ПЕРВОЕ ВЫСШЕЕ ТЕХНИЧЕСКОЕ УЧЕБНОЕ ЗАВЕДЕНИЕ РОССИИ

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение высшего образования
САНКТ-ПЕТЕРБУРГСКИЙ ГОРНЫЙ УНИВЕРСИТЕТ

УТВЕРЖДАЮ

Руководитель ОПОП ВО
доцент Ю.В. Ильюшин

МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ДЛЯ ПРОВЕДЕНИЯ ПРАКТИЧЕСКИХ ЗАНЯТИЙ ПО ДИСЦИПЛИНЕ
ТЕХНОЛОГИИ ОБРАБОТКИ ИНФОРМАЦИИ

Уровень высшего образования:	Подготовка кадров высшей квалификации
Направление подготовки:	09.06.01 Информатика и вычислительная техника
Направленность (профиль):	Системный анализ, управление и обработка информации (промышленность)
Форма обучения:	очная
Нормативный срок обучения:	4 года
Составитель:	д.т.н., профессор В.Я. Трофимец

Санкт-Петербург

ДОКУМЕНТ ПОДПИСАН УСИЛЕННОЙ КВАЛИФИЦИРОВАННОЙ ЭЛЕКТРОННОЙ ПОДПИСЬЮ
Сертификат: 00F3 503F 985D 6537 76D4 6643 BD9B 6D2D 1C
Владелец: Пашкевич Наталья Владимировна
Действителен: с 27.12.2022 по 21.03.2024

Содержание

1. Цель занятий

Цель проведения практических занятий – формирование умений и навыков, позволяющих эффективно применять информационные технологии в процессе решения учебных и профессионально-ориентированных задач.

Поставленная цель достигается путем выполнения студентами практических заданий с использованием методических разработок и контроля выполнения работ преподавателем.

2. Охрана труда и техника безопасности

Организация безопасной работы студентов при выполнении практических и лабораторных работ производится в соответствии с со следующими Государственными стандартами: ГОСТ 12.1.030-81 ССБТ «Электробезопасность. Защитное заземление, зануление», ГОСТ 12.2.032-78 ССБТ «Рабочее место при выполнении работ сидя. Общие эргономические требования».

Перед выполнением практических занятий с использованием компьютеров все студенты проходят инструктаж по технике безопасности, о чем делается запись в соответствующем журнале, которая подтверждается собственноручными подписями студентов и лицом, проводившим инструктаж.

В процессе выполнения занятий при обнаружении неисправностей в лабораторной установке следует немедленно прекратить работу и сообщить об этом преподавателю.

Запрещается:

– находиться в помещении в верхней одежде;

– выполнять работу в отсутствии преподавателя или дежурного лаборанта;

– класть сумки, одежду и другие вещи на столы и лабораторную технику.

Студенты, не соблюдающие правила техники безопасности, отстраняются от работы.

3. Библиографический список

3.1. Основная литература

1. Ясницкий, Л.Н. Интеллектуальные системы [Электронный ресурс]: учебник / Л.Н. Ясницкий. – эл. изд. – М.: Лаборатория знаний, 2016. – 224 с.

(http://biblioclub.ru/index.php?page=book_red&id=119452)

2. Статистические методы анализа данных [Электронный ресурс]: учебник / Л.И. Ниворожкина, С.В. Арженовский, А.А. Рудяга [и др.]; под общ. ред. д-ра экон. наук, проф. Л.И. Ниворожкиной. – М.: РИОР: ИНФРА-М, 2016. – 333 с.

(http://znanium.com/catalog.php?bookinfo=556760)

3. Информационные аналитические системы [Электронный ресурс]: учебник / Т.В. Алексеева, Ю.В. Амириди, В.В. Дик и др.; под ред. В.В. Дика. – М.: МФПУ Синергия, 2013. – 384 с. (http://znanium.com/catalog.php?bookinfo=451186)

3.2. Дополнительная литература

1. Романов, А.Н. Советующие информационные системы в экономике [Электронный ресурс]: учеб. пособие / А.Н. Романов, Б.Е. Одинцов. – М.: ИНФРА-М, 2017. – 485 с.

(http://znanium.com/catalog.php?bookinfo=854392)

2. Вейнберг Р.Р. Интеллектуальный анализ данных и систем управления бизнес-правилами в телекоммуникациях [Электронный ресурс]: монография / Р.Р. Вейнберг. – М.: НИЦ ИНФРА-М, 2016. – 173 с. (http://znanium.com/catalog.php?bookinfo=520998)

3. Нестеров, С.А. Интеллектуальный анализ данных средствами MS SQLServer 2008 [Электронный ресурс]: учебное пособие / С.А. Нестеров. – 2-е изд., испр. – М.: Национальный Открытый Университет «ИНТУИТ», 2016. – 338 с.
(http://biblioclub.ru/index.php?page=book_red&id=429083&sr=1)
4. Жуковский, О.И. Информационные технологии и анализ данных [Электронный ресурс]: учебное пособие / О.И. Жуковский; Министерство образования и науки Российской Федерации, Томский Государственный Университет Систем Управления и Радиоэлектроники (ТУСУР). – Томск: Эль Контент, 2014. – 130 с.
(http://biblioclub.ru/index.php?page=book_red&id=480500&sr=1)
5. Карпузова, В.И. Информационные технологии в менеджменте [Электронный ресурс]: учебное пособие / В.И. Карпузова, Э.Н. Скрипченко, К.В. Чернышева, Н.В. Карпузова. – 2-е изд., доп. – М.: Вузовский учебник: НИЦ ИНФРА-М, 2014. – 301 с.
(http://znanium.com/catalog.php?bookinfo=410374)
6. Чубукова, И.А. Data Mining [Электронный ресурс]: учебное пособие / И.А. Чубукова. – 2-е изд., испр. – М.: Интернет-Университет Информационных Технологий, 2008. – 383 с. (http://biblioclub.ru/index.php?page=book_red&id=233055)

4. Содержание практических занятий

Практическое занятие № 1
Создание хранилища данных

Запустите программу Deductor Studio Academic. Для создания нового пустого хранилища данных или подключения к существующему перейдите на вкладку Подключения меню ***Вид***, щелкните правой кнопкой мыши и запустите **Мастер подключений** (рис. 1).

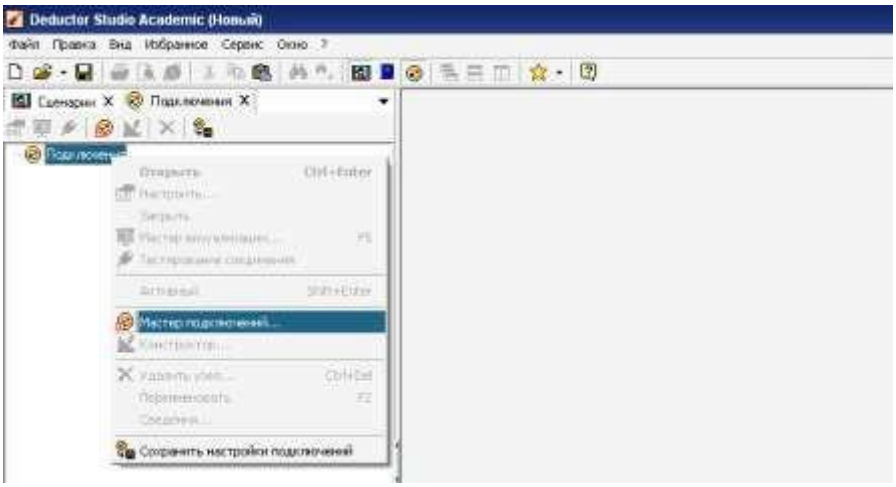


Рис. 1 Создание (подключение) хранилища данных

На первом шаге мастера следует выбрать тип источника (приемника) – Deductor Warehouse (рис. 2).

Рис. 2. Окно выбора типа подключения

На следующем шаге из единственно доступного в списке типа базы данных выберите Firebird. Задайте параметры базы данных, в которой будет создана физическая и логическая структура хранилища данных (рис. 3):

- база данных – *D:\farma.gdb* (или любой другой путь);
- логин – *sysdba*, пароль – *masterkey;*
- установите флажок **Сохранять пароль**.

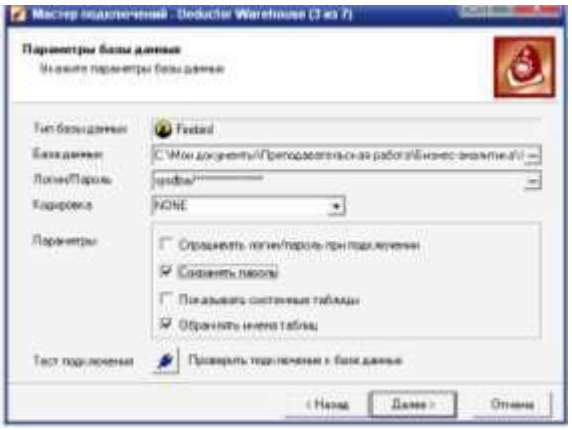


Рис. 3. Установка параметров базы данных

На следующей вкладке выберите последнюю версию для работы с ХД Deductor Warehouse. Нажмите кнопку **Создать файл базы данных с необходимой структурой *метаданных*** и по указанному ранее пути будет создан файл farma.gdb (появится сообщение о его успешном создании). Это и есть пустое хранилище данных.

Осталось выбрать визуализатор для подключения (здесь это **Сведения** и **Метаданные**) и задать имя, метку и описание нового хранилища.

Имя хранилища может быть введено только латинскими буквами. После нажатия кнопки Готово на дереве узлов подключений появится метка хранилища.

Для проверки доступа к новому хранилищу данных воспользуйтесь кнопкой **Тестирование соединения**. Если спустя некоторое время появится сообщение «Тестирование соединения прошло успешно», то хранилище готово к работе. Сохраните настройки подключений, нажав соответствующую кнопку.

Если соединение по какой-либо причине установить не удалось, то будет выдано сообщение об ошибке. В этом случае нужно проверить параметры подключения хранилища данных и при необходимости внести в них изменения (используйте для этого кнопку **Настроить подключение**).

Таким образом, создано пустое хранилище, в котором нет ни одного объекта (процесса, измерения, факта). Ранее мы спроектировали структуру хранилища данных аптеч-

ной сети. Осталось отразить ее в хранилище. Для этого предназначен **Редактор метаданных**, который вызывается нажатием кнопки **Открыть конструктор...** на вкладке **Подключения**. Выберите узел **Измерения**, щелкните правой кнопкой мыши, затем нажмите кнопку **Добавить** и создайте первое измерение **Код группы** со следующими параметрами:

- имя – GR_ID;
- метка – Группа.Код;
- тип данных – *целый.*

Метка – это семантическое название объекта хранилища данных, которое увидит пользователь, работающий с ХД.

Проделайте аналогичные действия для создания всех остальных измерений, взяв параметры из табл. 1.

Таблица 1

Параметры измерений

Измерение	Имя	Метка	Тип данных
1	2	3	4
Код группы	GR_ID	Группа.Код	Целый
Код товара	TV_ID	Товар.Код	Целый

Окончание таблицы 2.5

1	2	3	4
Код отдела	PART_D	Отдел.Код	Целый
Дата	S_DATE	Дата	Дата/время
Час покупки	S_HOUR	Час	Целый

В результате структура метаданных хранилища будет содержать пять измерений (рис. 4).

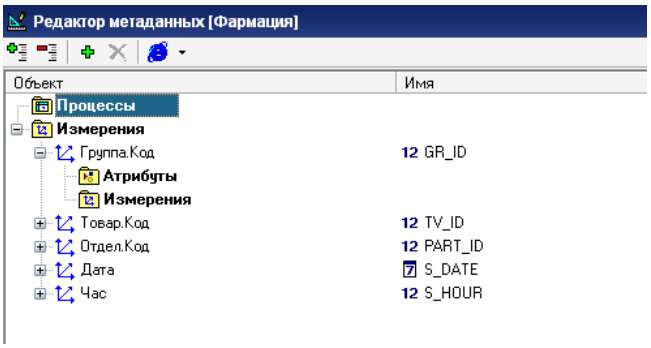


Рис. 4. Структура метаданных хранилища

К каждому измерению, кроме Дата и Час, добавьте текстовый атрибут. Для измерения Группа.Код это будет Группа.Наименование, для измерения Товар.Код – Товар.Наименование, для измерения Отдел.Код – Отдел.Наименование.

Каждое измерение может ссылаться на другое измерение, реализуя тем самым иерархию измерений. В нашем случае измерение *Товар.Код* ссылается на *Группа. Код.* Эту ссылку и установите путем простого добавления, а имя ссылки задайте *GR_ ID_ 1.*

После того как все измерения и ссылки на измерения созданы, приступайте к формированию процесса («снежинки»). Назовите его *Продажи* и добавьте в него ссылки на четыре существующих измерения: *Дата, Отдел.Код, Товар.Код, Час*. Кроме них, в процессе участвуют два факта: *Количество* и *Сумма,* причем первый – целочисленный, а второй – вещественный (рис. 5).

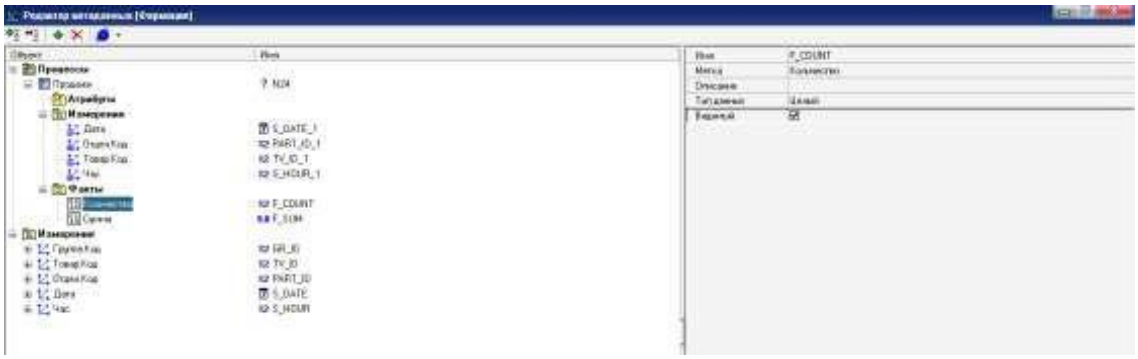


Рис. 5. Создание метаданных процесса

Практическое занятие № 2
Наполнение хранилища данных

После создания структуры хранилища данных оно представляет с собой пустой файл с настроенным семантическим слоем. В таком виде ХД готово к загрузке в него данных из внешних структурированных источников. Для этого необходимо написать соответствующий сценарий в Deductor Studio. Он должен выполнять следующие функции:

- импорт данных в Deductor Studio из базы данных, учетной системы или предопределенных файлов;
- опциональную предобработку данных, например очистку или преобразование формата;
- загрузку данных в измерения и процессы хранилища Deductor Warehouse.

В нашем примере исходными данными для ХД служат четыре текстовых файла: Группы товаров.txt, Товары.txt, Отделы.txt, Продажи.txt. Поэтому сценарий загрузки должен быть настроен на использование этих файлов в качестве источников данных (рис. 2.12).

При создании сценария необходимо строго придерживаться следующих правил.

□ Первыми загружаются все измерения, имеющие атрибуты. Только после загрузки всех измерений загружаются данные в процесс(ы).

□ Измерения нужно загружать, начиная с самого верхнего уровня иерархии и спускаясь ниже. Это крайне важно: в противном случае иерархия не будет создана.

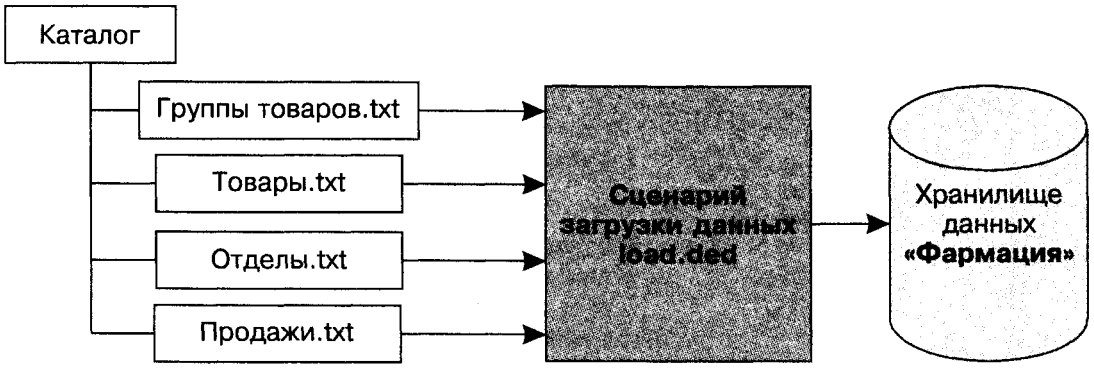


Рис. 1. Схема сценария загрузки

□ Допускается не загружать отдельно измерения, не имеющие атрибутов и не состоящие в иерархии измерений. Значения таких измерений можно создавать во время загрузки в процесс с помощью специальной опции.

Поясним второе правило (рис. 2). Измерение *Группа* находится в иерархии выше измерения *Товар*, поэтому последовательность загрузки измерений будет следующая: *Группа, Товар.*

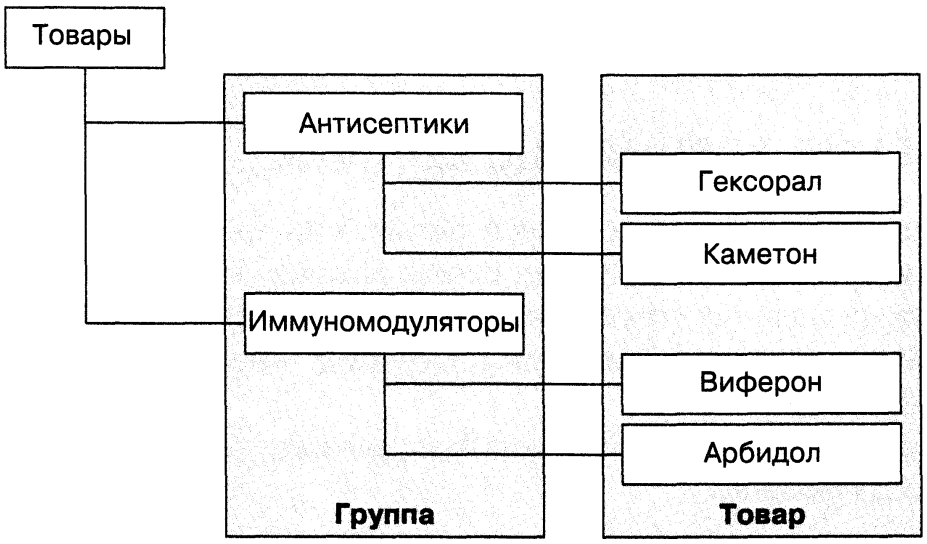


Рис. 2. Иерархия измерений

Импортируйте все четыре текстовых файла в Deductor в том порядке, как это показано на рис. 2.14. Для этого из контекстного меню или нажатием клавиши F6 нужно вызвать Мастер импорта, выбрать тип источника – текстовый файл и настроить параметры импорта. Последовательность создания узлов импорта должна быть такой, чтобы первыми следовали узлы импорта из файлов с таблицами измерений, и только в конце – таблица процесса Продажи.txt. Менять порядок веток сценария можно при помощи кнопок CTRL + ↑ и CTRL + ↓.

Покажем последовательность загрузки данных в измерение на примере первого измерения *Группа.Код*. Встав на первом узле, вызовите Мастер экспорта (контекстное меню или клавиша F8). Из списка типа приемников выберите Deductor Warehouse (рис. 2.15).

На следующей вкладке из списка доступных хранилищ выберите нужное под названием «Фармация». Далее требуется указать, в какое именно измерение будет загружаться информация. Это *Группа.Код*.

Осталось установить соответствие элементов объекта в хранилище данных с полями входного источника данных (то есть таблицы Группы товаров.txt). В случае, когда имена полей и (или) метки в семантическом слое хранилища данных совпадают, делать ничего не нужно (рис. 3).

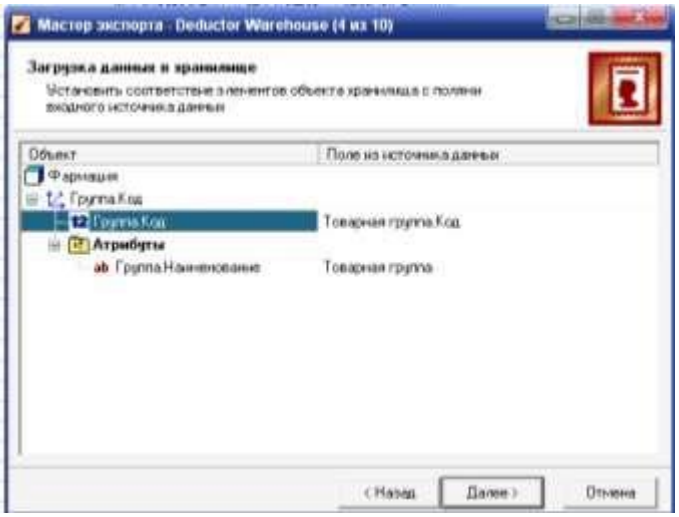


Рис. 5. Настройка соответствия полей

Нажатие кнопки **Пуск** на следующем шаге загрузит в измерение данные. При этом старые данные, если они были, обновятся.

Проделав аналогичные действия еще для двух измерений – *Отдел.Код, Товар.Код*, получим следующий сценарий (рис. 6).

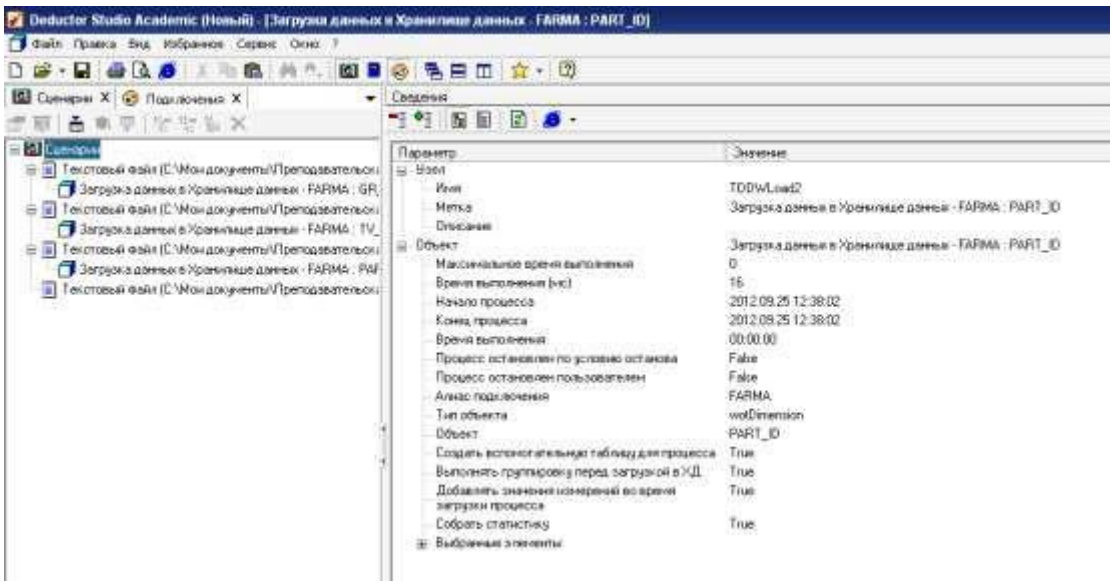


Рис. 6. Незаконченный сценарий загрузки данных в ХД

Загрузка измерений на этом заканчивается, несмотря на то что остались еще два измерения – *Дата* и *Час.* Но они не имеют атрибутов и не участвуют в иерархии, поэтому их значения можно загрузить на этапе экспорта в процесс.

Загрузите данные в процесс *Продажи.* В отличие от загрузки измерений, в **Мастере экспорта** появляются два специфических шага.

На одном из них нужно задать параметры контроля непротиворечивости данных в хранилище – указать измерения, по которым следует удалять данные из хранилища.

Выбирается действие, выполняемое в ситуации, когда в процесс загружается информация, которая совпадает по значениям из нескольких измерений. Может быть два варианта: удалить старые данные и загрузить новые или запретить удаление и оставить то, что было загружено ранее.

Поясним операцию удаления на примере (рис. 7). Допустим, в хранилище имеется процесс с двумя измерениями: *Клиент* и *Дата.* Необходимо загрузить в хранилище дан-

ные о продажах за последние два дня. Если в наборе данных, который мы загружаем, имеются все сведения о продажах за эти два дня, то можно указать: «Удалять данные по измерению и выбрать таким измерением *Дата*». Программа определит, что по измерению *Дата* в исходных данных всего два значения, а потом удалит из хранилища в процессе *Продажи* всю информацию за эти два дня и загрузит новую.

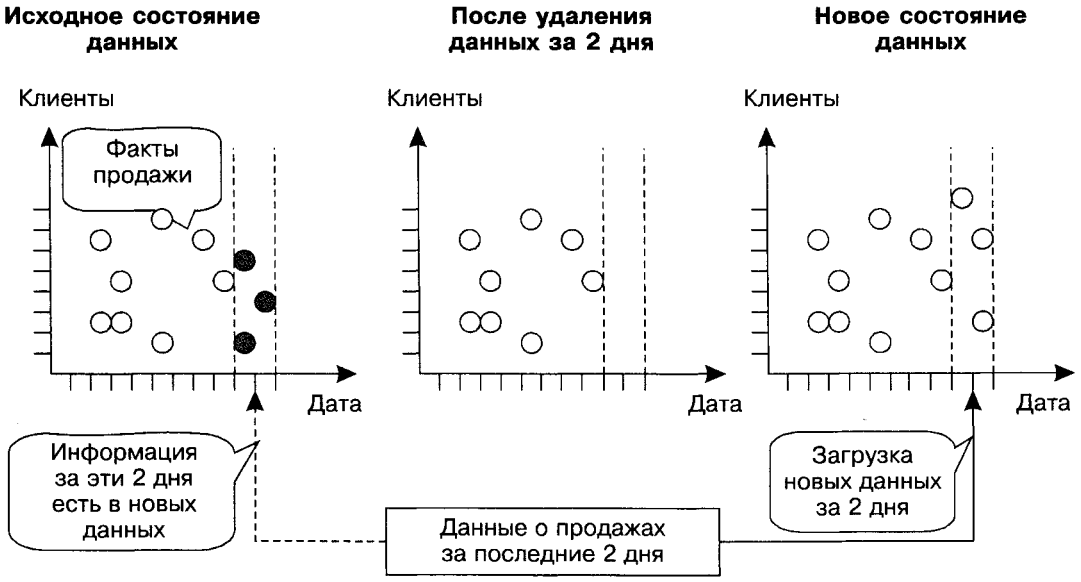


Рис. 7. Иллюстрация контроля непротиворечивости

Подобный способ загрузки удобен еще и тем, что позволяет избежать коллизий, например, когда в хранилище имеются некорректные данные за какой-то период. В таком случае лучше все данные за этот период удалить, а после загрузить новые корректные сведения.

На последней странице **Мастера экспорта** лучше оставить настройки по умолчанию.

Флажок **Автоматически добавлять значения измерений** позволяет «на лету» добавлять новые значения в существующие измерения. Но пользоваться опцией нужно с осторожностью. В случае бездумного ее применения можно очень быстро засорить хранилище данных, так как любое значение измерения, даже неверное, будет занесено как реально существующее.

Флажок **Группировать данные перед загрузкой в хранилище** полезен в следующей ситуации: вы до конца не уверены, что совокупность измерений процесса обеспечит уникальность точки в многомерном пространстве, и одновременно такой уровень детализации вас устраивает. В нашей задаче, если в таблице продаж встретятся две записи с одинаковыми значениями измерений (табл. 1), то при отсутствии установленного флажка **Группировать данные...** в хранилище попадет только вторая запись (последняя встретившаяся). Получится, что одна запись фактически потеряется, хотя нужно просуммировать значения полей *Количество* и *Сумма.*

Таблица 1

Случай, при котором совокупность измерений не дает уникальности

Дата	Код отдела	Код товара	Час покупки	Количество	Сумма
16.12.2008	1	3381	18	2	196,0
16.12.2008	1	3381	18	1	98,0

В **Мастере экспорта** можно задать любой вариант агрегации данных. Когда есть уверенность, что совокупность измерений процесса обеспечивает уникальность точки в многомерном пространстве, группировку можно не производить – это сэкономит время.

Окончательный сценарий загрузки приведен на рис. 8.

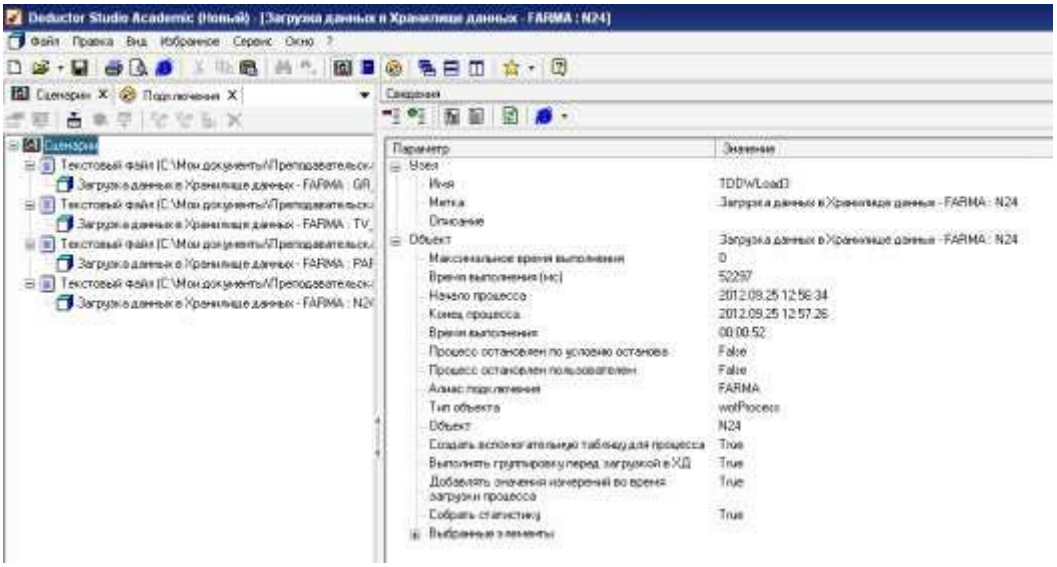


Рис. 8. Окончательный сценарий загрузки

В результате всех вышеописанных действий будет:

- ☐ создано и наполнено хранилище данных;
- ☐ написан сценарий загрузки (пополнения) информации из источников в ХД;
- ☐ продуман контроль непротиворечивости данных в ХД.

Заметим, что сценарий загрузки привязан не к данным непосредственно, а к их структуре, то есть в нем смоделирована последовательность действий, которые нужно выполнить для загрузки информации в ХД: имена файлов-источников, соответствие полей и т.д. Один раз созданный сценарий впоследствии применяется для пополнения хранилища данных. Как правило, эти процедуры проводятся по регламенту в нерабочее время (например, ночью) с использованием пакетного или серверного режима.

Практическое занятие № 3
Срезы хранилища данных и OLAP-кубы

Процесс получения данных из хранилища осуществляется при помощи **Мастера импорта** (контекстное меню или клавиша **F6**). Построим отчет, отражающий динамику сумм продаж по месяцам года в разрезе групп товаров и аптек. Для этого выполните следующие действия.

1. С помощью **Мастера импорта** выберите тип источника данных – Deductor Warehouse, на следующем шаге – хранилище *Фармация*, а затем – процесс *Продажи.* Далее задайте, какие измерения и атрибуты необходимо импортировать (рис. 1). Заметим, что благодаря иерархии внутри измерения *Товар.Код* появилась возможность доступа к измерению *Группа.Код.*

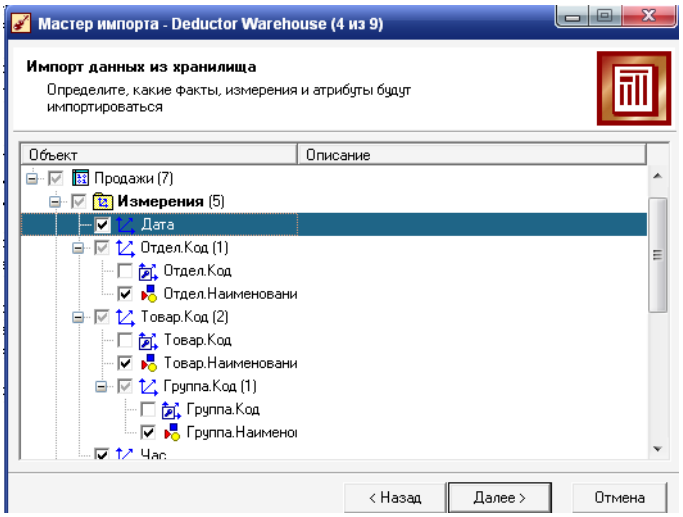


Рис. 1. Выбор импортируемых измерений и атрибутов

В этом же окне задайте импортируемые факты и виды их агрегации. В большинстве случаев требуется агрегация в виде суммы.

2. Определите срезы для выбранных измерений. Это целесообразно делать при большом количестве значений измерения, так как позволяет загружать с сервера, на котором расположено ХД, только интересующие значения измерений и тем самым экономить время загрузки данных. Установим срез по измерению *Дата*: «Все продажи за последние 4 месяца от имеющихся данных» (рис. 2).

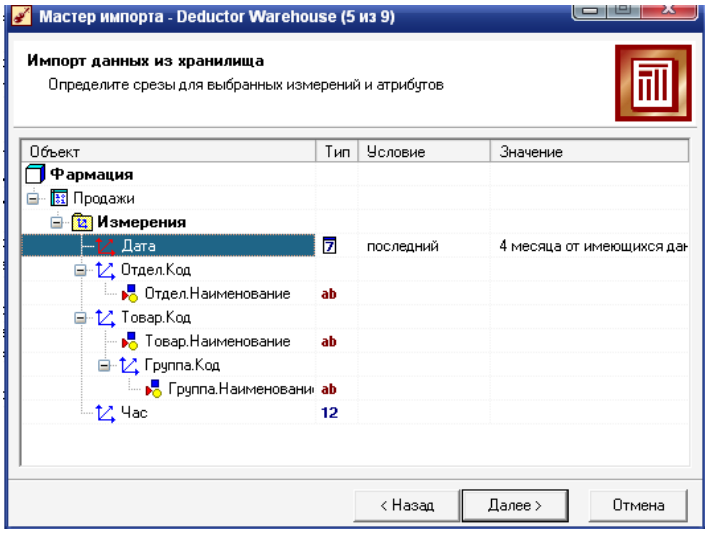


Рис. 2. Выбор срезов

3. Настройте так называемые динамические фильтры. Это означает, что при каждом выполнении узла импорта будет выводиться окно, аналогичное окну настройки среза, в котором он сможет указать требуемые разрезы по этому измерению. Опция позволяет строить динамические отчеты, в которых пользователю предоставляется только интересующая его информация, а конкретные условия фильтрации он выбирает в момент импорта данных.

Нажмите кнопку **Пуск**, дождитесь импорта данных и выберите визуализатор **Таблица**.

В вашем распоряжении имеется только измерение *Дата*, а для построения OLAP-отчета требуются отдельные измерения *Месяц* и *Год*. Их можно извлечь из даты, применив к узлу импорта из хранилища обработчик **Дата и время** (он выбирается в **Мастере**

обработки, который можно вызвать из контекстного меню или нажатием клавиши **F7**). Суть этого обработчика заключается в том, что на основе столбца с информацией о дате/времени формируются один или несколько столбцов, в которых указывается, к какому заданному интервалу времени принадлежит строка данных. Тип интервала задается на единственной вкладке настроек узла в зависимости от того, что вы хотите выделить из даты (рис. 4).

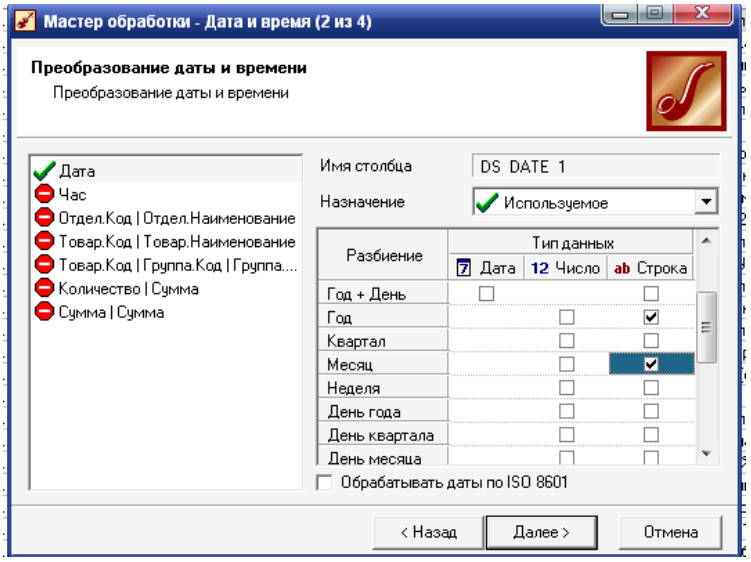


Рис. 4. Извлечение из даты месяца и года

В результате в выходном наборе будет создано два новых строковых столбца с метками *Дата (Год)* и *Дата (Месяц)*.

4. Для результирующего набора данных определите способ его отображения – куб и настройте назначения полей куба, то есть укажите измерения и факты. Для нашего отчета измерениями будут измерения *Дата (Месяц), Дата (Год), Отдел.Наименование* и *Группа.Наименование,* а фактами – *Количество* и *Сумма* проданных товаров (с агрегацией «Сумма»). При построении куба информационное поле *Дата* не будет отображаться, но будет доступно в детализации.

5. На следующем шаге нужно задать размещение измерений по строкам/столбцам (рис. 5).

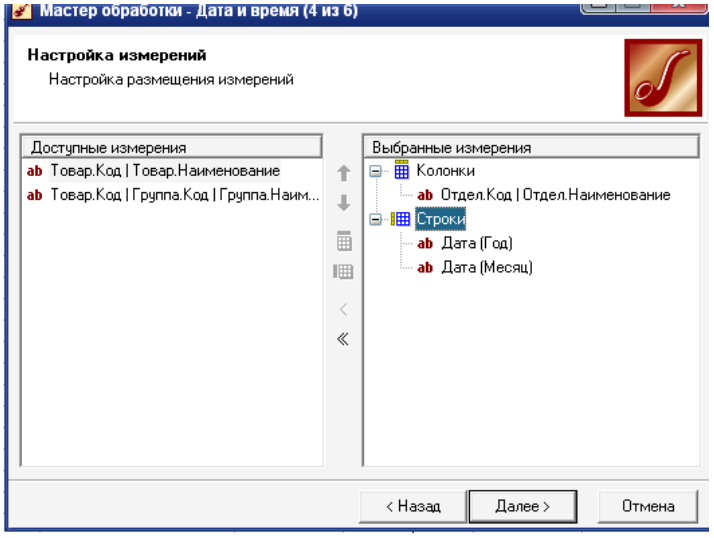


Рис. 5. Настройка размещения полей куба

6. На последнем шаге определите, какие факты нужно отображать в кубе на пересечении измерений, и их агрегацию (рис. 6).

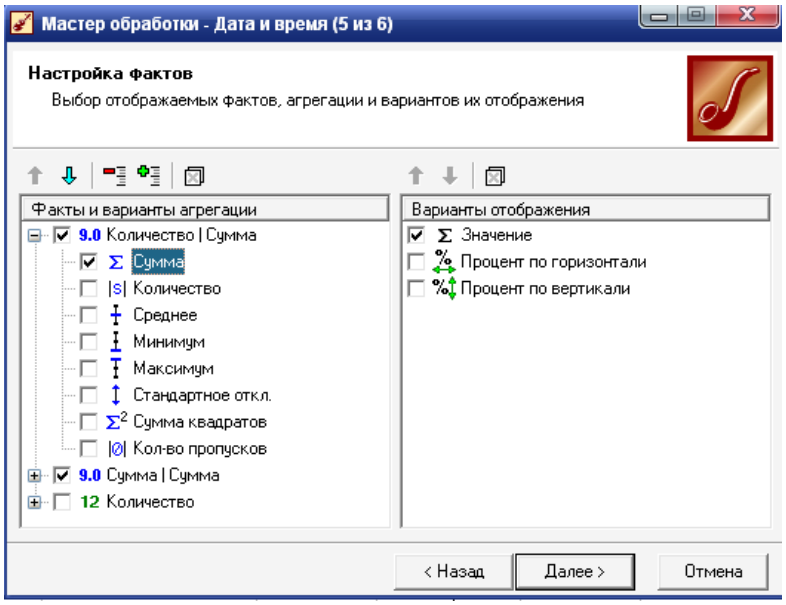


Рис. 6. Настройка отображения фактов

Таким образом, наш сценарий будет включать два узла.

В результате получим следующий многомерный отчет (рис. 7).

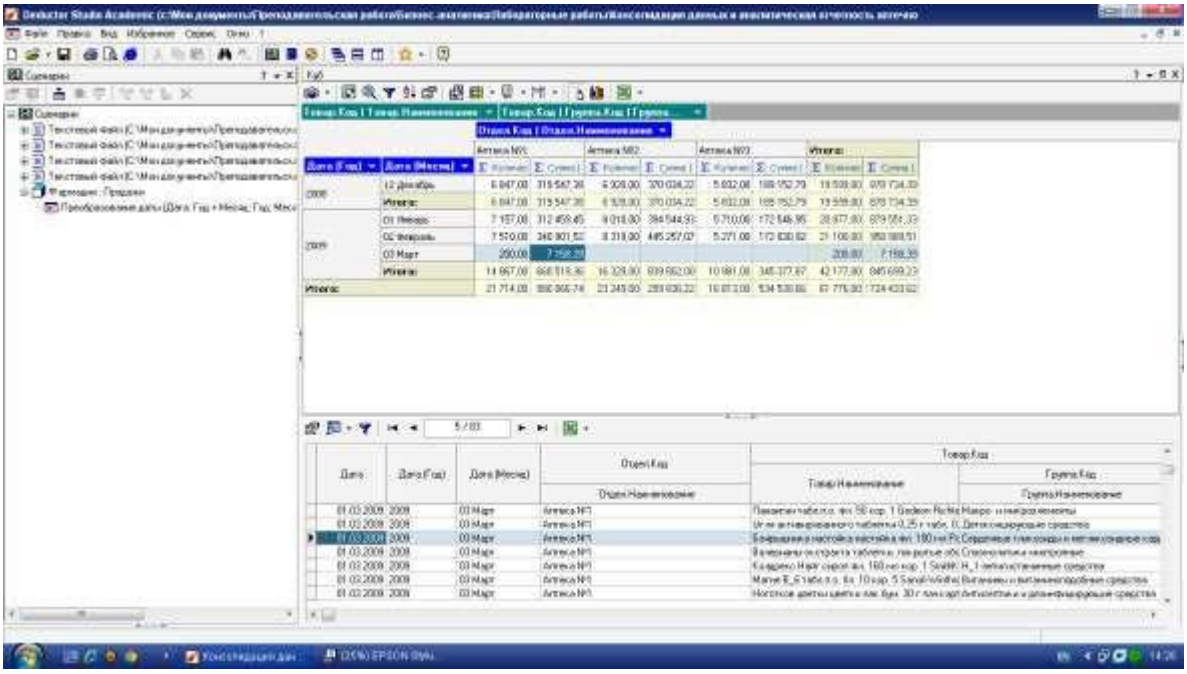


Рис. 7. OLAP-отчет о продажах в разрезе месяца, года и аптеки Фильтрация данных в кубе может производиться двумя способами:

- по значениям фактов;
- по значениям измерений.

Для фильтрации данных в кубе необходимо во всплывающем меню или на панели инструментов нажать кнопку **Селектор...**.

Пусть нужно определить товарные группы, приносящие 80% выручки. Выберите

измерение *Группа*, условие – *Доля от общего*, значение – 80 и настройте в кубе одно активное измерение, добавив вывод относительных долей и отсортировав по убыванию (рис. 8).

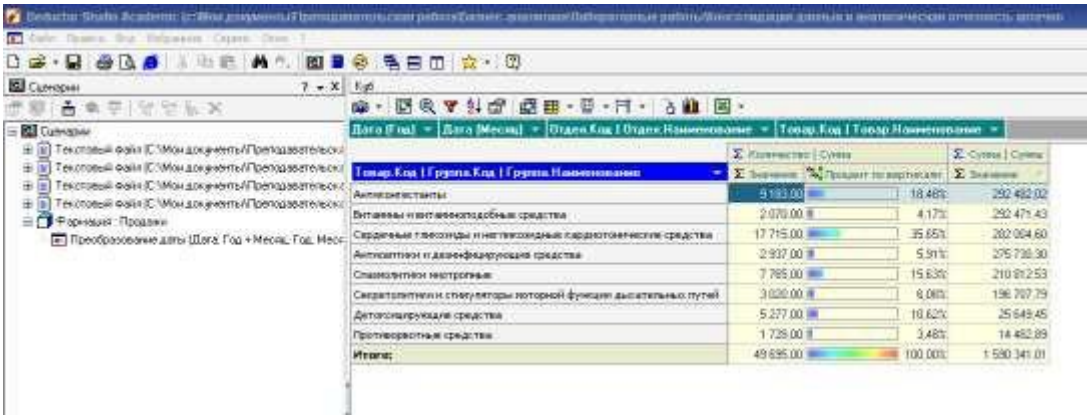


Рис. 8. Куб с группами препаратов, дающих 80% выручки

Такую выборку можно получить по любому факту. В данном примере это сумма. Если выбрать измерение *Товар* и отфильтровать по количеству, то получим лекарственные препараты, пользующиеся наибольшим спросом.

Одновременно с кубом всегда строится кросс-диаграмма. Ее отличие от обычной диаграммы в том, что она однозначно соответствует текущему состоянию куба и при любых его изменениях (транспонировании, вращении) тоже модифицируется. Например, построим отчет и кросс-диаграмму загруженности аптек (по количеству проданных единиц товаров) за последние 7 дней (рис. 9).

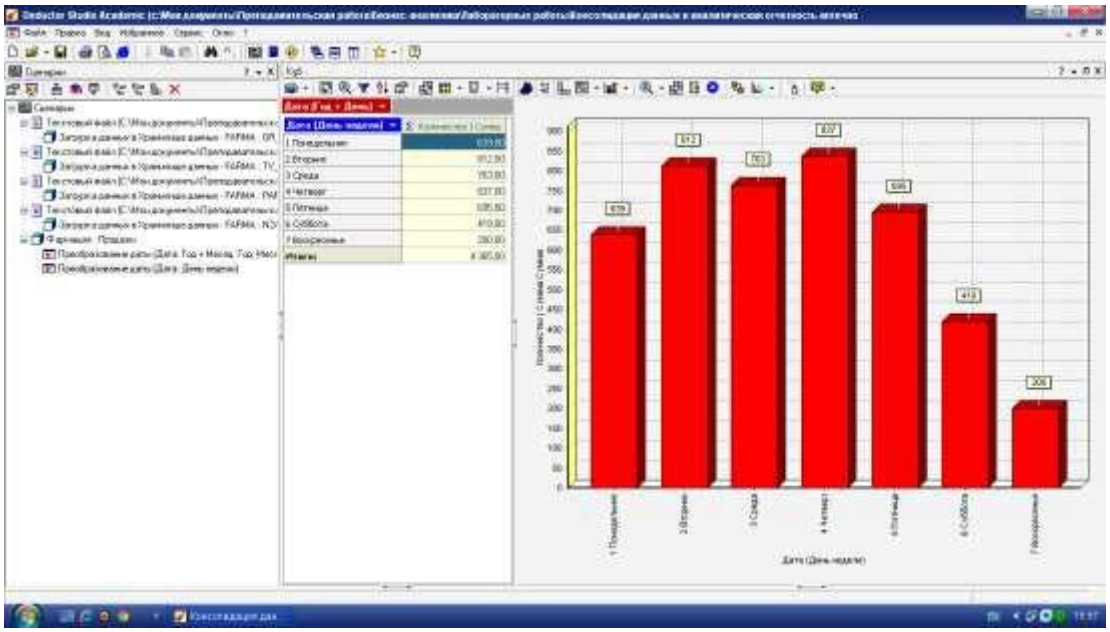


Рис. 9. Загруженность аптек по дням недели

Практическое занятие № 4
Исследование свойств многослойных нейронных сетей

Для изучения свойств нейронной сети в этой задаче будем использовать массив, в котором каждый элемент состоит из двух одинаковых чисел. Таким образом, входы нейросети будут равны ее выходам.

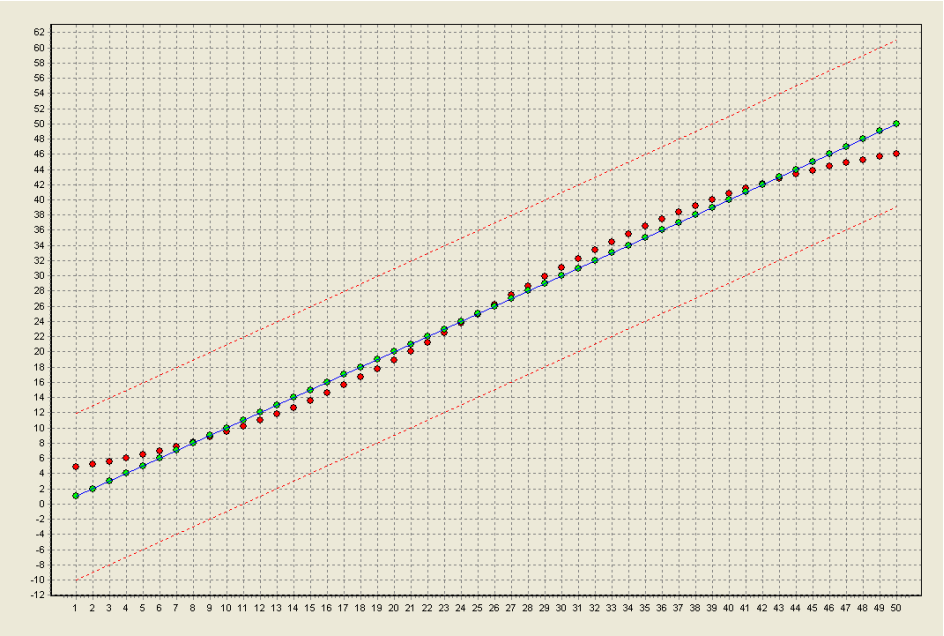
A	B
1	1
2	2
3	3
4	4
5	5
6	6
...	...

Создайте такой массив из 50 элементов со значениями от 1 до 50, сохраните его в виде текстового файла и импортируйте в Deductor Academic.

Теперь откройте «Мастер обработки» и выберите «Нейросеть». Обработайте данные с помощью этого инструмента, причем столбец A будет входным, а столбец B – выходным полем. При настройке сети оставьте все значения всех параметров по умолчанию: это должна быть сеть с двумя нейронами в единственном скрытом слое, активационная функция – сигмоида с крутизной 1,000. Алгоритм обучения – resilient propagation, обучение происходит до достижения 10000 эпох.

При определении способов отображения включите диаграмму рассеяния и инструмент «Что-если».

Прежде всего обратите внимание на диаграмму рассеяния. Она будет иметь примерно такой вид:



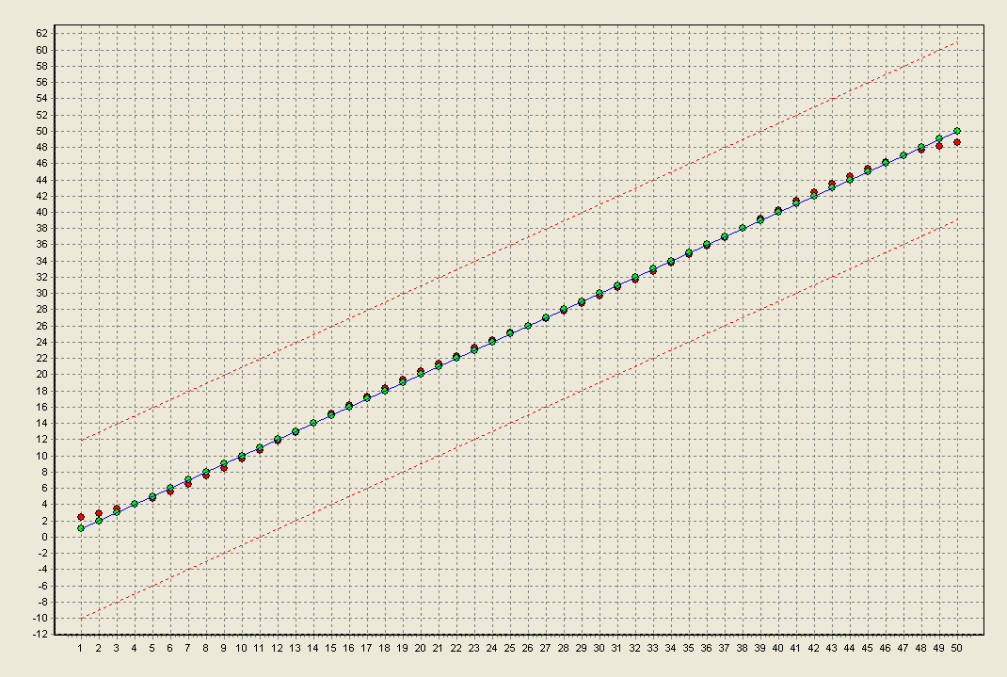
Хорошо видно, что ряд выходных значений напоминает сигмоидальную функцию. В самых нижних значениях он отклоняется от эталона в большую сторону, а в верхних – в меньшую. Таким образом, крайние значения всегда будут предсказаны с некоторой ошибкой, направленной к центру совокупности.

Каким образом поведет себя сеть за границами исходных данных? Для проверки этого воспользуйтесь инструментом «Что-если». Попробуйте ввести в качестве входного значения данные, немного выходящие за пределы значений исходных данных, а затем существенно выходящие за ее пределы.

Заметная на диаграмме сигмоидальная форма рассеяния вызывает аналогии с сигмоидальной формой активационной функции, использованной в модели. Возникает пред-

положение, что сигмоидальность рассеяния можно изменить, изменяя крутизну активационной функции. Возможно, с уменьшением крутизны наклона сигмоиды и кривая рассеяния станет менее крутой и, таким образом, более приближенной к эталону.

Для проверки этой гипотезы обучите сеть вновь на тех же исходных данных. Внесите изменения лишь в параметры активационной функции: обучите следующую сеть с крутизной активационной сигмоиды в 0,250, а затем еще одну – с крутизной в 4,000.



Если вы заметили изменения в характере рассеяния, то логично было бы предположить и изменение в поведении сети за границами исследуемого диапазона. Воспользуйтесь инструментом «Что-если» для ответа на следующие вопросы.

Практическое занятие № 5
Исследование проблемы переобучения нейронной сети

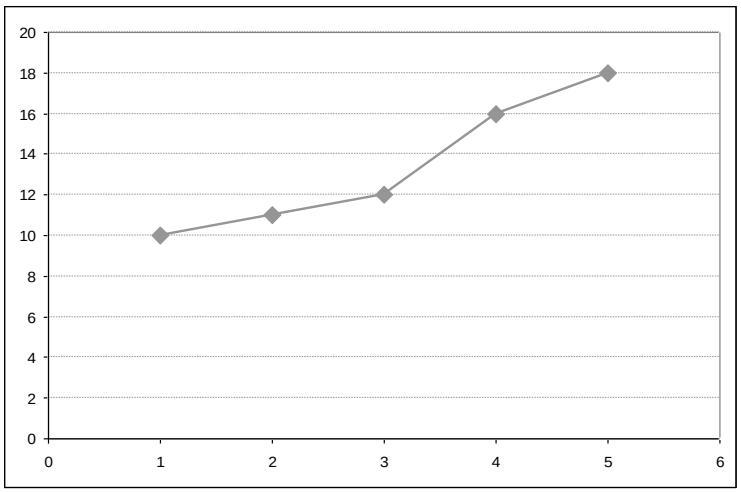
Эта известная проблема обычно возникает в случае, когда число учебных образцов невелико, а сеть является слишком мощной. В этом случае искусственной нейронной сети проще запомнить учебные образцы, а не выявлять закономерности, связывающие входные и выходные факторы. В результате сеть с чрезмерной точностью адаптируется к значениям конкретных данных, а не к диапазонам, которые эти данные должны представлять. Наглядно представить то, что происходит в этом случае, можно на примере из совсем другой области: аппроксимации данных посредством различных функций. Это легко сделать в программе Microsoft Excel.

Запишите на лист Microsoft Excel следующий набор данных:

x	y
1	10
2	11
3	12
4	16
5	18

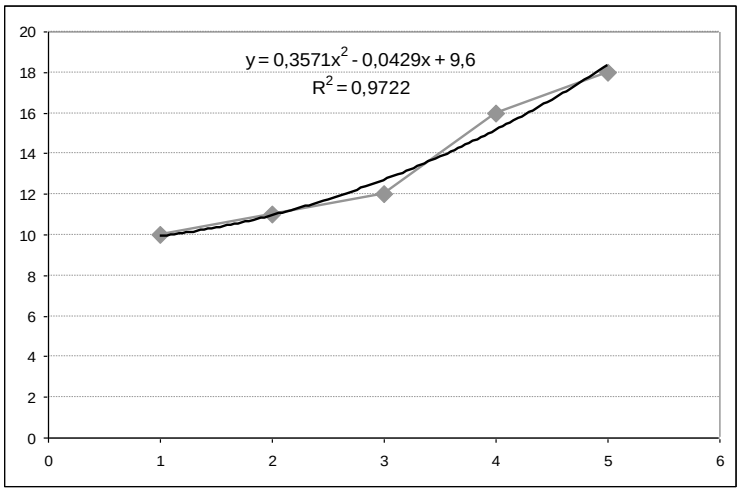
Условно предположим, что значения ***x*** представляют собой равноудаленные между собой временные точки (например, начало первой, второй и т.д. недель), а ***y*** – значения некоторого финансового показателя. Это значит, что мы можем попытаться спрогнозировать будущие, а также предположить промежуточные значения ***y*** на основе построения уравнения регрессии.

Постройте график зависимости ***y*** от (точечную диаграмму со значениями, соединенными сглаживающими линиями). У вас должна получиться ломаная линия примерно такого вида:



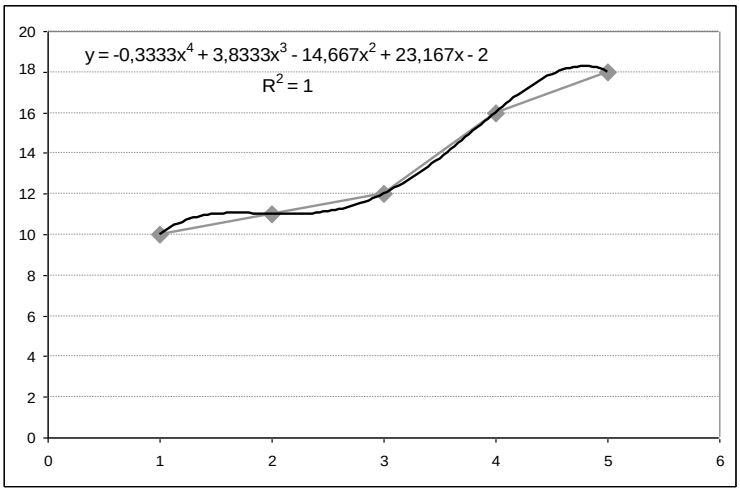
Теперь попробуйте добавить к ней линию тренда, указав в параметрах, что необходимо поместить на диаграмму уравнение и величину коэффициента достоверности аппроксимации R^2. Вы можете выбирать любой тип тренда, кроме линейной фильтрации, которая не имеет уравнения регрессии и, соответственно, коэффициента достоверности аппроксимации.

Попробуйте изменять тип тренда в попытке найти наилучший тренд. Вы увидите, что с хорошей достоверностью (более 0,9) явление представляют линейный, экспоненциальный, полиноминальный тренды. Не увеличивайте степень полинома (по умолчанию 2). Наилучшие результаты в данном случае показывает полином (достоверность выше 0,98):



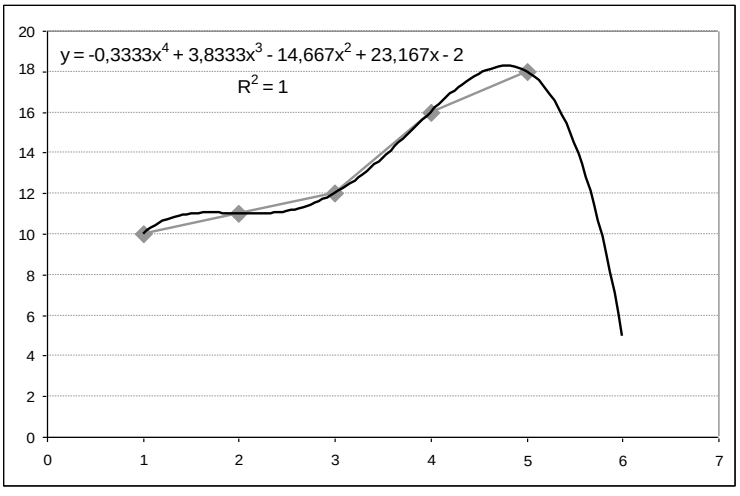
Можно наглядно увидеть, что линия тренда лежит достаточно близко к представленным данным и в определенной мере отражает общую закономерность монотонного роста показателя.

Теперь попробуем увеличить степень полиноминального тренда до 4. Его форма существенно изменится:



Мы можем увидеть, что величина коэффициента достоверности аппроксимации возросла до единицы. Это означает, что тренд точно прошел через все указанные нами точки. Однако форма его стала причудливой – вместо монотонно возрастающей функции мы видим волнообразную кривую с восходящими и нисходящими участками. Дело в том, что формула регрессии стала слишком сложной. В попытке максимально приблизиться к моделируемым образцам она перестала отражать общую тенденцию. Она идеально отражает исходные точки, но уже небольшое отступление от них может привести к существенной ошибке. Например, с уменьшением значения ***x*** с 5,0 до 4,8 значение ***y*** растет, а не убывает.

Такое «идеальное» уравнение регрессии нельзя применить на практике. Попробуйте в параметрах тренда выставить прогноз вперед на одну единицу. Вы увидите, что вопреки вполне наглядной положительной корреляции между ***x*** и ***y*** значение функции резко упадет, причем до такого значения, которое существенно ниже минимума исходной выборки.



Явление, которое мы наблюдали на этом примере, в значительной степени схоже с явлением переобучения многослойной нейронной сети. В обоих случаях усложнение алгоритма (а в нейронной сети усложнение алгоритма зависит от увеличения числа связей) приводит к тому, что уже известные образцы моделируются с высокой точностью, но общие закономерности при этом выявляются плохо, и потому для новых значений входных

факторов прогноз оказывается крайне неудачным. Некоторое отличие в данном случае состоит в следующем: как вы уже знаете по предыдущему заданию, многослойная нейронная сеть не допустила бы такого резкого падения значения ***y***, значительно выходящего за пределы исходной выборки. Однако общие черты явлений совпадают.

Для демонстрации явления используйте простой массив, подобный тому, что использовался в первой задаче, но всего из пяти элементов:

A	B
1	1
2	2
3	3
4	4
5	5

Импортируйте данные в Deductor Academic. Откройте «Мастер обработки» и выберите «Нейросеть». Как и в прошлом задании, столбец A будет входным, а столбец B – выходным полем.

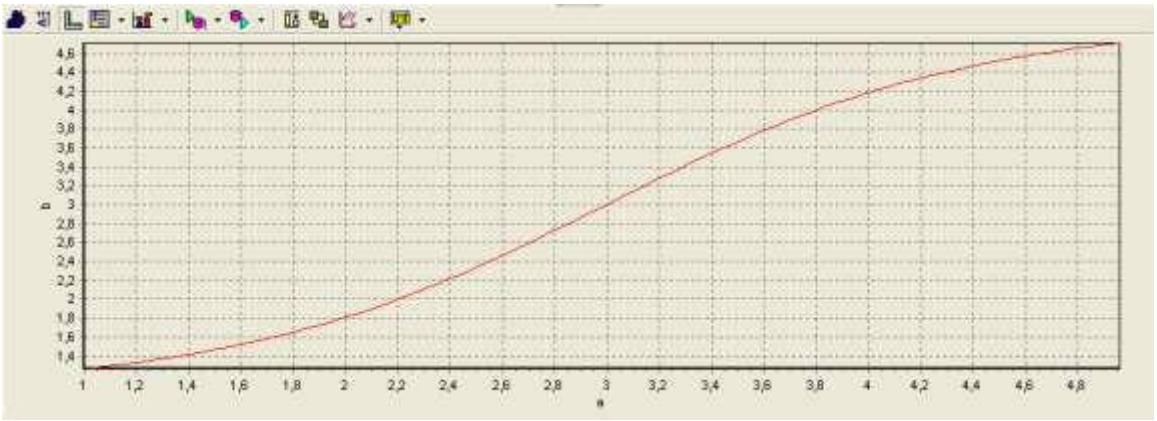
Пусть вас не смущает то, что на втором шаге настройки в тестовом множестве по умолчанию не окажется ни одного образца. В данном примере нас интересует отображение учебного множества, и мы не будем оценивать возможности работы сети на данных, не участвовавших в обучении.

Вначале используйте сеть с одним скрытым слоем, содержащим два нейрона – такая конфигурация используется по умолчанию. Оставим в качестве активационной функции сигмоиду с кривизной 1,0.

Однако для такого маленького набора данных алгоритм эластичного обратного распространения (resilient propagation), работающий в режиме «оффлайн», является слишком грубым. Он корректирует веса один раз за эпоху, после предъявления всех учебных образцов. Но при малом числе образцов доля каждого из них в формировании результата оказывается весьма существенной, и потому лучше корректировать веса после предъявления каждого из учебных векторов. Поэтому во всех примерах этого задания переключайте обучение на алгоритм обратного распространения (back propagation).

Проведите обучение сети. Для того, чтобы продемонстрировать его результат, выберите способ отображения «Что-если». На панели инструментов появившегося окна нажмите кнопку «Показать график». Именно график, который появится внизу окна, и представляет для нас наибольший интерес.

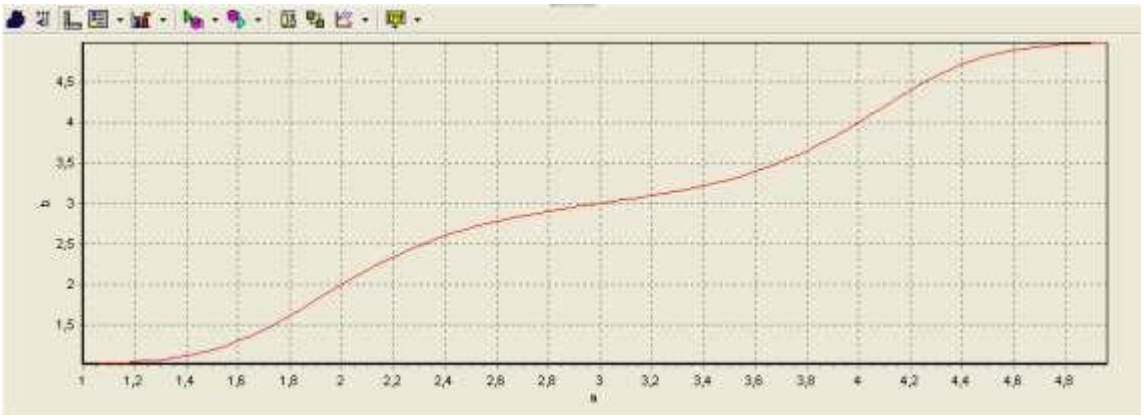
Рассмотрите его форму. Скорее всего, вы увидите нечто похожее на диаграмму, приведенную ниже. Не забывайте, что в связи с элементом случайности, связанным с начальной инициализацией весов, результаты обучения сети могут несколько отличаться, даже если обучение происходило на тех же данных.



Можно увидеть, что график функции представляет собой монотонно возрастающую кривую, которая, однако, не вполне точно проходит через пары точек {1;1}, {2;2} и т.д.

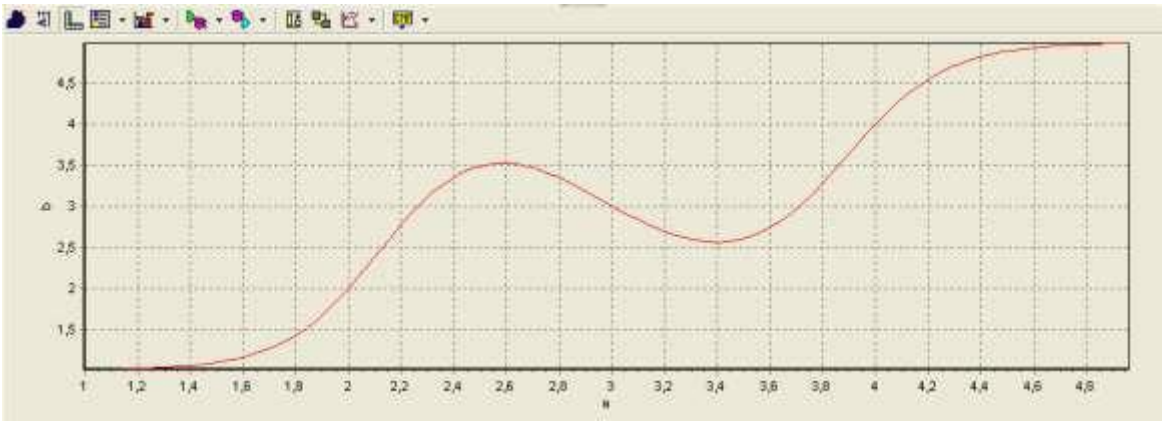
Приближение выходных сигналов к ожидаемым можно изменить, поменяв степень кривизны сигмоиды. Повторно обучите сеть на тех же данных, и с теми же параметрами нейронной сети, за исключением крутизны сигмоиды, которую установите в максимальное значение: 4,0.

В результате график несколько изменит форму. Возможно, он приобретет более сложную форму, однако можно увидеть, что примеры из обучающего множества представлены намного более точно.



Однако в использованной сети слишком мало связей, чтобы продемонстрировать проблему переобучения. Обучите еще одну сеть, у которой в скрытом слое будет 20 нейронов. Оставьте остальные параметры такими же, как у предыдущей сети: в качестве активационной функции применим сигмоиду с крутизной 4,0, а в качестве алгоритма обучения – back propagation с параметрами по умолчанию.

График вновь изменится и приобретет сложную форму, близкую к представленной ниже:



Можно увидеть, что кривая приобрела замысловатую форму, подобную той, что мы видели при изучении полиноминального тренда. При этом она довольно точно проходит через точки, определенные обучающим множеством, но в промежутках между ними ведет себя неожиданным образом.

Практическое занятие № 6
Исследование свойств карт Кохонена

В ходе выполнения этого и последующих заданий мы попытаемся наглядно увидеть, как происходит упрощение многомерного представления до двухмерной карты, и от каких факторов оно зависит.

Как известно, в основе функционирования самоорганизующихся карт Кохонена лежит алгоритм проецирования с сохранением топологического подобия. Многомерное пространство входных образцов проецируется в пространство с более низкой размерностью, причем образцы, близкие в исходном пространстве, оказываются рядом и на полученной карте.

В Deductor Studio существует возможность понижения размерности лишь на двухмерное пространство, а точнее, на конечную двухмерную сетку с прямоугольными или шестиугольными ячейками. От понимания того, как это происходит, зависит эффективность использования самоорганизующихся карт Кохонена в реальных экономических задачах.

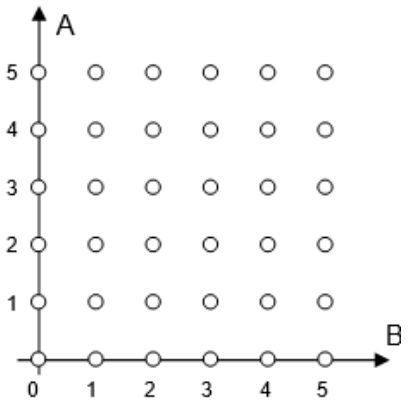
Перед началом работы необходимо напомнить, что в алгоритме функционирования самоорганизующихся карт Кохонена, как и многих других нейронных сетей, присутствует элемент случайности, связанный, прежде всего, с необходимостью случайной инициализации начальных весов нейронов. В связи с этим результаты, которые вы получите, могут несколько отличаться от тех, которые представлены в этом пособии. Возможно, в некоторых случаях вам придется несколько раз построить и обучить одинаковые сети, чтобы наглядно увидеть желаемые результаты.

Для того, чтобы понять закономерности функционирования карт Кохонена, мы вначале воспользуемся двухмерным входным пространством. Фактически, задача будет состоять в проецировании двухмерной поверхности на такую же поверхность. Хотя эта задача и лишена практического смысла, она позволяет понять, каким образом происходит «разворачивание» двухмерной сетки на пространстве образцов.

Прежде всего, подготовьте исходный массив следующего вида:

A	B
0	0
0	1
0	2
0	3
…	…
5	5

Всего в массиве должно быть 36 элементов, расположенных в единичных узлах прямоугольной сетки {0;0;5;5}, то есть элементы должны представлять все возможные комбинации пар целых чисел от 0 до 5.



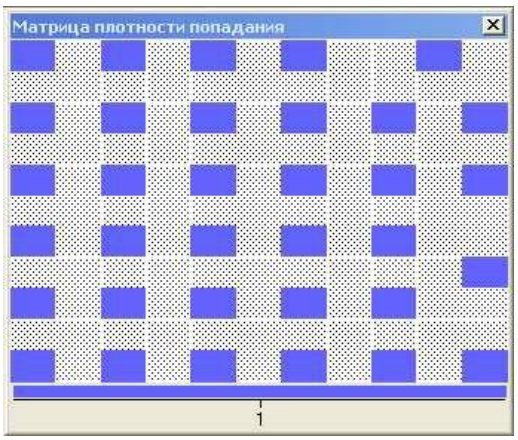
Как вы видите, это регулярная сетка, и при благоприятных условиях она будет спроецирована на карту Кохонена в неизменном виде.

Попробуйте обучить сеть этой задаче. Импортируйте данные (определите при этом вид данных для обеих переменных как непрерывный). В «Мастере обработки» выберите пункт «Карты Кохонена». Обратите особое внимание, что и столбец A и столбец B нужно установить, как входные. Оставьте все параметры по умолчанию, за исключением параметров карты. Мы будем использовать карту размером 11x11 с прямоугольными ячейками, которая наилучшим образом соответствует входному множеству. В идеальном случае образцы должны разместиться на карте в правильном прямоугольном порядке через один нейрон.

Приступите к обучению. Карты Кохонена обучаются быстрее, чем многослойные нейронные сети, и 500 эпох, установленных по умолчанию, будет достаточно. По окончанию обучения выберите в качестве способа отображения «Карту Кохонена», а в ней отметьте «Матрицу плотности попадания».

Матрица плотности попадания, или просто «матрица попадания», демонстрирует, какие из нейронов стали победителями и для скольких нейронов. Нейроны сети являются ячейками карты Кохонена, поэтому, если нейрон является победителем для одного или нескольких образцов, говорят, что образцы попали в эту ячейку. Отсюда и название этого способа отображения. На матрице плотности ячейки без образцов (нейроны, которые не стали победителями) обозначены точечной заливкой, а ячейки нейронов-победителей имеют цветовое выделение, показывающее, сколько образцов попало в ячейку.

Взгляните на матрицу плотности попадания. Скорее всего, вы увидите, что распределение является достаточно похожим на образец, но не вполне точным.



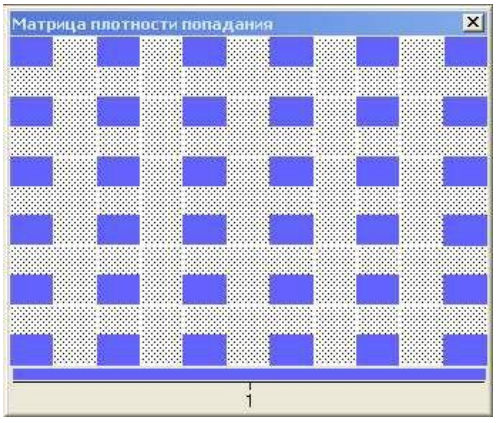
Можно заметить, что на карте Кохонена присутствуют не все 36 образцов. Нетрудно догадаться, что это связано с выделением нескольких образцов в тестовое множество.

С помощью способа отображения «Обучающее множество» можно определить, какие именно образцы оказались в тестовом множестве.

Вернитесь к карте Кохонена. Посмотрим, как распределены образцы по карте. Для этого нажмите на панели инструментов кнопку «Показать/скрыть окно данных». Под картами Кохонена появится таблица с перечнем образцов и информацией об их расположении. Образцы, попавшие в тестовое множество, выделены более темной заливкой.

С помощью окна данных можно понять, как образцы расположены на карте. Чтобы найти положение образца на карте, нужно поставить курсор в одну из строк таблицы и в контекстном меню выбрать пункт «Найти ячейку на карте». Чтобы совершить обратное действие и понять, какой образец находится в той или иной ячейке, нужно на панели окна данных установить способ фильтрации «Фильтр по ячейке». Тогда, выбрав ячейку на карте, можно увидеть, какие образцы в нее попали.

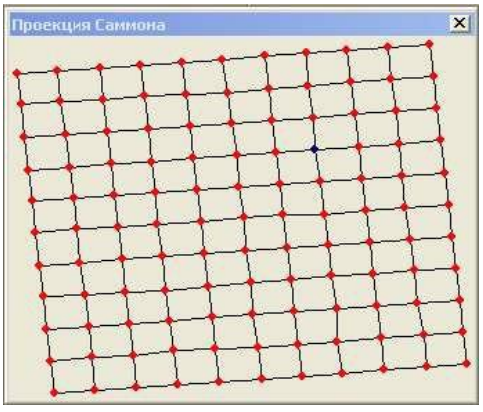
Для того, чтобы проверить распределение по карте всех 36 образцов, вновь обучим такую же сеть, но на этот раз полностью уберем тестовое множество, оставив все 100% образцов в обучающем множестве. После обучения вновь посмотрите на матрицу плотности попадания. Вероятнее всего, на этот раз вы увидите совершенно равномерное распределение.



Проверьте, является ли распределение таким же, как в исходном множестве.

Обратите внимание на то, что для карт Кохонена пространство исходных образцов данной задачи представляется совершенно симметричным как по осям диагоналей, так и по линиям, делящим область пополам параллельно осям координат. Поэтому существует восемь равновероятных регулярных расположений образцов данной задачи на карте Кохонена, получаемых друг из друга отражением или поворотом.

Еще один способ наглядного представления результатов работы сети – проекция Саммона, являющаяся результатом проецирования сети Кохонена вместе с ее связями на плоскость. Если на ваших картах нет этой проекции, ее можно добавить, воспользовавшись кнопкой «Настройка отображений» вкладки «Карта Кохонена»



Скорее всего, вы увидите хорошо развернутую, достаточно ровную сетку. Может создаться впечатление, что она всегда была такой, и образцы просто были аккуратно перенесены в ее ячейки. Однако это не так: подобную форму сетка приобрела в ходе обучения.

Повторно обучите такую же сеть, но установите всего лишь одну эпоху для обучения. Рассмотрите результаты обучения. Скорее всего, вы увидите достаточно «скомканную» сеть на проекции Саммона, а на матрице плотности попадания образцы будут распределены достаточно спонтанно.

Таким образом, можно пронаблюдать, как происходит «разворачивание» двухмерной сети, первоначально скомканной случайным образом, на пространстве образцов. Для этого обучите идентичные сети с количеством эпох 1, 3, 5, 10, 15, 20, 25, 30, 50, 100.

Возможно, вы заметите, что сеть чрезвычайно быстро, уже на первых эпохах разворачивается, и затем начинает уточнять положения нейронов. Уже после первых эпох на проекции Саммона можно увидеть не скомканную случайную структуру, а хорошо развернутую, хотя и не регулярную сетку. Угловые элементы также очень быстро оказываются в угловых ячейках сети.

Почему это происходит? Как буквально за одну эпоху угловые нейроны находят правильное место в пространстве образцов?

Дело в том, что по умолчанию нейроны располагаются в пространстве образцов вовсе не случайно. На шаге настройки параметров обучения вы можете увидеть поле «Способ начальной инициализации карты». Оно может содержать три значения из раскрывающегося списка.

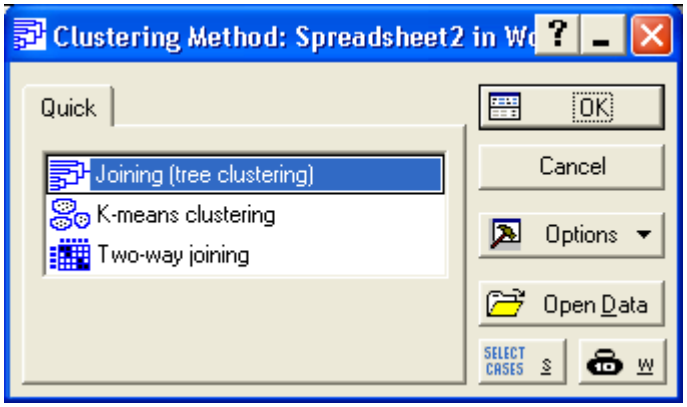
- случайными значениями
- из обучающего множества
- из собственных векторов

Для ускорения обучения по умолчанию используется способ инициализации «из собственных векторов». При этом начальные веса нейронов инициализируются значениями подмножества гиперплоскости, через которую проходят два главных собственных вектора матрицы ковариации входных значений обучающей выборки.

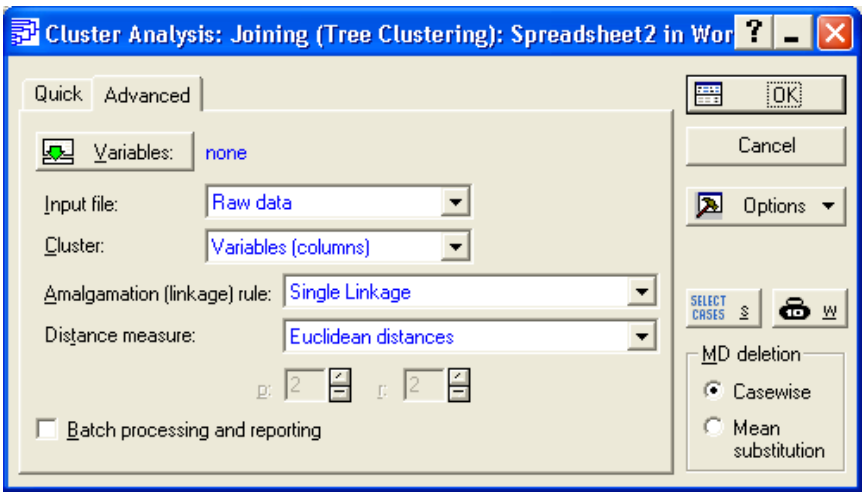
Попробуйте теперь инициализировать веса нейронов случайными значениями. Повторите этапы обучения, постепенно наращивая число эпох.

Практическое занятие № 7
Кластерный анализ в пакете Statistica

Запустите пакет и создайте новую рабочую книгу. Скопируйте в электронную таблицу данные из файла **tabl_2.xls**. Нажмите на панели инструментов кнопку **Cluster Analysis**. Откроется окно для выбора метода кластерного анализа.



В первой строке **Joining(tree clustering)** содержатся объединительные алгоритмы кластерного анализа, заканчивающиеся построением дендрограммы. Нажмите **Ok**. Откроется окно, которое позволяет задать необходимые параметры при использовании методов. В окне выберите вкладку **Advanced**.



1. Для задания переменных участвующих в классификации нажмите кнопку **Cases(rows)**. В открывшемся окне нажмите кнопку **Select All** а затем **Ok**.

Задайте тип входной информации в строке Input files.

1. **Raw data** (необработанные данные);
2. **Distance matrix** (матрица расстояний);

Выберите **Raw data.**

2. В строке **Cluster** установите способ классификации по признакам (**variables)** или по объектам (**cases)**. Выберите **cases**.

3. В строке **Amalgation [linkage] rule** определяется правило объединения кластеров.

В появившемся меню будут предложены следующие методы объединения (слияния) кластеров:

- **Single linkage** - метод одиночной связи ("принцип ближнего соседа")
- **Complete linkage** - метод полных связей ("принцип дальнего соседа")
- **Unweighted pair group average** - метод "средней связи"(невзвешенный)
- **Weighted pair group average** - взвешенный метод "средней связи"
- **Unweighted pair group centroid** - центроидный метод (невзвешенный)
- **Weighted pair group centroid** - взвешенный центроидный метод
- **Ward's method** - метод Уорда.

В качестве весов в 4 и 6 методах выступает число объектов в объединяемых кластерах.

4. В строке **Distance measure** задается метрика расстояний.

Euclidean distance - евклидово расстояние;

Squared euclidean distance - квадратичное евклидово расстояние

City - block (Manhattan) distance - манхэттенское расстояние или "расстояние городских кварталов".

Chebychew distance - расстояние Чебышева

Power distance - специальный класс метрических функций

Нажмите **Ok** Откроется окно для просмотра результатов кластерного анализа.

Просмотр результатов кластерного анализа.

Окно результатов анализа состоит из двух частей, разделенных горизонтальной линией. В верхней – показаны основные характеристики исходных данных и перечислены ранее введенные параметры, а в нижней расположены кнопки для получения подробной информации результатах кластерного анализа.

В верхней части этого окна указаны (в порядке следования на экране): Количество переменных; Количество наблюдений.

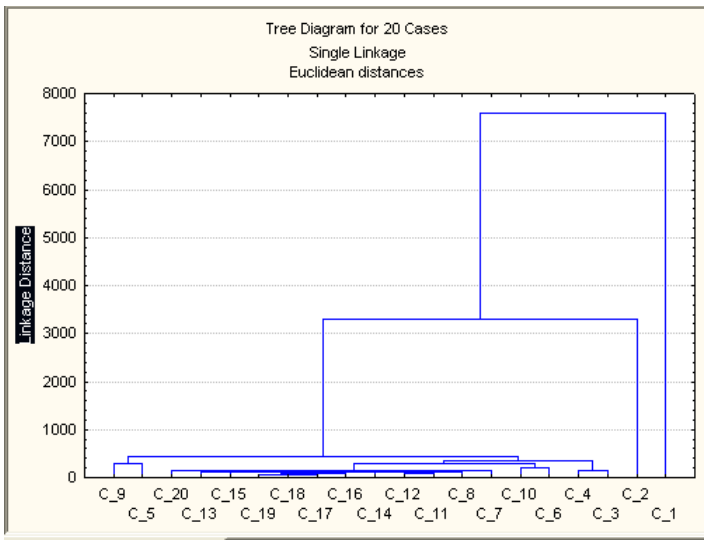
Осуществлена ли классификация наблюдений или переменных (зависит от установки параметра в строке **Cluster** предыдущем окне настройки);

Наблюдения с пропущенными данными были удалены или заменены средними значениями (зависит от установки параметра в, строке **Missing data** в предыдущем окне настройки);

Правило объединения кластеров (название иерархического агломеративного метода, заданного в строке **Amalgamation rules**, в предыдущем окне настройки);

Метрика расстояния (зависит от установки в строке **Distance** measure в предыдущем окне настройки);

Можно вызвать на экран горизонтальную или вертикальную дендрограмму. (**Horizontal hierarachical treeplot *или* Vertical icicle plot**). Наиболее традиционной является вертикальная дендрограмма.



По оси абсцисс отмечены наблюдения, а на оси ординат отмечены значения расстояний, при которых происходило последовательное объединение кластеров.

Кнопка **Amalgation schedule** открывает протокол объединения кластеров.

Кнопка **Graph of Amalgamation schedule** раскрывает окно, содержащее ступенчатое, графическое изображение изменений расстояний при объединении кластеров.

Для просмотра матрицы расстояний нажмите кнопку **Distance matrix**.

Это будет квадратная симметричная матрица, размерностью n*n, содержащая нули на главной диагонали. В заголовке матрицы указана ранее выбранная метрика расстояний.

В основном окне результатов классификации имеется строка **Save distance matrix as...** (сохранить матрицу расстояний как...) позволяющая задать имя файла, в котором будет сохранена матрица расстояний, которая в дальнейшем может быть подвергнута обработке.

Строка **Discriptive statistics** содержит такие важнейшие описательные статистики, как среднее (**means**) и среднеквадратическое отклонение (**standart deviations**) для каждого наблюдения. При проведении классификации ***n*** объектов по ***k*** признакам, для пользователя представляют большой интерес значения этих показателей: для каждого признака.

Задача.

По иерархическому агломеративному алгоритму провести классификацию $n = 6$ предприятий машиностроения, данные о деятельности которых характеризуются показателями: $x1$ – рентабельность (%), $x2$ – производительность труда (млн. руб./чел.) и представлены в таблице:

Показатель	**№ предприятия**					
	1	**2**	**3**	**4**	**5**	**6**
x1	23,4	17,5	9,7	18,2	23,4	17,5
x2	9,1	5,2	5,5	9,4	9,1	5,2