

Министерство науки и высшего образования Российской Федерации

Севастопольский государственный университет

МЕТОДИЧЕСКИЕ УКАЗАНИЯ

К ЛАБОРАТОРНОЙ РАБОТЕ № 4

«ИСПОЛЬЗОВАНИЕ МЕТОДОВ ИНТЕЛЛЕКТУАЛЬНОГО АНАЛИЗА ДАННЫХ В
ДИССЕРТАЦИОННЫХ ИССЛЕДОВАНИЯХ»

по дисциплине

«ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ В НАУКЕ И ОБРАЗОВАНИИ»

для аспирантов дневной и заочной форм обучения

Севастополь, 2018

Методические указания к лабораторной работе № 4 «Использование методов интеллектуального анализа данных в диссертационных исследованиях» по дисциплине «Информационные технологии в науке и образовании» для аспирантов дневной и заочной форм обучения. Сост.: Мащенко Е.Н. - Севастополь, изд-во СевГУ, 2018. - 12 С.

1. Цель работы:

Научиться строить линейный классификатор на основе нейронных сетей с использованием программной системы Deductor компании BaseGroup.

2. Краткие теоретические сведения

Решение задачи классификации является одним из важнейших применений нейронных сетей.

Задача классификации представляет собой задачу отнесения образца к одному из нескольких попарно не пересекающихся множеств.

Задача классификации представляет собой задачу отнесения образца к одному из нескольких попарно не пересекающихся множеств. Примером таких задач может быть, например, задача определения кредитоспособности клиента банка, медицинские задачи, в которых необходимо определить, например, исход заболевания, решение задач управления портфелем ценных бумаг (продать купить или "придержать" акции в зависимости от ситуации на рынке), задача определения жизнеспособных и склонных к банкротству фирм.

Цель классификации

При решении задач классификации необходимо отнести имеющиеся статические образцы (характеристики ситуации на рынке, данные медосмотра, информация о клиенте) к определенным классам. Возможно несколько способов представления данных. Наиболее распространенным является способ, при котором образец представляется вектором.

Компоненты этого вектора представляют собой различные характеристики образца, которые влияют на принятие решения о том, к какому классу можно отнести данный образец. Например, для медицинских задач в качестве компонентов этого вектора могут быть данные из медицинской карты больного. Таким образом, на основании некоторой информации о примере, необходимо определить, к какому классу его можно отнести. Классификатор таким образом относит объект к одному из классов в соответствии с определенным разбиением N -мерного пространства, которое называется пространством входов, и размерность этого пространства является количеством компонент вектора.

Прежде всего, нужно определить уровень сложности системы. В реальных задачах часто возникает ситуация, когда количество образцов ограничено, что осложняет определение сложности задачи. Возможно выделить три основных уровня сложности. Первый (самый простой) – когда классы можно разделить прямыми линиями (или гиперплоскостями, если пространство входов имеет размерность больше двух) – так называемая линейная разделимость. Во втором случае классы невозможно разделить линиями (плоскостями), но их возможно отделить с помощью более сложного деления – нелинейная разделимость. В третьем случае классы пересекаются и можно говорить только о вероятностной разделимости.

В идеальном варианте после предварительной обработки мы должны получить линейно разделимую задачу, так как после этого значительно упрощается построение классификатора. К сожалению, при решении реальных задач мы имеем ограниченное количество образцов, на основании которых и производится построение классификатора. При этом мы не можем провести такую предобработку данных, при которой будет достигнута линейная разделимость образцов.

Использование нейронных сетей в качестве классификатора

Сети с прямой связью являются универсальным средством аппроксимации функций, что позволяет их использовать в решении задач классификации. Как правило, нейронные сети оказываются наиболее эффективным способом классификации, потому что генерируют фактически большое число регрессионных моделей (которые используются в решении задач классификации статистическими методами).

К сожалению, в применении нейронных сетей в практических задачах возникает ряд проблем. Во-первых, заранее не известно, какой сложности (размера) может потребоваться сеть для достаточно точной реализации отображения. Эта сложность может оказаться чрезмерно высокой, что потребует сложной архитектуры сетей. Так, Минский в своей работе "Перцептроны" доказал, что простейшие однослойные нейронные сети способны решать только линейно разделимые задачи. Это ограничение преодолимо при использовании многослойных нейронных сетей. В общем виде можно сказать, что в сети с одним скрытым слоем вектор, соответствующий входному образцу, преобразуется скрытым слоем в некоторое новое пространство, которое может иметь другую размерность, а затем гиперплоскости, соответствующие нейронам выходного слоя, разделяют его на классы. Таким образом сеть распознает не только характеристики исходных данных, но и "характеристики характеристик", сформированные скрытым слоем.

Подготовка исходных данных

Для построения классификатора необходимо определить, какие параметры влияют на принятие решения о том, к какому классу принадлежит образец. При этом могут возникнуть две проблемы. Во-первых, если количество параметров мало, то может возникнуть ситуация, при которой один и тот же набор исходных данных соответствует примерам, находящимся в разных классах. Тогда невозможно обучить нейронную сеть, и система не будет корректно работать (невозможно найти минимум, который соответствует такому набору исходных данных). Исходные данные обязательно должны быть непротиворечивы. Для решения этой проблемы необходимо увеличить размерность пространства признаков (количество компонент входного вектора, соответствующего образцу). Но при увеличении размерности пространства признаков может возникнуть ситуация, когда число примеров может стать недостаточным для обучения сети, и она вместо обобщения просто запомнит примеры из обучающей выборки и не сможет корректно функционировать. Таким образом, при определении признаков необходимо найти компромисс с их количеством.

Далее необходимо определить способ представления входных данных для нейронной сети, т.е. определить способ нормирования. Нормировка необходима, поскольку нейронные сети работают с данными, представленными числами в диапазоне 0..1, а исходные данные

могут иметь произвольный диапазон или вообще быть нечисловыми данными. При этом возможны различные способы, начиная от простого линейного преобразования в требуемый диапазон и заканчивая многомерным анализом параметров и нелинейной нормировкой в зависимости от влияния параметров друг на друга.

Кодирование выходных значений

Задача классификации при наличии двух классов может быть решена на сети с одним нейроном в выходном слое, который может принимать одно из двух значений 0 или 1, в зависимости от того, к какому классу принадлежит образец. При наличии нескольких классов возникает проблема, связанная с представлением этих данных для выхода сети. Наиболее простым способом представления выходных данных в таком случае является вектор, компоненты которого соответствуют различным номерам классов. При этом i -я компонента вектора соответствует i -му классу. Все остальные компоненты при этом устанавливаются в 0. Тогда, например, второму классу будет соответствовать 1 на 2 выходе сети и 0 на остальных. При интерпретации результата обычно считается, что номер класса определяется номером выхода сети, на котором появилось максимальное значение. Например, если в сети с тремя выходами мы имеем вектор выходных значений (0.2, 0.6, 0.4), то мы видим, что максимальное значение имеет вторая компонента вектора, значит класс, к которому относится этот пример, – 2. При таком способе кодирования иногда вводится также понятие уверенности сети в том, что пример относится к этому классу. Наиболее простой способ определения уверенности заключается в определении разности между максимальным значением выхода и значением другого выхода, которое является ближайшим к максимальному. Например, для рассмотренного выше примера уверенность сети в том, что пример относится ко второму классу, определится как разность между второй и третьей компонентой вектора и равна $0.6 - 0.4 = 0.2$. Соответственно чем выше уверенность, тем больше вероятность того, что сеть дала правильный ответ. Этот метод кодирования является самым простым, но не всегда самым оптимальным способом представления данных.

Известны и другие способы. Например, выходной вектор представляет собой номер кластера, записанный в двоичной форме. Тогда при наличии 8 классов нам потребуется вектор из 3 элементов, и, скажем, 3 классу будет соответствовать вектор 011. Но при этом в случае получения неверного значения на одном из выходов мы можем получить неверную классификацию (неверный номер кластера), поэтому имеет смысл увеличить расстояние между двумя кластерами за счет использования кодирования выхода по коду Хемминга, который повысит надежность классификации.

Выбор объема сети

Правильный выбор объема сети имеет большое значение. Построить небольшую и качественную модель часто бывает просто невозможно, а большая модель будет просто запоминать примеры из обучающей выборки и не производить аппроксимацию, что, естественно, приведет к некорректной работе классификатора. Существуют два основных подхода к построению сети – конструктивный и деструктивный. При первом из них вначале берется сеть минимального размера, и постепенно увеличивают ее до достижения требуемой точности. При этом на каждом шаге ее заново обучают. Также существует так

называемый метод каскадной корреляции, при котором после окончания эпохи происходит корректировка архитектуры сети с целью минимизации ошибки. При деструктивном подходе вначале берется сеть завышенного объема, и затем из нее удаляются узлы и связи, мало влияющие на решение. При этом полезно помнить следующее правило: число примеров в обучающем множестве должно быть больше числа настраиваемых весов. Иначе вместо обобщения сеть просто запомнит данные и утратит способность к классификации – результат будет неопределен для примеров, которые не вошли в обучающую выборку.

Выбор архитектуры сети

При выборе архитектуры сети обычно опробуется несколько конфигураций с различным количеством элементов. При этом основным показателем является объем обучающего множества и обобщающая способность сети. Обычно используется алгоритм обучения Back Propagation (обратного распространения) с подтверждающим множеством.

Алгоритм построения классификатора на основе нейронных сетей

1. Работа с данными

- Составить базу данных из примеров, характерных для данной задачи
- Разбить всю совокупность данных на два множества: обучающее и тестовое (возможно разбиение на 3 множества: обучающее, тестовое и подтверждающее).

2. Предварительная обработка

- Выбрать систему признаков, характерных для данной задачи, и преобразовать данные соответствующим образом для подачи на вход сети (нормировка, стандартизация и т.д.). В результате желательно получить линейно отделяемое пространство множества образцов.
- Выбрать систему кодирования выходных значений (классическое кодирование, 2 на 2 кодирование и т.д.)

3. Конструирование, обучение и оценка качества сети

- Выбрать топологию сети: количество слоев, число нейронов в слоях и т.д.
- Выбрать функцию активации нейронов (например "сигмоида")
- Выбрать алгоритм обучения сети
- Оценить качество работы сети на основе подтверждающего множества или другому критерию, оптимизировать архитектуру (уменьшение весов, прореживание пространства признаков)
- Остановиться на варианте сети, который обеспечивает наилучшую способность к обобщению и оценить качество работы по тестовому множеству

Использование и диагностика

- Выяснить степень влияния различных факторов на принимаемое решение (эвристический подход).

- Убедиться, что сеть дает требуемую точность классификации (число неправильно распознанных примеров мало)
- При необходимости вернуться на этап 2, изменив способ представления образцов или изменив базу данных.
- Практически использовать сеть для решения задачи.

Для того, чтобы построить качественный классификатор, необходимо иметь качественные данные. Никакой из методов построения классификаторов, основанный на нейронных сетях или статистический, никогда не даст классификатор нужного качества, если имеющийся набор примеров не будет достаточно полным и представительным для той задачи, с которой придется работать системе.

Задание на лабораторную работу

Необходимо построить классификатор для решения задачи "Выдавать ли кредит клиенту" в аналитическом пакете Deductor (BaseGroup).

В качестве обучающего набора данных выступает база данных в электронной таблице Excel, содержащая информацию о клиентах, в частности: Сумма кредита, Срок кредита, Цель кредитования, Возраст, Пол, Образование, Частная собственность, Кредитная история. На основе этих данных необходимо построить модель, которая сможет дать ответ, входит ли Клиент, желающий получить кредит, в группу риска невозврата кредита, т.е. пользователь должен получить ответ на вопрос "Выдавать ли кредит?".

Ход работы:

1. Создать базу данных в электронной таблице Excel, содержащая информацию о клиентах, в частности: Сумма кредита, Срок кредита, Цель кредитования, Возраст, Пол, Образование, Частная собственность, Кредитная история. Последний столбец должен называться Выдача кредита и содержать значения «да» или «нет». Для заполнения столбца Выдача кредита необходимо разработать правила предоставления кредита по значениям атрибутов Сумма кредита, Срок кредита, Цель кредитования, Возраст, Пол, Образование, Частная собственность, Кредитная история. Атрибут Кредитная история может иметь три значения - «положительная» (Клиент ранее успешно возвращал кредиты), «отрицательная» (Клиент ранее имел долги по кредитам), и «не определена» (Клиент ранее не обращался за кредитами). База данных должна содержать не менее 150 строк.

2. Сохранить базу данных с именем credit.txt в формате «текстовые файлы с разделителями табуляции» для дальнейшего экспорта в аналитическую систему Deductor.

3. Скачать аналитическую систему Deductor Academic версии 5.3 по ссылке <http://www.basegroup.ru/download/deductor/>

4. Установить аналитическую систему Deductor Academic версии 5.3 на свой компьютер.

5. Импортировать данные из файла credit.txt при помощи мастера импорта (клавиша F6, рисунок 2).

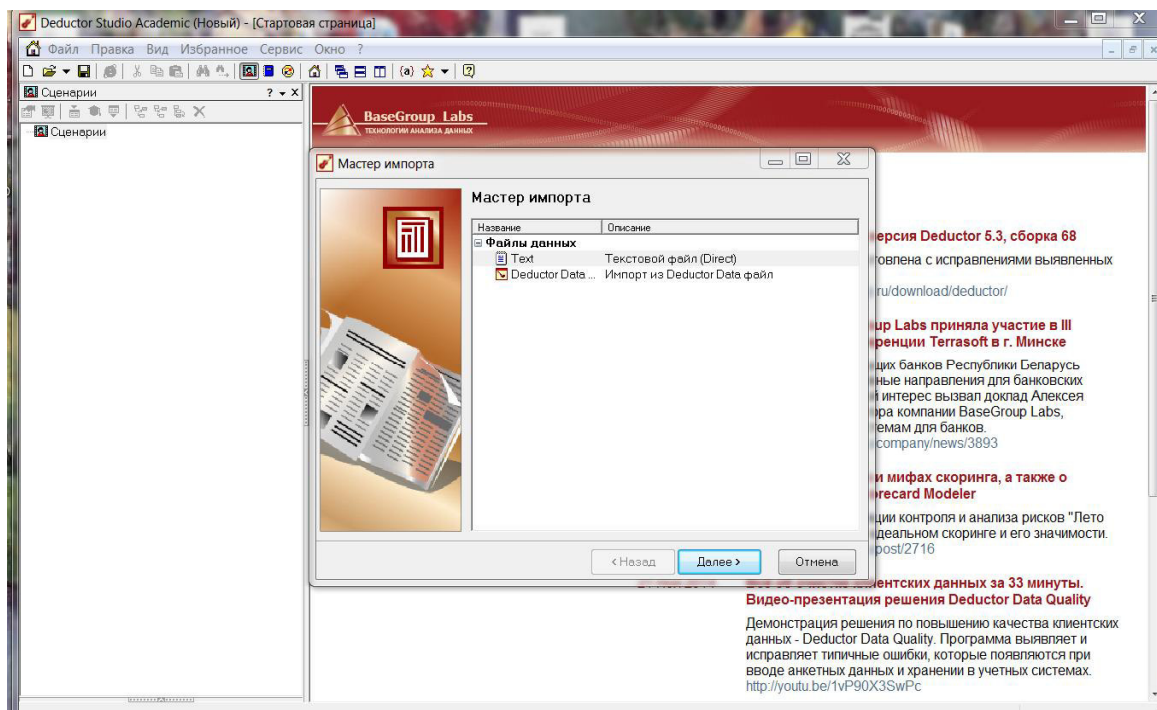


Рисунок 2 - Мастер Импорта



6. После успешного импорта запускаем мастер обработки

(рисунок 3)

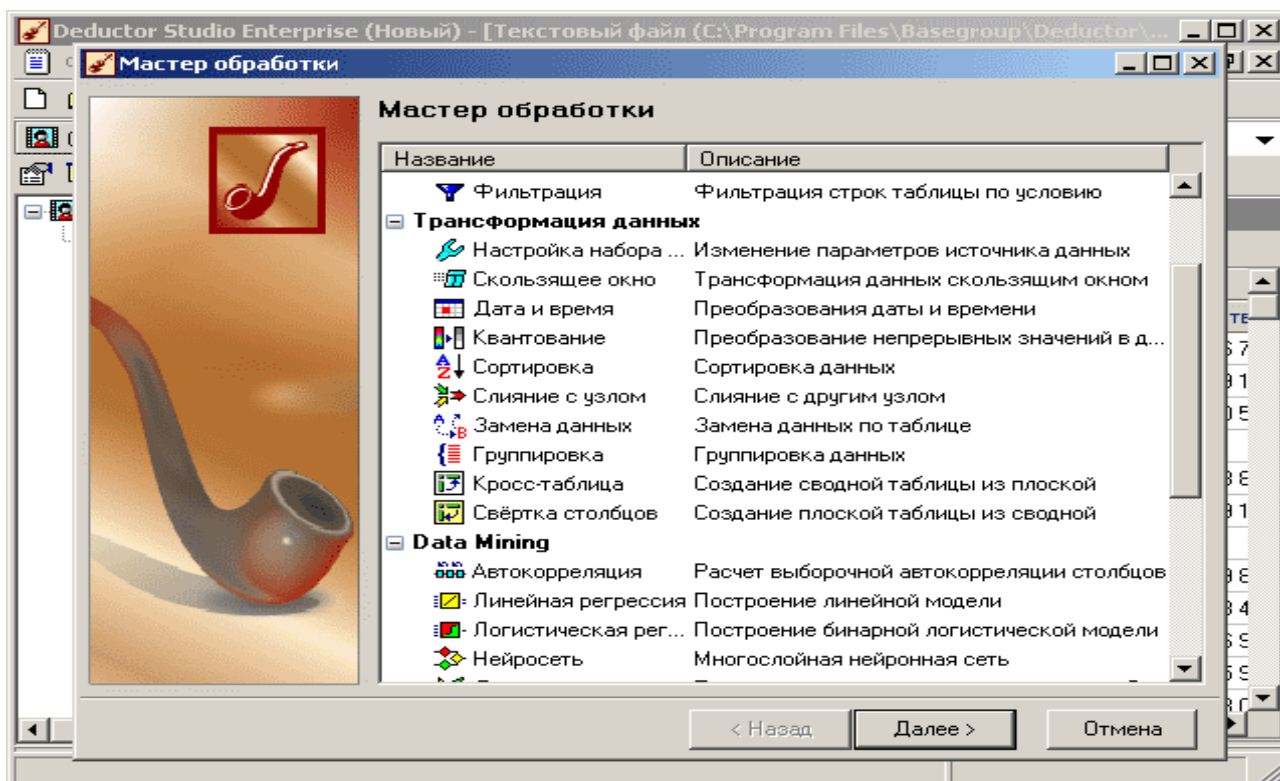


Рисунок 3 - Мастер Обработки

и выбираем метод обработки данных - нейронная сеть. Задаем назначения исходных столбцов данных. Выходной столбец в нашей задаче - "Давать кредит", все остальные - входные. Этот шаг проиллюстрирован на рисунке 4.

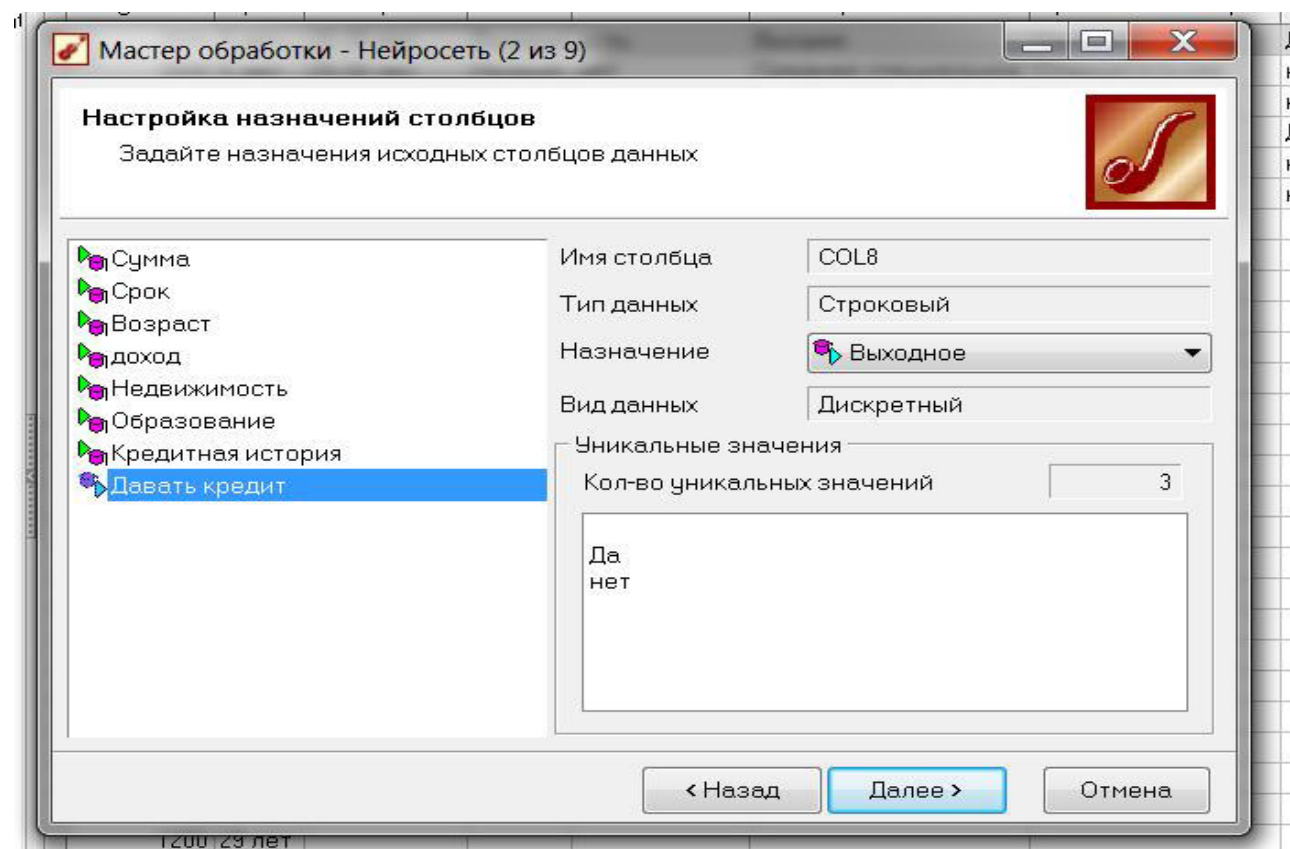


Рисунок 4 - Шаг "Настройка назначений столбцов"

7. На следующем шаге мастер предлагает разбить исходное множество данных на обучающее и тестовое. Способ разбиения исходного множества данных по умолчанию задан "Случайно". Этот шаг представлен на рисунке 5.

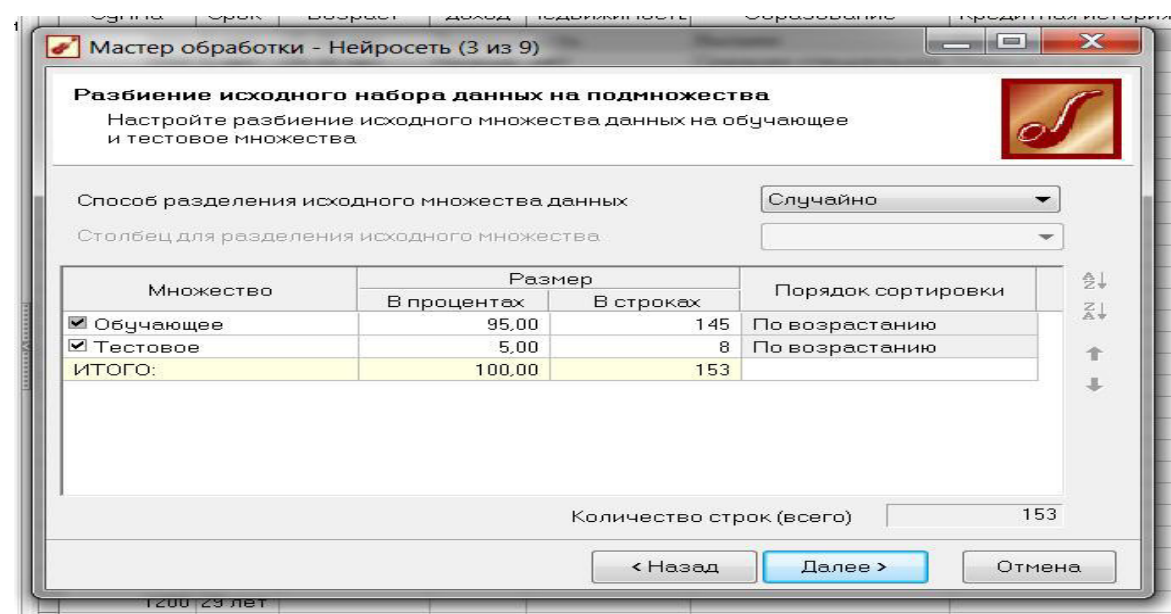


Рисунок 5 - Разбиение набора данных на подмножества

8. На следующем шаге необходимо определить структуру нейронной сети, т.е. указать количество нейронов в входном слое - 33 (количество входных переменных – в примере, в нашем случае - 8), в скрытом слое - 1, в выходном слое- 1 (количество выходных переменных). Активационная функция- Сигмоида, и ее крутизна равна единице. Этот шаг проиллюстрирован на рисунке 6.

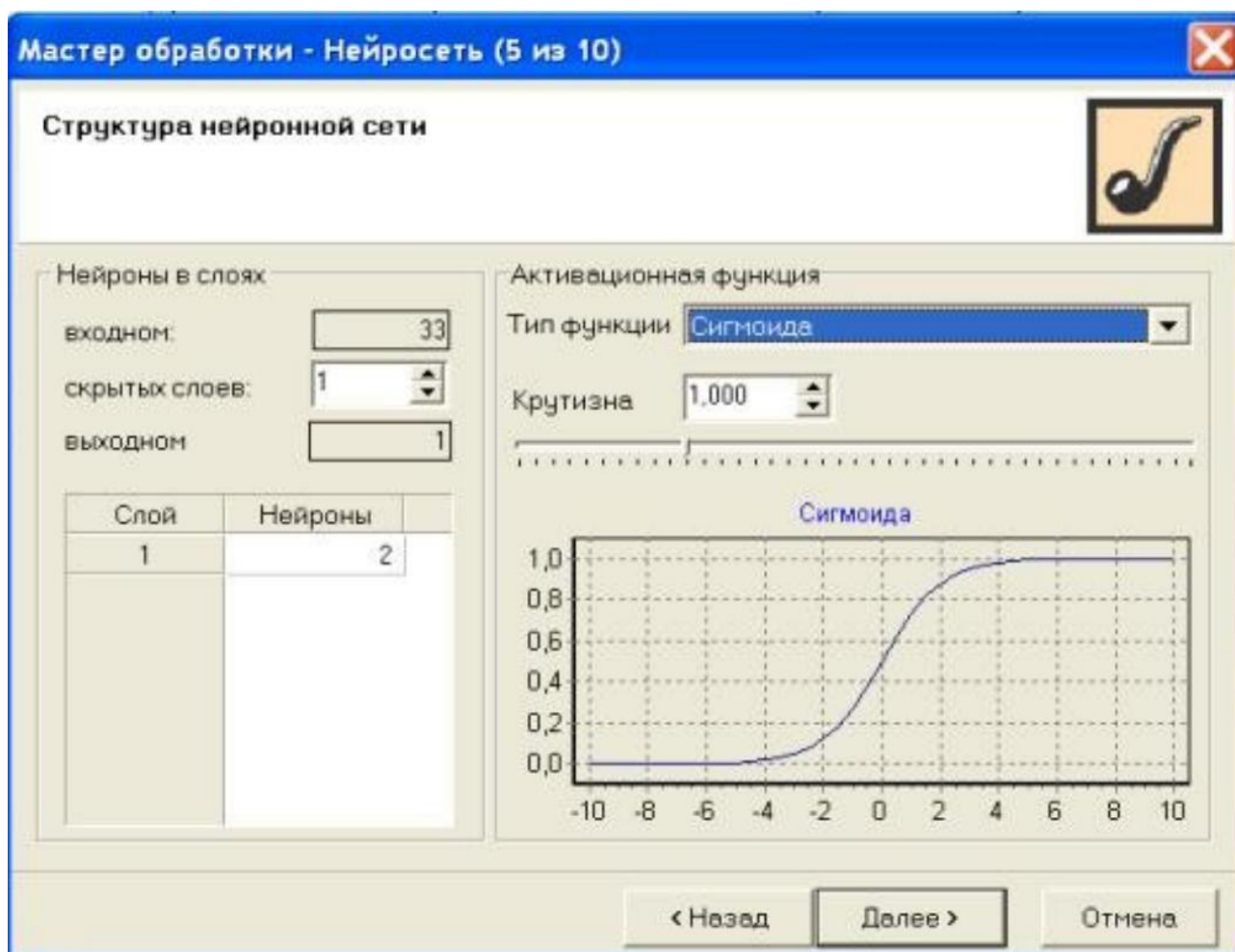


Рисунок 6 - Шаг «Структура нейронной сети»

9. Далее выбираем алгоритм обратного распространения ошибки (Back-Propagation) и параметры обучения нейронной сети. Этот шаг имеет название "Настройка процесса обучения нейронной сети", он представлен на рисунке 7.

10. На следующем шаге настраиваем условия остановки обучения. Будем считать пример распознанным, если ошибка меньше 0,005, и укажем условие остановки обучения при достижении эпохи 10000.

11. На следующем шаге запускаем процесс обучения и наблюдаем за изменением величины ошибки и процентом распознанных примеров в

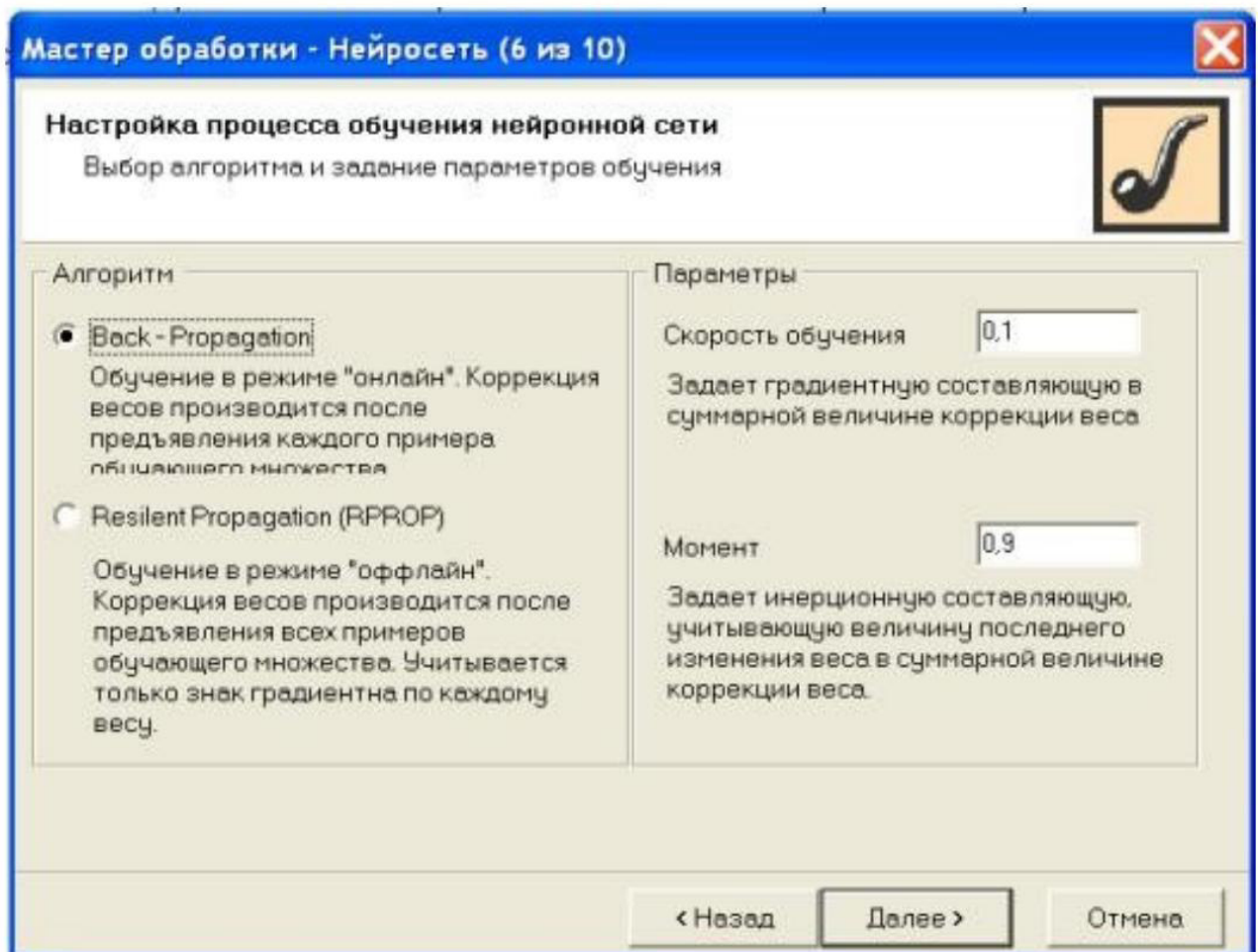


Рисунок 7 - Шаг «Настройка процесса обучения нейронной сети»

обучающем и тестовом множествах. В примере мы видим, что на эпохе № 4536 в обучающем множестве распознано 83,10% примеров, а на тестовом - 85,71% примеров. Фрагмент этого процесса проиллюстрирован на рисунке 8. Результаты вашего эксперимента могут отличаться от приведенных.

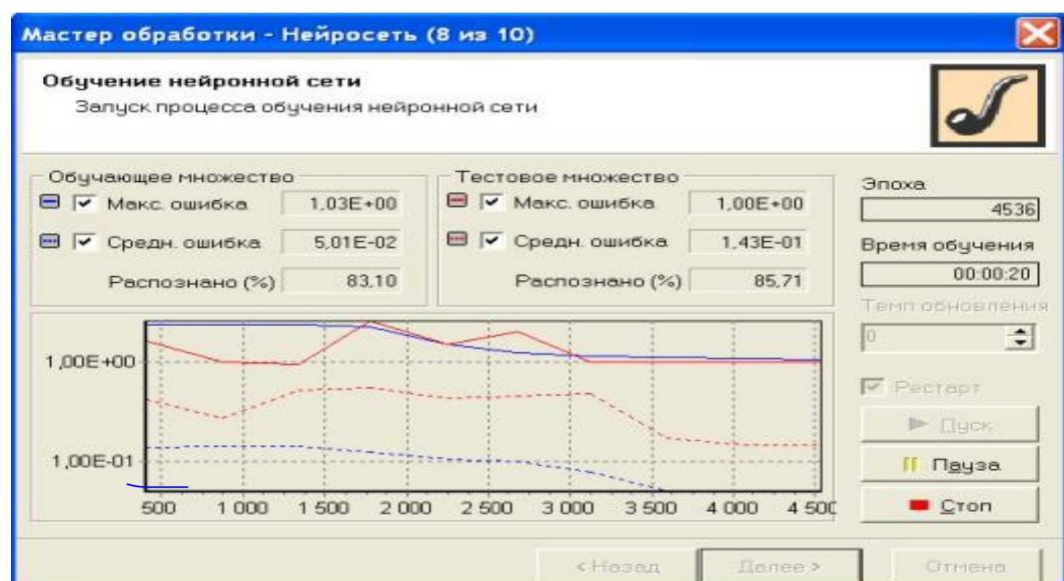


Рисунок 8 - Шаг «Обучение нейронной сети»

12. После окончания процесса обучения для интерпретации полученных результатов мы имеем возможность выбрать визуализаторы из списка предложенных. Выберем такие: таблица сопряженности, граф нейросети, анализ "что, если", и при помощи них проанализируем полученные данные. На рисунке 9 показана таблица сопряженности. По ее диагонали расположены примеры, которые были правильно распознаны, т.е. 55 клиентов, которым можно выдавать кредит, и 89 клиентов, которым выдавать кредит не стоит. В остальных ячейках расположены те клиенты, которые были отнесены к другому классу (1 и 4). Можно считать, что правильно классифицированы практически все примеры - 96,64%.

Кросс таблица				Кросс диаграмма				Граф нейро-сети				Что-если				Таблица сопряженности			
Давать кредит				12															
				Классифицировано															
Фактически				Да	Нет	Итого													
Да				55	4	59													
Нет				1	89	90													
Итого				56	93	149													

Рисунок 9 - Таблица сопряженности

13. Анализ с помощью визуализатора. Визуализатор "что-если" позволяет провести эксперимент. Данные по потенциальному получателю кредита (новые данные, которые не содержатся в базе обучения) следует ввести в соответствующие поля, и построенный линейный классификатор рассчитает значение поля "Давать кредит" - "Да" или "Нет", т.е. решит задачу классификации клиентов для принятия решения о выдаче кредита.

Содержание отчета: цель работы, задание на лабораторную работу, описание эксперимента по созданию, обучению и использованию классификатора, включая скриншоты экранных форм хода эксперимента, выводы.

Список литературы.

1. Чубукова И.А. Data Mining. Курс лекций интернет-университета INTUIT. 328 с. , 2006 г.
2. Применение нейронных сетей для задач классификации. Источник: <http://www.basegroup.ru/library/analysis/neural/classification/>