

## Работа с программой

Deductor — платформа для создания законченных аналитических решений. В платформу встроены современные методы извлечения, визуализации данных и анализа данных.

Для обучения нужно подготовить таблицу с входными значениями и соответствующими им выходными значениями, есть подготовить обучающую выборку. По такой таблице нейросеть сама находит зависимости выходных полей от входных. Далее эти зависимости можно использовать, подавая на вход нейросети некоторые значения. На выходе будут восстановлены зависимые от них значения. Причем на вход можно подавать значения, на которых нейросеть не обучалась.

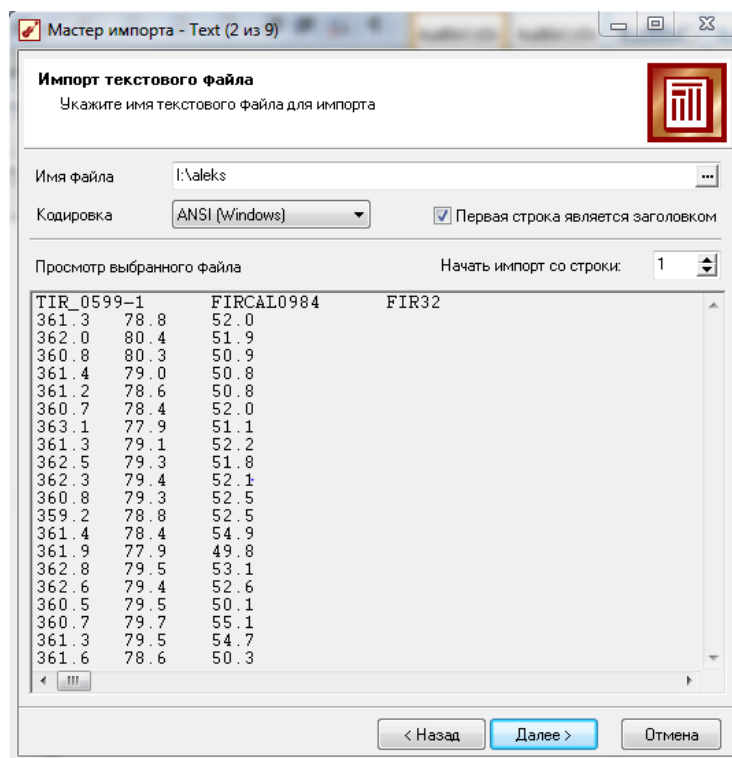


Рис. 3. Импорт таблицы с входными данными

В самом начале работы с нейросетью нужно определиться, что является ее входами, а что – выходами.

На рисунке 4 изображен пример определения входных и выходных значений.

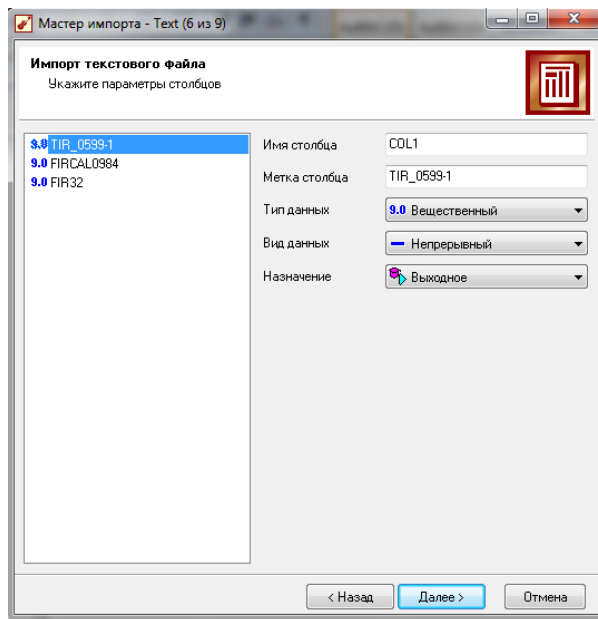


Рис. 4. Определения входных и выходных значений.

Затем происходит сам процесс импорта данных из текстового файла:

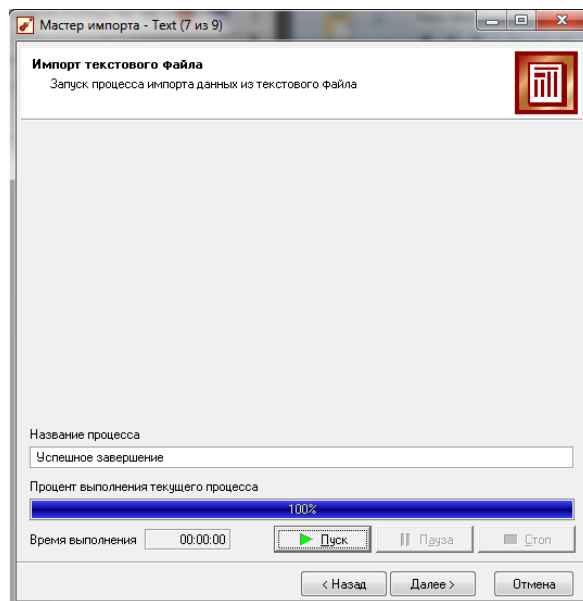


Рис. 5. Импорт данных из текстового файла

После импорта таблицы в программу можно приступить к обучению нейронной сети. Для этого на панели управления выбираем кнопку «Мастер обработки», из появившегося списка во вкладке «Data Mining» нажимаем «Нейронная сеть».

В появившемся окне определяем входы и выходы нейросети, для этого вначале нужно было ее подготовить - редактировать аномалии, заполнить или удалить пропуски, устранить дубликаты и противоречия, произвести квантование и табличную замену, данные привести к скользящему окну, преобразовать формат данных (рис. 6).

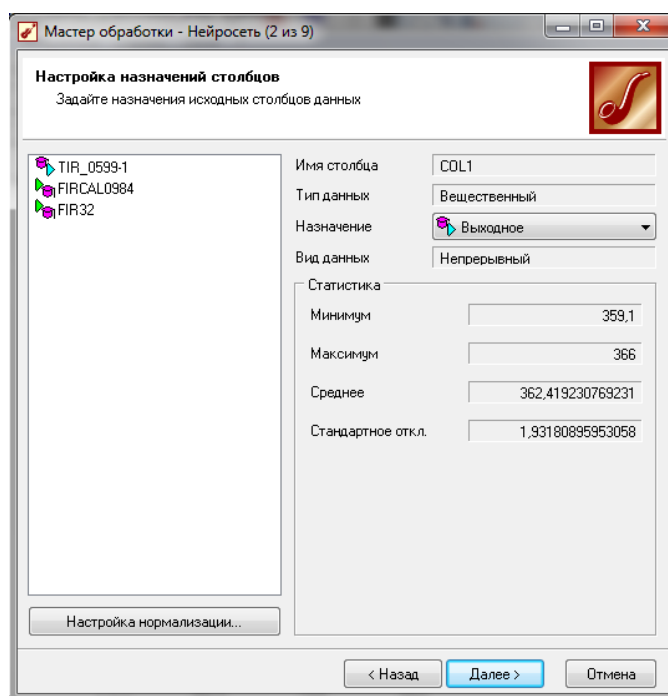


Рис. 6. Настройка назначения полей

После следует настроить обучающую выборку. Обучающую выборку разбивают на два множества – обучающее и тестовое. Обучающее множество включает записи (примеры), которые будут использоваться непосредственно для обучения сети, т.е. будут содержать входные и желаемые выходные (целевые) значения. Тестовое множество также включает записи (примеры), содержащие входные и желаемые выходные (целевые) значения, но используемое не для обучения сети, а для проверки результатов обучения (рис.7).

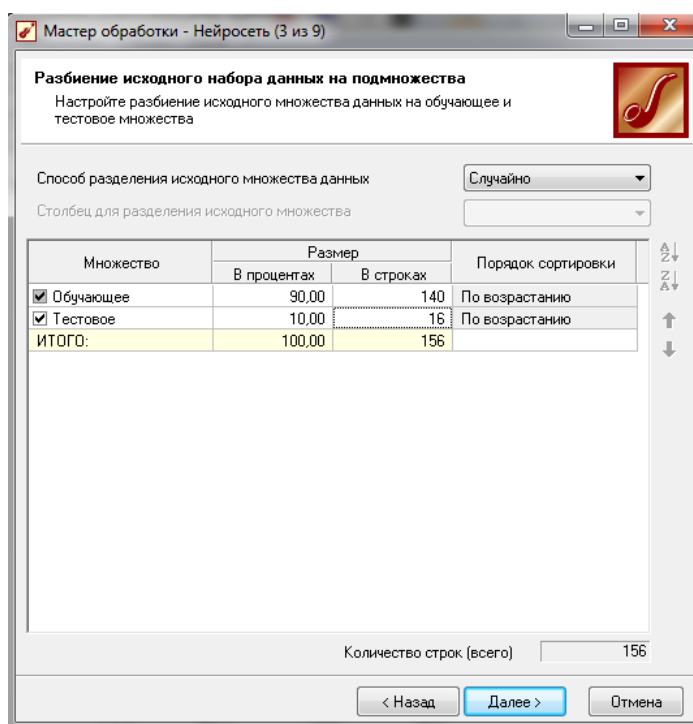


Рис. 7. Настройка обучающей выборки

Далее задаются параметры, определяющие структуру нейронной сети – количество скрытых слоев и нейронов в них, а также активационная функция нейронов.

К выбору количества скрытых слоев и количества нейронов для каждого скрытого слоя нужно подходить осторожно. Для решения многих задач вполне подойдет однослойная нейронная сеть. При выборе количества нейронов следует руководствоваться следующим правилом: «количество связей между нейронами должно быть значительно 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 (рис. 8).

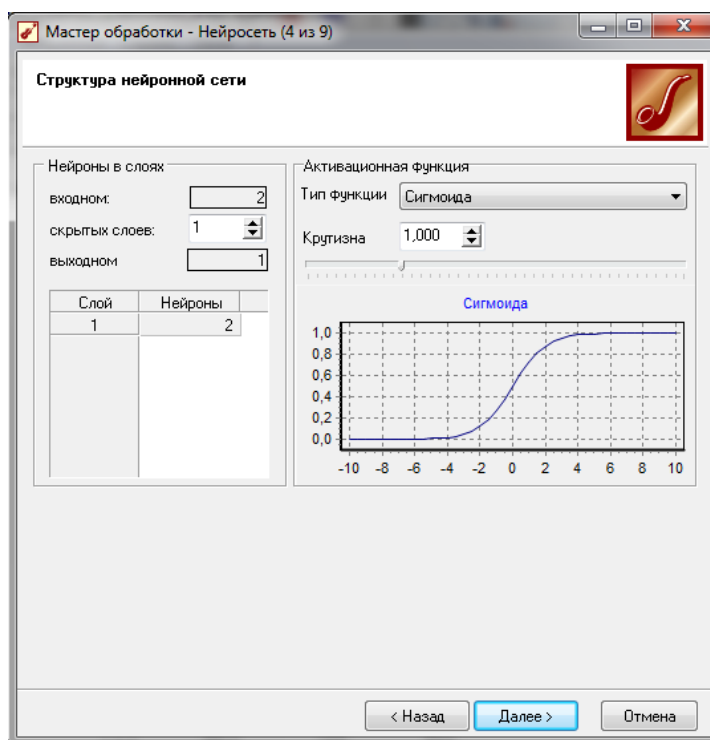


Рис. 8. Настройка структуры нейросети

После настройки конфигурации сети следует выбрать алгоритм ее обучения (рис. 9).

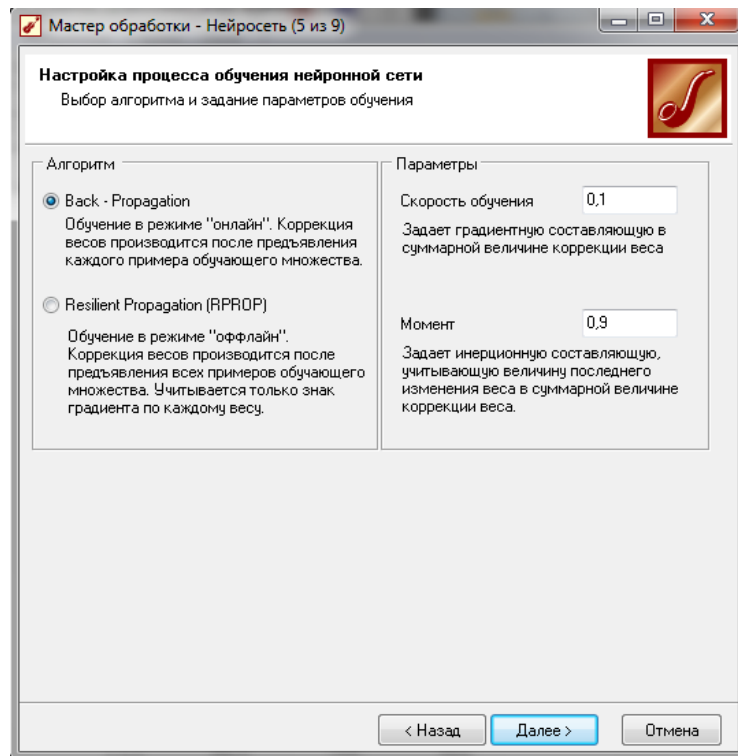


Рис. 9. Обучение нейросети

Метод обратного распространения ошибки – итеративный градиентный алгоритм обучения, который используется с целью минимизации среднеквадратичного отклонения текущих значений выходов сети от требуемых. Одним из важнейших свойств алгоритма обратного распространения ошибки является высокая устойчивость, а, следовательно, надежность.

Для алгоритма обратного распространения ошибки нужно указать два параметра:

- Скорость обучения – определяет величину шага при итерационной коррекции весов в нейронной сети (рекомендуется в интервале  $0...1$ ). При большой величине шага сходимость будет более быстрой, но имеется опасность «перепрыгнуть» через решение. С другой стороны, при малой величине шага обучение потребует слишком многих итераций.

- Момент – задается в интервале  $0...1$ . Рекомендуемое значение  $0,9 \pm 0,1$ .

Далее необходимо задать условия, при выполнении которых обучение будет прекращено (рис. 10).

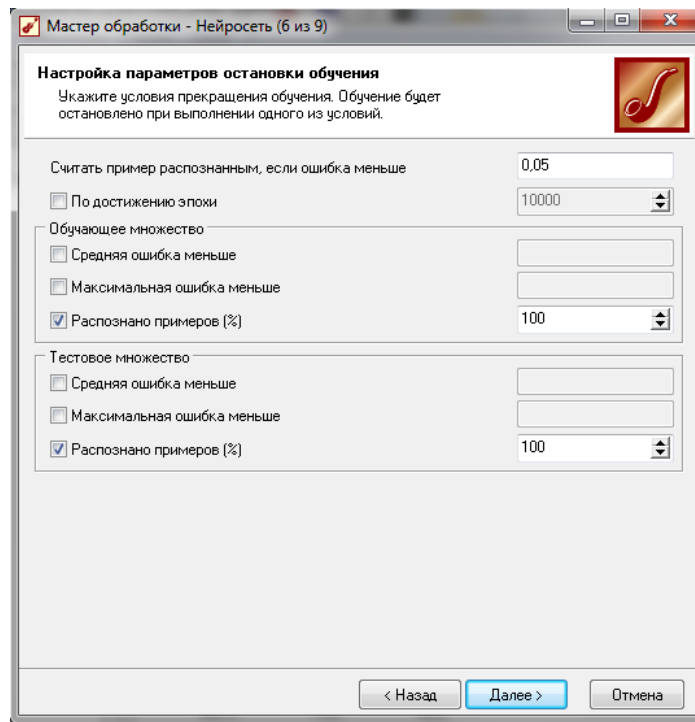


Рис. 10. Параметры остановки обучения

В данной лабораторной работе задавались следующие условия:

- *Считать пример распознанным, если ошибка меньше* – если рассогласование между эталонным и реальным выходом сети становится меньше заданного значения, то пример считается верно распознанным.

- *По достижении эпохи* – установка данного режима 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.

- *Обучающее и тестовое множество* – остановка обучения производится по достижении на обучающем и тестовом множестве заданной средней ошибки, максимальной ошибки или процента распознанных примеров. Распознанным считается пример, для которого отклонение расчетного и реального значения не больше параметра «Считать пример распознанным, если ошибка меньше».

Затем переходим непосредственно к процессу обучения (рис. 11).

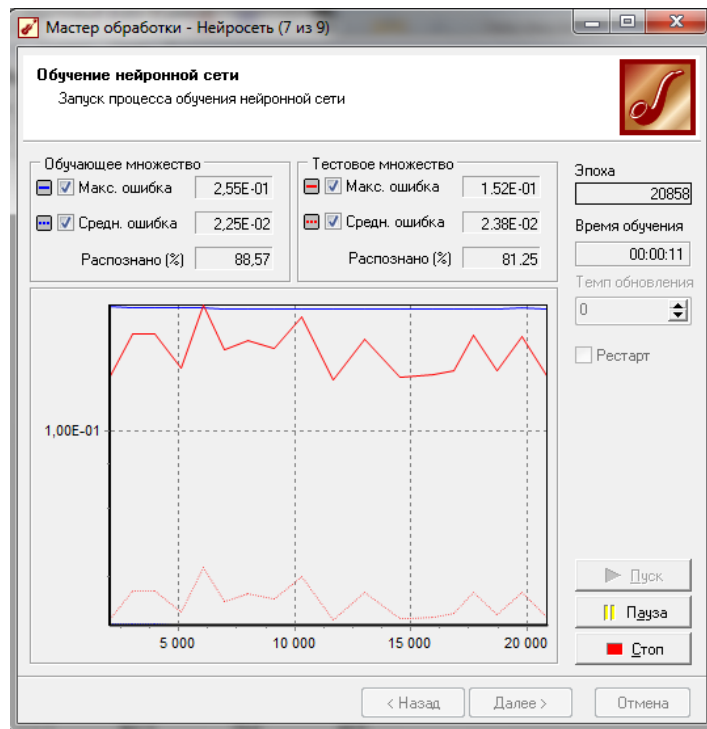


Рис. 11. Запуск процесса обучения нейросети

В начале все веса нейросети инициализируются случайными значениями. После обучения эти веса принимают определенные значения. Обучение может с большой долей вероятности считаться успешным, если процент распознанных примеров на обучающем и тестовом множествах достаточно велик (близок к 100%).

После обучения выбираем способ отображения результатов:

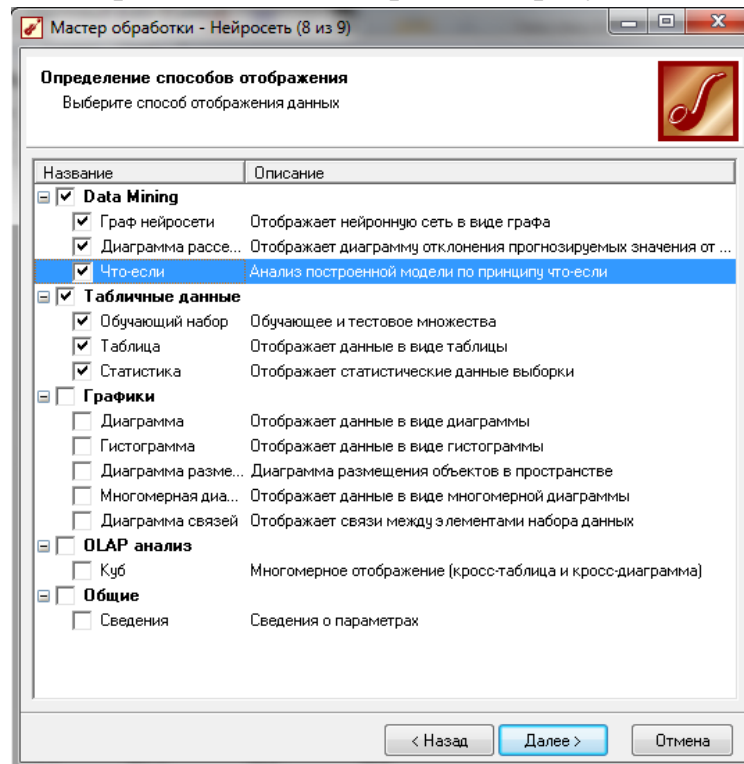


Рис. 12. Выбор способа отображения результатов