



**НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ  
УНИВЕРСИТЕТ «ТАШКЕНТСКИЙ ИНСТИТУТ  
ИНЖЕНЕРОВ ИРРИГАЦИИ И МЕХАНИЗАЦИИ  
СЕЛЬСКОГО ХОЗЯЙСТВА»**

**МЕТОДЫ И АЛГОРИТМЫ  
ПОСТРОЕНИЯ ПРИКЛАДНЫХ  
ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМ**

**ТАШКЕНТ – 2023**

**ИЗДАТЕЛЬСТВО «FAN ZIYOSI»**

УДК 330.115

**Д.Т.Мухамедиева. «Методы и алгоритмы построения прикладных интеллектуальных систем». Монография – Т.: Изд.«Fan ziyosi», 2023. 345 с.**

В работе описаны методы и алгоритмы построения прикладных интеллектуальных систем слабо формализованных процессов и практическое применение разработанных интеллектуальных систем. Рассмотрены основные составляющие Soft Computing - теория нечетких множеств, нейронные сети, эволюционные алгоритмы, особенности и принципы моделирования слабо формализованных процессов при нечетко заданной исходной информации.

Приведено описание практических интеллектуальных систем на основе технологий Soft Computing для альтернативного принятия решений. Книга рассчитана на широкий круг читателей, включающих специалистов по прикладной математике, инженеров, а также лиц, интересующихся вопросами применения технологии искусственного интеллекта при решении задач оценки состояния слабоформализуемых процессов, оптимизации, теории систем и общими вопросами принятия решений.

Рекомендовано к печати НТС

Научно-инновационного центра информационно-коммуникационных технологий при Ташкентском университете информационных технологий имени Мухаммада аль-Хорезми

Ответственный редактор

Доктор технических наук, профессор Ш.Х.Фазилов

© Изд.«Fan ziyosi» 2023г.

# СОДЕРЖАНИЕ

Введение	5
Глава 1. ОБЩИЕ СВЕДЕНИЯ ОБ ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМАХ	10
1.1. Общие понятия интеллектуальных систем	10
1.2. Цели и концепция интеллектуальной информационной системы	15
1.3. Концептуальная схема структурно-функциональной организации интеллектуальной системы обработки информации	19
1.4. История развития интеллектуальных систем	25
1.5. Классификация интеллектуальных систем	39
1.6. Создание базы знаний. Структура экспертных систем. База знаний.	47
1.7. Модели представления знаний и их классификация	56
Глава 2. КЛАССИФИКАЦИЯ И КЛАСТЕРИЗАЦИЯ	86
2.1. Задача классификации	86
2.2. Процесс классификации	90
2.3. Задача кластеризации	93
Глава 3. МЕТОД НЕЧЕТКИХ МНОЖЕСТВ	113
3.1. Основные понятия	113
3.2. Операции над нечеткими множествами	130
3.3. Операции, основанные на t-норм	141
3.4. Алгебраические операции над F- величинами	143
3.5. Численные методы для операций с F-величинами	159
3.6. Нечеткие отношения и нечеткие ограничения	161
3.7. Бинарные нечеткие отношения	164
3.8. Нечеткая логика	168
3.9. Нечеткие импликации	175
3.10. Семантический анализ нечеткой логики	181
3.11. Нечеткое лингвистическое моделирование	184
Глава 4. НЕЙРОННЫЕ СЕТИ	200
4.1. Однослойные и многослойные нейронные сети	200
4.2. Архитектура и типы нейронных сетей	203
4.3. Характеристики и вычислительные способности нейронных сетей	221

4.4. Искусственные нейронные сети. Персептрон	226
4.5. Нейронная сеть Хопфилда и алгоритм обучения	229
4.6. Многослойные нейронные сети. Алгоритм Хэмминга для обучения нейронных сетей	234
Глава 5. СТРУКТУРА ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМ УПРАВЛЕНИЯ	237
5.1. Принципы организации и управления интеллектуальными системами	237
5.2. Проблемы решения задач оптимального управления	258
5.3. Общая методология построения систем нечеткого управления	263
5.4. Нечеткий гибридный контроллер	270
5.5. Нечеткие координатно-параметрические адаптивные системы управления	274
5.6. Устройства управления в интеллектуальных системах управления. Управление на основе нечеткой логики	284
Глава 6. НЕЙРОННЫЕ РЕГУЛЯТОРЫ. НЕЧЕТКИЕ НЕЙРОННЫЕ РЕГУЛЯТОРЫ	293
6.1. Нейронная идентификация и управление	293
6.2. Структура нейронного регулятора	297
Глава 7. ПОСТРОЕНИЕ ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМ	310
7.1. Алгоритм обучения матричного представления нечетких систем логического вывода	310
7.2. Применение генетического алгоритма для решения задач оптимизации размещения и чередования культур в хлопковом севообороте	318
7.3. Применение нейронечеткого подхода для решения задач оценки почвенных разностей	326
7.4. Оценка плодородия почвы на основе нечеткой логической модели Сугено	333
Заключение	338
Список использованной литературы	339

## Введение

**Искусственный интеллект** - научное направление, в рамках которого ставятся и решаются задачи аппаратного или программного моделирования видов человеческой деятельности, традиционно считающихся интеллектуальными.

Искусственным интеллектом называют свойство интеллектуальных систем выполнять творческие функции, которые традиционно считаются прерогативой человека.

**Data Mining** - это процесс поддержки принятия решений, основанный на поиске в данных скрытых закономерностей.

**Технология Data Mining** - это процесс обнаружения в «сырых» данных ранее неизвестных, нетривиальных, практически полезных и доступных интерпретации знаний, необходимых для принятия решений в различных сферах человеческой деятельности.

Технология Data Mining предназначена для поиска в больших объемах данных неочевидных, объективных и полезных на практике закономерностей.

В основу технологии Data Mining положена концепция шаблонов (patterns), которые представляют собой закономерности, свойственные подвыборкам данных, которые могут быть выражены в форме, понятной человеку. Цель поиска закономерностей - представление данных в виде, отражающем искомые процессы. Построение моделей прогнозирования также является целью поиска закономерностей.

Средства Data Mining, в отличие от статистических, не требуют наличия строго определенного количества ретроспективных данных. Эта особенность может стать причиной обнаружения недостоверных, ложных моделей и, как результат, принятия на их основе неверных решений. Необходимо осуществлять контроль статистической значимости обнаруженных знаний.

Направления исследований:

1. Построение Интеллектуальных информационно-аналитических систем (Инт ИАС);

структурно-функциональная организация и методы построения подсистем Инт ИАС:

- развитие методов интеллектуального анализа данных и

знаний (DataMining -DM, Knowledge Discovery in Databases – KDD) для систематизации, структуризации и выявления внутренних взаимосвязей большого объема разнородных баз данных и знаний (БДиЗ);

- интеллектуализация процессов анализа и оценки текущих и прогнозируемых состояний исследуемых процессов;

- формирование возможных решений по управлению процессами в соответствии с полученными оценками их состояний.

2. Развитие и совершенствование базовых средств интеллектуальных Природных / Мягких вычислений (Natural / Soft Computing) вычислений Soft Computing (нечеткой логики, нейросетевых вычислений, эволюционных алгоритмов оптимизации) путем разработок методов и алгоритмов: прямых и обратных цепочек нечетких логических рассуждений; глубинных обучений и нейросетевых вычислений (Deep Learning and Networks); иммунных, роевых, ДНК, культурных и др. вычислений.

3. Развитие методов и алгоритмов интеллектуальных систем поддержки принятия решений (Инт СППР) с использованием усовершенствованных средств Soft Computing.

4. Интеллектуализация программного обеспечения (SoftWare) и средств автоматизации их разработки с использованием современных языков инженерии знаний: объектно-ориентированных и символьных языков программирования и интеллектуальных агентов, предоставляющих возможность представлять в ИАС слабоструктурированные модели функциональных задач; алгоритмы логического вывода, рассуждений и эволюции; модели представления вербальных, экспертных знаний (продукционных, фреймовых, семантических сетей). Интеллектуальные агенты позволяют принимать слабоструктурированные решения на основе механизмов логического вывода и методов обработки и управления знаниями (knowledge work and management methods) – совокупности процессов сбора, актуализации, хранения, продуцирования и распространения знаний. Следует при этом отметить актуальность совершенствования технологий автоматизации разработки интеллектуальных агентно-ориентированных программных средств путем развития существующих (CASE и др.).

5. Важной сферой интеллектуализации рассматриваемых ИАС

является обеспечение их информационной безопасности (ИБ). Одновременно с интенсивным развитием информационных технологий современный этап характеризуется и тенденцией появления новых видов информационных угроз, отличающихся неизвестными характеристиками и динамичностью их изменений. В таких условиях перспективным направлением повышения эффективности систем ИБ являются Интеллектуальные системы ИБ (Инт СИБ). Перспективными являются гибридные нечетко-нейронные экспертные системы (НН ЭС) ИБ, объединяющие интеллектуальные технологии экспертных систем, нейронных сетей, нечетких множеств (НМ) и нечеткого логического вывода (НЛВ), имуннокомпьютинга и эволюционных алгоритмов, предназначенных для идентификации новых появляющихся угроз и определения средств защиты от них.

Для построения Инт СИБ целесообразно использовать концепцию распределенных интеллектуальных мультиагентных систем, основанной на принципах иерархичности структуры, распределенности баз данных и знаний, децентрализация функций, координации взаимодействий интеллектуальных агентов различных уровней системы ИБ.

ИнтСИБ такого типа имеют иерархическую структуру, аналогичную биологическим системам. На нижних уровнях иерархии реализуются механизмы иммунной системы, а на верхних - механизмы адаптивной ассоциативной памяти и эволюции, т.е. обучения системы в процессе накопления «жизненного опыта».

Такие системы ИБ используют аналогии механизмов эволюции биологических систем, наделяющих их свойством самоорганизации и эволюции. Это обеспечивает более эффективное решение задач ИБ в условиях динамично изменяющейся внешней среды. Основными из них являются:

- 1) идентификация динамично изменяющихся угроз;
- 2) выявление механизмов защиты (МЗ) для оперативного реагирования и нейтрализации обнаруженных угроз;
- 3) идентификация и аутентификация пользователей;
- 4) мониторинг (интеллектуальный аудит) состояния ИБ компьютерных систем (КС);
- 5) настройка параметров нейронных сетей с целью адаптации системы ИБ к текущему состоянию внешней среды. Решение этих



задач средствами интеллектуальных технологий сводит до минимума воздействие атак с новыми неизвестными характеристиками за счет реализации механизмов жизнеобеспечения и информационной безопасности КС.

6. Рассмотренные интеллектуальные компоненты Инт ИАС являются достаточно сложными. Поэтому для их успешной реализации необходимо осуществлять разработки по совершенствованию аппаратных, вычислительных (программных), информационных и коммуникационных компонент современных ИАС. Одним из направлений решения этой проблемы является разработка высокопроизводительных распределенных вычислительных систем (РВС), представляющих собой совокупность территориально распределенных информационно-вычислительных ресурсных центров (РЦ), включающих аппаратные (hardware), программные (software), информационные (information) и коммуникационные ресурсы, функционирующие параллельно, взаимосвязано и координировано в соответствии с общим алгоритмом решения задач. Методология такой организации должна обеспечивать построение перестраиваемой смарт-архитектуры РВС, адаптирующейся к характеристикам текущего потока задач (расчетных, обработки графической информации, решение слабоструктурированных задач путем обработки знаний логических выводов и т.д.). Базовыми архитектурами для развития РВС являются многопроцессорные кластерные системы, объединяющие процессоры различных типов (многоядерные, графические и др.), грид-системы, клауд (облачные) системы (подкласс интернет-систем).

7. Концепция построения архитектуры и функционально-структурной организации ИнтИАС.

Рассмотренные характеристики задач, решаемых на верхних уровнях управления в условиях неопределенности внешней и внутренней среды, а также их непредсказуемых изменений обуславливают разработку методологии построения ИАС, основанной на принципах взаимного соответствия структурно-функциональных характеристик организаций и ИАС, использования ИнтИТ для решения предусмотренных функциональных задач, параллельности и поэтапной реализации их создания. Такая методология определяет следующие требования к

архитектуре и организационно-функциональным структурам современных ИнтИАС:

1) распределенность, иерархичность, интеллектуальность, мультиагентность и неоднородность программно-технических компонент;

2) интегрированность, единство информационного пространства для осуществления информационного менеджмента;

3) информационно-лингвистическая совместимость, защищенность корпоративной информации;

4) централизованность и децентрализованность управления;

5) масштабируемость задач – возможность решения задач различного функционального назначения;

6) масштабируемость пользователей -возможность работы для пользователей различных типов (менеджеров) различных уровней, функциональных исполнителей и др.).

Учет этих требований предполагает построение рассматриваемых ИнтИАС, состоящих из следующих основных функциональных подсистем: центрального и оперативного электронного информационного ресурса (ЭИР), аналитической обработки данных (ОАД) ЭИР, поддержки принятия решений (ППР), интеллектуального интерфейса. Центральный ЭИР является общей компонентой ИАС организации в целом. В ней хранятся предварительно обработанные, систематизированные и структурированные данные, полученные из внешних и внутренних источников. В оперативном ЭИР формируются и хранятся данные и знания, необходимые для решения задач ИАС стратегического уровня. Подсистема ОАД осуществляет оперативную обработку и интеллектуальный анализ данных и ситуаций и их оценку. Подсистема ППР формирует альтернативы возможных решений, соответствующих полученным оценкам ситуаций, оценку и ранжирование их в соответствии с принятыми критериям и формирует приемлемые решения для представления ЛПР с целью их анализа и принятия окончательного решения. Следует отметить, что Инт ИАС являются основной компонентой базовой платформы современной знание ориентированной цифровой экономики – Digital Knowledge Economy.

# Глава 1. ОБЩИЕ СВЕДЕНИЯ ОБ ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМАХ

## 1.1. Общие понятия интеллектуальных систем

Термин **Искусственный интеллект (ИИ)** – претенциозен, метафоричен.

Реальное содержание – повышение "интеллекта" ЭВМ; передача компьютеру некоторых функций человеческой интеллектуальной деятельности; создание помощника в решении интеллектуальных задач.

Более точно:

**Искусственный интеллект** – область исследований и прикладных разработок, направленных на создание программно-аппаратных средств, способных к решению таких задач, решение которых предполагает применение человеком своих интеллектуальных способностей.

Представлены три аспекта исследований в области ИИ:

- искусственный интеллект ↔ интеллект человека (факультет психологии);
- искусственный интеллект ↔ математический аппарат (мех-мат);
- искусственный интеллект ↔ программное обеспечение (ВМК).

Нас ИИ интересует именно в этом аспекте. Мы будем рассматривать проблемы ИИ в контексте создания программного обеспечения ЭВМ.

Традиционно основное внимание уделялось *точности результатов* работы вычислительных систем (ВС). Гораздо меньше внимания уделялось проблеме *удобства* работы с ВС.

Связанные понятия: **интеллектуальный интерфейс, дружественный интерфейс.**

**Дружественный интерфейс:**

- естественные языковые конструкции и структуры меню (не требуется знание синтаксиса формальных языков общения с компьютером);

- «интуитивный» уровень взаимодействия с компьютером, не требующий длительного обучения (для профессионала – несколько часов);
- разнообразные средства общения, пригодные для пользователей различного уровня подготовки (командный язык, меню, пиктограммы, диаграммы и др.);
- для каждого уровня пользователя адекватные возможности в: меню, запросах, подсистеме помощи;
- работа в реальном времени (необходимая скорость в диалоге);
- использование манипулятора типа «мышь», «горячих клавиш», сенсорного экрана и др.;
- минимальное использование клавиатуры;
- «интеллектуальные» средства (устойчивость к ошибкам, широкое использование принципа «по умолчанию»).

**Интеллектуальный интерфейс** – совокупность программных и аппаратных средств, позволяющая конечному пользователю решать на компьютере характерные для его повседневной деятельности задачи без помощи посредников-программистов.

Расширение взаимодействия между человеком и компьютером с помощью:

- увеличения диапазона способов ввода и вывода;
- обогащения грамматики ввода и вывода;
- попытки кооперации с пользователем в достижении целей.

В идеале система должна иметь "модель мира задачи", над которой работают система и пользователь и которая близка модели этого мира в уме пользователя.

Достаточно реальная (и близкая) перспектива – *речевой интерфейс*.

**Новые информационные технологии** (сейчас этот термин трактуется шире) – технологии, которые должны обеспечить возможность применения ЭВМ конечным пользователем в сфере его профессиональной деятельности без помощи посредника-программиста.

Прогресс в сфере экономики немыслим без применения современных информационных технологий, представляющих собой основу экономических информационных систем (ИС). ИС в экономике имеют дело с организацией и эффективной обработкой

больших массивов данных в компьютеризированных системах предприятий, обеспечивая информационную поддержку принятия решений менеджерами. Глобализация финансовых рынков, развитие средств электронной коммерции и формирование в Интернете доступных для анализа баз данных финансово-экономической информации, снижение стоимости программной реализации ИС привели к беспрецедентному росту их использования в экономике. ИС позволяют объективно оценить достигнутый уровень развития экономики, выявить резервы и обеспечить успех их деятельности на основе применения правильных решений.

Интеллектуальные информационные системы (ИИС) — естественный результат развития обычных информационных систем, сосредоточили в себе наиболее наукоемкие технологии с высоким уровнем автоматизации не только процессов подготовки информации для принятия решений, но и самих процессов выработки вариантов решений, опирающихся на полученные информационной системой данные. ИИС способны диагностировать состояние предприятия, оказывать помощь в антикризисном управлении, обеспечивать выбор оптимальных решений по стратегии развития предприятия и его инвестиционной деятельности. Благодаря наличию средств естественно-языкового интерфейса появляется возможность непосредственного применения ИИС бизнес-пользователем в качестве средств поддержки процессов анализа, оценки и принятия экономических решений. ИИС применяются для экономического анализа деятельности предприятия, стратегического планирования, инвестиционного анализа, оценки рисков и формирования портфеля ценных бумаг, финансового анализа, маркетинга и т.д.

Современная динамично изменяющаяся бизнес-среда требует профессионалов, способных в дополнение к экономическим знаниям применять современные информационные технологии, чтобы находить инновативные способы реализации бизнес-процессов.

Работы в области искусственного интеллекта в течение довольно длительного времени представлялись многим как причуды оторванных от реальности информатиков-интеллектуалов, обучающих компьютер игре в шахматы или распознаванию сцен,

или же пытающихся создать автономно ориентирующиеся в пространстве мобильные роботы. Механизмы, лежащие в основе таких программ и систем, объявлялись неалгоритмизуемыми, эвристическими, считались известными только посвященным, зачастую несли в себе аромат таинственности и волшебства.

Наибольший эффект от внедрения ИИС достигается там, где при принятии решений учитываются наряду с экономическими показателями слабо формализуемые факторы — экономические, политические, социальные. Так, в области экономического анализа и управления, менеджмента, антикризисного управления, стратегического планирования, инновационного менеджмента и инвестиционного анализа существует обширное поле деятельности для применения интеллектуальных технологий и систем. Поскольку в основе этих видов деятельности лежит проблема выбора решений.

По мере совершенствования принципов логического и правдоподобного вывода, применяемых в ИИС за счет использования нечеткой, модальной, временной логики, байесовских сетей вывода, ИИС начинают проникать в высокоинтеллектуальные области, связанные с разработкой стратегических решений по совершенствованию деятельности предприятий. Этому способствуют более современные алгоритмы анализа и синтеза предложений естественного языка, облегчающие общение пользователя с системой.

Включение в состав ИИС классических экономико-математических моделей, методов линейного, квадратичного и динамического программирования позволяет сочетать анализ объекта на основе экономических показателей с учетом факторов и рисков политических и внеэкономических факторов, оценивать последствия полученных их ИИС решений.

Наличие в составе ИИС объектно-ориентированной базы данных позволяет однородными средствами обеспечить хранение и актуализацию как фактов, так и знаний.

Проектирование ИИС как крупного программного комплекса как в отношении его жизненного цикла, так и в отношении технологии проектирования незначительно отличается от технологии проектирования ИС. Основная специфика связана с разработкой базы знаний.



Рис. 1.1. Классификация интеллектуальных информационных систем

ИИС можно классифицировать по разным основаниям. Мы выберем в качестве оснований классификации следующие: предметная область в экономике, степень автономности от корпоративной ИС или базы данных, по способу и оперативности взаимодействия с объектом, адаптивности, модели знаний (рисунок 1.1).

По степени интеграции ИИС могут быть: автономные в виде самостоятельных программных продуктов с собственной базой данных; сопрягаемые с корпоративной системой с помощью средств ODBC или OLE DB; полностью интегрированные. По оперативности принято различать динамические и статические ИИС. Предлагается различать ИИС реального времени с собственными сенсорами и эффекторами и советующие, в контур которых вовлечен пользователь.

По адаптивности различаются обучаемые ИИС типа нейронных сетей, т.е. системы, параметры, а возможно и структура которых могут изменяться в процессе обучения или самообучения, и ИИС, параметры которых изменяются администратором базы знаний. Наиболее часто используемые модели знаний приведены непосредственно на рисунке 1.1.

## 1.2. Цели и концепция интеллектуальной информационной системы

Реализация в модели объекта информатизации позволяет строить классическую схему управления по следующим этапам:

- планирование работ;
- сбор и анализ данных о происходящих процессах;
- анализ соответствия фактических результатов плановым показателям;
- разработка организационных, финансовых, маркетинговых и иных процедур, снижающих влияние неблагоприятных факторов: снижение рыночного спроса или изменения стоимости комплектующих изделий;
- адаптация дальнейших планов работ с учетом сложившихся условий.

При всей своей очевидности такая схема управления на практике не имеет универсальных решений. Она формируется с учетом специфики и масштаба бизнеса, существующего менеджмента, уровня детализации решаемых задач.

Выработка решений в виде стратегии функционирования и развития производится на основе миссии и целей предприятия с учетом доступных ресурсов и результатов обработки данных обратной связи от объекта управления. Высшее руководство, например, решает задачи определения целей и выработки стратегий развития, формирования и совершенствования организационной структуры, оценки позиций фирмы на рынке и поведения конкурентов, установления ассортимента выпускаемой продукции, организации деятельности аппарата управления компании в целом и т. п. Менеджеры среднего уровня принимают решения, связанные с календарным планированием производства, подбором и расстановкой кадров, реализацией инноваций, систем материального стимулирования и т.д.

Будем считать, что в основе деятельности менеджера-пользователя лежит процесс обнаружения, описания и разрешения проблемных ситуаций (ПрС). Возникающие в объекте управления ПрС находят свое отражение в базе данных в виде определенных значений атрибутов.



**Лицо, принимающее решение** — это субъект решения, т.е. руководитель или менеджер, наделенный надлежащими полномочиями и несущий ответственность за последствия принятого им и реализованного решения.

В своих действиях ЛПР опирается на собственные профессиональные навыки, прошлый опыт, интуицию. Однако при сложных и нечетко сформулированных задачах ЛПР не может использовать опыт, а опора только на интуицию увеличивает риск принятия неверного или неоптимального решения. В подобных ситуациях ЛПР вынужден привлекать к выработке решения экспертов — специалистов в разных областях знаний, для анализа проблемы и подготовки вариантов решений.

**Принятие решения** — это процесс выбора способа действий, уменьшающего расхождение между существующим (наблюдаемым) и желаемым (возможно, идеальным) состояниями организации. Процесс принятия решения состоит из упорядоченных определенным образом этапов (процедур), содержание которых описывается в терминах цели, проблемы, проблемной ситуации, альтернативы и самого решения как результата выбора альтернативы (варианта действий).

**Цель** — под этим понимают ожидаемое и желаемое состояние системы, в которое она должна перейти под действием управляющих воздействий и внутренних законов движения экономического объекта.

Различают стратегические и тактические цели. Стратегические цели носят более общий характер и рассчитаны на более длительный период времени, чем тактические.

**Проблема** — это расхождение между фактически наблюдаемым и желаемым или заданным состоянием управляемого объекта (организации). Проблема возникает, если:

- функционирование организации в данный момент не обеспечивает достижение стоящей перед нею цели;
- функционирование организации в будущем не гарантирует достижения поставленной цели;
- происходит пересмотр целей организации, вызванный, например, изменением общей макроэкономической ситуации, рыночной конъюнктуры и т.п.

**Проблемная ситуация** — это содержательное описание проблемы совместно с комплексом условий, факторов и обстоятельств, вызвавших её возникновение. Ситуационные факторы, породившие ту или иную проблему, можно подразделить на внутренние и внешние по отношению к организации (объекту управления).

Для каждой группы факторов имеются соответствующие модели анализа и оценок.

**Анализ проблемной ситуации** — это совместное рассмотрение проблемы в контексте вызвавших ее факторов.

Для краткости изложения будем именовать описывающие проблему признаки, события, оценки и показатели породивших ее причин факторами проблемы (проблемной ситуации). Факторы проблемы могут быть представлены численным значением, логическим высказыванием, лингвистической переменной или текстовым вербальным описанием.

С точки зрения точности формализации описания и последующего выбора решения различают проблемы:

- структурированные;
- слабоструктурированные;
- неструктурированные.

Проблема **структурированная**, если удастся представить все составляющие её элементы (признаки, проявления, причины, обстоятельства) и зависимости между ними в формализованной (аналитической или логической) форме.

Описание **слабоструктурированных** проблем возможно главным образом в виде качественных зависимостей между ее элементами, информация о части которых может отсутствовать. С точки зрения ЛПР, слабоструктурированные проблемы отличаются наличием неопределенностей как в характере зависимостей, так и в значениях их параметров.

**Неструктурированной** является проблема, для которой могут быть определены зависимости лишь между классами объектов и отношений, к которым они принадлежат.

**Решение** является реакцией организации на возникшую проблему: оно всегда принимается там, где возникает ПРС. С содержательной точки зрения, решение есть идентификатор программы или плана разрешения проблемной ситуации.

При интерпретации данных сложных слабоструктурированных проблем необходимо проанализировать результаты производственно-коммерческой деятельности предприятия, в рамках которого на основе отчетных, плановых и нормативных данных о его состоянии и функционировании, устанавливаются отклонения и причины их возникновения.

На стадии структурирования и диагностики проблемы необходимо привлечение специалистов различных служб и экспертов.

Процесс структуризации подразумевает решение следующих задач.

1. *Классификация* проблем по категориям (таксономия) — определение того, на что влияет существование той или иной проблемы или под влиянием каких сфер деятельности организации и внешних факторов они находятся.
2. *Идентификация* переменных или факторов, составляющих сущность проблемы, воздействующих на проблему или находящихся под ее влиянием.
3. *Оценка зависимостей между переменными* — как они воздействуют друг на друга.

Основной подход для решения задач первой группы — методология ситуационного анализа, т.е. разделение проблем и источников их возникновения на внешние и внутренние. Предлагается использовать в качестве признаков для таксономии проблем основные направления деятельности (функциональные области) конкретного предприятия, а также основные направления воздействия на него внешней среды.

Две другие задачи, которые приходится решать руководителю при структуризации проблемной ситуации, связаны с идентификацией факторов проблемы и определением существующих между ними взаимосвязей.

Логический анализ проблем-причин, показывает, что во многих случаях они позволяют сформулировать варианты решения проблем более высокого уровня.

### 1.3. Концептуальная схема структурно-функциональной организации интеллектуальной системы обработки информации

Концептуальная схема структурно-функциональной организации интеллектуальной системы обработки информации (ИСОИ) основана на использовании СИИ. Она состоит из трех комплексов программно-технических средств (см. рис.1.2). Первый комплекс - исполнительная система - представляет собой совокупность средств, выполняющих программы, спроектированные с позиций эффективного решения задач и имеет в ряде случаев проблемную ориентацию.



Рис.1.2. Концептуальная схема ИСОИ

Второй комплекс - совокупность средств интеллектуального интерфейса, имеющих гибкую структуру, которая обеспечивает возможность адаптации в широком спектре интересов конечных пользователей. Третьим комплексом средств, с помощью которых организуется взаимодействие первых двух, является база знаний, обеспечивающая использование вычислительными средствами первых двух комплексов целостной и независимой от обрабатываемых программ системы знаний о проблемной среде. Исполнительная система (ИС) объединяет всю совокупность средств, обеспечивающих выполнение сформированной программы. Интеллектуальный интерфейс - система программных и аппаратных средств, обеспечивающих для конечного пользователя использование компьютера для решения задач, которые возникают в среде его профессиональной деятельности либо без посредников, либо с незначительной их помощью. База знаний (БЗ) - занимает центральное положение по отношению к остальным компонентам вычислительной системы в целом, через БЗ осуществляется интеграция средств ВС, участвующих в решении задач.

Особенности базы знаний:

1. *Внутренняя интерпретируемость.* Каждая информационная единица должна иметь уникальное имя, по которому ИС находит ее, а также отвечает на запросы, в которых это имя упомянуто. Когда данные, хранящиеся в памяти, были лишены имен, то отсутствовала возможность их идентификации системой. Данные могла идентифицировать лишь программа, извлекающая их из памяти по указанию программиста, написавшего программу. Что скрывается за тем или иным двоичным кодом машинного слова, системе было неизвестно.

2. *Структурированность.* Информационные единицы должны обладать гибкой структурой.. Каждая информационная единица может быть включена в состав любой другой, и из каждой информационной единицы можно выделить некоторые составляющие ее информационные единицы. Другими словами, должна существовать возможность произвольного установления между отдельными информационными единицами отношений типа "часть - целое", "род - вид" или "элемент - класс".

3. *Связность.* В информационной базе между информационными единицами должна быть предусмотрена возможность установления связей различного типа. Прежде всего эти связи могут характеризовать отношения между информационными единицами.

4. *Семантическая метрика.* На множестве информационных единиц в некоторых случаях полезно задавать отношение, характеризующее ситуационную близость информационных единиц, т.е. силу ассоциативной связи между информационными единицами. Его можно было бы назвать *отношением релевантности* для информационных единиц. Такое отношение дает возможность выделять в информационной базе некоторые типовые ситуации (например, "покупка", "регулирование движения на перекрестке"). Отношение релевантности при работе с информационными единицами позволяет находить знания, близкие к уже найденным.

5. *Активность.* С момента появления ЭВМ и разделения используемых в ней информационных единиц на данные и команды создалась ситуация, при которой данные пассивны, а команды активны. Все процессы, протекающие в ЭВМ, инициируются командами, а данные используются этими командами лишь в случае необходимости. Для ИС эта ситуация не приемлема. Как и у человека, в ИС актуализации тех или иных действий способствуют знания, имеющиеся в системе. Таким образом, выполнение программ в ИС должно инициироваться текущим состоянием информационной базы. Появление в базе фактов или описаний событий, установление связей может стать источником активности системы.

Перечисленные пять особенностей информационных единиц определяют ту грань, за которой данные превращаются в знания, а базы данных перерастают в *базы знаний* (БЗ). Совокупность средств, обеспечивающих работу с знаниями, образует *систему управления базой знаний* (СУБЗ). В настоящее время не существует баз знаний, в которых в полной мере были бы реализованы внутренняя интерпретируемость, структуризация, связность, введена семантическая мера и обеспечена активность знаний.

Системы поддержки принятия решений (СППР), ориентированных на принятие слабоструктурированных решений в

условиях неопределенности (СППР-СС), относятся к классу интеллектуальных систем. Основное место в базе знаний и операционной подсистеме СППР-СС занимают модели, правила и механизмы систем нечеткого вывода (СНВ).

Модели, правила и механизмы систем нечеткого вывода (СНВ) занимают основное место в базе данных и знаний, а также моделях механизма логического вывода и имитационного моделирования операционной подсистемы интеллектуальных системы поддержки принятия решений (СППР), ориентированных на поддержку принятия слабоструктурированных решений в условиях неопределенности.

При исследовании концептуальных моделей сложных ИСОИ перспективным является представление их в виде формализованной иерархии  $0 \subset K_n \subset K \subset \Pi$  - взаимосвязанных множеств моделей объектов, классов объектов, компонент и пакетов. Согласно подходу, изложенному в [2], под компонентой (К), с точки зрения концептуальных моделей систем, понимается совокупность моделей классов объектов ( $K_n$ ) которая отражает группу однородных методов и данных соответствующей однородной группы задач исследуемой предметной области (ПО). Пакеты (Π) - это совокупность одной из групп однородных компонент. Для взаимодействия К и Π между собой они, также как и  $K_n$ , наделяются соответствующими интерфейсами.

Предлагаемая концепция структурно-функциональной организации СНВ, как основной функциональной компоненты иерархической модели ИСОИ указанного типа, предполагает использование следующих положений и принципов:

1) систематизированное описание, на выбранном языке формализма, и упорядоченное представление основных понятий, парадигм, моделей, методов и законов, описывающих исследуемые слабоформализуемые процессы и связанные с ними задачи;

2) формализованное представление типовых оболочек моделей задач СНВ в виде конструкций правил и механизмов нечеткого вывода;

3) упорядоченное представление правил построения, модификации, актуализации, пополнения и проверки корректности моделей ПП и СНВ;

4) систематизированное представление типовых алгоритмов и программных модулей, реализующих типовые процедуры формирования моделей ПП и СНВ, а также решения задач СНВ и СППР-СС;

5) модульная организация структуры СНВ, обеспечивающих автоматизацию соответствующих процедур СНВ и СППР-СС;

6) интерактивное взаимодействие СНВ с лицом, принимающим решение (ЛПР), на этапах постановки целей и условий решаемых задач, а также анализа текущих результатов и их корректировки;

Первое положение концепции предусматривает использование алфавита и грамматики языков формализма, используемых для описания основных понятийно – процедурных объектов исследуемых слабоформализуемых процессов и задач принятия слабоструктурируемых решений при мониторинге и управлении ими. К числу таких объектов относятся базовые понятия, модели и методы нечеткой логики и нечетких множеств, нечетких графов и сетей Петри, нечеткого вывода и принятия решений, теоретико-множественных моделей нечетких ситуаций и закономерностей динамики их изменения под воздействием различных воздействий, моделей и методов оценки нечетких ситуаций и прогнозирования динамики их развития, закономерностей и методов принятия нечетких решений и нечеткого управления.

В системах нечеткого вывода, принятия решений и управления распространение получили языки типа FCL (Fuzzy Control Language) [5]. Нотации этого языка совместно с нотациями унифицированного языка моделирования UML (Unified Modeling Language) [6] могут быть с успехом использованы при разработке СНВ.

Второе положение определяет формализованное описание оболочек моделей операций нечеткого вывода (импликации, композиции и основанных на них механизмов) и продукционной системы (например, табличных моделей ПП [1]) в виде конструкций с использованием алфавита и грамматики языков пакетов нечеткой логики типа Fuzzy Logic Tools среды MATLAB, а также языковых средств программы нечеткого моделирования Fuzzy TECH [5]. Последние также используются для



моделирования и решения задач СНВ (Fuzzy Inference Systems-FIS), которые в среде Fuzzy TECH названы нечетко-логическими системами - НЛС (Fuzzy Logic Systems - FLS).

Третье положение предусматривает наделение СНВ свойствами адаптации и решения возможностей решения задач при возникновении изменений в условиях, ситуациях и целях (появление новых и удаление некоторых устаревших типов и данных о них). Для этого в СНВ должны быть представлены на соответствующем языке формализма правила построения моделей продукционных систем, отражающих конкретные значения состояния, цели и задачи СНВ, а также правила их модификации, пополнения, актуализации и проверки их адекватности при возникновении соответствующих изменений в исследуемой предметной области.

В соответствии с четвертым положением в создаваемой СНВ формируются конструкции оболочек типовых, апробированных и используемых в задачах нечеткого моделирования и вывода, алгоритмов и программных модулей. Путем объединения их в соответствующую последовательность синтезируются алгоритмы и программы решения конкретных задач нечеткого вывода и принятия решений.

Как и всякая система СНВ также должна состоять из взаимосвязанных и объединенных системной целью функциональных подсистем (компонент, модулей). Этому соответствует пятое положение.

Шестое предусматривает интерактивность взаимодействий ЛПР с СНВ на начальном этапе и во время анализа текущих результатов.

## 1.4. История развития интеллектуальных систем

История работ в области Искусственного интеллекта:

"романтический период" ИИ → серьезные научные исследования → практические задачи.

**50-е – 60-е гг. XX века** – "романтический период" ИИ: "машинные стихи", "машинная музыка", машинный перевод, интеллектуальные игры (шашки, шахматы и др.);

**60-е – 70-е гг. XX века** – исследование методов решения задач (методов поиска решения);

**70-е – 80-е гг. XX века** – исследование методов представления знаний нужных для решения задач;

**80-е – 90-е гг. XX века** – исследование методов приобретения знаний (передачи их от человека ЭВМ);

**90-е гг. XX века** – наше время – теоретическое осмысление, поиск новых идей и задач, попытки практического использования.

**Постоянные проблемы:**

Начальный уровень "знаний" системы ИИ, проблема ее обучения человеком и ее самообучения.

Общение человека с системой ИИ (языки общения, программно-аппаратные средства).

Инструментальные средства – языки программирования для задач ИИ.

**Типы систем ИИ (в историческом аспекте):**

*Решатели задач;*

*Роботы (и Встраиваемые Интеллектуальные системы);*

*Экспертные системы;*

*Интеллектуальные Агенты.*

Системы ИИ должны быть способны помогать и на этапе содержательной постановки задачи, уточнения и необходимого пополнения содержательной постановки и ее формализации.

## *Тест Тьюринга.*

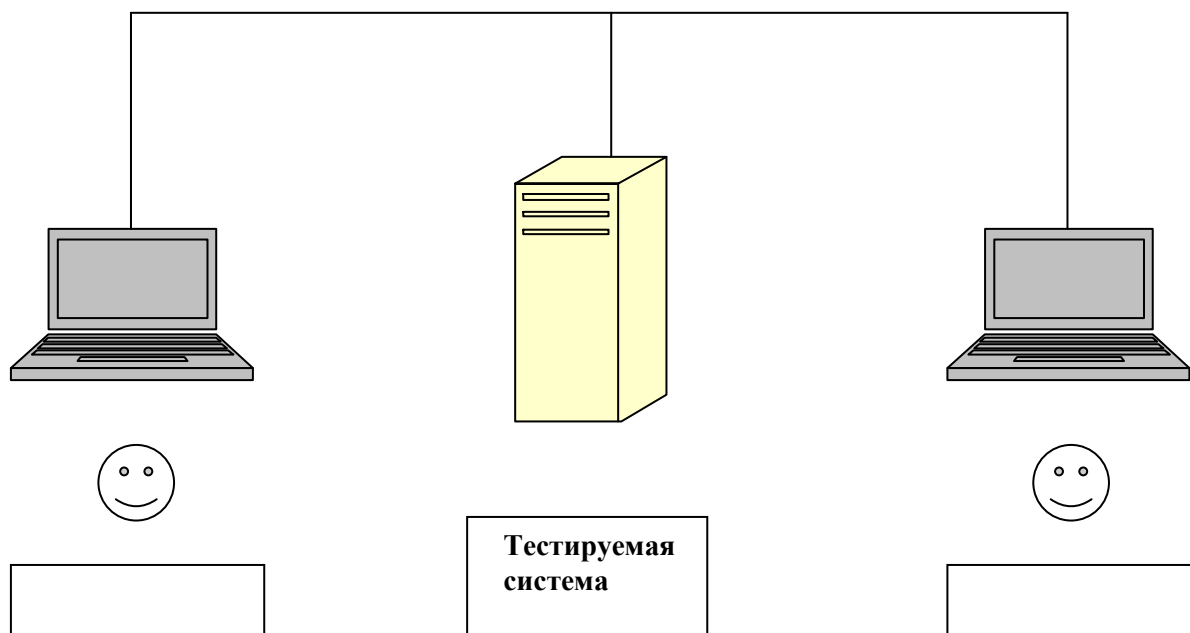


Рис.1.3. Схема теста Тьюринга

Автор ТЕСТА – один из основоположников кибернетики и ИИ Алан Тьюринг (США). Тест впервые был описан в Журнале *Mind* в 1950 году.

За терминалом работает *Следователь*. Его терминал связан с терминалом, за которым работает *Имитатор*, и с компьютером, на котором установлена *Тестируемая система (Система ИИ)*. Следователь обращается к своему «собеседнику» с вопросами, предлагает решить задачи. Кто отвечает ему (Имитатор или Система ИИ), он не знает. Выбирается «отвечающий» по датчику случайных чисел. Если в течение достаточно длительного времени Следователь не может отличить ответы человека (Имитатора) от ответов машины (Системы ИИ), то машину «можно считать разумной».

Несмотря на условность и неформальность теста Тьюринга, он - дает объективное понятие об интеллекте (задан стандарт для определения разумности / интеллектуальности);

- позволяет оставаться на функциональном уровне (не нужно знать, какие механизмы использует Система ИИ);
- может использоваться для тестирования / аттестации систем ИИ.

**Система Искусственного интеллекта (Интеллектуальная Система = ИС)** – программно-аппаратный комплекс, способный к решению таких задач, решая которые человек использует свои интеллектуальные способности.

Создание ИС – процесс сложный, он предполагает моделирование интеллектуальной деятельности человека (а мы знаем, что она чрезвычайно сложна и слабо изучена);

приходится много экспериментировать, создавать все новые и новые варианты, версии (иногда очень сильно отличающиеся друг от друга);

нужны языки высокого/сверхвысокого уровня для быстрого прототипирования.

**Особенности задач ИИ (с точки зрения программирования):**

1. сложные и динамически меняющиеся структуры данных;
2. большие по объему хранилища данных (базы знаний) и средства эффективной работы с ними;
3. символьные (в основном) данные;
4. модели, отражающие состояние проблемной среды;
5. переборные алгоритмы;
6. алгоритмы поиска по образцу;
7. гибкие структуры управления.

## **Представление задач в пространстве состояний**

### *Основные понятия*

Типичным представителем класса задач, для которых подходит представление в пространстве состояний, является головоломка, известная как игра в пятнадцать – см. рис. 1.4(а). В ней используется пятнадцать пронумерованных (от 1 до 15) подвижных фишек, расположенных в клетках квадрата 4×4. Одна клетка этого квадрата остается всегда пустой, так что одну из соседних с ней фишек можно передвинуть на место этой пустой клетки, изменив тем самым местоположение пустой клетки. Заметим, что более простым вариантом этой головоломки является

квадрат  $3 \times 3$  и восемь фишек на нем – пример соответствующей задачи показан на рис.1.4(б).

На рис.1.4(а) изображены две конфигурации фишек. В головоломке требуется преобразовать первую, или начальную, конфигурацию во вторую, или целевую конфигурацию. Решением этой задачи будет подходящая последовательность сдвигов фишек, например: передвинуть фишку 8 вверх, фишку 6 влево и т.д.

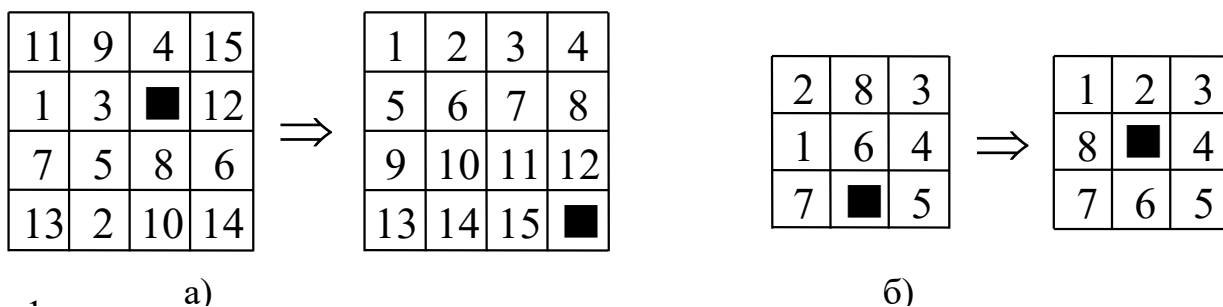


Рис.1.4. Конфигурации фишек

Важной особенностью класса задач, к которому принадлежит рассмотренная головоломка, относится наличие в задаче точно определенной начальной ситуации и точно определенной цели. Имеется также некоторое множество операций, или ходов, переводящих одну конфигурацию в другую. Именно из таких ходов состоит искомое решение задачи, которое можно в принципе получить методом проб и ошибок. Действительно, отправляясь от начальной ситуации, можно построить конфигурации, возникающие в результате выполнения возможных в этой ситуации ходов, затем построить множество конфигураций, получающихся после применения следующего хода, и так далее – пока не будет достигнута целевая конфигурация.

Введем теперь основные понятия, используемые при формализации задачи в пространстве состояний. Центральным из них является понятие *состояния*, характеризующего некоторый момент решения задачи. Например, для игры в пятнадцать (или в восемь) состояние – это просто некоторая конкретная конфигурация фишек.

Среди всех состояний задачи выделяются *начальное состояние* и *целевое состояние*, в совокупности определяющие задачу, которую надо решить – примеры их приведены на рис.1.4.

Другим важным понятием является понятие *оператора*, т.е. допустимого хода в задаче. Оператор преобразует одно состояние в другое, являясь по сути функцией, определенной на множестве состояний и принимающей значения из этого множества. Для игры в пятнадцать или в восемь удобнее выделить четыре оператора, соответствующие перемещениям пустой клетки (можно считать ее фишкой-«пустышкой») влево, вправо, вверх, вниз. В некоторых случаях оператор может оказаться неприменимым к какому-то состоянию: например, операторы сдвига вправо и вниз неприменимы, если пустая клетка расположена в правом нижнем углу. Значит, в общем случае оператор является частично определенной функцией отображения состояний.

В терминах состояний и операторов *решение задачи* есть определенная последовательность операторов, преобразующая начальное состояние в целевое. Решение задачи ищется в *пространстве состояний* – множестве всех состояний, достижимых из начального состояния при помощи заданных операторов. Например, в игре в пятнадцать или в восемь пространство состояний состоит из всех конфигураций фишек, которые могут быть образованы в результате возможных перемещений фишек.

Пространство состояний можно представить в виде направленного графа, вершины которого соответствуют состояниям, а дуги (ребра) – применяемым операторам. Указанные в виде стрелок направления соответствуют движению от вершины-аргумента применяемого оператора к результирующей вершине. Тогда решением задачи будет путь в этом графе, ведущий от начального состояния к целевому. На рис.2 показана часть пространства состояний для игры в пятнадцать. Каждая вершина соответствует некоторой конфигурации фишек. Все дуги между вершинами являются двунаправленными, поскольку в этой головоломке для любого оператора есть обратный ему (точнее, множество операторов состоит из двух пар взаимно-обратных операторов: влево-вправо, вверх-вниз).

Пространства состояний могут быть большими и даже бесконечными, но в любом случае предполагается счетность множества состояний.

Таким образом, в подходе к решению задачи с использованием пространства состояний задача рассматривается как тройка  $(S_I, O, S_G)$ , где

$S_I$  – начальное состояние;

$O$  – конечное множество операторов, действующих на не более чем счетном множестве состояний;

$S_G$  – целевое состояние.

Дальнейшая формализация решения задачи с использованием пространства состояний предполагает выбор некоторой конкретной *формы* описания состояний задачи. Для этого могут применяться любые подходящие структуры – строки, массивы, списки, деревья и т.п. Например, для игры в пятнадцать или восемь наиболее естественной формой описания состояния будет список положений фишек или же двумерный массив. Заметим, что от выбора формы описания состояния зависит в общем случае сложность задания операторов задачи, которые также должны быть определены при формализации задачи.

В рассмотренных выше примерах (игры в пятнадцать и восемь) искомое целевое состояние задавалось явно, т.е. известно было местоположение каждой фишки в целевой конфигурации. В более сложных случаях игры может быть несколько целевых состояний, либо же целевое состояние может быть определено неявно, т.е. охарактеризовано некоторым свойством, например, как состояние, в котором сумма номеров фишек в верхнем ряду не превосходит 10. В подобных случаях свойство, которому должно удовлетворять целевое состояние, должно быть описано исчерпывающим образом, к примеру, путем задания булевой функции, реализующей проверку нужного свойства состояния задачи.

Итак, для представления задачи в пространстве состояний необходимо определить следующее:

- форму описания состояний задачи и описание начального состояния;
- множество операторов и их воздействий на описания состояний;
- множество целевых состояний или же описание их свойств.

Перечисленные составляющие задают неявно граф-

пространство состояний, в котором требуется провести поиск решения задачи. Заметим попутно, что в отличие от такого неявного способа задания графа, при явном способе задания все вершины и дуги графа должны быть перечислены, например, с помощью таблиц.

Решение задачи в пространстве состояний подразумевает просмотр неявно заданного графа, для чего необходимо преобразование в явную форму достаточно большой его части, включающей искомую целевую вершину. Действительно, просмотр осуществляется как последовательный *поиск*, или *перебор* вершин, в *пространстве состояний*. В исходной точке процесса к начальному состоянию применяется тот или иной оператор и строится новая вершина-состояние, а также связывающие ее с корневой вершиной дуги. На каждом последующем шаге поиска к одной из уже полученных вершин-состояний применяется допустимый оператор и строится еще одна вершина графа и связывающие дуги. Этот процесс поиска продолжается до тех пор, пока не будет построена вершина, соответствующая целевому состоянию.

Рассмотрим теперь широко известную *задачу об обезьяне и банане* (в максимально упрощенной формулировке). В комнате находятся обезьяна, ящик и связка бананов, которая подвешена к потолку настолько высоко, что обезьяна может до нее дотянуться, только встав на ящик. Нужно найти последовательность действий, которая позволит обезьяне достать бананы. Предполагается, что обезьяна может ходить по комнате, двигать по полу ящик, взбираться на него и хватать бананы.

Ясно, что описание состояния этой задачи должно включать следующие сведения: местоположение обезьяны в комнате – в горизонтальной плоскости пола и по вертикали (т.е. на полу она или на ящике), местоположение ящика на полу и наличие у обезьяны бананов. Все это можно представить в виде четырехэлементного списка (*ПолОб*, *ВертОб*, *ПолЯщ*, *Цель*), где

*ПолОб* – положение обезьяны на полу (это может быть двухэлементный вектор координат);

*ПолЯщ* – положение обезьяны и ящика на полу;

*ВертОб* – это константа П или Я в зависимости от того, где находится обезьяна, на полу или на ящике;



*Цель* – это константа 0 или 1 в зависимости от того, достала ли обезьяна бананы или нет.

Зафиксируем также как константы три следующие точки в плоскости пола:

$T_O$  – точка первоначального местоположения обезьяны;

$T_Я$  – точка первоначального расположения ящика;

$T_Б$  – точка пола, расположенная непосредственно под связкой бананов.

Тогда начальное состояние задачи описывается списком  $(T_O, П, T_Я, 0)$ , а целевое состояние задается как любой список, последний элемент которого – 1.

Естественно определить операторы в этой задаче в соответствии с возможными действиями обезьяны:

1) *Перейти* ( $W$ ) – переход обезьяны к точке  $W$  горизонтальной плоскости пола;

2) *Передвинуть* ( $V$ ) – передвижение обезьяной ящика в точку  $V$  пола;

3) *Взобраться* – обезьяна взбирается на ящик;

4) *Схватить* – обезьяна хватает связку бананов.

Условия применимости и действие этих операторов легко определить в виде правил продукций вида: *аргумент оператора*  $\rightarrow$  *результат оператора*, причем

$X, Y, Z, W, V$  обозначают переменные:

1. *Перейти* ( $W$ ) :  $(X, П, Y, Z) \rightarrow (W, П, Y, Z)$

2. *Передвинуть* ( $V$ ) :  $(X, П, X, Z) \rightarrow (V, П, V, Z)$

3. *Взобраться* :  $(X, П, X, Z) \rightarrow (X, Я, X, Z)$

4. *Схватить*:  $(T_Б, Я, T_Б, 0) \rightarrow (T_Б, Я, T_Б, 1)$

Будем считать, что для решения задачи значимы лишь вышеупомянутые точки пола  $T_O, T_Я, T_Б$ , тогда получим пространство состояний задачи, изображенное на рис. 1.5. Это пространство содержит только 13 состояний, дуги графа-пространства промаркированы порядковым номером применяемого оператора. Пространство содержит четыре цикла хождения обезьяны между тремя значимыми точками (с ящиком или без него). В пространстве есть также две тупиковые ветви – когда обезьяна залезает на ящик, но не под связкой бананов. Жирными дугами (стрелками) показан решающий путь, состоящий из четырех

операторов: *Перейти* ( $T_я$ ); *Передвинуть*( $T_б$ ); *Взобраться*; *Схватить*.

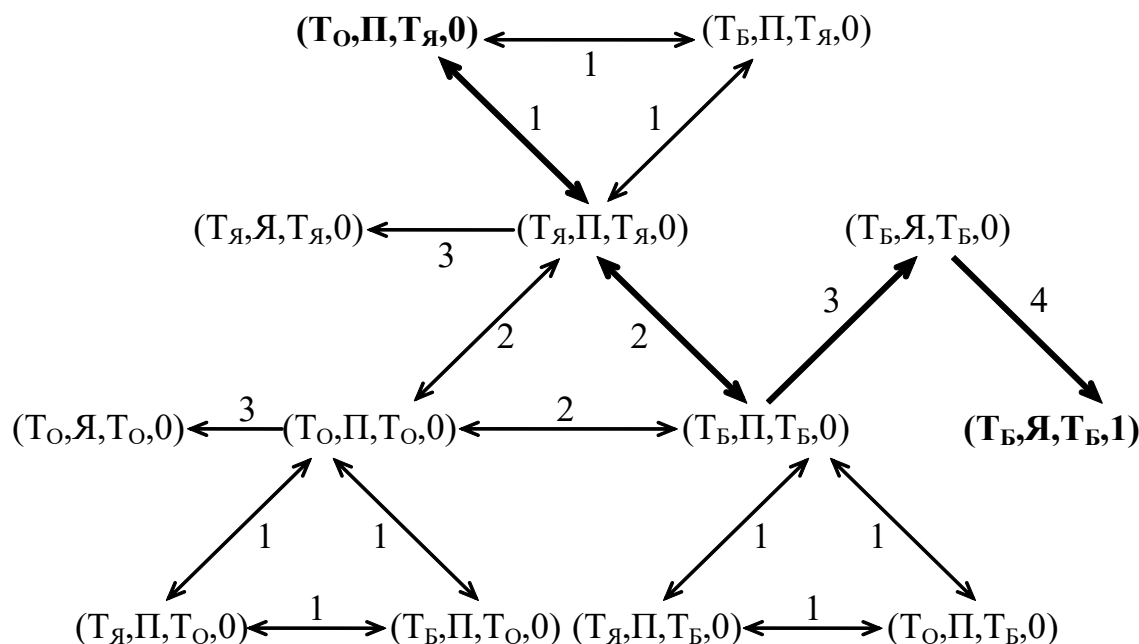


Рис.1.5. Пространство состояний задачи

Естественный язык в работах по искусственному интеллекту. Первые шаги.

В 60-е – 70-е гг. XX века специалистов в области искусственного интеллекта интересовала прежде всего проблема моделирования понимания сообщений/текстов. Были реализованы системы, способные «поддерживать диалог» с человеком на естественном языке, интерпретировать словесные команды роботу (действующему в «Мире кубиков»), решать задачи в словесной формулировке.

Система ELIZA (DOCTOR) – Дж. Вейзенбаум (1966)

Система относится к классу Conversation Machines, разработчики которых надеялись на то, что их творения смогут пройти *Тест Тьюринга*.

Версия DOCTOR моделирует диалог врача-психоаналитика (система) с пациентом (человек).

Фрагмент диалога с системой ELIZA:

USER: MY MOTHER TAKES CARE OF ME

ELIZA: WHO ELSE IN YOUR FAMILY TAKES CARE OF YOU

USER: MY FATHER  
ELIZA: YOUR FATHER  
USER: YOU ARE LIKE MY FATHER IN SOME WAYS  
ELIZA: WHAT RESEMBLANCE DO YOU SEE  
USER: YOU ARE NOT VERY AGGRESSIVE  
ELIZA: WHAT MAKES YOU THINK I AM NOT VERY  
AGGRESSIVE

Для формирования ответов система использует простейшие шаблоны и связанные с ними правила, например:

(<any-1> you <any-3> me) → (what makes you think I <any-3> you)

USER: IT SEEMS THAT YOU HATE ME  
ELIZA: WHAT MAKES YOU THINK I HATE YOU  
(<any-1> you are <any-3>) -> (what makes you think I am <any-3>)

USER: YOU ARE NOT VERY AGGRESSIVE  
ELIZA: WHAT MAKES YOU THINK I AM NOT VERY  
AGGRESSIVE

Система APRIL решает арифметические задачи в словесной формулировке. Результатом ее работы является ответ (формула или число), иногда сопровождаемый некоторыми комментариями, или сообщение (также сопровождаемое комментариями) о том, что задачу решить не удалось.

Система решает довольно узкий круг задач из школьного курса математики (1-2-й классы) – так называемые задачи «на владение». Каждая задача решается независимо от других.

На входной язык системы наложены достаточно жесткие ограничения, которые, впрочем, абсолютно естественны для рассматриваемого класса задач. Так, условие задачи должно быть сформулировано в виде трех простых предложений, а все слова, встречающиеся в тексте условия, должны быть знакомы системе (записаны в ее словарь).

Наиболее серьезное внимание было уделено разработке методов, позволяющих учитывать возможность варьирования словесных формулировок условий задач и употребления в тексте неполных (эллиптических) конструкций. Так, внешне непохожие задачи № 1, № 2 и № 3 система сводит к одному типу задач, а решая задачи, которые человек воспринимает как имеющие решение и притом единственное, несмотря на отсутствие

подразумеваемых уточняющих слов, она делает примерно те же предположения, что и человек.

Задача N 1: У А 5 яблок. У В 10 груш. Сколько фруктов у ребят?

ОТВЕТ: 15

Задача N 2: У человека 10 книг. У второго человека в 2 раза меньше. Сколько книг у них вместе?

ОТВЕТ: 15

Задача N 3: На стол положили 6 книг. Затем еще 4 книги. Сколько книг на столе?

ЕСЛИ ВО ВСЕХ ФРАЗАХ ИМЕЕТСЯ В ВИДУ "СТОЛ", ТО  
ЕСЛИ В НАЧАЛЬНЫЙ МОМЕНТ КОЛИЧЕСТВО  
ОБЪЕКТОВ "КНИГА" В МЕСТЕ "СТОЛ" = 0, ТО  
ПРЕДПОЛОЖИТЕЛЬНЫЙ ОТВЕТ: 10

Первые эксперименты показали и принципиальную возможность моделирования процессов владения языком, и чрезвычайную сложность естественного языка как объекта исследования.

ВІ объединяет в себе различные средства и технологии анализа и обработки данных масштаба предприятия.

На основе этих средств создаются ВІ-системы, цель которых - повысить качество информации для принятия управленческих решений.

ВІ-системы также известны под названием Систем Поддержки Принятия Решений (СППР, DSS, Decision Support System). Эти системы превращают данные в информацию, на основе которой можно принимать решения, т.е. поддерживающую принятие решений.

Gartner Group определяет состав рынка систем Business Intelligence как набор программных продуктов следующих классов:  
средства построения хранилищ данных (data warehousing, ХД);  
системы оперативной аналитической обработки (OLAP);  
информационно-аналитические системы (Enterprise Information Systems, EIS);  
средства интеллектуального анализа данных (data mining);  
инструменты для выполнения запросов и построения отчетов (query and reporting tools).

Классификация Gartner базируется на методе функциональных задач, где программные продукты каждого класса выполняют определенный набор функций или операций с использованием специальных технологий.

Мнение экспертов о Data Mining.

Приведем несколько кратких цитат [4] наиболее влиятельных членов бизнес-сообществ, которые являются экспертами в этой относительно новой технологии.

Руководство по приобретению продуктов Data Mining (Enterprise Data Mining Buying Guide) компании Aberdeen Group: "Data Mining - технология добычи полезной информации из баз данных. Однако в связи с существенными различиями между инструментами, опытом и финансовым состоянием поставщиков продуктов, предприятиям необходимо тщательно оценивать предполагаемых разработчиков Data Mining и партнеров.

Чтобы максимально использовать мощность масштабируемых инструментов Data Mining коммерческого уровня, предприятию необходимо выбрать, очистить и преобразовать данные, иногда интегрировать информацию, добытую из внешних источников, и установить специальную среду для работы Data Mining алгоритмов.

Результаты Data Mining в большой мере зависят от уровня подготовки данных, а не от "чудесных возможностей" некоего алгоритма или набора алгоритмов. Около 75% работы над Data Mining состоит в сборе данных, который совершается еще до того, как запускаются сами инструменты. Неграмотно применив некоторые инструменты, предприятие может бессмысленно растратить свой потенциал, а иногда и миллионы долларов".

Мнение Херба Эдельштайна (Herb Edelstein), известного в мире эксперта в области Data Mining, Хранилищ данных и CRM: "Недавнее исследование компании Two Crows показало, что Data Mining находится все еще на ранней стадии развития. Многие организации интересуются этой технологией, но лишь некоторые активно внедряют такие проекты. Удалось выяснить еще один важный момент: процесс реализации Data Mining на практике оказывается более сложным, чем ожидается.

IT-команды увлеклись мифом о том, что средства Data Mining просты в использовании. Предполагается, что достаточно запустить такой инструмент на терабайтной базе данных, и моментально

появится полезная информация. На самом деле, успешный Data Mining-проект требует понимания сути деятельности, знания данных и инструментов, а также процесса анализа данных".

Прежде чем использовать технологию Data Mining, необходимо тщательно проанализировать ее проблемы, ограничения и критические вопросы, с ней связанные, а также понять, чего эта технология не может.

*Data Mining не может заменить аналитика.*

Технология не может дать ответы на те вопросы, которые не были заданы. Она не может заменить аналитика, а всего лишь дает ему мощный инструмент для облегчения и улучшения его работы.

*Сложность разработки и эксплуатации приложения Data Mining.*

Поскольку данная технология является мультидисциплинарной областью, для разработки приложения, включающего Data Mining, необходимо задействовать специалистов из разных областей, а также обеспечить их качественное взаимодействие.

*Квалификация пользователя.*

Различные инструменты Data Mining имеют различную степень "дружелюбности" интерфейса и требуют определенной квалификации пользователя. Поэтому программное обеспечение должно соответствовать уровню подготовки пользователя. Использование Data Mining должно быть неразрывно связано с повышением квалификации пользователя. Однако специалистов по Data Mining, которые бы хорошо разбирались в бизнесе, пока еще мало.

*Извлечение полезных сведений невозможно без хорошего понимания сути данных.*

Необходим тщательный выбор модели и интерпретация зависимостей или шаблонов, которые обнаружены. Поэтому работа с такими средствами требует тесного сотрудничества между экспертом в предметной области и специалистом по инструментам Data Mining. Построенные модели должны быть грамотно интегрированы в бизнес-процессы для возможности оценки и обновления моделей. В последнее время системы Data Mining поставляются как часть технологии хранилищ данных.

### *Сложность подготовки данных.*

Успешный анализ требует качественной предобработки данных. По утверждению аналитиков и пользователей баз данных, процесс предобработки может занять до 80% процентов всего Data Mining-процесса.

Таким образом, чтобы технология работала на себя, потребуется много усилий и времени, которые уходят на предварительный анализ данных, выбор модели и ее корректировку.

*Большой процент ложных, недостоверных или бессмысленных результатов.*

С помощью Data Mining можно отыскивать действительно очень ценную информацию, которая вскоре даст большие дивиденды в виде финансовой и конкурентной выгоды.

Однако Data Mining достаточно часто делает множество ложных и не имеющих смысла открытий. Многие специалисты утверждают, что Data Mining-средства могут выдавать огромное количество статистически недостоверных результатов. Чтобы этого избежать, необходима проверка адекватности полученных моделей на тестовых данных.

### *Высокая стоимость.*

Качественная Data Mining-программа может стоить достаточно дорого для компании. Вариантом служит приобретение уже готового решения с предварительной проверкой его использования, например на демо-версии с небольшой выборкой данных.

*Наличие достаточного количества репрезентативных данных.*

Средства Data Mining, в отличие от статистических, теоретически не требуют наличия строго определенного количества ретроспективных данных. Эта особенность может стать причиной обнаружения недостоверных, ложных моделей и, как результат, принятия на их основе неверных решений. Необходимо осуществлять контроль статистической значимости обнаруженных знаний.

## 1.5. Классификация интеллектуальных систем

### 1.5.1. Проблемы представления и моделирования знаний

Важное место в теории искусственного интеллекта занимает проблема представления знаний.

*Знания* представляют собой совокупность сведений (у индивидуума, общества или у системы ИИ) о мире (конкретной предметной области, совокупности объектов или объекта), включающих в себя информацию о свойствах объектов, закономерностях процессов и явлений, правилах использования этой информации для принятия решений.

Первоначально вычислительная техника была ориентирована на обработку данных. Это было связано как с уровнем развития техники и программного обеспечения, так и со спецификой решаемых задач. Дальнейшее усложнение решаемых задач, их интеллектуализация, развитие вычислительной техники ставят задачу создания машин обработки знаний. *Существенным отличием знаний от данных является их интерпретируемость.* Если для интерпретации данных необходимы соответствующие программы и сами по себе они не несут содержательной информации, то знания всегда содержательны. Другой отличительной чертой знаний является наличие отношений, например, вида «тип — подтип», «элемент—множество» и т.д. Знания характеризуются наличием ситуативных связей, определяющих ситуативную совместимость отдельных событий и фактов, позволяющих устанавливать причинно-следственные связи.

Типы знаний, которые представляются в системах ИИ, охватывают следующее:

- структуру, форму, свойства, функции и возможные состояния объекта;
- возможные отношения между объектами, возможные события, в которых эти объекты могут участвовать;
- физические законы;
- возможные эффекты действий и состояний, причины и условия возникновения событий и состояний;
- возможные намерения, цели, планы, соглашения и т.д.



В общем виде модели представления знаний могут быть условно разделены на концептуальные и эмпирические.

**Концептуальная модель** дает эвристический метод для решения некоторой проблемы. Метод эвристичен, поскольку концептуальное описание не дает гарантии того, что он может быть применен во всех соответствующих практических ситуациях. Концептуальная модель делает возможным распознавание проблемы, позволяет уменьшать время для ее предварительного анализа.

**Практическое использование концептуальной модели влечет за собой необходимость преобразования ее в эмпирическую.** Знания могут быть накоплены в виде эмпирических моделей, как правило, описательного характера. Эти модели могут варьировать от простого набора правил до полного описания того, как ЛПР решает задачу.

Модели представления знаний можно условно разделить на *декларативные* и *процедурные*.

**Декларативная модель** представления знаний основывается на предположении, что проблема представления некоей предметной области решается независимо от того, как эти знания потом будут использоваться. Поэтому модель как бы состоит из двух частей: статических описательных структур знаний и механизма вывода, оперирующего этими структурами и практически независимого от их содержательного наполнения. При этом в какой-то степени оказываются раздельными синтаксические и семантические аспекты знания, что является определенным достоинством указанных форм представления из-за возможности достижения их определенной универсальности.

**Главное преимущество процедурных моделей представления знаний** заключается в большей эффективности механизмов вывода за счет введения дополнительных знаний о применении, что однако снижает их общность. **Другое важное преимущество** заключено в выразительной силе. Процедурные системы способны смоделировать практически любую модель представления знаний. Выразительная сила процедурных систем (проявляется в расширенной системе выводов, реализуемых в них).

Необходимо отметить, что деление моделей представления знаний на декларативные и процедурные весьма условно, так как в

реальных системах представления знаний используются в равной мере элементы и сочетания всех указанных выше форм моделей представления знаний.

Экспертные системы – это компьютерные приложения, разработанные для решения сложных задач в определенной области на уровне необычного человеческого интеллекта и опыта.

#### **Характеристики экспертных систем**

- Высокая производительность
- понятный
- надежный
- Очень отзывчивый

#### **Возможности экспертных систем**

Экспертные системы способны –

- консультирование
- Инструктаж и помощь человеку в принятии решений
- Демонстрируя
- Вывод решения
- Диагностирование
- Объясняя
- Интерпретация ввода
- Прогнозирование результатов
- Обоснование заключения
- Предлагая альтернативные варианты проблемы

Они неспособны к –

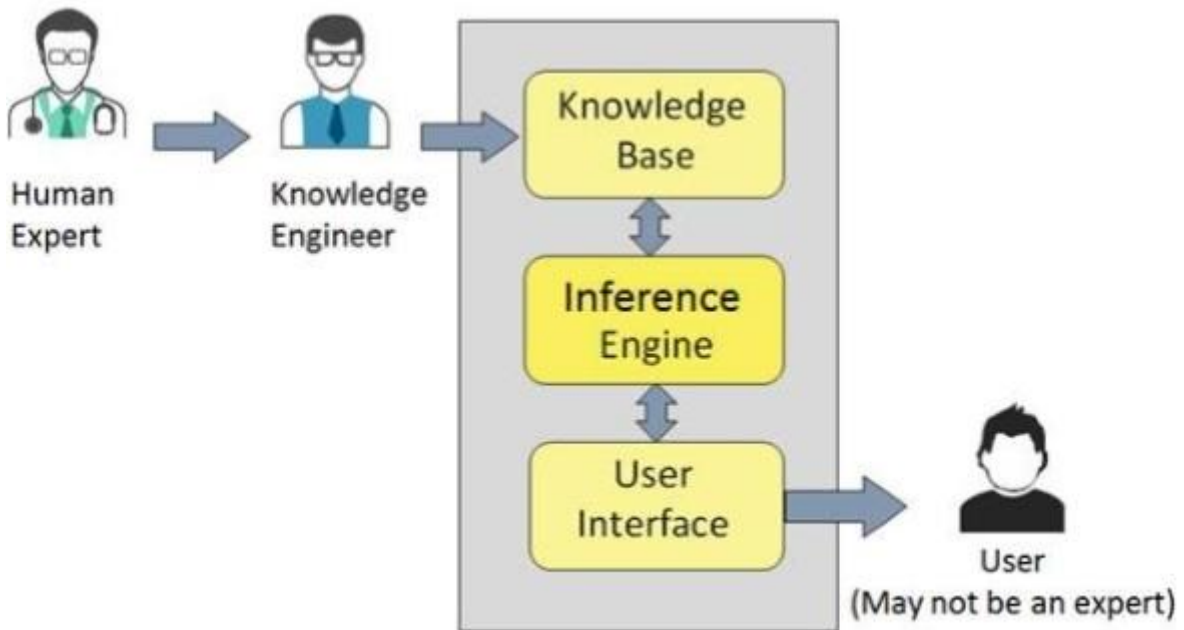
- Подмена человека, принимающего решения
- Обладая человеческими способностями
- Создание точного вывода для неадекватной базы знаний
- Уточнение собственных знаний

#### **Компоненты экспертных систем**

Компоненты ES включают в себя –

- База знаний
- Механизм логического вывода
- Пользовательский интерфейс

Давайте посмотрим их один за другим кратко.



### База знаний

Содержит специфичные для предметной области и высококачественные знания.

Знания необходимы для проявления интеллекта. Успех любого ES в основном зависит от сбора высокоточных и точных знаний.

### Что такое Знание?

Данные представляют собой сбор фактов. Информация организована в виде данных и фактов о предметной области. **Данные, информация и прошлый опыт**, объединенные вместе, называются знаниями.

### Компоненты базы знаний

База знаний ES – это хранилище фактических и эвристических знаний.

- **Фактические знания** – это информация, широко принятая инженерами знаний и учеными в области задач.
- **Эвристическое знание** – это практика, точное суждение, способность оценивать и угадывать.

**Фактические знания** – это информация, широко принятая инженерами знаний и учеными в области задач.

**Эвристическое знание** – это практика, точное суждение, способность оценивать и угадывать.

### **Представление знаний**

Это метод, используемый для организации и формализации знаний в базе знаний. Это в форме правил IF-THEN-ELSE.

### **Приобретение знаний**

Успех любой экспертной системы во многом зависит от качества, полноты и точности информации, хранящейся в базе знаний.

База знаний формируется на основе показаний различных экспертов, ученых и **инженеров знаний**. Инженер по знаниям – это человек, обладающий качествами эмпатии, быстрого обучения и навыков анализа кейсов.

Он получает информацию от эксперта-субъекта путем записи, собеседования и наблюдения за ним на работе и т. Д. Затем он систематизирует и систематизирует информацию в форме правил IF-THEN-ELSE, которые будут использоваться машиной помех. Инженер по знаниям также следит за развитием ES.

## **1.5.2. Модели представления знаний и их классификация. Продукционные и фреймовые модели представления знаний**

### **Продукционные модели**

Продукционные модели в последнее время широко используются в системах представления знаний.

Продукционные модели могут быть реализованы как процедурно, так и декларативно. Их простота и строгая форма послужили основой ряда интересных свойств, что сделало их удобным средством представления знаний.

**Продукционные модели** — это набор правил вида «условия — действие», где условиями являются утверждения о содержимом некой базы данных, а действия представляют собой процедуры, которые могут изменять содержимое БД.

Кроме того, со стороны компьютера имеется возможность определения простого и точного механизма использования знаний с высокой однородностью, описанных по одному синтаксису. Эти две отличительные черты, по-видимому, являются причинами столь широкого распространения метода представлений знаний правилами.

## Фреймы

Для представления и описания стереотипных объектов, событий или ситуаций были введены понятия «фреймы», которые являются сложными структурами данных.

Фреймы были впервые предложены в качестве аппарата для представления знаний М. Минским в 1975 г. Согласно его определению, *фреймы — это минимальные структуры информации, необходимые для представления класса объектов, явлений или процессов*. В общем виде фрейм может быть представлен в виде, показанном на рисунке 1.6 и описан строкой:

$\langle \text{ИФ}, (\text{ИС}, \text{ЗС}, \text{ПП}), \dots, (\text{ИС}, \text{ЗС}, \text{ПП}) \rangle$ ,

где *ИФ* — имя фрейма; *ИС* — имя слота; *ЗС* — значение слота; *ПП* — имя присоединенной процедуры.

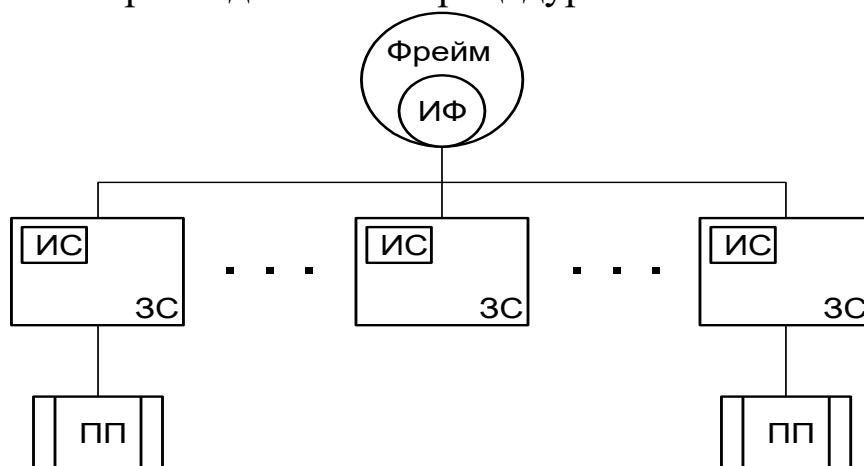


Рис. 1.6. Схема фрейма

**Слоты** — это некоторые незаполненные подструктуры фрейма, заполнение которых приводит к тому, что данный фрейм ставится в соответствие некоторой ситуации, явлению или объекту.

В качестве данных фрейм может содержать обращения к процедурам (так называемые присоединенные процедуры). Выделяют два вида процедур: процедуры-демоны и процедуры-слуги. **Процедуры-демоны** активизируются при каждой попытке добавления или удаления данных из слота (по умолчанию). **Процедуры-слуги** активизируются только при выполнении условий, определенных пользователем при создании фрейма.

Наибольшее практическое применение во фреймовых системах получили лишь отношения «абстрактное — конкретное».

Но в некоторых областях иногда требуется описывать и управлять структурированным объектом. Поэтому в таких случаях не обойтись без обработки отношений типа «целое - часть».

### **Семантические сети**

Важной схемой представления знаний являются семантические сети. Семантические сети не являются однородным классом схем представления. Имеется лишь несколько общих черт, объединяющих ряд механизмов представления, называемых семантическими сетями. Часто общей основой являются лишь сходство формального обозначения (направленный граф с помеченными вершинами и ребрами) и основной принцип, заключающийся в том, что элементы знаний должны храниться смежно, если они семантически связаны.

**Семантическая сеть** - это направленный граф с помеченными вершинами и дугами, в котором вершины соответствуют конкретным объектам, а дуги, их соединяющие, отражают имеющиеся между ними отношения.

Отношения, используемые в семантических сетях, можно разделить на следующие:

- **лингвистические** включающие в себя отношения типа «объект», «агент», «условие», «место», «инструмент», «цель», «время» и др.;
- **атрибутивные**, к которым относят форму, размер, цвет и т.д.;
- **характеризации глаголов**, т.е. род, время, наклонение, залог, число;
- **логические**, обеспечивающие выполнение операций для исчисления высказываний (дизъюнкция, конъюнкция, импликация, отрицание);
- **квантифицированные**, т.е. использующие кванторы общности и существования;
- **теоретико-множественные**, включающие понятия «элемент множества», «подмножество», «супермножество» и др.

В качестве примера, рассмотрим представление знаний, содержащихся в высказывании «Поставщик N отгрузил товар склада M автотранспортом. На рисунке 1.7 представлена интенциональная, а на рисунке 1.8 — экстенциональная

семантическая сеть. Факты обозначим овалом, а понятия и объекты прямоугольником.

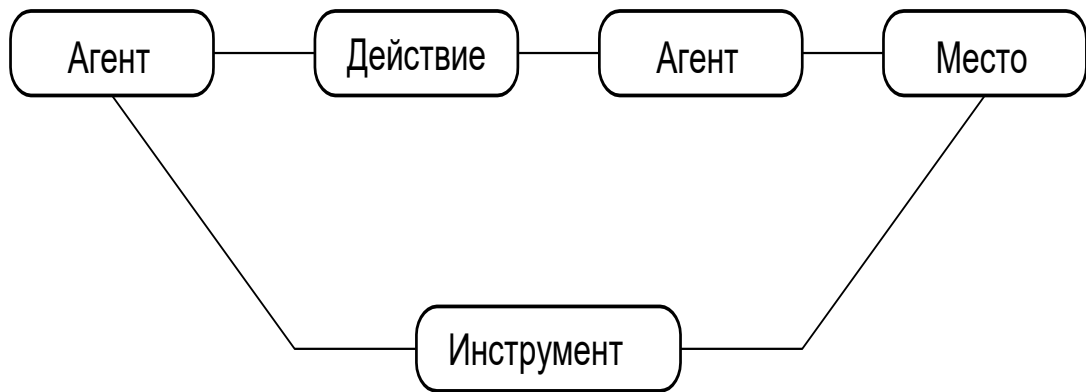


Рис. 1.7. Интенциональная семантическая модель

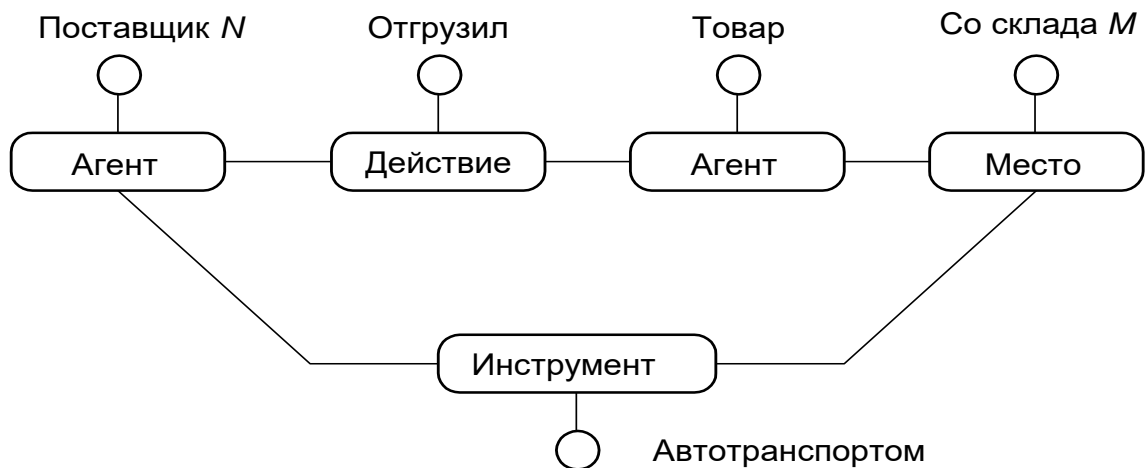


Рис.1.8. Экстенциональная семантическая сеть

## **1.6. Создание базы знаний. Структура экспертных систем. База знаний.**

Экспертные системы – это прикладные системы ИИ, в которых база знаний представляет собой формализованные эмпирические знания высококвалифицированных специалистов (экспертов) в какой-либо узкой предметной области. Экспертные системы предназначены для замены при решении задач экспертов в силу их недостаточного количества, недостаточной оперативности в решении задачи или в опасных (вредных) для них условиях.

Обычно экспертные системы рассматриваются с точки зрения их применения в двух аспектах: для решения каких задач они могут быть использованы и в какой области деятельности. Эти два аспекта накладывают свой отпечаток на архитектуру разрабатываемой экспертной системы.

Можно выделить следующие основные классы задач, решаемых экспертными системами:

- диагностика;
- прогнозирование;
- идентификация;
- управление;
- проектирование;
- мониторинг.

Наиболее широко встречающиеся области деятельности, где используются

экспертные системы:

- медицина;
- вычислительная техника;
- военное дело;
- микроэлектроника;
- радиоэлектроника;
- юриспруденция;
- экономика;
- экология;
- геология (поиск полезных ископаемых);
- математика.

Примеры широко известных и эффективно используемых (или использованных в свое время) экспертных систем:



DENDRAL– ЭС для распознавания структуры сложных органических молекул по результатам их спектрального анализа (считается первой в мире экспертной системой);

MOLGEN – ЭС для выработке гипотез о структуре ДНК на основе экспериментов с ферментами;

XCON – ЭС для конфигурирования (проектирования) вычислительных комплексов VAX 11 в корпорации DEC в соответствии с заказом покупателя;

MYCIN – ЭС диагностики кишечных заболеваний;

PUFF – ЭС диагностики легочных заболеваний;

MACSYMA – ЭС для символьных преобразований алгебраических выражений;

YES/MVS– ЭС для управления многозадачной операционной системой MVS больших ЭВМ корпорации IBM;

DART– ЭС для диагностики больших НМД корпорации IBM;

PROSPECTO – ЭС для консультаций при поиске залежей полезных ископаемых;

POMME – ЭС для выдачи рекомендаций по уходу за яблоневым садом;

набор экспертных систем для управления планированием, запуском и полетом космических аппаратов типа "челнок";

AIRPLANE – экспертная система для помощи летчику при посадке на авианосец;

ЭСПЛАН – ЭС для планирования производства на Бакинском нефтеперерабатывающем заводе;

МОДИС – ЭС диагностики различных форм гипертонии;

МИДАС – ЭС для идентификации и устранения аварийных ситуаций в энергосистемах;

NetWizard – ЭС для проектирования локальных систем.

### **1.6.1. Виды знаний**

В основе представления знаний лежат понятия семиотики – науки о знаковых системах (искусственных и естественных языках). В семиотике различают следующие разделы: синтактику, имеющую дело со структурой (синтаксисом) знаковых систем, семантику, рассматривающую смысл (интерпретацию) знаковых систем (другими словами, соответствие знаковой системы другой

знаковой системе), прагматику, имеющую дело с целенаправленностью знаковых систем. Другими словами синтактика отвечает на вопрос: как выглядит знание (как оно структурировано или как формализовано), семантика отвечает на вопрос "что означает знание", прагматика отвечает на вопрос "зачем" или "почему" необходимо (или передается) знание. Таким образом, имея дело со "знанием" необходимо уметь различать его синтаксис, семантику и прагматику. В этом разделе (методы представления знаний) рассматривается в основном синтаксис (структура знаний). Методы обработки знаний в основном характеризуют их семантику. Прагматика знаний в основном выражается в архитектуре интеллектуальной системы – особенностях реализации методов представления и обработки знаний, реализации интерфейсов системы с внешним миром.

С точки зрения глубины различают экстенциональные (конкретные, поверхностные) и интенциональные (абстрактные, глубинные) знания. Экстенциональные знания представляют собой факты об объектах реального мира. Примером их являются реляционная база данных, утверждения вида:

"Расулов имеет автомобиль";

"Расстояние от Земли до Солнца 150 млрд. км";

"Треугольник – есть геометрическая фигура".

Интенциональные знания представляют собой правила, связывающие между собой факты, или закономерности реального мира. Примерами их являются продукционная база знаний, утверждения вида:

"человек может иметь собственность, например, автомобиль";

"планета, на которой может быть жизнь, должна находиться от Солнца на расстоянии 100-300 млрд. км";

"если геометрическая фигура имеет три угла, то это треугольник".

Иногда разделение на экстенциональные и интенциональные знания условно и зависит от уровня абстрагирования. Например, при необходимости оперировать с правилами типа "если ... то" как с фактами, они выступают в роли экстенциональных знаний, а интенциональные знания, предназначенные для этого, обычно называют метазнаниями. Примером метазнания может быть утверждение

"Если мы имеем дело с Евклидовой геометрией, то справедливо утверждение "Сумма углов треугольника равна 180 градусов".

По используемым методам представления знания подразделяются на декларативные и процедурные.

Декларативные знания содержат описание объектов и отношений между ними. Их интерпретация или обработка осуществляется программами. Говорят, что при декларативном представлении семантические и синтаксические знания отделены друг от друга, что придает этой форме представления большую по сравнению с другими универсальность и общность.

Процедурные знания содержат в явном виде описание процедур, т.е. являются самоинтерпретируемыми. В этих процедурах могут быть запрограммированы действия, связанные с изменением предметной области и ее модели в базе знаний. При этом текущее состояние представляется в виде набора специализированных процедур, обрабатывающих определенный участок базы знаний. Это позволяет отказаться от хранения описаний всех возможных состояний предметной области, требуемых для работы системы ИИ, и ограничиться хранением исходного состояния и процедур, обеспечивающих преобразование модели предметной области, т.е. порождение всех других состояний из исходного. Процедурные представления реализуются, как правило, специальными языками программирования (например, PLANNER [19]). Включение семантики в базу знаний позволяет повысить эффективность поиска решений с использованием базы знаний. Но это достигается ценой специализации базы знаний, ориентации ее на особенности решаемых задач. Процедурные знания уступают декларативным в возможностях для накопления и коррекции знаний.

Разделение методов представления знаний на процедурные и декларативные является в определенной степени условным. "Чисто" декларативные или процедурные знания практически не используются. Пример чисто процедурного знания – программа, написанная на алгоритмическом языке (но не объектно-ориентированном). Пример чисто декларативного знания – реляционная база данных. В конкретных реализациях баз знаний присутствуют элементы и тех и других способов представления знаний.

Наиболее распространенными вариантами декларативного представления являются семантические сети и фреймы. Иногда говорят о реляционных базах знаний, не отличая их существенно от соответствующих баз данных. Среди процедурных методов представления знаний выделяются средства работы со списковыми структурами и исчисление высказываний первого порядка.

По степени формализации различают логические и эвристические методы представления знаний. Логические методы могут быть описаны в виде формальной теории (системы)

$$S = \langle B, F, A, R \rangle,$$

где: B – алфавит,  
F – формулы-факты,  
A – формулы-аксиомы,  
R – правила-вывода.

Примеры логических методов представления знаний: исчисление высказываний (логика предикатов) 1-го порядка, различные псевдофизические логики, продукционные модели представления знаний.

Эвристические методы представления знаний основаны на применении ряда приемов, принципов или подходов для описания знаний в удобном для понимания человеком или обработки компьютером виде. Примерами таких методов являются: семантические сети, фреймы, объектно-ориентированное представление. Эвристические методы являются более высокоуровневыми методами представления знаний и, как правило, могут быть описаны с помощью какого-либо логического метода с потерей ряда свойств, например, таких как наглядность, обозримость, универсальность.

Несколько в стороне от логических и эвристических методов представления и обработки знаний находятся нейронные сети. Их обычно не относят к методам инженерии знаний, а рассматривают в рамках нейрокибернетики (нейроинформатики). Их принципиальным отличием от методов инженерии знаний является то, что в них знания содержатся не в формализованном и локализованном виде, а в виде состояния множества нейроподобных элементов и распределены между этими элементами. Нейронная сеть обладает, как говорят, голографическими свойствами, т.е. порча какого-либо элемента

распределенного в них знания не приводит к порче всего знания, а всего лишь – к некоторому ухудшению его характеристик.

### 1.6.2. Логика предикатов первого порядка

Логика предикатов является развитием алгебры логики (или логики высказываний). В логике высказываний для обозначения фактов используются буквы (имена или идентификаторы), не имеющие структуры, и принимающие значения "1" или "0" ("да" или "нет"). В логике предикатов факты обозначаются  $n$ -арными логическими функциями – предикатами  $F(x_1, x_2, \dots, x_n)$ , где  $F$  – имя предиката (функтор) и  $x_i$  – аргументы предиката. Имена предикатов неделимы, т.е. являются так называемыми атомами. Аргументы могут быть атомами или функциями  $f(x_1, x_2, \dots, x_n)$ , где  $f$  – имя функции, а  $x_1, \dots, x_n$ , так же как и аргументы предикатов являются переменными или константами предметной области. В результате интерпретации (по-другому, конкретизации) предиката функторы и аргументы принимают значения констант из предметной области (строк, чисел, структур и т.д.). При этом следует различать интерпретацию на этапе описания предметной области (создания программ и баз знаний) и на этапе решения задач (выполнения программ с целью корректировки или пополнения баз знаний). В дальнейшем при работе с предикатами мы будем иметь дело с результатом их интерпретации в первом смысле, т.е. с их привязкой к некоторой предметной области.

Предикат с арностью  $n > 1$  может использоваться в инженерии знаний для представления  $n$ -арного отношения, связывающего между собой  $n$  сущностей (объектов) – аргументов предиката. Например, предикат отец("Иван", "Петр Иванович") может означать, что сущности "Иван" и "Петр Иванович" связаны родственным отношением, а именно, последний является отцом Ивана или наоборот. Уточнение семантики этого предиката связано с тем, как он используется, т.е. в каких операциях или более сложных отношениях он участвует и какую роль в них играют его 1-й и 2-й аргументы. Предикат компьютер(память, клавиатура, процессор, монитор) может обозначать понятие "компьютер" как отношение, связывающее между собой составные части компьютера.

Предикат с арьностью  $n=1$  может представлять свойство сущности (объекта), обозначенного аргументом или характеристику объекта, обозначенного именем предиката. Например, кирпичный(дом), оценка(5), улица("Красный проспект"), дата\_рождения("1 апреля 2017 г."), быстродействие("1 Мфлопс").

Предикат с арьностью  $n=0$  (без аргументов) может обозначать событие, признак или свойство, относящееся ко всей предметной области. Например, "конец работы".

При записи формул (выражений) помимо логических связок "конъюнкция" ( $\&$ ), "дизъюнкция" ( $\vee$ ), "отрицание" ( $\neg$ ), "следование" ("импликация") ( $\rightarrow$ ), заимствованных из логики высказываний, в логике предикатов используются кванторы всеобщности ( $\forall$ ) и существования ( $\exists$ ). Например, выражение  $\forall (x,y,z) (\text{отец}(x,y) \& \text{мать}(x,z)) \rightarrow \text{родители}(x,y,z)$  означает, что для всех значений  $x,y,z$  из предметной области справедливо утверждение "если  $y$  – отец и  $z$  – мать  $x$ , то  $y$  и  $z$  – родители  $x$ "; выражение  $(\exists x) (\text{студент}(x) \& \text{должность}(x, \text{"инженер"}))$  означает, что существует хотя бы один студент, который работает в должности инженера.

Переменные при кванторах называются связанными переменными в отличие от свободных переменных. Например, в выражении  $(\forall x) (\text{владелец}(x,y) \rightarrow \text{частная\_собственность}(y))$   $x$  – связанная переменная,  $y$  – свободная переменная.

Логика предикатов 1-го порядка отличается от логик высших порядков тем, что в ней запрещено использовать выражения (формулы) в качестве аргументов предикатов.

Решение задач в логике предикатов сводится к доказательству целевого утверждения в виде формулы или предиката (теоремы), используя известные утверждения (формулы) или аксиомы.

В конце 60-х годов Робинсоном для доказательства теорем в логике предикатов был предложен метод резолюции, основанный на доказательстве "от противного". Целевое утверждение инвертируется, добавляется к множеству аксиом и доказывается, что полученное таким образом множество утверждений является несовместным (противоречивым). Для выполнения доказательства методом резолюции необходимо провести определенные преобразования над множеством утверждений, а именно, привести их к совершенной конъюнктивной нормальной форме (СКНФ).

Приведение формул к СКНФ состоит из следующих этапов, легко реализуемых на ЭВМ.

1. Устранение импликации с помощью замены ее на отрицание и дизъюнкцию применением формулы

$$A \rightarrow B = \sim A \vee B;$$

2. Ограничение области действия символов отрицания, т.е. продвижение отрицания внутрь формулы с помощью закона де Моргана, т.е. применяя формулы

$$\neg( A \& B ) = ( \neg A ) \vee ( \neg B ) ;$$

$$\neg( A \vee B ) = ( \neg A ) \& ( \neg B ) ;$$

$$\neg( \exists x ) F ( x ) = ( \forall x ) \neg F ( x ) ;$$

$$\neg( \forall x ) F ( x ) = ( \exists x ) \neg F ( x ) .$$

3. Стандартизация или разделение переменных.

На этом этапе в каждой формуле переименовываются связанные переменные так, чтобы они стали уникальными для каждого квантора, с которым они связаны. Это делается на основании того факта, что связанную переменную в области действия квантора можно заменить на любую другую, не встречающуюся переменную, не изменив этим значение истинности формулы.

Например,  $( \forall x ) ( P(x) \vee Q(y) ) \& ( \forall x ) ( F(x) )$  преобразуется в  $( \forall x ) ( P(x) \vee Q(y) ) \& ( \forall z ) ( F(z) )$ .

4. Исключение кванторов существования. На этом этапе квантор существования заменяется так называемой функцией Сколема  $g(x)$  или перечислением предикатов с аргументами – константами из области определения переменной-аргумента.

Примеры преобразования:

исходная формула	результатирующая формула
$( \forall x ) y$	$g(x)$
$( \forall x ) ( \forall y ) z$	$g(x,y)$
$( \exists x ) ( F(x) )$	$F(a),F(b)...$

где  $a$  и  $b$  – константы. Функция Сколема задает отображение областей определения других переменных на область определения переменной, связанной с квантором существования.

5. Вынесение кванторов всеобщности в начало формулы.

6. Исключение кванторов всеобщности.

Исключение достигается просто удалением кванторов в предположении, что если в формуле есть некоторая переменная  $x$ ,

то формула справедлива для всех ее значений из области определения.

7. Собственно приведение формулы к СКНФ применением закона дистрибутивности

$$A \vee (B \& C) = (A \vee B) \& (A \vee C).$$

8. Исключение символов &. Это достигается заменой формулы вида  $(A \& B)$  на множество формул вида  $\{A, B\}$ .

Предположим надо привести к СКНФ формулу

$$(\forall x) \{ P(x) \rightarrow \{ (\forall y) [ P(y) \rightarrow P(f(x, y))] \& \neg(\forall y) [ Q(x, y) \rightarrow P(y)] \} \}.$$

Исключив импликацию, получим

$$(\forall x) \{ \neg P(x) \vee \{ (\forall y) [ \neg P(y) \vee P(f(x, y))] \& \neg(\forall y) [ \neg Q(x, y) \vee P(y)] \} \}.$$

Продвинув внутрь отрицание, получим

$$(\forall x) \{ (\neg P(x) \& \{ (\forall y) [ \neg P(y) \vee P(f(x, y))] \& [ (\exists y) [ Q(x, y) \& \neg P(y)] \} \} \}.$$

Разделив переменные, получим

$$(\forall x) \{ (\neg P(x) \& \{ (\forall y) [ \neg P(y) \vee P(f(x, y))] \& [ (\exists w) [ Q(x, w) \& \neg P(w)] \} \} \}.$$

Заменяв переменную  $w$  на сколемовскую функцию  $g(x)$ , исключим квантор существования и получим

$$(\forall x) \{ (\neg P(x) \& \{ (\forall y) [ \neg P(y) \vee P(f(x, y))] \& [ Q(x, g(x)) \& \neg P(g(x))] \} \} \}.$$

Вынесем кванторы всеобщности в начало формулы и получим

$$(\forall x)(\forall y) \{ (\neg P(x) \& [ \neg P(y) \vee P(f(x, y))] \& [ Q(x, g(x)) \& \neg P(g(x))] \} \}.$$

Применив закон дистрибутивности, получим

$$(\forall x)(\forall y) \{ [ P(x) \vee \neg P(y) \vee P(f(x, y))] \& [ \neg P(x) \vee Q(x, g(x)) \& [ \neg P(x) \vee \neg P(g(x))] \} \}.$$

Исключив кванторы всеобщности и заменив конъюнкцию формул их множеством, получим множество следующих формул (предложений):

$$\begin{aligned} &\neg P(x) \vee \neg P(y) \vee P[f(x, y)], \\ &\neg P(x) \vee Q[x, g(x, g(x))], \\ &\neg P(x) \vee \neg P[g(x)]. \end{aligned}$$



## 1.7. Модели представления знаний и их классификация

### 1.7.1. Использование специальных правил в представлении знаний. Правильная цепочка идей

Логика предикатов 1-го порядка легла в основу языков логического программирования, самым распространенным из которых является Prolog (различные его диалекты). Точнее, язык Prolog основан на модифицированной логике предикатов 1-го порядка (логике Хорна или логике дизъюнктов). Логика Хорна отличается от классической логики предикатов 1-го порядка тем, что она оперирует уже почти преобразованными к применению метода резолюции формулами без кванторов всеобщности и существования, представляющими собой множество дизъюнктов (предложений или клауз Хорна). "Почти" объясняется тем, что клаузы Хорна содержат импликацию и выглядят как

$$A \rightarrow B,$$

где  $A$  – предикат,

$B$  – предикат или конъюнкция или дизъюнкция предикатов (такое представление части  $B$  предложения возможно, т.к. конъюнкция и дизъюнкция рассматриваются как частные случаи предикатов).

Доказательство некоторого утверждения (целевого предиката) в логическом программировании сводится к процессу унификации, с помощью которого происходит рекурсивный перебор всех возможных подстановок значений переменных в целевом предикате, управляемый ограничениями, заданными множеством предложений. Множество предложений обычно в Прологе называется базой данных Пролога. База данных состоит из предложений-правил вывода вида  $A \rightarrow B$  и предложений-фактов, представляющих собой отдельные предикаты. При этом в предикатах-фактах параметрами могут только константы, а в предикатах-правилах – константы и неконкретизированные переменные. Последнее относится и к целевому предикату. В этом случае, если параметром является переменная, то это означает, что ее значение необходимо найти при доказательстве целевого предиката.

Унификация основана на сравнении (сопоставлении с образцом) целевого предиката, который надо доказать, с предикатами-фактами и предикатами-правилами из базы данных Prolog-программы. При этом успешность сопоставления двух предикатов определяется следующими условиями, упорядоченными в порядке их проверки:

- имена предикатов совпадают;
- количество параметров у предикатов совпадает;
- каждая пара сравниваемых параметров сопоставима.

Последнее условие для некоторой пары параметров истинно при трех вариантах:

- параметры являются константами (любого типа) и они равны;
- один параметр из пары является константой, а другой – переменной, в этом случае переменной присваивается значение константы;
- оба параметра являются неконкретизированными переменными, в этом случае эти переменные становятся "связанными", т.е. в дальнейшем при интерпретации программы рассматриваются как одна и та же переменная.

При унификации целевого предиката с правилом, сопоставлению подвергается сначала левая часть правила, а затем, в случае успешной унификации, последовательно проверяются предикаты, находящиеся в правой части. Т.е. правило в Прологе с точки зрения унификации рассматривается как предикат с именем "::<=" (импликация) и с параметрами А и В, а В рассматривается в свою очередь как предикат "," (конъюнкция) или ";" (дизъюнкция) с параметрами-предикатами правой части правила.

Следующий фрагмент программы (на Эдинбургской версии Пролога) описывает поведение гипотетического роботоманипулятора и может являться частью программного обеспечения его системы управления.

```
/* описание объектов, с которыми работает робот */
    куб("кубик", 10).
    цилиндр("прут", 100, 3).
/* описание их местоположения */
    объект("кубик", "стол").
    объект("прут", "коробка").
/* описание некоторых действий робота (команд) */
```

```

взять(_):- в_схвате(_), /* проверка занятости схвата */
    write("Схват занят"),
    nl,
    !.
взять(X):- объект(X,Y), /*определение положения объекта X*/
    повернуть_к(Y),
    выбрать_объект(X,Coord_X,Coord_Y),
    позиционировать_схват(Coord_X,Coord_Y),
    взять, /* включение схвата */
    assert(в_схвате(X)). /*запоминание что в схвате*/
положить(X,Y):- в_схвате(Z), /* определение, что в схвате*/
    Z<>X,
    write("Схват занят"),
    nl,
    !.
положить(X,Y):- в_схвате(X),
    повернуть_к(Y),
    центр(Y, Coord_X,Coord_Y),
    позиционировать_схват(Coord_X,Coord_Y),
    отпустить,
    retract(в_схвате(X)),
    !.
положить(X,Y):- not(в_схвате(_)),
    объект(X,Y),
    write("Объект "),
    write(X),
    write(" уже находится в "),
    write(Y),
    nl,
    !.
положить(X,Y):- not(в_схвате(_)),
    взять(X),
    положить(X,Y).

```

Здесь операторы (встроенные предикаты) `assert(X)` и `retract(X)` используются для добавления и удаления предиката-факта `X`, соответственно, оператор `!` используется для аннулирования попыток альтернативной унификации при неуспешной текущей унификации.

К недостаткам логики предикатов 1-го порядка как метода представления знаний можно отнести следующее:

- монотонность логического вывода, т.е. невозможность пересмотра полученных промежуточных результатов (они считаются фактами, а не гипотезами);
- невозможность применения в качестве параметров предикатов других предикатов, т.е. невозможность формулирования знаний о знаниях;
- детерминированность логического вывода, т.е. отсутствие возможности оперирования с нечеткими знаниями.

Но логику предикатов 1-го порядка можно использовать как основу для конструирования более сложных и удобных логических методов представления знаний. В этом качестве она используется в модальных и псевдофизических логиках.

### 1.7.2. Модальные логики

Первой попыткой расширить возможности логики 1-го порядка явилось появление множества модальных логик, в которых вводились различные кванторы (модальности) и аксиомы, отражающие тот или иной аспект реального мира. Наиболее известны модальные логики "возможного-необходимого" (алетическая логика), деонтическая логика (модальности "разрешено-обязательно"), эпистемическая логика (логика знания-веры), временная модальная логика (модальности "всегда-никогда", "часто-иногда").

Для интерпретации модальных логик возможностей предикатов, имеющих всего два значения (двузначной семантики), было недостаточно. Поэтому, сначала появилась трехзначная логика Лукасевича (логика Лукасевича), где логические переменные могут принимать значения 0, 1, 2, а затем семантика возможных миров (4-значная логика).

Логика направлений оперирует с понятиями "справа", "слева", "впереди", "сзади" или "на восток", "на запад" и т.п.

В логике взаимного расположения объектов описываются следующие базовые отношения: унарные – "иметь горизонтальное положение", "иметь вертикальное положение", бинарные –

"находиться внутри", "находиться вне", "находиться на поверхности", "находиться в центре", "находиться в середине", "быть там же, где..", "быть ненулевой проекцией", "находиться в  $\varepsilon$ -окрестности", "быть частью", "находиться на одной прямой", "находиться вокруг", "быть на краю", "быть параллельно", "быть перпендикулярно", "быть симметрично", "находиться в  $n$  единицах от..", "иметь точку опоры на..", "иметь точку подвеса на..", "соприкасаться", "быть выше", "быть ниже", "находиться на одинаковом уровне", "быть дальше", "быть ближе", "быть равноудаленными",  $n$ -арное отношение – "быть между".

Из базовых отношений с помощью логических связок строятся производные отношения, такие как "не соприкасаться" (отрицание "соприкасаться"), "быть вместе.." (следствие от "находиться там же.."), "висеть" (конъюнкция "иметь вертикальное положение" и "висеть на..."), "стоять" (конъюнкция "иметь вертикальное положение" и "иметь точку опоры на..") и т.п.

Правило-продукция (или просто правило) в общем случае можно представить в виде

$$\langle I, S, P, A \rightarrow B, F \rangle,$$

где  $I$  – идентификатор правила (обычно порядковый номер);

:  $S$  – область применимости;

$P$  – условие применимости;

$A$  – посылка правила;

$B$  – заключение;

$F$  – постусловие правила.

$A \rightarrow B$  является ядром правила-продукции и может по-разному интерпретироваться. Наиболее часто используемая форма интерпретации – логическая, при которой  $A$  является множеством элементарных условий, связанных логическими связками "И", "ИЛИ" и "НЕ",  $B$  – множеством элементарных заключений. При этом правило считается сработавшим (выполняется заключение  $B$ ), если посылка  $A$  истинна. Другой формой интерпретации ядра является вероятностная интерпретация, при которой правило срабатывает с некоторой вероятностью, зависящей от истинности посылки.

В качестве заключения обычно применяется операция добавления факта в базу данных интеллектуальной системы с указанием меры достоверности получаемого факта. В качестве

постусловия могут использоваться какие-либо дополнительные действия или комментарии, сопровождающие правило.

Обычно при описании баз знаний или экспертных систем правила представляются в более наглядном виде, например:

ПРАВИЛО 1:

ЕСЛИ

Образование=Высшее И

Возраст=Молодой И

Коммуникабельность=Высокая

ТО

Шансы найти работу=Высокие КД=0.9.

При срабатывании этого правила в базу данных интеллектуальной системы (например, экспертной системы) добавляется факт, означающий, что шансы найти работу высоки с достоверностью 0.9 или 90% (значение коэффициента достоверности КД). Понятия "Образование", "Возраст", "Коммуникабельность" служат для задания условия (в данном случае, конъюнкции), при котором срабатывает правило.

Факты хранятся в базе данных продукционной системы в форме  
(Объект, значение, КД)

или

(Объект, атрибут, значение, КД).

Но могут использоваться и другие структуры для хранения фактов, такие как семантические сети или фреймы. В этом случае говорят о комбинации разных методов представления знаний или о гибридных интеллектуальных (экспертных) системах. При интерпретации (выполнении) правила в ходе проверки условия система проверяет факты, находящиеся уже в базе данных, и, если соответствующего факта нет, обращается за ним к источнику данных (пользователю, базе данных и т.д.) с вопросом (или запросом).

Кроме правил в продукционных базах знаний могут использоваться метаправила для управления логическим выводом. Пример метаправила для гипотетической базы знаний, пример из которой был приведен ранее:

ЕСЛИ

Экономика = развивается

ТО

Увеличить приоритет правила 1

Достоинствами продукционного метода представления знаний являются следующие.

1. Наглядность и понятность знаний (по крайней мере, на уровне одного правила).

2. Возможность реализации немонотонного логического вывода и обработки противоречивых фактов.

3. Возможность введения различных модификаций в интерпретацию правил в соответствии с особенностями решаемых системой задач.

4. Возможность легкого наращивания базы знаний путем добавления новых правил.

Недостатками этого метода представления являются следующие.

1. Необозримость большой базы знаний и ее структуры.

2. Возможность легкого внесения серьезных искажений в базу знаний,

приводящих к неправильному функционированию системы (если в системе нет развитых средств проверки целостности базы знаний).

3. Ориентация на последовательную обработку правил.

### **1.7.3. Семантические сети**

Семантической сетью называется ориентированный граф с помеченными вершинами и дугами, где вершинам соответствуют конкретные объекты, дугам – отношения между ними.

В семантических сетях используются три основных типа объектов: понятия, события и свойства. Понятия представляют собой сведения об абстрактных или конкретных (физических) объектах предметной области.

События – это действия, которые могут внести изменения в предметную область, т.е. изменить состояние предметной области.

Свойства используются для уточнения понятий и событий. Применительно к понятиям свойства описывают их особенности или характеристики, например – цвет, размер, качество. Применительно к событиям свойства – продолжительность, место, время и т.д.

Семантические отношения условно делятся на четыре класса: лингвистические, логические, теоретико-множественные и квантифицированные. К наиболее распространенным лингвистическим отношениям относятся падежные и атрибутивные отношения. Падежными (или ролевыми) отношениями могут являться следующие:

– агент, отношение между событием и тем, что (или кто) его вызывает, например, отношение между "завинчиванием" (гайки) и рукой;

– объект, отношение между событием и тем, над чем производится действие, например, между "завинчиванием" и "гайкой";

– условие, отношение, указывающее логическую зависимость между событиями, например, отношение между "завинчиванием" (гайки) и "сборкой" (узла);

– инструмент, отношение между событием и объектом, с помощью которого оно совершается, например, между "завинчиванием" и "верстаком".

Атрибутивные отношения – это отношения между объектом и свойством, например, цвет, размер, форма, модификация и т.д.

Логические отношения – это операции, используемые в исчислении высказываний: дизъюнкция, конъюнкция, импликация, отрицание.

Теоретико-множественные отношения – это отношения между элементом множества (подмножества) и множеством, отношение части и целого и т.п. Этот тип отношений используется для хранения в базе знаний сложных (составных или иерархических) понятий.

Квантифицированные отношения – это логические кванторы общности и существования. Они используются для представления знаний типа: "любой студент должен посещать лабораторные занятия", "существует хотя бы один язык программирования, который должен знать любой выпускник НГТУ".

К базе знаний представленной семантической сетью, возможны следующие основные типы запросов:

- 1) запрос на существование;
- 2) запрос на перечисление.



При построении интеллектуальных банков знаний обычно используют разделение интенциональных и экстенциональных знаний. Экстенциональная семантическая сеть (или К-сеть) содержит информацию о фактах, о конкретных объектах, событиях, действиях. Интенциональная семантическая сеть (или А-сеть) содержит информацию о закономерностях, потенциальных взаимосвязях между объектами, неизменяемую информацию об объектах, т.е. модель мира. Экстенциональные (конкретные) знания создаются и обновляются в процессе работы с банком данных, а интенциональные (абстрактные) изменяются редко. Первые можно назвать экземпляром, а последние – моделью (схемой) базы данных.

Запрос к банку знаний, обрабатываемый системой управления базой знаний, представляет собой набор фактов (ситуацию), при описании которого допускается использование переменных вместо значений атрибутов, имен понятий, событий и отношений. Запрос можно представить в виде графа, в котором вершины, соответствующие переменным, не определены.

Поиск ответа сводится к задаче изоморфного вложения графа запроса (или его подграфа) в семантическую сеть.

Запрос на существование не содержит переменных и требует ответа типа ДА, если изоморфное вложение графа запроса в семантическую сеть удалось, и НЕТ – в противоположном случае. При обработке запроса на перечисление происходит поиск всех возможных изоморфных графу запроса подграфов в семантической сети, а также присваивание переменным в запросе значений из найденных подграфов.

Достоинством семантических сетей является их универсальность, достигаемая за счет выбора соответствующего применению набора отношений. В принципе с помощью семантической сети можно описать сколь угодно сложную ситуацию, факт или предметную область.

Недостатком семантических сетей является их практическая необозримость при описании модели мира реального уровня сложности. При этом появляется проблема размещения семантической сети в памяти ЭВМ. Если ее размещать всю в оперативной (виртуальной) памяти, на ее сложность накладываются жесткие ограничения. Если размещать во внешней

памяти, появляется проблема, как подгружать необходимые для работы участки.

Частично эта проблема структуризации семантических сетей решается выделением фрагментов семантической сети, называемых обычно высказываниями. Ниже приводится фрагмент семантической сети, состоящей из двух высказываний, связанных отношением импликации (следования).

Кроме того, в семантических сетях нельзя явно задавать наследование свойств, т.к. отношение "элемент класса – класс" является одним из многих типов отношений и его обработка (участие в процедуре поиска подходящих фрагментов) ничем не отличается от обработки других отношений.

Эта проблема структуризации семантических сетей и необходимости задания в них наследования свойств привела к идее структуризации семантических сетей, приведшей к появлению концепции фреймов.

#### **1.7.4. Фреймы**

В основе теории фреймов лежит восприятие фактов посредством сопоставления полученной извне информации с конкретными элементами и значениями, а также, с рамками, определенными для каждого концептуального объекта в памяти. Структура, представляющая эти рамки, называется фреймом.

Другими словами, фрейм – это структура, описывающая фрагмент базы знаний, который в какой-то степени рассматривается и обрабатывается обособленно от других фрагментов. Другие фрагменты, с которыми он связан, во фрейме представлены только их именами (идентификаторами) так же как и он в них.

В виде фрейма может описываться некоторый объект, ситуация, абстрактное понятие, формула, закон, правило, визуальная сцена и т.п.

Понятие фрейма неразрывно связано с абстрагированием и построением иерархии понятий.

Из понятия "фрейм", появившегося в конце 60-х годов в работах М. Минского, выросло в дальнейшем понятие объекта и объектно-ориентированного программирования с его идеями инкапсуляции

данных в объекте, наследования свойств и методами, привязанными к описанию объектов.

Так как фрейм является более общей и гибкой концепцией, чем "объект", в дальнейшем будем использовать терминологию, сложившуюся при описании фреймов, подразумевая, что с некоторыми ограничениями они могут быть перенесены в среду объектно-ориентированного программирования.

Фреймы подразделяются на два типа: фреймы-прототипы (или классы) и фреймы-примеры (или экземпляры). Фреймы-прототипы используются для порождения фреймов-примеров.

В общем виде фрейм можно описать как структуру, состоящую из имени фрейма, множества слотов, характеризующихся именами и значениями, и множества присоединенных процедур, связанных с фреймом или со слотами:

$$F = ( NF , P, ( NS_1 , VS_1 , P ), \dots, ( NS_i , VS_i , P_i ), \dots, ( NS_n , VS_n , P_n ) ),$$

где  $NF$  – имя фрейма,  $NS$  – имя слота,  $VS$  – значение слота,  $P$  – присоединенная процедура.

Во фреймах различают два типа присоединенных процедур: процедуры-демоны и процедуры-слуги. Первые из них запускаются автоматически при наличии некоторых условий или событий. Процедуры-слуги запускаются явно.

В реальных системах, базирующихся на фреймах, структура, описанная выше, может быть более сложной за счет иерархической структуры значений слотов (наличия более глубоких уровней в описании фрейма).

Метод прямого логического вывода можно применять тогда, когда факты появляются в базе данных не зависимо от того, какую задачу сейчас требуется решить (какой целевой факт доказать) и в разные моменты времени. В этом случае можно говорить о том, что факты управляют логическим выводом (решением задачи). Кроме того, этот метод целесообразно применять для формирования вторичных признаков (фактов) из первичных для подготовки решения задачи в дальнейшем с применением обратного логического вывода.

Метод обратного логического вывода можно применять тогда, когда необходимо минимизировать количество обращений к

источнику данных (например, пользователю), исключив из рассмотрения заведомо ненужные для решения задачи факты. Основной проблемой при разработке современных экспертных систем является проблема приобретения знаний, т.е. преобразование разного вида информации (данных) из внешнего представления в представление в виде знаний, пригодное для решения задач, для которых создается экспертная система. Эту проблему часто называют проблемой извлечения знаний из данных (в более общем виде, из внешнего мира), которая сводится к задаче обучения интеллектуальной системы.

Примерами задач извлечения знаний являются:

1) выявление причинно-следственных связей между атрибутами реляционной базы данных и формирование их в виде правил в продукционной экспертной системе;

2) формирование программы (или правил) решения задачи (например, планирования производственного процесса или поведение робота) на основе примеров удачного планирования, вводимых в компьютер;

3) выявление информативных признаков для классификации объектов, существенных с точки зрения решаемой задачи.

Обучающиеся системы можно классифицировать по двум признакам: уровень, на котором происходит обучение и применяемый метод обучения. По первому признаку различают обучение на символьном уровне (SLL – symbol level learning), при котором происходит улучшение представления знаний на основе опыта, полученного при решении задач, и обучение на уровне знаний (KLL – knowledge level learning), при котором происходит формирование новых знаний из существующих знаний и данных.

На символьном уровне обучение сводится к манипулированию уже существующими структурами, представляющими знание, например, корректировка коэффициентов достоверности правил-продукций, изменение порядка расположения (просмотра) правил-продукций в базе знаний вводимого пользователем описания решения задачи на достаточно формализованном языке, не сильно отличающемся от языка, на котором представляются знания в системе.

На уровне знаний обучение сводится к выявлению и формализации новых знаний. Например, из фактов

журавль умеет летать,  
воробей умеет летать,  
синица умеет летать,  
журавль есть птица,  
воробей есть птица,  
синица есть птица

система может сформулировать правило-продукцию

Если X есть птица  
то X умеет летать.

По признаку применяемого метода обучения различают системы, в которых используются аналитические или эмпирические методы обучения. Аналитические, в свою очередь, делятся на использующие глубинные (knowledge-rich) или поверхностные (knowledge-driven) знания.

Эмпирические делятся на использующие знания (knowledge-learning) или данные (data-driven).

С другой стороны в инженерии знаний известны три основных подхода к приобретению знаний: индуктивный вывод, вывод по аналогии и обучение на примерах. В основе индуктивного вывода лежит процесс получения знаний из данных и/или других знаний (в продукционных системах – правил из фактов и/или других правил). Вывод по аналогии основан на задании и обнаружении аналогий между объектами (ситуациями, образами, постановками задачи, фрагментами знаний) и применением известных методов (процедур) к аналогичным объектам. В основе обучения на примерах лежит демонстрация системе и запоминание ей примеров решения задач. Резкой границы между этими методами не существует, т.к. все они базируются на обобщении, реализованной в той или иной форме, т.е. реализуют переход от более конкретного знания (фактов) к более абстрактному знанию.

#### **1.7.4.1. Фреймовая модель представления знаний**

База знаний содержит набор фреймов и правил-продукций.

Пример базы знаний (фрагмент экспертной системы для проектирования экспертных систем):

TITLE = для выбора метода представления знаний

FRAME = Цель

Метод представления знаний: ()

```

ENDF
FRAME = Тип
    Решаемые задачи: (диагностика; проектирование)
ENDF
FRAME = Область
    Применение [Какова область применения?]: (медицина;
вычислительная техника)
ENDF
FRAME = Количество
    Число правил в базе знаний (численный): ()
    Число объектов в базе знаний (численный): ()
ENDF
FRAME = Действие
    Сообщение: ()
ENDF
RULE 1
    > (Количество.Число правил в базе знаний; 50)
    < (Количество.Число правил в базе знаний; 100)
    < (Количество.Число объектов в базе знаний; 30)
DO
    = (Тип.Решаемые задачи; диагностика) 100
ENDR
RULE 2
    > (Количество.Число правил в базе знаний; 100)
    > (Количество.Число объектов в базе знаний; 30)
DO
    = (Тип.Решаемые задачи; проектирование) 100
ENDR
RULE 3
    = (Область.Применение; медицина)
    = (Тип.Решаемые задачи; диагностика)
DO
    = (Метод представления знаний; Правила-продукции с
представлением нечетких знаний) 90
ENDR
RULE 4

```

= (Область.Применение; вычислительная техника)

= (Тип.Решаемые задачи; проектирование)

DO

= (Метод представления знаний; Фреймы) 100

= (Метод представления знаний; Правила-продукции с представлением нечетких знаний) 70

= (Метод представления знаний; Семантические сети) 70

MS(Действие.Сообщение; Доказано правило 4)

ENDR

К порядку следования фреймов и правил-продукций, нумерации правил-продукций жестких требований не предъявляется. Единственное ограничение – неизменность номера правила-продукции на протяжении всего сеанса работы с базой знаний. Начало нумерации и порядок нумерации правил-продукций может быть произвольным, но из соображений целесообразности лучше начинать нумерацию с единицы и нумеровать правила по порядку.

База знаний состоит из двух частей: постоянной и переменной. Переменная часть базы знаний называется базой данных и состоит из фактов, полученных в результате логического вывода. Факты в базе данных не являются постоянными. Их количество и значение зависит от процесса и результатов логического вывода.

До начала работы с экспертной оболочкой база знаний находится в текстовом файле. В файле с расширением \*.klb (KnowLedge Base) хранятся фреймы и правила-продукции (база знаний). При начале работы с программной оболочкой наличие данного файла обязательно. Этот файл создается пользователем с помощью специального редактора или вручную. В файле с расширением \*.dtb (DaTa Base) хранятся факты, полученные в процессе логического вывода (база данных). При начале работы с программной оболочкой наличие данного файла необязательно. Файл с базой данных создается программной оболочкой в процессе логического вывода. Первые части имен этих двух файлов совпадают. В файле с расширением \*.lvd (Linguistic Variable Descriptor) описаны лингвистические переменные, используемые в базе знаний.

При работе с программной оболочкой (после загрузки в оперативную память баз знаний и данных) фреймы и правила-продукции, находившиеся в файле с расширением \*.klb, остаются

неизменными. Факты, находившиеся в файле с расширением \*.dtb, могут изменяться в процессе логического вывода (появляться, удаляться или менять свое значение в результате срабатывания правил-продукций или диалога с пользователем).

База знаний может содержать специальную конструкцию SOURCE фреймоподобного типа:

SOURCE = <имя конструкции>

PARENT: <имя фрейма с описанием внешней базы данных>

<имя слота 1> [<арифметическое выражение>]: (<имя поля 1 в БД>

<имя слота 2> [<арифметическое выражение >]: (<имя поля 2 в БД>

...

<имя слота n> [<арифметическое выражение >]: (<имя поля n в БД>

ENDS

Конструкция SOURCE используется для связи базы знаний с какой-либо стандартной базой данных. На ее основе автоматически формируется SQL-запрос. В нем задается отображение структуры одноименного фрейма на поля базы знаний. Имя внешней базы данных определяется во фрейме, имеющем имя, совпадающее с именем базы данных. Слот PARENT конструкции SOURCE ссылается на фрейм, имеющий имя, совпадающее с именем базы данных. Другие слоты данного фрейма определяют имя таблицы в базе данных, имена полей таблицы, SQL-запрос. Если слот, предназначенный для задания SQL-запроса, не имеет значения, в поле вопроса такого слота можно использовать вычисляемое выражение, которое и будет использоваться в качестве значения слота.

Использование конструкции SOURCE и фрейма с описанием внешней базы данных позволяет в процессе логического вывода получать факты из внешней базы данных с помощью SQL-запроса.

Конструкция SOURCE должна удовлетворять следующим требованиям:

1) значение слота Parent должно соответствовать ALIAS в BDE для доступа к базе данных,

2) количество слотов должно соответствовать количеству слотов в одноименном с SOURCE фрейме и их имена тоже,



3) значения слотов соответствуют внутренним именам полей в базе данных (таблице), в качестве таблицы может выступать запрос в ACCESS,

4) если надо использовать вычисляемое поле, значение слота должно быть пусто, а в качестве вопроса к слоту (в квадратных скобках) пишется выражение для значения поля.

Кроме структуры SOURCE в БЗ должен быть фрейм с именем, соответствующим ALIAS, в котором должен быть обязательно слот с именем Table и значением-именем таблицы в базе данных (или запроса в ACCESS). Могут быть и другие слоты, описывающие базу данных и носящие информативный характер или участвующие в диалоге (на усмотрение автора БЗ).

Во время логического вывода при обращении к слоту, содержащемуся во фрейме, связанном с внешней базой данных (одноименном с конструкцией SOURCE), происходит открытие формы для автоматизированного формирования SQL-запроса и чтения всех слотов фрейма из базы данных.

Фреймы используются в базе знаний для описания объектов, событий, ситуаций, прочих понятия и взаимосвязей между ними. Фрейм – это структура данных, состоящая из слотов (полей). Формат внешнего представления фреймов:

```
FRAME (⟨тип фрейма⟩) = ⟨имя фрейма⟩
PARENT: ⟨имя фрейма-родителя⟩
  ⟨имя слота 1⟩ (⟨тип слота⟩) [⟨вопрос слота?⟩] {⟨комментарий
слота⟩}:
  (⟨значение 1⟩; ⟨значение 2⟩; ...; ⟨значение k⟩)
  ⟨имя слота 2⟩ (⟨тип слота⟩) [⟨вопрос слота?⟩] {⟨комментарий
слота⟩}:
  (⟨значение 1⟩; ⟨значение 2⟩; ...; ⟨значение l⟩)
  ...
  ⟨имя слота n⟩ (⟨тип слота⟩) [⟨вопрос слота?⟩] {⟨комментарий
слота⟩}:
  (⟨значение 1⟩; ⟨значение 2⟩; ...; ⟨значение m⟩)
ENDF
```

Фрейм может принадлежать к одному из трех типов фреймов: фрейм-класс (тип описывается зарезервированным словом «класс»), фрейм-шаблон (тип описывается зарезервированным словом «шаблон»), фрейм-экземпляр (тип описывается зарезервированным словом «экземпляр»). В базе знаний содержатся фреймы-классы и фреймы-шаблоны. При создании базы знаний тип фрейма-класса можно не описывать, этот тип фрейма понимается по умолчанию. Явно следует описывать только тип фрейма-шаблона.

Среди фреймов-классов выделяется специальный фрейм-класс «Цель», задающий перечень целей логического вывода (то есть обозначений задач, решаемых экспертной системой).

База данных содержит только фреймы-экземпляры. Тип фрейма-экземпляра в базе данных понимается по умолчанию.

Имя фрейма, фрейма-родителя, слота – последовательность символов (кириллические и/или латинские буквы, цифры, пробелы, знаки подчеркивания).

Тип слота – символьный, численный или лингвистическая переменная. Обязательным является описание численного типа слота (описывается зарезервированным словом «численный») и лингвистической переменной (описывается зарезервированным словом «лп»). Слот без описания типа по умолчанию понимается как символьный. Описание типа слота заключается в круглые скобки ().

Вопрос слота – любая последовательность символов, заключенная в квадратные скобки []. Вопрос слота не является обязательным. При отсутствии вопроса будет использована формулировка: «Выберите значение» или «Введите значение».

Комментарий слота – имя текстового (\*.txt) или графического файла (\*.bmp), заключенного в фигурные скобки {}. Комментарий слота не является обязательным.

Значение слота – любая последовательность символов. Значения слота разделяются точками с запятыми. Список значений слота необязателен. Слот фрейма-экземпляра имеет единственное значение, слот фрейма-класса и фрейма-шаблона имеет неограниченное число значений.

### 1.7.4.2. Правила-продукции

Правила-продукции описывают отношения между объектами, событиями, ситуациями и прочими понятиями. На основе отношений, задаваемых в правилах, выполняется логический вывод (решение выбранной задачи). В условиях и заключениях правил присутствуют ссылки на фреймы и их слоты. Формат внешнего представления правил:

```
RULE <номер правила>
  <условие 1>
  <условие 2>
  ...
  <условие m>
DO
  <заключение 1>
  <заключение 2>
  ...
  <заключение n>
ENDR
```

Номер правила – целое число. Начало и порядок нумерации правил произвольный, предпочтительнее правила нумеровать по порядку и начинать с единицы.

Формат записи условий и заключений одинаков и имеет следующий вид:

<отношение> (<имя слота>; <значение слота>) <коэффициент достоверности>.

Отношения в условиях и заключениях могут быть:

EQ | = равно;  
GT | > больше;  
LT | < меньше;  
DL удаление слота во фрейме-экземпляре;  
EX запуск внешней программы;  
FR вывод фрейма-экземпляра;  
GO запуск правила;  
MS выдача сообщения.

В заключениях правил используются только отношения EQ, DL, EX, FR, GO, и MS. Для строковых значений слотов могут быть

использованы только отношения EQ, DL, EX, FR, GO, MS. Для лингвистических переменных допустимы все отношения, так как с ними связаны как строковые, так и численные значения.

Имя слота – локальное или глобальное. Локальное имя слота соответствует имени слота в некотором фрейме. Глобальное имя слота содержит имя фрейма и имя слота, разделенных точкой.

Значение слота – строка или число (определяется типом слота). Если в качестве значения слота используется имя фрейма-шаблона, то в процессе логического вывода выполняется одновременное определение значений для всех слотов данного фрейма.

Коэффициент достоверности – число от 0 до 100. Коэффициент достоверности в заключении используется при формировании значения слота фрейма-экземпляра при срабатывании правила. Коэффициент достоверности в условии в этой версии не используется. По умолчанию коэффициент достоверности принимает значение 100.

### **1.7.4.3. Лингвистические переменные**

При формировании базы знаний для описания нечетких понятий используются лингвистические переменные в качестве слотов. Лингвистическая переменная позволяет при логическом выводе задавать как символьное, так и численное значение слота.

Лингвистическая переменная имеет одно или несколько символьных значений. Каждому символьному значению поставлена в соответствие функция принадлежности, которая определяет отношение между численным значением лингвистической переменной и коэффициентом достоверности для данного численного значения (соответствующего символьному значению). Для каждого символьного значения лингвистической переменной существует собственная функция принадлежности. Функция принадлежности определяется на отрезке метрической шкалы, одном и том же для всех символьных значений лингвистической переменной.

Описание лингвистических переменных хранится в текстовом файле (\*.lvd – Linguistic Variable Description). Первая часть имени файла должна соответствовать именам файлов, содержащих базу знаний и базу данных (\*.klb и \*.dtb).

Формат внешнего представления лингвистической переменной:

- ⟨число лингвистических переменных⟩
- ⟨имя лингвистической переменной 1⟩
- ⟨нижнее значение границы метрической шкалы⟩
- ⟨верхнее значение границы метрической шкалы⟩
- ⟨шаг метрической шкалы⟩
- ⟨число символьных значений лингвистической переменной 1⟩
- ⟨символьное значение 1⟩
  - ⟨значение функции принадлежности 1⟩
  - ...
  - ⟨значение функции принадлежности m⟩
  - ...
- ⟨символьное значение n⟩
  - ⟨значение функции принадлежности 1⟩
  - ...

#### 1.7.4.4. Логические модели представления знаний

##### Исчисление предикатов

Классическим механизмом представления знаний в системах является исчисление предикатов. В системах, основанных на исчислении предикатов, знания представляются с помощью перевода утверждений об объектах некоторой предметной области в формулы логики предикатов и добавления их как аксиом в систему. Рассмотрим основные положения логики предикатов.

Пусть имеется некоторое множество объектов, называемых *предметной областью M*. Знаки, обозначающие элементы этого множества, называют *предметными константами*, а знак, обозначающий произвольный элемент этого множества, — *предметной переменной*. *Терм* — это всякая предметная область или предметная константа.

Если  $f$  — функциональная  $n$ -местная функция, и  $t_1, t_2, \dots, t_n$  — термы, то  $f(t_1, t_2, \dots, t_n)$  есть терм.

Выражение  $P(x_1, x_2, \dots, x_n)$ , где  $x_i = \overline{1, n}$  — предметные переменные, а  $P$  принимает значения 0 и 1, называется *логической*

*функцией* или *предикатом*. Переменные принимают значения из произвольного конечного и бесконечного множества  $M$ .

**Предикатом** или **логической функцией** называется функция от любого числа аргументов, принимающая истинные значения 1 и 0. Если в данном выражении заменить  $x_i$  на  $y_i$  где  $y_i$  — предметные константы, то получим элементарную формулу, т.е. предикатные буквы применимы также и к предметным константам. Элементарные формулы иногда называют атомными. Из элементарных формул с помощью логических связок  $\vee$  (или),  $\wedge$  (и),  $\neg$  (отрицание),  $\rightarrow$  (импликация) строят предметные формулы (иногда их называют **правильно построенными формулами** — ППФ). ППФ — один из важных классов выражений в исчислении предикатов. Кроме логических связок в рассмотрение вводят кванторы общности  $\forall$  или существования  $\exists$ .

Если  $P$  — предикатная формула, а  $x$  — предметная переменная, то выражения  $\forall xP(x)$  и  $\exists xP(x)$  также считаются предметными формулами. В логике предикатов для компактной записи высказываний типа: «для любого  $x$  истинно  $P(x)$ » и «существует такое  $x$ , для которого истинно  $P(x)$ » вводятся две новые дополнительные операции: квантор общности  $\forall$  и квантор существования  $\exists$ . Посредством этих операций приведенные выше высказывания записываются в виде  $\forall xP(x)$  и  $\exists xP(x)$ . Выражение  $\forall xP(x)$  обозначает высказывание истинное, когда  $P(x)$  истинно при всех  $x \in M$  и ложно в противном случае.

Если  $P(x)$  в действительности не зависит от  $x$ , то выражения  $\forall xP(x)$  и  $\exists xP(x)$  обозначают то же, что и  $P(x)$ .

Конкретное вхождение переменной  $x$  в формулу  $P$  называется связанным, если оно либо непосредственно следует за каким-либо квантором, либо содержится в области действия некоторого квантора  $\forall$  или  $\exists$ . Вхождение переменной является свободным, если оно не является связанным. В выражении  $\forall xP(x, y)$   $x$  — связанная,  $y$  — свободная.

**Связанной переменной** называется переменная, если в  $P$  имеется вхождение этой переменной.

Под **интерпретацией** предикатных формул понимают конкретизацию предметной области, соответствующей данной предметной формуле, и установке соответствия между символами,

входящими в предмет, и элементами (а также функциями и отношениями), определяемыми в данной предметной области.

### **Вывод на предикатах**

*Выводом системы представления знаний на предикатах являются формулы, выводимые из аксиом с помощью правил вывода.* Для организации логического вывода могут использоваться правила.

Определим основные формы логического вывода.

**Индукция** — это форма мышления, посредством которой мысль наводится на какое-либо общее правило, общее положение, присущее всем единичным предметам какого-либо класса.

**Дедукция** — такая форма мышления, когда новая мысль выводится чисто логическим путем (т.е. по законам логики) из предшествующих мыслей. Такая последовательность мыслей называется выводом, а каждая компонента этого вывода является либо ранее доказанной мыслью, либо аксиомой, либо гипотезой. Последняя мысль данного вывода называется заключением.

### **Процесс стандартизации**

Любую ППФ исчисления предикатов можно представить в виде предложения, применяя к ней последовательность простых операций. Задача состоит в том, чтобы показать, как придать произвольной ППФ форму предложения. Этот процесс (преобразования ППФ в форму предложения) состоит из следующих этапов:

1) **исключение знаков импликации.** В форме предложения в исчислении предикатов явно используются лишь связки  $\vee$  и  $\neg$ . Знак импликации можно исключить в исходном утверждении вместо  $A \rightarrow B$  записать  $\neg A \vee B$ ;

2) **уменьшение области действия знаков отрицания.** Надо, чтобы знак отрицания  $\neg$  применялся не более чем к одной предикатной букве;

3) **стандартизация переменных,** при которой осуществляется переименование переменных с тем, чтобы каждый квантор имел свою переменную. Так, вместо  $(\forall x)\{P(x) \rightarrow (\exists x)Q(x)\}$  следует написать  $(\forall x)\{P(x) \rightarrow (\exists y)Q(y)\}$ ;

4) *исключение кванторов существования;*

5) *приведение к предваренной нормальной форме (ПНФ), .*

На этом этапе уже не осталось кванторов существования, а каждый квантор общности имеет свою переменную

б) *приведение матрицы к конъюнктивной нормальной форме (КНФ).* Любую матрицу можно представить в виде конъюнкций конечного множества дизъюнкций предикатов и (или) их отрицаний. Говорят, что такая матрица имеет КНФ. Заменить  $A \vee \{B \wedge C\}$  на  $\{A \vee B\} \wedge \{A \vee C\}$ ;

7) *исключение кванторов общности;*

8) *исключение связок.*

### **Механизм логического вывода**

Использование эффективных процедур и правил Механизмом Вывода имеет важное значение для выведения правильного, безупречного решения.

В случае ES, основанной на знаниях, Inference Engine получает и манипулирует знаниями из базы знаний, чтобы прийти к конкретному решению.

В случае ES на основе правил это –

- Применяет правила повторно к фактам, которые получены из более раннего применения правила.
- Добавляет новые знания в базу знаний, если требуется.
- Разрешает конфликт правил, когда несколько правил применимы к конкретному случаю.

Применяет правила повторно к фактам, которые получены из более раннего применения правила.

Добавляет новые знания в базу знаний, если требуется.

Разрешает конфликт правил, когда несколько правил применимы к конкретному случаю.

Чтобы порекомендовать решение, механизм вывода использует следующие стратегии:

- Вперед Цепочка
- Обратная цепочка

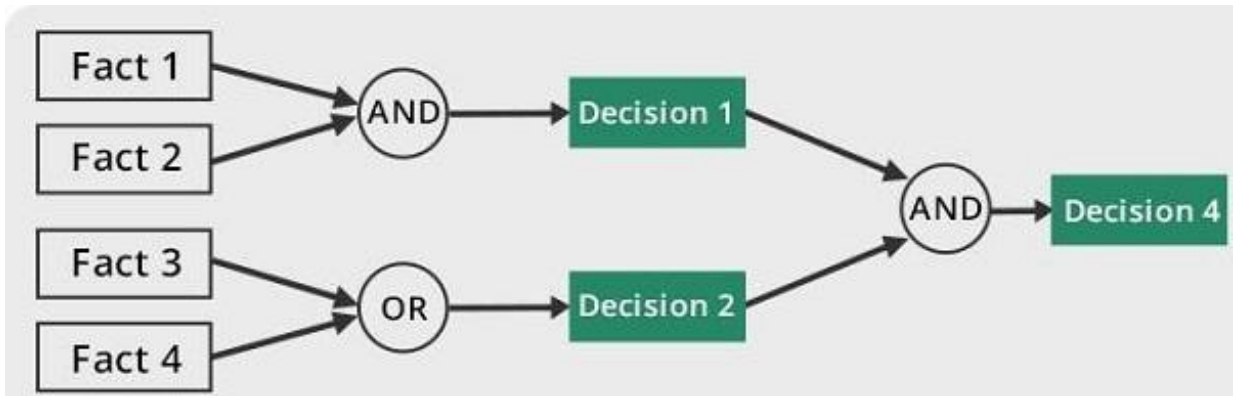
#### **Вперед Цепочка**

Это стратегия экспертной системы, чтобы ответить на вопрос: «**Что может произойти дальше?**»



Здесь механизм вывода следует цепочке условий и дериваций и, наконец, выводит результат. Он учитывает все факты и правила и сортирует их, прежде чем прийти к решению.

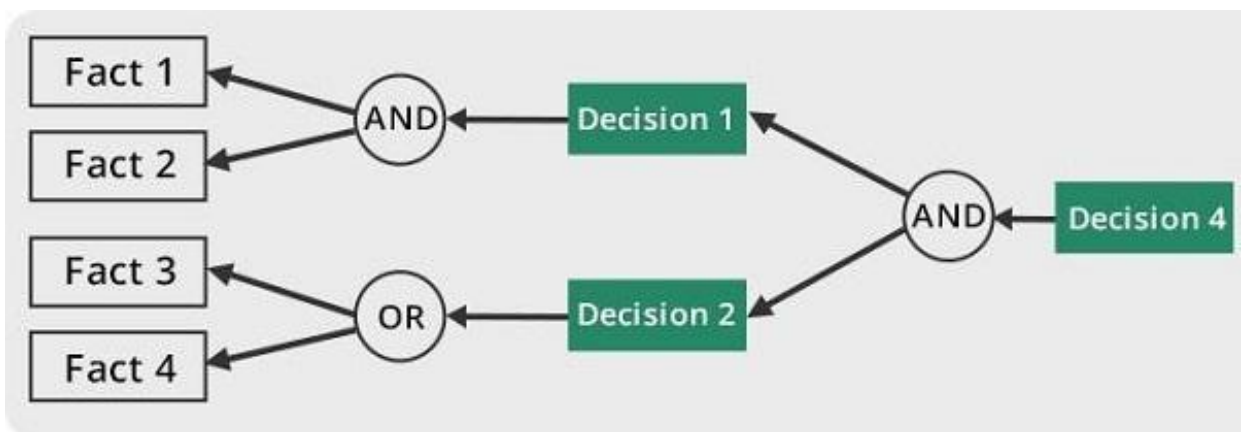
Эта стратегия используется для работы над выводом, результатом или эффектом. Например, прогноз состояния рынка акций как следствие изменения процентных ставок.



### Обратная цепочка

С помощью этой стратегии экспертная система узнает ответ на вопрос «Почему это произошло?»

На основе того, что уже произошло, Механизм вывода пытается выяснить, какие условия могли произойти в прошлом для этого результата. Эта стратегия используется для выяснения причины или причины. Например, диагностика рака крови у людей.



## **Пользовательский интерфейс**

Пользовательский интерфейс обеспечивает взаимодействие между пользователем ES и самой ES. Обычно это обработка естественного языка, которая используется пользователем, хорошо разбирающимся в предметной области. Пользователь ES не обязательно должен быть экспертом в области искусственного интеллекта.

Это объясняет, как ES пришел к конкретной рекомендации. Объяснение может появиться в следующих формах

- Естественный язык отображается на экране.
- Устные рассказы на естественном языке.
- Список номеров правил, отображаемых на экране.

Пользовательский интерфейс позволяет легко отследить достоверность выводов.

## **Требования эффективного пользовательского интерфейса ES**

- Это должно помочь пользователям достичь своих целей в кратчайшие сроки.
- Он должен быть разработан для работы с существующими или желаемыми методами работы пользователя.
- Его технология должна быть адаптирована к требованиям пользователя; не наоборот.
- Это должно эффективно использовать пользовательский ввод.

Это должно помочь пользователям достичь своих целей в кратчайшие сроки.

Он должен быть разработан для работы с существующими или желаемыми методами работы пользователя.

Его технология должна быть адаптирована к требованиям пользователя; не наоборот.

Это должно эффективно использовать пользовательский ввод.

## **Ограничения экспертных систем**

Ни одна технология не может предложить простое и полное решение. Большие системы являются дорогостоящими, требуют

значительного времени на разработку и компьютерных ресурсов. ES имеют свои ограничения, которые включают в себя –

- Ограничения технологии
- Сложное приобретение знаний
- ES сложно поддерживать
- Высокие затраты на разработку

### **Приложения Экспертной Системы**

В следующей таблице показано, где можно применять ES.

### **Эксперт Системные Технологии**

Есть несколько уровней доступных технологий ES. Технологии экспертных систем включают в себя –

**Среда разработки экспертных систем.** Среда разработки ES включает в себя оборудование и инструменты. Они –

- Рабочие станции, миникомпьютеры, мейнфреймы.
- Символические языки программирования высокого уровня, такие как **LIS t** Программирование (**LISP**) и **PRO** грамматика en **LOG** (**PROLOG**).

- Большие базы данных.

- **Инструменты** – они в значительной степени снижают усилия и затраты на разработку экспертной системы.

- Мощные редакторы и инструменты отладки с несколькими окнами.

- Они обеспечивают быстрое прототипирование

- Имеют встроенные определения модели, представления знаний и дизайна вывода.

- **Оболочки** . Оболочка – это всего лишь экспертная система без базы знаний. Оболочка обеспечивает разработчиков знаниями, механизмом вывода, пользовательским интерфейсом и средствами объяснения. Например, несколько оболочек приведены ниже –

- **Java Expert System Shell (JESS)**, которая предоставляет полностью разработанный Java API для создания экспертной системы.

- *Видван* , оболочка, разработанная в Национальном центре программных технологий в Мумбае. Она позволяет кодировать знания в форме правил IF-THEN.

<b>Заявка</b>	<b>Описание</b>
Домен дизайна	Дизайн объективов камер, автомобильный дизайн.
Медицинский домен	Системы диагностики, позволяющие вывести причину заболевания на основании наблюдаемых данных, проведения медицинских операций на людях.
Системы мониторинга	Постоянное сравнение данных с наблюдаемой системой или с предписанным поведением, таким как мониторинг утечек в длинном нефтепроводе.
Системы управления процессом	Управление физическим процессом на основе мониторинга.
Область знаний	Обнаружение неисправностей в транспортных средствах, компьютерах.
Финансы / Торговля	Обнаружение возможного мошенничества, подозрительных сделок, торговля на фондовом рынке, планирование авиаперевозок, планирование грузов.

**Среда разработки экспертных систем.** Среда разработки ES включает в себя оборудование и инструменты. Они –  
Рабочие станции, миникомпьютеры, мейнфреймы.

Символические языки программирования высокого уровня, такие как **LIS t** Программирование (LISP) и **PRO** грамматика en **LOG** (PROLOG).

Большие базы данных.

**Инструменты** – они в значительной степени снижают усилия и затраты на разработку экспертной системы.

Мощные редакторы и инструменты отладки с несколькими окнами.

Они обеспечивают быстрое прототипирование

Имеют встроенные определения модели, представления знаний и дизайна вывода.

**Оболочки**. Оболочка – это всего лишь экспертная система без базы знаний. Оболочка обеспечивает разработчиков знаниями, механизмом вывода, пользовательским интерфейсом и средствами объяснения. Например, несколько оболочек приведены ниже –

**Разработка экспертных систем: общие шаги**

Процесс разработки ES является итеративным. Шаги по разработке ES включают в себя –

**Определить проблемный домен**

- Проблема должна подходить для экспертной системы, чтобы решить ее.
- Найти экспертов в области задач для проекта ES.
- Установить экономическую эффективность системы.

**Дизайн системы**

- Определить технологию ES
- Знать и установить степень интеграции с другими системами и базами данных.
- Поймите, как концепции могут лучше всего представлять знания предметной области.

**Определить технологию ES**

Знать и установить степень интеграции с другими системами и базами данных.

Поймите, как концепции могут лучше всего представлять знания предметной области.

## **Разработайте прототип**

Из базы знаний: инженер знаний работает для –

Инженер по знаниям использует примеры для тестирования прототипа на предмет любых недостатков в производительности.

Конечные пользователи тестируют прототипы ES.

Тестировать и обеспечивать взаимодействие ES со всеми элементами ее среды, включая конечных пользователей, базы данных и другие информационные системы.

Хорошо документируйте проект ES.

Обучите пользователя использовать ES.

Постоянно обновляйте и обновляйте базу знаний.

Удовлетворение новых интерфейсов с другими информационными системами по мере развития этих систем.

**Доступность** – они легко доступны благодаря массовому производству программного обеспечения.

**Меньше себестоимости** – стоимость производства разумна. Это делает их доступными.

**Скорость** – они предлагают отличную скорость. Они уменьшают объем работы, которую выполняет человек.

**Меньше частоты ошибок** – уровень ошибок ниже по сравнению с человеческими ошибками.

**Снижение риска** – они могут работать в среде, опасной для человека.

**Устойчивый ответ** – они работают стабильно, не испытывая движения, напряжения или утомления.

## Глава 2. КЛАССИФИКАЦИЯ И КЛАСТЕРИЗАЦИЯ

### 2.1. Задача классификации

Классификацией является наиболее простой и одновременно наиболее часто решаемой задачей Data Mining. Ввиду распространенности задач классификации необходимо четкое понимания сути этого понятия.

Приведем несколько определений.

Классификация - системное распределение изучаемых предметов, явлений, процессов по родам, видам, типам, по каким-либо существенным признакам для удобства их исследования; группировка исходных понятий и расположение их в определенном порядке, отражающем степень этого сходства.

Классификация - упорядоченное по некоторому принципу множество объектов, которые имеют сходные классификационные признаки (одно или несколько свойств), выбранных для определения сходства или различия между этими объектами.

Классификация требует соблюдения следующих правил:

- в каждом акте деления необходимо применять только одно основание;
- деление должно быть соразмерным, т.е. общий объем видовых понятий должен равняться объему делимого родового понятия;
- члены деления должны взаимно исключать друг друга, их объемы не должны перекрещиваться;
- деление должно быть последовательным.

Различают:

- вспомогательную (искусственную) классификацию, которая производится по внешнему признаку и служит для придания множеству предметов (процессов, явлений) нужного порядка;
- естественную классификацию, которая производится по существенным признакам, характеризующим внутреннюю общность предметов и явлений. Она является результатом и важным средством научного исследования, т.к. предполагает и закрепляет результаты изучения закономерностей классифицируемых объектов.

В зависимости от выбранных признаков, их сочетания и процедуры деления понятий классификация может быть:

- простой - деление родового понятия только по признаку и только один раз до раскрытия всех видов. Примером такой классификации является дихотомия, при которой членами деления бывают только два понятия, каждое из которых является противоречащим другому (т.е. соблюдается принцип: "А и не А");
- сложной - применяется для деления одного понятия по разным основаниям и синтеза таких простых делений в единое целое. Примером такой классификации является периодическая система химических элементов.

Под классификацией будем понимать отнесение объектов (наблюдений, событий) к одному из заранее известных классов.

Классификацией - это закономерность, позволяющая делать вывод относительно определения характеристик конкретной группы. Таким образом, для проведения классификации должны присутствовать признаки, характеризующие группу, к которой принадлежит то или иное событие или объект (обычно при этом на основании анализа уже классифицированных событий формулируются некие правила).

Классификация относится к стратегии обучения с учителем (supervised learning), которое также именуют контролируемым или управляемым обучением.

Задачей классификации часто называют предсказание категориальной зависимой переменной (т.е. зависимой переменной, являющейся категорией) на основе выборки непрерывных и/или категориальных переменных.

Например, можно предсказать, кто из клиентов фирмы является потенциальным покупателем определенного товара, а кто - нет, кто воспользуется услугой фирмы, а кто - нет, и т.д. Этот тип задач относится к задачам бинарной классификации, в них зависимая переменная может принимать только два значения (например, да или нет, 0 или 1).

Другой вариант классификации возникает, если зависимая переменная может принимать значения из некоторого множества предопределенных классов. Например, когда необходимо предсказать, какую марку автомобиля захочет купить клиент. В



этих случаях рассматривается множество классов для зависимой переменной.

Классификация может быть одномерной (по одному признаку) и многомерной (по двум и более признакам).

Многомерная классификация была разработана биологами при решении проблем дискриминации для классифицирования организмов. Одной из первых работ, посвященных этому направлению, считают работу Р. Фишера (1930 г.), в которой организмы разделялись на подвиды в зависимости от результатов измерений их физических параметров. Биология была и остается наиболее востребованной и удобной средой для разработки многомерных методов классификации.

Рассмотрим задачу классификации на простом примере. Допустим, имеется база данных о клиентах туристического агентства с информацией о возрасте и доходе за месяц. Есть рекламный материал двух видов: более дорогой и комфортный отдых и более дешевый, молодежный отдых. Соответственно, определены два класса клиентов: класс 1 и класс 2. База данных приведена в таблице.

Таблица 2.1.

База данных клиентов туристического агентства

Код клиента	Возраст	Доход	Класс
1	18	25	1
2	22	100	1
3	30	70	1
4	32	120	1
5	24	15	2
6	25	22	1
7	32	50	2
8	19	45	2
9	22	75	1
10	40	90	2

**Задача.** Определить, к какому классу принадлежит новый клиент и какой из двух видов рекламных материалов ему стоит отсылать.

Для наглядности представим нашу базу данных в двухмерном измерении (возраст и доход), в виде множества объектов, принадлежащих классам 1 (оранжевая метка) и 2 (серая метка). На рис. 2.1 приведены объекты из двух классов.

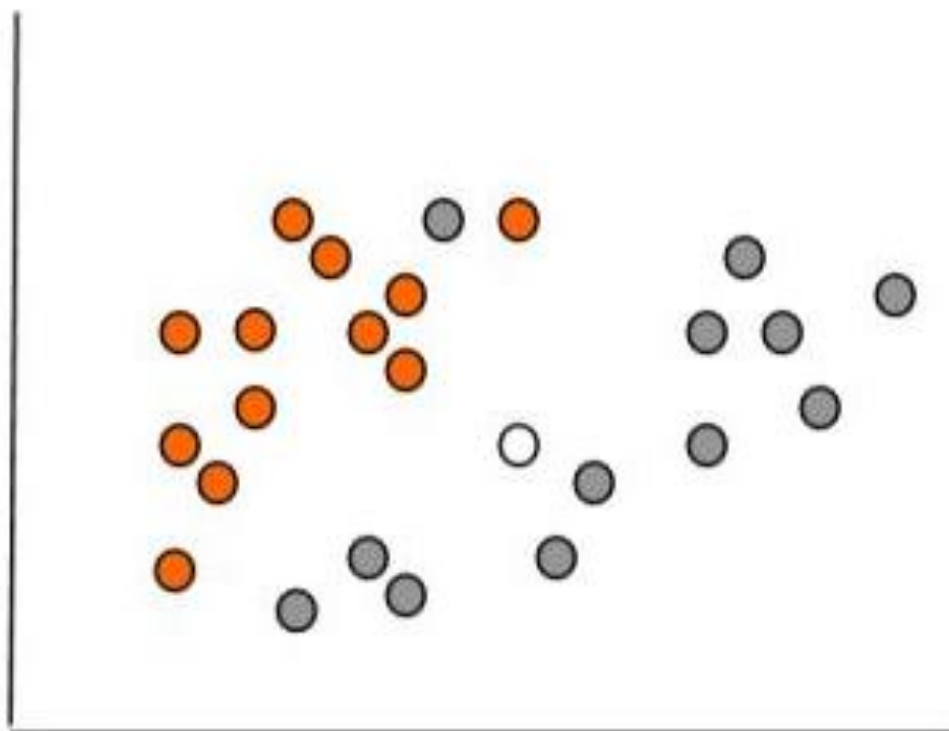


Рис. 2.1. Множество объектов базы данных в двухмерном измерении

Решение нашей задачи будет состоять в том, чтобы определить, к какому классу относится новый клиент, на рисунке обозначенный белой меткой.

## 2.2. Процесс классификации

Цель процесса классификации состоит в том, чтобы построить модель, которая использует прогнозирующие атрибуты в качестве входных параметров и получает значение зависимого атрибута. Процесс классификации заключается в разбиении множества объектов на классы по определенному критерию.

Классификатором называется некая сущность, определяющая, какому из предопределенных классов принадлежит объект по вектору признаков.

Для проведения классификации с помощью математических методов необходимо иметь формальное описание объекта, которым можно оперировать, используя математический аппарат классификации. Таким описанием в нашем случае выступает база данных.

Каждый объект (запись базы данных) несет информацию о некотором свойстве объекта.

Набор исходных данных (или выборку данных) разбивают на два множества: обучающее и тестовое.

**Обучающее множество** (training set) - множество, которое включает данные, используемые для обучения (конструирования) модели. Такое множество содержит входные и выходные (целевые) значения примеров.

Выходные значения предназначены для обучения модели.

**Тестовое** (test set) множество также содержит входные и выходные значения примеров. Здесь выходные значения используются для проверки работоспособности модели.

Процесс классификации состоит из двух этапов: конструирования модели и ее использования.

1. Конструирование модели: описание множества предопределенных классов.

- Каждый пример набора данных относится к одному предопределенному классу.

- На этом этапе используется обучающее множество, на нем происходит конструирование модели.

- Полученная модель представлена классификационными правилами, деревом решений или математической формулой.

2. Использование модели: классификация новых или неизвестных значений.

- Оценка правильности (точности) модели.
  - a) Известные значения из тестового примера сравниваются с результатами использования полученной модели.
  - b) Уровень точности - процент правильно классифицированных примеров в тестовом множестве.
  - c) Тестовое множество, т.е. множество, на котором тестируется построенная модель, не должно зависеть от обучающего множества.
- Если точность модели допустима, возможно использование модели для классификации новых примеров, класс которых неизвестен.

### **Методы, применяемые к задачам классификации**

#### **Методы, применяемые для решения задач классификации.**

Для классификации используются различные методы. Основные из них:

- классификация с помощью деревьев решений;
- байесовская (наивная) классификация;
- классификация при помощи искусственных нейронных сетей;
- классификация методом опорных векторов;
- статистические методы, в частности, линейная регрессия;
- классификация при помощи метода ближайшего соседа;
- классификация СВР-методом;
- классификация при помощи генетических алгоритмов.

#### **Точность классификации: оценка уровня ошибок.**

Оценка точности классификации может проводиться при помощи кросс- проверки. **Кросс-проверка** (Cross-validation) - это процедура оценки точности классификации на данных из тестового множества, которое также называют кросс-проверочным множеством. Точность классификации тестового множества сравнивается с точностью классификации обучающего множества. Если классификация тестового множества дает приблизительно

такие же результаты по точности, как и классификация обучающего множества, считается, что данная модель прошла кросс-проверку.

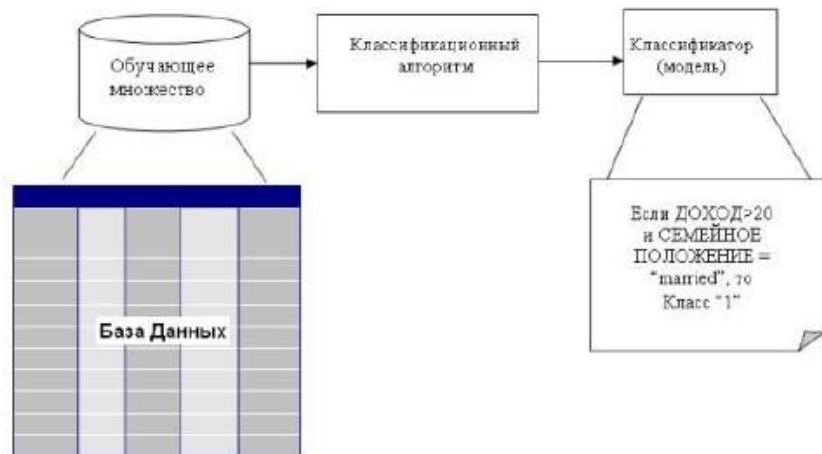


Рис. 2.2 Процесс классификации. Конструирование модели

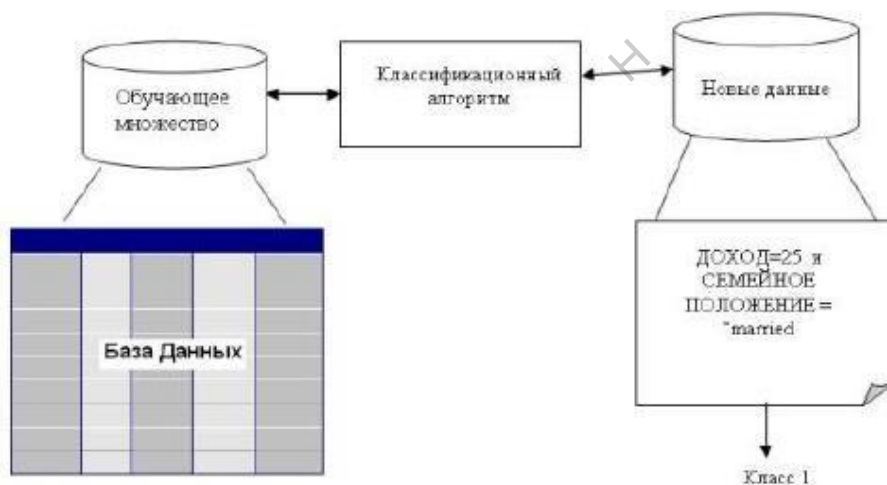


Рис. 2.3. Процесс классификации. Использование модели

Разделение на обучающее и тестовое множества осуществляется путем деления выборки в определенной пропорции, например, обучающее множество - две трети данных и тестовое - одна треть данных. Этот способ следует использовать для выборок с большим количеством примеров. Если же выборка имеет малые объемы, рекомендуется применять специальные методы, при использовании которых обучающая и тестовая выборки могут частично пересекаться.

## 2.3. Задача кластеризации

Задача кластеризации сходна с задачей классификации и является ее логическим продолжением. Отличие состоит в том, что классы изучаемого набора данных заранее непредопределены.

Синонимами термина "кластеризация" являются "автоматическая классификация", "обучение без учителя" и "таксономия".

Кластеризация предназначена для разбиения совокупности объектов на однородные группы (кластеры или классы). Если данные выборки представить как точки в признаковом пространстве, то задача кластеризации сводится к определению "сгущений точек".

Цель кластеризации - поиск существующих структур.

Кластеризация является описательной процедурой, она не делает никаких статистических выводов, но дает возможность провести разведочный анализ и изучить "структуру данных".

Самопонятие "кластер" определено неоднозначно: в каждом исследовании свои "кластеры". Кластер можно охарактеризовать как группу объектов, имеющих общесвойства.

Характеристиками кластера можно назвать два признака:

- внутренняя однородность;
- внешняя изолированность.

Вопрос, задаваемый аналитиками при решении многих задач, состоит в том, как организовать данные в наглядные структуры, т.е. развернуть таксономии. Наибольшее применение кластеризация первоначально получила в таких науках как биология, антропология, психология. Для решения экономических задач кластеризация длительное время мало использовалась из-за специфики экономических данных и явлений.

На рис. 2.4 схематически представлены задачи классификации и кластеризации.

Кластеры могут быть непересекающимися, или эксклюзивными (non-overlapping, exclusive), и пересекающимися (overlapping).

Следует отметить, что в результате применения различных методов кластерного анализа могут быть получены кластеры различной формы.

Например, возможны кластеры "цепочного" типа, когда кластеры представлены длинными "цепочками", кластеры удлиненной формы и т.д., а некоторые методы могут создавать кластеры произвольной формы.

Различные методы могут стремиться создавать кластеры определенных размеров (например, малых или крупных) либо предполагать в наборе данных наличие кластеров различного размера. Некоторые методы кластерного анализа особенно чувствительны к шумам или выбросам, другие - менее.

В результате применения различных методов кластеризации могут быть получены неодинаковые результаты, это нормально и является особенностью работы того или иного алгоритма. Данные особенности следует учитывать при выборе метода кластеризации.

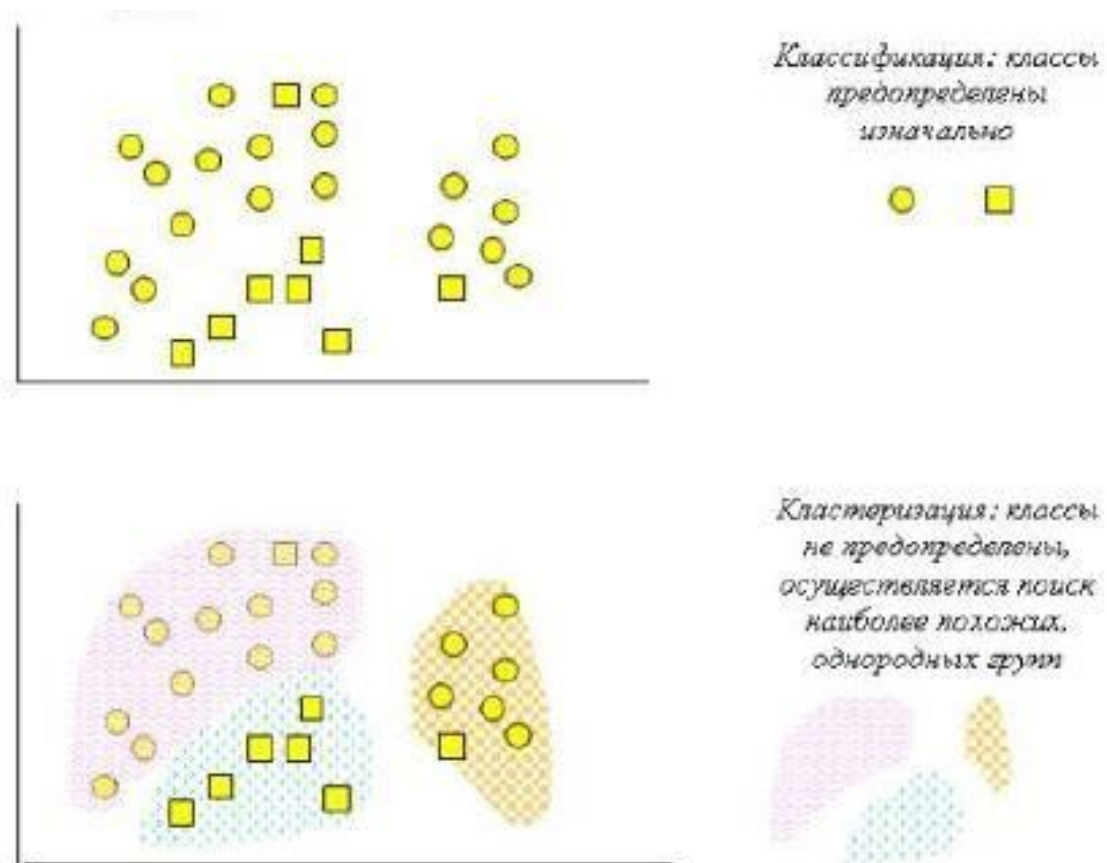


Рис. 2.4. Сравнение задач классификации и кластеризации

На сегодняшний день разработано более сотни различных алгоритмов кластеризации.

Приведем краткую характеристику подходов к кластеризации.

- Алгоритмы, основанные на разделении данных (Partitioning algorithms), в т.ч. итеративные:
  - разделение объектов на кластеры;
  - итеративное перераспределение объектов для улучшения кластеризации.
- Иерархические алгоритмы (Hierarchy algorithms):
  - агломерация: каждый объект первоначально является кластером, кластеры, соединяясь друг с другом, формируют больший кластер и т.д.
- Методы, основанные на концентрации объектов (Density-based methods):
  - основаны на возможности соединения объектов;
  - игнорируют шумы, нахождение кластеров произвольной формы
- Грид-методы (Grid-based methods):
  - квантование объектов в грид-структуры.
- Модельные методы (Model-based):
  - использование модели для нахождения кластеров, наиболее соответствующих данным.

### **Оценка качества кластеризации.**

Оценка качества кластеризации может быть проведена на основе следующих процедур:

- ручная проверка;
- установление контрольных точек и проверка на полученных кластерах;
- определение стабильности кластеризации путем добавления в модель новых переменных;
- создание и сравнение кластеров с использованием различных методов.

Разные методы кластеризации могут создавать разные кластеры, это является нормальным явлением. Однако создание схожих кластеров различными методами указывает на правильность кластеризации.



## **Процесс кластеризации.**

Процесс кластеризации зависит от выбранного метода и почти всегда является итеративным. Он может стать увлекательным процессом и включать множество экспериментов по выбору разнообразных параметров, например, меры расстояния, типа стандартизации переменных, количества кластеров и т.д. Полученные результаты требуют дальнейшей интерпретации, исследования и изучения свойств и характеристик объектов для возможности точного описания сформированных кластеров.

### **2.3.1. Применение кластерного анализа**

Опишем понятие "кластер" с математической точки зрения, а также рассмотрим методы кластерного анализа. Кластерный анализ включает в себя более 100 различных алгоритмов. В отличие от задач классификации, кластерный анализ не требует априорных предположений о наборе данных, не накладывает ограничения на представление исследуемых объектов, позволяет анализировать показатели различных типов данных (интервальным данным, частотам, бинарным данным). Все переменные должны измеряться в сравнимых шкалах.

Кластерный анализ может применяться к совокупностям временных рядов, здесь могут выделяться периоды схожести некоторых показателей и определяться группы временных рядов со схожей динамикой.

Задачи кластерного анализа можно объединить в следующие группы:

1. Разработка типологии или классификации.
2. Исследование полезных концептуальных схем группирования объектов.
3. Представление гипотез на основе исследования данных.
4. Проверка гипотез или исследований для определения, действительно ли типы (группы), выделенные тем или иным способом, присутствуют в имеющихся данных.

Как правило, при практическом использовании кластерного анализа одновременно решается несколько из указанных задач. Рассмотрим пример процедуры кластерного анализа.

№ примера	признак X	признак Y
1	27	19
2	11	46
3	25	15
4	36	27
5	35	25
6	10	43
7	11	44
8	36	24
9	26	14
10	26	14
11	9	45
12	33	23
13	27	16
14	10	47

Рис. 2.5. Набор данных A

Допустим, мы имеем набор данных A, состоящий из 14-ти примеров, у которых имеется по два признака X и Y. Данные по ним приведены рис.2.5.

Данные в табличной форме не носят информативный характер.

Представим переменные X и Y в виде диаграммы рассеивания, изображенной на рис.2.6.

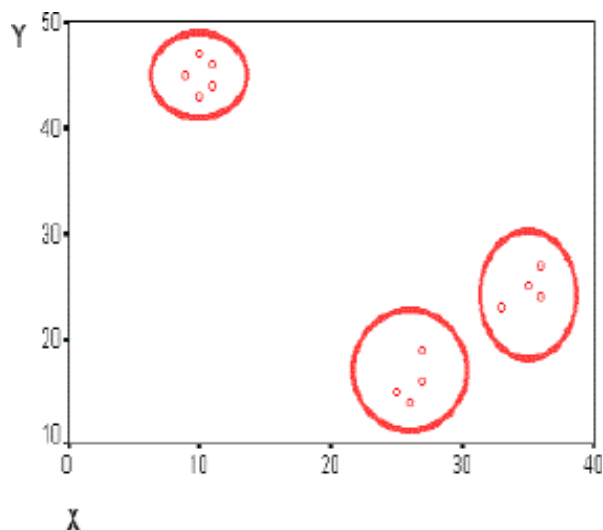


Рис. 2.6. Диаграмма рассеивания переменных X и Y

На рисунке мы видим несколько групп "похожих" примеров. Примеры (объекты), которые по значениям X и Y "похожи" друг на друга, принадлежат к одной группе (кластеру); объекты из разных кластеров не похожи друг на друга.

Критерием для определения схожести и различия кластеров расстояние между точками на диаграмме рассеивания. Это сходство можно "измерить", оно равно расстоянию между точками на графике. Способов определения меры расстояния между кластерами, называемой еще мерой близости, существует несколько. Наиболее распространенный способ - вычисление

$$D_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}.$$

евклидова расстояния между двумя точками  $i$  и  $j$  на плоскости, когда известны их координаты X и Y:

Кластер имеет следующие математические характеристики: центр, радиус, среднеквадратическое отклонение, размер кластера.

**Центр кластера** - это среднее геометрическое место точек в пространстве переменных.

**Радиус кластера** - максимальное расстояние точек от центра кластера. Кластеры могут быть перекрывающимися. Такая ситуация возникает, когда обнаруживается перекрытие кластеров. В этом случае невозможно при помощи математических процедур однозначно отнести объект к одному из двух кластеров. Такие объекты

называют спорными.

**Спорный объект** - это объект, который по мере сходства может быть отнесен к нескольким кластерам.

**Размер кластера** может быть определен либо по радиусу кластера, либо по среднеквадратичному отклонению объектов для этого кластера. Объект относится к кластеру, если расстояние от объекта до центра кластера меньше радиуса кластера. Если это условие выполняется для двух и более кластеров, объект является спорным.

Неоднозначность данной задачи может быть устранена экспертом или аналитиком.

Работа кластерного анализа опирается на два предположения. Первое предположение - рассматриваемые признаки объекта в принципе допускают желательное разбиение пула (совокупности) объектов на кластеры. Второе предположение - правильность выбора масштаба или единиц измерения признаков.

Выбор масштаба в кластерном анализе имеет большое значение. Рассмотрим пример.

Представим себе, что данные признака  $X$  в наборе данных  $A$  на два порядка больше данных признака  $Y$  (значения переменной  $X$  находятся в диапазоне от 100 до 700, а значения переменной  $Y$  - в диапазоне от 0 до 1).

Тогда, при расчете величины расстояния между точками, отражающими положение объектов в пространстве их свойств, переменная, имеющая большие значения, т.е. переменная  $X$ , будет практически полностью доминировать над переменной с малыми значениями, т.е. переменной  $Y$ . Таким образом из-за неоднородности единиц измерения признаков становится невозможно корректно рассчитать расстояния между точками.

Эта проблема решается при помощи предварительной стандартизации переменных.

**Стандартизация** (standardization) или **нормирование** (normalization) приводит значения всех преобразованных переменных к единому диапазону значений путем выражения через отношение этих значений к некоей величине, отражающей определенные свойства конкретного признака. Существуют различные способы нормирования исходных данных.

Два наиболее распространенных способа:

- деление исходных данных на среднеквадратичное отклонение соответствующих переменных;
- вычисление Z-вклада или стандартизованного вклада.

Наряду со стандартизацией переменных, существует вариант придания каждой из них определенного коэффициента важности, или веса, который бы отражал значимость соответствующей переменной. В качестве весов могут выступать экспертные оценки, полученные в ходе опроса экспертов - специалистов предметной области. Полученные произведения нормированных переменных на соответствующие веса позволяют получать расстояния между точками в многомерном пространстве с учетом неодинакового веса переменных.

В ходе экспериментов возможно сравнение результатов, полученных с учетом экспертных оценок и без них, и выбор лучшего из них.

### **Методы кластерного анализа**

Методы кластерного анализа можно разделить на две группы:

- иерархические;
- неиерархические.

Каждая из групп включает множество подходов и алгоритмов.

Используя различные методы кластерного анализа, можно получить различные решения для одних и тех же данных. Это считается нормальным явлением.

Рассмотрим иерархические и неиерархические методы подробно.

### **Иерархические методы кластерного анализа.**

Суть иерархической кластеризации состоит в последовательном объединении меньших кластеров в большие или разделении больших кластеров на меньшие.

**Иерархические агломеративные методы (Agglomerative Nesting, AGNES).**

Эта группа методов характеризуется последовательным объединением исходных элементов и соответствующим уменьшением числа кластеров.

В начале работы алгоритма все объекты являются отдельными кластерами. На первом шаге наиболее похожие объекты объединяются в кластер. На последующих шагах объединение продолжается до тех пор, пока все объекты не будут составлять

один кластер.

**Иерархические дивизимные (делимые) методы (DIvisive ANAlysis, DIANA).**

Эти методы являются логической противоположностью агломеративным методам. В начале работы алгоритма все объекты принадлежат одному кластеру, который на последующих шагах делится на меньшие кластеры, в результате образуется последовательность расщепляющих групп.

Принцип работы описанных выше групп методов в виде дендрограммы показан на рис.7.4.

Программная реализация алгоритмов кластерного анализа широко представлена в различных инструментах Data Mining, которые позволяют решать задачи достаточно большой размерности.

Иерархические методы кластеризации различаются правилами построения кластеров. В качестве правил выступают критерии, которые используются при решении вопроса "схожести" объектов при их объединении в группу (агломеративные методы) либо разделения на группы (дивизимные методы).

Иерархические методы кластерного анализа используются при небольших объемах наборов данных.

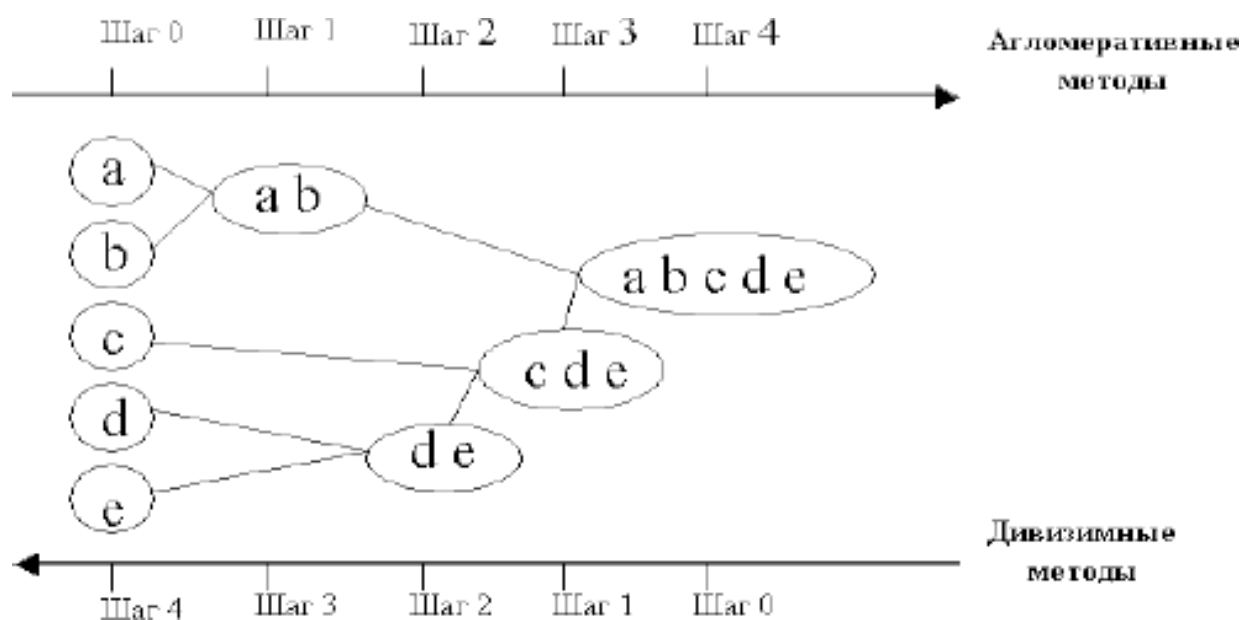


Рис. 2.7. Дендрограмма агломеративных и дивизимных методов

Преимуществом иерархических методов кластеризации является их наглядность.

Иерархические алгоритмы связаны с построением дендрограмм (от греческого dendron - "дерево"), которые являются результатом иерархического кластерного анализа.

Дендрограмма описывает близость отдельных точек и кластеров друг к другу, представляет в графическом виде последовательность объединения (разделения) кластеров.

Дендрограмма (dendrogram) - древовидная диаграмма, содержащая  $n$  уровней, каждый из которых соответствует одному из шагов процесса последовательного укрупнения кластеров.

Дендрограмму также называют древовидной схемой, деревом объединения кластеров, деревом иерархической структуры.

Дендрограмма представляет собой вложенную группировку объектов, которая изменяется на различных уровнях иерархии.

Существует много способов построения дендограмм. В дендограмме объекты могут располагаться вертикально или горизонтально.

### **Меры сходства.**

Для вычисления расстояния между объектами используются различные меры сходства (меры подобия), называемые также метриками или функциями расстояний.

**Евклидово расстояние** - наиболее популярная мера сходства.

**Квадрат евклидова расстояния.** Для придания больших весов более отдаленным друг от друга объектам можем воспользоваться квадратом евклидова расстояния путем возведения в квадрат стандартного евклидова расстояния.

**Манхэттенское расстояние** (расстояние городских кварталов), также называемое "хэмминговым" или "сити-блок" расстоянием. Это расстояние рассчитывается как среднее разностей по координатам. В большинстве случаев эта мера расстояния приводит к результатам, подобным расчетам расстояния евклида. Однако, для этой меры влияние отдельных выбросов меньше, чем при использовании евклидова расстояния, поскольку здесь координаты не возводятся в квадрат.

**Расстояние Чебышева.** Это расстояние стоит использовать, когда необходимо определить два объекта как "различные", если они отличаются по какому-то одному измерению.

**Процент несогласия.** Это расстояние вычисляется, если данные являются категориальными.

**Методы объединения или связи.**

Возникает следующий вопрос – как определить расстояния между кластерами? Существуют различные правила, называемые методами объединения или связи для двух кластеров.

**Метод ближнего соседа или одиночная связь.** Здесь расстояние между двумя кластерами определяется расстоянием между двумя наиболее близкими объектами (ближайшими соседями) в различных кластерах. Этот метод позволяет выделять кластеры сколь угодно сложной формы при условии, что различные части таких кластеров соединены цепочками близких друг к другу элементов. В результате работы этого метода кластеры представляются длинными "цепочками" или "волокнистыми" кластерами, "сцепленными вместе" только отдельными элементами, которые случайно оказались ближе остальных друг к другу.

**Метод наиболее удаленных соседей или полная связь.** Здесь расстояния между кластерами определяются наибольшим расстоянием между любыми двумя объектами в различных кластерах (т.е. "наиболее удаленными соседями"). Метод хорошо использовать, когда объекты действительно происходят из различных "рощ". Если же кластеры имеют в некотором роде удлинненную форму или их естественный тип является "цепочечным", то этот метод не следует использовать.

**Метод Варда (Ward's method).** В качестве расстояния между кластерами берется прирост суммы квадратов расстояний объектов до центров кластеров, получаемый в результате их объединения. В отличие от других методов кластерного анализа для оценки расстояний между кластерами, здесь используются методы дисперсионного анализа. На каждом шаге алгоритма объединяются такие два кластера, которые приводят к минимальному увеличению целевой функции, т.е. внутригрупповой суммы квадратов. Этот метод направлен на объединение близко расположенных кластеров и "стремится" создавать кластеры малого размера.

**Метод невзвешенного попарного среднего** (метод невзвешенного попарного арифметического среднего - unweighted pair-group method using arithmetic averages, UPGMA). В качестве



расстояния между двумя кластерами берется среднее расстояние между всеми парами объектов в них. Этот метод следует использовать, если объекты действительно происходят из различных "роц", в случаях присутствия кластеров "цепочного" типа, при предположении неравных размеров кластеров.

**Метод взвешенного попарного среднего** (метод взвешенного попарного арифметического среднего - weighted pair-group method using arithmetic averages, WPGMA). Этот метод похож на метод невзвешенного попарного среднего, разница состоит лишь в том, что здесь в качестве весового коэффициента используется размер кластера (число объектов, содержащихся в кластере). Этот метод рекомендуется использовать именно при наличии предположения о кластерах разных размеров.

**Невзвешенный центроидный метод** (метод невзвешенного попарного центроидного усреднения - unweighted pair-group method using the centroid average). В качестве расстояния между двумя кластерами в этом методе берется расстояние между их центрами тяжести.

**Взвешенный центроидный метод** (метод взвешенного попарного центроидного усреднения - weighted pair-group method using the centroid average, WPGMC). Этот метод похож на предыдущий, разница состоит в том, что для учета разницы между размерами кластеров (числе объектов в них), используются веса. Этот метод предпочтительно использовать в случаях, если имеются предположения относительно существенных отличий в размерах кластеров.

#### **Определение количества кластеров.**

Существует проблема определения числа кластеров. Иногда можно априорно определить это число. Однако в большинстве случаев число кластеров определяется в процессе агломерации/разделения множества объектов.

Процессу группировки объектов в иерархическом кластерном соответствии соответствует постепенное возрастание коэффициента, называемого критерием E. Скачкообразное увеличение значения критерия E можно определить как характеристику числа кластеров, которые действительно существуют в исследуемом наборе данных. Таким образом, этот способ сводится к определению скачкообразного увеличения некоторого коэффициента, который

характеризует переход от сильно связанного к слабо связанному состоянию объектов.

### **Итеративные методы кластерного анализа.**

При большом количестве наблюдений иерархические методы кластерного анализа не пригодны. В таких случаях используют неиерархические методы, основанные на разделении, которые представляют собой итеративные методы дробления исходной совокупности. В процессе деления новые кластеры формируются до тех пор, пока не будет выполнено правило остановки.

Такая неиерархическая кластеризация состоит в разделении набора данных на определенное количество отдельных кластеров. Существует два подхода. Первый заключается в определении границ кластеров как наиболее плотных участков в многомерном пространстве исходных данных, т.е. определение кластера там, где имеется большое "сгущение точек". Второй подход заключается в минимизации меры различия объектов.

### **Алгоритм k-средних (k-means).**

Наиболее распространен среди неиерархических методов алгоритм k-средних, также называемый быстрым кластерным анализом. В отличие от иерархических методов, которые не требуют предварительных предположений относительно числа кластеров, для возможности использования этого метода необходимо иметь гипотезу о наиболее вероятном количестве кластеров.

Алгоритм k-средних строит k кластеров, расположенных на возможно больших расстояниях друг от друга. Основной тип задач, которые решает алгоритм k-средних, - наличие предположений (гипотез) относительно числа кластеров, при этом они должны быть различны настолько, насколько это возможно. Выбор числа k может базироваться на результатах предшествующих исследований, теоретических соображениях или интуиции.

Общая идея алгоритма: заданное фиксированное число k кластеров наблюдения сопоставляются кластерам так, что средние в кластере (для всех переменных) максимально возможно отличаются друг от друга.

### **Описание алгоритма**

Первоначальное распределение объектов по кластерам.

Выбирается число k, и на первом шаге эти точки считаются

"центрами" кластеров. Каждому кластеру соответствует один центр. Выбор начальных центроидов может осуществляться следующим образом:

- выбор  $k$ -наблюдений для максимизации начальнорасстояния;
- случайный выбор  $k$ -наблюдений;
- выбор первых  $k$ -наблюдений.

В результате каждый объект назначен определенному кластеру.

## 2. Итеративный процесс.

Вычисляются центры кластеров, которыми затем и далее считаются координатные средние кластеров. Объекты опять перераспределяются.

Процесс вычисления центров и перераспределения объектов продолжается до тех пор, пока не выполнено одно из условий:

- кластерные центры стабилизировались, т.е. все наблюдения
  - принадлежат кластеру, которому принадлежали до текущей итерации;
  - число итераций равно максимальному числу итераций.

На рис.2.8 приведен пример работы алгоритма  $k$ -средних для  $k$ , равного двум.

Выбор числа кластеров является сложным вопросом. Если нет предположений относительно этого числа, рекомендуют создать 2 кластера, затем 3, 4, 5 и т.д., сравнивая полученные результаты.

### **Проверка качества кластеризации.**

После получения результатов кластерного анализа методом  $k$ -средних следует проверить правильность кластеризации (т.е. оценить, насколько кластеры отличаются друг от друга).

Для этого рассчитываются средние значения для каждого кластера. При хорошей кластеризации должны быть получены сильно отличающиеся средние для всех измерений или хотя бы большей их части.

Достоинства алгоритма  $k$ -средних:

- простота использования;
- быстрота использования;
- понятность и прозрачность алгоритма.

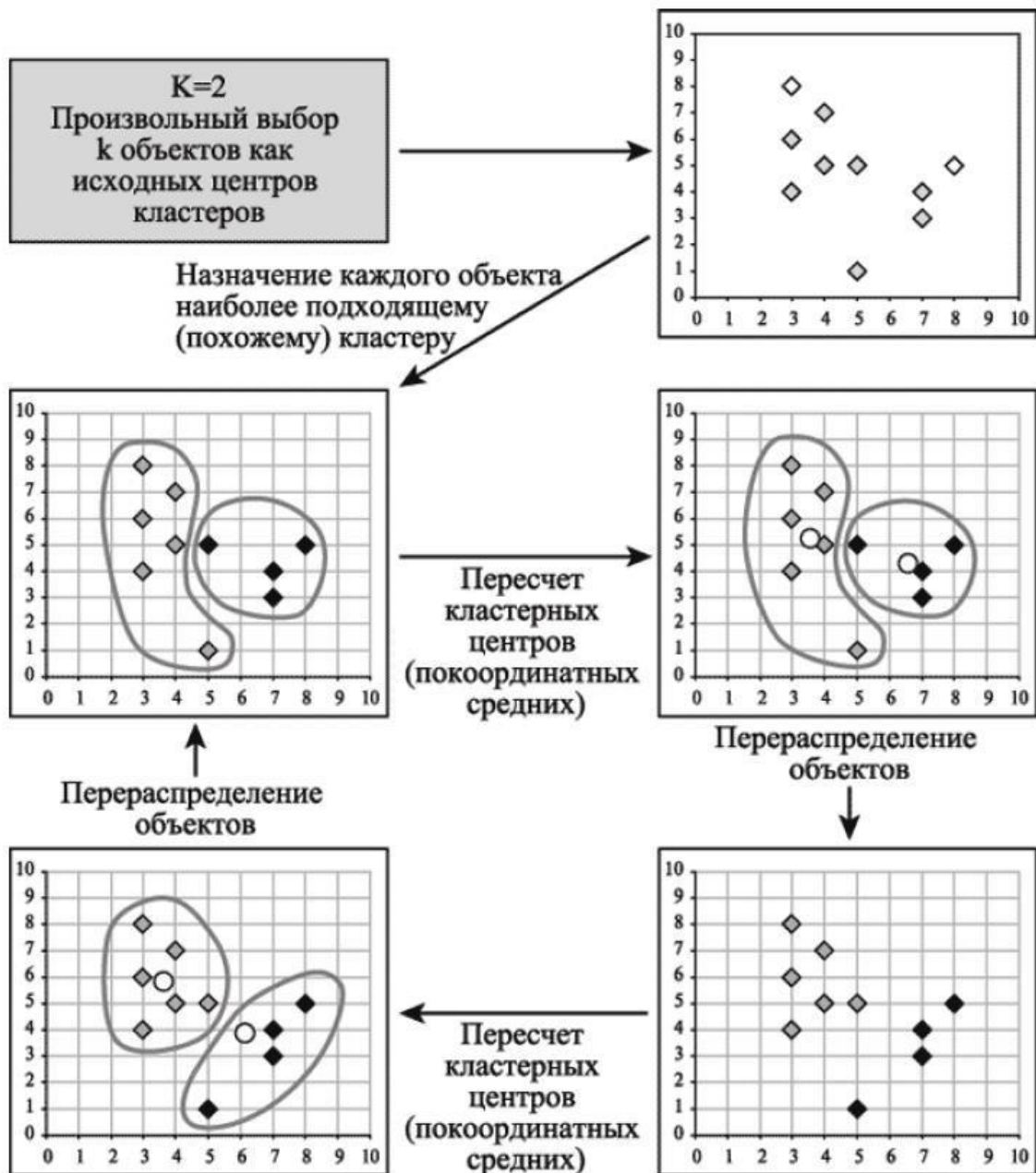


Рис. 2.8. Пример работы алгоритма k-средних (k=2)

Недостатки алгоритма k-средних:

- алгоритм слишком чувствителен к выбросам, которые могут исказить среднее. Возможным решением этой проблемы является использование модификации алгоритма - алгоритм k-медианы;
- алгоритм может медленно работать на больших базах данных. Возможным решением данной проблемы является

использование выборки данных.

**Алгоритм РАМ** ( partitioning around Medoids).

РАМ является модификацией алгоритма k-средних, алгоритмом k- медианы (k-medoids). Алгоритм менее чувствителен к шумам и выбросам данных, чем алгоритм k-means, поскольку медиана меньше подвержена влияниям выбросов. РАМ эффективен для небольших баз данных, но его не следует использовать для больших наборов данных.

**Предварительное сокращение размерности.**

Более понятные и прозрачные результаты кластеризации могут быть получены, если вместо множества исходных переменных использовать некие обобщенные переменные или критерии, содержащие в сжатом виде информацию о связях между переменными. Т.е. возникает задача понижения размерности данных. Она может решаться при помощи различных методов; один из наиболее распространенных - факторный анализ.

**Факторный анализ**- это метод, применяемый для изучения взаимосвязей между значениями переменных. Факторный анализ преследует две цели:

- сокращение числа переменных;
- классификацию переменных - определение структуры взаимосвязей между переменными.

Соответственно, факторный анализ может использоваться для решения задач сокращения размерности данных или для решения задач классификации.

Критерии или главные факторы, выделенные в результате факторного анализа, содержат в сжатом виде информацию о существующих связях между переменными. Эта информация позволяет получить лучшие результаты кластеризации и лучше объяснить семантику кластеров. Самим факторам может быть сообщен определенный смысл.

При помощи факторного анализа большое число переменных сводится к меньшему числу независимых влияющих величин, которые называются факторами.

Фактор в "сжатом" виде содержит информацию о нескольких переменных. В один фактор объединяются переменные, которые сильно коррелируют между собой. В результате факторного анализа отыскиваются такие комплексные факторы, которые как

можно более полно объясняют связи между рассматриваемыми переменными.

На первом шаге факторного анализа осуществляется стандартизация значений переменных. Факторный анализ опирается на гипотезу о том, что анализируемые переменные являются косвенными проявлениями сравнительно небольшого числа неких скрытых факторов.

Факторный анализ - это совокупность методов, ориентированных на выявление и анализ скрытых зависимостей между наблюдаемыми переменными. Скрытые зависимости также называют латентными.

Один из методов факторного анализа - метод главных компонент - основан на предположении о независимости факторов друг от друга.

Выбирая между иерархическими и неиерархическими методами, необходимо учитывать следующие их особенности.

**Неиерархические методы** выявляют более высокую устойчивость по отношению к шумам и выбросам, некорректному выбору метрики, включению незначимых переменных в набор, участвующий в кластеризации. Ценой, которую приходится платить за эти достоинства метода, является слово "априори". Необходимо заранее определить количество кластеров, количество итераций или правило останова, а также некоторые другие параметры кластеризации.

Если нет предположений относительно числа кластеров, рекомендуют использовать иерархические алгоритмы. Однако если объем выборки не позволяет это сделать, возможный путь - проведение ряда экспериментов с различным количеством кластеров, например, начать разбиение совокупности данных с двух групп и, постепенно увеличивая их количество, сравнивать результаты. Засчет такого "варьирования" результатов достигается достаточно большая гибкость кластеризации.

**Иерархические методы**, в отличие от неиерархических, отказываются от определения числа кластеров, а строят полное дерево вложенных кластеров.

Сложности иерархических методов кластеризации: ограничение объема набора данных; выбор меры близости; негибкость полученных классификаций.

Преимущество этой группы методов в сравнении с неиерархическими методами – их наглядность и возможность получить детальное представление о структуре данных.

При использовании иерархических методов существует возможность достаточно легко идентифицировать выбросы в наборе данных и, в результате, повысить качество данных.

Эта процедура лежит в основе двухшагового алгоритма кластеризации. Такой набор данных в дальнейшем может быть использован для проведения неиерархической кластеризации.

Иерархические методы не могут работать с большими наборами данных, а использование некоторой выборки, т.е. части данных, могло бы позволить применять эти методы.

Результаты кластеризации могут не иметь достаточного статистического обоснования. С другой стороны, при решении задач кластеризации допустима нестатистическая интерпретация полученных результатов, а также достаточно большое разнообразие вариантов понятия кластера. Такая нестатистическая интерпретация дает возможность аналитику получить удовлетворяющие его результаты кластеризации, что при использовании других методов часто бывает затруднительным.

В связи с появлением сверхбольших баз данных, появились новые требования, которым должен удовлетворять алгоритм кластеризации.

Основное из них - это масштабируемость алгоритма (масштабируемость здесь означает, что с ростом объемов данных время, затрачиваемое на обучение, и кластеризацию, растет линейно).

Отметим также другие свойства, которым должен удовлетворять алгоритм кластеризации: независимость результатов от порядка входных данных; независимость параметров алгоритма от входных данных.

Разработаны алгоритмы, в которых методы иерархической кластеризации интегрированы с другими методами. К таким алгоритмам относятся: BIRCH, CURE, CHAMELEON, ROCK.

**Алгоритм BIRCH** (Balanced Iterative Reducing and Clustering using Hierarchies).

Благодаря обобщенным представлениям кластеров, скорость кластеризации увеличивается, алгоритм при этом обладает

большим масштабированием.

В этом алгоритме реализован двухэтапный процесс кластеризации.

В ходе первого этапа формируется предварительный набор кластеров. На втором этапе к выявленным кластерам применяются другие алгоритмы кластеризации - пригодные для работы в оперативной памяти.

Аналогия, описывающая этот алгоритм. Если каждый элемент данных представить себе как бусину, лежащую на поверхности стола, то кластеры бусин можно "заменить" теннисными шариками и перейти к более детальному изучению кластеров теннисных шариков. Число бусин может оказаться достаточно велико, однако диаметр теннисных шариков можно подобрать таким образом, чтобы на втором этапе можно было, применив традиционные алгоритмы кластеризации, определить действительную сложную форму кластеров.

#### **Алгоритм WaveCluster.**

WaveCluster представляет собой алгоритм кластеризации на основе волновых преобразований. В начале работы алгоритма данные обобщаются путем наложения на пространство данных многомерной решетки. На дальнейших шагах алгоритма анализируются не отдельные точки, а обобщенные характеристики точек, попавших в одну ячейку решетки. В результате такого обобщения необходимая информация уместается в оперативной памяти. На последующих шагах для определения кластеров алгоритм применяет волновое преобразование к обобщенным данным.

Главные особенности WaveCluster:

- сложность реализации;
- алгоритм может обнаруживать кластеры произвольных форм;
- алгоритм не чувствителен к шумам;
- алгоритм применим только к данным низкой размерности.

#### **Алгоритм CLARA (Clustering LARge Applications).**

Алгоритм CLARA извлекает множество образцов из базы данных. Кластеризация применяется к каждому из образцов, на выходе алгоритма предлагается лучшая кластеризация.



Для больших баз данных этот алгоритм эффективнее, чем алгоритм PAM. Эффективность алгоритма зависит от выбранного в качестве образца набора данных. Хорошая кластеризация на выбранном наборе может не дать хорошую кластеризацию на всем множестве данных.

### **Алгоритмы Clarans, CURE, DBScan.**

Алгоритм Clarans (Clustering Large Applications based upon RANdomized Search) формулирует задачу кластеризации как случайный поиск в графе. В результате работы этого алгоритма совокупность узлов графа представляет собой разбиение множества данных на число кластеров, определенное пользователем. "Качество" полученных кластеров определяется при помощи критериальной функции. Алгоритм Clarans сортирует все возможные разбиения множества данных в поисках приемлемого решения. Поиск решения останавливается в том узле, где достигается минимум среди предопределенного числа локальных минимумов.

Среди новых масштабируемых алгоритмов также можно отметить алгоритм CURE - алгоритм иерархической кластеризации, и алгоритм DBScan, где понятие кластера формулируется с использованием концепции плотности (density).

Основным недостатком алгоритмов BIRCH, Clarans, CURE, DBScan является то, что они требуют задания некоторых порогов плотности точек, а это не всегда приемлемо. Эти ограничения обусловлены тем, что описанные алгоритмы ориентированы на сверхбольшие базы данных и не могут пользоваться большими вычислительными ресурсами.

## Глава 3. МЕТОД НЕЧЕТКИХ МНОЖЕСТВ

### 3.1. Основные понятия

Основное определение нечеткой логики включает в себя понятия объединения, пересечения и дополнения множеств (через характеристическую функцию; задать можно различными способами), понятие нечеткого отношения, а также одно из важнейших понятий - понятие лингвистической переменной. Даже такой минимальный набор определений позволяет использовать нечеткую логику в некоторых приложениях, для большинства же необходимо задать ещё и правило вывода (и оператор импликации).

Процесс нечеткого вывода представляет собой некоторую процедуру или алгоритм получения нечетких заключений на основе нечетких условий или предпосылок с использованием рассмотренных выше понятий нечеткой логики. Этот процесс соединяет в себе все основные концепции теории нечетких множеств: функции принадлежности, лингвистические переменные, нечеткие логические операции, методы нечеткой импликации и нечеткой композиции.

Под нечетким множеством понимается совокупность

$$A = \{(x, \mu_A(x)) | x \in X\},$$

где  $A$  – это нечеткое множество,  $X$  – универсальное множество, а  $\mu_A(x)$  - функция принадлежности (характеристическая функция), характеризующая степень принадлежности элемента  $x$  нечеткому множеству  $A$ .

Системы нечеткого вывода предназначены для реализации процесса нечеткого вывода и служат концептуальным базисом всей современной нечеткой логики. Достигнутые успехи в применении этих систем для решения широкого класса задач управления послужили основой становления нечеткой логики как прикладной науки с богатым спектром приложений. Системы нечеткого вывода позволяют решать задачи автоматического управления, классификации данных, распознавания образов, принятия решений, машинного обучения и многие другие.

### Определение 3.1. Нечеткие множества.

Пусть  $X$  универсальное множество, элементы которого обозначены через  $x$ . Принадлежность элементов в множестве  $A$  из  $X$  часто рассматривается как характеристическая функция  $\mu_A$  из  $X$  в  $\{0,1\}$ , т.е.

$$\mu_A(x) = \begin{cases} 1, & \text{если } x \in A, \\ 0, & \text{если } x \notin A. \end{cases} \quad (3.1)$$

Множество  $\{0,1\}$  называется множеством оценок.

Если предположить, что не множество  $\{0,1\}$ , а действительный интервал  $[0,1]$  является множеством оценок, тогда  $A$  будет нечетким множеством. В таком случае  $\mu_A(x)$  будет называться функцией принадлежности. Чем больше  $\mu_A(x)$  близко к 1, тем больше  $x$  принадлежит  $A$  [1].

Нечеткое множество  $A$  может характеризоваться множеством пар

$$A = \{(x, \mu_A(x)), x \in X\} \quad (3.2)$$

В общем случае, нечетким множеством  $A \subseteq X$  называется совокупность пар вида  $(x, \mu_A(x))$ , где  $x \in X, \mu_A(x): X \rightarrow [0,1]$  (иногда  $\mu_A(x): X \rightarrow L$  - структура типа решетки).

Нечеткое множество может быть конечным и бесконечным. Когда  $X$ -конечное множество, т.е.  $X = \{x_1, \dots, x_n\}$ , тогда нечеткое множество на  $X$  определяется следующим образом:

$$A = \mu_A(x_1)/x_1 + \dots + \mu_A(x_n)/x_n = \sum_{i=1}^n \mu_A(x_i)/x_i. \quad (3.3)$$

Когда  $X$  – бесконечное, тогда

$$A = \int_x \mu_A(x)/x. \quad (3.4)$$

**Пример.** Нечеткое множество  $A$  представлено в следующем виде:

$$A=0.1/1+0.3/2+0.5/3+0.7/4+0.8/5+0.9/6+0.95/7+1.0/8+0.97/9+ \\ +0.9/10+0.8/11+0.7/12+0.5/13+0.3/14+0.1/15.$$

Тогда классическая версия этого множества имеет вид:

$$A=\{1,2,3,4,5,6,7,8,9,10,11,12,13,14,15\} \text{ или}$$

$$A=1/1+1/2+1/3+1/4+1/5+1/6+1/7+1/8+1/9+ \\ +1/10+1/11+1/12+1/13+1/14+1/15.$$

Графическое изображение нечеткого множества  $A$  приведено на рис. 3.1.

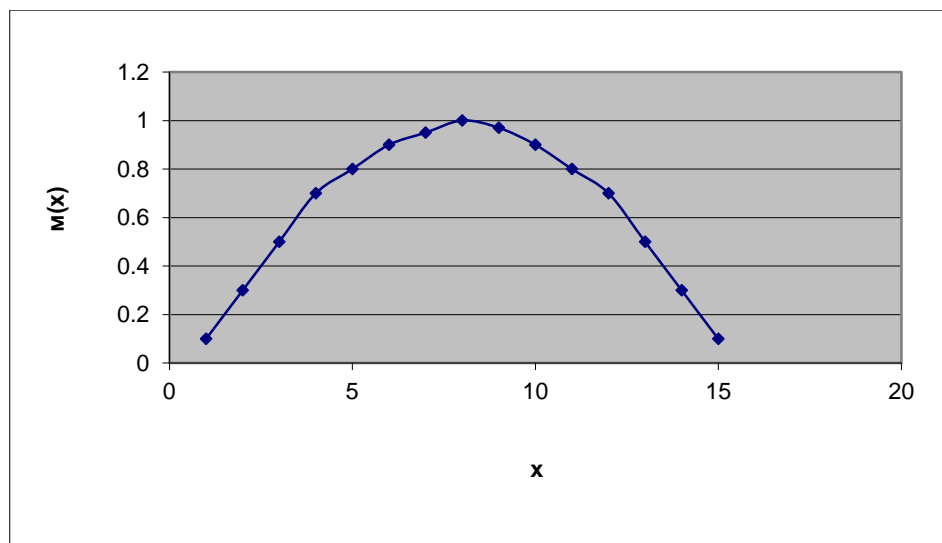


Рис.3.1. Нечеткое множество

Нечеткое множество может быть представлено в аналитической форме.

Пример.

$A =$  «действительное число близкое к 8»:

$$A = \int_R \frac{1}{1+(x+8)^2} / x.$$

Нечеткое множество данного типа, т.е. когда функция принадлежности его является обычной (crisp) функцией или степени принадлежности являются обычными числами, называется нечетким множеством 1-го типа. Функция принадлежности

нечеткого множества сама может быть нечетким множеством. Тогда нечеткое множество типа 2 определяется как [2,3]

$$\mu : X \times [0,1] \rightarrow [0,1].$$

Нечетким множеством типа  $m$  называется нечеткое множество в  $X$ , у которого значениями функции принадлежности является нечеткое множество типа  $m-1$ .

В [4,5] рассмотрен другой тип нечетких множеств. Когда значение функции принадлежности является случайной переменной. В этом случае вероятностное множество  $A$  в  $X$  определяется характеристической функцией

$$\mu_A : (x, \omega) \in X \times \Omega \rightarrow \mu(x, \omega) \in Q_c,$$

где  $\mu_A(x^*)$  - является  $(B, Bc)$  - измеряемой функцией для каждого фиксированного  $x \in X$ .

Имеются и другие расширения нечетких множеств. Если  $\forall x \in X, \exists! x^*$ , для которого условия

$$\mu(x^*) \geq \mu(x),$$

$$\mu(x) \geq \mu(x^*)$$

удовлетворяются, то  $x^*$  называют максимальным или минимальным значением функции принадлежности  $\mu(x)$

$$\begin{aligned} \mu(x^*) &= \max_{x \in X} \mu(x) \\ \mu(x^*) &= \min_{x \in X} \mu(x) \end{aligned} \tag{3.5}$$

Если существует  $x^*$ , удовлетворяющего условиям данным выше, то можно рассмотреть следующую задачу: найти последовательность  $x_1, x_2, \dots$ , из  $X$  таких, что

$$\lim_{i \rightarrow \infty} \mu(x_i) = \sup_{x \in X} \mu(x) \quad \text{или} \tag{3.6}$$

$$\lim_{i \rightarrow \infty} \mu(x_i) = \inf_{x \in X} \mu(x).$$

$\inf$  и  $\sup$  означают наибольшую нижнюю и наименьшую верхнюю границы, соответственно.

Два нечетких множества  $A$  и  $B$  равны, если их функции принадлежности равны, т.е. если

$$\forall x \in X, \mu_A(x) = \mu_B(x), A = B. \quad (3.7)$$

**Определение 3.2. Носитель множества, переходная точка и синглтон.**

Носитель нечеткого множества состоит из тех элементов  $x$ , которые имеют  $\mu_A(x) > 0$ :

$$\sup pA = \{x \in X, \mu(x) > 0\}. \quad (3.8)$$

Элемент  $x \in X$ , у которого  $\mu_A(x) = \frac{1}{2}$ , называется переходной точкой нечеткого множества  $A$ .

Нечеткое множество, носителем которого является одна точка в  $X$  с  $\mu_A = 1.0$ , называется синглтоном.

**Определение 3.3. Высота нечеткого множества, нормальное нечеткое множество.**

Высотой нечеткого множества  $A$  называют наивысшую границу функции принадлежности:

$$hgt(A) = \sup_{x \in X} \mu_A(x) \quad (3.9)$$

Нечеткое множество  $\Phi$  нормальное, если  $\exists x \in X, \mu_A(x) = 1$ .

Ясно, что высота нормального нечеткого множества  $A$  равна 1, (рис.3.2), т.е.

$$hgt(A)=1.$$

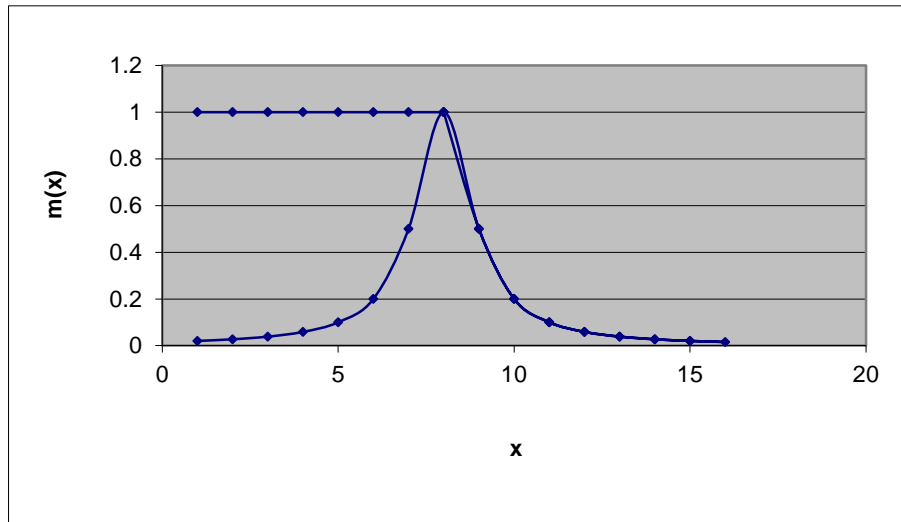


Рис.3.2. Нормальное нечеткое множество

Если  $\sup_{x \in X} \mu_A(x) < 1$ , то нечеткое множество  $A$  называется субнормальным (рис.3.3).

Пустым множеством  $\emptyset$ , называют множество, для которого  $\forall x \in X, \mu_{\emptyset}(x) = 0$ .

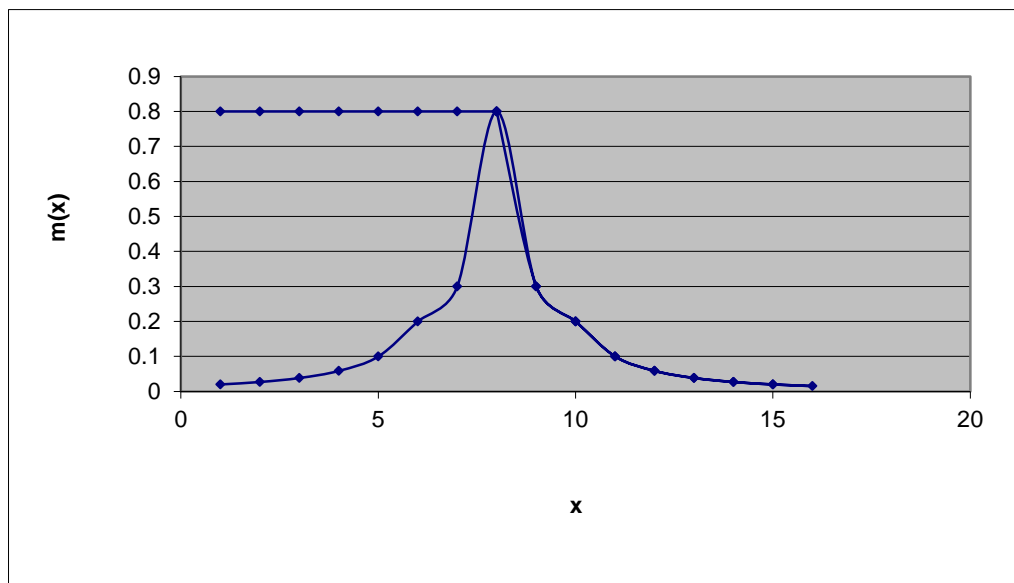


Рис.3.3. Субнормальное нечеткое множество

**Пример.** Дано нечеткое множество (рис.3.4).

$$A = 0.3/20 + 0.5/22 + 1.0/25 + 0.8/27 + 0.4/30$$

Здесь

$X = \{15, 20, 22, 25, 27, 30, 33, 35\}$  – универсум,

$\text{Supp}A = \{20, 22, 25, 27, 30\}$ ,

$A$  – нормальное множество, т.е.  $\exists 25 \in X, \mu_A(25) = 1$ .

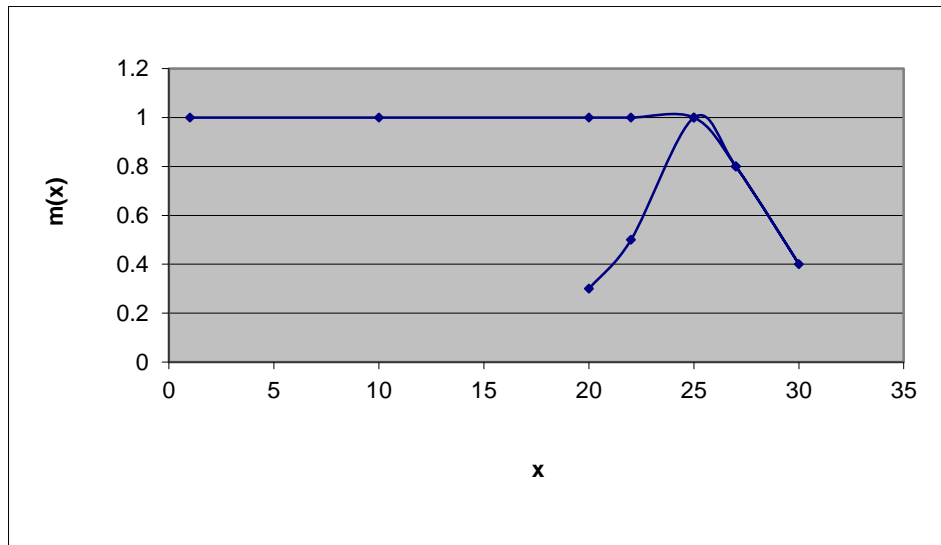


Рис.3.4. Нечеткое множество

### Определение 3.4. $\alpha$ - уровневое нечеткое множество.

Обычное множество элементов, чьи значения принадлежности выше чем некоторый порог  $\alpha \in [0,1]$ , называется  $\alpha$ -сечением  $A^\alpha$  множества  $A$ :

$$A^\alpha = \{x \in X, \mu_A(x) \geq \alpha\}. \quad (3.10)$$

Строгое  $\alpha$ -сечение определяется как

$$A^\alpha = \{x \in X, \mu_A(x) > \alpha\}.$$

Пример. Дано нечеткое множество

$$A = 0.2/5 + 0.4/6 + 0.6/7 + 0.8/8 + 0.9/9 + 1.0/10 + 0.9/11 + 0.8/12 + \\ + 0.6/13 + 0.4/14 + 0.2/15.$$

Если  $\alpha = 0.3$ ,  $\alpha = 0.5$  и  $\alpha = 0.8$ , то  $\alpha$ -уровневые множества, соответственно имеет вид:

$$A^{0.3} = 0.4/6 + 0.6/7 + 0.8/8 + 0.9/9 + 1.0/10 + 0.9/11 + 0.8/12 + 0.6/13 + 0.4/14,$$

$$A^{0.5} = 0.6/7 + 0.8/8 + 0.9/9 + 1.0/10 + 0.9/11 + 0.8/12 + 0.6/13,$$

$$A^{0.8} = 0.9/9 + 1.0/10 + 0.9/11.$$



Графическое представление  $\alpha$ -уровневых множеств дано на рис.3.5.

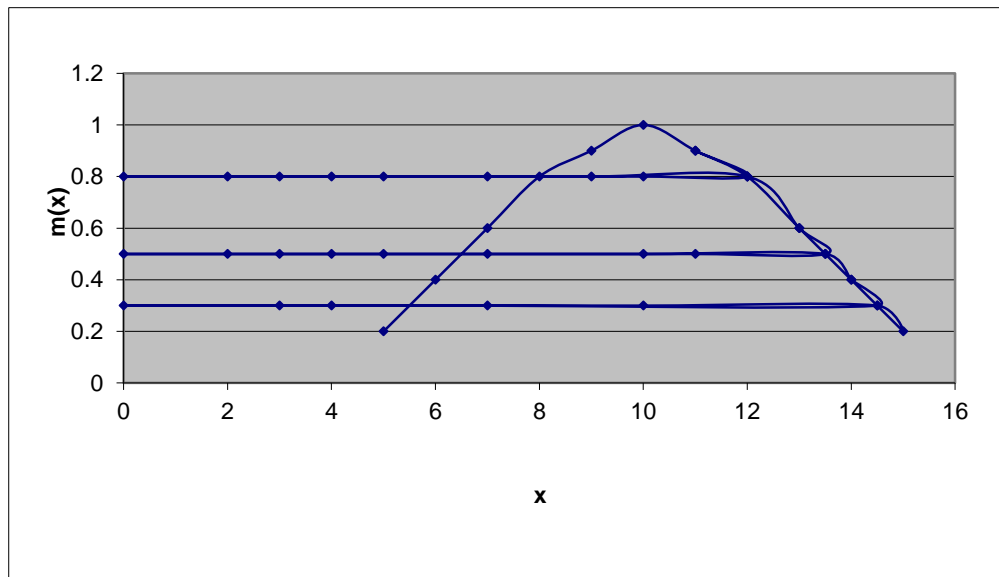


Рис.3.5.  $\alpha$ -уровневые множества нечеткого множества

$\alpha$ -уровневое нечеткое множество  $A^\alpha$  определяется как

$$A^\alpha = \{(x, \mu_{A^\alpha}(x) = \mu_A(x)) / x \in A^\alpha, \alpha \in [0,1]\}, \quad (3.11)$$

где  $\mu_A$  - функция принадлежности нечеткого множества  $A$  и  $\mu_{A^\alpha}$  - функция принадлежности  $\alpha$ -уровневого нечеткого множества  $A^\alpha$ , тогда как  $A^\alpha$  -  $\alpha$ -уровневое множество в форме

$$A^\alpha = \{x / \mu_A(x) \geq \alpha, \quad x \in X\}. \quad (3.12)$$

**Пример.** Пусть задано нечеткое множество  $A$ :

$$A = 0.2/30 + 0.4/35 + 0.6/40 + 0.8/45 + 1/50 + 0.7/55.$$

0.5-уровневым множеством  $A^{\alpha=0.5}$  является множество

$$A^{\alpha=0.5} = 0.6/40 + 0.8/45 + 1/50 + 0.7/55.$$

### Определение 3.5. Мощность нечеткого множества.

Предположим  $X$ -конечное множество и  $A$ -нечеткое множество, определенное на  $X$ . Тогда мощность  $|A|$  нечеткого множества  $A$  определяется как:

$$|A| = \sum_{x \in X} \mu_A(x) . \quad (3.13)$$

В случае, когда  $X$  – бесконечное множество,  $|A|$  не всегда существует. Однако, если  $A$  имеет конечный носитель, тогда мощность нечеткого множества  $A$  определяется как

$$|A| = \sum_{x \in \text{supp } A} \mu_A(x) . \quad (3.14)$$

### Определение 3.6. Включение для нечеткого множества.

Говорят, что  $A$  включено в  $B$  ( $A \subseteq B$ ), если и только если  $\forall x \in X, \mu_A(x) \leq \mu_B(x)$ . Когда неравенство строгое, включение считается строгим и обозначается  $A \subset B$ .

$X$  относится к  $A$  с уровнем  $\alpha$ , если и только если  $x \in A$ ,  $A^\alpha$  слабо включается в  $B$  ( $A - <_\alpha B$ ) всегда, когда все элементы  $X$  относятся с уровнем  $\alpha$  к  $\bar{A}$  или  $B$ : математически,  $A - <_\alpha B$ , если  $x \in (\bar{A} \cup B)^\alpha \forall x \in X$  или

$$\forall x \in X, \max(1 - \mu_A(x), \mu_B(x)) \geq \alpha .$$

Слабое уравнение  $A > - - < B$  определяется следующим образом:

обозначение принадлежности  $\mu_A(x)$  и  $\mu_B(x)$  либо больше или равны  $1/2$ , либо оба меньше или равны  $1/2$ .  $A > - - < B$ , если и только если

$$\forall x \in X, \min[\max(1 - \mu_A(x), \mu_B(x)), \min(1 - \mu_A(x), 1 - \mu_B(x))] \geq 1/2 .$$

Ниже дан ряд законов для операций ( $\cup, \cap$ ) с нечеткими множествами.

#### 1. Нейтральность

$$\min(1, \mu_A(x)) = \mu_A(x) \Rightarrow G \cap A = A ;$$

$$\max(0, \mu_A(x)) = \mu_A(x) \Rightarrow \emptyset \cup A = A .$$

## 2. Коммутативность

$$\min(\mu_A(x), \mu_B(x)) = \min(\mu_B(x), \mu_A(x)) = A \cap B = B \cap A;$$

$$\max(\mu_A(x), \mu_B(x)) = \max(\mu_B(x), \mu_A(x)) = A \cup B = B \cup A.$$

## 3. Ассоциативность

$$\begin{aligned} \min(\min(\mu_A(x), \mu_B(x)), \mu_C(x)) &= \min(\mu_A(x), \min(\mu_B(x), \mu_C(x))) \Rightarrow \\ \Rightarrow (A \cap B) \cap C &= A \cap (B \cap C); \end{aligned}$$

$$\begin{aligned} \max(\max(\mu_A(x), \mu_B(x)), \mu_C(x)) &= \max(\mu_A(x), \max(\mu_B(x), \mu_C(x))) \Rightarrow \\ \Rightarrow (A \cup B) \cup C &= A \cup (B \cup C). \end{aligned}$$

## 4. Монотонность

$$\begin{aligned} \mu_A(x) \leq \mu_C(x) \wedge \mu_B(x) \leq \mu_D(x) \Rightarrow \min(\mu_A(x), \mu_B(x)) \leq \\ \leq \min(\mu_C(x), \mu_D(x)) \Rightarrow A \subset C \wedge B \subset D \Rightarrow A \cap B \subset C \cap D; \end{aligned}$$

$$\begin{aligned} \max(\mu_A(x), \mu_B(x)) \leq \max(\mu_C(x), \mu_D(x)) \Rightarrow A \subset C \wedge B \subset D \Rightarrow \\ \Rightarrow A \cup B \subset C \cup D. \end{aligned}$$

## 5. Идемпотентность

$$\min(\mu_A(x), \mu_A(x)) = \mu_A(x) \Rightarrow A \cap A = A;$$

$$\max(\mu_A(x), \mu_A(x)) = \mu_A(x) \Rightarrow A \cup A = A.$$

## 6. Дистрибутивность

$$\begin{aligned} \min(\mu_A(x), \max(\mu_B(x), \mu_C(x))) &= \max(\min(\mu_A(x), \mu_B(x))), \\ \min(\mu_A(x), \mu_C(x)) \Rightarrow A \cap (B \cup C) &= (A \cap B) \cup (A \cap C); \end{aligned}$$

$$\begin{aligned} \max(\mu_A(x), \min(\mu_B(x), \mu_C(x))) &= \min(\max(\mu_A(x), \mu_B(x))), \\ \max(\mu_A(x), \mu_C(x)) \Rightarrow A \cup (B \cap C) &= (A \cup B) \cap (A \cup C). \end{aligned}$$

## 7. Поглощение

$$\min(\mu_A(x), \max(\mu_A(x), \mu_B(x))) = \mu_A(x) \Rightarrow A \cap (A \cup B) = A;$$

$$\max(\mu_A(x), \min(\mu_A(x), \mu_B(x))) = \mu_A(x) \Rightarrow A \cup (A \cap B) = A.$$

## 8. Закон Де Моргана

$$1 - \min(\mu_A(x), \mu_B(x)) = \max(1 - \mu_A(x), 1 - \mu_B(x)) \Rightarrow \overline{A \cap B} = \bar{A} \cup \bar{B};$$

$$1 - \max(\mu_A(x), \mu_B(x)) = \min(1 - \mu_A(x), 1 - \mu_B(x)) \Rightarrow \overline{A \cup B} = \bar{A} \cap \bar{B};$$

## 9. Двойное отрицание

$$1 - (1 - \mu_A(x)) = \mu_A(x) \Rightarrow \overline{\bar{A}} = A.$$

## 10. Отрицание основного и пустого множества

$$1 - 1 = 0 \Rightarrow \bar{G} = \emptyset;$$

$$1 - 0 = 1 \Rightarrow \overline{\emptyset} = G.$$

**Определение 3.7. Картезианское произведение.** Если  $A_1, \dots, A_n$  являются нечеткими множествами в  $U_1, \dots, U_n$ , соответственно, картезианским произведением  $A_1, \dots, A_n$  является нечеткое множество в пространстве  $U_1 \times U_2 \times \dots \times U_n$  с функцией принадлежности

$$\mu_{A_1 \times \dots \times A_n}(u_1, u_2, \dots, u_n) = \min\{\mu_{A_1}(u_1), \dots, \mu_{A_n}(u_n)\} \quad (3.15)$$

или

$$\mu_{A_1 \times \dots \times A_n}(u_1, u_2, \dots, u_n) = \mu_{A_1}(u_1) \cdot \mu_{A_2}(u_2) \cdot \dots \cdot \mu_{A_n}(u_n). \quad (3.16)$$

## Определение 3.8. Нечеткое разбиение.

Если  $A$  обычное подмножество  $X$ , тогда пара  $(A, \bar{A})$  является разбиением множества  $X$ , при условии, что  $A \neq \emptyset, A \neq X$ . Если  $A$  является нечетким множеством ( $A \neq \emptyset, A \neq X$ ), тогда пара  $(A, \bar{A})$  называется нечетким разбиением. Если система нечетких множеств

$A_1, \dots, A_m$  ( $A_i \neq \emptyset, A_i \neq X_i, i = \overline{1, m}$ ) такая, что

$$\forall x \in X, \sum_{i=1}^m \mu_{A_i}(x) = 1,$$

то эта система называется, нечетким разбиением множества  $X$ .

Характеристикой нечеткого множества выступает функция

*принадлежности* (Membership Function). Обозначим через  $MF_C : X \rightarrow [0,1]$ ,  $MF_C(x)$  — степень принадлежности  $x \in X$  к нечеткому множеству  $C$ , представляющей собой обобщение понятия характеристической функции обычного множества. Здесь  $X$  — *область рассуждений*. Тогда нечеткое множеством  $C$  может быть задано вектором упорядоченных пар:

$$C = \{MF_C(x) \setminus x : x \in X, MF_C(x) \in [0,1]\}.$$

Значение  $MF_C(x) = 0$  означает отсутствие принадлежности  $x$  к множеству  $X$ ,  $MF_C(x) = 1$  — полную принадлежность. *Носителем* нечеткого множества  $A$  с областью рассуждений  $X$  является множество:  $\{x \in X : MF_A(x) = 1\}$  т. е. те элементы  $X$ , которые полностью принадлежат  $A$ .

Проиллюстрируем это на примере неточного определения «горячий чай». В качестве  $X$  (области рассуждений) возьмем шкалу температуры в градусах Цельсия, она будет изменяться от 0 до 100 градусов. Нечеткое множество для понятия «горячий чай» может выглядеть следующим образом:

$$C = \{0/0, 0/10, 0/20, 0.1/30, 0.3/40, 0.6/50, 0.8/60, 0.9/70, 1/80, 1/90, 1/100\}.$$

Так, чай с температурой  $60^\circ\text{C}$  принадлежит к множеству «горячий» со степенью принадлежности 0.8. Для одного человека чай при температуре  $60^\circ\text{C}$  может оказаться горячим, для другого — не слишком горячим. Именно в этом и проявляется нечеткость задания соответствующего множества.

Для нечетких множеств, как и для обычных, определены основные логические операции. Самыми основными, необходимыми для расчетов, являются пересечение  $\cap$  и  $\cup$  объединение. Пересечение двух нечетких множеств  $A \cap B$  :  $MF_{A \cap B}(x) = \min(MF_A(x), MF_B(x))$ . Объединение двух нечетких множеств  $A \cup B$  :  $MF_{A \cup B}(x) = \max(MF_A(x), MF_B(x))$ . В теории нечетких множеств разработан общий подход к выполнению операторов пересечения, объединения и дополнения, реализованный в так называемых *треугольных нормах* и *конормах* [3]. Приведенные выше реализации операций пересечения и объединения — наиболее распространенные случаи.

### ***Гранулирование информации***

Ранее говорилось, что нечеткие вычисления оперируют с *гранулированной информацией* (Information Granulation), вычисления производятся со словами, а не с числами. Гранулирование информации — есть процесс объединения схожих точек или объектов в одну группу. Тогда нечеткость подобных групп есть прямое следствие нечеткости понятия сходства. Простыми примерами таких групп являются понятия «средний возраст», «деловая часть города» и др. Отметим, что стремление объединять схожие по свойствам объекты в одну группу характерно для человеческих рассуждений. Понятие гранулированной информации находят свое отражение в теории нечетких вычислений в терминах *нечеткой* и *лингвистической переменной* [4].

### ***Нечеткие и лингвистические переменные***

Целью введения нечеткого множества чаще всего является формализация нечетких понятий и отношений естественного языка. Данную формализацию можно выполнить, воспользовавшись понятиями *нечеткой* и *лингвистической переменной*. Нечеткая переменная описывается набором  $\langle X, N, A \rangle$ , где  $N$  — название переменной,  $X$  — *универсальное множество* (синоним область рассуждений),  $A$  — нечеткое множество на  $X$ . Понятие лингвистической переменной строится на базе нечеткой переменной, т.е. лингвистическая переменная находится на более высоком уровне [4].

Формально: лингвистическая переменная задается пятеркой  $\langle N, T, X, G, M \rangle$ , где  $N$  — имя переменной;  $T$  — *терм-множество*, каждый элемент которого (*терм*) представляется, как нечеткое множество на универсальном множестве  $X$ ;  $G$  — синтаксические правила, часто в виде формальной грамматики, порождающие название термов;  $M$  — семантические правила, задающие функции принадлежности нечетких термов, порожденных синтаксическими правилами  $G$ . Таким образом  $T$  — множество нечетких переменных (возможно бесконечное) имена переменных из  $T$  порождаются грамматикой  $G$ , а функции принадлежности — семантическими правилами  $M$ .

Существует большое разнообразие типовых форм кривых для

задания функций принадлежности. Наибольшее распространение получили: *треугольная, трапецидальная и гауссова* функции принадлежности [5].

Рассмотрим такое нечеткое понятие как «температура чая». Это и есть название лингвистической переменной. Сформируем для нее базовое терм-множество, которое будет состоять из трех нечетких переменных:

$$T_{(base)} = \{\text{«холодный»}, \text{«теплый»}, \text{«горячий»}\}$$

и зададим область рассуждений в виде  $X = [0, 100]$  (градусы цельсия). В предыдущем примере рассматривался дискретное множество  $X$ , а функция принадлежности задавалась набором: ключ, значение. Здесь универсальное множество непрерывно, а функция принадлежности действует из  $[0, 100]$  в  $[0, 1]$ .

В качестве грамматики  $G$  разрешим названиям лингвистических переменных иметь *модификаторы*: «очень», «не», «слегка». Модификаторы можно повторять несколько раз и комбинировать, например «слегка не очень очень горячий» чай. А также разрешим термы (возможно с модификаторами) соединять связками «и», «или», например: «теплый или слегка горячий» чай. Таким образом  $T$  будет состоять из базовых термов и всего, что можно породить из  $T_{(base)}$  при помощи указанной грамматики  $G$ .

В качестве функции принадлежности для каждого лингвистического терма из базового терм-множества  $T$  возьмем треугольную кривую, определяемую формулой:

$$MF(t) = \begin{cases} 1 - \frac{b-t}{b-a}, & a \leq t \leq b, \\ 1 - \frac{t-b}{c-b}, & b \leq t \leq c, \\ 0, & \text{иначе,} \end{cases}$$

$a, b, c$  — параметры кривой, для терма «холодный» возьмем: ( $a = 0, b = 0, c = 40$ ) «теплый»: ( $a = 30, b = 60, c = 80$ ), «горячий»: ( $a = 70, b = 100, c = 100$ ).

Наконец зададим семантику  $M$ :

$$MF_{\text{очень LV}}(x) = MF_{LV}(x)^2,$$

$$MF_{\text{слегка LV}}(x) = MF_{LV}(x)^{1/2},$$

$$MF_{\text{не LV}}(x) = 1 - MF_{LV}(x),$$

$$MF_{A \text{ и } B}(x) = \min( MF_A(x), MF_B(x) ),$$

$$MF_{A \text{ или } B}(x) = \max( MF_A(x), MF_B(x) ).$$

Здесь модификаторы «очень», «слегка» используются с семантикой ослабления / усиления термина к которому они относятся. Соответствующие им операции возведения в квадрат / извлечения корня называются операциями *концентрации/растяжения* соответственно.

Более подробную информацию можно найти в [2, 3, 5].

### ***Нечеткая логика***

В данном разделе речь пойдет о месте нечеткой логики в мягких вычислениях. Термин нечеткая логика используется в двух различных смыслах. В узком смысле, *нечеткая логика* — это логическая система, являющаяся расширением многозначной логики. Однако, даже для нечеткой логики в узком смысле, список основных операций очень отличается как по духу, так и по содержанию от списка основных операций для систем многозначных логик.

В широком смысле, нечеткая логика равнозначна теории нечетких множеств, т.е. классов с неточными, размытыми границами [6]. Таким образом, нечеткая логика, понимаемая в узком смысле, является разделом нечеткой логики в широком смысле.

Важной характеристикой нечеткой логики является то, что любая теория *Th* может быть *фаззифицирована* (Fuzzified) и, следовательно, обобщена путем замены понятия четкого множества в *Th* понятием нечеткого множества. Таким способом можно прийти к *нечеткой арифметике, нечеткой топологии, нечеткой теории вероятностей* и т. д. Выигрышем от фаззификации является большая общность и лучшее соответствие модели действительности. Однако с нечеткими числами труднее оперировать. Более того, значения большинства нечетких понятий зависят от контекста и/или приложения. Это та цена, которую необходимо заплатить за лучшее согласие с реальностью. Более подробную информацию можно найти в оригинальной статье [1].

Особое место в нечеткой логике занимает лингвистическая переменная «истинность». В классической логике истинность может принимать только два значения: истинно и ложно. В



нечеткой логике истинность размытая. Нечеткая истинность определяется аксиоматически, причем разные авторы делают это по-разному. Интервал  $[0, 1]$  используется как универсальное множество для задания лингвистической переменной «истинность». Обычная, четкая истинность может быть представлена нечеткими множествами-синглтонами. В этом случае четкому понятию истинно будет соответствовать функция принадлежности

$$MF_{\text{истинно}}(u) = \begin{cases} 0, & u \neq 1 \\ 1, & u = 1 \end{cases}, \text{ а четкому понятию ложно } MF_{\text{ложно}}(u) = \begin{cases} 0, & u \neq 0 \\ 1, & u = 0 \end{cases},$$

$$u \in [0,1].$$

Для задания нечеткой истинности Заде предложил такие функции принадлежности термов «истинно» и «ложно»:

$$MF_{\text{истинно}}(u) = \begin{cases} 0, & 0 \leq u \leq a, \\ 2 \cdot \left( \frac{u-a}{1-a} \right)^2, & a < u < \frac{a+1}{2}, \\ 1 - 2 \cdot \left( \frac{u-1}{1-a} \right)^2, & \frac{a+1}{2} < u \leq 1, \end{cases},$$

$$MF_{\text{ложно}}(u) = MF_{\text{истинно}}(1-u) \text{ для } \forall u \in [0,1].$$

Здесь  $a \in [0,1]$  — параметр, определяющий носители нечетких множеств «истинно» и «ложно». Для нечеткого множества «истинно» носителем будет полуоткрытый интервал  $(a,1]$ , а для нечеткого множества «ложно» —  $[0,a)$ . Функции принадлежности нечетких термов «истинно» и «ложно» изображены на рис. 1. Они построены при значении параметра  $a = 0.4$ . Как видно, графики функций принадлежности термов «истинно» и «ложно» представляют собой зеркальные отображения [7].

Вначале кратко напомнить основные положения обычной (булевой) логики. Рассмотрим два утверждения  $A$  и  $B$ , каждое из которых может быть истинным или ложным, т.е. принимать значения 1 или 0. Для этих двух утверждений всего существует  $2^2 = 4$  различных логических операций, из которых содержательно интерпретируются лишь пять:  $AND$ ,  $OR$ ,  $XOR$ ,  $\rightarrow$  (импликация),  $=$  (эквивалентность).

Предположим, что логическое утверждение может принимать не два значения истинности, а три, например: «истинно», «ложно» и «неопределенно». В этом случае мы будем иметь дело не с двухзначной, а трехзначной логикой. Общее количество бинарных операций, а, следовательно, и таблиц истинности, в трехзначной логике равно  $3^9$ .

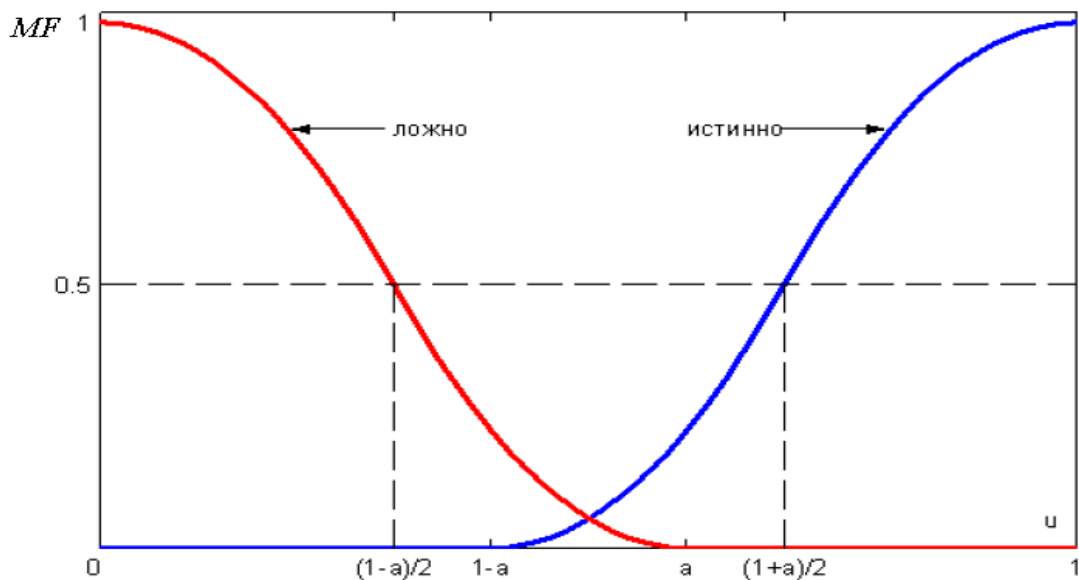


Рис. 3.6. лингвистическая переменная « истинность » по Заде

### *Нечеткие логические операции*

Нечеткая логика является разновидностью многозначной логики, в которой значения истинности задаются лингвистическими переменными или термами лингвистической переменной «истинность». Правила выполнения нечетких логических операций получают из булевых логических операций с помощью принципа обобщения.

Обозначим нечеткие логические переменные через  $A$  и  $B$ , а функции принадлежности, задающие истинностные значения этих переменных через  $MF_A(u)$  и  $MF_B(u)$ ,  $u \in [0 .. 1]$ . *Нечеткие логические операции AND, OR, XOR,  $\rightarrow$*  (импликация) выполняются по таким правилам:

$$MF_{A \text{ AND } B}(u) = \min(MF_A(u), MF_B(u)),$$

$$MF_{A \text{ OR } B}(u) = \max(MF_A(u), MF_B(u)),$$

$$MF_{A \text{ XOR } B}(u) = \max(MF_A(u), MF_B(u)) - \min(MF_A(u), MF_B(u)),$$

$$MF_{A \rightarrow B}(u) = \max(1 - MF_A(u), MF_B(u)).$$

В многозначной логике логические операции могут быть заданы таблицами истинности. В нечеткой логике количество возможных значений истинности может быть бесконечным, следовательно, в общем виде табличное представление логических операций невозможно. Однако в табличной форме можно представить нечеткие логические операции для ограниченного количества истинностных значений, например, для терм-множества {«истинно», «очень истинно», «не истинно», «более-менее ложно», «ложно»} [5].

### 3.2. Операции над нечеткими множествами

Пусть  $A$  и  $B$  – два нечетких множества в  $X$  с функциями принадлежности  $\mu_A$  и  $\mu_B$ , соответственно. Теоретико-множественные операции объединения, пересечения и дополнения для нечетких множеств определяются через их функции принадлежности следующим образом.

#### Объединение.

Объединение двух нечетких множеств  $A$  и  $B$  определяются следующим образом:

$$\forall x \in X, \mu_{A \cup B}(x) = \max\{\mu_A(x), \mu_B(x)\}, \quad (3.17)$$

где  $\mu_{A \cup B}$  - функция принадлежности для  $A$  и  $B$ .

**Пересечение.** Функция принадлежности  $\mu_{A \cap B}$ , определяется как

$$\forall x \in X, \mu_{A \cap B}(x) = \min\{\mu_A(x), \mu_B(x)\}. \quad (3.18)$$

#### Пример.

$$A = 0.07/2 + 0.2/3 + 0.4/4 + 0.63/5 + 0.87/6 + 1.0/7 + 0.89/8 + 0.5/9 + 0.2/10 + 0.07/11,$$

$$B = 0.05/6 + 0.11/7 + 0.21/8 + 0.32/9 + 0.46/10 + 0.69/11 + 0.87/12 + 1.0/13 + 0.9/14 + 0.5/15 + 0.25/16 + 0.09/18,$$

$$A \cup B = 0.07/2 + 0.2/3 + 0.4/4 + 0.63/5 + 0.87/6 + 1.0/7 + 0.89/8 + 0.5/9 + 0.46/10 + 0.69/11 + 0.87/12 + 1.0/13 + 0.9/14 + 0.5/15 + 0.25/16 + 0.09/18$$

(см. рис. 3.6).

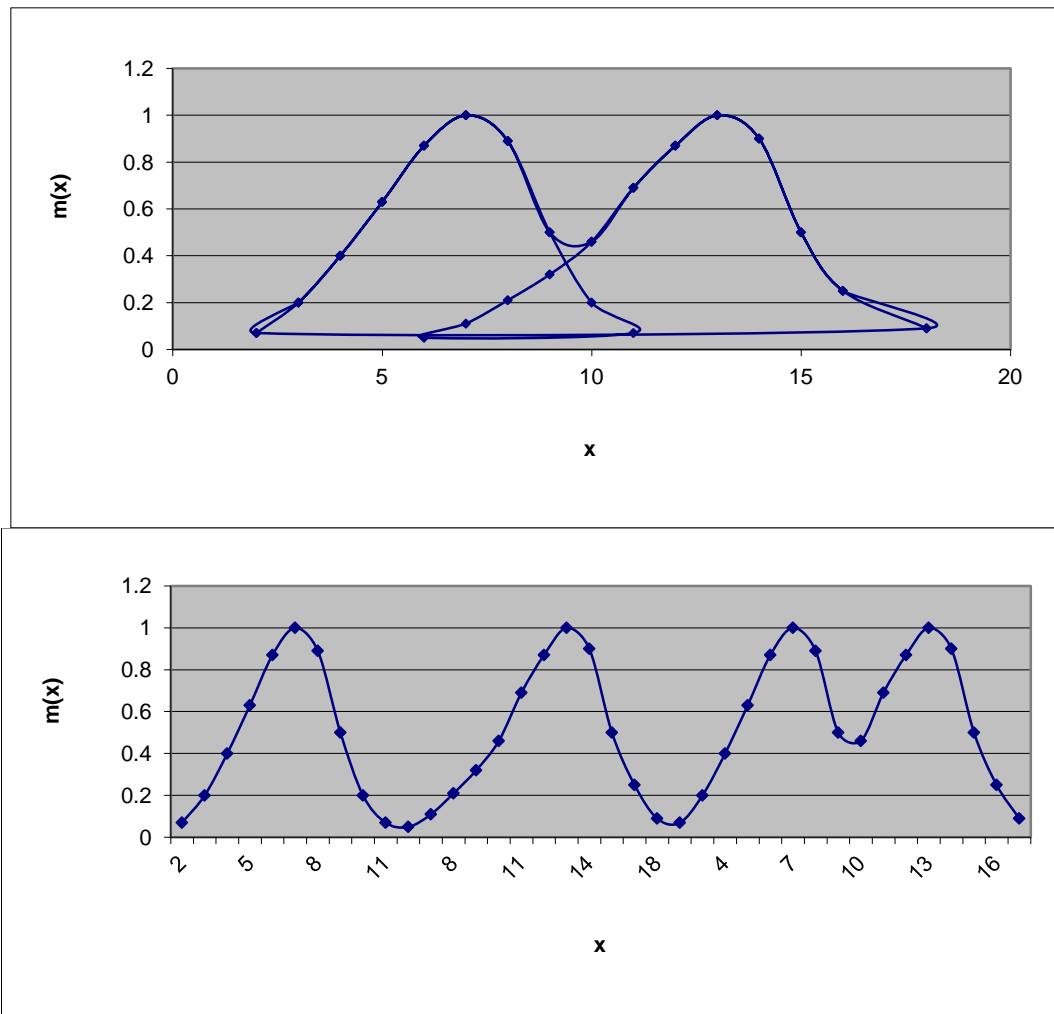


Рис.3.6.Объединение нечетких множеств

$$A = 0.15/2 + 0.41/3 + 0.66/4 + 0.85/5 + 0.97/6 + 1/7 + 0.9/8 + 0.6/9 + 0.42/10 + 0.3/11 + 0.18/12 + 0.1/13 + 0.03/14,$$

$$B = 0.05/5 + 0.1/6 + 0.16/7 + 0.25/8 + 0.35/9 + 0.47/10 + 0.62/11 + 0.8/12 + 0.94/13 + 1/14 + 0.97/15 + 0.83/16 + 0.5/17 + 0.2/18 + 0.07/19,$$

$$A \cup B = 0.05/5 + 0.1/6 + 0.16/7 + 0.25/8 + 0.35/9 + 0.42/10 + 0.3/11 + 0.18/12 + 0.1/13 + 0.03/14,$$

(см. рис. 3.7).

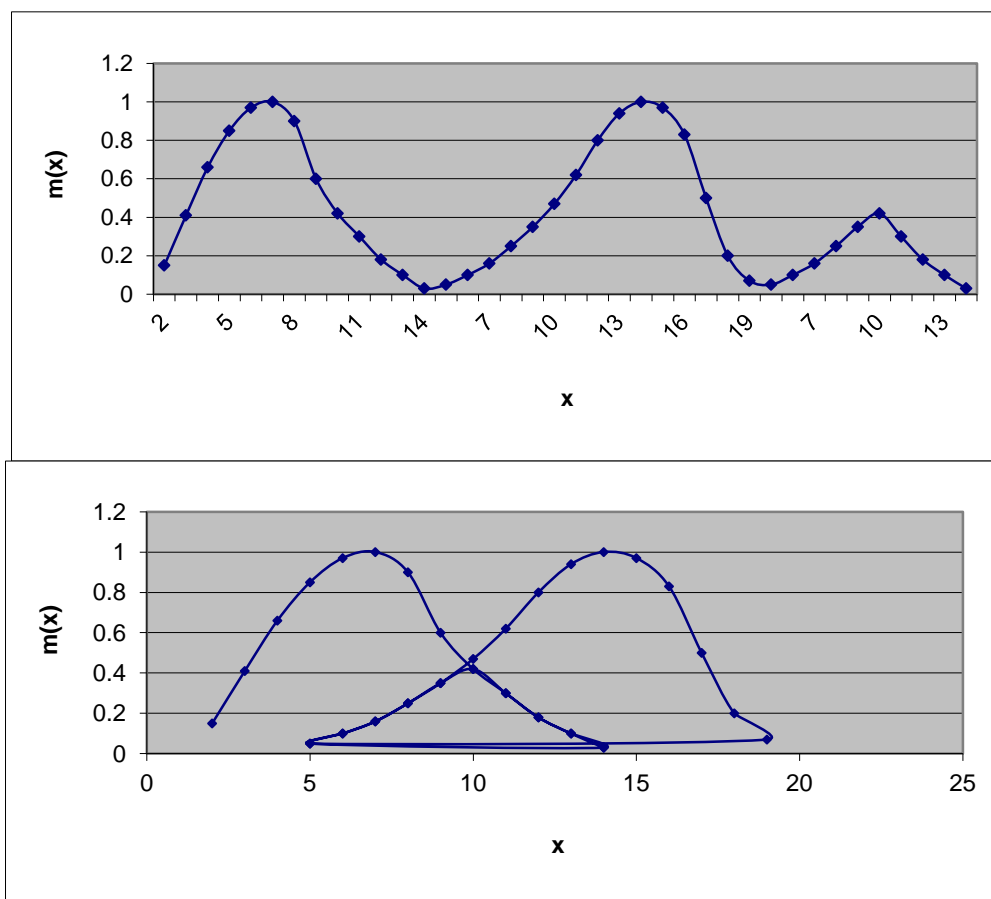


Рис.3.7. Пересечение нечетких множеств

**Определение 3.11. Дополнение.**

Дополнение  $\bar{A}$  множества  $A$  определяется следующим образом:

$$\forall x \in X, \mu_{\bar{A}}(x) = 1 - \mu_A(x). \quad (3.19)$$

Пример.

$$A = 0/1 + 0.05/2 + 0.14/3 + 0.27/4 + 0.5/5 + 0.76/6 + 0.93/7 + 1.0/8 + 0.96/9 + 0.84/10 + 0.62/11 + 0.37/12 + 0.25/13 + 0.16/14 + 0.09/15 + 0.03/16 + 0/17.$$

$$\bar{A} = 1.0/1 + 0.95/2 + 0.86/3 + 0.73/4 + 0.5/5 + 0.24/6 + 0.07/7 + 0/8 + 0.04/9 + 0.16/10 + 0.38/11 + 0.63/12 + 0.75/13 + 0.84/14 + 0.91/15 + 0.97/16 + 1.0/17.$$

(см. рис. 3.8).

Для объединения и пересечения нечетких множеств можно использовать и другие операторы.

**Алгебраическое произведение.**

$$\forall x \in X, \mu_{A \cdot B}(x) = \mu_A(x) \cdot \mu_B(x). \quad (3.20)$$

**Ограниченное произведение.**

$$\forall x \in X, \mu_{A \otimes B}(x) = \max\{0, \mu_A(x) + \mu_B(x) - 1\}. \quad (3.21)$$

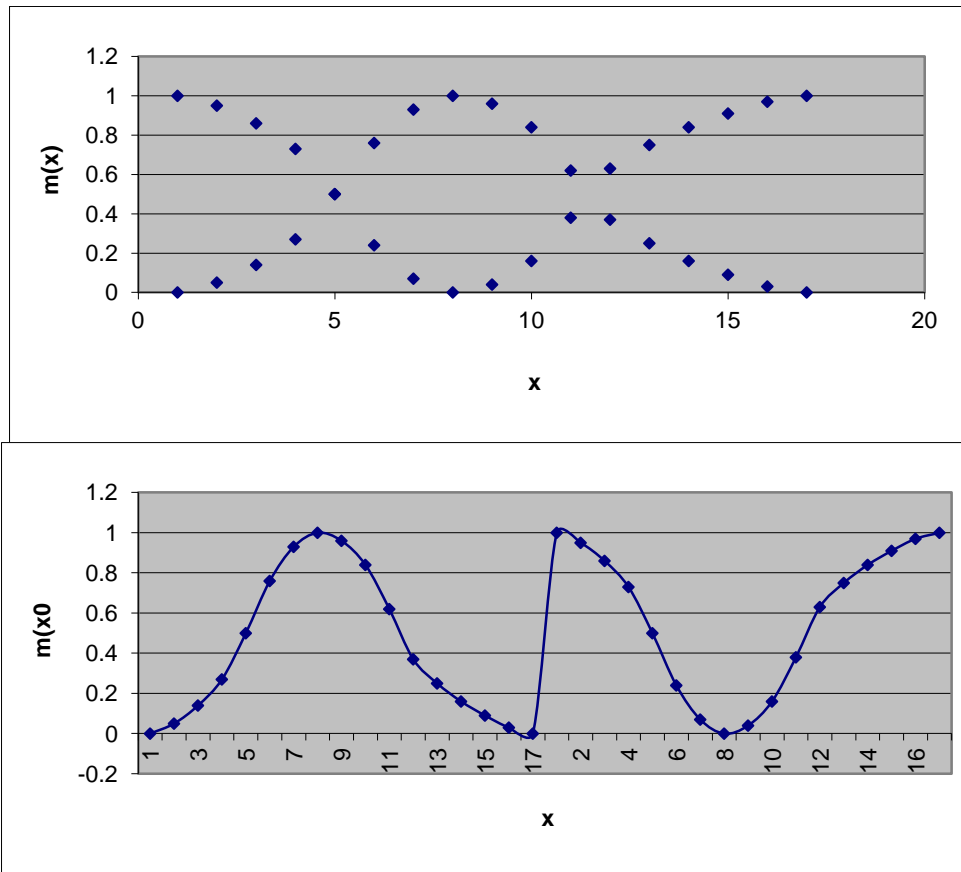


Рис.3.8. Дополнение нечеткого множества

**Строгое (drastic) произведение.**

$$\mu_{A \cap_x B}(x) = \begin{cases} \min\{\mu_A(x), \mu_B(x)\}, & \text{если } \mu_A(x) = 1 \text{ или } \mu_B(x) = 1 \\ 0, & \text{в остальных случаях} \end{cases} \quad (3.22)$$

**Алгебраическая сумма.**

$$\forall x \in X, \mu_{A+B}(x) = \mu_A(x) + \mu_B(x) - \mu_A(x) \cdot \mu_B(x). \quad (3.23)$$

**Ограниченная сумма.**

$$\forall x \in X, \mu_{A \cup B}(x) = \min\{1, \mu_A(x) + \mu_B(x)\}. \quad (3.24)$$

**Строгая (drastic) сумма.**

$$(3.25)$$

**Пример. Алгебраическое произведение нечетких множеств А и В.**

$$\mu_{A \cup B}(x) = \{\max\{\mu_A(x), \mu_B(x)\}, \text{ если } \mu_A(x) = 0 \text{ или } \mu_B(x) = 0, 0, \text{ в остальных случаях.}$$

$$A = 0.1/1 + 0.24/2 + 0.4/3 + 0.63/4 + 0.82/5 + 0.94/6 + 1.0/7 + 0.98/8 + 0.91/9 + 0.76/10 + 0.57/11 + 0.35/12 + 0.2/13 + 0.1/14 + 0.04/15,$$

$$B = 0.02/4 + 0.09/5 + 0.2/6 + 0.32/7 + 0.46/8 + 0.61/9 + 0.76/10 + 0.88/11 + 0.96/12 + 1.0/13 + 0.96/14 + 0.85/15 + 0.62/16 + 0.37/17 + 0.2/18 + 0.09/19$$

$$A * B = 0/3 + 0.01/4 + 0.07/5 + 0.19/6 + 0.32/7 + 0.45/8 + 0.55/9 + 0.58/10 + 0.5/11 + 0.34/12 + 0.2/13 + 0.96/14 + 0.03/15 + 0/16.$$

(см. рис. 3.9).

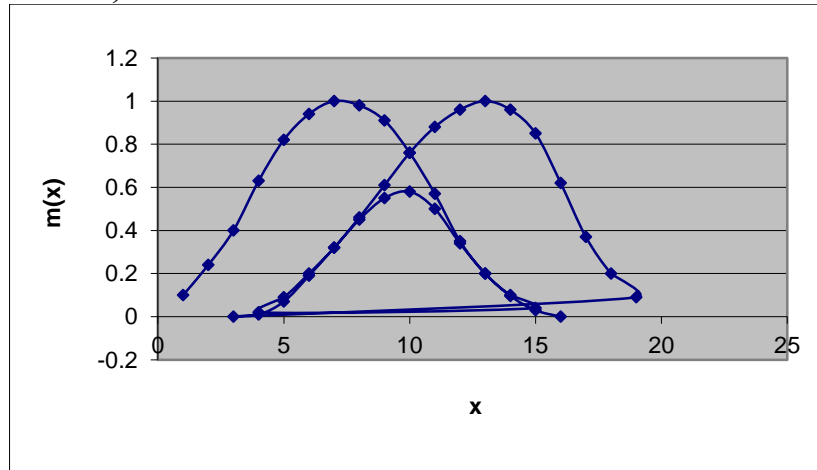


Рис.3.9. Алгебраическое произведение нечетких множеств

**Алгебраическая сумма нечетких множеств А и В.**

$$A = 0.03/1 + 0.1/2 + 0.28/3 + 0.52/4 + 0.75/5 + 0.94/6 + 1/7 + 0.96/8 + 0.87/9 + 0.71/10 + 0.55/11 + 0.4/12 + 0.28/13 + 0.19/14 + 0.12/15 + 0.06/16 + 0.02/17,$$

$$B = 0/1 + 0/2 + 0/3 + 0.02/4 + 0.06/5 + 0.12/6 + 0.17/7 + 0.25/8 + 0.35/9 + 0.5/10 + 0.68/11 + 0.82/12 + 0.95/13 + 1/14 + 0.95/15 + 0.62/16 + 0.35/17 + 0.17/18 + 0.06/19,$$

$$A \hat{+} B = 0.03/1 + 0.1/2 + 0.28/3 + 0.52/4 + 0.75/5 + 0.94/6 + 1.0/7 + 0.96/8 + 0.91/9 + 0.86/10 + 0.86/11 + 0.88/12 + 0.96/13 + 1.0/14 + 0.95/15 + 0.62/16 + 0.35/17 + 0.17/18 + 0.06/19.$$

(см. рис. 3.10).

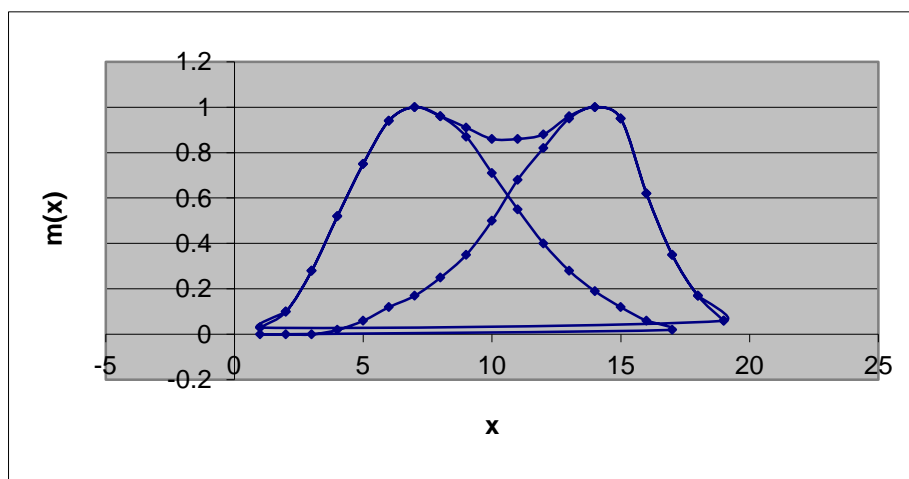


Рис. 3.10. Алгебраическая сумма нечетких множеств

### Ограниченная сумма нечетких множеств A и B.

$A=0.06/1+0.17/2+0.31/3+0.5/4+0.67/5+0.82/6+0.93/7+1.0/8+$   
 $+0.98/9+0.89/10+0.75/11+0.6/12+0.45/13+0.33/14+0.23/15+0.14/16+$   
 $+0.08/17+0.03/18,$   
 $B=0.03/4+0.08/5+0.15/6+0.26/7+0.4/8+0.55/9+0.7/10+0.85/11+$   
 $+0.95/12+1.0/13+0.96/14+0.85/15+0.6/16+0.33/17+0.18/18+0.09/19,$   
 $A \dot{\cup} B=0.06/1+0.17/2+0.31/3+0.53/4+0.75/5+0.97/6+1.0/7+1.0/8+1.0/9+$   
 $+1.0/10+1.0/11+1.0/12+1.0/13+1.0/14+1.0/15+0.64/16+0.41/17+$   
 $+0.21/18+0.09/19.$   
 (см. рис. 3.11).

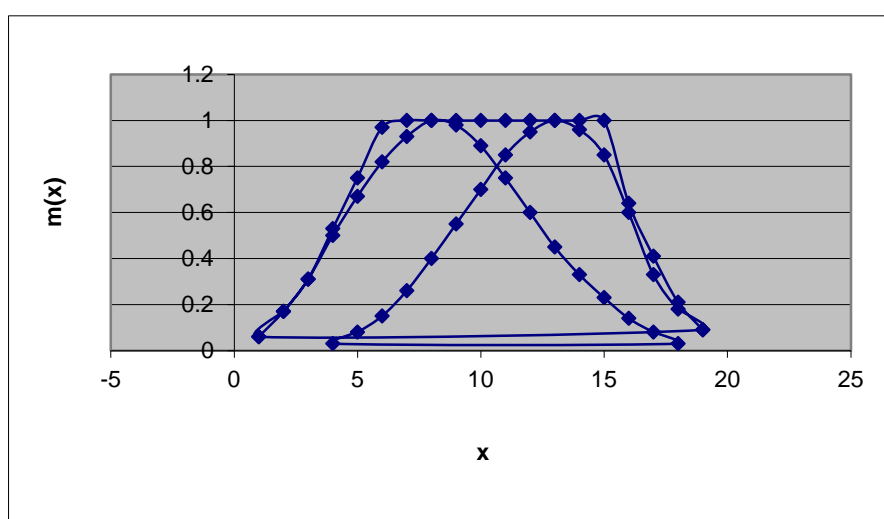


Рис.3.11. Ограниченная сумма нечетких множеств A и B.



## Ограниченное произведение нечетких множеств А и В.

$$A=0.03/1+0.15/2+0.5/3+0.77/4+0.93/5+1.0/6+0.96/7+0.85/8+0.71/9+0.55/10+0.4/11+0.27/12+0.18/13+0.11/14+0.05/15+0.01/16,$$

$$B=0.04/5+0.1/6+0.17/7+0.28/8+0.4/9+0.55/10+0.71/11+0.89/12+0.98/13+1.0/14+0.93/15+0.65/16+0.2/17+0.06/18+0.01/19,$$

$$A \cap B = 0/1+0/2+0/3+0/4+0/5+0.1/6+0.13/7+0.13/8+0.11/9+0.1/10+0.11/11+0.16/12+0.16/13+0.11/14+0/15+0/16+0/17+0/18+0/19.$$

(см. рис. 3.12)

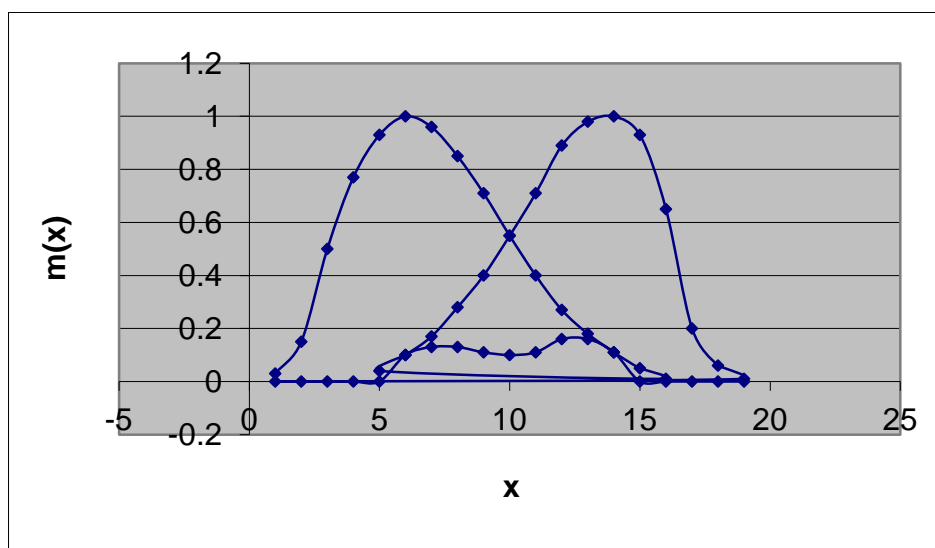


Рис.3.12. Ограниченное произведение нечетких множеств А и В.

### Определение 3.12. Ограниченная и симметрическая разности.

Ограниченная разность  $|_B$  нечетких множеств определяется формулой

$$\forall x \in X, \mu_{A|_B}(x) = \max(0, \mu_A(x) - \mu_B(x)). \quad (3.26)$$

$A|_B$  есть нечеткое множество элементы которого принадлежат А больше чем В.

Симметрическая разность нечетких множеств – это нечеткое множество  $A \nabla B$  элементов, которые больше принадлежат А нежели В:

$$\forall x \in X, \mu_{A \nabla B}(x) = |\mu_A(x) - \mu_B(x)|. \quad (3.27)$$

**Примеры ограниченной и симметрической разности нечетких множеств А и В:**

$$A=0.08/1+0.23/2+0.45/3+0.7/4+0.86/5+0.96/6+1/0/7+0.98/8+ \\ +0.92/9+0.82/10+0.67/11+0.47/12+0.3/13+0.13/14,$$

$$B=0.03/6+0.08/7+0.18/8+0.34/9+0.55/10+0.7/11+0.84/12+0.94/13+ \\ +0.99/14+1.0/15+0.96/16+0.82/17+0.6/18+0.2/19,$$

$$A|-B=0.08/1+0.23/2+0.45/3+0.7/4+0.86/5+0.93/6+0.92/7+0.8/8+ \\ +0.58/9+0.27/10+0/11,$$

$$A \nabla B =0.08/1+0.23/2+0.45/3+0.7/4+0.86/5+0.96/6+1.0/7+0.98/8+ \\ +0.92/9+0.82/10+0.03/11+0.36/12+0.65/13+0.86/14+1.0/15+ \\ 0.96/16+0.82/17+0.6/18+0.2/9.$$

(см. рис. 3.13.а и 3.13.б).

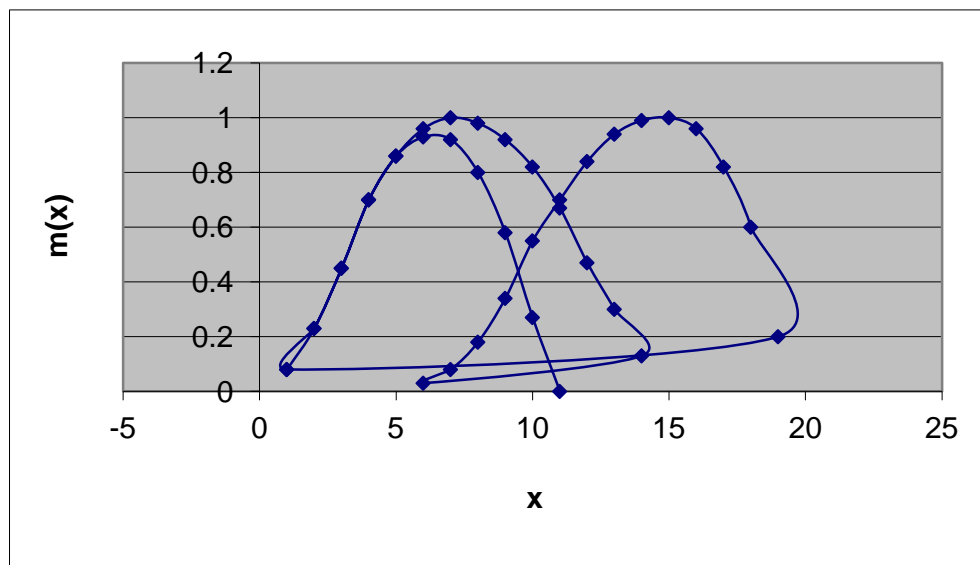


Рис.3.13.а. Ограниченная разность

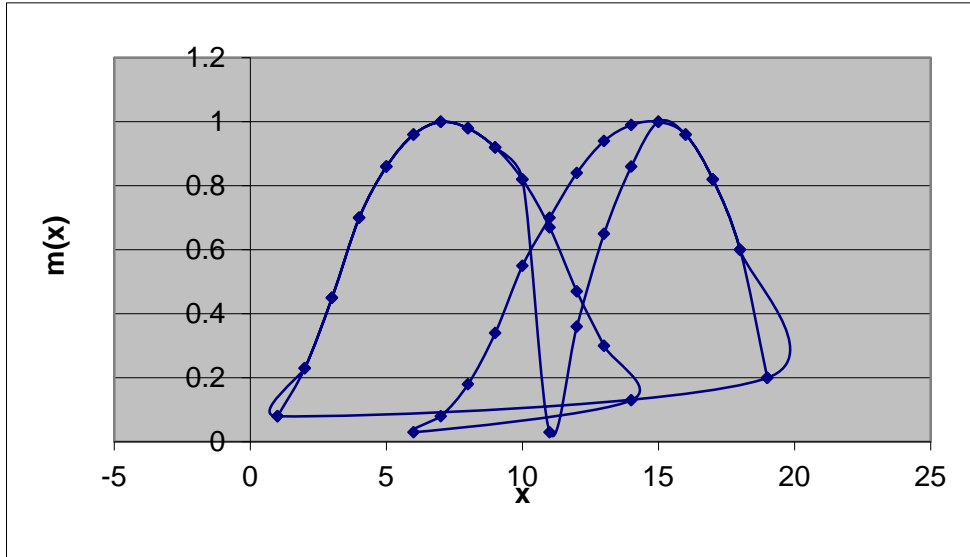


Рис.3.13.в. Симметрическая разность

**Определение 3.13.**  $m$ -я степень нечеткого множества  $A$  равна:

$$\mu_{A^m}(x) = [\mu_A(x)]^m, \quad \forall x \in X, \forall m \in R^+, \quad (3.28)$$

где  $R^+$  - положительно определенное множество действительных чисел.

**Определение 3.14.** Концентрация, растяжение нечетких множеств. Пусть  $A$  является нечетким множеством на универсуме:

$$A = \{x : \mu_A(x) \mid x \in X\}.$$

Тогда, возникающие посредством возведения в степень с помощью оператора концентрирования  $Con_m$  нечеткие множества

$$Con_m A = \{x : (\mu_A(x))^m \mid x \in X\}$$

называются концентрациями  $A$ , а посредством извлечения корня с помощью оператора расширения  $dil_n A = \{x : \sqrt[n]{\mu_A(x)} \mid x \in X\}$  - называются расширениями  $A$ .

**Следствие.** Так как выражение  $[\mu_A(x)]^n \leq \mu_A(x) \leq \sqrt[n]{\mu_A(x)}$  действительно для всех  $x \in X$  и  $n > 1$ , то и соотношение подмножеств  $Con_n A \subset A \subset dil_n A$  также является действительным.

## Примеры.

### Концентрация нечеткого множества: $n=2$ .

$$A=0.03/1+0.1/2+0.21/3+0.37/4+0.57/5+0.8/6 +0.96/7+1.0/8+0.94/9+ \\ +0.7/10+0.42/11+0.27/12+0.17/13+0.09/14+0.03/15,$$

$$A^2=0.0009/1+0.01/2+0.044/3+0.137/4+0.325/5+0.64/6+0.92/7+1.0/8+ \\ +0.884/9+0.49/10+0.174/11+0.07/12+0/03/13+0/01/14+0/0009/15$$

(см. рис. 3.14).

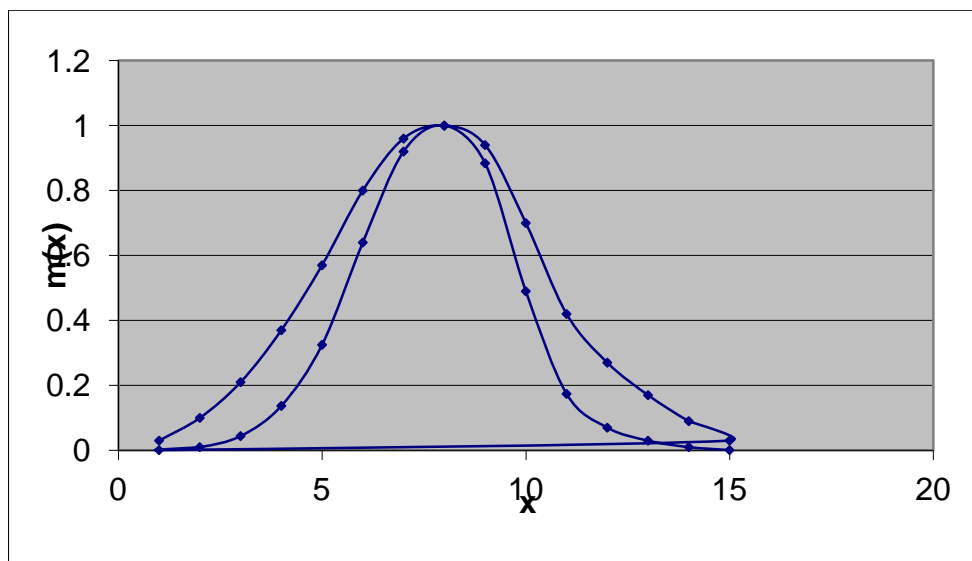


Рис.3.14. Концентрация нечетких множеств

### Растяжение нечеткого множества $n=2$ .

$$A=0.03/2+0.06/3+0.13/4+0.23/5+0.4/6+0.61/7+0.82/8+0.96/9+ \\ +1.0/10+0.94/11+0.74/12+0.51/13+0.33/14+0.23/15+0.16/16+0.1/17+ \\ +0.05/18+0.02/19,$$

$$A^{1/2} \\ =0.17/2+0.25/3+0.36/4+0.48/5+0.63/6+0.78/7+0.9/8+0.98/9+1.0/10+0.97/1 \\ 1+0.86/12+0.72/13+0.57/14+0.48/15+0.4/16+0.3/17+ \\ +0.22/18+0.15/19$$

(см. рис. 3.15).

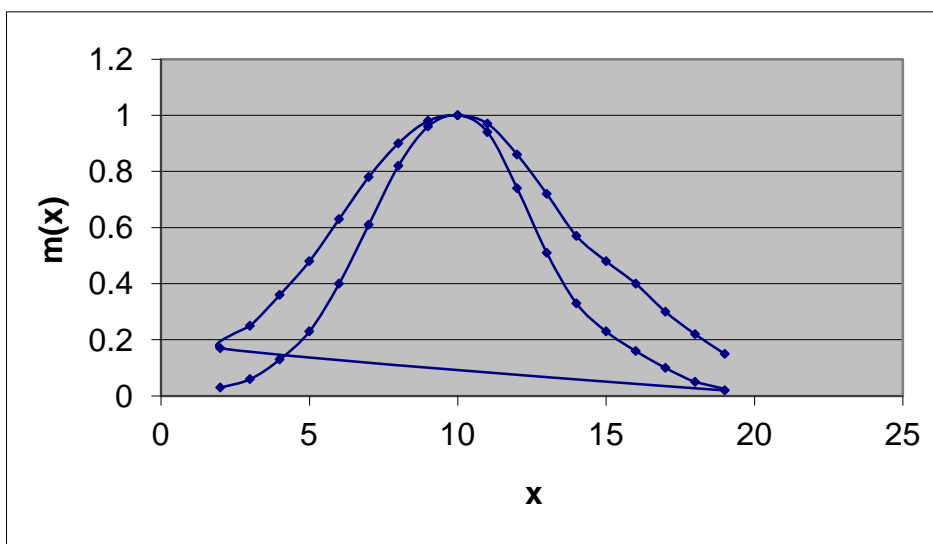


Рис.3.15. Растяжение нечетких множеств

Примеры преобразования нечетких множеств с использованием операций концентрации и растяжения приведены ниже [6].

А	$= \int \mu_A(x) / x$
Очень А	$= \int [\mu_A(x)]^2 / x$
Очень очень А	$= \int [\mu_A(x)]^4 / x$
Более или менее А	$= \int \sqrt{\mu_A(x)} / x$
Немного А	$= \int \sqrt[4]{\mu_A(x)} / x$
Не А	$= \int (1 - \mu_A(x)) / x$
Не очень А	$= \int (1 - [\mu_A(x)]^2) / x$

**Определение 3.15.** Нечеткие точки, нечеткие интервалы, нечеткие области.

Нечеткая точка есть выпуклое нечеткое подмножество действительной прямой  $\mathbb{R}$ .

Как показано в [7], нечеткие точки представляются симметричными интервалами для компонентов, отражающих эту неточность ( в случае гиперпирамидального представления).

В случае эллиптического гиперпараболоидального представления неточность учитывается относительно всех

направлений пространства посредством матрицы, которая играет роль аналогичную ковариационной матрице в математической статистике для случая одной особо наблюдаемой точки  $x_0$ .

Если границами интервала являются нормальные выпуклые нечеткие множества, то он называется нечетким интервалом.

Нечеткие интервалы могут определяться либо с помощью выбора четкого интервала для формирования ядра, от которого функция принадлежности уменьшается до нуля, или посредством выбора двух нечетких чисел в качестве концов интервала. Вообще мы можем построить нечеткий регион в пространстве  $R^k$  посредством выбора четкого региона, окруженного нечеткой переходной зоной, в которой функции принадлежности уменьшается до нуля монотонно. Альтернативный способ представления нечеткой области – это определение нечеткой гиперповерхности, формирующей его границу. Такая нечеткая гиперповерхность имеет четкую гиперповерхность свонго ядра, при удалении от которого значения функции принадлежности монотонно убывают во всех направлениях.

### 3.3. Операции, основанные на t-норм

t-норм – это бинарная операция  $t$  в  $[0,1]$ , т.е. бинарная функция  $t$  из  $[0,1]$  в  $[0,1]$ , которая является коммутативной, ассоциативной и монотонно неубывающей и имеет 1 в качестве нейтрального и 0 – нулевого элемента. При этом для любого  $x, y, z, u, v \in [0,1]$  следующие условия должны удовлетворяться для t-нормы  $t$  [7]:

$$xty=ytx; \quad xt(ytz)=(xty)tz$$

Если  $x \leq u$  и  $y \leq v$ , то  $xty \leq utv$ ;  $xt1=x$  и  $xt0=0$ .

Для каждой t-нормы можно получить операцию пересечения  $\cap_t$  над нечеткими множествами

$$\mu_{A \cap_t B}(x) = \mu_A(x) t \mu_B(x) \text{ для всех } x \in X. \quad (3.29)$$

Все операции пересечения генерируются в такой же форме из подходящих t-норм. Для  $A \cap B$  соответствующий t-норм есть операция  $t_0$  с:

$$ut_0v = \min\{u, v\} \text{ для } u, v \in [0,1]. \quad (3.30)$$

Алгебраическое произведение генерируется из следующей t-нормы  $t_1$ :

$$ut_1v = uv \text{ для } u, v \in [0,1]. \quad (3.31)$$

Ограниченное произведение характеризуется t-нормой  $t_2$ :

$$ut_2v = [u + v - 1]^+ \text{ для } u, v \in [0,1]; \quad (3.32)$$

и строгое (drastic) произведение получается посредством t-нормы  $t_3$  определяемой

$$ut_3v = \begin{cases} \min\{u, v\}, & \text{если } u = 1 \text{ или } v = 1 \text{ для } u, v \in [0,1], \\ 0, & \text{иначе.} \end{cases} \quad (3.33)$$

Используя дополнение (3.19) с операцией пересечения  $\cap_t$ , можно связать некоторую операцию объединения  $\cup_t$ , основанную на двоичной t-норме:

$$A \cup_t B = (\bar{A} \cap_t \bar{B})^-. \quad (3.34)$$

Основная идея операции пересечения и объединения на основе t-норм заключается в замене оператора  $\min$  t-нормой. Эта идея может быть использована также для нечеткого картезианского произведения. При этом используется, основанное на t-норме, нечеткое картезианское произведения

$$\mu_{A \otimes_t B}(u, v) = \mu_A(u) t \mu_B(v), \quad \forall u, v \in X.$$

Как видно, имеется широкий спектр операторов для выполнения операций над нечеткими множествами. Вопрос, в каких ситуациях какие операторы использовать, представляет немаловажный интерес. В [3] приводятся 8 критериев для выбора подходящих операторов: аксиоматическая сила; эмпирическая годность; возможность адаптации; вычислительная эффективность; компенсация; пределы компенсации; поведение операции; требуемый уровень шкалирования функций принадлежности.

### 3.4. Алгебраические операции над F- величинами

Операции сложения, умножения, вычитания и деления, определенные на множестве вещественных чисел, распространяются на класс  $F(R)$  следующим образом. Каждая из отмеченных выше бинарных операций на  $R$  есть отображение [10,44]

$$f: R * R \rightarrow R.$$

Если взять теперь два сегмента  $A=[a,b]$ ,  $B=[c,d]$ , то их сумма, например, определяется отображением

$$f: A * B \rightarrow R,$$

которое для  $\forall x \in A, \forall y \in B$  запишется в виде

$$f(x,y) = z = x+y,$$

где  $(x,y) \in A \times B$ . Очевидно, что

$$A+B = [a+c, b+d].$$

Теперь становится понятным переход к определению алгебраических операций над F-величинами.

Пусть теперь  $A, B \in F(R)$  и  $\circ$  - есть некоторая операция из набора  $\{+, -, *, / \}$ . Учитывая соотношения отображения можно записать:

$$\begin{aligned} \mu_{A \circ B}(z) &= \sup_U \{ \mu_A(x) \wedge \mu_B(y) \} \\ U &= \{x, y\} \in \sigma(A \times B) | x \circ y = z \end{aligned} \quad (3.35)$$

Если декартово произведение определить по второму типу, то получим

$$\mu_{A \circ B}(z) = \sup_U \{ \mu_A(x) \cdot \mu_B(y) \}, \quad (3.36)$$

Общий случай для всех четырех типов декартового произведения запишется в виде:

$$\mu_{A \circ B}(z) = \sup_U f_i \{ \mu_A(x), \mu_B(y) \}, i = \overline{1,4}, \quad (3.37)$$

где  $f_i$  - какая-либо из четырех типов функций, введенных ранее.

Таким образом, для получения F-функции  $\mu_{A \circ B}$  необходимо решить параметрическую задачу на нахождение условного экстремума, т.е. в зависимости от  $z \in R$  найти верхнюю грань функции  $\mu_{A \circ B}$  на множестве  $U$ , задаваемого ограничением (уравнением связи)

$$g(x, y; z) = x \circ y - z = 0. \quad (3.38)$$



Необходимо отметить, что решение поставленной задачи всегда существует в отличие от задачи поиска максимума некоторой функции на заданном множестве [19,28,134].

Если выразить одну из переменных в (3.38) через другую, например,  $y$  через  $x$  в виде  $y = u(x, z)$ , то, подставляя полученное выражение для  $y$  в (3.37), можно преобразовать данную задачу в следующую экстремальную задачу без ограничений, содержащую единственную переменную  $x$ :

$$\mu_{A \circ B}(z) = \sup_x f_i \{ \mu_A(x), \mu_B(u(x, z)) \}. \quad (3.39)$$

Другой подход состоит в использовании множителей Лагранжа. В этом случае задача (3.37) с учетом (3.38) преобразуется к виду

$$\mu_{A \circ B}(z) = \sup_{x, y, z} \{ f_i [\mu_A(x), \mu_B(y)] + \lambda g(x, y, z) \}, \lambda \in R. \quad (3.40)$$

В дальнейшем, если не оговорено противное, алгебраические операции определяются по первому типу, т.е. соотношением (3.35).

Алгебраические операции над  $F$ -величинами обладают следующими свойствами [9,47,97]:

**1. Коммутативность сложения и умножения.** Поскольку для всех четырех типов декартового произведения

$$A \times B = B \times A,$$

а сложение и умножение вещественных чисел коммутативно, то для  $A, B \in F(R)$  из (3.37) следует

$$A+B=B+A, AB=BA. \quad (3.41)$$

**2. Ассоциативность сложения и умножения.** Если сложение и умножение определяются по (3.35) или (3.36), то эти операции ассоциативны, т.е.

$$(A+B)+C=A+(B+C), (AB)C=A(BC). \quad (3.42)$$

Действительно, если символ  $*$  обозначает операцию произведения или взятия минимума, то

$$(\alpha * \beta) * \gamma = \alpha * (\beta * \gamma), \alpha, \beta, \gamma \in R.$$

Пусть

$$\mu_{A+B}(\xi) = \sup_{U_1} \{ \mu_A(x) * \mu_B(y) \},$$

где  $U_1 = \{(x, y) | x + y = \xi\}$ . Тогда

$$\mu_{(A+B)+C}(t) = \sup_{U_2} \{ \mu_{A+B}(\xi) * \mu_C(z) \},$$

где  $U_2 = \{(\xi, z) | \xi + z = t\}$ . Используя (3.39), получим

$$\begin{aligned} \mu_{(A+B)+C}(t) &= \sup_z \{ \mu_{A+B}(t-z) * \mu_C(z) \} = \\ &= \sup_z \{ \sup_{U_1} [\mu_A(x) * \mu_B(y)] * \mu_C(z) \} = \\ &= \sup_z \{ \sup_x [\mu_A(x) * \mu_B(t-z-x)] * \mu_C(z) \} = \\ &= \sup_z \sup_x \{ \mu_A(x) * \mu_B(t-z-x) * \mu_C(z) \} = \\ &= \sup_{z,x} \{ \mu_A(x) * \mu_B(t-z-x) * \mu_C(z) \}. \end{aligned}$$

С другой стороны, поскольку

$$\mu_{B+C}(\xi) = \sup_{U_3} \{ \mu_B(y) * \mu_C(z) \},$$

где  $U_3 = \{(y, z) | y + z = \xi\}$ , то

$$\mu_{A+(B+C)}(t) = \sup_{U_4} \{ \mu_A(x) * \mu_{B+C}(\xi) \},$$

где  $U_4 = \{(x, \xi) | x + \xi = t\}$ . Следовательно,

$$\begin{aligned} \mu_{A+(B+C)}(t) &= \sup_x \{ \mu_A(x) * \mu_{B+C}(t-x) \} = \\ &= \sup_x \{ \mu_A(x) * \sup_{U_3} [\mu_B(y) * \mu_C(z)] \} = \\ &= \sup_x \{ \mu_A(x) * \sup_z [\mu_B(t-z-x) * \mu_C(z)] \} = \\ &= \sup_x \sup_z \{ \mu_A(x) * \mu_B(t-z-x) * \mu_C(z) \} = \\ &= \sup_{z,x} \{ \mu_A(x) * \mu_B(t-z-x) * \mu_C(z) \}. \end{aligned}$$

Таким образом,  $\mu_{(A+B)+C} = \mu_{A+(B+C)}$ . Кроме того, если записать

$$\mu_{A+B+C}(t) = \sup_U \{ \mu_A(x) * \mu_B(y) * \mu_C(z) \},$$

где  $U = \{(x, y, z) | x + y + z = t\}$ , то это можно трактовать как трехместную операцию сложения  $F$ -величин. Делая замену  $y = t - z - x$ , получим, что

$$(A+B)+C = A+(B+C) = A+B+C.$$

**3. Дистрибутивность.** В общем случае это свойство операций не выполняется, т.е.

$$A(B+C) \neq AB+AC.$$

Для получения более точной формулировки закона дистрибутивности необходимо вывести некоторые соотношения. Покажем сначала, что

$$\mu_{A(B+C)}(t) = \sup_U \{ \mu_A(x) \wedge \mu_B(y) \wedge \mu_C(z) \}, \quad (3.43)$$

где  $U = \{(x, y, z) | x(y+z) = t\}$ . Подставляя  $z = t/x - y$  в (3.43), получим

$$\mu_{A(B+C)}(t) = \sup_{x,y} \{ \mu_A(x) \wedge \mu_B(y) \wedge \mu_C(t/x - y) \}.$$

С другой стороны

$$\mu_{B+C}(\tau) = \sup_U \{ \mu_B(y) \wedge \mu_C(z) \},$$

где  $U = \{(y, z) | y + z = \tau\}$ , и следовательно,

$$\mu_{B+C}(\tau) = \sup_y \{ \mu_B(y) \wedge \mu_C(\tau - y) \}.$$

Тогда

$$\mu_{A(B+C)}(t) = \sup_U \{ \mu_A(x) \wedge \mu_{B+C}(\tau) \},$$

где  $U = \{(x, \tau) | x\tau = t\}$ , т.е.

$$\begin{aligned} \mu_{A(B+C)}(t) &= \sup_x \{ \mu_A(x) \wedge \mu_{B+C}(t/x) \} = \\ &= \sup_x \{ \mu_A(x) \wedge \sup_y [\mu_B(y) \wedge \mu_C(t/x - y)] \} = \\ &= \sup_x \sup_y \{ \mu_A(x) \wedge \mu_B(y) \wedge \mu_C(t/x - y) \} = \\ &= \sup_{x,y} \{ \mu_A(x) \wedge \mu_B(y) \wedge \mu_C(t/x - y) \} \end{aligned}$$

что и требовалось показать. Аналогично показывается, что

$$\mu_{AB+AC}(t) = \sup_{xy} \{ \mu_A(\xi) \wedge \mu_A(\zeta) \wedge \mu_B(y) \wedge \mu_C(z) \}, \quad (3.44)$$

где  $U = \{(\xi, \zeta, y, z) | \xi y + \zeta z = t\}$ .

Сравним значения функций  $\mu_{A(B+C)}$  и  $\mu_{AB+AC}$  в некоторой произвольной точке  $t_0$ . Предположим, что  $x_0, y_0, z_0$  такие, что  $x_0(y_0 + z_0) = t_0$  и, кроме того,

$$\mu_{A(B+C)}(t_0) = \mu_A(x_0) \wedge \mu_B(y_0) \wedge \mu_C(z_0).$$

Если положить  $\xi_0 = \zeta_0 = x_0$ , то на основании равенства

$$\mu_A(\xi_0) \wedge \mu_A(\zeta_0) \wedge \mu_B(y_0) \wedge \mu_C(z_0) = \mu_{A(B+C)}(t_0)$$

из (3.44) следует, что не исключена возможность существования таких  $\xi, \zeta, y, z$  с условием  $\xi y + \zeta z = t_0$ , что

$$\mu_{A(B+C)}(t_0) < \mu_{AB+AC}(t_0)$$

и, следовательно,

$$A(B+C) \subseteq AB+AC. \quad (3.45)$$

Если  $A$  - выпуклая  $F$ -величина и  $b, c$  такие вещественные числа, что  $|b+c| = |b|+|c|$ , то выполняется равенство

$$(b+c)A = bA + cA. \quad (3.46)$$

Последнее свойство, которое отметим в данном разделе, заключается в том, что при  $A = \langle 1, a \rangle$  для произвольных  $B, C \in F(R)$  из (3.43) и (3.44) следует равенство

$$a(B+C) = aB + aC. \quad (3.47)$$

### **Прямой аналитический метод нахождения результатов алгебраических операций**

Для нахождения результатов алгебраических операций над  $F$ -величинами используются несколько аналитических и численных методов [10,112]. Причем, если решение ищется для общего вида записи задачи (3.37), то метод называют "**прямым**". Если метод основан на некоторой переформулировке исходной задачи с использованием  $\alpha$ -уровней, то его называют "**обратным**" или методом  $\alpha$ -уровневых сечений. Как и прежде, основное внимание будет уделяться операциям первого типа, которые определяются посредством (1.3.1).

**Умножение  $F$ -величин на скаляр.** Если  $B = \lambda = (1, \lambda)$ , то в силу взаимной однозначности отображения  $z = \lambda x$ , из (1.3.1) получим

$$\mu_{\lambda A}(z) = \mu_A(z/\lambda), \lambda \neq 0. \quad (3.48)$$

Если  $\lambda = 0$ , то очевидно, что

$$\lambda A = \left\langle \sup_x \mu_A(x), 0 \right\rangle, \quad (3.49)$$

т.е. если  $A$  - нормальная  $F$ -величина, то  $0 \cdot A = 0$ .

**Сложение  $F$ -величины со скаляром.** Аналогично предыдущему случаю, если  $B = \lambda = (1, \lambda)$ , и, следовательно,  $z = x + \lambda$ , то

$$\mu_{A+\lambda}(z) = \mu_A(z - \lambda), \lambda \in R, A \in F(R). \quad (3.50)$$

Этим самым осуществляется сдвиг функции  $\mu_A$  вправо или влево по вещественной оси на величину  $|\lambda|$ .

Нетрудно проверить, что соотношения (3.48)-(3.50) справедливы и для алгебраических операций второго типа.

Пусть  $A = (1 - (x-1)^2, (0, 1))$ . Тогда, согласно (3.48) и (3.49) имеем

$$\lambda A = \langle 1 - (x - \lambda)^2 / \lambda^2, (0, 2\lambda) \rangle, \lambda > 0,$$

$$\lambda A = \langle 1 - (x - \lambda)^2 / \lambda^2, (2\lambda, 0) \rangle, \lambda < 0,$$

$$\lambda A = 0, \lambda = 0.$$

Из (3.49) получим

$$\lambda + A = \langle 1 - (x - (\lambda + 1))^2, (\lambda, \lambda + 2) \rangle.$$

В этом разделе было отмечено, что нахождение  $F$ -величины  $A \circ B$ , т.е. нахождение ее  $F$ -функции, сводится к решению параметрической экстремальной задачи (3.37). Причем, с помощью замены из уравнения связи  $x \circ y = z$  данная задача преобразуется в задачу без ограничений (3.39), т.е. для операций первого типа получим

$$\mu_{A \circ B}(z) = \sup_x \{ \mu_A(x) \wedge \mu_B(u(x, z)) \}. \quad (3.51)$$

Из теории экстремальных задач [24,27,41,67,80] хорошо известно, что нахождение глобального максимума некоторой функции на заданном множестве  $U$  из  $R$  существенно упрощается, если эта функция унимодальна, т.е. имеет на  $U$  единственный максимум.

Нетрудно заметить, что если  $F$ -величина  $A$  строго выпукла и функция  $\mu_A$  достигает на  $\sigma(A)$  своей верхней грани, то  $\mu_A$  унимодальна на  $\sigma(A)$ . Если  $A$ -выпуклая, то это уже не так. Тем не менее, даже для выпуклой  $F$ -величины нахождение верхней грани ее  $F$ -функции значительно проще, чем для  $F$ -величины с произвольной  $F$ -функцией.

Следовательно, предпочтительнее решать задачу (3.51) для выпуклых  $F$ -величин, поскольку функция  $\mu_A(x) \wedge \mu_B(u(x, z))$  определяет выпуклую  $F$ -величину. Исключение составляет операция умножения, когда множества  $\sigma(A)$  и  $\sigma(B)$  содержат нуль в качестве внутренней точки. Кроме того, для выпуклых  $F$ -величин справедливо следующее утверждение.

Если  $A$  и  $B$  - выпуклые, то  $C = A \circ B$  - выпуклая  $F$ -величина.

Рассмотрим операцию сложения. Пусть  $z_1, z_2 \in \sigma(C)$ ,  $z_1 = x_1 + y_1$ ,  $z_2 = x_2 + y_2$ , и  $\mu_C(z_1) = \mu_A(x_1) \wedge \mu_B(y_1)$ ,  $\mu_C(z_2) = \mu_A(x_2) \wedge \mu_B(y_2)$ . Тогда для произвольного  $\gamma \in [0, 1]$  имеем

$$\begin{aligned}
\mu_C(\gamma z_1 + (1-\gamma)z_2) &= \mu_C(\gamma(x_1 + y_1) + (1-\gamma)(x_2 + y_2)) = \\
&= \mu_C(\gamma x_1 + (1-\gamma)x_2 + \gamma y_1 + (1-\gamma)y_2) \geq \\
&\geq \mu_A(\gamma x_1 + (1-\gamma)x_2) \wedge \mu_B(\gamma y_1 + (1-\gamma)y_2) \geq \\
&\geq \mu_A(x_1) \wedge \mu_A(x_2) \wedge \mu_B(y_1) \wedge \mu_B(y_2) = \\
&= (\mu_A(x_1) \wedge \mu_B(y_1)) \wedge (\mu_A(x_2) \wedge \mu_B(y_2)) = \\
&= \mu_C(z_1) \wedge \mu_C(z_2)
\end{aligned}$$

что и требовалось показать.

Поскольку  $C=A-B=A+(-B)$ , а  $(-B)$ - выпуклая  $F$ -величина, то  $C=A-B$  тоже выпуклая  $F$ -величина.

Рассмотрим операцию умножения, т.е.  $C=A \cdot B$ . Пусть  $z_1, z_2 \in \sigma(C)$ ,  $z_1 = x_1 \cdot y_1$ ,  $z_2 = x_2 \cdot y_2$  и  $\mu_C(z_1) = \mu_A(x_1) \wedge \mu_B(y_1)$ ,  $\mu_C(z_2) = \mu_A(x_2) \wedge \mu_B(y_2)$ . Положим для определенности, что  $x_1 < x_2$ ,  $y_1 < y_2$ ,  $z_1 < z_2$ . Тогда для произвольного  $z \in (z_1, z_2)$  найдутся  $x \in (x_1, x_2)$  и  $y \in (y_1, y_2)$  с условием  $z = x \cdot y$ . Пусть  $\gamma_0, \gamma_1, \gamma_2 \in [0,1]$ , для которых

$$\begin{aligned}
z &= \gamma_0 z_1 + (1-\gamma_0) z_2, \\
x &= \gamma_1 x_1 + (1-\gamma_1) x_2, \\
y &= \gamma_2 y_1 + (1-\gamma_2) y_1.
\end{aligned}$$

Тогда получим

$$\begin{aligned}
\mu_{AB}(z) &= \mu_{AB}(\gamma_0 z_1 + (1-\gamma_0) z_2) \geq \mu_A(x) \wedge \mu_B(y) \geq \\
&\geq \mu_A(x_1) \wedge \mu_A(x_2) \wedge \mu_B(y_1) \wedge \mu_B(y_2) = \\
&= (\mu_A(x_1) \wedge \mu_B(y_1)) \wedge (\mu_A(x_2) \wedge \mu_B(y_2)) = \\
&= \mu_{AB}(z_1) \wedge \mu_{AB}(z_2),
\end{aligned}$$

что и требовалось показать.

Для операции деления утверждение показывается аналогично, причем, если  $C=A/B$ , то  $0 \notin \sigma(B)$ .

Предположение о выпуклости  $F$ -величин обусловлено также тем, что большинство функций принадлежности на практике являются выпуклыми. В некоторых случаях может оказаться полезным следующий подход к решению задачи (3.37), который можно назвать принципом декомпозиции. Если возникает ситуация, когда  $A$  и  $B$  - невыпуклые, то их можно представить в виде объединения выпуклых  $F$ -величин. Из определения

алгебраических операций над  $F$ -величинами нетрудно заметить, что если  $A = \bigcup_{i=1}^n A_i$ ,  $B = \bigcup_{j=1}^n B_j$  то

$$A \circ B = \bigcup_{i,j} (A_i \circ B_j), i = \overline{1, m}, j = \overline{1, n}. \quad (3.52)$$

Следовательно, если  $A$  и  $B$  - невыпуклые, то представление их в виде объединения выпуклых  $F$ -величин в некоторых случаях может облегчить решение задачи (3.37). Имея ввиду все отмеченные выше обстоятельства, в дальнейшем будем предполагать, что все используемые  $F$ -величины являются выпуклыми.

В основе прямого аналитического метода для бинарных операций лежит классический подход к поиску точек экстремума функции на некотором множестве из  $R$ . Будем предполагать, что функция  $\mu_A$  всегда достигает своей верхней грани на  $\sigma(A)$  кроме тех случаев, когда верхняя грань достигается в точках  $-\infty$  или  $+\infty$ . Тогда, как известно [41], точками экстремума функции  $\mu_A(x)$  на  $\sigma(A)$  могут быть лишь те точки, в которых выполняется одно из следующих условий:

1. либо  $\mu_A(x)$  терпит разрыв;
2. либо  $\mu_A(x)$  непрерывна, но производная  $\mu'_A$  не существует;
3. либо  $\mu'_A$  существует и равна нулю;
4. либо  $x=a$  или  $x=b$ , если  $\sigma(A)=[a, b]$ .

Если множество  $\sigma(A)$  неограниченно, то нужно также изучить поведение функции  $\mu_A(x)$  при  $x \rightarrow -$  или  $+\infty$ .

Рассмотрим теперь каждую из операций над  $F$ -величинами.

**Сложение  $F$ -величин.** В этом случае уравнение связи имеет вид  $x+y=z$ , т.е. для произвольного фиксированного  $z_0$  величина  $\mu_{A+B}(z_0)$  равна верхней грани функции  $f_1(x, y) = \mu_A(x) \wedge \mu_B(y)$  для (3.35) на прямой в  $R^2$  с уравнением  $y=z_0-x$ . Соотношение (3.51) запишется в виде

$$\mu_{A+B}(z) = \sup_x \{ \mu_A(x) \wedge \mu_B(z-x) \}. \quad (3.53)$$

Из (3.53) видно, что значениями функции  $\mu_{A+B}$  являются верхние грани семейства  $F$ -величин  $A \cap (-B - z)$ , зависящих от  $z$  как

от параметра. Если в зависимости от  $z$  экстремальные точки функции  $\mu_A(x) \wedge \mu_B(z-x)$  можно выразить в виде соотношений:

$$x = \varphi_1(z), x = \varphi_2(z), \dots, x = \varphi_n(z),$$

то получим

$$C = A + B = \bigcup_{i=1}^n C_i,$$

где

$$\mu_{C_i}(z) = \mu_A(\varphi_i(z)) \wedge \mu_B(z - \varphi_i(z)).$$

Следует отметить то обстоятельство, что точка глобального максимума функции  $\mu_A(x) \wedge \mu_B(z-x)$  в некоторых случаях может быть получена непосредственно из решения уравнения

$$\mu_A(x) = \mu_B(z-x).$$

Проиллюстрируем выше изложенное рядом простых примеров.

Пусть  $\mu_A(x) = \exp\{-(x-a)^2/b\}$ ,  $\mu_B(y) = \exp\{-(y-d)^2/c\}$ ,  $b, c > 0$ . Тогда из уравнения  $\mu_A(x) = \mu_B(z-x)$  имеем  $(x-a)/\sqrt{b} = \pm(z-x-d)/\sqrt{c}$ , откуда

$$x_1 = \varphi_1(z) = (z\sqrt{b} + a\sqrt{c} - d\sqrt{b})/(\sqrt{b} + \sqrt{c}),$$

$$x_2 = \varphi_2(z) = (-z\sqrt{b} + a\sqrt{c} + d\sqrt{b})/(\sqrt{c} - \sqrt{b}),$$

и, следовательно,  $\mu_{C_1}(z) = \mu_A(\varphi_1(z))$ ,  $\mu_{C_2}(z) = \mu_A(\varphi_2(z))$ .

Делая соответствующие подстановки, после несложных преобразований получим

$$\mu_{C_1}(z) = \exp\{-z - (a+d))^2 / (\sqrt{b} + \sqrt{c})^2\},$$

$$\mu_{C_2}(z) = \exp\{-(z-a+d))^2 / (\sqrt{b} - \sqrt{c})^2\}.$$

Поскольку  $(\sqrt{b} - \sqrt{c})^2 < (\sqrt{b} + \sqrt{c})^2$ , то  $C = C_1 \cup C_2 = C_1$ , т.е.  $\mu_{A+B}(z) = \mu_{C_1}(z)$ .

Если  $A = \langle 1 - (x-a)^2/b, (a - \sqrt{b}, a + \sqrt{b}) \rangle$  и  $B = \langle 1 - (y-d)^2/c, (d - \sqrt{c}, d + \sqrt{c}) \rangle$ , то по аналогии с предыдущим примером из анализа уравнения

$$1 - (x-a)^2/b = 1 - (z-x-d)^2/c$$

получим, что

$$C = A + B = \langle 1 - (z-a-d)^2 / (\sqrt{b} + \sqrt{c})^2, (a+d - (\sqrt{b} + \sqrt{c}), a+d + (\sqrt{b} + \sqrt{c})) \rangle.$$

Пусть  $A = A_1 \cup A_2 \cup A_3$  и  $B = B_1 \cup B_2 \cup B_3$ , где



$$\begin{aligned}
A_1 &= \langle a_1x + b_1, [-b_1/a_1, (1-b_1)/a_1] \rangle, a_1 > 0, \\
A_2 &= \langle 1, [(1-b_1)/a_1, (1-b_2)/a_3] \rangle, \\
A_3 &= \langle a_2x + b_2, [(1-b_2)/a_2, -b_3/a_2] \rangle, a_2 < 0, \\
B_1 &= \langle a_3y + b_3, [-b_3/a_3, (1-b_3)/a_3] \rangle, a_3 > 0, \\
B_2 &= \langle 1, [(1-b_3)/a_3, (1-b_4)/a_4] \rangle, \\
B_3 &= \langle a_4y + b_4, [(1-b_4)/a_4, -b_4/a_4] \rangle, a_4 < 0.
\end{aligned}$$

В данном случае можно воспользоваться принципом декомпозиции, причем соответствующий анализ показывает, что

$$C = A + B = (A_1 \cup B_1) \cup (A_2 \cup B_2) \cup (A_3 \cup B_3).$$

Из уравнения  $a_1x + b_1 = a_3(z - x) + b_3$  находим, что  $x = \varphi(z) = a_3z/(a_1 + a_3) + (b_3 - b_1)/(a_1 + a_3)$  и, следовательно,

$$C_1 = A_1 + B_1 = \langle a_1a_3z + b_1a_3 + b_3a_1/(a_1 + a_3), [-(b_1a_3 + b_3a_1)/(a_1 + a_3), (1-b_1)/a_1 + (1-b_3)/a_3] \rangle$$

По аналогии для  $C_3$  имеем

$$C_3 = A_3 + B_3 = \langle a_2a_4z + b_2a_4 + b_4a_2/(a_2 + a_4), [(1-b_2)/a_2 + (1-b_4)/a_4, -(b_2a_4 + b_4a_2)/(a_2 + a_4)] \rangle$$

И, наконец,

$$C_2 = A_2 + B_2 = \langle 1, [(1-b_1)/a_1 + (1-b_3)/a_3, (1-b_2)/a_2 + (1-b_4)/a_4] \rangle$$

Приведем один пример на сложение второго типа, определяемого по (1.3.2).

Пусть  $\mu_A(x) = \exp\{-(x-a)^2/b\}$  и  $\mu_B(y) = \exp\{-(y-d)^2/c\}$ ,  $b, c > 0$ . Тогда из уравнения

$$\frac{d}{dx} \mu_A(x) \cdot \mu_B(z-x) = 0$$

получим, что

$$x = \varphi(z) = (bz - bd + ac)/(b + c)$$

и, следовательно,

$$\mu_{A+B}(z) = \mu_A(\varphi(z)) \cdot \mu_B(z - \varphi(z)) = \exp\{-(z - (a+d))^2/(b+c)\}.$$

**Вычитание F-величин.** В этом случае уравнение связи имеет вид

$$z = x - y,$$

т.е. для произвольного фиксированного  $z_0$  величина  $\mu_{A-B}(z_0)$  равна верхней грани функции  $f_1(x, y) = \mu_A(x) \wedge \mu_B(y)$  на прямой в  $R^2$  с уравнением  $y = z_0 - x$ . Соотношение (3.51) запишется в виде

$$\mu_{A-B}(z) = \sup_x \{\mu_A(x) \wedge \mu_B(x - z)\}.$$

Поскольку  $A-B=A+(-B)$ , вычитание сводится к сложению.

Пусть  $A = \langle 1 - (x-a)^2, (a-1, a+1) \rangle$  и  $B = \langle 1 - (y-b)^2, b-1, b+1 \rangle$ .

Тогда из уравнения  $1 - (x-a)^2 = 1 - (x-z-b)^2$  получаем

$$x^2 - 2ax + a^2 = x^2 - xz - bx - xz + z^2 + bz - bx + bz + b^2$$

$$2x(z-a+b) = z^2 + 2bz + b^2 - a^2$$

$$x = \frac{(z+b)^2 - a^2}{2(z-a+b)} = \frac{z+a+b}{2}.$$

Делая подстановку  $\mu_C(z) = \mu_A(\varphi(z))$ , находим

$$\mu_C(x) = 1 - \left( \frac{z+a+b}{2} - a \right)^2 = 1 - \left( \frac{z-a+b}{2} \right)^2.$$

Таким образом,

$$C = A - B = \langle 1 - (z-a+b)^2/4, (a-b-2, a-b+2) \rangle.$$

**Умножение F-величин.** В этом случае уравнения связи имеет вид

$$z = x y,$$

т.е. для произвольного фиксированного  $z_0$  величина  $\mu_{A \cdot B}(z_0)$  равна верхней грани функции  $f_1(x, y) = \mu_A(x) \wedge \mu_B(y)$  на гиперболе в  $\mathbf{R}^2$  с уравнением  $y = z_0/x$ . Соотношение (3.51) запишется в виде

$$\mu_{AB}(z) = \sup_x \{ \mu_A(x) \wedge \mu_B(z/x) \}.$$

Ввиду нелинейности ограничения нахождение произведения F-величин представляет собой более трудную задачу, чем суммирование и вычитание.

Пусть  $A = \langle 1 - \frac{1}{x^2}, (1, +\infty) \rangle$  и  $B = \langle 1 - \frac{1}{y^2}, (2, +\infty) \rangle$ . Тогда из уравнения

$$1 - \frac{1}{x^2} = 1 - \frac{4}{(z/x)^2}$$

получаем

$$4x^4 = z^2.$$

$$x_{1,2} = \varphi(z) = \pm \sqrt{z/2}.$$

Делая соответствующую подстановку  $\mu_A(z) = \mu_A(\varphi(z))$ , получаем

$$C = A \cdot B = \left\langle 1 - \frac{2}{z}, (2, +\infty) \right\rangle.$$

Пусть  $\mu_A(x) = \exp\{-(x-a)^2/b\}$  и  $\mu_B(y) = \exp\{-(y-d)^2/c\}$ ,  $b, c > 0$ . Из уравнения  $\mu_A(x) = \mu_B(z/x)$  получим  $x\sqrt{c}(x-a) = -\sqrt{b}(z-xd)$ ,  $x\sqrt{c}(x-a) = \sqrt{b}(z-xd)$ .

Из первого уравнения имеем  $x^2\sqrt{c} - x(a\sqrt{c} + d\sqrt{b}) + z\sqrt{b} = 0$ , откуда

$$x_{1,2} = \left( a\sqrt{c} + d\sqrt{b} \pm \sqrt{(a\sqrt{c} + d\sqrt{b})^2 - 4z\sqrt{bc}} \right) / 2\sqrt{c}.$$

Следовательно,

$$\begin{aligned} \mu_{C_1}(z) &= \exp\{-(d\sqrt{b} - a\sqrt{c} + \sqrt{L(z)})^2 / 4bc\}, \\ \mu_{C_2}(z) &= \exp\{-(d\sqrt{b} - a\sqrt{c} - \sqrt{L(z)})^2 / 4bc\}, \end{aligned}$$

где

$$L(z) = (a\sqrt{c} + d\sqrt{b})^2 - 4z\sqrt{bc}.$$

Осталось рассмотреть еще одно уравнение  $x^2\sqrt{c} - x(a\sqrt{c} - d\sqrt{b}) - z\sqrt{b} = 0$  из которого

$$x_{3,4} = \left( a\sqrt{c} - d\sqrt{b} \pm \sqrt{(a\sqrt{c} - d\sqrt{b})^2 + 4z\sqrt{bc}} \right) / 2\sqrt{c}.$$

Обозначая  $M(z) = (a\sqrt{c} - d\sqrt{b})^2 + 4z\sqrt{bc}$ , получим

$$\begin{aligned} \mu_{C_3}(z) &= \exp\{-(a\sqrt{c} + d\sqrt{b} - \sqrt{M(z)})^2 / 4bc\}, \\ \mu_{C_4}(z) &= \exp\{-(a\sqrt{c} + d\sqrt{b} + \sqrt{M(z)})^2 / 4bc\}. \end{aligned}$$

Учитывая, что  $\mu_{AB}(ad) = 1$  и, рассматривая арифметические значения корней, окончательно имеем:

$$\begin{aligned} C = AB &= C_1 \cup C_3, |a+d| = |a| + |d|, \\ C = AB &= C_2 \cup C_4, |a+d| \neq |a| + |d|. \end{aligned}$$

Рассмотрим частный случай данного примера. Пусть  $a = d, b = c$ , т.е.  $A=B$ . Тогда для  $a \geq 0$  получим

$$\begin{aligned} \mu_{C_1}(z) &= \exp\{-(a^2 - z)/b\}, z \leq a^2, \\ \mu_{C_3}(z) &= \exp\{-(a - \sqrt{z})^2/b\}, z \geq 0. \end{aligned}$$

Поскольку при  $0 < z < a^2$  имеем

$$(a^2 - z) - (a - \sqrt{z})^2 = a^2 - z - a^2 + 2a\sqrt{z} - z = 2\sqrt{z}(a - \sqrt{z}) > 0,$$

то окончательно

$$\mu_{AA}(z) = \begin{cases} \exp\{-a - \sqrt{z}\}^2 / b, & z \geq 0, \\ \exp\{-a^2 - z\} / b, & z \leq 0. \end{cases}$$

Если  $a \leq 0$ , то

$$\mu_{C_2}(z) = \exp\{-(a^2 - z) / b\}, z \leq a^2,$$

$$\mu_{C_4}(z) = \exp\{-(a - \sqrt{z})^2 / b\}, z \geq 0.$$

Таким образом,  $C_1 = C_2, C_3 = C_4$ , т.е. результат не зависит от положения точки  $a$ . При  $a=0$  получаем

$$\mu_A(z) = \begin{cases} \exp\{-z / b\}, & z \geq 0 \\ \exp\{z / b\}, & z \leq 0 \end{cases}$$

Интересно отметить, что если отображение  $z = x^2$  трактовать как возведение  $F$ -величины  $A$  в квадрат, то при  $a=0$  получим

$$\mu_{A^2}(z) = \exp\{-z / b\}, z \geq 0,$$

т.е. в этом смысле  $A \cdot A \neq A^2$ . Очевидно, что это справедливо для любой  $F$ -величины, носитель которой содержит нуль в качестве внутренней точки.

**Деление  $F$ -величин.** Уравнение связи в этом случае имеет вид

$$z = x / y, y \neq 0,$$

т.е. для произвольного фиксированного  $z_0$  величина  $\mu_{A/B}(z_0)$  равна верхней грани функции  $f_1(x, y) = \mu_A(x) \wedge \mu_B(y)$  на прямой в  $R^2$  с уравнением  $x = z_0 y$ . Следовательно, можно записать

$$\mu_{A/B}(z) = \sup_x \{\mu_A(zx) \wedge \mu_B(x)\}.$$

Вообще говоря, операция деления  $F$ -величины  $A$  на  $B$  сводится к умножению  $A$  на  $1/B$ . С другой стороны, ввиду линейности ограничения  $x = zy$  операция деления во многих случаях значительно проще операции умножения.

**Обратный аналитический метод нахождения результатов алгебраических операций**

Пусть  $X$  и  $Y$  - произвольные базовые множества,  $A \in F(X)$  и задано отображение  $f: X \rightarrow Y$ . Если для  $\forall y \in Y \exists x \in X$  такой, что  $\mu_{f(A)}(y) = \mu_A(x)$ , то справедливо равенство

$$\sigma_\alpha(f(A)) = f(\sigma_\alpha(A)).$$

Пусть  $y_0 \in f(\sigma_\alpha(A))$  и  $V \subseteq \sigma_\alpha(A)$  такое, что  $f(x) = y_0, x \in V$ . Тогда  $\mu_{f(A)}(y_0) = \sup_V \mu_A(x) > \alpha$ , т.е.  $y_0 \in \sigma_\alpha(f(A))$  и, следовательно,  $f(\sigma_\alpha(A)) \subseteq \sigma_\alpha(f(A))$ .

С другой стороны, пусть  $y_0 \in \sigma_\alpha(f(A))$ , т.е.  $\mu_{f(A)}(y_0) > \alpha$ . Тогда по условию существует  $x_0 \in \sigma_\alpha(A)$  такой, что  $y_0 = f(x_0)$ . Поскольку  $y_0 = f(x_0) \in f(\sigma_\alpha(A))$ , то  $\sigma_\alpha(f(A)) \subseteq f(\sigma_\alpha(A))$ .

Если  $A, B \in F(R)$ , то

$$\sigma_\alpha(A \circ B) = \sigma_\alpha(A) \circ \sigma_\alpha(B).$$

Как было отмечено ранее, алгебраические операции определяются отображением  $f: R * R \rightarrow R$ , т.е. для  $F$ -величин  $A$  и  $B$  имеем  $f(A * B) = A \circ B$ . Учитывая, что в данном случае справедливо равенство  $\sigma_\alpha(A \times B) = \sigma_\alpha(A) \times \sigma_\alpha(B)$ , с учетом  $\sigma_\alpha(f(A)) = f(\sigma_\alpha(A))$  получаем

$$\sigma_\alpha(A \circ B) = \sigma_\alpha(f(A \times B)) = f(\sigma_\alpha(A \times B)) = f(\sigma_\alpha(A) \times \sigma_\alpha(B)) = \sigma_\alpha(A) \circ \sigma_\alpha(B)$$

что и требовалось показать.

$F$ -величина  $A$  называется ограниченной, если  $\sigma(A)$  - ограниченное множество. Подкласс из  $F(R)$  ограниченных и выпуклых  $F$ -величин обозначим через  $\overline{F}(R)$ .

**Суть обратного метода** решения задачи

$$\mu_{A \circ B}(z) = \max_U \{ \mu_A(x) \wedge \mu_B(y) \},$$

$$U = \{ (x, y) \in \sigma(A \times B) \mid x \circ y = z \}$$

для ограниченных и выпуклых  $F$ -величин заключается в следующем. Если  $A, B \in \overline{F}(R)$ , то для произвольного  $\alpha$  из  $[0, 1]$ , пользуясь указанным следствием, можно найти  $\alpha$ -уровневое множество для  $A \circ B$ . Этим самым определяется элемент

$$f_{A \circ B}(\alpha) = (z_1(\alpha), z_2(\alpha)).$$

Разрешая уравнения  $z_1 = z_1(\alpha)$  и  $z_2 = z_2(\alpha)$  относительно  $\alpha$ , получим образ элемента  $f_{A \circ B}$  в классе  $\overline{F}(R)$ , т.е.  $F$ -функцию  $\mu_{A \circ B}$ . Таким образом, название данного метода вполне согласуется с идеей, положенной в его основу.

Пусть  $\mu_A(x) = \exp\{-(x-a)^2/b\}$ ,  $\mu_B(y) = \exp\{-(y-d)^2/c\}, b, c > 0$ . Найдем  $F$ -величину  $A+B$ . Имеем

$$\mu_A(x) = \exp\{-(x-a)^2/b\} = t, \quad \mu_B(y) = \exp\{-(y-d)^2/c\} = t,$$

откуда

$$-b \cdot \ln t = (x-a)^2, \quad -c \cdot \ln t = (y-d)^2,$$

следовательно,

$$\begin{aligned} x_1(t) &= a - \sqrt{-b \cdot \ln t}, & x_2(t) &= a + \sqrt{-b \cdot \ln t}, \\ y_1(t) &= d - \sqrt{-c \cdot \ln t}, & y_2(t) &= d + \sqrt{-c \cdot \ln t}. \end{aligned}$$

Далее

$$\begin{aligned} z_1(t) &= x_1(t) + y_1(t) = (a+d) - \sqrt{-\ln t}(\sqrt{b} + \sqrt{c}), \\ z_2(t) &= x_2(t) + y_2(t) = (a+d) + \sqrt{-\ln t}(\sqrt{b} + \sqrt{c}), \end{aligned}$$

откуда

$$-\ln t = (z_i(t) - (a+d))^2 / (\sqrt{b} + \sqrt{c})^2, \quad i = 1, 2,$$

т.е. для всех  $\mathbf{z}$  имеем

$$t = \mu_{A+B}(z) = \exp\{-(z - (a+d))^2 / (\sqrt{b} + \sqrt{c})^2\},$$

что совпадает с результатом, полученным прямым методом.

Рассмотрим операцию вычитания для  $A = \langle 1 - (x-a)^2, (a-1, a+1) \rangle$

и  $B = \langle 1 - (y-b)^2, (b-1, b+1) \rangle$ . Тогда из уравнений

$$\mu_A(x) = 1 - (x-a)^2 = t, \quad \mu_B(y) = 1 - (y-b)^2 = t$$

получим

$$\begin{aligned} x_1(t) &= a - \sqrt{1-t}, & x_2(t) &= a + \sqrt{1-t}, \\ y_1(t) &= b - \sqrt{1-t}, & y_2(t) &= b + \sqrt{1-t}. \end{aligned}$$

Далее из соотношений

$$\begin{aligned} z_1(t) &= x_1(t) - y_2(t) = (a-b) - 2\sqrt{1-t}, \\ z_2(t) &= x_2(t) - y_1(t) = (a-b) + 2\sqrt{1-t} \end{aligned}$$

следует, что для всех  $\mathbf{z}$  имеем

$$t = \mu_{A-B}(z) = 1 - (z - a + b)^2 / 4$$

т.е.

$$C = A - B = \langle 1 - (z - a + b)^2 / 4, (a-b-2, a-b+2) \rangle$$

что совпадает с результатом, полученным прямым методом.

Рассмотрим операцию умножения для  $A = \langle 1 - \frac{1}{x^2}, (1, +\infty) \rangle$  и

$B = \langle 1 - \frac{4}{y^2}, (2, +\infty) \rangle$ . В этом случае имеем

$$\mu_A(x) = 1 - 1/x^2 = t, \quad \mu_B(y) = 1 - 4/y^2 = t$$

откуда

$$x_{1,2}(t) = \pm 1/\sqrt{1-t}, \quad y_{1,2}(t) = \pm 2/\sqrt{1-t}.$$

В нашем случае

$$z(t) = x_1(t) \cdot y_1(t) = \frac{2}{1-t}.$$

Таким образом, для всех  $z > 0$  имеем

$$t = \mu_{A \cdot B}(z) = 1 - 2/z,$$

т.е.

$$C = A \cdot B = \langle 1 - 2/z, (2, +\infty) \rangle,$$

что совпадает с результатом, полученным прямым методом.

Рассмотрим операцию деления для  $A = \langle 1 - (x - a)^2, (a - 1, a + 1) \rangle$  и  $B = \langle 1 - (y - b)^2, (b - 1, b + 1) \rangle$ . Из уравнений

$$\mu_A(x) = 1 - (x - a)^2 = t, \quad \mu_B(y) = 1 - (y - b)^2 = t,$$

имеем

$$\begin{aligned} x_1(t) &= a - \sqrt{1-t}, & x_2(t) &= a + \sqrt{1-t}, \\ y_1(t) &= b - \sqrt{1-t}, & y_2(t) &= b + \sqrt{1-t}. \end{aligned}$$

В этом случае

$$z_1(t) = x_1(t)/y_2(t), \quad z_2(t) = x_2(t)/y_1(t).$$

Для решения достаточно одного уравнения. Следовательно,

$$z(t) = (a - \sqrt{1-t}) / (b + \sqrt{1-t}),$$

откуда

$$\sqrt{1-t} = (a - bz) / (z + 1),$$

т.е.

$$t = \mu_{A/B}(z) = 1 - (a - bz)^2 / (z + 1)^2.$$

При  $a=4$  и  $b=2$

$$C = A/B = \left\langle 1 - \frac{4 \cdot (2-z)^2}{(z+1)^2}, (1,5) \right\rangle.$$

Рассмотренные аналитические методы построения функции принадлежности  $\mu_{A \circ B}$  позволяют получить результат операции сразу в аналитическом виде, что очень удобно для практических приложений. Однако на практике могут встречаться более сложные аналитические выражения для исходных  $F$ -величин, для которых имеются значительные трудности при нахождении аналитического решения. К тому же, иногда возникает необходимость в численных методах работы с дискретно заданными  $F$ -функциями. В этом случае  $F$ -величина  $A \circ B$  также будет дискретной. Для практических приложений этого, как правило, вполне достаточно.

### 3.5. Численные методы для операций с F-величинами

Основная сложность при работе с нечеткими величинами заключается в том, что даже в случае простейших исходных функций принадлежности в результате элементарных операций над ними получаются функции принадлежности сложной формы, требующие для своего описания большого числа параметров. Поэтому во многих работах [23,35,110,121] предлагается аппроксимировать исходные функции принадлежности и результаты операций над ними определенным классом функций, зависящих от фиксированного числа параметров: треугольных, экспоненциальных, трапециевидных и т.д. В этом случае имеется возможность построения сравнительно простых основных операций, не выводящих за пределы избранного класса функций [109].

**Нечеткое число LR-типа.** Нечеткое число называется нечетким числом LR-типа или LR-числом [2,91,105,110], если

$$\mu(x) = \begin{cases} L\left(\frac{m-x}{\alpha}\right), & x \leq m, \alpha > 0, \\ R\left(\frac{x-m}{\beta}\right), & x > m, \beta > 0 \end{cases}$$

где **L** и **R** - две непрерывные неотрицательные функции со следующими свойствами:

1.  $L(-x) = L(x)$ ,  $R(-x) = R(x)$ .
2.  $L(0) = 1$ ,  $R(0) = 1$ .
3. **L**, **R** не возрастают на  $[0, +\infty]$ .
4.  $\lim_{x \rightarrow \infty} L(x) = 0$ ,  $\lim_{x \rightarrow \infty} R(x) = 0$ .

Таким образом, нечеткое число LR-типа характеризуется видом функций **L** и **R**, а также тремя параметрами  $m, \alpha, \beta$  и может быть записано как  $\tilde{x} = (m, \alpha, \beta)_{LR}$ .

Для нечетких LR-чисел арифметические операции записываются гораздо проще [112]:

$$\begin{aligned} (m, \alpha, \beta)_{LR} + (n, \gamma, \delta)_{LR} &= (m+n, \alpha+\gamma, \beta+\delta)_{LR}; \\ -(m, \alpha, \beta)_{LR} &= (-m, \alpha, \beta)_{LR}; \\ (m, \alpha, \beta)_{LR} &= (m-n, \alpha+\delta, \beta+\gamma)_{LR}; \end{aligned}$$



$$\begin{aligned}
(m, \alpha, \beta)_{LR} \cdot (n, \gamma, \delta)_{LR} &= (mn, \alpha n + \gamma m, \beta n + \delta m)_{LR}, m > 0, n > 0; \\
(m, \alpha, \beta)_{LR} \cdot (n, \gamma, \delta)_{LR} &= (mn, \gamma m - \beta n, \delta m - \alpha n)_{LR}, m > 0, n < 0; \\
(m, \alpha, \beta)_{LR} \cdot (n, \gamma, \delta)_{LR} &= (mn, \alpha n - \delta m, \beta n - \gamma m)_{LR}, m < 0, n > 0; \\
(m, \alpha, \beta)_{LR} \cdot (n, \gamma, \delta)_{LR} &= (mn, -\beta n - \delta m, -\alpha n - \gamma m)_{LR}, m < 0, n < 0; \\
(m, \alpha, \beta)_{LR}^{-1} &= \left( \frac{1}{m}, \frac{\beta}{m^2}, \frac{\alpha}{m^2} \right)_{LR}, m > 0; \\
(m, \alpha, \beta)_{LR} : (n, \gamma, \delta)_{LR} &= \left( \frac{m}{n}, \frac{\delta m + \alpha n}{n^2}, \frac{\gamma m + \beta n}{n^2} \right)_{LR}, m > 0, n > 0.
\end{aligned}$$

Формулы для операций умножения и деления зависят от знаков операндов, при этом носители обоих операторов не должны содержать ноль.

Анализ влияния степени и вида нечеткости параметров для нескольких типов функций принадлежности для представления нечетких чисел на чувствительность целевой функции оптимизации проведен в работе [135]. Попытки создания универсальных высоко транспортабельных структурированных пакетов программ для выполнения интервальных вычислений на базе предпроцессора AUGMENT привели к резкому увеличению времени выполнения операций по сравнению с традиционными (неинтервальными) вычислениями на тех же примитивных объектах (числах с плавающей точкой фиксированной разрядности) в зависимости от различных условий в 50-200 раз [13]. Оригинальное расширение стандартного языка программирования для возможности работы с нечеткими величинами путем введения переменных типа FUZZY приведено в работе [23]. Ведутся также работы по созданию лингвистических терминальных комплексов, состоящих из универсальных микропроцессоров; ОЗУ, спецпроцессоров, ориентированных на выполнение операций над нечеткими множествами; ПЗУ-термов, предназначенных для хранения первичных терм-значений лингвистических переменных и других устройств [54,82].

Однако все эти методы используют аппроксимацию результирующих функций принадлежности определенным классом функций, что ведет к потере информации и увеличению области неопределенности.

### 3.6. Нечеткие отношения и нечеткие ограничения

Термин «отношение» используется для обозначения некоторых типов отображений, которые задаются на одном и том же универсуме  $X$ . В подобном случае отображение  $\Gamma: X \rightarrow X$  это отображение из множества  $X$  в себя, которое определяется парой  $\{X, \Gamma\}$ , где  $\Gamma \subseteq X^2$ .

Поскольку элементами множества  $X^2$  являются упорядочивающие пары, можно сказать, что отношение – это множество упорядочивающих пар. Т.к. каждая пара соединяет только 2 элемента множества  $X^2$  между собой. Такое отношение называется бинарным. Если элементами множества  $X^n$  являются упорядочивающие  $n$ -арные пары, такое отношение называется  $n$ -арным отношением. Частный случай – тернарное отношение – множество из упорядочивающих троек.

Понятие нечеткое отношение – это обобщение четких (как было отмечено выше) отношений в теории нечетких множеств. Оно может моделировать ситуации, где взаимодействия между элементами являются более или менее сильными.

Можно различать различные типы отношений. Например, отношение эквивалентности порядка, превосходства и т.д.

#### **Определение 3.16. Нечеткое отношение [1,2].**

Пусть  $X_1, X_2, \dots, X_n$  -  $n$ -универсумов.  $n$ -арное нечеткое отношение  $R$  в  $X_1, X_2, \dots, X_n$  является множеством на  $X_1, X_2, \dots, X_n$  и выражается как:

$$R_{X_1 \times \dots \times X_n} = \{((X_1, \dots, X_n), (X_1, \dots, X_n)) \mid (X_1, \dots, X_n) \in X_1 \times \dots \times X_n\}. \quad (3.54)$$

Как отмечалось выше, обычное отношение – частный случай нечетких отношений.

Носителем нечеткого отношения  $R$  на множестве  $X$  называют подмножество декартова произведения  $X \times X$  вида

$$\text{sup } pR = \{(u, v) \mid (u, v) \in X \times X, \mu_R(u, v) > 0\}.$$

Носитель нечеткого отношения следует понимать как отношение на множестве  $X$ , связывающее все пары  $(u, v)$ , для которых степень выполнения данного нечеткого отношения не равна нулю.

По аналогии с нечеткими множествами определяется и множество уровня нечеткого отношения, т.е.

$$R_\alpha = \{(u, v) / (u, v) \in X \times X, \mu_R(u, v) \geq \alpha\}.$$

Перейдем к рассмотрению операций над нечеткими отношениями, некоторые из которых являются аналогами операций над нечеткими множествами. А некоторые присущи только нечетким отношениям.

Перейдем к рассмотрению операций над нечеткими отношениями, некоторые из которых являются аналогами операций над нечеткими множествами, а некоторые присущи только нечетким отношениям.

Пересечением нечетких отношений  $P$  и  $Q$  на  $X \times X$  называют нечеткое отношение  $P \cap Q$  определяемое функцией принадлежности

$$\mu_{P \cap Q}(u, v) = \mu_P(u, v) \wedge \mu_Q(u, v) = \min\{\mu_P(u, v), \mu_Q(u, v)\}, \\ \forall u, v \in X \times X.$$

Объединением нечетких отношений  $P$  и  $Q$  на  $X \times X$  называют нечеткое бинарное отношение  $P \cup Q$  определяемое функцией принадлежности

$$\mu_{P \cup Q}(u, v) = \mu_P(u, v) \vee \mu_Q(u, v) = \max\{\mu_P(u, v), \mu_Q(u, v)\}, \\ \forall u, v \in X \times X.$$

Дополнением нечеткого отношения  $R \subseteq X \times X$  называют отношение  $\bar{R}$  с функцией принадлежности

$$\mu_{\bar{R}}(u, v) = 1 - \mu_R(u, v), \quad \forall u, v \in X \times X.$$

Обратным отношением к отношению  $R$  называют отношение  $R^{-1}$  с функцией принадлежности

$$\mu_{R^{-1}}(u, v) = \mu_R(v, u), \quad \forall u, v \in X \times X.$$

Очевидно, что матрица  $R^{-1}$  является транспонированной к матрице  $R$ .

### **Определение 3.17. Нечеткое ограничение.**

Пусть  $v = (v_1, \dots, v_n)$  - переменная на  $X = X_1 \times \dots \times X_n$ .

Нечеткое ограничение, обозначаемое  $R(v)$ , есть нечеткое отношение  $R$ , которое действует как гибкое ограничение на значения переменной  $v$ . В данном контексте переменная может рассматриваться как тройка  $(v, x, R(v))$ ;  $v$  - имя переменной.

Проекцией нечеткого отношения  $R$  на  $X_{i_1} \dots X_{i_k}$ ,  $(i_1, \dots, i_k)$  - подпоследовательность  $(1, 2, \dots, n)$ , является отношение на  $X_{i_1} \times \dots \times X_{i_k}$ , определяемое как

$$proj(R; X_{i_1}, \dots, X_{i_k}) = \int_{X_{i_1} \times \dots \times X_{i_k}} \sup_{X_{j_1}, \dots, X_{j_k}} \mu_R(X_1, \dots, X_n) / (X_{i_1}, \dots, X_{i_k}), \quad (3.55)$$

где  $(j_1, \dots, j_k)$  - подпоследовательность дополняющаяся до  $(i_1, \dots, i_k)$  в  $(1, \dots, n)$ .

Проекция называется также маргинальными нечеткими ограничениями. Обратно, если  $R$  – нечеткое множество в  $X_{i_1} \times \dots \times X_{i_k}$ , тогда цилиндрическое расширение в  $X_1 \times \dots \times X_n$  - это нечеткое множество  $C(R)$  на  $X_1 \times \dots \times X_n$ , определяемое посредством

$$C(R) = \int_{X_1 \times \dots \times X_n} \mu_R(X_{i_1}, \dots, X_{i_k}) / (X_1, \dots, X_n).$$

$n$ -арное нечеткое ограничение  $R(v_1, \dots, v_n)$  называется разделимым если и только если

$$R(v_1, \dots, v_n) = R(v_1) \times \dots \times R(v_n),$$

где  $\times$  обозначает картезианское произведение и  $R(v_i)$  - проекция  $R$  на  $X_i$ , т.е.

$$\mu_R(X_1, \dots, X_n) = \min_{i=1, n} \mu_{proj[R; X_i]}(X_i).$$

В терминах цилиндрического расширения эта формула может быть переписана как

$$R = \bigcap_{i=1, n} C(proj[R; X_i]).$$

$R$  разделимо, если и только если оно является соединением ее проекций. Если  $R$  разделимо, то все ее маргинальные нечеткие ограничения также разделимы. Переменные  $v_1, \dots, v_n$  называются невзаимодействующими, если их ограничение  $R(v_1, \dots, v_n)$  является разделимым нечетким ограничением.

### Пример.

Вычислим нечеткое отношение  $R = A \times B$ , если

$$A = 0,1/4 + 0,3/5 + 0,4/6,$$

$$B = 0,33/10 + 0,45/11 + 0,78/12.$$

Вместо **min** операции будем использовать **max** и **prod** операции

$$\mu_R = \max(\mu_A, \mu_B).$$

$$R = A \times B = 0,33/(4.10) + 0,45/(4.11) + 0,78/(4.12) + 0,33/(5.10) + 0,45/(5.11) + 0,78/(5.12) + 0,4/(6.10) + 0,45/(6.11) + 0,78/(6.12).$$

$$R = \begin{vmatrix} 0,33 & 0,45 & 0,78 \\ 0,33 & 0,45 & 0,78 \\ 0,4 & 0,45 & 0,78 \end{vmatrix}$$

$$\mu_R = (\mu_A \times \mu_B).$$

$$R = A \times B = 0,033/(4.10) + 0,045/(4.11) + 0,078/(4.12) + 0,099/(5.10) + 0,135/(5.11) + 0,234/(5.12) + 0,132/(6.10) + 0,180/(6.11) + 0,312/(6.12).$$

$$R = \begin{vmatrix} 0,033 & 0,045 & 0,078 \\ 0,099 & 0,135 & 0,234 \\ 0,132 & 0,180 & 0,312 \end{vmatrix}$$

### 3.7. Бинарные нечеткие отношения

Бинарное нечеткое отношение – это обобщение классического бинарного отношения.

Бинарное отношение  $R$  в  $X \times Y$  - это нечеткое множество в  $X \times Y$ . Пусть  $R$  – бинарное нечеткое отношение на  $X \times Y$ . Домен отношения  $R$ , обозначаемый  $dom(R)$ , и его ранг –  $ran(R)$  определяются, соответственно:

$$\mu_{dom(R)}(x) = \sup_y \mu_R(x, y), \forall x \in X,$$

$$\mu_{ran(R)}(y) = \sup_x \mu_R(x, y), \forall y \in Y.$$

**Определение 3.18.** Sup-Star композиция: Если  $R$  и  $S$  являются нечеткими отношениями в  $U \times V$  и  $V \times W$ , соответственно, композиция  $R$  и  $S$  является нечетким отношением, обозначаемым  $R \circ S$  и определяется как:

$$R \circ S = \{[(x, y), \sup_{y \in Y} (\mu_R(x, y) * \mu_S(y, z))] \mid x \in X, y \in Y, z \in Z\}, \quad (3.56)$$

где  $*$  может являться любым оператором из класса треугольных норм, а именно: минимум, алгебраическое произведение, ограниченное произведение или строгое (drastic) произведение [3].

Уравнение (4.3) может интерпретироваться следующим образом:  $\mu_{R \circ S}(x, z)$  - это сила множества цепей, соединяющих  $X$  с  $Z$ . Каждая цепь имеет форму  $x$ - $y$ - $z$ . Сила такой цепи равна силе наислабейшего соединения. Сила отношения между  $x$  и  $z$  является силой наисильнейшего соединения между  $x$  и  $z$ .

Пусть  $A$  – нечеткое множество в  $X$ . (3.56) можно переписать:

$$\mu_{A \circ R}(y) = \sup_x \min(\mu_A(x), \mu_R(x, y)).$$

Мы говорим, что  $B = A \circ R$  является нечетким множеством, индуцированным из  $A$  через  $R$ . Эта индукция обобщает хорошо известное четкое правило: если  $x=a$  и  $y=f(x)$ , то  $y=f(a)$ .

Имеем  $B = \text{proj}[C(A) \cap R; Y]$ .

Нечеткое отношение можно представить на конечном универсуме.

Когда связанные универсумы  $X$  и  $Y$  конечны, нечеткое отношение  $R$  на  $X * Y$  может быть представлено в виде матрицы  $[R]$ , терм которого  $[R]_{ij}$  является  $\mu_R(x_i, y_j) = r_{ij}$ ,  $i = \overline{1, n}$ ;  $j = \overline{1, m}$ , где  $|X| = n$ ,  $|Y| = m$ .

Композиция конечных нечетких отношений с учетом

$$[S]_{jk} = S_{jk}, \quad k = \overline{1, p}; \quad P = |Z|,$$

может рассматриваться как матричное произведение

$$[ROS]_{ik} = \sum_j r_{ij} S_{jk},$$

где суммирование выполняется операцией  $\max$ , а умножение операцией  $\min$ .

Так как  $R \circ S$  может быть записано в виде:

$$\text{proj}[C(R) \cap C(S); X \times Z],$$

где  $R$  и  $S$  заданы на  $X \times Y$  и  $Y \times Z$ , соответственно, то другие композиции могут быть введены посредством модификации оператора, использованного для пересечения.

Изменяя  $\min$  на  $*$ , мы вводим  $R * S$  через

$$\mu_{R * S}(x, z) = \sup_y (\mu_R(x, y) * \mu_S(y, z)).$$

Мы можем встретить иной тип альтернативных композиций,  $\inf\text{-max}$ ,  $\text{sup-prod}$  и другие.

## Нечеткое отношение ЕСЛИ-ТО

Пусть  $A$  и  $B$  – нечеткие подмножества на универсумах  $X$  и  $Y$ .

Для связи нечетких подмножеств  $A$  и  $B$ , заданных на разных областях рассуждений  $X$  и  $Y$ , вводится понятие нечеткого условного утверждения (лингвистической импликации), т.е.

$$A \rightarrow B \quad \text{при «ЕСЛИ } A \text{ ТО } B\text{»}.$$

Полученное импликацией отношение  $R$  выражается в терминах картезианского произведения подмножеств  $A$  и  $B$ , обозначается как  $R = A \times B$ , и его функция принадлежности определяется как

$$\mu_R(x, y) = \mu_{A \times B}(x, y) = \min[\mu_A(x), \mu_B(y)], \quad x \in X, y \in Y \quad (3.57)$$

**Пример**

1. Задана нечеткая импликация: если  $A$  то  $B$ . Вычислить нечеткое отношение,  $R = A \times B$  с использованием  $\min$  операции композиции, если

$$a) A = 0.1/20 + 0.3/21 + 0.4/22,$$

$$B = 0.33/60 + 0.45/65 + 0.78/70;$$

$$R = A \times B = 0.1/(20,60) + 0.1/(20,65) + 0.1/(20,70) + 0.3/(21,60) + 0.3/(21,65) + \\ + 0.3/(21,70) + 0.33/(22,60) + 0.4/(22,65) + 0.4/(22,70).$$

Может встретиться также вложенное нечеткое отношение. В таких случаях нечеткое условное утверждение является вложенным, т.е. имеет форму ЕСЛИ  $A$  ТО ЕСЛИ  $B$  ТО  $C$ . Тогда нечеткое отношение  $R$  записывается в виде:

$$R = A \times (B \times C) = A \times B \times C \quad (3.58)$$

Нечеткая импликация может состоять из двух импликаций. Эти простые импликации соединяются с использованием соединителей «или» («иначе»), «и» и т.д.

**Пример.**

Пусть дана импликация

$$\text{ЕСЛИ } A_1 \text{ ТО } B_1$$

или (иначе)

$$\text{ЕСЛИ } A_2 \text{ ТО } B_2$$

где  $A_1, A_2$  – нечеткие подмножества на  $X$ , а  $B_1, B_2$  – нечеткие подмножества на  $Y$ .

Результирующее нечеткое отношение  $R$  вычисляется как объединение индивидуальных нечетких отношений  $R_i$  ( $i = 1, 2$ ):

$$R = \bigcup_{i=1,2} R_i = \bigcup_{i=1,2} A_i \times B_i . \quad (3.59)$$

Функция принадлежности  $R$  определяется как

$$\mu_R(x, y) = \max_x \{ \min[ \mu_{A_1}(x), \mu_{B_1}(y) ], \min[ \mu_{A_2}(x), \mu_{B_2}(y) ] \} . \quad (3.60)$$

Это может быть расширено на случай, когда имеем дело не с одной, а более чем двумя импликациями.

Пусть заданы нечеткое отношение  $R = X \times Y$  и значение  $A'$  нечеткого подмножества  $A$ . Отношение используется для вывода соответствующего значения  $B'$  посредством применения композиционного правила вывода, которое записывается как

$$B' = A' \circ R = A' \circ (A \times B) .$$

Функция принадлежности определяется как

$$\mu_{B'}(y) = \max_x \min[ \mu_{A'}(x), \mu_R(x, y) ] .$$

В случае тернарного нечеткого отношения формулы имеют вид:

$$C' = A' \circ (B' \circ R) = A' \circ (B' \circ (A \times B \times C)) ,$$

$$\mu_{C'}(z) = \max_x \min \left[ \mu_{A'}(x), \max_y \min[ \mu_B(y), \mu_R(x, y, z) ] \right] . \quad (3.61)$$

### Пример.

Задана нечеткая импликация ЕСЛИ  $A$  ТО ЕСЛИ  $B$  ТО  $C$ .

Вычислим нечеткое отношение  $R = A \times B \times C$ .

$$A = 0.3/5 + 0.5/6 + 0.8/7 ,$$

$$B = 0.3/15 + 0.5/16 + 0.8/17 ,$$

$$C = 0.2/25 + 0.4/26 + 0.6/27 ;$$

$$R = A \times B \times C =$$

$$= 0.2/(5, 15, 25) + 0.3/(5, 15, 26) + 0.3/(5, 15, 27) +$$

$$+ 0.2/(5, 16, 25) + 0.3/(5, 16, 26) + 0.3/(5, 16, 27) +$$

$$+ 0.2/(5, 17, 25) + 0.3/(5, 17, 26) + 0.3/(5, 17, 27) +$$

$$+ 0.2/(6, 15, 25) + 0.4/(6, 15, 26) + 0.3/(6, 15, 27) +$$

$$+ 0.2/(6, 16, 25) + 0.4/(6, 16, 26) + 0.5/(6, 16, 27) +$$

$$+ 0.2/(6, 17, 25) + 0.4/(6, 17, 26) + 0.5/(6, 17, 27) +$$

$$+ 0.2/(7, 15, 25) + 0.3/(7, 15, 26) + 0.3/(7, 15, 27) +$$

$$+ 0.2/(7, 16, 25) + 0.4/(7, 16, 26) + 0.6/(7, 16, 27) +$$



$$+0.2/(7,17,25)+0.4/(7,17,26)+0.56/(7,17,27).$$

### Определение 3.19. Нечеткий граф.

С понятием нечеткого отношения вплотную связано понятие нечеткого графа. Допустим, что  $E$  обычное множество узлов. Нечеткий граф определяется как [3]

$$G(X_i, X_j) = \{((X_i, X_j), \mu_G(X_i, X_j)) / (X_i, X_j) \in E \times E\}.$$

Если  $E$  – нечеткое множество, то нечеткий граф будет определяться аналогично нечетким отношениям.

#### Пример.

$E = \{X_1, X_2, X_3, X_4\}$ . Тогда нечеткий граф может быть представлен как

$$G(X_i, X_j) = \{[(X_1, X_2), 0.3], [(X_1, X_3), 0.6], [(X_1, X_1), 1], [(X_2, X_1), 0.4], [(X_3, X_1), 0.2], [(X_3, X_2), 0.5], [(X_4, X_3), 0.8]\}.$$

## 3.8. Нечеткая логика

Согласно одному из самых распространенных определений, логика есть анализ методов рассуждений. Изучая эти методы, логика интересуется, в первую очередь, формой, а не содержанием доводов в том или ином рассуждении. Истинность или ложность отдельных посылок или заключений не интересует логику. Ее интересует лишь, вытекает ли истинность заключения из истинности посылок. Систематическая формализация и каталогизация правильных способов рассуждений – одна из основных задач логики.

В логике из простых высказываний, путем соединения их различными способами, можно составить новые, более сложные высказывания. В дальнейшем мы будем рассматривать одни только истинностно-функциональные комбинации, в которых истинность или ложность новых высказываний определяется истинностью или ложностью составляющих высказываний.

Одной из простейших операций над высказываниями является отрицание. Например, если  $A$  есть высказывание, то отрицание  $A$  обозначается  $\neg A$  и читается «не  $A$ ».

Другой истинностно-функциональной операцией над высказываниями является конъюнкция  $\text{И}$ , которая обозначается  $A \& B$ . Высказывание истинно тогда и только тогда, когда истинны

оба высказывания  $A$  и  $B$ . Высказывание  $A$  и  $B$  называют конъюнктивными членами конъюнкции  $A \& B$ .

Операция дизъюнкции над высказываниями  $A$  и  $B$  соответствует связке ИЛИ и обозначается  $A \vee B$ . В обычном языке связка ИЛИ употребляется в двух различных смыслах: разделительном и соединительном. В операции дизъюнкции связка ИЛИ имеет соединительный смысл.

Следующей важной истинностно-функциональной операцией является следование: ЕСЛИ  $A$  ТО  $B$ . Это высказывание ложно, когда посылка  $A$  истинна, а заключение  $B$  ложно. Обозначение высказывания ЕСЛИ  $A$  ТО  $B$  следующее:  $A \supset B$ . Это выражение называют импликацией.

Выражение « $A$  тогда и только тогда, когда  $B$ » обычно обозначается через  $A \equiv B$ . Такое выражение называют эквивалентностью. Очевидно  $A \equiv B$  истинно тогда и только тогда, когда  $A$  и  $B$  имеют одно и то же истинностное значение.

Ниже приведена таблица истинности для всех этих операций над высказываниями:

$A$	$B$	$\neg A$	$A \& B$	$A \vee B$	$A \supset B$	$A \equiv B$
И	И	Л	И	И	И	И
Л	И	И	Л	И	И	Л
И	Л	Л	Л	И	Л	Л
Л	Л	И	Л	Л	И	И

Символы  $\neg, \&, \vee, \supset, \equiv$  называют пропозициональными связками. Всякое высказывание, построенное при помощи этих связок, имеет некоторое истинностное значение, зависящее от истинности значений составляющих высказываний. Пропозициональной формой называют выражение, построенное из пропозициональных букв  $A, B, C$  и т.д., с помощью пропозициональных связок.

Всякая пропозициональная форма определяет некоторую истинностную функцию, которая графически может быть представлена истинностной таблицей для этой пропозициональной формы. Истинностной функцией от  $n$  аргументов называют всякую функцию от  $n$  аргументов, принимающую истинностные значения И (истина) и Л (ложь), если аргументы ее пробегают те же значения.

Пропозициональную форму, которая истинна независимо от того, какие значения принимают встречающиеся в ней пропозициональные буквы, называют тавтологией. Пропозициональная форма является тавтологией тогда и только тогда, когда соответствующая истинностная функция принимает только значение И.

Например, следующие предложения являются пропозициональными тавтологиями:

- 1)  $\neg(A \& \neg A)$  - закон отрицания противоречия;
- 2)  $((A \vee B) \rightarrow (\neg A \rightarrow B))$  и  $(\neg B \rightarrow A)$  - выражение дизъюнкции через отрицание и импликацию;
- 3)  $((A \rightarrow B) \& A) \rightarrow B$ .

В свою очередь, пропозициональную форму, которая ложна при всех возможных истинностных значениях ее пропозициональных букв, называют противоречием, например пропозициональная форма  $(A \equiv \neg A)$  или  $(A \& (\neg A))$ .

Следует отметить, что импликация имеет следующее важное свойство, называемое правилом отделения (*modus—ponens*): ЕСЛИ  $(A \rightarrow B)$  истинно и  $A$  истинно ТО  $B$  истинно. Иначе это правило называют первой формой гипотетического силлогизма. Под силлогизмом подразумевается дедуктивное умозаключение, в котором одно суждение является необходимым двух других. Это свойство, как указывалось выше, играют важную роль при моделировании сложных технологических процессов.

До сих пор рассматривали бинарную (булеву) логику. Логика как основы рассуждений отличаются тремя составляющими: значением истинности; операторами; процедурами вывода.

**Определение 3.20. Нечеткая логика.** Нечеткая логика есть расширение теоретико-множественной многозначной логики, в которой значение истинности является лингвистической переменной.

Так как операторы типа  $\vee, \neg, \wedge, \Rightarrow$  в нечеткой логике тоже определяются с использованием таблиц истинности, то с применением принципа расширения Заде выводятся эти операторы.

Пространством истинности в этой логике является действительный интервал  $[0,1]$ . Эта логика, называемая многозначной или нечеткой, основывается на теории нечетких

множеств. Определим семантическую функцию истинности этой многозначной логики. Пусть  $P$  является высказыванием, а  $\nu(P)$  - его значением истинности, при этом  $\nu(P) \in [0,1]$ .

Значение отрицания для высказывания  $P$  определяется так:  $\nu(\neg P) = 1 - \nu(P)$ . Следовательно,  $\nu(\neg\neg P) = \nu(P)$ .

Связка-импликация  $\rightarrow$  всегда определяется следующим образом:

$$\nu(P \rightarrow Q) = \nu(\neg P \vee Q)$$

а эквивалентность

$$\nu(P \leftrightarrow Q) = \nu[P \rightarrow Q \wedge (Q \rightarrow P)].$$

Отметим, что разделительная дизъюнкция  $\text{ex}$ , дизъюнкция отрицаний или связка Шеффера  $/$ , конъюнкция отрицаний  $\downarrow$  и  $\rightarrow$  (не имеют общего названия) определяются как отрицание эквивалентности  $\leftrightarrow$ , конъюнкции  $\wedge$ , дизъюнкции  $\vee$  и импликации  $\rightarrow$ , соответственно. Тавтология и противоречие, соответственно, будут:

$$\nu(\dot{P}) = \nu(P \vee \neg P); \quad \nu(\dot{P}) = \nu(P \wedge \neg P).$$

Более обобщенно:

$$\nu(\dot{P}Q) = \nu((P \vee \neg P) \vee (Q \vee \neg Q)); \quad \nu(PQ) = \nu((P \wedge \neg P) \wedge (Q \wedge \neg Q)).$$

Определим основные связки нечеткой логики в двух наиболее часто встречающихся теориях нечетких множеств.

**Логика, основанная на  $(\mathfrak{R}(x), \cup, \cap, -)$ .** В этом случае дизъюнкцию и конъюнкцию определяют так:

$$\nu(P \vee Q) = \max(\nu(P), \nu(Q)); \quad \nu(P \wedge Q) = \min(\nu(P), \nu(Q)).$$

Ясно, что  $\wedge$  и  $\vee$  - коммутативные, ассоциативные, идемпотентные, дистрибутивные и не удовлетворяют закону исключения третьего, т.е.  $\nu(P \vee \neg P) \neq 1$  и  $\nu(P \wedge \neg P) \neq 0$ , но удовлетворяют закону поглощения

$$(\nu((P) \vee (P \wedge Q))) = \nu(P); \quad (\nu((P) \wedge (P \vee Q))) = \nu(P),$$

а также законам Де-Моргана

$$\nu(\neg(P \wedge Q)) = \nu(\neg P \vee \neg Q); \quad \nu(\neg(P \vee Q)) = \nu(\neg P \wedge \neg Q).$$

Закон исключения дизъюнкции

$$\nu[(\neg P \wedge Q) \vee (P \wedge \neg Q)] = \nu[(P \vee Q) \wedge (\neg P \vee \neg Q)].$$

Ниже приведены выражения для 16 связок:

PQ	$\dot{P}Q$	$P \vee Q$	$Q \rightarrow P$	P
pq	$\max(p, 1-p, q, 1-q)$	$\max(p, q)$	$\max(p, 1-q)$	p
PQ	$P \rightarrow Q$	Q	$P \leftrightarrow Q$	$P \wedge Q$
pq	$\max(1-p, q)$	q	$\min(\max(1-p, q), \max(p, 1-q))$	$\min(p, q)$
PQ	P/Q	$P \text{ ex } Q$	$\neg Q$	$Q \leftrightarrow P$
pq	$\max(1-p, 1-q)$	$\max(\min(1-p, q), pQ \min(p, 1-q))$	1-q	$\min(p, 1-q)$
PQ	$\neg P$	$P \approx \rightarrow Q$	$P \downarrow Q$	$\dot{P}Q$
pq	1-p	$\min(1-p, q)$	$\min(1-p, 1-q)$	$\min(p, 1-p, q, 1-q)$

Здесь полагается, что  $\vee(P) = p$  и  $\vee(Q) = q$ .

Кванторы в высказываниях будут:

$$\vee(\exists \times P(x) = \sup(\vee P(x))) ; \quad \vee(\forall \times P(x) = \inf(\vee P(x))) ,$$

где  $x$  – элемент области рассуждения.

**Многозначную логику, основанную на  $\mathfrak{H}(x), \cup, \cap, \neg$**  обычно называют К-стандартной последовательной логикой. В этой логике связки удовлетворяют следующим свойствам:

$$\text{Импликация } \vee[P \rightarrow (Q \rightarrow R)] = \vee[(P \wedge Q) \rightarrow R];$$

тавтология и противоречие

$$\vee(P \rightarrow P) = \vee(\dot{P}); \quad \vee(\dot{P} \rightarrow P) = \vee(P); \quad \vee(P \rightarrow \dot{P}) = \vee(\dot{P});$$

$$\vee(P \leftrightarrow P) = \vee(\dot{P}); \quad \vee(\dot{P} \rightarrow P) = \vee(\dot{P}); \quad \vee(P \rightarrow \dot{P}) = \vee(\neg P);$$

$$\vee(P \leftrightarrow \neg P) = \vee(\dot{P});$$

связки Шеффера и Пирса

$$\vee(\neg P) = \vee(P/P); \quad \vee(P \rightarrow Q) = \vee(P/(Q/Q)); \quad \vee(\dot{P}) = \vee(P/(P/P)).$$

В [4] показано, что многозначная логика является размытием (в смысле нечеткости) стандартного исчисления высказываний (в смысле принципа расширения). В этой логике каждому высказыванию  $P$  ставится в соответствие нормализованное, нечеткое множество в  $[0, 1]$ , т.е. пара  $\{\mu_p(0), \mu_p(1)\}$  интерпретируется как степень ложности и степень истинности, соответственно. Так как логические связки стандартного исчисления высказываний являются функционалами истинности, т.е. представляются в виде

функций, то их можно размыть. Следует отметить, что впервые данная логика была предложена независимо Клином и Дайнесом.

**Логика, основанная на  $(\tilde{\mathfrak{N}}(x), \dot{\cup}, \cap, -)$ .** В этом случае дизъюнкция и конъюнкция определяются так:

$$\vee(P \dot{\vee} Q) = \min(1, \vee(P) + \vee(Q)); \quad \vee(P \wedge Q) = \max(0, \vee(P) + \vee(Q) - 1).$$

Ясно, что  $\dot{\vee}$  и  $\wedge$  - коммутативные, ассоциативные, неидемпотентные, недистрибутивные и удовлетворяют закону

$$\vee(\neg(P \wedge Q)) = \vee(\neg P \dot{\vee} \neg Q);$$

закону Де-Моргана

$$\vee(\neg(P \dot{\vee} Q)) = \vee(\neg P \wedge \neg Q),$$

а также закону исключения третьего

$$\vee(P \dot{\vee} \neg P) = 1, \quad \vee(P \wedge \neg P) = 0.$$

Ниже заданы оценки 16 связок:

PQ	$P \cdot Q$	$P \dot{\vee} Q$	$Q \Rightarrow P$	P
pq	1	$\min(1, 1+q)$	$\min(1, p+1-q)$	p
PQ	$P \Rightarrow Q$	Q	$P \leftrightarrow Q$	$P \wedge Q$
pq	$\min(1, 1-p+q)$	q	$1 -  p - q $	$\max(0, p+q-1)$
PQ	$P \parallel Q$	$P \text{ ex } Q$	$\neg Q$	$Q \approx \Rightarrow P$
pq	$\min(1, 1-p+1-q)$	$ p - q $	$1 - q$	$\max(0, p - q)$
PQ	$\neg P$	$P \approx \Rightarrow Q$	$P \downarrow \downarrow Q$	$\dot{P}Q$
pq	$1 - p$	$\max(0, q - p)$	$\max(0, 1 - p - q)$	0

Здесь  $\vee, \rightarrow, \leftrightarrow, \wedge, /, \text{ex}, \Rightarrow, \downarrow$  обозначаются, соответственно через  $\dot{\vee}, \Rightarrow, \leftrightarrow, \wedge, \parallel, \text{ex}, \approx \Rightarrow, \downarrow \downarrow$ . Тавтология и противоречие удовлетворяют следующим свойствам:

$$\begin{aligned} \vee(P \Rightarrow P) &= \vee(\dot{P}), & \vee(\dot{P} \Rightarrow P) &= \vee(P), & \vee(P \Rightarrow \dot{P}) &= \vee(\dot{P}), \\ \vee(P \leftrightarrow P) &= \vee(\dot{P}), & \vee(\dot{P} \Rightarrow P) &= \vee(\dot{P}), & \vee(P \leftrightarrow \dot{P}) &= \vee(\neg P). \end{aligned}$$

В обозначениях Заде импликация  $\Rightarrow$  соответствует обычному включению нечетких множеств, ex и  $\approx \Rightarrow$  соответствует

симметрической  $\Delta$  и ограниченной  $\parallel$  разностям. Данная логика была известна под названием логики Лукасевича ( $L$ ).

Следует отметить, что две теории нечетких множеств и построенные на их основе логики не являются единственными известными на сегодняшний день. В этой связи целесообразно дать семантический анализ основных известных многозначных логик и начать его следует с изложения сведений из теории нечетких «сильных» множеств необходимых для формализации некоторых операций над нечеткими множествами, которые необходимы для проведения семантического анализа таких логик.

Пусть  $A$  и  $B$  являются нечеткими подмножествами четкого универсума  $U$ ; стало традиционным считать в теории нечетких множеств, что  $A$  является подмножеством  $B$  тогда и только тогда, когда

$$\mu_A \leq \mu_B, \text{ т.е. } \forall x \in U, \mu_A x \leq \mu_B x.$$

**Определение 3.21. Сильное нечеткое множество.** Если дан [5] нечеткий оператор импликации  $\rightarrow$  и нечеткое множество  $B$  из универсума  $U$ , то нечеткое «сильное» множество  $\mathfrak{R}B$  из  $B$  задается функцией принадлежности  $\mu_{\mathfrak{R}B}$  вида

$$\mu_{\mathfrak{R}B} A = \bigwedge_{x \in U} (\mu_A x \rightarrow \mu_B x).$$

Тогда степень, с которой  $A$  является подмножеством  $B$ , есть

$$\pi(A \subseteq B) = \mu_{\mathfrak{R}B} A.$$

**Определение 22. Оператор импликации.** Если дан [5] нечеткий оператор импликации  $\rightarrow$  на замкнутом единичном интервале  $[0,1]$ , то

$$a \leftarrow b, \quad b \rightarrow a, \\ a \leftrightarrow b = (a \rightarrow b) \wedge (a \leftarrow b) = (a \rightarrow b) \wedge (b \rightarrow a).$$

**Определение 3.23. Эквивалентность нечетких множеств.** При условиях определения 3.21 степень, с которой нечеткие множества  $A$  и  $B$  эквивалентны, или степень их «эквивалентности» имеет вид

$$\pi(A \equiv B) = \pi(A \subseteq B) \wedge \pi(A \supseteq B); \\ \pi(A \equiv B) = \bigwedge_{x \in U} (\mu_A x \leftrightarrow \mu_B x).$$

### 3.9. Нечеткие импликации

В [6] было показано, что для практических целей, в большинстве случаев, целесообразно работать с многозначными логиками, в которых логические переменные принимают значения из действительного интервала  $I=[0,1]$ , разбивая его на 10 подинтервалов, т.е. используя множество  $\vee_{10} [0,0.1,0.2,\dots,1]$ .

Операции импликации в анализируемых логиках, представленные в [6], имеют следующий вид:

1)  $s^\#$  - логика:

$$a \xrightarrow{s^\#} b = \begin{cases} 1, & \text{если } a \neq 1 \text{ или } b = 1; \\ 0, & \text{в противном случае;} \end{cases}$$

2) S-логика («стандартная последовательность»):

$$a \xrightarrow{s} b = \begin{cases} 1, & \text{если } a \leq b; \\ 0, & \text{в противном случае;} \end{cases}$$

3) G-логика («Геделинанская последовательность»):

$$a \xrightarrow{g} b = \begin{cases} 1, & \text{если } a \leq b; \\ b, & \text{в противном случае;} \end{cases}$$

4) G43-логика:

$$a \xrightarrow{G43} b = \min(1, b/a) \text{ при } a > 0, \quad a \xrightarrow{G43} b = 1;$$

5) L-логика, или уже обсуждавшаяся логика Лукасевича:

$$a \xrightarrow{L} b = \min(1, 1 - a + b);$$

6) KD-логика, показанная выше:

$$a \xrightarrow{KD} b = (1 - a) \vee b = \max(1 - a, b).$$

В свою очередь, ALI 1 – ALI 3 – логики, которые будут использованы в дальнейших главах данной книги, характеризуются следующими операциями импликации [2]:

7) ALI 1-логика:

$$a \xrightarrow{ALI1} b = \begin{cases} 1 - a, & \text{если } a < b; \\ 1, & \text{если } a = b; \\ b, & \text{если } a > b; \end{cases}$$

8) ALI 2 – логика:

$$a \xrightarrow{ALI2} b = \begin{cases} 1, & \text{если } a \leq b; \\ (1 - a) \wedge b, & \text{если } a > b; \end{cases}$$

9) ALI 3 - логика:



$$a \xrightarrow{ALIZ} b = \begin{cases} 1, \text{ если } a \leq b; \\ b/[a + (1-b)], \text{ в противном случае.} \end{cases}$$

Операции импликации удобно изображать в виде таблиц импликативных переходов для одиннадцатизначных логик. Такая таблица для S# - логики имеет следующий вид (по горизонтали над чертой откладываются значения выходной логической посылки, по вертикали – значения истинности входной):

$\bar{s}^\#$	0	0,1	0,2	0,3	0,4	0,5	0,6	0,7	0,8	0,9	1
0	1	1	1	1	1	1	1	1	1	1	1
0,1	1	1	1	1	1	1	1	1	1	1	1
0,2	1	1	1	1	1	1	1	1	1	1	1
0,3	1	1	1	1	1	1	1	1	1	1	1
0,4	1	1	1	1	1	1	1	1	1	1	1
0,5	1	1	1	1	1	1	1	1	1	1	1
0,6	1	1	1	1	1	1	1	1	1	1	1
0,7	1	1	1	1	1	1	1	1	1	1	1
0,8	1	1	1	1	1	1	1	1	1	1	1
0,9	1	1	1	1	1	1	1	1	1	1	1
1	0	0	0	0	0	0	0	0	0	0	0

$\bar{s}$	0	0,1	0,2	0,3	0,4	0,5	0,6	0,7	0,8	0,9	1
0	1	1	1	1	1	1	1	1	1	1	1
0,1	0	1	1	1	1	1	1	1	1	1	1
0,2	0	0	1	1	1	1	1	1	1	1	1
0,3	0	0	0	1	1	1	1	1	1	1	1
0,4	0	0	0	0	1	1	1	1	1	1	1
0,5	0	0	0	0	0	1	1	1	1	1	1
0,6	0	0	0	0	0	0	1	1	1	1	1
0,7	0	0	0	0	0	0	0	1	1	1	1
0,8	0	0	0	0	0	0	0	0	1	1	1
0,9	0	0	0	0	0	0	0	0	0	1	1
1	0	0	0	0	0	0	0	0	0	0	1

$\bar{G}$	0	0,1	0,2	0,3	0,4	0,5	0,6	0,7	0,8	0,9	1
0	1	1	1	1	1	1	1	1	1	1	1
0,1	0	1	1	1	1	1	1	1	1	1	1
0,2	0	0,1	1	1	1	1	1	1	1	1	1
0,3	0	0,1	0,2	1	1	1	1	1	1	1	1
0,4	0	0,1	0,2	0,3	1	1	1	1	1	1	1
0,5	0	0,1	0,2	0,3	0,4	1	1	1	1	1	1
0,6	0	0,1	0,2	0,3	0,4	0,5	1	1	1	1	1
0,7	0	0,1	0,2	0,3	0,4	0,5	0,6	1	1	1	1
0,8	0	0,1	0,2	0,3	0,4	0,5	0,6	0,7	1	1	1
0,9	0	0,1	0,2	0,3	0,4	0,5	0,6	0,7	0,8	1	1
1	0	0,1	0,2	0,3	0,4	0,5	0,6	0,7	0,8	0,9	1

$\bar{G}_{43}$	0	0,1	0,2	0,3	0,4	0,5	0,6	0,7	0,8	0,9	1
0	0	1	1	1	1	1	1	1	1	1	1
0,1	0	1	1	1	1	1	1	1	1	1	1
0,2	0	0,5	1	1	1	1	1	1	1	1	1
0,3	0	0,33	0,67	1	1	1	1	1	1	1	1
0,4	0	0,25	0,5	0,75	1	1	1	1	1	1	1
0,5	0	0,21	0,4	0,6	0,8	1	1	1	1	1	1
0,6	0	0,17	0,33	0,5	0,67	0,83	1	1	1	1	1
0,7	0	0,14	0,29	0,43	0,57	0,71	0,86	1	1	1	1
0,8	0	0,13	0,25	0,38	0,5	0,63	0,75	0,88	1	1	1
0,9	0	0,11	0,22	0,33	0,44	0,56	0,67	0,78	0,89	1	1
1	0	0,1	0,2	0,3	0,4	0,5	0,6	0,7	0,8	0,9	1

$\bar{L}$	0	0,1	0,2	0,3	0,4	0,5	0,6	0,7	0,8	0,9	1
0	1	1	1	1	1	1	1	1	1	1	1
0,1	0,9	1	1	1	1	1	1	1	1	1	1
0,2	0,8	0,9	1	1	1	1	1	1	1	1	1
0,3	0,7	0,8	0,9	1	1	1	1	1	1	1	1
0,4	0,6	0,7	0,8	0,9	1	1	1	1	1	1	1
0,5	0,5	0,6	0,7	0,8	0,9	1	1	1	1	1	1
0,6	0,4	0,5	0,6	0,7	0,8	0,9	1	1	1	1	1
0,7	0,3	0,4	0,5	0,6	0,7	0,8	0,9	1	1	1	1
0,8	0,2	0,3	0,4	0,5	0,6	0,7	0,8	0,9	1	1	1
0,9	0,1	0,2	0,3	0,4	0,5	0,6	0,7	0,8	0,9	1	1
1	0	0,1	0,2	0,3	0,4	0,5	0,6	0,7	0,8	0,9	1

$\overline{KD}$	0	0,1	0,2	0,3	0,4	0,5	0,6	0,7	0,8	0,9	1
0	1	1	1	1	1	1	1	1	1	1	1
0,1	0,9	0,9	0,9	0,9	0,9	0,9	0,9	0,9	0,9	0,9	1
0,2	0,8	0,8	0,8	0,8	0,8	0,8	0,8	0,8	0,8	0,9	1
0,3	0,7	0,7	0,7	0,7	0,7	0,7	0,7	0,7	0,8	0,9	1
0,4	0,6	0,6	0,6	0,6	0,6	0,6	0,6	0,7	0,8	0,9	1
0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,6	0,7	0,8	0,9	1
0,6	0,4	0,4	0,4	0,4	0,4	0,5	0,6	0,7	0,8	0,9	1
0,7	0,3	0,3	0,3	0,3	0,4	0,5	0,6	0,7	0,8	0,9	1
0,8	0,2	0,2	0,2	0,3	0,4	0,5	0,6	0,7	0,8	0,9	1
0,9	0,1	0,1	0,2	0,3	0,4	0,5	0,6	0,7	0,8	0,9	1
1	0	0,1	0,2	0,3	0,4	0,5	0,6	0,7	0,8	0,9	1

$\overrightarrow{ALI1}$	0	0,1	0,2	0,3	0,4	0,5	0,6	0,7	0,8	0,9	1
0	1	1	1	1	1	1	1	1	1	1	1
0,1	0	1	0,9	0,9	0,9	0,9	0,9	0,9	0,9	0,9	0,9
0,2	0	0,1	1	0,8	0,8	0,8	0,8	0,8	0,8	0,8	0,8
0,3	0	0,1	0,2	1	0,7	0,7	0,7	0,7	0,7	0,7	0,7
0,4	0	0,1	0,2	0,3	1	0,6	0,6	0,6	0,6	0,6	0,6
0,5	0	0,1	0,2	0,3	0,4	1	0,5	0,5	0,5	0,5	0,5
0,6	0	0,1	0,2	0,3	0,4	0,5	1	0,4	0,4	0,4	0,4
0,7	0	0,1	0,2	0,3	0,4	0,5	0,6	1	0,3	0,3	0,3
0,8	0	0,1	0,2	0,3	0,4	0,5	0,6	0,7	1	0,2	0,2
0,9	0	0,1	0,2	0,3	0,4	0,5	0,6	0,7	0,8	1	0,1
1	0	0,1	0,2	0,3	0,4	0,5	0,6	0,7	0,8	0,9	1

$\overrightarrow{ALI2}$	0	0,1	0,2	0,3	0,4	0,5	0,6	0,7	0,8	0,9	1
0	1	1	1	1	1	1	1	1	1	1	1
0,1	0	1	1	1	1	1	1	1	1	1	1
0,2	0	0,1	1	1	1	1	1	1	1	1	1
0,3	0	0,1	0,2	1	1	1	1	1	1	1	1
0,4	0	0,1	0,2	0,3	1	1	1	1	1	1	1
0,5	0	0,1	0,2	0,3	0,4	1	1	1	1	1	1
0,6	0	0,1	0,2	0,3	0,3	0,3	1	1	1	1	1
0,7	0	0,1	0,2	0,3	0,4	0,5	0,6	1	1	1	1
0,8	0	0,1	0,2	0,2	0,2	0,2	0,2	0,2	1	1	1
0,9	0	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1	1	1
1	0	0	0	0	0	0	0	0	0	0	0

$\overline{ALI3}$	0	0,1	0,2	0,3	0,4	0,5	0,6	0,7	0,8	0,9	1
0	1	1	1	1	1	1	1	1	1	1	1
0,1	0	1	1	1	1	1	1	1	1	1	1
0,2	0	1/11	1	1	1	1	1	1	1	1	1
0,3	0	1/12	2/11	1	1	1	1	1	1	1	1
0,4	0	1/13	1/6	3/11	1	1	1	1	1	1	1
0,5	0	1/14	2/13	1/4	4/11	1	1	1	1	1	1
0,6	0	1/15	1/7	3/13	1/3	5/11	1	1	1	1	1
0,7	0	1/16	2/15	3/14	4/13	5/12	6/11	1	1	1	1
0,8	0	1/17	1/8	1/5	2/7	5/13	1/2	7/11	1	1	1
0,9	0	1/18	2/17	3/16	4/15	5/14	6/13	7/12	8/11	1	1
1	0	1/19	1/9	3/17	1/4	1/3	3/7	7/13	2/3	9/11	1

Запишем аналитические выражения определения операции двойной импликации для предложенных нечетких логик ALI1-ALI3

$$a \leftrightarrow b = \begin{cases} 1 - ka, & \text{если } a < b; \\ 1, & \text{если } a = b \\ 1 - kb, & \text{если } a > b; \end{cases}$$

ALI1

$$a \leftrightarrow b = \begin{cases} 1, & \text{если } a = b; \\ \min[(\neg a \wedge b), (\neg b \wedge a)], & \text{в остальных случаях}; \\ 0, & \text{если } (ka = 1) \vee (kb = 1); \end{cases}$$

ALI2

$$a \rightarrow b = \begin{cases} 1, & \text{если } a = b; \\ \min\left(\frac{a}{b + \neg a}; \frac{b}{a + \neg b}\right), & \text{в остальных случаях}; \\ 0, & \text{если } (a = 0) \vee (b = 0), \end{cases}$$

ALI3

где  $ka = \max(a, 1-a)$ .

**Определение 3.24. Верх нечеткого множества.** Верхом нечеткого множества В называют

$$hB = \bigvee_U \mu_B X .$$

**Определение 3.25. Низ нечеткого множества.** Низом нечеткого множества В называют

$$pB = \bigwedge_U \mu_B X .$$

**Определение 3.26. Четкость нечеткого множества.**

Четкостью некоторого  $a \in V$  называют выражение вида  $ka = a \vee (1-a)$ . Четкость нечеткого множества  $B$  определяется как

$$kB = \bigwedge_U k\mu_B X .$$

### 3.10. Семантический анализ нечеткой логики

Дадим краткий семантический анализ предложенных нечетких логик, пользуясь при этом терминологией, принятой в теории сильных нечетких множеств. Для этой цели сформулируем следующее.

**Предложение 3.1.** (степень возможности включения множеств). Определим функции вида  $\pi_i(A \subseteq B)$  в нечетких логиках ALI1-ALI3.

$$1. \pi_1(A \subseteq B) = \begin{cases} 1 - \mu_A x, & \text{если } \mu_A x < \mu_B x; \\ 1, & \text{если } \mu_A x = \mu_B x; \\ \mu_B x, & \text{если } \mu_A x > \mu_B x. \end{cases}$$

$$2. \pi_2(A \subseteq B) = \begin{cases} 1, & \text{если } \mu_A x \leq \mu_B x; \\ (1 - \mu_A x) \wedge \mu_B x, & \text{если } \mu_A x > \mu_B x. \end{cases}$$

$$3. \pi_3(A \subseteq B) = \begin{cases} 1, & \text{если } \mu_A x \leq \mu_B x; \\ \frac{\mu_B x}{\mu_A x + (1 - \mu_B x)}, & \text{если } \mu_A x > \mu_B x. \end{cases}$$

Отметим, что для нечеткой логики ALI1 четкое включение возможно, если  $\mu_A x = 0$ , или  $A = \emptyset$ . Далее рассмотрим вопрос эквивалентности множеств.

**Предложение 3.2.** (степень возможности эквивалентности множеств). Здесь множество  $T = \{x \in U / \mu_A x \neq \mu_B x\}$  и  $A=B$  означает, что  $\forall x \mu_A x = \mu_B x$ , или, иными словами,  $T = \emptyset$ .

$$1. \pi_1(A \equiv B) = \begin{cases} 1 - [(1 - \mu_A x) \vee \mu_A x], & \text{если } \mu_A x < \mu_B x; \\ 1, & \text{если } \mu_A x = \mu_B x; \\ 1 - [(1 - \mu_B x) \vee \mu_B x], & \text{если } \mu_A x > \mu_B x. \end{cases}$$

$$2. \pi_2(A \equiv B) = \begin{cases} 1, & \text{если } A = B; \\ T\{[(1 - \mu_A x) \wedge \mu_B x], [(1 - \mu_B x) \wedge \mu_A x]\}, & \text{если } A \neq B; \\ 0, & \text{если } \exists x \parallel \mu_A x = 0, \mu_B x \neq 0 \text{ (или наоборот)} \\ \text{а также } \exists x \parallel \mu_A x = 1, \mu_B x \neq 1 \text{ (или наоборот)}. \end{cases}$$

$$3. \pi_3(A \equiv B) = \begin{cases} 1, & \text{если } A = B; \\ T\left[\frac{\mu_A x}{\mu_B x + (1 - \mu_A x)}, \frac{\mu_B x}{\mu_A x + (1 - \mu_B x)}\right], & \\ \text{в противном случае} \\ 0, & \text{если } \exists x \parallel \mu_A x = 0, \text{ но } \mu_B x \neq 0 \\ \text{(или наоборот)}. \end{cases}$$

Символ  $\parallel$  означает «такой как».

Из выражений  $\pi_i(A \equiv B)$  следует, что для всех трех нечетких логик возможность эквивалентности  $\pi_i(A \equiv B) = 1$  имеет место только при действительной эквивалентности множеств, т.е.  $A=B$ . Очевиден также и тот факт, что возможность эквивалентности равна 0 в тех случаях, когда одно из высказываний четко, т.е. либо истинно, либо ложно, а другое нечетко.

**Предложение 3.3.** (степень с которой нечеткое множество  $B$  пусто). В описании  $B = \emptyset$  означает, что  $\forall x \mu_B x = 0$  или эквивалентно  $hB=0$ .

$$1. \pi_1(B \equiv \emptyset) = \begin{cases} 1, & \text{если } B = \emptyset; \\ 0, & \text{в противном случае.} \end{cases}$$

$$2. \pi_2(B \equiv \emptyset) = \begin{cases} 1, & \text{если } hB < 1 \text{ или } B = \emptyset; \\ 0, & \text{в противном случае.} \end{cases}$$

$$3. \pi_3(B \equiv \emptyset) = \begin{cases} 1, & \text{если } B = \emptyset; \\ 0, & \text{в противном случае.} \end{cases}$$

Введем понятие несовместности нечетких множеств.

Существует два вида «несовместности»: первый вид определяется степенью, с которой одно множество  $A$  является подмножеством дополнения второго  $B^c$ ; второй вид – степень с которой пересечение множеств пусто, поэтому сформулируем следующее.

**Предложение 3.4.** (степень несовместности множеств  $A$  и  $B$  есть степень, с которой  $A$  и  $B$  несовместны).

$$\pi(Adisj_1 B) = \pi(A \subseteq B^c) \wedge \pi(B \subseteq A^c), \quad (3.62)$$

$$\pi(Adisj_2 B) = \pi(A \cap B) = \emptyset. \quad (3.63)$$

При  $T = \{x \mid \mu_A x > 1 - \mu_B x\}$  рассмотрим несовместность первого вида:

$$1. \pi_1(Adisj_1 B) = \begin{cases} 1, & \text{если } \exists x \parallel \mid \mu_A x = 1 - \mu_B x; \\ (1 - \mu_A x) \wedge (1 - \mu_B x), & \text{в остальных случаях}; \\ 0, & \text{никогда.} \end{cases}$$

$$2. \pi_2(Adisj_1 B) = \begin{cases} 1, & \text{если } \mu_A x \leq 1 - \mu_B x; \\ T[(1 - \mu_A x), (1 - \mu_B x)], & \text{в остальных случаях}; \\ 0, & \text{если } \exists x \parallel \mid \mu_A x = 1, \text{ но } \mu_B x \neq 0 \\ & \text{или } \mu_B x = 1, \text{ но } \mu_A x \neq 0. \end{cases} \quad (3.64)$$

$$3. \pi_3(Adisj_1 B) = \begin{cases} 1, & \text{если } \mu_A x = \mu_B x \text{ или } \mu_B x = 0; \\ T\left(\frac{1 - \mu_B x}{\mu_A x + (1 - \mu_B x)}, \frac{1 - \mu_B x}{\mu_B x + (1 - \mu_A x)}\right), & \\ \text{в противном случае} & \\ 0, & \text{никогда.} \end{cases}$$

Отметим, что степень несовместности множеств равна 0 только для нечеткой логики ALI2, при этом обязательным условием является нормальность одного из рассматриваемых нечетких множеств при одновременной субнормальности другого.

**Предложение 3.5.** (степень, с которой множество является подмножеством своего дополнения). Для рассматриваемых систем  $\pi_i(A \subseteq A^c)$  будет иметь следующий вид:



$$1. \pi_1(A \subseteq A^c) = \begin{cases} 1, & \text{если } hA = 0; \\ 0, & \text{если } hA = 1; \\ 1 - hA, & \text{в остальных случаях.} \end{cases}$$

$$2. \pi_2(A \subseteq A^c) = \begin{cases} 1, & \text{если } hA \leq 0,5; \\ 0, & \text{если } hA = 1; \\ 1 - hA, & \text{в остальных случаях.} \end{cases}$$

$$3. \pi_3(A \subseteq A^c) = \begin{cases} 1, & \text{если } hA \leq 0,5; \\ 0, & \text{если } hA = 1; \\ 1 - hA/(2hA), & \text{в остальных случаях.} \end{cases}$$

Очевидно, что для нечеткой логики ALI1 степень, с которой множество является подмножеством своего дополнения, равна степени, с которой это множество является пустым. Следует также отметить, что семантический анализ, проведенный в [2], а также анализ, проделанный выше, показывает большую схожесть в свойствах нечетких логик ALI1 и KD. Однако, как будет показано ниже, нечеткая логика ALI1, по сравнению с KD-логикой, имеет ряд преимуществ, позволяющих успешно ее использовать при формализации улучшенных правил нечеткого условного вывода и для моделирования различных технологических процессов.

### 3.11. Нечеткое лингвистическое моделирование

#### Базовая архитектура систем нечеткого вывода

Рассматриваемые системы нечеткого вывода являются частным случаем продукционных нечетких систем или систем нечетких правил продукций, в которых условия и заключения, отдельных правил формируются в форме нечетких высказывания относительно значений тех или иных лингвистических переменных.

#### Нечеткие лингвистические высказывания

Нечетким лингвистическим высказыванием будем называть высказывания следующих видов.

1. Высказывание "β есть α", где β – наименование лингвистической переменной, α – ее значение, которому

соответствует отдельный лингвистический терм из базового термножества  $T$  лингвистической переменной  $\beta$ .

2. Высказывание " $\beta$  есть  $\nabla\alpha$ ", где  $\nabla$  – модификатор, соответствующий таким словам, как: "ОЧЕНЬ", "БОЛЕЕ ИЛИ МЕНЕЕ", "МНОГО БОЛЬШЕ" и другим, которые могут быть получены с использованием процедур  $G$  и  $M$  данной лингвистической переменной.

3. Составные высказывания, образованные из высказываний видов 1 и 2 и нечетких логических операций в форме связок: "И", "ИЛИ", "ЕСЛИ-ТО", "НЕ".

Поскольку в системах нечеткого вывода нечеткие лингвистические высказывания занимают центральное место, далее будем их называть просто нечеткими высказываниями.

Рассмотрим некоторые примеры нечетких высказывания. Первое из них – "скорость автомобиля высокая" представляет собой нечеткое высказывание первого вида, в рамках которого лингвистической переменной "скорость автомобиля" присваивается значение "высокая". При этом предполагается, что на универсальном множестве  $X$  переменной "скорость автомобиля" определен соответствующий лингвистический терм "высокая", который задается в форме функций принадлежности некоторого нечеткого множества.

Нечеткое высказывание второго вида "скорость автомобиля очень высокая" означает, что лингвистической переменной "скорость автомобиля" присваивается значение "высокая" с модификатором "ОЧЕНЬ", который изменяет значение соответствующего лингвистического термина "высокая" на основе использования расчетной формулы.

### **Правила нечетких продукций в системах нечеткого вывода**

Простейший вариант правила нечеткой продукций, который наиболее часто используется в системах нечеткого вывода, может быть записан в форме:

**ПРАВИЛО <#>:** ЕСЛИ " $\beta_1$  есть  $\alpha$ ", ТО " $\beta_2$  есть  $\alpha$ ".

Здесь нечеткое высказывание " $\beta_1$  есть  $\alpha$ " представляет собой условие данного правила нечеткой продукций, а нечеткое высказывание " $\beta_2$  есть  $\alpha$ " – нечеткое заключение данного правила. При этом считается, что  $\beta_1 \neq \beta_2$ .

Система нечетких правил продукций или продукционная нечеткая система представляет собой некоторое согласованное множество отдельных нечетких продукций или правил нечетких продукций в форме "ЕСЛИ А, ТО В", где А и В – нечеткие лингвистические высказывания вида 1,2 или 3.

### **Основные этапы нечеткого вывода**

Основными этапами нечеткого вывода являются:

- Формирование базы правил систем нечеткого вывода.
- Фаззификация входных переменных.
- Агрегирование подусловий в нечетких правилах продукций.
- Активизация (композиция) подзаклучений в нечетких правилах продукций.
- Вывод заключений нечетких правил продукций.

Рассмотрим каждый в отдельности.

База правил нечетких продукций представляет собой конечное множество правил нечетких продукций, согласованных относительно используемых в них лингвистических переменных. Наиболее часто база правил представляется в форме структурированного текста:

ПРАВИЛО\_1: ЕСЛИ "Условие\_1" ТО "Заключение\_1"

ПРАВИЛО\_2: ЕСЛИ "Условие\_2" ТО "Заключение\_2"

...

ПРАВИЛО\_n: ЕСЛИ "Условие\_n" ТО "Заключение\_n".

Согласованность правил относительно используемых лингвистических переменных означает, что в качестве условий и заключений правил могут использоваться только нечеткие высказывания вида:

ПРАВИЛО <#>: ЕСЛИ " $\beta_1$  есть  $\alpha$ " И " $\beta_2$  есть  $\alpha$ ", ТО " $\beta_3$  есть  $\nu$ "

или

ПРАВИЛО <#>: ЕСЛИ " $\beta_1$  есть  $\alpha$ " ИЛИ " $\beta_2$  есть  $\alpha$ ", ТО " $\beta_3$  есть  $\nu$ ".

ПРАВИЛО <#>: ЕСЛИ " $\beta_1$  есть  $\alpha$ ", ТО " $\beta_2$  есть  $\alpha$ " И " $\beta_3$  есть  $\nu$ "

или

ПРАВИЛО <#>: ЕСЛИ " $\beta_1$  есть  $\alpha$ ", ТО " $\beta_2$  есть  $\alpha$ " ИЛИ " $\beta_3$  есть  $\nu$ ".

При этом в каждом из нечетких высказываний должны быть определены функции принадлежности значений терм-множества для каждой из лингвистических переменных.

В системах нечеткого вывода лингвистические переменные, которые используются в нечетких высказываниях подусловий правил нечетких продукций, часто называют входными лингвистическими переменными, а переменные, которые используются в нечетких высказываниях подзаклучений правил нечетких продукций, часто называют выходными лингвистическими переменными.

Таким образом, при задании или формировании базы правил нечетких продукций необходимо определить: множество правил нечетких продукций  $P = \{\text{ПРАВИЛО}_1, \text{ПРАВИЛО}_2, \dots, \text{ПРАВИЛО}_n\}$ , множество входных лингвистических переменных  $V = \{\beta_1, \beta_2, \dots, \beta_m\}$  и множество выходных лингвистических переменных  $W = \{\omega_1, \omega_2, \dots, \omega_l\}$ . Тем самым база правил считается заданной, если заданы множества  $P, V, W$ .

Под фаззификацией понимается не только этап выполнения нечеткого вывода, но и собственно процесс или процедура нахождения значений функций принадлежности нечетких множеств(термов) на основе обычных(не нечетких) исходных данных. Фаззификацию еще называют введением нечеткости.

Агрегирование представляет собой процедуру определения степени истинности условий по каждому из правил системы нечеткого вывода.

Активизация представляет собой процедуру или процесс нахождения степени истинности каждого из подзаклучений правил нечетких продукций.

Методы нечеткой композиции:

-min-активизация:

$$\mu'(y) = \min\{c_i, \mu(y)\},$$

где  $C = \{c_1, c_2, \dots, c_q\}$  - множество степеней истинности,  $q$  – общее количество подзаклучений в базе правил;

-prod-активизация:

$$\mu'(y) = c_i \cdot \mu(y);$$

-average-активизация:

$$\mu'(y) = 0,5 \cdot (c_i + \mu(y)),$$

где  $\mu(y)$  – функция принадлежности терма, который является значением некоторой выходной переменной  $\omega$ .

Вывод заключений нечетких правил продукций или аккумуляция в системах нечеткого вывода представляет собой процедуру или процесс нахождения функции принадлежности для каждой из выходных лингвистических переменных множества  $W = \{\omega_1, \omega_2, \dots, \omega_n\}$ .

Цель аккумуляции заключается в том, чтобы объединить или аккумулятировать все степени истинности заключений (подзаключений) для получения функции принадлежности каждой из выходных переменных. Причина необходимости выполнения этого этапа состоит в том, что подзаключения, относящиеся к одной и той же выходной лингвистической переменной, принадлежат различным правилам нечеткого вывода.

Дефаззификация в системах нечеткого вывода представляет собой процедуру или процесс нахождения обычного значения для каждой из выходных лингвистических переменных множества  $W = \{\omega_1, \omega_2, \dots, \omega_n\}$ .

Цель дефаззификации заключается в том, чтобы, используя результаты аккумуляции всех выходных лингвистических переменных, получить обычное количественное значение каждой из выходных переменных, которое может быть использовано специальными устройствами, внешними по отношению к системе нечеткого вывода.

Методы дефаззификации:

- Метод центра тяжести:

$$y = \frac{\int_{Min}^{Max} x \cdot \mu(x) dx}{\int_{Min}^{Max} \mu(x) dx},$$

где  $y$  – результат дефаззификации,  $x$  – переменная, соответствующая выходной лингвистической переменной  $\omega$ ,  $\mu(x)$  – функция принадлежности нечеткого множества, соответствующего выходной переменной  $\omega$  после этапа аккумуляции, Min и Max – левая и правая точки интервала носителя нечеткого множества рассматриваемой выходной переменной  $\omega$ ;

- Метод центра тяжести для одноточечных множеств:

$$y = \frac{\sum_{i=1}^n x_i \cdot \mu(x_i)}{\sum_{i=1}^n \mu(x_i)},$$

где  $n$  – число одноточечных (одноэлементных) нечетких множеств, каждое из которых характеризует единственное значение рассматриваемой лингвистической переменной;

- Метод центра площади:

Центр площади равен  $y = \mu$ , где значение  $\mu$  определяется из уравнения:

$$\int_{Min}^{\mu} \mu(x) dx = \int_{\mu}^{Max} \mu(x) dx,$$

другими словами, центр площади равен абсциссе, которая делит площадь, ограниченную графиком кривой функции принадлежности соответствующей выходной переменной, на две равные части;

- Метод левого модального значения:

$$y = \min\{x_m\},$$

где  $x_m$  – модальное значение (мода) нечеткого множества, соответствующего выходной переменной  $\omega$  после аккумуляции, другими словами, значение выходной переменной определяется как мода нечеткого множества для соответствующей выходной переменной или наименьшая из мод(самая левая), если нечеткое множество имеет несколько модальных значений;

- Метод правого модального значения:

$$y = \max\{x_m\},$$

где  $x_m$  – модальное значение(мода) нечеткого множества, соответствующего выходной переменной  $\omega$  после аккумуляции, другими словами, значение выходной переменной определяется как мода нечеткого множества для соответствующей выходной переменной или наибольшая из мод(самая правая), если нечеткое множество имеет несколько модальных значений;

Алгоритм Мамдани является одним из первых, который нашел применение в системах нечеткого вывода. Он был предложен в 1975г. Английским математиком Е. Мамдани (EbrahimMamdani) в качестве метода для управления паровым двигателем. По своей сути этот алгоритм порождает

рассмотренные выше этапы, поскольку в наибольшей степени соответствует их параметрам.

Формально алгоритм Мамдани<sup>1</sup> может быть определен следующим образом:

- Формирование базы правил систем нечеткого вывода. Совпадают с рассмотренными выше.

- Фаззификация входных переменных. Совпадают с рассмотренными выше.

- Агрегирование подусловий в нечетких правилах продукций. Для нахождения степени истинности условий каждого из правил нечетких продукций используются парные нечеткие логические операции. Те правила, степень истинности условий которых отлична от нуля, считаются активными и используются для дальнейших расчетов.

- Активизация подзаклучений в нечетких правилах продукций. Осуществляется по формуле min-активизации, при этом для сокращения времени вывода учитываются только активные правила нечетких продукций.

- Аккумуляция заключений нечетких правил продукций. Осуществляется по формуле объединения нечетких множеств, соответствующих термам подзаклучений, относящихся к одним и тем же выходным лингвистическим переменным.

- Дефаззификация выходных переменных. Традиционно используется метод центра тяжести для многоэлементных или одноэлементных множеств или метод центра площади.

### **Алгоритм Цукамото (Tsukamoto)**

Формально алгоритм Цукамото<sup>2</sup> может быть определен следующим образом:

- Формирование базы правил систем нечеткого вывода. Совпадают с рассмотренными выше.

- Фаззификация входных переменных. Совпадают с рассмотренными выше.

- Агрегирование подусловий в нечетких правилах продукций. Для нахождения степени истинности условий всех правил нечетких

---

<sup>1</sup> - Леоненков А.В. Нечеткое моделирование в среде MATLAB и fuzzyTECH. – СПб.: БХВ-Петербург, 2005. С.202.

<sup>2</sup> - Леоненков А.В. Нечеткое моделирование в среде MATLAB и fuzzyTECH. – СПб.: БХВ-Петербург, 2005. С.202.

продукций используются парные нечеткие логические операции. Те правила, степень истинности условий которых отлична от нуля, считаются активными и используются для дальнейших расчетов.

- Активизация подзаключений в нечетких правилах продукций. Осуществляется аналогично алгоритму Мамдани, после чего находятся обычные значения всех выходных лингвистических переменных в каждом из подзаключений активных правил нечетких продукций. В этом случае значение выходной лингвистической переменной  $\omega_j$  в каждом из подзаключений находится как решение уравнения:

$$c_i = \mu(\omega_j) (\forall i \in \{1, 2, \dots, q\}),$$

где  $q$  – общее количество подзаключений в базе правил.

- Аккумуляция заключений нечетких правил продукций. Фактически отсутствует, поскольку расчеты осуществляются с обычными действительными числами  $\omega_j$ .

- Дефаззификация выходных переменных. Используется модифицированный вариант в форме метода центра тяжести для одноточечных множеств:

$$y = \frac{\sum_{i=1}^n c_i \cdot \omega_i}{\sum_{i=1}^n c_i},$$

где  $n$  – общее количество активных правил нечетких продукций, в подзаключениях которых присутствует выходная лингвистическая переменная  $\omega_j$ .

### **Алгоритм Ларсена (Larsen)**

Формально алгоритм Ларсена<sup>3</sup> может быть определен следующим образом:

- Формирование базы правил систем нечеткого вывода. Совпадают с рассмотренными выше.

- Фаззификация входных переменных. Совпадают с рассмотренными выше.

- Агрегирование подусловий в нечетких правилах продукций. Используются парные нечеткие логические операции для нахождения степени истинности условий всех правил нечетких продукций (как правило,  $\max$  – дизъюнкция и  $\min$  – конъюнкция).

---

<sup>3</sup> - Леоненков А.В. Нечеткое моделирование в среде MATLAB и fuzzyTECH. – СПб.: БХВ-Петербург, 2005. С.203.



Те правила, степень истинности условий которых отлична от нуля, считаются активными и используются для дальнейших расчетов.

- Активизация подзаклучений в нечетких правилах продукций. Осуществляется использованием формулы prod-активации, посредством чего находится совокупность нечетких множеств:  $C_1, C_2, \dots, C_q$ , где  $q$  – общее количество подзаклучений в базе правил.

- Аккумуляция заклучений нечетких правил продукций. Осуществляется по формуле объединения нечетких множеств, соответствующих термам подзаклучений, относящихся к одним и тем же выходным лингвистическим переменным.

- Дефаззификация выходных переменных. Может использоваться любой из рассмотренных выше методов дефаззификации.

### Алгоритм Сугено(Sugeno)

Формально алгоритм Сугено<sup>4</sup>, предложенный Сугено и Такаги, может быть определен следующим образом:

- Формирование базы правил систем нечеткого вывода. В базе правил используются только правила нечетких продукций в форме:

ПРАВИЛО <#>: ЕСЛИ “ $\beta_1$  есть  $\alpha$ ” И ” $\beta_2$  есть  $\alpha$ ”, ТО ”  $\omega = \varepsilon_1 \cdot a_1 + \varepsilon_2 \cdot a_2$ ”.

Здесь  $\varepsilon_1, \varepsilon_2$  - некоторые весовые коэффициенты (пишут сбоку от правил, обычно равны 1). При этом значение выходной переменной  $\omega$  в заклучении определяется как некоторое действительное число.

- Фаззификация входных переменных. Совпадают с рассмотренными выше.

- Агрегирование подусловий в нечетких правилах продукций. Для нахождения степени истинности условий всех правил нечетких продукций, как правило, используется логическая операция min - конъюнкции. Те правила, степень истинности условий которых отлична от нуля, считаются активными и используются для дальнейших расчетов.

- Активизация подзаклучений в нечетких правилах продукций. Во-первых, с использованием метода min-активизации

---

<sup>4</sup> - Леоненков А.В. Нечеткое моделирование в среде MATLAB и fuzzyTECH. – СПб.: БХВ-Петербург, 2005. С.204.

находятся значения степеней истинности всех заключений правил нечетких продукций. Во-вторых, осуществляется расчет обычных значений выходных переменных каждого правила. Это выполняется с использованием формулы для заключения:

ПРАВИЛО <#>: ЕСЛИ “ $\beta_1$  есть  $\alpha$ ” И “ $\beta_2$  есть  $\alpha$ ”, ТО “ $\omega = \varepsilon_1 \cdot a_1 + \varepsilon_2 \cdot a_2$ ”,

в которую вместо  $a_1$  и  $a_2$  подставляются значения входных переменных до этапа фаззификации. Тем самым определяются множество значений  $C = \{c_1, c_2, \dots, c_n\}$  и множество значений выходных переменных  $W = \{\omega_1, \omega_2, \dots, \omega_n\}$ , где  $n$  – общее количество правил в базе правил.

- Аккумуляция заключений нечетких правил продукций. Фактически отсутствует, поскольку расчеты осуществляются с обычными действительными числами  $\omega_j$ .

- Дефаззификация выходных переменных. Используется модифицированный вариант в форме метода центра тяжести для одноточечных множеств.

### **Алгоритм Мамдани в системах нечеткого вывода**

Предположим, что базу знаний образуют два нечётких правила:

$P_1$ : если  $x$  есть  $A_1$  и  $y$  есть  $B_1$ , то  $z$  есть  $C_1$ ,

$P_2$ : если  $x$  есть  $A_2$  и  $y$  есть  $B_2$ , то  $z$  есть  $C_2$ ,

где  $x$  и  $y$  – имена входных переменных,  $z$  – имя переменной вывода,  $A_1, A_2, B_1, B_2, C_1, C_2$  – некоторые заданные функции принадлежности, при этом чёткое знание  $z_0$  необходимо определить на основании приведённой информации и чётких знаний  $x_0, y_0$ .

Данный алгоритм математически может быть описан следующим образом.

1. Нечёткость: находятся степени истинности для предпосылок каждого правила:  $A_1(x_0), A_2(x_0), B_1(x_0), B_2(x_0)$ .

2. Нечёткий вывод: находятся уровни отсечения для предпосылок каждого из правил (с использованием операции минимума):

$$\alpha_1 = A_1(x_0) \wedge B_1(y_0),$$

$$\alpha_2 = A_2(x_0) \wedge B_2(y_0),$$

где через « $\wedge$ » обозначена операция логического минимума.

3. Композиция: с использованием операции  $\max$  производится объединение найденных усеченных функций, что приводит к получению итогового нечёткого подмножества для переменной выхода с функцией принадлежности

4. Приведение к чёткости (для нахождения  $z_0$ ) производится, например, центроидным методом

$$z_0 = \frac{\int_{\Omega} z \mu_{\Sigma}(z) dz}{\int_{\Omega} \mu_{\Sigma}(z) dz}.$$

*Нечеткая переменная* – это кортеж вида  $\langle \alpha, X, A \rangle$ , где:

$\alpha$  – имя нечеткой переменной;

$X$  – её область определения;

$A$  – нечеткое множество на универсуме  $X$ .

Пример: Нечеткая переменная  $\langle$  «Тяжелый бронежилет»,  $\{x \mid 0 \text{ кг} < x < 35 \text{ кг}\}$ ,  $V = \{x, \mu(x)\}$  характеризует массу военного бронежилета. Будем считать его тяжелым, если его масса  $> 16$  кг (рис. 3.16).

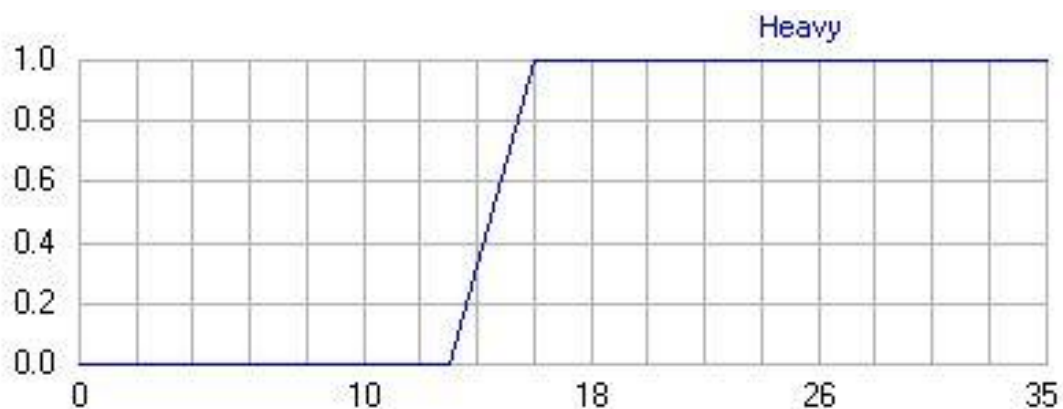


Рис. 3.16. График функции принадлежности  $\mu(x)$  для нечеткого множества  $V$

Лингвистическая переменная есть кортеж  $\langle \beta, T, X, G, M \rangle$ , где:  
 $\beta$  – имя лингвистической переменной;  
 $T$  – множество её значений (термов);  
 $X$  – универсум нечетких переменных;  
 $G$  – синтаксическая процедура образования новых термов;  
 $M$  – семантическая процедура, формирующая нечеткие множества для каждого терма данной лингвистической переменной.

Пример: Допустим, мы имеем субъективную оценку массы бронезилета. Она, например, может быть получена от военнослужащих (выступающих в роли экспертов), которые непосредственно имеют дело с подобной амуницией. Формализовать эту оценку можно с помощью следующей лингвистической переменной  $\langle \beta, T, X, G, M \rangle$  (рис. 3.17), где:

$\beta$  – Бронезилет;

$T$  – {«Легкий бронезилет (Light)», «Бронезилет средней массы (Medium)», «Тяжелый бронезилет (Heavy)»};

$X = [0; 35]$ ;

$G$  – процедура образования новых термов при помощи логических связок и модификаторов. Например, «очень тяжелый бронезилет»;

$M$  – процедура задания на универсуме  $X=[0; 35]$  значений лингвистической переменной, т.е. термов из множества  $T$ .

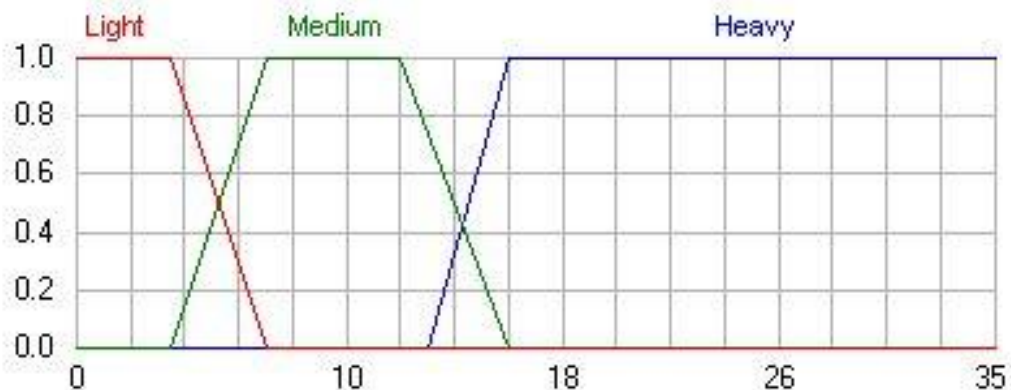


Рис. 3.17. Графики функций принадлежности значений лингвистической переменной «Бронезилет»

*Нечетким высказыванием* будем называть высказывание вида « $\beta$  IS  $\alpha$ », где:

$\beta$  – лингвистическая переменная;

$\alpha$  – один из термов этой переменной

Пример: «Бронежилет IS легкий». Здесь «Бронежилет» – это лингвистическая переменная, а «легкий» её значение.

Упрощенно говоря, правилом нечетких продукций (далее просто правилом) будем называть классическое правило вида «ЕСЛИ... ТО...», где в качестве условий и заключений будут использоваться нечеткие высказывания. Записываются такие правила в следующем виде:

IF ( $\beta_1$  IS  $\alpha_1$ ) AND ( $\beta_2$  IS  $\alpha_2$ ) THEN ( $\beta_3$  IS  $\alpha_3$ ).

Кроме «AND» также используются логическая связка «OR». Но такую запись обычно стараются избегать, разделяя такие правила на несколько более простых (без «OR»). Также каждое из нечетких высказываний в условии любого правила будем называть подусловием. Аналогично, каждое из высказываний в заключении называется подзаключением.

Пример: Следующие примеры помогут зафиксировать определение:

1) IF (Бронежилет тяжелый) THEN (Солдат уставший);

2) IF (Муж трезвый) AND (Зарплата высокая) THEN (Жена довольная).

Все. Этого минимума достаточно для понимая принципов работы алгоритма.

### **Принцип работы алгоритма Мамдани**

Данный алгоритм описывает несколько последовательно выполняющихся этапов (рисунке 3.18). При этом каждый последующий этап получает на вход значения полученные на предыдущем шаге.



Рисунок 3.18. Диаграмма деятельности процесса нечеткого вывода

Алгоритм примечателен тем, что он работает по принципу «черного ящика». На вход поступают количественные значения, на

выходе они же. На промежуточных этапах используется аппарат нечеткой логики и теория нечетких множеств. В этом и состоит элегантность использования нечетких систем. Можно манипулировать привычными числовыми данными, но при этом использовать гибкие возможности, которые предоставляют системы нечеткого вывода.

Для реализации алгоритма использовался объектно-ориентированный подход. Диаграмма (рисунке 3.9) показывает наиболее существенные связи и отношения между классами, задействованными в алгоритме.

Алгоритм Мамдани (MamdaniAlgorithm), включает в себя все этапы (рис. 19) и использует базу правил (List<Rule>) в качестве входных данных. Также алгоритм предполагает использование «активизированных» нечетких множеств (ActivatedFuzzySet) и их объединений (UnionOfFuzzySets).

Итак, этапы нечеткого вывода выполняются последовательно. И все значения, полученные на предыдущем этапе, могут использоваться на следующем.

### **Формирование базы правил**

*База правил* – это множество правил, где каждому подзаклучению сопоставлен определенный весовой коэффициент.

База правил может иметь следующий вид (для примера используются правила различных конструкций):

RULE\_1: IF «Condition\_1» THEN «Conclusion\_1» (F1) AND «Conclusion\_2» (F2);

RULE\_2: IF «Condition\_2» AND «Condition\_3» THEN «Conclusion\_3» (F3);

...

RULE\_n: IF «Condition\_k» THEN «Conclusion\_(q-1)» (Fq-1) AND «Conclusion\_q» (Fq);

где  $F_i$  – весовые коэффициенты, означающие степень уверенности в истинности получаемого подзаклучения ( $i = 1..q$ ). По умолчанию весовой коэффициент принимается равным 1. Лингвистические переменные, присутствующие в условиях называются входными, а в заключениях выходными.

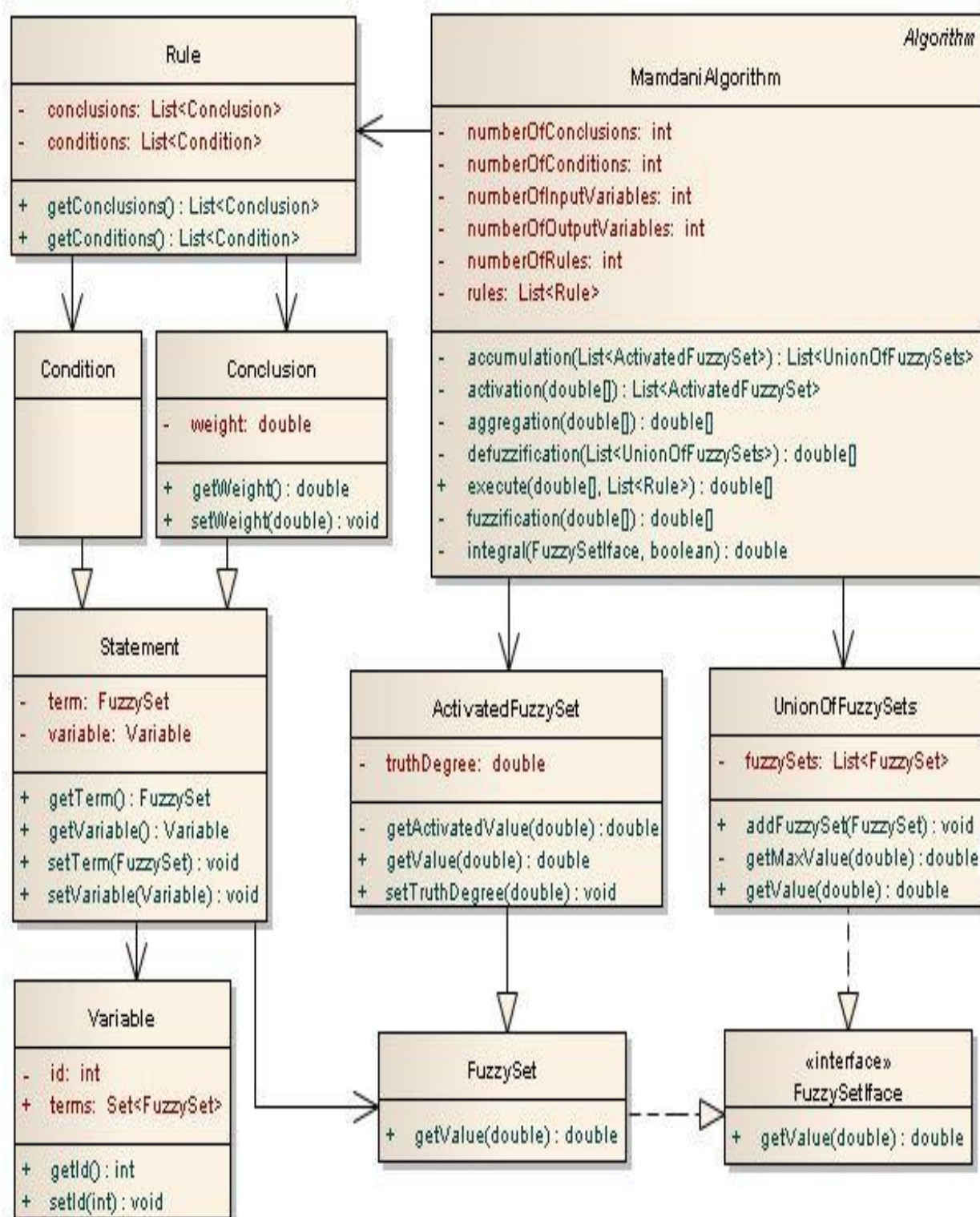


Рис. 3.19. Диаграмма классов реализации алгоритма Мамдани



Обозначения:

$n$  – число правил нечетких продукций (numberOfRules).

$m$  – кол-во входных переменных (numberOfInputVariables).

$s$  – кол-во выходных переменных (numberOfOutputVariables).

$k$  – общее число подусловий в базе правил (numberOfConditions).

$q$  – общее число подзаключений в базе правил (numberOfConclusions).

Данные обозначения будут использоваться в последующих этапах. В скобках указаны имена соответствующих переменных в исходном коде.

В результате объединения нескольких технологий искусственного интеллекта появился специальный термин – «мягкие вычисления» (soft computing), который ввел Л. Заде в 1994 году. В настоящее время мягкие вычисления объединяют такие области как: нечеткая логика, искусственные нейронные сети, вероятностные рассуждения и эволюционные алгоритмы. Они дополняют друг друга и используются в различных комбинациях для создания гибридных интеллектуальных систем.

Влияние нечеткой логики оказалось, пожалуй, самым обширным. Подобно тому, как нечеткие множества расширили рамки классической математическую теорию множеств, нечеткая логика «вторглась» практически в большинство методов Data Mining, наделив их новой функциональностью.



## Глава 4. НЕЙРОННЫЕ СЕТИ

### 4.1. Однослойные и многослойные нейронные сети

**Биологический нейрон.** Различные типы нервных клеток объединяются под общим термином – нейрон. Нейрон представляет собой особый вид клеток в живых организмах, обладающих электрической активностью, основное назначение которого заключается в оперативном управлении организмом. Схематическое изображение биологического нейрона представлено на рис.4.1. Нейрон состоит из тела клетки, содержащей ядро и цитоплазму, заключенные в оболочку мембрану. Нейрон включает также дерево входов (дендритов) и выходов (аксон и его окончания). Концевые разветвления аксона примыкают к другим клеткам через синаптические контакты (синапсы). От аксона отходят также отростки – коллатералы, которые также примыкают к другим клеткам.

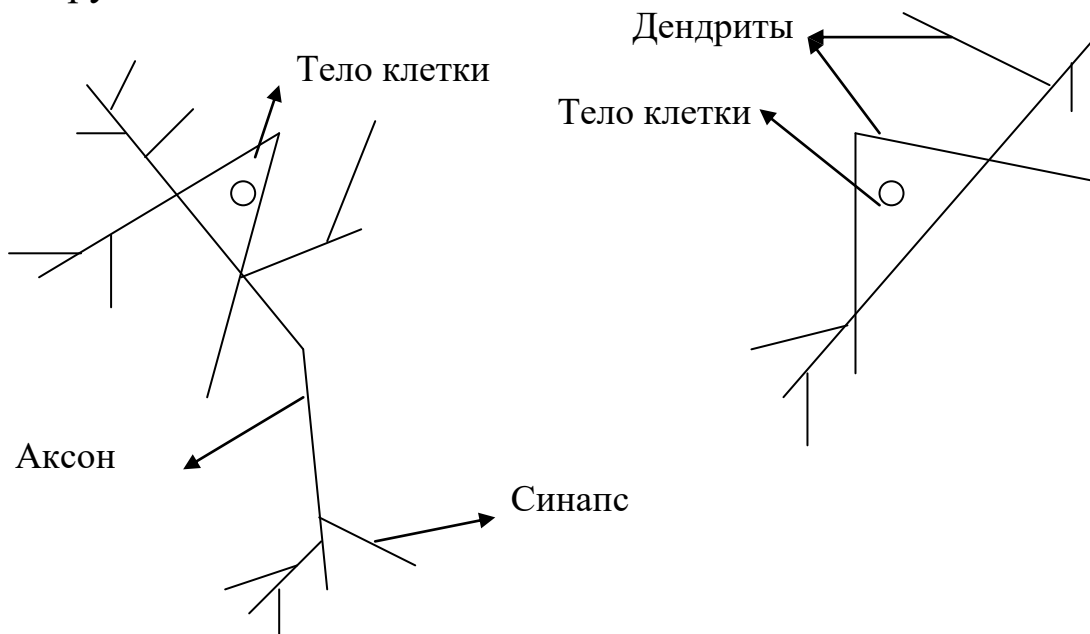


Рис.4.1. Биологический нейрон

Входные сигналы дендритного дерева (постсинаптические потенциалы) взвешиваются и суммируются на пути к начальным сегментам аксона, где генерируется выходной импульс (или импульсы). Интенсивность его, следовательно, является функцией от взвешенной суммы входных сигналов. Выходной сигнал проходит по ветвям аксона и достигает синапсов, которые соединяет аксоны и дендритными деревьями других нейронов.

Через синапсы сигнал трансформируется в новый входной сигнал для смежных нейронов. Этот входной сигнал может быть положительным и отрицательным (возбуждающим или тормозящим), в зависимости от вида синапсов. Величина входного сигнала, генерируемого синапсом, может быть различной даже при одинаковой величине сигнала, приходящего в синапс. Такое различие связано с эффективностью синапса, определяемой его весом. Последний может изменяться в процессе функционирования синапса.

Различают три группы нейронов: рецепторные, промежуточные и эффекторные. Рецепторные нейроны обеспечивают ввод в мозг сенсорной информации. Они трансформируют сигналы, поступающие на органы чувств (оптические сигналы в сетчатке глаза, акустические в ушной улитке и др.), в электрическую импульсацию своих аксонов. Эффекторные нейроны передают поступающие к ним сигналы исполнительным органам. На концах их аксонов имеются специальные синаптические соединения с исполнительными органами, например, мышцами, где возбуждение нейронов трансформируется в сокращения мышц. Промежуточные нейроны осуществляют обработку информации, получаемой от рецепторов, и формируют управляющие сигналы для эффекторов. Они образуют центральную нервную систему.

Число синапсов одного нейрона колеблется от сотен до тысяч. Состояние мембраны нейрона (ее электрический потенциал) зависит от величины и длительности сигналов, поступающих на синапсы. Когда потенциал мембраны достигает определенного – порогового значения (приблизительно 40 мВ), возникает нервный импульс – волна активации, распространяющаяся вдоль нервного волокна аксона. Раздражение, передающееся по волокну, представляет собой электрохимический процесс; скорость его распространения зависит от диаметра волокна и лежит в пределах от 1 м/с до 150 м/с. После проведения импульса нервовое волокно находится в состоянии полной невозбудимости (так называемый рефракторный период), т.е. не проводит нервные сигналы, какова бы не была сила раздражения.

**Математическая модель нейрона.** Математическая модель нейрона, которая обычно используется при моделировании

нейронных сетей, состоящих из большого количества взаимосвязанных моделей нейронов, представлена на рис. 4.2 [1,2,3]. На нейрон поступает набор входных

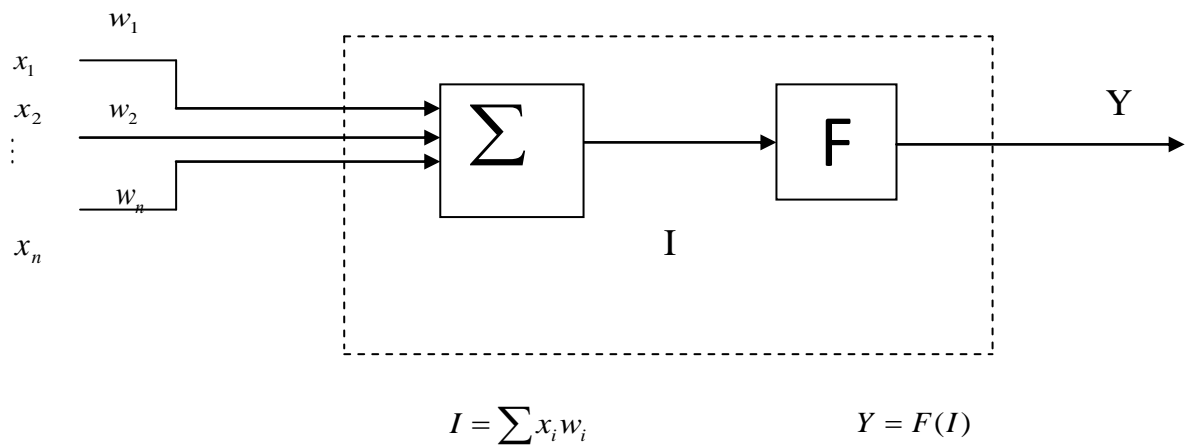


Рис. 4.2. Математическая модель нейрона

сигналов  $x_1, x_2, \dots, x_n$  (или вектор  $X$ ), представляющих собой выходные сигналы других нейронов. Этот входной вектор соответствует сигналам, поступающим в синапсы биологических нейронов. Каждый входной сигнал умножается на соответствующий вес связи,  $w_1, w_2, \dots, w_n$  - аналог эффективности синапса. Вес связи является скалярной величиной, положительной для возбуждающих и отрицательной для тормозящих связей. Взвешенные весами связей входные сигналы поступают на блок суммирования, соответствующий телу клетки, где осуществляется их алгебраическое суммирование и определяется уровень возбуждения нейрона

$$I = \sum w_i x_i.$$

Выходной сигнал нейрона определяется путем пропускания уровня возбуждения  $I$  через нелинейную функцию  $F$

$$Y = F(I - \theta),$$

где  $\theta$  - порог нейрона. Обычно в качестве функции  $F$  используются простейшие нелинейные функции следующих видов:

а) бинарная (пороговая)

$$y = \begin{cases} 1, & \text{при } I > \theta \\ 0, & \text{при } I \leq \theta. \end{cases}$$

б) сигмоидная

$$y = \frac{1}{1 - e^{-(I-\theta)}}.$$

Соединив определенным образом друг с другом и с внешней средой совокупность нейронов можно построить на основе их выше приведенной модели нейронную сеть. Входной вектор подается на сеть путем активации входных нейронов. Множество выходных сигналов нейронов сети  $y_1, y_2, \dots, y_n$  называют вектором выходной активности. Веса связей нейронов сети представляются в виде матрицы  $W$ , элемент  $w_{ij}$  которой есть вес связи между  $i$ -ым и  $j$ -ым нейронами. В процессе функционирования сети осуществляется преобразование входного вектора в выходной, т.е. некоторая переработка информации. Следует отметить, что конкретный вид выполняемого сетью преобразования информации обуславливается не только характеристиками моделей нейронов, но и особенностью ее архитектуры.

## 4.2. Архитектура и типы нейронных сетей

**Полносвязные и иерархические сети.** Нейронная сеть представляет собой совокупность отдельных нейронов, связанных в определенную структуру. Вычислительная мощность сети, задачи, которые она может решать, задаются именно этими связями. Связи соединяют входы одних нейронов с выходами других, а их «сила» задается весовыми коэффициентами (или, просто весами). Таким образом, сила влияния поведения одного нейрона на поведение другого определяется соответствующим весом связи. Поэтому, нейронные системы часто называют также коннекционистскими (от слова connection – соединение, связь).

Порядок связей в сети представляет ее архитектуру [1,2,36]. Можно выделить два типа архитектуры нейронных сетей: полносвязные и иерархические сети.

Как известно из теории графов, в полносвязной архитектуре все элементы сети соединены друг с другом. В терминах нейронных сетей это означает, что выход каждого нейрона соединен со входами всех других нейронов и его входы связаны с выходами остальных элементов. Кроме того, выход каждого нейрона подсоединен к его входу (так называемая «связь сам-на-себя»). Количество связей в полносвязной сети, содержащей  $N$

нейронов равна  $N \cdot N$ , так как от каждого узла исходит  $N$  связей. На рис.4.3 представлена полносвязная сеть, содержащая 6 нейронов.

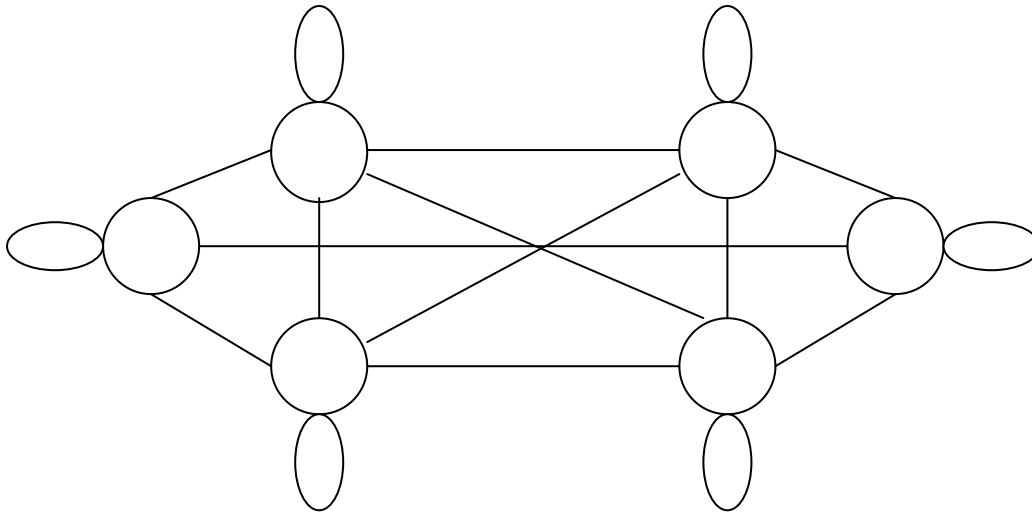


Рис. 4.3. Полносвязная нейронная сеть

В иерархической архитектуре можно выделить группы нейронов расположенных на отдельных слоях или уровнях. Каждый нейрон слоя связан с каждым из нейронов предыдущего и последующего слоев. Нейроны входного слоя получают сигналы от внешней среды и распределяют их по нейронам последующего слоя. Выходы нейронов выходного слоя поступают во внешнюю среду. Слои, расположенные между входным и выходным называются промежуточными или скрытыми (так как они не имеют непосредственных связей с внешней средой).

На рис. 4.4 изображена структура иерархической сети, включающей три слоя.

По направлению передачи сигналов в сети можно выделить сети без обратных связей, или не рекуррентные (feed-forward) сети и сети с обратными связями, или рекуррентные (feed-back) сети.

В сетях без обратных связей нейроны одного слоя принимают сигналы только от внешней среды или от нейронов предыдущего слоя и передают сигналы либо во внешнюю среду либо на входы нейронов последующего слоя.

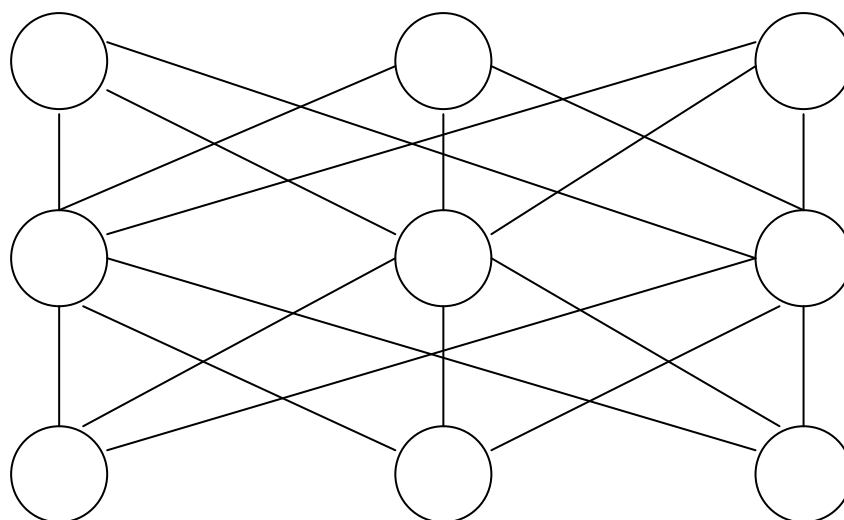


Рис. 4.4. Иерархическая нейронная сеть

В рекуррентных сетях нейроны определенного слоя могут, кроме того, принимать сигналы также от самих себя и других нейронов, расположенных на том же слое. Таким образом, в отличие от нереккуррентных сетей, значения выходных сигналов нейронов рекуррентной сети определяются не только значениями текущих сигналов на входах нейронов и весами соответствующих связей, но и значениями выходов некоторых нейронов в предыдущий момент времени. Это означает, что такая сеть обладает элементами памяти, что позволяет запоминать информацию о состояниях выходов на определенное время.

В случае, когда рекуррентная сеть имеет тормозящие связи (т.е. связи с отрицательными весами) с нейронами своего слоя, ее называют сетью с латеральным торможением. На рисунках 4.5, 4.6 и 4.7 изображены, соответственно, сеть без обратных связей, рекуррентная сеть и сеть с латеральным торможением.

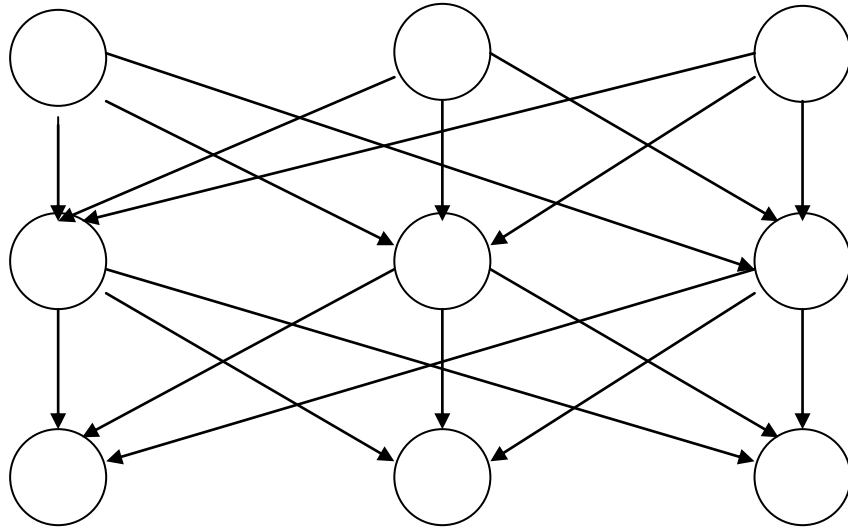


Рис.4.5. Сеть без обратных связей

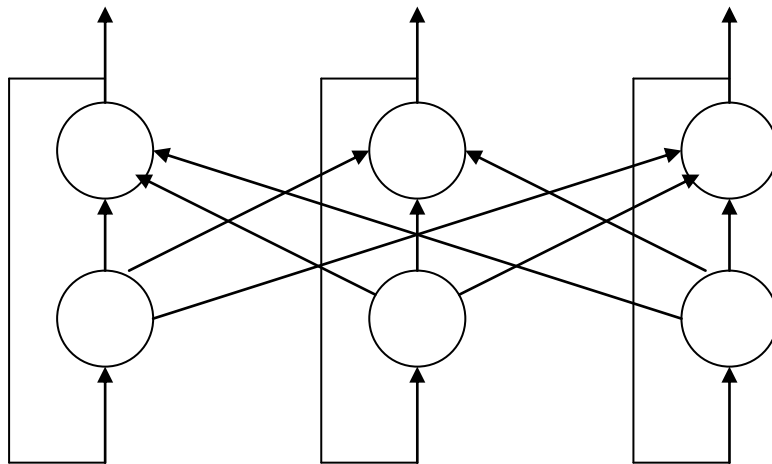


Рис.4.6. Рекуррентная сеть

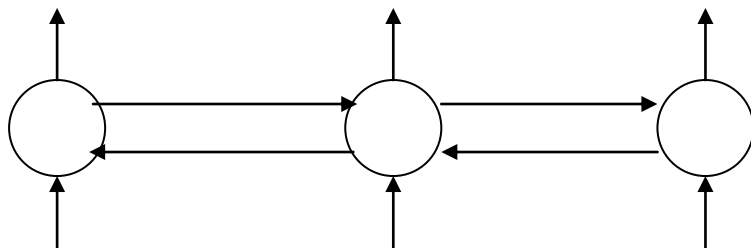


Рис 4.7. Сеть с латеральным торможением

**Однослойные сети искусственных нейронов.** Простой однослойной сетью называется иерархическая нерекуррентная сеть содержащая кроме слоя входных нейронов, предназначенных для распределения сигналов, один слой вычислительных нейронов, выходной сигнал каждого из которых определяется как функция от взвешенной суммы сигналов, поступающих на его вход. В простейшем случае выход – это простая взвешенная сумма сигналов, поступающих на вход каждого вычислительного нейрона. Совокупность выходных сигналов образует выходной вектор сети  $Y$ , размерность  $m$  которого равна количеству выходов сети. Если аналогичным образом определить входной вектор  $X$ , размерности  $n$ , и матрицу весовых коэффициентов  $W$ , размерности  $n \cdot m$ , мы получим зависимость выхода сети от ее входа в следующем векторном виде

$$Y = XW.$$

Можно показать, что вычислительная мощность подобной сети использующей линейную функцию активации, не увеличивается при введении новых слоев. Покажем это. Рассмотрим двухслойную сеть с матрицами весовых коэффициентов  $w_1$  и  $w_2$ , соответственно, для первого и второго слоев. Выходной вектор нейронов первого слоя определяется следующим образом

$$Y_1 = XW_1,$$

а второго слоя – по формуле

$$Y_2 = Y_1W_2 = XW_1W_2 = XW.$$

где  $W = W_1W_2$  - матрица весовых коэффициентов однослойной сети, эквивалентной исходной.



Аналогичным способом любую многослойную сеть с линейной функцией активации можно свести к эквивалентной однослойной.

Выше, при указании количества слоев сети, мы не учитывали входной слой, играющий роль распределителя. В дальнейшем, при указании слойности, мы будем принимать во внимание все слои, включая и входной.

**Многослойные сети искусственных нейронов.** Увеличение числа слоев в сети при использовании нелинейной функции активации позволяет повысить ее вычислительную мощность, т.е. обеспечить возможность построения более сложных отображений (функций).

Простые сети, не включающие скрытых слоев не могут решать ряд задач. Примером такой задачи служит всем хорошо известная проблема «исключающего или», которая не может быть решена простой сетью, не включающей вспомогательные слои. Для того, чтобы понять в чем именно заключается проблема, рассмотрим более подробно функцию нейрона. Рассмотрим модель нейрона, активация которого происходит согласно пороговой функции. Пусть нейрон имеет  $n$  входов. Рассмотрим  $n$ -мерное пространство координат, в котором каждое измерение соответствует одному из входов нейрона. Пусть дано уравнение

$$\sum_{i=1}^n w_i x_i - \theta = 0. \quad (4.1)$$

Как видно, уравнение (4.1) определяет гиперплоскость в  $n$ -мерном пространстве. Эта гиперплоскость делит пространство на два подпространства. Для одного из них будет выполняться неравенство

$$\sum_{i=1}^n w_i x_i - \theta > 0,$$

в то время, как для другого будет истинно

$$\sum_{i=1}^n w_i x_i - \theta < 0.$$

Допустим, что функция активации нейрона имеет вид (пороговая функция)

$$y = \begin{cases} 1, & \text{если } \sum_{i=1}^n w_i x_i - \theta \geq 0, \\ 0, & \text{иначе.} \end{cases}$$

Тогда можно сделать заключение, что нейрон сепарирует (разделяет) пространство входных сигналов на две части, при этом определенные комбинации входов вызывает активацию нейрона, в то время как при других комбинациях нейрон не активируется. Отсюда можно прийти к выводу, что двухслойная нейронная сеть может решать только линейно сепарабельные задачи.

### **Детерминированные нейронные сети.**

Детерминированными называются нейронные сети, в которых функция активации элементов детерминированная. В качестве функции активации используются различные функции. Перечислим основные виды функции активации наиболее широко используемые в исследованиях.

#### **1. Линейная функция**

$$y = k \sum_i x_i w_i.$$

График линейной функции приведен на рис.4.8. При  $k=1$  на выходы нейронов просто подаются их общие взвешенные входы. Сфера применения этой функции весьма ограничена. Построение многослойных сетей на основе этой функции не приводит к увеличению вычислительной мощности, так как любую многослойную сеть с линейными элементами можно свести к однослойной.

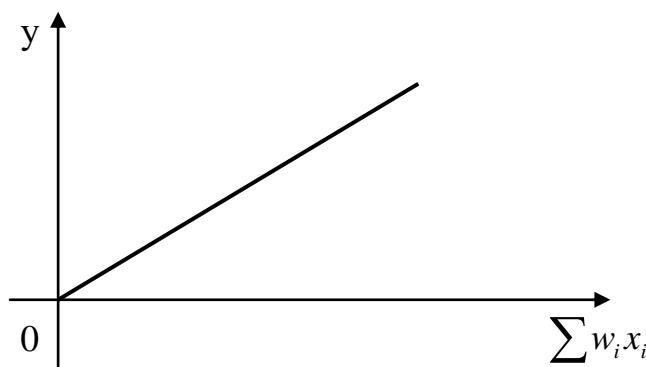


Рис.4.8. Линейная функция

## 2. Пороговая функция

$$y = \begin{cases} 1, & \text{если } \sum_{i=1}^n w_i x_i - \theta \geq 0, \\ 0, & \text{иначе.} \end{cases}$$

где  $\theta$  - порог.

График этой функции представлен на рис. 4.9. Нейроны, использующие пороговую функцию мгновенно изменяют свое состояние с «0» на «1» при достижении значения общего взвешенного входа величины порога.

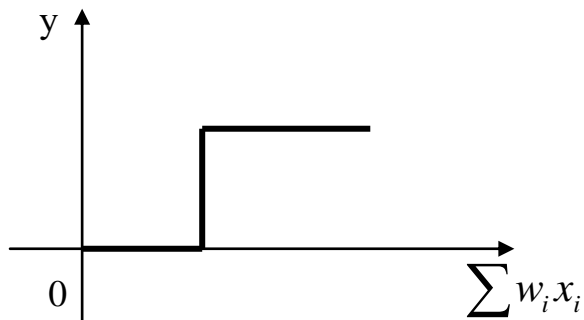


Рис. 4.9. Пороговая функция

## 3. Сигмоидная (полулинейная функция)

$$y = \frac{1}{1 + e^{-\sum w_i x_i}}.$$

График функции активации приведен на рис. 4.10. Это наиболее распространенная форма задания функции активации. При стремлении общего входа к  $-\infty$  уровень активности стремится к нулю, при очень больших значениях общего входа, уровень активности практически равен «1».

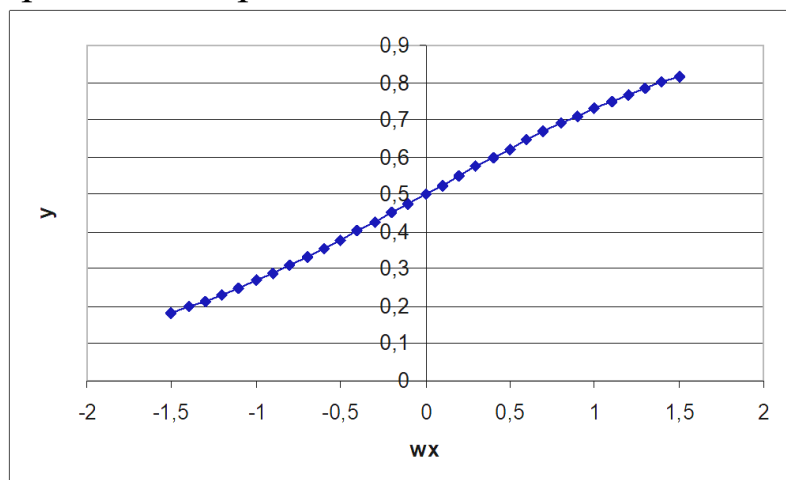


Рис. 4.10. Сигмоидная функция

#### 4. Гиперболический тангенс

$$y = th\left(\sum_i w_i x_i\right).$$

На рис. 4.11 – приведен график этой функции. В отличие от сигмоидной функции, при использовании гиперболического тангенса уровень активности нейронов колеблется от -1 до 1.

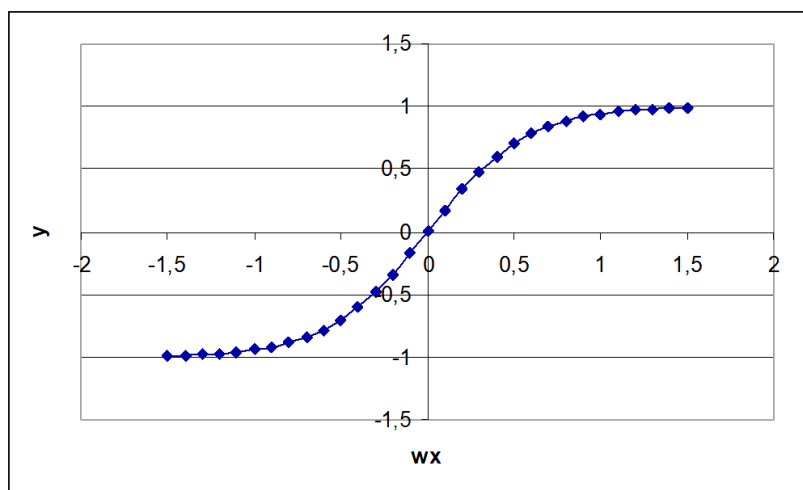


Рис. 4.11. Гиперболический тангенс

Последние три функции представляют особый интерес для построения многослойных нейронных сетей, способных решать следственные задачи. Области значений этих функций ( $[0,1]$  или  $[-1,1]$ ) открывают широкие возможности для проведения классификации выходных векторов. Как уже отмечалось, линейная функция используется гораздо реже. Детерминированные нейронные сети могут быть как без обратных связей (feed-forward) так и содержать такие связи (feed-back). Сети, содержащие обратные связи представляют огромный интерес для исследователей в области нейронных сетей. Такие сети называются рекуррентными. В нерекуррентных сетях, как простых, так и многослойных, при подаче входного вектора по сети протекает поток активации нейронов в направлении от входного слоя к выходному. После того, как определяются уровни активностей выходных нейронов, никаких изменений состояний нейронов не

производится. Этот процесс называется релаксацией нейронной сети.

Совсем иначе ведут себя рекуррентные сети. При подаче на вход сети вектора, определяются состояния нейронов, но затем, из-за того, что выходы нейронов имеют обратные связи, на их входы опять поступает новый вектор, и состояния снова изменяются. С рекуррентными сетями связано понятие стабильности. Сеть считается стабильной, если после конечного числа итераций нейроны принимают состояния, которые в дальнейшем не изменяются. При подаче вектора на вход стабильных рекуррентных сетей, вырабатываются выходные сигналы нейронов, которые затем опять поступают на входы, снова генерируя новый вектор состояний, но, по мере роста числа итераций, количество изменений состояний узлов уменьшается, пока сеть не установится в конечное состояние. Сети без обратных связей всегда стабильные, так как при подаче одного вектора на вход, узлы сети только один раз могут изменять свое состояние, вследствие постоянства входов нейронов.

Нестабильные сети не достигают конечного состояния за конечное число итераций. В них, при подаче входного вектора вырабатываются выходные сигналы, которые подаются опять на входы и таким образом, вызывают новые изменения состояния, причем, этот процесс продолжается бесконечно, без установления в какое-то неизменное состояние.

Релаксация в нестабильных нейронных сетях, называемая также хаотической релаксацией, представляет большой интерес для исследователей, но проблемы хаотической релаксации не рассматриваются в данной книге.

Долгое время вопрос о том, по каким признакам следует судить о стабильности сети, оставался открытым. Это сдерживало развитие работ в этом направлении. Однако, благодаря доказательству теоремы определяющей подмножество рекуррентных сетей, которые являются стабильными [37] этот вопрос решен, и сегодня многие научные разработчики заняты исследованиями сложных рекуррентных сетей.

Одним из крупнейших исследователей нейронных сетей является Хопфильд, предложивший особый вид сетей, которые так и называются – сети Хопфильда. Рассмотрим простую

двухслойную сеть, представленную на рис.4.12. На вход каждого нейрона, помимо соответствующего компонента входного вектора  $X$ , поступают выходные сигналы от других нейронов через нейроны-распределители первого слоя. В своих ранних работах [38] Хопфильд использовал пороговую функцию в качестве функции активации, которая записывалась, однако, не сколько иначе, чем приведенная выше.

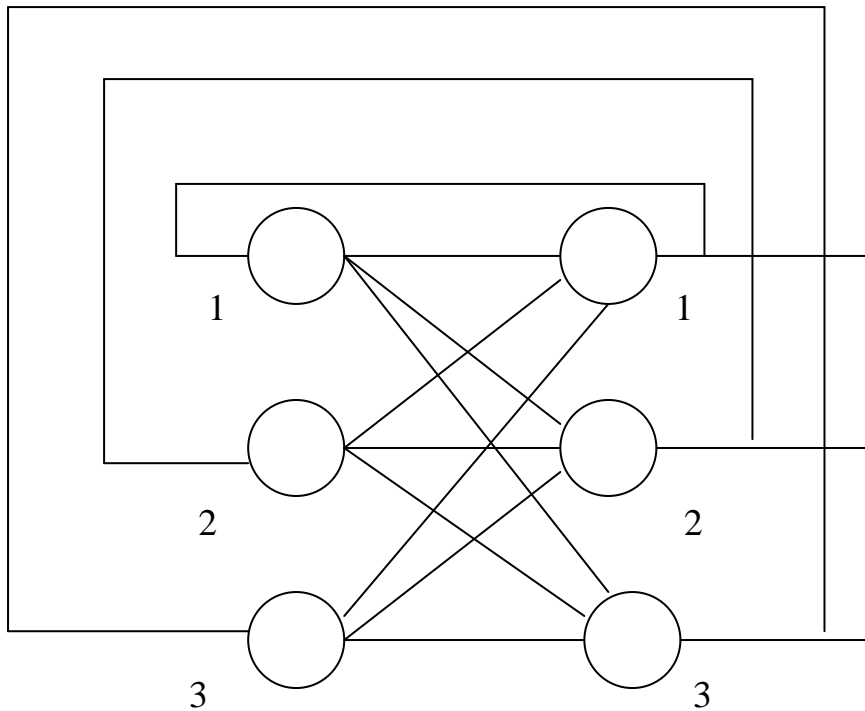


Рис.4.12. Простая рекуррентная сеть

$$y_j = \begin{cases} 1, & \text{если } \sum_{j \neq i} w_{ji} y_i + x_j > \theta_j, \\ 0, & \text{если } \sum_{j \neq i} w_{ji} y_i + x_j < \theta_j, \\ \text{не изменяется,} & \text{если } \sum_{j \neq i} w_{ji} y_i + x_j = \theta_j. \end{cases} \quad (4.2)$$

Из выражения (4.2) видно, что нейрон переходит в активное состояние, если взвешенная сумма выходов других нейронов плюс  $j$ -я компонента вектора  $X$  превышает порог; переходит в состояние Щ, если эта сумма меньше порога; и не изменяется, если она равна порогу. В первых работах Хопфильда состояние каждого нейрона

изменялось в дискретные случайные моменты времени, т.е. асинхронно. Совокупность выходных сигналов формирует двоичный вектор состояния сети.

Таким образом, нейронная сеть, имеющая  $n$  выходов, принимающих двоичные значения, может находиться в  $2^n$  состояниях. Каждое из этих состояний представляет собой одну из вершин гиперкуба в  $n$ -мерном пространстве, если по осям координат откладывать значения уровней активности нейронов. При подаче входного вектора сеть переходит из состояния в состояние, т.е. передвигается по вершинам гиперкуба пока не достигнет конечного состояния. Стабильным состояниям соответствуют определенные значения входных векторов, весовых коэффициентов и порогов. Так как в одной сети весовые коэффициенты и значения порогов остаются постоянными (режим обучения здесь не рассматривается), то стабильные состояния полностью определяются текущими входными векторами. На рисунке 4.13 приведен куб для сети с тремя выходными нейронами.

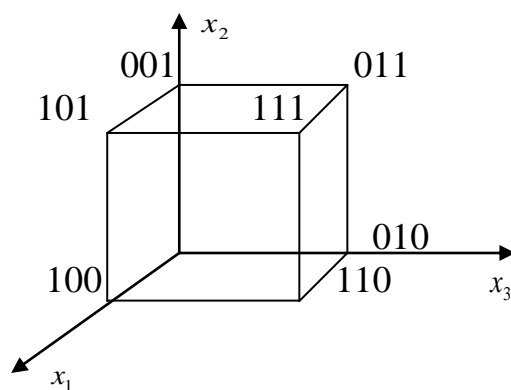


Рис.4.13. Куб состояний для сети с тремя выходными узлами

Кохонен и Гроссберг [39] показали, что если матрица весов рекуррентной сети симметрична относительно главной диагонали, содержащей нули, то эта сеть стабильна. Это утверждение доказывается следующим образом. Допустим, что имеется функция, уменьшающая свое значение при каждом изменении состояния сети. Очевидно, эта функция должна убывать до тех пор, пока не дойдет до нуля. В качестве самой функции можно взять следующую функцию Ляпунова

$$E = -\frac{1}{2} \left( \sum_i \sum_j w_{ij} y_i y_j - \sum_j x_j y_j + \sum_j \theta_j y_j \right). \quad (4.3)$$

Здесь  $E$  – так называемая «энергия» сети:  $x_j$  -  $j$ -ая компонента входного вектора, подаваемая извне;  $\theta_j$  - порог нейрона  $j$ ;

Учитывая выражение (4.3) запишем формулу для изменения энергии, вызванного изменением состояния нейрона  $j$

$$\Delta E = - \left[ \sum_{i \neq j} (W_{ij} Y_i) + X_j - \theta_j \right] \Delta y_j. \quad (4.4)$$

Рассмотрим выражение (4.4). Если общий взвешенный вход нейрона превосходит величину порога, то он переключается в активное состояние, а это значит что при положительном значении выражение в квадратных скобках  $y_j$  принимает значение «1». Таким образом, если до этого его значение было 0, то величина приращения  $\Delta y_j = 1$ , если же нейрон был и до этого активен, то его состояние не меняется и  $\Delta y_j = 0$ . Это означает, что величина  $\Delta E$  либо будет отрицательной, либо не изменится.

Если общий вход меньше значения порога, то нейрон принимает состояние «0», что означает, что величина  $\Delta y_j$  будет либо «0», либо «-1». Все выражение (4.4) в этом случае также будет либо отрицательным, либо равным нулю.

В случае равенства общего входа порогу оба множителя в (4.4) будут нулевыми, следовательно и  $\Delta E$  будет равно нулю.

Таким образом, при любых изменениях состояний нейронов энергия сети будет либо уменьшаться, либо не изменяться и при конечном числе обновлений состояний она будет сходиться к минимуму и далее оставаться неизменной, что и является признаком стабильности сети.

Таким образом, при симметричной матрице весов сети, ее поведение будет стабильным. Необходимо отметить, что симметричность матрицы весов является достаточным, но не необходимым условием.

Рекуррентные сети могут служить в качестве моделей ассоциативной памяти. Между человеческой памятью и памятью традиционного компьютера есть существенная разница. Для считывания информации из машинной памяти требуется указать адрес, по которому она расположена. Человеческая память



устроена несколько иначе. Для извлечения из нее информации требуется некоторая ключевая информация по содержанию. Например, если человек услышит название знакомой ему книги, он может вспомнить ее авторов, содержание, цвет обложки и т.д. Знатоки поэзии, по первым строкам стихотворения могут вспомнить полностью, знатоки музыки могут восстановить в памяти все произведение по фрагменту и т.д.

Для того, чтобы моделировать ассоциативную память на рекуррентных сетях, надо определить их целевые состояния и обеспечить минимизацию энергии в этих точках пространства состояний. Если эти условия будут выполнены, то при подаче части полного образа, или его зашумленного (испорченного) варианта, сеть, с высокой степенью вероятности, будет воспроизводить полный образ. Эта возможность обеспечиваемая рекуррентными сетями, может сыграть значительную роль в разработке методов восстановления информации.

В работе [39] Хопфильд предложил следующую идею для разработки ассоциативной памяти. Пусть необходимо запомнить образы  $Y^1, Y^2, \dots, Y^m$ , которые представляют собой бинарные векторы. Тогда для создания ассоциативной памяти на основе рекуррентной сети, которая бы «помнила» эти образы, ее весовые коэффициенты можно рассчитать по следующей формуле

$$W_{ij} = \sum_{k=1}^m y_i^k y_j^k,$$

где  $y_i^k$  -  $i$ -й компонент  $k$ -го вектора.

Таким образом, после определения весовых коэффициентов (заметьте, что в этом случае  $W_{ij} = W_{ji}$  - что обеспечивает стабильность сети) сеть может использоваться в качестве ассоциативной памяти. Если теперь на ее вход подать вектор  $y^j$ , представляющий собой неполной или испорченный вариант  $y^j$ , сеть с высокой степенью вероятности восстановит оригинал. Это происходит потому, что после подачи входного вектора сеть начинает «расслабляться», т.е. постепенно уменьшать свои энергии в сторону ближайшего минимума. Не исключен вариант, однако, когда этот минимум окажется локальным, а не глобальным.

До сих пор мы рассматривали рекуррентные сети с пороговыми элементами. Сигмоидная функция является более

правдоподобной биологически, чем пороговая, так как для перехода нейрона из одного состояния в другое требуется некоторое время. Хопфильд исследовал также сети, использующие сигмоидную функцию [4], имеющий вид

$$y = 1/(1 + e^{-\lambda \sum x_i w_i}). \quad (4.5)$$

Здесь, коэффициент  $\lambda$  указывает на крутизну функции. При стремлении  $\lambda$  к бесконечности функция (4.5) приближается к пороговой, при уменьшении  $\lambda$ , кривая функции сглаживается.

Для таких сетей также выполняется достаточное условие их стабильности, т.е. условие симметричности матрицы весовых коэффициентов.

**Стохастические нейронные сети.** В реальном мире приходится часто сталкиваться со стохастическими процессами. Трудность, а подчас и невозможность учета всех факторов, оказывающих влияние на поведение реальной системы вынуждает рассматривать такие системы как стохастические. Наиболее известным классом стохастических нейронных сетей являются, так называемые машины Больцмана [4,5,6]. Они строятся по принципу сетей Хопфильда, т.е. в них используется симметричная матрица весов, с той лишь разницей, что правило активации нейронов здесь стохастическое, а не детерминированное. Это означает, что вместо функции, непосредственно определяющей уровень выходного сигнала нейрона при заданных входных сигналах, здесь используется соотношение, определяющее вероятность срабатывания нейрона.

В начале раздела 4.2. было дано описание сетей Хопфильда. С этими сетями было связано понятие энергии и стабильности. Рассмотрим следующее соотношение, определяющее энергию сети

$$E = \sum_{i < j} w_{ij} y_i y_j + \sum_i \theta_i y_i. \quad (4.6)$$

Здесь  $y_i$  и  $y_j$  - уровни активности  $i$ -го и  $j$ -го нейронов, соответственно;  $\theta_i$  - порог  $i$ -го нейрона. Срабатывание нейрона приводит к уменьшению общей энергии сети на величину

$$\Delta E_k = \sum_i w_{ki} - \theta_k. \quad (4.7)$$

В детерминированных сетях Хопфильда этот факт может использоваться при построении ассоциативной памяти. В этом случае для каждого частного вектора из обучающего множества

(т.е. для каждой отдельной памяти) обеспечивается достижение минимума энергии  $E$ . Как отмечалось выше, при поступлении на вход сети внешнего вектора, который может быть зашумленным или же неполным, такая сеть может постепенно снижать свою энергию, пока не достигнет локального минимума и не установится в стабильном состоянии. Указанный подход не гарантирует глобального минимума, поэтому Экли и его сотрудники [5] ввели элемент случайности в поведение таких сетей. Как видно из соотношения (4.7), величина  $\Delta E_k$  представляет собой не что иное, как общий взвешенный вход сети. В стохастических сетях правило активации задается некоторой функцией плотности распределения вероятности  $f$

$$p_k = f(\Delta E_k) = f\left(\sum_i w_{ki} y_i - \theta_k\right),$$

где  $p_k$  - вероятность срабатывания  $k$ -го нейрона.

В качестве функции  $f$  не может использоваться пороговая функция, т.к. в этом случае сеть становится детерминированной. Также неприемлемо использовать линейную функцию и гиперболический тангенс, т.к. области их значений выходят за границы диапазона  $[0,1]$ , в котором может находиться значение вероятности. Поэтому, в качестве функции плотности распределения вероятности часто используется сигмоидная функция

$$p_k = \frac{1}{1 + e^{-\Delta E_k / T}}, \quad (4.8)$$

где  $T$  – некоторый параметр, определяющий уровень случайности в сети. Параметр  $T$  называется «температурой», это вызвано тем, что можно провести параллель между машиной Больцмана и термодинамической системой.

Рассмотрим процесс закаливания металлов. При низких температурах атомы в металлах находятся в узлах так называемой кристаллической решетки. Они жестко связаны и их энергия не позволяет покинуть позицию, задаваемую решеткой. При нагревании металла энергия атомов увеличивается и амплитуда их колебаний около позиций, определенных узлами решетки возрастает. При достижении температуры выше точки плавления структура решетки разрушается и атомы начинают свободное

движение, обладая высокими энергиями. При понижении температуры атомы снова теряют энергию, образуя конфигурацию, соответствующую минимуму энергии.

Подобный процесс имитируется в машинах Больцмана, поэтому он называется «имитируемым закаливанием» (simulated annealing). При достаточном числе обновлений состояний нейронов в сети, т.е. при итеративном применении функции (4.8), отношение вероятностей нахождения сети в двух глобальных состояниях А и В (здесь под глобальным состоянием имеется в виду совокупность состояний элементов сети) остается постоянной

$$\frac{P_A}{P_B} = e^{-(E_A - E_B)/T}.$$

В таком случае говорят, что если достигла «температурного равновесия». Очевидно, что при температурном равновесии элементы сети могут еще изменять свои состояния, т.к. функция активации – плотность распределения вероятности, которая принимает значения 0 и 1 только при устремлении общего взвешенного входа к  $-\infty$  и к  $+\infty$ , соответственно, т.е.

$$\lim_{\Delta E_k \rightarrow -\infty} p_k = 0, \quad \lim_{\Delta E_k \rightarrow +\infty} p_k = 1.$$

Вероятность нахождения сети в каком либо глобальном состоянии А задается формулой

$$P_A = e^{-E_A/T}, \quad (4.9)$$

представляющей собой распределение Больцмана, откуда и было взято название «машины Больцмана». Из соотношения (4.9) видно, что при высоких температурах вероятность нахождения сети в состояниях с низкой энергией низка, в этом случае сеть скорее всего может находиться в состояниях с высокой энергией. При понижении температуры наблюдается обратная тенденция: состояния с низкой энергией гораздо более вероятны, чем состояния с высокой энергией. Казалось бы, имеет смысл применять низкие температуры, чтобы сеть достигала низкоэнергичных состояний (напомним, что задачей является достижение минимума энергии). Однако, при низких температурах сеть медленнее достигает температурного равновесия. С другой стороны, при высоких температурах равновесие достигается быстрее. Компромиссным решением вышеуказанной проблемы

является задание сначала высокой температуры, а потом ее постепенное снижение. Таким образом, сеть как бы закаливается так же, как и металлы. «Имитируемое закаливание» позволяет сети достичь состояние с низкой энергией при относительно небольшом количестве итераций.

Для небольших задач сеть может предлагать достаточно приемлемое решение (т.е. при задании входного вектора будет генерироваться соответствующий выходной с достаточно высокой точностью). Однако, для более сложных задач становится проблематичным применение простых однослойных стохастических сетей. Эта проблема может быть решена за счет введения дополнительных скрытых слоев, обеспечивающих повышение вычислительной мощности сети а, следовательно, и точности решения задачи.

**Нечеткие нейронные сети.** Обычные, или «четкие» нейроны, которые мы до сих пор рассматривали обобщенно можно описать следующим образом. Искусственные нейроны – это процессорные элементы, имеющее несколько входов ( $n$ ,  $n > 1$ ) и один выход. Каждой входной связи поставлена в соответствие определенная функция  $f_i(x_i)$ , которая каким-то образом обрабатывает входное значение. В простейшем и наиболее распространенном случае она усиливает его, умножая на весовой коэффициент связи  $w_i$ , т.е.  $f_i(x_i) = x_i \cdot w_i$ . Выходное значение нейрона формируется  $n$ -мерной функцией  $g(y_1, y_2, \dots, y_n)$ , входными параметрами которой являются значения выданные функциями  $f_i$ . В качестве функции  $g$  наиболее часто используется функция  $g(I) = 1/(1 + e^{-I})$ , часто называемая сигмоидной, где  $I$  – это сумма значений функций  $f$  для всех связей с некоторой константой, называемой обратным порогом или смещением (bias), т.е.

$$I = \sum_i f_i(x_i) + \theta.$$

В нечетком нейроне используются понятия: нечеткое число, функция принадлежности, нечеткие операции. В этом случае каждая из функций  $f$  и  $g$  может оперировать нечеткими числами, используя их как в качестве аргументов, так и выходных значений.

### 4.3. Характеристики и вычислительные способности нейронных сетей

Человеческий мозг содержит огромное количество нейронов (около 14 млрд.) объединенных в единую сеть посредством синаптических связей. Все процессы восприятия, распознавания и мышления человека реализуется за счет протекания электрических сигналов в этой сети. Реальный процесс обработки сигналов в биологических нейронах гораздо сложнее описанного в предыдущем разделе и недостаточно изучен. Срабатывание нейронов происходит асинхронно в реальном времени. Активности нейронов меняются со временем. Сила синаптических связей также меняется, но гораздо медленнее. Мы думаем гораздо быстрее, чем учимся [40]. Если обозначить состояние  $i$ -го нейрона через  $x_i$ , а  $j$ -й вес связи через  $w_j$ , то можно записать

$$\frac{dw_j}{dt} \ll \frac{dx_i}{dt} .$$

Изменение активности нейронов зависит от величин сигналов поступающих на нейрон, а также от длительности их воздействия. Особенный интерес представляет динамика весов связей. Хебб [41] высказал идею об изменении весовых коэффициентов, которая считается биологически правдоподобной. Он отметил, что если два нейрона, связанные между собой активируются одновременно, то число узлов, связывающих аксон одного нейрона с телом (или дендритом) другого увеличиваются размеры этих узлов. И в этом и в другом случае это означает усиление взвешенной связи между двумя нейронами. На этой идее основан алгоритм обучения Хебба, который является исторически первым алгоритмом обучения нейронных сетей.

Искусственные нейронные сети состоят из множества искусственных нейронов, объединенных в определенную структуру. Различные типы нейросетевых архитектур описаны в разделе 4.2.

Одним из основных свойств является параллельность обработки информации в сетях. Каждый из нейронов является отдельным вычислительным устройством, работающим параллельно с некоторой группой других элементов. Каждый

нейрон приносит свою долю вычислений в выработке конечного вектора выходных сигналов сети. Это определяет высокую скорость нейронных вычислений при их реализации на параллельной аппаратуре.

Из принципа параллелизма следует распределенность представления информации в нейронных сетях. Если задать кому-либо вопрос «сколько чисел можно представить с помощью пальцев одной руки» то, скорее всего, он ответит «пять». Однако, с помощью комбинации загнутых и незагнутых пальцев можно представить 32 числа. Дело в том, что в первом случае используется одинарная система счисления, а во втором – двоичная. В одинарной системе счисления используется локальное представление, а в системе с основной и (при  $n > 1$ ) используется распределенное представление [2]. При распределенном представлении каждый элемент участвует в представлении каждого образа.

Надежность нейросетевых моделей является еще одним ценным свойством. Здесь надежность можно понимать двояко. С одной стороны при поступлении на вход сети зашумленного или неполного образа сеть может вести себя так, как если бы поступил оригинальный вариант образа. С другой стороны, при выходе из строя какой-то связи или даже целого нейрона, качество поведения сети понижается незначительно. Это свойство может позволить строить прикладные нейросетевые системы для областей, где требуется высокая надежность.

Не менее важным качеством является обучаемость нейронных сетей. Сеть не только способна выполнять функции автомата, отображая множество входных векторов, но и модифицировать свои параметры (структуру, весовые коэффициенты) таким образом, чтобы адаптировать свое поведение к требованиям конкретной задачи. Например, в супервизорных алгоритмах требуется обеспечить отображение множества входных векторов на множество выходных. При этом параметры сети подбираются соответствующим образом, в зависимости от конкретного алгоритма. Это качество позволяет вычислять параметры сети, не обладая значительными знаниями о предметной области. Например, при построении нейронных экспертных систем отпадает необходимость в трудоемком процессе работы с экспертами (те,

кто знакомы с проблемами построения экспертных систем, знают, что приобретение знаний – наиболее сложный этап в их разработке), при условии наличия достаточного статистического материала.

Способность нейронных сетей к обобщению – также весьма важное свойство. Благодаря этому свойству сети способны не только воспроизводить отображения, задаваемые в ходе обучения, но и строить новые. Это повышает «компетентность» систем, основанных на нейронных сетях и позволяет им играть роль аппроксиматора функций.

**Задачи, решаемые нейронными сетями.** Перейдем теперь к рассмотрению некоторых типов задач, для решения которых может быть эффективно применение парадигмы нейронных сетей. Среди таких задач можно условно выделить следующие:

- классификация;
- кластеризация;
- аппроксимация;
- автоассоциация.

**1) Классификация.** В задаче классификации задается вектор признаков объекта  $X^n = \{x_1, x_2, \dots, x_n\}$  и требуется отнести объект к одному из  $m$  непересекающихся классов  $C_i$

$$C_i \cap C_j = \emptyset, \quad i \neq j.$$

Например, при классификации летающих объектов, признаками могут служить наличие крыльев, двигателя, оперения и т.д. Классами здесь могут быть Самолет, Птица, Ракета, Неопознанный Летающий Объект и т.п. Каждая комбинация конкретных значений признаков задает вектор. При поступлении на вход классифицирующей системы этого вектора она должна указать класс, к которому принадлежит объект.

В терминах нейронной сети это означает следующее. Строится сеть, содержащая  $n$  входных и  $m$  выходных нейронов. Типы архитектур сети подробно рассмотрены в разделе 4.2. При подаче на входные элементы вектора признаков, выходной элемент с наивысшим уровнем активности указывает класс объекта. Для обеспечения такой возможности проводится обучение сети с выбором подходящего алгоритма.



**2) Кластеризация.** В задаче кластерного анализа требуется произвести разбиение входного множества векторов признаков на так называемые кластеры таким образом, чтобы векторы, относящиеся к одному и тому же кластеру были близки друг к другу, в то время как векторы из разных кластеров были бы далеки друг от друга.

Для решения такой задачи строится сеть с числом входных нейронов равных количеству компонент вектора признаков и количеством выходных элементов равным числу кластеров. Векторы весовых коэффициентов выходных нейронов подбираются процедурой обучения близкими к центрам соответствующих кластеров.

Алгоритм конкурентного обучения, обсуждаемый в разделе 4.4. позволяет решать эту задачу.

**3) Аппроксимация.** Нейронные сети могут служить универсальными аппроксиматорами функций. Допустим, мы имеем некоторую функцию  $F(X)$ , чтобы выполнялось соотношение

$$d(F(X), F''(X)) < \varepsilon$$

для заданного малого  $\varepsilon$ , где  $d()$  – расстояние между функциями.

В общем случае нам неизвестен вид функции  $F()$ , которую необходимо аппроксимировать. Нам задано лишь множество пар вида  $X_1 \rightarrow Y_1, \dots, X_n \rightarrow Y_n$ , где  $X_i$  -  $i$ -е наблюдение вектора входных переменных функции, а  $Y_i$  -  $i$ -е наблюдение значения функции. При использовании классических методов, например метода наименьших квадратов, нам необходимо выбрать модель аппроксимирующей функции (линейную, квадратичную и т.д.) и потом вести оценивание (идентификацию) параметров выбранной модели.

Нейронные сети являются универсальными аппроксиматорами, т.е. они могут аппроксимировать любую функцию с любой точностью  $\varepsilon$  при задании достаточного количества нейронов [40]. Причем отпадает необходимость в выборе вида модели.

При нейронной аппроксимации рассматриваются пары наблюдаемых значений переменных и значений функций для обучения сети. После обучения сеть может выполнять рассмотренные отображения с высокой степенью точности.

Комбинация переменных задают точки в  $N$ -мерном пространстве, где  $N$  – размерность вектора переменных функции. Каждая из таких точек связана с точкой в одномерном пространстве значений функции. При выборе точки в окрестности некоторой  $X_i$  сеть определяет точку в окрестности точки  $Y_i$  (благодаря способности нейронных сетей к обобщению), причем, чем меньше эта окрестность, тем точнее аппроксимация. Таким образом, чем больше выбирается пар типа  $(X, Y)$ , тем точнее аппроксимация.

На практике это позволяет проводить идентификацию объектов, для которых представляется сложным строить точное математическое функциональное описание.

4) **Автоассоциация.** Решение этого типа задач связано с необходимостью разработки моделей ассоциативной памяти.

Память компьютера и человеческая память организуется по различным принципам. Для получения доступа к интересующей нас информации необходимо указать адрес записи этой информации в памяти компьютера.

Существует иная модель, называемая ассоциативной памятью или памятью, адресуемой по содержанию. Для извлечения информации из этой памяти нужно задать часть запомненного образа, или его зашумленный вариант. При этом будет происходить регенерация искомого образа.

В нейронной модели ассоциативной памяти, с помощью соответствующих процедур обеспечивается запоминание образа группой нейронов. При подаче на вход сети части образа сеть будет выдавать полный корректный образ. Время извлечения образа не зависит от количества запомненных образов [40].

#### 4.4. Искусственные нейронные сети. Персептрон

Понятие «персептрон» впервые ввел американский нейрофизиолог Френк Розенблатт (Rosenblatt) в 1957 году. Персептрон является некоторым классом моделей мозга или отдельной его системы (например, зрительной).

Схематично его устройство показано на рис. 4.14.

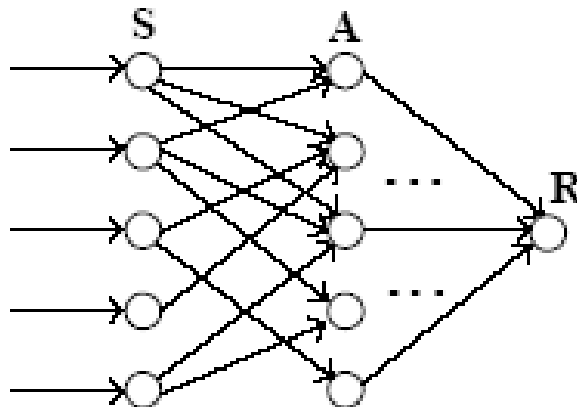


Рис. 4.14. Схематичный вид персептрона

Здесь  $S$  – набор чувствительных сенсорных элементов (сетчатка),  $A$  – набор ассоциирующих элементов (нейронов),  $R$  – реагирующий элемент (т.е. нейрон, передающий сигнал управления мышцам или железам). Нейрон имеет много входов и один выход. Входы в нейрон подразделяются на тормозящие и возбуждающие. Сенсорные элементы возбуждаются, если в результате воздействия раздражителя (например, света) величина входного сигнала окажется больше некоторого порогового значения.  $S$ -элементы случайным образом связаны с  $A$ -нейронами. При этом если число возбуждающих сигналов больше, чем тормозящих, то нейрон возбуждается и посылает сигнал на реагирующий элемент. В отличие от нейронов сигналы, поступающие на реагирующий элемент, суммируются с некоторыми весами. Реагирующий элемент выбирает некоторое действие, если

$$R(x) = \sum_{i=0}^n w_i x_i = (w, x) > 0,$$

где  $w = (w_0, w_1, \dots, w_n)$ ,  $x = (1, x_1, \dots, x_n)$ ,  $x_i$  – сигнал, поступающий на реагирующий элемент от  $i$ -го нейрона ( $x_0 \equiv 1$  – сигнал смещения).

С помощью персептрона можно осуществлять классификацию объектов по двум классам. Полагаем, что  $x \in \varpi_1$ , если  $R(x) > 0$ , и  $x \in \varpi_2$  – в противном случае. Функция  $R(x) = (w, x)$  будет линейной решающей функцией (ЛРФ), а гиперплоскость  $(w, x) = 0$  – линейной разделяющей поверхностью.

Предположим, что имеется некоторая обучающая выборка  $\{x_1, \dots, x_n\}$ , причем  $\{x_1, \dots, x_m\} \subset \varpi_1$  и  $\{x_{m+1}, \dots, x_n\} \subset \varpi_2$ . Требуется построить линейную решающую функцию  $R(x) = (w, x)$ , которая бы правильно разделяла элементы обучающей выборки, т.е.

$$\begin{aligned} (w, x_i) &> 0 \quad \forall i = 1, \dots, m; \\ (w, x_i) &< 0 \quad \forall i = m + 1, \dots, n; \end{aligned} \quad (4.10)$$

Построение такой ЛРФ осуществим с помощью итерационного алгоритма. Для этого обучающую выборку запишем в виде бесконечной циклической последовательности  $\{x_1, \dots, x_n, x_1, \dots, x_n, \dots\}$  и выберем некоторое начальное значение весов  $w^{(1)} = (w_1^{(1)}, \dots, w_n^{(1)})$ . Далее на каждом  $k$ -м шаге алгоритма «предъявляется»  $k$ -й вектор  $x_k$  обучающей выборки и значение весового вектора  $w^{(k)}$  корректируется или не корректируется в соответствии с правилом

$$w^{k+1} \begin{cases} w^k, & \text{если } (w^{(k)}, x_k) > 0 \text{ и } x_k \in \varpi_1; \\ w^k, & \text{если } (w^{(k)}, x_k) < 0 \text{ и } x_k \in \varpi_2; \\ w^k + x_k, & \text{если } (w^{(k)}, x_k) \leq 0 \text{ и } x_k \in \varpi_1; \\ w^k - x_k, & \text{если } (w^{(k)}, x_k) \geq 0 \text{ и } x_k \in \varpi_2. \end{cases} \quad (4.11)$$

То есть весовой вектор не меняется, если «предъявленный» вектор классифицируется правильно и увеличивается или уменьшается на  $x_k$  при неправильной классификации. Алгоритм завершает свою работу, если осуществляется  $n$ -кратная правильная классификация образов обучающей выборки.

Для упрощения алгоритма обучения персептрона вместо обучающей выборки  $\{x_1, \dots, x_n\}$  рассмотрим выборку  $\{y_1, \dots, y_n\}$ , где  $y_i = x_i$  для  $i = 1, \dots, m$  и  $y_i = -x_i$  для  $i = m+1, \dots, n$ . Тогда вместо системы (6.1) вектор  $w^*$  должен удовлетворять системе

$$(w, y_i) > 0 \quad \forall i = 1, \dots, n; \quad (4.12)$$

и коррекции весового вектора в алгоритме персептрона вместо формулы (4.11) будут осуществляться по следующей упрощенной формуле:

$$w^{k+1} \begin{cases} w^k, & \text{если } (w^{(k)}, y_k) > 0, \\ w^k + y_k, & \text{если } (w^{(k)}, x_k) \leq 0. \end{cases}$$

**Пример.** Обучить персептрон разделять образы по двум классам  $x \in \varpi_1$  и  $x \in \varpi_2$  если известно, что  $\{x_1, x_2\} \subset \varpi_1$  и  $\{x_3, x_4\} \subset \varpi_2$ , где  $x_1 = (1, 0, 1, 0)$ ,  $x_2 = (1, 1, 1, 0)$ ,  $x_3 = (0, 0, 1, 1)$ ,  $x_4 = (1, 1, 0, 0)$  (с помощью таких бинарных векторов можно кодировать, например, бинарные изображения). Процедуру обучения отразим в табл. 4.1.

Таблица 4.1

Векторы	X1	X2	X3	X4	X1	X2	X3	X4	X1	X2	X3	X4	X1	X2	X3	X4
w	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0
	0	0	0	-1	-1	0	0	-1	-1	0	0	-1	-1	0	0	-1
	0	1	1	1	1	2	1	1	1	2	1	1	1	2	1	1
	0	0	0	0	0	0	-1	-1	-1	-1	-2	-2	-2	-2	-3	-3
(w, X)	0	2	1	1	1	0	2	1	1	0	1	1	1	0	0	1
Коррекции	+	-	+	+	-	+	+	+	-	+	+	+	-	+	+	+

Векторы	X1	X2	X3	X4	X1	X2	X3	X4
w	0	1	1	0	0	0	0	0
	-1	0	0	-1	-1	-1	-1	-1
	1	1	2	2	2	2	2	2
	-3	-3	-3	-3	-3	-3	-3	-3
(w, X)	1	0	-1	1	2	1	-1	-1
Коррекции	-	+	-	+	-	-	-	-

В этой таблице в первой строке перечисляются элементы обучающей выборки, в следующих четырех строках приводятся коэффициенты весового вектора  $w$ , в шестой строке – результаты вычислений скалярных произведений, в последней строке “+” означает, что вектор классифицировался неправильно и нужно производить коррекцию коэффициентов, а “–” означает, что вектор классифицировался правильно и коррекцию делать не нужно. После четырех подряд идущих правильных классификаций алгоритм завершает работу, в результате получается весовой вектор  $w = (0, -1, 2, -3)$ . Заметим, что в данном примере мы получим разделяющую гиперплоскость в четырехмерном пространстве, проходящую через начало координат. В общем случае для нахождения разделяющей гиперплоскости (если она существует) в  $n$ -мерном пространстве необходимо вводить смещение: рассматривать  $(n+1)$ -мерные векторы  $x = (1, x_1, \dots, x_n)$  и искать  $(n+1)$ -мерный вектор весов  $w = (w_0, w_1, \dots, w_n)$ .

#### 4.5. Нейронная сеть Хопфилда и алгоритм обучения

В основе нейроинформатики лежит два представления: о строении мозга и о процессах обучения. При рассмотрении строения мозга ключевым элементом является понятие простейшего элемента, кирпичика мозга – «нейрона». Второе представление базируется на возможности, по аналогии с живыми организмами, формировать путем обучения такие связи между нейронами, чтобы множество нейронов (нейронная сеть) могло решать определенную задачу. Простейшей НС является персептрон.

НС допускают как прямое программирование, т.е. формирование связей по явным правилам (существует большой класс задач, для которых связи формируются по явным формулам), так и неявную настройку НС на решение определенных задач. Этот процесс и называют *обучением*.

Обучение обычно строится по принципу «поощрение-наказание»: системе предъявляется набор примеров с заданными ответами. Нейроны преобразуют входные сигналы, выдают ответ – также набор сигналов. Отклонение от правильного ответа

штрафуется путем изменения внутренней настройки сети. Обучение состоит в минимизации отклонения.

*Стандартный формальный нейрон* составлен из входного сумматора, нелинейного преобразователя и точки ветвления на выходе. Наиболее важный элемент нейрона – это *адаптивный сумматор*, который вычисляет скалярное произведение вектора входного сигнала  $\mathbf{x}$  (по аналогии с нейрофизиологией вход нейрона называют *дендритом*) на вектор настраиваемых параметров  $\mathbf{w}$  (рис. 7.1). Далее сигнал поступает на нелинейный преобразователь, называемый *функцией активации*. Он преобразует скалярный входной сигнал  $\mathbf{x}$  в сигнал  $\varphi(\mathbf{x})$  (выход нейрона называют *аксоном*). *Точка ветвления* служит для рассылки одного сигнала по нескольким адресам.

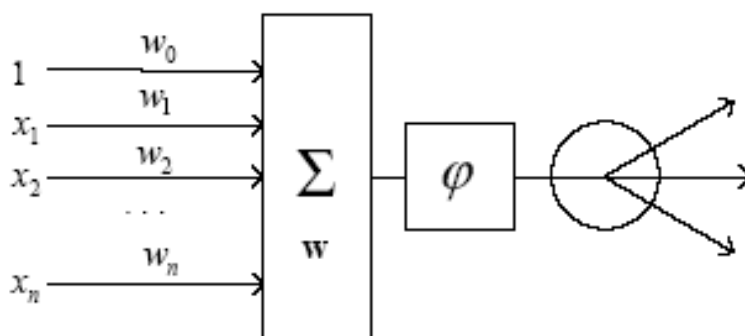


Рис.4.15. Стандартный вид нейрона

*Линейная связь (синапс)* отдельно от сумматоров не встречается. Синапс умножает входной сигнал  $\mathbf{x}$  на «вес синапса»  $w$ .

Среди множества НС можно выделить две базовые архитектуры – *слоистые и полносвязные сети*.

В **слоистых сетях** нейроны расположены в несколько слоев. Нейроны первого слоя принимают входные сигналы, преобразуют их и через точки ветвления передают нейронам второго слоя. Далее срабатывает второй слой и т.д. до  $k$ -го слоя, который выдает выходные сигналы. Если не оговорено противное, то каждый выходной сигнал  $i$ -го слоя подается на входы всех нейронов  $(i+1)$ -го слоя. Число нейронов в каждом слое может быть любым и никак заранее не связано с количеством нейронов в других слоях. Чаще всего встречаются двух- и трехслойные НС.

В **полносвязных сетях** каждый нейрон передает свой выходной сигнал остальным нейронам, включая самого себя. Выходными сигналами сети могут быть все или некоторые выходные сигналы нейронов после нескольких тактов функционирования сети. Все входные сигналы передаются всем нейронам.

НС могут вычислять линейные функции, нелинейные функции *одного переменного*, а также всевозможные суперпозиции – функции от функций, получаемые при каскадном соединении сетей.

НС Хопфилда относится к так называемым сетям *ассоциативной памяти*, т.е. эти сети хранят некоторую информацию об образах и “вспоминают” тот образ, который поступает на ее вход. Входной вектор должен быть биполярным, т.е.  $x_i \in \{-1, 1\}$ . Нейронная сеть Хопфилда для каждого предъявленного биполярного вектора находит наиболее близкий к нему образ-эталон и выдает его на выходе. Информация об образах-эталонах “зашита” в синаптических связях самой НС. Так как векторы биполярны, то квадрат расстояния между вектором  $x$  и эталонным вектором  $e_k$  будет равен  $\|x - e_k\|^2 = \|x\|^2 + \|e_k\|^2 - 2(x, e_k)$ . Тогда расстояние  $\|x - e_k\|$  будет минимальным в том и только том случае, когда скалярное произведение  $(x, e_k)$  максимально. В основе функционирования сети Хопфилда лежит идея минимизации некоторого функционала (так называемого функционала энергии)

$$H(x) = -0.5 \sum_k (x, e_k)^2 + \alpha \sum_i (x_i^2 - 1)^2 + \alpha \sum_i (x_i^2 - 1)^2, \quad \alpha > 0. \quad (4.13)$$

Первое слагаемое в выражении (4.13) будет минимальным в том случае, если вектор  $x$  будет близок к одному (или нескольким) из эталонных векторов, а второе слагаемое будет минимальным в том случае, когда координаты вектора  $x$  будут близки к биполярным. Коэффициент  $\alpha > 0$  регулирует приоритетность этих двух критериев. Рекомендуются в процессе функционирования сети его постепенно увеличивать. Минимизацию функционала  $H(x)$  можно осуществить методом градиентного спуска по формуле

$$x(t+1) = x(t) - h \text{grad} H(x(t)),$$

где  $t$  – номер итерации;  $h$  – градиентный шаг. Для вычисления координат вектора градиента найдем частные производные:



$$\begin{aligned} \frac{\partial H}{\partial x_i} = & -\sum_k (x, e_k) e_{ik} + 4\alpha(x_i^2 - 1)x_i = -\sum_k \left( \sum_j x_j e_{jk} \right) e_{ek} + 4\alpha(x_i^2 - 1)x_i = \\ & -\sum_j \left( \sum_k e_{jk} e_{ik} \right) x_j + 4\alpha(x_i^2 - 1)x_i = -\sum_j w_{ij} x_j + 4\alpha(x_i^2 - 1)x_i, \end{aligned}$$

где  $w_{ij} = \sum_k e_{jk} e_{ik}$  (здесь  $k$  – номер эталона). Второе слагаемое непосредственно вычисляется при  $i$ -м нейроне без участия сети. Вес связи между  $i$ -м и  $j$ -м нейронами равен  $w_{ij} = \sum_k e_{jk} e_{ik}$ . Таким образом, минимизация функционала энергии осуществляется по формуле

$$x_i(t+1) = x_i(t) + h \sum_j w_{ij} x_j(t)$$

или, в матричном виде, по формуле

$$X(t+1) = (I + hW)X(t), \quad (4.14)$$

где  $X(t)$  – вектор-столбец координат вектора  $x$ ;  $I$  – единичная матрица. Здесь  $W = EE^T$ , где  $E = [E_1, \dots, E_m]$  – матрица, составленная из вектор-столбцов  $E_k$  координат эталонных векторов. Матрица  $W$  имеет размер  $n \times n$ , и значения ее элементов не превосходят по модулю числа  $m$ . Поэтому  $\|W\| < nm$  и итерационный процесс (4.14) будет сходиться, если  $0 < h < 1/(nm)$ . Известно, что итерационный процесс будет гарантированно сходиться, если на главную диагональ матрицы  $W$  выставить нулевые элементы.

Сеть Хопфилда является однослойной, полносвязной и имеет структуру, показанную на рис. 7.1. Количество нейронов в слое равно размерности векторов. Исходный вектор  $x$  подается на вход каждого нейрона. В начальный момент весовые коэффициенты синапсов устанавливаются равными

$$q_{ij} = \begin{cases} h \sum_k e_{jk} e_{ik}, & i \neq j; \\ 0, & i = j. \end{cases}$$

**Алгоритм Хопфилда** состоит из следующей последовательности шагов:

1. Подается на вход сигнал  $x$  и полагается, что  $y(0) = x$ ,  $t = 0$ .
2. Рассчитываются новые состояния нейронов по формуле

$$s_i(t+1) = \sum_j q_{ij} y_j(t)$$

и новые значения аксонов (функции активации) по формуле

$$y_i(t+1) = f(s_i(t+1)), \text{ где } f(u) = \begin{cases} 1, & u \geq 0; \\ -1, & u < 0. \end{cases}$$

3. Проверяется условие стабилизации аксонов: если аксоны стабилизировались (т.е.  $y_i(t+1) = y_i(t)$  для всех  $i$ ), то алгоритм завершает работу, в противном случае переходим к пункту 2.

**Замечание.** Иногда сеть Хопфилда не может провести правильное распознавание и выдает на выходе несуществующий образ. Чтобы сеть уверенно распознавала образы, необходимо, чтобы они слабо коррелировали друг с другом, а их количество  $m$  было не больше чем  $0,15n$ , где  $n$  – размерность векторов.

**Пример.** Пусть имеется два 6-мерных эталонных вектора  $e_1 = (1,1,1,-1,1,-1)$  и  $e_2 = (-1,-1,-1,1,1,1)$ . Требуется построить сеть Хопфилда для этих векторов и классифицировать вектор  $x = (1,1,1,-1,1,1)$ .

**Решение.** Заметим, что условие устойчивости распознавания  $m < 0,15n$  в этом примере не выполняется, так как  $m=2$ ,  $n=6$ . Построим матрицу синаптических связей  $Q = I + hW$  для  $h = 0,5$ . Имеем

$$E = [E_1, E_2] = \begin{bmatrix} 1 & 1 & 1 & -1 & 1 & -1 \\ -1 & -1 & -1 & 1 & 1 & 1 \end{bmatrix}^T; \quad Q = I + 0,5EE^T = \begin{bmatrix} 2 & 1 & 1 & -1 & 0 & -1 \\ 1 & 2 & 1 & -1 & 0 & -1 \\ 1 & 1 & 2 & -1 & 0 & -1 \\ -1 & -1 & -1 & 2 & 1 & 1 \\ 0 & 0 & 0 & 0 & 2 & 0 \\ -1 & -1 & -1 & 1 & 0 & 2 \end{bmatrix}.$$

Тогда при предъявлении вектора  $x = (1,1,1,-1,1,1)$  на выходе сети после первой итерации получим вектор-столбец  $S = QX = (4,4,4,-3,2,-2)^T$ , а после применения функции активации  $f$  – вектор-столбец  $Y = (1,1,1,-1,1,-1)^T$ . Таким образом, уже после первой итерации вектор  $x$  будет отнесен к первому классу.

#### 4.6. Многослойные нейронные сети. Алгоритм Хэмминга для обучения нейронных сетей

В этой сети используется свойство расстояния Хэмминга для биполярных векторов. Если имеется два биполярных вектора  $x = (x_i)$  и  $y = (y_i)$ ,  $x_i, y_i \in \{-1, 1\}$ , то расстояние Хэмминга между векторами  $x$  и  $y$  будет равно  $\|x - y\|_1 = \sum_i |x_i - y_i| = 2d(x, y)$ , где  $d(x, y)$  – число различных компонент этих векторов. Кроме того, если  $a(x, y)$  – число совпадающих компонент векторов  $x$  и  $y$ , а  $n$  – размерность этих векторов, то

$$a(x, y) + d(x, y) = n; \quad (x, y) = \sum_i x_i y_i = a(x, y) - d(x, y) = 2a(x, y) - n.$$

Тогда мера близости  $a(x, y)$  между двумя биполярными векторами будет равна  $a(x, y) = \frac{n}{2} + \frac{1}{2} \sum_i x_i y_i$ . Если имеется  $m$  эталонных векторов, то мера близости между входным вектором  $x$  и эталонным вектором  $e_k$  будет равна

$$a(x, e_k) = \frac{n}{2} + \frac{1}{2} \sum_i e_{ik} y_i.$$

Таким образом, если синапсом  $k$ -го нейрона является вектор  $w_k = (w_{ik})_{i=0}^n$ , где  $w_{0k} = n/2$  ( $k=1, \dots, n$ ), то после поступления на вход нейрона вектора  $\tilde{x} = (1, x_1, \dots, x_n)$ , а на выходе этого нейрона получим значение меры близости данного вектора к  $k$ -му вектору-эталону.

НС Хэмминга имеет два слоя. В первом слое  $m$  нейронов (по числу эталонов), на каждый из которых поступают все компоненты входного вектора  $\tilde{x}$ . Каждый  $k$ -й нейрон этого слоя вычисляет значение меры близости  $a_k$  вектора  $x$  к эталонному вектору  $e_k$ . Во втором слое (так называемая сеть Maxnet) вектор мер близости  $a = (a_1, \dots, a_m)$  преобразуется в вектор  $(y_1, \dots, y_m)$ , в котором будет только одна ненулевая компонента. Номер этой компоненты должен быть равен номеру того эталона, к которому наиболее близок (в смысле метрики Хэмминга) входной вектор  $x$ . Сеть Maxnet является полносвязной и однослойной, состоящей из  $m$  нейронов. Выходные сигналы каждого нейрона поступают на входы всех нейронов сети. Сеть функционирует в итерационном режиме до тех пор, пока

значения выходных нейронов не стабилизируются. Схема сети Хэмминга показана на рис. 4.16.

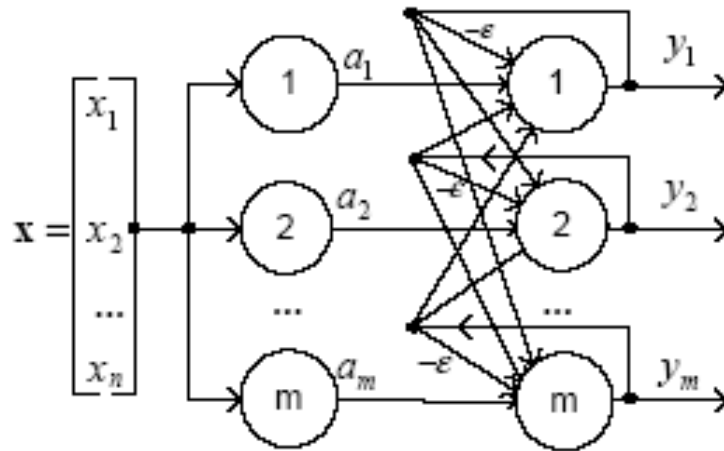


Рис. 4.16.Схемасети Хэмминга

**Алгоритм Хэмминга** состоит из следующей последовательности шагов:

1. На вход первого слоя подается сигнал  $\tilde{x} = (1, x_1, \dots, x_n)$  и вычисляется вектор  $a = (a_1, \dots, a_m)$  мер близостей ( $a_k = w_k \cdot \tilde{x}$ ), при этом полагаем  $y_i(0) = a_i$  ( $i = 1, \dots, m$ ) и  $t=0$ .

2. На вход каждого  $k$ -го нейрона второго слоя поступает вектор  $y_i(t)$ . Рассчитываются новые состояния нейронов по

формуле  $s_i(t+1) = \sum_j q_{ij} y_j(t)$ , где  $q_{ij} = \begin{cases} 1, & i = j, \\ -\epsilon, & i \neq j, \end{cases} \quad 0 < \epsilon \leq 1/n$ .

Вычисляются новые значения аксонов (функции активации)

$y_i(t+1) = f(s_i(t+1))$ , где  $f(u) = \begin{cases} u, & u \geq 0, \\ 0, & u < 0. \end{cases}$

3. Проверяется условие стабилизации аксонов: если аксоны стабилизировались (т.е.  $y_i(t+1) = y_i(t)$  для всех  $i$ ), то алгоритм завершает работу, в противном случае переходим к пункту 2.

**Пример.** Пусть имеется два 6-мерных эталонных вектора  $e_1 = (1,1,1,1,-1,1)$  и  $e_2 = (-1,1,-1,1,1,1)$ . Требуется с помощью сети Хэмминга найти номер эталонного вектора, к которому наиболее близок вектор  $x = (1,1,1,1,-1,-1)$

**Решение.** Вычислим значения мер близости между  $x$  и эталонными векторами:

$$a_1 = \frac{6}{2} + \frac{1}{2}(e_1, x) = 3 + \frac{1}{2} \cdot 4 = 5, \quad a_2 = \frac{6}{2} + \frac{1}{2}(e_2, x) = 3 + \frac{1}{2} \cdot (-2) = 2.$$

Далее с помощью сети Махнет выделим из вектора мер близостей (5,2) максимальную компоненту:

$$\text{шаг 1} \quad \begin{cases} s_1(1) = 5 - \frac{1}{6} \cdot 2 = \frac{14}{3}, \\ s_2(1) = 2 - \frac{1}{6} \cdot 5 = \frac{7}{6} \end{cases} \Rightarrow \begin{cases} y_1(1) = \frac{14}{3}, \\ y_2(1) = \frac{7}{6}; \end{cases}$$

$$\text{шаг 2} \quad \begin{cases} s_1(2) = \frac{14}{3} - \frac{1}{6} \cdot \frac{7}{6} = \frac{161}{36}, \\ s_2(2) = \frac{7}{6} - \frac{1}{6} \cdot \frac{14}{3} = \frac{7}{18} \end{cases} \Rightarrow \begin{cases} y_1(2) = \frac{161}{36}, \\ y_2(2) = \frac{7}{18}; \end{cases}$$

$$\text{шаг 3} \quad \begin{cases} s_1(3) = \frac{161}{36} - \frac{1}{6} \cdot \frac{7}{18} = \frac{119}{27}, \\ s_2(3) = \frac{7}{18} - \frac{1}{6} \cdot \frac{161}{36} = -\frac{77}{216} \end{cases} \Rightarrow \begin{cases} y_1(3) = \frac{119}{27}, \\ y_2(3) = 0; \end{cases}$$

$$\text{шаг 4} \quad \begin{cases} s_1(4) = \frac{119}{27} - \frac{1}{6} \cdot 0 = \frac{119}{27}, \\ s_2(4) = 0 - \frac{1}{6} \cdot \frac{119}{27} = -\frac{119}{162} \end{cases} \Rightarrow \begin{cases} y_1(4) = \frac{119}{27}, \\ y_2(4) = 0; \end{cases}$$

Таким образом, алгоритм Хэмминга относит вектор  $x$  к классу первого эталона.

## Глава 5. СТРУКТУРА ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМ УПРАВЛЕНИЯ

### 5.1. Принципы организации и управления интеллектуальными системами

В [16] выделены два направления применения систем Data Mining: как массового продукта и как инструмента для проведения уникальных исследований.

Следует отметить, что на сегодняшний день наибольшее распространение технология Data Mining получила при решении бизнес-задач. Возможно, причина в том, что именно в этом направлении отдача от использования инструментов Data Mining может составлять, по некоторым источникам, до 1000% и затраты на ее внедрение могут достаточно быстро окупиться.

Сейчас технология Data Mining используется практически во всех сферах деятельности человека, где накоплены ретроспективные данные.

Мы будем рассматривать четыре основные сферы применения технологии Data Mining подобно [22, 27]: наука, бизнес, исследования для правительства и Web-направление.

- Применение Data Mining для решения бизнес-задач. Основные направления: банковское дело, финансы, страхование, CRM, производство, телекоммуникации, электронная коммерция, маркетинг, фондовый рынок и другие.

- Применение Data Mining для решения задач государственного уровня. Основные направления: поиск лиц, уклоняющихся от налогов; средства в борьбе с терроризмом.

- Применение Data Mining для научных исследований. Основные направления: медицина, биология, молекулярная генетика и геновая инженерия, биоинформатика, астрономия, прикладная химия, исследования, касающиеся наркотической зависимости, и другие.

Применение Data Mining для решения Web-задач. Основные направления: поисковые машины (search engines), счетчики и другие.

Сельское хозяйство и агропромышленный комплекс являются ключевой отраслью народного хозяйства, которая обеспечивает людей продуктами питания, необходимыми для любой человеческой жизнедеятельности. В условиях роста населения и ограниченности территорий обостряется проблема нехватки конечной продукции сельскохозяйственной отрасли, а также обеспечения ее высокого качества для всех конечных потребителей. В решении данной задачи в значительной мере могут успешно помочь современные революционные технологии искусственного интеллекта. В настоящий момент данный инструмент повсеместно внедряется во все отрасли народного хозяйства, и, соответственно, не имеется особых преград для его использования в сфере агропромышленности. Эта технология в значительной мере позволяет автоматизировать производственные и управленческие процессы, а также дает возможность обнаружить взаимосвязи в больших объемах неструктурированных данных. Рассмотрим в обобщении и систематизации знаний о перспективных технологиях искусственного интеллекта в сельском хозяйстве, которые позволят обеспечить население качественными продуктами питания, а также дадут возможность предприятиям, реализующим их, получить соответствующие конкурентные преимущества. В работе систематизированы научные знания о современных технологиях сельскохозяйственной отрасли, в которую успешно внедряются технологии искусственного интеллекта: робототехника, фотосъемка и локальная фиксация индикаторов, аудио- и видеоанализ. Разработан и представлен перечень положительных эффектов от их внедрения и распространения. Данная публикация будет полезна специалистам сельскохозяйственной отрасли, а также ученым и исследователям, занимающимся вопросами и проблемами компьютерного программирования и моделирования искусственных интеллектуальных систем.

Технологии искусственного интеллекта (далее – ИИ; автоматизированные процессы и явления, протекающие на условиях, близких к оптимальным, и имеющие возможность при накоплении критической массы статистических данных самосовершенствоваться) [1; 2] в больших объемах и с большей скоростью проникают в различные отрасли народного хозяйства

(виды экономической деятельности) [3; 4; 5]. Это обусловлено экспоненциальным развитием вычислительных мощностей компьютерной техники и непрерывным поиском людьми наилучших решений, которые способны разрешить имеющиеся в настоящий момент проблемы [6]. Искусственный интеллект зарекомендовал себя как более действенный и эффективный способ в вопросах точной и достоверной оценки и диагностики (оценка финансовых рисков, природно-климатических явлений, диагностика заболеваний) [4; 7]. Данная технология показывает лучшие прогнозные результаты по сравнению с классическими методами обработки и анализа данных. В настоящий момент внедрение систем искусственного интеллекта находится на стадии зарождения, но успех подобных программ, несмотря на имеющиеся в технологии недостатки (определение оптимальной архитектуры моделируемых искусственных нейронных сетей, потребность в дорогой компьютерной технике, способной справиться с трудоемкими вычислительными алгоритмами, отсутствие необходимых информационных библиотек и баз данных), показывает лучшие результаты относительно использованных до этого методов [8;9; 10]. Электронно-вычислительные системы, получившие поддержку ИИ, показывают более точные, достоверные и эффективные результаты.

Сельское хозяйство и агропромышленный комплекс являются важной частью любой социально-экономической системы. Это обусловлено тем, что данная совокупность отраслей и направлений хозяйствования обеспечивает людей продуктами питания, которые, в свою очередь, являются жизненно необходимым условием существования человека и обеспечивают его первичные потребности [11]. Помимо этого стоит отметить, что более качественная сельскохозяйственная продукция имеет прямую взаимосвязь с текущим здоровьем людей и его продолжительностью, так как результатом деятельности агропромышленного комплекса являются продукты питания, которые, в свою очередь, выступают строительным материалом для человеческого организма [12; 13]. С ростом населения планеты и всеобщим стремлением к правовой справедливости все сильнее будет обостряться проблема нехватки продуктов питания, и, что не менее важно, будет увеличиваться нужда в росте их качества [14].



В значительной мере данными задачами помогут справиться технологии искусственного интеллекта, которые позволят автоматизировать хозяйственный и управленческий процесс, включая оптимизацию посевных работ, сбор урожая, контроль за состоянием почв, на которых произрастает урожай, правильным составом кормовых смесей, ликвидацию вредителей, автоматизацию кормления животных и многое другое. В рамках данной работы выполнен ряд задач:

определены перспективные технологии искусственного интеллекта, которые могут быть успешно реализованы в сфере сельского хозяйства и агропромышленности, продемонстрированы конкретные направления их использования;

выявлены особенности проанализированных технологий, а также продемонстрированы реальные примеры их использования в сельскохозяйственной среде.

Для наглядности интерпретации технологий, базирующихся на методах и средствах искусственного интеллекта, разработана схема, которая демонстрирует иерархию технических сельскохозяйственных средств, взаимодействующих друг с другом и опирающихся на разработки в ИИ.

Одним из перспективных направлений, в котором используются технологии ИИ и в настоящий момент уже имеются попытки применения в сфере агропромышленности, является анализ и обработка данных на основе компьютерного зрения. Оно заключается в разработке математических компьютерных алгоритмов, базирующихся на формировании специфических математических моделей, именуемых в науке искусственными нейронными сетями (математическая модель и ее компьютерная реализация, взаимодействие элементов которой схоже с деятельностью биологических нейронных сетей живых организмов) [7; 15]. Данная технология успешно справляется с рядом задач, которые включают в себя раннюю диагностику заболеваний животных и растений, что позволяет успешно и своевременно с ними бороться, а также не допускать их появления вообще. Материалом для анализа подобного рода мероприятий могут являться фотографии, сделанные как на микро - так и на макроуровне (популяция животных, посевные площади). И в одном, и в другом случае могут быть обнаружены

прогрессирующие патологии, на основании этого будут предприняты необходимые оперативные меры по их устранению.

Помимо фотосъемки сюда же следует отнести и сбор статистической информации на основе установленных в сельскохозяйственных объектах электронных компьютерных датчиков, которые в динамике будут отслеживать изменения требуемых показателей и информировать управляющего об отклонении от установленных нормативов с целью его вмешательства и перевода ситуации в нужное русло. Подобного рода мероприятиями в настоящий момент занимается компания IntelinAir, которая представлена на рынке как организация по реализации точного сельского хозяйства. Под этим как раз и понимается ведение деятельности и ее оптимизация (в нашем случае сельскохозяйственное) на основе регулярно фиксируемых показателей. В настоящий момент IntelinAir ориентирована на поддержку фермеров, занимающихся выращиванием кукурузы и сои в штатах Иллинойс, Айова, Огайо и Индиана. Еще одной организацией из сферы анализа и обработки данных на основе компьютерного зрения выступает канадский Исследовательский центр по фенотипированию и обработке изображений растений в Университете Саскачевана. Формируются статистические библиотеки параметров данных о различных сельскохозяйственных культурах, которые с развитием цифровых технологий могут быть успешно использованы и применены в различного рода агропромышленных проектах, а также в научно-исследовательских проектах, направленных на решение текущих сельскохозяйственных проблем и совершенствование технологий в этой сфере. С развитием возможностей электронно-вычислительных машин в повседневной деятельности человека все чаще задействована робототехника, которая постепенно начинает появляться и распространяться в отрасли сельского хозяйства [16]. Данная технология позволяет автоматизировать процессы хозяйственной деятельности, высвобождая драгоценное время на проведение исследований и разработок, необходимых для решения нетривиальных, еще неразрешенных задач. Стоит отметить, что данная технология тесно связана с предыдущим направлением и может использоваться для оптимизации внутренних процессов роботизированных систем, а также выступать платформой для

сбора необходимых данных с объектов сельского хозяйства и агропромышленного комплекса. Примером роботизированной техники, имитирующей функции человека, в сельском хозяйстве является робот Prospero [17]. Он обладает следующими возможностями: индивидуальная посадка рассады вплоть до выкапывания ямки под саженец и дальнейшего его размещения в ней, последующее наблюдение и сбор урожая. В компьютерную систему заложены алгоритмы распознавания ландшафта, на основании которого выбираются идеальные условия для выращивания необходимых культур. Ученые и исследователи Орхусского университета в Дании разработали робота «Hortibot» [16; 17], способного распознавать сельскохозяйственные культуры и в соответствии с этим ликвидировать сорняки путем механического воздействия или с использованием химических реагентов. Данная разработка считается значительным прорывом в научном мире, так как ранее автономно отличать полезные сельскохозяйственные культуры от сорных трав качественно не удавалось. Помимо таких уникальных единичных проектов на сельскохозяйственном рынке в сфере робототехники и внедрения систем, поддерживающих технологии искусственного интеллекта, в настоящий момент существует следующее:

внедрение в сельскохозяйственную практику автоматизированных дистанционных небольших летательных аппаратов (дронов) [18], которые локально могут доставлять необходимые для растений и животных ресурсы, переносить опасные химикаты без угрозы для жизни человека, а также проводить фотосъемку, полученные данные которой могут успешно использоваться в системах компьютерного зрения;

автоматизированные сельскохозяйственные комплексы, представляющие собой более высокотехнологичную и совершенную технику на манер тракторов, комбайнов, экскаваторов, способные выполнять несколько различных функций: посадка и сбор урожая, контроль за развитием растительных и животных культур, ликвидация вредителей и последствий природно-климатических явлений [16–19].

В связи с вышеизложенным, на наш взгляд, в отношении робототехники следует отметить, что в настоящий момент данные проекты носят характер прототипов, но задел и перспективы в

будущем прослеживаются уже сейчас. Внедрение подобной техники в практику агропромышленного комплекса позволит оптимизировать имеющиеся в данной сфере процессы и процедуры, что отразится на росте объемов продукции, ее качестве, снижении различного рода затрат. Примером экономической и социальной оценки в области использования в сельском хозяйстве робототехники может служить анализ микроробототехники и роевого искусственного интеллекта, представленный в публичных отчетах ИСИЭЗ [20]. По их расчетам, объем рынка в мировом разрезе для средств и техники, способной заменить медоносных пчел, составит более 4 млрд долл. Обслуживание подобных систем оценивается на уровне 200 млрд долл. в год. Данный рынок, безусловно, будет развиваться, так как решает проблемы повышения урожайности, автоматизации сельскохозяйственных работ, способствует развитию мониторинга экосистем окружающей среды. Мощностей современных компьютеров хватает не только на обработку фотографий и детерминированных количественных статистических данных, но и на динамический анализ на основе видеосъемки и видеонаблюдения [20]. В связи с этим технологии искусственного интеллекта могут быть использованы для создания неинвазивных (без вмешательства в естественную среду обитания животных) методов отслеживания характеристики параметров животных. Компания Neuromation планирует реализовать проект по отслеживанию жизнедеятельности свиней, который позволит контролировать динамику их веса в популяции без физического вмешательства со стороны человека. Согласно имеющейся практике, процедуры, проводимые в настоящее время над сельскохозяйственными животными и используемые для оценки их состояния здоровья, заставляют животных испытывать стресс, что, в свою очередь, негативно отражается на качестве конечной продукции агропромышленной отрасли (потеря полезного веса животных, используемого для создания продуктов питания). Планируемый разработчиками алгоритм позволит фиксировать вес поросят, а также сможет давать рекомендации для составления их рациона питания, чтобы на выходе не только получать большие объемы реализации, но и одновременно улучшать их качество.

Три выделенные в данной работе технологии, которые непосредственно связаны с применением технологий ИИ, в

настоящий момент все активнее будут выходить на передовые рубежи сельского хозяйства. Это обусловлено тем, что при автоматизации сельскохозяйственных процессов оптимальным образом решаются имеющиеся в агропромышленном комплексе задачи, что позволяет ускорить процессы выращивания необходимых объемов растительных культур и продуктов питания без риска снижения качества конечной продукции. Более подробно выделены следующие положительные моменты:

- снижение рисков недостижения планируемых показателей либо их своевременное обозначение и обоснование новых нормативных величин;

- оперативное реагирование на изменение природно-климатических условий, позволяющее своевременно принимать необходимые меры для защиты объектов сельского хозяйства и агропромышленного комплекса;

- рост урожайности сельскохозяйственных культур и увеличение объемов продуктивности животных, позволяющие обеспечить продуктами питания растущее население планеты;

- снижение производственных затрат на основе внедрения принципов точного производства и сбора оперативных данных для его эффективного и автоматизированного управления;

- решение логистических задач, которые позволят сократить количество посредников от производителя до конечного потребителя, что должно отразиться на снижении стоимости потребительского продукта;

- с течением времени сокращение нехватки квалифицированной рабочей силы и создание высокотехнологичных рабочих мест, необходимых для разработки компьютерных систем поддержки технологии искусственного интеллекта;

- оперативное оповещение необходимой информацией как самих товаропроизводителей сельскохозяйственной продукции, так и их клиентов.

Технологии ИИ сопряжены с глобальной цифровизацией экономики, которая базируется на автоматическом анализе огромных массивов числовых данных.

В заключение отметим, что технологии искусственного интеллекта успешно показывают себя в случае с более качественной обработкой количественных данных. При этом

анализируемая информация может быть разрозненной и слабоструктурированной. В здравоохранении – это данные диагностики и медицинских анализов; в экономике – динамика индексов, переливы денежных средств; в военной промышленности – индикативные показатели, фиксируемые средствами ПВО; в юриспруденции – унифицированный документооборот и др. В связи с этим в настоящий момент технология ИИ внедряется повсеместно. Отрасли сельского хозяйства не являются исключением, так как в данном направлении могут эффективно решаться задачи, имеющие в своей основе проблему комбинаторного взрыва (большой объем данных, который не может быть оперативно проанализирован, в результате чего решение проблемы таким образом является не актуальным). К ним могут быть отнесены задачи, привязанные к данным, полученным в результате фотосъемки, моделирование автономных человекоподобных систем, способных эффективно выполнять сельскохозяйственные процедуры, оперативный анализ данных, полученных в результате видеосъемки. Использование технологий ИИ в данном направлении, как это уже было отмечено ранее, не только позволит увеличить объем и качество сельскохозяйственной продукции, но и даст возможность компаниям агропромышленного комплекса стать лидерами в своей сфере деятельности. Результаты проделанной работы систематизируют знания об имеющихся в настоящий момент технологиях искусственного интеллекта, которые в различной мере зарекомендовали себя в качестве успешных методов решения проблем в сфере сельского хозяйства и агропромышленности, а также будут активно распространяться в этом направлении в течение следующих 5–7 лет. Их разработка в настоящее время позволит хозяйствующему субъекту извлечь доминирующие конкурентные преимущества и соответствующий значительный экономический эффект, обусловленный новшеством технологии. Дальнейшими этапами исследования станут подробное изучение различного рода технологий и методов искусственного интеллекта с целью поиска их наилучшего использования в различных сферах человеческой жизнедеятельности (в том числе в сельском хозяйстве и агропромышленности), определение их эффективного сочетания между собой и обоснование оптимальной внутренней структуры (математизация и программный код).

Применение Data Mining для решения бизнес-задач.

Банковское дело

Технология Data Mining используется в банковской сфере для решения ряда типичных задач.

Задача "Выдавать ли кредит клиенту?"

Классический пример применения Data Mining в банковском деле - решение задачи определения возможной некредитоспособности клиента банка. Эту задачу также называют анализом кредитоспособности клиента или "Выдавать ли кредит клиенту?".

Без применения технологии Data Mining задача решается сотрудниками банковского учреждения на основе их опыта, интуиции и субъективных представлений о том, какой клиент является благонадежным. По похожей схеме работают системы поддержки принятия решений и на основе методов Data Mining. Такие системы на основе исторической (ретроспективной) информации и при помощи методов классификации выявляют клиентов, которые в прошлом не вернули кредит.

Задача "Выдавать ли кредит клиенту?" при помощи методов Data Mining решается следующим образом. Совокупность клиентов банка разбивается на два класса (вернувшие и не вернувшие кредит); на основе группы клиентов, не вернувших кредит, определяются основные "черты" потенциального неплательщика; при поступлении информации о новом клиенте определяется его класс ("вернет кредит", "не вернет кредит").

Задача привлечения новых клиентов банка.

С помощью инструментов Data Mining возможно провести классификацию на "более выгодных" и "менее выгодных" клиентов. После определения наиболее выгодного сегмента клиентов банку есть смысл проводить более активную маркетинговую политику по привлечению клиентов именно среди найденной группы.

Другие задачи сегментации клиентов.

Разбивая клиентов при помощи инструментов Data Mining на различные группы, банк имеет возможность сделать свою маркетинговую политику более целенаправленной, а потому - эффективной, предлагая различным группам клиентов именно те виды услуг, в которых они нуждаются.

Задача управления ликвидностью банка. Прогнозирование остатка на счетах клиентов.

Проводя прогнозирования временного ряда с информацией об остатках на счетах клиентов за предыдущие периоды, применяя методы Data Mining, можно получить прогноз остатка на счетах в определенный момент в будущем. Полученные результаты могут быть использованы для оценки и управления ликвидностью банка.

Задача выявления случаев мошенничества с кредитными карточками.

Для выявления подозрительных операций с кредитными карточками применяются так называемые "подозрительные стереотипы поведения", определяемые в результате анализа банковских транзакций, которые впоследствии оказались мошенническими. Для определения подозрительных случаев используется совокупность последовательных операций на определенном временном интервале. Если система Data Mining считает очередную операцию подозрительной, банковский работник может, ориентируясь на эту информацию, заблокировать операции с определенной карточкой.

#### *Страхование*

Страховой бизнес связан с определенным риском. Здесь задачи, решаемые при помощи Data Mining, сходны с задачами в банковском деле.

Информация, полученная в результате сегментации клиентов на группы, используется для определения групп клиентов. В результате страховая компания может с наибольшей выгодой и наименьшим риском предлагать определенные группы услуг конкретным группам клиентов.

Задача выявления мошенничества решается путем нахождения некоего общего стереотипа поведения клиентов-мошенников.

#### *Телекоммуникации*

В сфере телекоммуникаций достижения Data Mining могут использоваться для решения задачи, типичной для любой компании, которая работает с целью привлечения постоянных клиентов, - определения лояльности этих клиентов. Необходимость решения таких задач обусловлена жесткой конкуренцией на рынке телекоммуникаций и постоянной миграцией клиентов от одной



компании в другую. Как известно, удержание клиента намного дешевле его возврата. Поэтому возникает необходимость выявления определенных групп клиентов и разработка наборов услуг, наиболее привлекательных именно для них. В этой сфере, так же как и во многих других, важной задачей является выявление фактов мошенничества.

Помимо таких задач, являющихся типичными для многих областей деятельности, существует группа задач, определяемых спецификой сферы телекоммуникаций.

#### *Электронная коммерция*

В сфере электронной коммерции Data Mining применяется для формирования рекомендательных систем и решения задач классификации посетителей Web-сайтов. Такая классификация позволяет компаниям выявлять определенные группы клиентов и проводить маркетинговую политику в соответствии с обнаруженными интересами и потребностями клиентов. Технология Data Mining для электронной коммерции тесно связана с технологией Web Mining [28].

#### *Промышленное производство*

Особенности промышленного производства и технологических процессов создают хорошие предпосылки для возможности использования технологии Data Mining в ходе решения различных производственных задач. Технический процесс по своей природе должен быть контролируемым, а все его отклонения находятся в заранее известных пределах;

т.е. здесь мы можем говорить об определенной стабильности, которая обычно не присуща большинству задач, встающих перед технологией Data Mining.

Основные задачи Data Mining в промышленном производстве [29]:

- комплексный системный анализ производственных ситуаций;
- краткосрочный и долгосрочный прогноз развития производственных ситуаций;
- выработка вариантов оптимизационных решений;
- прогнозирование качества изделия в зависимости от некоторых параметров технологического процесса;
- обнаружение скрытых тенденций и закономерностей

развития производственных процессов;

- прогнозирование закономерностей развития производственных процессов;
- обнаружение скрытых факторов влияния;
- обнаружение и идентификация ранее неизвестных взаимосвязей между производственными параметрами и факторами влияния;
- анализ среды взаимодействия производственных процессов и прогнозирование;
- изменения ее характеристик;
- выработку оптимизационных рекомендаций по управлению производственными процессами;
- визуализацию результатов анализа, подготовку предварительных отчетов и проектов допустимых решений с оценками достоверности и эффективности возможных реализаций.

#### *Маркетинг*

В сфере маркетинга Data Mining находит очень широкое применение.

Основные вопросы маркетинга "Что продается?", "Как продается?", "Кто является потребителем?"

В лекции, посвященной задачам классификации и кластеризации, подробно описано использование кластерного анализа для решения задач маркетинга, как, например, сегментация потребителей.

Другой распространенный набор методов для решения задач маркетинга - методы и алгоритмы поиска ассоциативных правил.

Также успешно здесь используется поиск временных закономерностей.

#### *Розничная торговля*

В сфере розничной торговли, как и в маркетинге, применяются:

- алгоритмы поиска ассоциативных правил (для определения часто встречающихся наборов товаров, которые покупатели покупают одновременно). Выявление таких правил помогает размещать товары на прилавках торговых залов, выработать стратегии закупки товаров и их размещения на складах и т.д.

- использование временных последовательностей, например, для определения необходимых объемов запасов товаров на складе.
- методы классификации и кластеризации для определения групп или категорий клиентов, знание которых способствует успешному продвижению товаров.

#### *Фондовый рынок*

Вот список задач фондового рынка, которые можно решать при помощи технологии Data Mining [30]:

- прогнозирование будущих значений финансовых инструментов и индикаторов по их прошлым значениям;
- прогноз тренда (будущего направления движения - рост, падение, флэт) финансового инструмента и его силы (сильный, умеренно сильный и т.д.);
- выделение кластерной структуры рынка, отрасли, сектора по некоторому набору характеристик;
- динамическое управление портфелем;
- прогноз волатильности;
- оценка рисков;
- предсказание наступления кризиса и прогноз его развития;
- выбор активов и др.

Кроме описанных выше сфер деятельности, технология Data Mining может применяться в самых разнообразных областях бизнеса, где есть необходимость в анализе данных и накоплен некоторый объем ретроспективной информации.

#### *Применение Data Mining в CRM*

Одно из наиболее перспективных направлений применения Data Mining - использование данной технологии в аналитическом CRM.

CRM (Customer Relationship Management) - управление отношениями с клиентами.

При совместном использовании этих технологий добыча знаний совмещается с "добычей денег" из данных о клиентах.

Важным аспектом в работе отделов маркетинга и отдела продаж является составление целостного представления о клиентах, информация об их особенностях, характеристиках, структуре клиентской базы. В CRM используется так называемое

профилирование клиентов, дающее полное представление всей необходимой информации о клиентах. Профилирование клиентов включает следующие компоненты: сегментация клиентов, прибыльность клиентов, удержание клиентов, анализ реакции клиентов. Каждый из этих компонентов может исследоваться при помощи Data Mining, а анализ их в совокупности, как компонентов профилирования, в результате может дать те знания, которые из каждой отдельной характеристики получить невозможно.

В результате использования Data Mining решается задача сегментации клиентов на основе их прибыльности. Анализ выделяет те сегменты покупателей, которые приносят наибольшую прибыль. Сегментация также может осуществляться на основе лояльности клиентов. В результате сегментации вся клиентская база будет поделена на определенные сегменты, с общими характеристиками. В соответствии с этими характеристиками компания может индивидуально подбирать маркетинговую политику для каждой группы клиентов.

Также можно использовать технологию Data Mining для прогнозирования реакции определенного сегмента клиентов на определенный вид рекламы или рекламных акций - на основе ретроспективных данных, накопленных в предыдущие периоды.

Таким образом, определяя закономерности поведения клиентов при помощи технологии Data Mining, можно существенно повысить эффективность работы отделов маркетинга, продаж и сбыта. При объединении технологий CRM и Data Mining и грамотном их внедрении в бизнес компания получает значительные преимущества перед конкурентами.

Исследования для правительства

В планах правительства США стоит создание системы, которая позволит отслеживать всех иностранцев, приезжающих в страну. Задача этого комплекса: начиная с пограничного терминала, на основе технологии биометрической идентификации личности и различных других баз данных контролировать, насколько реальные планы иностранцев соответствуют заявленным ранее (включая перемещения по стране, сроки отъезда и др.). Предварительная стоимость системы составляет более 10 млрд. долларов, разработчик комплекса - компания Accenture.

По данным аналитического отчета Главного контрольного

управления американского Конгресса, правительственные ведомства США участвуют приблизительно в двухстах проектах на основе анализа данных (Data Mining), собирающих разнообразную информацию о населении. Более ста из этих проектов направлены на сбор персональной информации (имена, фамилии, адреса e-mail, номера соцстрахования и удостоверений водительских прав), и на основе этой информации осуществляют предсказания возможного поведения людей. Поскольку в упомянутом отчете не приведена информация о секретных отчетах, надо полагать, что общее число таких систем значительно больше.

Несмотря на пользу, которую приносят системы отслеживания, эксперты упомянутого управления, так же как и независимые эксперты, предупреждают о значительном риске, с которым связаны подобные проекты. Причина опасений - проблемы, которые могут возникнуть при управлении и надзоре за такими базами.

#### *Data Mining для научных исследований*

##### *Биоинформатика*

Одна из научных областей применения технологии Data Mining - биоинформатика, направление, целью которого является разработка алгоритмов для анализа и систематизации генетической информации. Полученные алгоритмы используются для определения структур макромолекул, а также их функций, с целью объяснения различных биологических явлений.

##### *Медицина*

Несмотря на консервативность медицины во многих ее аспектах, технология Data Mining в последние годы активно применяется для различных исследований и в этой сфере человеческой деятельности. Традиционно для постановки медицинских диагнозов используются экспертные системы, которые построены на основе символьных правил, сочетающих, например, симптомы пациента и его заболевание. С использованием Data Mining при помощи шаблонов можно разработать базу знаний для экспертной системы.

##### *Фармацевтика*

В области фармацевтики методы Data Mining также имеют достаточно широкое применение. Это задачи исследования

эффективности клинического применения определенных препаратов, определение групп препаратов, которые будут эффективны для конкретных групп пациентов. Актуальными здесь также являются задачи продвижения лекарственных препаратов на рынок.

#### *Молекулярная генетика и геновая инженерия*

В молекулярной генетике и геновой инженерии выделяют отдельное направление Data Mining, которое имеет название анализ данных в микро-массивах (Microarray Data Analysis, MDA). Подробно с применением Microarray Data Analysis можно ознакомиться в [22].

Некоторые применения этого направления:

- ранняя и более точная диагностика;
- новые молекулярные цели для терапии;
- улучшенные и индивидуально подобранные виды лечения;
- фундаментальные биологические открытия.

Примеры использования Data Mining - молекулярный диагноз некоторых серьезных заболеваний; открытие того, что генетический код действительно может предсказывать вероятность заболевания; открытие некоторых новых лекарств и препаратов.

Основные понятия, которыми оперирует Data Mining в областях "Молекулярная генетика и геновая инженерия" - маркеры, т.е. генетические коды, которые контролируют различные признаки живого организма.

На финансирование проектов с использованием Data Mining в рассматриваемых сферах выделяют значительные финансовые средства.

#### *Химия*

Технология Data Mining активно используется в исследованиях органической и неорганической химии. Одно из возможных применений Data Mining в этой сфере - выявление каких-либо специфических особенностей строения соединений, которые могут включать тысячи элементов.

Далее мы рассмотрим технологии, в основу которых также положено понятие Mining или "добыча".

### *Web Mining*

Web Mining можно перевести как "добыча данных в Web". Web Intelligence или Web Интеллект готов "открыть новую главу" в стремительном развитии электронного бизнеса. Способность определять интересы и предпочтения каждого посетителя, наблюдая за его поведением, является серьезным и критичным преимуществом конкурентной борьбы на рынке электронной коммерции.

Системы Web Mining могут ответить на многие вопросы, например, кто из посетителей является потенциальным клиентом Web-магазина, какая группа клиентов Web-магазина приносит наибольший доход, каковы интересы определенного посетителя или группы посетителей.

Технология Web Mining охватывает методы, которые способны на основе данных сайта обнаружить новые, ранее неизвестные знания и которые в дальнейшем можно будет использовать на практике. Другими словами, технология Web Mining применяет технологию Data Mining для анализа неструктурированной, неоднородной, распределенной и значительной по объему информации, содержащейся на Web-узлах.

Согласно таксономии Web Mining [31], здесь можно выделить два основных направления: Web Content Mining и Web Usage Mining.

Web Content Mining подразумевает автоматический поиск и извлечение качественной информации разнообразных источников Интернета, перегруженных "информационным шумом". Здесь также идет речь о различных средствах кластеризации и аннотирования документов.

В этом направлении, в свою очередь, выделяют два подхода: подход, основанный на агентах, и подход, основанный на базах данных.

Подход, основанный на агентах (Agent Based Approach), включает такие системы:

- интеллектуальные поисковые агенты (Intelligent Search Agents);
- фильтрация информации / классификация;
- персонифицированные агенты сети.

Примеры систем интеллектуальных агентов поиска:

- Harvest (Brown и др., 1994),
- FAQ-Finder (Hammond и др., 1995),
- Information Manifold (Kirk и др., 1995),
- OCCAM (Kwok and Weld, 1996), and ParaSite (Spertus, 1997),
- ILA (Information Learning Agent) (Perkowitz and Etzioni, 1995),
- ShopBot (Doorenbos и др., 1996).

Подход, основанный на базах данных (Database Approach), включает системы:

- многоуровневые базы данных;
- системы web-запросов (Web Query Systems);

Примеры систем web-запросов:

- W3QL (Konopnicki и Shmueli, 1995),
- WebLog (Lakshmanan и др., 1996),
- Lorel (Quass и др., 1995),
- UnQL (Buneman и др., 1995 and 1996),
- TSIMMIS (Chawathe и др., 1994).

Второе направление Web Usage Mining подразумевает обнаружение закономерностей в действиях пользователя Web-узла или их группы.

Анализируется следующая информация:

- какие страницы просматривал пользователь;
- какова последовательность просмотра страниц.

Анализируется также, какие группы пользователей можно выделить среди общего их числа на основе истории просмотра Web-узла.

Web Usage Mining включает следующие составляющие:

- предварительная обработка;
- операционная идентификация;
- инструменты обнаружения шаблонов;
- инструменты анализа шаблонов.

При использовании Web Mining перед разработчиками возникает два типа задач. Первая касается сбора данных, вторая - использования методов персонификации. В результате сбора некоторого объема персонифицированных ретроспективных данных о конкретном клиенте, система накапливает определенные знания о нем и может рекомендовать ему, например, определенные



наборы товаров или услуг. На основе информации о всех посетителях сайта Web-система может выявить определенные группы посетителей и также рекомендовать им товары или же предлагать товары в рассылках.

Задачи Web Mining согласно [31] можно подразделить на такие категории:

- Предварительная обработка данных для Web Mining.
- Обнаружение шаблонов и открытие знаний с использованием ассоциативных правил, временных последовательностей, классификации и кластеризации;
- Анализ полученного знания.

### *Text Mining*

Text Mining охватывает новые методы для выполнения семантического анализа текстов, информационного поиска и управления. Синонимом понятия Text Mining является KDT (Knowledge Discovering in Text - поиск или обнаружение знаний в тексте).

В отличие от технологии Data Mining, которая предусматривает анализ упорядоченной в некие структуры информации, технология Text Mining анализирует большие и сверхбольшие массивы неструктурированной информации.

Программы, реализующие эту задачу, должны некоторым образом оперировать естественным человеческим языком и при этом понимать семантику анализируемого текста. Один из методов, на котором основаны некоторые Text Mining системы, - поиск так называемой подстроки в строке.

### *Call Mining*

По словам Энн Беднарц [32], "добыча звонков" может стать популярным инструментом корпоративных информационных систем.

Технология Call Mining объединяет в себя распознавание речи, ее анализ и Data Mining. Ее цель - упрощение поиска в аудио-архивах, содержащих записи переговоров между операторами и клиентами. При помощи этой технологии операторы могут обнаруживать недостатки в системе обслуживания клиентов, находить возможности увеличения продаж, а также выявлять тенденции в обращениях клиентов.

Среди разработчиков новой технологии Call Mining ("добыча" и анализ звонков) - компании CallMiner, Nexidia, ScanSoft, Witness Systems. В технологии Call Mining разработано два подхода - на основе преобразования речи в текст и на базе фонетического анализа.

Примером реализации первого подхода, основанного на преобразовании речи, является система CallMiner. В процессе Call Mining сначала используется система преобразования речи, затем следует ее анализ, в ходе которого в зависимости от содержания разговоров формируется статистика телефонных вызовов. Полученная информация хранится в базе данных, в которой возможен поиск, извлечение и обработка.

Пример реализации второго подхода - фонетического анализа - продукция компании Nexidia. При этом подходе речь разбивается на фонемы, являющиеся звуками или их сочетаниями. Такие элементы образуют распознаваемые фрагменты. При поиске определенных слов и их сочетаний система идентифицирует их с фонемами.

Аналитики отмечают, что за последние годы интерес к системам на основе Call Mining значительно возрос. Это объясняется тем фактом, что менеджеры высшего звена компаний, работающих в различных сферах, в т.ч. в области финансов, мобильной связи, авиабизнеса, не хотят тратить много времени на прослушивание звонков с целью обобщения информации или же выявления каких-либо фактов нарушений.

По словам Дэниэла Хонг, аналитика компании Datamonitor: "Использование этих технологий повышает оперативность и снижает стоимость обработки информации".

Типичная инсталляция продукции от разработчика Nexidia обходится в сумму от 100 до 300 тыс. долл. Стоимость внедрения системы CallMiner по преобразованию речи и набора аналитических приложений составляет около 450 тыс. долл.

По мнению Шоллера, приложения Audio Mining и Video Mining найдут со временем гораздо более широкое применение, например, при индексации учебных видеофильмов и презентаций в медиабibliothеках компаний.

## 5.2. Проблемы решения задач оптимального управления

В автоматическом управлении широкого класса объектов, особенно сложных промышленных процессов, в последние годы успешно применяются нечеткие системы управления (НСУ). НСУ, являясь качественно новым классом систем управления, хорошо зарекомендовали себя для управления сложными нелинейными неопределенными динамическими объектами, для которых классические детерминированные и стохастические контроллеры неприемлемы.

### Общая методология построения нечетких систем управления

**Определение 5.1. Нечеткий автомат [1-3].** Нечеткий конечный автомат характеризуется упорядоченной шестеркой  $A = \langle U, X, Y, s_0, \delta, \sigma \rangle$ , где  $U = \{a_1, \dots, a_m\}$ ,  $X = \{x_1, \dots, x_n\}$ ,  $Y = \{y_1, \dots, y_p\}$  - конечные множества входов, состояний и выходов, соответственно,  $\delta: X \times U \times X \rightarrow L$ ,  $\sigma: X \times Y \rightarrow L$  - функции переходов и выходов соответственно,  $s_0$  - нечеткое начальное состояние ( $s_0: X \rightarrow L$ ).  $\delta$  - порождает множество нечетких матриц переходов;  $\sigma$  - порождает нечеткую матрицу выхода. Детерминированный конечный автомат является частным случаем (когда  $\delta_{x_i x_j}$  равна 0 или 1) нечеткого конечного автомата.

**Определение 5.2. Нечеткий алгоритм.** Упорядоченное множество нечетких инструкций (высказываний), содержащих понятия, формализуемые нечеткими множествами называется нечетким алгоритмом [3,4].

В некоторых практических случаях применяются нечеткие алгоритмы, описываемые конечными нечеткими автоматами.

**Определение 5.3. Нечеткий контроллер.** Нечеткий контроллер есть основанный на знании контроллер, в котором нечеткая логика используется для представления знаний и логического вывода.

Базовая структура нечеткой системы управления представлена на рис.5.1. Текущий выход  $Y(t)$  управляемого процесса 1 в виде четкого сигнала поступает обратной связью на вход системы, где сравнивается в элементе 3 с четким заданием  $g(t)$ . Ошибка  $e(t)$  если

требуется, ее производные ( $e', e'' \dots$ ), интеграл от ошибки ( $\sum e(t)$ ) поступают в виде четких сигналов на вход нечеткого контроллера 2. Последний включает фаззификатор 4, предназначенный для трансформации четких сигналов  $e, e', \sum e$  и др., в нечеткие множества  $e^*, e^{**}, (\sum e)^*$  и др.

Механизм вывода 5, получая эти нечеткие сигналы с использованием базы данных 6, где хранятся функции принадлежности нечетких множеств, описывающих эти сигналы и базы знаний, где хранятся нечеткие правила регулирования осуществляет логический вывод для получения выходного нечеткого сигнала  $u^*(t)$  контроллера. Так как на вход управляемого процесса (на исполнительный орган) должен поступить четкий управляющий сигнал  $u$ , то элемент 7 (дефаззификатор) осуществляет трансформацию нечеткого управления  $u^*$  в четкий сигнал управления  $u(t)$ .

Ниже приводятся функции и принципы работы отдельных блоков и нечеткой системы в целом, представленной на рис. 5.1.

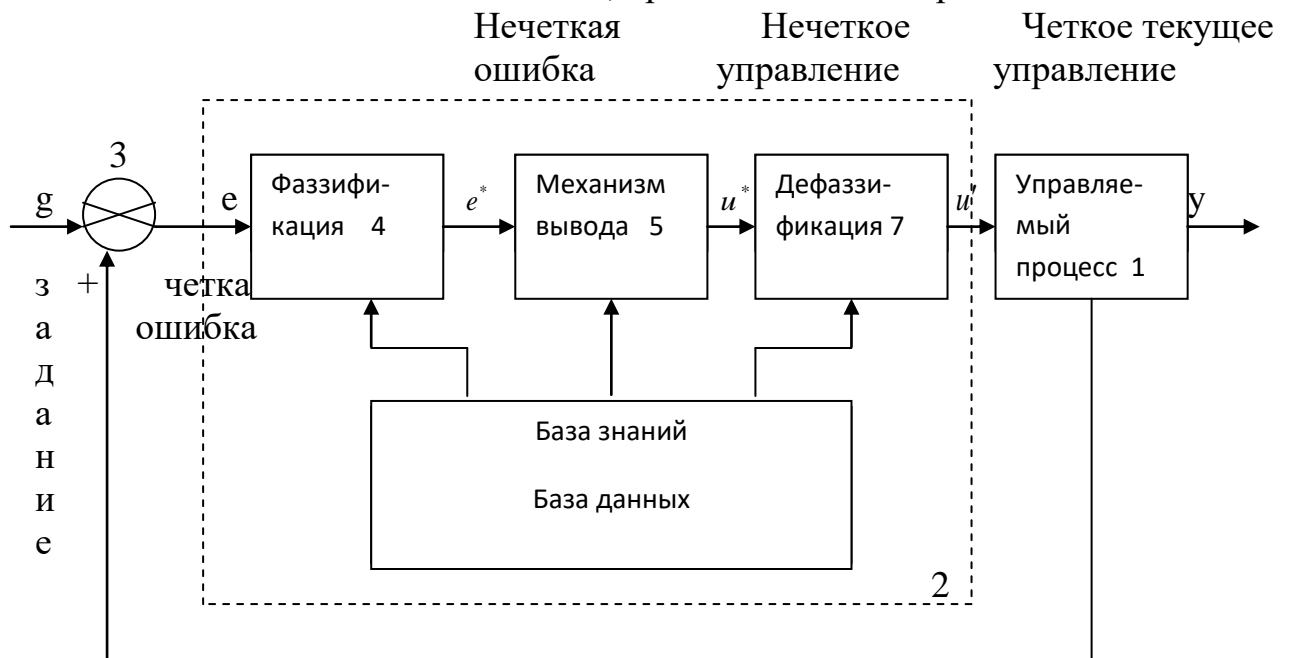


Рис.5.1. Базовая структура нечеткой системы управления

## 1. Фаззфикатор

**Определение 5.4. Фаззификатор.** Фаззификация есть отображение измеренных четких входов в нечеткие множества определенного универсума.

Так как входы и выходы управляемого объекта, следовательно, и ошибка в системе управления есть четкие сигналы, а нечеткий контроллер проводит обработку данных на базе теории нечетких множеств, то необходимо преобразовать четкие сигналы в нечеткие. Для этого используется следующий оператор фаззификации

$$F = \text{fuzzifier}(e_0),$$

где  $e_0$  - есть четкий сигнал, поступающий с системы, F-нечеткое множество, fuzzifier-оператор фаззификации.

Имеются следующие основные пути фаззификации.

- фаззификатор преобразовывает четкое значение в нечеткий синглтон. Входной сигнал  $e_0$  преобразовывается в нечеткое множество с функцией принадлежности  $\mu_A(e)$ , равной нулю, за исключением в точке  $e_0$ , где она равна 1;

- если измеренные данные сопровождаются случайной помехой, то фаззификатор преобразовывает вероятностные данные в нечеткие;

- если результаты измерения характеризуются обеими вероятностными и возможностными неопределенностями, то оператор fuzzifier использует концепцию «гибридного числа».

## 2. База данных.

Проектирование базы данных в нечетких системах управления включает дискретизацию, нормализацию универсума, нечеткое разделение пространства входов и выходов, определение функций принадлежности нечетких множеств.

Для дискретизации универсума должно быть осуществлено шкалирование, которое преобразовывает измеренные значения сигналов в значения дискретизированного универсума. Выбор уровня квантизации связан с априори знанием. Допустим, что нечеткий контроллер имеет правила регулирования следующего типа

$$R_i: \text{ЕСЛИ } e_i \text{ ЕСТЬ } A_i \text{ И } e'_i \text{ ЕСТЬ } B_i \text{ ТО } u \text{ ЕСТЬ } C_i$$

Простой пример нечеткого контроллера может быть представлен как

$$K_3[U(K)] = F[K_1e(K), K_2e'(K)]$$

где  $F$  – нечеткое отношение определяемое базой знаний:  $K_i, i = \overline{1,3}$  коэффициенты масштабирования.

В таблице 5.1 представлен пример дискретизации универсума на 13 уровней с семью термами [5].

Таблица 5.1

Уровень	Диапазон	ОБ	ОС	ОМ	НОЛЬ	ПМ	ПС	ПБ
-6	$x_0 \leq -3,2$	1.0	0.3	0.0	0.0	0.0	0.0	0.0
-5	$-3.2 < x_0 \leq -1.6$	0.7	0.7	0.0	0.0	0.0	0.0	0.0
-4	$-1.6 < x_0 \leq -0.8$	0.3	1.0	0.3	0.0	0.0	0.0	0.0
-3	$-0.8 < x_0 \leq -0.4$	0.0	0.7	0.7	0.0	0.0	0.0	0.0
-2	$-0.4 < x_0 \leq -0.2$	0.0	0.3	1.0	0.3	0.0	0.0	0.0
-1	$-0.2 < x_0 \leq -0.1$	0.0	0.0	0.7	0.7	0.0	0.0	0.0
0	$-0.1 < x_0 \leq +0.1$	0.0	0.0	0.3	1.0	0.3	0.0	0.0
1	$+0.1 < x_0 \leq +0.2$	0.0	0.0	0.0	0.7	0.7	0.0	0.0
2	$+0.2 < x_0 \leq +0.4$	0.0	0.0	0.0	0.3	1.0	0.3	0.0
3	$+0.4 < x_0 \leq +0.8$	0.0	0.0	0.0	0.0	0.7	0.7	0.0
4	$+0.8 < x_0 \leq +1.6$	0.0	0.0	0.0	0.0	0.3	1.0	0.3
5	$+1.6 < x_0 \leq +3.2$	0.0	0.0	0.0	0.0	0.0	0.7	0.7
6	$3.2 \leq x_0$	0.0	0.0	0.0	0.0	0.0	0.3	1.0

Нормализация универсума связана с дискретизацией последнего на конечное число сегментов, каждый из которых отображается в подходящий (соответствующий) сегмент нормализованного универсума. В табл. 5.2 представлена нормализация универсума  $[-6.0, +4,5)$ , который трансформируется в нормализованный интервал  $[-1,1]$  [5].

Нечеткое разделение определяет сколько термов будет участвовать в терм-множестве. Мощность терм-множества пространства входов определяет максимальное количество нечетких правил регулирования в базе знаний. Обычно для выбора оптимального нечеткого разбиения используется эвристическая процедура проб и ошибок.

Нормализация и первичные нечеткие множества, использующие функциональное определение

Нормализованный универсум	Нормализованные сегменты	Диапазон	$u_j$	$\sigma_j$	Первичные нечеткие множества
[-1.0,+1.0]	[-1.0,-0,5]	[-6.9,-4.1]	-1.0	0.4	ОБ
	[-0.5,-0,3]	[-4.1,-2.2]	-0.5	0.2	ОС
	[-0.3,-0.0]	[-2.2,-0.0]	-0.2	0.2	ОМ
	[-0.0,+0.2]	[-0.0,+1.0]	0.0	0.2	НОЛЬ
	[+0.2,+0.6]	[+1.0,+2.5]	0.2	0.2	ПМ
	[+0.6,+1.0]	[+2.5,+4.5]	0.5	0.2	ПС
			1.0	0.4	ПБ

Для определения функций принадлежности первичного нечеткого множества в зависимости от типа универсума, (дискретный или непрерывный) имеются два метода: числовой и функциональный.

В первом случае, степень функции принадлежности нечеткого множества представляется как числовой вектор, размерность которого зависит от степени дискретизации.

При этом функция принадлежности имеет вид

$$\mu_A(u) = \sum_{i=1}^n a_i / u_i.$$

Во втором случае функции принадлежности нечетких множеств имеют определенную функциональную форму, обычно, колоколообразную, треугольную, трапециодальную, и т.д. Такие функции используются в нечетком логическом контроллере (НЛК), так как позволяют манипулировать собой при помощи нечеткой арифметики.

### **5.3. Общая методология построения систем нечеткого управления**

#### **База знаний.**

Набор правил нечеткого управления, которые выражаются как нечеткие условные утверждения, формируют базу правил или набор правил НЛК.

Для проектирования БЗ необходимо определить переменные состояния (входные переменные), и переменные управления (выходные переменные), источники типов нечетких правил регулирования и др. Имеются 4 режима добывания нечетких правил [5,6].

а) Нечеткие правила управления имеют форму нечетких условных утверждений, которые соотносят переменные состояния в антецеденте и управляющие переменные процесса в консеквентах.

Формирование нечетких правил управления может быть достигнуто посредством двух эвристических подходов. Наиболее общий использует формализацию человеческой экспертизы. Типичным примером такой формализации является руководство к управлению технологическим процессом. Другой подход включает опрос опытных экспертов или операторов с помощью тщательно подготовленных вопросников.

б) Управляющие действия оператора.

В данном случае нечеткие правила могут быть выведены из наблюдения за управляющей деятельностью человека и выявления связи вход-выход.

в) Нечеткая модель процесса.

В лингвистическом подходе лингвистическое описание динамических характеристик управляемого процесса может быть рассмотрена как нечеткая модель процесса. Основываясь на нечеткой модели, мы можем генерировать сеть нечетких управляющих правил для поддержания оптимального функционирования динамической системы.

г) Обучение.

С помощью само-организирующихся контроллеров [5,7] можно формировать БЗ НЛК. Такие контроллеры имеют иерархическую структуру, состоящую из двух баз правил. Первая база – общая база правил НЛК. Вторая состоит из мета-правил,



которые демонстрируют человекоподобную способность обучения к созданию и изменению общей базы правил на основе желаемого функционирования системы в целом.

Современные подходы обучения, в частности neuro-fuzzy, генетические подходы позволяют извлекать знания и оптимизировать БЗ НЛК.

При конструировании НЛК систем используются два типа нечетких правил регулирования. Для многих НЛК используются нечеткие правила, которые для системы со многими входами и одним выходом имеют вид

ЕСЛИ  $x_1$  ЕСТЬ  $A_{11}$  И  $x_2$  ЕСТЬ  $A_{12}$  ... И  $x_n$  ЕСТЬ  $A_{1n}$  ТО  $y$  ЕСТЬ  $B_1$

ТАКЖЕ

ЕСЛИ  $x_1$  ЕСТЬ  $A_{21}$  И  $x_2$  ЕСТЬ  $A_{22}$  ... И  $x_n$  ЕСТЬ  $A_{2n}$  ТО  $y$  ЕСТЬ  $B_2$

.....

ТАКЖЕ

ЕСЛИ  $x_1$  ЕСТЬ  $A_{m1}$  И  $x_2$  ЕСТЬ  $A_{m2}$  ... И  $x_n$  ЕСТЬ  $A_{mn}$  ТО  $y$  ЕСТЬ  $B_m$

где  $x_1, x_2, \dots, x_n$  лингвистические переменные, представляющие переменные состояния и управления процесса;  $A_{11}, \dots, A_{in}$  и  $B_i$  - лингвистические значения переменных  $x_1, x_2, \dots, x_n$  в универсумах  $U, V$  и  $W$ , соответственно.

Нередко применяется следующая форма набора нечетких правил регулирования (с использованием ТСК-модели)

$$R_i: \text{ЕСЛИ } x_{i1} \text{ ЕСТЬ } A_{i1} \text{ И } x_{in} \text{ ЕСТЬ } A_{in} \text{ ТО } y = f_i(x_1, \dots, x_n). \quad (5.1)$$

Нечеткие правила управления этого типа, которые называются также «нечеткими правилами управления на основе оценки состояния» оценивают состояние процесса (например, состояние, ошибку состояния, интеграл состояния) в момент времени  $t$  как функцию  $(x_1, x_2, \dots, x_n, y)$  и правил управления.

Другой тип правил, используемых в так называемых «нечетких управлениях на основе оценки объекта» или «предикативное нечеткое управление» имеет вид

$$R_i: \text{ЕСЛИ } (u \text{ ЕСТЬ } C_i \rightarrow (x \text{ ЕСТЬ } A_i \text{ И } y \text{ ЕСТЬ } B_i \text{ ТО } u \text{ ЕСТЬ } C_i)). \quad (5.2)$$

Команда управления выводится из объективной оценки результатов нечеткого управления, которые удовлетворяют желаемым состояниям и целям. Команда управления  $U$  принимает четкое множество в качестве своего значения, и  $x, y$  являются

критериями качества для оценки  $i$ -го правила с помощью значений «хорошо» или «плохо». Наиболее подходящее правило управления выбирается на основе прогнозирования результатов  $(x, y)$ , соответствующих каждой команде управления  $C_i$ .

### Механизм вывода.

Ядром нечетких контроллеров является механизм вывода. Механизм вывода нечетких регуляторов базируется на нечеткой импликации, которая детально рассмотрена в гл. 4. Рассмотрим следующие определения [5].

**Определение 5.5. Треугольная норма  $*$**  - это двухместная функция из  $[0,1] \times [0,1]$  в  $[0,1]$ , т.е.  $*: [0,1] \times [0,1] \rightarrow [0,1]$ , которая включает пересечение, алгебраическое произведение, ограниченное произведение и строгое (drastic) произведение (см.3.3). Наибольшая треугольная норма – пересечение, а наименьшее – строгое (drastic) произведение. Операции, связанные с треугольными нормами определяются для всех  $x, y \in [0,1]$ :

Пересечение	$x \wedge y = \min(x, y)$ .
Алгебраическое произведение	$x \cdot y = xy$ .
Ограниченное произведение	$x \otimes y = \max\{0, x + y - 1\}$ .
Строгое (drastic) произведение	$x \cap y = \begin{cases} x, & y = 1 \\ y, & x = 1 \\ 0, & x, y < 1 \end{cases}$

**Определение 5.6. Треугольная конорма  $+$**  - это двухместная функция из  $[0,1] \times [0,1]$  в  $[0,1]$ , т.е.  $+: [0,1] \times [0,1] \rightarrow [0,1]$ , которая включает объединение, алгебраическую сумму, ограниченную сумму, строгую (drastic) и дизъюнктивную сумму (см.3.3). Операции, связанные с треугольными конормами, определены для всех  $x, y \in [0,1]$ :

Объединение	$x \vee y = \max(x, y)$ .
Алгебраическая сумма	$x \hat{+} y = x + y - xy$ .
Ограниченная сумма	$x \oplus y = \min(1, x + y)$ .
Строгая (drastic) сумма	$x \cup y = \begin{cases} x, & y = 0 \\ y, & x = 0 \\ 0, & x, y > 0 \end{cases}$
Дизъюнктивная сумма	$x \cap y = \max\{\min(x, 1 - y), \min(1 - x, y)\}$ .

Треугольные нормы используются для определения конъюнкций в приближенных рассуждениях, когда как треугольные конормы служат для определения дизъюнкций.

**Определение 5.7. Нечеткая конъюнкция.** Нечеткая конъюнкция определяется для всех  $u \in U$  и  $v \in V$  посредством

$$A \rightarrow B = A \times B = \int_{U \times V} \mu_A(u) * \mu_B(v) / (u, v),$$

где  $*$  - оператор, представляющий треугольную норму.

**Определение 5.8. Нечеткая дизъюнкция.** Нечеткая дизъюнкция определяется для всех  $u \in U$  и  $v \in V$  посредством

$$A \rightarrow B = A \times B = \int_{U \times V} \mu_A(u) + \mu_B(v) / (u, v),$$

где  $+$  - оператор, представляющий треугольную конорму.

**Определение 5.9. Операторы соединения предложений «и», «также».**

В большинстве существующих НЛК связка «и» обычно реализуется как нечеткая конъюнкция в пространстве картезианского произведения, в котором отмеченные переменные принимают значения в различных универсумах. Если нечеткая система характеризуется множеством нечетких правил управления, то порядок правил не имеет значения. Это требует, чтобы связка «также» имела свойства коммутативности и ассоциативности.

В этой связи необходимо отметить, что операторы в треугольных нормах и конормах обладают этими свойствами и таким образом квалифицируются в качестве кандидатов для интерпретации связки «также». Вообще целесообразно использовать треугольные конормы совместно с нечеткими конъюнкциями и дизъюнкциями, а треугольные нормы – совместно с нечеткой импликацией.

**Определение 5.10. Оператор композиции.**

В общей форме оператор композиции может быть выражен как «sup-star» композиция, где «star» обозначает какой-либо оператор, например,  $\min$ , умножение и т.д., который адекватен конкретному применению. В литературе используются 4 типа операторов композиции для композиционного правила вывода, а именно:

операция sup-min [8];

операция sup-произведение [9];

операция sup-ограниченное произведение [10];

операция sup-строгое произведение [10].

Резюмируя вышеизложенное можно сформулировать следующую последовательность этапов действия механизма вывода [5].

1) Сравнить текущее четкое значение каждой входной переменной с функцией принадлежности этой переменной в условных частях правил для нахождения степени принадлежности переменной каждому из ее лингвистических термов. (Этот этап часто называется фаззификацией).

2) Комбинировать (с помощью специфического T-norm оператора, обычно произведения или  $\min$ ) степени принадлежности условной части каждого правила для определения силы активации (firing strength).

3) Генерировать следствие (нечеткое либо четкое) каждого правила в зависимости от силы активации.

На примере следующих трех типов механизмов вывода рассмотрим процедуру нечеткого вывода [11].

**Тип 1.** Выход каждого активизированного правила определяется на основе силы активации (произведение или минимум степеней принадлежности) и функций принадлежности правой части соответствующего правила. Выходные функции принадлежности выходов, использованные в этой схеме, должны быть монотонными [12].

**Тип 2.** Общий нечеткий выход выводится с применением операции  $\max$  к нечетким выходам каждого правила. Каждый выход определяется минимумом силы активации и функций принадлежности правой части каждого правила. Ряд схем были предложены для определения результирующего четкого выхода на основе нечеткого выхода. В число их входят, например, центр площади, бисектор площади, середина максимума, критерий максимума [5,13].

**Тип 3.** Используются нечеткие ЕСЛИ-ТО правила Такаги и Сугено [14]. Выход каждого правила – это линейная или нелинейная комбинация выходных переменных плюс константа, а конечный выход – это «взвешенное среднее» выходов каждого правила.

База знаний, при этом, принимается как

$$\begin{aligned} \text{ЕСЛИ } x \text{ ЕСТЬ } A_1 \text{ И } y \text{ ЕСТЬ } B_1, \text{ ТО } z \text{ ЕСТЬ } C_1 \\ \text{ЕСЛИ } x \text{ ЕСТЬ } A_2 \text{ И } y \text{ ЕСТЬ } B_2, \text{ ТО } z \text{ ЕСТЬ } C_2 \end{aligned} \quad (5.3)$$

(для первых двух)

$$\begin{aligned} \text{ЕСЛИ } x \text{ ЕСТЬ } A_1 \text{ И } y \text{ ЕСТЬ } B_1, \text{ ТО } z_1 = ax + by + c \\ \text{ЕСЛИ } x \text{ ЕСТЬ } A_2 \text{ И } y \text{ ЕСТЬ } B_2, \text{ ТО } z_2 = px + qy + r \end{aligned} \quad (5.4)$$

(для третьего)

### Дефаззификация.

**Определение 5.11. Дефаззификация.** Дефаззификация – это отображение из допустимого пространства нечетких управляющих воздействий в пространство четких управляющих воздействий.

В настоящее время наиболее часто используемыми стратегиями являются критерий максимума, середина максимума (СМ) и центр области (ЦО).

Критерий максимума определяет точку, в которой распределение возможностей управляющего воздействия достигает максимального значения.

На рис.5.2 представлены результаты нечеткого логического вывода для всех трех типов указанных выше типов механизмов вывода.

СМ – стратегия генерирует управляющее действие, которое представляет среднее значение всех локальных управляющих воздействий, вычисляемых по максимальному значению функции принадлежности каждого нечеткого управляющего воздействия. В частности, в дискретном случае управляющее воздействие может выражаться как

$$z_0 = \sum_{j=1}^l \frac{W_j}{l}, \quad (5.5)$$

где  $W_j$  - носитель значения, в которой функция принадлежности достигает максимального значения  $\mu_z(W_j)$  и  $l$  - число таких носителей значений.

Широко используемая стратегия ЦО генерирует центр тяжести распределения возможностей результирующего управляющего воздействия. В случае дискретного универсума этот метод часто дает:

$$z_0 = \frac{\sum_{j=1}^n \mu_z(W_j)W_j}{\sum_{j=1}^n \mu_z(W_j)}, \quad (5.6)$$



## 5.4. Нечеткий гибридный контроллер

На рис. 5.3 представлена структура гибридной системы управления, основанной на схеме нечеткого регулирования параметров традиционных контроллеров.

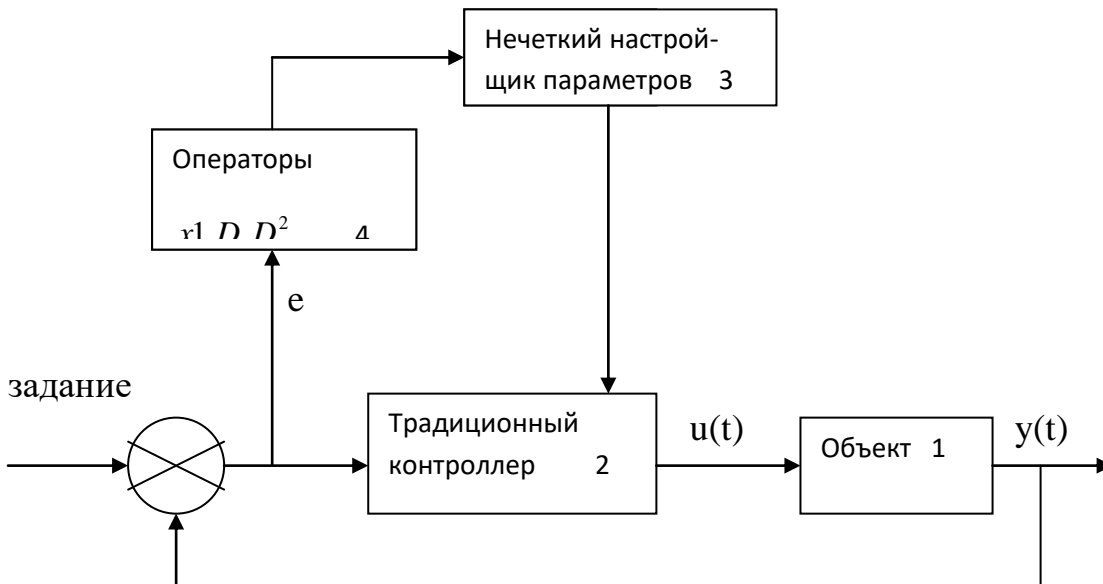


Рис. 5.3. Схема нечеткого регулирования параметров контроллера

Объект 1 управляется традиционным контроллером 2 (в частности – ПИД-контроллером) параметры которого генерируется нечетким настройщиком параметров 3 на основе информации об ошибке и ее производных, получаемых с блока 4. Суть работы нечеткого настройщика заключается в настраивании параметров контроллера в реальном режиме с использованием базы нечетких правил и нечеткого вывода. Не нарушая общности, далее будет рассматриваться процедура синтеза системы, включающей ПИД-контроллер. Дискретный алгоритм ПИД-контроллера имеет вид:

$$u(k) = K_p e(k) + K_I T_s \sum_{i=1}^n e(i) + \frac{K_d}{T_s} \Delta e(k),$$

где  $u(k)$  – управляющее воздействие,  $e(k)$  – ошибка между заданием и текущим выходом объекта,  $T_s$  – период квантования,  $\Delta e(k) = e(k) - e(k-1)$ . Параметры  $K_p, K_I, K_d$  – определяются совокупностью нечетких правил типа

Если  $e(k)$  есть  $A_i$  и  $\Delta e(k)$  есть  $B_i$ , ТО  $K'_p$  есть  $C_i$ ,  $K'_d$  есть  $D_i$   
И  $K'_i$  есть  $E_i$

где  $K'_p, K'_i, K'_d$  - нормализованные значения искомым параметров.

Можно использовать другие нечеткие правила настройки параметров ПИД-контроллера, например

Если  $e(k)$  есть  $A_i$  и  $\Delta e(k)$  есть  $B_i$ , ТО

$$u(k) = K_{p0}^i e(k) + K_{I0}^i T_S \sum_j e(j) + \frac{K_{d0}^i}{T_S} \Delta e(k), \quad (5.7)$$

где  $K_{p0}^i, K_{I0}^i, K_{d0}^i$  - константы.

В [16] используются следующие нечеткие правила настройки

Если  $e(k)$  есть  $A_i$  и  $\Delta e(k)$  есть  $B_i$ , ТО  $K'_p$  есть  $C_i$ ,  $K'_d$  есть  $D_i$

$$\text{И } \alpha = \alpha_i, \quad i = \overline{1, m}, \quad (5.8)$$

где  $\alpha = T_i/T_d$ ,  $A_i, B_i, C_i$  и  $D_i$  - нечеткие множества,  $\alpha_i$  - константа.

Нечеткие правила (5.7)-(5.8) могут быть получены на базе экспертных знаний.

Например, нечеткое правило вокруг  $a_1$  можно сформулировать как

ЕСЛИ  $e(k)$  есть ПБ и  $\Delta e(k)$  есть НОЛЬ, ТО  $K'_p$  есть Б,  $K'_d$  есть М

И  $\alpha = 2$ ,

$\alpha$  - можно рассматривать как нечеткий синглтон.

Вокруг точки  $b_1$  соответствующим ходом рассуждения можно получить

ЕСЛИ  $e(k)$  есть НОЛЬ и  $\Delta e(k)$  есть ОБ, ТО  $K'_p$  есть М,  $K'_d$  есть Б

И  $\alpha = 5$ .

Таким образом, получается совокупность нечетких правил настройки, которые описываются в виде таблиц 13.1-13.3 [16].

Значение истинности силы активации  $i$ -го правила  $\mu_i$  определяется как

$$\mu_i = \mu_{A_i}[e_k] \wedge \mu_{B_i}[\Delta e_k]. \quad (5.9)$$



Используя вычисленные значения  $\mu_i$  и выбирая способ импликации можно определить для каждого правила  $K_p$  и  $K_d$  и далее  $T_i$ .

Здесь ОБ – отрицательно большой, ОС – отрицательно средний, ОМ – отрицательно маленький, ПМ – положительно маленький, ПС – положительно средний, ПБ – положительно большой.

Таблица 5.3

Нечеткие правила обучения для  $K'_p$

		$\Delta e(k)$						
		ОБ	ОС	ОМ	НОЛЬ	ПМ	ПС	ПБ
e(k)	ОБ	Б	Б	Б	Б	Б	Б	Б
	ОС	М	Б	Б	Б	Б	Б	М
	ОМ	М	М	Б	Б	Б	М	М
	НОЛЬ	М	М	М	Б	М	М	М
	ПМ	М	М	Б	Б	Б	М	М
	ПС	М	Б	Б	Б	Б	Б	М
	ПБ	Б	Б	Б	Б	Б	Б	Б

Таблица 5.4

Нечеткие правила обучения для  $K'_d$

		$\Delta e(k)$						
		ОБ	ОС	ОМ	НОЛЬ	ПМ	ПС	ПБ
e(k)	ОБ	М	М	М	М	М	М	М
	ОС	Б	Б	М	М	М	Б	Б
	ОМ	Б	Б	Б	Б	Б	Б	Б
	НОЛЬ	Б	Б	Б	Б	Б	Б	Б
	ПМ	Б	Б	Б	М	Б	Б	Б
	ПС	Б	Б	М	М	М	Б	Б
	ПБ	М	М	М	М	М	М	М

Нечеткие правила обучения для  $\alpha$ 

		$\Delta e(k)$						
		ОБ	ОС	ОМ	НОЛЬ	ПМ	ПС	ПБ
e(k)	ОБ	2	2	2	2	2	2	2
	ОС	3	3	2	2	2	3	3
	ОМ	4	3	3	2	3	3	4
	НОЛЬ	5	4	3	3	3	4	5
	ПМ	4	3	3	2	3	3	4
	ПС	3	3	2	2	2	3	3
	ПБ	2	2	2	2	2	2	2

$K'_p$  определяется как [16]

$$K'_p = \sum_{i=1}^m \mu_i K'_{pi} \quad (5.10)$$

Аналогичным образом определяется  $K'_d$  и  $\alpha$

$$K'_d = \sum_{i=1}^n \mu_i K'_{di}; \quad \sum_{i=1}^m \mu_i \alpha_i. \quad (5.11)$$

Истинные ненормализованные значения  $K'_p$ ,  $K'_d$  и  $\alpha$  определяются как

$$\begin{aligned} K_p &= (K_{p \max} - K_{p \min}) K'_p + K_{p \min} \\ K_d &= (K_{d \max} - K_{d \min}) K'_d + K_{d \min} \\ K_i &= K_p^2 (\alpha K_d) \end{aligned} \quad (5.12)$$

**Пример** [16]. Представим себе, что объект управления описывается передаточной функцией

$$G(S) = \frac{4,228}{(S + 0,5)(S^2 + 1,64S + 8,456)}. \quad (5.13)$$

Компьютерная симуляция проведена следующим образом. Сначала определяются текущие значения ошибки  $e$  и ее первой производной. Потом применением процедуры нечеткого вывода определяются значения  $K'_p$ ,  $K'_d$  и  $\alpha$ . И наконец с использованием (13.6) определяются искомые параметры ПИД-контроллера.

## 5.5. Нечеткие координатно-параметрические адаптивные системы управления

Возникновение теории нечетких множеств определило новое направление в развитии теории управления. Уже первые исследования выявили реальные выгоды учитывания нечеткой информации, которая не может формализовываться в традиционных системах управления, содержащая нечеткий контроллер, может рассматриваться как нечеткий аналог системы управления с координатной обратной связью. Более того, если нечеткий контроллер имеет входной и выходной интерфейсы, то он становится практически аналогичным контроллеру, реализующему некий нелинейный алгоритм [17,18].

В этом параграфе мы предполагаем, что нечеткий контроллер с координатной обратной связью реализуется любым известным методом, который иногда не обеспечивает желаемого управления вследствие возмущений в системе, а также неадекватности модели процесса. В этом случае представляется целесообразным использовать нечеткую систему, содержащую нечеткий контроллер с координатной обратной связью, также как и нечеткий контроллер с параметрической обратной связью.

В настоящее время имеются многочисленные важные публикации по проектированию самонастраивающихся [19] бинарных [20] и лингвистических самоорганизующихся [21] систем управления. Основываясь на научных и методических принципах проектирования таких систем, решается задача синтеза нечеткой координатно-параметрической системы управления.

Предполагается, что объект управления подвержен воздействию вектора возмущений ( $F$ ), причем:

$$\|\vec{F}\| \leq M . \quad (5.14)$$

Исходя из этого, а также учитывая сложность описания объекта аналитической моделью, объект управления описывается лингвистической (нечеткой) моделью первого порядка или, придерживаясь установившейся терминологии, таблицей лингвистических правил (ТЛП) вида:

$$\tilde{X} = \varphi(\tilde{X}, \tilde{U}), \quad (5.15)$$

где  $\tilde{X}$  и  $\tilde{X}$  - нечеткие переменные, соответственно, выхода и скорости изменения выхода объекта,  $\tilde{U}$  - нечеткая переменная входа, а  $\varphi$  - нестационарный оператор, устанавливающий соответствие между упомянутыми переменными, например:

Если  $X$  есть  $\tilde{X}_i$  и если  $\tilde{U}$  есть  $\tilde{U}_j$ , то  $\tilde{X}$  есть  $\tilde{X}_k$ ,  $i = \overline{1, N_1}$ ,  $j = \overline{1, N_2}$ ,  $k = \overline{1, N_3}$

Пусть лингвистическим термам нечетких переменных  $\tilde{X}_i, \tilde{U}_j$  и  $\tilde{X}_k$  соответствуют экспоненциальные функции принадлежности:

$$\mu_i(x(t)) = \exp(-g_{1i}|x(t) - \bar{x}_i|), \quad i = \overline{1, N_1},$$

$$\mu_j(u(t)) = \exp(-g_{2j}|u(t) - \bar{u}_j|), \quad j = \overline{1, N_2},$$

$$\mu_k(\dot{x}(t)) = \exp(-g_{3k}|\dot{x}(t) - \bar{\dot{x}}_k|), \quad k = \overline{1, N_3},$$

где  $\dot{x}(t), u(t)$  и  $x(t)$  - базовые переменные соответствующих лингвистических переменных;  $g_{1i}, g_{2j}$  и  $g_{3k}$  определяются заданием некоторого  $\alpha$ -уровня нечеткого множества (в данном случае  $\alpha = 0,5$ );  $\bar{x}_i, \bar{u}_j$  и  $\bar{\dot{x}}_k$  - элементы соответствующих множеств, степень принадлежности которых равна единице.

В случае описания объекта моделью выше первого порядка, возможна декомпозиция ТЛП на ряд таблиц. Предполагаем, что  $\tilde{U}$  в (5.15) может быть представлена следующим образом [51]:

$$\tilde{U} = f(\tilde{U}_0, \tilde{R}),$$

где  $\tilde{U}_0$  - лингвистическая переменная выхода нечеткого регулятора в контуре координатного управления (его наличие предполагается), а  $\tilde{R}$  - лингвистическая переменная адаптации, т.е. контура параметрического управления:

$$\tilde{R} = \psi(\tilde{X}, \tilde{X}, \tilde{M}).$$

Задача заключается в определении  $\tilde{U}$ , обеспечивающего требуемое качество регулирования, в частности  $\tilde{E} \rightarrow 0$  при  $t \rightarrow \infty$ .

Введем понятие обобщенного нечеткого настраиваемого объекта, понимая под ним замкнутую систему координатного управления (рис.5.4). Структура системы содержит: 1 – задатчик; 2 – устройство сравнения; 3 – регулятор нечеткого координатного управления; 4 – объект управления;  $\tilde{R}$  и  $\tilde{E}$  - соответственно, нечеткие вектора адаптации параметрического управления и ошибки.

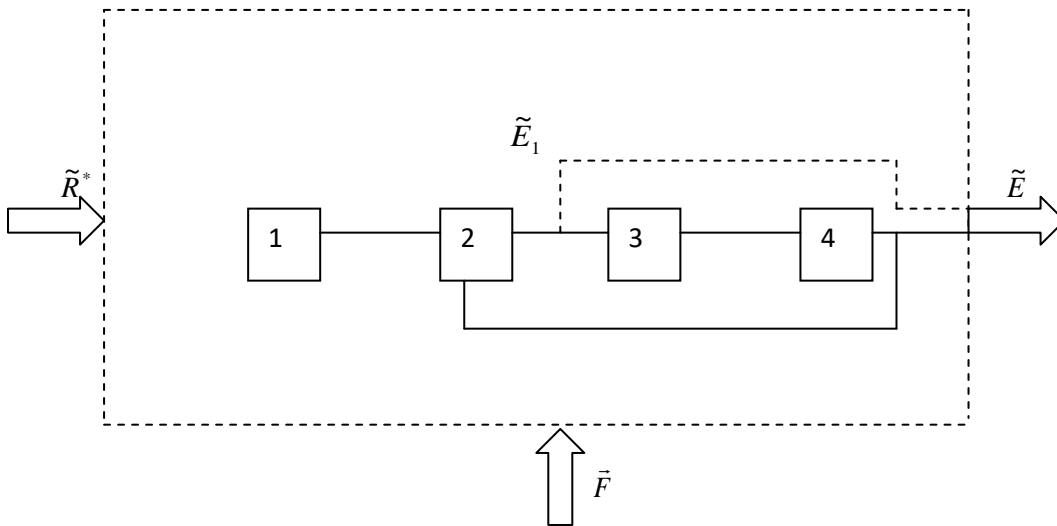


Рис.5.4. Структура обобщенного нечеткого настраиваемого объекта

Тогда структурную схему нечеткой САУ можно представить в виде (рис.5.5). Здесь А – задатчик эталонного движения системы; В – обобщенный нечеткий настраиваемый объект; С – устройство параметрического управления, включающее в себя устройство сравнения реального и эталонного сигналов ошибки и так называемый «регулятор параметрического управления», который подстраивает параметры регулятора координатного управления и объекта [22,23].

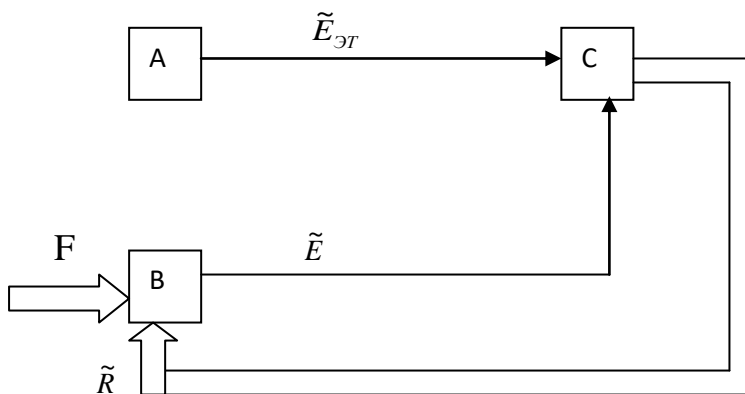


Рис. 5.5. Структурная схема нечеткой САУ

Разделив, как в [20], контуры координатного и параметрического управления или, иными словами, определив

координатную и оперативную обратные связи, можно рассматривать приведенную на рис. 5.6 систему, как нечеткую координатно-параметрическую САУ, реализующую оба вида связей. При этом устройство А представляет собой, согласно принятой терминологии, операторно-координатное звено, а С - звено координатно-операторного типа.

Рассмотрим возможные реализации задатчика эталонного движения (блок А). Он может задавать динамику изменения вектора ошибки либо как функцию времени, либо в виде дифференциального включения, а функциональные зависимости могут определяться либо в виде соответствий на лингвистических шкалах, либо в виде нечетких отношений. Другой подход связан с заданием желаемой области в фазовом пространстве, как это предложено в работе [23]; при этом вид желаемой области определяется практическими соображениями. Ограничим наше исследование рассмотрением случая, когда фазовое пространство задатчика лингвистическое. В этом случае по каждой оси откладывается конечный набор термов и пространство будет дискретным, состоящим из конечного числа «точек», а мощность этого пространства:

$$CardG = \prod_{i=1}^n CardG_i,$$

где  $G_i$  - множество термов по  $i$ -й оси. Схема на рис.5.6 преобразуется к следующему виду (рис.5.6), где  $\bar{\varepsilon}^*$  - лингвистический вектор ошибки (координаты изображающей точки -  $(\varepsilon, \varepsilon^*, \dots, \varepsilon^{(n)})$ );  $\rho$  - операторный сигнал, т.е. функционал, оценивающий «расстояние» от изображающей точки до границ желаемой области; С - блок адаптации, т.е. «нечеткий регулятор» параметрического управления.

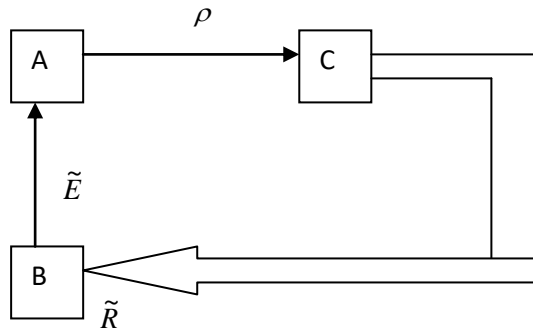


Рис.5.6. Структура нечеткой координатно-параметрической САУ

Лингвистическое фазовое пространство задатчика с заданными на нем граничными множествами для двумерного случая можно изобразить в виде ТЛП (Табл.5.6). Символом О в таблице обозначены точки, соответствующие наиболее быстро протекающему желаемому переходному процессу, а символ \* – точки, соответствующие наиболее медленному из желаемых переходных процессов. Так как лингвистические термы каждой оси обладают свойством упорядоченности, то им в соответствие можно поставить последовательность натуральных чисел (табл.5.6). Определим отношение порядка для векторов, первая компонента которых зафиксирована:

$$(S_i, T_1) < (S_i, T_2) < \dots < (S_i, T_j) < \dots < (S_i, T_k),$$

если

$$T_1 < T_2 < \dots < T_j < \dots < T_k; \quad i = \overline{1, L}, \quad j = \overline{1, K},$$

где L и K – мощность терм-множеств соответственно, ошибки и скорости изменения ошибки.

Таблица 5.6

		E						
		ОБ	ОС	ОМ	НОЛЬ	ПМ	ПС	ПБ
E*	ОБ		О	О				
	ОС							
	ОМ		*	*				
	НОЛЬ	*О			*О			*О
	ПМ					*	*	
	ПС							
	ПБ					О	О	

Точки, обладающие наибольшей второй компонентой при фиксированной первой, будем называть верхними граничными точками, а точки с наименьшей второй компонентой – нижними граничными точками. Нетрудно заметить, что если изображающая точка имеет вектор  $c$ , идентичной первой компоненте и значение второй компоненты, больше чем верхняя граничная точка, то для улучшения качества переходного процесса следует уменьшить сигнал управления. Аналогично, если вторая компонента изображающей точки меньше, чем у соответствующей нижней граничной точки, то сигнал управления необходимо увеличить. И наконец, для точек, вектора которых лежат между векторами верхней и нижней граничных точек или совпадают с ними, управление менять не нужно, так как движение системы является желаемым.

Определим функцию:

$$SAT(\bar{P}_{ab}) = \begin{cases} -1, & \bar{P}_{ab} \in \Omega_2 \\ 0, & \bar{P}_{ab} \in \Omega_3 \\ 1, & \bar{P}_{ab} \in \Omega_1 \end{cases} \quad (5.16)$$

где  $\bar{P}_{ab}$  - обозначение изображающей точки с вектором  $(a,b)$ , а  $\Omega_1, \Omega_2$  и  $\Omega_3$  - соответственно множества векторов, соответствующих режимам увеличения, уменьшения и неизменности управления. Эти множества можно назвать множествами, соответственно, положительного, отрицательного и нулевого знаков. Последнее можно назвать также желаемым множеством. Для определения «расстояния» изображающей точки от желаемого множества введем функционал следующего вида:

$$\rho(\bar{P}_{ab}, \Omega) = \left[ \begin{array}{l} \min(|a-c| + |b-d|) \\ (c, d) \in \Omega(\bar{P}) \end{array} \right] SAT(\bar{P}_{ab}), \quad (5.17)$$

где

$$\Omega(\bar{P}) = \begin{cases} \text{верхняя граница,} & \text{если } SAT(\bar{P}) = 1, \\ \text{нижняя граница,} & \text{если } SAT(\bar{P}) = -1. \end{cases}$$

Очевидно, что

$$\rho = 0, \text{ если } P_{ab} \in \Omega_3; \quad \rho > 0, \text{ если } P_{ab} \in \Omega_1; \quad \rho < 0, \text{ если } P_{ab} \in \Omega_2.$$



Можно записать

$$\rho(\bar{P}_{ab}, \Omega) = 0.$$

Таким образом, на основании (5.15), (5.16) и (5.17) можно сформировать множество термов корректирующего-адаптирующего сигнала, вырабатываемого «нечетким регулятором» параметрического управления:

$$\begin{aligned} R_{ij} &= \{r_{ij}, \mu_{ij}(r(t))\} \\ \mu_{ij}(r(t)) &= \exp(-g_{4ij}|r(t) - \bar{r}_{ij}|) \\ r_{ij} &= \lambda M \rho(\bar{P}_{ij}, \Omega) \quad i = \bar{1}, k, \quad j = \bar{1}, L \end{aligned} \quad (5.18)$$

где  $\lambda$  - масштабный коэффициент,  $K$  и  $L$  – мощности термножеств ошибки и скорости изменения ошибки. В результате имеем набор лингвистических правил вида:

Если  $E$  есть  $\tilde{E}_i$  И если  $\dot{E}$  есть  $\tilde{E}_j$ , То  $R$  есть  $\tilde{R}_{ij}$ ,

которые формируют ТЛП контура «нечеткого регулятора» параметрического управления.

Окончательно структура САУ с координатно-параметрическими «нечеткими регуляторами» примет вид как на рис. 5.7, где 1 – задатчик уставок; 2 – сравнивающее устройство и входной интерфейс темы; 3 – нечеткий регулятор основного контура координатного управления; 4 – лингвистический сумматор; 5 – выходной интерфейс системы; 6 – объект управления; 7 – контур адаптации, т.е. «нечеткий регулятор» параметрического управления.

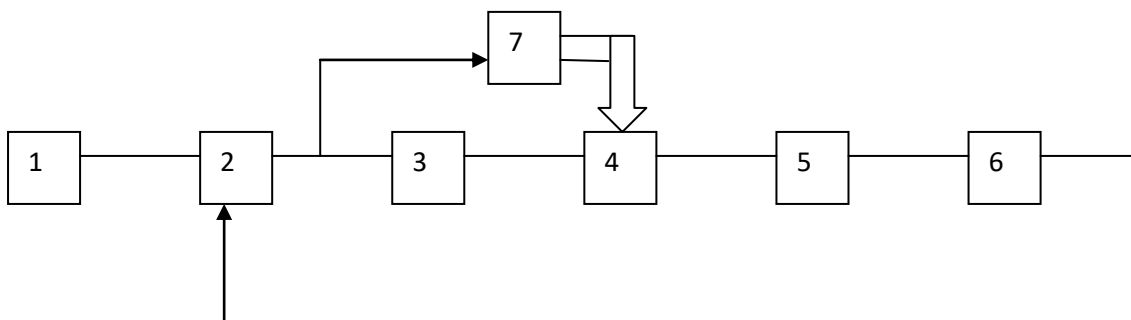


Рис.5.7. Окончательная структура САУ с координатно-параметрическим нечетким регулятором

Результирующая переменная управления  $u^*$  определяется соотношением:

$$\tilde{U} = \tilde{U}_0 \oplus \tilde{R},$$

где  $\oplus$  - символ операции, выполняемой лингвистическим сумматором.

Нечеткая адаптивная САУ с координатно-параметрическим управлением была применена для поддержания температурного режима атмосферного блока установки первичной переработки нефти. Необходимо отметить две особенности данного процесса, предопределившие синтез упомянутой САУ.

1. Даже в условиях нормальной эксплуатации установки, процессу присуще разнообразие перерабатываемого сырья, которое поступает на установку с различных месторождений. При этом, прежде чем подать сырье на установку, нефть с различных месторождений смешивается в резервуарах товарного парка, т.е. при одинаковом удельном весе сырье может иметь разный качественный состав. В этом случае, управляющая стратегия операторов установки во многом определяется знанием состава, месторождения поступающего сырья, его особенностей и т.д. Эти соображения и предопределили попытку описания объекта лингвистическими моделями, в построении которых участие принимали эксперты. Например, в табл.5.7. приведена лингвистическая модель динамики объекта по каналу «расход орошения – температура верха колонны». На основе этой таблицы по разработанной процедуре, синтезирован нечеткий регулятор, ТЛП которого представлена в табл.5.8. (в ТЛП использованы следующие термы: ОБ – отрицательно-большой, ОС – отрицательно-средний, ОМ – отрицательно малый, НО - ноль. И аналогично для положительного диапазона).

2. На ход технологического процесса оказывает влияние ряд факторов, которые по сути являются возмущениями. Прежде всего следует отметить частую смену производительности установки, что приводит к существенным изменениям динамики объекта. Это связано как с нестабильностью поставок сырья на установку, так и нестабильностью работы сырьевых насосов и изменением гидравлического сопротивления трубопроводов.

Таблица 5.7.

		$\tilde{X}$							
$\tilde{Y}$		ОБ	ОС	ОМ	НОЛЬ	ПМ	ПС	ПБ	
	ОБ	НОЛЬ	ОМ	ОС	ОБ				
	ОС	ПМ	НОЛЬ	ОМ	ОС				
	ОМ	ПС	ПМ	НОЛЬ	ОМ				
	НОЛЬ	ПБ	ПС	ПМ	НОЛЬ	ОМ	ОС	ОБ	
	ПМ				ПМ	НОЛЬ	ОМ	ОС	
	ПС				ПС	ПМ	НОЛЬ	ОМ	
	ПБ				ПБ	ПС	ПМ	НОЛЬ	
		$\tilde{X}$							

Таблица 5.8.

		$\tilde{E}$							
$\tilde{E}$		ОБ	ОС	ОМ	НОЛЬ	ПМ	ПС	ПБ	
	ОБ	ОБ	ОБ		ПБ				НОЛЬ
	ОС		ОБ	ПБ	ПС		НОЛЬ	ОМ	
	ОМ		ПБ	ПС	ПМ	НОЛЬ	ОМ	ОС	
	НОЛЬ	ПБ	ПС	ПМ	НОЛЬ	ОМ	ОС	ОБ	
	ПМ	ПС	ПМ	НОЛЬ	ОМ	ОС	ОБ		
	ПС	ПМ	НОЛЬ		ОС	ОБ			
	ПБ	НОЛЬ			ОБ		ОБ	ОБ	
		$\tilde{U}_0$							

Исходя из изложенного, был осуществлен синтез контура параметрического управления по предложенному способу. Синтезированная ТЛП представлена в табл. 5.9. Как видно, грубо ТЛП можно разбить на три зоны: зона положительного корректирующего сигнала ( $\Pi_i, i = \overline{1,9}$ ), зона отрицательного корректирующего сигнала ( $O_i, i = \overline{1,9}$ ) и зона нулевого корректирующего сигнала (НОЛЬ), которые определены функцией  $SAT(P)$ . Элементы ТЛП (терм корректирующего сигнала) построены следующим образом:

$$O_i : \mu(r(t)) = \exp(-g|r(t) + r_i|); SAT(\bar{P}) = -1; i = \overline{1,9};$$

$$\Pi_i : \mu(r(t)) = \exp(-g|r(t) - \bar{r}_i|); SAT(\bar{P}) = 1; i = \overline{1,9};$$

$$НОЛЬ : \mu(r(t)) = \exp(-g|r(t)|); SAT(\bar{P}) = 0.$$

Таблица 5.9

	$\varepsilon 1$	$\varepsilon 2$	$\varepsilon 3$	$\varepsilon 4$	$\varepsilon 5$	$\varepsilon 6$	$\varepsilon 7$	$\varepsilon 8$	$\varepsilon 9$	$\varepsilon 10$	$\varepsilon 11$	$\varepsilon 12$	$\varepsilon 13$	$\varepsilon 14$	$\varepsilon 1$
$\dot{\varepsilon} 1$	04	03	02	01	02	03	04	05	06	07	08	09	08	07	08
$\dot{\varepsilon} 2$	03	02	01	HO	01	02	03	04	05	06	07	08	07	06	05
$\dot{\varepsilon} 3$	02	01	HO	HO	HO	01	02	03	04	05	06	07	06	05	04
$\dot{\varepsilon} 4$	01	HO	HO	HO	HO	HO	01	02	03	04	05	06	05	04	03
$\dot{\varepsilon} 5$	HO	HO	HO	П1	HO	HO	HO	01	02	03	04	05	04	03	02
$\dot{\varepsilon} 6$	HO	HO	П1	П2	П1	HO	HO	HO	01	02	03	04	03	02	01
$\dot{\varepsilon} 7$	HO	П1	П2	П3	П2	П1	HO	HO	HO	01	02	03	02	01	HO
$\dot{\varepsilon} 8$	П1	П2	П3	П4	П3	П2	П1	HO	HO	HO	01	02	01	HO	HO
$\dot{\varepsilon} 9$	П2	П3	П4	П5	П4	П3	П2	П1	HO	HO	HO	01	HO	HO	HO
$\dot{\varepsilon} 10$	П3	П4	П5	П6	П5	П4	П3	П2	П1	HO	HO	HO	HO	HO	П1
$\dot{\varepsilon} 11$	П4	П5	П6	П7	П6	П5	П4	П3	П2	П1	HO	HO	HO	П1	П2
$\dot{\varepsilon} 12$	П5	П6	П7	П8	П7	П6	П5	П4	П3	П2	П1	HO	П1	П2	П3
$\dot{\varepsilon} 13$	П6	П7	П8	П9	П8	П7	П6	П5	П4	П3	П2	П1	П2	П3	П4

Расчетные значения  $\bar{r}_i$ ,  $i = \overline{1,9}$  приведены в таблице 5.10, а  $g=107,3$ .

Таблица 5.10

$\bar{r} 1$	$\bar{r} 2$	$\bar{r} 3$	$\bar{r} 4$	$\bar{r} 5$	$\bar{r} 6$	$\bar{r} 7$	$\bar{r} 8$	$\bar{r} 9$
0,045	0,075	0,105	0,135	0,165	0,195	0,225	0,255	0,285

Применение контура параметрического управления улучшило динамические характеристики САУ, что было выявлено в ходе опытно-промышленной эксплуатации предложенной системы.

## 5.6. Устройства управления в интеллектуальных системах управления.

### Управление на основе нечеткой логики

Основной сложностью аналитического исследования устойчивости систем управления в которых используется блоки, реализующие механизм нечеткого вывода, является существенная нелинейность данных блоков, как правило, не допускающая линеаризации. Непосредственное применение известных аналитических методов исследования устойчивости, в частности второго метода Ляпунова, к системам с нечеткой логикой приводит к громоздким результатам. Указанное обстоятельство способствовало ослаблению интереса к данным методам [1].

Ниже изложен подход, позволяющий получать «грубые», достаточно просто связанные с параметрами системы, оценки устойчивости сравнительно несложными средствами.

**1. Постановка задачи.** Рассмотрим замкнутую автономную нелинейную импульсную систему автоматического управления, приведенную на рис.5.8.

Система состоит из линейного непрерывного стационарного динамического звена (ЛДЗ), импульсного элемента (ИЭ) с фиксатором нулевого порядка ( $\Phi_0$ ) и нечеткого логического регулятора (НЛР), реализующего статическую нелинейную зависимость между своим входным  $\vec{x}$  и выходным сигналами.

Сигнал  $u^*$  на выходе фиксатора нулевого порядка  $\Phi_0$  не изменятся между моментами срабатывания импульсного элемента ИЭ  $t_k(k=,1,2,...)$  и равен сигналу на входе ИЭ в момент срабатывания

$$u^*(t) = e(t_k) \text{ при } t \in [t_k, t_{k+1}] \quad (5.19)$$

Характеристика НЛР определяется набором нечетких продукционных правил:

П<sub>1</sub>: если  $x^1$  есть  $A_{11}$  и  $x^2$  есть  $A_{12}$ , и ...  $x^n$  есть  $A_{1n}$ , то  $u=a_1$ ;

П<sub>2</sub>: если  $x^1$  есть  $A_{21}$  и  $x^2$  есть  $A_{22}$ , и ...  $x^n$  есть  $A_{2n}$ , то  $u=a_2$ ;

.....

П<sub>1m</sub>: если  $x^1$  есть  $A_{m1}$  и  $x^2$  есть  $A_{m2}$ , и ...  $x^n$  есть  $A_{mn}$ , то  $u=a_m$ ,

где  $x^1, x^2, \dots, x^n$  – компоненты  $\vec{x}$ ;  $A_{ij}$  – нечеткие множества,

определенные на множестве действительных чисел  $\mathbb{R}$  и имеющие

функции принадлежности  $\mu_{ij}(x^j)$  соответственно ( $i=1, 2, \dots, m; j=1, 2, 3, \dots, n$ );  $a_1, a_2, \dots, a_m$  – действительные числа.

Приведенные продукционные правила удобно представить в виде:

$P_1$ : если  $\vec{x}$  есть  $A_1$ , то  $u=a_1$ ,

$P_2$ : если  $\vec{x}$  есть  $A_2$ , то  $u=a_2$ ,

.....

$P_m$ : если  $\vec{x}$  есть  $A_m$ , то  $u=a_m$ ,

где  $A_i$  – фиктивные нечеткие множества, определенные на  $n$ -мерном множестве действительных чисел  $R^n$  и имеющие функции принадлежности  $\mu_i(\vec{x}) = \min(\mu_{i1}(x^1), \mu_{i2}(x^2), \mu_{in}(x^n))$  соответственно.

Выходной сигнал НЛР  $u$  рассчитывается в соответствии с алгоритмом

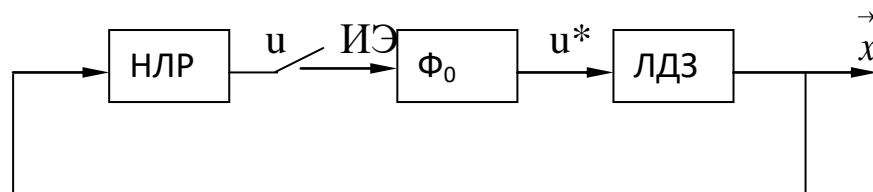


Рис.5.8. Система управления с нечетким логическим регулятором.

нечеткого вывода Сугэно (Sugeno) нулевого порядка [2]

$$u(\vec{x}) = \frac{\sum_{i=1}^m a_i \mu_i(\vec{x})}{\sum_{i=1}^m \mu_i(\vec{x})} \quad (5.20)$$

Требуется: для системы на рис. 1 определить достаточное условие устойчивости.

## 2. Характеристика блока нечеткого логического вывода.

Зависимость (5.20) достаточное сложна для непосредственного построения критерия устойчивости, поэтому для его получения воспользуемся методикой, изложенной ниже.

Предположим, что указанная зависимость. т.е.  $u = u(\vec{x})$ , обладает свойством  $u(0)=0$  (в большинстве случаев добиться выполнения указанного свойства удастся линейным преобразованием фазовых координат).

Представим, далее, формулу (5.20) в виде

$$\vec{u}(x) = \vec{k}^T(x) \vec{x} \quad (5.21)$$

где  $\vec{k}^T(x)$  - коэффициент передачи НЛР, зависящий от входного сигнала  $\vec{x}$

**Воспользовавшись свойствами векторной нормы [3], получим**

$$\left| \vec{u}(x) \right| = \left\| \vec{k}^T(x) \right\| \cdot \left\| \vec{x} \right\| \quad (5.22)$$

(здесь и ниже под векторной нормой понимается  $l_\infty$ -норма  $\left\| \vec{x} \right\| = \max\{|x^1|, |x^2|, \dots, |x^n|\}$ , где  $n$  - размерность вектора  $\vec{x}$ , а под матричной нормой - максимальная строчная норма

$$\|A\| = \max_{1 \leq i \leq n} \sum_{j=1}^n |a_{ij}| ,$$

где  $a_{ij}$  - элемент матрицы  $A$  находящийся в  $i$ -й строке и  $j$ -м столбце), откуда

$$\frac{\left| \vec{u}(x) \right|}{\left\| \vec{x} \right\|} \leq \left\| \vec{k}^T(x) \right\|, \quad (5.23)$$

что позволяет записать

$$\max_{\forall \vec{x} \in R^n} \frac{\left| \vec{u}(x) \right|}{\left\| \vec{x} \right\|} \leq \max_{\forall \vec{x} \in R^n} \left\| \vec{k}^T(x) \right\| \quad (5.24)$$

На основании выражения (5.24) в качестве характеристики нелинейной зависимости НЛР выберем параметр

$$K_H = \max_{\forall \vec{x} \in R^n} \left\| \vec{k}^T(x) \right\|, \quad (5.25)$$

при этом ввиду сложности определения численного значения  $K_H$  в случае произвольного НЛР воспользуемся его оценкой.

Для этого, вначале решая совместно (5.22), (5.23) и (5.25), придем к соотношению

$$K_H = \max_{\vec{x} \in R^n} \frac{\left| \sum_{i=1}^m a_i \mu_i(\vec{x}) \right|}{\left\| \vec{x} \right\| \sum_{i=1}^m \mu_i(\vec{x})}. \quad (5.26)$$

Используя свойства модуля [3], из предыдущего выражения получим

$$K_H \leq \max_{\vec{x} \in R^n} \frac{\left| \sum_{i=1}^m |a_i| \mu_i(\vec{x}) \right|}{\left\| \vec{x} \right\| \left\| \sum_{i=1}^m \mu_i(\vec{x}) \right\|}. \quad (5.27)$$

Отметим далее, что множитель

$$\frac{\left| \sum_{i=1}^m |a_i| \mu_i(\vec{x}) \right|}{\left\| \vec{x} \right\| \left\| \sum_{i=1}^m \mu_i(\vec{x}) \right\|} = X_c, \quad (5.28)$$

определяет, по сути, координаты центра масс  $X_c$  невесомого стержня с расположенными на нем грузами с массами  $\mu_i(\vec{x})$  в точках с координатами  $|a_i|$ , где  $i = 1, 2, \dots, m$  (один из возможных вариантов расположения грузов на стержне показан на рис. 2).

Очевидно, что координата точки центра масс  $X_c$  не может превышать координаты крайнего справа груза, имеющего массу, отличную от 0. Таким образом

$$X_c \leq \max[|a_1|1_0(\mu_1(\vec{x})), |a_2|1_0(\mu_2(\vec{x})), \dots, |a_m|1_0(\mu_m(\vec{x}))], \quad (5.29)$$

где

$$1_0(t) = \begin{cases} 1 & \text{при } t > 0, \\ 0 & \text{при } t = 0. \end{cases}$$

Из сопоставления соотношений (5.27)-(5.29) следует

$$K_H \leq \max_{\vec{x} \in R^n} \frac{1}{\left\| \vec{x} \right\|} \max[|a_1|1_0(\mu_1(\vec{x})), |a_2|1_0(\mu_2(\vec{x})), \dots, |a_m|1_0(\mu_m(\vec{x}))], \quad (5.30)$$

что дает возможность записать выражение для искомой оценки

$$K_H \leq \hat{K}_H = \max \left( \max_{\vec{x} \in B_1} \frac{|a_1|}{\left\| \vec{x} \right\|}, \max_{\vec{x} \in B_2} \frac{|a_2|}{\left\| \vec{x} \right\|}, \dots, \max_{\vec{x} \in B_m} \frac{|a_m|}{\left\| \vec{x} \right\|} \right), \quad (5.31)$$

где  $B_1, B_2, \dots, B_m$  – носители нечетких множеств  $A_1, A_2, \dots, A_m$  соответственно [2].

Методику использования соотношения (5.29) поясним на примере.



Пример 1. Рассмотрим НЛР, описываемый набором нечетких правил:

$\Pi_1$ : если  $x_1$  есть  $P$  и  $x^2$  – произвольно, то  $u=1$ ,

$\Pi_2$ : если  $x_1$  есть  $N$  и  $x^2$  – произвольно, то  $u=-1$ ,

$\Pi_3$ : если  $x^1$  – произвольно и  $x^2$  есть  $P$ , то  $u=1$ ,

$\Pi_4$ : если  $x^1$  – произвольно и  $x^2$  есть  $N$ , то  $u=-1$ ,

$\Pi_5$ : если  $x^1$  есть  $Z$  и  $x^2$  есть  $Z$ . то  $u=0$ ,

где  $x^1$  и  $x^2$  – 1-я и 2-я компоненты вектора  $\vec{x}$  соответственно.

На рис. 3 приведены функции принадлежности нечетких множеств  $Z$ ,  $P$  и  $N$ .

Для правила  $\Pi_1$  носитель нечеткого множества  $A_1 - x^1 \in [1, +\infty]$ ,  $x^2 \in [-\infty, +\infty]$ , значение  $a_1=1$ , следовательно

$$\max_{x^1 \in [1, +\infty], x^2 \in [-\infty, +\infty]} \frac{|1|}{\left\| \begin{matrix} \rightarrow \\ x \end{matrix} \right\|} = 1.$$

Для правила  $\Pi_2$  носитель нечеткого множества  $A_2 - x^1 \in [-\infty, -1]$ ,  $x^2 \in [-\infty, +\infty]$ , значение  $a_2=-1$ , следовательно

$$\max_{x^1 \in [-\infty, -1], x^2 \in [-\infty, +\infty]} \frac{|-1|}{\left\| \begin{matrix} \rightarrow \\ x \end{matrix} \right\|} = 1.$$

Для правила  $\Pi_3$  носитель нечеткого множества  $A_3 - x^2 \in [1, +\infty]$ ,  $x^1 \in [-\infty, +\infty]$ , значение  $a_3=1$ , следовательно

$$\max_{x^2 \in [1, +\infty], x^1 \in [-\infty, +\infty]} \frac{|1|}{\left\| \begin{matrix} \rightarrow \\ x \end{matrix} \right\|} = 1.$$

Для правила  $\Pi_4$  носитель нечеткого множества  $A_4 - x^2 \in [-\infty, -1]$ ,  $x^1 \in [-\infty, +\infty]$ , значение  $a_4=-1$ , следовательно

$$\max_{x^2 \in [-\infty, -1], x^1 \in [-\infty, +\infty]} \frac{|-1|}{\left\| \begin{matrix} \rightarrow \\ x \end{matrix} \right\|} = 1.$$

Для правила  $\Pi_5$  носитель нечеткого множества  $A_5 - x^1 \in [-2, -2]$ ,  $x^2 \in [-2, +2]$ , значение  $a_5=0$ , следовательно

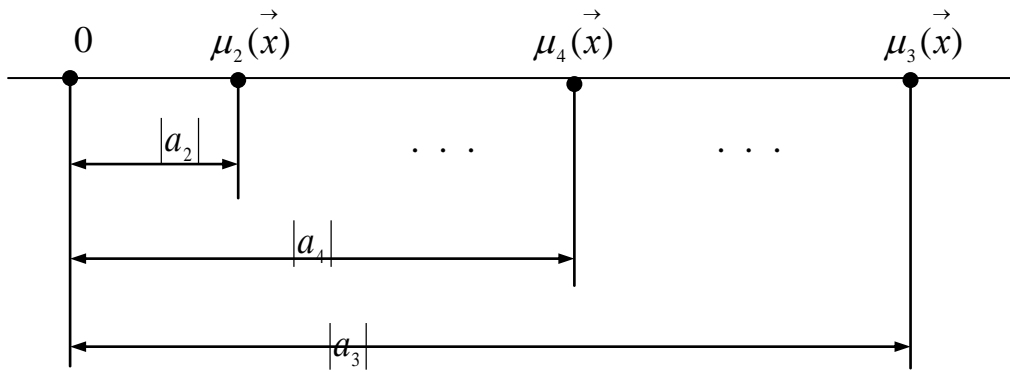


Рис 5.9. Иллюстрация в выводу формулы (5.31).

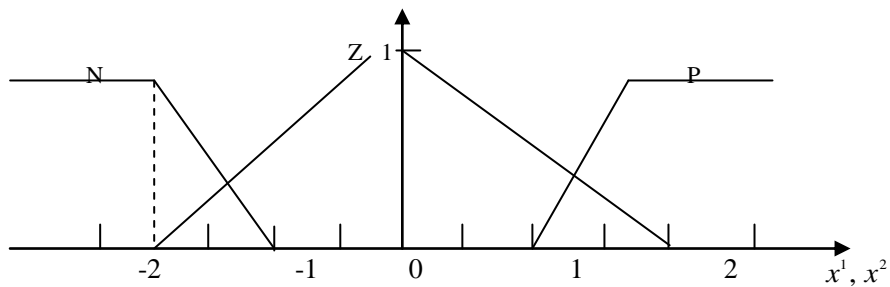


Рис. 5.10. Функция принадлежности нечетких множеств.

$$\max_{x^1 \in [-2; -2], x^2 \in [-2, +2]} \frac{\|0\|}{\left\| \begin{matrix} \rightarrow \\ x \end{matrix} \right\|} = 0.$$

Поставляя данные частные результаты в выражение (5.31), получим  $\hat{K}_H = \max(1, 1, 1, 1, 1, 0) = 1$ . Укажем, что действительное значение  $K_H = 0.642$ .

**3. Достаточное условие устойчивости.** Используя описанную методику определения численного значения величины  $\hat{K}_H$ , для произвольного НЛР можно получить критерий асимптотической устойчивости системы на рис. 5.8. Допустим, что ЛДЗ (рис. 5.8) описывается векторно – матричным дифференциальным уравнением.

$$\frac{d\vec{x}}{dt} = A\vec{x} + \vec{b}u, \quad (5.32)$$

где  $A$  и  $\vec{b}$  - постоянные матрица и вектор соответственно.

Последовательное соединение импульсного элемента с фиксатором нулевого порядка и линейного динамического звена описывается разностным уравнением [4]

$$\vec{x}_{k+1} = A_0 \vec{x}_k + \vec{b}_0 u(\vec{x}_k), \quad (5.33)$$

где  $A_0 = e^{AT_0}$ ,  $\vec{b}_0 = A^{-1}(e^{AT_0} - I)\vec{b}$ ,  $T_0$  - период работы импульсного элемента,  $I$  - единичная матрица,  $u(\vec{x}_k)$  - нелинейное преобразование, осуществляемое НЛР,  $k=0,1,2,\dots$

Для определения критерия устойчивости рассматриваемой системы воспользуемся вторым методом А. М. Ляпунова. В качестве функции А. М. Ляпунова выберем простейший ее вид

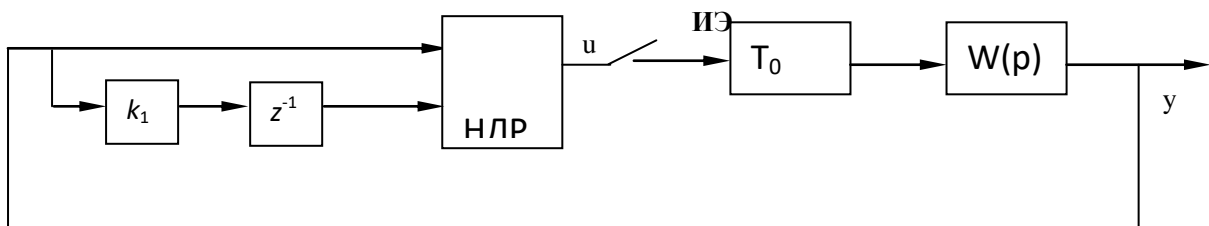


Рис. 5.11. Пример системы с НЛР

норму от переменных состояния системы (метод сжимающих отображений) [4]. Отметим, что применение в качестве функции А. М. Ляпунова квадратичной формы вида  $\vec{x}^T A \vec{x}$  (где  $A$  - положительно определенная матрица) от переменных состояния системы, по всей видимости, позволит получить более широкие области устойчивости, однако при этом математические выкладки будут значительно сложнее [4].

Решая совместно (5.21) и (5.33), получим

$$\vec{x}_{k+1} = A_0 \vec{x}_k + \vec{b}_0 \vec{k}^T(\vec{x}_k) \vec{x}_k. \quad (5.34)$$

Согласно принципу сжимающих отображений, достаточным условием асимптотической устойчивости положения равновесия системы (5.33) будет выполнение неравенства [4].

$$\|A_0 + \vec{b}_0 \vec{k}^T(\vec{x}_k)\| < 1 \quad (5.35)$$

при произвольном  $\vec{x}_k$ .

С учетом выражений (5.25) и (5.31) неравенство (5.35) можно решить относительно параметров системы.

Пример 2. Рассмотрим автономную систему, приведенную на рис. 5.11 ( $k_1$  – пропорциональное звено,  $z^{-1}$ -звено задержки на один такт).

Объект управления описывается передаточной функцией

$$W(p) = -\frac{k_0}{1 + pT}.$$

Импульсный элемент с фиксатором нулевого порядка имеет период квантования  $T_0$ . Используется НЛР из примера 1.

Для системы на рис. 14.4 справедливо разностное уравнение

$$\begin{bmatrix} x_{k+1}^1 \\ x_{k+1}^2 \end{bmatrix} = \begin{bmatrix} e^{\frac{T_0}{T}} & 0 \\ -k_1 & 0 \end{bmatrix} \begin{bmatrix} x_k^1 \\ x_k^2 \end{bmatrix} + \begin{bmatrix} \left(1 - e^{\frac{T_0}{T}}\right)k_0 \\ 0 \end{bmatrix} u(-x_k^1, x_k^2),$$

$$u(-x_k^1, x_k^2) = \vec{k}^T(-x_k^1, x_k^2) \begin{bmatrix} -x_k^1 \\ x_k^2 \end{bmatrix},$$

где  $x_k^1 = -y_k, x_k^2 = k_1 y_{k-1}$ .

Условие устойчивости (5.25) в рассматриваемом случае имеет вид.

$$\left\| \begin{bmatrix} e^{\frac{T_0}{T}} & 0 \\ -k_1 & 0 \end{bmatrix} + \begin{bmatrix} \left(1 - e^{\frac{T_0}{T}}\right)k_0 \\ 0 \end{bmatrix} \vec{k}^T(-x_k^1, x_k^2) \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} \right\| < 1.$$

На основании свойств матричной нормы [3.4] в выражений (5.25), (5.31) можно показать, что достаточным условием выполнения данного неравенства будет

$$\left| e^{\frac{T_0}{T}} - 2\hat{K}_H k_0 \left( e^{\frac{T_0}{T}} - 1 \right) \right| < 1,$$

откуда достаточное условие асимптотической устойчивости положения равновесия рассматриваемой системы принимает вид

$$k_1 < 1$$

$$k_0 < \frac{1 + e^{\frac{T_0}{T}}}{2 \left( 1 - e^{\frac{T_0}{T}} \right) \hat{K}_H}.$$

Приняв  $\hat{K}_H = 1$  (из пример 1),  $\kappa_1 = 0.1$  и  $T_0 = 0.1$ , определим область рассматриваемой системы, изображенную на рис.5.12 (ниже линии 1). Для сравнения на данном рисунке показана также действительная область устойчивости системы, полученная путем имитационного моделирования с использованием системы МАТЛАБ (область ниже линии 2).

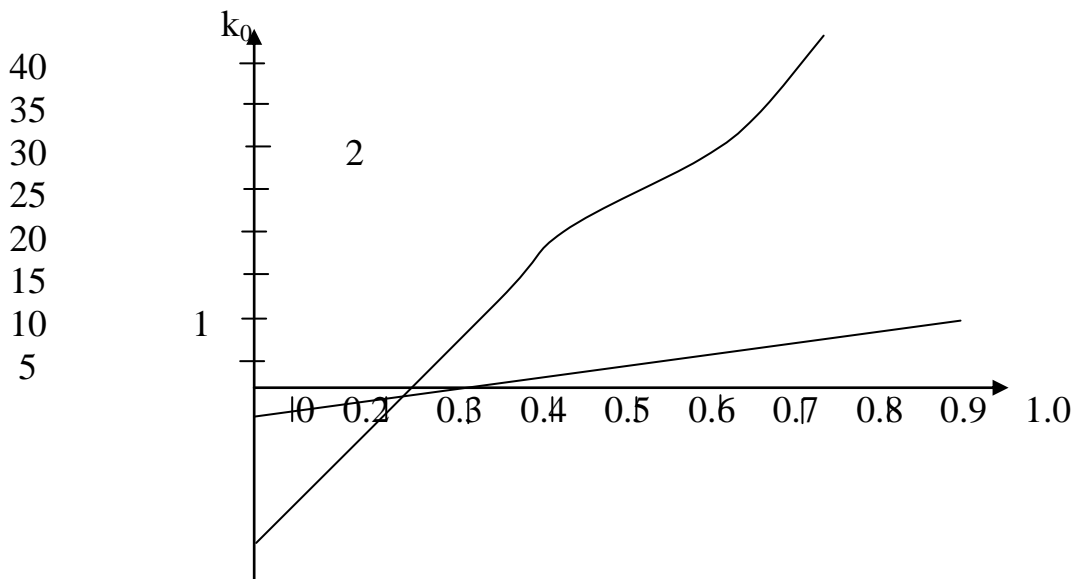


Рис.5.12

## Глава 6. НЕЙРОННЫЕ РЕГУЛЯТОРЫ. НЕЧЕТКИЕ НЕЙРОННЫЕ РЕГУЛЯТОРЫ

### 6.1. Нейронная идентификация и управление

**Нейронная идентификация.** Нейронные сети как универсальные аппроксиматоры функций, широко используются для идентификации сложных нелинейных неизвестных объектов и систем [21-25]. На рис.6.1 представлена схема нейронной идентификации прямой модели объекта.

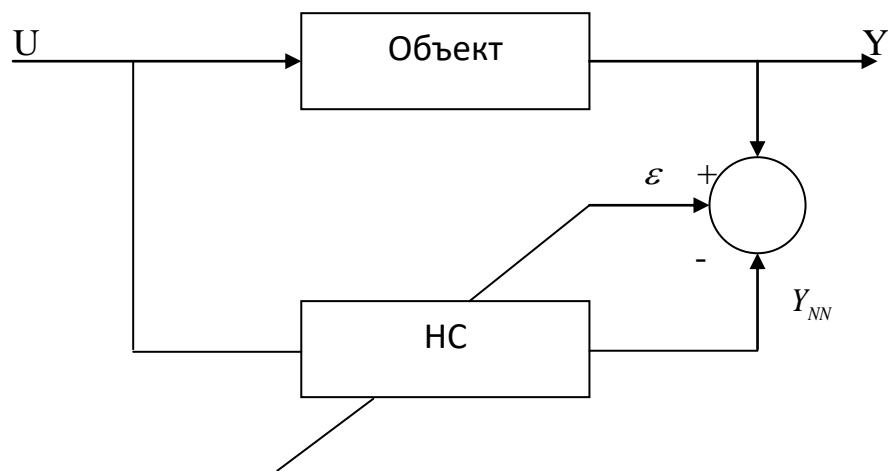


Рис.6.1. Схема идентификации прямой модели объекта

В этом случае определяются такие значения весов нейронной сети, представляющей собой искомую модель, которые обеспечивают минимальное значение среднеквадратической ошибки между выходом модели (нейронного идентификатора)  $Y_{NN}$  и текущим выходом неизвестного объекта. При таком типе идентификатора (т.е. идентификатора прямой модели объекта) сравнивается выход объекта с выходом нейронного идентификатора.

При определении инверсной модели объекта вход последнего ( $u$ ) сравнивается с выходом нейронного идентификатора ( $Y_{NN}$ ). Задача идентификации заключается в определении таких весовых коэффициентов нейронной сети, в которой реализуется искомая модель объекта, которые обеспечивают минимальное значение среднеквадратической ошибки  $\varepsilon_{ck}^2$  (рис.6.2).

Различают [25-26] еще два типа нейронного идентификатора: адаптивной и обучающийся. В первом случае выход нейронной сети сходится с сигналом обучения в пределах одного испытания.

Второй тип обучается в каждый период испытания. Выход нейронной сети сходится с сигналом обучения после нескольких испытаний.

Ради простоты, сначала, рассмотрим класс объектов управления, которые описываются следующей передаточной функцией

$$A(Z^{-1})Y(k) = Z^{-d}G_0B(Z^{-1})U(k),$$

$$A(Z^{-1}) = 1 + \sum_{i=1}^n a_i Z^{-i}, \quad (6.1)$$

$$B(Z^{-1}) = 1 + \sum_{i=1}^m b_i Z^{-i},$$

где  $a_i, b_i$  - неизвестные параметры объекта,  $d$  - время чистого запаздывания. Предполагается, что порядки  $n$  и  $m$ , и время чистого запаздывания  $d$  известны.

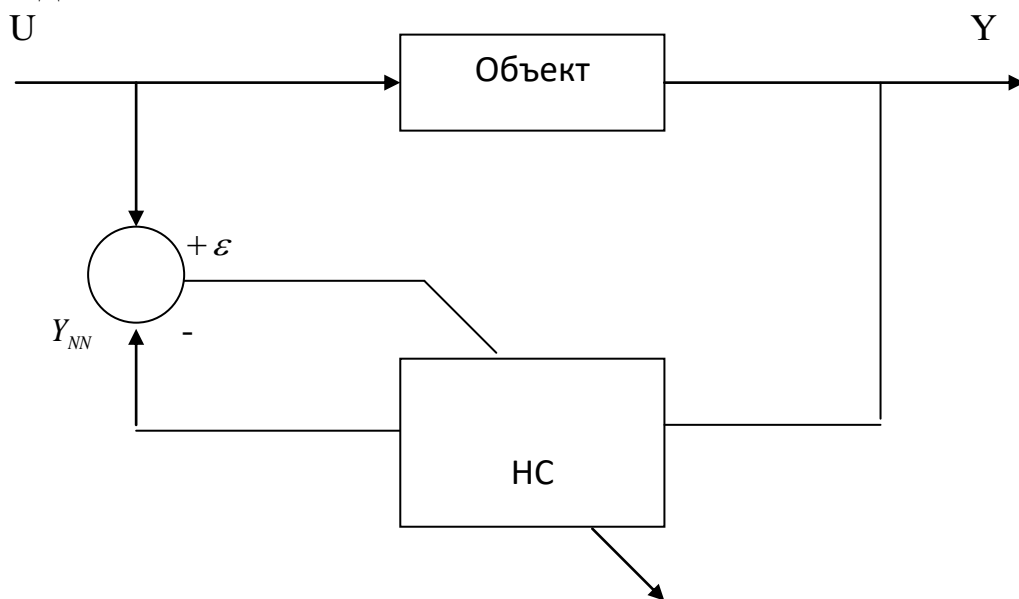


Рис.6.2. Схема идентификации обратной модели объекта

Выход объекта (6.1) может выражаться как

$$Y(k) = -\sum_{i=1}^n a_i Y(k-i) + G_0 \left\{ U(k-d) + \sum_{i=1}^m b_i u(k-i-d) \right\}. \quad (6.2)$$

Используя методы обучения, можно выбирать весовые матрицы нейронной модели таким образом, чтобы

$\varepsilon_{ck}^2 = [Y(k) - Y_{NN}(k)]^2$  было минимальным. Видно, что выход объекта  $Y(k)$  при идентификации прямой модели становится сигналом обучения. Используя уравнение (6.2) вход нейронного идентификатора  $I(k)$  и неизвестный вектор параметров  $\alpha$  могут быть определены следующими уравнениями [25]

$$I^T(k) = [U(k-d), Y(k-1), \dots, Y(k-n), \\ U(k-d-1), \dots, U(k-m-d)], \quad (6.3)$$

$$\alpha^T = [G_0, a_1, \dots, a_n, G_0 b_1, \dots, G_0 b_m]. \quad (6.4)$$

В свою очередь выход нейронной сети (если она линейная) может выражаться как

$$Y_{NN}(k) = w^T(k)W(k)I(k), \quad (6.5)$$

где  $w^T(k), W(k)$  - вектор и матрица весов.

Если сигнал обучения представить как

$$Y(k) = \alpha^T I(k),$$

то можно легко получить

$$\varepsilon(k) = [\alpha^T - w^T(k)W(k)]I(k). \quad (6.6)$$

Исходя из (6.6) можно констатировать, что условием  $\varepsilon = 0$  является  $\alpha^T = w^T(k)W(k)$ , что показывает на прямое соответствие между весовыми матрицами  $w^T(k), W(k)$  и искомыми параметрами  $\alpha^T$ . На рис.6.5, представлена структура идентификатора прямой передаточной функции объекта для случая  $n=2, m=1, d=1$ . Как видно из рис. 6.5 входами нейронной сети (идентификатора) являются входные и выходные сигналы объекта. Для того, чтобы на входе нейронной сети во время обучения всегда был ограниченный сигнал, объект охвачен отрицательной обратной связью. Если выход объекта всегда ограничен, то в такой обратной связи необходимости нет.

Как выше было указано, при получении инверсной передаточной функции неизвестного объекта входной сигнал последнего используется как сигнал обучения, который определяется как

$$U(k-d) = (1/G_0)Y(k) + \sum_{i=1}^n a_i Y(k-i) + \\ + G_0 \left\{ U(k-d) - G_0 \sum_{i=1}^m b_i U(k-i-d) \right\}.$$



Входной вектор нейронной сети и вектор неизвестных параметров  $\alpha$  определяются аналогично, как это сделано для идентификатора прямой модели

$$I^T(k) = [Y(k), Y(k-1), \dots, Y(k-n), U(k-1-d), \dots, U(k-m-d)], \quad (6.7)$$

$$\alpha^T = 1/G_0 [1, a_1, \dots, a_n, G_0 b_1, \dots, G_0 b_m]. \quad (6.8)$$

Из условия

$$\varepsilon(k) = U(k-d) - U_{NN}(k).$$

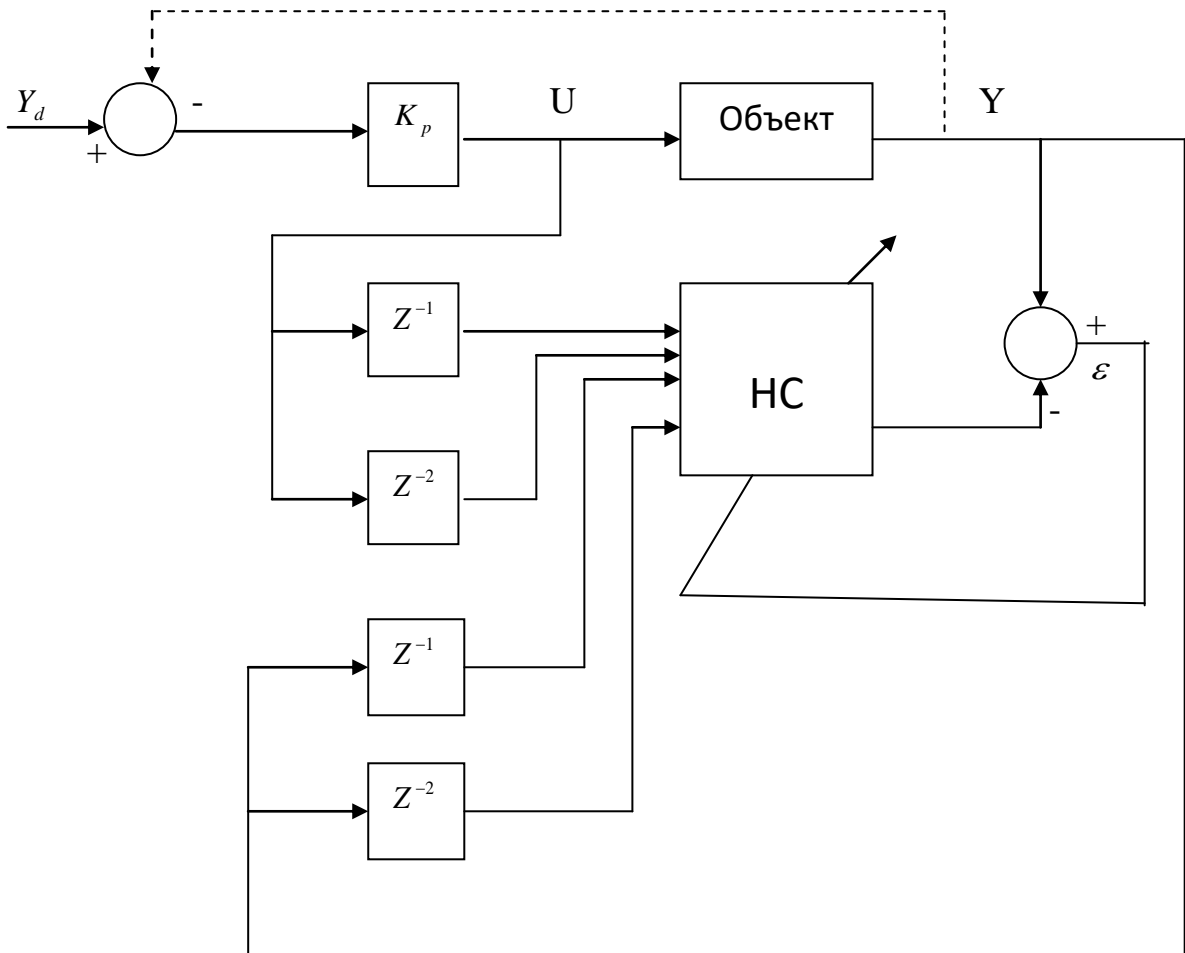


Рис.6.5 Блок-схема идентификатора прямой модели

Можно получить, что при  $\alpha^T = w^T(k)W(k)$ ,  $\varepsilon$  приравнивается нулю. Это условие, с использованием соответствующего алгоритма обучения, позволяет найти неизвестные параметры объекта.

Структура инверсного нейронного идентификатора для объекта с  $n=2$ ,  $m=1$ ,  $d=1$  представлена на рис. 6.6.

До сих пор мы рассматривали идентификацию линейных объектов линейными нейронными сетями. Выбором соответствующих нелинейных функций активации в моделях нейронов, в частности, сигмоидной, и добавлением их к рассмотренным нейронным сетям можно идентифицировать любой нелинейный объект.

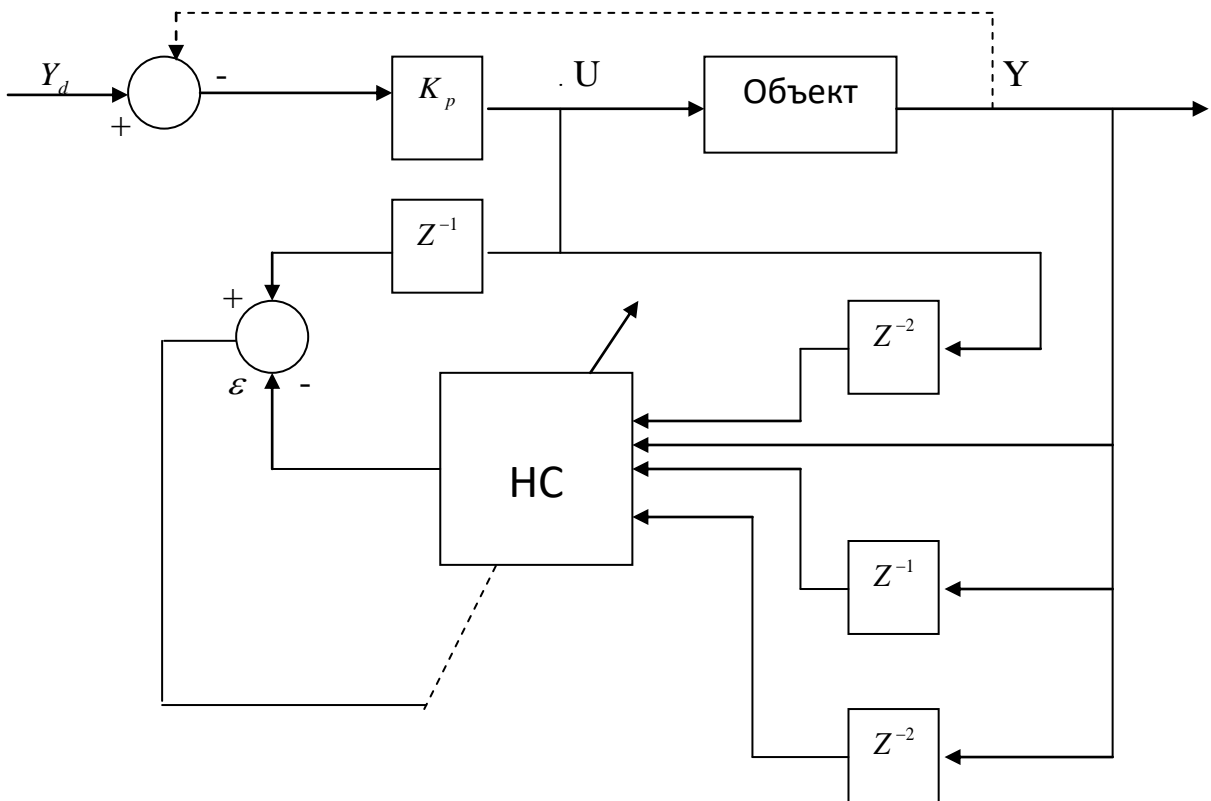


Рис.6.6 Блок-схема идентификатора обратной модели

## 6.2. Структура нейронного регулятора

**Нейронное управление.** Нейронные сети широко используются в качестве контроллеров в различных системах управления [27-33]. Это связано в первую очередь с тем, что гибкость структуры нейронных сетей позволяет реализовать широкий спектр нелинейных законов регулирования. При этом повышается робастность контроллеров. Во вторых, способность к обучению позволяет нейронным сетям реализовать различные схемы систем управления.

На рис.6.6 представлена структура нейронной системы управления с прямым контроллером. В этой системе нейронный контроллер реализует инверсную модель объекта. Такая структура нейронной системы управления не всегда реализуема в силу того, что не всегда возможно получить физически реализуемую обратную модель объекта.

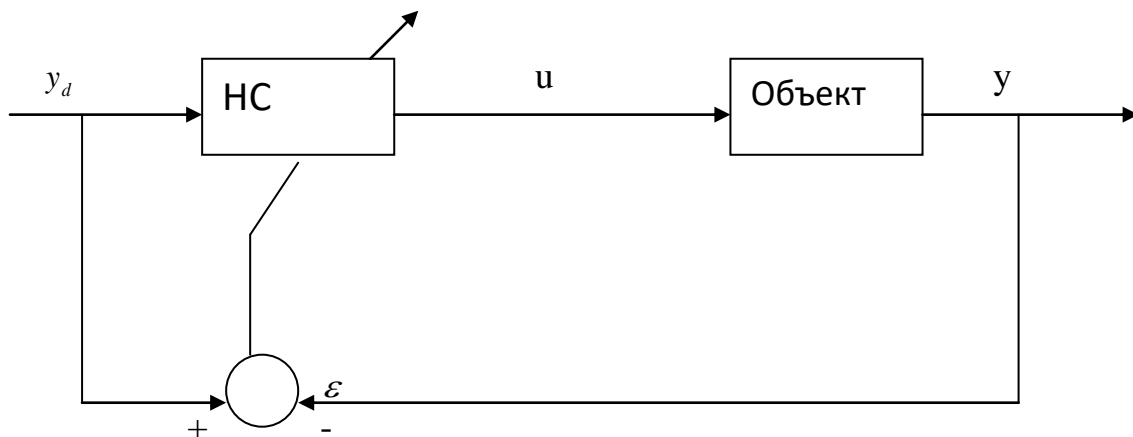


Рис.6.6. Структура нейронной системы управления с прямым контроллером

На рис. 6.7 представлена структура нейронной системы с идентификатором. Такой класс систем целесообразен для управления объектами с неизвестными характеристиками. Идентифицируемая нейронная модель объекта используется для настройки нейронного контроллера системы. Такая структура нейронного идентификатора была подробно рассмотрена выше. Ниже рассмотрим развернутую структуру нейронных контроллеров. На рис. 6.8 представлена структура линейного нейронного контроллера реализованного на базе трехслойной нейронной сети.

На рис.6.9 показана трехслойная нейронная сеть, реализующая контроллер в нелинейных системах регулирования.

На рис. 6.10 представлена нейронная система управления процессом.

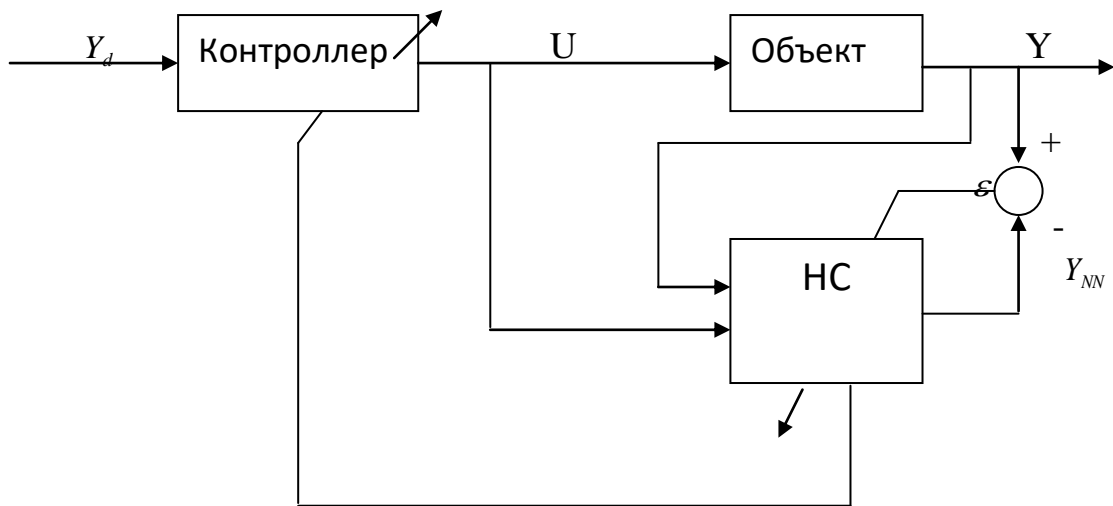


Рис.6.7. Структура системы с идентификатором

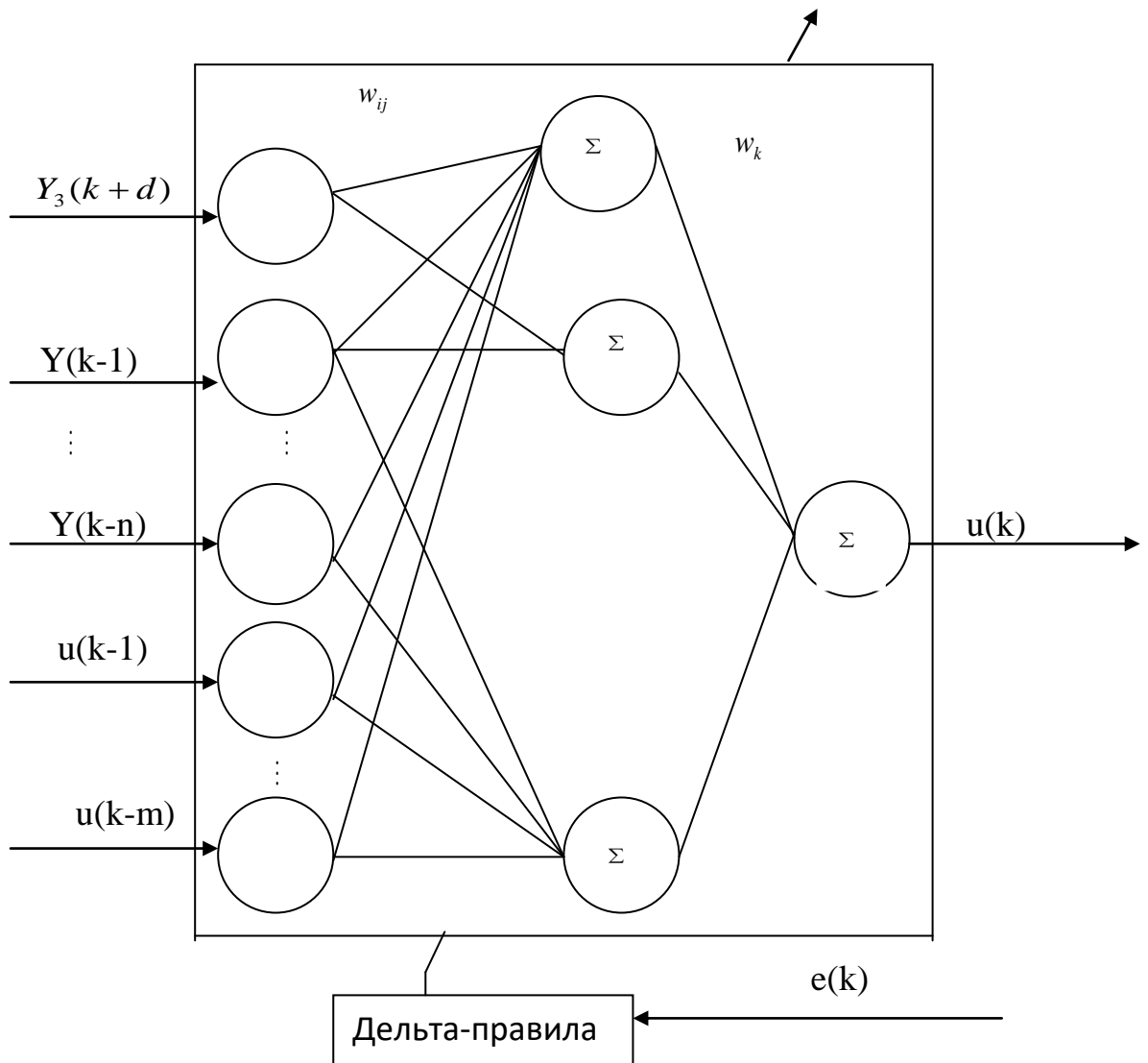


Рис.6.8. Линейная трехслойная нейронная сеть

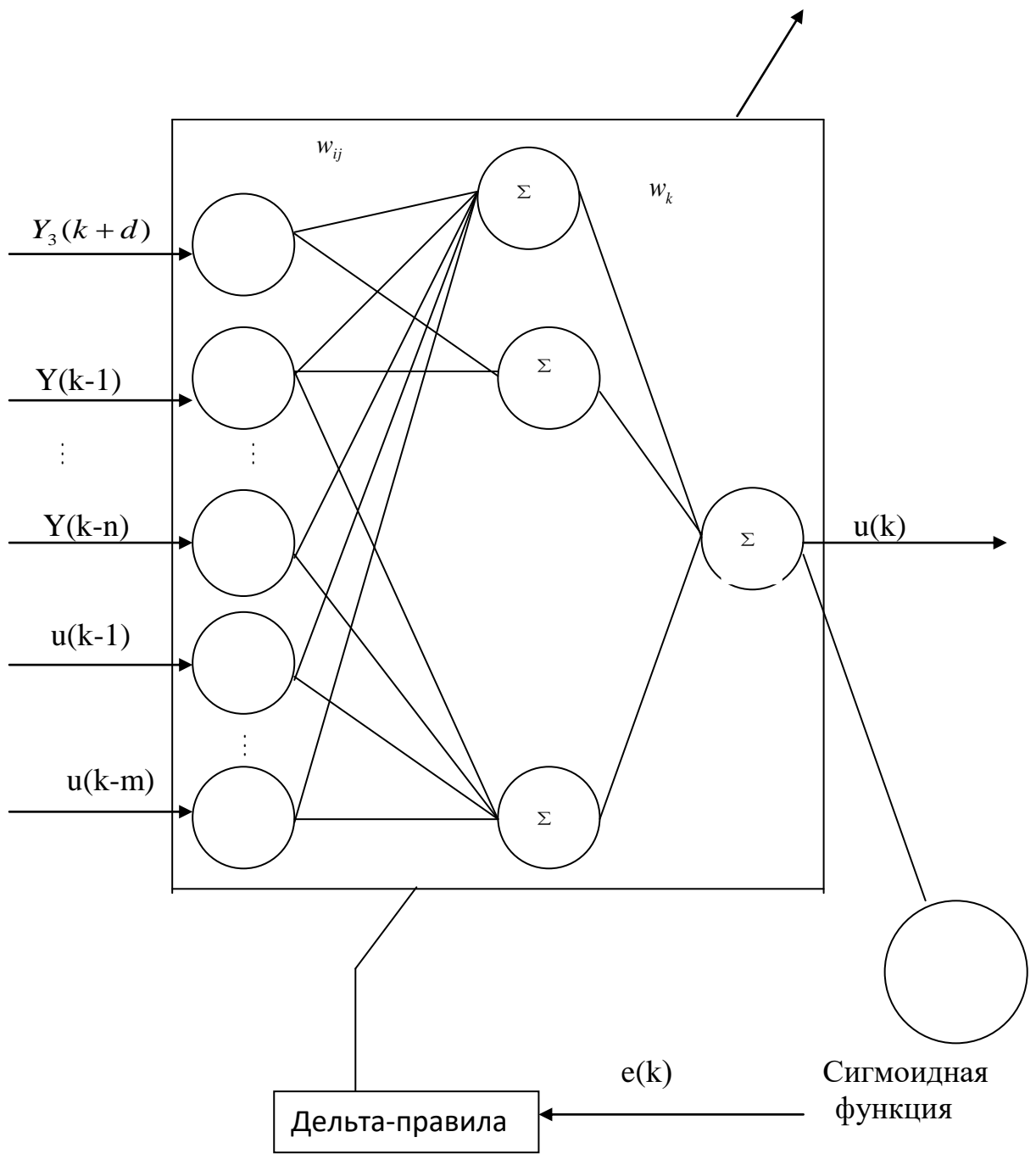


Рис.6.9. Нелинейная трехслойная нейронная сеть

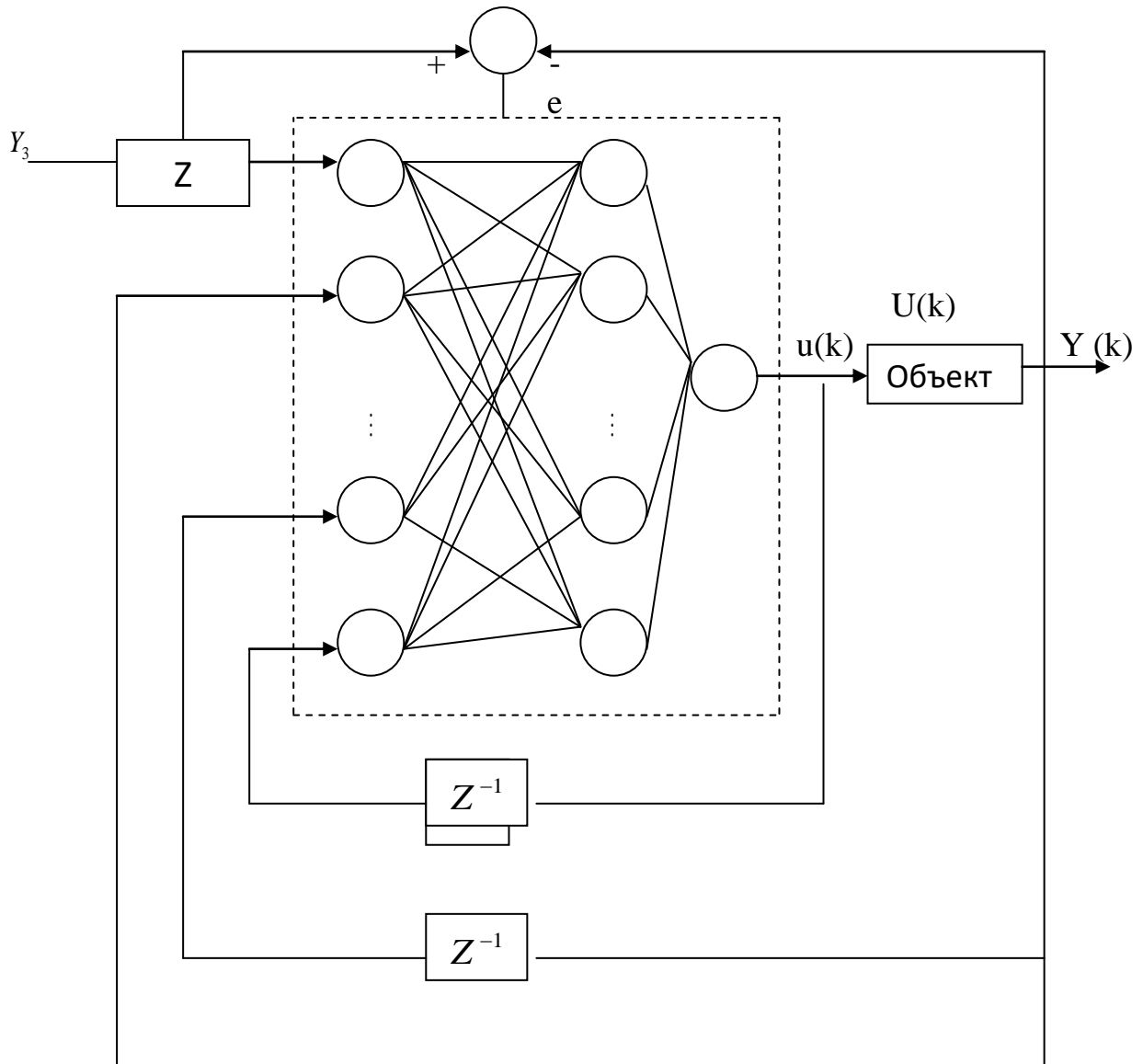


Рис. 6.10. Структура динамической нейронной системы управления

Ниже рассматривается проблема устойчивости нейронных систем управления [32]. Исследуется устойчивость нейронной системы с прямым контроллером. Нейронная сеть обучается с учетом динамики объекта. Квадратичная ошибка задается как

$$E = e^t e / 2, \quad (6.9)$$

где  $e = Y - Y_d$ .

Далее используется метод backpropagation, основанный на (6.9)

$$\Delta W = \nabla_w E = -e^t (\partial Y / \partial U, \partial U / \partial W). \quad (6.10)$$

Матрица  $\partial Y / \partial U$  учитывает динамику объекта, необходимую при исследовании устойчивости системы.

Согласно структуры системы, выход  $U$  нейронного контроллера определяется как

$$U = N(W, I).$$

Если все нейроны имеют одну и ту же сигмоидную функцию  $f$ , то получится следующее выражение

$$N(W, I) = f(W_1 f(W_2 I)),$$

где  $w_1$  - весовая матрица между входным и средним слоями,  $w_2$  - весовая матрица между средним и выходными слоями,

$$f(x) = 1 / (1 + e^{-x}).$$

Соотношение между весовыми матрицами  $w_1$  и  $w_2$  и вектором весов  $W$  - даются с помощью

$$W_1 = \begin{bmatrix} W_1^t 1 \\ \vdots \\ W_1^t j \\ \vdots \\ W_1^t n \end{bmatrix}; \quad W_2 = \begin{bmatrix} W_2^t 1 \\ \vdots \\ W_2^t j \\ \vdots \\ W_2^t n \end{bmatrix}; \quad W^t = (\dots, W_1^t j, \dots, W_2^t, i).$$

Применим метод Ляпунова. Используя квадратичную форму ошибок объекта и параметров, функция Ляпунова принимается как [34]

$$V(e, w) = e^t P e / 2 + W^t U^{-1} w / 2.$$

Ошибки из равновесной точки определяются как [32].

$e(t) = Y(t) - Y_d(t)$  (ошибка выхода);  $w = W - W_0$  (ошибка параметров);

$$u(t) = U(t) - U_0(t) \text{ (ошибка управляющего входа),}$$

где  $Y_d$  - желаемое значение выхода,  $W_0$  - весовой вектор, в котором сходится искомый,  $U_0(t)$  - управляющий вход, произведенный нейронной сетью в равновесной точке, которая определяется как  $U_0 = N(W_0, I_0)$ ,  $I_0$  - вход нейронной сети в равновесной точке.

При предположении, что объект является положительной действительной системой, имеющей единичный вход, получается следующее отношение

$$\dot{e}(t) = F e(t) + g u(t), \quad (6.11)$$

$$\varepsilon(t) = c^t e(t), \quad (6.12)$$

$$P F + F^t P = -Q, \quad (6.13)$$

$$Pg = c, \quad (6.14)$$

где  $\dot{e}(t) = d(e(t))/dt$ ,  $F$  есть  $R^{n \times m}$  - постоянная матрица,  $c$  и  $g$  являются  $R^n$  постоянными векторами. Если объект положительная действительная система, то существует положительно определенная матрица  $Q$ .

При условии  $(dV/dt) < 0$  гарантируется как устойчивость динамической системы, так и сходимость НС, поскольку в качестве функции Ляпунова выступает положительно-определенная функция. Это условие исследуется с использованием (6.11)-(6.14)

$$\begin{aligned} dV/dt = e^t (PF + F^t P)e/2 + (e^t Pg + g^t Pe)u/2 + \\ + d(w^t U^{-1} w/2)/dt = -e^t Qe/2 + \varepsilon u + \dot{w}^t U^{-1} w. \end{aligned} \quad (6.15)$$

Т.к. первый член в (6.15) отрицательный, то условие  $dV/dt < 0$  при условии удовлетворения отношения

$$\varepsilon u + \dot{w}^t U^{-1} w \leq 0, \quad (6.16)$$

где  $\dot{w}^t = dw/dt = \Delta w$ , что представляет значение корреляции для обучения. При рассмотрении условия равенства (6.16) получается следующее правило обучения

$$\dot{w}^t = -\varepsilon u + w^{-t} U (w = -U^{-t} \varepsilon u w^{-1}), \quad (6.17)$$

где  $w^{-t}$  определяется как вектор, удовлетворяющий условию

$$w^{-t} w = w^t w^{-t} = 1.$$

Проверка устойчивости системы с прямым контроллером, обучаемым backpropagation методом, сводится к следующему. Выбирается квадратичная форма как для модели, так и для ошибки параметров в качестве функции Ляпунова. Однако, в этом случае, т.к. метод backpropagation выводится из квадратичной ошибки  $E$ , определенный посредством (6.9), выбирается следующая расширенная квадратичная ошибка  $v(e)$  в качестве функции Ляпунова. Уравнение (6.9) задается как особый случай из (6.18) при  $P=1$  (единичная матрица). Функция Ляпунова не принимает во внимание сходимость вектора ошибки

$$v(e) = E = e^t P e / 2. \quad (6.18)$$

При использовании (6.11)-(6.14) мы предполагаем, что объект действительная положительная система и имеет единичный вход. Тогда  $dV/dt$  получается



$$\begin{aligned} dV/dt &= e^t(PF + F^tP)e/2 + (e^tPg + g^tPe)u/2 = \\ &= -e^tQe/2 + \varepsilon u = -e^tQe/2 + \varepsilon N(W, I). \end{aligned} \quad (6.19)$$

Метод backpropagation задается как

$$\begin{aligned} \Delta W &= -\eta e^t P(\partial e / \partial u)(\partial u / \partial W) = \\ &= -\eta e^t Pg(\partial N / \partial W) = -\eta \varepsilon(\partial N / \partial W) \end{aligned} \quad (6.20)$$

где  $\eta$  - положительный параметр для настройки коэффициента обучения. Поскольку метод backpropagation, определенный по (6.20), не может быть замещен в (6.19), мы учитываем малое отклонение от точки равновесия, что связано с необходимостью анализа локальной устойчивости. Управляющий вход  $u$  задается как

$$\Delta u = (\partial N / \partial W)^t \Delta W (\partial N / \partial I) \Delta I. \quad (6.21)$$

В предположении условия (6.21), (6.19) заменяются на

$$\begin{aligned} dV/dt &= -e^tQe/2 + \varepsilon \{ (\partial N / \partial W)^t \Delta W + (\partial N / \partial I)^t \Delta I \} = \\ &= -e^tQe/2 - \eta \varepsilon^2 (\partial N / \partial W)^t (\partial N / \partial W) + (\partial N / \partial I)^t \Delta I, \end{aligned} \quad (6.22)$$

где информация обратной связи определяется как

$$\Delta I^t = (e, u)^t, \quad u = (g/|g|)u.$$

Уравнение (6.22) переписывается в виде

$$\begin{aligned} dV/dt &= -e^tQe/2 - \eta \varepsilon^2 (\partial N / \partial W)^t (\partial N / \partial W) + \\ &+ \varepsilon (\partial N / \partial e)^t e + \varepsilon (\partial N / \partial u)^t u = -e^t(Q/2 + R - S)e + e^tTu, \end{aligned} \quad (6.23)$$

где

$$\begin{aligned} R &= \eta Pg(\partial N / \partial W)^t (\partial N / \partial W)(Pg)^t, \\ S &= Pg(\partial N / \partial e)^t, \\ T &= Pg(\partial N / \partial u)^t. \end{aligned} \quad (6.24)$$

Т.к. устойчивость гарантируется когда  $dV/dt \leq 0$ , мы исследуем знак каждого члена в (6.23). Однако, знак второго члена ( $e^tTu$ ) неопределен. Можно оценить знак  $\int (dV/dt) \delta t$  если рассмотреть определенный временной интервал. Предполагается, что объект – положительная действительная система приводится к пассивной системы [35], если

$$\int_{-\infty}^T (e^t u) dt \geq 0. \quad (6.25)$$

Знак второго члена становится отрицательным когда  $T < 0$ . Т.к.  $R > 0$  в общем удовлетворяется, знак  $\int (dV/dt) \delta t$  становится

отрицательным, когда  $S, T < 0$ . В результате обеспечивается устойчивость системы. Отрицательная определенность  $S$  и  $T$  зависит от значения  $(\partial N / \partial I)$ , вычисляемого как

$$\partial N / \partial I = f'(net^0)W_1 f'(net^1)W_2, \quad (6.26)$$

где

$$f'(net^0) = d(f(W_1 f(W_2 I))) / d(W_1 f(W_2 I)),$$

$$f'(net^1) = d(f(W_2 f)) / d(W_2 I).$$

Отрицательная определенность  $S$  и  $T$  зависит от весового вектора нейронной сети, который жестко связан с начальным значением и траекторией сходимости весового вектора.

Эти результаты показывают, что система становится устойчивой, когда достигается отрицательная определенность  $S$  и  $T$ . Однако, даже если условия отрицательной определенности не удовлетворяются, система может быть устойчива, когда нормы положительно определенных матриц  $Q$  и  $R$  принимают большие значения. Положительная определенность  $Q$  зависит от положительно действительного объекта. Норма положительно определенной матрицы  $R$  жестко зависит как от параметра настройки  $\eta$ , так и от нормы весовой матрицы, как показано в (6.23). Поэтому, когда возможно выбрать большое значение для  $\eta$  и большую  $(\partial N / \partial W)$  норму, система имеет хороший шанс оказаться устойчивой. Однако, при использовании дискретного типа метода backpropagation, существует верхнее ограничение для  $\|\Delta W\|$ , что ведет к неустойчивости, когда  $\Delta W$  становится большим.

Если можно отслеживать траекторию сходимости весового вектора  $W$ , то мы могли бы количественно оценивать условие устойчивости. Приведенный анализ оценивает характеристики устойчивости лишь качественно и показывает, что устойчивость с использованием метода backpropagation зависит как от начального весового вектора, так и от параметра  $\eta$ . То есть, метод backpropagation не может гарантировать глобальную асимптотическую устойчивость сам по себе. Необходимо найти

условие устойчивости методом проб и ошибок, изменяя параметр настройки  $\eta$  и начальное значение весового вектора.

Рассмотрим задачу синтеза нейронного контроллера для автоматического регулирования температуры ректификационной колонны. Структура одномерной системы автоматического управления технологическом процессом представлена на рис. 13.39. Регулируемая координата  $x$  в сравнивающем устройстве 1 сравнивается с заданием  $g$  и сигнал рассогласования (ошибки)  $e=g-x$  поступает параллельно в дифференциатор 2 и интегратор 3.

Выходные сигналы дифференциатора ( $\dot{e}$ ), интегратора ( $\int e$ ) и сигнал ошибки  $e$  умножаются на масштабные коэффициенты  $k_{\dot{e}}, k_{\int e}, k_e$  в элементах 4-6, а затем поступают в базу знаний 7, где формируется управляющий сигнал  $u$ , который после масштабирования элементом 8 поступает на исполнительный механизм объекта 9. Масштабные элементы 4-6 и 8 позволяют привязать регулятор к конкретным технологическим процессам.

Задача синтеза нейронной САУ, заключается в определении таких значений коэффициентов  $k_{\dot{e}}, k_{\int e}, k_e, k_u$ , структуры и параметров НС при которых в системе с объектом 9 (рис.13.39) получилось бы переходная характеристика, удовлетворительно совпадающая с желаемой. БЗ регулятора реализована на основе нейронной сети, в связи с чем конструирование желаемой БЗ эквивалентно определению рациональной архитектуры и матрицы весовых коэффициентов синтезируемой нейронной сети.

Архитектура разрабатываемой нейронной сети представляет собой трехслойную «feedforward» структуру. Первый слой включает входные нейроны (узлы), средний состоит из скрытых, а последний – из выходных узлов.

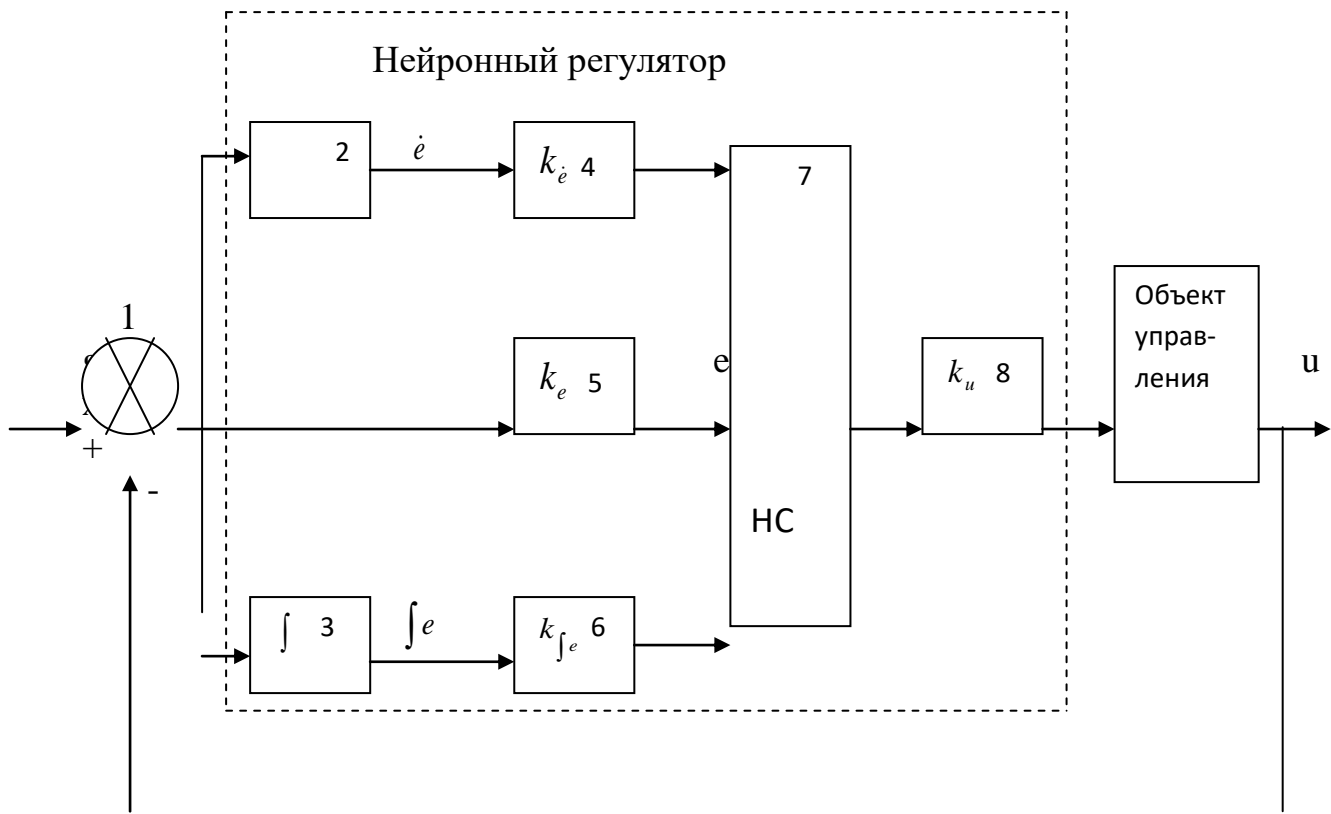


Рис.6.15. Структура нейронного регулятора

Суть задачи синтеза САУ на базе нейронной сети заключается в следующем. Представим, что имеется совокупность продукционных правил, отражающая желаемое (эталонное) поведение конструируемой системы управления

$$\left. \begin{array}{l}
 \text{IF } g = g_1 \text{ and } x = x_1 \text{ THEN } \dot{x} = \dot{x}_1 \\
 \text{IF } g = g_2 \text{ and } x = x_2 \text{ THEN } \dot{x} = \dot{x}_2 \\
 \dots\dots\dots \\
 \text{IF } g = g_n \text{ and } x = x_n \text{ THEN } \dot{x} = \dot{x}_n
 \end{array} \right\} \quad (6.27)$$

Представление этих знаний в БЗ проектируемого САУ заключается в обучении НС, т.е. определении весовых коэффициентов  $w_{ij}$  и порогов нейронов  $p_j$  таким образом, чтобы при появлении каждой текущей ситуации  $(g_i, x_i)$  на входе системы,

получился бы соответствующий отклик  $\dot{x}_i$  на ее выходе, согласно (6.27). Следует отметить, что существенное отличие от задач распознавания при этом – обучение нейронной сети в составе замкнутой системы управления, т.е. для обучения нейронной сети используется не рассогласование (ошибка) между выходами регулятора (собственно нейронные сети) и эталона (образа), а ошибка между желаемой характеристикой системы управления в целом и текущим значением выхода проектируемой системы (выходом объекта управления).

Обучение нейронной сети проведено с использованием алгоритма «backpropagation». Согласно рис.6.25, сигналы ошибки регулирования и его скорости изменения (называемых обучающими данными) после масштабирования с соответствующими коэффициентами  $k_e, k_{\dot{e}}$  подаются на вход нейронной сети, т.е. на сенсорные нейроны. После необходимого преобразования с соответствующими весами  $w_{ij}$  полученные сигналы сравниваются, на втором слое, с порогами нейронов  $p_j$  и вычисляются функции активации  $u_j^2$

$$u_j^2 = \frac{\sum_{i=1}^{n_1} y_i w_{ij} - p_j}{1 + \left| \sum_{i=1}^{n_1} y_i w_{ij} - p_j \right|}, \quad j = \overline{n_1 + 1, n_2},$$

где  $n_1, n_2$  - число нейронов, соответственно, первого и второго слоя.

Данные сигналы – входные сигналы нейронов третьего слоя. Аналогичным образом вычисляются выходные сигналы  $u_j^3, j = \overline{n_2 + 1, n_3}$ , где  $n_3$  - число нейронов третьего слоя.

Вычисляется суммарное отклонение между текущими и желаемыми (эталонными) переменными системы управления

$$\Delta(x, \dot{x}, t) = k_{\dot{e}}(\dot{x}d(t) - \dot{x}(t)) + k_e(xd(t) - x(t)), \quad (6.28)$$

где  $xd$  и  $\dot{x}d$  – обучающие данные НС. Тогда на выходном слое НС ошибка вычисляется по формуле

$$G_i^2 = \Delta(1 - |u_i^3|)^2, \quad i = \overline{n_2 + 1, n_3}. \quad (6.29)$$

В скрытом слое НС ошибка вычисляется по формуле

$$G_i^1 = \Delta(1 - |u_i^2|)^2 \sum_{j=n_2+1}^{n_3} G_j w_{ij}, \quad i = \overline{n_1 + 1, n_2}. \quad (6.30)$$

Коррекция весов синаптических связей и порогов нейронов производится следующим образом

$$S_{ij} = G_i^2 u_j + G_i^1 y_j; \quad w_{ij}^h = w_{ij}^c + \alpha S_{ij}; \quad p_i^h = p_i^c + \beta G_i \dots \quad (6.31)$$

где  $w_{ij}^h, w_{ij}^c, p_i^h, p_i^c$  - соответственно, новое и старое значения весов связи и порогов нейронов,  $\alpha$  и  $\beta$  - шаги коррекции.

Для машинного моделирования нейронной САУ разработаны программные средства. Исходные данные: количество входных и выходных параметров, архитектура связей и порогов нейронов, модель объекта управления, а также желаемая характеристика проектируемой системы.

Описанный выше подход использован при синтезе САУ с НС для управления температурой ректификационной колонны. В результате синтеза САУ с НС определены соответствующие значения весов связей и пороги нейронов. Масштабные коэффициенты определены как:  $k_e = 0.5, k_\theta = 0.05, k_u = 0.28$ . Количество нейронов на входном, скрытом и выходном слое равны, соответственно 2, 14, 1.

## Глава 7. ПОСТРОЕНИЕ ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМ

### 7.1. Алгоритм обучения матричного представления нечетких систем логического вывода

Применение теории нечеткой логики при решении задач классификации дает возможность получить принципиально новые модели и методы анализа данных систем. Предлагается нейро - нечеткий алгоритм синтеза систем нечеткого вывода. Описывается двухэтапный адаптивный алгоритм синтеза систем нечеткого вывода. На первом этапе производится кластеризация исходных нечетких параметров с целью сокращения числа входных параметров нечетких правил, а на втором - осуществляется синтез нечетких моделей (правил вывода) типа Мамдани и применение матричного представления нечеткой логики для решения задач классификации..

При решении прикладных задачах в условиях неопределенности нечеткой природы информацию, необходимую для построения и реализации системы принятия решений, можно разделить на две части: численную (количественную), и лингвистическую (качественную), поступающую от эксперта. Значительная часть *систем нечеткого вывода* использует второй вид знаний, чаще всего представляемых в форме базы нечетких правил. Они отображают структуру нечеткой модели задачи в целом и содержат основные знания (экспертную информацию) о моделируемой системе, т.е. главную составляющую «интеллекта» рассматриваемой задачи. Поэтому корректное формирование нечеткой базы правил является важным условием эффективного решения поставленной задачи. Для того, чтобы такая модель была адекватна реальной ситуации, количество формируемых правил в *систем нечеткого вывода* обычно должно быть равно числу условий  $A$  правил, т.е. числу элементов входного вектора системы. Чрезмерно большое их число приводит к увеличению размерности модели и, соответственно, сложности решаемой задачи. Кроме того, объем имеющейся доступной информации, в том числе экспертной, о моделируемой системе часто оказывается недостаточным для построения более сложной и адекватной модели. Следует также учитывать наличие объективных ограничений на точность получения исходных данных. Поэтому при их формировании и оценке в процессе

построения исследуемых моделей следует использовать принцип разумной полноты и точности. Это обуславливает важность систематизации и классификации исходной информации с целью разумного сокращения количества правил систем нечеткого вывода.

При решении задач классификации наиболее распространение получили методы: Mamdani, Tsukamoto, Larsen, Takagi-Sugeno. Ниже представлен сравнительный анализ нечетких моделей [51-53]:

Модели Мамдани может быть определена следующим образом [54,55]:

Определение нечеткой базы правил;

Фаззификация входных переменных;

Агрегирование подусловий, для каждого правила выполняется расчет значений степеней принадлежности. Для расчетов используются правила, где значения степеней принадлежности предпосылок не нулевые;

Активизация подзаклучений в нечетких правилах рассчитывается по формуле, учитывая только активные правила;

Аккумуляция заключений нечетких правил продукций, где выполняется объединение нечетких множеств и получают итоговое множество нечеткое для выходных лингвистических переменных;

При дефаззификации результат приводят к четкому представлению, используя метод центра тяжести.

Рассматривается задача нечеткого логического вывода (НЛВ), описываемая нечеткой моделью Мамдани [52,60]:

$$\bigcup_{p=1}^{k_j} \left( \bigcap_{i=1}^n x_i = a_{i,jp} - \text{с весом } w_{jp} \right) \rightarrow y_j = f_j. \quad (7.1)$$

Здесь:  $x_i$ ,  $y_j$  - входные и выходные переменные;  $j = \overline{1, m}$  - номер правила;  $a_{i,jp}$  - лингвистический терм, которым оценивается входная переменная  $x_i$ , в строке-конъюнкции с номером  $jp$  ( $jp = \overline{1, k_j}$ )  $j$ -го правила;  $w_{jp} = [0,1]$  - весовой коэффициент правила с номером  $jp$ .

В модели (7.1) входные переменные оцениваются нечеткими терминами  $a_{i,jp}$  (например, квантификаторами типа ОН – очень низкий, Н - низкий, НС – ниже среднего, С - средний, ВС – выше



среднего, В - высокий, ОВ – очень высокий), которые описываются собственными функциями принадлежности (ФП). В общем виде ФП описываются выражением:

$$\tilde{\mu}^l(x_i^j) = \frac{1}{1 + \left(\frac{x_i^j - c_l^j}{s_l^j}\right)^2}. \quad (7.2)$$

Здесь  $c_l^j, s_l^j$  - параметры ФП,  $j$  - номер правила,  $l = a_{i,jp}$  – индекс терма.

Данный алгоритм математически может быть описан следующим образом.

1. Нечёткость: находятся степени истинности для предпосылок каждого правила.

2. Нечёткий вывод: находятся уровни отсечения для предпосылок каждого из правил с использованием операции минимума.

3. Композиция: с использованием операции  $\max$  производится объединение найденных усеченных функций, что приводит к получению итогового нечёткого подмножества для переменной выхода с функцией принадлежности

4. Приведение к чёткости производится, например, центроидным методом

Основным назначением этого алгоритма, реализуемого на первом этапе, является сокращение исходного числа входных параметров правил (7.1) и, соответственно, числа правил, с использованием процедур кластеризация и формирование правил СНВ с предварительными (грубыми) значениями параметров описывающих их ФП (7.2).

Алгоритм кластеризации реализуется в следующей последовательности.

Шаг 1. Разделение множества  $\{X_1, X_2, \dots, X_N\}$  на непересекающиеся классы эквивалентности. Представим, что нам известно минимальное и максимальное значение каждой входной и выходной информации. По ним можно определить интервалы, в которых находятся допустимые значения. Для входного сигнала  $x_i$  такой интервал обозначим  $[x_{iq}^-, x_{iq}^+]$ . Если значения  $x_{iq}^-$  и  $x_{iq}^+$  неизвестны, то можно воспользоваться обучающими данными и выбрать из них соответственно минимальное и максимальное значения.

Каждый определенный таким образом интервал разделим на  $K$  областей {отрезков), причем значение  $K$  для каждого сигнала подбирается индивидуально, а отрезки могут иметь одинаковую или различную длину.

Для оценки значений лингвистических переменных будем использовать указанную выше семиуровневую шкалу квантификаторов термов. Каждый из этих термов представляет нечеткое множество, заданное с помощью соответствующей функции принадлежности.

Пользуясь введенными качественными термами (классификаторами) и знаниями эксперта, представим нечеткие правила в виде таблицы, элементами которой являются функции принадлежности термов нечетких правил.

Используя таблицу и операции  $\wedge$  (*И - min*) и  $\vee$  (*ИЛИ - max*), легко записать систему нечетких логических уравнений, связывающих функции принадлежности заключений НЛВ и входных переменных.

В общем случае каждая переменная входного вектора  $X_q^* = (x_{1q}^*, \dots, x_{nq}^*) \quad q = \overline{1, N}$  имеет свои собственные функции принадлежности нечетким термам (ОН, Н, НС, С, ВС, В, ОВ), которые используются в правилах СНВ. Для упрощения моделирования будем использовать для всех переменных входного вектора только одну форму функций принадлежности.

Шаг 2. Построение нечетких правил на основе обучающих данных.

Вначале определим степени принадлежности обучающих данных (3) к каждой области, выделенной на шаге 1. Эти степени будут выражаться значениями ФП нечетких множеств соответствующей группы данных.

Для идентификации параметров заключений в правилах (7.1) предлагается использовать следующий алгоритм:

1. Фиксируются значения входных и выходных параметров состояния объекта:

$$X_q^* = (x_{1q}^*, \dots, x_{nq}^*), \quad Y^* = \{y_q^*\} \quad q = \overline{1, N}.$$

2. Определяются значения функций принадлежности входных параметров  $\mu^q(x_i^*)$  при фиксированных значениях вектора  $X_q^* = (x_{1q}^*, \dots, x_{nq}^*)$ .

3. Вычисляются значения функций принадлежности выходных параметров  $\mu^{y_q}(x_{1q}^*, x_{2q}^*, \dots, x_{nq}^*)$  при фиксированных значениях вектора  $X_q^* = (x_{1q}^*, \dots, x_{nq}^*)$ .

4. Нормализация

$$U_{ij} = l \frac{x_{ij} - x_{minj}}{x_{maxj} - x_{minj}}, l = 4.$$

5. Приведение к нечеткости (фазификация процесса)

$$\mu_{ij}^k = \frac{1}{1 + \frac{(U_{ij} - k)^2}{\sigma^2}},$$

$$\mu_{ij}^* = \max_k \mu_{ij}^k, l = 4, k = 0, 1.$$

6. Расчет термов базы знаний

$$V_{ij} = \arg \mu_{ij}^*.$$

7. Нечеткий логический вывод – Модель Мамдани.

8. Расчет адекватности

$$\mu_{ij} = \frac{1}{1 + \frac{(U_{ij} - k)^2}{\sigma^2}}.$$

В матричном представлении [57] для описания степени истинности высказывания  $A$  вводится двухмерный вектор  $x$ . Если ввести базисные векторы  $e^{(0)}$  и  $e^{(1)}$ , то вектор  $x$  может быть представлен в виде

$$x = x_0 e^{(0)} + x_1 e^{(1)},$$

причем компоненты вектора  $x$  – числа  $x_0, x_1$  – удовлетворяют условиям

$$0 \leq x_0, x_1 \leq 1, x_0 + x_1 = 1.$$

Сами базисные векторы  $e^{(0)}$  и  $e^{(1)}$  описывают высказывания со степенью уверенности 0 и 1 соответственно. Операции отрицания соответствует перестановка компонент вектора  $x: \neg x = x_1 e^{(0)} + x_0 e^{(1)}$ .

В «классической» нечеткой логике оценка неопределенности высказывания вызывает затруднения [59]. Представление нечетких высказываний в векторном виде позволяет вычислять их

неопределенность с помощью известной формулы теории информации:

$$S(x) = -x_0 \log_2 x_0 - x_1 \log_2 x_1. \quad (7.3)$$

Логические операции над векторами – конъюнкцию, дизъюнкцию и импликацию – можно представить в матричном виде. Составим из компонент вектора  $x$  конъюнктивную  $C(x)$ , дизъюнктивную  $D(x)$  и имплицативную  $I(x)$  матрицы размером  $2 \times 2$  [57]:

$$C(x) = \begin{pmatrix} 1 & x_0 \\ 0 & x_1 \end{pmatrix}; D(x) = \begin{pmatrix} x_0 & 0 \\ x_1 & 1 \end{pmatrix}; I(x) = \begin{pmatrix} x_1 & 0 \\ x_0 & 1 \end{pmatrix}. \quad (7.4)$$

С помощью этих матриц логические операции над нечеткими векторами могут быть представлены в виде

$$x \wedge y = C(x)y; x \vee y = D(x)y; x \rightarrow y = I(x)y, \quad (7.5)$$

где вектор  $y$  следует мыслить, как матричный столбец  $1 \times 2$ ,

составленный из его компонент  $y = \begin{pmatrix} y_0 \\ y_1 \end{pmatrix}$ , а точка обозначает

обычное матричное умножение. Результатом логических операций, сформулированных в таком виде, тоже будет нечеткий вектор. Детальное обоснование сделанного выбора матриц (7.4) представлено в работе [57].

Нечеткий предикат  $x(s)$  – это векторная функция, заданная на множестве определения  $S$ , принимающая значения в пространстве нечетких векторов  $F$ , элементы которого описываются формулами (7.3) и (7.4), то есть

$$x(s) = \begin{pmatrix} x_0(s) \\ x_1(s) \end{pmatrix}; \quad (7.6)$$

$$0 \leq x_0(s), x_1(s) \leq 1; x_0(s) + x_1(s) = 1. \quad (7.7)$$

Из определения предиката следует, что в качестве значения  $x_1(s)$  используется значение функции принадлежности нечеткой переменной  $\mu_x(s)$ . В то время как для выполнения условий (7.4) необходимо  $x_0(s) = 1 - \mu_x(s)$ . Операции над нечеткими предикатами сводятся к операциям над нечеткими векторами [60-64].

В приложениях не четкой логики большое распространение получил алгоритм Мамдани, в котором предлагается находить

следствие правила  $y$  как конъюнкцию предпосылки правила  $x$  и некоторой переменной  $w$ , заданной на области определения  $y$ :

$$y = x \wedge w. \quad (7.8)$$

В матричном виде это выражение имеет вид:

$$y = C(x)w. \quad (7.9)$$

Нечеткое правило *modusponens* может быть записано следующим образом через импликацию:

$$x \rightarrow y = z. \quad (7.10)$$

В матричном виде это правило выглядит как:

$$I(x)y = z. \quad (7.11)$$

Из формул (7.9) и (7.11) следует связь между переменными  $w$  и  $z$ :

$$z = C(x)I(x)w. \quad (7.12)$$

Согласно этой формуле, задавая переменную  $w$ , как того требует алгоритм Мамдани, зададим и нечеткость  $z$  правила *modusponens* (7.10). В этом смысле можно строго считать, что алгоритм Мамдани является логическим выводом, хотя обычно его рассматривают только как модель вывода. При решении конкретных задач во многих случаях бывает легче определить переменную  $w$ . Матричный метод позволяет при этом указать степень нечеткости  $z$  правила (7.10).

Вычислительный эксперимент по оценке эффективности предложенного метода классификации проводился на базах данных UCI машинного обучения. Репозиторий UCI (UCI Machine Learning Repository) представляет собой набор реальных и модельных задач машинного обучения, которые используются научным сообществом для эмпирического анализа алгоритмов машинного обучения. Содержит реальные данные по прикладным задачам в области биологии, медицины, физики, техники, социологии, и др. Архив был создан в 1987 году Дэвидом Ага и коллегами аспирантами Калифорнийского университета в городе Ирвин (School of Information & Computer Science University of California, Irvine, USA, <http://www.ics.uci.edu>) . С этого времени, он широко используется студентами, преподавателями и исследователями во всем мире в качестве основного источника машинного обучения наборов данных.

На основе матричного представления нечеткой логики Мамдани на модельных задачах получены результаты классификации и проведен сравнительный анализ .

Таблица 7.1

Результаты, полученные по простой модели Мамдани (%).

N	Базы взяты для тестирования				Изменения значений L						
	Имя	N	M	Имя класса	3	5	7	9	11	13	15
1	Animal	101	16	7	100,0	100,0	100,0	100,0	100,0	100,0	100,0
2	Wine	178	13	3	98,33	100,0	100,0	100,0	100,0	100,0	100,0
3	Iris	150	4	3	67,33	96,00	97,3	98,00	98,67	100,0	100,0
4	Liver	178	10	2	80,34	90,45	96,63	98,88	99,44	99,44	99,44
5	Caesarian	80	5	2	57,50	57,50	57,50	57,50	57,50	57,50	57,50
6	Fire	244	10	2	86,07	93,85	96,72	99,18	99,18	100,0	100,0
7	Ecoli	336	7	8	66,96	87,50	93,45	97,02	98,81	99,70	100,0

Таблица 7.2

Результаты, рассчитанные по матричной модели Мамдани (%).

N	Базы взяты для тестирования				Изменения значений L						
	Имя	N	M	Имя класса	3	5	7	9	11	13	15
1	Animal	101	16	7	100,0	100,0	100,0	100,0	100,0	100,0	100,0
2	Wine	178	13	3	100,0	100,0	100,0	100,0	100,0	100,0	100,0
3	Iris	150	4	3	85,33	98,00	98,67	99,33	99,33	100,0	100,0
4	Liver	178	10	2	86,52	94,94	97,75	98,88	99,44	100,0	99,44
5	Caesaria n	80	5	2	57,50	57,50	57,50	57,50	57,50	57,50	57,50
6	Fire	244	10	2	96,31	97,54	100,0	100,0	100,0	100,0	100,0
7	Ecoli	336	7	8	77,38	89,88	94,05	96,43	98,81	100,0	100,0

Эксперименты были выполнены 10 раз для каждого набора данных с использованием 10-кратной кросс-проверки (crossvalidation) [66, 167].

Предложенный адаптивный алгоритм позволяет упрощать процедуры синтеза нечетких правил СНВ за счет существенного снижения размерности множества исходных переменных и осуществлять оперативную корректировку нечетких моделей в условиях изменяющихся параметров среды. Результаты вычислительных экспериментов, проведенных для классификации элементов тестовых баз данных с использованием известных

методов, показали более высокую точность предложенного двухэтапного адаптивного алгоритма классификации. Предложенный подход был апробирован также для решения задач прогнозирования с использованием реальных данных. Полученные результаты показали высокую эффективность нечетких моделей прогнозирования, синтезированных по предложенному алгоритму [61-63].

Перспективным направлением исследований по рассматриваемой проблематике является разработка методов и алгоритмов синтеза правил СНВ с использованием комбинации средств "Soft Computing"- технологии: нечетких множеств, нейронных сетей и генетических алгоритмов.

## **7.2. Применение генетического алгоритма для решения задач оптимизации размещения и чередования культур в хлопковом севообороте**

Оптимальная организация территории севооборотных полей и массивов позволяет провести целесообразное размещение культур по полям севооборота и организовать их чередование в период ротации. При этом решаются две взаимосвязанные задачи. Первая - по размещению хлопчатника и других культур по полям севооборота. Реализация позволяет определить - на каком поле следует высевать ту или иную культуру. Такая задача рассматривается как статическая и ее модель разрабатывается с учетом основных природно-экономических условий хозяйства. Вторая задача рассматривается как динамическая. Она позволяет определить в какие годы ротации на каком поле следует высевать ту или иную культуру. Разработка модели задачи основывается на информации, получаемой от статической задачи и из таблицы чередования культур в севообороте. Основная цель статьи решение задач оптимизации размещения и чередования культур в хлопковом севообороте. Для решения этой задачи применяется генетический алгоритм. Проводился вычислительный эксперимент.

Для рационального размещения хлопчатника и кормовых культур с их сочетанием по полям севооборота определяется несколько показателей, характеризующих пригодность полей к

посеву той или иной культуры. Главным из них, по нашему мнению, является урожайность севооборотной культуры на том или ином поле. Поэтому прогноз урожайности культур по севооборотам полей занимает особое место в системе задач оптимизации севооборота [71]. Такая задача реализуется в два этапа. На первом этапе определяется потенциальная урожайность севооборотных культур на каждом поле севооборота. Расчет выполняется по модели множественной корреляции по типам почв с предварительной оценкой пригодности почвы для посева данного вида культуры. При этом в модель включается комплекс природных и экономических факторов, характеризующих состояние почвы и других условий производства. Результаты прогноза служат основной информацией для статической модели размещения культур по полям севооборота [72]. На втором этапе решается задача по прогнозу урожайности культур севооборотных полей с учетом чередования хлопчатника и его предшественников на период ротации. Опыт ученых и практика освоения севооборотов в хозяйствах показывают, что в первые годы посева хлопчатника после распашки люцерны на севооборотных полях получают высокие урожаи, начиная с 3 и 4 годов посева продуктивность полей снижается, соответственно уменьшается и урожайность хлопчатника [73]. Это должно учитываться в модели как основное явление, происходящее при чередовании культур в севообороте. Таким образом, динамика урожайности культур по годам служит основной информацией при определении чередования культур.

Учитывая изложенное, нами разработаны статическая и линейно-динамическая модели задачи размещения культур по полям севооборотов и их чередования в период ротации [74].

Для математической формализации модели задач введем дополнительные обозначения:

$P_r^k$  - размер  $r$ -го севооборотного поля в  $K$ -схеме севооборота;

$u_{rj}^k$  - урожайность  $j$ -й культуры на  $r$ -м поле севооборота при  $K$ -схеме севооборота;

$t_{jz}^k$  - трудовые затраты на 1 га  $j$ -й культуры;  $c_{jk}^k$  - стоимость валовой продукции с 1 га  $j$ -й культуры;

$Z_{1j}^k$  - производственные затраты на 1 га  $j$ -й культуры;



$\alpha_1^k, \alpha_2^k, \alpha_3^k$  - количество полей в  $K$ -й схеме севооборота;  
 $T^{**}$  - трудовые ресурсы;  
 $V_1$  - общий объем производственных затрат;  
 $Q_i$  - необходимый объем производства  $i$ -го вида сельскохозяйственной продукции;  $x_{zj}$  коэффициент, определяющий долю  $r$ -го поля в  $K$ -й схеме севооборота для посева  $j$ -й культуры;

$J \in J_1, J_2, J_3$  - множество индексов переменных по севооборотным культурам.

С учетом принятых обозначений статическая модель задачи оптимизации размещения культур по полям севооборота выглядит следующим образом.

Требуется найти оптимальный план размещения культур по полям севооборота, позволяющий получить наибольший размер чистого дохода, выражаемый в виде линейной функции:

$$F(x) = \sum_{j=1}^J \sum_{K=1}^K P_r^K (c_{1j} - Z_{1j}) x_j^k \rightarrow \max$$

при выполнении следующих ограничений:

$$\sum_{j=1}^J \sum_{r=1}^R x_{rj}^k = 1, \quad (7.13)$$

$$x_{rj}^k = \begin{cases} 1, & \text{если } j\text{-я культура высевается на } r\text{-м поле в } K\text{-й схеме севооборота;} \\ 0, & \text{в противном случае;} \end{cases}$$

$$\sum_{j=\gamma+1}^{J_1} \sum_{r=1}^R u_{rj}^K x_j^k \geq \alpha_1^K u_j^K \left( \min_j \right), \gamma = 0, 3, 6, 9, \dots \quad (7.14)$$

$$\sum_{j=\gamma+2}^{J_2} \sum_{r=1}^R u_{rj}^K x_j^k \leq \alpha_2^K u_j^K \left( \max_j \right), \quad (7.15)$$

$$\sum_{j=\gamma+3}^{J_3} \sum_{r=1}^R u_{rj}^K x_j^k \leq \alpha_3^K u_j^K \left( \max_j \right), \quad (7.16)$$

$$\sum_{j=1}^J \left( \sum_{k=1}^K \sum_{r=1}^R P_r^k \cdot h_{rj}^k \right) \cdot x_j^k \leq T^{**}, \quad (7.17)$$

$$\sum_{j=1}^J \left( \sum_{k=1}^K \sum_{r=1}^R P_r^k c_{rj}^K \right) \cdot x_j^k \geq 0, \quad (7.18)$$

$$\sum_{j=1}^J \left( \sum_{k=1}^K \sum_{r=1}^R P_r^k z_{rj}^K \right) \cdot x_j^k \leq V_1, \quad (7.18)$$

$$\sum_{j=1}^J \left( \sum_{k=1}^K \sum_{r=1}^R P_r^k u_{rj}^K \right) \cdot x_j^k \geq Q, \quad (7.20)$$

$$x_{rj}^k \geq 0. \quad (7.21)$$

Содержание отдельных ограничений и условий следующее. Равенство (7.13) отражает условия выбора одного и только одного поля севооборота под посев данной культуры. Неравенства (7.14)–(7.16) реализует условие, при котором каждая культура должна быть весляна там, где от нее получается максимальный урожай. Такое ограничение вводится по каждому виду культур, которые можно условно разделить на три группы: первая - хлопчатник, вторая - кормовые культуры, третья - кукуруза, пшеница и ячмень.

Неравенства (7.17), (7.18) и (7.19) позволяют оценить размещаемые культуры и севообороты с точки зрения их трудоемкости (7.17), производственных затрат (7.19) и возможности получения валовой продукции (7.18) на севооборотном поле.

Неравенство (7.20) реализует требования обязательного выполнения плана заготовок сельхозпродуктов, соотношение (7.21) - условия по неотрицательности переменных.

Решение задачи по данной модели позволяет определить, на каких полях следует размещать тот или другой вид культур. Такой расчет является основой для планирования чередования культур в севообороте. Для решения этой задачи оптимизации используется генетический алгоритм. В общем случае процедура оптимизации на основе обычного последовательного комплекса-метода выглядит следующим образом: требуется отыскать максимум некоторой функции  $F(x)$  при выполнении условий (7.13)–(7.21).

Работа такого алгоритма образована последовательностью следующих шагов:

- создание начальной популяции;
- операция скрещивание с увеличением популяции;
- операция мутации;
- первая селекция (определение наихудших особей) без сокращения популяции;
- операция выбора, заменяется значения наилучшему во всей популяции;

При решении задач по статической модели определены номера и размеры полей севооборота, на которых возможно размещение той или иной культуры по выбранным схемам. Основными условиями, включенными в модель, были величины урожайности культур на одном и том же поле, объем производства хлопка, производственные, трудовые затраты и условно чистый доход. Критерием оптимальности задачи служили минимизация производственных затрат и максимизация условного чистого дохода, исчисляемого как разность между стоимостью валовой продукции с 1 га и производственными затратами на 1 га. Все исходные данные рассчитывались исходя из многолетних данных колхоза. Информация подготавливалась по каждой схеме севооборота. Это дало возможность оценить все севооборотные схемы и поля в комплексе с учетом основных показателей производства колхоза. Исходные данные приведены в приложении 6. Задачи решались по двум указанным выше критериям оптимальности. Путем сопоставительного анализа итоговых результатов и их эффективности для дальнейшего исследования выбран второй вариант результатов, полученных при критерии минимизации производственных затрат. Основные результаты этого варианта приведены в табл.7.3. При выбранном размещении культур по полям севооборота объем производства хлопка составит 32,5 тыс ц,

Таблица 7.3

## Размещение культур по полям севооборота

Севооборотное поле	Схема севооборота			
	1:2:7	1:4:1:4	2:4:1:3	2:8
	X	X	X	X
I	26,0	36,2	21,5	34,6
	K + K	X	Л + Я	X
	24,0	38,0	21; 7	35,4
	X	X	Л + Я	X
III	24,6	38,2	22,0	32,1
	X	X	X	X
IV	24,0	36,2	22,0	32,3
	Л + Я	X	X	X
V	22,5	38,9	22,4	35,6
	X	K + K	X	X
VI	25,0	37,2	22,2	34,4
	X	K	K + K	X
VII	25,6	37,8	22,6	31,5
	X	X	X	Л + Я
VIII	24,0	37,3	22,6	34,8
IX	X	X	X	Л + Я
	24,0	37,5	22,0	35,8
X	Л + П	X	X	X
	24,6	35,2	22,0	34,2

В числителе - вид севооборотной культуры, в знаменателе - размер севооборотного поля.

Здесь: X - хлопок; K - кукуруза; Кп - корнеплоды;

I - люцерна; Я - ячмень; П - пшеница

Таблица 7.4

## Урожайность культур по схемам севооборота в среднем

Севооборотные культуры	Вид продукции	Урожайность культур по схемам			
		2:8	2:4:1:3	1:4:1:4	1:2:7
Люцерна+ячмень	Сено	76,0	74,5	—	74,5
	Зерно	32,0	31,6	—	31,6
	Солома	32,0	31,6	—	31,6
Люцерна (2 года)	Сено	157,0	152,0	—	152,0
Люцерна+пшеница	Сено	75,2	74,5	—	74,5
	Зерно	32,0	30,0	-	30,2
	Солома	32,0	30,2	—	30,2
Люцерна (2 года)	Сено	157,0	152,0	—	152,0
Кукуруза	Зерно	—	90,0	93,0	90,0
	Силос	—	492,8	492,8	492,8
Кукуруза а+корнеплоды	Зерно	—	87,5	87,5	87,5
	Силос	-	482,2	482,2	482,2
	Зел.корм	—	490,0	490,0	490,0
	Свекла	—	411,0	411,0	411,0
Кукуруза а+люцерна	Зел.корм	—	390,0	390,0	390,0
Хлопчатник	Хлопок	35,8	37,0	35,0	37,15

При выбранном размещении культур по полям севооборота объем производства хлопка составит 32,5 тыс ц, сено люцерны - 11,3 тыс ц, зерно кукурузы - 10,2 тыс ц. Урожайности культур, при которых достигается такой объем производства, приведены в таблице 2. Как показывают данные табл. 7.3, на 10 полях по указанным схемам высеваются кормовые культуры в сочетании с зерновыми, на остальных 30 полях - хлопчатник. Так, по схеме

1:2:7 хлопчатник в первый год ротации высевается на I, III IV, VI-IX полях. Такое размещение культур по полям севооборота позволяет по всем схемам перейти к организации чередования культур на перспективу. Результаты статической задачи (номера полей и их размеры) служат исходной информацией и отправной точкой для решения задачи размещения и чередования культур на весь период ротации. Такая задача решается с применением линейно-динамической модели.

Статическая модель может быть разработана на основе транспортной задачи линейного программирования. Однако при этом невозможно учитывать различные варианты посева культур на том или ином поле с условием выполнения планов заготовок. В работе [46] описана модель размещения хлопчатника и кормовых культур по полям севооборота с учетом их перспективного чередования. Модель задачи позволяет в статике определить размещение культур по полям севооборота без учета потенциальной урожайности полей. В работе [46] автор отмечает: "... для каждого периода ротации составляет отдельная матрица задачи, которая решается самостоятельно". Как известно, при таком подходе нарушается зависимость чередования хлопчатника от предшественников и возникает многократное решение одной и той же задачи. При этом сопоставление результатов одного года ротации с другим несколько затруднительно и нарушается комплексность расчета.

Применение новых подходов на основе эволюционных методов для оптимизации нечетких баз правил позволит значительно снизить время формирования решения и повысить достоверность его принятия в интеллектуальных системах.

Решена задача оптимизации размещения и чередования культур в хлопковом севообороте на основе использования генетического алгоритма с искусственным отбором и создана программное обеспечение для решения данной поставленной задачи на основе использования генетического алгоритма с искусственным отбором. Сложившееся состояние землепользования и размещение объектов позволяют внести некоторую корректировку в систему введения севооборотов (исходя из оптимальных их схем) и организовать территории полей севооборотов.

### **7.3. Применение нейронечеткого подхода для решения задач оценки почвенных разностей**

Роль сорта в эффективном использовании удобрений огромна. Однако большинство исследователей ограничивались только констатацией фактов о различной продуктивности сортов в тех или иных условиях минерального питания. Сорта несут в себе "отпечаток" тех условий, в которых они созданы. Отсюда необходимо изучать, очевидно, особенности их питания, формирования урожая и качества продукции при разных формах, дозах, соотношениях, сроках и способах внесения минеральных удобрений на различных почвах зоны возделывания данной культуры. Основная цель статьи оценки почвенных разностей на основе использования нейронечеткого подхода. Рассматриваются три почвенные разности: орошаемый типичный серозем, сероземно – луговая почва и новоорошаемый светлый серозем. Проведен вычислительный эксперимент по оценке типа почвы с учетом характеристик, таких как плотность почвы и гумус в процентном отношении.

Агрохимики изучали в основном реакцию разных сортов хлопчатника на удобрения, фиксируя изменения роста и урожайности. Изменения те морфологических, физиологических процессов, структуры урожая, технологических свойств волокна, масличности семян у разных сортов при разных нормах и соотношениях удобрений мало изучены. Кроме того, проведение таких исследований на одной какой-нибудь почве еще не дает необходимого материала, чтобы поверить об отзывчивости того или иного сорта хлопчатника на удобрения на других почвах [71-73].

Почвы зоны хлопководства, во первых, сильно различаются по глубине залегания грунтовых вод. Понятно, что сорт, выведенный в условиях автоморфных почв, на гидроморфных почвах будет расти и развиваться, но влияние новых почвенных условий бесследно не пройдут. Известно, что в почвах с близким залеганием грунтовых вод, наблюдается совершенно другой водный, воздушный, температурный режим, иной состав микроорганизмов, другое соотношение аммиачных и нитратных форм азота, фосфорный соединений и калия. Весной

гидроморфная почва сильно насыщается водой, создаются анаэробные условия для микроорганизмов, в ней больше аммиачного азота, хуже растворимость фосфорных соединений, почва имеет сравнительно низкую температуру и т.д. Влияние водного режима почвы настолько сильно, что почва, показывающая в первом минимуме азот при одних условиях влажности, в других показывает в первом минимуме фосфор [74-76].

Во-вторых, механический состав в значительной степени определяет специфичность почвенных условий. Сорт, выведенный на почвах легкого механического состава (у которых отличные водно-воздушный, температурный и пищевой режимы), нельзя рекомендовать для почв тяжелого механического состава, так как сорта, оказавшись в совершенно другой среде, снижают урожайность. Во всех почвенных зонах на более легких почвах поднимается эффективность удобрений, особенно азотистых и калийных [77-78].

В третьих, высококультуренные почвы оптимально сочетают в себе факторы и растения, используя их наиболее продуктивно, обеспечивают максимально возможный урожай. Поэтому сорт, выведенный на такой почве, на средне и слабокультуренной снизит урожай.

Рассмотрим три почвенные разности: орошаемый типичный серозем, сероземно – луговая почва и новоорошаемый светлый серозем.

Почвообразующими породами орошаемых типичных серозем являются мощные мезовидные отложения с благоприятными воднофизическими свойствами. Грунтовые воды залегают глубже 10 м. Поэтому они не оказывают влияния на формирование почв. Земли издавна обрабатывались, и на них возделывались разные сельскохозяйственные культуры.

Описываемый орошаемый типичный серозем характеризуется очень низким содержанием как в пахотном, так и подпахотном слое гумуса, азота, а также подвижных форм фосфора и калия. По - видимому, основная причина этого в том, что участок многие годы использовался под овощные культуры без внесения достаточного количества удобрений.

С развитием орошения, вызвавшим подъем грунтовых вод до 1,5 - 2,0 м, орошаемые сероземно - луговые почвы развиваются в



условиях постоянного капиллярного увлажнения. В результате начинается луговой процесс: верхние горизонты становятся более гумусированными, а глубокие - переувлажненные, приобретают сизый оттенок - признак оглиения.

По механическому составу указанные почвы тяжелосуглинистые. Профиль разреза по содержанию физической глины довольно однороден. Водно-физические и агрохимические свойства этой почвы также отличаются от орошаемого типичного серозема.

По механическому составу весь полуметровый почвенный профиль новоорошаемый светлый серозем относится к легкому суглинку. Среди частиц больше всего крупной пыли, что оказывает большое влияние на водно - физические свойства почвы (табл.6). Наименьшая и полная влагоемкость небольшие. Воздух занимает достаточный объем почвы как при наименьшей, так и полной влагоемкости.

Почва характеризуется малым содержанием гумуса и общего азота в пахотном слое. Видимо, основной причиной этого является интенсивное разложение органических веществ в первые годы орошения. Постепенное уменьшение их количества наблюдается в нижних слоях. Мало содержится также как общего, так и подвижного фосфора. Наибольшее количество его отмечено в пахотном слое. Калий подвижный в пахотном и подпахотном слоях имеется в значительном количестве, в нижних слоях его заметно меньше.

Для оценки земли целесообразно использование нечеткой и нейронечеткой логической модели.

"Нечеткость" человеческого знания и образа мышления считается сейчас необходимым компонентом любой системы, по своим возможностям адекватной способностям и интеллекту человека. Дальнейшее построение строгой математической теории и ее применения в различных областях знаний связаны с такими известными учеными как Р. Беллман, А. Кофман, Т. Саати, а также с самим Л.А. Заде[82-83].

Процесс построения нечеткой экспертной системы для оценки почв выполняется по следующему алгоритму:

1. Нормирование входных данных. В общем случае каждая входная переменная  $x_i, i = \overline{1, n}$  имеют свои собственные функции

принадлежности нечетким термам (Н - низкий, нС – ниже среднего, С - средний, вС – выше среднего, Высокий), которые используются в уравнениях. Для упрощения моделирования будем использовать для всех переменных  $x_i, i = \overline{1, n}$  только одну форму функций принадлежности. Для этого приведем интервалы изменения каждой переменной к одному универсальному интервалу  $[0, l]$  с помощью следующих соотношений:

$$\mu^j(x_i) = \tilde{\mu}^j(u),$$

$$u = l \frac{x_i - x_i}{x_i - x_i}, \quad j = H, \text{ нС}, C, \text{ вС}, B,$$

где  $[x_i, \overline{x_i}]$ - интервал изменения переменной  $x_i, i = \overline{1, n}$ .

2. Фаззификация. Аналитическая модель функций принадлежности имеет следующий вид:

$$\tilde{\mu}^j(u) = \frac{1}{1 + \left(\frac{u-b}{c}\right)^2}. \quad (7.22)$$

Выбор таких функций обусловлен тем, что они являются хорошими аппроксимациями функций принадлежности, полученных от эксперта методом парных сравнений.

3. Принятие решение. Для входных данных  $X^* = (x_1^*, x_2^*, \dots, x_n^*)$  определяется значения функций принадлежности  $\mu^j(x_i^*), i = \overline{1, n}$ .

4. Вычисление  $\mu^{d_j}(x_1^*, x_2^*, \dots, x_n^*)$  при  $X^* = (x_1^*, x_2^*, \dots, x_n^*)$  для всех  $d_1, d_2, \dots, d_m$ . Здесь:

$$\mu(a) \wedge \mu(b) = \min[\mu(a), \mu(b)],$$

$$\mu(a) \vee \mu(b) = \max[\mu(a), \mu(b)].$$

4°. Определяется  $d_j^*$ , для которого:

$$\mu^{d_j^*}(x_1^*, x_2^*, \dots, x_n^*) = \max_{j=1, n} [\mu^{d_j}(x_1^*, x_2^*, \dots, x_n^*)].$$

После определения алгоритма работы системы и разбивки (если требуется) задачи на подзадачи следует разработка схемы обучающих примеров для каждой из подзадач. Схема примера включает список входных и выходных параметров для данной подзадачи.

Проведен вычислительный эксперимент по оценки типа почвы с учетом следующих характеристик: средняя температура за

апрель, среднее количество осадков за апрель, гумус (%), плотность (г\см<sup>3</sup>), (табл.7.5).

Таблица 7.5

Фрагмент входных параметров вычислительного эксперимента

№	Средняя температура за апрель (°C)	Среднее количество осадков за апрель (мм)	Гумус (%)	Плотность (г\см <sup>3</sup> )	Тип почвы
1	15	68,8	1,14	1,27	1
2	15	68,8	0,83	1,37	1
3	15	68,8	0,51	1,34	1
4	15	68,8	0,42	1,32	1
5	14,1	50,5	1,24	1,26	2
6	14,1	50,5	0,98	1,41	2
7	14,1	50,5	0,62	1,38	2
8	14,1	50,5	0,43	1,42	2
9	15,8	42,2	0,64	1,4	3
10	15,8	42,2	0,46	1,41	3
11	15,8	42,2	0,56	1,44	3
12	15,8	42,2	0,21	1,47	3

Результаты вычислительного эксперимента по алгоритму принятия диагностических решений с использованием нейронных технологий приводятся в табличном (табл.7.6) и графическом виде (рис.7.1 и рис.7.2).

Таблица 7.6.

Фрагмент выходных данных вычислительного эксперимента

1.№, 2 Средняя температура за апрель ( $^{\circ}C$ ), 3 Среднее количество осадков за апрель (мм), 4 Гумус (%), 5 Плотность ( $г/см^3$ ), 6 Тип почвы, 7 Тип почвы полученный по нечеткой модели, 8 Тип почвы полученный по нейронечеткой модели, 9 Тип почвы полученный по линейной регрессионной модели, 10 Тип почвы полученный по нелинейной регрессионной модели, 11 Ошибка нечеткой модели, 12 Ошибка нейронечеткой модели, 13 Ошибка линейной регрессионной модели, 14 Ошибка нелинейной регрессионной модели.

1	15,00	68,80	1,14	1,27	1,00	1,00	1,00	1,00	1,00	0,00	0,00	0,05	0,06
2	15,00	68,80	0,83	1,37	1,00	1,00	1,00	1,00	1,00	0,00	0,00	0,03	0,03
3	15,00	68,80	0,51	1,34	1,00	1,00	1,00	1,00	1,00	0,00	0,00	0,05	0,01
4	15,00	68,80	0,42	1,32	1,00	1,00	1,00	1,00	1,00	0,00	0,00	0,08	0,01
5	14,10	50,50	1,24	1,26	2,00	2,00	2,00	2,00	2,00	0,00	0,00	0,04	0,03
6	14,10	50,50	0,98	1,41	2,00	2,00	2,00	2,00	2,00	0,00	0,00	0,04	0,02
7	14,10	50,50	0,62	1,38	2,00	2,00	2,00	2,00	2,00	0,00	0,00	0,00	0,01
8	14,10	50,50	0,43	1,42	2,00	2,00	2,00	2,00	2,00	0,00	0,00	0,01	0,00
9	15,50	46,20	0,64	1,40	3,00	3,00	3,00	2,99	3,00	0,00	0,00	0,28	0,01
10	15,70	47,20	0,46	1,41	3,00	3,00	3,00	3,01	3,00	0,00	0,00	0,25	0,00
11	15,80	48,20	0,56	1,44	3,00	3,06	3,06	2,98	2,96	2,12	1,98	0,76	1,44
12	15,90	49,20	0,21	1,47	3,00	3,13	3,12	2,95	2,91	4,24	3,93	1,81	2,86
13	15,20	67,80	0,51	1,34	1,00	0,99	0,99	1,17	1,16	1,45	1,33	17,11	15,84
14	15,10	65,80	0,42	1,32	1,00	0,96	0,96	1,28	1,26	4,36	4,13	27,93	25,51
15	14,30	49,00	1,24	1,26	2,00	1,94	1,95	2,21	2,22	2,97	2,65	10,56	10,98
16	14,50	49,50	0,98	1,41	2,00	1,96	1,97	2,27	2,27	1,98	1,63	13,32	13,34

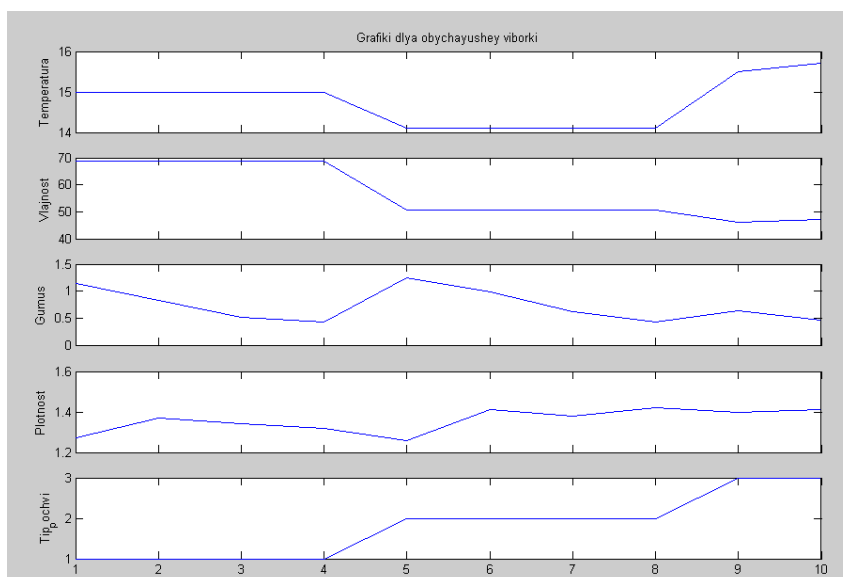


Рис.7.1 Обучающая выборка для оценки почвы

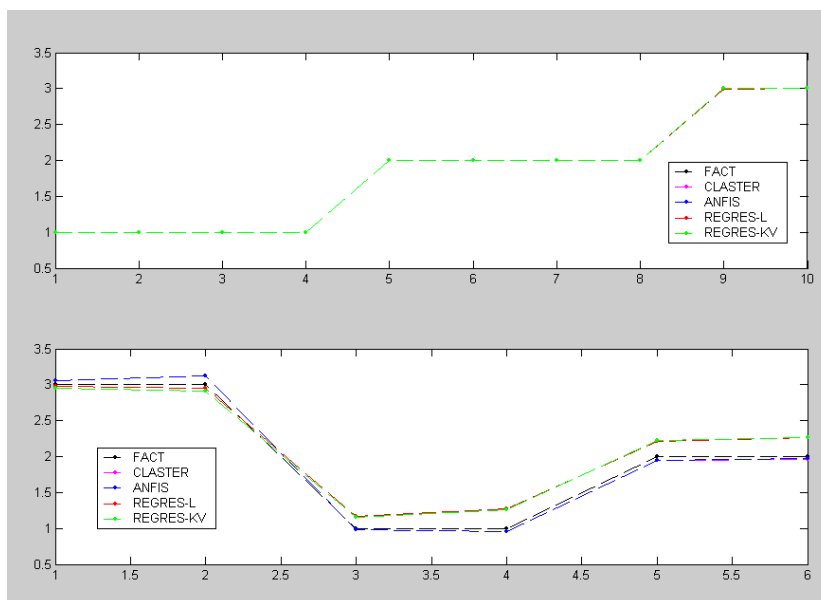


Рис.7.2 График результатов оценки почвы

Таким образом, преимущества нейронечетких экспертных систем перед обычными, которые, как уже говорилось, проявляются только при решении трудно решаемых задач. Как показывает вычислительный эксперимент погрешность по нечеткой и нейронечеткой модели составляет 1,33-4,13%, по линейной регрессионной модели - 1,81-27,93% а по нелинейной регрессионной модели - 2,86-25,51%.

#### **7.4. Оценка плодородия почвы на основе нечеткой логической модели Сугено**

С окультуриванием почв не только растет урожайность сельскохозяйственных культур, но и повышается эффективность минеральных удобрений, при этом у отдельных видов удобрений изменяется поразному. На различных почвах создается разное взаимодействие между почвой и удобрениями, и различные сорта, безусловно, по разному реагируют на них, потому что каждый сорт выводился при одном каком-нибудь из этих условий взаимодействия, и влияние его в нем фенотипически закреплено. Установлено, что плодородие разных типов почв количественно лучше всего характеризуется наряду с влагозапасами, объемной массой, тесно связанной с такими общепризнанными компонентами плодородия, как количество гумуса, азота, фосфора и т.д. Основная цель работы построение нечеткой логической модели Сугено для оценки плодородия почвы.

Шаблонное отношение к почвенным условиям означает, что генетически закрепленное приспособление к тем или иным почвенным условиям, потом искусственно разрушается [71-73]. Именно это не дает сорту показать свою потенциальную урожайность и высокое качество. Поэтому существующие сорта хлопчатника нужно изучать на разных почвах, внося разные формы, дозы удобрений разными способами и неодинаковые сроки с тем, чтобы найти то взаимодействие между сортом, почвой и удобрениями, которое оптимально для того или иного сорта [74-75]. При длительном применении фосфорных удобрений в почвах всех типов и почвенных разностях увеличивается содержание подвижных форм фосфора. Поэтому эффективность вновь вносимых фосфорных удобрений падает [76]. По мере окультуривания почв растет эффективность калийных удобрений на фоне азотных и фосфорных [77-78].

Важной задачей считается построение нечеткой модели на основе экспериментальных данных, улучшение построения нечеткой логической модели Сугено для оценки плодородия почвы. Решение задач интеллектуального анализа характеризуется

недостаточностью числовых вычислений и неполнотой важной информации об условиях задачи [79-81].

Для построения модели для оценки плодородия почвы от экспертов получена выборка  $(X_r, y_r)$ ,  $r = \overline{1, M}$ , где  $X_r = (x_{r,1}, x_{r,2}, \dots, x_{r,n})$  - входной вектор  $r$ -пары и  $y_r$  - соответствующий вектор выхода.

Нашей задачей является построение нечеткой модели следующим образом:

$$\bigcup_{p=1}^{k_j} \left( \bigcap_{i=1}^n x_i = a_{i,jp} - w_{jp} \text{ с весом} \right) \rightarrow y_j = b_{j,0} + b_{j,1}x_1 + \dots + b_{j,n}x_n + b_{j,n+1}x_1^2 + \dots + b_{j,2n}x_n^2 + \dots + b_{j,n+l-1}x_1^l + \dots + b_{j,ln}x_n^l. \quad (7.23)$$

При построении этой модели, если  $l=0$ , случай рассматривается как модель формы Сингилтона [82]. Линейная модель в представлении Сугено, состоящая из выводов нечетких правил в случае  $l=1$ , изучена в работах [83]. Случай с  $l=2$  рассмотрен в работе [84].

В процессе построения модели необходимо найти значения коэффициентов нечетких правила вывода так

$$B = (b_{1,0}, b_{2,0}, \dots, b_{m,0}, b_{1,1}, b_{2,1}, \dots, b_{m,1}, \dots, b_{1,n}, b_{2,n}, \dots, b_{m,n}, \dots, b_{1,n}, b_{2,n}, \dots, b_{m,n}),$$

$$i = \overline{1, n}, j = \overline{1, m}$$

и минимизировать следующую функцию:

$$\sum_{r=1, M} (y_r - y_r^f)^2 \rightarrow \min, \quad (2)$$

где  $y_r^f$  - выход входных данных в  $r$ -строке выборки  $(X_r)$  в нечёткой базе знаний -  $b$ -параметром.

Решение задачи (1) соответствует решению следующего уравнения  $Y = A \cdot B$ ,

где

$$A = \begin{bmatrix} \beta_{1,1}, \dots, \beta_{1,m}, & x_{1,1} \cdot \beta_{1,1}, \dots, x_{1,1} \cdot \beta_{1,m}, & \dots, & x_{1,n} \cdot \beta_{1,1}, \dots, x_{1,n} \cdot \beta_{1,m} \\ \vdots \\ \beta_{M,1}, \dots, \beta_{M,m}, & x_{M,1} \cdot \beta_{1,1}, \dots, x_{M,1} \cdot \beta_{1,m}, \dots, x_{M,n} \cdot \beta_{M,1}, \dots, x_{M,n} \cdot \beta_{M,m} \end{bmatrix},$$

в дискретном случае  $\beta_{j,r} = \frac{\mu_{f_j}(X_r) \cdot f_j}{\sum_{k=1}^m \mu_{f_j}(X_r)}$ , а в непрерывном случаи

$$\beta_{j,r} = \frac{\mu_f(X_r) \cdot f_j}{\int_{f_-}^{f_+} \mu_f(X_r) df}$$





нечёткого множества построена функция принадлежности. После этого определяются правила, которые связывают выходные и входные переменные с соответствующими нечёткими множествами.

Если  $x_1^1 = C$  и  $x_2^1 = C$  и  $x_3^1 = H$  то

$$y = 46,14 + 0,54 \frac{\sum_{j=1}^n \mu(x_1^{1j}) x_1^{1j}}{\sum_{j=1}^n \mu(x_1^{1j})} - 0,64 \frac{\sum_{j=1}^n \mu(x_2^{1j}) x_2^{1j}}{\sum_{j=1}^n \mu(x_2^{1j})} - 21,17 \frac{\sum_{j=1}^n \mu(x_3^{1j}) x_3^{1j}}{\sum_{j=1}^n \mu(x_3^{1j})}.$$

Если  $x_1^1 = B$  и  $x_2^1 = B$  и  $x_3^1 = C$  то

$$y = -13,42 - 0,86 \frac{\sum_{j=1}^n \mu(x_1^{1j}) x_1^{1j}}{\sum_{j=1}^n \mu(x_1^{1j})} + 0,5 \frac{\sum_{j=1}^n \mu(x_2^{1j}) x_2^{1j}}{\sum_{j=1}^n \mu(x_2^{1j})} + 11,51 \frac{\sum_{j=1}^n \mu(x_3^{1j}) x_3^{1j}}{\sum_{j=1}^n \mu(x_3^{1j})}.$$

Если  $x_1^1 = H$  и  $x_2^1 = H$  и  $x_3^1 = B$  то

$$y = -5,45 - 0,06 \frac{\sum_{j=1}^n \mu(x_1^{1j}) x_1^{1j}}{\sum_{j=1}^n \mu(x_1^{1j})} + 0,15 \frac{\sum_{j=1}^n \mu(x_2^{1j}) x_2^{1j}}{\sum_{j=1}^n \mu(x_2^{1j})} + 0,009 \frac{\sum_{j=1}^n \mu(x_3^{1j}) x_3^{1j}}{\sum_{j=1}^n \mu(x_3^{1j})}.$$

В орошаемом типичном сероземепочва - среднесуглинистая с содержанием крупнопылеватых частиц 42,2 - 50,2%, что благоприятствует физическим и водно-физическим свойствам почвы. Песчаных фракций здесь 6,0 - 8,6% массы почва, физической глины - 41,8 - 45,1%. Плотность пахотного слоя (0-30 см) наименьшая - 1,27 г/см, так как он имеет рыхлое сложение. Подпахотный слой сильно уплотнен, и его плотность доходит до 1,37 г/см<sup>3</sup>. Пористость почвы составляет в пахотном слое 52,6, в подпахотном - 49,1 % от общего объема, полная влагоемкость - соответственно 46,3 и 45,2 % объема почвы. В рыхлом подпахотном слое наименьшая влагоемкость равна 28,8, в сильно уплотненном пахотном слое - 28% объема почвы. При полной влагоемкости воздухосодержание в пахотном слое почвы равно 6,0% ее объема, в подпахотном - 3,8 % при наименьшей влагоемкости - соответственно 23,6 и 21,8 %.

В орошаемом сероземно - луговомпочвепахотный слой рыхлый, подпахотный - уплотнен. Рыхлое сложение пахотного

слоя увеличивает пористость почвы до 53,0% объема, уплотненность подпахотного уменьшает ее до 48,4%. В рыхлом пахотном слое почвы полная влагоемкость равна 45,6%, наименьшая - 30,8% объема, в подпахотном -соответственно 47,2 и 30,4%. Воздухосодержание при полной влагоемкости сильно уменьшается- в пахотном до 7,5, в подпахотном 1,0% объема, при наименьшей - возрастает соответственно до 21,2% и 18,0%. Гидроморфные условия почвообразования формируют специфичные агрохимические свойства: увеличивается содержание органического вещества, обеспеченность подвижным фосфором низкая, а подвижным калием средняя.

Орошаемая сероземноуговая почва образовалась на слоистых аллювиальных отложениях, а орошаемый типичный и новоорошаемый светлый сероземы на лессах. Орошаемая сероземно - луговая почва развивается под постоянным влиянием близко (2-1,5 м) залегающих грунтовых вод. В ее профиле на глубине 80 см имеется признак оглиения. Механический состав орошаемой сероземно - луговой почвы тяжелосуглинистый, новоорошаемого светлого серозема - легкосуглинистый. Орошаемая сероземно - луговая почва отличается значительно большей влагоемкостью и, наоборот, маленьким воздухосодержанием, в ней больше гумуса, общего азота, и уменьшение их в подпахотном слое. Не такое резкое, как в орошаемом типичном и новоорошаемом светлом сероземах. Наименьшие эти показатели у новоорошаемого светлого серозема.

Орошаемая сероземно - луговая почва и новоорошаемый светлый серозем по содержанию подвижного калия являются среднеобеспеченными, тогда как орошаемый типичный серозем низкообеспеченный.

Таким образом, оценка плодородия почвы показывают, что почвы сильно различаются по глубине залегания грунтовых вод. В почвах с близким залеганием грунтовых вод, кроме оглеения, наблюдается совершенно другой водный, воздушный, температурный режим, иной состав микроорганизмов, другое соотношение аммиачных и нитратных форм азота, фосфорный соединений и калия. Влияние водного режима почвы настолько сильно, что почва, показывающая в первом минимуме азот при одних условиях влажности, в других показывает в первом

минимуме фосфор. Механический состав в значительной степени определяет специфичность почвенных условий. Сорт, выведенный на почвах легкого механического состава у которых отличные водно-воздушный, температурный и пищевой режимы, нельзя рекомендовать для почв тяжелого механического состава, так как сорта, оказавшись в совершенно другой среде, снижают урожайность. Во всех почвенных зонах на более легких почвах поднимается эффективность удобрений, особенно азотистых и калийных. Высокоокультуренные почвы оптимально сочетают в себе факторы и растения, используя их наиболее продуктивно, обеспечивают максимально возможный урожай. Поэтому сорт, выведенный на такой почве, на средне и слабоокультуренной снизит урожай.

### **Заключение**

Таким образом, решены задачи оценки состояния систем, которые располагают неполной и асимметричной информацией о множествах выбора решений. В такого рода конфликтах проблема компромиссов является центральной. Особенно в условиях нечеткого описания предметной области. Развитие теории нечетких множеств обусловлено развитием техники и технологий, требующих новых подходов к решению организационно-технологических задач, процессов принятия решений человеком, логико-лингвистических систем управления. Для решения этих задач потребовались методы, позволяющие использовать элементы естественного языка. Набором таких методов обладает теория нечетких множеств и нейронных сетей. Они позволяют наилучшим образом структурировать все то, что разделено не очень точными границами. С нечеткостью сталкиваются различные специалисты: экономисты, лингвисты, специалисты по теории информации, биологи, экологи, социологи и другие. В связи с тем, что нечеткие классификаторы являются статичными, объединение их с другими интеллектуальными методами, обеспечивающими обучение и адаптацию классификатора к имеющимся данным, как например нейронные сети и генетические алгоритмы, представляет собой современное направление исследований в области интеллектуального анализа данных.

### Список использованной литературы

1. WIPO (2019). WIPO Technology Trends 2019: Artificial Intelligence. Geneva: World Intellectual Property Organization.
2. AI research how AI boosts industry profits and innovation.
3. HolonIQ and source government strategy and policy papers.
4. <https://ec.europa.eu/digital-single-market/en/news/eu-member-states-sign-cooperate-artificial-intelligence>.
5. <https://www.norden.org/en/declaration/ai-nordic-baltic-region>.
6. [https://www.itu.int/dms\\_pub/itu-s/opb/gen/S-GEN-UNACT-2019-1-PDF-E.pdf](https://www.itu.int/dms_pub/itu-s/opb/gen/S-GEN-UNACT-2019-1-PDF-E.pdf).
7. <https://www.thenational.ae/uae/uae-india-artificial-intelligence-bridge-formed-1.754872>.
8. [https://www.diplomatie.gouv.fr/IMG/pdf/declaration\\_franco-canadienne\\_sur\\_l\\_intelligence\\_artificielle\\_cle4a12cb.pdf](https://www.diplomatie.gouv.fr/IMG/pdf/declaration_franco-canadienne_sur_l_intelligence_artificielle_cle4a12cb.pdf).
9. [https://www.g20-insights.org/related\\_literature/g20-japan-ai-principles](https://www.g20-insights.org/related_literature/g20-japan-ai-principles).
10. <https://oecd-opsi.org/wp-content/uploads/2019/11/AI-Report-Online.pdf>.
11. <http://publication.pravo.gov.ru/Document/View/0001201910110003?index=0&rangeSize=1>.
12. <http://fi.chinaembassy.org/eng/kxjs/P020171025789108009001.pdf>
13. <https://niti.gov.in/sites/default/files/2019-01/NationalStrategy-for-AI-Discussion-Paper.pdf>
14. <https://www.kantei.go.jp/jp/singi/tougou-innovation/pdf/aisenryaku2019.pdf>
15. [https://www.orfonline.org/wpcontent/uploads/2018/11/Ai\\_Book.pdf](https://www.orfonline.org/wpcontent/uploads/2018/11/Ai_Book.pdf)
16. OECD.AI (2020), powered by EC/OECD (2020), STIP Compass database, accessed on 21/02/2020, <http://oecd.ai>.
17. <https://www.aisingapore.org/>
18. OECD.AI (2020), powered by EC/OECD (2020), STIP Compass database, accessed on 21/02/2020, <http://oecd.ai>.
19. <https://www.ki-strategie-deutschland.de/home.html>
20. <https://www.whitehouse.gov/wp-content/uploads/2019/06/National-AI-Research-and-Development-Strategic-Plan-2019-Update-June-2019.pdf>

<https://www.oxfordinsights.com/ai-readiness2019>

21. Бостром Н. Искусственный интеллект. Этапы. Угрозы. Стратегии. М.: Манн, Иванов и Фербер, 2016. 496 с.

22. Люгер Д.Ф. Искусственный интеллект. Стратегии и методы решения сложных проблем. М.: Вильямс, 2005. 864 с.

23. Росс А. Индустрии будущего. М.: АСТ, 2017. 288 с.

24. Сигель Э. Просчитать будущее. Кто кликнет, купит, совет или умрет. М.: Альпина Паблишер, 2018. 374 с.

25. Шваб К.М. Четвертая промышленная революция. М.: Эксмо, 2017. 288 с.

26. Denning P.J., Lewis T.G. Exponential Laws of Computing Growth. *Communications of the ACM*, 2017, vol. 60, no. 1, pp. 54–65.

27. Николенко С., Кадурина А., Архангельская Е.В. Глубокое обучение. Погружение в мир нейронных сетей. СПб.: Питер, 2018. 480 с.

28. Искусственный интеллект как стратегический инструмент экономического развития страны и совершенствования ее государственного управления. Ч. 1. Опыт Великобритании и США / И.А. Соколов [и др.] // *International Journal of Open Information Technologies*. 2017. № 9. С. 57–75.

29. Искусственный интеллект как стратегический инструмент экономического развития страны и совершенствования ее государственного управления. Ч. 2. Перспективы применения искусственного интеллекта в России для государственного управления / И.А. Соколов [и др.] // *International Journal of Open Information Technologies*. 2017. № 9. С. 76–101.

30. Цветкова Л.А. Технологии искусственного интеллекта как фактор цифровизации экономики России и мира // *Экономика науки*. 2017. № 2. С. 126–144.

31. Ускова Т.В., Селименков Р.Ю., Чекавинский А.Н. Агропромышленный комплекс региона: состояние, тенденции, перспективы. Вологда: ИСЭРТ РАН, 2013. 136 с.

32. Курцвейл Р., Гроссман Т. Transcend. Девять шагов на пути к вечной жизни. М.: Манн, Иванов и Фербер, 2017. 384 с.

33. Jeffrey S.L. *The Life Plan*. New York Times, 2012. 352 p.

34. Фадеева О. Шанхайский синдром планеты Земля: когда нас будет слишком много и что из этого выйдет // *NakedScience*.

2019. № 41. URL: <https://naked-science.ru/article/nakedscience/shanhayskiy-sindrom-planetu>
35. Форсайт Д.А., ПонсЖ. Компьютерное зрение. Современный подход. М.: Вильямс, 2004. 928 с.
36. Шаныгин С.В. Роботы как средство механизации сельского хозяйства // Изв. высш. учеб. заведений. Машиностроение. 2013. № 3. С. 39–42.
37. Труфляк Е.В. Интеллектуальные технические средства в сельском хозяйстве. Краснодар: КубГАУ, 2016. 42 с.
38. Петришин Л.П. Совершенствование информационной среды развития сельскохозяйственных предприятий // Науковий вісник Львівського національного університету ветеринарно і медицини та біотехнологій імені С.З. Гжицького. 2017. Т. 19. № 81. С. 37–44.
39. Рунов Б.А., Новиков Н.Н. Анализ применения робототехнических средств в сельском хозяйстве // Вестн. ВНИИМЖ. 2017. № 2 (26). С. 113–117.
40. Новые агролесоводственные технологии для сельского хозяйства. <https://issek.hse.ru/trendletter/news/207696712.html>
41. [https://www.texnoman.uz/post/big-dataning-asosiy-8-atamasi\\_.html](https://www.texnoman.uz/post/big-dataning-asosiy-8-atamasi_.html)
42. <https://www.colocationamerica.com/blog/big-data-and-education>
43. Ifeanyi G. Ndukwe, Ben K. Daniel and Russell J. Butson. Data Science Approach for Simulating Educational Data: Towards the Development of Teaching Outcome Model (TOM). *Big Data Cogn. Comput.* 2018, 2(3), 24; <https://doi.org/10.3390/bdcc2030024>
44. <https://precisioncampus.com/blog/benefits-big-data-education/>
45. Ifeanyi G. Ndukwe, Ben K. Daniel and Russell J. Butson. Data Science Approach for Simulating Educational Data: Towards the Development of Teaching Outcome Model (TOM). *Big Data Cogn. Comput.* 2018, 2(3), 24; <https://doi.org/10.3390/bdcc2030024>
46. Baig, M.I., Shuib, L. & Yadegaridehkordi, E. Big data in education: a state of the art, limitations, and future research directions. *Int J Educ Technol High Educ* 17, 44 (2020). <https://doi.org/10.1186/s41239-020-00223-0>

47. Nian-Shing Chen, Chengjiu Yin, Pedro Isaias & Joseph Psotka (2020) Educational big data: extracting meaning from data for smart education, *Interactive Learning Environments*, 28:2, 142-147, DOI: 10.1080/10494820.2019.1635395

48. [https://www.jstor.org/stable/26388386?seq=1#metadata\\_info\\_tab\\_contents](https://www.jstor.org/stable/26388386?seq=1#metadata_info_tab_contents)

49. Aldowah, H., Al-Samarraie, H., & Fauzy, W. M. (2019). Educational data mining and learning analytics for 21st century higher education: A review and synthesis. *Telematics and Informatics*, 37, 13–49.

50. Baker, R., & Siemens, G. (2014). Learning analytics and educational data mining. In: Keith-Sawyer R (ed) *Cambridge handbook of the leaning sciences* (2nd edn). Cambridge University Press: New York, NY, pp. 253–272.

51. Заде Л.А. Понятие лингвистической переменной и его применение к принятию приближенных решений. - М.: Мир, 1976. - 165 с.

52. Алиев Р.А., Алиев Р.Р. Теория интеллектуальных систем и ее применение. - Баку, Изд-во Чашыюглы, 2001. – 720 с.

53. Борисов А.Н., Крумберг О.А., Федоров И.П. Обработка нечеткой информации в системах принятия решений. М.: Радио и связь, 1989. 304 с.

54. Пегат А. Нечеткое моделирование и управление. – М.: БИНОМ. Лаборатория знаний, 2009. – 798 с. (серия «Адаптивные и интеллектуальные системы»).

55. Heckerman, D. (1998) "A Tutorial on Learning with Bayesian Networks," In Jordan, M. (Ed.), *Learning in Graphical Models*, MIT Press.

56. Vapnik, V. and Lerner, A. J., "Generalized portrait method for pattern recognition," *Automation and emote Control*, vol. 24, no. 6, 1963.

57. Марценюк, М.А. Матричное представление нечеткой логики [Текст] / М.А. Марценюк // *Нечеткие системы и мягкие вычисления. Научный журнал Российской ассоциации нечетких систем и мягких вычислений.* –2007. –Т. 2. –№ 3. –С. 7–35.

58. C. J. C. Burges. "A Tutorial on Support Vector Machines for Pattern Recognition" *Knowledge Discovery and Data Mining*, 2(2), 1998.

59. Vityaev E.E., Kostin V.V., Podkolodny N.A., Kolchanov N.A. Natural classification of nucleotide sequences. // Proc. of the Third International Conference On Bioinformatics of Genome Regulation and Structure (BGRS'2002, Novosibirsk, Russia, July 14-20, 2002), v3, ICG, Novosibirsk, 2002, pp. 197-199.

60. Штовба С.Д. "Введение в теорию нечетких множеств и нечеткую логику". <http://www.matlab.exponenta.ru>.

61. Бекмуратов Т.Ф., Мухамедиева Д.Т., Бобомурадов О.Ж. Нечеткая модель прогнозирования урожайности// Научный журнал СО РАН "Проблемы информатики". – Новосибирск, 2010. № 3. – С. 11-23.

62. Bekmuratov T.F., Muhamedieva D.T., Bobomuradov O.J. Model prediction of yield initial conditions. Ninth International Conference on Application of Fuzzy Systems and Soft Computing. ICAFS – 2010. . Edited by R.A. Aliev, K.W. Bonfig, M. Jamshidi, W. Pedrycz, I.B. Turksen. b – Quadrat Verlag. Prague, Czech Republic. August 26-27, 2010. – pp. 321-328.

63. Bekmuratov T.F., Muhamedieva D.T., Bobomuradov O.J. Fuzzy inference system for forecasting problems. IJUCI - International Journal of Ubiquitous Computing and Internationalization. V.3, No. 1, April 2011. ITIRC. - pp. 1-6.

64. Y.J. Oyang, S.C. Hwang, Y.Y. Ou, C.Y. Chen, Z.W. Chen, Data classification with the radial basis function network based on a novel kernel density estimation algorithm, IEEE Transactions on Neural Networks 16 (1) (2005) 225–236.

65. G. Chen, et al . A new approach to classification based on association rule mining. Decis. Support Syst. 2006;42:674-689.

66. Y.C. Hu, Fuzzy integral-based perceptron for two-class pattern classification problems, Information Sciences 177 (2007) 1673–1686.

67. K.G. Srinivasa, K.R. Venugopal, L.M. Patnaik, A self-adaptive migration model genetic algorithm for data mining applications, Information Sciences 177 (2007) 4295–4313.

68. Baker, R. S., & Inventado, P. S. (2014). Educational data mining and learning analytics. In J. A. Larusson & B. White (Eds.), Learning analytics (pp. 61–75). Springer, New York, NY.

69. Bienkowski, M., Feng, M., & Means, B. (2012). Enhancing teaching and learning through educational data mining and learning



analytics: An issue brief. US Department of Education, Office of Educational Technology, 1, 1–57.

70. Lee, LK., Cheung, S.K.S. & Kwok, LF. Learning analytics: current trends and innovative practices. *J. Comput. Educ.* 7, 1–6 (2020). <https://doi.org/10.1007/s40692-020-00155-8>

71. Романов, В.Н. Влияние приемов основной обработки почвы в севообороте на динамику влажности и агрофизические свойства чернозема выщелоченного / В.Н. Романов, В.К. Ивченко, И.О. Ильченко, М.В. Луганцева. // *Достижения науки и техники АПК.* – 2018. – № 5. – С. 32-34.

72. Ниязалиев, Б.И. Влияние органо-минеральных компостов на продуктивность хлопчатника. / Б.И. Ниязалиев // *Аграрная наука.* – 2016. – № 2. – С. 5-6.

73. Hulugalle, N. R. Soil properties under cotton-corn rotations in australian cotton farms / N. R. Hulugalle, B. McCorkell, V. F. Heimoana, L. A. Finlay // *Journal of Cotton Science.* – 2016. – Vol. 20. – Is. 4. – P. 294-298.

74. Naqvi R. Z., Transcriptomics reveals multiple resistance mechanisms against cotton leaf curl disease in a naturally immune cotton species, *Gossypium arboreum* / R. Z. Naqvi, S. S. E. A. Zaidi, K. P. Akhtar, et al. // *Scientific reports.* – 2017. – Vol. 7. – Art. Number 15880.

75. Kazemeini, S. A. Effect of nitrogen and wheat residue on cotton (*Gossypium hirsutum* L.) yield and weed control / S. A. Kazemeini, R. Moradi Talebbeigi, M. Valizade // *Archives of Agronomy and Soil Science.* – 2016. – Vol. 62. – Is. 3. – P. 395-412.

76. Locke, M. A. Conservation management improves runoff water quality: Implications for environmental sustainability in a glyphosate-resistant cotton production system / M. A. Locke, L. J. Krutz, Steinriede R.W., S. Testa // *Soil Science Society of America Journal.* – 2015. – Vol. 79. – Is. 2. – P. 660-671.

77. Zhang, D. M. Lint yield and nitrogen use efficiency of field-grown cotton vary with soil salinity and nitrogen application rate / D. M. Zhang, W. J. Li, C. S. Xin et al. // *Field crops research.* – 2012. – Vol. 138. – P. 63-70.

78. Lofton, J. Utilization of poultry litter, tillage, and cover crops for cotton production on highly degraded soils in northeast

Louisiana / J. Lofton, B. Haggard, D. Fromme, B. Tubana // Journal of Cotton Science. – 2014. – Vol. 18. – Is. 3. – P. 376-384.

79. Pettigrew, W. T. Growth and agronomic performance of cotton when grown in rotation with soybean / W. T. Pettigrew, H. A. Bruns, K. N. Reddy // Journal of Cotton Science. – 2016. – Vol. 20. – Is. 4. – P. 299-308.

80. Zhang H. Root Development of Transplanted Cotton and Simulation of Soil Water Movement under Different Irrigation Methods / H. Zhang, H. Liu, C Sun et al. // Water. – 2017. – Vol. 9. – Is. 7. – Art. Number 503.

81. Yang, G.Z., Tang, H.Y., Nie Y.C. Responses of cotton growth, yield, and biomass to nitrogen split application ratio / G. Z. Yang, H. Y. Tang, Y. C. Nie // European journal of agronomy. 2011. Vol. 35. Is. 3. P. 164-170.

82. Абдыкаримова А. Т., «Технология больших данных», Наука и жизнь Казахстана, №2(78), 2019 г., с.223-226.

83. Egamberdiev N., Mukhamedieva D., Khasanov U. Presentation of preferences in multi-criterional tasks of decision-making//Journal of Physics: Conference Series, 2020, 1441(1), 012137

84. Sotvoldiev D., Muhamediyeva, D.T., Juraev, Z. Deep learning neural networks in fuzzy modeling//Journal of Physics: Conference Series, 2020, 1441(1), 012171

85. Muhamediyeva, D.T. Structure of fuzzy control module with neural network //International Journal of Mechanical and Production Engineering Research and Development, 2018, 9(2), pp. 649–658, IJMPERDAPR201965

86. Muhamediyeva, D.T. Structure of fuzzy control module with neural network//International Journal of Mechanical and Production Engineering Research and Development, 2018, 9(2), pp. 649–658, IJMPERDAPR201965

87. Muhamediyeva, D.T. Fuzzy cultural algorithm for solving optimization problems//Journal of Physics: Conference Series, 2020, 1441(1), 012152