

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

**МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ
ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ**

Б1.В.08 Нейрокомпьютерные системы

Направление подготовки 27.03.04 Управление в технических системах

Профиль образовательной программы “ Информационные управляющие комплексы систем безопасности объектов”

Квалификация (степень) выпускника бакалавр

Форма обучения заочная

СОДЕРЖАНИЕ

1. Конспект лекций	3
1.1 Лекция №1 Основные понятия нейрокомпьютерных систем.....	3
1.2 Лекция №2 Многослойный персептрон. Обучение многослойного персептрона методом обратного распространения ошибки	13
2. Методические указания по выполнению лабораторных работ.....	41
2.1. Лабораторная работа ЛР-1 Основные понятия нейрокомпьютерных систем.....	41
2.2. Лабораторная работа № ЛР-2 Модели и архитектура нейрокомпьютерных систем.....	48
2.3. Лабораторная работа № ЛР-3 Однослойный персептрон. Обучение однослойного персептрона по дельта-правилу.....	53
2.4. Лабораторная работа № ЛР-4 Многослойный персептрон. Обучение многослойного персептрона методом обратного распространения ошибки.....	58
2.5. Лабораторная работа № ЛР-5 Самоорганизующиеся обучаемые системы Синапс Хебба и его свойства. Обучение нейрона по алгоритму Хебба.....	61
2.6. Лабораторная работа № ЛР-6 Нейросети Хемминга и Хопфилда.....	67

1. КОНСПЕКТ ЛЕКЦИЙ

1.1. Лекция № 1 (2 часа).

Тема: «Основные понятия нейрокомпьютерных систем»

1.1.1. Вопросы лекции:

1. Понятие нейрокомпьютерной системы.
2. Понятие ИИ.
3. Основные направления развития ИИ.
4. Основные понятия ИИС.
5. Классификация ИИС.

1.1.2. Краткое содержание вопросов:

1. Понятие нейрокомпьютерной системы.

В последние десятилетия в мире бурно развивается новое направление ИИ, специализирующаяся на искусственных нейронных сетях (НС).

Нейрокомпьютеры являются предметом исследований сразу нескольких дисциплин, поэтому единое определение нейрокомпьютера можно дать только с учетом различных точек зрения, адекватных разным направлениям науки.

Математическая статистика. Нейрокомпьютеры - это системы, позволяющие сформировать описания характеристик случайных процессов и совокупности случайных процессов, имеющих в отличие от общепринятого, сложные, зачастую неизвестные функции распределения.

Математическая логика и теория автоматов. Нейрокомпьютеры - это системы, в которых алгоритм решения задачи представлен логической сетью элементов частного вида - нейронов с полным отказом от булевских элементов типа И, ИЛИ, НЕ.

Теория управления. В качестве объекта управления выбирается частный случай, хорошо формализуемый объект - многослойная нейронная сеть, а динамический процесс ее настройки представляет собой процесс решения задачи.

Вычислительная математика. В отличие от классических методов решения задач нейрокомпьютеры реализуют алгоритмы решения задач, представленные в виде нейронных сетей.

Что позволяет разрабатывать алгоритмы, потенциально более параллельные, чем любая другая их физическая реализация. Множество нейросетевых алгоритмов решения задач составляет новый перспективный раздел вычислительной математики, условно называемый нейроматематикой.

Вычислительная техника. Нейрокомпьютер - это вычислительная система с архитектурой MSIMD, в которой реализованы два принципиальных технических решения:

- Упрощен до уровня нейрона процессорный элемент однородной структуры и резко усложнены связи между элементами;
- Программирование вычислительной структуры перенесено на изменение весовых связей между процессорными элементами.

Общее определение: Нейрокомпьютерная система - это вычислительная система с архитектурой аппаратного и программного обеспечения, адекватной выполнению алгоритмов, представленных в нейросетевом логическом базисе.

Таким образом нейрокомпьютерная система может быть представлена моделями как программного, так и аппаратного исполнения.

В основе такой системы лежит понятие нейронной сети (НС), которая позволяет решать => виды задач:

автоматизация процессов распознавания образов	адаптивное управление	аппроксимация функционалов	прогнозирование	создание ЭС	другие
---	-----------------------	----------------------------	-----------------	-------------	--------

Например, с помощью НС можно:

- Предсказывать показатели биржевого рынка,

- Выполнять распознавание оптических или звуковых сигналов,
- Создавать самообучающиеся системы, способные управлять автомашиной при парковке или синтезировать речь по тексту.

Определение: НС – это громадный распределенный параллельный процессор, состоящий из элементарных единиц обработки информации (нейронов), которые накапливают экспериментальные знания и представляют их для последующей обработки.

Существуют мнения, что связь НС с биологией слаба и зачастую несущественна, однако функционирование НС часто напоминает человеческое познание.

Выделим 2 сходства НС с мозгом человека:

- Знания поступают в НС из окружающей среды и используются в процессе обучения;
- Для накопления знаний применяются связи между нейронами, называемые синаптическими весами.

2. Понятие ИИ.

Интеллект — что означает ум, рассудок, разум, мыслительные способности человека.

Искусственный интеллект (ИИ) — обычно толкуется как свойство автоматических систем брать на себя отдельные функции интеллекта человека, например, выбирать и принимать оптимальные решения на основе ранее полученного опыта и рационального анализа внешних воздействий.

ИИ как наука существует около полувека. Первой ИИС считается программа «Логик-Теоретик», предназначенная для доказательства теорем. Ее работа впервые была продемонстрирована 9 августа 1956 г. В создании программы участвовали такие известные ученые, как А. Ньюэлл, А. Тьюринг, К. Шеннон, Дж. Лоу, Г. Саймон и др.

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

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 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.

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.

2. Основные направления развития ИИ.

ИИС проникают во все сферы нашей жизни, поэтому трудно провести строгую классификацию направлений, по которым ведутся исследования в области ИИ. Рассмотрим кратко некоторые из них.

Разработка ИИС, основанных на знаниях - это одно из главных направлений ИИ. Основной целью построения таких систем является 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 БЗ, образующих ядро СОЗ. Частным случаем СОЗ является ЭС.

Разработка естественно-языковых интерфейсов и машинный перевод. Системы машинного перевода строятся как ИИС, т.к. в их основе лежат БЗ в определенной предметной области и сложные модели, обеспечивающие дополнительную трансляцию «исходный язык оригинала - язык смысла — язык перевода». Данное направление охватывает также исследования методов и разработку систем, обеспечивающих реализацию процесса общения человека с ПК на естественном языке.

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.

Обработка визуальной информации. Задача обработки изображений связана с трансформированием графических образов, результатом которой 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 знаний признаков.

Игры и машинное творчество. Машинное творчество охватывает сочинение компьютерной музыки, ИИС для изобретения новых объектов. Создание интеллектуальных компьютерных игр является одним из самых развитых коммерческих направлений в сфере разработки ПО.

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

Новые архитектуры компьютеров. Это направление связано с созданием компьютеров не фон-неймановской архитектуры, 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

Мы, в нашем курсе, интеллектом будем называть способность мозга решать задачи путем приобретения, запоминания и целенаправленного преобразования знаний в процессе обучения на опыте и адаптации к разнообразным обстоятельствам.

3. Основные понятия ИИС.

Понятие данные, знания и информация отражают три аспекта предметной области сознаний субъекта.

1. Синтаксический
2. Семантический
3. Прагматический

Синтаксический аспект действительности реализует данные, которые представляют собой записанные посредством какого-либо носителя, факты и их зависимости. Факты предметной области могут быть представлены в виде описаний на естественном языке, в виде графических диаграмм и математических формул.

Знания представляют собой данные, осмысленные или понятые субъектом, которые запоминаются для последующего целенаправленного использования. Знания представляют семантический аспект предметной области.

Информация – приращение знаний субъекта. Информация – новые и полезные данные, осмысливаемые на основе имеющегося знания. Информация – отражение прагматического аспекта предметной области.

ИИС – это ИС, которая основана на концепции использования БЗ для генерации алгоритмов прикладных задач различных классов в зависимости от конкретных информационных потребностей пользователей.

4. Классификация ИИС.

Выделяют 4 основных признака интеллектуальности ИС:

1. Развитые коммуникативные способности, т.е. способы взаимодействия конечного пользователя с системой;
2. Умение решать сложные плохо формализуемые задачи, которые требуют построения оригинального алгоритма решения в зависимости от конкретной ситуации, характеризующейся неопределенностью и динамичностью исходных данных и знаний;
3. Самообучаемость, т.е. умение системы автоматически извлекать знания из накопленного опыта и применять их для решения задач;
4. Адаптивность – адекватное отражение действительности, способность системы к развитию в соответствии с объективными изменениями области знаний.

Каждому из перечисленных признаков условно соответствует свой класс ИИС. Различные системы могут обладать одним или несколькими признаками интеллектуальности с различной степенью проявления.

Под интеллектуальным интерфейсом будем понимать интерфейс, обеспечивающий непосредственное взаимодействие конечного пользователя и ПК при решении задачи в составе человеко-машинной системы, выполняющий 3 группы функций:

- обеспечение для пользователя возможности постановки задачи для ЭВМ путем сообщения только ее условия (без задания программы решения);
- обеспечение для пользователя возможности формирования сред решения задачи с использованием только терминов и понятий из области профессиональной деятельности пользователя, естественных форм представления информации;
- обеспечение гибкого диалога с использованием разных средств, в том числе не регламентируемых заранее, с коррекцией возможных ошибок пользователя.

Гипертекстовые системы

В основе гипертекстного подхода к изложению знания лежит так называемая "метафора гипертекста", суть которой в том, что в памяти человека знания "упакованы" в

виде отдельных идей и фактов, между которыми установлены логико-смысловые связи. ГС реализует такую структуру на физических носителях памяти.

Примерами гипертекста служат толковые словари и энциклопедии, состоящие из статей, в которых содержатся ссылки на другие статьи.

1.2. Лекция № 2 (2 часа).

Тема: «Многослойный персептрон. Обучение многослойного персептрона методом обратного распространения ошибки»

1.2.1. Вопросы лекции:

1. Многослойный персептрон. Метод обратного распространения ошибки.
2. Семантические сети. Основные понятия семантических сетей. Типы отношений в семантических сетях. Абстрактные и конкретные сети.
3. Фреймы и объекты. Основные понятия фрейма: слоты, присоединенные процедуры-слуги и процедуры-демоны, наследование свойств. Связь понятия фрейма и объекта в объектно-ориентированном программировании. Сети фреймов. Принципы обработки данных в сети фреймов.

1.2.2. Краткое содержание вопросов:

1. Многослойный персептрон. Метод обратного распространения ошибки.

Проблема обучения многослойного персептрона

Возникает вопрос - почему для обучения многослойного персептрона нельзя применить уже известное дельта-правило Розенблатта?

Ответ состоит в том, что для прим-я метода Розенблатта необходимо знать не только тек. выходы нейронов, но и требуемые *правильные* зн-я.

В случае многослойной сети эти правильные значения имеются только для нейронов *выходного* слоя. Требуемые значения выходов для нейронов скрытых слоев неизвестны, что и ограничивает применение дельта-правила.

Возможные варианты решения проблемы

1. Разработка наборов выходных сигналов, соответствующих входным, для каждого слоя НС, что, конечно, является очень трудоемкой операцией и не всегда осуществимо.

2. Динамическая подстройка весовых коэффициентов синапсов, в ходе которых выбираются, как правило, наиболее слабые связи и изменяются на малую величину в ту или иную сторону, а сохраняются только те изменения, которые повлекли уменьшение ошибки на выходе всей сети. Очевидно, что данный метод "тыка", несмотря на свою кажущуюся простоту, требует громоздких рутинных вычислений.

3. Распределение сигналов ошибки от выходов НС к ее входам, в направлении, обратном прямому распределению сигналов в обычном режиме работы. Так называемый метод ОРО (обработка распределения ошибки).

Т.к. вывод алгоритма ОРО слишком громоздок, то для упрощения понимания мат. выкладок р-м => понятия и обоз-я:

1. Пусть $E(w)$ - непр-но диф-мая функция стоимости, зависящая от некоторого неизвестного вектора весовых коэффициентов, тогда $\nabla E(w) = \left[\frac{\partial E}{\partial w_1}, \frac{\partial E}{\partial w_2}, \dots, \frac{\partial E}{\partial w_m} \right]^T$ - *вектор градиента* функции стоимости.

2. Под *функциональным сигналом* будем понимать – входной сигнал (стимул), поступающий в сеть и передаваемый вперед от нейрона к нейрону по всей сети. Такой сигнал достигает конца сети в виде выходного сигнала.

Составим таблицу используемых обозначений.

Таблица 1 - Используемые обозначения

Обоз-я	Действия
--------	----------

i, j и k	Будем относить к различным нейронам сети. Когда сигнал проходит по сети слева направо, считается, что нейрон j находится на один слой правее нейрона i , а нейрон k - еще на один слой правее нейрона j , если последний принадлежит скрытому слою.
n	Номер итерации, соответствует n – му обучающему образцу(примеру), поданному на вход сети.
$E(n)$	Текущая сумма квадратов ошибок (или энергия ошибки) на итерации n .
E_{av}	Средняя энергия $E(n)$ по всем значениям n (т.е. по всему обучающему множеству) называется средней энергией ошибки.
$e_j(n)$	Описывает сигнал ошибки на выходе нейрона j на итерации n .
$d_j(n)$	Желаемый отклик нейрона j .
$y_j(n)$	Описывает функциональный сигнал, генерируемый на выходе нейрона j на итерации n .
$w_{ji}(n)$	Синаптический вес, связывающий выход нейрона i со входом нейрона j на итерации n .
$\Delta w_{ji}(n)$	Коррекция, применяемая к весу $w_{ji}(n)$ на шаге n .
$v_j(n)$	Индукцированное локальное поле (т.е. взвешенная сумма всех синхронных входов + порог) нейрона j на итерации n . Это значение передается в функцию активации, связанную с нейроном j .
$\varphi_j(\cdot)$	Функция активации, соответствующая нейрону j и описывающая нелинейную взаимосвязь входных и выходных сигналов этого нейрона.
$x_i(n)$	i -й элемент входного вектора.
$o_k(n)$	k -й элемент выходного вектора.
m_l	Размерность (количество узлов) слоя l многослойного персептрона; $l=1,2,\dots,L$, где L – глубина сети. Т.е. m_0 – размерность входного слоя; m_1 – размерность 1-го скрытого слоя; m_L – размерность выходного слоя.

Метод обратного распространения ошибок

Известно:

- что сигнал ошибки выходного нейрона j на итерации n (соответствующей n – му примеру обучения) определяется соотношением:

$$e_j(n) = d_j(n) - y_j(n). \quad (6.1)$$

- текущая энергия ошибки нейрона j определяется как $\frac{1}{2}e_j^2(n)$. Соответственно текущая общая энергия ошибки сети вычисляется путем сложения этих величин по всем нейронам выходного слоя, т.к. это видимые нейроны, то можно записать:

$$E(n) = \frac{1}{2} \sum_{j \in C} e_j^2(n), \quad (6.2)$$

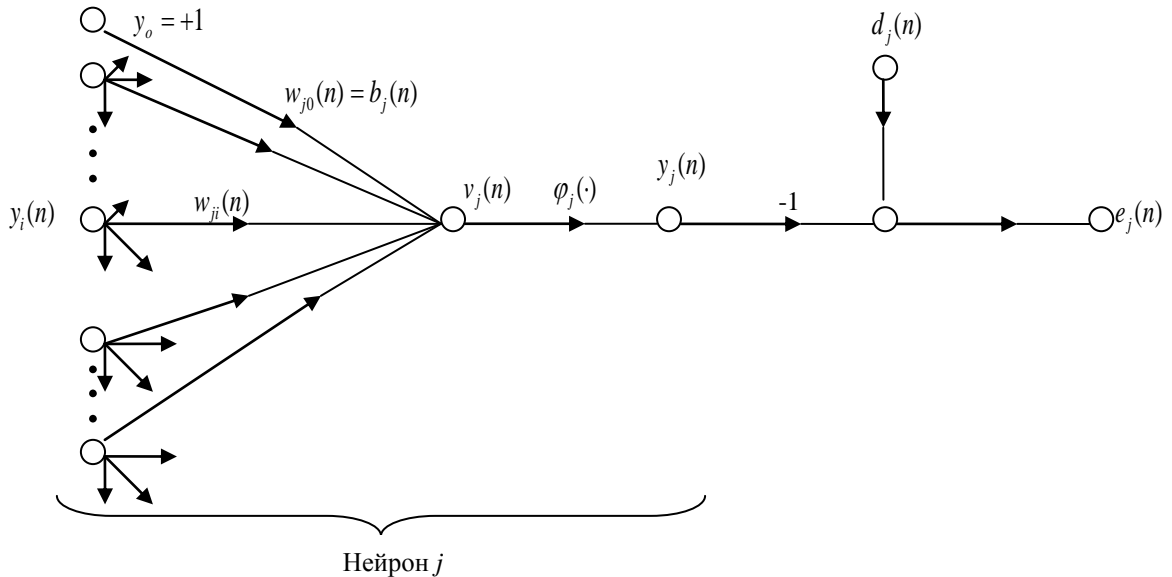
где множество C включает все нейроны выходного слоя НС.

Пусть N – общее число образов в обученном множестве (т.е. мощность этого множества), тогда энергия среднеквадратической ошибки вычисляется как нормализованная по N сумма всех зн-й энергии ошибки $E(n)$:

$$E_{av}(n) = \frac{1}{N} \sum_{n=1}^N E(n). \quad (6.3)$$

Для данного множества энергия E_{av} – представляет собой функцию стоимости – меру эффективности обучения. А значит цель прочеса обучения многослойного персептрона является настройка свободных параметров НС с минимизации величины E_{av} .

Р-м граф передачи сигнала в пределах некоторого нейрона j .



На данном рис. изображен нейрон j , на который поступает поток сигналов от нейронов, расположенных в предыдущем слое, тогда инд. лок. поле, полученное на входе ф-и активации, связанной с данным нейроном будет:

$$v_j(n) = \sum_{i=0}^m w_{ji}(n) y_i(n), \quad (6.4)$$

где m – общее число входов(за искл. порога) нейрона j .

Функциональный сигнал на выходе нейрона j на итерации n будет:

$$y_j(n) = \varphi_j(v_j(n)). \quad (6.5)$$

Тогда алгоритм ОРО состоит в применении к син.весу $w_{ji}(n)$ коррекции $\Delta w_{ji}(n)$, пропорциональной частной производной $\frac{\partial E(n)}{\partial w_{ji}(n)}$.

В соотв- и с правилом цепочки, градиент м. представить =>

$$\frac{\partial E(n)}{\partial w_{ji}(n)} = \frac{\partial E(n)}{\partial e_j(n)} \frac{\partial e_j(n)}{\partial y_j(n)} \frac{\partial y_j(n)}{\partial v_j(n)} \frac{\partial v_j(n)}{\partial w_{ji}(n)}. \quad (6.6)$$

Диф-я обе части ур-я (6.2) по $e_j(n)$, получаем:

$$\frac{\partial E(n)}{\partial e_j(n)} = e_j(n). \quad (6.7)$$

Диф-я обе части ур-я (6.1) по $y_j(n)$, получим:

$$\frac{\partial e_j(n)}{\partial y_j(n)} = -1. \quad (6.8)$$

Диф-я обе части ур-я (6.5) по $v_j(n)$, получим:

$$\frac{\partial y_j(n)}{\partial v_j(n)} = \phi'_j(v_j(n)), \quad (6.9)$$

где штрих справа от имени ф-ции обоз-ет диф-е по аргументу.

И, наконец, диф-я обе части ур-я (6.4) по $w_{ji}(n)$, получим:

$$\frac{\partial v_j(n)}{\partial w_{ji}(n)} = y_i(n). \quad (6.10)$$

Подставим результаты (6.7)-(6.10) в выражение (6.6), окончательно получим:

$$\frac{\partial E(n)}{\partial w_{ji}(n)} = -e_j(n) \phi'_j(v_j(n)) y_i(n). \quad (6.11)$$

Согласно дельта-правила коррекция весовых коэф-в опр-ся:

$$\Delta w_{ji}(n) = -\eta \frac{\partial E(n)}{\partial w_{ji}(n)}. \quad (6.12)$$

Исп-е знака «-» в (6.12) связано с реализацией градиентного спуска в пр-ве весов.

След-но, подставляя (6.11) в (6.12), получим:

$$\Delta w_{ji}(n) = \eta \delta_j(n) y_i(n), \quad (6.13)$$

где $\delta_j(n)$ - локальный градиент, кот. указ-ет на требуемое измен-е син.веса и выч-ся:

$$\delta_j(n) = -\frac{\partial E(n)}{\partial v_j(n)} = e_j(n) \phi'_j(v_j(n)) \quad (6.14)$$

2. Семантические сети. Основные понятия семантических сетей. Типы отношений в семантических сетях. Абстрактные и конкретные сети.

Однозначное определение семантической сети в настоящее время отсутствует. В инженерии знаний под ней подразумевается граф, отображающий смысл целостного образа. Узлы графа соответствуют понятиям и объектам, а дуги – отношениям между объектами. Формально сеть можно задать в следующем виде:

$$H = \langle I, C, G \rangle$$

I – множество информационных единиц;

C – множество типов связей между информационными единицами;

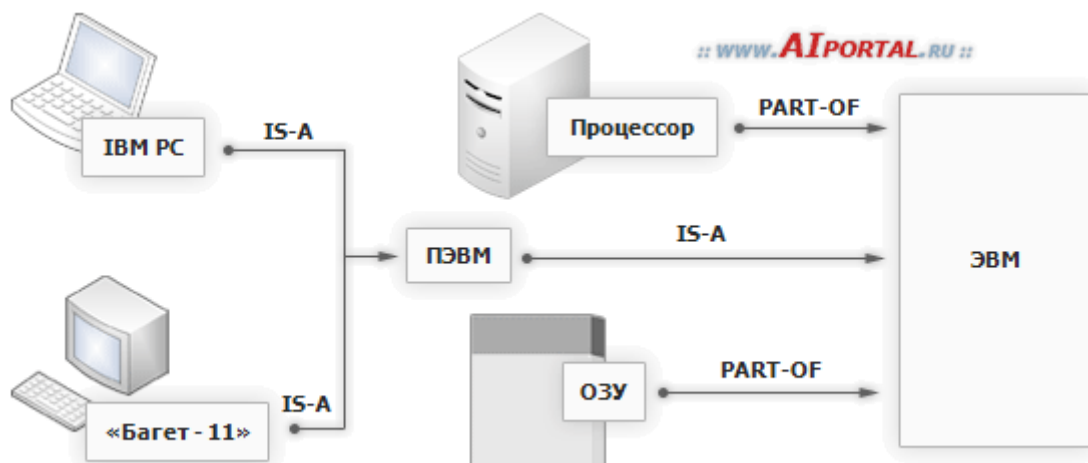
G – отображение, задающее конкретные отношения из имеющихся типов C между элементами I.

Семантическая сеть как модель наиболее часто используется для представления декларативных знаний. С помощью этой модели реализуются такие свойства системы знаний, как интерпретируемость и связность, в том числе по отношению IS-A и PART-OF. За счет этих свойств семантическая сеть позволяет снизить объем хранимых данных, обеспечивает вывод умозаключений по ассоциативным связям.

Одной из первых известных моделей, основанных на семантической сети, является TLC-модель (Teachable Language Comprehender – доступный механизм понимания языка), разработанная Куиллианом в 1968 году. Модель использовалась для представления семантических отношений между концептами (словами) с целью описания структуры долговременной памяти человека в психологии.

Как правило, различают экстенциональные и интенциональные семантические сети. Экстенциональная семантическая сеть описывает конкретные отношения данной ситуации. Интенциональная – имена классов объектов, а не индивидуальные имена объектов. Связи в интенциональной сети отражают те отношения, которые всегда присущи объектам данного класса.

Примером семантической сети может служить фрагмент описания вычислительной техники, показанный на рисунке.



Пример семантической сети

С помощью такой сети, используя отношение IS-A и PART-OF, можно вывести факты: «Багет-11» – это ЭВМ; IBM PC имеет процессор и т.д. Для отображения процедурных знаний используются процедурные семантические сети. В этом случае факты, отношения и процедуры представлены как вершины, а связи объединяют их в единое понятие.

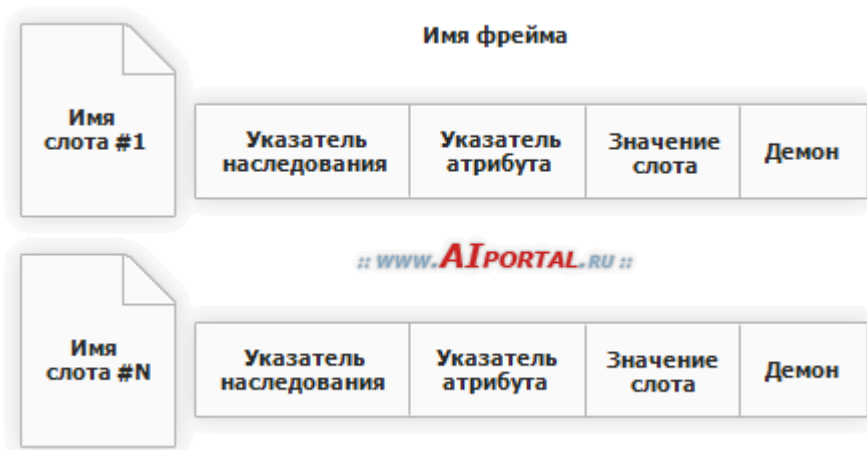
2. Фреймы и объекты. Основные понятия фрейма: слоты, присоединенные процедуры-слуги и процедуры-демоны, наследование свойств. Связь понятия фрейма и объекта в объектно-ориентированном программировании. Сети фреймов. Принципы обработки данных в сети фреймов.

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.

Фрейм (англ. frame – рамка, каркас) – структура данных для представления некоторого концептуального объекта. Информация, относящаяся к фрейму, содержится в составляющих его слотах.

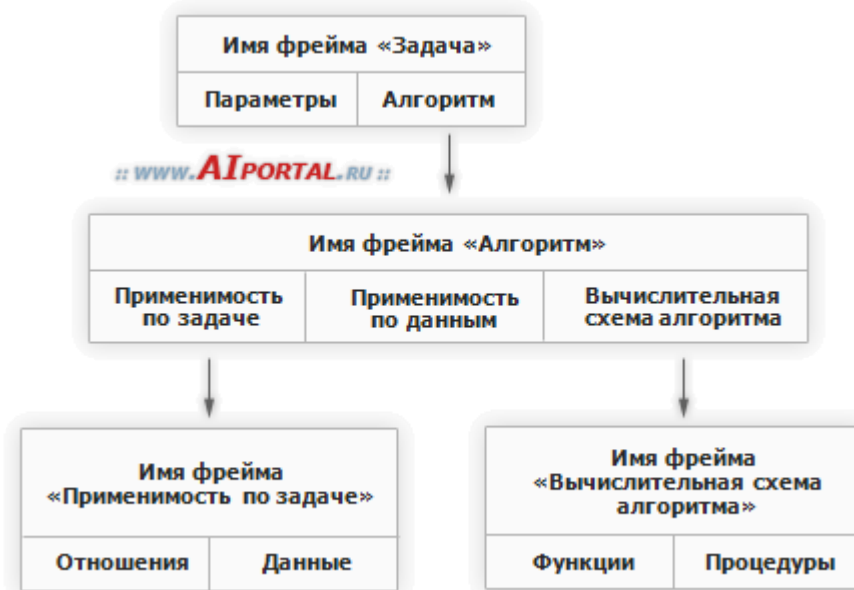
Слот (англ. slot – щель, прорезь) может быть терминальным (листом иерархии) или представлять собой фрейм нижнего уровня.

Каждый фрейм, как показано на рисунке ниже, состоит из произвольного числа слотов, причем несколько из них обычно определяются самой системой для выполнения специфических функций, а остальные определяются пользователем.



Пояснение:

- 1) Имя фрейма (имя фрейма) – это идентификатор, присваиваемый фрейму. Фрейм должен иметь имя, единственное в данной фреймовой модели (уникальное имя);
 - 2) Имя слота (имя слота) – это идентификатор, присваиваемый слоту. Слот должен иметь уникальное имя во фрейме, к которому он принадлежит. Обычно имя слота не несет никакой смысловой нагрузки и является лишь идентификатором данного слота, но в некоторых случаях оно может иметь специфический смысл;
 - 3) Указатель наследования – только для фреймовых моделей иерархического типа; они показывают, какую информацию об атрибутах слотов во фрейме верхнего уровня наследуют слоты с такими же именами во фрейме нижнего уровня;
 - 4) Указатель атрибутов – указатель типа данных слота. К таким типам относятся: FRAME (указатель), INTEGER (целое), REAL (вещественное), BOOL (булево), LISP (присоединенная процедура), TEXT (текст), LIST (список), TABLE (таблица), EXPRESSION (выражение) и другие;
 - 5) Значение слота – значение, соответствующее типу данных слота и удовлетворяющее условиям наследования;
 - 6) Демон – процедура, автоматически запускаемая при выполнении некоторого условия. Демон запускается при обращении к конкретному слоту фреймовой модели. Например, демон IF-NEEDED запускается, если в момент обращения к слоту его значение не было установлено, IF-ADDED запускается при подстановке в слот значения, IF-REMOVED запускается при стирании значения слота.
- Пример фреймовой модели иерархического типа представлен на рисунке ниже:



Фреймы образуют иерархию. Иерархия во фреймовых моделях порождает единую многоуровневую структуру, описывающую либо объект, если слоты описывают только свойства объекта, либо ситуацию или процесс, если отдельные слоты являются именами процедур, присоединенных к фрейму и вызываемых при его актуализации.

Формально фрейм – это тип данных вида:

$$F = \langle N, S_1, S_2, S_3 \rangle$$

N – имя объекта;

S_1 – множество слотов, содержащих факты, определяющие декларативную семантику фрейма;

S_2 – множество слотов, обеспечивающих связи с другими фреймами (каузальные, семантические и т. д.);

S_3 – множество слотов, обеспечивающих преобразования, определяющие процедурную семантику фрейма.

Фреймы подразделяются на:

- 1) Фрейм-экземпляр – конкретная реализация фрейма, описывающая текущее состояние в предметной области;
- 2) Фрейм-образец – шаблон для описания объектов или допустимых ситуаций предметной области;
- 3) Фрейм-класс – фрейм верхнего уровня для представления совокупности фреймов образцов.

Состав фреймов и слотов в каждой конкретной фреймовой модели может быть разным, однако в рамках одной системы целесообразно единое представление для устранения лишнего усложнения.

Разнотипные объекты или объекты, соответствующие концепции «множественности миров», заключающейся, к примеру, в том, что лошадь – животное бескрылое для одного (реального) мира и одновременно крылатое (Пегас в мифическом мире) для другого, могут описываться отличающимися друг от друга фреймами.

В целом фреймовая модель допускает представление всех свойств декларативных и процедурных знаний. Глубина вложенности слотов во фрейме (число уровней) зависит от предметной области и языка, реализующего модель.

2. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ВЫПОЛНЕНИЮ ЛАБОРАТОРНЫХ РАБОТ

2.1 Лабораторная работа №1 (2 часа).

Тема: «Основные понятия нейροкомпьютерных систем»

2.1.1 Цель работы: изучить представление знаний в интеллектуальных системах, изучить методы работы со знаниями

2.1.2 Задачи работы:

1. Проведите анализ представленных определений искусственного интеллекта.
2. Какие сложные задачи решает искусственный интеллект?
3. Представьте определение СИИ.

2.1.3 Перечень приборов, материалов, используемых в лабораторной работе:

1. ПЭВМ

2.1.4 Описание (ход) работы:

Продукционные правила – это правила, имеющие форму: ЕСЛИ «Условие» – ТО «Событие». Продукционные правила описывают знания в виде взаимосвязей типа: «причина» – «следствие», «явление» – «реакция», «признак» – «факт» и.т.п. Конкретизация продукционных правил меняется в зависимости от сущности представляемых знаний.

Например:

- ЕСЛИ «Температура в реакторе превышает 120 ° С » ТО «Снизить подачу топлива на 5%»;
- ЕСЛИ «Вышел из строя вентилятор кондиционера» ТО «Температура в помещении повышается»;

Продукционное представление знаний с человеческой точки зрения является прямым описанием логических выводов при решении конкретных задач. Совокупность знаний о конкретной предметной области в этом случае представляется соответствующим набором продукционных правил, который образует базу знаний. При построении продукционных правил допустимо использование логических операторов И, ИЛИ, например:

- ЕСЛИ «Температура в реакторе превышает 120 ° С » И «Температура хладагента превышает 90 ° С » ТО «Прекратить подачу топлива»;
- ЕСЛИ «Температура в реакторе превышает 90 ° С » ИЛИ «Температура хладагента превышает 60 ° С » ТО «Снизить подачу топлива на 40%».

Недостатком языка продукционных правил можно считать отсутствие явных связей между правилами и целями, к достижению которых необходимо стремиться. Таким образом, для активизации одного из продукционных правил необходимо проверка всей продукционной базы знаний, что при больших объемах информации приводит к существенным затратам временных и технических ресурсов интеллектуальной системы. Возможность решения этой проблемы заключается в разработке перспективных продукционных баз знаний, в которых одни продукционные правила могут активировать и деактивировать другие продукционные правила, влияя на количество перебираемых правил в текущем цикле и, следовательно, на выбор пути достижения цели управления.

Отличительной чертой и основным преимуществом продукционной базы знаний является простота анализа, дополнения, модификации и аннулирования определенных продукционных правил. Помимо этого, представление знаний в таком синтаксически однотипном виде существенно облегчает техническую реализацию системы использования знаний. Вследствие этого в настоящее время продукционные базы знаний получили наибольшее распространение в интеллектуальных технических системах.

Знаниями можно назвать описания отношений между абстрактными понятиями и сущностями, являющимися конкретными объектами реального мира. Изначально семантические сети разрабатывались как модели долговременной человеческой памяти в психологии, но впоследствии эта модель перекочевала в инженерию знаний. В семантической сети абстрактные понятия и отношения между ними описываются в виде узлов и дуг. Сущности и понятия в такой сети являются узлами, а отношения между ними – дугами. Атрибуты семантических сетей можно разделить на лингвистические (объект, условие, место, инструмент, цель и т.п.), атрибутивные (форма, размер, цвет и т.п.), характеристические (род, время, наклонение и т.п.), логические (да, нет, отрицание, объединение и т.п.).

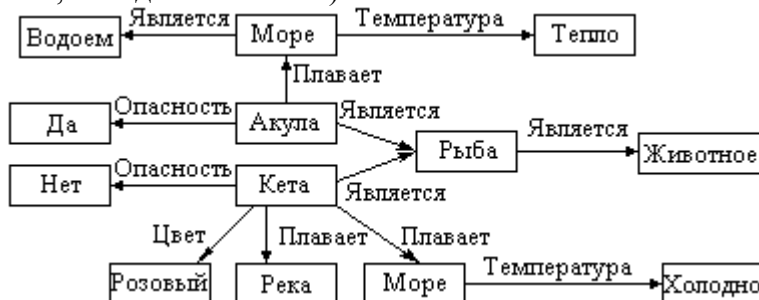


Рис.1.5. Семантическое представление знаний биолога

Допустим, фрагмент знаний ихтиолога о биологии рыб можно описать следующей семантической сетью (рис.1.5). В качестве другого примера рассмотрим представление знаний, содержащихся в высказывании: «Робот сверлит отверстие в детали с помощью сверла 10» (рис.1.6).



Рис.1.6. Семантическое представление технического знания

Недостаток семантических сетей – дублирование информации при построении сетей и смешение групп знаний, относящихся к различным ситуациям. Например, семантическая сеть, представленная на рис.1.5, имеет дуближ понятия «море», а

отношение «температура» может использоваться не только для описания среды обитания животных. Выходом из данной ситуации стала наметившаяся в последнее время тенденция к построению разделенных семантических сетей.

Основным преимуществом семантических сетей является то, что они имитируют понимание и использование человеком естественного языка, что позволяет применять их при техническом моделировании рассуждений, доказательстве теорем, построении незадаанных явно причинно-следственных связей и лингвистических конструкций, т.е. семантические сети позволяют реализовать устройства, имитирующие мыслительные акты более высокого уровня по сравнению с продукционными правилами. Представление знаний в виде семантических сетей широко используется в интеллектуальных системах интерпретации естественного языка и автоматического машинного перевода, в диалоговых вопросно-ответных системах естественного человека-машинного общения, в блоках логической интерпретации систем технического зрения.

Приобретением знаний называется выявление знаний из источников и преобразование их в нужную форму, а также перенос в базу знаний ИС. Источниками знаний могут быть книги, архивные документы, содержимое других баз знаний и т. п., т. е. некоторые *объективизированные знания*, переведенные в форму, которая делает их доступными для потребителя. Другим типом знаний являются *экспертные знания*, которые имеются у специалистов, но не зафиксированы во внешних по отношению к нему хранилищах. Экспертные знания являются *субъективными*. Еще одним видом субъективных знаний являются *эмпирические знания*. Такие знания могут добываться ИС путем наблюдения за окружающей средой (если у ИС есть средства наблюдения).

Ввод в базу знаний объективизированных знаний не представляет особой проблемы, выявление и ввод субъективных и особенно экспертных знаний достаточно трудны. Чтобы разработать методологию приобретения субъективных знаний, получаемых от эксперта, надо четко различать две формы репрезентации знаний. Одна форма связана с тем, как и в каких моделях хранятся эти знания у человека-эксперта. При этом эксперт не всегда осознает полностью, как репрезентированы у него знания. Другая форма связана с тем, как инженер по знаниям, проектирующий ИС, собирает их описывать и представлять. От степени согласованности этих двух форм репрезентации между собой зависит эффективность работы инженера по знаниям.

В когнитивной психологии изучаются формы репрезентации знаний (*когнитивные структуры знаний*) характерные для человека. Примерами могут служить [Хафман, 1986]: представление класса понятий через его элементы (например, понятие "птица" репрезентируется рядом чайка, воробей, скворец, ...)

представление понятий класса с помощью базового прототипа, отражающего наиболее типичные свойства объектов класса (например, понятие "птица" репрезентируется прототипом нечто с крыльями, клювом, летает ...) представление с помощью признаков (для понятия "птица", например, наличие крыльев, клюва, двух лап, перьев).

Кроме понятий репрезентируются и отношения между ними. Как правило, отношения между понятиями определяются процедурным способом, а отношения между составляющими понятием (определяющими структуру понятия) - декларативным способом. Наличие двух видов описаний заставляет в моделях представления знаний одновременно иметь оба компонента, например семантическую сеть и продукционную систему, как это представлено в когнитивной модели [Anderson, 1983].

При приобретении знаний важную роль играют так называемое *поле знаний* в котором содержатся основные понятия, используемые при описании предметной области, и свойства всех отношений, используемых для установления связей

между понятиями. Поле знаний связано с концептуальной моделью проблемной области, в которой еще не учтены ограничения, которые неизбежно возникают при

формальном представлении знаний в базе знаний. Переход от описания некоторой области в поле знаний к описанию в базе знаний аналогичен переходу от концептуальной модели базы данных к ее логической схеме, когда уже зафиксирована система управления базой данных. Важно отметить, что переход непосредственно к формальным представлениям в базе знаний без этапа концептуального описания в поле знаний приводит к многочисленным ошибкам, что замедляет процесс формирования базы знаний ИС.

Возможны три режима взаимодействия инженера по знаниям с экспертом-специалистом: *протокольный анализ, интервью и игровая имитация профессиональной деятельности*. Протокольный анализ заключается в фиксации (например, путем записи на магнитную ленту "мыслей вслух" эксперта во время решения проблемы и в последующем анализе полученной информации. В режиме интервью инженер по знаниям ведет с экспертом активный диалог, направляя его в нужную сторону. При игровой имитации эксперт помещается в ситуации, похожие на те в которых протекает его профессиональная деятельность. Наблюдая за его действиями в различных ситуациях, инженер по знаниям, формирует свои соображения об экспертных знаниях, которые впоследствии могут быть уточнены с экспертом в режиме интервью. Принципы игровой имитации нашли применение в разнообразных деловых играх, специальных тренажерах.

Каждый из упомянутых способов извлечения знаний имеет свои преимущества и недостатки. Так, при анализе протоколов инженеру по знаниям нелегко отделить понятия, важные для включения в словарь предметной области, от тех, которые при "мыслях вслух" появляются случайно. Кроме того, в протоколах обнаруживаются пробелы, когда рассуждение эксперта как бы прерывается и продолжается уже на основе пропущенных шагов вывода. Заполнение подобных лакун возможно лишь в режиме интервью. Таким образом, во всех трех подходах к извлечению знаний из экспертов необходим этап интервью, что делает его одним из важнейших методов приобретения знаний.

Существует не менее двух десятков стратегий интервьюирования. Наиболее известны три: разбиение на ступени, репертуарная решетка и подтверждение сходства,

При разбиении на ступени эксперту предлагается назвать наиболее важные, по его мнению, понятия предметной области и указать между ними отношения структуризации, т. е. отношения типа "род-вид", "элемент-класс", "целое-часть" и т. п. Эти понятия используются на следующем шаге опроса как базовые. Стратегия нацелена на создание иерархии понятий предметной области, выделение в понятиях тесно связанных между собой групп-*гаксонов* (кластеров) .

Стратегия репертуарной решетки направлена на выявление характеристических свойств понятий, позволяющих отделять одни понятия от других. Методика состоит в предъявлении эксперту троек понятий с предложением назвать признаки для каждой двух понятий, которые отделяли бы их от третьего. Так как каждое понятие входит в несколько троек, то на основании такой процедуры происходит уточнение объемов понятий и формируются "симптокомплексы" понятий, с помощью которых эти понятия могут идентифицироваться в базе знаний.

Стратегия подтверждения сходства состоит в том, что эксперту предлагается установить принадлежность каждой пары понятий из предметной области к некоторому отношению сходства (толерантности). Для этого эксперту задается последовательность достаточно простых вопросов, цель которых заключается в уточнении того понимания сходства, которое вкладывает эксперт в утверждение о сходстве двух понятий предметной области.

Процесс взаимодействия инженера по знаниям (аналитика) с экспертом-специалистом включает три основных этапа.

1. Подготовительный этап. Для успеха общения оба участника должны тщательно подготовиться к диалогу или игре. Желательно, чтобы эксперт был не только компетентным специалистом, но и заинтересованным (морально или материально) лицом

в достижении конечной цели-построении ИС. Он должен быть доброжелателен к аналитику и уметь объяснять свои знания (наилучший случай когда эксперт имеет опыт преподавательской работы).

Аналитику необходимо: глубоко познакомиться со специальной литературой по предметной области" чтобы не задавать очень "глупых" вопросов (просто "глупые" вопросы бывают чрезвычайно полезны), а также увеличить количество "пакетов ожиданий" [Шенк и др., 1987]; уметь слушать и грамотно задавать вопросы; настроиться на роль "ученикам, а не "экзаменатора"; разбираться в моделях когнитивной психологии, а также в моделях представления знаний, чтобы из знаний эксперта выделять четкие структуры.

В любой совместной деятельности большое значение имеют психологические качества исследователей, такие как личность, манера поведения, стиль научного мышления. Существуют различные классификации научных работников. В качестве примера приведем следующую: инициатор - быстро реагирует на перспективные проблемы, т. е. один из первых ощущает необходимость решения проблемы с элементами неопределенности; диагност-способен к быстрой оценке сильных и слабых сторон решения задачи, эрудит-наделен исключительной памятью, отличается повышенным вниманием к деталям и стремлением к упорядоченности; ремесленник - способен воплощать в жизнь плохо оформленные идеи других; эстет - стремится исследовать проблемы, приводящие к изящным решениям, не склонен к кропотливому труду; методолог-интересован методологическими аспектами исследований; независимый-стремится к индивидуальному решению проблем; фанатик-самоотверженно увлечен своей научной проблемой, того же требует и от окружающих.

Принадлежность научного работника к тому или иному типу определяется с помощью косвенных методик (тестов личности, интеллекта, когнитивных стилей.проектных методик). Автоматизация опроса и получения психологического портрета испытуемого реализована, например, в системе АВТАНТЕСТ [Гаврилова. 1984].

Для роли эксперта наиболее предпочтительны инициатор эрудит, диагност и ремесленник (в паре с аналитиком-эрудитом), а для роли аналитика-диагност, методолог, эрудит, инициатор. При этом наилучшее сочетание дают сочетания разных типов. Благодаря различиям в подходах к решению задачи, в точках зрения, стиле мышления восприятия, памяти и т. п. участники в такой паре с разных сторон подходят к поставленной цели, в результате увеличивается общее количество гипотез, идей, альтернативных вариантов, а следовательно, обогащается поле знаний. Однако не все сочетания даже из приемлемых типов улучшают взаимодействие, а некоторые типы (например, фанатик, эстет, независимый.ремесленник) часто слабо приспособлены для творческого взаимодействия, что приводит к возникновению скрытых и явных конфликтов, которые усложняют процесс продуктивного общения.

Важное значение имеет также лидерство в паре. В ходе любого диалога одна сторона обычно занимает позицию ведущей, чаще эту роль берет интервьюер, т. е. аналитик. Роль лидера в диалоге позволяет аналитику направлять и систематизировать процесс создания поля знания, не давая эксперту "размыться" или излишне детализовать процесс. С другой стороны, догматизм и настойчивость могут привести к неадекватному полю. Имеет место также эффект "фасада", т. е. желание эксперта не ударить "в грязь лицом" перед аналитиком, и отсюда генерирование неподтвержденных гипотез.

2. Установление "общего кода". Для создания лингвистического альянса взаимодействия участники взаимодействия должны пытаться сократить "расстояние" между объектом (т. е. исследуемой предметной областью) и аналитиком. Необходимо определить главные понятия, т. е. выработать словарную основу базы знаний; уровень детализации; взаимосвязи между понятиями.

3. Гносеологический этап. На этом этапе происходит выяснение закономерностей, присущих предметной области, условий достоверности и истинности утверждений, структурирование за счет введения отношений и т. п. Этот этап является определяющим во взаимодействии аналитика и эксперта. В процессе анализа игры или диалога вербализуется и формализуется знание эксперта и зачастую для него самого порождается новое знание. Репрезентация внешнего мира в его памяти получает материальное воплощение в форме поля знаний.

В процессе извлечения знаний сначала желательно получить от эксперта поверхностные знания такие например, как репрезентация признаков), постепенно переходя к глубинным структурам и более абстрактным понятиям (таким, например, как прототипы).

При формировании поля знаний учитываются особенности эмпирического знания: модальность, противоречивость, неполнота и т. д.

Аналитик должен за частным всегда видеть общее, т. е. строить цепочки "факт - обобщенный факт - эмпирический закон - теоретический закон". Центральное звено цепочки - формализация эмпирики. При этом иногда основным на этапе формализации становится не извлечение "слепых" непонятных связей, а понимание внутренней структурной связи понятий предметной области. Искусство аналитика состоит в стремлении к созданию ясной и понятной модели проблемной области.

Следует также учитывать, что эксперты в проблемной области не всегда опираются на логические рассуждения. В их представлениях о проблемной области и методах решения задач, характерных для нее, широкое применение находят ассоциативные рассуждения и рассуждения правдоподобия. Опишем примерную методику работы с экспертом по формированию поля знаний.

Подготовительный этап

1. Четкое определение задач проектируемой системы (сужение поля знаний): определение, что на входе и выходе; определение режима работ, консультации, обучение и др.

2. Выбор экспертов: определение количества экспертов; выбор уровня компетентности (не всегда хорошо выбирать самый высокий уровень сразу); определение способов и возможности заинтересовать экспертов в работе; тестирование экспертов.

3. Знакомство аналитика со специальной литературой в предметной области

4. Знакомство аналитика и экспертов (в дальнейшем для простоты будем считать, что эксперт один).

5. Знакомство эксперта с популярной литературой по искусственному интеллекту (желательно, но необязательно).

6. Попытка аналитика создать поле знаний первого приближения априорным знаниям из литературы (прототип поля знаний).

Основной этап

1. "Накачка" поля знаний: а) в зависимости от предметной области выбор способа интервьюирования; б) протоколирование мыслей вслух или запись на магнитофон рассуждений эксперта (аналитик по возможности не должен пока вмешиваться в рассуждения).

2. "Домашняя работа". Попытка аналитика выделить некоторые причинно-следственные связи в рассуждениях эксперта; построение словаря предметной области (возможно, на карточках) и подготовка вопросов к эксперту.

3. "Подкачка" поля зрения. Обсуждение с экспертом прототипа поля знаний и домашней работы, а также ответы на вопросы аналитика.

4. Формализация концептуальной модели.
5. Построение поля знаний второго приближения.

Системы приобретения знаний от экспертов

Одно из первых рассмотрении интервью как метода инженерии знаний проведено в [Newel 1972], Проблемы, возникающие при извлечении экспертных знаний, некоторые психологи связывают с так называемой когнитивной защитой. В [Kelly, 1985] была развита теория человеческого познания, основанная на понятии "персональных конструкторов", которые человек создает и пытается приспособить к реалиям мира. В [Bose, 1984] теория персональных конструкторов использована для создания системы извлечения экспертных знаний и показала свою способность успешно преодолевать когнитивную защиту, т. е. нежелание экспертов достичь четкого и осознанного ими истолкования основных понятий, отношений между понятиями и приемов решения задач в интересующей инженера по знаниям проблемной области.

Методы интервьюирования эксперта предметной области знаний с использованием нескольких различных стратегий применены при создании системы TEIRESIAS [Davis, 1982]. В [Kahnetal, 1984] выделено восемь различных стратегий интервью, в [Kahnetal, 1985] на основе этих стратегий исследуется возможность автоматического интервьюирования. Автоматизации метода протокольного анализа посвящены работы [Waterman, 1971, 1973; Krippendorf, 1980].

В [Kahnetal. 1985] на примере диагностической системы MORE; описана техника интервьюирования, направленная на выяснение следующих сущностей, гипотез, симптомов, условий, связей и путей. Гипотеза - событие идентификация которого имеет своим результатом диагноз. Симптом-событие, являющееся следствием существования гипотезы, наблюдение которого приближает последующее принятие гипотезы. Условие - событие или некоторое множество событий, которое не является непосредственно симптоматическим для какой-либо гипотезы, но которое может иметь диагностическое значение для некоторых других событий. Связи-соединения сущностей (в том числе, других связей). Путь- выделенный тип связи, который соединяет гипотезы с симптомами. В соответствии с этим используются следующие стратегии интервью: дифференциация гипотез, различение симптомов, симптомная обусловленность, деление пути и др.

Дифференциация гипотез направлена на поиск симптомов, которые обеспечивают более точное различение гипотез. Наиболее мощными в этом смысле являются те симптомы, которые происходят из одного диагностируемого события, Различение симптомов выявляет специфические характеристики симптома, которые, с одной стороны, идентифицируют его как следствие некоторой гипотезы, с другой-противопоставляют другим. Симптомная обусловленность направлена на выявление негативных симптомов, т. е. симптомов, отсутствие которых имеет больший диагностический вес, чем их присутствие. Деление пути обеспечивает нахождение симптоматических событий, которые лежат на пути к уже найденному симптому. Если такой симптом существует, то он имеет большое диагностическое значение, чем уже найденный.

Аналогичные стратегии интервьюирования эксперта использованы при создании инструментальной диагностической системы ИДИС [Голубев и др., 1987].

В системе KRITON [DiederichetaL, 1987] для приобретения знаний используются два источника: эксперт с его знаниями, полученными на практике (эти знания, как правило, неполны, отрывочны, плохо структурированы); книжные знания, документы, описания инструкции (эти знания хорошо структурированы и фиксированы традиционными средствами). Для извлечения знаний из первого источника в KRITON применена техника интервью, использующая стратегии репертуарной решетки и разбиения на ступени. При этом применяется прием переключения стратегий: если при предъявлении тройки семантически связанных понятий эксперт не в состоянии назвать

признак, отличающий два из них от третьего, система запускает стратегию разбиения на ступени и предпринимает попытку выяснения таксономической структуры этих понятий с целью выявления признаков, их различающих.

Для выявления процедурных знаний эксперта в KRITON применен метод протокольного анализа. Он осуществляется в пять шагов. На первом шаге протокол делится на сегменты на основании пауз, которые делает эксперт в процессе записи. Второй шаг-семантический анализ сегментов, формирование высказываний для каждого сегмента. На третьем шаге из текста выделяются операторы и аргументы. Далее делается попытка поиска по образцу в базе знаний для обнаружения переменных в высказываниях (переменная вставляется в высказывание, если соответствующая ссылка в тексте не обнаружена). На последнем шаге утверждения упорядочиваются в соответствии с их появлением в протоколе.

Анализ текста используется в KRITON для выявления хорошо структурированных знаний из книг, документов, описаний, инструкций.

В [Morik, 1987] описан метод выявления модели предметной области. Первая фаза-формирование инженером знаний грубой модели предметной области путем определения предикатов и сортов их возможных аргументов и сообщения системе фактов об области, выразимых этими предикатами. Система выявляет свойства предикатов и устанавливает отношения между ними, структурируя таким образом предметную область. На второй фазе с помощью метазнаний (общих структур), отражающих особенности человеческого мышления, осуществляется проверка соответствия фактов предикатам, индуктивный вывод правил из фактов, вывод правил из других правил.

В системах SIMER и ДИАПС [Осипов. 1987; OsipovetaL, 1987] основным методом приобретения знаний является автоматизированное интервьюирование эксперта, которое управляется знаниями, приобретенными системой. В системах SIMER и ДИАПС не выявляется предварительная модель области. Все объекты (события) и их атрибуты определяются в режиме прямого интервьюирования эксперта. Предполагается только, что на множестве объектов могут быть заданы ряд отношений из известного (конечного) множества: "элемент-множество", "часть - целое", "пример - прототип", отношения структурного сходства объектов, структурной иерархии и некоторые другие. Все отношения попарно различаются формальными свойствами. Так, отношений структурного сходства не обладает транзитивностью, но симметрично. Отношение структурной иерархии, напротив, не обладает симметричностью, однако транзитивно. На выяснение этих и ряда других свойств отношений и объектов направлено интервью.

В частности, для установления структурного сходства на первой фазе интервью для каждого вновь вводимого понятия эксперту предлагается указать (с помощью меню) те понятия предметной области, с которыми может быть связано данное (без спецификации отношения). Затем в процессе интервью для каждой пары понятий (из выделенных на первой фазе) связь специфицируется, устанавливаются свойства и тип отношения, в число элементов которого включается исследуемая пара. Так, для включения некоторой пары понятий X и Y , о которых эксперт сообщил, что X влияет на Y (например X увеличивает возможность Y), в число элементов некоторого отношения R , обладающего среди прочих свойств симметричностью, необходимо задать эксперту вопрос: "Увеличивает ли Y возможность?". При положительном ответе на этот вопрос (и если прочие свойства уже установлены и удовлетворяют определению отношения R) пара (X, Y) включается в R . Для установления структурного сходства и структурной иерархии понятий используются стратегии подтверждения сходства и разбиения на ступени.

В модели имеются метапроцедуры и метаправила, которые проверяют корректность модели, используют формальные свойства отношений для пополнения модели и генерируют правила.

Сформулируем основные этапы реализации системы приобретения знаний.

1. Интервью для определения актуальной области, в которой происходит процесс решения интересующей проблемы, и расчленение ее на автономные области.
2. Автоматизированное интервью для выявления и формирования декларативной модели предметной области.
3. Протокольный анализ к выявленным на предыдущем этапе понятиям и отношениям предметной области для пополнения модели процедурными знаниями.
(этапы 2 и 3 можно использовать попеременно до тех пор, пока модель не достигнет нужной полноты).
4. Протокольный анализ для пополнения декларативных знаний модели.б. Проверка полноты модели. Обычно протокольный анализ выявляет пустоты в модели. Имеется в виду случай, когда понятия, использованные в "мыслях вслух", недостаточно описаны. В этом случае интервью и протокольный анализ повторяются.

2.2 Лабораторная работа №2 (2 часа).

Тема: «Модели и архитектура нейрокомпьютерных систем»

2.2.1 Цель работы: изучить формальные логические модели знаний

2.2.2 Задачи работы:

1. Научиться решать задачи связанные, с формализацией знаний.
2. Проектирование экспертных систем в заданной предметной области с помощью продукционной модели.

2.2.3 Перечень приборов, материалов, используемых в лабораторной работе:

1. ПЭВМ

2.2.4 Описание (ход) работы:

Формализация знаний основана на системе исчисления **предикатов первого порядка**, которая в свою очередь основывается на исчислении высказываний. Высказыванием называется предложение, принимающее только два значения: истина или ложь. Например: «Иван студент». Из простых высказываний с помощью слов: и, или, не, если – то, могут формироваться более сложные высказывания.

Иван студент и Татьяна студентка;

Иван студент или Татьяна студентка.

Логика высказываний оперирует логическими связями между высказываниями, то есть решает вопросы типа:

Можно ли на основе высказывания А получить высказывание В?

Истинно ли высказывание В при истинности высказывания А?

Элементарные высказывания, т.е. те, которые нельзя разделить на частичные, могут рассматриваться как переменные логического типа, над которыми разрешены следующие логические операции: отрицание (\neg); конъюнкция или логическое умножение (\wedge); дизъюнкция или логическое сложение (\vee); импликация (\rightarrow); эквивалентность (\leftrightarrow). Исчисление высказываний позволяет формализовать лишь малую часть множества рассуждений, поскольку этот аппарат не позволяет учитывать внутреннюю структуру высказывания, которая существует в естественных языках.

Пример 1. Пусть сформулированы следующие высказывания:

P: Все люди смертны;

Q: Сократ – человек;

R: Сократ – смертен.

Можно составить формулу:

$$(P \wedge Q) \rightarrow R$$

Однако эта формула не является общезначимой, поскольку относится только к Сократу. Кроме того, высказывание R не выводится из P и Q , то есть при его отсутствии невозможно записать импликацию. Для достижения общезначимости Q необходимо разделить на две части: «Сократ» (субъект) и «человек» (свойство субъекта), что можно записать в виде некоторой функции:

человек (Сократ)

или в общем случае

человек (x)

Такая запись имеет внутреннюю структуру, т.к. значение высказывания является функцией его компонент, не является элементарным высказыванием и называется предикатом первого порядка.

Исчисление предикатов первого порядка – это формальный язык, используемый для представления отношений между объектами и для выявления новых отношений между объектами на основе существующих [7,10]. Алфавит языка исчисления предикатов первого порядка включает переменные, константы, предикаты, логические операции, функции, кванторы (\forall, \exists). Конструкцией предложений в языке исчисления предикатов первого порядка управляют синтаксические правила.

Терм – это переменная, константа или результат применения функции к терму, например, a , x , $f(x)$. Предложения языка исчисления предикатов первого порядка есть формулы, определенные следующим образом:

1. Если P – n -арный предикат (предикат от n аргументов) и t_1, t_2, \dots, t_n – термы, тогда $P(t_1, t_2, \dots, t_n)$ – атомическая формула (атом).
2. Атом – это правильно построенная формула.
3. Если F_1 и F_2 – атомы, то $F_1 \wedge F_2, F_1 \vee F_2, F_1 \rightarrow F_2, \neg F_1$ – тоже атомы.
4. Если F – формула и x – не связанная квантором переменная в F , тогда $\forall x (F)$ и $\exists x (F)$ – также атомы.

Чтобы избежать неоднозначности, необходимо определять формулы, в которых все переменные квантованы, т.е. связаны кванторами, например, $\forall x \exists y \text{ ЛЮБИТ}(x, y)$.

Такая формула называется замкнутой. Замкнутая формула имеет единственное истинное значение. Формула $\exists y \text{ ЛЮБИТ}(x, y)$ является незамкнутой или открытой.

Для построения модели некоторой предметной области следует описать известные факты на языке логики предикатов и, используя ее результаты, построить систему, способную на основе имеющихся фактов строить некоторые новые предложения и отвечать на поставленные вопросы.

Пример 2. Пусть заданы предикаты:

$E(x)$ – « x » въезжает в страну;

$\forall(x)$ – « x » высокопоставленное лицо;

$S(x, y)$ – « y » обыскивает « x »;

$C(y)$ – « y » – таможенник;

$P(x)$ – « x » способствует провозу наркотиков.

Тогда произвольные предложения на естественном языке могут быть записаны в виде:

1. Таможенники обыскивают всех, кто въезжает в страну, кроме высокопоставленных лиц:

$$\forall x (E(x) \wedge \neg \forall(x) \rightarrow (\exists y (S(x, y) \wedge C(y)))).$$

2. Некоторые люди, въезжавшие в страну и способствовавшие провозу наркотиков, были обысканы исключительно людьми, способствовавшими провозу наркотиков:

$$\exists x (E(x) \wedge P(x) \wedge (\forall y (S(x, y) \rightarrow P(y)))).$$

3. Никто из высокопоставленных лиц не способствовал провозу наркотиков:

$$\forall x (P(x) \rightarrow \neg \forall (x)).$$

4. Некоторые таможенники способствуют провозу наркотиков:

$$\exists x (P(x) \wedge C(x)).$$

Задача состоит в том, чтобы, признав фактами предложения 1, 2, 3, доказать, что предложение 4 является истинным.

Для машинного решения вышеприведенной задачи используется методика автоматического формирования суждений, или метод дедукции. При этом последовательно реализуются процедуры: исключение знаков импликации; ограничение области действия знака отрицания; переименование переменных; вынесение кванторов в начало формулы; исключение кванторов и др. При автоматизации вывода доказательств методами исчисления предикатов требуется определить ряд процедур для выбора правил, позволяющих предотвратить «комбинаторный взрыв» и обеспечить проведение немонотонных рассуждений. Решением стало создание декларативных (непроцедурных) языков программирования, в частности Пролога. Программирование на Прологе состоит из этапов:

- объявление некоторых фактов об объектах и отношениях между ними;
- определения некоторых правил об объектах и отношениях между ними;
- формулировки вопросов об объектах и отношениях между ними.

Реально исчисление предикатов первого порядка в промышленных ЭС практически не используется. Формально-логическая модель представления знаний применима в основном в исследовательских системах, т.к. предъявляет очень высокие требования и ограничения к предметной области.

Рассмотрим применение аппарата нечеткой логики на примере оценки надежности поставщика, в котором кроме фактора финансового состояния учитывается и фактор формы собственности. Пусть государственное предприятие не имеет задолженность с уверенностью 60 и предполагается, что его рентабельность удовлетворительна с уверенностью 80. Фрагмент множества правил имеет следующий вид:

Правило 1: Если Задолженность = "нет" и Рентабельность = "удовл."
То Финансовое_состояние = "удовл." cf 100

Правило 2: Если Финансовое_состояние = "удовл."
То Надежность += "есть" cf 90

Правило 3: Если Предприятие = "государств."
То Надежность += "есть" cf 50

Результат выполнения первого правила:
cf(посылки) = min(60, 80) = 60,
cf(Фин_сост. = "удовл.") = 60 * 100 / 100 = 60.

Результат выполнения второго правила: cf(Надежность = "есть") = 60 * 90 / 100 = 54

Результат выполнения третьего правила: cf(Надежность = "есть") = 54 + 50 - 54 * 50 / 100 = 67

Динамические модели. Моделирование рассуждений человека, как правило, не сводится только к прямой или обратной аргументации. Сложные проблемы решаются путем выдвижения во времени нескольких гипотез с анализом подтверждающих фактов и непротиворечивости следствий. Причем для многоцелевых проблемных областей происходит увязка гипотез по общим ограничениям. При этом возможны задержки в принятии решений, связанные со сбором подтверждающих фактов, доказательством подцелей, входящих в ограничения.

Следовательно, для подобных динамических проблем важна рациональная организация памяти системы для запоминания и обновления получаемых промежуточных результатов, обмен данными между различными источниками знаний для достижения нескольких целей, изменение стратегий вывода с выдвижения гипотез (прямая аргументация) к их проверке (обратная аргументация). Целям построения таких гибких механизмов вывода

служит применение технологии "доски объявлений", через которую в результате осуществления событий источники знаний обмениваются сообщениями.

В целях динамического реагирования на события некоторые производционные модели используют специальные правила-демоны, которые формулируются следующим образом:

"Всякий раз, как происходит некоторое событие, выполнить некоторое действие".
Например:

Всякий раз, как становится известным значение переменной "Поставщик",

Выполнить набор правил "Финансовый анализ предприятия"

В программном средстве GURU подобное правило будет записано следующим образом:

IF: *KNOWN("Поставщик")* = *true*
THEN: *CONSULT FIN_AN*

Для динамических экспертных систем характерна также обработка времени как самостоятельного атрибута аргументации логического вывода:

*Если в течение дня уровень запаса понизился больше, чем на 50 %
То выполнить набор правил "Выбор поставщика для поставки"*

Общим недостатком всех формализмов представления знаний, основанных на правилах, является недостаточно глубокое отражение семантики проблемной области, что может сказываться на гибкости формулирования запросов пользователей к экспертным системам. Этот недостаток снимается в объектно-ориентированных методах представления знаний.

Изучение базы знаний ЭС лучше всего начать с анализа концептуальной схемы, поскольку концептуальная схема дает наиболее общее представление о структуре базы знаний. Рассмотрим в качестве примера базу знаний ЭС прогнозирования продолжительности жизни человека. На рис. 1 представлена концептуальная схема этой ЭС. На этой схеме показаны связи между объектами, которые используются в процессе логических выводов. Терминальные объекты выделены на схеме жирными рамками.

«Продолжительность» жизни, являющаяся целью консультации, определяется «основной продолжительностью» и некоторым «фактором», который может привести к увеличению или сокращению «продолжительности» жизни по сравнению с «основной продолжительностью» в зависимости от того, будет значение этого «фактора» положительным или отрицательным.

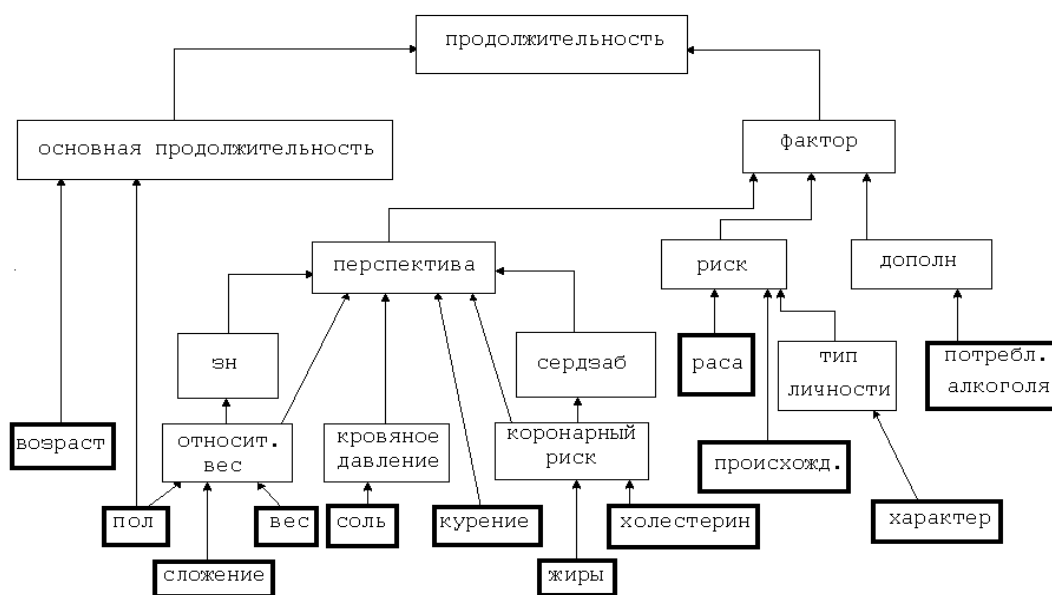


Рис. 1. Концептуальная схема ЭС

Рис. 1. Концептуальная схема ЭС

Как видно из рис. 1, концептуальная схема дает представление не только о структуре базы знаний, но и о подходе к решению задач ЭС. Однако на концептуальной

схеме указываются только имена объектов и не указываются их значения, поэтому для более полного представления процесса решения задач необходимо использовать граф И/ИЛИ.

На рис. 2 представлен фрагмент графа И/ИЛИ, соответствующего концептуальной схеме, показанной на рис. 1. На рис. 2 в кружках указаны номера правил, а в прямоугольниках – условия и заключения правил. На графе И-вершины помечены дугой.

Для построения графа И/ИЛИ найдите в базе знаний правило, указанное преподавателем, и представьте его в виде графа. Затем выберите одно из условий этого правила и найдите в базе знаний такие правила, у которых в заключении стоит та же пара <объект> = <значение>, что и в условии исходного правила. Добавьте граф этого правила к исходному графу.

После этого возьмите второе условие исходного правила и аналогичным образом продолжите другую ветвь графа И/ИЛИ. Продолжение каждой из ветвей строится до тех пор, пока в условиях правил не встретятся объекты, значения которых запрашиваются у пользователя в процессе консультации. Эти условия соответствуют терминальным вершинам графа И/ИЛИ.

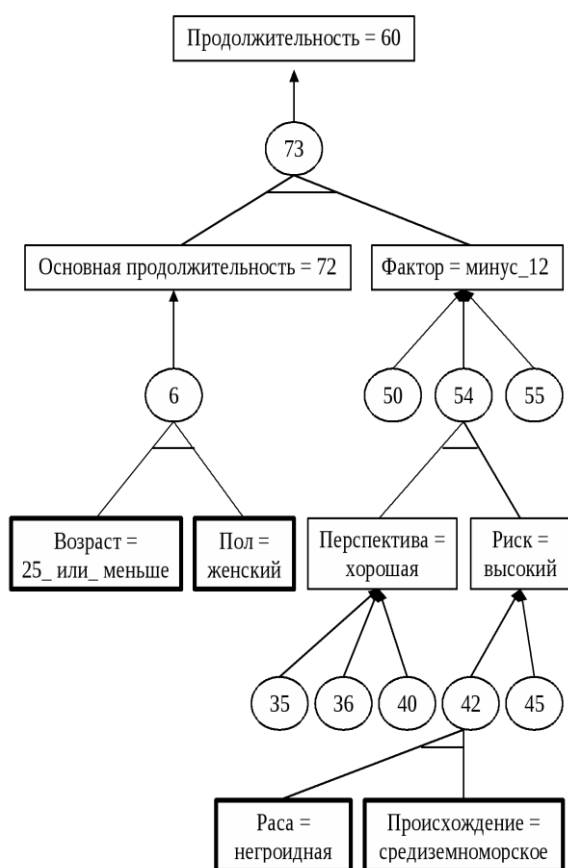


Рис. 2. Фрагмент графа И/ИЛИ для одной из гипотез

2.3 Лабораторная работа №3 (2 часа).

Тема: «Однослойный персептрон. Обучение однослойного персептрона по дельта-правилу»

2.3.1 Цель работы: Изучить алгоритм обучения персептрона.

2.3.2 Задачи работы:

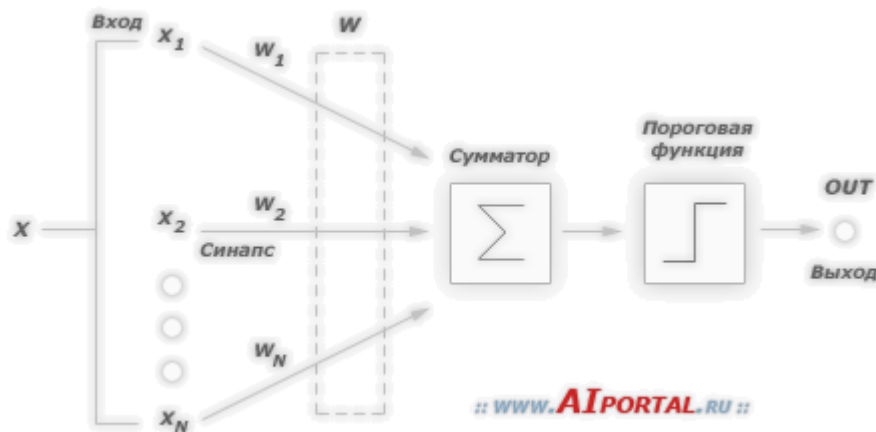
1. Алгоритм обучения персептрона.
2. Правило корректировки весов.
3. Дельта правило.

2.3.3 Перечень приборов, материалов, используемых в лабораторной работе:

1. ПЭВМ

2.3.4 Описание (ход) работы:

Допустим, есть следующая модель персептрона:



Алгоритм обучения персептрона следующий:

1. Присвоить синаптическим весам w_1, w_2, \dots, w_N некоторые начальные значения. Например, нулю.
2. Подать входной образ X и вычислить OUT . Если OUT правильный, то переходят к шагу 4. Иначе к шагу 3.
3. Применяя дельта-правило (см. ниже) вычислить новые значения синаптических весов.
4. Повторить шаги 2-4 данного алгоритма обучения персептрона пока сеть не станет выдавать ожидаемый выход на векторах из обучающей выборки или пока отклонение не станет ниже некоторого порога.

Т.о. образом логика обучения персептрона следующая: если сигнал персептрона при некотором образе верен, то ничего корректировать не надо, если нет – производится корректировка весов.

Правила корректировки весов следующие:

1. Если OUT неверен и равен нулю, то необходимо увеличить веса тех входов, на которые была подана единица.
2. Если OUT неверен и равен единице, то необходимо уменьшить веса тех входов, на которые была подана единица.

Поясним эти правила. Допустим, что на вход был подан некоторый обучающий двоичный вектор X . Этому вектору соответствует выход OUT равный единице. И этот выход неправильный. Тогда веса, присоединенные к единичным входам, должны быть уменьшены, так как они стремятся дать неверный результат. Аналогично, если некоторому другому обучающему вектору X соответствует неправильный выход OUT равный нулю, то веса, присоединенные к единичным входам, должны быть уже уменьшены.

Дельта-правило

Дельта-правило является математической моделью правил корректировки весов. Введем величину δ , которая равна разности между требуемым T и реальным OUT выходом:
$$\delta = T - OUT$$

Тогда, веса персептрона после коррекции будут равны:

$$w_N(i + 1) = w_N(i) + \eta \delta x_N$$

где:

- i – номер текущей итерации обучения персептрона;
- η (греческая буква «эта») – коэффициент **скорости обучения**, позволяет управлять средней величиной изменения весов;
- x_N – величина входа соответствующая w_N синаптическому весу. Добавление величины x_N в произведение позволяет избежать изменение тех весов, которым на входе соответствовал ноль.

Существует доказательство сходимости этого алгоритма обучения персептрона за конечное число шагов. Модель персептрона в настоящий момент представляет больше историческую ценность, чем практическую. Но именно на его примере удастся более наглядно показать некоторые важные принципы функционирования, в том числе и обучения нейронных сетей.

2.4 Лабораторная работа №4 (2 часа).

Тема: «Многослойный персептрон. Обучение многослойного персептрона методом обратного распространения ошибки»

2.4.1 Цель работы: изучить метод обратного распространения ошибки

2.4.2 Задачи работы:

1. Обучить следующую нейронную сеть, применив алгоритм обратного распространения ошибки.

2.4.3 Перечень приборов, материалов, используемых в лабораторной работе:

1. ПЭВМ

2.4.4 Описание (ход) работы:

Допустим необходимо обучить следующую нейронную сеть, применив алгоритм обратного распространения ошибки:

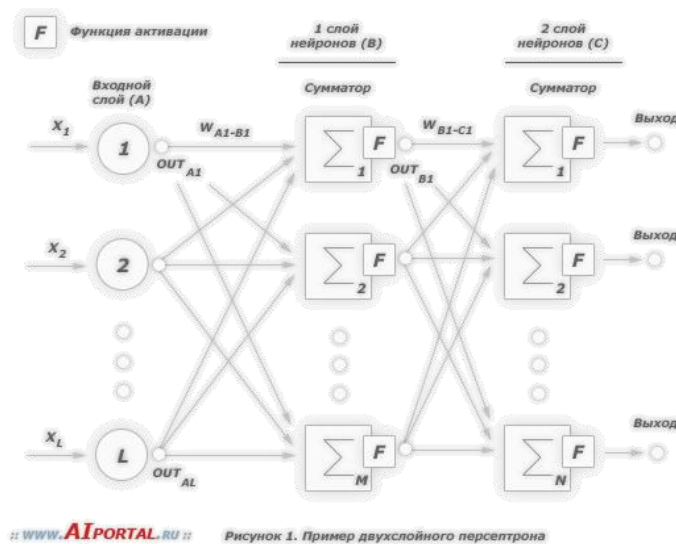


Рисунок 1. Пример двухслойного персептрона

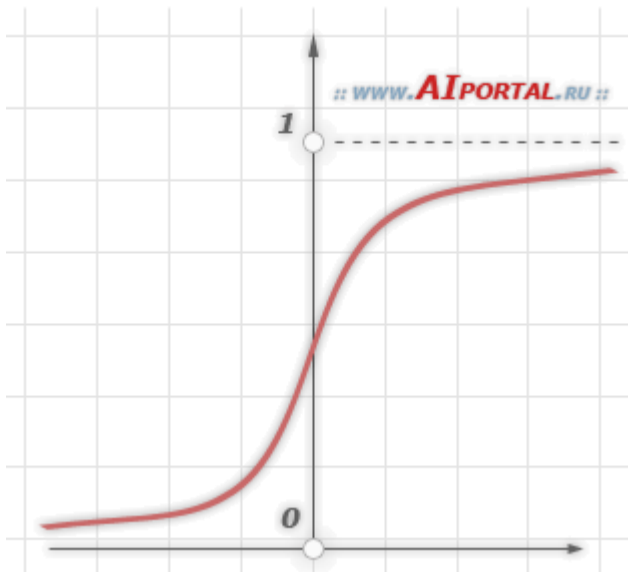
На приведенном рисунке использованы следующие условные обозначения:

- каждому слою нейронной сети соответствует своя буква, например: входному слою соответствует буква A , а выходному – C ;
- все нейроны каждого слоя пронумерованы арабскими цифрами;
- w_{A1-B1} – синаптический вес между нейронами $A1$ и $B1$;
- OUT_{A1} – выход нейрона $A1$.

В качестве активационной функции в многослойных персептронах, как правило, используется сигмоидальная активационная функция, в частности логистическая:

$$OUT = \frac{1}{1 + \exp(-\alpha Y)}$$

где α – параметр наклона сигмоидальной функции. Изменяя этот параметр, можно построить функции с различной крутизной. Оговоримся, что для всех последующих рассуждений будет использоваться именно логистическая функция активации, представленная только, что формулой выше.



Сигмоид сужает диапазон изменения так, что значение OUT лежит между нулем и единицей. Многослойные нейронные сети обладают большей представляющей мощностью, чем однослойные, только в случае присутствия нелинейности. Сжимающая

функция обеспечивает требуемую нелинейность. В действительности имеется множество функций, которые могли бы быть использованы. Для алгоритма обратного распространения ошибки требуется лишь, чтобы функция была всюду дифференцируема. Сигмоид удовлетворяет этому требованию. Его дополнительное преимущество состоит в автоматическом контроле усиления. Для слабых сигналов (т.е. когда OUT близко к нулю) кривая вход-выход имеет сильный наклон, дающий большое усиление. Когда величина сигнала становится больше, усиление падает. Таким образом, большие сигналы воспринимаются сетью без насыщения, а слабые сигналы проходят по сети без чрезмерного ослабления.

Целью обучения сети алгоритмом обратного распространения ошибки является такая подстройка ее весов, чтобы приложение некоторого множества входов приводило к требуемому множеству выходов. Для краткости эти множества входов и выходов будут называться векторами. При обучении предполагается, что для каждого входного вектора существует парный ему целевой вектор, задающий требуемый выход. Вместе они называются обучающей парой. Сеть обучается на многих парах.

Алгоритм обратного распространения ошибки следующий:

1. Инициализировать синаптические веса маленькими случайными значениями.
2. Выбрать очередную обучающую пару из обучающего множества; подать входной вектор на вход сети.
3. Вычислить выход сети.
4. Вычислить разность между выходом сети и требуемым выходом (целевым вектором обучающей пары).
5. Подкорректировать веса сети для минимизации ошибки (как см. ниже).
6. Повторять шаги с 2 по 5 для каждого вектора обучающего множества до тех пор, пока ошибка на всем множестве не достигнет приемлемого уровня.

Операции, выполняемые шагами 2 и 3, сходны с теми, которые выполняются при функционировании уже обученной сети, т.е. подается входной вектор и вычисляется получающийся выход. Вычисления выполняются послойно. На рис. 1 сначала вычисляются выходы нейронов слоя B (слой A входной, а значит никаких вычислений в нем не происходит), затем они используются в качестве входов слоя C , вычисляются выходы OUT^{CN} нейронов слоя C , которые и образуют выходной вектор сети OUT . Шаги 2 и 3 образуют так называемый «проход вперед», так как сигнал распространяется по сети от входа к выходу.

Шаги 4 и 5 составляют «обратный проход», здесь вычисляемый сигнал ошибки распространяется обратно по сети и используется для подстройки весов.

Рассмотрим подробнее 5 шаг – корректировка весов сети. Здесь следует выделить два нижеописанных случая.

Случай 1. Корректировка синаптических весов выходного слоя

Например, для модели нейронной сети на рис. 1, это будут веса имеющие следующие обозначения: w_{B1-C1} и w_{B2-C1} . Определимся, что индексом P будем обозначать нейрон, из которого выходит синаптический вес, а Q – нейрон в который входит:



Введем величину δ , которая равна разности между требуемым T_q и реальным OUT_q выходами, умноженной на производную логистической функции активации (формулу логистической функции активации см. выше):

$$\delta_q = OUT_q \left(1 - OUT_q \right) \left(T_q - OUT_q \right)$$

Тогда, веса выходного слоя после коррекции будут равны:

$$w_{p-q}(i+1) = w_{p-q}(i) + \eta \delta_q OUT_p$$

где:

- i – номер текущей итерации обучения;
- w_{p-q} – величина синаптического веса, соединяющего нейрон P с нейроном Q ;
- η (греческая буква «эта») – коэффициент «скорости обучения», позволяет управлять средней величиной изменения весов;
- OUT_p – выход нейрона P .

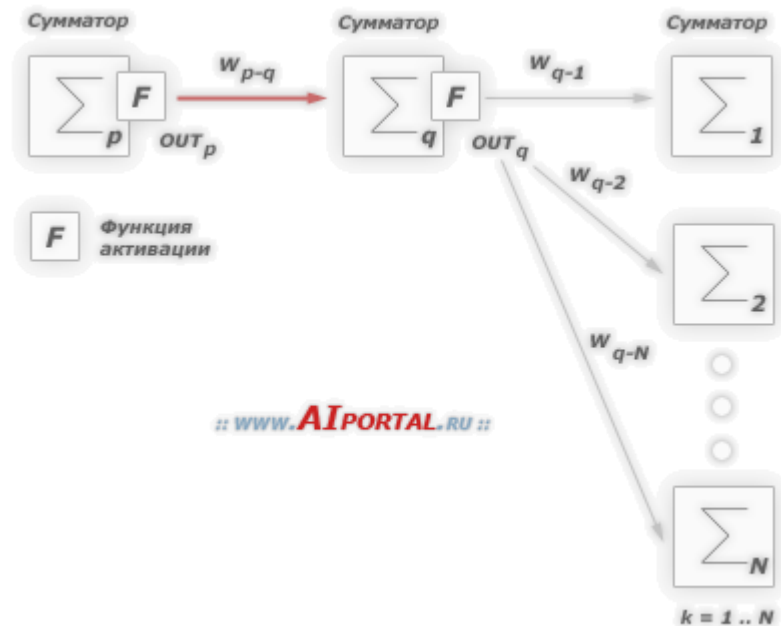
Приведем пример вычислений для синаптического веса w_{B1-C1} :

$$\delta_{C1} = OUT_{C1} \left(1 - OUT_{C1} \right) \left(T_1 - OUT_{C1} \right)$$

$$w_{B1-C1}(i+1) = w_{B1-C1}(i) + \eta \delta_{C1} OUT_{B1}$$

Случай 2. Корректировка синаптических весов скрытого слоя

Для модели нейронной сети на рис. 1, это будут веса соответствующие слоям A и B . Определимся, что индексом P будем обозначать нейрон из которого выходит синаптический вес, а Q – нейрон в который входит (обратите внимание на появление новой переменной k):



Введем величину δ , которая равна:

$$\delta_q = OUT_q \left(1 - OUT_q \right) \sum_{k=1}^M \delta_k w_{q-k}$$

где:

- $\sum_{k=1}^N$ – сумма от 1 по N.

Тогда, веса скрытых слоев после коррекции будут равны:

$$w_{p-q}(i+1) = w_{p-q}(i) + \eta \delta_q OUT_p$$

Приведем пример вычислений для синаптического веса w_{A1-B1} :

$$\delta_{B1} = OUT_{B1} \left(1 - OUT_{B1} \right)$$

$$w_{A1-B1}(i+1) = w_{A1-B1}(i) + \eta \delta_{B1} OUT_{A1}$$

Для каждого нейрона в скрытом слое должно быть вычислено δ и подстроены все веса, ассоциированные с этим слоем. Этот процесс повторяется слой за слоем по направлению к входу, пока все веса не будут подкорректированы.

2.5 Лабораторная работа №5 (2 часа).

Тема: «Самоорганизующиеся обучаемые системы Синапс Хебба и его свойства. Обучение нейрона по алгоритму Хебба»

2.5.1 Цель работы: изучить представление знаний семантическими сетями, изучить фреймовые модели

2.5.2 Задачи работы:

1. Решение задач по представлению знаний семантическими сетями

2. Решение задач представления знаний фреймовыми моделями

2.5.3 Перечень приборов, материалов, используемых в лабораторной работе:

1. ПЭВМ

2.5.4 Описание (ход) работы:

Общепринятого определения семантической сети не существует. Обычно под ней подразумевают систему знаний некоторой предметной области, имеющую определенный смысл в виде целостного образа сети, узлы которой соответствуют понятиям и объектам, а дуги - отношениям между объектами. При построении семантической сети отсутствуют ограничения на число связей и на сложность сети. Для того чтобы формализация оказалась возможной, семантическую сеть необходимо систематизировать. Семантические сети систематизируют функции отношений между понятиями с помощью следующих признаков:

- множество — подмножество (типы отношений «абстрактное — конкретное», «целое — часть», «род — вид»);
- индексы (свойства, имена прилагательные в языке и т.п.);
- конъюнктивные связи (логическое И);
- дизъюнктивные связи (логическое ИЛИ);
- связи по ИСКЛЮЧАЮЩЕМУ ИЛИ;
- отношения «близости»;
- отношения «сходства — различия»;
- отношения «причина — следствие» и др.

При построении семантической сети отсутствуют ограничения на число элементов и связей. Поэтому систематизация отношений между объектами в сети необходима для дальнейшей формализации. Пример семантической сети представлен на рис. 1

Для реализации семантических сетей существуют специальные сетевые языки: Lisp, PROLOG для реализации систем SIMER+MIR и др. Широко известны экспертные системы, использующие семантические сети в качестве языка представления знаний: PROSPECTOR, CASNET, TORUS.

Систематизация отношений конкретной семантической сети зависит от специфики знаний предметной области и является сложной задачей. Особого внимания заслуживают общезначимые отношения, присутствующие во многих предметных областях. Именно на таких отношениях основана концепция семантической сети. В семантических сетях, так же как при фреймовом представлении знаний, декларативные и процедурные знания не разделены, следовательно, база знаний не отделена от механизма

Задание для выполнения.

Нарисовать семантическую сеть, описывающую предметную область.

1. Электрогидравлические усилители мощности
2. Гидравлические усилители мощности
3. Электрические усилители мощности
4. Датчики для САУ
5. Микропроцессоры
6. Микроконтроллеры
7. Индикаторы
8. Резисторы
9. Конденсаторы
10. Транзисторы
11. Тиристоры
12. Диоды
13. Интегральные микросхемы
14. Аналоговые микросхемы
15. Источники электропитания

16. Структура БИТТУ
17. Видеоигры
18. Видеокамеры
19. Вузы Балакова
20. Ваши преподаватели
21. Предприятия Балакова
22. Библиотека БИТТУ
23. Отечественные автомобили
24. Комплектующие ПК

Средства САПР, ориентированные на автоматизацию процедур структурного синтеза опираются на методы искусственного интеллекта (ИИ).

ИИ – это наука о знаниях, способах их получения, представления, переработки и использования в искусственных системах.

В системах ИИ для описания знаний применяют способы, основанные на понятиях фрейма и семантической сети. Фреймы – естественная форма представления сведений об элементах синтезируемых объектов в системах структурного синтеза. В настоящее время концепция фреймов быстро развивается и расширяется, благодаря развитию методов объектно-ориентированного программирования.

Фреймы – это структуры данных, в которой в определенном порядке представлены сведения о свойствах объекта.

Когда человек оказывается в новой ситуации, он извлекает из памяти ранее накопленные блоки знаний, имеющие отношение к текущей ситуации, и пытается применить их. Эти блоки знаний и представляют собой фреймы. Вероятно, знания человека организованы в виде сети фреймов, отражающих его прошлый опыт. Например: типовой номер в гостинице. Он имеет кровать, ванную комнату, шкаф для одежды, телефон и т.д. Детали каждого конкретного номера могут отличаться от приведенного описания. Но они легко уточняются, когда человек оказывается в конкретном номере: цвет обоев, положение выключателей.

Таким образом, любое представление о предмете, объекте, стереотипной ситуации у человека всегда обрамлено (отсюда frame – «рамка») характеристиками и свойствами объекта или ситуации.

Основной структурной единицей фрейма является слот – вложенная во фрейм структура данных, который представляется в виде:

«имя слота»: $\{(A_i, v_i)\}, \{r_i\}$

где A_i - имя признака, v_i - его значение, r_i - связь с другими слотами.

Слоты – это некоторые незаполненные подструктуры фрейма, после заполнения которых конкретными данными, фрейм будет представлять ту или иную ситуацию, явление или объект предметной области. При конкретизации фрейма ему и его слотам присваиваются конкретные имена и происходит заполнение слотов.

В качестве значений слотов могут выступать имена других фреймов, что обеспечивает построение сети фреймов.

В общем виде фрейм выглядит следующим образом:

«Имя фрейма»:

[<роль 1>] (<имя слота 1> : <значение слота 1>);

[<роль 2>] (<имя слота 2> : <значение слота 2>);

.....

[<роль n>] (<имя слота n> : <значение слота n>).

В общем случае структура данных фрейма может содержать более широкий набор информации, в который входят следующие атрибуты.

Имя фрейма. Оно служит для идентификации фрейма в системе и должно быть уникальным. Фрейм представляет собой совокупность слотов, число которых может быть произвольным. Число слотов в каждом фрейме устанавливается проектировщиком системы, при этом часть слотов определяется самой системой для выполнения специфических функций, примерами которых являются: слот-указатель родителя данного фрейма, слот-указатель дочерних фреймов, слот для ввода имени пользователя, слот для ввода даты определения фрейма, слот для ввода даты изменения фрейма и т.д.

Имя слота. Оно должно быть уникальным в пределах фрейма.

Значение слота. Оно должно соответствовать указанному типу данных и условию наследования. Значением слота могут быть числа или математические соотношения, тексты на естественном языке или программы, правила вывода или ссылки на другие слоты данного фрейма или других фреймов.

Пример фрейма РУКОВОДИТЕЛЬ

Имя слота	Значение слота	Тип значения слота
Имя	Иванов И. И.	Строка символов
Рожден	01.01.1965	Дата
Возраст	age(dama, рожден)	Процедура
Специальность	Юрист	Строка символов
Отдел	Отдел кадров	Строка символов
Зарплата	80000	Число
Адрес	ДОМ_АДРЕС	Фрейм

Каждый фрейм можно рассматривать как семантическую сеть, состоящую из выделенных вершин и связей. Верхний уровень фрейма представляет соответствующее понятие, а последующие уровни — терминальные слоты, которые содержат конкретные значения.

Например имеет место ситуация:

Студент Сидоров получил книгу Л.Н. Толстого «Воскресение» в библиотеке им. Н.В. Гоголя, расположенной в Москве».

Описание данной ситуации может быть представлено в виде фрейма:

ПОЛУЧЕНИЕ:

ОБЪЕКТ (КНИГА: (Автор, Л.Н. Толстой), (Название, Воскресение)); АГЕНТ (СТУДЕНТ: (Фамилия, Сидоров));

МЕСТО (БИБЛИОТЕКА: (Название, им. Н.В. Гоголя), (Расположение, г. Москва)).

Здесь ОБЪЕКТ, АГЕНТ и МЕСТО - это роли, которые играют слоты соответственно КНИГА, СТУДЕНТ и БИБЛИОТЕКА в рамках фрейма ПОЛУЧЕНИЕ.

Данную ситуацию можно представить в виде семантической сети – формы представления знаний в виде совокупности понятий и отношений между ними в некоторой предметной области (рис. 1), где можно выделить три характерных уровня. На нулевом уровне представлены конкретные значения сущностей ПО (Толстой, Воскресение, Сидоров и т.д.), на первом - понятия, используемые для описания ПО (КНИГА, СТУДЕНТ, БИБЛИОТЕКА), и на втором - описываемая ситуация ПОЛУЧЕНИЕ. Связи между отдельными понятиями, участвующими в ситуации ПОЛУЧЕНИЕ, также имеют некоторые имена, которые выражают роли понятий в рамках данной ситуации.



Рис. 1 Семантическая сеть

Совокупность фреймов, моделирующая какую-либо предметную область, представляет собой иерархическую структуру, в которой фреймы соединяются с помощью родовидовых связей. На верхнем уровне иерархии находится фрейм, содержащий наиболее общую информацию, истинную для всех остальных фреймов. Фреймы обладают способностью наследовать значения характеристик своих родителей, находящихся на более высоком уровне иерархии. Так, фрейм АФРИКАНСКИЙ СЛОН наследует от фрейма СЛОН значение СЕРЫЙ характеристики ЦВЕТ (рис. 2). Значения характеристик фреймов могут передаваться по умолчанию фреймам, находящимся ниже них в иерархии, но если последние содержат собственные значения данных характеристик, то в качестве истинных принимаются именно они. Это обстоятельство позволяет легко учитывать во фреймовых системах различного рода исключения. В частности, во фрейме АЗИАТСКИЙ СЛОН значением слота ЦВЕТ будет КОРИЧНЕВЫЙ, а не СЕРЫЙ, которое могло бы в нем находиться, если бы предпочтение при выборе отдавалось не собственному значению, а наследуемому от фрейма СЛОН.

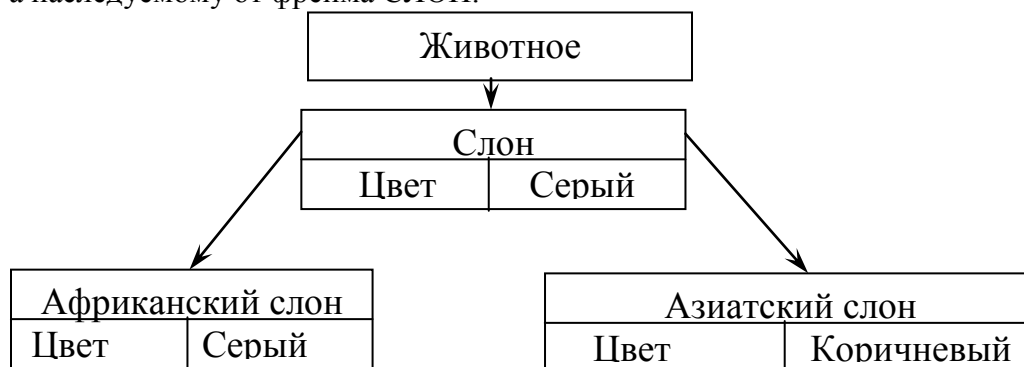


Рис.2 Иерархическая структура совокупности фреймов

Фреймы делят на различные группы – фреймы-описания и ролевые фреймы; символические и конкретные.

Фрейм-описание:

ФРУКТЫ: виноград (болгарский, 20 т)
яблоки (джонатан, 10 т)
вишня (владимирская, 200 кг)).

Ролевой фрейм:

ПЕРЕВЕЗТИ: что (прокат, 200кг)
откуда (Алчевск)
куда (Москву)
чем (железнодорожным транспортом)
когда (в ноябре 2008 г.)).

Символический фрейм:

«РЕЗИСТОР; номинал = X1; мощность = X2; класс точности = X3; тип конструкции = X4; ГОСТ = X5»,

где X1 ÷ X5 – переменные, принимающие различные значения.

Конкретный фрейм:

«АРМ; тип = АРМ2-05; заводской номер = 37; операционная система = Windows; назначение = АРМ конструктора; структура предприятия; вычислительная сеть САПР»,
где «структура предприятия» и «вычислительная сеть САПР» - ссылки на другие фреймы.

На рисунке 3 изображена простейшая иерархическая структура, в которой каждый фрейм имеет только один суперкласс.

Каждый подкласс или экземпляр класса наследует слоты своего суперкласса. Если подкласс (экземпляр класса) и суперкласс имеют слоты с совпадающими именами, то определения значений слотов, сделанные внутри подкласса (экземпляра класса), перекрывают определения суперкласса. Например, ответ на вопрос: «Способен ли пингвин Федя летать?» будет отрицательным. При поиске ответа на этот вопрос фрейм-экземпляр «пингвин Федя» наследует все слоты фрейма «пингины». Значение слота «способность летать» фрейма «пингины» перекрывает значение одноименного слота «птицы».

ЗАДАНИЕ ДЛЯ САМОСТОЯТЕЛЬНОГО ВЫПОЛНЕНИЯ:

Для выбранной самостоятельно предметной области разработать фреймы в виде взаимосвязанных таблиц, семантической сети и сложной иерархической структуры.

Электрогидравлические усилители мощности

Гидравлические усилители мощности

Электрические усилители мощности

Датчики для САУ

Микропроцессоры

Микроконтроллеры

Индикаторы

Резисторы

Конденсаторы

Транзисторы

Тиристоры

Диоды

Интегральные микросхемы

Аналоговые микросхемы

Источники электропитания

Структура БИТТУ

Видеоигры

Видеокамеры

Вузы Балакова

Ваши преподаватели

Предприятия Балакова

Библиотека БИТТУ

Отечественные автомобили

Комплекующие ПК

2.6 Лабораторная работа №6 (2 часа).

Тема: «Нейросети Хемминга и Хопфилда»

2.6.1 Цель работы: изучение процесса разработки генетического алгоритма

2.6.2 Задачи работы:

1. Разработка генетического алгоритма настройки искусственной нейронной сети\
2. Анализ современных методов искусственного интеллекта, включающих генетические алгоритмы и искусственные нейронные сети.

2.7.3 Перечень приборов, материалов, используемых в лабораторной работе:

1. ПЭВМ

2.7.4 Описание (ход) работы:

Постановка задачи оптимизации включает в себя множество допустимых решений $X = \{x_1, x_2, \dots, x_n\}$ и числовую функцию, определенную на этом множестве, которая называется целевой функцией.

Нельзя отождествлять критерий (критерии) оптимальности и целевую функцию.

Целевая функция – это аналитическая зависимость между критерием (критериями) оптимальности и подлежащими оптимизации параметрами с указанием направления экстремума.

Отличие понятий «критерий» и «целевая функция» состоит в следующем:

1. Целевая функция может включать в себя более одного критерия.
2. Для целевой функции всегда и обязательно указывается вид экстремума:

Различают два вида задач оптимизации:

1. Задачу минимизации.
2. Задачу максимизации.

Чтобы решить задачу минимизации функции на множестве, необходимо найти такой вектор (а также соответствующее значение целевой функции), чтобы неравенство: выполнялось для всех. При этом называют оптимальным решением (точнее здесь – минимальным решением), а - оптимумом (минимумом).

Чтобы решить задачу максимизации функции на множестве, необходимо найти такой вектор (а также соответствующее значение целевой функции), чтобы неравенство: выполнялось для всех. При этом называют оптимальным (максимальным) решением, а – оптимумом (максимумом).

В общем виде находится именно вектор, т.к., например, при решении двухпараметрической задачи, он будет включать в себя два параметра, трехпараметрической – три параметра и т.д.

Рассмотрим Диофантово (только целые решения) уравнение: $a+2b+3c+4d=30$, где a, b, c и d - некоторые положительные целые. Применение ГА за очень короткое время находит искомое решение (a, b, c, d).

Конечно, Вы можете спросить: почему бы не использовать метод грубой силы: просто не подставить все возможные значения a, b, c, d (очевидно, $1 \leq a, b, c, d \leq 30$) ?

Архитектура ГА-систем позволяет найти решение быстрее за счет более 'осмысленного' перебора. Мы не перебираем все подряд, но приближаемся от случайно выбранных решений к лучшим.

Для начала выберем 5 случайных решений: $1 \leq a, b, c, d \leq 30$. Вообще говоря, мы можем использовать меньшее ограничение для b, c, d , но для упрощения пусть будет 30 (таблица 6).

Таблица 6 – Первое поколение хромосом

Хромосома	(a,b,c,d)
1	(1,28,15,3)
2	(14,9,2,4)
3	(13,5,7,3)
4	(23,8,16,19)
5	(9,13,5,2)

Чтобы вычислить коэффициенты выживаемости (fitness), подставим каждое решение в выражение $a+2b+3c+4d$. Расстояние от полученного значения до 30 и будет нужным значением (таблица 7).

Таблица 7 - Коэффициенты выживаемости первого поколения хромосом (набора решений)

Хромосома	Коэффициент выживаемости
1	$ 114-30 =84$
2	$ 54-30 =24$
3	$ 56-30 =26$
4	$ 163-30 =133$
5	$ 58-30 =28$

Так как меньшие значения ближе к 30, то они более желательны. В нашем случае большие численные значения коэффициентов выживаемости подходят, увы, меньше. Чтобы создать систему, где хромосомы с более подходящими значениями имеют большие шансы оказаться родителями, мы должны вычислить, с какой вероятностью (в %) может быть выбрана каждая. Одно решение заключается в том, чтобы взять сумму обратных значений коэффициентов, и исходя из этого вычислять проценты (таблица 8). Все решения были сгенерированы Генератором Случайных Чисел – ГСЧ.

Таблица 8 - Вероятность оказаться родителем

Хромосома	Подходимость
1	$(1/84)/0.135266 = 8.80\%$
2	$(1/24)/0.135266 = 30.8\%$
3	$(1/26)/0.135266 = 28.4\%$
4	$(1/133)/0.135266 = 5.56\%$
5	$(1/28)/0.135266 = 26.4\%$

Для выбора 5-и пар родителей (каждая из которых будет иметь 1 потомка, всего - 5 новых решений), представим, что у нас есть 10000-сторонняя игральная кость, на 880 сторонах отмечена хромосома 1, на 3080 - хромосома 2, на 2640 сторонах - хромосома 3, на 556 - хромосома 4 и на 2640 сторонах отмечена хромосома 5. Чтобы выбрать первую пару, кидаем кость два раза и выбираем выпавшие хромосомы. Таким же образом выбирая остальных, получаем результат в таблице 9.

Таблица 9 - Симуляция выбора родителей

Хромосома отца	Хромосома матери
3	1
5	2
3	5
2	5
5	3

Каждый потомок содержит информацию о генах и отца и от матери. Вообще говоря, это можно обеспечить различными способами, однако в нашем случае можно использовать т.н. "кроссовер" (cross-over). Пусть мать содержит следующий набор решений: a1,b1,c1,d1, а отец - a2,b2,c2,d2, тогда возможно 6 различных кроссоверов (| = разделительная линия) (таблица 10):

Таблица 10 - Кроссоверы между родителями

Хромосома-отец	Хромосома-мать	Хромосома-потомок
a1 b1,c1,d1	a2 b2,c2,d2	a1,b2,c2,d2 or a2,b1,c1,d1
a1,b1 c1,d1	a2,b2 c2,d2	a1,b1,c2,d2 or a2,b2,c1,d1
a1,b1,c1 d1	a2,b2,c2 d2	a1,b1,c1,d2 or a2,b2,c2,d1

Есть достаточно много путей передачи информации потомку, и кроссовер - только один из них. Расположение разделителя может быть абсолютно произвольным, как и то, отец или мать будут слева от черты.

А теперь попробуем проделать это с нашими потомками и выведем результаты в таблицу 11.

Таблица 11 - Симуляция кроссоверов хромосом родителей

Хромосома-отец	Хромосома-мать	Хромосома-потомок
(13 5,7,3)	(1 28,15,3)	(13,28,15,3)
(9,13 5,2)	(14,9 2,4)	(9,13,2,4)
(13,5,7 3)	(9,13,5 2)	(13,5,7,2)
(14 9,2,4)	(9 13,5,2)	(14,13,5,2)
(13,5 7, 3)	(9,13 5, 2)	(13,5,5,2)

Теперь мы можем вычислить коэффициенты выживаемости (fitness) потомков (таблица 12).

Таблица 12 - Коэффициенты выживаемости потомков (fitness)

Хромосома-потомок	Коэффициент выживаемости
(13,28,15,3)	$ 126-30 =96$
(9,13,2,4)	$ 57-30 =27$
(13,5,7,2)	$ 57-30 =22$
(14,13,5,2)	$ 63-30 =33$
(13,5,5,2)	$ 46-30 =16$

Средняя приспособленность (fitness) потомков оказалась 38.8, в то время как у родителей этот коэффициент равнялся 59.4. Следующее поколение может мутировать. Например, мы можем заменить одно из значений какой-нибудь хромосомы на случайное целое от 1 до 30. Продолжая, таким образом, одна хромосома, в конце концов, достигнет коэффициента выживаемости 0, то есть станет решением. Системы с большей популяцией (например, 50 вместо 5-и) сходятся к желаемому уровню (0) более быстро и стабильно.

Работа ГА представляет собой итер-ый процесс, который продолжается до тех пор, пока поколения не перестанут существенно отличаться друг от друга, или не пройдет заданное кол-во поколений или заданное время. Для каждого поколения реализуются отбор, кроссовер (скрещивание) и мутация. Рас-м этот алгоритм.

Шаг 1: генерируется начальная популяция, состоящая из N особей со случайными наборами признаков.

Шаг 2 (борьба за существование): вычисляется *абсолютная приспособленность* каждой особи популяции к условиям среды $f(i)$ и суммарная приспособленность особей популяции, характеризующая приспособленность всей популяции. Затем при **пропорциональном отборе** для каждой особи вычисляется ее *относительный вклад в суммарную приспособленность популяции* $P_s(i)$, т.е. относительные ее абсолютные приспособленности $f(i)$ к суммарной приспособленности всех особей популяции (1):

$$P_s(i) = \frac{f(i)}{\sum_{i=1}^N f(i)} \quad (1)$$

В выражении (1) сразу обращает на себя внимание возможность сравнения абсолютной приспособленности i -й особи $f(i)$ не с суммарной приспособленностью всех особей популяции, а со средней абсолютной приспособленностью особ и популяции (2):

$$\bar{f} = \frac{1}{N} \sum_{i=1}^N f(i) \quad (2)$$

Тогда получим (3):

$$P(i) = \frac{f(i)}{\bar{f}} = \frac{f(i)}{\frac{1}{N} \sum_{i=1}^N f(i)} \quad (3)$$

Если взять логарифм по основанию 2 от выражения (3), то получим **количество информации, содержащееся в признаках особи о том, что она выживет и даст потомство** (4).

$$I(i) = \text{Log}_2 \frac{f(i)}{\bar{f}} \quad (4)$$

Формально **приспособленность** особи представляет собой **количество информации, содержащееся в ее фенотипе о продолжении ее генотипа в последующих поколениях.**