

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ

«БРАТСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Кафедра управления в технических системах

УТВЕРЖДАЮ:

Проректор по учебной работе

_____ Е. И. Луковникова

« _____ » _____ 201 г.

**РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ
ИНТЕЛЛЕКТУАЛЬНЫЕ СИСТЕМЫ
УПРАВЛЕНИЯ ИНФОКОММУНИКАЦИЯМИ**

Б1.В.ДВ.06.02

НАПРАВЛЕНИЕ ПОДГОТОВКИ

11.03.02 Инфокоммуникационные технологии и системы связи

ПРОФИЛЬ ПОДГОТОВКИ

Многоканальные телекоммуникационные системы

Программа академического бакалавриата

Квалификация выпускника: бакалавр

| | |
|--|-----------|
| 1. ПЕРЕЧЕНЬ ПЛАНИРУЕМЫХ РЕЗУЛЬТАТОВ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ, СООТНЕСЕННЫХ С ПЛАНИРУЕМЫМИ РЕЗУЛЬТАТАМИ ОСВОЕНИЯ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ | 3 |
| 2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ | 3 |
| 3. РАСПРЕДЕЛЕНИЕ ОБЪЕМА ДИСЦИПЛИНЫ | 4 |
| 3.1 Распределение объёма дисциплины по формам обучения..... | 4 |
| 3.2 Распределение объёма дисциплины по видам учебных занятий и трудоемкости | 4 |
| 4. СОДЕРЖАНИЕ ДИСЦИПЛИНЫ | 5 |
| 4.1 Распределение разделов дисциплины по видам учебных занятий | 5 |
| 4.2 Содержание дисциплины, структурированное по разделам и темам | 6 |
| 4.3 Лабораторные работы..... | 52 |
| 4.4 Семинары / практические занятия..... | 52 |
| 4.5 Контрольные мероприятия: курсовой проект (курсовая работа), контрольная работа, РГР, реферат..... | 52 |
| 5. МАТРИЦА СООТНЕСЕНИЯ РАЗДЕЛОВ УЧЕБНОЙ ДИСЦИПЛИНЫ К ФОРМИРУЕМЫМ В НИХ КОМПЕТЕНЦИЯМ И ОЦЕНКЕ РЕЗУЛЬТАТОВ ОСВОЕНИЯ ДИСЦИПЛИНЫ | 54 |
| 6. ПЕРЕЧЕНЬ УЧЕБНО-МЕТОДИЧЕСКОГО ОБЕСПЕЧЕНИЯ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ ОБУЧАЮЩИХСЯ ПО ДИСЦИПЛИНЕ | 55 |
| 7. ПЕРЕЧЕНЬ ОСНОВНОЙ И ДОПОЛНИТЕЛЬНОЙ ЛИТЕРАТУРЫ, НЕОБХОДИМОЙ ДЛЯ ОСВОЕНИЯ ДИСЦИПЛИНЫ..... | 55 |
| 8. ПЕРЕЧЕНЬ РЕСУРСОВ ИНФОРМАЦИОННО – ТЕЛЕКОММУНИКАЦИОННОЙ СЕТИ «ИНТЕРНЕТ» НЕОБХОДИМЫХ ДЛЯ ОСВОЕНИЯ ДИСЦИПЛИНЫ | 55 |
| 9. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ..... | 56 |
| 9.1. Методические указания для обучающихся по выполнению лабораторных работ / семинаров / практических работ | 56 |
| 9.2. Методические указания по выполнению контрольной работы..... | 64 |
| 10. ПЕРЕЧЕНЬ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, ИСПОЛЬЗУЕМЫХ ПРИ ОСУЩЕСТВЛЕНИИ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА ПО ДИСЦИПЛИНЕ | 69 |
| 11. ОПИСАНИЕ МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЙ БАЗЫ, НЕОБХОДИМОЙ ДЛЯ ОСУЩЕСТВЛЕНИЯ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА ПО ДИСЦИПЛИНЕ | 69 |
| Приложение 1. Фонд оценочных средств для проведения промежуточной аттестации обучающихся по дисциплине..... | 70 |
| Приложение 2. Аннотация рабочей программы дисциплины | 75 |
| Приложение 3. Протокол о дополнениях и изменениях в рабочей программе | 76 |

1. ПЕРЕЧЕНЬ ПЛАНИРУЕМЫХ РЕЗУЛЬТАТОВ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ, СООТНЕСЕННЫХ С ПЛАНИРУЕМЫМИ РЕЗУЛЬТАТАМИ ОСВОЕНИЯ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ

Вид деятельности выпускника

Дисциплина охватывает круг вопросов, относящихся к проектному виду профессиональной деятельности выпускника в соответствии с компетенцией и видами деятельности, указанными в учебном плане.

Цель дисциплины

Изложение базовых характеристик основных направлений развития систем искусственного интеллекта, моделей и методов представления знаний; умение использовать полученные знания и навыки при проектировании интеллектуальных систем управления инфокоммуникациями (ИСУИ).

Задачи дисциплины

Подготовить обучающихся к самостоятельной работе по решению практических задач, связанных с применением методов и средств искусственного интеллекта для управления инфокоммуникациями, построением экспертных систем инфокоммуникационных сетей с использованием оболочки представления знаний ESWin, моделированием нейронной сети инфокоммуникационного объекта.

| Код компетенции | Содержание компетенций | Перечень планируемых результатов обучения по дисциплине |
|-----------------|--|---|
| 1 | 2 | 3 |
| ПК-11 | умением проводить технико-экономическое обоснование проектных расчетов с использованием современных подходов и методов | знать: - основные принципы применения инфокоммуникационных технологий и требования информационной безопасности; уметь: - проводить технико-экономическое обоснование проектных расчетов с использованием современных подходов и методов; владеть: навыками работы с инструментальным программным обеспечением для построения интеллектуальных систем в инфокоммуникациях. |

2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ

Дисциплина Б1.В.ДВ.06.02 «Интеллектуальные системы управления инфокоммуникациями» относится к вариативной части, дисциплина по выбору.

Дисциплина «Интеллектуальные системы управления инфокоммуникациями» базируется на знаниях, полученных при изучении таких учебных дисциплин, как Б1.Б.06 Математический анализ, Б1.Б.21 Теория вероятности и математическая статистика, Б1.В.ДВ.07.01 Основы теории автоматического управления.

Дисциплина «Интеллектуальные системы управления инфокоммуникациями» представляет основу для изучения дисциплин: Б1.В.15 Проектирование и эксплуатация систем передачи, Б3.В.03(П) Производственная (преддипломная практика), Б3.Б.01 Защита выпускной квалификационной работы, включая подготовку к процедуре защиты и процедуру защиты.

Такое системное междисциплинарное изучение направлено на достижение требуемого ФГОС уровня подготовки по квалификации бакалавр.

4. СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

4.1. Распределение разделов дисциплины по видам учебных занятий

- для очной формы обучения:

| № раздела и темы | Наименование раздела и тема дисциплины | Трудоемкость, (час.) | Виды учебных занятий, включая самостоятельную работу обучающихся и трудоемкость; (час.) | | |
|------------------|--|----------------------|---|----------------------|------------------------------------|
| | | | учебные занятия | | самостоятельная работа обучающихся |
| | | | лекции | практические занятия | |
| 1 | 2 | 3 | 4 | 5 | 6 |
| 1. | Состояние и перспективы развития теории и практики интеллектуальных систем управления инфокоммуникациями | 12 | 3 | - | 9 |
| 1.1. | Понятие интеллектуальности | 4 | 1 | - | 3 |
| 1.2. | Предмет интеллектуальные системы | 8 | 2 | - | 6 |
| 2. | Структура интеллектуально-адаптивных систем | 16 | 3 | 4 | 9 |
| 3. | Методы построения алгоритмов адаптации | 15 | 3 | 4 | 8 |
| 3.1. | Виды моделей | 7,5 | 1,5 | 2 | 4 |
| 3.2. | Методы (алгоритмы) интеллектуальных вычислений | 7,5 | 1,5 | 2 | 4 |
| 4. | Системы прямого управления | 15 | 3 | 4 | 8 |
| 4.1. | Понятие образа | 1,5 | 0,5 | - | 1 |
| 4.2. | Проблема обучения распознаванию образов (ОРО) | 2,5 | 0,5 | - | 2 |
| 4.3. | Геометрический и структурный подходы | 1,5 | 0,5 | - | 1 |
| 4.4. | Систематика методов распознавания | 2,5 | 0,5 | - | 2 |
| 4.5. | Аналитические методы | 7 | 1 | 4 | 2 |
| 5. | Системы непрямого управления | 15 | 3 | 4 | 8 |
| 5.1. | Биологический нейрон | 1,25 | 0,25 | - | 1 |
| 5.2. | Формальный нейрон | 1,25 | 0,25 | - | 1 |
| 5.3. | Виды функций активации | 1,5 | 0,5 | - | 1 |
| 5.4. | Искусственные нейронные сети | 1,25 | 0,25 | - | 1 |
| 5.5. | Архитектура нейронной сети | 1,25 | 0,25 | - | 1 |
| 5.6. | Функционирование нейросети | 2,5 | 0,5 | 1 | 1 |
| 5.7. | Общая схема обучения нейронной сети | 3,5 | 0,5 | 2 | 1 |
| 5.8. | Классификация нейронных сетей | 2,5 | 0,5 | 1 | 1 |
| 6. | Представление знаний в интеллектуальных системах | 9 | 3 | - | 6 |
| 7. | Нейронные сети в интеллектуально-адаптивных системах | 13 | 3 | 4 | 6 |
| 7.1. | Метод роящихся частиц | 5 | 1,0 | 2 | 2 |
| 7.2. | Генетические алгоритмы | 3,5 | 0,5 | 1 | 2 |
| 7.3. | Моделирование отжига (Simulated Annealing) | 4,5 | 1,5 | 1 | 2 |
| 8. | Понятие экспертной системы | 13 | 3 | 4 | 6 |

| | (ЭС). Структура ЭС. Классификации ЭС | | | | |
|------|--|------------|-----------|-----------|-----------|
| 8.1. | Экспертные системы: базовые понятия | 1,5 | 0,5 | - | 1 |
| 8.2. | Классификация экспертных систем | 1,5 | 0,5 | - | 1 |
| 8.3. | Составные части экспертной системы и порядок ее функционирования | 1,5 | 0,5 | - | 1 |
| 8.4. | Функционирование базы знаний экспертной системы | 1,5 | 0,5 | - | 1 |
| 8.5. | Обратный метод логического дедуктивного вывода | 3,5 | 0,5 | 2 | 1 |
| 8.6. | Прямой метод логического дедуктивного вывода | 3,5 | 0,5 | 2 | 1 |
| | ИТОГО | 108 | 24 | 24 | 60 |

4.2. Содержание дисциплины, структурированное по разделам и темам

Раздел 1:

Состояние и перспективы развития теории и практики интеллектуальных систем управления инфокоммуникациями (ИСУИК)

1.1. Понятие интеллектуальности

В 1950 году британский математик Алан Тьюринг опубликовал в журнале «Mind» свою работу «Вычислительная машина и интеллект», в которой описал тест для проверки программы на интеллектуальность. Он предложил поместить исследователя и программу в разные комнаты и до тех пор, пока исследователь не определит, кто за стеной - человек или программа, считать поведение программы разумным. Это было одно из первых определений интеллектуальности, то есть А. Тьюринг предложил называть интеллектуальным такое поведение программы, которое будет моделировать разумное поведение человека.

С тех пор появилось много определений интеллектуальных систем (ИС) и искусственного интеллекта (ИИ). Сам термин ИИ (AI - Artificial Intelligence) был предложен в 1956 году на семинаре в Дартмутском колледже (США). Приведем некоторые из этих определений. Д. Люгер определяет «ИИ как область компьютерных наук, занимающуюся исследованием и автоматизацией разумного поведения».

Другое определение: «ИИ - это одно из направлений информатики, целью которого является разработка аппаратно-программных средств, позволяющих пользователю-непрограммисту ставить и решать свои, традиционно считающиеся интеллектуальными задачи, общаясь с ЭВМ на ограниченном подмножестве естественного языка».

Выделяются следующие трудно формализованные задачи, которые повлияли на развитие интеллектуальных систем и искусственного интеллекта:

- доказательство теорем - поиск доказательства математической теоремы требует не только произвести дедукцию, исходя из гипотез, но также создать интуитивные догадки и гипотезы о том, какие промежуточные утверждения следует доказать для вывода доказательства основной теоремы;

- управление роботами;

- распознавание изображений (машинное зрение) – в традиционном распознавании образов появился хорошо разработанный математический аппарат, и для не очень сложных объектов оказалось возможным строить практически работающие системы классификации по признакам, по аналогии и т. д. В качестве признаков могут рассматриваться любые характеристики распознаваемых объектов. Признаки должны быть инвариантны к ориентации, размеру и вариациям формы объектов. Алфавит признаков придумывается разработчиком си-

стемы. Качество распознавания во многом зависит от того, насколько удачно придуман алфавит признаков.

Распознавание состоит в априорном получении вектора признаков для выделенного на изображении отдельного распознаваемого объекта, и лишь затем в определении того, какому из эталонов этот вектор соответствует.

Ранние традиционные системы распознавания, основывающиеся на последовательной организации процесса распознавания и классификации объектов, эффективно решать задачи восприятия сложной зрительной информации не могли;

- машинный перевод и понимание текстов на естественном языке - началом работ по машинному переводу следует считать 1954 год, когда в США с помощью ЭВМ было переведено шестьдесят фраз. Задача разбора предложения на естественном языке даже с ограниченным словарем в сотню слов является типичной задачей ИИ;

- игровые программы – к числу первых игровых программ можно отнести программу Артура Самуэля по игре в чекерс (американские шашки), написанную в 1947 году, причем в ней использовался ряд основополагающих идей ИИ, таких, как перебор вариантов и самообучение.

Победителем этого состязания стала советская шахматная программа «Каисса» (Каисса - богиня, покровительница шахмат). Эта программа была создана в Москве, в Институте проблем управления Академии наук в команде разработчиков программы-чемпиона, лидерами которой были Владимир Арлазаров, Михаил Донской и Георгий Адельсон-Вельский. «Каисса» показала всему миру способности советских специалистов в области эвристического программирования;

- машинное творчество (синтез музыки, стихотворений, текстов) - для различных музыкальных задач было создано специальное программное обеспечение: системы многоканального сведения; системы обработки звука; системы синтеза звука; системы интерактивной композиции; программы алгоритмической композиции и др.

- экспертные системы – методы ИИ нашли применение при создании автоматических консультирующих систем. До 1968 года исследователи в области ИИ работали на основе общего подхода - упрощения комбинаторики, базирующегося на уменьшении перебора альтернатив исходя из здравого смысла, применения числовых функций оценивания и различных эвристик.

В начале 70-х годов произошел качественный скачок и пришло понимание, что необходимы глубокие знания в соответствующей области и выделение знаний из данных, получаемых от эксперта. Появляются экспертные системы (ЭС), или системы, основанные на знаниях.

ЭС MYCIN (середина 70-х годов, Стэнфордский университет) ставила диагноз при инфекционных заболеваниях крови.

ЭС JUDITH помогает специалистам по гражданским делам и вместе с юристом и с его слов усваивает фактические и юридические предпосылки дела, а затем предлагает рассмотреть различные варианты подходов к разрешению дела.

Разработка инструментальных средств для создания ЭС ведется постоянно. Появляются экспертные системы оболочки, совершенствуются технологии создания ЭС. Язык Пролог (1975-79 годы) становится одним из основных инструментов создания ЭС. Язык CLIPS (С Language Integrated Production System) начал разрабатываться в космическом центре Джонсона NASA в 1984 году.

Нейрокибернетика - разработка машин, демонстрирующих «разумное» поведение, на основе архитектур, напоминающих устройство мозга и называемых нейронными сетями (НС). Одна из ключевых особенностей нейронных сетей состоит в том, что они способны обучаться на основе опыта, полученного в обучающей среде.

В 1957 году Ф. Розенблат изобрел устройство для распознавания на основе НС - перцептрон, который успешно различал буквы алфавита, хотя и отличался высокой чувствительностью к их написанию

НС хорошо подходят для распознавания образов и решения задач классификации, оптимизации и прогнозирования. Поэтому основными областями применения НС являются:

- промышленное производство и робототехника;

- военная промышленность и авиация;
- банки и страховые компании;
- службы безопасности;
- биомедицинская промышленность;
- телевидение и связь; и другие области.

1.2. Предмет ИС

Предметом информатики является обработка информации по известным законам. Предметом интеллектуальных систем является изучение интеллектуальной деятельности человека, подчиняющейся заранее неизвестным законам. Интеллектуальные системы решают все те задачи, что не могут быть решены с помощью алгоритмических методов.

ИС – это адаптивная система, позволяющая строить программы целесообразной деятельности по решению поставленных перед ними задач на основании конкретной ситуации, складывающейся на данный момент в окружающей их среде.

Адаптивная система - это система, которая сохраняет работоспособность при непредвиденных изменениях свойств управляемого объекта, целей управления или окружающей среды путем смены алгоритма функционирования, программы поведения или поиска оптимальных, в некоторых случаях просто эффективных, решений и состояний. Традиционно, по способу адаптации различают самонастраивающиеся, самообучающиеся и самоорганизующиеся системы

ИС решают задачи, которые обладают следующими особенностями:

- в них неизвестен алгоритм решения задач (такие задачи будем называть интеллектуальными задачами);
- в них используется помимо традиционных данных в числовом формате информация в виде изображений, рисунков, знаков, букв, слов, звуков;
- в них предполагается наличие выбора (не существует алгоритма - это значит, что нужно сделать выбор между многими вариантами в условиях неопределенности). Свобода действий является существенной составляющей интеллектуальных задач.

Интеллектуальные робототехнические системы (ИРС) содержат переменную, настраиваемую модель внешнего мира и реальной исполнительской системы с объектом управления. Цель и управляющие воздействия формируются в ИРС на основе знаний о внешней среде, объекте управления и на основе моделирования ситуаций в реальной системе.

ИС должны уметь:

- в наборе фактов распознать существенные;
- из имеющихся фактов и знаний делать выводы;
- проводить самооценку - обладать рефлексией, то есть средствами для оценки результатов собственной работы;
- с помощью подсистем объяснения ИС может ответить на вопрос, почему получен тот или иной результат;
- обобщать, улавливая сходство между имеющимися фактами.

Обучаемость, адаптивность, накопление опыта и знаний - важнейшие свойства ИС.

Центральное понятие в ИС – это знание.

Определение знания:

(философия) Знания есть результат, полученный познанием окружающего мира и его объектов.

(философия) Знания - система суждений с принципиальной и единой организацией, основанная на объективной закономерности.

Знания - это формализованная информация, на которую ссылаются или которую используют в процессе логического вывода (рис. 1.1).

(частный случай п.3.) Под знаниями будем понимать совокупность фактов и правил. Понятие правила, представляющего фрагмент знаний, имеет вид:

если <условие> то <действие>

Например, если X истинно и Y истинно, то Z истинно с достоверностью P.

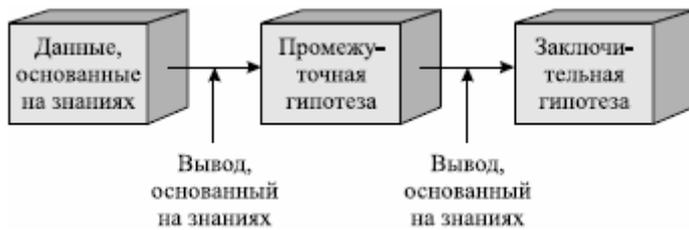


Рис. 1.1. Процесс логического вывода в ИС

Знания, используемые в ИС делятся на две группы:

- статические знания - знания, введенные в ИС на этапе проектирования;
- динамические знания (опыт) - знания, полученные ИС в процессе функционирования или эксплуатации в реальном масштабе времени.

Рассмотрим структуру стандартной интеллектуальной системы:

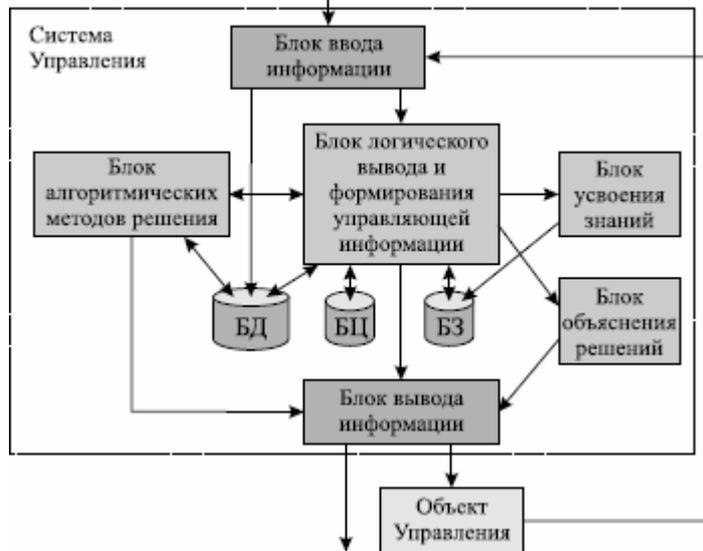


Рис. 1.2. Структура ИС

стрелками обозначено направление движения информации, двунаправленными стрелками обозначено взаимодействие типа «запрос-ответ» и «действие-подтверждение», весьма распространенное в информационных системах. Входом системы является «Блок ввода информации», предназначенный для ввода числовых данных, текста, речи, распознавания изображений. Информация на вход системы может поступать (в зависимости от решаемой задачи) от пользователя, внешней среды, объекта управления. Далее входная информация поступает в «Блок логического вывода», либо сразу в базу данных (БД) – сово-

купность таблиц, хранящих, как правило, символьную и числовую информацию об объектах предметной области.

«Блок логического вывода» (БЛВ) и формирования управляющей информации обеспечивает нахождение решений для нечетко формализованных задач ИС, осуществляет планирование действий и формирование управляющей информации для пользователя или объекта управления на основе Базы Знаний (БЗ), БД, Базы Целей (БЦ) и «Блока Алгоритмических Методов Решений» (БАМР).

БЗ - совокупность знаний, например, система продукционных правил, о закономерностях предметной области.

БЦ - это множество локальных целей системы, представляющих собой совокупность знаний, активизированных в конкретный момент и в конкретной ситуации для достижения глобальной цели.

БАМР содержит программные модули решения задач предметной области по жестким алгоритмам.

«Блок усвоения знаний» (БУЗ) осуществляет анализ динамических знаний с целью их усвоения и сохранения в БЗ.

«Блок объяснения решений» (БОР) интерпретирует пользователю последовательность логического вывода, примененную для достижения текущего результата.

На выходе системы «Блок вывода информации» обеспечивает вывод данных, текста, речи, изображений и другие результаты логического вывода пользователю и/или Объекту Управления (ОУ).

Контур обратной связи позволяет реализовать свойства адаптивности и обучения ИС. На этапе проектирования эксперты и инженеры по знаниям наполняют базу знаний и базу це-

лей, а программисты разрабатывают программы алгоритмических методов решений. База данных создается и пополняется, как правило, в процессе эксплуатации ИС.

Динамика работы ИРС может быть описана следующим образом. При поступлении информации на внешнем языке системы на вход БВИ производится ее интерпретация во внутреннее представление для работы с символьной моделью системы. БЛВ выбирает из БЗ множество правил, активизированных поступившей входной информацией, и помещает эти правила в БЦ как текущие цели системы. Далее БЛВ по заданной стратегии, например, стратегии максимальной достоверности, выбирает правило из БЦ и пытается доопределить переменные модели внешнего мира и исполнительный системы с объектом управления. На основе этого активизируются новые правила БЗ и начинается логический вывод в системе продукций (правил). Эта процедура заканчивается, как только решение будет найдено, либо когда будет исчерпана БЦ. Найденное решение из внутреннего представления интерпретируется «Блоком Вывода информации» во внешний язык подсистемы управления низшего уровня и объекта управления.

Раздел 2:

Структура интеллектуально-адаптивных систем

Лекция проводится в интерактивной форме с текущим контролем (3 час.)

Исследования разработчиков новых интеллектуальных систем направлены на разработку методов соединения человеческого интеллекта и компьютерных систем. В последнее время стремительно развиваются и используются аналитические технологии.

Аналитические технологии - это методики, которые на основе определенных моделей, алгоритмов, математических теорем позволяют по известным данным оценить значения неизвестных характеристик и параметров.

Простейшим примером аналитической технологии является теорема Пифагора, которая позволяет определить длину гипотенузы, имея известные длины катетов по известной формуле $c^2 = a^2 + b^2$.

Другим примером аналитической технологии можно назвать алгоритм обработки информации человеческим мозгом. Даже мозг ребенка может решать задачи, неподвластные современным компьютерам, например, распознавание знакомых лиц в гурьбе или эффективное управление несколькими десятками мышц при игре в футбол. Уникальностью мозга являются способности к решению новых задач - игре в шахматы, вождению автомобиля и т.д. Но при этом, мозг плохо приспособлен к обработке больших объемов числовой информации.

Аналитические технологии нужны в первую очередь людям, принимающим важные решения - руководителям, аналитикам, экспертам, консультантам. Доход компании в большей степени определяется качеством этих решений - точностью прогнозов, оптимальностью избранных стратегий. И от качества этих решений зависит доход компании.

С помощью аналитических технологий можно решать проблемы:

1. прогнозирования (курсов валют, цен на сырье, спроса, дохода компании, уровня безработицы, числа страховых случаев);
2. оптимизации (расписаний, маршрутов, плана закупок, плана инвестиций, стратегии развития).

Для реальных задач бизнеса и производства не существует четких алгоритмов решения и руководители и эксперты решают такие задачи только на основе личного опыта. Часто классические методики оказываются малоэффективными для многих практических задач, поскольку невозможно точно описать реальность с помощью небольшого числа параметров модели, или расчет модели занимает слишком много времени и вычислительных ресурсов. Аналитические технологии позволяют создавать модели, существенным образом повышающие эффективность решений.

Выделяют следующие аналитические технологии:

1. Детерминированные технологии

Аналитические технологии типа теоремы Пифагора используются человеком уже много столетий. За это время было создано огромное количество формул, теорем и алгоритмов для решения классических задач - определения объемов, решения систем линейных уравнений, поиска корней многочленов. Разработаны сложные и эффективные методы для решения задач оптимального управления, решения дифференциальных уравнений и т.д.

Для применения алгоритма необходимо, чтобы данная задача целиком описывалась определенной детерминированной моделью (некоторым набором известных функций и параметров). В таком случае алгоритм дает точный ответ. Например, для применения теоремы Пифагора нужно проверить, что треугольник - прямоугольный.

2. Вероятностные технологии

На практике часто встречаются задачи, связанные с наблюдением случайных величин - например, задача прогнозирования курса акций. Для подобных задач нельзя строить детерминированные модели, поэтому применяется принципиально другой, вероятностный подход.

Параметры вероятностных моделей - это распределения случайных величин, их средние значения, дисперсии и т.д. Как правило, эти параметры заранее неизвестны, а для их оценки используются статистические методы, применяемые к выборкам зафиксированных значений (исторических данных).

Такого рода методы предполагают, что известна некоторая вероятностная модель задачи. Например, в задачи прогнозирования курса можно предположить, что завтрашний курс акций зависит только от курса за последние 2 дня. Если это верно, то наблюдение курса на протяжении нескольких месяцев позволяют довольно точно оценить коэффициенты этой зависимости и прогнозировать курс в будущем.

3. Новые технологии

В последние 10 лет происходит бурное развитие аналитических систем нового типа. В их основе - технологии искусственного интеллекта, имитирующие естественные процессы, например, деятельность нейронов мозга или процесс естественного отбора.

При разработке аналитических технологий учитывается их способность:

- понимания задачи, общего процесса и знания возможностей других систем и людей, принимающих участие во взаимодействии;
- связь с пользователями с помощью понимания естественного языка, рисунков, изображений, и знаков;
- знания, основанные на здравом смысле;
- координирование принятия решений, планирования и действия;
- обучение на предыдущем опыте и адаптация поведения.

Понимание этих возможностей в людях и воплощение их при разработке программ является центральным в созданиях новейших аналитических технологий, способных приобретать и использовать знания.

При добавлении интеллекта к компьютерным системам устраняются многие ограничения в решении реальных задач.

Раздел 3:

Методы построения алгоритмов адаптации

1) Виды моделей интеллектуальных вычислений:

- a) Классификация;
- b) Регрессионный анализ;
- c) Прогнозирование временных последовательностей;
- d) Кластеризация;
- e) Ассоциация;
- f) Последовательность.

2) Методы интеллектуальных вычислений:

- a) Нейронные сети;
- b) Деревья решений;

- c) Системы размышлений на основе аналогичных случаев;
- d) Алгоритмы выявления ассоциаций;
- e) Нечеткая логика;
- f) Генетические алгоритмы;
- g) Эволюционное программирование;
- h) Комбинированные методы.

Виды моделей

Рассмотрим основные виды моделей, которые используются для нахождения нового знания на основе данных информационного хранилища. Целью интеллектуальных технологий является нахождение нового знания, которое пользователь может в дальнейшем применить для улучшения результатов своей деятельности. Результат моделирования - это выявление отношений в данных.

Можно выделить по крайней мере шесть методов выявления и анализа знаний:

- классификация;
- регрессия;
- прогнозирование временных последовательностей (рядов);
- кластеризация;
- ассоциация;
- последовательность.

Первые три используются главным образом для предвидения, в то время как последние удобны для описания существующих закономерностей в данных.

Классификация - самая распространенная модель интеллектуального анализа данных. С ее помощью обнаруживаются признаки, характеризующие группу, к которой принадлежит тот или другой объект. Это делается с помощью анализа уже классифицированных объектов и формулирования некоторого набора правил.

В многих видах бизнеса проблемой является потеря постоянных клиентов.

Классификация помогает обнаружить характеристики «неустойчивых» покупателей и создать модель, предвещающую, кто именно склонен пойти к другому поставщику. Используя ее, можно определить эффективные виды скидок и другие выгодные предложения, действующие для разных покупателей. Благодаря этому, можно удержать клиентов, израсходовав столько денег, сколько необходимо.

Один раз определенный эффективный классификатор используется для классификации новых записей в базе данных в уже существующие классы и в этом случае он приобретает характер прогноза. Например, классификатор, умеющий идентифицировать риск отдачи займа, может быть использован для принятия решения, велик ли риск предоставления займа определенному клиенту. То есть, классификатор используется для прогнозирования вероятности возвращения займа.

Регрессионный анализ используется, когда отношения между переменными могут быть выражены количественно в виде некоторой комбинации этих переменных. Полученная комбинация используется для предвидения значения, которое может принимать целевая (зависимая) переменная, вычисляемая на заданном наборе значений входных (независимых) переменных.

В простейшем случае, для этого используются стандартные статистические методы, такие как линейная регрессия, но большинство реальных моделей не укладываются в ее рамки. Например, размеры продаж или фондовые цены сложны для предвидения, так как могут зависеть от комплекса взаимоотношений переменных.

Прогнозирование временных последовательностей. Основой для любых систем прогнозирования служит историческая информация, хранимая в информационных хранилищах в виде временных рядов. Если можно построить математическую модель и найти шаблоны, адекватно отображающие эту динамику, есть вероятность, что с их помощью можно предвидеть и поведение системы в будущем. Прогнозирование временных последовательностей позволяет на основе анализа поведения временных рядов оценивать будущие значения прогнозируемых переменных. Эти модели должны включать в себя особые признаки времени: иерархия периодов (месяц-квартал-год), особые отрезки времени (пяти- шести или семидневная рабочая неделя), сезонность, праздники и др.

Кластеризация отличается от классификации тем, что классы заранее не заданы и с помощью модели кластеризации средства интеллектуальных вычислений самостоятельно создают однородные группы данных.

Ассоциация относится к анализу структуры и применяется, когда несколько событий связаны между собой. Классический пример анализа структуры покупок относится к представлению приобретения какого-нибудь количества товаров как одиночной экономической операции (транзакции). Поскольку большое количество покупок происходит в супермаркетах, а покупатели для удобства используют корзины, куда и складывается весь товар, то для нахождения ассоциаций служит анализ содержимого корзины. Целью подхода является нахождение трендов (одинаковых участков) среди большого числа транзакций, которые можно использовать для объяснения поведения покупателей. Такая информация может быть использована для регулирования запасов, изменения размещения товаров на территории магазина и принятия решения по проведению рекламной кампании для увеличения продаж или для продвижения определенного вида продукции. Например, исследования, проведенное в супермаркете, может показать, что 65% покупающих картофельные чипсы, берут также и «кока-колу», а при наличии скидки за такой комплект «колу» приобретают в 85% случаев. Имея такие данные, менеджерам легко оценить, насколько действующая предоставленная скидка.

Хотя этот подход пришел исключительно из розничной торговли, он может также применяться в финансовой сфере для анализа портфеля ценных бумаг и нахождения наборов финансовых услуг, которые клиенты часто приобретают вместе. Это может использоваться для создания некоторого набора услуг, как части кампании по стимулированию продаж.

Последовательность имеет место, если существует цепочка связанных во времени событий. Традиционный анализ структуры покупок имеет дело с набором товаров, которые представляют одну транзакцию. Вариант такого анализа встречается, если существует дополнительная информация (номер кредитной карты клиента или номер его банковского счета) для связывания разных покупок в единую временную серию. В такой ситуации важно не только сосуществование данных внутри одной транзакции, но и порядок, в котором эти данные появляются в разных транзакциях и время между этими транзакциями. Правила, устанавливающие эти отношения, могут быть использованы для определения типового набора предшествующих продаж, которые могут повести за собой следующие продажи определенного товара. После покупки дома в 45% случаев в продолжение месяца покупается и новая кухонная плита, а в течении следующих двух недель 60% новоселов обзаводятся холодильником.

Методы (алгоритмы) интеллектуальных вычислений

- нейронные сети;
- деревья решений;
- системы размышлений на основе аналогичных случаев;
- алгоритмы определения ассоциаций и последовательностей;
- нечеткая логика;
- генетические алгоритмы;
- эволюционное программирование;
- визуализация данных.

Иногда применяется комбинация перечисленных методов.

Нейронные сети это системы с архитектурой, условно имитирующей работу нейронов. Математическая модель нейрона представляет собой некоторый универсальный нелинейный элемент с возможностью широкого изменения и настраивания его характеристик.

Нейронные сети представляют собой совокупность связанных между собой слоев нейронов, которые получают входные данные, осуществляют их обработку и генерируют на выходе результат. Между узлами видимых входного и выходного слоев может находиться определенное число скрытых слоев. Нейронные сети реализуют непрозрачный процесс. Это означает, что построенная модель, как правило, не имеет четкой интерпретации. Много пакетов, которые реализуют алгоритмы нейронных сетей, применяются в сфере обработки коммерческой информации, при распознавании образов, расшифровки рукописного текста, интерпретации кардиограмм.

Аппаратные или программные реализации алгоритмов нейромерей называются нейрокомпьютером.

- нейрокомпьютеры дают стандартный способ решения многих нестандартных задач. И неважно, что специализированная машина лучше решит один класс задач. Важнее, что один нейрокомпьютер решит и эту задачу, и другую, и третью и не надо каждый раз проектировать специализированную ЭВМ, нейрокомпьютер сделает все сам и почти не хуже;

- вместо программирования обучения. Нейрокомпьютер учится, нужно лишь формировать обучающие множества. Работа программиста заменяется новой работой учителя. Лучше это ли хуже? Не то, ни другое. Программист приказывает машине все детали работы, учитель создает «обучающую среду», к которой приспосабливается нейрокомпьютер. Появляются новые возможности для работы;

- нейрокомпьютеры эффективны там, где нужен аналог человеческой интуиции, в частности, для распознавания образов, чтения рукописных текстов, подготовки аналитических прогнозов, перевода с одного языка на другой и т.п. Именно для таких задач обычно трудно составить явный алгоритм;

- нейронные сети разрешают создать эффективное программное и математическое обеспечения для компьютеров с высокой степенью распараллеливания обработки;

- нейрокомпьютеры «демократичны», они просты, как текстовые процессоры, поэтому с ними может работать любой, даже совсем неопытный пользователь.

Деревья решений - метод, широко применяемый в области финансов и бизнеса, где чаще встречаются задачи числового прогноза. В результате применения этого метода, для обучающей выборки данных создается иерархическая структура правил классификации типа, «ЕСЛИ... ТОГДА...», имеющих вид дерева. Для того чтобы решить, к какому классу отнести некоторый объект или ситуацию, надо ответить на вопрос, стоящий в узлах этого дерева, начиная с его корня. Вопросы могут иметь вид «Значение параметра А больше Х ?» или вида «Значение переменной В принадлежит подмножеству признаков С ?». Если ответ положительный, переход к правому узлу следующего уровня, если отрицательный - то к левому узлу; потом снова ответ на вопрос, связанный с соответствующим узлом. Таким образом, в конце концов, можно дойти до одного из конечных узлов, где определен класс объекта. Этот метод хорош наглядным представлением правил и его легко понять.

Сегодня наблюдается подъем интереса к продуктам, применяющим деревья решений. В основном это объясняется тем, что большинство коммерческих проблем решается ими быстрее, чем алгоритмами нейронных мерей, они проще и понятней для пользователей.

В то же время нельзя сказать, что деревья решений всегда действуют безотказно: для определенных типов данных они могут оказаться неприемлемыми. Дело в том, что отдельным узлам на каждой ветви отводится меньшее число записей данных - дерево может сегментировать данные на большое количество частных случаев. Чем больше таких частных случаев, тем меньше обучающих примеров попадает в каждый такой частный случай, и их классификация становится менее надежной. Если дерево слишком «ветвистое» - состоит из неоправданно большого числа мелких ветвей - оно не будет давать статистически обоснованных ответов. Как показывает практика, у большинства систем, использующих деревья решений, эта проблема не находит удовлетворительного решения.

Системы размышлений на основе аналогичных случаев

Идея алгоритма крайне проста. Для того, чтобы сделать прогноз на будущее или выбрать правильное решение, системы находят в прошлом близкие аналоги имеющейся ситуации и выбирают тот же ответ, что был для них правильным. Поэтому, этот метод еще называют методом «ближайшего соседа». Системы размышлений на основе аналогичных случаев дают хорошие результаты в разных задачах.

Главный их минус в том, что они вообще не создают каких-нибудь моделей или правил, обобщающих предыдущий опыт, - в выборе решения они базируются на всем массиве доступных исторических данных, поэтому невозможно сказать, на основе каких конкретно факторов эти системы строят свои ответы.

Алгоритмы выявления ассоциаций находят правила об отдельных предметах, которые появляются вместе в одной транзакции, например в одной покупке.

Последовательность - эта тоже ассоциация, но зависящая от времени.

Ассоциация записывается как $A \rightarrow B$, где A называется предпосылкой, B - следствием.

Частота появления каждого отдельного предмета или группы предметов, определяется очень просто - подсчитывается количество появления этого предмета во всех событиях (покупках) и делится на общее количество событий. Эта величина измеряется в процентах и носит название «распространенность». Низкий уровень распространенности (менее одного тысячной процента) говорит о несущественности ассоциации.

Для определения важности каждого полученного ассоциативного правила необходимо получить величину, которая носит название «доверительность A к B » (взаимосвязь A и B). Эта величина показывает, как часто с появлением A появляется B и рассчитывается как отношение частоты появления (распространенности) A и B вместе к распространенности A . То есть, если доверительность A к B равняется 20%, то это означает, что при покупке товара A в каждом пятом случае приобретают и товар B . Если распространенность A не равна распространенности B , то и доверительность A к B не равняется доверительности B к A . В самом деле, покупка компьютера чаще ведет к покупке дискет, чем покупка дискеты к покупке компьютера.

Еще одной важной характеристикой ассоциации является мощность ассоциации. Чем больше мощность, то сильнее влияние, которое появление A делает на появление B . Мощность рассчитывается по формуле: (доверительность A к B) / (распространенность B).

Некоторые алгоритмы поиска ассоциаций сначала сортируют данные и только после этого определяют взаимосвязь и распространенность. Единственным расхождением таких алгоритмов является скорость или эффективность нахождения ассоциаций. Это важно, в связи с огромным количеством комбинаций, что необходимо перебрать для нахождения более значимых правил. Алгоритмы поиска ассоциаций могут создавать свои базы данных распространенности, доверительности и мощности, к которым можно обращаться при запросе. Например: «Найти все ассоциации, в которых для товара X доверительность более 50% и распространенность не менее 2,5%».

При нахождении последовательностей прибавляется переменная времени, позволяющая работать с серией событий для нахождения последовательных ассоциаций на протяжении некоторого периода времени.

Подводя итоги этому методу анализа, необходимо сказать, что случайно может возникнуть такая ситуация, когда товары в супермаркете будут сгруппированы с помощью найденных моделей, но это, вместо ожидаемой прибыли, даст обратный эффект. Это может произойти из-за того, что клиент не будет долго ходить по магазину в поисках желаемого товара, покупая при этом еще что-то, что попадает на глаза, и то, что он никогда не планировал купить.

Нечеткая логика применяется для наборов данных, где принадлежность данных к какой-нибудь группе является вероятностью в интервале от 0 до 1. Четкая логика манипулирует результатами, которые могут быть или истиной, или ложью. Нечеткая логика применяется в тех случаях, когда существует «может быть» в дополнении к «да» или «нет».

Областью внедрения алгоритмов нечеткой логики являются любые аналитические системы, в том числе:

- нелинейный контроль за процессами (производство);
- усовершенствование стратегий управления и координации действий, например сложное промышленное производство;
- самообучающиеся системы (или классификаторы);
- исследование рискованных и критических ситуаций;
- распознавание образов;
- финансовый анализ (рынки ценных бумаг);
- исследование данных (корпоративные хранилища).

В Японии это направление переживает бум. Здесь функционирует специально созданная лаборатория Laboratory for International Fuzzy Engineering Research (LIFE). Программой организации является создание более близких к человеку вычислительных устройств. LIFE объединяет 48 компаний в числе которых находятся: Hitachi, Mitsubishi, NEC, Sharp, Sony, Honda, Mazda, Toyota. Из иностранных участников LIFE можно выделить: IBM, Fuji Xerox, к деятельности LIFE также проявляет интерес NASA.

Мощность и интуитивная простота нечеткой логики как методологии решения проблем гарантирует ее успешное использование во встроенных системах контроля и анализа информации. При этом происходит подключение человеческой интуиции и опыта оператора.

В отличие от традиционной математики, которая требует на каждом шаге моделирования точных и однозначных формулировок закономерностей, нечеткая логика предлагает совсем другой уровень мышления, благодаря чему творческий процесс моделирования происходит на высоком уровне абстракций, при котором постулируется лишь минимальный набор закономерностей.

Недостатками нечетких систем является:

- отсутствие стандартной методики конструирования нечетких систем;
- невозможность математического анализа нечетких систем существующими методами.

Генетические алгоритмы являются мощным средством решения разных комбинаторных задач и задач оптимизации. Тем не менее, генетические алгоритмы вошли сейчас в стандартный инструментарий методов интеллектуальных вычислений. Этот метод назван так потому, что в какой-то степени имитирует процесс естественного отбора в природе. Пусть нам надо найти решения задачи, наиболее оптимальные с точки зрения некоторого критерия, где каждое решение целиком описывается определенным набором чисел или величин нечисловой природы. Скажем, если нам надо выбрать совокупность фиксированного числа параметров рынка, существенно влияющих на его динамику, это будет набор имен этих параметров. Об этом наборе можно говорить как о совокупности хромосом, определяющих качества индивида - данного решения поставленной задачи. Значения параметров, определяющих решения, называются генами. Поиск оптимального решения при этом похож на эволюцию популяции индивидов, представленных наборами хромосом.

В эволюции действуют три механизма: во-первых, отбор сильнейших - наборов хромосом, которым отвечают наиболее оптимальные решения; во-вторых, скрещивание - производство новых индивидов с помощью смешивания хромосомных наборов отобранных индивидов; и, в-третьих, мутации - случайные изменения генов у некоторых индивидов популяции. В результате изменения поколений вырабатывается решение поставленной задачи, которое уже не может быть дальше улучшено.

Генетические алгоритмы имеют два слабых места. Во-первых, постановка задачи не дает возможности проанализировать статистическую значимость получаемого с их помощью решения и, во-вторых, эффективно сформулировать задачу, определить критерий отбора хромосом по силам только специалисту. В силу этих факторов, генетические алгоритмы надо рассматривать скорее как инструмент научного исследования, чем средство анализа данных для практического применения в бизнесе и финансах.

Эволюционное программирование самая молодая область интеллектуальных вычислений. Гипотезы о виде зависимости целевой переменной от других переменных формулируются системой в виде программ на некотором внутреннем языке программирования. Если это универсальный язык, то теоретически на ней можно выразить зависимость любого вида. Процесс построения таких программ строится как эволюция в мире программ (этим метод немного похож на генетические алгоритмы). Если система находит программу, которая точно выражает искомую зависимость, она начинает вносить в нее небольшие модификации и отбирает среди построенных таким образом дочерних программ те, которые повышают точность. Система «выращивает» несколько генетических линий программ, конкурирующих между собой в точности нахождения искомой зависимости. Специальный транслирующий модуль, переводит найденные зависимости с внутреннего языка системы на понятный пользователю язык (математические формулы, таблицы и др.), делая их легкодоступными. Для того, чтобы сделать полученные результаты более понятными для пользователя-нематематика, существует большой арсенал разнообразных средств визуализации выявленных зависимостей.

Поиск зависимости целевых переменных от других проводится в форме функций какого-нибудь определенного вида. Например, в одном из наиболее удачных алгоритмов этого типа - методе группового учета аргументов (МГУА) зависимость ищут в форме полиномов. Причем сложные полиномы заменяются несколькими простыми, учитывающих лишь некоторые признаки (группы аргументов). Обычно используются попарные объединения призна-

ков. Этот метод не имеет больших преимуществ по сравнению с нейронными сетями с готовым набором стандартных нелинейных функций, но, полученные формулы зависимости, в принципе, поддаются анализу и интерпретации (хотя на практике это все-таки сложно).

Программы визуализации данных в определенном смысле не является средством анализа информации, поскольку они только представляют ее пользователю. Тем не менее, визуальное представление, скажем, сразу четырех переменных наглядно обобщает огромнейшие объемы данных.

Комбинированные методы. Часто производители соединяют указанные подходы. Объединение алгоритмов нейронных сетей и технологии деревьев решений способствует построению более точной модели и повышению быстродействия. Для решения каждой проблемы следует искать свой оптимальный метод.

Раздел 4:

Системы прямого управления

4.1. Понятие образа

Образ, класс – классификационная группировка в системе классификации, объединяющая (выделяющая) определенную группу объектов по некоторому признаку.

Образное восприятие мира – одно из загадочных свойств живого мозга, позволяющее разобраться в бесконечном потоке воспринимаемой информации и сохранять ориентацию в океане разрозненных данных о внешнем мире. Воспринимая внешний мир, мы всегда производим классификацию воспринимаемых ощущений, т. е. разбиваем их на группы похожих, но не тождественных явлений. Например, несмотря на существенное различие, к одной группе относятся все буквы А, написанные различными почерками, или все звуки, соответствующие одной и той же ноте, взятой в любой октаве и на любом инструменте, а оператор, управляющий техническим объектом, на целое множество состояний объекта реагирует одной и той же реакцией. Характерно, что для составления понятия о группе восприятий определенного класса достаточно ознакомиться с незначительным количеством ее представителей. Ребенку можно показать всего один раз какую-либо букву, чтобы он смог найти эту букву в тексте, написанном различными шрифтами, или узнать ее, даже если она написана в умышленно искаженном виде. Это свойство мозга позволяет сформулировать такое понятие, как образ.

Образы обладают характерным свойством, проявляющимся в том, что ознакомление с конечным числом явлений из одного и того же множества дает возможность узнавать сколь угодно большое число его представителей. Примерами образов могут быть: река, море, жидкость, музыка Чайковского, стихи Маяковского и т. д. В качестве образа можно рассматривать и некоторую совокупность состояний объекта управления, причем вся эта совокупность состояний характеризуется тем, что для достижения заданной цели требуется одинаковое воздействие на объект. Образы обладают характерными объективными свойствами в том смысле, что разные люди, обучающиеся на различном материале наблюдений, большей частью одинаково и независимо друг от друга классифицируют одни и те же объекты. Именно эта объективность образов позволяет людям всего мира понимать друг друга.

Способность восприятия внешнего мира в форме образов позволяет с определенной достоверностью узнавать бесконечное число объектов на основании ознакомления с конечным их числом, а объективный характер основного свойства образов позволяет моделировать процесс их распознавания. Будучи отражением объективной реальности, понятие образа столь же объективно, как и сама реальность, а поэтому это понятие может быть само по себе объектом специального исследования.

В литературе, посвященной проблеме обучения распознавания образов (ОРО), часто вместо понятия образа вводится понятие класса.

4.2. Проблема обучения распознаванию образов (ОРО)

Одним из самых интересных свойств человеческого мозга является способность отвечать на бесконечное множество состояний внешней среды конечным числом реакций. Может быть, именно это свойство позволило человеку достигнуть высшей формы существования живой материи, выражающейся в способности к мышлению, т. е. активному отражению объективного мира в виде образов, понятий, суждений и т. д. Поэтому проблема ОРО возникла при изучении физиологических свойств мозга.

В целом проблема распознавания образов состоит из двух частей: обучения и распознавания. Обучение осуществляется путем показа отдельных объектов с указанием их принадлежности тому или другому образу. В результате обучения распознающая система должна приобрести способность реагировать одинаковыми реакциями на все объекты одного образа и различными — на все объекты различных образов. Очень важно, что процесс обучения должен завершиться только путем показов конечного числа объектов без каких-либо других подсказок. В качестве объектов обучения могут быть либо картинки, либо другие визуальные изображения (буквы), либо различные явления внешнего мира, например звуки, состояния организма при медицинском диагнозе, состояние технического объекта в системах управления и др. Важно, что в процессе обучения указываются только сами объекты и их принадлежность образу. За обучением следует процесс распознавания новых объектов, который характеризует действия уже обученной системы. Автоматизация этих процедур и составляет проблему обучения распознаванию образов. В том случае, когда человек сам разгадывает или придумывает, а затем навязывает машине правило классификации, проблема распознавания решается частично, так как основную и главную часть проблемы (обучение) человек берет на себя.

Проблема обучения распознаванию образов интересна как с прикладной, так и с принципиальной точки зрения. С прикладной точки зрения решение этой проблемы важно прежде всего потому, что оно открывает возможность автоматизировать многие процессы, которые до сих пор связывали лишь с деятельностью живого мозга. Принципиальное значение проблемы тесно связано с вопросом, который все чаще возникает в связи с развитием идей кибернетики: что может и что принципиально не может делать машина? В какой мере возможности машины могут быть приближены к возможностям живого мозга? В частности, может ли машина развить в себе способность перенять у человека умение производить определенные действия в зависимости от ситуаций, возникающих в окружающей среде? Пока стало ясно только то, что если человек может сначала сам осознать свое умение, а потом его описать, т. е. указать, почему он производит действия в ответ на каждое состояние внешней среды или как (по какому правилу) он объединяет отдельные объекты в образы, то такое умение без принципиальных трудностей может быть передано машине. Если же человек обладает умением, но не может объяснить его, то остается только один путь передачи умения машине — обучение примерами.

Круг задач, которые могут решаться с помощью распознающих систем, чрезвычайно широк. Сюда относятся не только задачи распознавания зрительных и слуховых образов, но и задачи распознавания сложных процессов и явлений, возникающих, например, при выборе целесообразных действий руководителем предприятия или выборе оптимального управления технологическими, экономическими, транспортными или военными операциями. В каждой из таких задач анализируются некоторые явления, процессы, состояния внешнего мира, всюду далее называемые объектами наблюдения. Прежде чем начать анализ какого-либо объекта, нужно получить о нем определенную, каким-либо способом упорядоченную информацию. Такая информация представляет собой характеристику объектов, их отображение на множестве воспринимающих органов распознающей системы.

Но каждый объект наблюдения может воздействовать по-разному, в зависимости от условий восприятия. Например, какая-либо буква, даже одинаково написанная, может в принципе как угодно смещаться относительно воспринимающих органов. Кроме того, объекты одного и того же образа могут достаточно сильно отличаться друг от друга и, естественно, по-разному воздействовать на воспринимающие органы.

Каждое отображение какого-либо объекта на воспринимающие органы распознающей системы, независимо от его положения относительно этих органов, принято называть изображением объекта, а множества таких изображений, объединенные какими-либо общими свойствами, представляют собой образы.

При решении задач управления методами распознавания образов вместо термина «изображение» применяют термин «состояние». Состояние – это определенной формы отображение измеряемых текущих (или мгновенных) характеристик наблюдаемого объекта. Совокупность состояний определяет ситуацию. Понятие «ситуация» является аналогом понятия «образ». Но эта аналогия не полная, так как не всякий образ можно назвать ситуацией, хотя всякую ситуацию можно назвать образом.

Ситуацией принято называть некоторую совокупность состояний сложного объекта, каждая из которых характеризуется одними и теми же или схожими характеристиками объекта. Например, если в качестве объекта наблюдения рассматривается некоторый объект управления, то ситуация объединяет такие состояния этого объекта, в которых следует применять одни и те же управляющие воздействия. Если объектом наблюдения является военная игра, то ситуация объединяет все состояния игры, которые требуют, например, мощного танкового удара при поддержке авиации.

Выбор исходного описания объектов является одной из центральных задач проблемы ОРО. При удачном выборе исходного описания (пространства признаков) задача распознавания может оказаться тривиальной и, наоборот, неудачно выбранное исходное описание может привести либо к очень сложной дальнейшей переработке информации, либо вообще к отсутствию решения. Например, если решается задача распознавания объектов, отличающихся по цвету, а в качестве исходного описания выбраны сигналы, получаемые от датчиков веса, то задача распознавания в принципе не может быть решена.

4.3. Геометрический и структурный подходы

Каждый раз, когда сталкиваются с незнакомыми задачами, появляется естественное желание представить их в виде некоторой легко понимаемой модели, которая позволяла бы осмыслить задачу в таких терминах, которые легко воспроизводятся нашим воображением. А так как мы существуем в пространстве и во времени, наиболее понятной для нас является пространственно-временная интерпретация задач.

Любое изображение, которое возникает в результате наблюдения какого-либо объекта в процессе обучения или экзамена, можно представить в виде вектора, а значит и в виде точки некоторого пространства признаков. Если утверждается, что при показе изображений возможно однозначно отнести их к одному из двух (или нескольких) образов, то тем самым утверждается, что в некотором пространстве существует две (или несколько) области, не имеющие общих точек, и что изображения – точки из этих областей. Каждой такой области можно приписать наименование, т. е. дать название, соответствующее образу.

Проинтерпретируем теперь в терминах геометрической картины процесс обучения распознаванию образов, ограничившись пока случаем распознавания только двух образов. Заранее считается известным лишь только то, что требуется разделить две области в некотором пространстве и что показываются точки только из этих областей. Сами эти области заранее не определены, т. е. нет каких-либо сведений о расположении их границ или правил определения принадлежности точки к той или иной области.

В ходе обучения предъявляются точки, случайно выбранные из этих областей, и сообщается информация о том, к какой области принадлежат предъявляемые точки. Никакой дополнительной информации об этих областях, т. е. о расположении их границ, в ходе обучения не сообщается. Цель обучения состоит либо в построении поверхности, которая разделяла бы не только показанные в процессе обучения точки, но и все остальные точки, принадлежащие этим областям, либо в построении поверхностей, ограничивающих эти области так, чтобы в каждой из них находились только точки одного образа. Иначе говоря, цель обучения состоит в построении таких функций от векторов-изображений, которые были бы, например, положительны на всех точках одного и отрицательны на всех точках другого образа. В связи

с тем, что области не имеют общих точек, всегда существует целое множество таких разделяющих функций, а в результате обучения должна быть построена одна из них.

Если предъявляемые изображения принадлежат не двум, а большему числу образов, то задача состоит в построении по показанным в ходе обучения точкам поверхности, разделяющей все области, соответствующие этим образам, друг от друга. Задача эта может быть решена, например, путем построения функции, принимающей над точками каждой из областей одинаковое значение, а над точками из разных областей значение этой функции должно быть различно.

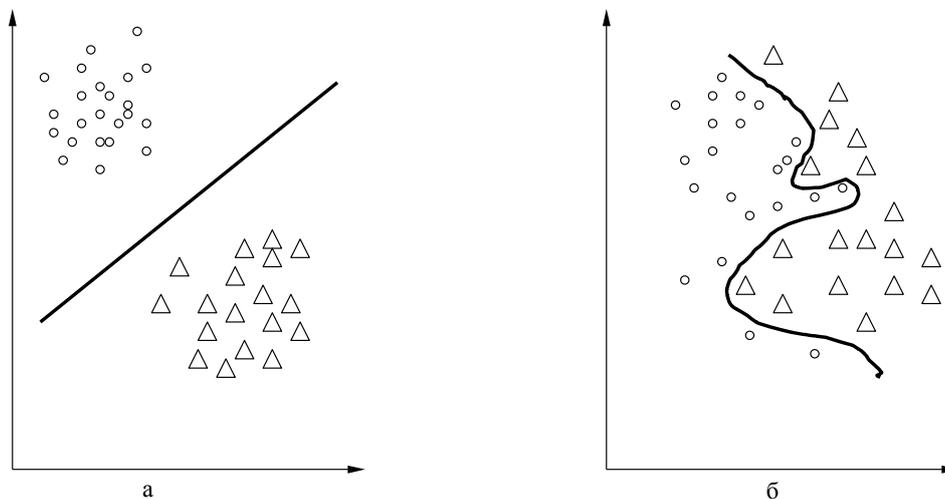


Рис. 4.1. Два образа

На первый взгляд кажется, что знание всего лишь некоторого количества точек из области недостаточно, чтобы отделить всю область. Действительно, можно указать бесчисленное количество различных областей, которые содержат эти точки, и как бы ни была построена по ним поверхность, выделяющая область, всегда можно указать другую область, которая пересекает поверхность и вместе с тем содержит показанные точки. Однако известно, что задача о приближении функции по информации о ней в ограниченном множестве точек, существенно более узкой, чем все множество, на котором функция задана, является обычной математической задачей об аппроксимации функций. Разумеется, решение таких задач требует введения определенных ограничений на классе рассматриваемых функций, а выбор этих ограничений зависит от характера информации, которую может добавить учитель в процессе обучения. Одной из таких подсказок является гипотеза о компактности образов. Интуитивно ясно, что аппроксимация разделяющей функции будет задачей тем более легкой, чем более компактны и чем более разнесены в пространстве области, подлежащие разделению. Так, например, в случае, показанном на Рис. 4.1, а разделение заведомо более просто, чем в случае, показанном на Рис. 4.1, б. Действительно, в случае, изображенном на Рис. 4.1, а, области могут быть разделены плоскостью, и даже при больших погрешностях в определении разделяющей функции она все же будет продолжать разделять области. В случае же на Рис. 4.1, б, разделение осуществляется замысловатой поверхностью и даже незначительные отклонения в ее форме приводят к ошибкам разделения. Именно это интуитивное представление о сравнительно легко делимых областях привело к гипотезе компактности.

Наряду с геометрической интерпретацией проблемы обучения распознаванию образов существует и иной подход, который назван структурным, или лингвистическим. Поясним лингвистический подход на примере распознавания зрительных изображений. Сначала выделяется набор исходных понятий – типичных фрагментов, встречающихся на изображениях, и характеристик взаимного расположения фрагментов – «слева», «снизу», «внутри» и т. д. Эти исходные понятия образуют словарь, позволяющий строить различные логические высказывания, иногда называемые предположениями. Задача состоит в том, чтобы из большого количества высказываний, которые могли бы быть построены с использованием этих понятий, отобрать наиболее существенные для данного конкретного случая.

Далее, просматривая конечное и по возможности небольшое число объектов из каждого образа, нужно построить описание этих образов. Построенные описания должны быть столь полными, чтобы решить вопрос о том, к какому образу принадлежит данный объект. При реализации лингвистического подхода возникают две задачи: задача построения исходного словаря, т. е. набор типичных фрагментов, и задача построения правил описания из элементов заданного словаря.

В рамках лингвистической интерпретации проводится аналогия между структурой изображений и синтаксисом языка. Стремление к этой аналогии было вызвано возможностью использовать аппарат математической лингвистики, т. е. методы по своей природе являются синтаксическими. Использование аппарата математической лингвистики для описания структуры изображений можно применять только после того, как произведена сегментация изображений на составные части, т. е. выработаны слова для описания типичных фрагментов и методы их поиска. После предварительной работы, обеспечивающей выделение слов, возникают собственно лингвистические задачи, состоящие из задач автоматического грамматического разбора описаний для распознавания изображений. При этом проявляется самостоятельная область исследований, которая требует не только знания основ математической лингвистики, но и овладения приемами, которые разработаны специально для лингвистической обработки изображений.

4.4. Систематика методов распознавания



Схема 1. Классификация методов распознавания.

Рассмотрим три группы, относящихся к группе признаков:

1. Логические.

Системы из этой подгруппы используют бинарное представление признаков, то есть признак может принимать только два значения :0 или 1. В этом же заключается основной недостаток подобных систем, так как два образа с выраженностью произвольного при-

знака равной 0,6 и 0,9 будут представлены координатой 1. Основное преимущество логических систем в их быстродействии.

2. Аналитические.

Системы из этой подгруппы характеризуются набором формул и критериев. Основной недостаток систем этой группы - отсутствие единого методического подхода к построению аналитической части. Преимуществом систем этой группы является возможность адаптации к используемому виду и размерности данных соответствующих аналитических выражений.

3. Статистические.

Системы из этой подгруппы используют статистические оценки (среднее значение, дисперсию, статистические критерии и так далее) для реализации процесса классификации. Основной недостаток систем этой группы - зависимость результатов от качества используемых данных. Основное преимущество - при достоверных и однородных данных получается высокий уровень надежности результатов.

4.5. Аналитические методы

В работе для распределения по классам используются аналитические методы распознавания. Рассмотрим подробно эту группу методов.

Метод обобщенного портрета. Его особенности:

- метод обобщенного портрета полагает наличие характеристической точки в пространстве признаков, в координатах которой обобщены признаки конкретного класса;
- в методе обобщенного портрета обязательно наличие некоторой области (области обобщения) вокруг характеристической точки. Эта область и является искомым классом;
- вариантом метода является случай, когда характеристическую точку не указывают, а область, занимаемую классом, ограничивают.

Варианты метода обобщенного портрета:

- в качестве характеристической точки выбирается статистическое значение (среднее значение). Тогда область обобщения – это гиперсфера с центром в характеристической точке и радиусом, равным расстоянию от характеристической точки до самого дальнего «представителя множества».

- характеристическая точка представлена в распределенном виде, то есть для каждого признака существуют экстремальные его значения (максимальное, минимальное), совокупность этих экстремумов заменяют собой характеристическую точку. Таким образом, область обобщения – гиперпараллелепипед, размеры ребер которого равны размаху минимум – максимум по соответствующему признаку.

- характеристическая точка не указывается, а область ограничивается. Необходимое ограничение области выполняется следующим образом: из всего множества точек отбираются те, для каждой которых существуют 2 прямые, соединяющие эту точку с двумя другими точками класса. Причем эти прямые расположены так, что все точки рассматриваемого класса расположены по одну сторону от прямой. В результате вокруг класса можно построить оболочку из отрезков этих прямых. Эта оболочка будет отделять внешний класс от внутреннего. Справедливо и для отделения внешних точек.

Алгоритмы метода обобщенного портрета реализуют идею минимизации эмпирического риска в классе линейных и кусочно-линейных функций.

В общем случае метод обобщенного портрета состоит в специальном способе построения разделяющей гиперплоскости. В случае, если обучающая последовательность может быть разделена гиперплоскостью, существует целое семейство разделяющих гиперплоскостей. Особенность метода заключается в том, что с его помощью строится оптимальная разделяющая гиперплоскость (то есть гиперплоскость, которая из всех разделяющих гиперплоскостей наиболее далеко отстоит от ближайшего к ней элемента последовательности). Важной особенностью метода обобщенного портрета является возможность установить (в случае, если это так), что «безошибочного» разделения элементов последовательности не существует.

Метод построения разделяющей гиперплоскости. Построение разделяющей гиперплоскости используется для определения принадлежности объекта к тому или иному классу образов. Рассмотрим этапы построения разделяющей гиперплоскости.

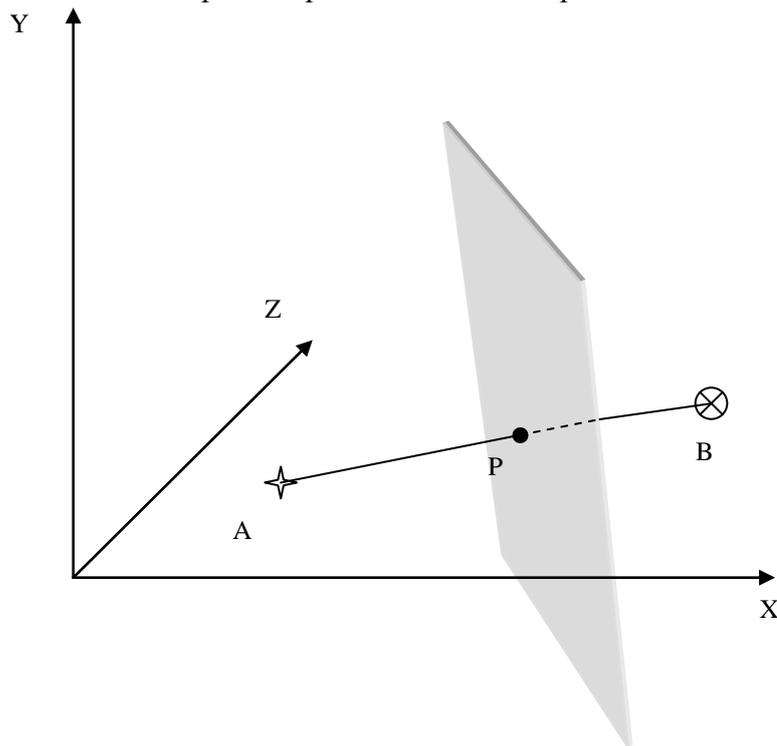


Рис. 4.2. Построение разделяющей гиперплоскости.

На рис. 4.2. схематически изображен процесс построения разделяющей гиперплоскости. Нам известны координаты двух точек(объектов): A, B, которые представлены на рисунке в виде звездочки (точка A), и в виде кружочка (точка B). Необходимо построить разделяющую гиперплоскость, которая отделяла бы образы «похожие» на A, от образов «похожих» на B.

Для построения, нам необходимо провести между точками A и B, соединяющую их линию. На середине этой линии отложим точку P, через которую будет проходить разделяющая гиперплоскость. Теперь, зная координаты точки P, а также имея вектор AB, построить плоскость, проходящую через точку P, и перпендикулярную вектору AB. Теперь чтобы принять решение о принадлежности образа к тому или иному классу, нужно определить расстояние от этого образа до гиперплоскости, кроме того, знак полученного расстояния будет говорить о положении образа с одной или другой стороны от гиперплоскости.

Опишем общее уравнение плоскости:

$$Ax + By + Cz + D = 0$$

Тогда исходя из уравнения плоскости, проходящей через точку, $P(x_1, y_1, z_1)$, перпендикулярно прямой с направляющим вектором $N(A, B, C)$:

$$A(x - x_1) + B(y - y_1) + C(z - z_1) = 0$$

находим коэффициент **D**

$$D = -(Ax_1 + By_1 + Cz_1),$$

где вектор $N(A, B, C)$ – перпендикулярен к плоскости. Направляющие косинусы этого вектора:

$$\cos \alpha = \frac{A}{\sqrt{A^2 + B^2 + C^2}}$$

$$\cos \beta = \frac{B}{\sqrt{A^2 + B^2 + C^2}}$$

$$\cos \gamma = \frac{C}{\sqrt{A^2 + B^2 + C^2}}$$

Подставляя эти значения в формулу для нахождения расстояния от точки $M(a,b,c)$ до плоскости:

$$\delta = a \cos \alpha + b \cos \beta + c \cos \gamma - p; \quad \delta - \text{искомое расстояние}$$

$$p = -\frac{D}{\sqrt{A^2 + B^2 + C^2}}$$

Получаем:

$$\delta = a \frac{A}{\sqrt{A^2 + B^2 + C^2}} + b \frac{B}{\sqrt{A^2 + B^2 + C^2}} + c \frac{C}{\sqrt{A^2 + B^2 + C^2}} + \frac{Ax_1 + By_1 + Cz_1}{\sqrt{A^2 + B^2 + C^2}}$$

Разделяющая гиперплоскость не может быть построена без базисных точек. В качестве базисных точек могут быть выбраны:

- наугад взятые точки, по одной из множества А и множества В;
- точки, выбранные из А и В с учетом определенных обстоятельств (например, имеющие минимальные расстояния между собой);
- точки, выбранные на основании каких-либо статистических характеристик.

Пусть нам даны множества А и В, каждое из которых состоит из 5 элементов (рис. 4.3):

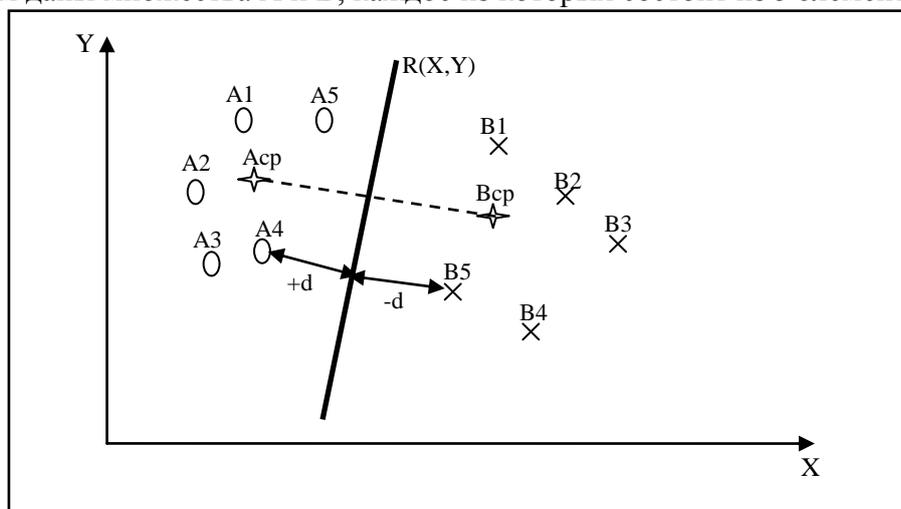


Рис. 4.3. Разделение классов А и В гиперплоскостью

В качестве базисных точек выберем точки, координаты которых являются средними арифметическими значениями соответствующие координат точек каждого класса.

В качестве разделяющей классы А и В гиперплоскости выберем следующую плоскость:

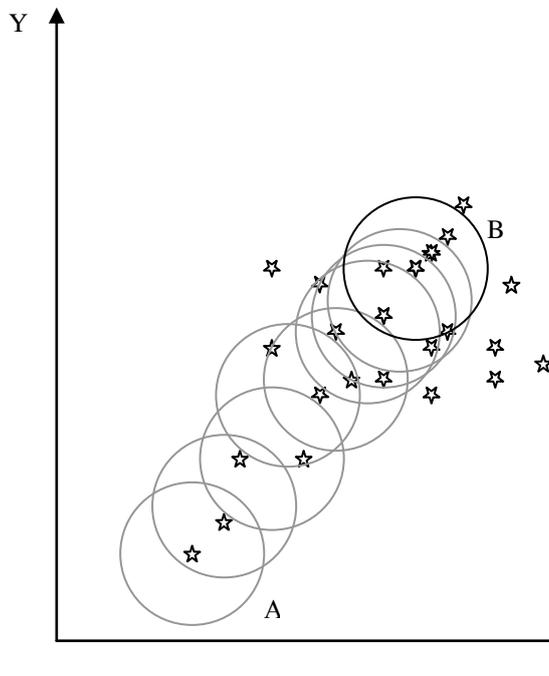
$$R(x,y) = 2(A_{\text{сред}} B_{\text{сред}}) * P + |B_{\text{сред}}|^2 - |A_{\text{сред}}|^2 = 0, \text{ это векторное уравнение.}$$

Отметим, что все расстояния точек A_i от гиперплоскости отрицательны:

Все расстояния точек B_i от разделяющей гиперплоскости положительны, следовательно, $R(x,y)$ разделяет множества А и В и если добавить новый элемент в В, то такое расстояние будет положительно.

Существуют модификации данного метода, более сложные и основываются на построении градиента.

Метод «ФОРЕЛЬ» относится к аналитическим методам. В методе используется критерий, оценивающий плотность распределений образов в объеме, ограниченном сферой фиксированного радиуса. При переходе от шага к шагу центр этой сферы движется в сторону увеличения плотности точек – образов (рис. 4.4). Сфера стабилизируется в определенном положении, когда плотность точек внутри ее становится максимальной, и любое перемещение сферы ведет к ухудшению ситуации. Точки-образы, попадающие внутрь стабилизировавшейся сферы, образуют класс N1. Процесс повторяется с любой точки из оставшихся и завершается новой стабилизацией сферы, внутри которой находится класс N2 и так далее. Количество классов, получаемых при использовании метода, связано с величиной радиуса сферы.



Пример работы алгоритма, исходное положение формального элемента - окружности в точке А. в точке В - стабильное результирующее положение окружности.

Рис. 4.4. Процесс перемещения формального элемента (сферы) по множеству объектов

Алгоритм. Задается радиус сферы R.

1-й шаг.

Строится сфера радиуса R с центром в любой точке так, чтобы внутрь среды попала хотя бы одна точка - образ Z_I . Определяется центр сферы. Этот центр назовем $C^{(1)}$.

2-й шаг.

Определяются точки - образы $Z_I^{(1)}$, для которых $|Z_{I(1)} - C_I^{(1)}| < R$ т.е. эти точки попадают внутрь сферы.

3-й шаг.

Вычисляется центр тяжести таких $Z_I^{(1)}$.

Если внутрь сферы попало k точек, то координаты центра тяжести $X_C = 1/k \sum_{I=1}^k x_I$;

$$Y_C = 1/k \sum_{I=1}^k y_I^{(1)} .$$

Этот центр тяжести назовем $C^{(2)}$.

4-й шаг.

Строится сфера радиуса R с центром в точке $C^{(2)}$. Определяются точки $Z_I^{(2)}$, для которых $|Z_I^{(2)} - C_I^{(2)}| < R$.

5-й шаг.

Вычисляется центр тяжести таких $Z_I^{(2)}$; этот центр тяжести назовем $C^{(3)}$ и так далее.

Процесс «движения» среды, т.е. переход от $C^{(k)}$ к $C^{(k+1)}$ останавливается тогда, когда расстояние $|C^{(k)} - C^{(k+1)}| < \Delta$.

Сфера с центром в $C^{(k+1)}$ представляет собой таксон S1.

6-й шаг.

Точки - образы $\in S1$ из рассмотрения исключаются. В качестве начальной берётся любая точка - образ из оставшихся, и процесс продолжается с шага 1.

7-й шаг.

Последовательность таксонов S1, S2, ..., является последовательностью классов P1, P2, ..., соответствующих радиусу сферы = R.

Замечание. Минимум суммарного объема сфер - таксонов дает рациональное разбиение набора образов на классы.

При всей своей наглядности и интерпретируемости результатов алгоритм «Форель» обладает существенным недостатком: результаты таксономии в большинстве случаев зависят от начального выбора центра гиперсферы и ее радиуса. Существуют различные приёмы, уменьшающие эту зависимость, но радикального решения этой задачи нет.

Метод дробящихся эталонов. Для построения решающих правил нужна обучающая выборка. Обучающая выборка – это множество объектов, заданных значениями признаков и принадлежность которых к тому или иному классу достоверно известна «учителю» и сообщается учителем «обучаемой» системе. По обучающей выборке система строит решающие правила. Качество решающих правил оценивается по контрольной (экзаменационной) выборке, в которую входят объекты, заданные значениями признаков, и принадлежность которых тому или иному образу известна только «учителю».

Метод построения эталонов. Для каждого класса по обучающей выборке строится эталон, имеющий значения признаков

$$\bar{x}^0 = \{x_1^0, x_2^0, \dots, x_N^0\},$$

где $x_i^0 = \frac{1}{K} \sum_{k=1}^K x_{ik}$,

K – количество объектов данного образа в обучающей выборке,

i – номер признака.

По существу, эталон – это усреднённый по обучающей выборке абстрактный объект (рис. 4.5). Абстрактным мы его называем потому, что он может не совпадать не только ни с одним объектом обучающей выборки, но и ни с одним объектом генеральной совокупности.

Распознавание осуществляется следующим образом. На вход системы поступает объект \bar{x}^* , принадлежность которого к тому или иному образу системе неизвестна. От этого объекта измеряются расстояния до эталонов всех образов, и \bar{x}^* система относит к тому образу, расстояние до эталона которого минимально. Расстояние измеряется в той метрике, которая введена для решения определённой задачи распознавания.

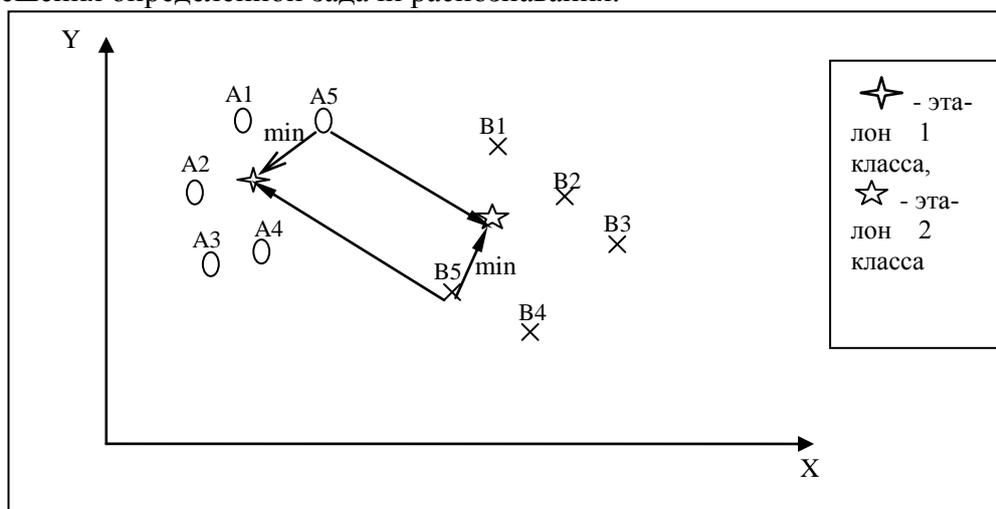


Рис. 4.5. Решающее правило «Минимум расстояния до эталона класса»

Процесс обучения состоит в следующем. На первом этапе в обучающей выборке «охватывают» все объекты каждого класса гиперсферой возможно меньшего радиуса. Сделать это можно, например, так. Строится эталон каждого класса. Вычисляется расстояние от эталона до всех объектов данного класса, входящих в обучающую выборку. Выбирается максимальное из этих расстояний r_{\max} . Строится гиперсфера с центром в эталоне и радиусом $R = r_{\max} + \varepsilon$. Она охватывает все объекты данного класса. Такая процедура проводится для всех классов (образов). На рис. 4.6 приведён пример двух образов в двухмерном признаковом пространстве.

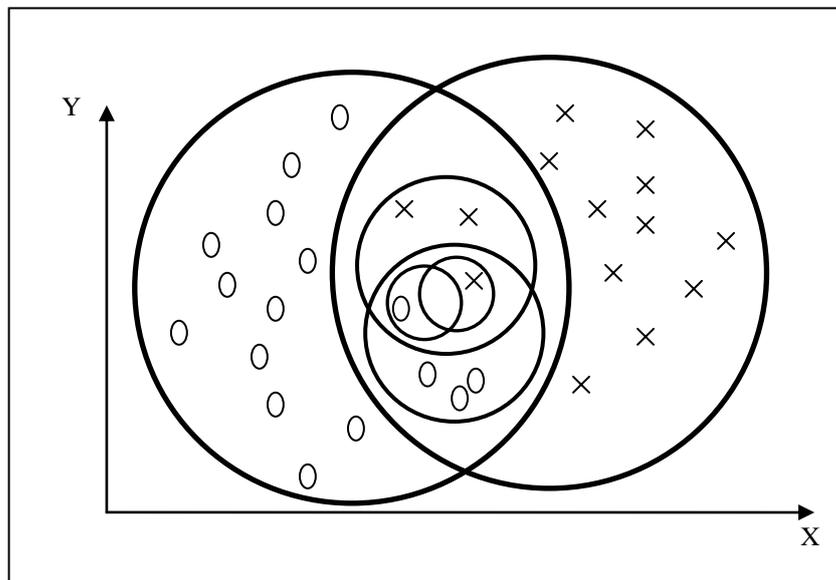


Рис. 4.6. Решающее правило типа «Метод дробящихся эталонов»

Если гиперсферы различных образов пересекаются и в области перекрытия оказываются объекты более чем одного образа, то для них строятся гиперсферы второго уровня, затем третьего до тех пор, пока области не окажутся непересекающимися, либо в области пересечения будут присутствовать объекты только одного образа.

Распознавание осуществляется следующим образом. Определяется местонахождение объекта относительно гиперсфер первого уровня. При попадании объекта в гиперсферу, соответствующую одному и только одному образу, процедура распознавания прекращается. Если же объект оказался в области перекрытия гиперсфер, которая при обучении содержала объекты более чем одного образа, то переходим к гиперсферам второго уровня и проводим действия такие же, как для гиперсфер первого уровня. Этот процесс продолжается до тех пор, пока принадлежность неизвестного объекта тому или иному образу не определится однозначно. Правда, это событие может и не наступить. В частности, неизвестный объект может не попасть ни в одну из гиперсфер какого-либо уровня. В этих случаях «учитель» должен включить в решающие правила соответствующие действия. Например, система может либо отказаться от решения об однозначном отнесении объекта к какому-либо образу, либо использовать критерий минимума расстояния до эталонов данного или предшествующего уровня. Какой из этих приёмов эффективнее, сказать трудно, так как метод дробящихся эталонов носит в основном эмпирический характер.

Метод k-внутригрупповых средних

Опишем алгоритм:

1 шаг. Выбирается K исходных центров \bar{Z}_i будущих кластеров, например, первые K образов. $\bar{Z}_i = \bar{x}_i$, где $i=(1,2, \dots, K)$.

2 шаг. Вычисляется матрица расстояний между объектами и выбранными центрами классов по евклидовой метрике.

$$p(x_i, \bar{z}_j) = \left[\sum_{k=1}^p (x_{ik} - z_{jk})^2 \right]^{1/2},$$

где: \bar{x}_i - объект, характеризующийся набором признаков;

$$\bar{x}_i = (\bar{x}_{i1}, \bar{x}_{i2}, \dots, \bar{x}_{ip}) - (i=1, 2, \dots, n),$$

$$\bar{z}_j = (\bar{z}_{j1}, \bar{z}_{j2}, \dots, \bar{z}_{jp}) - (j=1, 2, \dots, K),$$

n - количество объектов, p - количество признаков.

3 шаг. Распределяются объекты \bar{x} по классам S_j на k-том шаге итерации по следующему правилу:

$x \in S_j$ \Leftrightarrow если $p(x, z_j) < p(x, z_i)$ для всех $i=(1, 2, \dots, K), i \neq j$, где S_j - множество объектов j -того класса с центром \bar{z}_j . В случае равенства расстояний элемент \bar{x} относится к классу, имеющему меньший порядковый номер.

4 шаг. Корректируем центры классов, то есть вычисляются новые центры классов по формуле:

$$\bar{z}_j + 1 = \frac{1}{u_j} \sum_{\bar{x} \in S_j} \bar{x}$$

где u_j - число объектов, входящих в класс S_j .

5 шаг. Проверяем сходимость алгоритма, то есть равенства

$$\bar{z}_j + 1 = \bar{z}_j \quad (j=1, 2, \dots, K).$$

Если это равенство выполняется, то процедура нахождения классов заканчивается. В противном случае осуществляется переход к следующей итерации, то есть алгоритм повторяется с шага 2.

Качество работы алгоритма зависит от:

- числа выбираемых центров класса;
- координат центров классов;
- геометрических особенностей расположения данных в евклидовом пространстве;
- выбранной меры близости (сходства) между объектами.

Раздел 5:

Системы непрямого управления

Лекция проводится в интерактивной форме с разбором конкретных ситуаций (3 час.)

5.1. Биологический нейрон

Центральная нервная система имеет клеточное строение. Единица – нервная клетка, нейрон. Нейрон имеет следующие основные свойства:

1. Участвует в обмене веществ и рассеивает энергию. Меняет внутреннее состояние с течением времени, реагирует на входные сигналы и формирует выходные воздействия и поэтому является активной динамической системой;
2. Имеет множество синапсов – контактов для передачи информации;
3. Нейрон взаимодействует путем обмена электрохимическими сигналами двух видов: электротоническими (с затуханием) и нервными импульсами (спайками), распространяющимися без затухания.

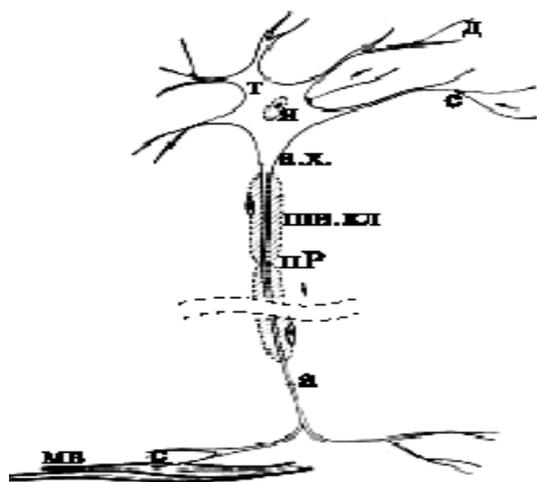


Рис. 5.1. Биологический нейрон

Биологический нейрон содержит следующие структурные единицы (рис. 5.1):

Тело клетки (т) – сома: содержит ядро (**я**), митохондрии (обеспечивают клетку энергией), другие органеллы, поддерживающие жизнедеятельность клетки;

Дендриты (д) – входные волокна, собирают информацию от других нейронов. Активность в дендритах меняется плавно. Длина их обычно не больше 1 мм;

Мембрана – поддерживает постоянный состав цитоплазмы внутри клетки, обеспечивает проведение нервных импульсов;

Цитоплазма – внутренняя среда клетки;

Аксон (а), один или ни одного у каждой клетки,

– длинное, иногда больше метра, выходное нервное волокно клетки. Импульс генерируется в

л оном холмике (**а.х.**). Аксон обеспечивает проведение импульса и передачу воздействия на другие нейроны или мышечные волокна (**мв**). Ближе к концу аксон часто ветвится;

Синапс (с) – место контакта нервных волокон – передает возбуждение от клетки к клетке. Передача через синапс почти всегда однонаправленная. Различают пресинаптические и постсинаптические клетки – по направлению передачи импульса;

Шванновские клетки (шв.кл.). Специфические клетки, почти целиком состоящие из миелина, органического изолирующего вещества. Плотнo «обматывают» нервное волокно 250 слоями миелина. Неизолированные места нервного волокна между шванновскими клетками называются **перехватами Ранвье (пР)**. За счет миелиновой изоляции скорость распространения нервных импульсов возрастает в 5-10 раз и уменьшаются затраты энергии на проведение импульсов. Миелинизированные волокна встречаются только у высших животных.

В центральной нервной системе человека насчитывается от 100 до 1000 типов нервных клеток, в зависимости выбранной степени детализации. Они отличаются картиной дендритов, наличием и длиной аксона и распределением синапсов около клетки.

Клетки сильно связаны между собой. У нейрона может быть больше 1000 синапсов. Близкие по функциям клетки образуют скопления, шаровидные или параллельные слоистые. В мозгу выделены сотни скоплений. Кора головного мозга – тоже скопление. Толщина коры – 2 мм, площадь – около квадратного фута.

5.2. Формальный нейрон

Биологический нейрон – сложная система, математическая модель которого до сих пор полностью не построена. Введено множество моделей, различающихся вычислительной сложностью и сходством с реальным нейроном. Одна из важнейших – **формальный нейрон (рис. 5.2)**. Несмотря на простоту ФН, сети, построенные из таких нейронов, могут сформировать произвольную многомерную функцию на выходе.

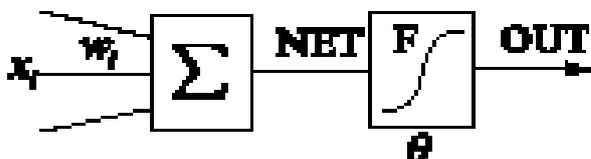


Рис. 5.2. Формальный нейрон

Нейрон состоит из взвешенного сумматора и нелинейного элемента.

Функционирование нейрона определяется формулами:

$$NET = \sum_i w_i x_i \quad (1) \quad OUT = F(NET - \Theta) \quad (2)$$

где x_i – входные сигналы, совокупность всех входных сигналов нейрона образует вектор **x**;

w_i – *весовые коэффициенты*, совокупность весовых коэффициентов образует вектор весов **w**;

NET – взвешенная сумма входных сигналов, значение NET передается на нелинейный элемент;

Θ – *пороговый уровень* данного нейрона;

F – нелинейная функция, называемая *функцией активации (функция преобразования)*.

Нейрон имеет несколько входных сигналов **x** и один выходной сигнал OUT. Параметрами нейрона, определяющими его работу, являются: вектор весов **w**, пороговый уровень и вид функции активации F.

Функционирование нейрона происходит следующим образом.

В текущий момент времени через входные синапсы на нейрон направляются сигналы от других нейронов и/или из внешнего мира. Каждый синапс имеет параметр, называемый *весом синапса*, и представляющий какое-либо число. Сигнал, проходящий через синапс, умножается на вес этого синапса. В зависимости от веса, сигнал может быть усилен (модуль веса

> 1) или ослаблен (модуль веса < 1) по амплитуде. Сигналы от всех синапсов, ведущих к данному нейрону, принимает сумматор.

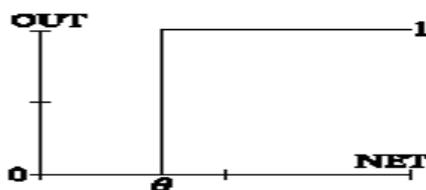
Сумматор производит суммирование всех пришедших сигналов и подает на собственно нейрон (преобразователь) одно число - полученную сумму. Величина этого числа будет зависеть как от величин исходных сигналов, так и от весов синапсов. Нейрон, получивший это число, преобразует его согласно своей функции, в результате которой получается другое число, и отправляет его по «аксону» всем остальным нейронам через соответствующие синапсы. Последующие нейроны производят с полученными сигналами такие же операции, лишь с тем различием, что во-первых, веса их синапсов могут быть уже другими, во-вторых, другие нейроны могут иметь другой вид функции преобразования.

5.3. Виды функций активации

Основные виды функций активации, получившие распространение в искусственных НС.

1. Жесткая ступенька:

$$OUT = \begin{cases} 0, & NET < \theta \\ 1, & NET \geq \theta \end{cases}$$

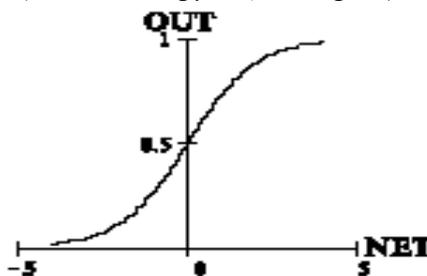


Используется в классическом формальном нейроне. Развита полная теория, позволяющая синтезировать произвольные логические схемы на основе ФН с такой нелинейностью. Функция вычисляется двумя-тремя машинными инструкциями, поэтому нейроны с такой нелинейностью требуют малых вычислительных затрат.

Эта функция чрезмерно упрощена и не позволяет моделировать схемы с непрерывными сигналами. Отсутствие первой производной затрудняет применение градиентных методов для обучения таких нейронов.

2. Логистическая функция (сигмоида, функция Ферми):

$$OUT = \frac{1}{1 + e^{-NET}}$$



Применяется очень часто для многослойных перцептронов и других сетей с непрерывными сигналами. Гладкость, непрерывность функции – важные положительные качества. Непрерывность первой производной позволяет обучать сеть градиентными методами (например, метод обратного распространения ошибки).

Функция симметрична относительно точки (NET=0, OUT=1/2), это делает равноправными значения OUT=0 и OUT=1, что существенно в работе сети. Тем не менее, диапазон выходных значений от 0 до 1 несимметричен, из-за этого обучение значительно замедляется.

Данная функция – сжимающая, т.е. для малых значений NET коэффициент передачи $K=OUT/NET$ велик, для больших значений он снижается. Поэтому диапазон сигналов, с которыми нейрон работает без насыщения, оказывается широким.

Значение производной легко выражается через саму функцию. Быстрый расчет производной ускоряет обучение.

Выбор функции активации определяется:

1. Спецификой задачи;
2. Удобством реализации на ЭВМ, в виде электрической схемы или другим способом;
3. Алгоритмом обучения: некоторые алгоритмы накладывают ограничения на вид функции активации, их нужно учитывать.

Чаще всего вид нелинейности не оказывает принципиального влияния на решение задачи. Однако удачный выбор может сократить время обучения в несколько раз.

Итак, модель формального нейрона не является биоподобной и скорее похожа на математическую абстракцию, чем на живой нейрон. Тем удивительнее оказывается многообразие задач, решаемых с помощью таких нейронов и универсальность получаемых алгоритмов.

5.4. Искусственные нейронные сети

Искусственные нейронные сети (НС) – совокупность моделей биологических нейронных сетей. Представляют собой сеть элементов – искусственных нейронов – связанных между собой синаптическими соединениями. Сеть обрабатывает входную информацию и в процессе изменения своего состояния во времени формирует совокупность выходных сигналов.

Работа сети состоит в преобразовании входных сигналов во времени, в результате чего меняется внутреннее состояние сети и формируются выходные воздействия. Обычно НС оперирует цифровыми, а не символьными величинами.

Большинство моделей НС требуют обучения. В общем случае, *обучение* – такой выбор параметров сети, при котором сеть лучше всего справляется с поставленной проблемой. Обучение – это задача многомерной оптимизации, и для ее решения существует множество алгоритмов.

Искусственные нейронные сети – набор математических и алгоритмических методов для решения широкого круга задач. Выделим характерные черты искусственных нейросетей как универсального инструмента для решения задач:

1. НС дают возможность лучше понять организацию нервной системы человека и животных на средних уровнях: память, обработка сенсорной информации, моторика;

2. НС – средство обработки информации:

а) гибкая модель для нелинейной аппроксимации многомерных функций;

б) средство прогнозирования во времени для процессов, зависящих от многих переменных;

в) классификатор по многим признакам, дающий разбиение входного пространства на области;

г) средство распознавания образов;

д) инструмент для поиска по ассоциациям;

е) модель для поиска закономерностей в массивах данных.

3. НС свободны от ограничений обычных компьютеров благодаря параллельной обработке и сильной связанности нейронов.

4. В перспективе НС должны помочь понять принципы, на которых построены высшие функции нервной системы: сознание, эмоции, мышление.

Современные искусственные НС по сложности и «интеллекту» приближаются к нервной системе таракана, но уже сейчас демонстрируют ценные свойства:

1. Обучаемость. Выбрав одну из моделей НС, создав сеть и выполнив алгоритм обучения, мы можем обучить сеть решению задачи, которая ей по силам. Нет никаких гарантий, что это удастся сделать при выбранных сети, алгоритме и задаче, но если все сделано правильно, то обучение бывает успешным;

2. Способность к обобщению. После обучения сеть становится нечувствительной к малым изменениям входных сигналов (шуму или вариациям входных образов) и дает правильный результат на выходе;

3. Способность к абстрагированию. Если предъявить сети несколько искаженных вариантов входного образа, то сеть сама может создать на выходе идеальный образ, с которым она никогда не встречалась.

5.5. Архитектура нейронной сети

Основой работы самообучающихся нейропрограмм является нейронная сеть, представляющая собой совокупность нейронов - простых элементов, связанных между собой определенным образом. Нейроны и межнейронные связи задаются программно на обычном компьютере или могут иметь «материальную» основу - особую микросхему (нейрочип), которые применяются в специально созданных нейрокомпьютерах. Структура взаимосвязей между нейронами в нейрокомпьютере или нейропрограмме аналогична таковой в биологических объектах. Искусственный нейрон имеет коммуникации с другими нейронами через синапсы, передающие сигналы от других нейронов к данному (дендриты) или от данного нейрона к другим (аксон). Кроме того, нейрон может быть связан сам с собой. Несколько нейронов, связанных между собой определенным образом, образуют нейронную сеть.

Нейросеть, также как и биологический аналог, должна иметь каналы для связи с внешним миром. Одни каналы обеспечивают поступление информации из внешнего мира на нейросеть, другие выводят информацию из нейросети во внешний мир. Поэтому одни нейроны сети рассматриваются как входные, другие же - как выходные. Часть нейронов может не сообщаться с внешним миром, а взаимодействовать с входными, выходными и такими же нейронами («скрытые» нейроны).

Очевидно, что существует огромное количество способов соединения нейронов, растущее с увеличением числа нейронов в сети. Наиболее употребительной является слоистая архитектура (рис. 5.3), в которой нейроны располагаются «слоями». В наиболее общем случае аксоны каждого нейрона одного слоя направлены к нейронам следующего слоя. Таким образом, нейроны первого слоя являются входными (принимающими информацию из внешнего мира), нейроны последнего слоя - выходными (выдающими информацию во внешний мир).

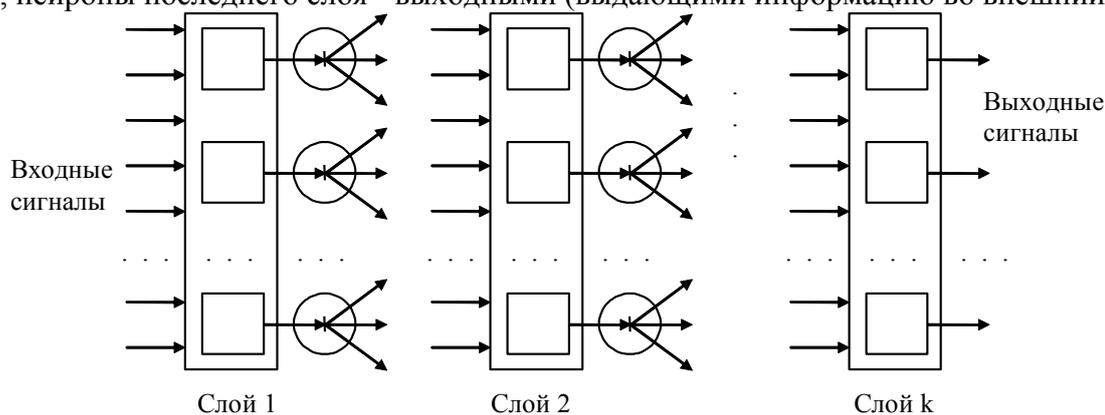


Рис. 5.3. Слоистая сеть

Число нейронов в каждом слое может быть любым и никак заранее не связано с количеством нейронов в других слоях. Стандартный способ подачи входных сигналов: все нейроны первого слоя получают каждый входной сигнал. Особое распространение получили трехслойные сети, в которых каждый слой имеет свое наименование: первый – входной, второй – скрытый, третий – выходной.

Другой вид архитектуры – полносвязная, когда каждый нейрон соединен с каждым, в том числе сам с собой. Пример простейшей нейросети из 3 нейронов показан на рис. 5.4. Для удобства изображения из каждого нейрона выходит не один, а несколько аксонов, направленных на другие нейроны или во внешний мир, что аналогично присоединенным к одному аксону через синапсы нескольким дендритам.

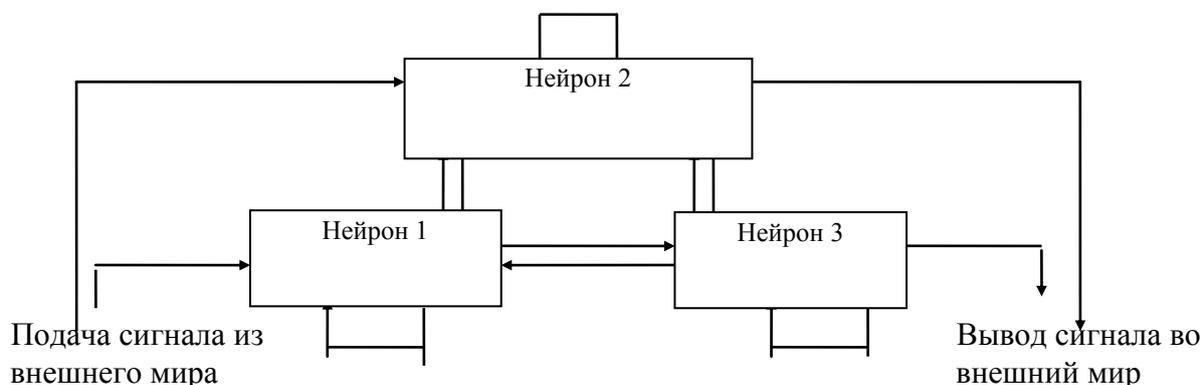


Рис. 5.4. Схема простейшей нейронной сети из 3 нейронов. Сеть имеет 13 синапсов, 4 из которых служат для связи с внешним миром, а остальные соединяют нейроны между собой

Слоистая или полносвязная архитектуры не налагают существенных ограничений на участвующие в них элементы. Единственное жесткое требование, предъявляемое архитектурой к элементам сети, это соответствие размерности вектора входных сигналов элемента (она определяется архитектурой) числу его входов.

Если полносвязная сеть функционирует до получения ответа заданное число тактов k , то ее можно представить как частный случай k -слойной сети, все слои которой одинаковы и каждый из них соответствует такту функционирования полносвязной сети.

Существенное различие между полносвязной и слоистой сетями возникает тогда, когда число тактов функционирования заранее не ограничено – слоистая сеть так работать не может.

5.6. Функционирование нейросети

В случае эмуляции нейросети на обычном компьютере все математические операции осуществляет программа. Нейронная сеть при этом представляет собой массив синаптических весов. Этот массив может находиться либо на диске компьютера в виде файла определенного формата (при хранении нейросети) либо в оперативной памяти компьютера (когда нейросеть функционирует).

При создании новой нейросети в памяти компьютера отводится место под массив синаптических весов, называемый картой. Этот массив заполняется совершенно случайными числами из определенного диапазона. Поэтому каждая созданная сеть даже при одних и тех же параметрах (число нейронов, их характеристика) является уникальной. Уникальность сетей проявляется в том, что сети с одинаковыми параметрами, обучающиеся на одинаковых задачах, ведут себя неодинаково. Это касается времени обучения, качества обучения, уверенности в выдаваемых ответах при тестировании. В этом еще одно сходство сетей с биообъектами.

Рассмотрим работу сети безотносительно к процессу обучения или выдачи решения. После инициализации сети веса всех синапсов случайны. В начальный момент времени на входные нейроны через входные синапсы (один или несколько) подается из внешнего мира вектор входных сигналов, представляющий набор чисел (или одно число). Далее этот сигнал начинает распространяться по всем связям между нейронами, изменяясь при прохождении через каждый нейрон согласно схеме функционирования нейрона. В конечном итоге, после одного прохода выходные нейроны выдадут во внешний мир какие-либо сигналы. Вся процедура однократного прохождения сигналов по нейронной сети является тактом функционирования сети. Можно не считать сигналы с выходных нейронов после одного такта функционирования, а продолжить обмен сигналами еще несколько раз, не подавая сигналов на вход. Обычно количество тактов функционирования между подачей сигналов на вход и снятием сигналов с выхода фиксировано для данной сети. Как правило, этот параметр задается

при инициализации сети и хранится в файле сети вместе с матрицей коэффициентов (как и некоторые другие параметры - число нейронов, характеристика нейронов и т.д.).

Таким образом, нейронная сеть, получающая на входе некоторый сигнал, способна после прохода его по нейронам выдавать на выходе определенный ответ, который зависит от весовых коэффициентов всех нейронов и от самого сигнала. Очевидно, что при осуществлении таких процедур на только что инициализированной сети мы будем получать на выходе сигналы, лишенные всякого смысла (весовые коэффициенты случайны). Чтобы добиться выдачи сетью требуемого результата, необходимо обучить ее.

5.7. Общая схема обучения нейронной сети

Алгоритмы обучения бывают с учителем и без. Алгоритм называется *алгоритмом с учителем*, если при обучении известны и входные, и выходные вектора сети. Имеются пары вход + выход – известные условия задачи и решение. В процессе обучения сеть меняет свои параметры и учится давать нужное отображение $X \rightarrow Y$. Сеть учится давать результаты, которые нам уже известны. За счет способности к обобщению сетью могут быть получены новые результаты, если подать на вход вектор, который не встречался при обучении.

Обучение с учителем

Для обучения нейронной сети с учителем необходима обучающая выборка (задачник), состоящая из примеров. Каждый пример представляет собой задачу одного и того же типа с индивидуальным набором условий (входных параметров) и заранее известным ответом. Например, в качестве входных параметров в одном примере могут использоваться данные обследования одного больного, тогда заранее известным ответом в этом примере может быть диагноз. Несколько примеров с разными ответами образуют задачник. Задачник располагается в базе данных, каждая запись которой является примером.

Общая схема обучения нейросети с учителем:

1. Из обучающей выборки берется текущий пример (изначально, первый) и его входные параметры (представляющие в совокупности вектор входных сигналов) подаются на входные синапсы обучаемой нейросети. Обычно каждый входной параметр примера подается на один соответствующий входной синапс;

2. Нейросеть производит заданное количество тактов функционирования, при этом вектор входных сигналов распространяется по связям между нейронами (прямое функционирование);

3. Измеряются сигналы, выданные теми нейронами, которые считаются выходными;

4. Производится интерпретация выданных сигналов, и вычисляется оценка, характеризующая различие между выданным сетью ответом и требуемым ответом, имеющимся в примере. Оценка вычисляется с помощью соответствующей функции оценки. Чем меньше оценка, тем лучше распознан пример, тем ближе выданный сетью ответ к требуемому. Оценка, равная нулю, означает что требуемое соответствие вычисленного и известного ответов достигнуто. Заметим, что только что инициализированная (необученная) нейросеть может выдать правильный ответ только совершенно случайно.

5. Если оценка примера равна нулю, ничего не предпринимается. В противном случае на основании оценки вычисляются поправочные коэффициенты для каждого синаптического веса матрицы связей, после чего производится подстройка синаптических весов (обратное функционирование). В коррекции весов синапсов и заключается обучение;

6. Осуществляется переход к следующему примеру задачника и вышеперечисленные операции повторяются. Проход по всем примерам обучающей выборки с первого по последний считается одним циклом обучения;

При прохождении цикла каждый пример имеет свою оценку. Вычисляется, кроме того, суммарная оценка множества всех примеров обучающей выборки. Если после прохождения нескольких циклов она равна нулю, обучение считается законченным, в противном случае циклы повторяются.

Количество циклов обучения, также как и время, требуемое для полного обучения, зависят от многих факторов - величины обучающей выборки, количества входных парамет-

ров, вида задачи, типа и параметров нейросети и даже от случайного расклада весов синапсов при инициализации сети.

Обучение без учителя

Алгоритм относится к обучению *без учителя*, если известны только входные вектора, и на их основе сеть учится давать *наилучшие* значения выходов. Что понимается под «наилучшими» – определяется алгоритмом обучения.

Для алгоритмов обучения с учителем нужны известные входные и выходные вектора, а эта информация не всегда доступна.

Вопрос о биологической правдоподобности обучения с учителем также открыт. Нейроны зрительной коры, например, учатся реагировать на световые импульсы лишь под действием самих импульсов, без внешнего учителя. Однако высшие этапы обучения, например, у детей, невозможны без «учителя» в лице его родителя. Кроме того, отдельные области в мозге вполне могут выполнять роль «учителей» для других, управляя их активностью. Поэтому нельзя однозначно сказать, какой тип обучения биологически правдоподобнее, с учителем или без.

При обучении без учителя сеть имеет для обучения лишь известные входные вектора. Правильные значения выходных векторов неизвестны. Какие вектора будут сформированы на выходе, зависит от алгоритма обучения. Обучающее множество состоит из S известных входных векторов x^s $s=1... S$, в процессе обучения сеть учится формировать выходные вектора y^s для каждого вектора из обучающего множества x^s .

Успех обучения во многом зависит от числа нейронов в сети, или, точнее, от числа синапсов. Именно весовые коэффициенты синапсов хранят «опыт» сети. Теоретически, бесконечно увеличивая число нейронов и синапсов, всегда можно добиться полного обучения сети на данном задатнике, однако это ли является целью создателя экспертной системы? Очевидно, нет. Главное, чтобы обученная сеть хорошо распознавала примеры, как раз не входящие в задатник.

Проблема заключается в том, что сеть с заведомо большим (избыточным) числом синапсов (относительно данного задатника) может хорошо обучиться, просто «механически запомнив» имеющиеся примеры. Такая сеть обучится быстро (нет необходимости как можно более точной подстройки весов) за счет количества, а не качества.

Хорошим практическим выходом из данной затруднительной ситуации были бы сети, способные автоматически наращивать число нейронов при невозможности дальнейшего обучения, не теряя при этом уже имеющегося опыта.

5.8. Классификация нейронных сетей

Однослойный перцептрон

Круг проблем, которые под силу однослойному перцептрону, очень ограничен. Рассмотрим однослойную сеть из одного нейрона (рис. 5.5).

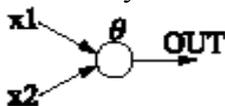


Рис. 5.5. Сеть из одного нейрона.

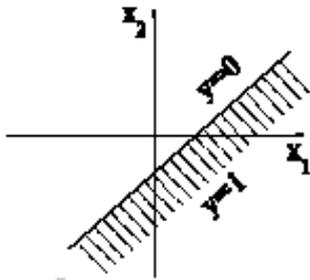


Рис. 5.6. Возможный выход, если F имеет вид жесткой ступеньки

Выход сети: $y = F(w_1x_1 + w_2x_2 - \Theta)$. Если F имеет вид жесткой ступеньки с двумя возможными значениями 0 и 1, то выход сети будет иметь вид (рис. 5.6). Гиперплоскость (в случае многих входов), разделяющая различные значения выхода, называется *решающей поверхностью*. Для жесткой ступеньки решающая поверхность задается уравнением: $w_1x_1 + w_2x_2 = \Theta$.

Для двухвходового нейрона она имеет вид прямой, произвольно повернутой и смещенной из начала координат. Угол поворота определяется коэффициентами w_1 , w_2 , а смещение из начала координат – порогом.

Если выбрана гладкая функция активации, то выход сети будет плавно меняться от нуля до единицы в направлении, перпендикулярном прямой. Зависимость выходного сигнала от входов удобно представлять полутоновой картой: черному цвету соответствует значение выхода, равное 0, белому – равное 1. По осям отложены значения входов. Реальная зависимость для функции активации в виде гиперболического тангенса – на рис. 5.7.

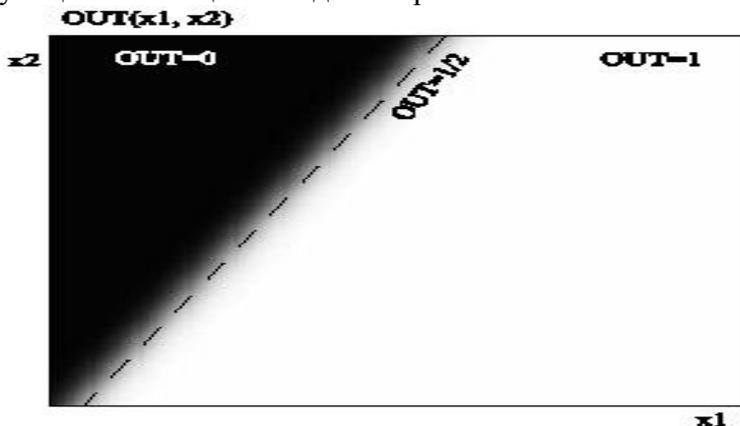


Рис. 5.7. Выход однослойного перцептрона в виде полутоновой карты.

Многослойный перцептрон

Формальные нейроны могут объединяться в сети различным образом. Самым распространенным видом сети стал *многослойный перцептрон* (рис. 5.8).

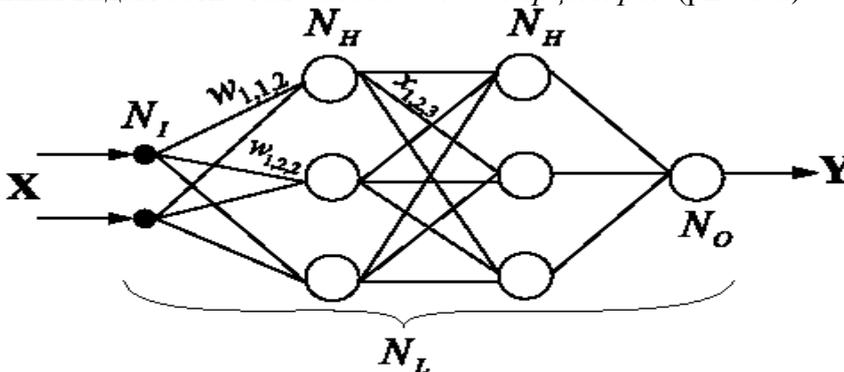


Рис. 5.8 . Многослойный перцептрон

Сеть состоит из произвольного количества слоев нейронов. Нейроны каждого слоя соединяются с нейронами предыдущего и последующего слоев по принципу «каждый с каждым». Первый слой (слева) называется *сенсорным* или *входным*, внутренние слои называются *скрытыми* или *ассоциативными*, последний (самый правый, на рисунке состоит из одного нейрона) – *выходным* или *результативным*. Количество нейронов в слоях может быть произвольным. Обычно во всех скрытых слоях одинаковое количество нейронов.

Обозначим количество слоев и нейронов в слое. Входной слой: N_I нейронов; N_H нейронов в каждом скрытом слое; N_O выходных нейронов. x – вектор входных сигналов сети, y – вектор выходных сигналов.

Существует путаница с подсчетом количества слоев в сети. Входной слой не выполняет никаких вычислений, а лишь распределяет входные сигналы, поэтому иногда его считают, иногда – нет. Обозначим через N_L полное количество слоев в сети, считая входной.

Работа многослойного перцептрона (МСП) описывается формулами:

$$NET_{jl} = \sum_i w_{ijl} x_{ijl} \quad OUT_{jl} = F(NET_{jl} - \Theta_{jl}) \quad x_{i,j(l+1)} = OUT_{jl}$$

где индексом i всегда будем обозначать номер входа, j – номер нейрона в слое, l – номер слоя.

x_{ijl} – i -й входной сигнал j -го нейрона в слое l ;

w_{ijl} – весовой коэффициент i -го входа нейрона номер j в слое l ;

NET_{jl} – сигнал NET j -го нейрона в слое l ; OUT_{jl} – выходной сигнал нейрона;

Θ_{jl} – пороговый уровень нейрона j в слое l ;

Введем обозначения: w_{jl} – вектор-столбец весов для всех входов нейрона j в слое l ; W_l – матрица весов всех нейронов в слое l . В столбцах матрицы расположены вектора w_{jl} . Аналогично x_{jl} – входной вектор-столбец слоя l .

Каждый слой рассчитывает нелинейное преобразование от линейной комбинации сигналов предыдущего слоя. Отсюда видно, что линейная функция активации может применяться только для тех моделей сетей, где не требуется последовательное соединение слоев нейронов друг за другом. Для многослойных сетей функция активации должна быть нелинейной, иначе можно построить эквивалентную однослойную сеть, и многослойность оказывается ненужной. Если применена линейная функция активации, то каждый слой будет давать на выходе линейную комбинацию входов. Следующий слой даст линейную комбинацию выходов предыдущего, а это эквивалентно одной линейной комбинации с другими коэффициентами, и может быть реализовано в виде одного слоя нейронов.

Многослойная сеть может формировать на выходе произвольную многомерную функцию при соответствующем выборе количества слоев, диапазона изменения сигналов и параметров нейронов.

Как и ряды, многослойные сети оказываются универсальным инструментом аппроксимации функций. Видно отличие работы нейронной сети от разложения функции в ряд:

Ряд:
$$f(x) = \sum_i c_i f_i(x)$$

Нейронная сеть:
$$f(x) = F \left(\underbrace{\sum_{i_N} w_{i_N j_N N} \dots \sum_{i_2} w_{i_2 j_2 2} F \left(\underbrace{\sum_{i_1} w_{i_1 j_1 1} x_{i_1 j_1 1} - \theta_{j_1}}_{\text{слой 1}} \right)}_{\text{слой 2}} \right)_{\text{слой N}}$$

За счет поочередного расчета линейных комбинаций и нелинейных преобразований достигается аппроксимация произвольной многомерной функции при соответствующем выборе параметров сети.

В многослойном перцептроне нет обратных связей. Такие модели называются *сетями прямого распространения*. Они не обладают внутренним состоянием и не позволяют без дополнительных приемов моделировать развитие динамических систем.

Алгоритм решения задач с помощью МСП

Чтобы построить МСП, необходимо выбрать его параметры. Чаще всего выбор значений весов и порогов требует *обучения*, т.е. пошаговых изменений весовых коэффициентов и пороговых уровней.

Общий алгоритм решения:

1. Определить, какой смысл вкладывается в компоненты входного вектора x . Входной вектор должен содержать формализованное условие задачи, т.е. всю информацию, необходимую для получения ответа.

2. Выбрать выходной вектор y таким образом, чтобы его компоненты содержали полный ответ поставленной задачи.

3. Выбрать вид нелинейности в нейронах (функцию активации). При этом желательно учесть специфику задачи, т.к. удачный выбор сократит время обучения.

4. Выбрать число слоев и нейронов в слое.

5. Задать диапазон изменения входов, выходов, весов и пороговых уровней, учитывая множество значений выбранной функции активации.

6. Присвоить начальные значения весовым коэффициентам и пороговым уровням и дополнительным параметрам (например, крутизне функции активации, если она будет настраиваться при обучении). Начальные значения не должны быть большими, чтобы нейроны не оказались в насыщении (на горизонтальном участке функции активации), иначе обучение будет очень медленным. Начальные значения не должны быть и слишком малыми, чтобы выходы большей части нейронов не были равны нулю, иначе обучение также замедлится.

7. Провести обучение, т.е. подобрать параметры сети так, чтобы задача решалась наилучшим образом. По окончании обучения сеть готова решить задачи того типа, которым она обучена.

8. Подать на вход сети условия задачи в виде вектора x . Рассчитать выходной вектор y , который и даст формализованное решение задачи.

Примеры задач, решаемых с помощью МСП:

1. Задача классификации;
2. Распознавание букв алфавита;
3. Аппроксимация многомерной функции;
4. Синтез речи: NET-talk;

Сеть Кохонена

Задача классификации заключается в разбиении объектов на классы, когда основой разбиения служит вектор параметров объекта. Объекты в пределах одного класса считаются эквивалентными с точки зрения критерия разбиения. Сами классы часто бывают неизвестны заранее, а формируются динамически. Классы зависят от предъявляемых объектов, и поэтому добавление нового объекта требует корректирования системы классов.

Если для классификации применять нейронные сети, необходимо формализовать задачу. Самый очевидный способ: выберем в качестве входных данных вектор параметров единственного объекта. Результатом работы сети будет код класса, к которому принадлежит предъявленный на входе объект. В нейросетях принято кодирование номером канала. Поэтому сеть будет иметь M выходов, по числу классов, и чем большее значение принимает выход номер m_0 , тем больше «уверенность» сети в том, что входной объект принадлежит к классу m_0 . Полезно применить функцию активации SOFTMAX, тогда сумма выходов всегда будет равна единице. Каждый выход можно будет трактовать как вероятность того, что объект принадлежит данному классу. Все выходы образуют полную группу, т.к. сумма выходов равна единице, и объект заведомо относится к одному из классов.

Выберем евклидову меру близости. В этом случае ядро класса, минимизирующее сумму мер близости для объектов этого класса, совпадает с центром тяжести объектов:

$$c^{m_0} = \frac{1}{N(m_0)} \sum_{p:m(p)=m_0} x^p \quad \text{где } N(m_0) \text{ – число объектов } x^p \text{ в классе } m_0.$$

При разбиении на классы должна быть минимизирована суммарная мера близости для всего множества x^p входных объектов.

Сеть нейронов, использующая евклидову меру близости для классификации объектов, называется *сетью Кохонена* (рис. 5.9).

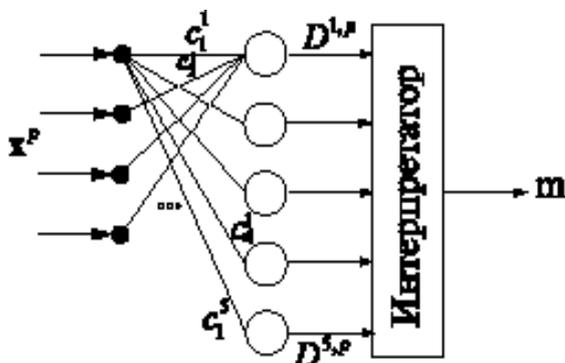


Рис. 5.9. Сеть Кохонена

Нейроны слоя Кохонена генерируют сигналы $D^{m,p}$. Интерпретатор выбирает максимальный сигнал слоя Кохонена и выдает номер класса m , соответствующий номеру входа, по которому интерпретатором получен максимальный сигнал. Это соответствует номеру класса объекта, который был предъявлен на входе, в виде вектора x^p .

Ядра c_m являются весовыми коэффициентами нейронов. Каждый нейрон Кохонена запоминает одно ядро класса, и отвечает за определение объектов в своем классе, т.е. величина выхода нейрона тем больше, чем ближе объект к данному ядру класса. Общее количество классов совпадает с количеством нейронов Кохонена. Меняя количество нейронов, можно динамически менять количество классов.

Сеть Хопфилда

Для сетей прямого распространения были приняты ограничения: все сигналы в сети распространяются только от входа к выходу, но не наоборот. Сеть также предполагалась послойно – полносвязной. Оба эти ограничения несправедливы для биологических НС и сужают возможности модели. Сеть прямого распространения не имеет внутреннего состояния: значения выходов нейронов зависят *только* от входного вектора и не меняются во времени, если вход неизменен. Моделирование динамических процессов на таких сетях возможно только искусственными приемами, например, когда сеть на каждом шаге прогнозируется малое изменение состояния для исследуемого динамического объекта.

Чтобы расширить диапазон решаемых задач, были предложены сети с обратными связями. Полное математическое описание пока создано только для простейших случаев сетей с обратными связями. Дж. Хопфилд внес вклад в разработку и теории, и моделей таких сетей.

Рассмотрим однослойную сеть с обратными связями, где значение выхода каждого нейрона подается обратно на входы всех нейронов того же слоя (рис. 5.10):

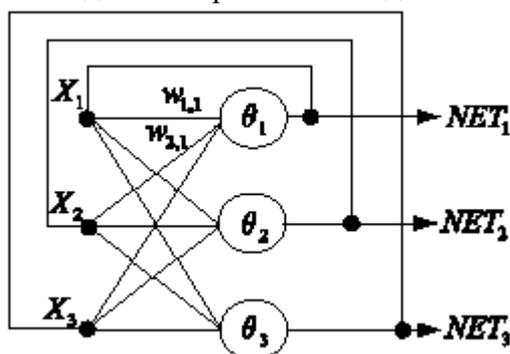


Рис. 5.10. Сеть с обратными связями

Входной вектор для сети в данном случае совпадает с выходным NET, взятым на предыдущей итерации. Предполагаем, что расчет сигналов NET происходит мгновенно и одновременно во всех нейронах, а распространение по обратным связям дает задержку в одну итерацию и происходит одновременно во всех нейронах: сеть работает синхронно. На каждой итерации по вектору X получается новый вектор NET, который отличается от X небольшим. Длина шага будет пропорциональна h – небольшому положительному числу, выбранного нами при создании сети.

Сформированный сигнал NET подается обратно на вход сети, и на следующей итерации становится новым значением X . Так, на каждой итерации происходит небольшое изменение вектора X в направлении антиградиента P , т.е. в сторону минимума многочлена $P(x)$.

Сети с обратными связями могут работать в качестве *ассоциативной памяти*. Это означает, что по вектору, поданному на вход, сетью будет создан на выходе один из запомненных ранее векторов, наиболее «похожий» (в некотором выбранном смысле) на данный входной вектор. Такой способ выборки данных называется *адресацией по содержанию*, в отличие от адресации по номеру ячейки памяти, принятому в ЭВМ фон-неймановского типа. Этот способ адресации широко используется в биологических НС. Например, один лишь запах жасмина может вызвать в памяти целый набор ассоциаций, причудливо связанных друг с другом и включающих в себя визуальные, звуковые и кинестетические образы. Память с такой адресацией является весьма перспективной для создания систем искусственного интеллекта, систем распознавания речевых сигналов и изображений.

Решение задач с помощью сетей Хопфилда

Решение некоторой задачи с помощью сетей Хопфилда распадается на этапы:

1. Построить функцию энергии таким образом, чтобы точка глобального минимума этой функции совпала с решением задачи. При этом градиент функции энергии должен допускать вычисление с помощью НС;
2. Записать формулы для расчета параметров сети (весовых коэффициентов и пороговых уровней) для расчета градиента функции энергии;
3. Разорвать цепочку обратной связи и предъявить сети входной вектор. Рассчитать значения выходов;
4. Замкнуть обратную связь и предоставить сети возможность самостоятельно менять свое состояние (релаксация). Остановить процесс релаксации после того, как выходной вектор перестанет меняться, т.е. по достижении минимума функции энергии. Полученные выходы сети дают решение задачи.

Раздел 6:

Представление знаний в интеллектуальных системах

Для представления нечетких понятий и оперирования с ними американский ученый Л.Заде в 60-х годах разработал теорию нечетких множеств, а затем – нечеткую логику, базирующуюся на ней. В основе теории нечетких множеств лежит интерпретация факта принадлежности элемента a множеству A как факта, который может быть истинным или ложным с некоторой оценкой истинности $\mu_A(a)$, пробегающей значения от 0 до 1. Эта оценка истинности называется функцией принадлежности элемента a множеству A . Операции включения и равенства в теории нечетких множеств определяются обычно следующим образом (по Л.Заде):

$$F \subseteq G :$$

$$\forall a, \mu_F(a) \leq \mu_G(a)$$

$$F = G :$$

$$\forall a, \mu_F(a) = \mu_G(a).$$

Дополнение множества F к G определяется так, что

$$\forall a, \mu_{\bar{F}}(a) = 1 - \mu_G(a).$$

Пересечение и объединение множеств определяются следующим образом:

$$\forall a, \mu_{F \cap G}(a) = \min(\mu_F(a), \mu_G(a)),$$

$$\forall a, \mu_{F \cup G}(a) = \max(\mu_F(a), \mu_G(a)).$$

Эти определения не единственные, хотя они не противоречат интуитивным представлениям о соответствующих операциях над нечеткими множествами.

Частным случаем теории нечетких множеств (при $\mu = 1$ или 0) является классическая теория множеств. Однако встречаются и другие определения операций над нечеткими множествами.

Так же как на основе классической теории множеств строится двоичная (булева) логика, так и на базе теории нечетких множеств строится теория нечетких множеств. Она оперирует с высказываниями, для которых функция принадлежности, описанная ранее, определена на множестве истинных высказываний. Функция принадлежности интерпретируется как мера истинности, уверенности или достоверности и отражает нечеткость знаний.

Предположим, существуют следующие высказывания:

"Иванов – хороший человек" с $\mu = 0.8$,

"Политик – хороший человек" с $\mu = 0.3$.

Тогда конъюнкция этих двух высказываний (имеющая смысл как уточнение мнения об Иванове, когда стало известно, что он – политик) определяется функцией принадлежности $\mu = 0.3$, а дизъюнкция – $\mu = 0.8$.

В теории нечетких множеств функция принадлежности может интерпретироваться как субъективное представление об истинности высказываний или объективная нечеткость знаний (информации). В первом случае описание нечетких высказываний является как бы снимком состояния некоторой интеллектуальной системы, обученной на примерах взаимодействия с внешней средой или заполненной субъективными знаниями экспертов. Во втором случае нечеткость является следствием каких-либо помех при поступлении информации в систему и интерпретации ее в виде знаний. В обоих случаях функцию принадлежности можно интерпретировать как вероятностную меру истинности и применять теорию вероятности к ее обработке и анализу. Это справедливо, т.к. интеллектуальная система работает с множеством разных субъектов, имеющих разные субъективные представления об истинности высказываний, или с множеством разных ситуаций, в которых разные помехи создают вероятностное описание истинности информации (знаний).

Недостатки классической логики и основанной на ней логики предикатов первого порядка как метода представления знаний об окружающем мире привели к появлению псевдофизических логик. В их основе лежит представление нечетких или размытых понятий в виде так называемых лингвистических переменных, придуманных Заде для того, чтобы приблизить семантику (смысл) денотата (знака) к семантике, которая вырабатывается в мозгу человека в процессе его обучения (опыта). Для этого множество образов (десигнатов), с которыми должна оперировать интеллектуальная система, представляется в виде точек на шкалах.

Лингвистической переменной называют переменную, значениями которой являются слова или предложения естественного или искусственного языка.

Например, можно рассматривать лингвистические переменные «возраст» (в годах), «расстояние до объекта» (в м или км) и т.п. Значение линг. переменной «возраст» могут быть значения: «юный», «молодой», «зрелый», «пожилой», «старый», «дряхлый». С переменной «расстояние» – «вплотную», «очень близко», «близко», «рядом», «недалеко», «далеко», «очень далеко», «у черта на куличиках». Взаимосвязь между этими двумя представлениями (множеством точек на шкале и множеством знаковых значений) задается с помощью функции принадлежности $\mu_x(t)$, где x – значение лингвистической переменной, t – значение на шкале.

Значение функции принадлежности интерпретируется как вероятность того, что значение t на шкале можно заменить знаком x или наоборот. Очевидно, что можно пронормировать значения функции принадлежности в соответствии с формулой:

$$\sum_x \mu_x(t) = 1$$

или в соответствии с

$$\sum_t \mu_x(t) = 1$$

На рис. 6.1 приведен пример описания лингвистической переменной «возраст». Здесь каждая кривая описывает ее одно символическое значение. Наиболее используемыми псевдофизическими логиками являются пространственная, временная и каузальная (причинно-следственная).

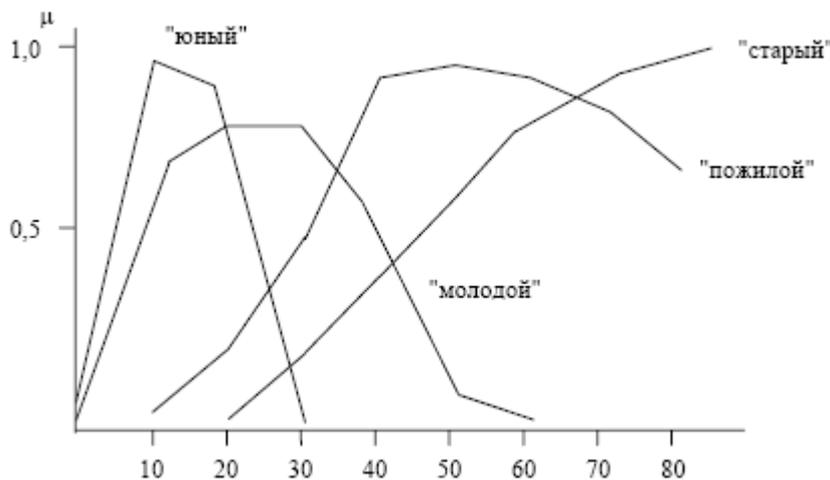


Рис. 6.1. Описание лингвистической переменной «Возраст»

Значения, которые может принимать лингвистическая переменная, называется термами. Термы лингвистических переменных являются нечеткими переменными. Терм, состоящий из одного слова, называется атомарным термом. Производные от атомарных термов, получаемые путем уточнения «очень», «не очень», «более или менее», называются подтермами.

Нечеткая логика работает даже тогда, когда хороших решений нет, тем не менее решение, пусть даже плохое, принять надо. Поскольку значение лингвистической переменной «размазано» по числовой прямой, даже при наложении друг на друга взаимоисключающих ограничений решение получается за счет «хвостов» функции совместимости.

Функции принадлежности

Для построения функций принадлежности можно использовать типовые функции, обладающие следующими свойствами:

- функция должна быть непрерывной, и, по возможности, гладкой на всей области определения;
- функция должна иметь несколько параметров, изменяя которые можно управлять формой функции;
- функция должна равняться 1 при значении носителя нечеткого множества с максимальной принадлежностью к нечеткому множеству или стремиться к 1 при стремлении параметра носителя к значению наибольшей принадлежности (например, функция принадлежности множества «старых людей» должна стремиться к 1 при значении возраста к ∞).

Для настройки параметров функции принадлежности можно задавать ряд точек носителя и значений принадлежности в этих точках. Обычно такими точками могут быть точки на уровнях значения функции принадлежности 0; 1; 0.5; 0.1; 0.9.

Первый вид функция принадлежности:

$$\mu(x) = \begin{cases} a & \text{при } x < x_0 \\ \left[1 + \left(\frac{x - x_0}{a} \right)^{-b} \right]^{-1} & \text{при } x > x_0 \end{cases}$$

Функция положительна при $x \geq x_0$, имеет точку перехода при $x = x_0 + a$ и стремится к 1 при $x \rightarrow \infty$.

Для задания параметров функции принадлежности необходимо задать:

- значение x_0 ;

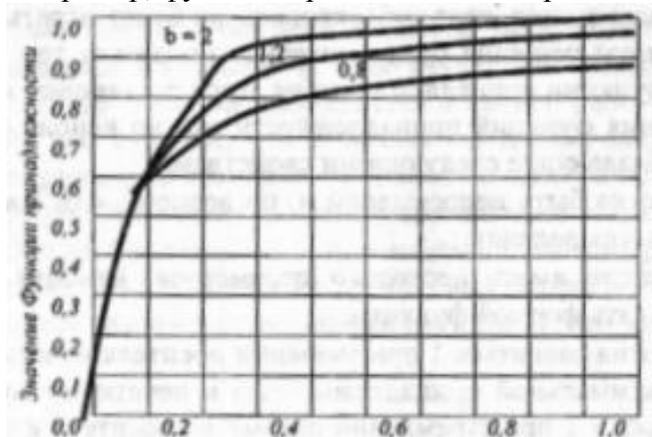
- значение $x_{0,5}$ на уровне $\mu=0,5$;
- значение x_c на уровне $\mu=c$, при $c>0,5$.

Параметры функции принадлежности определяются из выражений:

$$a = x_c.$$

$$b = \frac{\ln \frac{c}{1-c}}{\ln \frac{x_c - x_0}{a}}$$

Например, функция принадлежности при $x_0=0$ и $a = 0.1$

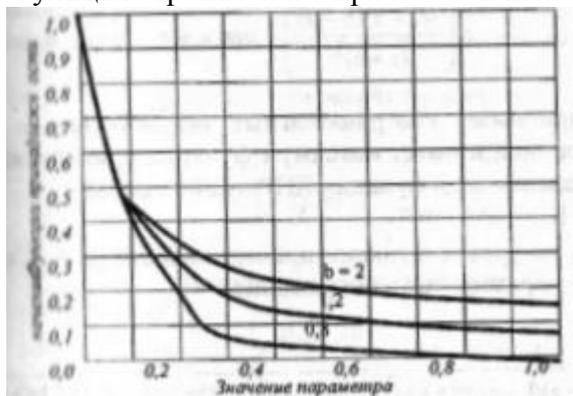


Например, составим функцию принадлежности «Старый».

Рассмотрим функцию принадлежности другого вида

$$\mu(x) = \begin{cases} 1 & \text{при } x \leq x_0 \\ 1 - (1 + ((x - x_c)/a)^{-b})^{-1} & \text{при } x > x_0 \end{cases}$$

Функция стремится к 0 при $x \rightarrow \infty$ и имеет точку перехода при $x = x_0 + a$. График функции:



Функция может использоваться для нечетких множеств типа «x по возможности не больше x_0 ».

Для задания параметров функции принадлежности необходимо задать:

- значение x_0 ;
- значение $x_{0,5}$ на уровне $\mu=0,5$;
- значение x_c на уровне $\mu=c$, при $c<0,5$.

Параметры функции принадлежности определяются из выражений:

$$a = x_c.$$

$$b = \frac{\ln \frac{1-c}{c}}{\ln \frac{x_c - x_0}{a}}$$

Раздел 7:

Нейронные сети в интеллектуально-адаптивных системах

Ещё со времён Ламарка развитие живого мира рассматривается как процесс постоянного совершенствования (приспособления) особей под влиянием среды. Моделируя отбор лучших планов, как процесс эволюции в популяции особей, можно получить решение задачи оптимизации, задав начальные условия эволюционного процесса, населив виртуальную вселенную существами — носителями информации и указав цель эволюционного процесса.

7.1. Метод роящихся частиц

Это наиболее простой и, что удивительно, один из самых молодых методов эволюционного программирования, появившийся в середине 90-х годов. Группа исследователей, занимавшихся этологией птиц и рыб, их социальной организацией, пришла к выводу о возможности решения задач оптимизации с помощью моделирования поведения групп животных. В основу метода положен тот факт, что при формировании стаи птицы стремятся к некоторому центру «притяжения», постепенно замедляя скорость полёта.

В реализации данного алгоритма многомерное пространство поиска населяется роем частиц (элементарных решений). Координаты частицы в пространстве однозначно определяют решение задачи оптимизации. Помимо координат каждая из частиц описывается скоростью перемещения и ускорением. В процессе перемещения частицы осуществляют «прочёсывание» пространства решений и тем самым находят текущий оптимум, к которому на следующем шаге устремляются остальные частицы. Для введения случайной составляющей в процесс поиска могут быть включены «сумасшедшие» частицы, закон движения которых отличается от закона движения остальных.

Схема работы алгоритма выглядит следующим образом:

1. Создаётся исходная «случайная» популяция частиц;
2. Для каждой частицы рассчитывается целевая функция;
3. Лучшая частица с точки зрения целевой функции объявляется «центром притяжения»;
4. Векторы скоростей всех частиц устремляются к этому «центру», при этом чем дальше частица находится от него, тем большим ускорением она обладает;
5. Осуществляется расчёт новых координат частиц в пространстве решений;
6. Шаги 2-5 повторяются заданное число раз;
7. Последний «центр тяжести» объявляется найденным оптимальным решением.

Этот алгоритм благодаря своей простоте (менее десяти строк кода) и скорости считается очень перспективным для задач планирования.

Одной из разновидности метода роящихся частиц является канонический метод роя частиц.

Рассмотрим задачу глобальной безусловной минимизации целевой функций $\Phi(\mathbf{X})$ в n -мерном арифметическом пространстве \mathbf{R}^n :

$$\min_{\mathbf{X} \in \mathbf{R}^n} \Phi(\mathbf{X}) = \Phi(\mathbf{X}^*) \quad (7.1)$$

Множество частиц обозначим $\mathbf{P} = \{P_i, i \in [1: N]\}$, где N — количество частиц в рое (размер популяции). В момент времени $t = 0, 1, 2, \dots$ координаты частицы P_i определяются вектором $\mathbf{X}_{it} = (x_{it,1}, x_{it,2}, \dots, x_{it,n})$, а ее скорость — вектором $\mathbf{V}_{it} = (v_{it,1}, v_{it,2}, \dots, v_{it,n})$. Начальные координаты и скорости частицы P_i равны $\mathbf{X}_{i0} = \mathbf{X}_i^0$, $\mathbf{V}_{i0} = \mathbf{V}_i^0$, соответственно.

Итерации в каноническом методе PSO выполняются по следующей схеме:

$$\mathbf{X}_{i,t+1} = \mathbf{X}_{i,t} + \mathbf{V}_{i,t+1}; \quad (7.2)$$

$$\mathbf{V}_{i,t+1} = \alpha \mathbf{V}_{i,t} + U[0, \beta] \otimes (\mathbf{X}_{i,t}^b - \mathbf{X}_{i,t}) + U[0, \gamma] \otimes (\mathbf{X}_{g,t} - \mathbf{X}_{i,t}). \quad (7.3)$$

Здесь $U[a, b]$ представляет собой n -мерный вектор псевдослучайных чисел, равномерно распределенных в интервале $[a, b]$; \otimes – символ покомпонентного умножения векторов; $\mathbf{X}_{i,t}^b$ – вектор координат частицы P_i с наилучшим (в смысле (1)) значением целевой функции $\Phi(\mathbf{X})$ за все время поиска; $\mathbf{X}_{g,t}$ – вектор координат соседней с данной частицы с наилучшим за время поиска значением целевой функции $\Phi(\mathbf{X})$; α, β, γ – свободные параметры алгоритма.

Пересчет координат частиц по формулам (7.2), (7.3) может происходить по синхронной схеме (обновление координат частиц выполняется только после определения текущих скоростей всех N частиц) или по асинхронной схеме (расчет новых координат частиц производится до завершения указанных вычислений).

В процессе итераций вектор $\mathbf{X}_{i,t}^b$ образует, так называемый, собственный путь (private guide) частицы P_i , а вектор $\mathbf{X}_{g,t}$ – локальный путь (local guide) этой частицы.

Свободный параметр α определяет «инерционные» свойства частиц (при $\alpha < 1$ движение частиц, очевидно, замедляется). Рекомендуемое значение параметра α равно 0.7298. В процессе оптимизации может быть эффективным постепенное уменьшение коэффициента α от 0.9 до 0.4. При этом большие значения параметра обеспечивают широкий обзор пространства поиска, а малые – точную локализацию минимума целевой функции. Рекомендуемые значения свободных параметров β, γ равны 1.49618.

Важнейшим понятием в методе PSO является понятие соседства частиц, которое определяется соответствующей топологией соседства.

Второй компонент в формуле (3) называется «когнитивным» компонентом (по социальной аналогии) и формализует тенденцию частицы вернуться в положение с минимальным значением целевой функции. Третий компонент в формуле (3) называется «социальным» компонентом. Компонент отражает влияние на данную частицу ее соседей.

7.2. Генетические алгоритмы

Генетические алгоритмы это алгоритмы, использующие методы «эволюционного вычисления». Эволюционное вычисление занимается решением проблем, используя приложения эволюционных механизмов. Эта область развилась из трех более или менее независимых разработок: генетических алгоритмов, эволюционного программирования и эволюционных стратегий. Весь три дисциплины начали изучаться в 60-х годах и 70-х годах (первые две в США, третья в Германии), но только в конце 80-х годов они начали признаваться. В настоящее время генетические алгоритмы рассматриваются как одни из наиболее успешных методов машинного обучения. По существу, они являются процедурами оптимизации, имитирующими при проектировании модели данных такие процессы, как генетическая рекомбинация, мутация и отбор, аналогичные тем, что обуславливают естественную эволюцию. Первые генетические алгоритмы были предложены в начале 70-х годов Джоном Холландом (John Holland) с целью имитации эволюционных процессов в живой природе. Схему конструирования генетического алгоритма для решения задач можно представить следующим образом:

1. разработайте хорошее кодирование задачи в терминах строк ограниченного алфавита;

2. постройте искусственную среду в компьютере, где решения могут конкурировать друг с другом. Обеспечьте объективную оценку успеха или неудачи, что в профессиональных терминах называется функцией пригодности;

3. разработайте способы, которыми возможные решения могут быть объединены. Здесь очень популярна так называемая операция скрещивания (кроссинговер), в которой строки отца и матери просто отрезаются и после выделения отрезанные части объединяются вместе. При репродукции могут быть применены другие операторы, такие как операторы мутации, инверсии и т.д.;

4. введите хорошо изменяемую начальную популяцию и заставьте компьютер играть в «эволюцию», удаляя плохие решения из каждого поколения и заменяя их потомством или мутациями хороших решений;

5. остановитесь, когда будет порождено семейство успешных решений.

Схема достаточно проста, но успех будет зависеть, прежде всего, от адекватности задачи кодирования (так называемая техника представления) и нахождения эффективных операторов мутации.

Удачи и недостатки генетических алгоритмов частично сходны с таковыми при естественном отборе вообще. Двумя недостатками являются большое перепроизводство объектов и случайный характер процесса поиска. В общем случае требуется масса вычислительной мощности, чтобы достигнуть чего-нибудь значительного. С другой стороны, методика устойчива; если решение существует, то генетический алгоритм с большой вероятностью найдет его. Однако, в специфической проблемной области исследования операций, генетические алгоритмы часто не согласовываются с общепринятыми алгоритмами, их работоспособность главным образом является следствием их широкой применимости и концептуальной ясности. Вместе со своими родственниками, нейронными сетями, они являются ядром самообучающихся систем.

Схема работы генетических алгоритмов выглядит следующим образом:

1. Задается способ кодирования параметров задачи в «хромосомах»;
2. Создается исходная популяция (начальное решение);
3. Рассчитывается функция приспособленности каждой особи;
4. Для наиболее приспособленных особей производится «скрещивание» и рождение новых особей, содержащих признаки обоих родителей. При этом передача признаков осуществляется в соответствии с одним из выбранных способов «обмена участком хромосом». Наименее приспособленные особи «отмирают» и заменяются новорожденными;

5. Осуществляется мутация некоторой части особей добавлением в процесс поиска элемента случайности;

6. Шаги 2-5 итерационно повторяются заданное число раз;

7. Из оставшейся популяции выбираются лучшие особи, «хромосомы» декодируются, а получаемое таким образом решение соответствует локальному оптимуму.

7.3. Моделирование отжига (Simulated Annealing)

Метод, разработанный в 80-х годах, основан на аналогии с физическим процессом охлаждения металла и переходом его в состояние с минимальной энергией кристаллической решетки. Изначально он использовался для моделирования поведения системы атомов при разных температурах, но впоследствии был расширен для решения комбинаторных задач.

Схема применения метода достаточно проста:

1. Задается начальное условие;
2. Выбираются соседние состояния и определяется вероятность перехода в каждое из этих состояний, которая зависит от их «энергии»;
3. Уменьшается температура (от нее зависит вероятность перехода в новое состояние);
4. Шаги 2-3 повторяются заданное число раз.

Раздел 8:

Понятие экспертной системы (ЭС). Структура ЭС. Классификации ЭС

8.1 Экспертные системы: базовые понятия

В современном обществе при решении задач управления сложными многопараметрическими и сильно связанными системами, объектами, производственными и технологическими процессами приходится сталкиваться с решением неформализуемых либо трудно формализуемых задач. Такие задачи часто возникают в следующих областях: авиация, космос и оборона, нефтеперерабатывающая промышленность и транспортировка нефтепродуктов, химия, энергетика, металлургия, целлюлозно-бумажная промышленность, телекоммуникации и связь, пищевая промышленность, машиностроение, производство цемента, бетона и т.п. транспорт, медицина и фармацевтическое производство, административное управление, прогнозирование и мониторинг. Наиболее значительными достижениями в этой области стало создание систем, которые ставят диагноз заболевания, предсказывают месторождения полезных ископаемых, помогают в проектировании электронных устройств, машин и механизмов, решают задачи управления реакторами и другие задачи.

Экспертные системы – это прикладные системы ИИ, в которых база знаний представляет собой формализованные эмпирические знания высококвалифицированных специалистов (экспертов) в какой-либо узкой предметной области. Экспертные системы предназначены для замены при решении задач экспертов в силу их недостаточного количества, недостаточной оперативности в решении задачи или в опасных (вредных) для них условиях.

Экспертная система (ЭС) - программа, которая использует знания специалистов (экспертов) о некоторой конкретной узко специализированной предметной области и в пределах этой области способна принимать решения на уровне эксперта-профессионала.

Осознание полезности систем, которые могут копировать дорогостоящие или редко встречающиеся человеческие знания, привело к широкому внедрению и расцвету этой технологии в 80-е, 90-е годы прошлого века.

Основу успеха ЭС составили два важных свойства:

1. в ЭС знания отделены от данных, и мощность экспертной системы обусловлена в первую очередь мощностью базы знаний и только во вторую очередь используемыми методами решения задач;

2. решаемые ЭС задачи являются неформализованными или слабоформализованными и используют эвристические, экспериментальные, субъективные знания экспертов в определенной предметной области.

Основными категориями решаемых ЭС задач являются:

- диагностика,
- управление (в том числе технологическими процессами),
- интерпретация,
- прогнозирование,
- проектирование,
- отладка и ремонт,
- планирование,
- наблюдение (мониторинг),
- обучение.

С экспертными системами связаны следующие распространенные заблуждения:

1. ЭС будут делать не более (а скорее даже менее) того, чем может эксперт, создавший данную систему. Для опровержения данного постулата можно построить самообучающуюся

ЭС в области, в которой вообще нет экспертов, либо объединить в одной ЭС знания нескольких экспертов, и получить в результате систему, которая может то, чего ни один из ее создателей не может.

2. ЭС никогда не заменит человека-эксперта. Уже заменяет, иначе зачем бы их создавали?

Примерами широко известных и эффективно используемых (или использованных в свое время) экспертных систем являются:

- DENDRAL – ЭС для распознавания структуры сложных органических молекул по результатам их спектрального анализа (считается первой в мире экспертной системой);

- MOLGEN – ЭС для выработки гипотез о структуре ДНК на основе экспериментов с ферментами;

- XCON – ЭС для конфигурирования (проектирования) вычислительных комплексов VAX 11 в корпорации DEC в соответствии с заказом покупателя;

- MYCIN – ЭС диагностики кишечных заболеваний;

- PUFF – ЭС диагностики легочных заболеваний;

- MACSYMA – ЭС для символьных преобразований алгебраических выражений;

- YES/MVS – ЭС для управления многозадачной операционной системой MVS больших ЭВМ корпорации IBM;

- DART – ЭС для диагностики больших НМД корпорации IBM;

- PROSPECTOR – ЭС для консультаций при поиске залежей полезных ископаемых;

- AIRPLANE – экспертная система для помощи летчику при посадке на авианосец;

- ЭСПЛАН – ЭС для планирования производства на Бакинском нефтеперерабатывающем заводе;

- МОДИС – ЭС диагностики различных форм гипертонии;

- МИДАС – ЭС для идентификации и устранения аварийных ситуаций в энергосистемах.

8.2 Классификация экспертных систем

1. По назначению ЭС делятся на:

1.1. ЭС общего назначения;

1.2. Специализированные ЭС;

1.3. проблемно-ориентированные для задач диагностики, проектирования, прогнозирования;

1.4. предметно-ориентированные для специфических задач, например, контроля ситуаций на атомных электростанциях.

2. По степени зависимости от внешней среды выделяют:

2.1. Статические ЭС, не зависящие от внешней среды;

2.2. Динамические, учитывающие динамику внешней среды и предназначенные для решения задач в реальном времени. Время реакции в таких системах может задаваться в миллисекундах, и эти системы реализуются, как правило, на языке C++.

3. По типу использования различают:

3.1. Изолированные ЭС;

3.2. ЭС на входе/выходе других систем;

3.3. Гибридные ЭС или, иначе говоря, ЭС интегрированные с базами данных и другими программными продуктами (приложениями).

4. По сложности решаемых задач различают:

4.1. Простые ЭС - до 1000 простых правил;

4.2. Средние ЭС - от 1000 до 10000 структурированных правил;

4.3. Сложные ЭС - более 10000 структурированных правил.

5. По стадии создания выделяют:

5.1. Исследовательский образец ЭС, разработанный за 1-2 месяца с минимальной БЗ;

5.2. Демонстрационный образец ЭС, разработанный за 2-4 месяца, например, на языке типа LISP, PROLOG, CLIPS;

5.3. Промышленный образец ЭС, разработанный за 4-8 месяцев, например, на языке типа CLIPS с полной БЗ;

5.4. Коммерческий образец ЭС, разработанный за 1,5-2 года, например, на языке типа C++, Java с полной БЗ.

8.3 Составные части экспертной системы и порядок ее функционирования

Основными компонентами ЭС являются ниже перечисленные элементы.

Машина логического вывода – механизм рассуждений, оперирующий знаниями и данными с целью получения новых данных из знаний и других данных, имеющихся в рабочей памяти. Для этого обычно используется программно реализованный механизм дедуктивного логического вывода (какая-либо его разновидность) или механизм поиска решения в сети фреймов или семантической сети.

Машина логического вывода может реализовывать рассуждения в виде:

- дедуктивного вывода (прямого, обратного, смешанного);
- нечеткого вывода;
- вероятностного вывода;
- унификации (подобно тому, как это реализовано в Прологе);
- поиска решения с разбиением на последовательность подзадач;
- поиска решения с использованием стратегии разбиения пространства поиска с учетом уровней абстрагирования решения или понятий, с ними связанных;
- монотонного или немонотонного рассуждения;
- рассуждений с использованием механизма аргументации;
- ассоциативного поиска с использованием нейронных сетей;
- вывода с использованием механизма лингвистической переменной.

Подсистема общения служит для ведения диалога с пользователем, в ходе которого ЭС запрашивает у пользователя необходимые факты для процесса рассуждения, а также, дающая возможность пользователю в какой-то степени контролировать и корректировать ход рассуждений экспертной системы.

Подсистема объяснений необходима для того, чтобы дать возможность пользователю контролировать ход рассуждений экспертной системы. Если нет этой подсистемы, экспертная система выглядит для пользователя как «вещь в себе», решениям которой можно либо верить, либо нет. Нормальный пользователь выбирает последнее, и такая ЭС не имеет перспектив для использования.

Подсистема приобретения знаний служит для корректировки и пополнения базы знаний. В простейшем случае это – интеллектуальный редактор базы знаний, в более сложных экспертных системах – средства для извлечения знаний из баз данных, неструктурированного текста, графической информации и т.д.

Порядок взаимодействия составных частей ЭС приведен на функциональной схеме ЭС – рис. 8.1.

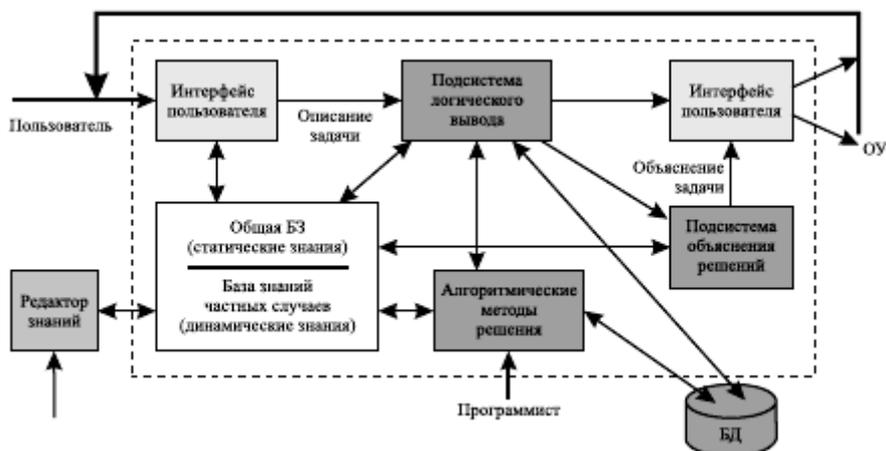


Рис. 8.1. – Функциональная схема ЭС

Основу ЭС составляет подсистема логического вывода, которая использует информацию из базы знаний (БЗ), генерирует рекомендации по решению искомой задачи. Чаще всего для представления знаний в ЭС используются системы продукций и семантические сети. Допустим, БЗ состоит из фактов и правил (если <посылка> то <заключение>). Если ЭС определяет, что посылка верна, то правило признается подходящим для данной консультации и оно запускается в действие. Запуск правила означает принятие заключения данного правила в качестве составной части процесса консультации.

Обязательными частями любой ЭС являются также модуль приобретения знаний и модуль отображения и объяснения решений. В большинстве случаев, реальные ЭС в промышленной эксплуатации работают также на основе баз данных (БД). Только одновременная работа со знаниями и большими объемами информации из БД позволяет ЭС получить неординарные результаты, например, поставить сложный диагноз (медицинский или технический), открыть месторождение полезных ископаемых, управлять ядерным реактором в реальном времени.

8.4 Функционирование базы знаний экспертной системы

Обычно при описании баз знаний (как правило продукционных) экспертных систем правила представляются в наглядном виде, например:

```
ПРАВИЛО _____ 1 :  
ЕСЛИ  
Образование=Высшее И Возраст=Молодой И  
Коммуникабельность=Высокая  
ТО  
Шансы найти работу=Высокие КД=0.9.
```

При срабатывании этого правила в базу данных интеллектуальной системы (например, экспертной системы) добавляется факт, означающий, что шансы найти работу высоки с достоверностью 0.9 или 90 % (значение коэффициента достоверности КД). Понятия «Образование», «Возраст», «Коммуникабельность» служат для задания условия (в данном случае, конъюнкции), при котором срабатывает правило.

Факты хранятся в базе данных продукционной системы в форме
(Объект, значение, коэффициент достоверности)

или

(Объект, атрибут, значение, коэффициент достоверности).

Кроме правил в продукционных базах знаний могут использоваться метаправила для управления логическим выводом. Пример метаправила для гипотетической базы знаний:

```
ЕСЛИ  
Экономика = развивается  
ТО  
Увеличить приоритет правила 1
```

При интерпретации (выполнении) правила в ходе проверки условия система проверяет факты, находящиеся уже в базе данных, и, если соответствующего факта нет, обращается за ним к источнику данных (пользователю, базе данных и т.д.) с вопросом (или запросом).

В БЗ ЭС могут использоваться и другие структуры для хранения фактов, такие как семантические сети или фреймы. В этом случае говорят о комбинации разных методов представления знаний или о гибридных интеллектуальных экспертных системах.

Для представления нечетких знаний факты и правила в продукционных системах снабжаются коэффициентами достоверности (или уверенности), которые могут принимать значения из разных интервалов в разных системах (например, (0, 1), (0, 100), (-1, +1)).

8.5. Обратный метод логического дедуктивного вывода

Для решения задач в продукционной интеллектуальной системе существует два основных метода дедуктивного логического вывода: обратный и прямой. Может использоваться и комбинация этих двух методов.

При обратном логическом выводе процесс интерпретации правил начинается с правил, непосредственно приводящих к решению задачи. В них в правой части находятся заключения с фактами, являющимися решением (целевыми фактами). При интерпретации этих правил в процесс решения могут вовлекаться другие правила, результатом выполнения которых являются факты, участвующие в условиях конечных правил и т.д.

Метод обратного логического вывода можно применять тогда, когда необходимо минимизировать количество обращений к источнику данных (например, пользователю), исключив из рассмотрения заведомо ненужные для решения задачи факты.

В самом общем виде алгоритм обратного логического вывода, записанный на псевдокоде в виде функции, выглядит так (условие ограничено конъюнкцией элементарных условий).

```
функция Доказана_Цель (Цель) : boolean;
Поместить Цель в стек целей.
пока стек целей не пуст цикл
  Выбор цели из стека целей и назначение ее текущей.
  Поиск множества правил, в правой части которых находится
  текущая цель (множества подходящих правил).
  Считать, что Цель не доказана.
  пока множество подходящих правил не пусто
  и Цель не доказана
  цикл
  Выбор из этого множества одного текущего правила с
  использованием определенной стратегии. Считать текущим
  элементарным условием первое.
  пока не проверены все элементарные условия правила и не
  надо прервать проверку условия
  цикл
  если в текущем элементарном условии участвует
  факт, встречающийся в правой части какого-то
  правила
  то
  если не Доказана_Цель (Этот факт) то
  Надо прервать проверку условия
  конец_если иначе
  Запросить информацию о факте.
  Проверить элементарное условие.
  если элементарное условие истинно то
  Добавить факт в базу данных. Перейти к
  следующему элементарному Условию.
  иначе
  Надо прервать проверку условия.
  конец_если
конец_если
конец_если
конец_цикла
если условие правила истинно то
Выполнить заключение. Исключить Цель из стека
целей. Считать, что Цель доказана.
конец_если
конец_цикла
конец_цикла
конец_функции.
```

Существует много различных стратегий выбора правила из подходящих. Наиболее простой и часто встречающейся стратегией является «первая попавшаяся». При этой стратегии решение задачи зависит от порядка расположения (перебора) правил в базе знаний.

8.6. Прямой метод логического дедуктивного вывода

В прямом методе логического вывода интерпретация правил начинается от известных фактов, т.е. сначала выполняются правила, условия которых можно проверить с использованием фактов, уже находящихся в базе данных.

В общем виде алгоритм прямого вывода приведен ниже.

пока Цель не доказана цикл

Формирование множества подходящих правил (по их условиям и наличию фактов) .

Выбор одного правила из этого множества (с использованием определенной стратегии выбора) . Считать текущим элементарным условием первое .

пока не проверены все элементарные условия правила и не надо прервать проверку условия

цикл

если элементарное условие истинно то

Перейти к следующему элементарному условию .

иначе

Надо прервать проверку условия .

конец_если конец_цикла

Выполнить заключение .

если при формировании заключения появился целевой факт то

Считать, что Цель доказана .

конец_если конец_цикла .

Метод прямого логического вывода можно применять тогда, когда факты появляются в базе данных не зависимо от того, какую задачу сейчас требуется решить (какой целевой факт доказать) и в разные моменты времени. В этом случае можно говорить о том, что факты управляют логическим выводом (решением задачи). Кроме того, этот метод целесообразно применять для формирования вторичных признаков (фактов) из первичных для подготовки решения задачи в дальнейшем с применением обратного логического вывода.

4.3. Лабораторные работы

Учебным планом не предусмотрены.

4.4. Практические занятия

| <i>№ п/п</i> | <i>Номер раздела дисциплины</i> | <i>Наименование тем практических занятий</i> | <i>Объем (час.)</i> | <i>Вид занятия в интерактивной, активной, инновационной формах, (час.)</i> |
|--------------|---------------------------------|--|---------------------|--|
| 1 | 2. | Методы распознавания образов, основанные на построении разделяющей гиперплоскости и гиперсферы | 4 | - |
| 2 | 3, 8. | Моделирование простой нейронной сети | 6 | - |
| 3 | 4. | Метод Форель | 4 | - |
| 4 | 5. | Метод масок | 4 | - |
| 5 | 7, 8. | Моделирование и обучение нейронной сети, типа перцептрон | 6 | Разбор конкретных ситуаций (4 часа) |
| ИТОГО | | | 24 | 4 |

4.5. Контрольные мероприятия: контрольная работа

Цель: разработка экспертной системы для поддержки принятия управленческих или диагностических решений в оболочке ESWin.

Структура: каждое индивидуальное задание предполагает разработку обучающимся базы знаний, которая должна включать:

- фрейм-образ в соответствии с индивидуальным заданием;
- фрейм цели в соответствии с индивидуальным заданием;
- правила логического вывода;
- заключения о возможности/невозможности решения задачи.

Основная тематика: разработка экспертной системы в среде ESWin.

Рекомендуемый объём: пояснительная записка объёмом 10-15 страниц должна содержать титульный лист, индивидуальное задание, текст базы знаний, скриншоты работы программы ESWin с комментариями, заключение.

Выдача задания, приём кр проводится в соответствии с календарным учебным графиком.

| Оценка | Критерии оценки контрольной работы |
|---------------|---|
| зачтено | Контрольная работа сдана в установленные сроки. Текст базы знаний и скриншоты экспертной системы ESWin в пояснительной записке соответствуют стандартам ФГБОУ ВО «БрГУ». Ошибки и замечания исправлены. |
| не зачтено | Контрольная работа не сдана в установленные сроки. |

5. МАТРИЦА СООТНЕСЕНИЯ РАЗДЕЛОВ УЧЕБНОЙ ДИСЦИПЛИНЫ К ФОРМИРУЕМЫМ В НИХ КОМПЕТЕНЦИЯМ И ОЦЕНКЕ РЕЗУЛЬТАТОВ ОСВОЕНИЯ ДИСЦИПЛИНЫ

| <i>№, наименование разделов дисциплины</i> | <i>Компетенции</i> | <i>Кол-во часов</i> | <i>Компетенции</i> | Σ <i>комп.</i> | t_{cp} , час | <i>Вид учебных занятий</i> | <i>Оценка результатов</i> |
|--|--------------------|---------------------|--------------------|--------------------------|----------------|----------------------------|---------------------------|
| | | | <i>ПК</i> | | | | |
| | | | 11 | | | | |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | |
| 1. Состояние и перспективы развития теории и практики интеллектуальных систем управления инфокоммуникациями | | 12 | + | 1 | 12 | Лк, СРС | Зачёт |
| 2. Структура интеллектуально-адаптивных систем | | 16 | + | 1 | 18 | Лк, ПЗ, СРС | Зачёт |
| 3. Методы построения алгоритмов адаптации | | 15 | + | 1 | 17 | Лк, ПЗ, СРС | Зачёт |
| 4. Системы прямого управления | | 15 | + | 1 | 17 | Лк, ПЗ, СРС | Зачёт |
| 5. Системы непрямого управления | | 15 | + | 1 | 17 | Лк, ПЗ, СРС | Зачёт |
| 6. Представление знаний в интеллектуальных системах | | 9 | + | 1 | 9 | Лк, СРС | Зачёт |
| 7. Нейронные сети в интеллектуально-адаптивных системах автоматического управления | | 13 | + | 1 | 15 | Лк, ПЗ, СРС | Зачёт |
| 8. Понятие экспертной системы (ЭС). Структура ЭС. Классификации ЭС | | 13 | + | 1 | 15 | Лк, ПЗ, СРС | Зачёт, кр |
| всего часов | | 108 | 108 | 1 | 108 | | |

6. ПЕРЕЧЕНЬ УЧЕБНО-МЕТОДИЧЕСКОГО ОБЕСПЕЧЕНИЯ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ ОБУЧАЮЩИХСЯ ПО ДИСЦИПЛИНЕ

1. Смолин, Д. В. Введение в искусственный интеллект: конспект лекций [Текст] : учебное пособие / Д. В. Смолин. - М. : Физматлит, 2004. - 208 с.
2. Люгер, Дж. Ф. Искусственный интеллект. Стратегии и методы решения сложных проблем : учебное пособие / Дж. Ф. Люгер; Пер. с англ. - 4-е изд. - Москва : Вильямс, 2005. - 864 с.
3. Хабаров, С. П. Интеллектуальные информационные системы. PROLOG- язык разработки интеллектуальных и экспертных систем [Электронный ресурс] : учебное пособие / С. П. Хабаров. - Санкт-Петербург : СПбГЛТУ, 2013. - 138 с.

7. ПЕРЕЧЕНЬ ОСНОВНОЙ И ДОПОЛНИТЕЛЬНОЙ ЛИТЕРАТУРЫ, НЕОБХОДИМОЙ ДЛЯ ОСВОЕНИЯ ДИСЦИПЛИНЫ

| № | Наименование издания | Вид занятия | Количество экземпляров в библиотеке, шт. | Обеспеченность, (экз./ чел.) |
|----------------------------------|--|-------------|--|------------------------------|
| 1 | 2 | 3 | 4 | 5 |
| Основная литература | | | | |
| 1. | Гаскаров Д.В. Интеллектуальные информационные системы. Учеб. для вузов. - М.: Высш. шк., 2003. | Лк | 15 | 1,0 |
| 2. | Рак, И.П. Интеллектуальные информационные системы: методические указания / сост.: И.П. Рак, А.В. Селезнёв, Э.В. Сысоев. - Тамбов: Издательство ТГТУ, 2010. - 16 с. http://window.edu.ru/resource/187/73187 | Лк, кр | ЭР | 1,0 |
| 3. | Рутковский, Л. Методы и технологии искусственного интеллекта : учебник / Л. Рутковский; Пер. с пол. - М. : Горячая линия- Телеком, 2010. - 520 с. | ПЗ | 10 | 0,7 |
| Дополнительная литература | | | | |
| 4. | Барский, А.Б. Нейронные сети: распознавание, управление, принятие решений. – М.: Финансы и статистика, 2004. | Лк | 2 | 0,1 |
| 5. | Кедрин, В. С. Искусственные нейронные сети : учебное пособие / В. С. Кедрин, Ю. А. Шичкина. - Братск : БрГУ, 2012. - 107 с. | ПЗ | 91 | 1,0 |
| 6. | Иванов, М. Ю. Основы логического программирования : методические указания / М. Ю. Иванов. - Братск : БрГУ, 2009. - 27 с. | кр | 31 | 1,0 |
| 7. | Уткин, В. Б. Информационные технологии управления : учебник для вузов / В. Б. Уткин, К. В. Балдин. - Москва : Академия, 2008. - 400 с. | Лк | 10 | 0,7 |

8. ПЕРЕЧЕНЬ РЕСУРСОВ ИНФОРМАЦИОННО-ТЕЛЕКОММУНИКАЦИОННОЙ СЕТИ «ИНТЕРНЕТ» НЕОБХОДИМЫХ ДЛЯ ОСВОЕНИЯ ДИСЦИПЛИНЫ

1. Электронный каталог библиотеки БрГУ
http://irbis.brstu.ru/CGI/irbis64r_15/cgiirbis_64.exe?LNG=&C21COM=F&I21DBN=BOOK&P21DBN=BOOK&S21CNR=&Z21ID=.
2. Электронная библиотека БрГУ
<http://ecat.brstu.ru/catalog>.
3. Электронно-библиотечная система «Университетская библиотека online»
<http://biblioclub.ru>.
4. Электронно-библиотечная система «Издательство «Лань»
<http://e.lanbook.com>.

5. Информационная система "Единое окно доступа к образовательным ресурсам"
<http://window.edu.ru> .

6. Научная электронная библиотека eLIBRARY.RU <http://elibrary.ru> .

7. Университетская информационная система РОССИЯ (УИС РОССИЯ)
<https://uisrussia.msu.ru/> .

8. Национальная электронная библиотека НЭБ
<http://xn--90ax2c.xn--p1ai/how-to-search/> .

9. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ

9.1. Методические указания для обучающихся по выполнению практических занятий

Практическое занятие №1

Методы распознавания образов, основанные на построении разделяющей гиперплоскости и гиперсферы

Цель работы:

Изучить методы распознавания образов, основанные на построении разделяющей гиперплоскости и гиперсферы.

Задание:

Входные данные:

- задать мерность пространства – N (максимальное значение 5);
- файл с координатами точек и их номерами (максимальное число точек - 10).

Разработать программное средство, реализующее следующие функции:

1. считывание номера и координаты точек из файла;
2. возможность выбора метода распознавания: гиперплоскость или гиперсфера.

Для гиперплоскости:

1. выбрать две базовые точки. Предусмотреть следующие варианты выбора: случайным образом, должны задаваться пользователем и выбрать две точки с наименьшим расстоянием между собой из всех возможных вариантов (нужно рассмотреть все возможные пары точек и определить расстояние между ними по формуле $A_1A_2 = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2}$;
2. разбить точки на два класса;
3. вывести на экран или в файл: номера точек и в какой класс они попали.

Для гиперсферы:

1. выбрать центр гиперсферы. Предусмотреть следующие варианты: случайном образом из множества точек или задается номер точки пользователем;
2. радиус сферы должен вычисляться двумя способами: задается пользователем или вычисляется следующим образом:

а) находим расстояния R_i между центром гиперсферы и с каждой из оставшихся точек по формуле $R_i = \sqrt{(O_1 - X_1)^2 + (O_2 - X_2)^2 + \dots + (O_n - X_n)^2}$, где (O_1, O_2, \dots, O_n) – координаты центра сферы, а (X_1, X_2, \dots, X_n) – координаты точки;

б) находим среднее расстояние $\frac{\sum_{i=1}^{n-1} R_i}{n-1}$, где n – общее количество точек. Это среднее расстояние и будет радиусом.

3. Строим гиперсферу и производим разделение всех точек на два класса;
4. Вывести на экран или в файл: номера точек и в какой класс попали.

Порядок выполнения:

Соответствует пунктам задания.

Форма отчетности:

Отчет сдается в печатном виде. В отчете должны присутствовать:

1. отмеченные на координатной плоскости точки примера 1 с нумерацией;
 2. ручной просчет для примера 1;
 3. результат, который выдает написанная обучающимся программа для примера 1;
- Ручной просчет – это описание решения деления на классы множества точек методом гиперплоскости и гиперсферы.

Пример 1 рассматривает точки в двухмерном пространстве.

Этапы составления отчета:

- для примера 1 нарисовать на координатной плоскости 6 точек, любые две точки из них выбрать в качестве базовых, выделить их на плоскости;
- нарисовать на координатной плоскости гиперплоскость. Записать уравнение гиперплоскости;
- описать ручной просчет для метода гиперплоскость относительно выделенных базовых точек;
- отметить на координатной плоскости, в какой класс попали оставшиеся четыре точки;
- снова нарисовать на координатной плоскости те же 6 точек, выбрать одну из них в качестве центра гиперсферы и рассчитать радиус гиперсферы;
- нарисовать гиперсферу на плоскости и записать ее уравнение;
- описать ручной просчет для метода гиперсфера относительно заданного центра и рассчитанного радиуса;
- отметить на координатной плоскости, в какой класс попали оставшиеся пять точек;
- привести результат работы программы относительно примера 1 по двум методам (можно вставить скриншоты).

Задания для самостоятельной работы:

Предусмотрены вариантом обучающегося.

Рекомендации по выполнению заданий и подготовке к практическому занятию

Ознакомиться с теоретическим материалом, представленным во втором разделе данной дисциплины.

Основная литература

1. Рутковский, Л. Методы и технологии искусственного интеллекта : учебник / Л. Рутковский; Пер. с пол. - М. : Горячая линия- Телеком, 2010. - 520 с.

Дополнительная литература

2. Кедрин, В. С. Искусственные нейронные сети : учебное пособие / В. С. Кедрин, Ю. А. Шичкина. - Братск : БрГУ, 2012. - 107 с.

Практическое занятие №2

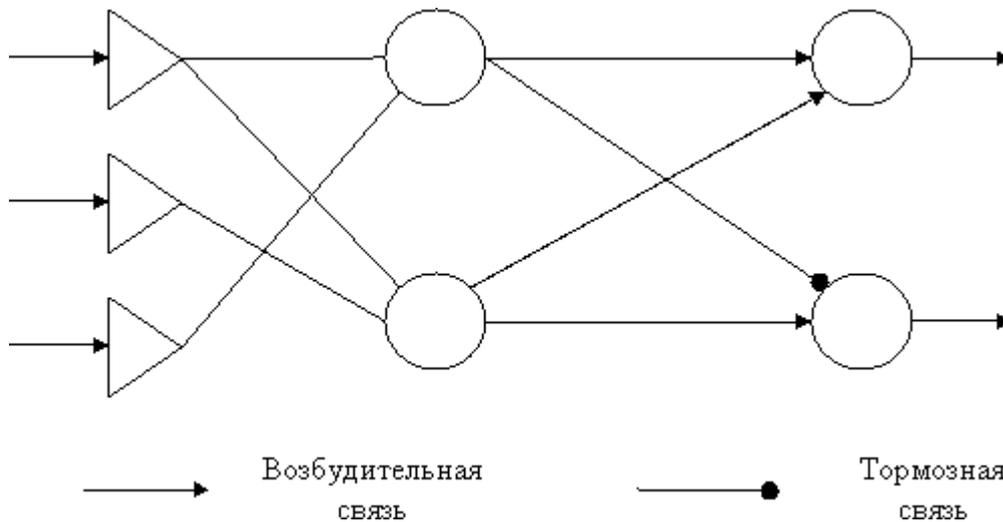
Моделирование простой нейронной сети

Цель работы:

Изучить формальную модель нейрона и рассмотреть взаимодействие нейронов между собой.

Задание:

Необходимо разработать программу, имитирующую работу простейшей нейронной сети, состоящей из 4 нейронов.



В программе должны быть реализованы следующие функции:

1. количество рецепторов фиксировано 3;
2. количество нейронов фиксировано 4;
3. задаются связи между рецепторами и двумя ближайшими нейронами;
4. связи между нейронами фиксированы: три возбудительные и одна тормозная;
5. задается раздражение на рецепторах (0 – нет раздражения, 1 – есть раздражение);
6. задается порог на каждом нейроне;
7. задается время суммации – один или два.

Порядок выполнения:

Изучить теоретические данные.

Алгоритм работы:

1. Суммирование 1.

- а) на рецепторы подается сигнал в виде нулей и единиц, т.е. каждый рецептор содержит одну цифру (0 или 1);
- б) рецепторы, содержащие единицу, активируют свои связи к нейронам, значение на нейроне увеличивается на единицу;
- в) далее рассматриваются все нейроны, значение которых изменилось. Если получившееся значение превышает пороговое значение рассматриваемого нейрона, то этот нейрон активирует все свои выходящие связи к другим нейронам, если это увеличивающая связь, то значение на результирующем нейроне увеличивается на единицу, если это тормозящая связь, то значение на результирующем нейроне уменьшается на единицу;
- г) если значение на выходном нейроне, превышает пороговое значение нейрона, то на выход подается единица, иначе ноль.

2. Суммирование 2.

- а) на рецепторы подается сигнал в виде нулей и единиц, т.е. каждый рецептор содержит одну цифру (0 или 1);
- б) рецепторы, содержащие единицу, активируют свои связи к нейронам, значение на нейроне увеличивается на единицу;
- в) далее рассматриваются все нейроны, значение которых изменилось. Если получившееся значение превышает пороговое значение рассматриваемого нейрона, то этот нейрон активирует все свои входящие связи к другим нейронам, если это увеличивающая связь, то значение на результирующем нейроне увеличивается на единицу, если это тормозящая связь, то значение на результирующем нейроне уменьшается на единицу;
- г) переход к п. 1 при этом значение на нейронах не сбрасываются. Переход к п. 1 осуществляется один раз. Далее переход к п. 5. Если значение на выходном нейроне, превышает пороговое значение нейрона, то на выход подается единица, иначе ноль.

Форма отчетности:

Отчет должен состоять из следующих пунктов:

1. Нарисовать нейронную сеть;

2. Отметить на рецепторах раздражение и пороги на нейронах;
3. Вручную просчитать работу сети, отметить промежуточные результаты и записать сигналы на выходе;
4. Вставить результат написанной вами программы для нейронной сети с такими же параметрами (желательно в виде скриншота);
5. Отметить второе раздражение на рецепторах;
6. Вручную просчитать работу сети, только уже при времени суммации два. Отметить полученные результаты;
7. Вставить результат написанной вами программы для нейронной сети с такими же параметрами и временем суммации 2 (желательно в виде скриншота).

Задания для самостоятельной работы:

Предусмотрены вариантом обучающегося.

Рекомендации по выполнению заданий и подготовке к практическому занятию

Ознакомиться с теоретическим материалом, представленным в третьем и восьмом разделах данной дисциплины.

Основная литература

1. Рутковский, Л. Методы и технологии искусственного интеллекта : учебник / Л. Рутковский; Пер. с пол. - М. : Горячая линия- Телеком, 2010. - 520 с.

Дополнительная литература

2. Кедрин, В. С. Искусственные нейронные сети : учебное пособие / В. С. Кедрин, Ю. А. Шичкина. - Братск : БрГУ, 2012. - 107 с.

Практическое занятие №3

Метод Форель

Цель работы:

Изучить метод распознавания Форель, определить его особенности и отличия от методов распознавания, основанных на построении гиперплоскости и гиперсферы.

Задание:

Входные данные:

1. задается мерность пространства N (максимальное значение 10);
 2. Файл с координатами точек и их номерами (максимальное число точек - 10).
- Разработать программное средство, реализующее следующие функции:
3. считать координаты точек из файла;
 4. задать начальную точку для метода «Форель»;
 5. предусмотреть возможность задания радиуса пользователем или радиус рассчитывается аналогичным образом, как для гиперсферы в первой лабораторной работе, только нужно реализовать возможность выбор среднего, минимального или максимального расстояния между точками;
 6. выполнить деление точек на классы методом Форель;
 7. вывести на экран или в файл номера точек и к какому классу они принадлежат.

Порядок выполнения:

Изучить теоретические данные. Выполнить пункты задания 1 – 7.

Форма отчетности:

Отчет сдается в печатном виде. В отчете должны присутствовать:

1. Титульный лист;
2. Постановка задачи;
3. Графическое описание алгоритма решения задачи (например: в виде блок – схемы);

4. Результаты:

- вставить скриншот интерфейса программы;
- рассмотреть работу программы на множестве из 10 точек в двумерном пространстве (изобразить точки на координатной плоскости);
- привести результаты разбиения точек на классы;
- рассмотреть различные варианты задания радиуса (максимального, среднего и минимального) и заданного пользователем. Привести результаты и сравнить их.

5. Фрагменты кода.

Задания для самостоятельной работы:

Предусмотрены вариантом обучающегося.

Рекомендации по выполнению заданий и подготовке к практическому занятию

Ознакомиться с теоретическим материалом, представленным в четвёртом разделе данной дисциплины.

Основная литература

1. Рутковский, Л. Методы и технологии искусственного интеллекта : учебник / Л. Рутковский; Пер. с пол. - М. : Горячая линия- Телеком, 2010. - 520 с.

Дополнительная литература

2. Кедрин, В. С. Искусственные нейронные сети : учебное пособие / В. С. Кедрин, Ю. А. Шичкина. - Братск : БрГУ, 2012. - 107 с.

Практическое занятие №4

Метод масок

Цель работы:

Изучить логический метод распознавания – метод «Масок», определить его особенности и отличия от изученных выше методов.

Задание:

1. Согласно номеру варианта представить предложенный набор букв в качестве двоичных векторов. Для этого каждую букву нарисовать в матрице (размер матрицы должен быть одинаков для всех букв). Например: буква «Т» в матрице размера 5x5 будет выглядеть следующим образом:

| | | | | |
|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 1 |
| 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 |

Двоичный вектор будет выглядеть как:
1111100100001000010000100.

2. Записать вектора в файл;

3. Составить программу, которая будет выполнять следующие функции:

- считать двоичные вектора из файла;

- ввод порога схожести и порога близости;
- разделить полученный набор образов (двоичных векторов) на классы, используя метод «масок»;
- вывести следующую информацию на экран или в файл: сколько групп и какие образы попали в группу, до объединения групп; какие группы были объединены; сколько получилось классов, какие образы попали в какой класс;
- после этого взять новый образ и определить, к какому классу он принадлежит;
- предусмотреть возможность изменения нового образа.

Варианты заданий

| Вариант | Набор образов | Новый образ |
|---------|------------------------------|-------------|
| 1. | A, B, Г, Д, О, С, Л, Е, И, З | Й |
| 2. | 1, 3, 6, 5, 0, 2, 4, 7, 9, 8 | В |
| 3. | Ц, П, Р, Н, Е, К, У, В, Х, Ь | Ф |
| 4. | I, J, K, L, D, P, O, T, E, V | Х |
| 5. | H, L, Z, A, U, K, R, B, C, N | D |
| 6. | Ы, Ш, М, W, B, N, Ж, К, S, I | A |
| 7. | A, Й, Г, Д, О, С, Л, Е, И, З | U |
| 8. | L, П, Р, Н, Е, К, У, В, X, F | Б |
| 9. | 1, 3, 6, 5, I, 2, 4, 7, 9, 0 | 8 |
| 10. | !, T, L, O, (, J, H, P, N, П | [|
| 11. | Б, В, Д, Е, А, Ц, Н, С, Ъ, Я | О |
| 12. | X, J, K, L, D, P, O, T, E, V | В |
| 13. | З, В, 6, 8, Р, Э, 5, С, 0, Б | Ь |
| 14. | Н, Й, Г, Д, О, С, Л, Е, И, З | К |
| 15. | H, J, Z, A, U, K, R, B, Y, N | P |

Порядок выполнения:

Изучить теоретические данные. Выполнить пункты задания 1 - 3.

Форма отчетности:

Отчет сдается в печатном виде. В отчете должны присутствовать:

1. Титульный лист;
2. Постановка задачи;
3. Графическое описание алгоритма решения задачи (например: в виде блок – схемы);
4. Результаты:
 - указать номер варианта, набор образов и новый образ;
 - привести матрицы кодирования и двоичный вектор для каждого образа;
 - вставить скриншот интерфейса программы;
 - рассмотреть работу программы для вашего варианта, подобрать наиболее удачные значения порога схожести и близости;
 - привести результаты разбиения на группы, классы.
5. Фрагменты кода.

Задания для самостоятельной работы:

Предусмотрены вариантом обучающегося.

Рекомендации по выполнению заданий и подготовке к практическому занятию

Ознакомиться с теоретическим материалом, представленным в пятом разделе данной дисциплины.

Основная литература

1. Рутковский, Л. Методы и технологии искусственного интеллекта : учебник / Л. Рутковский; Пер. с пол. - М. : Горячая линия- Телеком, 2010. - 520 с.

Дополнительная литература

2. Кедрин, В. С. Искусственные нейронные сети : учебное пособие / В. С. Кедрин, Ю. А. Шичкина. - Братск : БрГУ, 2012. - 107 с.

Практическое занятие №5

Моделирование и обучение нейронной сети типа перцептрон

Занятие проводится в интерактивной форме с разбором конкретных ситуаций (4 час.)

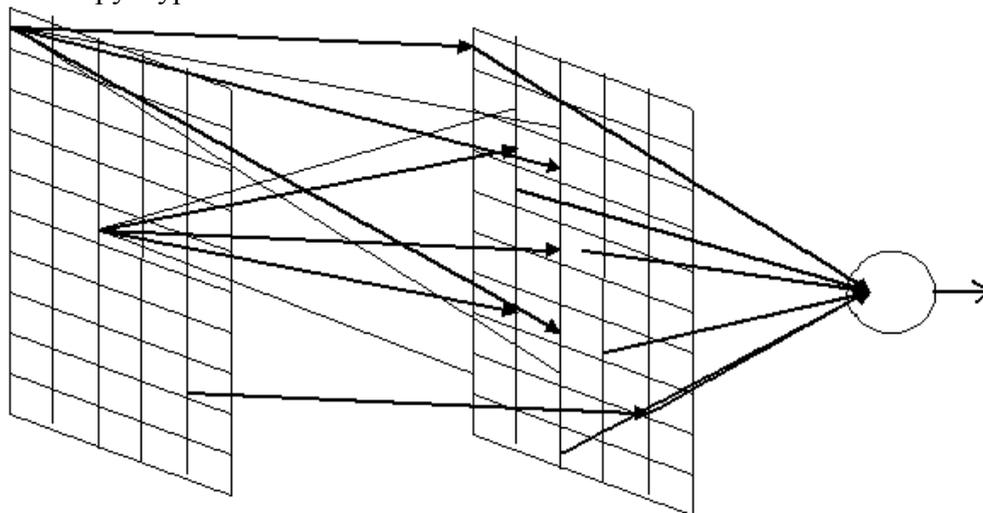
Цель работы:

Изучить модель и принцип функционирования нейронной сети типа перцептрон.

Задание:

Необходимо разработать программу, имитирующую работу нейронной сети, типа перцептрон, который должен распознавать образы.

Структура сети:



Первая матрица размера $n \times n$ представляет собой матрицу рецепторов. Вторая матрица размера $n \times n$ – это матрица нейронов. От каждого рецептора выходит v возбуждающих связей и t тормозных связей (в какой нейрон они будут входить определяется случайным образом). Каждый нейрон имеет одну выходную связь, все эти связи приходят на сумматор.

Входные данные:

- размер матриц n (матрицы квадратные, одинакового размера, максимальный размер = 10);
- порог, одинаковый для всех нейронов матрицы – p ;
- количество тормозных t и возбуждающих v связей, исходящих из каждого элемента рецепторной матрицы (количество связей каждого типа не более 10);
- веса возбуждающих (α), тормозных (β) и выходных (γ) связей;
- приращение весов связей ($\Delta\alpha$, $\Delta\beta$, $\Delta\gamma$);
- количество шагов обучения - h .

Направление связей задается один раз от рецепторов до нейронов и сохраняется на протяжении всей работы программы.

Порядок выполнения:

Изучить теоретические данные. Реализуем программно работу сети согласно следующему алгоритму:

1. На матрице рецепторов задается раздражение. В качестве раздражения подается образ представленной в виде матрицы, состоящей из 0 и 1. Например: буква «Г» в матрице размера 5×5 будет выглядеть следующим образом:

| | | | | |
|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 1 |
| 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 |

2. Все образы должны быть закодированы в матрицу одинакового размера;

3. Набор образов выбираете в соответствии с вашим вариантом;

4. Если рецептор возбужден (то есть на него подается 1), то по всем исходящим от него связям поступают сигналы величиной α и β (в зависимости от типа связи);

5. Нейрон суммирует все входящие на него сигналы по заданным связям. Если получается значение, превышающее значение порога p , то на выходе нейрон выдает сигнал весом γ ;

6. На сумматоре складываются все значения, и выдается окончательный результат.

Прежде, чем использовать нейронную сеть (по алгоритму описанному выше) ее необходимо обучить. Обучение НС происходит по правилу Хэбба, которое заключается в увеличение весов связей (входящих и выходящих) активированного нейрона (нейрон, который выдает сигнал). Нарращивание весов происходит по следующей формуле:

$$X_{i+1} = X_i + \alpha,$$

где X_{i+1} - новый вес связи, X_i - вес связи, α - приращение веса.

Разрабатываемая вами программа должна работать в двух режимах: режим обучения и режим работы.

Для обучения необходимо закодировать набор образов в соответствии с вариантом. Каждый образ подается на рецепторную матрицу h раз. Для каждого образа обучение выполняется по правилу Хэбба.

После обучения нейронная сеть используется уже для работы по следующему алгоритму:

1. На рецепторную матрицу по очереди подается каждый образ из набора (только уже без обучения, а по алгоритму работы сети) на выходе для каждого образа получается числовое значение;

2. После этого на рецепторную матрицу подается новый образ. Вычисляется для него числовое значение. В зависимости от того, к какому числовому значению (полученных на первом шаге) оно будет ближе, принимается решение на какой из образов из набора похож новый образ. Если полученное значение сильно отличается от полученных на первом шаге, то говорят, о том, что этому образу нейронная сеть не обучена.

| Вариант | Набор образов | Новый образ |
|---------|------------------|-------------|
| 1. | В, Д, О, Л, И, З | Й |
| 2. | 3, 5, 0, 9, 8 | В |
| 3. | Ц, П, Н, Е, В | Ф |
| 4. | И, J, L, Т, Е | Х |
| 5. | Н, U, R, В, N | D |
| 6. | М, W, В, N, Ж | А |
| 7. | А, Г, С, Е, И | U |
| 8. | L, K, Y, X, F | Б |
| 9. | 3, 6, 5, 9, 0 | 8 |
| 10. | Т, О, Л, N, П | U |
| 11. | Б, В, С, Ъ, Я | О |
| 12. | L, D, P, O, E | В |
| 13. | В, 6, 8, 5, Б | Ь |
| 14. | Н, Й, С, Л, И | К |
| 15. | Н, U, K, R, В | Р |

Форма отчетности:

Отчет сдается в печатном виде. В отчете должны присутствовать:

1. Титульный лист;
2. Постановка задачи;
3. Графическое описание алгоритма решения задачи (например: в виде блок – схемы);
4. Результаты:
 - указать номер варианта, набор образов и новый образ;
 - привести матрицы кодирования и двоичный вектор для каждого образа;
 - вставить скриншот интерфейса программы;

- рассмотреть работу программы для вашего варианта, подобрать наиболее удачные значения порога схожести и близости;

- привести результаты разбиения на группы, классы.

5. Фрагменты кода.

Задания для самостоятельной работы:

Предусмотрены вариантом обучающегося.

Рекомендации по выполнению заданий и подготовке к практическому занятию

Ознакомиться с теоретическим материалом, представленным в седьмом и восьмом разделах данной дисциплины.

Основная литература

1. Рутковский, Л. Методы и технологии искусственного интеллекта : учебник / Л. Рутковский; Пер. с пол. - М. : Горячая линия- Телеком, 2010. - 520 с.

Дополнительная литература

2. Кедрин, В. С. Искусственные нейронные сети : учебное пособие / В. С. Кедрин, Ю. А. Шичкина. - Братск : БрГУ, 2012. - 107 с.

9.2. Методические указания по выполнению контрольной работы

Программа ESWin 2.0 (в дальнейшем ESWin) разработана фирмой ИНСИКОМ Новосибирского технического университета. Она предназначена для создания продукционно-фреймовых экспертных систем в различных предметных областях и работы с ними. Разберём алгоритм работы в оболочке ESWin на примере выполнения двух заданий.

Задание 1. Разработать базу знаний экспертной системы, советующей по вопросу предоставления кредита.

В нашем случае база знаний должна включать: фрейм-образ кредитополучателя, фрейм цели – предоставление кредита, правила логического вывода заключения о предоставлении кредита.

При принятии решении о предоставлении кредита будем принимать во внимание следующие факторы:

– среднемесячный доход кредитополучателя;

– возраст кредитополучателя;

– срок кредита;

– количество иждивенцев;

– источник дохода, принимающий одно из значений: заработная плата, предпринимательская деятельность, социальное пособие.

Указанные факторы составляют образ кредитополучателя – фрейм «Кредитополучатель». Слотам этого фрейма присвоим имена как в табл. 1.

Таблица 1 - Имена слотов фрейма «Кредитополучатель»

| Название слота | Имя слота |
|--|-----------|
| Среднемесячный доход кредитополучателя | Доход |
| Возраст кредитополучателя | Возраст |
| Срок кредита | Срок |
| Количество иждивенцев | Иждивенцы |
| Источник дохода | Источник |

Для принятия решения о предоставлении кредита будем использовать следующие правила.

Правило 1.

Если *доход менее 100 тыс. р.*, то кредит не предоставляется с уверенностью 100%.

Правило 2.

Если *возраст кредитополучателя более 55 лет*, то кредит не предоставляется с уверенностью 90%.

Правило 3.

Если *возраст кредитополучателя более 50 лет и срок кредита более 5 лет*, то кредит не предоставляется с уверенностью 90%.

Правило 4.

Если количество *иждивенцев у кредитополучателя более двух и срок кредита более 5 лет*, то кредит не предоставляется с уверенностью 70%.

Правило 5.

Если количество *иждивенцев у кредитополучателя более двух и срок кредита менее 6 лет*, то кредит предоставляется с уверенностью 60%.

Правило 6.

Если *источником дохода является заработная плата*, то кредит предоставляется с уверенностью 100%.

Правило 7.

Если *источником дохода является предпринимательская деятельность*, то кредит предоставляется с уверенностью 90%.

Правило 8.

Если *источником дохода является социальное пособие*, то кредит предоставляется с уверенностью 70%.

Примечание. При задании условий в правилах могут использоваться только четыре операции сравнения: =, >, <, <>.

Рекомендации к выполнению задания

1. Запустите стандартное приложение Блокнот.
2. Наберите текст базы знаний в соответствии с требованиями программы ESWin, как это показано в табл. 2.

Таблица 2 - Текст базы знаний «Предоставление кредита»

| Текст | Комментарии |
|--|---|
| TITLE=предоставление кредита | Ключевые слова для формирования заголовка |
| COMPANY=БГЭУ | базы знаний |
| Frame=Кредитополучатель | Начало фрейма «Кредитополучатель» |
| Parent: | Ключевое слово |
| Доход(численный)[размер дохода] | Слот 1 |
| Возраст (численный) | Слот 2 |
| Срок (численный) [срок кредита] | Слот 3 |
| Иждивенцы (численный)[количество] | Слот 4 |
| Источник: (заработная плата; предпринимательская деятельность; социальное пособие) | Слот 5. В скобках через «;» перечисляются допустимые значения, которые будут выбираться из списка |
| EndF | Конец фрейма «Кредитополучатель» |
| Frame=Цель | Начало фрейма «Цель» |
| Parent: | Ключевое слово |
| Предоставление кредита() | Формулировка цели задачи, решаемой экспертной системой |
| EndF | Конец фрейма «Цель» |
| Rule 1 | Начало правила 1 |
| < (Кредитополучатель.Доход; 100) | Проверка условия Доход < 100 |
| Do | Начало заключения |
| = (Цель.Предоставление кредита;Нет) 100 | Запись заключения, выдаваемого экспертной системой на экран в виде текста |

| | |
|---|---|
| | «цель.предоставление кредита = нет с уверенностью 100 %» |
| EndR | Конец правила 1 |
| Rule 2 | Начало правила 2 |
| > (Кредитополучатель.Возраст; 55) | Проверка условия Возраст > 55 |
| Do | Начало заключения |
| = (Цель.Предоставление кредита;Нет) 90 | Запись заключения |
| EndR | Конец правила 2 |
| Rule 3 | Начало правила 3 |
| > (Кредитополучатель.Возраст; 50) | Проверка условия Возраст > 50 |
| > (Кредитополучатель.Срок; 5) | Проверка условия Срок > 5 |
| Do | Начало заключения |
| = (Цель.Предоставление кредита;Нет) 90 | Запись заключения |
| EndR | Конец правила 3 |
| Rule 4 | Начало правила 4 |
| > (Кредитополучатель.Иждивенцы; 2) | Проверка условия Иждивенцы > 2 |
| > (Кредитополучатель.Срок; 5) | Проверка условия Срок > 5 |
| Do | Начало заключения |
| = (Цель.Предоставление кредита;Нет) 70 | Запись заключения |
| EndR | Конец правила 4 |
| Rule 5 | Начало правила 5 |
| > (Кредитополучатель.Иждивенцы; 2) | Проверка условия Иждивенцы >2 |
| < (Кредитополучатель.Срок; 6) | Проверка условия Срок < 6 |
| Do | Начало заключения |
| = (Цель.Предоставление кредита;Да) 60 | Запись заключения |
| EndR | Конец правила 5 |
| Rule 6 | Начало правила 6 |
| = (Кредитополучатель.Источник; Заработная плата) | Проверка условия Источник= Заработная плата |
| Do | Начало заключения |
| = (Цель.Предоставление кредита;Да) 100 | Запись заключения |
| EndR | Конец правила 6 |
| Rule 7 | Начало правила 7 |
| = (Кредитополучатель. Источник; предпринимательская деятельность) | Проверка условия Источник = предпринимательская деятельность |
| Do | Начало заключения |
| = (Цель.Предоставление кредита;Да) 90 | Запись заключения |
| EndR | Конец правила 7 |
| Rule 8 | Начало правила 8 |
| = (Кредитополучатель. Источник; социальное пособие) | Проверка условия Источник = социальное пособие |
| Do | Начало заключения |
| = (Цель.Предоставление кредита;Да) 80 | Запись заключения |
| EndR | Конец правила 8 |

3. Сохраните набранный текст базы знаний в файле с именем **Кредит.KLB** (указание расширения .KLB обязательно) в своей папке.

Созданная база знаний является ядром экспертной системы «Кредитополучатель».

Задание 2. Использовать экспертную систему «Кредитополучатель» для получения совета по вопросу предоставления кредита физическому лицу, у которого ежемесячный доход равен 90 тыс. р.

Рекомендации к выполнению задания

1. Запустите программу ESWin. Появится окно этой программы как на рис. 1.

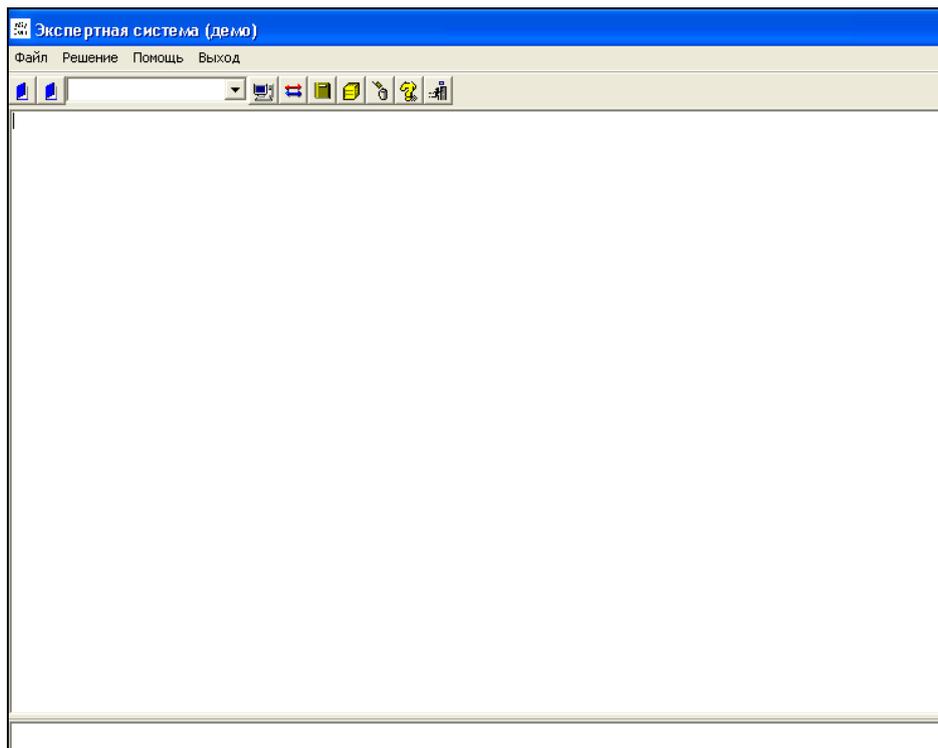


Рис. 1. Окно программы ESWin

2. Откройте созданную базу знаний (сохраненную в файле **Кредит.KLB**) по команде **Файл/Открыть базу знаний**;

3. Осуществите поиск решения по команде **Решение/Поиск решения**. Появится окно как на рис. 2.

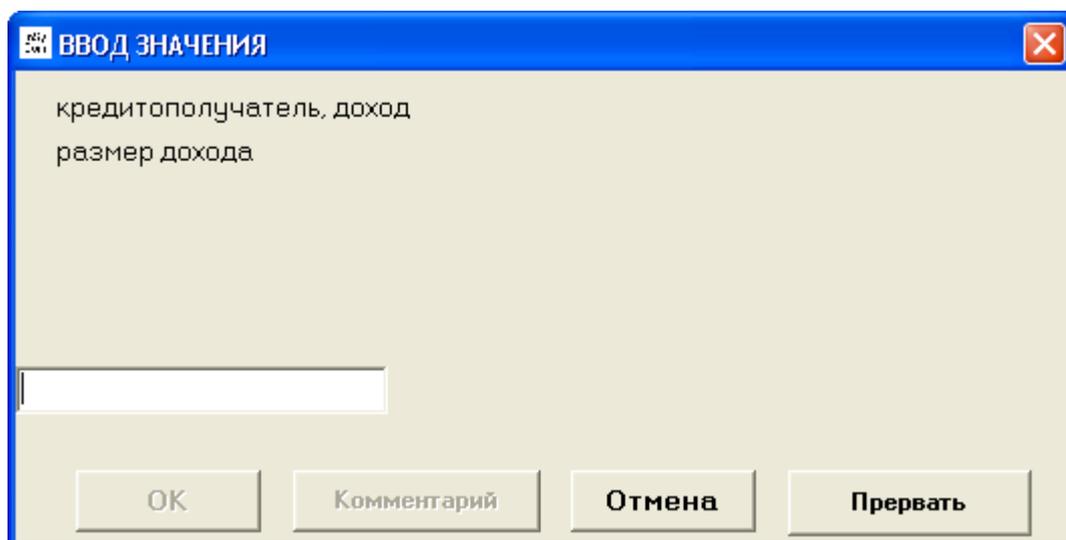


Рис. 2. Диалоговое окно для ввода значения слота Доход

4. Введите число 90 – значение слота Доход. Появится окно как на рис. 3. В верхней части окна отображается текст базы знаний, а в нижней части – заключение экспертной системы о непредоставлении кредита с уверенностью 100%.

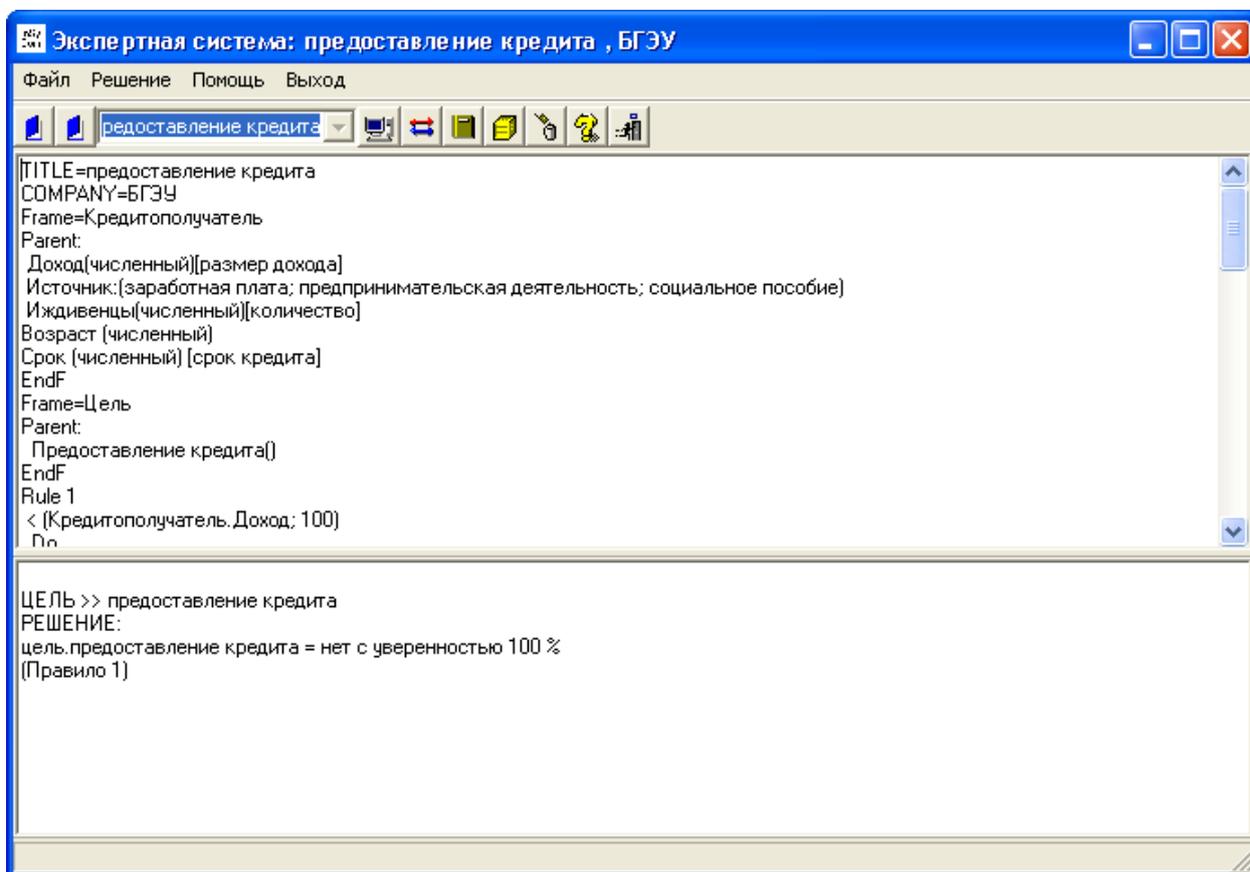


Рис. 3. Окно экспертной системы «Кредитополучатель» с заключением

5. Произведите очистку базы данных (удаление значений слотов фрейма «Кредитополучатель») по команде Решение/Очистка базы данных.

Задания для самостоятельной работы. Использовать экспертную систему «Кредитополучатель» для получения совета по вопросу предоставления кредита физическому лицу, у которого:

- ежемесячный доход равен 200 тыс. р. и возраст 60 лет;
- ежемесячный доход равен 300 тыс. р., возраст 53 года и срок кредита 6 лет;
- ежемесячный доход равен 456 тыс. р., возраст 41 год, иждивенцев 3 человека и срок кредита 7 лет;
- ежемесячный доход равен 232 тыс. р., возраст 35 лет, срок кредита 4 года и иждивенцев – 1;
- ежемесячный доход равен 312 тыс. р., возраст 32 года, иждивенцев 2 человека и источник дохода – заработная плата;
- ежемесячный доход равен 242 тыс. р., возраст 27 лет, иждивенцев 1 человек и источник дохода – предпринимательская деятельность;
- ежемесячный доход равен 514 тыс. р., возраст 37 лет, иждивенцев нет (0 человек) и источник дохода – социальное пособие.

Рекомендации к выполнению заданий

Выполнение каждого задания начинайте с ввода команд:

- **Решение/Очистка базы данных;**
- **Решение/Поиск решения.**

Затем в появляющихся на экране окнах **Ввод значения** вводите запрашиваемые значения слотов фрейма «Кредитополучатель», указанные непосредственно в задании.

Анализируйте выводимое экспертной системой заключение. Обращайте внимание на то, какое правило «сработало» при выводе заключения и убеждайтесь в соответствии текста заключения его описанию в данном правиле базы знаний.

По завершении выполнения всех заданий просмотрите отчет о процессе поиска решений экспертной системой по команде

Основная литература

1. Рак, И.П. Интеллектуальные информационные системы: методические указания / сост.: И.П. Рак, А.В. Селезнёв, Э.В. Сысоев. - Тамбов: Издательство ТГТУ, 2010. - 16 с. <http://window.edu.ru/resource/187/73187>

Дополнительная литература

2. Иванов, М. Ю. Основы логического программирования : методические указания / М. Ю. Иванов. - Братск : БрГУ, 2009. - 27 с.

10. ПЕРЕЧЕНЬ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, ИСПОЛЬЗУЕМЫХ ПРИ ОСУЩЕСТВЛЕНИИ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА ПО ДИСЦИПЛИНЕ

1. ОС Windows 7 Professional.
2. Microsoft Office 2007 Russian Academic OPEN No Level.
3. Антивирусное программное обеспечение Kaspersky Security.

11. ОПИСАНИЕ МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЙ БАЗЫ, НЕОБХОДИМОЙ ДЛЯ ОСУЩЕСТВЛЕНИЯ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА ПО ДИСЦИПЛИНЕ

| <i>Вид занятия</i> | <i>Наименование аудитории</i> | <i>Перечень основного оборудования</i> | <i>№ ПЗ</i> |
|--------------------|-------------------------------|---|-------------|
| 1 | 3 | 4 | 5 |
| ПЗ | Дисплейный класс | AMD Athlon 64 (5GHz/250Gb/2Gb/DD-RW), 2 ядра | ПЗ 1-5 |
| СР | ЧЗ №3 | Оборудование 15- CPU 5000/RAM 2Gb/HDD (Монитор TFT 19 LG 1953S-SF); принтер HP LaserJet P3005 | - |

**ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ДЛЯ ПРОВЕДЕНИЯ
ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ОБУЧАЮЩИХСЯ ПО ДИСЦИПЛИНЕ**

1. Описание фонда оценочных средств (паспорт)

| № компетенции | Элемент компетенции | Раздел | Тема | ФОС |
|---------------|---|--|---|---------------------------|
| ПК-11 | Умение проводить технико-экономическое обоснование проектных расчетов с использованием современных подходов и методов | 1. Состояние и перспективы развития теории и практики интеллектуальных систем управления инфокоммуникациями (ИСУИК) | 1.1. Понятие интеллектуальности | Вопрос к зачёту 1.1 |
| | | | 1.2. Предмет ИС | Вопросы к зачёту 1.2, 1.3 |
| | | 2. Структура интеллектуально-адаптивных систем | 2.1. Структура интеллектуально-адаптивных систем | Вопрос к зачёту 2.1 |
| | | 3. Методы построения алгоритмов адаптации | 3.1. Виды моделей | Вопрос к зачёту 3.1 |
| | | | 3.2. Методы (алгоритмы) интеллектуальных вычислений | Вопрос к зачёту 3.1 |
| | | 4. Системы прямого управления | 4.1. Понятие образа | Вопрос к зачёту 4.1 |
| | | | 4.2. Проблема обучения распознаванию образов (ОРО) | Вопрос к зачёту 4.2 |
| | | | 4.3. Геометрический и структурный подходы | Вопрос к зачёту 4.3 |
| | | | 4.4. Систематика методов распознавания | Вопрос к зачёту 4.4 |
| | | | 4.5. Аналитические методы | Вопросы к зачёту 4.5-4.9 |
| | | | 5. Системы непрямого управления | 5.1. Биологический нейрон |
| | | 5.2. Формальный нейрон | | Вопрос к зачёту 5.2 |
| | | 5.3. Виды функций активации | | Вопрос к зачёту 5.3 |
| | | 5.4. Искусственные нейронные сети | | Вопрос к зачёту 5.4 |
| | | 5.5. Архитектура нейронной сети | | Вопрос к зачёту 5.5 |
| | | 5.6. Функционирование нейросети | | Вопрос к зачёту 5.6 |
| | | 5.7. Общая схема обучения нейронной сети | | Вопрос к зачёту 5.7 |
| | | 5.8. Классификация нейронных сетей | | Вопросы к зачёту 5.8-5.11 |
| | | 6. Представление знаний в интеллектуальных системах | 6.1. Представление знаний в интеллектуальных системах | Вопрос к зачёту 6.1 |

| | | | | |
|--|--|--|---|---------------------|
| | | 7. Нейронные сети в интеллектуально-адаптивных системах | 7.1. Метод роящихся частиц | Вопрос к зачёту 7.1 |
| | | | 7.2. Генетические алгоритмы | Вопрос к зачёту 7.2 |
| | | | 7.3. Моделирование отжига (Simulated Annealing) | Вопрос к зачёту 7.3 |
| | | 8. Понятие экспертной системы (ЭС). Структура ЭС. Классификации ЭС | 8.1. Экспертные системы: базовые понятия | Вопрос к зачёту 8.1 |
| | | | 8.2. Классификация экспертных систем | Вопрос к зачёту 8.2 |
| | | | 8.3. Составные части экспертной системы и порядок ее функционирования | Вопрос к зачёту 8.3 |
| | | | 8.4. Функционирование базы знаний экспертной системы | Вопрос к зачёту 8.4 |
| | | | 8.5. Обратный метод логического дедуктивного вывода | Вопрос к зачёту 8.5 |
| | | | 8.6. Прямой метод логического дедуктивного вывода | Вопрос к зачёту 8.6 |

2. Вопросы к зачёту

| № п/п | Компетенции | | ВОПРОСЫ К ЗАЧЁТУ | № и наименование раздела |
|--------------------------------------|-------------|---|---|---|
| | Код | Определение | | |
| 1 | 2 | 3 | 4 | 5 |
| 1. | ПК-11 | Умение проводить технико-экономические обоснование проектных расчётов с использованием современных подходов и методов | 1.1. Понятие интеллектуальности | 1. Состояние и перспективы развития теории и практики интеллектуальных систем управления инфокоммуникациями (ИСУИК) |
| | | | 1.2. Предмет интеллектуальные системы | |
| | | | 1.3. Структура интеллектуальных систем. | |
| | | | 2.1. Структура интеллектуально-адаптивных систем | 2. Структура интеллектуально-адаптивных систем |
| | | | 3.1. Типы моделей интеллектуальных вычислений | 3. Методы построения алгоритмов адаптации |
| | | | 3.2. Методы интеллектуальный вычислений | |
| | | | 4.1. Понятие образа | 4. Системы прямого управления |
| | | | 4.2. Проблема обучения распознавания образов | |
| | | | 4.3. Геометрические и структурные подходы | |
| | | | 4.4. Систематика методов распознавания | |
| | | | 4.5. Аналитические методы | |
| | | | 4.6. Метод построения разделяющих гиперплоскостей | |
| | | | 4.7. Метод «Форель» | |
| 4.8. Метод дробящихся эталонов | | | | |
| 4.9. Метод к-внутригрупповых средних | | | | |

| | | | | |
|--|--|--|--|---|
| | | | 5.1. Биологический нейрон 5.2. Формальный нейрон 5.3. Виды функций активации 5.4. Искусственные нейронные сети 5.5. Архитектура нейронной сети 5.6. Функционирование нейронной сети 5.7. Общая схема обучения нейронной сети 5.8. Однослойный перцептрон 5.9. Многослойный перцептрон 5.10. Сеть Кохонена 5.11. Сеть Хопфилда | 5. Системы непрямого управления |
| | | | 6.1. Представление знаний в интеллектуальных системах | 6. Представление знаний в интеллектуальных системах |
| | | | 7.1. Метод роящихся частиц 7.3. Вырожденность передаточной функции. Условие разрешимости. 7.2. Генетические алгоритмы 7.3. Моделирование отжига | 7. Нейронные сети в интеллектуально-адаптивных системах автоматического управления |
| | | | 8.1. Экспертная система : базовые понятия 8.2. Классификация экспертных систем 8.3. Составные части экспертной системы и порядок её функционирования 8.4. Функционирование базы знаний экспертной системы 8.5. Обратный метод логического дедуктивного вывода 8.6. Прямой метод логического дедуктивного вывода | 8. Понятие экспертной системы (ЭС). Структура ЭС. Классификации ЭС |

3. Описание показателей и критериев оценивания компетенций

| Показатели | Оценка | Критерии |
|--|----------------|---|
| Знать (ПК-11): - основные принципы применения инфокоммуникационных технологий и требования информационной безопасности; Уметь (ПК-11): - проводить технико-экономическое обоснование проектных расчетов с использованием современных подходов и методов; Владеть (ПК-11): - навыками работы с инструментальным программным обеспечением для построения интел- | зачтёно | Обучающийся должен во время ответа показать знания: общих структур интеллектуально-адаптивных систем управления, особенностей структур нейронных сетей, общей концепции теории адаптивного управления, основных терминов, используемых в научно-технической литературе по теории интеллектуальных систем управления. Обучающийся должен иметь навыки владения: разработки на языке высокого уровня программ, имитирующих нейронные сети, понимания материала и способности высказывания мыслей на научно-техническом языке. Обучающийся во время ответа должен продемонстрировать умения: решать стандартные задачи профессиональной деятельности на основе информационной и библиографической культуры, синтезировать систему адаптивного управления для линейного инфокоммуникационного |

| | | |
|---|-------------------|---|
| лектуальных систем в инфокоммуникациях. | | объекта, строить алгоритмы адаптивного управления нелинейным инфокоммуникационным объектом. |
| | не зачтено | На оба вопроса обучающийся отвечает неубедительно. На дополнительные вопросы преподавателя также не может ответить. |

4. Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и опыта деятельности

Дисциплина «Интеллектуальные системы управления инфокоммуникациями» направлена на ознакомление обучающихся с базовыми принципами математического моделирования интеллектуальных систем управления инфокоммуникациями для обеспечения адаптивного управления, умение использовать полученные знания и навыки при проектировании интеллектуальных систем управления связи и коммуникации.

Изучение дисциплины предусматривает:

- лекции;
- практические занятия;
- контрольную работу;
- самостоятельную работу;
- зачёт.

В ходе освоения раздела 1 «Состояние и перспективы развития теории и практики интеллектуальных систем управления инфокоммуникациями» обучающиеся должны уяснить: понятие интеллекта, интеллектуальной системы.

В ходе освоения раздела 2 «Структура интеллектуально-адаптивных систем» обучающиеся должны знать структуру интеллектуальной системы.

В ходе освоения раздела 3 «Методы построения алгоритмов адаптации» обучающиеся должны уяснить: типы моделей и методы интеллектуальных вычислений.

В ходе освоения раздела 4 «Системы прямого управления» обучающиеся должны знать: понятие образа, проблему обучения распознавания образов, геометрические и аналитические методы распознавания образов.

В ходе освоения раздела 5 «Системы непрямого управления» обучающиеся должны уяснить: биологический и формальный нейроны, архитектуру, функционирование, общую схему обучения и классификацию нейронных сетей.

В ходе освоения раздела 6 «Представление знаний в интеллектуальных системах» обучающиеся должны знать общие подходы к представлению знаний в интеллектуальных системах.

В ходе освоения раздела 7 «Нейронные сети в интеллектуально-адаптивных системах автоматического управления» обучающиеся должны обратить внимание на три метода нейронного программирования.

В ходе освоения раздела 8 «Понятие экспертной системы (ЭС). Структура ЭС. Классификации ЭС» обучающиеся должны знать: базовые понятия, классификацию, составные части и порядок функционирования экспертной системы и базы знаний, два метода логического дедуктивного вывода.

На практических занятиях обучающиеся приобретают навыки моделирования и расчёта нейронных сетей.

В выполнении контрольной работы происходит закрепление на практике теоретических принципов проектирования экспертной системы в среде ESWin 2.0.

При подготовке к зачёту рекомендуется особое внимание уделить следующим вопросам: структура интеллектуальной системы, геометрические и аналитические методы распознавания образов, нейронные сети, экспертные системы.

Работа с литературой является важнейшим элементом в получении знаний по дисциплине. Прежде всего, необходимо воспользоваться списком рекомендуемой литературы. Дополнительные сведения по изучаемым темам можно найти в периодической печати и Интернете.

Предусмотрено проведение аудиторных занятий в интерактивной форме (лекции с текущим контролем, практические занятия с разбором конкретных ситуаций) в сочетании с внеаудиторной работой.

АННОТАЦИЯ
рабочей программы дисциплины
Интеллектуальные системы управления инфокоммуникациями

1. Цель и задачи дисциплины

Целью изучения дисциплины является изложение базовых характеристик основных направлений развития систем искусственного интеллекта, моделей и методов представления знаний; умение использовать полученные знания и навыки при проектировании интеллектуальных систем управления инфокоммуникациями.

Задачей дисциплины является подготовка обучающихся к самостоятельной работе по решению практических задач, связанных с применением методов и средств искусственного интеллекта для управления инфокоммуникациями, построением экспертных систем инфокоммуникационных сетей с использованием оболочки представления знаний ESWin, моделированием нейронной сети инфокоммуникационного объекта.

2. Структура дисциплины

2.1 Распределение трудоемкости по отдельным видам учебных занятий, включая самостоятельную работу: Лк – 24 час.; ПЗ – 24 час.; СР – 60 час.

Общая трудоемкость дисциплины составляет 108 час, 3 зачётные единицы.

2.2 Основные разделы дисциплины:

1. Состояние и перспективы развития теории и практики интеллектуальных систем управления инфокоммуникациями;
2. Структура интеллектуально-адаптивных систем;
3. Методы построения алгоритмов адаптации;
4. Системы прямого управления;
5. Системы непрямого управления;
6. Представление знаний в интеллектуальных системах;
7. Нейронные сети в интеллектуально-адаптивных системах автоматического управления;
8. Понятие экспертной системы (ЭС). Структура ЭС. Классификации ЭС.

3. Планируемые результаты обучения (перечень компетенций)

Процесс изучения дисциплины направлен на формирование следующей компетенции:
ПК-11 – умением проводить технико-экономическое обоснование проектных расчетов с использованием современных подходов и методов.

4. Вид промежуточной аттестации: зачёт.

*Протокол о дополнениях и изменениях в рабочей программе
на 201__-201__ учебный год*

1. В рабочую программу по дисциплине вносятся следующие дополнения:

2. В рабочую программу по дисциплине вносятся следующие изменения:

Протокол заседания кафедры № _____ от «___» _____ 201__ г.,
(разработчик)

Заведующий кафедрой _____
(подпись)

(Ф.И.О.)

Программа составлена в соответствии с федеральным государственным образовательным стандартом высшего образования по направлению подготовки 11.03.02 Инфокоммуникационные технологии и системы связи от «06» марта 2015 г. № 174

для набора 2015 года: и учебным планом ФГБОУ ВО «БрГУ» для очной формы обучения от «13» июля 2015 г. № 475.

для набора 2016 года: и учебным планом ФГБОУ ВО «БрГУ» для очной формы обучения от «06» июня 2016г. №429.

для набора 2017 года: и учебным планом ФГБОУ ВО «БрГУ» для очной формы обучения от «06» марта 2017 г. № 125.

Программу составил:

Крумин О.К., доцент кафедры УТС, к.т.н. _____

Рабочая программа рассмотрена и утверждена на заседании кафедры УТС
от 28 декабря 2018 г, протокол № 6

Заведующий кафедрой УТС _____ Игнатьев И.В.

СОГЛАСОВАНО:

Заведующий выпускающей кафедрой _____ Игнатьев И.В.

Директор библиотеки _____ Сотник Т.Ф.

Рабочая программа одобрена методической комиссией ФЭиА факультета
от 28 декабря 2018 г, протокол № 5

Председатель методической комиссии факультета _____ Ульянов А.Д.

СОГЛАСОВАНО:

Начальник
учебно-методического управления _____ Нежевец Г.П.

Регистрационный № _____