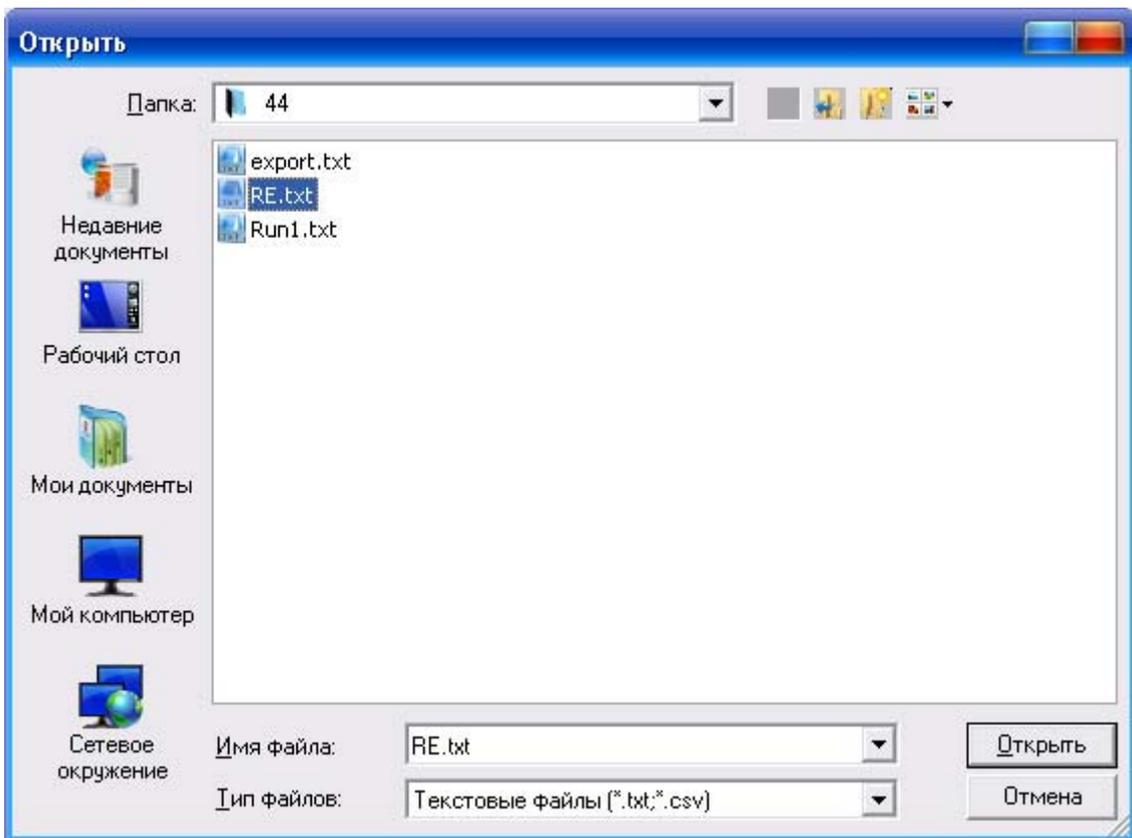
С помощью мастера импорта можно только загрузить в Deductor текстовый файл. Поскольку файл Re.dat пакета BrainMaker является текстовым, перепишем его в свой каталог и изменим расширение файла на txt.



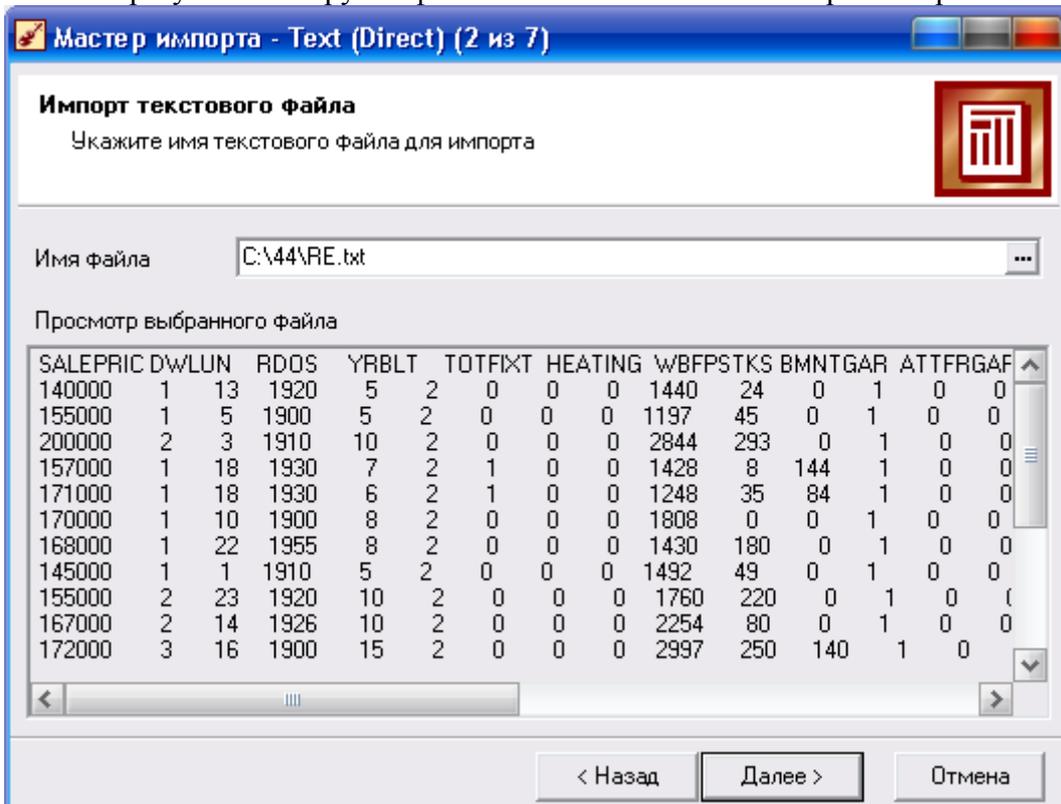
Следующее окно Мастера импорта позволяет начать поиск файла, содержащего примеры для обучения и тестирования нейронной сети:



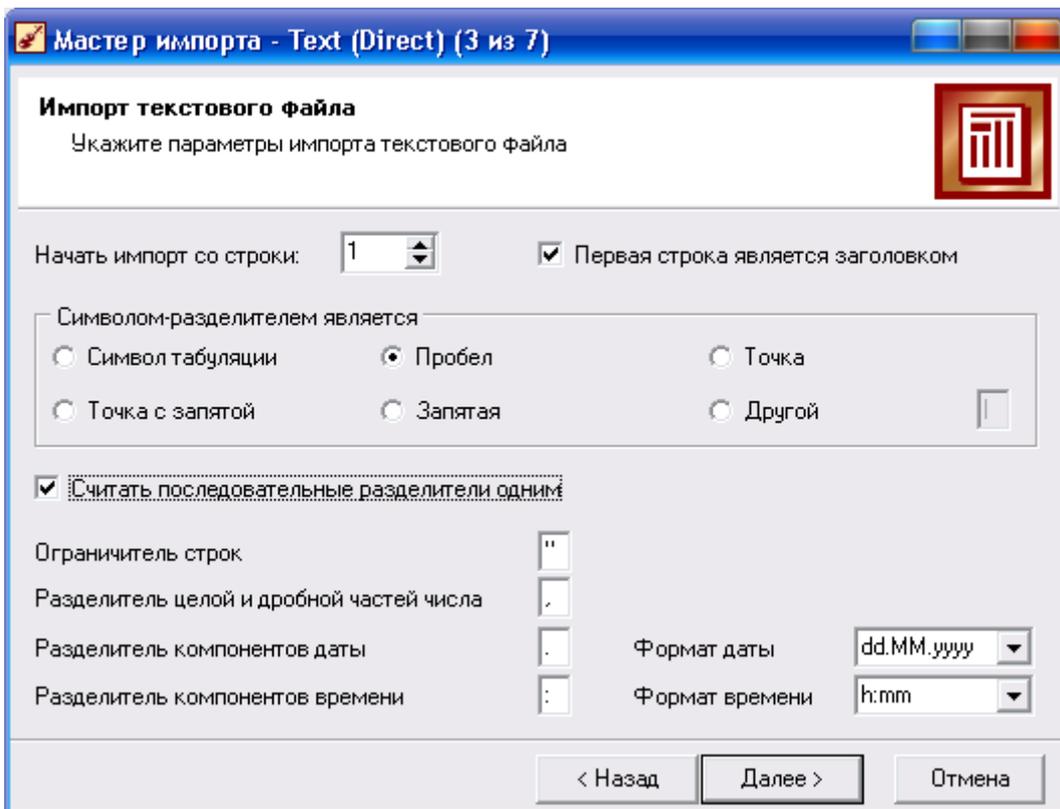
Поиск файла для обучения сети производится стандартными средствами ОС:



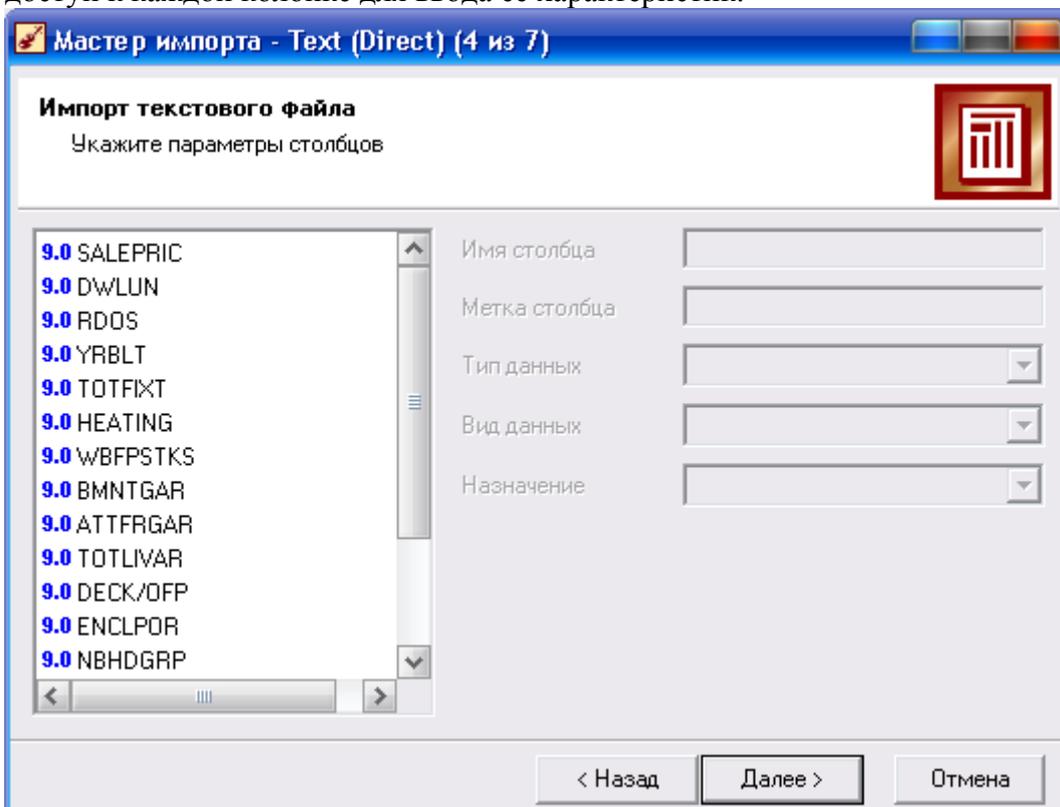
В результате загрузки файл появляется в окне Мастера импорта:



Далее определяются особенности загруженного файла:

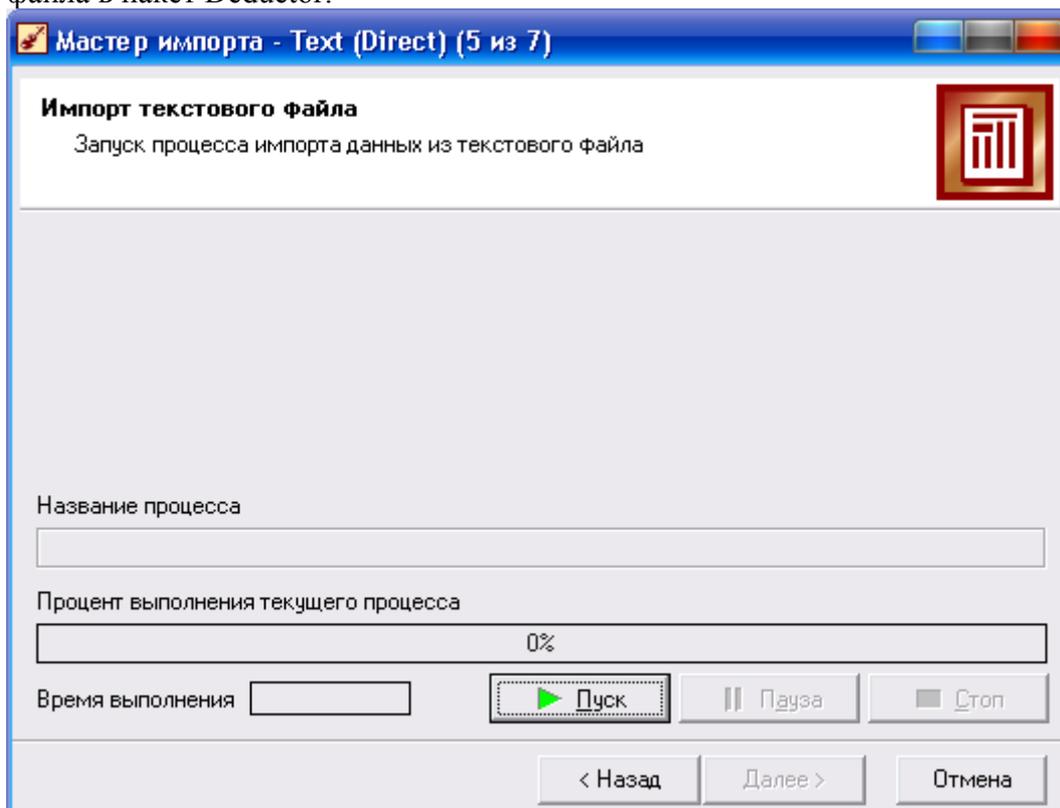


При правильной характеристике файла Мастер импорта выводит состав колонок загружаемого файла. При этом имена колонок выводятся таким образом, чтобы облегчить доступ к каждой колонке для ввода её характеристик:

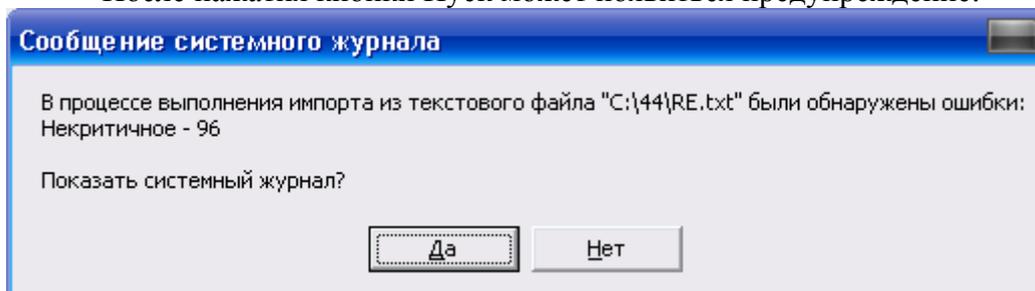


Для ввода характеристик какой-нибудь колонки надо активизировать её имя, щёлкнув по нему.

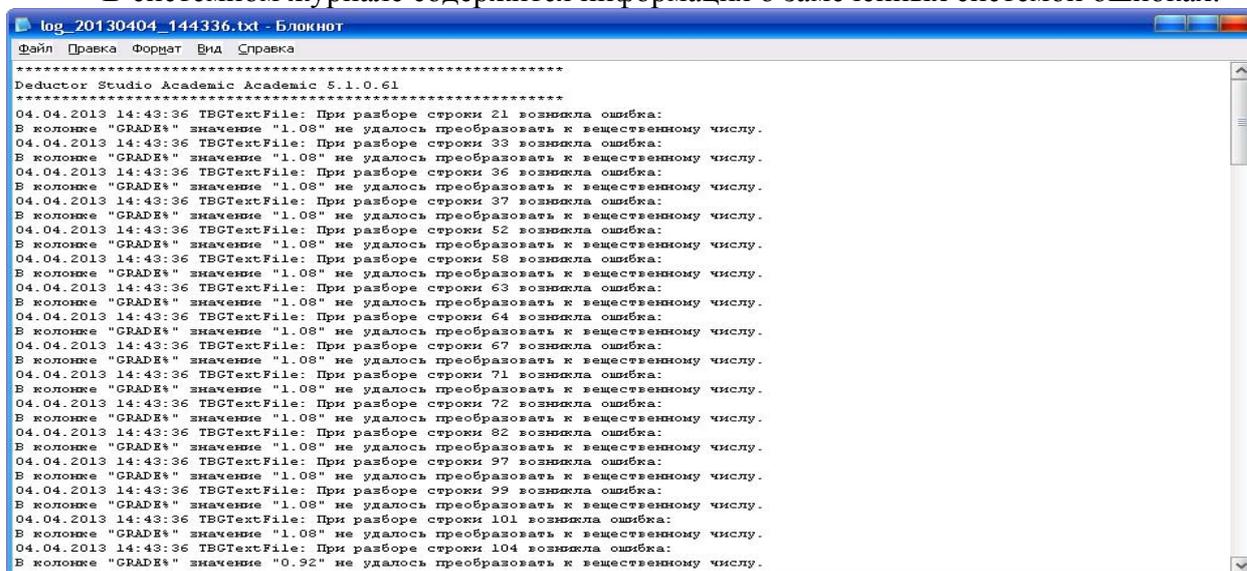
На данном этапе характеризовать колонки не надо. Можно ограничиться импортом файла в пакет Deductor:



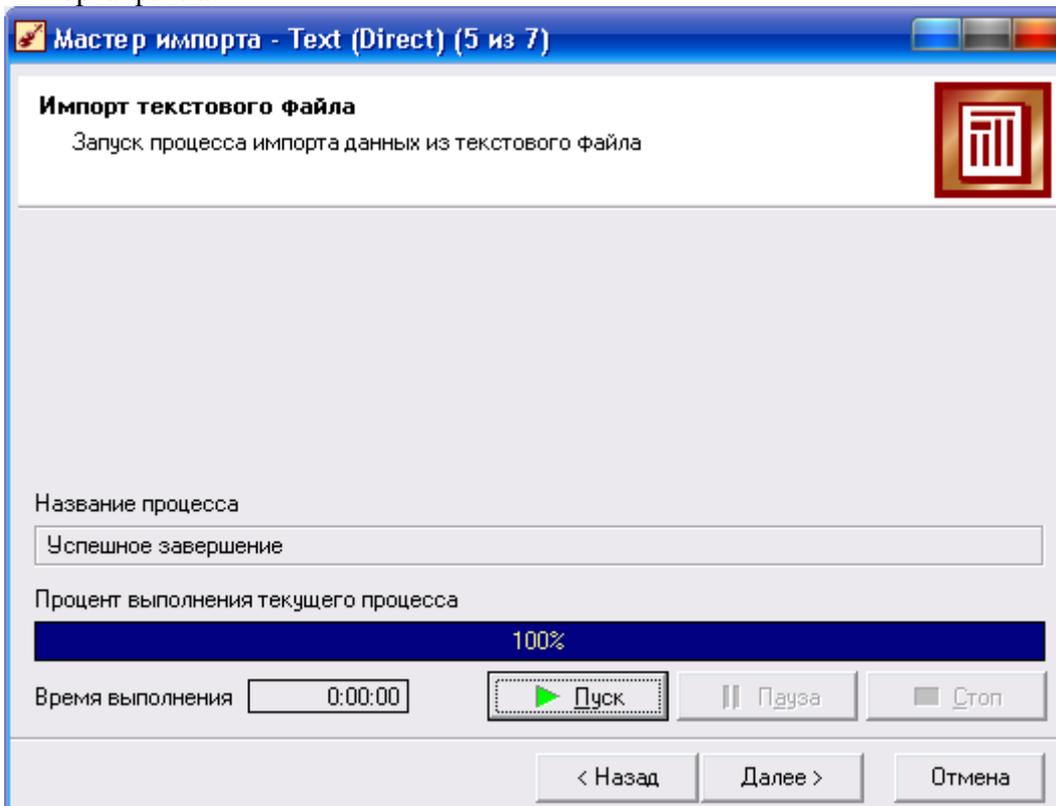
После нажатия кнопки Пуск может появиться предупреждение:



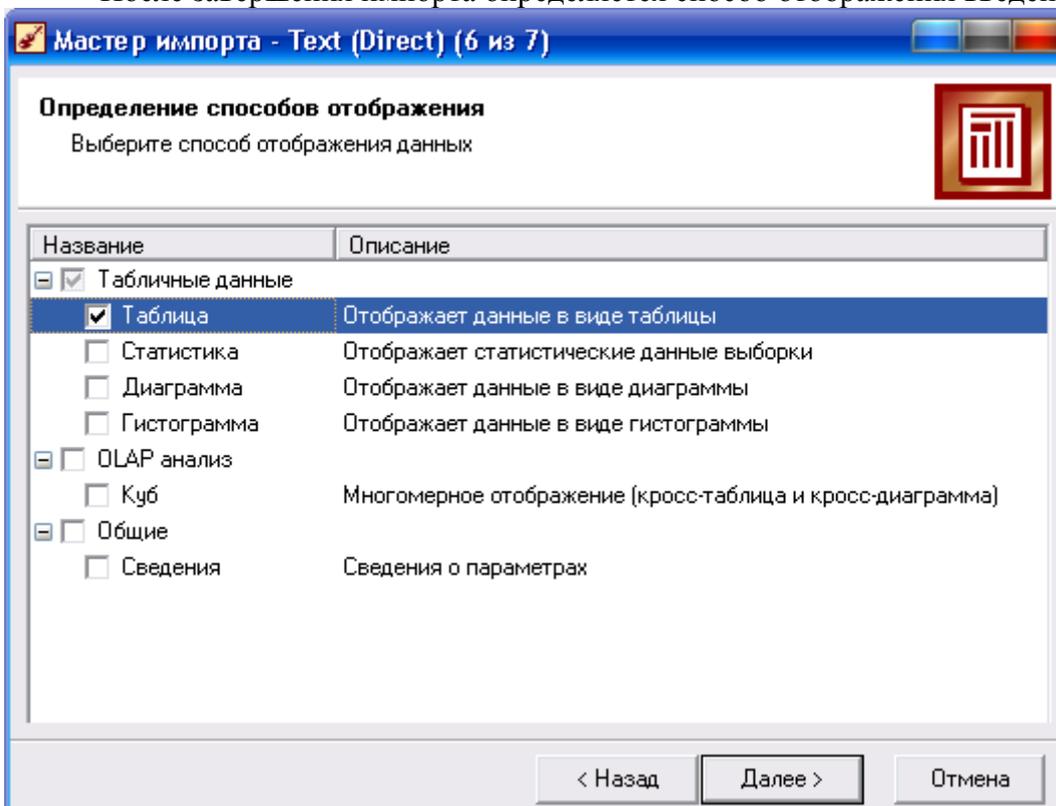
В системном журнале содержится информация о замеченных системой ошибках:



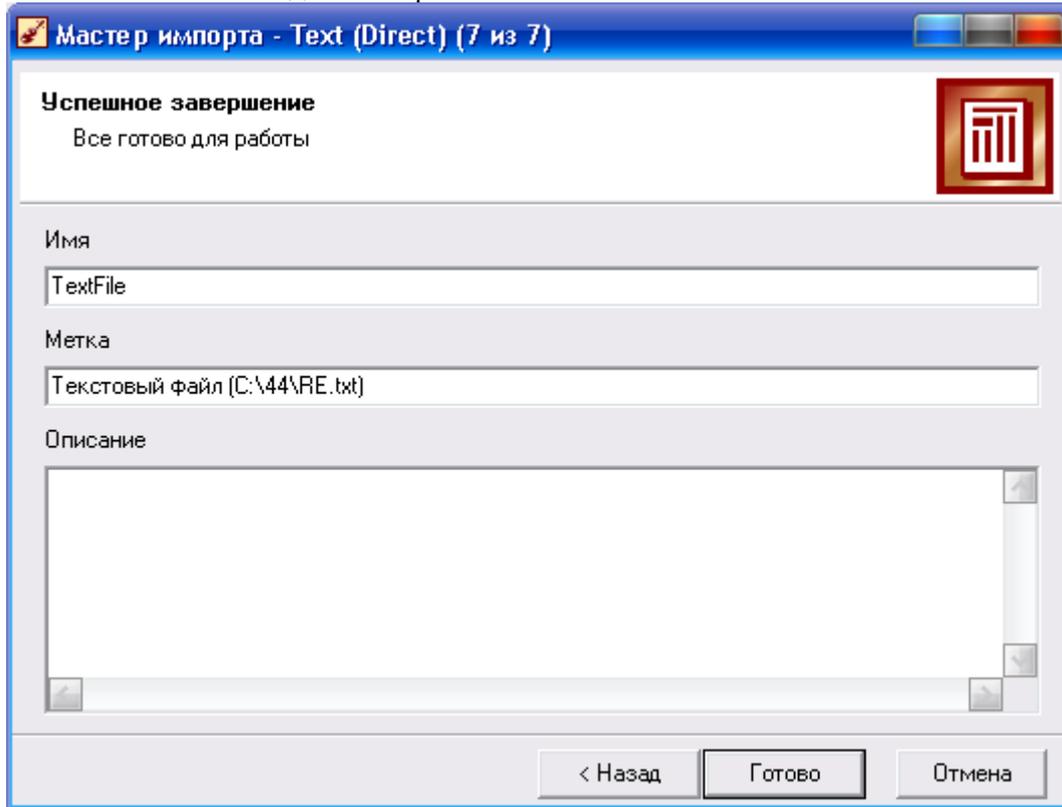
Некритичные ошибки устранять необязательно. Для устранения серьёзных ошибок необходимо вернуться назад и переопределить в исходном файле характеристики отмеченных колонок, или отключить их от участия в анализе. При успешном завершении импорта файла:



После завершения импорта определяется способ отображения введённых данных:



При успешном завершении загрузки данных система сообщает имя и местоположение введённого файла:

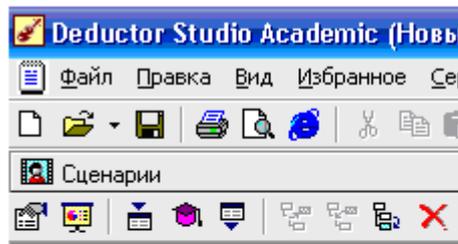


На экран выводятся заявленные в п. 6 из 7 введённые данные:

SALEPRIC	DWLUN	RDOS	YRBLT	TOTFKT	HEATING	WBFPSKS	BMNTGAR	ATFFRGAR	TOTLVAR	DECK/OFF	EN
140000	1	13	1920	5	2	0	0	0	1440	24	
155000	1	5	1900	5	2	0	0	0	1197	45	
200000	2	3	1910	10	2	0	0	0	2844	293	
157000	1	18	1930	7	2	1	0	0	1428	8	
171000	1	18	1930	6	2	1	0	0	1248	35	
170000	1	10	1900	8	2	0	0	0	1808	0	
168000	1	22	1955	8	2	0	0	0	1430	180	
145000	1	1	1910	5	2	0	0	0	1432	49	
155000	2	23	1920	10	2	0	0	0	1760	220	
167000	2	14	1926	10	2	0	0	0	2254	60	
172000	3	16	1900	15	2	0	0	0	2997	250	
172000	2	22	1920	9	2	0	0	0	2430	90	
155000	1	13	1920	5	2	0	0	0	1435	0	
223000	3	11	1900	15	2	0	0	0	3654	390	
240000	3	0	1900	15	2	0	0	0	3942	225	
170000	2	16	1910	10	2	0	0	0	2012	316	
139900	1	6	1910	5	2	0	0	0	1563	16	
192000	2	12	1900	10	2	0	0	0	2361	353	
150000	1	16	1950	5	2	0	0	0	1248	16	
183500	2	10	1900	10	2	0	0	0	2208	138	
153000	1	1	1955	5	2	0	0	0	1204	35	
160000	1	14	1925	5	2	0	0	0	1278	112	
146000	1	19	1915	7	2	0	0	0	1176	174	
178000	2	19	1920	10	2	0	2	0	2105	254	
155000	1	8	1920	10	2	0	0	0	1274	0	
160000	1	23	1930	6	2	0	1	56	1372	312	
154000	1	12	1910	8	2	0	0	0	1358	95	
148000	1	7	1925	5	2	0	1	0	1529	0	
155000	1	8	1925	5	2	0	0	0	1364	182	
175000	2	19	1940	8	2	1	0	0	1494	60	
150000	1	3	1920	5	2	0	0	0	1396	0	
145000	1	12	1900	5	2	0	0	0	1336	0	
161000	1	15	1900	5	2	0	0	0	1692	54	
230000	3	18	1900	15	2	0	0	0	3339	234	
179000	2	22	1900	10	2	0	0	0	1973	60	
193000	2	17	1850	10	2	0	0	0	2250	72	

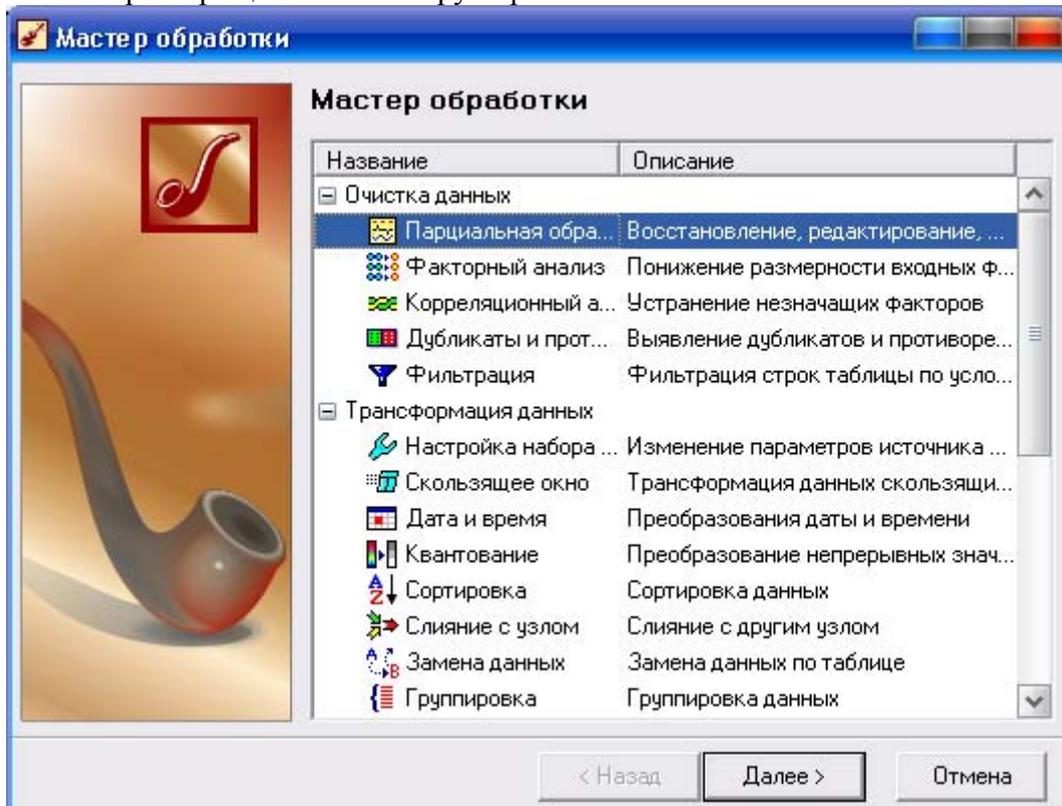
Нейросетевая обработка загруженного файла.

После загрузки в систему Deductor исходных данных картина меняется:

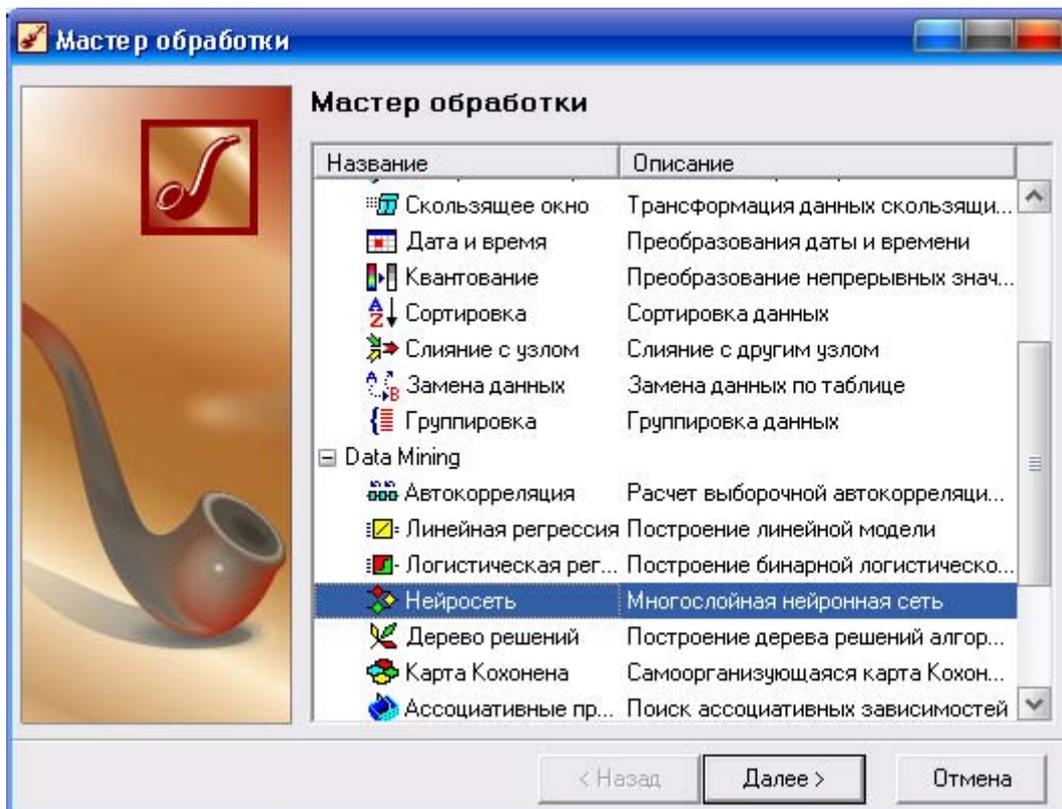


↑ Мастер экспорта
 ↑ Мастер обработки
 ↑ Мастер импорта
 ↑ Мастер визуализации
 ↑ Настроить узел

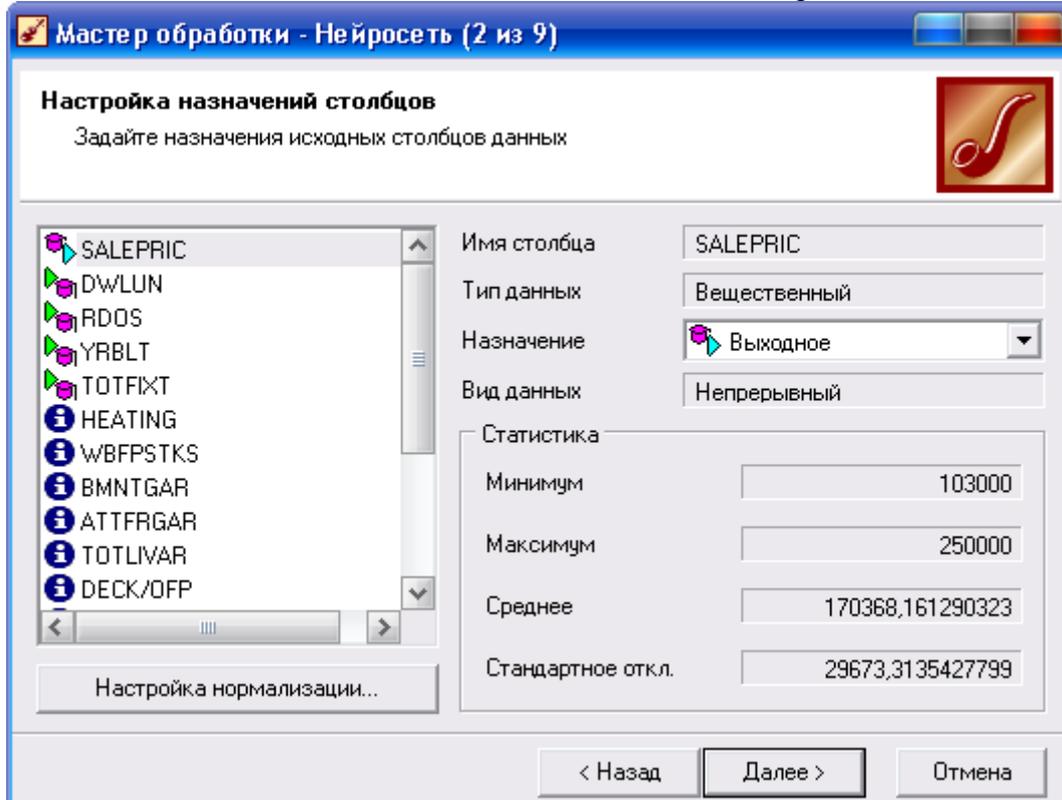
При обращении к Мастеру обработки появляется окно:



В этом окне необходимо найти и активировать пункт "Нейросеть":

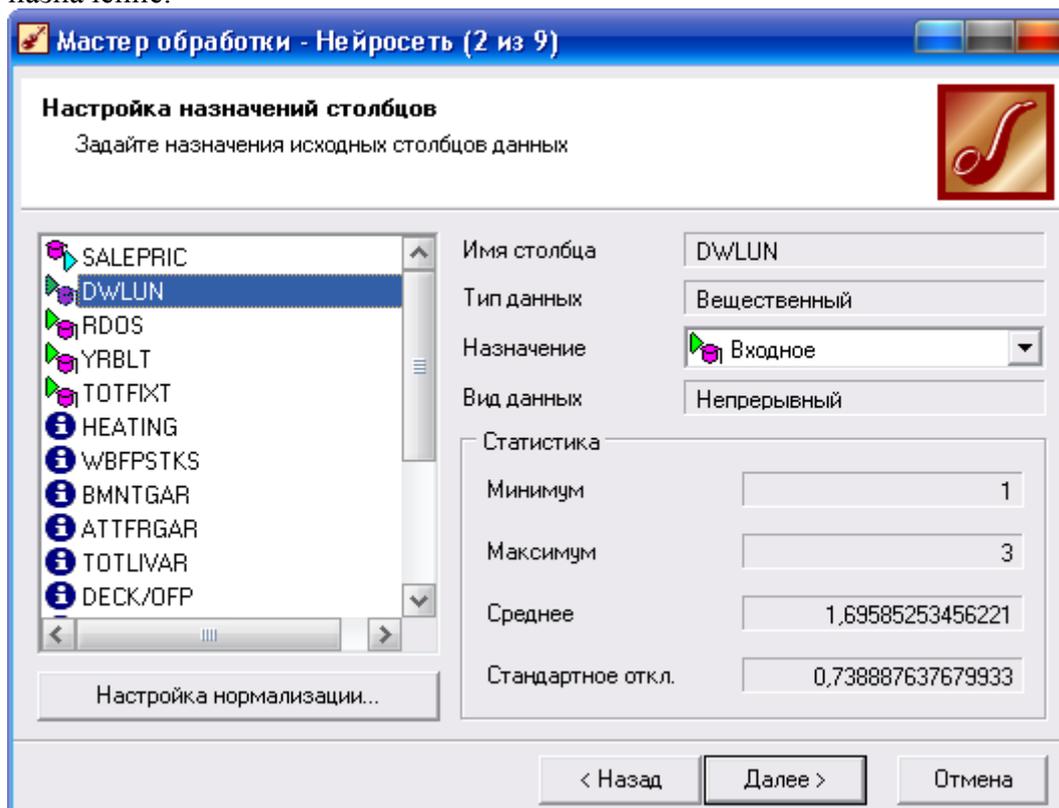


Открывается 2 окно Мастера обработки. Это окно позволяет настроить систему, определив параметры каждой колонки таблицы с исходными данными. Определить надо назначение колонки. Остальные данные должны были определяться в Мастере импорта.



В данной задаче используется 17 входных колонок и 1 выходная. Для настройки назначения каждой колонки необходимо выделить её имя и определить 1 параметр:

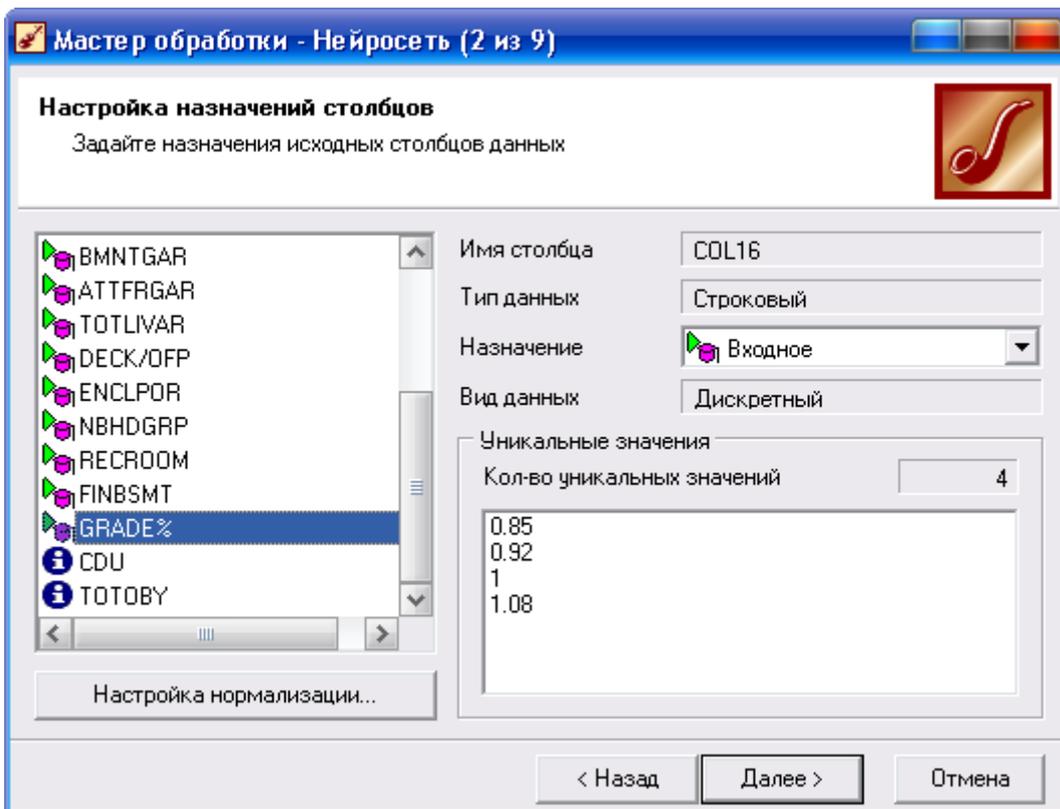
данная величина имеет -> неиспользуемое, входное, выходное или информационное назначение:



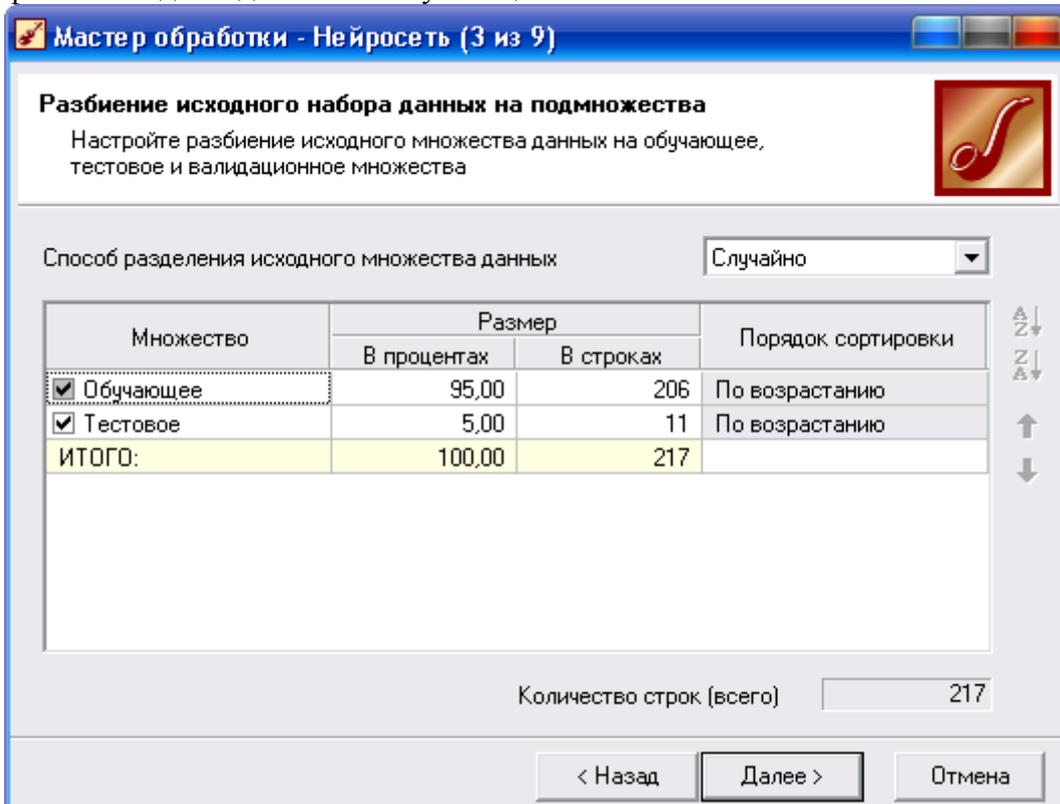
Для остальных переменных установить значения в соответствии с таблицей:

№	Имя переменной	Информационное назначение
1	SALEPRIC	Выходное
2	DWLUN	Входное
3	RDOS	Входное
4	YRBLT	Входное
5	TOTFIXT	Входное
6	HEATING	Входное
7	WBFSTKS	Входное
8	BMNTGAR	Входное
9	ATTFRGAR	Входное
10	TOTLIVAR	Входное
11	DECK/OFP	Входное
12	ENCLPOR	Входное
13	NBHDGRP	Входное
14	RECROOM	Входное
15	FINBSMT	Входное
16	GRADE%	Входное
17	CDU	Входное
18	TOTOBY	Входное

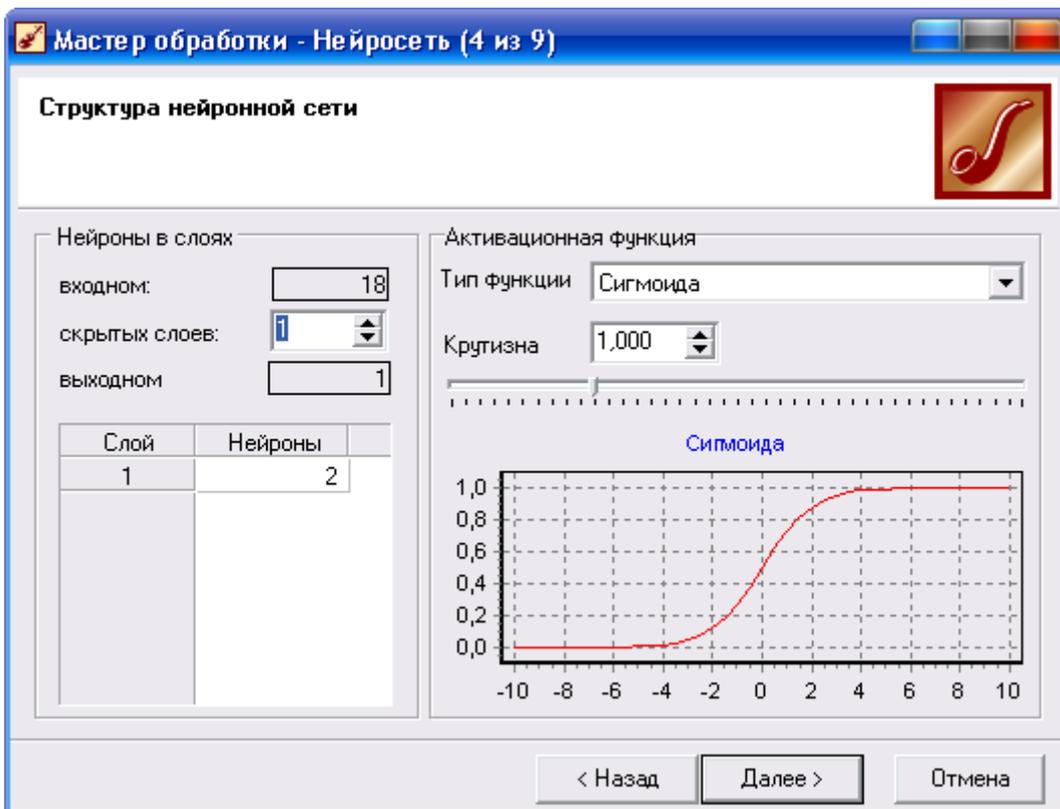
Только у колонки Grade с типом данных “Строковый” внешний вид окна изменён:



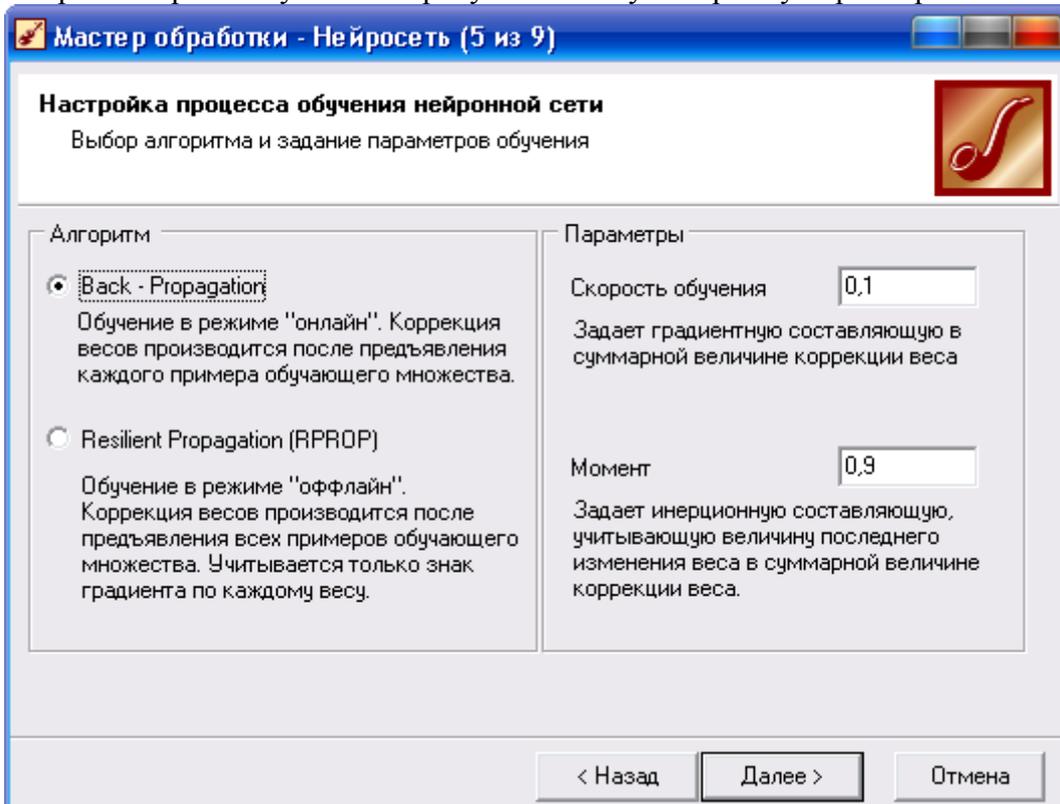
При переходе к 3 окну Мастера обработки производится разделение введённого файла исходных данных на обучающее и тестовое множества:



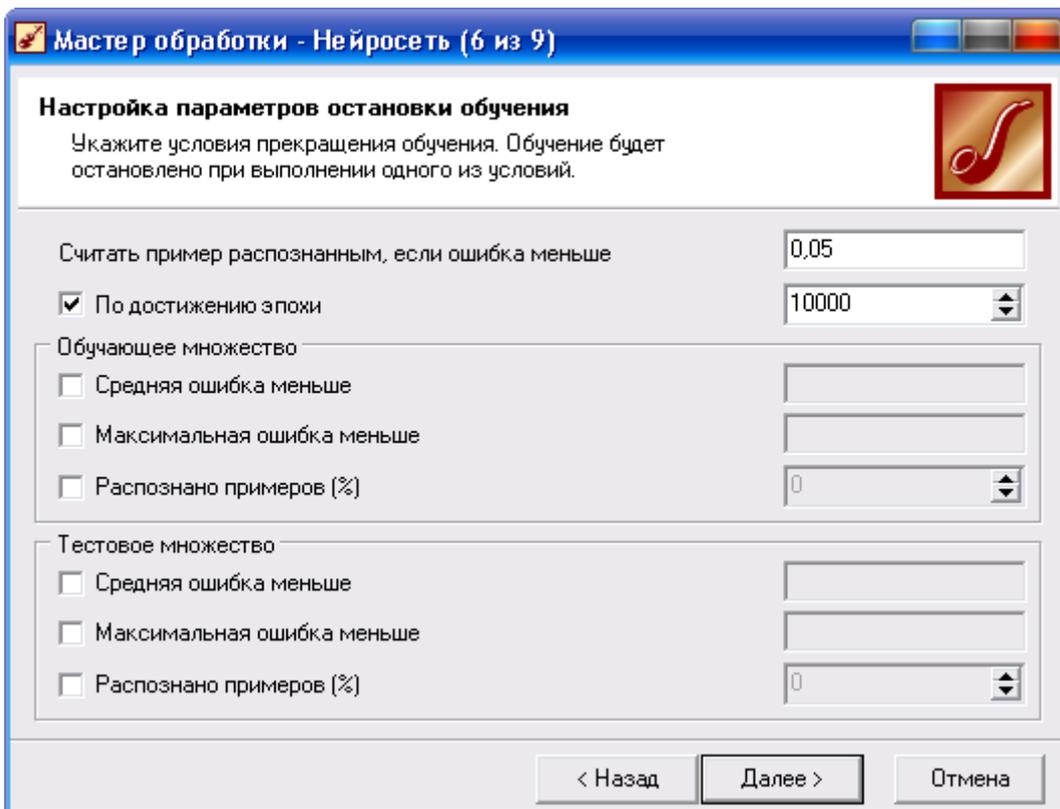
В 4 окне определяются количество слоёв, количество нейронов в каждом слое, и активационная функция нейронов (сигмоида, гипертангенс или арктангенс) в каждом слое:



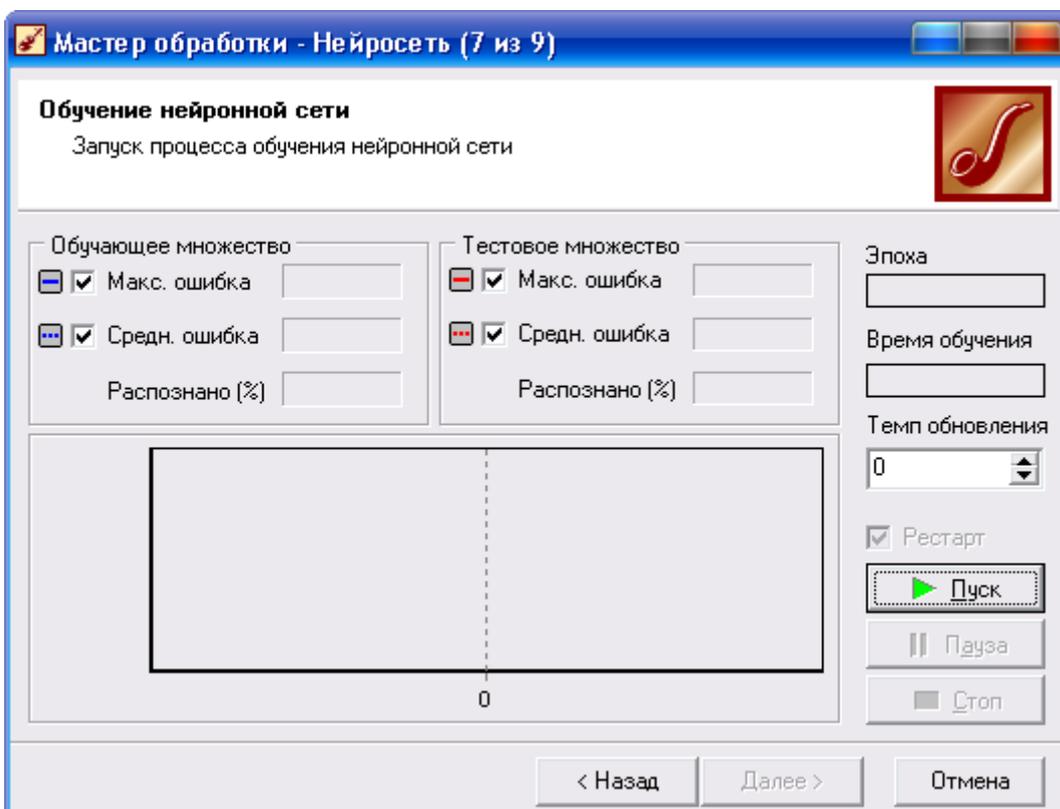
5 кадр Мастера обработки позволяет настроить процесс обучения нейронной сети, выбрав алгоритм обучения и требуемые этому алгоритму параметры:



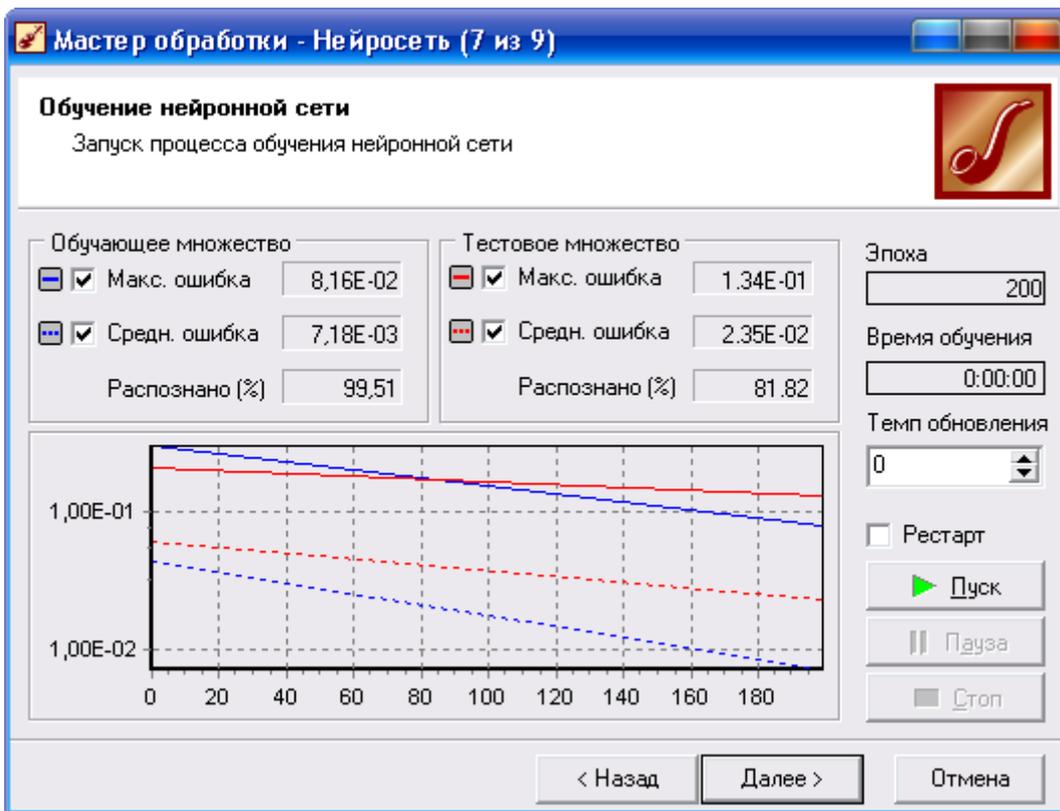
6 кадр позволяет настроить параметры остановки обучения. Основными критериями являются величина ошибки сети (на обучающем или тестовом множестве), % правильно распознанных примеров и количество эпох:



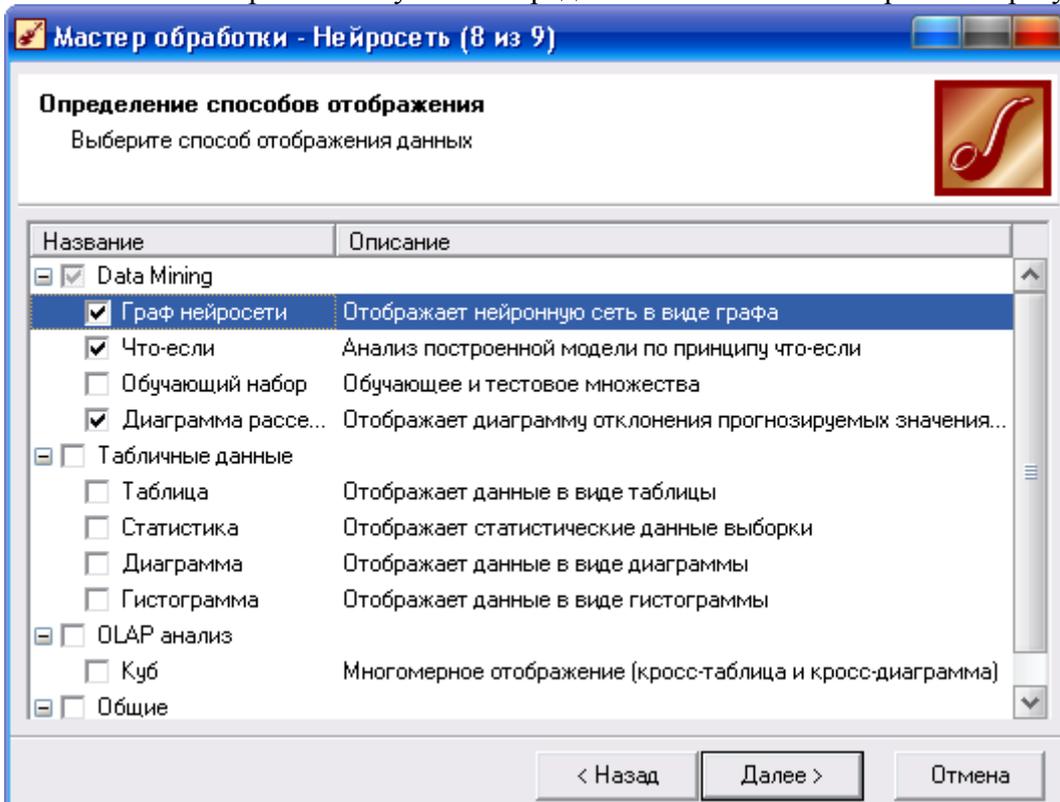
При запуске процесса обучения система Deductor определяет и отображает на экране результаты своей работы:



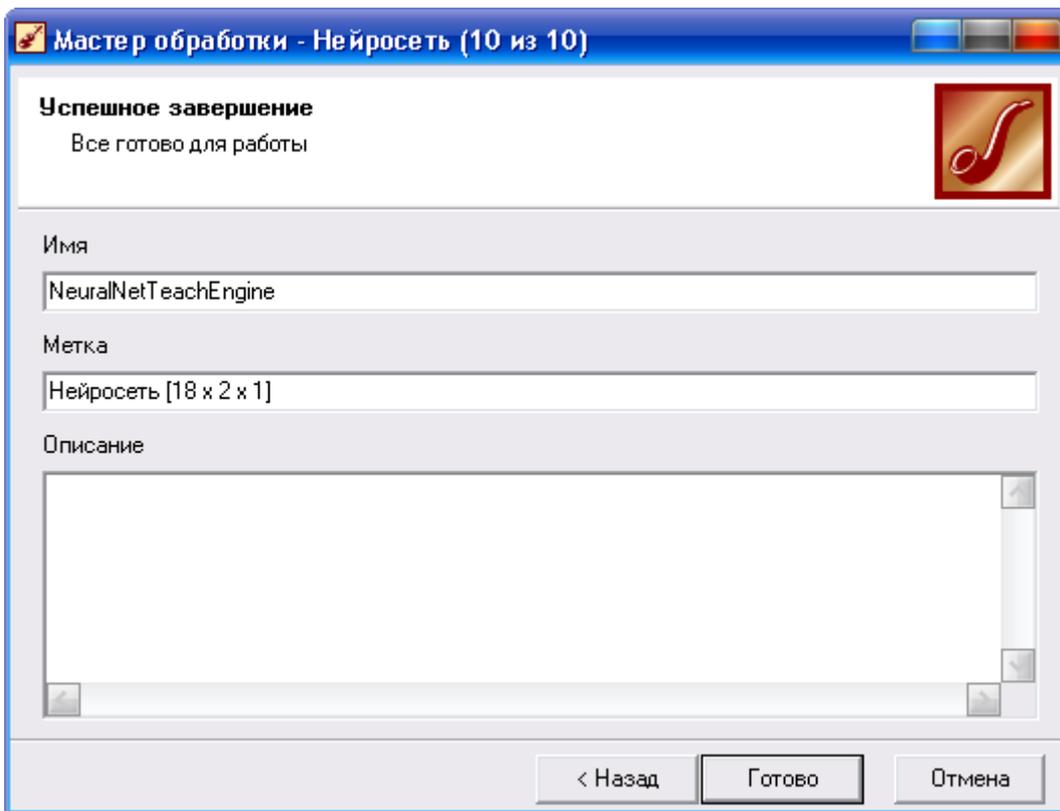
После нажатия кнопки Пуск:



После завершения обучения определяются способы отображения результатов

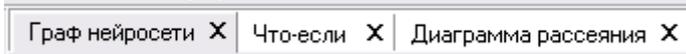


На 10 шаге Мастер обработки сообщает о параметрах созданной сети:

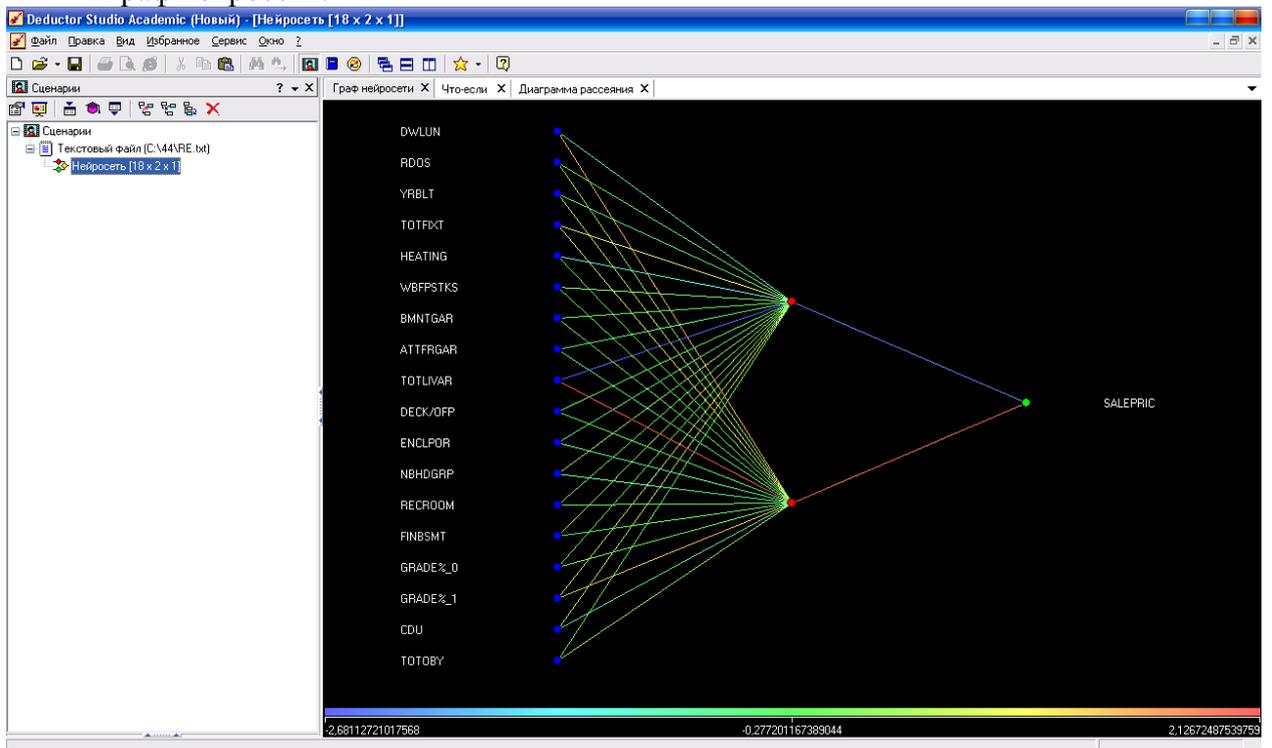


Визуализатор.

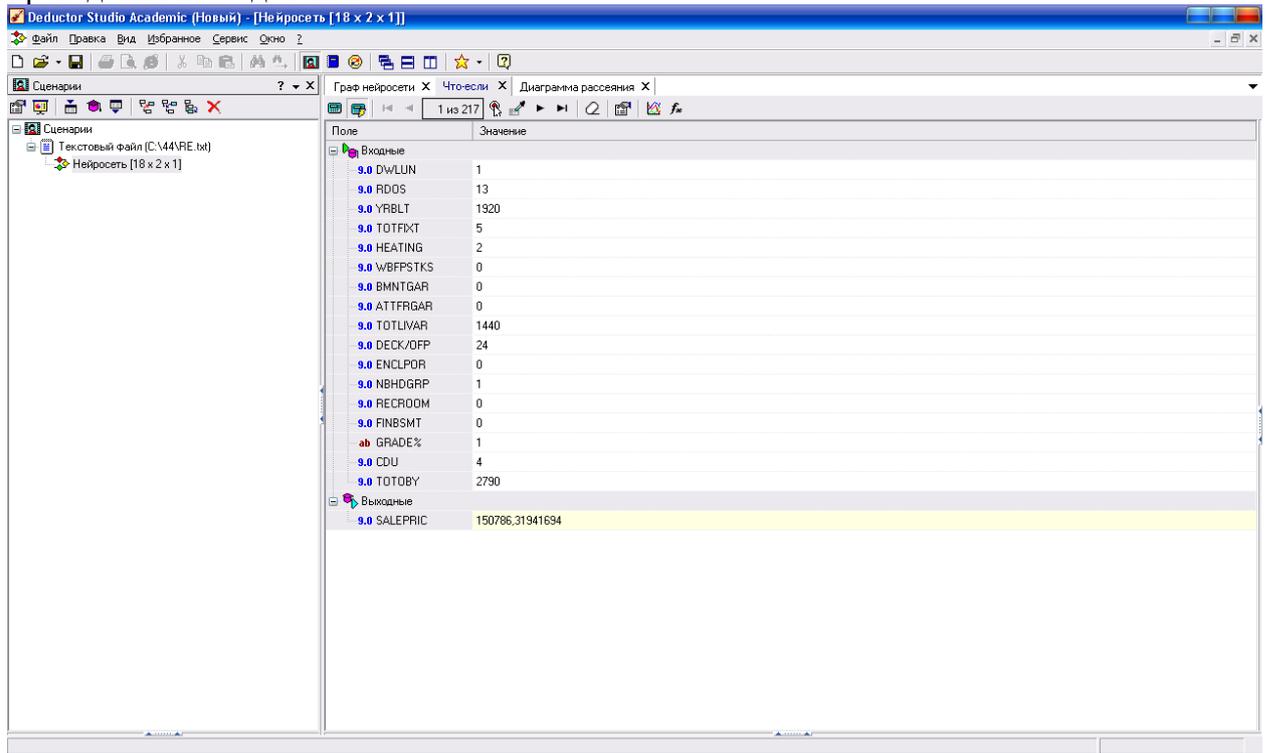
Запрошенные способы отображения результатов доступны для переключения в основном информационном окне пакета:



Граф нейросети:



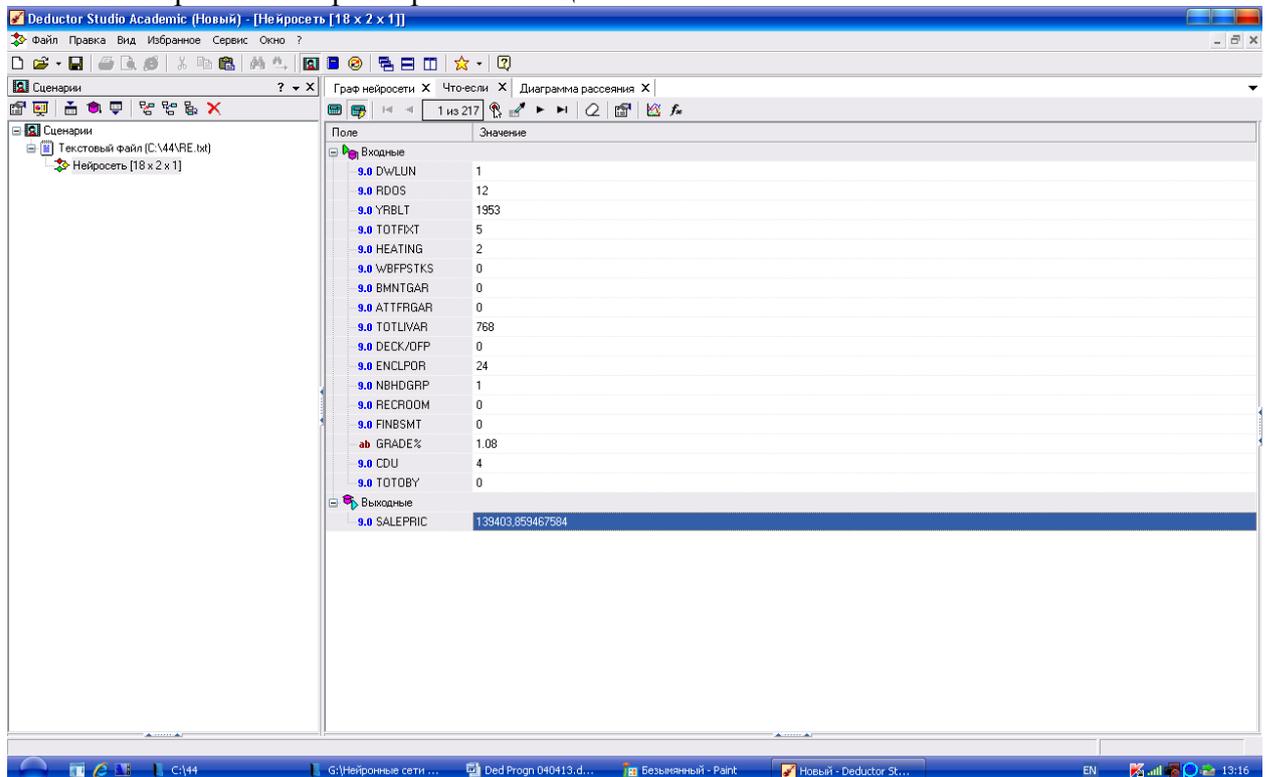
Визуализатор «Что если» позволяет использовать обученную нейросеть для проведения исследования.



В начале - при постановке задачи предполагалось определить стоимость двух домов. Один из них был продан за \$136500 и характеризовался следующими параметрами:

1 12 1953 5 2 0 0 0 768 0 24 1 0 0 1.08 4 0

Набираем эти параметры в таблице «Что если»:



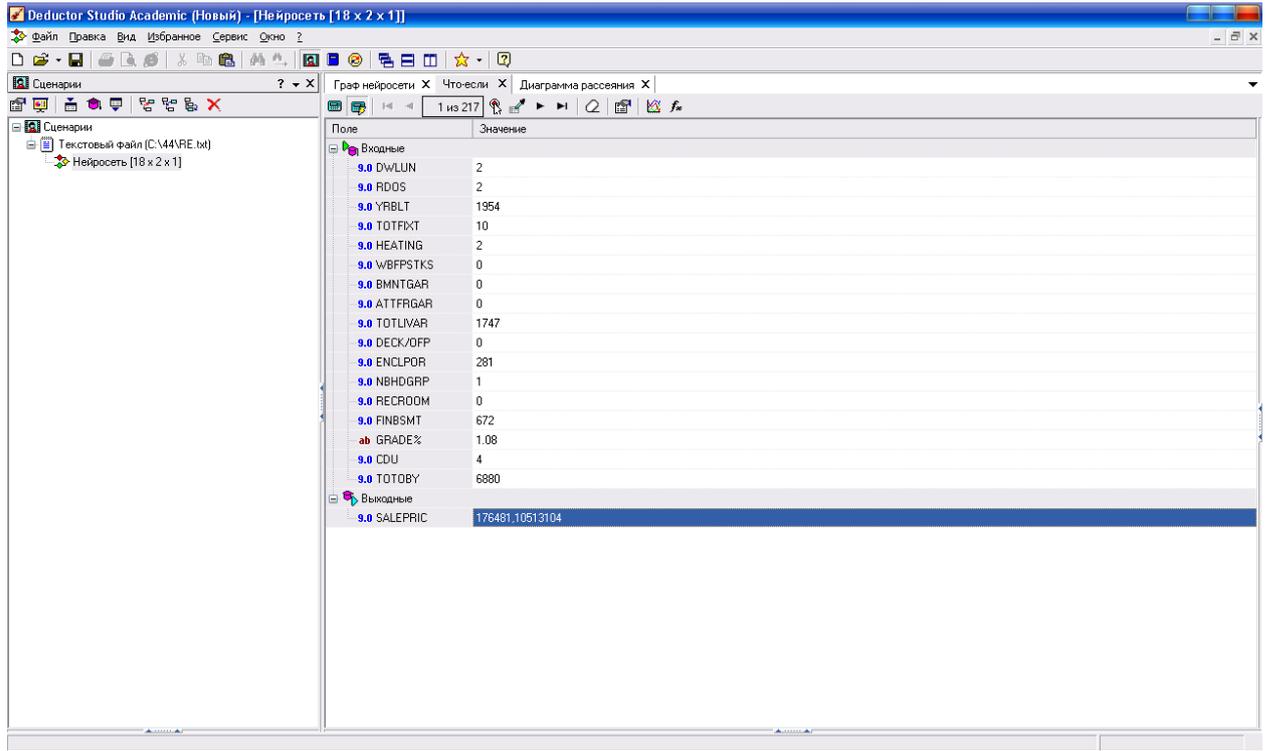
Результат выведен в нижней части таблицы под строкой



т.е. нейросеть по этим данным прогнозировала стоимость дома не 136500, а 139404.

Второй дом был продан за \$170000 и характеризовался:

2 2 1954 10 2 0 0 0 1747 0 281 1 0 672 1.08 4 6880

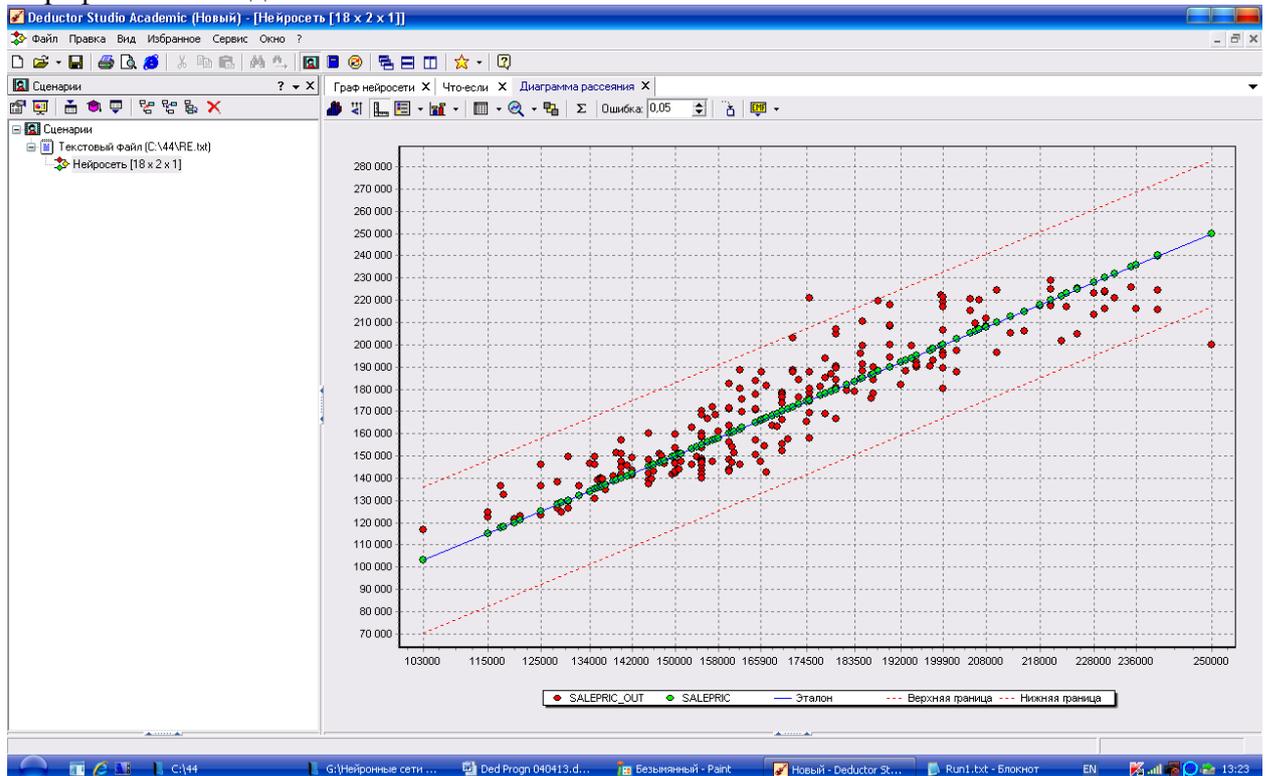


Результат:

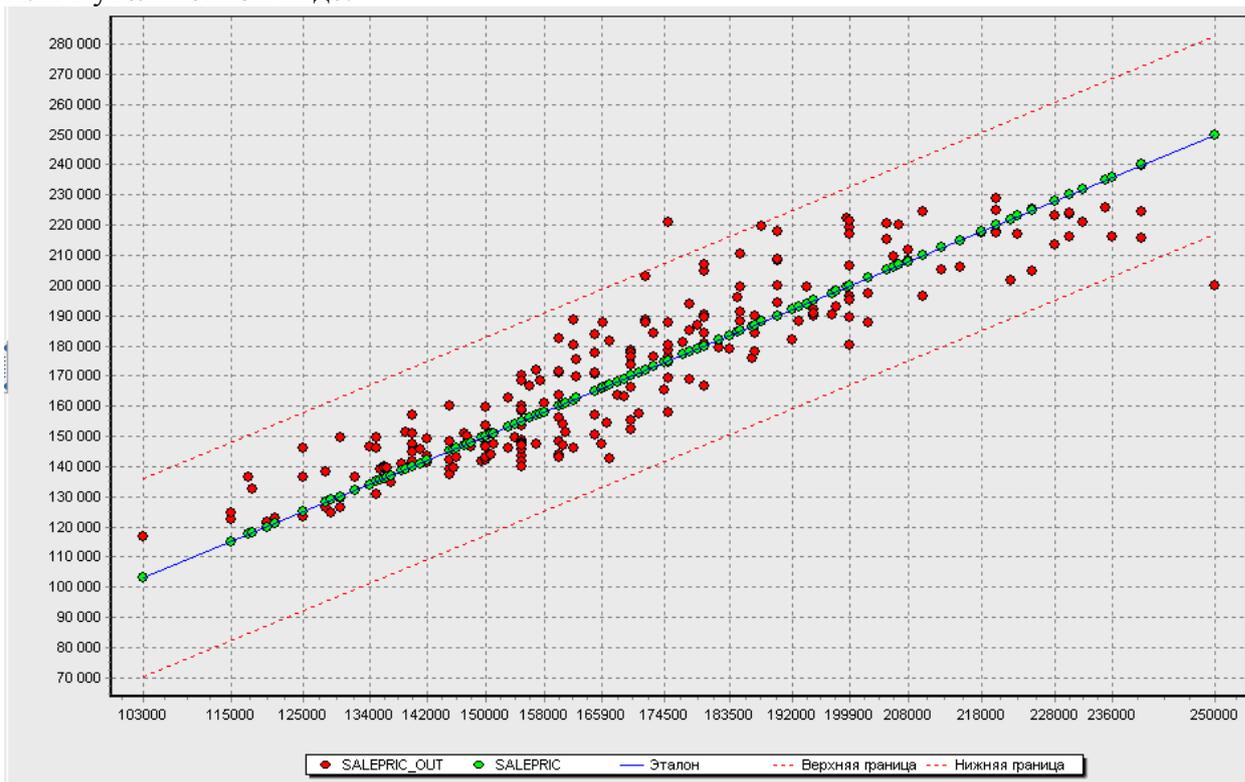


т.е. нейросетевой прогноз составил не 170000, а 176481.

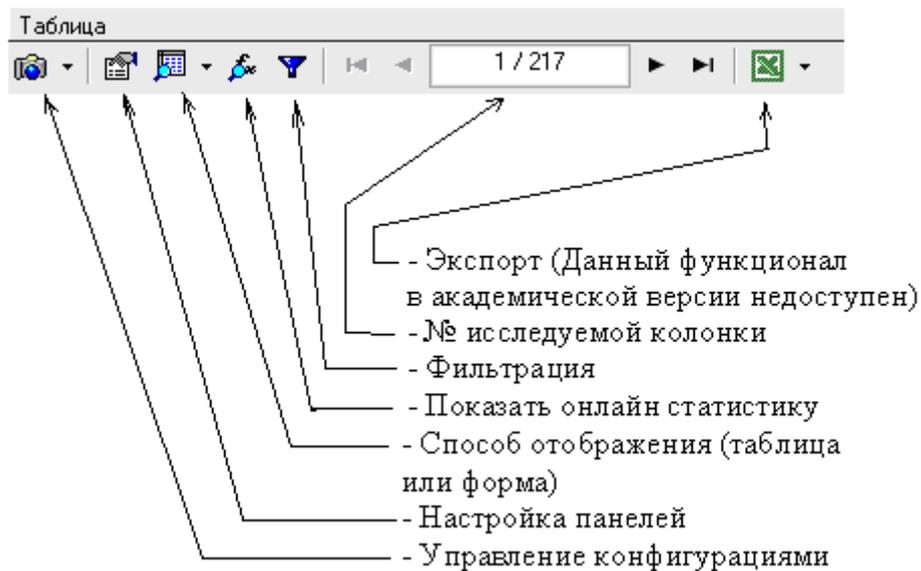
Диаграмма рассеивания показывает результат работы нейросети во время обучения в графическом виде:



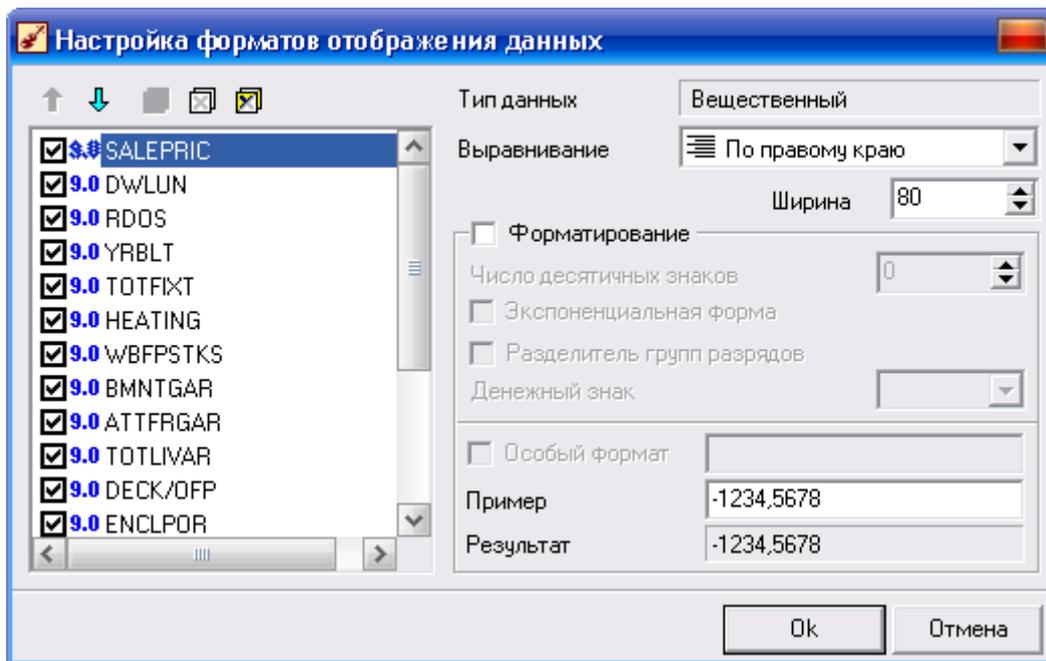
Или в увеличенном виде:



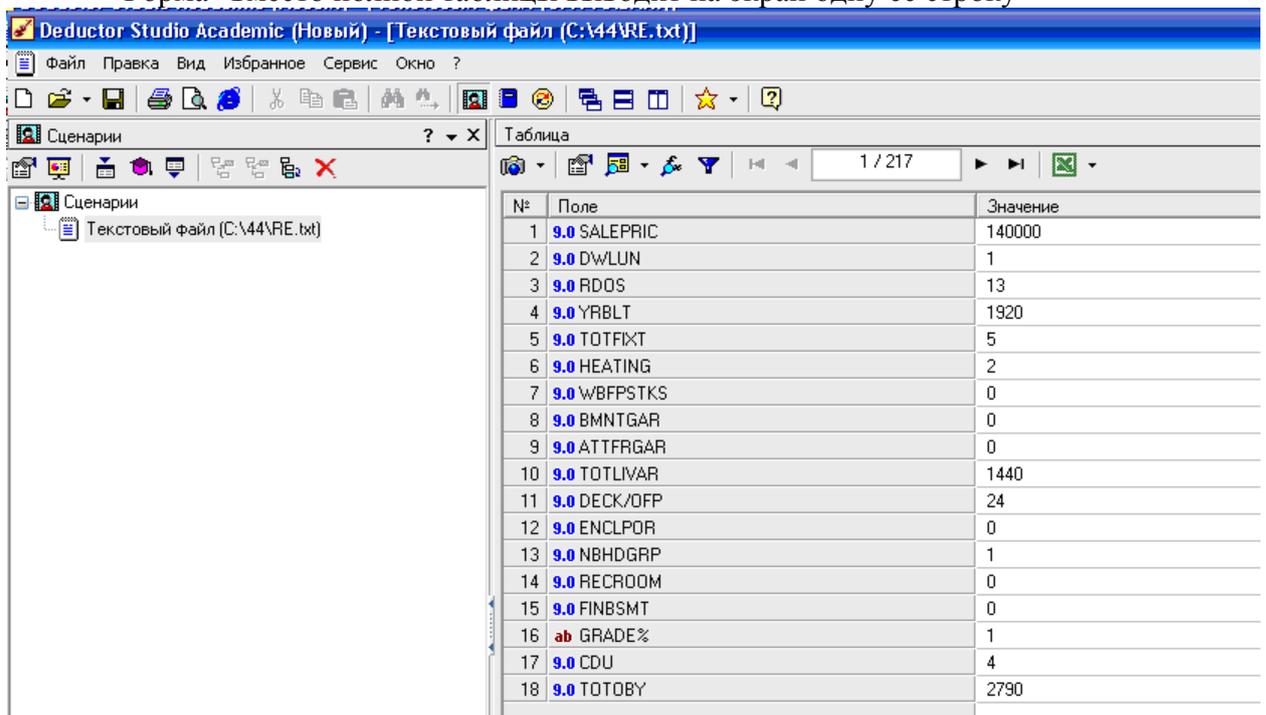
Кроме того, сформированная таблица может быть изучена и преобразована с помощью следующего меню:



Где Настройка панелей вызывает дополнительное окно:



“Форма” вместо полной таблицы выводит на экран одну её строку



“Показать online статистику”:



В общем окне появляется нижняя вставка в виде таблицы:

The screenshot shows the Deductor Studio Academic interface. The main window displays a table with columns: SALEPRIC, DWLUN, RDOS, YRBLT, TOTFIXT, HEATING, WBFPSKTS, BMNTGAR, ATTFRGAR, TOTLVAR, DECK/DFP, EN. The summary statistics window is open, showing the following data:

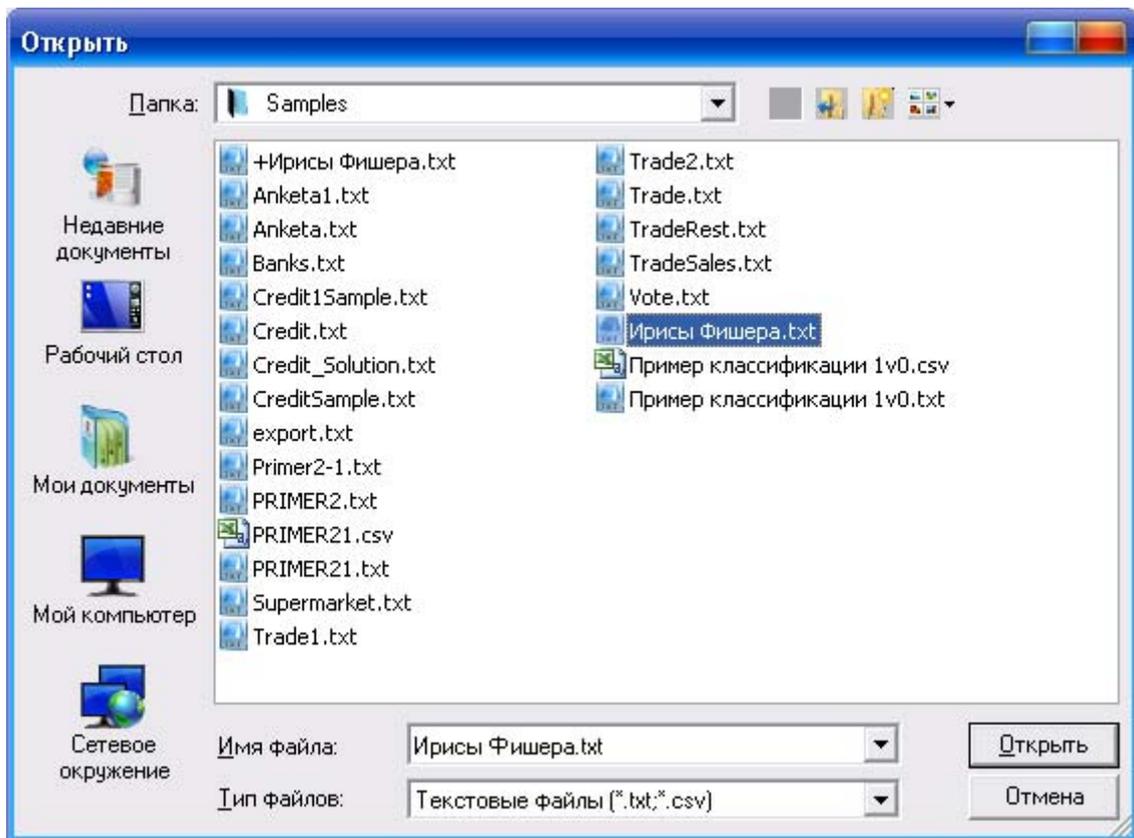
Метка столбца	Мини...	Макс...	Сред...	Стан...	Σ Сумма	Σ ² Сумм...	s Кол-в...	σ Кол-в...
1 9.0 SALEPRIC	103000	250000	161290323	3135427799	36969891	3681548681		0
2 9.0 DWLUN	1	3	3253456221	7637679933	368	742		0
3 9.0 RDOS	0	23	3230414747	3692102125	2655	42653		0
4 9.0 YRBLT	1850	1986	1658986175	7499044137	415168	794410050		0
5 9.0 TOTFIXT	5	17	3953917051	3750025824	1965	20377		0
6 9.0 HEATING	2	3	2488479263	3655843532	437	883		0
7 9.0 WBFPSKTS	0	1	3341013825	4711749663	13	13		0
8 9.0 BMNTGAR	0	2	138248848	1566522609	17	29		0
9 9.0 ATTFRGAR	0	228	5115207373	3170165646	404	69520		0
10 9.0 TOTLVAR	714	4185	3036866359	4310174743	437420	1022544672		0
11 9.0 DECK/DFP	0	738	2073732719	3535065196	33451	10281761		0
12 9.0 ENCLPOR	0	452	3447004608	3317784104	14243	2683521		0

Онлайн-статистика из нижней вставки выводится в виде следующей таблицы:

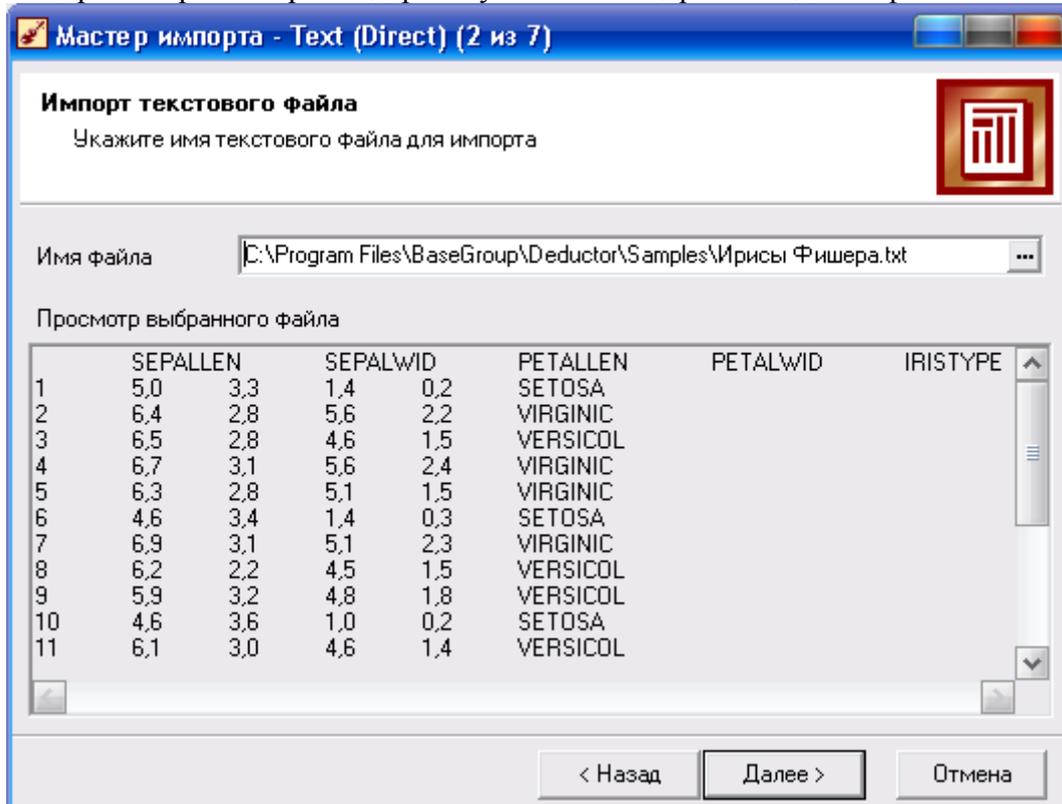
	Метка столбца	Статистика: Кол-во значений = 217							
		Мини...	Макс...	Сред...	Стан...	Σ Сумма	Σ ² Сумм...	s Кол-в...	σ Кол-в...
1	9.0 SALEPRIC	103000	250000	161290323	3135427799	36969891	3681548681		0
2	9.0 DWLUN	1	3	3253456221	7637679933	368	742		0
3	9.0 RDOS	0	23	3230414747	3692102125	2655	42653		0
4	9.0 YRBLT	1850	1986	1658986175	7499044137	415168	794410050		0
5	9.0 TOTFIXT	5	17	3953917051	3750025824	1965	20377		0
6	9.0 HEATING	2	3	2488479263	3655843532	437	883		0
7	9.0 WBFPSKTS	0	1	3341013825	4711749663	13	13		0
8	9.0 BMNTGAR	0	2	138248848	1566522609	17	29		0
9	9.0 ATTFRGAR	0	228	5115207373	3170165646	404	69520		0
10	9.0 TOTLVAR	714	4185	3036866359	4310174743	437420	1022544672		0
11	9.0 DECK/DFP	0	738	2073732719	3535065196	33451	10281761		0
12	9.0 ENCLPOR	0	452	3447004608	3317784104	14243	2683521		0

ПЗ 6. Автоматическая классификация данных об ирисах Фишера с помощью карт Кохонена.

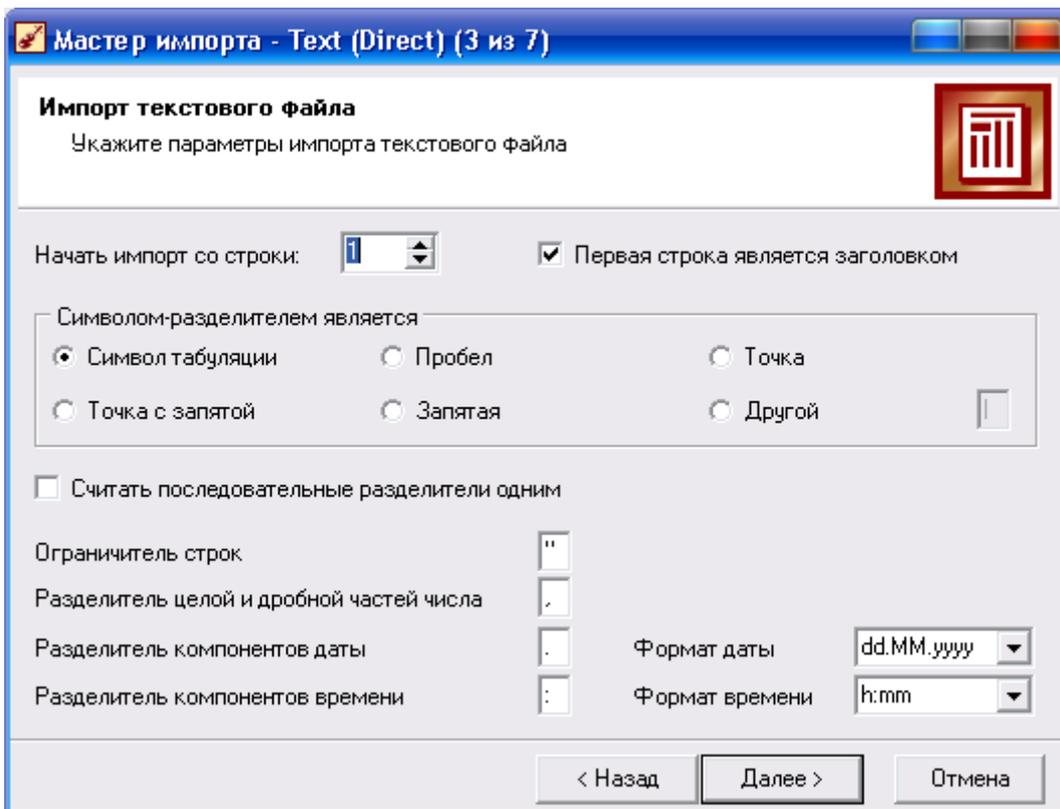
Файл с исходными данными располагаем среди демонстрационных файлов пакета Deductor: ... \BaseGroup\Deductor\Samples.



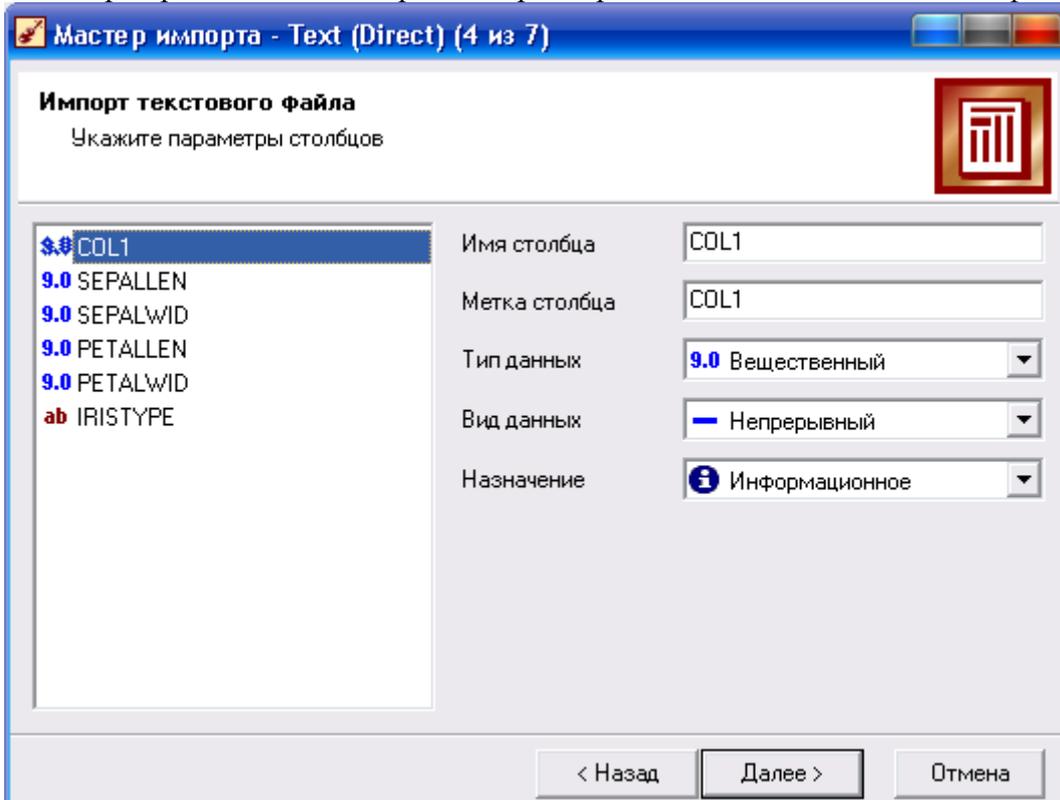
2 кадр мастера импорта содержит указание на адрес исходного файла.



3 кадр мастера импорта позволяет указать пакету параметры файла:

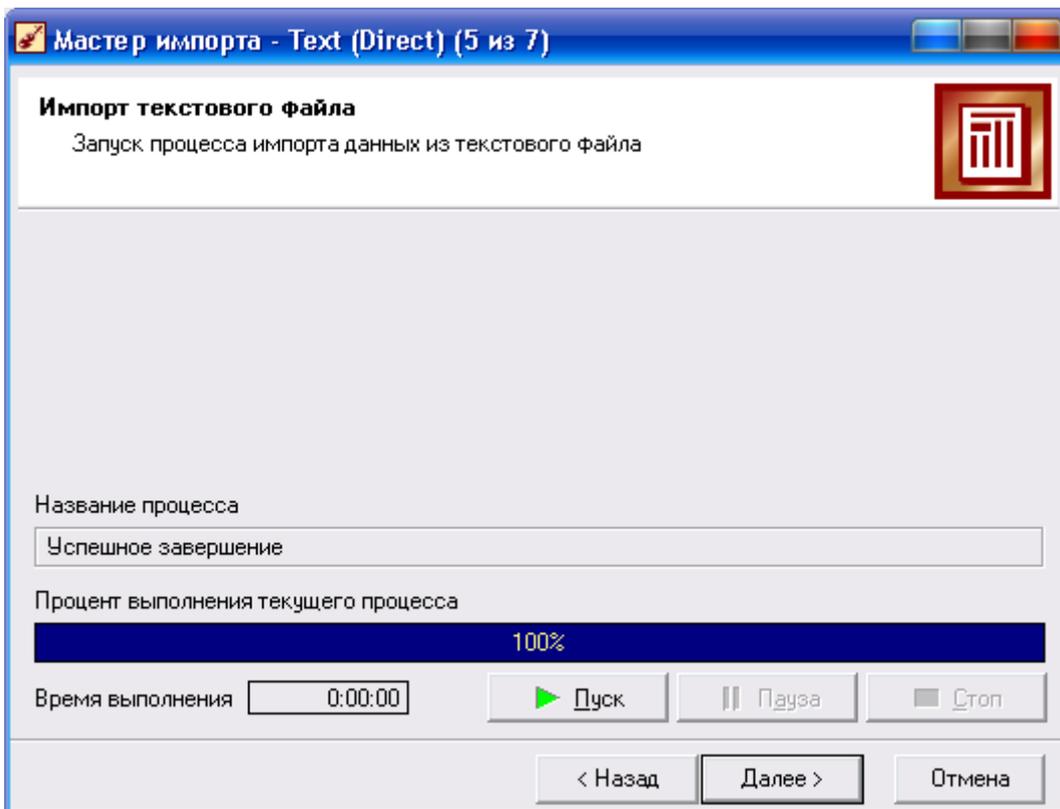


В 4 кадре производится настройка параметров всех столбцов исходного файла.

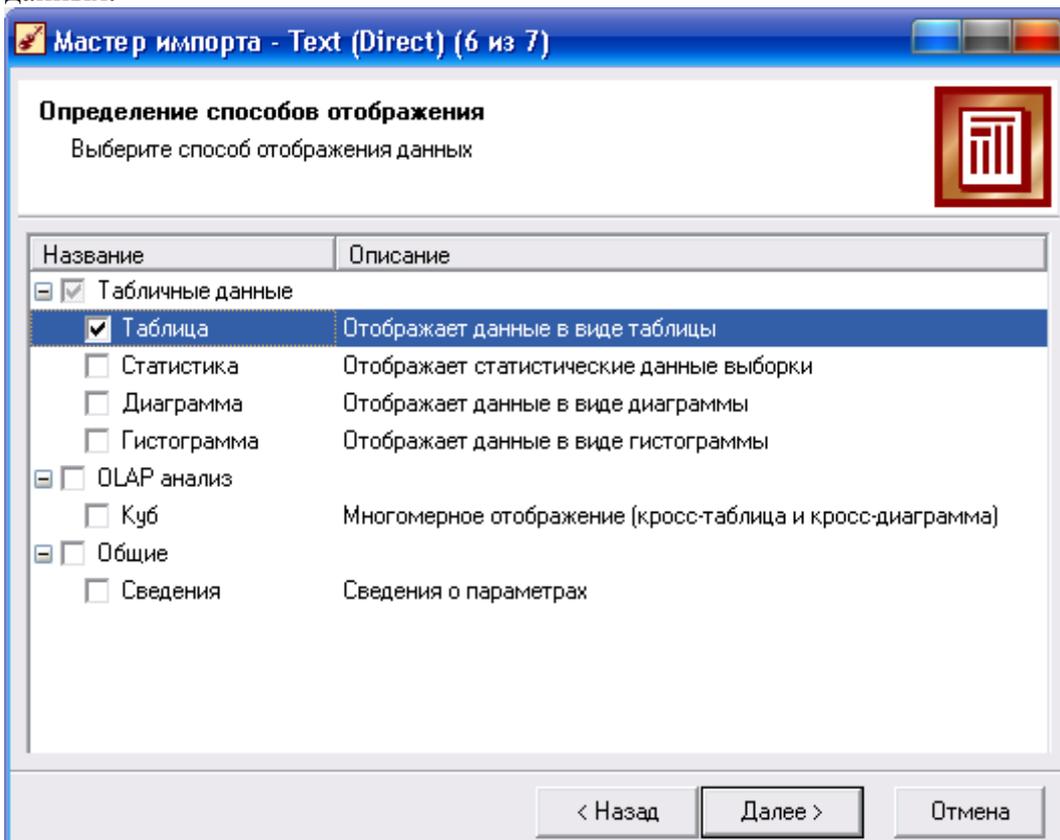


COL1 – информационный, IRISTYPE – выходной, остальные файлы настраиваются, как входные.

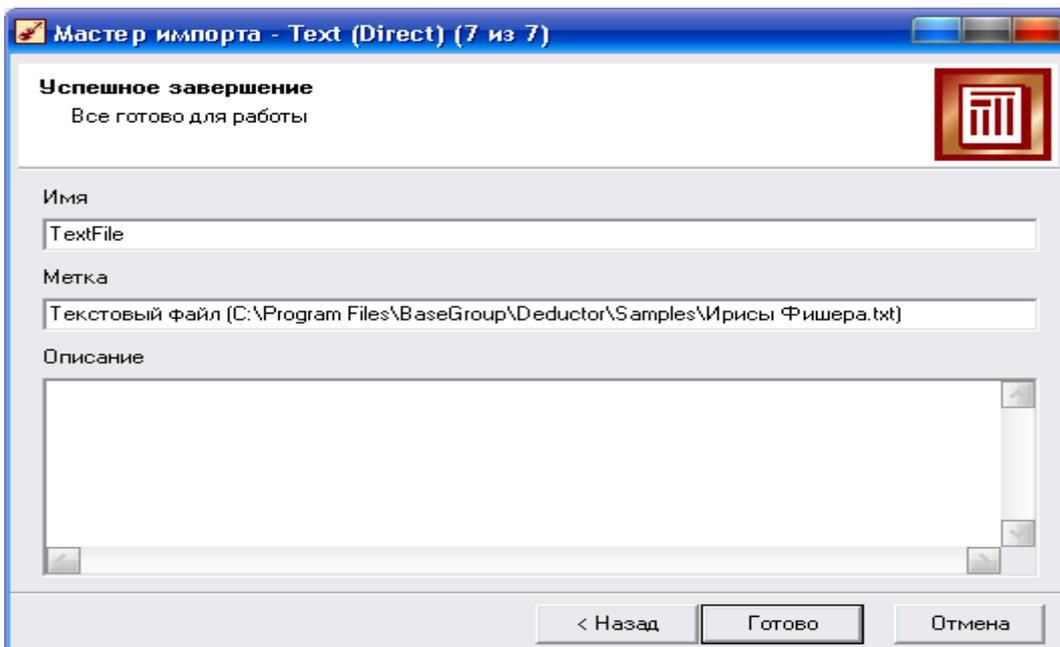
При нажатии кнопки Далее начинается импорт файла с исходными данными. Файл загружается при нажатии кнопки Пуск.



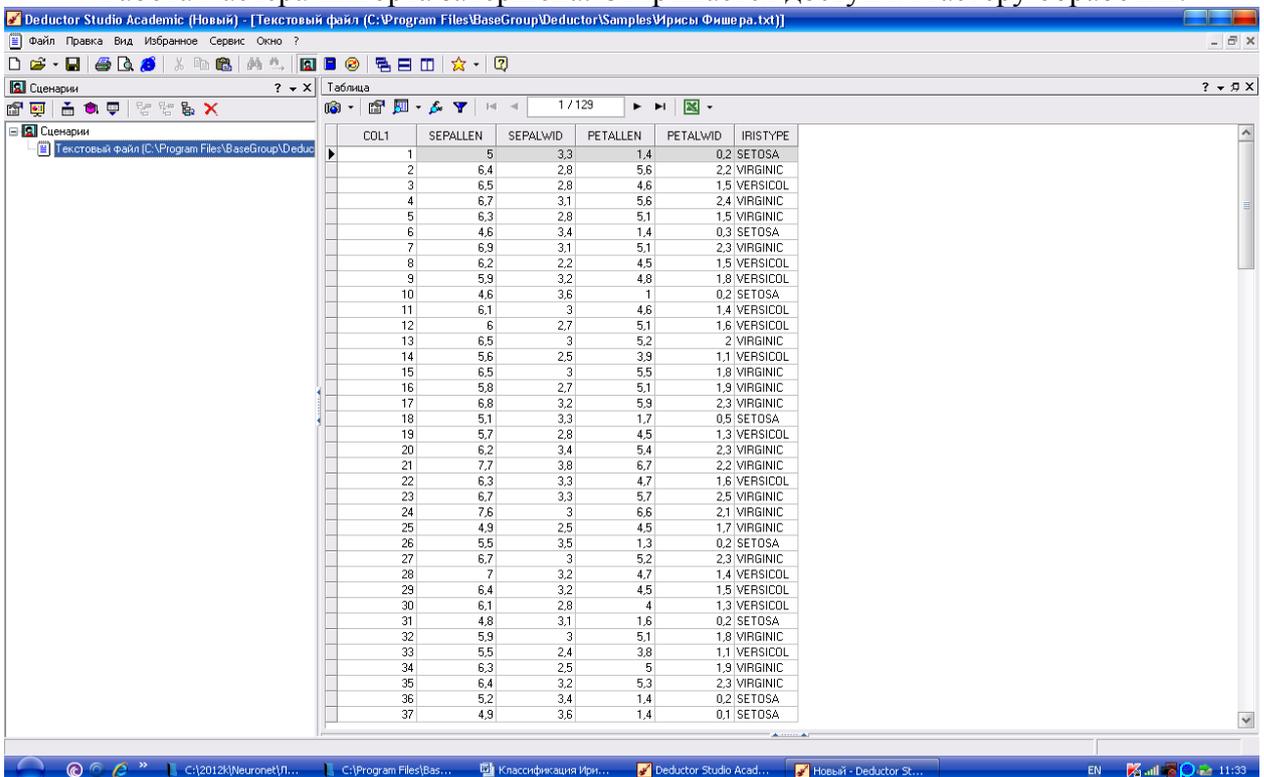
Кнопка Далее выводит на экран 6 окно мастера импорта для выбора способа отображения данных:



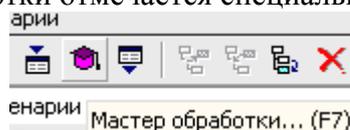
7 кадр завершает работу мастера импорта.



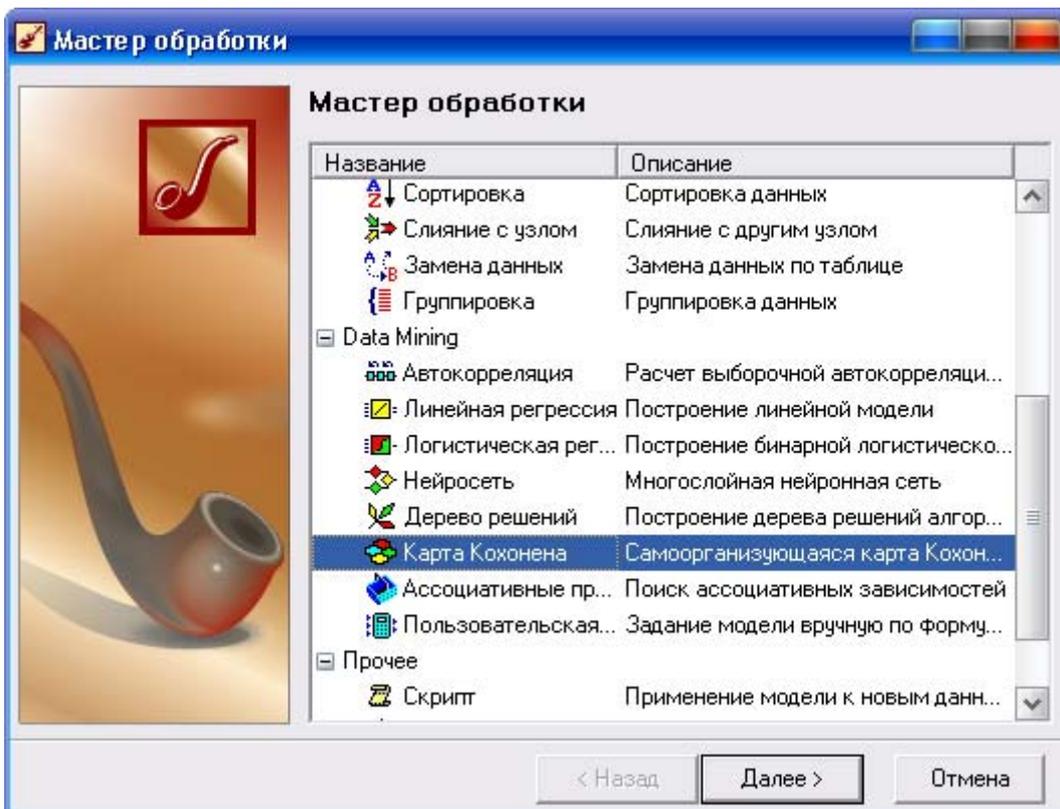
Работа мастера импорта завершена. Открывается доступ к Мастеру обработки.



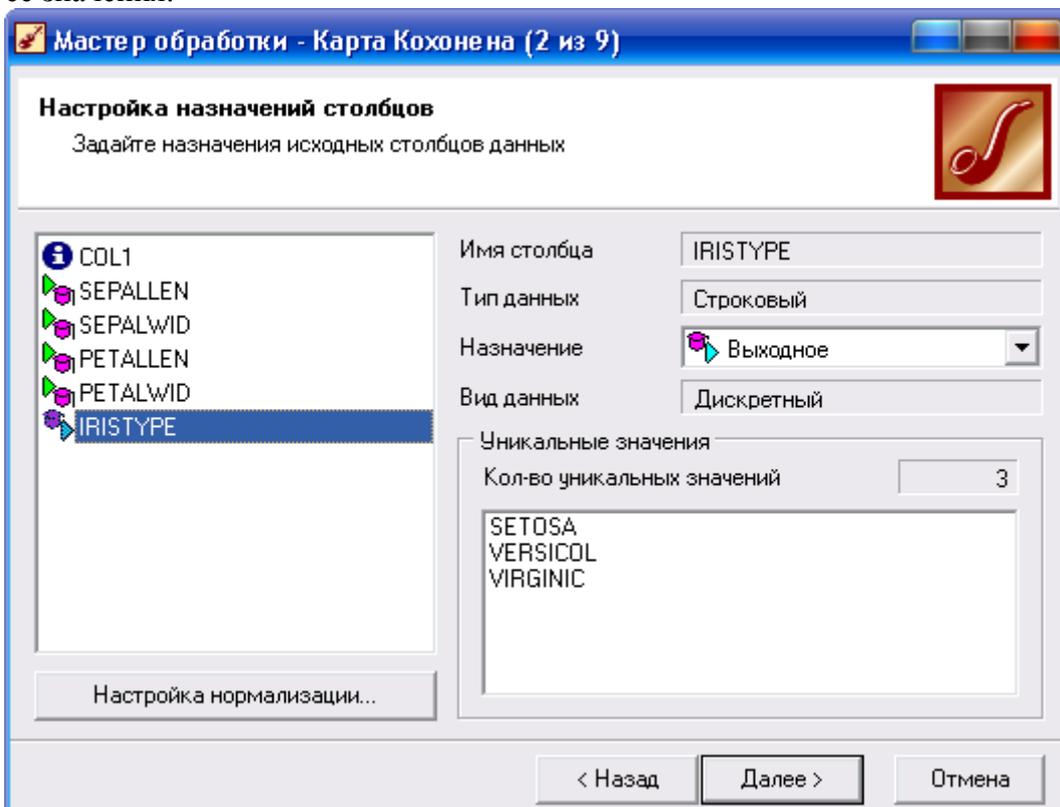
Доступ к Мастеру обработки отмечается специальным значком:



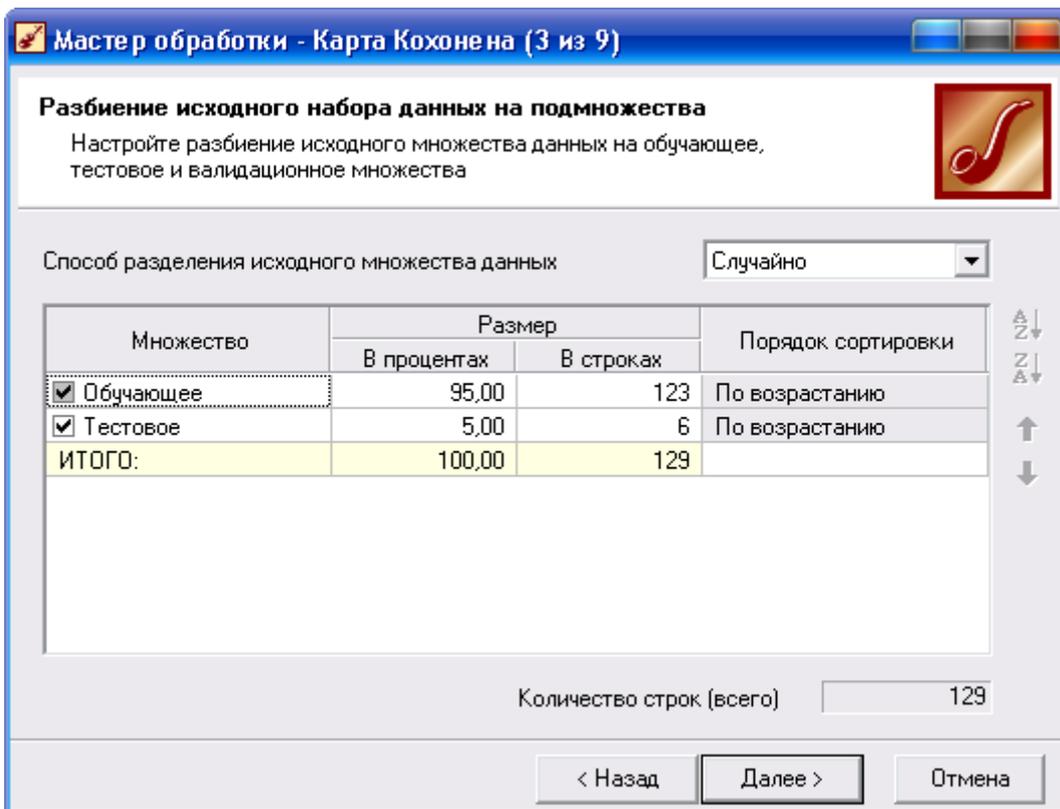
Начало работы с мастером обработки связано с выбором программы для исследования данных.



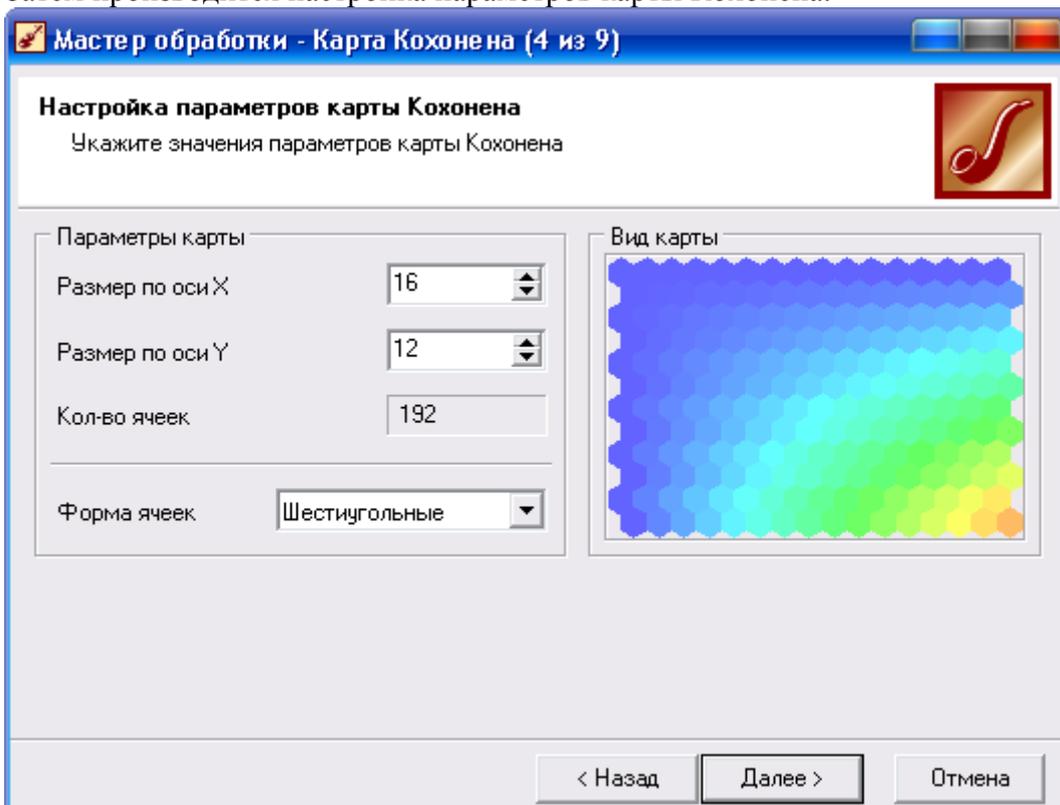
Настройка столбцов с исходными данными в мастере обработки. Мастер приводит статистические данные для каждого столбца. Для выходной переменной указываются все её значения:



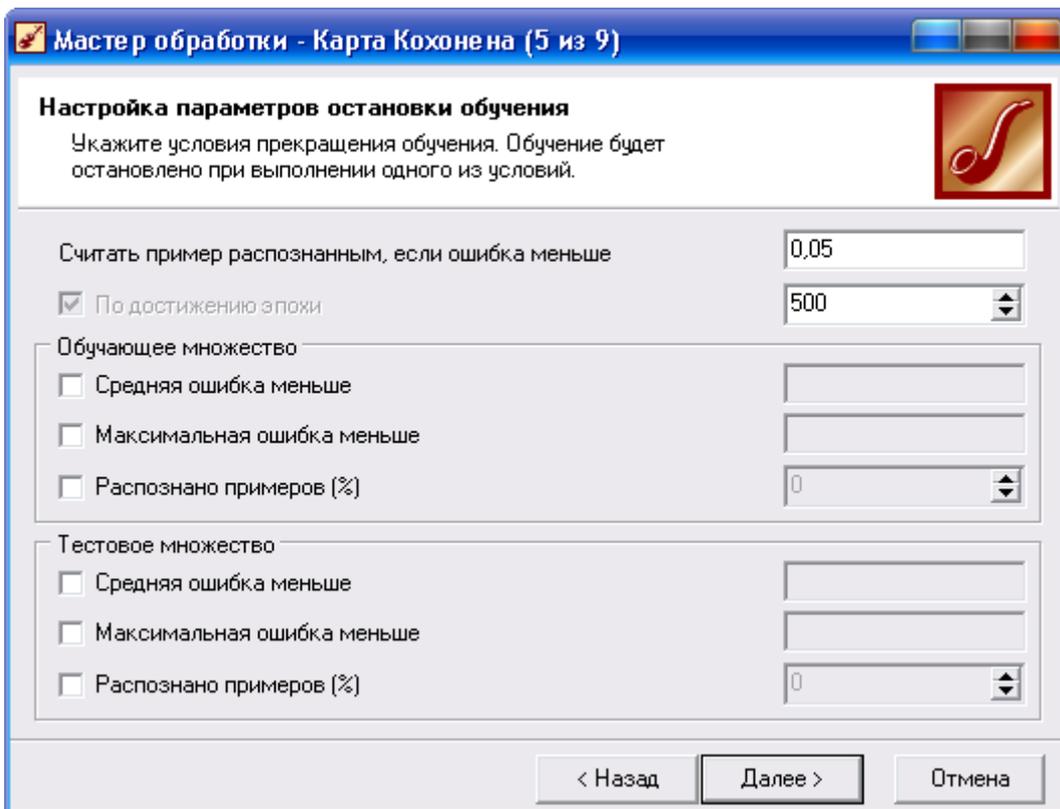
В 3 окне мастера обработки исходный набор данных разбивается на подмножества.



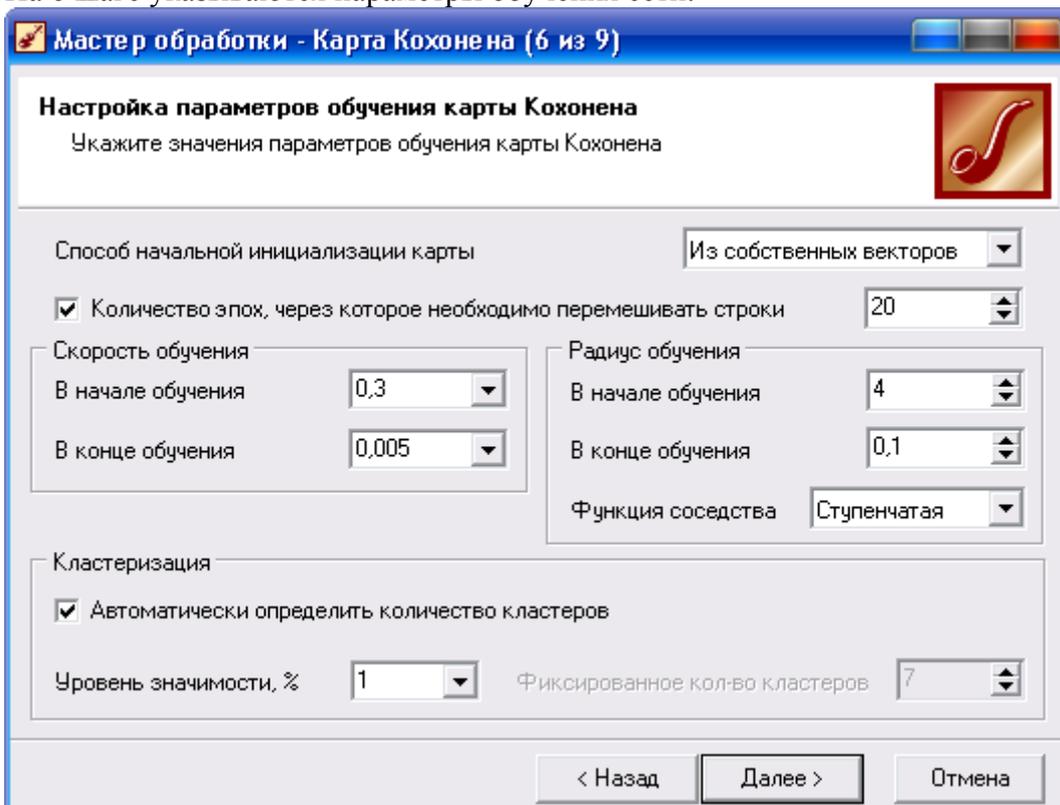
Затем производится настройка параметров карты Кохонена.



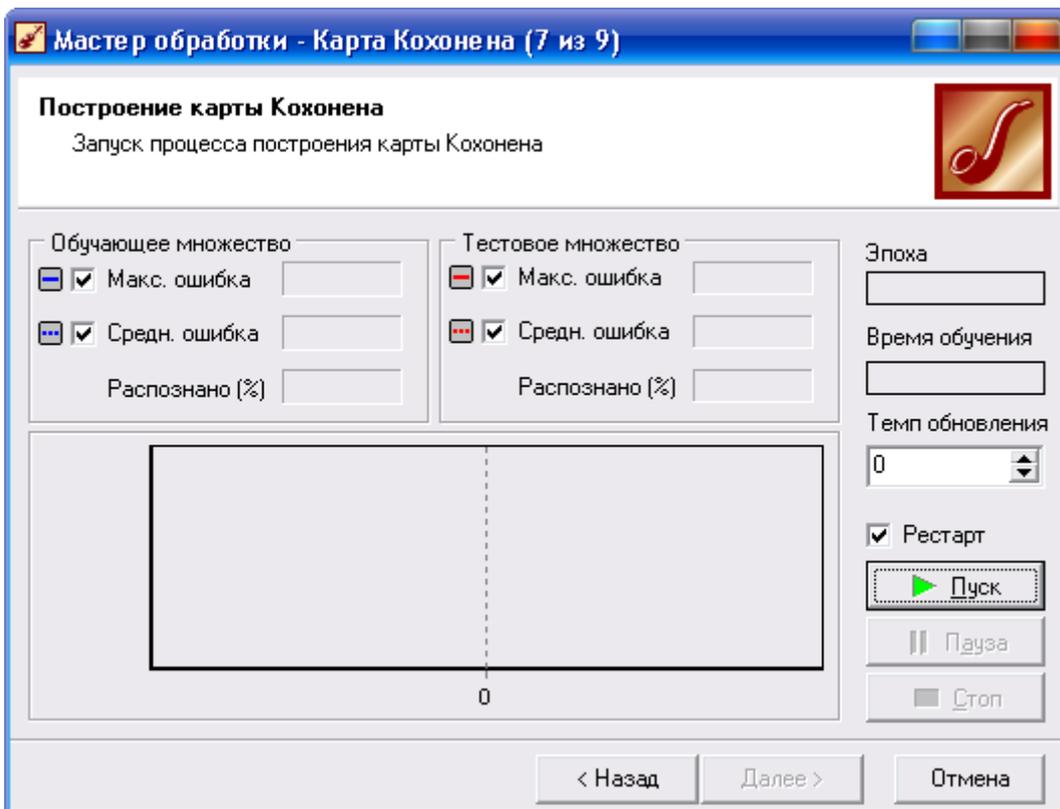
5 шаг мастера обработки позволяет определить условия остановки обучения.



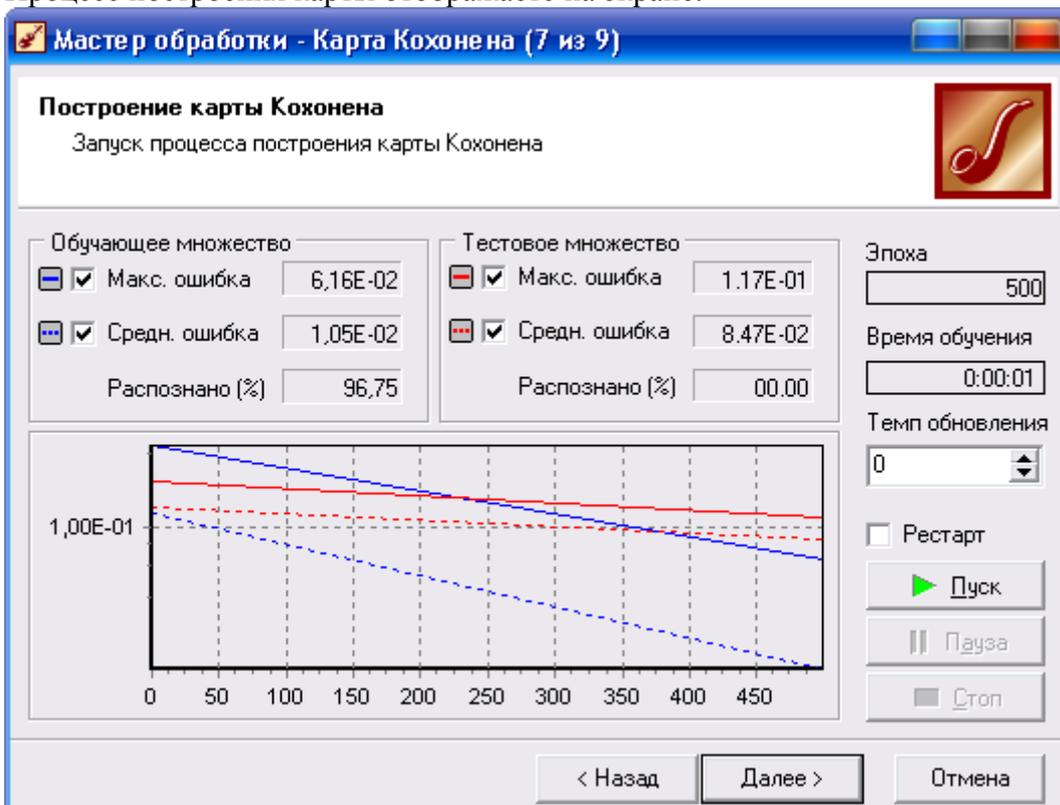
На 6 шаге указываются параметры обучения сети.



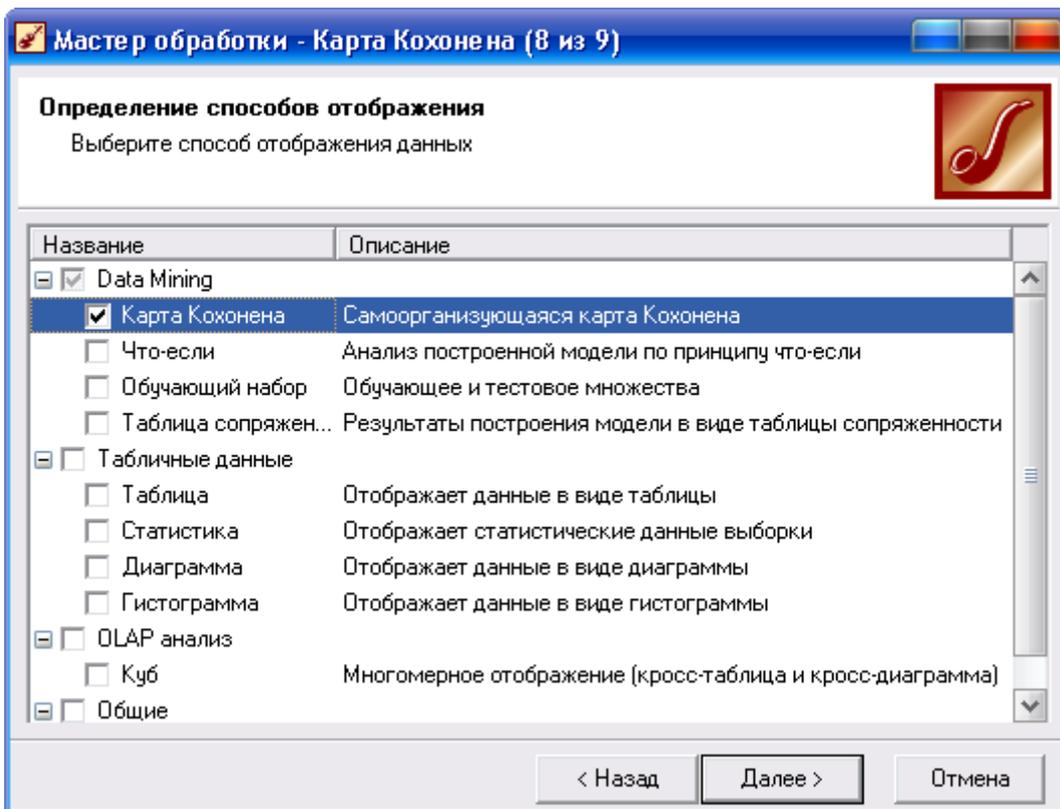
На 7 шаге запускается процесс построения карты Кохонена.



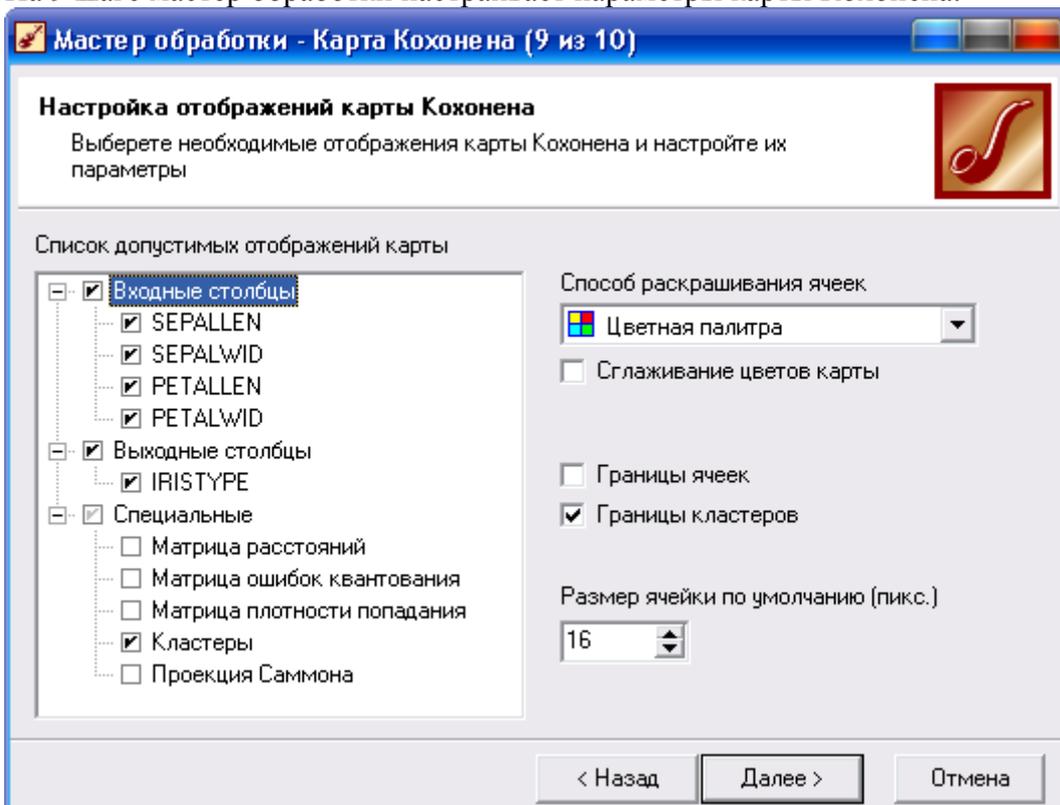
Процесс построения карты отображается на экране.



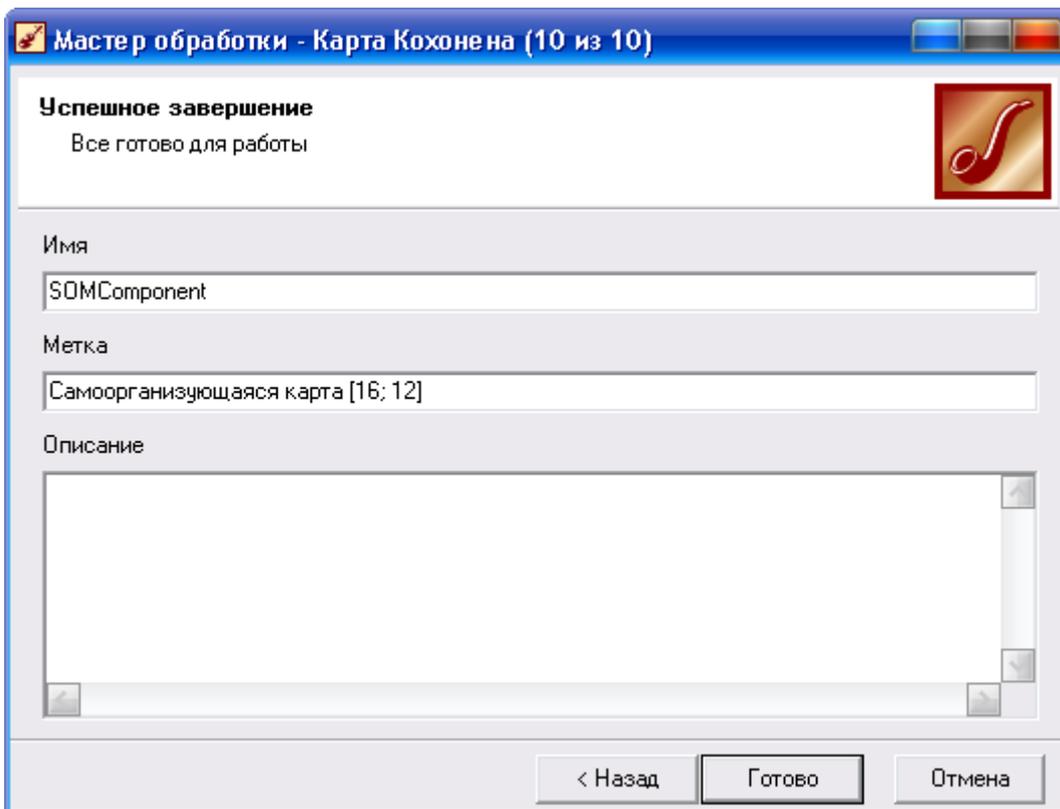
После окончания построения выбирается способ отображения результатов.



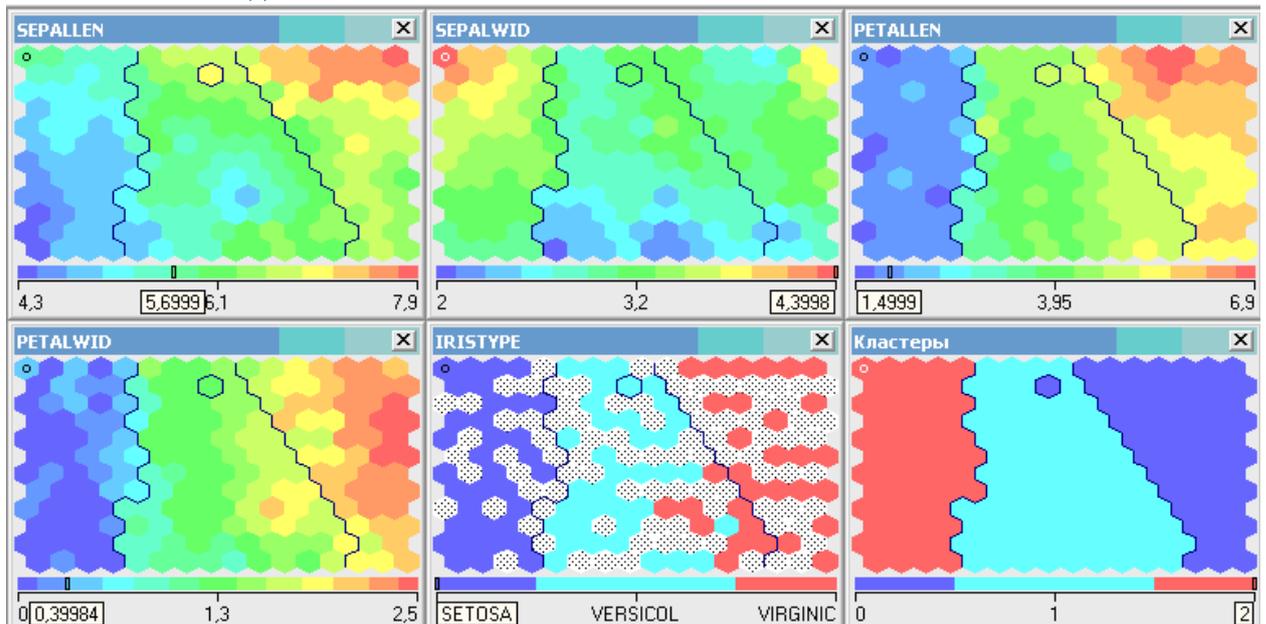
На 9 шаге мастер обработки настраивает параметры карты Кохонена:



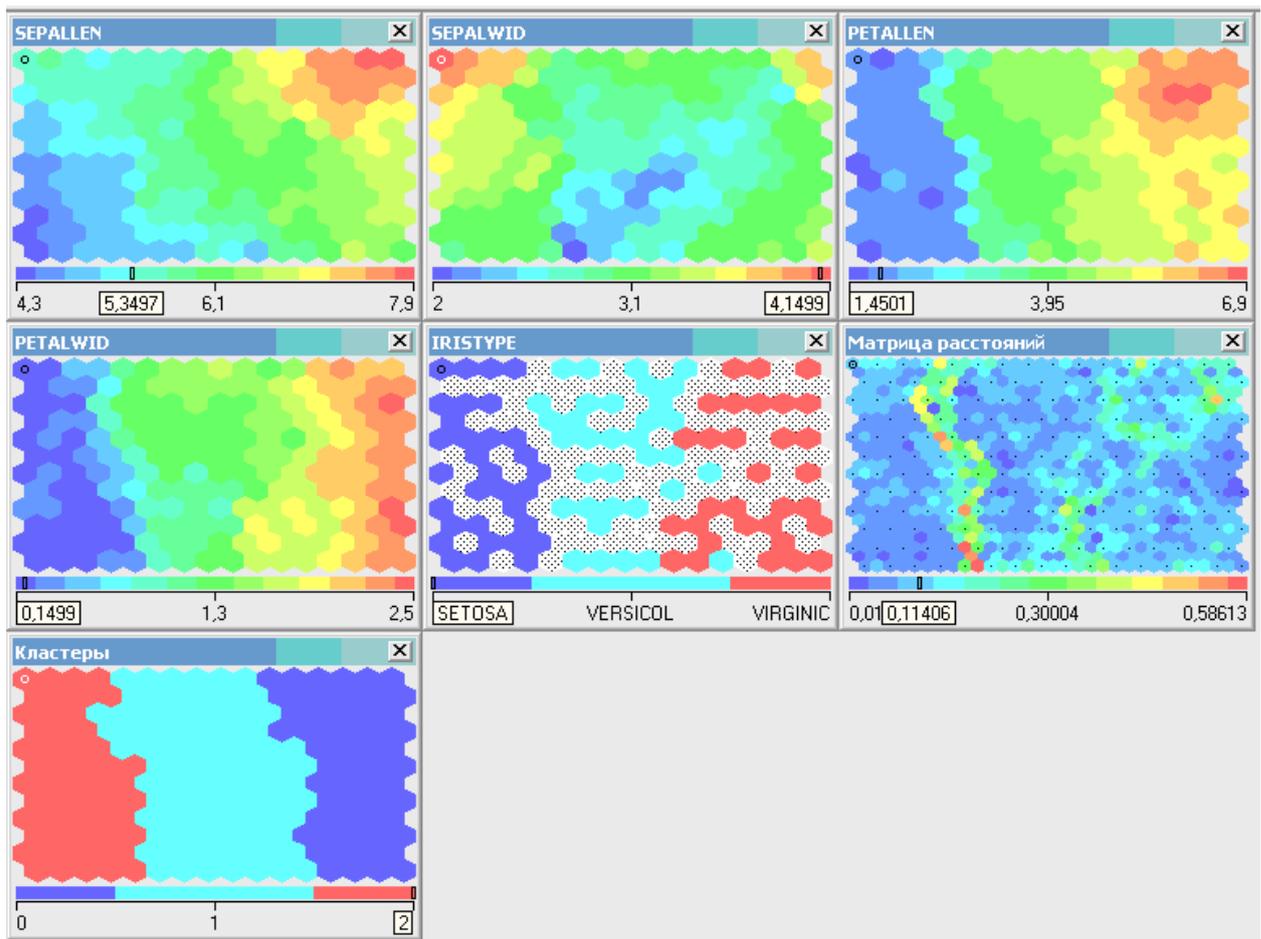
Когда всё готово для работы, можно добавить описание результата.



После окончания работы мастера обработки выводятся графические отображения в соответствии со сделанным заказом.



Можно настроить вывод: без указания границ кластеров, но с демонстрацией матрицы расстояний.



Карта Кохонена демонстрирует чёткое разделение входных данных на три кластера.

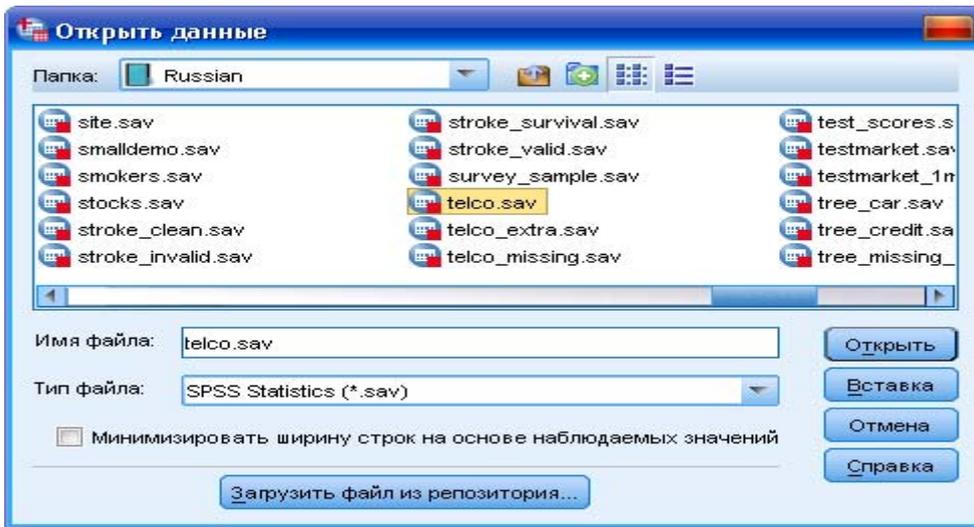
ПЗ 7. Использование радиальной основной функции для классификации телекоммуникационных клиентов.

Рассмотрим демонстрационный пример пакета SPSS – telco.sav:

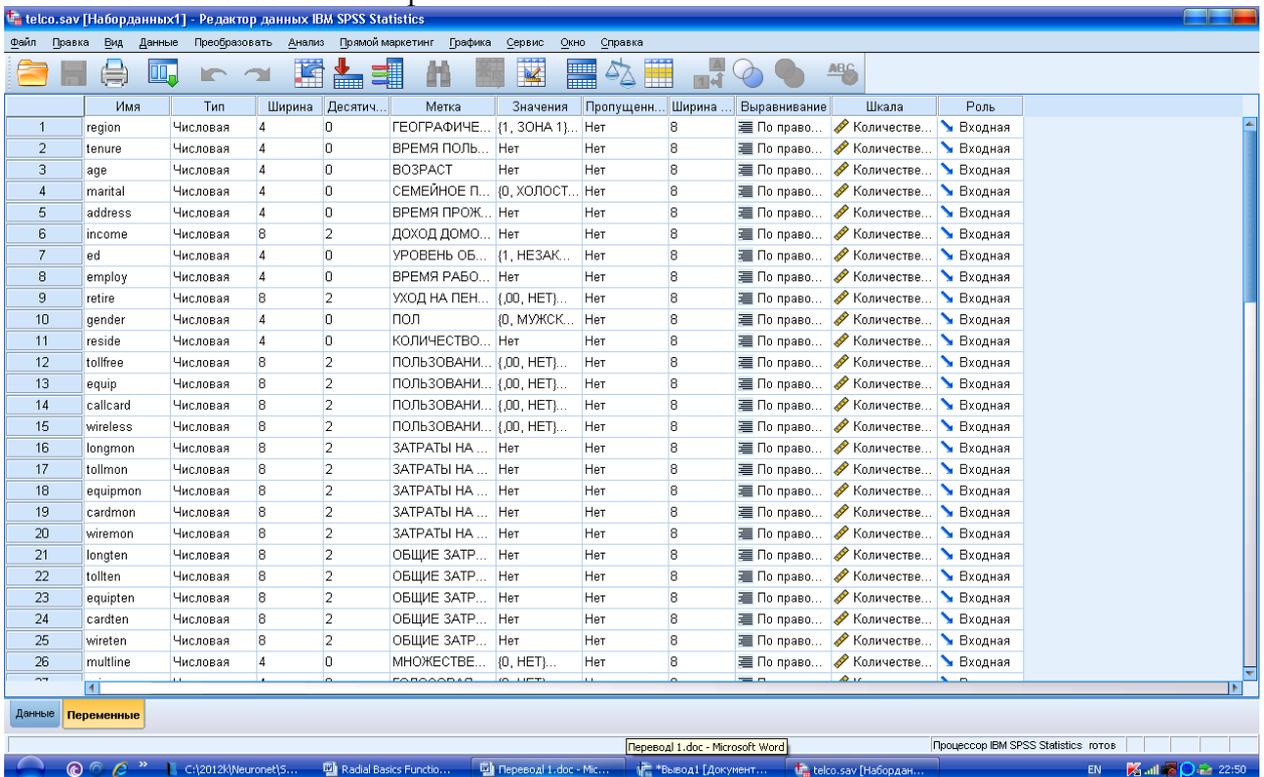
Телекоммуникационный поставщик сегментировал клиентскую базу сервисными образцами использования, поместив клиентов в четыре группы.

Информация о действующих клиентах содержится в
 ...\\IBM\SPSS\Statistics\Samples-Russian\telco.sav.

Это файл гипотетических данных, относящихся к усилиям телекоммуникационной компании уменьшить отток абонентов в своей базе данных клиентов. Каждое наблюдение соответствует отдельному клиенту и регистрирует различную демографическую информацию и информацию о пользовании услугами.



Состав показателей в файле telco.sav:

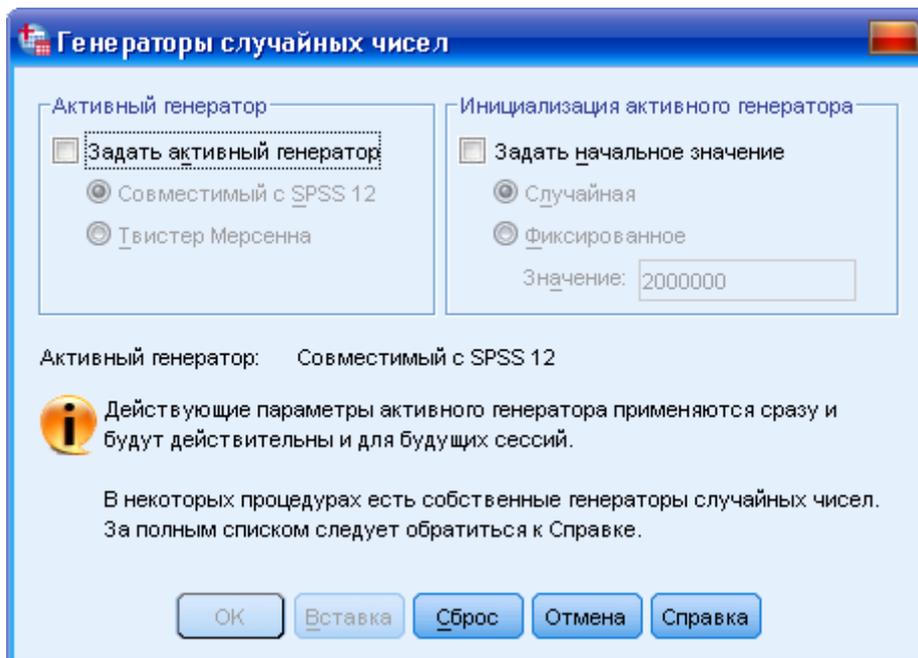


Для классификации клиентов используется Радиальная процедура Основной функции.

Подготовка данных для анализа (перемешивание исходных данных, разделение на обучающее и тестирующее множества, и т.д.) производится в демонстрационном примере с помощью генератора случайных чисел пакета.

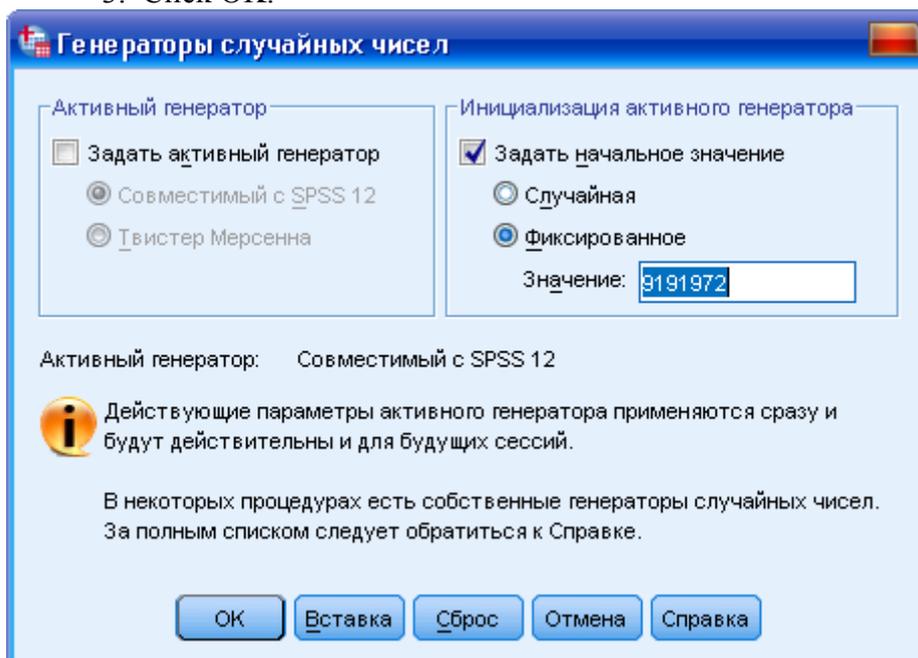
Чтобы настроить генератор случайных чисел, из меню выберите:

Transform > Random Number Generators...

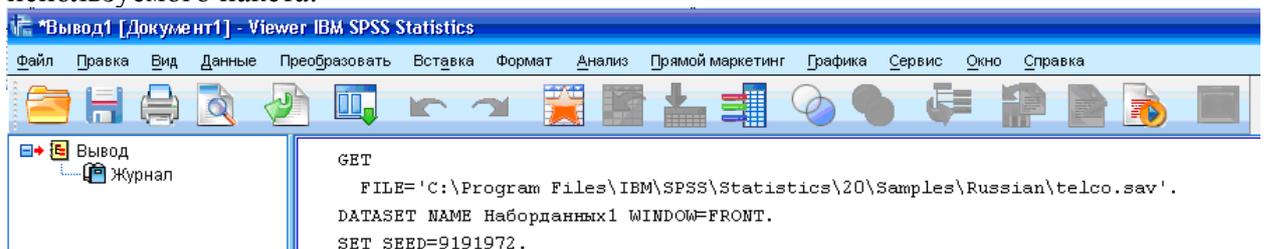


Для управления генератором случайных чисел выполнить:

1. Select Set Starting Point – установить начальную точку.
2. Select Fixed Value, and type 9191972 as the value.
3. Click OK.

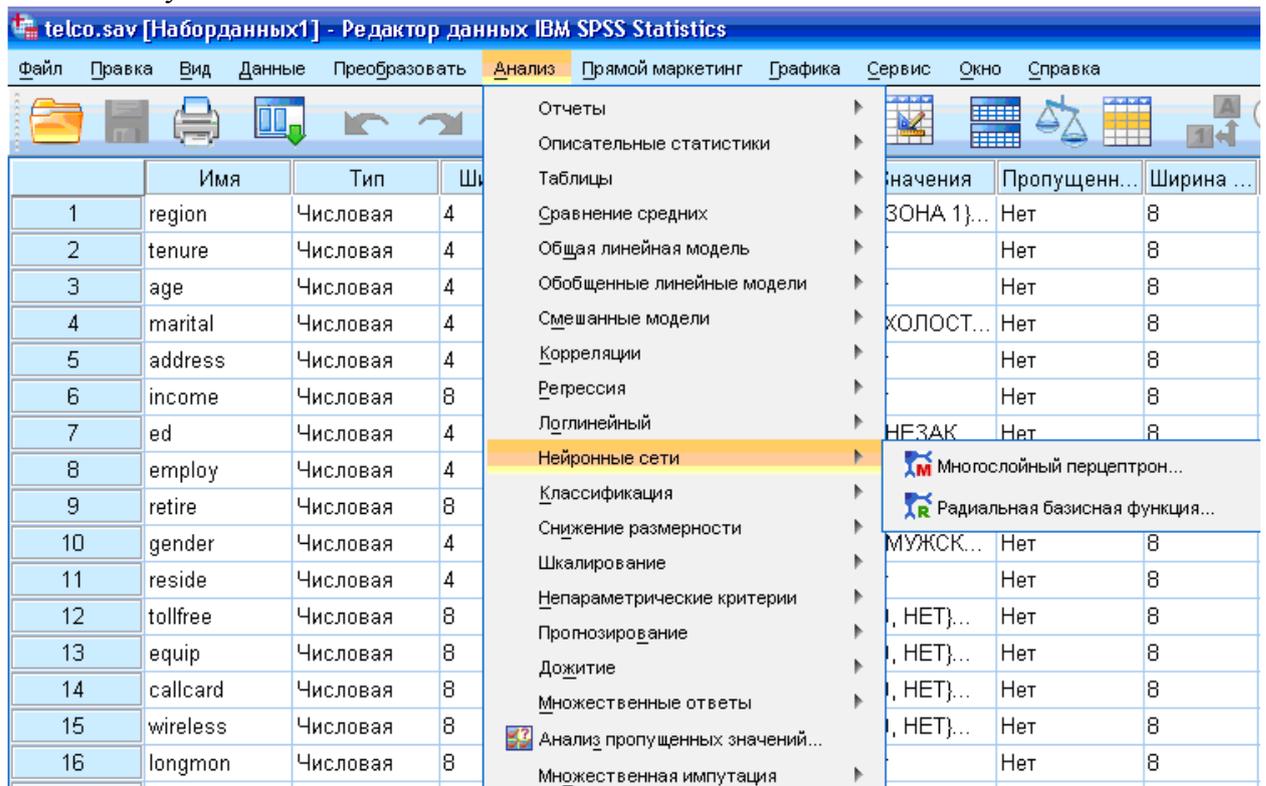


В окне SPSS появляется протокол выполняемых действий в виде команд используемого пакета:

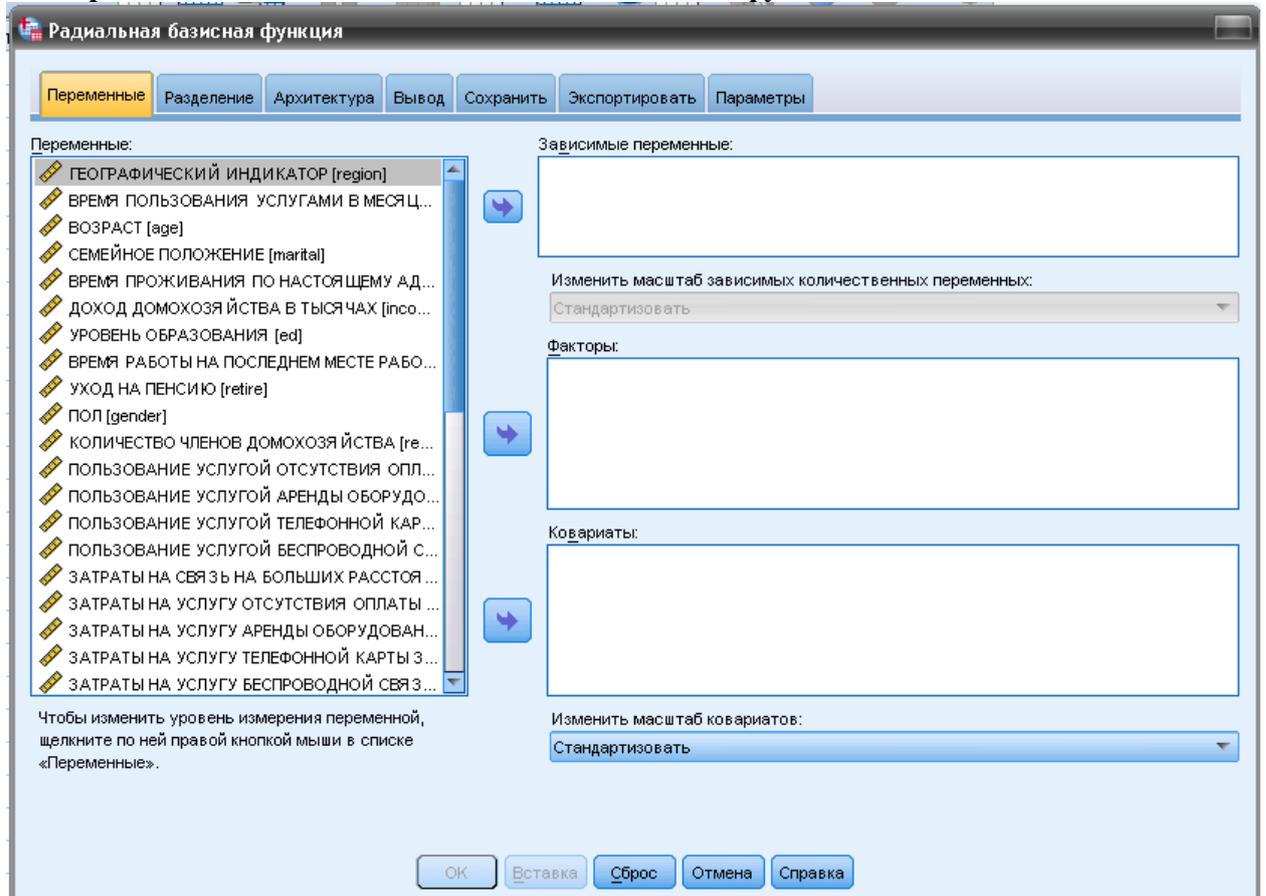


Управление анализом

4. Чтобы управлять Радиальным анализом Основной функции, из меню выберите: Analyze > Neural Networks > Radial Basis Function...



На экране появляется окно Радиальной основной функции:



В этом окне имеется 7 вкладок, позволяющих настроить пакет.

Радиальная Основная функция: Вкладка Variables (Переменные)

С левой стороны перечисляются все имеющиеся в файле переменные. В правой стороне содержится три окна, в которые переносятся соответствующие переменные: зависимые переменные, факторы и ковариаты.

5. Выбирается *Customer category [custcat]* как зависимая переменная.

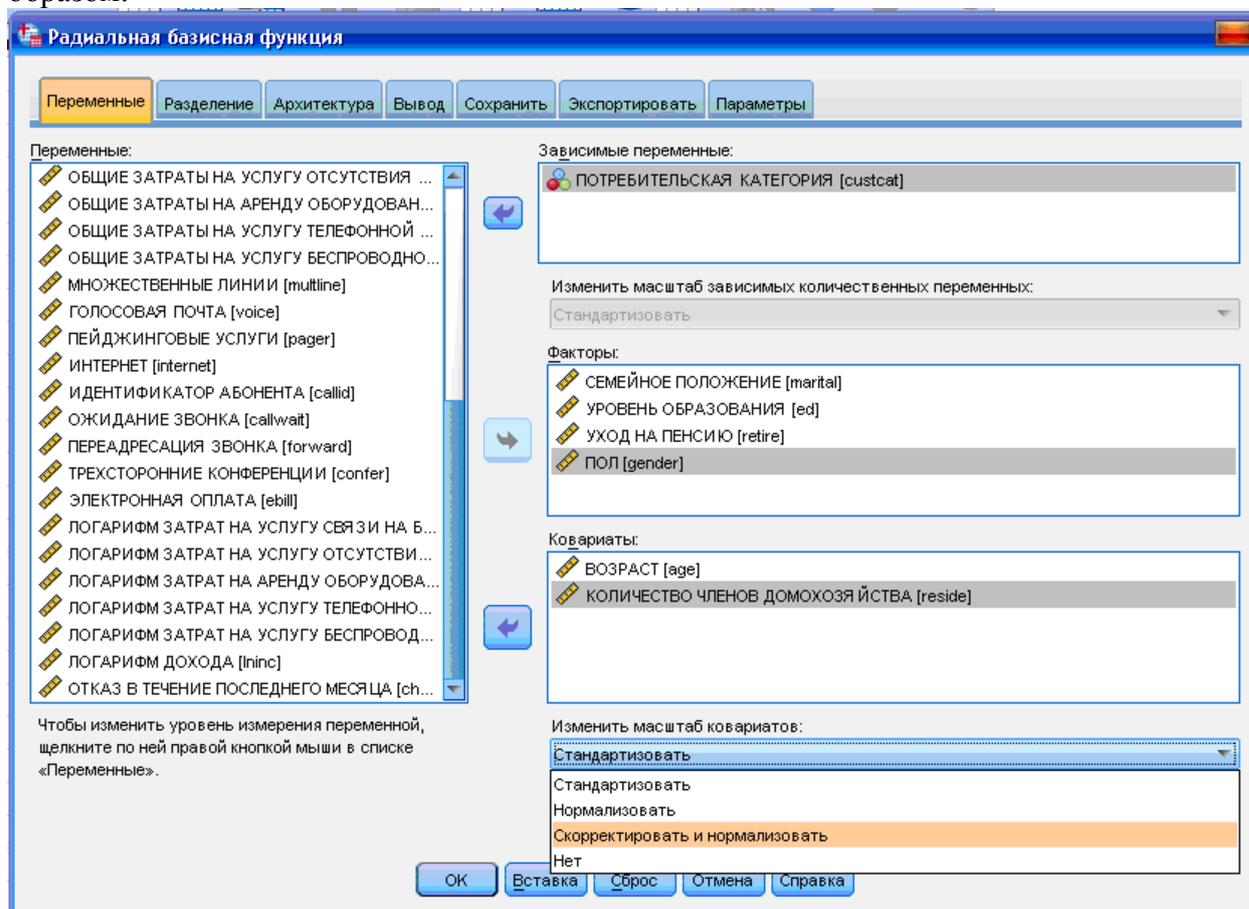
6. Выбираются *Marital status [marital]*, *Level of education [ed]*, *Retired [retire]*, and *Gender [gender]* как факторы.

7. Выбираются *Age in years [age]* (Возраст в годах [возраст]) через *Number of people in household [reside]* (Число людей в домашнем хозяйстве [проживают]) как covariates.

8. Выбирается Adjusted Normalized (Скорректировать Нормализовать), как метод для того, чтобы повторно измерить covariates.

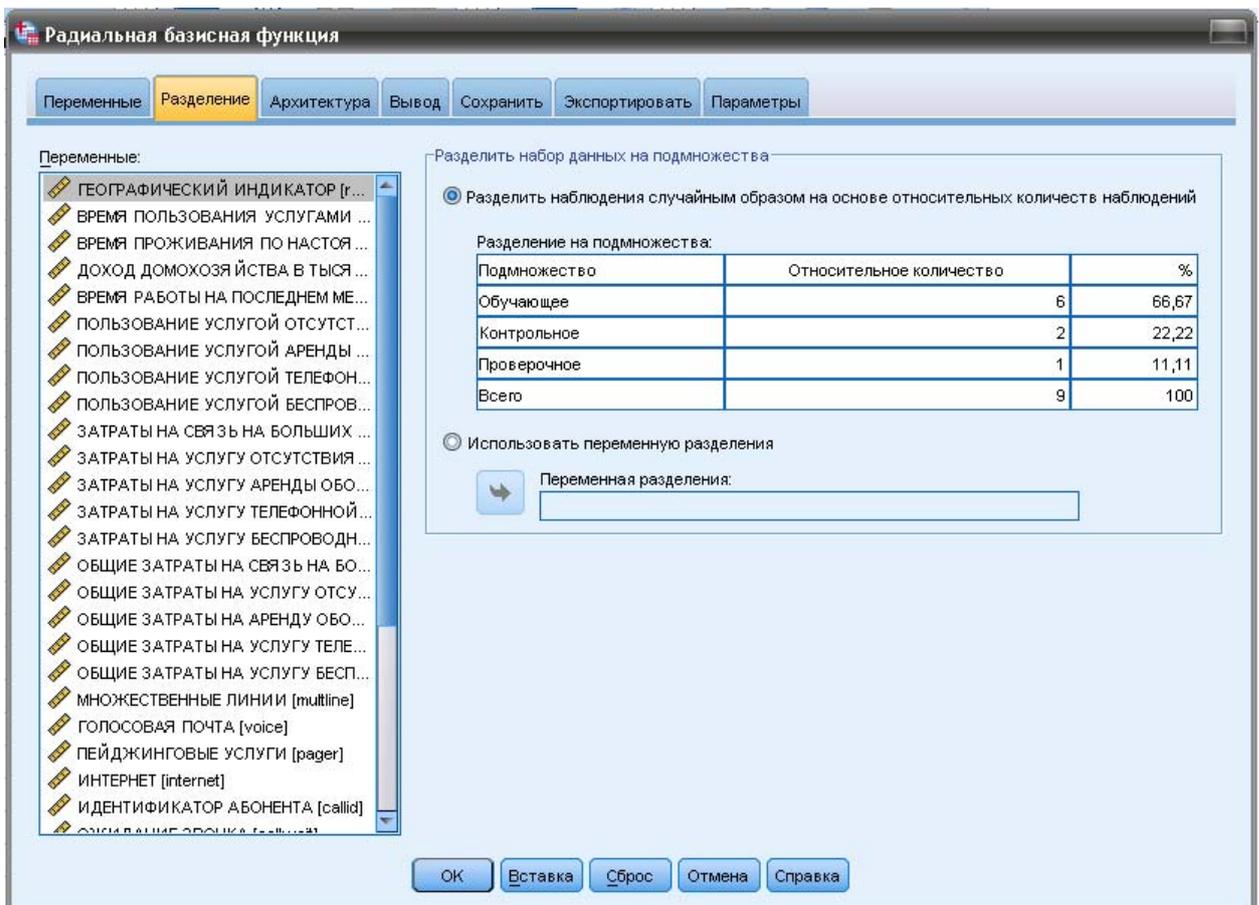
9. Щелкнуть вкладку Partitions.

Вкладка Переменные будет выглядеть следующим образом:



Радиальная Основная функция: Вкладка Partitions (Разделение)

На этой вкладке производится разделение наблюдений на Обучающее, Контрольное и Поверочное подмножества. Это разделение можно выполнить либо при случайном разделении в соответствии с заданным относительным количеством, либо с помощью разделяющей переменной.



Определяя относительные числа случаев, легко создать фракционное разделение, для которого было бы трудно определить проценты. Скажите, что Вы хотите назначить 2/3 набора данных к обучению и 2/3 остающихся случаев к тестированию.

10. Тип 6 как относительное число для учебного образца.

11. Тип 2 как относительное число для образца тестирования.

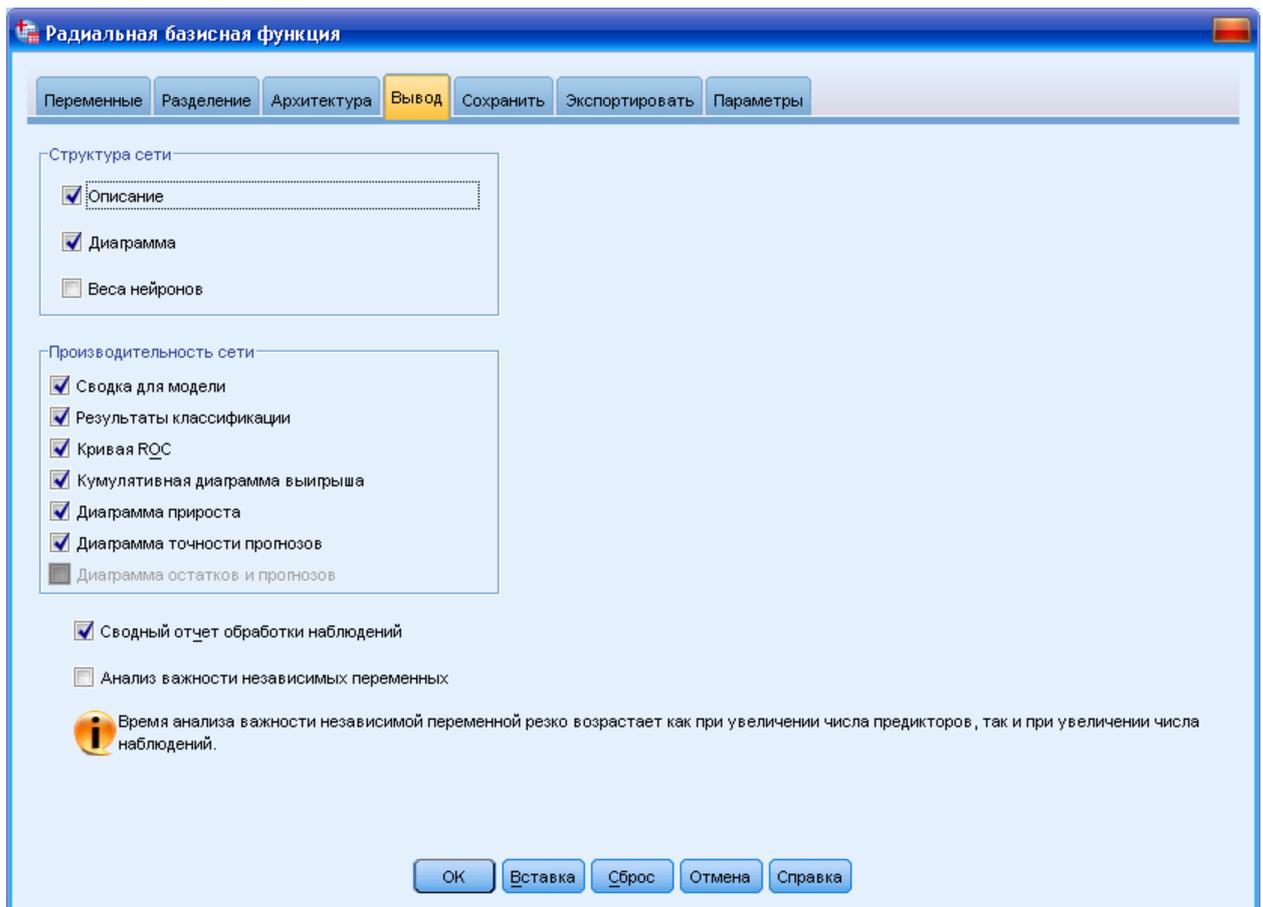
12. Тип 1 как относительное число для образца затяжки.

Были определены в общей сложности 9 относительных случаев. $6/9 = 2/3$, или приблизительно 66.67%, назначены на учебный образец; $2/9$, или приблизительно 22.22%, назначены на тестирование; $1/9$, или приблизительно 11.11% назначены к образцу затяжки.

13. Щелчок по вкладке Output.

Радиальная Основная функция: Вкладка Output (Вывод)

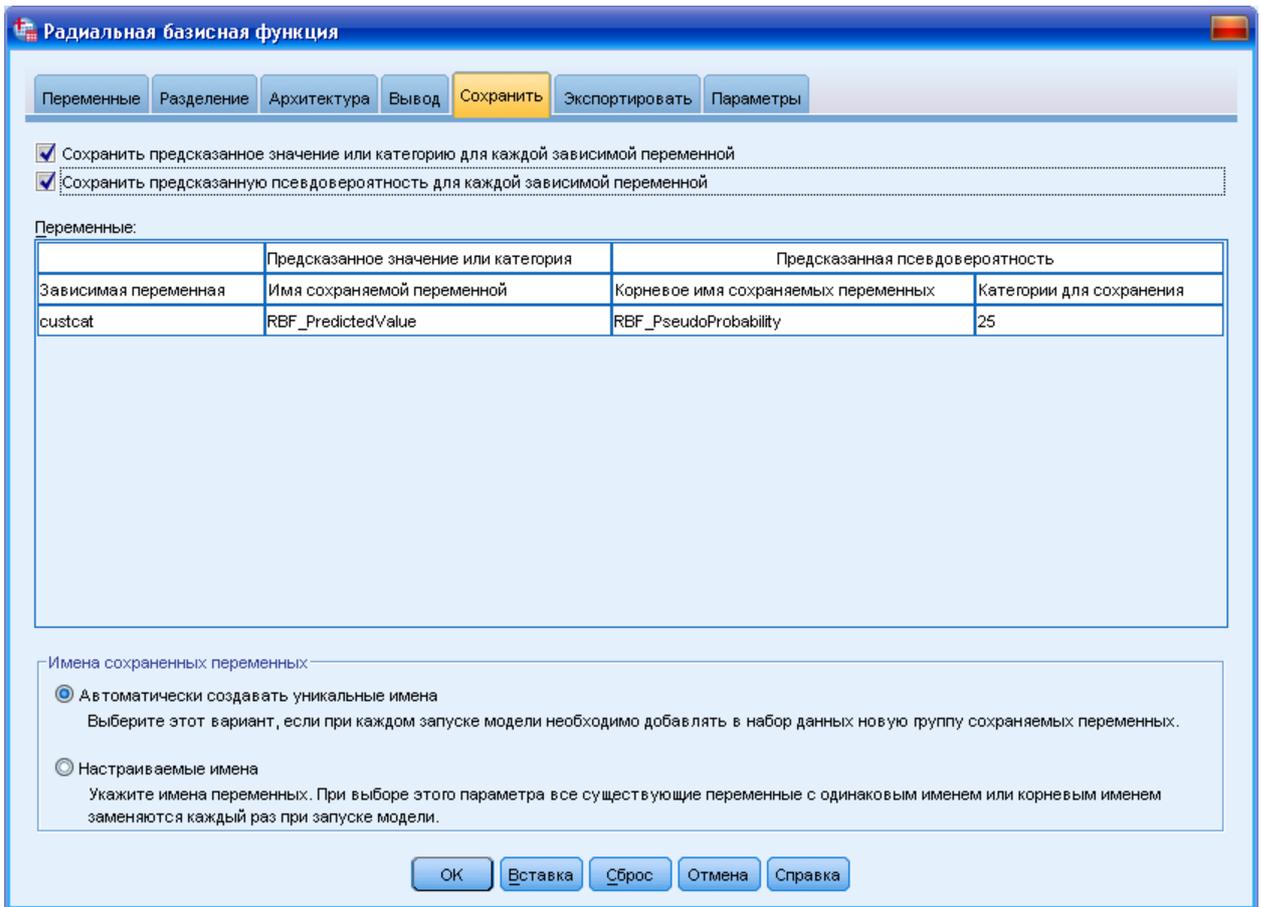
В этой вкладке определяется, что надо будет вывести по окончании исследования: описание сети, диаграмму и веса нейронов. Кроме того, надо определиться, как будет характеризоваться производительность сети, нужны ли сводный отчет обработки наблюдений и анализ важности независимых переменных.



14. Отмечаем Описание и Диаграмма в группе Структура сети.
15. Выбираем кривые: Кривая ROC, (Кумулятивную диаграмму выигрыша) Совокупную диаграмму прибыли, диаграмму Прироста, Диаграмма точности прогнозов и Сводка для модели.
16. Щелчок по вкладке Save.

Радиальная Основная функция: Вкладка Save (Сохранить)

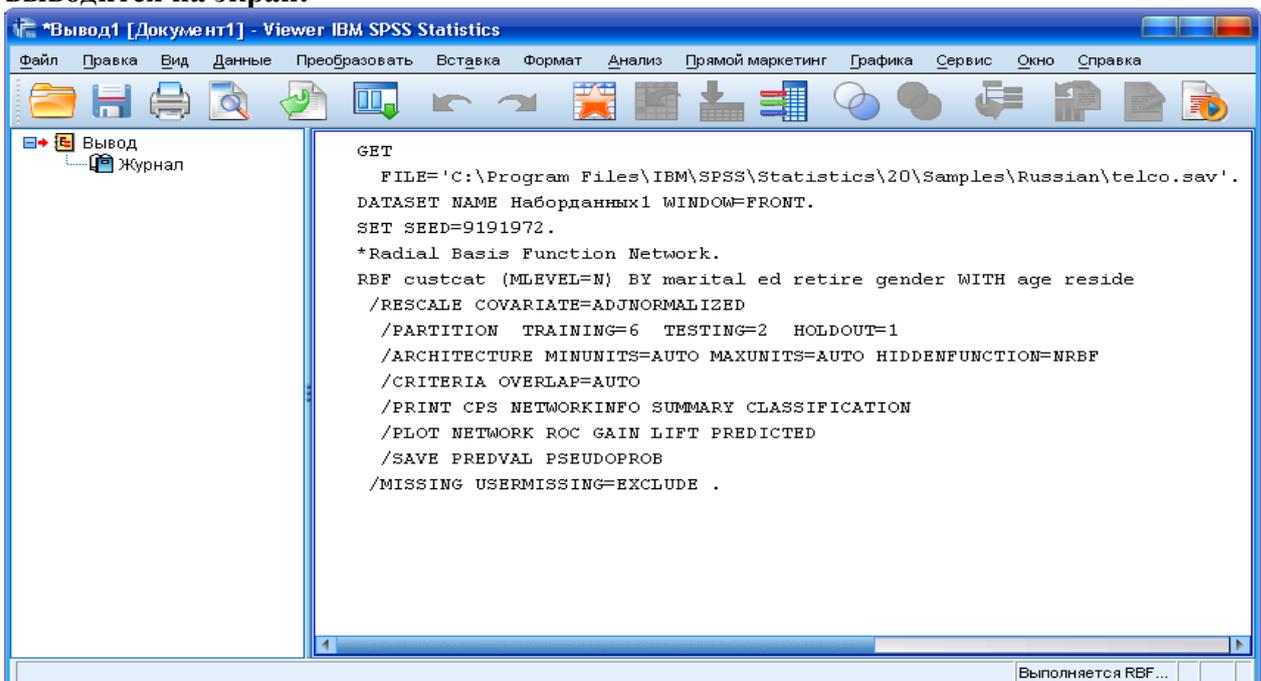
В этой вкладке определяется, надо ли сохранять предсказанное значение или категорию, или предсказанную псевдовероятность для каждой зависимой переменной.



17. Отметить Сохранить ожидаемое значение или категорию для каждой зависимой переменной и Сохранить предсказанную псевдовероятность для каждой зависимой переменной.

18. Нажимаем ОК.

При нажатии кнопки ОК выполняется всё заказанное исследование, а программа его выводится на экран:



Кроме программы исследования выводится отчёт о выполненной работе:

Отчёт о классификации телекоммуникационных клиентов с помощью Радиальной базисной функции.

Программа SPSS:

```

GET
  FILE='C:\Program Files\IBM\SPSS\Statistics\20\Samples-Russian\telco.sav'.
DATASET NAME Наборданных1 WINDOW=FRONT.
SET SEED=9191972.
*Radial Basis Function Network.
RBF custcat (MLEVEL=N) BY marital ed retire gender WITH age reside
/RESCALE COVARIATE=ADJNORMALIZED
/PARTITION TRAINING=6 TESTING=2 HOLDOUT=1
/ARCHITECTURE MINUNITS=AUTO MAXUNITS=AUTO HIDDENFUNCTION=NRBF
/CRITERIA OVERLAP=AUTO
/PRINT CPS NETWORKINFO SUMMARY CLASSIFICATION
/PLOT NETWORK ROC GAIN LIFT PREDICTED
/SAVE PREDVAL PSEUDOPROB
/MISSING USERMISSING=EXCLUDE .
  
```

Радиальная базисная функция

Примечания

Вывод создан		15-DEC-2012 23:38:29
Комментарии		
	Данные	C:\Program Files\IBM\SPSS\Statistics\20\Samples-Russian\telco.sav
	Активный набор данных	Наборданных1
Ввод	Метка файла	Telecommunications customer database
	Фильтр	<нет>
	Толщина	<нет>
	Расщепить файл	<нет>
	Кол-во строк в рабочем файле данных	1000
	Задание пропущенных	Пропущенные пользователем и системой значения рассматриваются как отсутствующие.
Обработка пропущенных значений	Использованные наблюдения	Статистики основаны на наблюдениях с допустимыми значениями для всех переменных, используемых в процедуре.
Обработка весов		неприменимо

Редактор синтаксиса			RBF custcat (MLEVEL=N) BY marital ed retire gender WITH age reside /RESCALE COVARIATE=ADJNORMAL IZED /PARTITION TRAINING=6 TESTING=2 HOLDOUT=1 /ARCHITECTURE MINUNITS=AUTO MAXUNITS=AUTO HIDDENFUNCTION=NRBF /CRITERIA OVERLAP=AUTO /PRINT CPS NETWORKINFO SUMMARY CLASSIFICATION /PLOT NETWORK ROC GAIN LIFT PREDICTED /SAVE PREDVAL PSEUDOPROB /MISSING USERMISSING=EXCLUDE .
Ресурсы	Процессорное время		00:00:08,02
	Время вычислений		00:00:33,19
	Предсказанное значение		RBF_PredictedValue
Созданные или преобразованные переменные		custcat=1	RBF_PseudoProbability_1
	Предсказанная псевдовероятность	custcat=2	RBF_PseudoProbability_2
		custcat=3	RBF_PseudoProbability_3
		custcat=4	RBF_PseudoProbability_4

[Наборданных1]

C:\Program Files\IBM\SPSS\Statistics\20\Samples-Russian\telco.sav

Сводка обработки наблюдений

		N	Процент
Выборка	Обучающая	665	66,5%
	Контрольная	224	22,4%
	Проверочная	111	11,1%
Валидные		1000	100,0%
Исключенные		0	
Всего		1000	

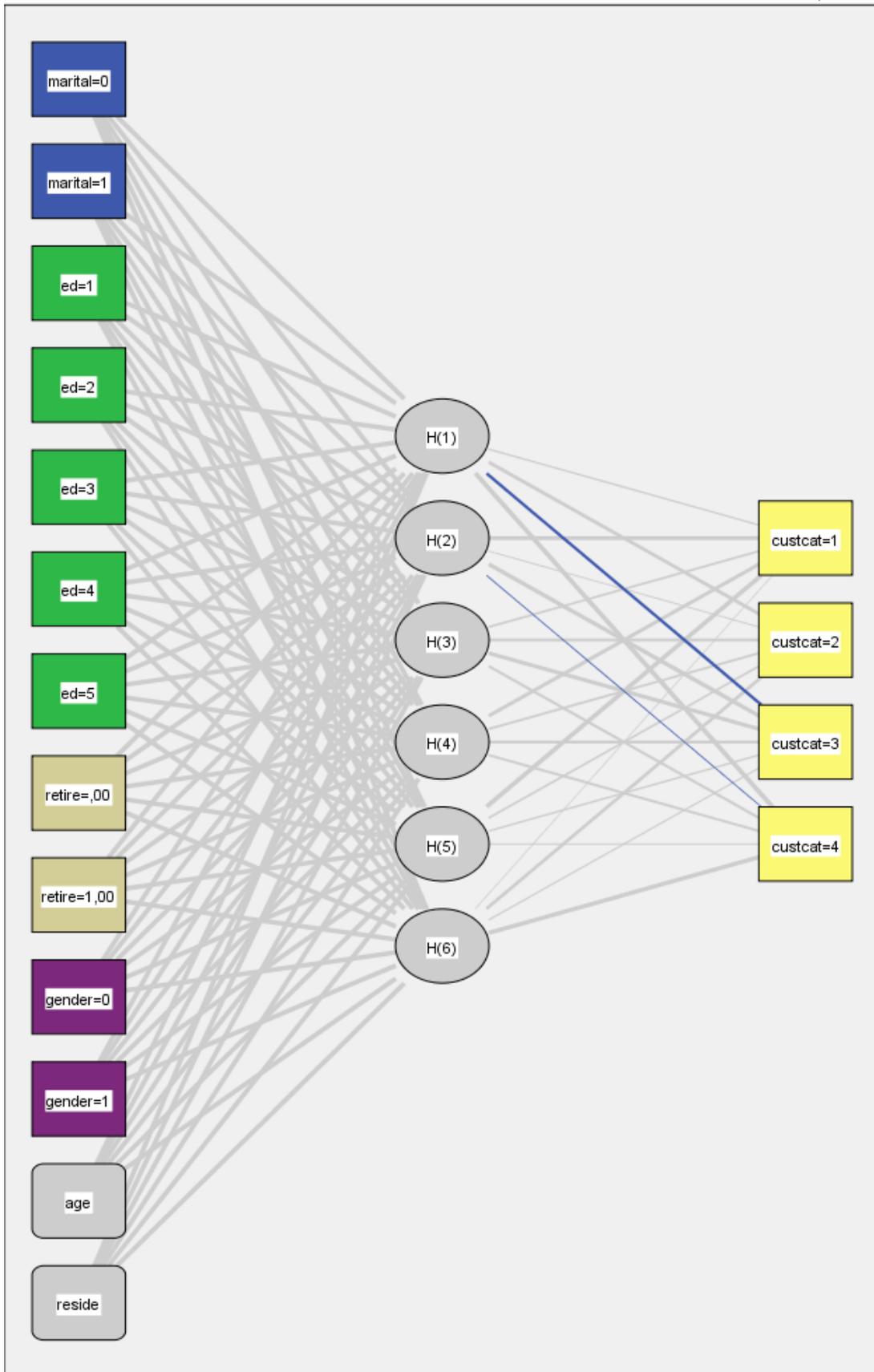
Структура нейронной сети.

Информация о сети

		1	СЕМЕЙНОЕ ПОЛОЖЕНИЕ
	Факторы	2	УРОВЕНЬ ОБРАЗОВАНИЯ
		3	УХОД НА ПЕНСИЮ
		4	ПОЛ
Входной слой		1	ВОЗРАСТ
	Ковариаты	2	КОЛИЧЕСТВО ЧЛЕНОВ ДОМОХОЗЯЙСТ ВА
	Количество нейронов		13
	Метод изменения масштаба для ковариат		Скорректировано и нормализовано
Скрытый слой	Количество нейронов		6 ^a
	Функция активации		Softmax
	Зависимые переменные	1	ПОТРЕБИТЕЛЬС КАЯ КАТЕГОРИЯ
Выходной слой	Количество нейронов		4
	Функция активации		Единичная матрица
	Функция ошибки		Сумма квадратов

а. Определяется по критерию проверки данных: «лучшее» количество скрытых нейронов — это количество, при котором ошибка в тестовых данных минимальна.

— Вес нейрона > 0
— Вес нейрона < 0



Функция активации скрытого слоя: Softmax

Функция активации выходного слоя: Единичная матрица

Результаты исследования.

Сводка для модели

	Ошибка суммы квадратов	236,339
Обучающая	Процент неверных предсказаний	62,0%
	Время обучения	0:00:02,73
Контрольная	Ошибка суммы квадратов	80,329 ^a
	Процент неверных предсказаний	64,7%
Проверочная	Процент неверных предсказаний	61,3%

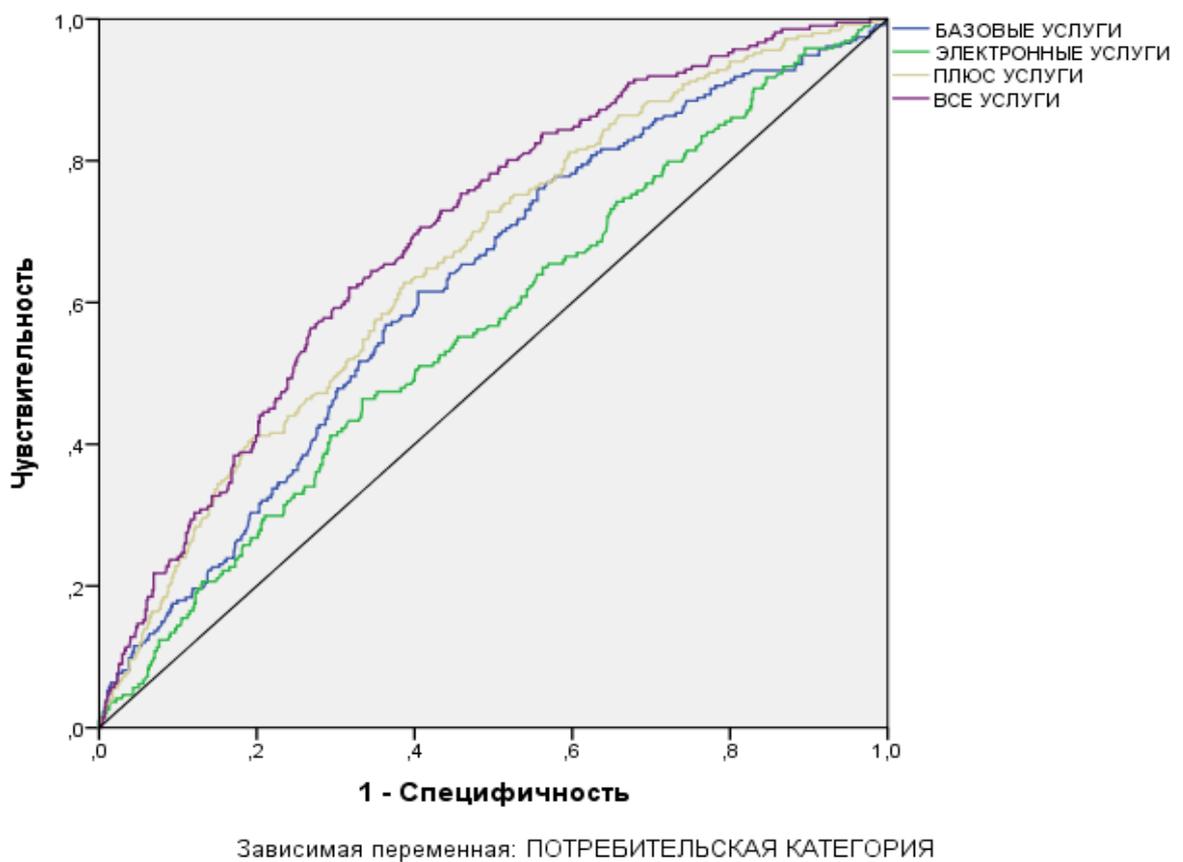
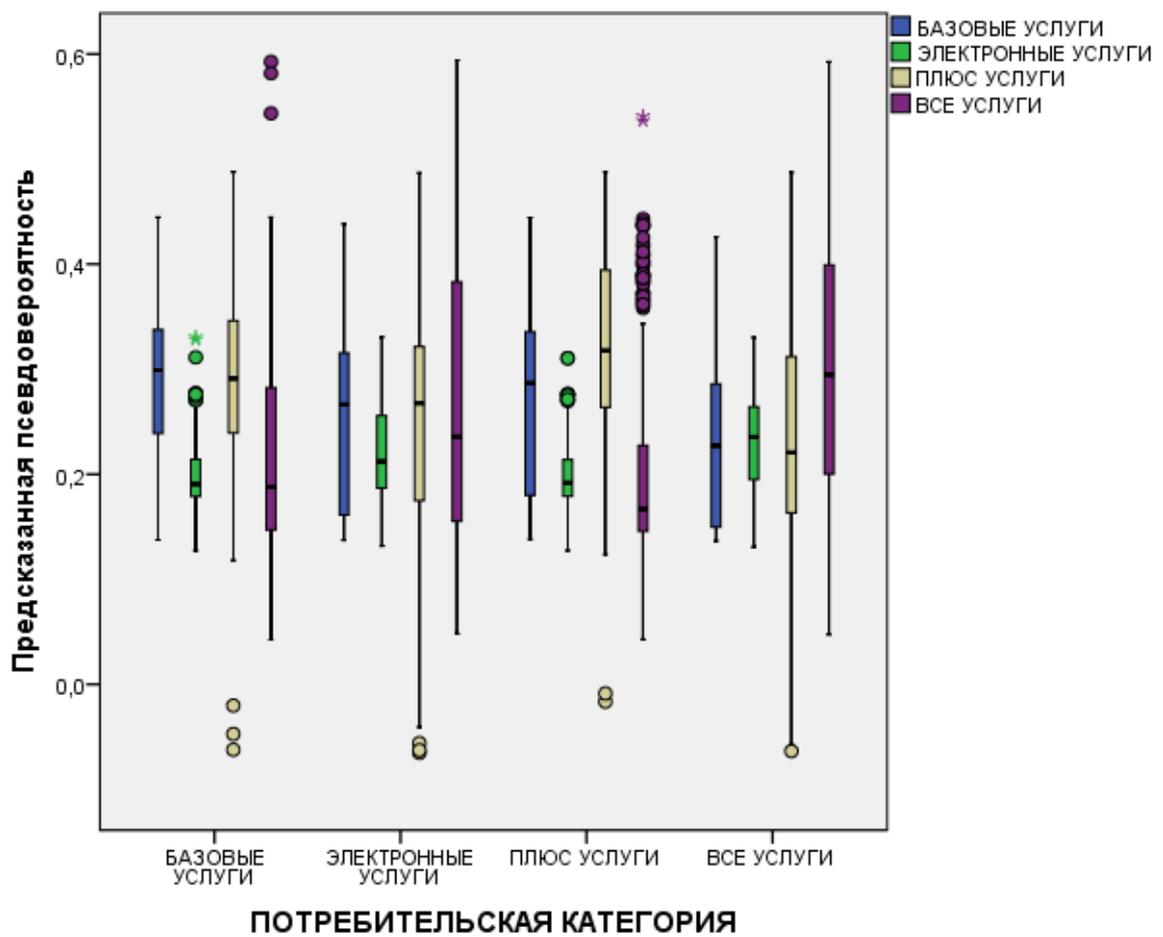
Зависимая переменная: ПОТРЕБИТЕЛЬСКАЯ КАТЕГОРИЯ

а. Количество скрытых единиц определяется по критерию проверки данных: «лучшее» количество скрытых единиц — это количество, при котором ошибка в тестовых данных минимальна.

Классификация

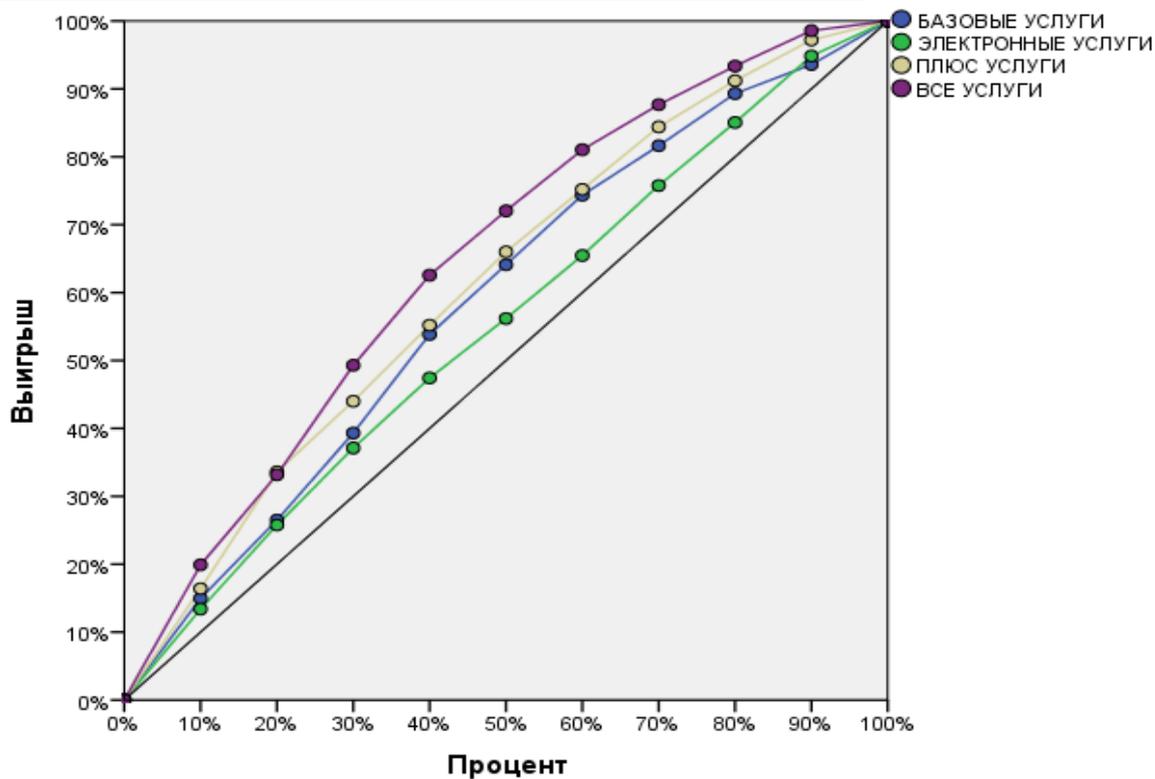
Выборка	Наблюденные	Предсказанные				Процент корректных
		БАЗОВЫЕ УСЛУГИ	ЭЛЕКТРОННЫЕ УСЛУГИ	ПЛЮС УСЛУГИ	ВСЕ УСЛУГИ	
Обучающая	БАЗОВЫЕ УСЛУГИ	65	0	69	41	37,1%
	ЭЛЕКТРОННЫЕ УСЛУГИ	38	0	44	59	0,0%
	ПЛЮС УСЛУГИ	51	0	99	35	53,5%
	ВСЕ УСЛУГИ	32	0	43	89	54,3%
	Общий процент	28,0%	0,0%	38,3%	33,7%	38,0%
Контрольная	БАЗОВЫЕ УСЛУГИ	18	0	28	13	30,5%
	ЭЛЕКТРОННЫЕ УСЛУГИ	18	0	13	22	0,0%
	ПЛЮС УСЛУГИ	18	0	32	15	49,2%
	ВСЕ УСЛУГИ	4	0	14	29	61,7%
	Общий процент	25,9%	0,0%	38,8%	35,3%	35,3%
Проверочная	БАЗОВЫЕ УСЛУГИ	12	0	12	8	37,5%
	ЭЛЕКТРОННЫЕ УСЛУГИ	6	0	8	9	0,0%
	ПЛЮС УСЛУГИ	13	0	14	4	45,2%
	ВСЕ УСЛУГИ	4	0	4	17	68,0%
	Общий процент	31,5%	0,0%	34,2%	34,2%	38,7%

Зависимая переменная: ПОТРЕБИТЕЛЬСКАЯ КАТЕГОРИЯ

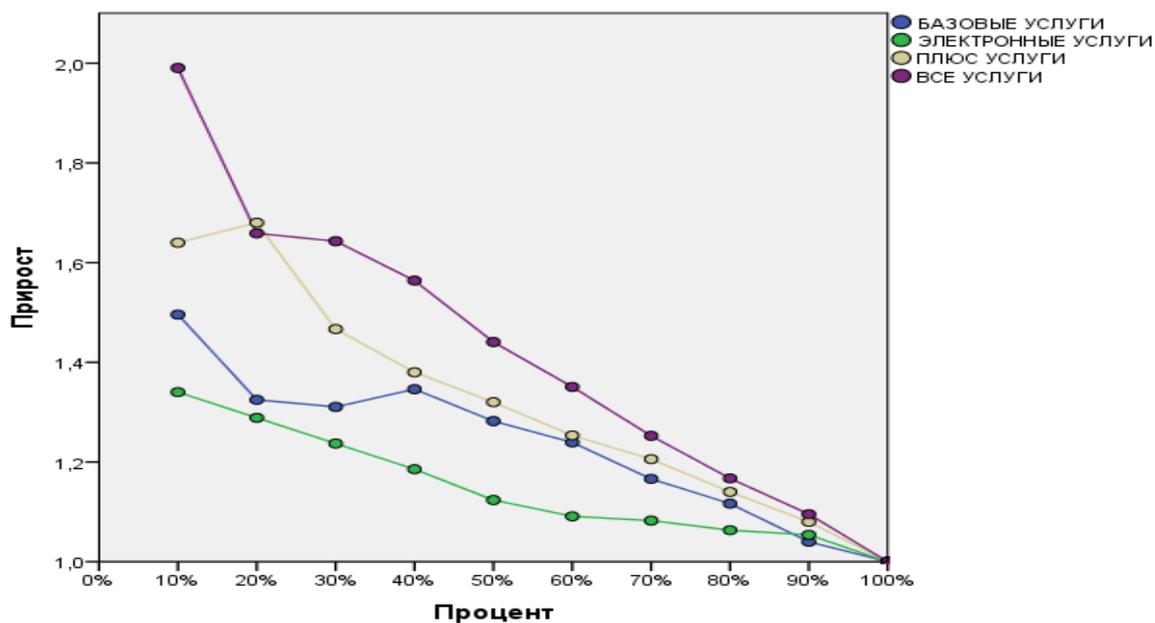


Площадь под кривой

		Площадь
ПОТРЕБИТЕЛЬСКАЯ КАТЕГОРИЯ	БАЗОВЫЕ УСЛУГИ	,622
	ЭЛЕКТРОННЫЕ УСЛУГИ	,565
	ПЛЮС УСЛУГИ	,659
	ВСЕ УСЛУГИ	,696



Зависимая переменная: ПОТРЕБИТЕЛЬСКАЯ КАТЕГОРИЯ



Зависимая переменная: ПОТРЕБИТЕЛЬСКАЯ КАТЕГОРИЯ

Программа исследования и данные.

```
GET
  FILE='C:\Program Files\IBM\SPSS\Statistics\20\Samples-Russian\telco.sav'.
DATASET NAME Наборданных1 WINDOW=FRONT.
APPLY DICTIONARY
  /FROM 'C:\Program Files\IBM\SPSS\Statistics\20\Samples-Russian\telco.sav'
  /SOURCE VARIABLES=region tenure age marital address income ed employ retire
gender reside tollfree equip callcard wireless longmon tollmon equipmon
cardmon wiremon longten tollten equipten cardten wireten multiline voice pager
internet callid callwait
forward confer ebill loglong logtoll logequi logcard logwire lninc custcat
churn
  /FILEINFO
  /VARINFO ALIGNMENT FORMATS LEVEL ROLE MISSING VALLABELS=REPLACE
ATTRIBUTES=REPLACE VARLABEL WIDTH.

SAVE TRANSLATE OUTFILE='C:\2012k\Neuronet\Лекции 2\telco.dat'
  /TYPE=TAB
  /MAP
  /REPLACE
  /FIELDNAMES
  /CELLS=VALUES.
```

Data written to C:\2012k\Neuronet\Лекции 2\telco.dat.
42 variables and 1000 cases written.

Variable: region	Type: Number	Width: 4	Dec: 0
Variable: tenure	Type: Number	Width: 4	Dec: 0
Variable: age	Type: Number	Width: 4	Dec: 0
Variable: marital	Type: Number	Width: 4	Dec: 0
Variable: address	Type: Number	Width: 4	Dec: 0
Variable: income	Type: Number	Width: 8	Dec: 2
Variable: ed	Type: Number	Width: 4	Dec: 0
Variable: employ	Type: Number	Width: 4	Dec: 0
Variable: retire	Type: Number	Width: 8	Dec: 2
Variable: gender	Type: Number	Width: 4	Dec: 0
Variable: reside	Type: Number	Width: 4	Dec: 0
Variable: tollfree	Type: Number	Width: 8	Dec: 2
Variable: equip	Type: Number	Width: 8	Dec: 2
Variable: callcard	Type: Number	Width: 8	Dec: 2
Variable: wireless	Type: Number	Width: 8	Dec: 2
Variable: longmon	Type: Number	Width: 8	Dec: 2
Variable: tollmon	Type: Number	Width: 8	Dec: 2
Variable: equipmon	Type: Number	Width: 8	Dec: 2
Variable: cardmon	Type: Number	Width: 8	Dec: 2
Variable: wiremon	Type: Number	Width: 8	Dec: 2
Variable: longten	Type: Number	Width: 8	Dec: 2
Variable: tollten	Type: Number	Width: 8	Dec: 2
Variable: equipten	Type: Number	Width: 8	Dec: 2
Variable: cardten	Type: Number	Width: 8	Dec: 2
Variable: wireten	Type: Number	Width: 8	Dec: 2
Variable: multiline	Type: Number	Width: 4	Dec: 0
Variable: voice	Type: Number	Width: 4	Dec: 0
Variable: pager	Type: Number	Width: 4	Dec: 0
Variable: internet	Type: Number	Width: 4	Dec: 0
Variable: callid	Type: Number	Width: 4	Dec: 0
Variable: callwait	Type: Number	Width: 4	Dec: 0
Variable: forward	Type: Number	Width: 4	Dec: 0
Variable: confer	Type: Number	Width: 4	Dec: 0
Variable: ebill	Type: Number	Width: 4	Dec: 0
Variable: loglong	Type: Number	Width: 8	Dec: 2
Variable: logtoll	Type: Number	Width: 8	Dec: 2
Variable: logequi	Type: Number	Width: 8	Dec: 2

Variable: logcard	Type: Number	Width: 8	Dec: 2
Variable: logwire	Type: Number	Width: 8	Dec: 2
Variable: lninc	Type: Number	Width: 8	Dec: 2
Variable: custcat	Type: Number	Width: 8	Dec: 0
Variable: churn	Type: Number	Width: 4	Dec: 0

Приложение 1. Описание пакета “BrainMaker 3.11 Pro”

Описание пакета составлено на основании [2, 3].

Данный нейропакет предназначен для создания прикладных нейронных сетей (НС). Он использует парадигму НС с обратным распространением ошибки.

Состав пакета BrainMaker следующий:

- NetMaker – анализ данных и подготовка обучающей выборки;
- BrainMaker – обучение НС, анализ обученной сети, работа с обученной сетью;
- Utilities – анализ файлов пакета и другой сервис.

Программа “NetMaker”

Данная программа предназначена для подготовки обучающей выборки. Она предоставляет широкие возможности по статистическому анализу данных, т.е. пакет BrainMaker 3.11 Pro, в который включена программа NetMaker, предоставляет не только нейросетевые методы анализа и обработки данных, но и классический метод – статистический. Это делает данный пакет мощным средством для обработки данных.

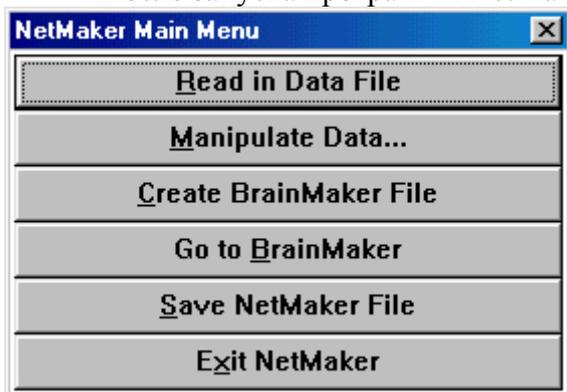
NetMaker может работать с такими форматами, как Excel, Lotus, dBase, а также некоторыми другими текстовыми и двоичными форматами файлов.

Данная программа обладает следующими развитыми средствами предварительной обработки и подготовки данных:

- статистические функции – помощь в предварительном анализе входных данных;
- операция хронологического сдвига – специальная подготовка данных для анализа изменения в данных во времени;
- арифметические операции с колонками данных;
- анализ диапазонов данных – помощь в поиске данных, выбивающихся из общей закономерности;
- циклический анализ – нахождение периодических закономерностей во входных данных;
- анализ корреляции данных – помощь в обнаружении взаимосвязей между входными переменными и наличия временного сдвига между ними.

Программа NetMaker приводится в активное состояние путем запуска исполняемого файла **netmaker.exe**.

После запуска программы NetMaker пользователь видит главное меню (рис.1).



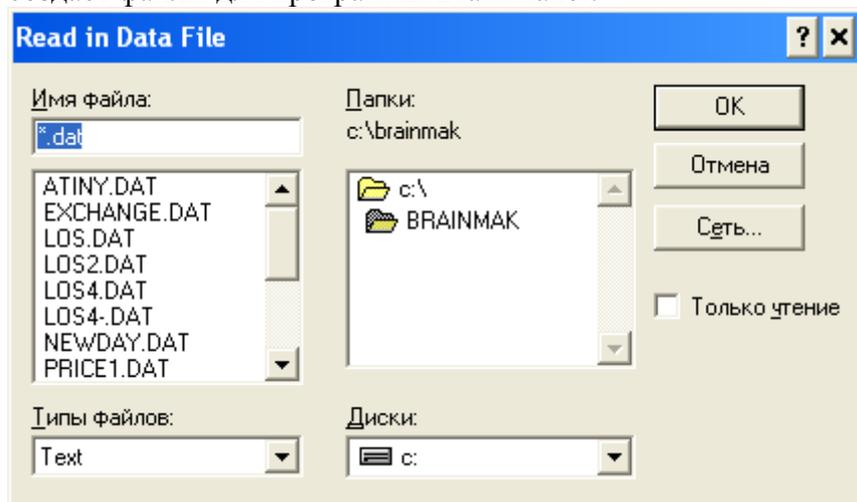
Окно главного меню программы NetMaker

Функции, выполняемые при нажатии на соответствующие кнопки главного меню следующие:

Read in Data File – выводится диалог открытия файла, в котором предоставляется возможность выбора типа загружаемого файла (Excel, Lotus, dBase, текстовый или двоичный). После этого файл данных, выбранный пользователем, загружается.

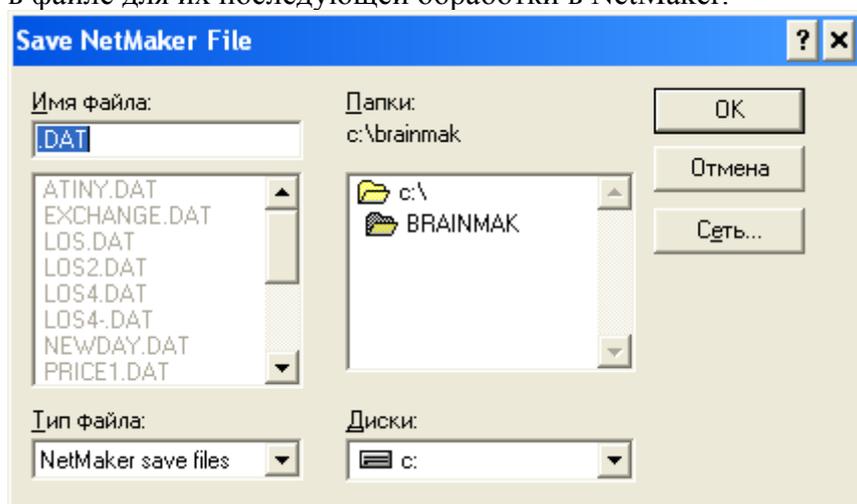
Manipulate Data... - активизирует основное окно программы и закрывает данный диалог.

Create BrainMaker File – на основе подготовленных пользователем данных создает файлы для программы BrainMaker.



Go to BrainMaker – закрывает программу NetMaker и загружает программу BrainMaker.

Save NetMaker File – позволяет сохранить подготовленные пользователем данные в файле для их последующей обработки в NetMaker.



Exit NetMaker – закрывает данную программу.

После того как пользователь загрузил файл с данными (**Read in Data File**) и нажал кнопку **Manipulate Data...**, появляется главное окно программы NetMaker (рис. 2).

	NotUsed	NotUsed	NotUsed	NotUsed	NotUsed
	Sales	Costs	Dow	Volume	IntRate
1	11.6	4.3	2500	60	9.75
2	10.1	4.3	2535	67.5	9.75
3	10.6	4.3	2514	58	9.75
4	8.3	4.3	2523	70.5	9.75
5	10.1	4.3	2527	54	9.75

Окно программы NetMaker

В данном окне пользователь видит данные в табличном виде. Верхняя текстовая строка содержит информацию о типе колонки – входная (**Input**), образ для обучения (**Pattern**), не используемая (**Not Used**) и аннотация (**Annote**)– не используется, но выводится в окно программы BrainMaker.

Находящаяся ниже строка с кнопками - это заголовок столбцов.

Кнопки, находящиеся в окне слева, – заголовки строк.

В левом верхнем углу окна программы при необходимости выводятся строки – **Select column** или **Select row** – они говорят пользователю о необходимости выбрать столбец или строку для выполнения выбранной им операции.

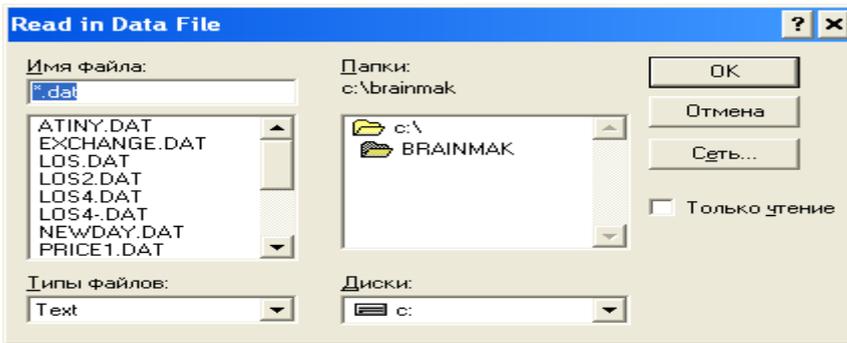
Операции над строками и столбцами, которые пользователь может использовать, представлены пунктами соответствующих меню. Далее приведено краткое описание пунктов меню и функций, которые они выполняют.

Меню File

File	Column	Row	Label	Number	Symbol	Operate	Indicators
Create BrainMaker Files...						Ctrl+B	NotUsed NotUse
Create Running Fact File...							Sales FoodSt
Read In Data File...						Ctrl+F	90 113
Import Data from Clipboard							91 115
Save NetMaker File...						Ctrl+S	90 116.5
Preferences...							90 115.8
Transpose Rows & Columns							90 116.7
About NetMaker Professional...							91 117
Return to Main Menu							
Exit						Alt+X	

File → Create BrainMaker File... (Ctrl+B).

Позволяет сохранить подготовленную обучающую выборку в формате BrainMaker (файл с расширением .dat). После выбора данного пункта меню пользователь видит перед собой окно диалога, позволяющее задать имена файлов описания сети (расширение .def), данных для обучения НС (расширение .fct) и данных для тестирования обученной сети (расширение .tst).



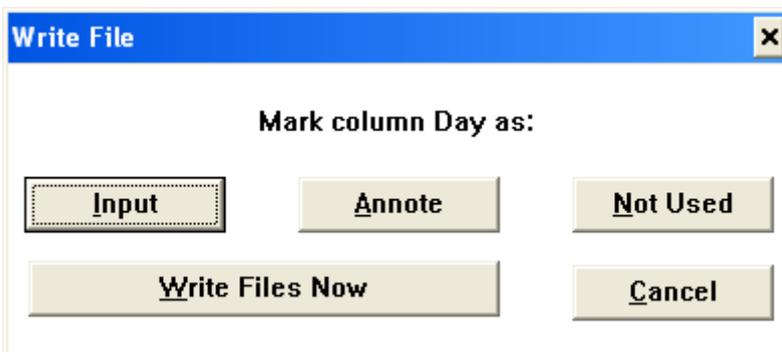
Сохраняется файл с расширением .dat:

NetMaker Professional - ATINY.DAT							
File	Column	Row	Label	Number	Symbol	Operate	Indicators
			Day	DOW	Interest	Sales	FoodStk
			NotUsed	NotUsed	NotUsed	NotUsed	NotUsed
1			1	3027	4.3	90	113
2			2	3034	4.2	91	115
3			3	3032	5.1	90	116.5
4			4	3039	5.2	90	115.8
5			5	3037	5.3	91	116.7
6			6	3044	5.4	91	117
7			7	3042	5.6	91	117.9
8			8	3048	5.2	92	117.7
9			9	3030	5.8	91	119
10			10	3033	5.4	90	116.2
11			11	3030	5.1	91	115.9
12			12	3028	4.6	92	116.1
13			13	3031	4.1	92	116.3
14			14	3032	4.3	92	116.6

Если какие-либо колонки данных помечены как не используемые (Not Used), то программа предложит их пометить.

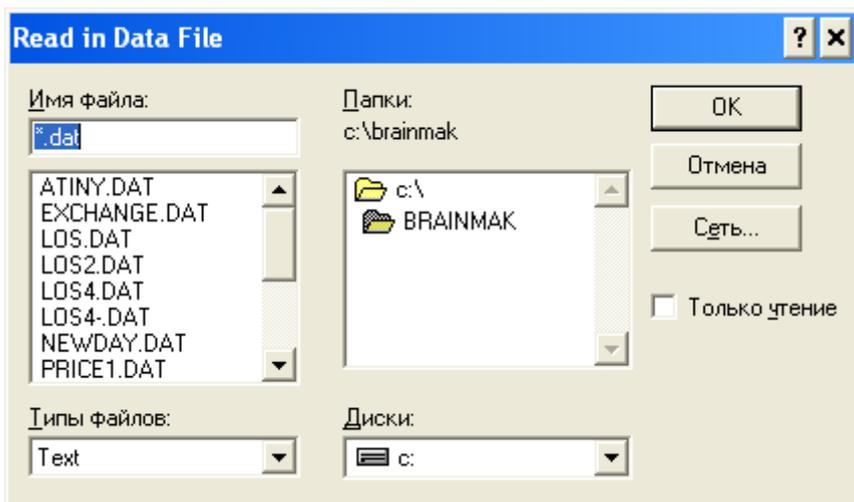
File → Create Running Fact File...

Позволяет пользователю создавать файлы в формате BrainMaker для их обработки обученной нейронной сетью. После выбора данного пункта меню пользователь видит окно диалога сохранения файла, в котором имеется возможность ввода необходимого имени файла.



File → Read in Data File

После выбора данного пункта меню выводится окно диалога открытия файла, в котором предоставляется возможность выбора типа загружаемого файла (Excel, Lotus, dBase, текстовый или двоичный).



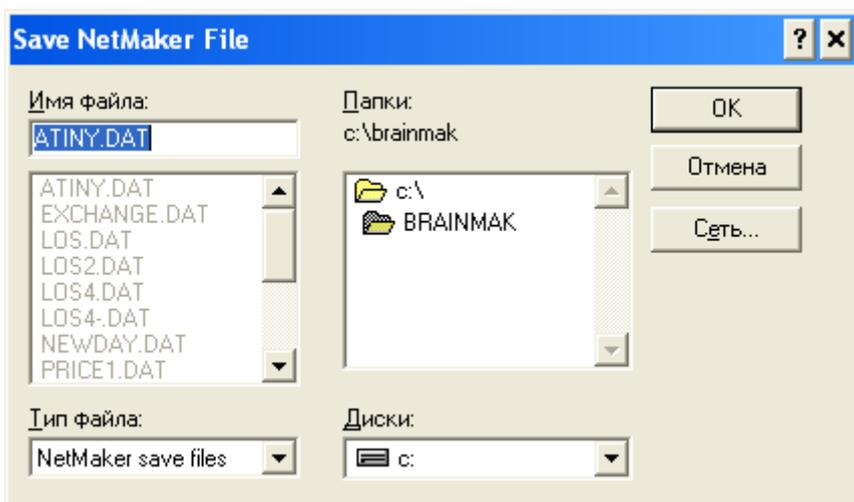
После этого файл данных, выбранный пользователем, загружается.

File → Import Data from Clipboard

Позволяет использовать данные из других программ. Данные копируются в clipboard и добавляются к обрабатываемым данным.

File → Save NetMaker File...

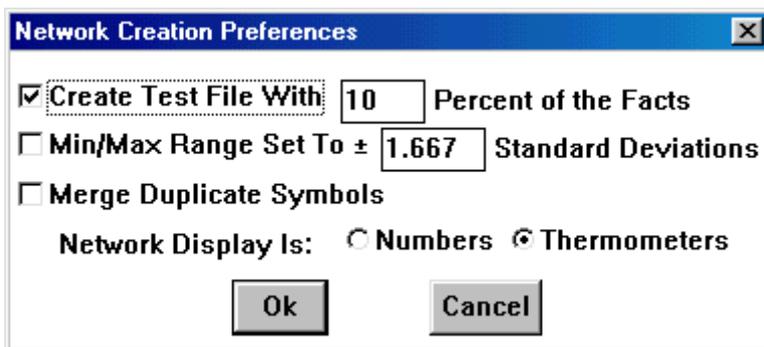
Позволяет сохранить подготовленные данные в файле для их последующей обработки в NetMaker. После выбора данного пункта меню пользователь видит окно диалога сохранения файла, в котором имеет возможность ввести необходимое имя файла.



File → Preferences...

Позволяет задавать опции для создания файла для BrainMaker.

После выбора данного пункта меню пользователь видит следующее окно диалога.



Окно диалога настройки файла BrainMaker

Create Test File With ... Percent of the Facts – позволяет задать в процентах количество данных из обрабатываемых, которые пользователь желает использовать для тестирования обученной нейронной сети.

Min/Max Range Set To ± ... Standard Deviations - позволяет задать минимальное и максимальное значение для данных в виде коэффициента к стандартному отклонению.

Merge Duplicate Symbols – позволяет соединять одинаковые символы.

Network Display Is – вывод на экран BrainMaker данных в виде гистограмм или чисел.

File → Transpose Rows & Columns

Выбор данного пункта меню позволяет транспонировать таблицу с данными, т.е. поменять местами столбцы и строки.

NetMaker Professional - ATINY.DAT																		
File	Column	Row	Label	Number	Symbol	Operate	Indicators										NotUsed	NotUs
NotUsed																		
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O			
1		2	3	4	5	6	7	8	9	10	11	12	13	14	15			
2	3027	3034	3032	3039	3037	3044	3042	3048	3030	3033	3030	3028	3031	3032	3049			
3	4.3	4.2	5.1	5.2	5.3	5.4	5.6	5.2	5.8	5.4	5.1	4.6	4.1	4.3	4.6			
4	90	91	90	90	91	91	91	92	91	90	91	92	92	92	93			
5	113	115	116.5	115.8	116.7	117	117.9	117.7	119	116.2	115.9	116.1	116.3	116.6	116.8			

File → About NetMaket...

Выводит на экран диалог с информацией о программе.

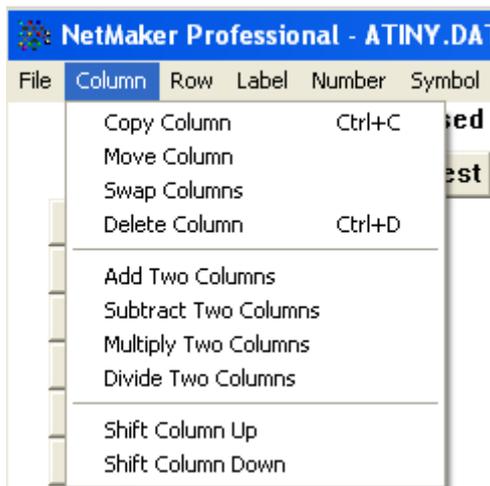
File → Return To Main Menu

Выводит главное меню программы на экран.

File → Exit (Alt+X)

Завершает работу программы.

Меню Column



Column → Copy Column (Ctrl+C)

Копирует данные из данного столбца в новую колонку. После выбора данного пункта на экране появляется окно диалога, в котором пользователь вводит название новой колонки. Новая колонка добавляется в конец таблицы.

Column → Move Column

Перемещает колонки. После выбора данного пункта меню пользователю предлагается первоначально выбрать первую колонку (появляется сообщение в верхнем левом углу окна программы),

NetMaker Professional - ATINY.DAT

File Column Row Label Number Symbol Operate Indicators

Select NotUsed NotUsed NotUsed NotUsed NotUsed

1st col	Day	DOW	Interest	Sales	FoodStk
1		3027	4.3	90	113
2	2	3034	4.2	91	115

а после этого выбрать вторую колонку для обмена.

NetMaker Professional - ATINY.DAT

File Column Row Label Number Symbol Operate Indicators

Select NotUsed NotUsed NotUsed NotUsed NotUsed

2nd col	Day	DOW	Interest	Sales	FoodStk
1	1	3027	4.3	90	113
2	2	3034	4.2	91	115

Результат перемещения: колонки Day и Dow перемещены в правую часть таблицы.

NetMaker Professional - ATINY.DAT							
File	Column	Row	Label	Number	Symbol	Operate	Indicators
NotUsed NotUsed NotUsed NotUsed NotUsed							
	Interest	Sales	FoodStk	Day	DOW		
1	4.3	90	113	1	3027		
2	4.2	91	115	2	3034		
3	5.1	90	116.5	3	3032		
4	5.2	90	115.8	4	3039		
5	5.3	91	116.7	5	3037		

Column → Swap Column

Меняет две колонки местами.

Column → Delete Column (Ctrl+D)

Удаление колонки с данными. Если пользователь нажмет кнопку ОК в появившемся окне диалога, то столбец удаляется.

Column → Add Two Column

Позволяет складывать данные двух столбцов. После выбора данного пункта меню пользователю предлагается выбрать первую колонку (появляется сообщение в верхнем левом углу окна программы). Далее необходимо выбрать вторую колонку, для того чтобы выполнить данную операцию. В появившемся затем окне диалога пользователь вводит название результирующей колонки. Новая колонка добавляется в конец таблицы.

Column → Subtract Two Column

Позволяет вычитать значения данных одной колонки из значений данных другой. Действия пользователя после выбора данного меню аналогичны пункту меню **Column → Add Two Column**.

Column → Multiply Two Column

Позволяет умножать значения данных одной колонки на значения данных другой. Действия пользователя после выбора данного меню аналогичны пункту меню **Column → Add Two Column**.

Column → Divide Two Column

Позволяет делить значения данных одной колонки на значения данных другой. Действия пользователя после выбора данного меню аналогичны пункту меню **Column → Add Two Column**.

Column → Shift Column Up

Сдвигает строки данных вверх, при этом верхние строки уничтожаются, а нижние заполняются значением самой последней строки данного столбца. После выбора данного пункта меню пользователю предлагается выбрать колонку (появляется сообщение **Select Column** в верхнем левом углу окна программы) для ее сдвига. Далее на экран выводится окно диалога для ввода количества строк, на которое надо сдвинуть данный столбец с данными. После на экран выводится окно диалога для ввода названия нового столбца. Новый столбец, добавляется в конец таблицы.

Column → Shift Column Down

Сдвигает строки данных вниз, при этом нижние строки уничтожаются, а верхние заполняются значением самой первой строки данного столбца. Действия пользователя после выбора данного меню аналогичны, описанным в пункте меню **Column → Shift Column Up**.

Меню Row



Row → Copy Row

Позволяет копировать строку данных. После выбора данного пункта меню пользователю предлагается выбрать строку (появляется сообщение) для ее копирования. Копируемая строка появляется ниже выбранной, нижележащие строки сдвигаются вниз.

Row → Shuffle Row

Располагает строки в случайном порядке.

Row → Make Min/Max Row

Выводит на экран строки с минимальными и максимальными значениями данных для соответствующих столбцов. Для отключения этой опции достаточно еще раз выбрать данный пункт меню. Строки min и max располагаются в верхней части таблицы:

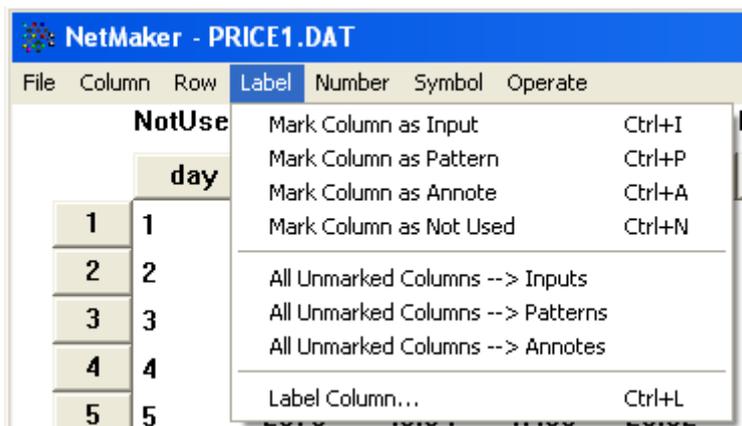
The screenshot shows the NetMaker Professional interface for a file named ATINY.DAT. The menu bar includes File, Column, Row, Label, Number, Symbol, Operate, and Indicators. The table below has columns labeled 'Label', 'Number', 'Symbol', 'Operate', and 'Indicators'. The first row is labeled 'Min' and the second row is labeled 'Max'. The 'Number' column contains values 3.1, 5.8, 4.3, 4.2, 5.1, and 5.2. The 'Symbol' column contains values 90, 98, 90, 91, 90, and 90. The 'Operate' column contains values 113, 125.6, 113, 115, 116.5, and 115.8. The 'Indicators' column contains values 1, 30, 1, 2, 3, and 4. The 'Label' column contains values 3027, 3078, 3027, 3034, 3032, and 3039.

Label	Number	Symbol	Operate	Indicators	
Min	3.1	90	113	1	3027
Max	5.8	98	125.6	30	3078
1	4.3	90	113	1	3027
2	4.2	91	115	2	3034
3	5.1	90	116.5	3	3032
4	5.2	90	115.8	4	3039

Row → Delete Row

Удаляет строку. Если пользователь нажмет кнопку ОК в появившемся окне диалога, то строка удаляется.

Меню Label



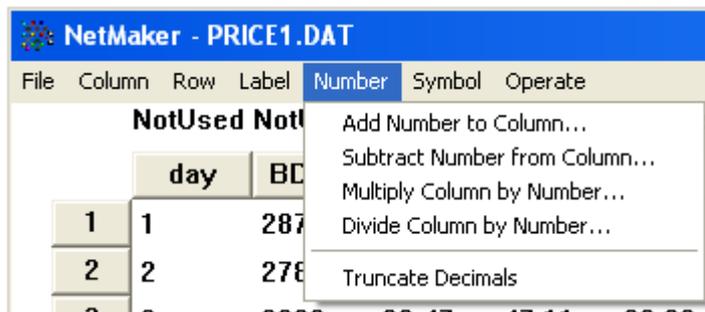
Label → Mark Column as Input (Pattern, Annote, Not Used)
Помечает столбец данных как содержащий: входные данные (Input); указание учителя для обучения (Pattern); аннотацию (Annote); не используемые и не выводимые на экран (Not Used). Значения строк столбца, помеченного как аннотация, просто выводятся на экран, но не используется для обучения НС. После выбора данного пункта меню пользователю предлагается выбрать колонку для обозначения.

Select	NotUsed	NotUsed	NotUsed	NotUsed	NotUsed
column	Interest	Sales	FoodStk	Day	DOW
Min	3.1	90	113	1	3027
Max	5.8	98	125.6	30	3078
1	4.3	90	113	1	3027
2	4.2	91	115	2	3034

Label → All Unmarked Columns → Input (Pattern, Annote)

Помечает все колонки, обозначенные Not Used как: колонки входных данных (Input), колонки с указанием учителя для обучения (Pattern) или как колонки аннотаций (Annote) – данные колонки просто выводятся на экран, но не используются для обучения.

Меню Number



Number → Add Number To Column...

Добавляет ко всем числовым значениям строк данного столбца необходимое число. После выбора данного пункта меню пользователь выбирает колонку для выполнения данной операции. Далее на экран выводится окно диалога для ввода значения, которое

необходимо добавить к столбцу. После этого появляется окно диалога для ввода названия столбца результата. Колонка результата добавляется в конец таблицы.

Number → Subtract Number From Column...

Вычитает из всех числовых значений строк данного столбца необходимое число. Действия пользователя после выбора данного меню аналогичны, описанным в пункте меню **Number → Add Number To Column...**

Number → Multiply Column by Number...

Умножает все числовые значения строк данного столбца на необходимое число. Действия пользователя после выбора данного меню аналогичны, описанным в пункте меню **Number → Add Number To Column...**

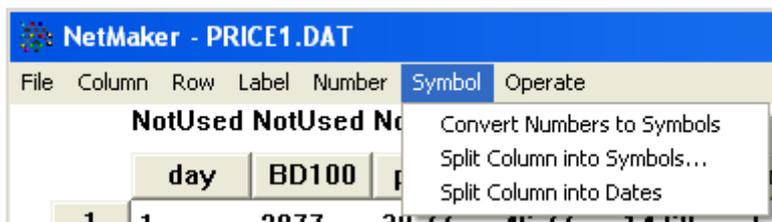
Number → Divide Column by Number...

Делит все числовые значения строк данного столбца на необходимое число. Действия пользователя после выбора данного меню аналогичны, описанным в пункте меню **Number → Add Number To Column...**

Number → Truncate Decimals

Округляет значения строк данного столбца. После выбора данного пункта меню пользователю предлагается выбрать колонку (появляется сообщение в верхнем левом углу окна программы) для выполнения данной операции. Далее появляется окно диалога для ввода названия столбца результата. Он добавляется в конец таблицы.

Меню Symbol



Symbol → Convert Numbers To Symbol

Преобразует числовые значения строк столбца в символы. После выбора данного пункта меню пользователю предлагается выбрать колонку (появляется сообщение в верхнем левом углу окна программы) для выполнения данной операции. Далее появляется окно диалога для ввода нового столбца. Новый столбец добавляется в конец таблицы.

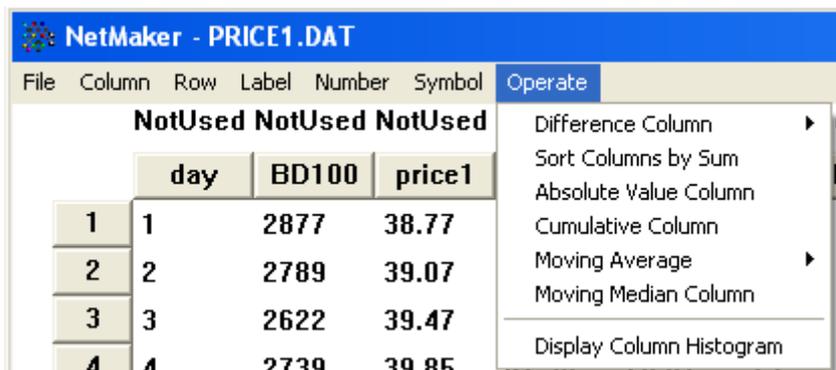
Symbol → Split Column into Symbols...

Преобразование числовых значений строк столбца в символы. Весь интервал данных столбца разбивается на необходимое количество подинтервалов. Каждому подинтервалу присваивается строка символов. После выбора данного пункта меню пользователю предлагается выбрать колонку (появляется сообщение в верхнем левом углу окна программы) для выполнения данной операции. Далее на экран выводится окно диалога с выбором базового символа, количества подинтервалов и интервала. После появляется диалог для ввода названия результирующего столбца. Новая колонка добавляется в конец таблицы.

Symbol → Split Column into Dates

Преобразует столбец в столбцы со значениями, соответствующими месяцу, году, дню недели. Новые колонки добавляются в конец таблицы.

Меню Operate



Operate → Difference Column → Simple

Создает новую колонку с числовыми значениями, которые являются разностью числовых значений исходной колонки и числовых значений этой же колонки, но сдвинутой на необходимое количество строк. После выбора данного пункта пользователю необходимо ввести число, равное числу сдвигаемых строк. Затем появляется окно диалога для ввода названия результирующего столбца. Новый столбец добавляется в конец таблицы.

Operate → Difference Column → Percent

Создает новую колонку, которая является процентной разницей числовых значений исходной колонки и числовых значений этой же колонки, но сдвинутой на необходимое количество строк. Действия пользователя после выбора данного меню аналогичны, описанным в пункте меню **Operate → Difference Column → Simple**.

Operate → Difference Column → Log

Создает новую колонку числовых значений, которая является логарифмической разницей числовых значений исходной колонки и числовых значений этой же колонки, но сдвинутой на необходимое количество строк. Действия пользователя после выбора данного меню аналогичны, описанным в пункте меню **Operate → Difference Column → Simple**.

Operate → Sort Columns by Sum

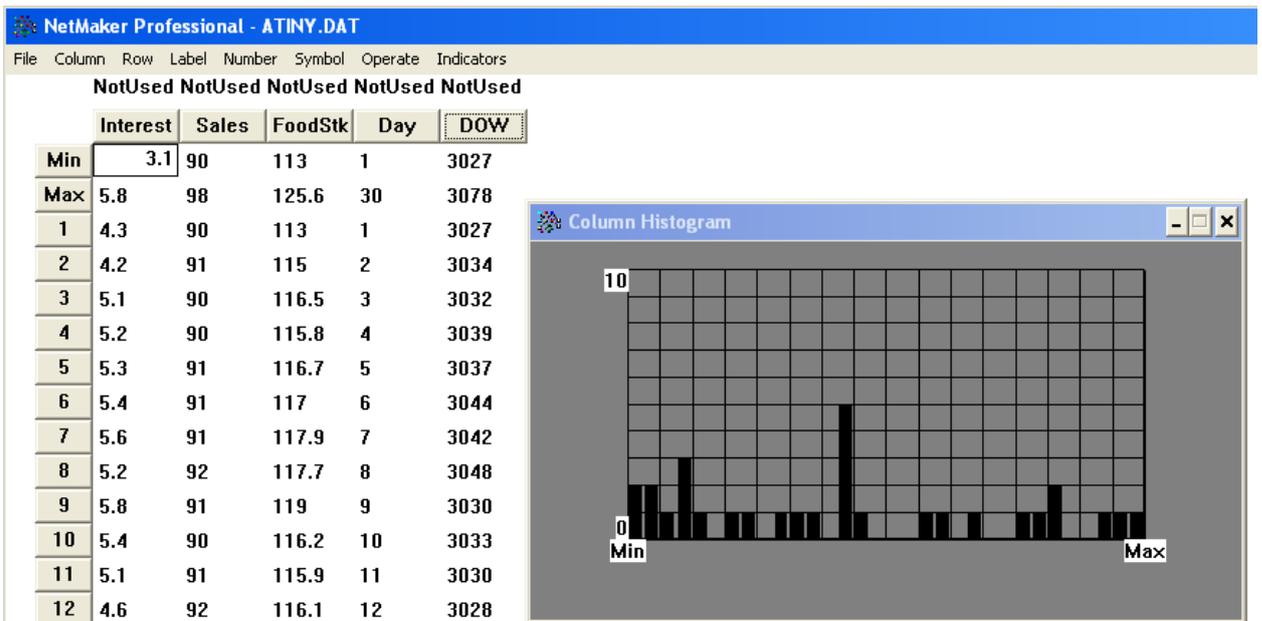
Сортирует порядок расположения столбцов данных в зависимости от значения суммы значений всех строк столбца.

Operate → Absolute Value Column

Создает столбец с абсолютными числовыми значениями строк исходного столбца. После выбора данного пункта меню в появившемся окне необходимо ввести название нового столбца. Новая колонка добавляется в конец таблицы.

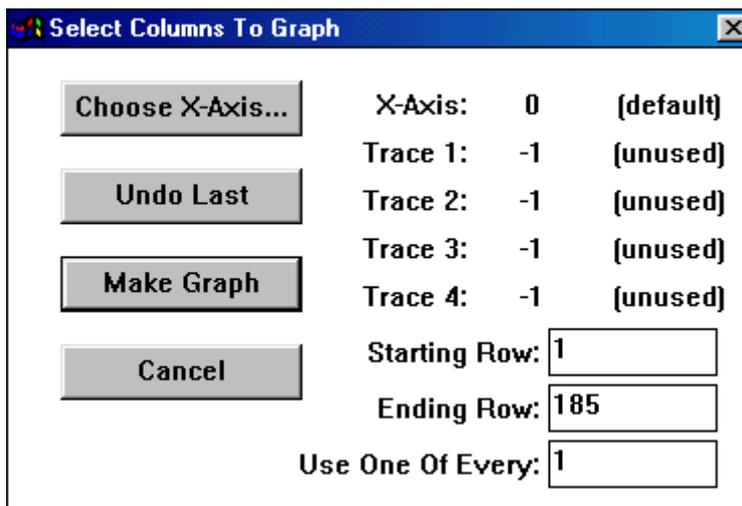
Operate → Display Column Histogram

Позволяет вывести на экран диаграмму распределения значений строк исходного столбца. Эта диаграмма показывает распределение по количеству строк с одинаковым значением.



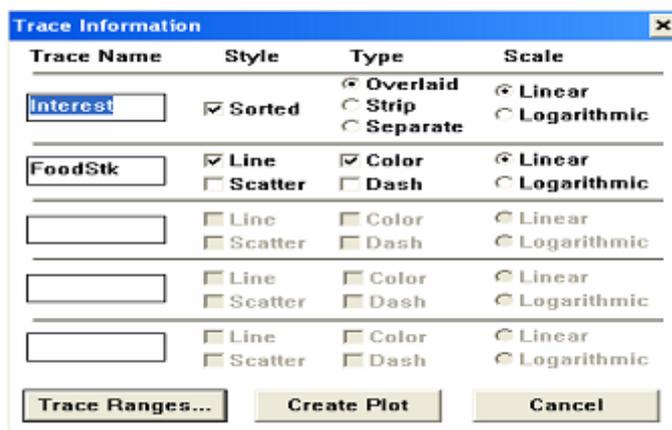
Operate → Graph Column...

Выводит на экран окно графика числовых значений строк выбранных столбцов. После выбора данного пункта меню появляется окно диалога.

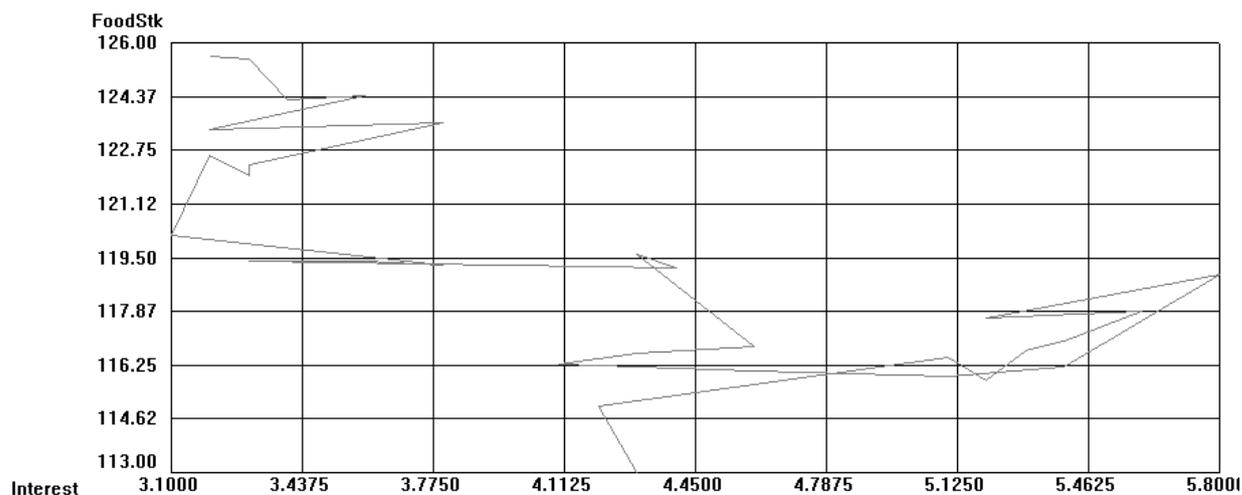


Окно диалога выбора столбцов для отображения графика

Необходимо нажать кнопку **Choose X-Axis...** для выбора столбца, определяющего ось X. Далее выберите столбцы для представления оси Y. Данных столбцов и соответственно графиков может быть не более 4.



Затем нажмите кнопку **Make Graph** и далее в новом окне **Create Plot**.



Operate → Repeat...

Позволяет выполнить следующую за данной операцией. В появившемся окне диалога введите необходимое количество повторений – больше 1. Затем выберите необходимую операцию (кроме меню Label). Ее действие будет выполнено над следующими (количество задается в предыдущем диалоге) за выбранным в ходе выполнения необходимой операции столбцом (строкой).

Operate → Cyclic Analysis...

Анализ содержимого столбца на цикличность изменения числовых значения строк столбца. После выбора данного пункта меню в появившемся окне диалога необходимо ввести количество строк для периода данных, используемого при анализе. Нажмите кнопку **Analyze** и далее **Create Plot**.

Operate → Data Correlator...

Позволяет определить, насколько коррелируют числовые значения данных в колонках. После выбора данного пункта меню необходимо выбрать два столбца для анализа.

Программа “BrainMaker”

Данная программа нейропакета “BrainMaker 3.11 Pro” предназначена для обучения НС. При этом используется алгоритм обучения НС с обратным распространением ошибки. После обучения сети в BrainMaker имеется возможность эксплуатации такой сети в рабочем режиме.

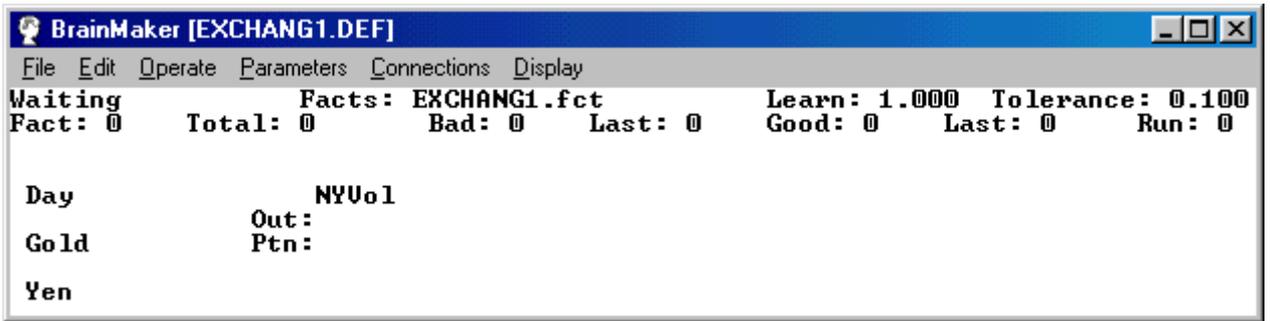
В программе BrainMaker имеются мощные средства для выполнения анализа обученной НС:

- Анализ чувствительности – определение степени влияния отдельных входов НС на ее ответ.
- Контурный анализ – графическая визуализация зависимости влияния пар входных переменных на ответ НС.

Запустить программу BrainMaker можно несколькими способами:

1. Запуск исполняемого файла **brainmak.exe**.
2. Переход из программы NetMaker в данную программу, используя пункт главного меню NetMaker **Go to BrainMaker**.

После запуска BrainMaker и загрузки файла с обучающей выборкой или с параметрами обученной сети пользователь видит следующее окно.



Главное окно программы BrainMaker

В левом верхнем углу BrainMaker сообщает пользователю о том, какую операцию он сейчас выполняет: **Waiting** – ожидает действий от пользователя (закончил обучение), **Training** – обучение сети, **Testing** – тестирование сети.

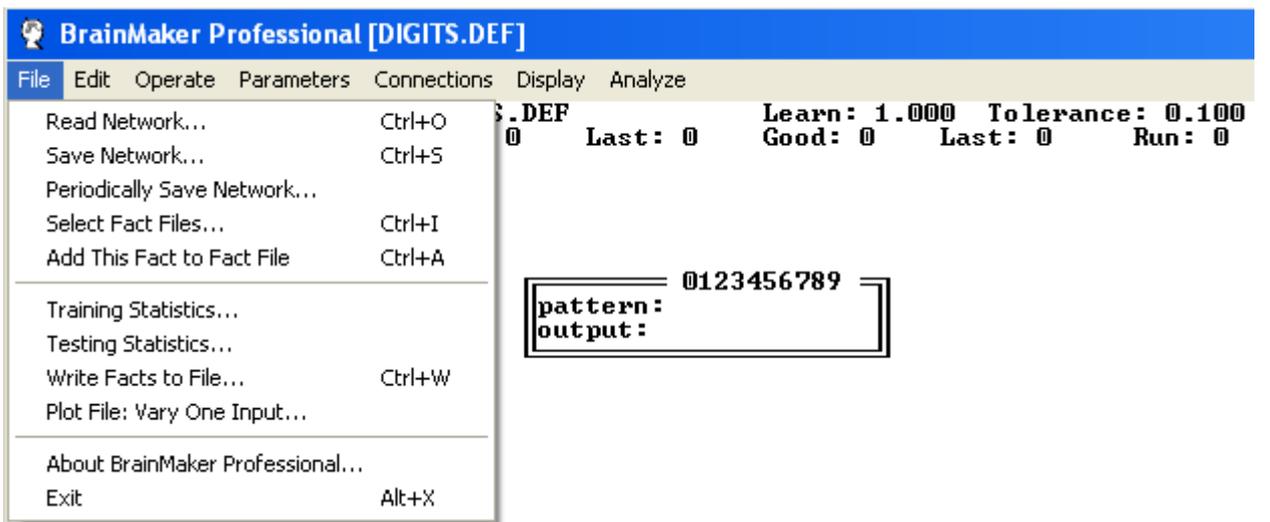
В двух верхних текстовых строках отображены значения следующих параметров:

- **Facts** – имя файла обучающей выборки.
- **Tolerance** – значение ошибки при обучении.
- **Fact** – номер текущего факта.
- **Total** – общее количество обработанных фактов.
- **Bad** – текущее количество неправильно распознанных фактов.
- **Last** – количество неправильно распознанных фактов при последнем проходе в процессе обучения НС.
- **Good** – текущее количество правильно распознанных фактов.
- **Last** – количество правильно распознанных фактов при последнем проходе в процессе обучения НС.
- **Run** – общее количество проходов НС.

Операции, которые пользователь может использовать, представлены пунктами соответствующих меню.

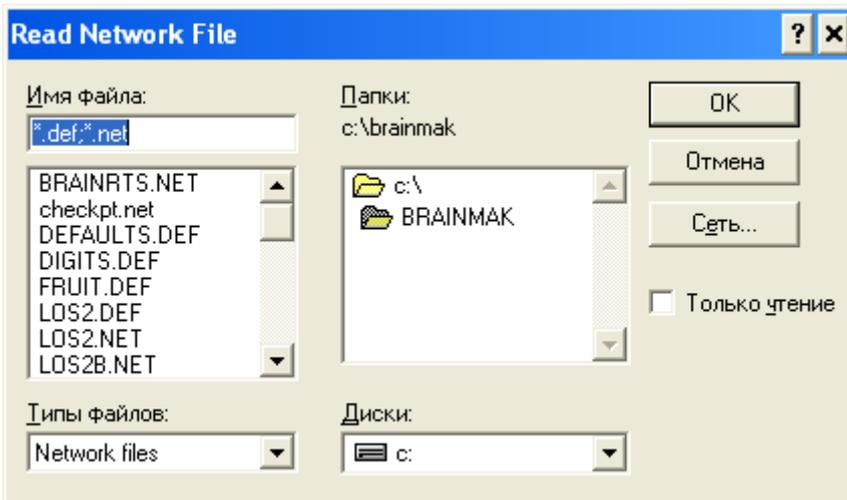
Далее приведено краткое описание пунктов меню и функций, которые они выполняют.

Меню File



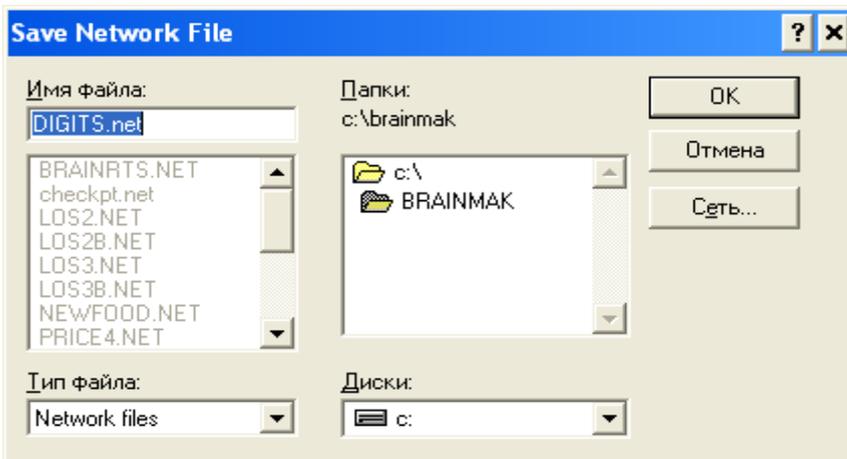
File → Read Network...

Позволяет загрузить файл с подготовленной в NetMaker обучающей выборкой или с параметрами модели НС, обученной в BrainMaker.



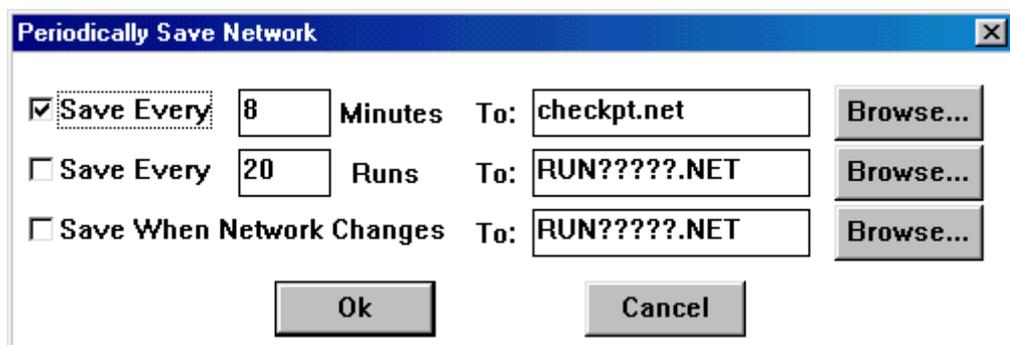
File → Save Network...

Позволяет сохранить параметры (веса, опции обучения) модели НС, обученной в BrainMaker.



File → Periodically Save Network...

Позволяет задать условия автоматического сохранения параметров НС в процессе ее обучения. После выбора данного пункта меню на экран выводится следующее окно диалога.



Окно диалога настройки периодического сохранения сети

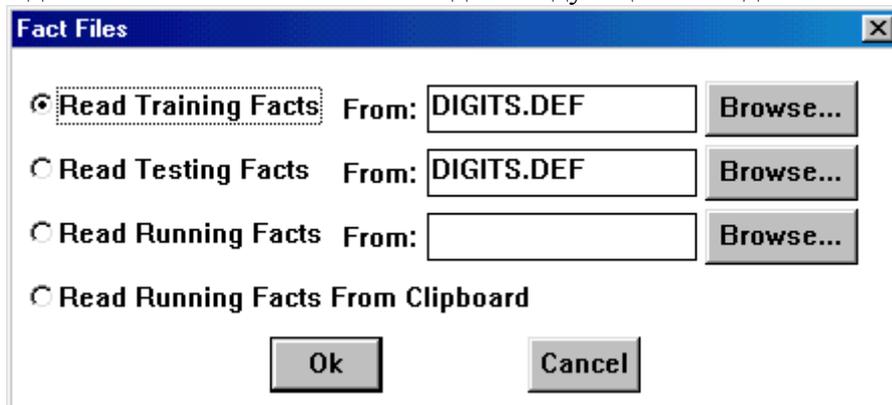
Опция **Save Every ... Minutes** – сохранение параметров обучаемой сети через необходимое количество минут.

Опция **Save Every ... Runs** – сохранение параметров обучаемой сети через необходимое количество проходов НС при обучении.

Опция **Save When Network Changes** – сохранение параметров обучаемой сети при любых изменениях.

File → Select Fact Files...

Позволяет выбрать файл с входными векторами (далее по тексту факты) обучающей выборки или фактами для их обработки обученной НС в рабочем режиме. После выбора данного меню пользователь видит следующее окно диалога.



Окно диалога выбора файла фактов

Опция **Read Training Facts** – обучающие факты из файла подготовленной ОБ.

Опция **Read Testing Facts** – тестовые факты из файла подготовленной ОБ.

Опция **Read Running Facts** – выбор файла данных, подготовленных для их обработки обученной НС в рабочем режиме.

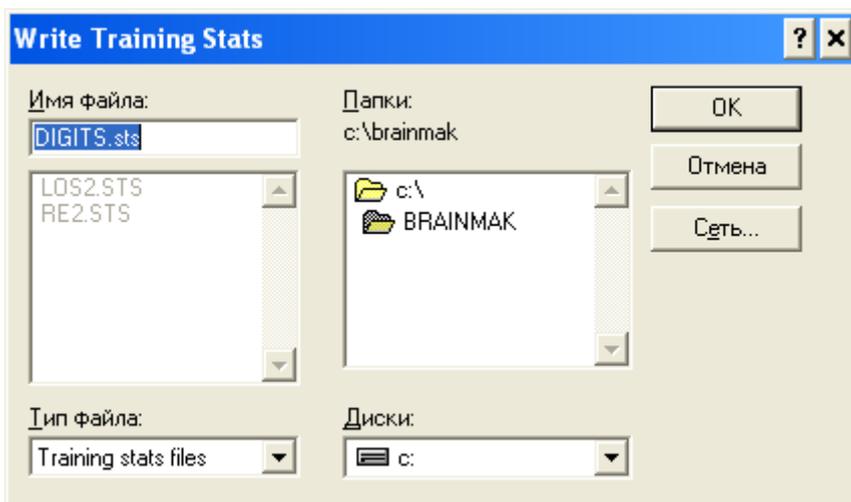
Опция **Read Running Facts From Clipboard** – факты из буфера обмена.

File → Add This Fact to Fact File

Добавляет к файлу фактов текущий факт, видимый в данный момент на экране. Так как BrainMaker позволяет редактировать с использованием мыши факты, представленные в его окне, то пользователь имеет возможность создания или редактирования обучающей выборки средствами BrainMaker.

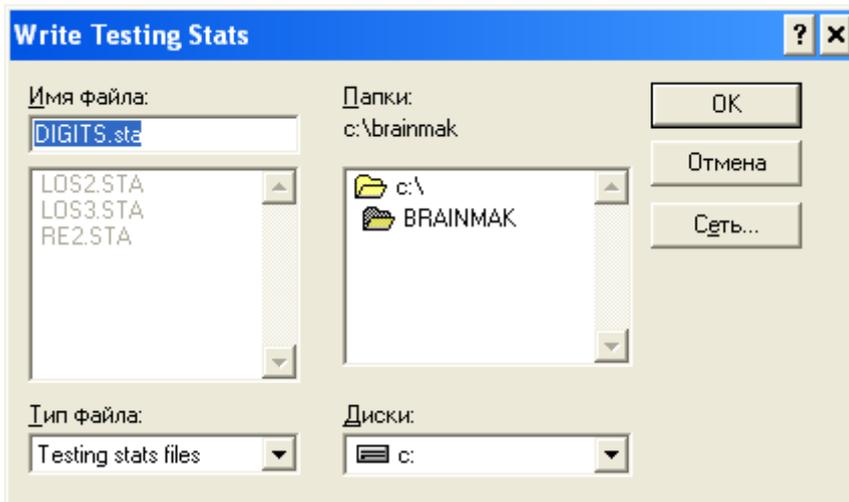
File → Training statistics...

Определяет файл для сохранения статистики обучения.



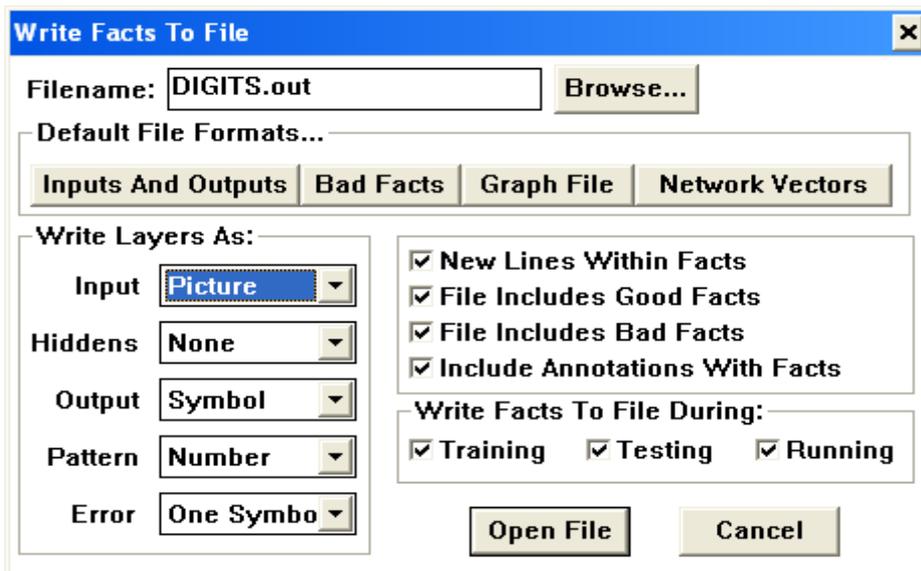
File → Testing statistics...

Определяет файл для сохранения статистики тестирования.



File → Facts to File...

Определяет файл для сохранения результатов каждого шага обучения или результатов работы обученной НС в рабочем режиме. После выбора данного пункта меню появляется окно диалога, позволяющее задать формат сохраняемой информации и режим работы данного пакета, при котором будет производиться запись информации.



Группа **Write Facts to File During** задает режим работы пакета, при котором в файле сохраняется необходимая информация: **Running** – рабочий режим, **Testing** – режим тестирования НС, **Training** – режим обучения сети.

Группа **Write Layers As** позволяет задать, в каком виде будут сохраняться параметры слоев НС (**Input** – входной, **Hiddens** – скрытые, **Output** – выходной, **Pattern** – значение образа для обучения, **Error** – величина ошибки обучения): **None** – не сохраняется, **Picture** – в виде изображения, используя символы псевдографики, **Number** – числа, **Symbol** – в виде текстовой информации, **One Symbol** – один символ.

Опция **New Line Within Facts** – использовать символ перевода строки (при необходимости) для очередной записи в файл.

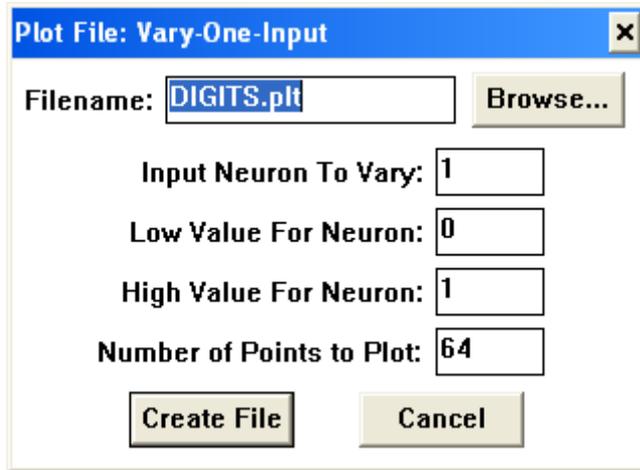
Опция **File Includes Good Facts** – записывать в файл хорошие факты.

Опция **File Includes Bad Facts** – записывать в файл плохие факты.

Опция **Include Annotation With Files** – включать в файл аннотацию.

File → Plot File: Vary One Input...

Позволяет сохранить в файле информацию об изменении выхода при изменении одного входа в заданном интервале.



После выбора данного пункта меню появляется окно диалога, позволяющее задать входной нейрон (поле **Input Neuron to Vary**) и интервал изменения входа (поля **Low/High Value For Neuron**).

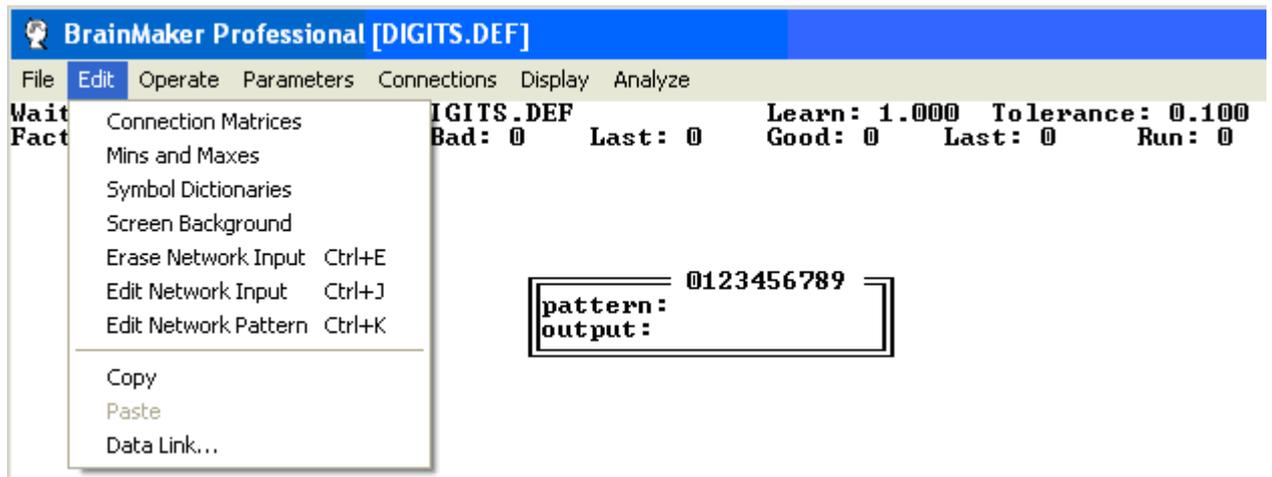
File → About BrainMaker Professional...

Позволяет выводить окно диалога с информацией о программе.

File → Exit

Выход из программы.

Меню Edit



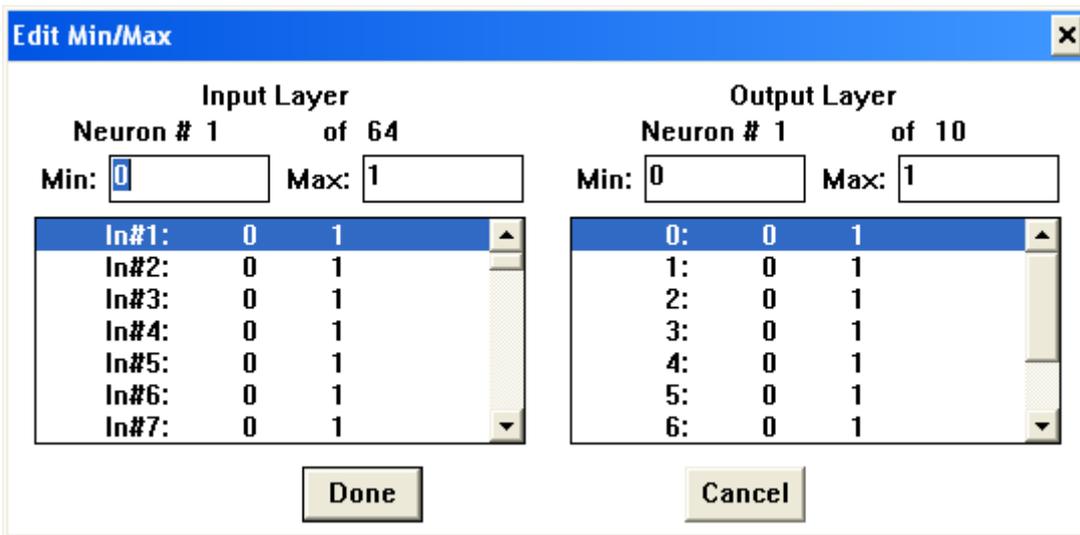
Edit → Connection Matrix

Позволяет редактировать веса нейронов в обученной или обучаемой сети.



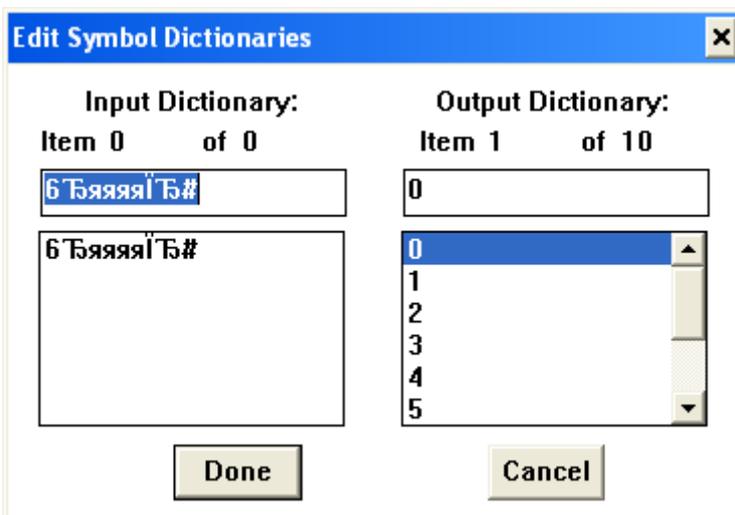
Edit → Mins and Maxes

Позволяет изменять интервал, в котором может изменяться величина веса необходимого нейрона входного или выходного слоя.

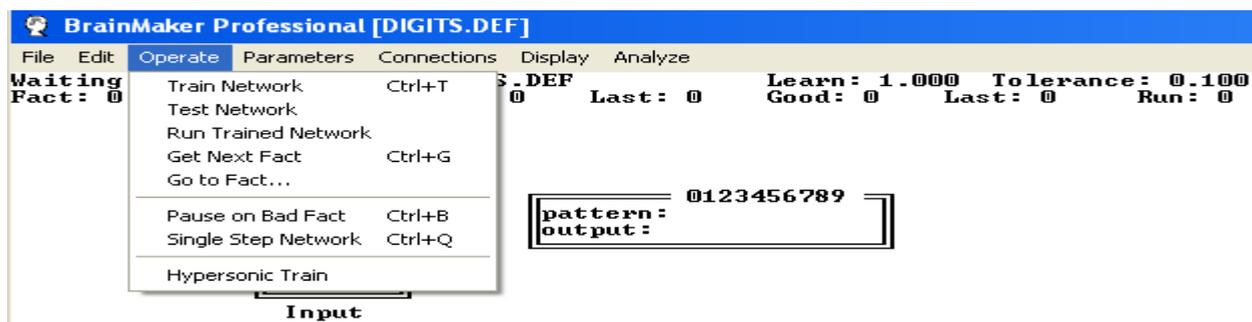


Edit → Symbol Dictionaries

Позволяет изменять значения элементов входного словаря или словаря указаний учителя для обучения.



Меню Operate



Operate → Train Network

Запуск процесса обучения НС. Повторный выбор данного пункта меню приостанавливает обучение.

Operate → Test Network

Тестирование НС с использованием тестовых фактов, определенных в файле обучающей выборки.

Operate → Run Trained Network

Запускает обученную НС в рабочем режиме. Данные работы обученной НС сохраняются в файле, определенном в окне диалога меню **File → Select Fact File....**

Operate → Get Next Fact

Позволяет производить обучение сети в пошаговом режиме.

Operate → Go to Facts...

Позволяет выбрать по номеру факт для следующего шага обучения (при пошаговом режиме обучения). После выбора данного пункта меню на экране появляется окно диалога, в которое необходимо ввести номер нужного факта.



Operate → Pause on Bad Fact

Пошаговый режим с остановкой на каждом плохом факте (неправильно распознанным НС). После выбора данного пункта меню на экране появляется окно диалога с одной единственной кнопкой **Next Fact**, нажатие которой позволяет перейти к следующему шагу обучения.



Повторный выбор данного пункта меню закрывает данный диалог.

Operate → Single Step Network

Пошаговый режим с остановкой на каждом факте. После выбора данного пункта меню на экране появляется окно диалога с одной единственной кнопкой **Next Fact**, нажатие которой позволяет перейти к следующему шагу обучения.



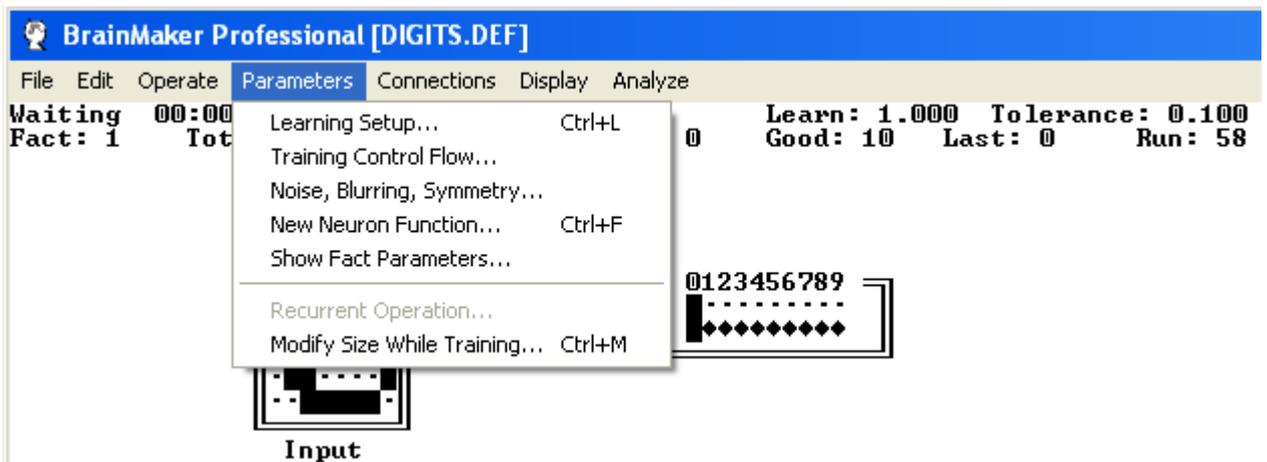
Повторный выбор данного пункта меню закрывает данный диалог.

Operate → Hypersonic Train

Режим очень быстрого обучения с потерей качества.

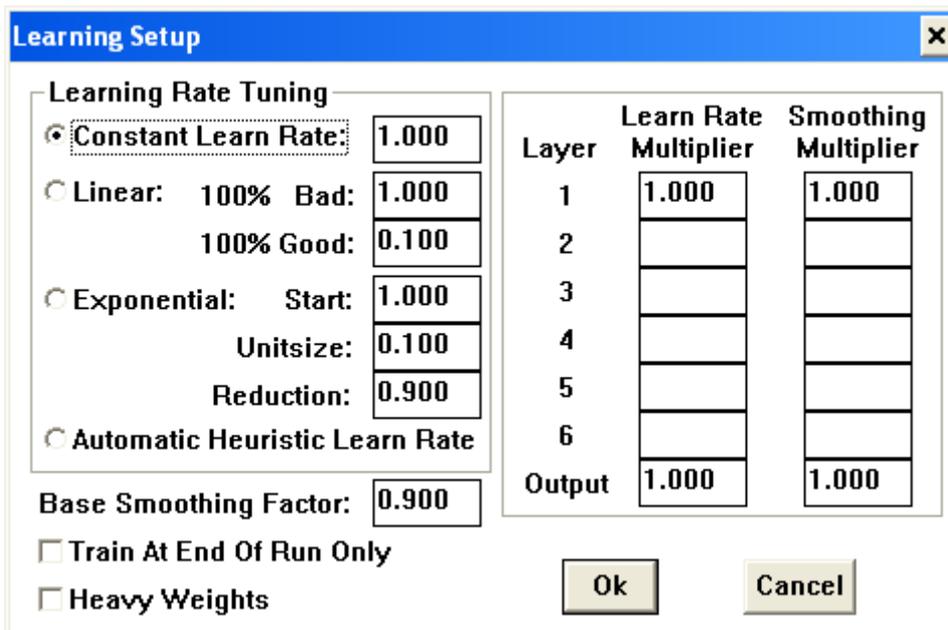


Меню Parameters



Parameters → Learning Setup

Позволяет устанавливать необходимые параметры обучения.



Группа **Learning Rate Tuning** позволяет изменять заданным образом шаг обучения.

- Опция **Constant Learn Rate** – постоянный коэффициент обучения.
- Опция **Linear 100% Bad, 100% Good** – линейное изменение коэффициента обучения, начиная с необученной сети – поле **100% Bad** и уменьшаясь до значения в поле **100% Good**.
- Опция **Exponential** – экспоненциальное изменение шага обучения.
- Опция **Automatic Heuristic Rate** – определение шага обучения с использованием эвристического алгоритма.

Поле **Base Smoothing Factor** – определяет базовый фактор сглаживания для обучения.

Справа в данном окне диалога имеется возможность задавать коэффициенты умножения для каждого слоя, для шага обучения и для фактора сглаживания.

Parameters → Training Control Flow

Позволяет задать параметры обучения – вероятность ошибки, условия останова обучения и т.д. После выбора данного пункта меню на экране появляется окно диалога, в котором можно задать необходимые условия.

Группа **Tolerances** позволяет задать ошибку распознавания при обучении (**Training Tolerance**) и тестировании (**Testing Tolerance**). Опция **Tolerance Tuning** – настройка динамического изменения вероятности ошибки.

Группа **Test While Training**: позволяет задать условие тестирования обучаемой НС.

Опция **Test After Every ... Runs (Facts)** – тестирование НС после необходимого числа прохождений фактов или после определенного числа шагов обучения. Опция **Save After Every ... Runs** – сохранение текущих параметров обучаемой НС в файл после определенного числа шагов обучения.

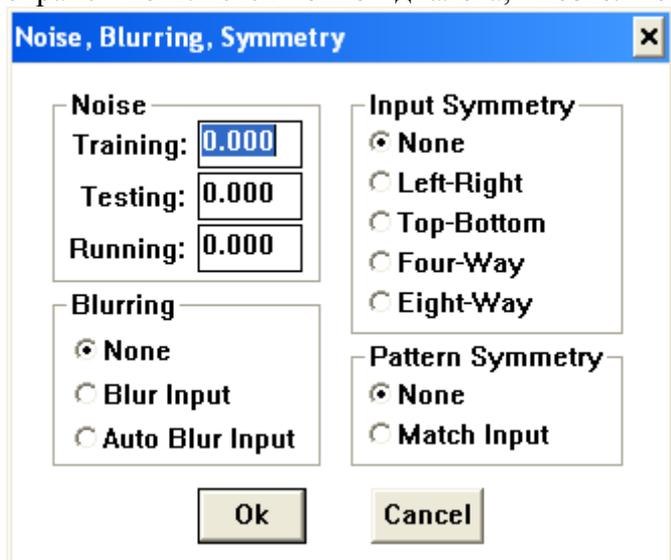
Группа **Stop Training When** задает условия окончания обучения: по количеству шагов обучения (**Run Number**), по проценту хороших фактов (**% of Good Training Facts**), по средней ошибке обучения (**Training Avg Error <=**), по значению параметра R-squared (**Training R-squared >=**), по величине квадратической ошибки обучения (**Training Squared Error <=**).

Если заданы опции тестирования обучаемой НС, то условие окончания процесса обучения задается также в группе **If Testing While Training**: по проценту хороших фактов (**% of Good Testing Facts**), по средней ошибке тестирования (**Testing Avg Error <=**), по значению параметра R-squared (**Testing R-squared >=**), по величине квадратической ошибки тестирования (**Testing Squared Error <=**) или условия того, чтобы все тестовые факты были хорошими (**All Testing Facts Are Good**). Если заданы опции настройки ошибки обучения **Tolerance Tuning**, то можно задать условие минимальной ошибки распознавания.

Parameters → Noise, Blurring, Symmetry

Позволяет задавать параметры шума, смазывания входной информации и симметрии входной и образцовой информации. После выбора данного пункта меню на

экране появляется окно диалога, позволяющее задавать необходимые параметры.



Группа **Noise: Training** – шум при обучении НС, **Testing** – шум при тестировании, **Running** – шум в рабочем режиме.

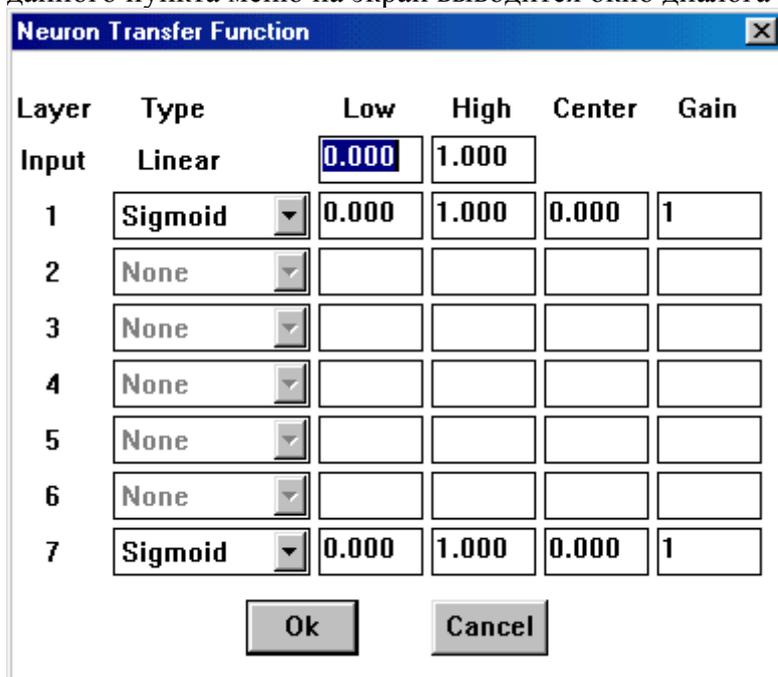
Группа **Blurring: None** – не используется, **Blur Input** – для входа, **Auto Blur Input** – автоматическое размывание входа.

Группа **Input Symmetry: None** – не используется, **Left-Right** – симметрия по оси Y, **Top-Bottom** – симметрия по оси X, **Four-Way** – симметрия по осям X и Y, **Eight-Way** – симметрия по осям X и Y и по диагоналям.

Группа **Pattern Input: None** – не используется, **Match Input** – такой же тип симметрии, как и у входа.

Parameters → New Neuron Function...

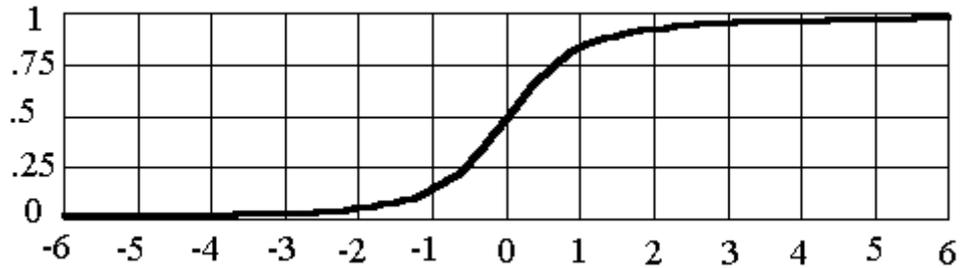
Позволяет использовать в нейронах разные функции возбуждения. После выбора данного пункта меню на экран выводится окно диалога для выбора функций возбуждения.



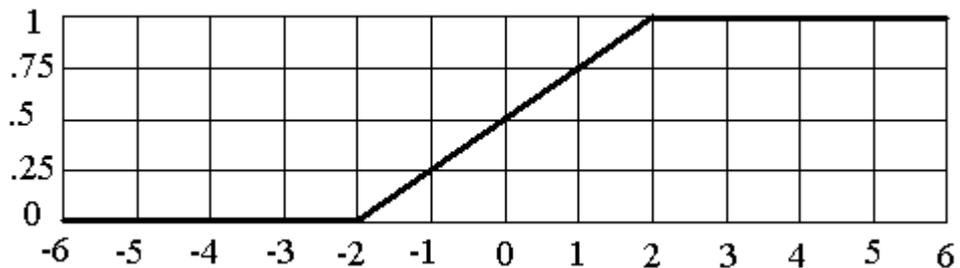
Окно диалога выбора функций возбуждения

В выпадающем списке выберете нужную функцию возбуждения соответствующего слоя. Седьмой (последний) слой соответствует выходному слою.

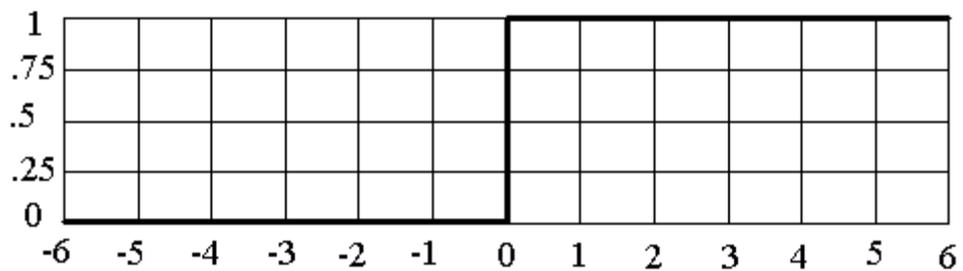
Sigmoid – сигмоидальная функция. Ее графическое представление:



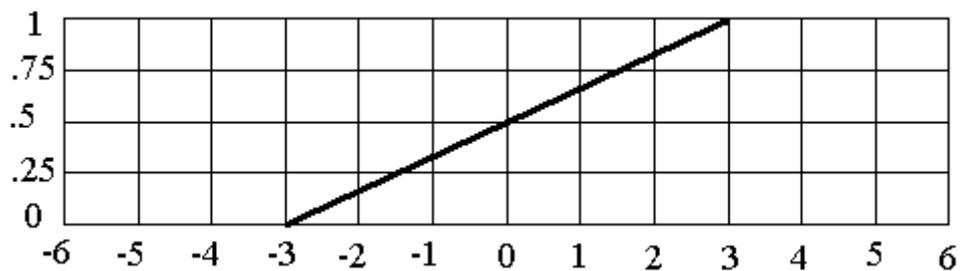
Threshold – линейная пороговая функция. Ее графическое представление:



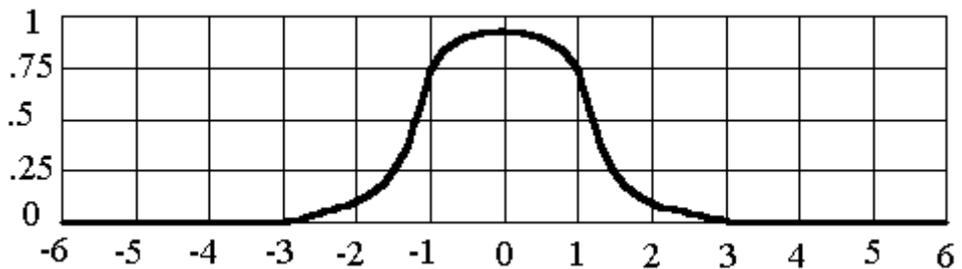
Step – бинарная функция возбуждения. Ее графическое представление:



Linear – линейная функция возбуждения. Ее графическое представление:



Gaussian – гауссовая. Ее графическое представление:



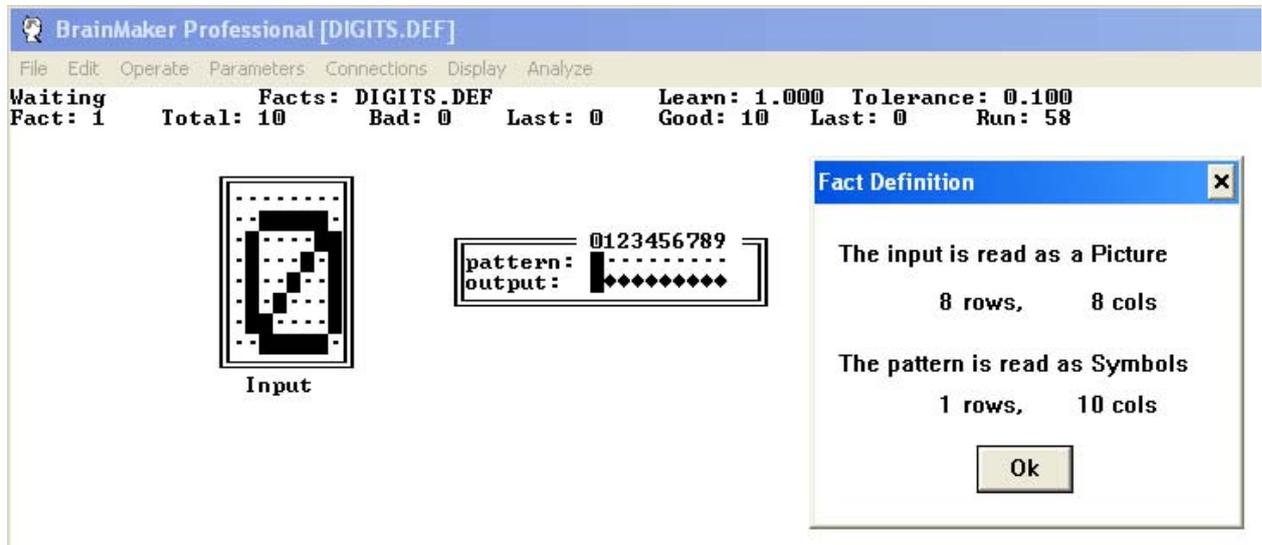
Поле **Low** – минимальное значение функции.

Поле **High** – максимальное значение функции.

Поле **Center** – точка, в которой функция принимает значение $(High+Low)/2$.

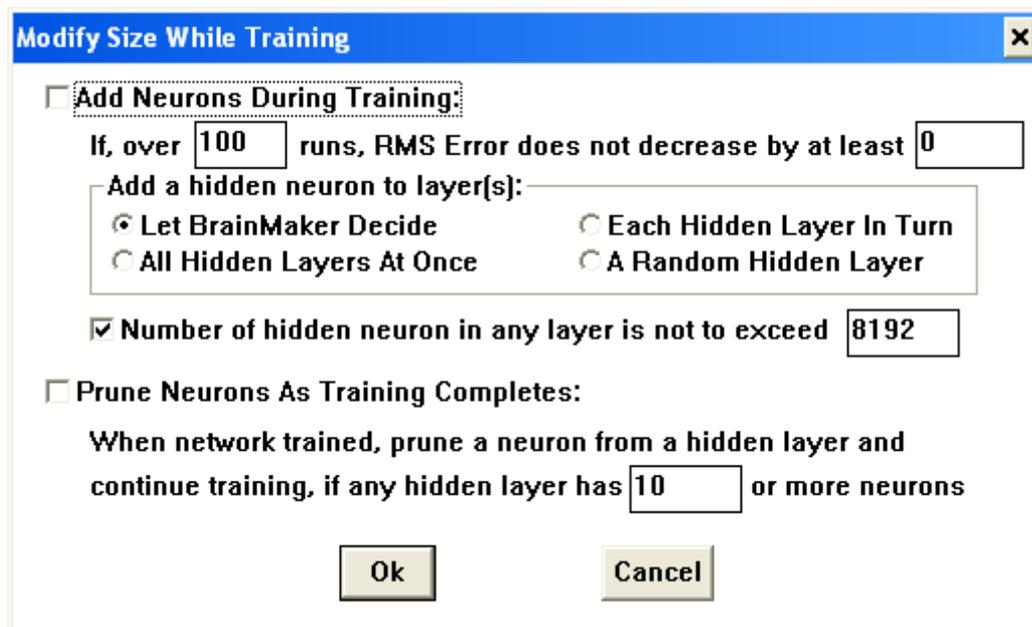
Parameters → Show Fact Parameters...

Позволяет получить информацию о типе и количестве входной информации и информации об образе для обучения.



Parameters → Modify Size While Training...

Позволяет задать условие изменения размера сети (добавление нейронов и скрытых слоев НС). После выбора данного пункта меню на экране появляется окно диалога, в котором можно задать необходимые условия.



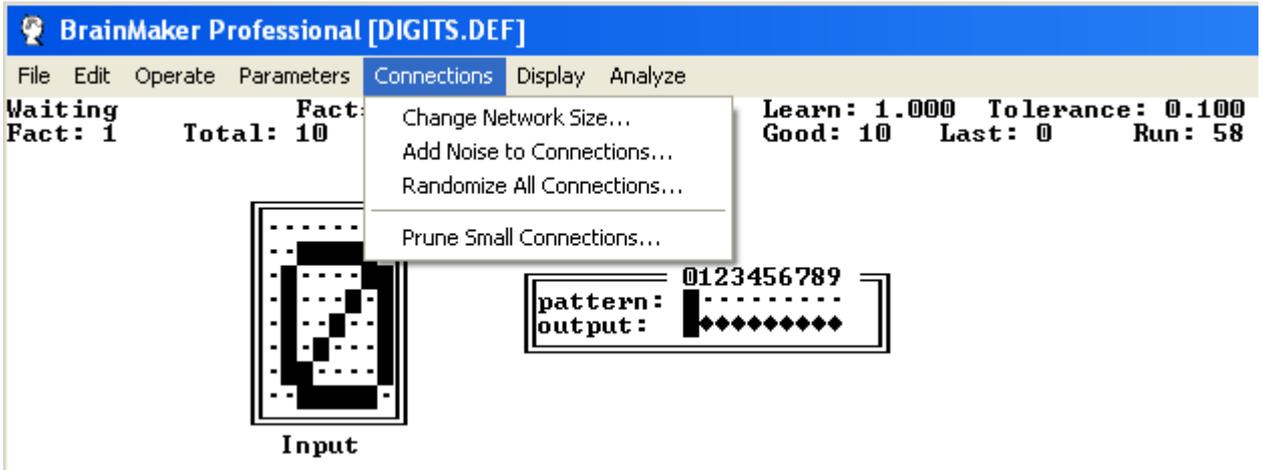
Опция **Adds Neuron During Training** – условие добавления нейронов во время обучения. Первое поле в данной для ввода информации - количество запусков. Следующее поле задает ограничение средней ошибки обучения, ниже которой не следует добавлять нейроны. Группа переключателей расположенная ниже задает алгоритм добавления нейронов:

- **Let BrainMaker Decide** – по решению BrainMaker;

- **All Hidden Layer At Once** – все внутренние слои один раз;
- **Each Hidden Layer In Turn** – добавить в каждый слой измененный;
- **A Random Hidden Layer** – добавить случайный внутренний слой.

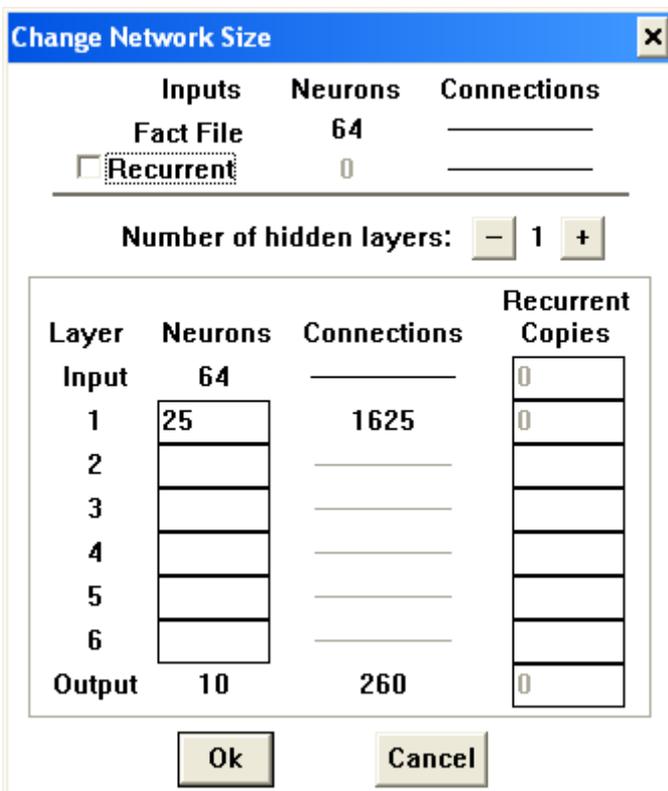
Опция **Prune Neurons As Training Completes** – позволяет удалить необходимое количество нейронов во внутренних слоях, если их количество превышает заданное. После этого обучение продолжается.

Меню Connections



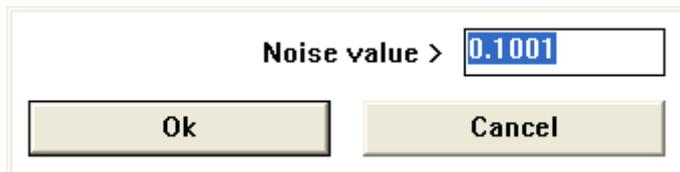
Connections → Change Network Size...

Выводит на экран окно диалога, позволяющее изменить количество внутренних слоев НС (поле **Number of hidden layers**) и количество нейронов в них (поля **Layer (1..6) Neurons Connection**).



Connections → Add Noise to Connections

Позволяет вносить определенные помехи в веса соединений между нейронами НС при обучении для улучшения качества работы НС в рабочем режиме. После выбора данного пункта меню на экран выводится окно диалога для ввода величины шума.



A dialog box titled "Noise value >" with a text input field containing "0.1001". Below the input field are two buttons: "Ok" and "Cancel".

Connections → Randomize All Connection

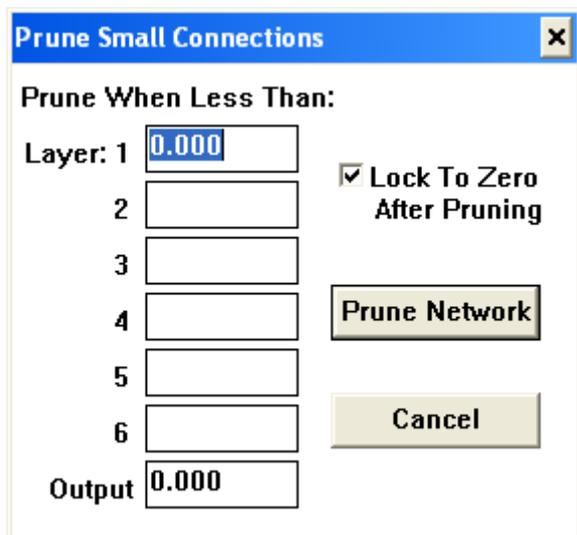
Присваивает весам всех соединений случайное значение, т.е. возвращает НС в исходное необученное состояние. После выбора данного пункта меню на экран выводится окно диалога для ввода нового коэффициента рандомизации (т.е. коэффициента для случайного распределения).



A dialog box titled "New random constant >" with a text input field containing "5". Below the input field are two buttons: "Ok" and "Cancel".

Connections → Prune Small Connections

Удаляет (обнуляет веса) соединения нейронов с весом меньше заданного. После выбора данного пункта меню на экран выводится окно диалога, в котором можно задать необходимое значение веса для нейронов внутренних слоев и выходного слоя НС.

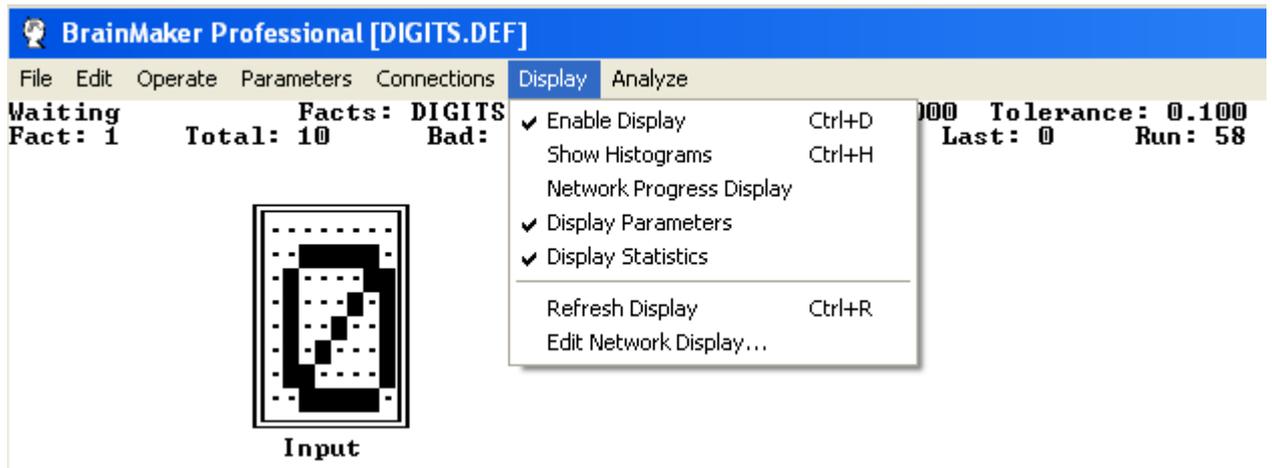


A dialog box titled "Prune Small Connections" with a close button (X) in the top right corner. The dialog contains the following elements:

- Prune When Less Than:** A label above a list of input fields.
- Layer: 1** with an input field containing "0.000".
- 2** with an empty input field.
- 3** with an empty input field.
- 4** with an empty input field.
- 5** with an empty input field.
- 6** with an empty input field.
- Output** with an input field containing "0.000".
- A checkbox labeled **Lock To Zero After Pruning** which is checked.
- A button labeled **Prune Network**.
- A button labeled **Cancel**.

Опция **Lock To Zero After Pruning** позволяет обнулять веса нейронов в процессе обучения при достижении заданного значения.

Меню Display

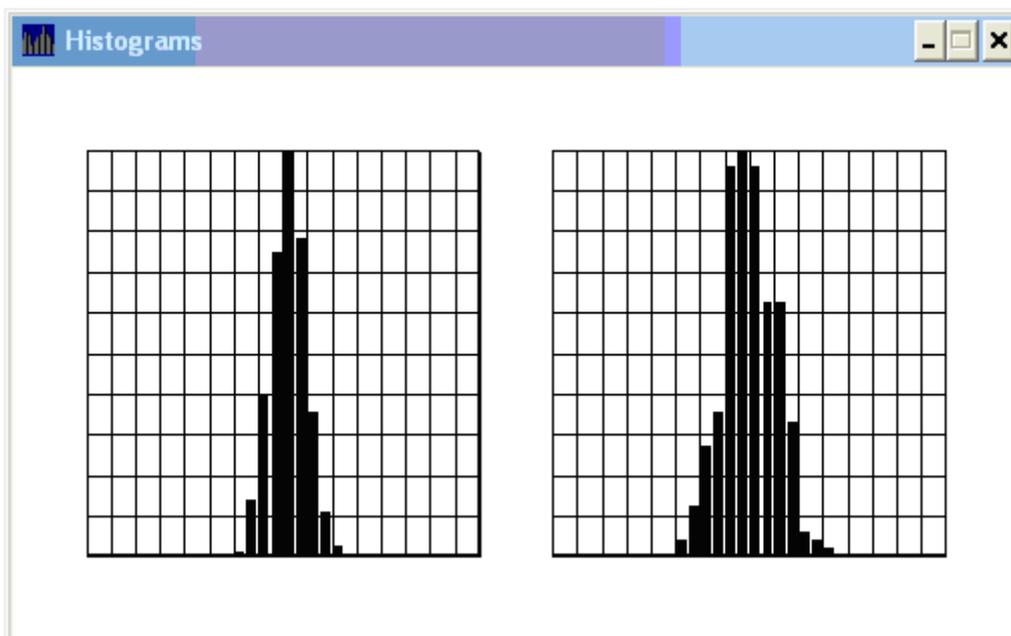


Display → Enable Display

Выводит на экран (или скрывает) отображение входных и выходных слоев НС, а также значение указания учителя для обучения НС. При использовании медленных компьютеров позволяет уменьшить время обучения сети.

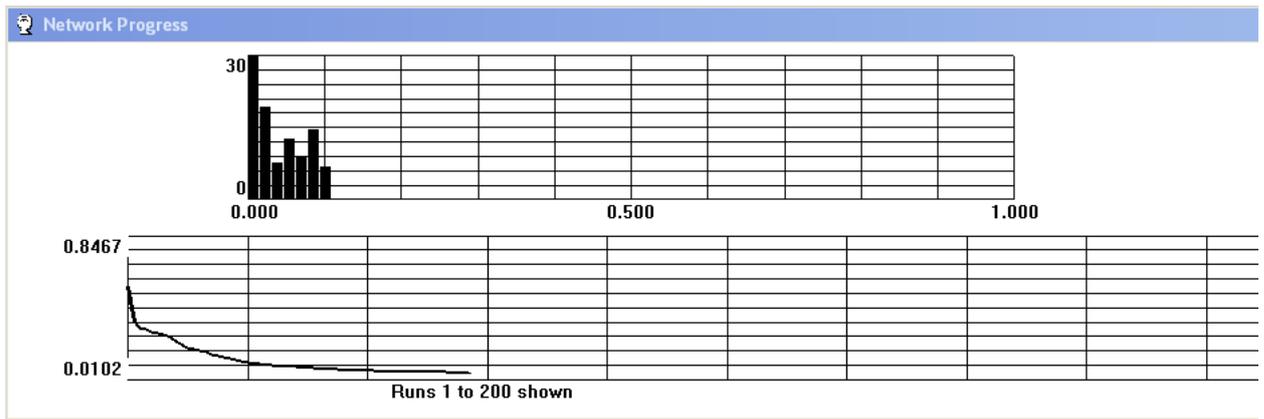
Display → Show Histogram

Скрывает/отображает окно гистограммы распределения весов соединений нейронов НС.



Display → Network Progress Display

Скрывает/отображает окно прогресса обучения сети. Нижний график отображает тенденцию изменения ошибки обучения. Верхний график отображает распределение по величине ошибки весов соединений.



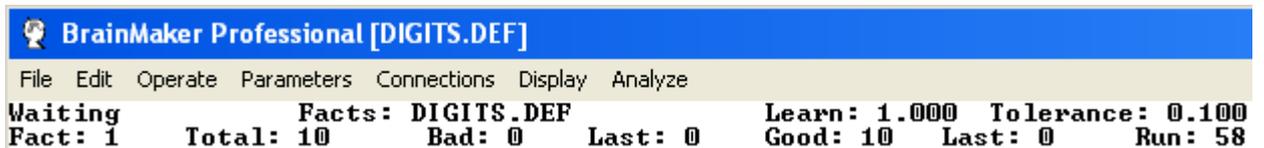
Display → Display Parameters

Скрывает/отображает строку параметров данной сети.



Display → Display Statistics

Скрывает/отображает строку статистики обучения НС.

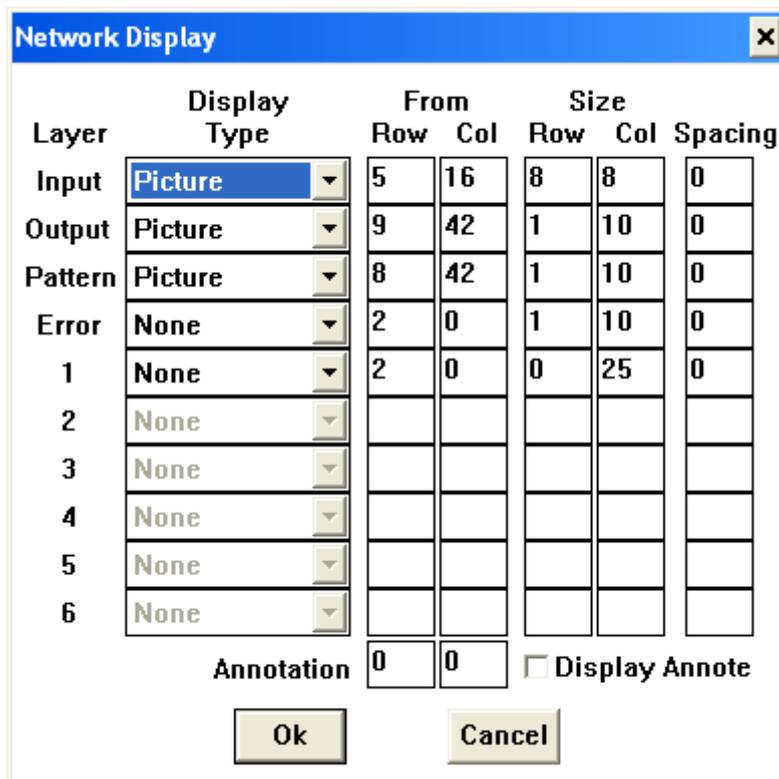


Display → Refresh Display

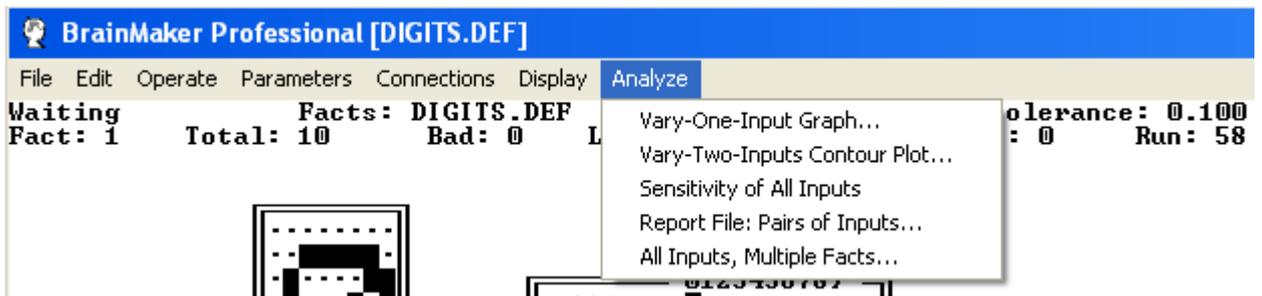
Обновляет окно программы.

Display → Edit Network Display

Позволяет задать, в каком виде будут отображаться слои НС (**Input** – входной, **1..6** – внутренние, **Output** – выходной, **Pattern** – значение указания учителя для обучения, **Error** – величина ошибки обучения): **None** – не сохраняется, **Picture** – в виде изображения, используя символы псевдографики, **Number** – числа, **Symbol** – в виде текстовой информации, **One Symbol** – один символ, **Thermometer** – горизонтальная гистограмма, **VertTherm** – вертикальная гистограмма.

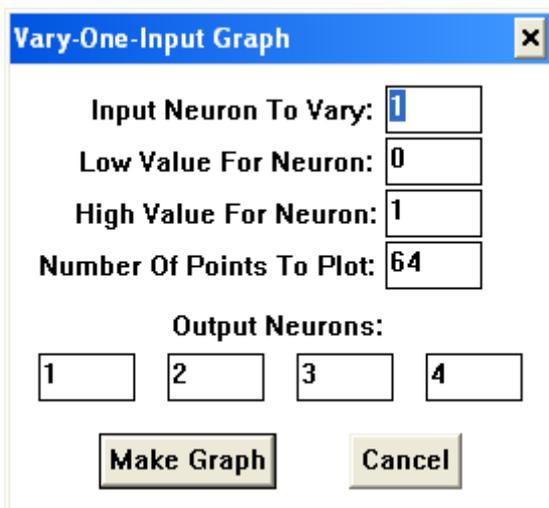


Меню Analyze



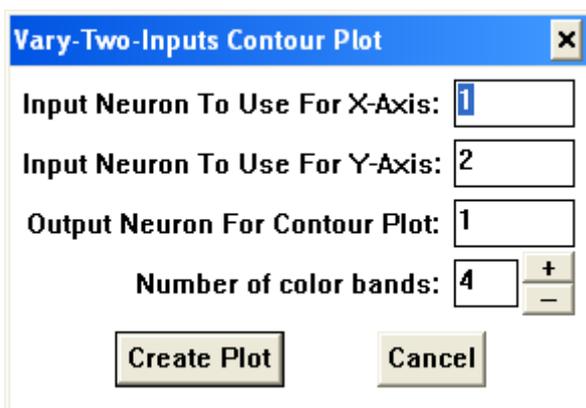
Analyze → Vary-One Input Graph...

Позволяет проанализировать влияние изменения одного входа на выход сети. После выбора данного пункта меню на экране появляется окно диалога, в котором пользователь определяет номер входного нейрона (**Input Neuron To Vary**) и интервал изменения входного значения (**Low Value For Neuron – High Value For Neuron**) на данном нейроне и выходной нейрон (**Output Neurons**).



Analyze → Vary-Two Inputs Contour Plot...

Позволяет проанализировать влияние изменения двух входов на выход сети. После выбора данного пункта меню на экране появляется окно диалога, в котором пользователь определяет номер входного нейрона для представления оси X - в поле **Input Neuron To Use For X-Axis**, оси Y - в поле **Input Neuron To Use For Y-Axis** и выходной нейрон для анализа - в поле **Output Neuron For Contour Plot**.



В данном окне диалога задается количество цветов для отображения графика в поле **Number of color bands**.

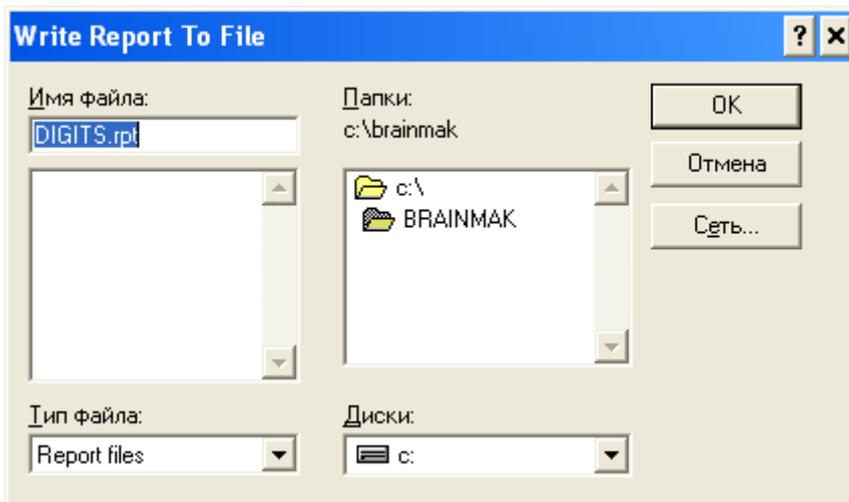
Analyze → Sensitivity of All Inputs

Определяет чувствительность всех входов сети.



Analyze → Report File: Pairs of Inputs...

Позволяет создать файл отчета о влиянии изменения величины входа двух входных нейронов на выход нейрона последнего слоя НС. После выбора данного пункта меню пользователю необходимо ввести имя файла отчета.



Analyze → All Inputs Multiple Facts...

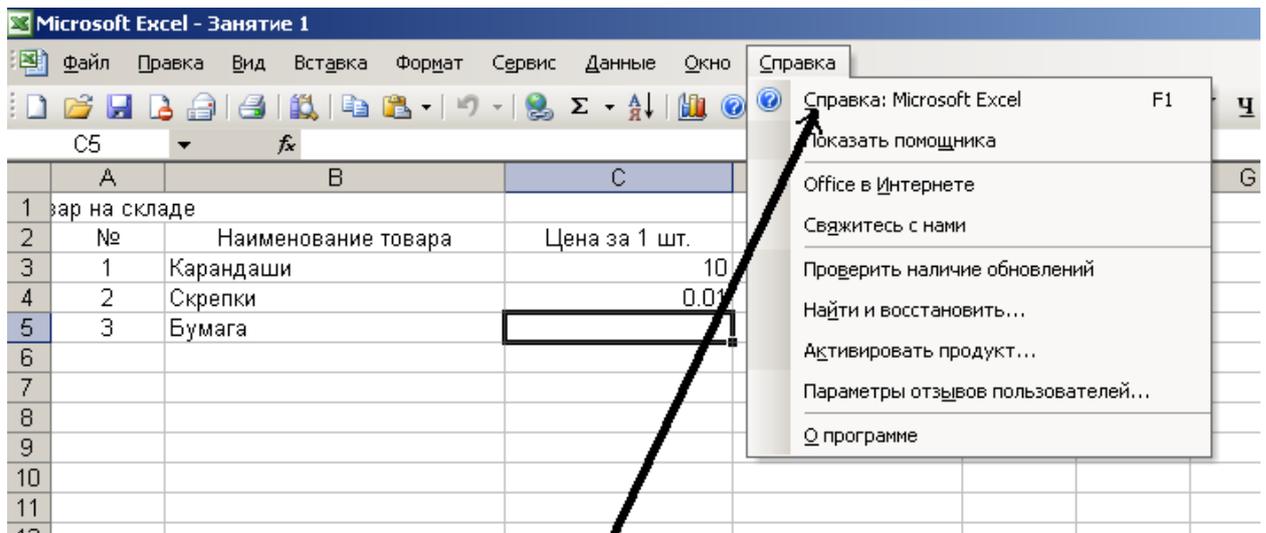
Анализ изменения входа при подаче на вход сети фактов. После выбора данного пункта меню на экране появляется окно диалога, в котором пользователь выбирает источник фактов (**Read Facts From**) и имя файла для записи в него отчета.



Приложение 2. Использование Excel для подготовки данных к исследованию.

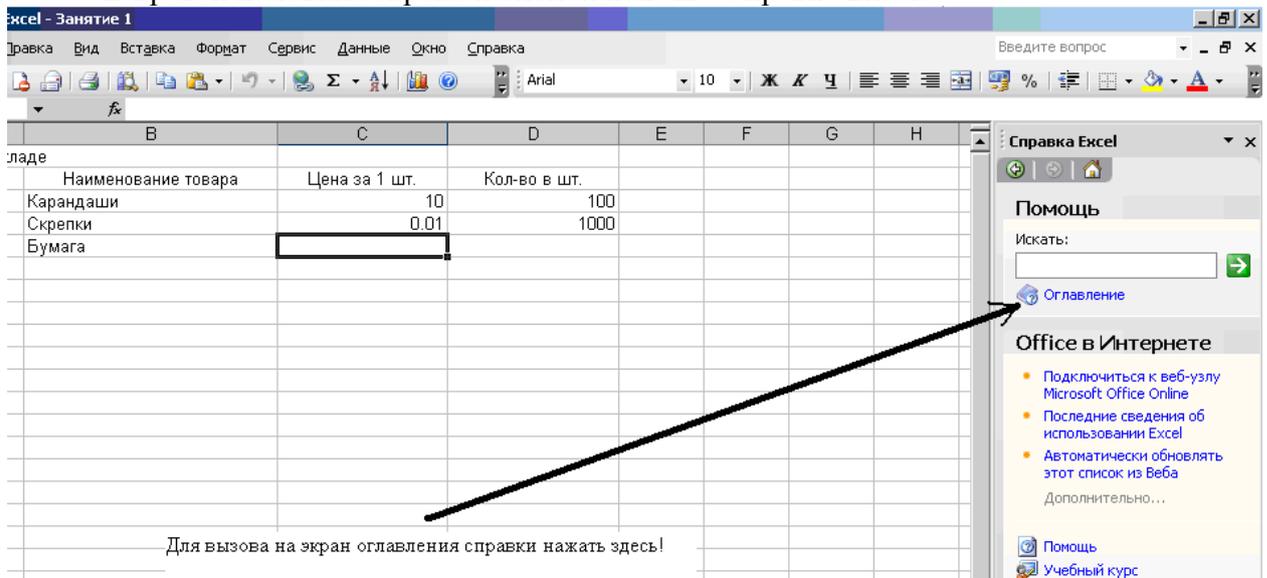
Справочная система Excel.

В основном меню программы Excel есть пункт «Справка»:



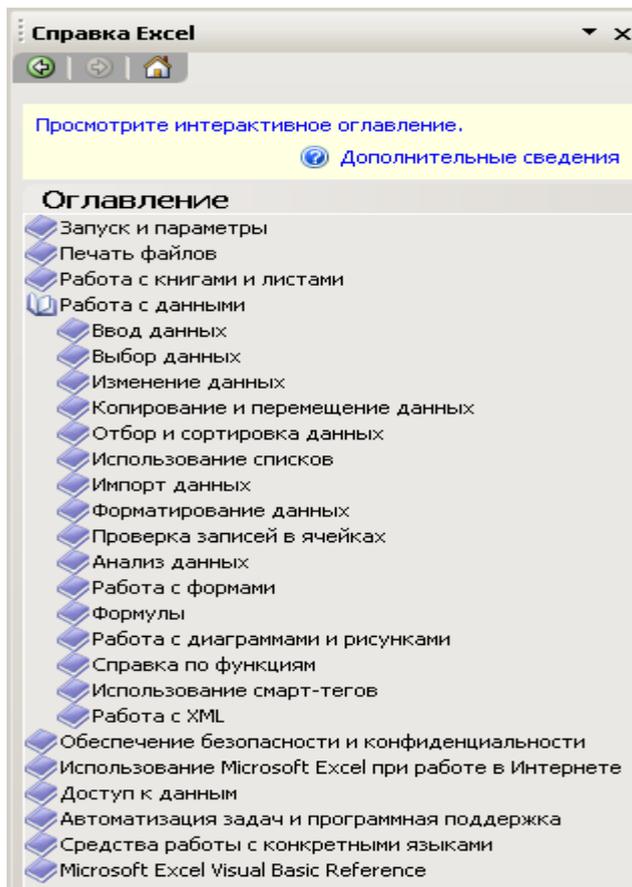
Для вызова справки - нажать! Или использовать клавишу F1

В правой половине экрана появится панель «Справка Excel»:

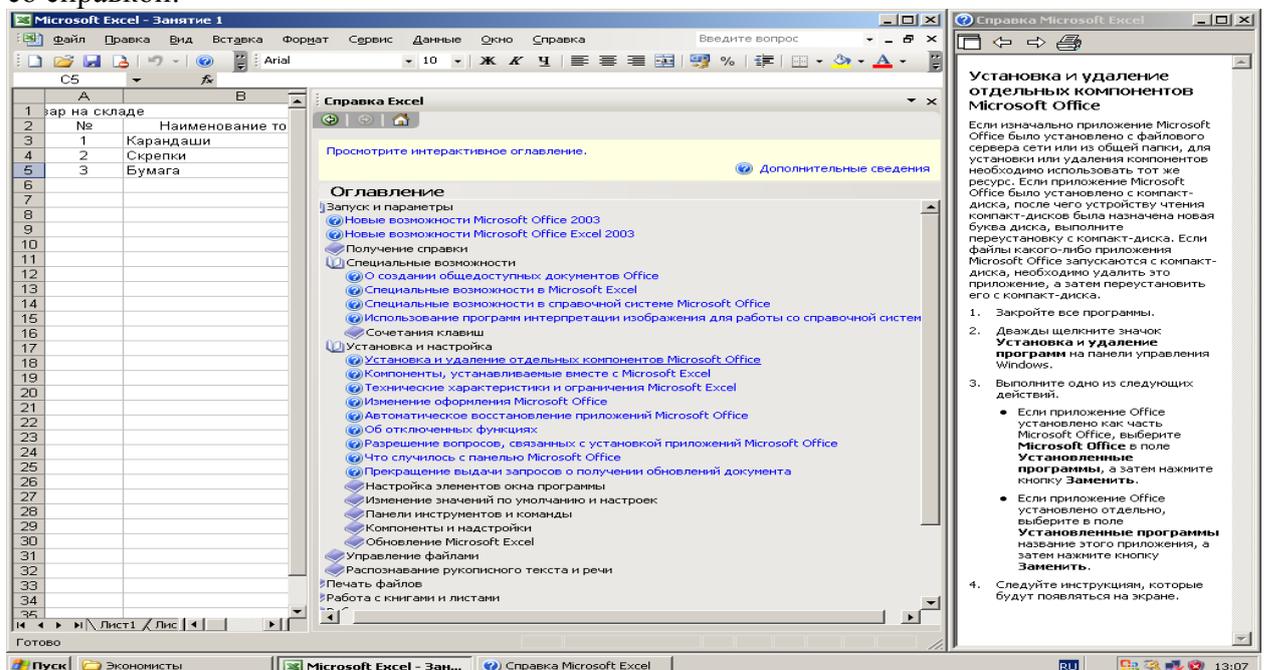


Для вызова на экран оглавления справки нажать здесь!

На экране появится оглавление, которое можно раскрыть на требуемом пункте, например – на работе с данными:



Если щёлкнуть по открывшемуся в оглавлении пункту, на экране появляется ещё одно окно, в котором содержится конкретная инструкция по выполнению требуемого действия. Размер дополнительных окон можно изменить так, чтобы удобно было работать со справкой:



Оглавление справочной системы имеет иерархическую структуру. Например, если выбрать из меню «Работа с данными» пункт «Ввод данных в ячейки листа», появится меню следующего уровня:

Ввод данных в ячейки листа

- ▶ Ввод чисел, текста, даты или времени суток
- ▶ Ввод чисел с фиксированным количеством десятичных разрядов или конечных нулей
- ▶ Ввод одного и того же значения в несколько ячеек одновременно
- ▶ Ввод и изменение одного и того же значения на нескольких листах
- ▶ Автоматическое заполнение повторяющихся записей в столбце
- ▶ Заполнение серии ячеек числами, датами либо другими элементами
- ▶ Ввод одинаковых данных в разные рабочие листы

Если из этого меню выбрать пункт «Ввод чисел, текста, даты или времени суток», в крайней правой панели появится текст, который может быть выделен стандартным для Microsoft Office способом и помещён в документ Word.

Ввод данных в ячейки листа

Щелкните ячейку, в которую необходимо ввести данные.

Наберите данные и нажмите клавишу ENTER или клавишу табуляции.

Числа и текст в списке

Введите данные в ячейку в первом столбце и нажмите клавишу табуляции для перемещения в соседнюю ячейку.

После ввода данных в первую строку нажмите клавишу ENTER, чтобы перейти на начало следующей.

Если ячейка, расположенная в начале следующей строки не становится активной, выберите команду Параметры в меню Сервис, а затем — вкладку Правка . В группе Параметры установите флажок Переход к другой ячейке после ввода , а затем из списка в направлении выберите значение Вниз.

Даты . При вводе даты используйте точку или дефис в качестве разделителя, например 09.05.2002 или Янв-2002. Чтобы ввести текущую дату, нажмите клавиши CTRL+; (точка с запятой).

Время суток . Для отображения времени суток в 12-часовом формате введите букву а или р, отделенную пробелом от значения времени, например 9:00 р. В противном случае время будет интерпретировано на основе 24-часового формата. Чтобы ввести текущее время, нажмите клавиши CTRL+SHIFT+: (двоеточие).

Ввод чисел с фиксированным количеством десятичных разрядов или конечных нулей.

В меню Сервис выберите команду Параметры, а затем откройте вкладку Правка.

Установите флажок Фиксированный десятичный формат при вводе.

В рамке разрядов введите положительное или отрицательное число, которое задает автоматическое умножение вводимого значения на десять в соответствующей степени.

Например, если ввести 3 в рамке разрядов, а затем — 2834 в ячейке, то значение будет равно 2,834. Если ввести -3 в рамке разрядов, а затем — 283, то значение будет равно 283000.

Изменение параметра Фиксированный десятичный формат при вводе не оказывает влияния на введенные до этого данные.

Чтобы для данного числа параметр «Фиксированный десятичный формат при вводе» не использовался, при наборе числа введите десятичную точку.

Ввод одного и того же значения в несколько ячеек одновременно:

Выделите ячейки, в которые необходимо ввести данные. Они не обязательно должны быть смежными.

Введите данные и нажмите клавиши CTRL+ENTER.

Ввод и изменение одного и того же значения на нескольких листах:

Если выбрана группа рабочих листов, то изменения в одном из них применяются ко всем выделенным листам. Данные могут быть изменены.

Выберите листы, на которые необходимо ввести данные.

При вводе или изменении данных меняются все выделенные ячейки. В ходе этих изменений могут быть заменены данных на активном и других листах.

Чтобы выделить Выполните следующее

Щелкните ярлычок листа.

Отдельный лист

Если ярлычок нужного листа не виден, найдите этот ярлычок с помощью кнопок прокрутки листов, а затем щелкните его.

Два или более смежных листа

Щелкните ярлычок первого листа, а затем, удерживая нажатой клавишу SHIFT, щелкните ярлычок последнего листа.

Два или более несмежных листа

Щелкните ярлычок первого листа, а затем, удерживая нажатой клавишу CTRL, последовательно щелкните ярлычки остальных листов.

Все листы книги

Щелкните правой кнопкой мыши ярлычок какого-либо листа, а затем выберите команду Выделить все листы в контекстном меню.

Примечание. Если ярлычки листов выделены цветом, имя на ярлычке при выделении будет подчеркиваться цветом, заданным пользователем. Если цвет ярлычка листа совпадает с цветом фона, значит этот лист не выделен.

Отмена выделения нескольких листов:

Чтобы отменить выделение нескольких листов книги, щелкните любой невыделенный лист.

Если на экране видны только выделенные листы, наведите указатель на выделенный лист и нажмите правую кнопку мыши. Затем выберите в контекстном меню команду Разгруппировать листы.

Укажите ячейку или выделите диапазон ячеек, в которые необходимо ввести данные.

Введите или измените данные в первой выделенной ячейке.

Нажмите клавишу ENTER или клавишу табуляции.

Примечание. Чтобы отменить выделение нескольких листов книги, щелкните любой невыделенный лист. Если на экране видны только выделенные листы, наведите указатель на выделенный лист и нажмите правую кнопку мыши. Затем выберите команду Разгруппировать листы в контекстном меню.

Автоматическое заполнение повторяющихся записей в столбце

Если несколько первых знаков, вводимых в ячейку, совпадают со знаками записи, ранее введенной в этом столбце, то недостающая часть набора будет произведена автоматически. В Microsoft Excel автоматический ввод производится только для тех

записей, которые содержат текст или текст в сочетании с числами. Записи, полностью состоящие из чисел, дат или времени, необходимо вводить самостоятельно.

Для подтверждения предлагаемого варианта, нажмите клавишу ENTER. Законченная запись по формату полностью совпадает с существующей записью, включая знаки верхнего и нижнего регистра.

Для замены автоматически введенных знаков продолжите ввод самостоятельно.

Для удаления автоматически введенных знаков нажмите клавишу BACKSPACE.

Чтобы выбрать значение из списка записей, которые уже имеются в столбце, нажмите правую кнопку мыши, а затем выберите в контекстном меню команду Выбрать из списка .

Заполнение серии ячеек числами, датами либо другими элементами:

Выделите первую из заполняемых ячеек.

Введите начальное значение для ряда значений.

Введите значение в соседнюю ячейку, чтобы определить образец заполнения.

Если требуется получить ряд 2, 3, 4, 5..., введите 2 и 3 в первые две ячейки. Если требуется получить ряд 2, 4, 6, 8..., введите 2 и 4. Если требуется получить ряд 2, 2, 2, 2..., вторую ячейку можно оставить пустой.

Чтобы задать тип ряда значений, перетащите маркер заполнения правой кнопкой мыши, а затем выберите соответствующую команду в контекстном меню. Например, если начальное значение — дата «янв-2002», то для получения ряда «фев-2002», «мар-2002» и т. д. выберите команду Заполнить по месяцам, а для получения ряда «янв-2003», «янв-2004» и т. д. выберите команду Заполнить по годам.

Для того чтобы управлять созданием ряда вручную или использовать для заполнения ряда значений клавиатуру, используйте команду контекстного меню — Прогрессия.

Выделите ячейку или ячейки, содержащие начальные значения.

Перетащите маркер заполнения через заполняемые ячейки.

Для заполнения в возрастающем порядке перетащите маркер вниз или вправо.

Для заполнения в убывающем порядке перетащите маркер вверх или влево.

Ввод одинаковых данных в разные рабочие листы:

Если данные были введены на одном листе, то их можно быстро скопировать в соответствующие ячейки других листов.

Выберите лист, содержащий введенные данные, и листы, на которые их необходимо скопировать.

Выделите ячейки, содержащие данные, которые необходимо скопировать.

Выберите команду Заполнить в меню Правка , а затем — команду По листам.

Принцип работы электронных таблиц

В отличие от обычных таблиц каждая ячейка электронной таблицы имеет адрес, который образуется так же, как и в популярной игре «Морской бой» из латинской буквы столбца и номера строки, где расположена ячейка.

В ячейке электронной таблицы может находиться текст, числа и формулы. Формулы начинаются со знака «=». В формулах можно использовать адреса других электронных ячеек, в этом случае они замещаются данными, находящимися в этих ячейках. Так, если в ячейке «E8» содержится число «50», а в ячейку «D8» введена формула «=E8*10», то в ячейке «d8» получится результат - число «500».

Ввод и редактирование данных в ячейках

Чтобы ввести данные в ячейку электронной таблицы необходимо щелкнуть на ней левой клавишей мыши. Ячейка выделится более жирным контуром, после чего в нее можно начинать вводить данные. Для окончания ввода можно воспользоваться клавишей

«Enter» или щелкнуть мышью на другой ячейке. Если данные в ячейку вводятся в первый раз, то для завершения ввода можно воспользоваться стрелками на клавиатуре, при этом автоматически выделится ячейка, расположенная рядом с данной ячейкой по направлению действия стрелки.

Для редактирования данных в ячейке нужно дважды щелкнуть на ней мышью. При редактировании данных можно пользоваться стрелками на клавиатуре и мышью для позиционирования курсора в ячейке.

Форматирование данных в ячейках

Чтобы отобразить данные в ячейке в заданном формате, нужно нажать на ячейке правой клавишей мыши и в контекстном меню выбрать пункт «Формат ячейки». Появится следующее окно.

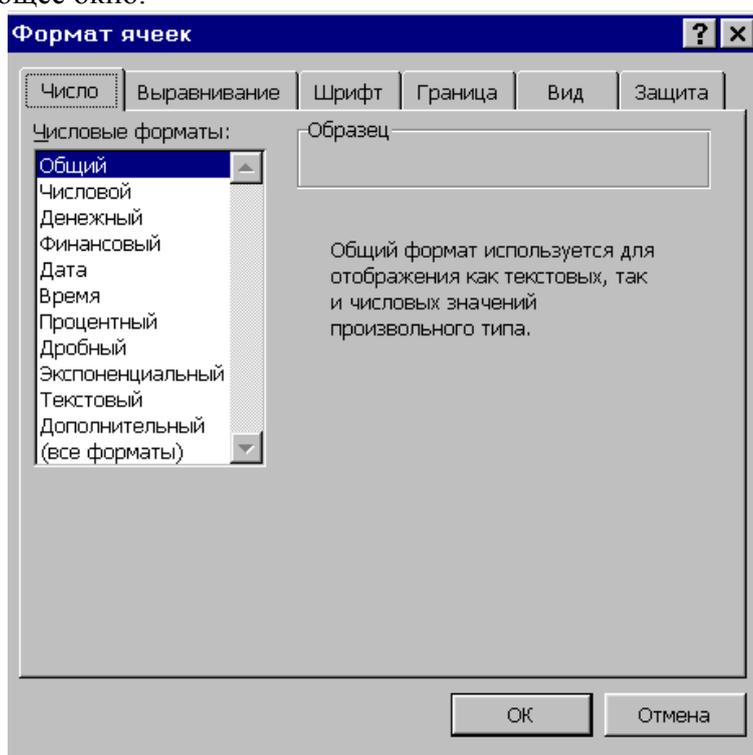


Рис. 1. Окно диалога выбора формата ячейки

С помощью этого окна можно устанавливать различные форматы для данных в ячейках. Многие из опций этого диалогового окна повторяются в панели инструментов форматирования.

Форматирование данных в Excel

Форматирование текста и отдельных знаков.

Чтобы по-особенному выделить текст, можно отформатировать весь текст в ячейке или отдельные знаки. Выделите знаки, которые нужно отформатировать, и нажмите кнопку на панели инструментов Форматирование.



Вращение текста и границ.

Данные в столбце занимают, как правило, гораздо меньше места, чем нужно для подписи столбца. Чтобы не создавать неоправданно широкие столбцы или использовать

На вкладке «число» устанавливаем значение «текст», на вкладке «выравнивание» - отмечаем «переносить по словам» и в окне «Ориентация» устанавливаем значение «45 градусов». Надписи по вертикали не умещаются, поэтому получается не очень красиво:



Увеличим высоту строки:



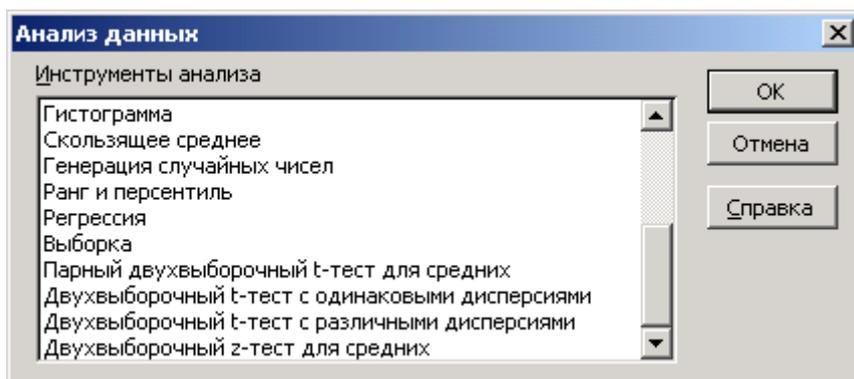
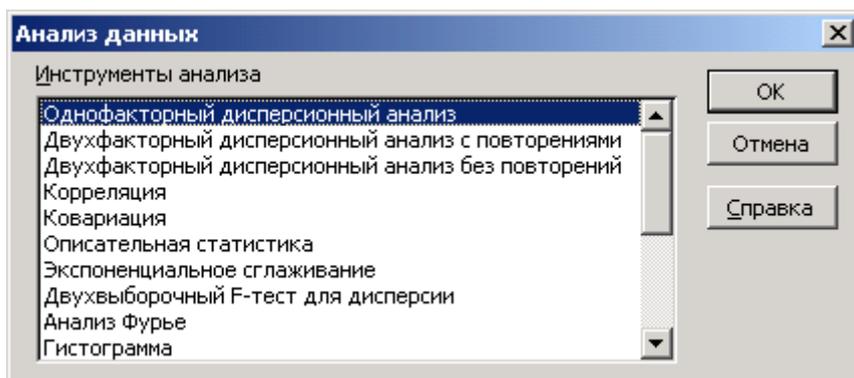
надписи выстроились так, как было задумано.

Основной состав средств анализа данных

В состав Microsoft Excel входит набор средств анализа данных (так называемый пакет анализа), предназначенный для решения сложных статистических и инженерных задач. Для анализа данных с помощью этих инструментов следует указать входные данные и выбрать параметры; анализ будет выполнен с помощью подходящей статистической или инженерной макрофункции, а результат будет помещен в выходной диапазон. Другие средства позволяют представить результаты анализа в графическом виде.

- ▶ Дисперсионный анализ
- ▶ Корреляционный анализ
- ▶ Ковариационный анализ
- ▶ Описательная статистика
- ▶ Экспоненциальное сглаживание
- ▶ Двухвыборочный F-тест для дисперсии
- ▶ Анализ Фурье
- ▶ Гистограмма
- ▶ Скользящее среднее
- ▶ Генерация случайных чисел
- ▶ Ранг и перцентиль
- ▶ Регрессия
- ▶ Выборка
- ▶ T-тест
- ▶ Z-тест

Другие функции. В Microsoft Excel представлено большое число статистических, финансовых и инженерных функций. Некоторые из них являются встроенными, другие доступны только после установки пакета анализа.



Выполнение статистического анализа

В меню Сервис выберите команду Анализ данных.

Если эта команда недоступна, загрузите пакет анализа:

В меню Сервис выберите команду Надстройки.

В списке надстроек выберите Пакет анализа и нажмите кнопку ОК.

Выполните инструкции программы установки, если это необходимо.

Выберите нужную функцию в диалоговом окне Анализ данных и нажмите кнопку ОК.

Установите параметры анализа в соответствующем диалоговом окне.

Чтобы получить дополнительные сведения о параметрах, нажмите в диалоговом окне кнопку Справка.

Описательная статистика

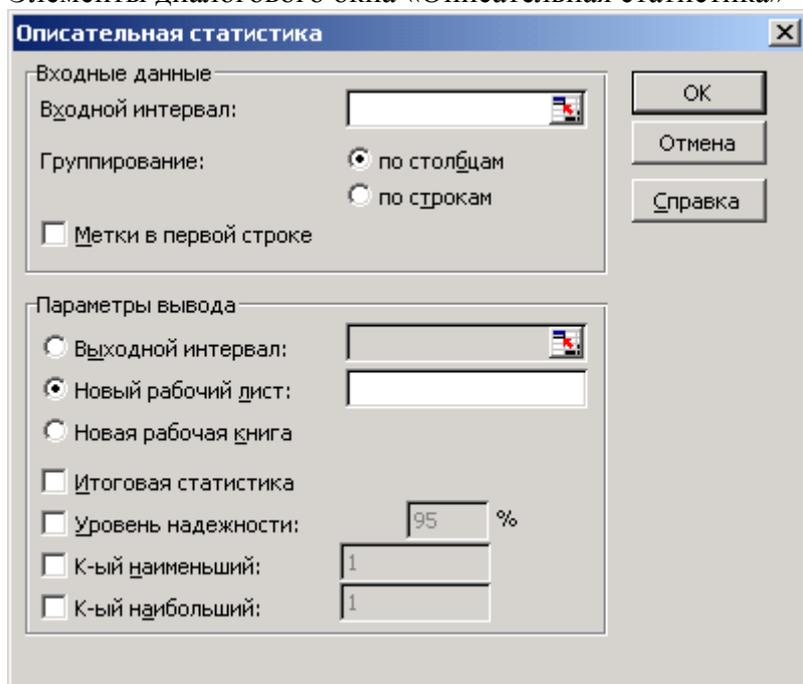
Это средство анализа служит для создания одномерного статистического отчета, содержащего информацию о центральной тенденции и изменчивости входных данных.

Статистический отчет включает в себя 16 характеристик, часть из которых – не обязательная, заказывается дополнительно:

- Среднее,
- Стандартная ошибка (среднего),
- Медиана,
- Мода,
- Стандартное отклонение,
- Дисперсия выборки,
- Эксцесс,
- Асимметричность,
- Интервал,
- Минимум,
- Максимум,
- Сумма,

Счет,
Наибольшее (#),
Наименьшее (#),
Уровень надежности

Элементы диалогового окна «Описательная статистика»



Диалоговое окно «Описательная статистика».

- Входной интервал
- Ссылка на диапазон, содержащий анализируемые данные.

Ссылка должна состоять не менее чем из двух смежных координат (начало – и конец данных), данные в которых расположены по строкам или столбцам.

- Группирование

Установите переключатель в положение По столбцам или По строкам в зависимости от расположения данных во входном диапазоне.

- Метки в первой строке/Метки в первом столбце

Если первая строка исходного диапазона содержит названия столбцов, установите переключатель в положение Метки в первой строке.

Если названия строк находятся в первом столбце входного диапазона, установите переключатель в положение Метки в первом столбце.

Если входной диапазон не содержит меток, то необходимые заголовки в выходном диапазоне будут созданы автоматически.

- Уровень надежности

Установите флажок, если в выходную таблицу необходимо включить строку для уровня надежности. В поле введите требуемое значение. Например, значение 95% вычисляет уровень надежности среднего со значимостью 0.05.

- К-ый наибольший

Установите флажок, если в выходную таблицу необходимо включить строку для k-го наибольшего значения для каждого диапазона данных. В соответствующем окне введите число k. Если k равно 1, эта строка будет содержать максимум из набора данных.

- К-ый наименьший

Установите флажок, если в выходную таблицу необходимо включить строку для k-го наименьшего значения для каждого диапазона данных. В соответствующем окне введите число k. Если k равно 1, эта строка будет содержать минимум из набора данных.

- Выходной диапазон

Введите ссылку на левую верхнюю ячейку выходного диапазона.

Этот инструмент анализа выводит два столбца сведений для каждого набора данных.

Левый столбец содержит метки статистических данных; правый столбец содержит статистические данные.

Состоящий из двух столбцов диапазон статистических данных будет выведен для каждого столбца или для каждой строки входного диапазона в зависимости от положения переключателя Группирование.

- Новый лист

Установите переключатель, чтобы открыть новый лист в книге и вставить результаты анализа, начиная с ячейки A1.

Если в этом есть необходимость, введите имя нового листа в поле, расположенном напротив соответствующего положения переключателя.

- Новая книга

Установите переключатель, чтобы открыть новую книгу и вставить результаты анализа в ячейку A1 на первом листе в этой книге.

- Итоговая статистика

Установите флажок, если в выходном диапазоне необходимо получить по одному полю для каждого из следующих видов статистических данных: Среднее, Стандартная ошибка (среднего), Медиана, Мода, Стандартное отклонение, Дисперсия выборки, Эксцесс, Асимметричность, Интервал, Минимум, Максимум, Сумма, Счет, Наибольшее (#), Наименьшее (#), Уровень надежности.

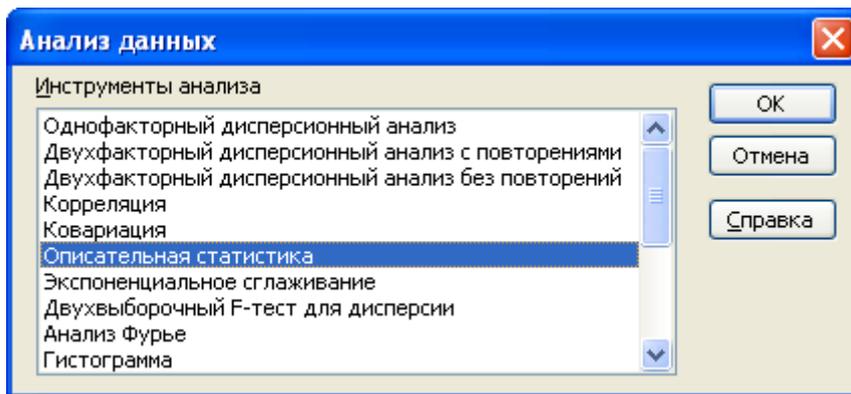
Пример получения описательной статистики.

Исходная таблица:

	E	F	G	H	I	J
1						
2						
3						
4		Измерения	Прямая последовательность	Обратная последовательность	Последовательность с 0	
5		Первое	1	9	0	
6		Второе	2	8	1	
7		Третье	3	7	2	
8		Четвёртое	4	6	3	
9		Пятое	5	5	4	
10		Шестое	6	4	5	
11		Седьмое	7	3	6	
12		Восьмое	8	2	7	
13		Девятое	9	1	8	
14						
15						

После подготовки исходной таблицы запускаем Сервис -> Анализ_данных. Если в сервисе анализа данных нет, находим Сервис -> Надстройка -> Анализ данных. После этого в сервисе появится опция Анализ данных.

Выбираем Описательную статистику:

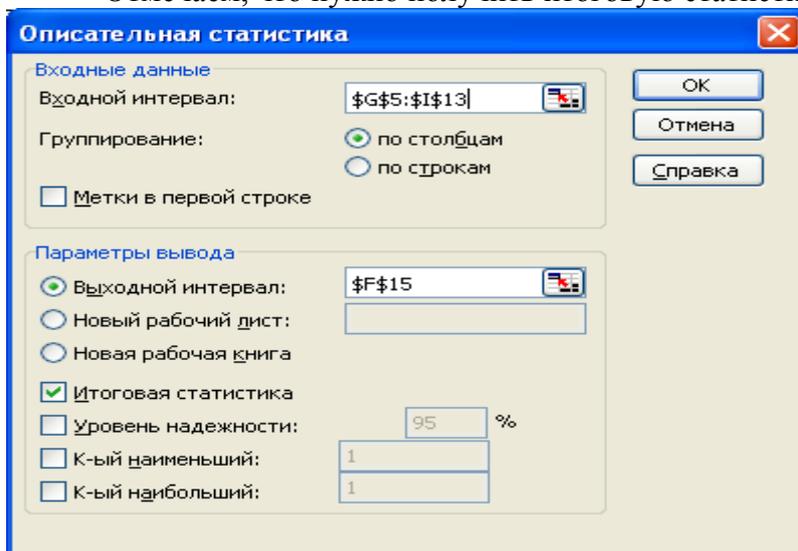


В появившемся окне производим настройку, заполняя только требуемые позиции:
Указываем входной интервал $GG5:I13$, т.е. в каком месте таблицы находятся обрабатываемые данные

Отмечаем, что данные расположены по столбцам

Указываем левую верхнюю границу выводимых результатов: $F15$

Отмечаем, что нужно получить итоговую статистику



После заполнения параметров описательной статистики на экране в указанном нами месте появится результат:

	C	D	E	F	G	H	I	J	K	L
1										
2										
3										
4				Измерения	Прямая последовательность	Обратная последовательность	Последовательность с 0			
5				Первое	1	9	0			
6				Второе	2	8	1			
7				Третье	3	7	2			
8				Четвёртое	4	6	3			
9				Пятое	5	5	4			
10				Шестое	6	4	5			
11				Седьмое	7	3	6			
12				Восьмое	8	2	7			
13				Девятое	9	1	8			
14										
15				Столбец1		Столбец2		Столбец3		
16										
17				Среднее	5	Среднее	5	Среднее	4	
18				Стандартная	0,912870929	Стандартная ошибка	0,912870929	Стандартн	0,912871	
19				Медиана	5	Медиана	5	Медиана	4	
20				Мода	#1/Д	Мода	#1/Д	Мода	#1/Д	
21				Стандартное	2,738612788	Стандартное отклонени	2,738612788	Стандартн	2,738613	
22				Дисперсия в	7,5	Дисперсия выборки	7,5	Дисперсия	7,5	
23				Эксцесс	-1,2	Эксцесс	-1,2	Эксцесс	-1,2	
24				Асимметричн	0	Асимметричность	0	Асимметр	0	
25				Интервал	8	Интервал	8	Интервал	8	
26				Минимум	1	Минимум	1	Минимум	0	
27				Максимум	9	Максимум	9	Максимум	8	
28				Сумма	45	Сумма	45	Сумма	36	
29				Счет	9	Счет	9	Счет	9	
30										

Результат содержит 13 запрошенных статистических показателей. Поскольку при настройке окна описательной статистики не было указано, что метки находятся в первой строке, названия меток для выходной таблицы результатов Excel придумал самостоятельно.

Скольльзящее среднее

Скольльзящее среднее используется для расчета значений в прогнозируемом периоде на основе среднего значения переменной для указанного числа предшествующих периодов.

Скольльзящее среднее, в отличие от простого среднего для всей выборки, содержит сведения о тенденциях изменения данных.

Этот метод может использоваться для прогноза сбыта, запасов и других процессов. Расчет прогнозируемых значений выполняется по следующей формуле.

$$F_{(t+1)} = \frac{1}{N} \sum_{i=1}^N A_{t-i+1}$$

где:

N — число предшествующих периодов, входящих в скольльзящее среднее;

A_j — фактическое значение в момент времени j ;

F_j — прогнозируемое значение в момент времени j .

Элементы диалогового окна «Скольльзящее среднее»

- **Входной диапазон**

Введите ссылку на диапазон исследуемых данных. Входной диапазон должен состоять из одного столбца или одной строки, содержащих не менее четырех ячеек с данными.

- **Метки в первой строке**

Установите флажок, если первая строка входного интервала содержит заголовки. Снимите флажок, если заголовки отсутствуют; в этом случае подходящие названия для данных выходного диапазона будут созданы автоматически.

- **Интервал**

Введите число значений, необходимое для расчета скользящего среднего. Значение по умолчанию равно 3.

- **Выходной диапазон**

Введите ссылку на левую верхнюю ячейку выходного диапазона. Если установлен флажок Стандартные погрешности, то выходной диапазон состоит из двух столбцов, и значения стандартных погрешностей содержатся в правом столбце. Если исходных значений для построения прогноза или для вычисления стандартной ошибки недостаточно, Microsoft Excel возвратит значение ошибки #Н/Д.

Выходной диапазон и исходные данные должны находиться на одном листе. По этой причине параметры Новый лист и Новая книга недоступны.

- **Вывод графика**

Установите флажок для автоматического создания встроенной диаграммы на листе, содержащем выходной диапазон.

- **Стандартные погрешности**

Установите флажок, чтобы включить в выходной диапазон столбец стандартных погрешностей. Снимите флажок, чтобы получить выходной диапазон в виде одного столбца без значений стандартных погрешностей.

Пример вычисления скользящего среднего.

Во время эксперимента получен следующий ряд значений, характеризующих исследуемую величину:

	А	В	С
1			
2			
3			
4	Номер измерения	Значение	
5	1	12	
6	2	35	
7	3	46	
8	4	35	
9	5	57	
10	6	46	
11	7	69	
12	8	103	
13	9	92	
14	10	92	
15	11	103	
16	12	126	
17	13	184	
18			

Определим скользящее среднее, которое в отличие от простого среднего для всей выборки, содержит сведения о тенденциях изменения данных.

Для этого вызовем: Сервис – Анализ_данных – Скользящее_среднее:

Скользящее среднее

Входные данные

Входной интервал: 

Метки в первой строке

Интервал:

Параметры вывода

Выходной интервал: 

Новый рабочий лист:

Новая рабочая книга

Вывод графика Стандартные погрешности

OK
Отмена
Справка

Установим адрес диапазона исследуемых данных, размер группы определим в 3 числа, результат поместим в таблице с адреса D4.

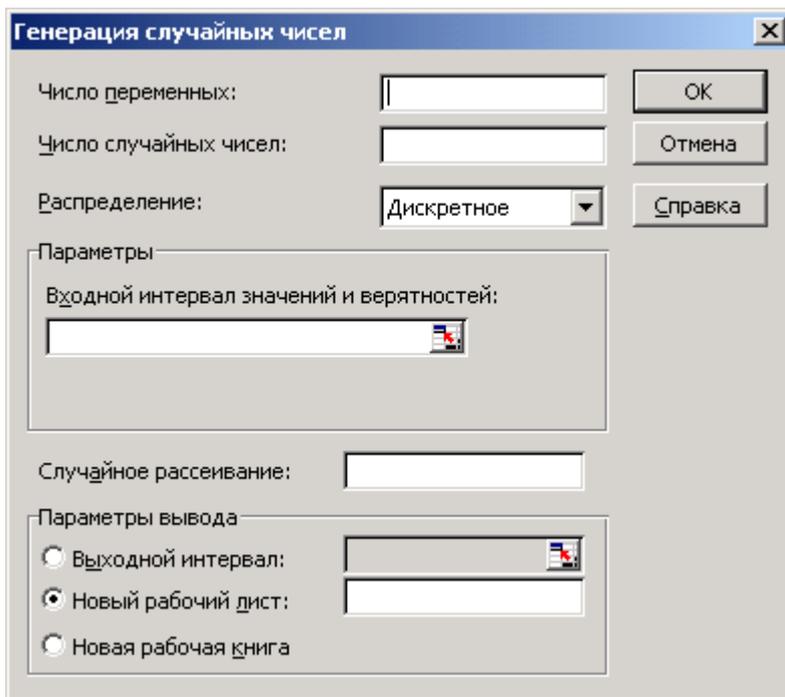
	A	B	C	D	E
1					
2					
3					
4	Номер измерения	Значение		Скользящее среднее	
5	1	12		#N/D	
6	2	35		#N/D	
7	3	46		31	
8	4	35		38,66666667	
9	5	57		46	
10	6	46		46	
11	7	69		57,33333333	
12	8	103		72,66666667	
13	9	92		88	
14	10	92		95,66666667	
15	11	103		95,66666667	
16	12	126		107	
17	13	184		137,6666667	
18					

Скользящее среднее монотонно возрастает (чего не скажешь об исходной колонке B), что говорит о наличии тренда – т.е. стремления полученного ряда значений постоянно расти.

Генерация случайных чисел

Используется для заполнения диапазона случайными числами, извлеченными из одного или нескольких распределений. С помощью данной процедуры можно моделировать объекты, имеющие случайную природу, по известному распределению вероятностей.

Например, можно использовать нормальное распределение для моделирования совокупности данных по росту индивидуумов, или использовать распределение Бернулли для двух вероятных исходов, чтобы описать совокупность результатов бросания монеты.



Элементы диалогового окна «Генерация случайных чисел»

- Число переменных

Введите число столбцов значений, которые необходимо разместить в выходном диапазоне. Если это число не введено, то все столбцы в выходном диапазоне будут заполнены.

- Число случайных чисел

Введите число случайных значений, которое необходимо вывести для каждой переменной. Каждое случайное значение будет помещено в строке выходного диапазона. Если число случайных чисел не будет введено, все строки выходного диапазона будут заполнены.

- Распределение

Выберите распределение, которое необходимо использовать для генерации случайных переменных.

- Равномерное

Характеризуется верхней и нижней границами. Переменные извлекаются с одной и той же вероятностью для всех значений интервала. Обычно приложения используют равномерное распределение в интервале 0...1.

- Обычное

Характеризуется средним значением и стандартным отклонением. Обычно приложения для этого распределения используют среднее значение 0 и стандартное отклонение 1.

- Бернулли

Характеризуется вероятностью успеха (величина p) в данной попытке. Случайные переменные Бернулли имеют значение 0 или 1. Например, можно выбрать равномерную случайную переменную в интервале 0...1. Если переменная меньше или равна вероятности успеха, то случайной переменной Бернулли присваивается значение 1, а в противном случае, она принимает значение 0.

- Биномиальное

Характеризуется вероятностью успеха (величина p) для некоторого числа попыток. Например, можно сгенерировать случайные переменные Бернулли число попыток, сумма которых будет биномиальной случайной переменной.

- Пуассона

Характеризуется значением лямбда, равным 1/среднее. Распределение Пуассона часто используется для характеристики числа случайных событий, происходящих в единицу времени, например, среднее количество автомобилей, приезжающих на платную стоянку.

- Модельное

Характеризуется нижней и верхней границей, шагом, числом повторений значений и числом повторений последовательности.

- Дискретное

Характеризуется значением и соответствующим ему интервалом вероятности. Диапазон должен состоять из двух столбцов: левого, содержащего значения, и правого, содержащего вероятности, связанные со значением в данной строке. Сумма вероятностей должна быть равна 1.

- Параметры

Введите параметры выбранного распределения.

- Случайное рассеивание

Введите произвольное значение, для которого необходимо генерировать случайные числа. Впоследствии можно снова использовать это значение для получения тех же самых случайных чисел.

- Выходной диапазон

Введите ссылку на левую верхнюю ячейку выходного диапазона. Размер выходного диапазона будет определен автоматически, и на экран будет выведено сообщение в случае возможного наложения выходного диапазона на исходные данные.

- Новый лист

Установите переключатель, чтобы открыть новый лист в книге и вставить результаты анализа, начиная с ячейки A1. Если в этом есть необходимость, введите имя нового листа в поле, расположенном напротив соответствующего положения переключателя.

- Новая книга

Установите переключатель, чтобы открыть новую книгу и вставить результаты анализа в ячейку A1 на первом листе в этой книге.

Пример 1.

Генерация 10 случайных чисел, распределённых по нормальному закону.

Вызываем: Сервис – Анализ_данных – Генерация случайных чисел – и настраиваем полученное окно:

Генерация случайных чисел

Число переменных: 1 OK

Число случайных чисел: 10 Отмена

Распределение: Нормальное Справка

Параметры

Среднее = 0

Стандартное отклонение = 1

Случайное рассеивание:

Параметры вывода

Выходной интервал: \$F\$5

Новый рабочий лист:

Новая рабочая книга:

Получаем:

	A	B	C	D	E	F	G
1							
2							
3							
4							
5						2,139313	
6						-1,01713	
7						1,143392	
8						-0,82855	
9						0,548525	
10						-0,41319	
11						0,301112	
12						1,533608	
13						2,627312	
14						0,97099	
15							

Пример 2.

Сгенерировать 3 последовательности по 10 случайных чисел, подчиняющихся биномиальному закону распределения с вероятностью успеха 0,9 при 100 испытаниях.

Генерация случайных чисел

Число переменных: 3

Число случайных чисел: 10

Распределение: Биномиальное

Параметры

Значение $p =$ 0,9

Число испытаний = 100

Случайное рассеивание:

Параметры вывода

Выходной интервал: \$B\$3

Новый рабочий лист:

Новая рабочая книга

OK Отмена Справка

Получаем:

	A	B	C	D	E
1					
2					
3		86	93	84	
4		88	91	88	
5		93	92	89	
6		86	92	92	
7		91	92	95	
8		84	88	88	
9		92	91	91	
10		95	92	95	
11		88	88	90	
12		91	83	90	
13					

Выборка данных из таблицы

Если имеющаяся совокупность исходных данных слишком велика для обработки или построения диаграммы, можно использовать представительную выборку.

В инструментарии Excel для этого существует такое средство статистического анализа, как выборка.

Это средство позволяет создать выборку из генеральной совокупности, рассматривая имеющийся входной диапазон, как генеральную совокупность.

Кроме того, если предполагается периодичность входных данных, то можно создать выборку, содержащую значения только из отдельной части цикла.

Например, если входной диапазон содержит данные для квартальных продаж, создание выборки с периодом 4 разместит в выходном диапазоне значения продаж из одного и того же квартала.

Выборка

Входные данные
 Входной интервал: 
 Метки

Метод выборки
 Периодический
 Период:
 Случайный
 Число выборок:

Параметры вывода
 Выходной интервал: 
 Новый рабочий лист:
 Новая рабочая книга

OK
 Отмена
 Справка

Элементы диалогового окна «Выборка»

- Входной диапазон

Введите ссылку на блок данных на рабочем листе, содержащем значения генеральной совокупности, из которой необходимо извлечь выборку. Выборка будет извлечена сначала из первого столбца, затем из второго столбца и так далее.

- Заголовки

Установите флажок, если первая строка или первый столбец входного диапазона содержит заголовки. Снимите флажок, если заголовки во входном диапазоне отсутствуют; в этом случае подходящие названия в выходном диапазоне будут созданы автоматически.

- **Метод выборки**

Установите переключатель в положение Периодический или Случайный, чтобы задать необходимый диапазон выборки.

- **Период**

Введите периодический интервал, в соответствии с которым будет произведена выборка. Входное значение, номер которого совпадает с номером, заданным в поле *периода*, и каждое последующее с номером, кратным *периоду*, будет скопировано в выходной столбец. Процесс создания выборки прекратится при достижении конца входного диапазона.

- **Число выборок**

Введите в поле число случайных значений, которые необходимо разместить в выходном столбце. Позиция каждой извлекаемой переменной во входном диапазоне выбирается случайно, и любое исходное значение может быть выбрано более одного раза.

- **Выходной диапазон**

Введите ссылку на левую верхнюю ячейку выходного диапазона. Выходные данные будут записаны в виде одного столбца. Если установить переключатель в положение Периодический, число значений в выходном диапазоне будет равно числу значений во входном диапазоне, деленному на значение периода. Если установить переключатель в положение Случайный, то число значений в выходном диапазоне будет равно числу выборок.

- **Новый лист**

Установите переключатель, чтобы открыть новый лист в книге и вставить результаты анализа, начиная с ячейки A1. Если в этом есть необходимость, введите имя нового листа в поле, расположенном напротив соответствующего положения переключателя.

- **Новая книга**

Установите переключатель, чтобы открыть новую книгу и вставить результаты анализа в ячейку A1 на первом листе в этой книге.

Пример создания выборки данных из генеральной совокупности.

Имеется исходная таблица, содержащая 1 переменную с 29 реализациями. Необходимо выбрать для исследования случайным образом 10 реализаций.

Вызываем: Сервис – Анализ_данных – Выборка, и настраиваем появившееся окно:

Выборка

Входные данные

Входной интервал:

Метки

Метод выборки

Периодический
Период:

Случайный
Число выборок:

Параметры вывода

Выходной интервал:

Новый рабочий лист:

Новая рабочая книга

OK
Отмена
Справка

В результате получаем:

	A	B	C	D	E	F	G
1							
2							
3						91	
4		86				95	
5		88				86	
6		93				92	
7		86				88	
8		91				95	
9		84				91	
10		92				92	
11		95				90	
12		88				93	
13		91					
14		93					
15		91					
16		92					
17		92					
18		88					
19		91					
20		92					
21		88					
22		83					
23		84					
24		88					
25		89					
26		92					
27		95					
28		88					
29		91					
30		95					
31		90					
32		90					
33							

Если необходимо, чтобы выборка была представительной, репрезентативность её нужно проверять и обеспечивать отдельно.

Приложение 3. Поиск информации в Интернет.

Базовая поисковая техника Google.

Для поиска требуемой информации составляется поисковое предписание, содержащее характеризующие эту информацию ключевые слова.

Дополнительные правила составления поискового запроса:

- Google не воспринимает более 10 слов для поиска, включая специальный синтаксис.
- Знак (+) в поисковом предписании используется для поиска наиболее общих слов (пробелы между знаком и помечаемым этим знаком словом не допускаются). Часто используемые слова "I", "a", "the", "of" и т.п. игнорируются поисковиком, но можно заставить его их искать если поставить перед ними "+". Например: "Война +и мир"
- Знак (-) в поисковом предписании используется для исключения термина из поиска (пробелы между знаком и помечаемым этим знаком словом не допускаются)
- Для поиска фразы надо поместить её в кавычки «»
- Знак «*» означает любое слово. Google не поддерживает поиска по корням слов (stemming), то есть возможности использования звёздочки (или другого знака маски) вместо букв в искомом слове. Например, moon* в поисковике, поддерживающем маски, найдёт "moonlight," "moonshot," "moonshadow," и т.д. Google же использует звёздочку как заменитель целого слова. Поиск по фразе "three * mice" в Google даст в результате "three blind mice," "three blue mice," "three red mice," и т.д.
- Обойти лимит в 10 слов можно используя звёздочки. Каждая звёздочка заменяет одно слово. Как оказалось, Google просто не считает количество звёздочек в запросе.

Нестандартные возможности поисковой машины Google.

Так как Google является полнотекстовым поисковиком, он индексирует всё содержимое страниц. Дополнительные команды, называемые спец. синтаксисом (операторы для продвинутого поиска) позволяют пользователям Google искать конкретные части web страниц или тип информации. Это позволяет сузить число результатов поиска.

Такие операторы имеют следующий синтаксис: «operator:search_term» (в этом выражении также не должно быть никаких пробелов).

Примеры этих операторов:

- **site:** инструктирует Google ограничить поиск конкретным web-сайтом (доменом); название сайта (домена) указывается сразу после двоеточия и без пробела. "site:" сужает поиск до одного сайта или домена верхнего уровня. Например:
site:loc.gov
site:thomas.loc.gov
site:edu
site:nc.us
- **filetype:** инструкция произвести поиск только в пределах текста указываемого типа файлов. Тип файла указывается после двоеточия (точку перед расширением файла указывать не нужно). "filetype:" ищет среди

расширений файлов, а точнее - в файлах с определённым расширением.

Например:

homeschooling filetype:pdf

"leading economic indicators" filetype:ppt

- **link:** производить поиск внутри гиперссылок содержащих поисковый запрос. "link:" возвращает список страниц, имеющих ссылку на заданную. Например, введите "link:www.google.com" и увидите список сайтов имеющих ссылку на Google. Не обязательно вводить "http://"; Google проигнорирует эту часть текста даже если её ввести. "link:" отлично работает как с "глубокими" адресами, вроде "<http://www.raelity.org/apps/blosxom/>", так и с верхнеуровневыми URL, такими как "raelity.org".

- **cache:** оператор демонстрирует версию страницы, которая существовала, когда она индексировалась Google. URL страницы указывается сразу после двоеточия. "cache:" ищет копию страницы проиндексированной Google даже если страница уже недоступна по оригинальному URL или её содержимое полностью изменилось. Например:

cache:www.yahoo.com

- **intitle:** производить поиск внутри названия документа. "intitle:" ограничивает поиск до заглавий страниц (titles). Вариации, "allintitle:" ищет страницы в заголовки которых находятся все слова поиска. Например:

intitle:"george bush"

allintitle:"money supply" economics

- **inurl:** искать внутри URL документа. "inurl:" ограничивает поиск до адресов (URL) страниц. Команда хороша для поиска страниц помощи и поиска, так как они имеют довольно стабильную структуру. "allinurl:" вариация, которая ищет все введённые слова в URL. Например:

inurl:help

allinurl:search help

- **intext:** ведёт поиск только по тексту страниц (т.е. игнорирует текст ссылок, URL, и заглавий). Есть вариация "allintext:", но она плохо ладит с другими командами. Например:

intext:"yahoo.com"

intext:html

- **inanchor:** ищет текст в якорях ссылок на страницах (anchors). Якори ссылок – это текст описания ссылки. Например, во фрагменте кода HTML [[href="http://www.oreilly.com"](http://www.oreilly.com) O'Reilly and Associates] якорем ссылки является "O'Reilly and Associates." Пример:

inanchor:"tom peters"

- **related:** находит страницы, похожие на запрашиваемую. Например, поиск "related:google.com" даст множество поисковиков, включая HotBot, Yahoo!, and Northern Light. Аналогично:

related:www.yahoo.com

related:www.cnn.com

- **info:** предоставляет ссылки на более подробную информацию о запрошенном URL. Информация включает ссылки на кэш URL, список страниц имеющих ссылки на данную, страницы, связанные с данной, страницы, содержащие данный URL. Например:

info:www.oreilly.com

info:www.nytimes.com/technology

Такие операторы можно использовать как по отдельности, так и в различных сочетаниях, в том числе – с различными ключевыми словами. Но некоторые из них

отлично работают в сочетании друг с другом, некоторые друг другу мешают, а некоторые просто друг с другом не работают.

Индивидуальные, не сочетающиеся с другими, команды: `gphonebook:`, `bphonebook:`, `phonebook:`, `link:`. Остальные спец. команды можно смешивать как угодно.

Карта сайта

Чтобы выявить каждую страницу на сайте, Google сканирует его, используя оператор `"site:"` и дополнительные ключевые слова, которые должны содержаться на **каждой** странице сайта.

Например, составим запрос такого вида: **`site:http://www.microsoft.com microsoft`**. Этот запрос выполняет поиск по слову «microsoft» в пределах сайта `http://www.microsoft.com`.

Как много страниц на сервере Microsoft содержат слово «Microsoft»? Для выяснения этого вопроса надо иметь в виду, что Google исследует не только содержание страниц, но также их название и URL. Слово «Microsoft» стоит в URL каждой страницы `http://www.microsoft.com`.

Таким образом - единственным запросом можно инициировать обработку каждой страницы на сайте Microsoft, проиндексированной Google.

Нахождение листинга директории

Листинг директории представляет собой список файлов и директорий удаленного сервера в окне браузера. Такие листинги открывают широкие возможности для углубленного сбора информации. Как правило: такие страницы директорий имеют в Title и теле страницы выражение «Index Of». Отсюда очевидно и строение запроса для поиска таких листингов - это «`intitle:index.of`». В результате такого запроса будут найдены страницы со словом «index of» в разделе Title документа.

К сожалению – этот запрос вернет слишком большое число страниц не по теме, к примеру, страницы вида:

- Index of Native American Resources on the Internet
- LibDex—Worldwide index of library catalogues
- Iowa State Entomology Index of Internet Resources

Исходя из названий найденных документов, очевидно, что эти страницы не соответствуют заданному запросу и вряд ли окажутся искомыми списками директорий.

Следующие запросы обеспечат более точные результаты:

`intitle:index.of "parent directory"`

`intitle:index.of name size`

Такие запросы более точно выдадут то, что нам нужно, поскольку ориентированы не только на фразу «index of» в Title страницы, но и на ключевые слова, всегда имеющиеся в листингах директорий: «parent directory», «name», «size».

Определение версии WEB-сервера

Точная версия программного обеспечения web сервера – это один из элементов, необходимых администратору для точной настройки Web-сайта. Если непосредственно соединиться с сервером, то HTTP (web) заголовки (headers) этого сервера предоставят нужную информацию. Однако можно получить эту информацию **из кэша Google без всякого соединения с сервером**. Такой метод основан на использовании списка директорий.

Список файлов директории включает имя серверного софта и его версию.

Выглядит такой запрос просто: «`intitle:index.of server.at`»

Он основан на содержании фразы «**index of**» в разделе **title** страницы директории и фразы «**server.at**», содержащейся в конце любого листинга директории. К примеру, так выглядит запрос, определяющий версию сервера aol.com:

«**intitle:index.of server.atsite:aol.com**».

Использование Google в качестве сканера CGI директорий

Для выполнения подобной задачи, CGI сканер изначально знает - какие именно директории нужно искать на сервере. Как правило - это директории, в которых располагаются файлы данных и имеют вид, подобный представленным ниже:

/cgi-bin/cgiemail/uargg.txt
/random_banner/index.cgi
/random_banner/index.cgi
/cgi-bin/mailview.cgi
/cgi-bin/maillist.cgi
/cgi-bin/userreg.cgi
/iissamples/ISSamples/SQLQHit.asp
/iissamples/ISSamples/SQLQHit.asp
/SiteServer/admin/findvserver.asp
/scripts/cphost.dll
/cgi-bin/finger.cgi

Зная синтаксис требуемых директорий, а также владея техникой поиска, изложенной выше, можно использовать Google как CGI сканер.

- Например, поиск в Google следующего вида:
allinurl:/random_banner/index.cgi вернет документы с адресами страниц конкретных программ генерации рекламных баннеров.

Использование Google как внутренней поисковой системы Web-сайта.

При поиске информации на серверах можно использовать не только их "родные" формы поиска, но и Google. Например? Поисковый запрос:

""george bush" site:nytimes.com" - поиск статей про Дж.Буша на сайте Нью Йорк Таймс.

Примеры поиска.

Найти независимые ресурсы (поиск числа, продолжения фразы, описания услуги, и т.д.):

1. число

1.1. статистику (базы данных или статьи) по динамике рынка услуг мобильной связи в России

2. продолжение фразы (рекламный слоган фирмы или обращение к клиентам)

2.1. ...себе ... засохнуть...

3. подробное описание услуги/товара (по категории)

3.1. мобильные телефоны с MP3

4. автора (по названию статьи или книги)

4.1. Windows 2000 для чайников

5. Наиболее популярный ресурс (по строке)

5.1. Интернет-трейдинг

6. WEB-представительство ВУЗа, предлагающего самое дешевое второе образование (по специальности)

6.1. WEB-дизайнер

7. WEB-представительство фирм, предлагающих вакансии с наибольшей оплатой (по вакансиям)

- 7.1. IT-менеджер
- 8. **WEB-представительство организации** (по названию)
- 8.1. Всемирный Банк
- 9. **WEB-представительство российской фирмы-лидера рынка** (по названию вида деятельности)
- 9.1. Экспортер древесины в ЕЭС
- 10. **WEB-представительство или персональную страницу** (ICQ, e-mail, домашний телефон или адрес) человека (по ФИО или должности)
- 10.1. Николай Фоменко (шоумен)

Поисковая система Alta Vista.

У каждой поисковой системы существует свой язык запросов, который определяет правила, в соответствии с которыми формулируются запросы на поиск информации.

В классификационных и словарных ИПС запрос составляется на основе ключевых слов, которые являются наиболее яркой характеристикой искомой информации (по сути, без этих слов данная информация обойтись не может). Лучше, если эти ключевые слова имеют специфический смысл, присущий только искомому информационному материалу, отличающему данный материал от всех остальных.

Ключевые слова могут набираться на разных регистрах клавиатуры - в зависимости от этого поисковая машина будет по-разному проводить поиск.

Поисковая система AltaVista является одной из самых информационно насыщенных. Обратиться к ней можно по адресу: <http://www.altavista.com>

Рассмотрим правила составления поисковых запросов, использования операторов и команд в языке запросов системы AltaVista:

1) Запрос на поиск информации (поисковое предписание) представляет собой поисковый образ.

2) Поисковый образ может состоять из одного или нескольких ключевых слов.

3) В зависимости от способа соединения ключевых слов в поисковом запросе различают простые и сложные запросы.

4) Сложный запрос отличается от простого тем, что в нем можно указать дату создания искомого документа (чтобы выделить материалы, имеющие последнее обновление после указанной даты), специальную логику поиска (определяемую использованием операторов AND, OR, NOT, NEAR), выбрать один из трех вариантов упорядочивания результатов поиска при их выводе: “только в качестве итога”, “компактная форма”, и “стандартная форма” (последняя используется по умолчанию), и использовать круглые скобки для выделения логически самостоятельных частей запроса.

5) Наличие в ключевом слове заглавной буквы заставит поисковую машину при простом поиске искать слова именно с таким написанием, как в запросе. Если же заглавные буквы не использовались, то поисковая машина учитывает любые варианты написания этих слов. Например, если поисковое предписание состоит из одного слова Computer, будут найдены информационные материалы, содержащие это слово именно в таком начертании. Если же это слово не будет содержать заглавных букв, то при поиске будут учитываться слова в таких начертаниях, как computer, COMPUTER, COMPuter, и др. Необходимо учитывать, что при использовании поискового образа, состоящего только из одного слова computer, AltaVista предоставляет около 2000 ссылок. Просмотреть такое количество ссылок практически невозможно, а значит, информационный поиск нельзя считать эффективным (при правильно составленном запросе необходимая информация находится в числе первых двух десятков ссылок).

6) В том случае, если неизвестно правильное написание слова, или интерес представляет множество однокоренных слов, используется оператор неопределенности - “*” (звездочка). Поставив этот символ после любой последовательности букв (не менее трех), влияние которых необходимо учесть при поиске, можно произвести широкий

поиск, при котором ключевое слово будет модифицироваться: поиск будет вестись как для жестко указанной до звездочки совокупности букв, так и для слов, содержащих любые буквы (числом до 5) вместо звездочки. Например, если указать ключевое слово `comp*`, то при поиске будут учитываться, как ключевые - `computer`, `computers`, `compute`, и др.

7) Для соединения нескольких ключевых слов могут использоваться операторы “пробел”, “кавычки”, логические операторы “+”, “-”, AND, OR, NOT, NEAR.

8) Оператор “пробел” соединяет слова в поисковом предписании таким образом, что для поиска каждое из этих слов используется отдельно. При этом, порядок слов в запросе не имеет значения. В процессе поиска учитывается только расстояние каждого слова от начала документа и частота его использования в документе.

9) Оператор “кавычки” соединяет слова так, что они образуют фразу, в которой все указанные в предписании слова в документе стоят рядом друг с другом и в той же последовательности, как это указано в предписании. Поэтому, если задать поисковое предписание в виде слов “personal computer” и в виде “computer personal”, то результаты поиска будут разными.

10) Оператор “+”, соединяющий слова, сообщает поисковой машине, что в документе необходимо искать основное слово (первое), но документ надо показывать в результате поиска только если далее в тексте встречаются остальные слова из поискового предписания. Оператор ставится непосредственно перед каждым второстепенным словом. Например, по поисковому образу:

`computer +personal +digital`

будет вестись поиск основного слова `computer`, но текст будет считаться актуальным только если в нем встречаются так же слова `personal` и `digital`.

11) Оператор “-”, стоящий перед словом, обозначает, что основное слово должно использоваться в тексте без второстепенного. Например, поисковое предписание `computer -personal` сообщает поисковой машине, что надо искать основное слово `computer`, но в тексте не должно встречаться слово `personal` (т.е. интересуют материалы о компьютерах, но не персональных).

12) Операторы AND, OR, NOT, NEAR используются в сложных запросах.

13) Оператор AND (вместо него можно использовать символ `&`) определяет, что соединяемые им слова должны встречаться вместе (т.е. в простых запросах он эквивалентен знаку “+”).

14) Оператор OR (вместо него можно использовать знак “|”) определяет, что соединяемые им слова независимы друг от друга (в простых запросах он эквивалентен пробелу).

15) Оператор NOT обозначает отрицание (в простых запросах он эквивалентен знаку “-”).

16) Оператор NEAR (вместо него можно использовать символ “~”) определяет, что в искомом тексте указанное им ключевое слово отстоит от основного не далее, чем на 10 слов (например, в поисковом предписании:

`провайдер* NEAR “очень дешево”`

предусматривается, что в искомом тексте слово “провайдер” и словосочетание “очень дешево” находятся не в разных концах текста, а рядом друг с другом - между ними может находиться не более 10 слов).

Специальные возможности поисковой системы.

Для ограничения поиска в AltaVista используются специальные команды (тэги): `anchor`, `applet`, `title`, `url`, `host`, `link`, `image`, `from`, `subject`.

1) Команда `anchor` позволяет найти в Сети слово, содержащееся в “теле” ссылки. Для этого после команды `anchor` через двоеточие указывается искомое слово. Например, поисковый образ содержит:

anchor:home

По этому запросу будет найдено все множество страниц, содержащих внутри ссылок слово home, в том числе - и в такой ссылке: "If you would like go home, press here".

2) Команда `applet` позволяет найти заданный названием модуль Java. Например, если модуль Java называется word, то найти его можно, записав поисковый образ: `applet:word`.

3) Команда `title` используется в том случае, если искомое слово находится в заголовке текста. Например, по запросу вида:

`title:links`

будут найдены документы, содержащие слово links в заглавии, в том числе текст с заглавием "Cool Links".

4) Команда `url` предписывает искать url-адрес, содержащий заданное слово. Например, если неизвестно, в каком корневом домене находится host-компьютер МЭСИ, можно задать поисковое предписание: `url:mesi`. Среди множества адресов с таким словом будет и адрес `http://www.mesi.ru/`.

5) Команда `host` позволяет узнать, какие Web-сайты есть на заданном host-компьютере. Например, для того, чтобы узнать, какие сайты есть на хосте `www.intel.ru` необходимо набрать запрос: `host:intel.ru`. Если же в запросе указать только часть имени, то в результате поиска будут найдены сайты, имеющие другие адреса, но содержащие заданную часть имени.

Используя эту команду, можно вести поиск в заданной стране. Например, по запросу `host:*.ru +kreml` будет найдена информация о Московском, Рязанском и других Кремлях. При этом нужно помнить, что поиск ведется только для сайтов, зарегистрированных в поисковой системе AltaVista, другие сайты ей недоступны.

6) Команда `link` позволяет найти адреса страниц (сайтов), содержащих ссылку на конкретную (заданную в поисковом образе) Web-страницу. Например, для того, чтобы узнать, кто ссылается на сайт `www.mesi.ru` необходимо задать предписание: `link:www.mesi.ru`. Результатом будет список страниц, на которых содержатся ссылки на сайт `mesi.ru`.

7) Команда `image` позволяет найти иллюстрацию в Internet. Для этого надо знать название файла, в котором она хранится. Формат команды тот же.

8) Команда `from` позволяет искать в телеконференциях Usenet почтовое сообщение, отправленное конкретным человеком, имя которого указывается после двоеточия в команде. Например: `from:Иван +Федоров` (или `Ivan +Fedorov`).

9) Команда `subject` позволяет искать сообщения в телеконференциях Usenet на конкретную, заданную в поисковом предписании тему.

Поисковая система AltaVista может работать (и вести поиск) на разных языках, в том числе и на русском.

Описанные принципы управления поисковой системой во многом аналогичны используемым и в других поисковых системах.

Приложение 4. Исходные данные для задачи «Ирисы Фишера».

Полный текст исходных данных можно найти в пакете Statistica фирмы StatSoft в файле: ...\Examples\Datasets\ Irisdat.sta и IrisSNN.sta:

Данные: Irisdat.sta (5v * 150с)					
Fisher (1936) iris data: length & width of sepals and petals, 3 types of Iris					
	1	2	3	4	5
	SEPALLEN	SEPALWID	PETALLEN	PETALWID	IRISTYPE
1	5,0	3,3	1,4	0,2	SETOSA
2	6,4	2,8	5,6	2,2	VIRGINIC
3	6,5	2,8	4,6	1,5	VERSICOL
4	6,7	3,1	5,6	2,4	VIRGINIC
5	6,3	2,8	5,1	1,5	VIRGINIC
6	4,6	3,4	1,4	0,3	SETOSA
7	6,9	3,1	5,1	2,3	VIRGINIC
8	6,2	2,2	4,5	1,5	VERSICOL
9	5,9	3,2	4,8	1,8	VERSICOL
10	4,6	3,6	1,0	0,2	SETOSA
11	6,1	3,0	4,6	1,4	VERSICOL

Поскольку выходная переменная - номинальная с тремя состояниями, некоторые программы (например, Statistica Neural Network) будут представлять выходную переменную в виде трёх выходных элементов. Каждое из трех возможных номинальных значений выражается высоким выходом одного из этих элементов и низким - двух оставшихся (например, типу *Setosa* может соответствовать выходной набор {0,97; 0,02; 0,01}).

Всего файл содержит 150 реализаций, имеющих структуру:

№	SEPALLEN	SEPALWID	PETALLEN	PETALWID	IRISTYPE
	- Длина чашел	- Ширина чашел	- Длина лепестка	- Ширина лепестка	- Тип ириса

При решении поставленной задачи:

1. 130 реализаций из файла исходных данных, перемешанных случайным образом, предъявить сети в качестве обучающего (training) набора

	SEPALLEN	SEPALWID	PETALLEN	PETALWID	IRISTYPE
1	5,0	3,3	1,4	0,2	SETOSA
2	6,4	2,8	5,6	2,2	VIRGINIC
3	6,5	2,8	4,6	1,5	VERSICOL
4	6,7	3,1	5,6	2,4	VIRGINIC
5	6,3	2,8	5,1	1,5	VIRGINIC
6	4,6	3,4	1,4	0,3	SETOSA
7	6,9	3,1	5,1	2,3	VIRGINIC
8	6,2	2,2	4,5	1,5	VERSICOL
9	5,9	3,2	4,8	1,8	VERSICOL
10	4,6	3,6	1,0	0,2	SETOSA
11	6,1	3,0	4,6	1,4	VERSICOL
12	6,0	2,7	5,1	1,6	VERSICOL

13	6,5	3,0	5,2	2,0	VIRGINIC
14	5,6	2,5	3,9	1,1	VERSICOL
15	6,5	3,0	5,5	1,8	VIRGINIC
16	5,8	2,7	5,1	1,9	VIRGINIC
17	6,8	3,2	5,9	2,3	VIRGINIC
18	5,1	3,3	1,7	0,5	SETOSA
19	5,7	2,8	4,5	1,3	VERSICOL
20	6,2	3,4	5,4	2,3	VIRGINIC
21	7,7	3,8	6,7	2,2	VIRGINIC
22	6,3	3,3	4,7	1,6	VERSICOL
23	6,7	3,3	5,7	2,5	VIRGINIC
24	7,6	3,0	6,6	2,1	VIRGINIC
25	4,9	2,5	4,5	1,7	VIRGINIC
26	5,5	3,5	1,3	0,2	SETOSA
27	6,7	3,0	5,2	2,3	VIRGINIC
28	7,0	3,2	4,7	1,4	VERSICOL
29	6,4	3,2	4,5	1,5	VERSICOL
30	6,1	2,8	4,0	1,3	VERSICOL
31	4,8	3,1	1,6	0,2	SETOSA
32	5,9	3,0	5,1	1,8	VIRGINIC
33	5,5	2,4	3,8	1,1	VERSICOL
34	6,3	2,5	5,0	1,9	VIRGINIC
35	6,4	3,2	5,3	2,3	VIRGINIC
36	5,2	3,4	1,4	0,2	SETOSA
37	4,9	3,6	1,4	0,1	SETOSA
38	5,4	3,0	4,5	1,5	VERSICOL
39	7,9	3,8	6,4	2,0	VIRGINIC
40	4,4	3,2	1,3	0,2	SETOSA
41	6,7	3,3	5,7	2,1	VIRGINIC
42	5,0	3,5	1,6	0,6	SETOSA
43	5,8	2,6	4,0	1,2	VERSICOL
44	4,4	3,0	1,3	0,2	SETOSA
45	7,7	2,8	6,7	2,0	VIRGINIC
46	6,3	2,7	4,9	1,8	VIRGINIC
47	4,7	3,2	1,6	0,2	SETOSA
48	5,5	2,6	4,4	1,2	VERSICOL
49	7,2	3,2	6,0	1,8	VIRGINIC
50	4,8	3,0	1,4	0,3	SETOSA
51	5,1	3,8	1,6	0,2	SETOSA
52	6,1	3,0	4,9	1,8	VIRGINIC
53	4,8	3,4	1,9	0,2	SETOSA
54	5,0	3,0	1,6	0,2	SETOSA
55	5,0	3,2	1,2	0,2	SETOSA
56	6,1	2,6	5,6	1,4	VIRGINIC
57	6,4	2,8	5,6	2,1	VIRGINIC

58	4,3	3,0	1,1	0,1	SETOSA
59	5,8	4,0	1,2	0,2	SETOSA
60	5,1	3,8	1,9	0,4	SETOSA
61	6,7	3,1	4,4	1,4	VERSICOL
62	6,2	2,8	4,8	1,8	VIRGINIC
63	4,9	3,0	1,4	0,2	SETOSA
64	5,1	3,5	1,4	0,2	SETOSA
65	5,6	3,0	4,5	1,5	VERSICOL
66	5,8	2,7	4,1	1,0	VERSICOL
67	5,0	3,4	1,6	0,4	SETOSA
68	4,6	3,2	1,4	0,2	SETOSA
69	6,0	2,9	4,5	1,5	VERSICOL
70	5,7	2,6	3,5	1,0	VERSICOL
71	5,7	4,4	1,5	0,4	SETOSA
72	5,0	3,6	1,4	0,2	SETOSA
73	7,7	3,0	6,1	2,3	VIRGINIC
74	6,3	3,4	5,6	2,4	VIRGINIC
75	5,8	2,7	5,1	1,9	VIRGINIC
76	5,7	2,9	4,2	1,3	VERSICOL
77	7,2	3,0	5,8	1,6	VIRGINIC
78	5,4	3,4	1,5	0,4	SETOSA
79	5,2	4,1	1,5	0,1	SETOSA
80	7,1	3,0	5,9	2,1	VIRGINIC
81	6,4	3,1	5,5	1,8	VIRGINIC
82	6,0	3,0	4,8	1,8	VIRGINIC
83	6,3	2,9	5,6	1,8	VIRGINIC
84	4,9	2,4	3,3	1,0	VERSICOL
85	5,6	2,7	4,2	1,3	VERSICOL
86	5,7	3,0	4,2	1,2	VERSICOL
87	5,5	4,2	1,4	0,2	SETOSA
88	4,9	3,1	1,5	0,2	SETOSA
89	7,7	2,6	6,9	2,3	VIRGINIC
90	6,0	2,2	5,0	1,5	VIRGINIC
91	5,4	3,9	1,7	0,4	SETOSA
92	6,6	2,9	4,6	1,3	VERSICOL
93	5,2	2,7	3,9	1,4	VERSICOL
94	6,0	3,4	4,5	1,6	VERSICOL
95	5,0	3,4	1,5	0,2	SETOSA
96	4,4	2,9	1,4	0,2	SETOSA
97	5,0	2,0	3,5	1,0	VERSICOL
98	5,5	2,4	3,7	1,0	VERSICOL
99	5,8	2,7	3,9	1,2	VERSICOL
100	4,7	3,2	1,3	0,2	SETOSA
101	4,6	3,1	1,5	0,2	SETOSA
102	6,9	3,2	5,7	2,3	VIRGINIC

103	6,2	2,9	4,3	1,3	VERSICOL
104	7,4	2,8	6,1	1,9	VIRGINIC
105	5,9	3,0	4,2	1,5	VERSICOL
106	5,1	3,4	1,5	0,2	SETOSA
107	5,0	3,5	1,3	0,3	SETOSA
108	5,6	2,8	4,9	2,0	VIRGINIC
109	6,0	2,2	4,0	1,0	VERSICOL
110	7,3	2,9	6,3	1,8	VIRGINIC
111	6,7	2,5	5,8	1,8	VIRGINIC
112	4,9	3,1	1,5	0,1	SETOSA
113	6,7	3,1	4,7	1,5	VERSICOL
114	6,3	2,3	4,4	1,3	VERSICOL
115	5,4	3,7	1,5	0,2	SETOSA
116	5,6	3,0	4,1	1,3	VERSICOL
117	6,3	2,5	4,9	1,5	VERSICOL
118	6,1	2,8	4,7	1,2	VERSICOL
119	6,4	2,9	4,3	1,3	VERSICOL
120	5,1	2,5	3,0	1,1	VERSICOL
121	5,7	2,8	4,1	1,3	VERSICOL
122	6,5	3,0	5,8	2,2	VIRGINIC
123	6,9	3,1	5,4	2,1	VIRGINIC
124	5,4	3,9	1,3	0,4	SETOSA
125	5,1	3,5	1,4	0,3	SETOSA
126	7,2	3,6	6,1	2,5	VIRGINIC
127	6,5	3,2	5,1	2,0	VIRGINIC
128	6,1	2,9	4,7	1,4	VERSICOL
129	5,6	2,9	3,6	1,3	VERSICOL

2. 10 случайно отобранных реализаций, не включённых в обучающий набор, предъявить сети в качестве тестового (testing) набора. Например, такие:

1	6,9	3,1	4,9	1,5	VERSICOL
2	6,4	2,7	5,3	1,9	VIRGINIC
3	6,8	3,0	5,5	2,1	VIRGINIC
4	5,5	2,5	4,0	1,3	VERSICOL
5	4,8	3,4	1,6	0,2	SETOSA
6	4,8	3,0	1,4	0,1	SETOSA
7	4,5	2,3	1,3	0,3	SETOSA
8	5,7	2,5	5,0	2,0	VIRGINIC
9	5,7	3,8	1,7	0,3	SETOSA
10	5,1	3,8	1,5	0,3	SETOSA

3. 10 случайно отобранных реализаций, не включённых в обучающий набор, предъявить обученной сети для распознавания. Например, такие:

1	5,5	2,3	4,0	1,3	VERSICOL
2	6,6	3,0	4,4	1,4	VERSICOL
3	6,8	2,8	4,8	1,4	VERSICOL
4	5,4	3,4	1,7	0,2	SETOSA
5	5,1	3,7	1,5	0,4	SETOSA
6	5,2	3,5	1,5	0,2	SETOSA
7	5,8	2,8	5,1	2,4	VIRGINIC
8	6,7	3,0	5,0	1,7	VERSICOL
9	6,3	3,3	6,0	2,5	VIRGINIC
10	5,3	3,7	1,5	0,2	SETOSA

Литература.

1. А.А. Ежов, С.А. Шумский Нейрокомпьютинг и его применения в экономике и бизнесе <http://www.intuit.ru/department/expert/neurocomputing/>
2. Brain Maker Professional, Neural Network Simulation Software. User Guide and Reference Manual. California Scientific Software, 1993.
3. Моделирование нейронных сетей с обратным распространением ошибки: Методическое пособие к лабораторным работам /Сост. Э.Ю. Кирсанов, А.Г. Пиянин. Казань: Изд-во гос. техн. ун-та, 1999. 40 с.
4. Deductor «О системе», Система: BaseGroup\Deductor\Bin\Deductor.chm
5. Сергей Блинов, BrainMaker - прогнозирование на финансовых рынках blis@aha.ru опубликовано в <http://old.osp.ru/text/print/302/179543.html>
6. Методы нейроинформатики. Сборник научных трудов. Под редакцией д.физ-мат. наук А.Н. Горбаня
7. Андрей Масалович «НЕЙРОННАЯ СЕТЬ - ОРУЖИЕ ФИНАНСИСТА», http://www.tora-centre.ru/library/ns/nn_fin2.htm.
8. Миркес Е.М. Нейроинформатика Учебное пособие. www.softcraft.ru.
9. Гаврилов Андрей Владимирович [Лабораторный практикум по нейронным сетям](#) (Новосибирский Государственный Технический Университет, Кафедра Вычислительной Техники).
10. Гилев С.Е., Коченов Д.А., Миркес Е.М., Россиев Д.А. Контрастирование, оценка значимости параметров, оптимизация их значений и их интерпретация в нейронных сетях // Доклады III Всероссийского семинара “Нейроинформатика и ее приложения”. – Красноярск: Изд-во КГТУ, 1995.- С.66-78.
11. Россиев А. А. Результаты практического применения программных продуктов для решения прикладных задач. (www.ict.edu.ru/ft/004239/avtorossiev.pdf)
12. Демонстрационные примеры пакета Статистика - ... \StatSoft\STATISTICA \Examples\Datasets\Irisdat.sta.
13. Интернет – ресурс http://www.statsoft.ru/statportal/tabID_32/MId_141/ModelID_0/PageID_192/DesktopDefault.aspx
14. Уоссермен Ф. Нейрокомпьютерная техника : теория и практика. М. : Мир, 1992. – 240 с.
15. А.П.Пятибратов, Л.П Гудыно, А.А.Кириченко «Вычислительные системы, сети и телекоммуникации», ИНФРА-М, 2008. – 736с.:ил.
16. Нейронные сети Statistica Neural Networks. Методология и технологии современного анализа данных. Под ред. Боровикова В.П., М., Горячая линия – Телеком, 2008.
17. Н. М. Абдикеев «Проектирование интеллектуальных систем в экономике», М., 2003.
18. Методы добычи данных. Internet-ресурс – <http://www.statsoft.ru>

Сетевое электронное издание учебного пособия.

Кириченко А.А.

профессор кафедры архитектуры программных систем Федерального государственного автономного образовательного учреждения высшего профессионального образования “Национальный исследовательский университет “Высшая школа экономики” при правительстве РФ”.

**«Нейропакеты – современный интеллектуальный инструмент
исследователя»**

Текст публикуется в авторской редакции.

ISBN 978-5-9904911-1-3

