

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
**«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ  
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ»**  
(НИУ «Б е л Г У»)

ИНСТИТУТ ИНЖЕНЕРНЫХ ТЕХНОЛОГИЙ И ЕСТЕСТВЕННЫХ НАУК

КАФЕДРА МАТЕМАТИЧЕСКОГО И ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ  
ИНФОРМАЦИОННЫХ СИСТЕМ

**ИСПОЛЬЗОВАНИЕ НЕЙРОННЫХ СЕТЕЙ ДЛЯ  
ПРОГНОЗИРОВАНИЯ ПОТРЕБИТЕЛЬСКОГО СПРОСА В  
СЕКМЕНТЕ СТРОИТЕЛЬНЫХ ТОВАРОВ**

Магистерская диссертация  
обучающейся по направлению подготовки 02.04.01 Математика и  
компьютерные науки очной формы обучения, группы 07001531  
Помётовой Натальи Николаевны

Научный руководитель  
к.т.н., доцент  
Чашин Ю.Г.

Рецензент  
к.т.н., доцент  
Заливин А.Н.

БЕЛГОРОД 2017

## СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	4
1. ТЕОРЕТИЧЕСКИЕ ОСНОВЫ НЕЙРОСЕТЕВОГО ПРОГНОЗИРОВАНИЯ.....	6
1.1. Актуальность задачи прогнозирования.....	6
1.2. Нейронные сети как средство анализа процесса.....	8
1.3. Классификация нейронных сетей по количеству слоев .....	10
1.3.1. Однослойные нейронные сети .....	10
1.3.2. Многослойные нейронные сети.....	11
1.4. Обучение искусственной нейронной сети .....	11
1.5. Методы нейросетевого прогнозирования .....	13
1.5.1. Нейросети обратного распространения .....	13
1.5.2. Сеть Кохонена .....	18
1.5.3. Гибридные нейронные сети .....	22
2. НЕЙРОСЕТЕВОЕ ПРОГНОЗИРОВАНИЕ СПРОСА СТРОИТЕЛЬНЫХ ТОВАРОВ .....	27
2.1. Выбор программного средства для решения задачи.....	27
2.1.1. Службы Microsoft Analysis Services.....	27
2.1.2. Платформа Microsoft BI Solution builder .....	28
2.1.3. Oracle Data Mining Suite .....	29
2.1.4. Статистический пакет Statistical Package for the Social Sciences (SPSS).....	30
2.1.5. Oracle Crystal Ball.....	31

2.1.6. Пакет MATLAB.....	32
2.2. Обоснование выбора программного средства для исследования.....	33
2.3. Выбор архитектуры и метода обучения нейронной сети в системе Matlab.....	34
2.3.1. Модель обучения нейронной сети .....	34
2.3.2. Алгоритм обучения метода обратного распространения .....	36
2.3.3. Гибридный алгоритм обучения (MATLAB ANFIS).....	38
2.4. Моделирование нейронной сети для прогнозирования потребительского спроса строительных товаров .....	39
2.4.1. Настройка сети обратного распространения .....	39
2.4.3. Настройка гибридной нейронной сети .....	40
3. СБОР И АНАЛИЗ ПОЛУЧЕННЫХ РЕЗУЛЬТАТОВ .....	41
3.1. Информационное обеспечение задачи. Данные о продажах.....	42
3.2. Проведение эксперимента.....	47
3.3. Сравнительный анализ методов .....	58
ЗАКЛЮЧЕНИЕ .....	60
БИБЛИОГРАФИЧЕСКИЙ СПИСОК.....	61

## ВВЕДЕНИЕ

В современном мире, с постоянно меняющимися экономическими условиями, спросом и предложениями, невозможно заниматься бизнесом на основе лишь накопленного собственного опыта. Следует анализировать текущее состояние рынка, тенденции его развития прогнозировать возможные его изменения, а также успешность своей деятельности в будущем.

В нестабильных условиях рыночной экономики деятельность любого предприятия в значительной степени зависит от того, насколько точно и достоверно оно может предвидеть перспективы своего развития в будущем, т.е. осуществлять прогнозирование продаж, а, соответственно, и прибыли. Прогноз является одной из главных обязанностей любого руководителя, так как это важная составляющая успеха компании.

Аппарат нейронных сетей – мощный инструмент для прогнозирования потребительского спроса. Нейронные сети позволяют отыскать скрытые зависимости между показателями и получить при этом наиболее точный прогноз данных, потому и является более предпочтительным.

Целью магистерской диссертации является выявление оптимальной методики прогнозирования с использованием нейронных сетей в отрасли строительных материалов.

Для достижения поставленной цели необходимо решить ряд следующих задач:

- изучить нейросетевые методы прогнозирования;
- выбрать несколько методов нейросетевого прогнозирования для проведения эксперимента;
- провести анализ программных средств, позволяющий выполнить

прогнозирование методом нейронных сетей, обосновать выбор программного средства;

- провести анализ временных рядов продаж строительных товаров определить их характер (стационарность, сезонность, и др. показатели);
- провести отбор наиболее значимых факторов, влияющих на объемы продаж строительных товаров по их статистическим оценкам;
- сформировать входные данные для обучения НС;
- провести эксперимент для определения методики обучения сети, позволяющей спрогнозировать продажи строительных товаров наиболее точно.

Структура магистерской диссертации обусловлена целью и задачами исследования. Работа состоит из введения, трех глав и заключения.

Введение раскрывает актуальность задачи прогнозирования, определяет степень научной разработки темы, объект, предмет, цель и задачи, раскрывает теоретическую и практическую значимость работы.

В первой главе рассматриваются нейросетевые методы для прогнозирования.

Во второй главе осуществляется выбор программного средства, а так же настройка нейронной сети для прогноза.

Третья глава посвящена сбору входных данных, проведению эксперимента, а так же сравнительному анализу результатов.

В заключении подводятся итоги исследования, формируются окончательные выводы по рассматриваемой теме.

Магистерская диссертация содержит 64 страницы, 38 рисунков, 2 таблицы, 1 листинг.

# 1. ТЕОРЕТИЧЕСКИЕ ОСНОВЫ НЕЙРОСЕТЕВОГО ПРОГНОЗИРОВАНИЯ

## 1.1. Актуальность задачи прогнозирования

«Того, кто не задумывается о далеких трудностях, непременно поджидают близкие неприятности» - писал Конфуций.

Прогнозирование происходящего в природе и обществе свойственно человеку с момента его появления на Земле. Одним из наиболее древних известных источников применения прогнозирования будущего является книга «Прогностика», написанная Гиппократом более 2000 лет назад.

В современном мире роль прогнозирования в жизни общества значительно возросла. Оно влияет не только на жизнь конкретного человека в отдельности, но и на существование различных организаций и управление ими. В начале XX века А. Файоль описывал прогнозирование как самостоятельную функцию управления. Именно ему принадлежат слова «Управлять – значит: предсказывать, планировать, организовывать». [26]

Как и любой человек, живущий в обществе, управленец постоянно должен прогнозировать последствия своих решений и действий. Умение прогнозирования у человека может быть развито в большей или меньшей степени, но без него его шансы на выживание в окружающей среде минимальны. [13] Это касается и существования организаций во внешней среде, которое, в первую очередь, зависит от системы управления конкретной организации. Чем выше уровень прогнозирования в организации, тем больше вероятность осуществления оптимального эффективного управления ею с целью сохранения ее целостности и обеспечения ее развития.[21]

Статистика проведенных исследований показывает, что эффективность принимаемых управленческих решений тесно связано с прогнозированием, применяемым в организации. Так, в одном из них были получены данные в результате опроса 217 вице-президентов из 109 крупнейших американских компаний. Было выявлено, руководители, которые использование прогнозирование при составлении планов деятельности организации, добились большего успеха в соотношении получаемой прибыли к объему реализации товара/услуги и, в конечном итоге, в общем доходе от деятельности организации. [26]

Прогнозирование в управлении организацией необходимо для:

- определения социально-экономических задач, которые может поставить и решить организация в течение прогнозируемого периода времени;
- выявления тенденций используемых технологий и научных разработок и их влияния на будущую деятельность организации;
- выявления всех возможных вариантов путей развития науки, техники, экономики, культуры и выбор наиболее оптимального пути развития каждого из них в отношении организационной деятельности;
- определения материальных, трудовых, природных ресурсов, которыми будет располагать организация на прогнозируемый период времени;
- выявления потребностей экономики и общества в продаваемой продукции.

В итоге, прогнозирование дает организациям следующие преимущества:

- предоставление организации дополнительного конкурентного преимущества на внешнем рынке, которое повышает адаптивные способности к выживанию организации, а также обеспечивает расширение горизонта принимаемых управленческих решений;
- снижение неопределенности и рисков в развитии организации, а также обеспечение опережающей информацией;

- обеспечение грамотного опережающего обучения персонала, т.е. грамотное осуществление инвестиций в «человеческий капитал»;
- принятие более взвешенных и грамотных управленческих решений;
- усиление эффективности (оптимальное использование ресурсов) и продуктивности (достижение поставленных целей) управления.

На основании проведенного исследования можно сделать следующий вывод: прогнозирование является одной из важнейших функций в обеспечении существования организаций, а также в управлении ими.

## **1.2. Нейронные сети как средство анализа процесса**

С математической точки зрения нейронная сеть представляет собой многослойную сетевую структуру, состоящую из однотипных (и сравнительно простых) процессорных элементов - нейронов. Нейроны, связанные между собой сложной топологией межсоединений, группируются в слои (как правило, два-три), среди которых выделяются входной и выходной слои. В нейронных сетях, применяемых для прогнозирования, нейроны входного слоя воспринимают информацию о параметрах ситуации, а выходной слой сигнализирует о возможной реакции на эту ситуацию. Перед постановкой на "боевое дежурство" нейронная сеть проходит специальный этап настройки - обучения.

Как правило, сети предъявляется большое количество заранее подготовленных входных данных, для каждого из которых известна требуемая реакция сети. Если сеть реагирует на очередной пример неадекватно, т.е. состояние выходного слоя значительно отличается от заданного, в таком случае, внутренняя структура сети подвергается модификациям для минимизации ошибки (в большинстве случаев это происходит за счет корректировки веса соединений).



После определенного периода обучения сеть достигает состояния, соответствующего минимальной суммарной ошибке. Для некоторых задач суммарная ошибка может составлять 2-3%, для других может доходить до 10-15%, ряд задач вообще не поддается решению на нейронных сетях.

Вкратце перечислим основные преимущества нейронных сетей:

1. Наиболее ценное свойство нейронных сетей – способность обучаться на множестве входных параметров в тех случаях, когда неизвестны закономерности и зависимости между входными и выходными данными. В таком случае (а к ним можно отнести до 80% задач финансового анализа) пасуют как традиционные математические методы, так и экспертные системы.

2. Ещё одним ценным свойством нейронных сетей является то, они способны успешно решать опираясь на неполную, искаженную, зашумленную и внутренне противоречивую входную информацию.

3. Для использования методов математического анализа потребуется профессионал-математик. Эксплуатация обученной нейронной сети по силам и старшекласснику.

4. Нейросетевые пакеты могут быть легко интегрированы с базами данных, электронной почтой и т.д., тем самым автоматизируя процесс ввода и первичной обработки данных.

5. Внутренний параллелизм, присущий нейронным сетям, позволяет практически безгранично (насколько хватит кошелька) наращивать мощность нейросистемы. Вы можете начать с простого и дешевого пакета, потом перейти на профессиональную версию, потом добавить одну-две-три платы-ускорителя, потом перейти на специализированный нейрокомпьютер с гарантией полной преемственности всего ранее созданного программного обеспечения.

В маркетинге нейронные сети применяются для прогнозирования изменения спроса на товары или услуги в зависимости от их стоимости,

функций и т.п. Для обучения сети достаточно представить реальные данные по рынку за различные периоды. Нейросеть хороша тем, что для неопытного пользователя неважно, какая существует зависимость между входными данными, главное - что нейросеть может предсказать будущее значение спроса, зная сегодняшнее его значение, средний уровень цен, общий объем рынка.

### 1.3. Классификация нейронных сетей по количеству слоев

#### 1.3.1. Однослойные нейронные сети

Простейшая сеть состоит из группы нейронов, образующих слой, как показано в правой части рисунка 1.1. Отметим, что вершины-круги слева служат лишь для распределения входных сигналов. Они не выполняют каких-либо вычислений, и поэтому не будут считаться слоем. [25]

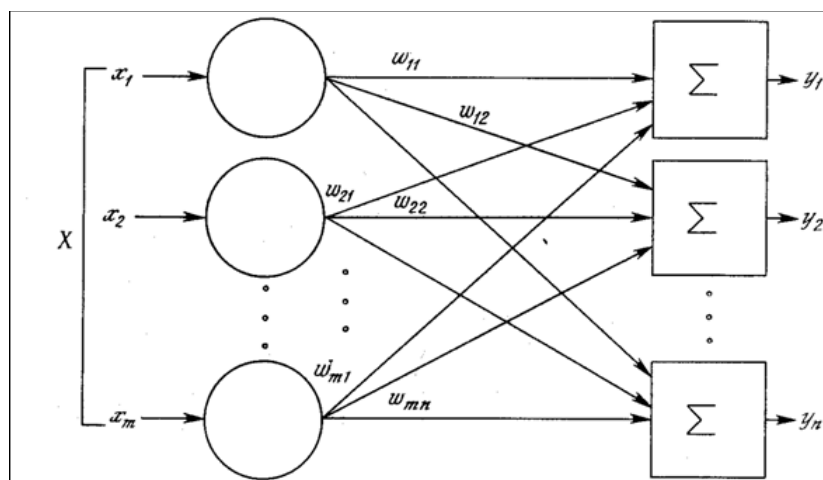


Рис. 1.1. Однослойная нейронная сеть

Удобно считать веса элементами матрицы  $W$ . Матрица имеет  $n$  строк и  $m$  столбцов, где  $m$  - число входов, а  $n$  - число нейронов. Например,  $w_{2,3}$  - это вес, связывающий третий вход со вторым нейроном. Таким образом, вычисление выходного вектора  $N$ , компонентами которого являются выходы

OUT нейронов, сводится к матричному умножению  $N=XW$ , где  $N$  и  $X$  - векторы-строки.

### 1.3.2. Многослойные нейронные сети

Многослойные сети могут образовываться каскадами слоев. Выход одного слоя является входом для последующего слоя. Подобная сеть показана на рисунке 1.2, она изображена со всеми соединениями.

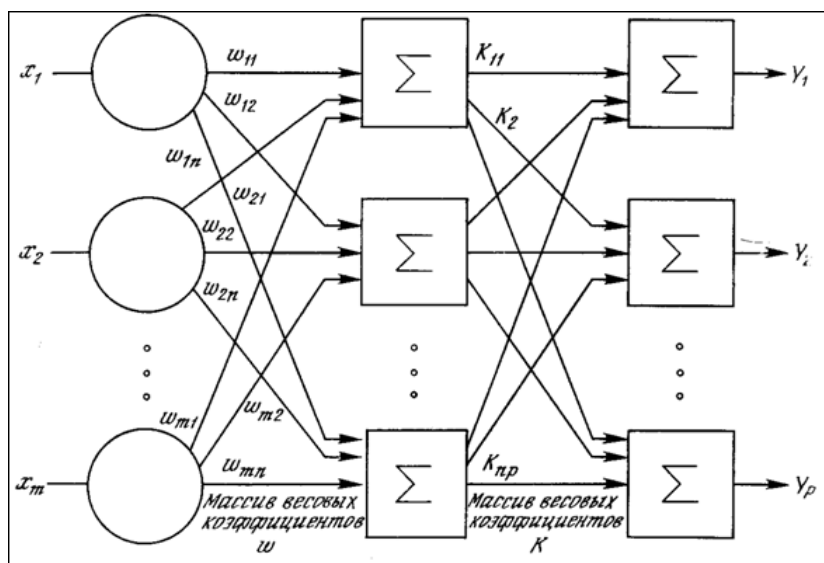


Рис. 1.2. Двухслойная нейронная сеть

Многослойные сети приводят к увеличению вычислительной мощности по сравнению с однослойной сетью лишь в том случае, если активационная функция между слоями будет нелинейной.

### 1.4. Обучение искусственной нейронной сети

Целью обучения нейросети является найти для некоторого множества входов желаемое (или, по крайней мере, сообразное с ним) множество выходов. Каждое такое множество входное (или выходное) рассматривается как отдельный вектор. Обучение происходит путем последовательного

предъявления входных векторов и одновременной подстройкой весов в соответствии с определенной процедурой. В процессе обучения веса сети постепенно становятся такими, чтобы из каждого входного вектора вырабатывался выходной.

Существуют следующие стратегии обучения: "обучение с учителем" и "обучение без учителя".

"Обучение с учителем" предполагает, что для каждого входного вектора существует целевой вектор, представляющий собой требуемый выход. Вместе эти два вектора являются обучающей парой. Векторы обучающего множества предъявляются последовательно, затем вычисляется ошибка и веса постепенно подстраиваются для каждого вектора до тех пор, пока значение ошибки по всему обучающему массиву не достигнет приемлемо низкого уровня.

Обучение без учителя является намного более правдоподобной моделью обучения в биологической системе. Развита Кохоненом [16] и многими другими [18], она не нуждается в целевом векторе для выходов и, следовательно, не требует сравнения с predetermined идеальными ответами. Обучающее множество в этом случае состоит лишь из входных векторов. Обучающий алгоритм подстраивает веса нейросети так, чтобы получались согласованные выходные векторы, т.е. чтобы предъявление достаточно близких входных векторов давало одинаковые выходы.

Ошибка обучения для сконструированной нейронной сети вычисляется путем сравнения выходных и целевых (желаемых) значений. Из полученных разностей формируется функция ошибок.

Функция ошибок - это целевая функция, которая требует минимизации в процессе управляемого обучения нейронной сети. [25]

## 1.5. Методы нейросетевого прогнозирования

### 1.5.1. Нейросети обратного распространения

Обратное распространение - это систематический метод для обучения многослойных искусственных нейронных сетей. Он имеет солидное математическое обоснование.

На рисунке 1.3 показан нейрон, который используется в качестве основного строительного блока в сетях обратного распространения. На вход сети подается множество входных данных извне, либо идущий от предшествующего слоя. Каждый из входов перемножается с весом, а их произведения суммируются. Эта сумма, обозначаемая NET, вычисляется для каждого нейрона сети отдельно. После того, как величина NET вычислена, она модифицируется с помощью активационной функции и получается сигнал OUT. [25]

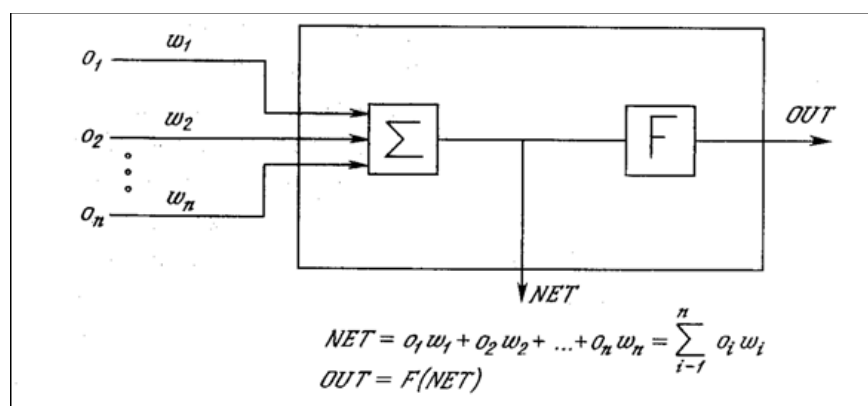


Рис. 1.3. Искусственный нейрон с активационной функцией

На рисунке 1.4. показана активационная функция, которая наиболее часто используется для сетей обратного распространения. Сигмоидальная функция или по-другому сигмоид весьма удобна, так как имеет простую производную, что используется при реализации алгоритма.

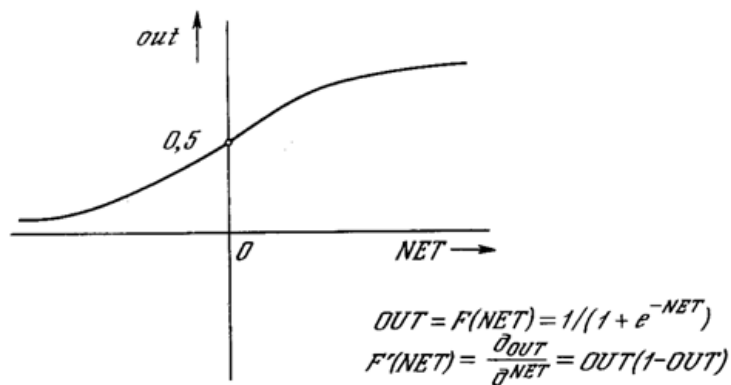


Рис. 1.4. Сигмоидальная активационная функция

Многослойная сеть, изображенная на рисунке 1.5, способна обучаться с помощью метода обратного распространения. (Для ясности рисунок упрощен). Первый (входной) слой нейронов служит лишь в качестве распределительных точек, входы здесь не суммируются. Входной сигнал просто проходит через них к весам на их выходах. А каждый нейрон последующих слоев, в свою очередь, выдает сигналы NET и OUT, как описано выше.

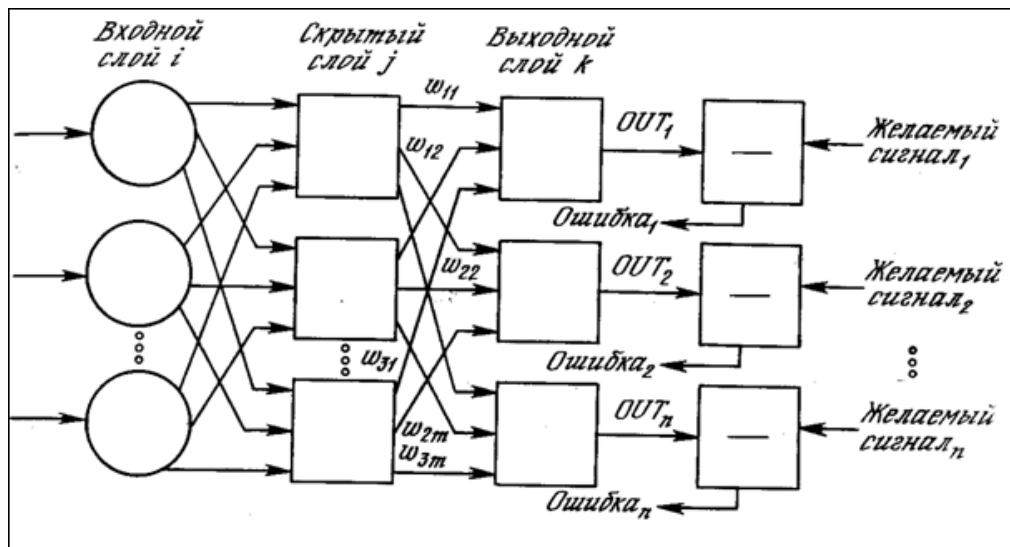


Рис. 1.5. Двухслойная сеть обратного распространения

Целью обучения сети является такая подстройка ее весов, чтобы подача на нейросеть некоторого множества входов приводила к требуемому множеству выходов. Для краткости эти множества входов и выходов будем

называть векторами. При обучении предполагается, что для каждого входного вектора существует парный ему целевой вектор, задающий требуемый выход. Вместе они называются обучающей парой. Как правило, нейросеть обучается на многих парах.

Обучение сети обратного распространения включает в себя следующие этапы:

1. Выбор очередной обучающей пары из обучающего множества; подача входного вектора на вход сети.
2. Вычисление выхода сети.
3. Вычисление разности между выходом сети и требуемым выходом (целевым вектором обучающей пары).
4. Корректировка веса сети для минимизации ошибки.
5. Повторение шагов с 1 по 4 для каждого вектора обучающего множества до тех пор, пока значение ошибки на всем множестве не достигнет приемлемого уровня.

На рисунке 1.6 показан процесс обучения для одного веса от нейрона  $p$  в скрытом слое  $j$  к нейрону  $q$  в выходном слое  $k$ . Выход нейрона слоя  $k$ , вычитаясь из целевого значения (Target), дает сигнал ошибки. Он умножается на производную сжимающей функции OUT (1-OUT), вычисленную для этого нейрона слоя  $k$ , давая, таким образом, величину  $\delta$ .

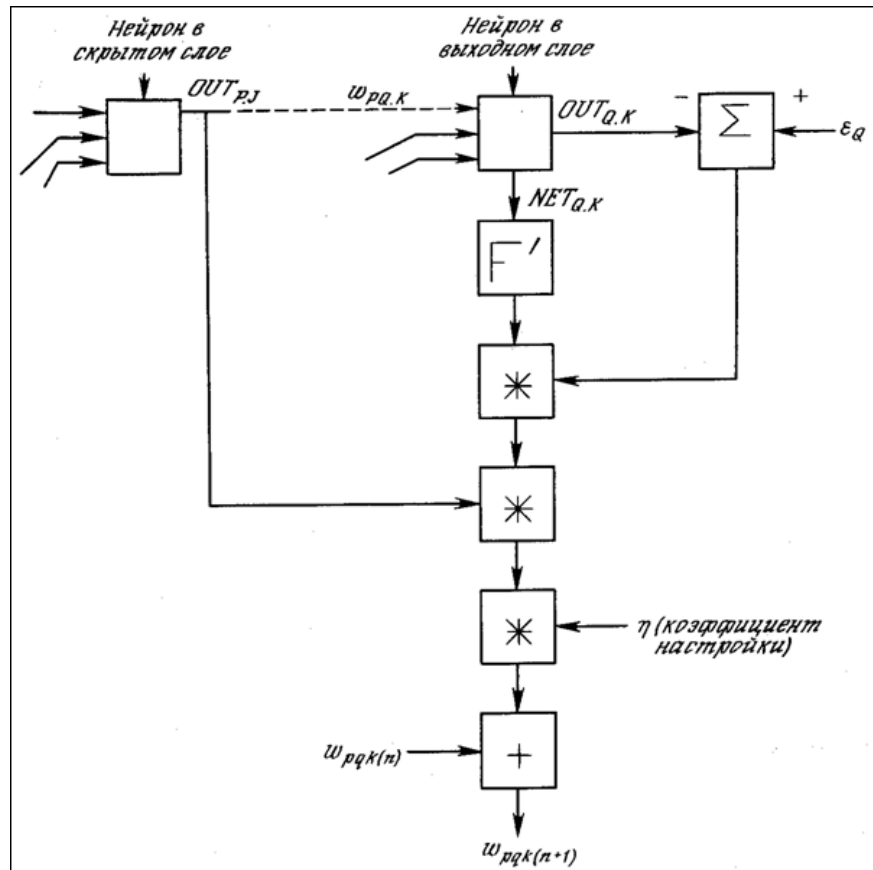


Рис. 1.6. Алгоритм настройки веса в выходном слое

Формула вычисления величины  $\delta$  выглядит следующим образом 1.1:

$$\delta = \text{OUT} (1 - \text{OUT}) (\text{Target} - \text{OUT}) \quad (1.1)$$

Затем  $\delta$  умножается на величину  $\text{OUT}$  нейрона  $j$ , из которого выходит рассматриваемый вес.

Это произведение в свою очередь умножается на коэффициент скорости обучения  $\eta$  (обычно от 0.01 до 1.0), и результат прибавляется к весу.

Такая же процедура выполняется для каждого веса от нейрона скрытого слоя к нейрону в выходном слое.



Следующие уравнения (1.2 и 1.3) иллюстрируют это вычисление:

$$\Delta w_{pq,k} = \eta \delta_{q,k} \text{OUT}_p \quad (1.2)$$

$$w_{pq,k}(n+1) = w_{pq,k}(n) + \Delta w_{pq,k} \quad (1.3)$$

где:

- $w_{pq,k}(n)$  - величина веса от нейрона  $p$  в скрытом слое к нейрону  $q$  в выходном слое на шаге  $n$  (до коррекции), так что индекс  $k$  относится к слою, в котором заканчивается данный вес, т.е. с которым он объединен;
- $w_{pq,k}(n+1)$  - величина веса на шаге  $n+1$  (после коррекции);
- $\delta_{q,k}$  - величина  $\delta$  для нейрона  $q$ , в выходном слое  $k$ ;
- $\text{OUT}_{p,j}$  - величина  $\text{OUT}$  для нейрона  $p$  в скрытом слое  $j$ .

Для каждого нейрона в данном скрытом слое должно быть вычислено  $\delta$  и скорректированы все веса, ассоциированные с этим слоем. Этот процесс повторяется поочередно, слой за слоем в направлении входа, пока все веса не будут подкорректированы (рисунок 1.7).

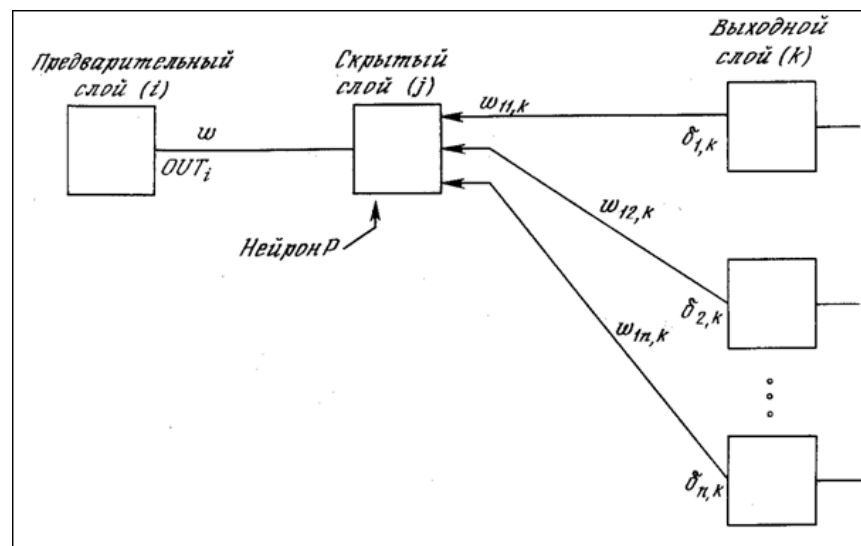


Рис. 1.7. Настройка веса в скрытом слое

Достоинства метода обратного распространения:

- достаточно высокая эффективность. Прямой ход, обратный ход и вычисления градиента требуют порядка  $O(Nn + NM)$  операций;
- через каждый нейрон проходит информация только о связанных с ним нейронах. Это дает возможность легко реализовывать этот метод на вычислительных устройствах с параллельной архитектурой.
- высокая степень общности. Алгоритм легко записать для произвольного числа слоёв, произвольного количества входов и выходов, произвольной функции потерь и произвольных функций активации, возможно, различных у разных нейронов.

К недостаткам этого метода можно отнести то, что метод не всегда имеет хорошую сходимость, а так же то, что при чрезмерно большом числе весов сеть склонна к переобучению.

С первым недостатком приходится бороться путем применения большого количества различных эвристических ухищрений. [25]

### **1.5.2. Сеть Кохонена**

Сети (слои) Кохонена (Kohonen T.) относятся к самоорганизующимся нейронным сетям. Самоорганизующаяся сеть позволяет выявлять кластеры (группы) входных векторов, обладающих некоторыми общими свойствами. Кластеризация - это разделение исследуемого множества объектов на группы "похожих" объектов, называемых кластерами. Кластеризация позволяет сгруппировать сходные данные, что облегчает решение ряда задач Data Mining. [29] В том числе и задачи прогнозирования. Относя новый объект к одному из кластеров, можно прогнозировать поведение объекта, поскольку его поведение будет схожим с поведением объектов кластера.

Важно отметить роль содержательной интерпретации каждого кластера. Каждому кластеру необходимо присвоить содержательное название, отражающее суть объектов кластера. Для этого необходимо выявить, признаки, объединяющие объекты в кластер. Это может потребовать статистического анализа свойств объекта кластера. С помощью сетей Кохонена производится кластеризация объектов, описываемых количественными характеристиками.

Сеть (слой) Кохонена (рисунок 1.8) - это однослойная сеть, построенная из нейронов типа WTA (Winner Takes All - победитель получает все).

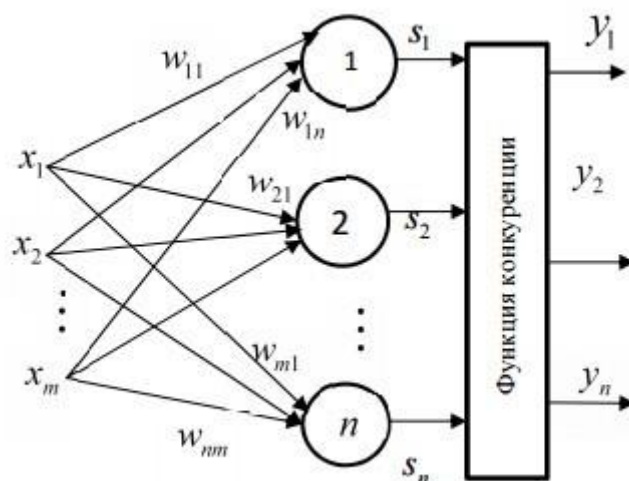


Рис. 1.8. Структура сети Кохонена

Каждый нейрон сети соединен со всеми компонентами  $m$ -мерного входного вектора  $x_i = (x_{i1}, x_{i2}, \dots, x_{im})$ . Входной вектор - это описание одного из объектов, подлежащих кластеризации. Количество нейронов совпадает с количеством кластеров, которое должна выделить сеть. В качестве нейронов сети Кохонена применяются линейные взвешенные сумматоры (формула 1.4):

$$s_j = b_j + \sum_{i=1}^m w_{ij} x_i, \quad (1.4)$$

где  $j$  - номер нейрона,  $i$  - номер входа,  $s_j$  — выход адаптивного сумматора,  $w_{ij}$  - вес  $i$ -го входа  $j$ -го нейрона,  $b_j$  — порог.

Каждый  $j$ -ый нейрон описывается вектором весов  $w_j = (w_{1j}, w_{2j}, \dots, w_{mj})$ , где  $m$  – число компонентов входных векторов.

С выходов адаптивных сумматоров сигнал поступает на функцию конкуренции, работающую по правилу "победитель получает всё". Функция конкуренции находит выход адаптивный сумматор с максимальным значением выхода.

В сетях Кохонена используется обучение без учителя. Для обучения сети применяются механизмы конкуренции. При подаче на вход сети вектора  $x$  побеждает тот нейрон, вектор весов которого в наименьшей степени отличаются от входного вектора. Для нейрона-победителя выполняется соотношение 1.5:

$$d(x, w_j) = \min_{1 \leq i \leq n} d(x, w_i), \quad (1.5)$$

где  $n$  - количество нейронов,  $j$  - номер нейрона-победителя,  $d(x, w)$  - расстояние (в смысле выбранной метрики) между векторами  $x$  и  $w$ . Чаще всего в качестве меры расстояния используется евклидова мера.

Перед процессом обучения производится инициализация сети, то есть первоначальное задание векторов весов. В простейшем случае задаются случайные значения весов.

Процесс обучения сети Кохонена состоит из циклического повторения ряда шагов: [27]

1. Подача исходных данных на входы. Обычно это случайная выборка одного из входных векторов.
2. Нахождение выхода каждого нейрона.
3. Определение "выигравшего" нейрона (веса которого в наименьшей степени отличаются от соответствующих компонентов входного вектора), или нейрона-победителя.
4. Корректировка весов "выигравшего" нейрона по правилу Кохонена по формуле 1.6.

$$w_i^{(k+1)} = w_i^{(k)} + \eta_i^{(k)} [x - w_i^{(k)}] \quad (1.6)$$

где  $x$  - входной вектор,  $k$  - номер цикла обучения,  $\eta_i^{(k)}$  - коэффициент скорости обучения  $i$ -го нейрона в  $k$ -ом цикле обучения.

5. Переход на шаг 1, если обучение не завершено. Часто это условие задается числом циклов обучения. Можно так же проверять достижение малой величины функционала ошибки.

Недостатки карт Кохонена: [25]

- Субъективность. Не всегда понятно, какие именно особенности карты обусловлены кластерной структурой данных, а какие - свойствами потенциальной функции. Сглаженность границ кластеров и степень детализации карты существенно зависят от выбора параметра  $\beta$ .
- Наличие искажений. Близкие объекты исходного пространства могут переходить в далёкие точки на карте. В частности, объективно существующие кластеры могут разрываться на фрагменты [9]. И наоборот, далёкие объекты могут случайно оказаться на карте рядом, особенно, если они были одинаково далеки от всех кластеров.
- Зависимость от инициализации. На процесс обучения существенно

влияет начальное распределение весов. Это сказывается не только на расположении кластеров, но даже на их количестве.

Не секрет, что популярность карт Кохонена обусловлена в значительной степени их эстетической привлекательностью и общим впечатлением наукообразия и загадочности, которое они производят на неискущённых пользователей. На практике они применяются в основном как вспомогательное средство для предварительного визуального анализа и понимания общей структуры многомерных данных. [25]

### 1.5.3. Гибридные нейронные сети

Гибридная нейронная сеть - это нейронная сеть с четкими сигналами, весами и активационной функцией, но с объединением  $x_i$  и  $w_i$ ,  $p_1$  и  $p_2$  с использованием непрерывных операций. [27]

Входы, выходы и веса гибридной нейронной сети - вещественные числа, принадлежащие отрезку  $[0,1]$ .

Рассмотрим следующие примеры элементарных гибридных нейронных сетей, а именно: нечеткий нейрон «И» и нечеткий нейрон «ИЛИ».

Сигналы  $x_i$  и веса  $w_i$  в случае нечеткого нейрона «И» объединяются с помощью треугольной конормы  $p_i = S(w_i, x_i)$ ,  $i = 1,2$ , а выход образуется с применением треугольной нормы по формуле 1.7:

$$y = \text{AND}(p_1, p_2) = T(p_1, p_2) = T(S(w_1, x_1), S(w_2, x_2)) \quad (1.7)$$

На рисунке 1.9 представлена структура гибридного нейрона «И».

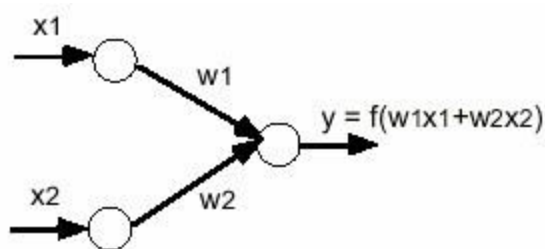


Рис. 1.9. Структура гибридного нейрона "И"

Если принять  $T = \min$ ,  $S = \max$ , тогда нечеткий нейрон "И" реализует композицию min-max:  $y = \min \{w_1 \vee x_1, w_2 \vee x_2\}$ .

В случае с нечетким нейроном «ИЛИ» сигналы  $x_i$  и веса  $w_i$  объединяются с помощью треугольной нормы  $p_i = T(w_i, x_i)$ , при  $i = 1, 2, \dots$ , а выход образуется с применением треугольной конормы по формуле 1.8:

$$y = OR(p_1, p_2) = S(p_1, p_2) = S(T(w_1, x_1), T(w_2, x_2)) \quad (1.8)$$

На рисунке 1.10 представлена структура гибридного нейрона «ИЛИ».

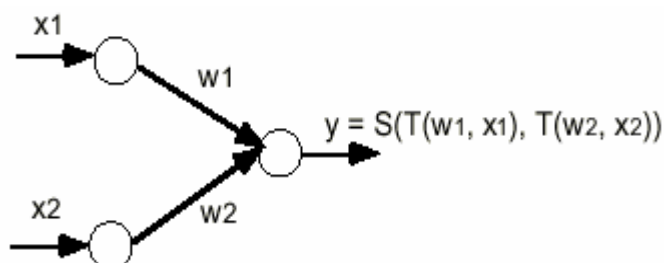


Рис. 1.10. Нечеткий нейрон "ИЛИ"

Если принять  $T = \min$ ,  $S = \max$ , тогда нечеткий нейрон "ИЛИ" реализует композицию max-min:  $y = \max \{w_1 \wedge x_1, w_2 \wedge x_2\}$ .

Алгоритм Сугено является наиболее часто используемым при построении адаптивных систем нейро-нечеткого вывода. [17]

На основе алгоритма нечеткого логического вывода строится система рассуждений (рисунок 1.11)



Рис. 1.11. Система нечетких рассуждений

Система нечетких рассуждений состоит из пяти функциональных блоков: [17]

- блок фаззификации, преобразующий численные входные значения в степени соответствия лингвистическим переменным;
- база правил, содержащая набор нечетких правил типа если-то;
- база данных, в которой определены функции принадлежности нечетких множеств используемых в нечетких правилах;
- блок принятия решений, совершающий операции вывода на основании имеющихся правил;
- блок дефаззификации, преобразующий результаты вывода в численные значения.

Традиционно база правил и база данных объединяются в общий блок - база знаний. Рассмотрим класс адаптивных сетей функционально эквивалентных системам нечетких рассуждений. Подобная архитектура носит название ANFIS (это аббревиатура Adaptive-Network-Based Fuzzy Inference System - адаптивная сеть нечеткого вывода). Она была предложена Янгом (Jang) в начале девяностых. [34] ANFIS является одним из первых вариантов гибридных нейро-нечетких сетей - нейронной сети прямого распространения сигнала особого типа. Архитектура нейро-нечеткой сети изоморфна нечеткой базе знаний. В нейро-нечетких сетях используются дифференцируемые реализации треугольных норм (умножение и



вероятностное ИЛИ), а также гладкие функции принадлежности. Это позволяет применять для настройки нейро-нечетких сетей быстрые алгоритмы обучения нейронных сетей, основанные на методе обратного распространения ошибки.

Ниже описываются архитектура и правила функционирования каждого слоя ANFIS-сети. ANFIS реализует систему нечеткого вывода Сугено в виде пятислойной нейронной сети прямого распространения сигнала. Назначение слоев следующее: [34]

- первый слой-термы входных переменных;
- второй слой - antecedentes (посылки) нечетких правил;
- третий слой - нормализация степеней выполнения правил;
- четвертый слой - заключения правил;
- пятый слой - агрегирование результата, полученного по различным правилам.

Алгоритм обучения адаптивной системы нейро-нечеткого вывода можно разбить на следующие шаги [25]:

1. Определение для выборки правил, которые влияют на результат, это те правила  $\hat{\tau}_i$ , которых не равно нулю.
2. Если есть нерассмотренные правила, влияющие на результат считаем очередное правило и осуществляем переход на шаг 3, иначе на шаг 5.
3. Рассчитываем приращение для параметров функций принадлежности, определенных в заключение рассматриваемого правила.
4. Изменяется значение параметров функций принадлежности на рассчитанную величину и переходим на шаг 2.
5. Рассчитывается выходное значение сети после изменения параметров выходного слоя.
6. Определяется остаточная ошибка.

7. Если есть нерассмотренные правила, влияющие на результат считываем очередное правило и переходим на шаг 8, иначе на шаг 12.
8. По предпосылкам правил определяем какие узлы первого слоя влияют на результат.
9. Рассчитываем производную функции принадлежности для определенных на предыдущем шаге узлов.
10. Рассчитываем приращение для параметров функций принадлежностей.
11. Изменяем значение параметров функций принадлежности на рассчитанную величину и переходим на шаг 7.
12. Завершаем обучение и совершаем выход из процедуры.

Третий и четвертый шаги алгоритма реализуют обучение узлов выходного слоя, пятый и шестой шаги вычисляют остаточную ошибку после обучения выходного слоя, и шаги реализуют обучение узлов первого слоя методом градиентного спуска.

## **2. НЕЙРОСЕТЕВОЕ ПРОГНОЗИРОВАНИЕ СПРОСА СТРОИТЕЛЬНЫХ ТОВАРОВ**

### **2.1. Выбор программного средства для решения задачи**

Поскольку метод нейронных сетей является трудоемким в реализации, а мы преследуем цель выбора оптимальной методики, то для его применения следует выбрать готовые программные решения, которых в настоящее время существует достаточно много. К ним можно отнести, например, Microsoft Analysis Services, Microsoft BI Solution Builder, Oracle Data Mining Suite, Oracle Crystal Ball, SPSS, MATLAB.

#### **2.1.1. Службы Microsoft Analysis Services**

В службах SQL Server Службы Analysis Services [42] алгоритм нейронной сети (Майкрософт) сочетает каждое возможное состояние входного атрибута с каждым возможным состоянием прогнозируемого атрибута и использует обучающие данные для вычисления вероятностей. Далее эти вероятности можно использовать для классификации или регрессии, а также для прогнозирования исхода прогнозируемого атрибута на основе входных атрибутов.

В модели интеллектуального анализа данных, которая создается при помощи алгоритма нейронной сети Microsoft, может содержаться несколько сетей, что определяется количеством столбцов, используемых для входа и прогнозирования или только для прогнозирования. Количество сетей, содержащихся в одной модели интеллектуального анализа данных, зависит от количества состояний, содержащихся во входных и прогнозируемых столбцах, используемых моделью интеллектуального анализа данных.

## 2.1.2. Платформа Microsoft BI Solution builder

Платформа Microsoft Power BI или просто Power BI – это новое направление развития Бизнес – аналитики (BI), представляющее собой облачные технологии для анализа и визуализации данных. Основное преимущество данной технологии – это возможность построения красивых информационных панелей (dashboard), как правило, с ключевыми показателями деятельности компании, доступных на любом устройстве.

В BI-решениях от Microsoft сочетаются простота внедрения, отличная масштабируемость и возможность использования знакомой пользователям среды Microsoft Office.

Power BI содежит в себе два компонента: SQL Server 2008 и PerfomancePoint Server 2007.

Продукт SQL Server 2008 помогает сделать больше ценных выводов на основе изучения бизнес-данных и ускорить достижение целевых результатов, а значит, и завоевать конкурентное преимущество. Кроме того, продукт SQL Server 2008 предоставляет следующие возможности:

- службы интеграции SQL Server 2008 позволяют оперативно объединять многочисленные источники данных;
- службы аналитики SQL Server 2008 обогащают содержание данных и позволяют проводить сложный бизнес-анализ;
- службы отчетов SQL Server 2008 предназначены для составления, доставки и управления комплексными отчетами.

Сервер Microsoft Office PerformancePoint Server 2007 разрабатывался специально для повышения эксплуатационной и финансовой производительности во всех отделах и на всех уровнях организации. Являясь интегрированным приложением, он призван поддерживать формальную бизнес-аналитику (BI) и процессы управления производительностью.

С помощью PerformancePoint Server 2007 можно:

- отслеживать ход работы благодаря системе показателей или панели мониторинга;
- быстро находить причины отклонений от запланированных или ожидаемых показателей;
- вносить изменения в корпоративные планы или бюджет в связи с изменившимися условиями.

### **2.1.3. Oracle Data Mining Suite**

Корпорация Oracle предлагает своим пользователям набор продуктов класса и сервисов класса Data Mining (Добыча Данных или ИАД (Интеллектуальный Анализ Данных)). [43] Этот набор помогает компаниям понять и предвидеть поведение клиентов (покупателей, заказчиков) и создавать полные интегрированные решения по управлению отношениями с клиентами (CRM).

В настоящее время используется комплект Oracle Data Mining Suite (ранее известный как "Oracle Darwin") для построения моделей предсказания (predictive models), обнаруживающих скрытые в ваших данных закономерности и новую ценную информацию, которая может быть использована:

- в системах раннего обнаружения экстремальных ситуаций;
- для более полного понимания поведения и запросов клиентов и рыночной ситуации;
- для определения эффективности различных видов продаж;
- как помощь в борьбе со злоумышленниками.

Информация, извлеченная из данных таким образом, обеспечивает бизнес-анализ, результатом которого является улучшение отношений с

клиентами и, тем самым, увеличение доходов и экономия в основном бизнесе.

#### **2.1.4. Статистический пакет Statistical Package for the Social Sciences (SPSS)**

Пакет SPSS (Statistical Package for the Social Sciences) – универсальный статистический пакет компании SPSS Inc.[44] По мнению разработчиков пакета, SPSS является одним из лидирующих программных продуктов в области статистического анализа данных для решения вопросов в правительственной, академической и бизнессфере. SPSS является модульной программой. Ее основу составляет базовый модуль (SPSS Base), позволяющий осуществлять управление данными и содержащий наиболее распространенные методы статистического анализа данных: проведение описательной статистики; построение линейных и нелинейных моделей; осуществление преобразования данных; проведение факторного, кластерного, дисперсионного анализов; вычисление корреляций; построение графиков; подготовка отчетов и пр.

##### Достоинства SPSS:

- развитый аппарат статистического анализа;
- универсальность (т.е. пакет может быть использован для решения широкого круга вопросов из различных предметных областей, требующих проведения статистического анализа данных);
- широкий набор статистических и графических процедур (более 50 типов диаграмм) анализа данных, а также процедур создания отчетов;
- детальная контекстно-ориентированная справочная система, которая позволяет неопытному пользователю с большей легкостью ориентироваться в программе;
- совместимость с операционными системами Windows, Mac, Linux.

##### Недостатки SPSS:

- высокие требования к системе компьютера (требуется 1GB оперативной памяти, 800MB памяти на жестком диске и процессор с частотой 1GHz и выше);
- высокая цена по сравнению со статистическими пакетами аналогичного уровня.

### 2.1.5. Oracle Crystal Ball

Oracle Crystal Ball – приложение к Microsoft Excel для моделирования бизнес-процессов, определения рисков, прогнозирования неопределенных переменных и оптимизации полученных результатов. [37] Crystal Ball помогает принимать верные тактические решения для достижения стратегических целей компании и получать конкурентные преимущества в условиях большого числа постоянно меняющихся рыночных показателей.

Основные функции и преимущества:

- Автоматизация «what-if» анализа. Возможность вычисления рисков: Какова вероятность достижения целей компании? Как рассчитать существенные факторы риска? Ответы на эти и на другие вопросы, задаваемые в рамках методологии «что-если» могут быть получены путем присвоения возможных значений неизвестным переменным. Crystal Ball автоматизирует трудоемкие процессы, основанные на сценариях « what-if» с использованием методов моделирования Монте-Карло, присваивая разнообразные значения или распределяя вероятности для каждой неизвестной переменной. Приложение многократно генерирует различные значения в рамках указанного допустимого ряда, после чего выполняет повторный расчет модели, сохраняя результаты каждого сценария «what-if».
- Расширенные функциональные возможности: для опытных

пользователей в состав Crystal Ball включены инструменты для вычисления корреляций, анализа чувствительности, аппроксимации распределения и точные средства управления, обеспечивающие более качественный анализ.

- Легко интерпретируемые графики и отчеты: Crystal Ball позволяет быстро создавать графики и отчеты, которые иллюстрируют различные допущения и итоговые прогнозы, существенно облегчая принятие решения.

- Простота использования: приложение Crystal Ball просто в изучении и не требует расширенных статистических или компьютерных знаний.

- Достаточно навыков работы с Microsoft Excel. Большинство функциональных возможностей Crystal Ball реализуется несколькими щелчками мыши.

- Интеграция с EPM приложениям: используя возможности Smart View, пользователь получает доступ к данным различных EPM-продуктов, таких как Essbase, Hyperion Planning, Strategic Finance и может использовать их для дальнейшего анализа в Microsoft Excel. Впоследствии эти данные могут быть применены как непосредственно для прогнозирования в Crystal Ball, так и связаны с существующими моделями. При необходимости, результаты проведенного в системе анализа могут быть отправлены обратно в приложения EPM.

### **2.1.6. Пакет MATLAB**

Пакет MATLAB (The MathWorks) также предоставляет пользователям возможность работы с нейронными сетями. [41] Входящий в стандартную поставку MATLAB "Neural Network Toolbox" [45] предоставляет широкие возможности для работы с нейронными сетями всех типов.

Пакет прикладных программ Neural Network ToolBox содержит средства для конструирования нейронных сетей, базирующихся на поведении математического аналога нейрона. Пакет обеспечивает



эффективную поддержку проектирования, обучения, анализа и моделирования множества известных типов сетей – от базовых моделей перцептрона до самых современных ассоциативных и самоорганизующихся сетей. Данная среда предоставляет возможность создания нейронных сетей как уже по заранее существующим шаблонам. При этом пользователям будет необходимо определить количество входов и выходов сети, весовые коэффициенты, связывающие нейроны, функции активации и еще много различных параметров.

Также система MATLAB Neural Network Toolbox предоставляет максимальные возможности настройки и контроля сети в процессе обучения, именно она оптимально подходит в качестве системы прогнозирования данных.

Преимуществом пакета MATLAB является то, что при его использовании пользователь не ограничен моделями нейронных сетей и их параметрами, которые жестко заложены в нейросимуляторе. Пакет дает возможность самостоятельно сконструировать ту сеть, которую пользователь считает оптимальной для решения поставленной задачи. В пакете возможно конструирование сети любой сложности и нет необходимости привязываться к ограничениям, накладываемым нейросимулятором.

Таким образом, MATLAB является достаточно гибким и наиболее подходящим инструментом, позволяющим заменить различные статистические программные решения, и по-новому взглянуть на проблему прогнозирования финансового состояния предприятия.

## **2.2. Обоснование выбора программного средства для исследования**

Выбор программного продукта для использования метода нейронных сетей был выполнен по следующим критериям: возможность обработать данные онлайн, простота использования, скорость работы, наличие документации. В таблице 2.1 представлен сравнительный анализ

программных средств по выбранным критериям. Знаком «+» отмечены программные средства, обладающие соответствующим критерием, в противном случае – знаком «-».

Таблица 2.1

Сравнительный анализ программных средств

Программные решения	Обработка онлайн данных	Простота использования	Скорость работы	Документированность
SAS Business Intelligence	-	-	+	+
Microsoft BI Solution builder +	-	+	+	-
Oracle Data Mining Suite +	-	-	+	+
SPSS	-	+	+	-
Microsoft SQL Server Analysis Services	+	+	-	+
Oracle Crystal Ball	-	+	-	+
MATLAB	-	+	+	+

В процессе анализа существующих решений было выявлено, что среда моделирования MATLAB является наиболее подходящим средством ввиду ее простоты в использовании, высокой скорости работы, а также наличия большого количества методических материалов по ее использованию, что очень ценно для потенциальных пользователей.

### 2.3. Выбор архитектуры и метода обучения нейронной сети в системе Matlab

#### 2.3.1. Модель обучения нейронной сети

Нейронная сеть, реализованная в среде моделирования MATLAB для решения задачи прогнозирования в первом случае, представляет собой однослойную нейронную сеть, схема которой представлена на рисунке 2.1.

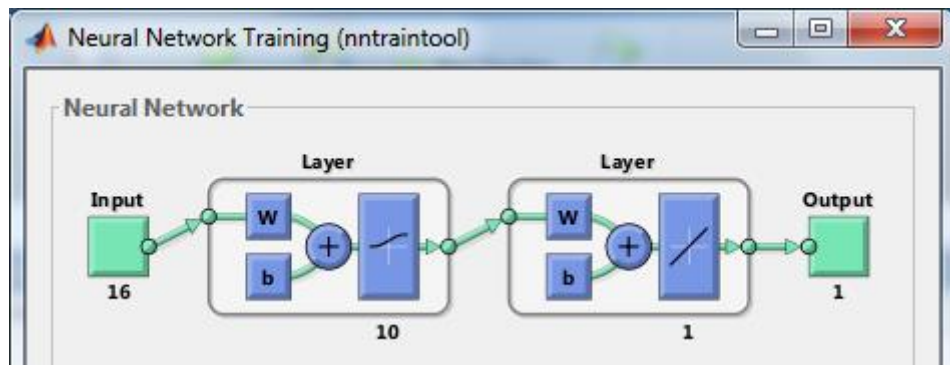


Рис. 2.1. Модель однослойной нейронной сети

Во втором случае – это нейронная сеть с тремя слоями по десять, пять, одному нейрону в каждом слое соответственно (рисунок 2.2).

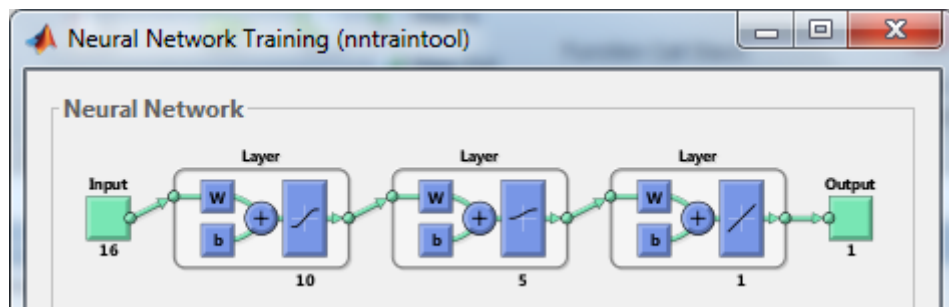


Рис. 2.2. Модель многослойной нейронной сети

Здесь  $b$  – это множество входов,  $w$  – множество весов, знак «+» – сумматор. Сеть состоит из шестнадцати входов, одного выходов и трех слоев, в каждом слое функция активации и количество нейронов различны. В первом слое количество нейронов равно десяти. Во втором слое (скрытом) находится пять нейронов. В третьем слое количество нейронов равно количеству входов. Нейроны первого слоя настроены на гиперболическую функцию активации, второго слоя – на логарифмическую, а третьего на линейную функцию активации, которая не изменяет выходной сигнал.

Таким образом, на вход нейронной сети подается сигнал  $x$ . Сумматор «+» умножает каждое входное множества  $b_i$  на веса  $w_i$  и затем происходит

суммирование взвешенных входов. Затем полученное значение пропускается через функцию активации соответствующего слоя и рассчитывается выход. Далее происходит корректировка весов с помощью метода обратного распространения ошибки и после чего итерация повторяется до тех пор, пока не будет достигнута заданная точность либо не будет исчерпано количество итераций.

На рисунке 2.3 отображена структура гибридной сети, смоделированная системой ANFIS-редактор. [1],[35]

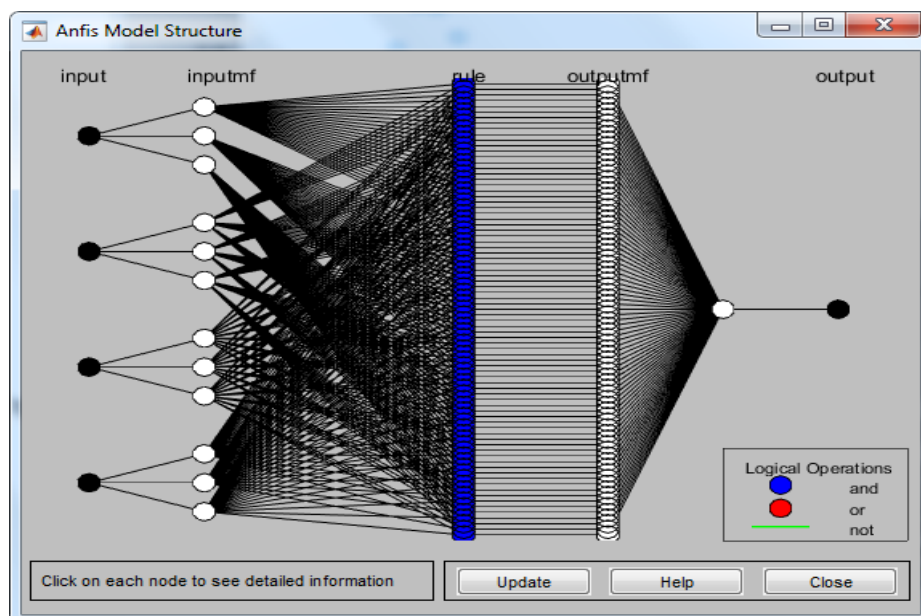


Рис. 2.3. Структура гибридной сети

### 2.3.2. Алгоритм обучения метода обратного распространения

Обучение нейронной сети выполняется методом обратного распространения ошибки. В процессе обучения происходит подстройка ее весов таким образом, чтобы некоторое множество входов приводило к требуемому множеству выходов. При этом изначально предполагается, что каждому входному множеству соответствует парное ему целевое множество, задающий требуемый выход. Вместе они являются обучающей парой.

На каждой итерации изменение веса производится методом градиентного спуска по формуле 2.1: [10],[11]

$$\omega_{ij}^{(n)}(t) = \omega_{ij}^{(n)}(t-1) + \Delta\omega_{ij}^{(n)}(t) \quad (2.1)$$

Для ограничения пространства поиска при обучении ставится задача минимизации целевой функции ошибки НС, которая находится по методу наименьших квадратов по формуле 2.2:

$$E(\omega) = \frac{1}{2} \sum_{j=1}^p (y_j - d_j)^2, \quad (2.2)$$

где:

- $y_j$  – значение  $j$ -го выхода нейронной сети;
- $d_j$  – целевое значение  $j$ -го выхода;
- $p$  – число нейронов в выходном слое.

Комбинация метода наименьших квадратов с методом градиентного спуска называется алгоритмом Левенберга-Марквардта, который и используется в среде MATLAB. [12]

Полный алгоритм обучения нейронной сети (НС) состоит из пяти шагов:

1. Подача на вход НС одного из входных сигналов и определение значения выходов нейронов нейронной сети.
2. Рассчитать для выходного слоя нейронной сети и рассчитать изменения весов выходного слоя  $n$  (2.1).
3. Рассчитать соответственно и  $\Delta\omega_{ij}^n$  для остальных слоев НС,  $n = N-1..1$  (2.1).
4. Скорректировать все веса НС по формуле 2.1.

5. Если высчитанная по формуле 2.2 ошибка велика, то перейти на шаг 1.

### 2.3.3. Гибридный алгоритм обучения (MATLAB ANFIS)

ANFIS-редактор позволяет автоматически синтезировать из экспериментальных данных нейро-нечеткие сети.

Редактор ANFIS использует гибридный двухпроходный алгоритм. В прямом ходе определяются параметры слоя 4 и используется метод наименьших квадратов. В обратном – уточняются параметры первого слоя, используется обратное распространение ошибки.

Алгоритм обучения прямой ход:

Пусть:

$S_1 = \{ \{a_{11}, b_{11}, c_{11}\}, \dots, \{a_{np}, b_{np}, c_{np}\} \}$  – параметры предпосылок,

$S_2 = \{q_{10}, q_{11}, \dots, q_{1n}, \dots, q_{pn}, q_{p0}, \dots, q_{pn}\}$  – параметры заключений,

$S = S_1 \cup S_2$ .

При фиксированных параметрах из  $S_1$  имея  $K$  обучающих примеров, можно представить задачу обучения в виде  $AX = B$ , где  $X$  содержит параметры из  $S_2$ .

Так как число обучающих примеров обычно больше числа параметров в  $S_2$ , то точное решение данной системы уравнений не возможно, но можно воспользоваться методом наименьших квадратов для нахождения аппроксимации  $X^*$ , которая минимизирует ошибку  $\|AX - B\|^2$ .

Для вычислений МНК используется итеративная процедура:

$$S_{i+1} = S_i - (S_i a_{i+1} a_{i+1}^T S_i) / (1 + a_{i+1}^T S_i a_{i+1}), \quad (2.3)$$

$$X_{i+1} = X_i + S_{i+1} a_{i+1} (b^T a_{i+1} - a_{i+1}^T X_i), \quad (2.4)$$

где:

- $X_0 = 0$ ;  $S_0 = \gamma I$ ,  $\gamma$ - большое число,  $I$  – единичная матрица;
- $a_i^T$  -  $i$ -я строка матрицы  $A$ ;
- $b_i^T$  -  $i$ -ый элемент вектора  $B$ ;
- $X^* = X_K$ .

Алгоритм обучения - обратный ход:

Для каждого параметра  $\alpha_i$  из  $S_1$  обновление осуществляется по правилу:

$$\Delta\alpha_i = -\eta (\partial E / \partial \alpha_i), \quad (2.5)$$

где:

- $E$  – ошибка на обучающей выборке;
- $\eta$  – скорость обучения;
- $k$  – размер шага.

## **2.4. Моделирование нейронной сети для прогнозирования потребительского спроса строительных товаров**

### **2.4.1. Настройка сети обратного распространения**

Настройка нейронной сети в среде MATLAB осуществлялась следующими командами: [1],[12]

- создание сети, задание количества слоев и количества нейронов в слое, выбор функции активации:

```
prognoz=newff(minmax(Data_vh'),[10 1],{'logsig' 'purelin'});
```

- ограничение итераций обучения, определяемое количеством эпох  

```
prognoz.trainParam.epochs = 1000;
```
- запуск этапа тренировки, где prognoz – тип сети обучения, Data\_vh

входные сигналы, Data\_vih – обучающая выборка

```
prognoz = train(prognoz, Data_vh',Data_vih');
```

- выбор метода нахождения ошибки (корректировки весов)

суммарное квадратичное отклонение

```
prognoz.performFcn='sse';
```

- расчет значений выборки (обучающая выборка) для

проверки точности модели

```
P1 = sim(prognoz,vxod');
```

- прогнозирование для входных сигналов P2

```
P2=sim(prognoz,test_vhod');
```

Настройка многослойной нейронной сети будет отличаться заданием большего числа слоев – в данном случае три, числа нейронов и выбора для каждого слоя функции активации:

```
prognoz=newff(minmax(vxod'),[10 3 1],{ 'tansig','logsig', 'purelin'}).
```

### **2.4.3. Настройка гибридной нейронной сети**

Настройка гибридной сети будет осуществляться посредством ANFIS-редактора, встроенного в Matlab. [1],[40]

Загрузка ANFIS-редактора осуществляется по команде `anfisedit`. ANFIS-редактор включает в себя 3 верхних меню - File, Edit и View, а так же следующие рабочие области: визуализации, свойств ANFIS, загрузки данных, генерация исходной системы нечеткого логического вывода, обучения, тестирования, вывода текущей информации, а также кнопки Help и Close, которые позволяют вызвать окно справки и закрыть ANFIS-редактор, соответственно.

На рисунке 2.4. представлено рабочее окно ANFIS-редактора.



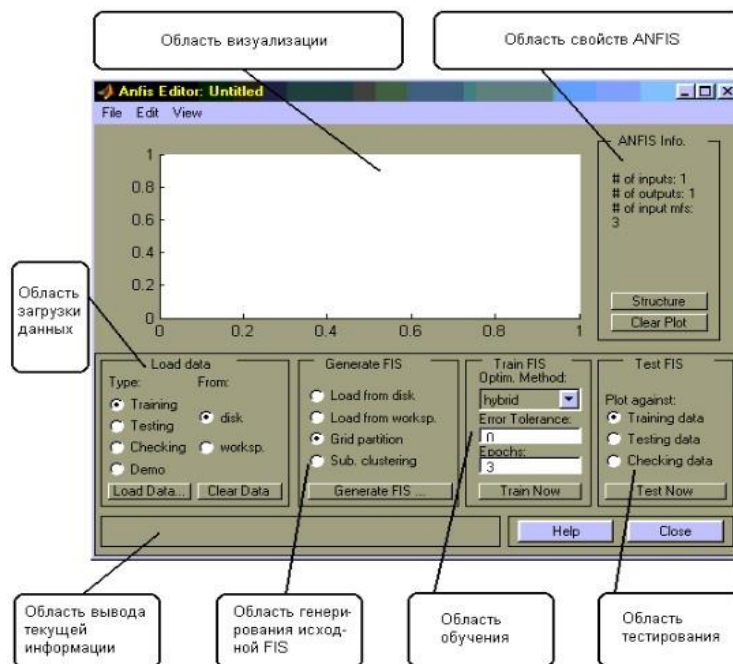


Рис. 2.4. Рабочая область ANFIS редактора

Меню File и View одинаковые для всех GUI-модулей, используемых с системами нечеткого логического вывода.

Для загрузки данных используется кнопка Load Data, после чего в Редактор можно загрузить входные и тестовые данные.

Для обучения данных в области обучения редактора ANFIS нужно задать количество эпох для обучения и выбрать метод, с помощью которого будет происходить обучение. Кроме этого, можно задать значение ошибки, как условие останова обучения.

После обучения можно проверить работу НС на тестовых данных. Для этого в области тестирования выбираем Testing data и нажимаем Test now.

### 3. СБОР И АНАЛИЗ ПОЛУЧЕННЫХ РЕЗУЛЬТАТОВ

### **3.1. Информационное обеспечение задачи. Данные о продажах**

Выявленная в результате исследований методика, может найти применение в прогнозировании работы компаний по продаже строительных товаров. Входным информационным обеспечением может служить исключительно статистика продаж строительных товаров за определенное время, на основе которой будет делаться прогноз. В задаче прогнозирования будем использовать статистические данные продаж компании ООО «Баустов».

Компания ООО «Баустов» является крупным российским дистрибьютором, а так же производителем строительных и отделочных материалов. В состав Группы Компаний «Баустов» входят организации, занимающиеся оптовыми продажами строительных материалов, производством лакокрасочной и геотекстильной продукции, а также выполнением строительных работ на территории Москвы и Московской области.

Офис компании располагается по адресу: Москва, ул. Новоостاپовская, 5. ООО «Баустов» является обладателем крупного склада в Московской области, который расположен в 10 км от Москвы. Имея склад, компании очень важно прогнозировать объемы продаж того или иного строительного товара, особенно в сезонное для него время.

Структурная схема компании ООО «Баустов» наглядно представлена на рисунке 3.1:

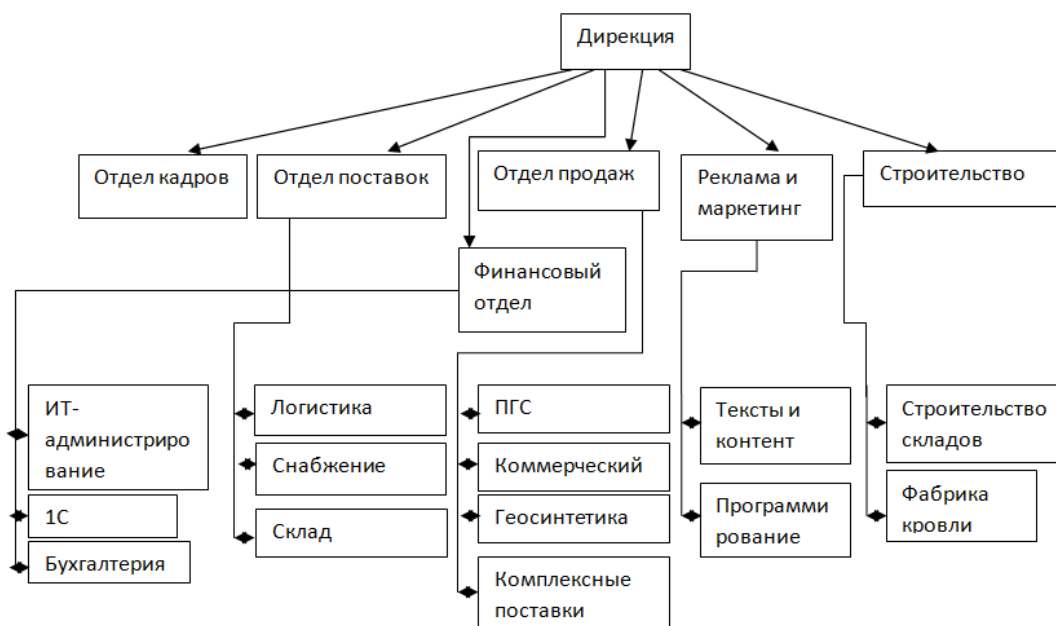


Рис. 3.1. Структурная схема ООО «Баустов»

Для практического применения методики необходимо отталкиваться от реальных данных о продаже строительных товаров. Рассматривать все строительные материалы вместе, а тем более прогнозировать все за раз невозможно ввиду совершенно разной статистики продаж того или иного товара в определенный период времени, что даст весьма усредненный результат. В рамках настоящей работы будет рассмотрен конкретный товар – базальтовый утеплитель Rockwool Лайт Баттс с определенными характеристиками, а именно: плотность 37 кг/м<sup>3</sup>, ширина 100 мм. На рисунке представлена выборка данных о продаже товара за последние пять лет с 2012 г. по 2016 г. с шагом в неделю. Данные расположены в порядке месяцев, начиная с января, заканчивая декабрем, сверху вниз. Всего 48 записей за каждый год.

	A	B	C	D	E	F	G
2			Продажи				
3			2012	2013	2014	2015	2016
4		1	0	0	0	0	0
5		2	18550	0	9542	23600	41580
6		3	69650	54900	37801	89200	48720
7	Январь	4	17500	76494	56151	74400	0
8		5	192500	252540	183500	186400	238980
9		6	173600	215940	167352	201200	278460
10		7	245000	274500	242220	356000	325920
11	Февраль	8	392000	384300	448841	436000	364980
12		9	980000	1098000	697300	800000	1429260
13		10	1122100	1282098	1027600	960000	1388520
14		11	1407000	1379820	1093660	1396000	1629600
15	Март	12	2377550	1912716	1835000	2412000	3032400
16		13	2800000	2763300	2563330	2308000	3788400
17		14	2352000	3209820	2080890	3017200	3721200
18		15	2460150	2400960	2811220	3200000	3215520
19	Апрель	17	2415000	2577738	2605700	2748000	2764860
20		18	1960000	1978596	1831330	2008000	2362920
21		19	1680000	1387140	1480111	1504000	1898400
22		21	1592500	2007876	1790960	1719600	1961400
23	Май	22	1393000	1537200	1666180	1912000	2053800
24		23	791000	1098000	1020260	1000000	1270920
25		24	1172500	1423740	1835000	1280000	1692600
26		28	1858500	1826340	1724900	1600000	2122260
27	Июнь	29	700000	693936	814740	668000	819000
28		30	779100	579012	1101000	1104000	1213380
29		31	888650	885720	1097330	1024000	1604400
30		32	486500	644160	473430	600000	756000
31	Июль	33	889000	1007598	917133	990400	1213800
32		37	1045100	1115934	1021361	1208000	1498980
33		38	1144500	1456680	1266150	1152000	1697640
34		39	1676500	1837320	1713890	1824000	2055060
35	Август	40	1687000	1665300	1843441	1956000	2100000
36		41	525000	615978	532150	520000	743400
37		42	1011500	1113738	1020260	1192000	1302000

Рис. 3.2. Таблица данных о продажах утеплителя за 5 лет

Для корректного прогнозирования масштабируем данные по продажам в системе Matlab. Листинг кода приведен ниже:

Листинг 3.1.

$Min = \min(A(:,5));$

$Max = \max(A(:,5));$

$B = Min;$

$K = 1/(Max - Min);$

$A(:,5) = (A(:,5) - B) .* K;$

В результате получаем данные продаж в интервале [0,1]. Таблица данных продаж в среде MATLAB изображена на рисунке 3.3:

	1	2	3	4	5	6
2	0.0049	0	0.0025	0.0062	0.0110	
3	0.0184	0.0145	0.0100	0.0235	0.0129	
4	0.0046	0.0202	0.0148	0.0196	0	
5	0.0508	0.0667	0.0484	0.0492	0.0631	
6	0.0458	0.0570	0.0442	0.0531	0.0735	
7	0.0647	0.0725	0.0639	0.0940	0.0860	
8	0.1035	0.1014	0.1185	0.1151	0.0963	
9	0.2587	0.2898	0.1841	0.2112	0.3773	
10	0.2962	0.3384	0.2712	0.2534	0.3665	
11	0.3714	0.3642	0.2887	0.3685	0.4302	
12	0.6276	0.5049	0.4844	0.6367	0.8004	
13	0.7391	0.7294	0.6772	0.6092	1	
14	0.6208	0.8473	0.5493	0.7964	0.9823	
15	0.6494	0.6338	0.7421	0.8447	0.8488	
16	0.6375	0.6804	0.6878	0.7254	0.7298	
17	0.5174	0.5223	0.4834	0.5300	0.6237	
18	0.4435	0.3662	0.3907	0.3970	0.5011	
19	0.4204	0.5300	0.4727	0.4539	0.5177	

Рис. 3.3. Данные продаж в интервале [0,1]

Как видно из таблицы товар обладает свойством сезонности. Пик продаж приходится на весну-осень, зимой же наблюдаются минимальные продажи.

Ввиду выраженной сезонности товара, компании очень важно прогнозировать его объемы продаж для понимания того, сколько данного товара нужно завести на склад.

Поэтому в данном случае очень важно учитывать этот коэффициент. Коэффициент сезонности – это показатель, на который изменяется социальная норма закупок строительного товара в зависимости от времени года.

Для высчитывания коэффициента сезонности в нашей таблице в системе MS Excel рассчитываем среднее значение продаж за каждый год (т.е. сумму продаж за весь год делим на количество месяцев).

По формуле рассчитываем коэффициент сезонности для каждой недели

$$k_{\text{сезон.}} = N_i / s.z., \quad (3.1)$$

где  $N_i$  – фактическая сумма продаж за определённую неделю;  $s.z.$  – среднее значение по неделям.

Кроме этого, нужно учесть коэффициенты инфляции, а так же коэффициент изменения рубля по отношению к доллару.

Коэффициент инфляции - это показатель, характеризующий соотношение цен прошедшего и текущего периодов. Данные об инфляции на потребительском рынке были взяты с официального сайта Центрального Банка России за период с 2012 г. по 2016 г.

Для вычисления коэффициента изменения рубля по отношению к доллару США были взяты данные динамики курса валюты доллар США за период с 2012 г. по 2016 г. так же с официального сайта ЦБРФ.

Коэффициент изменения рубля по отношению к доллару США высчитывался аналогично коэффициенту сезонности. На рисунке 3.4 приведена полученная в результате сводная таблица входных данных для обучения нейронной сети обратного распространения.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
	Месяц	Неделя	2012				2013				2014				2015				2016
			Продажи	Козф.инф.	Козф.сез.	Козф.КВ	Продажи	Козф.инф.	Козф.сез.	Козф.КВ	Продажи	Козф.инф.	Козф.сез.	Козф.КВ	Продажи	Козф.инф.	Козф.сез.	Козф.КВ	Продажи
3		1	0,005	0,13	0	0,656	0	0,252	0	0,641	0,003	0,153	0	0,557	0,006	1	0	0,618	0,011
4	Январь	2	0,018	0,13	0,006	0,664	0,014	0,252	0	0,639	0,01	0,153	0,003	0,566	0,024	1	0,007	0,689	0,013
5		3	0,005	0,13	0,023	0,667	0,02	0,252	0,017	0,638	0,015	0,153	0,012	0,573	0,02	1	0,027	0,714	0
6		4	0,051	0,13	0,006	0,665	0,067	0,252	0,024	0,632	0,048	0,153	0,018	0,6	0,049	1	0,023	0,757	0,063
7		5	0,046	0,096	0,064	0,671	0,057	0,145	0,079	0,632	0,044	0,182	0,06	0,6	0,053	0,577	0,057	0,765	0,074
8	Февраль	6	0,065	0,096	0,057	0,679	0,072	0,145	0,067	0,634	0,064	0,182	0,054	0,592	0,094	0,577	0,061	0,719	0,086
9		7	0,103	0,096	0,081	0,68	0,101	0,145	0,086	0,632	0,118	0,182	0,079	0,606	0,115	0,577	0,108	0,678	0,096
10		8	0,259	0,096	0,13	0,678	0,29	0,145	0,12	0,645	0,184	0,182	0,146	0,614	0,211	0,577	0,133	0,673	0,377
11	Март	9	0,296	0,151	0,324	0,681	0,338	0,088	0,342	0,643	0,271	0,265	0,226	0,616	0,253	0,314	0,243	0,683	0,367
12		10	0,371	0,151	0,371	0,665	0,364	0,088	0,399	0,647	0,289	0,265	0,333	0,62	0,368	0,314	0,292	0,667	0,43
13		11	0,628	0,151	0,465	0,674	0,505	0,088	0,43	0,651	0,484	0,265	0,355	0,615	0,637	0,314	0,424	0,659	0,8
14		12	0,739	0,151	0,786	0,675	0,729	0,088	0,596	0,654	0,677	0,265	0,595	0,608	0,609	0,314	0,733	0,643	1
15		13	0,621	0,081	0,925	0,668	0,847	0,132	0,861	0,655	0,549	0,234	0,832	0,607	0,796	0,119	0,702	0,633	0,982
16	Апрель	14	0,649	0,081	0,777	0,667	0,634	0,132	1	0,653	0,742	0,234	0,675	0,606	0,845	0,119	0,917	0,561	0,849
17		15	0,637	0,081	0,813	0,685	0,68	0,132	0,748	0,667	0,688	0,234	0,912	0,606	0,725	0,119	0,973	0,566	0,73
18		16	0,517	0,081	0,798	0,703	0,522	0,132	0,803	0,658	0,483	0,234	0,845	0,608	0,53	0,119	0,835	0,568	0,624
19	Май	17	0,443	0,135	0,648	0,697	0,366	0,171	0,616	0,653	0,391	0,234	0,594	0,609	0,397	0,091	0,61	0,562	0,501
20		18	0,42	0,135	0,555	0,691	0,53	0,171	0,432	0,654	0,473	0,234	0,48	0,6	0,454	0,091	0,457	0,553	0,518
21		19	0,368	0,135	0,526	0,68	0,406	0,171	0,626	0,66	0,44	0,234	0,581	0,59	0,505	0,091	0,523	0,541	0,542
22		20	0,209	0,135	0,46	0,696	0,29	0,171	0,479	0,665	0,269	0,234	0,54	0,592	0,264	0,091	0,581	0,582	0,335
23		21	0,309	0,231	0,261	0,695	0,376	0,109	0,342	0,669	0,484	0,161	0,331	0,577	0,338	0,049	0,304	0,61	0,447
24	Июнь	22	0,491	0,231	0,387	0,693	0,482	0,109	0,444	0,682	0,455	0,161	0,595	0,577	0,422	0,049	0,389	0,591	0,56
25		23	0,185	0,231	0,614	0,713	0,183	0,109	0,569	0,688	0,215	0,161	0,56	0,599	0,176	0,049	0,486	0,602	0,216
26		24	0,206	0,231	0,231	0,703	0,153	0,109	0,216	0,688	0,291	0,161	0,264	0,609	0,291	0,049	0,203	0,58	0,32
27	Июль	25	0,235	0,319	0,257	0,709	0,234	0,213	0,18	0,691	0,29	0,127	0,357	0,604	0,27	0,208	0,336	0,614	0,424
28		26	0,128	0,319	0,294	0,703	0,17	0,213	0,276	0,693	0,125	0,127	0,356	0,614	0,158	0,208	0,311	0,622	0,2
29		27	0,235	0,319	0,161	0,704	0,266	0,213	0,201	0,683	0,242	0,127	0,154	0,617	0,261	0,208	0,182	0,624	0,32
30		28	0,276	0,319	0,294	0,711	0,295	0,213	0,314	0,692	0,27	0,127	0,298	0,629	0,319	0,208	0,301	0,648	0,395

Рис 3.4. Сводная таблица входных данных обучения НС

Теперь подготовим обучающую и тестовую выборки. Для этого разделим полученную таблицу на две таблицы: одна выборка для обучения, вторая – для тестирования работы НС.

На рисунке 3.5. представлена тестовая выборка, с помощью которой будем проверять точность прогноза, сделанного обученной нейронной сетью.

Для этого мы будем подавать значения за четыре года, получать прогноз от нейронной сети и сравнивать с реальными данными.

0,285	0,088	0,54	0,648	0,352	0,145	0,499	0,687	0,331	0,332	0,543	0,782	0,371	0,195	0,542	0,708	0,421
0,172	0,088	0,357	0,646	0,199	0,145	0,415	0,689	0,192	0,332	0,407	0,796	0,211	0,195	0,427	0,713	0,213
0,091	0,088	0,215	0,655	0,099	0,145	0,234	0,699	0,102	0,332	0,236	0,84	0,094	0,195	0,243	0,728	0,111
0,051	0,14	0,113	0,656	0,046	0,132	0,116	0,698	0,038	0,681	0,125	0,883	0,053	0,2	0,108	0,733	0,073
0,009	0,14	0,064	0,68	0,014	0,132	0,055	0,69	0,009	0,681	0,046	0,925	0,011	0,2	0,061	0,761	0,009
0	0,14	0,012	0,684	0	0,132	0,017	0,694	0	0,681	0,011	0,962	0	0,2	0,013	0,783	0

Рис. 3.5. Тестовые данные

### 3.2. Проведение эксперимента

В ходе эксперимента будем использовать сформированную матрицу значений продаж, а так же тестовую выборку для проверки результата и прогнозирования.

Смоделируем однослойную нейронную сеть с характеристиками: 1 слой, 2 нейрона. Запустим полученную программу в Matlab. На рисунке 3.6 изображены параметры нейронной сети. Параметры Gradient, Mu показывают изменения значений градиента функции и математического ожидания соответственно при каждой итерации. А параметр Val Fail показывает количество итераций, при которых ошибка была меньше заданной.

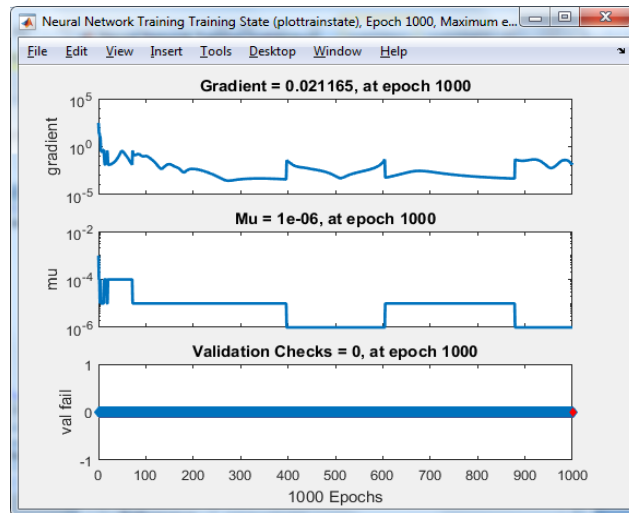


Рис. 3.6. Параметры модели нейронной сети

На рисунке 3.7 приведен график зависимости величины ошибки от количества итераций. По оси X значение эпох, по оси Y – значения суммарной среднеквадратичной ошибки.

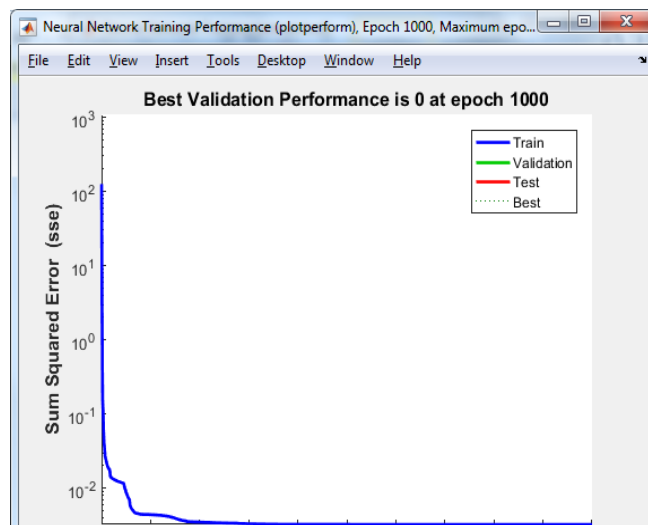


Рис 3.7. Зависимость суммарной квадратичной ошибки от количества эпох

Из графика видно, что с каждой следующей итерацией значение ошибки уменьшается. Для 1000 эпохи значение суммарной квадратичной ошибки составляет  $2.04 \cdot 10^{-3}$ .



Далее используя нашу тестовую выборку, спрогнозируем пять значений, и сравним спрогнозированные данные с реальными. На рисунке 3.8 представлен график, где кружочком выделены реальные данные, а звездочкой – спрогнозированные.

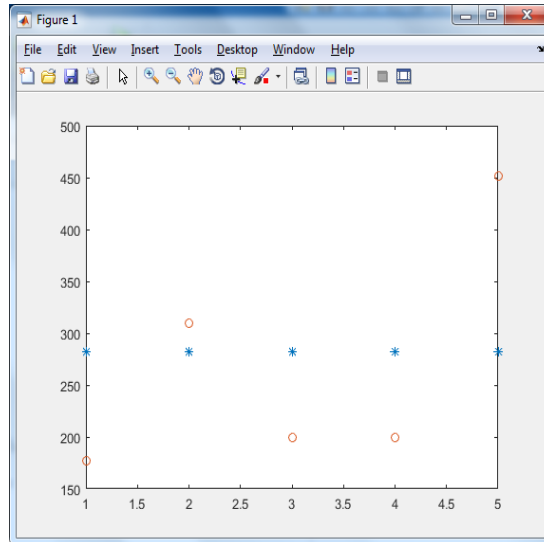
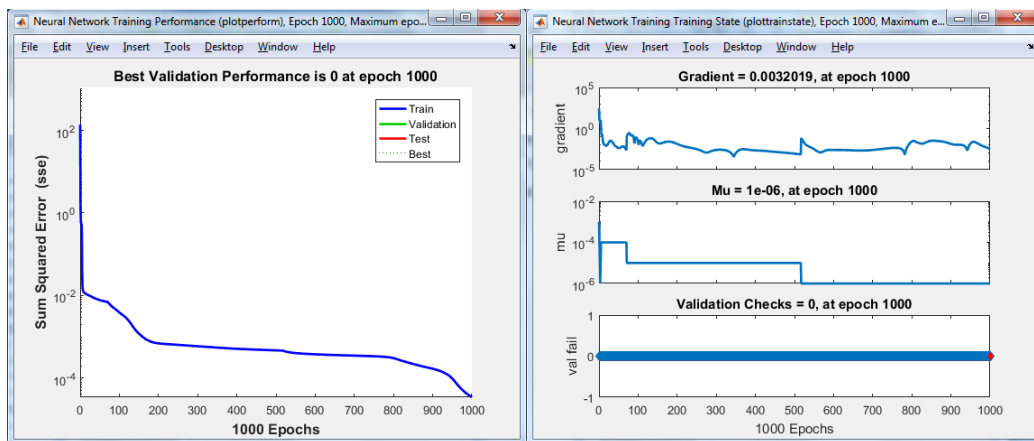


Рис. 3.8. График спрогнозированных и реальных данных

Как видно из графика реальные данные достаточно сильно отличаются от данных, которые спрогнозировала нейронная сеть.

Продолжая анализ, обучим нейронную сеть со следующими параметрами: 1 слой, 10 нейронов. Результаты представлены на рисунке 3.9:



а)

б)

Рисунок 3.9. а) Параметры модели нейронной сети;

б) Зависимость суммарной квадратичной ошибки от количества эпох

Из графиков видно, что значение суммарной квадратичной ошибки стало меньше, теперь оно равно  $2,04 \cdot 10^{-4}$ .

Проверяем работу нейронной сети для прогнозирования, для этого построим график прогнозных и реальных значений (рисунок 3.10).

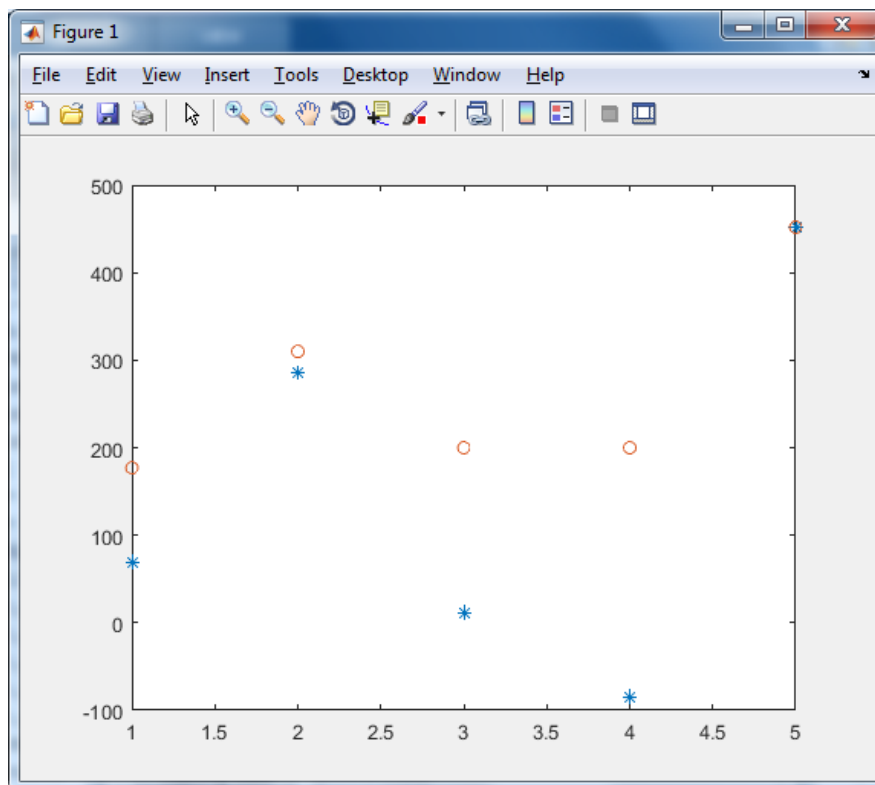
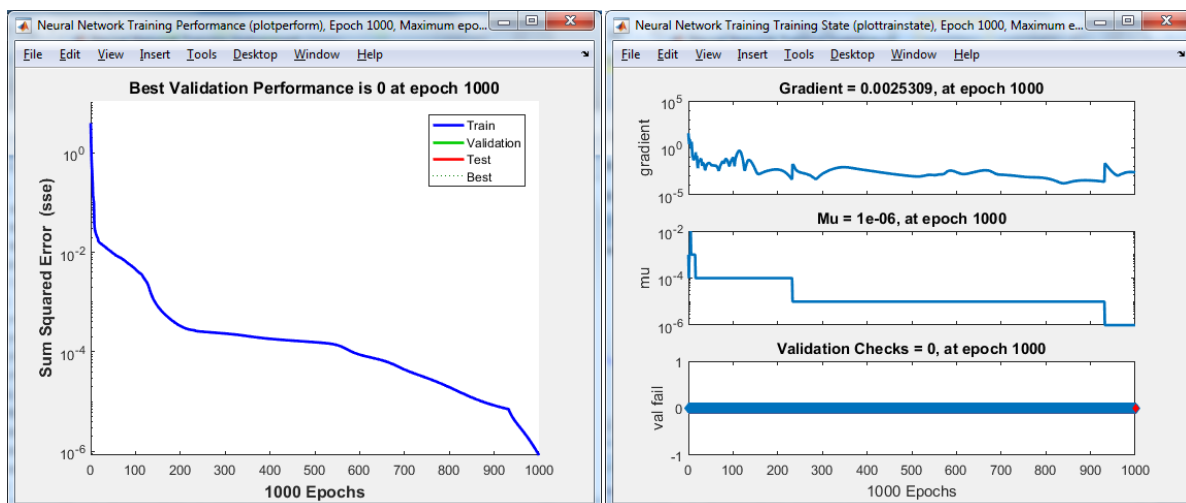


Рис 3.10. Тестирование нейронной сети

Из графика видно, что сеть с данными характеристиками лучше осуществляет прогноз. Однако по-прежнему прогнозные данные значительно отличаются от реальных.

Далее от однослойной сети перейдем к более сложной архитектуре – двухслойной нейронной сети. Зададим следующие параметры [2 3 1]. Результаты ее тестирования приведены на рисунке 3.11:



а)

б)

Рисунок 3.11. а) Параметры модели нейронной сети;  
 б) Зависимость суммарной квадратичной ошибки от количества эпох

Построив график разности прогнозных и реальных значений (рисунок 3.12), наблюдаем, что двухслойная нейронная сеть лучше справляется с задачей прогнозирования. Ошибка SSE обучения сети стала равна  $2,04 \cdot 10^{-5}$ .

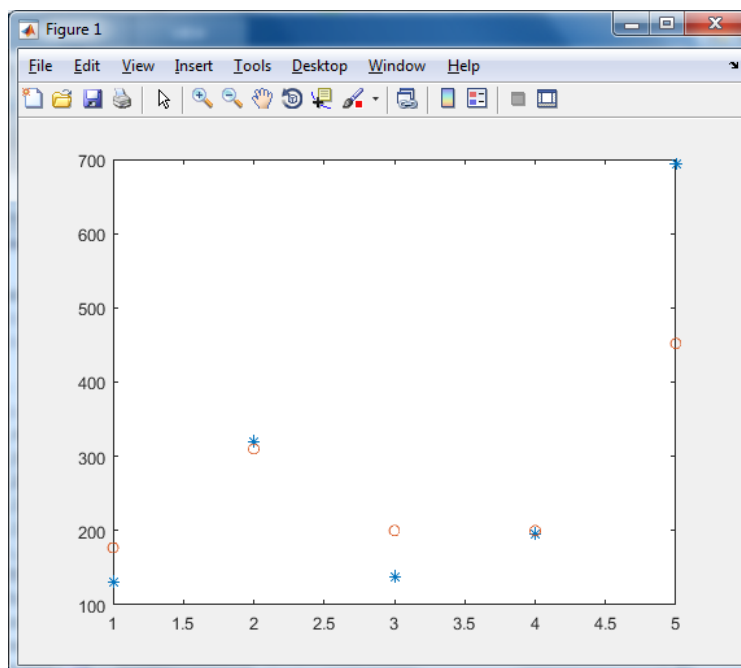


Рис. 3.12. График отличия прогнозных данных от реальных

Теперь увеличим количество нейронов в каждом слое, задав следующие параметры [8 8 1].

В результате получаем следующие характеристики сети представленные на рисунке 3.13:

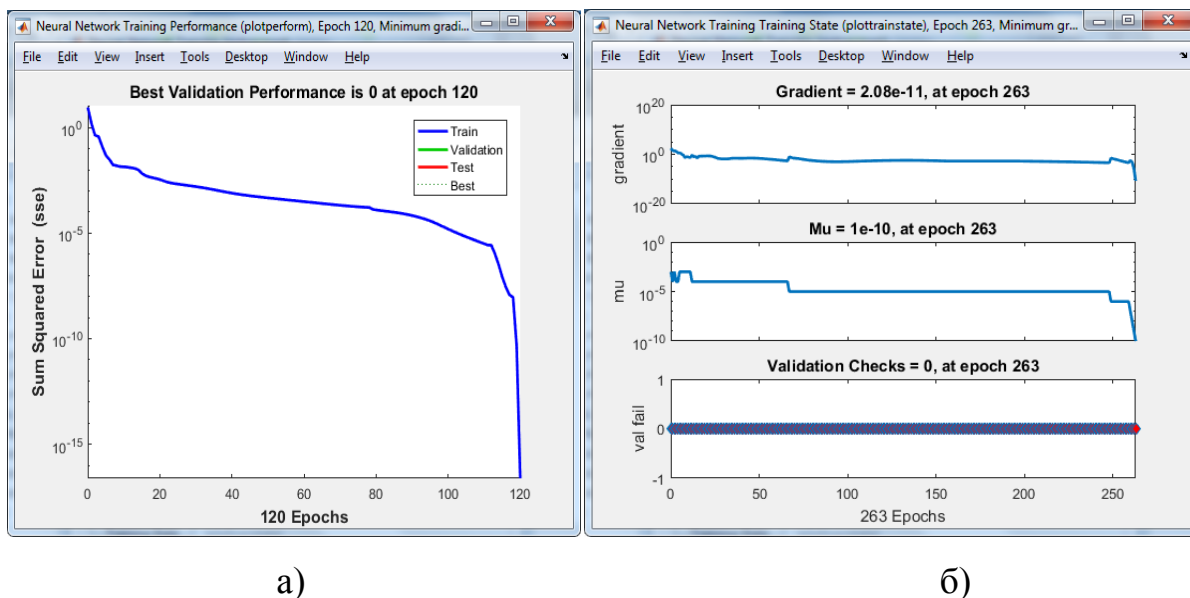


Рисунок 3.13. а) Параметры модели нейронной сети;  
б) Зависимость суммарной квадратичной ошибки от количества эпох

На графике а) видно, что сеть обучилась гораздо быстрее. Для обучения ей потребовалось 120 итераций. Ошибка суммарной квадратичной ошибки в этом случае составила  $2,04 \cdot 10^{-15}$ .

Теперь протестируем сеть и построим график прогнозных и реальных значений. На рисунке 3.14 видно, что прогнозные значения приблизились к реальным. Данная структура сети является приемлемой для прогнозирования потребительского спроса строительных товаров.

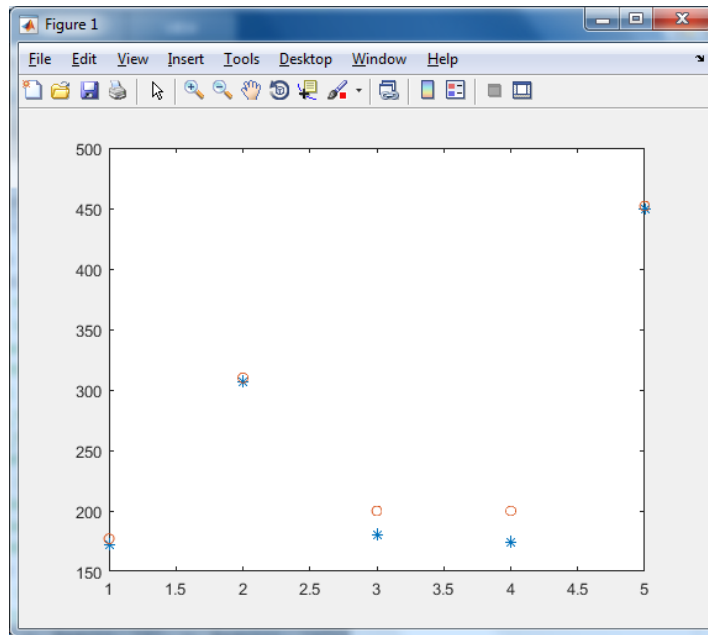


Рис. 3.14. График прогнозных и реальных значений продаж

Увеличивая количество нейронов в скрытом слое, кажется, что чем больше нейронов добавим, тем правильнее и точнее будет результат. Однако это не так. Взяв следующие параметры сети [20 10 1], заметим, что сеть стала обучаться с большей ошибкой и прогнозные данные существенно отличаются от реальных.

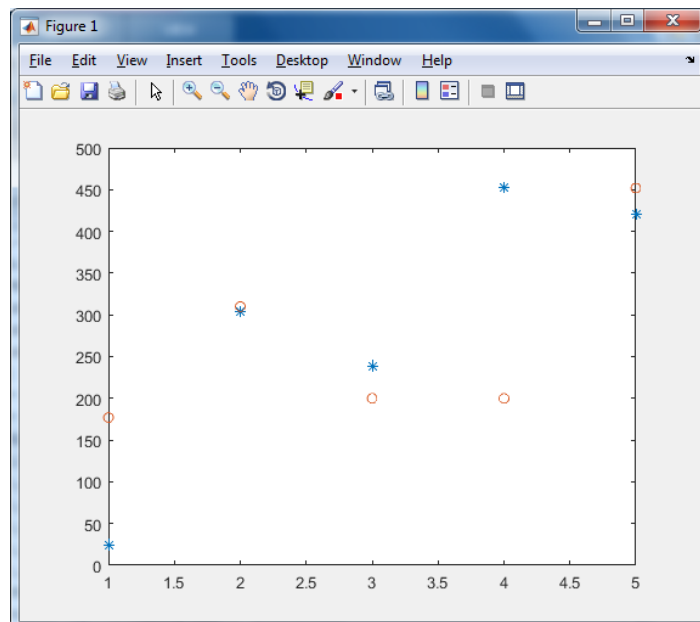


Рис 3.15. График отличия прогнозных данных от реальных

Дальше эксперимент будет проводиться с использованием ANFIS – редактора. Для его вызова в командной строке вводим anfisedit.

Для обучения и прогнозирования в ANFIS создадим два файла: training.dat, testing.dat. В них будут содержаться данные, необходимые для обучения нейронной сети и тестовые данные для прогнозирования. Данные в них – матрицы, в которой 9 столбцов – 8 входных данных и 1 выходной. В первом файле – 48 строк, во втором – 6.

Содержимое файла training.dat представлено на рисунке 3.16:

Файл	Правка	Формат	Вид	Справка					
0.003	0.153	0	0.557	0.006	1	0	0.618	0.011	
0.01	0.153	0.003	0.566	0.024	1	0.007	0.689	0.013	
0.015	0.153	0.012	0.573	0.02	1	0.027	0.714	0	
0.048	0.153	0.018	0.6	0.049	1	0.023	0.757	0.063	
0.044	0.182	0.06	0.6	0.053	0.577	0.057	0.765	0.074	
0.064	0.182	0.054	0.592	0.094	0.577	0.061	0.719	0.086	
0.118	0.182	0.079	0.606	0.115	0.577	0.108	0.678	0.096	
0.184	0.182	0.146	0.614	0.211	0.577	0.133	0.673	0.377	
0.271	0.265	0.226	0.616	0.253	0.314	0.243	0.683	0.367	
0.289	0.265	0.333	0.62	0.368	0.314	0.292	0.667	0.43	
0.484	0.265	0.355	0.615	0.637	0.314	0.424	0.659	0.8	
0.677	0.265	0.595	0.608	0.609	0.314	0.733	0.643	1	
0.549	0.234	0.832	0.607	0.796	0.119	0.702	0.633	0.982	
0.742	0.234	0.675	0.606	0.845	0.119	0.917	0.561	0.849	
0.688	0.234	0.912	0.606	0.725	0.119	0.973	0.566	0.73	
0.483	0.234	0.845	0.608	0.53	0.119	0.835	0.568	0.624	
0.391	0.234	0.594	0.609	0.397	0.091	0.61	0.562	0.501	
0.473	0.234	0.48	0.6	0.454	0.091	0.457	0.553	0.518	
0.44	0.234	0.581	0.50	0.505	0.091	0.522	0.541	0.512	

Рис. 3.16. Входные данные для ANFIS-редактора

Содержимое файла testing.dat представлено на рисунке 3.17:

Файл	Правка	Формат	Вид	Справка					
0.331	0.332	0.543	0.782	0.371	0.195	0.542	0.708	0.421	
0.192	0.332	0.407	0.796	0.211	0.195	0.427	0.713	0.213	
0.102	0.332	0.236	0.84	0.094	0.195	0.243	0.728	0.111	
0.038	0.681	0.125	0.883	0.053	0.2	0.108	0.733	0.073	
0.009	0.681	0.046	0.925	0.011	0.2	0.061	0.761	0.009	
0	0.681	0.011	0.962	0	0.2	0.013	0.783	0	

Рис. 3.17. Тестовые данные для ANFIS-редактора

В меню Load data выбираем Training и From disk, нажимаем кнопку load data. В появившемся окне выбираем созданный ранее файл training.dat.

В меню Load data выбираем Testing и From disk, нажимаем кнопку load data. В появившемся окне выбираем созданный ранее файл testing.dat.

Данные для обучения и проверки загружены (рисунок 3.18).

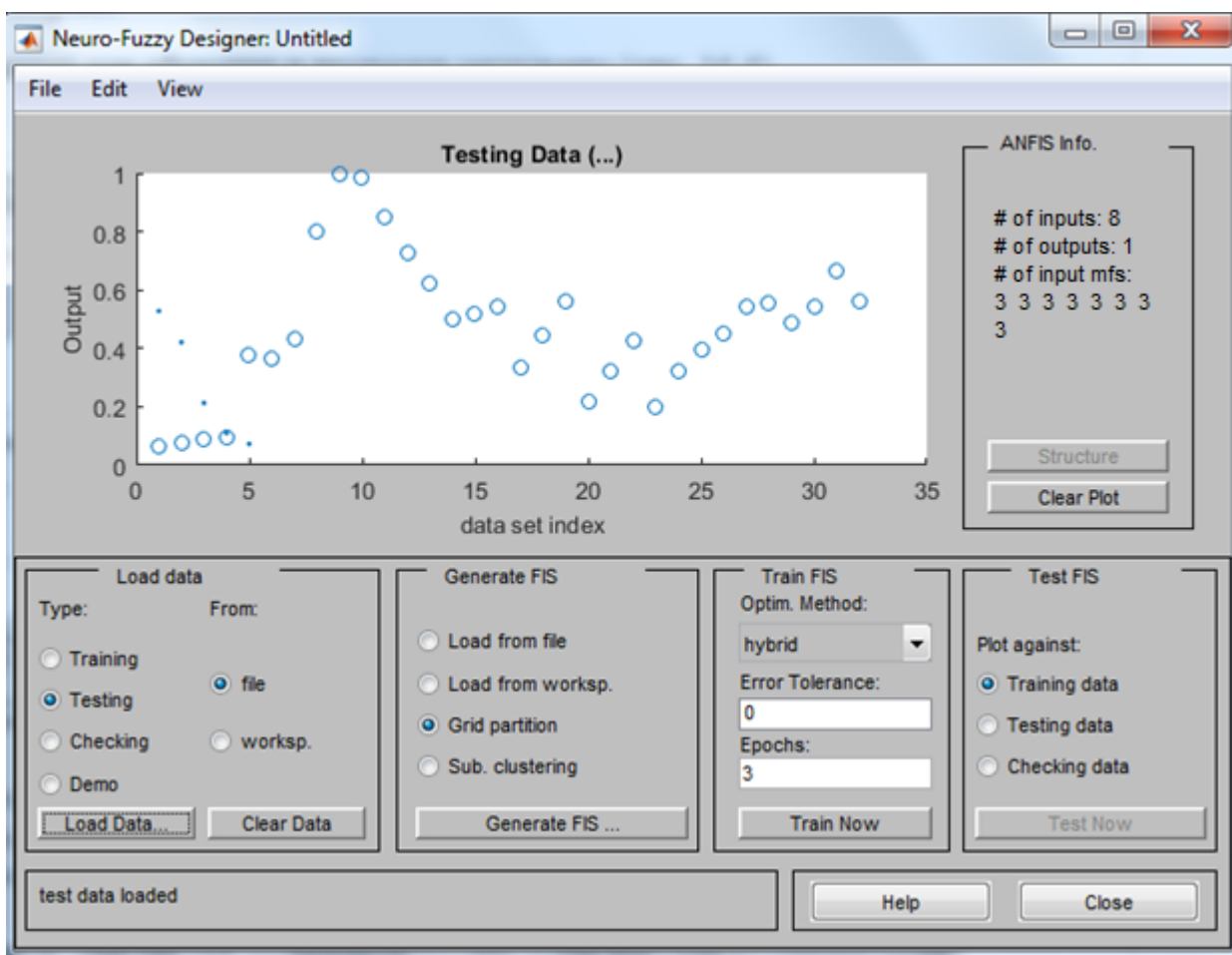


Рис. 3.18. Загрузка входных данных в ANFIS редактор

После нажатия Generate FIS выбираем модель сети. В данном случае в модели восемь входных переменных, каждой из которых соответствуют по 3 терма типа gaussmf. Выходная переменная задается линейной функцией (рисунок 3.19).

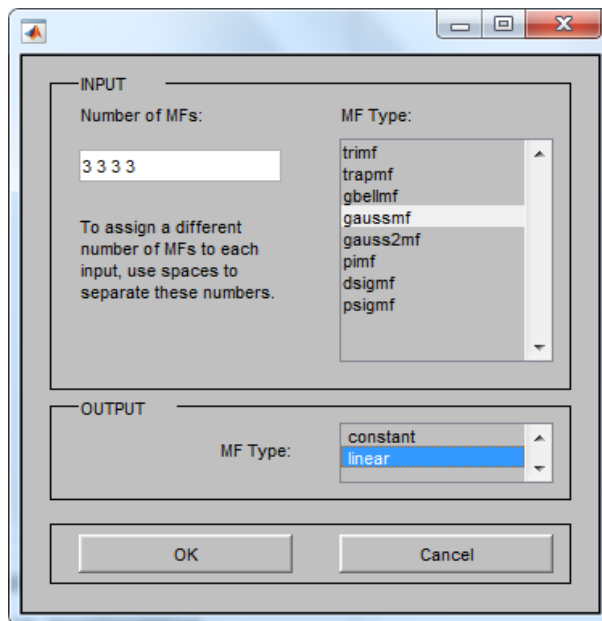


Рис. 3.19. Настройка гибридной сети

После загрузки данных, производим обучение. График значений ошибки обучения представлен на рисунке 3.20.

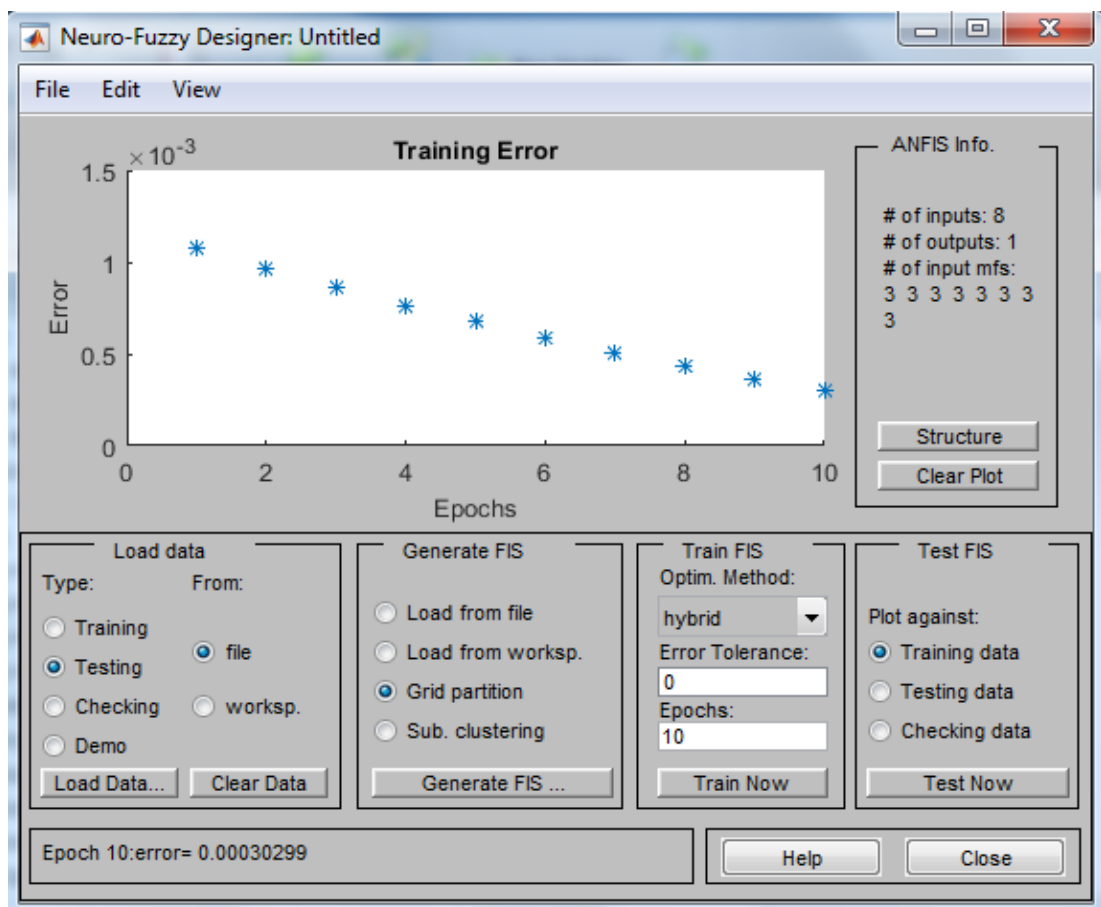


Рис. 3.20. График зависимости ошибки обучения от количества эпох



Ошибка обучения составила всего 0.000300299. Теперь проверим обученную сеть на тестовой выборке. Результаты тестирования представлены на рисунке 3.21:

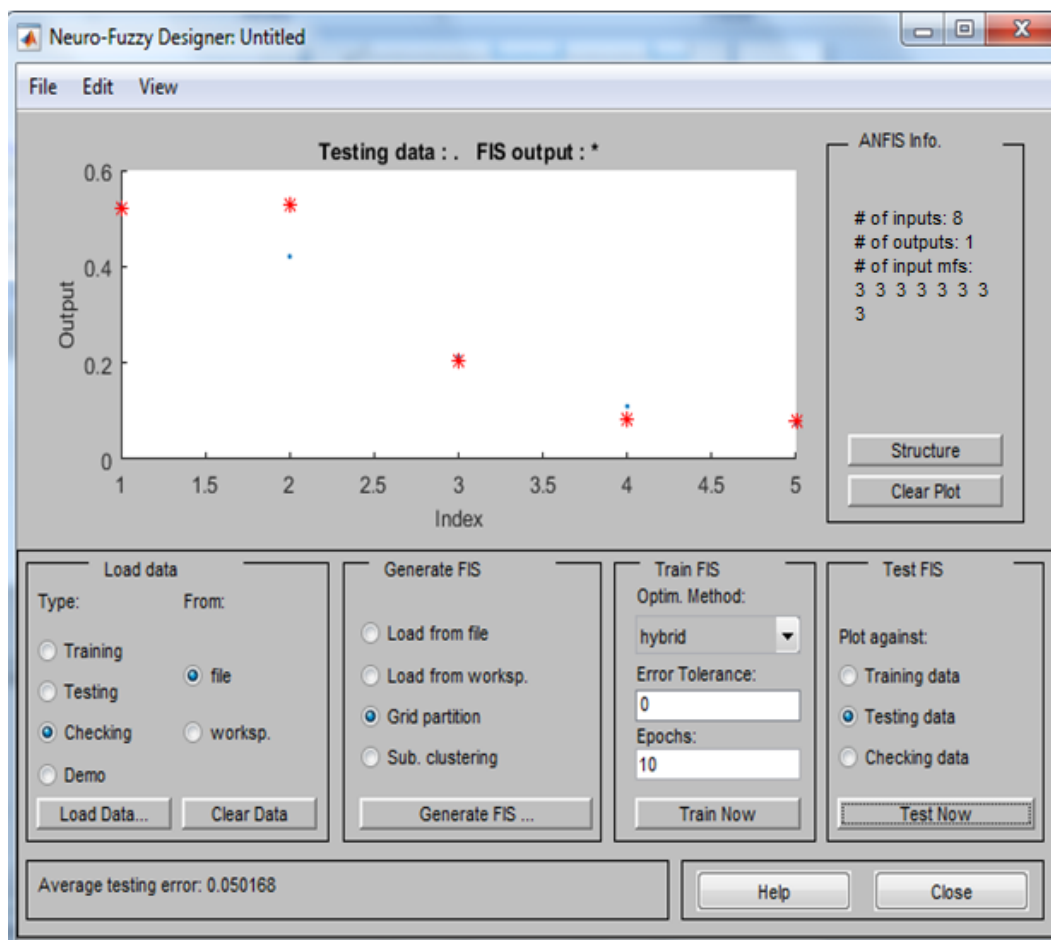


Рис. 3.21. Разница между прогнозными значениями и реальными данными

Как видно из рисунка, гибридная сеть так же дает достаточно точный прогноз, а значит, может использоваться для поставленной задачи. Следует отметить, что использование большого количества входов делает обучение НС ресурсоемким.

### 3.3. Сравнительный анализ методов

Для сравнения методов обратного распространения и гибридного метода, сравним ошибки, полученные при обучении.

На рисунке 3.22 представлена гистограмма ошибок, полученных при обучении методом обратного распространения.

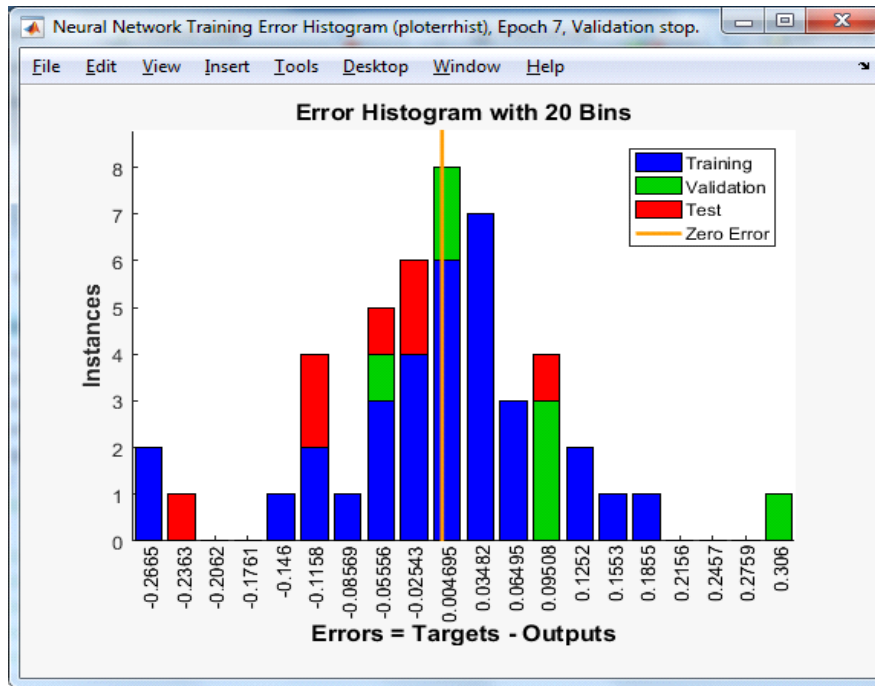


Рис. 3.22. Гистограмма ошибок метода обратного распространения

На рисунке 3.23 отображен график зависимости ошибки обучения от количества эпох гибридной нейронной сети. Как видно из графика, с каждой итерацией обучения ошибка становится меньше.

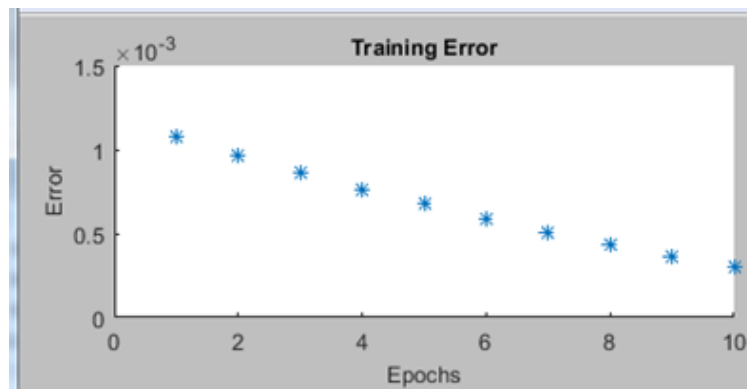


Рис. 3.23. График ошибки обучения гибридной НС

Для наглядности сведем данные графиков в одну таблицу. В таблице отображены: № эпохи, название методов и соответствующие каждому методу значения ошибок.

Таблица 3.1

Сводные данные значений ошибки обучения

№ эпохи	Значение ошибки обучения	
	Метод обратного распространения	Гибридная сеть
1	0.306	0.0115
2	0.2759	0.0089
3	0.2457	0.0081
4	0.2156	0.0075
5	0.1855	0.0069
6	0.1553	0.0063
7	0.1252	0.0048
8	0.09508	0.0043
9	0.03482	0.0037
10	0.004695	0.003

Анализируя данные таблицы 3.1, делаем вывод о целесообразности применения гибридной сети для прогнозирования спроса в сегменте строительных товаров, в связи с наименьшим значением ошибки обучения. Однако следует отметить, что гибридные сети на большом количестве входных данных требуют больших аппаратных ресурсов.

## ЗАКЛЮЧЕНИЕ

Цель настоящей магистерской диссертации заключалась в выявлении оптимальной методики прогнозирования с использованием нейронных сетей в отрасли строительных материалов.

Для достижения указанной цели были выполнены определенные во введении задачи. А именно: изучены методы нейросетевого прогнозирования, из которых были взяты два метода для эксперимента, а так же проведен анализ программных средств для решения задачи.

На этапе проектирования сформирована структура нейронной сети, выбран алгоритм обучения. После чего выполнено моделирование нейронной сети для прогнозирования в системе MATLAB.

Собирая входные данные для обучения НС, был проведен анализ временного ряда на стационарность и сезонность. В результирующую таблицу входных данных были добавлены коэффициенты, влияющие на продажи. Данные, подаваемые на вход НС были отмасштабированы в интервале  $[0,1]$ .

Проведение эксперимента показало, что НС не только способны прогнозировать потребительский спрос строительных товаров, но и показывать при этом достаточно точный результат. На основании сравнения значений ошибки сети прогнозирования была определена оптимальная методика прогноза.

Таким образом, задачи решены в полном объеме, цель достигнута – определена оптимальная методика прогнозирования с использованием нейронных сетей в отрасли стройматериалов.

## БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Бильгаева Л.П., Власов К.Г. ПРОГНОЗИРОВАНИЕ ПРОДАЖ В СРЕДЕ MATLAB // Естественные и математические науки в современном мире: сб. ст. по матер. XLIX междунар. науч.-практ. конф. № 12(47). – Новосибирск: СибАК, 2016. – с. 64– 76.
2. Бушуева, Л. И. Методы прогнозирования объема продаж / Л. И. Бушуева // Маркетинг. 2001. – № 5. – с. 50 – 53
3. Галушкин А.И. Теория нейронных сетей. Кн.1,- М.: ИПРЖР, 2001. – 416с
4. Головкин В.А.. Нейронные сети: обучение, организация и применение. М., ИПРЖР, 2001. – 256с.
5. Головкин В.А.. От многослойных персептронов к нейронным сетям глубокого доверия: парадигмы обучения и применение. В сб.: Нейроинформатика-2015. XVII Всероссийская научно-техническая конференция с международным участием. Лекции по нейроинформатике, с.47–. НИЯУ МИФИ, 2015.
6. Горбань, А. Н. Обучение нейронных сетей / А. Н. Горбань М.: Изд. СССР-США СП "ParaGraph", 1990. – 160с.
7. Горбань А.Н. Нейронные сети на персональном компьютере / А. Н. Горбань, Д. А. Россиев Новосибирск: Наука. Сибирская издательская фирма РАН, 1996. – 276с.
8. Горбань А.Н., Дунин-Барковский В.Л., Кардин А.Н. и др. Нейроинформатика, Отв. Ред. Новиков Е.А., РАН, Сиб. Отд., Институт выч. Моделирования – Новосибирск: Наука, 1998. – 296с.
9. Дебок Г., Кохонен Т.. Анализ финансовых данных с помощью самоорганизующихся карт. М., "Альпина", 2001. – 317с.
10. Дьяконов В.П., Круглов В.В. MATLAB 6.5 SP1/7/7 SP1/7 SP2 + Simulink 5/6. Инструменты искусственного интеллекта и биоинформатики. Серия «Библиотека профессионала». – М.: СОЛОН-ПРЕСС, 2006. – 456с.

11. Дьяконов В. П. Matlab 6: учебный курс. – СПб.: Питер, 2001. – 592с.
12. Еремин Д. М., Гарцеев И. Б. Искусственные нейронные сети в интеллектуальных системах управления. — М.: МИРЭА, 2004. – 75с.
13. Ерохина Л.И. Прогнозирование, планирование в системе управления на предприятиях сферы сервиса / Л.И. Ерохина, Е.В. Башмачникова; Тольят. гос. ин-т сервиса. – М.: КНОРУС, 2004. – 202с.
14. Каллан Р. Основные концепции нейронных сетей = The Essence of Neural Networks First Edition. – М.: Вильямс, 2001. – 288с.
15. Комарцова Л.Г., Максимов А.В.. Нейрокомпьютеры. М., Изд-во МГТУ им. Баумана, 2004. – 400с.
16. Кохонен Т. Самоорганизующиеся карты. М., "Бином", 2008. – 656с.
17. Круглов В.В., Дли М.И., Голунов Р.Ю.. Нечёткая логика и искусственные нейронные сети. Физматлит, 2001. –201с.
18. Круглов В. В., Борисов В. В. Искусственные нейронные сети. Теория и практика. – М.: Горячая линия - Телеком, 2001. – 382 с.
19. Назаров, А. В., Нейросетевые алгоритмы прогнозирования и оптимизации систем / А. В. Назаров, А. И. Лоскутов СПб.: Наука и техника, 2003 – 384с.
20. Прогнозирование и планирование в условиях рынка: Учеб. Пособие для вузов. / Под ред. Т.Г. Морозовой, А.В. Пикулькина. – М.: ЮНИТИ–ДАНА, 2001. – 318с.
21. Ромашова И.Б. Прогнозирование в системе управления современным предприятием / И.Б. Ромашова. – Нижний Новгород: Нижегород. гос. университет им. Н.И. Лобачевского, 2000. – 328с.
22. Тадеусевич Рышард, Боровик Барбара, Гончаж Томаш, Леппер Бартош. Элементарное введение в технологию нейронных сетей с примерами программ / Перевод И. Д. Рудинского. — М.: Горячая линия — Телеком, 2011. – 408с.
23. Тархов Д.А.. Нейронные сети. Модели и алгоритмы. (Справочник.) М., Радиотехника, 2005. –256с.

24. Терехов В. А., Ефимов Д. В., Тюкин И. Ю. Нейросетевые системы управления. — М.: Высшая школа, 2002. —184с.
25. Уоссермен, Ф. Нейрокомпьютерная техника: Теория и практика. —М.: Мир, 1992. —184с.
26. Файоль А. Управление — это наука и искусство / А. Файоль, Г. Эмерсон, Ф. Тэйлор, Г. Форд. — М.: Республика, 1992. — 351с.
27. Хайкин С.. Нейронные сети: полный курс. 2–е изд. М., "Вильямс", 2006. —1104с.
28. Четыркин, Е. М. Статистические методы прогнозирования / Е. М. Четыркин-М.: Статистика, 1997. —200 с.
29. Чубукова И.А., Data Mining, Лекция 11: Методы классификации и прогнозирования. Нейронные сети, с.155-159, 2006.
30. Яхьяева Г.Э.. Нечёткие множества и нейронные сети. М., Интернет-Университет Информационных Технологий; БИНОМ. Лаборатория знаний, 2006. —315с.
31. Яхьяева Г.Э.. Основы теории нейронных сетей. Интернет-университет информационных технологий, изд-во "Открытые системы",2016. — 200с.
32. Kohonen T. 1984. Self-organization and associative memory. Series in Information Sciences, vol. 8. Berlin: Springer Verlag.
33. Hastie T., Tibshirani R., Friedman J. The Elements of Statistical Learning. — Springer, 2001.
34. АНС «СибАК» — [Электронный ресурс] / Научно-практические конференции ученых и студентов. Публикации Scopus. Авторские и коллективные монографии — Режим доступа: [www.sibac.info](http://www.sibac.info) , свободный.
35. Центральный банк Российской Федерации/ официальный сайт / [Электронный ресурс]. — Режим доступа:[www.cbr.ru](http://www.cbr.ru), свободный
36. Oracle Cryslat Ball. / — [Электронный ресурс]. — Режим доступа: [www.hyperion.ru/products/appl/r\\_pr/](http://www.hyperion.ru/products/appl/r_pr/) , свободный

37. MATLAB – book1. / – [Электронный ресурс]. – Режим доступа:  
[www.matlab.exponenta.ru/neuralnetwork/book1/index.php](http://www.matlab.exponenta.ru/neuralnetwork/book1/index.php) , свободный
38. MATLAB – book2. / – [Электронный ресурс]. – Режим доступа:  
[www.matlab.exponenta.ru/neuralnetwork/book2/18/trainlm.php](http://www.matlab.exponenta.ru/neuralnetwork/book2/18/trainlm.php) , свободный
39. Вебинар «Машинное обучение с MATLAB» – [Электронный ресурс]. –  
Режим доступа: <http://matlab.ru/webinars/mashinnoe-obuchenie-s-matlab> ,  
требуется регистрация на сайте.
40. MATLABinRussia. Официальный Youtube канал на русском языке –  
[Электронный ресурс]. – Режим доступа:  
<https://www.youtube.com/user/MATLABinRussia> , свободный.