Оглавление

[Общие концепты 4](#_Toc187720066)

[Часть 1. Классический искусственный интеллект 5](#_Toc187720067)

[Основные направления искусственного интеллекта 5](#_Toc187720068)

[Интеллектуальные системы, как основа новых информационных технологий 5](#_Toc187720069)

[Понятие искусственного интеллекта. Определения и пути создания искусственного интеллекта 5](#_Toc187720070)

[Бионическое, эвристическое и эволюционное направления искусственного интеллекта 6](#_Toc187720071)

[Классификация систем искусственного интеллекта 7](#_Toc187720072)

[Основные понятия и определения формализации знаний в интеллектуальных системах 8](#_Toc187720073)

[Проблемная область формализации знаний в интеллектуальных системах 9](#_Toc187720074)

[Данные и знания. Свойства и характеристики знаний 9](#_Toc187720075)

[Процедурные и декларативные знания. Классификация знаний. Формальные языки 10](#_Toc187720076)

[Модели представления знаний. Классификация моделей знаний 12](#_Toc187720077)

[Методы инженерии знаний. Теоретические аспекты получения знаний. Методы извлечения явных и скрытых знаний 12](#_Toc187720078)

[Проблемы структурирования знаний. Проблема сборки знаний в единую модель предметной области 14](#_Toc187720079)

[Логическая модель представления знаний 15](#_Toc187720080)

[Исчисление высказываний 15](#_Toc187720081)

[Основы исчисления предикатов 16](#_Toc187720082)

[Семантические сети. Понятия, события и свойства. Виды семантических сетей 17](#_Toc187720083)

[Семантические сети Куиллиана. Интерпретация семантической сети 18](#_Toc187720084)

[Фреймы. Слоты. Структура фрейма. Указатели наследования и типа данных 19](#_Toc187720085)

[Слоты. Значение слота. Присоединенные процедуры 20](#_Toc187720086)

[Модель, основанная на правилах. Антецедент и консеквент. Продукционная система Поста и ее ограничения 21](#_Toc187720087)

[Марковский и rete-алгоритм. Триплет объект-атрибут-значение. Достоинства и недостатки продукционных систем 23](#_Toc187720088)

[База правил, рабочая память и механизм вывода. Компонента вывода и управляющая компонента. Операции цикла вывода 24](#_Toc187720089)

[Демоны, присоединенные процедуры и механизм наследования в интеллектуальной системе с фреймовой моделью представления знаний. Правила вывода в логических моделях 26](#_Toc187720090)

[Структура и классификация экспертных систем (ЭС), отличие ЭС от других программных продуктов, цикл работы ЭС, технология проектирования и разработки ЭС 26](#_Toc187720091)

[Нечеткая логика. Нечеткие множества. Строгое представление нечетких множеств. Основные характеристики нечетких множеств 29](#_Toc187720092)

[Операции над нечеткими множествами. Логические операции. Алгебраические операции 30](#_Toc187720093)

[Четкое множество «∝-уровня». Нечеткая и лингвистическая переменные. Нечеткие числа 31](#_Toc187720094)

[Часть 2. Вычислительный интеллект 33](#_Toc187720095)

[Особенности человеческого восприятия информации и современных систем управления. Вычислительный интеллект (ВИ). Методы ВИ 33](#_Toc187720096)

[Искусственные нейронные сети. Основные проблемы, решаемые искусственными нейронными сетями 34](#_Toc187720097)

[Биологический нейрон. Искусственный нейрон. Активационная функция. Классификация и свойства искусственных нейронных сетей. Обучение искусственных нейронных сетей 35](#_Toc187720098)

[Теорема Колмогорова. Алгоритм обучения персептрона. Линейная разделимость и персептронная представляемость 36](#_Toc187720099)

[Нейронная сеть обратного распространения. Алгоритм обучения сети обратного распространения 38](#_Toc187720100)

[Нейронная сеть встречного распространения (сеть Кохонена). Алгоритм обучения сети Кохонена 39](#_Toc187720101)

[Двухслойная сеть встречного распространения и ее обучение. Входные и выходные звезды Гроссберга и их обучение 40](#_Toc187720102)

[Двухслойная сеть встречного распространения. Алгоритм обучения сети встречного распространения 42](#_Toc187720103)

[Стохастические методы обучения. Обучение Больцмана. Алгоритм обучения Больцмана. Обучение Коши 43](#_Toc187720104)

[Сети с обратными связями. Сеть Хопфилда 44](#_Toc187720105)

[Правило обучения Хебба. Ортогонализация образов 45](#_Toc187720106)

[Двунаправленная ассоциативная память 46](#_Toc187720107)

[Адаптивная резонансная теория 47](#_Toc187720108)

[Методы оптимизации комбинаторных задач различной степени сложности. Генетические алгоритмы 48](#_Toc187720109)

[Базовый генетический алгоритм. Последовательные модификации базового генетического алгоритма 49](#_Toc187720110)

[Параллельные модификации базового генетического алгоритма. Классификация генетических алгоритмов 50](#_Toc187720111)

[Когнитивная компьютерная графика. Определение. Степень организованности информации 51](#_Toc187720112)

[Задача сжатия информации. Анаморфозы (определение) 52](#_Toc187720113)

[Метод анаморфирования 52](#_Toc187720114)

[Проблемы реализации анаморфоз 52](#_Toc187720115)

[Численные методы построения анаморфоз 53](#_Toc187720116)

[Основные законы гибридного интеллекта 54](#_Toc187720117)

[Основные методы гибридизации 55](#_Toc187720118)

[Общий подход к построению гибридной интеллектуальной системы 57](#_Toc187720119)

[Принципы построения гибридных интеллектуальных систем 57](#_Toc187720120)

# Общие концепты

1. Общепризнанного **определения** **знания**, как и **определения** **ИИ**, не существует.
2. **Признаки ИС**: данные -> знания; алгоритмы -> методы ИИ.

# Часть 1. Классический искусственный интеллект

# Основные направления искусственного интеллекта

Модуль 1 - Лекция 2, раздел ""​

​ Искусственный интеллект включает в себя множество направлений, таких как:

* Экспертные системы
* Нейронные сети
* Системы обработки естественного языка
* Эволюционные методы
* Генетические алгоритмы
* Нечеткие множества
* Системы извлечения знаний

Каждое направление ориентировано на решение задач в специфических областях: от автоматического анализа и прогнозирования до сложных систем управления​

# Интеллектуальные системы, как основа новых информационных технологий

Модуль 1 - Лекция 2, весь текст​

**Интеллектуальные системы (ИС)** – это **аппаратно-программные комплексы**, способные обрабатывать информацию и решать задачи, которые традиционно требовали человеческого интеллекта.

Они применяются в диагностике, прогнозировании, управлении и других сферах. **Ключевая черта ИС** – способность работать с **неформализованными задачами**, требующими эвристических методов. Основные направления развития ИС включают в себя работу с большими объемами знаний и их адаптацию к конкретным условиям.

# Понятие искусственного интеллекта. Определения и пути создания искусственного интеллекта

Модуль 1 - Лекция 2, раздел "Понятие ИИ"​

**Интеллект** – способность любого организма (или устройства) достигать некоторой измеримой степени успеха при поиске одной из многих возможных целей в обширном многообразии сред.

**Искусственный интеллект** – наука и технология создания машин, имитирующих человеческий интеллект. Это включает понимание, обучение, адаптацию и вывод.

**В узком смысле** – программная система, имитирующая на компьютере мышление человека.

**Искусственный интеллект** – это возможность решения задач, которые до сих пор не удавалось решить человеку, машинным способом с помощью программных средств.

***В теории ИИ*** *можно выделить* ***два основных направления*** *их исследования:* ***общетеоретическое****, связанное с разработкой моделей и методов реализации отдельных аспектов получения и преобразования знаний и* ***прикладное****, нацеленное на разработку комплексных технологий конечных пользователей.*

Пути создания ИИ разделяются на бионическое моделирование мозга, эвристическое программирование и эволюционные подходы. Основные проблемы включают формализацию знаний и разработку методов обработки данных.

# Бионическое, эвристическое и эволюционное направления искусственного интеллекта

Модуль 1 - Лекция 2, раздел "Бионическое, эвристическое и эволюционное направления"​

Три основных подхода к созданию ИИ:

1. **Бионическое моделирование** стремится воспроизвести структуру мозга на уровне нейронных сетей.
2. **Эвристическое программирование** направлено на создание алгоритмов, использующих правила и эвристики для сокращения объема поиска решений.
3. **Эволюционные методы** основаны на принципах естественного отбора, применяемых для адаптации систем.

**Бионическое моделирование.**

*Непосредственное моделирование человеческого мозга (т.е. моделирование каждой нервной клетки и связей между ними) с целью создания автоматов, обладающих интеллектом.*

**Эвристическое программирование.**

*Творческое решение задач.*

*Радикальное уменьшение вариантов, необходимых при использовании метода проб и ошибок.*

*Всегда существует вероятность упустить наилучшее решение.*

*Два метода: метод анализа целей и средств и метод планирования (первый – выбор и осуществление таких операций, которые последовательно уменьшают разницу между исходным и конечным состоянием задачи; второй – вырабатывается упрощенная формулировка исходной задачи, которая также решается методом анализа целей и средств.*

**Эволюционное моделирование.**

*Попытка смоделировать не то, что есть, а то, что могло бы быть, если бы эволюционный процесс направлялся в нужном направлении и оценивался предложенными критериями.*

*Высвобождает время для проблемы выбора целей и выявления параметров среды, которые заслуживают исследования.*

# Классификация систем искусственного интеллекта

Модуль 1 - Лекция 2, раздел "Классификация систем ИИ"​

Системы ИИ классифицируются по интеллектуальным признакам:

* **Коммуникативные способности** (системы с интеллектуальным интерфейсом)
  + Интеллектуальные базы данных
  + Естественно-языковые интерфейсы
  + Гипертекстовые системы
  + Системы контекстной помощи
  + Когнитивная графика
* Для **сложно формализуемых задач** (экспертные системы)
  + Классифицирующие
  + Доопределяющие
  + Трансформирующие
  + Мультиагентные
* Способность к **самообучению** (самообучающиеся системы)
  + Индуктивные
  + Нейросети
  + Системы, основанные на прецедентах
  + Информационные хранилища
* **Адаптивные информационные системы**
  + CASE-технологии
  + Компонентные технологии

Изображение выглядит как текст, снимок экрана, Шрифт, черно-белый

Автоматически созданное описание

# Основные понятия и определения формализации знаний в интеллектуальных системах

Модуль 1 - Лекция 3, раздел "Основные понятия и определения"​

*Признаки ИС: данные -> знания; алгоритмы -> методы ИИ.*

**Формализация знаний** – это процесс приведения знаний, поступающих из разных источников, к стандартной форме для их структурирования и использования. Основные понятия:

* **Данные** – это необработанная информация, которая преобразуется в знания при структурировании.
* **Знания** – это формализованная информация, необходимая для решения задач; Совокупность сведений о сущностях (объектах, предметах) реального мира, их свойствах и отношениях между ними в определенной *предметной области*.
* **Знания** – это выявленные закономерности предметной области (принципы, связи, законы), позволяющие решать задачи в этой области.
* **Предметная Область** – область человеческих знаний, в терминах которой формулируются задачи и в рамках которой они решаются. Т.е. ПрО – описание части реального мира, которое в силу своей приближенности рассматривается как её информационная модель.
* **Проблемная область** – это содержательное описание в терминах ПрО проблемы совместно с комплексом условий, факторов и обстоятельств, вызвавших ее возникновение.

Формализация позволяет создать **базы знаний**, используемые в ИС для вывода решений​.

# Проблемная область формализации знаний в интеллектуальных системах

Модуль 1 - Лекция 3, раздел "Проблемная область"​

**Проблемная область** – это часть реального мира, формализуемая для решения задач. Она описывает объекты, их свойства и взаимосвязи, структурированные для обработки в интеллектуальной системе (содержательное описание в терминах ПрО проблемы совместно с комплексом условий, факторов и обстоятельств, вызвавших ее возникновение).

Формализация знаний в ПрО помогает создать точные модели, обеспечивающие эффективное решение задач​.

В полном объеме информация, содержащаяся в знаниях, должна включать сведения о (иногда говорят, что это составляет проблемную область решаемой задачи):

* системе понятий предметной области, в которой решаются задачи;
* системе понятий формальных моделей, на основе которых решаются задачи;
* соответствии систем понятий, упомянутых выше;
* методах решения задачи;
* текущем состоянии предметной области.

# Данные и знания. Свойства и характеристики знаний

Модуль 1 - Лекция 3, разделы "Данные и знания" и "Свойства, характеристики знаний"​

**Данные** – это факты и идеи, представленные в формализованной форме.

**Знания** – это структурированная информация (данные), которая интерпретируется для решения задач.

***Свойства знаний:*** именование, структура, взаимосвязи, шкалирование, схожесть, активность *(аналогично class ООП)*.

***(Формально)*** Знания обладают свойствами, такими как:

* **Внутренняя интерпретируемость** – наличие в памяти ЭВМ сведений не только о значении, но и о наименовании информационной единицы.
* **Внутренняя (рекурсивная) структурированность** – вложенность одних информационных единиц в другие или в самих себя. Она предусматривает установку отношений принадлежности элементов к классу, родовидовые отношения типа «часть-целое» и т.п. В целом внутренняя структурированность характеризует структуру знания.
* **Внешняя взаимосвязь единиц** – с какой информационной единицей имеет связь данная информационная единица и какова эта связь. *Отдельные информационные единицы не могут описывать динамические ситуации (ситуативная связь фактов или явлений разных ИнфЕд).*
* **Шкалирование** – использование шкал, предназначенных для фиксации соотношения различных величин. Прежде всего шкалирование необходимо для фиксации соотношений качественной информации.
* **Погружение в пространство с семантической метрикой** используется для задания меры близости информационных единиц (матрицы сходства).
* **Активность** – выражается в возможности вызова той или иной процедуры в зависимости от структуры, сложившейся между информационными единицами.

***Базы данных*** *фиксируют* ***экстенсиональную*** *семантику заданной проблемной области, состояние конкретных объектов, конкретные значения параметров для определенных моментов времени и временных интервалов.* ***База знаний*** *определяет* ***интенсиональную*** *семантику моделей и содержит описание абстрактных сущностей: объектов, отношений, процессов.*

Знания можно разделить на 2 группы **(характеристики)**: факты (общепризнанные) / эвристика (знания эксперта).

# Процедурные и декларативные знания. Классификация знаний. Формальные языки

Модуль 1 - Лекция 3, раздел "Процедурные и декларативные знания"​

**Декларативные знания** представляют собой утверждения и факты о предметной области. *Совокупность сведений о качественных и количественных характеристиках конкретных объектов, явлений и их элементов, представленных в виде фактов и эвристик.*

**Процедурные знания** описывают действия для достижения целей (например, алгоритмы). Хранятся в памяти ИИС в виде описаний процедур, с помощью которых их можно получить. *Процедурные знания – это методы, алгоритмы, программы решения различных задач, последовательности действий (в выбранной проблемной области);* ***составляют ядро баз знаний.***

**Знания + Вывод = Система**

**Формальные языки** – языки, характеризующийся **точными правилами** построения выражений и их понимания. Он строится в соответствии с четкими правилами, обеспечивая непротиворечивое, точное и компактное отображение свойств и отношений изучаемой предметной области (моделируемых объектов). **Состоят из: алфавита и синтаксиса** (правила построения выражений). Формальные языки, такие как логика, используются для представления и обработки знаний. Их применение позволяет создать унифицированные методы вывода и работы с данными​.

Изображение выглядит как текст, диаграмма, Шрифт, линия

Автоматически созданное описание

Изображение выглядит как текст, снимок экрана, Шрифт, линия

Автоматически созданное описание

# Модели представления знаний. Классификация моделей знаний

Модуль 1 - Лекция 3, разделы "Модели представления знаний" и "Классификация моделей знаний"​

**Представление знаний (явление)** – формализация и структурирование, с помощью которых отражаются характерные признаки знаний.

**Представление знаний (процесс)** – процесс (способ) описания знаний человека о проблемной области посредством выражений на формальном языке, называемом языком представления знаний.

**Модель представления знаний (МПЗ)** – способ и результат **формального описания** знаний в БЗ. Она должна быть понятной пользователю и обеспечивать однородность представления знаний.

В основе использования МПЗ лежит **аксиоматический метод**. Аксиоматический метод: выделяется некоторое небольшое множество истинных утверждений, опираясь на которые можно вывести все истинные утверждения данной науки.

Классификация МПЗ:

* **Логические модели** обеспечивают строгий вывод на основе правил формальной логики.
* **Продукционные модели** используют правила вида "если..., то...".
* **Сетевые модели (семантические сети)** представляют знания в виде графов, отражающих связи между объектами.
* **Фреймовые модели** структурируют знания в иерархической форме, объединяя данные и процедуры​

# Методы инженерии знаний. Теоретические аспекты получения знаний. Методы извлечения явных и скрытых знаний

Модуль 1 - Лекция 4, раздел "Методы инженерии знаний"​

**Инженерия знаний** – совокупность процессов и методов для **выявления** (получения), **структурирования** и **представления** знаний.

Изображение выглядит как текст, диаграмма, снимок экрана, Шрифт

Автоматически созданное описание

Теоретические аспекты инженерии. Аспекты проблематики: психологический, лингвистический и методологический (гносеологический).

**Гносеологический аспект** (методологические проблемы) – эксперт часто **впервые формулирует** некоторые закономерности, до того момента составлявшие его личный опыт.

**Лингвистический аспект** касается исследования языковых проблем, так как язык – это основное средство общения в процессе извлечения знаний.

**Психологический аспект** – самый проблемный в процессе общения инженера и эксперта. Структурные компоненты: участники общения (партнеры), средства общения (процедура), предмет общения (знания).

Три «слоя» психологических проблем, возникающих при извлечении знаний:

* контактный (проблемы взаимодействия партнеров);
  + демографический (пол, возраст, национальность, образование);
  + личностный (особенности личностного портрета, темперамент, мотивация).
* процедурный (проблемы грамотного проведения процедуры извлечения);
  + ситуация (место, время, продолжительность);
  + оборудование (вспомогательные средства, освещенность, мебель);
  + профессиональные приемы (темп и громкость речи, мимика, поза, интонации)
* когнитивный (проблема получения знания как такового)

**Извлечение явные знаний:** коммуникативные (общение с экспертом) + текстологические (документы). Коммуникативные: пассивные (ведущий эксперт) + активные (ведущий инженер).

Пассивные – наблюдение, лекции, мысли вслух.

Активные: групповые (мозговой штурм, ролевые игры, круглый стол) + индивидуальные (интервью, экспертные игры, анкетирование).

**Извлечение скрытых знаний:** методами психосемантики для выявления неосознаваемых знаний (латентные, скрытые, имплицитные) на стыке когнитивной психологии, психолингвистики, психологии восприятия и исследования индивидуального сознания. Основные методы:

**Метод многомерного шкалирования** – анкеты со шкалированием по свойствам объекта. Статистика по выборке – выявление новых проекций пространств для выявления свойств.

**Метафорический подход** ориентирован на выявление скрытых составляющих практического опыта эксперта и основан на сравнении объектов предметной области с абстрактными объектами из мира метафор.

**Метод репертуарных решеток** – выявление личностных свойств, которые проявляются через систему личностных конструктов. Этот метод может применяться и для извлечения знаний. Репертуарная решетка – матрица, столбцы соответствуют определенным группам объектов (элементов), в качестве которых могут выступать люди, предметы, понятия, отношения, звуки и др. Строки матрицы соответствуют конструктам, которые представляют собой биполярные признаки, параметры, шкалы, отношения или способы поведения.

# Проблемы структурирования знаний. Проблема сборки знаний в единую модель предметной области

Модуль 1 - Лекция 4, разделы "Проблемы структурирования знаний" и "Проблема сборки знаний"​

**Проблема:** объём и содержание знаний зависит от автора и целей. Решение: выбрать объекты ПрО и определить их связи.

**Методы структурирования знаний:** структурный подход (алгоритмическая декомпозиция; модули, каждый модуль – отдельный этап), объектный (ООП, декомпозиция типа объект = экземпляр класса), структурно-объектный (последовательное изучение + разложение по ролевым стратам).

Структурирование знаний предполагает создание моделей, которые отражают связи и взаимодействия объектов. Проблемы включают:

* Разнообразие источников знаний.
* Противоречивость данных.
* Необходимость интеграции разнородных данных в единую модель.  
  Эффективная сборка знаний требует согласования и создания унифицированной структуры​.

При создании единой системы следует учитывать, что, скорее всего, они собирались не одним лишь когнитологом, а командой (психолог, игролог-математик, методологист и тд), то есть неизменно встаёт вопрос – согласование полученных знаний и их структуризация. Требуется единый метод/концепция, которая красной нитью свяжет различные методологии.

# Логическая модель представления знаний

Модуль 1 - Лекция 5, раздел "Логическая модель представления знаний"​

Логические модели представляют знания в виде фактов и утверждений, формализованных с помощью логических формул.

Достоинства:

* Строгость и доказуемость.
* Возможность автоматического вывода на основе правил логики.

Недостатки:

* Ограниченная применимость для слабоструктурированных задач

Основная идея: вся ПрО – совокупность фактов и утверждений в оболочке из формул. 𝑆 = <𝐵, 𝐹, 𝐴, 𝑅>, где:

* 𝐵 – счетное множество базовых символов (алфавит);
* 𝐹 – множество (перечислимое) формул (называемых также правильно построенными формулами), построенных из элементов 𝐵 с использованием некоторого набора синтаксических правил;
* 𝐴 – выделенное подмножество априори истинных формул (аксиом);
* 𝑅 – конечное множество отношений между формулами, называемое правилами вывода.

# Исчисление высказываний

Модуль 1 - Лекция 5, раздел "Исчисление высказываний"​

**Исчисление высказываний** – это система формализации знаний с использованием логических переменных и операций (И, ИЛИ, НЕ).  
Оно позволяет составлять сложные логические формулы, строить таблицы истинности и проводить дедуктивные выводы. Этот метод применяется для анализа и проверки истинности высказываний.

**Предикат**, в логике — понятие, определяющее предмет суждения (субъект) и раскрывающее его содержание. Исчисление высказываний – часть логики предикатов.

**Высказывание** – некоторое предложение, смысл которого можно выразить значениями ИСТИНА (TRUE) или ЛОЖЬ (FALSE). **Элементарные высказывания** нельзя разделить на части. Элементарные высказывания рассматриваются как переменные логического типа.

Алфавит **исчисления высказывания**:

* Высказывания / *логические переменные* *Р, Q, R, S, ...,*
* Логические константы / *TRUE (ИСТИНА) и FALSE (ЛОЖЬ),*
* Логические операции И, ИЛИ, НЕ, эквивалент, импликация, ( )  
  *⋀ («и», конъюнкция), ⋁ («или», дизъюнкция), ¬ («не», отрицание), ≡ («тогда и только тогда, когда», эквиваленция), →(«следует», импликация) и круглые скобки.*

Логическая формула – выражение из высказываний + операций. Логическая переменная и/или константа – тоже формулы. В формуле есть либо другие формулы, либо связки (операции).

*В принципе всё по базовым фичам алгебры логики.*

# Основы исчисления предикатов

Модуль 1 - Лекция 5, раздел "Основы исчисления предикатов"​

*Исчисление предикатов расширяет возможности исчисления высказываний, позволяя описывать внутреннюю структуру объектов.  
Оно использует предикаты (отношения) и кванторы (например, "все" и "существует") для формализации знаний и выполнения логического вывода в сложных система*

Исчисление высказываний – очень грубая аппроксимация, так как нету высказывание – единой целое (без структуры). Силлогизм – умозаключение, где из посылок следуют выводы.

**Предикат** – в отличие от высказывание разделяет факт на взаимодействие, отношение или свойство. Типа: «Таня сестра Маши» – предикат «СЕСТРЫ (Таня, Маша)». Формальная структура: «Предикатный Символ (терм1, терм2)» – всё вместе **предикатная формула (предикат / атом)**. Предикаты бывают одноместные / двухместные / трехместные / тд по количество термов.

**Исчисление предикатов** – аксиоматическая система, построенная согласно формальной теории F = (A, V, W, R), где

* A – алфавит,
* V – синтаксис (синтаксические правила),
* W – базовые аксиомы,
* R – правила для вывода

**Алфавит ИП** содержит: индивидные константы, предметные переменные, функциональные константы, высказывания, предикатные константы. Алфавит, как и для ИВ, содержит: ∧, ∨, →, ≡, НЕ – а также расширен кванторами: ∀ (свёртка И на всём множестве) и ∃ (свёртка ИЛИ на всём множестве).

**Синтаксис ИП** аналогичен синтаксису ИВ. Правильно построенные формулы (ППФ) в рамках ИВ остаются ППФ в рамках ИП, но должны быть представлены в соответствующей предикатной форме. Добавляются правила:

1. Атом – есть ППФ
2. Если P(x) – ППФ, а x – переменная, то ∀x P(x) и ∃x P(x) – тоже ППФ
3. Если 𝑃(𝑥,𝑦) – ППФ и х, у – переменные и f – функция, то  
   ∀𝑥∀𝑦 𝑃(𝑥, 𝑓(𝑦))и ∃𝑥∃𝑦 𝑃(𝑥, 𝑓(𝑦)) – ППФ. Последние формулы верны и для смешанных кванторов.

**Базовые аксиомы ИП** могут быть переняты из системы аксиом ИВ, но должны быть дополнены с учётом кванторов ЛЮБОЙ и СУЩЕСТВУЕТ.

*(𝐴4) ∀𝑥𝑃(𝑥) → 𝑃(𝑦),*

*(А5) 𝑃(𝑦) → ∃𝑥𝑃(𝑥).*

Правила вывода ИП те же, что в ИВ – правило подстановки, правила заключения. Дополнено правилом, учитывающим кванторы.

# Семантические сети. Понятия, события и свойства. Виды семантических сетей

Модуль 1 - Лекция 6, раздел "Семантические сети"​

**Семантические сети** – это графы, где вершины представляют объекты или понятия ПрО, а ребра – связи между ними.

Используются три основных типа объектов: понятия (объекты ПрО), события (действия) и свойства (характеристики объектов).

Виды семантических сетей:

* Ситуационные (причинно-следственные связи),
* Целевые сети (используются в системах планирования, «цель-подцель», «цель-средство»),
* Классификационные («род-вил», «класс-подкласс»),
* Функциональные («аргумент-функция»)

Они используются для представления знаний в системах обработки естественного языка и экспертных системах​. Особенность семантической сети как модели представления знаний, которая может одновременно считаться и ее достоинством, и ее недостатком, заключается в невозможности в явном виде разделить БЗ и механизм логического вывода. Поэтому интерпретация семантической сети осуществляется только с помощью использующих ее процедур.

# Семантические сети Куиллиана. Интерпретация семантической сети

Модуль 1 - Лекция 6, раздел "Семантические сети Куиллиана"​

При построении семантической сети отсутствуют ограничения на число связей и сложность сети. Семантическая сеть должна быть систематизирована для того, чтобы формализация оказалась возможной.

**Семантические сети TLC (Куиллиана)** ***систематизируют функции*** отношений между понятиями с помощью следующих признаков:

* множество-подмножество (типы отношений «целое-часть»; род-вид» и т. д.);
* отношений «близости»;
* отношений «сходства-различия»;
* логических связей (И, ИЛИ, НЕ);
* количественных связей (больше, меньше, равно, ...);
* пространственных связей (далеко от, за, над, ...);
* временных связей (раньше, позже, в течение, ...);
* атрибутных связей (иметь свойство, иметь значение);
* лингвистических связей и др.

Основной идеей семантической сети TLC было описание значений класса, к которому принадлежит объект, его прототипа и установление связи со словами, отображающими свойства объекта.

Изображение выглядит как текст, снимок экрана, диаграмма, Шрифт

Автоматически созданное описание

# Фреймы. Слоты. Структура фрейма. Указатели наследования и типа данных

Модуль 1 - Лекция 6, раздел "Фреймы"​

**Фреймы** – это структуры, описывающие объекты и события через слоты (атрибуты). Ключевые элементы: **слоты** (содержат значения или ссылки на другие фреймы), **указатели наследования** (позволяют использовать свойства родительских фреймов), **фреймы** (удобны для представления знаний в системах, где важна иерархия и связи между объектами).

**Фрейм** – это абстрактный образ для представления некоего стереотипа восприятия (объекта, понятия или ситуации) путем сопоставления факта с конкретными элементами и значениями в рамках, определенных для объекта, в структуре БЗ. Под абстрактным образом понимается некоторая обобщенная и упрощенная модель или структура. Фреймом также называется и формализованная модель для отображения образа.

**Фреймовая МПЗ** базируется так же на понятии функционального программирования – способа составления программ, в которых единственным действием является вызов функции, единственным способом расчленения программ на части является введение имени для функции и задание для этого имени выражения, вычисляющего значение функции, а единственным правилом композиции – оператор суперпозиции других функций.

*Короче, простыми словами: объект = функция. Функция может вызывать функцию / возвращать вычисленное значение.*

Фреймы быают:

* фреймы-образцы, или прототипы
* фреймы-экземпляры
* фреймы-структуры, которые применяются для обозначения объектов и понятий (заем, залог, вексель);
* фреймы-роли (менеджер, кассир, клиент);
* фреймы-сценарии (банкротство, собрание акционеров, и т. д.);
* фреймы-ситуации (тревога, авария, рабочий режим устройства и др.).

Структура фрейма: Имя Фрейма ( [ Имя Слота / Значение / Тип ] ) (см след. вопрос *слоты*).

Указатели наследования – позволяют наследовать какие-либо слоты из родительских фреймов (посредством одинакового имени и указателей на родителя AKO). Типы наследования: U (Unique) – не наследовать, S (Same) – наследовать, R (Range) – текущее значение попадает в диапазон родителя, O (Override) – брать родительский, если данный пуст.

Указатель типа данных показывает тип значения слота. Наиболее употребляемые типы: frame – указатель фрейма; real, integer, Boolean, text, table, expression, lisp - вещественный, целый, логический текстовый тип, таблица, выражение, связанная процедура и т. д.

Значение слота должно соответствовать указанному типу данных и условию наследования. Существует несколько способов получения слотом значений во фрейме-экземпляре:

* по умолчанию от фрейма-образца (значение по умолчанию);
* через наследование свойств от фрейма, указанного в слоте АКО;
* по формуле, указанной в слоте;
* через присоединенную процедуру;
* явно из диалога с пользователем;
* из базы данных.

# Слоты. Значение слота. Присоединенные процедуры

Модуль 1 - Лекция 6, раздел "Слоты"​

*Слот – это атрибут фрейма, который может содержать:*

* *Конкретное значение.*
* *Ссылку на другой фрейм.*
* *Присоединенные процедуры, активируемые при доступе к слоту.*

*Это позволяет моделировать сложные объекты и их поведение в динамической среде​.*

Фрейм – структура, состоящая из: **имени фрейма**, **слотов** (имена и значения) и **присоединенных процедур**, связанных с фреймом или со слотами. Основные структурные элементы фрейма определяются с помощью слотов. Текущие значения слотов помещаются в **шпации**. Число слотов в каждом фрейме устанавливается проектировщиком системы, при этом **часть слотов определяется самой системой** для выполнения специфических функций (системные слоты), примерами которых являются: **слот-указатель родителя данного фрейма** (АКО), **слот-указатель дочерних фреймов**, слот для определения даты создания фрейма и т. д

**Слот имеет уникальное имя** (в пределах фрейма); в качестве имени слота может выступать любой произвольный текст. Либо имя процедуры для вычисления значения. Слоты могут содержать сложные структуры данных: массивы, списки, множества и тд.

Во фреймах различают **два типа присоединенных процедур**: процедуры-демоны и процедуры-слуги. Демоны – автоматически запускаются при обращении к фрейму (IF-NEEDED – если значений было пустым; IF-ADDED – при попытке изменения слота; IF-REMOVED – при попытке удаления слота).

В качестве значения слота: непосредственно значение, процедура («служебная» в LISP / «метод» в ООП).

# Модель, основанная на правилах. Антецедент и консеквент. Продукционная система Поста и ее ограничения

Модуль 1 - Лекция 7, раздел "Продукционная модель знаний"​

*Продукционная модель представляет знания в виде правил "если... то...", где:*

* *Антецедент – условия применения правила.*
* *Консеквент – действие, выполняемое при выполнении условий.*

*Продукционная система Поста, основанная на таких правилах, ограничена простотой формализации и невозможностью работы с неопределенностью*.

Термин «продукция» был введен американским логиком Е. Постом в 40-х годах в работах по обоснованию и формализации алгоритмических систем. Продукцией называется правило вида ∝→ 𝛽, где ∝, 𝛽 - слова над некоторым алфавитом А. Совокупность правил типа ∝→ 𝛽 получила название системы продукций. В записях формально логики: 𝑃(𝑥, 𝑦) → 𝐴 & 𝐷. Здесь 𝑃(𝑥, 𝑦) – логическое выражение; 𝐴 - заключение; 𝐷 - действие, то есть продукция – это пара <антецедент, консеквент>, а «→» - знак секвенции.

В наиболее общем виде под продукцией понимается выражение:

< (𝑖), 𝑄, 𝑁, 𝑃 → 𝐴, 𝐷 >

𝑖 – уникальное имя продукции,

𝑄 – характеризует сферу применения данной продукции

𝑁 – условие применимости ядра продукции (активация, если ИСТИНА)

𝑃 → 𝐴 - называется ядром продукции. Это основной элемент. Допускается форма: если 𝑃, то 𝐴, иначе 𝐵.

Элемент 𝐷 описывает постусловия продукции. Они активизируются лишь в случае, если ядро продукции реализовалось. Постусловия описывают действия и процедуры, которые необходимо выполнить после реализации ядра.

Ядра продукции можно классифицировать по различным основаниям. Прежде всего все ядро делится на два больших типа: детерминированные (точно определенные) и недетерминированные (вероятностные). Детерминированные – если P, то A; вероятностные – если P, то  
с вероятностью *p* А (см. *нечёткая логика*).

Детерминированные продукции могут быть однозначными и альтернативными (связка «или»). Здесь тоже нужен свой механизм выбора альтернативы из нескольких возможных. Это может быть, например, механизм присваивания весов, вероятностных или экспертных оценок.

Продуктивная система формально может быть представлена как:

ПС = <РБД, БЗ, МВ>

Значения переменных обычно задаются в так называемой **рабочей базе данных – РБД** (рабочая память) в виде набора конкретных фактов, формат которых согласуется с форматом правил. Совокупность правил-продукций является сердцем системы и называется **базой знаний или правил (БЗ)**. Механизм интерпретации правил как раз и является **механизмом вывода (MB)**.

В каждом цикле интерпретации выполняются следующие процессы:

* определяется множество означиваний, то есть выделяется набор продукций БЗ и фактов из РБД, согласующихся между собой;
* выполняются сами означивания (сопоставления), то есть реализация указаний из правой части продукций.

В продукционной модели легко просматриваются аналогии: РБД – наборы фактов, описывающих текущие состояния предметной области; БЗ – модель предметной области; MB - модель механизма принятия решений (вывода) - прямой аналог человеческим рассуждениям.

# Марковский и rete-алгоритм. Триплет объект-атрибут-значение. Достоинства и недостатки продукционных систем

Модуль 1 - Лекция 7, раздел "Механизм вывода в продукционной системе знаний"​

* *Марковский алгоритм последовательно применяет правила для поиска решения.*
* *Rete-алгоритм оптимизирует поиск, анализируя только изменившиеся данные.*
* *Триплет объект-атрибут-значение используется для структурирования знаний.*

*Достоинства: простота реализации. Недостатки: слабая масштабируемость*

Марковский алгоритм: последовательно применяет правило вывода к каждому факту (если возможно) и переходит к следующему правилу.

Rete-алгоритм строит предварительный граф, где узлы – правила. Путь от корня до листа образует полное условие некоторой продукции, каждый узел хранит список фактов, соответствующих условию. При добавлении или модификации факта он прогоняется по сети, при этом отмечаются узлы, условиям которых данный факт соответствует. При выполнении полного условия правила, когда система достигает листа графа, правило выполняется.

Антецеденты и консеквенты правил формируются из атрибу­тов и значений («если **А(**стартер**)** **З(**не работает**)**, то **А(**двигатель**) З(**не заводится**)**»). Более эффективной моделью является модель триплетов объект-атрибут-значение («если **О(**в машине**)** **А(**стартер**)** **З(**не работает**)**, то **А(**двигатель**) О(**машины**) З(**не заведётся**)**» *очень условно, но типа так*). С введением триплетов правила из базы правил могут срабаты­вать более одного раза в процессе одного логического вывода, по­скольку одно правило может применяться к различным экземп­лярам объекта (но не более одного раза к каждому экземпляру).

Плюсы ПС:

1. Подавляющая часть человеческих знаний может быть записана в виде продукций
2. Системы продукций являются модульными. За небольшим исключением удаление и добавление продукций не приводит к изменениям в остальных продукциях.
3. При необходимости системы продукций могут реализовывать любые алгоритмы и, следовательно, способны отражать любое процедурное знание, доступное ЭВМ.
4. Наличие в продукциях указателей на сферу применений продукции позволяет наиболее эффективно организовать память, сократив время поиска в ней необходимой информации. Классификация сфер может быть многоуровневой, что еще более повышает эффективность поиска знаний, так как позволяет наследовать информацию в базе знаний.
5. При объединении систем продукций и сетевых представлений получаются средства, обладающие большой вычислительной мощностью.
6. Естественный параллелизм в системе продукций, асинхронность их реализации делают продукционные системы удобной моделью вычислений для ЭВМ новой архитектуры, в которой идея параллельности и асинхронности является центральной.

Недостатки:

1. При большом числе продукций становится сложной проверка непротиворечивости системы продукций. Это заставляет при добавлении новых продукций тратить много времени на проверку непротиворечивости новой системы.
2. Из-за присущей системе недетерминированности (неоднозначного выбора выполняемой продукции из множества активизированных продукций) возникают принципиальные трудности при проверке корректности работы системы.

# База правил, рабочая память и механизм вывода. Компонента вывода и управляющая компонента. Операции цикла вывода

Модуль 2 - Лекция 8, раздел "База правил и механизм вывода"​

*База правил содержит знания, представленные в виде продукций.  
Рабочая память – динамическое хранилище данных, используемое для вывода решений.  
Механизм вывода сопоставляет правила с данными из рабочей памяти, определяя, какие действия нужно выполнить. Управляющая компонента контролирует ход процесса вывода​.*

**База знаний (БЗ)** – это база знаний (см. всё выше). Основу базы знаний составляют факты и правила.

**База правил (БП)** — формализованные с помощью правил продукций знания о конкретной предметной области.

**Механизмы вывода (МВ)** – модуль, отвечающий за преобразование фактов о текущей обстановке в какие-то конкретные советы / решения посредством правил вывода (по знаниям БЗ).

**Рабочая память (РП)** – область памяти, в которой хранится множество фактов, описывающих текущую ситуацию, и все пары атрибут-значение, которые установлены к определенному моменту. РП – это динамическая часть базы знаний, содержимое которой зависит от окружения решаемой задачи. В простейших ЭС факты РП не изменяются в процессе решения задачи, но существуют системы, в которых допускается изменение фактов в РП.

**Механизм вывода** выполняет две основные функции: просмотр существующих в рабочей памяти фактов и правил из БП, а также добавление в РП новых фактов; определение стратегии поиска и применения правил. Для выполнения указанных функций механизм вывода включает **компоненту вывода и управляющую компоненту.**

**Цикл работы ЭС.**

Экспертная система работает в двух режимах: приобретения знаний и решения задач (режим консультации, или режим использования).

**Приобретение знаний** – это передача потенциального опыта решения проблемы от некоторого источника знаний и преобразование его в вид, который позволяет использовать эти знания в программе (например, от эксперта с помощью инженера знаний).

В **режиме консультации** общение с ЭС осуществляет конечный пользователь, которого интересует результат и (или) способ получения решения. В режиме консультации данные о задаче пользователя обрабатываются диалоговой компонентой, которая распределяет роли участников (пользователя и ЭС), организует их взаимодействие в процессе кооперативного решения задачи, преобразует данные пользователя о задаче во внутренний язык системы, преобразует сообщения системы, представленные на внутреннем языке, в язык пользователя.

Мне кажется это более правильное:

Механизм вывода работает циклически, при этом в одном цикле может срабатывать только одно правило. В цикле выполняются следующие операции:

1) **сопоставление** - образец правила сопоставляется с имеющимися в РП фактами;

2) **выбор** - если в конкретной ситуации может быть применено сразу несколько правил, то из них выбирается одно, наиболее подходящее по заданному критерию, т. е. происходит разрешение конфликтного выбора;

3) **срабатывание** - если образец правила при сопоставлении совпал с какими-либо фактами из РП, то правило срабатывает, оно отмечается в БП;

4) **действие** - рабочая память подвергается изменению путем добавления в нее заключения сработавшего правила.

Условием окончания цикла является либо исчерпание всех правил из БП, либо выполнение некоторого условия, которому удовлетворяет содержимое рабочей памяти (например, появление в ней какого-то образца), либо комбинация этих способов.

**Операции цикла вывода:**

* Ввод (наблюдений)
* Интерпретация
* Усвоение
* Гипотезы
* Выводы (на основе правил)
* Заключение

# Демоны, присоединенные процедуры и механизм наследования в интеллектуальной системе с фреймовой моделью представления знаний. Правила вывода в логических моделях

Модуль 2 - Лекция 8, раздел "Механизм наследования в интеллектуальных системах"​

*Флешбэк на 10 вопросов назад. Повторяемся.*

Присоединённые процедуры – составная часть фрейма, связанная с самим фреймом или с одним из его слотов.

Демоны – присоединённые процедуры, запускаемые автоматически при обращении к фреймворку (IF-NEEDED, IF-ADDED, IF-REMOVED).

Наследование посредством AKO-указателей на родителя, есть 4 типа: U (Unique) – не наследовать, S (Same) – полное наследование, R (Range) – заданное значение попадает в диапазон родителя, O (Override) – использует значение родителя, если текущее пустое.

Правила вывода – составная часть логический моделей. Базовые аспекты см. в разделах *Исчисление Высказываний* и *Исчисление Предикатов*.

# Структура и классификация экспертных систем (ЭС), отличие ЭС от других программных продуктов, цикл работы ЭС, технология проектирования и разработки ЭС

Модуль 2 - Лекция 8, весь текст​

Изображение выглядит как текст, снимок экрана, диаграмма, Шрифт

Автоматически созданное описание

**Структура** снизу вверх: интерфейс с пользователем – трансляция естественного языка на внутренний язык представления знаний.

Полученная информация записывается в рабочую память. Если содержатся новые знания – связь с БЗ посредством модуля приобретения знаний.

Через подсистему логического вывода на основе знаний из БЗ формируется решение для пользователя. Если требуется объяснение – модуль отображения и разъяснения решений.

**Классификация ЭС** **по типу решаемых задач:**

1. Интерпретирующие (задачи распознавания образов; хим состав веществ и тд)
2. Прогнозирующие
3. Диагностические
4. Системы проектирования
5. Системы планирования
6. Система мониторинга
7. Наладочные системы
8. Системы оказания помощи
9. Обучающие системы
10. Системы контроля

**Классификация ЭС** **в зависимости от способа и учета временного признака:**

1. Статические ЭС разрабатываются в предметных областях, в которых база знаний и интерпретируемые данные не меняются во времени.
2. Квазидинамические ЭС описывают ситуацию, которая меняется с некоторым интервалом времени.
3. Динамические системы работают с датчиками объектов в реальном времени.

**Классификация ЭС по способу формирования решения:**

1. Анализирующие (осуществляется выбор решения из множества известных решений на основе анализа данных)
2. Синтезирующие (решение синтезируется из отдельных фрагментов знаний)

**Классификация ЭС по степени интеграции:**

1. Автономные (работают непосредственно в режиме консультаций с пользователем)
2. Гибридные (программные комплексы, агрегирующие стандартные пакеты прикладных программ и средства манипулирования данными. В таких системах происходит стыковка не только разных пакетов, но и разных технологий)

**Отличия ЭС от других программных продуктов:**

1. Моделируют механизм мышления человека
2. Формирует определенные соображения и выводы, основываясь на тех знаниях, которыми она располагает
3. При решении задач основными являются эвристические и приближенные методы, которые не всегда гарантируют успех. Не требуют исчерпывающей исходной информации, и существует определенная степень уверенности (или неуверенности) в том, что предлагаемое решение является верным.

**Цикл работы ЭС**

Изображение выглядит как текст, диаграмма, План, Технический чертеж

Автоматически созданное описание

Режим приобретения знаний – знания передаются системе (экспертом / инженером знаний / иным специалистом).

Режим консультации – осуществляет работу с конечным пользователем. Диалоговая компонента распределяет роли между пользователями, преобразует естественный язык в формальный язык системы и наоборот. После обработки данные поступают в РП (рабочую память), затем на основе БЗ интерпретатор формирует решение. Если у пользователя остались вопросы – система предоставит подробное пояснение решения.

**Технологии проектирования и разработки ЭС.**

Промышленная технология создания ЭС включает следующие этапы:

* исследование выполнимости проекта;
* разработку общей концепции системы;
* разработку и тестирование серии прототипов;
* разработку и испытание головного образца;
* разработку и проверку расширенных версий системы;
* привязку системы к реальной рабочей среде.

Проектирование ЭС основано на трех главных принципах:

1. Мощность ЭС = мощностью БЗ и возможностями ее пополнения;
2. Знания являются правдоподобными, эвристическими, эмпирическими, неопределенными;
3. Неформальный характер решаемых задач и используемых знаний делает необходимым обеспечение активного диалога пользователя с ЭС при ее работе

Этапы разработки:

1. Идентификация (определяются задачи, подлежащие решению, цели разработки, эксперты и типы пользователей)
2. Концептуализация (содержательный анализ проблемной области, выявляются используемые понятия и их взаимосвязи)
3. Формализации (выбираются способы представления знаний, определяются способы интерпретации знаний, моделируется работа системы)
4. Выполнение (заполнение базы знаний)
5. Тестирование (проверка в интерактивном режиме)
6. Опытная эксплуатация (пригодность ЭС для конечного пользователя)

# Нечеткая логика. Нечеткие множества. Строгое представление нечетких множеств. Основные характеристики нечетких множеств

Модуль 2 - Лекция 9, весь текст​

**Нечеткая логика** позволяет определить промежуточные значения для таких общепринятых оценок, как да-нет, истинно-ложно, черное-белое и т. п. Вводит понятие характеристической функции, определяющую степень истинности или ложности.

**Нечеткие множества** – основополагающее явление нечеткой логики. **Формальная запись НМ:** пусть Е – универсальное (universal) или несущее множество, 𝑥 – элемент Е, а 𝑅 – некоторое свойство. Определим для несущего множества Е обычное (четкое) подмножество А, элементы которого удовлетворяют свойству 𝑅, как множество упорядоченных пар

𝐴 = {𝜇𝐴(𝑥) ∕ 𝑥},

где 𝜇𝐴(𝑥) – характеристическая функция принадлежности (ХФП).  
𝜇𝐴(𝑥) – характеристическая функция ***принадлежности***, принимающая значение 1, если элемент 𝑥 удовлетворяет свойству 𝑅, и 0 — в противном случае. Указывает степень (или уровень) принадлежности элемента 𝑥 подмножеству А. Множество М называется множеством принадлежности

*Справка: ХФ – для чёткого множества (либо 0, либо 1). ХФП – для нечёткого множества (в принадлежности в диапазоне).*

**Основные характеристики НМ**

**Высота НМ** – верхняя граница значений его функции принадлежности.

**Нормальное НМ** – если высота = 1. Иначе – **субнормальное**.

**Пустое НМ** – если для любого x 𝜇𝐴(𝑥)*=0*.

**Унимодальное НМ** – если ХФП=1 только у одного элемента.

**Носитель НМ** – обычное подмножество, для каждого элемента которого ХФП>0.

**Точки перехода НМ** – те элементы НМ, у которых ХФП=0.5.

# Операции над нечеткими множествами. Логические операции. Алгебраические операции

Модуль 2 - Лекция 10, весь текст​

***Логические операции.***

**Включение (А ⊂ 𝐵).** Пусть А и 𝐵 — нечеткие множества на универсальном множестве 𝐸. Говорят, что А содержится в 𝐵, если

∀𝑥 ∈ 𝐸 𝜇𝐴(𝑥) ≤ 𝜇𝐵(𝑥)

**Равенство (А = 𝐵).** Множества А и 𝐵 равны, если ∀𝑥 ∈ 𝐸 𝜇𝐴(𝑥) = 𝜇𝐵(𝑥)

**Объединение (𝐴 ∪ 𝐵)**. Объединением нечетких множеств А и 𝐵 называется нечеткое множество, обозначаемое А и 𝐵, функция принадлежности которого определяется как:

∀𝑥 ∈ 𝐸 𝜇𝐴∪𝐵(𝑥) = max(𝜇𝐴(𝑥), 𝜇𝐵(𝑥))

**Пересечение (𝐴 ∩ 𝐵).** Пересечением А и 𝐵 называется наибольшее нечеткое подмножество, содержащееся одновременно в множествах А и 𝐵:

𝜇𝐴∩𝐵(𝑥) = min(𝜇𝐴(𝑥), 𝜇𝐵(𝑥))

**Дополнение (𝐵 =** **А̅).** Дополнение А̅ нечеткого множества А имеет функцию принадлежности

𝜇𝐵(𝑥) = 1 − 𝜇𝐴(𝑥)

**Разность (А − 𝐵).** Разностью А − 𝐵 = 𝐴 ∩ 𝐵̅ называют множество с функцией принадлежности

𝜇𝐴−𝐵(𝑥) = 𝜇𝐴∩𝐵̅(x) = min(𝜇𝐴(𝑥), 1 − 𝜇𝐵(𝑥)).

***Алгебраические операции.***

**Алгебраическое произведение (А ∙ 𝐵)** нечетких множеств А и 𝐵 обозначается и определяется так

∀𝑥 ∈ 𝐸 𝜇𝐴∙𝐵(𝑥) = 𝜇𝐴(𝑥) ∙ 𝜇𝐵(𝑥).

**Алгебраическое сумма (А ̂+ 𝐵)**

∀𝑥 ∈ 𝐸 𝜇𝐴+𝐵(𝑥) = 𝜇𝐴(𝑥) + 𝜇𝐵(𝑥) – 𝜇𝐴(𝑥) ∙ 𝜇𝐵(𝑥).

**Оператор увеличения нечеткости (Ф(А, k))**

Изображение выглядит как Шрифт, рукописный текст, каллиграфия, типография

Автоматически созданное описание

# Четкое множество «∝-уровня». Нечеткая и лингвистическая переменные. Нечеткие числа

Модуль 2 - Лекция 10, раздел "Четкое множество ∝-уровня"

Множеством α-уровня НМ А универсального множества Е называется чёткое множество Аα универсального множества Е, для которого:

где .

Понятия нечеткой и лингвистической переменных используются при описании объектов и явлений с помощью нечетких множеств.

**Лингвистические переменные (ЛП)** были предложены как средство моделирования нечеткости естественного человеческого языка. В настоящее время ЛП являются распространенным способом описания сложных систем, параметры которых по каким-либо причинам должны рассматриваться не с количественных позиций, а как качественные.

Другой сферой применения ЛП является нечеткий логический вывод, отличие которого от обычного заключается в том, что истинность логических высказываний описывается не двумя значениями ЛОЖЬ и ИСТИНА или 0 и 1, а множеством значений в интервале [0, 1].

В основе понятия лингвистической переменной лежит термин **«нечеткая переменная»**, обозначающий нечеткое множество, которому было присвоено некоторое наименование.

Нечеткая переменная характеризуется тройкой 〈∝, 𝑋, 𝐴〉, где:

∝ – наименование переменной;

𝑋 – универсальное множество (область определения ∝),

А – нечеткое множество на 𝑋, описывающее ограничения (т. е. 𝜇𝐴(𝑥)) на значения нечеткой переменной ∝.

**Лингвистической переменной** называется набор 〈𝛽, 𝑇, 𝑋, 𝐺, 𝑀〉, где

𝛽 – наименование лингвистической переменной;

𝑇 – множество ее значений, представляющих собой наименования нечетких переменных, областью определения каждой из которых является множество 𝑋; множество 𝑇 называется базовым терм-множеством лингвистической переменной;

𝐺 – синтаксическая процедура, позволяющая оперировать элементами множества 𝑇, в частности, генерировать новые значения; множество 𝑇 ∪ 𝐺(𝑇) где 𝐺(𝑇) – множество сгенерированных значений, называется расширенным терм-множеством лингвистической переменной;

𝑀 – семантическая процедура, позволяющая превратить каждое новое значение лингвистической переменной, образуемое процедурой 𝐺, в нечеткую переменную, т. е. сформировать соответствующее нечеткое множество.

**Нечеткие числа** – нечеткие переменные, определенные на числовой оси. Иначе говоря, нечеткое число определяется как нечеткое множество А на множестве действительных чисел ℝ с функцией принадлежности  
𝜇𝐴(𝑥) ∈ [0, 1], где 𝑥 - действительное число, т. е. 𝑥 ∈ ℝ.

# Часть 2. Вычислительный интеллект

# Особенности человеческого восприятия информации и современных систем управления. Вычислительный интеллект (ВИ). Методы ВИ

Модуль 3 - Лекция 11, раздел "Особенности человеческого восприятия информации и современных систем управления"​

Одним из **недостатков человеческого восприятия** являются ограниченные возможности по переработке поступающей информации, тем более что эта информация может быть неполной, чрезвычайно разнородной и искаженной.

Функциональное предназначение **мнемических процессов** (запоминание, воспроизведение, сохранение и забывание усвоенной ранее информации) при принятии управленческих решений заключается в обеспечении:

* быстрой актуализации необходимой ЛПР в конкретной ситуации информации;
* ситуативной избирательности памяти (ранжировании данных по их значимости);
* надлежащего объема памяти и поддержания ее в рабочем состоянии при постоянно возрастающем утомлении;
* структурирования единиц (паттернов) в памяти в соответствии с их логическими, функциональными и семантическими взаимосвязями;
* удерживания в памяти противоречивой информации.

В когнитивной психологии установлено, что **кратковременная память** ЛПР способна оперировать не более чем **с 7±2 факторами**.

**Когнитивные деформации**: эвристика доступности (более вероятно то событие, которое легче вспомнить); эвристика конкретности (более значима простая и понятная информация); эффект Ирвина (наиболее вероятно наступление желаемого события); эвристика репрезентативности (более вероятно наступление того события, которое соответствует накопленному опыту человека).

**Современные интеллектуальные системы** позволяют не только избежать приведенных когнитивных деформаций памяти ЛПР, но и обладают возможностями позволяющими структурировать, ранжировать и актуализировать чрезвычайно сложную и большую по объемам информацию.

**Вычислительный интеллект (ВИ)** – научное направление, где решаются задачи искусственного интеллекта на основе новых нетрадиционных методов вычислений,

**Технология ВИ** – совокупность нетрадиционных методов вычислений и средств обработки знаний, документооборота, методов выработки и выбора альтернативных вариантов решений, объединенных в целостную технологическую систему для принятия и доведения решений до исполнителей.

Методы ВИ:

* Нейросетевые
* Нечеткая логика
* Генетические
* (+) Когнитивная графика (способы визуализации для активирования наглядно-образных механизмов мышления ЛПР)
* (+) Нелинейная графика (эволюционные процессы в нелинейных динамических процессах)
* (+) Теория Хаоса (сложные нелинейные динамические системы, резко чувствительные к начальным условиям; система кажется случайной даже если правила детерминированы)

# Искусственные нейронные сети. Основные проблемы, решаемые искусственными нейронными сетями

Модуль 3 - Лекция 11, раздел "Искусственные нейронные сети (ИНС). Основные проблемы"​

Искусственные нейронные сети (ИНС) строятся по принципам организации и функционирования их биологических аналогов. Такие сети предназначены для решения широкого круга задач: распознавания образов, идентификации, прогнозирования, оптимизации, управления сложными объектами.

Основные проблемы, решаемые ИНС:

* Классификация образов
* Кластеризация и категоризация
* Аппроксимация функций
* Предсказание / прогноз
* Оптимизация

2 подхода к созданию НС: rule-base (на правилах если-то) и case-based (на основе ситуативных исторических примерах).

# Биологический нейрон. Искусственный нейрон. Активационная функция. Классификация и свойства искусственных нейронных сетей. Обучение искусственных нейронных сетей

Модуль 3 - Лекция 12, разделы "Биологический нейрон", "Искусственный нейрон", "Классификация и свойства ИНС" и "Обучение ИНС"​

**Биологический нейрон:** ядро, сома, аксон и несколько дендритов. Дендриты связаны с аксонами разных нейронов и получают от них сигналы, на стыке – синапс (синаптическая щель, нейромедиаторы и всё такое).

**Искусственный нейрон:** искусственный нейрон обладает группой синапсов – однонаправленных входных связей, соединенных с выходами других нейронов, а также имеет аксон — выходную связь данного нейрона, с которой сигнал (возбуждения или торможения) поступает на синапсы следующих нейронов.

Изображение выглядит как текст, снимок экрана, диаграмма, линия

Автоматически созданное описание

**Активационные функции.**

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

Выход нейрона есть функция его состояния: 𝑦 = 𝑓{𝑆}, где 𝑓 - активационная функция, более точно моделирующая нелинейную передаточную характеристику биологического нейрона и предоставляющая нейронной сети большие возможности.

**Классификация и свойства искусственных нейронных сетей (ИНС):**

* Однослойные ИНС (входной слой – не считается; вычисления сугубо на втором слое)
* Многослойные ИНС (опять же, входной слой не считается; может содержать несколько промежуточных слоёв)

Так как умножение матриц ассоциативно, то двухслойная линейная сеть эквивалентна одному слою с весовой матрицей, равной произведению двух весовых матриц. Следовательно, любая многослойная линейная сеть может быть заменена эквивалентной однослойной сетью.

**Обучение ИНС**

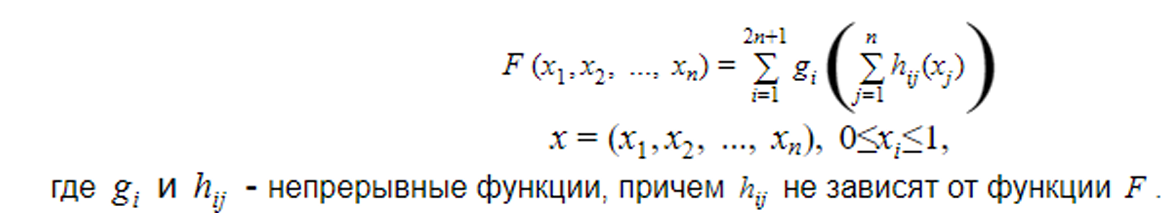
Обучение ИНС -последовательное предъявление ИНС входных векторов и одновременная подстройка весов в соответствии с заданной процедурой. Различают алгоритмы с учителем и без учителя.

# Теорема Колмогорова. Алгоритм обучения персептрона. Линейная разделимость и персептронная представляемость

Модуль 3 - Лекция 12, раздел "Теорема Колмогорова. Алгоритм обучения персептрона"​

**Теорема Колмогорова.** Рассмотрим двухслойную сеть, n входов, один выход, сигмоидальная АФ для скрытого слоя, тождественная АФ для выходного слоя.

Любая непрерывная функция 𝐹, определенная на 𝑛-мерном единичном кубе, может быть представлена в виде суммы 2𝑛+1 суперпозиций непрерывных и монотонных отображений единичных отрезков.



Эта теорема означает, что для реализации функций многих переменных достаточно операций суммирования и композиции функций одной переменной. К сожалению, при всей своей математической красоте, теорема Колмогорова малоприменима на практике. Это связано с тем, что функции ℎ𝑖𝑗 − в общем случае негладкие и трудновычислимые; также неясно, каким образом можно подбирать функции 𝑔𝑖 для данной функции 𝐹. Роль этой теоремы состоит в том, что она показала принципиальную возможность реализации сколь угодно сложных зависимостей с помощью относительно простых автоматов типа нейронных сетей.

**Развитие. Теорема Колмогорова–Арнольда–Хехт–Нильсена (КАХН).** Из КАХН следует, что для любой функции многих переменных существует отображающая ее НС фиксированной размерности.

**Персептрон** – однослойная нейронной сети, при этом каждый персептронный нейрон в качестве активационной функции использует функцию единичного скачка (пороговую).

**Обучение персептрона** (это обучение с учителем) состоит в подстройке весовых коэффициентов 𝑤𝑖, где 𝑖 =1, 2, . . . , 𝑛. Алгоритм обучения:

**Шаг 0.** Проинициализировать весовые коэффициенты 𝑤𝑖 , 𝑖 = 1, 2, . . . , 𝑛 небольшими случайными значениями, например, из диапазона [-0,3; 0,3].

**Шаг 1.** Подать на вход персептрона один из обучающих векторов 𝑋𝑘 и вычислить его выход 𝑦.

**Шаг 2.** Если выход правильный (𝑦 = 𝑦𝑘), перейти на шаг 4. Иначе вычислить ошибку – разницу между верным и полученным значениями выхода: 𝛿 = 𝑦𝑘 − 𝑦.

**Шаг 3.** Весовые коэффициенты модифицируются по следующей формуле:

𝑤𝑖𝑗𝑡+1 = 𝑤𝑖𝑗𝑡 + 𝜈 ∙ 𝛿 ∙ 𝑥𝑖

Здесь 𝑡 и 𝑡 + 1 – номера соответственно текущей и следующей итераций;  
𝜈 - коэффициент скорости обучения (0<𝜈≤1) (эмпирически [0,5; 0,7]);  
𝑥𝑖. 𝑖-я компонента входного вектора 𝑋𝑘.

**Шаг 4.** Шаги 1–3 повторяются для всех обучающих векторов. Один цикл последовательного предъявления всей выборки называется **эпохой**. Обучение завершается по истечении нескольких эпох, когда сеть перестанет ошибаться.

Нейрон способен отделить только такие два множества векторов входов, для которых имеется гиперплоскость, отсекающая одно множество от другого. Такие множества называют **линейно разделимыми**.

Изображение выглядит как линия, зарисовка, рисунок, Шрифт

Автоматически созданное описание

Невозможность реализации однослойным персептроном этой функции получила название **проблемы исключающего ИЛИ**. Видно, что однослойный персептрон крайне ограничен в своих возможностях точно представить наперед заданную логическую функцию.

# Нейронная сеть обратного распространения. Алгоритм обучения сети обратного распространения

Модуль 3 - Лекция 13, раздел "Нейронные сети обратного распространения"​

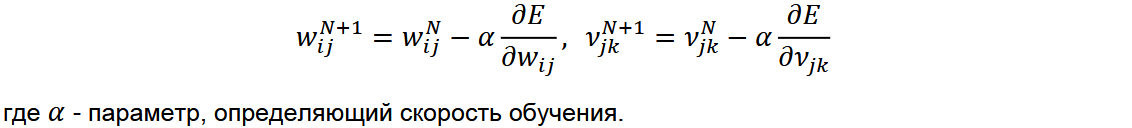
**Сети обратного распространения** – получили своё название из-за используемого алгоритма обучения, в котором ошибка распространяется от выходного слоя к входному, т.е. в направлении, противоположном направлению распространения сигнала при нормальном функционировании сети.

При обучении нейронной сети ставится **задача минимизации целевой функции ошибки**, которая находится по методу наименьших квадратов.

Изображение выглядит как текст, Шрифт, снимок экрана, линия

Автоматически созданное описание

Обучение нейросети производится известным **оптимизационным методом градиентного спуска**, т.е. на каждой итерации производится следующее изменение веса



Обычно используется **сигмоидальная АФ**.

**Алгоритм обучения сети обратного распространения**

**Шаг 1.** Инициализация сети.

Весовым коэффициентам присваиваются малые случайные значения, например, из диапазона (-0,3; 0,3); задаются 𝜀 – параметр точности обучения, 𝛼 – параметр скорости обучения (как правило, 𝛼≈0,1 и может еще уменьшаться в процессе обучения), 𝑁𝑚𝑎𝑥 – максимально допустимое число итераций.

**Шаг 2.** Вычисление текущего выходного сигнала.

На вход сети подается один из образов обучающей выборки и определяются значения выходов всех нейронов нейросети.

**Шаг 3.** Настройка синаптических весов.

Рассчитать изменение весов для выходного слоя нейронной сети по формулам:

Изображение выглядит как текст, Шрифт, линия, белый

Автоматически созданное описание

**Шаг 4.** Шаги 2–3 повторяются для всех обучающих векторов. Обучение завершается по достижении для каждого из обучающих образов значения функции ошибки, не превосходящего 𝜀 или после максимально допустимого числа итераций – 𝑁𝑚𝑎𝑥.

# Нейронная сеть встречного распространения (сеть Кохонена). Алгоритм обучения сети Кохонена

Модуль 3 - Лекция 13, раздел "Сеть встречного распространения (сеть Кохонена)"​

Алгоритм обучения «победитель забирает все» для задачи классификации Кохонена (Т. Kohonen). Суть: классификация образов без учителя. Схожесть – по евклидову расстоянию d(x, y) = √∑ (xi − yi)2. На выходе активируется один из нейронов.

**Алгоритм обучения сети Кохонена**

**Шаг 1.** Инициализация сети. Весовым коэффициентам сети 𝑤𝑖𝑗, 𝑖 = 1,2, … , 𝑛, 𝑗=1,2, … , 𝑚 присваиваются малые случайные значения. Задаются значения 𝛼0 – начальный темп обучения и D0— максимальное расстояние между весовыми векторами (столбцами матрицы 𝑊).

**Шаг 2.** Предъявление сети нового входного сигнала 𝑋.

**Шаг 3.** Вычисление расстояния от входа X до всех нейронов сети:

dj = ∑(xi − wij)2, 𝑗 = 1,2, … , 𝑚.

**Шаг 4.** Выбор нейрона 𝑘, 1 ≤ 𝑘 ≤ 𝑚 с наименьшим расстоянием dk.

**Шаг 5.** Настройка весов нейрона 𝑘 и всех нейронов, находящихся от него на

расстоянии, не превосходящем DN:

Изображение выглядит как Шрифт, типография, каллиграфия, рукописный текст

Автоматически созданное описание

**Шаг 6.** Уменьшение значений αN, DN.

**Шаг 7.** Шаги 2—6 повторяются до тех пор, пока веса не перестанут меняться (или пока суммарное изменение всех весов станет очень мало)

После обучения классификация выполняется посредством подачи на вход сети испытуемого вектора, вычисления расстояния от него до каждого нейрона с последующим выбором нейрона с наименьшим расстоянием как индикатора правильной классификации

# Двухслойная сеть встречного распространения и ее обучение. Входные и выходные звезды Гроссберга и их обучение

Модуль 3 - Лекция 13, раздел "Двухслойная сеть встречного распространения"​

**Входные и выходные звезды Гроссберга**

Входная звезда Гроссберга (S. Grossberg) состоит из нейрона, на который подается группа входов, умноженных на синапсические веса.

Выходная звезда является нейроном, управляющим группой весов.

Изображение выглядит как круг, снимок экрана, диаграмма, белый

Автоматически созданное описаниеИзображение выглядит как круг, снимок экрана, диаграмма, часы

Автоматически созданное описаниеВходные и выходные звезды могут быть взаимно соединены в сети любой сложности.

**Обучение входной звезды.**

Входная звезда выполняет распознавание образов, т. е. она обучается реагировать на определенный входной вектор 𝑋 и ни на какой другой. Это обучение реализуется путем настройки весов таким образом, чтобы они соответствовали входному вектору. Входная звезда имеет тождественную активационную функцию f(s) = s , т. е. выход входной звезды определяется как взвешенная сумма ее входов:

Y = ∑ wixi

В процессе обучения веса корректируются следующим образом:

𝑤𝑖𝑁+1 = 𝑤𝑖𝑁 +∝𝑁 (𝑥𝑖 – 𝑤𝑖𝑁),

где 𝑤𝑖 – весовой коэффициент входа 𝑥𝑖; ∝𝑁 – нормирующий коэффициент обучения, который имеет начальное значение 0,1 и постепенно уменьшается в процессе обучения.

**Обучение выходной звезды.**

Если входная звезда учится реагировать на определенный вход, то выходная звезда обу-

чается выдавать требуемый целевой выход.

Чтобы обучить нейрон выходной звезды, его веса настраиваются в соответствии с требу-

емым целевым вектором 𝑌.

Формула коррекции весов имеет следующий вид:

𝑤𝑖𝑁+1 = 𝑤𝑖𝑁 + 𝛽𝑁(𝑦𝑖 − 𝑤𝑖𝑁),

где 𝛽𝑁 представляет собой нормирующий коэффициент обучения, который вначале приблизительно равен единице и постепенно уменьшается до нуля в процессе обучения.

**Двухслойная сеть встречного распространения.**

Состоит из двух слоёв: слой Кохонена и звезд Гроссберга. Слой Кохонена классифицирует входные векторы в группы схожих. Это достигается с помощью такой подстройки весов слоя Кохонена, что близкие входные векторы активируют один и тот же нейрон данного слоя. Затем слой Гроссберга дает требуемые выходы.

Изображение выглядит как диаграмма, линия, Технический чертеж, План

Автоматически созданное описание

В режиме нормального функционирования предъявляются входные векторы 𝑋 и 𝑌, и обученная сеть дает на выходе векторы 𝑋′ и 𝑌′, являющиеся аппроксимациями соответственно для 𝑋 и 𝑌. Это не было бы интересно, если не учитывать способность этой сети к обобщению. Благодаря обобщению предъявление только вектора 𝑋 (с вектором 𝑌 = 0) порождает как выходы 𝑋′ так и выходы 𝑌′. Если F - функция, отображающая 𝑋 в 𝑌′, то сеть аппроксимирует ее. Кроме того, если функция F обратима (если функция y = f(x) такова, что для любого её значения y0 уравнение f(x) = y0 имеет относительно x единственный корень, то говорят, что функция f обратима), то предъявление только вектора 𝑌 (при 𝑋 = 0) порождает выходы 𝑋′.

**Алгоритм обучения двухслойной сети встречного распространения**

**Шаг 1.** Произвести единичную нормировку всех векторов 𝑋, 𝑌 обучающего множества.

**Шаг 2.** Весовым коэффициентам сети 𝑤𝑖𝑗, 𝑣𝑗𝑖, 𝑖 = 1, 2, … , 2𝑛, 𝑗 = 1, 2, … , 𝑚 присвоить малые случайные значения и произвести единичную нормировку матриц W, V по столбцам. Положить ∝0= 0,7, 𝛽0 = 0,1.

**Шаг 3.** Подать на вход сети обучающий набор (𝑋, 𝑌) и определить единственный нейрон – «победитель» в слое Кохонена (весовой вектор которого дает максимальное скалярное произведение с входным вектором). Выход этого нейрона установить равным 1, выходы всех остальных нейронов слоя Кохонена положить равными 0. Скорректировать веса выигравшего нейрона:

Изображение выглядит как Шрифт, текст, линия, типография

Автоматически созданное описание

**Шаг 4.** Подать выходной вектор слоя Кохонена на вход слоя Гроссберга. Скорректировать веса слоя Гроссберга, связанные с выигравшим нейроном слоя Кохонена:

𝑣𝑘𝑖𝑁+1 = 𝑣𝑘𝑖𝑁 + 𝛽𝑁(𝑧𝑖 − 𝑣𝑘𝑖𝑁) (здесь 𝑘 - номер выигравшего нейрона).

**Шаг 5.** Уменьшить значения ∝0, 𝛽0.

**Шаг 6.** Повторять шаги 3—5 до тех пор, пока каждая входная пара из обучающего множества не будет порождать аналогичную выходную пару.

# Двухслойная сеть встречного распространения. Алгоритм обучения сети встречного распространения

Модуль 3 - Лекция 13, раздел "Двухслойная сеть встречного распространения"​

Двухслойная сеть встречного распространения (см. выше) – комбинация сеть Кохонена + звёзды Гроссберга.

Алгоритм обучения сети встречного распространения (Кохонена) – см. выше.

# Стохастические методы обучения. Обучение Больцмана. Алгоритм обучения Больцмана. Обучение Коши

Модуль 3 - Лекция 14, раздел "Стохастические методы обучения" и "Обучение Больцмана"​

**Стохастические методы** обучения выполняют псевдослучайные изменения величин весов, сохраняя те изменения, которые ведут к улучшениям (для «перескока» локальных экстремумов).

**Обучение Больцмана. «Имитация отжига»**

Алгоритм обучения Больцмана

**Шаг 1.** Определить переменную T, представляющую искусственную температуру. Придать T большое начальное значение.

**Шаг 2.** Подать на вход сети один из входных образов обучающей выборки и вычислить реальный выход и значение функции ошибки сети (как в алгоритме обратного распространения).

**Шаг 3.** Придать случайное изменение ∆𝑤𝑖𝑗 выбранному весу 𝑤𝑖𝑗 и пересчитать выход сети и изменение функции ошибки в соответствии со сделанным изменением веса.

**Шаг 4.** Если функция ошибки уменьшилась, то сохранить изменение веса. Если изменение веса приводит к увеличению функции ошибки, то вероятность сохранения этого изменения вычисляется с помощью распределения Больцмана: . Выбирается случайное число 𝑟 из равномерного распределения от нуля до единицы. Если вероятность 𝑃(∆𝑤𝑖𝑗) больше, чем 𝑟, то изменение сохраняется, в противном случае величина веса возвращается к предыдущему значению.

**Шаг 5.** Повторять шаги 3 и 4 для каждого из весов сети, постепенно уменьшая температуру 𝑇, пока не будет достигнуто допустимо низкое значение целевой функции.

**Шаг 6.** Повторять шаги 2—5 для всех векторов обучающей выборки (возможно неоднократно), пока функция ошибки не станет допустимой для каждого из них.

**Обучение Коши**

В этом методе распределение Больцмана заменяется на распределение Коши. Распределение Коши имеет более высокую вероятность больших шагов. Дисперсия распределения Коши бесконечна.

# Сети с обратными связями. Сеть Хопфилда

Модуль 3 - Лекция 14, раздел "Сети с обратными связями. Сеть Хопфилда"​

**Сети с обратными связями.**

В общем случае может быть рассмотрена нейронная сеть, содержащая произвольные обратные связи, т. е. пути, передающие сигналы от выходов к входам. Отклик таких сетей является динамическим, т. е. после подачи нового входа вычисляется выход и, передаваясь по обратной связи, модифицирует вход. Затем выход повторно вычисляется, и процесс повторяется снова и снова. Для устойчивой сети последовательные итерации приводят ко все меньшим изменениям выхода, и в результате выход становится постоянным. Для многих сетей процесс никогда не заканчивается, такие сети называются неустойчивыми. Неустойчивые сети обладают интересными свойствами и могут рассматриваться в качестве примера хаотических систем, но для большинства практических приложений используются сети, которые дают постоянный выход.

**Сеть Хопфилда.**

Изображение выглядит как диаграмма, линия, круг, Технический чертеж

Автоматически созданное описание

У каждого нейрона – пороговая активационная функция. модели Хопфилда предполагается условие симметричности связей 𝑤𝑖𝑗 = 𝑤𝑗𝑖 с нулевыми диагональными элементами 𝑤𝑖𝑖 = 0. Стационарные состояния, отвечающие минимумам, могут интерпретироваться как образы памяти нейронной сети. Сходимость к такому образу соответствует процессу извлечения из памяти. При произвольной матрице связей W образы также произвольны. Для записи в память сети какой-либо конкретной информации требуется определенное значение весов W, которое может получаться в процессе обучения.

# Правило обучения Хебба. Ортогонализация образов

Модуль 3 - Лекция 14, раздел "Правило обучения Хебба. Процедура ортогонализации образов"​

**Правило Хебба (см. выше)** – правило обучения сети Хопфилда. Пусть задана обучающая выборка образов 𝑋𝑘, 𝑘 = 1, 2, … , 𝐾. Требуется построить матрицу связей W такую, что соответствующая нейронная сеть будет иметь в качестве стационарных состояний образы обучающей выборки.

Сеть Хопфилда нашла широкое применение в системах ассоциативной памяти, позволяющих восстанавливать идеальный образ по имеющейся неполной или зашумленной его версии.

**Процедура ортогонализации образов**

Два различных запоминаемых векторных образа сети 𝑋𝑘, 𝑋𝑙 (𝑘 ≠ 𝑙) называются ортогональными, если их скалярное произведение равно нулю:

Если все запоминаемые образы сети 𝑋𝑘, 𝑘 = 1, 2, … , 𝐾 попарно ортогональны, емкость памяти сети Хопфилда увеличивается до 𝑛 , т. е. сеть может запомнить количество образов, не превосходящее число нейронов в ней. На этом свойстве основано улучшение правила Хебба: перед запоминанием в нейронной сети исходные образы следует ортогонализовать. Процедура расчета весовых коэффициентов в этом случае имеет следующий вид:

**Шаг 1.** Вычисляются элементы матрицы B = (bkl), 𝑘 = 1, 2, … , 𝐾:

**Шаг 2.** Определяется матрица С, обратная к матрице B: C = B−1.

**Шаг 3.** Задаются весовые коэффициенты сети Хопфилда:

Существенным недостатком метода является его нелокальность: прежде чем начать обучение, необходимо наперед знать все обучающие образы. Добавление нового образа требует полного переобучения сети.

# Двунаправленная ассоциативная память

Модуль 3 - Лекция 14, раздел "Сеть ДАП (двунаправленная ассоциативная память)"​

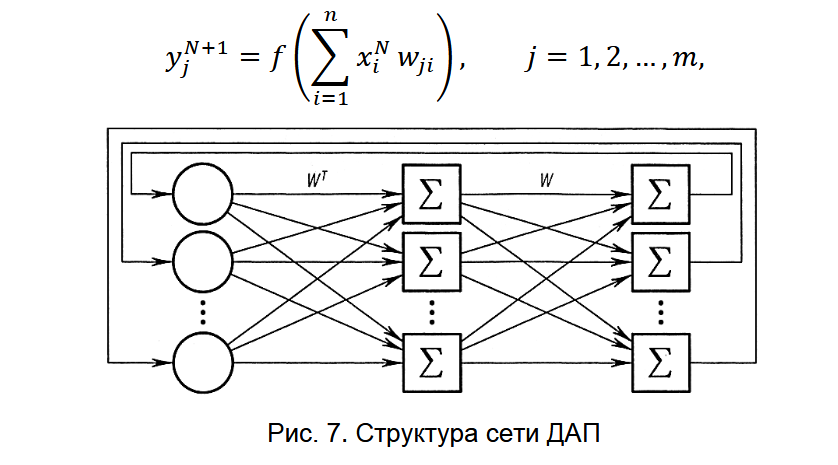
Сеть Хопфилда реализует так называемую автоассоциативную память. Это означает, что образ может быть завершен или исправлен, но не может быть ассоциирован с другим образом.

**Двунаправленная ассоциативная память (ДАП)** является гетероассоциативной: она сохраняет пары образов и выдает второй образец пары, когда ассоциированный с ним первый образец подается на вход сети. Как и сеть Хопфилда, сеть ДАП способна к обобщению, вырабатывая правильные реакции, несмотря на искаженные входы.

Сеть ДАП (рис. 7) содержит два слоя нейронов. Элементы весовой матрицы 𝑤𝑖𝑗 отражают связь между *i*-м нейроном первого слоя и 𝑗-м нейроном второго слоя, 𝑖 = 1, 2, … , 𝑛, 𝑗 = 1, 2, … , 𝑚.

В процессе функционирования сети входной вектор 𝑋 умножается на транспонированную матрицу весов сети WT и подается на вход первого слоя, в результате чего вырабатывается вектор выходных сигналов нейронов первого слоя Y. Вектор Y затем умножается на матрицу W и подается на вход второго слоя, который вырабатывает выходные сигналы, представляющие собой новый входной вектор 𝑋. Этот процесс повторяется до тех пор, пока сеть не достигнет стабильного состояния, в котором ни вектор 𝑋, ни вектор Y не изменяются. Нейроны в обоих слоях сети ДАП функционируют аналогично нейронам сети Хопфилда.

Основным недостатком сети ДАП, как и сети Хопфилда, является небольшая емкость памяти. Так, число запоминаемых ассоциаций не может превышать числа нейронов в меньшем слое.



# Адаптивная резонансная теория

Модуль 3 - Лекция 14, раздел "Сеть APT (адаптивная резонансная теория)"​

Адаптивная резонансная теория включает **две парадигмы**, каждая из которых определяется формой входных данных и способом их обработки. АРТ-1 разработана для обработки двоичных входных векторов, в то время как АРТ-2 может классифицировать как двоичные, так и непрерывные векторы.

Сеть АРТ-1 обучается без учителя и реализует **простой алгоритм кластеризации**. В соответствии с этим алгоритмом первый входной сигнал считается образцом первого кластера. Следующий входной сигнал сравнивается с образцом первого кластера. Говорят, что входной сигнал принадлежит первому кластеру, если расстояние до образца первого кластера меньше порога. В противном случае второй входной сигнал - образец второго кластера.

**Алгоритм функционирования сети АРТ-1**

**Шаг 1.** Инициализация сети: 𝑁 = 1, 𝑚 = 1; tijN = bijN = 1, 𝑖 = 1, 2, … , 𝑛, 𝑗 = 1, 2, … , 𝑚, где bijN – синаптический вес связи от 𝑖-го нейрона первого слоя к 𝑗-му нейрону второго слоя на итерации с номером N, tijN – синаптический вес связи от 𝑗-го нейрона второго слоя к 𝑖-му нейрону первого слоя на итерации с номером N. Веса bijN и tijN, 𝑖 = 1, 2, … , 𝑛 определяют образец, соответствующий нейрону 𝑗. Задать 0<𝑟<1 – значение порога.

**Шаг 2.** Предъявление сети нового бинарного входного сигнала 𝑋 = (𝑥1, 𝑥2, …, 𝑥𝑛).

**Шаг 3.** Вычисление значений соответствия

**Шаг 4.** Выбор образца с наибольшим соответствием. Если yk=0 создать новый кластер и перейти на шаг 8.

**Шаг 5.** Сравнение с порогом. Если отношение порога > r перейти на шаг 7 (вычисляется отношение скалярного произведения входного сигнала и образца с наибольшим значением соответствия к числу единичных бит входного сигнала. Значение отношения сравнивается с порогом, введенном на первом шаге)

**Шаг 6.** Исключение примера с наибольшим значением соответствия. Значение соответствия образца 𝑦𝑘 временно устанавливается равным нулю. Переход к шагу 4 (поиск нового значения 𝑦𝑘).

**Шаг 7.** Адаптация примера с наибольшим значением соответствия.

**Шаг 8.** Включение всех, исключенных на шаге 6, образцов. Положить N=N+1. Возврат на шаг 2.

# Методы оптимизации комбинаторных задач различной степени сложности. Генетические алгоритмы

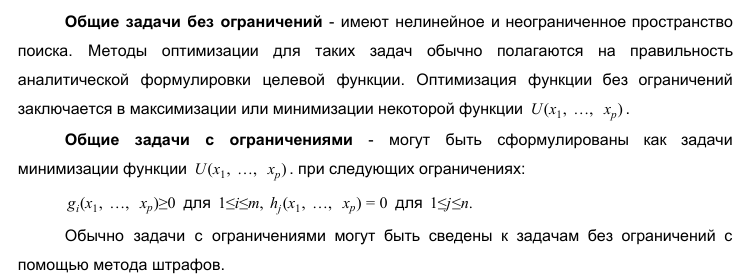
Модуль 3 - Лекция 15, раздел "Методы оптимизации комбинаторных задач различной степени сложности" и "Генетические алгоритмы"​

В общем случае оптимизация или поиск наилучшего значения (набора параметров) некоторой заданной целевой функции является достаточно сложной задачей. Сложность оптимизации обусловливается, прежде всего, видом целевой функции, которая может иметь как глобальный, так и локальный оптимумы.

**Комбинаторные задачи** – характеризуются конечным и дискретным пространством поиска. Сущность любой комбинаторной задачи можно сформулировать следующим образом: найти на множестве 𝑋 элемент 𝑥, удовлетворяющий совокупности условий 𝐾(𝑥), в предположении, что пространство поиска 𝑋 содержит некоторое конечное число различных точек.

**Общие задачи без ограничений** – имеют нелинейное и неограниченное пространство поиска.

**Общие задачи с ограничениями** – могут быть сформулированы как задачи минимизации функции 𝑈(𝑥1, … , 𝑥𝑝) при ограничениях.



**Генетические алгоритмы.** ГА являются стохастическим эвристическим методом, в котором вероятность выбора состояния 𝑆(𝑡 + 1) зависит от состояния 𝑆(𝑡) и косвенно от предыдущих состояний.

Каждый из стохастических и эвристических методов имеет свои достоинства и недостатки, обусловленные формулировкой и размерностью решаемой задачи. При этом математически доказано, что для комбинаторных задач оптимизации средняя эффективность всех алгоритмов для всех возможных задач одинакова.

Использования ГА позволяет выделить следующие условия, при выполнении которых задача решается эффективно:

* Q большое пространство поиска, ландшафт которого является негладким (содержит несколько экстремумов);
* Q сложность формализации оценки качества решения функцией степени пригодности;
* Q многокритериальность поиска;
* Q поиск приемлемого решения по заданным критериям в отличие от поиска единственного оптимального.

Генетические алгоритмы (ГА) относятся к числу универсальных методов оптимизации, позволяющих решать задачи различных типов (комбинаторные, общие задачи с ограничениями и без ограничений) и различной степени сложности. При этом ГА характеризуются возможностью как однокритериального, так и многокритериального поиска в большом пространстве, ландшафт которого является негладким.

# Базовый генетический алгоритм. Последовательные модификации базового генетического алгоритма

Модуль 3 - Лекция 15, раздел "Базовый генетический алгоритм"​

При работе с ГА решения задачи должны быть представлены в виде строки с бинарной, целочисленной или вещественной кодировкой. Способ кодирования предполагает работу со строками фиксированной или переменной длины, возможна также и контекстно-зависимая кодировка.

**Базовый (стандартный) генетический алгоритм.**

Изображение выглядит как текст, снимок экрана, Шрифт, черно-белый

Автоматически созданное описание

**Последовательные модификации ГА.**

Как показывает анализ, модификации ГА отличаются, прежде всего, способом селекции индивидов.

**SSGA** – затрагивает способ формирования промежуточной популяции. Вместо формирования промежуточной популяции выбирает пара лучших решений, и их ребёнок замещает худшего индивида.

**ГА селекционеров (ГАС)** – вводится дополнительное виртуальное понятие «селекционер» – он отбирает T% популяции для создания потомков (родители), которые дают N/2 пар случайным образом.

**Многокритериальные ГА (МГА)** – для задач с многокритериальными оценками. За счёт большого объёма комбинаций селекции и, как следствие, возможных выборов рождено множество модификаций МГА. Одна из самых распространённых – VEGA (VEctor). Основная фича: промежуточная популяция имеет N/k (N – размер, k – количество критериев) особей. Остальное – всё как у людей.

# Параллельные модификации базового генетического алгоритма. Классификация генетических алгоритмов

Модуль 3 - Лекция 15, раздел "Параллельные модификации базового генетического алгоритма" и "Классификация генетических алгоритмов"​

Стандартный ГА представляет собой строго синхронизованный последовательный алгоритм, который в условиях большого пространства поиска или сложного ландшафта пространства поиска может быть неэффективен по критерию времени. Эту проблему позволяет решить другой вид ГА – параллельный генетический алгоритм (ПГА). Следует отметить, что **любая последовательная модификация стандартного ГА может быть преобразована в параллельную**.

По степени распараллеливания можно выделить следующие типы параллельных ГА:

* ПГА на базе популяции;
* ПГА на базе подпопуляций;
* ПГА на базе индивидов.

По степени распараллеливания процессов можно выделить следующие модели

* **Синхронная модель** "ведущий-ведомый", где главный процесс хранит целую популяцию в собственной памяти, выполняет селекцию, скрещивание и мутацию, но оставляет вычисление степени пригодности новых индивидов к подчиненным процессам;
* **Полусинхронная модель** "ведущий-ведомый", где новый индивид обрабатывается по мере освобождения одного из процессов;
* **Асинхронная параллельная модель**, где индивиды популяции хранятся в общей памяти, к которой можно обращаться к параллельным процессам. Каждый процесс выполняет оценку степени пригодности, а также генетические операции.

**Классификация генетических алгоритмов**

В ходе исследований в области генетических алгоритмов и эволюционных алгоритмов в целом появилось большое количество направлений, и их число непрерывно растёт. Классификация ЭА и основные модификации стандартного ГА:

Изображение выглядит как диаграмма, План, зарисовка, Технический чертеж

Автоматически созданное описание

# Когнитивная компьютерная графика. Определение. Степень организованности информации

Модуль 3 - Лекция 16, разделы "Определение" и "Степень организованности информации"​

**Когнитивная компьютерная графика (ККГ)** – компьютерные системы визуализации данных, позволяющие активировать наглядно-образные механизмы мышления ЛПР, облегчающие принятие решения в сложной обстановке или нахождение решения сложной проблемы.

**Степень организованности информации** включает в себя: структурную, параметрическую и динамическую организованности. Для того чтобы достигнуть высокой степени организации информационных образов в информационных системах отображения, передачи, воспроизведения и представления информации необходимо располагать запасами ее пространственных, структурных, параметрических и динамических ресурсов.

# Задача сжатия информации. Анаморфозы (определение)

Модуль 3 - Лекция 16, раздел "Задача сжатия информации. Анаморфозы"​

Способы визуализации информационных образов для представления информации в средствах интеллектуальной поддержки принятия решений выполняют еще одну важную задачу – задачу сжатия информации.

АНАМОРФОЗА — жен., греч. безобразная, но правильно искаженная картина, принимающая в граненом или гнутом зеркале свой вид; безобраза. Толковый словарь Даля. В.И. Даль. 1863 1866.

АНАМОРФОЗА — (греч. аna на, сверх и morfe форма), эффект наложения одного изобразительного мотива на другой, их зрительного слияния.

# Метод анаморфирования

Модуль 3 - Лекция 16, раздел "Метод анаморфирования"​

Среди анаморфоз можно выделить: линейные, площадные и объемные. При этом от анаморфированных изображений требуется максимально возможное сохранение их формы.

**Метод анаморфирования** позволяет визуализировать сложные распределения различных показателей на топографической или электронной карте местности, уменьшая при этом размерность пространства решения пропорционально количеству анализируемых показателей.

**Сущность метода** в том, что графическая часть решения, построенная на топографической или электронной карте местности в евклидовой метрике, преобразуется (анаморфируется) на основе выбранных показателей в двумерный визуальный образ. При этом внутренняя структура визуального образа изменяется таким образом, что распределение выбранных показателей становится равномерным, при сохранении топологического подобия с исходным визуальным образом.

# Проблемы реализации анаморфоз

Модуль 3 - Лекция 16, раздел "Проблемы реализации анаморфоз"​

Построение анаморфоз на компьютере встречает трудности двух типов. Первые из них связаны с **компьютерной реализацией алгоритма**. Это задачи преобразования исходных площадных фигур в форму, пригодную для компьютерной обработки (с этой задачей достаточно успешно справляются такие ГИС, как: ArcInfo, MapInfo, «Панорама» и «Интеграция»), численной реализации алгоритма с контролем сохранения взаимной однозначности преобразованных площадных фигур и представления результатов расчетов в виде, удобным для их дальнейшей обработки.

Второй класс трудностей связан с тем, что условие выравнивания заданной плотности **не определяет анаморфозу однозначно**. Существует бесконечно много преобразований, удовлетворяющих этому условию

# Численные методы построения анаморфоз

Модуль 3 - Лекция 16, раздел "Численные методы построения анаморфоз"​

Один из первых численных методов построения анаморфоз был предложен У. Тоблером [Tobler, 1979]. Достоинства этого алгоритма состоят в его простоте и отсутствии преимуществ одних ячеек по отношению к другим. Очевидные недостатки: результат существенно зависит от выбора направлений координатных осей; итерационный процесс обладает медленной сходимостью.

Метод треугольников разработан в Московском государственном университете им. Ломоносова в 1983 году [Петров и др., 1983]. Очевидные достоинства метода: простота, независимость от выбора какой-либо системы координат, сохранение топологического подобия с оригиналом. Основной недостаток метода состоит в существенной зависимости результата от случайных выборов, при его реализации: в порядке обработки вершин, в случайном поиске нового положения выбранной вершины.

Алгоритм построения анаморфированных изображений лаборатории Лоуренс Беркли [Selvin и др., 1984]. Этот метод позволяет получить анаморфозы хорошего качества и его реализация относительно проста. Основные недостатки метода состоят в том, что:

* окончательный результат существенно зависит от порядка, в котором берутся ячейки;
* ошибки, допущенные на каждом шаге, не исправляются позже и накапливаются.

*В целом дальше чуть не наглый копипаст лекции 17.*

# Основные законы гибридного интеллекта

Модуль 3 - Лекция 17, раздел "Основные законы гибридного интеллекта"​

В.Ф.Венда сформулировал три закона гибридного интеллекта.

**Закон взаимной адаптации.** Синтез и динамика развития любого гибрида-метода – процесс взаимной адаптации компонент гибрида.

Закон утверждает, что необходимое и достаточное условие возникновения и развития гибрида – наличие процессов внутренней (между компонентами гибрида) и внешней (гибрида с внешней средой) взаимной адаптации. Структура гибрида - отображение определенной закономерности процесса взаимной адаптации его внутренних компонентов. Эта структура устойчива, если взаимоадаптация гибрида и внешней среды характеризируется состоянием, при котором эффективность решения задач гибридом лежит в заданном интервале.

**Закон дискретных рядов структур.** Любой гибрид может быть реализован посредством одной из дискретного ряда его возможных структур.

Закон утверждает, что существует некоторый метод получения одной структуры, входящей в дискретный ряд, из другой структуры этого ряда. Кроме этого, в таком ряду должны существовать целевые структуры, позволяющие сделать гибридизацию целенаправленной

**Закон трансформации.** Трансформация одной структуры гибрида в другую может происходить только через общие для обеих структур знания. Закон описывает образование новых знаний их интерференцию между собой, построение умозаключений, а также возникновение и роль ассоциаций в трансформации и связывании образов и мыслей. Любые новые знания могут быть получены только путем перехода от одного знания к новому, взаимосвязанному, ассоциированному с предыдущим.

В соответствии с законом трансформации новая структура не может быть порождена как таковая, и возникает только на базе предыдущей структуры. При этом сохраняется достигнутая при старой структуре взаимная адаптация части компонентов, которая соответствует новой структуре. Закон трансформации показывает роль фундаментальных, формализованных знаний, которые постепенно развиваясь играют роль того общего, что связывает вновь появляющееся эвристическое знание.

# Основные методы гибридизации

Модуль 3 - Лекция 17, раздел "Основные методы гибридизации"​

**Метод H1.** Метод основан на том, что одна и также задача может быть решена несколькими известными автономными методами, а вычисления, начиная с некоторого пункта алгоритма, могут быть продолжены тем или иным методом.

В структуре алгоритма выделены четыре фрагмента. Фрагмент 1 выполняет предобработку исходных данных общую для всех используемых методов и готовит информацию для принятия решений о том, в соответствии с каким методом будут продолжаться вычисления во фрагменте 3. Принятие решения выполняет фрагмент 2 – гибридизатор, который должен располагать знаниями достаточными для принятия решений о том, как на очередном шаге вычислений сделать выбор метода для продолжения процесса решения задачи.

Фрагмент 4 выполняет функции оценки эффективности принятого гибридизатором решения, корректировки, в случае необходимости, знаний гибридизатора, продолжения решения и определения ситуации завершения работы алгоритма для выдачи результатов решения задачи.

Изображение выглядит как диаграмма, линия, снимок экрана, текст

Автоматически созданное описание

Принципиальной особенностью метода H1 является то, что решение принимает блок, встроенный в алгоритм решения задачи. Решение задачи продолжает один метод, а гибрид после завершения работы алгоритма перестает существовать. Это дает основание говорить о гибридном алгоритме решения задачи, а формируемая в ходе работы алгоритма последовательность применяемых методов может рассматриваться как гибридный метод решения задачи.

**Метод H2.**

Отличие от метода H1 данный подход предполагает, что **метод решения исходной неоднородной задачи неизвестен**, однако она предварительно была редуцирована на некоторое количество однородных задач, для каждой из которых известно некоторое множество методов ее решения. Есть также знания об областях релевантности автономных методов. В этом случае гибридизатор, используя декомпозицию неоднородной задачи, множество автономных моделей, знания об областях релевантности, а также знания ЛПР, строит алгоритм решения задачи как структуру над алгоритмами решения однородных задач, обрабатываемую интерпретатором. После получения интерпретатором решения задачи, оно анализируется ЛПР и принимается. В противном случае знания гибридизатора корректируются, может быть построен новый алгоритм, с выполнением повторного решения неоднородной задачи и так далее, до тех пор, пока итерационный процесс, по мнению ЛПР, может быть завершен и найдено решение исходной задачи.

При использовании рассматриваемого метода символьная структура, построенная гибридизатором, может рассматриваться как гибридная модель и как гибридный метод решения неоднородной задачи.

Изображение выглядит как диаграмма, снимок экрана, линия, Шрифт

Автоматически созданное описание

Особенность метода H2 – ориентация на неоднородные задачи, конструирование символьного эквивалента гибрида, который может быть сохранен, отредактирован и интерпретирован в любое время в зависимости от ситуации решения задачи. Важным моментом метода H2 является использование для решения задачи знаний ЛПР и возможность коррекции знаний гибридизатора.

Метод H2 может применяться в двух вариантах в зависимости от того, какие знания будут использоваться в гибридизаторе.

В первом варианте используются модели, построенные в полном соответствии с методом без каких-либо его изменений.

Во втором случае метод может быть изменен. Например, некоторая часть процедуры метода может быть заменена на процедуру, взятую из другого метода.

# Общий подход к построению гибридной интеллектуальной системы

Модуль 3 - Лекция 17, раздел "Общий подход к построению гибридной интеллектуальной системы"​

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

Анализируя приведенную модель, можно сделать вывод о том, что в ней можно выделить две информационные составляющие – информацию, поступающую от пользователя и различных уровней иерархии (In) и информацию, передаваемую ЛПР (IЭ).

…

# Принципы построения гибридных интеллектуальных систем

Модуль 3 - Лекция 17, раздел "Принципы построения гибридных интеллектуальных систем"​

Я всё.