

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ  
ФЕДЕРАЦИИ**

Федеральное государственное бюджетное образовательное учреждение  
высшего профессионального образования

«ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ  
УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ» (ТУСУР)

Н. В. ЗАМЯТИН

**НЕЧЕТКАЯ ЛОГИКА И НЕЙРОННЫЕ СЕТИ**

Учебное пособие

Томск

УДК 519.5 ББК 22.126 Ч49

Рецензенты:

Доктор технических наук, профессор

Ходашинский И.А. (ТУСУР)

Доктор физико-математических наук, профессор

Сущенко С.П. (ТГУ)

Печатается по решению редакционного совета ТУСУР

**Замятин Н. В.**

Нечеткая логика и нейронные сети: учеб. пособие /

Н.В. Замятин ; Томский. гос. ун-т АСУ и радиоэл-ки .- Томск: Изд-во. Томск. гос. ун-та АСУ и радиоэл-ки, 2014. – 203 с. -

гос. ун-та, 2010. – 96 с. – ISBN 978-5-9984-0055-1.

Рассматриваются положения теории нечетких логик и нейронных сетей. Приводятся основные определения и понятия, описываются операции над нечеткими множествами и нейросетевые парадигмы. Приведены нечеткие отображения, нечеткие числа и основные математические операции над ними. Алгоритмы обучения нейронных сетей. Изложены основные нечеткой логики и нечетких высказываний, рассмотрены алгоритмы обработки нечеткого вывода и нейронных сетей, а также применение нечеткой логики и нейронных сетей.

Предназначено для студентов, обучающихся по направлению «Бизнес-информатика», а также может быть полезно студентам других направлений при изучении дисциплин информационного профиля.

## ОГЛАВЛЕНИЕ

<b>Предисловие.....</b>	<b>5</b>
<b>Введение.....</b>	<b>7</b>
<b>1. Глава 1. Основы теории нечетких множеств.....</b>	<b>18</b>
1.1. Четкие и нечеткие множества.....	18
1.2. Понятие нечеткой переменной.....	24
1.3. Функции принадлежности.....	44
1.4. Нечеткая арифметика.....	58
1.5. Основные операции над нечеткими множествами.....	71
1.6. Нечеткие отношения и их свойства.....	87
1.7. Нечеткая импликация.....	93
<b>2. Глава 2. Нечеткий вывод и нечеткие модели.....</b>	<b>100</b>
2.1 Структура, основные элементы и операции в нечетких моделях.....	100
2.2. Фаззификация. Дефаззификация результирующей функции принадлежности.....	120
2.3 Свойства правил и нечетких моделей.....	130
2.4. Базы правил.....	142
2.5. Нормирование (масштабирование) входов и выходов нечеткой модели...150	
2.6. Нечеткая модель Мамдани.....	157
2.7. Нечеткая модели Такати-Сугено.....	165
2.8. Области применения нечеткой логики.....	172
<b>3. Глава 3. Нейронные сети.....</b>	<b>175</b>
3.1. Введение в нейронные сети.....	175
3.2. Биологический нейрон.....	183
3.3. Искусственный (математический) нейрон.....	189
3.4. Активационные функции нейронных элементов.....	200
3.5. Алгоритмы обучение нейронных сетей.....	206
3.6. Нейронные сети прямого распространения.....	226

	4
3.7. Сети Хемминга.....	238
3.8. Сети Хопфилда.....	242
3.9. Сеть Кохонена.....	249
3.10. АРТ-сети.....	253
3.11. Применение нейронных сетей.....	264
Заключение.....	290
Библиографический список.....	291

## ПРЕДИСЛОВИЕ

Понятие intelligence означает «рассуждать разумно». Искусственный интеллект (ИИ) занимается моделированием разумного поведения в искусственно созданных системах. Термин «искусственный интеллект» (artificial intelligence) был предложен в 1956 году. Единого определения ИИ на сегодняшний день так и не существует. В качестве рабочего определения можно предложить следующее.

Искусственный интеллект – это одно из направлений информатики, целью которого является разработка программно-аппаратных средств, позволяющих решать интеллектуальные задачи, общаясь с ЭВМ на ограниченном подмножестве естественного языка. При воссоздании разумных рассуждений и действий возникают определенные трудности. Во-первых, в большинстве случаев, выполняя какие-то действия, человек не осознает, как это делает, не известен точный способ, метод или алгоритм понимания текста, распознавания лиц, доказательства теорем, решения задач, сочинения стихов и т.д.

Во-вторых, на современном уровне развития компьютер слишком далек от человеческого уровня понимания и работает по другим принципам. Искусственный интеллект является наукой, искусством, техникой и психологией. Методы искусственного интеллекта разнообразны. Они активно заимствуются из других наук, адаптируются и изменяются под решаемую задачу. Для создания интеллектуальной системы необходимо привлекать специалистов из прикладной области, поэтому в рамках искусственного интеллекта сотрудничают лингвисты, нейрофизиологи, психологи, экономисты, информатики, программисты и т.д. Идея создания искусственного подобия человека для решения сложных задач и моделирования человеческого разума существовала еще в древнейшие времена.

Оживающая механическая статуя бога Амона создана в Древнем Египте, Бог Гефест ковал человекоподобных существ.

Испанский философ, математик и поэт Раймонд Луллий (родоначальник искусственного интеллекта) в XIII веке создавал механическую машину для решения задач на основе разработанной им всеобщей классификации понятий. В XVIII веке Лейбниц и Декарт независимо друг от друга предложили универсальные языки классификации всех наук. Это были первые теоретические работы в области искусственного интеллекта.

В 1956 году. Д. Маккарти, М. Минский, К. Шеннон и Н. Рочестер организовали семинар по теории автоматов, нейронным сетям, интеллекту. На этом семинаре появились термин и отдельная наука – искусственный интеллект. Одним из основателей теории искусственного интеллекта считается английский ученый Алан Тьюринг, опубликовавший в 1950 году статью «Вычислительные машины и разум», в которой описывался «тест Тьюринга», для оценки способности компьютера к осмысленному диалогу с человеком.

Хотя искусственный интеллект преследует множество целей, одной из основных задач является моделирование интеллекта человека и машины и вычисление их общих принципов. К интеллектуальным задачам относятся задачи, алгоритм нахождения которых неизвестен.

Поэтому труднорешаемой (нерешаемой) называют такую задачу, для которой не существует эффективного алгоритма решения. Решением подобных задач и занимается искусственный интеллект. Исследователи изучают процессы мышления и разумное поведение для того, чтобы найти методы решения подобных задач, подобно как человек в своей деятельности сталкивается с ними достаточно часто и успешно решает. В конце 50-х годов появились модель лабиринтного поиска и теория распознавания образов, в начале 60 годов эвристическое программирование, Эвристики позволяют сокращать количество рассматриваемых вариантов. В

середине 60-х годов к решению задач стали активно подключать методы математической логики и экспертные системы. Такие системы применимы к слабоформализуемым задачам, обладающих следующими особенностями:

- ошибочностью, неоднозначностью, неполнотой и противоречивостью исходных данных;
- ошибочностью, неоднозначностью, неполнотой и противоречивостью знаний о проблемной области и решаемой задаче;
- большой размерностью пространства решения (то есть перебор при поиске решения весьма велик);
- динамически изменяющимися данными и знаниями.

## **ВВЕДЕНИЕ**

### **История развития нечеткой логики и нейронных сетей**

Нарастание объемов информации и вычислительных мощностей заставляет приспособлять наши способности к познанию окружающего мира. Классические математические методы заложили основы современных вычислительных алгоритмов распознавания образов и речи, для управления системами различных классов и решения других подобных задач. Процедуры искусственного интеллекта обусловили создание экспертных и прогностических систем, основанных на символьной обработке и использующих базы правил. Эти подходы используют узкоспециализированные вычислительные процедуры, либо базы знаний, ограниченные двоичной логикой.

Поэтому нужны технологии, позволяющие получать непрерывные или дискретные решения в результате обучения по доступным данным. Это эволюционные методы такие, как нечеткая логика, нейронные сети, использующие стохастические алгоритмы для обучения модели с учителем или путем самоорганизации, генетические алгоритмы.

Нечеткая логика, предложенная американским математиком Лотфи Заде 35 лет назад [14], позволяет описывать качественные, неточные знания об окружающем мире, а также применять эти знания для получения новой информации. Нечеткое моделирование является перспективным направлением в области управления и принятия решений, и необходима при наличии неопределенности, затрудняющей применение точных количественных методов.

Диапазон применения нечетких методов с каждым годом расширяется, охватывая такие области, как проектирование промышленных роботов и бытовых электроприборов, управление доменными печами и движением поездов метро, автоматическое распознавание речи и изображений. Нечеткая логика подобна и ближе к человеческому мышлению и рассуждениям, чем традиционные формально-логические системы, и может описывать и представлять сложные трудноформализуемые процессы.

Первые реализации нечетких моделей относятся к середине 1970-х гг: в Великобритании Эбрахим Мамдани использовал нечеткую логику для управления парогенератором. Предложенный алгоритм, основанный на нечетком логическом выводе, позволил избежать большого объема вычислений.

В начале 1980-х гг. нечеткая логика и теория нечетких множеств получили свое дальнейшее развитие в целом ряде программных средств поддержки принятия решений и в экспертных системах анализа данных.

Причины быстрого развития нечеткой логики

- нечеткая логика поддерживает разработку быстрого прототипа с последующим усложнением его функциональности,
- нечеткая логическая модель более проста для понимания, чем математическая модель на основе дифференциальных или разностных уравнений.

- нечеткие модели оказываются более простыми для своей аппаратной реализации, и более того могут реализовываться параллельно

Нечеткие технологии применялись в различных технических устройствах и бытовых приборах, на фондовых рынках используются трейдерские системы, основанных на нечеткой логике, превосходящие традиционные информационные системы по скорости и динамике.

Нечеткая логика применяется для управления внутренними компонентами персональных компьютеров, компрессией речи и видео, устранение влияния отраженных ТВ-сигналов и радиосигналов, сетевой маршрутизацией и распознаванием речи.

Первые нейросети появились в конце 50-х годов. В 1957 году Ф. Розенблаттом создан систему, моделирующую человеческий глаз и его взаимодействие с мозгом, – персептрон. Объединение нейронных сетей и нечеткой логики это следующий шаг в обработке информации. Возможности нейронных сетей могут усиливаться за счет применения нечетких множеств и нечетком выводе для оценивания функции принадлежности элементов к множествам. Нечеткие операторы позволяют представить сложные системы с помощью переменных, значения которых определяются на интуитивном уровне. Другой подход состоит в формировании правил вывода непосредственно в процессе обучения. Этим определяется взаимопроникновение и дружелюбность нейронных моделей и систем, основанных на нечеткой логике.

В 1963 году Д. Робинсон реализовал метод автоматического доказательства теорем, получивший название «принцип резолюции», и на основе этого метода в 1973 году был создан язык логического программирования Prolog. В США появились первые коммерческие системы, основанные на знаниях, – экспертные системы. Происходит коммерциализация искусственного интеллекта. Коммерческая система R1 компании DEC позволила сэкономить примерно 40 миллионов долларов за год. К 1988 году компанией DEC было

развернуто 40 экспертных систем. В компании Du Pont применялось 100 систем, и экономия составляла примерно 10 миллионов в год. В 1981 году Япония, в рамках 10-летнего плана по разработке интеллектуальных компьютеров на базе Prolog, приступила к разработке компьютера 5-го поколения, основанного на знаниях. В 1997 году компьютер «Дип Блю» победил в игре в шахматы человека, доказав возможность того, что искусственный интеллект может сравняться с человеком или превзойти его в ряде интеллектуальных задач (пусть и в ограниченных условиях).

В 1954-1964 гг. создаются отдельные программы и проводятся исследования в области поиска решения логических задач. Создается программа, автоматически доказывающая теоремы. Она основана на оригинальном обратном выводе Маслова, аналогичном методу резолюций Робинсона.

Отечественными учеными в 60-е годы, разработан алгоритм «Кора», моделирующий деятельность человеческого мозга при распознавании образов. В 1964 году предлагается метод автоматического поиска доказательства теорем в исчислении предикатов, получивший название «обратный метод Маслова». В 1965-1980 гг. профессор Д. А. Поспелов создал направление по ситуационному управлению (представлению знаний). Появилась необходимость в языках представления знаний и управления ими. Такими языками стали нечеткая логика, генетические алгоритмы, эвристики и нейронные сети.

Аппарат теории нечетких множеств, несмотря на широкие возможности применения, оказался все-же сложным для воплощения. Учитывая имеющийся уровень технологии, нечеткая логика заняла свое место среди других специальных научных дисциплин - где-то посередине между экспертными системами и нейронными сетями.

Второе рождение нечеткая логика пережила в начале восьмидесятых годов, когда нечеткие управляющие алгоритмы использовались для создания электронных систем различного применения

Третий период начался с конца 80-х годов, и характеризуется практическим применением нечеткой логики в различных сферах науки и техники. Нечеткая логика отталкивается от классической логики, которая понимается как раздел математики, посвященный изучению и построению математических доказательств. Основным достоинством и причиной широкого распространения математической логики является использование аксиоматического (бездоказательного) метода в построении различных математических теорий.

Простейшим видом математической логики является логика высказываний, объектами изучения которой служат высказывания, их взаимосвязи и семантика.

Основным понятием классической логики высказываний является понятие высказывания, означающее, что любое утверждение представляется истинным или ложным. Простое число 77 - это истина. В классической логике только два понятия истина или ложь, что естественно для четкой математики, где все величины точные. Однако существуют такие выражения, как “скорость большая”. Поэтому необходимо пояснить, что означает значение “большая”. В четкой логике это сделать затруднительно, поэтому появился такой раздел математики, как нечеткая логика, которая близка к дискретной математике.

Для обозначения истинности и ложности используется множество  $A=\{0,1\}$ , где 1 – истина, а 0 – ложь. Для составления сложных высказываний используют простые высказывания соединяя их знаками логических операций И, ИЛИ, НЕ, ИЛИ-НЕ, И-НЕ, “если...то”, “тогда и только , тогда”.

Значения истинности сложных высказываний полностью определяются совокупностью истинностей простых высказываний, что позволяет возможность конструировать сложнее высказывания на основании значений простых высказываний.

Для обращения с неточно известными величинами обычно применяется теория вероятностей. Однако случайность связана с неопределенностью, касающейся принадлежности некоторого объекта к обычному множеству. Это различие между нечеткостью и случайностью приводит к тому, что математические методы нечетких множеств совершенно не похожи на методы теории вероятностей. Они во многих отношениях проще вследствие того, что понятию вероятностной меры в теории вероятностей соответствует более простое понятие функции принадлежности в теории нечетких множеств.

По этой причине даже в тех случаях, когда неопределенность в процессе принятия решений может быть представлена вероятностной моделью, обычно удобнее оперировать с ней методами теории нечетких множеств без привлечения аппарата теории вероятностей.

Подход на основе теории нечетких множеств является альтернативой количественным методам анализа систем. Он имеет три основные отличительные черты:

1. Вместо или в дополнение к числовым переменным используются нечеткие величины и "лингвистические" переменные;
2. Простые отношения между переменными описываются с помощью нечетких высказываний;
3. Сложные отношения описываются нечеткими алгоритмами.

Такой подход дает приближенные, но эффективные способы описания поведения систем, очень сложных и плохо определенных, что они не поддаются точному математическому анализу. До работ Л. Заде подобная качественная

информация, по существу, просто терялась, потому что было непонятно, как ее использовать.

При обработке информации необходимо учитывать такие виды неопределенности:

1. Низкая точность оперативной информации, получаемой с объектов управления, возникающая ввиду большой погрешности источников данных, их невысокой надежности, отказов каналов связи, большого запаздывания при передаче информации, отсутствия возможности замеров параметров во всех точках процесса, необходимых для моделей.

2. Неточность моделей объектов управления, вызванная неэквивалентностью решений системных многоуровневых иерархических моделей и используемых на практике отдельных локальных задач. Неточность моделей может возникать из-за неправильной декомпозиции общей задачи, излишней идеализации модели сложного процесса, разрыва существенных связей в исследуемых системах, линеаризации, дискретизации, замены фактических характеристик паспортными данными.

3. Нечеткость в процессе принятия решений в многоуровневых иерархических системах, обусловленная тем, что наличие четких (точных) целей и решений, затрудняет процесс и предопределяет длительный итеративный характер согласования решений.

4. Наличие человека в контуре управления и ведение процесса на естественном языке, приводит к необходимости представления знаний человека в виде алгоритмов и согласованности полученного ЭВМ решения с его оценкой

5. Ненадежность исходной информации, получаемой от человека в режиме принятия решения, неточность оценок, недоопределенность понятий и терминов, неуверенность человека в своих заключениях;

6. Нечеткость (неоднозначность) естественного языка (лингвистическая неопределенность и языка представления правил в системах экспертного типа;

Когда человек сталкивается с неопределенностью реальной системы в процессе принятия решений, то он поступает различными способами:

1. Чаще всего сознательно (или бессознательно) игнорирует существование неопределенности и использует детерминированные модели;
2. Выбирает один наиболее существенный, с его точки зрения, вид неопределенности и использует соответствующую теорию, так как разработанные в настоящее время количественные методы принятия решений помогают выбрать наилучшие из множества возможных решений лишь в условиях конкретного вида неопределенности;
3. Проводит дополнительные исследования системы или получает информацию в ходе контроля (адаптация и обучение) или управления.



Рис. 1. Классификация неопределенностей

Согласно работе М. Блэка [1], неопределенность имеет место, когда универсальное множество состоит более чем из одной точки. Если для этих элементов множества заданы соответствующие вероятности или другие вероятностные характеристики, то имеет место вероятностная неопределенность.

Если известны только граничные элементы множества - интервальная неопределенность.

При задании для каждого элемента множества соответствующей степени принадлежности - нечеткость.

Неопределенность можно проклассифицировать:

- по степени неопределенности (полная определенность, вероятностная, лингвистическая, интервальная, полная неопределенность),
- по характеру неопределенности (параметрическая, структурная, ситуационная)
- по использованию получаемой в ходе управления информации (устраняемая и неустраняемая)

Для широкого класса задач априорная неопределенность может быть сведена к параметрической, когда вероятностные законы распределения для исследуемых ситуаций, величин и наблюдаемых процессов известны с точностью до конечного числа параметров.

Термин “нечеткость” стал уже привычным, тем более, что происходит взаимопроникновение нечеткой логики, нейронных сетей, эволюционных алгоритмов друг в друга и реализация на этой основе новых приложений в информационных технологиях.

С точки зрения интеллекта мозг человека состоит из двух полушарий - левого и правого. Если правое полушарие предназначено для восприятия образов в целом, то левое для анализа этих образов. Поэтому можно привести такое грубое сравнение, левое полушарие отвечает за точный анализ объектов

(точная математика), то правое за образный анализ (нечеткая математика или нечеткая логика). Новые идеи, изобретения, мысли рождаются в правом полушарии, а объективизация, логическое обоснование, уточнение – это деятельность левого полушария.

Знания можно разделить на макрознания и микрознания. Микрознания (специальные) хорошо упорядочены, ими легко пользоваться, потому что логически осмыслены. Но прежде, чем ими пользоваться, нужно руководствоваться здравым смыслом, который находится в правом полушарии. Эти макрознания трудно систематизировать, интерпретировать и конкретизировать. Вместе с тем, человек, обычно легко управляется с этими знаниями, иногда это выглядит, как интуиция. При решении сложных трудноформализуемых задач человек опирается на знания, и отбрасывая множество решений, выбирает наиболее правильное.

Поэтому искусственный интеллект, так и остался искусственным, потому что компьютер не обладает такими способностями, хотя способы формирования знаний уже используются. Иначе говоря, создаются человеко-машинные системы, в которых роль правого полушария отводится человеку, а хранения упорядочивание, хранения, анализ отводится информационной системе. В человеческом мозге большую роль играет связь между левым и правым полушариями [ ]. В искусственном интеллекте такую роль должны играть язык, базы знаний и метод обработки знаний.

Для организации этого субъективизма необходимо вводить нечеткость и соответствующие процедуры обработки этих нечеткостей.

На основе такого подхода появились новые возможности исследования:

1. Возможность создания искусственного интеллекта, в некоторых случаях превосходящих возможности человека.
2. Создание компьютеров, программируемых на естественном языке

3. Использование данных любой степени гранулированности, что позволяет значительно сократить объемы данных и повысить скорость управления.

4. Возможность подстройки уровня гранулированности информации по требуемую точность управления и моделирования.

За прошедшее время развилось много теорий и направлений, но существуют базовые понятия, такие как: четкая, нечеткая и лингвистические переменные, четкие и нечеткие множества, четкая и нечеткая логики.

Традиционные методы математики предназначены обычно для обработки точно заданных данных.

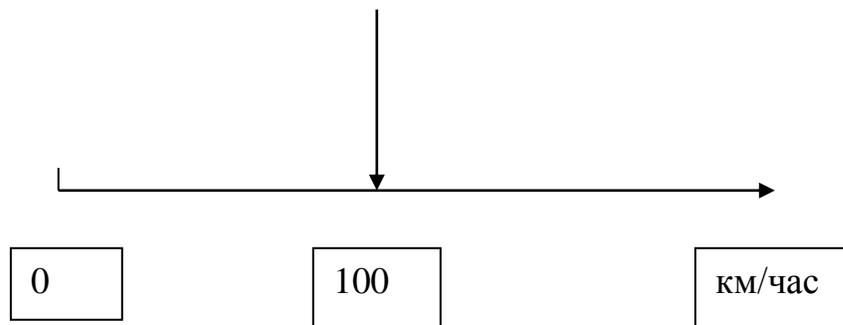


Рис.2. Представление точных измерений скорости

Человек же оперирует понятиями приближенными, так как не может точно измерить параметр, да и не требуется в повседневной жизни хранить большое количество информации. С помощью определенных характеристических функций можно определять соответствие значений скорости неким качественным параметрам.

Заде (1976 г.) ввел понятие информационной гранулы, т.е. качественного параметра. В данном примере имеется три гранулы, Если точности недостаточно, то можно ввести дополнительные гранулы: достаточно высокая скорость, очень высока, очень низкая. Степень гранулированности данных будет определяться потребностями пользователей, или контекстом задачи.

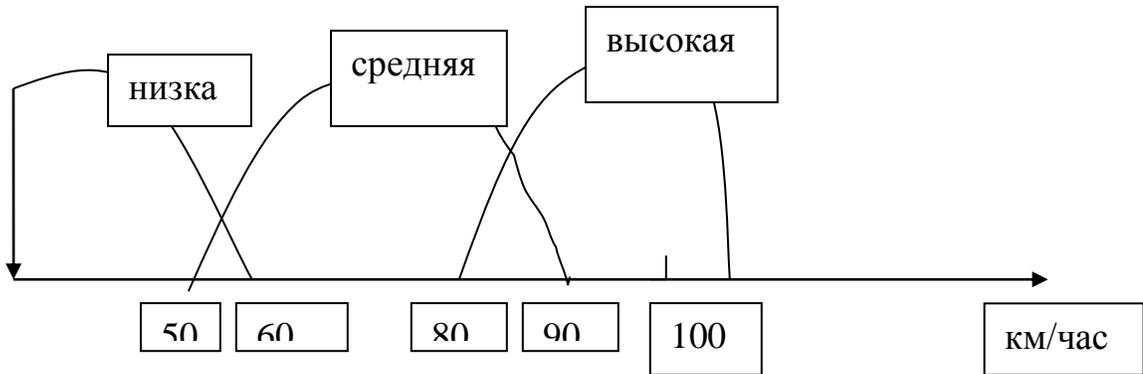


Рис.3. Визуальное представление приближенной оценки скорости

Данные, получаемые от измерительного устройства более точны, чем от человека, поэтому необходимо использовать понятие ширины гранулы. Следовательно, параметр средняя скорость может иметь различные значения ширины гранул и в конечном итоге может свестись к точному значению или точке на оси скорости. Такие переменные называются нечеткими множествами. Английское слово *fuzz*, от которого образовано прилагательное *fuzzy* обозначает ворс, специальный термин, определяющий свойство тканей. Если посмотреть на такую ткань, то она кажется нечеткой и размытой. Поэтому термин *fuzzy* логично применить к понятиям нечеткой логики.

## Глава 1. Основы теории нечетких множеств

### 1.1. Четкие и нечеткие множества

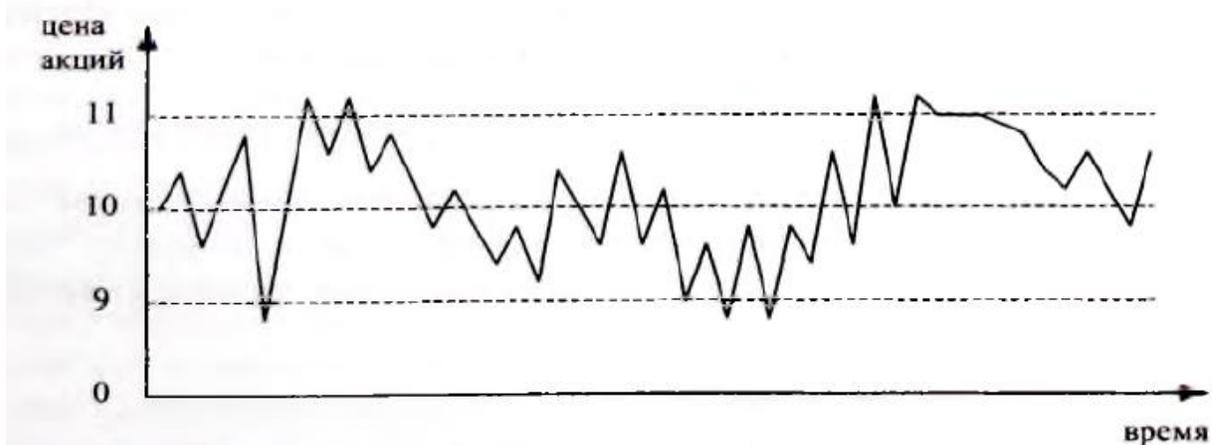
В окружающем мире некоторые величины сложно оценить с помощью измерительных устройств: красота, порядок, опасность, успех и т. п. Человек прибегает к собственным «измерительным устройствам», позволяющих ему оценивать подобные величины и ситуации, преобразовывать их, распознавать (идентифицировать), и затем управлять ими.

Качественно оценивая действительность, люди выработали логические и интеллектуальные способности, которыми вычислительные системы не обладают.

Нечеткие, качественные оценки позволяют значительно расширить методы математического моделирования, требующие точной информации о входных величинах, и которые можно рассматривать как дополнение к точным измерениям для создания полных моделей действительности.

Оценка параметров системы с использованием лингвистических значений, основанных на восприятиях человека, не требует измерительных устройств. Вместе с тем, если использовать нечеткие числа, то подобные устройства все же необходимы. Посредством нечетких чисел обобщают большие объемы точных данных, являющихся результатами измерений или обращений к базам данных, например информацию о цене на акции некоторой компании (рис. 1.1). Данные, представленные в точной (четкой) форме, можно обобщенно представить в виде нечеткого числа:

- «приблизительно в пределах между 9 и 11» или
- «около 10».



**Рис. 1.1.** Пример большого объема данных о точном значении параметра

### **Понятие четкой переменной**

Понятие четких (определенных) множеств известно из классической логики. Рассмотрим множество  $X$  всех чисел от 0 до 10, которое назовем универсумом рассуждения. Подмножество  $A$  множества  $X$  всех действительных чисел от 5 до 8 будет  $A = [5,8]$

Характеристическая функция множества  $A$  ставит в соответствие число 1 или 0 каждому элементу в  $X$ , в зависимости от того принадлежит данный элемент подмножеству  $A$  или нет (Рис 1.2.).

Элементы, которым поставлено соответствие 1, есть элементы, находящиеся во множестве  $A$ , а элементы, которым поставлен в соответствие 0, элементы, не находящиеся во множестве  $A$ .

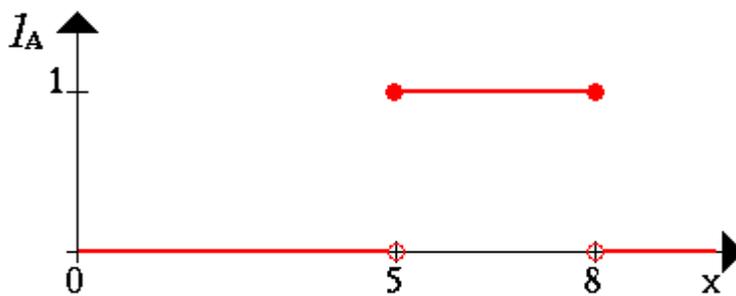


Рис. 1.2. Характеристическая функция множества  $A$

**Характеристической функцией** множества  $A$  называют функцию  $\mu_A(x)$ , заданную на универсальном множестве  $U$  и принимающую значение единица на тех элементах множества  $U$ , которые принадлежат  $A$ , и значение нуль на элементах, которые не принадлежат  $A$ :

$$\mu_A(x) = \begin{cases} 0, & \text{если } x \notin A \\ 1, & \text{если } x \in A, (x \in U) \end{cases} \quad (1.1)$$

Обычная четкая переменная характеризуется тройкой  $(X, U, R(X; u))$ , где  $X$  - название переменной,  $U$  - универсальное множество (конечное или

бесконечное),  $u$  - общее название элементов множества  $U$ ,  $R(X;u)$  - подмножество множества  $U$ , представляющее собой ограничение на значения элементов  $u$ , обусловленное названием  $X$ .

Вместо  $R(X;u)$  обычно  $R(X)$ ,  $R(u)$  или  $R(x)$ , где  $x$  обозначает общее название значений переменной  $X$ , а  $R(X)$  это ограничение на  $u$  или ограничение, обусловленное переменной  $X$ .

Переменной соответствует уравнение назначения

$$x = u : R(X), \quad (1.2)$$

$$\text{Или } x = u, \quad u \in R(X). \quad (1.2)$$

Переменной  $x$  назначено значение  $u$  с учетом ограничения  $R(X)$ . Следовательно, уравнение назначения удовлетворяется тогда и только тогда, когда  $u \in R(X)$ .

Пусть переменную возраст, в качестве  $U$  можно взять множество целых чисел  $0, 1, 2, 3, \dots$ , а  $R(X)$  может быть подмножеством  $0, 1, 2, \dots, 100$ .

Многомерные переменные  $X_1, \dots, X_n$  - связаны с универсальными множествами  $U_1, \dots, U_n$ , а упорядоченный набор  $X = (X_1, \dots, X_n)$  называются составной переменной. Универсальным множеством для  $X$  является декартово произведение  $U = U_1 \times U_2 \times \dots \times U_n$ , а ограничением  $R(X_1, \dots, X_n)$  будет отношение в  $U_1 \times \dots \times U_n$ . Это отношение определяется характеристической функцией

$$\mu_R : U_1 \times \dots \times U_n \rightarrow \{0, 1\}, \text{ причем}$$

$$\mu_R(u_1, \dots, u_n) = \begin{cases} 1, & \text{если } (u_1, \dots, u_n) \in R(X_1, \dots, X_n), \\ 0, & \text{если } (u_1, \dots, u_n) \notin R(X_1, \dots, X_n), \end{cases} \quad (1.3)$$

а  $u_i$  - общее название элементов множества  $U_i, i=1, \dots, n$ . Соответственно этому  $n$ -арное (многомерное) уравнение назначения имеет вид

$$(x_1, \dots, x_n) = (u_1, \dots, u_n) : R(X_1, \dots, X_n), \quad (1.4)$$

которое следует понимать как  $x_i = u_i, i=1, \dots, n$ ,

при ограничении  $(u_1, \dots, u_n) \in R(X_1, \dots, X_n)$ , где  $x_i, i=1, \dots, n$  - общее название значений переменной  $X_i$ .

Четкую переменную можно представить дорожной сумкой с ярлыком и с твердыми стенками.  $X$  соответствует названию на ярлыке,  $U$  - списку предметов, которые в принципе умещаются в сумку, а  $R(X)$  - части списка, в котором перечислены предметы, обычно помещаемые в сумку. (Например, такой предмет, как лодка, не может быть в списке  $U$ , такой предмет, как компьютер, может быть в  $U$ , но не быть в  $R(X)$ , а такой предмет, как пачка сигарет или пара ботинок, может быть в  $R(X)$ .)

В таком понимании уравнение назначения  $x = u : R(X)$  соответствует тому, что предмет  $u$ , удовлетворяющий ограничению  $R(X)$  (т. е. находящийся в списке предметов), помещен в сумку  $X$  (рис. 2.1).

Множественная переменная  $X = (X_1, \dots, X_n)$  соответствует сумке с ярлыком  $X$ , имеющему  $n$  отделений, обозначенных  $X_1, \dots, X_n$ , причем перегородки между

этими отделениями можно устанавливать по своему усмотрению.

Ограничение  $R(X_1, \dots, X_n)$  соответствует списку из  $n$ -наборов предметов  $(u_1, \dots, u_n)$ , таких, что  $u_1$  можно поместить в отделение  $X_1$ ,  $u_2$  - в отделение  $X_2$ , ... и  $u_n$  - в отделение  $X_n$  одновременно (рис. 1.3).

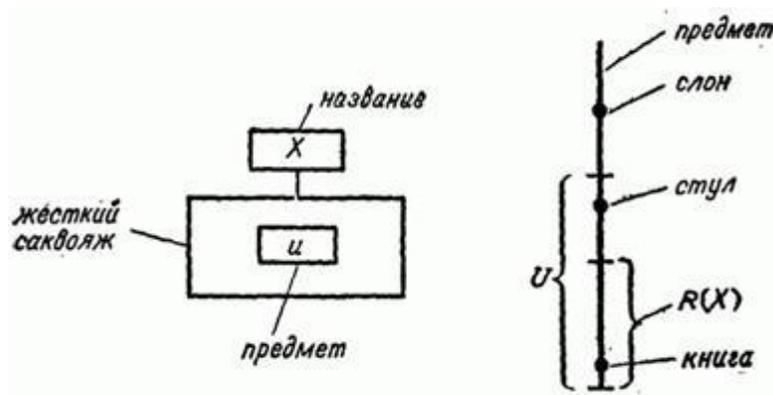


Рис. 1.3. Аналогия с дорожной сумкой для унарной четкой переменной.

Если  $n = 2$ , то при заданном положении перегородок, можно поместить пальто в отделение  $X_1$ , а костюм - в отделение  $X_2$ , в то время как при другом положении перегородок можно поместить пальто в отделение  $X_2$ , а коробку с туфлями - в отделение  $X_1$ .

В этом случае обе пары (пальто, костюм) и (туфли, пальто) окажутся в списке  $R(X_1, X_2)$  предметов, которые умещаются в сумку  $X$ .

Уравнение назначения

$$(x_1, \dots, x_n) = (u_1, \dots, u_n) : R(X_1, \dots, X_n)$$

соответствует помещению предметов  $u_1$  в  $X_1$ , ... ,  $u_n$  в  $X_n$  одновременно при условии, что  $n$ -набор предметов  $(u_1, \dots, u_n)$  имеется в списке  $R(X_1, \dots, X_n)$ .

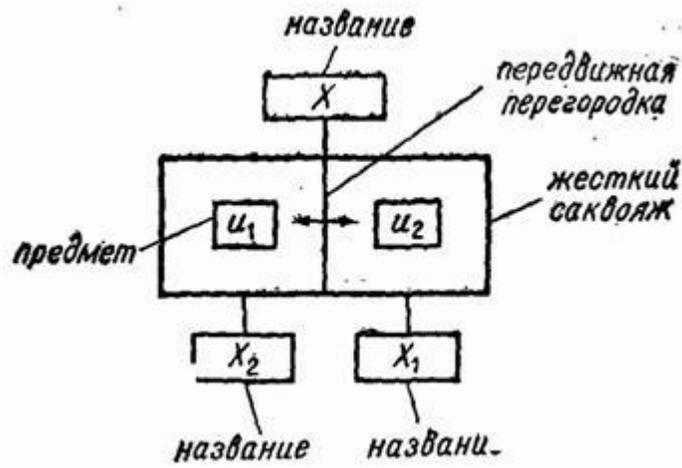


Рис. 1.4. Дорожная сумка для бинарной четкой переменной.

На рис. 1.5 показаны ограничения на  $u_1$  и  $u_2$ , вызванные ограничением  $R(X_1, X_2)$ .

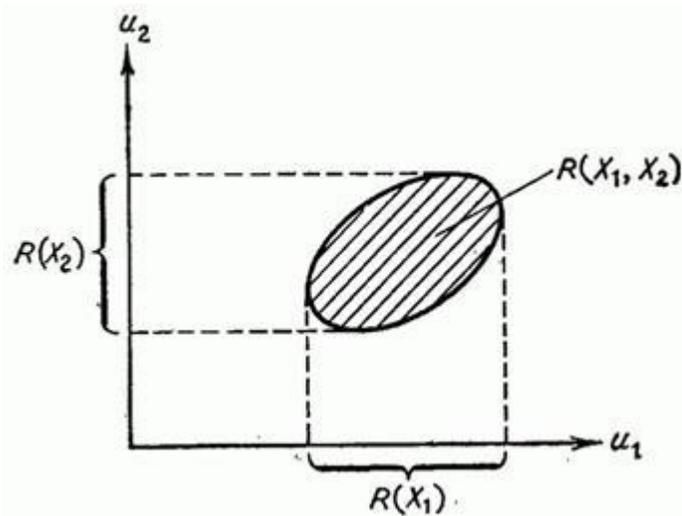


Рис. 1.5. Ограничения  $R(X_1, X_2)$ .

## 1.2. Понятие нечеткой переменной

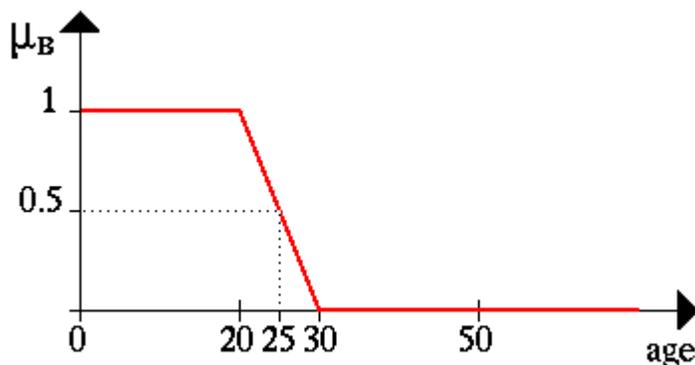
Окружающий мир нечеток. Поэтому качественные оценки человек использует даже при имеющихся средствах точного измерения. Водитель чаще всего говорит, хотя имеет спидометр:

- «Я ехал очень быстро»,
- «Я ехал со скоростью примерно 100 км/ч»,
- «Я ехал со скоростью более 100 км/ч».

Точное значение скорости, водитель в каждый момент времени, водитель не может помнить из-за ограниченных возможностей человеческой памяти, а во-вторых, достаточно сделать грубую оценку, избавляющую от больших объемов ненужной информации

.Множество молодых людей представим как  $V = \{\text{множество молодых людей}\}$ . Возраст начинается с 0, и нижний предел этого множества должен быть нулем. Верхний предел установим равным 20 годам, т.е.  $V = [0,20]$ . Для устранения четкого разделения на молодых и не молодых, нужно вводить гибкие формулировки “ДА, он|она принадлежит к достаточно молодым людям или Нет, он|она не очень молод|молода”. Иначе кто-то в двадцатилетнем возрасте молодой, а на следующий день уже не молодой.

Тогда характеристическая функция множества молодых людей,



То есть 25-летние все еще *молоды* со степенью **50 процентов**.

Рассмотрим в качестве примера универсальное множество  $U = \{1, 2, 3, \dots, 10\}$  и два его подмножества:  $A$  – множество чисел, меньших 7, и  $B$  – множество четных чисел. Характеристические функции множеств  $A$  и  $B$  имеют вид

$$\mu_A(x) = \begin{cases} 1, & \text{если } x < 7 \\ 0, & \text{если } x \geq 7 \end{cases}, \quad \mu_B(x) = \begin{cases} 1, & \text{если } x = 2, 4, 6, 8, 10 \\ 0, & \text{если } x = 1, 3, 5, 7, 9 \end{cases}. \quad (1.6)$$

Представим характеристические функции  $\mu_A$  и  $\mu_B$  в виде таблицы:

$x$ ( $x \in U$ )	1	2	3	4	5	6	7	8	9	10
$\mu_A(x)$	1	1	1	1	1	1	0	0	0	0
$\mu_B(x)$	0	1	0	1	0	1	0	1	0	1

При четком представлении все элементы универсума рассуждения кодируются с помощью 0 или 1. Для приведения к нечеткости, нужно ввести значения между 0 и 1. Можно допустить бесконечное число значений между 0 и 1, называемое единичным интервалом  $I = [0, 1]$ .

Нечеткая переменная характеризуется тройкой  $(X, U, R(X; u))$ , где  $X$  — название переменной,  $U$  — универсальное множество (конечное или бесконечное),  $u$  — общее название элементов множества  $U$ ,  $R(X, u)$  — нечеткое подмножество множества  $U$ , представляющее собой нечеткое ограничение на значения переменной  $u$ , обусловленное  $X$ .

Уравнение назначения для  $X$  имеет вид

$$x = u : R(X)$$

и отражает то, что элементу  $x$  назначается значение  $u$  с учетом ограничения  $R(X)$ .

По Л. Заде степень, с которой удовлетворяется это равенство, называют принадлежностью значения  $u$  с  $R(X)$  и обозначают через  $c(u)$ . По определению

$$c(u) = \mu_{R(X)}(u), \quad u \in U, \quad (1.7)$$

где  $\mu_{R(X)}(u)$  — степень принадлежности  $u$  ограничению  $R(X)$ .

Принадлежность  $u$  с  $R(X)$  — это мера того, насколько значение  $u$  удовлетворяет ограничению  $R(X)$ , и она не имеет никакого отношения к тому, насколько вероятно или невероятно это значение.

Нечеткую переменную также можно представить дорожной сумкой с ярлыком, имеющей уже мягкие стенки. Тогда  $X$  — надпись на ярлыке (название сумки),  $U$  — список предметов, которые можно поместить в сумка, а  $R(X)$  — часть этого списка, в которой для каждого предмета  $u$  указано число  $c(u)$ , характеризующее степень легкости, с которой предмет  $u$  можно поместить в сумку  $X$  (рис. 1.1.6).

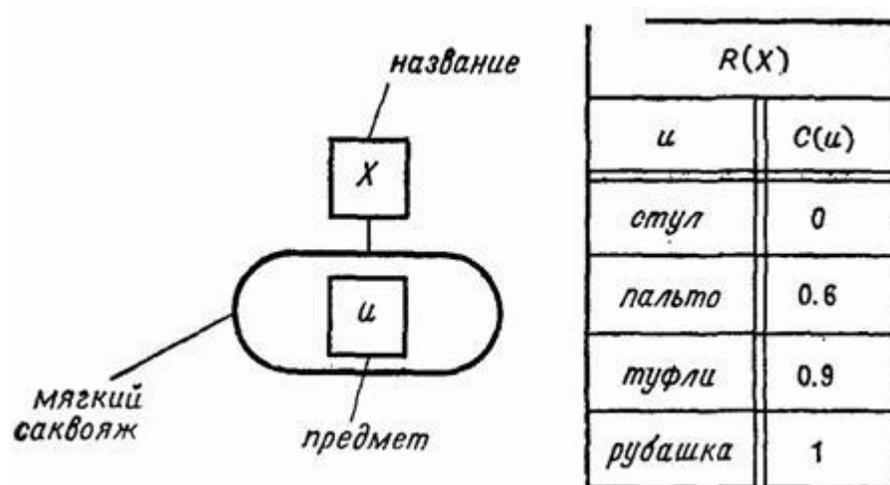


Рис. 1.1.6. Аналогия с сумкой для унарной нечеткой переменной

Если  $X_1, \dots, X_n$  — многомерные нечеткие переменные в  $U_1, \dots, U_n$  соответственно, то  $X \triangleq (X_1, \dots, X_n)$  есть  $n$ -арная составная переменная в  $U = U_1 \times \dots \times U_n$ .

Соответственно этому в  $n$ -арном уравнении назначения вида

$$(x_1, \dots, x_n) = (u_1, \dots, u_n) : R(X_1, \dots, X_n) \quad (1.8)$$

$x_i, i = 1, \dots, n$ , — общее название значений переменной  $X_i$ ,  $u_i$  — общее название элементов множества  $U_i$ ;  $R(X) \triangleq R(X_1, \dots, X_n)$  —  $n$ -арное нечеткое отношение в  $U$ , представляющее ограничение, обусловленное переменной  $X \triangleq (X_1, \dots, X_n)$ .

Принадлежность  $(u_1, \dots, u_n)$  к  $R(X_1, \dots, X_n)$  определяется так:

$$c(u_1, \dots, u_n) = \mu_{R(X)}(u_1, \dots, u_n),$$

где  $\mu_{R(X)}$  — функция принадлежности ограничения на  $u \triangleq (u_1, \dots, u_n)$ .

В аналогии с дорожной сумкой составную нечеткую переменную можно уподобить мягкой сумкой  $X$ , имеющему  $n$  отделений  $X_1, \dots, X_n$ . Функция принадлежности  $c(u_1, \dots, u_n)$  характеризует степень легкости, с которой предметы  $u_1, \dots, u_n$  можно поместить в соответствующие отделения  $X_1, \dots, X_n$  одновременно (рис. 1.1.7.).

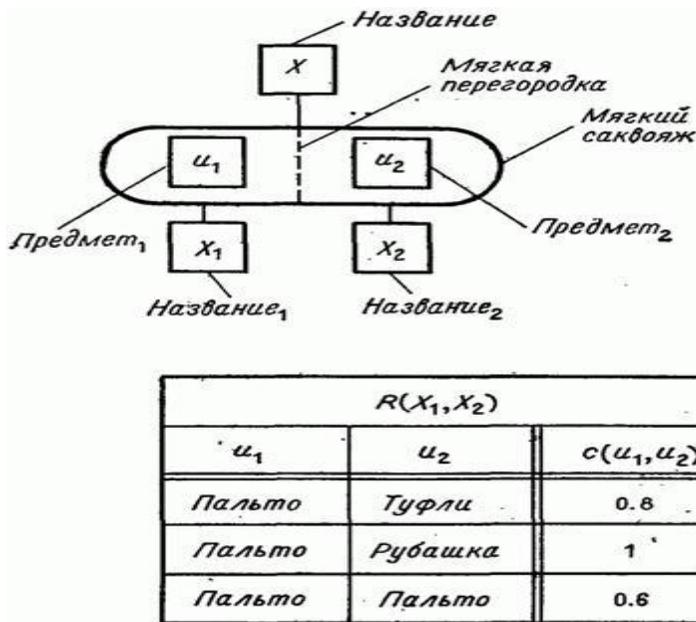


Рис. 1.1.7. Аналогия с сумкой для бинарной нечеткой переменной.

### Нечеткие множества

При помощи нечетких множеств можно формально определить неточные и многозначные понятия, такие как «высокая температура», «молодой человек», «средний рост» либо «большой город». Перед определением нечеткого множества необходимо задать область рассуждений. В случае неоднозначного понятия «много денег» большой будет признаваться одна сумма, если ограничиться диапазоном [0, 1000 руб] и совсем другая - в диапазоне [0, 1000000 руб]. Область рассуждений, называемая в дальнейшем пространством или множеством (универсумом), обозначается символом  $U$  и является четким множеством.

Нечеткое подмножество  $A$  универсального множества  $U$  определяется функцией принадлежности  $\mu_A: U \rightarrow [0, 1]$ , ставящей в соответствие каждому элементу  $u \in U$  число  $\mu_A(u)$  из интервала  $[0, 1]$ , определяющее **степень принадлежности** элемента  $u$  подмножеству  $A$ .

1. Носителем нечеткого множества  $A$  называется множество таких точек в  $U$ , для которых величина  $\mu_A(u)$  положительна.
2. Высотой нечеткого множества  $A$  называется величина  $\sup_U \mu_A(u)$ .
3. Нечеткое множество  $A$  **нормально**, если его высота равна единице

$$\sup_U \mu_A(u) = 1. \quad (1.9)$$

В противном случае нечеткое подмножество  $A$  субнормально. Нечеткое подмножество не малое и не большое универсального множества  $U = 1+2+\dots+10$ , имеющее вид:

$$\text{не малое и не большое} = 0.2/2 + 0.3/3 + 0.4/4 + 0.5/5 + 0.4/6 + 0.3/7 + 0.2/8$$

субнормально. Субнормальное нечеткое множество нормируют, поделив функцию  $\mu_A$  на величину  $\sup_U \mu_A(u)$ .

4. Точкой перехода нечеткого множества  $A$  называется такой элемент множества  $U$ , степень принадлежности которого множеству  $A$  равна 0.5. является  $u = 5.5$ .
5. Нечеткое подмножество универсального множества  $U$  может быть подмножеством другого нечеткого или обычного подмножества множества  $A$ . Более точно,  $A$  есть подмножество  $B$  или содержится в  $B$  тогда и только тогда, когда  $\mu_A(u) \leq \mu_B(u)$  для любого  $u \in U$ , т. е.

$$A \subset B \Leftrightarrow \mu_A(u) \leq \mu_B(u), \quad u \in U.$$

Если

$$\begin{aligned} U &= a+b+c+d, \\ A &= 0.5a+0.8b+0.3d, \\ B &= 0.7a+b+0.3c+d, \end{aligned}$$

то  $A \subset B$ .

Пусть универсальное множество  $U$  представляет собой интервал  $[0,100]$ , и переменная  $u$ , принимающая значения из этого интервала, интерпретируется как «возраст». Нечеткое подмножество универсального множества  $U$ , обозначаемое термином «старый», можно определить функцией принадлежности вида

$$\mu_A(u) = \begin{cases} 0 & \text{при } 0 \leq u \leq 50, \\ \left(1 + \left(\frac{u-50}{5}\right)^{-2}\right)^{-1} & \text{при } 50 \leq u \leq 100. \end{cases} \quad (1.10)$$

Носителем нечеткого множества **старый** является интервал  $[50,100]$ , высота множества **старый** близка к 1, а точкой перехода является значение  $u = 55$ .

Для представление нечетких множеств, используются следующие обозначения:

Обычное (не нечеткое) конечное множество

$$U = \{u_1, \dots, u_n\}$$

Записывается в виде

$$U = u_1 + \dots + u_n,$$

или

$$U = \sum_{i=1}^n u_i,$$

где знак  $+$  обозначает объединение, а не арифметическое суммирование. Эта запись рассматривается как представление множества  $U$  в виде объединения одноточечных множеств.

Нечеткое подмножество  $A$  универсального множества  $U$  записывается:

$$A = \mu_1 u_1 + \dots + \mu_n u_n,$$

или

$$A = \sum_{i=1}^n \mu_i u_i,$$

где  $\mu_i, i = 1, \dots, n$ , — степень принадлежности элемента  $u_i$  нечеткому множеству  $A$ . Для  $u_i$  — числа, может возникнуть двойное толкование записи  $\mu_i u_i$ , связанное с невозможностью различить компоненты  $\mu_i$  и  $u_i$ .

Поэтому разделяют значения  $\mu_i$  и  $u_i$  чертой:

$$A = \mu_1 / u_1 + \dots + \mu_n / u_n,$$

или

$$A = \sum_{i=1}^n \mu_i / u_i.$$

Пусть  $U = \{a, b, c, d\}$ , или в принятых обозначениях

$$U = a + b + c + d.$$

Нечеткое подмножество  $A$  универсального множества  $U$  записывается в виде

$$A = 0.3a + b + 0.9c + 0.5d,$$

или

$$A = 0.3/25 + 0.9/3.$$

Если универсальное множество состоит из чисел от 1 до 10, т. е.

$$U = 1 + \dots + 10,$$

то нечеткое подмножество, обозначаемое словом **несколько**, определяется как:

$$\text{несколько} = 0.5/3 + 0.8/4 + 1/5 + 1/6 + 0.8/7 + 0.5/8.$$

Для счетного универсального множества

$$U = 0 + 1 + 2 + \dots$$

нечеткое множество, обозначаемое словом **малый**, записывается так:

$$\text{малый} = \sum_0^{\infty} \left( 1 + \left( \frac{u}{10} \right)^2 \right)^{-1} / u. \quad (1.11)$$

Это представление нечеткого множества в виде объединения составляющих его нечетких одноточечных множеств  $\mu_i u_i$  (или  $\mu_i / u_i$ ). Из определения операции объединения нечетких множеств следует, что в случае, когда  $u_i = u_j$  запись нечеткого множества  $A$  преобразуется:

$$\mu_i u_i + \mu_j u_i = (\mu_i \vee \mu_j) u_i.$$

Так запись

$$A = 0.3a + 0.8a + 0.5b$$

можно преобразовать в

$$A = (0.3 \vee 0.8)a + 0.5b = 0.8a + 0.5b.$$

Если носитель нечеткого множества  $A$  имеет непрерывную мощность континуума, то используется запись:

$$A = \int_U \mu_A(u) / u, \quad (1.12)$$

имея в виду, что  $\mu_A(u)$  — степень принадлежности элемента  $u$  множеству  $A$ , а знак  $\int$  обозначает объединение нечетких одноточечных множеств  $\mu_A(u) / u, u \in U$ .

Пусть универсальное множество представляет собой интервал  $[0, 100]$ , а переменная  $u$ , принимающая значения из этого интервала, интерпретируется как **возраст**. Тогда нечеткое подмножество, определяемое словом **старый**, можно записать как

$$\text{старый} = \int_{50}^{100} \left( 1 + \left( \frac{u-50}{5} \right)^{-2} \right)^{-1} / u. \quad (1.13)$$

Точкой перехода для этого нечеткого множества, т. е. значением  $u$ , для которого

$$\mu_{\text{старый}}(u) = 0.5,$$

Для представления нечетких множеств используются диаграммы Эйлера-Венна, на которых универсальное множество изображается прямоугольником, а его подмножества – кругами или эллипсами (рис. 1.1.8.).

Выделение в универсальном множестве  $U$  одного множества (множества  $A$ ), разбивает прямоугольник на две непересекающиеся области, в которых характеристическая функция  $\mu_A$  принимает разные значения:  $\mu_A=1$  внутри эллипса и  $\mu_A=0$  вне эллипса. Добавление еще одного множества – множества  $B$ , (рис. 1.1.8 (б)), снова делит каждую из уже имеющихся двух областей на две подобласти. Образуется  $2 \cdot 2 = 2^2 = 4$  непересекающиеся области, каждая из которых соответствует определенной паре значений характеристических функций  $(\mu_A, \mu_B)$ . Например, пара (01) соответствует области, в которой  $\mu_A=0$ ,  $\mu_B=1$ . Эта область включает в себя те элементы универсального множества  $U$ , которые не принадлежат множеству  $A$ , но принадлежат множеству  $B$ .

Добавление третьего множества – множества  $C$ , (рис. 1.1.8 (в)), снова делит на две подобласти каждую из уже имеющихся четырех областей. Образуется  $4 \cdot 2 = 2^3 = 8$  непересекающихся областей. Каждая из них соответствует определенной тройке значений характеристических функций  $(\mu_A, \mu_B, \mu_C)$ . Эти тройки можно рассматривать как номера областей, записанные в двоичной системе счисления. Например, №  $101_2=5_{10}$ , т.е. область, в которой находятся элементы множеств  $A$  и  $C$ , но нет элементов множества  $B$ , – это область №5. Таким образом, каждая из восьми областей имеет свой двоичный номер, несущий информацию о принадлежности или непринадлежности элементов этой области множествам  $A$ ,  $B$  и  $C$ .

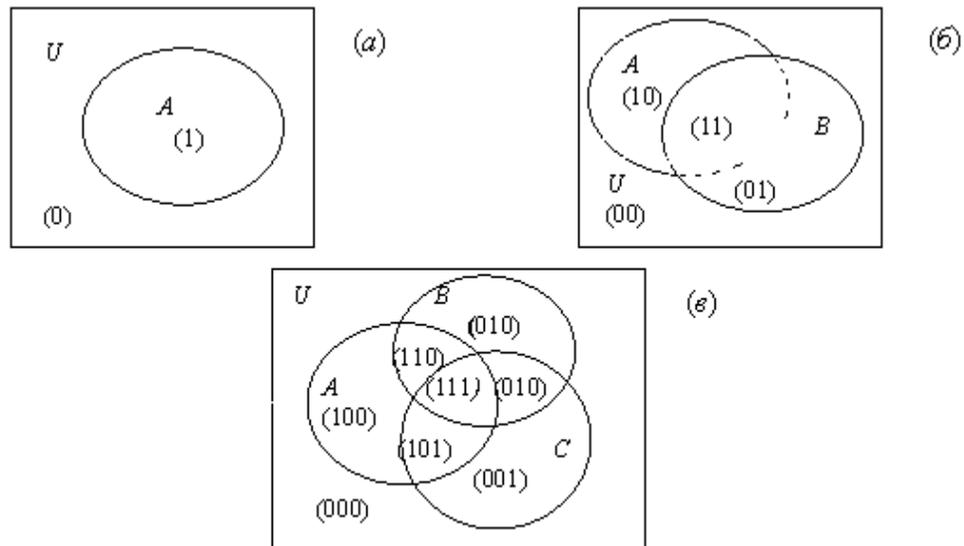


Рис. 1.1. Диаграммы Эйлера-Венна

Добавляя четвертое, пятое и т.д. множества, получаем  $2^4, 2^5, \dots, 2^n$  областей, каждая из которых имеет свой вполне определенный двоичный номер, составленный из значений характеристических функций множеств. Подчеркнем, что последовательность нулей и единиц в любом из номеров выстроена в определенном порядке. Только при условии упорядоченности, двоичный номер области несет информацию о принадлежности или непринадлежности элементов этой области каждому из множеств.

Примечание. последовательность  $n$  действительных чисел  $(x_1, x_2, \dots, x_n)$ ,  $x_i \in R (i = 1, 2, \dots, n)$  в линейной алгебре рассматривается как  $n$ -мерный арифметический вектор с координатами  $x_1, x_2, \dots, x_n$ . Двоичный номер области также может быть назван двоичным вектором, координаты которого принимают значения во множестве  $B = \{0, 1\} : (x_1, x_2, \dots, x_n)$ ,  $x_i \in B (i = 1, 2, \dots, n)$ . Число различных  $n$ -мерных двоичных векторов равно  $2^n$ .

### Лингвистическая переменная

Вычислительные машины оказались эффективными при работе только с точными системами, и не решили основных проблем гуманитарных областях

знаний. Существует принцип несовместимости: высокая точность несовместима с большой сложностью системы. Методы анализа систем, основанные на точной обработке численных данных, не способны моделировать огромную сложность процессов человеческого мышления и принятия решений. Поэтому при изучении поведения организованных систем пришлось отказаться от строгости математического анализа, и использовать приближенные методы. Это привело к появлению лингвистических переменных, значениями которых являются слова или предложения на естественном или формальном языке.

Слова и предложения менее конкретны, чем описание числами. Например, фраза «Джон молод» менее конкретна, чем «Джону 25 лет» и слово «молод» можно рассматривать, как лингвистическое значение переменной. Лингвистическое значение **молод** играет такую же роль, как и численное значение 25, но является менее точным. Также лингвистические значения **очень молодой, не молодой, чрезвычайно молодой, не очень молодой** и т. д. сопоставляются с численными значениями 20, 21, 22, 23.

Лингвистические переменные служат средством приближенного описания явлений, очень сложных некорректно определенных, и неподдающихся точному описанию. Значения лингвистической переменной изображают графически как площадки (как нечеткие переменные) только с нечетко очерченными границами. Принцип обобщения позволяет часть математического аппарата четких систем, приспособить к лингвистическим переменным.

Совокупность значений лингвистической переменной составляет термножество этой переменной, имеющей бесконечное число элементов. Например, термножество лингвистической переменной **Возраст** можно записать так:

$T$  (Возраст) = молодой + не молодой + очень молодой + не очень молодой + очень очень молодой + ... + старый + не старый + очень старый + не очень старый + не очень молодой и не очень старый + ... .. + среднего + ... + не старый и не среднего возраста + чрезвычайно старый.

Знак + обозначает здесь объединение, а не арифметическое суммирование. Аналогично, терм-множество лингвистической переменной Внешность можно записать следующим образом:

$T$  (Внешность) = прекрасная + хорошенькая + миловидная + красивая + привлекательная + не прекрасная + очень хорошенькая + очень очень красивая + более или менее хорошенькая + довольно хорошенькая + довольно красивая + безусловно красивая + не очень привлекательная, но и не очень непривлекательная + ...

В случае лингвистической переменной **Возраст** числовая переменная **возраст**, принимающая значения 0, 1, 2, 3, ..., 100, является базовой переменной лингвистической переменной **Возраст**.

Нечеткое ограничение на значения базовой переменной характеризуется функцией принадлежности, которая каждому значению базовой переменной ставит в соответствие число из интервала  $[0,1]$ .

Например, значения функции принадлежности таких численных значений переменной **возраст**, как 22, 28 и 35 с нечетким ограничением **молодой**, могут быть 1, 0.7 и 0.2 соответственно. Смысл лингвистической величины **молодой** можно представить графиком функции принадлежности лингвистического значения **молодой** относительно базовой переменной **возраст**.

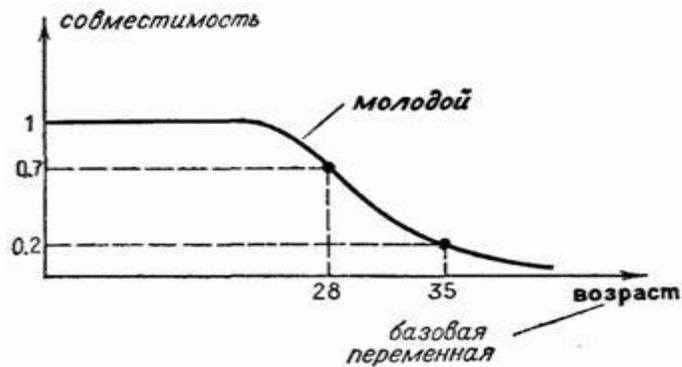


Рис. 1.2.1. Функция совместимости для значения молодой

Джон принадлежит классу молодых людей, а класс молодых людей - нечеткое множество, т. е. что не существует резкого перехода от молодости к «немолодости». Имя «Джон» рассматривается как название составной лингвистической переменной, компонентами которой являются лингвистические переменные **Возраст, Рост, Вес, Внешность** и т. п. Тогда высказывание «Джон молод» представляется, как уравнение назначения

$\text{Возраст} = \text{молодой}$ , предписывающей значение молодой лингвистической переменной  $\text{Возраст}$ . Значение молодой представляется, как название некоторого нечеткого ограничения на базовую переменную **возраст**, причем смысл этого ограничения определяется его функцией принадлежности.



Рис. 1.2.2. Назначение лингвистических значений признакам Джона

Лингвистическую переменную, можно представить в виде иерархической структуры Возраст, нечеткими ограничениями, представляющими смысл ее значений, и значениями базовой переменной **возраст**.

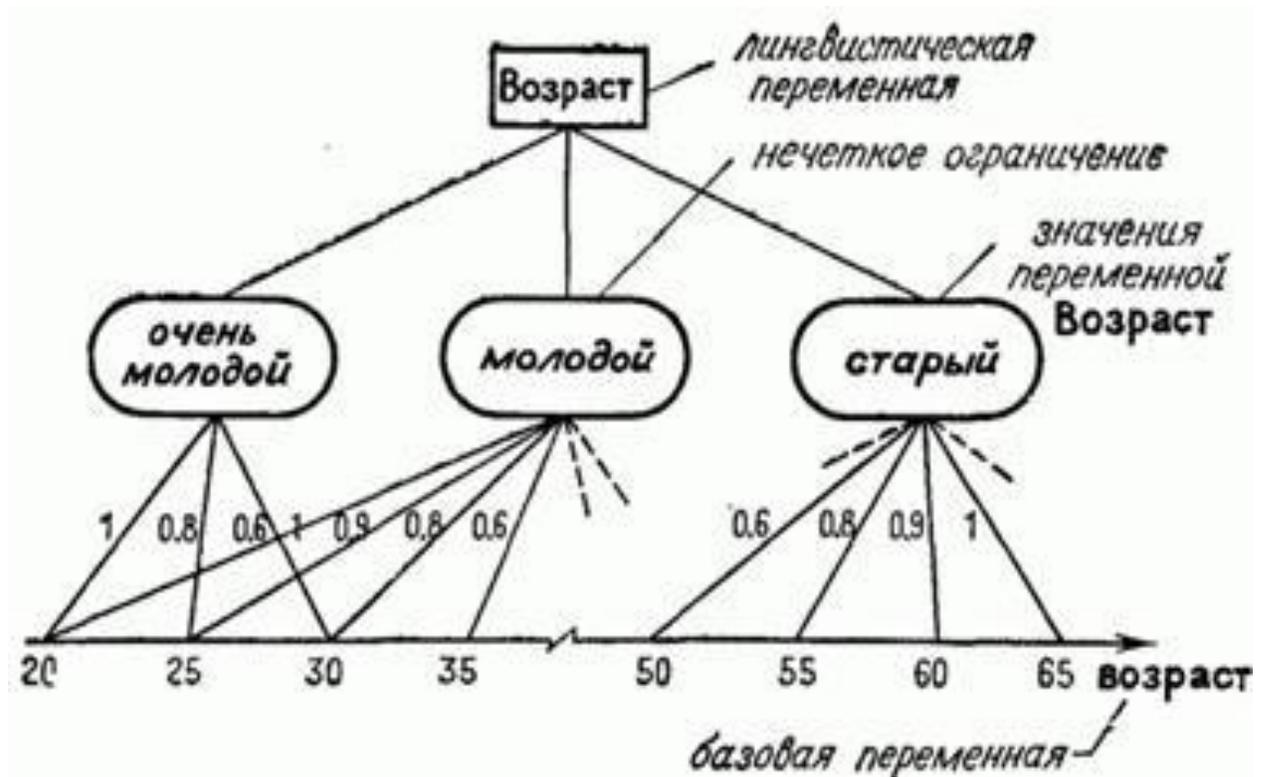


Рис. 1.2.3. Иерархическая, структура лингвистической переменной.

Лингвистическая переменная характеризуется набором  $(\mathcal{X}, T(\mathcal{X}), U, G, M)$ , в котором  $\mathcal{X}$  — название переменной,  $T(\mathcal{X})$  (или просто  $T$ ) обозначает термножество переменной  $\mathcal{X}$ , (множество названий лингвистических значений переменной  $\mathbf{X}$ ). Каждое из таких значений является нечеткой переменной  $X$  со значениями из универсального множества  $U$  с базовой переменной  $u$ .

$G$  — синтаксическое правило (имеющее обычно форму грамматики), порождающее названия  $X$  значений переменной  $\mathbf{X}$ .  $M$  — семантическое

правило, ставящее в соответствие каждой нечеткой переменной  $X$  ее смысл  $M(X)$ , (нечеткое подмножество  $M(X)$  универсального множества  $U$ ).

Конкретное название  $X$ , порожденное синтаксическим правилом  $G$ , называется термом. Терм, состоящий из одного слова или нескольких слов, всегда фигурирующих вместе друг с другом, называется атомарным термом. Терм, состоящий из одного или более атомарных термов, называется составным термом. Конкатенация некоторых компонент составного терма является подтермом. Если  $X_1, X_2, \dots$  — термы в  $T$ , то  $T$  можно представить в виде объединения

$$T = X_1 + X_2 + \dots$$

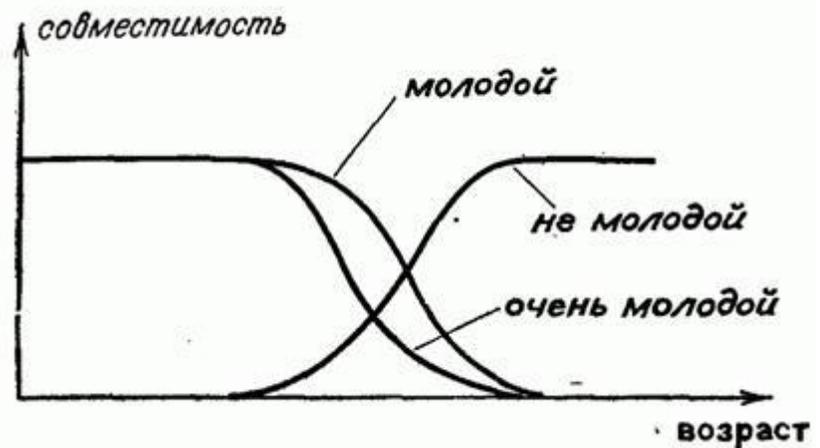
Смысл  $M(X)$  терма  $X$  определяется как ограничение  $R(X)$  на базовую переменную  $u$ , обусловленное нечеткой переменной  $X$ :

$$M(X) \triangleq R(X),$$

имея в виду, что  $R(X)$  и, следовательно,  $M(X)$  можно рассматривать как нечеткое подмножество множества  $U$ , имеющее название  $X$ . Связь между ее лингвистическим значением и базовой переменной показана на рис. 1.2.3.

Лингвистическое значение включает связки, такие, как “и, или, ..., ни и т. п.; отрицание не; неопределенности, как очень, более или менее, совершенно, совсем, безусловно, чрезвычайно, отчасти и т. п”. Связки, неопределенности и отрицание можно трактовать как операторы, видоизменяющие смысл первичных термов особым образом.

Если функция принадлежности лингвистического значения “молодой” изображается кривой, показанной на рис. 1.2.4, то смысл лингвистического значения “очень молодой” получается возведением в квадрат значений функции совместимости лингвистического значения **молодой**. Смысл лингвистического значения “не молодой” можно получить, вычитая из 1 значения этой функции



совместимости (рис. 1.2.4).

Рис. 1.2.4. Функции совместимости значений “молодой, не молодой, очень молодой”.

Лингвистическая переменная “Возраст” является по своей природе числовой переменной. Смысл лингвистического значения, такого, как молодой, определяется функцией принадлежности, ставящей в соответствие каждому значению базовой переменной из интервала  $[0,100]$  число из интервала  $[0,1]$ , представляющее принадлежность данного возраста с понятием “молодой”.

Лингвистическая переменная “Внешность” не имеет четко определенной базовой переменной. Степень красоты с помощью физических величин трудно выразить. В этом случае каждой женщине приписывается степень принадлежности классу прекрасных женщин, например Фэй - степень 0.9, Адели - 0.7, Кетти - 0.8 и Вере - 0.9. Но эти значения функции принадлежности основаны лишь на впечатлениях, которые точно не формализуются.

Базовая переменная - численная по природе. Лингвистическими переменными можно оперировать путем использования принципа обобщения для нечетких множеств. Для красоты способ обращения с лингвистическими переменными носит качественный характер. Поскольку присутствуют вычисления, лингвистический подход не является по своей природе целиком качественным, а имеет соответствующие вычисления.

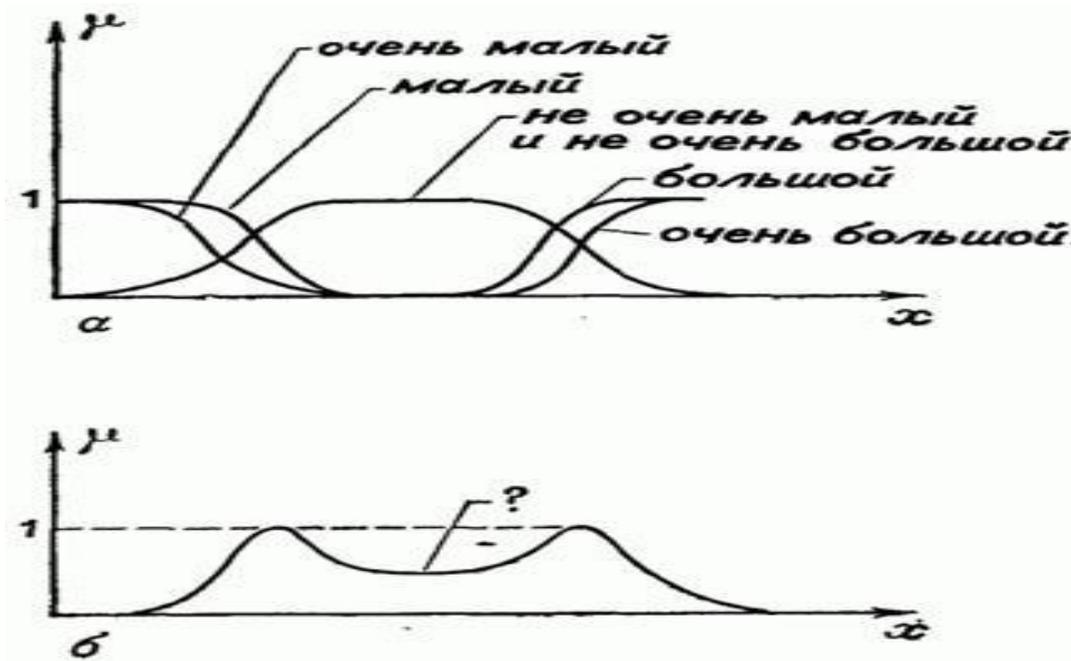


Рис. 1.2.5. а. Функции принадлежности значений малый, очень малый, большой, очень большой и не очень малый и не очень большой. б.

Лингвистическое приближение представляет задачу нахождения приближенного лингвистического описания функции принадлежности. Поэтому лингвистическая переменная представляет область приближенных рассуждений.

Нечеткую переменную можно представить сумкой с мягкими стенками (Рис.1.2.6). Если лингвистическая переменная определяется как переменная, значениями которой являются нечеткие переменные, то лингвистическая переменная представляется в виде твердой сумки, в которую можно поместить

мягкие сумки с ярлыками. На каждом из этих ярлыков приведено описание нечеткого ограничения на множество предметов, которые можно поместить в соответствующую мягкую сумку.

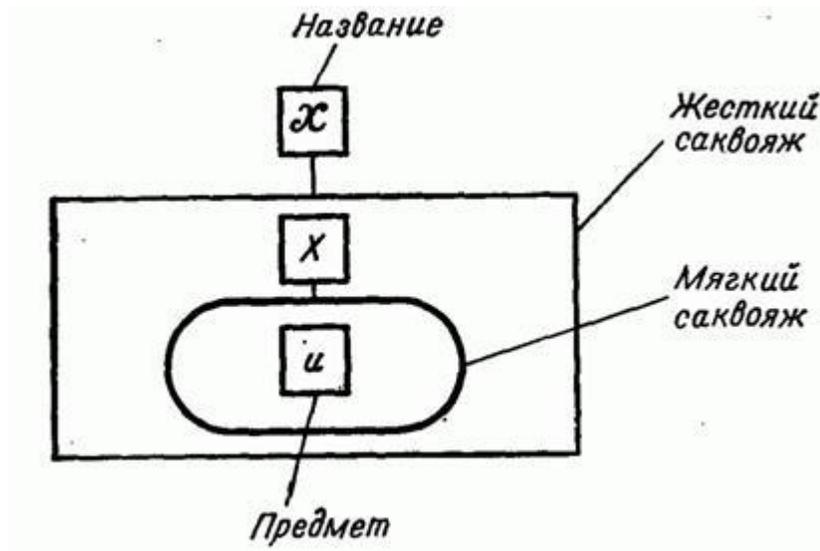


Рис. 1.2.6 Аналогия с саквояжем для лингвистической переменной

### 1.3. Функции принадлежности

Нечеткие множества описывают расплывчатые понятия человеческого языка и для этого характеризуются функцией принадлежности. На практике функция принадлежности задается вне самой задачи и, ее адекватность не может быть проверена теоретически. Поэтому в каждом методе построения функции принадлежности формулируются свои требования и обоснования.

Функция принадлежности приписывает каждому элементу  $x \in X$  степень его принадлежности к нечеткому множеству  $A$ , при этом возможны три варианта:

1)  $\mu_A(x) = 1$  означает полную принадлежность элемента  $x$  к нечеткому множеству  $A$ , т.е.  $x \in A$ ;

2)  $\mu_A(x) = 0$  означает отсутствие принадлежности элемента  $x$  к нечеткому множеству  $A$ , т.е.  $x \notin A$ ;

3)  $0 < \mu_A(x) < 1$  означает частичную принадлежность элемента  $x$  к нечеткому множеству  $A$ .

Если  $\mathbf{X}$  - это пространство с конечным количеством элементов, т.е.  $\mathbf{X} = \{x_1, \dots, x_N\}$ , то нечеткое множество  $A \subseteq \mathbf{X}$  представляется, как

$$A = \frac{\mu_A(x_1)}{x_1} + \frac{\mu_A(x_2)}{x_2} + \dots + \frac{\mu_A(x_n)}{x_n} = \sum_{i=1}^n \frac{\mu_A(x_i)}{x_i} \quad (1.14)$$

Эта запись имеет символьный характер и означает приписывание конкретным

элементам  $x_1, \dots, x_n$  степеней принадлежности  $\mu_A(x_1), \dots, \mu_A(x_n)$ . Запись  $\frac{\mu_A(x_i)}{x_i}$ ,  $i = 1, \dots, n$  означает пару  $(x_i, \mu_A(x_i))$ ,  $i = 1, \dots, n$ . (1.14)

Функций принадлежности определяется следующими условиями:

1. Функция принадлежности должна быть положительной,

т.е.  $\mu_{A_i}(x) > 0$ .

2. Функция принадлежности должна быть нормальной

$$\text{Sup} \mu_{A_i} x = 1.$$

Если условие нормальности принято, то запрещается использование функций принадлежности, не удовлетворяющих этому условию.

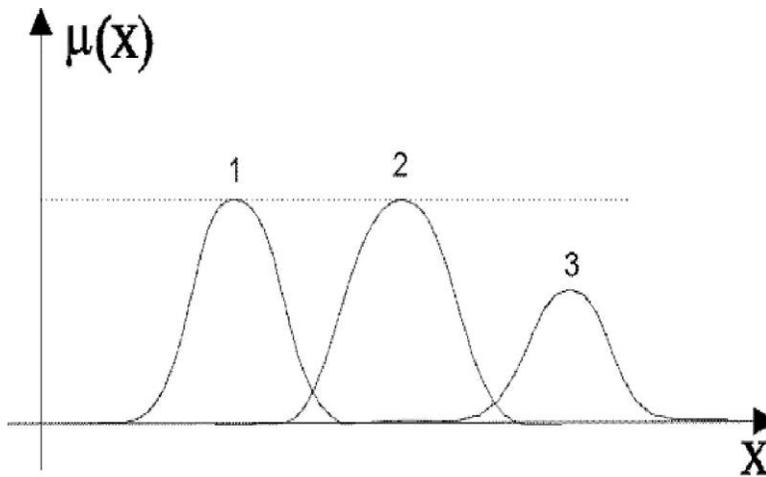


Рис. 1.3.1. Нормальные (1,2) и запрещенные (3) функции принадлежности

3. В базовом множестве термов  $T$  запрещается использование пар термов, представленных рис. (1.3.2. а, б). В первом случае отсутствует разделение понятий, представленных соседними термами  $\tau_i$  и  $\tau_{i+1}$ , во втором - участку  $[c, d]$  из области определения не поставлено в соответствие какое-либо понятие.

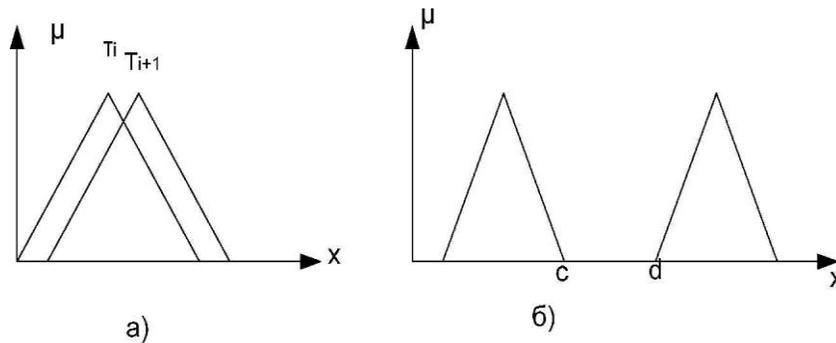


Рис. 1.3.2. Пары термов запрещенные к использованию

4. Термы с минимальными и максимальными номерами не должны соответствовать колоколообразным функциям принадлежности. Для этих термов функции принадлежности имеют S-образный вид

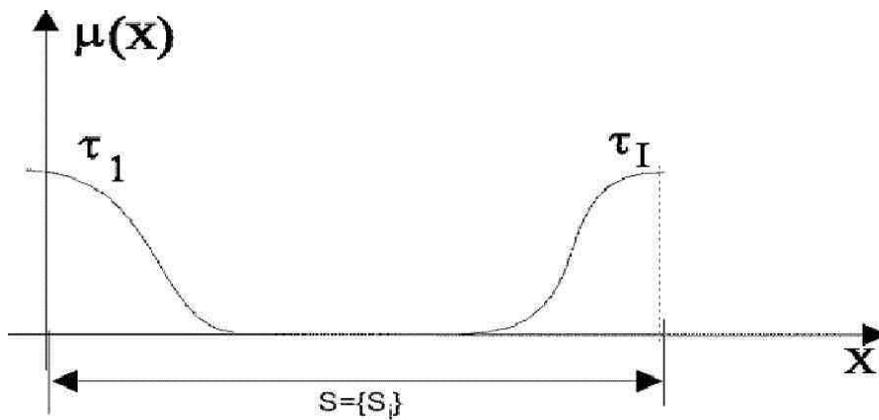


Рис. 1.3.3. Функция принадлежности для термов с минимальными и максимальными номерами

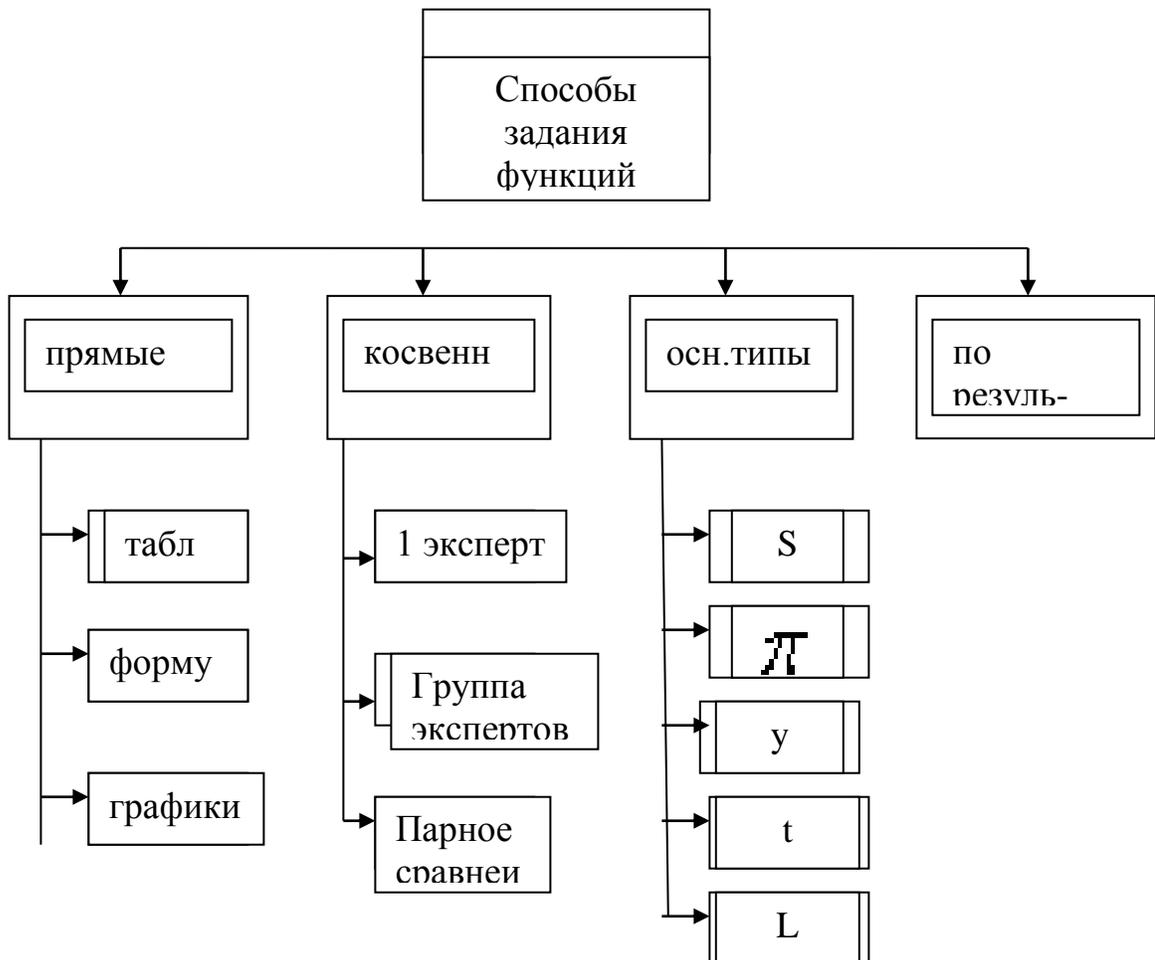
Можно выделить следующие способы задания функций принадлежности (Рис. 1.3.4.)

- прямые
- косвенные
- типовые формы
- по результатам эксперимента

В прямых методах эксперт просто задает правила определения значений функции принадлежности  $\mu_A$  множества  $A$ . Эти значения согласуются с его предпочтениями на множестве  $U$ .

Целесообразность прямых методов обосновывается в [4]: «По своей природе оценка является приближением. Во многих случаях не требуется высокая точность. Человеческий мозг использует допустимость такой неточности, кодируя информацию элементами нечетких множеств/ Поток информации, поступающий в мозг через органы зрения, слуха, осязания и др., сжимается в тонкую струйку информации, необходимой для решения поставленной задачи с минимальной степенью точности».

Прямые методы, основанные на непосредственном определении функции принадлежности, используются в случае, когда эксперты свободны от случайных ошибок и работают как «надежные и правильные приборы».



ε

Рис. 1.3.4. Классификация способов построения функций принадлежности

Примеры прямых методов: непосредственное задание  $\mu_A$  таблицей, формулой, графиком. Прямые методы задания функции принадлежности используются для измеримых величин, таких как скорость, время, расстояние, давление, температура и т. д., или когда выделяются полярные значения.

Во многих задачах при характеристике объекта можно выделить набор признаков и для каждого из них определить полярные значения,

соответствующие значениям функции принадлежности, 0 или 1. При этом используют следующую процедуру:

- определить список свойств, по которым оценивается объект;
- найти в этом списке полярные свойства и сформировать полярную шкалу;
- для каждой пары полюсов оценить, как сильно он обладает этим свойством.

Например, в задаче распознавания лиц можно выделить следующие шкалы:

	Признаки	<b>0</b>	<b>1</b>
$x_1$	Высота лба	Низкий	Широкий
$x_2$	Профиль носа	Курносый	Горбатый
$x$	Длина носа	Короткий	Длинный
$x_4$	Разрез глаз	Узкие	Широкие
$x_5$	Цвет глаз	Светлые	Темные
$x_6$	Форма	Остроконечный	Квадратный
$x$	Толщина губ	Тонкие	Толстые
$x_8$	Цвет лица	Темный	Светлый
$x$	Очертание лица	Овальное	Квадратное

Для конкретного лица  $A$  эксперт, исходя из приведенной шкалы, задает

$\mu_A(x) \in [0,1]$ , формируя векторную функцию принадлежности  $\{\mu_A(x_1), \mu_A(x_2), \dots, \mu_A(x_9)\}$ .

Также используются прямые методы для группы экспертов, когда, например, группе экспертов предъявляют конкретное лицо, и каждый должен дать один из двух ответов: «этот человек лысый» или «этот человек не лысый», тогда количество утвердительных ответов, деленное на общее число экспертов, дает значение  $\mu_{\text{«лысый»}}$  (данного лица).

Допустим, что  $\mathbf{X} = \mathbf{N}$  - множество натуральных чисел. Определим понятие множества натуральных чисел, «близких числу 7». Это можно сделать определением следующего нечеткого множества  $A \subseteq \mathbf{X}$ :

$$A = \frac{0,2}{4} + \frac{0,5}{5} + \frac{0,8}{6} + \frac{1}{7} + \frac{0,8}{8} + \frac{0,5}{9} + \frac{0,2}{10} .$$

Если  $\mathbf{X} = \mathbf{R}$ , где  $\mathbf{R}$  - множество действительных чисел, то множество действительных чисел, «близких числу 7», можно определить функцией принадлежности вида

$$\mu_A(x) = \frac{1}{1+(x-7)^2}. \quad (1.15)$$

Поэтому нечеткое множество действительных чисел, «близких числу 7», описывается выражением

$$A = \int_x \left[ \frac{1}{1+(x-7)^2} \right] dx.$$

Нечеткие множества натуральных или действительных чисел, «близких числу 7», можно записать различными способами. Например, функцию принадлежности (3.10) можно заменить выражением

$$\mu_A(x) = \begin{cases} 1 - \sqrt{\frac{|x-7|}{3}} & \text{при } 4 \leq x \leq 10, \\ 0 & \text{в противном случае.} \end{cases} \quad (1.16)$$

На рис. 1.3.1 а и б представлены две функции принадлежности нечеткого множества  $A$  действительных чисел, «близких числу 7».

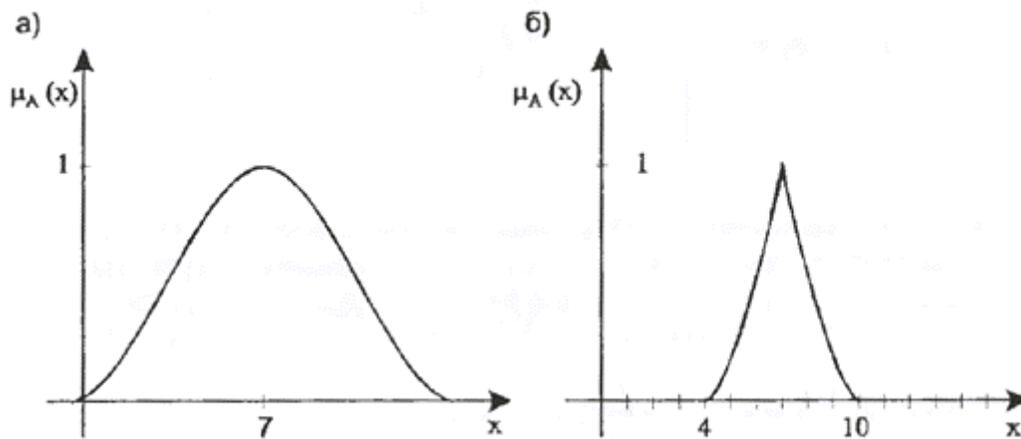


Рис. 1.3.1. Функции принадлежности нечеткого множества действительных чисел, «близких числу 7».

Выразим неточное определение «комфортная температура для купания». Зададим область рассуждений в виде множества  $X = [15^\circ, \dots, 25^\circ]$ . Отдыхающий I, лучше всего чувствующий себя при температуре  $21^\circ$ , определил бы для себя нечеткое множество

$$A = \frac{0,1}{16} + \frac{0,3}{17} + \frac{0,5}{18} + \frac{0,8}{19} + \frac{0,95}{20} + \frac{1}{21} + \frac{0,9}{22} + \frac{0,8}{23} + \frac{0,75}{24} + \frac{0,7}{25}. \quad (1.17)$$

Отдыхающий II, предпочитающий температуру  $20^\circ$ , предложил бы другое определение этого множества:

$$B = \frac{0,1}{15} + \frac{0,2}{16} + \frac{0,4}{17} + \frac{0,7}{18} + \frac{0,9}{19} + \frac{1}{20} + \frac{0,9}{21} + \frac{0,85}{22} + \frac{0,8}{23} + \frac{0,75}{24} + \frac{0,7}{25}. \quad (1.18)$$

С помощью нечетких множеств  $A$  и  $B$  представлено неточное определение понятия «комфортная температура для купания».

**Косвенные** методы определения значений функции принадлежности используются, если нет измеримых свойств, и значения функции принадлежности должно удовлетворять заранее сформулированным условиям.

К таким условиям могут относиться следующие:

- функция принадлежности отражает близость к заранее выделенному эталону;

- при парном сравнении объектов, если один объект в  $a$  раз предпочтительнее второго, то второй объект оценивается в  $1/a$  раз предпочтительнее первого; и т. д.

Косвенные методы основаны на слабых предположениях об экспертах как «измерительных приборах». Косвенные методы более трудоемкие, чем прямые, но их преимущество в стойкости по отношению к искажениям в ответах. Один из наиболее известных методов является метод парных сравнений Т. Л. Саати. Если значения функций принадлежности известны, то парные сравнения представляются матрицей отношений.

Эксперт формирует матрицу, при этом предполагается, что диагональные элементы равны 1, а для элементов симметричных относительно диагонали  $a_{ij} = 1/a_{ji}$ , т. е. если один объект в  $a$  раз предпочтительнее другого, то второй объект оценивается в  $1/a$  раз предпочтительнее первого.

При формировании оценок парных сравнений, эксперта просят отразить опыт следующим образом:

- установить какой из двух предлагаемых объектов более важен;
- оценить различия в виде ранга важности по определенной ранговой шкале.

### **Стандартные формы функций принадлежности.**

1. Функция принадлежности класса  $S$  (рис. 1.3.2) определяется как

$$s(x, a, b, c) = \begin{cases} 0 & \text{для } x \leq a, \\ 2\left(\frac{x-a}{c-a}\right)^2 & \text{для } a \leq x \leq b, \\ 1-2\left(\frac{x-c}{c-a}\right)^2 & \text{для } b \leq x \leq c, \\ 1 & \text{для } x \geq c. \end{cases} \quad (1.19)$$

где  $b = (a+c)/2$ . Функция принадлежности, относящаяся к этому классу, имеет графическое представление (рис.1.3.6), напоминающее букву «S», причем ее форма зависит от подбора параметров  $a$ ,  $b$  и  $c$ . В точке  $x = b = (a+c)/2$  функция принадлежности класса  $S$  принимает значение, равное 0,5.

2. Функция принадлежности класса  $\pi$  (рис. 1.3.7) определяется через функцию принадлежности класса  $S$ :

$$\pi(x, b, c) = \begin{cases} s(x, c-b, c-b/2, c) & \text{для } x \leq c, \\ 1-s(x, c, c+b/2, c+b) & \text{для } x \geq c. \end{cases} \quad (1.20)$$

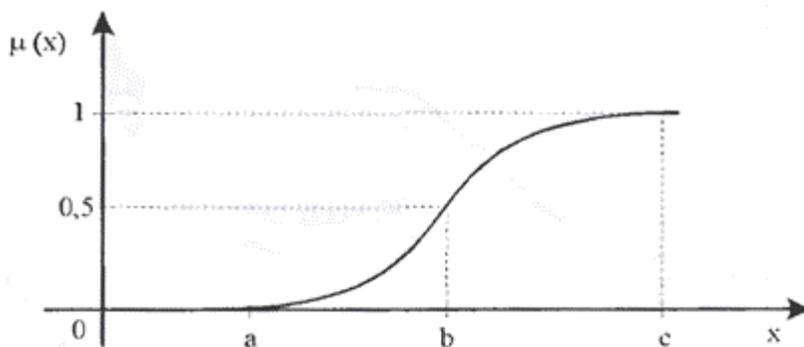


Рис. 1.3.6. Функция принадлежности класса  $S$ .

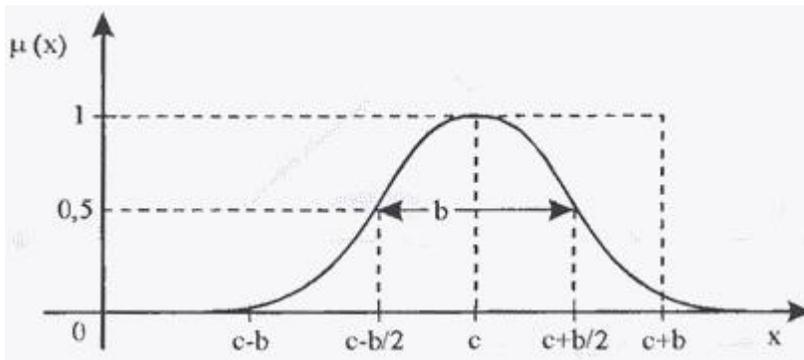


Рис. 1.3.7. Функция принадлежности класса  $\pi$ .

Функция принадлежности класса  $\pi$  принимает нулевые значения для  $x \geq c+b$  и  $x \leq c-b$ . В точках  $x = c \pm b/2$  ее значение равно 0,5.

3. Функция принадлежности класса  $\gamma$  (рис. 1.3.8) задается выражением

$$\gamma(x, a, b) = \begin{cases} 0 & \text{для } x \leq a, \\ \frac{x-a}{b-a} & \text{для } a \leq x \leq b, \\ 1 & \text{для } x \geq b. \end{cases} \quad (1.21)$$

Функции принадлежности классов  $s$  и  $\gamma$  подобны.

4. Функция принадлежности класса  $t$  (рис. 1.3.9) определяется в виде

$$t(x, a, b, c) = \begin{cases} 0 & \text{для } x \leq a, \\ \frac{x-a}{b-a} & \text{для } a \leq x \leq b, \\ \frac{c-x}{c-b} & \text{для } b \leq x \leq c, \\ 0 & \text{для } x \geq c. \end{cases} \quad (1.22)$$

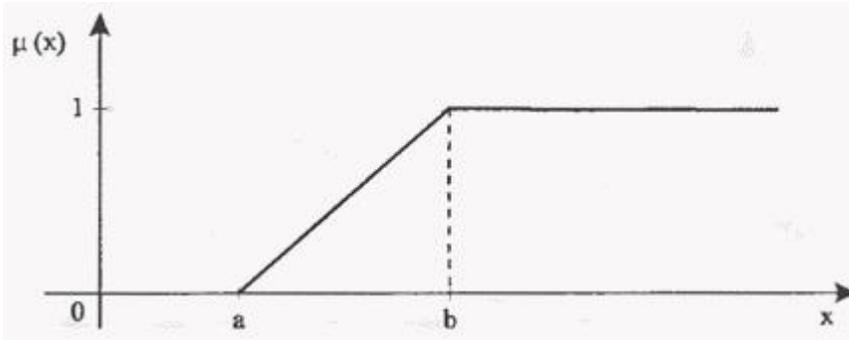


Рис. 1.3.8. Функция принадлежности класса  $\mathcal{Y}$ .

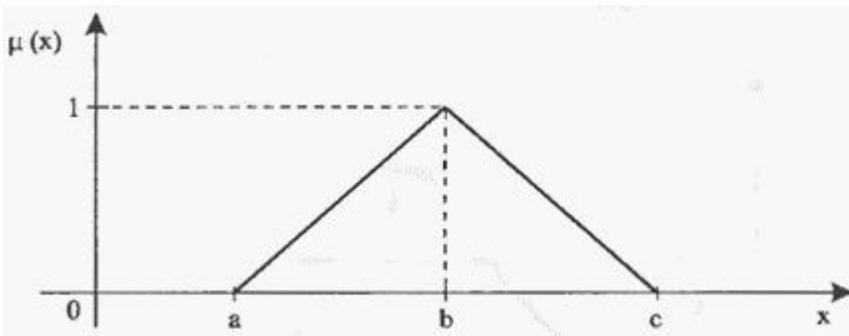


Рис. 1.3.9. Функция принадлежности класса  $\mathcal{Z}$ .

В некоторых приложениях функция принадлежности класса  $\mathcal{Z}$  может быть альтернативной по отношению к функции класса  $\mathcal{X}$ .

5. Функция принадлежности класса  $\mathcal{L}$  (рис. 1.3.10) определяется выражением

$$L(x, a, b) = \begin{cases} 1 & \text{для } x \leq a, \\ \frac{b-x}{b-a} & \text{для } a \leq x \leq b, \\ 0 & \text{для } x \geq b. \end{cases} \quad (1.23)$$

Рассмотрим три неточных формулировки:

- 1) «малая скорость автомобиля»;
- 2) «средняя скорость автомобиля»;

3) «большая скорость автомобиля».

В качестве области рассуждений примем диапазон  $[0, x_{\max}]$ , где  $x_{\max}$  - это максимальная скорость. На рис. 1.3.11 представлены нечеткие множества  $A$ ,  $B$  и  $C$ , соответствующие приведенным формулировкам.

Функция принадлежности:

множества  $A$  имеет тип  $L$ ,

множества  $B$  - тип  $t$ ,

множества  $C$  - тип  $\gamma$ .

В точке  $x = 40$  км/час функция принадлежности нечеткого множества «малая скорость автомобиля» принимает значение 0,5, т.е.  $\mu_A(40) = 0,5$ . Такое же значение принимает функция принадлежности нечеткого множества «средняя скорость автомобиля», т.е.  $\mu_B(40) = 0,5$ , тогда как  $\mu_C(40) = 0$ .

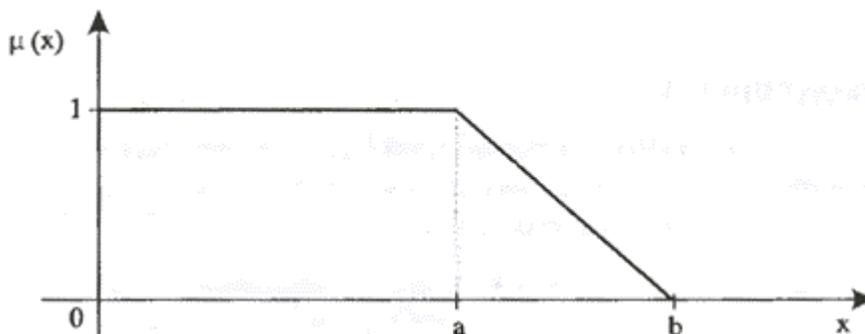


Рис. 1.3.11. Функция принадлежности класса  $L$ .

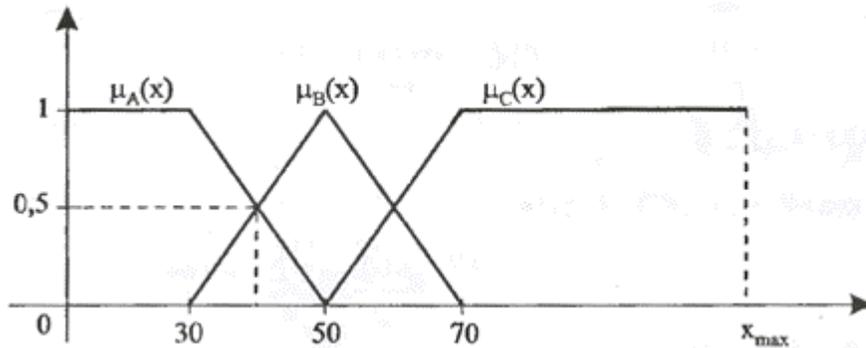


Рис. 1.3.12. Функции принадлежности нечетких множеств «малая» ( $\mu_A(x)$ ), «средняя» ( $\mu_B(x)$ ), «большая» ( $\mu_C(x)$ ) скорость автомобиля.

Если для измеряемого параметра известна вероятность перехода системы от допустимого состояния к недопустимому, в зависимости от режима работы, то функция принадлежности, может быть определена из условия предпочтения тех режимов, для которых степень риска меньше. Условие предпочтения может быть записано в следующей форме

$$x_1 \succ x_2 \Leftrightarrow [1 - P(x < x_1)] \geq [1 - P(x < x_2)]$$

, где  $P(x)$  – закон распределения

Тогда в качестве функции принадлежности можно принять функцию

$$\mu_D(x_3) = [1 - P(x < x_3)]$$

Эта функция удовлетворяет требованиям к функции принадлежности и монотонно изменяется при варьировании режима работы. Практически  $P(x < x_3)$  определяется на основе построения гистограмм накопленных частот перехода подсистем для разных режимов работы.

На рис.1.3.13 приведена гистограмма накопленных частот срабатывания клапанов (функция не доходит до значений, равных 1, так как некоторые клапаны не срабатывают). Тогда функция принадлежности к подмножеству

допустимых режимов для клапанов - может быть записана в виде таблицы или в аналитической форме.

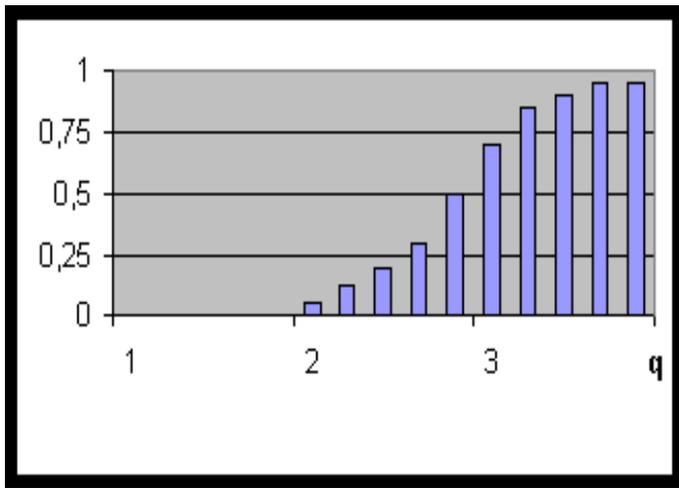


Рис. 1.3.13. Гистограмма накопленных частот

#### 1.4. Нечеткая арифметика

Нечеткие числа широко используются в повседневной жизни. Говоря: «приблизительно три», «приблизительно двадцать пять» и т. п., и тем самым используем нечеткие числа.

$X_1$  = «примерно 9»,

$X_2$  = «примерно 10»,

$y = x_1 + x_2$ .

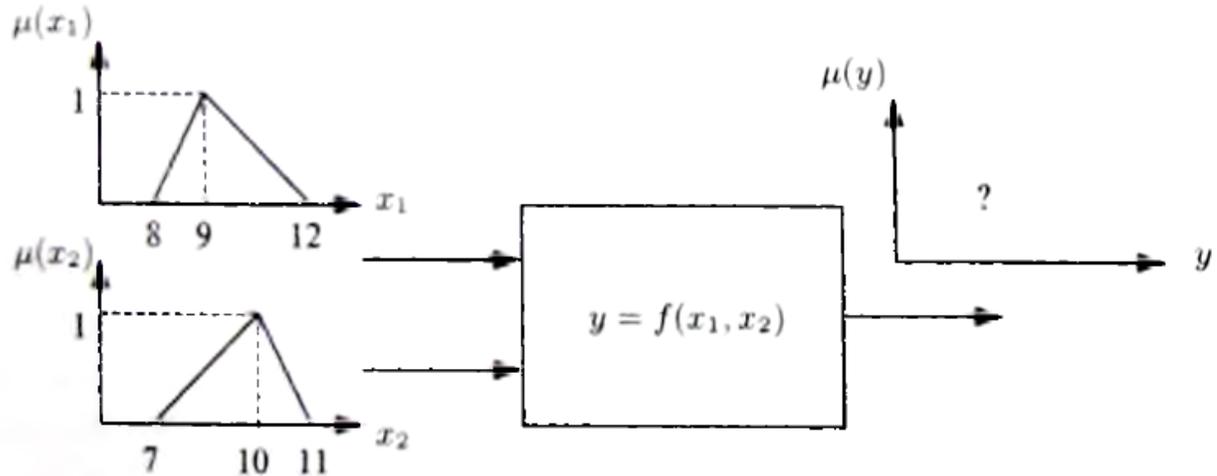


Рис. 1.4.1. Выходное нечеткое значения модели с нечеткой информацией о входных значениях

Нечетким числом называется нечеткое множество  $A$ , определенное на множестве действительных чисел  $A \subseteq \mathbf{R}$ , функция принадлежности которого  $\mu_A: \mathbf{R} \rightarrow [0,1]$  и отвечает условиям:

- 1)  $\sup_{x \in \mathbf{R}} \mu_A(x) = 1$ , нечеткое множество  $A$  нормализовано;
- 2)  $\mu_A[\lambda x_1 + (1-\lambda)x_2] \geq \min\{\mu_A(x_1), \mu_A(x_2)\}$ , множество  $A$  выпуклое;
- 3)  $\mu_A(x)$  функция принадлежности множества нечеткого числа

На рис. 1.4.2. представлены примеры нечетких чисел. В теории нечетких систем различаются положительные и отрицательные нечеткие числа.

Нечеткое число  $A \subseteq \mathbf{R}$  положительно, если  $\mu_A(x) = 0$  для всех  $x < 0$ .

Нечеткое число  $A \subseteq \mathbf{R}$  отрицательно, если  $\mu_A(x) = 0$  для всех  $x > 0$ .

На рис. 1.4.3. представлен пример положительного и отрицательного нечетких чисел, а также такого нечеткого числа, которое не является ни положительным, ни отрицательным.

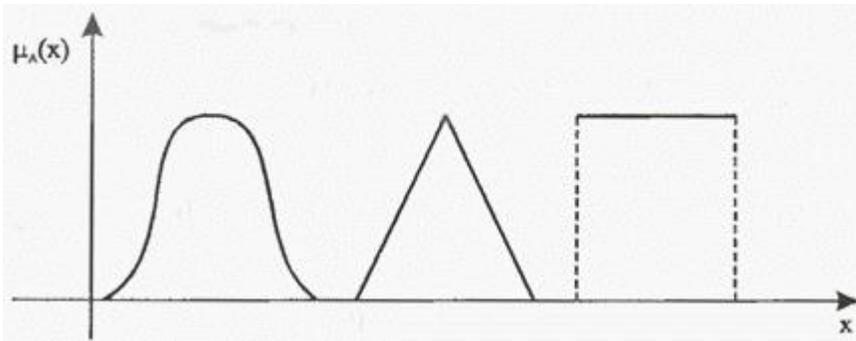


Рис. 1.4.2.. Примеры нечетких чисел.

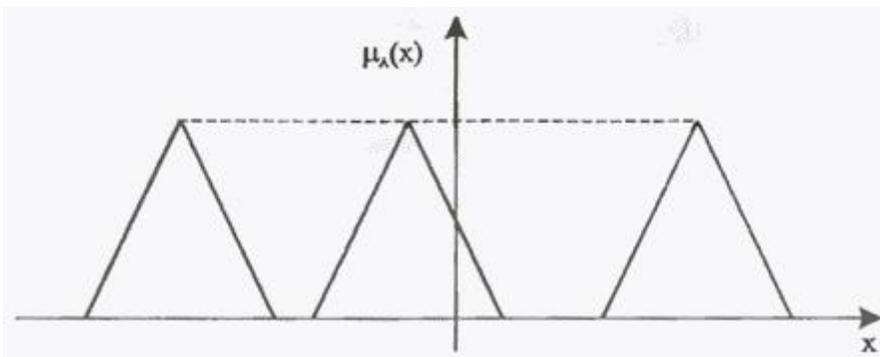


Рис. 1.4.3. Примеры нечетких чисел: положительного, отрицательного, а также такого, которое не является ни положительным, ни отрицательным.

Нечеткое число  $X$  на действительной прямой называется нормальным, если  $\max \mu^{\wedge}(x) = 1$ .

Для арифметических операций на нечетких числах используют принцип расширения (обобщения), позволяющий определять суммирование, вычитание, умножение и деление для двух нечетких чисел  $A_1, A_2 \subseteq \mathbf{R}$ , также как и для четких.

Пусть имеется четкое отображение  $f$  пространства  $X$  в пространство  $Y$

$$f: X \rightarrow Y.$$

Пусть  $A$  будет заданным нечетким множеством, определенным в пространстве  $X$ , т.е.  $A \subseteq X$ . Если нечеткое множество  $A$  имеет вид (3.3), т.е.

$$A = \frac{\mu_A(x_1)}{x_1} + \frac{\mu_A(x_2)}{x_2} + \dots + \frac{\mu_A(x_n)}{x_n}$$

и отображение  $f$  является взаимно однозначным, то принцип расширения заключается в том, что генерируемое этим отображением и определенное в пространстве  $Y$  нечеткое множество  $B$  имеет вид

$$B = f(A) = \frac{\mu_A(x_1)}{f(x_1)} + \frac{\mu_A(x_2)}{f(x_2)} + \dots + \frac{\mu_A(x_n)}{f(x_n)}.$$

Пусть 
$$A = \frac{0,1}{3} + \frac{0,4}{2} + \frac{0,7}{5}$$

и  $f(x) = 2x + 1$ . В соответствии с принципом расширения получаем

$$B = f(A) = \frac{0,1}{7} + \frac{0,4}{5} + \frac{0,7}{11}.$$

Основные арифметические операции на нечетких числах  $A_1, A_2 \subseteq \mathbf{R}$  определяются следующим образом:

а) суммирование двух нечетких чисел  $A_1$  и  $A_2$  обозначается

$$A_1 \oplus A_2 \stackrel{\text{def}}{=} B,$$

Функция принадлежности суммы задается в виде

$$\mu_B(y) = \sup_{\substack{x_1, x_2 \\ y=x_1+x_2}} \min\{\mu_{A_1}(x_1), \mu_{A_2}(x_2)\}, \quad (1.24)$$

б) вычитание двух нечетких чисел  $A_1$  и  $A_2$  обозначается

$$A_1 \ominus A_2 \stackrel{def}{=} B, \quad (3.104)$$

Функция принадлежности разности задается в виде

$$\mu_B(y) = \sup_{\substack{x_1, x_2 \\ y=x_1-x_2}} \min\{\mu_{A_1}(x_1), \mu_{A_2}(x_2)\}, \quad (1.25)$$

в) умножение двух нечетких чисел  $A_1$  и  $A_2$  обозначается

$$A_1 \odot A_2 \stackrel{def}{=} B,$$

Функция принадлежности произведения задается в виде

$$\mu_B(y) = \sup_{\substack{x_1, x_2 \\ y=x_1 \cdot x_2}} \min\{\mu_{A_1}(x_1), \mu_{A_2}(x_2)\}, \quad (1.26)$$

г) деление двух нечетких чисел  $A_1$  и  $A_2$  обозначается

$$A_1 \oslash A_2 \stackrel{def}{=} B,$$

Функция принадлежности частного задается в виде

$$\mu_B(y) = \sup_{\substack{x_1, x_2 \\ y=x_1, x_2}} \min \{ \mu_{A_1}(x_1), \mu_{A_2}(x_2) \} \quad (1.27)$$

В первую очередь нужны нечеткие числа, имеющие непрерывные функции принадлежности.

Сложим и перемножим два нечетких числа, имеющих вид

$$A_1 = \frac{0,7}{2} + \frac{1}{3} + \frac{0,6}{4},$$

$$A_2 = \frac{0,8}{3} + \frac{1}{4} + \frac{0,5}{6}.$$

В соответствии с формулой (1.24) получаем

$$\begin{aligned} A_1 \oplus A_2 &= \frac{\min(0,7;0,8)}{5} + \frac{\max\{\min(0,7;1), \min(1;0,8)\}}{6} + \frac{\max\{\min(1;1), \min(0,6;0,8)\}}{7} + \\ &+ \frac{\max\{\min(0,7;0,5), \min(0,6;1)\}}{8} + \frac{\min(1;0,5)}{9} + \frac{\min(0,6;0,5)}{10} = \\ &= \frac{0,7}{5} + \frac{0,8}{6} + \frac{1}{7} + \frac{0,6}{8} + \frac{0,5}{9} + \frac{0,5}{10}. \end{aligned} \quad (1.28)$$

На основании выражения (1.26) получаем

$$\begin{aligned} A_1 \odot A_2 &= \frac{\min(0,7;0,8)}{6} + \frac{\min(0,7;1)}{8} + \frac{\min(1;0,8)}{9} + \\ &+ \frac{\max\{\min(0,7;0,5), \min(1;1), \min(0,6;0,8)\}}{12} + \\ &+ \frac{\min(0,6;1)}{16} + \frac{\min(1;0,5)}{18} + \frac{\min(0,6;0,5)}{24} = \\ &= \frac{0,7}{6} + \frac{0,7}{8} + \frac{0,8}{9} + \frac{1}{2} + \frac{0,6}{16} + \frac{0,5}{18} + \frac{0,5}{24}. \end{aligned} \quad (1.29)$$

Сложили и перемножили два нечетких числа, получив в качестве суммы нечеткое множество (1.28), а в качестве произведения - нечеткое множество (1.29).

Эти нечеткие множества являются нормальными и выпуклыми и представляют нечеткие числа.

Иногда результатом арифметических операций над нечеткими числами оказывается четкое число, потому что нечеткое число не отвечает условию выпуклости. Если нечеткие числа  $A_1$  и  $A_2$  имеют непрерывные функции принадлежности, то результатом арифметических операций суммирования, вычитания, умножения и деления всегда будут нечеткие числа (теорема Дюбуа и Прейда [9]).

Унарные операций на нечетких числах.

1. Операция изменения знака. В результате операции  $f(x) = -x$  получаем нечеткое число, противоположное нечеткому числу  $A \subseteq \mathbf{R}$ . Это число обозначается -  $A \subseteq \mathbf{R}$ , а его функция принадлежности равна

$$\mu_{-A}(x) = \mu_A(-x).$$

Нечеткие числа  $A$  и  $-A$  симметричны относительно оси  $x$ .

2. Операция обращения. В результате операции  $f(x) = x^{-1}$ ,  $x \neq 0$ , получаем нечеткое число, обратное нечеткому числу  $A \subseteq \mathbf{R}$ . Это число обозначается  $A^{-1} \subseteq \mathbf{R}$ , а его функция принадлежности равна

$$\mu_{A^{-1}}(x) = \mu_A(x^{-1}).$$

Нечеткое число  $A$  положительно или отрицательно. Если  $A$  таковым не является, то нечеткое множество  $B = f(A) = A^{-1}$  не выпукло и, следовательно,  $B$  не может считаться нечетким числом.

3. Операция масштабирования. В результате операции  $f(x) = \lambda^x$ ,  $\lambda \neq 0$ , получаем нечеткое число, масштабированное относительно нечеткого числа  $A \subseteq \mathbf{R}$ . Это число обозначается  $\lambda A \subseteq \mathbf{R}$ , а его функция принадлежности равна  $\mu_{\lambda A}(x) = \mu_A(x\lambda^{-1})$ .

4. Операция экспонирования. В результате операции  $f(x) = e^x$ ,  $x > 0$ , получаем степень нечеткого числа  $A \subseteq \mathbf{R}$ . Это число обозначается  $e^A \subseteq \mathbf{R}$ , а его функция принадлежности равна

$$\mu_{e^A}(x) = \begin{cases} \mu_A(\log x) & \text{для } x > 0, \\ 0 & \text{для } x < 0, \end{cases}$$

поэтому  $e^A$  - положительное нечеткое число.

5. Операция абсолютного значения. Абсолютное значение нечеткого числа  $A \subseteq \mathbf{R}$  обозначается  $|A| \subseteq \mathbf{R}$  и определяется как

$$\mu_{|A|}(x) = \begin{cases} \max(\mu_A(x), \mu_A(-x)) & \text{для } x \geq 0, \\ 0 & \text{для } x < 0. \end{cases}$$

$|A|$  - положительное нечеткое число.

Если

$$A = \frac{0,7}{1} + \frac{1}{2} + \frac{0,6}{5},$$

то нечеткое число  $-A$  имеет вид

$$-A = \frac{0,6}{-5} + \frac{1}{-2} + \frac{0,7}{-1},$$

тогда как нечеткое число  $A^{-1}$  записывается в виде

$$A^{-1} = \frac{0,6}{2} + \frac{1}{0,5} + \frac{0,7}{1}.$$

При этом  $A + (-A) \neq \frac{1}{0},$

а также  $A \cdot A^{-1} \neq \frac{1}{1}.$

Арифметические операции над нечеткими числами требуют проведения достаточно сложных вычислений. Поэтому Дюбуа и Прейд [9] предложили форму представления нечетких чисел при помощи трех параметров, что значительно упрощает нечеткую арифметику.

### **(L-R)- аппроксимация нечетких чисел**

Нечеткие числа (L-R) - типа [5] - это нечеткие числа, задаваемые по определенным правилам с целью снижения объема вычислений. Для нечетких чисел (L-R)-типа левые ветви функций принадлежности операндов **A** и **B** аппроксимируются одной монотонно возрастающей функцией **L**, зависящей от двух параметров, подбираемых для каждого операнда в отдельности  $L(a_L, \mathbf{a})$  и  $L(b_L, \mathbf{b})$ . Аналогично для правых ветвей и монотонно убывающей функции **R** имеем  $R(a, a_R)$ ,  $R(b, b_R)$ . Полученные аппроксимации называются L-R нечеткими числами и

обозначаются  $(a_L, a, a_R)$ ,  $(b_L, b^*, b_R)$ . К классу (L-R) функций относятся функции, графики которых имеют следующий вид (рис. 4.2):

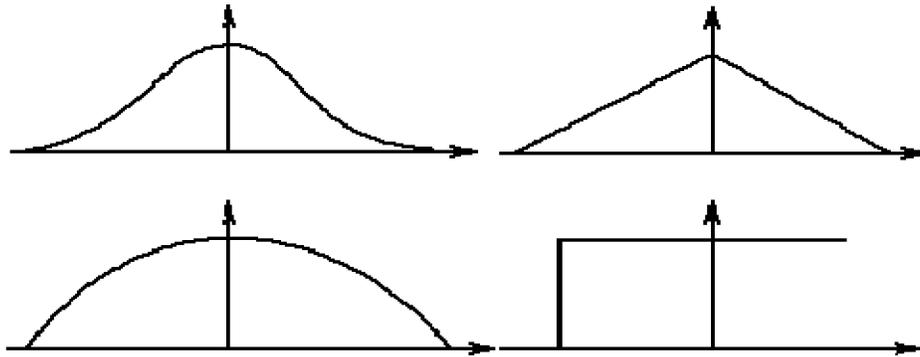


Рис. 1.4.4.. Зависимости для (L-R) функций

В качестве примеров функций L и R можно привести

$$L(x) = P(x) = e^{-|x|^p}, \quad p > 0,$$

$$L(x) = P(x) = \frac{1}{1+|x|^p}, \quad p > 0,$$

$$L(x) = P(x) = \max(0, 1 - |x|^p), \quad p > 0,$$

$$L(x) = P(x) = \begin{cases} 1 & \text{для } x \in [-1; 1], \\ 0 & \text{для } x \notin [-1; 1]. \end{cases}$$

Результат сложения и вычитания L-R нечетких чисел есть также L-R нечеткое число. Результат умножения и деления L-R нечетких чисел будет L-R нечетким числом лишь приблизительно. L-R аппроксимация полезна тем, что сами функции L и R в промежуточных вычислениях не участвуют, а используются лишь при получении окончательного результата.

При решении практических задач нашли простейшие случаи нечетких чисел и нечетких интервалов, получившие свое название по виду их функций принадлежности. Эти нечеткие числа и интервалы можно рассматривать как частный случай нечетких чисел и интервалов (L-R)-типа, если в качестве соответствующих функций R- типа использовать их предельные случаи, а именно - линейные функции (треугольные (рис. 1.4.4) или трапецидальные). При этом треугольные нечеткие числа однозначно задаются тройкой  $(a_L, a, a_R)$ , а трапецидальные четверкой  $(a_L, a_1, a_2, a_R)$ , координаты верхнего основания трапеции, т.е. отпадает необходимость вычисления промежуточных значений результатов арифметических операций.

Для треугольной функции принадлежности нечеткое число  $x \sim x^*$  представляется следующим соотношением:

Пусть L и R - функции, выполняющие отображение

$$(-\infty, \infty) \rightarrow [0, 1]$$

и удовлетворяющие условиям:

$$1) L(-x) = L(x) \quad R(-x) = R(x)$$

$$2) L(0) = 1, \quad R(0) = 1$$

$$3) L \text{ и } R - \text{ функции, невозрастающие на интервале } [0, +\infty).$$

Нечеткое число  $A \subseteq \mathbf{R}$  будет нечетким числом типа L-R тогда, когда его функция принадлежности имеет вид

$$A(x) = \begin{cases} L\left(\frac{m-x}{\alpha}\right), & \text{если } x \leq m, \\ R\left(\frac{x-m}{\beta}\right), & \text{если } x \geq m, \end{cases} \quad (1.30)$$

где  $m$  - действительное число, называемое средним значением нечеткого числа  $A(\mu_A(m)=1)$ ,  $\alpha$  - положительное действительное число ( левосторонний разброс),  $\beta$  - положительное действительное число, (правосторонний разброс).

При увеличении разбросов  $\alpha$  и  $\beta$  число  $A$  становится «более» нечетким. Нечеткое число типа L – R можно сокращенно записать в виде

$$A = (m_A, \alpha_A, \beta_A)_{LR}$$

Нечеткое число «примерно 9» можно определить как

$$A = (9,3,3)_{LR}$$

Функция принадлежности этого числа  $L(x) = P(x) = \frac{1}{1+x^2}$ .

Арифметические операции над нечеткими числами типа L-R сводятся к операции над тремя параметрами.

Нечеткое число, противоположное нечеткому числу, равно  $-A = (-m_A, \alpha_A, \beta_A)_{LR}$

Сумма нечетких чисел

$$A = (m_A, \alpha_A, \beta_A) \text{ и } B = (m_B, \alpha_B, \beta_B)$$

имеет вид

$$A+B = (m_A + m_B, \alpha_A + \alpha_B, \beta_A + \beta_B)_{LR}$$

Другие арифметические операции (например, умножение и деление) над нечеткими числами типа L-R более сложны, а их результат имеет приближенный характер.

Функция принадлежности  $\mu_A(x)$  нечеткого числа типа L-R принимает значение 1 только в точке  $x = m$ . Если  $\mu_A(x) = 1$  во всех точках на интервале  $[m_1, m_2]$ , где  $m_1 < m_2$  и  $m_1, m_2 \in \mathbf{R}$ , то возникает понятие плоского нечеткого числа.

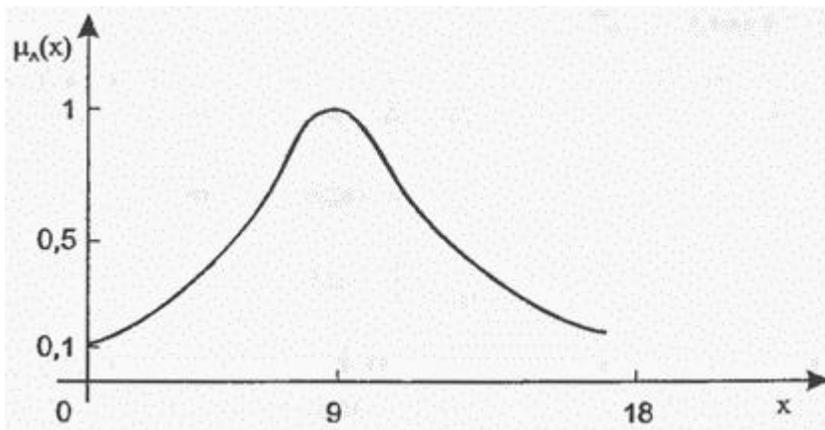


Рис. 1.4.5. Плоское нечеткое число

Это определение плоского нечеткого числа используют для моделирования нечетких интервалов. Плоским нечетким числом типа L-R называется нечеткое число с функцией принадлежности

$$A(x) = \begin{cases} L\left(\frac{m_1 - x}{\alpha}\right), & \text{если } x \leq m_1, \\ 1, & \text{если } m_1 \leq x \leq m_2, \\ P\left(\frac{x - m_2}{\beta}\right), & \text{если } x \geq m_2. \end{cases} \quad (1.31)$$

Плоское нечеткое число  $A$  можно отождествить с нечетким интервалом  $A$  вида

$$A = (m_1, m_2, \alpha_A, \beta_A)_{LR} \quad (3.138)$$

Рассмотрим неточное утверждение «стоимость велосипеда в этом магазине составляет от 3 до 6 тысяч рублей». Адекватной формализацией этого утверждения может считаться нечеткий интервал  $A$  вида

$$A=(3,6,\alpha, \beta)_{LR} \quad (3.139)$$

На рис. 1.4.6 представлен примерный график функции принадлежности нечеткого интервала.

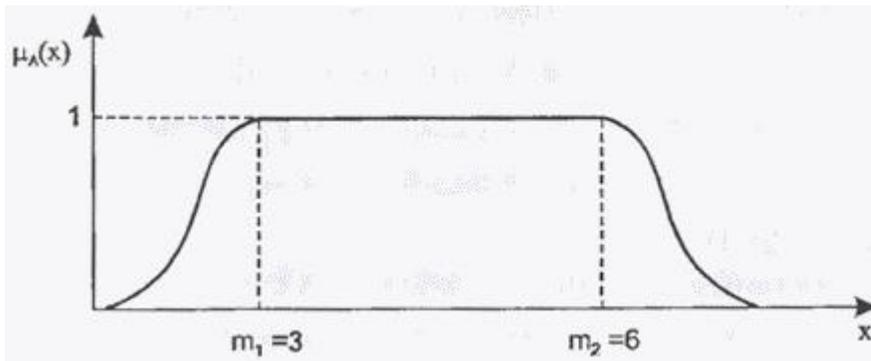


Рис. 1.4.6. Нечеткий интервал «от 3 до 6 тысяч рублей».

### 1.5. Основные операции на нечеткими множествами

Нечеткая логика создавалась на основе классической четкой логики, имеющей недостатки вследствие своей двузначности. Введение понятия нечеткого множества повышает точность моделирования, но различные обобщения (расширения) приводят к неоднозначности определений, имеющих аналогии и в классической теории множеств.

1. Любое определение операции справедливо, если вместо нечетких множеств используются классические множества. Термин функции принадлежности заменяется на характеристические функции. Хотя в принципе это одно и то же, вследствие принципа обобщения
2. Сравнение нечетких множеств и выполнение над ним различных операций возможно при определении их на одном и том же универсуме.

3. Одна и та же функция принадлежности может описывать различные качественные понятия, поэтому свойство в форме нечеткого множества выражает различные субъективные функции и различные нечеткие множества.
4. Операции определяются через действия над их функциями принадлежности.

### **Операция пересечения.**

Операцию пересечения нечетких множеств представляется как расширение соответствующей операции над обычными множествами. Поэтому пересечение обычных множеств является частным случаем пересечения нечетких множеств.

В классической логике пересечение множеств  $A$  и  $B$  определяется без использования функций принадлежности по формуле

$$A \cap B = \{x : x \in A \text{ и } x \in B\}$$

Основные свойства операции пересечения нечетких множеств  $A$  и  $B$ , заданных на универсальном множестве  $X$ , также как четких множеств, определяются следующими соотношениями, графическое представление на рис. 1.5.1.

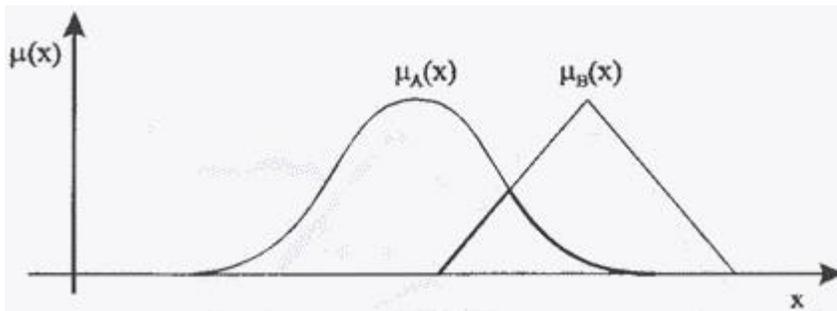


Рис. 1.5.1. Графическое представление операции пересечения нечетких множеств.

### **Коммутативность:**

$$A \cap B = B \cap A.$$

Свойство означает, что порядок следования множеств-операндов не влияет на конечный результат.

**Ассоциативность:**

$$(A \cap B) \cap C = A \cap (B \cap C)$$

Свойство определяет возможность пошагового вычисления логического произведения нескольких множеств путем нахождения произведений их пар. Порядок, в котором формируются пары, не влияет на конечный результат.

**Идемпотентность:**

$$A \cap A = A$$

**Поглощение** (пересечение с пустым множеством 0):

$$A \cap \emptyset = \emptyset$$

**Тождественность** (пересечение с универсальным множеством):

$$A \cap X = A$$

где  $X$  — универсальное множество.

**Закон логического противоречия:**

$$A \cap \bar{A} = \emptyset$$

Для нечетких множеств последняя операция может быть задана различными способами и потому имеет неоднозначный смысл.

Имеются два нечетких множества  $A$  и  $B$ , где  $A$  — множество дешевых автомобилей,  $B$  — множество комфортабельных автомобилей

$$A = \left\{ \frac{1}{x_1}, \frac{0.8}{x_2}, \frac{0.6}{x_3}, \frac{0.4}{x_4}, \frac{0.2}{x_5}, \frac{0}{x_6} \right\}$$

$B$  — множество комфортабельных автомобилей:

$$B = \left\{ \frac{0}{x_1}, \frac{0.2}{x_2}, \frac{0.4}{x_3}, \frac{0.6}{x_4}, \frac{0.8}{x_5}, \frac{1}{x_6} \right\}$$

Нужно определить множество  $C$  автомобилей, являющихся одновременно дешевыми и комфортабельными. Поскольку  $A = B$ , то, в случае четких множеств, получили бы пустое множество.

Автомобиль п 4 может быть отнесен к дешевым со степенью  $\mu=0.4$  и к комфортабельным — со степенью  $\mu=0.6$ . Для определения автомобиля дешевым и комфортабельным, Л. Заде предложил вычислять значения функции принадлежности произведения множеств с использованием оператора MIN :

$$\mu_{A \cap B}(x) = \text{MIN}(\mu_A(x), \mu_B(x)), \quad \forall x \in X$$

Применяя эту формулу для нахождения множества  $A \cap B$  дешевых и, одновременно с этим, комфортабельных автомобилей, получаем выражение:

$$C = \left\{ \frac{0}{x_1}, \frac{0.2}{x_2}, \frac{0.4}{x_3}, \frac{0.4}{x_4}, \frac{0.2}{x_5}, \frac{0}{x_6} \right\}$$

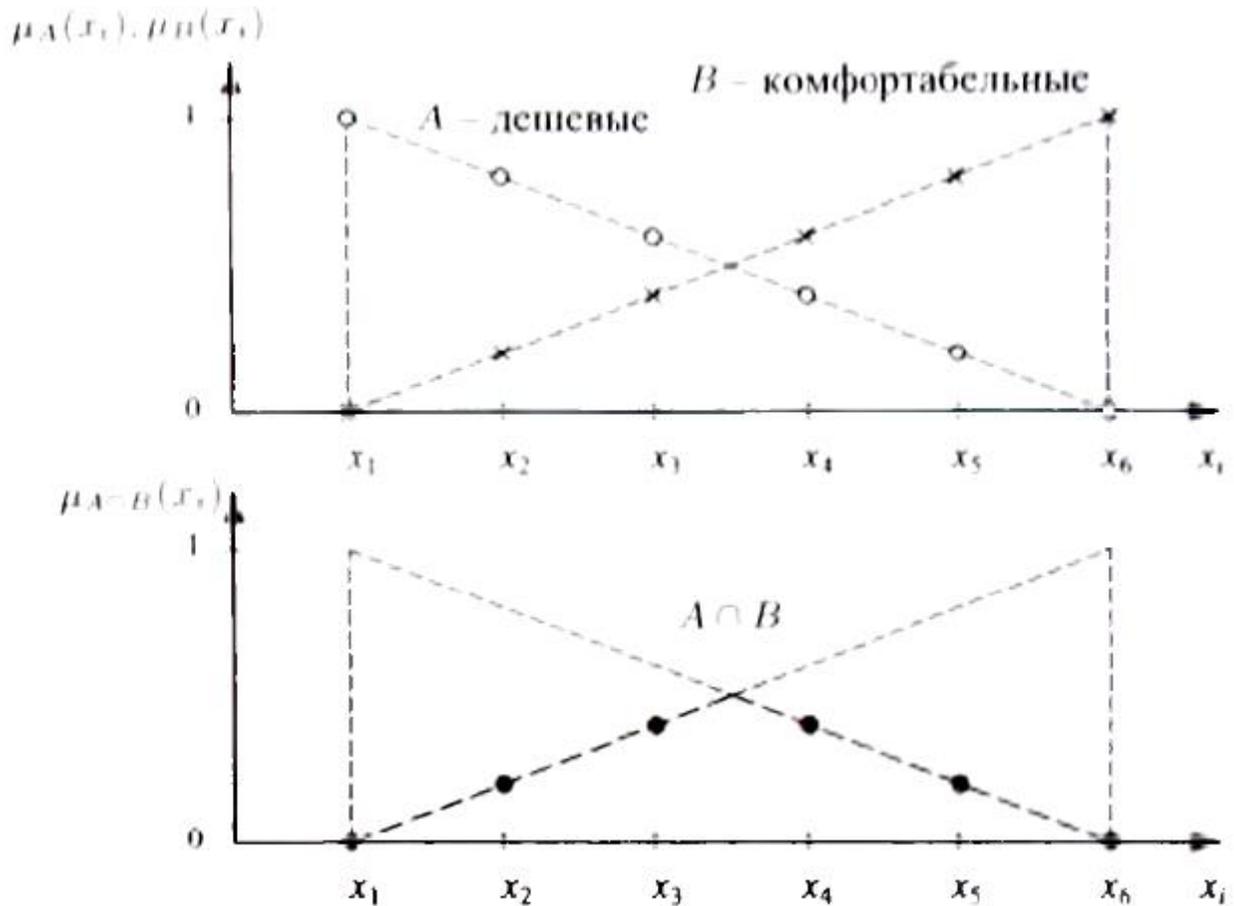


Рис. 1.5.2. Произведение  $A * B$  нечетких множеств дешевых ( $A$ ) и комфортабельных ( $B$ ) автомобилей, полученное с использованием оператора MIN

#### Достоинства операции MIN:

1. Простота и скорость вычислений.
2. Возможность «сглаживания» действия оператора MIN, при повышении объема вычислений, с понижением точности.

#### Недостатки операции MIN:

1. Точность модели ниже, чем при использовании других операторов.
2. Менее гладкая поверхность, чем для других операторов (оператор жесткий).
3. Нечувствительность и резкое изменение значений выходной величины модели и нечеткого регулятора, содержащих операторы MIN.

Существуют другие операторы снижающие жесткость и повышающие точность и гладкость полученной функции.

- мягкий оператор MIN с функцией дельта.
- мягкий оператор MIN с функцией soft
- оператор PROD

Реализация пересечения нечетких множеств операторов как MIN, так и PROD, указывает на неоднозначность способа выполнения данной операции, поэтому оператор min (пересечение) часто выбирается интуитивно, исходя из опыта, на основе каких-либо гипотез или же методом проб.

Вычисление функции принадлежности произведения нечетких множеств с использованием оператора PROD осуществляется в соответствии с формулой

$$\mu_{A \cap B}(x) = \mu_A(x) \cdot \mu_B(x), \quad \forall x \in X$$

Преимущество оператора PROD в том, что значение  $\mu_{A \cap B}(x)$  имеет количественную зависимость от фактических значений обеих функций принадлежности  $\mu_A(x)$  и  $\mu_B(x)$  (за исключением случая равенства одной из функций нулю). Потеря информации здесь не так существенна, как для оператора MIN, когда значение зависит лишь от меньшего (в пределах заданной области изменения  $x$  значений) компонентов  $\mu_A(x)$  и  $\mu_B(x)$ . Сравнение результатов вычисления логического произведения с использованием операторов MIN и PROD представлено на рис. 1.5.3.

### **Нечеткое расширение пересечения или t- норма**

В качестве операторов пересечения  $A \cap B$  используются t -нормы, определяющие различные формы реализации операции пересечения. Такие t-нормы представляет собой функцию T, моделирующую операцию пересечения двух нечетких множеств A и B, удовлетворяющую перечисленным ниже свойствам, которые выполняются для всех  $x \in X$ .

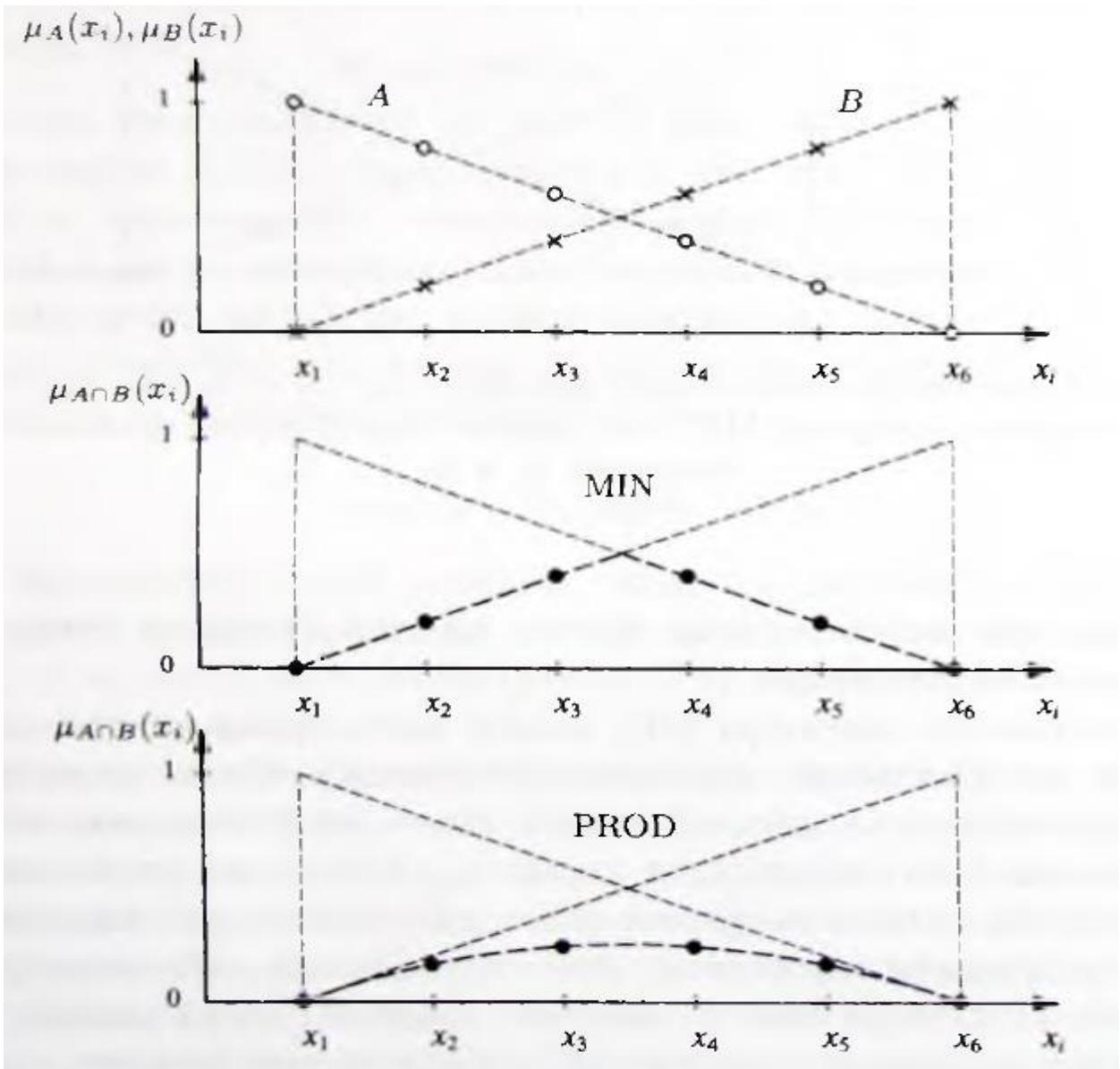


Рис. 1.5.3. Функции принадлежности логического произведения нечетких множеств  $A \cap B$ , полученные с использованием операторов MIN и PROD

Свойство коммутативности указывает на то, что для данной операции порядок следования множеств не является существенным. Свойство ассоциативности говорит о том, что операция пересечения выполняется для более, чем двух множеств. Основные операции над нечеткими множествами, необходимо выполнять последовательно, но порядок образования пар

множеств не влияет на конечный результат. Свойство монотонности означает, что при возрастании значений аргументов результат операции не убывает.

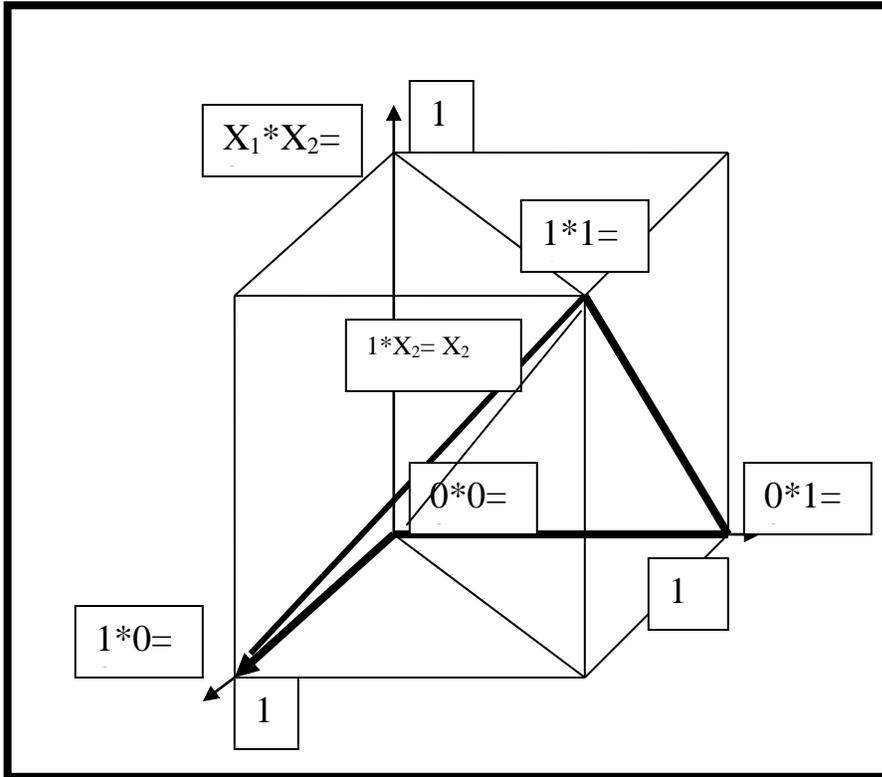


Рис. 1.5.4. Геометрическая интерпретация t-норм

**Пространства отображения:**

$$T : [0, 1] \times [0, 1] \rightarrow [0, 1]$$

**Свойство обнуления:**

$$T(0, 0) = 0,$$

**Случай, когда пара содержит один элемент с  $\mu(x) = 1$ :**

$$T(\mu_A(x), 1) = \mu_A(x), \quad T(\mu_B(x), 1) = \mu_B(x)$$

**Свойство коммутативности:**

$$T(\mu_A(x), \mu_B(x)) = T(\mu_B(x), \mu_A(x))$$

**Свойство ассоциативности:**

$$T(\mu_A(x), T(\mu_B(x), \mu_C(x))) = T(T(\mu_A(x), \mu_B(x)), \mu_C(x))$$

**Условие монотонности:**

$$\begin{aligned} \mu_A(x) \leq \mu_C(x), \mu_B(x) \leq \mu_D(x) &\Rightarrow \\ \Rightarrow T(\mu_A(x), \mu_B(x)) \leq T(\mu_C(x), \mu_D(x)) \end{aligned}$$

Выделяют параметризованные и непараметризованные t-нормы. Результат действия непараметризованных t-норм является постоянным, тогда как для параметризованных t-норм он будет изменяться как количественно, так и качественно, при изменении любого параметра, являющегося степенью свободы оператора.

В соответствии с четкой логикой, температура не может быть одновременно средней и высокой, в то время как нечеткая логика позволяет существование такого множества. Функция принадлежности этого множества не является строго определенной и зависит от используемого оператора t-нормы.

Оператор MIN приводит к самым высоким значениям функции принадлежности, поэтому другие операторы t-норм называют sub-MIN-операторами или sub-MIN-нормами. Значения функции принадлежности произведения множеств  $\mu_{A \cap B}(x)$ , получаемые с помощью sub-MIN-операторов (t-норм), будут меньше, чем при использовании оператора MIN, что означает, что sub-MIN-операторы являются более строгими, требующими более высокой степени выполнения условий A и B, входящих в нечеткое произведение. Поэтому об операторе MIN говорят, как о наиболее качественном среди t-норм. Согласно степени качества, t-нормы могут быть упорядочены следующим образом: минимум > произведение Гамахера > алгебраическое произведение > произведение Эйнштейна > ограниченная разность > усиленное произведение. Для определения операции пересечения также используются операторы, не являющиеся t-нормами (т. е. не обладающие свойствами t-норм).

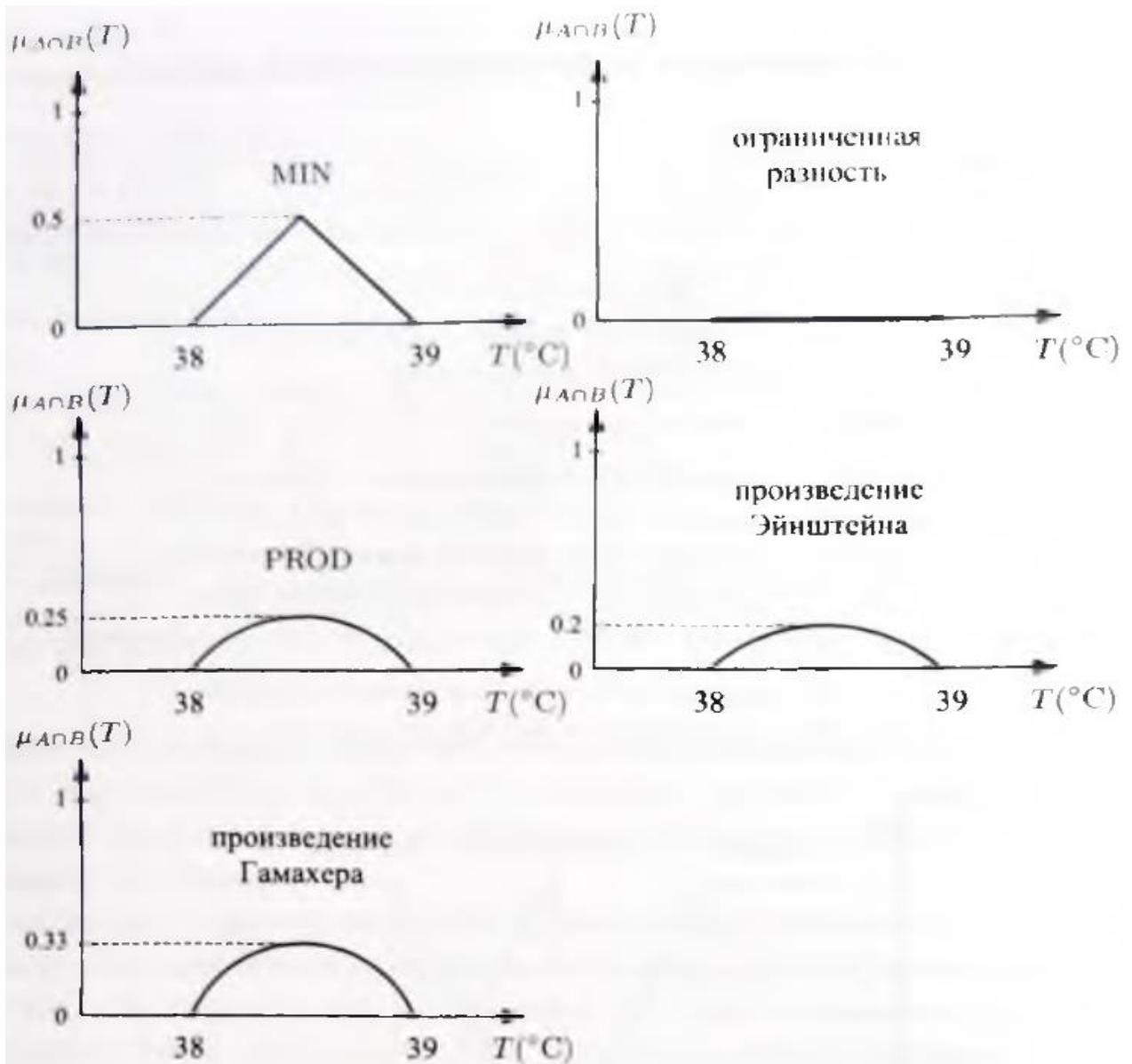


Рис. 1.4.5. Графические изображения операторов t - норм

Параметризованный оператор пересечения на основе среднего (Driankov 1993),  

$$\mu_{A \cap B}(x) = \gamma \cdot \text{MIN}(\mu_A(x), \mu_B(x)) + 0.5(1 - \gamma)(\mu_A(x) + \mu_B(x)), \quad \forall x \in X, \quad (4.25)$$

где  $\gamma \in [0, 1]$ .

При  $\gamma = 1$  данный оператор сводится к оператору MIN. При  $\gamma = 0$  получается оператор среднего арифметического:

$$\mu_{A \cap B}(x) = 0.5(\mu_A(x) + \mu_B(x)), \quad \forall x \in X$$

Поскольку неравенство:

$$0.5(\mu_A(x) + \mu_B(x)) \geq \text{MIN}(\mu_A(x), \mu_B(x)), \quad \forall x \in X$$

всегда справедливо, оператор среднего называют также super-MIN-оператором.

По степени качества он превосходит наиболее качественную t-норму — оператор MIN.

Также используются для многих нечетких множеств операторы

- среднего гармонического
- среднего геометрического
- обобщенного среднего

Все операторы, определенные на основе оператора обобщенного среднего, относятся к типу super-MIN и вследствие этого являются более качественными, чем оператор MIN..

### **Объединение (логическая сумма) нечетких множеств**

В классической логике логическая сумма множеств  $A + B$  определяется без использования понятия функции принадлежности, согласно выражению

$$A \cup B = \{x: x \in A \text{ или } x \in B\}$$

Пример логического суммирования представлен на рис. 1.5.6. Результат объединения четких множеств является однозначным, поскольку объединение выполняется всегда одним и тем же способом. В случае нечетких множеств возможен ряд способов выполнения объединения, и тем самым результат его неоднозначен. Все операции нечетких множеств, совпадают с операциями классической логики, также как и свойства

1. Функция принадлежности суммы нечетких множеств  $A_1, A_2, \dots, A_n$  выражается зависимостью

$$\mu_{A_1 \cup A_2 \cup \dots \cup A_n}(x) = \bigcup_{i=1}^n \mu_{A_i}(x) = \max[\mu_{A_1}(x), \mu_{A_2}(x), \dots, \mu_{A_n}(x)] \quad \text{для } x \in X.$$

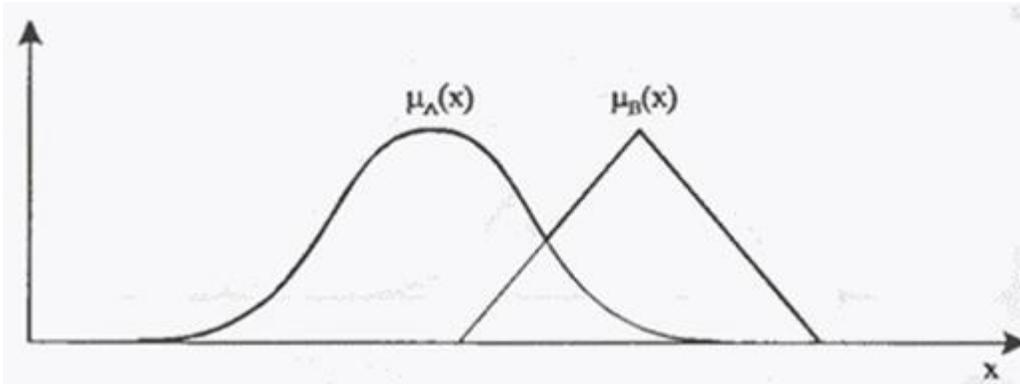


Рис. 1.5.6. Графическое представление операции суммирования нечетких множеств.

2. Свойство выпуклости нечетких множеств сохраняется для их пересечения, а свойство вогнутости - для их суммы, т.е.

1) если  $A$  и  $B$  - выпуклые нечеткие множества, то  $A \cap B$  - выпуклое нечеткое множество;

2) если  $A$  и  $B$  - вогнутые нечеткие множества, то  $A \cup B$  - вогнутое нечеткое множество.

3. Коммутативность:

$$A \cup B = B \cup A.$$

Порядок следования множеств, участвующих в операции объединения, не влияет на конечный результат.

4. Ассоциативность:

$$A \cup (B \cup C) = (A \cup B) \cup C = A \cup B \cup C$$

Объединение нескольких множеств можно выполнять, последовательно формируя пары множеств, при этом порядок их формирования не является существенным.

4. Идемпотентность:

$$A \cup A = A$$

5. Объединение с пустым множеством 0:

$$A \cup 0 = A$$

6. Поглощение (объединение с универсальным множеством X):

$$A \cup X = X$$

7. Закон исключенного третьего A:

$$A \cup A = X$$

При переходе к нечетким множествам некоторые свойства операции объединения, имеющие место для четких множеств, выполняться не будут.

Множество A дешевых автомобилей и множество B=A комфортабельных автомобилей имеется,  $X_i$  — номер автомобиля:

$$A = \left\{ \frac{1}{x_1}, \frac{0.8}{x_2}, \frac{0.6}{x_3}, \frac{0.4}{x_4}, \frac{0.2}{x_5}, \frac{0}{x_6} \right\}$$

$$B = \left\{ \frac{0}{x_1}, \frac{0.2}{x_2}, \frac{0.4}{x_3}, \frac{0.6}{x_4}, \frac{0.8}{x_5}, \frac{1}{x_6} \right\}$$

Универсальное множество имеет вид

$$X = \left\{ \frac{1}{x_1}, \frac{1}{x_2}, \frac{1}{x_3}, \frac{1}{x_4}, \frac{1}{x_5}, \frac{1}{x_6} \right\}$$

Требуется определить множество  $C = A \cap B$ , содержащее дешевые или комфортабельные автомобили. Если для нахождения объединения A и B использовать оператор типа алгебраической суммы, то результатом будет множество:

$$\mu_C(x) = \mu_A(x) + \mu_B(x) - \mu_A(x) \cdot \mu_B(x),$$

$$C = A \cap B = \left\{ \frac{1}{x_1}, \frac{0.84}{x_2}, \frac{0.76}{x_3}, \frac{0.76}{x_4}, \frac{0.84}{x_5}, \frac{1}{x_6} \right\}$$

Функции принадлежности множеств A, B, C представлены на рис.1.5.7..

Если объединение нечетких множеств  $A \cup A$  определяется на основе оператора алгебраической суммы, то нарушается выполнение свойства 6 данной операции ( $A \text{ и } A = X$ ), всегда справедливое в случае четких множеств.

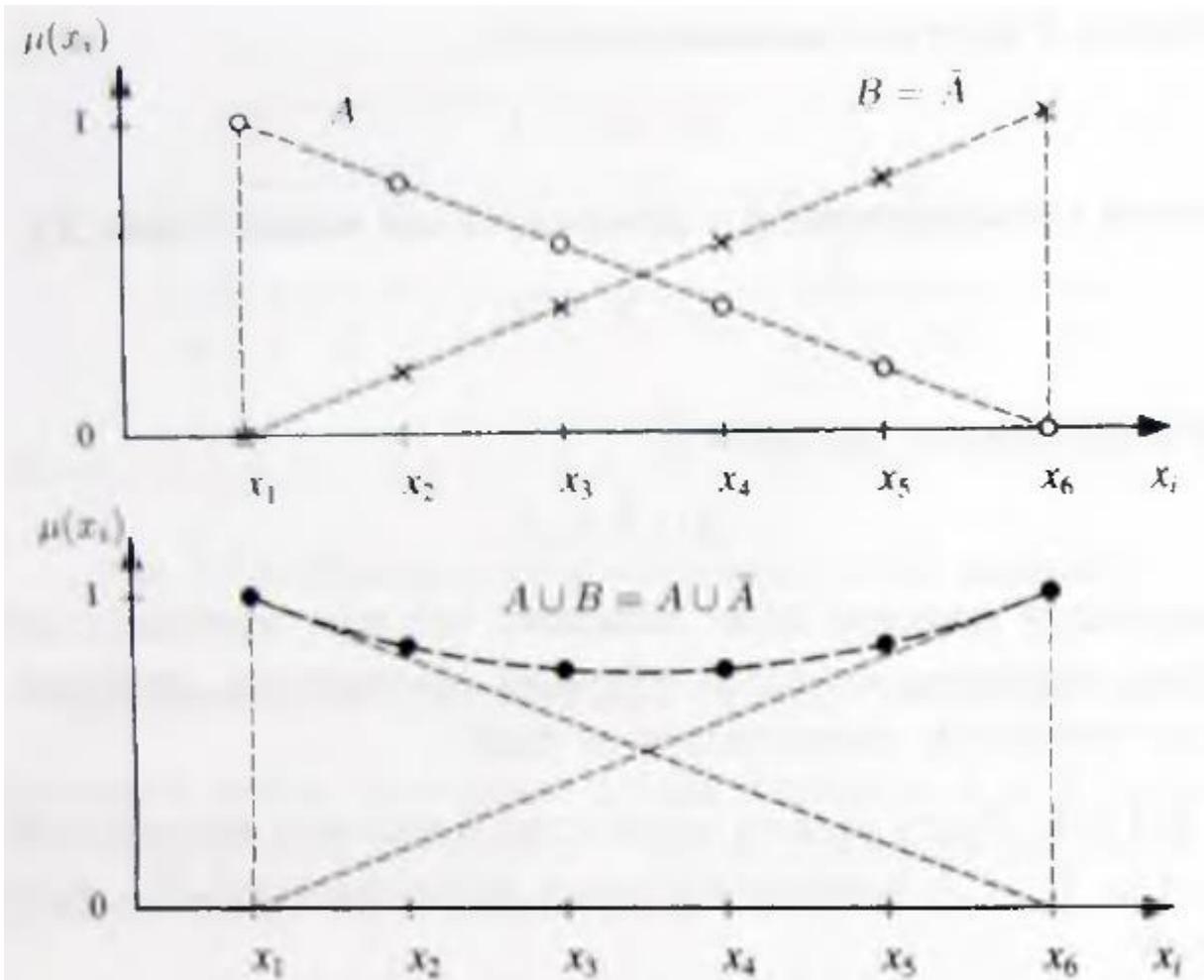


Рис. 1.5.7. Функции принадлежности множеств  $A, B, C$

Первыми операторами, предложенными в качестве основы для выполнения операции объединения нечетких множеств, являлись оператор MAX и алгебраическая сумма. По мере развития нечеткой логики число этих операторов увеличивалось. В настоящее время наиболее распространенными операторами объединения множеств являются t-конормы, также называемые s-нормами.

Оператор s-нормы, или t-конормы представляет собой функцию, реализующую операцию ИЛИ объединения двух нечетких множеств А и В, удовлетворяющую перечисленным ниже свойствам, которые выполняются для всех  $x \in X$ .

**Пространство отображения:**

$$S: [0,1] \times [0,1] \rightarrow [0,1]$$

**Свойство обнуления:**

$$S(0,0) = 0$$

**Случай, когда пара содержит один элемент, для которого  $\mu_{\{x\}} = 0$ :**

$$S(\mu_A\{x\}, 0) = S(0, \mu_A\{x\}) = \mu_A(x)$$

**Свойство коммутативности:**

$$S(\mu_A\{x\}, \mu_B\{x\}) = S(\mu_B\{x\}, \mu_A\{x\}),$$

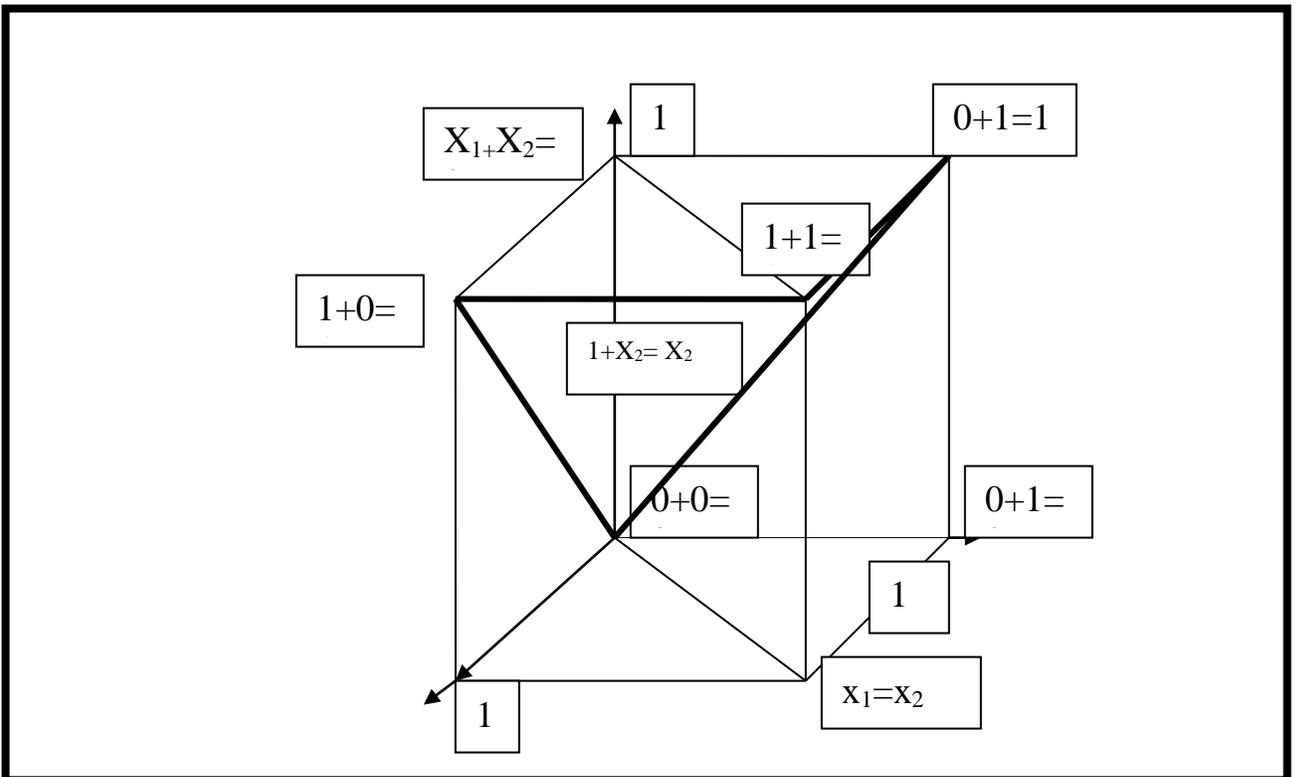


Рис. 1.5.8. Графическое представление S - нормы

**Свойство ассоциативности:**

$$S(\mu_A(x), S(\mu_B(x), \mu_C(x))) = S(S(\mu_A(x), \mu_B(x)), \mu_C(x))$$

**Условие монотонности:**

$$\mu_A(x) \leq \mu_B(x), \mu_C(x) \leq \mu_D(x) \rightarrow S(\mu_A(x), \mu_B(x)) \leq S(\mu_C(x), \mu_D(x))$$

Выделяют параметризованные и непараметризованные операторы s-норм. Результат действия непараметризованных операторов является постоянным.

Конкретные s-нормы различаются по степени качества. Наибольший результат вычислений дает оператор усиленной суммы, наименьший — оператор МАХ. Последовательность s-норм, упорядоченных по степени качества, имеет следующий вид:

усиленная сумма > ограниченная сумма > сумма Эйнштейна > алгебраическая сумма > сумма Гамахера > МАХ.

Вычисление функции принадлежности множества  $A \cup B$  с помощью оператора МАХ приводит к наименьшему результату,

Операторы t-норм и s-норм образуют комплементарные пары, удовлетворяющие условию

$$T[\mu_A(x), \mu_B(x)] = 1 - S[1 - \mu_A(x), 1 - \mu_B(x)]$$

Если задана t-норма, то может быть найдена комплементарная ей s-норма. В табл. 3 приведены комплементарные пары t- и s-норм.

комплементарные пары t- и s-норм. Табл.3

t-норма	комплементарная s-норма
MIN	MAX
алгебраическое произведение	алгебраическая сумма
произведение Гамахера	сумма Гамахера
произведение Эйнштейна	сумма Эйнштейна
усиленное произведение	усиленная сумма
ограниченная разность	ограниченная сумма
параметризованный оператор пересечения Гамахера	параметризованный оператор объединения Гамахера
параметризованный оператор пересечения Ягера	параметризованный оператор объединения Ягера

В качестве основы для операции объединения нечетких множеств можно также использовать оператор алгебраической суммы.

$$\begin{aligned} \mu_{A \cup B}(x) &= \gamma \cdot \text{MAX}[\mu_A(x), \mu_B(x)] + \\ &+ 0.5 \cdot (1 - \gamma) \cdot [\mu_A(x) + \mu_B(x)], \\ \gamma &\in [0, 1], \quad \forall x \in X. \end{aligned}$$

Этот оператор характеризуется наибольшей степенью качества среди всех операторов объединения, а также обладает свойством аддитивности. Результирующая функция принадлежности возрастает пропорционально росту исходных функций, вследствие чего указанный оператор, как и оператор среднего арифметического, можно назвать линейным.

Пусть  $A$  — множество быстрых автомобилей,  $B$  — множество комфортабельных автомобилей,  $x_i$  — обозначение автомобиля:

$$A = \left\{ \frac{1}{x_1}, \frac{1}{x_2}, \frac{1}{x_3}, \frac{1}{x_4} \right\},$$

$$B = \left\{ \frac{0.6}{x_1}, \frac{0.7}{x_2}, \frac{0.9}{x_3}, \frac{1}{x_4} \right\}$$

Нужно приобрести быстрый ИЛИ комфортабельный автомобиль. Применяя s-норму типа MAX, приходим к результату:

$$C = A \cup B = \left\{ \frac{1}{x_1}, \frac{1}{x_2}, \frac{1}{x_3}, \frac{1}{x_4} \right\} \cup \left\{ \frac{0.6}{x_1}, \frac{0.7}{x_2}, \frac{0.9}{x_3}, \frac{1}{x_4} \right\} = \left\{ \frac{1}{x_1}, \frac{1}{x_2}, \frac{1}{x_3}, \frac{1}{x_4} \right\}.$$

Оператор MAX показал возможность приобретения любого из автомобилей, поскольку степени их принадлежности множеству «быстрый ИЛИ комфортабельный» одинаковы и равны 1. При использовании оператора арифметической суммы множества A и B на первом шаге преобразуются в нечеткий набор, а затем, на втором шаге нечеткое множество.

Шаг 1

$$A \cup B = \left\{ \frac{1}{x_1}, \frac{1}{x_2}, \frac{1}{x_3}, \frac{1}{x_4} \right\} \cup \left\{ \frac{0.6}{x_1}, \frac{0.7}{x_2}, \frac{0.9}{x_3}, \frac{1}{x_4} \right\} = \left\{ \frac{1}{x_1}, \frac{0.6}{x_1}, \frac{1}{x_2}, \frac{0.7}{x_2}, \frac{1}{x_3}, \frac{0.9}{x_3}, \frac{1}{x_4}, \frac{1}{x_4} \right\}.$$

Шаг 2

$$A \cup B = \left\{ \frac{1.6}{x_1}, \frac{1.7}{x_2}, \frac{1.9}{x_3}, \frac{2}{x_4} \right\}$$

Оператор арифметической суммы дает информацию о том, что следует приобрести автомобиль X4, степень принадлежности которого набору «быстрый или комфортабельный» является наибольшей.

## 1.6. Нечеткие отношения

Понятие нечеткого отношения является основным понятием нечеткой логики. Эти отношения позволяют представлять неточные утверждения типа « $x$  почти равно  $y$ » или « $x$  значительно больше чем  $y$ ».

Нечеткое отношение  $R$  (relation) между двумя непустыми множествами  $\mathbf{X}$  и  $\mathbf{Y}$  называют нечеткое множество, определенное на декартовом произведении  $\mathbf{X} \times \mathbf{Y}$ , т.е.

$$R \subseteq \mathbf{X} \times \mathbf{Y} = \{(x, y) : x \in \mathbf{X}, y \in \mathbf{Y}\}.$$

также, нечеткое отношение это множество пар

$$R = \{((x, y), R(x, y))\},$$

где

$$\mu_R : \mathbf{X} \times \mathbf{Y} [0, 1]$$

- это функция принадлежности, приписывающая каждой паре  $(x, y)$  ее степень принадлежности  $\mu_R(x, y)$ , определяемую как силу связи между элементами  $x \in \mathbf{X}$  и  $y \in \mathbf{Y}$ . Нечеткое отношение можно представить в виде

$$R = \sum_{\mathbf{X} \times \mathbf{Y}} \frac{\mu_R(x, y)}{(x, y)} \quad \text{или}$$

$$R = \int_{\mathbf{X} \times \mathbf{Y}} \frac{\mu_R(x, y)}{(x, y)}.$$

Неточное утверждения « $y$  примерно равно  $x$ » имеет  $\mathbf{X} = \{3, 4, 5\}$  и  $\mathbf{Y} = \{4, 5, 6\}$ .

Отношение  $R$  определяется:

$$R = \frac{1}{(4, 4)} + \frac{1}{(5, 5)} + \frac{0,8}{(3, 4)} + \frac{0,8}{(4, 5)} + \frac{0,8}{(5, 4)} + \frac{0,8}{(5, 6)} + \frac{0,6}{(3, 5)} + \frac{0,6}{(4, 6)} + \frac{0,4}{(3, 6)}.$$

Функция принадлежности  $\mu_R(x, y)$  отношения  $R$  имеет вид

$$\mu_R(x, y) = \begin{cases} 1, & \text{если } x = y, \\ 0,8, & \text{если } |x - y| = 1, \\ 0,6, & \text{если } |x - y| = 2, \\ 0,4, & \text{если } |x - y| = 3. \end{cases}$$

Отношение  $R$  задаем в виде матрицы

$$\begin{array}{c} \begin{array}{ccc} \mathcal{X}_1 & \mathcal{Y}_2 & \mathcal{Y}_3 \end{array} \\ \begin{array}{l} x_1 \\ x_2 \\ x_3 \end{array} \begin{bmatrix} 0,8 & 0,6 & 0,4 \\ 1 & 0,8 & 0,6 \\ 0,8 & 1 & 0,8 \end{bmatrix}, \end{array}$$

где  $x_1 = 3$ ,  $x_2 = 4$ ,  $x_3 = 5$ , а  $y_1 = 4$ ,  $y_2 = 5$ ,  $y_3 = 6$ .

Область  $\mathbf{X} = \mathbf{Y} = [0, 120]$  представляет длительность жизни человека и отношение  $R$  с функцией принадлежности

$$\mu_R(x, y) = \begin{cases} 0, & \text{если } x - y \leq 0, \\ \frac{x - y}{30}, & \text{если } 0 < x - y < 30, \\ 1, & \text{если } x - y \geq 30 \end{cases}$$

представляет неточное утверждение «дама  $x$  намного старше дамы  $y$ ».

Нечеткое отношение  $R$  - это нечеткое множество, поэтому сохраняют силу операции пересечения, суммирования и дополнения:

$$\mu_{R \cap S}(x, y) = \min(\mu_R(x, y), \mu_S(x, y)),$$

$$\mu_{R \cup S}(x, y) = \max(\mu_R(x, y), \mu_S(x, y)),$$

$$\mu_{\bar{R}}(x, y) = 1 - \mu_R(x, y).$$

Для нечеткой логики важны комбинации двух нечетких отношений. Рассмотрим три четких множества  $\mathbf{X}$ ,  $\mathbf{Y}$ ,  $\mathbf{Z}$  и два нечетких отношения  $R \subseteq \mathbf{X} \times \mathbf{Y}$  и  $S \subseteq \mathbf{Y} \times \mathbf{Z}$  с функциями принадлежности  $\mu_R(x, y)$  и  $\mu_S(y, z)$ .

Комбинацией типа  $\text{sup-}T$  нечетких отношений  $R \subseteq \mathbf{X} \times \mathbf{Y}$  и  $S \subseteq \mathbf{Y} \times \mathbf{Z}$  называется нечеткое отношение  $R \circ S \subseteq \mathbf{X} \times \mathbf{Z}$  с функцией принадлежности

$$\mu_{R \circ S}(x, z) = \sup_{y \in \mathbf{Y}} \left\{ \left[ \mu_R(x, y)^T * \mu_S(y, z) \right] \right\}.$$

Форма функции принадлежности  $\mu_{R \circ S}(x, z)$  комбинации  $R \circ S$  зависит от  $T$ -нормы. Если в качестве  $T$ -нормы применяется  $\min$ , ( $T(a, b) = \min(a, b)$ ), то равенство можно представить в виде

$$\mu_{R \circ S}(x, z) = \sup_{y \in \mathbf{Y}} \left\{ \min \left[ \mu_R(x, y), \mu_S(y, z) \right] \right\}.$$

Это «комбинация типа  $\text{sup-min}$ ».

Если множество  $\mathbf{Y}$  имеет конечное количество элементов, то комбинация типа  $\text{sup-min}$  сводится к комбинации типа  $\text{max-min}$  в форме

$$\mu_{R \circ S}(x, z) = \max_{y \in \mathbf{Y}} \left\{ \min \left[ \mu_R(x, y), \mu_S(y, z) \right] \right\}.$$

Если отношения  $R$  и  $S$  представлены матрицами

$$R = \begin{bmatrix} 0,2 & 0,5 \\ 0,6 & 1 \end{bmatrix}, \quad S = \begin{bmatrix} 0,3 & 0,6 & 0,8 \\ 0,7 & 0,9 & 0,4 \end{bmatrix},$$

причем  $\mathbf{X} = \{x_1, x_2\}$ ,  $\mathbf{Y} = \{y_1, y_2\}$ ,  $\mathbf{Z} = \{z_1, z_2\}$ . Комбинация типа max-min отношений  $R$  и  $S$  имеет вид

$$Q = R \circ S = \begin{bmatrix} 0,2 & 0,5 \\ 0,6 & 1 \end{bmatrix} \circ \begin{bmatrix} 0,3 & 0,6 & 0,8 \\ 0,7 & 0,9 & 0,4 \end{bmatrix} = \begin{bmatrix} q_{11} & q_{12} & q_{13} \\ q_{21} & q_{22} & q_{23} \end{bmatrix},$$

где

$$q_{11} = \max[\min(0,2;0,3), \min(0,5;0,7)] = 0,5,$$

$$q_{12} = \max[\min(0,2;0,6), \min(0,5;0,9)] = 0,5,$$

$$q_{13} = \max[\min(0,2;0,8), \min(0,5;0,4)] = 0,4,$$

$$q_{21} = \max[\min(0,6;0,3), \min(1;0,7)] = 0,7,$$

$$q_{22} = \max[\min(0,6;0,6), \min(1;0,9)] = 0,9,$$

$$q_{23} = \max[\min(0,6;0,8), \min(1;0,4)] = 0,6.$$

Поэтому

$$Q = \begin{bmatrix} 0,5 & 0,5 & 0,4 \\ 0,7 & 0,9 & 0,6 \end{bmatrix}.$$

Для практики нужны комбинации нечеткого множества с нечетким отношением. Рассмотрим нечеткое множество  $A \subseteq \mathbf{X}$  и нечеткое отношение  $R \subseteq \mathbf{X} \times \mathbf{Y}$  с функциями принадлежности  $\mu_A(x)$  и  $\mu_R(x,y)$ , обозначаемое  $A \circ R$  и определяемое как нечеткое множество  $B \subseteq \mathbf{Y}$ .

$$B = A \circ R$$

с функцией принадлежности

$$\mu_B(y) = \sup_{x \in X} \left\{ \left[ \mu_A(x)^T * \mu_R(x, y) \right] \right\}.$$

Форма записи этого выражения зависит от используемой  $T$ -нормы и от свойств множества  $X$ . Возможны 4 случая:

1) при  $T(a, b) = \min(a, b)$  комбинация типа sup-min

$$\mu_B(y) = \sup_{x \in X} \left\{ \min [\mu_A(x), \mu_R(x, y)] \right\},$$

2) при  $T(a, b) = \min(a, b)$  и  $X$  множество с конечным количеством элементов, и комбинация типа max-min

$$\mu_B(y) = \max_{x \in X} \left\{ \min [\mu_A(x), \mu_R(x, y)] \right\},$$

3) при  $T(a, b) = a \cdot b$  комбинация типа sup-произведение (sup-prod)

$$\mu_B(y) = \sup_{x \in X} \left\{ \mu_A(x) \cdot \mu_R(x, y) \right\},$$

4) при  $T(a, b) = a \cdot b$  и  $X$  множество с конечным количеством элементов и комбинация типа max-произведение (max-prod)

$$\mu_B(y) = \max_{x \in X} \left\{ \mu_A(x) \cdot \mu_R(x, y) \right\}.$$

Для  $X = \{x_1, x_2, x_3\}$ ,  $Y = \{y_1, y_2\}$  и  $A$  имеет вид

$$A = \frac{0,4}{x_1} + \frac{1}{x_2} + \frac{0,6}{x_3},$$

отношение  $R$  представлено матрицей

$$R = \begin{matrix} & y_1 & y_2 \\ x_1 & \begin{bmatrix} 0,5 & 0,7 \end{bmatrix} \\ x_2 & \begin{bmatrix} 0,2 & 1 \end{bmatrix} \\ x_3 & \begin{bmatrix} 0,9 & 0,3 \end{bmatrix} \end{matrix}.$$

Комбинацию  $A \circ R$  получаем для max-min в виде нечеткого множества  $B$

$$B = \frac{\mu_B(y_1)}{y_1} + \frac{\mu_B(y_2)}{y_2},$$

причем

$$\mu_B(y_1) = \max \{ \min(0,4; 0,5), \min(1; 0,2), \min(0,6; 0,9) \} = 0,6,$$

$$\mu_B(y_2) = \max \{ \min(0,4; 0,7), \min(1; 1), \min(0,6; 0,3) \} = 1.$$

Поэтому

$$B = \frac{0,6}{y_1} + \frac{1}{y_2}.$$

### 1.7. Нечеткая импликация

Импликацией называется вид отношения, имеющего форму правила, используемого при рассуждениях. Различают классическую и нечеткую импликацию.

Классическая импликация выражается с помощью соотношения

ЕСЛИ А ТО В

Сокращенная ее форма имеет вид:

$$A \rightarrow B,$$

где  $A$  — утверждение, называемое антецедентом (условием),  $B$  - утверждение, называемое консеквентом (заключением, результатом).

Утверждения в классической логике могут быть абсолютно истинными ( $A = 1$ ,  $B = 1$ ) либо ложными ( $A = 0$ ,  $B = 0$ ). Истинность или ложность импликации зависит от конкретных значений  $\mu_A$  и  $\mu_B$  (истинности антецедента и консеквента). Значение истинности импликации определяется ее функцией принадлежности  $\mu_{A \rightarrow B}$ , принимающей только два значения 0 и 1. Функция принадлежности классической импликации задается в форме таблицы истинности и вычисляется по выражению

$$\mu_{A \rightarrow B}(x, y) = \text{MAX}(1 - \mu_A, \mu_B)$$

Функция принадлежности классической импликации  $\mu_{A \rightarrow B}$

$\mu_p$	$\mu_q$	$\mu_{p \rightarrow q}$
1	1	1
1	0	0
0	1	1
0	0	1

Оператор классической импликации имеет ряд свойств, затрудняющих его использование в нечеткой логике.

Импликация следующего вида:

ЕСЛИ (состояние автомобиля  $A = \text{новый}$ ), ТО (расход топлива  $B = \text{малый}$ )

Область значений  $A$  переменной «состояние автомобиля» имеет бинарную форму представления (новый:  $x = 1$ , старый:  $x = 0$ ). Аналогичным образом задана область значений  $B$  переменной «расход топлива» (малый:  $y = 1$ , большой:  $y = 0$ ).

Утверждение (состояние автомобиля = новый) = А является антецедентом,  
 Утверждение (расход топлива == малый) = В является консеквентом.

Заменяя лингвистические значения (новый, старый) на  $x$  и значения (малый, большой) на  $y$ , получаем четыре возможных состояния  $S_i$  импликации.

S1: ЕСЛИ (состояние автомобиля = новый) ТО (расход топлива = малый),

$$\mu_A = 1, \mu_B = 1, \mu_{A \rightarrow B} = 1$$

При  $x = \text{новый}$ ,  $y = \text{малый}$  импликация является истинной.

S2: ЕСЛИ (состояние автомобиля = новый) ТО (расход топлива = большой),

$$\mu_A = 1, \mu_B = 0, \mu_{A \rightarrow B} = 0$$

При  $x = \text{новый}$  и  $y = \text{большой}$  импликация является ложной.

Условие (состояние автомобиля = новый) не изменилось, и потому изменившееся заключение (расход топлива = большой) не может быть истинным.

S3: ЕСЛИ (состояние автомобиля = старый) ТО (расход топлива = малый),

$$\mu_A = 0, \mu_B = 1, \mu_{A \rightarrow B} = 1$$

При  $x = \text{старый}$  и  $y = \text{малый}$  импликация является истинной. Эта импликация касается только факта (состояние автомобиля = новый), не говоря ничего о противоположном ему факте (состояние автомобиля = старый). Согласно классической логике в данном случае могут быть истинными как заключение (расход топлива = малый), так и заключение (расход топлива = большой), входящее в S4.

S4: ЕСЛИ (состояние автомобиля = старый), ТО (расход топлива = большой),

$\mu_A = 1, \mu_B = 1, \mu_{A \rightarrow B} = 1$ . Импликация для данных значений  $x, y$  является истинной также, как и в S3.

Недостаток оператора классической импликации состоит в том, что если условие вообще не выполняется ( $\mu_B = 0$ ), то импликация является истинной, и это приводит к взаимно исключающим выводам (расход топлива = малый) и (расход топлива = большой).

В случае нечетких систем, при одновременной активации множества нечетких правил, использование оператора классической импликации оказывает неправильное представление. Поэтому нужны однозначные операции.

Нечеткая импликация представляет собой правило, простейшая форма которого выражается в виде

ЕСЛИ  $(x = A)$  ТО  $(y = B)$ ,

где  $(x = A)$  — условие (антецедент), а  $(y = B)$  — заключение (консеквент).

Здесь  $A$  и  $B$  — нечеткие множества, заданные своими функциями принадлежности  $\mu_A$ ,  $\mu_B$  и областями определения  $X$  и  $Y$  соответственно.

Обозначение нечеткой импликации имеет вид:

$A \rightarrow B$ .

Различие между классической и нечеткой импликацией состоит в том, что в случае классической импликации условие и заключение могут быть либо абсолютно истинными, либо абсолютно ложными, в то время как для нечеткой импликации допускается истинность, со значением, в непрерывном интервале  $[0,1]$ . В практике редко встречаются ситуации, когда условия правил удовлетворяются полностью, и по этой причине нельзя полагать, что заключение абсолютно истинно. Нечеткая импликация (нечеткое отношение) задается функцией принадлежности, область определения которой является декартовым произведением  $X \times Y$  соответствующих областей условия и заключения.

Функция принадлежности импликации есть основа для нечетких выводов при вычислении выходного значения нечеткой модели (регулятора) с заданными входными значениями. Формирование нечеткого вывода, используя функции принадлежности условия  $\mu_A(x)$  и заключения  $\mu_B(y)$ , осуществляют оператором импликации. Это может быть оператор импликации Мамдани, основанный на

предположении, что степень истинности заключения  $\mu_B(y)$  не может быть выше, чем степень выполнения условия  $\mu_A(x)$ :

$$\mu_{A \rightarrow B}(x, y) = \text{MIN}(\mu_A(x), \mu_B(y))$$

Для правила ЕСЛИ (состояние автомобиля = новый, ТО (расход топлива = малый) И, если автомобиль не является новым, то расход топлива у него не может быть низким, как у нового автомобиля.

В нечетком выводе также используется оператор алгебраического произведения PROD:

$$\mu_{A \rightarrow B}(x, y) = (\mu_A(x) * \mu_B(y))$$

Исследованы и другие операторы, результаты применения которых зависят от конкретной задачи (табл. 1.6.1). Оператор Лукасевича имеет наилучшие характеристики по определенному набору критериев. Остальные операторы, приведенные в табл. расположены по убыванию степени удовлетворения этим критериям.

Табл. 1.6.1

импликация Лукасевича	$\text{MIN}(1, 1 - \mu_A(x) + \mu_B(y))$
импликация Клини--Динса	$\text{MAX}(1 - \mu_A(x), \mu_B(y))$
импликация Клини--Динса--Лукасевича	$1 - \mu_A(x) + \mu_A(x) \cdot \mu_B(y)$
импликация Гёделя	$\begin{cases} 1 & \text{для } \mu_A(x) \leq \mu_B(y) \\ \mu_B(y) & \text{в других случаях} \end{cases}$
импликация Ягера	$(\mu_A(x))^{\mu_B(y)}$
импликация Заде	$\text{MAX}(1 - \mu_A(x), \text{MIN}(\mu_A(x), \mu_B(y)))$

Возьмем нечеткую импликацию:

ЕСЛИ (состояние автомобилях = новый), ТО (расход топлива = малый),

где нечеткие множества «новый» и «малый» заданы функциями принадлежности  $\mu_A$  и  $\mu_B$ , представленными на рис. 1.6..1.

Функция принадлежности импликации  $\mu_{A \rightarrow B}(x,y)$  представлена на рис. 4.34. Используя подходящий метод вывода и имея определенное значение  $x_0$  переменной  $x$ , содержащейся в условии правила, можно определить функцию принадлежности заключения  $\mu_B(y)$ , которую затем можно использовать для вычисления четкого значения  $y$  на выходе нечеткой модели (рис. 1.6.1).

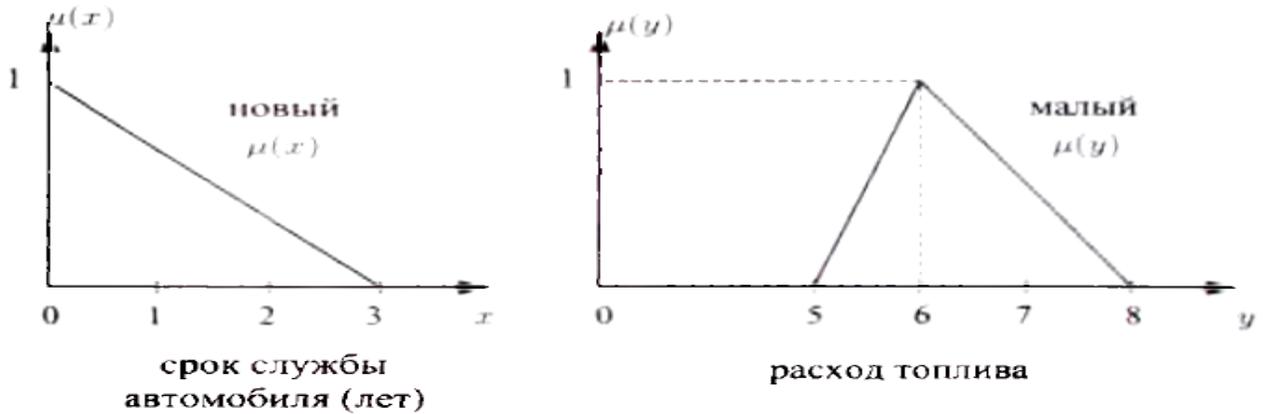


Рис. 1.6.1. Функции принадлежности нечетких множеств «новый» и «малый», содержащихся в условии и заключении функции принадлежности импликации  $\mu_{A \rightarrow B}(x,y)$  с использованием оператора Мамдани.

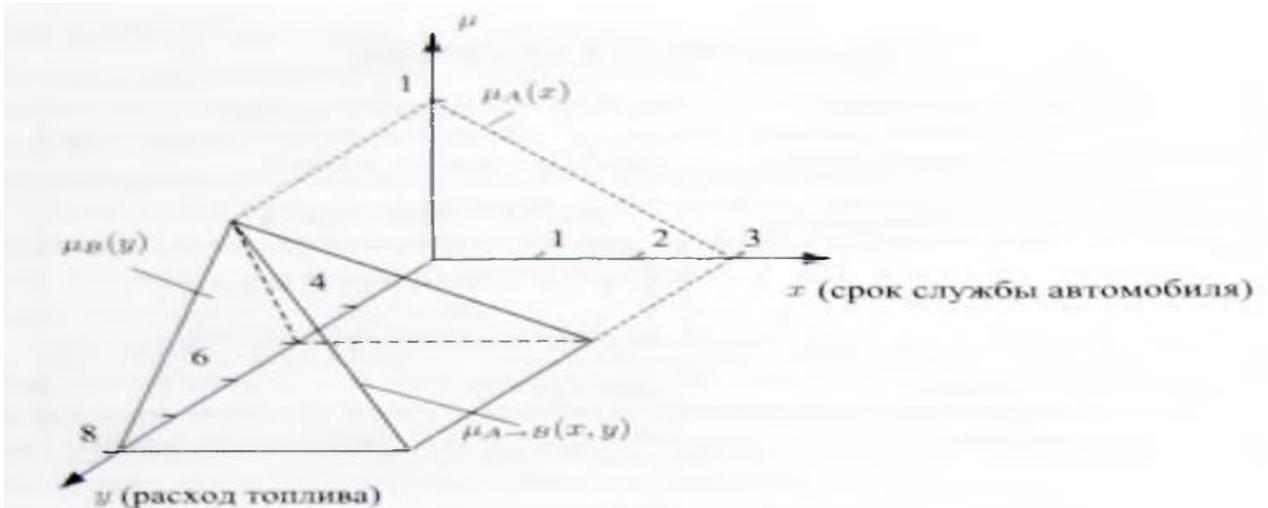


Рис. 1.6.2. Функция принадлежности импликации (оператора Мамдани )

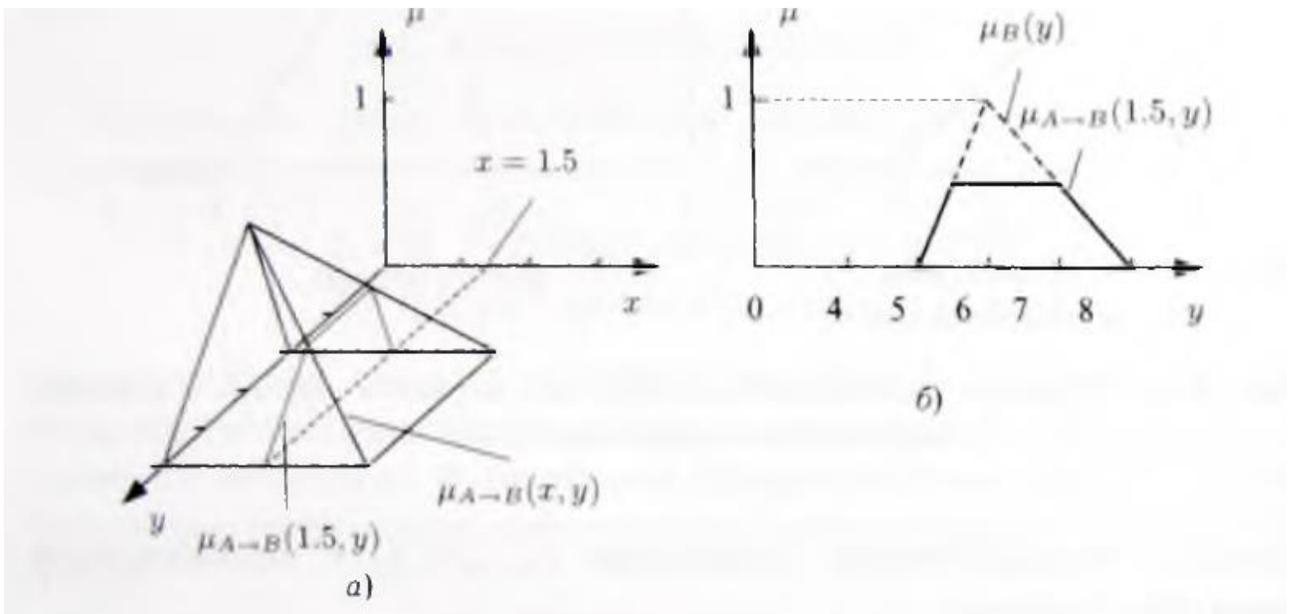


Рис. 1.6.3. Функция принадлежности импликации  $\mu_{A \rightarrow B}(x, y)$  для заданного значения переменной  $x = 1.5$  (а) и ее проекция на плоскость  $\{\mu, y\}$  (б)

## Глава 2. Нечеткий вывод и нечеткие модели

### 2.1. Структура, основные элементы и операции в нечетких моделях

Классическая логика, оперирует только понятиями «истина» или «ложь», и не решает проблему неопределенности при выводах в человеческих суждениях. Выразить неопределенность позволяет логическая система, в которой, кроме понятий «истина» или «ложь», используются другие значения истинности. Такую систему предложил польский математик Ян Лукасевич с тремя значениями истинности: «0-ложь», «1-истина», «0,5-возможно». Затем появилась нечеткая логика с интервалом значений от 0 до 1.

Мышление человека основан на нечеткой логике и вместо чисел используется классы объектов в виде нечетких множеств с нечеткой истинностью, нечеткими связями и нечеткими правилами вывода. Нечеткая логика имеют следующие отличительные черты:

- использование лингвистических переменных вместо числовых переменных или в дополнение к ним;
- простые отношения между переменными описываются с помощью нечетких высказываний;
- сложные отношения между переменными описываются нечеткими алгоритмами.

Применение нечетких моделей целесообразно в случаях, когда имеются недостаточность и неопределенность знаний об исследуемой системе,

Например, если:

- получение информации сложно или невозможно,
- источником основной информации являются экспертные данные, эвристические описания процессов функционирования,
- информация о системе разнокачественная или оценка параметров проводится с использованием разных шкал.

Нечеткая логика использует обобщение направлений классической логики и нечетких множеств, поэтому в интеллектуальных системах применяются различные модели нечеткого логического вывода.

Существуют нечеткие модели логического вывода:

- модель нечеткого логического вывода, представленная в виде сопоставления четкого набора нечетких ситуаций (описываемых кортежами нечетких переменных) и соответствующих принимаемых решений.

- модель нечеткого логического вывода построенная на нечетких отношениях при прямом произведении множества правил нечеткого логического вывода и нечеткого множества принимаемых решений.

- модель, представленную экспертным путем в виде нечетких эталонных ситуаций, связанных с определенными решениями (ситуационное управление).

- модель, основанная на нечетких (приближенных) высказываниях (суждениях), нечетком выводе и построении логико-лингвистических моделей на основе композиционного правила Заде.

Под приближенными рассуждениями понимается процедура, при которой из некоторых посылок получают некоторые следствия, возможно также нечеткие.

Приближенные рассуждения лежат в основе способностей человека понимать естественный язык, играть в шахматы, разбирать почерк, требующие умственных усилий в неполностью определенной среде.

В классической двухзначной логике приближенные рассуждения основаны на следующих тавтологиях.

- модус поненс:  $(A \wedge (A \Rightarrow B)) \Rightarrow B$ ;
- модус толленс:  $((A \Rightarrow B) \wedge \bar{B}) \Rightarrow \bar{A}$ ;
- силлогизм:  $((A \Rightarrow B) \wedge (B \Rightarrow C)) \Rightarrow (A \Rightarrow C)$ ;
- контрапозиция:  $(A \Rightarrow B) \Rightarrow (\bar{B} \Rightarrow \bar{A})$ .

## Основные правила вывода в двоичной логике

В четкой (двоичной) логике решения об истинности одних суждений выводятся на основании истинности других суждений. Такой вывод представляется в виде схемы: над горизонтальной чертой записываются все суждения, на основании которых принимается решение, а под чертой - полученный результат. Схема корректного вывода обладает тем свойством, что поскольку истинны все суждения над чертой, то истинно также и суждение под чертой, так как из истинных суждений может выводиться только истинный результат.

Пусть  $A$  и  $B$  - это суждения (а не нечеткие множества), причем запись  $A=1$  ( $B=1$ ) означает, что логическим значением суждения  $A$  ( $B$ ) считается «истина», тогда как запись  $A=0$  ( $B=0$ ) означает, что логическим значением суждения  $A$  ( $B$ ) считается «ложь». Два правила вывода часто применяются и в двоичной логике.

Правило вывода modus ponens

Условие	$A$	$A$ есть истинно
Импликация	$A \rightarrow B$	Если $A$ , то $B$ – истинно
Вывод	$B$	$B$ есть истинно

Пусть суждение  $A$  имеет вид «Ян является водителем», а суждение  $B$  - «У Яна есть водительское удостоверение». В соответствии с правилом modus ponens, если  $A=1$ , то и  $B=1$ , поскольку если истинно, что «Ян является водителем», то также истинно, что «У Яна есть водительское удостоверение». Из истинности предпосылки и импликации (суждения над чертой) следует истинность вывода (суждения под чертой).

## Правило вывода modus tollens

Условие	$\bar{B}$	(3.188)
Импликация	$A \rightarrow B$	
Вывод	$\bar{A}$	

Понимается, что если «У Яна нет водительского удостоверения», т.е.  $B = 0$  ( $\bar{B} = 1$ ), то «Ян не является водителем», т.е.  $A = 0$  ( $\bar{A} = 1$ ). В этом примере тоже из истинности предпосылки и импликации следует истинность вывода.

Вместе с тем имеется ограничение: если высказывание  $A^*$  близкое к  $A$  (непрерывные значения в интервале 0 - 1) является истинным, и если прецедент отсутствует, то модус поненс не сможет вывести никакого, приближенного заключения. Даже, при близком к  $A$  утверждении, что  $A'$  является истинным, модус поненс не может быть применен. Поэтому возможным способом вывода при неопределенной информации является применение нечеткого логического вывода. Нечетким логическим выводом называется получение заключения в виде нечеткого множества, соответствующего текущим значениям входов, с использованием нечеткой базы знаний и нечетких операций. Поэтому в нечеткой логике применяется композиционное правило Заде. (см 1.7. Суждения, присутствующие в правилах modus ponens и modus tollens, представляются нечеткими множествами.

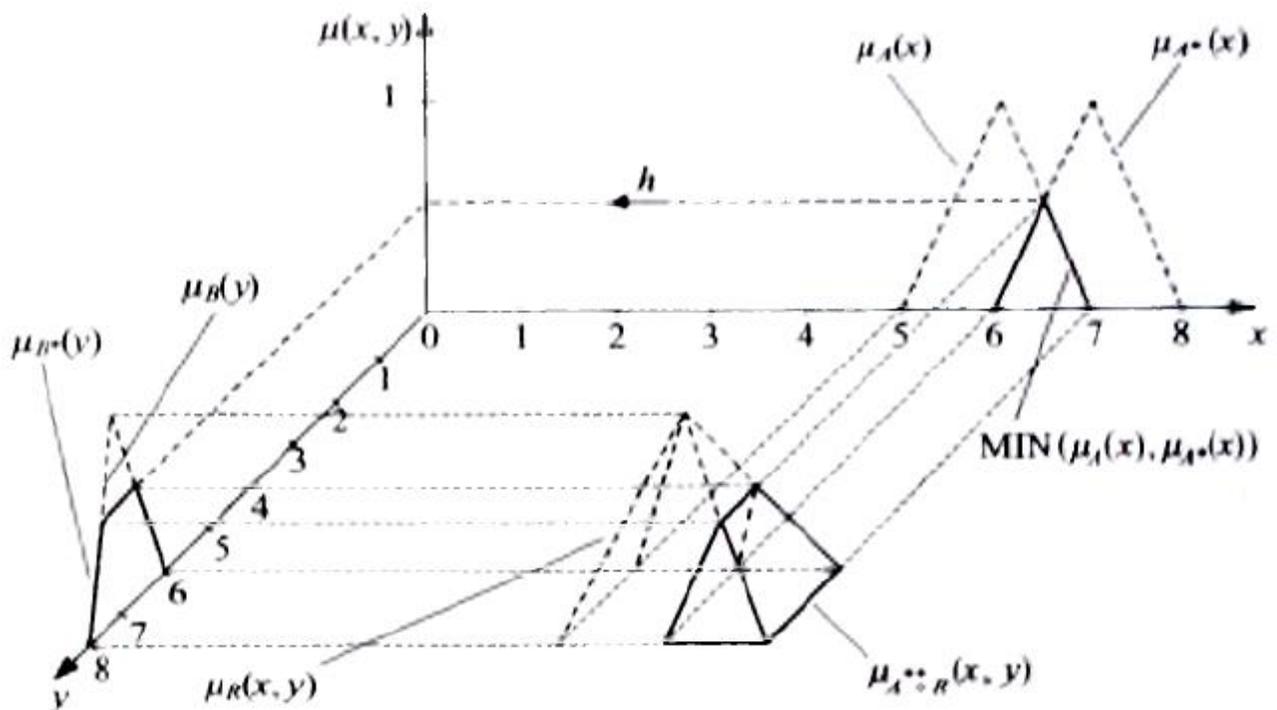
Пусть  $A$  и  $B$  два универсальных множества с базовыми переменными  $x$  и  $y$  соответственно,  $R$  — нечеткое отношение двух аргументов, заданное на области определения  $X$  и  $Y$ . Результатом композиции  $A^*$  и  $R$  (обозначается  $A^* \circ$

R) является нечеткое множество  $B^*$  с областью определения  $Y$  и функцией принадлежности, имеющей вид:

$$\mu_{B^*}(y) = \text{MAX}_{x \in X} \text{MIN}(\mu_{A^*}(x), \mu_R(x, y)), \quad 2.1.$$

где  $R(x, y)$  — продолжение множества  $A^*(x)$  на область определения  $X \times Y$ . Операция  $\text{MIN}$  -  $t$  - норма, а операция  $\text{MAX}$  - Это  $S$  - норма. (Рис 2.1.1.)

В результате композиции отношения  $R(x, y)$  и нечеткого множества  $A^*$ , получается нечеткое множество  $A^* \circ R(x, y)$ , функция принадлежности которого имеет вид поверхности в трехмерном пространстве.



$$h = \text{MAX}_{x \in X} \text{MIN}(\mu_A(x), \mu_{A^*}(x))$$

Рис. 2.1.1. Нечеткий вывод на основе правила ЕСЛИ ( $x = A$ ) ТО ( $y = B$ ) в случае, когда входным параметром  $x$  модели является нечеткое множество  $A^*(x)$

Проекция композиции  $A^* \circ R$  на область определения  $Y$  приводит к получению модифицированного нечеткого множества  $B^*$ , соответствующего заключению

используемого правила. Функция принадлежности этого множества имеет вид (2.1) и представляет собой функцию  $\mu_B(y)$ , высота которой ограничена значением  $h$ , выражающим степень выполнения условия принадлежности множеству.

Пусть  $X$  и  $Y = \{1,2,3,4\}$

Нечеткое множество  $A = \text{“малый”}$  задается таблицей

Элемент множества $A$	1	2	3	4
$\mu_A(x)$	1	0.6	0.2	0

Нечеткое множество  $R = \text{“примерно равны”}$  представлено матрицей

$x \backslash y$	1	2	3	4
1	1	0.5	0.5	0
2	0.5		0.5	0
3	0	0.5	1	0.5
4	0		0.5	1

Тогда, согласно композиционному правилу,  $B=A \circ R$

$$[1 \ 0.6 \ 0.2 \ 0] \circ \begin{array}{c} \left| \begin{array}{cccc} 1 & 0.5 & 0 & 0 \\ 0.5 & 1 & 0.5 & 0 \\ 0 & 0.5 & 1 & 0.5 \\ 0 & 0 & 0.5 & 1 \end{array} \right| = [1 \ 0.6 \ 0.5 \ 0.2]$$

То есть  $B$  представляет нечеткое множество

Элемент множества $B$	1	2	3	4
$\mu_B(y)$	1	0.6	0.5	0.2

Результат объясняется следующим образом:

$B =$  “более и или менее малый”, если терм “более и или менее” представляется, как оператор увеличения нечеткости

$x$  - малый условие

$x, y$  - примерно равны условие

---

$y$  – более и или менее малый нечеткий вывод

### Обобщенное (нечеткое) правило вывода *modus ponens*

Условие	$x$ это $A'$
Импликация	IF $x$ это $A$ THEN $y$ это $B$
Вывод	$y$ это $B'$

где  $A, A' \subseteq X$  и  $B, B' \subseteq X$  - нечеткие множества, а  $x$  и  $y$  - лингвистические переменные типа «малая скорость», «умеренная температура» или «молодой человек». Формализуются такие переменные приписыванием им некоторых нечетких множеств. Схема вывода с лингвистическими переменными:

Условие	Скорость автомобиля большая
Импликация	Если скорость автомобиля очень большая, то уровень шума высокий
Вывод	Уровень шума в автомобиле не очень высокий

В этой схеме условие, импликация и вывод - неточные утверждения. В качестве лингвистических переменных выделим:  $x$  - скорость автомобиля,  $y$  - уровень шума.

Терм  $T_1 = \{\text{«малая»}, \text{«средняя»}, \text{«большая»}, \text{«очень большая»}\}$

- множество значений лингвистической переменной  $x$ .

Терм  $T_2 = \{\text{«малый»}, \text{«средний»}, \text{«не очень высокий»}, \text{«высокий»}\}$

- множество значений лингвистической переменной  $y$ .

Каждому терму  $T_1$  и  $T_2$  соответствует нечеткое множество. Из анализа этих схем вывода получаем нечеткие множества:

$A = \text{«очень большая скорость автомобиля»},$

$A' = \text{«большая скорость автомобиля»}$  и  $B = \text{«высокий уровень шума»},$

$B' = \text{«не очень высокий уровень шума»}.$

Различия между четким и нечетким правилами следующие (см также раздел 1.7). В обоих случаях импликация имеет один и тот же вид  $A \rightarrow B$ , где  $A$  и  $B$  - это суждения (правило) либо нечеткие множества. Суждение  $A$  из импликации четкого правила также присутствует в предпосылке этого правила. В то же время условие нечеткого правила не связано с нечетким множеством  $A$ , но содержит некоторое нечеткое множество  $A'$ , которое может в определенном смысле быть близким к  $A$ , однако не обязательно  $A = A'$ .

Нечеткое множество  $A = \text{«очень большая скорость автомобиля»}$  не равно нечеткому множеству  $A' = \text{«большая скорость автомобиля»}$  и выводы схем отличаются друг от друга. Вывод нечеткого правила относится к нечеткому множеству  $B'$ , определяемое композицией нечеткого множества  $A'$  и нечеткой импликации  $A \rightarrow B$ , т.е.

$$B' = A' \circ (A \rightarrow B)$$

Нечеткая импликация  $A \rightarrow B$  есть нечеткое отношению  $R \subseteq \mathbf{X} \times \mathbf{Y}$  с функцией принадлежности  $\mu_R(x, y)$ , а функцию принадлежности нечеткого множества  $B'$  представляют выражением

$$\mu_{B'}(y) = \sup_{x \in \mathbf{X}} \left\{ \mu_{A'}(x) \overset{T}{*} \mu_{A \rightarrow B}(x, y) \right\}, \quad \text{где}$$

$\mu_{A \rightarrow B}(x, y) = \mu_R(x, y)$  - это функция принадлежности отношения  $R \subseteq \mathbf{X} \times \mathbf{Y}$

В частном случае, когда  $T$ -норма имеет тип min

$$\mu_{B'}(y) = \sup_{x \in \mathbf{X}} \left\{ \min [\mu_{A'}(x), \mu_{A \rightarrow B}(x, y)] \right\}$$

Если  $A = A'$  и  $B = B'$ ,

то обобщенное нечеткое правило modus ponens упрощается до правила классического modus ponens.

### Обобщенное нечеткое правило modus tollens

Обобщенное (нечеткое) правило вывода modus tollens, где  $A, A' \subseteq \mathbf{X}$  и  $B, B' \subseteq \mathbf{Y}$  - это нечеткие множества, в то время как  $x$  и  $y$  - это лингвистические переменные.

Условие	$y$ это $B'$
Импликация	IF $x$ это $A$ THEN $y$

	это $B$
Вывод	$x$ это $A'$

Подобный пример с лингвистическими переменными со скоростью автомобиля и уровнем шума:

Условие	Уровень шума в автомобиле не очень высокий
Импликация	Если скорость автомобиля очень большая, то уровень шума высокий
Вывод	Скорость автомобиля большая

Нечеткое множество  $A'$  в выводе схемы (3.195) определяется комбинацией отношений

$$A' = (A \rightarrow B) \circ B'$$

Причем

$$\mu_{A'}(x) = \sup_{y \in Y} \left\{ \mu_{A \rightarrow B}(x, y)^T * \mu_{B'}(y) \right\}$$

Если  $T$ -норма имеет тип  $\min$ , то

$$\mu_{A'}(x) = \sup_{y \in Y} \left\{ \min \left[ \mu_{A \rightarrow B}(x, y), \mu_{B'}(y) \right] \right\}$$

Если  $A' = \bar{A}$  или  $B' = \bar{B}$ ,

то обобщенное нечеткое правило вывода modus tollens упрощается до классического правила modus tollens.

## Правила нечеткого вывода

Функции принадлежности в выводах схем modus ponens и modus tollens, зависят от функции принадлежности  $\mu_{A \rightarrow B}(x, y)$  нечеткой импликации  $A \rightarrow B$ , которая равнозначна нечеткому отношению  $R \subseteq \mathbf{X} \times \mathbf{Y}$ . Существуют варианты задания функции  $\mu_{A \rightarrow B}(x, y) = \mu_R$  на основе известных функций принадлежности  $\mu_A(x)$  и  $\mu_B(y)$ .

### 1. Правило типа minimum

$$\mu_{A \rightarrow B}(x, y) = \mu_R(x, y) = \mu_A(x) \wedge \mu_B(y) = \min[\mu_A(x), \mu_B(y)]. \quad (2.2)$$

2. Правило типа «prod»  $\mu_{A \rightarrow B}(x, y) = \mu_R(x, y) = \mu_A(x) \cdot \mu_B(y)$ . «правило Ларсена». (2.3)

### 3. Правило Лукашевича

$$\mu_{A \rightarrow B}(x, y) = \mu_R(x, y) = 1 \wedge [1 - \mu_A(x) + \mu_B(y)] = \min[1, 1 - \mu_A(x) + \mu_B(y)]. \quad (2.4)$$

### 4. Правило типа max-min

$$\begin{aligned} \mu_{A \rightarrow B}(x, y) &= \mu_R(x, y) = [\mu_A(x) \wedge \mu_B(y)] \vee [1 - \mu_A(x)] = \\ &= \max\{\min[\mu_A(x), \mu_B(y)], 1 - \mu_A(x)\}. \end{aligned} \quad \text{«правило Заде»}. \quad (2.5)$$

### 5. Бинарное правило

$$\mu_{A \rightarrow B}(x, y) = \mu_R(x, y) = [1 - \mu_A(x)] \vee \mu_B(y) = \max[1 - \mu_A(x), \mu_B(y)]. \quad (2.6)$$

6. Правило Гогуэна 
$$\mu_{A \rightarrow B}(x, y) = \mu_R(x, y) = 1 \wedge \frac{\mu_B(y)}{\mu_A(x)} = \min\left\{1, \frac{\mu_B(y)}{\mu_A(x)}\right\}. \quad (2.7)$$

## 8. Правило Гёделя

$$\mu_{A \rightarrow B}(x, y) = \mu_R(x, y) = \begin{cases} 1, & \text{если } \mu_A(x) \leq \mu_B(y), \\ \mu_B(y), & \text{если } \mu_A(x) > \mu_B(y). \end{cases} \quad (2.8)$$

## 9. Вероятностное правило

$$\begin{aligned} \mu_{A \rightarrow B}(x, y) &= \mu_R(x, y) = 1 \wedge [1 - \mu_A(x) + \mu_A(x) \mu_B(y)] = \\ &= \min[1, 1 - \mu_A(x) + \mu_A(x) \mu_B(y)]. \end{aligned} \quad (2.9)$$

Существуют и другие определения нечеткой импликации (см. раздел 1.7.).

### **Нечеткие алгоритмы**

Понятие нечеткого алгоритма, впервые введенное Л.Заде, является важным инструментом для приближенного анализа сложных систем и процессов принятия решения. Под нечетким алгоритмом понимается упорядоченное множество нечетких инструкций (правил), в формулировке которых содержатся нечеткие указания (термы).

Например, нечеткие алгоритмы могут включать в себя инструкции типа:

- а)  $x = \text{очень малое}$
- б)  $x \text{ приблизительно равно } 5$
- в)  $\text{слегка увеличить}$
- г)  $\text{если } x \text{ – в интервале } [4.9, 5.4]$
- д)  $\text{ЕСЛИ } x \text{ – малая, ТО } y \text{ – большое, ИНАЧЕ } K \text{ – } y \text{ – небольшое.}$

Использованные здесь термы «очень малое», «приблизительно равно», «слегка увеличить», и т.п. отражают неточность представления исходных данных и неопределенность, присущую самому процессу принятия решения.

Две последние инструкции (г и д) представляют собой нечеткие высказывания, построенные по схеме логической импликации ЕСЛИ – ТО, где условие ЕСЛИ соответствует принятию лингвистической переменной  $x$

некоторого значения  $A$ , а вывод (действие)  $TO$  означает необходимость выбора значения  $B$  для лингвистической переменной  $y$

$$(x = A) \rightarrow (y = B) /$$

Указанные правила получили широкое распространение. Процедура построения правил вывода в конкретной задаче выгладит при этом следующим образом.

На основе заданной цели (рис.2.1.2) с помощью упрощения, позволяющего выделить наиболее существенные факторы, определяются начальное состояние системы, желаемое конечное состояние и правила действия, приводящие систему в желаемое конечное состояние.

Набор таких правил, обеспечивающих получение «хорошего» (как правило), приближенного решения поставленной задачи, реализуется с помощью процедуры вывода.

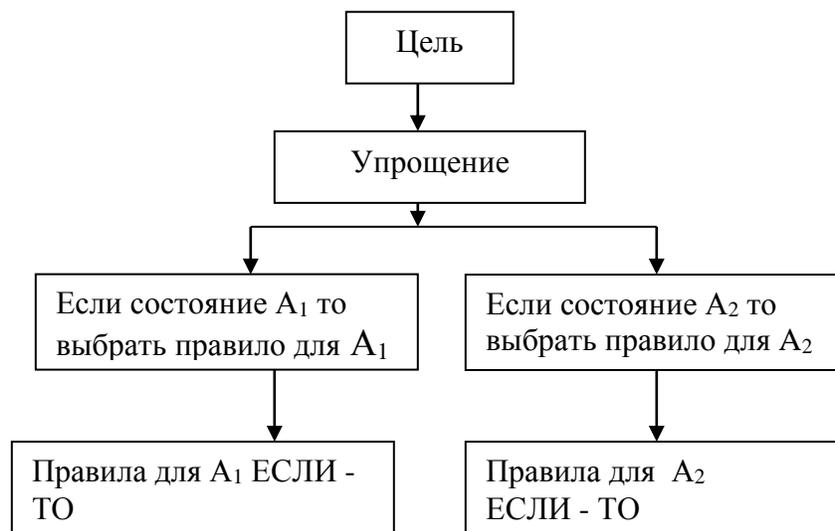


Рис.2.1 2. Схема алгоритма выполнения правил

Рассмотрим правило:

Если прибыль клиента **Большая**, то кредитный рейтинг клиента **Хороший**

Лингвистические переменные “**Большая**” и “**Хороший**”. Пусть “большая прибыль” соответствует величине 2000 рублей и имеет функцию принадлежности и значения прибыли в виде множества

Большая прибыль	0.1	0.3	0.7	0.8	0.9	1	1
$M_A(x_1)$	1.4	1.6	1.7	1.8	1.9	2	2.2

Хороший рейтинг можно представить значениями от 14 до 20. Нечеткое множество понятия “Хороший рейтинг” представлен в виде

$M_B(y_1)$	0.1	0.2	0.3	0.5	0.8	0.9	1
Хороший рейтинг	14	15	16	17	18	19	20

Правило в словесной форме представлено в виде двух нечетких множеств. Для формирования адекватной модели необходимо подстраивать значения функций принадлежности. Правила затем хранятся в базе знаний и применяются по мере необходимости.

Понятие большая прибыль имеет различные значения: “довольно большая прибыль”, “очень большая прибыль”, “значительная прибыль”, т.е.  $A$  не равно  $A^*$  и соответственно будет  $B$  не равно  $B^*$ . Пусть в процессе моделирования используется значение “очень большая прибыль” и нечеткое множество в виде

$M_A(x_1)$	0.5	1	0.8	0.2
Очень большая прибыль	14	15	16	17

заключение на основе двух правил и одного вывода. Если “большая прибыль”, то “хороший рейтинг” и если “очень большая прибыль”, то “неплохой рейтинг”.

В четкой логике такого сопоставления сделать невозможно, потому  $A$  не равно  $A^*$ . Нечеткая логика позволяет это сделать, так же как и человек, путем сопоставления.

Если “большая прибыль”, то “хороший рейтинг” и если “очень большая прибыль”, то “рейтинг неплохой”.  $X$  - полное множество значений “большая прибыль”,  $Y$  – полное множество значений “хороший рейтинг”. Используя значения для функций принадлежности и выражение для правила, получим вывод для  $A^*$  рейтинг 17, исходя из максимума функции принадлежности (Рис. 2.1.3.)

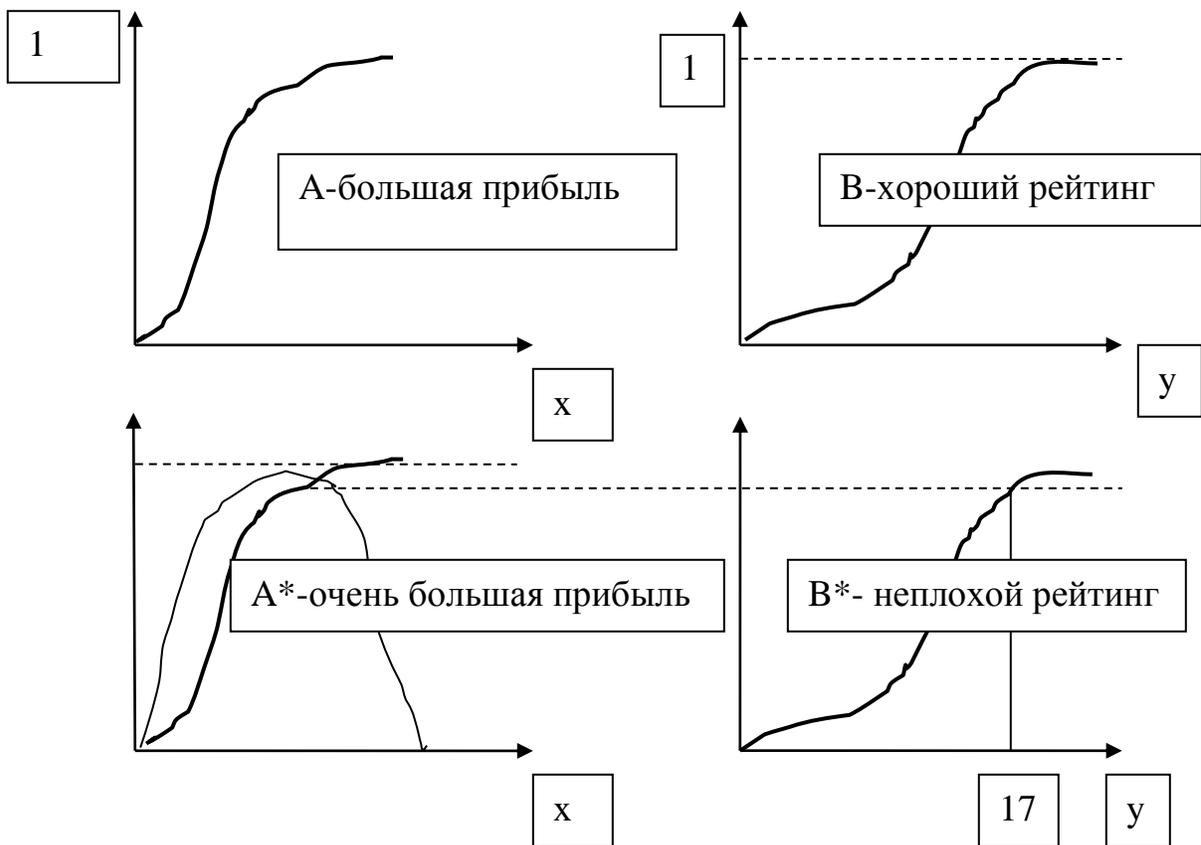


Рис. 2.1.3. Графическое представление нечеткого вывода

Типовая структура нечеткой модели системы вывода имеет два входа и один выход (Рис.2.1.4.) На входы нечеткой модели поступают числовые

значения  $x_1^*$  и  $x_2^*$ . Блок «фаззификации» вычисляет их степени принадлежности входным нечетким множествам  $A_i$ ,  $B_j$ . Для выполнения указанной операции блок фаззификации должен иметь доступ к точно определенным функциям принадлежности  $\mu_{A_i}(x_1)$  и  $\mu_{B_j}(x_2)$  входов. Примеры таких функций принадлежности приведены на рис. 2.1.4.. Вычисленные и представленные на выходе блока фаззификации степени принадлежности  $\mu_{A_i}(x_1^*)$  и  $\mu_{B_j}(x_2^*)$  дают информацию о том, в какой степени числовые значения  $x_1^*$  и  $x_2^*$  принадлежат конкретным нечетким множествам, т. е. насколько эти величины являются малыми ( $A_1$ ,  $B_1$ ) или большими ( $A_2$ ,  $B_2$ ).

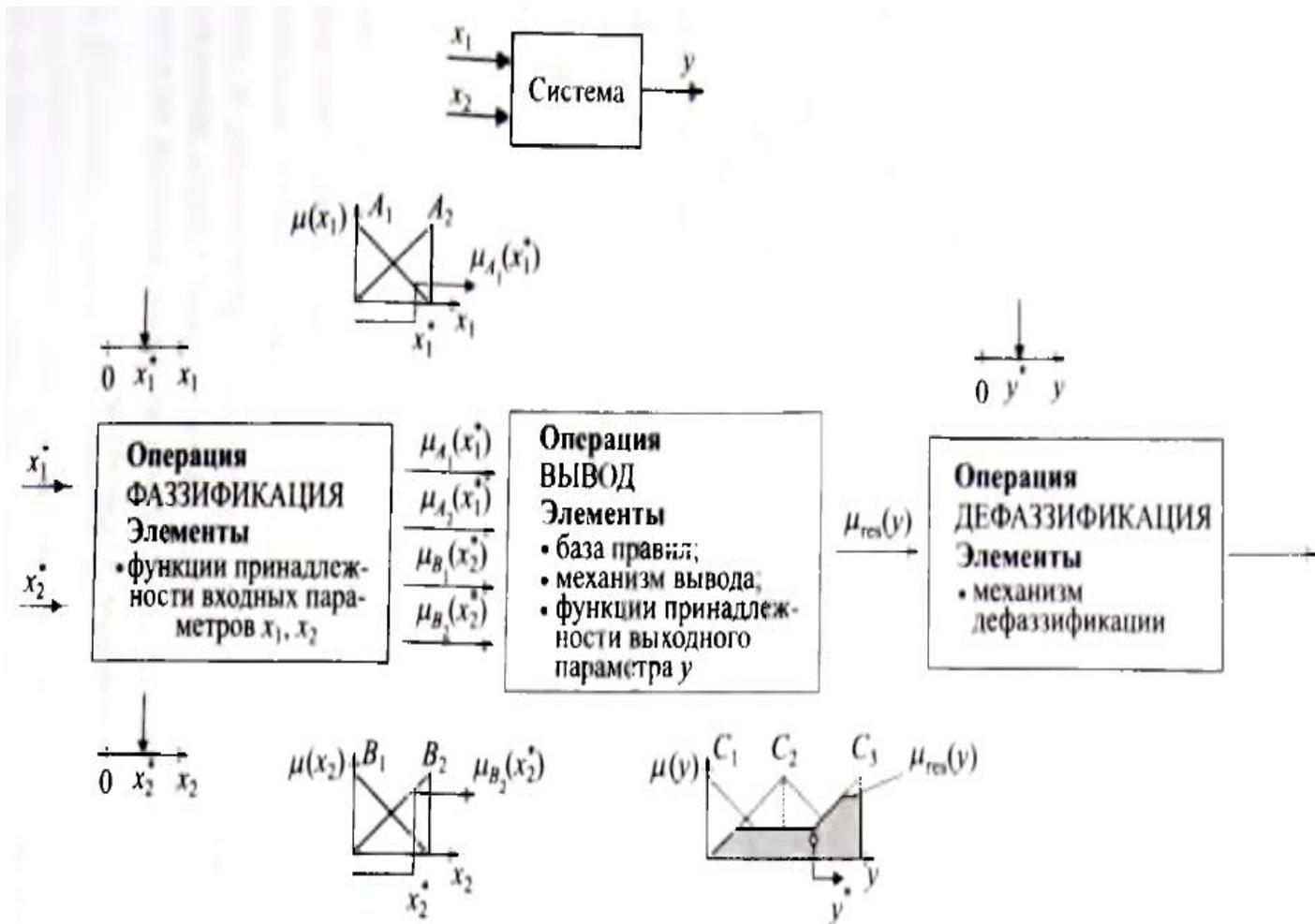


Рис. 2.1.4. Структурная схема модели с двумя входами и одним выходом

Блок «**вывод**» на входе получает степени принадлежности  $\mu_A$ ,  $\mu_B$ , на выходе вычисляет выходную функцию принадлежности, имеющую сложную форму.

Для выполнения вычислений блок вывода должен включать в себя следующие строго определенные элементы:

- база правил,
- механизм вывода,
- функции принадлежности выходного параметра  $y$ .

База правил содержит логические правила, задающие имеющие причинно-следственные отношения между нечеткими значениями ее входных и выходных величин. База правил может иметь следующий вид:

**R1 : ЕСЛИ (  $x_1 = A_1$  ) И (  $x_2 = B_2$  ) ТО {  $y = C_1$  },**

**R2 : ЕСЛИ {  $x_1 = A_2$  И (  $x_2 = B_2$  ) ТО (  $z = C_2$  ),**

**R3 : ЕСЛИ (  $x_1 = A_3$  ) И (  $x_2 = B_3$  ) ТО (  $y = C_3$  ),**

**R4 : ЕСЛИ (  $x_1 = A_4$  ) И (  $x_2 = B_4$  ) ТО (  $y = C_4$  ),**

где нечеткие значения входных параметров ( $A_1$  — малый,  $A_2$  — большой и т. д.) представлены на рис. 2.1.4., а выходных — на рис. 2.1.5..  $C_1$  = малый (примерно 0),  $C_2$  = средний (примерно 4),  $C_3$  = большой (примерно 8)

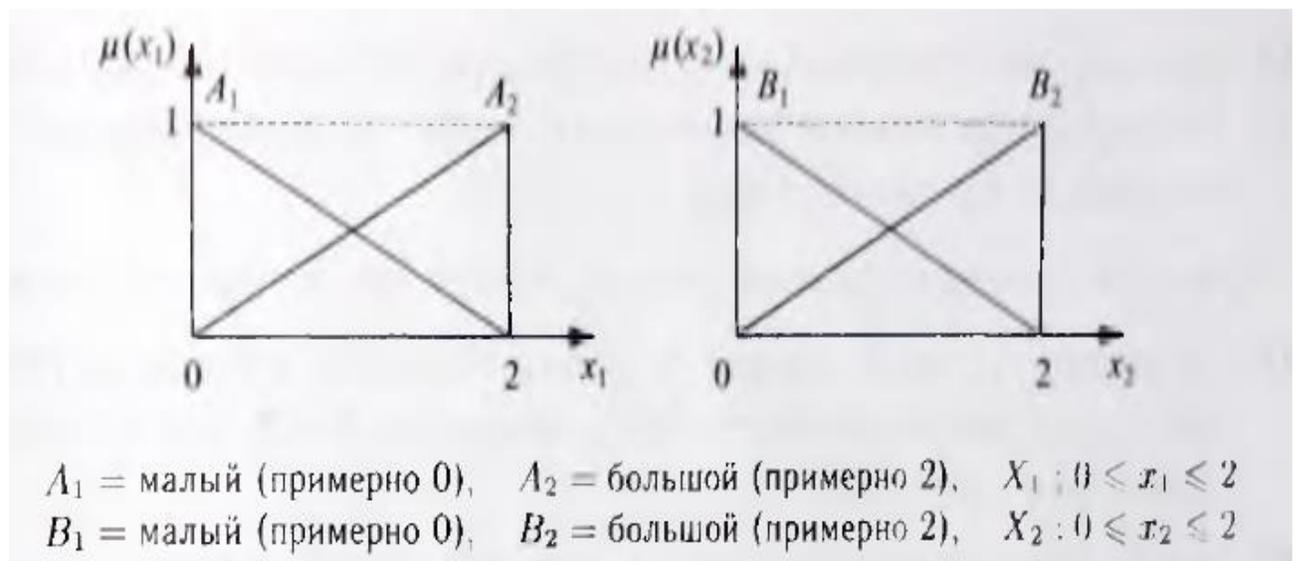


Рис. 2.1.5. Функций принадлежности нечетких множеств с указанием их областей определения

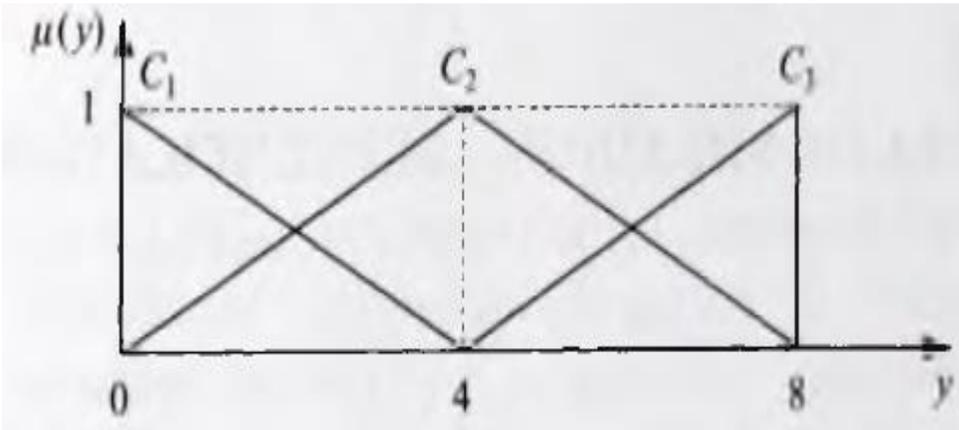


Рис. 2.1.6. Функции принадлежности нечетких значений выхода модели с указанием области определения

Блок вывода определяет функции принадлежности  $\mu_{rez}(y)$  механизмом вывода, состоящем из следующих элементов:

- элемент, вычисляющий степень выполнения каждого правила  $R_i$  в отдельности,
- элемент, вычисляющий активизированные функции принадлежности заключений каждого правила  $R_i$ ,
- элемент, вычисляющий результирующую функцию принадлежности  $\mu_{rez}(y)$  выходного значения на основе активизированных заключений отдельных правил.

Механизм вывода системы с двумя входами:

ИМ1: агрегация условий правил с использованием оператора PROD (произведение) для пересечения множеств (И) и оператора MAX для объединения множеств (ИЛИ),

ИМ2: определение активизированных функций принадлежности заключений правил с использованием оператора импликации Мамдани,

ИМ3: определение результирующей функции принадлежности выходного значения (аккумуляция) с использованием оператора MAX.

Блок «дефаззификация» на основе результирующей функции принадлежности вычисляет четкое числовое значение  $y^*$  выходного параметра, являющееся

результатом для входных числовых значений  $x_1^*$ ,  $x_2^*$ , Эта операция выполняется посредством процедуры дефаззификации, с определенным способом вычисления. (например вычисление центра тяжести).

Блок вывода на основе степеней принадлежности  $\mu_{A_1}(x_1)$ ,  $\mu_{B_1}(x_2)$  входных значений определяет результирующую функцию принадлежности  $\mu_{rez}(y)$  выходного значения модели (рис. 2.1.6). Операция вывода включает в себя следующие шаги:

- 1) вычисление степеней выполнения отдельных правил (точнее, их условий),
- 2) определение активизированных функций принадлежности заключений отдельных правил,
- 3) определение результирующей функции принадлежности вывода из всех правил, входящих в базу.

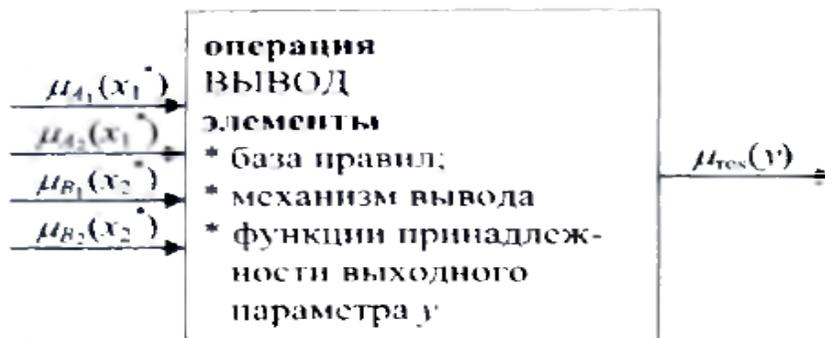


Рис. 2.1.7. Блок вывода нечеткой модели

Пример нечеткого вывода при моделировании системы

Для большинства задач моделирования реализуемое системой отображение входных значений в выходные задается не в математической форме, а в виде численных измерений входной и выходной информации, либо в форме знаний о системе (лингвистические переменные), полученных ее оператором или экспертом в результате наблюдений над ее поведением.

Имеется нечеткая модель системы с двумя входами и одним выходом, реализующую отображение системы:  $Y = X_1 + X_2$   $X_1 = [0,10]$ ,  $X_2 = [0,10]$ ,  $Y = [0,20]$ .

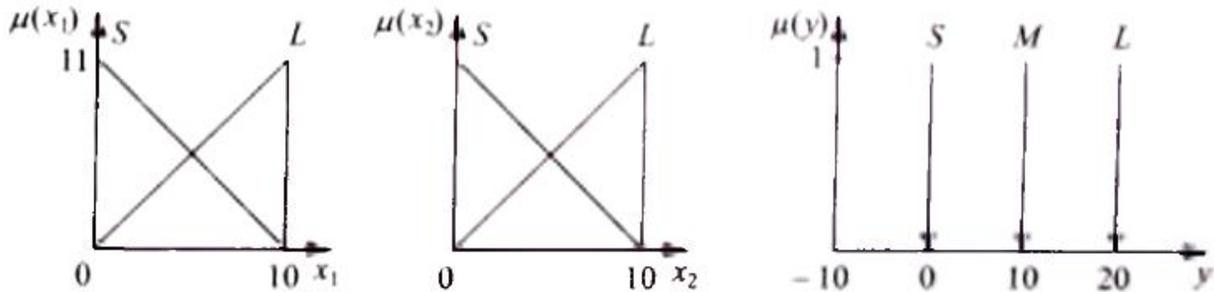


Рис. 2.1.8. Функции принадлежности, используемые в нечеткой модели

Пусть функции принадлежности значений входных и выходных параметров системы имеют вид как на рис. 2.1.7, и задана система правил:

- $r_1$  : ЕСЛИ  $(x_1 = S)$  И  $(x_2 = S)$  ТО  $(y = S)$ ,  
 $r_2$  : ЕСЛИ  $(x_1 = S)$  И  $(x_2 = L)$  ТО  $(y = M)$ ,  
 $r_3$  : ЕСЛИ  $(x_1 = L)$  И  $(x_2 = S)$  ТО  $(y = M)$ ,  
 $r_4$  : ЕСЛИ  $(x_1 = L)$  И  $(x_2 = L)$  ТО  $(y = L)$ .

$S$ — малый,  $M$  - средний,  $L$  - большой

Среди правил имеются два правила ( $r_2$  и  $r_3$ ) с одинаковым заключением ( $y = M$ ). Их можно объединить в одно правило  $R_2$ , что позволяет уменьшить количество правил до трех. В результате получаем базу правил следующего вида:

- $R_1$  : ЕСЛИ  $(x_1 = S)$  И  $(x_2 = S)$  ТО  $(y = S)$ ,  
 $R_2$  : ЕСЛИ  $(x_1 = S)$  И  $(x_2 = L)$  ИЛИ  
 $(x_1 = L)$  И  $(x_2 = S)$  ТО  $(y = M)$ ,  
 $R_3$  : ЕСЛИ  $(x_1 = L)$  И  $(x_2 = L)$  ТО  $(y = L)$ .

Объединение правил не является обязательным — можно также использовать модель с базой, содержащей 4 правила. Общая схема нечеткой модели представлена на рис. 2.1.8. Элементы модели:

- механизм вывода: MAX-MIN,
- агрегация условий: операторы MIN и MAX.

Чтобы иметь возможность сравнивать различные ситуации, вычислим значения  $y^*$  на выходе нечеткой модели и значения  $y$  на выходе моделируемой системы для входных значений  $x_1^* = 2.5$  и  $x_2^* = 7.5$ .

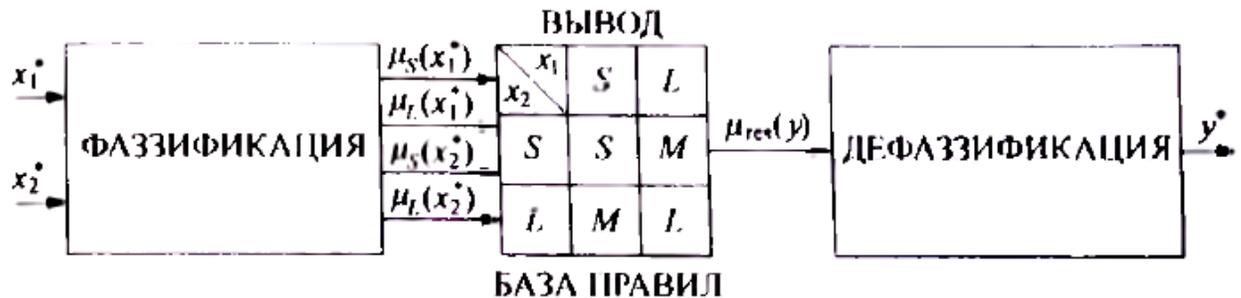


Рис. 2.1.9.. Общая схема рассматриваемой нечеткой модели

На рис. 2.1.10 представлен процесс вычисления. Для входных значений  $x_1 = 2.5$ ,  $x_2 = 7.5$  на выходе моделируемой системы, реализующей отображение  $y = x_1 + x_2$ , получаем значение 10. Это же значение будет получено и на выходе

## 2.2. Фаззификация. Дефаззификация результирующей функции принадлежности

Цель фаззификации – установление соответствия между конкретным (обычно числовым) значением конкретной входной переменной системы нечеткого вывода и ее соответствующим лингвистическим значением в виде нечеткого множества и функции принадлежности.

До начала этапа фаззификации вычисляются области определения всех входных переменных, являющихся универсальными множествами, на которых будут определяться лингвистические значения и соответствующие им нечеткие множества.

В общем случае входной переменной  $x_i$  может быть поставлено в соответствие терм-множество лингвистических значений  $T_x = \{\tau_j : j = 1, J_{x_i}\}$ , множество

имен лингвистической переменной  $L_{x_i} = \{l_j; j = 1, J_{x_i}\}$  и соответствующие

нечеткие множества  $\tilde{A} = \{\mu_{l_j}(x_i) \mid x_i \in D(x_i) : j = \overline{1, J_{x_i}}\}$ , где  $J_{x_i}$  – число

лингвистических значений входной переменной  $x_i$ . Формально процедура фаззификации состоит в проверке выполнения условий « $\alpha$  есть  $\beta$ », содержащихся в правилах условного нечеткого вывода, находящихся в базе знаний системы. Более просто и наглядно ее можно представить в графической форме (рис.5.4)

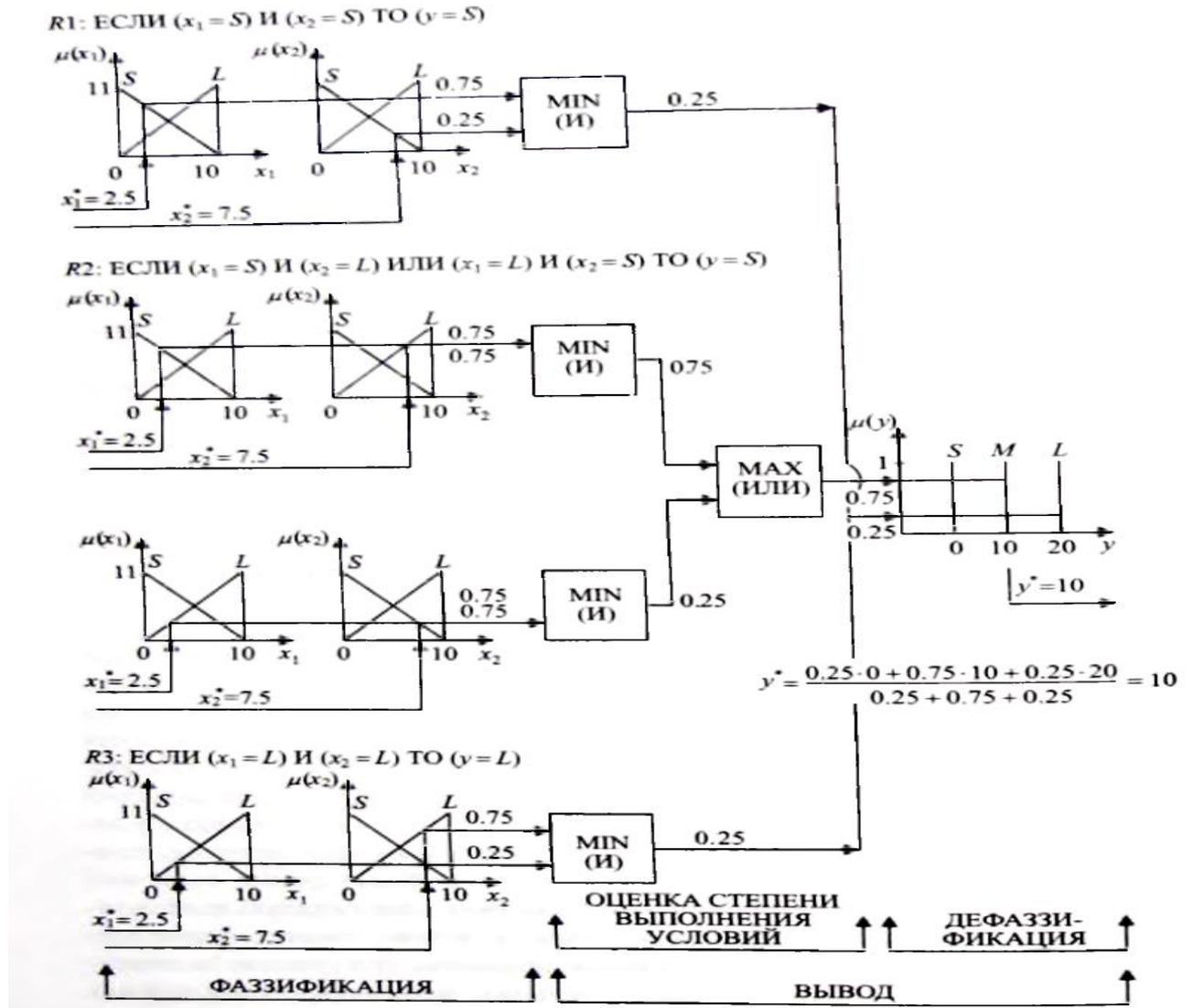


Рис2.1.10... Схема вычисления выходного значения  $y^*$  нечеткой модели

для входных значений  $x_1 = 2.5, x_2 = 7.5$

Значение функции принадлежности можно интерпретировать как оценку истинности выполнения нечеткого высказывания, поэтому в данном случае (см.

рис. 2.2.1.) истинность высказывания «  $x_i$  есть  $y_j$  » больше, чем истинность высказывания «  $x_i$  есть  $l_{j+1}$  »  $\mu_{l_j}(\alpha_t) > \mu_{l_{j+1}}$ .

При фаззификации могут быть получены два результата, что определяет возможность использования нескольких правил нечеткого условного вывода для получения окончательного заключения. Поэтому при построении системы должно быть принято решение, как будут применяться результаты фаззификации: будет использоваться один результат или оба.

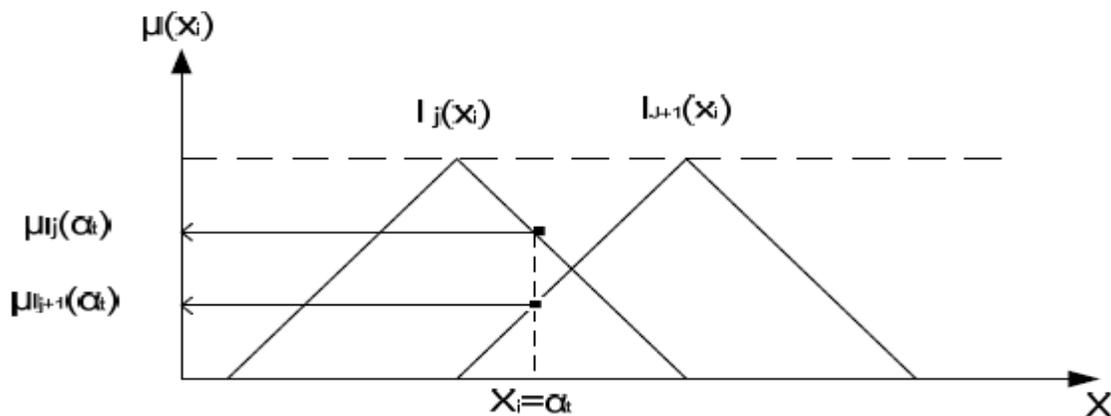


Рис. 2.2.1. Процедура фаззификации

Процесс фаззификации для конкретной входной переменной считается законченным, когда проверены все соответствующие ей возможные лингвистические значения. Если соблюдаются правила построения исходных функций принадлежности, то текущее значение входной переменной идентифицируется только с двумя соседними лингвистическими значениями (рис. 2.2.1.).

В блоке фаззификации, представленном на рис. 2.2.1., вычисляются степени принадлежности числовых значений входных параметров модели входным нечетким множествам.

Равенство 0.3 степени принадлежности входного значения  $x_1 = 1.4$  нечеткому множеству  $A_1$  (малый) означает, что степень соответствия данного значения наиболее типичному малому значению (0) равна 0.3. Утверждение о том, что значение  $x_1 = 1.4$  большое, является истинным со степенью 0.7. Таким образом, указанное значение  $x_1$  в большей степени соответствует типичному большому значению (2), чем типичному малому (0).

Для вычисления степеней принадлежности значений конкретным нечетким множествам, функции принадлежности последних должны быть точно заданы на качественном (вид функции) и количественном (ее параметры) уровне. Как форма функции принадлежности, так и ее параметры, оказывают существенное влияние на точность модели



Рис. 2.2.2. Блок фазификации и пример его работы

Примером математического описания функций принадлежности (рис. 2.2.2.) является совокупность выражений вида:

$$\begin{aligned}\mu_{A_1}(x_1) &= 0.5(2 - x_1), & \mu_{A_2}(x_1) &= 0.5x_1, \\ \mu_{B_1}(x_2) &= 0.5(2 - x_2), & \mu_{B_2}(x_2) &= 0.5x_2.\end{aligned}$$

В процессе фаззификации четкий входной вектор  $X^*$  преобразуется в вектор  $M$  степеней принадлежности, которые, в свою очередь, являются входными данными для блока вывода:

$$X^* = \begin{bmatrix} x_1^* \\ x_2^* \end{bmatrix} \xrightarrow{\text{фаззификация}} M = \begin{bmatrix} \mu_{A_1}(x_1^*) \\ \mu_{A_2}(x_1^*) \\ \mu_{B_1}(x_2^*) \\ \mu_{B_2}(x_2^*) \end{bmatrix}$$

### Определение подходящего правила

Для выработки заключения в базе знаний системы выбирают правила нечеткого вывода, в условной части которых содержатся высказывания и присутствует входная переменная с установленным в процессе фаззификации лингвистическим значением.

В зависимости от использования результатов фаззификации, выбирается различное число правил вывода, которые рассматриваются для получения заключения. Алгоритм поиска правил сказывается на эффективности системы в целом.

Время поиска правил важно для нечетких систем управления техническими объектами, когда задача управления решается в реальном масштабе времени и время ограничено условиями динамики управляемого процесса.

Агрегация высказываний в условной части и обработка правил. Обычно в условной части правил нечеткого вывода содержится составное нечеткое высказывание, составленное из простых правил, логически связанных. Внутри простых высказываний возможно использование логических связок, модификаторов и квантификаторов. В соответствии со структурой условной

части выбранных правил нечеткого условного вывода по правилам нечеткой логики должна быть выполнена свертка высказываний с целью получения интегрального условия, соответствующего всей совокупности высказываний в условной части правила.

На этапе фаззификации для простого высказывания получается оценка его истинности в виде значения некоторой функции принадлежности  $\mu_{\text{rez}}(y)$ . В результате выполнения свертки простых высказываний с соответствующими оценками истинности получается интегральная оценка истинности выполнения условной части правил  $\mu_1, \mu_2, \dots, \mu_p$ , где  $p$  - число выбранных для построения заключения правил. В зависимости от установленного соглашения по использованию правил для формирования заключения может быть использовано одно правило с максимальным значением истинности выполнения условной части или же несколько правил, на основе которых будет выработано интегральное заключение. Количество правил может быть уменьшено, если установить некоторый порог истинности.

Последним этапом будет вычисление нечеткой импликации по одной из приведенных формул и получено некоторое множество нечетких заключений, вытекающих из конкретных правил вывода. Интегральное заключение представляется в форме нечеткого множества, на котором нужно будет выбрать единственное заключение. Эта задача решается на этапе дефаззификации.

### **Дефаззификация**

В общем случае после вычисления импликаций имеется набор нечетких

множеств  $M = \{\mu_{\tilde{w}_1}(z), \mu_{\tilde{w}_2}(z), \dots, \mu_{\tilde{w}_p}(z)\} \bar{W} = \{w_1, w_2, \dots, w_p\}$ ,

где  $p$  – число обработанных нечетких правил в виде множество допустимых значений вывода (управления).

На полученном нечетком множестве допустимых управлений выбирают единственное значение, представленное в количественной форме.

**Дефаззификация** в системах нечеткого вывода - это процедура нахождения обычного (не нечеткого) значения, которое используется внешними по отношению к системе нечеткого вывода элементами. Для реализации дефаззификации используются несколько методов, классификация которых представлена на рис. 2.2.3.

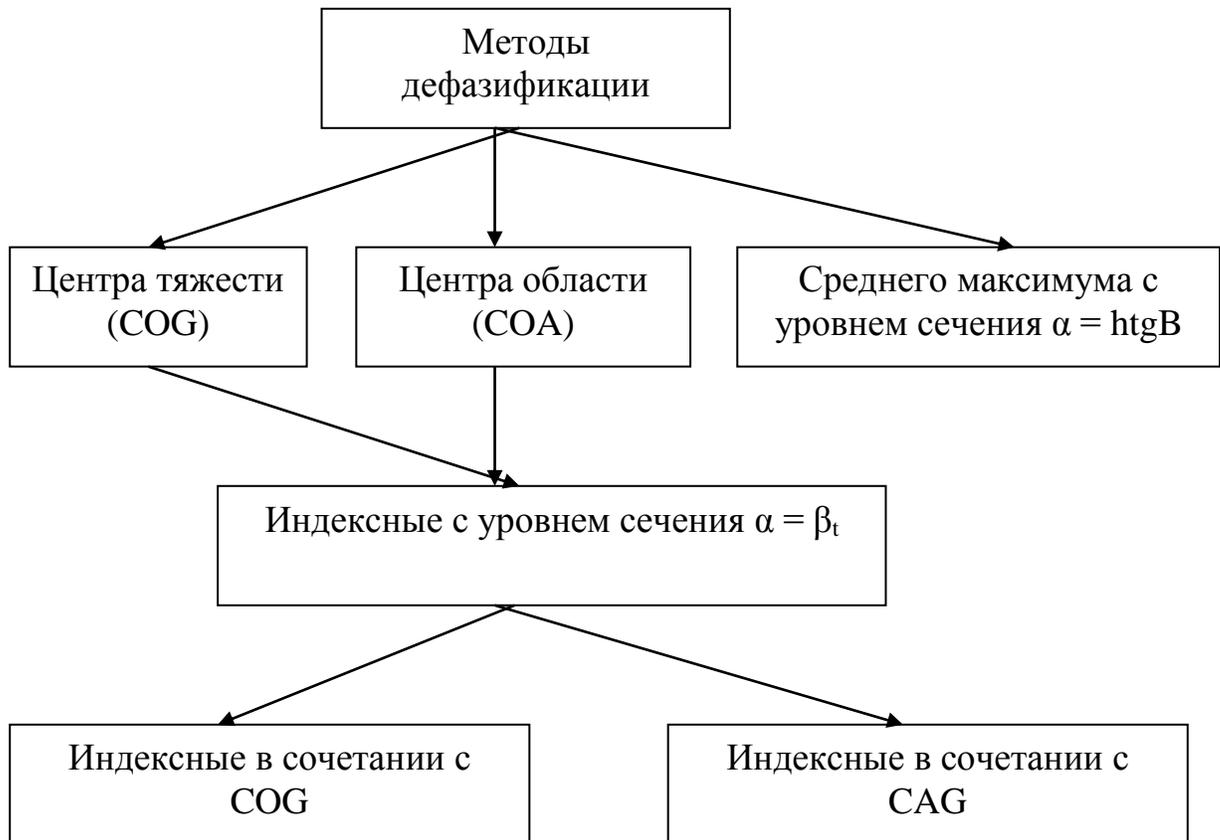


Рис.2.2.3. Классификация методов дефаззификации.

Наиболее простым и часто используемым считается метод среднего максимума функции принадлежности (COM), рис. 2.2.4.). В этом случае для унимодальной функции принадлежности результат дефаззификации определяется по координате ее максимума (рис.2.2.3, а), для трапецидальных функций результат определяется по середине верхнего основания (рис., 2.2.3, б). Для

функций принадлежности с несколькими экстремумами метод максимума не дает однозначного решения.

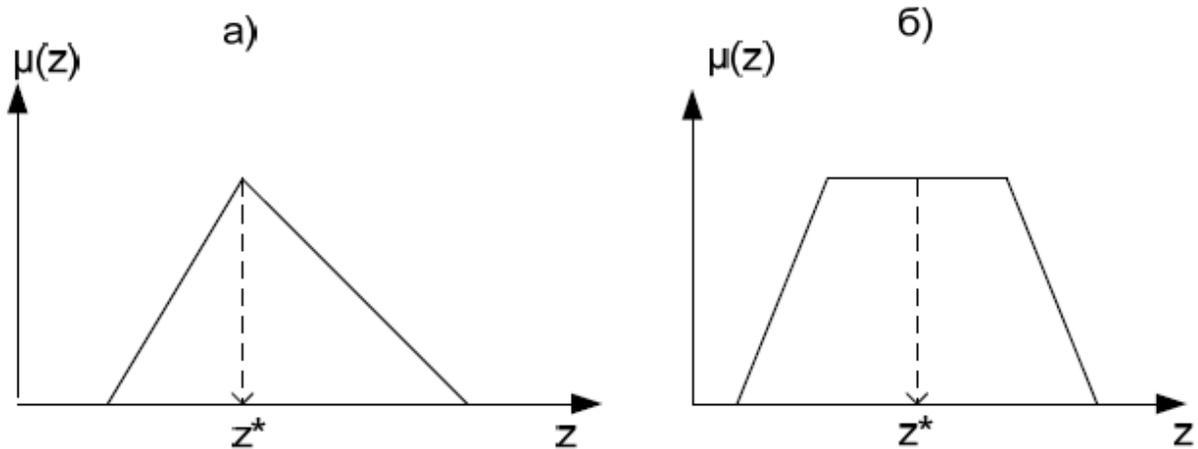


Рис. 2.2.4. Метод максимума функции принадлежности

2. Метод центра площади (COA). Искомое значение  $z^*$  определяется из уравнения

$$\int_{\min Z}^{z^*} \mu(z) dz = \int_{z^*}^{\max Z} \mu(z) dz.$$

Иными словами, определяется абсцисса прямой, делящей площадь по кривой функции принадлежности на две равные части. Метод центра площадей достаточно неудобен при реализации, и не может быть использован, если функция принадлежности задана дискретными значениями

3. Метод центра тяжести (COG). Этот метод состоит в том, что функцию принадлежности рассматривают как систему материальных точек, массы которых равны значениям функции принадлежности.

Известно, что координата центра тяжести является обобщенной характеристикой системы материальных точек [20]. Для непрерывных функций принадлежности координата центра тяжести определяется соотношением

$$CG = z^* = \frac{\int_{\min Z}^{\max Z} z \times \mu(z) dz}{\int_{\min Z}^{\max Z} \mu(z) dz}.$$

Для дискретных функций 1

$$CG = z^* = \frac{\sum_{i=1}^N z_i \mu(z_i)}{\sum_{i=1}^N \mu(z_i)} \quad (\text{CG-Centre of Gravity}).$$

Поскольку значения функции принадлежности интерпретируются как оценки истинности, при выполнении дефаззификации, нужно обращать внимание на максимальное значение функция принадлежности, представляющей нечеткое заключение.

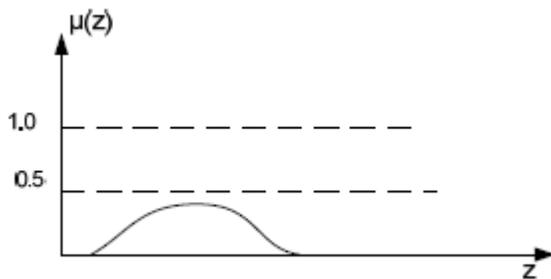


рис. 2.2.4. Метод центра тяжести

Если это значение мало (рис. 2.2.4.), то от принятия решения лучше воздержаться. Не исключается и вариант по пересмотру условий задачи.

Результирующая функция принадлежности  $\mu_B(z)$  может иметь достаточно сложную форму (рис.2.2.5.), и преобразование ее в физическую переменную нужно выполнять с помощью техники усреднения. Метод центра тяжести (center of gravity – cog) является методом, использующим усреднение:

Этот метод обобщается на многомерный случай нечеткого множества В,

для двумерного случая

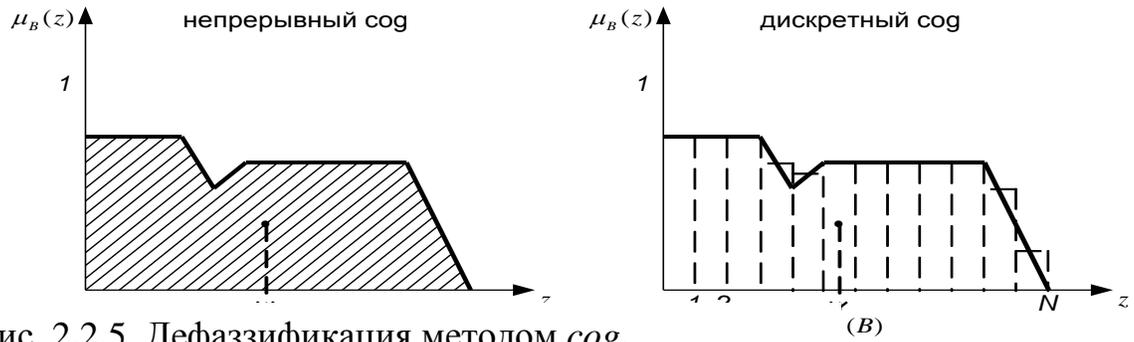
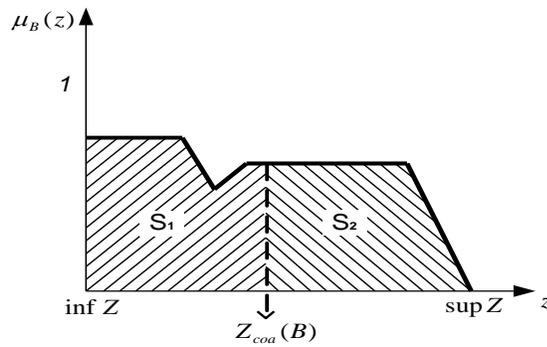


Рис. 2.2.5 Дефаззификация методом *cog*.



имеем:

Рис.2.2.6.. Дефаззификация методом *coa* (площадь  $S_1 = \text{площадь } S_2$ ).

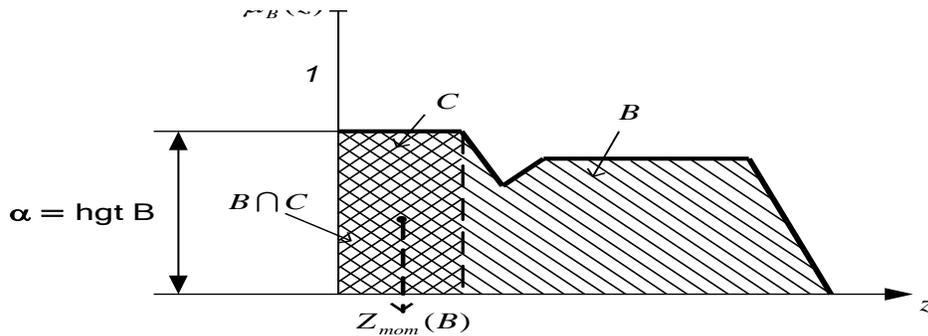


Рис.2.2.7. Дефаззификация методом *mom*.

Возможно также сочетание с методом “*coa*”.

Большой класс методов с фильтрующими свойствами составляют **индексные методы**, в которых  $\alpha = \beta_t$  ( $\beta_t$  – заданные значения уровней) в сочетании с методами “сog” и “coa”). В практических задачах обычно  $\beta_t = 0,5$ .

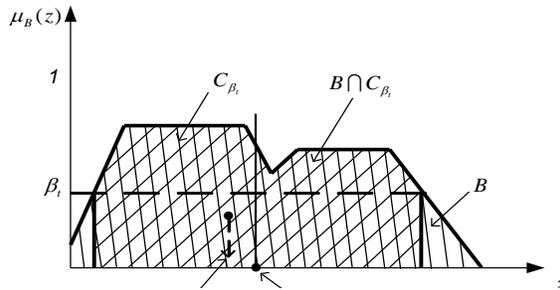


Рис.2.2.8. Дефаззификация индексными методами с уровнем разреза  $\beta_t$ .

Классификация методов дефаззификации приведена на рис.2.2.3.

### 2.3. Свойства правил и нечетких моделей

Центральным элементом в нечеткой логике является база правил, потому что в ней содержится информация о структуре модели. Базу правил можно сравнить с каркасом палатки, на которую натягивается ткань. Конструкция каркаса определяет форму и внешний вид всей палатки. По аналогии остальные элементы, такие как функции принадлежности, типы используемых операций, механизмы вывода и дефаззификации определяют структуру и свойства модели. Знание свойства правил и нечетких моделей является условием для обеспечения адекватности модели.

Существуют следующие свойства правил и нечетких моделей

1. Локальный характер правил
2. Зависимость числа правил от числа входных параметров нечетких моделей.
3. Непротиворечивость правил
4. Связность базы правил

## 5. Избыточность базы правил

Возникающие дефекты баз правил, приводят к необходимости разработки эффективных средств проверки их согласованности. Основные требования к базам правил:

- в базы правил следует включать знания только высококвалифицированных специалистов.
- полнота базы правил должна отвечать на все вопросы в рамках конкретного приложения.
- база правил должна давать четкие выводы на любые заданные вопросы в рамках приложения.

Проверка базы правил на состоятельность и непротиворечивость может рассматриваться как этап решения проблемы передачи знаний от человека к компьютерной системе. Ошибки могут возникать на любой стадии этого процесса:

- знания самого эксперта могут быть неполными или противоречивыми;
- знания эксперта с ошибками переносят в машинное представления либо по вине инженера по знаниям, имеющим поверхностное представление о проблемной области, либо по вине эксперта, не знакомого с информационными технологиями,
- орфографические и семантические ошибки при вводе данных в компьютер.

Основные дефекты баз правил – неполнота, противоречивость, избыточность. Также бывают безусловные ошибки, которые следует обязательно устранять. База правил не должна содержать правил, применяемых в одной и той же ситуации, но делающих противоположные выводы. В технической диагностике – это рекомендации “Тумблер включить” и “Тумблер выключить”. В медицинской диагностике – это два взаимоисключающих диагноза:

“Истощение” и “Ожирение”. В большинстве случаев степень допустимости дефекта определяется индивидуально.

### **Локальный характер правил**

Условия формирования правил характеризуют некоторую область пространства существования входных значений  $X_1 * X_2 * X_3 \dots X_n$ , в то время, как его заключение выдает область пространства выходных значений  $Y$ .

Условия правила имеющего вид:

ЕСЛИ  $(X_1 = A_1)$  И  $(X_2 = A_2)$  ТО  $(Y=B)$

задает окрестность точки с координатами  $(x_1, x_2)$ , а заключение этого правила связывает с данной точкой окрестности  $(y)$ . Если состояния входов в точности соответствует значению  $(x_1, x_2)$ , то значение выходов будет точно соответствовать  $(y)$ . В этом случае поверхность модели будет состоять из сегментов, в узлах которых расположены правила.

Изменение модального значения  $b(y)$  множества **B** вызовет смещение в верх либо вниз опорной точки поверхности модели, что приведет к изменению поверхности только одного сегмента. Таким образом, изменение носит локальный характер. Если изменение опорных точек (узлов) находятся на краях, то может измениться поверхность нескольких сегментов (но не всей поверхности), то есть тоже локально (рис.2.3.1.).

Изменение заключения правила приводит к локальному изменению сегментов поверхности модели, прилегающих к заданной правилом опорной точке в пространстве  $X_1 * X_2 * X_3 \dots X_n$ . На другие сегменты поверхности, не прилегающие к данной точке, изменение заключения либо вообще влияние не

оказывает

либо

влияет

слабо.

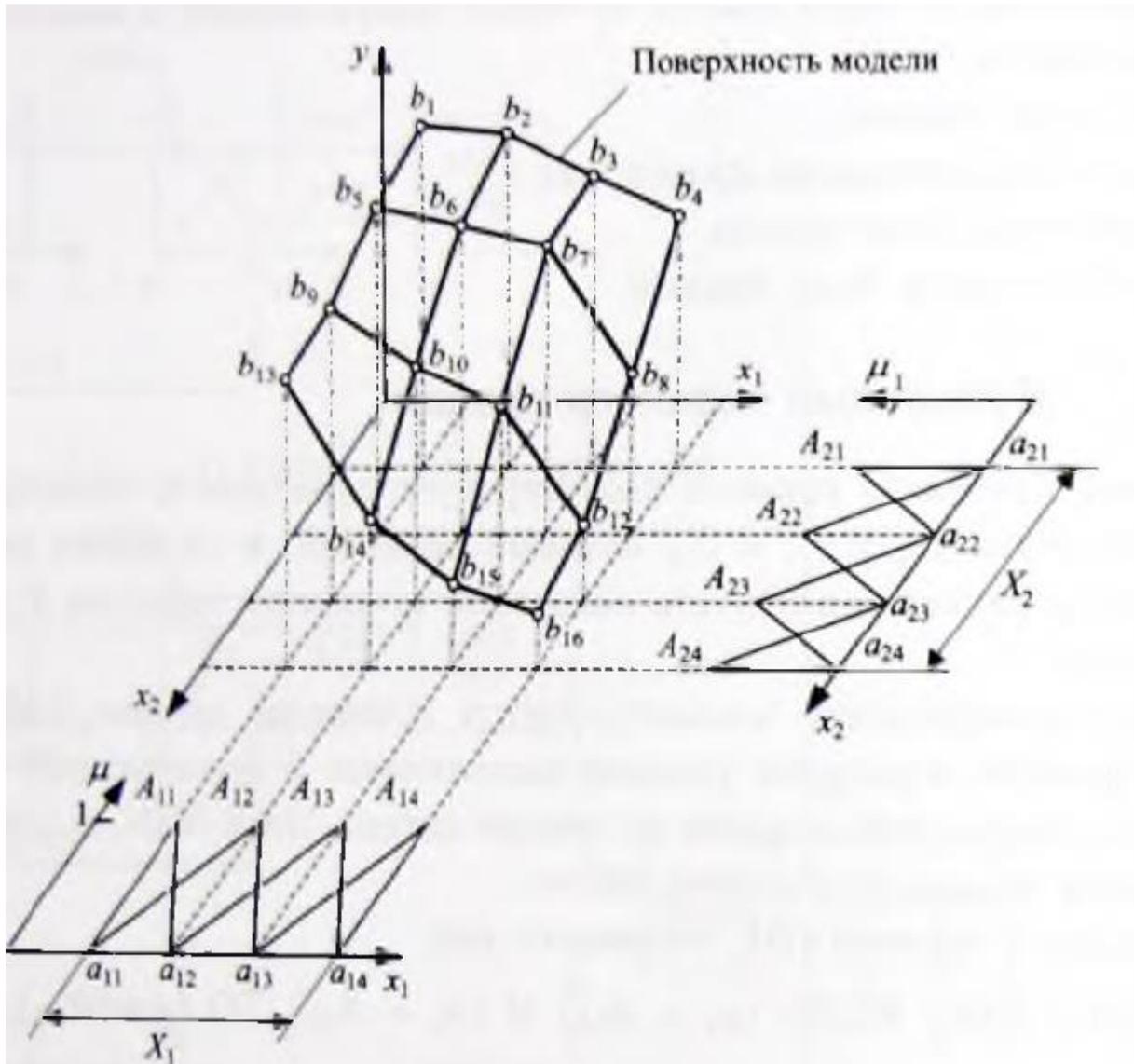


Рис. 2.3.1. База правил и поверхность отображения

С другой стороны характер изменения условий правил (содержащихся в них значений нечетких множеств) не является локальным. Так изменение значения  $A$ , повлияет на изменение условия всех правил, содержащих данное множество, и приведет к смещению опорных точек вправо или влево.

### Зависимость числа правил от числа входных параметров нечетких моделей

С повышением уровня сложности модели (с увеличением числа правил или числа множеств) увеличивается ее способность более точно описывать

реальную систему. Сложность модели можно считать ее достоинством. Но с увеличением сложности возрастает объем информации, описывающий систему и необходимый для определения параметров модели (функции принадлежности нечетких множеств) При этом объем информации необходимый для построения более сложной модели недостаточен, а это недостаток модели.

На усложнение модели влияет число входов и число правил. Если число входов модели  $x_i$  будет  $n$ , и каждый из них задается числом множеств  $z$ , то число  $r$  правил, имеющих простые условия, экспоненциально зависит от числа входов и числа, имеющихся множеств.

При увеличении числа входов с одного до двух, число правил возрастает с 3 до 9. Чтобы полностью определить модель с одним входом нужно задать 6 параметров функции принадлежности, а в случае модели с двумя входами 15 параметров.

Увеличение числа нечетких множеств с 3 до 4 приводит к увеличению числа правил с 9 до 16. а число параметров функций принадлежности с 15 до 24.

Если предположить, что для каждого входа используется одинаковое число нечетких множеств  $A_{ij}$ , и что с каждым правилом связано свое нечеткое множество  $B$ , то число параметров для задания функции принадлежности можно найти по формуле

$$P=r + n*z$$

Резкое увеличение числа правил и параметров функций принадлежности с увеличением количества входов приведено в таблице 3

Рост числа правил и параметров функций принадлежности при увеличении числа входов называют проклятием размерности.

В случае, когда для определения параметров функций принадлежности требуется информация о большом количестве точек в пространствах входных и выходных значений, экспертные методы терпят неудачу, и приходится

использовать методы настройки моделей на основе нормативных значений входных и выходных данных.

При большом количестве настраиваемых параметров применение других методов также затруднительно, поэтому нужно стремиться к разумному числу входных параметров.

Табл. 3

Число входов n	Число множеств z	Число правил r	Число параметров p
1	2	2	4
2	2	4	8
3	2	8	14
1	3	3	6
2	3	9	15
3	3	27	36
·			
5	3	125	140
·			
10	3	59049	59079

### Полнота нечеткой модели

Рассмотрим нечеткую модель, правила которой содержат набор нечетких условий:

R1 ЕСЛИ  $(x_1 = A_{11}) \dots$  И  $(x = A_{1i}) \dots$  И  $(x = A_{1n})$  ТО  $(x = B_1)$

R2 ЕСЛИ  $(x = A_{1j}) \dots$  И  $(x = A_{ji}) \dots$  И  $(x = A_{jn})$  ТО  $(x = B_j)$

Rm ЕСЛИ  $(x = A_{m1}) \dots$  И  $(x = A_{mi}) \dots$  И  $(x = A_{mn})$  ТО  $(x = B_m)$

Область X входных значений определяется, как декартово произведение областей  $X_i$ ,  $i=1,2,\dots,n$ , числовых значений отдельных параметров

$$X_1 * X_2 * X_3 \dots X_n$$

Символом Y обозначает значения выходных параметров.

Нечеткая модель называется полной, если с каждым входным состоянием  $X$  ( $x_1, x_2, \dots, x_n$ ), принадлежащим области  $X$ , связывается выходное состояние  $Y$ . Нечеткая модель называется неполной, если с некоторыми входными состояниями нельзя связать ни одного выходного состояния  $Y$ .

Полная модель является точной, однако условием достижения этого является ее полнота. Причиной неполноты модели может быть нечеткость разбиения области входных значений.

Нечеткое разбиение области входных значений переменной  $x$  является полным, если выполняется условие

$$\sum \mu_{A_i}(x_i) > 0 \quad x_i \in X_i, i=1..m$$

где  $m$  – число нечетких множеств  $A_i$ .

Определение предполагает, что каждое из значений  $x_i$  переменной  $x_i$  из области значений  $X$  принадлежит хотя бы одному нечеткому множеству  $A_i$ . Неполное и нечеткое разбиение не является абстрактным понятием, а встречается в некорректно построенных самообучающихся нечетких моделях. В ходе обучения изменяются параметры функций принадлежности, что приводит к смещению, расширению или даже сужению последних, и появляются интервалы, которые не покрываются ни одним нечетким множеством. Модель является неполной, если для какого входного состояния  $x$  невозможно определить значения  $y$  на выходе. Причиной этого является неполнота базы правил.

Условия правил содержат лингвистические оценки входных состояний, а заключение указывает какое выходное состояние (также в лингвистической форме) соответствует входному состоянию, заданному в условиях. Если в базе правил не содержится ни одного правила, задающего значение  $y$  для входного значения  $x=M$ , и если состояние  $x$  на входе оценивается в точности, как среднее ( $\mu_M(x)=1$ ), что соответствует ситуации  $x=x_m$ , то не активизируется ни одно из правил, и вычислить выходное значение становится невозможным.

При сравнении поверхностей полной нечеткой модели и неполной, то имеются значительные отличия, говорящие о большей адекватности полной модели для реальной системы. Точность модели уменьшается по мере увеличения расстояния между узлами поверхности (т.е. уменьшением количества правил в базе правил).

Отсутствие ряда правил, т.е. отсутствие информации о том, какие выходные значение соответствует входному лингвистическому состоянию значению, задаваемому посредством логического произведения (наличие в базе правил пустых полей) означает, что модели можно доверять только в пределах зоны, соответствующей полному набору правил. При удалении от этой зоны адекватность модели снижается и степень доверия уменьшается.

Существуют понятия лингвистической и численной полноты базы правил.

База правил модели называется лингвистически полной, если каждому входному лингвистическому состоянию  $A$  ставиться в соответствие хотя бы одно лингвистическое состояние  $B$ .

При этом **лингвистическая полнота** базы правил не является абсолютным или необходимым условием полноты нечеткой модели, потому что полноту модели можно достичь и правильным подбором формы функции принадлежности.

**Числено полной** называется база правил, в которой хотя бы одно четкое значение входной переменной приводит к активации хотя бы одного правила. В то же время при неполном разбиении входных значений лингвистическая модель может быть неполной числено. Нечеткая модель может быть неполной, если ее неоднородности лежат в области неиспользуемых моделью частях пространства входных и выходных переменных. Подобно база правил может быть лингвистически неполной, если не встречающиеся в правилах входные и выходные состояния модели не являются для нее рабочими.

Используемые неполные модели и базы правил обладают определенным риском, поэтому их необходимо подвергать тщательному анализу.

### Непротиворечивость базы правил

База правил называется непротиворечивой (согласованной), если она не содержит несовместные правила, т.е. правила имеющие одинаковые условия, но разные заключения.

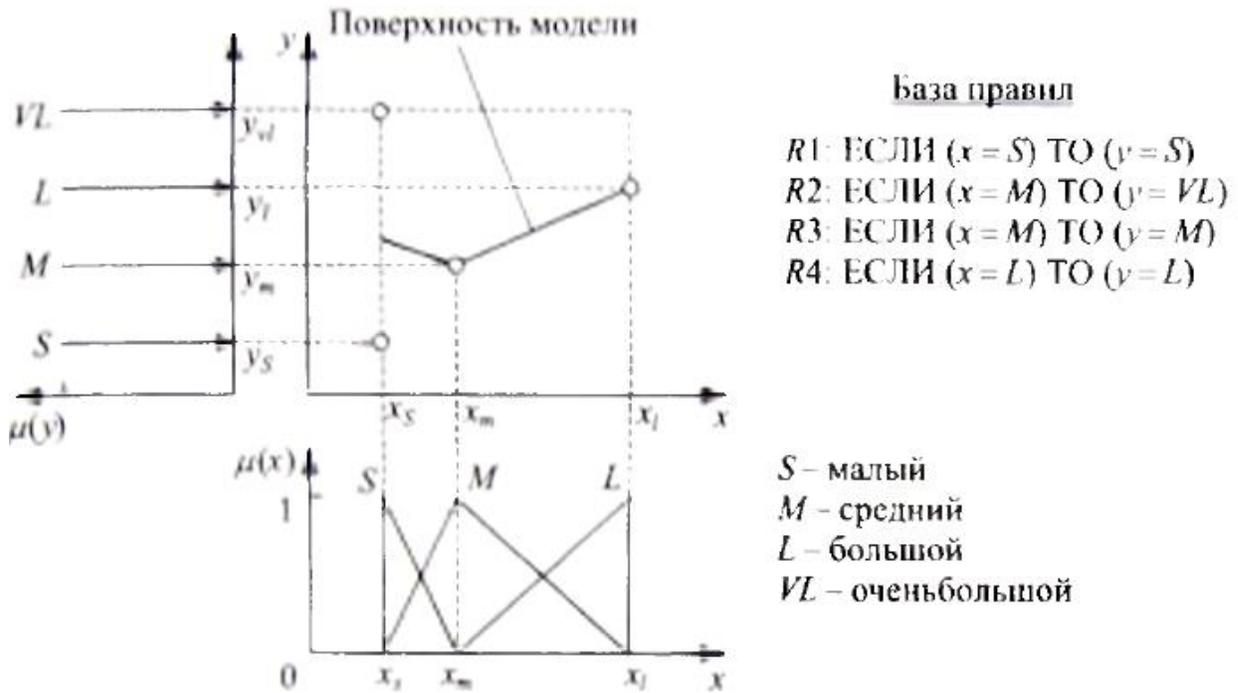


Рис. 2.3.2. Непротиворечивость базы правил

Противоречия в системе могут проявляться следующим образом:

1. В виде внутренне противоречивых правил, когда условия или условие и заключение правила взаимоисключают друг друга.
2. В виде противоречий между двумя правилами, когда при одинаковых ситуациях (описываемых одним и тем же словом состояния) формируются противоречивые заключения.
3. В виде внутренних противоречивых цепочек вывода, когда в процессе логического вывода применяется правило, которое делает слово состояния недопустимым: добавляется факт,

противоречащий исходным фактам или уже полученным выводам.

4. В виде противоречий между цепочками вывода, когда из одной ситуации выводятся цепочки с противоречивыми выводами.

Противоречивость является наиболее серьезной ошибкой. Если система вывода содержит правила, формирующая противоречивые выводы, возможно появление конкурирующих гипотез, и заключительные рекомендации системы будут зависеть от стратегии выбора правил (от способа разрешения коллизии). В результате система может сделать не те выводы, которые ожидал бы от нее эксперт, или вообще не прийти ни к какому решению. Последнее наиболее вероятно в том случае, когда помимо противоречий в базе правил присутствует также и неполнота. К категории противоречий можно отнести также ошибки заикливания, когда условие и заключение одного правила меняются местами в другом правиле.

### **Связность базы правил**

Понятие связности базы правил вводится с помощью определения.

База правил называется связной, если в ней нет смежных правил  $R_j, R_k$  таких, что пересечение содержащихся в их заключениях нечетких множеств является пустым, т. е.  $V_j \wedge V_k = 0$ , т.е. для любого  $u$ , принадлежащего области значений  $Y$  выходного параметра, выполняется соотношение:

$$\mu_{V_j}(u) \cdot \mu_{V_k}(u) \neq 0, \quad \forall u : u \in Y.$$

Под «смежными» следует понимать правила, задаваемые в смежных ячейках таблицы правил.

Связность базы правил повышает гладкость поверхности модели, а несвязность приводит к появлению участков резкого подъема или крутого спуска. Связность базы правил нечеткой модели (регулятора) зависит от вида поверхности моделируемой системы. Если поверхность имеет обширные

участки крутого спуска, то для нее невозможно получить точное представление с помощью модели, имеющей связную базу правил. Если же поверхность гладкая, то модель со связной базой правил будет точной. Чаще всего требуются гладкие поверхности отображения входных параметров в выходные, поскольку участки крутого спуска подразумевают сильные и резкие изменения управляющих переменных объекта. Вместе с тем, данное требование не является абсолютно необходимым, поскольку могут существовать и процессы (объекты, системы), в которых требуются резкие изменения управляющих переменных и, стало быть, регуляторы с несвязными базами правил.

**Избыточность правил.** Иногда встречаются нечеткие модели, содержащие два или более идентичных правила (т. е. правила, у которых совпадают условия и заключения). Причинами подобной ситуации могут быть:

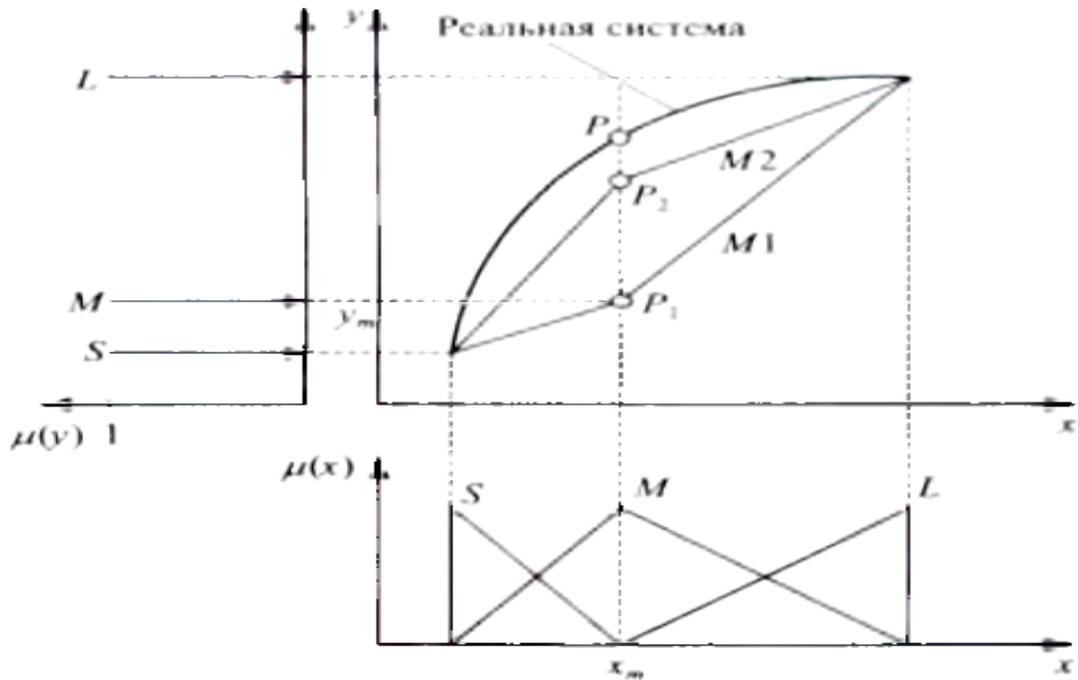
- 1) ошибка, допущенная при проектировании базы правил (при большом числе правил);
- 2) генерация дополнительных правил, похожих на имеющиеся, с целью усиления их заключений.

В первом случае избыточное правило следует исключить.

Во втором случае поверхность модели  $M_1$  в точке  $x = x_m$  значительно отличается от поверхности реальной системы (точки  $P_1$  и  $P_2$ ). Эта значительная ошибка возникла вследствие неправильного выбора параметра  $y_m$  выходной функции принадлежности  $\mu_M(y)$ . Самообучающаяся модель может сформировать дополнительное правило, совпадающее с  $R_2$ , и два одинаковых правила  $R_2$  можно заменить одним правилом  $R^*2$ , заключение которого имеет вид логической суммы:

$$(ЕСЛИ (x = M) ТО (y = M)) \cup (ЕСЛИ (x = M) ТО (y = M)) = \\ = ЕСЛИ (x = M) ТО (y = M \cup M).$$

При выполнении логического суммирования на основе оператора  $MAX$  будет получено множество  $M^* = M \cup M = M$  (Рис.5.3.3.).



База правил модели $M1$			
$R_i$	$R1$	$R2$	$R3$
$x$	$S$	$M$	$L$
$y$	$S$	$M$	$L$

База правил модели $M2$				
$R_i$	$R1$	$R2$	$R2$	$R3$
$x$	$S$	$M$	$M$	$L$
$y$	$S$	$M$	$M$	$L$

Рис. 2.3.3. Сравнение нечетких моделей  $M1$  и  $M2$ , соответственно без избыточности и с избыточностью базы правил

В случае использования других операторов, например SUM (суммирование функций принадлежности), результирующее множество  $M^*$  будет иметь вид  $M \cup M \neq M$ , приводящий к усилению получаемого заключения и уменьшения ошибки модели. Таким образом, несколько совпадающих правил можно заменить одним правилом, заключение которого соответствующим образом усилено.

## 2.4. База правил

База правил (лингвистическая модель) представляет собой множество нечетких правил  $R^{(k)}$ ,  $k = 1, \dots, N$ , вида

$$R^{(k)} : \mathbf{IF} (x_1 \text{ это } A_1^k \text{ AND } x_2 \text{ это } A_2^k \dots \text{ AND } x_n \text{ это } A_n^k)$$

$$\mathbf{THEN} (y_1 \text{ это } B_1^k \text{ AND } y_2 \text{ это } B_2^k \dots \text{ AND } y_m \text{ это } B_m^k),$$

где  $N$  - количество нечетких правил,  $A_i^k$  - нечеткие множества

$$A_i^k \in \mathbf{X}_i \subset R, \quad i = 1, \dots, n,$$

$B_j^k$  - нечеткие множества

$$B_j^k \subseteq \mathbf{Y}_j \subset R, \quad j = 1, \dots, m,$$

$x_1, x_2, \dots, x_n$  - входные переменные лингвистической модели, причем

$$(x_1, x_2, \dots, x_n)^T = x \in \mathbf{X}_1 \times \mathbf{X}_2 \times \dots \times \mathbf{X}_n,$$

$y_1, y_2, \dots, y_m$  - выходные переменные лингвистической модели, причем

$$(y_1, y_2, \dots, y_m)^T = y \in \mathbf{Y}_1 \times \mathbf{Y}_2 \times \dots \times \mathbf{Y}_m.$$

Символами  $\mathbf{X}_i$ ,  $i = 1, \dots, n$  и  $\mathbf{Y}_j$ ,  $j = 1, \dots, m$  обозначаются соответственно пространства входных и выходных переменных.

Конкретные правила  $R^{(k)}$ ,  $k=1, \dots, N$  связаны между собой логическим оператором «ИЛИ». Выходы  $y_1, y_2, \dots, y_m$  взаимно независимы. Поэтому используют нечеткие правила со скалярным выходом в форме

$$R^{(k)} : \mathbf{IF} (x_1 \text{ это } A_1^k \text{ AND } x_2 \text{ это } A_2^k \dots \text{ AND } x_n \text{ это } A_n^k)$$

$$\mathbf{THEN} (y \text{ это } B^k),$$

$$\text{где } B_j^k \subseteq \mathbf{Y}_j \subset R \text{ и } k=1, \dots, N.$$

Посылка (antecedent) правила содержит набор условий, а следствие (consequent) содержит вывод. Переменные  $x = (x_1, x_2, \dots, x_n)^T$  и  $y$  могут принимать как лингвистические (например, «малый», «средний», «большой»), так и числовые значения. Если ввести обозначения

$$\mathbf{X} = \mathbf{X}_1 \times \mathbf{X}_2 \times \dots \times \mathbf{X}_n,$$

$$A^k = A_1^k \times A_2^k \times \dots \times A_n^k,$$

то правило (3.229) можно представить в виде нечеткой импликации

$$R^{(k)} : A^k \rightarrow B^k, \quad k=1, \dots, N.$$

Правило представляют, как нечеткое отношение, определенное на множестве  $\mathbf{X} \times \mathbf{Y}$ , т.е.  $R^{(k)} \subseteq \mathbf{X} \times \mathbf{Y}$  - это нечеткое множество с функцией принадлежности

$$\mu_{R^{(k)}}(x, y) = \mu_{A^k \rightarrow B^k}(x, y).$$

При синтезе модулей нечеткого вывода оценивают достаточность количества нечетких правил, их непротиворечивость и наличие корреляции между отдельными правилами.

### Построение нечетких правил

Для упрощения, создают базу правил для нечеткой системы с двумя входами и одним выходом. При этом необходимы обучающие данные в виде множества пар

$$(x_1(i), x_2(i), d(i)), i = 1, 2, \dots,$$

где  $x_1(i)$ ,  $x_2(i)$  - переменные, подаваемые на вход модуля нечеткого вывода, а  $d(i)$  - ожидаемое (эталонное) значение выходной переменной. Задача заключается в формировании таких нечетких правил, чтобы синтезированный на их основе модуль вывода при получении входных переменных генерировал корректные выходные переменные.

Шаг 1. Разделение пространства входных и выходных переменных на области.

Минимальное и максимальное значения каждой переменной обычно известно, и по ним определяют интервалы, в которых находятся допустимые значения.

Например, для входной переменной  $x_1$  такой интервал обозначим  $[x_1^-, x_1^+]$ . Если значения  $x_1^-$  и  $x_1^+$  неизвестны, то можно воспользоваться обучающими данными (данными эксперта) и выбрать из них минимальное и максимальное значения

$$x_1^- = \min(x_1), \quad x_1^+ = \max(x_1).$$

Аналогично для переменной  $x_2$  определим интервал  $[x_2^-, x_2^+]$ , а для эталонного сигнала  $d$  - интервал  $[d^-, d^+]$ .

Каждый определенный таким образом интервал разделяют на  $(2N+1)$  областей (отрезков), причем значение  $N$  для каждой переменной подбирается индивидуально, а отрезки могут иметь одинаковую или различную длину. Отдельные области обозначают:  $M_N$  (Малый  $N$ ), ...,  $M_1$  (Малый 1),  $S$  (Средний),  $D_1$  (Большой 1), ...,  $D_N$  (Большой  $N$ ) и для каждого из них определяют одну функцию принадлежности. На рис. 2.4.1 представлен пример такого разделения, где область определения переменной  $x_1$  разбита на пять подобластей ( $N=2$ ), переменной  $x_2$  - на семь подобластей ( $N=3$ ), тогда как область определения выходной переменной  $y$  - на пять подобластей ( $N=2$ ).

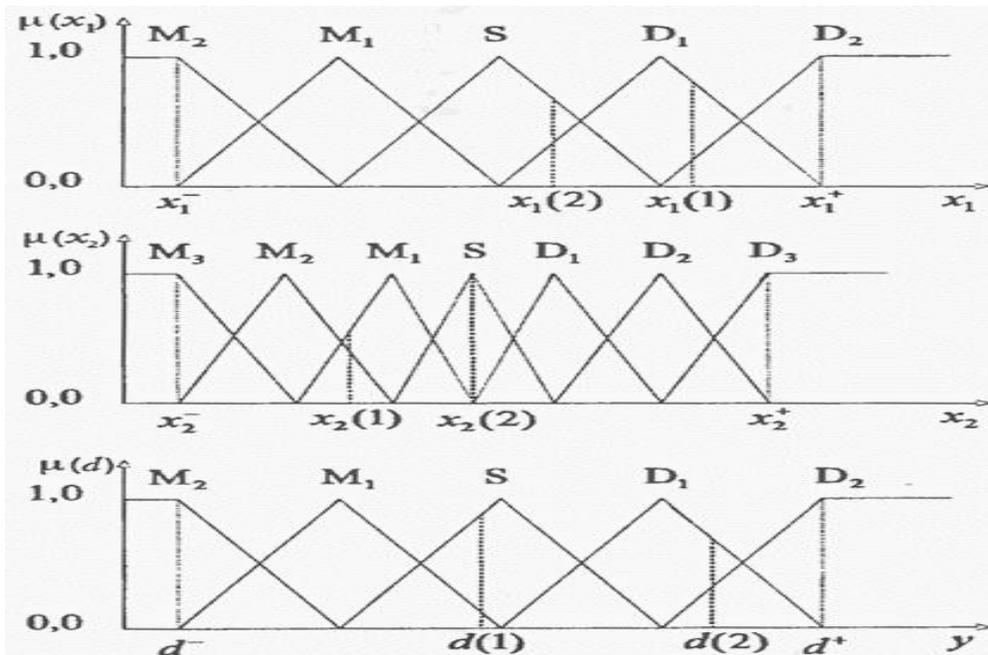


Рис. 2.4.1. Разделение пространств входных и выходных переменных на области и соответствующие им функции принадлежности.

Для каждой функция принадлежности возьмем треугольную форму; одна из вершин располагается в центре области и ей соответствует значение функции, равное 1. Две других вершины лежат в центрах соседних областей, им соответствуют значения функции, равные 0. Можно предложить много других способов разделения входного и выходного пространства на отдельные области и использовать другие формы функций принадлежности.

Шаг 2. Построение нечетких правил на основе обучающих данных.

Вначале определяют степени принадлежности обучающих данных  $(x_1(i), x_2(i)$  и  $d(i))$  к каждой области, выделенной на шаге 1. Эти степени будут выражаться значениями функций принадлежности соответствующих нечетких множеств для каждой группы данных. Для случая, показанного на рис. 2.4.1., степень принадлежности данного  $x_1(1)$  к области  $D_1$  составляет 0,8, к области  $D_2$  - 0,2, а к остальным областям - 0. Аналогично для данного  $x_2(2)$  степень принадлежности к области  $S$  составляет 1, а к остальным областям - 0. Затем сопоставляют обучающие данные  $x_1(i), x_2(i)$  и  $d(i)$  к областям, в которых они имеют максимальные степени принадлежности. Переменная  $x_1(1)$  имеет наибольшую степень принадлежности к области  $D_1$ , а  $x_2(2)$  - к области  $S$ . Для каждой пары обучающих данных записывают одно правило, т.е.

$$(x_1(1), x_2(1); d(1)) \Rightarrow$$

$$\{x_1(1)[\max: 0,8 \text{ в } D_1], x_2(1)[\max: 0,6 \text{ в } M_1]; d(1)[\max: 0,9 \text{ в } S]\} \Rightarrow$$

$$R1: \mathbf{IF} (x_1 \text{ это } D_1 \mathbf{AND} x_2 \text{ это } M_1) \mathbf{THEN} y \text{ это } S; \Rightarrow$$

$$(x_1(2), x_2(2); d(2)) \Rightarrow$$

$$\{x_1(2)[\max: 0,7 \text{ в } S], x_2(2)[\max: 1,0 \text{ в } S]; d(2)[\max: 0,7 \text{ в } D_1]\} \Rightarrow$$

R2: **IF**( $x_1$  это  $S$  **AND**  $x_2$  это  $S$ ) **THEN**  $y$  это  $D_1$ .

Шаг 3. Приписывание каждому правилу степени истинности.

Обычно в наличии имеется большое количество пар обучающих данных, и по каждой из них может быть сформулировано одно правило, поэтому некоторые из этих правил окажутся противоречивыми. Это относится к правилам с одной и той же посылкой (условием), но с разными следствиями (выводами). Приписывают каждому правилу степень истинности и выбирают правило с наибольшей степенью. Таким образом, значительно уменьшается общее количество правил. Для правила вида

R: **IF**( $x_1$  это  $A_1$  **AND**  $x_2$  это  $A_2$ ) **THEN** ( $y$  это  $B$ )

степень истинности, обозначаемая как  $SP(R)$ , определяется как

$$SP(R) = \mu_{A_1}(x_1) \cdot \mu_{A_2}(x_2) \cdot \mu_B(y).$$

Таким образом, первое правило  $(R^{(1)})$  из нашего примера имеет степень истинности

$$SP(R^{(1)}) = \mu_{D_1}(x_1) \cdot \mu_{M_1}(x_2) \cdot \mu_S(y) = 0,8 \times 0,6 \times 0,9 = 0,432,$$

а второе правило -

$$SP(R^{(2)}) = \mu_S(x_1) \cdot \mu_S(x_2) \cdot \mu_{D_1}(y) = 0,7 \times 1,0 \times 0,7 = 0,49.$$

#### Шаг 4. Создание базы нечетких правил.

Способ построения базы нечетких правил представлен на рис. 2.4.2. Эта база представляется матрицей, которая заполняется нечеткими правилами следующим образом: если правило имеет вид

$$R^{(1)}: \text{IF } (x_1 \text{ это } D_1 \text{ AND } x_2 \text{ это } M_1) \text{ THEN } y \text{ это } S,$$

то на пересечении строки  $D_1$  (соответствующей сигналу  $x_1$ ) и столбца  $M_1$  (сигнал  $x_2$ ) вписываем название нечеткого множества, присутствующего в следствии, т.е.  $S$  (соответствующего выходному сигналу  $y$ ). Если имеется несколько нечетких правил с одной и той же посылкой, выбирается с наивысшей степенью истинности.

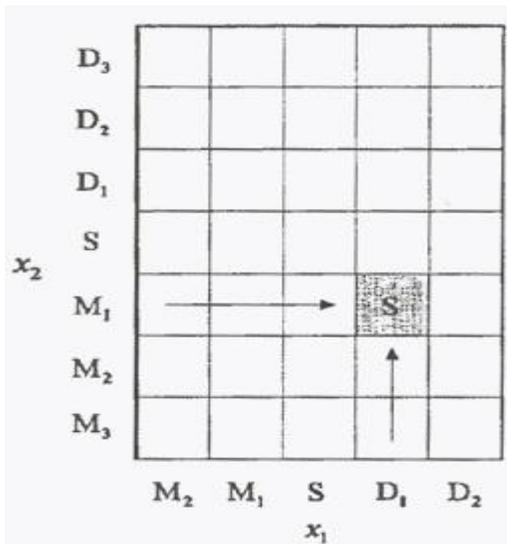


Рис. 4.4.3. Форма базы нечетких правил

На рис 2.4.4. представлена структурная схема алгоритма построения базы правил на основе численных данных.

5. Следующий шаг это дефаззификация.

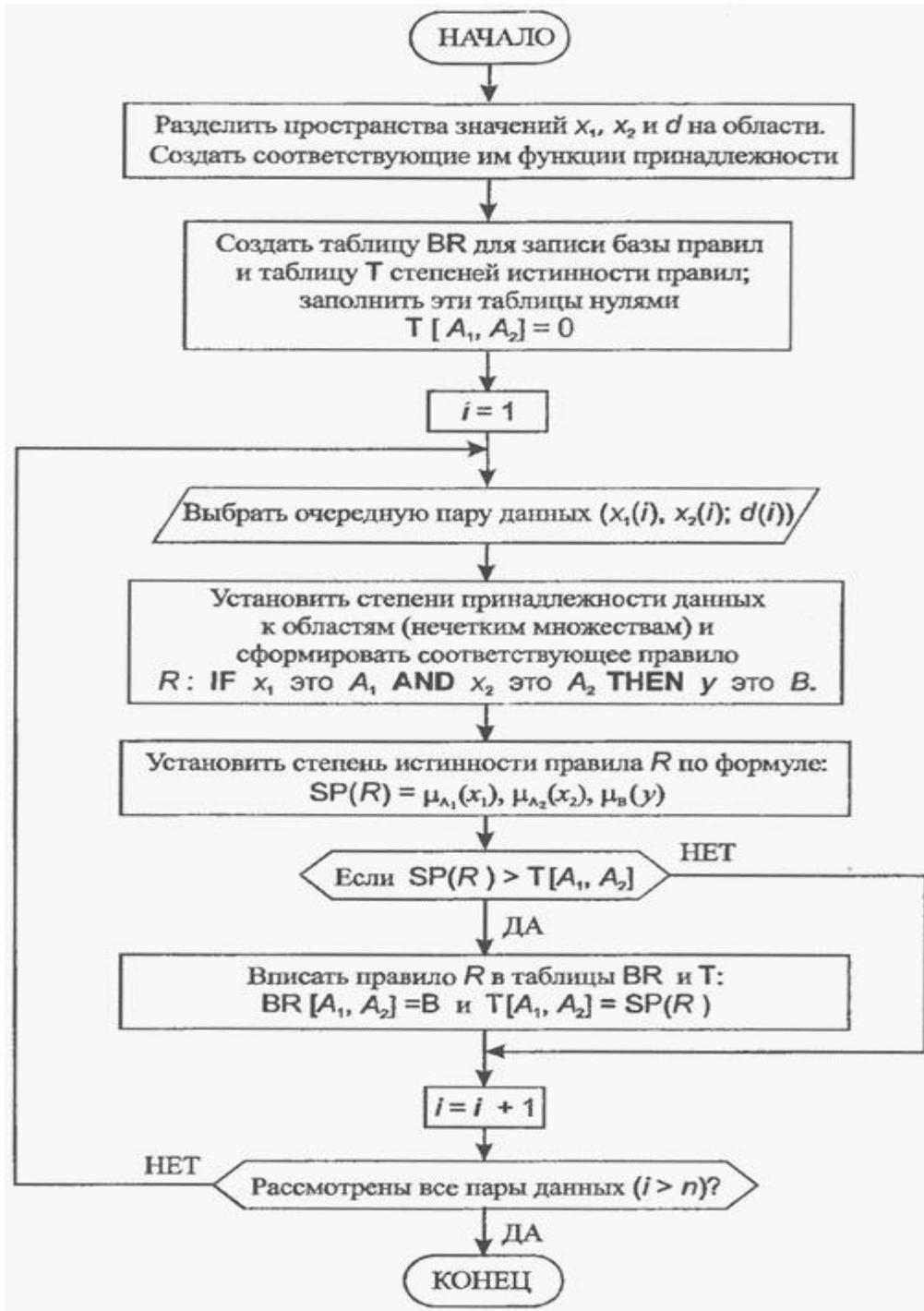


Рис. 2.4.4. Структурная схема алгоритма построения базы правил на основе численных данных

## 2.5. Нормирование (масштабирование) входов и выхода нечеткой модели

В реальных системах значения входов  $x_i$  и выхода  $y$  обычно имеют ограниченные пределы изменения (рис. 2.5.1).

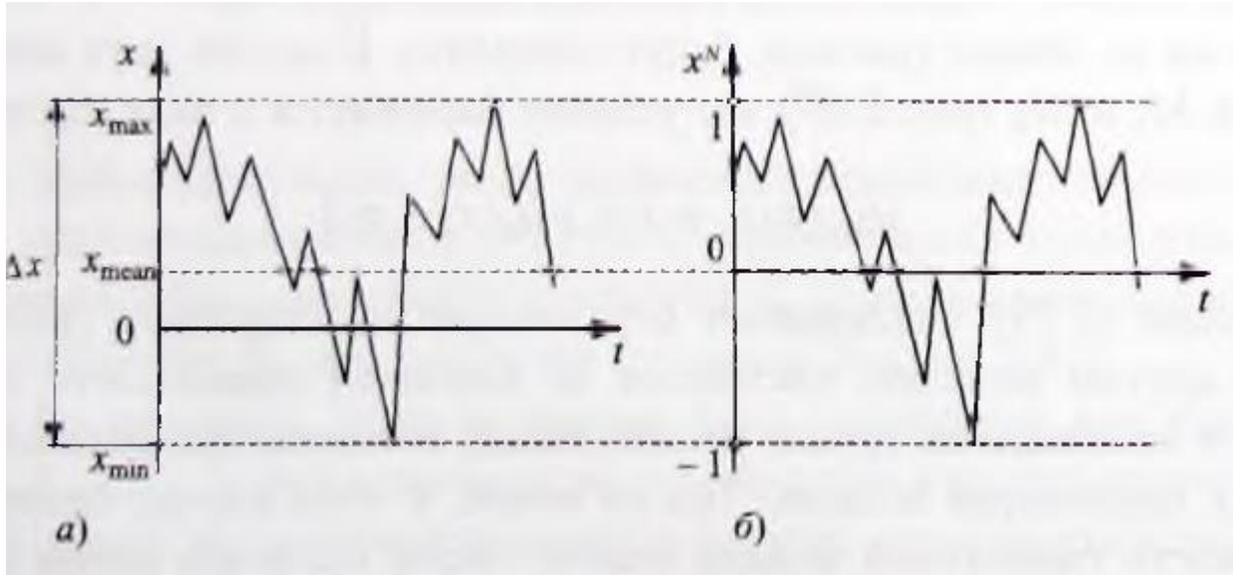


Рис. 2.5.1. График изменения реальной величины  $x$  в ходе функционирования системы (а) и график величины  $x_N$  после нормирования с использованием какого-либо метода (б)

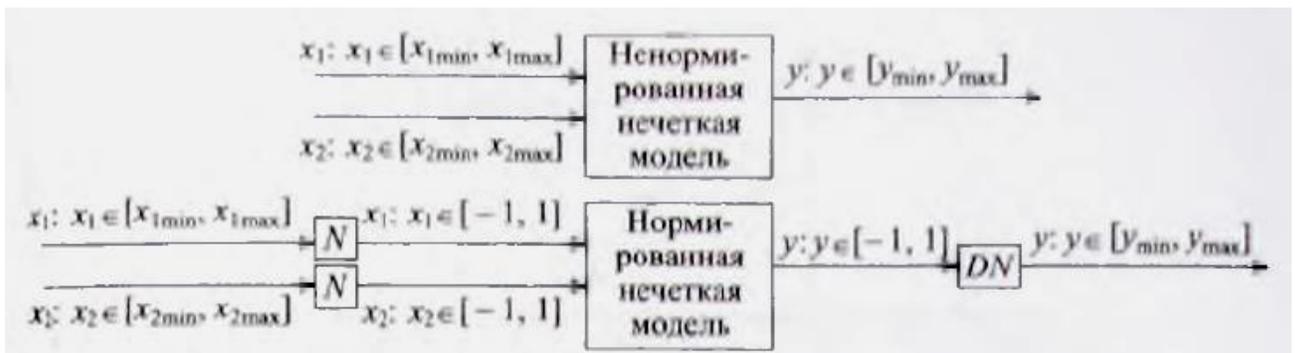


Рис. 2.5.2. Нормирование входов и выходов модели (N — операция нормирования,  $x^N$  — нормированная величина, DN — операция денормирования)

Ограниченность сигналов в системе может быть подтверждена результатами наблюдений и измерений, и если ограничения  $x_{max}$  и  $x_{min}$  известны, то выполняют их нормирование, называемое также масштабированием.

Нормирование величины, имеющей интервал изменения  $[x_{\min}, x_{\max}]$ , заключается в приведении его путем подходящего масштабирования к нормированному интервалу  $[-1,1]$ . Также может использоваться интервал  $[0,1]$ . Принцип нормирования сигналов нечеткой модели показан на рис. 2.5.2.. Преимущество нормирование:

1. Для реальных систем, являющихся подобными на качественном уровне, получаем сходные нормированные нечеткие модели, а при управлении подобными на качественном уровне процессами — сходные нормированные нечеткие регуляторы.
2. Приобретенный опыт проектирования моделей вывода для качественно подобных систем помогают разработчику быстро, создавать модели и регуляторы, которые затем требуют только настройки (при этом часто незначительной).

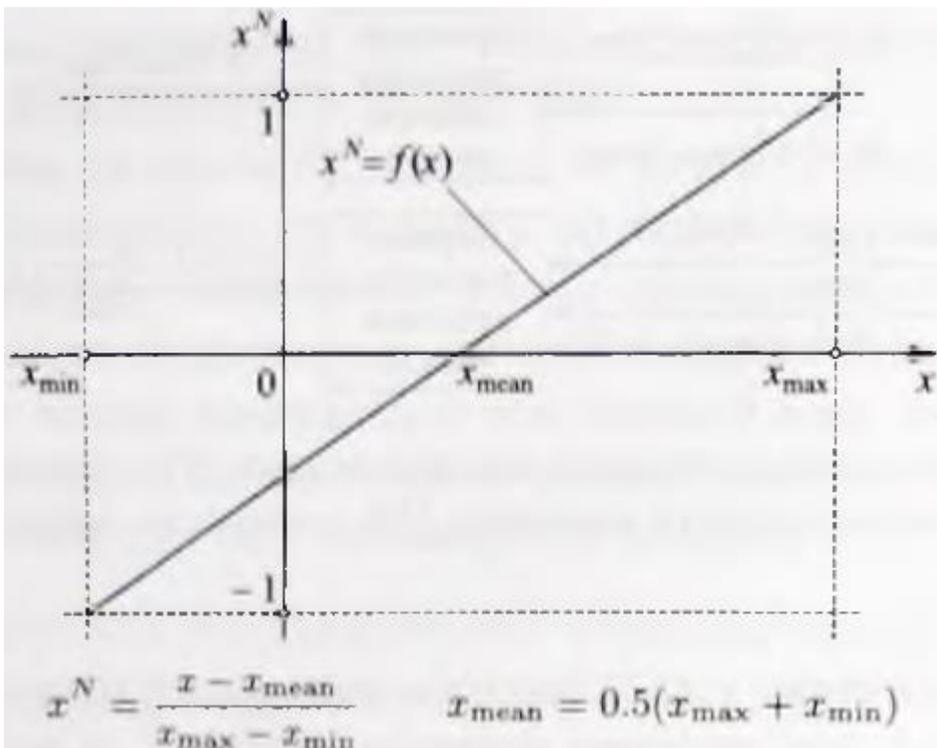


Рис.2.5.3. Нормирование величины  $x$  с использованием интервала  $[-1,1]$

1. Метод нормирования с использованием интервала  $[-1,1]$  (Рис.2.5.3.). Достоинство: использовании интервала  $[-1,1]$  целиком, Недостаток: нулевые значения величин  $x$  и  $x^N$  не совпадают, а совпадение иногда важно.

2. Упрощенный метод нормирования. Величина  $x$  делится на некоторый постоянный коэффициент (рис. 2.5.4.). Достоинство: меньшие вычислительные затраты, нормированная и ненормированная величины имеют общую точку отсчета (нулевое значение). Недостаток: интервал  $[-1,1]$  используется не полностью (рис. 2.5.3.), При  $x = x_{\min}/n$  значение нормированной величины  $x$  отлично от  $-1$ . В Упрощенный метод нормирования используется для симметричных интервалов изменения сигналов, если  $|x_{\min}| = |x_{\max}|$ .

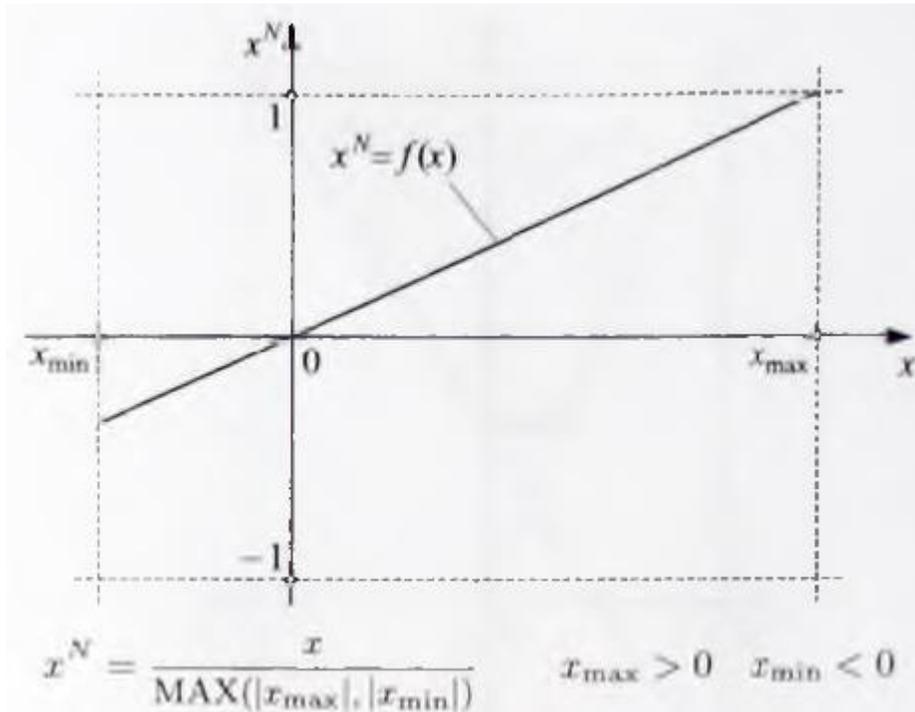


Рис.. 2.5.4. Упрощенное нормирование величины  $x$  с использованием интервала  $[-1,1]$

3. Метод нормирования с использованием интервала  $[0,1]$ . После вычисления с помощью нормированной нечеткой модели выходного значения  $y$ , также являющегося нормированным, необходимо выполнить его денормирование (рис.2.5.5.).

Денормирование представляет собой обратную по отношению к нормированию операцию, и для ее выполнения должны быть известны максимальное ( $y_{max}$ ) и минимальное ( $y_{min}$ ) выходные значения моделируемой системы.

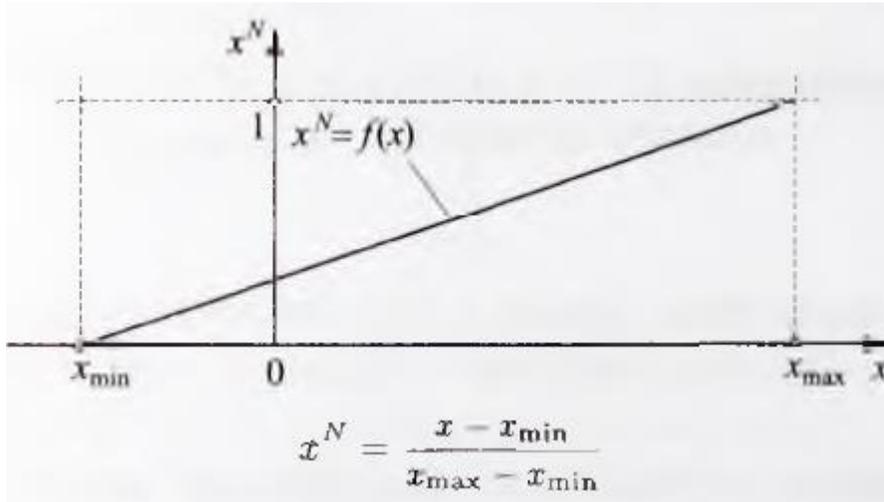
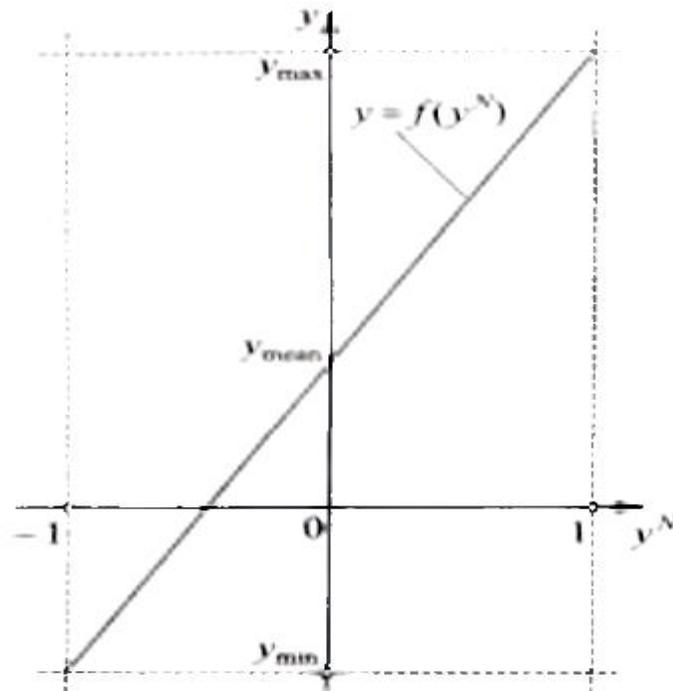


Рис. 2.5.5. Нормирование величины  $x$  с использованием интервала  $[0,1]$

Формулы денормирования получаются из формул нормирования.

1. Денормирование на интервале  $[-1,1]$ . Нормированное выходное значение  $y_N$  нормированной модели имеет симметричный интервал изменения  $[-1,1]$ , в то время как интервал  $[y_{min} \wedge y_{max}]$  изменения денормированного выходного значения  $y$  часто является несимметричным. Поэтому, в противоположность нормированию, возникает необходимость отображения симметричного интервала. Преимущество метода денормирования, изображенного на рис. 2.5.6., состоит в полном использовании выходного диапазона модели, а недостаток связан с несовпадением нулевых значений в шкалах  $y_{max}$  и  $y_{min}$ .
2. Для выполнения упрощенного нормирования могут использоваться два метода, представленные на рис. 2.5.6 и 2.5.7. В случае денормирования с большим коэффициентом преобразования (рис. 2.5.6) при  $y = -1$  на выходе нечеткой модели будет получено значение  $y = -y_{max}$ , лежащее за пределами интервала изменения  $[y_{min}, y_{max}]$  реальной системы, и тем самым, нечеткая модель будет вычислять несуществующие выходные значения. Такое

денормирование вводит в систему управления дополнительную нелинейность в виде ограничения (насыщения) сигнала.



$$y = u^N \cdot (y_{\max} - y_{\min}) + y_{\text{mean}} \quad y_{\text{mean}} = 0.5(y_{\max} + y_{\min})$$

Рис.2.5.6. Денормирование  $u_N \rightarrow y$  из интервала  $u_N: [-1,1]$  с полным использованием интервала  $y: [y_{\min}, y_{\max}]$ .

- упрощенное денормирование с меньшим коэффициентом преобразования (рис. 2.5.7). Значение, выдаваемое нечеткой моделью при  $u = 1$ , отличается от максимального, выдаваемого реальной системой (представление этой системы нечеткой моделью не будет точным).

- упрощенное денормирование с симметричным интервалом изменения  $[y_{\min}, y_{\max}] > T$ . е. если  $y_{\min} = -y_{\max}$ . На рис. 2.5.7. показано денормирование из интервала  $y: [0, 1]$ .

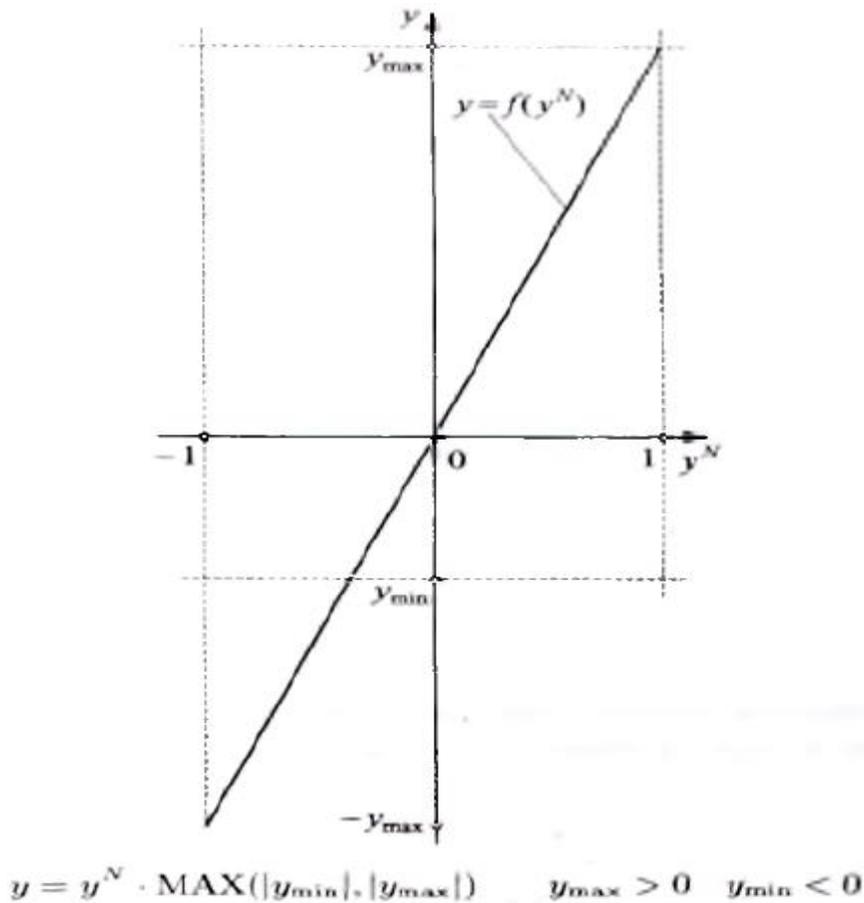


Рис.2.5.7.. Упрощенное денормирование из интервала  $y$ :  $[-1,1]$  с коэффициентом преобразования, равным  $\text{MAX}(|y_{min}|, |y_{max}|)$ .

Если база правил нечеткой модели задана, то процесс настройки модели состоит в выборе подходящих значений для каждого отдельного нечеткого множества. При изменении этих значений в интервале  $[-1,1]$ , изменяются коэффициенты передачи нечеткой модели, и вследствие ограниченности диапазона изменения указанным интервалом становится легче понять процесс настройки нечеткой модели и приобрести опыт в данной области. Выполнять нормирование модели во всех возможных случаях нет необходимости. Тем не менее, оно применяется во многих профессиональных системах нечеткого моделирования и управления.

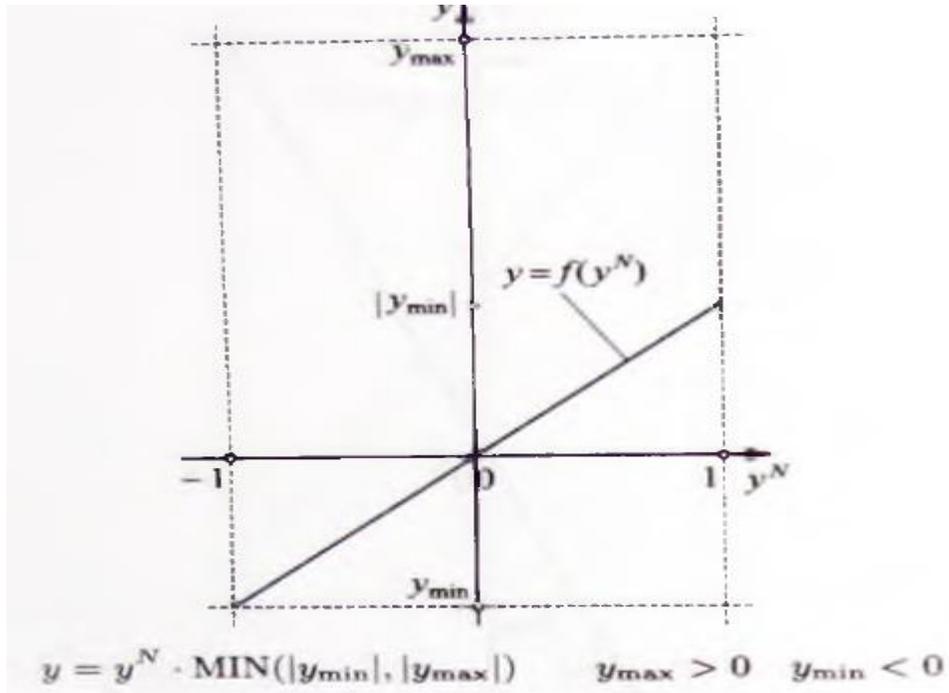


Рис. 2.5.8. Упрощенное денормирование из интервала с коэффициентом преобразования, равным заданной переменной.

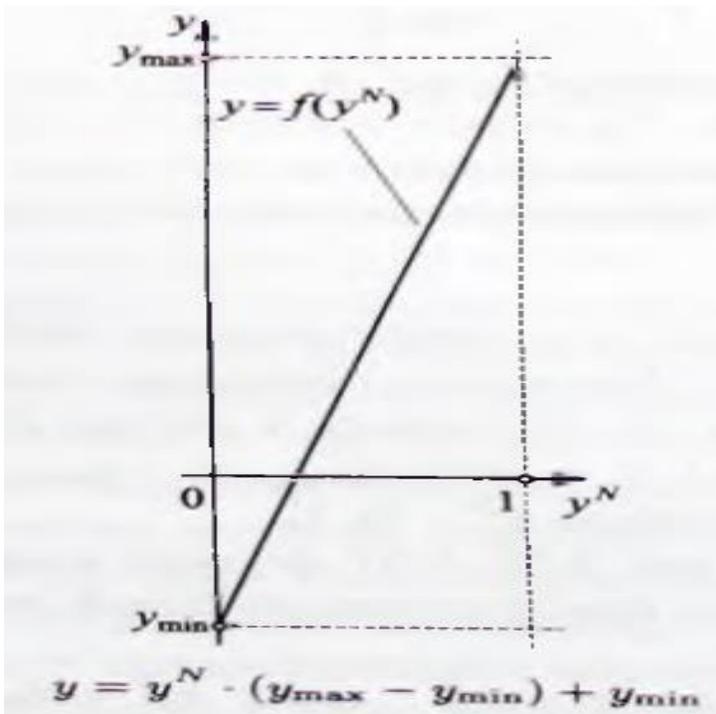


Рис. 2.5.9. Денормирование из интервала в интервал

## 2.6. Нечеткая модель Мамдани

Развитие нечеткой логики приводит к появлению нечетких моделей, целью создания которых является обеспечение большей точности и размерности, а также упрощение их структуры. Необходимость разработки моделей вызвана разнообразием требующих моделирования реальных систем, различными видами информации об этих системах и разной степенью ее доступности.

Основное преимущество нечетких моделей, по сравнению с традиционными аналитическими моделями, связано с возможностью использования для их разработки значительно меньших объемов информации о системе, при этом информация может носить приближенный, нечеткий характер. Важнейшим и наиболее часто используемым типом нечеткой модели является модель Мамдани.

Лингвистическая нечеткая модель, воспроизводящая человеческий способ мышления, предложена в первых работах Л.Заде. К нечеткому управлению динамическими объектами лингвистическую модель применил Мамдани (Mamdani 1974,1977), представивший также способ построения модели человека-оператора, управляющего объектом. Предложенный Мамдани метод моделирования оказался простым и доступным, и поэтому используется чаще всего, хотя разработаны и другие типы моделей. В методе Мамдани моделируемая система рассматривается как черный ящик, характеризующийся недостаточностью информации о происходящих внутри него физических явлениях

В модели Мамдани, выполняющей такое отображение входов (вектор  $X$ ) в выход  $Y$ , необходимо обеспечить точную аппроксимацию реальной системы. Указанное отображение предполагает существование некоторой геометрической поверхности (поверхности отображения) в пространстве, задаваемом декартовым произведением  $X * Y$ .

Для этого модель Мамдани представляет собой множество правил, где каждое правило задает в указанном пространстве некоторую точку. На основе множества нечетких точек формируется нечеткая поверхность, интерполяции между точками в котором зависят от используемого аппарата нечеткой логики.

$$y = (x - 2)^2 + 1$$

Пусть моделируемая система реализует отображение

(рис. 2.6.1). Нечеткую модель Мамдани можно представить в виде множества правил

$R1$  : ЕСЛИ ( $x = A_1$ ) ТО ( $y = B_1$ ),

$R2$  : ЕСЛИ ( $x = A_2$ ) ТО ( $y = B_2$ ),

$R3$  : ЕСЛИ ( $x = A_3$ ) ТО ( $y = B_3$ ),

где  $A_1$  — примерно 1,  $A_2$  = примерно 2,  $A_3$  = примерно 4,

$B_1$  = примерно 2,  $B_2$  = примерно 1,  $B_3$  = примерно 5,  $x : 1 < x < 4$ .

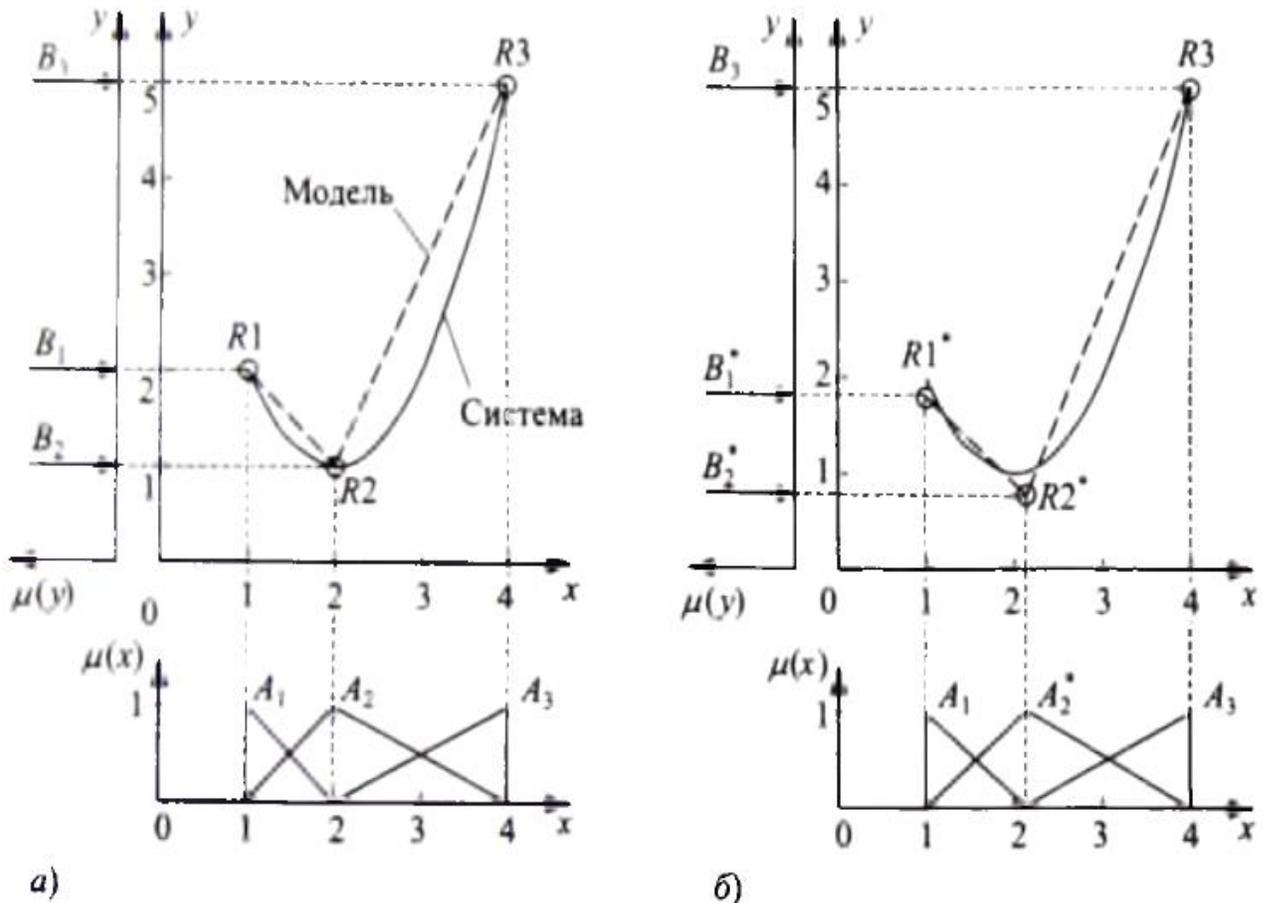


Рис.2.6.1. Расположение точек на нечеткой модели

Используемые в правилах функции принадлежности показаны на рис. 2.6.1, а. Каждое правило определяет особенность поведения системы, геометрически соответствующую точке пространства  $X*Y$ . Точки модели располагаются на характеристике реальной системы (рис.2.6.1.) и являются точками ее пересечения с характеристикой модели. В частности, правило  $R1$ , имеющее вид ЕСЛИ ( $x$  примерно 1) ТО ( $y$  примерно 2), задает точку  $R1$ , являющейся существенной одновременно для системы и ее модели.

Точки нечеткой модели могут не принадлежать характеристике (поверхности) реальной системы, но другое расположение этих точек может обеспечить более высокую точность модели. В данной ситуации параметры функций принадлежности изменяются (что в свою очередь приводит к новым нечетким множествам  $A^*_2, B^*_1, B^*_2$ ), и таким образом, правила будут иметь следующий вид

$$\begin{aligned} R1^* &: \text{ЕСЛИ } (x = A_1) \text{ ТО } (y = B_1^*), \\ R2^* &: \text{ЕСЛИ } (x = A_2^*) \text{ ТО } (y = B_2^*), \\ R3 &: \text{ЕСЛИ } (x = A_3) \text{ ТО } (y = B_3). \end{aligned} \quad (2.6)$$

Правила  $R^*1$  и  $R^*2$  в (2.6) “не сообщают правду” о системе, поскольку задаваемые ими точки не принадлежат ее характеристике. Вместе с тем, средняя точность здесь будет выше, чем в случае модели, изображенной на рис. 2.6.1, а. Вид характеристики нечеткой модели на участках между «существенными» точками, для задания каждой из которых используется отдельное правило, зависит от используемого аппарата нечеткой логики (т. е. методов выполнения фаззификации, дефаззификации и т. д.). Если ввести другую функцию принадлежности множества  $A_2$ , то вид характеристики модели изменится (рис. 2.6.2,б). Как показано на рис. 2.6.2, введение трапецевидной функции принадлежности множества  $A_2$  приводит к изменению типа интерполяции, выполняемой моделью на участках между ее «существенными» точками

$R1, R2, R3$  — интерполяция имеет нелинейный характер, но при этом является локально линейной. Использование на участках между «существенными» точками модели Мамдани нелинейной интерполяции может привести к повышению точности модели, из-за того, что характер изгиба поверхности модели между данными точками будет совпадать с характером изгиба поверхности системы.

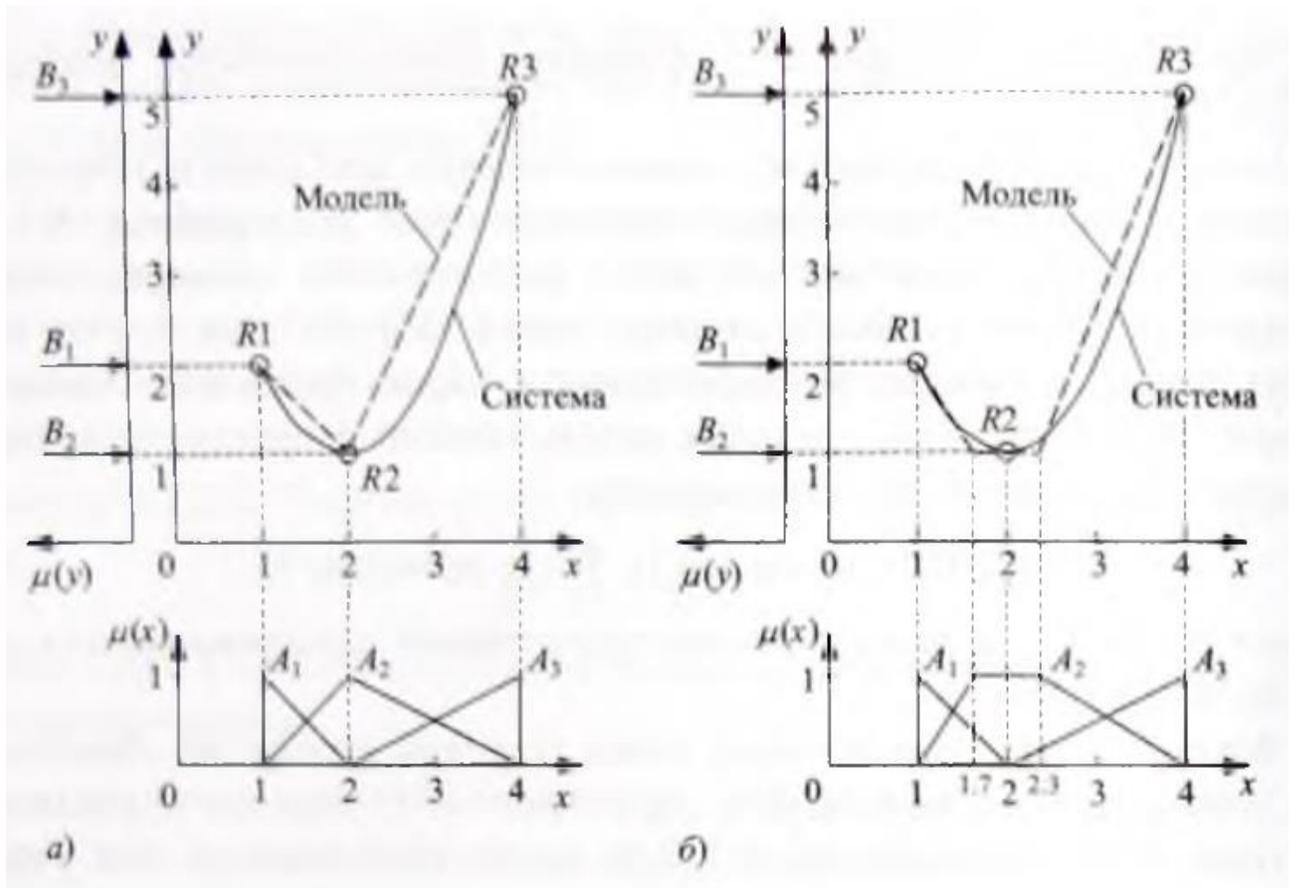


Рис. 2.6.2. Влияние изменения функции принадлежности нечеткого множества  $A2$  на вид характеристики модели

На практике характер выпуклости указанной поверхности в общем случае неизвестен, и имеется лишь небольшой объем информации о координатах отдельных точек, для которых выполнялись измерения. Кроме того, в случае нечеткой модели на характер выпуклости поверхности влияет столь большое

число элементов модели, что предугадать тип локальной выпуклости оказывается достаточно сложным, особенно для систем со многими входами.

Влияние типа оператора (используемого для агрегации условий в модели системы с двумя входами) на вид поверхности нечеткой модели.

Обычно для одного и того же входного вектора для различных операторов выходные значения будут также различными, хотя и могут совпадать в отдельных точках, например, в тех, которые задаются условиями правил, т. е. в «существенных» точках модели. Процедуры интерполяции, выполняемые моделью для операторов PROD и MIN будут различными. Поверхность нечеткой модели, соответствующая оператору PROD, является более гладкой, чем в случае оператора MIN. Чтобы определить, какая модель является более точной, необходимо иметь тестовое множество измерений векторов значений входных и выходных параметров моделируемой системы, на основе которого следует найти величину средней (либо квадратичной) ошибки.

Для каждого сегмента входного пространства  $X_1$  и  $X_2$  участки поверхности будут задаваться по-разному. Для оператора PROD, являющегося нелинейным, интерполяционные поверхности являются линейными.

Используя непрерывно дифференцируемые функции принадлежности входных параметров модели (например, гауссовы функции), можно обеспечить непрерывность первой производной (а также производных более высоких порядков) для данной поверхности, при условии, что не используются операторы типа MIN, связанные с возможностью резкого изменения значений.

Для любой нечеткой модели можно теоретически вывести формулу, задающую ее поверхность в явном виде (т. е.  $y = f(X)$ ). Вместе с тем, на практике эти формулы не выводят в силу трудоемкости данного процесса, которая значительно возрастает с увеличением числа входов и функций принадлежности. Значение на выходе нечеткой модели получают путем последовательного вычисления выходных значений отдельных ее элементов

при заданном векторе входных значений  $X$ . С учетом этого, на практике, при применении модели Мамдани, точные формулы, задающие интерполяционную поверхность, неизвестны и приводит к получению различных нелинейных поверхностей со сложной формой определения.

### **Лингвистические и нелингвистические модели Мамдани**

В самом начале практического использования в моделях Мамдани применялись только лингвистические переменные типа «малый» или «большой». Нечеткие модели, использующие подобные переменные для обозначения нечетких множеств, называются лингвистическими моделями. На практике, присвоение нечетким множествам лингвистических переменных часто оказывается лишенным особого смысла. Нечеткое множество «примерно 9» находится настолько близко к множеству «примерно 10», что считать его «средним» можно лишь с большим трудом. Наличие большого числа множеств приводит к использованию большого числа переменных (например, «малый», «близкий к малому», «между малым и средним», «близкий к среднему», «средний» и т. д.), делает их трудноразличимыми. Лучше использовать термины в виде нечетких чисел («примерно 1», «примерно 2», «примерно 3» и т. п.), поскольку с их помощью проще представить позицию каждого множества.

При малом числе нечетких множеств их ядра находятся на достаточно большом расстоянии друг от друга. Нечеткие модели, в которых метки множеств представлены в форме нечетких чисел, называются нелингвистическими моделями.

### **Для нечетких моделей правила «сообщают правду», если:**

- для реализации операции И используются  $t$ -нормы, а для операции ИЛИ -  $s$ -нормы, т. е. такие операторы, для которых выполняются условия  $\mu_{A \cap B}(x) = 0$  и  $\mu_{A \cup B}(y) \leq 1$  (этим условиям не удовлетворяют, например, операторы SUM (арифметическая сумма) и MEAN),

• в качестве функций принадлежности входных нечетких множеств используются функции, имеющие конечный носитель (данному условию не удовлетворяют гауссовы функции) и принимающие нулевое значение в точках, соответствующих модальным значениям ( $\mu(x) = 1$ ) смежных с ними функций.

На рис.2.6.3. приведены примеры функций, позволяющие достичь эффекта «сообщения правды» правилами, а на рис. 2.6.4, которые не позволяют

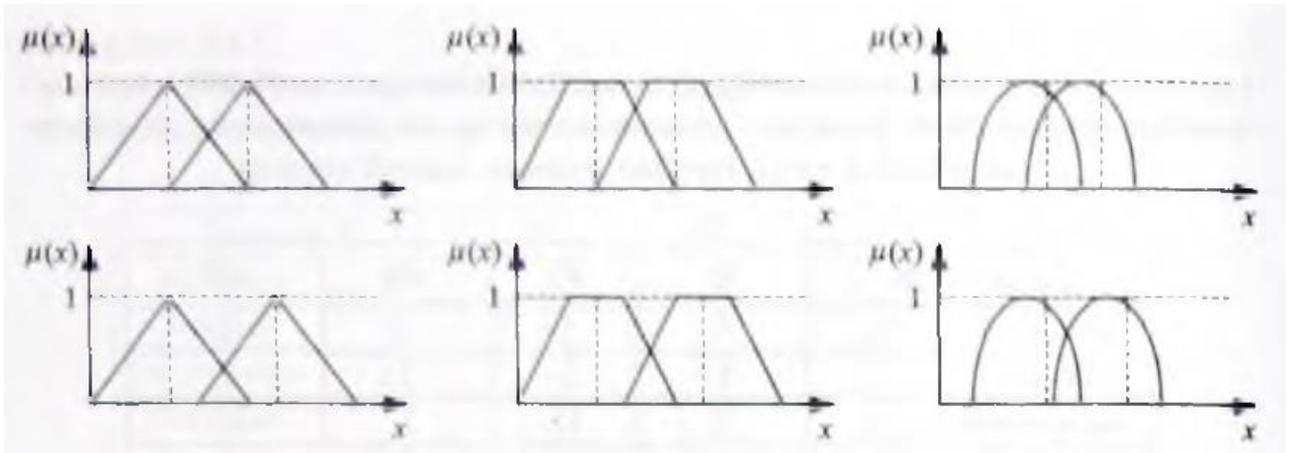


Рис. 2.6.3. Функции, позволяющие достичь эффекта «сообщения правды»

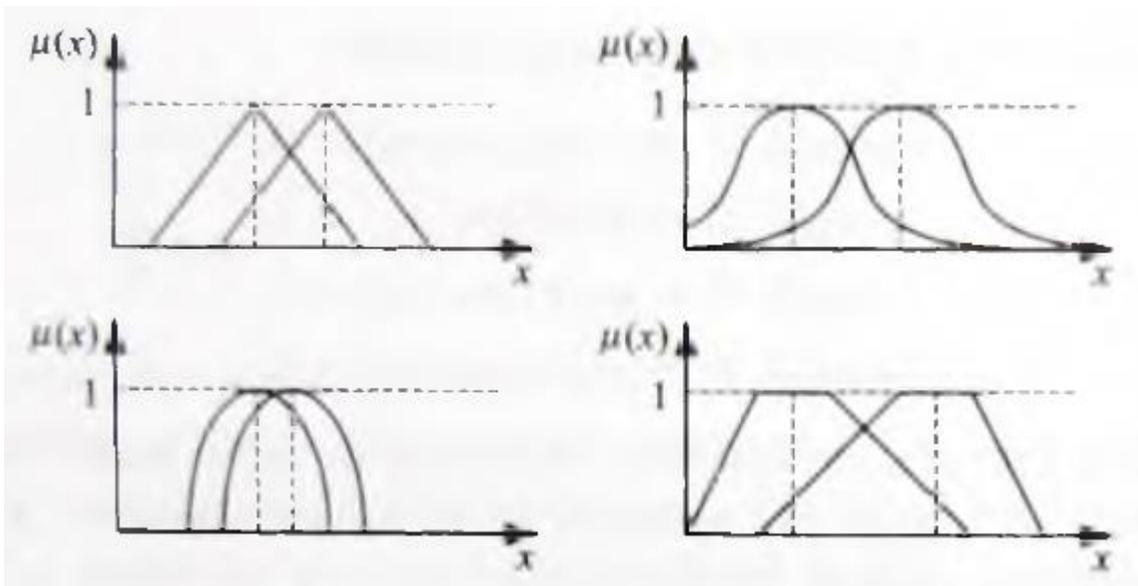


Рис. 2.6.4. Функции, не позволяющие достичь эффекта «сообщения правды»

Влияние удаленных (не являющихся смежными) опорных точек модели Мамдани на локальную интерполяцию. Существуют два способа построения нечеткой модели.

1. На форму интерполяционной поверхности, соответствующей заданному сегменту входного пространства, влияют только те опорные точки, которые непосредственно примыкают к данному сегменту.
2. Влияние могут оказывать и некоторые другие (а иногда даже все), а не только непосредственно примыкающие точки. Настройка параметров функций принадлежности (адаптация модели на основе измерений данных о реальной системе) становится значительно сложнее.

Локальные интерполяционные поверхности, расположенные между каждой парой смежных опорных точек, зависят не только от этих, но и от всех остальных опорных точек, поскольку в каждой из интерполяционных формул присутствуют все координаты  $u$ , задаваемые заключениями правил

Анализ результатов использования функций принадлежности, носители которых выходят за пределы, задаваемые модальными значениями смежных функций принадлежности, позволяет сделать следующие выводы.

- Интерполяционная поверхность модели имеет тот же качественный тип (в рассматриваемом примере линейный), что и в случае функций принадлежности, не выходящих за пределы модальных значений смежных с ними функций. Таким образом, расширение носителей функций принадлежности не изменяет свойств интерполяции.
- Настройка параметров модели должна происходить не локально, с использованием только непосредственно прилегающих опорных точек, а глобально, с использованием всех опорных точек системы. Учитывая явление, называемое «проклятием размерности», глобальная настройка значительно сложнее локальной.

- Полученная после завершения процесса настройки модель является корректной (поверхность модели проходит через опорные точки  $R_i$  системы), но при этом ее правила «не сообщают правду», что делает модель более сложной для понимания.

Таким образом, использование в моделях Мамдани функций принадлежности, носители которых не выходят за пределы, задаваемые смежными с ними функциями (в частности, функций, удовлетворяющих условию разбиения единицы), является более предпочтительным.

## 2.7. Нечеткая модель Такаги—Сугено

Впервые модели Такаги—Сугено (TS-модели) были описаны в (Takagi 1985). Эти модели также называют моделями Такаги—Сугено—Канга (Nquyen 1995; Yager 1994,1995), квазилинейными моделями и нечеткими линейными моделями (Babuska 1995a,b). От моделей Мамдани модели Такаги—Сугено отличаются формой правил. Если в случае модели Мамдани, описывающей систему с одним входом и одним выходом, правила имели вид

ЕСЛИ (x есть A) ТО (y есть B)

(где A, B — нечеткие множества типа «малый» или «близкий к 5»), то в случае TS-модели правила имеют вид

ЕСЛИ (x есть A) ТО ( $y = f(x)$ )

Вместо нечеткого множества заключение каждого правила содержит функцию  $f(x)$ , которая может быть нелинейной, хотя обычно используются линейные функции. Таким образом, правила TS-модели имеют вид

ЕСЛИ (x есть A) ТО ( $y = ax + b$ )

Если в модели база правил имеет вид

R1: ЕСЛИ (x есть  $A_1$ ) ТО ( $y = f_1(x)$ )

.....

Rm : ЕСЛИ (x есть  $A_m$ ) ТО ( $y = f_m(x)$ )

то значение на выходе модели вычисляется на основе степеней активизации отдельных заключений,  $f_i$ ,  $i = 1, \dots, m$ , в соответствии с формулой

$$y = \frac{\sum_{i=1}^m \mu_{A_i}(x) f_i(x)}{\sum_{i=1}^m \mu_{A_i}(x)}.$$

Рассмотрим TS-модель с базой правил вида (5.120) и функциями принадлежности, представленными на рис. 2.7.1:

$$\begin{aligned} R1: & \text{ ЕСЛИ } (x \text{ есть } A_1) \text{ ТО } (y = -x + 3), \\ R2: & \text{ ЕСЛИ } (x \text{ есть } A_2) \text{ ТО } \left( y = \frac{4x - 10}{3} \right), \\ R3: & \text{ ЕСЛИ } (x \text{ есть } A_3) \text{ ТО } \left( y = \frac{-x + 24}{3} \right). \end{aligned} \quad (2.7)$$

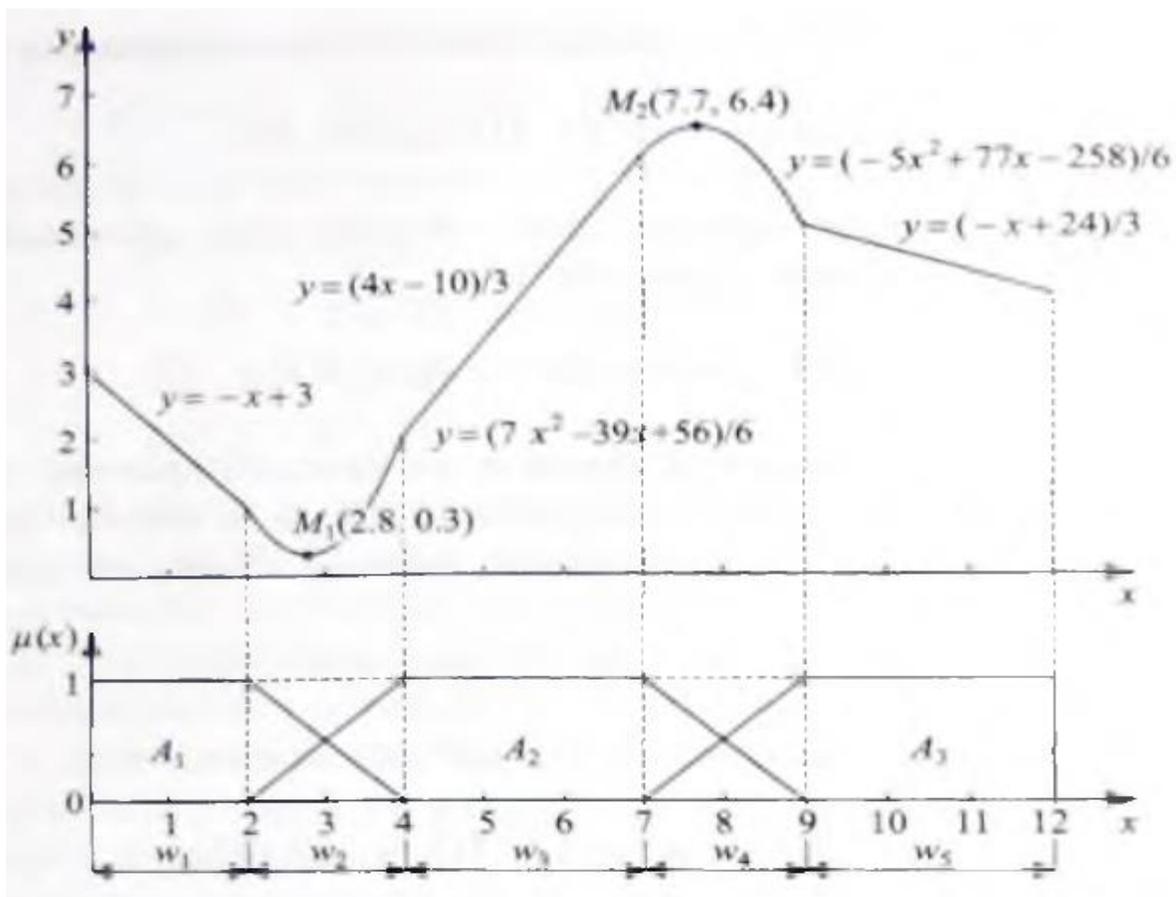


Рис.2.7.1. Функции принадлежности входных параметров и поверхность отображения для рассматриваемой модели Такаги—Сугено

С помощью формул (2.7) введем следующие логические переменные:

$$w_1 = \begin{cases} 1 & \text{для } 0 \leq x < 2, \\ 0 & \text{в других случаях,} \end{cases}$$

$$w_2 = \begin{cases} 1 & \text{для } 2 \leq x < 4, \\ 0 & \text{в других случаях,} \end{cases}$$

$$w_3 = \begin{cases} 1 & \text{для } 4 \leq x < 7, \\ 0 & \text{в других случаях,} \end{cases}$$

$$w_4 = \begin{cases} 1 & \text{для } 7 \leq x < 9, \\ 0 & \text{в других случаях,} \end{cases}$$

$$w_5 = \begin{cases} 1 & \text{для } 9 \leq x \leq 12, \\ 0 & \text{в других случаях.} \end{cases}$$

Тогда функции принадлежности будут иметь вид

$$\mu_{A_1} = w_1 - 0.5(x - 4)w_2,$$

$$\mu_{A_2} = 0.5(x - 2)w_2 + w_3 - 0.5(x - 9)w_4,$$

$$\mu_{A_3} = 0.5(x - 7)w_4 + w_5.$$

Рассматриваемые функции принадлежности удовлетворяют условию разбиения единицы:

$$\sum_{i=1}^3 \mu_{A_i}(x) = 1.$$

Значение на выходе модели определяется по формуле:

$$y = \sum_{i=1}^3 \mu_{A_i}(x) f_i(x) = w_1(-x + 3) + \frac{w_2(7x^2 - 39x + 56)}{6} +$$

$$+ \frac{w_3(4x - 10)}{3} + \frac{w_4(-5x^2 + 77x - 258)}{6} + \frac{w_5(-x + 24)}{3}. \quad (2.8.)$$

Анализ формулы (2.8) позволяет сделать вывод, что поверхность модели в точности соответствует заключениям правил только в тех областях входного пространства, степени принадлежности элементов которых соответствующим множествам  $A_i$ , удовлетворяют условию

$$\mu_{A_i}(x) = 1$$

(области  $w_1, W_3, w_5$ ). В областях, где эти степени являются дробными, поверхность модели переходит из одной линейной формы (задаваемой соответствующим заключением) в другую. Ширина областей перехода определяется шириной областей дробных значений функций принадлежности ( $\mu_{A_i}(x) < 1$ ), а математическое выражение функций перехода зависит от типа используемых функций принадлежности.

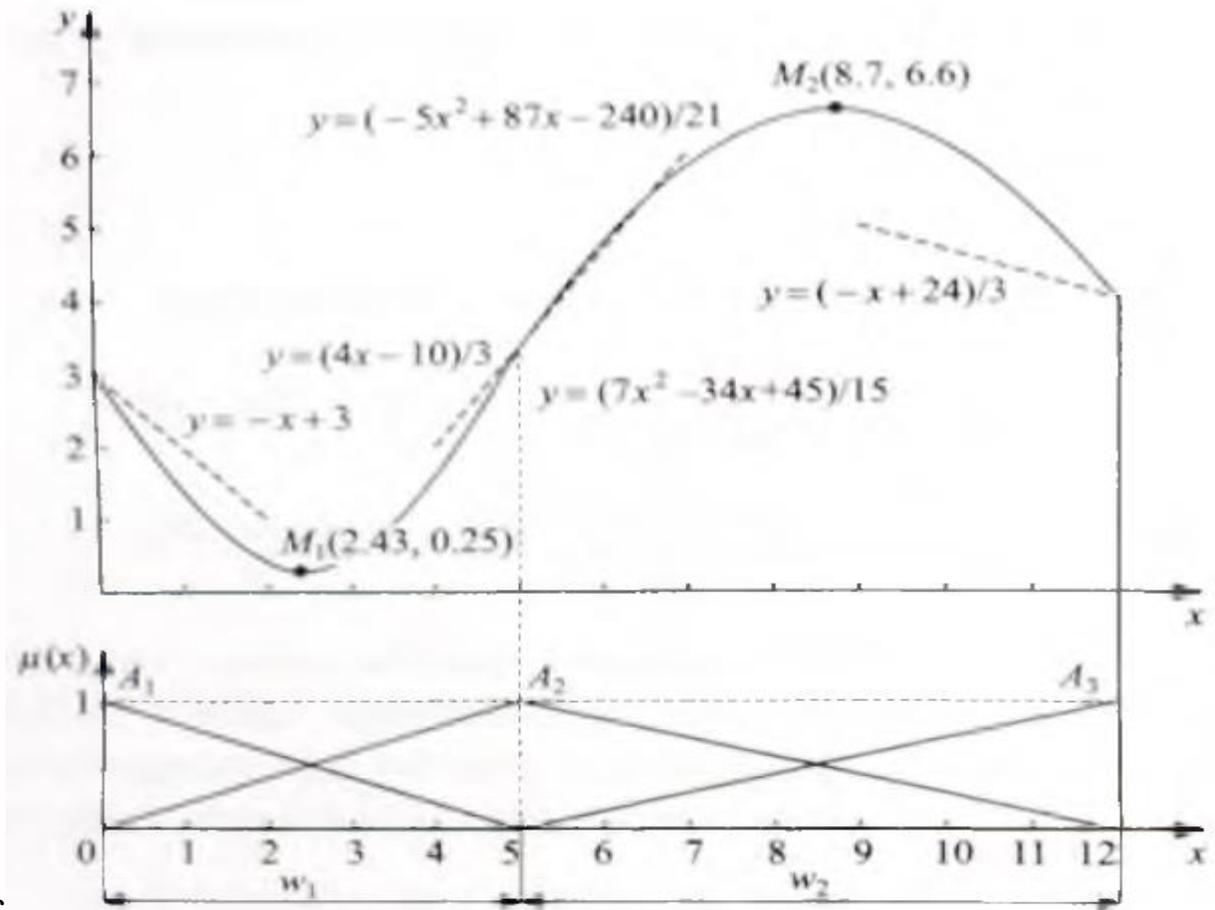
Функции перехода имеют квадратичную форму. Характеризующая модель 4) функция непрерывна и не имеет скачков на границах областей. Вместе с тем, производная этой функции непрерывной не является и изменяется на границах областей скачкообразно, что является следствием типа используемых функций принадлежности (их кусочной линейности). Непрерывность производной поверхности модели и, тем самым, большую гладкость последней можно обеспечить путем использования непрерывно дифференцируемых функций принадлежности, например гауссовых функций.

В данном примере использованы трапециевидные функции принадлежности, имеющие зоны четкости, характеризующиеся тем, что их элементы полностью (со степенью, равной 1) принадлежат соответствующему множеству. в

Поверхность модели с той же базой правил, но при условии использования других типов функций принадлежности будет отличаться.

Рассмотрим TS-модель системы типа SISO с базой правил вида и треугольными функциями принадлежности, представленными на рис. 2.7.2:

$R1$  : ЕСЛИ ( $x$  есть  $A_1$ ) ТО ( $y = -x + 3$ ),  
 $R2$  : ЕСЛИ ( $x$  есть  $A_2$ ) ТО ( $y = (4x - 10)/3$ ),  
 $R3$  : ЕСЛИ ( $x$  есть  $A_3$ ) ТО ( $y = (-x + 24)/3$ ).



ис

Рис.2.7.2. Пример модели Такаги—Сугено с треугольными (не трапециевидными) функциями принадлежности

Определим логические переменные  $w_1$ ,  $w_2$ :

$$w_1 = \begin{cases} 1 & \text{для } 0 \leq x < 5, \\ 0 & \text{в других случаях,} \end{cases}$$

$$w_2 = \begin{cases} 1 & \text{для } 5 \leq x \leq 12, \\ 0 & \text{в других случаях.} \end{cases}$$

Получим выражение, задающее поверхность модели:

$$\begin{aligned}
 y &= \frac{-0.2w_1(x-5)(-x+3)}{5} + \frac{0.2w_1x(4x-10)}{3} + \\
 &+ \frac{-w_2(x-12)(4x-10)}{21} + \frac{w_2(x-5)(-x+24)}{21} = \\
 &= \frac{w_1(7x^2 - 34x + 45)}{15} + \frac{w_2(-5x^2 + 87x - 240)}{21}.
 \end{aligned}$$

Как видно из этого выражения и рис. 2.7.2. поверхность модели не является линейной ни в одной из областей  $w_1$ , задаваемых заключениями правил. И даже в точках, полностью (со степенью, равной 1) принадлежащих множествам  $A_1$ ,  $A_2$  или  $A_3$ , касательные к поверхности модели не соответствуют задаваемым правилами линейным функциям (рис. 2.7.2.).

В частности, касательная к поверхности модели в точке  $x = 0$ , полностью принадлежащей множеству  $A_1$ , задается уравнением

$$y = \frac{14x - 34}{15}$$

хотя, в соответствии с заключением правила  $R_1$ , принадлежащего базе, следовало бы ожидать, что данное уравнение будет иметь вид  $y = -x + 3$ .

Такая же ситуация имеет место в точках  $x = 5$  и  $x = 12$ , принадлежащих со степенью, равной 1, соответственно множествам  $A_2$  и  $A_3$ . Поэтому применять TS-модели следует в тех случаях, когда функции принадлежности имеют трапециевидную или подобную ей форму (рис. 2.7.2).

При использовании трапециевидных функций с нелинейными ребрами (например, гауссовых функций), вместо поверхности, задаваемой непосредственно заключениями соответствующих правил  $R_i$  (даже в тех областях, в которых степень принадлежности элементов множеству  $A_i$  равна 1), будет получена поверхность, в той или иной степени измененная под влиянием функций /\* из других правил.

гауссовы функции принадлежности имеют бесконечный носитель, не удовлетворяют условию разбиения единицы и расширяют области влияния отдельных заключений. Для систем с двумя (или более) входами использование трапецевидных функций принадлежности приводит к получению прямоугольных (или гиперпрямоугольных) сегментов, в которых степени принадлежности пересечения множеств равны 1 (рис. 2.7.3.).

Взаимосвязи между моделями Мамдани и TS-моделями и возможности отображать и преобразовывать один тип модели в другой.

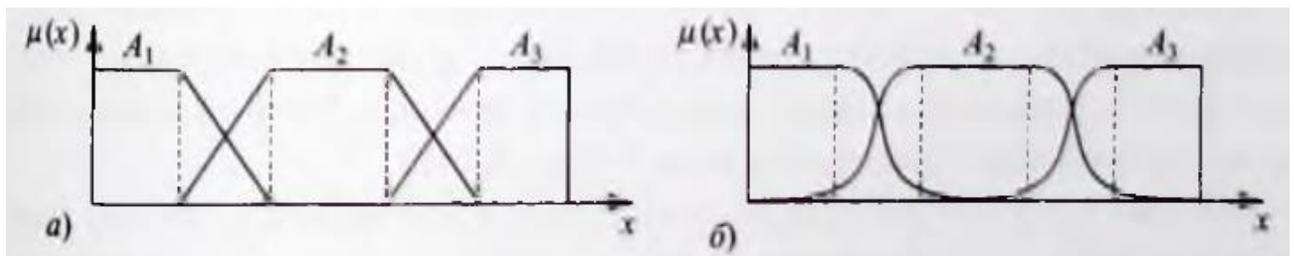


Рис. 2.7.3.. Типы функций принадлежности, рекомендуемые для использования в моделях Такаги—Сугено: трапецевидные (а) и трапецевидные (б) функции

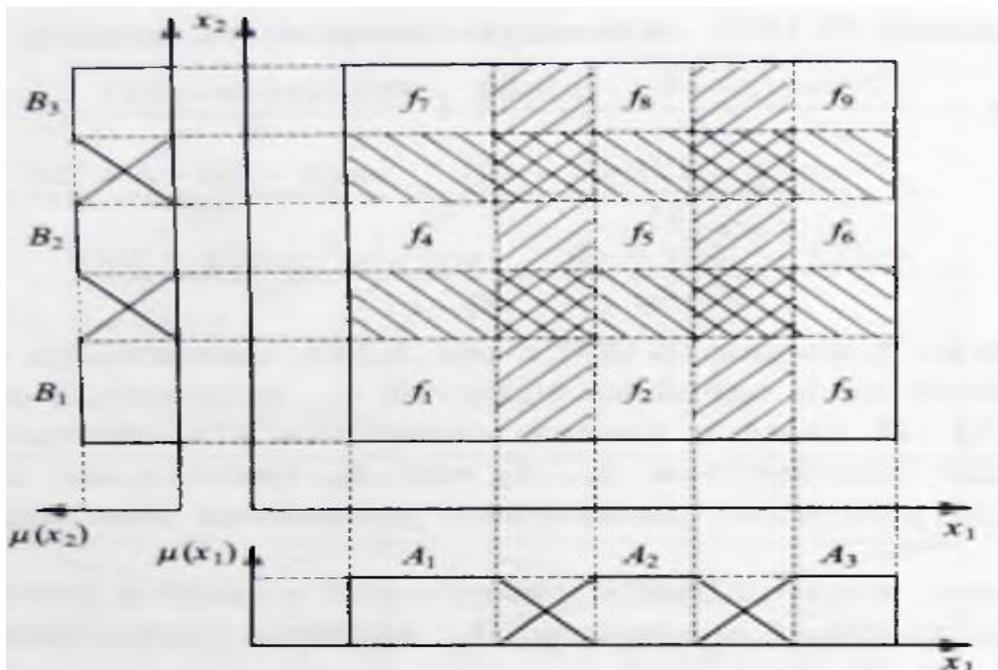


Рис. 2.7.4. Рекомендуемая форма функций принадлежности для модели Такаги—Сугено с двумя входными параметрами, позволяющая получить

прямоугольные сегменты, поверхности которых в точности соответствуют функциям, задаваемых заключениями правил  $R_k$ . Заштрихованные участки соответствуют областям перехода между отдельными заключениями  $f_k$

## 2.8. Области применения нечеткой логики

Когда только появилась теория нечеткой логики, число практических применений для нечеткой логики начало быстро расти. В настоящее время нечеткие технологии становятся все более актуальными среди представителей самых различных профессий. Существует несколько причин, на основании которых отдают предпочтение применению систем именно с нечеткой логикой:

- эта логика концептуально легче для понимания;
- нечеткая логика - гибкая система и устойчива к неточным входным данным;
- может моделировать нелинейные функции произвольной сложности;
- в данной логике учитывается опыт специалистов-экспертов;
- нечеткая логика основана на естественном языке человеческого общения.

Системы, основанные на нечеткой логике, разработаны и успешно внедрены в таких областях, как управление технологическими процессами, управление транспортом, управление бытовой техникой, медицинская и техническая диагностика, финансовый менеджмент, финансовый анализ, биржевое прогнозирование, распознавание образов, исследование рисков и критических операций, прогнозирование землетрясений, составление автобусных расписаний, климатический контроль в зданиях.

Многие современные задачи управления просто не могут быть решены классическими методами из-за очень большой сложности математических моделей, их описывающих.

Отличительные особенности fuzzy-систем по сравнению с прочими:

- возможность оперировать входными данными, заданными нечетко: например, непрерывно изменяющиеся во времени значения (динамические задачи), значения, которые невозможно задать однозначно (результаты статистических опросов, рекламные компании и т.д.);
- возможность нечеткой формализации критериев оценки и сравнения: оперирование критериями "большинство", "возможно", "предпочтительно" и т.д.;
- возможность проведения качественных оценок как входных данных, так и выводимых результатов: оперируете не только собственно значениями данных, но их степенью достоверности (не путать с вероятностью!) и ее распределением;
- возможность проведения быстрого моделирования сложных динамических систем и их сравнительный анализ с заданной степенью точности: оперируя принципами поведения системы, описанными fuzzy-методами, во-первых, не тратите много времени на выяснение точных значений переменных и составление уравнений, которые их описывают, во-вторых, можете оценить разные варианты выходных значений.

Использование аппарата нечеткой логики рекомендуется:

- для очень сложных процессов, когда не существует простой математической модели;
- для нелинейных процессов высоких порядков;
- если должна производиться обработка (лингвистически сформулированных) экспертных знаний.

Использование аппарата нечеткой логики не рекомендуется, если:

- приемлемый результат может быть получен с помощью общей теории управления;
- уже существует формализованная и адекватная математическая модель;
- проблема не разрешима.

## Глава 3. Нейронные сети

### 3.1. Введение в нейронные сети

Начало эпохи нейронных сетей считают сороковые годы 20 столетия, когда В. Макколлокх и В. Питтс предложили концепцию информационных вычислений, основанную на двоичных решающих элементах, названных нейронами. Каждый из этих элементов принимает значений 0 или 1 в зависимости от состояния покоя или активности. Некоторое состояние определяется также влиянием соседних нейронов. Новое состояние нейрона определяется суммой взвешенных воздействием других нейронов и выражается линейной комбинацией их выходных значений. В модели принято, что нейрон становится активным при превышении определенного порога. Веса могут быть положительными или отрицательными, что соответствует возбуждению или торможению нейрона. В. Макколлокх и В. Питтс показали, что такое функционирование сети нейронов позволяет выполнять вычисления, также как программируемые цифровые компьютеры. Особенность сети в том, что операции могут выполняться параллельно.

Реальные нейроны имеют отличие от нейронов, предложенных Макколлокх и В. Питтс:

- не являются пороговыми устройствами, а имеют нелинейное соотношение между входом и выходом,
- вырабатывают последовательность (фазовые соотношения), а не одиночный импульс,
- Не все нейроны имеют одинаковую задержку импульсов,
- Количество вещества, поступающего в трансмиссионную щель, меняется случайным образом.

На первом этапе развития нейронных сетей канадский ученый Д. Хебб сформулировал постулат о том, что если два нейрона являются взаимноактивными, то их взаимодействие усиливается.

Второй этап развития (50-60 годы) называют золотым веком развития нейронных сетей. Ф. Розенблат рассматривал специальный тип нейронных сетей, названный ими персептрон, как элемент биологической сенсорной информации. Это простейший элемент, состоящий из двух слоев: входного и выходного (входной слой за собой не считается). Нейроны выходного слоя принимают сигналы от входного, а не наоборот. Ф. Розенблат разработал итеративный алгоритм определения весов, с помощью которого входной образ преобразуется в требуемый выходной и доказал сходимость этого алгоритма.

Третий этап развития начался после 60-тых годов и получил название застойного, что случилось после работы М. Минского и С. Паперта (1969 г.), доказавшие, что существует ряд задач (например “исключающее ИЛИ”), которые не могут быть решены однослойным персептроном. После этой работы развитие нейронных сетей замедлилось на 10 лет. Ф. Розенблат разработал многослойные сети, но отсутствие алгоритма их обучения не позволило развиваться дальше.

Только некоторые ученые и среди них С. Гроссберг и Т. Кохонен продолжали исследования, опубликовав около 150 работ.

Четвертый этап соответствует середине 80-тых годов. Появился алгоритм определения весов в многослойных сетях со скрытыми слоями (Back propagation), предложенный в 1974 г. П. Вербозом, но развитый только в 1985 г. Этот обучающий алгоритм основан на простом эффективном принципе: веса  $w_{ij}$  изменяются итеративно таким образом, что мало отличаются от исходных. Это достигается использованием градиентного метода, обуславливающее требуемое изменение весов. Итерации применяются многократно, пока не будет достигнуто заданное условие ошибки получаемого выхода от желаемого. Появление этого алгоритма оказало существенное влияние на дальнейшее развитие нейросетевых технологий.

Проблема стабильности-пластичности является одной из самых сложных и трудно решаемых задач при построении искусственных систем, моделирующих восприятие. Некоторые нейронные системы не приспособлены запоминанию и отделению новых образов от старых. Поэтому разработаны ART и RBF сетей, положившие начало новым парадигмам, близким к функционированию человеческого мозга.

### Классификация нейронных сетей

Нейронные сети различают по структуре сети (связей между нейронами), особенностями модели нейрона, особенностями способа обучения сети.

По структуре нейронные сети можно разделить (рис. 10.2.1) на неполносвязные (или слоистые) и полносвязные, со случайными и регулярными связями, с симметричными и несимметричными связями.

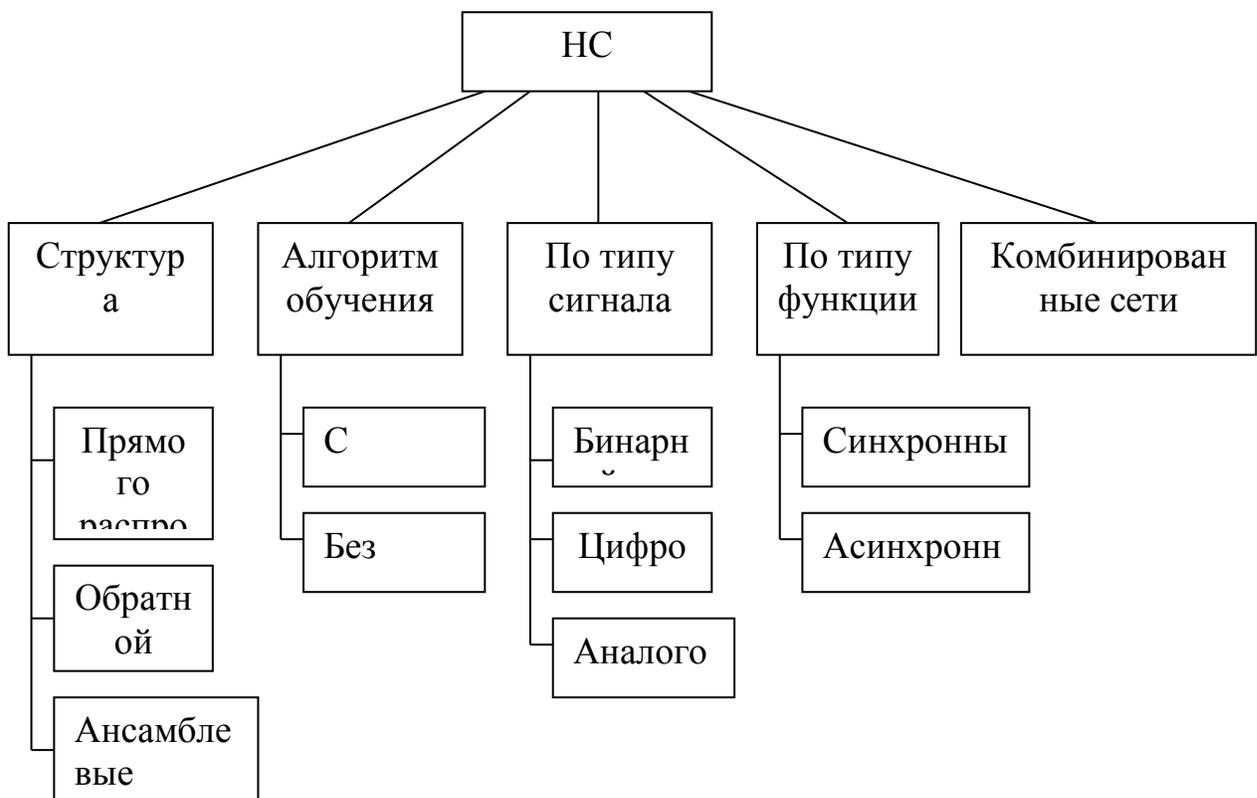


Рис 3.1.1 – Классификация нейросетевых парадигм

Неполносвязные нейронные сети (описываемые неполносвязным ориентированным графом и обычно называемые перцептронами) подразделяются на однослойные (простейшие перцептроны) и многослойные, с прямыми, перекрестными и обратными связями. В нейронных сетях с прямыми связями нейроны  $j$ -го слоя по входам могут соединяться только с нейронами  $i$ -ых слоев, где  $j > i$ , т.е. с нейронами нижележащих слоев. В нейронных сетях с перекрестными связями допускаются связи внутри одного слоя, т.е. вышеприведенное неравенство заменяется на  $j \geq i$ . В нейронных сетях с обратными связями используются и связи  $j$ -го слоя по входам с  $i$ -ым при  $j < i$ . Кроме того, по виду связей различают перцептроны с регулярными и случайными связями.

По используемым на входах и выходах сигналам нейронные сети можно разделить на аналоговые и бинарные.

По моделированию времени нейронные сети подразделяются на сети с непрерывным и дискретным временем. Для программной реализации применяется, как правило, дискретное время.

По способу подачи информации на входы нейронной сети различают:

- подачу сигналов на синапсы входных нейронов;
- подачу сигналов на выходы входных нейронов;
- подачу сигналов в виде весов синапсов входных нейронов;
- аддитивную подачу на синапсы входных нейронов.

По способу съема информации с выходов нейронной сети различают:

- съем с выходов выходных нейронов;
- съем с синапсов выходных нейронов;
- съем в виде значений весов синапсов выходных нейронов;
- аддитивный съем с синапсов выходных нейронов.

По организации обучения разделяют обучение нейронных сетей с учителем и без учителя. При обучении с учителем предполагается, что есть внешняя среда, которая предоставляет обучающие примеры (значения входов и соответствующие им значения выходов) на этапе обучения или оценивает правильность функционирования нейронной сети и в соответствии со своими критериями меняет состояние нейронной сети или поощряет (наказывает) нейронную сеть, запуская тем самым механизм изменения ее состояния. Под состоянием нейронной сети, которое может изменяться, обычно понимается:

- веса синапсов нейронов (карта весов – map) (коннекционистский подход);
- веса синапсов и пороги нейронов (обычно в этом случае порог является более легко изменяемым параметром, чем веса синапсов);
- установление новых связей между нейронами (свойство биологических нейронов устанавливать новые связи и ликвидировать старые называется пластичностью).

По способу обучения разделяют обучение по входам и по выходам. При обучении по входам обучающий пример представляет собой только вектор входных сигналов, а при обучении по выходам в него входит и вектор выходных сигналов, соответствующий входному вектору.

По способу предъявления примеров различают предъявление одиночных примеров и «страницы» примеров. В первом случае изменение состояния нейронной сети (обучение) происходит после предъявления каждого примера. Во втором – после предъявления «страницы» (множества) примеров на основе анализа сразу их всех.

### **Классификация по топологии**

Полносвязные сети представляют сети, в которых имеются все возможные связи: многослойные или слоистые. Сеть разбивается на слои, где

каждый слой содержит совокупность нейронов с едиными входными сигналами. Слои нумеруются слева направо, начиная с 0.

В свою очередь многослойные сети делятся на:

- **Монотонные.**

Каждый слой разбит на два блока (кроме последнего): возбуждающий блок и тормозящий блок. Если от блока 1 к нейронам блока 2 ведут только возбуждающие/тормозящие связи, то любой выходной сигнал есть монотонная неубывающая/невозрастающая функция любого выходного сигнала блока.

- **Сети со связями.**

Сети без обратных связей: обычная многослойная сеть с одним ограничением, что нейроны  $q$ -ого слоя могут передавать сигнал нейронам  $(q+1)$ -ого слоя и никакому другому слою.

Сети с обратными связями делятся на:

- **слоисто циклические:** многослойная сеть, у которой выход передает свои сигналы 1-му слою (не 0-му, а 1-му). Состоит из слоев, каждый из которых представляет собой полносвязную сеть;

- **полносвязные-слоистые:** по своей структуре в них не разделяют фазы обмена внутри слоя и передачи следующему, на каждом такте нейроны всех слоев применяют сигналы от нейронов как своего слоя, так и последующих.

- **слабосвязанные сети:** в слабосвязанных сетях нейроны располагаются в узлах прямоугольной формы или гексагональной решетки. Каждый нейрон связан с 4 (окрестность фон Неймана), 6 (окрестность Голея) или 8 (окрестность Мура) своими ближайшими соседями.

### **Коннекционистская парадигма**

Основная идея состоит в следующем: для того, чтобы реализовать некоторые возможности мозга, необходимо воссоздать его архитектурные особенности. Коннекционистская машина или нейронная сеть является высокосвязной сетью

простых процессоров (искусственных нейронов), каждый из которых имеет много входов и много выходов.

В биологических нейронах способность к обработке выражается в электрохимических характеристиках межнейронных соединений (синапсов). В коннекционистских системах она моделируется заданием силы связи или веса каждому входу. Биологические нейроны взаимодействуют путем передачи электрических импульсов, идентичных друг другу, а информация заключена в пространственно-временных связях между ними. Нейроны непрерывно суммируют эффект всех приходящих импульсов. Если результат суммирования возбуждающий, то генерируется выходной импульс, если тормозящий, то выходной импульс не генерируется.

В искусственных нейронных сетях каждый нейрон непрерывно обновляет свое состояние порождением значения активации, которая является функцией входных сигналов и внутренних параметров нейрона. Активация используется для генерации выхода через некоторую функцию.

Типичные характеристики искусственных нейронных сетей (ИНС):

1. Параметры нейрона (веса) настраиваются посредством подачи на входы обучающих векторов и изменением весов таким образом, чтобы нейрон выдавал требуемые выходные сигналы. Таким образом, нейрон является адаптивной, а не заранее запрограммированной системой.
2. Работу ИНС целесообразно представлять как эволюцию динамической системы и описывать системой дифференциальных уравнений.
3. Нейронные сети устойчивы к шумам в сигналах и отказам компонентов (нейронов и синапсов). Отказ компонента не влечет отказа всей ИНС в целом, а лишь несколько ухудшает ее характеристики.

4. Характерной особенностью работы нейронных сетей является то, что они способны находить статистические закономерности или особенности в обучающей выборке. Это позволяет нейронной сети отнести новый входной объект к одному из уже усвоенных сетью образов либо к новому классу.
5. Типичными приложениями нейронных сетей являются классификация образов и ассоциативная память, восстанавливающая полный образ по частичным данным.
6. Не существует простого соответствия между нейронами и семантическими объектами. Скорее, представление "концепции" или "идеи" в сети осуществляется через вектор активностей нейронов, распределенный по сети, так что каждый нейрон участвует одновременно в представлении многих объектов ("идей").

### **Свойства искусственных нейронных сетей**

Искусственные нейронные сети состоят из элементов, функциональные возможности которых идентичны большинству элементарных функций биологического нейрона. Эти элементы организуются по способу, соответствующему анатомии мозга. Несмотря на такое поверхностное сходство, искусственные нейронные сети демонстрируют удивительные свойства, присущие мозгу. Они обучаются на основе опыта, обобщают предыдущие прецеденты на новые случаи и извлекают существенные свойства из поступающей информации, содержащей излишние данные.

Несмотря на функциональное сходство, искусственные нейронные сети вряд ли смогут моделировать функции человеческого мозга. Самые сложные нейронные сети, находятся ниже уровня дождевого червя. С другой стороны неправильно игнорировать удивительное сходство в функционировании некоторых нейронных сетей с человеческим мозгом.

Искусственные нейронные сети могут менять свое поведение в зависимости от внешней среды. Этот фактор в большей степени, чем любой другой, ответствен

за тот интерес, который они вызывают. После предъявления входных сигналов, они самонастраиваются, чтобы обеспечивать требуемую реакцию. Разработано много обучающих алгоритмов, каждый со своими сильными и слабыми сторонами.

Отклик сети после обучения может быть до некоторой степени нечувствителен к небольшим изменениям входных сигналов. Эта внутренне присущая способность видеть образ сквозь шум и искажения жизненно важна для распознавания образов в реальном мире. Она позволяет преодолеть требование строгой точности, предъявляемое обычным компьютером, к системе нечеткого мира, в котором живем. Искусственная нейронная сеть обобщает автоматически благодаря своей структуре, а не использованием «человеческого интеллекта» в форме специально написанных компьютерных программ.

### **Абстрагирование**

Некоторые из искусственных нейронных сетей обладают способностью извлекать сущность из входных сигналов. Например, сеть может быть обучена на последовательность искаженных версий буквы «А». После соответствующего обучения предъявление такого искаженного примера приведет к тому, что сеть породит букву совершенной формы. В некотором смысле она научится порождать то, что никогда не видела.

### **3.2. Биологическая модель нейрона**

Структура искусственных нейронных сетей создана по результатам изучения человеческого мозга, хотя сходство между ними и незначительно. Искусственные нейронные сети имеют такие аналогичные мозгу свойства, как способность обучаться на опыте, основанном на знаниях, делать абстрактные умозаключения и совершать ошибки, что является более характерным для человеческой мысли, чем для созданных человеком компьютеров.

Большая часть мозга остается трудной для понимания. Основные исследования направлены на идентификации функций мозга. Нейрон оказался сложнее, чем представлялось ранее, и до сих пор нет полного понимания процесса его функционирования. Но все-таки, мозг может быть использован в качестве модели для развития искусственных нейронных сетей, что и привело к структурам нейронных сетей, решающих человеческие задачи.

По оценкам человеческий мозг содержит свыше тысячи миллиардов вычислительных элементов, называемых нейронами, связанные сотнями триллионов нервных нитей, называемых синапсами. Эта сеть нейронов отвечает за все явления, которые называются мыслями, эмоциями, познанием, а также и за совершение множества сенсомоторных и автономных функций.

Мозг является основным потребителем энергии тела. Включая в себя лишь 2% массы тела, в состоянии покоя он потребляет приблизительно 20% кислорода тела. Даже во время сна, расходование энергии продолжается. Потребляя примерно 20 Вт, мозг с энергетической точки зрения невероятно эффективен. Компьютеры с одной минимальной долей вычислительных возможностей мозга потребляют сотни и тысячи ватт, и требуют сложных систем для охлаждения.

Нейрон является основным строительным блоком нервной системы и содержит клетки, подобные другим клеткам тела; однако определенные отличия выполняют вычислительные функции и функции связи внутри мозга.

Нейрон состоит из трех основных частей: тела клетки, дендритов и аксонов (рис. 3.2.1.).

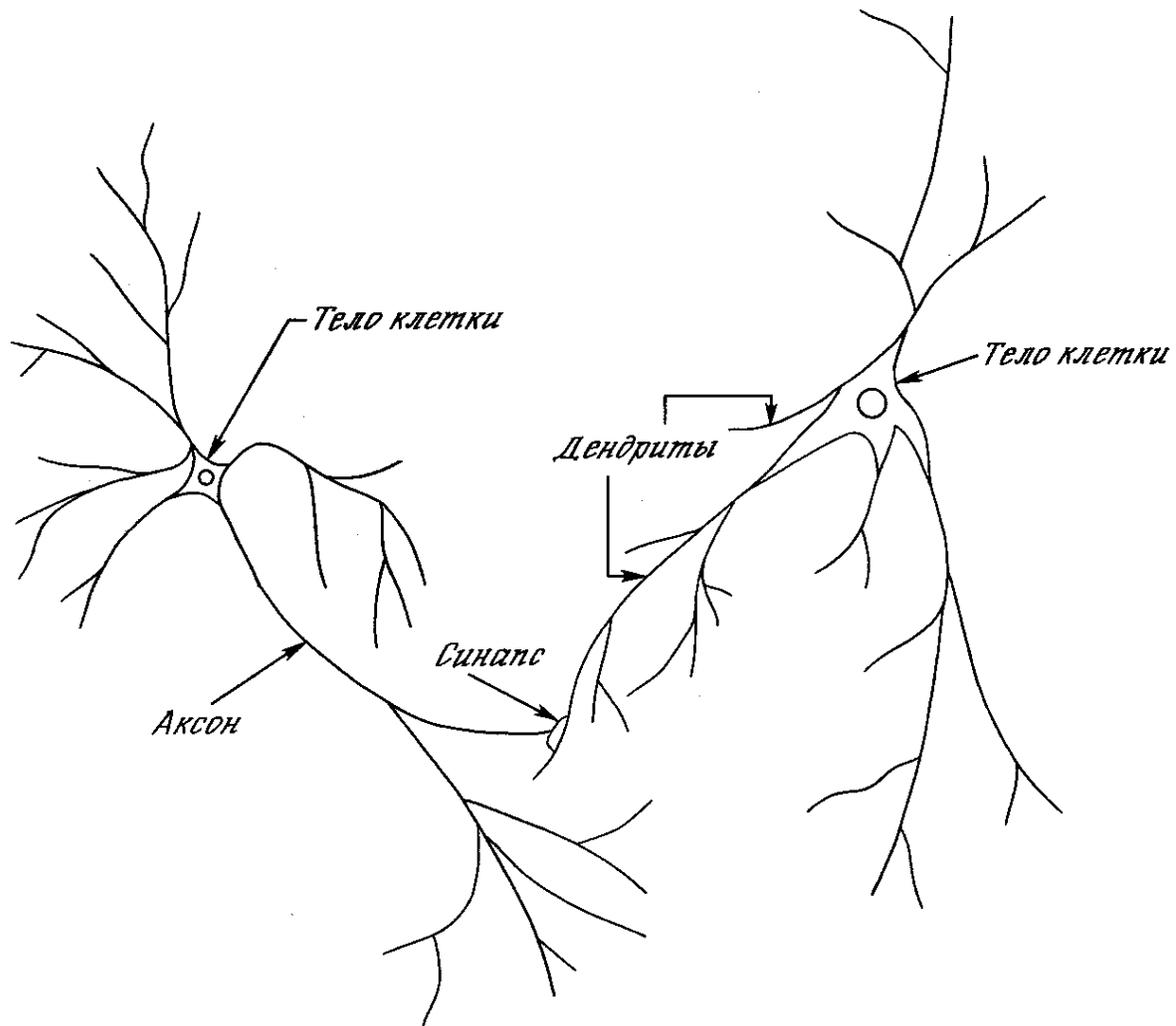


Рис. 3.2.1. Компоненты нейрона

Функционально дендриты получают сигналы от других клеток через контакты, называемые синапсами. Затем сигналы проходят в тело клетки, где они суммируются с другими такими же сигналами. Если суммарный сигнал в течение короткого промежутка времени является достаточно большим, клетка возбуждается, вырабатывая в аксоне импульс, который передается на следующие клетки. Эта схема функционирования, хотя и упрощенная, объясняет большинство известных процессов мозга.

Как установлено в мозге живого существа имеется 2 типа связей:

- передача химических сигналов через синапсы,

- передача электрических сигналов внутри нейрона.

Сотни типов нейронов определены, каждый со своей характерной формой тела клетки, имеющей обычно от 5 до 100 мкм в диаметре. Входные сигналы от других нейронов попадают в клетку через дендриты, проходящие к телу клетки. На дендритах располагаются синаптические соединения, получающие сигналы от других аксонов. Кроме этого, существует огромное количество синаптических связей от аксона к аксону, от аксона к телу клетки и от дендрита к дендриту; пока их функции не очень ясны.

В отличие от электрических цепей, синаптические контакты обычно не являются физическими или электрическими соединениями, а имеется узкое пространство, называемое синаптической щелью, отделяющее дендрит от передающего аксона. Специальные химические вещества, выбрасываемые аксоном в синаптическую щель, и диффундируют к дендриту. Эти химические вещества, называемые нейротрансмиттерами, улавливаются специальными рецепторами на дендрите и внедряются в тело клетки (Рис. 3.2.2).

Определено более 30 видов нейротрансмиттеров. Некоторые из них являются возбуждающими и стремятся вызывать возбуждение клетки и выработать выходной импульс. Другие являются тормозящими и стремятся подавить такой импульс. Тело клетки суммирует сигналы, полученные от дендритов, и если их результирующий сигнал выше порогового значения, вырабатывается импульс, проходящий по аксону к другим нейронам.

Аксон может быть как коротким (0,1 мм), так и превышать длину 1 м, распространяясь в другую часть тела человека. На конце аксон имеет множество ветвей, каждая из которых завершается синапсом, откуда сигнал передается в другие нейроны через дендриты, а в некоторых случаях прямо в тело клетки. Таким образом, один нейрон может генерировать импульс, который возбуждает или затормаживает множество других нейронов, каждый из которых, в свою очередь, через свои дендриты может воздействовать на

другие нейроны. Высокая степень связанности и простота нейрона, обеспечивает вычислительную мощьность.

Синаптическая связь, завершающая ветвь аксона, представляет собой маленькие утолщения, содержащие сферические структуры, называемые синаптическими пузырьками, каждый из которых содержит большое число нейротрансмиттерных молекул. Когда нервный импульс приходит в аксон, некоторые из этих пузырьков высвобождают свое содержимое в синаптическую щель, тем самым инициализируя процесс взаимодействия нейронов (рис. 3.2.2.).

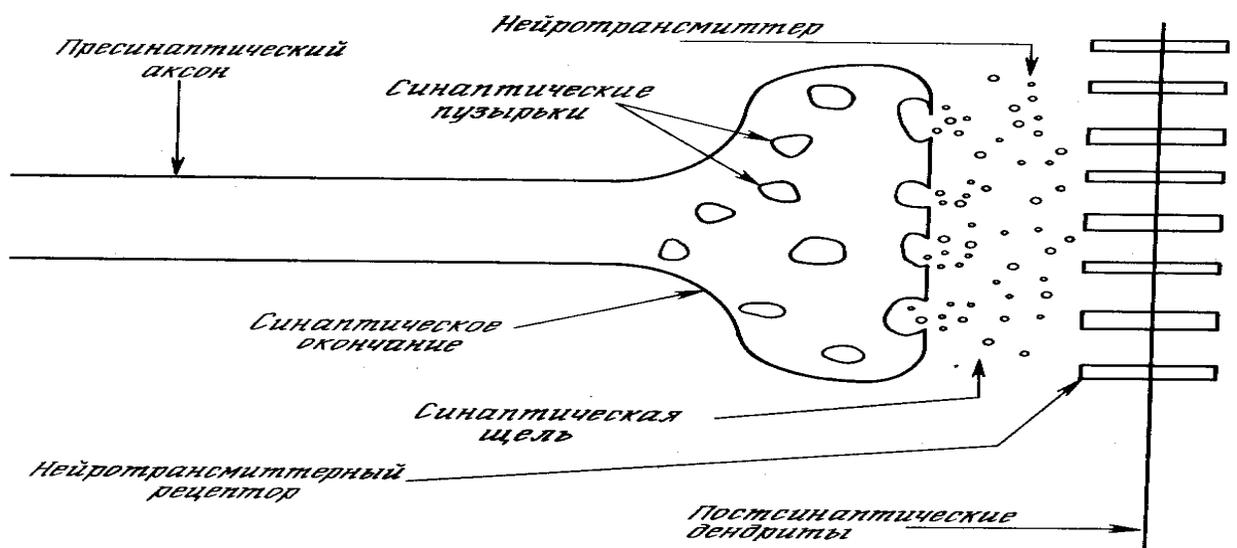


Рис. 3.2.2. Схема синаптической цепи

Кроме распространения такого бинарного сигнала, обеспечиваемого возбуждением первого импульса, в нейронах при слабой стимуляции могут также распространяться электрохимические сигналы с последовательной реакцией. Локальные по своей природе, эти сигналы быстро затухают с удалением от места возбуждения, если не будут усилены. Природа использует это свойство первых клеток путем создания вокруг аксонов изолирующей оболочки из клеток, которая прерывается (приблизительно через каждый миллиметр) вдоль аксона узкими разрывами, называемыми узлами, или перехватами. Поэтому импульсы, приходящие в аксон, передаются

скачкообразно от узла к узлу, и аксон не расходует энергию для поддержания своего химического градиента по всей своей длине.

Мембрана клетки имеет около 5 нм толщины и состоит из двух слоев молекул. В мембрану встроены насосы, каналы, рецепторы, ферменты и структурные белки.

Уровень возбуждения нейрона определяется эффектом возбуждения и торможения большого числа входов, суммируемых телом клетки в течение короткого временного интервала. Получение возбуждающей нейротрансмиттерной молекулы будет увеличивать уровень возбуждения нейрона; или их меньшее количество уменьшает уровень возбуждения. Таким образом, нейронный сигнал является импульсным или частотно-модулируемым (ЧМ). Этот метод модуляции, широко используемый в технике (например, ЧМ-радио), имеет значительные преимущества при наличии помех по сравнению с другими способами.

Исследования показали сложность биохимических процессов в мозге. Например, предполагается наличие свыше 30 веществ, являющихся нейротрансмиттерами, и большое количество рецепторов с различными ответными реакциями. Более того, действие определенных нейротрансмиттерных молекул зависит от типа рецептора в постсинаптической мембране, некоторые нейротрансмиттеры могут быть возбуждающими для одного синапса и тормозящими для другого.

Существует подобие между мозгом и цифровым компьютером: оба оперируют электронными сигналами, оба состоят из большого количества простых элементов, оба выполняют функции, являющиеся, грубо говоря, вычислительными. Тем не менее, существуют и фундаментальные отличия. По сравнению с микросекундными и даже наносекундными интервалами вычислений современных компьютеров нервные импульсы являются слишком медленными. Хотя каждый нейрон требует наличия миллисекундного интервала

между передаваемыми сигналами, высокая скорость вычислений мозга обеспечивается огромным числом параллельных вычислительных блоков, причем количество их намного превышает доступное современным ЭВМ. Диапазон ошибок представляет другое фундаментальное отличие: ЭВМ присуща свобода от ошибок, если входные сигналы безупречно точны и ее аппаратное и программное обеспечение не повреждены. Мозг же часто производит лучшее угадывание и приближение при частично незавершенных и неточных входных сигналах. Часто он ошибается, но величина ошибки должна гарантировать наше выживание в течение миллионов лет.

Первые цифровые вычислители часто рассматривались как «электронный мозг». С точки зрения наших текущих знаний о сложности мозга, такое заявление, принципиально, не соответствует истине. Эти две системы, конечно, различаются в каждой своей части. Они оптимизированы для решения различных типов проблем, имеют существенные различия в структуре и их работа оценивается различными критериями.

Искусственные нейронные сети когда-нибудь будут дублировать функции человеческого мозга, однако прежде чем добиться этого, необходимо понять организацию и функции мозга. Современные нейросети базируются на очень упрощенной модели, игнорирующей большинство тех знаний, которые мы имеем о детальном функционировании мозга. Развитие искусственных нейронных сетей требует теоретического объяснения, что приведет к улучшению математических методов, моделирующих такие системы. Для описания сложности головного мозга, необходимы совершенно новые математические методы.

### **3.3. Искусственный (математический) нейрон**

Искусственный нейрон (ИН) имитирует поведение природной нервной клетки мозга. Обобщенная схема  $i$ -ого нейрона представлена на рис. 3.3.1.

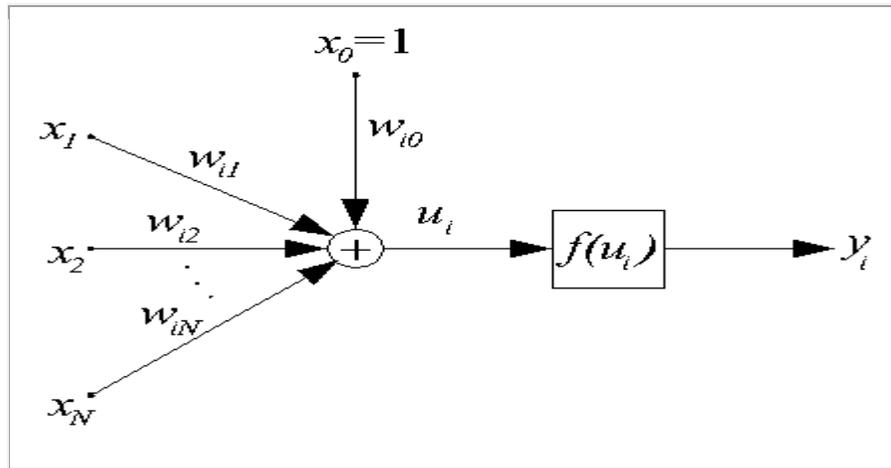


Рис. 3.3.1. Обобщенная структурная схема искусственного нейрона

Здесь  $\mathbf{X} = [x_1, x_2, \dots, x_N]^T$  — вектор входных сигналов нейрона,  $x_0 = 1$  — сигнал смещения. В общем случае  $x_i$  — действительные числа, возможно, размерные, но чаще нормированные. Во многих моделях ИНС (искусственной нейронной сети)  $x_i$  дискретны и могут принимать значения из множеств  $\{0, 1\}$  или  $\{-1, 1\}$ .

$\mathbf{W}_i = [w_{i1}, w_{i2}, \dots, w_{iN}]^T$  — вектор весов (синапсов) входных сигналов  $i$ -ого нейрона (в общем случае, действительные числа). Веса входных сигналов используются для вычисления взвешенной суммы  $s_i$  входных сигналов  $i$ -ого ИН по формуле

$$u_i = \sum_{j=0}^N w_{ij} \cdot x_j. \quad (3.1)$$

Вес  $w_{ij}$  для размерного входного сигнала  $x_j$  имеет обратную к нему размерность. Например, если  $x_j$  имеет размерность [кг], то размерность  $w_{ij}$  — [кг<sup>-1</sup>].

Взвешенная сумма входных сигналов  $u_i$  служит аргументом функции активации ИН  $f(u_i)$ , определяющей значение выходного сигнала  $y_i$ .

В простейших однослойных (без обратных связей и каскадов) сетях входные сигналы  $x_j$  нейрона — входные сигналы сети в целом, а выходные сигналы  $y_i$  нейрона — выходные сигналы сети в целом. В многослойных сетях роль входных сигналов  $x_j$  некоторых нейронов играют выходные сигналы  $y_i$  предыдущих слоев ИНС.

Входному вектору  $\mathbf{X}$  значение  $y_i$  определяет принадлежность входного вектора тому или иному  $i$ -му классу. Степень успеха зависит от "правильности" подбора/назначения весовых коэффициентов  $w_{ij}$  и функции активации  $f(u_i)$ . Однако, в практических ИНС обычно функции активации назначаются однократно и подбираются только весовые коэффициенты  $w_{ij}$ .

**Перцептрон** предложен в 1943 г. и называется также моделью МакКаллока-Питса. В этой модели искусственный нейрон считается бинарным элементом, его структурная схема представлена на рис. 1.

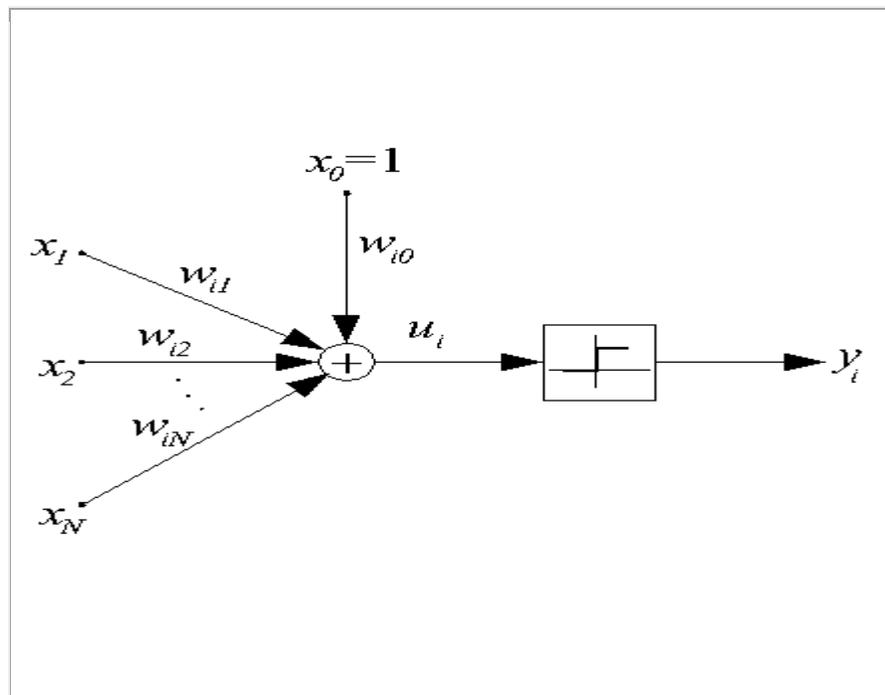


Рис. 3.3.2. Структурная схема перцептрона

Выходной сигнал нейрона может принимать только два значения  $\{0, 1\}$

$$y_i = f(u_i) = 1, \text{ если } u_i \geq 0;$$

$$y_i = f(u_i) = 0, \text{ если } u_i < 0.$$

### Двухвходовой нейрон

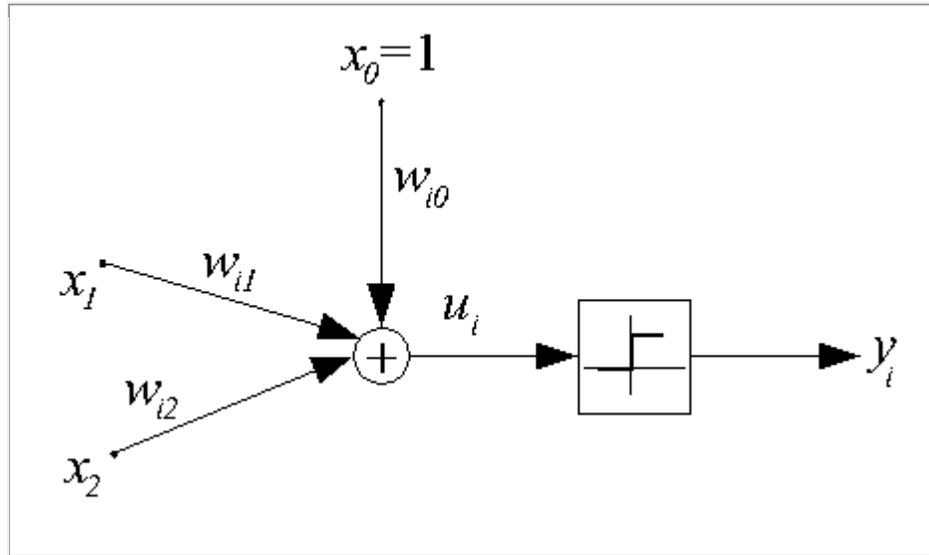


Рис. 3.3.3. Структурная схема двухвходового персептрона

Для нейрона выражение  $u_i = w_{i0} + w_{i1} \cdot x_1 + w_{i2} \cdot x_2$  определяет плоскость в двумерном пространстве, пересекающаяся с плоскостью  $\langle x_1, x_2 \rangle$  по линии, определяемой уравнением  $w_{i0} + w_{i1} \cdot x_1 + w_{i2} \cdot x_2 = 0$ , (рис. 3.3.4)

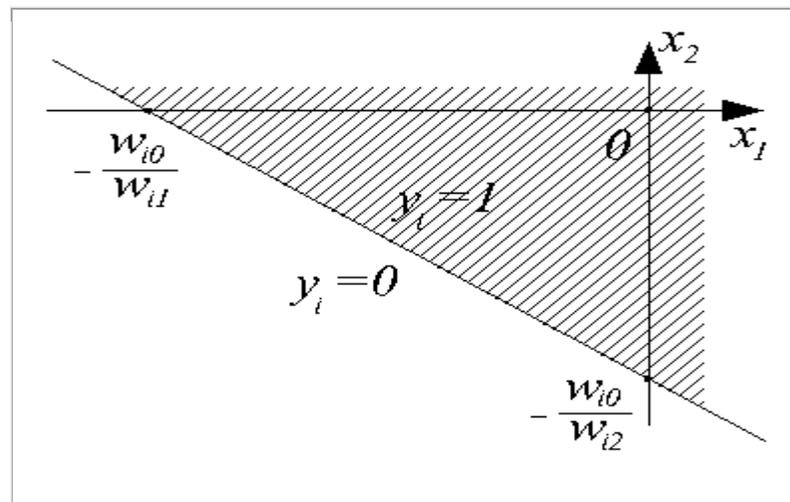


Рис.3.3.4 . Разделение пространства входных данных двухвходовым персептроном.

Пространство входных сигналов  $\langle x_1, x_2 \rangle$  делится на две области: в одной из них (штрих) значения  $u_i > 0$ , функция активации принимает значение 1; в другой —  $u_i < 0$ , и  $y_i = 0$ .

Персептрон является простейшим линейным классификатором. С его помощью можно обеспечить, реализацию логических функций "И" и "ИЛИ" над входами  $x_1$  и  $x_2$ , (рис. 4.)

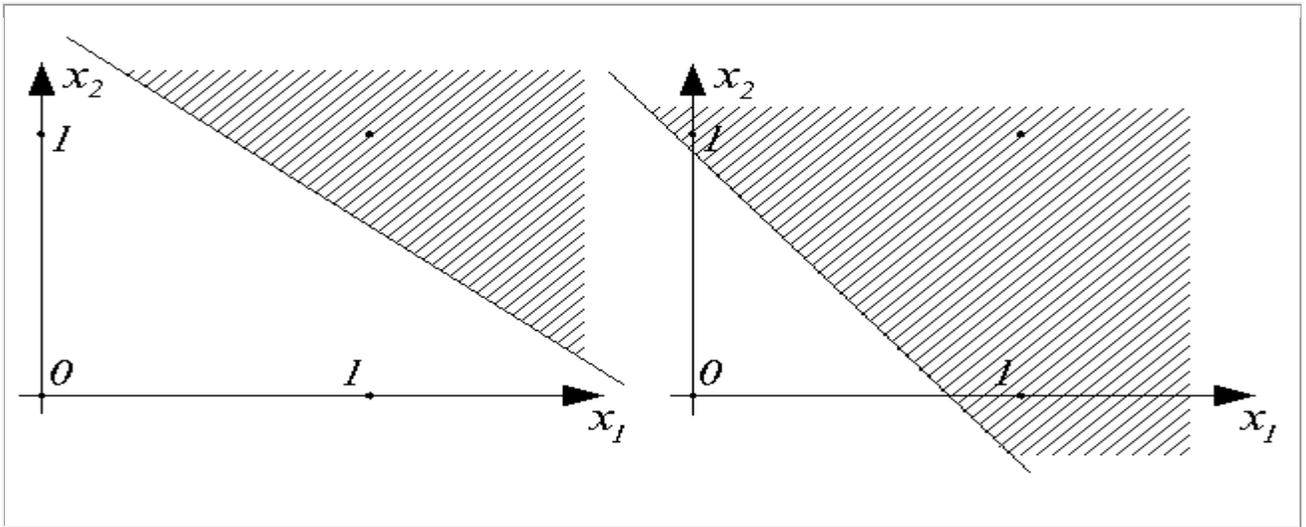


Рис. 3.3.5. Реализация логических функций И и ИЛИ двухвходовым персептроном

Логическую функцию "исключающее ИЛИ" двухвходовым персептроном реализовать невозможно (Рис. 3.3.6).

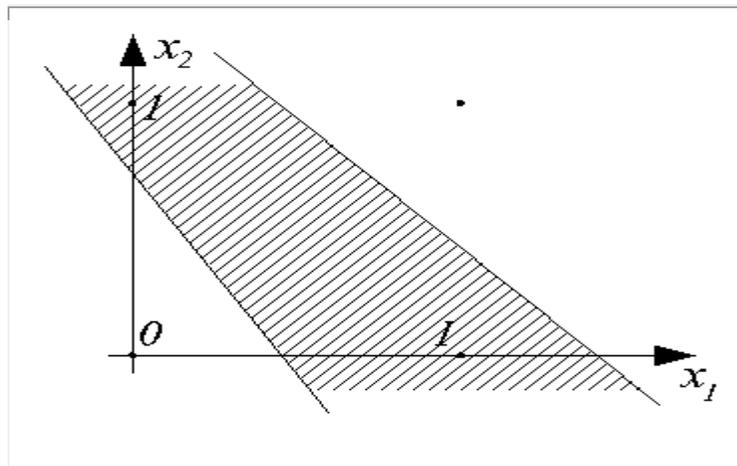


Рис. 3.3.6. Разделение пространства входных данных для реализации функции "исключающее ИЛИ"

### Сигмоидальный нейрон

Сигмоидальный нейрон устраняет основной недостаток персептрона — разрывность функции активации  $f(u_i)$ . Структурная схема нейрона данного вида представлена на рис. 3.3.7.

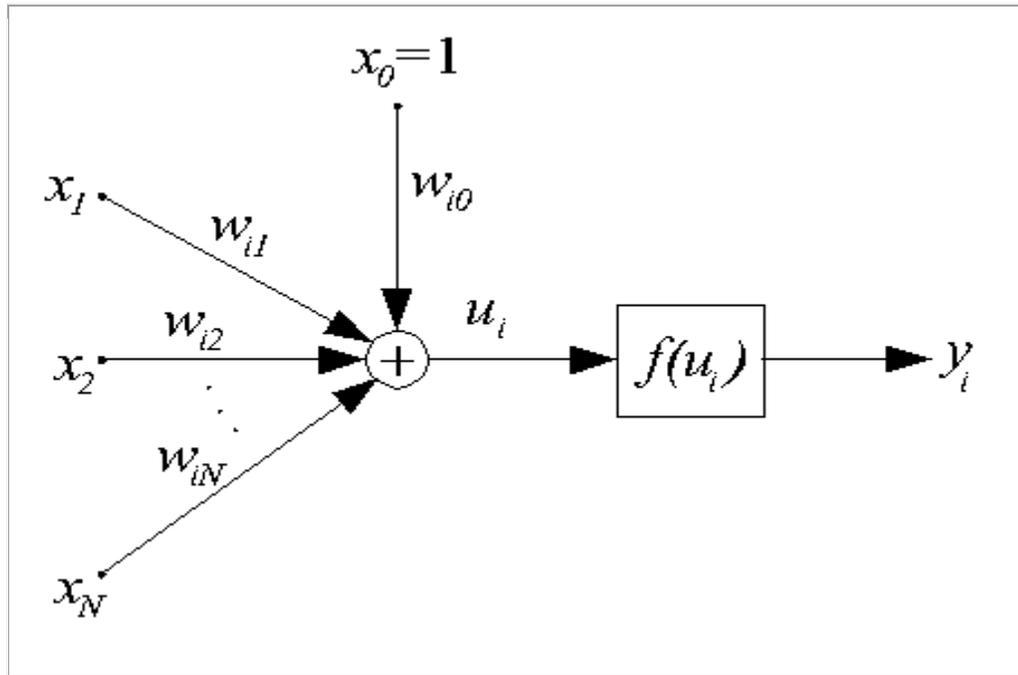


Рис. 3.3.7. Структурная схема сигмоидального нейрона

В качестве функции активации  $f(u_i)$  выступает сигмоидальная функция. На практике используются как униполярные, так и биполярные функции активации.

### Нейроны типа WTA

Нейроны типа WTA (Winner Takes All — победитель получает все) всегда используются группами, в которых конкурируют между собой.

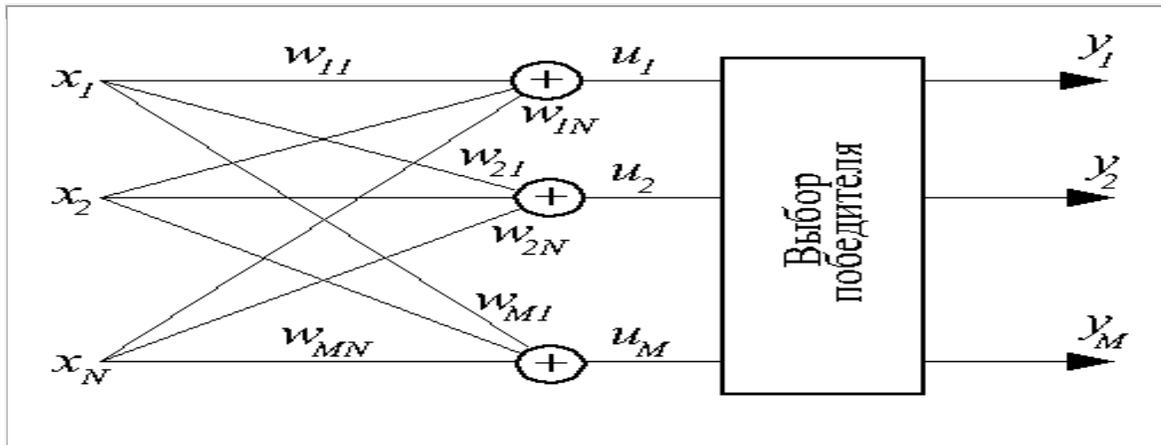


Рис. 3.3.8. Структурная схема слоя нейронов типа WTA

Каждый конкурирующий нейрон в группе получает одни и те же входные сигналы. Каждый нейрон рассчитывает выходной сигнал своего сумматора

$$u_i = \sum_{j=1}^N w_{ij} \cdot x_j$$

. обычным образом

По результатам сравнения всех  $u_i$ ,  $i = 1, 2, \dots, M$ , выбирается нейрон-победитель, обладающий наибольшим значением  $u_i$ . Выходной сигнал  $y_i$  нейрона-победителя получает значение 1, выходные сигналы всех остальных нейронов — 0.

Для обучения нейронов типа WTA не требуется учитель. Начальные значения весовых коэффициентов всех нейронов выбираются случайным образом с последующей нормализацией относительно 1.

При предъявлении каждого обучающего вектора  $\mathbf{X}^k$  определяется нейрон-победитель, что дает ему право уточнить свои весовые коэффициенты по упрощенному (в силу бинарности  $y_i$ )

$$w_{ij}(t+1) = w_{ij}(t) + \eta \cdot (x_j^k - w_{ij}(t)). \quad (3.2)$$

Все проигравшие нейроны оставляют свои весовые коэффициенты неизменными.

В каждом цикле обучения побеждает нейрон, чей текущий вектор входных весов  $\mathbf{W}_i$  наиболее близок входному вектору  $\mathbf{X}^k$ . При этом вектор  $\mathbf{W}_i$  корректируется в сторону вектора  $\mathbf{X}^k$ . Поэтому в ходе обучения каждая группа близких друг другу входных векторов (кластер) обслуживается отдельным нейроном.

Результат обучения слоя нейронов типа WTA на последовательности девяти двухкомпонентных входных векторов  $\mathbf{X}^1, \mathbf{X}^2, \dots, \mathbf{X}^9$  иллюстрирует рис. 3.3.9. Здесь были выделены три кластера входных векторов  $\{\mathbf{X}^1, \mathbf{X}^8\}$ ,  $\{\mathbf{X}^3, \mathbf{X}^4, \mathbf{X}^5\}$  и  $\{\mathbf{X}^2, \mathbf{X}^6, \mathbf{X}^7, \mathbf{X}^9\}$ . За их распознавание отвечают три нейрона с векторами входных весов  $\mathbf{W}_1$ ,  $\mathbf{W}_2$  и  $\mathbf{W}_3$  соответственно.

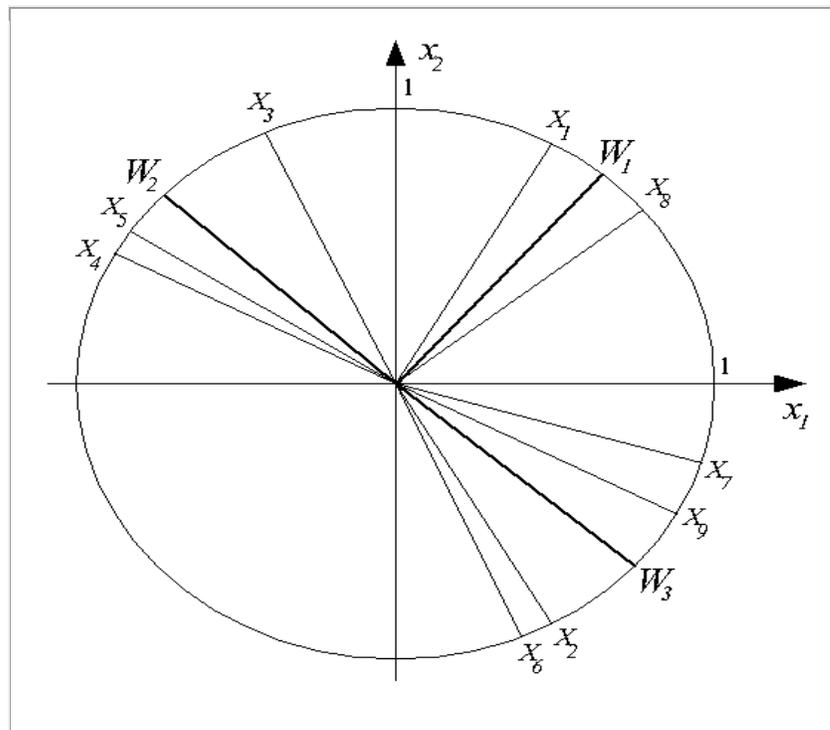


Рис. 3.3.9. Результат обучения слоя нейронов типа WTA

При использовании нейронов типа WTA возникает возможность появления нейронов, ни разу не победивших в конкурентной борьбе в ходе обучения и поэтому оставшихся в начальном состоянии. Для исключения "ложных"

срабатываний в режиме классификации “мертвые” нейроны после окончания обучения должны быть удалены.

Для уменьшения количества “мертвых нейронов” (и, следовательно, повышения точности распознавания) используется модифицированное обучение, основанное на учете числа побед нейронов и штрафования наиболее “зарвавшихся” среди них. Дисквалификация может быть реализована либо назначением порога числа побед, после которого слишком активный нейрон “засыпает” на заданное число циклов обучения, либо искусственным уменьшением величины  $u_i$  пропорционально

### Нейрон Хебба

Д. Хебб, в 1949 г. вычислил усиление связи двух взаимодействующих клеток при их одновременном возбуждении. Это позволило ему предложить правило уточнения входных весов нейрона в следующем виде:

$$w_{ij}(t+1) = w_{ij}(t) + \eta \cdot x_j^k \cdot y_i(t), \quad (3.3)$$

где  $\eta$  — коэффициент обучения, значение которого выбирается из интервала (0, 1). Правило Хебба применимо для нейронов с различными функциями активации. Обучение нейрона может производиться как с учителем, так и без него. В первом случае в правиле Хебба вместо фактического значения выходного сигнала  $y_i(t)$  используется ожидаемая реакция  $d_i^k$ .

Особенностью правила Хебба является возможность достижения весом  $w_{ij}$  произвольно большого значения за счет многократного суммирования приращения в циклах обучения. Одним из способов стабилизации процесса обучения по Хеббу служит уменьшение уточняемого веса  $w_{ij}(t)$  на величину, пропорциональную коэффициенту забывания  $\gamma$ . При этом правило Хебба

принимает

$$w_{ij}(t+1) = w_{ij}(t) \cdot (1 - \gamma) + \eta \cdot x_j^k \cdot y_i(t). \quad (3.4)$$

Значение коэффициента забывания  $\gamma$  выбирается из интервала  $(0, 1)$ , рекомендуется соблюдать условие  $\gamma < 0,1$ .

При обучении по правилу Хебба нейрона с линейной функцией активации стабилизация не достигается даже при использовании забывания. В 1991 г. Е.Ойя предложил модификацию правила Хебба, имеющую следующий вид:

$$w_{ij}(t+1) = w_{ij}(t) + \eta \cdot (x_j^k - y_i(t)) \cdot w_{ij}(t) \cdot y_i(t).$$

### Радиальный нейрон

Радиальные нейроны существенно отличаются от нейронов других типов. Они используются только группами, составляя первый слой в многослойных радиальных нейронных сетях. Структурная схема такого нейрона дана на рис. 3.3.10.

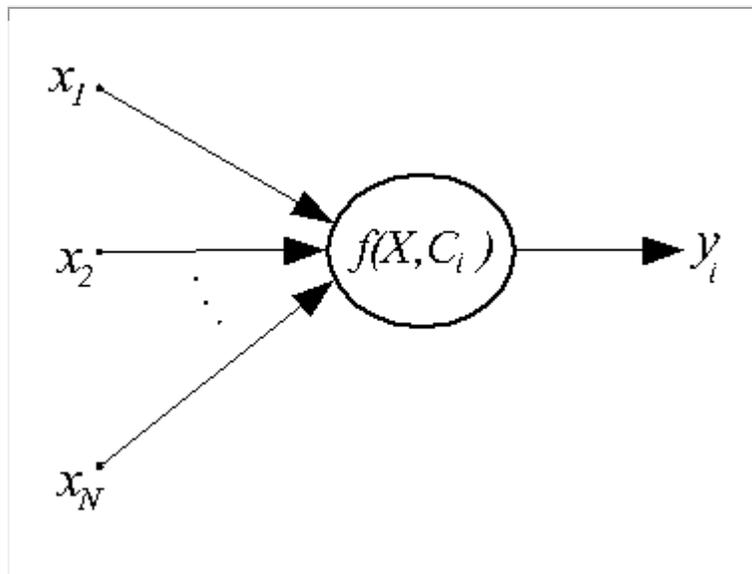


Рис. 3.3.10 . Структурная схема радиального нейрона

Здесь  $f(\mathbf{X}, C_i)$  — радиальная функция с центром в точке с координатами  $C_i$ . Такие функции разнообразны, но на практике чаще всего используется функция Гаусса, имеющая следующий вид:

$$f(\mathbf{X}, C_i) = f(\|\mathbf{X} - C_i\|_2^2) = \exp\left(-\frac{\|\mathbf{X} - C_i\|_2^2}{2 \cdot \sigma_i^2}\right), \quad (3.5)$$

где  $\|\mathbf{X} - C_i\|_2$  — евклидова норма расстояния между входным вектором  $\mathbf{X}$  и центром  $C_i$  нейрона,  $\sigma_i$  — параметр, определяющий "ширину" функции. На рис. 3.3.10 представлены графики этой функции в скалярном варианте для различных значений  $\sigma_i$ .

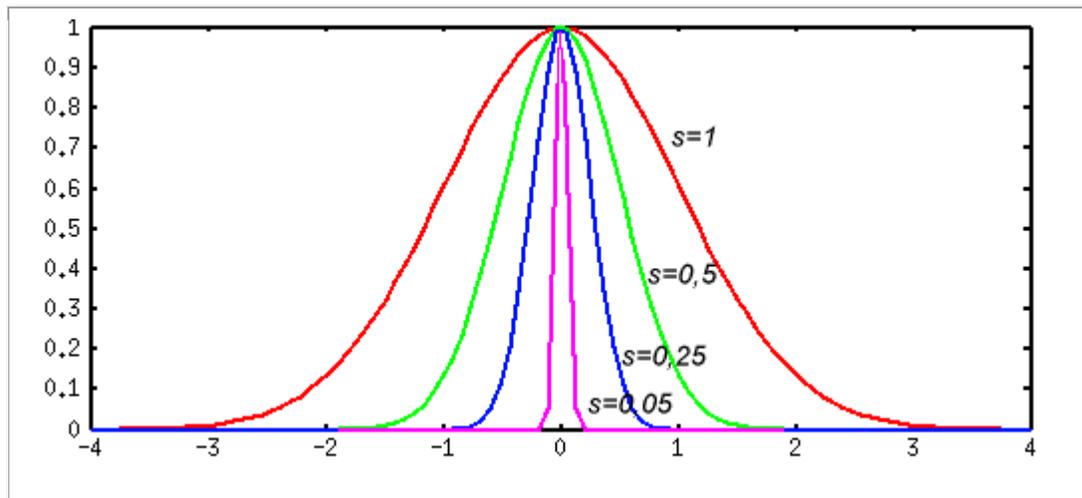


Рис. 3.3.11 . График одномерной радиальной функции

На рис. 3.3.12 дан график для двух входных сигналов.

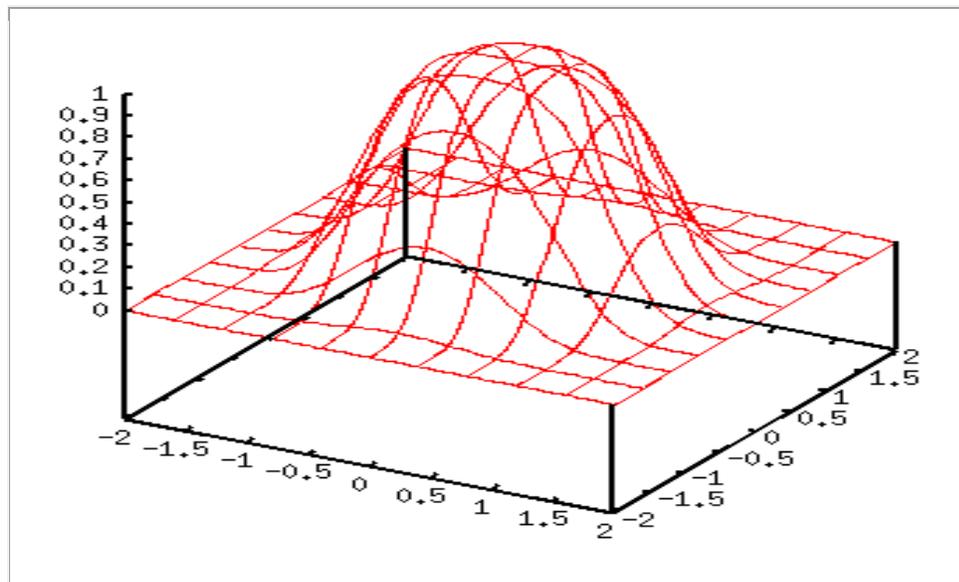


Рис. 3.3.12. График двумерной радиальной функции

Принципиальное отличие радиального нейрона от сигмоидального нейрона и персептрона в том, что они разбивают многомерное пространство входных сигналов гиперплоскостью, а радиальный — гиперсферой.

Обучение радиального нейрона заключается в подборе параметров радиальной функции  $C_i$  и  $\sigma_i$ .

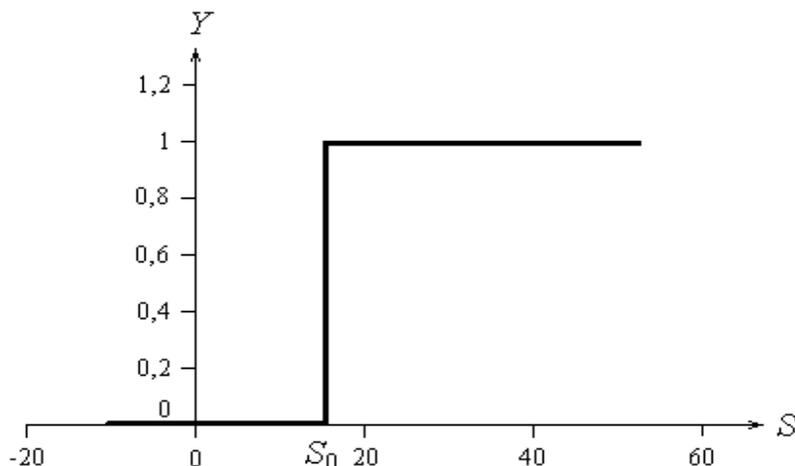
### 3.4. Активационные функции нейронных элементов

Наиболее распространенными функциями активации, нелинейными усилительными характеристиками нейронного элемента или передаточными функциями являются следующие: пороговая, сигнум, логистическая, гиперболический тангенс, линейная, линейная ограниченная, радиальная базисная и др.

1. Пороговая бинарная функция – это функция, для которой нейронный элемент остается неактивным до достижения входом порогового значения  $S_0$  (рис. 3.4.1.).

$$Y(s) = \begin{cases} 0, & S \leq S_0 \\ 1, & S > S_0 \end{cases}$$

Если  $S_0=0$ , то бинарная пороговая функция называется единичной функцией активации с жестким ограничением ( $\text{hardlim}(S)$ ).



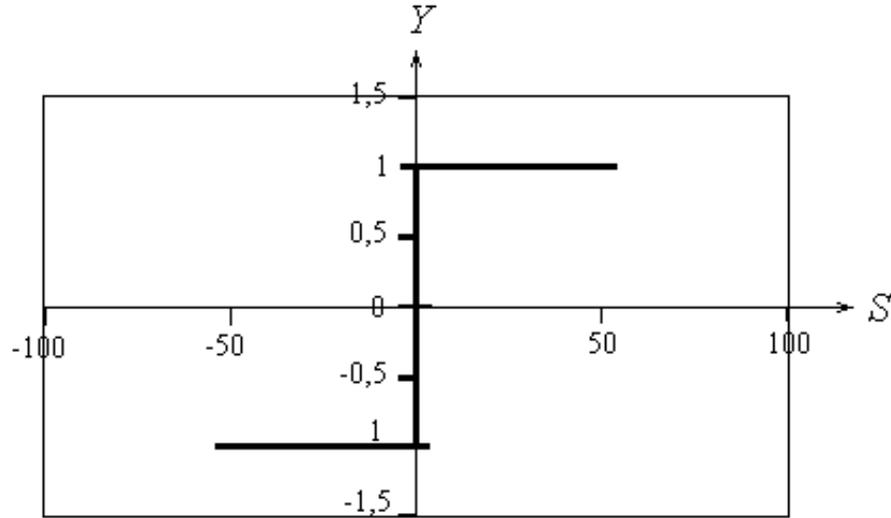


Рис. 3.4.1.). Бинарная пороговая функция

2. Сигнум, или модифицированная пороговая функция, для которой значение  $S_0=0$  (рис. 3.4.2), задается уравнением

$$Y(s) = \begin{cases} -1, & S < S_0 \\ 0, & S = 0 \\ 1, & S > S_0 \end{cases}$$

Рис. 3.4.2. Функция сигнум

3. Сигмоидная логистическая функция (S-образная, имеющая две горизонтальные асимптоты и одну точку перегиба) является возрастающей сжимающей функцией, значения которой принадлежат интервалу  $(0; 1)$  (рис. 3.4.3):

$$Y(s) = \frac{1}{1 + \exp(-c \cdot s)},$$

где  $c > 0$  – коэффициент, характеризующий крутизну логистической функции, усиливающей слабые сигналы ( $S$ )).

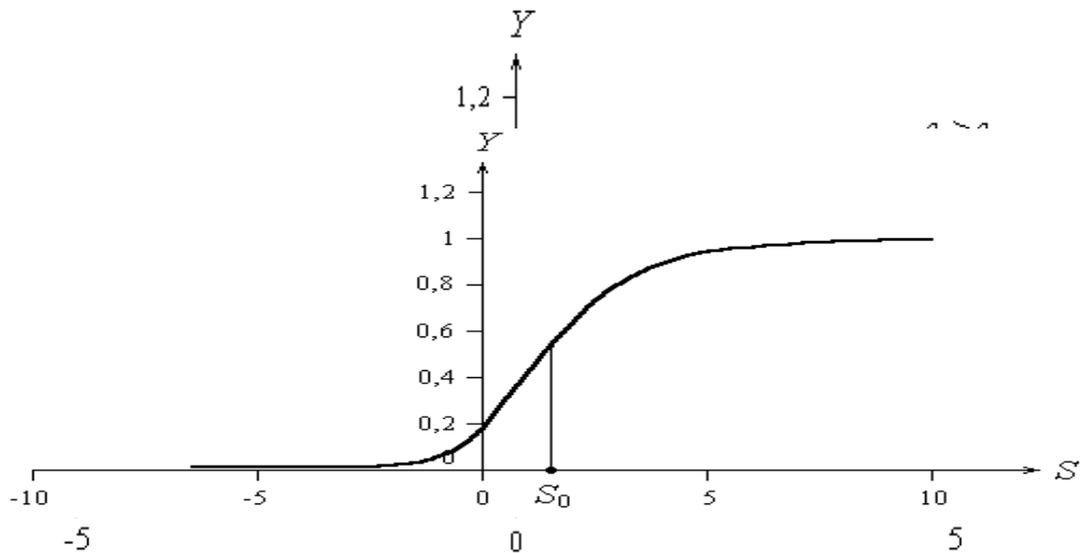


Рис. 3.4.4. Сигмовидная функция

В общем случае сигмовидная функция определяется уравнением:

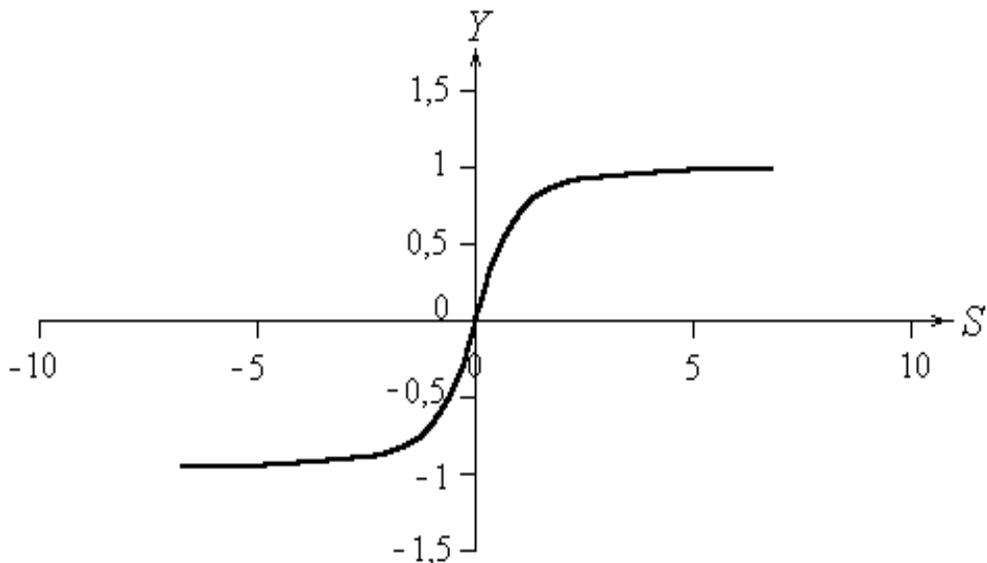
$$Y(s) = \frac{1}{1 + \exp(-c \cdot (s - s_0))}, \text{ где } S_0 \text{ – значение смещения. Если значение}$$

коэффициента  $c$  велико, то график общей сигмовидной функции приближается к пороговой функции (рис. 3.4.4).

Рис. 3.4.4. Общая сигмовидная функция

Биполярная сигмоидная функция (рис. 3.4.5), уравнение которой

$$Y(s) = \frac{2}{1 + \exp(-c \cdot s)} - 1,$$



принимает значения в диапазоне  $(-1; 1)$ .

Рис. 34.5. Биполярная сигмовидная функция

4. Гиперболический тангенс (рис. 7) аналогичен биполярной логистической функции без смещения и является симметричной функцией ( $\text{tansig}(S)$ ):

$$Y(s) = \frac{\exp(c \cdot s) - \exp(-c \cdot s)}{\exp(c \cdot s) + \exp(-c \cdot s)}.$$

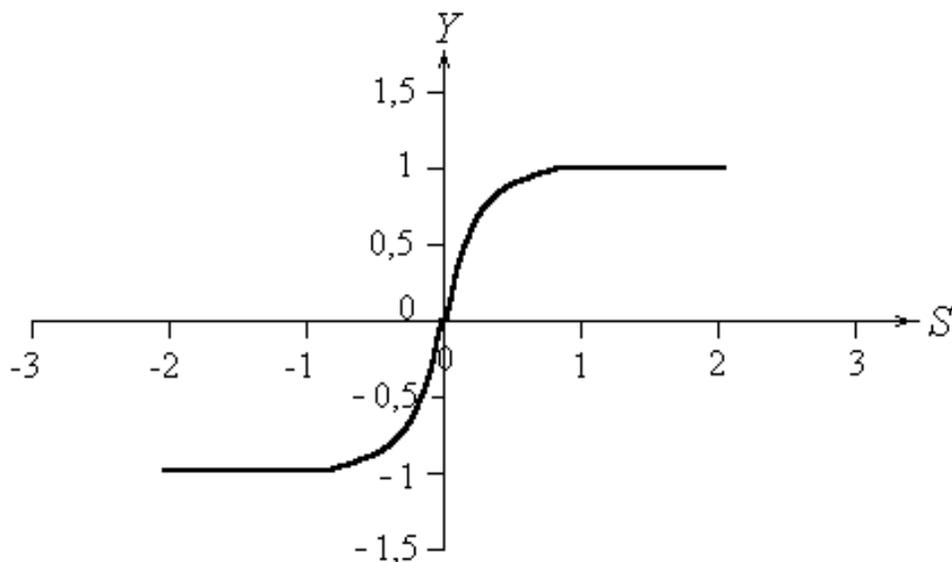


Рис. 34.6. Гиперболический тангенс

5. Линейная функция активации, уравнение которой  $Y(s)=k \cdot s$ , где  $k$  – угловой коэффициент наклона прямой, представлена на рис. 3 3.7 (S).

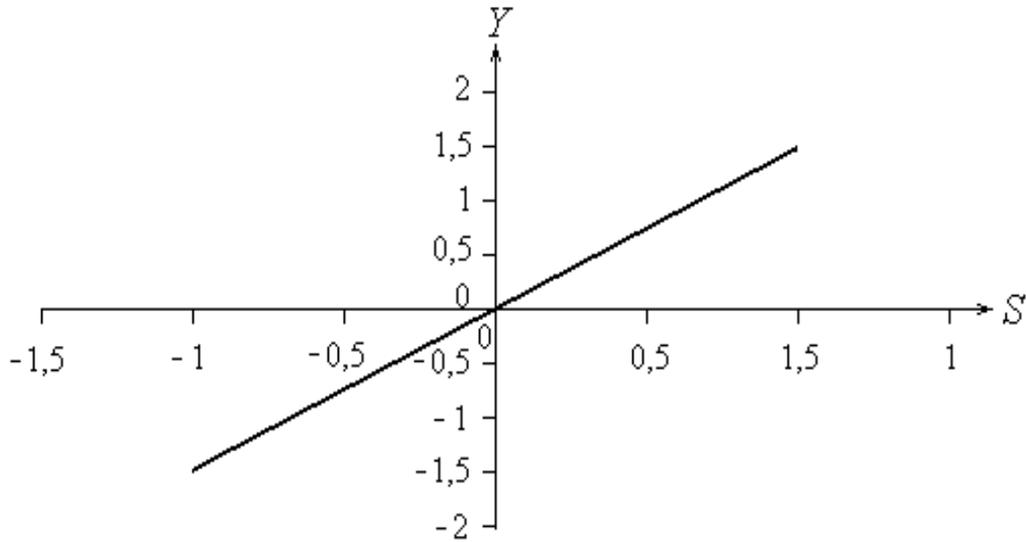


Рис. 3 4.7. Линейная функция

6. Линейная, ограниченная на отрезке  $[-a; a]$  функция активации (рис. 3 3.8) определяется по формуле:

$$Y(s) = \begin{cases} p, & s > a \\ -p, & s < -a \\ s, & -a \leq s \leq a \end{cases}$$

где  $p$ , как правило, равно единице:

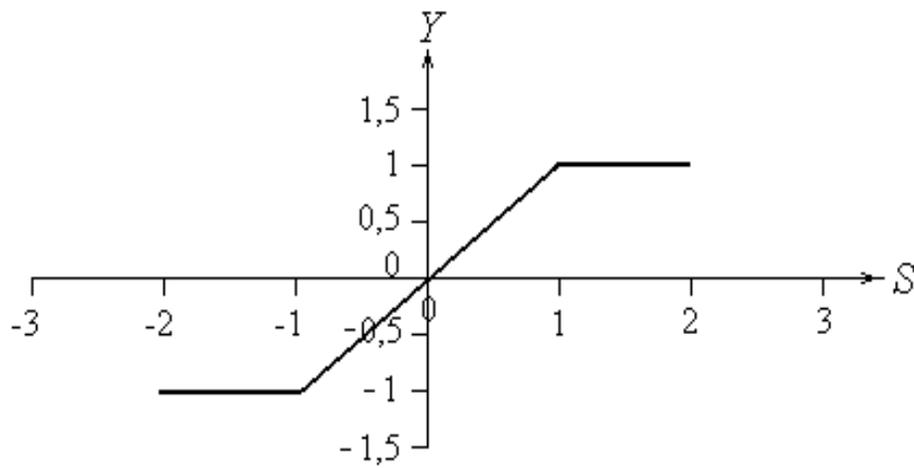


Рис. 34.8. Линейная ограниченная функция

7. Радиально-базисная функция активации характеризуется функцией Гаусса для нормального закона распределения, в соответствии с которой:

$$Y(s) = \exp\left(\frac{-s^2}{2 \cdot \sigma^2}\right),$$

где  $\sigma$  – среднее квадратичное отклонение, характеризующее крутизну радиально-базисной функции (рис.10). Величина  $s$  определяется в соответствии с евклидовым расстоянием между входным и весовым векторами:

$$S^2 = |X - W|^2 = \left( \sum_{i=1}^n (x_i - w_i)^2 \right).$$

Наряду с перечисленными функциями могут использоваться и другие функции активации, например, логарифмическая функция [2].

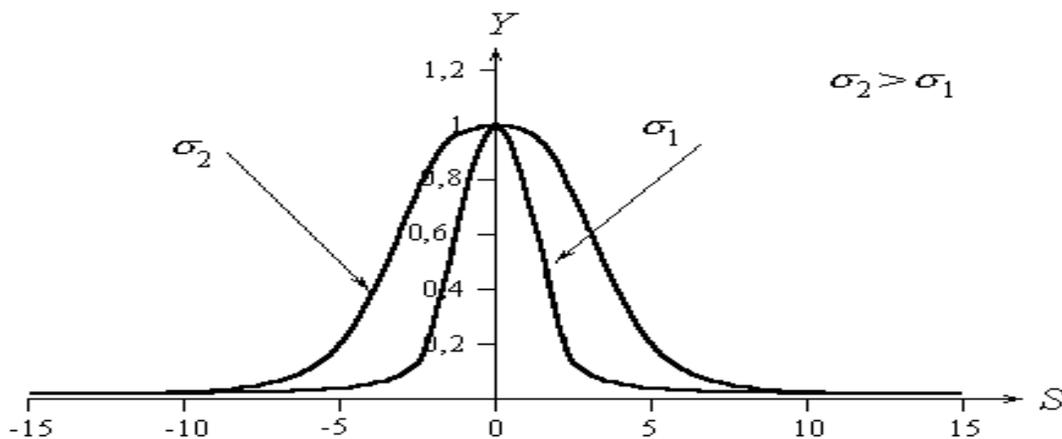


Рис. 34.9. Радиально-базисная функция

Многомерные радиальные распределения позволяют производить многомерный анализ путем сведения его к анализу одномерных симметричных распределений,

таких как многомерное нормальное распределение или равномерное в шаре с центром в начале координат [6].

Выбор функции активации определяется:

1. Спецификой задачи.
2. Удобством реализации на ЭВМ в виде электрической схемы или другим способом.
3. Алгоритмом обучения: некоторые алгоритмы накладывают ограничения на вид функции активации, их нужно учитывать.

Чаще всего вид нелинейности не оказывает принципиального влияния на решение задачи. Однако удачный выбор может сократить время обучения в несколько раз.

### **3.5. Алгоритмы обучения нейронных сетей**

Среди всех свойств искусственных нейронных сетей наиболее интересно так их способность к обучению, напоминающее процесс интеллектуального развития человека. Но возможности обучения искусственных нейронных сетей ограничены, и нужно еще решить много сложных задач для понимания процедур обучения.

Сеть обучается, чтобы множества входов соответствовало желаемому множеству выходов. Каждое такое входное (или выходное) множество рассматривается как вектор. Обучение осуществляется путем последовательного предъявления входных векторов с одновременной подстройкой весов в соответствии с определенной процедурой. В процессе обучения веса сети постепенно становятся такими, чтобы каждый входной вектор вырабатывал выходной вектор. Различают алгоритмы обучения с учителем и без учителя.

Обучение с учителем предполагает, что для каждого входного вектора существует целевой вектор, представляющий собой требуемый выход. Вместе

они называются обучающей парой. Обычно сеть обучается на некотором числе таких обучающих пар. Предъявляется выходной вектор, вычисляется выход сети и сравнивается с соответствующим целевым вектором, разность (ошибка) с помощью обратной связи подается в сеть и веса изменяются в соответствии с алгоритмом, стремящимся минимизировать ошибку. Векторы обучающего множества предъявляются последовательно, вычисляются ошибки и веса подстраиваются для каждого вектора до тех пор, пока ошибка по всему обучающему массиву не достигнет приемлемо низкого уровня.

Несмотря на многочисленные прикладные достижения, обучение с учителем критиковалось за свою биологическую неправдоподобность. Трудно вообразить обучающий механизм в мозге, который бы сравнивал желаемые и действительные значения выходов, выполняя коррекцию с помощью обратной связи. Обучение без учителя является намного более правдоподобной моделью обучения в биологической системе. Развита Кохоненом [3] и многими другими, она не нуждается в целевом векторе для выходов и, следовательно, не требует сравнения с predetermined идеальными ответами. Обучающее множество состоит лишь из входных векторов. Обучающий алгоритм подстраивает веса сети так, чтобы получались согласованные выходные векторы, т. е. чтобы предъявление достаточно близких входных векторов давало одинаковые выходы. Процесс обучения выделяет статистические свойства обучающего множества и группирует сходные векторы в классы. Предъявление на вход вектора из данного класса даст определенный выходной вектор, но до обучения невозможно предсказать, какой выход будет производиться данным классом входных векторов. Следовательно, выходы подобной сети должны преобразовываться в некоторую понятную форму, обусловленную процессом обучения.

Большинство современных алгоритмов обучения выросло из концепций Хэбба (см. 3.2.Нейрон Хебба). Им предложена модель обучения без учителя, в которой

синаптическая сила (вес) возрастает, если активированы оба нейрона, источник и приемник. Таким образом, часто используемые пути в сети усиливаются и феномен привычки и обучения через повторение получает объяснение.

В искусственной нейронной сети, использующей обучение по Хэббу, наращивание весов определяется произведением уровней возбуждения передающего и принимающего нейронов. Это можно записать как

$$w_{ij}(n+1) = w(n) + \alpha * x_i * y_j,$$

где  $w_{ij}(n)$  – значение веса от нейрона  $i$  к нейрону  $j$  до подстройки,  $w_{ij}(n+1)$  – значение веса от нейрона  $i$  к нейрону  $j$  после подстройки,  $\alpha$  – коэффициент скорости обучения,  $y_i$  – выход нейрона  $i$  и вход нейрона  $j$ ,  $x_j$  – выход нейрона  $j$ .

Сети, использующие обучение по Хэббу, конструктивно развивались, однако за последние 20 лет были разработаны более эффективные алгоритмы обучения с учителем, приводящие к сетям с широким диапазоном характеристик обучающих входных образов и большими скоростями обучения, чем использующие простое обучение по Хэббу.

Адаптация и самоорганизация искусственных нейронных сетей достигается в процессе их обучения. Целью обучения нейронных сетей является такая подстройка их весов, которая обеспечивала бы для некоторого множества входов требуемое множество выходов.

При решении прикладных задач с помощью нейронных сетей необходимо собрать достаточный объем данных, для того чтобы обучить нейронную сеть решению этих задач. Обучающий набор данных – это набор наблюдений, содержащих признаки изучаемого объекта. Первоначальный выбор признаков осуществляется на основе имеющегося опыта, с учетом мнения экспертов. Вопрос о том, сколько необходимо иметь данных для обучения сети, оказывается непростым. Известны эвристические правила, устанавливающие связь между количеством

используемых для обучения данных и размерами сети. Простейшее из них гласит, что количество наблюдений должно быть в 10 раз больше числа связей в сети.

Процесс обучения – это процесс определения параметров модели процесса или явления, реализуемого нейронной сетью. Ошибка обучения для конкретной конфигурации сети определяется после прохода через сеть всех имеющихся данных и сравнения выходных значений с целевыми значениями в случае обучения с учителем. Соответствующие разности позволяют сформировать так называемую функцию ошибок. Если ошибка сети, выходной слой которой имеет  $n$  нейронов,  $e_i = y_i - t_i$  есть разность между реальным и желаемым сигналами на выходе  $i$ -го нейрона, то в качестве функций ошибок могут быть использованы следующие функции:

- сумма квадратов ошибок 
$$sse = \sum_{i=1}^n e_i^2,$$
- средняя квадратичная ошибка 
$$mse = \frac{1}{n} \sum_{i=1}^n e_i^2,$$
- регулируемая или комбинированная ошибка

$$mse_{reg} = \frac{\gamma}{n} \sum_{i=1}^n e_i^2 + \frac{1-\gamma}{n} \sum_{i=1}^n e_i^2, \text{ где } \gamma - \text{ параметр регуляризации,}$$

- средняя абсолютная ошибка 
$$mae = \frac{1}{n} \sum_{i=1}^n |e_i|.$$

При моделировании нейронных сетей с линейными функциями активации нейронов можно построить алгоритм, гарантирующий достижение абсолютного минимума ошибки обучения. Для нейронных сетей с нелинейными функциями активации в общем случае нельзя гарантировать достижение глобального минимума функции ошибки. Поверхность функции ошибок определяется как совокупность точек-значений ошибок в  $N+1$ -мерном пространстве всевозможных сочетаний весов и смещений с общим числом  $N$ . Цель обучения при

геометрическом анализе или изучении поверхности ошибок состоит в том, чтобы найти на ней глобальный минимум. В случае линейной модели сети и минимизации суммы квадратов ошибок поверхность ошибок представляет собой параболоид, имеющий единственную точку минимума. В случае нелинейной модели поверхность ошибок имеет сложное строение и может иметь локальные минимумы, плоские участки, седловые точки и длинные узкие овраги.

Как правило, при обучении такой сети вычисляется градиент функции ошибок в случайно выбранной точке поверхности, а затем эта информация используется для продвижения вниз по склону. Алгоритм продвижения завершается в точке минимума, локального или глобального. По существу алгоритмы обучения нейронных сетей аналогичны алгоритмам поиска глобального экстремума функции многих переменных.

Сети с большим количеством весов позволяют воспроизводить сложные функции, но в этом случае возможно так называемое «переобучение» сети, когда ошибки обучения малы, но полученная модель имеет слабое отношение к истинной зависимости. Нейронная сеть же с небольшим количеством весов может оказаться недостаточно гибкой, чтобы смоделировать имеющуюся зависимость. Для преодоления эффекта переобучения используется контрольная проверка. Часть обучающих наблюдений резервируется как контрольные наблюдения и не используется при обучении сети. Если контрольная ошибка перестает убывать или начинает расти, то это означает, что сеть слишком близко следует исходным данным и обучение следует остановить. В этом случае необходимо уменьшить количество нейронов или слоев, так как сеть является слишком мощной для решаемой задачи. Если же сеть имеет недостаточную мощность для воспроизведения имеющейся зависимости, необходимо добавить нейронов или даже слоев.

Способность сети, обученной на некотором множестве данных выдавать правильные результаты для достаточно широкого класса новых данных, в том

числе и не представленных при обучении, называется свойством обобщения нейронной сети.

Для настройки параметров нейронных сетей широко используется также процедура адаптации, когда подбираются веса и смещения с использованием произвольных функций их настройки, обеспечивающие соответствие между входами и желаемыми значениями на выходе.

Методы определения экстремума функции нескольких переменных делятся на три категории – методы нулевого, первого и второго порядка:

- методы нулевого порядка, в которых для нахождения экстремума используется только информация о значениях функции в заданных точках;
- методы первого порядка, где для нахождения экстремума используется градиент функционала ошибки по настраиваемым параметрам;
- методы второго порядка, вычисляющие матрицу вторых производных функционала ошибки (матрицу Гессе).

Кроме перечисленных методов можно выделить также стохастические алгоритмы оптимизации и алгоритмы глобальной оптимизации, перебирающие значения аргументов функции ошибки.

### **Правило обучения Хебба**

Правила обучения Хебба относятся к ассоциативным обучающим правилам. Согласно правилу Хебба обучение сети происходит в результате усиления силы связи между одновременно активными элементами, его можно определить так:

$$w_{ij}(t+1) = w_{ij}(t) + x_i \cdot y_j, \quad (3.6)$$

где  $t$  – время,  $x_i$ ,  $y_j$  – соответственно выходные значения  $i$ -го и  $j$ -го нейронов,  $\alpha$  – скорость обучения равна 1. В начальный момент предполагается, что  $w_{ij}(0)=0$  для всех  $i$  и  $j$ . Правило Хебба можно использовать как при обучении с учителем, так и без него. Если в качестве выходных значений сети

используются целевые значения, тогда реализуется обучение с учителем. При использовании в качестве у реальных значений, которые получаются при подаче входных образов на вход сети, правило Хебба соответствует обучению без учителя. В этом случае коэффициенты сети в начальный момент времени инициализируются случайным образом. Обучение заканчивается после подачи всех имеющихся входных образов на нейронную сеть. В общем случае это правило не гарантирует сходимости процедуры обучения сети.

Требуется реализовать обучение нейронной сети по правилу Хебба с учителем для операции «логическое И», используя биполярную кодировку двоичных сигналов.

Общее количество входных образов, подаваемых на нейронную сеть,  $L=4$ . Правило Хебба в случае сети с тремя нейронами на входе и одним на выходе реализуется по формулам:

$$\begin{aligned} w_{11}(t+1) &= w_{11}(t) + x_1 \cdot y, \\ w_{21}(t+1) &= w_{21}(t) + x_2 \cdot y, \\ S_0(t+1) &= S_0(t) - y. \end{aligned} \quad (3.7)$$

Обучение с учителем для «логического И» соответствует данным, приведенным в табл. 1.

Таблица 1

**Результаты обучения сети по правилу Хебба**

$X_1$	$X_2$	$X_3=S_0$	$Y$	$W_{11}$	$W_{21}$	$S_0$
-1	-1	-1	-1	1	1	1
1	-1	-1	-1	0	2	2
-1	1	-1	-1	1	1	3
1	1	-1	1	2	2	2

Эта таблица формируется по строкам. Считая  $S_0(0)=0$ , начальные  $w_{11}^{(0)} = 0$ ,

$w_{21}^{(0)} = 0$ , для четырех входных образов получаем

последовательно следующие значения для первого образа

$$w_{11}^{(1)} = w_{11}^{(0)} + x_1 \cdot y = 0 + (-1) \cdot (-1) = 1,$$

$$w_{21}^{(1)} = w_{21}^{(0)} + x_2 \cdot y = 0 + (-1) \cdot (-1) = 1,$$

$$S_0(1) = S_0(0) - y = 0 - (-1) = 1;$$

для второго

$$w_{11}^{(2)} = w_{11}^{(1)} + x_1 \cdot y = 1 + 1 \cdot (-1) = 0,$$

$$w_{21}^{(2)} = w_{21}^{(1)} + x_2 \cdot y = 1 + (-1) \cdot (-1) = 2,$$

$$S_0(2) = S_0(1) - y = 1 - (-1) = 2;$$

для третьего

$$w_{11}^{(3)} = w_{11}^{(2)} + x_1 \cdot y = 0 + (-1) \cdot (-1) = 1,$$

$$w_{21}^{(3)} = w_{21}^{(2)} + x_2 \cdot y = 2 + 1 \cdot (-1) = 1,$$

$$S_0(3) = S_0(2) - y = 2 - (-1) = 3;$$

и для четвертого образа

$$w_{11}^{(4)} = w_{11}^{(3)} + x_1 \cdot y = 1 + 1 \cdot 1 = 2,$$

$$w_{21}^{(4)} = w_{21}^{(3)} + x_2 \cdot y = 1 + 1 \cdot 1 = 2,$$

$$S_0(4) = S_0(3) - y = 3 - 1 = 2.$$

Таким образом, в результате обучения получается уравнение разделяющей линии для операции «логическое И», представленной ранее на рис. 3.2.6:

$$2 \cdot x_1 + 2 \cdot x_2 - 2 = 0, \quad \text{или} \quad x_2 = 1 - x_1.$$

Правило обучения по Хеббу в общем случае задается формулой:

$$w_{ij}(t+1) = w_{ij}(t) + \eta \cdot x_i \cdot y_j, \quad i=1,2,\dots,L, \quad j=1,2,\dots,p,$$

где  $\eta$  – коэффициент обучения или темп обучения, в нейроимитаторе NNT это правило реализует процедура настройки learnh. Правило Хебба с модификацией весов определяется по формуле:

$$w_{ij}(t+1) = w_{ij}(t) + \eta \cdot x_i \cdot y_j - \alpha \cdot \Delta w_{ij}(t),$$

где  $\alpha$  – коэффициент убывания или возрастания веса.

Для решения задачи «исключающего ИЛИ» можно использовать сеть второго порядка, включающую произведение переменных, тогда взвешенная сумма  $S = 2 \cdot x_1 + 2 \cdot x_2 - 4 \cdot x_1 \cdot x_2 - 1$ . (Рис.3.2.7)

### **Правило обучения перцептрона**

Перцептрон – нейронная сеть прямой передачи сигнала с бинарными входами и бинарной пороговой функцией активации.

Правило обучения Розенблатта в общем случае является вариантом правил обучения Хебба, формирующих симметричную матрицу связей, и в тех же обозначениях имеет вид:

$$w_{ij}(t+1) = w_{ij}(t) + \eta \cdot (t_i - y_j) \cdot x_i, \quad (3.8)$$

где  $\eta$  – коэффициент обучения,  $0 < \eta < 1$ ,  $t_j$  – эталонные или целевые значения.

Правило обучения перцептрона с нормализацией используется тогда, когда имеется большой разброс в значениях компонент входного вектора. Тогда каждая компонента делится на длину вектора:

$$x'_i = \frac{x_i}{\sqrt{x_1^2 + x_2^2 + \dots + x_n^2}}, \quad i=1,2,\dots,n.$$

### **Алгоритм обучения Розенблатта:**

1. Весовые коэффициенты сети инициализируются случайным образом или устанавливаются нулевыми.
2. На вход сети поочередно подаются входные образы из обучающей выборки, которые затем преобразуются в выходные сигналы  $Y$ .

3. Если реакция нейронной сети  $y_j$  совпадает с эталонным значением  $t_j$ , то весовой коэффициент  $w_{ij}$  не изменяется.
4. Если выходное значение не совпадает с эталонным значением, то есть  $y_j$  не равно  $t_j$ , то производится модификация весовых коэффициентов  $w_{ij}$  по соответствующей формуле.
5. Алгоритм повторяется до тех пор, пока для всех входных образов не будет достигнуто совпадение с целевыми значениями или не перестанут изменяться весовые коэффициенты.

Линейная разделяющая поверхность, формируемая персептроном, позволяет решать ограниченный круг задач.

### **Правило обучение Видроу - Хоффа**

Правило Видроу – Хоффа используется для обучения сети, состоящей из слоя распределительных нейронов и одного выходного нейрона с линейной функцией активации. Такая сеть называется адаптивным нейронным элементом (Adaptive Linear Element) или «ADALINE». Выходное значение такой сети определяется по формуле

$$y = \sum_{i=1}^n w_{i1} \cdot x_i - S_0 ,$$

где  $n$  – число нейронов распределительного слоя,  $S_0$  – смещение.

Правило обучения Видроу – Хоффа известно под названием дельта-правила (delta-rule). Оно предполагает минимизацию среднеквадратичной ошибки нейронной сети, которая для  $L$  входных образов определяется следующим образом:

$$E = \frac{1}{2} \cdot \sum_{k=1}^L \left( y^{(k)} - t^{(k)} \right)^2 , \quad (3.9)$$

где  $y^{(k)}$  и  $t^{(k)}$  – выходное и целевое значения сети соответственно для  $k$ -го образа. Правило Видроу – Хоффа базируется на методе градиентного спуска в пространстве весовых коэффициентов и смещений нейронной сети. По этому методу веса и смещения изменяются с течением времени следующим образом:

$$w_{i1}(t+1) = w_{i1}(t) - \alpha \cdot \frac{\partial E(k)}{\partial w_{i1}(t)}, \quad S_0(t+1) = S_0(t) - \alpha \cdot \frac{\partial E(k)}{\partial S_0(t)}, \quad i=1,2,\dots,n,$$

где  $\alpha$  – скорость обучения,  $0 < \alpha < 1$ .

Вычислим частные производные:

$$\frac{\partial E(k)}{\partial w_{i1}(t)} = \frac{\partial E(k)}{\partial y^{(k)}} \cdot \frac{\partial y^{(k)}}{\partial w_{i1}(t)} = (y^{(k)} - t^{(k)}) \cdot x_i^{(k)},$$

$$\frac{\partial E(k)}{\partial S_0(t)} = \frac{\partial E(k)}{\partial y^{(k)}} \cdot \frac{\partial y^{(k)}}{\partial S_0(t)} = -(y^{(k)} - t^{(k)}),$$

где  $x_i^{(k)}$  –  $i$ -я компонента  $k$ -го образа. По дельта-правилу

$$w_{i1}(t+1) = w_{i1}(t) - \alpha \cdot (y^{(k)} - t^{(k)}) \cdot x_i^{(k)},$$

$$S_0(t+1) = S_0(t) + \alpha \cdot (y^{(k)} - t^{(k)}), \quad i = 1, 2, \dots, n.$$

Выражение для пересчета весов сети эквивалентно правилу обучения Розенблатта, если на выходе линейного адаптивного элемента использовать пороговый элемент. Видроу и Хофф доказали, что минимизация среднеквадратичной ошибки сети производится независимо от начальных значений весовых коэффициентов. Суммарная среднеквадратичная ошибка сети  $E$  должна достигнуть заданного минимального значения  $E_{\min}$ . Для алгоритма Видроу – Хоффа рекомендуется выбирать значение скорости обучения  $\alpha$  по следующему правилу:  $\alpha(l) = \frac{1}{l}$ , где  $l$  – номер итерации в алгоритме обучения.

Адаптивный шаг обучения для линейной нейронной сети позволяет повысить скорость обучения и определяется по формуле [2]:

$$\alpha(t) = \frac{1}{1 + \sum_{i=1}^n x_i^2(t)}.$$

### Способ обучения с обратным распространением ошибки

Этот способ обратного распространения ошибки предложен несколькими авторами независимо в 1986 г. для многослойных сетей с прямой передачей сигнала. Многослойная нейронная сеть способна осуществлять любое отображение входных векторов в выходные (рис. 3.5.1)

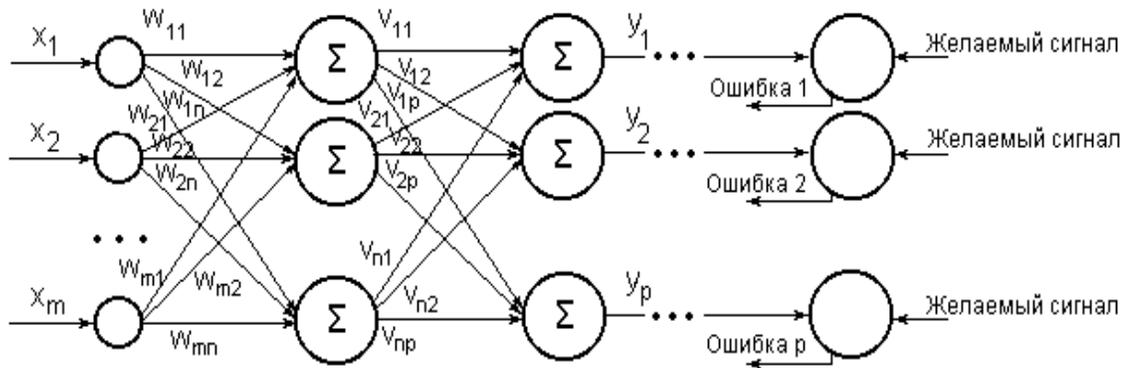


Рис. 3.5.1. Многослойная сеть со скрытым слоем из  $n$  нейронов

Входной слой нейронной сети выполняет распределительные функции. Выходной слой нейронов служит для обработки информации от предыдущих слоев и выдачи результатов. Слои нейронных элементов, расположенные между входным и выходным слоями, называются скрытыми. Как и выходной слой, скрытые слои являются обрабатывающими. Выход каждого нейрона предыдущего слоя сети соединен синаптическими связями со всеми входами нейронных элементов следующего слоя. В качестве функций активации в многослойных сетях чаще других используются сигмоидальная функция и гиперболический тангенс. Метод обратного распространения ошибки минимизирует среднеквадратичную ошибку нейронной сети, при этом

используется метод градиентного спуска в пространстве весовых коэффициентов и смещений нейронных сетей. Согласно методу градиентного спуска по поверхности ошибок изменение весовых коэффициентов и смещений сети осуществляется по формулам:

$$w_{ij}(t+1) = w_{ij}(t) - \alpha \cdot \frac{\partial E(k)}{\partial w_{ij}(t)}, \quad (3.10)$$

$$S_{0j}(t+1) = S_{0j}(t) - \alpha \cdot \frac{\partial E(k)}{\partial S_{0j}(t)}, \quad i=1,2,\dots,n; \quad j=1,2,\dots,p,$$

где  $E$  – среднеквадратичная ошибка сети для одного из  $k$  образов, определяемая по формуле

$$E = \frac{1}{2} \sum_{j=1}^p (y_j - t_j)^2,$$

здесь  $t_j$  – желаемое или целевое выходное значение  $j$ -го нейрона.

Ошибка  $j$ -го нейрона выходного слоя равна:  $\gamma_j = y_j - t_j$ .

Для любого скрытого слоя ошибка  $i$ -го нейронного элемента определяется рекурсивно через ошибки нейронов следующего слоя  $j$ :

$$\gamma_i = \sum_{j=1}^m \gamma_j \cdot F'(S_j) \cdot w_{ij},$$

где  $m$  – количество нейронов следующего слоя по отношению к слою  $i$ ,  $w_{ij}$  – вес или синаптическая связь между  $i$ -ым и  $j$ -ым нейронами разных слоев,  $S_j$  – взвешенная сумма  $j$ -го нейрона.

Весовые коэффициенты и смещения нейронных элементов с течением времени изменяются следующим образом:

$$w_{ij}(t+1) = w_{ij}(t) - \alpha \cdot \gamma_j \cdot F'(S_j) \cdot y_i,$$

$$S_{0j}(t+1) = S_{0j}(t) + \alpha \cdot \gamma_j \cdot F'(S_j), \quad i=1,2,\dots,n, \quad j=1,2,\dots,p,$$

где  $\alpha$  – скорость обучения. Это правило называется обобщенным дельта-правилом. Для сигмоидальной функции активации:

$$w_{ij}(t+1) = w_{ij}(t) - \alpha \cdot \gamma_j \cdot y_j(1 - y_j) \cdot y_i$$

$$S_{0j}(t+1) = S_{0j}(t) + \alpha \cdot \gamma_j \cdot y_j(1 - y_j), \quad i=1,2,\dots,n, \quad j=1,2,\dots,p,$$

ошибка  $j$ -го нейрона выходного слоя определяется как

$$\gamma_j = y_j - t_j,$$

а  $j$ -го нейрона скрытого слоя

$$\gamma_j = \sum_{i=1}^m \gamma_j \cdot y_i \cdot (1 - y_i) \cdot w_{ij}, \text{ где}$$

$m$  – количество нейронов следующего слоя по отношению к слою  $j$ .

К недостаткам метода обратного распространения ошибки относят следующие:

- медленную сходимость градиентного метода при постоянном шаге обучения;
- возможное смешение точек локального и глобального минимумов;
- влияние случайной инициализации весовых коэффициентов на скорость поиска минимума.

Для их преодоления предложено несколько модификаций алгоритма обратного распространения ошибки:

1) с импульсом, позволяющим учесть текущий и предыдущий градиенты (метод тяжелого шарика), изменение веса тогда:

$$\Delta w_{ij}(t+1) = \alpha \cdot \gamma_j \cdot F'(S_j) \cdot y_i + \eta \cdot \Delta w_{ij}(t),$$

где  $\alpha$  – коэффициент скорости обучения,  $\eta$  – импульс или момент, обычно  $0 < \alpha < 1$ ,  $\eta \approx 0,9$ ;

2) с адаптивным шагом обучения, изменяющимся по формуле

$$\alpha(t) = \frac{1}{1 + \sum x_i^2(t)};$$

3) с модификацией по Розенбергу, предложенной им для решения задачи преобразования английского печатного текста в качественную речь:

$$\Delta w_{ij}(t+1) = (1 - \alpha) \gamma_j \cdot F'(S_j) \cdot y_i + \alpha \cdot \Delta w_{ij}(t),$$

$$w_{ij}(t+1) = w_{ij}(t) + \eta \cdot \Delta w_{ij}(t+1).$$

Использование производных второго порядка для коррекции весов алгоритма обратного распространения ошибки, как показала практика, не дало значительного улучшения решений прикладных задач.

Для многослойных сетей прямой передачи сигнала с логистической функцией активации рекомендуется случайные начальные значения весов инициализировать по правилу (Р. Палмер)

$$w_{ij} \approx \frac{1}{\sqrt{n(i)}},$$

где  $n(i)$  – количество нейронных элементов в слое  $i$ .

По мнению других авторов [2], начальные весовые коэффициенты следует выбирать в диапазоне  $[-0.05; 0.05]$  или  $[-0.1; 0.1]$ . При этом смещение  $S_0$  принимает единичные значения в начальный момент времени. Кроме этого, количество нейронных элементов скрытых слоев должно быть меньше тренировочных образов. Для обеспечения требуемой обобщающей способности сети можно использовать сеть с несколькими скрытыми слоями, размерность которых меньше, чем для сети с одним скрытым слоем. Однако нейронные сети, имеющие несколько скрытых слоев, обучаются значительно медленнее.

Искусственные нейронные сети могут обучаться по существу тем же самым образом посредством случайной коррекции весов. Вначале делаются большие случайные коррекции с сохранением только тех изменений весов, которые

уменьшают целевую функцию. Затем средний размер шага постепенно уменьшается, и глобальный минимум в конце концов достигается.

### **Обучение нейронных сетей способом отжига**

Это напоминает отжиг металла, поэтому для ее описания часто используют термин «имитация отжига». В металле, нагретом до температуры, превышающей его точку плавления, атомы находятся в сильном беспорядочном движении. Как и во всех физических системах, атомы стремятся к состоянию минимума энергии (единому кристаллу в данном случае), но при высоких температурах энергия атомных движений препятствует этому. В процессе постепенного охлаждения металла возникают все более низкоэнергетические состояния, пока не будет достигнут глобальный минимум. В процессе отжига распределение энергетических уровней описывается следующим соотношением:

$$P(e) = \exp(-e/kT) \quad (3.11)$$

где  $P(e)$  – вероятность того, что система находится в состоянии с энергией  $e$ ;  $k$  – постоянная Больцмана;  $T$  – температура по шкале Кельвина.

При высоких температурах  $P(e)$  приближается к единице для всех энергетических состояний. Таким образом, высокоэнергетическое состояние почти столь же вероятно, как и низкоэнергетическое. По мере уменьшения температуры вероятность высокоэнергетических состояний уменьшается по сравнению с низкоэнергетическими. При приближении температуры к нулю становится весьма маловероятным, чтобы система находилась в высокоэнергетическом состоянии.

### **Алгоритм больцмановского обучения**

Этот стохастический метод непосредственно применим к обучению искусственных нейронных сетей:

1. Определить переменную  $T$ , представляющую искусственную температуру. Придать  $T$  большое начальное значение.
2. Предъявить сети множество входов и вычислить выходы и целевую функцию.
3. Дать случайное изменение весу и пересчитать выход сети и изменение целевой функции в соответствии со сделанным изменением веса.
4. Если целевая функция уменьшилась (улучшилась), то сохранить изменение веса.

Если изменение веса приводит к увеличению целевой функции, то вероятность сохранения этого изменения вычисляется с помощью распределения Больцмана:

$$P(c) = \exp(-c/kT) \quad (3.12)$$

где  $P(c)$  – вероятность изменения  $c$  в целевой функции;  $k$  – константа, аналогичная константе Больцмана, выбираемая в зависимости от задачи;  $T$  – искусственная температура.

Выбирается случайное число  $r$  из равномерного распределения от нуля до единицы. Если  $P(c)$  больше, чем  $r$ , то изменение сохраняется, в противном случае величина веса возвращается к предыдущему значению.

Система выполняет случайный шаг в направлении, искажающем целевую функцию, позволяя ей тем самым вырваться из локальных минимумов, где любой малый шаг увеличивает целевую функцию.

Для завершения больцмановского обучения повторяют шаги 3 и 4 для каждого из весов сети, постепенно уменьшая температуру  $T$ , пока не будет достигнуто допустимо низкое значение целевой функции. В этот момент предъявляется другой входной вектор, и процесс обучения повторяется. Сеть обучается на всех векторах обучающего множества, с возможным повторением, пока целевая функция не станет допустимой для всех них.

Величина случайного изменения веса на шаге 3 может определяться различными способами. Например, подобно тепловой системе весовое изменение  $w$  может выбираться в соответствии с гауссовским законом распределения:

$$P(w) = \exp(-w^2/T^2),$$

где  $P(w)$  – вероятность изменения веса на величину  $w$ ,  $T$  – искусственная температура.

Скорость уменьшения температуры должна быть обратно пропорциональна логарифму времени, чтобы была достигнута сходимость к глобальному минимуму. Скорость охлаждения в такой системе выражается следующим образом:

$$T(t) = \frac{T_0}{\log(1+t)}$$

где  $T(t)$  – искусственная температура как функция времени;  $T_0$  – начальная искусственная температура;  $t$  – искусственное время.

Машины Больцмана часто требуют для обучения очень большого ресурса времени.

### **Обучение Коши**

В работе [6] развит метод быстрого обучения подобных систем. В этом методе при вычислении величины шага распределение Больцмана заменяется на распределение Коши. Распределение Коши имеет, как показано на рис.3.4.2 более длинные «хвосты», увеличивая тем самым вероятность больших шагов. В действительности распределение Коши имеет бесконечную (неопределенную) дисперсию. С помощью такого простого изменения максимальная скорость уменьшения температуры становится обратно пропорциональной линейной величине, а не логарифму, как для алгоритма обучения Больцмана. Это резко

уменьшает время обучения. Эта связь может быть выражена следующим образом:

$$T(t) = \frac{T_0}{1+t} \quad (3.13)$$

Распределение Коши имеет вид

$$P(x) = \frac{T(t)}{T(t)^2 + x^2} \quad (3.14)$$

где  $P(x)$  есть вероятность шага величины  $x$ .

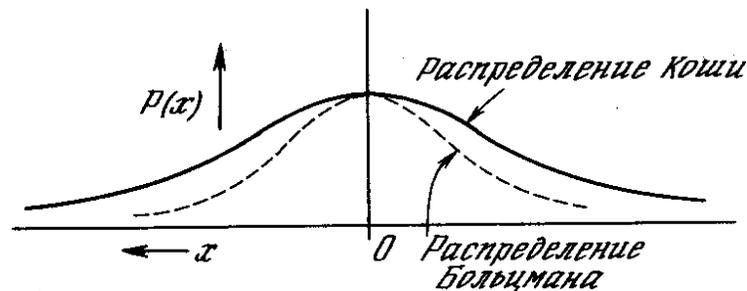


Рис. 3.4.2. Распределение Коши и распределение Больцмана

В уравнении (5.6)  $P(x)$  может быть проинтегрирована стандартными методами.

Решая относительно  $x$ , получаем

$$x_c = \rho T(t) \operatorname{tg}(P(x)),$$

где  $\rho$  – коэффициент скорости обучения;  $x_c$  – изменение веса.

Для нахождения  $x$  в этом случае (метод Монте Карло) выбирается случайное число из равномерного распределения на открытом интервале  $(-\pi/2, \pi/2)$  (необходимо ограничить функцию тангенса). Оно подставляется в формулу (5.7) в качестве  $P(x)$ , и с помощью текущей температуры вычисляется величина шага.

## Метод искусственной теплоемкости

Несмотря на улучшение, достигаемое с помощью метода Коши, время обучения может оказаться все еще слишком большим. Способ, уходящий своими корнями в термодинамику, может быть использован для ускорения этого процесса. В этом методе скорость уменьшения температуры изменяется в соответствии с искусственной «теплоемкостью», вычисляемой в процессе обучения.

Во время отжига металла происходят фазовые переходы, связанные с дискретными изменениями уровней энергии. При каждом фазовом переходе может иметь место резкое изменение величины, называемой теплоемкостью. Теплоемкость определяется как скорость изменения температуры с энергией. Изменения теплоемкости происходят из-за попадания системы в локальные энергетические минимумы.

Искусственные нейронные сети проходят аналогичные фазы в процессе обучения. На границе фазового перехода искусственная теплоемкость может скачкообразно измениться. Эта псевдотеплоемкость определяется как средняя скорость изменения температуры с целевой функцией. Сильное начальное возбуждение делает среднюю величину целевой функции фактически не зависящей от малых изменений температуры, т. е. теплоемкость близка к константе. Аналогично при очень низких температурах система останавливается в точке минимума, так что теплоемкость снова близка к константе. Ясно, что в каждой из этих областей допустимы сильные изменения температуры, так как не происходит улучшения целевой функции.

При критических температурах небольшое уменьшение температуры приводит к большому изменению средней величины целевой функции. Средняя величина целевой функции испытывает скачкообразное изменение. В этих критических точках алгоритм должен изменять температуру очень медленно, чтобы гарантировать, что система не окажется случайно в точке, локального минимума. Критическая температура может быть обнаружена по резкому

уменьшению искусственной теплоемкости, т. е. средней скорости изменения температуры с целевой функцией. При достижении критической температуры скорость изменения температуры должна замедляться, чтобы гарантировать сходимость к глобальному минимуму. При всех остальных температурах может без риска использоваться более высокая скорость снижения температуры, что приводит к значительному снижению времени обучения.

### 3.6. Нейронные сети прямого распространения

#### Персептрон

Модель МакКаллока-Питтса лежит в основе простейшей однонаправленной нейронной сети, названной персептроном. Эту сеть предложил и исследовал Розенблатт в конце пятидесятых - начале шестидесятых годов XX века. На рис. 3.6.1 представлена структура персептрона, иногда называемого простейшим персептроном.

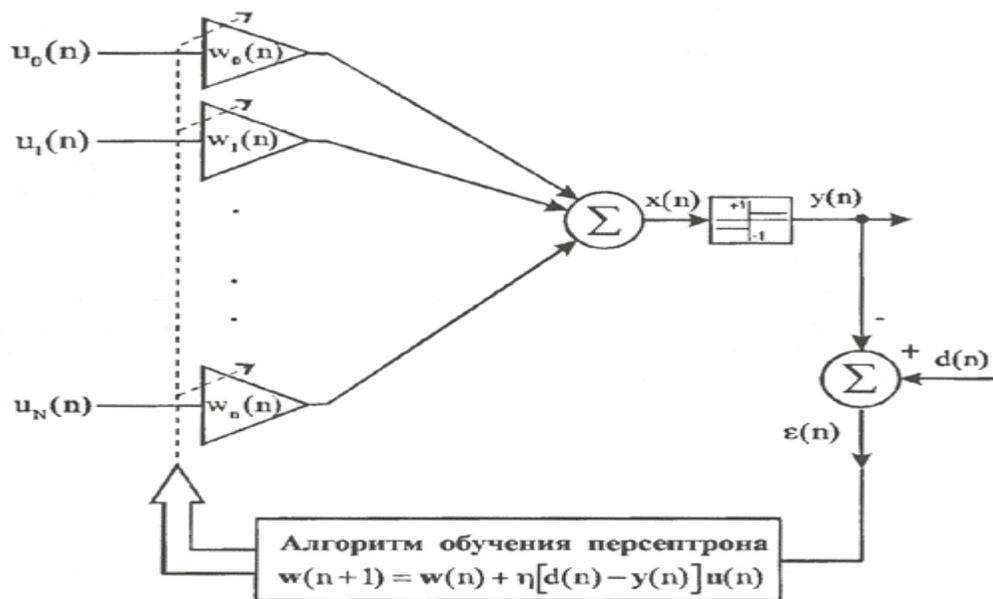


Рис. 3.6.1 Персептрон.

В качестве функции  $f$  в модели МакКаллока-Питтса (2.2) применялась биполярная функция активации (см.3.3.3).

Сигнал  $x$  на выходе линейной части персептрона задается выражением

$$x = \sum_{i=1}^N w_i u_i - \nu = \sum_{i=0}^N w_i u_i, \quad (3.15)$$

где  $w_0 = \nu$ ,  $u_0 = -1$ .

Задача персептрона заключается в классификации вектора  $u = [u_1, \dots, u_N]^T$  в смысле отнесения его к одному из двух классов, обозначаемых символами  $L_1$  и  $L_2$ . Персептрон относит вектор  $u$  к классу  $L_1$ , если выходной сигнал  $y$  принимает значение 1, и к классу  $L_2$ , если выходной сигнал  $y$  принимает значение  $-1$ . После этого персептрон разделяет  $N$ -мерное пространство входных векторов  $u$  на два полупространства, разделяемые  $(N-1)$ -мерной гиперплоскостью, задаваемой уравнением

$$\sum_{i=1}^N w_i u_i - \nu = \sum_{i=0}^N w_i u_i = 0.$$

Гиперплоскость (2.9) называется решающей границей. Если  $N=2$ , то решающая граница - это прямая линия, задаваемая уравнением

$$w_1 u_1 + w_2 u_2 - \nu = 0.$$

Точка  $(u_1, u_2)$ , лежащая над этой прямой (рис. 3.6.2.), относится к классу  $L_1$ , тогда как точка  $(u_1, u_2)$ , лежащая под этой прямой, относится к классу  $L_2$ .

Точки, лежащие на границе решения, можно произвольно отнести и к классу  $L_1$  и к классу  $L_2$ .

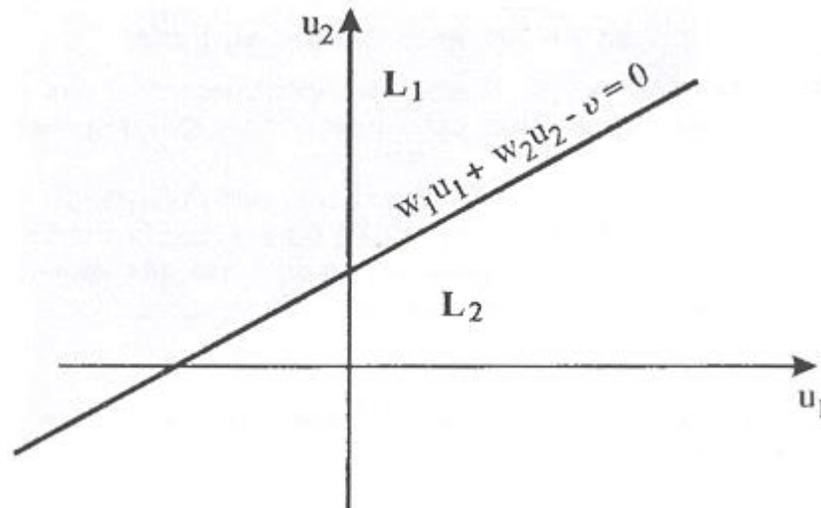


Рис. 3.6.2. Решающая граница для  $N = 2$ .

Если веса  $w_i$ ,  $i = 0, 1, \dots, N$  в уравнении гиперплоскости (2.9) неизвестны, тогда на вход перцептрона последовательно подаются так называемые обучающие сигналы  $u^{(n)}$ ,  $n = 1, 2, \dots$ , где  $u^{(n)} = [u_1^{(n)}, \dots, u_N^{(n)}]^T$ .

Неизвестные значения весов будут определяться в процессе обучения перцептрона. Роль «учителя» заключается в корректном отнесении сигналов  $u^{(n)}$  к классам  $L_1$  или  $L_2$ , несмотря на неизвестность весов уравнения решающей границы (2.9). По завершении процесса обучения перцептрон должен правильно классифицировать поступающие на его вход сигналы, в том числе и те, которые отсутствовали в обучающей последовательности  $u^{(n)}$ ,  $n = 1, 2, \dots$ . Множества векторов  $u^{(n)}$ ,  $n = 1, 2, \dots$ , для которых выходной перцептрона принимает соответственно значения 1 и  $-1$ , линейно отделены, т.е. лежат в двух различных полупространствах, разделенных гиперплоскостью

(2.9). Другими словами, допускается разделение обучающей последовательности  $\{u(n)\}$  на две последовательности  $\{u_1(n)\}$  и  $\{u_2(n)\}$  так, что  $\{u_1(n)\} \in L_1$  и  $\{u_2(n)\} \in L_2$ .

В  $n$ -й момент времени сигнал на выходе линейной части персептрона определяется выражением

$$x(n) = \sum_{i=0}^N w_i(n) u_i(n) = w^T(n) u(n), \quad (3.16)$$

где

$$u(n) = [-1, u_1(n), u_2(n), \dots, u_N(n)]^T,$$

$$w(n) = [v(n), w_1(n), w_2(n), \dots, w_N(n)]^T.$$

Выходной сигнал персептрона может быть описан выражением

$$y(n) = \operatorname{sgn}(w^T(n) u(n)). \quad (3.17)$$

С учетом введенных обозначений рекурсии (2.14) и (2.15) принимают вид

$$w(n+1) = w(n) + [d(n) - y(n)] u(n). \quad (3.18)$$

Разность  $d(n) - y(n)$  можно интерпретировать как погрешность между эталонным (заданным) сигналом  $d(n)$  и фактическим выходным сигналом  $y(n)$ .

По завершении обучения решающая граница персептрона определяется выражением

$$\sum_{i=0}^N w_i(n_0) u_i = 0, \quad (3.19)$$

а персептрон корректно классифицирует как сигналы, которые принадлежат к обучающей выборке  $\{u(n)\}$ .

Из рис. 3.6.2 следует, что не существует прямой, которая отделила бы точки со значениями функции исключающего ИЛИ (XOR), равными  $-1$ , от точек со значениями, равными  $1$ . В этом случае роль примерной границы играет эллипс, и поэтому алгоритм (2.18) не был бы сходящимся. Проблему XOR можно разрешить с помощью двухслойного персептрона.

### Многослойная нейронная сеть

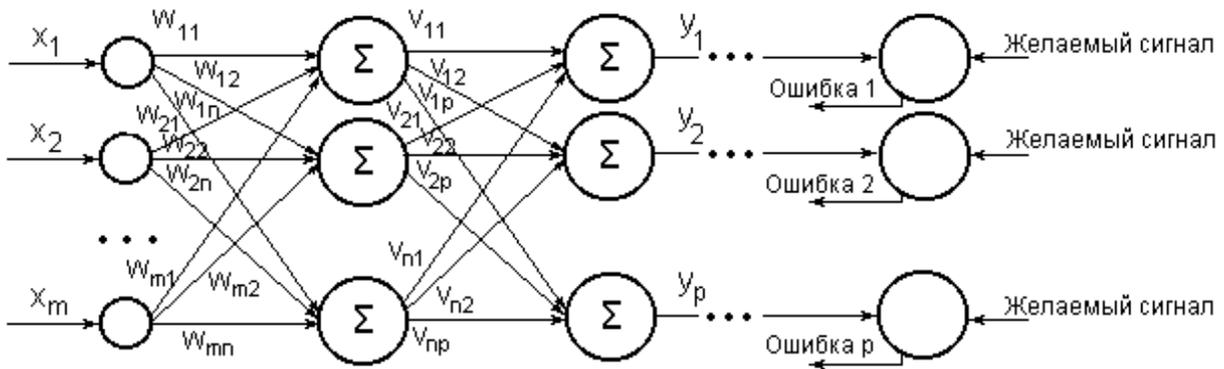


Рис. 3.6.3.. Многослойная нейронная сеть.

В каждом слое расположено  $N_k$  нейронов с нелинейной функцией (сигмоидой либо гиперболическим тангенсом). Нейронная сеть имеет  $m$  входов, на которые подаются входные сигналы  $x_1(n), \dots, x_m(n)$ .

Выходной сигнал  $i$ -го нейрона в  $r$ -м слое обозначается  $y_i(n)$ ,  $i=1 \dots p$ .

детальная структура  $i$ -го нейрона в  $r$ -м слое на рис 3.2.1.

Нейрон имеет  $N_k$  входов, причем  $x_j^{(k)}(n) = +1$  для  $i = 0$  и  $k = 1, \dots, L$ . Обратим внимание на факт, что входной сигнал нейрона  $AD_i^k$  связан с выходным сигналом  $(k-1)$  слоя следующим образом:

$$x_j^{(k)}(n) = \begin{cases} u_i(n) & \text{для } k = 1, \\ y_i^{(k-1)} & \text{для } k = 2, \dots, L, \\ +1 & \text{для } i = 0, k = 1, \dots, L. \end{cases} \quad (3.20)$$

На рис.3.5.3  $w_{ij}^{(k)}(n)$  обозначает вес связи  $i$ -го нейрона,  $i = 1, \dots, N_k$ , расположенного в  $k$ -м слое, которая соединяет этот нейрон с  $j$ -м входным сигналом  $x_j^{(k)}(n)$ ,  $j = 0, 1, \dots, N_{k-1}$ .

Вектор весов нейрона обозначим

$$w_i^{(k)}(n) = \left[ w_{i,0}^{(k)}(n), \dots, w_{i,N_{k-1}}^{(k)}(n) \right]^T, \quad k = 1, \dots, L, \quad i = 1, \dots, N_k. \quad (3.21)$$

Выходной сигнал нейрона в  $n$ -й момент времени,  $n = 1, 2, \dots$  определяется как

$$y_i^{(k)}(n) = f\left(s_i^{(k)}(n)\right), \quad (3.22)$$

причем

$$s_i^{(k)}(n) = \sum_{j=0}^{N_{k-1}} w_{ij}^{(k)}(n) x_j^{(k)}(n). \quad (3.23)$$

выходные сигналы нейронов в  $L$ -м слое

$$y_1^L(n), y_2^L(n), \dots, y_{N_L}^L(n) \quad (3.24)$$

одновременно являются выходными сигналами всей сети. Они сравниваются с так называемыми эталонными сигналами сети

$$d_1^L(n), d_2^L(n), \dots, d_{N_L}^L(n), \quad (3.25)$$

в результате чего получаем погрешность

$$\varepsilon_i^{(L)}(n) = d_i^{(L)}(n) - y_i^{(L)}(n), \quad i = 1, \dots, N_L. \quad (3.26)$$

Мера погрешности основана на сравнении сигналов (2.70) и (2.71), в виде суммы квадратов разностей, т.е.

$$Q(n) = \sum_{i=1}^{N_L} \varepsilon_i^{(L)2}(n) = \sum_{i=1}^{N_L} \left( d_i^{(L)}(n) - y_i^{(L)}(n) \right)^2. \quad (3.27)$$

Мера погрешности (3.2.7) - это функция от весов сети. Название алгоритма связано со способом расчета погрешностей в конкретных слоях. Вначале рассчитываются погрешности в последнем слое (на основе выходных и эталонных сигналов), далее - в предпоследнем и так вплоть до первого слоя. Начальные значения весов, образующих сеть, выбираются случайным образом и, как правило, устанавливаются близкими к нулю. Шаг коррекции  $\eta$  чаще всего принимает большие значения (близкие единице) на начальных этапах процесса обучения, но впоследствии его следует уменьшать по мере того как веса приближаются к некоторым заранее определенным значениям. Существуют различные модификации нейронных

### **Сети радиальных базисных функций (RBF-сети)**

Сети радиальных базисных функций (Radial Basis Functions, RBF-Netze, RBF-сети) представляют специальный тип нейронных сетей с прямыми связями. Основное их назначение – аппроксимация и интерполяция многомерных функций для решения, в частности, задач прогнозирования. Они были предложены в 1985 году Повелом (Powell, 1985). Их математическую основу составляет теория аппроксимации и интерполяции многомерных функций. Сколь угодно точная аппроксимация функций достигается при этом путем комбинации радиально симметричных функций.

RBF-сети обладают рядом характерных свойств:

1. Архитектура сетей с прямыми связями первого порядка (FF-сети): связи от нейронов одного слоя к нейронам следующего слоя;
2. Быстрое обучение;
3. Отсутствие «патологий» сходимости. В них в отличие от backpropagation-сетей не возникает проблемы локальных минимумов;
4. Длительное время их подготовки и настройки из-за необходимости выполнения более сложных расчетов;
5. RBF-сети - хорошие аппроксиматоры функций.

RBF-сети имеют только один слой скрытых нейронов. На рис.3.6.4. показана структура RBF-сети, содержащая один входной слой, один скрытый слой нейронов, число которых обычно соответствует числу элементов в обучающей последовательности, и один выходной слой из одного Единственный выходной нейрон выдает значение функции:

$$y = f(\mathbf{x}) = f(x_1, x_2, \dots, x_n). \quad (3.28)$$

Таким образом, по своему построению RBF-сети – это двухслойные сети прямого распространения первого порядка, причем оба слоя соединены весовой

матрицей. Входной вектор  $\mathbf{x}$  передается на нейроны скрытого слоя. При этом каждый нейрон скрытого слоя получает полную информацию о входном векторе  $\mathbf{x}$ .

Каждый нейрон  $i$  скрытого слоя рассчитывает значение одномерной функции  $h_i$  (например, «колокол» Гаусса):

$$h_i(\mathbf{x}) = \exp\left[-\frac{(\|\mathbf{x} - \mathbf{x}_i\|)^2}{2\sigma_i^2}\right] \quad (3.29)$$

где  $\mathbf{x}$  – входной вектор,  $\mathbf{x}_i$  –  $i$ -я опорная точка или  $i$ -й образ обучающей последовательности,  $\sigma_i$  – параметр рассеяния для одномерной функции  $h_i$ .

В качестве метрики  $\|\mathbf{x} - \mathbf{x}_i\|$  обычно используется эвклидово расстояние:

$$\|\mathbf{x} - \mathbf{y}\| = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + \dots + (x_n - y_n)^2} \quad (3.30)$$

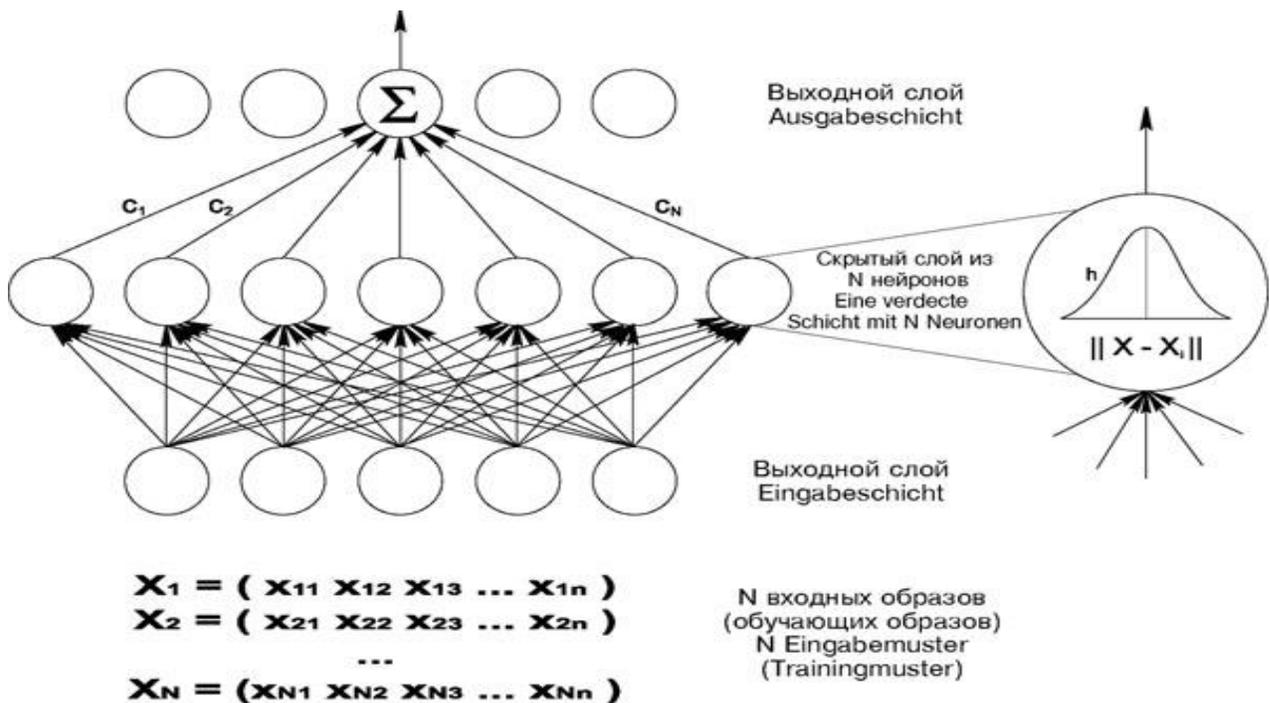


Рис. 3.6.4. Структура RBF-сети

Радиальные базисные функции  $h_i$  (3.6.4) выполняют предварительную обработку входных векторов, определяя их близость к центрам  $\mathbf{x}_i$ . Выходные

значения  $h_i(\mathbf{x})$  – это степени сродства между входным вектором  $\mathbf{x}$  и отдельными центрами  $\mathbf{x}_i$ . На основе значений  $h_i(\mathbf{x})$  определяется взвешенная сумма.

Центры  $\mathbf{x}_i$  определяются на основе обучающей последовательности и имеют ту же размерность  $n$ , что и входной вектор. Параметр  $\sigma_i$  определяется экспериментально. Рис. 3.6.5. иллюстрирует принцип работы нейрона скрытого слоя.

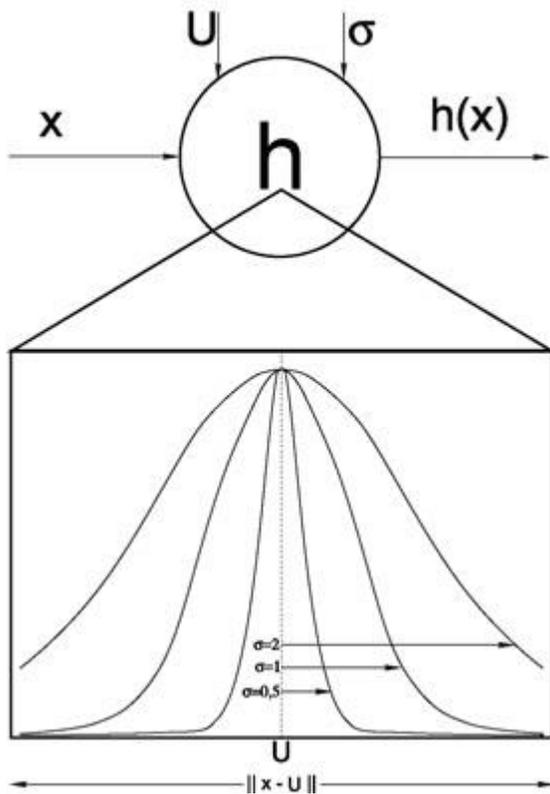


Рис. 3.6.5. Принцип работы нейрона скрытого слоя.

Из формулы (6.1) следует: выход нейрона  $i$  скрытого слоя тем больше, чем ближе текущий вход  $\mathbf{x}$  к центру  $\mathbf{x}_i$ . Выход  $f$  RBF-сети определяется в виде взвешенной суммы

$$f(\mathbf{x}) = y = \sum_{i=1}^k c_i h_i(\mathbf{x}) \quad (3.31)$$

Нейроны выходного слоя образуют, следовательно, линейную комбинацию выходов нейронов скрытого слоя. Их функция активации или выхода – это радиальная базисная функция (6.1).

Одномерные RBF-сети могут быть обобщены на многомерные функции. В этом случае в выходном слое размещаются  $m$  нейронов по числу аппроксимируемых (или интерполируемых) функций, причем значения на выходе  $j$ -го нейрона выходного слоя определяются следующим образом:

$$y_j = \frac{\sum_{i=1}^k c_{ij} h_i}{\sum_{i=1}^k h_i} \quad (3.32)$$

В RBF-сетях в качестве опорных точек в простейшем случае могут быть использованы образы обучающей последовательности. Преимущества RBF-сети имеют относительно других нейросетей:

1. Функция активации или выхода скрытого нейрона принимает большие значения лишь в тех случаях, когда входной образ находится вблизи от опорной точки этого нейрона. Для входов вне области, «покрытой» образами обучающей последовательности, сеть формирует лишь небольшие значения на своих выходах, причем для сетей с сигмоидальными функциями активации или выхода (например, в многослойных перцептронах) эти выходы могут быть вообще не предсказуемы.
2. Простая структура RBF-сетей, содержащих только один слой скрытых нейронов, делает возможным прямой (не итеративный) расчет весов сети. В этом – их большое преимущество по сравнению с другими типами

нейронных сетей, которые в большинстве случаев используют трудоемкие рекуррентные алгоритмы обучения.

3. Возможность инициализации RBF-сетей на основе прямого расчета весов с последующим их дообучением на основе алгоритмов обучения с поощрением (например, на основе алгоритма с обратным распространением ошибки – Backpropagation).

### **Рекуррентные и рециркуляционные нейронные сети**

В рециркуляционных сетях распространение информации осуществляется по двунаправленным связям, имеющим разные весовые коэффициенты в различных направлениях. Обучение рециркуляционных сетей осуществляется без учителя. В отличие от них рекуррентные нейронные сети характеризуются обучением с учителем и обратными связями, по которым передаются результаты обработки сетью данных на предыдущем этапе. Обучение таких сетей базируется на алгоритме обратного распространения ошибки, поэтому их относят к одному классу.

*Рекуррентные нейронные сети* применяются для решения задач прогнозирования и управления. В 1986 г. Джорданом предложена архитектура рекуррентной сети, в которой выходы элементов последнего слоя соединены с нейронами промежуточного слоя с помощью специальных входных нейронов, называемых контекстными. Количество контекстных нейронов равно числу выходных элементов рекуррентной сети. Функцией активации выходного слоя является линейная функция. Взвешенная сумма  $i$ -го элемента промежуточного слоя тогда

$$S_i(t) = \sum_{j=1}^n w_{ij} \cdot x_j(t) + \sum_{k=1}^p w_{ki} \cdot y_k(t-1) - S_{0i},$$

где  $w_{ji}$  – весовой коэффициент между  $j$ -ым нейроном входного слоя и  $i$ -ым нейроном промежуточного слоя,  $w_{ki}$  – вес между  $k$ -ым контекстным нейроном и

$i$ -ым нейроном промежуточного слоя,  $S_{0i}$  – смещение  $i$ -го нейрона промежуточного слоя,  $n$  – размерность входного слоя,  $p$  – количество нейронов выходного слоя. Выходное значение  $i$ -го нейрона скрытого слоя определяется тогда так:

$$y_i(t) = F(S_i(t)).$$

Нелинейное преобразование в сети осуществляют, как правило, логистическая функция или гиперболический тангенс.

Другой вариант рекуррентной сети предложен Элманом в 1990 г. Выходы нейронных элементов промежуточного слоя такой сети соединяются с контекстными нейронами входного слоя. Количество контекстных нейронных элементов равно числу нейронов промежуточного слоя. Применяются также рекуррентные сети с совмещением этих двух подходов. Алгоритм обучения рекуррентной сети включает следующие шаги:

1. Все контекстные нейроны устанавливаются в нулевое состояние при  $t=0$ .
2. Входной образ подается на сеть и осуществляется прямое распространение сигнала.
3. Модифицируются весовые коэффициенты и смещения в соответствии с алгоритмом обратного распространения ошибки.
4.  $t$  увеличивается на единицу, и если среднеквадратичная ошибка сети  $E > E_{\min}$ , то осуществляется возврат к шагу 2.

### 3.7. Сеть Хемминга

Когда нет необходимости, чтобы сеть в явном виде выдавала образ, то есть достаточно, например, получать его номер, ассоциативную память успешно реализует сеть Хэмминга. Данная сеть характеризуется меньшими затратами на

память и объемом вычислений, что становится очевидным из ее структуры (рис. 3.7.1).

Искусственная нейронная сеть Хемминга (рис.3.7.1) была предложена в 1987 г. Р. Липманом. Она представляет собой сеть с двумя обрабатывающими слоями: первый слой - слой Хемминга, второй слой - немного изменённая сеть Хопфилда.

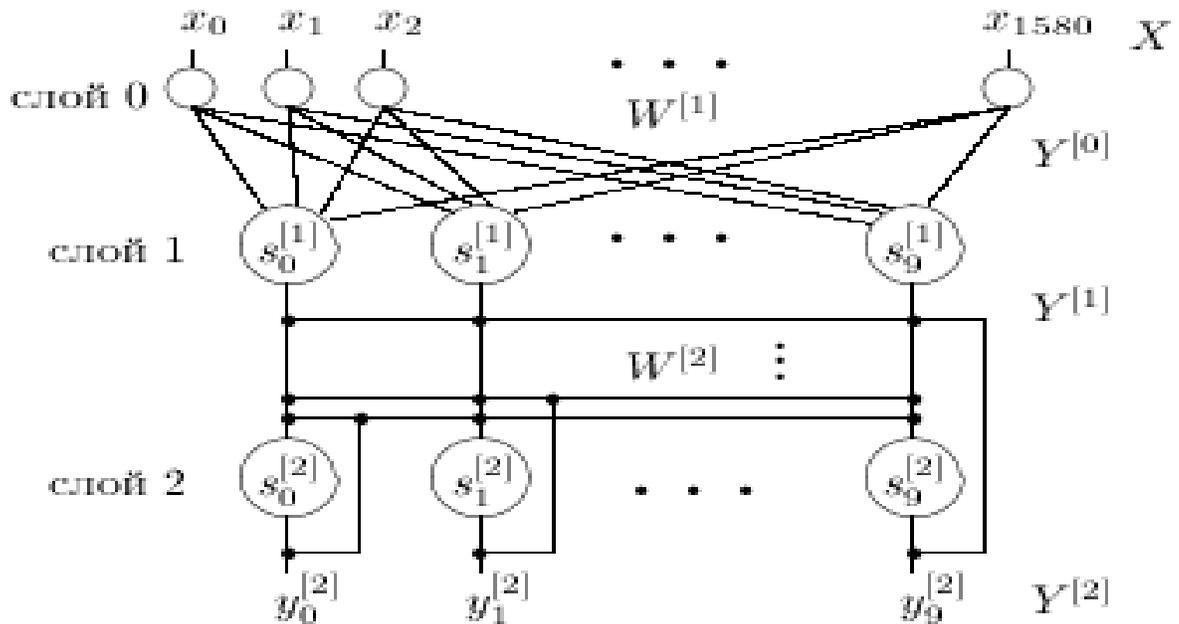


Рис. 3.7.1. Нейронная сеть Хемминга

Размерность скрытого слоя (слой 1 на рис.1) определяется количеством идеальных образцов, хранимых сетью. Размерность выходного слоя (слой 2 на рис.1) равна размерности скрытого слоя нейронной сети. На выходе получаем вектор  $y$  ( $y_0 \dots y_9$ ). Номер  $j$ , для которого  $y_j^{[2]} > 0$ , соответствует номеру класса входного образца.

Первый слой - слой Хемминга, представляет собой нейронную сеть с прямыми связями. Этот слой вычисляет меру подобия (расстояние Хемминга) между всеми эталонными образцами, хранящимися в сети, и текущим входом

сети. Расстояние Хемминга это количество отличающихся компонент во входном и эталонном векторах.

Для этого слоя используется линейная функция активации ( $y_i^{[1]} = s_i^{[1]}$ ). Состояние и выходное значение нейрона  $i$  слоя 1 определяется следующим выражением:

$$y_i^{[1]} = s_i^{[1]} = \sum_{j=0}^{1580} y_j^{[0]} \cdot w_{ij}^{[1]}, \quad (3.33)$$

где

- $y_i^{[1]}$  - выходное значение нейрона  $i$  слоя 1;
- $s_i^{[1]}$  - состояние нейрона  $i$  слоя 1;
- $y_j^{[0]}$  - вход нейронной сети номер  $j$ ;
- $w_{ij}^{[1]}$  - весовой коэффициент синаптической связи, соединяющей вход  $j$  нейронной сети и нейрон  $i$  слоя 1;

Второй слой используется для разрешения конфликтов, когда входной вектор оказывается похож по Хеммингу на более чем один идеал.

Начальное состояние нейронов слоя 2 определяется выходом первого слоя:

$$s_i^{[2]}(0) = y_i^{[1]}$$

Отношение переходов между состояниями нейронов слоя 2 определяется следующим образом:

$$s_i^{[2]}(t+1) = y_i^{[2]}(t) - c \cdot \sum_{j \neq i} y_j^{[2]}(t), \quad (3.34)$$

где

- $s_i^{[2]}(t)$  - состояние нейрона  $i$  слоя 2 в момент времени  $t$ ;
- $y_i^{[2]}(t)$  - выходное значение нейрона  $i$  слоя 2 в момент времени  $t$ ;
- $c$  - коэффициент торможения,  $0 < c < 0.1$ ;

отношение переходов (1) эквивалентно обычному определению состояния нейрона (сумма взвешенных входов) с постоянной матрицей весов(2).

$$w_{ij}^{[2]} = \begin{cases} 1 & , i = j \\ -c & , i \neq j \end{cases}$$

В качестве функции активации слоя 2 можно использовать следующую функцию :

$$y_i^{[2]} = f(s_i^{[2]}) = \begin{cases} s_i^{[2]} & , s_i^{[2]} > 0 \\ 0 & , s_i^{[2]} \leq 0 \end{cases} \quad (3.35)$$

Для нейронной сети Хемминга используется метод обучения "без учителя". Весовые коэффициенты слоя 1 определяются следующим образом:

$$w_{ji}^{[1]} = 0.5 * x_i^j$$

где

- $w_{ji}^{[1]}$  - весовой коэффициент синаптической связи, соединяющей вход  $i$  нейронной сети и нейрон  $j$  слоя 1;
- $x_i^j$  - компонента номер  $i$  идеального (учебного) вектора номер  $j$ ;

Весовые коэффициенты слоя 2 фиксированы отношением переходов

Алгоритм обучения нейронной сети сводится к начальной инициализации матрицы весов слоя 1 по (3.35).

1. На входы сети подается неизвестный вектор  $\mathbf{x} = \{x_i: i = 0, \dots, n-1\}$ , на основе которого рассчитываются состояния нейронов первого слоя (верхний индекс в скобках указывает номер слоя):

$$y_j^{(1)} = z_j^{(1)} = \sum_{i=0}^{n-1} w_{ij} x_i + T_j, j = 0, \dots, m-1$$

После этого полученными значениями инициализируются значения выходов нейронов второго слоя:

$$y_j^{(2)} = y_j^{(1)}, j = 0, \dots, m-1$$

2. Вычислить новые состояния нейронов второго слоя:

$$z_j^{(2)}(p+1) = y_j^{(2)}(p) - \varepsilon \sum_{k=0}^{m-1} y_k^{(2)}(p), k \neq j, j = 0, \dots, m-1$$

и значения их аксонов:

$$y_j^{(2)}(p+1) = S[z_j^{(2)}(p+1)], j = 0, \dots, m-1,$$

где  $S$  - функция активации или выхода.

3. Проверить, изменились ли выходы нейронов второго слоя за последнюю итерацию. Если да – перейди к шагу 2. Иначе-конец
4. Конец.

Из анализа алгоритма следует, что роль первого слоя нейронов условна: воспользовавшись один раз на шаге 1 значениями его весовых коэффициентов, сеть больше не обращается к нему, поэтому первый слой может быть вообще исключен из сети (заменен на матрицу весовых коэффициентов).

### 3.8. Сети Хопфилда

Американский исследователь Хопфилд в 80-х годах 20-го века предложил специальный тип нейросетей. Названные в его честь сети Хопфилда являются рекуррентными или сетями с обратными связями и были предназначены для

распознавания образов. Обобщенная структура этой сети представляется в виде системы с обратной связью выхода со входом. Сеть Хопфилда является однослойной.

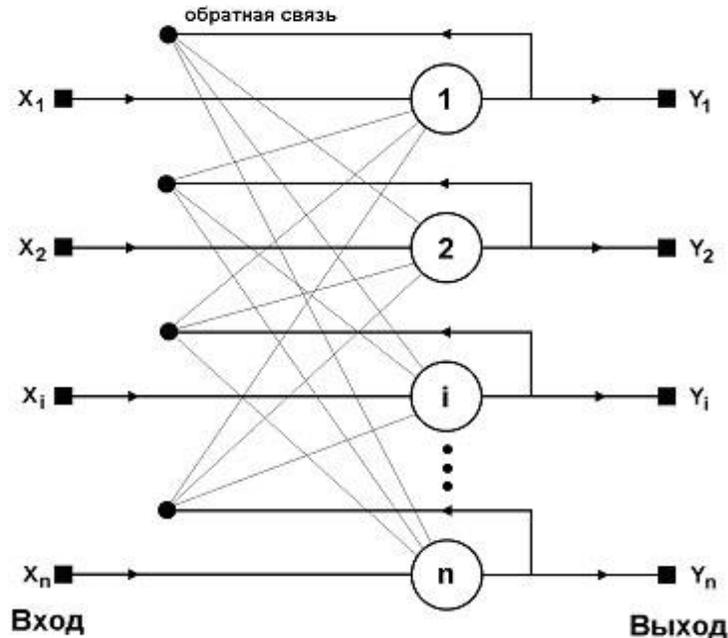


Рис.3.8.1. Сеть Хофилда

Характерная особенность такой сети состоит в том, что входные сигналы нейронов являются одновременно входными сигналами сети:  $x_i(k)=y_i(k-1)$ , при этом возбуждающий вектор особо не выделяется. В классической системе Хопфилда отсутствует связь нейрона с собственным выходом, что соответствует  $w_{ij}=0$ , а матрица весов является симметричной:  $w_{ij}=w_{ji}$

$$\hat{W} = \hat{W}^T.$$

Симметричность матрицы весов гарантирует сходимость процесса обучения. Процесс обучения сети формирует зоны притяжения некоторых точек равновесия, соответствующих обучающим данным. При использовании ассоциативной памяти имеем дело с обучающим вектором  $\bar{x}$ , либо с

множеством этих векторов, которые в результате проводимого обучения определяют расположение конкретных точек притяжения (аттракторов)

Обозначим через  $\mathbf{x}_s$  вектор значений признаков образа  $s$ -го класса, а через  $x_{si}$  – его  $i$ -ю составляющую. При этом алгоритм Хопфилда может быть описан следующим образом:

1. расчет весов связей между нейронами

$$w_{ij} = \begin{cases} \sum_{s=0}^{k-1} x_{si} x_{sj}, & i \neq j \\ 0 & , i = j \end{cases}$$

2. где  $k$  – число запоминаемых классов образов;

3. инициализация сети путем ввода:

$$y_i(0) = x_i, \quad 1 \leq i \leq n,$$

4. где  $y_i(0)$  – выход (от output)  $i$ -го нейрона в начальный (нулевой) момент времени;

5. итерационное правило:

Repeat

$$o_j(t+1) = S \left( \sum_{i=0}^{n-1} w_{ij} o_i(t) \right)$$

6. Until  $o_j(t+1) = o_j(t), \quad j = 1, 2, \dots, n$

где  $o_i(t)$  – выход  $i$ -го нейрона в момент времени  $t$ .

Соотношения (5.4), (5.5) представим в векторной форме:

$$\mathbf{o}(t+1) = \mathbf{S}(\mathbf{W}\mathbf{o}(t)), \quad t = 1, 2, \dots$$

$$\mathbf{o}(0) = \mathbf{x}(0),$$

где  $\mathbf{x}$  – входной вектор,  $\mathbf{W}$  – симметричная матрица сети Хопфилда,  $\mathbf{S}$  – вектор-функция активации или выхода нейронов.

Взвешенная сумма входов  $j$  – го нейрона сети Хопфилда равна:

$$z_j = net_j = \sum_i w_{ij} o_i$$

Функция активации или выхода имеет вид:

$$o_j = f(z_j) = \begin{cases} 1 & \text{для } z_j > 0 \\ -1 & \text{для } z_j \leq 0 \end{cases}$$

Таким образом, алгоритм Хопфилда может быть сформулирован следующим образом. Пусть имеется некоторый образ, который следует запомнить в сети Хопфилда, тогда веса искомой сети для распознавания этого образа могут быть рассчитаны, т.е. процесс обучения исключается. Если этот образ характеризуется  $n$  – мерным вектором  $\mathbf{x} = (x_1, \dots, x_n)'$ , то веса соединений определяются по формулам:

$$w_{ij} = \begin{cases} x_i x_j & \text{для } i \neq j \\ 0 & \text{для } i = j \end{cases}$$

то есть вес связи  $i$ -го нейрона с  $j$ -м равен произведению  $i$ -й и  $j$ -й составляющих вектора  $\mathbf{x}$ , характеризующего данный образ. В этом случае сеть Хопфилда, характеризуемая матрицей  $\mathbf{W}$  и порогами  $T_i = 0$ ,  $i = 1, 2, \dots, n$ , запоминает предъявленный образ. Каждый нейрон имеет функцию активации сигнум со значениями  $\pm 1$ :

$$\text{sgn } a = \begin{cases} 1, a \geq 0 \\ -1, a < 0 \end{cases}$$

Это означает, что выходной сигнал  $i$ -го нейрона определяется функцией

$$y_i = \operatorname{sgn}\left(\sum_{j=1}^N w_{ij} x_j + b_i\right),$$

где  $N$  обозначает количество нейронов,  $N=n$ . Постоянная составляющая  $b_i$ , определяющая порог срабатывания отдельных нейронов, равна 0. Тогда циклическое прохождение сигнала в сети Хопфилда можно представить соотношением

$$y_i(k) = \operatorname{sgn}\left(\sum_{j=1}^N w_{ij} y_j(k-1)\right) \quad (3.38)$$

с начальным условием  $y_j(0) = x_j$ .

В процессе функционирования сети Хопфилда можно выделить два режима: обучения и классификации. В режиме обучения на основе известных обучающих выборок  $\mathbf{x}_i$  подбираются весовые коэффициенты  $w_{ij}$ .

В режиме классификации при зафиксированных значениях весов и вводе конкретного начального состояния нейронов  $\mathbf{y}(0) = \mathbf{x}$  возникает переходный процесс, протекающий в соответствии с выражением (2) и заканчивающийся в одном из локальных минимумов, для которого  $\mathbf{y}(x) = \mathbf{y}(x-1)$

Обучение не носит рекуррентного характера. Поэтому вводят значения (правило Хебба)

$$w_{ij} = \frac{1}{N} x_i x_j,$$

Поскольку

$$\sum w_{ij} x_j = \frac{1}{N} \sum x_i (x_j x_j) = x_i \sum_N 1 \cdot \frac{1}{N} = x_i, \quad (3.39)$$

Так как вследствие биполярности значений элементов вектора  $\mathbf{x}$  всегда  $x_j^2 = 1$ .

При вводе большого количества обучающих выборок  $\mathbf{x}^{(k)}$  для  $k=1,2,\dots,p$  веса  $w_{ij}$  подбираются согласно обобщенному правилу Хебба, в соответствии с которым

$$w_{ij} = \frac{1}{N} \sum_{k=1}^l x_i^{(k)} x_j^{(k)}.$$

Благодаря такому режиму обучения веса принимают значения, определяемые усреднением множества обучаемых выборок.

Сеть Хопфилда может быть представлена в терминах «энергии» сети. В такой интерпретации запомненные образы соответствуют локальным минимумам энергетической поверхности.

Если преобразования в сети будут вблизи соответствующего минимума, то сеть остановится в локальном минимуме, соответствующем ближайшему образу в обучающей выборке. Для сети Хопфилда «энергия» взаимодействия пары узлов

$$e_{ij} = -w_{ij}x_i x_j.$$

Энергия сети может быть найдена как сумма по всем парам узлов:

$$E = \sum_{\substack{\text{pairs} \\ i \neq j}} e_{ij} = -\frac{1}{2} \sum_{i,j} w_{ij} x_i x_j. \quad (3.39)$$

Коэффициент  $\frac{1}{2}$  учитывает, что каждая пара присутствует в сумме дважды, как  $w_{ij}x_i x_j$  и как  $w_{ji}x_j x_i$ , а также, что  $w_{ij}=w_{ji}$ ,  $w_{ii}=0$ . Для определения изменения энергии при изменении значения входа  $k$ -го узла выделим в полной энергии  $E$  соответствующее слагаемое:

$$E = -\frac{1}{2} \sum_{\substack{i \neq k \\ j \neq k}} w_{ij} x_i x_j - \frac{1}{2} \sum_i w_{ki} x_k x_i - \frac{1}{2} \sum_{j \neq k} w_{jk} x_j x_k = S_{const} - x_k \sum w_{ki} x_i$$

иначе

$$E = S - x_k a^k,$$

где  $a^k = \sum w_{kj} x_j$  - аргумент функции активации.

Если  $x_k' - x_k = \Delta x_k$  - процесс итерации

1.  $a^k \geq 0$ . Тогда выходы могут принимать значение 1. В этом случае  $\Delta x_k \geq 0$ .
2. Следовательно  $\Delta x_k a^k \geq 0$  и  $\Delta E \leq 0$ .
3.  $a^k < 0$ . Тогда выход принимает значение  $-1$ . В этом случае  $\Delta x_k \leq 0$  и снова  $\Delta x_k a^k \geq 0$  и  $\Delta E \leq 0$ .

### 3.9. Сеть Кохонена

Сети Кохонена (Kohonen-Netze, self-organizing maps = SOM, Kohonen maps, Kohonen feature maps) или самоорганизующиеся карты предназначены для решения задач автоматической классификации, когда обучающая последовательность образов отсутствует. Соответственно не нужно и определение ошибки классификации, на минимизации которой основаны

алгоритмы обучения (например, алгоритм обучения с обратным распространением ошибки - Backpropagation).

Сеть Кохонена - это двухслойная нейросеть, содержащая входной слой (слой входных нейронов) и слой Кохонена (слой активных нейронов). Слой Кохонена может быть: одномерным, двумерным, или трехмерным. В первом случае активные нейроны расположены в цепочку. Во втором случае они образуют двумерную сетку (обычно в форме квадрата или прямоугольника), а в третьем случае они образуют трехмерную конструкцию.

В силу отсутствия обучающей последовательности образов, для каждого из которых известна от учителя принадлежность к тому или иному классу, определение весов нейронов слоя Кохонена (Referenz - Codebook Vektoren) основано на использовании алгоритмов автоматической классификации (кластеризации или самообучения).

Сеть представляет топологическую карту Кохонена, содержащей входной слой из  $n$  входных нейронов и слой Кохонена из  $m$  выходных нейронов, расположенных в виде прямоугольника (рис.3.9.1). Нейроны входного слоя служат для ввода значений признаков распознаваемых образов. Активные нейроны слоя Кохонена предназначены для формирования областей (кластеров) различных классов образов, причем число этих кластеров априори не известно. На этом рисунке показаны связи всех входных нейронов лишь с одним нейроном слоя Кохонена. Каждый нейрон слоя Кохонена соединен также с соседними нейронами. Введем следующие обозначения:

$$\mathbf{w}_j = (w_{j1}, w_{j2}, \dots, w_{jn})^T, j=1, 2, \dots, m \quad (3.40)$$

- вектор весовых коэффициентов  $j$ -го нейрона слоя Кохонена,

$$\mathbf{x} = (x_1, x_2, \dots, x_n)^T, \quad (3.41)$$

- входной вектор или вектор значений признаков входного образа.

На стадии обучения (точнее самообучения) сети входной вектор  $\mathbf{x}^c$  попарно сравнивается со всеми векторами  $\mathbf{w}_j$  всех нейронов слоя Кохонена. Вводится некоторая функция близости (например, в виде эвклидова расстояния). Активный нейрон с номером  $c$  слоя Кохонена, для которого значение функции близости  $d(\mathbf{x}, \mathbf{w}_c)$  между входным вектором  $\mathbf{x}$ , характеризующим некоторый образ, и вектором  $\mathbf{w}_c$  максимально, объявляется "победителем". При этом образ, характеризующийся вектором  $\mathbf{x}$ , относится к классу, который представляется нейроном-"победителем". В результате осуществляется преобразование  $n$ -мерного входного пространства  $\mathbf{R}^n$  на  $m$ -мерную сетку (слой Кохонена).

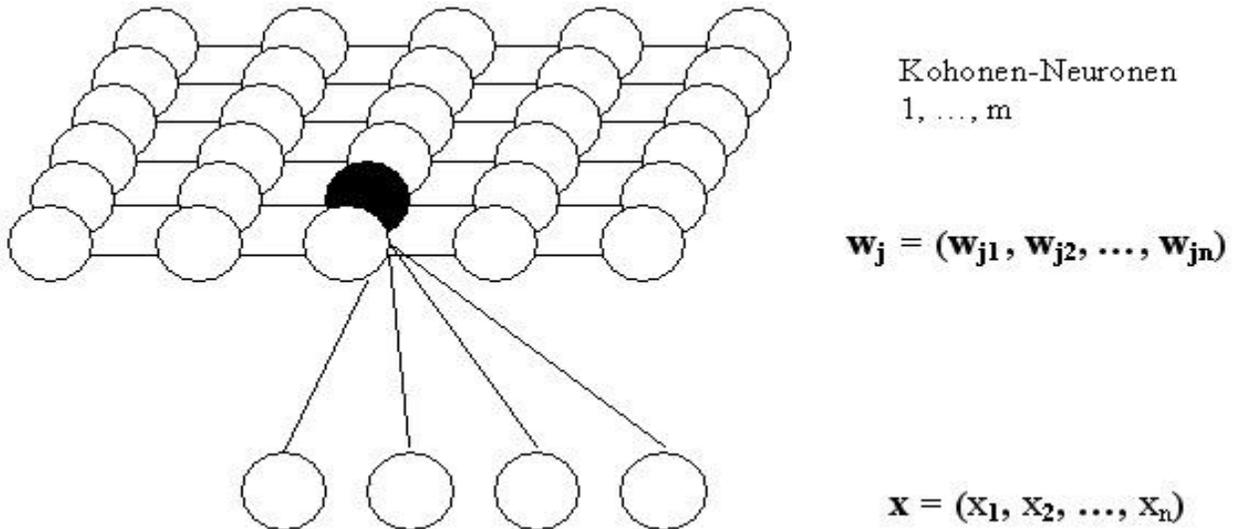


Рис. 3.9.1. Топологическая карта сети Кохонена с  $n$  входными и  $m$  выходными нейронами.

Отображение реализуется в результате рекуррентной (итеративной) процедуры самообучения. Отличительная особенность этого отображения - формирование кластеров или классов. По завершении процесса самообучения на стадии реального использования сети Кохонена неизвестные входные образы относятся к одному из выявленных кластеров (классов).

алгоритм самообучения сетей Кохонена с  $n$  нейронов во входном слое и  $m$  активных нейронов в слое Кохонена.

Для определения расстояния между входным вектором  $x$  (4.2) и весовым вектором  $w_j$  (4.1)  $j$ -го нейрона слоя Кохонена можно использовать различные функции близости (обычно евклидово расстояние). При этом "выигрывает" нейрон с весовым вектором  $w_c$ , наиболее близким к входному вектору  $x$ :

$$\|x - w_c\| = \min_j \|x - w_j\|$$

или при использовании функции  $\text{index}$ , определяющей номер минимального расстояния:

$$c = \text{index} \min_j \|x - w_j\|$$

При использовании скалярного произведения для нормированных векторов (т.е. векторов единичной длины)

$$x'w_j = \sum_{i=1}^n x_i w_{ij} = \text{net}_j = z_j$$

"выигрывает" нейрон с максимальным значением этого скалярного произведения.

На стадии самообучения сети Кохонена осуществляется коррекция весового вектора не только нейрона-"победителя", но и весовых векторов остальных активных нейронов слоя Кохонена, однако естественно в существенно меньшей степени - в зависимости от удаления от нейрона-"победителя". При этом форма и величина окрестности вокруг нейрона-"победителя", весовые коэффициенты нейронов которой также корректируются, в процессе обучения изменяются. Сначала начинают с очень большой области - она, в частности, может включать все нейроны слоя Кохонена. Изменение весовых векторов осуществляется по правилу:

$$\mathbf{w}_j(t+1) = \mathbf{w}_j(t) + \eta(t) d_{cj}(t) [\mathbf{x}(t) - \mathbf{w}_j(t)], j=1,2,\dots,m \quad (3.42)$$

где  $\mathbf{w}_j(t)$  - значение весового вектора на  $t$ -м шаге самообучения сети,  $d_{cj}(t)$  - функция близости между нейронами слоя Кохонена и  $\eta(t)$  - изменяемый во времени коэффициент коррекции. В качестве  $\eta(t)$  обычно выбирается монотонно уменьшающаяся функция ( $0 < \eta(t) < 1$ ), т. е. алгоритм самообучения начинается сравнительно большими шагами адаптации и заканчивается относительно небольшими изменениями.

алгоритм Кохонена

**Инициализация.** Выбираются (случайным образом) начальные значения всех  $n$ -мерных весовых векторов слоя Кохонена, а также стартовые значения коэффициента коррекции  $\eta$  и радиуса близости  $d$ .

**Шаг 1.** Выбирается некоторый образ, характеризующийся вектором значений признаков  $\mathbf{x}_j$ .

**Шаг 2.** Определяется нейрон-“победитель” с номером  $c$  на основе (4.4) или (4.5).

**Шаг 3.** Для нейрона-“победителя” с номером  $c$  и для нейронов из радиуса близости вокруг него определяются новые значения весовых векторов на основе (4.6).

**Шаг 4.** Осуществляется модификация коэффициента коррекции  $\eta$  и радиуса близости  $d$ .

**Шаг 5.** Выбирается критерий сходимости, согласно ему стоп или переход к шагу 1.

Обратим внимание, что в соответствии с (4.6) изменение того или иного весового вектора  $\mathbf{w}_j$  осуществляется в направлении входного вектора и пропорционально расстоянию между входным вектором  $\mathbf{x}$  и этим весовым вектором  $\mathbf{w}_j$ . В идеальном случае при использовании этой процедуры весовые векторы нейронов слоя Кохонена равномерно распределяются в пространстве входных признаков.

### 3.10. ART-сети

Проблема стабильности-пластичности является одной из самых сложных и трудно решаемых задач при построении искусственных систем, моделирующих восприятие. Характер восприятия внешнего мира живыми организмами (и, прежде всего, человеком) постоянно связан с решением дилеммы, является ли некоторый образ "новой" информацией с сохранением этого образа в памяти, либо этот образ является вариантом "старой", и в этом случае реакция системы должна учитывать ранее накопленный опыт. Специальное запоминание этого образа в последнем случае не требуется. Таким образом, восприятие одновременно пластично, адаптировано к новой информации, и при этом оно стабильно, то есть не разрушает память о старых образах.

Многие нейронные системы не приспособлены к решению этой задачи. Многослойный персептрон, обучающийся по методу обратного распространения, запоминает весь пакет обучающей информации, при этом образы обучающей выборки предъявляются в процессе обучения многократно. Попытки затем обучить персептрон новому образу приведут к модификации синаптических связей с, разрушением памяти о предыдущих образах. Таким образом, персептрон не способен к запоминанию новой информации, необходимо полное переобучение сети.

Аналогичная ситуация имеет место и в сетях Кохонена и Хемминга, обучающихся на основе самоорганизации. Данные сети всегда выдают положительный результат при классификации. Тем самым, эти нейронные сети не в состоянии отделить новые образы от искаженных или зашумленных старых образов.

Нейронные сети адаптивной резонансной теории (Adaptive Resonance Theory = ART) или ART-сети образуют целый класс различных нейросетей,

предложенных Карпентером (Carpenter) и Гроссбергом (Grossberg) (Бостонский университет, 1987-1991).

Два противоречивых требования возникают. С одной стороны сеть должна выявлять (обнаруживать) образы новых классов, ранее не представленных сети (свойство пластичности). С другой же стороны изученные классы образов должны сохраняться (свойство устойчивости). Эти два свойства – пластичности и стабильности в известной мере противоречивы. Сети ART разработаны для установления новых образов (классов) нейронной сетью, без забывания старых образов (классов). Семейство ART-сетей включает следующие сети:

1. ART-1: для бинарных входных векторов, когда признаки распознаваемых образов принимают два значения 1 или 0;
2. ART-2: расширение ART-1-сетей на непрерывные входные векторы;
3. ART-2а: оптимальная версия ART-2-сетей с повышенной скоростью сходимости;
4. ART-3: моделирование временных и химических процессов на базе ART-2;
5. ARTMAP: комбинация двух ART-сетей (например, ART-1 и ART-2);

Принцип работы ART-сетей. При вводе значений образа ART-1-сеть сравнивает его с классом из числа уже изученных. Если такой класс находится, то выполняется небольшая модификация прототипа этого класса для близкого отображения к новому образу. Классификация образа на этом заканчивается.

Если такой класс не находится, то вводится новый класс. При этом предъявленный образ несколько модифицируется и используется затем в качестве прототипа для нового класса. При этом уже изученные классы не изменяются. На рис. 3.10.1 показаны основные компоненты ART-1-сетей:



правила:  $i$ -я компонента вектора  $\mathbf{S}$  принимает значение единица, если по крайней мере две из трех следующих переменных приняли значение 1:

1. коэффициент усиления  $g_1$  (для всех нейронов одинаков);
2.  $i$ -я компонента  $I_i$  входного вектора  $\mathbf{I}$ ;
3.  $i$ -я компонента  $v_i$  ожидаемого вектора  $\mathbf{V}$  (взвешенная сумма выходов слоя распознавания).

На рис. 3.10.2 представлена часть ART-1-сети с четырьмя нейронами в слое сравнения  $F_1$  и тремя нейронами слоя распознавания  $F_2$ .

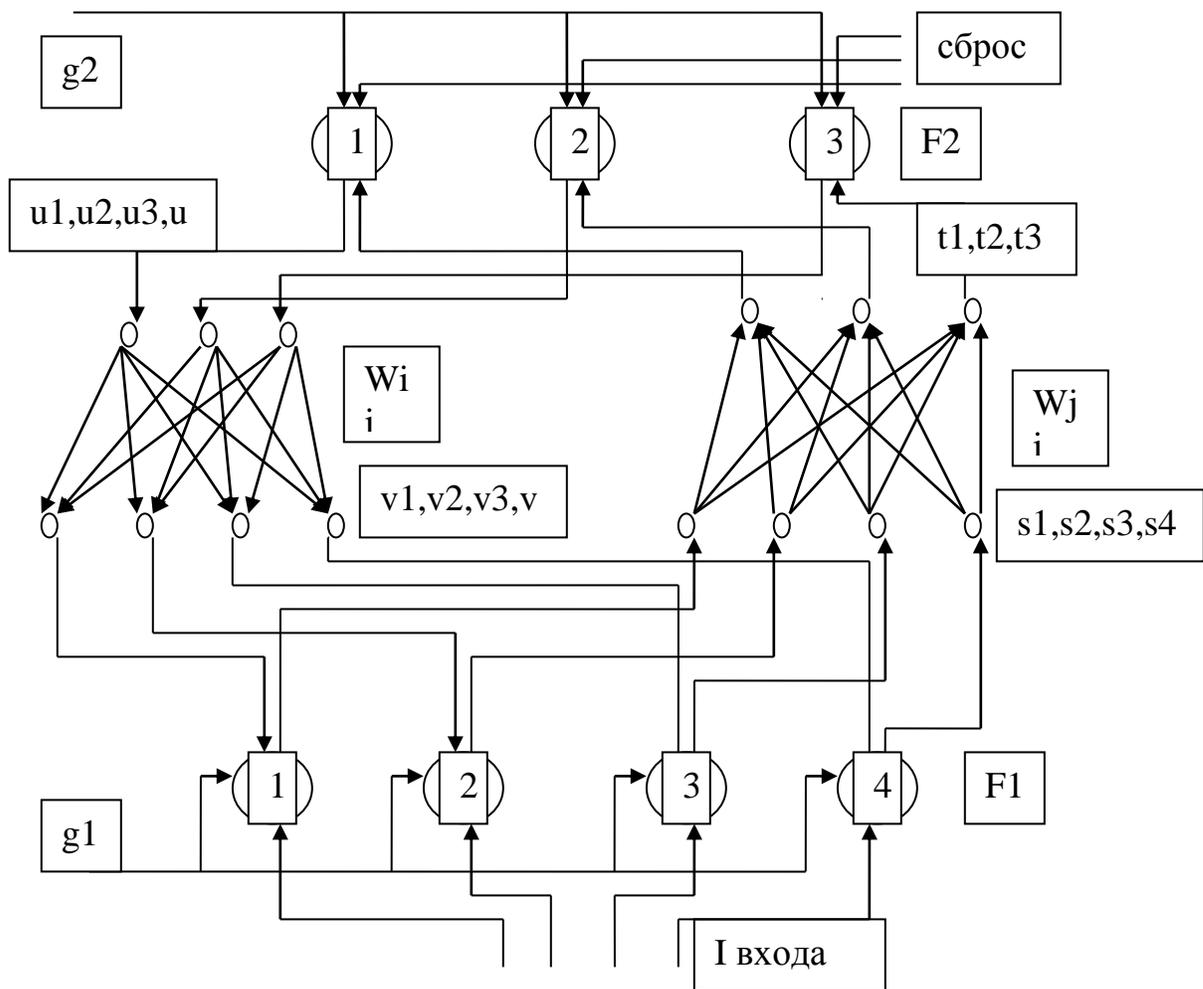


Рис. 3.10.2. ART-1-сеть четырьмя нейронами в слое сравнения  $F_1$  и тремя нейронами слоя распознавания  $F_2$ .

### **Слой распознавания**

Он сопоставляет каждому входному вектору соответствующий класс. Если для входного вектора не удастся найти достаточно близкий класс из числа уже выявленных, то открывается (образуется) новый класс.

Класс, представляемый  $j$ -м нейроном слоя распознавания и наиболее близкий ко входному вектору  $I$  или вектору  $S$ , определяется следующим образом:

$$t_{j \max} = \max_k \left\{ t_k = S'W_k = \sum_{i=1}^m s_i w_{ik} \right\} \quad (3.43)$$

где  $S'W_k$  – скалярное произведение векторов  $S$  и  $W_k$ . При этом работает нейрон  $j$  слоя распознавания, действительный весовой вектор  $W_j = (w_{1j}, w_{2j}, \dots, w_{mj})$  bottom-up-матрицы  $W_{ij}$  которого имеет наибольшее сходство с вектором  $S$ . Этот  $j$ -й нейрон объявляется “победителем” и предъявленный образ относится к  $j$ -му классу. Компоненты вектора  $U$  на выходе слоя распознавания определяются при этом следующим образом:

$$u_j = \begin{cases} 1, & \text{если } t_j = S'W_j = \max \\ 0, & \text{в противном случае} \end{cases}$$

то есть  $u_j = 1$ , если скалярное произведение весового вектора  $W_j = (w_{1j}, w_{2j}, \dots, w_{mj})$  и вектора  $S = (s_1, s_2, \dots, s_m)$  максимально.

### **Весовые матрицы**

В ART-1-сетях используются две весовые матрицы:

1. Действительная матрица  $W_{ij}$ . Она служит для расчета степени сходства в фазе распознавания;
2. Бинарная матрица  $W_{ji}$ , предназначенная для перепроверки степени корректности классификации входного образа с помощью матрицы  $W_{ij}$ .

Ключи g1, g1 и сброс

ART-1-сетях используются два ключа  $g_1$  и  $g_2$ , используемые для синхронизации работы нейросети.  $g_1$  принимает значение 1, если по меньшей мере одна составляющая входа имеет значение 1 и одновременно ни один нейрон слоя распознавания не находится в состоянии 1:

$$g_1 = (I_1 \vee I_2 \vee \dots \vee I_m) \wedge \neg(u_1 \vee u_2 \vee \dots \vee u_m)$$

$g_2$  реализует логическое ИЛИ для входного вектора.  $g_2 = 1$ , если по меньшей мере одна составляющая входного вектора равна 1:

$$g_2 = (I_1 \vee I_2 \vee \dots \vee I_m)$$

Для актуального нейрона («победителя») слоя распознавания сброс составляющая равна 1 (то есть функция сброса активируется), если различие входного вектора  $\mathbf{I}$  и вектора  $\mathbf{S}$  превышает некоторый порог (параметр толерантности).

В ART-1-сетях можно выделить следующие пять фаз внутренней обработки информации:

1. Инициализация сети: в начале инициализируются обе весовые матрицы  $W_{ij}$  и  $W_{ji}$ ;
2. Распознавание: на этой фазе для входного вектора  $\mathbf{I}$  или для вектора  $\mathbf{S}$ , определенного на основе вектора  $\mathbf{I}$ , определяется наиболее близкий класс;
3. сравнение: в фазе сравнения ожидаемый вектор (прототип)  $\mathbf{V}$  класса образов, к которому отнесен входной образ, сравнивается со входным вектором  $\mathbf{I}$  или вектором  $\mathbf{S}$  (проверка корректности классификации). При слишком малом совпадении векторов  $\mathbf{V}$  и  $\mathbf{S}$  осуществляется повторное распознавание;
4. поиск: на этой стадии производится поиск альтернативного класса образов или же при необходимости открывается новый класс;

5. адаптация весов: на этой стадии осуществляется модификация обеих весовых матриц.

Инициализация. Пусть  $i = 1, 2, \dots, m$  – индекс нейронов слоя сравнения  $F_1$ , а  $j = 1, 2, \dots, n$  – индекс нейронов слоя распознавания  $F_2$ . Веса  $w_{ij}$  матрицы  $\mathbf{W}_{ij}$  сначала устанавливаются небольшими в соответствии с неравенством

$$w_{ij} < \frac{L}{L - 1 + m} \quad (3.44)$$

для всех нейронов  $i$  слоя сравнения и нейронов  $j$  слоя распознавания, где  $m$  – число составляющих входного вектора  $\mathbf{I}$ ,  $L$  – постоянная ( $L > 1$ , типичное значение  $L = 2$ ). При выборе весов  $w_{ij}$  слишком большими все входные вектора отображаются на один нейрон слоя распознавания.

Все веса бинарной матрицы  $\mathbf{W}_{ji}$  вначале устанавливаются единицами:  $w_{ji} = 1$  для всех  $j$  слоя распознавания и всех  $i$  слоя сравнения.

Параметр сходства  $p$  выбирается между 0 и 1 ( $p$  принадлежит  $[0; 1]$ ) – в зависимости от желаемой степени совпадения. Небольшое значение  $p$  (близкое к 0) «не обращает внимания» на большие различия образов внутри одного класса. При выборе  $p = 1$  требуется абсолютное (точное) совпадение между входом и прототипом (стереотипом) класса образов.

Распознавание. В начале процесса обучения (точнее самообучения) сети входной вектор нулевой:  $\mathbf{I} = (0, 0, \dots, 0)$ . Соответственно коэффициент усиления  $g_2 = 0$ . В результате все нейроны слоя распознавания отключены. Следствием этого является нулевой вектор  $\mathbf{V}$ .

При подаче на вход сети ненулевого входного вектора  $\mathbf{I}$  коэффициенты усиления (ключи) принимают значения:  $g_1 = 1$  и  $g_2 = 1$ . Это приводит к срабатыванию тех нейронов слоя сравнения  $F_1$ , входы которых  $I_i = 1$ . В результате формируется вектор  $\mathbf{S}$ , который сначала является точной копией вектора  $\mathbf{I}$ . Затем для каждого  $j$  слоя распознавания  $F_2$  вычисляется скалярное

произведение весового вектора  $\mathbf{W}_j$  и вектора  $\mathbf{S}$ :  $t_j = \text{net}_j = \mathbf{W}_j' \mathbf{S}$ . Скалярное произведение является мерой сходства между векторами  $\mathbf{W}_j$  и  $\mathbf{S}$ . Нейрон  $j$  – «победитель» – с максимальным значением скалярного произведения «выигрывает» сравнение («соревнование») и возбуждается (срабатывает), все же остальные нейроны не срабатывают.

Из (3.43) следует, что при этом лишь  $j$ -я составляющая вектора  $\mathbf{U}$  (вход для матрицы) принимает значение 1. Все же остальные составляющие - нули.

*Фаза сравнения.* Единственный нейрон  $J$  – «победитель» слоя распознавания, весовой вектор которого наиболее близок к входному вектору, выдает единицу 1. Эта единица распространяется через бинарные веса  $w_{ji}$  матрицы. Остальные же нейроны слоя распознавания выдают нули. В результате каждый нейрон  $i$  слоя сравнения получает бинарный сигнал  $v_i$ , равный значению  $w_{ji}$  (1 или 0)

$$v_i = \sum_{j \in \text{Recog}} u_j w_{ji} = w_{ji} \quad (3.45)$$

Так как входной вектор  $\mathbf{I}$  не нулевой и  $J$ -й нейрон слоя распознавания находится в состоянии 1, то из (3.4) следует  $g_1 = 0$ , ибо только одна компонента в фазе распознавания установлена в 1.

После этого рассчитывается новое значение для вектора  $\mathbf{S}$ . Этот расчет сводится по существу к покомпонентному сравнению векторов  $\mathbf{V}$  и  $\mathbf{I}$ . Если векторы  $\mathbf{V}$  и  $\mathbf{I}$  различаются в некоторой составляющей, то соответствующая составляющая нового вектора  $\mathbf{S}$  полагается равной 0, т.е.  $\mathbf{S} = \mathbf{V} \wedge \mathbf{I} = \mathbf{W}_J \wedge \mathbf{I}$ .

Вывод: если предшествующее значение вектора  $\mathbf{S}$  и вектор  $\mathbf{I}$  отличаются существенно, то новый вектор  $\mathbf{S}$  содержит много нулевых составляющих, хотя соответствующие составляющие входного вектора  $\mathbf{I}$  являются единицами. Это означает, что наиболее правдоподобный класс образов не совпадает со входом. При этом сброс-компонента нейросети, отвечающая за сравнение

векторов  $\mathbf{I}$  и  $\mathbf{S}$ , вырабатывает сигнал сброса. Этим сигналом нейрон, объявленный ранее “победителем”, устанавливается в нулевое состояние.

Функция сброс запускается при невыполнении неравенства, а именно при  $|\mathbf{S}|/|\mathbf{I}| < p$ . Значение параметра сходства  $p$  выбирается между 0 и 1: значение  $p = 1$  требует полного совпадения, а при  $p = 0$  совершенно различные образы отображаются в один класс.

Или иначе; при выборе большого значения  $p$  образуется больше классов образов; при выборе же небольших значений  $p$  входные образы разделяются на меньшее число классов (кластеров). Обычно значение  $p$  выбирается между 0.7 и 0.95 :  $p \in [0.7; 0.95]$

Если степень сходства выше установленной степени сходства  $\text{sim} > p$ , то вход  $\mathbf{I}$  считается опознанным. В противном случае включается функция сброс и процесс классификации повторяется.

Поиск. При включении (активации) функции сброс(т.е. при недостаточной степени сходства векторов  $\mathbf{I}$  и  $\mathbf{S}$ ) вектор  $\mathbf{U}$  обнуляется (устанавливается нулевой вектор  $\mathbf{U} = \{0, 0, \dots, 0\}$ ), что является условием старта (запуска) повторной процедуры классификации. При этом вектор  $\mathbf{S}$  полагается равным входному вектору  $\mathbf{I}$ :  $\mathbf{S} = \mathbf{I}$ , и классифицируется непосредственно входной образ, отображаемый вектором  $\mathbf{I}$ . Процесс классификации опять-таки включает распознавание и сравнение и продолжается до тех пор, пока не будет выявлен класс, обеспечивающий достаточную степень сходства и исключающий запуск функции сброс. В противном случае, когда такой класс выявить не удастся, в слое распознавания открывается новый класс.

Весовые векторы матрицы  $W_{ji}$  рассчитываются так, чтобы обеспечить максимальное значение скалярного произведения для входа  $\mathbf{I}$ .

Адаптация весов. В ART-1-нейросетях различают два типа обучения: медленное и быстрое. При медленном обучении входные векторы настолько

кратковременно подаются на вход нейросети, что веса сети не успевают достигнуть своих асимптотических значений. Динамика нейросети (точнее динамика весов нейросети) описывается при этом дифференциальными уравнениями.

При быстром обучении входные векторы (образы) сохраняются на входе нейросети в течение времени, достаточного для достижения весовыми матрицами их стабильных значений. Мы ограничимся рассмотрением лишь быстрого обучения.

На стадии адаптации осуществляется уточнение весов обеих матриц. Для весов  $W_{ji}$  матрицы:

$$w_{iJ} = \frac{L s_i}{L - 1 + \sum_{k=1}^m s_k}, \quad (3.46)$$

где  $L$  - постоянная ( $L > 1$ , причем обычно  $L = 2$ ),  $s_i$  -  $i$ -я составляющая выходного вектора слоя сравнения,  $J$  - номер нейрона - «победителя» слоя распознавания и  $w_{iJ}$  вес (точнее  $i$ -я составляющая)  $W_{*J}$  или иначе  $w_{iJ}$  - это вес соединения  $i$ -го нейрона слоя сравнения и  $J$ -го нейрона слоя распознавания.

Веса бинарной  $W_{ij}$  матрицы изменяются в соответствии с правилом:

$$w_{ji}(t+1) = s_i \wedge w_{ji}(t), \quad (3.47)$$

т.е. соответствующий весовой вектор матрицы модифицируется таким образом, что он воспроизводит вектор  $\mathbf{S}$ .

Поток информации внутри ART-1-нейросети. Для каждого входного вектора  $\mathbf{I}$  ART-1-сеть ищет адекватный класс образов в слое распознавания  $F_2$ . Для этой цели на входе  $\mathbf{I}$  в слое сравнения  $F_1$  генерируется образ долговременной памяти или ДВП-образ, LTM-Muster)  $\mathbf{S}$ . При этом одновременно функция сброс отключается. Вектор  $\mathbf{S}$  трансформируется затем в

вектор  $\mathbf{T}$ , который в свою очередь активирует некоторый класс  $J$  в слое  $F_2$ . Слой распознавания  $F_2$  генерирует затем вектор  $\mathbf{U}$ , который преобразуется в так называемый ожидаемый вектор  $\mathbf{V}$  или иначе в вектор  $\mathbf{V}$ , ожидаемый для класса  $J$ .

Если степень совпадения между  $\mathbf{V}$  и входом ниже установленного порога, то генерируется новый вектор  $\mathbf{S}$ . В этом случае активируется функция Сброс, а выявленный ранее класс  $J$  в слое распознавания  $F_2$  забывается, потому что введенный образ не может быть отнесен к этому классу. Затем осуществляется новый поиск подходящего класса для входного вектора  $\mathbf{I}$ . Этот поиск заканчивается тогда, когда в слое сравнения не активируется функция сброс или же слой распознавания  $F_2$  дополняется еще одним классом.

ART-2 и ART-2a. Их главное отличие от ART-1-сетей: они обрабатывают не бинарные, а действительные входные векторы. ART-2a-сеть представляет собой «более быстрый» вариант сети ART-2. В силу этого многими авторами ART-2a-сети рекомендуются для решения сложных задач. Сравнительными исследованиями установлено, что качество классификации (образование классов) сетями ART-2 и ART-2a почти во всех случаях идентично. Скорость же обучения (сходимости) при использовании ART-2a-сетей значительно выше по сравнению с ART-2-сетями.

ART-3-сети. Они разработаны для моделирования биологических механизмов. Другое их достоинство: простота использования в каскадных архитектурах нейросетей.

ARTMAP. Они объединяют элементы обучения и самообучения (или обучения с поощрением и без поощрения, supervised and unsupervised learning). Для этого обычно формируется комбинация из двух ART-сетей.

FUZZY-ART-сети (нечеткие ART-сети). Они представляют собой прямое расширение ART-1-сетей путем применения средств нечеткой логики (Fuzzy Logik).

### **3.11. Применение нейронных сетей**

Нейросетевая технология динамично развивается и находит применение в таких областях, как распознавание образов, диагностики сложных технических объектов (например, двигателя), экология и науки об окружающей среде (прогнозы погоды и различных катаклизмов), построение математических моделей, диагностике различных заболеваний, измерении уровня сахара в крови, идентификация личности), системах управления, геология (прогнозирование наличия полезных ископаемых) и т.д. На данный момент как в России, так и за рубежом, накоплен богатый опыт применения отдельных типов нейронных сетей к различным задачам. Имеется потребность разработки единой подхода конструирования и обучения различного вида нейронных сетей применительно к решению широкого класса задач моделирования физико-технических объектов.

Нейронную сеть лучше использовать тогда, когда неизвестен точный вид связей между входами и выходами. Если бы он был известен, то связь можно было бы моделировать непосредственно. Другая существенная особенность нейронных сетей состоит в том, что зависимость между входом и выходом находится в процессе обучения сети. Для обучения нейронных сетей применяются алгоритмы двух типов (разные типы сетей используют разные типы обучения): управляемое ("обучение с учителем") и не управляемое ("без учителя"). Чаще всего применяется обучение с учителем, и именно этот метод мы сейчас рассмотрим (о неуправляемом обучении будет рассказано позже).

В данном разделе приведены оригинальные разработки для применения в различных областях

## **Нейронные сети для управления свойствами тампонажных цементов**

Автоматизации процесса прогнозирования свойств цементов выполняется на основе обученной нейронной сети. Под тампонажным цементом понимается продукт, состоящий из смеси тонкомолотых вяжущих веществ (портландцемент, шлак, известь), минеральных и органических добавок, после затворения которого водой получают раствор, а затем цементный камень.

Основные действия заключаются в следующем:

1. Подготовка данных
2. Обучение нейронной сети;
3. Прогноз свойств тампонажных цементов.

Для прогнозирования свойствами цемента используется трёхслойная нейронная сеть с входным, выходным и одним скрытым слоем. Данная нейронная сеть является полносвязной, поскольку каждый нейрон следующего слоя связан со всеми аксонами нейронов предыдущего слоя. Количество нейронов в каждом слое равно количеству входов нейронной сети. Входами нейронной сети являются значения четырёх компонентов тампонажного цемента, а выходами – значения свойств тампонажного цемента.

Перед началом обучение нейронной сети необходимо задать скорость обучения и параметр сигмоиды. По-умолчанию значения этих величин равны 0,1 и 0,5 соответственно. Для обучения нейронной сети готовились эталонные смеси с известными значениями компонентов. Прогноз свойств тампонажных цементов осуществляется на основе обученной нейронной сети.

## **Нейронная сеть для распознавание номеров вагонов**

На крупных предприятиях, где обрабатывается большое количество подвижного состава в течении суток, необходимо учитывать номера вагонов, их вес, объем и тару. Операции по учету вагонов выполняются вне помещений, в любую погоду и на значительных расстояниях. Решение задачи в применении новой информационной технологии учета железнодорожных вагонов. Для этой

цели может использоваться автоматизированная информационная система для учета железнодорожных вагонов (АИС УЖДВ), представляющая программно-аппаратный комплекс, в состав которого входят (рис 1):

- 1) система технического зрения (СТЗ) с нейронной сетью для распознавания номеров вагонов - основа автоматизированной технологии учета,
- 2) автономные электронные терминалы (ЭТ) (электронные записные книжки) для оперативного ввода информации по порожняку,
- 3) программная система учета, обработки и хранения полученной информации,
- 4) инфракрасные локаторы для синхронизации системы технического зрения,

При осмотре и регистрации подвижного состава информация о номерах, объеме и tare вводится в АЭТ, заменяющий бумагу и предназначенный для работы вне помещений. Введенные данные через специальный интерфейс переносятся в основную базу данных главного компьютера.

После загрузки вагонов или после других процедур с вагонами, состав проходит мимо камеры системы технического зрения, которая регистрирует и распознает номера вагонов, сравнивает их с введенными из АЭТ данными по порожняку. На основании этой информации ведется база данных на отправленные вагоны, в которой указан номер, вес тары, вес груза, а также другая необходимая информация.

Для распознавания номеров цифр в СТЗ была выбрана НС *Липпмана-Хемминга*, представляющая собою сеть с двуслойной топологией, прямой связью между слоями и с обучением с супервизором (рис. 1). Число нейронов во входном слое равно размерности векторов памяти, а число нейронов в выходном слое равно числу векторов памяти.

НС Липпмана-Хемминга была выбрана по причине приемлемого по сравнению с другими сетями быстродействия, по причине относительной простоты реализации, и отсутствия процесса обучения в обычном его итерационном смысле. Обучение этой НС производится непосредственно в процессе распознавания на основе эталонных изображений цифр.

Задача системы - автоматизировать процесс считывания номеров и обеспечить учет полученной информации посредством автоматизированного программно-аппаратного комплекса, который должен выполнять в автоматическом режиме следующие функции:

- 1) для каждого проходящего ж/д вагона определять его номер,
- 2) считывать весовые характеристики вагона со специальных весов,
- 3) заносить всю полученную и рассчитанную информацию в базу данных для возможности ее дальнейшего анализа, учета и хранения,
- 4) обеспечивать безбумажную технологию учета параметров вагонов с использованием автономных электронных терминалов (АЭТ).

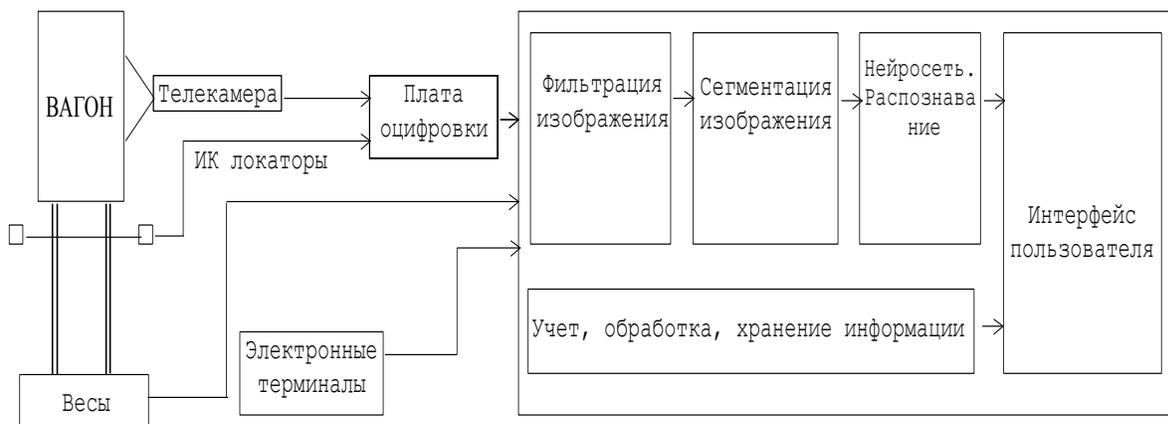


Рис.3.11.1. Структурная схема АИС УЖДВ

При прохождении вагона мимо телекамеры, с инфракрасных локаторов на плату оцифровки поступает сигнал, означающий, что номер вагона находится в поле видимости телекамеры. В этот момент захватывается и оцифровывается

изображение. Далее, изображение поступает в блок распознавания. Если в АИС УЖДВ используются электронные весы, то информация о весе, номере вагона, дате и времени распознавания заносится в соответствующие БД.

### **Бесконтактный видеоуровнемер**

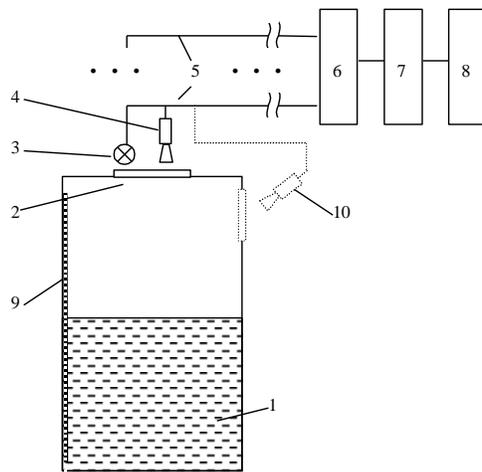
При учете жидкости, сыпучих веществ в емкостях существуют трудности измерения в тумане, запыленности, узких пространствах, электризации лучше использовать бесконтактные оптические методы.

Бесконтактный видеоуровнемер, состоящий из (рис. 1):

- *видеокамеры*, закрепленной неподвижно над поверхностью жидкости таким образом, что ее оптическая ось проходит через вертикальную линейку на внутренней стенке резервуара,
- канала передачи сигнала,
- устройства ввода изображения,
- цифрового вычислительного устройства,
- устройства отображения результатов измерения,
- измерительной линейки, расположенной перпендикулярно поверхности жидкости и представлена делениями, расположенными через определенные интервалы. Рядом с делениями наносятся соответствующие уровням цифры,

На этапе предобработки получаемого с камеры изображения уровня и ближайшей цифры выполняются процедуры: нейросетевой фильтрации, выделения границ области изменения освещенности, граничной сегментации. После нахождения узловых точек выполняется вычисление значения уровня.

Для фильтрации изображения применен фильтр, основанный на четырехслойной нейронной сети прямого распространения. На входы нейронной сети поступают сигналы с помехами. На выходе формируется отклик без помех.



- |                           |                                       |
|---------------------------|---------------------------------------|
| 1 – резервуар с жидкостью | 6 – коммутатор                        |
| 2 – прозрачное окно       | 7 – устр-во ввода телеизображения     |
| 3 – лампа подсветки       | 8 – цифровое вычислительное устр-во   |
| 4 – телевизионная камера  | 9 – линейка                           |
| 5 – канал связи           | 10 – альтернативный вариант установки |

Рис. 3.11.2. Бесконтактный видеоуровнемер

Для обеспечения универсальности системы, реализована процедура масштабирования графического изображения: после выделения знакомест (т.е. выделения в исходном изображении областей, предположительно являющихся цифрами). Обозначенные изображения поочередно приводятся к требуемому размеру, а только потом производится их распознавание.

Для распознавания номеров цифр использована НС *Липпмана-Хемминга*, представляющая сеть с двуслойной топологией, прямой связью между слоями и с самообучением (рис. 4). Число нейронов во входном слое равно размерности векторов памяти, а число нейронов в выходном слое равно числу векторов памяти.

НС Липпмана-Хемминга была выбрана по причине быстрого действия, относительной простоты реализации, и отсутствия итерационного процесса обучения. Обучение этой НС производится непосредственно в процессе распознавания на основе эталонных изображений цифр.

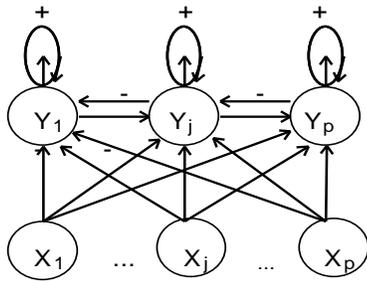


Рис. 3.11.3. Нейронная сеть Липпмана-Хемминга

Число нейронов  $N$  во входном слое равно размерности векторов памяти, а число нейронов в выходном слое равно числу  $M$  векторов памяти (12, по числу цифр, плюс специальные символы).

### Нейросетевая модель теплотребления зданий

Система теплотребления представляется в пространстве состояний четырьмя координатами: температурой теплоносителя в подающем трубопроводе  $T_{np}$ , температурой наружного воздуха  $T_{нар}$ , температурой отводимого теплоносителя  $T_{обр}$  и температурой внутри помещения  $T_{вн}$ . Для прогнозирования теплотребления и управления использовалась многослойная нейронная сеть с обучением обратным распространением ошибки. Порядка 230 тыс. записей от 7 датчиков использовалось для обучения нейронной сети. При этом качество данных низкое, сигнал зашумлен, имеются многочисленные пропуски, связанные с потерей контакта с датчиком, отключением системы, выбросы, связанные с ошибками в работе контрольной аппаратуры и программного обеспечения, что требует проведения полноценной предобработки имеющихся данных.

Большой интервал опроса в 1000 мсек. в некоторой степени решил проблему пропущенных значений (если в один интервал попадало несколько

отсчетов бралось их среднее значение) и позволил совместить показания датчиков на временной оси. Предобработка данных включала линейное масштабирование на интервал  $[0; 1]$ .

Экспериментальным путем подбирались оптимальная структура нейросетевой модели: многослойный перцептрон (МСП)  $4 \times 9 \times 1$ . Параметрическая идентификация (обучение нейросети) проводилось по алгоритму Resilient Propagation. Выбор был обусловлен заметным сокращением времени обучения (в 5 раз) в сравнении с обычным алгоритмом обратного распространения ошибки (Back Propagation). В отличие от стандартного метода обратного распространения ошибки алгоритм Resilient Propagation использует только знаки частных производных для подстройки весовых коэффициентов, коррекция весов происходит после предъявления сети всех примеров из обучающей выборки. Для работы алгоритма задаются параметры, определяющие величину следующей коррекции веса: шаг спуска - коэффициент увеличения скорости обучения при не достижении результата (использовалось значение 0.5); шаг подъема – коэффициент уменьшения скорости обучения в случае пропуска алгоритмом оптимального результата (использовалось значение 1.2). В качестве функции активации использовалась сигмоида с коэффициентом крутизны 1.0.

Обе модели показали приемлемые результаты по точности прогнозирования, хотя время обучения второй модели было значительно меньше, в среднем порядка 2500 эпох против 7500.

Синтезированная прогностическая модель влияния уличной температуры и температуры воды в подающем трубопроводе системы теплоснабжения здания на температуру внутри здания использовалась для управления процессом теплоснабжения с оптимальным законом регулирования.

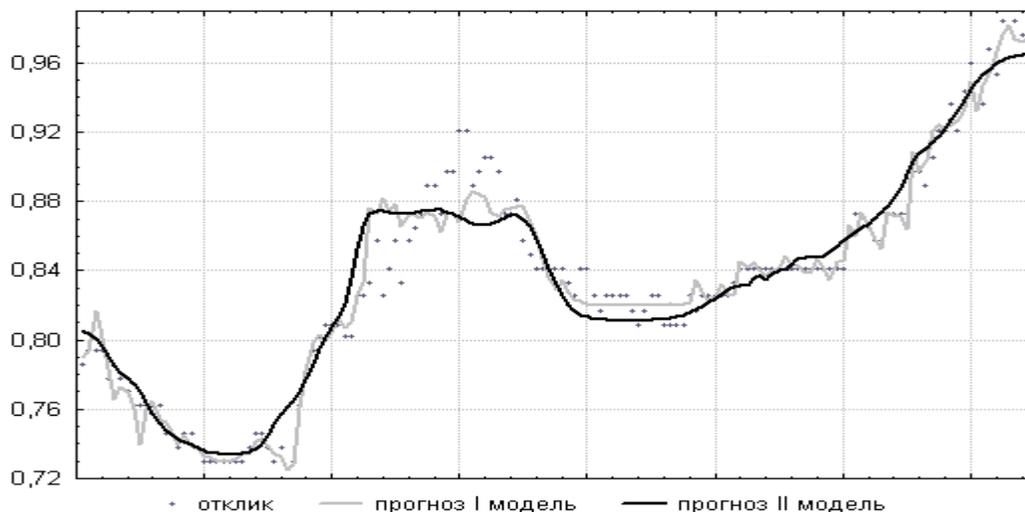


Рис. 3.11.4. Предсказание  $T_{вн}$  на данных не участвующих в обучении

Синтезированная прогностическая модель влияния уличной температуры и температуры воды в подающем трубопроводе системы теплоснабжения здания на температуру внутри здания использовалась для управления процессом теплопотребления с оптимальным законом регулирования.

### Нейросетевое моделирование химических систем

Прогнозирование свойств многокомпонентных химических систем углеводородов целесообразно выполнять с использованием нейронных сетей. Для моделирования многокомпонентной смеси бензина число выходных нейронов соответствует числу параметров, подлежащих измерениям (всего 5). Прогнозирование параметров бензина с учетом с учетом групп углеводородов осуществлялось на соответствующей нейронной сети. Рис.4. Для обучения нейронной сети на ее входы подавались переменные, соответствующие концентрациям компонентов, полученных из результатов кластерного анализа или метода главных компонент. Количество переменных может добавляться или уменьшаться в зависимости от точности предсказания моделируемых параметров.

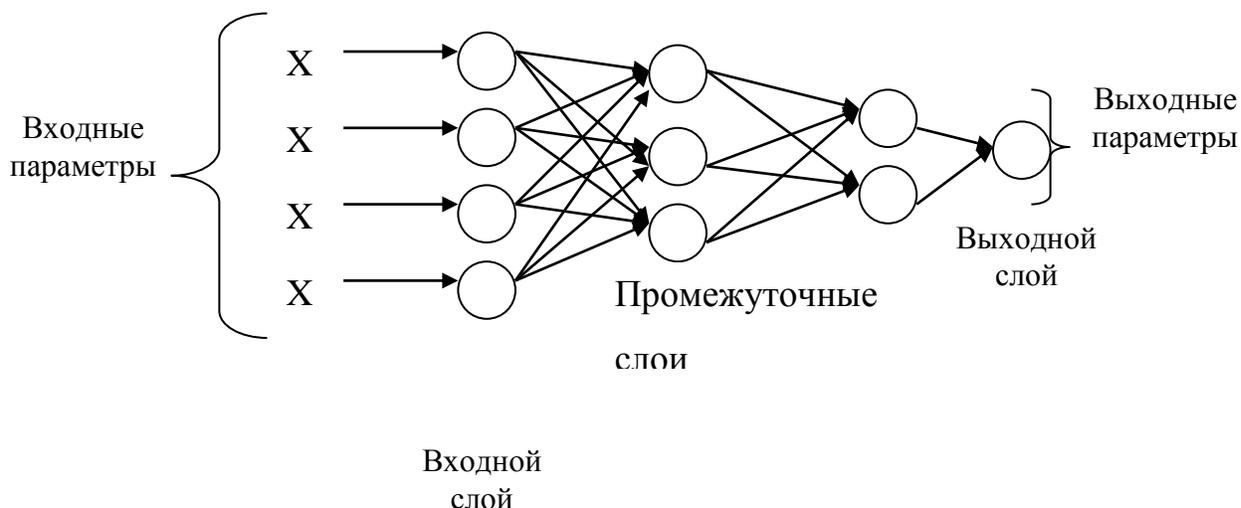


Рис. 3.11.5. Структура нейронной сети для калибровочной модели бензина

Количество входов, значений величин весов подбиралось экспериментально. Если веса, взаимодействующие с данным входом очень маленькие, в дополнение к соответствующим переменным добавляются еще группы углеводов, но при этом выполняется очередная процедура кластерного анализа с разбиением на число предыдущих кластеров плюс 1. Параметры нейронной сети приведены в таблице 1.

Числа входов, соответствующее группам углеводов, составляет от 10 до 30 групп. Суммарные их значения подавались на входы нейронной сети. Сначала выполнялось обучение нейронной сети, а затем вычислялась ошибка пресказания на контрольных смесях. Результаты показали, что существует насыщение, после которого увеличивать число групп не имеет смысла. Аналогично выполнялись эксперименты по определению числа нейронов в скрытом слое.

При выполнении процедур обучения возможно возникновения осцилляций и попадание в локальные минимумы. Во избежание этого к номеру текущего нейрона добавляется единица. Также возможно устранение осцилляции корректировкой скорости обучения нейронной сети.

Результаты исследования моделей, связывающих входные переменные в виде хроматограмм с выходными переменными в виде параметров бензинов, позволили получить адекватные модели представляющие полносвязную нейронную сеть с прямым распространением, параметры которой представлены в табл. 1.

**Табл. 1.** Параметры нейронной сети.

Нейронная сеть	Число вход. перем.	Число слоев	Число нейронов в слоях	Значение скорости
Модель бензина	28	3	10	0.2

Результаты прогнозирования октановых чисел показали ошибке в пределах 1 процента.

Применение нейронных сетей для моделирования смесей углеводородов в виде бензинов, позволяет с высокой точностью получить адекватные модели. Нейронные сети нашли применение в автоматизированных информационных системах контроля параметров бензина (АИС КПБ), на основе хроматографических преобразователей, достоинствами которых является минимальный размер пробы, низкая стоимость анализа, высокая точность определения параметров бензина.

### **Нейронные сети в защите информации**

Для защиты информации при передаче ее в сетях применяются секретные ключи. Для процедуры обмена секретными ключами, необходимо использование безопасных (защищенных от прослушивания) и, соответственно, очень дорогих каналов связи.

Для устранения этого требования, разработан метод ККК (названный так по первым буквам фамилий создателей - Идо Кантер, Вольфганг Кинцель, Еран

Кантер), основанный на синхронизации взаимно обучающихся нейронных сетей. Обычно нейронные сети обучаются при взаимодействии с учителем, который, поочередно подает на вход нейронной сети образцы.

При этом учитель «знает» правильный ответ, который должен выдать «ученик», и подстраивает его весовые коэффициенты таким образом, чтобы уменьшить разницу между выдаваемым и правильным ответом. Процесс обучения можно интерпретировать, как поиск глобального минимума функции ошибки, т. е. движение к некоторой фиксированной точке многомерного пространства.

Две нейронные сети обучаются на выходных данных друг друга. Сети синхронизируются и их весовые коэффициенты станут равными. В то же время сторонний слушатель, перехватывая передаваемые образцы данных и используя аналогичную нейронную сеть, не сможет достаточно быстро синхронизировать ее с нейронными сетями участников обмена, потому что нейронные сети партнеров совершают хаотичные, но одинаковые) движения в многомерном пространстве своих весовых коэффициентов.

В каждом сеансе связи создается свой «одноразовый» ключ, что снимает проблему компрометации ключей. Также это снимает проблему управления ключами, т. е. проблемы хранения, защиты, распределения ключей.

Метод ККК не основан на теории чисел и предъявляет достаточно низкие требования к памяти и быстродействию аппаратных средств.

Каждая из сторон в методе ККК использует двухслойную нейронную сеть. Первый слой состоит из  $K$  независимых нейронов, второй слой из одного нейрона, вычисляющего произведение их  $K$  выходов.

Каждый из нейронов первого слоя имеет  $N$  входов с весами  $w_{k,n}$  ( $1 \leq k \leq K$ ,  $1 \leq n \leq N$ ). Веса нейронов являются целыми числами в интервале  $\{-L, \dots, +L\}$ .

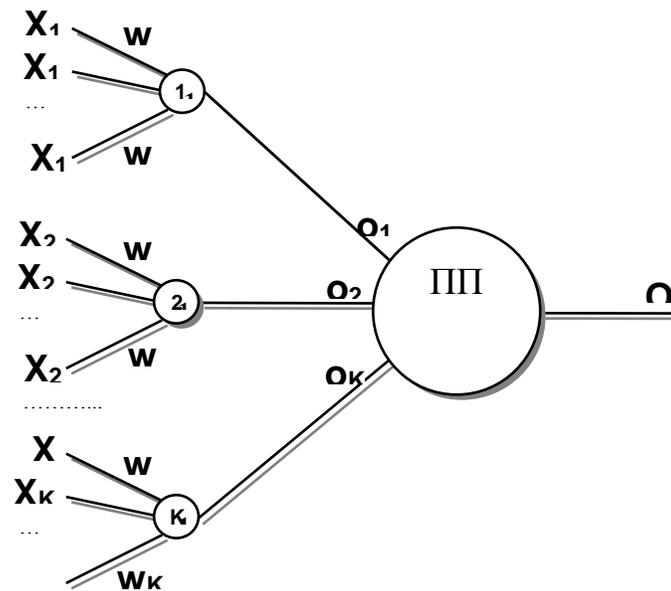


Рис. 3.11.6. Структура нейронной сети

Выбор дискретных значений весов обусловлен удобством практической реализации метода, т. к. обмен сообщениями в современных сетях основан на последовательности бит, которые наиболее удобно представлять в виде целых чисел.

На вход сети подаются вектора  $X_k = (x_{k1}, \dots, x_{kn})$ , состоящие из  $N$  бит:  $x_{ki} \in \{-1, +1\}$ . Входные вектора можно рассматривать, как последовательность бит.

В качестве активационной функции для нейронов первого уровня используется функция знака  $y = \text{sign}(x)$  (ступенька).

Выход  $k$ -го нейрона первого уровня  $o_k$  имеет простую геометрическую интерпретацию. Он показывает взаимное расположение вектора весов нейрона и входного вектора относительно гиперплоскости в  $N$ -мерном пространстве, перпендикулярной к вектору весов нейрона. Если вектора расположены по одну сторону гиперплоскости, выход равен 1, иначе -1.

Выход всей нейронной сети определяется как  $Y = y_1 \bullet y_2 \bullet \dots \bullet y_k$ . Видно, что  $Y \in \{-1, +1\}$ .

Участники обмена А и В начинают с произвольных матриц весовых коэффициентов  $\{w_{k,n}\}$ . На каждой итерации формируется случайный входной вектор  $X = (x_1, \dots, x_n)$ . Стороны вычисляют выходы своих нейронных сетей  $Y^A$  и  $Y^B$  и обмениваются их значениями по каналу связи.

Если выходы нейронных сетей различаются ( $Y^A \neq Y^B$ ), то стороны переходят к следующей итерации без изменения весовых коэффициентов.

В противном случае стороны изменяют весовые коэффициенты тех нейронов, для которых  $y_k = Y$ , по следующему правилу:

$$w_{k,n} = \text{bound}_{-L, L}(w_{k,n} - y_k \bullet y_k)$$

Где  $\text{bound}_{-L, L}(x)$  возвращает любое значение  $x$ , вышедшее за границу интервала  $-L, \dots, +L$ , на ближайшую границу интервала.

Если выходы нейронных сетей не отличаются на протяжении 50-70 итераций, то стороны могут считать, что синхронизация сетей завершена. После этого матрицы весовых коэффициентов у обеих нейронных сетей становятся одинаковыми.

. Синхронизация нейронных сетей с одним нейроном, имеющий один вход, полностью обуславливается существованием границ значений весового коэффициента:  $-L, \dots, +L$ ,  $a_i, b_i$  – текущие значения данного коэффициента. На каждой итерации выбирается случайный входной вектор  $x_i$  и стороны либо игнорируют его, либо совершают одновременное движение весовых коэффициентов в одну и ту же сторону.

Так как коэффициенты будут задевать конечные границы интервала достаточно часто (из-за хаотичного движения), то в итоге расстояние между весовыми коэффициентами станет равным 0, и сети станут окончательно одинаковыми.

Если число входов больше одного, то веса нейронов совершают управляемые движения в многомерном кубе. Однако протокол обмена с помощью нейронных сетей из одного нейрона может быть просто атакован и взломан любой другой подобной нейронной сетью. Поэтому подобные сети не могут быть использованы для формирования общего секретного ключа в открытых каналах связи. Вариант с несколькими нейронами является более сложным, так как каждая из сторон может обновлять веса различных нейронов на каждой итерации. В этом случае будет статистическая версия.

После синхронизации нейронных сетей участники обмена получают возможность сгенерировать пару идентичных секретных ключей (например, применив хеш-функцию к весовым коэффициентам) и приступить к обмену сообщениями.

Преимущества метода заключаются в следующем: кардинально упрощается схема обмена ключами. Участники обмена начинают сеанс связи с произвольными нейронными сетями. В начале сеанса, после синхронизации нейронных сетей, вырабатывается ключ шифрования. В дальнейшем, передаваемые сообщения шифруются с использованием этого ключа. Сгенерированный ключ может даже не показываться пользователю.

### **Применение нейронных сетей для прогнозирования окон в сетях ТСП/Р**

Нейронные сети применены в задачах прогнозирования и оптимизации процесса передачи данных в сетях ТСП/Р. Для ускорения и оптимизации процесса передачи больших объемов данных протокол ТСП определяет метод управления потоком, называемый «методом скользящего окна», который позволяет отправителю посылать очередной сегмент, не дожидаясь подтверждения о получении в пункте назначения предшествующего сегмента. Размер окна определяет, сколько сегментов отправителю разрешено посылать, не дожидаясь подтверждения со стороны получателя. Размер окна выбирается

таким образом, чтобы подтверждения успевали приходить вовремя, и остановки передачи не происходило – для этого и предназначен метод скользящего окна. Размер окна может динамически изменяться получателем и зависит от качества и загруженности канала связи.

Размер окна, объявленный получателем, ограничивается произведением полосы пропускания канала (бит/с) на RTT (время распространения пакета туда и обратно). RTT – это сумма времени транспортировки сегмента от отправителя к получателю и времени движения отклика от получателя к отправителю. Рассчитанное RTT – это время между отправкой байта с конкретным номером последовательности и получением подтверждения на этот номер последовательности.

Конечной целью регулирования трафика является установление соответствия между темпом передачи и возможностями приема. Причиной перегрузки может быть не только ограниченность размера буфера, но и недостаточная пропускная способность какого-то участка канала. С учетом этого обстоятельства каждый отправитель формирует два окна: «окно получателя» и «окно перегрузки». Каждое из этих окон задает число байтов, которое может послать отправитель. Реальное число байтов, которое разрешено послать, равно минимальному из этих окон.

Имеются значительные резервы уменьшения потерь в пропускной способности сети. Во-первых, перегрузка не прогнозируется, а обнаруживается по самому факту отсутствия пакетов подтверждения после очередного перемещения окна, во-вторых, по этому факту нельзя судить о величине перегрузки и следующее состояние окна устанавливается методом «проб и ошибок». Задача заключается в формировании заблаговременно, не заходя в критическую область обнаружения потерянных пакетов, оценок как месторасположения временного интервала возможного проявления перегрузки, так и величины самой перегрузки. Следовательно, необходимо осуществлять

прогноз RTT-задержки, а уже потом по величине спрогнозированной RTT-задержки на следующем шаге можно будет судить об уровне перегрузки, и принимать решения. Прогнозирование RTT-задержки относится к такому классу задач, где зависимость между входными и выходными переменными сложна и нахождение закономерностей в больших объемах статистических данных требует нетривиальных алгоритмов и занимает много времени. Наиболее эффективным средством решения задачи прогнозирования будет использование нейросетевого предиктора (от англ. predictor – предсказатель) в качестве механизма прогнозирования.

Функциональная схема нейросетевой системы прогнозирования размера TSP-окна представлена на рис. 3.11.7

Принцип действия системы следующий. Из ядра TCP/IP получаем данные об RTT-задержки для всех текущих соединений (сокетах), эти данные сохраняются в буфере, из буфера последовательность данных (вектор) поступает на вход нейронной сети, на выходе формируется прогноз следующего значения RTT-задержки. На основании нейросетевого прогноза рассчитывается размер следующего TSP-окна (размер буфера окна перегрузки). Посредством ядра TCP/IP выполняется управляющее воздействие на пропускную способность канала. При помощи протокола SNMP в MIB формируются качественные и количественные данные о проходящем потоке данных (трафик) через данный сетевой адаптер. Также в качестве механизма доступа к сетевой среде целесообразно использовать специализированные сетевые драйверы (например, WinPcap). Подобные драйверы предоставляют доступ к сетевой среде в обход стандартных средств операционной системы, что предоставляет неограниченные возможности как для сбора статистической информации по сетевому трафику, так и для разбора каждого конкретного пакета, т.к. есть возможность получить данные из абсолютно всех полей пакетов всех уровней, в то время как стандартные средства предоставляют доступ только к полям,

относящимся к протоколам верхних уровней и не пропускают, например, пакеты с неправильной контрольной суммой (Checksum).



Рис.3.11.7. Функциональная схема системы прогнозирования размеров окон с использованием нейронных сетей

### Нейросетевые модели энергопотребления

Множество факторов оказывают влияние на уровень энергопотребления, в том числе суточная и годовая периодичность природных явлений, а также недельный цикл работы и отдыха. Некоторые факторы либо слабо формализуемы, либо отсутствуют, либо затруднено их накопление. Проанализировать и учесть влияние каждого практически невозможно.

Для получения результатов прогноза, удовлетворяющих заданным условиям необходимо провести качественный анализ данных и выбрать диапазон их изменения, наиболее точно отражающий реальное поведение исследуемого параметра. Целью предварительного анализа данных является выявление основных закономерностей: тенденции, сезонные изменения, циклы, случайные или нерегулярные изменения.

Для решения задачи предсказания ЭП разработана модель предиктора, осуществляющая предсказание выходного вектора объекта по его предыстории. В качестве предиктора выбрана многослойная нейронная сеть (МНС) с алгоритмом обучения методом обратного распространения ошибки. В МНС каждый слой рассчитывает нелинейное преобразование от линейной комбинации сигналов предыдущего слоя, таким образом, достигается аппроксимация произвольной многомерной функции при соответствующем выборе количества слоев, диапазоне изменения сигналов и параметров нейронов.

При синтезе модели предиктора учитывались следующие факторы, оказывающие влияние на уровень потребления электроэнергии:

- номер недели  $W$  для учета влияния сезонных составляющих;
- тип дня  $D$  (от 1 до 7 – с понедельника по воскресенье и 8 для праздничных дней);
- среднесуточная температура  $T$ .

Перечисленные факторы использовались в качестве входных переменных ИНС. Схема организации входов предиктора приведена на рисунке 3.11.8

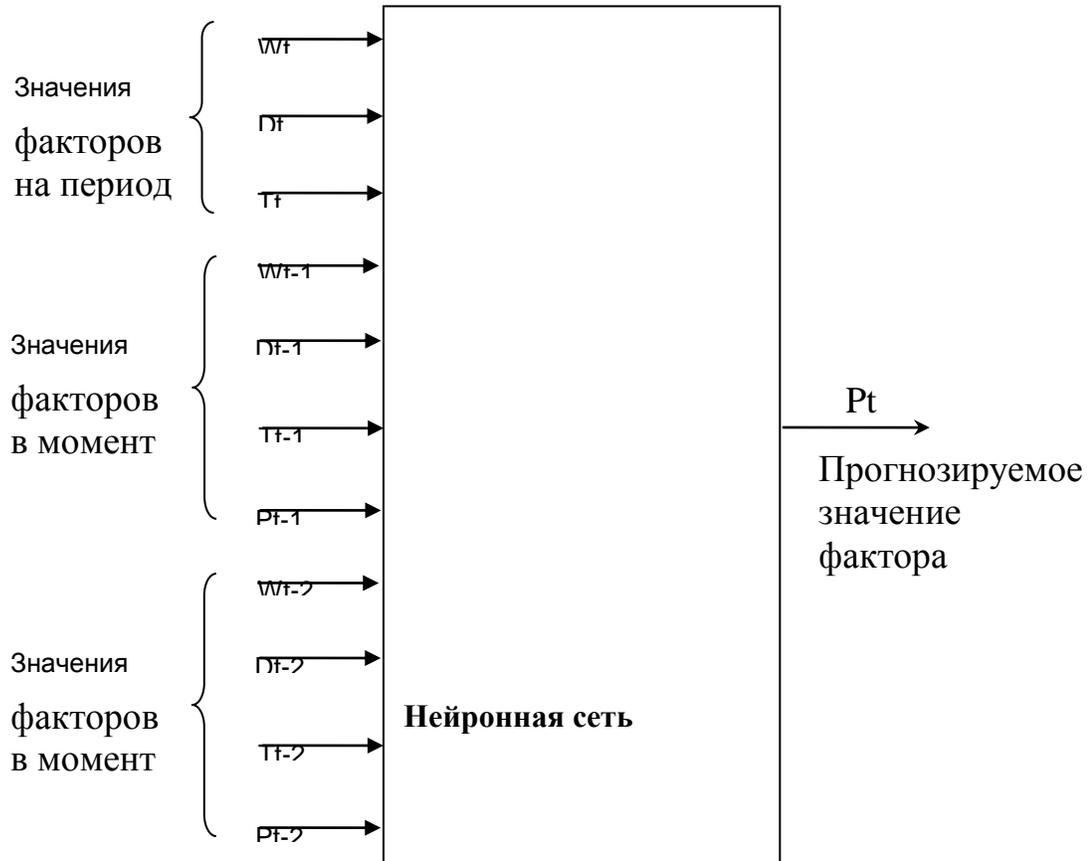


Рис.3.11.8. Схема организации предиктора электрической нагрузки

При ретроспективе данных  $n = 1$  (используются данные только за один предыдущий день), использовалась следующая структура нейронной сети:

$$P_t = NN(W_t, D_t, T_t, W_{t-1}, D_{t-1}, T_{t-1}, P_{t-1}),$$

где  $W_t$  – номер недели в момент времени  $t$ ;

$D_t$  – тип дня в момент времени  $t$ ;

$T_t$  – температура в момент времени  $t$ ;

$W_{t-1}$  – номер недели в момент времени  $t-1$ ;

$D_{t-1}$  – тип дня в момент времени  $t-1$ ;

$T_{t-1}$  – среднесуточная температура в момент времени  $t-1$ ;

$P_{t-1}$  – фактическое ЭП в момент времени  $t-1$ .

При построении модели, учитывающей ретроспективные данные за два последних дня, структура нейронной сети описывается следующей формулой

$$P_t = NN(W_t, D_t, T_t, W_{t-1}, D_{t-1}, T_{t-1}, P_{t-1}, W_{t-2}, D_{t-2}, T_{t-2}, P_{t-2}).$$

Описанный предиктор позволяет прогнозировать нагрузку на выбранный период только на конкретный час. Соответственно для прогнозирования почасового ЭП на выбранный период необходимо применить 24 описанных выше предиктора, либо применив в качестве дополнительного параметра номер часа, использовать подобную архитектуру НС с более мощной конфигурацией. Как показала практика использование одной, но более мощной комплексной НС по сравнению с применением нескольких более простых реального выигрыша не приносит и приводит к усложнению модели.

После выбора архитектуры нейронной сети для каждой модели, проводились исследования зависимости ошибки прогноза от: количества обучающих примеров, способа масштабирования и т.д. В результате выявлен наиболее оптимальный набор параметров НС для каждой модели.

Описание параметров нейронной сети для моделей

Параметр	1-й метод	2-й метод
Число нейронов во входном слое	7	4
Число нейронов в скрытом слое	10	10
Число нейронов в выходном слое	1	24
Значение функции активации	2.3	1.7
Средняя ошибка прогноза	1-3%	1-2%
Максимальная ошибка прогноза	4-5%	5-6%

На рисунке 3.11.9. изображены графики изменения фактических и прогнозируемых значений энергопотребления.

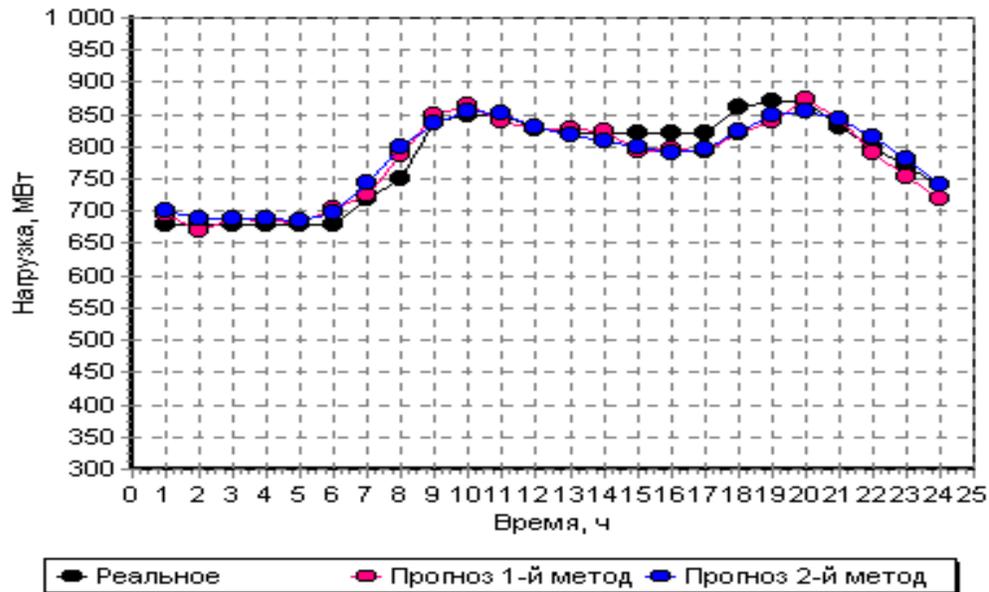


Рис. 3.11.9. Графики фактического и предсказанного энергопотребления

Для уменьшения влияния скачков нагрузки, применялось сглаживание значений, методом скользящего среднего. Это позволило снизить ошибку в среднем примерно на 1%. Учитывая значения ошибок для каждой модели можно сделать вывод, что разработанные модели позволяют осуществлять прогнозирование требуемых показателей более достоверно, по сравнению с действующей на данный момент системой.

### Многослойный персептрон для сжатия с потерями

При передаче по сетям необходимо сжатие данных. Существует сжатие данных без потерь (сжатие Хаффмена) и с потерями (JPEG). Применение ИНС для сжатия изображений с потерями привлекает своей эффективностью. Пусть дано прямоугольное изображение, каждый пиксел которого характеризуется своей яркостью. Изображение разбивается на  $P$  прямоугольных кадров размером  $N_h \times N_v$  каждый. Кадр сжимается в вектор данных размерностью  $L$

$(L < N_h \cdot N_v)$ , который затем восстанавливается (например, после хранения или передачи по медленным каналам связи) в кадр того же размера  $N_h \times N_v$ .

Для сжатия возможно использование простейшего двухслойного персептрона с линейными функциями активации (см. рис. 3.11.10).

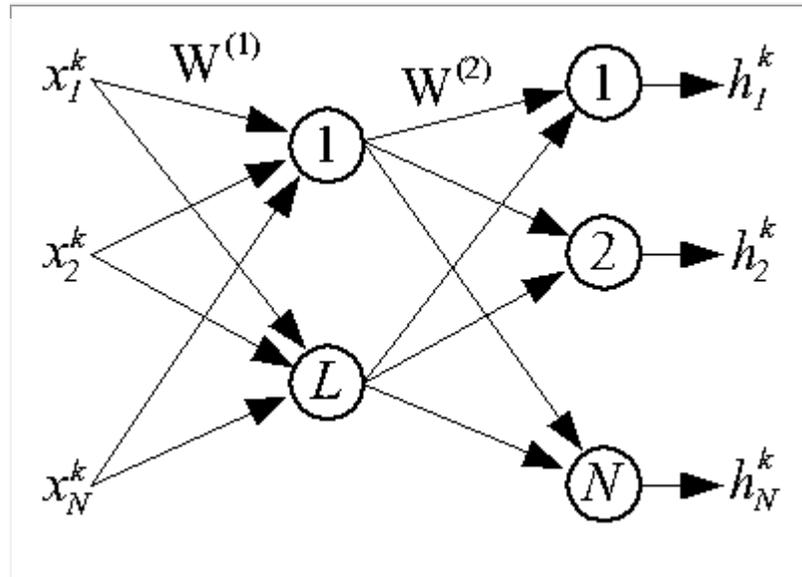


Рис. 3.11.10. Структурная схема двухслойного персептрона с линейными функциями активации

Здесь  $x_j^k$ ,  $j = 1, 2, \dots, N$ ,  $N = N_h \cdot N_v$  — значение яркости  $j$ -ого пикселя в  $k$ -ом кадре,  $k = 1, 2, \dots, p$ .

Сжатие (компрессия) данных осуществляется первым слоем нейронов, а восстановление (декомпрессия) — выходным слоем. Сеть является автоассоциативной, поскольку ее выходной вектор  $\mathbf{H}^k$  должен совпадать с входным  $\mathbf{X}^k$ .

Веса первого слоя нейронов в матричной форме обозначаются  $\mathbf{W}^{(1)}$ , а выходного слоя —  $\mathbf{W}^{(2)}$ . Вследствие линейности функций активации и однонаправленности распространения сигналов имеем:

$$\mathbf{H}^k = \mathbf{W}^{(2)} \cdot \mathbf{W}^{(1)} \cdot \mathbf{X}^k.$$

Обучение сети, состоящее в оптимальном подборе весов, составляющих матрицы  $\mathbf{W}^{(1)}$  и  $\mathbf{W}^{(2)}$ , подразумевает минимизацию целевой функции в виде:

$$E(\mathbf{W}) = \frac{1}{2} \sum_{k=1}^P \sum_{i=1}^M (h_i^k - x_i^k)^2.$$

### **Искусственные нейронные сети сегодня**

Имеется много впечатляющих демонстраций возможностей искусственных нейронных сетей: сеть научили превращать текст в фонетическое представление, которое затем с помощью уже иных методов превращалось в речь [7]; другая сеть может распознавать рукописные буквы [1]; сконструирована система сжатия изображений, основанная на нейронной сети [2]. Все они используют сеть обратного распространения – наиболее успешный, по-видимому, из современных алгоритмов. Обратное распространение, независимо предложенное в трех различных работах [8, 5, 6,], является систематическим методом для обучения многослойных сетей, и тем самым преодолевает ограничения, указанные Минским.

Как подчеркивается в следующих главах, обратное распространение не свободно от проблем. Прежде всего нет гарантии, что сеть может быть обучена за конечное время. Много усилий, израсходованных на обучение, пропадает напрасно после затрат большого количества машинного времени. Когда это происходит, попытка обучения повторяется – без всякой уверенности, что результат окажется лучше. Нет также уверенности, что сеть обучится наилучшим возможным образом. Алгоритм обучения может попасть в «ловушку» так называемого локального минимума и будет получено худшее решение.

Разработано много других сетевых алгоритмов обучения, имеющих свои специфические преимущества. Но никакая из сегодняшних сетей не является

панацеей, все они страдают от ограничений в своих возможностях обучаться и вспоминать.

Областью, продемонстрировала свою работоспособность, имеет уникальные потенциальные возможности, много ограничений и множество открытых вопросов.. Существует, следовательно, опасность, что искусственные нейронные сети начнут продавать раньше, чем придет их время, обещая функциональные возможности, которых пока невозможно достигнуть. Если это произойдет, то область в целом может пострадать от потери кредита доверия и вернется к застою периоду семидесятых годов. Для улучшения существующих сетей требуется много основательной работы. Должны быть развиты новые технологии, улучшены существующие методы и расширены теоретические основы, прежде чем данная область сможет полностью реализовать свои потенциальные возможности.

### **Перспективы на будущее**

Искусственные нейронные сети предложены для задач, простирающихся от управления боем до присмотра за ребенком. Потенциальными приложениями являются те, где человеческий интеллект малоэффективен, а обычные вычисления трудоемки или неадекватны. Этот класс приложений во всяком случае не меньше класса, обслуживаемого обычными вычислениями, и можно предполагать, что искусственные нейронные сети займут свое место наряду с обычными вычислениями в качестве дополнения такого же объема и важности.

### **Искусственные нейронные сети и экспертные системы**

Нейронные сети, скорее всего, будут сосуществовать совместно с нечеткой логикой, экспертными системами и различными методами оптимизации, взятыми из природы (муравьиные колонии, полет мухи, рост деревьев)

Можно представить себе искусственную систему, подражающую такому разделению труда. Искусственная нейронная сеть реагировала бы в

большинстве случаев подходящим образом на внешнюю среду. Так как такие сети способны указывать доверительный уровень каждого решения, то сеть «знает, что она не знает» и передает данный случай для разрешения экспертной системе. Решения, принимаемые на этом более высоком уровне, были бы конкретными и логичными, но они могут нуждаться в сборе дополнительных фактов для получения окончательного заключения. Комбинация двух систем была бы более мощной, чем каждая из систем в отдельности, следуя при этом высокоэффективной модели, даваемой биологической эволюцией.

### **Соображения надежности**

Прежде чем искусственные нейронные сети можно будет использовать там, где поставлены на карту человеческая жизнь или ценное имущество, должны быть решены вопросы, относящиеся к их надежности.

Подобно людям, структуру мозга которых они копируют, искусственные нейронные сети сохраняют в определенной мере непредсказуемость. Единственный способ точно знать выход состоит в испытании всех возможных входных сигналов. В большой сети такая полная проверка практически неосуществима и должны использоваться статистические методы для оценки функционирования. В некоторых случаях это недопустимо. Например, что является допустимым уровнем ошибок для сети, управляющей системой космической обороны? Большинство людей скажет, любая ошибка недопустима, так как ведет к огромному числу жертв и разрушений. Это отношение не меняется от того обстоятельства, что человек в подобной ситуации также может допускать ошибки.

Проблема возникает из-за допущения полной безошибочности компьютеров. Так как искусственные нейронные сети иногда будут совершать ошибки даже при правильном функционировании, то, как ощущается многими, это ведет к

ненадежности – качеству, которое мы считаем недопустимым для наших машин.

Сходная трудность заключается в неспособности традиционных искусственных нейронных сетей "объяснить", как они решают задачу. Внутреннее представление, получающееся в результате обучения, часто настолько сложно, что его невозможно проанализировать, за исключением самых простых случаев. Это напоминает нашу неспособность объяснить, как мы узнаем человека, несмотря на различие в расстоянии, угле, освещении и на прошедшие годы. Экспертная система может проследить процесс своих рассуждений в обратном порядке, так что человек может проверить ее на разумность. Сообщалось о встраивании этой способности в искусственные нейронные сети [3], что может существенно повлиять на приемлемость этих систем.

### **Заключение**

Нечеткая логика и искусственные нейронные сети являются важным расширением понятия вычисления. Они обещают создание автоматов, выполняющих функции, бывшие ранее исключительной прерогативой человека. Машины могут выполнять скучные, монотонные и опасные задания, и с развитием технологии возникнут совершенно новые приложения.

Теории нечеткой логики и искусственных нейронных сетей развиваются стремительно, но в настоящее время она недостаточна, чтобы быть опорой для наиболее оптимистических проектов. В ретроспективе видно, что теория развивалась быстрее, чем предсказывали пессимисты, но медленнее, чем надеялись оптимисты, – типичная ситуация. Сегодняшний взрыв интереса привлек к нечеткой логике и нейронным сетям тысячи исследователей. Резонно ожидать быстрого роста нашего понимания нечеткой логики и искусственных нейронных сетей, ведущего к более совершенным сетевым парадигмам и множеству прикладных возможностей.

### 3.12. Библиографический список

1. Заде, Л. А. Понятие лингвистической переменной и его применение к принятию приближенных решений / Л.А. Заде: пер. с англ. –М.: Мир, 1976. – 165 с.
2. Борисов, А. Н. Обработка нечеткой информации в системах принятия решений / А.Н. Борисов [и др.]. – М.: Радио и связь, 1989. – 304 с.
3. Круглов, В. В. Искусственные нейронные сети. Теория и практика / В.В. Круглов, В.В. Борисов. – М.: Горячая линия – Телеком, 2002. – 382 с. – ISBN 5-93517-031-0.
4. Штовба, С. Д. Проектирование нечетких систем средствами MATLAB. – М.: Горячая линия – Телеком, 2007. – 288 с. – ISBN 5-9351-359-X.
5. Макеева А.В. Основы нечеткой логики. Учебное пособие для Вузов. –Н. Новгород: ВГИПУ, 2009. – 59 с.
6. Чернов, В. Г. Основы теории нечетких множеств : учеб. пособие / В.Г. Чернов ; Владим. гос. ун-т.- Владимир : Изд-во Владим. гос. ун-та, 2010. – 96 с.
7. Пегат А. Нечеткое моделирование и управление / А. Пегат ; пер. с англ. — М. : БИНОМ. Лаборатория знаний, 2009. — 798 с.
8. Кричевский М.Л. Интеллектуальный анализ данных в менеджменте. Учеб. Пособие/ СПбГУАП. СПб., 2005.
9. Прикладные нечеткие системы: пер с япон./К. Асаи, Д. Ватада, С. Иваи и др. ; под редакцией Т Тэрано, К Асаи, М. Сугэно. М.: Мир, 1993 – 368 с.
10. Рутковская Д., Пилиньский М., Рутковский Л. Нейронные сети, генетические алгоритмы и нечеткие системы: Пер. с польск. И. Д. Рудинского. - М.: Горячая линия -Телеком, 2006. - 452 с.

11. Саати, Т. Аналитическое планирование. Организация систем / Т. Саати, К. Кернс : пер. с англ. – М.: Радио и связь. 1991. – 224 с.
12. Ф. Уоссермен. Нейрокомпьютерная техника: теория и практика. Пер. с англ. Ю.А.Зуева, В.А.Точенова, М. Мир- 1992, 184 с.