

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«ОРЕНБУРГСКИЙ ГОСУДАРСТВЕННЫЙ АГРАРНЫЙ УНИВЕРСИТЕТ»**

**Методические рекомендации для
самостоятельной работы обучающихся по дисциплине**

Б1.В.16 Системы искусственного интеллекта

**Направление подготовки (специальность) 09.03.01 Информатика и вычислительная
техника**

**Профиль образовательной программы “Автоматизированные системы обработки
информации и управления”**

Форма обучения очная

Содержание

1.	Организация самостоятельной работы.....	3
2.	Методические рекомендации	4
	по самостоятельному изучению вопросов	4
3.	МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ	16
	ПО ПОДГОТОВКЕ К ЗАНЯТИЯМ	16
3.1	Практическое занятие 1, 2, 3, 4. «Понятие искусственного интеллекта».....	16
3.2	Практическое занятие 5, 6, 7, 8. «Модели представления знаний в системах искусственного интеллекта».....	16
3.3	Практическое занятие 10, 11, 12. «ЭС и технологии ее разработки».....	16
3.4	Практическое занятие 13, 14, 15, 16, 17. «Биологические прототипы ИИ».....	17

1. Организация самостоятельной работы

1.1 Организационно-методические данные дисциплины

№ п.п.	Наименование темы	Общий объем часов по видам самостоятельной работы (из табл. 5.1 РПД)				
		подготовка курсового проекта (работы)	подготовка реферата/эссе	индивидуальные домашние задания (ИДЗ)	самостоятельное изучение вопросов (СИВ)	подготовка к занятиям (ПкЗ)
1	2	3	4	5	6	7
1	Понятие искусственного интеллекта				10	10
2	Модели представления знаний в системах искусственного интеллекта.				10	12
3	ЭС и технологии ее разработки				12	12
4	Биологические прототипы ИИ				12	12

2. Методические рекомендации

по самостояльному изучению вопросов

2.1 Функциональная структура использования СИИ

Функциональная структура СИИ.

Направления в моделировании ИИ.

Исторически сложились три базовых направления в моделировании ИИ.

В рамках первого подхода объектом исследований являются структура и механизмы работы мозга человека, а конечная цель заключается в раскрытии тайн мышления. Необходимыми этапами исследований в этом направлении являются построение моделей на основе психофизиологических данных, проведение экспериментов с ними, выдвижение новых гипотез относительно механизмов интеллектуальной деятельности, совершенствование моделей.

Второй подход в качестве объекта исследования рассматривает ИИ. Здесь речь идет о моделировании интеллектуальной деятельности с помощью вычислительных машин. Целью работ в этом направлении является создание алгоритмического и программного обеспечения вычислительных машин, позволяющего решать интеллектуальные задачи не хуже человека.

Третий подход ориентирован на создание смешанных человеко-машинных, или, как еще говорят, интерактивных интеллектуальных систем, на симбиоз возможностей естественного и искусственного интеллекта. Важнейшими проблемами в этих исследованиях является оптимальное распределение функций между естественным и искусственным интеллектом и организация диалога между человеком и машиной.

Эта структура состоит из трех комплексов вычислительных средств (см. рис. 1).

Первый комплекс представляет собой совокупность средств, выполняющих программы (исполнительную систему), спроектированных с позиций эффективного решения задач, имеет в ряде случаев проблемную ориентацию. Исполнительная система (ИС) объединяет всю совокупность средств, обеспечивающих выполнение сформированной программы.

Второй комплекс - совокупность средств интеллектуального интерфейса, имеющих гибкую структуру, которая обеспечивает возможность адаптации в широком спектре интересов конечных пользователей. Интеллектуальный интерфейс - система программных и аппаратных средств, обеспечивающих для конечного пользователя использование компьютера для решения задач, которые возникают в среде его профессиональной деятельности либо без посредников либо с незначительной их помощью.

Третьим комплексом средств, с помощью которых организуется взаимодействие первых двух, является база знаний, обеспечивающая использование вычислительными средствами первых двух комплексов целостной и независимой от обрабатывающих программ системы знаний о проблемной среде. База знаний (БЗ) - занимает центральное положение по отношению к остальным компонентам вычислительной системы в целом, через БЗ осуществляется интеграция средств ВС, участвующих в решении задач.

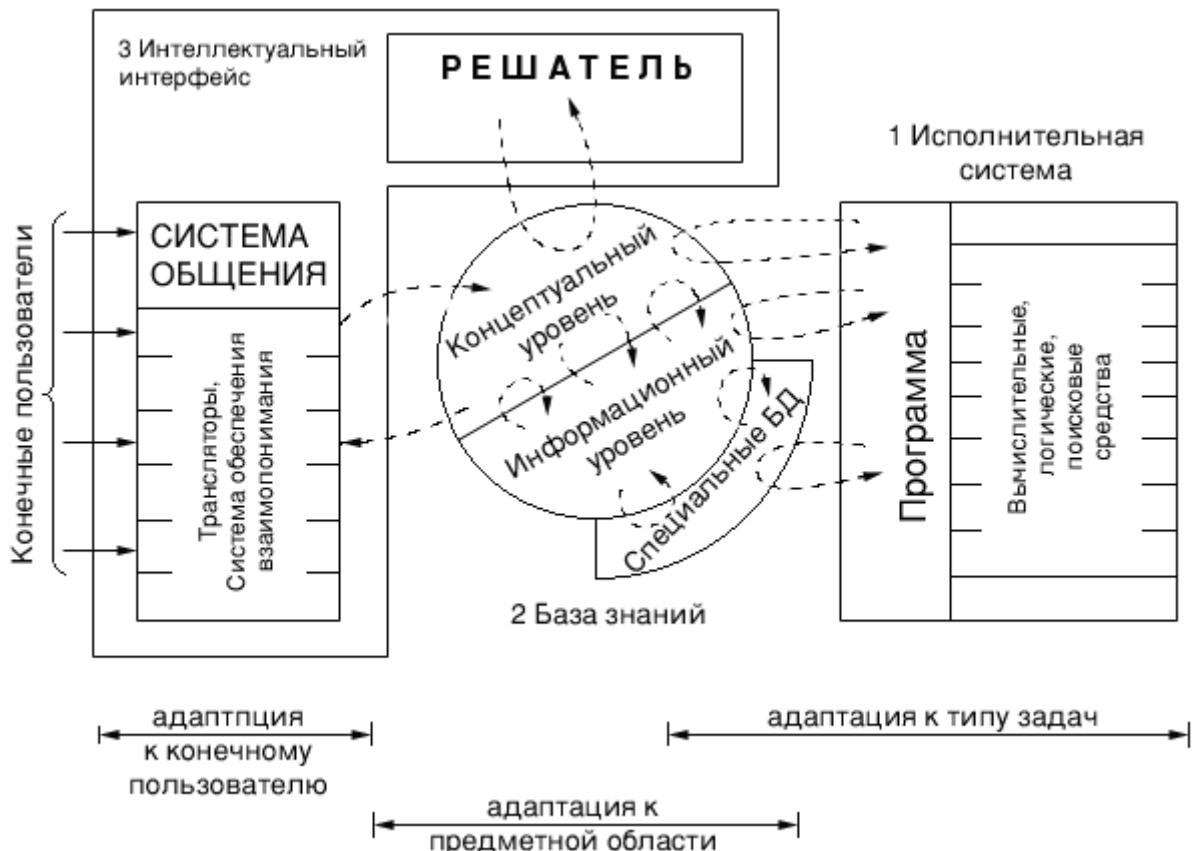


Рисунок 1 - Функциональная структура СИИ

2.2 Сценарии; ленемы. Базы знаний. Измерение БЗ

База знаний (БЗ; англ. knowledge base, KB) — база данных, содержащая правила вывода и информацию о человеческом опыте и знаниях в некоторой предметной области (ISO/IEC/IEEE 24765-2010^[1], ISO/IEC 2382-1:1993^[2]). В самообучающихся системах база знаний также содержит информацию, являющуюся результатом решения предыдущих задач^[1].

Современные базы знаний работают совместно с системами поиска и извлечения информации. Для этого требуется некоторая модель классификации понятий и определённый формат представления знаний. Иерархический способ представления в базе знаний набора понятий и их отношений называется онтологией.

Онтологию некоторой области знаний вместе со сведениями о свойствах конкретных объектов часто называют «базой знаний». Вместе с тем, полноценные базы знаний (в отличие от обычной базы данных) содержат в себе не только фактическую информацию, но и правила вывода, позволяющие делать автоматические умозаключения об уже имеющихся или вновь вводимых фактах и тем самым производить семантическую (осмысленную) обработку информации.

Область наук об искусственном интеллекте, изучающая базы знаний и методы работы со знаниями, называется инженерией знаний.

Применение.

База знаний — важный компонент интеллектуальной системы. Наиболее известный класс таких программ — это экспертные системы. Они предназначены для поиска способов решения проблем из некоторой предметной области, основываясь на записях БЗ и на пользовательском описании ситуации.

Простые базы знаний могут использоваться для создания экспертных систем хранения данных в организации: документации, руководств, статей технического обеспечения. Главная цель создания таких баз — помочь менее опытным людям найти уже существующее описание способа решения какой-либо проблемы.

Базы знаний и интеллектуальные системы

Двумя наиболее важными требованиями к информации, хранящейся в базе знаний интеллектуальной системы, являются:

- 1) Достоверность конкретных и обобщённых сведений, имеющихся в базе данных;
- 2) Релевантность информации, получаемой с помощью правил вывода базы знаний.

Ниже перечислены некоторые из особенностей, которые могут (но не обязаны) быть у системы, оперирующей базами знаний.

- Автоматическое доказательство (вывод). Способность системы выводить новые знания из старых, находить закономерности в БЗ. Часто принимается, что база знаний отличается от базы данных именно наличием механизма вывода.
- Доказательство заключения. Способность системы после выдачи ответа «объяснить» ход её рассуждений, причем «по первому требованию».
- Интроспекция. Нахождение противоречий, нестыковок в БЗ, контроль правильной организации БЗ.
- Машинное обучение. Превращение БЗ в гибкую систему, адаптация к проблемной области. Аналогична человеческой способности «набирать опыт».

Базы знаний на примере языка Пролог.

В языке Пролог базы знаний описываются в форме конкретных фактов и правил логического вывода над базами данных и процедурами обработки информации, представляющих сведения и знания о людях, предметах, фактах, событиях и процессах в логической форме. В ответах на простейшие запросы к базам знаний система логического программирования Пролог выдает значения «истина» и «ложь» в зависимости от наличия соответствующих фактов.

Обобщённые сведения в языке Пролог задаются с помощью правил логического вывода, выполняющих роль определения понятий, а также логических процедур, состоящих из наборов правил логического вывода. Достоверность обобщенных сведений зависит от наличия необходимых фактов и достоверности данных в базах знаний.

2.3 Средства формирования пояснений. Формирование пояснений на основе знаний. Подсистема формирования пояснений в MYCIN. Формирование пояснений на основе фреймов. Организация вывода пояснений в системе CENTAUR

На начальном этапе исследований в области экспертных систем, которые выполнялись в Стэнфордском университете в 60-70-х годах/поясняющая информация предоставлялась в виде трассировки процесса выполнения программы и использовалась в основном для отладки разрабатываемых систем. Этого было достаточно для разработчиков экспериментальных систем, подобных MYCIN, но не соответствовало тому уровню сервиса пользователя, который необходим для коммерческого программного продукта. Впоследствии вопросу формирования информации, которая давала бы возможность пользователю четко представить себе ход рассуждений программы, стало уделяться значительно больше внимания.

Исследователи пришли к заключению, что автоматическое формирование пояснений требует доступа к модели предметной области точно так же, как и извлечение знаний.

Другими словами, представление о знаниях в конкретной области необходимо для предоставления пользователю информации о поведении системы в процессе формирования результата точно так же, как и для приобретения новых знаний. Такое знание позволит перекинуть мост между деталями реализации процесса вывода (например, в какой последовательности активизировались правила) и стратегией поведения системы (например, какие соображения побудили систему выбрать ту или иную гипотезу из множества конкурирующих). В последние десятилетия специалисты серьезно потрудились над развитием этой идеи, и обзор некоторых из полученных результатов читатель найдет в разделе 16.2. Совершенно очевидно, что проблемы извлечения знаний и формирования пояснений тесно связаны. По сути, они представляют две стороны одной медали. Существенным толчком для совершенствования средств, используемых для предоставления пользователю пояснений, как, впрочем, и для извлечения знаний, стало развитие методов графического интерфейса в современных операционных системах, которые обеспечивают возможность вывода не только статической, но и динамической видеинформации со звуковым сопровождением.

Формирование пояснений на основе знаний. Подсистема формирования пояснений в MYCIN

Модуль формирования пояснений в экспертной системе MYCIN (о ней шла речь в главе 3) автоматически вызывается по завершении каждого сеанса консультаций. Для того чтобы предоставить пользователю информацию о том, почему система рекомендует именно такие значения медицинских параметров, модуль извлекает список правил, активизированных в процессе сеанса, и распечатывает его вместе с заключениями, которые были сделаны этими правилами. Модуль также позволяет пользователю задавать вопросы системе по поводу выполненной консультации, причем вопросы могут носить и обобщенный характер.

Способность системы отвечать на вопросы пользователя, касающиеся выполненной работы, основывается на следующих функциях:

вывод на экран правил, активизированных на любой стадии консультации; запись и сохранение в процессе работы активизированных правил и связанных с ними событий, например задаваемых вопросов и сформированных заключений; использование индексации правил, которая дает возможность извлечь определенное правило в ответ на вопрос, содержащийся в пользовательском запросе.

Как отмечалось в главе 3, процесс выполнения консультаций в экспертной системе, использующей обратную цепочку логического вывода, включает поиск в дереве целей (рис. 2). Следовательно, справки о ходе выполнения консультации можно разделить на два типа:

почему система сочла необходимым задать пользователю определенный вопрос; как система пришла к определенному заключению.

Чтобы ответить на вопрос почему, нужно просмотреть дерево целей "вверх" и определить, какую цель более высокого уровня пытается достичь система. Чтобы ответить на вопрос как, нужно просмотреть дерево "вниз" и выяснить, достижение каких подцелей привело к данной цели (в текущее состояние). Таким образом, процесс формирования пояснений можно рассматривать как некоторый вид прослеживания дерева целей, т.е. свести к задаче поиска в дереве.

Тот факт, что MYCIN отслеживает прохождение структур цель-подцель в процессе выполнения вычислений, позволяет этой системе отвечать на вопросы вроде приведенного ниже.

"Почему вас интересует, является ли окраска микроорганизма грамотрицательной?"

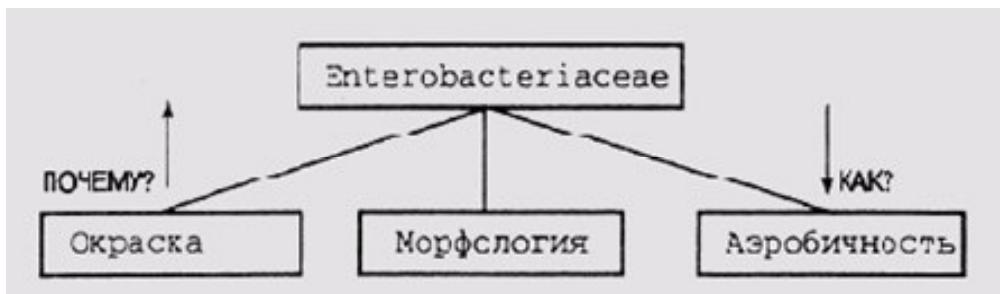


Рис. 2. Формирование ответов на основе дерева целей в системе MYCIN

В ответ система может процитировать правило, которое утверждает, что если микроорганизм имеет грамотрицательную окраску и соблюдаются другие условия, то этот организм относится к классу enterobacteriaceae. Помимо цитирования правила, в ответе также указывается, что текущая цель — выяснение класса микроорганизма.

В MYCIN сохраняется список всех решений, принятых в течение сеанса работы, а затем этот список используется для пояснения и уточнения принятых решений в ответ на вопрос пользователя почему, например, такой:

"Почему вы предполагаете, что Организм-1 это протеин?"

В ответ MYCIN процитирует правило, относящееся к этому заключению, и выведет значение степени уверенности в достоверности этого правила. Системе MYCIN можно также задавать вопросы общего характера. Такие вопросы касаются правил безотносительно к текущему состоянию базы данных, т.е. безотносительно к конкретному пациенту. Например:

"Что прописывать при заражении инфекцией pseudonomas?"

В ответ система выведет тот набор препаратов, который рекомендуется в правилах, касающихся инфекции pseudonomas.

Однако пользователь не может получить доступ к информации, хранящейся в виде статических знаний или в таблицах знаний, поскольку эти источники данных не имеют формата порождающих правил. Кроме того, механизм формирования рекомендаций о курсе лечения и выборе предпочтительных медикаментов представлен в программе в виде LISP-функций, которые пользователь не может просмотреть (а если и сможет, то вряд ли что-нибудь в них поймет). Пользователь также не может задать системе вопросы о том, в каком порядке просматриваются правила в процессе формирования решения и в каком порядке анализируются условия, специфицированные в этих правилах.

Подытоживая сказанное о средствах формирования пояснений в системе MYCIN, отметим, что хотя вопросы как и почему и дают пользователю определенную информацию о порядке логического вывода, но признать ее достаточной можно с большими оговорками. Пользователю трудно проследить за логикой процесса по длинной распечатке активизированных правил. Правда, такая распечатка содержит довольно ценную информацию для инженеров по знаниям и разработчиков системы, особенно на стадии ее настройки.

В тех экспертных системах, в которых используется прямая цепочка рассуждений, список активизированных правил несет еще меньше полезной для пользователя информации, поскольку на промежуточной стадии вычислений трудно по нему судить, куда же ведет цепочка рассуждений.

Формирование пояснений на основе фреймов

Вы уже имели возможность убедиться в том, что сформировать достаточно информативные пояснения, пользуясь только результатами трассировки активизированных правил, довольно сложно. Информация, содержащаяся в правилах, не

несет достаточных сведений о контексте, в котором эти правила были активизированы. Фактически мы узнаем только о выполнении тех условий, которые явно специфицированы в активизированном правиле. Отсутствуют сведения о том, для какой задачи более высокого уровня потребовалось активизировать то или иное правило. Например, желательно знать, потребовалось ли это для подтверждения какой-либо гипотезы, или для принятия решения о выборе между конкурирующими гипотезами, или для запроса данных у пользователя и т.п.

В результате исследователи и разработчики экспертных систем в 1980-х годах задумались над тем, как объединить порождающие правила с фреймами, которые могли бы предоставить необходимую информацию о контексте. Во фреймах могут быть представлены важные отношения между данными и гипотезами, которые не нашли отражения в порождающих правилах. Кроме того, фреймы могут служить удобными контейнерами для поясняющих текстов. Когда в начале 1990-х годов в распоряжении программистов появились мультимедийные средства, фреймы стали использоваться для хранения не только поясняющих текстов, но и ссылок на сопутствующие изображения, например схемы, или даже на целые руководства в электронном виде.

Организация вывода пояснений в системе CENTAUR

Первый вариант реализации системы PUFF, который был выполнен на основе оболочки EMYCIN (см., например, [Aikins et al., 1984]), оказался вполне работоспособным в том смысле, что хорошоправлялся с решением проблем в своей области, но схема представления знаний в нем не совсем удовлетворила разработчиков по следующим причинам.

С ее помощью трудно представлять типовые образцы данных или типичные классы пациентов.

Добавление или модификация правил при расширении базы знаний или уточнении ранее введенных знаний требовали слишком больших затрат.

Изменение порядка, в котором информация запрашивается у пользователя, также было связано с определенными сложностями, поскольку запросы формируются интерпретатором автоматически, как только активизируются правила.

Особую проблему представляет формирование ясных пояснений, поскольку практически для выполнения этой задачи доступна информация, лишь незначительно более обширная, чем просто результаты трассировки последовательности активизации правил.

Эйкинс (Aikins) также обратила внимание на то, что отчетливо выраженная модульность и единообразие порождающих правил имеет и обратную сторону. Большинство наборов правил обладает неявно выраженной группировкой, которая существует либо в виде определенного порядка индексации, скрытой в интерпретаторе (например, в наборе ORGRULES и PATRULES системы MYCIN, которые относятся к микроорганизмам и пациентам соответственно), либо в виде условий и операций, которые манипулируют лексемами целей в рабочей памяти. Такая организация зачастую имеет иерархический характер, предполагающий таксономический характер организации гипотез (в задачах классификации) и декомпозицию целей на подцели (в задачах конструирования). Многие из упомянутых выше проблем можно свести к минимуму, выделив отдельные виды знаний и манипулируя ими по-разному. Как вы помните из главы 13, в системе CENTAUR разработчики объединили методы программирования, основанные на правилах и концепции фреймов, таким образом, чтобы компенсировать слабости каждого из подходов и усилить их достоинства.

Что касается формирования пояснений, то в системе CENTAUR сделан акцент на контексте, в котором формировалось суждение, и на зависимости между применяемой экспертом методикой и этапом процесса решения проблемы. Для того чтобы понять, почему задан определенный вопрос, нужно принимать во внимание не только то правило,

которое при этом активизировано, но и ту гипотезу, которая анализируется на данном этапе. Таким образом, Эйкинс вполне разделяет опасения, высказанные Кленси, хотя ее подход и менее амбициозен, поскольку в нем дело не дошло до когнитивного моделирования.

Работа консультирующей программы CENTAUR состоит в основном из выполнения интерпретатором текущего списка актуальных задач, кроме того, в ней значительно меньше места отводится построению цепочки правил. Основное назначение списка актуальных задач — способствовать формированию пояснений, почему система поступает именно так в ходе данного сеанса консультаций. Поэтому каждый элемент в списке задач содержит информацию, как об источнике задачи, так и о причине, которая побудила систему включить данную задачу в список актуальных.

Источником задачи может быть активный прототип (т.е. тот, который наиболее близок к специфическим данным рассматриваемого случая) или другая задача. Новые задачи добавляются в список актуальных либо слотами управления прототипов, либо в процессе выполнения тех задач, которые ранее были включены в список актуальных. Информация о причине добавления генерируется на основе наименования прототипа и имени управляющего слота, ответственного за включение задачи.

В ходе выполнения программы прототип находится в одном из трех возможных состояний:

неактивный, т.е. не рассматриваемый в качестве гипотезы;
потенциальный кандидат, т.е. такой, который имеет смысл рассмотреть исходя из имеющихся значений данных;
активный, т.е. выбранный из множества потенциальных кандидатов и помещенный в список гипотез.

Прототипы заболеваний представляют гипотезы! Список гипотез — это, по сути, список пар "прототип-коэффициент уверенности", упорядоченных в убывающем порядке значений коэффициентов уверенности. Имеются еще два списка, которые служат для отслеживания подтвержденных и отвергнутых прототипов.

Основные события в сеансе выполнения консультации с помощью системы CENTAUR следующие:

ввод исходных данных;
отбор прототипов с использованием антецедентных правил;
оценка прототипов и выбор единственного, который назначается "текущим";
анализ известных фактов и заполнение на их основе полей данных текущего прототипа;
проверка соответствия фактов и ожидаемых значений;
выявление данных, не учтенных начальным вариантом диагноза;
уточнение диагноза на основе выявленных данных;
суммирование и вывод результатов.

В самом начале сеанса в качестве текущего выбирается прототип CONSULTATION (консультация), а в список активных включаются две задачи текущего прототипа: FILL-IN (заполнение) и CONFIRM (подтверждение), которые извлекаются из управляющих слотов TO-FILL-IN и IF-CONFIRMED прототипа. Структура слотов прототипа CONSULTATION представлена ниже.

CONSULTATION

.....

TO-FILL-IN:

Запросить значение

TRACING-LEVEL для задачи
CONSULTATKDN Запросить значение
AGENTA-PRINTING для задачи
CONSULTATION Запросить значение
STRATEGY для задачи
CONSULTATION

IF-CONFIRMED:

Установить порог подтверждения равным 0.

Установить относительное заполнение слотов, необходимое для подтверждения прототипа, равным 0.75

Установить процедуру по умолчанию для заполнения слотов: заполнение в убывающем порядке по степени важности Определить предмет консультации Выбрать лучший из текущих прототипов Заполнить прототип

Применить задачи из слота IF-CONFIRMED прототипа Отметить все факты, принимаемые во внимание прототипом Применить правила уточнения, связанные с подтвержденными прототипами; применить правила подведения итога, связанные с подтвержденными прототипами; выполнить операции, связанные с подтвержденными прототипами

Слот TO-FILL-IN прототипа фактически содержит три подзадачи, каждая из которых устанавливает определенную переменную сеанса консультаций: переменная TRACING-LEVEL задает уровень детализации трассировки, переменная AGENTA-PRINTING указывает, будут ли выводиться на печать наименования задач по мере включения их в список активных или по мере выполнения, а переменная STRATEGY может принимать значения CONFIRMATION (выбор наилучшего варианта и подтверждение его), или ELIMINATION (выбор наихудшего варианта и исключение его), или FIXED-ORDER (использование предопределенного порядка обработки гипотез).

Первые три задачи слота IF-CONFIRMED устанавливают значения переменных, управляющих процессом консультации. В частности, одна из переменных определяет, какая часть полей прототипа должна быть заполнена прежде, чем его можно будет считать подтвержденным. Наличие таких переменных позволяет инженеру по знаниям экспериментировать с разными режимами управления процессом и, возможно, настраивать работу системы в разных предметных областях. Остальные задачи слота IF-CONFIRMED управляют отдельными этапами проведения консультации.

После того как будет определен предмет консультации (в настоящем контексте это область легочных заболеваний), следующим текущим прототипом становится PULMONARY-DISEASE. Первой задачей, специфицированной в этом прототипе, является запрос данных у пользователя. Протокол такого диалога представлен ниже (в переводе на русский язык). Нумерованная строка в протоколе — запрос системы, а текст после двух звездочек, выделенный полужирным шрифтом, — ответ пользователя. В квадратных скобках приведены комментарии — сообщения, которыми система дает знать пользователю, что она отыскала соответствующий прототип в базе знаний и в дальнейшем будет рассматривать его в качестве кандидата для более подробного анализа.

Значение меры правдоподобия той или иной гипотезы выбирается в диапазоне от -1000 до 1000 исключительно из соображений упрощения вычислений. Этот параметр отражает степень уверенности системы в обоснованности выдвинутой (активизированной) гипотезы на основе имеющихся данных о конкретной истории болезни. Фактически при определении меры правдоподобия система сравнивает введенные пользователем данные с теми, которые хранятся в слотах прототипа-кандидата. Полученные значения служат основанием для выбора самой правдоподобной из имеющихся гипотез (прототипов). Назначение параметра "мера правдоподобия" в системе CENTAUR такое же, как и коэффициента уверенности в системах MYCIN и EMYCIN, причем для операций с мерами

правдоподобия используются те же алгоритмы, что и для операций с коэффициентами уверенности. Обратите внимание — в процессе диалога с пользователем система не объясняет, почему выбрано именно такое, а не иное значение меры правдоподобия. Для пользователя алгоритм вычисления меры правдоподобия является "черным ящиком". В экспертных системах, полностью основанных на правилах, в протоколе трассировки обычно выводятся только те исходные данные, которые активизируют правило, получившее наивысшую оценку при разрешении конфликта. Пользователю в такой ситуации остается только гадать, как система отреагировала на данные, которые были введены, но не упоминаются в протоколе. Как видно в приведенном выше протоколе диалога с пользователем, программа CENTAUR сразу же дает знать пользователю, какие предварительные соображения вызвали у нее введенные значения отдельных параметров.

2.4 Решение задач распознавания образов. Нейрокомпьютерная сеть Кохонена.

Теория распознавания образа — раздел информатики и смежных дисциплин, развивающий основы и методы классификации и идентификации предметов, явлений, процессов, сигналов, ситуаций и т. п. объектов, которые характеризуются конечным набором некоторых свойств и признаков. Такие задачи решаются довольно часто, например, при переходе или проезде улицы по сигналам светофора. Распознавание цвета загоревшейся лампы светофора и знание правил дорожного движения позволяет принять правильное решение о том, можно или нельзя переходить улицу.

Необходимость в таком распознавании возникает в самых разных областях — от военного дела и систем безопасности до оцифровки аналоговых сигналов.

Проблема распознавания образа приобрела выдающееся значение в условиях информационных перегрузок, когда человек не справляется с линейно-последовательным пониманием поступающих к нему сообщений, в результате чего его мозг переключается на режим одновременности восприятия и мышления, которому такое распознавание свойственно.

Неслучайно, таким образом, проблема распознавания образа оказалась в поле междисциплинарных исследований - в том числе в связи с работой по созданию искусственного интеллекта, а создание технических систем распознавания образа привлекает к себе всё большее внимание.

Направление в распознавании образов

Можно выделить два основных направления:

- Изучение способностей к распознаванию, которыми обладают живые существа, объяснение и моделирование их;
- Развитие теории и методов построения устройств, предназначенных для решения отдельных задач в прикладных целях.

Формальная постановка задачи

Распознавание образов — это отнесение исходных данных к определенному классу с помощью выделения существенных признаков, характеризующих эти данные, из общей массы несущественных данных.

При постановке задач распознавания стараются пользоваться математическим языком, стремясь - в отличие от теории искусственных нейронных сетей, где основой является получение результата путём эксперимента, - заменить эксперимент логическими рассуждениями и математическими доказательствами.

Классическая постановка задачи распознавания образов^[4]: Дано множество объектов. Относительно них необходимо провести классификацию. Множество представлено подмножествами, которые называются классами. Заданы: информация о классах, описание всего множества и описание информации об объекте, принадлежность которого к определенному классу неизвестна. Требуется по имеющейся информации о классах и описании объекта установить - к какому классу относится этот объект.

Наиболее часто в задачах распознавания образов рассматриваются монохромные изображения, что дает возможность рассматривать изображение как функцию на

плоскости. Если рассмотреть точечное множество на плоскости $f(x, y)$, где функция выражает в каждой точке изображения его характеристику — яркость, прозрачность, оптическую плотность, то такая функция есть формальная запись изображения.

Множество же всех возможных функций на плоскости — есть модель множества всех изображений. Вводя понятие сходства между образами можно поставить задачу распознавания. Конкретный вид такой постановки сильно зависит от последующих этапов при распознавании в соответствии с тем или иным подходом.

Некоторые методы распознавания графических образов.

Для оптического распознавания образов можно применить метод перебора вида объекта под различными углами, масштабами, смещениями и т. д. Для букв нужно перебирать шрифт, свойства шрифта и т. д.

Второй подход — найти контур объекта и исследовать его свойства (связность, наличие углов и т. д.)

Ещё один подход — использовать искусственные нейронные сети. Этот метод требует либо большого количества примеров задачи распознавания (с правильными ответами), либо специальной структуры нейронной сети, учитывающей специфику данной задачи.

Персептрон как метод распознавания образов.

Ф. Розенблatt, вводя понятие о модели мозга, задача которой состоит в том, чтобы показать, как в некоторой физической системе, структура и функциональные свойства которой известны, могут возникать психологические явления, описал простейшие эксперименты по различению. Данные эксперименты целиком относятся к методам распознавания образов, но отличаются тем, что алгоритм решения не детерминированный.

Простейший эксперимент, на основе которого можно получить психологически значимую информацию о некоторой системе, сводится к тому, что модели предъявляются два различных стимула и требуется, чтобы она реагировала на них различным образом. Целью такого эксперимента может быть исследование возможности их спонтанного различения системой при отсутствии вмешательства со стороны экспериментатора, или, наоборот, изучение принудительного различения, при котором экспериментатор стремится обучить систему проводить требуемую классификацию.

В опыте с обучением персептрону обычно предъявляется некоторая последовательность образов, в которую входят представители каждого из классов, подлежащих различию. В соответствии с некоторым правилом модификации памяти правильный выбор реакции подкрепляется. Затем персептрону предъявляется контрольный стимул и определяется вероятность получения правильной реакции для стимулов данного класса. В зависимости от того, совпадает или не совпадает выбранный контрольный стимул с одним из образов, которые использовались в обучающей последовательности, получают различные результаты:

1. Если контрольный стимул не совпадает ни с одним из обучающих стимулов, то эксперимент связан не только с чистым различием, но включает в себя и элементы обобщения.

2. Если контрольный стимул возбуждает некоторый набор сенсорных элементов, совершенно отличных от тех элементов, которые активизировались при воздействии ранее предъявленных стимулов того же класса, то эксперимент является исследованием чистого обобщения.

Персептроны не обладают способностью к чистому обобщению, но они вполне удовлетворительно функционируют в экспериментах по различению, особенно если контрольный стимул достаточно близко совпадает с одним из образов, относительно которых персептрон уже накопил определенный опыт.

Примеры задач распознавания образов.

- Оптическое распознавание символов
- Распознавание штрих-кодов
- Распознавание автомобильных номеров
- Распознавание лиц
- Распознавание речи
- Распознавание изображений
- Распознавание локальных участков земной коры, в которых находятся месторождения полезных ископаемых
- Классификация документов

Нейронная сеть Кохонена.

Нейронные сети Кохонена — класс нейронных сетей, основным элементом которых является слой Кохонена. Слой Кохонена состоит из адаптивных линейных сумматоров («линейных формальных нейронов»). Как правило, выходные сигналы слоя Кохонена обрабатываются по правилу «Победитель получает всё»: наибольший сигнал превращается в единичный, остальные обращаются в ноль.

По способам настройки входных весов сумматоров и по решаемым задачам различают много разновидностей сетей Кохонена. Наиболее известные из них:

- сети векторного квантования сигналов, тесно связанные с простейшим базовым алгоритмом кластерного анализа (метод динамических ядер или К-средних);
- самоорганизующиеся карты Кохонена (англ. self-organising maps, SOM);
- сети векторного квантования, обучаемые с учителем (англ. learning vector quantization).

Сети (слои) Кохонена (Kohonen T.) относятся к самоорганизующимся нейронным сетям. Самоорганизующаяся сеть позволяет выявлять кластеры (группы) входных векторов, обладающих некоторыми общими свойствами. Кластеризация — это разделение исследуемого множества объектов на группы "похожих" объектов, называемых кластерами. Синонимами термина "кластер" (англ. Cluster — сгусток, пучок, группа) являются термины класс, таксон, сгущение. Задача кластеризации принципиально отличается от задачи классификации. Решением задачи классификации является отнесение каждого из объектов к одному из заранее определенных классов. В задаче кластеризации происходит отнесение объекта к одному из заранее неопределенных классов. Разбиение объектов по кластерам осуществляется при одновременном формировании кластеров. Кластеризация позволяет сгруппировать сходные данные, что облегчает решение ряда задач Data Mining :

- Изучение данных, облегчение анализа. Содержательный анализ полученных кластеров позволяет обнаружить закономерности. Например, можно выявить группы клиентов сети сотовой связи, для которых можно предложить новый тарифный план. Другие примеры — выявление групп покупателей торговой сети,

сегментация рынка. Анализ содержания кластера позволяет применить к объектам различных кластеров разные методы анализа.

- Прогнозирование. Относя новый объект к одному из кластеров, можно прогнозировать поведение объекта, поскольку его поведение будет схожим с поведением объектов кластера.
- Обнаружение аномалий. Содержательный анализ кластеров помогает выявить аномалии. Обычно, это кластеры, в которые попадает мало объектов.

Важно отметить роль содержательной интерпретации каждого кластера. Каждому кластеру необходимо присвоить содержательное название, отражающее суть объектов кластера. Для этого необходимо выявить, признаки, объединяющие объекты в кластер. Это может потребовать статистического анализа свойств объекта кластера. С помощью сетей Кохонена производится кластеризация объектов, описываемых количественными характеристиками. Формально задача кластеризации описывается следующим образом . Дано множество объектов $I = \{i_1, i_2, \dots, i_n\}$, каждый из которых характеризуется вектором x_i , $i = 1, 2, \dots, n$ атрибутов (параметров): $x_i = \{x_{i1}, x_{i2}, \dots, x_{im}\}$. Требуется построить множество кластеров C и отображение F множества I на множество C , то есть $F : I \rightarrow C$. Задача кластеризации состоит в построении множества $C = \{c_1, c_2, \dots, c_k\}$, где c_k — кластер, содержащий "похожие" объекты из множества I :

$ck = \{i_j \mid i_j \in I \wedge d(i_j, c_k) < s\}$, (1) s — величина, определяющая меру близости для включения объектов в один кластер, $d(i_j, c_k)$ — мера близости между объектами, называемая расстоянием. Если расстояние $d(i_j, c_k)$ меньше некоторого значения s , то объекты считаются близкими и помещаются в один кластер. В противном случае считается, что объекты отличны друг от друга и их помещают в разные кластеры. Условие (1) известно как гипотеза компактности. Кластеризация основана на использовании расстояния между векторами [3, 5]. Неотрицательное число $d(x, y)$ называется расстоянием (метрикой) между векторами x и y , если выполняются следующие условия: 1. $d(x, y) \geq 0$ для всех x и y . 2. $d(x, y) = 0$, тогда и только тогда, когда $x = y$. 3. $d(x, y) = d(y, x)$. 4. $d(x, y) \leq d(x, z) + d(z, y)$ — неравенство треугольника. В сетях Кохонена обычно применяется евклидово расстояние $(x, y) = \sqrt{\sum_{i=1}^m (x_i - y_i)^2}$. Евклидово расстояние между векторами x и y представляет собой евклидову норму разности векторов, или длину отрезка, соединяющего точки x и y . Евклидово расстояние является частным случаем расстояния Минковского (H. Minkowski) $(x, y) = \sqrt[m]{\sum_{i=1}^m |x_i - y_i|^p}$, где $p \geq 1$ — параметр, определяющий тип нормы вектора x .

Другой частный случай — 1-норма, которая называется манхэттенским расстоянием (расстоянием городских кварталов) $(x, y) = \sum_{i=1}^m |x_i - y_i|$. Манхэттенское расстояние — это расстояние, которое мы проходим, двигаясь параллельно осям координат, как в Манхэттене или других городах с прямоугольной продольно-поперечной планировкой улиц. Известны и другие виды расстояний [5]. Сеть (слой) Кохонена (рис. 1) — это однослочная сеть, построенная из нейронов типа WTA (Winner Takes All — победитель получает все) — см. раздел 2.6. Рис. 1. Структура сети Кохонена Каждый нейрон сети соединен со всеми компонентами m -мерного входного вектора $x_i = (x_{i1}, x_{i2}, \dots, x_{im})$. Входной вектор — это описание одного из объектов, подлежащих кластеризации. Количество нейронов совпадает с количеством кластеров, которое должна выделить сеть. В качестве нейронов сети Кохонена применяются линейные взвешенные сумматоры $y_i = \sum_{j=1}^m w_{ij} x_j$, где j — номер нейрона, i — номер входа, w_{ij} — вес i -го входа j -го нейрона, w_{ij} — порог. Каждый j -ый нейрон описывается вектором весов $w_j = (w_{j1}, w_{j2}, \dots, w_{jm})$, где m — число компонентов входных векторов. С выходов аддитивных сумматоров сигнал поступает на функцию конкуренции,ирующую по правилу "победитель получает всё". Функция конкуренции находит выход аддитивный сумматор с максимальным значением выхода. Пусть k — номер такого сумматора. Тогда на выходе сети формируется выходной

сигнал $1 \text{ k } y =$, остальные выходные сигналы равны нулю. Если максимум достигается одновременно на выходах нескольких сумматоров, то выходной сигнал, равный единице, соответствует одному из них, например, первому.

3. МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ПО ПОДГОТОВКЕ К ЗАНЯТИЯМ

3.1 Практическое занятие 1, 2, 3, 4. «Понятие искусственного интеллекта».

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

1. 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.
2. Согласно второй точки зрения это связано с новыми идеями решения задач на ЭВМ, с разработкой новых технологий программирования и с переходом к компьютерам не фон-неймановской архитектуры.
3. 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.

3.2 Практическое занятие 5, 6, 7, 8. «Модели представления знаний в системах искусственного интеллекта».

В отличие от данных знания обладают следующими свойствами:

- 1) Внутренней интерпретируемостью — вместе с информацией в БЗ представлены информационные структуры, позволяющие не только хранить знания, но и использовать их;
- 2) Структурированностью — выполняется декомпозиция сложных объектов на более простые и установление связей между ними;
- 3) Связанностью — отражаются закономерности относительно фактов, процессов, явлений и причинно-следственные отношения между ними;
- 4) Активностью — знания предполагают целенаправленное использование информации, способность управлять информационными процессами по решению определенных задач.

3.3 Практическое занятие 10, 11, 12. «ЭС и технологии ее разработки».

Три наиболее часто используемые и популярные на сегодняшний день модели представления знаний: производственные модели – модели основанные на правилах, позволяют представить знание в виде предложений типа: «ЕСЛИ условие, ТО действие». Производственные модели обладают тем недостатком, что при накоплении достаточно большого числа правил, они начинают противоречить друг другу; сетевые модели или семантические сети – как правило, это граф, отображающий смысл целостного образа. Узлы графа соответствуют понятиям и объектам, а дуги – отношениям между объектами; фреймовые модели – основываются на таком понятии как фрейм (англ. frame – рамка, каркас). Фрейм – структура данных для представления некоторого концептуального объекта. Информация, относящаяся к фрейму, содержится в составляющих его слотах. Слоты могут быть терминальными либо являться сами фреймами, т.о. образуя целую иерархическую сеть.

3.4 Практическое занятие 13, 14, 15, 16, 17. «Биологические прототипы ИИ».

Фреймовая модель основана на концепции Марвина Мински (Marvin Minsky) – профессора Массачусетского технологического института, основателя лаборатории искусственного интеллекта, автора ряда фундаментальных работ. Фреймовая модель представляет собой систематизированную психологическую модель памяти человека и его сознания.