

Теоретический материал к лабораторным работам №8-10 и ко всем самостоятельным работам

РАБОТА С ПРОГРАММОЙ НЕЙРОСИМУЛЯТОР

Программа нейросимулятор позволяет проектировать, оптимизировать, обучать и тестировать нейронные сети слоистой структуры, проводить исследования полученных нейросетевых математических моделей, решать с помощью них задачи прогнозирования, оптимизации, управления, распознавания образов. Решение любой задачи методом нейросетевого моделирования, как правило, включает выполнение следующих этапов (рис.1).

Этап 1. Постановка задачи.

На этом этапе определяются цели моделирования, устанавливаются входные и выходные параметры модели, устанавливается структура (состав и длина) входного вектора X , и выходного вектора D .

В качестве компонент входного вектора X важно выбрать значимые параметры, т.е. те, которые оказывают существенное влияние на результат. Если есть сомнения в значимости того или иного входного параметра, то его лучше включить во входной вектор, рассчитывая, что в последующем с помощью создаваемой нейросети можно будет оценить степень его влияния на результат и, если она окажется слабой, то этот параметр исключить.

Выходной вектор D формируется таким, чтобы его компоненты давали возможность получить ответы на все поставленные вопросы.

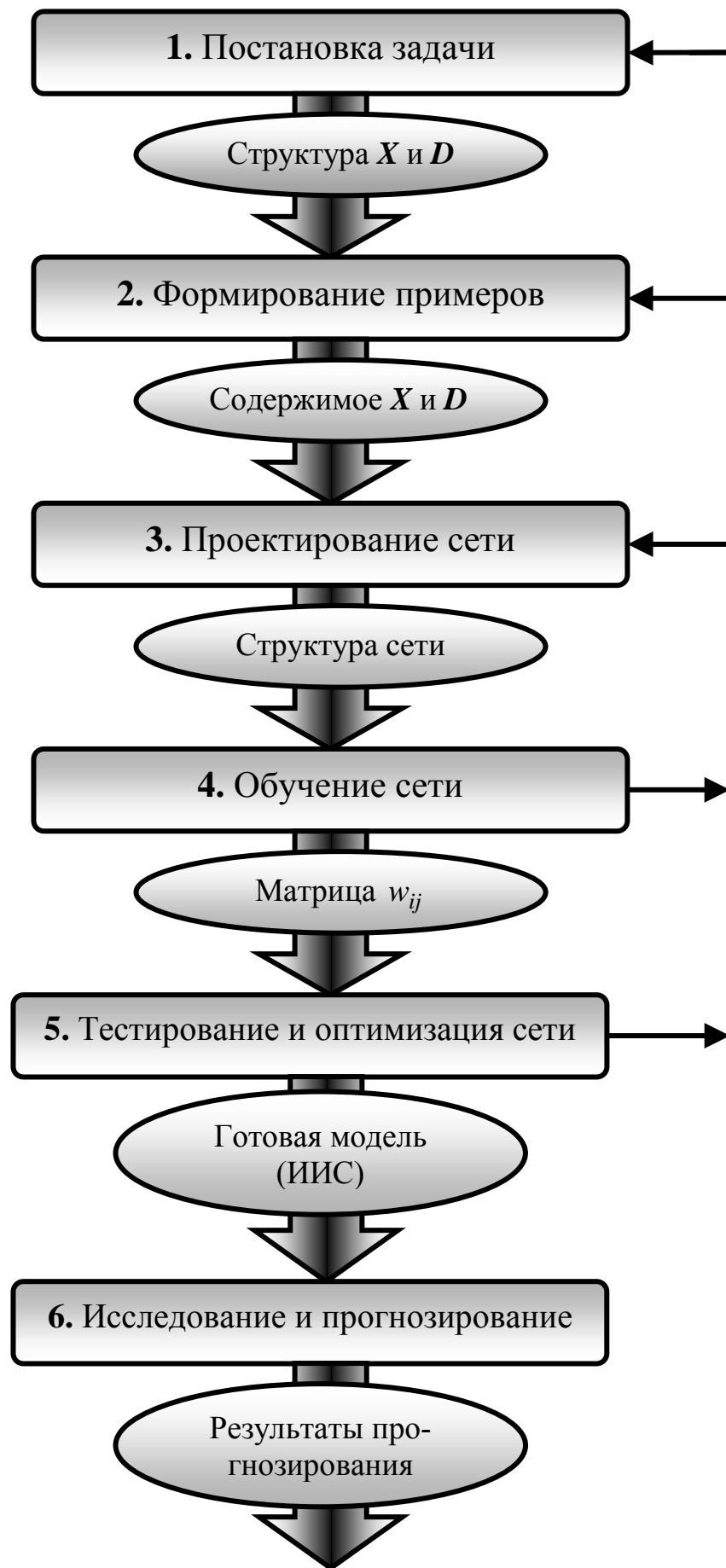


Рис. 1. Алгоритм применения метода нейросетевого математического моделирования

Компоненты входного вектора X и выходного вектора D представляют собой числа. Это могут быть значения каких-либо величин, например, температура тела, артериальное давление, частота пульса и др. Это могут быть также числа, кодирующие наличие или отсутствие каких-либо признаков, например, единица, если пол мужской и двойка, если пол женский. В некоторых случаях, если данные нечеткие и есть сомнение в их правильности, полезно кодировать оценку их вероятности. Например, если у врача есть сомнения в правильности выставляемого диагноза, он может кодировать не сам диагноз, а его вероятность, или степень развития болезни, применяя десяти- или стобалльную систему оценки (см. лабораторную работу №7).

Этап 2. Формирование примеров.

На этом этапе формируется содержимое входных и выходных векторов. В результате создается множество пар $X_q - D_q$ ($q = 1, \dots, Q$). Каждая такая пара составляет пример, характеризующий предметную область.

Значения компонент векторов X_q и D_q формируют различными способами: путем проведения социологических исследований, анкетирования, специальных экспериментов над предметной областью, берут из сети Интернет, из СМИ, из архивных материалов организаций и из других источников.

Все множество примеров разбивают на обучающее L и тестирующее T (рис. 2). Обычно объем тестирующего множества выбирают не менее 10% от обучающего. Какой необходим минимальный объем обучающего множества, зависит от задачи. Обычно рекомендуется не менее пятидесяти примеров. Однако в нашей практике встречались случаи, когда для решения задачи хватало и десяти обучающих примеров.

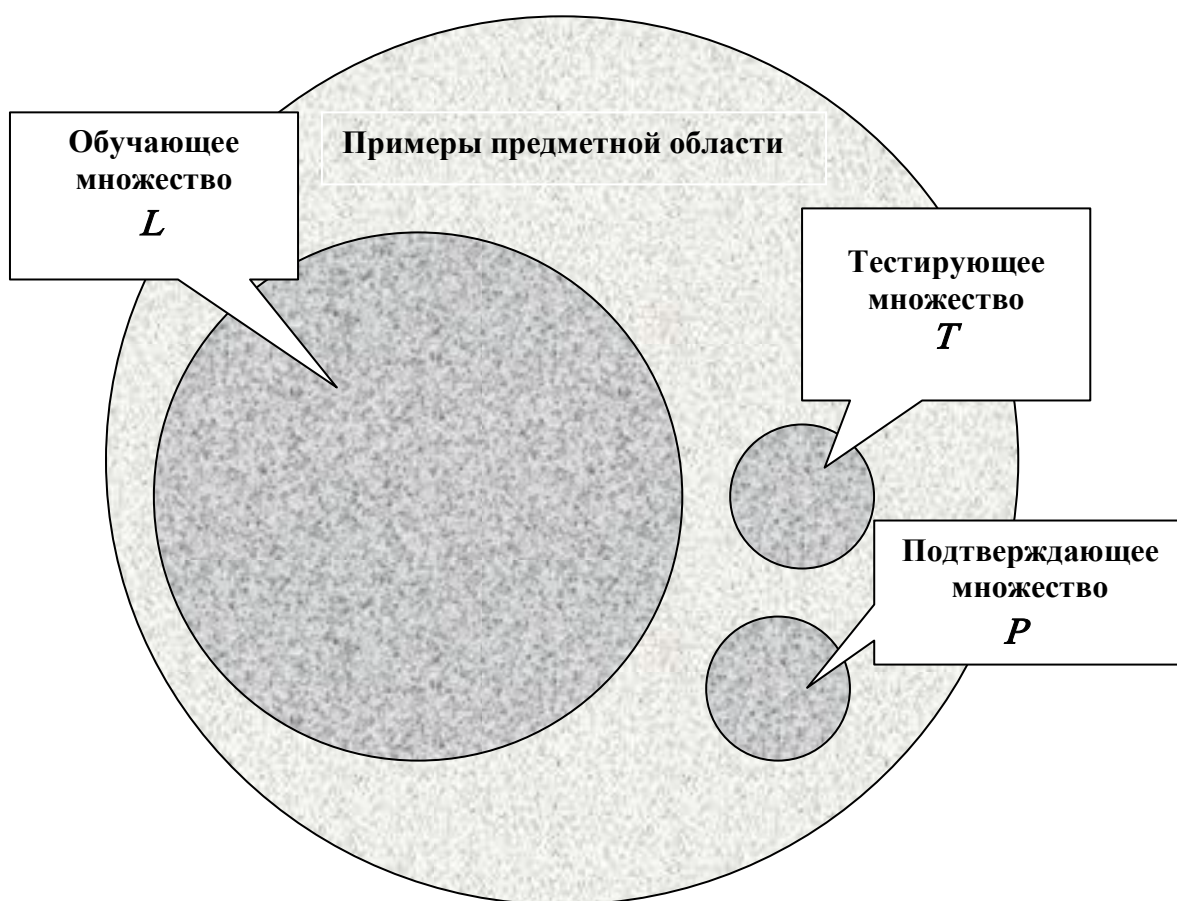


Рис. 2. Деление примеров предметной области на обучающее множество L , тестирующее множество T и подтверждающее множество P

В особо ответственных случаях рекомендуется помимо обучающего множества L и тестирующего множества T , формировать еще и подтверждающее множество P из примеров, принадлежащих той же самой предметной области, но не пересекающееся ни с множеством L , ни с множеством T (см. рис. 2).

Этап 3. Проектирование сети.

Структура персептрона выбирается из следующих соображений.

Число входных нейронов N_x должно быть равно размерности входного вектора X .

Число выходных нейронов N_y должно быть равно размерности выходного вектора D .

Число скрытых слоев, согласно теореме Арнольда – Колмогорова – Хехт-Нильсена, должно быть не менее одного. На последующих этапах число скрытых слоев может корректироваться, если это позволит улучшить качество работы сети.

Число нейронов в скрытых слоях N рассчитывается с помощью формул, являющихся следствием из теоремы Арнольда – Колмогорова – Хехт-Нильсена:

$$\frac{N_y Q}{1 + \log_2(Q)} \leq N_w \leq N_y \left(\frac{Q}{N_x} + 1 \right) (N_x + N_y + 1) + N_y,$$

$$N = \frac{N_w}{N_x + N_y},$$

где N_y – размерность выходного сигнала; Q – число элементов множества обучающих примеров; N_w – необходимое число синаптических связей; N_x – размерность входного сигнала.

На последующих этапах число нейронов в скрытых слоях может корректироваться, если это позволит улучшить качество работы сети.

Активационные функции скрытых нейронов, согласно теореме Арнольда – Колмогорова – Хехт-Нильсена, рекомендуется задать сигмоидными, однако в дальнейшем, их вид может быть изменен, если это позволит улучшить качество работы сети.

При корректировке структуры персептрона следует иметь ввиду, что увеличение скрытых нейронов обычно позволяет добиться меньшей ошибки обучения, однако чрезмерное их увеличение приводит к эффекту гиперразмерности – потере обобщающих свойств сети, выражающемуся в возрастании ошибки обобщения.

Этап 4. Обучение сети.

Обучение сети – очень важный, но не окончательный этап создания нейросетевой интеллектуальной системы. Цель обучения – подобрать синаптические веса w_{ij} так, чтобы на каждый входной вектор X_q множества обучающих примеров сеть выдавала вектор Y_q , минимально отличающийся от заданного выходного вектора D_q . Эта цель достигается путем использования алгоритмов обучения нейронной сети. Обычно пробуются сразу несколько алгоритмов обучения. Характерная кривая обучения – зависимость квадратичной и максимальной ошибок обучения от числа эпох обучения, приведена на рис. 3.

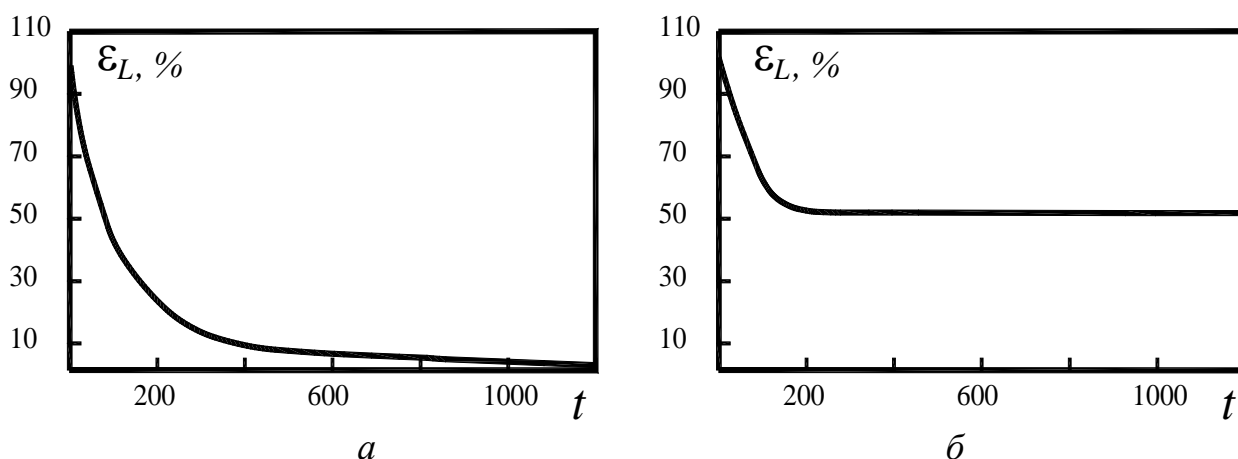


Рис. 3. Характерные зависимости ошибок обучения от числа эпох в случае, когда сеть обучается успешно (а), и когда процесс обучения не дает желаемого результата (б)

Но может случиться так, что сеть не захочет обучаться – ошибка обучения с увеличением числа эпох не будет стремиться к нулю (см. рис. 3, б). Рассмотрим возможные причины этого нежелательного явления.

1-я причина. Недостаточное количество скрытых слоев и скрытых нейронов.

Рекомендуется увеличить число скрытых слоев и скрытых нейронов.

2-я причина. Попадание в локальный минимум.

Рекомендуется «встряхнуть» сеть, начать обучение с другими начальными значениями сил синаптических связей, или сменить метод обучения.

3-я причина. Попадание в овраг.

Рекомендуется использовать антиовражный метод обучения, например, метод обратного распространения с моментом.

4-я причина. Наличие в обучающем множестве посторонних выбросов, выпадающих из общих закономерностей предметной области.

Рекомендуется обнаружить посторонние выбросы, постараться убедиться, что они действительно являются не характерными для предметной области, являются следствием ошибок, неточности измерений и др., и удалить посторонние выбросы из обучающего множества примеров.

5-я причина. Наличие в обучающем множестве противоречащих друг другу примеров. Например, одному и тому же вектору симптомов соответствуют разные диагнозы заболеваний.

Обнаружить такие примеры в обучающем множестве можно путем его визуального анализа, либо путем применения тех же методик, которые используются при обнаружении посторонних выбросов. Затем следует разо-

браться в причинах возникновения противоречащих примеров. Некоторые из примеров могут просто оказаться ошибочными, и их нужно удалять. Другая причина может быть связана с тем, что в самой структуре входного вектора отсутствуют какие-то параметры (например, возраст больного, рост, вес, цвет его глаз и др.), также оказывающие влияние на результат. В этом случае рекомендуется пересмотреть постановку задачи, увеличить размерность входного вектора X , добавив дополнительные параметры, которые своими значениями обеспечат непротиворечивость примеров обучающего множества.

6-я причина. Паралич сети.

Рекомендуется сменить сигмоидную активационную функцию на логарифмическую, либо на функцию Ле-Кана, или сменить способ предобработки входных параметров.

7-я причина. Слишком большая скорость обучения.

Рекомендуется уменьшить скорость обучения.

Этап 5. Тестирование (проверка) и оптимизация сети.

Цель тестирования – убедиться, что математическая нейросетевая модель адекватна моделируемой предметной области и пригодна для последующего использования. Цель оптимизации – добиться наилучших обобщающих свойств сети, т.е. – минимальной ошибки обобщения.

Проверка обобщающих свойств производится на тестирующем множестве примеров, т.е. на тех примерах, которые не участвовали в обучении сети. Результаты тестирования полезно представить графически в виде гистограммы, на которой значения желаемых выходов персептрона (D_q) можно сопоставить с прогнозными (Y_q), т.е. теми, которые вычислил персептрон. Пример такой гистограммы приведен на рис. 4.

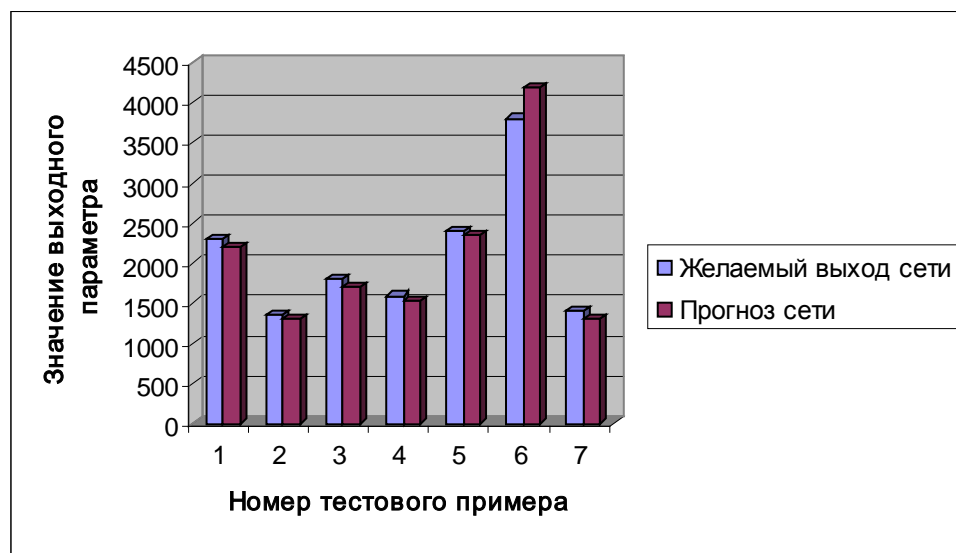


Рис. 4. Пример гистограммы, показывающей соотношение желаемых выходов сети с прогнозными

Если разница между компонентами желаемого выходного вектора тестирующего множества примеров D_q и прогнозными значениями Y_q окажется незначительной, то можно переходить к следующему **этапу 6**, не выполняя оптимизацию сети. Однако чтобы лишний раз убедиться в адекватности разрабатываемой нейросетевой математической модели полезно вернуться на **этап 2** и те примеры, которые были тестирующими, (либо часть тестирующих примеров) включить в обучающее множества, а часть примеров, бывших обучающими, сделать тестирующими. После этого снова повторить **этапы 3, 4, 5**.

Если погрешность обобщения сети окажется неприемлемо большой, можно попытаться оптимизировать сеть. Оптимизация сети состоит в подборе наиболее подходящей для данной задачи структуры сети – количества скрытых слоев, количества скрытых нейронов, количества синаптических связей, вида и параметров активационных функций нейронов. В некоторых нейропакетах предусмотрена автоматическая оптимизация сети. Но иногда бывает полезно выполнить оптимизацию вручную, построив график зависимости погрешности обобщения от числа скрытых нейронов (см. рис. 5) и других параметров персептрона, и выбрав с помощью этих графиков структуру сети, обеспечивающую минимальную погрешность обобщения.

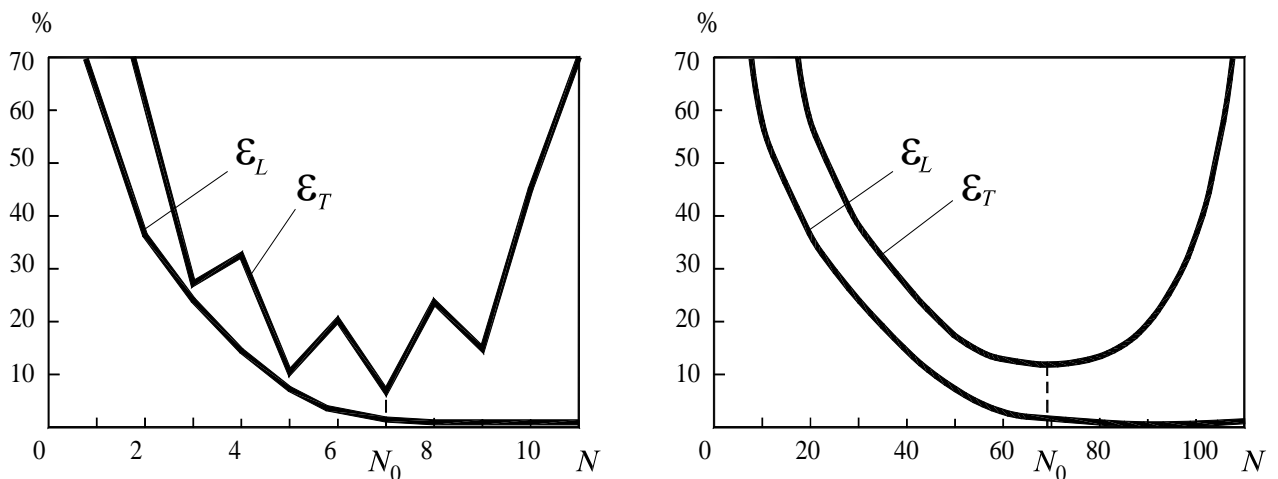


Рис. 5. Характерные зависимости ошибки обучения ϵ_L и ошибки обобщения (тестирования) ϵ_T от количества нейронов скрытых слоев персептрона N

При выборе оптимальной структуры сети следует помнить, что цель оптимизации сети состоит в минимизации погрешности обобщения ϵ_T , но не

погрешности обучения ε_L . Именно по величине погрешности обобщения судят о качестве сети, о ее обобщающих и, следовательно, прогностических свойствах. Погрешность же обучения – это всего лишь промежуточный результат. Желательно, чтобы она была небольшой, но добиваться ее минимального значения вовсе не обязательно и даже вредно, поскольку, как это видно из рис. 5, приводит к эффекту гиперразмерности (переобучения), т.е. росту погрешности обобщения.

Понятно, что оптимизация нейронной сети подразумевает многократные возвраты назад – на **этапы 4, 3, 2, 1**.

После оптимизации сети, ее обобщающие свойства рекомендуется проверить на примерах подтверждающего множества P . Дело в том, что в процессе оптимизации сеть могла приспособиться к примерам тестирующего множества. А если эти примеры по каким-либо причинам не характерны для всей предметной области, то на других примерах, которых не было ни в тестирующем, ни в обучающем множествах, она может дать неожиданно большую ошибку прогнозирования. Для исключения такого явления, и чтобы окончательно убедиться, что сеть имеет хорошие обобщающие свойства не только на тестирующем множестве примеров, вычисляют ошибку прогноза сети ε_P на подтверждающем множестве, т.е. на тех примерах, которые ни в обучении, ни в тестировании не участвовали.

Результатом оптимизации и проверки сети является готовая к использованию нейросетевая математическая модель предметной области – интеллектуальная информационная система.

Этап 6. Исследование модели, прогнозирование.

Путем проведения вычислительных экспериментов над математической нейросетевой моделью достигаются цели моделирования, находятся ответы на все поставленные вопросы. Например, могут быть решены такие задачи, как оптимизация моделируемого объекта, прогнозирование его будущих свойств, выявление закономерностей предметной области и др.

Нейросетевая математическая модель, если она правильно спроектирована и обучена, впитала в себя закономерности моделируемой предметной области. Она реагирует на изменение входных параметров и ведет себя так же, как вела бы себя сама предметная область.

И надо поставить над моделью как можно больше экспериментов. Надо постараться извлечь из этих виртуальных экспериментов как можно больше полезной информации.

Таким образом, в результате работы с программой НЕЙРОСИМУЛЯТОР, в результате выполнения приведенного на рис. 1 алгоритма работы с этой программой получают два вида продуктов:

- после выполнения **этапа 5** создается готовая к использованию ***интеллектуальная информационная система***, являющаяся ***математической моделью предметной области***,
- после выполнения **этапа 6** получают ***результаты исследования математической модели***, представленные в виде графиков, номограмм, гистограмм, полезных рекомендаций и выводов.