

МППР

Правила оформления

1. Вопрос скопировать из списка, вставить под заголовок “Ответы” и пометить как заголовок 4
2. Ответ на вопрос - обычный текст, шрифт Arial 11
3. Позже вместо списка вопросов сделаю оглавление, чтобы было удобнее

Вопросы

1. Основные направления искусственного интеллекта.
2. Интеллектуальные системы, как основа новых информационных технологий.
3. Понятие искусственного интеллекта. Определения и пути создания искусственного интеллекта.
4. Бионическое, эвристическое и эволюционное направления искусственного интеллекта.
5. Классификация систем искусственного интеллекта.
6. Основные понятия и определения формализации знаний в интеллектуальных системах.
7. Проблемная область формализации знаний в интеллектуальных системах.
8. Данные и знания. Свойства и характеристики знаний.
9. Процедурные и декларативные знания. Классификация знаний. Формальные языки.
10. Модели представления знаний. Классификация моделей знаний.
11. Методы инженерии знаний. Теоретические аспекты получения знаний. Методы извлечения явных и скрытых знаний.
12. Проблемы структурирования знаний. Проблема сборки знаний в единую модель предметной области.
13. Логическая модель представления знаний.
14. Исчисление высказываний.
15. Основы исчисления предикатов.
16. Семантические сети. Понятия, события и свойства. Виды семантических сетей.
17. Семантические сети Куиллиана. Интерпретация семантической сети.
18. Фреймы. Слоты. Структура фрейма. Указатели наследования и типа данных.
19. Слоты. Значение слота. Присоединенные процедуры.
20. Модель, основанная на правилах. Антецедент и консеквент. Продукционная система Поста и ее ограничения.
21. Марковский и rete-алгоритм. Триплет объект-атрибут-значение. Достоинства и недостатки продукционных систем.
22. База правил, рабочая память и механизм вывода. Компонента вывода и управляющая компонента. Операции цикла вывода.
23. Демоны, присоединенные процедуры и механизм наследования в интеллектуальной системе с фреймовой моделью представления знаний. Правила вывода в логических моделях.
24. Структура и классификация экспертных систем (ЭС), отличие ЭС от других программных продуктов, цикл работы ЭС, технология проектирования и разработки ЭС.

25. Нечеткая логика. Нечеткие множества. Строгое представление нечетких множеств. Основные характеристики нечетких множеств.
26. Операции над нечеткими множествами. Логические операции. Алгебраические операции.
27. Четкое множество « α -уровня». Нечеткая и лингвистическая переменные. Нечеткие числа.
28. Особенности человеческого восприятия информации и современных систем управления. Вычислительный интеллект (ВИ). Методы ВИ.
29. Искусственные нейронные сети. Основные проблемы, решаемые искусственными нейронными сетями.
30. Биологический нейрон. Искусственный нейрон. Активационная функция. Классификация и свойства искусственных нейронных сетей. Обучение искусственных нейронных сетей.
31. Теорема Колмогорова. Алгоритм обучения персептранона. Линейная разделимость и персептранная представляемость.
32. Нейронная сеть обратного распространения. Алгоритм обучения сети обратного распространения.
33. Нейронная сеть встречного распространения (сеть Кохонена). Алгоритм обучения сети Кохонена.
34. Двухслойная сеть встречного распространения и ее обучение. Входные и выходные звезды Гроссберга и их обучение.
35. Двухслойная сеть встречного распространения. Алгоритм обучения сети встречного распространения.
36. Стохастические методы обучения. Обучение Больцмана. Алгоритм обучения Больцмана. Обучение Коши.
37. Сети с обратными связями. Сеть Хопфилда.
38. Правило обучения Хебба. Ортогонализация образов.
39. Двунаправленная ассоциативная память.
40. Адаптивная резонансная теория.
41. Методы оптимизации комбинаторных задач различной степени сложности. Генетические алгоритмы.
42. Базовый генетический алгоритм. Последовательные модификации базового генетического алгоритма.
43. Параллельные модификации базового генетического алгоритма. Классификация генетических алгоритмов.
44. Когнитивная компьютерная графика. Определение. Степень организованности информации.
45. Задача сжатия информации. Анаморфозы (определение).
46. Метод анаморфирования.
47. Проблемы реализации анаморфоз.
48. Численные методы построения анаморфоз.
49. Основные законы гибридного интеллекта.
50. Основные методы гибридизации.
51. Общий подход к построению гибридной интеллектуальной системы.
52. Принципы построения гибридных интеллектуальных систем.

Ответы

0. Пример -

текст

1. Основные направления искусственного интеллекта.

Разработка интеллектуальных информационных систем или систем, основанных на знаниях.

Это одно из главных направлений ИИ. Основной целью построения таких систем являются выявление, исследование и применение знаний высококвалифицированных экспертов для решения сложных задач, возникающих на практике. При построении систем, основанных на знаниях (СОЗ), используются знания, накопленные экспертами в виде конкретных правил решения тех или иных задач. Это направление преследует цель имитации человеческого искусства анализа неструктурированных и слабоструктурированных проблем. В данной области исследований осуществляется разработка моделей представления, извлечения и структурирования знаний, а также изучаются проблемы создания баз знаний (БЗ), образующих ядро СОЗ. Частным случаем СОЗ являются экспертные системы (ЭС).

Разработка естественно-языковых интерфейсов и машинный перевод. Проблемы компьютерной лингвистики и машинного перевода разрабатываются в ИИ с 1950-х гг. Системы машинного перевода с одного естественного языка на другой обеспечивают быстроту и систематичность доступа к информации, оперативность и единообразие перевода больших потоков, как правило, научно-технических текстов. Системы машинного перевода строятся как интеллектуальные системы, поскольку в их основе лежат БЗ в определенной предметной области и сложные модели, обеспечивающие дополнительную трансляцию «исходный язык оригинала - язык смысла - язык перевода». Они базируются на структурно-логическом подходе, включающем последовательный анализ и синтез естественно-языковых сообщений. Кроме того, в них осуществляется ассоциативный поиск аналогичных фрагментов текста и их переводов в специальных базах данных (БД). Данное направление охватывает также исследования методов и разработку систем, обеспечивающих реализацию процесса общения человека с компьютером на естественном языке (так называемые системы ЕЯ-общения).

Генерация и распознавание речи. Системы речевого общения создаются в целях повышения скорости ввода информации в ЭВМ, "разгрузки зрения и рук, а также для реализации речевого общения на значительном расстоянии. В таких системах под текстом понимают фонемный текст (как слышится).

Обработка визуальной информации. В этом научном направлении решаются задачи обработки, анализа и синтеза изображений. Задача обработки изображений связана с трансформированием графических образов, результатом которого являются новые изображения. В задаче анализа исходные изображения преобразуются в данные другого типа, например, в текстовые описания. При синтезе изображений на вход системы поступает алгоритм построения

изображения, а выходными данными являются графические объекты (системы машинной графики).

Обучение и самообучение. Эта актуальная область ИИ включает модели, методы и алгоритмы, ориентированные на автоматическое накопление и формирование знаний с использованием процедур анализа и обобщения данных. К данному направлению относятся не так давно появившиеся системы добычи данных (Data-mining) и системы поиска закономерностей в компьютерных базах данных (Knowledge Discovery).

Распознавание образов. Это одно из самых ранних направлений ИИ, в котором распознавание объектов осуществляется на основании применения специального математического аппарата, обеспечивающего отнесение объектов к классам, а классы описываются совокупностями определенных значений признаков.

Игры и машинное творчество. Машинное творчество охватывает сочинение компьютерной музыки, стихов, интеллектуальные системы для изобретения новых объектов. Создание интеллектуальных компьютерных игр является одним из самых развитых коммерческих направлений в сфере разработки программного обеспечения. Кроме того, компьютерные игры предоставляют мощный арсенал разнообразных средств, используемых для обучения.

Программное обеспечение систем ИИ. Инструментальные средства для разработки интеллектуальных систем включают специальные языки программирования, ориентированные на обработку символьной информации (LISP, SMALLTALK, РЕФАЛ), языки логического программирования (PROLOG), языки представления знаний (OPS 5, KRL, FRL), интегрированные программные среды, содержащие арсенал инструментальных средств для создания систем ИИ (KE, ARTS, GURU, G2), а также оболочки экспертных систем (BUILD, EMYCIN, EXSYS Professional, ЭКСПЕРТ), которые позволяют создавать прикладные ЭС, не прибегая к программированию.

Новые архитектуры компьютеров. Это направление связано с созданием компьютеров на фон-неймановской архитектуре, ориентированных на обработку символьной информации. Известны удачные промышленные решения параллельных и векторных компьютеров, однако в настоящее время они имеют весьма высокую стоимость, а также недостаточную совместимость с существующими вычислительными средствами.

Интеллектуальные роботы. Создание интеллектуальных роботов составляет конечную цель робототехники. В настоящее время в основном используются программируемые манипуляторы с жесткой схемой управления, названные роботами первого поколения. Несмотря на очевидные успехи отдельных разработок, эра интеллектуальных автономных роботов пока не наступила. Основными сдерживающими факторами в разработке автономных роботов являются нерешенные проблемы в области интерпретации знаний, машинного зрения, адекватного хранения и обработки трехмерной визуальной информации.

2. Интеллектуальные системы, как основа новых информационных технологий

Технически интеллектуальные системы - технические и программные системы, ориентированные на решение большого и очень важного класса задач, называемых неформализованными. Система становится интеллектуальной, если в ней:

- данные заменяются на знания;
- алгоритмы функционирования заменяются на методы искусственного интеллекта.

ИС особенно эффективны в применении к слабо структурированным задачам, в которых отсутствует строгая формализация, и для решения которых применяются эвристические процедуры, позволяющие в большинстве случаев получить решение

На сегодняшний день не существует единого определения, которое однозначно описывает эту научную область. Среди многих точек зрения на нее доминируют следующие три.

Согласно первой исследования в области ИИ относятся к фундаментальным, в процессе которых разрабатываются новые модели и методы решения задач, традиционно считавшихся интеллектуальными и не поддававшихся ранее формализации и автоматизации.

Согласно второй точке зрения это направление связано с новыми идеями решения задач на ЭВМ, с разработкой новых технологий программирования и с переходом к компьютерам не фон-неймановской архитектуры.

Третья точка зрения, наиболее прагматическая, основана на том, что в результате исследований, проводимых в области ИИ, появляется множество прикладных систем, способных решать задачи, для которых ранее создаваемые системы были непригодны. По последней трактовке ИИ является экспериментальной научной дисциплиной, в которой роль эксперимента заключается в проверке и уточнении интеллектуальных систем, представляющих собой аппаратно-программные информационные комплексы.

3. Понятие искусственного интеллекта. Определения и пути создания искусственного интеллекта.

Под интеллектом мы понимаем способность любого организма (или устройства) достигать некоторой измеримой степени успеха при поиске одной из многих возможных целей в обширном многообразии сред. Необходимо отличать знания от интеллекта, имея в виду, что знания - полезная информация, накопленная индивидуумом, а интеллект - это его способность предсказывать состояние внешней среды в сочетании с умением преобразовывать каждое предсказание в подходящую реакцию, ведущую к заданной цели.

Определения и пути создания ИИ:

1. Искусственный интеллект — наука и технология создания интеллектуальных машин, особенно интеллектуальных компьютерных программ. ИИ связан со сходной задачей использования компьютеров для

понимания человеческого интеллекта, но не обязательно ограничивается биологически правдоподобными методами.

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

Основной проблемой искусственного интеллекта является разработка методов представления и обработки знаний.

К программам искусственного интеллекта относятся:

1. игровые программы (стохастические, компьютерные игры);
2. естественно-языковые программы - машинный перевод, генерация текстов, обработка речи;
3. распознающие программы - распознавание почерков, изображений, карт;
4. программы создания и анализа графики, живописи, музыкальных произведений.

Выделяются следующие направления искусственного интеллекта:

1. экспертные системы;
2. нейронные сети;
3. естественно-языковые системы;
4. эволюционные методы и генетические алгоритмы;
5. нечеткие множества;
6. системы извлечения знаний.

Экспертные системы - система, способная частично заменить специалиста-эксперта в разрешении проблемной ситуации, ориентирована на решение конкретных задач.

Нейронные сети реализуют нейросетевые алгоритмы. Делятся на:

- сети общего назначения, которые поддерживают около 30 нейросетевых алгоритмов и настраиваются на решение конкретных задач;
- объектно-ориентированные - используемые для распознавания символов, управления производством, предсказание ситуаций на валютных рынках;
- гибридные - используемые вместе с определенным программным обеспечением.

Естественно-языковые (ЕЯ) системы делятся на:

- программные продукты естественного языкового интерфейса в БД (представление естественно-языковых запросов в SQL-запросы);
- естественно-языковой поиск в текстах, содержательное сканирование текстов (используется в поисковых системах Internet, например, Google);
- масштабируемые средства распознавания речи (портативные синхронные переводчики);
- средства голосового ввода команд и управления (безлюдные производства);
- компоненты речевой обработки, как сервисные средства программного обеспечения (ОС Windows).

Нечёткие множества - реализуют логические отношения между данными. Эти программные продукты используются для управления экономическими объектами, построения экспертных систем и систем поддержки принятия решений.

Генетические алгоритмы - это методы анализа данных, которые невозможно проанализировать стандартными методами. Как правило, используются для обработки больших объёмов информации, построения прогнозных моделей. Используются в научных целях при имитационном моделировании.

Системы извлечения знаний - используются для обработки данных из информационных хранилищ.

4. Бионическое, эвристическое и эволюционное направления искусственного интеллекта.

Принято различать три основные пути моделирования интеллекта и мышления:

- классический (бионический);
- эвристического программирования;
- эволюционного моделирования.

Бионическое моделирование

Непосредственное моделирование человеческого мозга (т.е. моделирование каждой нервной клетки и связей между ними) с целью создания автоматов, обладающих интеллектом, чрезвычайно сложно. Мозг представляет собой самую сложную и лишь частично изученную структуру. Сложнейшее переплетение связей коры головного мозга практически не поддаются расшифровке. Известно лишь примерное расположение зон мозга, отвечающих за ту или иную функцию.

Попытки смоделировать работу головного мозга соединением между собой множества процессоров подобно нейронной сети, показали, что некоторое увеличение скорости и потока обрабатываемой информации идет лишь до уровня одного - двух десятков процессоров, а затем начинается резкий спад производительности. Процессоры как бы "теряются", перестают контролировать ситуацию или проводят большую часть времени в ожидании соседа. Некоторых успехов удалось добиться лишь в приборах, работающих в "двумерном варианте", т.е. обрабатывающих не последовательную, а параллельную информацию, например, в системах распознаваниях образов. В них одна плоскость данных одновременно взаимодействует с другой, причем количество единиц информации может достигать нескольких миллионов. Таким образом происходит единовременный охват изучаемого объекта, а не последовательное изучение его частей.

Эвристическое программирование

Второй подход к решению задачи искусственного интеллекта связан с эвристическим программированием и решает задачи, которые в общем можно назвать творческими.

Практичность этого метода заключается в радикальном уменьшении вариантов, необходимых при использовании метода проб и ошибок. Правда, всегда существует вероятность упустить наилучшее решение, так что говорят, что этот метод предлагает решения с некоторой вероятностью правильности.

Обычно используют два метода: метод анализа целей и средств и метод планирования. Первый заключается в выборе и осуществлении таких операций, которые последовательно уменьшают разницу между исходным и конечным состоянием задачи. Во втором методе вырабатывается упрощенная формулировка исходной задачи, которая также решается методом анализа целей и средств. Один из полученных вариантов дает решение исходной задачи.

Эволюционное моделирование

Третий подход является попыткой смоделировать не то, что есть, а то, что могло бы быть, если бы эволюционный процесс направлялся в нужном направлении и оценивался предложенными критериями.

Идея эволюционного моделирования сводится к экспериментальной попытке заменить процесс моделирования человеческого интеллекта моделированием процесса его эволюции. При моделировании эволюции предполагается, что разумное поведение предусматривает сочетание способности предсказывать состояние внешней среды с умением подобрать реакцию на каждое предсказание, которое наиболее эффективно ведет к цели.

Этот метод открывает путь к автоматизации интеллекта и освобождению от рутинной работы. Это высвобождает время для проблемы выбора целей и выявления параметров среды, которые заслуживают исследования. Такой принцип может быть применен для использования в диагностике, управлении неизвестными объектами, в игровых ситуациях.

Доп. инфа:

Итак, существуют три пути моделирования интеллекта: бионический, эвристический и эволюционный. В зависимости от использованных средств можно выделить **три фазы в исследовании**.

Первая фаза - создания устройств, выполняющих большое число логических операций с высоким быстродействием.

Вторая фаза включает разработку проблемно-ориентированных языков для использованного на оборудовании, созданном в первой фазе.

Третья фаза наиболее выражена в эволюционном моделировании. В ходе развития этой фазы отпадает необходимость в точной формулировке постановки задачи, т.е. задачу можно сформулировать в терминах цели и допустимых затрат, а метод решения будет найден самостоятельно по этим двум параметрам.

5. Классификация систем искусственного интеллекта.

Интеллектуальная информационная система (ИИС) основана на концепции использования базы знаний для генерации алгоритмов решения прикладных задач различных классов в зависимости от конкретных информационных потребностей пользователей.

Для ИИС характерны следующие признаки:

- развитые коммуникативные способности;
- умение решать сложные плохо формализуемые задачи;
- способность к самообучению;
- адаптивность.

Каждому из перечисленных признаков условно соответствует свой класс ИИС.

Различные системы могут обладать одним или несколькими признаками интеллектуальности с различной степенью проявления.

Средства ИИ могут использоваться для реализации различных функций, выполняемых ИИС. На рис. 1.1 приведена классификация ИИС, признаками которой являются следующие интеллектуальные функции:



Рис. 1.1. Классификация интеллектуальных информационных систем

- коммуникативные способности - способ взаимодействия конечного пользователя с системой;
- решение сложных плохо формализуемых задач, которые требуют построения оригинального алгоритма решения в зависимости от конкретной ситуации, характеризующейся неопределенностью и динамичностью исходных данных и знаний;
- способность к самообучению - умение системы автоматически извлекать знания из накопленного опыта и применять их для решения задач;
- адаптивность - способность системы к развитию в соответствии с объективными изменениями области знаний.

6. Основные понятия и определения формализации знаний в интеллектуальных системах.

Формализация знаний — приведение знаний, поступающих из разных источников, к одинаковой форме, для повышения их доступности.

Понятие «Знание» в ИИ

Знания - это совокупность сведений о сущностях (объектах, предметах) реального мира, их свойствах и отношениях между ними в определенной *предметной области*.

Иными словами, **знания** - это выявленные закономерности *предметной области* (принципы, связи, законы), позволяющие решать задачи в этой области. С точки зрения *ИИ* **знания** можно определить как формализованную информацию, на которую ссылаются в процессе логического вывода.

В этом случае, под *ПрО* понимается область человеческих знаний, в терминах которой формулируются задачи и в рамках которых они решаются. Т.е. *ПрО*

представляется описанием части реального мира, которое в силу своей приближенности рассматривается как ее *информационная модель*.

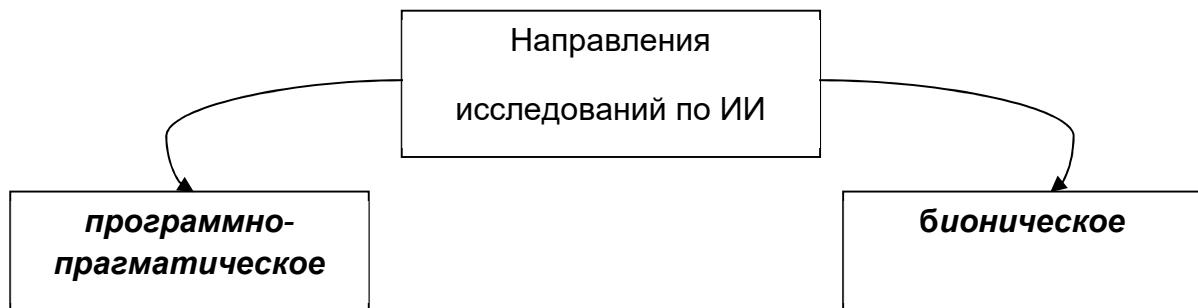
Проблемная область - это содержательное описание в терминах ПрО проблемы совместно с комплексом условий, факторов и обстоятельств, вызвавших ее возникновение.

В исследованиях по ИИ можно выделить два основных направления:

программно-прагматическое («не имеет значения, как устроено «мыслящее» устройство, главное, чтобы на заданные входные воздействия оно реагировало, как человеческий мозг»).

К ярким представителем программно-прагматического направления можно отнести экспертные системы - сложные программные комплексы, аккумулирующие знания специалистов - экспертов для обеспечения высокоэффективного решения неформализованных задач в узкой предметной области.

бионическое («единственный объект, способный мыслить - это человеческий мозг, поэтому любое «мыслящее» устройство должно каким-то образом воспроизвести его структуру»). В рамках данного подхода сформировалась новая наука *нейроинформатика*.



- эвристического программирования;
- эволюционного моделирования

7. Проблемная область формализации знаний в интеллектуальных системах.

Общепризнанного определения знания, как и определения искусственного интеллекта, не существует.

Общее определение трактует **знание как всю совокупность данных (информации), необходимую для решения задачи**. Данные в привычном понимании также являются знаниями. Однако знания в информационном плане не ограничиваются рамками данных.

(Данные — представление фактов и идей в формализованном виде, пригодном для передачи и обработки в некотором информационном процессе.)

В полном объеме **информация**, содержащаяся в **знаниях**, должна включать сведения о: **системе понятий предметной области, в которой решаются задачи; системе понятий формальных моделей, на основе которых решаются задачи;**

соответствии систем понятий, упомянутых выше; методах решения задачи; текущем состоянии предметной области.

Из перечисленных компонентов только последний (**текущее состояние предметной области**) в явном виде соответствует понятию "**данные**".

В целом обо всей приведенной выше информации иногда говорят, что она составляет **проблемную область** решаемой задачи.

8. Данные и знания. Свойства и характеристики знаний.

Понятие "Знание" в ИИ

Знания - это совокупность сведений о сущностях (объектах, предметах) реального мира, их свойствах и отношениях между ними в определенной предметной области.

Иными словами, знания - это выявленные закономерности предметной области (принципы, связи, законы), позволяющие решать задачи в этой области. С точки зрения ИИ знания можно определить как формализованную информацию, на которую ссылаются в процессе логического вывода.

Данные и знания

Общепризнанного определения знания, как и определения искусственного интеллекта, не существует. Известные трактовки этого понятия отражают его различные аспекты, поэтому приведем несколько определений.

Наиболее общее определение трактует знание как всю совокупность данных (информации), необходимую для решения задачи. В этом определении подчеркивается, что данные в привычном понимании также являются знаниями. Однако знания в информационном плане не ограничиваются рамками данных.

(Данные — представление фактов и идей в формализованном виде, пригодном для передачи и обработки в некотором информационном процессе.)

В полном объеме информация, содержащаяся в знаниях, должна включать сведения о: системе понятий предметной области, в которой решаются задачи; системе понятий формальных моделей, на основе которых решаются задачи; соответствии систем понятий, упомянутых выше; методах решения задачи; текущем состоянии предметной области.

Из перечисленных компонентов только последний (текущее состояние предметной области) в явном виде соответствует понятию "**данные**".

В целом обо всей приведенной выше информации иногда говорят, что она составляет проблемную область решаемой задачи.

Свойства и характеристики знаний

Несмотря на сложности формулировки определения знания считается общепризнанным, что знания имеют ряд свойств, позволяющих отличать их от данных: **внутреннюю интерпретируемость; внутреннюю (рекурсивную) структурированность; внешнюю взаимосвязь единиц; шкалирование; погружение в пространство с семантической метрикой; активность.**

Если данные обладают этими свойствами, можно говорить о перерастании данных в знания.

Раскроем подробнее приведенные выше свойства знаний.

Внутренняя интерпретируемость означает наличие в памяти ЭВМ сведений не только о значении, но и о наименовании информационной единицы. Следует отметить, что это свойство присуще некоторым моделям представления данных, например, реляционной.

Внутренняя (рекурсивная) структурированность отражает вложенность одних информационных единиц в другие или в самих себя. Она предусматривает установку отношений принадлежности элементов к классу, родовидовые отношения типа «часть-целое» и т.п. В целом внутренняя структурированность характеризует структуру знания.

Внешняя взаимосвязь единиц определяет, с какой информационной единицей имеет связь данная информационная единица и какова эта связь. С помощью этого свойства устанавливается связь различных отношений, отражающих семантику и прагматику связей понятий, а также отношений, отражающих смысл системы в целом.

Отдельные информационные единицы не могут описывать динамические ситуации, когда некоторые факты, содержащиеся в структуре одной единицы, вступают в ситуативную связь с фактами или явлениями, описанными в структуре другой единицы. Для описания таких связей используются специальные информационные элементы, в которых указываются имена взаимосвязанных информационных единиц и имена существующих отношений.

Шкалирование означает использование шкал, предназначенных для фиксации соотношения различных величин. Прежде всего шкалирование необходимо для фиксации соотношений качественной информации.

Погружение в пространство с семантической метрикой используется для задания меры близости информационных единиц.

Пример. Пусть разработано несколько вариантов построения системы связи. Требуется определить, насколько структура существующей системы связи близка к одному из имеющихся вариантов. Для этого можно использовать метод матриц сходства - различия, в соответствии с которым матрицы заполняются оценками попарного сходства и различия элементов структур реальной системы связи и рассматриваемого варианта. В качестве оценок обычно выступают числа в диапазоне от -1 до +1 при условии, что -1 характеризует полное различие, а +1 - полное сходство.

На основании матриц сходства - различия определяется степень сходства текущей ситуации с заранее заданной (планируемой).

Активность знаний выражается в возможности вызова той или иной процедуры в зависимости от структуры, сложившейся между информационными единицами.

Активность знаний обусловлена тем, что в отличие от обычных программ, в которых процедуры играют роль активаторов данных, в интеллектуальных системах определенная структура данных активизирует выполнение той или иной процедуры. Практически это осуществляется включением в состав информационной единицы элемента, содержащего имя процедуры, или представлением знаний в виде правил, причем правила записываются в следующем виде: "если произошли события А1 и А2 и ... и Ак, то необходимо выполнить процедуру В". Использование правил значительно упрощает объяснение того, как и почему получено то или иное заключение (вывод).

Перечисленные особенности информационных единиц определяют ту грань, за которой данные превращаются в знания, а базы данных перерастают в базы знаний (БЗ). Совокупность средств, обеспечивающих работу со знаниями, образует систему управления базой знаний (СУБЗ). В настоящее время не существует баз знаний, в которых в полной мере были бы реализованы все пять особенностей знаний.

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

Если рассматривать знания с точки зрения решения задач в некоторой предметной области, то их удобно разделить на две большие категории - факты и эвристику.

9. Процедурные и декларативные знания. Классификация знаний. Формальные языки.

Процедурные и декларативные знания

Знания можно разделить на процедурные и декларативные. Декларативные знания - это совокупность сведений о качественных и количественных характеристиках конкретных объектов, явлений и их элементов, представленных в виде фактов и эвристик. Традиционно такие знания накапливались в виде разнообразных таблиц и справочников, а с появлением ЭВМ приобрели форму информационных массивов (файлов) и баз данных. Процедурные знания хранятся в памяти ИИС в виде описаний процедур, с помощью которых их можно получить. В виде процедурных знаний обычно описывается информация о предметной области, характеризующая способы решения задач в этой области, а также различные инструкции, методики и тому подобная информация. Другими словами, процедурные знания - это методы, алгоритмы, программы решения различных задач, последовательности действий (в выбранной проблемной области) - они составляют ядро баз знаний.

Таким образом, при использовании знаний происходит переход к формуле
знания + вывод = система.

Работа со знаниями, иначе называемая **обработкой знаний**, лежит в основе всего современного периода развития ИИ. В свою очередь обработка знаний включает в себя:

извлечение знаний из источников (под источниками понимаются материальные средства хранения знаний, а также события и явления, но при этом считается, что человек источником не является);

приобретение знаний от профессионалов (экспертов); **представление знаний**, т.е. их формализация, позволяющая в дальнейшем использовать знания для проведения логического вывода на ЭВМ;

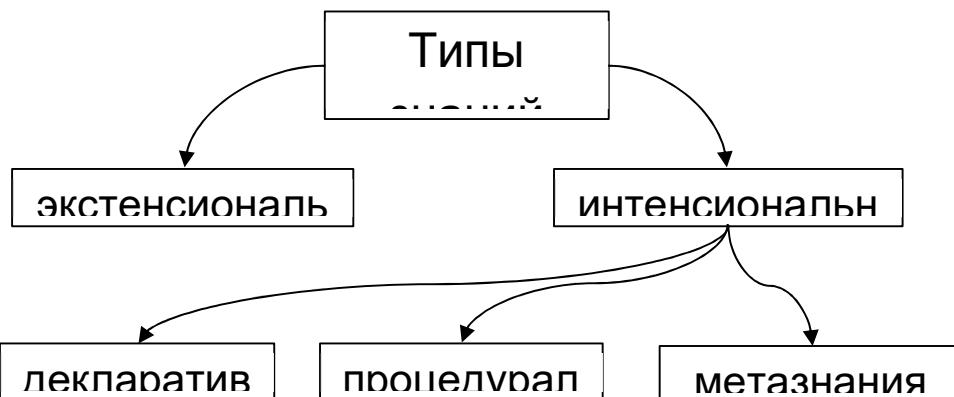
манипулирование знаниями, включающее пополнение, классификацию, обобщение знаний и вывод на знаниях;

объяснение на знаниях, позволяющее дать ответ, как и почему проведен тот или иной вывод.

В памяти ЭВМ знания представляются в виде некоторой знаковой системы. С понятием "знак" связываются понятия "**экстенсионал**" и "**интенсионал**".

Экстенсионал знака - это его конкретное значение или класс допустимых значений.

Интенсионал знака - это его смысл, характеристика содержания. Интенсионал знака определяет содержание связанного с ним понятия.



Соответственно различают два типа знаний: экстенсиональные и интенсиональные.

Экстенсиональные знания - это набор количественных и качественных характеристик различных конкретных объектов. Они представляются перечислениями объектов предметной области, экземпляров объектов, свойств объектов. Иными словами, экстенсиональные знания - это данные, хранящиеся в базах данных.

Иногда экстенсиональные знания называются предметными, или фактографическими знаниями.

Интенсиональные знания - это совокупность основных терминов, применяемых в проблемной области, и правил над ними, позволяющих получать новые знания. Интенсиональные знания описывают абстрактные объекты, события, отношения.

Интенсиональные знания подразделяются на декларативные, процедуральные и метазнания.

Декларативные знания отражают понятия проблемной области и связи между ними. Они не содержат в явном виде описания каких-либо процедур. Иначе декларативные знания называются понятийными, или концептуальными.

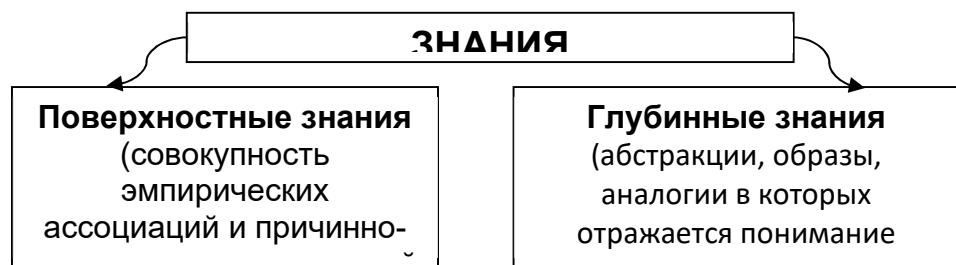
Процедуральные знания описывают процедуры, т.е. указывают операции над понятиями, позволяющие получать новые понятия. В отличие от декларативных знаний они содержат в явном виде описания процедур. Примером процедуральных знаний является программа, хранящаяся в памяти ЭВМ. Иногда процедуральные знания называются алгоритмическими.

Метазнания - это знания об организации всех остальных типов знаний. Иначе они называются специальными. Метазнания содержат признаки декларативных и процедуральных знаний.

Поверхностные — знания о видимых взаимосвязях между отдельными событиями и фактами в предметной области.

Глубинные — абстракции, аналогии, схемы, отображающие структуру и природу процессов, протекающих в предметной области. Эти знания объясняют явления и могут использоваться для прогнозирования поведения объектов.

Классификация знаний (по глубине и жесткости)



Классификация знаний по «глубине»



Классификация знаний по «жесткости»

Формальные языки

Формализованный (формальный) язык - язык, характеризующийся точными правилами построения выражений и их понимания. Он строится в соответствии с четкими правилами, обеспечивая непротиворечивое, точное и компактное отображение свойств и отношений изучаемой предметной области (моделируемых объектов).

В отличие от естественных языков формальным языкам присущи четко сформулированные правила семантической интерпретации и синтаксического преобразования используемых знаков, а также то, что смысл и значение знаков не изменяется в зависимости от каких-либо прагматических обстоятельств (например, от контекста).

Большинство формальных языков (созданных конструкций) строится по следующей схеме: сначала выбирается алфавит, или совокупность исходных символов, из которых будут строиться все выражения языка; затем описывается синтаксис языка, то есть правила построения осмысленных выражений. Буквами в алфавите формального языка могут быть и буквы алфавитов естественных языков, и скобки, и специальные знаки и т.п. Из букв, по определенным правилам можно составлять слова и выражения. Осмысленные выражения получаются в формальном языке, только если соблюдены определенные в языке правила образования. Для каждого формального языка совокупность этих правил должна быть строго определена и модификация любого из них приводит чаще всего к появлению новой разновидности (диалекта) этого языка.

Формальные языки широко применяются в науке и технике. В процессе научного исследования и практической деятельности формальные языки обычно используются в тесной взаимосвязи с естественным языком, поскольку последний обладает гораздо большими выразительными возможностями. В то же время формальный язык является средством более

точного представления знаний, чем естественный язык, а следовательно, средством более точного и объективного обмена информацией между людьми.

Формальные языки часто конструируются на базе языка математики. Веком бурного развития различных формальных языков можно считать XX век.

С точки зрения информатики, среди формальных языков наиболее значительную роль играют формальный язык логики (язык алгебры логики) и языки программирования.

Возникновение языков программирования приходится на начало 50-х годов XX в.

Языков программирования и их диалектов (разновидностей) насчитывается несколько тысяч. Классифицировать их можно по-разному. Некоторые авторы разбивают все многообразие языков программирования на процедурные и декларативные. В процедурных языках преобразование данных задается с помощью описания последовательности действий над ними. В декларативных языках преобразование данных задается посредством описания отношений между самими данными. Согласно другой классификации, языки программирования можно разделить на процедурные, функциональные, логические, объектно-ориентированные. Однако любая классификация несколько условна, поскольку, как правило, большинство языков программирования включает в себя возможности языков разных типов. Особое место среди языков программирования занимают языки, обеспечивающие работу систем управления базами данных (СУБД). Часто в них выделяют две подсистемы: **язык описания данных и язык манипулирования**.

10. Модели представления знаний. Классификация моделей знаний.

Центральной проблемой обработки знаний, является проблема представления знаний.

Проблема представления знаний - это проблема представления взаимосвязей в конкретной предметной области в форме, понятной системе искусственного интеллекта. Представление знаний - это их формализация и структурирование (в целях облегчения решения задачи), с помощью которых отражаются характерные признаки знаний: внутренняя интерпретируемость, структурированность, связность, семантическая метрика и активность. Представление знаний - это процесс (способ) описания знаний человека о проблемной области посредством выражений на формальном языке, называемом языком представления знаний.

Модель представления знаний (МПЗ) - это способ и результат формального описания знаний в БЗ. Она должна быть понятной пользователю и обеспечивать однородность представления знаний, за счет чего упрощаются управление знаниями и логический вывод, а также удовлетворять ряду других требований.

К настоящему времени разработано достаточно много различных МПЗ, и работа по созданию новых моделей продолжается. Однако наибольшее распространение получили четыре модели:

1. модель семантической сети,
2. фреймовая,
3. продукционная
4. и логические.

В основе использования МПЗ лежит аксиоматический метод. Аксиоматический метод в любой науке состоит в том, что выделяется некоторое небольшое множество истинных утверждений, опираясь на которые можно вывести все истинные утверждения данной науки. Классическим примером аксиоматического метода является аксиоматическое построение геометрии Евклида и Лобачевского, каждая из которых базируется на ряде постулатов. Следует отметить, что одну и ту же теорию можно строить, исходя из различного набора аксиом.

Таким образом, необходимо:

1. Построить алфавит теории, т.е. задать счетное множество символов (буквы и знаки включены в этот термин) и определить множество объектов языка - выражений. Под выражением имеется в виду конечная последовательность символов языка. Понятно, что символ может несколько раз появляться в выражении, а может и не появляться (роль формальных выражений в символическом языке аналогична роли слов в обычном языке).
2. Выделить подмножество таких выражений, которые будем называть формулами (обычно имеется хорошо разработанная процедура, позволяющая по данному выражению определить, является ли оно формулой). С формулами не будем связывать никакого значения (смысла); никакого значения не будем связывать и с входящими в них буквами и символами. Все операции будут формализованы.
3. Из бесконечного множества истинных формул (тавтологий) выделим небольшую группу (1 - 10) так называемых аксиом теории (как правило, всегда имеется возможность эффективно выяснить, является ли данная формула аксиомой). За аксиомы берутся некоторые тавтологии, из которых по формальным правилам выводятся все остальные тавтологии.
4. Указать конечное множество отношений между формулами, которые называют правилами вывода. Правила вывода сопоставляют некоторым последовательностям формул новые формулы. Записывают правила вывода в форме фигуры, где формулы, стоящие над чертой называются посылками, а формулы, стоящие под чертой, называется следствием посылок по данному правилу вывода. С помощью правил вывода из аксиом получаются новые истинные формулы, называемые теоремами.

Доказательством (выводом) называется конечная последовательность формул A_1, A_2, \dots, A_p такая, что каждая A_k есть либо аксиома теории, либо непосредственное следствие каких-либо предыдущих формул по одному из правил вывода. Теоремой называется такая формула A теории, что существует вывод, в котором последней формулой является формула A . Теоремы, как правило, выражаются равенствами, импликациями и эквивалентностями. Доказательство теорем превращается в последовательность таких формул, и построение формальных доказательств можно поручить ЭВМ.

К МПЗ предъявляются требования полноты и непротиворечивости.

Классификация моделей знаний:



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

Выявление и сбор нужных знаний - центральный вопрос, который встает перед разработчиками систем с интеллектом. Проблема в том, что эксперт и, вообще говоря, любой человек несет в себе огромное количество «скрытых» знаний, то есть знаний, которые им не осознаются и часто проявляются в сложных ситуациях, когда, возможно, решается вопрос жизни и смерти. Нам необходим весь объем знаний и скрытых, и явных, но разбросанных по разным источникам, который бы обеспечивал решение задач, ранее решаемых только высококлассными специалистами. Сложность проблемы привела к разработке целого арсенала методов, обобщенная классификация которых приведена на рис. 1.



Рис. 1. Структура поля методов инженерии знаний

1.1. Теоретические аспекты получения знаний

Инженеру по знаниям необходимо четко понимать природу и особенности процесса извлечения знаний. Из множества аспектов проблематики получения знаний выделяют три основных: **психологический, лингвистический и методологический (гносеологический)**.

Считается, что **психологический** аспект является ведущим из трех аспектов извлечения знаний, поскольку он определяет успешность и эффективность взаимодействия инженера по знаниям с источником знаний первого типа - **экспертом**. Психологический аспект выделяется еще и потому, что извлечение знаний происходит чаще всего в процессе непосредственного общения разработчиков ЭС, а законы психологии общения, к сожалению, часто недооцениваются или игнорируются начинающими инженерами по знаниям.

Упрощенно выделяют следующие структурные компоненты: **участники общения (партнеры), средства общения (процедура), предмет общения (знания)**.

В соответствии с этой структурой выделяют **три «слоя» психологических проблем**, возникающих при извлечении знаний:

- **контактный** (проблемы взаимодействия партнеров);
- **процедурный** (проблемы грамотного проведения процедуры извлечения);
- **когнитивный** (проблема получения знания как такового).

На взаимодействие партнеров влияют следующие их параметры:

демографические (пол, возраст, национальность, образование);

личностные (особенности личностного портрета, темперамент, мотивация).

На проведение процедуры извлечения влияют профессиональные параметры:

ситуация (место, время, продолжительность);

оборудование (вспомогательные средства, освещенность, мебель);

профессиональные приемы (темпер и громкость речи, мимика, поза, интонации).

Наименее исследованы проблемы когнитивного слоя, связанные с изучением пространства памяти эксперта и реконструкцией его понятийной структуры и модели рассуждений.

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

Гносеологический аспект извлечения знаний объединяет методологические проблемы получения нового научного знания, поскольку при создании БЗ эксперт часто впервые

формулирует некоторые закономерности, до того момента составлявшие его личный опыт.

Внутренняя согласованность эмпирического знания характеризуется понятиями модальности, противоречивости и неполноты. Модальность знания означает возможность его существования в различных категориях. Возможные противоречия не всегда должны разрешаться в поле знаний, а напротив, именно они служат чаще всего отправной точкой в рассуждениях экспертов. Неполнота знания связана с невозможностью полного описания ПрО.

Системность ориентирует инженера по знаниям на рассмотрение любой ПрО с позиций закономерностей системного целого и взаимодействия составляющих его частей.

Процесс познания глубоко субъективен, то есть он существенно зависит от особенностей самого познающего субъекта, поэтому более корректно говорить о глубине понимания, чем об **объективности** знания. Таким образом, инженер по знаниям должен сосредоточиться на понимании проблемы.

Историзм связан с развитием и изменением представлений о ПрО с течением времени.

Методологическая структура познания может быть представлена как некоторая последовательность этапов:

- **описание и обобщение фактов;**
- **выявление связей между фактами**, формулирование правил и закономерностей;
- **построение модели ПрО;**
- **объяснение и прогнозирование явлений.**

1.2. Методы извлечения явных знаний

Рассмотрим обобщенную классификацию этих методов (рис. 2).

Первый класс образуют методы, которые ориентированы на непосредственный контакт инженера по знаниям с экспертом (источником знаний),

второй класс - текстологические методы, основанные на приобретении знаний из документов и специальной литературы.

Коммуникативные методы разделяются на пассивные и активные. В **пассивных** методах ведущую роль играет эксперт, в **активных** - инженер по знаниям. При решении

конкретных задач, как правило, используются как пассивные, так и активные методы. Активные методы, в свою очередь, делятся на **индивидуальные** и **групповые**. В групповых методах знания получают от множества экспертов, в индивидуальных - от единственного. Индивидуальные методы получили более широкое применение на практике.



Рис. 2. Классификация методов извлечения данных

Пассивные коммуникативные методы включают наблюдение, анализ протоколов «мыслей вслух», процедуры извлечения знаний из лекций.

Метод наблюдения является одним из наиболее применяемых на начальных этапах разработки экспертных систем. Его суть заключается в фиксировании всех действий эксперта, его реплик и объяснений. При этом аналитик не вмешивается в работу эксперта, а только наблюдает за процессом решения реальных задач либо за решением проблем, имитирующих реальные задачи. Наблюдения за процессом решения реальных задач позволяют инженеру по знаниям глубже понять предметную область. Однако эксперт в этом случае испытывает большое психологическое напряжение, понимая, что осуществляет не только свою профессиональную деятельность, но и демонстрирует ее инженеру по знаниям. Наблюдение за имитацией процесса снимает это напряжение, но приводит к снижению полноты и качества извлекаемых данных. Наблюдения за имитацией незаменимы в тех случаях, когда наблюдения за реальным процессом невозможны из-за специфики изучаемой предметной области.

Метод анализа протоколов «мыслей вслух» отличается от метода наблюдения тем, что эксперт не только комментирует свои действия, но и объясняет цепочку своих рассуждений, приводящих к решению. Основной проблемой, возникающей при

использовании этого метода, является **принципиальная сложность для любого человека словесного описания собственных мыслей и действий**. Повысить полноту и качество извлекаемых знаний можно за счет многократного уточняющего протоколирования рассуждений эксперта.

Метод извлечения знаний из лекций предполагает, что эксперт передает свой опыт инженеру по знаниям в форме лекций. При этом инженер по знаниям может заранее сформулировать темы лекций. Если этого не удается сделать, то инженер конспектирует лекции и задает вопросы. Качество информации, предоставленной экспертом в ходе лекции, определяется четкостью сформулированной темы, а также способностями лектора в структурировании и изложении своих знаний и рассуждений.

Активные индивидуальные методы включают **методы анкетирования, интервьюирования, свободного диалога и игры с экспертом**.

Преимуществом методов анкетирования является то, что анкета или вопросник составляются инженером по знаниям заранее и используются для опроса экспертов. Составление анкеты следует проводить с учетом рекомендаций, выработанных в социологии и психологии. Основные требования к анкетам такие:

Анкета не должна быть монотонной и однообразной, чтобы не вызывать скуку или усталость. Для этого необходимо разнообразить тематику и формы задания вопросов, включить вопросы-шутки и применить стиль игры.

Анкета должна быть приспособлена к языку эксперта.

Следует учитывать, что вопросы влияют друг на друга, поэтому важно расположить их в правильной последовательности.

В анкете должно содержаться оптимальное число избыточных вопросов, часть которых предназначена для контроля правильности ответов, а другая часть - для снятия напряжения.

Метод интервьюирования отличается от метода анкетирования тем, что позволяет аналитику опускать ряд вопросов в зависимости от ситуации, вставлять новые вопросы в анкету, изменять темы и разнообразить ситуацию общения. Важную роль в методе интервьюирования играют типы вопросов, которые классифицируются по форме, по функциям, по воздействию.

По форме вопросы классифицируются следующим образом:

открытый вопрос называет тему или предмет, оставляя эксперту полную свободу в отношении формы и содержания ответа;

закрытый вопрос предлагает эксперту выбрать ответ из предложенного набора;

личный вопрос касается личного опыта эксперта;

безличный вопрос направлен на выявление наиболее распространенных закономерностей предметной области;

прямой вопрос непосредственно указывает на интересующий предмет или тему (используется при «зажатости» эксперта);

косвенный вопрос исподволь затрагивает рассматриваемую проблему;

верbalный вопрос - традиционный устный вопрос;

вопрос с использованием наглядного материала позволяет разнообразить интервью и снять усталость эксперта (используются фотографии, рисунки, карточки).

По функциям вопросы делятся на основные, зондирующие и контрольные:

основной вопрос направлен на выявление знаний;

зондирующий вопрос направляет рассуждения эксперта в нужную сторону;

контрольный вопрос проверяет достоверность и объективность информации, полученной в интервью ранее.

По воздействию выделяют вопросы нейтральные и наводящие:

нейтральный вопрос подчеркивает беспристрастность инженера по знаниям к предмету исследования;

наводящий вопрос ориентирует эксперта принять во внимание позицию инженера по знаниям.

Дополнительно в интервью рекомендуется включать следующие вопросы: контактные (снимающие психологический барьер между аналитиком и экспертом); буферные (разграничивающие отдельные темы интервью); оживляющие память экспертов (реконструирующие отдельные случаи из практики); «провоцирующие» (способствующие получению неподготовленных ответов).

При использовании метода интервьюирования следует иметь в виду, что его эффективность во многом определяется языком вопросов (понятностью, лаконичностью, терминологией); порядком вопросов (логическая последовательность); уместностью вопросов (этичностью и вежливостью).

Прежде чем готовить вопросы, аналитик должен овладеть ключевым набором знаний исследуемой предметной области, поскольку любой вопрос имеет смысл только в контексте.

Метод свободного диалога позволяет извлекать знания в форме беседы с экспертом, поэтому здесь не предусматривается использование жесткого вопросника или плана. В то же время подготовка к свободному диалогу должна проводиться по специальной методике, в которую входит общая, специальная, конкретная и психологическая подготовка. Общая подготовка направлена на повышение научной эрудиции, овладение общей культурой, знакомство с системной методологией. Специальная подготовка сводится к овладению теорией и навыками интервьюирования. Конкретная подготовка предполагает изучение предметной области, подготовку ситуации общения, знакомство с экспертом, тестирование эксперта. Психологическая подготовка включает знакомство с теорией общения и с когнитивной психологией.

Активные групповые методы включают «мозговой штурм», дискуссии за круглым столом и ролевые игры. Групповые методы позволяют творчески интегрировать знания множества экспертов.

Метод «мозгового штурма» - один из наиболее известных и широко применяемых методов генерирования новых идей путем творческого сотрудничества группы специалистов. Являясь, в некотором смысле, единым мозгом, группа пытается штурмом преодолеть трудности, мешающие разрешить рассматриваемую проблему. В процессе такого штурма участники выдвигают и развиваются собственные идеи, стимулируя появление новых и комбинируя их. Для обеспечения максимального эффекта «мозговой штурм» должен подчиняться определенным правилам и основываться на строгом разделении во времени процесса выдвижения идей и процесса их обсуждения и оценки. На первой стадии штурма запрещается осуждать выдвинутые идеи и предложения (считается, что критические замечания уводят к частностям, прерывают творческий процесс, мешают выдвижению идей). Роль аналитика состоит в том, чтобы активизировать творческое мышление участников заседания и обеспечить выдвижение возможно большего числа идей.

После выдвижения идей выполняются тщательное их обсуждение, оценка и отбор лучших. На стадии обсуждения участники «мозгового штурма» должны сконцентрироваться на положительных сторонах идей, найти в них рациональные зерна и предложить направления их развития. Выдвигаемые в процессе обсуждения дополнительные идеи могут базироваться на идеях других участников или, наоборот, служить для них фундаментом, катализатором. Значительный эффект дает комбинирование идей при одновременном выявлении преимуществ и недостатков синтезируемых при этом вариантов.

Метод «мозгового штурма» эффективен при решении не слишком сложных задач общего организационного характера, когда проблема хорошо знакома всем участникам заседания и по рассматриваемому вопросу имеется достаточная информация.

Индивидуальный «мозговой штурм» проводится по тем же правилам, что и коллективный, но выполняется одним экспертом, который одновременно генерирует идеи, дает им объективную оценку и критикует их.

Массовый «мозговой штурм» проводится в массовой аудитории (до нескольких десятков человек). Отбор идей проводится на промежуточных этапах. Эксперты группируются по 6-8 человек, при этом важно, чтобы непосредственное отношение к задаче имел лишь руководитель группы, а остальные были лишь знакомы с нею (иначе амбиции могут сыграть негативную роль). Штурм проводится в два этапа. На первом этапе оперативные группы осуществляют прямой коллективный «мозговой штурм».

При этом желательно, чтобы каждая группа работала над задачей, наиболее близкой по тематике к профилю вошедших в нее специалистов. На втором этапе руководители каждой группы в течение нескольких минут оценивают выдвинутые идеи, отбирают из них наиболее интересные и сообщают их на «пленарном заседании».

Двойной «мозговой штурм» органически соединяет в себе процессы генерирования идей и их доброжелательной позитивной критики.

Обратный «мозговой штурм» отличается от прямого тем, что в нем больше внимания уделяется критике высказанных идей.

Метод дискуссии за круглым столом предполагает равноправное обсуждение экспертами поставленной проблемы. Отличительной особенностью метода дискуссии

является коллективное рассмотрение предметной области с разных точек зрения и исследование спорных гипотез.

Экспертные игры предназначены для извлечения знаний и базируются на деловых, диагностических и компьютерных играх (рис. 3).



Рис. 3. Классификация экспертных игр

По числу участников игры подразделяют на **индивидуальные** (игры с экспертом) и **групповые** (ролевые игры в группе). По применению специального оборудования - игры с тренажерами и игры без реквизита. Особый класс представляют собой компьютерные игры.

В играх с экспертом инженер по знаниям берет на себя чью-нибудь роль в моделируемой ситуации. Ролевые игры в группе предусматривают участие в игре нескольких специалистов. Участники игры наделяются определенными ролями, а собственно игра проводится по составленному когнитологом сценарию. В целях повышения эффективности ролевых игр в них необходимо вводить элементы состязательности.

Игры с применением тренажеров позволяют фиксировать трудноуловимые знания, которые возникают в реальных ситуациях и могут быть потеряны при выходе из них.

Компьютерные экспертные игры в настоящее время используются в основном в целях обучения. Они полезны для «разминки» экспертов перед сеансом извлечения знаний.

Текстологические методы включают методы извлечения знаний, основанные на изучении текстов учебников, специальной литературы и документов.

Текстология - это наука, целью которой является практическое прочтение текстов, изучение и интерпретация литературных источников, а также рассмотрение семиотических, психолингвистических и других аспектов извлечения знаний из текстов.

Особую сложность представляет извлечение знаний из специальной литературы и методик, поскольку в них очень высока степень концентрации специальных знаний.

1.3. Методы выявления скрытых знаний

Для выявления скрытых знаний обычно рекомендуются методы **психосемантики**, которые позволяют исследовать структуры сознания через моделирование индивидуальной системы знаний человека и выявлять элементы знаний, которые могут им не осознаваться (латентные, скрытые, имплицитные). Эта наука объединяет методы когнитивной психологии, психолингвистики, психологии восприятия и исследования индивидуального сознания.

Методы многомерного шкалирования основаны на статистических методах обработки экспертных оценок сходства между анализируемыми объектами, которые выбираются из определенной шкалы. Результаты обработки представляются в виде точек некоторого координатного пространства. Возможность визуализации результатов является безусловным преимуществом метода, однако она быстро утрачивается с увеличением размерности пространства.

Шкалированием называют поиск подпространства, для которого величина $D^m - D^{m*}$ имеет минимальное значение. Здесь D^m , D^{m*} - матрицы расстояний между объектами (признаками) в исходном пространстве D^m и в искомом подпространстве D^{m*} , где m и m^* - размерности соответствующих пространств. Если $m^* = 2$, шкалирование превращается в проецирование на плоскость. Значения расстояний в матрицах D могут выбираться экспертом из предложенной шкалы либо вычисляться по совокупности признаков, описывающих объект. Во втором случае расстояния можно вычислить разными способами. Одной из самых популярных метрик является евклидово расстояние:

$$d_{i,j} = \left(\sum_{k=1}^K \left(x_i^k - x_j^k \right)^2 \right)^{\frac{1}{2}},$$

где x_i^k , x_j^k – значения k -го признака у i -го и j -го объектов соответственно; K – общее число признаков.

Расстояния-метрики должны удовлетворять следующим условиям:

$$d(x, y) \geq 0; d(x, x) = 0; d(x, y) = d(y, x); d(x, y) + d(y, z) \geq d(x, z).$$

Метрическим шкалированием называют образование новых классов с использованием метрических расстояний. Этот тип обработки данных ориентирован на максимальное сближение числовых значений матриц D^m и D^{m*} . Существует также **неметрическое шкалирование**, которое не предъявляет жестких требований к сближению пространств и во многих случаях более оправдано в связи с условностью понятия «расстояние».

Важно отметить, что в шкалировании отыскиваются не новые признаки, а новые пространства, поэтому его результаты следует интерпретировать как восстановленную (на плоскости или в объеме) структуру расположения точек.

Главными недостатками метода многомерного шкалирования являются:

- субъективные оценки сходства между объектами и признаками обрабатываются как расстояния в пространстве, а результаты анализируются на основе геометрической интерпретации. Это ограничивает размерность выявляемых пространств и требует серьезного упрощения реальных знаний эксперта, следствием которого могут стать неадекватные БЗ;
- выделенные подпространства не имеют иерархической организации, что затрудняет их интерпретацию;
- используется только один вид отношений между понятиями (отношение сходства).

Метафорический подход ориентирован на выявление скрытых составляющих практического опыта эксперта и основан на сравнении объектов предметной области с абстрактными объектами из мира метафор, в результате чего можно выявить новые свойства анализируемых объектов и определить отношение эксперта к ним. Используя метафорические сравнения, эксперт выходит за рамки объективности и действует в соответствии со своими субъективными представлениями.

Метод репертуарных решеток широко применяется в психологических исследованиях для выявления личностных свойств, которые проявляются через систему личностных конструктов. Этот метод может применяться и для извлечения знаний.

Репертуарная решетка представляет собой матрицу, которая заполняется экспертом. Столбцам матрицы соответствуют определенные группы объектов (элементов), в качестве которых могут выступать люди, предметы, понятия, отношения, звуки и др. Строки матрицы соответствуют конструктам, которые представляют собой биполярные признаки, параметры, шкалы, отношения или способы поведения. **Дж. Келли, автор этого метода**, называл конструктом признак или свойство, определяющие сходство двух или нескольких объектов и их отличие от других объектов. Иными словами, конструкты - это признаки, которые могут использоваться для обобщения и разделения объектов на классы. Конструкты можно применить не к любым объектам, а только в некотором «диапазоне их пригодности».

Примерами личностных конструктов могут быть «умный - глупый», «мужской - женский», «хороший - плохой».

Самым распространенным и простым методом анализа репертуарной решетки является кластерный анализ. Слабым местом в теории Дж. Келли является предположение о том, что человек может точно описать конструкты, которые он использует, чтобы объяснить, чем сравниваемые объекты похожи друг на друга и чем отличаются. Процедура выявления и вербализации конструктов очень утомительна для экспертов, поэтому во многих методиках используются готовые наборы конструктов, релевантные рассматриваемым объектам.

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

Проблемы структурирования знаний

Объем и содержание нужных знаний должны определяться принятыми разработчиками понятием модели предметной области (МПрО) и методикой ее построения. А понятия эти у разных авторов - разные, зависят от целей, которые ставят перед собой авторы, принятых подходов, субъективных предпочтений, ориентации на определенные классы задач, методы формализации и т.д. Сходятся они в одном: нужно выделить объекты предметной области (ПрО), их свойства и отношения между ними. Сегодня чаще говорят о необходимости построения онтологии ПрО как неформальной системы концептуализации знаний, как метода их структурирования. Выделяются три основных подхода к структурированию знаний: структурный, объектный и объектно-структурный.

Структурный подход основан на идее алгоритмической декомпозиции, когда каждый модуль системы выполняет один из этапов общего процесса. В рамках этого подхода разработано большое число выразительных средств: диаграммы потоков данных, структурированные словари (тезаурусы), языки спецификации системы, таблицы решений, стрелочные диаграммы, деревья переходов, деревья целей и т.д.

Объектно-ориентированный подход связан с декомпозицией, при которой каждый объект рассматривается как экземпляр определенного класса. К базовым понятиям подхода относятся такие как: абстрагирование, класс, иерархия, наследование, типизация, инкапсуляция, модульность, полиморфизм.

- Абстрагирование - это упрощенное описание системы, в которой выделяются ее наиболее существенные свойства и детали, а незначительные аспекты опускаются. Реальность представляется моделью сущности (объекта) и моделью (методом) ее поведения. При этом объекты соответствуют понятиям ПрО, а методы - операциям, которые должны выполняться над объектами.
- Класс - множество объектов, у которых структура и свойства одинаковы.
- Иерархия - это упорядоченная система абстракций - классов.
- Наследование - соотношение между классами, когда один класс использует структурную или функциональную часть другого класса (или нескольких других).
- Типизация - ограничение, накладываемое на класс, которое препятствует взаимозаменяемости объектов, принадлежащих, разным классам.
- Инкапсуляция - ограничение доступа к внутренней структуре и механизмам функционирования объекта.
- Модульность - свойство системы, допускающее возможность ее декомпозиции на ряд взаимосвязанных частей (модулей).
- Полиморфизм - возможность наделения объекта различными свойствами и стратегиями поведениями. Другими словами, предполагается, что одно имя может соответствовать различным классам объектов, входящим в один суперкласс. То есть объект, обозначенный этим именем, может по-разному реагировать на некоторые действия.
- Объектно-структурный подход предполагает проведение последовательного анализа информации о рассматриваемой ПрО и ее представления в виде стратифицированной модели, в которой все знания раскладываются по ролевым стратам: кто, что, зачем, как, где, когда, почему, сколько и т.п. (табл. 1).

Таблица 1.

Уровень страты	Категория страты	Вид анализа
1	КТО	Организационный анализ: коллектив разработчиков
2	ЧТО	Концептуальный анализ: основные принципы, понятийная структура
3	ЗАЧЕМ	Стратегический анализ: назначение и функции системы
4	КАК	Функциональный анализ: гипотезы и модели принимаемых решений
5	ГДЕ	Пространственный анализ: окружение, оборудование, коммуникации
6	КОГДА	Временной анализ: временные параметры и ограничения

7	ПОЧЕМУ	Причинно-следственный (каузальный) анализ
8	СКОЛЬКО	Экономический анализ: ресурсы, прибыль, окупаемость

Проблема сборки знаний в единую модель предметной области

Из приведенного обзора поля методов инженерии знаний становится ясно, что рядовой инженер по знаниям (или когнитолог), и не только он, должен все перечисленное реально знать и понимать настолько, чтобы свободно ориентироваться, когда, что и как применять при разработке БЗ ЭС реальной сложности.

Очевидно, что в работе с экспертом (экспертами) должны принимать участие психологи, методологи, игротехники-математики и другие люди, умеющие четко взаимодействовать между собой, что само по себе не просто. Но если они даже справляются со своими задачами, возникают вопросы согласования полученных знаний, их представления, визуализации, структурирования.

В итоге формируется весьма запутанная общая картина. Необходим единый метод, с единой концепцией и структурой, который сквозным образом приведет к нужному результату, который в начале никто себе и не представляет и для которого все перечисленные методы будут играть лишь ту роль, для которой они более всего подходят. То есть будут использоваться тогда, когда это действительно нужно и понятно зачем. Очевидно, это должны быть специальный метод, методика, соответствующие программные средства, которые будут увязывать все в единое целое: процессы работы с экспертом при выполнении всех типов анализов, фиксацию получаемых знаний, их обработку, выдачу на выходе результатов в виде готовой к использованию модели предметной области (БЗ).

13. Логическая модель представления знаний

Логическая модель представления знаний — модели, основанные на правилах формальной логики.

Основная идея подхода при построении логических моделей представления знаний — вся информация, необходимая для решения прикладных задач, рассматривается как совокупность фактов и утверждений, которые представляются как формулы в некоторой логике. Знания отображаются совокупностью таких формул, а получение новых знаний сводится к реализации процедур логического вывода. В основе логических моделей представления знаний лежит понятие формальной теории, задаваемое кортежем:

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

B - счетное множество базовых символов (алфавит);

F - множество (перечислимое) формул (называемых также правильно построенными

формулами), построенных из элементов B с использованием некоторого набора синтаксических

правил;

A - выделенное подмножество априори истинных формул (аксиом);

R - конечное множество отношений между формулами, называемое правилами вывода.

Достоинства логических моделей представления знаний:

1. В качестве «фундамента» здесь используется классический аппарат математической логики, методы которой достаточно хорошо изучены и формально обоснованы.
2. Существуют достаточно эффективные процедуры вывода, в том числе реализованные в языке логического программирования Пролог, использующие механизмы автоматического доказательства теорем для поиска и логически осмысленного вывода информации.
3. В базах знаний можно хранить лишь множество аксиом, а все остальные знания получать из них по правилам вывода.

14. Исчисление высказываний.

Вызованием называется некоторое предложение, смысл которого можно выразить значениями ИСТИНА (TRUE) или ЛОЖЬ (FALSE).

Например, предложения «собака серая» и «собака черная» будут высказываниями. Из простых высказываний можно составить более сложные:

«собака серая или собака черная», «собака серая и собака черная», «если собака не серая, то собака черная».

При этом элементарными высказываниями считаем те, которые нельзя разделить на части. Элементарные высказывания рассматриваются как переменные логического типа.

Исчисление высказываний

Исчисление высказываний – это аксиоматическая логическая система, интерпретацией которой является алгебра высказываний.

Описание всякого исчисления включает в себя описание символов этого исчисления (алфавита); формул, являющихся конечными конфигурациями символов и определение выводимых формул.

Алфавит исчисления высказываний состоит из символов трех категорий:

- 1) Символы первой категории: $x, y, z, \dots, x_1, x_2, \dots$, которые называются переменными высказывания;
- 2) Символы второй категории: $\vee, \wedge, \rightarrow, \neg$, которые называются логическими связками. \vee – дизъюнкция (логическое сложение), \wedge – конъюнкция (логическое умножение), \rightarrow – импликация (логическое следование), \neg – отрицание;
- 3) Символы третьей категории: скобки.

Других символов исчисление высказываний не имеет.

Формулы исчисления высказываний представляют собой последовательности символов алфавита исчисления высказываний. Для обозначения формул будем пользоваться заглавными буквами латинского алфавита. Эти буквы не являются символами исчисления. Они представляют собой условные обозначения формул.

Определение формулы исчисления высказываний.

1. Всякая переменная x, y, z, \dots является формулой.
2. Если A и B – формулы, то слова $(A \wedge B), (A \vee B), (A \rightarrow B), \neg A$ – формулы.
3. Никакая другая строчка символов не является формулой.

Переменные высказывания называются *элементарными формулами*.

Определение подформулы.

1. Подформулой элементарной формулы является только она сама.
2. Если формула имеет вид $\neg A$, то ее подформулами являются: она сама, формула A и все подформулы формулы A .
3. Если формула имеет вид $(A * B)$ (под символом $*$ понимается любая из трех связок $\vee, \wedge, \rightarrow$), то ее подформулами являются: она сама, формулы A и B , все подформулы формул A и B .

Замечание. Скобки в записи формул можно опускать по тем же правилам, что и в алгебре высказываний.

Система аксиом исчисления высказываний:

I группа. $I_1: x \rightarrow (y \rightarrow x); I_2: (x \rightarrow (y \rightarrow z)) \rightarrow ((x \rightarrow y) \rightarrow (x \rightarrow z)).$

15. Основы исчисления предикатов.

Понятие предиката

Если высказывание отражает какой-либо факт и далее оперирует с ним как с единой формулой, не разделяя его, скажем, на субъекты и объекты, то предикатная форма, напротив, отражает данный факт уже как взаимодействие, отношение или свойство некоторых сущностей. Это отношение принято выделять прописными буквами перед скобками, в которых указываются те или иные сущности, находящиеся в данном отношении.

Рассмотрим несколько предложений:

- а) Лена и Таня сестры;
- б) грибы в лесу;
- с) капля долбит камень;
- д) снег белый;
- е) мальчик послал книгу брату.

В правилах исчисления предикатов эти предложения можно записать следующим образом:

- а') СЕСТРЫ (Лена, Таня);
- б') В (грибы, лес);
- в') ДОЛБИТ (капля, камень);
- г') БЕЛЫЙ (снег);
- д') ПОСЛАЛ (мальчик, книга, брат).

В первом предложении выделено отношение родства, во втором - предлогом В - пространственные отношения. В предложении в') выделено действие между субъектом и объектом, в предложении г') - свойство (в данном случае - цвет), в предложении д') - также действие.

То, что стоит перед скобками, называется **предикатным символом** (предикатной константой), то, что стоит в скобках, - термами. Каждый терм занимает свое место. **Предикатные символы могут быть предлогами, существительными, глаголами, прилагательными** и т.п. Терм, как правило, существительное или то, что его заменяет. Все это вместе образует предикатную формулу (или, короче, - предикат).

Термов может быть несколько. По их количеству предикаты разделяются на одноместные (г'), двуместные (а', б', в'), трёхместные (д') и т.д. Предикатная формула еще называется атомом.

Алфавит ИП

ИП - аксиоматическая система, построенная согласно формальной теории $F = (A, V, W, R)$.

Алфавит ИП - А содержит:

индивидуальные константы a, b, c, \dots, k ;

предметные переменные x, y, z, \dots, u ;
функциональные константы f, g, h, \dots, w ;
высказывания p, q, r, s, \dots, l ;
предикатные константы P, Q, R, \dots, V .

Исчисление предикатов в определенном смысле продолжение и расширение исчисления высказываний, поэтому в алфавит включены все те же пропозициональные связки $\wedge, \vee, \rightarrow, \equiv$. Но перечень логических знаков в ИП расширяется еще двумя, называемыми кванторами: \forall и \exists . Квантор \forall , читается, как «все», «для всех», «всякий», «каков бы ни был» и т.н. Поэтому он называется квантором всеобщности (общности). Квантор всеобщности является сверткой бесконечной конъюнкции: $\forall x P(x) = P(x_1) \wedge P(x_2) \wedge \dots \wedge P(x_k)$. Квантор существования - $\exists x P(x)$ - читается, как «некоторый», «хотя бы один», «существует», и является сверткой бесконечной дизъюнкции: $\exists x P(x) = P(x_1) \vee P(x_2) \vee \dots \vee P(x_k)$. Выражение $\forall x P(x)$ читается: «для любого x выполняется условие $P(x)$ ». Выражение $\exists x P(x)$ - «существует хотя бы один x , при котором выполняется $P(x)$ (то есть $P(x) = I$)».

Синтаксис ИП

Множество синтаксических правил V ИВ применимо и в ИП. Правильно построенные формулы (ППФ) в рамках исчисления высказываний остаются ППФ и в исчислении предикатов. Но формулы ИВ, для того, чтобы участвовать во всех преобразованиях формул ИП должны быть представлены в предикатной форме. То есть в каждом высказывании должен быть выделен предикат и его аргументы.

Добавляются еще правила:

1. атом есть ППФ;
2. если $P(x)$ - ППФ и x – переменная, то $\forall x P(x)$ и $\exists x P(x)$ - ППФ;
3. если $P(x, y)$ и x, y - переменные и f - функция, то $\forall x \forall y P(x, f(y))$ и $\exists x \exists y P(x, f(y))$ - ППФ.

Последние формулы верны и для смешанных кванторов.

Каждому квантору соответствует только одна переменная, в наших примерах x или y . Эта переменная называется связанной или квантифицированной, она пишется сразу за квантором. Область действия квантора - формула, к которой применяется эта квантификация. Каждое вхождение квантифицированной переменной в область действия квантификации является **связанным**, любая другая переменная в данной области, не являющаяся связанной, называется свободной.

Каждую предикатную формулу можно интерпретировать, то есть оценить ее как И или Л. При этом можно оценить «перекрытие» кванторов на одну и ту же переменную:

$\forall x \exists x P(x)$ интерпретируется как $\exists x P(x)$, а $\exists x \forall x P(x)$ интерпретируется как $\forall x P(x)$.

Будем понимать под A предикат $A(x, y)$ и отметим важные соотношения:

$$\forall x \forall y A = \forall y \forall x A \quad (6.1)$$

$\exists x \exists y A = \exists y \exists x A$, то есть одноименные кванторы можно менять местами. Иное дело разноименные кванторы. Здесь выполняется только такое условие:

$$\exists x \forall y A \rightarrow \forall y \exists x A$$

Последняя импликация поясняется следующим примером. Пусть имеем для целых чисел истинное утверждение: $\forall y \exists x (x + y = 0)$ (для любого y найдется такой x , что выполняется равенство $x + y = 0$). Переставим кванторы: $\exists x \forall y (x + y = 0)$. Получим выражение: существует такой x , при котором выполняется условие $(x + y = 0)$ для всех y , что некорректно.

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

Из википедии:

Семантическая сеть - один из способов представления знаний. Обычно представлена в виде ориентированного графа, вершины которого соответствуют объектам предметной области, а дуги (ребра) задают отношения между ними.

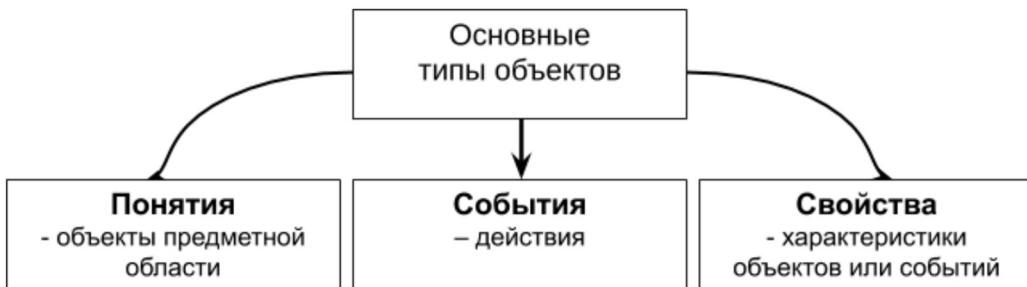
Помимо графа можно представить семантическую сеть множеством вершин V и множеством отношений между ними E (внизу есть примеры)

Примечание: Первое ДЗ по МППР - это атрибутивная семантическая сеть и семантическая сеть предикатного типа. я осуждаю

Из основной лекции:

Термин "семантическая" означает "смысловая", а сама семантика - наука, устанавливающая отношения между символами и объектами, которые они обозначают, т. е. наука, определяющая смысл знаков

Основная идея подхода к представлению знаний, основанных на аппарате семантических сетей, состоит в том, чтобы рассматривать проблемную среду как совокупность сущностей (объектов) и (отношений) связей между ними. Система знаний представляется семантической сетью. Сущности представляются при этом поименованными вершинами, а отношения - направленными поименованными ребрами. Семантическая сеть наиболее близка к тому, как представляются знания в текстах на естественном языке. В ее основе лежит идея о том, что вся необходимая информация может быть описана как совокупность троек ($a r b$), где a и b - два объекта или понятия, а r двоичное отношение между ними. Графически семантическая сеть представляется в виде помеченного ориентированного графа, в котором вершинам соответствуют объекты (понятия), а дугам - их отношения. Дуги помечаются именами соответствующих отношений. Понятия, события и свойства Используются три основных типа объектов: понятия, события и свойства.



Виды семантических сетей

Семантическая сеть является моделью широкого назначения. Выделяют следующие виды семантических сетей:

1. **ситуационные** сети, которые описывают временные, пространственные и причинно-следственные (каузальные - зависимость между переменными или понятиями, при которой изменение в одной (или более) переменной или понятии влечет за собой изменение в другой (других)) отношения;
2. **целевые** сети, используемые в системах планирования и синтеза, которые описывают отношения "цель-средства" и "цель-подцель";
3. **классификационные** сети, использующие отношения "род-вид", "класс-подкласс";
4. **функциональные** сети, использующие отношения "аргумент-функция" и т.д.

Особенность семантической сети как модели представления знаний, которая может одновременно считаться и ее достоинством, и ее недостатком, заключается в невозможности в явном виде разделить БЗ и механизм логического вывода. Поэтому интерпретация семантической сети осуществляется только с помощью использующих ее процедур.

Из дополнительной лекции:

В зависимости от того, какие типы отношений используются, можно выделить несколько видов СС.

1. Однородные СС. В них используется один тип отношений (неважно какой, но один). В этом смысле любые ориентированные графы являются примерами однородных СС.
2. Иерархические СС. Здесь используются отношения: часть - целое (Part-of-while), класс - подкласс или род - вид (АКО - A Kind Of), экземпляр класса - класс (Is A). Каждое из этих отношений образует свою иерархию. Яркими примерами иерархических СС являются организационные структуры, выстраиваемые по отношению «целое - часть». (Например, схема центральной избирательной комиссии: областная, городская, окружная, районная, участковая.)
3. Сценарные СС. Используются отношения строгого или нестрогого порядка. Сюда же могут быть отнесены СС с отношениями времени, причина - следствие.
4. Функциональные СС. В них используются функциональные отношения типа «вход - выход» (рис. 1).

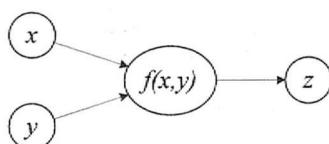


Рис. 1. СС типа «вход - выход»

На рис. 2 приведен пример СС с простой структурой

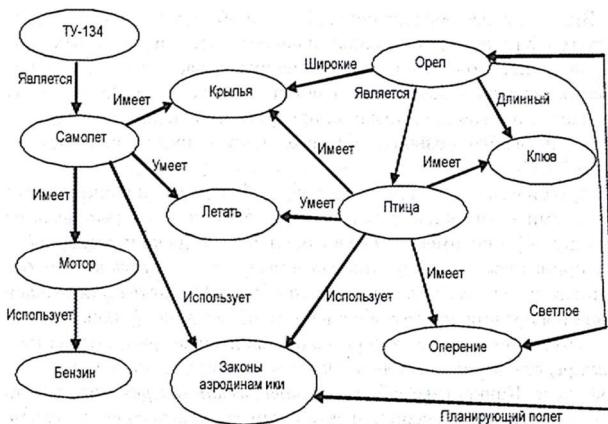


Рис. 2. СС самолета и птицы

(далее - для понимания, и на случай если доп. вопросом попросят написать в математическом виде отношения в подграфе, т.к. это спрашивали при сдаче дз)

Пример предикативной семантической сети:

Рассмотрим текст, состоящий из следующих предложений. Ваня взял книгу у Маши.

Лена взяла книгу у Пети. Петя любит Машу, а Маша любит конфеты.

Предикатная форма: (Взял (Ваня, Книга, Маша); Взял (Лена, Книга, Петя); Любит (Петя, Маша); Любит (Маша, Конфеты)). Семантическая сеть, соответствующая этим предложениям, представлена на рис. 4.

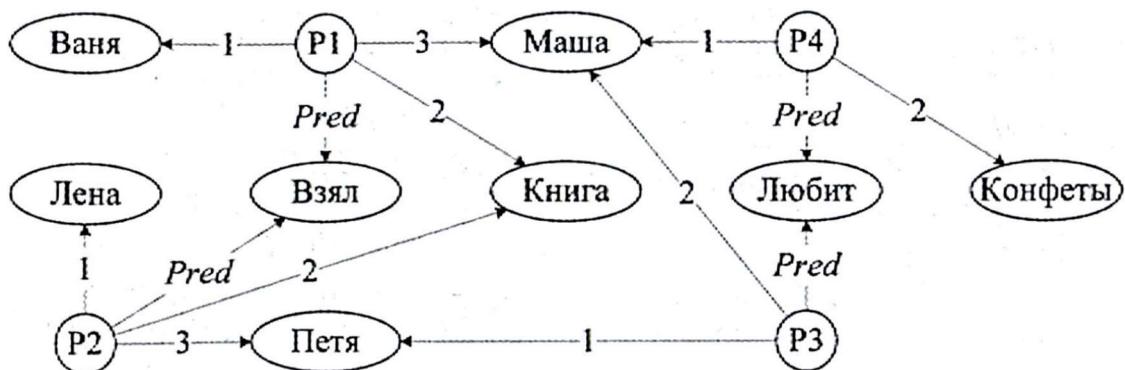


Рис. 4. Предикатная семантическая сеть про книги, любовь и конфеты

Из примеров видно, что:

- 1) частные факты и общие высказывания представляются в виде вершин графа;
- 2) предложение состоит из подграфов (подсетей) специального вида со связями между вершинами-понятиями и предикатными вершинами.

Подграф называется фрагментом сети или предложением. Для объединения составных предложений вводятся специальные вершины «И», «ИЛИ», «НЕ» (\wedge , \vee , \neg). Их предикатная форма: $P1(\Lambda(A, B))$; $P1(V(D, E, F))$; $P1(\neg G)$ рис. 4

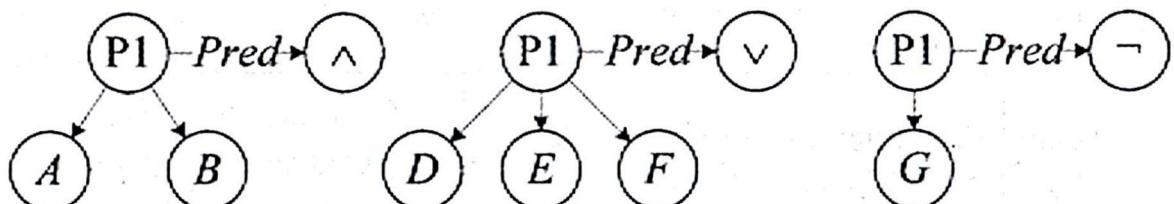


Рис. 4. Примеры «ячеек» предикатной сети, моделирующих пропозициональные связи

Ещё пример:

Говорят, что: ИЛИ Петя любит Машу (P1), ИЛИ Паша любит Машу (P2), ИЛИ Ваня любит Машу (P3), И Ваня выше Пети (P4). Рисунок 5 фиксирует эту ситуацию. Здесь введены дополнительные вершины: P5($\vee(P1, P2, P3)$) и P6($P5 \vee P4$) для предложений типа «ИЛИ» и «Я».

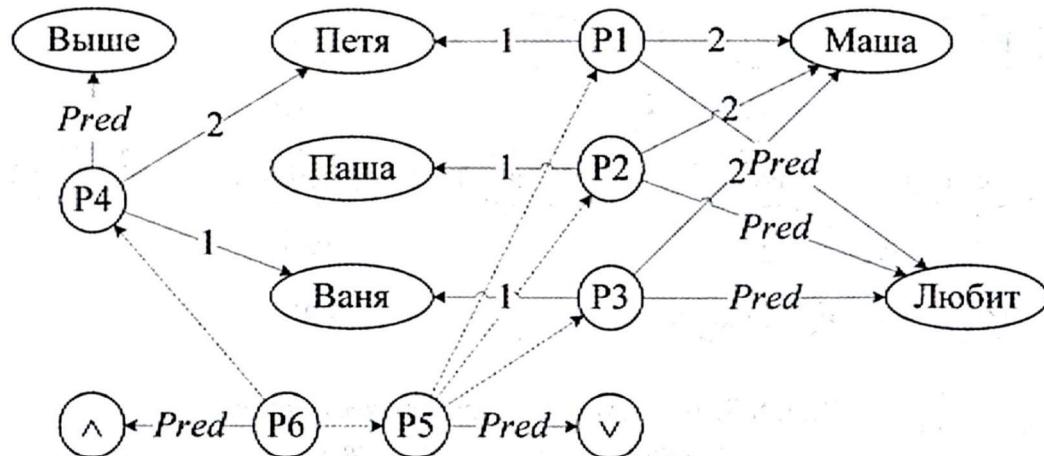


Рис. 5. Предикатная сеть с вершинами типа «И» и «ИЛИ»

(*можно ещё написать/сказать это:*)

Достоинства семантических сетей:

1. Простота и прозрачность описания. Однако с увеличением размеров сети существенно увеличивается время поиска, теряется наглядность.
2. Открытость, позволяющая дополнять и модифицировать СС при необходимости.
3. По сравнению с логикой предикатов СС имеют важное преимущество: вся точно известная информация расположена в базе знаний вокруг соответствующей вершины, т.е. фокусируется в одном месте.

Недостатки семантических сетей:

1. Неоднозначность описания.
2. Отсутствие формального аппарата установления противоречивости описания.
3. Сложность внесения изменений.

Основное применение семантические сети находят в системах обработки естественных языков, а также в системах распознавания образов, в которых СС используются для хранения знаний о структуре, форме и свойствах физических объектов. Могут создаваться СС и более сложной структуры.

17. Семантические сети Куиллиана. Интерпретация семантической сети.

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

(Куиллиана) систематизируют функции отношений между понятиями с помощью следующих признаков:

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

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

Пример семантической сети приведен на рис. 6.1.

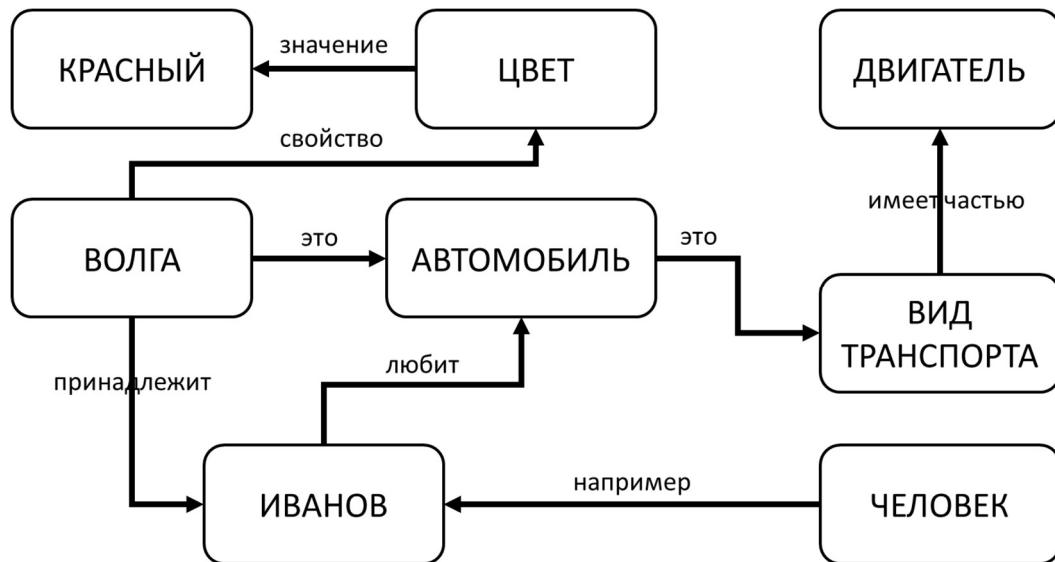


Рис. 6.1. Семантическая сеть

Для реализации семантических сетей существуют специальные языки: **.NET**, язык **SIMER+MIR** и др.

PROSPECTOR, TORUS — известные экспертные системы, использующие семантические сети в качестве модели представления знаний.

Основной способ интерпретации семантической сети.

Основным способом интерпретации семантической сети является **способ сопоставления частей сетевой структуры**. Он основан на построении подсети (подграфа), соответствующей задаваемому вопросу, и сопоставлении ее с общей сетью, имеющейся в БЗ. Запросная подсеть накладывается на имеющийся в базе знаний фрагмент.

Для поиска отношений между концептуальными объектами используется другой способ перекрестного поиска. Согласно этому способу ответ на вопрос выводится путем обнаружения в имеющейся сети узла, в котором пересекаются дуги, исходящие из различных узлов запросной подсети.

Систематизация отношений семантической сети является сложной задачей, зависящей от специфики знаний предметной области. Общезначимые отношения, присутствующие во многих предметных областях, являются основой концепции семантических сетей. Семантические сети получили широкое распространение в системах распознавания речи и экспертных системах. Необходимость структуризации семантических сетей привела к появлению концепции фреймов.

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

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

Слоты

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

Слот имеет уникальное в пределах фрейма имя; в качестве имени слота может выступать любой произвольный текст. Имена системных слотов обычно зарезервированы, в различных системах они могут иметь различные значения. Системные слоты могут иметь имена: AKO, RELATIONS и т. д. Системные слоты служат для редактирования базы знаний и управления выводом во фреймовой системе.

Слот может содержать не только конкретное значение, но также имя процедуры, позволяющей вычислить это значение по заданному алгоритму. Такие процедуры называются присоединенными или связанными процедурами. Вызов связанной процедуры осуществляется при обращении к слоту, в котором она помещена. Например, слот «возраст» может одержать имя процедуры, которая вычисляет возраст по дате рождения, записанной в другом слоте, и текущей дате.

В слоте могут содержаться данные сложных типов: массивы, списки, множества, фреймы и т. д. Значение слота может представлять собой арифметическое значение, фрагмент текста и т. д.

Формально структура фрейма может быть представлена как список свойств:

(<имя_фрейма>
<имя_слота1> (<значение_слота1>) <тип_значения_слота1>
<имя_слота2> (<значение_слота2>) <тип_значения_слота2>)

<имя_слота N> (<значение_слота N>) <тип_значения_слота N>)

Структура фрейма

Фрейм имеет структуру, представленную в табл. 6.1.

Таблица 6.1

Структура фрейма

Имя фрейма		
Имя слота	Значение слота	Значение типа данных слота

Совокупность данных предметной области может быть представлена множеством взаимосвязанных фреймов, образующих единую фреймовую систему, в которой объединяются декларативные и процедурные знания, такая система имеет, как правило, иерархическую структуру, в которой фреймы соединены друг с другом с помощью родовидовых связей. На верхнем уровне находится фрейм, содержащий наиболее общую информацию, истинную для всех фреймов.

Указатели наследования и типа данных. Значение слота.

Важнейшим свойством теории фреймов является заимствование из теории семантических сетей - так называемое наследование свойств. И во фреймах, и в семантических сетях наследование происходит по АКО -связям. Слот связи указывает на фрейм более высокого уровня иерархии, откуда неявно наследуются, т.е. переносятся, значения аналогичных слотов.

Указатели наследования показывают, какую информацию об атрибутах слотов из фрейма верхнего уровня наследуют слоты с одинаковыми именами в данном фрейме. Указатели наследования характерны для фреймовых систем иерархического типа. Наследование происходит по АКО-связям (A-Kind-Of). Слот АКО указывает на фрейм более высокого уровня иерархии, откуда наследуются значения одинаковых слотов.

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 должно находиться в пределах интервала значений, указанных в одноименном слоте родительского фрейма; O (Override) - при отсутствии значения в текущем слоте оно наследуется из фрейма верхнего уровня].

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

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

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

19. Слоты. Значение слота. Присоединенные процедуры.

Слот имеет уникальное в пределах фрейма имя; в качестве имени слота может выступать любой произвольный текст. Имена системных слотов обычно зарезервированы, в различных системах они могут иметь различные значения. Системные слоты могут иметь имена: AKO, RELATIONS и т. д. Системные слоты служат для редактирования базы знаний и управления выводом во фреймовой системе.

Слот может содержать не только конкретное значение, но также имя процедуры, позволяющей вычислить это значение по заданному алгоритму. Такие процедуры называются присоединенными или связанными процедурами. Вызов связанной процедуры осуществляется при обращении к слоту, в котором она помещена. Например, слот «возраст» может одержать имя процедуры, которая вычисляет возраст по дате рождения, записанной в другом слоте, и текущей дате.

В слоте могут содержаться данные сложных типов: массивы, списки, множества, фреймы и т. д. Значение слота может представлять собой арифметическое значение, фрагмент текста и т. д.

Формально структура фрейма может быть представлена как список свойств:

```
(<имя_фрейма>
  <имя_слота1> (<значение_слота1>) <тип_значения_слота1>
  <имя_слота2> (<значение_слота2>) <тип_значения_слота2>
  .
  .
  .
  <имя_слота N> (<значение_слота N>) <тип_значения_слота N>)
```

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

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

Присоединенные процедуры:

Во фреймах различают два типа присоединенных процедур: процедуры-демоны и процедуры-слуги.

Демоны (процедуры) автоматически запускаются при обращении к соответствующему слоту. Типы демонов связаны с условием запуска процедуры. Демон с условием IF-NEEDED запускается, если в момент обращения к слоту его значение не было установлено. Демон IF-ADDED запускается при попытке изменения слота. Демон IF-REMOVED запускается при попытке удаления значения слота. Возможны также другие типы демонов. Процедуры-слуги запускаются явно.

В качестве значения слота может быть присоединенная процедура, называемая **служебной** в языке LISP или **методом** в языках объектно-ориентированного программирования. Присоединенная процедура запускается по сообщению, передаваемому из другого фрейма. Демоны и присоединенные процедуры являются процедурными знаниями, объединенными с декларативными в единую систему.

Фреймовые системы в последние годы реализуют на объектно-ориентированных языках, но в этих языках нет средств для реализации присоединенных процедур, поэтому требуется интеграция объектно-ориентированного языка с другими средствами обработки знаний (например, с языком PROLOG). Существуют также специализированные языки, такие как семейство KL(1) представления знаний на основе фреймовой модели.

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

Если проследить за речью человека (не ее содержанием, а структурным построением фраз), то легко заметить, что значительная часть наших мыслей оформляется в виде правил типа «ЕСЛИ (далее следует ряд условий), ТО (далее следует что надо сделать)».

Примеры: «Если ты будешь хорошо учиться, то папа купит тебе велосипед», «Закаляйся, если хочешь быть здоров», «(Если) Не продается вдохновенье, но (то) можно рукопись продать» и т.п.

Несмотря на то, что такая форма представления знаний чрезвычайно популярна для человека, она является достаточно сложной для реализации в виде жестких алгоритмических схем. Проблемы появляются в связи с тем, что знания многообразны как по уровням, так и по спектру, допускают неоднозначное толкование, чаще всего имеют ассоциативный характер использования. Трудно четко разложить их на причины и следствия и описывать их, например, на языке исчисления высказываний и предикатов. Добавим еще, что знания постоянно меняются, пополняются, модифицируются и приспосабливаются к условиям среды.

Термин «продукция» был введен американским логиком Е. Постом в 40-х годах в работах по обоснованию и формализации алгоритмических систем. Продукцией называется правило вида $\alpha \rightarrow \beta$, где α, β - слова над некоторым алфавитом А. Совокупность правил типа $\alpha \rightarrow \beta$ получила название системы продукции.

Рассмотрим этот механизм представления, интерпретируя левую и правую части правил продукции как логические функции. Наиболее популярная форма такая:

$$P(x, y) \rightarrow A \& D. \quad (1)$$

Здесь $P(x, y)$ - логическое выражение; A - заключение; D - действие, то есть продукция - это пара **<антецедент, консеквент>**, а **« \rightarrow »** - знак секвенции. Для простоты изложения кванторы опущены. Выход за рамки формальной системы осуществляется путем добавления такой понятийной категории, как действия.

Если при замене x и y конкретными значениями $a \in X, b \in Y$ $P(a, b)$ принимает значение И, то выдается заключение A и выполняется действие D . Формы заключения могут быть различными:

$P(x, y) \rightarrow A$ (здесь A - статическое высказывание, истинное всегда);

$P(x, y) \rightarrow D$ (здесь D - одно или несколько действий, не требующих интерпретации);

$P(x, y) \rightarrow A(x, y) \& D(x, y)$ (здесь $A(x, y)$ - логическая функция, истинная при $P(x, y) = N$, $D(x, y)$ - действие, которое доопределяется путем замены x, y на a, b соответственно).

В отличие от импликации случай $I \rightarrow L$ здесь не рассматривается. Это значит, что при истинности левой части реализуется все, что рекомендуется в правой, то есть продукция истинна всегда.

В простейшем случае правая и левая части продукции могут быть простыми высказываниями. Главное отличие продукции от логических формул - они допускают более широкий спектр интерпретации.

Основным ограничением продукции правил Поста, с точки зрения программирования, является отсутствие стратегии управления, которая позволяла бы регламентировать применение правил. Понять это ограничение можно легко, представив гипотетическую ситуацию с визитом в библиотеку. Если для поиска нужной книги не использовать никакой системы управления процессом, можно потратить уйму времени просматривая все возможные варианты.

21. Марковский и rete-алгоритм. Триплет объект-атрибут-значение. Достоинства и недостатки продукции систем.

Марковский алгоритм - упорядоченная группа продукции, применяемых согласно приоритетов к входной символьной строке.

Если правило с высшим приоритетом является непригодным, то используется следующее правило с низким приоритетом и т.д.

Марковский алгоритм завершает свою работу после выявления одного из следующих условий:

1. Последняя продукция не была применима к строке.
2. Была применена продукция, которая заканчивается точкой.

Марковский алгоритм не является эффективным способом создания систем с большим количеством правил.

Такие системы требуют алгоритма, который имеет полную информацию обо всех правилах, и может применить любое из них, не делая попытки последовательно проверяя каждое.

Решение - rete-алгоритм, (Чарльз Л. Форго в 1979 год)

Rete-алгоритм функционирует как сеть, предназначенная для хранения большого объема знаний.

Основан на использовании динамической структуры данных, которая автоматически реорганизуется с целью оптимизации поиска аналогично В-дереву, который применяется при индексации структур реляционных баз данных.

Rete-алгоритм - высокоскоростное средство сравнения фактов с шаблонами, быстродействие которого достигается благодаря хранению в оперативной памяти информации о правилах, которые находятся в сети.

В rete-алгоритме воплощены два эмпирических наблюдения, на основании которых была предложена структура данных:

Временная избыточность – действие, оказываемое в результате запуска одного из правил, обычно связана с несколькими фактами.

Структурное сходство – один и тот же шаблон часто находится в левой части более чем одного правила.

Факты описываются с помощью триплетов объект-атрибут-значение.

В этом случае отдельная сущность рассматривается как объект, а данные, хранящиеся в рабочей памяти, показывают значения, которые принимают атрибуты этого объекта.

Примеры триплетов:

кошка — кличка — Мурка;

кошка — окрас — рыжий;

кошка — порода — сибирская.

Правила из базы правил при триплетах могут срабатывать более одного раза в процессе одного логического вывода, так как одно правило может применяться к различным экземплярам объекта (но не более одного раза к каждому экземпляру).

Достоинства — простота представления знаний и организация логического вывода.

1. Подавляющая часть человеческих знаний может быть записана в виде продукции.

2. Системы продукции являются модульными. За небольшим исключением удаление и добавление

продукции не приводит к изменениям в остальных продукциях.

3. При необходимости системы продукции могут реализовывать любые алгоритмы и, следовательно, способны отражать любое процедурное знание, доступное ЭВМ.

4. Наличие в продукциях указателей на сферу применений продукции позволяет наиболее эффективно организовать память, сократив время поиска в ней необходимой информации. Классификация сфер может быть многоуровневой, что еще более повышает эффективность поиска знаний, так как позволяет наследовать информацию в базе знаний.

5. При объединении систем продукции и сетевых представлений получаются средства, обладающие большой вычислительной мощностью.

6. Естественный параллелизм в системе продукции, асинхронность их реализации делают производственные системы удобной моделью вычислений для ЭВМ новой архитектуры, в которой идея параллельности и асинхронности является центральной.

Недостатки:

1. При большом числе продукции становится сложной проверка непротиворечивости системы продукции. Это заставляет при добавлении новых продукции тратить много времени на проверку непротиворечивости новой системы.

2. Из-за присущей системе недетерминированности (неоднозначного выбора выполняемой продукции из множества активизированных продукции) возникают принципиальные трудности при проверке корректности работы системы, Традиционно база знаний создаётся предметным экспертом, но такой подход не всегда может гарантировать качество управления.

Эта проблема может быть устранена на основе использования нейросетевых методов и технологий, нечёткой логики, а также введением физических и информационных ограничений в формализованное описание модели объектов управления.

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

База знаний состоит из набора правил, а сама экспертная система должна содержать три основные компоненты: **базу правил, рабочую память и механизм вывода**.

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

Рабочая память (РП) - область памяти, в

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

Механизм вывода выполняет две основные

функции: просмотр существующих в рабочей памяти фактов и правил из БП, а также добавление в РП новых фактов; определение стратегии поиска и применения правил.

Компонента вывода и управляющая компонента

Для выполнения указанных функций механизм вывода включает **компоненту вывода и управляющую компоненту**.

Действие компоненты вывода основано на применении правила вывода, называемого *Modus Ponendo Ponens* (правило отделения), которое звучит в производственных системах так: «если верно А и верно, что из А следует В, то верно В». То есть если в РП присутствует факт А, и в БП существует правило вида «ЕСЛИ А, ТО В», то факт В признается истинным и заносится в рабочую память. Такой вывод легко реализуется на компьютере.

Невысокие интеллектуальные возможности производственных систем обусловлены тем, что ЭС способны вывести лишь небольшое количество заключений, используя заданное множество правил. При этом компонента вывода ЭС должна иметь способность функционировать в условиях недостатка информации.

Операции цикла вывода

Управляющая компонента определяет, порядок применения правил, а также устанавливает факты, которые могут быть изменены в случае продолжения работы. Механизм вывода работает циклически, при этом в одном цикле может срабатывать только одно правило. Схема цикла показана на рис. 8.1. В цикле выполняются следующие операции:

- 1) **сопоставление** - образец правила сопоставляется с имеющимися в РП фактами;
- 2) **выбор** - если в конкретной ситуации может быть применено сразу несколько правил, то из них выбирается одно, наиболее подходящее по заданному критерию, т. е. происходит разрешение конфликтного выбора;

- 3) **срабатывание** - если образец правила при сопоставлении совпал с какими-либо фактами из РП, то правило срабатывает, оно отмечается в БП;
- 4) **действие** - рабочая память подвергается изменению путем добавления в нее заключения сработавшего правила.

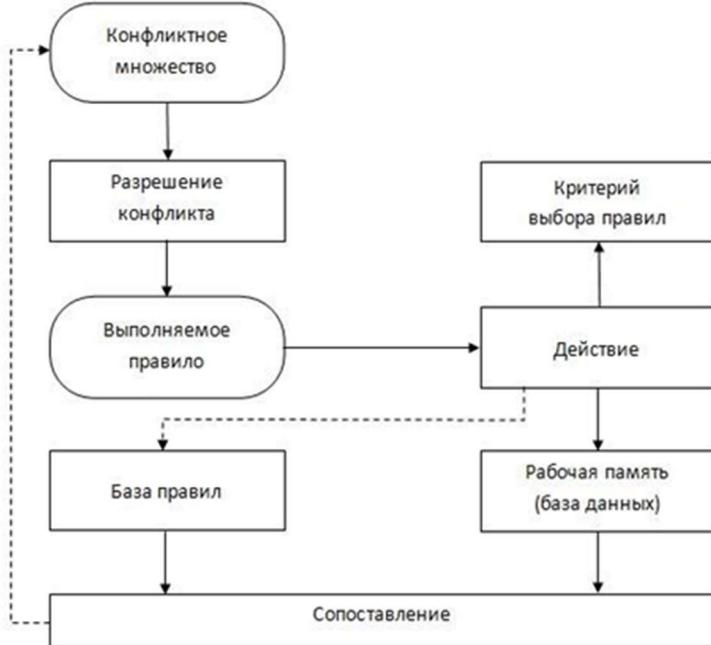


Рис. 8.1. Цикл работы механизма вывода

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

Особенностью ЭС является то, что они не имеют процедур, способных построить в пространстве состояний сразу весь путь решения задачи. Траектория поиска решения полностью определяется данными, получаемыми от пользователя в процессе вывода.

Стратегия управления выводом зависит от выбранного метода поиска, описанного выше, от него будет зависеть порядок применения и срабатывание правил. Процедура вывода сводится к определению направления поиска и способа его осуществления. Процедуры, реализующий поиск, как правило, недоступны инженерам по знаниям. При выборе стратегии необходимо решить, какую точку в пространстве состояний принять в качестве исходной (в прямом или обратном направлении), какова стратегия перебора (в ширину, в глубину, по подзадачам или иначе).

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

Про то, что такое демоны и присоединенные процедуры, написано в 19 вопросе

**Если кратко - демоны запускаются автоматически по какому-то условию,
присоединенные процедуры - вручную**

В интеллектуальных системах с фреймовым представлением знаний используются три способа управления логическим выводом: демоны, присоединенные процедуры и механизм наследования, который является основным механизмом вывода.

Управляющие функции механизма наследования заключаются в автоматическом поиске и определении значений слотов фреймов нижележащих уровней по значениям слотов фреймов верхних уровней, а также в запуске присоединенных процедур и демонов.

Присоединенные процедуры и демоны могут реализовать любой механизм вывода, однако, эта реализация имеет конкретный характер и требует значительных затрат труда проектировщиков и программистов.

Итак, в интеллектуальных системах с фреймовым представлением знаний невозможно четко отделить процедурные знания от декларативных, поскольку демоны и присоединенные процедуры одновременно являются и знаниями, и средствами управления логическим выводом.

При попытке вставить значение в слот Место проведения автоматически запускается демон ЗАКАЗ, который является процедурой. Демон КТО? автоматически запускается при обращении к слоту Докладчик. Если значение этого слота не определено, то происходит генерация запроса «Кто выступает?», получение ответа и его запись в качестве значения слота.

Реализация вывода с помощью присоединенных процедур требует наличия механизма обмена между фреймами, таким механизмом обычно бывает механизм сообщений.

В семантических сетях, так же как при фреймовом представлении знаний, декларативные и процедурные знания не разделены, следовательно, база знаний не отделена от механизма вывода. Процедура вывода обычно представляет совокупность процедур обработки сети.

Правила вывода в логических моделях

Для рассмотрения вывода в логических моделях необходимо ввести следующие определения.

Неравнозначностью (исключающим «или», сложением по модулю 2 двух высказываний А и В называется высказывание, истинное, когда истинностные значения А и В не совпадают, и ложное в противном случае. Обозначение: $A \oplus B$, читается «или А, или В», «или» понимается в разделительном смысле. Составное высказывание «Сегодня понедельник или вторник» состоит из двух простых: А = «Сегодня понедельник»; В = «Сегодня вторник». Эти высказывания соединены связкой «или» в разделительном смысле. Таким образом, данное высказывание представимо формулой $A \oplus B$.

Если описание системы представлено совокупностью сложных высказываний—логических формул, истинных для данной системы, то с помощью логических преобразований могут быть получены новые представления, характеризующие данную систему (истинные для данной системы). Таким образом, появляется возможность вывода новых знаний из имеющихся с помощью допустимых в логике преобразований. Итак, операция, называемая выводом, — это получение заключения из группы заданных представлений; вывод является формулой, полученной из некоторой группы описанных формул. Как правило, эти формулы истинны при каких бы то ни было допустимых интерпретациях.

Действие компоненты вывода в логических моделях основано на применении следующих правил вывода:

Modus Ponens «если верно А и верно, что из А следует В, то верно В»:

$A \rightarrow B, AB.$

Modus Tollens «если верно, что из А следует В и В ложно, то ложно А»:

$A \rightarrow B, \neg B \neg A.$

Modus Ponendo Tollens «если верно или А или В (в разделительном смысле) и верно одно из них, то другое ложно»:

$A \oplus B, A \neg B, A \oplus B, B \neg A.$

Modus Tollendo Ponens

«если верно А или В (в разделительном смысле) и ложно одно из них, то верно другое»:

$A \oplus B, \neg AB, A \oplus B, \neg B A.$

«если верно А или В (в неразделительном смысле) и ложно одно из них, то верно другое»:

$AvB, \neg AB, AvB, \neg B A.$

Правило транзитивности «если из А следует В, а из В следует С, то из А следует С»:

$A \rightarrow B, B \rightarrow CA \rightarrow C$

Закон противоречия «если из А следует В и В, то ложно А»:

$A \rightarrow B, A \rightarrow \neg B \neg A$

Правило контрапозиции (или метод резолюций) «если из А следует В, то из того, что ложно В, следует, что ложно А»:

$A \rightarrow B \neg B \rightarrow \neg A.$

Пример:

Студент сдал экзамен на «хорошо» или «отлично». Он не получил отличной оценки. Следовательно, он сдал экзамен на «хорошо».

Введем обозначения: А = «студент сдал экзамен на хорошо», В = «студент сдал экзамен на отлично», тогда приведенное рассуждение описывается схемой
 $A \oplus B, \neg B A.$

которая, как показано выше, является схемой правильного рассуждения.

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

Структура экспертных систем

Обобщенная структура ЭС представлена на рис. 1.

Реальные ЭС могут иметь более сложную структуру, однако, блоки, показанные на рис. 1, обязательно присутствуют в реальных ЭС.

Идеальная экспертная система должна содержать пять основных компонентов (рис. 1):

1. интерфейс с пользователем;

2. подсистему логического вывода;
3. базу знаний;
4. модуль отображения и объяснения решений.
5. модуль приобретения знаний;

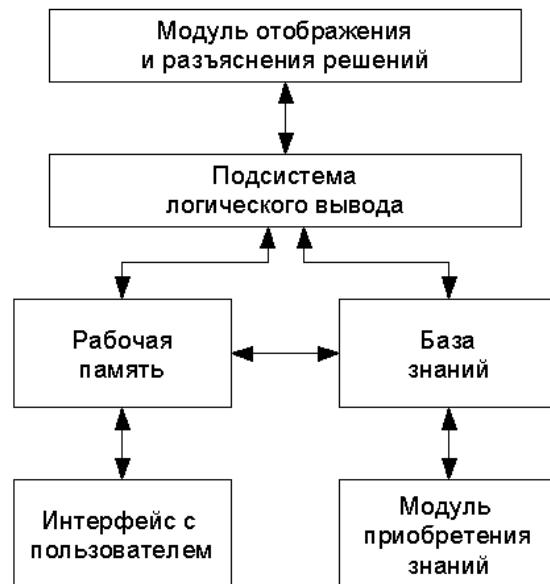


Рис. 1. Структура экспертной системы

Взаимодействие пользователя с экспертной системой осуществляется через **интерфейс с пользователем**. В программах, реализующих интерфейс с пользователем, происходит преобразование (трансляция) предложений естественного языка (или другого проблемно-ориентированного языка непроцедурного типа) на внутренний язык представления знаний данной экспертной системы.

Описание задачи пользователя на выбранном языке представления знаний поступает в **подсистему логического вывода**, которая использует информацию из базы знаний, генерирует рекомендации по решению искомой задачи. Основу базы знаний составляют факты и правила. В подсистеме логического вывода реализуется некоторая стратегия выбора соответствующего факта из базы знаний, которая зависит от способа представления знаний в ЭС и характера решаемых задач.

Модуль отображения и объяснения решения демонстрирует промежуточные и окончательные решения задачи и объясняет пользователю

действия системы. Считается, что если метод рассуждений не может быть объяснен пользователю, то он должен быть признан неудовлетворительным.

Функция **модуля приобретения знаний** состоит в поддержке процесса извлечения знаний о соответствующей узкоспециализированной предметной области.

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

Классификация:

1. По типу решаемых задач:
 1. Интерпретирующие системы предназначены для формирования описания ситуаций по результатам наблюдений(распознавания)
 2. Прогнозирующие системы предназначены для логического анализа возможных последствий заданных ситуаций или событий.
 3. Диагностические системы предназначены для обнаружения источников неисправностей по результатам наблюдений
 4. Системы проектирования предназначены для структурного синтеза конфигураций объектов при заданных ограничениях.
 5. Системы планирования предназначены для подготовки последовательности операций, приводящих к заданной цели.
 6. Системы мониторинга анализируют поведение контролируемой системы и прогнозируют вероятность достижения поставленной цели
 7. Наладочные системы предназначены для выработки рекомендаций по устранению неисправностей
 8. Системы оказания помощи при ремонте оборудования выполняют планирование процесса устранения неисправностей в сложных объектах
 9. Обучающие системы проводят анализ знаний студентов по определенному предмету, отыскивают пробелы в знаниях и предлагают средства для их ликвидации
 10. Системы контроля обеспечивают адаптивное управление поведением сложных человеко-машинных систем, прогнозируяявление сбоев и планируя действия, необходимые для их предупреждения

2. По способу и учета временного признака
 1. Статические ЭС. Данные не меняются во времени
 2. Квазидинамические ЭС описывают ситуацию, которая меняется с некоторым интервалом времени
 3. Динамические системы работают с датчиками объектов в реальном времени
3. По степени интеграции
 1. Автономные
 2. Гибридные

Экспертные системы отличаются от прочих прикладных программ по следующим признакам:

- a. Экспертные системы моделируют механизм мышления человека
- b. Экспертная система формирует определенные соображения и выводы, основываясь на тех знаниях, которыми она располагает
- c. При решении задач основными являются эвристические и приближенные методы
- d. Экспертные системы имеют дело с предметами реального мира, операции с которыми требуют наличия большого опыта, накопленного человеком
- e. Одной из основных характеристик ЭС является ее производительность
- f. Экспертная система должна обладать способностью объяснить полученное решение, доказать его обоснованность
- g. компетентность - в конкретной предметной области экспертная система должна достигать того же уровня, что и эксперты
- h. символные рассуждения - знания, на которых основана экспертная система, представляют в символном виде понятия предметной области
- i. глубина - экспертная система должна решать нетривиальные задачи, отличающиеся сложностью используемых знаний или обилием информации
- j. самосознание - экспертная система должна включать в себя механизм объяснения того, каким образом она приходит к решению задачи

Цикл ЭС

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

Приобретение знаний - это передача потенциального опыта решения проблемы от некоторого источника знаний и преобразование его в вид, который позволяет использовать эти знания в программе. В этом режиме общение с ЭС осуществляется экспертом проблемной области через посредничество инженера по знаниям.

В режиме консультации общение с ЭС осуществляется конечный пользователь, которого интересует результат и (или) способ получения решения. В режиме консультации данные о задаче пользователя обрабатываются диалоговой компонентой, которая распределяет роли участников (пользователя и ЭС), организует их взаимодействие в процессе кооперативного решения задачи, преобразует данные пользователя о задаче во внутренний язык системы, преобразует сообщения системы, представленные на внутреннем языке, в язык пользователя. Таким образом, процесс работы ЭС в режиме консультации может быть представлен в виде схемы (рис. 2). После работы данные поступают в РП. На основе входных данных из РП, общих данных о проблемной области и правил из БЗ решатель (интерпретатор) формирует решение задачи. В отличие от традиционных программ ЭС в режиме решения задачи не только исполняет предписанную последовательность действий, но предварительно ее формирует. Если ответ не понятен пользователю, он может потребовать объяснения, как ответ получен.

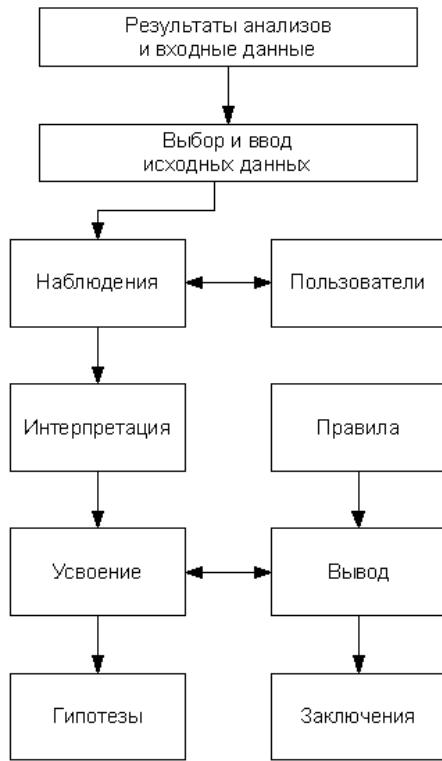


Рис. 2. Схема работы ЭС

Такой процесс продолжается до тех пор, пока не поступит информация, достаточная для окончательного заключения.

Промышленная технология создания ЭС включает следующие этапы:

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

Проектирование ЭС основано на трех главных принципах:

- Мощность ЭС обусловлена мощностью БЗ и возможностями ее пополнения

- b. Знания, позволяющие эксперту (или экспертной системе) получать эффективные решения, являются правдоподобными, эвристическими, эмпирическими, неопределенными
- c. Неформальный характер решаемых задач и используемых знаний делает необходимым обеспечение активного диалога пользователя с ЭС при ее работе

При разработке ЭС используется концепция «быстрого прототипа». Разработчики на начальном этапе создают прототип ЭС (возможно не единственный), который должен удовлетворять двум противоречивым требованиям:

- 1) он должен уметь решать типовые задачи конкретного приложения,
- 2) трудоемкость и время его разработки должны быть незначительны.

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

Нечеткая логика (НЛ) - наиболее удобный способ построения систем управления сложными технологическими процессами.

НЛ позволяет определить промежуточные значения для таких общепринятых оценок, как да-нет, истинно-ложно, черное-белое и т. п.

Нечеткие множества

Самым главным понятием систем, основанных на нечеткой логике, является понятие нечеткого множества (НМ).

Из классической математики известно понятие четких (определенных) множеств.

Множество A - четкое множество, если A - часть некоторого универсального для данной прикладной задачи множества U , характеризующегося следующими условиями:

- все элементы множества четко различимы между собой, в множестве нет повторяющихся элементов, нескольких экземпляров некоторых элементов;
- относительно каждого элемента можно четко определить, принадлежит ли он данному множеству или нет.

Эти условия позволяют характеризовать четкое множество его характеристической функцией, заданной на универсальном множестве U и принимающей значения в множестве $\{0, 1\}$:

$$\mu_A(u) = \begin{cases} 0, & u \notin A, \\ 1, & u \in A; \end{cases} \quad u \in U.$$

Строгое представление нечетких множеств

Пусть E - универсальное (universal) или несущее множество, x - элемент E , а R - некоторое свойство. Определим для несущего множества E обычное (четкое) подмножество A , элементы которого удовлетворяют свойству R , как множество упорядоченных пар

$$= \{ () / \},$$

где $\mu ()$ — характеристическая функция, принимающая значение 1, если элемент x удовлетворяет свойству R , и 0 — в противном случае.

Нечеткое подмножество отличается от обычного тем, что для элементов x из множества E нет однозначного ответа «да-нет» относительно свойства R . В связи с этим, нечеткое подмножество A универсального множества E определяется как множество упорядоченных пар с характеристической функцией принадлежности $\mu(x)$, принимающей значения в некотором вполне упорядоченном множестве M , например, $M=[0, 1]$.

Функция принадлежности указывает степень (или уровень) принадлежности элемента x подмножеству A . Множество M называется множеством принадлежности. Если $M=\{0, 1\}$, то нечеткое подмножество A может рассматриваться как обычное или четкое множество.

Основные характеристики нечетких множеств

Пусть $M=[0, 1]$ и A — нечеткое множество с элементами из универсального (несущего) множества E и множеством принадлежности M .

Тогда высотой нечеткого множества называется верхняя граница значений его функции принадлежности:

$$\in \mu ().$$

Нормальным называется нечеткое множество, высота которого равна 1. Если высота меньше 1, нечеткое множество называется субнормальным.

Говорят, что нечеткое множество пусто, если $\forall x \in E \mu(x) = 0$.

Непустое субнормальное множество можно нормализовать по формуле

$$\mu' = \frac{\mu(x)}{\max_{x \in E} \mu(x)}.$$

Нечеткое множество является унимодальным, если $\mu(x) = 1$ только на одном элементе x из универсального множества E .

Носителем нечеткого множества A (обозначается как $\text{supp } A$) является обычное подмножество со свойством $\mu(x) > 0$, т. е. $A = \{x \mid \mu(x) > 0\}$.

Элементы $x \in E$, для которых $\mu(x) = 0,5$, называются точками перехода множества A .

Более подробно в лекции № 9.

[Лекция 09 часть1 2018.Принятие решений в условиях неопределенности](#)

26. Операции над нечеткими множествами. Логические операции. Алгебраические операции.

Как и с обычными множествами, с нечеткими множествами можно производить стандартные операции: **объединение, пересечение, декартово произведение** и т. д. Отличие заключается лишь в том, что реализации этих

операций в разных задачах могут отличаться. Здесь мы рассмотрим наиболее распространенные интерпретации действий с нечеткими множествами.

Логические операции

Включение. Пусть A и B — нечеткие множества на универсальном множестве U . Говорят, что A содержится в B , если

$$\forall x \in U \quad \mu_A(x) \leq \mu_B(x).$$

Обозначение: $A \subset B$.

Например, если A - множество чисел, очень близких к 10, а B - множество чисел, близких к 10, то $A \subset B$. Формально это можно проверить, используя функции принадлежности. Если A и B - обычные множества, а $\mu_A(x)$ и $\mu_B(x)$ - характеристические функции, то из неравенства (1) следует, что если некоторый элемент x принадлежит множеству A , т. е. $\mu_A(x) = 1$, то он принадлежит и множеству B , поскольку $\mu_B(x) = 1$.

Множества A и B равны, если $\forall x \in U \quad \mu_A(x) = \mu_B(x)$.

Обозначение: $A = B$.

Объединение. Объединением нечетких множеств A и B называется нечеткое множество, обозначаемое $A \cup B$, функция принадлежности которого определяется как:

$$\forall x \in U \quad \mu_{A \cup B}(x) = (\mu_A(x), \mu_B(x)).$$

Иначе говоря, объединением A и B называется наименьшее нечеткое подмножество, включающее как A , так и B .

Пересечение. Пересечением A и B называется наибольшее нечеткое подмножество, содержащееся одновременно в множествах A и B :

$$\mu_{A \cap B}(x) = (\min(\mu_A(x), \mu_B(x))).$$

Дополнение. Дополнение \underline{A} нечеткого множества A имеет функцию принадлежности $\mu_{\underline{A}}(x) = 1 - \mu_A(x)$. Обозначение: $\underline{A} = \underline{B}$ или $\underline{\underline{A}} = \underline{\underline{B}}$.

Разность. Разностью $A - B = A \cap \underline{B}$ называют множество с функцией принадлежности

$$\mu_{A - B}(x) = (\max(0, \mu_A(x) - \mu_B(x))).$$

Для операций **пересечения** и **объединения** выполняются следующие свойства (A, B, C - нечеткие множества; U - пустое множество, т. е. $\forall x \in U \quad \mu_U(x) = 0$; U - универсальное множество):

- 1) $A \cap B = B \cap A, \quad A \cup B = B \cup A$ (коммутативность);
- 2) $(A \cap B) \cap C = A \cap (B \cap C),$
 $(A \cup B) \cup C = A \cup (B \cup C)$ (ассоциативность);
- 3) $A \cap A = A, \quad A \cup A = A$ (идемпотентность);
- 4) $A \cap (B \cup C) = (A \cap B) \cup (A \cap C),$
 $A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$ (дистрибутивность);
- 5) $A \cup \emptyset = A;$
- 6) $A \cap \emptyset = \emptyset;$
- 7) $A \cap E = A;$
- 8) $A \cup E = E;$
- 9) $\overline{A \cap B} = \overline{A} \cup \overline{B}, \quad \overline{A \cup B} = \overline{A} \cap \overline{B}$ (теоремы де Моргана).

В отличие от четких множеств, для нечетких множеств в общем случае имеем:

$$A \cap \overline{A} \neq \emptyset, \quad A \cup \overline{A} \neq E.$$

Алгебраические операции

Алгебраическое произведение нечетких множеств A и B обозначается $A \cdot B$ и определяется так:

$$\forall x \in E \quad \mu_{A \cdot B}(x) = \mu_A(x) \cdot \mu_B(x).$$

Алгебраическая сумма этих множеств обозначается $A \hat{+} B$ и определяется так:

$$\forall x \in E \quad \mu_{A \hat{+} B}(x) = \mu_A(x) + \mu_B(x) - \mu_A(x) \mu_B(x).$$

Для операций \cdot , $\hat{+}$ выполняются следующие свойства:

- 1) $A \cdot B = B \cdot A, \quad A \hat{+} B = B \hat{+} A$ (коммутативность);
- 2) $(A \cdot B) \cdot C = A \cdot (B \cdot C),$
 $(A \hat{+} B) \hat{+} C = A \hat{+} (B \hat{+} C)$ (ассоциативность);
- 3) $A \cdot \emptyset = \emptyset, \quad A \hat{+} \emptyset = A,$
 $A \cdot E = A, \quad A \hat{+} E = E;$
- 4) $\overline{A \cdot B} = \overline{A} \hat{+} \overline{B}, \quad \overline{A \hat{+} B} = \overline{B \hat{+} A} = \overline{B} \cdot \overline{A}$ (теоремы де Моргана).

В отличие от четких множеств, для нечетких множеств следующие свойства, вообще говоря, не выполняются:

- 1) $A \cdot A = A, \quad A \hat{+} A = A$ (идемпотентность);
- 2) $A \cdot (B \hat{+} C) = (A \cdot B) \hat{+} (A \cdot C),$
 $A \hat{+} (B \cdot C) = (A \hat{+} B) \cdot (A \hat{+} C)$ (дистрибутивность);
- 3) $A \cdot \overline{A} = \emptyset, \quad A \hat{+} \overline{A} = E.$

При совместном использовании операций \cap , \cup , $\hat{+}$, \cdot выполняются следующие свойства:

- 1) $A \cdot (B \cup C) = (A \cdot B) \cup (A \cdot C);$
- 2) $A \cdot (B \cap C) = (A \cdot B) \cap (A \cdot C);$
- 3) $A \hat{+} (B \cup C) = (A \hat{+} B) \cup (A \hat{+} C);$
- 4) $A \hat{+} (B \cap C) = (A \hat{+} B) \cap (A \hat{+} C).$

Умножение на число. Если α — положительное число такое, что $\alpha \max_{x \in A} \mu_A(x) \leq 1$, то нечеткое множество αA имеет функцию принадлежности $\mu_{\alpha A}(x) = \alpha \mu_A(x)$.

Дизъюнктивная сумма:

$$A \oplus B = (A - B) \cup (B - A) = (A \cap \bar{B}) \cup (\bar{A} \cap B)$$

с функцией принадлежности

$$\mu_{A \oplus B}(x) = \max\{\min[\mu_A(x), 1 - \mu_B(x)], \min[1 - \mu_A(x), \mu_B(x)]\}.$$

Выпуклая комбинация нечетких множеств. Пусть A_1, A_2, \dots, A_n — нечеткие множества универсального множества E , а w_1, w_2, \dots, w_n — неотрицательные числа, сумма которых равна 1. Тогда выпуклой комбинацией A_1, A_2, \dots, A_n называется нечеткое множество A с функцией принадлежности

$$\forall x \in E \quad \mu_A(x) = w_1 \mu_{A_1}(x) + w_2 \mu_{A_2}(x) + \dots + w_n \mu_{A_n}(x).$$

Декартово (прямое) произведение нечетких множеств. Пусть A_1, A_2, \dots, A_n — нечеткие подмножества универсальных множеств E_1, E_2, \dots, E_n соответственно. Декартовым или прямым произведением $A = A_1 \times A_2 \times \dots \times A_n$ называется нечеткое подмножество множества $E = E_1 \times E_2 \times \dots \times E_n$ с функцией принадлежности

$$\mu_A(x_1, x_2, \dots, x_n) = \min(\mu_{A_1}(x_1), \mu_{A_2}(x_2), \dots, \mu_{A_n}(x_n)).$$

Оператор увеличения нечеткости используется для преобразования четких множеств в нечеткие и для увеличения нечеткости нечеткого множества.

Пусть A — нечеткое множество, E — универсальное множество, и для всех элементов $x \in E$ определены нечеткие множества $k(x)$. Совокупность всех множеств $k(x)$ называется ядром оператора увеличения нечеткости Φ . Результатом действия оператора Φ на нечеткое множество A является нечеткое множество вида

$$\Phi(A, k) = \bigcup_{x \in E} \mu_A(x) k(x),$$

где $\mu_A(x) k(x)$ — произведение числа на нечеткое множество.

27. Четкое множество « α -уровня». Нечеткая и лингвистическая переменные. Нечеткие числа.

Множеством α -уровня нечеткого подмножества A универсального множества E называется четкое подмножество A_α универсального множества E , определяемое в виде $A_\alpha = \{x | \mu_A(x) \geq \alpha\}$, где $\alpha \leq 1$.

Примеры

8. Пусть $A = 0,2/x_1 + 0/x_2 + 0,5/x_3 + 1/x_4$, тогда $A_{0,3} = \{x_3, x_4\}$, $A_{0,7} = \{x_4\}$.

Достаточно очевидно следующее свойство: если $\alpha_1 > \alpha_2$, то $A_{\alpha_1} \subset A_{\alpha_2}$.

9. Пусть $E = \{1, 2, \dots, 10\}$;

$A = \text{«малые числа»} = 1/1 + 1/2 + 0,8/3 + 0,5/4 + 0,4/5 + 0,1/6$;

$B = \text{«большие числа»} = 0,1/5 + 0,2/6 + 0,5/7 + 0,8/8 + 1/9 + 1/10$.

Тогда:

$\bar{A} = \text{«НЕ малые числа»} = 0,2/3 + 0,5/4 + 0,6/5 + 0,9/6 + 1/7 + 1/8 + 1/10$;

$A \cap B = \text{«малые числа И большие числа»} = 0,1/5 + 0,1/6$;

$A \cup B = \text{«малые числа ИЛИ большие числа»} =$

$= 1/1 + 1/2 + 0,8/3 + 0,5/4 + 0,4/5 + 0,2/6 + 0,5/7 + 0,8/8 + 1/9 + 1/10$.

Приведенные определения операций над нечеткими множествами являются наиболее распространенными.

Нечеткая и лингвистическая переменные

Используются при описании объектов и явлений с помощью нечетких множеств.

Лингвистические переменные (ЛП) - средство моделирования нечеткости естественного человеческого языка, в котором границы между отдельными понятиями часто неявны и размыты.

ЛП ставят в соответствие качественным значениям некоторую количественную интерпретацию с заданной долей уверенности.

Сферой применения ЛП - нечеткий логический вывод, где истинность логических высказываний описывается не значениями ЛОЖЬ и ИСТИНА или 0 и 1, а множеством значений в интервале $[0, 1]$.

В основе понятия ЛП - термин **«нечеткая переменная»**, обозначающий нечеткое множество, которому было присвоено некоторое наименование.

Нечеткая переменная характеризуется тройкой $\langle \alpha, X, A \rangle$,

где α - наименование переменной;

X - универсальное множество (область определения α),

A - нечеткое множество на X , описывающее ограничения (т. е. $\mu_A(x)$) на значения нечеткой переменной α .

Лингвистическую переменную можно задать как пятерку $\langle \beta, T(\beta), X, G, M \rangle$, где

β - название лингвистической переменной;

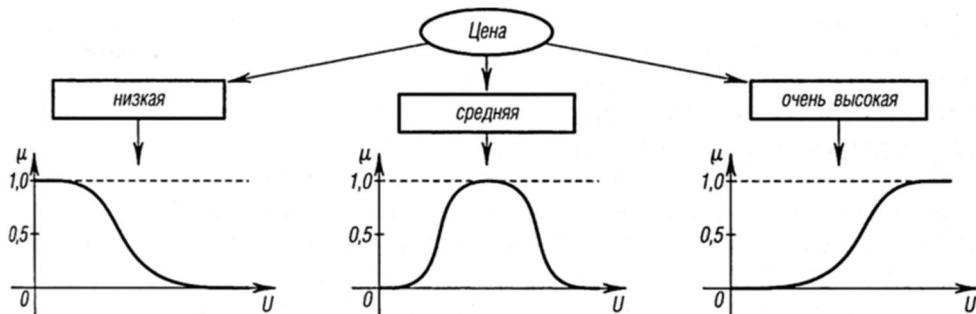
$T(\beta)$ - базовое терм-множество лингвистической переменной β , т. е. множество названий лингвистических значений переменной T_1, T_2, \dots, T_n , каждому из которых соответствует нечеткая переменная α с областью определения в виде универсального множества X .

G - синтаксическое правило (обычно грамматика), порождающее значения

лингвистической переменой;

М - семантическое правило, которое ставит в соответствие каждой нечеткой переменной α ее смысл, т. е. нечеткое подмножество универсального множества X.

Лингвистическая переменная «цена»



Основная область применения нечеткой логики - выполнение арифметических операций с нечеткими множествами, описывающими ту или иную модель или систему так, как если бы это были обычные числа.

В качестве нечетких множеств при этом могут использоваться и термы лингвистических переменных, с помощью которых исследуемая система или модель была описана в силу невозможности применения обычных, количественных математических методов.

Для снижения трудоемкости операций с нечеткими множествами используют специальный их тип — нечеткие числа.

Нечеткие числа - нечеткие переменные, определенные на числовой оси.

Иначе говоря, нечеткое число определяется как нечеткое множество A на множестве действительных чисел \mathbb{R} с функцией принадлежности $\mu_A(x) \in [0, 1]$, $(x) \in [0, 1]$, где x - действительное число, т. е. $x \in \mathbb{R}$.

Нечеткое число A **нормально**, если $\max_x \mu_A(x) = 1$;

выпуклое, если для любых чисел $x \leq y \leq z$ выполняется условие

$$\mu_A(x) = \min(\mu_A(y), \mu_A(z)).$$

Множество α – уровня нечеткого числа A определяется как $A_\alpha = \{x | \mu_A(x) \geq \alpha\}$.

Подмножество $S_A \in \mathbb{R}$ называется **носителем нечеткого числа A** (обозначается как $Supp A$), если $S_A = \{x | \mu_A(x) > 0\}$.

Нечеткое число **унимодальное**, если условие $\mu_A(x)=1$ справедливо для одной и только одной действительной точки.

Выпуклое нечеткое число A называется **нечетким нулем**, если $\mu_A(0) = supp_x \mu_A(x)$.

Нечеткое число A **положительно**, если $\forall x \in S_A \ x > 0$,

отрицательно, если $\forall x \in S_A \ x < 0$.

Нечеткий вывод: <https://neuronus.com/theory/fl/308-nechetkie-vyvody.html>

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

(лекция 11, модуль 3)

Особенности человеческого восприятия информации и современных систем управления

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

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

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

Помимо этого, в когнитивной психологии установлено, что кратковременная память ЛПР способна оперировать не более чем с 7 ± 2 факторами.

Кроме этого, памяти ЛПР свойственны такие когнитивные деформации, как: эвристика доступности (более вероятно то событие, которое легче вспомнить); эвристика конкретности (более значима простая и понятная информация); эффект Ирвина (наиболее вероятно наступление желаемого события); эвристика репрезентативности (более вероятно наступление того события, которое соответствует накопленному опыту человека).

Подводя итог всему вышесказанному, а также учитывая современные тенденции в теории принятия решений можно сделать следующие выводы:

- нельзя переоценивать значение математических методов и считать, что формальные методы математики являются универсальным средством решения задач в сфере управления военной, производственной, экономической и других видов деятельности;
- методы, основанные на результатах опыта и интуиции, будут актуальны еще продолжительное время;
- рассуждения правдоподобного характера (с позиции «здравого смысла») помогают сформировать математические модели, в основе которых лежит накопленный опыт разработки различных моделей, причем с формальной точки зрения эти математические модели можно рассматривать как некоторую систему аксиом. Другими словами, эти модели обладают совокупностью знаний, которые определяют взаимосвязь между различными наблюдениями явлений в соответствии с фундаментальной теорией, но не следующих непосредственно из этой теории. В последнее время такой подход становится общепринятым при проведении различных системных исследований в военной, экономической, социальной и других областях;

- эвристические процедуры и методы в системных исследованиях и в конкретных задачах будут иметь большое значение;

Кроме того, необходимо учитывать и то, что быстрое увеличение уровня структурной сложности современных **систем различного управления** (СУ) и большое разнообразие выполняемых ими операций предопределяет трудности принятия решений ЛПР. При этом свою деятельность они должны основывать на понимании того, что каждая СУ обладает следующими особенностями:

- **нестационарностью** (изменчивостью) отдельных параметров системы и стохастичностью своего поведения в целом;
- **的独特性和不可预测性** поведения системы в конкретных условиях, т. к. в системе присутствуют активные элементы - люди, приводящие систему к «свободе воли»;
- **способностью изменения своей структуры** при сохранении целостности;
- **формированием различных вариантов поведения**, что обусловлено наличием в системе активных элементов;
- **способностью противостоять энтропийным тенденциям**, т.е. обладают свойством гомеостатичности;
- **способностью адаптироваться** к изменяющимся условиям; способностью и стремлением к целеобразованию (в системах с активными элементами цели формируются внутри системы).

Одно время особенности систем различного управления (СУ) многие исследователи пытались преодолеть с помощью **классических систем искусственного интеллекта**, т.е. систем, основанных на символьных вычислениях и других формальных методах.

Однако довольно скоро выяснилось, что с помощью символьной обработки информации, в большинстве случаев, не удается решить прикладные задачи для сложных эргодических систем поддержки принятия решений (систем экономического планирования, социальных систем большой размерности), если для них невозможно получить полную информацию или если их определение недостаточно полно.

Как показали исследования, проведенные 1980 - 90-х годах, выходом в сложившейся ситуации явилось использование **систем на основе вычислительного интеллекта** (в зарубежной литературе чаще употребляется название - мягкие вычисления - Soft Computing).

Вычислительный интеллект

Под **вычислительным интеллектом** (ВИ) понимают научное направление, где решаются задачи искусственного интеллекта на основе новых нетрадиционных методов вычислений, а под технологией ВИ понимают совокупность нетрадиционных методов вычислений и средств обработки знаний, документооборота, методов выработки и выбора альтернативных вариантов решений, объединенных в целостную технологическую систему для принятия и доведения решений до исполнителей.

В настоящее время считают, что **ВИ включает в себя следующие основные методы:**

- нейросетевые – методы, использующие обучение, адаптацию, классификацию, системное моделирование и идентификацию систем на основе исходных данных;

- нечеткой логики – методы, основанные на теории нечетких множеств и обеспечивающей эффективные средства математического отражения неопределенности и нечеткости исходной информации, позволяющие построить модель, адекватную исследуемой предметной области;
- генетические – методы, использующие синтез, настройку и оптимизацию исследуемых систем с помощью специальным образом организованного случайного поиска и эволюционного моделирования.

Перечисленные методы считаются основными в ВИ, однако, необходимо заметить, что число новых методов, примкнувших к ним в последнее время постоянно расширяется, не являясь строго определенным. Например:

- когнитивная компьютерная графика – методы визуализации данных, позволяющие активировать наглядно-образные механизмы мышления ЛПР, облегчающие принятие решения в сложной обстановке или нахождение решения сложной проблемы;
- нелинейная динамика - это наука, изучающая структуру и свойства эволюционных процессов в нелинейных динамических системах.

(в лекции далее побольше расписаны методы ВИ)

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

Нейронная сеть [1] (также **искусственная нейронная сеть**, ИНС) — [математическая модель](#), а также её программное или аппаратное воплощение, построенная по принципу организации и функционирования биологических нейронных сетей — сетей [нервных клеток](#) живого организма.

Первоначально нейронные сети предполагалось использовать для моделирования деятельности человеческого мозга, однако в настоящее время они получили широкое применение и в технике. Искусственный интеллект (на основе искусственных нейронных сетей – *artificial neural networks*) обладает ценными «качествами» (универсальность, высокое быстродействие, способность к обучению, устойчивость к отказам) и используется для решения многих неформализуемых или трудно формализуемых задач:

- · распознавания образов;
- · распознавания и синтеза речи;
- · распознавания аэрокосмических изображений;
- · обработки сигналов при наличии больших шумов;
- · прогнозирования;
- · оптимизации;
- · прогнозирования котировок ценных бумаг и курса валют;

- · предупреждения мошенничества с кредитными карточками;
- · оценки стоимости недвижимости;
- · оценки финансового состояния предприятий и риска невозврата кредитов;
- · обработки радиолокационных сигналов;
- · контроля движения на скоростных автомагистралях и железных дорогах;
- · диагностики в медицине;
- · добычи знаний из больших объемов данных в бизнесе, финансах и научных исследованиях;
- · управления в реальном времени.

Классификация искусственных нейронных сетей

Нейронные сети различают по:

- · структуре сети (связей между нейронами);
- · особенностям модели нейрона;
- · особенностям обучения сети.

По структуре нейронные сети можно разделить (рисунок 4) на:

- · неполносвязные (или слоистые) и полносвязные;
- · со случайными и регулярными связями;
- · с симметричными и несимметричными связями.



Рисунок 4 – Классификация нейронных сетей по структуре

С точки зрения топологии можно выделить три основных типа нейронных сетей:

- · полносвязные (рисунок 5, а);
- · многослойные или слоистые (рисунок 5, б);
- · слабосвязные (с локальными связями) (рисунок 5, в).

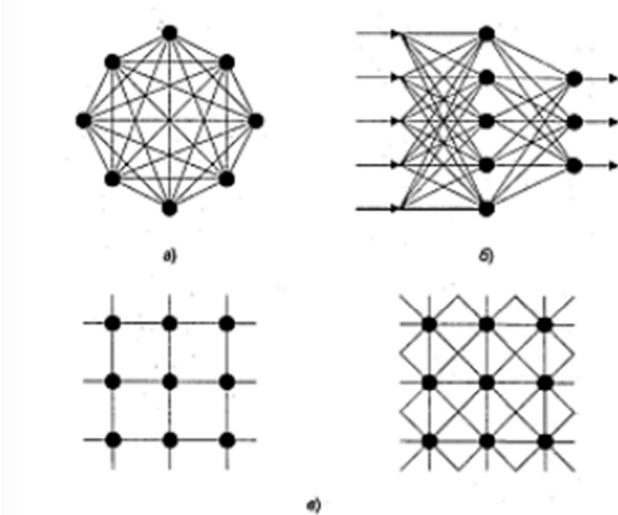


Рисунок 5 – Архитектуры нейронных сетей:

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

Биологический нейрон - является особой биологической клеткой, которая обрабатывает информацию. Он состоит из тела, или сомы, 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 (передатчика), который в конце разветвляется на волокна. На окончаниях этих волокон находятся специальные образования - синапсы, которые влияют на величину импульсов.

Искусственный нейрон

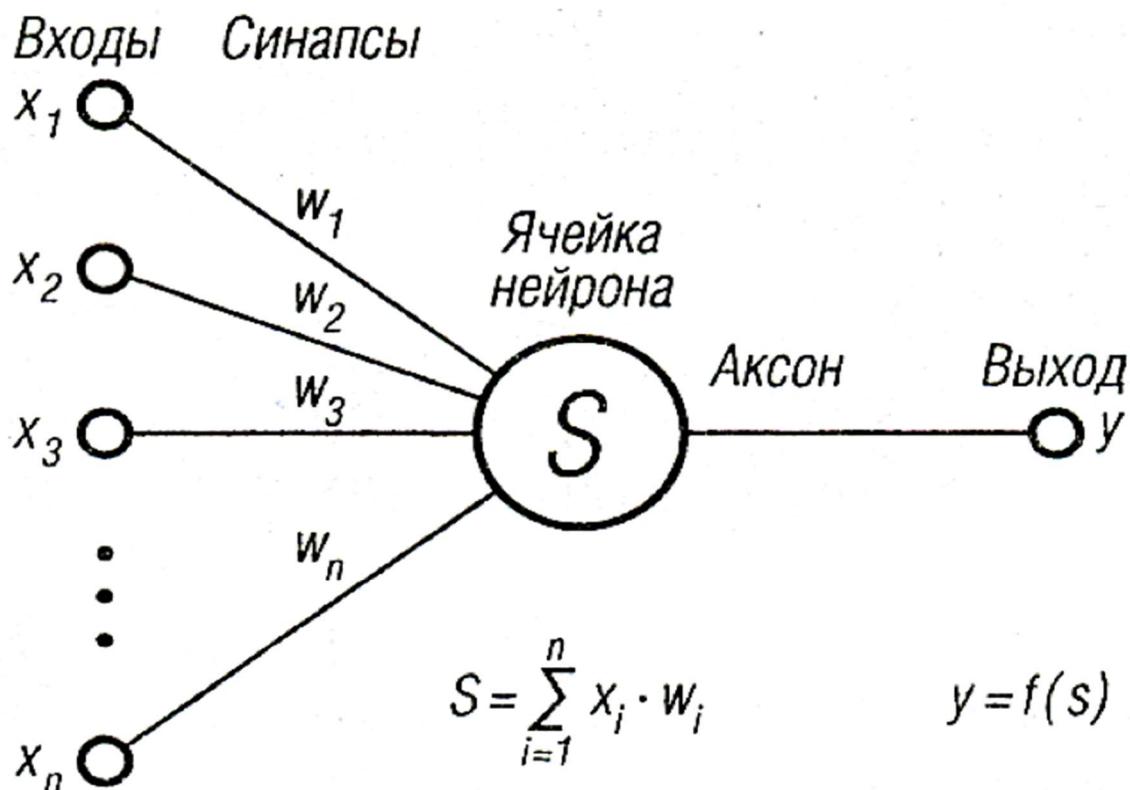
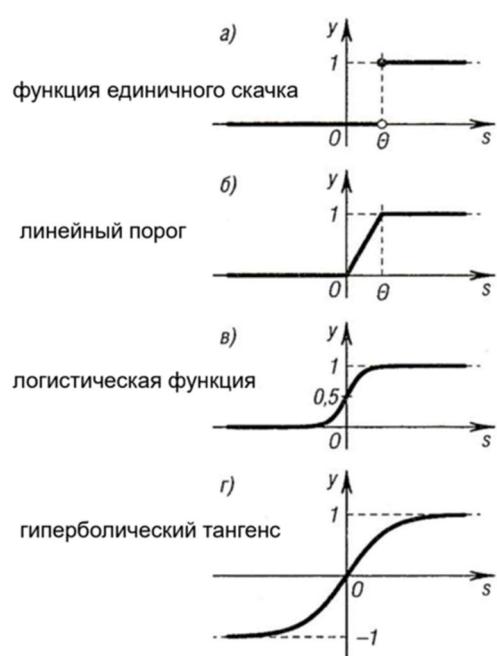


Рис. 2. Модель искусственного нейрона

Активационная функция - сигнал возбуждения или торможения.

Функции активации нейронов

Название	Формула	Область значений
Пороговая (функция единичного скачка)	$f(s) = \begin{cases} 0, & s < \Theta; \\ 1, & s \geq \Theta \end{cases}$	{0, 1}
Линейная	$f(s) = ks$	($-\infty, +\infty$)
Логистическая (сигмоидальная)	$f(s) = \frac{1}{1 + e^{-as}}$	(0, 1)
Гиперболический тангенс	$f(s) = \frac{e^{as} - e^{-as}}{e^{as} + e^{-as}}$	(-1, 1)
Линейная с насыщением (линейный порог)	$f(s) = \begin{cases} 0, & s < 0; \\ s/\Theta, & 0 \leq s < \Theta \\ 1, & s \geq \Theta \end{cases}$	[0, 1]



Классификация:

1. Однослойные ИНС
2. Многослойные ИНС

В искусственных и биологических сетях многие соединения могут отсутствовать. Многослойные сети могут привести к увеличению вычислительной мощности по сравнению с однослойной сетью лишь в том случае, если активационная функция между слоями нелинейна. Так как умножение матриц ассоциативно, то двухслойная линейная сеть эквивалентна одному слою с весовой матрицей, равной произведению двух весовых матриц. Следовательно, любая многослойная линейная сеть может быть заменена эквивалентной однослойной сетью. Различают алгоритмы обучения с учителем и без учителя.

31. Теорема Колмогорова. Алгоритм обучения персептрана. Линейная разделимость и персептронная представляемость.

Подавая на входы любые числа x_1, x_2, \dots, x_n , мы получим на выходе значение некоторой функции $y = F(x_1, x_2, \dots, x_n)$, которое является ответом (реакцией) сети. Ответ сети зависит как от **входного сигнала**, так и от **значений ее внутренних параметров** - весов нейронов. Точный вид этой функции:

$$F(x_1, x_2, \dots, x_n) = \sum_{i=1}^m v_i \sigma\left(\sum_{j=0}^n x_j w_{ij}\right),$$

где $\sigma(s) = \frac{1}{1+e^{-s}}$

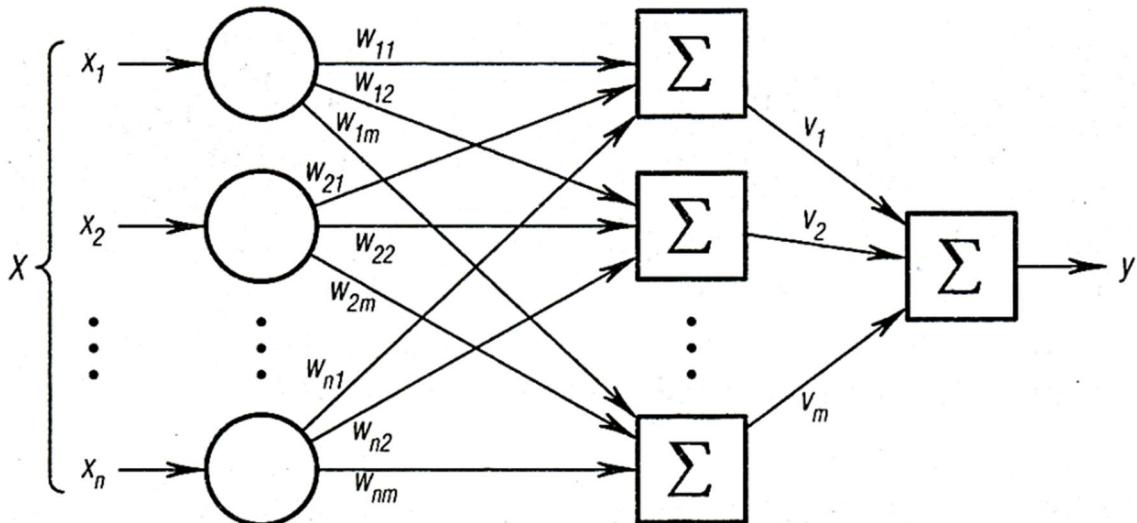


Рис. 6. Пример ИИНС

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

$$F(x_1, x_2, \dots, x_n) = \sum_{i=1}^{2n+1} g_i \left(\sum_{j=1}^n h_{ij}(x_j) \right)$$

$$x = (x_1, x_2, \dots, x_n), 0 \leq x_i \leq 1,$$

где g_i и h_{ij} - непрерывные функции, причем h_{ij} не зависят от функции F .

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

Персептроном, как правило, называют однослойную нейронную сеть, при том каждый персептронный нейрон в качестве активационной функции использует функцию единичного скачка (пороговую).

Обучение персептрана состоит в подстройке весовых коэффициентов w_i , где i

$i = 1, 2, \dots, n$. Обученный персептрон способен разделять требуемое множество образов на два класса. (К первому классу относятся входные образы, для которых на выходе персептрана получено нулевое значение, ко второму классу - образы, для которых получено единичное значение.)

Обучение персептрана — это обучение с учителем, т. е. должен существовать набор векторов (обучающая выборка). Будем называть персептран обученным на данной обучающей выборке, если при подаче на вход каждого вектора x на выходе всякий раз получается соответствующее значение y .

Алгоритм включает несколько шагов:

Шаг 0. Проинициализировать весовые коэффициенты w небольшими i , $i = 1, 2, \dots, n$ случайными значениями, например, из диапазона $[-0,3; 0,3]$.

Шаг 1. Подать на вход персептрана один из обучающих векторов X и вычислить его выход y .

Шаг 2. Если выход правильный, перейти на шаг 4. Иначе вычислить ошибку — разницу между верным и полученным значениями выхода.

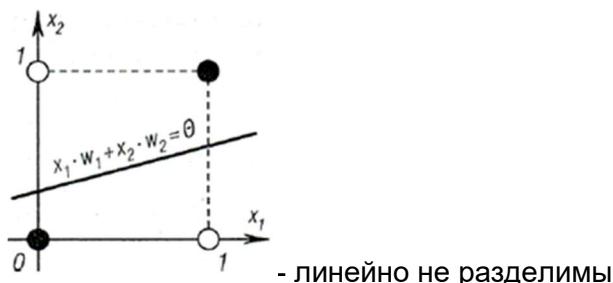
Шаг 3. Весовые коэффициенты модифицируются по следующей формуле:

$$w_{ij}^{t+1} = w_{ij}^t + v \cdot \delta \cdot x_i$$

Здесь t и $t + 1$ — номера соответственно текущей и следующей итераций; v — коэффициент скорости обучения ($0 < v \leq 1$); x_i — i -я компонента входного вектора X .

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

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



Персептран способен научиться всему, что он способен представлять. Понятие представляемости относится к способности персептрана (или другой сети) моделировать определенную функцию. Обучаемость же требует наличия систематической процедуры настройки весов сети для реализации этой функции.

32. Нейронная сеть обратного распространения. Алгоритм обучения сети обратного распространения

Нейронные сети обратного распространения — это современный инструмент поиска закономерностей, прогнозирования, качественного анализа. Такое название — сети обратного распространения — они получили из-за используемого алгоритма обучения, в котором ошибка распространяется от выходного слоя к

входному, т.е. в направлении, противоположном направлению распространения сигнала при нормальном функционировании сети.

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

$$E(W, V) = \frac{1}{2} \sum_{k=1}^p (y_k - d_k)^2,$$

где y_k — полученное реальное значение k -го выхода нейросети при подаче на нее одного из входных образов обучающей выборки; d_k — требуемое значение k -го выхода для этого образца.

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

$$w_{ij}^{N+1} = w_{ij}^N - \alpha \frac{\partial E}{\partial w_{ij}}, \quad v_{jk}^{N+1} = v_{jk}^N - \alpha \frac{\partial E}{\partial v_{jk}}$$

где α — параметр, определяющий скорость обучения.

В качестве активационной функции в сети обратного распространения обычно используется сигмоидальная функция

$$f(s) = \frac{1}{1+e^{-s}},$$

где s — взвешенная сумма входов нейрона. Эта функция удобна для вычислений в градиентном методе, так как имеет простую производную:

$$f'(s) = \frac{e^{-s}}{(1+e^{-s})^2} = f(s)(1-f(s)).$$

Функция ошибки в явном виде не содержит зависимости от весовых коэффициентов w_{ij} и v_{jk} .

После некоторых упрощений, вычислений и введения обозначений

$$\delta_k = \frac{\partial E}{\partial y_k} \cdot \frac{\partial y_k}{\partial s_k} = (y_k - d_k)y_k(1 - y_k),$$

Получим следующие выражения для производных:

$$\frac{\partial E}{\partial v_{jk}} = \delta_k y_j^c, \quad \frac{\partial E}{\partial w_{ij}} = \left(\sum_{k=1}^p \delta_k v_{jk} \right) y_j^c (1 - y_j^c) x_i.$$

Алгоритм обучения сети обратного распространения

Рассмотрим полный алгоритм обучения нейросети:

Шаг 1. Инициализация сети.

Весовым коэффициентам присваиваются малые случайные значения, например, из диапазона (-0,3; 0,3); задаются — параметр точности обучения, — параметр скорости обучения (как правило, $\alpha \approx 0,1$ и может еще уменьшаться в процессе обучения), N_{max} — максимально допустимое число итераций.

Шаг 2. Вычисление текущего выходного сигнала.

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

Шаг 3. Настройка синаптических весов.

Рассчитать изменение весов для выходного слоя нейронной сети по формулам

$$v_{jk}^{N+1} = v_{jk}^N - \alpha \frac{\partial E}{\partial v_{jk}}, \text{ где } \frac{\partial E}{\partial v_{jk}} = \delta_k y_j^c, \quad \delta_k = (y_k - d_k) y_k (1 - y_k),$$

Рассчитать изменение весов для скрытого слоя по формулам

$$w_{ij}^{N+1} = w_{ij}^N - \alpha \frac{\partial E}{\partial w_{ij}}, \quad \text{где } \frac{\partial E}{\partial w_{ij}} = \left(\sum_{k=1}^p \delta_k v_{jk} \right) y_j^c (1 - y_j^c) x_i.$$

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

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

Прототип - объект, наиболее типичный для своего класса. Один из самых простых подходов к классификации состоит в том, чтобы предположить существование определенного числа классов и произвольным образом выбрать координаты прототипов. Затем каждый вектор из набора данных связывается с ближайшим к нему прототипом, и новыми прототипами становятся центроиды всех векторов, связанных с исходным прототипом. В качестве меры близости двух векторов обычно выбирается евклидово расстояние.

На этих принципах основано функционирование сети Кохонена, обычно используемой для решения задач классификации. Данная сеть обучается **без учителя** на основе самоорганизации. По мере обучения векторы весов нейронов становятся прототипами классов- групп векторов обучающей выборки. На этапе решения информационных задач сеть относит новый предъявленный образ к одному из сформированных классов.

Сеть Кохонена состоит из одного слоя нейронов. Число входов каждого нейрона n равно размерности вектора параметров объекта. Количество нейронов t совпадает с требуемым числом классов, на которые нужно разбить объекты.

Обучение начинается с задания небольших случайных значений элементам весовой матрицы W . В дальнейшем происходит процесс самоорганизации, состоящий в модификации весов при предъявлении на вход векторов обучающей выборки.

Далее выбирается нейрон с номером для которого это расстояние минимально. На текущем шаге обучения N будут модифицироваться только веса нейронов из окрестности нейрона.

Первоначально в окрестности любого из нейронов находятся все нейроны сети, но с каждым шагом эта окрестность сужается. В конце этапа обучения подстраиваются только веса нейрона. Темп обучения с течением времени также уменьшается. Образы обучающей выборки предъявляются последовательно и каждый раз происходит подстройка весов.

Алгоритм обучения сети Кохонена

Шаг 1. Инициализация сети. Весовым коэффициентам сети присваиваются малые случайные значения. Задаются значения - начальный темп обучения и — максимальное расстояние между весовыми векторами (столбцами матрицы W).

Шаг 2. Предъявление сети нового входного сигнала X .

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

Шаг 4. Выбор нейрона с наименьшим расстоянием.

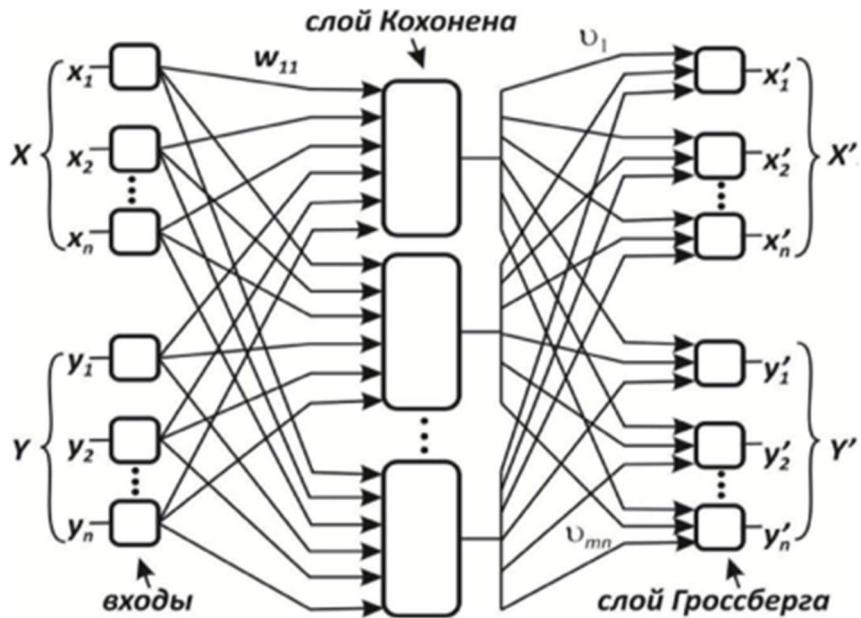
Шаг 5. Настройка весов нейрона и всех нейронов, находящихся от него на расстоянии, не превосходящем расстояния, вычисленным на 3 шаге

Шаг 6. Уменьшение значений темпа обучения и расстояния между векторами

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

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

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



Сеть встречного распространения состоит из двух слоев: слоя нейронов Кохонена и слоя нейронов Гроссбера. Слой Кохонена классифицирует входные векторы в группы схожих. Это достигается с помощью такой подстройки весов слоя Кохонена, что близкие входные векторы активируют один и тот же нейрон данного слоя. Затем слой Гроссбера дает требуемые выходы.

В режиме нормального функционирования предъявляются входные векторы X и Y , и обученная сеть дает на выходе векторы X' и Y' .

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

Шаг 2. Весовым коэффициентам сети $w_{ij}, v_{ji}, i=1, 2, \dots, 2n, j=1, 2, \dots, m$ присвоить малые случайные значения и произвести единичную нормировку матриц W , V по столбцам. Положить $\alpha_0 = 0,7, \beta_0 = 0,1$.

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

$$w_{ij}^{(N+1)} = w_{ij}^N + \alpha_N (z_i - w_{ij}^N), \text{ где } Z = (X, Y).$$

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

$v_{ki}^{(N+1)} = v_{ki}^N + \beta_N (z_i - v_{ki}^N)$ (здесь k - номер выигравшего нейрона).

Шаг 5. Уменьшить значения α_0 , β_0 .

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

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

Выходная звезда является нейроном, управляющим группой весов. Входные и выходные звезды могут быть взаимно соединены в сети любой сложности.

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

$$Y = \sum_{i=1}^n w_i x_i.$$

В процессе обучения веса корректируются следующим образом:

$$w_i^{(N+1)} = w_i^N + \alpha_N (x_i - w_i^N),$$

где w_i — весовой коэффициент входа x_i ; α_N — нормирующий коэффициент обучения, который имеет начальное значение 0,1 и постепенно уменьшается в процессе обучения.

Если входная звезда учится реагировать на определенный вход, то выходная звезда обучается выдавать требуемый целевой выход.

Чтобы обучить нейрон выходной звезды, его веса настраиваются в соответствии с требуемым целевым вектором Y .

Формула коррекции весов имеет следующий вид:

$$w_i^{(N+1)} = w_i^N + \beta_N (y_i - w_i^N),$$

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

35. Двухслойная сеть встречного распространения. Алгоритм обучения сети встречного распространения.

Двухслойная сеть встречного распространения

Сеть встречного распространения состоит из двух слоев: слоя нейронов Кохонена и слоя нейронов Гроссберга. Автор сети Р. Хехт-Нильсен (R. Hecht-Nielsen) объединил эти две архитектуры, в результате чего сеть приобрела свойства, которых не было у каждой из них в отдельности.

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

На рис. 4 показана сеть встречного распространения.

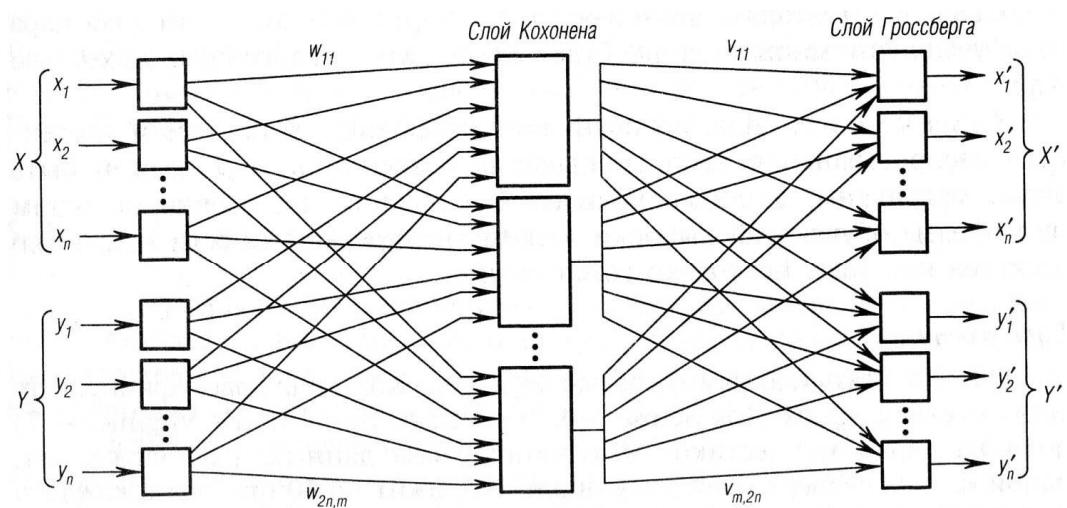


Рис.4. Двухслойная сеть встречного распространения

В режиме нормального функционирования предъявляются входные векторы X и Y , и обученная сеть дает на выходе векторы x' и y' , являющиеся аппроксимациями соответственно для X и Y . Векторы x' и y' предполагаются здесь нормированными векторами единичной длины, следовательно, порождаемые на выходе векторы также должны быть нормированными.

В процессе обучения векторы X и Y подаются одновременно и как входные векторы сети, и как желаемые выходные сигналы. В результате получается отображение, при котором предъявление пары входных векторов порождает их копии на выходе.

Это не было бы интересно, если не учитывать способность этой сети к обобщению. Благодаря обобщению предъявление только вектора X (с вектором

$= 0$) порождает как выходы $'$ так и выходы $'$. Если $-$ функция, отображающая в $'$, то сеть аппроксимирует ее. Кроме того, если функция обратима (если функция $y = f(x)$ такова, что для любого ее значения y_0 уравнение $f(x) = y_0$ имеет относительно x единственный корень, то говорят, что функция f обратима), то предъявление только вектора $($ при $= 0$) порождает выходы $'$.

Уникальная способность порождать функцию и обратную к ней делает сеть встречного распространения полезной в ряде приложений. Например, в задаче аппроксимации многомерной векторной функции сеть обучается на известных значениях этой функции.

Алгоритм обучения двухслойной сети встречного распространения

Шаг 1. Произвести единичную нормировку всех векторов $,$ обучающего множества.

Шаг 2. Весовым коэффициентам сети $, , = 1, 2, \dots, 2, = 1, 2, \dots,$ присвоить малые случайные значения и произвести единичную нормировку матриц $,$ по столбцам. Положить $\alpha_0 = 0,7, \beta_0 = 0,1.$

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

$$^{+1} = + (-), \text{ где } = (,).$$

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

$$^{+1} = + (-) \text{ (здесь } - \text{ номер выигравшего нейрона).}$$

Шаг 5. Уменьшить значения $\alpha_0, \beta_0.$

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

Замечание. Для улучшения обобщающих свойств сети встречного распространения темп уменьшения значений α и β должен быть очень маленьким, а общее количество итераций достаточно большим (все образы

обучающей выборки желательно предъявить сети несколько десятков или даже несколько сотен раз).

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

Стохастические методы обучения

Искусственная нейронная сеть обучается посредством некоторого процесса, модифицирующего ее веса. Если обучение успешно, то предъявление сети множества входных сигналов приводит к появлению желаемого множества выходных сигналов. **Стохастические методы обучения** выполняют псевдослучайные изменения величин весов, сохраняя те изменения, которые ведут к улучшениям.

Локальные минимумы мешают всем алгоритмам обучения, основанным на поиске минимума функции ошибки, включая сети обратного распространения, и представляют серьезную и широко распространенную проблему.

Стохастические методы позволяют решить эту проблему. Стратегия коррекции весов, вынуждающая веса принимать значение глобального оптимума, возможна.

В качестве объясняющей аналогии предположим, что на рис. 1 изображен шарик на поверхности в коробке. Если коробку сильно потрясти в горизонтальном направлении, то шарик будет быстро перекатываться от одного края к другому.

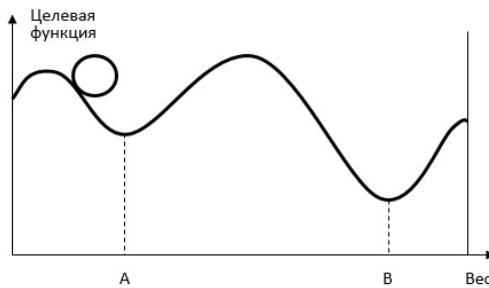


Рис. 1. Проблема локальных минимумов

Нигде не задерживаясь, в каждый момент шарик будет с равной вероятностью находиться в любой точке поверхности. Если постепенно уменьшать силу встряхивания, то будет достигнуто условие, при котором шарик

будет на короткое время «застревать» в точке В. При еще более слабом встряхивании шарик будет на короткое время останавливаться как в точке А, так и в точке В. При непрерывном уменьшении силы встряхивания будет достигнута критическая точка, когда сила встряхивания достаточна для перемещения шарика из точки А в точку В, но недостаточна для того, чтобы шарик мог выбраться из В в А. Таким образом, окончательно шарик остановится в точке глобального минимума, когда амплитуда встряхивания уменьшится до нуля.

Обучение Больцмана

Искусственные нейронные сети могут обучаться, по существу, тем же самым образом посредством случайной коррекции весов. Вначале делаются большие случайные коррекции с сохранением только тех изменений весов, которые уменьшают целевую функцию. Затем средний размер шага постепенно уменьшается, и глобальный минимум в конце концов достигается.

Такая процедура напоминает отжиг металла, поэтому для ее описания часто используют термин **«имитация отжига»**. В металле, нагретом до температуры, превышающей его точку плавления, атомы находятся в сильном беспорядочном движении. Как и во всех физических системах, атомы стремятся к состоянию минимума энергии, но при высоких температурах энергия атомных движений препятствует этому. В процессе постепенного охлаждения металла возникают все более низкоэнергетические состояния, пока в конце концов не будет достигнуто наиболее низкое из возможных состояний, глобальный минимум. В процессе отжига распределение энергетических уровней описывается следующим соотношением:

$$() = \frac{1}{e^{-\frac{E}{kT}}}$$

где $()$ - вероятность того, что система находится в состоянии с энергией E ,
- постоянная Больцмана; k - температура по шкале Кельвина.

При высоких температурах вероятность $()$ приближается к единице для всех энергетических состояний. Таким образом, высокоэнергетическое состояние почти столь же вероятно, как и низкоэнергетическое. По мере уменьшения температуры вероятность высокоэнергетических состояний уменьшается по сравнению с низкоэнергетическими. При приближении температуры к нулю становится весьма маловероятным, чтобы система находилась в высокоэнергетическом состоянии.

Этот стохастический метод непосредственно применим к обучению искусственных нейронных сетей и относится к классу алгоритмов обучения с учителем.

Алгоритм обучения Больцмана

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

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

Шаг 3. Придать случайное изменение $\Delta \theta$ выбранному весу θ и пересчитать выход сети и изменение функции ошибки в соответствии со сделанным изменением веса.

Шаг 4. Если функция ошибки уменьшилась, то сохранить изменение веса. Если изменение веса приводит к увеличению функции ошибки, то вероятность сохранения этого изменения вычисляется с помощью распределения Больцмана: $P(\Delta \theta) = e^{-\frac{\Delta E}{kT}}$. Выбирается случайное число из равномерного распределения от нуля до единицы. Если вероятность $(\Delta \theta)$ больше, чем x , то изменение сохраняется, в противном случае величина веса возвращается к предыдущему значению.

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

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

Замечание 1. На шаге 4 система может делать случайный шаг в направлении, портящем функцию ошибки, позволяя ей тем самым вырываться из локальных минимумов, где любой малый шаг увеличивает целевую функцию.

Замечание 2. В работах, посвященных большинству обучению, показано, что для достижения сходимости к глобальному минимуму скорость уменьшения искусственной температуры должна подчиняться закону

$$= \frac{\theta}{(1 +)}$$

где θ - номер итерации обучения. Этот результат предсказывает очень медленную сходимость процесса обучения, что является существенным недостатком метода.

Обучение Коши

В этом методе распределение Больцмана заменяется на распределение Коши. Распределение Коши имеет, как показано на рис. 2, более высокую вероятность больших шагов. Дисперсия распределения Коши бесконечна.

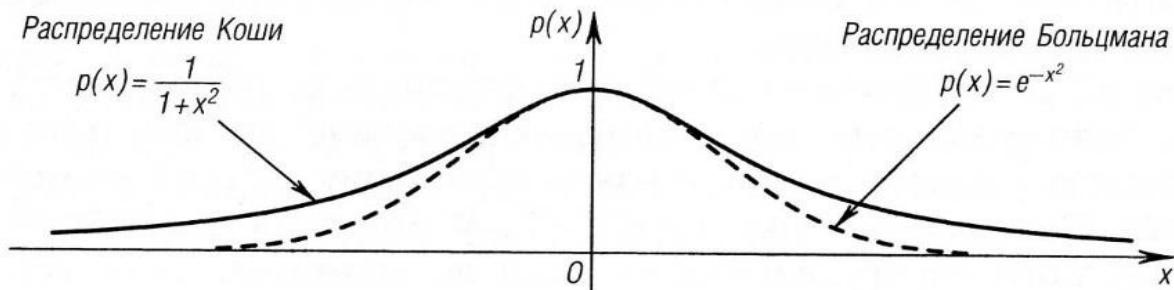


Рис. 2. Распределение Коши и распределение Больцмана

С помощью такого простого изменения максимальная скорость уменьшения температуры становится обратно пропорциональной линейной величине, а не логарифму, как для алгоритма обучения Больцмана. Эта связь может быть выражена следующим образом: $= \frac{\theta}{1 + } .$

Таким образом, время обучения резко уменьшается. Распределение Коши имеет вид

$$(\Delta) = \frac{1}{\pi} \frac{1}{\Delta^2 + \Delta^2}$$

где (Δ) - вероятность принять изменение веса Δ .

Несмотря на улучшение скорости обучения, даваемое распределением Коши по сравнению с распределением Больцмана, время сходимости все еще может в 100 раз превышать время для алгоритма обратного распространения.

Комбинирование обратного распространения с обучением Коши.

Коррекция весов в комбинированном алгоритме, использующем обратное распространение и обучение Коши, состоит из двух компонент: компоненты, вычисляемой с использованием алгоритма обратного распространения, и случайной компоненты, определяемой распределением Коши.

Эти компоненты вычисляются для каждого веса, и их сумма является величиной, на которую изменяется вес. Как и в алгоритме Коши, после вычисления изменения веса вычисляется целевая функция. Если имеет место улучшение, изменение сохраняется. В противном случае оно сохраняется с вероятностью, определяемой распределением Коши.

Коррекция веса вычисляется с использованием представленных ранее уравнений для каждого из алгоритмов:

$$w^{+1} = w - \eta \Delta + (1 - \eta)w, \quad ,$$

где η - коэффициент, управляющий относительными величинами обучения Коши и обратного распространения в компонентах весового шага.

Если η приравнивается нулю, метод становится обучением Коши. Если η приравнивается единице, метод становится алгоритмом обратного распространения.

Комбинированная сеть, использующая обратное распространение и обучение Коши, обучается быстрее, чем каждый из алгоритмов в отдельности. Сходимость к глобальному минимуму гарантируется алгоритмом Коши, и во многих экспериментах по обучению сеть практически не попадала в локальные минимумы.

37. Сети с обратными связями. Сеть Хопфилда.

Сети с обратными связями

После обучения на этапе функционирования сети каждый нейрон выполняет свою функцию - передачу выходного сигнала один раз. В общем случае может быть рассмотрена нейронная сеть, содержащая произвольные **обратные связи**, т. е. пути, передающие сигналы от выходов к входам. Отклик таких сетей является динамическим, т. е. после подачи нового входа вычисляется выход и, передаваясь по обратной связи, модифицирует вход. Затем выход повторно вычисляется, и процесс повторяется снова и снова. Для *устойчивой* сети последовательные итерации приводят к все меньшим изменениям выхода, и в результате выход становится постоянным. Для многих сетей

процесс никогда не заканчивается, такие сети называются **неустойчивыми**. Неустойчивые сети обладают интересными свойствами и могут рассматриваться в качестве примера хаотических систем, но для большинства практических приложений используются сети, которые дают постоянный выход.

Сеть Хопфилда

Рассмотрим однослойную сеть с обратными связями, состоящую из n входов и n нейронов (см. рис. ниже). Каждый вход связан со всеми нейронами. Так как выходы сети заново подаются на входы, то y_i - это значение i -го выхода, который на следующем этапе функционирования сети становится i -м входом.

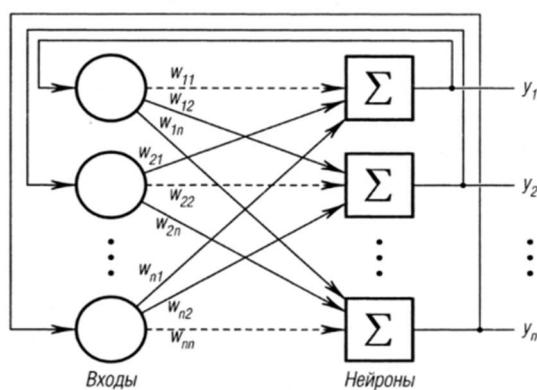


Рис: Модель сети Хопфилда

38. Правило обучения Хебба. Ортогонализация образов

Пусть задана обучающая выборка образов X_k , $k=1, 2, \dots, K$. Требуется построить матрицу связей W такую, что соответствующая нейронная сеть будет иметь в качестве стационарных состояний образы обучающей выборки.

В случае одного обучающего образа , правило Хебба приводит к матрице

Два различных запоминаемых векторных образа сети называются ортогональными, если их скалярное произведение равно нулю: . Если все запоминаемые образы сети попарно ортогональны, емкость памяти сети

Хопфилда увеличивается до , т. е. сеть может запомнить количество образов, не превосходящее число нейронов в ней. На этом свойстве основано улучшение правила Хебба: перед запоминанием в нейронной сети исходные образы следует ортогонализовать. Процедура расчета весовых коэффициентов в этом случае имеет следующий вид:

Шаг 1. Вычисляются элементы матрицы :

Шаг 2. Определяется матрица , обратная к матрице .

Шаг 3. Задаются весовые коэффициенты сети Хопфилда:

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

39. Двунаправленная ассоциативная память.

Сеть Хопфилда реализует так называемую автоассоциативную память. Это означает, что образ может быть завершен или исправлен, но не может быть ассоциирован с другим образом. Двунаправленная ассоциативная память (ДАП), разработанная в 1988 году Бертом Коско (B. Kosko), является гетероассоциативной: она сохраняет пары образов и выдает второй образец пары, когда ассоциированный с ним первый образец подается на вход сети. Как и сеть Хопфилда, сеть ДАП способна к обобщению, вырабатывая правильные реакции, несмотря на искаженные входы. Сеть ДАП (рис. 7) содержит два слоя нейронов. Элементы весовой матрицы отражают связь между -м нейроном первого слоя и -м нейроном второго слоя, $= 1, 2, \dots, , = 1, 2, \dots, .$

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

Этот процесс может быть выражен следующим образом:

$$\theta_{\alpha}^{t+1} = \theta_{\alpha}^t \left(\sum_{\beta=1}^{\bar{m}} \theta_{\beta}^{t+1} \theta_{\beta\alpha} \right), \quad \alpha = 1, 2, \dots, \bar{m},$$

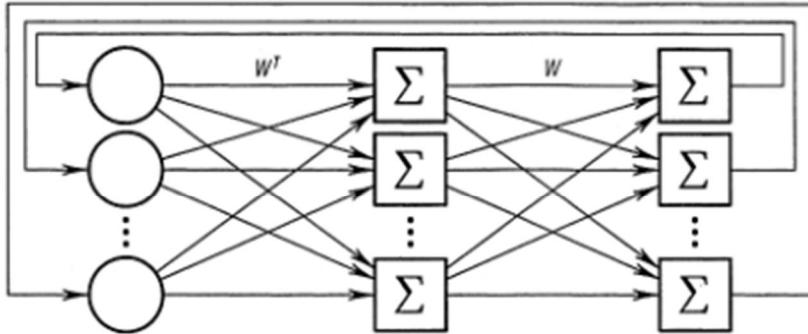


Рис. 7. Структура сети ДАП

$$\theta_{\alpha}^{t+1} = \theta_{\alpha}^t \left(\sum_{\beta=1}^{\bar{m}} \theta_{\beta}^{t+1} \theta_{\beta\alpha} \right), \quad \alpha = 1, 2, \dots, \bar{m},$$

где

$$\theta_{\beta}(\theta) = \{-1, \theta < \theta_{\beta}; 1, \theta > \theta_{\beta}; \theta^{(0)}(\theta), \theta = \theta_{\beta}\}.$$

$\theta^{(0)}(\theta)$ значение функции активации данного нейрона на предыдущем шаге.

Пусть задана обучающая выборка ассоциированных образов $(\theta^{\alpha}, \theta^{\beta})$, $\alpha = 1, 2, \dots, \bar{m}$. Весовая матрица сети ДАП вычисляется как сумма произведений всех векторных пар обучающего набора:

$$\theta_{\beta\alpha} = \sum_{\theta=1}^{\bar{n}} \theta_{\beta} \theta_{\alpha}^{\theta}, \quad \alpha = 1, 2, \dots, \bar{m}, \quad \theta = 1, 2, \dots, \bar{n}.$$

В отличие от сети Хопфилда, весовая матрица в сети ДАП не квадратная, что во многих случаях позволяет оптимизировать вычислительные затраты, необходимые для функционирования сети. Если рассмотреть пример с запоминанием букв А, В, С, то сеть Хопфилда в этом случае имела бы $10 \times 7 = 70$ входов и требовала для своей работы хранения весовой матрицы размером 70×70 , содержащей 4900 элементов. Ассоциируем с каждым из входных образов сети двухбитовый вектор: символ А будет связан с вектором $(1, -1, -1)$, символ В с вектором $(-1, 1, -1)$, символ С с вектором $(-1, -1, 1)$. Таким образом, например, при подаче на вход искаженной версии буквы А, сеть после стабилизации должна выдавать образ $(1, -1, -1)$. Так как ассоциированные пары заранее известны, это приведет к правильному распознаванию зашумленного входа. Но для работы такой сети требуется хранение всего $70 \times 3 = 210$ элементов весовой матрицы.

Основным недостатком сети ДАП, как и сети Хопфилда, является небольшая емкость памяти. Так, число запоминаемых ассоциаций не может превышать числа нейронов в меньшем слое. Если все пороговые значения θ_{α} нулевые, то оценка еще ухудшается: размер запоминаемой выборки не должен превосходить $\frac{1}{2\bar{n}\bar{m}_2\bar{m}}$, где \bar{n} —

число нейронов в меньшем слое. Если этот лимит превышен, сеть начинает вырабатывать неверные выходные сигналы, воспроизводя ассоциации, которым не обучена.

40. Адаптивная резонансная теория

Серьезная проблема для нейронных сетей - правильное соотношение стабильности и *пластичности* при запоминании образов. Существуют наборы эталонов (даже состоящие всего из 4-х векторов), которые при циклическом предъявлении в обучении дают никогда не сходящиеся наборы параметров сети. Предъявление всего одного нового образа в обучающем множестве часто приводит к долгому *переобучению*. Если сеть работает в реальном времени, например, обрабатывает сенсорную информацию, то обучающее множество может все время меняться. Для большинства моделей нейронных сетей это приводит к отсутствию обучения вообще.

Человеческая память, напротив, эффективно хранит и корректирует запоминаемые образы. Ни предъявление нового образа, ни изменение старых не приводит к уничтожению памяти или невозможности запоминания. Даже удаление части нервной ткани чаще всего не прерывает работу сети и не стирает запомненные образы, а лишь делает их менее четкими.

Сеть *APT* - попытка приблизить механизм запоминания образов в искусственных НС к биологическому. Результатом работы *APT* является устойчивый набор запомненных образов и возможность выборки "похожего" вектора по произвольному предъявленному на входе вектору. Важное качество *APT* - динамическое запоминание новых образов без полного *переобучения* и отсутствие потерь уже запомненных образов при предъявлении новых.

Стандартная AP система представляет собой модель обучения без учителя. Как правило она состоит из составленных из нейронов поля сравнения и поля распознавания, а также параметра бдительности и модуля сброса. На вход поля сравнения подается вектор чисел, для которого определяется соответствующий нейрон в поле распознавания, то есть тот, чьи веса больше всего похожи на входной вектор. Каждый нейрон поля распознавания тормозит другие нейроны из этого поля (сила воздействия пропорциональна степени соответствия).

После классификации входа модуль сброса сравнивает степень соответствия при распознании с параметром бдительности. В случае если порог преодолен, происходит обучение: веса победившего нейрона подгоняются к значениям входного вектора. Если же порог не был преодолен, то победивший нейрон подавляется и запускается процедура поиска. В ходе этой процедуры распознающие нейроны отключаются один за другим с помощью функции сброса пока порог бдительности не будет преодолен. На каждом цикле поиска выбирается наиболее активный распознающий нейрон, и отключается, в случае если активация не достигает порога бдительности. Параметр бдительности имеет значительное влияние на систему: высокие значения создают высоко детализированную память (множество мелких категорий), тогда как небольшие значения создают более общие образы (меньшее количество более крупных категорий).

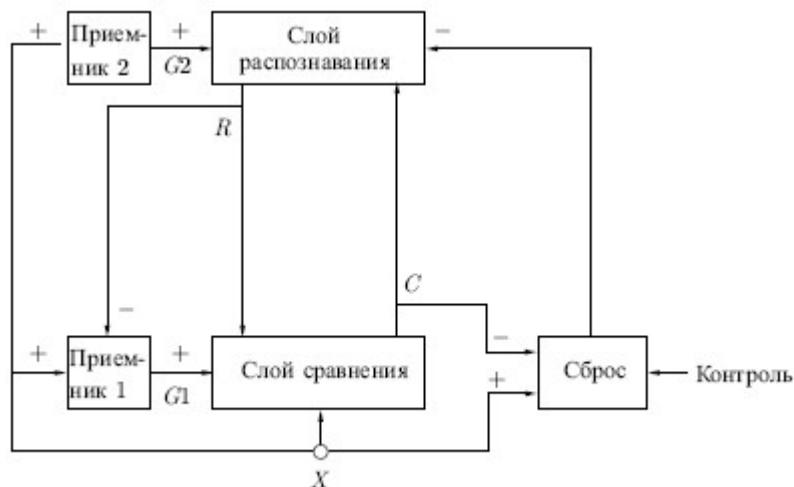
Этапы классификации APT:

Этап распознавания — входной вектор сравнивается с классификацией, представленной в каждом узле выходного слоя. Выход нейрона становится «1», если он

наилучшим образом соответствует применяемой классификации, в противном случае он становится «0».

Фаза сравнения. На этом этапе выполняется сравнение входного вектора с вектором слоя сравнения. Условием для сброса является то, что степень сходства будет меньше, чем параметр бдительности.

Фаза поиска — На этом этапе сеть будет искать сброс, а также соответствие, выполненное на вышеуказанных этапах. Следовательно, если бы не было сброса и совпадение было бы достаточно хорошим, то классификация окончена. В противном случае процесс будет повторен, и для сохранения правильного соответствия необходимо отправить другой сохраненный шаблон.



41. Методы оптимизации комбинаторных задач различной степени сложности.

Генетические алгоритмы.

Методы оптимизации комбинаторных задач различной степени сложности

В настоящее время не существует метода оптимизации, который позволил бы решить любую задачу (был универсальным) и при этом однозначно определен как лучший среди других методов по точности решения.

По степени приближения к точному решению, а также по характеру пространства поиска задачи могут быть разделены на следующие категории.

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

Общие задачи без ограничений - имеют нелинейное и неограниченное пространство поиска. Методы оптимизации для таких задач обычно полагаются на

правильность аналитической формулировки целевой функции. Оптимизация функции без ограничений заключается в максимизации или минимизации некоторой функции (x_1, \dots, x_n) .

Общие задачи с ограничениями - могут быть сформулированы как задачи минимизации функции (x_1, \dots, x_n) при следующих ограничениях:

$$(g_1(x_1, \dots, x_n) \geq 0 \text{ для } 1 \leq i \leq m, \quad h_i(x_1, \dots, x_n) = 0 \text{ для } 1 \leq i \leq l).$$

Обычно задачи с ограничениями могут быть сведены к задачам без ограничений с помощью метода штрафов.

Если пространство поиска содержит конечное число точек, то наиболее точное решение может быть уверенно получено методом полного перебора. Этот метод имеет один очевидный недостаток - сложность вычислений, а следовательно, время, затрачиваемое на нахождение оптимального решения, существенно зависит от размерности пространства поиска. Метод перебора может быть достаточно эффективным только в небольшом пространстве поиска.

Градиентные методы, являющиеся основой линейного и нелинейного, динамического программирования, а также численного анализа, более универсальны, но менее точны. При этом усложнение ландшафта пространства поиска приводит к снижению эффективности таких методов. Методы градиента не гарантируют получение единственного оптимального решения, за исключением случая, когда пространство отображения является выпуклым и не допускает появления второстепенных вершин, плато и т. д.

С другой стороны, **эвристические методы**, к которым относятся **генетические алгоритмы (ГА)**, являются наиболее универсальными, поэтому не гарантируют нахождения глобального оптимума, являющегося единственным решением задачи.

Характеристикой задачи и, соответственно, основой для классификации методов оптимизации является также сложность задачи. По степени сложности однозначно выделяются следующие задачи.

Линейные задачи - сложность которых определяется как (n) , где n — размерность входных данных задачи.

Полиномиальные задачи (n) - для них известен алгоритм, сложность которого составляет полином заданной, постоянной и не зависящей от размерности входной величины степени.

Экспоненциальные задачи - сложность которых не менее порядка (e^n) , где e - константа или полином от n .

ГА являются стохастическим эвристическим методом, в котором вероятность выбора состояния $(t+1)$ зависит от состояния (t) и косвенно от предыдущих

состояний. Стохастические методы позволяют решать широкий класс таких задач, поскольку не требуют жесткой формализации. Следует отметить, что стохастические методы оптимизации используются для решения NP-сложных комбинаторных задач, т. е. таких задач, к которым сводима любая задача из класса NP. При этом NP-сложные задачи не обязательно относятся к классу NP.

Каждый из стохастических и эвристических методов имеет свои достоинства и недостатки, обусловленные формулировкой и размерностью решаемой задачи. **При этом математически доказано, что для комбинаторных задач оптимизации средняя эффективность всех алгоритмов для всех возможных задач одинакова.** На рис. 1 приведена классификация эвристических и стохастических алгоритмов.

Полученные в результате решения большого количества задач результаты, усредненные по 10 запускам, доказывают справедливость утверждения о сравнимости эффективности всех перечисленных алгоритмов поиска глобального оптимума. Вместе с тем результаты, полученные при одном запуске, говорят о наибольшей эффективности двух методов поиска - ГА и поиска с учетом запретов.

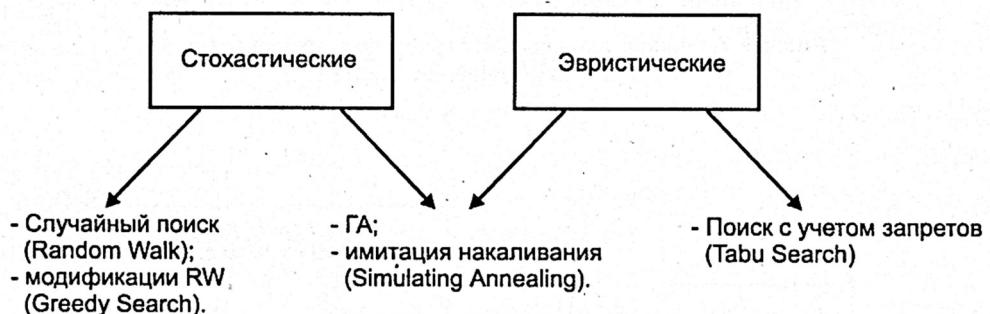


Рис. 1. Классификация эвристических и стохастических алгоритмов

Стохастические методы позволяют решать широкий класс таких задач, поскольку не требуют жесткой формализации.

Особенностью ГА является работа с пространством поиска с помощью комбинирования решений, а поиска с учетом запретов - использование памяти состояний.

Анализ результатов использования ГА позволяет выделить следующие условия, при выполнении которых задача решается эффективно:

- **Q большое пространство поиска**, ландшафт которого является негладким (содержит несколько экстремумов);
- **Q сложность формализации оценки качества решения функцией степени пригодности;**
- **Q многокритериальность поиска;**

- **Q поиск приемлемого решения по заданным критериям** в отличие от поиска единственного оптимального.

Генетические алгоритмы

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

Перечисленные далее причины коммерческого успеха инструментальных средств в области искусственного интеллекта могут рассматриваться как общие требования к разработке систем анализа данных, используемых ГА:

интегрированность - разработка ИС, легко интегрирующихся с другими информационными технологиями и средствами;

открытость и переносимость - разработка ИС в соответствии со стандартами, обеспечивающими возможность исполнения в разнородном программно-аппаратном окружении, и переносимость на другие платформы без перепрограммирования;

использование языков традиционного программирования - переход к ИС, реализованным на языках традиционного программирования (C, C++ и т. д.), что упрощает обеспечение интегрированности, снижает требования приложений к быстродействию ЭВМ и к объемам оперативной памяти;

архитектура "клиент-сервер" - разработка ИС, поддерживающих распределенные вычисления в архитектуре "клиент-сервер", что позволяет снизить стоимость оборудования, используемого в приложениях, децентрализовать приложения и повысить их производительность.

Перечисленные требования обусловлены необходимостью создания интегрированных приложений, т. е. приложений, объединяющих в рамках единого комплекса традиционные программные системы с системами искусственного интеллекта и ГА в частности.

Интеграция ГА и нейронных сетей позволяет решать проблемы поиска оптимальных значений весов входов нейронов, а **интеграция ГА и нечеткой логики** позволяет оптимизировать систему производственных правил, которые могут быть использованы для управления операторами ГА (дву направленная интеграция).

Одним из наиболее востребованных приложений ГА в области Data Mining является поиск наиболее оптимальной модели (поиск алгоритма, соответствующего специфике конкретной области).

Задача формализуется таким образом, чтобы её решение могло быть закодировано в виде вектора («генотипа») генов, где каждый ген может быть битом, числом или неким другим объектом.

В классических реализациях генетического алгоритма (ГА) предполагается, что генотип имеет фиксированную длину.

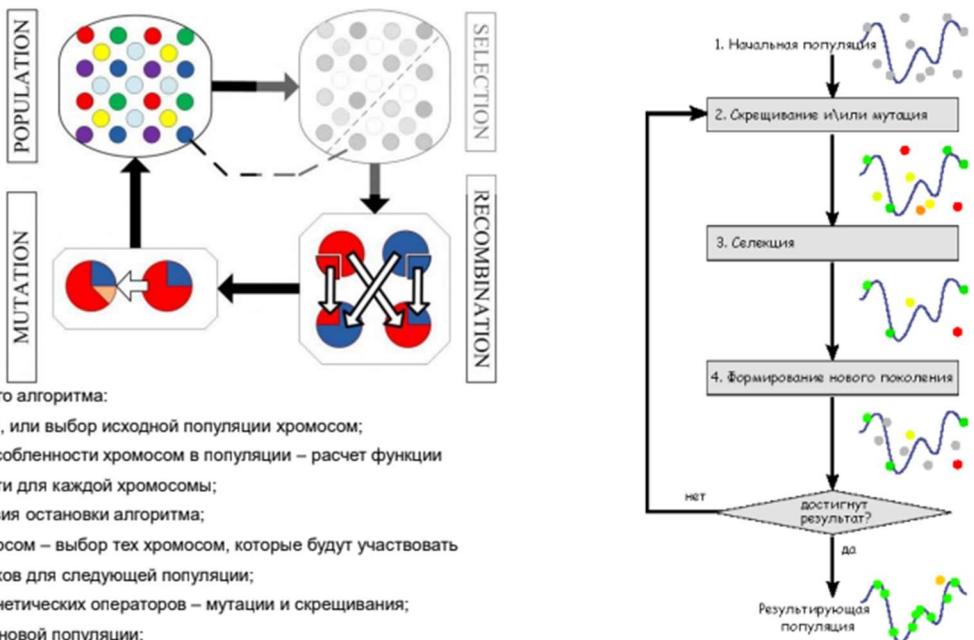
Однако существуют вариации ГА, свободные от этого ограничения.

Обычно **случайным**, образом создаётся множество генотипов начальной популяции. Они оцениваются с использованием «функции приспособленности», в результате чего с каждым генотипом ассоциируется определённое значение («приспособленность»), которое определяет насколько хорошо фенотип, им описываемый, решает поставленную задачу.

Из полученного множества решений («поколения») с учётом значения «приспособленности» выбираются решения (лучшие особи имеют большую вероятность быть выбранными), к которым применяются «генетические операторы» («скрещивание» — crossover и «мутация» — mutation), результат - получение новых решений. Для них также вычисляется значение приспособленности, и затем производится отбор («селекция») лучших решений в следующее поколение.

Набор действий 2, 3 повторяется итеративно, так моделируется «эволюционный процесс», продолжающийся несколько жизненных циклов (поколений), пока не будет выполнен критерий остановки алгоритма:

- нахождение глобального, либо субоптимального решения;
- исчерпание числа поколений, отпущеных на эволюцию;
- исчерпание времени, отпущеного на эволюцию;
- исчерпание времени, на улучшение предыдущего результата;



42. Базовый генетический алгоритм. Последовательные модификации базового генетического алгоритма.

Эволюционные алгоритмы, моделирующие процессы естественной эволюции, были предложены уже в 60-х годах прошлого века. Их особенностью является то, что они опираются на естественную эволюцию в природе, используя основные ее механизмы (отбор или селекцию, скрещивание и мутацию). Известны утверждения: "алгоритм является хорошим оптимизационным методом, потому что его принцип используется в природе", и наоборот: "алгоритм не может быть хорошим оптимизационным методом, потому что вы не находите его в природе".

Моделирование процесса естественной эволюции для эффективной оптимизации является первостепенной задачей теоретических и практических исследований в области эволюционных алгоритмов.

В 70-х годах прошлого века независимо друг от друга появились **два различных направления в области эволюционных алгоритмов: генетический алгоритм Холланда и эволюционные стратегии (ЭС) Речёнберга и Швефела**. Эволюционные стратегии используют операторы селекции и мутации, а если использовать биологические термины, то эволюционная стратегия моделирует естественную эволюцию с помощью непарной репродукции.

Эволюционные стратегии ($\mu + \lambda$).

Шаг 1. Создание первоначальной популяции размера λ .
 Шаг 2. Вычисление пригодности $F(x_i)$ $i = 1, \dots, \lambda$.
 Шаг 3. Селекция (отбор) $\mu < \lambda$ лучших индивидов.
 Шаг 4. Создание $\lambda - \mu$ потомков каждого из μ индивидов с небольшими вариациями.
 Шаг 5. Возврат к шагу 2.

Рис. П2.4. Разновидность эволюционных алгоритмов — эволюционные стратегии

Алгоритмы поиска, которые моделируют **парную репродукцию**, называются генетическими алгоритмами. Парная репродукция характеризуется рекомбинацией двух родительских строк для создания потомков. Эта рекомбинация называется **скрещиванием**.

Предпочтение разных генетических операторов в ЭС и ГА определило отношение к используемому размеру популяции. Так, Холланд подчеркивал важность рекомбинации в больших популяциях, в то время как Реченберг и Швефель, главным образом, рассматривали мутацию в очень маленьких популяциях.

При работе с ГА решения задачи должны быть представлены в виде строки с бинарной, целочисленной или вещественной кодировкой. Способ кодирования предполагает работу со строками фиксированной или переменной длины, возможна также и контекстно-зависимая кодировка. Основным отличием генетических программ (ГП) от ГА является работа с деревьями решений. При этом в ГП отсутствует необходимость в генетическом представлении задачи. Такая схема представления вносит гибкость в описание структур данных, однако решения могут стать очень объемными без улучшения производительности. Это справедливо и для эволюционных программ (ЭП). На рисунке ниже приведен базовый или стандартный ГА (СГА), предложенный Холландом, который явился основой для различных модификаций.

СГА.

Шаг 0. Определение генетического представления задачи.
 Шаг 1. Создание первоначальной популяции индивидов $P(0) = x_1^0, \dots, x_N^0$, $t = 0$.
 Шаг 2. Вычисление средней пригодности $f_{cp}(t) = \sum_i^N f(x_i)/N$. Вычисление нормализованного значения степени пригодности $f(x_i)/f_{cp}(t)$ для каждого индивида.
 Шаг 3. Назначение каждому индивиду x_i вероятности $p(x_i, t)$ пропорционально нормализованной пригодности. Выбор N векторов из $P(t)$, используя полученное распределение. Это дает набор отобранных родителей.
 Шаг 4. Формирование случайным образом из данного набора $N/2$ пар. Применение к каждой паре скрещивания, а также других генетических операторов, таких как мутация, для формирования новой популяции $P(t+1)$.
 Шаг 5. $t = t + 1$, возврат к шагу 2.

Последовательные модификации базового генетического алгоритма.

Как показывает анализ, модификации ГА отличаются, прежде всего, способом селекции индивидов. В основных модификациях ГА несколько способов селекции используется для достижения различных целей - упрощения формирования промежуточной популяции, распараллеливания работы алгоритма, возможности анализа и предсказания поведения ГА. Было произведено сравнение четырех различных схем селекции (для СГА и SSGA, рассматриваемых далее, показавшее, что

эффективность всех методов примерно одинакова. Таким образом, в настоящее время абсолютно лучший метод селекции не определен.

Модификация

стандартного варианта ГА (Steady State GA) [Whitley и Kauth, 1988] затронула способ формирования промежуточной популяции (Mating Pool), являющейся результатом отбора (селекции) для формирования наследников с помощью генетических операторов. SSGA не формируют промежуточную популяцию как стандартный ГА, а осуществляют последовательно выбор пары наилучших индивидов, применяя к ним генетические операторы с целью формирования наследников, которые заменяют худшие индивиды популяции. Данная модификация ГА представлена на рисунке ниже.

SSGA.

Шаг 0. Определение генетического представления задачи.

Шаг 1. Создание первоначальной популяции $P(0) = x_1^0, \dots, x_N^0, t = 0$.

Шаг 2. Вычисление относительной (нормализованной) степени пригодности

$$f_n(x_i) = f(x_i)/\sum_i^n f(x_i)/N.$$

Шаг 3. Выбор пары из лучших индивидов. Выбор худшего индивида. Применение скрещивания и мутации к выбранной паре лучших индивидов. Результат замещает худший индивид.

Шаг 4. $t = t + 1$, возврат к шагу 2.

При проектировании ГА могут быть выгодно использованы знания, полученные селекционерами в области искусственной селекции. Генетические алгоритмы селекционеров (ГАС) моделируют именно искусственную селекцию. ГАС представлен на рисунке ниже, где под виртуальным селекционером понимается некоторый механизм селекции, который и является основным отличием ГАС от стандартного ГА.

ГАС.

Шаг 0. Определение генетического представления задачи.

Шаг 1. Создание первоначальной популяции $P(0)$ размером N , $t = 0$.

Шаг 2. Виртуальный селекционер отбирает $T\%$ популяции для создания потомков. Это дает набор отобранных родителей.

Шаг 3. Формирование случайным образом из данного набора $N/2$ пар. Применение к каждой паре скрещивания и мутации, формируя новую популяцию $P(t + 1)$.

Шаг 5. $t = t + 1$, возврат к шагу 2.

Шаг 6. Возврат к шагу 3.

Селекция основывается преимущественно на статистических методах, которые позволяют произвести теоретический анализ и прогнозировать эффективность механизмов селекции, мутации и рекомбинации с помощью введенных уравнений селекции, реакции на селекцию и понятия наследственности.

Еще одна модификация ГА затрагивает решение многокритериальных задач. Многокритериальный ГА (МГА) также является модификацией стандартного ГА и

отличается способом селекции, поскольку при отборе пар родителей в этом случае используется не один, а несколько критериев. При этом предлагается большое число вариантов схем селекции и соответственно вариантов МГА. На рисунке ниже приведен вариант МГА, предложенный SchafFer в 1984 г., — векторный ГА (VEGA). Сравнительные оценки показывают, что по эффективности VEGA имеет средние показатели, однако не оценивалась вычислительная сложность для различных вариантов МГА, по которой VEGA может существенно улучшить свои показатели.

Многокритериальный ГА.

Шаг 0. Определение генетического представления задачи.

Шаг 1. Создание первоначальной популяции $P(0) = x_1^0, \dots, x_N^0, t = 0$.

Шаг 2. Последовательное выполнение шагов 2.1–2.3.

Шаг 2.1. Вычисление значения степени пригодности каждого индивида по критерию $i = 1, \dots, k$.

Шаг 2.2. Для j от 1 до N/k осуществление селекции индивида из популяции в промежуточную популяцию.

Шаг 2.3. Возврат к шагу 2.1, если $j < k$.

Шаг 3. Формирование случайным образом из данного набора $N/2$ пар. Применение к каждой паре скрещивания, а также других генетических операторов, таких как мутация, формируя новую популяцию $P(t + 1)$.

Шаг 4. $t = t + 1$, возврат к шагу 2.

Рис. П2.8. Многокритериальный генетический алгоритм

43. Параллельные модификации базового генетического алгоритма.

Классификация генетических алгоритмов.

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

По степени распараллеливания можно выделить следующие типы параллельных ГА:

ПГА на базе популяции;

ПГА на базе подпопуляций;

ПГА на базе индивидов.

ПГА на базе популяции сохраняет стандартную структуру ГА, работающего с целой популяцией, распараллеливание реализуется на этапе скрещивания и мутации

(см. шаг 4, рис. 3). По степени распараллеливания процессов можно выделить следующие модели:

синхронная модель "ведущий-ведомый", где главный процесс хранит целую популяцию в собственной памяти, выполняет селекцию, скрещивание и мутацию, но оставляет вычисление степени пригодности новых индивидов к подчиненным процессам;

полусинхронная модель "ведущий-ведомый", где новый индивид обрабатывается по мере освобождения одного из процессов;

асинхронная параллельная модель, где индивиды популяции хранятся в общей памяти, к которой можно обращаться к параллельным процессам. Каждый процесс выполняет оценку степени пригодности, а также генетические операции.

Каждый процесс работает независимо от других. Единственное отличие между этой моделью и стандартным ГА заключается в механизме селекции. Очевидным в этом случае является вариант использования $N/2$ параллельных процессоров при популяции в N индивидов. Тогда каждый процессор дважды случайным образом выбирает два индивида из общей памяти и оставляет лучшего. Два выбранных индивида затем подвергаются скрещиванию, мутации и оценке степени пригодности. Возникающие в результате наследники размещаются в общей памяти.

Распределенный ПГА.

Шаг 0. Определение генетического представления задачи.

Шаг 1. Создание первоначальной популяции индивидов и разделение на подпопуляции SP_1, \dots, SP_N .

Шаг 2. Формирование структуры подпопуляций.

Шаг 3. Для $SP_i, i = 1, \dots, N$ — выполнение параллельно шагов 3.1–3.3.

Шаг 3.1. Применение в течение t поколений селекции и генетических операторов.

Шаг 3.2. Перемещение k хромосом в соседние подпопуляции.

Шаг 3.3. Получение хромосом из соседних подпопуляций.

Шаг 4. Возврат к шагу 3.

Рис. 7. Распределенный параллельный генетический алгоритм

Особенность **ПГА на базе подпопуляции** заключается в использовании независимых конкурирующих подпопуляций, которые обмениваются индивидами с заданной частотой (распределенный ПГА, рис. 7). При этом каждый процессорный блок выполняет последовательный ГА с собственной подпопуляцией, при условии максимизации одной общей для всех функции степени пригодности. В этом случае для обмена индивидами должна быть определена структура связей подпопуляций. С точки зрения оценки и сравнения

эффективности может быть рассмотрен вариант распределенной модели, в которой обмен индивидами не осуществляется. Результаты проведенных экспериментов свидетельствуют о большей эффективности распределенного ПГА по сравнению с этим частным случаем, а также со стандартным ГА.

Существенным недостатком модели может стать снижение степени разнообразия при интенсивном обмене индивидами. С другой стороны, недостаточно частое перемещение может привести к преждевременной сходимости подпопуляций. При построении такой модели важно определить следующее:

связи между процессорами для обмена индивидами;

частоту обмена индивидами (оптимальной является частота обмена через 20 поколений);

степень перемещения или число обмениваемых индивидов (оптимальным является 20 % подпопуляции);

способ селекции индивида для обмена;

критерий, по которому полученный индивид сможет заменить члена подпопуляции,

С точки зрения времени и даже числа поколений, затрачиваемых на решение задачи, ПГА эффективнее стандартного ГА, но при этом некоторые задачи могут быть слишком простыми для ПГА. Параллельный поиск имеет смысл в том случае, если пространство поиска большое и сложное. Увеличение числа процессоров в данной модели улучшает скорость сходимости, но не качество решения.

ПГА на базе индивидов имеют одну строку индивида, постоянно находящуюся в каждом процессорном элементе (ячейке). Индивиды выбирают пары и рекомбинируют с другими индивидами в их непосредственном ближайшем окружении (по вертикали и горизонтали). Выбранный индивид затем совмещается с индивидом, постоянно находящимся в ячейке. В результате формируется один наследник, который может или не может заменить индивида в ячейке в зависимости от выбранной схемы замещения. Таким образом, модель является полностью распределенной и не нуждается в централизованном управлении (рис. 8).

ПГА на базе индивидов.

- Шаг 0. Определение генетического представления задачи.
- Шаг 1. Создание первоначальной популяции индивидов и формирование структуры популяции.
- Шаг 2. Локальное повышение каждым индивидом своей производительности (hill-climbing).
- Шаг 3. Выполнение каждым индивидом селекции с целью поиска пары.
- Шаг 4. Применение к паре скрещивания, а также других генетических операторов, таких как мутация.
- Шаг 5. Локальное повышение наследником своей производительности (hill-climbing).
- Замещение наследником родителя в соответствии с заданным критерием качества.
- Шаг 6. Возврат к шагу 3.

Рис. 8. Параллельный генетический алгоритм на базе индивидов

При работе с моделью на базе индивидов необходимо задать:

структуру связей ячеек;

схему селекции;

схему замещения.

Исследования этой модели показали, что для сложных задач она способна обеспечить лучшие решения, чем стандартный ГА.

Классификация генетических алгоритмов

В ходе исследований в области генетических алгоритмов и эволюционных алгоритмов в целом появилось большое количество направлений, и их число непрерывно растет.

Классификация ЭА и основные модификации стандартного ГА, приведенного на рис. 2, отражены на рис. 9.

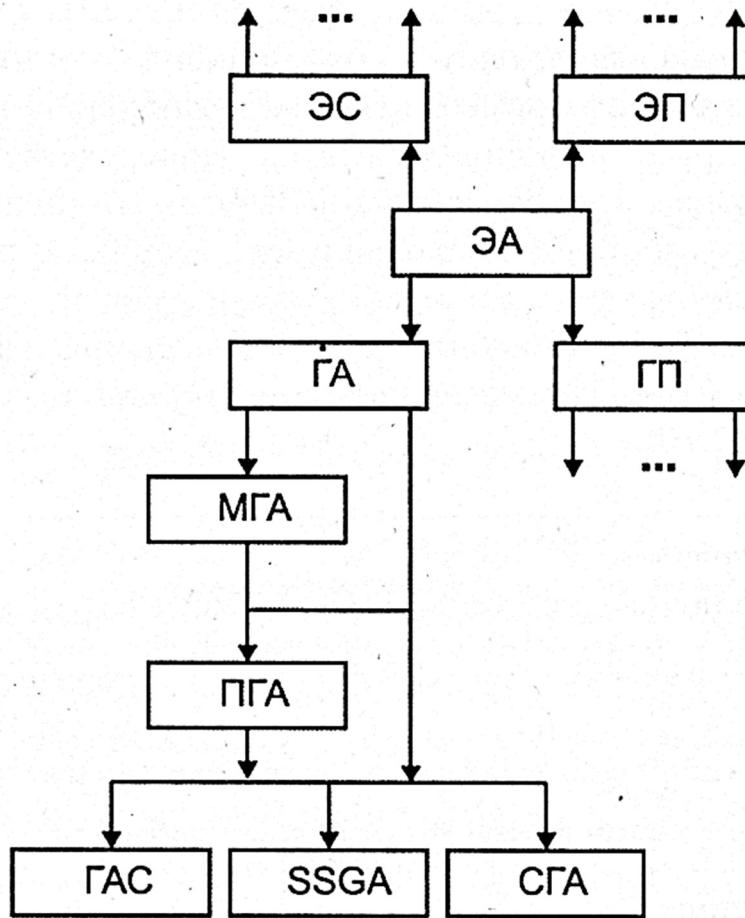


Рис. 9. Классификация эволюционных алгоритмов

44. Когнитивная компьютерная графика. Определение. Степень организованности информации.

Когнитивная компьютерная графика (ККГ) – компьютерные системы визуализации данных, позволяющие активировать наглядно-образные механизмы мышления ЛПР, облегчающие принятие решения в сложной обстановке или нахождение решения сложной проблемы. Суть концепции ККГ заключается в том, что если на экране дисплея удаётся визуализировать существенные свойства и отношения между объектами некоторой предметной области (любой степени абстрактности), то такой ККГ – образ, как правило, содержит в себе информацию (на уровне графических деталей компьютерного изображения) о возможных и не всегда заранее известных следствиях этих свойств и отношений, помогающую проанализировать новые закономерности исследуемой предметной области

Степень организованности информации

В общем случае, применительно к искусственным системам, степень организованности информации включает в себя структурную, параметрическую и динамическую организованности. Так, например, при сопоставлении возможных вариантов организованности информационных образов: одномерных, и двухмерных, целесообразно использовать пространственно-структурно-параметрические показатели, которые будут более высокими у двухмерных информационных образов. Для того чтобы достигнуть высокой степени организации информационных образов в информационных системах отображения, передачи, воспроизведения и представления информации необходимо располагать запасами ее пространственных, структурных, параметрических и динамических ресурсов. Одним из путей получения высокоорганизованной информации и подбора интеллектуальных средств, реализующих эту информацию, является визуализация информационных образов. В свою очередь, выбор рациональных вариантов реализации интеллектуальных средств требует решения главной задачи: как адаптировать выбираемые средства к конкретному ЛПР с его субъективными психофизиологическими данными. Если эта задача не будет хотя бы частично решена, то и главная цель повышения оперативности принятия решения, при резком увеличении информационной трудоемкости его выработки, также не будет достигнута.

Для достижения этой цели, необходимо выяснить, по каким каналам органов чувств ЛПР легче всего воспринимает поступающую к нему информацию. Как известно, наиболее развиты зрительные и слуховые каналы или, как принято, в психологии, каналы зрительной и слуховой модальности по которым человек получает порядка 95-97% информации об окружающем мире. Одно из самых первых оснований для выбора той или иной модальности было предложено Н. Винером (N. Wiener), который рассматривал критерии эффективности этих каналов. По Винеру, критерием является отношение между числом зрительных и слуховых образов на уровне коры головного мозга. Организация различных полей коры головного мозга, связанных со зрением, и площадей коры головного мозга, обслуживающих слух, может быть использована при сравнении эффективности зрения и слуха. Проведенные в работе [Глазер, Цукерман, 1961] опыты показали, что это отношение равно 100:1. Анализ трудов по теории информации и инженерной психологии [Беляев, Капустян,

1999; Белый и др., 1999] выявил количество информации, которое может быть передано по зрительному и слуховому каналам. При прохождении по нервным волокнам, идущим от глаза и уха к мозгу ЛПР (в среднем 300 имп/с), пропускная способность зрительного канала равна $6 \cdot 108$ дв.ед./с, а слухового - $18 \cdot 106$ дв.ед./с, что значительно меньше ($6 \cdot 108 / 18 \cdot 106 = 33,33$) количества информации проходящей через зрительный канал. Но сам по себе способ определения количества информации, с учетом лишь проводимости нервного волокна, не принимая во внимание мыслительную деятельность, доказывает лишь то, что поток информации к ЛПР поступает, в основном по зрительному каналу.

Задачу выбора ЛПР варианта решения на основе восприятия им информации и его мыслительной деятельности, можно рассмотреть с точки зрения гипотезы выдвинутой Ньюэллом [Newell], Саймоном [Simon] и Шоу [Show] [Глезер, Цукерман, 1961] Эта гипотеза, в рамках информационного анализа, объясняет, как представляются объекты мышления в психике ЛПР. В результате проведенного анализа было выяснено, что внутренний образ может быть отнесен к зрительной модальности в том случае, если он по природе своей таков, что может служить точкой приложения, входом для информационных процессов, характерных для обработки визуальных образов. Эти процессы для переработки визуальной информации, применяемые при обработке внутренних образов, могут, по мнению этих исследователей, расцениваться как «мыслительные образы».

Отсюда можно сделать вывод, что при реализации средств интеллектуальной поддержки принятия решения следует ориентироваться на то, чтобы информация, представляемая ЛПР носила бы визуальный характер в виде визуальных информационных образов. Тогда ее восприятие и обработка не встретят противоречий со стороны внутренних образов мыслительных процессов, и приведет к резкому сокращению времени принятия решения.

45. Задача сжатия информации. Анаморфозы (определение).

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

Пусть визуальный информационный образ – это вход нашей системы переработки информации, а показания, снятые человеком и пропущенные через его мыслительный аппарат – выход. Количество информации, переданное от входа к

выходу, при прочих равных условиях будет изменяться с изменением характера входа. И это связано не только с изменением его информационного содержания, но и с изменением способа кодирования информационного образа и формы его представления.

Допустим, что ЛПР получает информационное сообщение с помощью двух видов визуальных информационных образов: одномерного, двухмерного. Первый – одномерный визуальный образ, в котором сообщение передается текстом, а слова следуют одно за другим («бегущая строка»). Второй вид визуального образа передает то же самое информационное сообщение с помощью двухмерного представления информации на плоскости: графика, чертежа, пиктограммы, графического образа. Для того чтобы оценить эффективность передачи и восприятия информационного образа необходимо вероятность правильного восприятия P , разделить на время восприятия t (χ) – $Q=P/t$ ($\chi-1$). Если вероятность правильного восприятия в обоих случаях равна единице, то время оказывается разным, так как длительность восприятия сообщения в первом и во втором случае разная. Во втором случае, ЛПР достаточно одного взгляда на графический образ, чтобы полностью воспринять информационное сообщение.

При переходе от одномерного визуального информационного образа к двухмерному, ЛПР группирует первичные элементы (слова и смысловые понятия первичного информационного сообщения), увеличивая число этих элементов в информационной единице, «спрессовывая» их в новый визуальный образ, т.е. производя перекодировку входа с изменением формы представления визуального образа и его информационной емкости.

Принятие решения во многих практических задачах связанных с местностью сталкивается с необходимостью получения и анализа весьма обширной и специфической информации. В основном это сведения об отдельных характеристиках местности (оценки проходимости и свойств местности, анализа дорожной сети и т.п.) и объектах, расположенных на ней.

Развитие многих прикладных задач, связанных с пространственно-временным анализом, предполагает не только совершенствование способов отображения географических явлений, но и показ отношений и связей с другими явлениями, особенно в тех случаях, когда их анализируют как системы. Возникает необходимость исследовать изменяющиеся в пространстве характеристики сразу нескольких явлений. Выполнить такой анализ проще, если хотя бы одну из меняющихся в пространстве характеристик полагать равномерно распределенной и на ее фоне анализировать все остальные с ней взаимосвязанные. Для этой цели прибегают к преобразованию визуального образа, (площадной фигуры) взятого за основу, из евклидовой метрики в условное «усредненное пространство». Под таким преобразованием понимается переход от визуального образа, в основу которого положена обычная метрика, к другому визуальному образу, в основу которого положена метрика рассматриваемого явления. Подобные преобразованные визуальные образы принято называть анаморфозами.

Среди анаморфоз можно выделить линейные, площадные и объемные (в виде трансформированных диаграмм, обычных или рельефных карт).

46. Метод анаморфирования.

Метод анаморфирования для формирования варианта решения рассмотрим на примере связанном с анализом проходимости местности и ее изменениями в ходе различных мероприятий.

Под проходимостью понимают возможность самостоятельного, без проведения инженерных мероприятий, движения машины (машин) данного типа и класса по неподготовленной для этого местности, с учетом расположенных на ней объектов. Количественно проходимость оценивается главным параметром движения – скоростью движения.

С учетом этого, в каждой ячейке предварительно нанесенной на ЭКМ регулярной сетки шага k , рассчитываются значения скоростей движения колесной (гусеничной) техники в заданном направлении, зависящие от целого ряда условий (рельефа местности, грунтов, времени года, залесенности, водных преград и т. д.) и принять их за показатели анаморфирования каждой ячейки. После применения алгоритма анаморфирования регулярная сетка деформируется пропорционально значениям скоростей движения колесной (гусеничной) техники в каждой ячейке, на ней образуются группы ячеек увеличенной площади (относительно ячеек, показатель анаморфирования которых равен среднему, а площадь $S=1$ усл. ед.) показывающие направления и участки местности наиболее вероятного передвижения колесной техники, а также группы ячеек уменьшенной площади показывающие места, где передвижение колесной (гусеничной) техники маловероятно или невозможно.

Пусть D – область на плоскости R^2 (площадная фигура, построенная на основе выбранного показателя), которая должна быть анаморфизирована. Распределение показателя описывается функцией плотности $\rho(z)$, определенной априори на части D ($z = (x, y)$ – точка на плоскости R^2). Без потери общности можно полагать, что $\rho(z)$ определена на всей плоскости R^2 . Тогда она является const вне области D (например, как среднее значение $\bar{\rho}$ функции $\rho(z)$).

Анаморфоза задается преобразованием $h: R^2 \rightarrow R^2$ ($h: (x, y) \mapsto (u, v)$) или двумя функциями двух переменных: $U(x, y)$ и $V(x, y)$,

где $u = U(x, y)$, $v = V(x, y)$. (1)

Эти функции должны быть определены и непрерывны на D .

Коэффициент изменения площади в окрестности точки (x, y) преобразованием h равен значению якобиана преобразования h в этой точке

$$J(U, V) = \frac{\partial U}{\partial x} \cdot \frac{\partial V}{\partial y} - \frac{\partial U}{\partial y} \cdot \frac{\partial V}{\partial x}.$$

Поэтому условие того, что преобразование (1) делает величину $\rho(x, y)$ постоянной и равной $\bar{\rho}$, может быть записано как $J(U, V) = \rho(x, y) / \bar{\rho}$.

Таким образом, задача нахождения анаморфозы сводится к задаче решения уравнения

$$\frac{\partial U}{\partial x} \cdot \frac{\partial V}{\partial y} - \frac{\partial U}{\partial y} \cdot \frac{\partial V}{\partial x} = \frac{\rho(x, y)}{\bar{\rho}}, \quad (2)$$

для которого $[U(x, y), V(x, y)]$ определяют взаимно-однозначное преобразование.

47. Проблемы реализации анаморфоз.

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

Второй класс трудностей связан с тем, что условие выравнивания заданной плотности не определяет анаморфозу однозначно. Существует бесконечно много преобразований, удовлетворяющих этому условию. Это следует из того, что для двух неизвестных функций $U(x, y)$ и $V(x, y)$, задающих анаморфозу, имеется только одно уравнение (2). Построенное анаморфизированное изображение без нарушения постоянства плотности может быть изменено применением любого преобразования, сохраняющего площадь, например:

- 1) $(u, v) \mapsto (k \cdot u, k^{-1} \cdot v)$ (растяжение вдоль одной из осей и сжатие вдоль другой с тем же коэффициентом);
- 2) $(u, v) \mapsto (u + f(v), v)$, $(u, v) \mapsto (u, v + g(u))$ (сдвиг горизонтальных и вертикальных прямых вдоль самих себя на различные расстояния).

В качестве требования при выборе анаморфозы можно использовать условие конформности преобразования (1). Конформное преобразование изменяет все расстояния умножением на один и тот же коэффициент, не зависящий от направления (углы между прямыми линиями сохраняются). Преобразование, которое локально изменяет все расстояния с помощью умножения на $\sqrt{\rho(x, y)}$ единственно, и не зависит от выбора системы координат.

Конформное преобразование с заданным коэффициентом линейного растяжения, равным $\sqrt{\rho(x, y)}$ (или с коэффициентом изменения площадей, равным $\rho(x, y)$) существует не всегда. Условие конформности преобразования (2) может быть записано в виде

$$\begin{cases} \left(\frac{\partial U}{\partial x}\right)^2 + \left(\frac{\partial V}{\partial x}\right)^2 = \rho(x, y) \\ \left(\frac{\partial U}{\partial y}\right)^2 + \left(\frac{\partial V}{\partial y}\right)^2 = \rho(x, y) \\ \frac{\partial U}{\partial x} \cdot \frac{\partial V}{\partial y} + \frac{\partial V}{\partial x} \cdot \frac{\partial U}{\partial y} = 0 \end{cases}$$

Это условие состоит из трех уравнений относительно двух неизвестных функций $U(x, y)$ и $V(x, y)$. Обычно такая система не имеет решений. Для существования такого преобразования необходимо, чтобы функция плотности $\sqrt{\rho(x, y)}$ удовлетворяла уравнению

$$\Delta \ln \rho \equiv \frac{\partial}{\partial x} \left(\frac{\partial \rho / \partial x}{\rho} \right) + \frac{\partial}{\partial y} \left(\frac{\partial \rho / \partial y}{\rho} \right) = 0,$$

где $\Delta = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$ - лапласиан.

Это уравнение означает что кривизна, определенная метрикой $\rho(x, y)(dx^2 + dy^2)$, должна быть равна нулю.

Таким образом, возникает задача поиска среди различных преобразований того, которое инвариантно к любой замене декартовой системы координат и конформно.

48. Численные методы построения анаморфоз.

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

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

Алгоритм построения анаморфизированных изображений лаборатории Лоуренс Беркли [Selvin и др., 1984]. Этот метод позволяет получить анаморфозы хорошего качества и его реализация относительно проста. Основные недостатки метода состоят в том, что: - окончательный результат существенно зависит от порядка, в котором берутся ячейки; - ошибки, допущенные на каждом шаге, не исправляются позже и накапливаются.

Анализируя недостатки приведенных алгоритмов, можно сделать выводы о путях их улучшения: - необходимо, чтобы на каждом шаге на сдвиг точек (в том числе – вершин) в той или иной степени влияли все ячейки разбиения. Т.е. на каждом шаге сдвиг точки должен быть равен векторной сумме сдвигов, от влияния отдельных ячеек.

Во-вторых, влияние ячейки на точку должно состоять в перемещении этой точки вдоль прямой, соединяющей ее с некоторой точкой ячейки (например, с ее центром масс). Это условие связано с требованием инвариантности алгоритма по отношению к выбору системы координат.

В-третьих, перемещение точки под влиянием ячейки должно убывать с увеличением расстояния от этой ячейки.

Первый алгоритм такого рода был предложен в [Dougenik и др., 1985]. В нем рассмотрена ситуация, когда на бесконечной территории имеется ограниченная часть (без потери общности можно предположить, что эта часть имеет форму круга), на которой плотность распределения показателя отличается от средней по всей плоскости (на остальной части территории плотность равна средней). При естественном анаморфировании сдвиги точек меньше для тех из них, которые расположены дальше от указанной части территории. В [Гусейн-Заде, Тикунов, 1999] показано, что при таком определении влияния ячейки на вершины, которое уменьшается с увеличением расстояния, рассматриваемый алгоритм будет сходиться быстрее. В качестве ячеек используются любые связные области произвольной формы.

Этот алгоритм можно описать следующим образом. На каждом шаге вычисляется векторная сумма влияний центров ячеек на вершины и центры многоугольных ячеек, составляющих визуальный образ, которые перемещаются в соответствии с полученными векторами сдвигов. Для полученной конфигурации вычисляются новые площади ячеек. Итерационный процесс прекращается, когда все относительные отклонения площадей ячеек становятся меньше заданной величины ϵ .

В общем случае очевидными достоинствами анаморфозы электронной карты местности на основе выбранного показателя являются:

- наглядность – выявление скрытых закономерностей поведения параметров, зависящих от выбранного показателя анаморфирования;
- быстрый визуальный анализ – анаморфирование исходной матрицы выбранного показателя, по любому его допустимому значению позволяет принимать решения в направлениях и областях с данным значением показателя;
- возможность построения сценариев развития ситуации на основе анализа динамической анаморфозы, учитывающей выполнение задач связанных с быстрой эволюцией выбранного показателя;
- возможность принятия решения с учетом комплексного показателя анаморфирования (например, скорости движения колесной техники и степени заражения местности и т.д.) рис.1-3; уменьшение пространства принятия решения на количество взятых показателей анаморфирования.

49. Основные законы гибридного интеллекта.

В.Ф. Венда сформулировал **три закона гибридного интеллекта**.

Закон взаимной адаптации.

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

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

Закон дискретных рядов структур.

Любой гибрид может быть реализован посредством одной из дискретного ряда его возможных структур.

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

Закон трансформации.

Трансформация одной структуры гибрида в другую может происходить только через общие для обеих структур знания.

Закон описывает образование новых знаний их интерференцию между собой, построение умозаключений, а также возникновение и роль ассоциаций в трансформации и связывании образов и мыслей. Любые новые знания могут быть получены только путем перехода от одного знания к новому, взаимосвязанному, ассоциированному с предыдущим.

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

В теории систем приведенные законы описывают гибридизацию как процесс создания гибридных методов, гибридных моделей, гибридных алгоритмов и гибридных программ.

50. Основные методы гибридизации.

Метод Н1 (рис. 1). Метод основан на том, что одна и также задача может быть решена несколькими известными автономными методами, а вычисления, начиная с некоторого пункта алгоритма, могут быть продолжены тем или иным методом.

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



Рис. 1. Структура метода гибридизации Н1

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

Принципиальной особенностью метода Н1 является то, что решение принимает блок, встроенный в алгоритм решения задачи. Решение задачи продолжает один метод, а гибрид после завершения работы алгоритма перестает существовать. Это дает основание говорить о гибридном алгоритме решения задачи, а формируемая в ходе работы алгоритма последовательность применяемых методов может рассматриваться как гибридный метод решения задачи.

Метод Н2 (рис. 2).

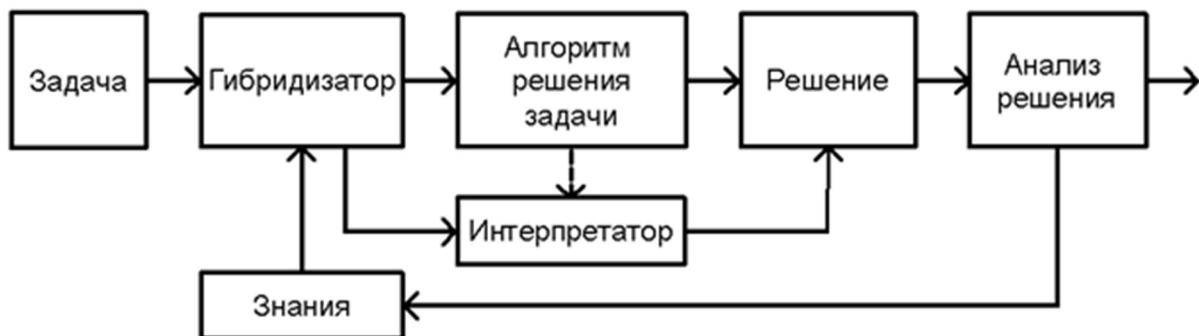


Рис. 2. Структура метода гибридизации Н2

В отличие от метода Н1 данный подход предполагает, что метод решения исходной неоднородной задачи неизвестен, однако она предварительно была редуцирована на некоторое количество однородных задач, для каждой из которых известно некоторое множество методов ее решения. Есть также знания об областях релевантности автономных методов. В этом случае гибридизатор, используя декомпозицию неоднородной задачи, множество автономных моделей, знания об областях релевантности, а также знания ЛПР, строит алгоритм решения задачи как структуру над алгоритмами решения однородных задач, обрабатываемую интерпретатором. После получения интерпретатором решения задачи, оно анализируется ЛПР и принимается. В противном случае знания гибридизатора корректируются, может быть построен новый алгоритм, с выполнением повторного решения неоднородной задачи и так далее, до тех пор, пока итерационный процесс, по мнению ЛПР, может быть завершен и найдено решение исходной задачи.

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

Особенность метода Н2 - ориентация на неоднородные задачи, конструирование символьного эквивалента гибрида, который может быть сохранен, отредактирован и интерпретирован в любое время в зависимости от ситуации решения задачи. Важным моментом метода Н2 является использование для решения задачи знаний ЛПР и возможность коррекции знаний гибридизатора.

Метод Н2 может применяться в двух вариантах в зависимости от того, какие знания будут использоваться в гибридизаторе.

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

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

Гибридизацию первого вида в литературе называют крупнозернистой, а второго – мелкозернистой.

Обобщая вышесказанное, можно сделать вывод о том, что гибридизация это сложный и трудоемкий процесс, требующий, во-первых - привлечения широкого спектра знаний ЛПР и экспертов о предметной области, задачах, методах их решения и во-

вторых выполнения сложной переработки информации и организации большого количества экспериментов.

В связи с резким расширением в последние годы круга решаемых с помощью ГиИС задач управления актуальным становится их разработка и применение, как инструмента решения задач такого типа.

51. Общий подход к построению гибридной интеллектуальной системы.

Общий подход к построению ГиИС, применяемых при решении задач может быть представлен **следующим образом**.

ГиИС, применяемая при решении задач синтезируется агрегированием на множестве автономных моделей с помощью отношений интеграции. Отношения интеграции должны заменить отношения редукции и отношения, связывающие однородные задачи. В результате агрегирования синтезируется, подбирается структура релевантная неоднородностям задачи, которая может рассматриваться как метод ее решения. Метод можно динамично изменять в зависимости от результатов анализа неоднородной задачи, что приводит к неоднократному повторению редукции, установления соответствия и агрегированию в цикле «анализ неоднородной задачи - синтез ГиИС».

52. Принципы построения гибридных интеллектуальных систем.

1. **Принцип системного анализа сложной задачи.** Решению сложной задачи должен предшествовать системный анализ ее свойств, состава и структуры, что позволяет сделать более отчетливыми границы однородных областей и подобрать релевантные этим областям автономные методы. Вывод о применении для решения задачи ГиИС делается по результатам ее системного анализа.

2. **Принцип неоднородности.** При переходе к решению практических задач, возникающих не в искусственно созданной, а в реально сложившейся и эволюционирующей среде, разработчик неизбежно сталкивается с многообразием парадигм, методов и переменных в науке, дисциплин в обучении, мнений и моделей внешнего мира на практике, фаз управления, целей решения задач, отношений на знаниях. Следствием такого многообразия является неоднородность сложных задач, требующая от разработчика отказа от попыток применить для моделирования решения автономные методы.

3. **Принцип конструирования.** Метод решения сложной задачи синтезируется из методов, моделей, модулей, инструментальных средств и технологий всякий раз заново, когда возникает необходимость решения сложной задачи. Приступая к конструированию, необходимо знать плюсы и минусы методов и инструментария, из которых строятся ГиИС, оценки надежности знаний о подзадаче, трудоемкость решения подзадачи тем или иным методом, или инструментарием.

4. **Принцип плюрализма.** Нет ни одного, окончательно разработанного метода для объяснения или решения сложной, неоднородной задачи. Тем не менее, существует некоторое подмножество уже разработанного множества методов и моделей, которое может быть использовано для моделирования решения сложной, неоднородной задачи.

5. **Принцип приоритета знаний.** ГиИС должна быть построена таким образом, чтобы первый приоритет в решении сложной задачи отдавался точным знаниям и жестким вычислениям и только второй - эвристическим знаниям и мягким вычислениям. При этом использование эвристик должно рассматриваться как коррекция решения, полученного на точных знаниях. В случае отсутствия точных знаний возможно применение одних эвристик, однако в этом случае должна быть заранее поставлена цель получения таких точных знаний.

6. **Принцип постепенности.** Прежде чем разрабатывать ГиИС, необходимо накопить опыт и знания построения или использования автономных моделей. Это позволит избежать ошибок в автономных моделях, которые неизбежно перейдут в ГиИС и снизят ее качество.

7. **Первый принцип наследования.** Чтобы функциональная ГиИС унаследовала сумму плюсов методов-прототипов, необходимо, чтобы степень зернистости и интерфейсы, реализующие отношения интеграции, были установлены как на уровне декларативных, так и процедурных представлений.

8. **Второй принцип наследования.** Архитектура ГиИС наследуется исходя как из состава и структуры сложной задачи, так и от плюсов и минусов автономных методов. Отказ от релевантности проблеме ведет к хорошим, но бесполезным гибридам. Отказ от учета плюсов и минусов автономных методов ведет к снижению качества ГиИС.

9. **Принцип самоорганизации.** ГиИС должна обучаться и извлекать знания из одного элемента для совершенствования других элементов. Это обеспечивает адаптацию ГиИС к новым условиям и снижает трудозатраты на ее эксплуатацию.

10. **Принцип полноты.** ГиИС должна строиться с использованием как можно большего числа классов автономных методов: аналитических, статистических, символьных, коннекционистских и эволюционных.

11. **Принцип снижения производительности.** ГиИС может использоваться там, где не существенно снижение производительности вычислений по сравнению с автономными моделями.

Вопросы которых не было

Приведите оценку уровня компетентности экспертов
семинар 7

Уровень компетентности экспертов оценивается следующим образом:

$$K_{\Theta i} = \frac{K_{ai} + K_{ui}}{K_{amax} + K_{umax}} = \frac{K_{ai} + K_{ui}}{2}, \quad (1)$$

где $K_{\Theta i}$ - коэффициент, отражающий уровень компетентности i -го эксперта; K_{ai} - коэффициент, отражающий уровень аргументации i -го эксперта; K_{ui} - коэффициент, отражающий уровень информированности i -го эксперта; K_{amax} и K_{umax} - максимальные оценки коэффициентов K_{ai} и K_{ui} обычно равные 1.

Коэффициенты $K_{\Theta i}, K_{ai}, K_{ui}$ могут принимать значения от 0 до 1. K_{ui} определяется на основе самооценки информированности эксперта в баллах по 10-балльной шкале, умноженной на 0,1. Например, если эксперт регулярно, один раз в неделю, знакомится с новыми материалами, он заслуживает оценки в 10 баллов, если один раз в две недели, то - 9 баллов, если один раз в месяц, то - 8 баллов, если один раз в два месяца, то - 7 баллов, если один раз в три месяца, то - 6 баллов, если один раз в 6 месяцев, то - 5 баллов. K_{ai} определяется в результате суммирования баллов, представленных в эталонной таблице (табл. 1).

Основа метода экспертного анализа

семинар 6

Метод экспертных оценок Дельфи довольно часто используют при принятии решения. Он представляет собой процедуру как удаленного, так и локального анкетирования экспертов с последующей детальной обработкой всей полученной от них информации.

В основе метода Дельфи лежат следующие принципы сбора и обработки информации:

- 1) эксперты должны получить анкету с корректными вопросами и дать четкий ответ на каждый из них;
- 2) опрос экспертов может проводиться как в один, так и в несколько туров, в ходе которых они постоянно уточняют свои ответы за счет ознакомления с материалами и обмена мнениями с другими экспертами;
- 3) итоговый результат заключается в получении количественной оценки специалистов по каждому из рассматриваемых альтернативных вариантов в целях выбора преобладающего мнения для определения наилучшего варианта решения.

Следует отметить, что полной согласованности мнений экспертов не всегда удается добиться вследствие различий научных школ и течений, которые они представляют, а также их индивидуальных особенностей.

Три подхода к повышению согласованности мнений экспертов

Для такой корректировки на практике нужно применять следующие подходы:

- 1) использовать коэффициенты важности (весовые коэффициенты) ранговых оценок мнений экспертов, вычисленные с учетом уровня их компетентности;
- 2) использовать коэффициенты важности (весовые коэффициенты) ранговых оценок мнений экспертов, вычисленные с учетом результатов решения ими тестового задания и полученного на его основе уровня согласованности их мнений;
- 3) исключить из рассмотрения результатов работы экспертной группы результаты того эксперта, который имеет наименьший коэффициент согласованности мнений с остальными экспертами группы.

Первый подход. Весовые коэффициенты α_j^* ($j=1, \dots, z$) ранговых оценок мнений экспертов рассчитываем с учетом уровня компетентности экспертов K_j по формуле:

$$\alpha_j^* = K_j / \sum_{j=1}^z K_j. \quad (\text{П2.2})$$

Тогда с учетом весовых коэффициентов скорректированные значения рангов вариантов r_{ij}^* по сравнению с теми значениями r_{ij} , которые первоначально присвоили им эксперты, вычисляются по формуле

$$r_{ij}^* = \alpha_j^* \cdot r_{ij}. \quad (\text{П2.3})$$

Если эксперты имеют одинаковые весовые коэффициенты ранговых оценок, то $\alpha_j = 1/z$ и $r_{ij}^* = r_{ij}$.

При этом для стандартизированной ранговой системы при средних значениях рангов всех вариантов выполняется условие $r^* = r$.

Второй подход. Весовые коэффициенты α_j^{**} ранговых оценок мнений экспертов рассчитываются с учетом результатов решения экспертами тестового задания поэтапно:

- на первом этапе на основе ранговых оценок отдельных экспертов r_{ij} строим общую матрицу исходных ранговых оценок сравниваемых вариантов;
- на втором этапе для сравнения ранговых оценок каждой пары экспертов вычисляем коэффициент ρ_{jk} ранговой корреляции Спирмена по формуле

$$\rho_{jk} = 1 - \frac{6 \sum_{i=1}^m (r_{ij} - r_{ik})^2}{m(m^2 - 1) - 0,5(s_j - s_k)}. \quad (\text{П2.4})$$

При этом

$$s_j = \sum_{j=1}^H (t_j^3 - t_j) \text{ и } s_k = \sum_{k=1}^h (t_k^3 - t_k),$$

где i — номер оцениваемого варианта; m — число вариантов, подлежащих сравнению; r_{ij} , r_{ik} — ранги, данные i -му варианту соответственно j -м и k -м экспертами; s_j и s_k — показатели повторяемости рангов для сравниваемых экспертами вариантов; t_j , t_k — число повторений каждого ранга соответственно у j -го и k -го экспертов; H, h — число рангов, которые повторяются соответственно у j -го и k -го экспертов.

Если у экспертов повторяющиеся ранги отсутствуют, выражение (П2.4) упрощается и принимает вид

$$\rho_{jk} = 1 - \frac{6 \sum_{i=1}^m (r_{ij} - r_{ik})^2}{m(m^2 - 1)}, \quad j = 1, \dots, z \text{ и } k = 1, \dots, z; \quad (\text{П2.5})$$

- на третьем этапе на основе значений, полученных с помощью выражений (П2.4) или (П2.5), составляем матрицу коэффициентов ρ_{ij} ранговой корреляции Спирмена;
- на четвертом этапе вычисляем суммарный коэффициент согласованности мнений каждого j -го эксперта ρ_j с остальными экспертами рабочей группы по формуле

$$\rho_j = \sum_{k=1}^z \rho_{jk}. \quad (\text{П2.6})$$

При этом значения ρ_j находятся в границах от $-(z-1)$ (полная несогласованность) до $+(z-1)$ (полная согласованность эксперта с остальными членами рабочей группы). При нормировании величины ρ_j относительно значения полной несогласованности экспертов получаем, что значение модифицированного суммарного коэффициента согласованности мнений каждого j -го эксперта ρ_j^{**} с остальными экспертами рабочей группы вычисляется по формуле

$$\rho_j^{**} = \rho_j + (z-1); \quad (\text{П2.7})$$

- на пятом этапе весовые коэффициенты ранговых оценок экспертов α_j^{**} рассчитываем с учетом значений ρ_j^{**}

$$\alpha_j^{**} = \rho_j^{**} \left/ \sum_{j=1}^z \rho_j^{**} \right.. \quad (\text{П2.8})$$

Третий подход предусматривает исключение из рассмотрения итогов работы группы результатов эксперта, имеющего наименьший коэффициент согласованности мнений с остальными экспертами рабочей группы, и состоит из последовательного выполнения нескольких этапов:

- на первом этапе для каждой пары экспертов проводим сравнение результатов ранжирования предложенных им вариантов и вычисляем коэффициент ρ_{jk} ранговой корреляции Спирмена, т. е. коэффициент согласованности их мнений по формуле (П2.4). Если у экспертов повторяющиеся ранги отсутствуют, следует использовать формулу (П2.5);
- на втором этапе для каждого эксперта вычисляем суммарный коэффициент согласованности его мнения ρ_j со всеми экспертами рабочей группы:

$$\rho_j = \sum_{k=1}^z \rho_{jk}; \quad (\text{П2.10})$$

- на третьем этапе определяем эксперта \mathcal{E}_w , имеющего наименьший коэффициент согласованности мнений ρ_w с остальными экспертами рабочей группы, и исключаем результаты его работы из результатов работы экспертной группы

$$\rho_w = \min_{j \in z} \rho_j. \quad (\text{П2.11})$$

Для каждого из трех приведенных подходов, направленных на повышение уровня согласованности мнений экспертов, формулы для обработки результатов экспертного опроса имеют следующий вид:

$$r = \frac{1}{m} \sum_{i=1}^m \sum_{j=1}^z r_{ij} = \frac{(m+1)z}{2}; \quad (\text{П2.12})$$

Три подхода для вычисления численности экспертов в экспертной группе

семинар 7

Существуют различные подходы к определению численности экспертов z в составе рабочей группы:

1) число экспертов должно быть не меньше числа альтернативных вариантов, подлежащих ранжированию: $z \geq m$;

2

2) число экспертов должно быть достаточным для принятия корректного решения и определяется по следующей формуле: $z > 0,5 (0,3/b + 5)$, где b — ошибка результата экспертного анализа. Так, при допустимой ошибке экспертного анализа в 5% ($b = 0,05$) в состав рабочей группы должно входить не менее шести экспертов;

3) согласно принципу Гештальта, число экспертов должно быть в пределах $z = 7 \pm 2$, т. е. не менее пяти и не более девяти человек. При большом числе экспертов сложно согласовать их мнения с учетом возможной принадлежности к разным научным школам и т. д.

коэффициент конкордации и его формула, если у экспертов не связанные факторы или что-то такое

тоже семинар 7

8) W — коэффициент конкордации, т. е. коэффициент согласованности мнений экспертов, определяемый по формуле Кендалла,

$$W = \frac{S}{S_{max}} = \frac{S}{\frac{1}{12} z^2 (m^3 - m) - z \sum_{j=1}^z T_j}.$$

Если связанных факторов у экспертов нет, то

$$W = \frac{12 \cdot S}{z^2 (m^3 - m)}.$$

Коэффициент конкордации используют для оценки степени согласованности мнений нескольких экспертов. Это — коэффициент ранговой корреляции для группы, состоящей из z экспертов. При этом W определяют как отношение фактически полученной величины S к ее максимальному значению S_{max} для одной и той же группы экспертов и числа факторов. При значении $W \geq 0,7$ считают, что мнения экспертов вполне согласованы;

Оценка значимости коэффициента конкордации по критерию согласия Пирсона

тоже семинар 7

9) оценку значимости коэффициента конкордации по критерию согласия Пирсона χ^2 .

При этом величина $Wz(m - 1)$ имеет χ^2 -распределение с $v = m - 1$ степенями свободы.

Если у экспертов имеются связанные факторы, расчетное значение χ^2 -распределения с $v = m - 1$ степенями свободы рассчитывают по следующей формуле:

$$\chi^2 = \frac{S}{\frac{1}{12} zm(m + 1) - \frac{1}{m - 1} \sum_{j=1}^z T_j}.$$

Если у экспертов отсутствуют связанные варианты, формула упрощается и принимает следующий вид:

$$\chi^2 = \frac{12S}{zm(m + 1)}.$$

Данные, знания, информация: определения, отличия

Данные — представление фактов и идей в формализованном виде, пригодном для передачи и обработки в некотором информационном процессе.

Наиболее общее определение трактует знание как всю совокупность данных (информации), необходимую для решения задачи.

В этом определении подчеркивается, что данные в привычном понимании также являются знаниями. Однако знания в информационном плане не ограничиваются рамками данных.