

1. Цель изучения дисциплины. Объекты изучения. Области применения СОЗ. Исторический обзор развития ИИ.

Цель изучения - обучение студентов систематизированным представлениям о системах обработки знаний (СОЗ), методах и средствах разработки и использования искусственного интеллекта в современных информационных технологиях.

Объекты изучения - методы и средства разработки и использования искусственного интеллекта в современных информационных технологиях.

Область применения ИИ - доказательство теорем; игры; распознавание образов; поддержка принятия решений; адаптивное программирование; сочинение машинной музыки и стихов; обработка данных на естественном языке; обучающиеся нейронные сети и т. п.

Исторический обзор развития ИИ

В **1956** году Джон МакКарти организовал первую конференцию по ИИ (Дормутская конференция). На этой конференции впервые был озвучен термин «Искусственный интеллект».

В **1958** году Марвин Мински вместе с Джоном МакКарти основал в Массачусетском технологическом институте лабораторию искусственного интеллекта.

Тьюринг предложил специальный тест в качестве **способа распознать разумные машины**. В 1950-е гг. было сделано открытие о том, что компьютеры могут управлять символами также, как и числовыми данными. Это привело к разработке таких программ, как программа для доказательства теорем Logic Theorist (авторы Ньюэлл, Симон, Шоу) и программа General Problem Solver (авторы Ньюэлл, Симон).

Самым крупным открытием было создание в 1950-е годы программы для игры в шашки (Самуэль), которая обыграла своего создателя.

В **1950-е годы** был разработан язык искусственного интеллекта IPL (авторы Ньюэлл, Симон, Шоу) для программы Logic Theorist, который был языком обработки списков и предшественником языка LISP.

Язык LISP был разработан в лаборатории Массачусетского технологического института (MIT), тором которого был Джон МакКарти.

В **1960-е** произошел скачок в развитии ИИ, вызванный прогрессом в компьютерных технологиях.

В **1960-е** годы Джон МакКарти основал лабораторию ИИ в Стэнфордском университете, в которой был создан **первый робот**, способный перемещаться по искусственному миру и выполнять простые приказания.

Исследовательским полигоном для развития методов ИИ на первом этапе явились всевозможные игры, головоломки, математические задачи

1970-е годы характеризуются **резким спадом интереса к ИИ**, так как практическое применение ИИ было минимальным, возникли проблемы с финансированием исследований в США и Великобритании.

Дуг Ленет из Стэнфордского университета создал **программу «Автоматический математик»** и позднее – EURISKO, что позволило открыть новые теоремы в математике.

Впервые была применена нечеткая логика Лотфи Заде для управления паровым двигателем.

В **1970-е** годы во Франции был разработан язык логического программирования **ПРОЛОГ** и появились компьютерные программы с ИИ.

Продажи аппаратных и программных средств, связанных с ИИ, превысили 400 млн. долларов.

В **1986** году большую часть этого дохода принесли экспертные системы (ЭС), написанные на языке LISP, и соответственно, LISP-машины, которые развивались, становились лучше и дешевле.

Нейронные сети в 1980-е годы пережили возрождение и нашли свое применение для распознавания речи и в самообучении машин.

Спад был обусловлен сбоями в работе ЭС, а подъем был обусловлен успехами распознавания речи в PMB.

1990-е стали новой эпохой в развитии ИИ. Было обнаружено, что решение ряда задач с использованием ИИ стало более эффективным, чем традиционными методами.

Расширяются приложения ИИ: системы распознавания фальшивых кредитных карточек; системы классификации; системы распознавания изображений; системы автоматического планирования; системы предсказания прибыли и потребности в персонале; «добыча» («раскопка») знаний из баз данных, системы персонализации и т.п.

Философские, моральные и социальные аспекты

- Возможно ли создать мыслящую машину, в то время как мы и сами до конца не понимаем, что есть процесс мышления?
- Как мы поймем, что машина разумна?
- Если она поступает разумно, значит ли это, что у нее есть сознание?
- Если машина сочиняет стихи или музыку, то кто является автором?

2. Классификация интеллектуальных систем, системы с интеллектуальным интерфейсом

Принципы ИС

- Развитые коммуникативные способности
- Умение решать сложные, плохо формализуемые задачи
- Способность к самообучению
- Адаптивность

Классификация ИС второго типа

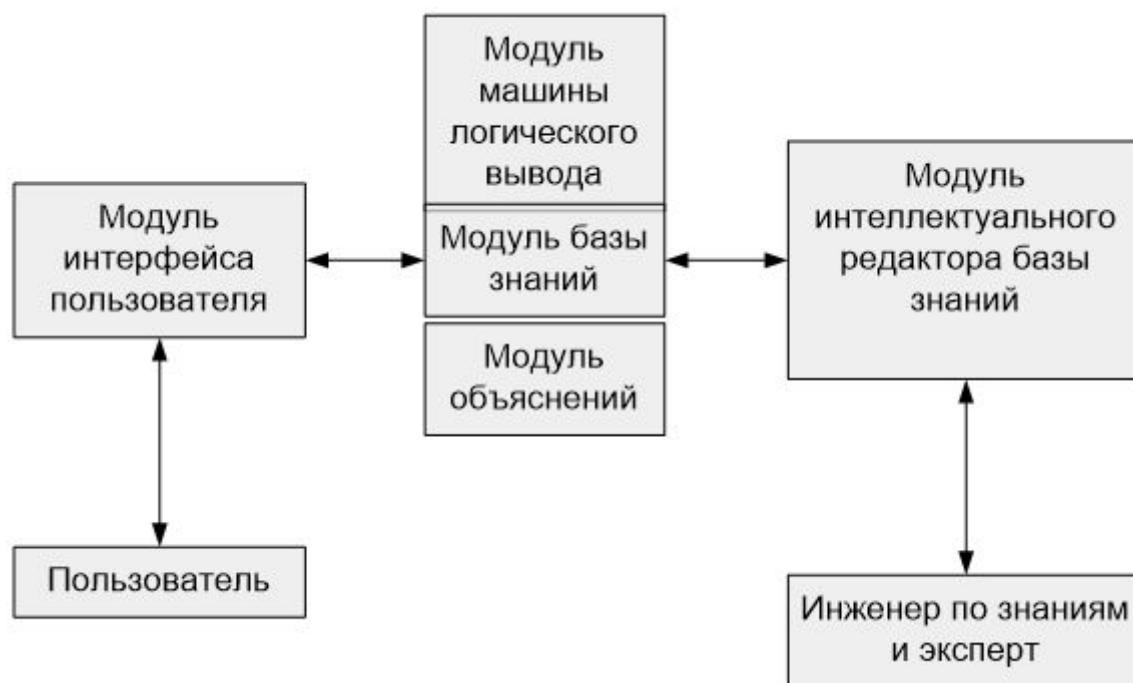
- Системы с интеллектуальным интерфейсом:
 - Интеллектуальные БД - выборка информации, не представленной в явной форме
 - Естественнo-языковой интерфейс - доступ к ИБД, контекстный поиск, голосовой ввод
 - Гипертекстовые системы - распространение знаний, приложения для документации
 - Системы контекстной помощи - частный случай гипертекстовой системы; поиск относящихся к ситуации рекомендаций
 - Когнитивная графика - общение с пользователем ИС с помощью графических образов; системы мониторинга, обучающие тренажеры
- Экспертные системы:
 - Классифицирующие
 - Действующие
 - Трансформирующие
 - Мультиагентные
- Самообучающиеся системы:
 - Индуктивные
 - Нейронные сети
 - Системы на прецедентах
 - Информационные хранилища
- Адаптивные системы:
 - CASE-технологии
 - Компонентные технологии

3. Экспертные системы. Экспертные системы: классификация и структура. Определение экспертной системы (ЭС) и инженерии знаний. Структура ЭС и назначение основных узлов. Классификация ЭС: по решаемой задаче, по связи с реальным временем, по типу ЭВМ, по степени интеграции.

Экспертная система - это сложный программный комплекс, аккумулирующий знания специалистов в конкретных предметных областях и тиражирующий этот эмпирический опыт для консультации менее квалифицированных пользователей.

Инженерия знаний - это процесс организации знаний в базу знаний и построения экспертной системы.

Структура ЭС



- **Пользователь** - специалист, для которого предназначена система
- **Инженер по знаниям** - специалист по ИИ, выступающий в роли промежуточного буфера между экспертом и БЗ
- **Модуль интерфейса пользователя** - программа, реализующая диалог пользователя с ЭС как на стадии ввода информации, так и получения результатов

- **Модуль базы знаний** - ядро ЭС, совокупность знаний предметной области, записанная на машинный носитель в форме, понятной эксперту и пользователю
- **Модуль машины логического вывода** - программа, моделирующая ход рассуждения эксперта на основании знаний, имеющихся в БЗ
- **Модуль объяснений** - программа, позволяющая пользователю получить ответы на вопросы: "Как была получена та или иная рекомендация?"
- **Модуль интеллектуального редактора БЗ** - программа, представляющая инженеру по знаниям возможность создавать и редактировать БЗ в диалоговом режиме

Классификация Гавриловой

- По типу решаемой задачи:
 - Интерпретация
 - Диагностика
 - Мониторинг
 - Проектирование
 - Прогнозирование
 - Обучения
 - Управление
- По связи с реальным временем:
 - Статические. БЗ и интерпретируемые данные не меняются во времени
 - Квазидинамические. Интерпретируют ситуацию, которая меняется с некоторым фиксированным интервалом времени
 - Динамические. Непрерывная интерпретация поступающих в систему данных
- По типу используемой ЭВМ:
 - СуперЭВМ
 - Мейнфреймы
 - Рабочие станции
 - Мини ЭВМ
 - ПК
 - КПК
- По степени интеграции:
 - Автономные
 - Интегрированные. Сложные программные комплексы

4. Уровни разработки экспертных систем. Этапы разработки: идентификации, концептуализации, формализации, реализации, тестирования и сопровождения. Оценка качества ЭС со стороны пользователя, со стороны эксперта и со стороны разработчика

Уровень проектирования ЭС

Уровень проектирования ЭС	Характеристика уровня
Демонстрационный прототип (50-100 правил)	Система решает часть поставленной задачи, демонстрируя целесообразность разработки ЭС
Исследовательский прототип (200-500 правил)	Система удовлетворительно решает всю задачу, но может быть ненадежной вследствие неполного тестирования
Действующий прототип (500-1000 правил)	Система проявляет высокое качество решения задачи при достаточной надежности, но для сложных задач получение решения может быть недостаточно эффективным по времени и (или) используемой памяти
Промышленная система (1000-1500 правил)	Система показывает высокое качество, надежность и эффективность работы в реальных условиях эксплуатации
Коммерческая система (1000-3000 правил)	Промышленная система, пригодная не только для использования внутри организации - заказчика, но и для продажи другим потребителям

Этапы разработки

Этап	Описание
Идентификация	Эксперты и инженеры-когнитологи. Определение участников процесса проектирования и распределение ролей между ними, неформальное описание задачи и оценка целесообразности разработки ЭС, определение целей разработки и необходимых для нее ресурсов
Концептуализация	Решение задачи выявления знаний, относящихся к процессу решения поставленной задачи

Формализация	Осуществляется выработка ключевых понятий и отношений инженером-когнитологом, который выражает знания формально на некотором языке, определяет пригодность доступных инструментальных средств для решения данной задачи и принимает решение о выборе подходящего средства
Реализация	Разработчики, используя выбранные инструментальные средства, воплощают формальную модель в программе для ЭВМ. Разработка текстовой документации
Тестирование	Оценка качества работы прототипа и выработка рекомендаций по его модификации
Сопровождение	Обеспечение контроля и повышения качества, развитие функциональных возможностей ЭС

5. Машина логического вывода. Принцип работы интерпретатора. Конфликтное множество. Способы управления выводом. Понятие машины логического вывода. Функции управляющего компонента и компонента вывода. Применение правила *modus ponens*. Конфликтное множество. Способы разрешения конфликтов

Функции машины логического вывода

- Просмотр существующих фактов из рабочей памяти (базы данных) и правил из базы знаний, а также добавление в рабочую память новых фактов
- Определение порядка просмотра и выполнение правил

Функции компонента вывода

- Выполнение вывода на основе правила *modus ponens*

Функции управляющего компонента

- Сопоставление - сопоставление образца правила с имеющимися фактами
- Выбор - в случае конфликтной ситуации выбрать одно правило согласно критерию разрешения конфликтов
- Срабатывание - если образец правила совпал с фактами из рабочей памяти, то правило срабатывает
- Действие - в рабочую память добавляется заключение сработавшего правила

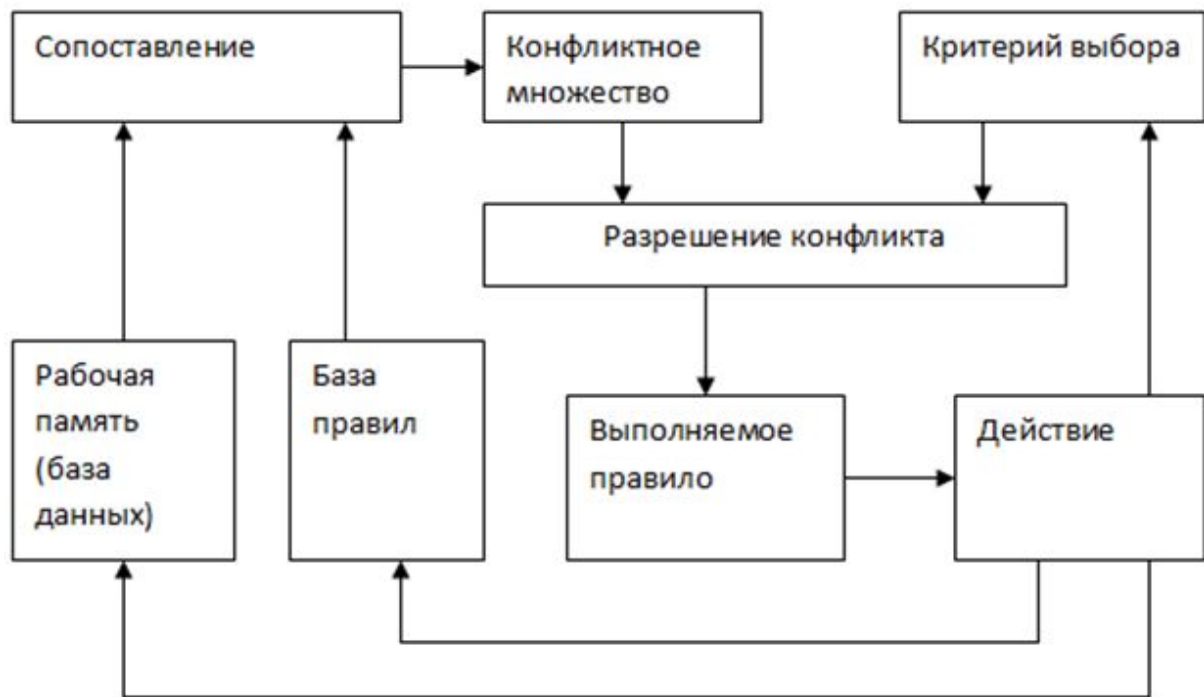
Правило *modus ponens*

Если истинно утверждение A и истинно утверждение $A \rightarrow B$, то B тоже истинно

Цикл работы интерпретатора

Интерпретатор продукций работает циклически. В каждом цикле он просматривает все правила, чтобы выявить те посылки, которые совпадают с известными и истинными на данный момент фактами. После выбора правило срабатывает, его заключение заносится в рабочую память, а затем цикл повторяется сначала.

В одном цикле может сработать только одно правило. Если несколько правил успешно сопоставлены с фактами, то такая ситуация называется конфликтной. Совокупность отобранных правил составляет так называемое **конфликтное множество**. Интерпретатор выполняет **разрешение конфликтов** путём выбора по определённому критерию единственного правила, которое срабатывает в данном цикле.



Свойства механизмов разрешения конфликтов

- **Разнообразие.** Не следует применять к одним и тем же данным правило, которое уже было к ним применено ранее.
- **Специфика.** Более специфичные правила, которые включают большее количество компонентов в предпосылках, имеют приоритет перед более общими.
- **Новизна.** Идея состоит в том, чтобы следовать за "текущей волной" и меньше внимания уделять тем данным, которые были давно сформированы.

6. Стратегии логического вывода, применяемые в продукционных ЭС. Понятие прямого и обратного вывода в глубину и в ширину. Демонстрация на примере продукционных правил. Рекомендации применения стратегий вывода.

В системах с **прямым выводом** по известным фактам отыскивается заключение, которое из этих фактов следует. Если такое заключение удаётся найти, то оно заносится в рабочую память. Прямой вывод часто называют выводом, управляемым данными, или выводом, управляемым антецедентами (т. е. предпосылками).

Например, имеется фрагмент базы знаний из двух правил.

Правило 1 – ЕСЛИ «отдых – летом» И «человек – активный», ТО «ехать – в горы».

Правило 2 – ЕСЛИ «любит – солнце», ТО «отдых – летом».

Предположим, в систему введены истинные факты – «любит – солнце» И «человек – активный».

Прямой вывод – исходя из фактических данных позволяет получить рекомендацию.

Первый проход

Шаг 1. Попробуем правило 1, не работает (не хватает данных «отдых – летом»).

Шаг 2. Попробуем правило 2, работает, в базу поступает факт «отдых – летом».

Второй проход

Шаг 3. Попробуем правило 1, работает, активизируется цель «ехать – в горы», которая и выступает как совет, который дает экспертная система.

При **обратном методе вывода** вначале выдвигается некоторая гипотеза, а затем механизм вывода как бы возвращается назад, переходя к фактам, пытаясь найти те факты, которые подтверждают гипотезу (правая часть правила). Если она оказалась правильной, то выбирается следующая гипотеза, детализирующая первую и являющаяся к ней подцелью. Далее отыскиваются факты, подтверждающие истинность подчиненной гипотезы. Вывод такого типа называется управляемым целями, или управляемым консеквентами (т. е. следствиями). **Обратный вывод применяется в тех случаях, когда цели известны и их сравнительно немного.**

Пример

Обратный вывод позволяет подтвердить выбранную цель при помощи имеющихся правил и данных.

Первый проход

Шаг 1. Цель – «ехать – в горы»: попробуем правило 1 – не хватает данных «отдых – летом», они становятся новой целью и выполняется поиск правила, у которого цель находится в левой части.

Шаг 2. Цель – «отдых – летом»: правило 2 подтверждает цель и активирует её.

Второй проход

Шаг 3. Попробуем правило 1, подтверждается искомая цель.

При «**поиске в глубину**» в качестве очередной подцели выбирается та, которая соответствует следующему, более детальному уровню описания задачи. Например, диагностирующая система, сделав на основе известных симптомов предложение о наличии определённого заболевания, будет продолжать запрашивать уточняющие признаки и симптомы этой болезни до тех пор, пока полностью не опровергнет выдвинутую гипотезу.

При «**поиске в ширину**», напротив, система вначале проанализирует все симптомы, находящиеся на одном уровне пространства состояний, даже если они относятся к разным заболеваниям, и лишь затем перейдёт к симптомам следующего уровня детальности.

Рекомендации

В системах, база знаний которых насчитывается сотни правил, желательным является использование стратегии управления выводом, позволяющей минимизировать время поиска решения и тем самым повысить эффективность вывода. К числу таких стратегий относятся: «поиск в глубину», «поиск в ширину», разбиение на подзадачи и альфа-бета алгоритм.

7. Основы теории приближенных рассуждений.

Причины непригодности вероятностного вывода

- Рассуждения на основе теории вероятностей становятся все более трудными и неудобными
- Трудно создать цепочку рассуждений с расчетом точных вероятностей
- Разработанные системы используют лишь грубые оценки с использованием вероятностей

Основные правила комбинирования

Эффективный способ решения проблемы учета неопределенностей – присвоить коэффициент определенности как посылке, так и всей импликации. Коэффициент определенности (ct) приблизительно эквивалентен вероятности того, что посылка является истинной.

$ct(\text{заключение}) = ct(\text{посылка}) * ct(\text{импликация})$

$ct(e1 \text{ И } e2) = \min [ct(e1), ct(e2)]$

$ct(e1 \text{ ИЛИ } e2) = ct1 + ct2 - ct1 * ct2$ - правило дополнения

Биполярные схемы для коэффициентов определенности

Коэффициент принимает значения от -1 до 1:

- -1 - высказывание абсолютно неверно
- 0 - у системы нет знаний
- 1 - система полностью уверена

Промежуточные величины отражают степень доверия или недоверия к указанным ситуациям.

Формулы для расчета коэффициентов:

$ct(HE\ c) = -ct(e)$

$ct(e1 \text{ И } e2) = \min [ct(e1), ct(e2)]$

$ct(e1 \text{ ИЛИ } e2) =$

- $ct1 + ct2 - ct1 * ct2$, $ct1 > 0$, $ct2 > 0$
- $ct1 + ct2 + ct1 * ct2$, $ct1 \leq 0$, $ct2 \leq 0$
- $(ct1 + ct2) / (1 - \min [|ct1|, |ct2|])$, $ct1 * ct2 < 0$

Обратимое правило (rev): может применяться к любому вероятностному значению, которое может быть связано с посылкой.

Необратимое правило (nrev): может применяться только к положительному значению посылки.

8. Данные и знания. Определение данных и знаний. Этапы преобразования данных и знаний в процессе компьютерной обработки. Классификация знаний.

Данные – это отдельные факты, характеризующие объекты, процессы и явления предметной области, а также их свойства.

Информация – это результат преобразования и анализа данных. Отличие информации от данных состоит в том, что данные – это фиксированные сведения о событиях и явлениях, которые хранятся на определенных носителях, а информация появляется в результате обработки данных при решении конкретных задач. Например, в базах данных хранятся различные данные, а по определенному запросу система управления базой данных выдает требуемую информацию.

Знания – это зафиксированная и проверенная практикой обработанная информация, которая использовалась и может многократно использоваться для принятия решений.

Знание – это совокупность сведений, образующих целостное описание, соответствующее определенному уровню осведомленности об описываемой проблеме.

Знания – это закономерности предметной области (принципы, связи, законы), полученные в результате практической деятельности и профессионального опыта, позволяющие специалистам ставить и решать задачи в этой области.

Формальные знания могут быть в виде документов (стандартов, нормативов), регламентирующих принятие решений или учебников, инструкций с описанием решения задач.

Неформальные знания – это знания и опыт специалистов в определенной предметной области.

Этапы обработки данных

- исходная форма существования данных (результаты наблюдений и измерений, таблицы, справочники, графики и т.п.)
- представление на специальных языках описания данных, предназначенных для ввода в ЭВМ
- базы данных на машинных носителях

При обработке на ЭВМ данные трансформируются, условно проходя следующие этапы:

1. D1 – данные как результат измерений и наблюдений
2. D2 – данные на материальных носителях информации (таблицы, протоколы, справочники)
3. D3 – модели (структуры) данных в виде диаграмм, графиков, функций
4. D4 – данные в компьютере на языке описания данных
5. D5 – базы данных на машинных носителях информации

Этапы обработки знаний

1. Z1 – знания в памяти человека как результат мышления

2. Z2 - материальные носители знаний (учебники, методические пособия)
3. Z3 - поле знаний - условное описание основных объектов предметной области, их атрибутов и закономерностей, их связывающих
4. Z4 - знания, описанные на языках представления знаний (продукционные языки, семантические сети, фреймы и другие модели)
5. Z5 - база знаний на машинных носителях информации

Классификация знаний

По степени обобщенности:

- **Поверхностные** знания описывают совокупности причинно-следственных отношений между отдельными понятиями предметной области
- **Глубинные** знания характеризуют абстракции, аналогии, образцы, которые отображают глубину понимания всех процессов, происходящих в предметной области

По степени отражения явлений

- **Жёсткие** знания позволяют получить однозначные четкие рекомендации при задании начальных условий
- **Мягкими** называют знания, которые используют для представления теорию нечетких множеств, нейронных сетей, вероятностных и приближенных рассуждений, генетических алгоритмов. «Мягкие» знания допускают множественные, расплывчатые решения и многовариантные рекомендации

По видам

- **Декларативные** знания - описания фактов и явлений, фиксируют наличие или отсутствие таких фактов. Декларативные знания описывают объекты, классы объектов и отношения. Под объектом-фактом понимается факт, который задается своим значением. Класс объектов - это имя, под которым объединяется конкретная совокупность объектов-фактов. Отношения определяют связи между классами объектов и отдельными объектами, возникшие в рамках предметной области.
- **Процедурные** знания - описания действий, которые возможны для манипулирования фактами при достижении поставленных целей (методики, правила и т.п.)

Модели представления знаний

- **Логические** модели представления знаний подразделяются на следующие:
 - знания на базе исчисления высказываний и предикатов
 - знания на основе псевдофизической логики
 - знания реляционного вида
- **Эвристические** модели позволяют моделировать рассуждения при отсутствии точных теоретических обоснований. Подразделяются на:
 - вероятностные
 - сетевые
 - фреймовые
 - семантические
 - продукционные
 - нечеткие
 - нейросетевые

9. Модели представления знаний в виде предикатов и продукционных моделей.

Определение предикатов первого порядка

Определение продукционных правил.

Достоинства и недостатки каждой модели представления знаний и примеры моделей представления знаний.

Логическая модель представления знаний (предикаты)

Логическая модель представления знаний основана на системе исчисления предикатов первого порядка.

Предикат - логическая функция от одной или нескольких нелогических переменных. Функция может принимать значения истина (t) или ложь (f). В рамках логики утверждение считается истинным, если относящееся к нему предположение считается истинным и заключение самого утверждения тоже истина.

Предикат первого порядка - предикат, аргументами которого не могут быть другие предикаты или функции.

Синтаксис языка предикатов включает: предикативные символы, символы переменных, константы, а также разделители (), [], “, ‘.

Предикативные символы используются для обозначения отношений. Объекты отношений записываются в круглые скобки после предикативного имени и называются аргументами. Полная запись отношения называется **атомарной формулой**. Например, атомарная формула «Является (Иванов, специалист по компьютерам)» содержит два предикативных термина.

Термы могут быть константами и переменными. Разрешено также в качестве термов использовать функции, которые обязательно должны быть определены в рамках предметной области. Разработчик ЭС заранее определяет, как интерпретировать порядок термов в отношении.

Правильно построенная функция (ППФ) - допустимое выражение в исчислении предикатов, в частности атомарные формулы. В языке предикатов для каждой ППФ обязательно определяется конкретная интерпретация. Как только для ППФ определена интерпретация, то формула имеет значение «истина», если соответствующее утверждение истинно; в противном случае ППФ имеет значение «ложь».

Из формул можно составить предложение с помощью логических связок: конъюнкция, дизъюнкция, импликация, отрицание.

Квантор - это операция, в которой участвуют все значения переменной одного предиката.

Различают квантор общности \forall и квантор существования \exists .

Если перед предикатом записан квантор \forall для какой-то переменной, например $\forall(x)$, то говорят о том, что значение предиката будет истинным только в том случае, если все значения переменной x будут истинными

Достоинства:

- Наличие теоретического материала по методам логического вывода
- Наличие стандартной типовой процедуры логического вывода (доказательства теорем)

Недостатки:

- Невозможность выразить через переменные другие предикаты
- Сложность логического вывода при больших массивах данных
- Сложность использования эвристик в процессе логического вывода, монотонность логического вывода
- Возможность комбинаторного взрыва, слабая структурированность описаний

Пример: язык Prolog

Продукционная модель

Продукционная модель позволяет представить знания в виде предложений типа "ЕСЛИ (условие), ТО (действие)".

Под **«условием»** (антецедентом) понимается некоторое предложение - образец, по которому осуществляется поиск в базе знаний, а под **«действием»** (консеквентом) - действия, выполняемые при успешном исходе поиска. Они могут быть промежуточными, выступающими далее, как условия, и терминальными или целевыми, завершающими работу системы.

Чаще всего вывод на такой базе знаний бывает **прямой** (от данных к поиску цели) или **обратный** (от цели для ее подтверждения к данным). Существуют также системы с двунаправленными выводами.

Данные - это исходные факты, хранящиеся в базе фактов, на основании которых запускается машина вывода или интерпретатор правил, выбирающий правила из продукционной базы знаний.

Продукционная модель чаще всего применяется в промышленных экспертных системах.

Достоинства:

- Наглядность
- Модульность
- Простой механизм логического вывода

Недостатки:

- Неясность взаимных отношений правил
- Отличие от структур знаний, свойственных человеку
- Сложность оценки целостного образа знаний
- Низкая эффективность обработки знаний
- Возникновение конфликтных ситуаций и необходимость их разрешения
- При большом количестве правил возможность появления противоречивых правил, что требует разработки специальных механизмов контроля вводимых и редактируемых правил

Примеры: язык OPS 5; "оболочки" или "пустые" ЭС: EXSYS Professional, Карра, ЭКСПЕРТ, ЭКО, инструментальные системы ПИЭС.

10. Модели представления знаний в виде фреймов. Определение фрейма. Структура фрейма. Способы представления слотов. Наследование. Фреймы-образцы и фреймы-экземпляры. Фреймы-сценарии

Достоинства и недостатки фреймовой модели представления знаний

Фреймовая модель основана на теории Минского и представляет систематизированную психологическую модель памяти человека и его сознания.

Фрейм – это абстрактный образ для представления некоего восприятия; структура данных для представления некоторого концептуального объекта.

Структура фрейма

- **Имя фрейма** служит для идентификации фрейма в системе и должно быть уникальным. Фрейм представляет собой совокупность слотов.
- **Имя слота.** Оно должно быть уникальным в пределах фрейма. В качестве имени слота может выступать произвольный текст.
- **Указатели наследования.** Показывают, какую информацию об атрибутах слотов фрейма верхнего уровня наследуют слоты с аналогичными именами в данном фрейме. В конкретных системах указатели наследования могут быть организованы различными способами:
 - **U(Unique)** – значение слота не наследуется;
 - **S(Same)** – значение слота наследуется;
 - **R(Range)** – значение слота должны находиться в пределах интервала значений, указанных в одноименном слоте родительского фрейма;
 - **O(Override)** – при отсутствии значений в текущем слоте оно наследуется из фрейма верхнего уровня. Однако в случае определения значения текущего слота оно может быть уникальным. Этот тип указателя выполняет одновременно функции указателя U и S.
- **Указатель типа данных.** Он показывает тип значения слота:
 - Frame – указатель на фрейм
 - Real – вещественное число
 - Integer – целое число
 - Boolean – логический тип
 - Text – фрагмент текста
 - List – список
 - Table – таблица
 - Expression – выражение
 - Lisp – связанная процедура и т. п.

- **Значение слота.** Оно должно соответствовать типу данных и условию наследования.
- **Демоны.** Демоном называется процедура, автоматически запускаемая при выполнении некоторого условия. Например: IF-NEEDED, IF-ADDED, IF-REMOVED
- **Присоединенная процедура.** В качестве значения слота может использоваться процедура. Она запускается по сообщению, переданному из другого фрейма.

Различают **фреймы-образцы** (фреймы-прототипы), хранящиеся в базе знаний и **фреймы-экземпляры** (фреймы-примеры), которые создаются для отображения реальных ситуаций на основе поступающих данных.

Примеры фреймов-сценариев: банкротство, собрание акционеров и т. п.

Достоинства:

- Каждый объект описывается перечнем свойств
- Отражает концептуальную основу организации памяти человека
- Наглядность описания
- Гибкость

Недостатки:

- В зависимости от эксперта будет разный перечень свойств
- Специфика программирования (сложно запрограммировать)
- Сложности при расширении

11. Модели представления знаний в виде семантических моделей. Определение семантических сетей. Виды семантических сетей. Виды отношений. Достоинства и недостатки семантической модели представления знаний.

Семантическая сеть - это ориентированный граф, вершины которого - понятия, а дуги - отношения между ними.

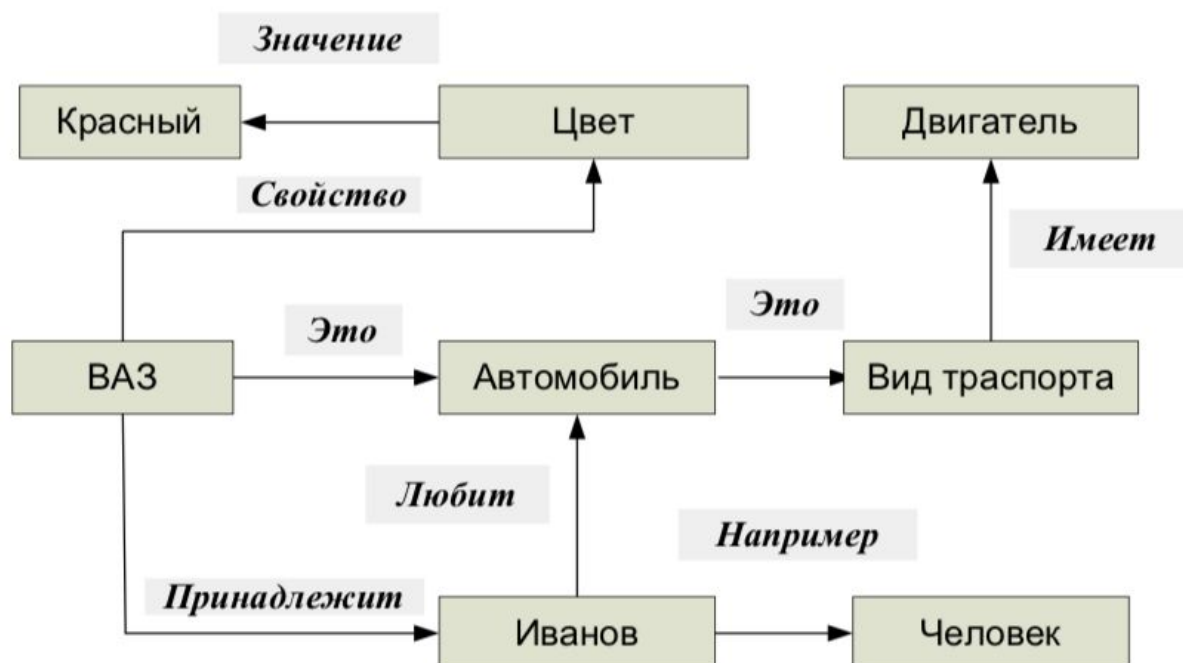
В качестве понятий обычно выступают абстрактные или конкретные объекты, а отношения - это связи типа: «это», «имеет частью», «принадлежит», «любит». Характерной особенностью семантических сетей является обязательное наличие трех типов отношений:

- класс – элемент класса (цветок - роза)
- свойство – значение (цветок - желтый)
- пример элемента класса (роза - чайная)

Семантическая сеть называется **однородной**, если содержит только один тип отношений, и **неоднородной** – в противном случае.

По типам отношений семантическая сеть подразделяется на **бинарную** (связывающую два объекта) и **N-арную** (для связи нескольких объектов).

Пример семантической сети



Виды отношений

- связи типа «часть-целое», «класс-подкласс», «элемент-множество»

- функциональные связи, определяемые глаголами «производит», «влияет»
- количественные (больше, меньше, равно)
- пространственные (далеко от, близко от, над, под и т.п.)
- временные (раньше, позже и т.п.)
- атрибутивные (иметь свойство, иметь значение)
- логические связи (И, ИЛИ, НЕ) и др.

Поиск решения в базе знаний, представленной в виде семантической сети, сводится к задаче поиска подграфа.

Достоинства:

- Универсальность, достигаемая за счет выбора соответствующего набора отношений
- Наглядность системы знаний, представленной графически
- Близость структуры сети, представляющей систему знаний, семантической структуре фраз на естественном языке
- Соответствие современным представлениям об организации долговременной памяти человека

Недостатки:

- Сетевая модель не содержит ясного представления о структуре предметной области, которая ей соответствует, поэтому формирование и модификация такой модели затруднительны
- Сетевые модели представляют собой пассивные структуры, для обработки которых необходим специальный аппарат формального вывода и планирования

12. Модели представления знаний в виде нейронных сетей и нечетких множеств. Представление знаний нейронными сетями и нечеткими правилами. Дообучение и переобучение нейронной сети. Две формы представления знаний в нейронной сети. Пример представления знаний в виде нечетких правил. Достоинства и недостатки каждой модели представления знаний.

Нейронная модель представления знаний

Нейронные сети могут выступать в качестве модели представления знаний.

Во-первых, это обучающая выборка, которая представляет неявную базу знаний (дообучения нейросетевой экспертной системы).

Во-вторых, это синоптическая карта, сформированная по результатам выбора оптимальной архитектуры и обучения нейронной сети. Однако при дополнении обучающей выборки требуется дообучение или переобучение нейронной сети.

Дообучение сети не требует изменения архитектуры нейронной сети, а **переобучение** может привести к изменению архитектуры нейронной сети.

Достоинства:

- не требуется разрабатывать множество продукционных правил и реализовывать механизм разрешения конфликтов
- обучающее множество можно оптимизировать по количеству примеров

Недостатки:

- невозможность представления функциональной зависимости между входами и выходами обучающей выборки
- трудность формирования обучающей выборки;
- возникновение проблемы «стабильности-пластичности»;
- трудность определения оптимальной архитектуры нейронной сети

Представление знаний в виде нечетких правил

Нечеткое множество (fuzzy sets) – множество, элементы которого принадлежат ему в той или иной степени.

Нечеткая логика (fuzzy logic) – умозаключение с использованием нечетких множеств или нечетких правил.

Нечеткое правило (fuzzy rule) – условное высказывание вида "ЕСЛИ X есть A, ТО Y есть B", где A и B нечеткие множества. Нечеткое правило образует связь между нечеткими множествами.

Нечеткая система (fuzzy system) – множество нечетких правил, преобразующих входные данные в выходные.

В нечеткой логике основополагающим является понятие **лингвистической переменной**, значениями которой являются не числа, а слова естественного языка, называемые **термами**.

Характеристикой нечеткого множества выступает **функция принадлежности**. $MF_C(x)$ – степень принадлежности значения «х» к нечеткому множеству С, представляющая собой обобщение понятия характеристической функции обычного множества.

Все системы с нечеткой логикой функционируют по одному принципу:

1. показания измерительных приборов фаззифицируются (переводятся в нечеткий формат)
2. обрабатываются
3. дефаззифицируются
4. в виде привычных сигналов подаются на исполнительные устройства.

Фаззификация – это сопоставление множества значений x ее функции принадлежности $M(x)$, т. е. перевод значений x в нечеткий формат.

Дефаззификация представляет собой процесс, обратный фаззификации

Достоинства:

- включение в анализ качественные переменные;
- оперирование нечеткими входными данными;
- оперирование лингвистическими критериями;
- быстро моделирование сложные динамические системы и сравнивать их с заданной степенью точности;
- преодоление недостатков и ограничений существующих методов оценки проектных рисков.

Недостатки:

- существует субъективность в выборе функций принадлежности и формировании правил нечеткого ввода;
- отсутствие информированности о методе, а также незначительно внимание к применению метода профессиональными финансовыми учреждениями;
- необходимость специального программного обеспечения, а также специалистов, умеющих с ним работать.

13. Нечеткие когнитивные карты. Области применения

Нечеткие когнитивные карты (НKK) – нечеткий ориентированный граф с обратной связью, узлы которого являются нечеткими множествами, а направленные ребра графа представляют причинно-следственные связи между **концептами** и характеризуют степень влияния (вес) связываемых концептов.

Процесс формирования и использования НKK состоит из следующих шагов:

1. Определение списка концептов, которые могут характеризовать события, действия, величины или цели
2. Определение степени влияния между каждой парой концептов или задание функций принадлежности на каждом терме
3. Построение нечеткой когнитивной карты
4. Анализ и интерпретация нечеткой когнитивной карты

$F = (N, E, C, f)$, где

- $N = \{N_1, N_2, \dots, N_n\}$ множество из n концепций, представляющих собой узлы графа
- $E: (N_i, N_j) \rightarrow e_{ij}$ функция, которая каждой паре концепций (N_i, N_j) , ставит в соответствие значение e_{ij} , которое принадлежит промежутку $[-1; 1]$ и обозначает причинно следственные связи
- $C: N_i \rightarrow C_i$, функция активности концепций системы. Каждому узлу ставится в соответствие мера активности в момент времени t . Она может принимать значения от 0 (нет активности) до 1 (активен). $C(0)$ задает вектор начальных значений активности узлов. $C(t)$ – вектор состояний (активности) узлов на итерации t .

- $f: R \rightarrow L$ функция преобразования $C(t)$ в $C(t + 1)$, где $t \geq 0$

$C_i(t + 1) = f\left(\sum_{j=1}^n e_{ji} C_j(t)\right)$, функция f может быть дискретной (bivalent, trivalent) и непрерывной (сигмоида)

Результатом моделирования НKK является последовательность векторов состояний, которые определяют состояние моделируемой системы в последующих итерациях. Результаты моделирования позволяют анализировать значение каждой из концепции, представляющее собой степень значимости концепции через некоторое время. Разные результаты моделирования могут быть получены заданием разных векторов начальной инициализации.

Области применения

Используются во многих областях для моделирования динамических систем, так как обладают наглядностью, гибкостью отображения предметной области, возможностями абстрактного представления и элементами нечеткой логики (суждений).

Примеры использования нечетких когнитивных карт:

1. Выбор эффективного варианта инвестиционного проекта в условиях наличия неопределенности (невозможность предсказания инфляции, спроса, технических и ресурсных показателей самого проекта)

2. Автоматическое управление воротами плотины на гидроэлектростанциях
3. Упрощенное управление роботами
4. Наведение телекамер при трансляции спортивных событий
5. Замена экспертов при анализе работы биржи
6. Предотвращение нежелательных температурных флуктуаций в системах кондиционирования воздуха
7. Эффективное и стабильное управление автомобильными двигателями
8. Управление экономичной скоростью автомобилей
9. Оптимизированное планирование автобусных расписаний
10. Системы прогнозирования землетрясений
11. Медицина: диагностика рака и других заболеваний
12. Сочетание методов нечеткой логики и нейронных сетей
13. Распознавание рукописных символов

14. Нейросетевые экспертные системы. Примеры реализации нейронных экспертных систем. Определение нейросетевой экспертной системы (НЭС). Представление знаний в виде обучающего множества и синаптической карты. Проблемы выбора оптимальной архитектуры НЭС, ее обучения и дообучения. Примеры созданных НЭС.

Нейросетевая ЭС - экспертная система, реализованная на базе нейронных сетей.

В основе построения нейросетевых экспертных систем (НЭС) лежит принцип обучения нейронной сети на известных примерах с последующим тестированием по любому входному вектору.

Нейронные сети могут выступать в качестве модели представления знаний. **В первую очередь** это **обучающая выборка**, которая представляет неявную базу знаний (до обучения нейросетевой экспертной системы). **Во-вторых**, это **синаптическая карта**, сформированная по результатам выбора оптимальной архитектуры и обучения нейронной сети.

Однако при дополнении обучающей выборки требуется дообучение или переобучение нейронной сети.

Проблема обучения и дообучения НЭС

Дообучение сети не требует изменения архитектуры нейронной сети, а переобучение может привести к изменению архитектуры нейронной сети.

Проблемы выбора оптимальной архитектуры НЭС

Два подхода для выбора архитектуры:

1. Взять заведомо большее число слоев и нейронов, а потом их редуцировать.
2. Постепенно наращивать количество слоев и нейронов в них, производя обучение после каждого изменения в структуре сети. Дает наиболее оптимальную структуру нейронной сети, но занимает значительное время.

Примеры созданных НЭС:

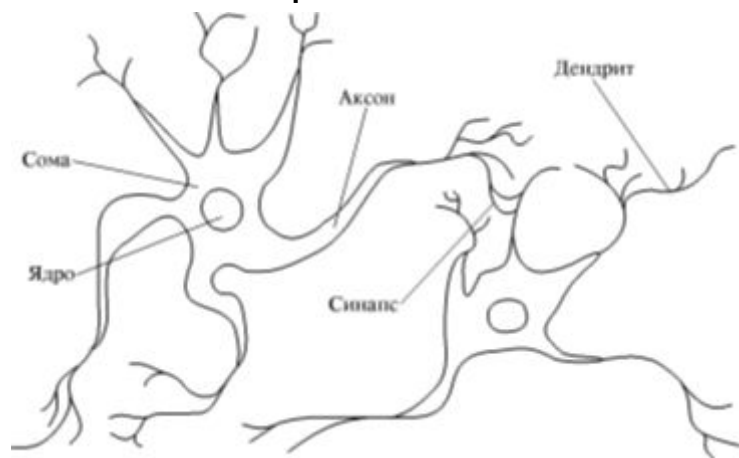
- В Италии разработана экспертная система для диагностики и лечения артериальной гипертензии. Система включает в себя три нейросетевых модуля, причем ответы одних являются входными данными для других.
- В Санкт-Петербурге разработана нейросетевая ЭС по диагностике некоторых классов болезней, которые плохо диагностируются врачами. Результаты проверки качества работы НЭС свидетельствуют о высокой достоверности результатов (94%).

15. Искусственная нейронная сеть.

Биологический нейрон и математическая модель искусственного нейрона и принцип его функционирования. Принципы обучения нейронных сетей: с учителем и без учителя. Слоистые и полносвязные нейронные сети. Виды активационных функций и их характеристики.

Искусственные нейронные сети (ИНС) — математические модели, а также их программные или аппаратные реализации, построенные по принципу организации и функционирования биологических нейронных сетей — сетей нервных клеток живого организма.

Биологический нейрон



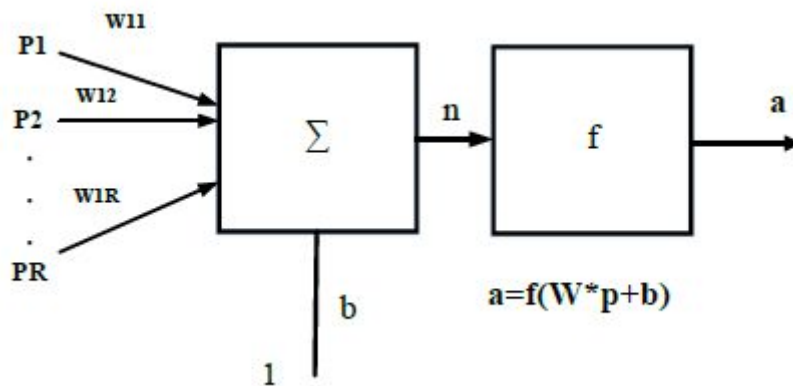
Дендриты - входы нейрона

Аксон - выход нейрона

Синапс - точка соединения аксона и дендрита

Нейрон состоит из тела и отростков нервных волокон двух типов – **дендритов**, по которым принимаются импульсы, и единственного **аксона**, по которому нейрон может передавать импульс. Тело нейрона включает **ядро**, которое содержит информацию о наследственных свойствах, и плазму, обладающую молекулярными средствами для производства необходимых нейрону материалов. Нейрон получает сигналы (импульсы) от аксонов других нейронов через дендриты (приемники) и передает сигналы, сгенерированные телом клетки, вдоль своего аксона (передатчик), который в конце разветвляется на волокна. На окончаниях этих волокон находятся специальные образования – **синапсы**, которые влияют на силу импульса.

Модель искусственного нейрона



P_1, P_2, \dots, P_R - входы (дендриты)

$W_{11}, W_{12}, \dots, W_{1R}$ - веса (синапсы)

a - выход (аксон)

Обучение с учителем предполагает наличие полного набора размеченных данных для тренировки модели на всех этапах ее построения. Наличие полностью размеченного датасета означает, что каждому примеру в обучающем наборе соответствует ответ, который алгоритм и должен получить. В основном обучение с учителем применяется для решения двух типов задач: классификации (предсказание дискретных значений - классов) и регрессии (работает с непрерывными данными, определяя по x ожидаемое y).

В **обучении без учителя** у модели есть неразмеченный набор данных. Нейронная сеть пытается самостоятельно найти корреляции в данных, извлекая полезные признаки и анализируя их. Решаемые задачи: кластеризация (разбиение на классы), обнаружение аномалий, ассоциации (рекомендации).

Слоистые и полносвязные нейронные сети

Неполносвязные нейронные сети описываются неполносвязным ориентированным графом и разделяются на **однослойные** и **многослойные (слоистые)** с прямыми, перекрестными и обратными связями.

В **многослойных** нейронных сетях нейроны объединяются в слои. **Слой** содержит совокупность нейронов с едиными входными сигналами. Число нейронов в слое может быть любым и не зависит от количества нейронов в других слоях. В сетях с **обратными связями** информация с последующих слоев передается на предыдущие слои.

В **полносвязных** нейронных сетях каждый нейрон передает свой выходной сигнал остальным нейронам, в том числе и самому себе. Все входные сигналы подаются всем нейронам. Выходными сигналами сети могут быть все или некоторые выходные сигналы нейронов после нескольких тактов функционирования сети.

Виды активационных функций

Название	Формула	Область значений
Пороговая	$f(s) = \begin{cases} 0, & s < \theta, \\ 1, & s \geq \theta \end{cases}$	0, 1
Знаковая (сигнатурная)	$f(s) = \begin{cases} 1, & s > 0, \\ -1, & s \leq 0 \end{cases}$	-1, 1
Сигмоидальная (логистическая)	$f(s) = \frac{1}{1 + e^{-s}}$	(0, 1)
Полулинейная	$f(s) = \begin{cases} s, & s > 0, \\ 0, & s \leq 0 \end{cases}$	$(0, \infty)$
Линейная	$f(s) = s$	$(-\infty, \infty)$
Радиальная базисная (гауссова)	$f(s) = \exp(-s^2)$	(0, 1)

16. Основы нейронных сетей (классификация НС, структура нейрона со смещением, функции активации нейронной сети, структура однослойной нейронной сети, проблема функции исключающее или, алгоритм обучения персептрона, общий алгоритм обучения НС)

Классификация

По структуре:

- Полносвязные
- Неполносвязные:
 - Однослойные
 - Многослойные:
 - С прямыми связями
 - С обратными связями
 - С перекрестными связями

По способу подачи информации на входы:

- подачу сигналов на синапсы входных нейронов
- подачу сигналов на выходы входных нейронов
- подачу сигналов в виде весов синапсов входных нейронов

По способу съема информации с выходов:

- съём с выходов выходных нейронов
- съём с синапсов выходных нейронов
- съём в виде весов синапсов выходных нейронов

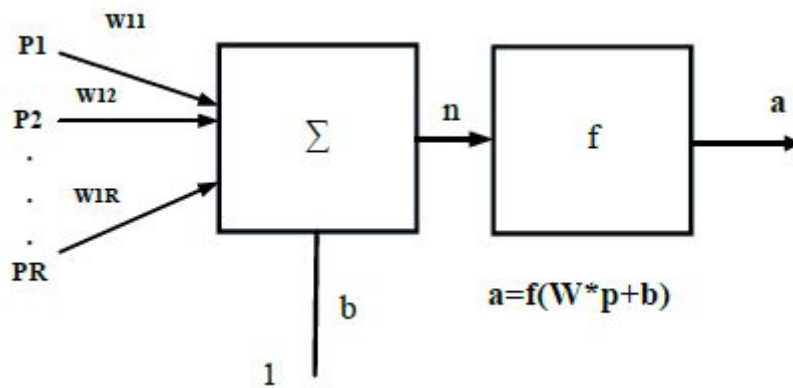
По типу связей:

- Без обратных связей
- С обратными связями

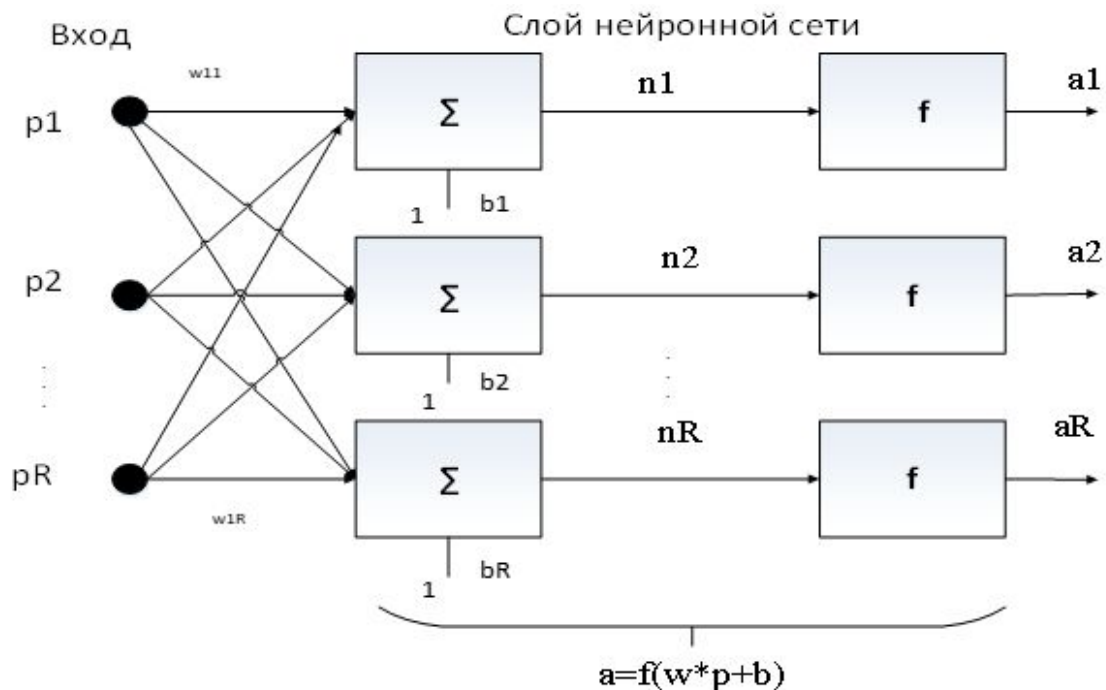
По принципу обучения:

- С учителем
- Без учителя

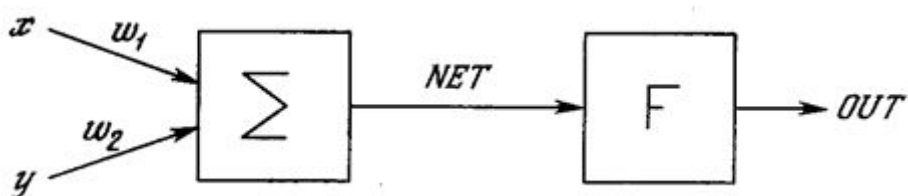
Структура нейрона со смещением



Структура однослойной нейронной сети



Проблема функции “исключающее или”

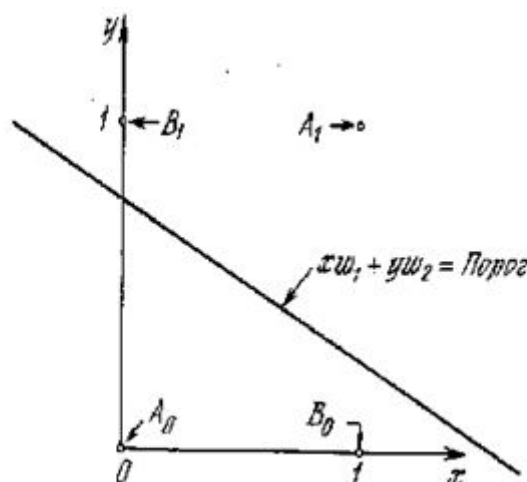


В сети на рисунке выше функция F является обычным порогом, так что OUT принимает значение ноль, когда NET меньше 0,5, и единица в случае, когда NET больше или равно 0,5. Нейрон выполняет следующее вычисление:

$$NET = x \cdot w_1 + y \cdot w_2$$

Никакая комбинация значений двух весов не может дать соотношения между входом и выходом, задаваемого таблицей. Это означает, что какие бы значения ни приписывались весам и порогу, сеть неспособна воспроизвести соотношение между входом и выходом, требуемое для представления функции ИСКЛЮЧАЮЩЕЕ ИЛИ.

Точки	Значения x	Значения y	Требуемый выход
A ₀	0	0	0
B ₀	1	0	1
B ₁	0	1	1
A ₁	1	1	0



Алгоритм обучения персептрона

Для обучения сети образ X подается на вход и вычисляется выход Y . Если Y правилен, то ничего не меняется. Однако если выход неправилен, то веса, присоединенные к входам, усиливающим ошибочный результат, модифицируются, чтобы уменьшить ошибку.

1. Подать входной образ и вычислить выходное значение Y
2. Рассмотреть следующие случаи:
 - а. Если выход правильный, то перейти на шаг 1
 - б. Если выход неправильный и равен нулю, то добавить все входы к соответствующим им весам;
 - с. Если выход неправильный и равен единице, то вычесть каждый вход из соответствующего ему веса.
3. Перейти на шаг 1

Общий алгоритм обучения НС

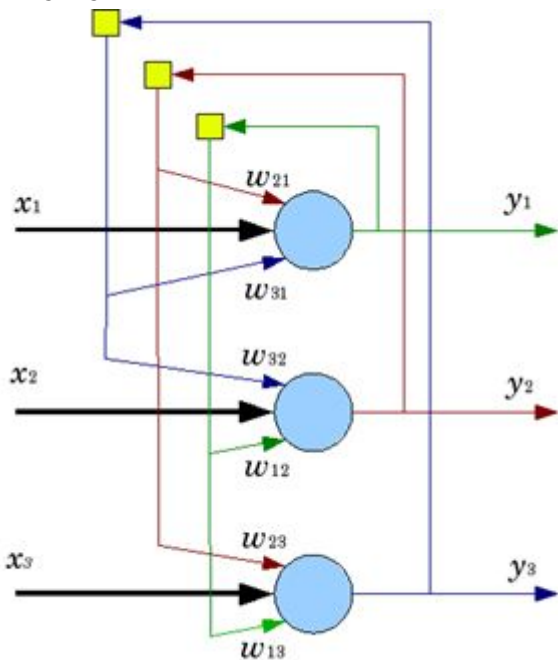
1. Задается скорость обучения α в диапазоне от 0 до 1 и минимальная среднеквадратичная ошибка сети E_m , которую необходимо достичь в процессе обучения
2. Случайным образом инициализируются веса синапсов W_{j1} и порог функции
3. Подается входной вектор и вычисляется выходное значение Y
4. Осуществляется изменение весов синапсов и порога функции
5. Алгоритм выполняется до тех пор, пока среднеквадратичная ошибка сети E не станет меньше заданной E_m .

17. Нейронная сеть Хопфилда как динамическая система. Структурная схема сети Хопфилда и алгоритм функционирования сети

НС Хопфилда является **релаксационной сетью с обратными связями**.

НС Хопфилда реализует существенное свойство автоассоциативной памяти – восстановление по искаженному (зашумленному) образцу ближайшего к нему эталонного. В этом случае входной вектор используется как начальное состояние сети, и далее сеть эволюционирует согласно своей динамике. Причем любой пример, находящийся в области притяжения хранимого образца, может быть использован как указатель для его восстановления. Выходной восстановленный образец формируется, когда сеть достигает равновесия. Такая сеть может быть использована как автоассоциативная память, как фильтр, а также для решения некоторых задач оптимизации. В отличие от многих нейронных сетей, работающих до получения ответа через определённое количество тактов, сети Хопфилда работают до достижения равновесия, когда следующее состояние сети в точности равно предыдущему.

Структурная схема сети Хопфилда



В процессе функционирования сети Хопфилда можно выделить **два режима**: обучения и классификации. В режиме **обучения** на основе известных векторов подбираются весовые коэффициенты сети. В режиме **классификации** при фиксированных значениях весов и вводе конкретного начального состояния нейронов возникает переходный процесс (см. формулу ниже), завершающийся в одном из локальных минимумов.

$$y_i(k) = \text{sgn}\left(\sum_{j=0}^n w_{ij} y_j(k-1)\right)$$

Алгоритм функционирования сети

1. На входы сети подается неизвестный сигнал. Фактически его ввод осуществляется непосредственной установкой значений аксонов: (в данном случае индекс в скобках указывает номер итерации), поэтому обозначение на схеме сети входных синапсов в явном виде носит чисто условный характер.
2. Вторым этапом работы заключается в смене состояния сети: рассчитывается новое состояние нейронов и новые значения аксонов:

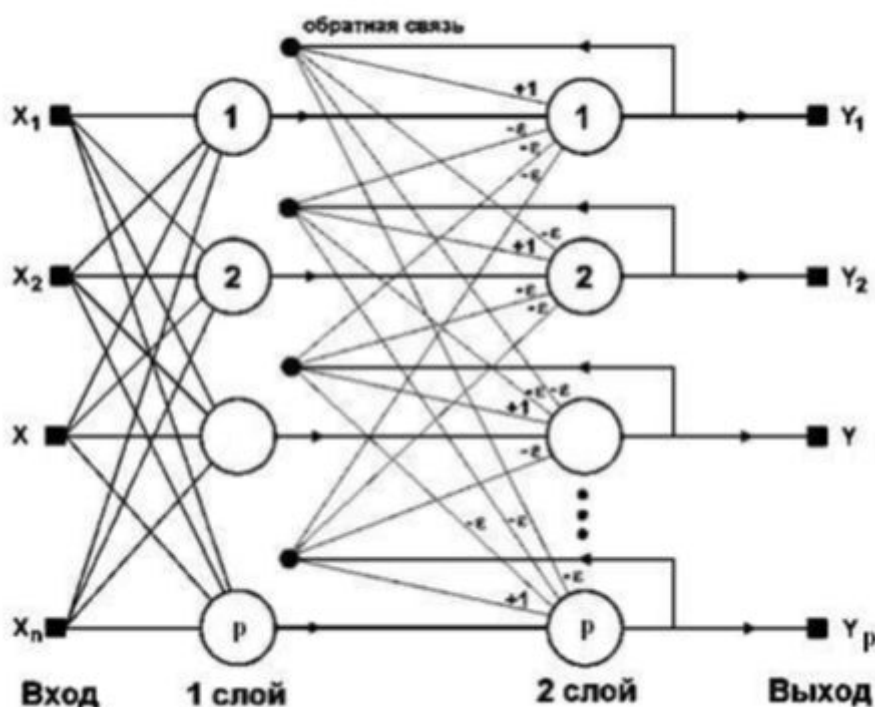
$$y_i(k) = \text{sgn}\left(\sum_{j=0}^n w_{ij} y_j(k-1)\right)$$

3. На третьем этапе выполняется проверка, изменились ли выходные значения аксонов за последнюю итерацию. Если да – переход к пункту 2, иначе (если выходы стабилизировались) – конец. При этом выходной вектор представляет собой образец, наилучшим образом (в смысле сети Хопфилда) сочетающийся с входными данными.

18. Нейронная сеть Хэмминга. Структурная схема сети Хемминга и алгоритм функционирования сети

Когда нет необходимости, чтобы сеть в явном виде выдавала образец, то есть достаточно, скажем, получать номер образца, ассоциативную память успешно реализует сеть Хэмминга. Данная сеть характеризуется, по сравнению с сетью Хопфилда, **меньшими затратами на память и объемом вычислений**.

Структурная схема сети Хемминга



Первый слой имеет однонаправленное распространение сигналов от входа к выходу и фиксированные значения весов. Нейроны **второго слоя** функционируют в режиме "победитель получает все". Количество нейронов в каждом слое сети равно количеству запоминаемых векторов.

Обучение сети Хемминга заключается в **предварительном расчете** значений весовых коэффициентов и порогов активации нейронов по следующим правилам. Весам первого слоя присваиваются значения, рассчитываемые по выражению:

$$w_{i,k} = \frac{1}{2} x_{i,k}, i = \overline{1, n}, k = \overline{1, p}$$

при этом пороги активационных функций равны:

$$T_k = \frac{n}{2}, k = \overline{1, p}$$

Как отмечалось ранее, веса связи i -го и j -го нейронов второго слоя равны:

$$w_{ij}^2 = \begin{cases} -\varepsilon, & i \neq j \\ 1, & i = j \end{cases}$$

$$0 < \varepsilon < \frac{1}{p}$$

После обучения сети по формулам, сеть способна распознавать вектора, подаваемые на её вход. На входы сети подается неизвестный вектор X , исходя из которого, рассчитываются состояния нейронов первого слоя. Значения выходных сигналов нейронов определяются по формуле:

$$y_j^1 = \sum_{i=1}^n w_{ji} x_i + T_j, j = \overline{1, m}$$

Эти сигналы становятся начальными состояниями нейронов второго слоя. Этот слой определяет "победителя", то есть нейрон, выходной сигнал которого близок к 1. Такой нейрон указывает на вектор образа с минимальным расстоянием Хэмминга до входного вектора X . Нейрон-победитель определяется итерационным процессом расчета состояний нейронов второго слоя за счет ослабления весами входных сигналов слоя:

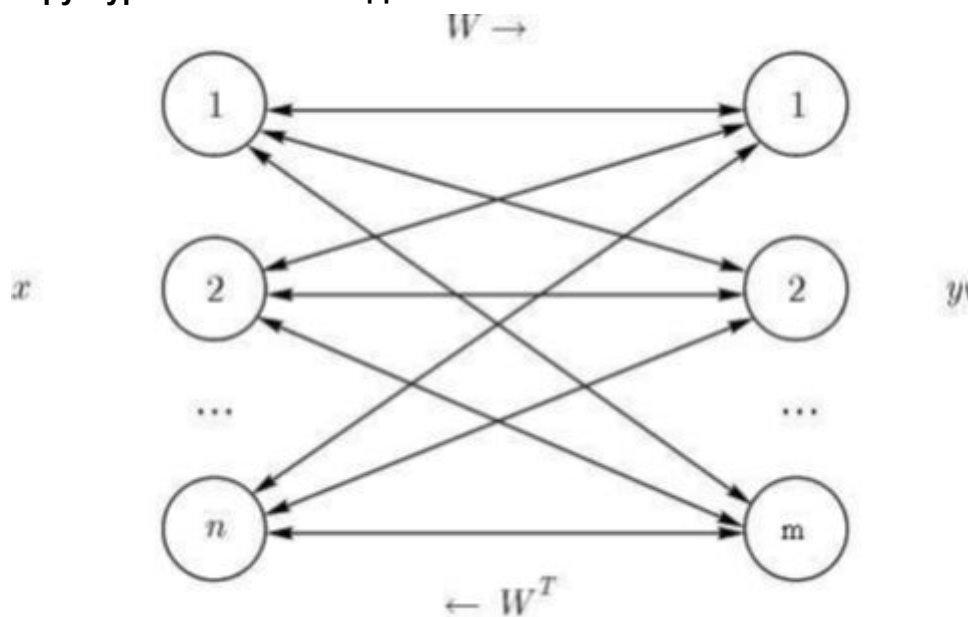
$$y_j^2(z+1) = f(y_j^2(z) - \varepsilon \sum_{k=1}^p y_k^2(z)), k \neq j, j = \overline{1, p}$$

Итерационный процесс во втором слое завершается, когда активным остается только один нейрон (его выход имеет значение близкое к 1), тогда как остальные нейроны пребывают в близком к нулю состоянии.

19. Двухнаправленная ассоциативная память (ДАП). Структурная схема сети ДАП и алгоритм функционирования сети. Достоинства ДАП.

Обобщением сети Хопфилда на случай двухслойной рекуррентной структуры, позволяющей кодировать множества пар взаимосвязанных векторов (X^k, Y^k) , считается двухнаправленное ассоциативное запоминающее устройство, называемое **Двухнаправленной ассоциативной памятью**. В общем случае размерности n и m соответственно векторов X и Y не совпадают. Сигналы в такой сети распространяются в двух направлениях. В первом цикле сигналы вначале проходят в одну сторону для задания состояний нейронов-получателей, то в следующем цикле эти нейроны сами становятся источниками, направляющими сигналы в обратную сторону. При этом выбор начального направления распространения сигналов не регламентирован и может произвольно выбираться пользователем.

Структурная схема сети ДАП



Обучение ДАП заключается в предварительном расчете матрицы весов W . Входные обучающие данные представляют собой множество пар биполярных векторов:

$$\left\{ (X^k = (x_1^k, x_2^k, \dots, x_n^k), Y^k = (y_1^k, y_2^k, \dots, y_m^k)), k = \overline{1, p} \right\}$$

На основе этого множества формируется матрица:

$$W = \sum_{k=1}^p (X^k)^T * Y^k$$

При прямом распространении сигналов веса описываются матрицей W , а при обратном — матрицей W^T .

Алгоритм функционирования сети

На вход сети подается вектор $X(0)$. Он обрабатывается матрицей весов W сети, в результате чего вырабатывается вектор выходных сигналов нейронов $Y(1)$. Вектор $Y(1)$ затем обрабатывается транспонированной матрицей W^T весов, которая вырабатывает новые выходные сигналы, представляющие собой новый входной вектор $X(1)$. Процесс повторяется до тех пор, пока сеть не достигнет стабильного состояния, в котором ни вектор $X(f)$, ни вектор $Y(f)$ не изменяются. Заметим, что нейроны в слоях 1 и 2 функционируют, как и в других парадигмах, вычисляя сумму взвешенных входов и вычисляя по ней значение функции активации F . Этот процесс в векторной форме может быть выражен следующим образом:

$$i = 0$$

$$Y(i+1) = F(X(i)W) \quad ,$$

$$X(i+1) = F(Y(i+1)W^T)$$

В результате формируются два стабильных вектора $X(f)$ и $Y(f)$

Достоинства

- Простота
- Быстрая сходимость процесса обучения и восстановления информации
- Структурная простота сети, позволяющая реализовать ее в виде СБИС и УБИС
- Совместимость с аналоговыми схемами и оптическими системами

20. Машина Больцмана. Алгоритм обучения

В основе сети Больцмана лежит **метод имитационного отжига** (управляемого охлаждения), который является разновидностью процедуры случайного поиска. В своей базовой форме машина Больцмана является сетью Хопфилда и их структуры полностью совпадают. Метод имитации отжига представляет собой алгоритмический аналог физического процесса управляемого охлаждения. Данный метод позволяет находить глобальный минимум функции нескольких переменных.

При затвердевании расплавленного металла его температура должна уменьшаться постепенно до момента полной кристаллизации. Если процесс остывания протекает слишком быстро, то образуются нерегулярности структуры металла, которые вызывают внутренние напряжения. В результате общее энергетическое состояние тела, зависящее от внутренней напряженности, остается более высоким, чем при медленном охлаждении. **Быстрая фиксация** энергетического состояния тела на уровне выше нормального соответствует сходимости оптимизационного алгоритма к точке **локального минимума**. Энергия состояния тела соответствует целевой функции, а абсолютный минимум – точке глобального минимума.

Алгоритм обучения

1. Определить переменную T , представляющую искусственную температуру. Придать T большое начальное значение.
2. Предъявить сети множество входов и вычислить выходы и целевую функцию.
3. Придать случайное изменение весу и пересчитать выход сети и изменение целевой функции в соответствии со сделанным изменением веса.
4. Если целевая функция уменьшилась (улучшилась), то сохранить изменение веса. Если изменение веса приводит к увеличению целевой функции, то вероятность сохранения этого изменения вычисляется с помощью распределения Больцмана:

$$P(c) = e^{\frac{-c}{kT}}$$

где $P(c)$ — вероятность изменения c в целевой функции;

k — константа, аналогичная константе Больцмана, выбираемая в зависимости от задачи;

T — искусственная температура.

Выбирается случайное число r из равномерного распределения от нуля до единицы. Если $r > 0$, то изменение сохраняется, в противном случае величина веса возвращается к предыдущему значению. Это позволяет системе делать случайный шаг в направлении, увеличивающем целевую функцию, и дает ей тем самым возможность вырваться из локальных минимумов, где любой малый шаг увеличивает целевую функцию.

Для завершения обучения машины Больцмана повторяются шаги 3 и 4 для каждого из весов сети, с постепенным уменьшением температуры T , пока не будет достигнуто допустимо низкое значение целевой функции. Сеть обучается на всех векторах обучающего множества, с возможным повторением, пока целевая функция не станет допустимой для всех обучающих векторов.

Машина Больцмана, учась на высокой температуре, ведет себя как случайная модель, а на низких температурах проявляет себя как детерминированная. Из-за случайной

компоненты в процессе обучения, нейрон может принять новое значение состояния, которое увеличивается быстрее, чем уменьшается общее пространство состояний. Имитация физического отжига позволяет продвигаться к глобальному минимуму, избегая локальный. Для достижения сходимости к глобальному минимуму энергии скорость уменьшения температуры должна быть обратно пропорциональна логарифму времени.

21. Математические основы алгоритма обратного распространения ошибки (ОРО). Укрупненный алгоритм ОРО. Прямой и обратный проход. Обучение с моментом

Математические основы алгоритма обратного распространения ошибки

Алгоритм обратного распространения ошибки (ОРО) — итеративный **градиентный алгоритм обучения**, который используется с целью **минимизации среднеквадратичного отклонения** текущего выхода и желаемого выхода многослойных НС. На каждый нейрон первого слоя подаются все элементы внешнего входного сигнала. Нейроны выполняют взвешенное суммирование элементов входных сигналов, прибавляя к сумме смещение нейрона. Над полученной суммой выполняется нелинейное преобразование активационной функцией. Значение функции активации и есть выход нейрона

Алгоритм ОРО

1. Задать шаг обучения α и требуемую среднеквадратичную ошибку E_m
2. Инициализация весов синапсов случайным образом малыми значениями
3. Случайным образом выбирается один пример из ОБ и подаётся на вход сети.
4. При этом для каждого примера выполняются следующие действия:
 - a. Производится фаза прямого распространения входного сигнала по нейронной сети
 - b. Затем фаза обратного распространения сигнала в результате которой определяются ошибки j для всех слоев сети по формуле (6).
 - c. Производится модификация весов по формуле (8)
5. Вычисляется суммарная среднеквадратичная ошибка НС:

$$E = \frac{1}{k} \sum_{i=1}^L \sum_{j=1}^P (y_j - d_j)^2$$

P - число нейронов в выходном слое

L - размер обучающей выборки

k - кол-во примеров в обучающей выборке

6. Если $E > E_m$, то переход к п. 3, иначе - конец.

Прямой и обратный проход

Прямой проход. Расчет весовых сумм S_i и активаторов u_i для каждого нейрона.

Обратный проход. Начиная с выходов, выполняется обратное движение через ячейки выходного и промежуточного слоя, при этом программа рассчитывает ошибки по формулам (1) и (2):

- для нейрона выходного слоя: $\delta_0 = (d_0 - y_0) * y_0 * (1 - y_0)$ (1)
- для всех нейронов скрытого слоя: $\delta_i = (w_{mj} * \delta_0) * y_i * (1 - y_i)$ (2)

Здесь m обозначает все нейроны, связанные со скрытым узлом, w – заданный вектор веса, y – выход активационной функции.

Веса в сети обновляются следующим образом по формуле (3) и (4):

- для весов синапсов между скрытым и выходным слоем: $w_{ij}^* = w_{ij} - \rho * \delta_0 * y_i$ (3)
- для весов синапсов между скрытым и входным слоем: $w_{ij}^* = w_{ij} - \rho * \delta_i * y_i$ (4)

Параметр ρ характеризует коэффициент скорости обучения $[0,1]$. Подбирается экспериментально.

Обучение с моментом

Избежать сходимости к локальному минимуму помогает обучение с моментом.

$$w_{ij}(t+1) = -\rho * \delta_i * y_i + \alpha * w_{ij}(t)$$

Первый член соответствует обычному методу градиентного спуска, а второй - методу тяжелого шарика или момента и не зависит от градиента.

Параметр α - **коэффициент инерционности**, влияние которого увеличивается с ростом особенно в окрестности локального минимума.

В результате возрастает значение целевой функции и обеспечивается выход за пределы локального минимума.

Однако показатель момента не должен доминировать в процессе обучения, т. к. это приведет к неустойчивости обучения (отсутствие сходимости алгоритма).

22. Алгоритм ОРО. Недостатки алгоритма обратного распространения ошибки. Адаптивный шаг обучения. Описание основных шагов алгоритма обучения (прямая и обратная волна). Выбор шага обучения.

Описание основных шагов алгоритма обучения

См. вопрос выше

Недостатки алгоритма

1. Инициализация влияет на качество и скорость обучения
2. Трудность определения оптимальной архитектуры НС (количество слоёв и нейронов в слое)
3. Сложность выбора параметра скорости обучения. При малых значениях увеличивается время обучения, а при больших можно попасть в локальный экстремум. Обычно берут $\eta = 0,1$.
4. Градиентный спуск не гарантирует глобального минимума
5. Случайный выбор первого примера из ОВ также сказывается на качестве обучения
6. В процессе обучения возможен «паралич» сети (точки на сигмоидах могут сместиться в область насыщения при больших значениях весов синапсов)
7. Возможно переобучение сети

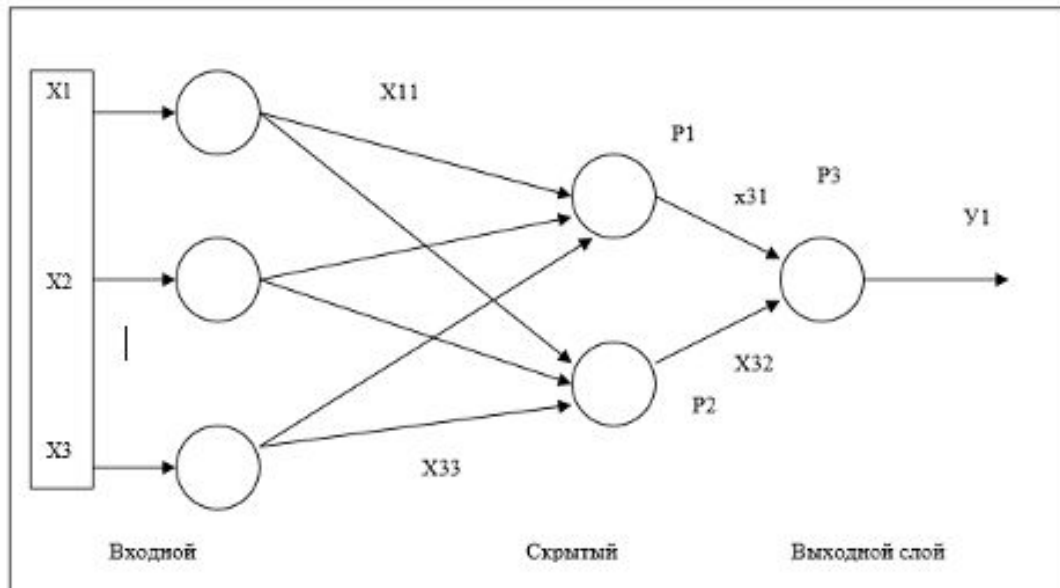
Адаптивный шаг обучения

$$w_{ij}(t+1) = -\eta \cdot \delta_i \cdot y_j + \alpha \cdot w_{ij}(t)$$

Если шаг α выбрать слишком маленьким, то процесс обучения затягивается. В противном случае процесс может не сойтись. Поэтому в некоторых работах предлагается выбор α проводить по формуле $\alpha = 1/K$, где K - номер итерации. В целом шаг обучения находится экспериментальным путем.

23. Принципы обучения нейронной сети с помощью генетических алгоритмов

Рассмотрим обучение нейронной сети на примере трехслойной НС



В каждом нейроне P1, P2, P3 активационная функция имеет вид:

$$F(sum) = \frac{1}{1 + e^{-b \cdot sum}}$$

$$sum = \sum_{i=1}^n x_i * w_i$$

где b - параметр функции активации, ответственный за ее крутизну;

x_i – значение на i-м входе;

w_i – вес i-го входа;

n – количество входов нейрона.

Таким образом, целевая функция будет представлять собой максимальное для набора примеров **относительное отклонение от эталонного значения**, выраженное в процентах. Чем меньше значение целевой функции, тем сеть лучше.

При обучении нейросетей **размер популяции выбирается достаточно большим**, как минимум 100 особей.

Хромосома в нашем случае будет представлять массив из 11 действительных чисел, по четыре на каждый скрытый нейрон и три для выходного нейрона, которые представляют собой веса соответствующих входов нейронов и параметры их функций активации:

w_{11}	w_{12}	w_{13}	b_1	w_{21}	w_{22}	w_{23}	b_2	w_{31}	w_{32}	b_3
----------	----------	----------	-------	----------	----------	----------	-------	----------	----------	-------

Для элементов хромосомы – генов – введем ограничения, обусловленные природой нейросетей: $-1 < w_{ij} < 1$ и $b_i > 0$.

В качестве **операторов скрещивания, отбора и редукции** выбираются стандартные генетические операторы. Дополнительно для оператора скрещивания можно установить вероятность применения 0,95 и использовать элитные хромосомы.

В качестве **оператора мутации** будем использовать случайное изменение значений весов и параметра функции активации для каждого нейрона на случайную величину. Вероятность мутации 0,01. Причем одновременно будет изменяться параметр функции только для одного нейрона, и для каждого будет изменяться один из входных весов. Какие конкретно веса и параметры будут меняться, определяется по равномерному закону. Например, мутация может быть следующей: значение b_2 увеличивается на 0,1; значение W_{11} уменьшается на 0,05; W_{21} – уменьшается на 0,01; W_{31} – увеличивается на 0,05.

Исходная популяция будет формироваться на основе равномерного распределения для каждого элемента хромосомы.

Целью работы генетического алгоритма является поиск значений весов для всех скрытых и выходных нейронов, а также параметра b_i их функций активации.

24. Нейронные сети адаптивного резонанса.

Проблема «пластичности-стабильности».

Принцип адаптивного резонанса. Архитектура сети ART-1. Функционирование сети ART в процессе классификации. Обучение сети ART-1

Сети адаптивного резонанса - разновидность искусственных нейронных сетей, основанная на теории адаптивного резонанса (ART) Стивена Гроссберга и Гейла Карпендера. Включает в себя модели обучения с учителем и без учителя, которые используются при решении задач распознавания образов и предсказания.

Проблема «пластичности-стабильности»

Нейронная сеть должна каждый раз реагировать на незнакомый образ (**быть пластичной**) и не «забывать» старые образы (**быть стабильной**). Проблема стабильности-пластичности является одной из самых сложных и трудно решаемых задач при построении искусственных систем, моделирующих восприятие.

Принцип адаптивного резонанса

Привлекательной особенностью нейронных сетей с адаптивным резонансом является то, что они сохраняют пластичность при запоминании новых образов, и, в то же время, предотвращают модификацию старой памяти.

Нейросеть имеет внутренний детектор новизны - тест на сравнение предъявленного образа с содержимым памяти. При удачном поиске в памяти предъявленный образ классифицируется с одновременной уточняющей модификацией синаптических весов нейрона, выполнившего классификацию. Такую ситуацию называют **возникновением адаптивного резонанса** в сети в ответ на предъявление образа. Если резонанс не возникает в пределах некоторого заданного порогового уровня, то тест новизны считается успешным и образ воспринимается сетью как новый. Модификация весов нейронов, не испытавших резонанса, при этом не производится.

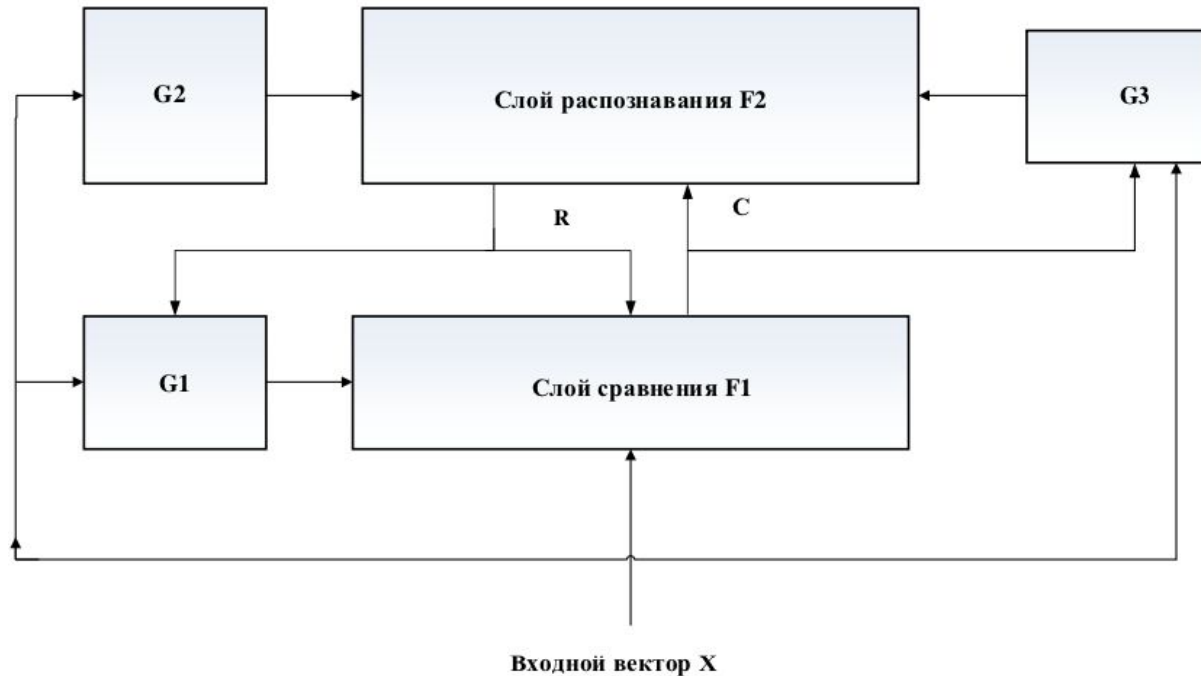
Важным понятием в теории адаптивного резонанса является так называемый **шаблон критических черт** (critical feature pattern) информации. Этот термин показывает, что не все черты (детали), представленные в некотором образе, являются существенными для системы восприятия. Результат распознавания определяется присутствием специфических критических особенностей в образе.

А также

Сеть ART представляет собой векторный классификатор. Входной вектор классифицируется в зависимости от того, на какой из множества ранее запомненных образов он похож. Свое классификационное решение сеть ART выражает в форме возбуждения одного из нейронов распознающего слоя. Если входной вектор не соответствует ни одному из запомненных образов, создается новая категория путем запоминания образа, идентичного новому входному вектору. Если определено, что входной вектор похож на один из ранее запомненных с точки зрения определенного критерия сходства, запомненный вектор будет изменяться (обучаться) под

воздействием нового входного вектора таким образом, чтобы стать более похожим на этот входной вектор. Запомненный образ не будет изменяться, если текущий входной вектор не окажется достаточно похожим на него. Таким образом, решается дилемма стабильности-пластичности. Новый образ может создавать дополнительные классификационные категории, однако он не может заставить измениться существующую память.

Архитектура сети АРТ-1



Архитектура включает два слоя нейронов — так называемые "слой сравнения" и "слой распознавания". Приемник 1, Приемник 2 и Сброс обеспечивают управляющие функции, необходимые для обучения и классификации.

Функционирование сети АРТ в процессе классификации

Процесс классификации в АРТ состоит из трех основных фаз: распознавание, сравнение и поиск. **Много букв, сокращать нечего (:с).**

Фаза распознавания. В начальный момент времени входной вектор отсутствует на входе сети; следовательно, все компоненты входного вектора X можно рассматривать как нулевые. Тем самым сигнал $G2$ устанавливается в 0 и, следовательно, в нуль устанавливаются выходы всех нейронов слоя распознавания. Поскольку все нейроны слоя распознавания начинают работу в одинаковом состоянии, они имеют равные шансы выиграть в последующей конкуренции. Затем на вход сети подается входной вектор X , который должен быть классифицирован. Этот вектор должен иметь одну или более компонент, отличных от нуля, в результате чего и $G1$, и $G2$ становятся равными единице. Это "подкачивает" нейроны слоя сравнения, обеспечивая один из двух единичных входов, необходимых для возбуждения нейронов в соответствии с правилом двух третей, и тем самым позволяя нейрону возбуждаться, если соответствующая компонента входного вектора равна единице. Таким образом, в течение данной фазы вектор C в точности дублирует вектор X .

Далее, для каждого нейрона в слое распознавания вычисляется свертка вектора его весов V_j и вектора C (см. рис. 11.5). Нейрон с максимальным значением свертки имеет веса, наилучшим образом соответствующие входному вектору. Он выигрывает

конкуренцию и возбуждается, одновременно затормаживая все остальные нейроны этого слоя. Таким образом, единственная компонента r_j вектора R (см. рис. 11.3) становится равной единице, а все остальные компоненты становятся равными нулю.

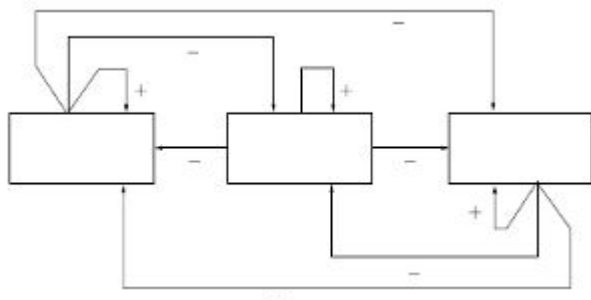


РИС. 11.5.

В результате сеть АРТ запоминает образы в весах нейронов слоя распознавания — один нейрон для каждой категории классификации. Нейрон слоя распознавания, веса которого наилучшим образом соответствуют входному вектору, возбуждается, его выход устанавливается в единичное значение, а выходы остальных нейронов этого слоя устанавливаются в нуль.

Фаза сравнения. Единственный возбужденный нейрон в слое распознавания возвращает единицу обратно в слой сравнения в виде своего выходного сигнала r_j . Эта единственная единица может быть визуально представлена в виде "веерного" выхода, подающегося через отдельную связь с весом t_{ij} на каждый нейрон в слое сравнения, обеспечивая каждый нейрон сигналом p_j , равным величине t_{ij} (нулю или единице) (см. рис. 11.6). T_j — это "эталонный" образец категории j .

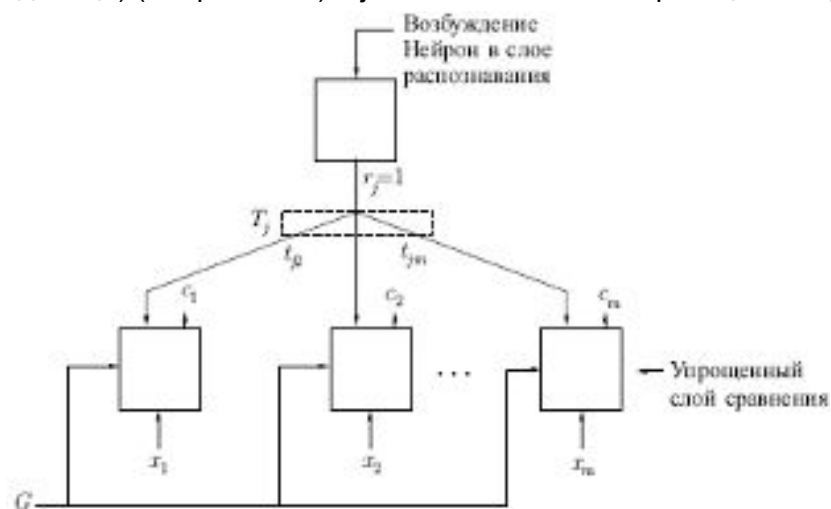


РИС. 11.6.

Алгоритмы инициализации и обучения построены таким образом, что каждый весовой вектор T_j имеет двоичные значения весов; кроме того, каждый весовой вектор V_j представляет собой масштабированную версию соответствующего вектора T_j . Это означает, что все компоненты P (вектора возбуждения слоя сравнения) также являются двоичными величинами.

Так как вектор R не является больше нулевым, сигнал $G1$ устанавливается в нуль. Таким образом, в соответствии с правилом двух третей, возбудиться могут только нейроны, получающие на входе одновременно единицы от входного вектора X и вектора P .

Другими словами, обратная связь от распознающего слоя действует так, чтобы установить компоненты \mathbf{v} в нуль в случае, если входной вектор не соответствует входному образу (т. е. если \mathbf{v} и \mathbf{v}_i не имеют совпадающих компонент).

Если имеются существенные различия между \mathbf{v} и \mathbf{v}_i (малое количество совпадающих компонент векторов), несколько нейронов на фазе сравнения будут возбуждаться и будет содержать много нулей, в то время как \mathbf{v}_i содержит единицы. Это означает, что возвращенный вектор \mathbf{v} не является искомым и возбужденные нейроны в слое распознавания должны быть заторможены. Такое торможение производится блоком сброса (см. рис. 11.2), который сравнивает входной вектор \mathbf{v} и вектор \mathbf{v}_i и вырабатывает сигнал сброса, если степень сходства этих векторов меньше некоторого уровня. Влияние сигнала сброса заключается в установке выхода возбужденного нейрона в нуль, отключая его на время текущей классификации.

Фаза поиска. Если не выработан сигнал сброса, сходство является адекватным и процесс классификации завершается. В противном случае, другие запомненные образы должны быть исследованы с целью поиска лучшего соответствия. При этом торможение возбужденного нейрона в распознающем слое приводит к установке всех компонент вектора \mathbf{v} в 0, устанавливается в 1 и входной вектор \mathbf{v} опять прикладывается в качестве \mathbf{v}_i . В результате другой нейрон выигрывает соревнование в слое распознавания и другой запомненный образ возвращается в слой сравнения. Если \mathbf{v} не соответствует \mathbf{v}_i , возбужденный нейрон в слое распознавания снова тормозится. Этот процесс повторяется до тех пор, пока не встретится одно из двух событий:

1. Найден запомненный образ, сходство которого с вектором \mathbf{v} выше уровня параметра сходства, т. е. θ . Если это происходит, проводится обучающий цикл, в процессе которого модифицируются веса векторов \mathbf{v} и \mathbf{v}_i , связанных с возбужденным нейроном в слое распознавания.
2. Все запомненные образы проверены, определено, что они не соответствуют входному вектору, и все нейроны слоя распознавания заторможены. В этом случае предварительно не распределенный нейрон в распознающем слое выделяется этому образу и его весовые векторы \mathbf{v} и \mathbf{v}_i устанавливаются соответствующими новому входному образу.

25. Сеть Кохонена: Принципы работы сети Кохонена. Алгоритм обучения сети Кохонена. Режим интерполяции и аккредитации

Сеть Кохонена используют для отображения нелинейных зависимостей на двумерные (чаще всего) сетки, представляющие метрические и топологические зависимости входных векторов, объединяемых в кластеры.

Нейронная сеть Кохонена имеет один слой нейронов. Количество входов каждого нейрона равно размерности входного вектора. Количество нейронов непосредственно определяет сколько различных кластеров сеть может распознать.

Основная цель обучения в KNC состоит в выявлении структуры в n - мерных входных данных и предоставлении ее на карте в виде распределенных нейронных активностей. Каждый нейрон несет информацию о кластере, объединяющем в группу схожие по критерию близости входные вектора, формируя для данной группы собирательный образ.

Принцип обучения

Для нейрона-победителя синаптические связи усиливаются, а для остальных нейронов не изменяются или могут уменьшаться. Победителем в конкуренции является нейрон, который в результате подачи на вход сети определенного образа имеет максимальную взвешенную активность

$$S_j = \sum_i w_{ij} x_i = W_j X^T$$

где $X = \{x_1, x_2, \dots, x_n\}$ — входной образ,

$W_j = \{w_{1j}, w_{2j}, \dots, w_{nj}\}$ — вектор столбец весовых коэффициентов j -го выходного нейрона.

Пусть $S_k = \max S_j$

Тогда активность выходных нейронов определяется по формуле

$$y_j = F(S_j) = \begin{cases} 1, & \text{если } j = k \\ 0, & \text{если } j \neq k \end{cases}$$

После обучения нейронной сети при подаче входного образа активность нейрона-победителя принимается равной единице, а остальных нейронов нулю. Это правило известно под названием «**победитель берет все**»

Алгоритм обучения

1. Присвоение весовым коэффициентам начальных значений малыми случайными числами
2. Подаем на вход один из векторов X^P

3. Рассчитывается выход слоя Кохонена $D^{m,p}$ и определяется номер выигравшего нейрона m_0 , выход которого максимален
4. Коррекция весов только выигравшего нейрона m_0
5. $W^{m_0} = W^{m_0} + a(X^p - W^{m_0})$

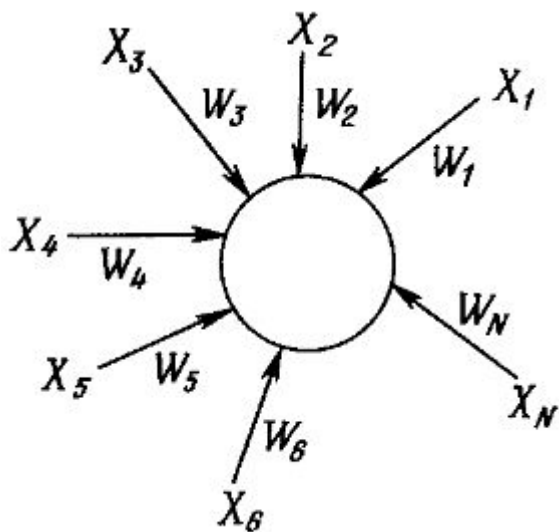
a - малая положительная величина или монотонно убывающая

В режиме **аккредитации** активируется **единственный нейрон** Кохонена с максимальным значением выхода

В режиме **интерполяции** активируются **несколько нейронов** сети Кохонена

26. Нейронные сети встречного распространения: характеристика входной звезды ГРОССБЕРГА, обучение входной звезды, выходные звезды ГРОССБЕРГА; обучение сети встречного распространения

Характеристика входной звезды ГРОССБЕРГА



Нейрон в форме входной звезды Гроссберга имеет n входов, которым соответствуют весовые коэффициенты $W=(w_1, w_2, \dots, w_n)$, и один выход Y , являющийся взвешенной суммой входов. Входная звезда обучается выдавать на выходе сигнал всякий раз, когда на входы поступает определенный вектор.

Обучение входной звезды Гроссберга

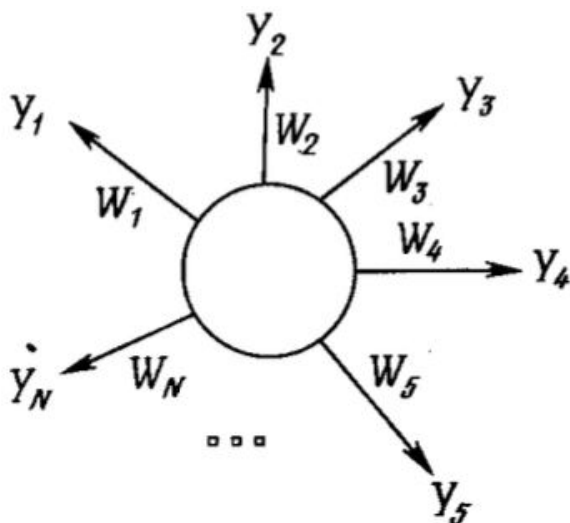
Входная звезда выполняет распознавание образов, т. е. она обучается реагировать на **определенный входной вектор X и ни на какой другой**. Это обучение реализуется путем настройки весов таким образом, чтобы они соответствовали входному вектору. Выход входной звезды определяется как взвешенная сумма ее входов, как это описано в предыдущих разделах.

Процесс обучения выражается следующим образом:

$$w_i(t+1) = w_i(t) + \alpha[x_i - w_i(t)],$$

где w_i – вес входа x_i ; x_i – i -й вход; α – нормирующий коэффициент обучения, который имеет начальное значение 0,1..0,2 и постепенно уменьшается в процессе обучения.

Характеристика выходной звезды Гроссберга



Обучение выходной звезды Гроссберга

Выходная звезда Гроссберга выполняет противоположную функцию – **при поступлении сигнала на вход выдается определенный вектор**. Нейрон этого типа имеет один вход и m выходов с весами $W=(w_1, w_2, \dots, w_m)$, которые подстраиваются по формуле:

$$w_i(t+1) = w_i(t) + b[y_i - w_i(t)],$$

где y_i - выходной вектор

w_i - весовой вектор выходной звезды

b представляет собой нормирующий коэффициент обучения, который в начале приблизительно равен единице и постепенно уменьшается до нуля в процессе обучения.

Выходная звезда имеет дополнительную функцию; она вырабатывает требуемый возбуждающий сигнал для других нейронов всякий раз, когда возбуждается.

Обучение сети встречного распространения

1 шаг. Весовые векторы слоя Кохонена настраиваются таким образом, чтобы провести распределение входных векторов по кластерам. Каждому из которых соответствует один нейрон-победитель. Обучение проводится без учителя. Точность кластеризации будет гарантирована только тогда, когда обучающая выборка является представительной.

2 шаг. На втором шаге обучение осуществляется с учителем. Проводится подстройка весовых коэффициентов выходного слоя Гроссберга на примерах с заданным выходом по формуле обучения выходной звезды Гроссберга. При этом настраиваются только веса, соответствующие нейрону-победителю в текущем такте, который посылает выходной сигнал равный единице.

Темпы обучения нейронов слоя Кохонена и Гроссберга должны быть согласованы; при этом в слое Кохонена подстраиваются также веса всех нейронов в окрестности победителя, которая постепенно сужается до одного нейрона.

27. Исчисление высказываний, ИВ и естественный язык. Связки и формулы булевой алгебры. Выполняемые, невыполняемые и общезначимые высказывания. Метод редукции на примере

Исчисление высказываний — это формальная теория, в которой осуществляется попытка формализации понятий логического закона и логического следования.

Высказывание — это повествовательное предложение, которое истинно или ложно. В классическом исчислении высказываний значимым является лишь истинностное значение высказывания, поэтому используемые в дальнейшем высказывательные (пропозициональные) переменные могут принимать одно из этих значений.

ИВ и естественный язык

Идёт дождь (высказывание p), дорога мокрая (высказывание q).

Связка некоммуникативна.

«Если p , то q » можно представить

$p \rightarrow q$

Здесь p - это посылка, а q - это заключение.

Связки и формулы булевой алгебры

Связки:

- Отрицание
- Конъюнкция
- Дизъюнкция
- Импликация
- Эквивалентность

Формулы булевой алгебры

См.: [Булева алгебра. Основные тождества](#)

Выполняемые, невыполняемые и общезначимые высказывания

Формула называется **выполнимой**, если она истинна при некоторой интерпретации.

Например, формула $(p \& q)$ выполняется, если p, q - истинны.

Формула называется **невыполнимой**, если она ложна при любой интерпретации $(p \& \sim p)$.

Формула называется **общезначимой** (или тавтологией), если она истинна при любой интерпретации $(\sim(\sim p) = p)$.

Метод редукции на примере

Задача: Определить общезначима ли формула двумя методами: метод редукции и метод упрощения формулы с помощью законов булевой алгебры.

$[(p \& q) \rightarrow r] \vee (r \rightarrow p)$

Решение:

Допустим формула принимает ложное значение и доказываем её абсурдность

$[(p \& q) \rightarrow r] \vee (r \rightarrow p) = 0$

Оба компонента должны быть = 0

Правая компонента = 0, только в одном случае

$(r \rightarrow p) = 0$ При $r = 1$ $p = 0$

Подставим $r = 1$ $p = 0$ в левую компоненту

$[(p \& q) \rightarrow r] = [(1 \& q) \rightarrow 1 = 1$ при любом q значение = 1

Следовательно, формула общезначима

28. Методы дедукции: прямая и обратная. Доказательство выводимости цели из фактов на примерах с помощью методов дедукции. Правило «modus ponens»

Дедукция (от лат. deductio выведение) – аналитический процесс, основанный на **применении общих правил к частным случаям**, с выводом результата, т.е. это умозаключение, в котором вывод про отдельный класс делается на основе абстрактного класса в целом, переход в процессе познания от общего к частному и единичному.

Прямая дедукция

$H1 \& H2 \& \dots \& Hn \& \sim C = \text{false}$

Обратная дедукция

$\sim H1 \vee \sim H2 \vee \dots \vee \sim Hn \vee C = \text{true}$

H_i - посылки; C - логическое следствие из H_i .

Пример решения задачи

Задача: Установить выводимость заключения (C) из посылок ($H1 - H3$) тремя методами: прямой дедукции, обратной дедукции и методом резолюций.

$C = \sim q$
$H1 = p \rightarrow q$
$H2 = (p \rightarrow \sim q) \& \sim r$
$H3 = q \rightarrow r$

Прямая дедукция $H1 \& H2 \& \dots \& Hn \& \sim C = 0$	Обратная дедукция $\sim H1 \vee \sim H2 \vee \dots \vee \sim Hn \vee C = 1$
<p>Решение:</p> $(p \rightarrow q) \& (p \rightarrow \sim q) \& \sim r \& (q \rightarrow r) \& q = 0$ $(\sim p \vee q) \& (\sim p \vee \sim q) \& \sim r \& (\sim q \vee r) \& q = 0$ $(\sim p \vee q) \& (\sim p \vee \sim q) \& \sim r \& \sim q \& q = 0$ $(\sim p \vee q) \& (\sim p \vee \sim q) \& \sim r \& 0 = 0$ $0 = 0$, доказано	<p>Решение:</p> $1. \sim(p \rightarrow q) = \sim(\sim p \vee q) = p \& \sim q$ $2. \sim[(p \rightarrow \sim q) \& \sim r] = \sim[(\sim p \vee \sim q) \& \sim r] = p \& q \vee r$ $3. \sim(q \rightarrow r) = \sim(\sim q \vee r) = q \& \sim r$ $p \& \sim q \vee p \& q \vee r \vee q \& \sim r \vee \sim q = 1$ $p \& \sim q \vee p \& q \vee r \vee q \vee \sim q = 1$ $p \& \sim q \vee p \& q \vee r \vee 1 = 1$ $1 = 1$, доказано

Правило вывода **modus ponens**: если A и $A \rightarrow B$ — выводимые формулы, то B выводима.

29. Принцип резолюций, метод резолюций; стратегии, используемые при доказательстве теорем с помощью метода резолюций

Множество дизъюнктов невыполнимо тогда, когда пустой дизъюнкт типа $P \vee \sim P = \emptyset$ является логическим следствием из этого множества.

Принцип резолюции

Способ доказательства истинности утверждения $A \rightarrow B$ состоит в том, чтобы показать ложность его отрицания: $\sim(A \rightarrow B) = \emptyset$

Доказательство принципа резолюций

Пусть A, B, X – формулы. Предположим две формулы истинны: $(A \vee X)$ и $(B \vee \sim X)$

Если X истина, то B должна быть истинной.

Если X ложна, то A должна быть истинной.

В обоих случаях $(A \vee B)$ истина.

$\{A \vee X, B \vee \sim X\} \vdash A \vee B$

$\{\sim X \rightarrow A, X \rightarrow B\} \vdash A \vee B$

Если X – высказывание, а A и B – дизъюнкты, это правило называется правилом резолюций.

Лемма общезначимости правила резолюций

Пусть S_1 и S_2 – дизъюнкты в нормальной форме множества S , “ a ” – литера.

Если $a \in S_1$ и $\sim a \in S_2$, то дизъюнкт $R = (S_1 \setminus \{a\}) \vee (S_2 \setminus \{\sim a\})$ является логическим следствием нормальной формы S , и дизъюнкт R называется резольвентой дизъюнктов S_1 и S_2 .

Метод резолюций

1. Приводим все посылки и отрицание заключения к КНФ (продвигаем отрицание внутрь, избавляемся от импликации, эквивалентности).
2. Посылки представляют собой конъюнкцию дизъюнктов. Выписываем каждый дизъюнкт с новой строки.
3. Каждый дизъюнкт – это дизъюнкция (может быть одночленная), состоящая из предложений и отрицаний предложений.
4. Выбирается два любые дизъюнкта, содержащие один и тот же атом, но с противоположными знаками.
5. Формируется резольвента, из которой удаляются атомы с противоположными знаками.
6. Процесс продолжается до получения пустого дизъюнкта, который выражает противоречие. Это завершает доказательство от противного (из P и $\sim P$ выводится ложь)

Стратегии

Стратегия опорного множества:

Применима только при поиске доказательства

1. Некоторые предложения экспериментатор называют аксиомами, а все другие относятся к опорному множеству.

2. Программе запрещено проводить поиск между двумя аксиомами. Все другие резолюции допустимы.

Стратегия сначала вширь

1. Первоначально все предложения имеют уровень 0.
2. Стратегия порождает уровень 1 путем получения резольвент.
3. Из предыдущих (уровни 0 и 1) стратегия порождает уровень 2 и т.д.

Стратегия «предпочтение единичным элементам»

1. Производится дедуктивный вывод предложений, содержащих возможно меньшее число литер.
2. Короткие предложения легче обрабатывать.
3. Стратегия даёт наивысший приоритет резолюциям единичных элементов.
4. Таким образом стратегия устанавливает следующий порядок нахождения резольвент: единичный элемент с единичным элементом; единичный элемент с предложениями 2-го порядка и т.д.

30. Основы нечетких систем. Достоинства нечетких систем. Понятие нечеткого множества, лингвистической переменной, терма, степени принадлежности, базы нечетких правил. Перспективы развития нечетких систем

Нечеткое множество – множество, элементы которого принадлежат ему в той или иной степени.

Нечеткая логика – умозаключение с использованием нечетких множеств или нечетких правил.

Нечеткое правило – условное высказывание вида "ЕСЛИ X есть A , ТО Y есть B ", где A и B - нечеткие множества. Нечеткое правило образует связь между нечеткими множествами.

Нечеткая система – множество нечетких правил, преобразующих входные данные в выходные.

Нечетким множеством на универсальном множестве называется совокупность пар $(\mu_A(u), u)$, где $\mu_A(u)$ - степень принадлежности элемента $u \in U$ к нечеткому множеству A .

Степень принадлежности - это число из диапазона $[0; 1]$. Чем выше степень принадлежности, тем с большей мерой элемент универсального множества соответствует свойствам нечеткого множества.

Кардинальное число нечеткого множества равно сумме коэффициентов принадлежности всех элементов к этому множеству.

Лингвистическая переменная (ЛП) - переменная, значением которой являются нечеткие множества, описываемые в форме слов или предложений на естественном или искусственном языке. Формально ЛП задается набором множеств:

$\{X, T(X), U, G, M\}$,

где X - название переменной;

$T(X)$ - терм-множество переменной X ;

U - универсальное множество;

G - синтаксическое правило, порождающее названия значений переменной X ;

M - семантическое правило, которое ставит в соответствие каждому значению ЛП ее смысл.

Пример. $X = \text{ОЦЕНКА}$ $T(\text{ОЦЕНКА}) = \{\text{неудовлетворительно, удовлетворительно, хорошо, отлично}\}$ $U = [0, 100]$.

Терм-множество - множество всех возможных значений лингвистической переменной.

Терм - любой элемент терм-множества, который задается нечетким множеством посредством функции принадлежности.

Основой для проведения операции нечеткого логического вывода является **база правил**, содержащая нечеткие высказывания в форме «если - то» и функции принадлежности для соответствующих лингвистических термов. Базовое правило

вывода типа «если – то» называется нечеткой импликацией, принимающей форму «если x это A , то y это B », где A и B – это лингвистические значения, идентифицированные нечетким способом через соответствующие функции принадлежности для переменных x и y . Импликацию записывают так же в сокращенном виде $A \rightarrow B$.

На **базу правил** существуют следующие ограничения:

- существует хотя бы одно правило для каждого лингвистического термина выходной переменной.
- для любого термина входной переменной имеется хотя бы одно правило, в котором этот терм используется в качестве предпосылки (левая часть правила).

Области применения систем, в которых используется нечеткая логика:

- выбор эффективного варианта инвестиционного проекта в условиях наличия неопределенности (невозможность предсказания инфляции, спроса, технических и ресурсных показателей самого проекта).
- автоматическое управление воротами плотины на гидроэлектростанциях;
- упрощенное управление роботами;
- наведение телекамер при трансляции спортивных событий;
- замена экспертов при анализе работы биржи;
- предотвращение нежелательных температурных флуктуаций в системах кондиционирования воздуха;
- эффективное и стабильное управление автомобильными двигателями;
- управление экономичной скоростью автомобилей;
- улучшение эффективности и оптимизация промышленных систем управления;
- позиционирование приводов в производстве полупроводников wafer-steppers;
- оптимизированное планирование автобусных расписаний;
- системы архивации документов;
- системы прогнозирования землетрясений;
- медицина: диагностика рака;
- сочетание методов нечеткой логики и нейронных сетей;
- распознавание рукописных символов в карманных компьютерах (записных книжках);
- распознавание движения изображения в видеокамерах;
- автоматическое управление двигателем пылесосов с автоматическим определением типа поверхности и степени засоренности;
- однокнопочное управление стиральными машинами;
- распознавание рукописных текстов, объектов, голоса;
- моделирование судебных процессов;
- САПР производственных процессов;
- управление скоростью линий и температурой при производстве стали;
- управление метрополитенами для повышения удобства вождения, точности остановки и экономии энергии;
- оптимизация потребления бензина в автомобилях;
- повышение чувствительности и эффективности управления лифтами;
- повышение безопасности ядерных реакторов.

31. Нечеткие правила, система нечеткого логического вывода, модель нечеткого вывода Мамдани–Заде. Способы реализации агрегатора нечеткой системы Мамдани.

В общем случае механизм логического вывода включает четыре этапа: введение нечеткости (фаззификация), нечеткий вывод, композиция и приведение к четкости, или дефаззификация.

Схема системы нечеткого логического вывода



Модель нечеткого вывода Мамдани-Заде

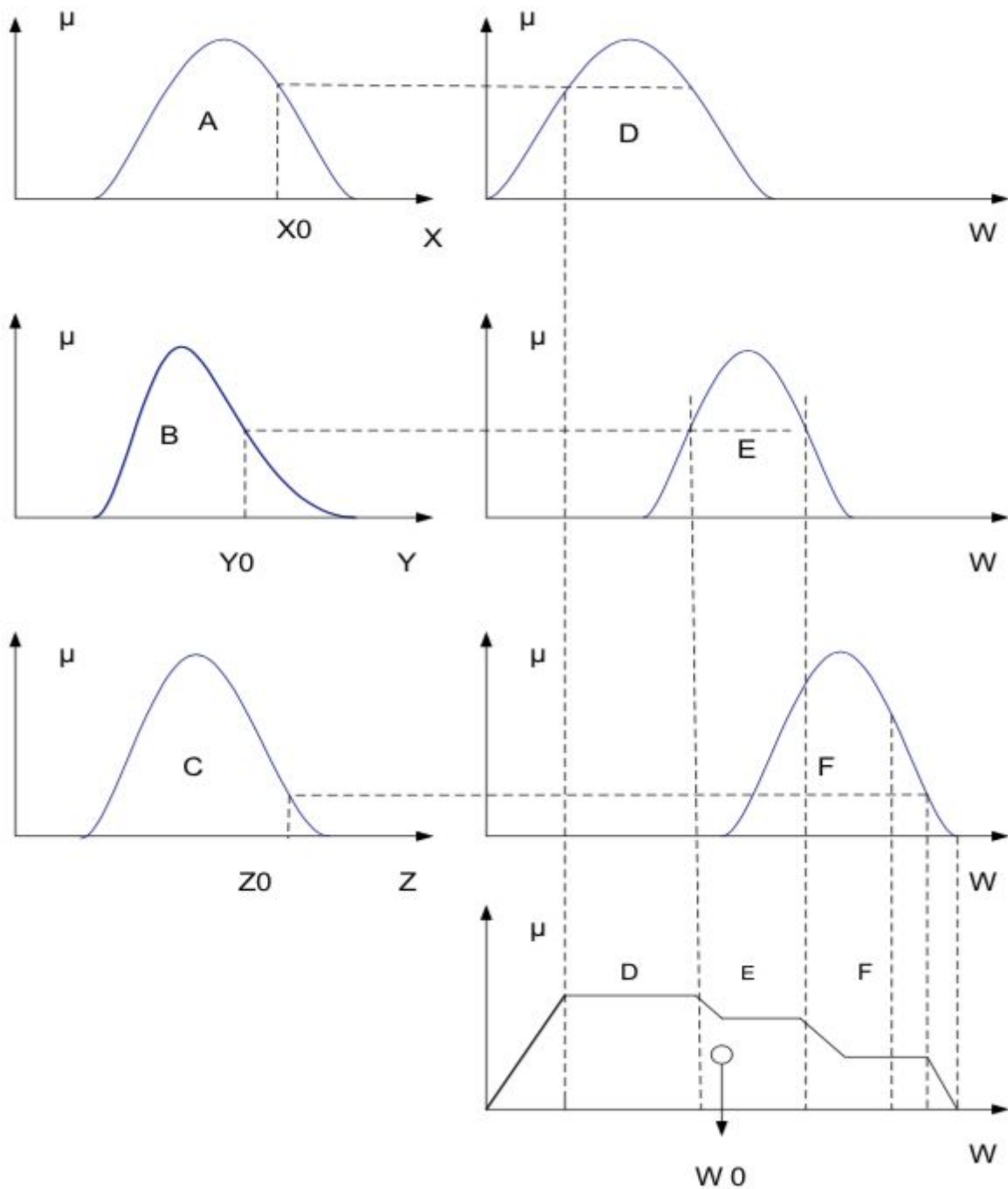
В модели Мамдани-Заде присутствуют следующие операторы:

- оператор логического или арифметического произведения для определения результирующего уровня активации, в котором учитываются все компоненты вектора x условия
- оператор логического или арифметического произведения для определения значения функции принадлежности для всей импликации $A \rightarrow B$
- оператор логической суммы как агрегатор равнозначных результатов импликации многих правил
- оператор дефаззификации, трансформирующий нечеткий результат $\mu(y)$ в четкое значение выходной переменной y

Алгоритм нечеткого вывода Мамдани – это наиболее распространенный способ логического вывода в нечетких системах. В нем используется **минимаксная композиция** нечетких множеств. Данный механизм включает в себя следующую последовательность действий:

1. Процедура фаззификации: определяются степени истинности, т.е. значения функций принадлежности для левых частей каждого правила (предпосылок). Для базы правил с m правилами обозначим степени истинности как $A_{ik}(x_k)$, $i=1, 2, \dots, m$, $k=1, 2, \dots, n$.
2. Нечеткий вывод. Сначала определяются уровни "отсечения" для левой части каждого из правил: $\alpha_{fi} = \min(A_{ik}(x_k))$

3. Далее находятся "усеченные" функции принадлежности: $B_i^*(y) = \min(\alpha_i, B_i(y))$. $B_i(y)$ уже заданы.
4. Композиция, или объединение полученных усеченных функций, для чего используется композиция нечетких множеств по критерию максимума. Т.е. графики функций принадлежности $B_i^*(y)$ накладываются друг на друга, образуя ломаную линию.
5. Далее осуществляется процедура приведения к четкости, например, методом среднего центра, или центроидным методом



32. Фаззификатор, дефаззификатор. Методы реализации фаззификатора и дефаззификатора.

Методы реализации фаззификатора

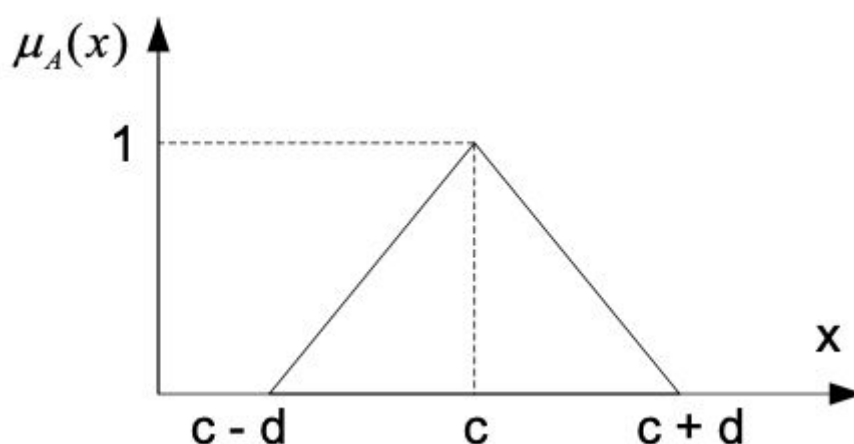
Фаззификатор преобразует N-мерный входной вектор $X = [x_1, x_2, \dots, x_N]$ в нечеткое множество A , характеризуемое функцией принадлежности $\mu(x)$ с четкими переменными. Несмотря на то, что нечеткие системы могут иметь функции принадлежности произвольной структуры, с практической точки зрения наибольшей популярностью пользуются функции **гауссовского типа, а также треугольные и трапециевидальные функции**.

Общая форма обобщенной гауссовской функции для переменной x с центром c и вариацией σ и параметром b для множества F имеет вид

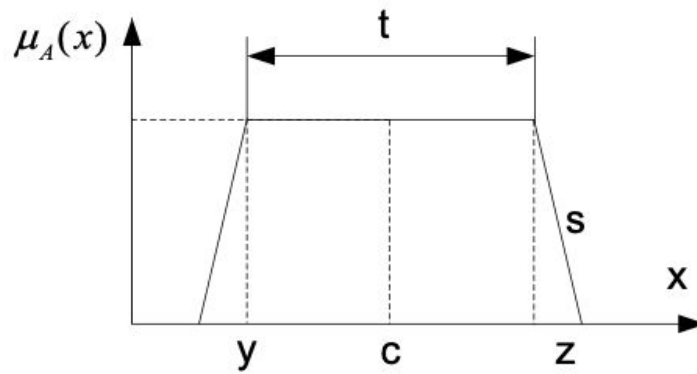
$$\mu_A(x) = \exp \left[- \left(\frac{x - c}{\sigma} \right)^{2b} \right]$$

Помимо гауссовской функции принадлежности, на практике часто применяется **симметричная треугольная функция**, которую можно записать в виде

$$\mu_A(x) = \begin{cases} 1 - \frac{|x - c|}{d} & \text{для } x \in [c - d, c + d] \\ 0 & \text{для остальных} \end{cases}$$



Обобщением треугольной функции является трапециевидальная функция принадлежности, форма и обозначение которой приведены на рисунке.



$$\mu_{A^{(i)}}(x_i) = \begin{cases} 0 & \text{для } x > z \text{ или } x < y \\ 1 & \text{для } c - t/2 \leq x \leq c + t/2 \\ s(z - x) & \text{для } c + t/2 \leq x \leq z \\ s(z - y) & \text{для } y \leq x \leq c - t/2 \end{cases}$$

Методы реализации дефаззификатора

Дефаззификатор трансформирует нечеткое множество в полностью детерминированное точечное решение y . Нечеткое множество представляет зависимость $\mu(y)=\mu A \rightarrow B(y)$ как функцию от выходной переменной y . Преобразование этого множества в единственное точечное решение возможно многими способами. Наиболее известны среди них:

дефаззификация относительно центра области

$$y_c = \frac{\int \mu(y) y dy}{\int \mu(y) dy}$$

либо в дискретной форме

$$y_c = \frac{\sum \mu(y_i) y_i}{\sum \mu(y_i)}$$

дефаззификация относительно среднего центра

$$y_c = \frac{\sum \mu(y_{ci}) y_{ci}}{\sum \mu(y_{ci})}$$

33. Этапы логического вывода для двух переменных на примере механизма Мамдани, Sugeno, Larsen.

Хочешь [еще](#)?

В алгоритме Мамдани:

- 1) Находятся степени истинности для предпосылок каждого правила: $A_1(x_0)$, $A_2(x_0)$, $B_1(y_0)$, $B_2(y_0)$
- 2) Определяются уровни “отсечения” для левой части каждого из правил, где через «^» обозначена операция логического минимума (min)

$$\alpha_1 = A_1(x_0) \wedge B_1(y_0)$$

$$\alpha_2 = A_2(x_0) \wedge B_2(y_0)$$

- 3) Находятся “усеченные” функции принадлежности

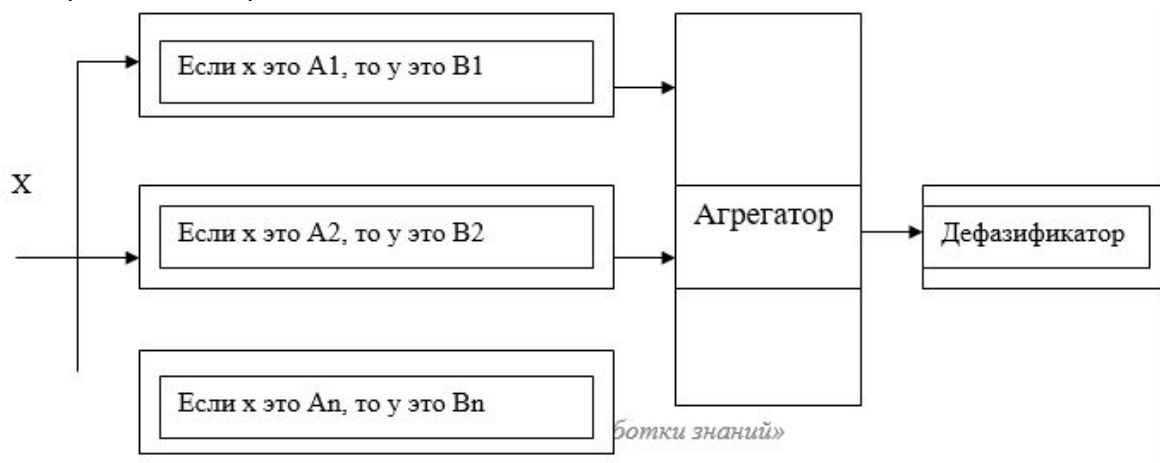
$$C'_1(z) = (\alpha_1 \wedge C_1(z)),$$

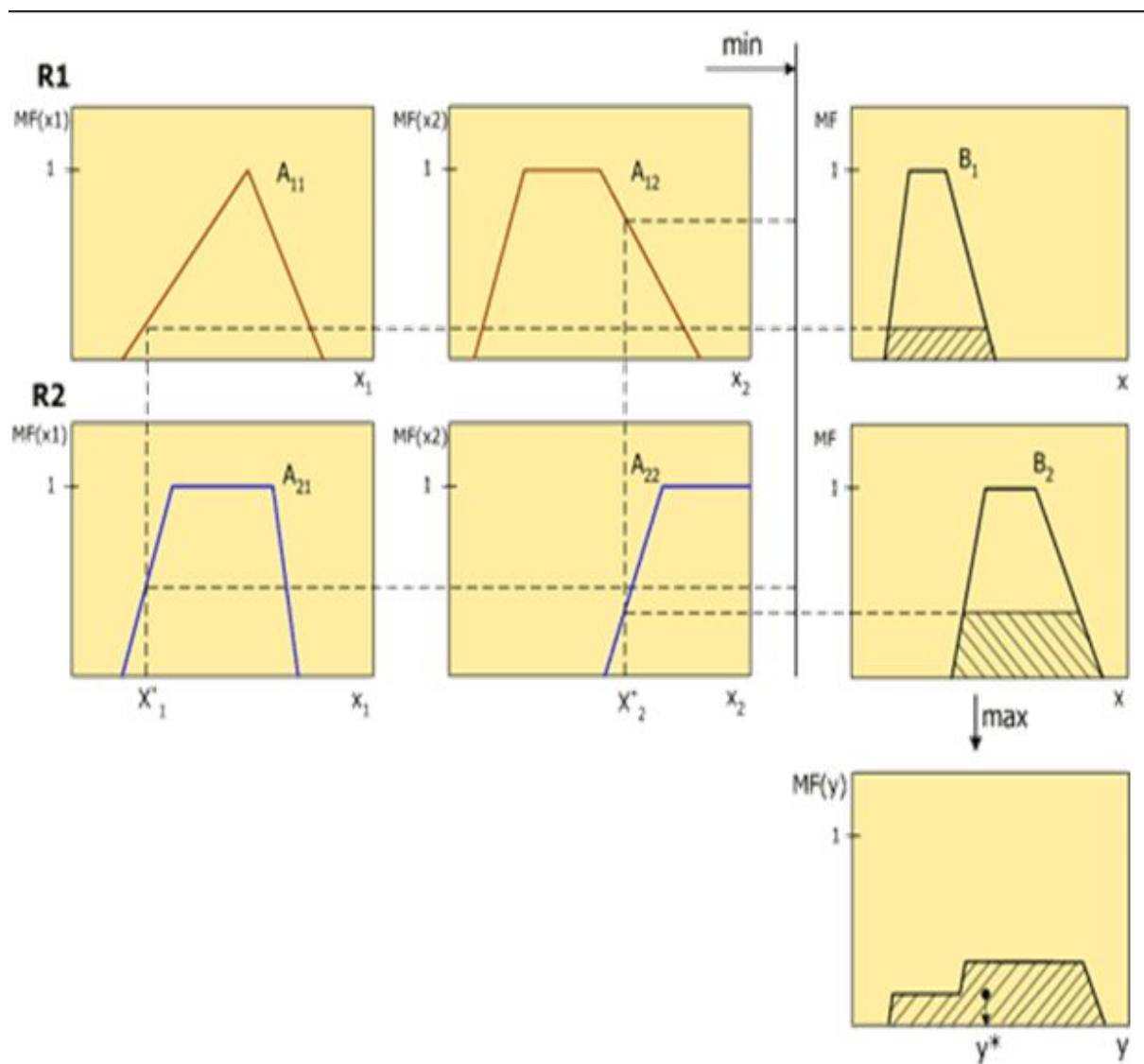
$$C'_2(z) = (\alpha_2 \wedge C_2(z)).$$

- 4) Композиция: с использованием операции МАКСИМУМ (max, далее обозначаемой как «v») производится объединение найденных усеченных функций, что приводит к получению итогового нечеткого подмножества для переменной выхода с функцией принадлежности

$$\mu_{\Sigma}(z) = C(z) = C'_1(z) \vee C'_2(z) = (\alpha_1 \wedge C_1(z)) \vee (\alpha_2 \wedge C_2(z)).$$

- 5) Осуществляется процедура приведения к четкости, например, методом среднего центра.





В алгоритме Сугено:

Пусть вот есть значит правила:

П1: если x есть A1 и y есть B1, то $z_1 = a_1 x + b_1 y$,

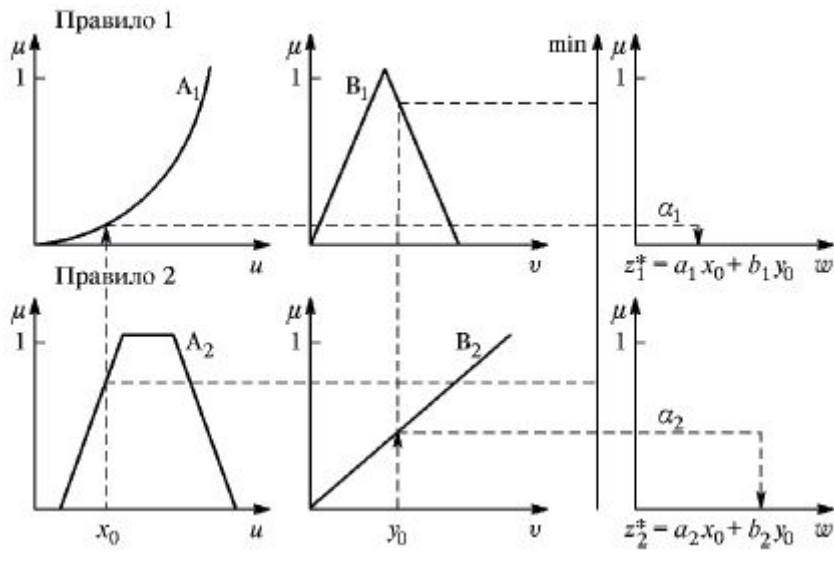
П2: если x есть A2 и y есть B2, то $z_2 = a_2 x + b_2 y$.

- 1) Первый этап выполняется по аналогии с Мамдани.
- 2) На втором этапе находятся индивидуальные выходы правил
 $z_1 = a_1 x + b_1 y$,
 $z_2 = a_2 x + b_2 y$.
- 3) На третьем этапе определяется четкое значение переменной вывода по

$$Z_0 = \frac{\alpha_1 z_1 + \alpha_2 z_2}{\alpha_1 + \alpha_2}.$$

формуле

Изображение работы алгоритма:



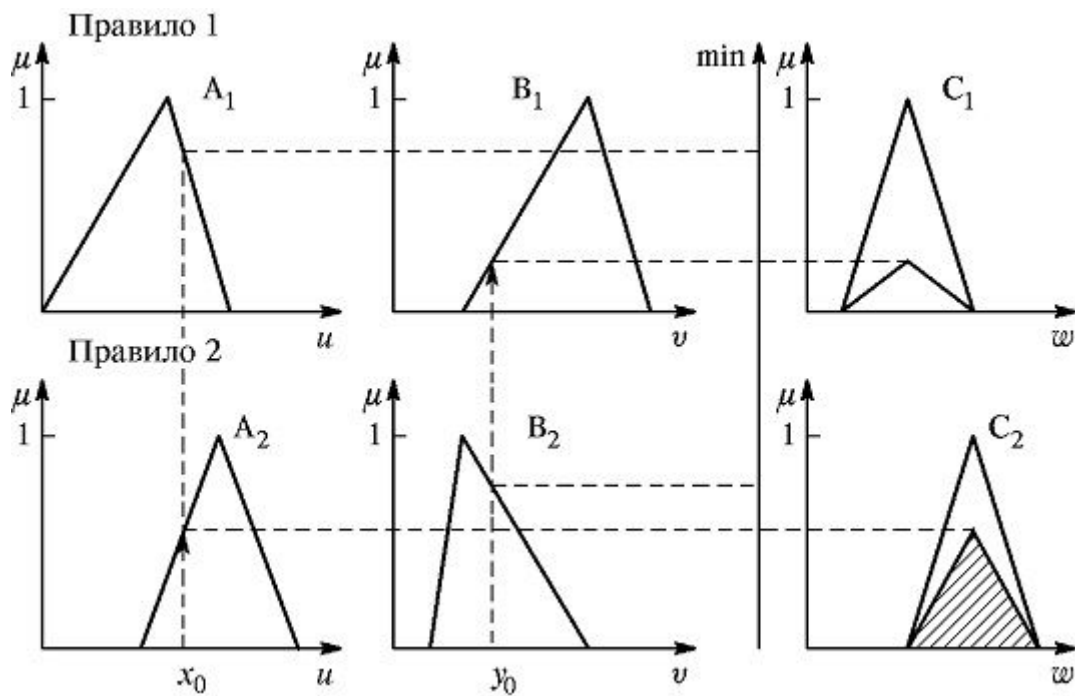
Алгоритм Ларсена:

- 1) Первый этап как в Мамдани
- 2) На втором этапе находятся значения

$$\alpha_1 = A_1(x_0) \wedge B_1(y_0),$$

$$\alpha_2 = A_2(x_0) \wedge B_2(y_0),$$
 Затем частные нечеткие подмножества $\alpha_1 C_1(z)$, $\alpha_2 C_2(z)$.
- 3) Находится итоговое нечеткое подмножество с функцией принадлежности

$$\mu_S(z) = C(z) = (\alpha_1 C_1(z)) \vee (\alpha_2 C_2(z))$$
- 4) При необходимости производится приведение к четкости



34. Методы извлечения знаний. Коммуникативные (пассивные и активные) и текстологические.

Извлечение знаний - процесс получения знаний от экспертов.

Классификация по источнику знаний:

- **Коммуникативные** методы извлечения знаний охватывают методы и процедуры контактов инженера по знаниям с экспертом
- **Текстологические** включают методы извлечения знаний из документов (методик, пособий, руководств) и специальной литературы (статей, монографий, учебников)

Коммуникативные методы подразделяются на **пассивные** и **активные**.

Пассивные методы подразумевают, что ведущая роль в процедуре извлечения как бы передается эксперту, а инженер по знаниям только протоколирует рассуждения эксперта во время его реальной работы по принятию решения или записывает то, что эксперт расскажет в форме лекции.

В **активных** методах, напротив, инициатива находится в руках инженера по знаниям, который активно контактирует с экспертом различными способами – в играх, диалогах, беседах за круглым столом и т. п.

Активные методы разделяются на две группы в зависимости от числа экспертов: **групповые** и **индивидуальные**.

Методы извлечения знаний

Коммуникативные			Текстологические
Пассивные	Активные		
	Групповые	Индивидуальные	
Наблюдение Протокол "мыслей вслух" Лекции	Круглый стол Мозговой штурм Рольевые игры	Анкетирование Интервью Свободный диалог Экспертные игры	Анализ литературы Анализ учебников Анализ документов

35. Понятие извлечения знаний.

Гносеологический и психологический аспекты извлечения знаний.

Извлечение знаний - процесс получения знаний от экспертов.

Психологический аспект является главным, так как определяет успешность и эффективность взаимодействия инженера по знаниям (когнитолога) с источником знаний – экспертом. Потери информации при разговорном общении велики. Поэтому проблема повышения информированности общения аналитика и эксперта связана с применением психологических знаний. Рекомендуется организовать процесс извлечения знаний как совместный процесс поиска истины. Модель общения включает участников, средства общения и предмет общения (знания). В соответствии с этими компонентами выделяют **три слоя психологических проблем: контактный, процедурный и когнитивный.**

Контактный слой. Степень информированности общения аналитика и эксперта на уровне контактного слоя зависит от пола, возраста, темперамента личности и мотивации участников общения. **Установлено, что хорошие результаты дают гетерогенные пары** (мужчина/женщина). Желательно, чтобы участники общения обладали следующими качествами: доброжелательность, хорошая память, внимательность, наблюдательность, воображение, собранность, настойчивость, общительность, находчивость.

Процедурный слой. Обеспечивает эффективность извлечения знаний. **Необходимо подобрать к эксперту метод как ключ к замку.** Параметры процедурного слоя: ситуация общения (место, время, продолжительность), оборудование (вспомогательные средства, освещенность, мебель), профессиональные приемы (тема, стиль, методы).

Когнитивный слой. Связан с изучением семантического пространства памяти эксперта и воссозданием его понятийной структуры. Характеризуется когнитивным стилем и репрезентативностью. **Когнитивный стиль человека** – это совокупность критериев, используемых им в процессе познания. Желательно чтобы когнитолог и эксперт обладали следующими характеристиками:

- полнезависимость (**способность выявлять главные аспекты и отбрасывать второстепенные**)
- рефлексивность (склонность к рассудительности и самоанализу)
- противоположное качество - импульсивность (быстрое и недостаточно обоснованное решение)
- когнитивная эквивалентность (определяет способность человека к извлечению понятий и разбиению их на классы и подклассы)

Эксперт должен обладать устойчивостью представлений, т.е. жесткостью структуры восприятия. Когнитолог, наоборот, должен обладать гибкостью, т.е. умением легко приспосабливаться к новой обстановке.

Гносеологический аспект объединяет проблемы получения нового научного знания, т. к. процесс познания часто сопровождается появлением новых понятий и теорий. В

процессе разработки базы знаний эксперты часто формулируют некоторые закономерности на основе накопленного эмпирического опыта.

Гносеологическая цепочка: «факт → обобщенный факт → эмпирическая закономерность → теоретический закон».

Когнитолога интересуют эмпирические знания эксперта, которые могут оказаться несогласованными. Знания характеризуются следующими аспектами:

- системности (показывает место новых знаний в многоуровневой организации)
- объективность знаний (определить практически невозможно)
- неполнота знаний (связана с невозможностью исчерпывающего описания любой предметной области)
- историзм знаний (связан с развитием и изменением до 95% представлений о предметной области с течением времени)

На начальных этапах когнитолог использует структуру умозаключений эксперта и разные теории для построения формальной модели знаний.

36. Теоретические аспекты приобретения знаний. Автоматизированные системы приобретения знаний. Метод репертуарной решетки Келли приобретения знаний. Методы диад, триад и полного контекста. Пример применения метода триад

Процессы извлечения знаний рассматривают в трех основных аспектах: психологическом, лингвистическом и гносеологическом

Лингвистический аспект. В области лингвистических проблем важными являются понятия: общий код, понятийная структура, словарь пользователя. Общий код – это промежуточный язык общения между экспертом и когнитологом. Этот язык включает совокупность общенаучных и специальных понятий из профессиональной литературы. Он позволяет преодолеть языковой барьер между когнитологом и экспертом. В дальнейшем общий код преобразуется в понятийную структуру или семантическую сеть, которая связывает понятия, хранящиеся в памяти человека. Конечный пользователь не обязан обладать профессиональным языком предметной области, поэтому разрабатывается словарь пользователя, который создается путем доработки словаря общего кода.

Об остальных читайте выше.

Проблема автоматизированного приобретения знаний связана с разработкой специальных информационных технологий, обеспечивающих поддержку процедур извлечения и структурирования знаний.

Репертуарная решетка представляет собой матрицу, которая заполняется либо самим испытуемым, либо экспериментатором в процессе беседы.

Столбцам матрицы соответствует группа объектов, или, иначе элементов. В качестве объектов могут выступать люди, предметы, понятия, отношения и т.п. – все, что интересует психодиагностика.

Строки матрицы представляют собой **конструкты** – биполярные признаки, параметры, шкалы, противоположные способы поведения.

Конструкты либо задаются исследователем, либо выявляются у испытуемого с помощью специальных приемов и процедур.

Конструкт - это некоторый признак или свойство, по которому два или несколько объектов сходны между собой и, следовательно, отличны от третьего объекта или нескольких других объектов.

Слово репертуарная означает, что элементы выбираются по определенным правилам аналогично репертуару ролей в пьесе. При переводе с английского языка термин матрица не используется, так как на пересечении строк и столбцов могут стоять не только цифры.

Репертуарные решетки можно считать специфической разновидностью структурированного интервью.

Весь процесс интервьюирования эксперта можно подразделить на четыре этапа:

- формирование элементов
- формирование конструкторов
- заполнение оценочной решетки
- именование выявленных кластеров

Три первых этапа интервьюирования эксперта осуществляются в блоке основного интервьюирования. Вначале эксперту предлагается сформировать список элементов. Элементы должны быть сформированы в соответствии с требованиями.

Далее программа предлагает эксперту сформировать конструкторы в соответствии с методом триад, диад или полного контекста.

Этап основного интервьюирования заканчивается заполнением оценочной решетки. Эксперт должен давать оценки по шкале от минус пяти до плюс пяти или от минус двадцати до плюс двадцати или от минус пятидесяти до плюс пятидесяти.

Таким образом, в модуле гипотез и свидетельств формируется матрица экспертных оценок, содержащая в себе непосредственно экспертные оценки элементов по конструкторам.

Выявление свидетельств осуществляется одним из трех методов:

- **Метод триад.** Последовательно выдаются триады гипотез, пользователю требуется определить какое-либо признак, который есть у двух гипотез, и нет у третьей. Это будет положительный полюс, при вводе которого нужно ввести отрицательный полюс и общее свидетельство. Если при вводе названий положительного, отрицательного или общего свидетельств произойдет дублирование с ранее введенными идентификаторами, то будет выдано диагностическое сообщение и введенная информация не будет использована.
- **Метод полного контекста.** При методе полного контекста испытуемому предъявляется список всех элементов, и его просят подумать о важных качествах, характерных для разных групп элементов. Испытуемый классифицирует элементы различными способами по различным основаниям, например, его просят разделить элементы на две группы по сходству между собой, а свойство, по которому элементы сходны между собой является выявленным полюсом конструктора. Затем испытуемого могут попросить разделить элементы по другому основанию на другое количество групп.
- **Метод диад.** Последовательно выдаются две случайные гипотезы, пользователю требуется определить, чем они похожи. Это будет свидетельство, далее вводится положительный полюс и отрицательный.

Пример метода триад

Для проведения теста по методу репертуарных решеток Келли испытуемому предлагают двенадцать так называемых ролевых персонажей. Например «Я», «через 10 лет», «Мать», «Отец», «Друг», «Подруга», «Супруг», «Брат», «Сестра», «Человек, вызывающий неприятные чувства», «Тот, кому сочувствуешь», «Человек, оказавший сильное положительное влияние на жизнь» (аналог «Учителя», «Гуру»), «Нелюбимый политик». Для каждого конкретного испытуемого подбираются наиболее подходящие аналоги. Например, если человек не имеет брата или сестры (или обоих), то можно

заменить на близкого родственника (родственницу). Эти персонажи располагаются в таблице по горизонтали.

Оценка по методу репертуарной решетки

Затем формируют тройки из этих людей таким образом, чтобы каждый персонаж встречался как минимум два раза во всей решетке. В ячейках матрицы обозначают этих людей кружочками. Слева по вертикали пишут положительные качества, справа — отрицательные. Качества должны объединять двоих людей из тройки, третий должен отличаться от них. К примеру, «Мать» и «Отец» — добрые, «Нелюбимый политик» — злой. В ячейках персонажей, по которым проводится сравнение, необходимо, чтобы испытуемый выставлял оценки качеств. Для положительных: 1 — сильно выраженное, 2 — среднее, 3 — слабое. Для отрицательных: 7 — сильное, 6 — среднее, 5 — слабое

37. ДСМ-метод. Методы индукции и аналоги

Сокращение **ДСМ** расшифровывается **Джон Стюарт Милль**. Этот ученый в середине XIX в. предложил **принципы индуктивного метода**, которые положены в основу метода автоматического порождения гипотез. Способы установления причинно-следственных отношений, предложенные Миллем, основываются на идеях выявления сходства и различия в наблюдаемых ситуациях. Милль сформулировал следующие принципы индукции.

Принцип единственного различия: «Если после введения какого-либо фактора появляется (или после его удаления исчезает) известное явление, причем мы не вводим и не удаляем никакого другого обстоятельства, которое могло бы иметь влияние, то указанный фактор составляет причину явления».

Принцип единственного сходства, который гласит: «Если все обстоятельства явления, кроме одного, могут отсутствовать, не уничтожая этим явления, то это обстоятельство является причиной данного явления».

Принцип единственного остатка: «Если вычесть из какого-либо явления ту его часть, которая является следствием известных причин, то остаток явления есть следствие остальных причин». Рассмотрим схему:

$A, B, C \rightarrow D, E$

$A, B, C \rightarrow D, E$

.....

$B, C \rightarrow E$

После того, как из примеров $A, B, C \rightarrow D, E$ было «вычтено» причинно-следственное отношение $A \rightarrow D$, были получены наблюдения $B, C \rightarrow E$, на основании которых можно предположить, что B и C являются возможными причинами явления E .

Обобщенный алгоритм ДСМ-метода

Пусть задано множество причин $A = \{A_1, A_2, \dots, A_p\}$, множество следствий $B = \{B_1, B_2, \dots, B_m\}$ и множество оценок $Q = \{q_1, q_2, \dots, q_r\}$. Выражение вида $A_i \rightarrow B_j$ называется **положительной гипотезой**, обозначающей утверждение « A_i является причиной B_j с оценкой вероятности q_k ». **Отрицательной гипотезой** называется выражение вида $A_i \nrightarrow B_j$, обозначающей утверждение « A_i не является причиной B_j с оценкой вероятности q_k ». Гипотезы с оценками 1 и 0 можно рассматривать как явления, истинность и ложность которых твердо установлена. Остальные значения между 0 и 1 будем обозначать рациональными числами k/n , где $k = 1, 2, \dots, n-1$, а n характеризует число примеров.

1. На основе исходного множества положительных (+) и отрицательных (-) примеров (наблюдений) формируется набор гипотез, которые записываются в матрицы M^+ и M^- . Гипотезы формируются на основе выявления сходства и различия в примерах.
2. К исходному множеству примеров добавляются новые наблюдения, которые могут либо подтвердить выдвинутые гипотезы, либо опровергнуть их. При этом оценки гипотез меняются следующим образом. Если некоторая гипотеза имела оценку $q_k = k/n$, то при появлении нового примера $(n+1)$ проводится проверка на подтверждение этой гипотезы. В случае подтверждения оценка

определяется по формуле $q_k = (k+1)/(n+1)$, иначе $q_k = (k-1)/(n+1)$. В процессе накопления информации оценки гипотез могут приближаться либо к 1, либо к 0. Изменение оценок может иметь колебательный характер, что ведет к исключению таких гипотез из матриц M^+ и M^- .

3. Циклическое добавление примеров, сопровождается изменением оценок достоверности гипотез с периодическим изменением множеств M^+ и M^- .
4. Завершение процесса индуктивного вывода при выполнении условий окончания цикла. В качестве таких условий могут использоваться меры близости значений q_k к 0 или 1, а также дополнительные условия, связанные с ограничением времени.

Метод по аналогии. В современных модификациях ДСМ-метода используются также метод по аналогии. Вывод по аналогии – это вывод о принадлежности определенного признака исследуемому объекту или отношению на основе его сходства в существенных чертах с другим уже известным единичным объектом.

Схема вывода по аналогии

Объекту А присущи признаки P,Q,S,T

Объекту В присущи признаки P,Q,S, ...

Заключение: по-видимому объекту В присущ признак Т.

38. Понятие ЕЯ-систем. Четыре класса ЕЯ-систем. Структура обобщенной ЕЯ-системы.

Естественно-языковая система (ЕЯ-система) – это автоматизированная информационная система, использующая в качестве языка общения с пользователем естественный язык.

Четыре класса ЕЯ-систем

1. Интеллектуальные вопросно-ответные системы («Сколько нефти было перевезено железнодорожным транспортом в 1991 году?»)
 - Системы налагают достаточно жесткие ограничения на процесс общения
 - Вопросы были ориентированы на предметную область и пользователь не мог задать вопросы, касающиеся возможностей системы
 - Знания пользователя о языке общения и языке предметной области должны совпадать со знаниями, имеющимися у разработчиков системы
 - Жесткая структура диалога с инициативой у пользователя
 - Необходимость формирования запроса в одном предложении, что приводило к увеличению длины предложения и повышению вероятности ошибок пользователя.
2. Системы общения с базами данных (Сколько студентов факультета автоматики и вычислительной техники проживает в общежитии? - 152)
 - Общение в такой ЕЯ-системе ведется в виде связного диалога. Инициатива в основном принадлежит пользователю. Перехват инициативы допускается для уточнения незнакомых слов.
 - Система PARNAX, предназначенная для перевода запросов с итальянского языка на специализированный язык NATURAL системы СУБД ADABASE.
 - Система TEAM, предназначенная для перевода запросов с английского языка на язык реляционной базы данных.
3. Диалоговые системы решения задач
 - В таких ЕЯ-системах понимание входных высказываний происходит с учетом текущего состояния диалога и имеющихся у системы целей.
 - Здесь используется гибкая структура диалога с произвольным перехватом инициативы или альтернативная структура диалога с заранее предусмотренным перехватом инициативы.
 - Жесткая структура диалога используется в первых двух классах ЕЯ-систем (интеллектуальные вопросно-ответные системы и системы общения с базами данных).
4. Системы обработки связных текстов
 - Примерами таких систем являются RESEARCHER и TAILOR, которые образуют единый комплекс, позволяющий пользователю получать сведения из рефератов, патентов.

- RESEARCHER получает рефераты патентов на английском языке и строит на их основе базу знаний, делает обобщения из различных патентов. Внутреннее представление рефератов патентов включает следующие виды информации:
 - i. Иерархические описания, отражающие деление объектов на компоненты
 - ii. Физические и функциональные отношения между компонентами
 - iii. Свойства объектов
 - iv. Вопросно-ответные функции выполняет программа TAILOR

Обобщенная схема ЕЯ-систем



Диалоговый компонент можно рассматривать на 3-х уровнях

- общая структура - тип диалога и класс решаемых задач
- тематическая структура - структура конкретной задачи
- шаг диалога - взаимодействие участников в акте диалога

Компонент понимания высказываний

Данный компонент включает в себя анализ и интерпретацию высказывания. Анализ сводится к 3 составляющим:

1. Анализ слов
2. Анализ предложений
3. Анализ связных текстов

Компонент генерации высказываний

1. Определяется информация, которая должна быть сообщена пользователю
2. Выделяются аспекты, интересующие пользователя
3. Разбивается сообщаемая информация на части в соответствии с будущими предложениями и устанавливается последовательность этих частей.
4. На втором этапе решаются следующие подзадачи:
 - a. построение синтаксиса структуры отдельных предложений;
 - b. морфологический синтез словоформ.

39. Рассуждения по прецедентам. CBR-цикл. Достоинства и недостатки использования прецедентов. Примеры реализации экспертных систем на базе прецедентов.

Рассуждения по прецедентам – это метод формирования умозаключений, опирающийся не на логический вывод от исходных посылок, а на поиск и анализ случаев формирования подобных умозаключений в прошлом.

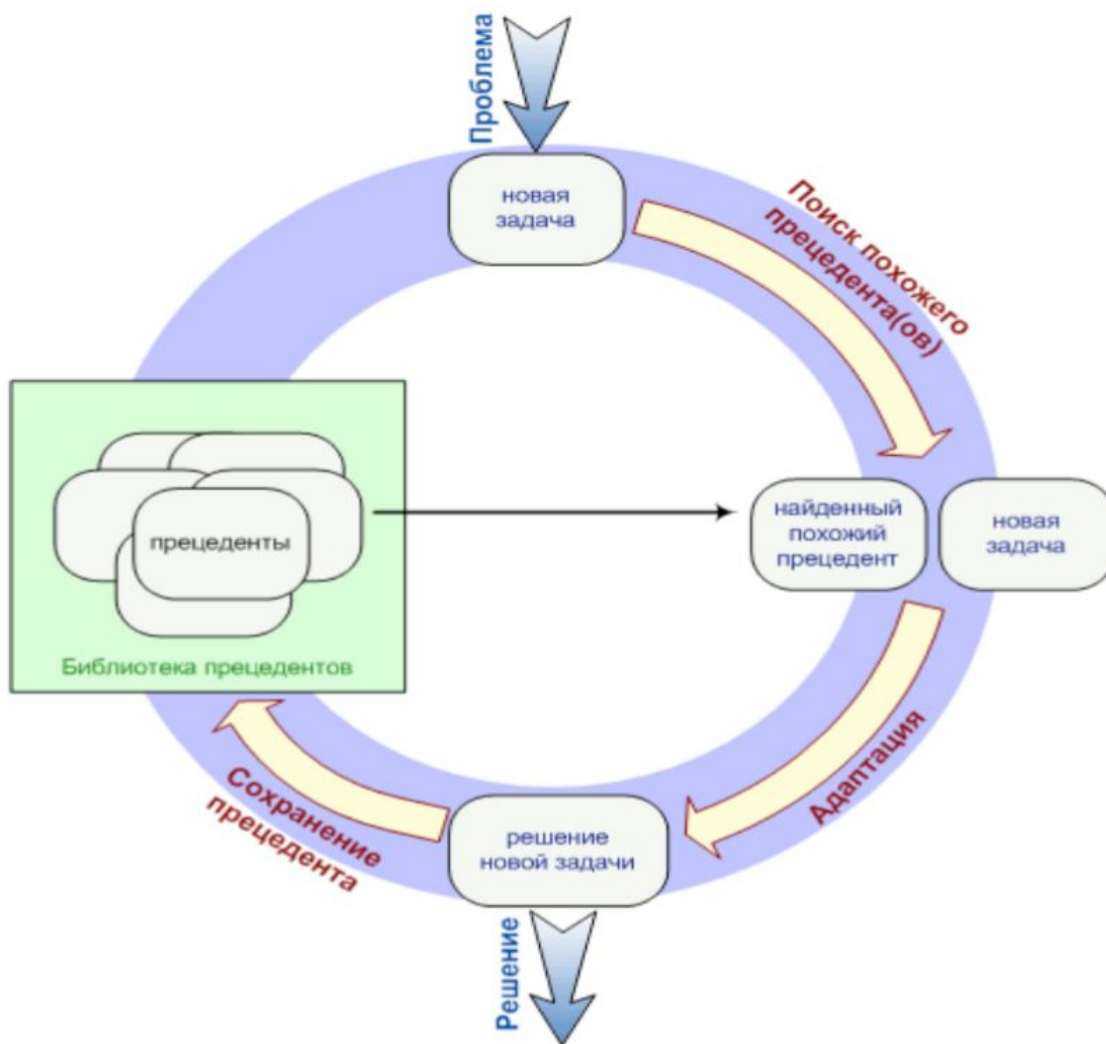
Прецедент можно определить, как единичную запись предыдущего опыта. Какую именно информацию содержит такая запись, зависит от предметной области и целей использования прецедента.

Вывод на основе прецедентов (CBR – Case-Based Reasoning) является подходом, позволяющим решить новую задачу, используя или адаптируя решение уже известной задачи.

В случае применения CBR-метода для решения задач прецедент содержит, по меньшей мере, постановку задачи и способ ее решения. Множество всех прецедентов, накопленных в процессе работы CBR-метода, формируют информационное хранилище, называемое **библиотекой прецедентов**.

CBR-цикл:

1. Поиск похожего прецедента(ов) – поиск прецедента(ов), у которых постановка задачи наиболее похожа на описание новой задачи
2. Адаптация – получение на базе найденного прецедента решения для новой задачи. Этот этап может также включать проверку полученного нового решения на корректность и толерантность к ошибкам, и, возможно, дополнительную коррекцию решения
3. Сохранение прецедента – сохранение той части полученного опыта, которая может оказаться полезной для решения новых задач (пополнение или корректировка библиотеки прецедентов).



Достоинства

- **Легкость приобретения знаний** (в противоположность системам, основанным на правилах). Создание системы, основанной на правилах, требует таких трудоемких этапов как получение, формализация и обобщение экспертных знаний, верификация системы на корректность и полноту. В системах, основанных на прецедентах, приобретение знаний происходит путем формального описания случаев из практики (нет необходимости обобщения, и вытекающей отсюда угрозы пере обобщения).
- **Возможность объяснения полученного решения** (в противоположность системам, основанным на нейронных сетях). CBR-система может объяснить полученное решение путем демонстрации успешного прецедента с отражением показателей сходства и рассуждений, использовавшихся при адаптации прецедента к новой ситуации. Такое объяснение может быть даже лучше, чем объяснения, выдаваемые системами, основанными на правилах. Последние иногда выдают очень длинные последовательности рассуждений, а сами правила конечному пользователю (в отличие от эксперта) могут казаться неочевидными или слишком сложными.
- **Возможность работы в предметных областях, которые невозможно полностью понять, объяснить или смоделировать.**

- **Возможность обучения в процессе работы.** Причем обучение будет происходить только в определенных направлениях, которые реально встречаются на практике и востребованы (нет избыточности).
- **Возможность избежать повторения ошибки** (обучение сбоям и их причинам для избегания их появления в дальнейшем).
- **Возможность получения решений путем модификации прецедентов** позволяет уменьшить объем вычислений в предметных областях, где генерация решения «с нуля» требует больших усилий.

Недостатки

- Метод применим только в областях, где выполняется принцип регулярности и имеет место повторяемость видов задач. Если все время решаются принципиально новые задачи или если решения сходных задач различны, то метод на базе прецедентов неприемлем.
- Некомпактное (без обобщения) хранение знаний (опыта).
- Сложность и специфичность процессов поиска подобных случаев и адаптации решения.

Примеры реализации экспертных систем на базе прецедентов

- Корпорацией NEC для управления общей корпоративной базой данных разработан Case-метод [Kitano 96, Leake 96]. Утверждается, что накопление данных в виде прецедентов, выявление закономерностей и последующее внедрение полученных знаний повышают общий профессиональный уровень сотрудников корпорации.
- В Университете города Росток (ФРГ) разработана система для прогноза эпидемий [Bull 97]. В качестве прецедентов используются последовательности т.н. сценариев, в которые входят новые случаи заболеваний, нагрузка на органы здравоохранения, контекстная информация о сезоне и погоде. Чтобы обнаружить "заметные" изменения в последовательностях сценариев, используется статистический метод добычи данных, называемый G-тест. Для сверки результата используются методы вывода по прецедентам.

40. Интеллектуальный анализ данных

Интеллектуальный анализ данных (англ. Data Mining) — **выявление скрытых закономерностей** или взаимосвязей между переменными **в больших массивах необработанных данных**. Как правило подразделяется на задачи классификации, моделирования и прогнозирования. **Процесс автоматического поиска закономерностей в больших массивах данных**. Термин Data Mining Введен Григорием Пятецким-Шапиро в 1989 году.

ИАД включает методы и модели статистического анализа и машинного обучения, отходя от них в сторону **автоматического анализа данных**. Инструменты ИАД позволяют проводить анализ данных предметными специалистами (аналитиками), не владеющими соответствующими математическими знаниями.

Основные технологии интеллектуального анализа данных

- Правила вывода
- Нейронные сети
- Нечеткая логика
- Визуализация
- Статистика
- Кластеризация
- Интегрированные технологии

Для анализа данных используются средства многомерного хранения и аналитической обработки данных (**OLAP - Online Analytical Processing**), представляющие бизнес-операции в виде фактов (объем продаж, число единиц на складе и т.д.) и измерений (время, география, поставщик, покупатель, товар и т.д.). Средства OLAP позволяют осуществлять стратегический обзор ситуации и в реальном времени получать ответы на вопросы, интересующие аналитика.

Стадии ИАД

1. Выявление закономерностей (свободный поиск)
2. Использование выявленных закономерностей для предсказания неизвестных значений (прогностическое моделирование)
3. Анализ исключений, предназначенный для выявления и толкования аномалий в найденных закономерностях

Статистические методы Data Mining

Статистические методы представляют собой четыре взаимосвязанных раздела:

1. Предварительный анализ природы статистических данных (проверка гипотез стационарности, нормальности, независимости, однородности, оценка вида функции распределения, ее параметров и т.п.)
2. Выявление связей и закономерностей (линейный и нелинейный регрессионный анализ, корреляционный анализ и др.)
3. Многомерный статистический анализ (линейный и нелинейный дискриминантный анализ, кластерный анализ, компонентный анализ, факторный анализ и др.)
4. Динамические модели и прогноз на основе временных рядов

Кибернетические методы Data Mining

1. Искусственные нейронные сети (распознавание, кластеризация, прогноз)
2. Эволюционное программирование (в т.ч. алгоритмы метода группового учета аргументов)
3. Генетические алгоритмы (оптимизация)
4. Ассоциативная память (поиск аналогов, прототипов)
5. Нечеткая логика
6. Деревья решений
7. Системы обработки экспертных знаний

Задачи решаемые Data Mining

1. Классификация – отнесение входного вектора (объекта, события, наблюдения) к одному из заранее известных классов.
2. Кластеризация – разделение множества входных векторов на группы (кластеры) по степени «похожести» друг на друга.
3. Регрессия – установление зависимости непрерывными входным и выходным векторами.
4. Ассоциация – поиск повторяющихся паттернов. Например, поиск устойчивых связей в корзине покупателя (market basket analysis) – вместе с пивом покупают орешки.
5. Последовательные шаблоны – аналогично задаче ассоциации, но с учетом временной составляющей. Например, поиск причинно-следственных связей.
6. Прогнозирование – аналогично задаче регрессии, но с учетом временной составляющей. Например, прогноз трендов финансовых показателей.
7. Анализ отклонений – выявление наиболее нехарактерных паттернов. Например, выявление нетипичной сетевой активности позволяет обнаружить вредоносные программы.

Сферы применения Data Mining

- **Применение Data Mining для решения бизнес-задач.** Основные направления: банковское дело, финансы, страхование, CRM, производство, телекоммуникации, электронная коммерция, маркетинг, фондовый рынок и другие.
- **Применение Data Mining для решения задач государственного уровня.** Основные Направления: поиск лиц, уклоняющихся от налогов; средства в борьбе с терроризмом.
- **Применение Data Mining для научных исследований.** Основные направления: медицина, биология, молекулярная генетика и геномная инженерия, биоинформатика, астрономия, прикладная химия, исследования, касающиеся наркотической зависимости, и другие.
- **Применение Data Mining для решения Web-задач.** Основные направления: поисковые машины (search engines), счетчики и другие.

41. Понятие онтологии. Классификация онтологий. Языки описания онтологий. Основные правила разработки онтологий. Примеры существующих онтологий.

Онтология в информатике – это попытка всеобъемлющей и подробной формализации некоторой области знаний с помощью концептуальной схемы. Обычно такая схема состоит из структуры данных, содержащей все релевантные классы объектов, их связи и правила (теоремы, ограничения), принятые в этой области.

Основные сферы применения – моделирование бизнес-процессов, семантическая паутина (англ. Semantic Web), искусственный интеллект.

Основные понятия онтологии:

- **Классы** онтологии составляют таксономию – иерархию понятий по отношению вложения.
- **Объекты** в онтологии могут иметь атрибуты.
- **Отношения выражают свойства понятий** (классов) и их зависимости с другими понятиями (классами). Отношение можно представить как атрибут, значением которого является другое понятие (класс).
- **Экземпляры или индивиды** – элементы нижнего уровня онтологии. Экземпляры могут представлять собой как физические объекты (люди, дома), так и абстрактные (числа, слова).
- Отношения и атрибуты также в некоторых источниках называют **слотами** (или ролями).
- Онтология вместе с набором индивидуальных экземпляров классов образует **базу знаний**.

Классификация онтологий

- По степени формальности:
 - Формальные: формальные таксономии, онтологии свойств на основе фреймов
 - Неформальные: тезаурусы, словари терминов, каталоги
- По наполнению, содержанию:
 - Общие онтологии
 - Онтологии задач
 - Предметные онтологии
- По цели создания:
 - Онтологии представления
 - Онтологии верхнего уровня
 - Онтологии предметных областей
 - Прикладные онтологии

Языки описания онтологий

- **OWL** – Web Ontology Language, стандарт W3C, язык для семантических утверждений, разработанный как расширение RDF и RDFS

- KIF (англ. Knowledge Interchange Format – формат обмена знаниями) – основанный на S-выражениях (символических выражениях данных) синтаксис для логики
- Common Logic (CL) – преемник KIF (стандартизован – ISO/IEC 24707:2007)
- CycL – онтологический язык, использующийся в проекте Cyc. Основан на исчислении предикатов с некоторыми расширениями более высокого порядка

Основные правила разработки онтологий

1. **Не существует единственного правильного способа** моделирования предметной области – всегда существуют жизнеспособные альтернативы. Лучшее решение почти всегда зависит от предполагаемого приложения и ожидаемых расширений.
2. **Разработка онтологии – это обязательно итеративный процесс.** В процессе создания важно возвращаться к уже созданным классам и отношениям и уточнять и добавлять информацию в случае необходимости. Иногда полезно вносить и кардинальные изменения для улучшения общей структуры онтологии.
3. **Понятия в онтологии должны быть близки к объектам** (физическим или логическим) и отношениям в интересующей предметной области. Скорее всего, это существительные (объекты) или глаголы (отношения) в предложениях, которые описывают предметную область.

Примеры существующих онтологий

- Biochemical ontologies (биохимические онтологии), содержащие более 30 онтологий, которые описывают атомы, молекулы, связи и процессы, связанные с ними.
- Infrastructure Product Ontology – онтология, описывающая коммунальную инфраструктуру и охватывает пять секторов коммунальных услуг: водоснабжение, канализация, газ, электричество и связь.
- Protein Ontology – онтология белков, описывающая все декларативные знания о них, в том числе их классификацию, что позволяет строить логические рассуждения.

42. Гибридные экспертные системы

Под гибридной системой принято понимать систему, в которой для решения задачи используется более одного метода имитации интеллектуальной деятельности человека. Интеграция технологий дает возможность использовать индивидуальную силу технологий для решения частей задачи. Выбор технологий зависит от самой задачи. Гибридные системы позволяют эффективно соединять формализованные и неформализованные знания.

В качестве компонентов такой системы могут выступать:

- Нечеткая логика
- Нейрокомпьютинг
- Генетические вычисления
- Вероятностные вычисления

“Мягкие” системы можно также считать гибридными системами.

Интерпретация мягкой системы может быть следующей:

МЭС = управление неопределенностью + обучаемость + самоадаптация

А мягкие вычисления это нечеткие системы + нейронные сети + генетические алгоритмы

Мягкая система обладает следующими особенностями:

- Использует статистические данные, которые интерпретируются как обучающие выборки для нечетких нейронных сетей
- Представляет знания в виде лингвистических переменных(функций принадлежности), нечетких продукций
- настройка и подбор параметров функций принадлежности выполняется с помощью генетических алгоритмов.

МЭС должна представлять инструментальную среду для экспертной деятельности, содержащую совокупность различных программ с общей логикой работы.

Нечеткая нейронная сеть представляет собой средство для извлечения знаний для МЭС. Она извлекает знания из статистических выборок, интерпретируя их как обучающие выборки для обучения НС.

Представление знаний в МЭС. Если использовать нечеткие нейронные сети на этапе извлечения знаний, то, кроме функций принадлежности и нечетких продукций, порождается совокупность обученных НС, которые входят в базу знаний МЭС. Оптимизация (редукция) множества извлеченных правил выполняется на основе генетического алгоритма

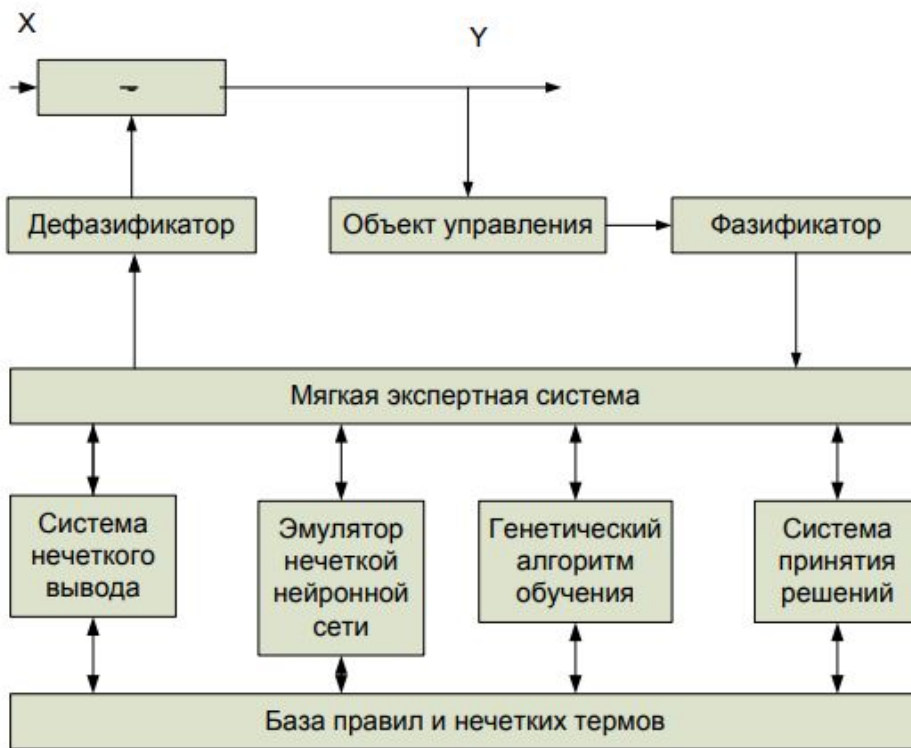


Рис. 6.1. Структурная схема мягкой экспертной системы

База знаний МЭС содержит следующие компоненты:

- функции принадлежности;
- нечеткие продукции;
- обученные нечеткие нейронные сети;
- процедуры интерпретации хромосом генетического алгоритма;
- критерий оптимальности.