

ФЕДЕРАЛЬНОЕ АГЕНТСТВО ПО ОБРАЗОВАНИЮ
Государственное образовательное учреждение высшего профессионального
образования
УЛЬЯНОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ФАКУЛЬТЕТ МАТЕМАТИКИ И ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ
Кафедра телекоммуникационных технологий и сетей

С.В. Липатова

**Сборник задач по курсу
«Интеллектуальные информационные системы»**

Учебное пособие



**Ульяновск
2010**

УДК 004.8

*Печатается по решению Ученого совета факультета математики, физики
и информационных технологий*

Ульяновского государственного университета

Рецензенты:

профессор кафедры информационных технологий УлГУ, д.т.н., профессор

К.В. Кумунжиев,

зав. кафедрой автоматизации обработки информации и математики УВВТУ (военный институт), к.т.н., доцент, **В.Г.Шубович**

Сборник задач по курсу «Интеллектуальные информационные системы» учебное пособие / С.В. Липатова. – Ульяновск: УлГУ, 2010. - 64 с.

Данное учебное пособие ориентировано на курсы «Системы искусственного интеллекта», «Интеллектуальные информационные системы», «Представление знаний» и «Основы интеллектики». В пособие включены необходимые для решения базовых задач искусственного интеллекта теоретические материалы, варианты задач и описан процесс их решения.

Пособие предназначено для студентов факультетов математики и информационных технологий, иностранных языков и профессиональных коммуникаций.

УДК 004.8

© Ульяновский государственный университет, 2010

© Липатова С.В., 2010

Содержание

Введение	2
Представление знаний	3
Продукционная модель.....	3
<i>Пример решения задачи</i>	4
<i>Задачи</i>	7
Семантическая сеть.....	8
<i>Пример решения задачи</i>	10
<i>Задачи</i>	12
Фреймовая модель	13
<i>Пример решения задачи</i>	16
<i>Задачи</i>	22
Нейронные сети	24
Алгоритм обучения сети по Δ -правилу	26
<i>Пример решения задачи</i>	27
<i>Задачи</i>	29
Алгоритм обратного распространения ошибки	32
<i>Пример решения задачи</i>	34
<i>Задачи</i>	36
Генетический алгоритм	41
<i>Пример решения задачи</i>	43
<i>Задачи</i>	46
Нечеткие множества и нечеткая логика	51
Операции над нечеткими множествами	51
<i>Пример решения задачи</i>	51
<i>Задачи</i>	53
Нечеткий вывод	58
<i>Пример решения задачи</i>	59
<i>Задачи</i>	62
Литература	66

Введение

Сегодня уже ни у кого не возникает сомнений в необходимости и нужности применения технологий искусственного интеллекта на практике. Существует множество аппаратных (нейроплаты, нечеткие контроллеры, роботы и т.д.) и программных реализаций, применяемых в различных предметных областях. Экспертные системы и системы поддержки принятия решений доказали свою полезность. Большое количество пакетов прикладных программ построено на базе технологий искусственного интеллекта. Например, нейронные сети используются в FineReader (программа распознавания текста) и Brain Marker (программа прогнозирования на финансовых рынках). Поддерживают методы генетических алгоритмов и нейронных сетей The AI Trilogy (пакет прикладных программ для финансового анализа) и MahtLab (программа для анализа данных). Нечеткая логика стала основой для CubiCalc (пакет для построения нечетких экспертных систем) и FuziCalc (программа в стиле Excel). Это только некоторые наиболее известные программы и только некоторые из них, можно использовать, не обладая элементарными знаниями в области искусственного интеллекта.

Поэтому специалисты в IT-сфере должны владеть теоретическими и практическими знаниями и навыками использования средств и методов искусственного интеллекта.

Данное учебное пособие адресовано студентам, изучающим дисциплины, связанные с искусственным интеллектом.

В пособии представлены задачи, позволяющие сформировать базовые навыки при изучении методов представления знаний, обучения нейронных сетей, использования генетических алгоритмов и нечетких вычислений.

Рекомендуется совместно с данным задачник использовать учебное пособие «Интеллектуальные информационные системы» [4] и литературу, предлагаемую в нем.

Представление знаний

Существует множество моделей представления знаний. Наиболее распространены три модели представления знаний (таблица 1): фреймовая, продукционная и семантическая. Выбор метода представления знаний зависит от особенностей предметной области (какие структуры знаний наиболее часто встречаются, присутствуют ли иерархичность или сетевые конструкции, характер входных и выходных данных в задачах и т.д.), опыта когнитолога, выбранного инструментария разработки.

Таблица 1. Основные модели представления знаний, используемые на практике

Модель	Достоинства	Недостатки
Продукции	Наглядность, высокая модульность, легкость внесения дополнений и изменений, простота механизма логического вывода, простота интерпретации.	При накоплении большого числа (нескольких сотен) продукций они начинают противоречить друг другу, возникают трудности при добавлении правил, зависящих от уже имеющихся в базе знаний, отсутствует целостный образ знаний, неясна взаимосвязей между правилами.
Семантические сети	Наглядность, соответствует представлениям об организации долговременной памяти человека, позволяет снизить объем хранимых данных.	Представляют собой пассивные структуры, для обработки которых необходим специальный аппарат формального вывода и планирования, произвольная структура и различные типы вершин и связей усложняют процедуру обработки информации, сетевая модель не дает ясного представления о структуре предметной области.
Фреймы	Гибкость, наглядность, удобный способ включения процедурных знаний, сводимость к другим моделям, модульность.	Отсутствие универсальной процедуры управления выводом кроме механизма наследования, является идеологической концепцией.

Продукционная модель

Продукция – это предложение-образец вида «Если, то», по которому осуществляется поиск в базе знаний.

В продукции выделяют левую часть (начинается с «если» и заканчивается перед «то») и правую (начинается после «то»). Левая часть продукции - антецедент – условие выполнения правой часть продукции. Правая часть – консеквент – действие, выполняемое в случае нахождения элементов, удовлетворяющих левой части. Действие может быть промежуточным и выступать затем в качестве консеквента или целевым, завершающим процедуру вывода.

Антецедент формируется из фактов, входных данных задачи и логических связей (и, или, не). Консеквент может представлять из себя действие по изменению фактов, данных, рекомендацию, решение задачи. Кроме этого, любая продукция имеет имя и приоритет, определяющий последовательность проверки продукций машиной вывода.

Продукции отражают причинно-следственные связи, которые и позволяют человеку принимать решения, базируясь на знаниях и предположениях о том, что есть и что будет, если что-то сделать.

Пример решения задачи

Задача. Построить продукционную модель представления знаний в предметной области «Ресторан» (посещение ресторана).

Описание процесса решения. Для построения продукционной модели представления знаний необходимо выполнить следующие шаги:

- 1) Определить целевые действия задачи (являющиеся решениями).
- 2) Определить промежуточные действия или цепочку действий, между начальным состоянием и конечным (между тем, что имеется, и целевым действием).
- 3) Опередить условия для каждого действия, при котором его целесообразно и возможно выполнить. Определить порядок выполнения действий.
- 4) Добавить конкретики при необходимости, исходя из поставленной задачи.
- 5) Преобразовать полученный порядок действий и соответствующие им условия в продукции.
- 6) Для проверки правильности построения продукций записать цепочки продукций, явно проследив связи между ними.

Этот набор шагов предполагает движение при построении продукционной модели от результата к начальному состоянию, но возможно и движение от начального состояния к результату (шаги 1 и 2).

Решение.

- 1) Обязательное действие, выполняемое в ресторанах – поглощение пищи и ее оплата. Значит, есть уже два целевых действия «съесть пищу» и «оплатить», которые взаимосвязаны и следуют друг за другом.
- 2) Прежде чем что-либо съесть в ресторане, туда нужно прийти, дождаться официанта и сделать заказ. Кроме того, нужно выбрать, в какой именно ресторан пойти. Значит, цепочка промежуточных действий: «выбор ресторана и путь туда», «сделать заказ официанту».
- 3) Прежде чем идти в ресторан, необходимо убедиться, что есть необходимая сумма денег. Выбор ресторана может обуславливаться многими причинами, выберем территориальный признак – к какому ближе в тот и идем. В разных ресторанах работают разные люди, поэтому в зависимости от выбора ресторана, официанты будут разные. Кроме того, разные рестораны специализируются на разных кухнях, поэтому заказанные блюда будут в разных ресторанах отличаться. Значит вначале идут действия, позволяющие выбрать ресторан, затем характеризующие рестораны, а уже после заказ, еда, и оплата заказа.
- 4) Пусть в задаче будут рассматриваться два ресторана: «Вкусная еда» и «Вкуснятина». Первый – паб и заказы приносят быстрее, чем во втором, второй – пиццерия. В первом работает официант Сергей, а во втором официантка Марина. Петр – это клиент.
- 5) Выше описанное можно преобразовать в следующие предложения типа «Если, то»:
 - *Если субъект хочет есть и у субъекта есть достаточная сумма денег, то субъект может пойти в ресторан.*
 - *Если субъект ближе к ресторану «Вкусная еда», чем к ресторану «Вкуснятина» и субъект может пойти в ресторан, то субъект идет в ресторан «Вкусная еда».*
 - *Если субъект ближе к ресторану «Вкуснятина», чем к ресторану «Вкусная еда» и субъект может пойти в ресторан, то субъект идет в ресторан «Вкуснятина».*
 - *Если субъект идет в ресторан «Вкуснятина» и в ресторане «Вкуснятина» работает официант Марина, то у субъекта принимает заказ Марина.*
 - *Если субъект идет в ресторан «Вкусная еда» и в ресторане «Вкусная еда» работает официант Сергей, то у субъекта принимает заказ Сергей.*
 - *Если субъект выбрал блюда и у субъекта принимает заказ Марина, то заказ принесут через 20 мин.*

- Если субъект выбрал блюда и у субъекта принимает заказ Сергей, то заказ принесут через 10 мин.
- Если заказ принесут через 20 мин. или заказ принесут через 10 мин., то субъект может есть.
- Если субъект может есть, то после еды субъект должен оплатить заказ.

Введем обозначения для фактов (Ф), действий (Д) и продукций (П), тогда:

Субъект = Петр;

Ф1= субъект хочет есть;

Ф2= у субъекта есть достаточная сумма денег;

Ф3= субъект ближе к ресторану «Вкусная еда», чем к «Вкуснятина»;

Ф4=в ресторане «Вкуснятина» работает официант Марина;

Ф5=в ресторане «Вкусная еда» работает официант Сергей;

Ф6= субъект выбрал блюда;

Д1= субъект может пойти в ресторан;

Д2=субъект идет в ресторан «Вкусная еда»;

Д3=субъект идет в ресторан «Вкуснятина»;

Д4= у субъекта принимает заказ Марина;

Д5=у субъекта принимает заказ Сергей;

Д6=заказ принесут через 20 мин.

Д7=заказ принесут через 10 мин.

Д8=после еды субъект должен оплатить заказ.

Для продукций установим приоритет (в скобках перед запятой, чем выше приоритет, чем раньше проверяется правило).

П1(4 , Ф1 и Ф2)= Д1;

П5(3 , Д2 и Ф5)= Д5;

П2(5 , Ф3 и Д1)= Д2;

П6(2 , Д4)= Д6;

П3(4 , не Ф3 и Д1)= Д3;

П7(2 , Д5)= Д7;

П4(3 , Д3 и Ф4)= Д4;

П8(1 , Д6 или Д7)= Д8;

6) Для отображения взаимосвязи продукций построим граф (рис. 1).

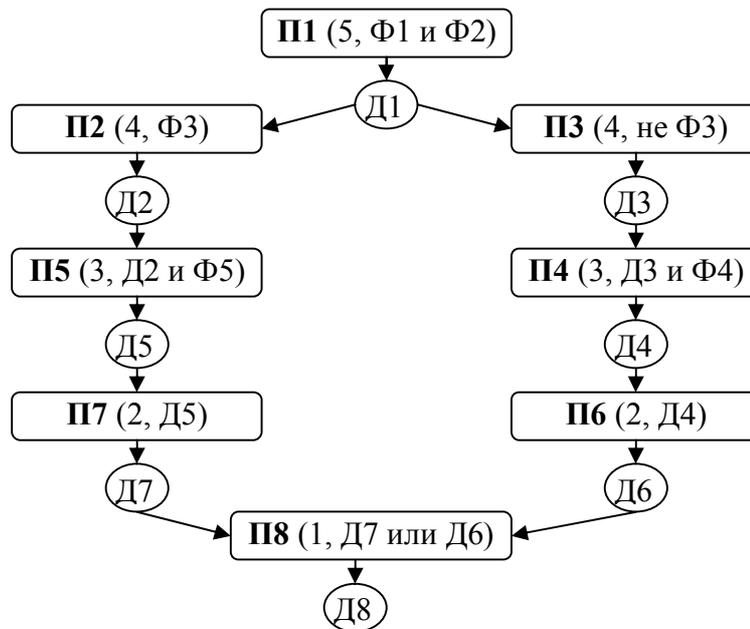


Рис. 1. Схема продукций предметной области «Ресторан».

Задачи

1. Построить продукционную модель представления знаний в предметной области «Аэропорт» (диспетчерская).
2. Построить продукционную модель представления знаний в предметной области «Железная дорога» (продажа билетов).
3. Построить продукционную модель представления знаний в предметной области «Торговый центр» (организация).
4. Построить продукционную модель представления знаний в предметной области «Автозаправка» (обслуживание клиентов).
5. Построить продукционную модель представления знаний в предметной области «Автопарк» (пассажирские перевозки).
6. Построить продукционную модель представления знаний в предметной области «Компьютерные сети» (организация).
7. Построить продукционную модель представления знаний в предметной области «Университет» (учебный процесс).
8. Построить продукционную модель представления знаний в предметной области «Компьютерная безопасность» (средства и способы ее обеспечения).
9. Построить продукционную модель представления знаний в предметной области «Компьютерная безопасность» (угрозы).
10. Построить продукционную модель представления знаний в предметной области «Интернет-кафе» (организация и обслуживание).

11. Построить продукционную модель представления знаний в предметной области «Разработка информационных систем» (ведение информационного проекта).
12. Построить продукционную модель представления знаний в предметной области «Туристическое агентство» (работа с клиентами).
13. Построить продукционную модель представления знаний в предметной области «Зоопарк» (организация).
14. Построить продукционную модель представления знаний в предметной области «Кухня» (приготовление пищи).
15. Построить продукционную модель представления знаний в предметной области «Больница» (прием больных).
16. Построить продукционную модель представления знаний в предметной области «Кинопрокат» (ассортимент и работа с клиентами).
17. Построить продукционную модель представления знаний в предметной области «Прокат автомобилей» (ассортимент и работа с клиентами).
18. Построить продукционную модель представления знаний в предметной области «Операционные системы» (функционирование).
19. Построить продукционную модель представления знаний в предметной области «Информационные системы» (виды и функционирование).
20. Построить продукционную модель представления знаний в предметной области «Предприятие» (структура и функционирование).

Семантическая сеть

Семантическая сеть — это ориентированный граф, вершины которого — понятия, а дуги — отношения между ними. Узлы в семантической сети обычно соответствуют объектам, концепциям, событиям или понятиям. Любой фрагмент сети, например одна вершина, две вершины и соединяющие их дуги, называют подсетью. Логический вывод (поиск решения) на семантической сети заключается в том, чтобы найти или сконструировать подсеть, удовлетворяющую некоторым условиям.

Отношения, представляемые дугами, в семантической сети могут быть различными (таблица 2). Типы отношений выбираются в зависимости от вида семантической сети (таблица 3) и решаемой задачи.

Таблица 2. Основные виды отношений в семантических сетях.

Тип	Описание
являться наследником (a-kind-of)	задает иерархические связи между классами
являться экземпляром	определяет значение, описывает конкретный объект, понятие

(is-a, например)	
это (are, есть)	может использоваться вместо связи a-kind-of в отношениях подразумевающих равенство или эквивалентность
являться частью (has-part)	определяет структурные связи, описывает части или целые объекты.
Функциональные	определяются обычно глаголами, отражают различные отношения (учить, владеть и т.д.).
Количественные	отображают количественные соотношения между вершинами (больше, меньше и т.д.)
Пространственные	отображают пространственные отношения между вершинами (близко, далеко и т.д.)
Временные	описывают временные связи между вершинами (скоро, долго, сейчас и т.д.)
Атрибутивные	описывают свойства объектов, понятий
Логические	описывают логические связи между вершинами (и, или, не)

Таблица 3. Типы семантических сетей.

Тип	Описание
<i>По типу знания</i>	
экстенциональные	описывает конкретные отношения данной ситуации.
интенциональные	описывают имена классов объектов, а не индивидуальные имена объектов, связи отражают те отношения, которые всегда присущи объектам данного класса.
<i>По типу ограничений на дуги и вершины</i>	
Простые	вершины сети не обладают внутренней структурой
иерархические	вершины обладают внутренней структурой, в иерархической сети есть возможность разделять сеть на подсети и устанавливать отношения не только между вершинами, но и между подсетями (различные подсети, существующие в сети, могут быть упорядочены в виде дерева подсетей, вершины которого—подсети, а дуги — отношения видимости)
динамические (сценарии)	сети с событиями
<i>По количеству типов отношений</i>	
Однородные	обладают только одним типом отношений
Неоднородные	количество типов отношений больше двух
<i>По арности отношений</i>	
Бинарные	все отношения в графе связывают ровно два понятия
N-арные	в сети есть отношения, связывающие более двух объектов

Пример решения задачи

Задача. Построить сетевую модель представления знаний в предметной области «Ресторан» (посещение ресторана).

Описание процесса решения. Для построения сетевой модели представления знаний необходимо выполнить следующие шаги:

- 1) Определить абстрактные объекты и понятия предметной области, необходимые для решения поставленной задачи. Оформить их в виде вершин.
- 2) Задать свойства для выделенных вершин, оформив их в виде вершин, связанных с исходными вершинами атрибутивными отношениями.
- 3) Задать связи между этими вершинами, используя функциональные, пространственные, количественные, логические, временные, атрибутивные отношения, а также отношения типа «являться наследником» и «являться частью».
- 4) Добавить конкретные объекты и понятия, описывающие решаемую задачу. Оформить их в виде вершин, связанных с уже существующими отношениями типа «являться экземпляром», «есть».
- 5) Проверить правильность установленных отношений (вершины и само отношение при правильном построении образуют предложение, например «Двигатель является частью автомобиля»).

Решение.

- 1) Ключевые понятия данной предметной области – ресторан, тот, кто посещает ресторан (клиент) и те, кто его обслуживают (повара, метрдотели, официанты, для простоты ограничимся только официантами). У обслуживающего персонала и клиентов есть общие характеристики, поэтому целесообразно выделить общее абстрактное понятие – человек. Продукцией ресторана являются блюда, которые заказывают клиенты.
Исходя из этого, вершины графа будут следующими: «Ресторан», «Человек», «Официант», «Клиент», «Заказ» и «Блюдо».
- 2) У этих объектов есть определенные свойства и атрибуты. Например, рестораны располагаются по определенным адресам, каждое блюдо из меню имеет свою цену. Поэтому добавим вершины «Адрес» и «Цена».
- 3) Определим для имеющихся вершин отношения и их типы, используя таблицу 2.
- 4) Добавим знание о конкретных фактах решаемой задачи. Пусть имеется два ресторана: «Вкуснятина» и «Вкусная еда», в первом работает

официантка Марина, а во втором официант Сергей. Пётр решил пойти в ресторан «Вкусная еда» и сделал заказ официанту на 2 блюда: картофель фри за 30 р., бифштекс за 130 р. Также известны адреса этих ресторанов и их специфика.

Исходя из этого, добавим соответствующие вершины в граф и соединим их функциональными отношениями и отношениями типа «например или являться экземпляром». Полученный в результате граф изображен на рис. 2.

- 5) Осуществим проверку установленных связей. Например, возьмем вершину «Блюдо» и пройдем по установленным связям. Получаем следующую информацию: блюдо является частью заказа, примерами блюд могут служить картофель фри и бифштекс.

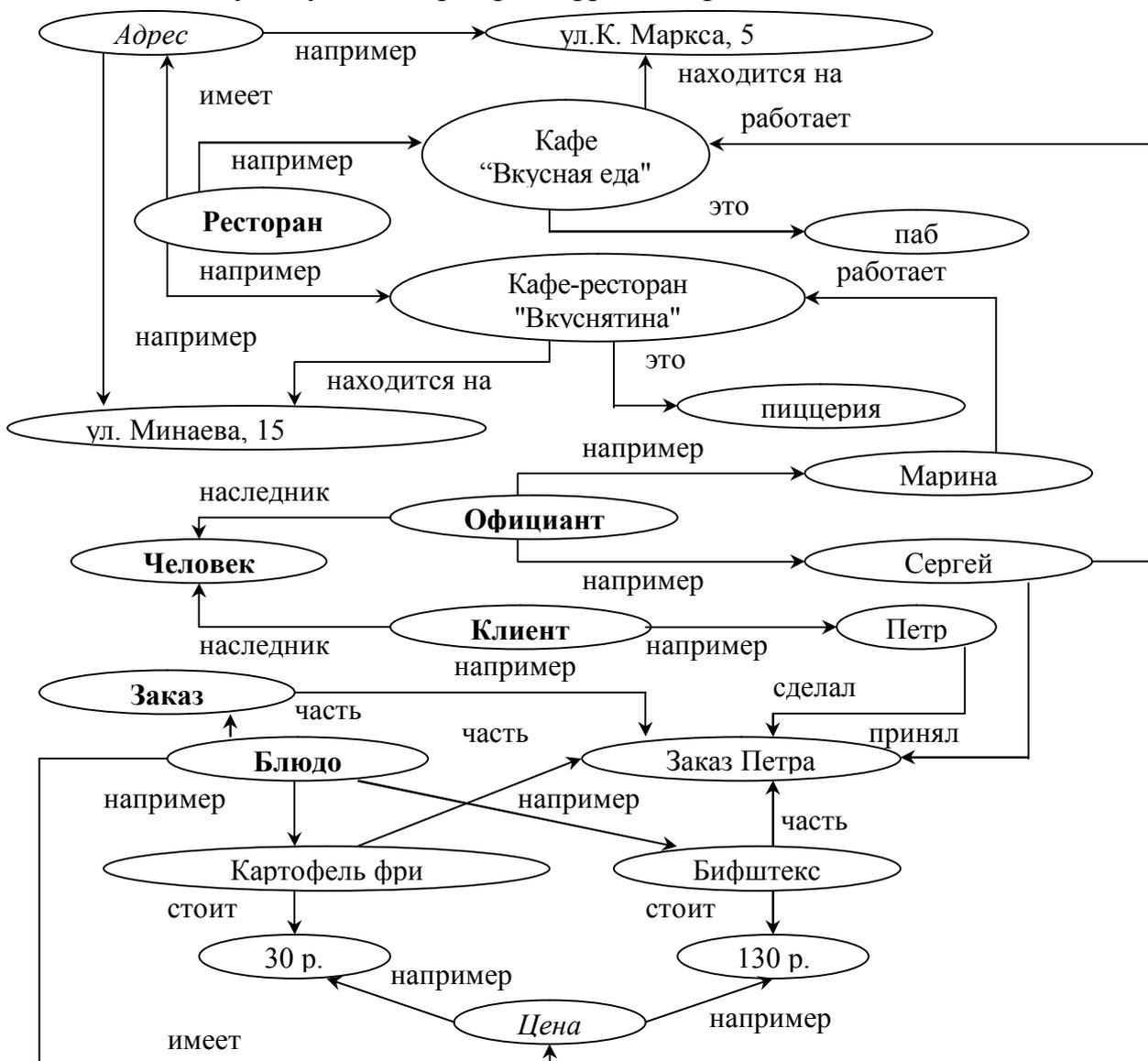


Рис. 2. Семантическая сеть предметной области «Ресторан».

Для получения ответа на какой-либо вопрос по этой задаче, необходимо найти соответствующий участок сети и, используя связи, получить результат.

Например, вопрос «Какова цена заказа Петра (сколько Петр заплатил за заказ)?» Из запроса понятно, что необходимо найти следующие вершины: «Цена», «Перт» и «Заказ» или «Заказ Петра». Часть семантической сети, находящаяся между этими вершинами, содержит ответ, а именно, частью заказа Петра являются картофель фри и бифштекс, которые стоят 30 и 130 р. соответственно. Больше информации о заказе Петра в модели нет, поэтому делаем вывод – Петр заплатил 160 р.

Задачи

1. Построить семантическую модель (сеть) представления знаний в предметной области «Аэропорт» (диспетчерская).
2. Построить семантическую модель (сеть) представления знаний в предметной области «Железная дорога» (продажа билетов).
3. Построить семантическую модель (сеть) представления знаний в предметной области «Торговый центр» (организация).
4. Построить семантическую модель (сеть) представления знаний в предметной области «Автозаправка» (обслуживание клиентов).
5. Построить семантическую модель (сеть) представления знаний в предметной области «Автопарк» (пассажирские перевозки).
6. Построить семантическую модель (сеть) представления знаний в предметной области «Компьютерные сети» (организация).
7. Построить семантическую модель (сеть) представления знаний в предметной области «Университет» (учебный процесс).
8. Построить семантическую модель (сеть) представления знаний в предметной области «Компьютерная безопасность» (средства и способы ее обеспечения).
9. Построить семантическую модель (сеть) представления знаний в предметной области «Компьютерная безопасность» (угрозы).
10. Построить семантическую модель (сеть) представления знаний в предметной области «Интернет-кафе» (организация и обслуживание).
11. Построить семантическую модель (сеть) представления знаний в предметной области «Разработка информационных систем» (ведение информационного проекта).

12. Построить семантическую модель (сеть) представления знаний в предметной области «Туристическое агентство» (работа с клиентами).
13. Построить семантическую модель (сеть) представления знаний в предметной области «Зоопарк» (организация).
14. Построить семантическую модель (сеть) представления знаний в предметной области «Кухня» (приготовление пищи).
15. Построить семантическую модель (сеть) представления знаний в предметной области «Больница» (прием больных).
16. Построить семантическую модель (сеть) представления знаний в предметной области «Кинопрокат» (ассортимент и работа с клиентами).
17. Построить семантическую модель (сеть) представления знаний в предметной области «Прокат автомобилей» (ассортимент и работа с клиентами).
18. Построить семантическую модель (сеть) представления знаний в предметной области «Операционные системы» (функционирование).
19. Построить семантическую модель (сеть) представления знаний в предметной области «Информационные системы» (виды и функционирование).
20. Построить семантическую модель (сеть) представления знаний в предметной области «Предприятие» (структура и функционирование).

Фреймовая модель

Фреймовая модель представления знаний была предложена М. Минским в 1979 году и является развитием семантических сетей.

Фрейм (англ. *frame*) - абстрактный образ для представления некоторого стереотипа восприятия. Каждый фрейм имеет собственное название и список слотов и их значений.

Значениями могут быть данные любого типа, а также название другого фрейма. Таким образом, фреймы образуют сеть. Кроме того, существует связь между фреймами типа АКО (a kind of), которая указывает на фрейм более высокого уровня иерархии, откуда неявно наследуются список и значения слотов. При этом возможно множественное наследование – перенос свойств от нескольких прототипов.

Любой фрейм может быть представлен следующим образом:

(ИМЯ ФРЕЙМА:

(имя 1-го слота: значение 1-го слота),

(имя 2-го слота: значение 2-го слота),

.....

(имя N-го слота: значение N-го слота)).

Табличное представление слота выглядит следующим образом (таблица 4):

Таблица 4. Структура фрейма.

ИМЯ ФРЕЙМА			
Имя слота	Значение слота	Способ получения значения	Демон

При табличном представлении фрейма кроме уже описанных составляющих фрейма указываются и дополнительные параметры. Способ получения значения определяет, как именно устанавливается значение конкретного слота. Существует несколько способов (таблица 5), выбор способа зависит от свойств самих данных.

Таблица 5. Способы получения значений слотов.

Способ	Описание
По умолчанию от прототипа (родителя)	Слоту присваивается значение, определенное по умолчанию во фрейме-прототипе, некоторые стандартные значения.
Через наследование	Отличается от первого способа тем, что значение задано в специальном слоте родительского фрейма, соединенного с текущим связью АКО.
По формуле	Слоту назначается формула, результат вычисления которой является значением слота.
Через присоединенную процедуру	Слоту назначается процедура, позволяющая получить значение слота алгоритмически.
Из внешних источников данных	При использовании модели в интеллектуальных системах данные, являющиеся значениями слотов, могут поступать из баз данных, от системы датчиков, от пользователя.

В теории фреймов допускается, чтобы к слотам присоединялись различные специальные процедуры. Для этого используются так называемые демоны. Демоном (таблица 6) называется процедура, автоматически запускаемая при выполнении некоторого условия (события) при обращении к соответствующему слоту. Демонов может быть несколько. Наиболее похож механизм присоединенных процедур к триггерам в реляционных базах данных.

Таблица 6. Наиболее распространенные демоны.

Демон	Событие	Описание
IF-REMOVED	если удалено	Выполняется, когда информация удаляется из слота.
IF-ADDED	если добавлено	Выполняется, когда новая информация записывается в слот.
IF-NEEDED	по требованию	Выполняется, когда запрашивается информация из пустого слота.
IF-DEFAULT	по умолчанию	Выполняется, когда устанавливается значение по умолчанию.

Существует несколько видов фреймов, которые позволяют описать предметную область и решаемую задачу. В таблице 7 представлены наиболее распространенные типы фреймов, указаны типы знаний, которые они отображают, а также примеры фреймов данного типа из различных предметных областей.

Таблица 7. Типы фреймов.

Тип фрейма	Тип знания	Описание	Пример
<i>По познавательному назначению</i>			
Фреймы-прототипы (шаблоны, образцы)	интенциональные	отражают знания об абстрактных стереотипных понятиях, которые являются классами каких-то конкретных объектов	человек, автомобиль
Фреймы-экземпляры (примеры)	экстенциональные	отражают знания о конкретных фактах предметной области	Иванов И.И., ВАЗ-2110
<i>По функциональному назначению</i>			
Фреймы-структуры (объекты)	декларативные	отображают абстрактные и конкретные предметы и понятия предметной области (содержат набор характеристик, описывающий объект или понятие)	заем, залог, вексель, человек, лекция
Фреймы-операции	процедурные	отображают различные процессы преобразования или использования объектов предметной области (содержат набор характеристик процесса)	процессы получение заёма, синтеза устройств
Фреймы-ситуации	прагматические	отображают типичные ситуации, в которых могут находиться фреймы объекты и фреймы роли (содержат набор характеристик, идентифицирующих ситуацию)	авария, тревога, рабочий режим устройства
Фреймы-сценарии	Технологические	отображают развитие ситуации, типовую структуру для некоторого действия, понятия, события, отображает динамику (содержат набор характеристик, позволяющих обеспечить развитие системы по данному сценарию)	банкротство, празднование именин, сдача экзамена
Фреймы-роли	функциональные	отображают типичную роль, выполняемую фреймом-объектом в определенной ситуации (содержат набор характеристик роли)	менеджер, кассир, клиент, студент, преподаватель

Пример решения задачи

Задача. Построить фреймовую модель представления знаний в предметной области «Ресторан» (посещение ресторана).

Описание процесса решения. Для построения фреймовой модели представления знаний необходимо выполнить следующие шаги:

- 1) Определить абстрактные объекты и понятия предметной области, необходимые для решения поставленной задачи. Оформить их в виде фреймов-прототипов (фреймов-объектов, фреймов-ролей).
- 2) Задать конкретные объекты предметной области. Оформить их в виде фреймов-экземпляров (фреймов-объектов, фреймов-ролей).
- 3) Определить набор возможных ситуаций. Оформить их в виде фреймов-ситуаций (прототипы). Если существуют прецеденты по ситуациям в предметной области, добавить фреймы-экземпляры (фреймы-ситуации).
- 4) Описать динамику развития ситуаций (переход от одних к другим) через набор сцен. Оформить их в виде фреймов-сценариев.
- 5) Добавить фреймы-объекты сценариев и сцен, которые отражают данные конкретной задачи.

Решение.

- 1) Ключевые понятия данной предметной области – ресторан, тот, кто посещает ресторан (клиент) и те, кто его обслуживают (повара, метрдотели, официанты, для простоты ограничимся только официантами). У обслуживающего персонала и клиентов есть общие характеристики, поэтому целесообразно выделить общее абстрактное понятие – человек. Тогда фреймы «Ресторан» и «Человек» являются прототипами-образцами, а фреймы «Официант» и «Клиент» – прототипами-ролями. Также нужно определить основные слоты фреймов – характеристики, имеющие значения для решаемой задачи.

ЧЕЛОВЕК			
Имя слота	Значение слота	Способ получения значения	Демон
пол	Мужской или	из внешних источников	
возраст	От 0 до 120 лет	из внешних источников	

РЕСТОРАН			
Имя слота	Значение слота	Способ получения значения	Демон
Название		из внешних источников	
Адрес		из внешних источников	
Часы работы		из внешних источников	
Специализация		из внешних источников	

Класс	Средний или высший	из внешних источников	
-------	--------------------	-----------------------	--

Фреймы-наследники содержат все слоты своих родителей, они явно прописываются только в случае изменения какого-либо параметра.

ОФИЦИАНТ (АКО ЧЕЛОВЕК)			
Имя слота	Значение слота	Способ получения значения	Демон
возраст	От 18 до 55 лет	из внешних источников	
стаж работы		из внешних источников	
зарплата		из внешних источников	
график работы		из внешних источников	
место работы	Фрейм-объект	из внешних источников	

КЛИЕНТ (АКО ЧЕЛОВЕК)			
Имя слота	Значение слота	Способ получения значения	Демон
Вид оплаты	Наличные или карточка	По умолчанию (наличные)	
Статус	Обычный или Vip	По умолчанию (обычный)	
Форма заказа	Заказ есть или нет	По умолчанию (заказа нет)	
Чаевые		Из внешних источников	

- 2) Фреймы-образцы описывают конкретную ситуацию: какие рестораны имеются в городе, как именно организовывается посещение, кто является посетителем, кто работает в выбранном ресторане и т.д. Поэтому определим следующие фреймы-образцы, являющиеся наследниками фреймов-прототипов:

КАФЕ-РЕСТОРАН "ВКУСНЯТИНА" (АКО РЕСТОРАН)			
Имя слота	Значение слота	Способ получения значения	Демон
Название	Вкуснятина	из внешних источников	
Адрес	г. Ульяновск, улица Минаева, 15	из внешних источников	
Часы работы	9:00-00:00	из внешних источников	
Специализация	Пиццерия	из внешних источников	
Класс	Средний или высший	из внешних источников	

КАФЕ "ВКУСНАЯ ЕДА" (АКО РЕСТОРАН)			
Имя слота	Значение слота	Способ получения значения	Демон
Название	Вкусная еда	из внешних источников	
Адрес	г. Ульяновск, улица Карла Маркса, 5	из внешних источников	
Часы работы	9:00-00:00	из внешних источников	
Специализация	Паб	из внешних источников	
Класс	Средний	из внешних источников	

СЕРГЕЙ (АКО ОФИЦИАНТ)			
Имя слота	Значение слота	Способ получения значения	Демон
возраст	27	из внешних источников	

пол	мужской	из внешних источников	
стаж работы	5	из внешних источников	
зарплата	7 000	из внешних источников	
график работы	Через день с 18:00 до 00:00	из внешних источников	
место работы	КАФЕ "ВКУСНАЯ"	из внешних источников	

МАРИНА (АКО ОФИЦИАНТ)			
Имя слота	Значение слота	Способ получения значения	Демон
возраст	24	из внешних источников	
Пол	женский	из внешних источников	
стаж работы	2	из внешних источников	
зарплата	8 200	из внешних источников	
график работы	Каждый день с 9:00 до 14:00	из внешних источников	
место работы	КАФЕ-РЕСТОРАН "ВКУСНЯТИНА"	из внешних источников	

ПЁТР (АКО КЛИЕНТ)			
Имя слота	Значение слота	Способ получения значения	Демон
пол	мужской	из внешних источников	
возраст	19	из внешних источников	
Вид оплаты	Наличные	По умолчанию (наличные)	
Статус	Обычный	По умолчанию (обычный)	
Форма заказа	Заказа нет	По умолчанию (заказа нет)	
Чаевые	7 % от суммы заказа	Из внешних источников	

- 3) Фреймы-ситуации описывают возможные ситуации. В ресторане клиент попадает в несколько типичных ситуаций: заказ и оплата. Возможны и другие не типичные ситуации: клиент подавился, у клиента нет наличности для оплаты счета и т.д. Рассмотрим типичные ситуации (их может быть больше):

ЗАКАЗ			
Имя слота	Значение слота	Способ получения значения	Демон
Перечень блюд		из внешних источников	IF-ADDED (изменяет слот «Перечень цен»)
Перечень цен		Присоединенная процедура	IF-ADDED (изменяет слот «Сумма заказчик»)
Сумма заказа		Присоединенная	
Принял заказ	Фрейм-образец	из внешнего источника	
Сделал заказ	Фрейм-образец	из внешнего источника	

ОПЛАТА

Имя слота	Значение слота	Способ получения значения	Демон
Вид платежа		из внешних источников	IF-ADDED (изменяет слот «Чаевые»)
Чаевые		Присоединенная	
Оплатил	Фрейм-образец	Присоединенная процедура	
Заказ	Фрейм-образец	из внешних источников	IF-ADDED (изменяет слот «Оплатил»)

- 4) Ситуации возникают после наступления каких-то событий, выполнения условий и могут следовать одна за другой. Динамику предметной области можно отобразить в фреймах-сценариях. Их может быть множество, опишем наиболее общий и типичный сценарий посещения ресторана:

ПОСЕЩЕНИЕ РЕСТОРАНА			
Имя слота	Значение слота	Способ получения значения	Демон
Посетитель	Фрейм-объект	из внешних источников	
Ресторан	Фрейм-объект	из внешних источников	IF-ADDED, IF-REMOVED (изменяют слот «Официант»)
Официант	Фрейм-объект	присоединенная процедура (определяет по выбранному ресторану)	
Сцена 1	Вход, выбор	из внешних источников	
Сцена 2	Заказ	из внешних источников	
Сцена 3	Еда	из внешних источников	
Сцена 4	Оплата	из внешних источников	
Сцена 5	Выход	из внешних источников	

- 5) Пусть в рамках нашей задачи Пётр посетил ресторан «Вкусная еда». Тогда фреймы будут заполнены следующим образом:

ПОСЕЩЕНИЕ «Вкусной еды» (АКО ПОСЕЩЕНИЕ РЕСТОРАНА)			
Имя слота	Значение слота	Способ получения значения	Демон
Посетитель	ПЁТР	из внешних источников	
Ресторан	КАФЕ "ВКУСНАЯ ЕДА"	из внешних источников	IF-ADDED, IF-REMOVED (изменяют слот «Официант»)
Официант	СЕРГЕЙ	присоединенная процедура (определяет по выбранному ресторану)	
Сцена 1	Вход, выбор	из внешних источников	

Сцена 2	ЗАКАЗ ПЕТРА	из внешних источников	
Сцена 3	Еда	из внешних источников	
Сцена 4	ОПЛАТА ПЕТРА	из внешних источников	
Сцена 5	Выход	из внешних источников	

ЗАКАЗ ПЕТРА (АКО ЗАКАЗ)			
Имя слота	Значение слота	Способ получения значения	Демон
Перечень блюд	Отбивная, темное пиво	из внешних источников	IF-ADDED (изменяет слот «Перечень цен»)
Перечень цен	250, 75	Присоединенная процедура	IF-ADDED (изменяет слот «Сумма заказк»)
Сумма заказа	325	Присоединенная	
Принял заказ	СЕРГЕЙ	из внешнего источника	
Сделал заказ	ПЕТР	из внешнего источника	

ОПЛАТА ПЕТРА (АКО ОПЛАТА)			
Имя слота	Значение слота	Способ получения значения	Демон
Вид платежа	Наличные	из внешних источников	IF-ADDED (изменяет слот «Чаевые»)
Чаевые	30	Присоединенная процедура	
Оплатил	ПЕТР	из внешних источников	
Заказ	ЗАКАЗ ПЕТРА	из внешних источников	IF-ADDED (изменяет слот «Оплатил»)

Взаимосвязь различных видов фреймов отображается графически в виде графа (рис. 3).

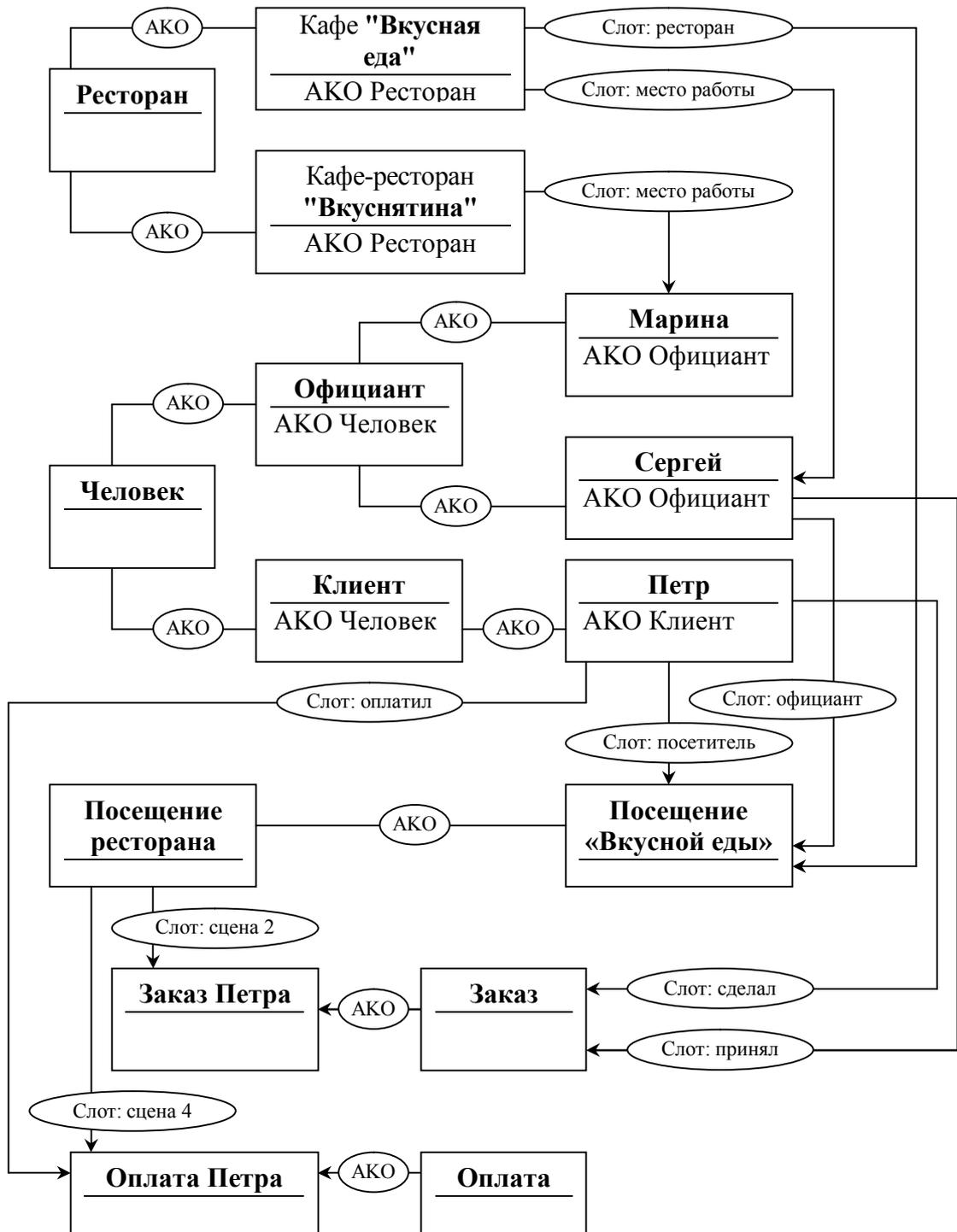


Рис. 3. Схема фреймов для предметной области «Ресторан».

Использование фреймовой модели аналогично семантической, только в процессе получения ответа кроме вершин учитываются и слоты.

Например, получить ответ на вопрос «Кто работает официантом в ресторане “Вкусная еда”?» можно следующим образом: из запроса понятно, что необходимо найти фрейм «Ресторан “Вкусная еда”» и проследить связь с фреймом «Сергей», являющимся наследником фрейма «Официант». Также можно найти слот «Место работы» и проверив его значение во фреймах

наследниках фрейма «Официант» определить, что официантом в ресторане «Вкусная еда» работает Сергей.

Задачи

1. Построить фреймовую модель представления знаний в предметной области «Аэропорт» (диспетчерская).
2. Построить фреймовую модель представления знаний в предметной области «Железная дорога» (продажа билетов).
3. Построить фреймовую модель представления знаний в предметной области «Торговый центр» (организация).
4. Построить фреймовую модель представления знаний в предметной области «Автозаправка» (обслуживание клиентов).
5. Построить фреймовую модель представления знаний в предметной области «Автопарк» (пассажирские перевозки).
6. Построить фреймовую модель представления знаний в предметной области «Компьютерные сети» (организация).
7. Построить фреймовую модель представления знаний в предметной области «Университет» (учебный процесс).
8. Построить фреймовую модель представления знаний в предметной области «Компьютерная безопасность» (средства и способы ее обеспечения).
9. Построить фреймовую модель представления знаний в предметной области «Компьютерная безопасность» (угрозы).
10. Построить фреймовую модель представления знаний в предметной области «Интернет-кафе» (организация и обслуживание).
11. Построить фреймовую модель представления знаний в предметной области «Разработка информационных систем» (ведение информационного проекта).
12. Построить фреймовую модель представления знаний в предметной области «Туристическое агентство» (работа с клиентами).
13. Построить фреймовую модель представления знаний в предметной области «Зоопарк» (организация).
14. Построить фреймовую модель представления знаний в предметной области «Кухня» (приготовление пищи).
15. Построить фреймовую модель представления знаний в предметной области «Больница» (прием больных).

16. Построить фреймовую модель представления знаний в предметной области «Кинопрокат» (ассортимент и работа с клиентами).
17. Построить фреймовую модель представления знаний в предметной области «Прокат автомобилей» (ассортимент и работа с клиентами).
18. Построить фреймовую модель представления знаний в предметной области «Операционные системы» (функционирование).
19. Построить фреймовую модель представления знаний в предметной области «Информационные системы» (виды и функционирование).
20. Построить фреймовую модель представления знаний в предметной области «Предприятие» (структура и функционирование).

Нейронные сети

Искусственная нейронная сеть – математическая модель, реализуемая программно или аппаратно, построенная по подобию естественных нейронных сетей (сетей нервных клеток живого организма), представляющая собой соединение простых взаимодействующих между собой процессоров - искусственных нейронов.

Схема искусственного нейрона представлена на рис. 4, где $X_1..X_N$ – входы нейрона, $W_1..W_N$ – синаптические веса связей нейрона, S – взвешенная сумма входных значений нейрона, $F(S)$ – функция активации, значением которой является Y – выходное значение нейрона.

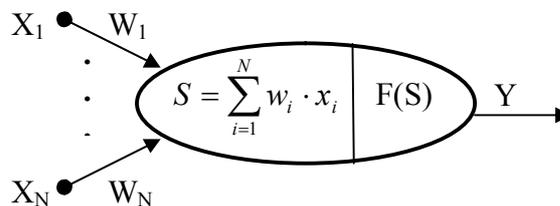
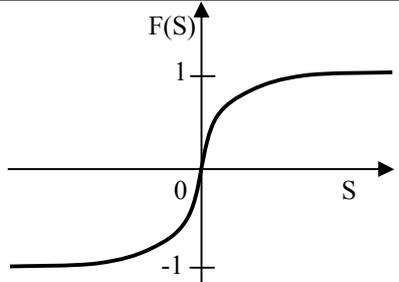


Рис. 4. Формальный нейрон (математическая модель)

Функции активации могут различны, наиболее часто используемые представлены в таблице 8.

Таблица 8. Основные функции активации.

Название	Формула	График (пример)
Пороговая	$F(S) = \begin{cases} 1, & \text{при } S \geq T, \\ 0, & \text{при } S < T, \end{cases}$ $T = const.$	
Линейная	$F(S) = k \cdot S,$ $k = const.$	
Сигмоидальная	$F(S) = \frac{1}{1 + e^{-S \cdot k}},$ $k = const.$	

Гиперболический тангенс	$F(S) = th\left(\frac{S}{k}\right) = \frac{e^{\frac{S}{k}} + e^{-\frac{S}{k}}}{e^{\frac{S}{k}} - e^{-\frac{S}{k}}},$ $k = const.$	
-------------------------	---	---

Существует множество нейронных сетей, которые классифицируются по нескольким признакам (таблица 9). Наибольшее распространение получили слоистые сети прямого распространения.

Таблица 9. Типы искусственных нейронных сетей.

Тип	Описание
По топологии	
Полносвязные	Каждый нейрон связан с другим нейроном в сети (из-за высокой сложности обучения не используется).
Слоистые	Нейроны располагаются слоями, каждый нейрон последующего слоя связан с нейронами предыдущего. Есть однослойные и многослойные сети.
По типу связей	
Прямого распространения	Все связи между нейронами идут от выходов нейронов предыдущего слоя к входам нейронов последующего.
Рекуррентные	Допускаются связи выходов нейронов последующих слоев с входами нейронов предыдущих.
По организации обучения	
С учителем	При обучении используются обучающие выборки, в которых определены требуемые от сети выходные значения, такие сети используют для решения задач классификации.
Без учителя	Нейронная сеть сама в процессе работы выделяет классы объектов и относит объект к определенному классу, такие сети используют для задач кластеризации.
По типу сигнала	
Бинарные	На вход нейронных сетей подают только нули или единицы.
Аналоговые	Подаваемые на входы нейронов сигналы могут быть произвольными (вещественными числами).
По типу структур	
Однородная	Все нейроны в нейронной сети используют одну функцию активации.
Неоднородная	Нейроны в нейронной сети имеют разные функции активации.

Для решения конкретной задачи нужно выбрать подходящую нейронную сеть. При этом нужно учитывать не только перечисленные в таблице критерии, но и архитектуру сети. Выбор архитектуры подразумевает определение количества слоев и нейронов в этих слоях. Не существует

формального алгоритма по определению нужной архитектуры, поэтому на практике выбирают или заведомо маленькую сеть и постепенно ее наращивают или заведомо большую и постепенно выявляют неиспользуемые связи и сокращают сеть.

Нейронная сеть, прежде чем использоваться на практике для решения какой-либо задачи, должна быть обучена. Обучение нейронной сети - это процесс настройки синаптических весов. Существует множество алгоритмов, ориентированных на определенные типы сетей и на конкретные задачи, рассмотрим алгоритмы для однослойной и многослойной сетей.

Алгоритм обучения сети по Δ -правилу

Простейшая нейронная сеть – однослойная (рис. 5), представляющая из себя расположенные параллельно нейроны, получающие на входы одинаковые сигналы, но имеющие различные синаптические связи. Количество входов и выходов такой нейронной сети соответствует количеству нейронов.

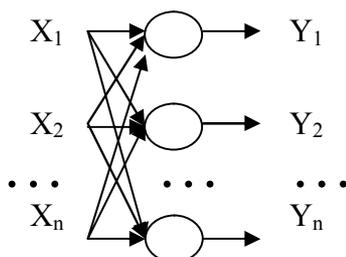


Рис. 5. Однослойная нейронная сеть

Такие нейронные сети можно обучать с помощью алгоритма обучения по Δ -правилу.

Алгоритм обучения по Δ -правилу:

1 шаг: инициализация матрицы весов (и порогов, в случае использования пороговой функции активации) случайным образом.

2 шаг: предъявление нейронной сети образа (на вход подаются значения из обучающей выборки – вектор X), берется соответствующий выход (вектор D).

3 шаг: вычисление выходных значений нейронной сети (вектор Y).

4 шаг: вычисление для каждого нейрона величины расхождения реального результата с желаемым.

$$\varepsilon_i = (d_i - y_i),$$

где d_i – желаемое выходное значение на i -нейроне, y_i - реальное значение на i -нейроне.

5 шаг: изменение весов (и порогов при использовании пороговой функции) по формулам:

$$w_{ij}(t+1) = w_{ij}(t) - \eta \cdot \varepsilon_i \cdot x_j,$$

$$\theta_i(t+1) = \theta_i(t) - \eta \cdot \varepsilon_i,$$

где t -номер текущей итерации цикла обучения, w_{ij} - вес связи j -входа с i -нейроном, η - коэффициент обучения, задается от 0 до 1, x_j - входное значение, θ_i - пороговое значение i -нейрона.

6 шаг: проверка условия продолжения обучения (вычисление значения ошибки и/или проверка заданного количества итераций). Если обучение не завершено, то 2 шаг, иначе заканчиваем обучение.

Пример решения задачи

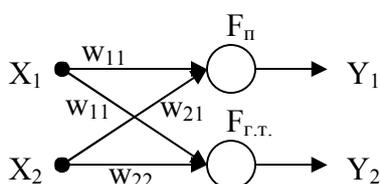
Задача. Просчитать одну итерацию цикла обучения по Δ -правилу однослойной бинарной неоднородной нейронной сети, состоящей из 2 нейронов и имеющей функции активации: гиперболический тангенс ($k=1$) и пороговую функцию ($T=0,7$). В качестве обучающей выборки использовать таблицу истинности для операций эквивалентности и дизъюнкции (не использовать первую строчку таблицы). Синаптические веса задать случайным образом.

Описание процесса решения. Для обучения нейронной сети по Δ - правилу необходимо:

- 1) Графически отобразить структуру нейронной сети. Определить размерность матрицы синаптических весов.
- 2) Определить обучающую выборку, представив ее в табличном виде.
- 3) Выбрать входные данные, на которых будет рассматриваться итерация цикла обучения.
- 4) Следуя алгоритмы обучения по Δ –правилу, просчитать одну итерацию цикла и представить новые синаптические веса в матричном виде.

Решение.

- 1) По заданию нейронная сеть состоит из двух нейронов, значит, входов у однослойной нейронной сети будет 2 и выходов 2, а синаптических весов 4. Первый нейрон имеет пороговую функцию активации, второй – гиперболический тангенс.



- 2) По заданию нейронная сеть бинарная, поэтому на ее входы могут подаваться только нули и единицы, так как входа 2, то возможных комбинаций входных значений будет 4 (обучающая выборка будет состоять из 4 векторов). Выход первого нейрона согласно заданию соответствует оператору эквивалентности, а второго – дизъюнкции. Поэтому таблица с обучающей выборкой будет выглядеть следующим образом:

X1	X2	D1	D2
0	0	1	0
0	1	0	1
1	0	0	1
1	1	1	1

- 3) Пусть в качестве вектора обучения будет рассматриваться 3-ая строка таблицы.
- 4) Следуя алгоритмы обучения по Δ -правилу выполним 6 шагов

1 шаг: зададим матрицу весов случайным образом из интервала $[0,1]$:

Wij(1)	1	2
1	0.7	1
2	0.5	0.2

2 шаг: вектор $X = \{1, 0\}$, вектор $D = \{0, 1\}$.

3 шаг: вычисление выходных значений нейронной сети (вектор Y).

$$T = 0.7;$$

$$S_1 = x_1 \cdot w_{11} + x_2 \cdot w_{21} = 1 \cdot 0.7 + 0 \cdot 0.5 = 0.7;$$

$$Y_1 = \begin{cases} 1, & \text{при } S_1 \geq T \\ 0, & \text{при } S_1 < T \end{cases} = \begin{cases} 1, & \text{при } 0.7 \geq 0.7 \\ 0, & \text{при } 0.7 < 0.7 \end{cases} = 1.$$

$$k = 1,$$

$$S_2 = x_1 \cdot w_{12} + x_2 \cdot w_{22} = 1 \cdot 0.9 + 0 \cdot 0.2 = 0.9,$$

$$Y_2 = th\left(\frac{S_2}{k}\right) = \frac{e^{0.9} + e^{-0.9}}{e^{0.9} - e^{-0.9}} \approx 1.39.$$

4 шаг:

$$\varepsilon_1 = (d_1 - y_1) = (0 - 1) = -1,$$

$$\varepsilon_2 = (d_2 - y_2) = (1 - 1.39) = -0.39.$$

5 шаг: задаем η - коэффициент обучения от 0 до 1 и изменяем веса:

$$\eta = 0.8,$$

$$w_{11}(2) = w_{11}(1) - 0.8 \cdot \varepsilon_1 \cdot x_1 = 0.7 - 0.8 \cdot (-1) \cdot 1 = 1.5,$$

$$w_{21}(2) = w_{21}(1) - 0.8 \cdot \varepsilon_1 \cdot x_2 = 0.5 - 0.8 \cdot (-1) \cdot 0 = 0.5,$$

$$\theta_1(2) = \theta_1(1) - 0.8 \cdot \varepsilon_1 = 0.7 - 0.8 \cdot (-1) = 1.5,$$

$$w_{12}(2) = w_{12}(1) - 0.8 \cdot \varepsilon_2 \cdot x_1 = 0.9 - 0.8 \cdot (-0.39) \cdot 1 = 1.212,$$

$$w_{22}(2) = w_{22}(1) - 0.8 \cdot \varepsilon_2 \cdot x_2 = 0.2 - 0.8 \cdot (-0.39) \cdot 0 = 0.2.$$

Wij(2)	1	2
1	1.5	1.212
2	0.5	0.2

6 шаг: вычислим среднеквадратичную ошибку (можно выбрать другие методы оценки ошибки)

$$\varepsilon = \sum_{i=1}^H (d_i - y_i)^2 = \sum_{i=1}^2 \varepsilon_i^2 = \varepsilon_1^2 + \varepsilon_2^2 = (-1)^2 + (-0.39)^2 = 1.1521.$$

H - количество нейронов.

Так как мы рассматриваем одну итерацию цикла обучения в любом случае выходим из цикла.

Задачи

1. Просчитать одну итерацию цикла обучения по Δ -правилу однослойной бинарной однородной нейронной сети, состоящей из 2 нейронов и имеющей пороговую функцию активации ($T=0,7$). В качестве обучающей выборки использовать таблицу истинности для операций дизъюнкции и импликации (не использовать первую строчку таблицы). Синаптические веса задать случайным образом.
2. Просчитать одну итерацию цикла обучения по Δ -правилу однослойной бинарной однородной нейронной сети, состоящей из 2 нейронов и имеющей линейную функцию активации ($k=0,6$). В качестве обучающей выборки использовать таблицу истинности для операций конъюнкции и дизъюнкции (не использовать первую строчку таблицы). Синаптические веса задать случайным образом.
3. Просчитать одну итерацию цикла обучения по Δ -правилу однослойной бинарной однородной нейронной сети, состоящей из 2 нейронов и имеющей сигмоидальную функцию активации ($k=1$). В качестве обучающей выборки использовать таблицу истинности для операций импликации и конъюнкции (не использовать первую строчку таблицы). Синаптические веса задать случайным образом.
4. Просчитать одну итерацию цикла обучения по Δ -правилу однослойной бинарной однородной нейронной сети, состоящей из 2 нейронов и имеющей функцию активации гиперболический тангенс ($k=1$). В качестве обучающей выборки использовать таблицу истинности для операций эквивалентности и импликации (не использовать первую строчку таблицы). Синаптические веса задать случайным образом.
5. Просчитать одну итерацию цикла обучения по Δ -правилу однослойной бинарной неоднородной нейронной сети, состоящей из 2 нейронов и

- имеющей функции активации: гиперболический тангенс ($k=2$) и пороговую функцию ($T=0,5$). В качестве обучающей выборки использовать таблицу истинности для операций эквивалентности и конъюнкции (не использовать первую строчку таблицы). Синаптические веса задать случайным образом.
6. Просчитать одну итерацию цикла обучения по Δ -правилу однослойной бинарной неоднородной нейронной сети, состоящей из 2 нейронов и имеющей функции активации: сигмоидальную ($k=1$) и линейную ($k=0,6$). В качестве обучающей выборки использовать таблицу истинности для операций импликации и конъюнкции (не использовать первую строчку таблицы). Синаптические веса задать случайным образом.
 7. Просчитать одну итерацию цикла обучения по Δ -правилу однослойной бинарной неоднородной нейронной сети, состоящей из 2 нейронов и имеющей функции активации: линейную ($k=0,7$) и пороговую ($T=0,75$). В качестве обучающей выборки использовать таблицу истинности для операций конъюнкции и эквивалентности (не использовать первую строчку таблицы). Синаптические веса задать случайным образом.
 8. Просчитать одну итерацию цикла обучения по Δ -правилу однослойной бинарной неоднородной нейронной сети, состоящей из 2 нейронов и имеющей функции активации: пороговую ($T=0,8$) и сигмоидальную ($k=1$). В качестве обучающей выборки использовать таблицу истинности для операций конъюнкции и импликации (не использовать первую строчку таблицы). Синаптические веса задать случайным образом.
 9. Просчитать одну итерацию цикла обучения по Δ -правилу однослойной бинарной неоднородной нейронной сети, состоящей из 2 нейронов и имеющей функции активации: гиперболический тангенс ($k=2$) и линейную ($k=0,8$). В качестве обучающей выборки использовать таблицу истинности для операций дизъюнкции и эквивалентности (не использовать первую строчку таблицы). Синаптические веса задать случайным образом.
 10. Просчитать одну итерацию цикла обучения по Δ -правилу однослойной бинарной неоднородной нейронной сети, состоящей из 2 нейронов и имеющей функции активации: гиперболический тангенс ($k=2$) и сигмоидальную ($k=0,9$). В качестве обучающей выборки использовать таблицу истинности для операций импликации и дизъюнкции (не

использовать первую строчку таблицы). Синаптические веса задать случайным образом.

11. Просчитать одну итерацию цикла обучения по Δ -правилу однослойной бинарной однородной нейронной сети, состоящей из 3 нейронов и имеющей функцию активации гиперболический тангенс ($k=3$). В качестве обучающей выборки использовать таблицу истинности для $X1 \& X2 \rightarrow X3$, $X1 \& X2$ и $X2 \rightarrow X3$ (не использовать первую строчку таблицы). Синаптические веса задать случайным образом.
12. Просчитать одну итерацию цикла обучения по Δ -правилу однослойной бинарной однородной нейронной сети, состоящей из 3 нейронов и имеющей сигмоидальную функцию активации ($k=1$). В качестве обучающей выборки использовать таблицу истинности для $X1 \rightarrow X2 \& X3$, $X1 \& X2$ и $X1 \& X3$ (не использовать первую строчку таблицы). Синаптические веса задать случайным образом.
13. Просчитать одну итерацию цикла обучения по Δ -правилу однослойной бинарной однородной нейронной сети, состоящей из 3 нейронов и имеющей линейную функцию активации ($k=0,9$). В качестве обучающей выборки использовать таблицу истинности для $X3 \rightarrow X1 \& X2$, $X2 \& X3$, $X2 \rightarrow X3$ (не использовать первую строчку таблицы). Синаптические веса задать случайным образом.
14. Просчитать одну итерацию цикла обучения по Δ -правилу однослойной бинарной однородной нейронной сети, состоящей из 3 нейронов и имеющей пороговую функцию активации ($T=0,4$). В качестве обучающей выборки использовать таблицу истинности для $(X2 \rightarrow X1) \& X3$ (не использовать первую строчку таблицы). Синаптические веса задать случайным образом.
15. Просчитать одну итерацию цикла обучения по Δ -правилу однослойной аналоговой однородной нейронной сети, состоящей из 3 нейронов и имеющей линейную функцию активации ($k=0,9$). Синаптические веса и обучающую выборку задать случайным образом (не нули).
16. Просчитать одну итерацию цикла обучения по Δ -правилу однослойной аналоговой однородной нейронной сети, состоящей из 3 нейронов и имеющей сигмоидальную функцию активации ($k=0,8$). Синаптические веса и обучающую выборку задать случайным образом (не нули).
17. Просчитать одну итерацию цикла обучения по Δ -правилу однослойной аналоговой однородной нейронной сети, состоящей из 3 нейронов и

имеющей пороговую функцию активации ($T=0,8$). Синаптические веса и обучающую выборку задать случайным образом (не нули).

18. Просчитать одну итерацию цикла обучения по Δ -правилу однослойной аналоговой однородной нейронной сети, состоящей из 3 нейронов и имеющей функцию активации – гиперболический тангенс ($k=1$). Синаптические веса и обучающую выборку задать случайным образом (не нули).
19. Просчитать одну итерацию цикла обучения по Δ -правилу однослойной аналоговой неоднородной нейронной сети, состоящей из 3 нейронов и имеющей функции активации: сигмоидальную ($k=1$), линейную ($k=0,8$) и пороговую ($T=0,5$). Синаптические веса и обучающую выборку задать случайным образом (не нули).
20. Просчитать одну итерацию цикла обучения по Δ -правилу однослойной аналоговой неоднородной нейронной сети, состоящей из 3 нейронов и имеющей функции активации: гиперболический тангенс ($k=1$), сигмоидальную ($k=0,8$) и пороговую ($T=0,6$). Синаптические веса и обучающую выборку задать случайным образом (не нули).

Алгоритм обратного распространения ошибки

Многослойная искусственная нейронная сеть (рис. 6) может содержать произвольное количество слоев (K), каждый слой состоит из нескольких нейронов, число которых также может быть произвольно (N_k – количество нейронов в слое), количество входов n , количество выходов $N=N_K$ - числу нейронов в выходном (последнем) слое.

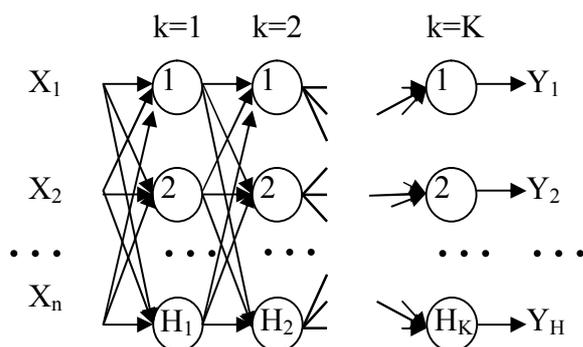


Рис. 6. Многослойная нейронная сеть прямого распространения

Слои между первым и последним называются промежуточными или скрытыми. Веса в такой сети имеют три индекса i - номер нейрона следующего слоя, для которого связь входная, j – номер входа или нейрона текущего слоя, для которого связь выходная, k – номер текущего слоя в нейронной сети (для входов, вектора X , $k=0$).

Многослойные нейронные сети прямого распространения обучаются методом обратного распространения ошибки.

Алгоритм обучения методом обратного распространения ошибки:

1 шаг: инициализация матриц весов случайным образом (в циклах).

2 шаг: предъявление нейронной сети образа (на вход подаются значения из обучающей выборки – вектор X) и берется соответствующий выход (вектор D).

3 шаг (прямой проход): вычисление в циклах выходов всех слоев и получение выходных значений нейронной сети (вектор Y).

$$y_i^k = f\left(\sum_{j=0}^{H_{k-1}} w_{ij}^k \cdot y_j^{k-1}\right),$$

$$y_j^0 = x_j,$$

$$y_0^{k-1} = 1,$$

$$x_0 = 1,$$

где y_i^k - выход i -нейрона k -слоя, f - функция активации, w_{ij}^k - синаптическая связь между j -нейроном слоя $k-1$ и i -нейроном слоя k , x_j - входное значение.

4 шаг (обратный проход): изменение весов в циклах по формулам:

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

$$\delta_i^k = (d_i - y_i) \cdot y_i \cdot (1 - y_i) -$$

для последнего (выходного) слоя,

$$\delta_i^k = y_i \cdot (1 - y_i) \cdot \sum_{l=1}^{H_{k+1}} \delta_l^{k+1} \cdot w_{li}^{k+1} -$$

для промежуточных слоев, где t -номер текущей итерации цикла обучения (номер эпохи), η - коэффициент обучения задается от 0 до 1, y_i^k - выход i -нейрона k -слоя,

w_{ij}^k - синаптическая связь между j -нейроном слоя $k-1$ и i -нейроном слоя k , d_i – желаемое выходное значение на i -нейроне, y_i - реальное значение на i -нейроне выходного слоя.

5 шаг: проверка условия продолжения обучения (вычисление значения ошибки и/или проверка заданного количества итераций). Если обучение не завершено, то 2 шаг, иначе заканчиваем обучение. Среднеквадратичная ошибка вычисляется следующим образом:

$$\varepsilon = \frac{1}{Q} \cdot \sum_{q=1}^Q \sum_{i=1}^H (d_i - y_i)^2,$$

где Q – общее число примеров, H – количество нейронов в выходном слое, d_i – желаемое выходное значение на i -нейроне, y_i – реальное значение на i -нейроне выходного слоя.

Пример решения задачи

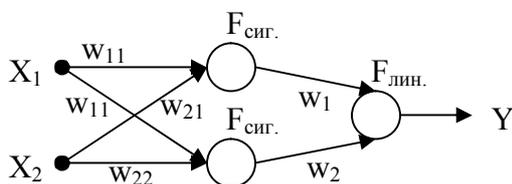
Задача. Просчитать одну итерацию цикла обучения методом обратного распространения ошибки многослойной бинарной неоднородной нейронной сети, состоящей из 2 слоёв, причем в первом слое находится 2 нейрона и используется сигмоидальная функция активации ($k=0,9$), а во втором – 1, линейная ($l=0,7$) функция. В качестве обучающей выборки использовать таблицу истинности для операции «штрих Шеффера» (не использовать первую строчку таблицы). Синаптические веса задать случайным образом.

Описание процесса решения. Для обучения нейронной сети методом обратного распространения ошибки необходимо:

- 1) Графически отобразить структуру нейронной сети. Определить размерность и количество матриц синаптических весов (для каждого слоя своя матрица).
- 2) Определить обучающую выборку, представив ее в табличном виде.
- 3) Выбрать входные данные, на которых будет рассматриваться итерация цикла обучения.
- 4) Следуя алгоритмы обучения методом обратного обучения ошибки просчитать одну итерацию цикла и представить новые синаптические веса в матричном виде.

Решение.

- 1) По заданию нейронная сеть состоит из трех нейронов, два входных, один выходной, значит синаптических весов 6. Первый слой нейронов имеет сигмоидальную функцию активации, второй – линейная.



- 2) По заданию нейронная сеть бинарная, поэтому на ее входы могут подаваться только нули и единицы, так как входа 2, то возможных комбинаций входных значений будет 4 (обучающая выборка будет состоять из 4 векторов). Выход нейронной сети согласно заданию соответствует оператору «штрих Шеффера». Поэтому таблица с обучающей выборкой будет выглядеть следующим образом:

X1	X2	D
0	0	1
0	1	1
1	0	1
1	1	0

- 3) Пусть в качестве вектора обучения будет рассматриваться 2-ая строка таблицы.
- 4) Следуя алгоритму обучения по Δ -правилу, выполним 5 шагов:
1 шаг: зададим матрицу весов случайным образом из интервала $[0,1]$:

Wij(1)	1	2
1	0.6	0.9
2	0.1	0.5

Wg(1)	1	2
	0.3	0.8

2 шаг: вектор $X=\{0,1\}$, $D=\{1\}$.

3 шаг (прямой проход): вычисление в циклах выходов всех слоев и получение выходных значений нейронной сети (вектор Y).

$$S_1 = x_1 \cdot w_{11} + x_2 \cdot w_{21} = 0 \cdot 0.6 + 1 \cdot 0.1 = 0.1;$$

$$S_2 = x_1 \cdot w_{12} + x_2 \cdot w_{22} = 0 \cdot 0.9 + 1 \cdot 0.5 = 0.5;$$

$$k = 0.9;$$

$$Y_1 = \frac{1}{1 + e^{-S_1 \cdot k}} = \frac{1}{1 + e^{-0.1 \cdot 0.9}} = 0.5224,$$

$$Y_2 = \frac{1}{1 + e^{-S_2 \cdot k}} = \frac{1}{1 + e^{-0.5 \cdot 0.9}} = 0.61,$$

$$S_3 = Y_1 \cdot w_1 + Y_2 \cdot w_2 = 0.5224 \cdot 0.3 + 0.61 \cdot 0.8 = 0.64472;$$

$$l = 0.7,$$

$$Y = l \cdot S = 0.4513.$$

4 шаг (обратный проход): изменение весов:

$$\eta = 0.7,$$

$$\delta^2 = (d - Y) \cdot Y \cdot (1 - Y) = (1 - 0.4513) \cdot 0.4513 \cdot (1 - 0.4513) = 0.3587$$

$$w_1(2) = w_1(1) + \eta \cdot \delta^2 \cdot Y_1 = 0.3 + 0.7 \cdot 0.3587 \cdot 0.5224 = 0.431,$$

$$w_2(2) = w_2(1) + \eta \cdot \delta^2 \cdot Y_2 = 0.8 + 0.7 \cdot 0.3587 \cdot 0.61 = 0.953.$$

$$\delta_1^1 = Y_1 \cdot (1 - Y_1) \cdot \sum_{l=1}^{H_{k+1}} \delta_l^{k+1} \cdot w_l^{k+1} = Y_1 \cdot (1 - Y_1) \cdot \delta^2 \cdot w_1 = 0.5224 \cdot (1 - 0.5224) \cdot 0.3587 \cdot 0.3 = 0.0268,$$

$$\delta_2^1 = Y_2 \cdot (1 - Y_2) \cdot \sum_{l=1}^{H_{k+1}} \delta_l^{k+1} \cdot w_l^{k+1} = Y_2 \cdot (1 - Y_2) \cdot \delta^2 \cdot w_2 = 0.61 \cdot (1 - 0.61) \cdot 0.3587 \cdot 0.8 = 0.0682,$$

$$w_{11}(2) = w_{11}(1) + \eta \cdot \delta_1^1 \cdot x_1 = 0.6 + 0.7 \cdot 0.0268 \cdot 0 = 0.6,$$

$$w_{12}(2) = w_{12}(1) + \eta \cdot \delta_2^1 \cdot x_1 = 0.9 + 0.7 \cdot 0.0682 \cdot 0 = 0.9,$$

$$w_{21}(2) = w_{21}(1) + \eta \cdot \delta_1^1 \cdot x_2 = 0.1 + 0.7 \cdot 0.0268 \cdot 1 = 0.119,$$

$$w_{22}(2) = w_{22}(1) + \eta \cdot \delta_2^1 \cdot x_2 = 0.5 + 0.7 \cdot 0.0682 \cdot 1 = 0.548.$$

Wij(2)	1	2
1	0.6	0.9
2	0.119	0.548

Wg(2)	1	2
	0.431	0.953

5 шаг:

$$\varepsilon = \sum_{i=1}^H (d_i - y_i)^2 = (1 - 0.4513)^2 = 0.237.$$

Так как мы рассматриваем одну итерацию цикла обучения, в любом случае выходим из цикла.

Задачи

1. Просчитать одну итерацию цикла обучения методом обратного распространения ошибки многослойной бинарной однородной нейронной сети, состоящей из 2 слоёв, причем в первом слое находится 2 нейрона, а во втором – 1. Функция активации нейронов сети - пороговая ($T=0,6$) функция. В качестве обучающей выборки использовать таблицу истинности для операции «исключающее или» (не использовать первую строчку таблицы). Синаптические веса задать случайным образом.
2. Просчитать одну итерацию цикла обучения методом обратного распространения ошибки многослойной бинарной однородной нейронной сети, состоящей из 2 слоёв, причем в первом слое находится 2 нейрона, а во втором – 1. Функция активации нейронов сети - сигмоидальная ($k=1$) функция. В качестве обучающей выборки использовать таблицу истинности для операции импликации (не использовать первую строчку таблицы). Синаптические веса задать случайным образом.
3. Просчитать одну итерацию цикла обучения методом обратного распространения ошибки многослойной бинарной однородной нейронной сети, состоящей из 2 слоёв, причем в первом слое находится 2 нейрона, а во втором – 1. Функция активации нейронов сети - линейная ($k=0,6$) функция. В качестве обучающей выборки использовать таблицу истинности для операции «штрих Шеффера» (не использовать первую строчку таблицы). Синаптические веса задать случайным образом.
4. Просчитать одну итерацию цикла обучения методом обратного распространения ошибки многослойной бинарной однородной нейронной сети, состоящей из 2 слоёв, причем в первом слое находится

- 2 нейрона, а во втором – 1. Функция активации нейронов сети – гиперболический тангенс ($k=1$). В качестве обучающей выборки использовать таблицу истинности для операции «стрелка Пирса» (не использовать первую строчку таблицы). Синаптические веса задать случайным образом.
5. Просчитать одну итерацию цикла обучения методом обратного распространения ошибки многослойной бинарной неоднородной нейронной сети, состоящей из 2 слоёв, причем в первом слое находится 2 нейрона и используется сигмоидальная функция активации ($k=0,9$), а во втором – 1, пороговая ($T=0,7$). В качестве обучающей выборки использовать таблицу истинности для операции «исключающее или» (не использовать первую строчку таблицы). Синаптические веса задать случайным образом.
 6. Просчитать одну итерацию цикла обучения методом обратного распространения ошибки многослойной бинарной неоднородной нейронной сети, состоящей из 2 слоёв, причем в первом слое находится 2 нейрона и используется линейная функция активации ($k=0,5$), а во втором – 1, сигмоидальная ($k=0,7$) функция. В качестве обучающей выборки использовать таблицу истинности для операции импликации (не использовать первую строчку таблицы). Синаптические веса задать случайным образом.
 7. Просчитать одну итерацию цикла обучения методом обратного распространения ошибки многослойной бинарной неоднородной нейронной сети, состоящей из 2 слоёв, причем в первом слое находится 2 нейрона и используется пороговая функция активации ($T=0,4$), а во втором – 1, линейная ($k=0,6$) функция. В качестве обучающей выборки использовать таблицу истинности для операции «штрих Шеффера» (не использовать первую строчку таблицы). Синаптические веса задать случайным образом.
 8. Просчитать одну итерацию цикла обучения методом обратного распространения ошибки многослойной бинарной неоднородной нейронной сети, состоящей из 2 слоёв, причем в первом слое находится 2 нейрона и используется пороговая функция активации ($T=0,6$), а во втором – 1, гиперболический тангенс ($k=2$). В качестве обучающей выборки использовать таблицу истинности для операции «стрелка Пирса» (не использовать первую строчку таблицы). Синаптические веса задать случайным образом.

9. Просчитать одну итерацию цикла обучения методом обратного распространения ошибки многослойной аналоговой неоднородной нейронной сети, состоящей из 2 слоёв, причем в первом слое находится 3 нейрона, а во втором – 2. Функция активации нейронов сети - линейная ($k=0,6$) функция. Синаптические веса и обучающую выборку задать случайным образом (не нули).
10. Просчитать одну итерацию цикла обучения методом обратного распространения ошибки многослойной аналоговой неоднородной нейронной сети, состоящей из 2 слоёв, причем в первом слое находится 3 нейрона, а во втором – 2. Функция активации нейронов сети - сигмоидальная ($k=1$) функция. Синаптические веса и обучающую выборку задать случайным образом (не нули).
11. Просчитать одну итерацию цикла обучения методом обратного распространения ошибки многослойной аналоговой неоднородной нейронной сети, состоящей из 2 слоёв, причем в первом слое находится 3 нейрона, а во втором – 2. Функция активации нейронов сети - пороговая ($T=0,65$) функция. Синаптические веса и обучающую выборку задать случайным образом (не нули).
12. Просчитать одну итерацию цикла обучения методом обратного распространения ошибки многослойной аналоговой неоднородной нейронной сети, состоящей из 2 слоёв, причем в первом слое находится 3 нейрона, а во втором – 2. Функция активации нейронов сети – гиперболический тангенс ($k=3$) функция. Синаптические веса и обучающую выборку задать случайным образом (не нули).
13. Просчитать одну итерацию цикла обучения методом обратного распространения ошибки многослойной аналоговой неоднородной нейронной сети, состоящей из 2 слоёв, причем в первом слое находится 2 нейрона и используется сигмоидальная функция активации ($k=0,9$), во втором – 2, пороговая ($T=0,7$). Синаптические веса и обучающую выборку задать случайным образом (не нули).
14. Просчитать одну итерацию цикла обучения методом обратного распространения ошибки многослойной аналоговой неоднородной нейронной сети, состоящей из 2 слоёв, причем в первом слое находится 2 нейрона и используется линейная функция активации ($k=0,5$), во втором – 2, сигмоидальная ($k=0,7$) функция. Синаптические веса и обучающую выборку задать случайным образом (не нули).

15. Просчитать одну итерацию цикла обучения методом обратного распространения ошибки многослойной аналоговой неоднородной нейронной сети, состоящей из 2 слоёв, причем в первом слое находится 2 нейрона и используется пороговая функция активации ($T=0,4$), во втором – 2, линейная ($k=0,6$) функция. Синаптические веса и обучающую выборку задать случайным образом (не нули).
16. Просчитать одну итерацию цикла обучения методом обратного распространения ошибки многослойной аналоговой неоднородной нейронной сети, состоящей из 2 слоёв, причем в первом слое находится 2 нейрона и используется пороговая функция активации ($T=0,6$), во втором – 1, гиперболический тангенс ($k=2$). Синаптические веса и обучающую выборку задать случайным образом (не нули).
17. Просчитать одну итерацию цикла обучения методом обратного распространения ошибки многослойной бинарной однородной нейронной сети, состоящей из 3 слоёв, использующей пороговую функцию активации ($T=0,5$), в первом слое 2 нейрона, во втором – 2, в третьем - 1. Синаптические веса и обучающую выборку задать случайным образом (не нули).
18. Просчитать одну итерацию цикла обучения методом обратного распространения ошибки многослойной бинарной однородной нейронной сети, состоящей из 2 слоёв, использующей пороговую функцию активации ($T=0,5$), в первом слое 3 нейрона, во втором – 1. В качестве обучающей выборки использовать таблицу истинности для $X_1 \rightarrow X_2 \& X_3$ (не использовать первую строку таблицы). Синаптические веса задать случайным образом.
19. Просчитать одну итерацию цикла обучения методом обратного распространения ошибки многослойной бинарной однородной нейронной сети, состоящей из 2 слоёв, использующей сигмоидальную функцию активации ($k=0,5$), в первом слое 3 нейрона, во втором – 1. В качестве обучающей выборки использовать таблицу истинности для $(X_1 \rightarrow X_2) \& X_3$ (не использовать первую строку таблицы). Синаптические веса задать случайным образом.
20. Просчитать одну итерацию цикла обучения методом обратного распространения ошибки многослойной аналоговой неоднородной нейронной сети, состоящей из 3 слоёв, причем в первом слое находится 2 нейрона и используется пороговая функция активации ($T=0,6$), во втором – 2, гиперболический тангенс ($k=2$), в третьем 1, линейная

($k=0,7$). Синаптические веса и обучающую выборку задать случайным образом (не нули).

Генетический алгоритм

Генетический алгоритм (англ. genetic algorithm) — это эвристический алгоритм поиска, используемый для решения задач оптимизации и моделирования, основанный на концепциях естественного отбора и генетики.

Для моделирования эволюционных процессов в генетическом алгоритме используются операторы (таблица 10) и стратегии отбора (таблица 11).

Таблица 10. Основные виды операторов генетических алгоритмов.

Оператор	Описание	Пример
Операторы скрещивания		
Одноточечный кроссовер	выбирается одна точка разрыва и родительские хромосомы обмениваются одной из получившихся частей	Родитель 1: 1001011 01001 Родитель 2: 0100011 00111 Потомок 1: 1001011 00111 Потомок 2: 0100011 01001
Двухточечный кроссовер	выбираются две точки разрыва и родительские хромосомы обмениваются сегментом, который находится между двумя этими точками	Родитель 1: 100 101101 001 Родитель 2: 010 001100 111 Потомок 1: 100 001100 001 Потомок 2: 010 101101 111
Равномерный кроссовер	каждый бит первого потомка случайным образом наследуется от одного из родителей, второму потомку достается бит другого родителя	Родитель 1: 100101101001 Родитель 2: 010001100111 Вероятность: 90 % Случайные числа (100): 2, 24, 8, 93, 55, 13, 67, 43, 99, 61, 5, 89 Потомок 1: 100001100001 Потомок 2: 010101101111
Операторы мутации		
Одноточечная мутация	произвольный бит хромосомы с определенной вероятностью изменяется на противоположный	до: 100101100111 после: 100101000111
Транслокация	перенос какого-либо участка хромосомы в другой сегмент этой же хромосомы	до: 100111100111 после: 111000110111
Инверсия	перестановка генов в обратном порядке внутри произвольно выбранного участка хромосомы	до: 100111100111 после: 100100111111

Таблица 11. Виды отбора особей в генетических алгоритмах.

Вид отбора	Описание
Пропорциональный	каждой особи назначает вероятность $P_s(i)$, равную отношению ее приспособленности к суммарной приспособленности популяции,

	осуществляется отбор (с замещением) всех n (устанавливается заранее) особей для дальнейшей генетической обработки, согласно величине $P_s(i)$
Рулетка	вид пропорционального отбора, когда особи отбираются с помощью n «запусков» рулетки (колесо рулетки содержит по одному сектору для каждого члена популяции, размер i -ого сектора пропорционален соответствующей величине $P_s(i)$)
Турнирный	из популяции, содержащей m особей, выбирается случайным образом t особей и выбирается наиболее приспособленная (между выбранными особями проводится турнир), эта операция повторяется m раз
Отбор усечением	из отсортированной в порядке убывания степени приспособленности популяции с учетом порога приспособленности $T \in [0;1]$ (ниже порога особи в отборе не участвуют) случайным образом $m/2$ раз выбираются родительские пары
Ранговый	для каждой особи ее вероятность попасть в промежуточную популяцию пропорциональна ее порядковому номеру в отсортированной по возрастанию приспособленности популяции
Элитный	добавляет к любому другому виду отбора принцип элитизма – сохранения в новой популяции одной или нескольких наиболее приспособленных особей

Перед запуском генетического алгоритма на выполнение необходимо закодировать признаки (параметры, по которым ведется отбор), сформировать из них фенотип, определить фитнес-функцию (критерий приспособленности).

Существует различные виды генетического алгоритма, они отличаются используемыми операторами, видами отбора, а также различают последовательные и параллельные алгоритмы, по все они в той или иной форме содержат следующую последовательность шагов:

- 1 шаг.** Формирование начальной популяции.
- 2 шаг.** Оценка особей популяции (используется фитнес-функция).
- 3 шаг.** Отбор (используется один из методов отбора).
- 4 шаг.** Скрещивание (используется оператор кроссовера).
- 5 шаг.** Мутация (используется один или несколько операторов мутации).
- 6 шаг.** Формирование новой популяции.
- 7 шаг.** Если популяция не сошлась, то 2, иначе – останов (прекращение функционирования генетического алгоритма).

Пример решения задачи

Задача. Описать функционирование одной эпохи генетического алгоритма на примере произвольной задачи (не менее пяти признаков закодировать случайным образом, начальная популяция содержит не менее 10 особей). Использовать следующие параметры генетического алгоритма: фитнес-функция – сумма всех бит, деленная на среднее значение суммы бит особей популяции; метод отбора – рулетка с принципом элитизма; оператор скрещивания – двухточечный кроссовер; оператор мутации – одиночная мутация.

Описание процесса решения. Для использования генетического алгоритма необходимо:

- 1) Определить набор признаков, характеризующие решения задачи оптимизации или моделирования. Определить фенотип, закодировать признаки (можно использовать код Грея).
- 2) Использовать последовательность шагов генетического алгоритма с соответствующими операторами.

Решение.

- 1) Фенотип (задаем десятичные значения случайным образом):

Признак	Двоичное значение признака	Десятичное значение признака	Код Грея
Признак 1	0011	3	0010
Признак 2	1100	12	1010
Признак 3	1110	14	1001
Признак 4	0111	7	0100
Признак 5	1001	9	1101

- 2) **1 шаг.** Формирование начальной популяции.

Определено пять признаков, пусть особь содержит любые 2 из них (два первых - значения первого критерия, три последних второго), случайным образом сгенерируем 10 особей, каждая особь длиной 8 бит:

Особь 1: 00111110

Особь 6: 00111110

Особь 2: 11001110

Особь 7: 11000111

Особь 3: 00111001

Особь 8: 00110111

Особь 4: 11001001

Особь 9: 10101010

Особь 5: 00110111

Особь 10: 01010101

2 шаг. Оценка особей популяции (используется фитнес-функция равная сумме бит в особи).

Особь	Сумма бит в особи	Приспособленность особи
-------	-------------------	-------------------------

1	5	1, 389
2	5	1, 389
3	4	1, 112
4	4	1, 112
5	5	1, 389
6	5	1, 389
7	5	1, 389
8	5	1, 389
9	4	1, 112
10	4	1, 112

Среднее значение суммы бит в популяции = 3,6.

3 шаг. Отбор (используется метод отбора – рулетка с принципом элитизма).

Строим рулетку (сектора пропорциональны приспособленности, рис. 7) и запускаем ее 8 раз (выбираем 4 пары, рис. 9):

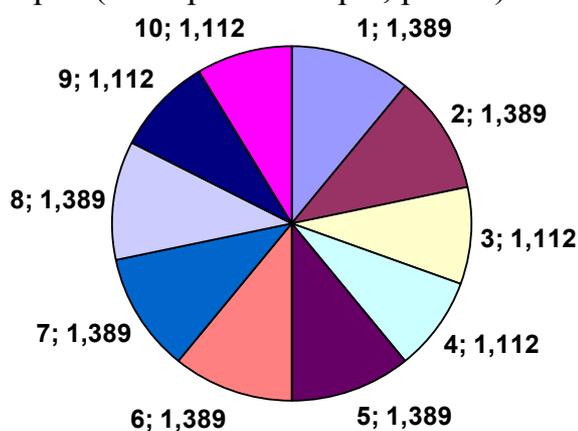


Рис. 7. Рулетка для задачи генетического алгоритма

Запуски рулетки (случайным образом):

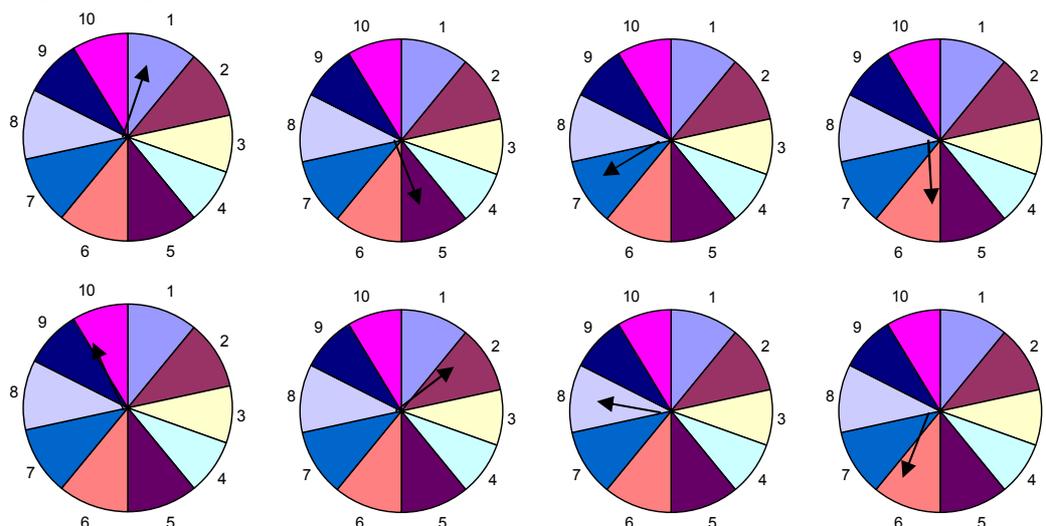


Рис. 8. Запуски рулетки для задачи генетического алгоритма

Таким образом, образовались следующие пары:

1 и 5, 7 и 5, 10 и 2, 8 и 6.

4 шаг. Скрещивание (используется оператор – двухточечный кроссовер).

Выбираем две точки разрыва (случайным образом, но числа должны различаться хотя бы на 2 и не быть равными 1 или длине особи): 2 и 5 и применяем оператор к выбранным парам особей:

Особь 1: 00|111|110 Особь 2.1: 00|110|110
 Особь 5: 00|110|111 Особь 2.2: 00|111|111

Особь 7: 11|000|111 Особь 2.3: 11|110|111
 Особь 5: 00|110|111 Особь 2.4: 00|000|111

Особь 10: 01|010|101 Особь 2.5: 01|001|101
 Особь 2: 11|001|110 Особь 2.6: 11|010|110

Особь 6: 00|111|110 Особь 2.7: 00|110|110
 Особь 8: 00|110|111 Особь 2.8: 00|111|111

5 шаг. Мутация (используется оператор – одноточечная мутация).
 Определим вероятность мутации 30 % и бит – третий, подвергающийся мутации.

Особь	Случайное число	Особь	Мутированная особь
1	5	00111110	00011110
2	33	11001110	
3	67	00111001	
4	78	11001001	
5	90	00110111	
6	12	00111110	00011110
7	45	11000111	
8	53	00110111	
9	74	10101010	
10	29	01010101	01110101

6 шаг. Формирование новой популяции.

Особь 2.1: 00110110 Особь 2.5: 01001101 Особь 2.9: 00011110
 Особь 2.2: 00111111 Особь 2.6: 11010110 Особь 2.10: 00011110
 Особь 2.3: 11110111 Особь 2.7: 00110110 Особь 2.11: 01110101
 Особь 2.4: 00000111 Особь 2.8: 00111111 Особь 2.12: 00111110

1-8 – наследники, 9-11 – матировавшие особи, 12 - сохраняем одну особь с максимальной приспособленностью – принцип элитизма.

7 шаг. Популяция достаточно разнообразна – нет признаков сходимости. Так как рассматривается лишь одна эпоха генетического алгоритма – выход из алгоритма.

Задачи

1. Описать функционирование одной эпохи генетического алгоритма на примере произвольной задачи (не менее пяти признаков закодировать случайным образом, начальная популяция содержит не менее 10 особей). Использовать следующие параметры генетического алгоритма: фитнес-функция – сумма всех бит, деленная на максимум суммы всех бит среди особей популяции; метод отбора – рулетка; оператор скрещивания – одноточечный кроссовер; оператор мутации – одиночная мутация.
2. Описать функционирование одной эпохи генетического алгоритма на примере произвольной задачи (не менее пяти признаков закодировать случайным образом, начальная популяция содержит не менее 10 особей). Использовать следующие параметры генетического алгоритма: фитнес-функция – сумма всех бит, деленная на минимум суммы всех бит среди особей популяции; метод отбора – турнирный отбор; оператор скрещивания – двухточечный кроссовер; оператор мутации – транслокация.
3. Описать функционирование одной эпохи генетического алгоритма на примере произвольной задачи (не менее пяти признаков закодировать случайным образом, начальная популяция содержит не менее 10 особей). Использовать следующие параметры генетического алгоритма: фитнес-функция – единица, деленная на минимум суммы всех бит среди особей популяции; метод отбора – ранговый отбор; оператор скрещивания – равномерный кроссовер; оператор мутации – инверсия.
4. Описать функционирование одной эпохи генетического алгоритма на примере произвольной задачи (не менее пяти признаков закодировать случайным образом, начальная популяция содержит не менее 10 особей). Использовать следующие параметры генетического алгоритма: фитнес-функция – сумма всех бит, умноженная на минимум суммы всех бит среди особей популяции; метод отбора – отбор усечением; оператор скрещивания – равномерный кроссовер; оператор мутации – одноточечная мутация.
5. Описать функционирование одной эпохи генетического алгоритма на примере произвольной задачи (не менее пяти признаков закодировать случайным образом, начальная популяция содержит не менее 10 особей). Использовать следующие параметры генетического

алгоритма: фитнес-функция – единица, деленная на максимум суммы всех бит в особи в популяции; метод отбора – пропорциональный отбор; оператор скрещивания – одноточечный кроссовер; оператор мутации – инверсия.

6. Описать функционирование одной эпохи генетического алгоритма на примере произвольной задачи (не менее пяти признаков закодировать случайным образом, начальная популяция содержит не менее 10 особей). Использовать следующие параметры генетического алгоритма: фитнес-функция – сумма всех бит, деленная на количество бит в особи; метод отбора – рулетка с использованием принципа элитизма; оператор скрещивания – равномерный кроссовер; оператор мутации – инверсия.
7. Описать функционирование одной эпохи генетического алгоритма на примере произвольной задачи (не менее пяти признаков закодировать случайным образом, начальная популяция содержит не менее 10 особей). Использовать следующие параметры генетического алгоритма: фитнес-функция – сумма всех бит особи, деленная на количество бит в особи; метод отбора – пропорциональный с использованием принципа элитизма; оператор скрещивания – одноточечный кроссовер; оператор мутации – одноточечная мутация.
8. Описать функционирование одной эпохи генетического алгоритма на примере произвольной задачи (не менее пяти признаков закодировать случайным образом, начальная популяция содержит не менее 10 особей). Использовать следующие параметры генетического алгоритма: фитнес-функция – сумма всех бит особи, деленная на количество особей в популяции; метод отбора – ранговый с использованием принципа элитизма; оператор скрещивания – одноточечный кроссовер; оператор мутации – одноточечная мутация.
9. Описать функционирование одной эпохи генетического алгоритма на примере произвольной задачи (не менее пяти признаков закодировать случайным образом, начальная популяция содержит не менее 10 особей). Использовать следующие параметры генетического алгоритма: фитнес-функция – сумма всех бит особи, деленная на количество особей в популяции; метод отбора – турнирный с использованием принципа элитизма; оператор скрещивания – равномерный кроссовер; оператор мутации – транслокация.

10. Описать функционирование одной эпохи генетического алгоритма на примере произвольной задачи (не менее пяти признаков закодировать случайным образом, начальная популяция содержит не менее 10 особей). Использовать следующие параметры генетического алгоритма: фитнес-функция – сумма всех бит особи, деленная на количество особей в популяции; метод отбора – отбор усечением с использованием принципа элитизма; оператор скрещивания – двухточечный кроссовер; оператор мутации – транслокация.
11. Описать функционирование одной эпохи генетического алгоритма на примере произвольной задачи (не менее пяти признаков закодировать случайным образом, начальная популяция содержит не менее 10 особей). Использовать следующие параметры генетического алгоритма: фитнес-функция – сумма всех бит, деленная на максимум суммы всех бит особи в популяции; метод отбора – рулетка; оператор скрещивания – двухточечный кроссовер; оператор мутации – одиночная мутация.
12. Описать функционирование одной эпохи генетического алгоритма на примере произвольной задачи (не менее пяти признаков закодировать случайным образом, начальная популяция содержит не менее 10 особей). Использовать следующие параметры генетического алгоритма: фитнес-функция – сумма всех бит особи, деленная на максимум суммы всех бит особи в популяции; метод отбора – турнирный отбор; оператор скрещивания – равномерный кроссовер; оператор мутации – инверсия.
13. Описать функционирование одной эпохи генетического алгоритма на примере произвольной задачи (не менее пяти признаков закодировать случайным образом, начальная популяция содержит не менее 10 особей). Использовать следующие параметры генетического алгоритма: фитнес-функция – единица, деленная на минимум суммы всех бит особи в популяции; метод отбора – ранговый отбор; оператор скрещивания – одноточечный кроссовер; оператор мутации – инверсия.
14. Описать функционирование одной эпохи генетического алгоритма на примере произвольной задачи (не менее пяти признаков закодировать случайным образом, начальная популяция содержит не менее 10 особей). Использовать следующие параметры генетического алгоритма: фитнес-функция – сумма всех бит, умноженная на минимум суммы всех бит особи в популяции; метод отбора – отбор

- усечением; оператор скрещивания – равномерный кроссовер; оператор мутации – транслокация.
15. Описать функционирование одной эпохи генетического алгоритма на примере произвольной задачи (не менее пяти признаков закодировать случайным образом, начальная популяция содержит не менее 10 особей). Использовать следующие параметры генетического алгоритма: фитнес-функция – единица, деленная на максимум суммы всех бит среди особей популяции; метод отбора – пропорциональный отбор; оператор скрещивания – одноточечный кроссовер; оператор мутации – транслокация.
 16. Описать функционирование одной эпохи генетического алгоритма на примере произвольной задачи (не менее пяти признаков закодировать случайным образом, начальная популяция содержит не менее 10 особей). Использовать следующие параметры генетического алгоритма: фитнес-функция – сумма всех бит особи, деленная на количество бит в особи; метод отбора – рулетка с использованием принципа элитизма; оператор скрещивания – одноточечный кроссовер; оператор мутации – транслокация.
 17. Описать функционирование одной эпохи генетического алгоритма на примере произвольной задачи (не менее пяти признаков закодировать случайным образом, начальная популяция содержит не менее 10 особей). Использовать следующие параметры генетического алгоритма: фитнес-функция – сумма всех бит особи, деленная на количество бит в особи; метод отбора – пропорциональный с использованием принципа элитизма; оператор скрещивания – двухточечный кроссовер; оператор мутации – инверсия.
 18. Описать функционирование одной эпохи генетического алгоритма на примере произвольной задачи (не менее пяти признаков закодировать случайным образом, начальная популяция содержит не менее 10 особей). Использовать следующие параметры генетического алгоритма: фитнес-функция – сумма всех бит особи, деленная на количество особей в популяции; метод отбора – ранговый с использованием принципа элитизма; оператор скрещивания – равномерный кроссовер; оператор мутации – транслокация.
 19. Описать функционирование одной эпохи генетического алгоритма на примере произвольной задачи (не менее пяти признаков закодировать случайным образом, начальная популяция содержит не менее 10

особей). Использовать следующие параметры генетического алгоритма: фитнес-функция – сумма всех бит особи, деленная на количество особей в популяции; метод отбора – турнирный с использованием принципа элитизма; оператор скрещивания – равномерный кроссовер; оператор мутации – одноточечная мутация.

20. Описать функционирование одной эпохи генетического алгоритма на примере произвольной задачи (не менее пяти признаков закодировать случайным образом, начальная популяция содержит не менее 10 особей). Использовать следующие параметры генетического алгоритма: фитнес-функция – сумма всех бит особи, деленная на количество особей в популяции; метод отбора – отбор усечением с использованием принципа элитизма; оператор скрещивания – двухточечный кроссовер; оператор мутации – одноточечная мутация.

Нечеткие множества и нечеткая логика

Нечетким множеством A называется совокупность пар $A = \{ \langle x, \mu_A(x) \rangle | x \in U \}$, где μ_A — функция принадлежности, т.е. $\mu_A : U \rightarrow [0,1]$, характеристическая функция множества $A \subseteq U$, значения которой указывают, является ли $x \in U$ элементом множества A , U - так называемое универсальное множество, из элементов которого образованы все остальные множества, рассматриваемые в данном классе задач.

Значение $\mu_A(x)$ называется степенью принадлежности элемента x нечеткому множеству A .

Операции над нечеткими множествами

Аналогично четким множествам над нечеткими множествами можно производить ряд операций, которые могут определяться 3 способами (таблица 12).

Таблица 12. Виды определений операций над нечеткими множествами.

Максиминные	$\mu_{A \cup B}(x) = \max \{ \mu_A(x), \mu_B(x) \},$ $\mu_{A \cap B}(x) = \min \{ \mu_A(x), \mu_B(x) \}.$
Алгебраические	$\mu_{A \cup B}(x) = \mu_A(x) + \mu_B(x) - \mu_A(x) \cdot \mu_B(x),$ $\mu_{A \cap B}(x) = \mu_A(x) \cdot \mu_B(x).$
Ограниченные	$\mu_{A \cup B}(x) = \min \{ 1, \mu_A(x) + \mu_B(x) \},$ $\mu_{A \cap B}(x) = \max \{ 0, \mu_A(x) + \mu_B(x) - 1 \}.$

Дополнение нечеткого множества во всех трех случаях определяется одинаково:

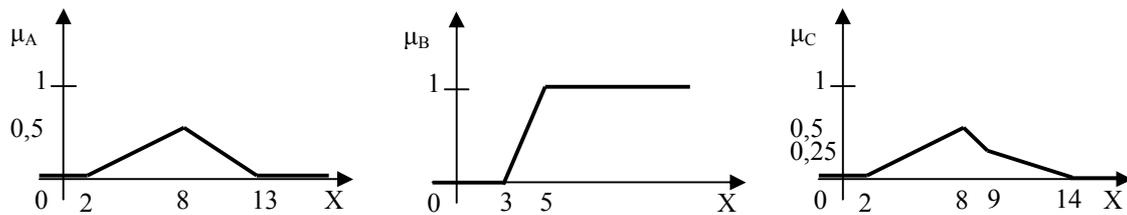
$$\mu_{\bar{A}}(x) = 1 - \mu_A(x).$$

При графическом определении функций принадлежности объединенного множества необходимо в каждой точке множества выбрать максимальное значение из двух (точку того графика, который выше) и объединить все полученные точки в график, который и будет отображением новой функции принадлежности. Пересечение аналогично объединению, только выбирается минимальное значение в каждой точке. При построении дополнения необходимо зеркально отобразить график от оси, параллельной оси абсцисс и проходящей через точку 0,5 оси ординат.

Пример решения задачи

Задача. Дано 3 нечетких множества A, B, C (заданы их функции принадлежности). Построить функцию принадлежности нечеткого

множества $D = \bar{A} \cap (A \cup C \cup B)$ и определить степень принадлежности одного элемента множеству D , используя метод ограничений.

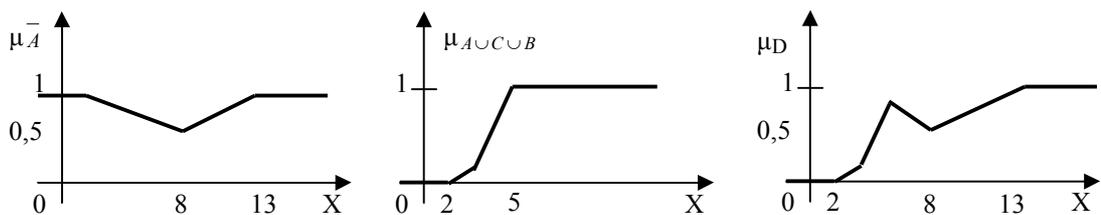


Описание процесса решения. Для построения функции принадлежности нового множества необходимо:

- 1) Определить последовательность выполнения операций в формуле.
- 2) Построить на отдельных графиках промежуточные множества, согласно определенной последовательности действий. Свести промежуточные множества на одном графике и определить итоговую функцию принадлежности.
- 3) Используя определенный в задаче метод, определить аналитически степень принадлежности элемента, входящего в ядро итогового множества.
- 4) Проверить аналитические вычисления по построенному графику функции принадлежности.

Решение.

- 1) Множество $D = \bar{A} \cap (A \cup C \cup B)$, значит, последовательность операций будет следующей: \bar{A} , $A \cup C \cup B$, $\bar{A} \cap (A \cup C \cup B)$.
- 2) Построим согласно этой последовательности операций графики функций принадлежности:



- 3) Ядро множества D состоит из элементов из интервала $(2,13)$. Выберем элемент 8.

$$\mu_A(8) = 0.5;$$

$$\mu_B(8) = 1;$$

$$\mu_C(8) = 0.5;$$

$$\mu_{\bar{A}}(8) = 1 - \mu_A(8) = 1 - 0.5 = 0.5;$$

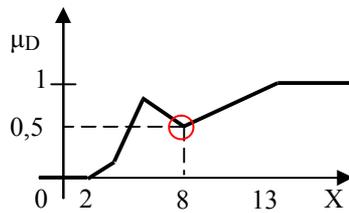
$$\mu_{\bar{C}}(8) = 1 - \mu_C(8) = 1 - 0.5 = 0.5;$$

$$\mu_{A \cup C}(8) = \min\{1, \mu_C(8) + \mu_A(8)\} = \min\{1, 0.5 + 0.5\} = 1;$$

$$\mu_{A \cup C \cup B}(8) = \min\{1, \mu_{A \cup C}(8) + \mu_B(8)\} = \min\{1, 1+1\} = 1;$$

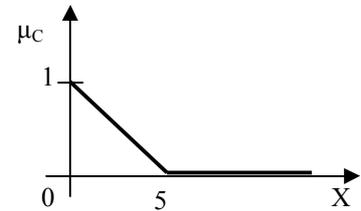
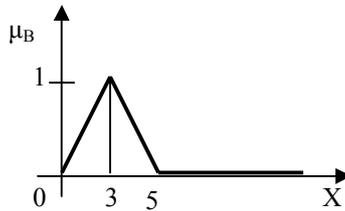
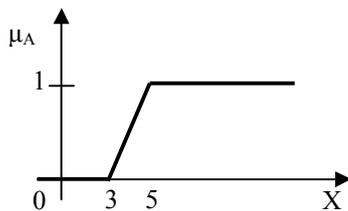
$$\mu_{A \cap (A \cup C \cup B)}(8) = \max\{0, \mu_A(8) + \mu_{A \cup C \cup B}(8) - 1\} = \max\{0, 0.5 + 1 - 1\} = 0.5.$$

4) $\mu_D(8) = 0.5;$

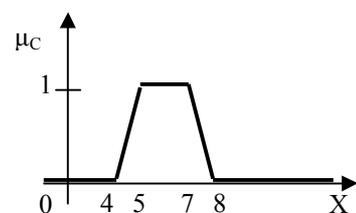
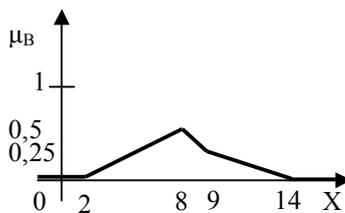
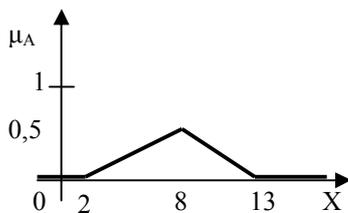


Задачи

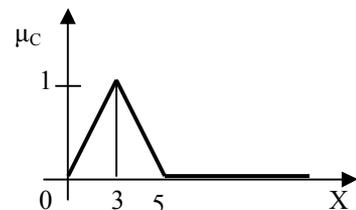
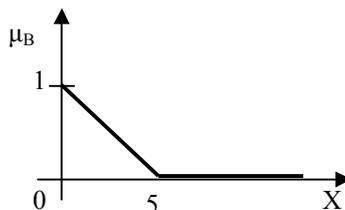
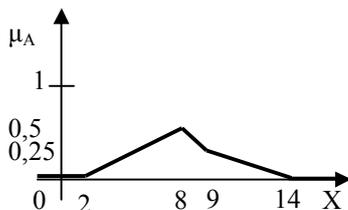
- 1) Дано 3 нечетких множества A, B, C (заданы их функции принадлежности). Построить функцию принадлежности нечеткого множества $D = A \cup B \cap C$ и определить степень принадлежности одного элемента множеству D, используя максиминный способ.



- 2) Дано 3 нечетких множества A, B, C (заданы их функции принадлежности). Построить функцию принадлежности нечеткого множества $D = A \cup B \cap C$ и определить степень принадлежности одного элемента множеству D, используя алгебраический способ.

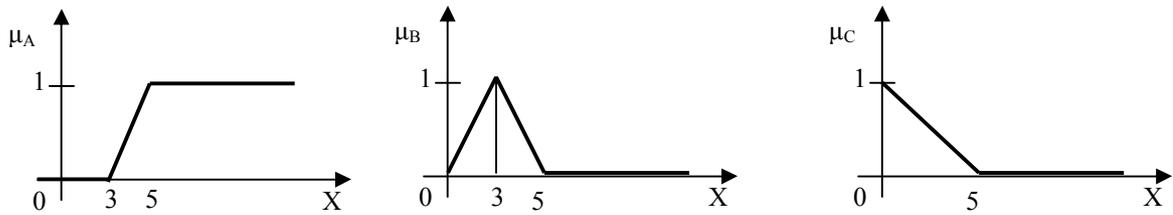


- 3) Дано 3 нечетких множества A, B, C (заданы их функции принадлежности). Построить функцию принадлежности нечеткого множества $D = A \cup B \cap C$ и определить степень принадлежности одного элемента множеству D, используя метод ограничений.

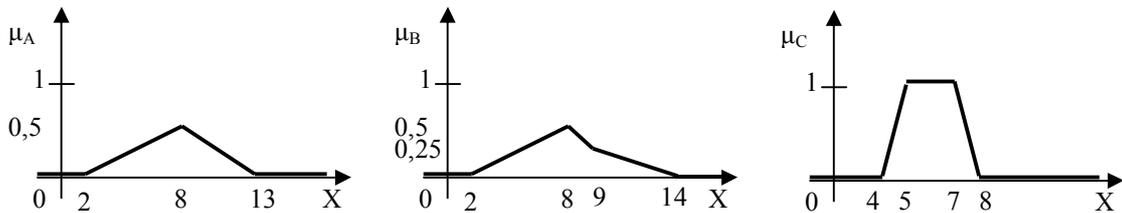


- 4) Дано 3 нечетких множества A, B, C (заданы их функции принадлежности). Построить функцию принадлежности нечеткого

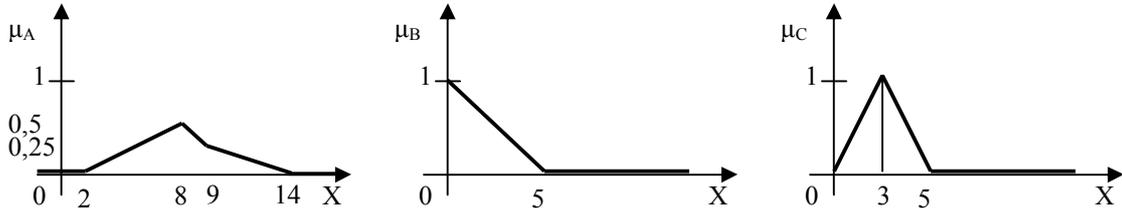
множества $D = A \cup \bar{B} \cap C$ и определить степень принадлежности одного элемента множеству D , используя максиминный способ.



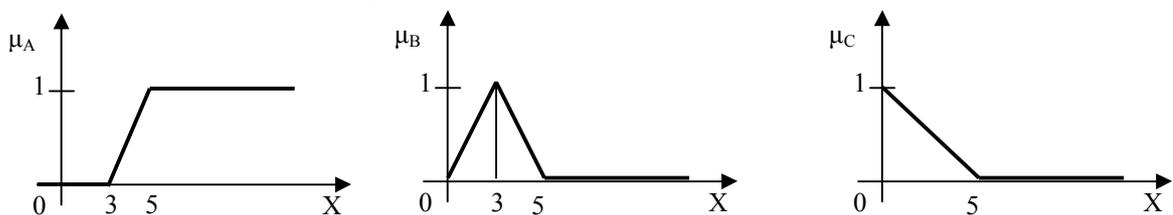
- 5) Дано 3 нечетких множества A , B , C (заданы их функции принадлежности). Построить функцию принадлежности нечеткого множества $D = A \cup \bar{B} \cap C$ и определить степень принадлежности одного элемента множеству D , используя алгебраический способ.



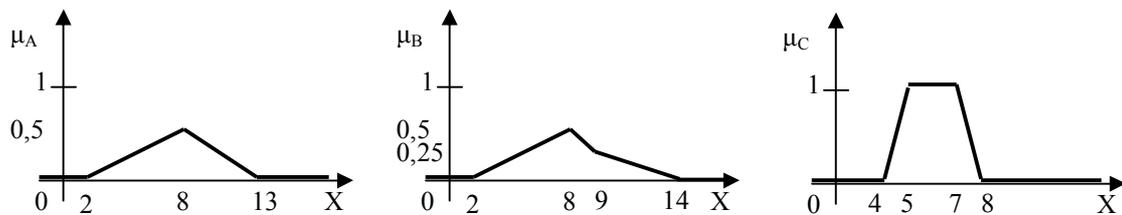
- 6) Дано 3 нечетких множества A , B , C (заданы их функции принадлежности). Построить функцию принадлежности нечеткого множества $D = A \cup \bar{B} \cap C$ и определить степень принадлежности одного элемента множеству D , используя метод ограничений.



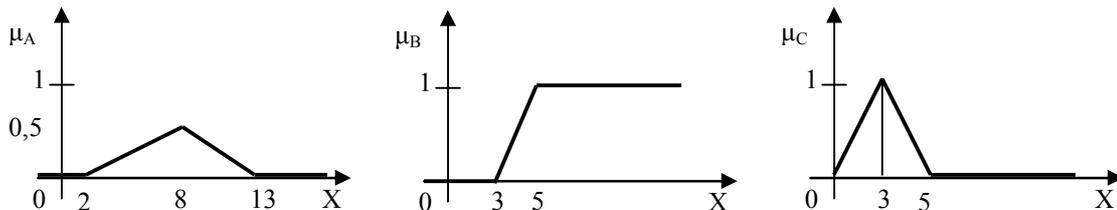
- 7) Дано 3 нечетких множества A , B , C (заданы их функции принадлежности). Построить функцию принадлежности нечеткого множества $D = A \cap B \cup \bar{C}$ и определить степень принадлежности одного элемента множеству D , используя максиминный способ.



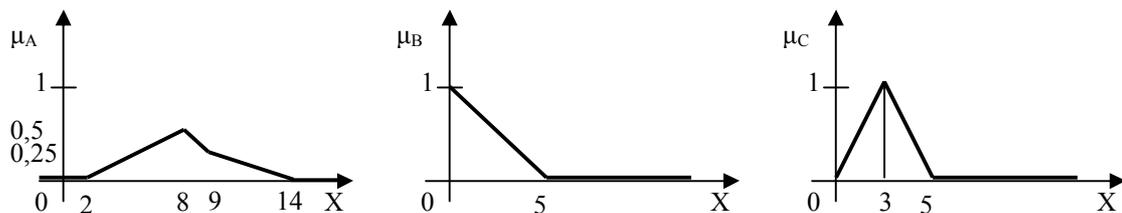
- 8) Дано 3 нечетких множества A , B , C (заданы их функции принадлежности). Построить функцию принадлежности нечеткого множества $D = A \cap B \cup \bar{C}$ и определить степень принадлежности одного элемента множеству D , используя алгебраический способ.



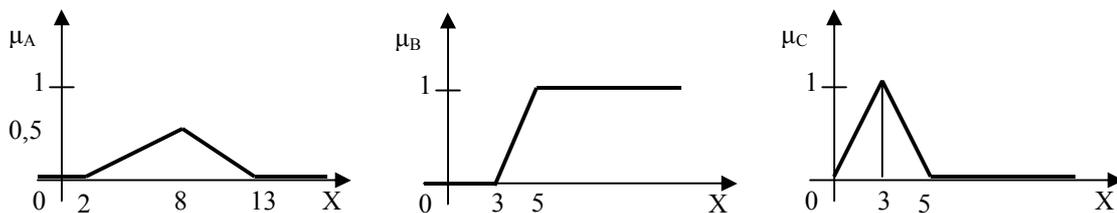
- 9) Дано 3 нечетких множества A , B , C (заданы их функции принадлежности). Построить функцию принадлежности нечеткого множества $D = A \cap \bar{B} \cup \bar{C}$ и определить степень принадлежности одного элемента множеству D , используя метод ограничений.



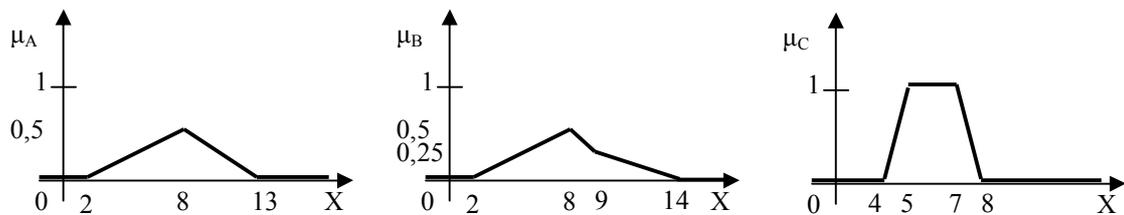
- 10) Дано 3 нечетких множества A , B , C (заданы их функции принадлежности). Построить функцию принадлежности нечеткого множества $D = \bar{A} \cup B \cap C$ и определить степень принадлежности одного элемента множеству D , используя максиминный способ.



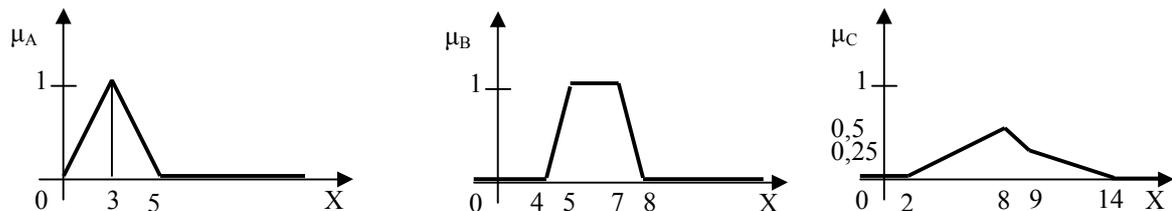
- 11) Дано 3 нечетких множества A , B , C (заданы их функции принадлежности). Построить функцию принадлежности нечеткого множества $D = \bar{A} \cup B \cap C$ и определить степень принадлежности одного элемента множеству D , используя алгебраический способ.



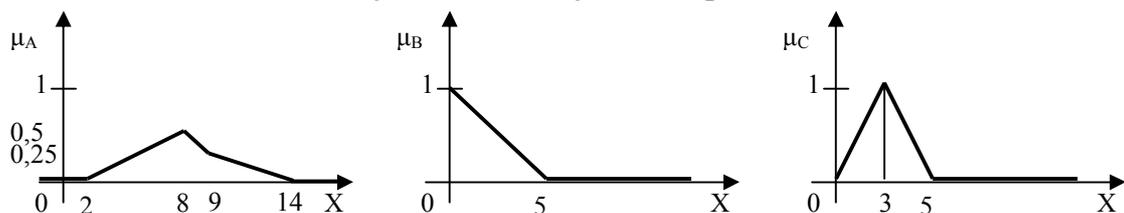
- 12) Дано 3 нечетких множества A , B , C (заданы их функции принадлежности). Построить функцию принадлежности нечеткого множества $D = \bar{A} \cup B \cap C$ и определить степень принадлежности одного элемента множеству D , используя метод ограничений.



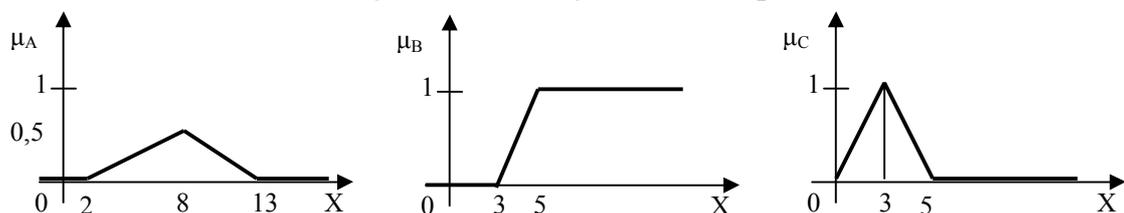
- 13) Дано 3 нечетких множества A , B , C (заданы их функции принадлежности). Построить функцию принадлежности нечеткого множества $D = \overline{(A \cup B)} \cap \overline{C}$ и определить степень принадлежности одного элемента множеству D , используя максиминный способ.



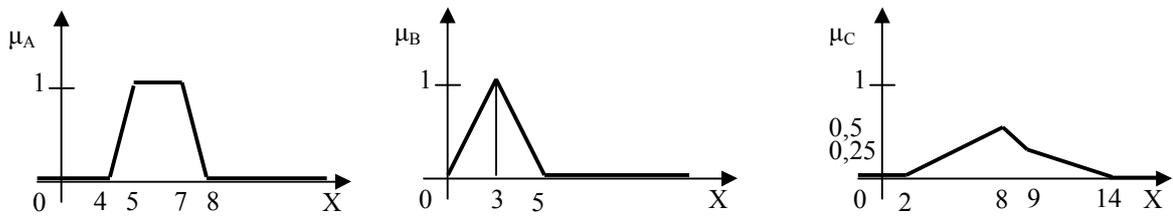
- 14) Дано 3 нечетких множества A , B , C (заданы их функции принадлежности). Построить функцию принадлежности нечеткого множества $D = \overline{(A \cup B)} \cap \overline{C}$ и определить степень принадлежности одного элемента множеству D , используя алгебраический способ.



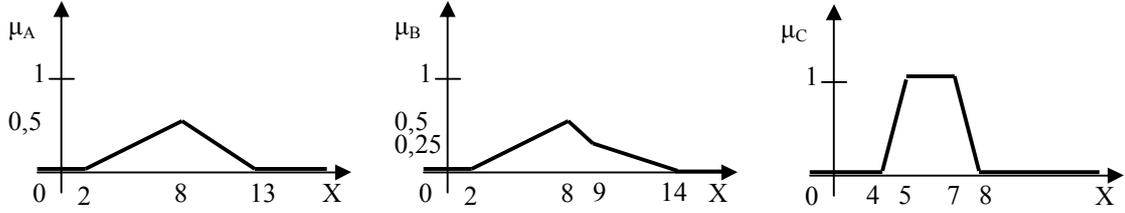
- 15) Дано 3 нечетких множества A , B , C (заданы их функции принадлежности). Построить функцию принадлежности нечеткого множества $D = \overline{(A \cup B)} \cap \overline{C}$ и определить степень принадлежности одного элемента множеству D , используя метод ограничений.



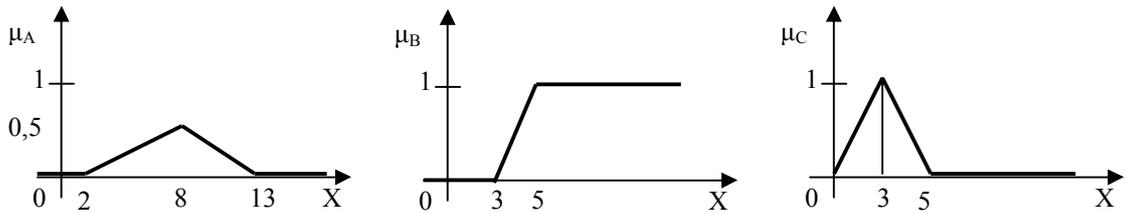
- 16) Дано 3 нечетких множества A , B , C (заданы их функции принадлежности). Построить функцию принадлежности нечеткого множества $D = \overline{A} \cap (C \cup B) \cap \overline{C}$ и определить степень принадлежности одного элемента множеству D , используя максиминный способ.



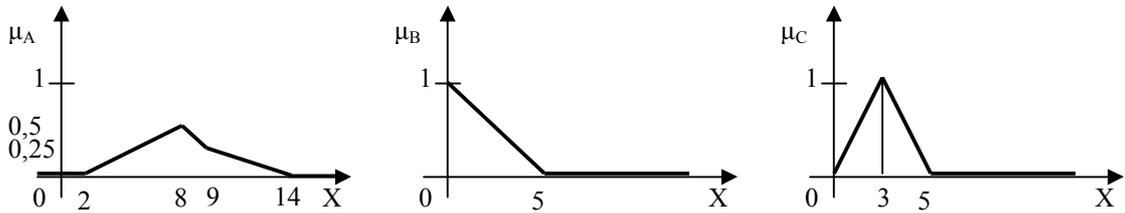
17) Дано 3 нечетких множества A, B, C (заданы их функции принадлежности). Построить функцию принадлежности нечеткого множества $D = \bar{A} \cap (C \cup B) \cap \bar{C}$ и определить степень принадлежности одного элемента множеству D, используя алгебраический способ.



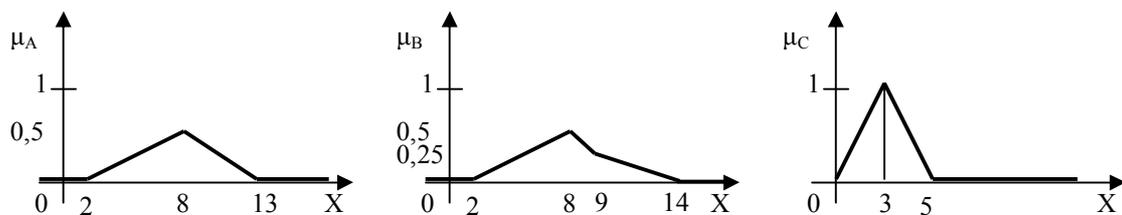
18) Дано 3 нечетких множества A, B, C (заданы их функции принадлежности). Построить функцию принадлежности нечеткого множества $D = \bar{A} \cap (C \cup B) \cap \bar{C}$ и определить степень принадлежности одного элемента множеству D, используя метод ограничений.



19) Дано 3 нечетких множества A, B, C (заданы их функции принадлежности). Построить функцию принадлежности нечеткого множества $D = \bar{A} \cup \bar{B} \cap \bar{C}$ и определить степень принадлежности одного элемента множеству D, используя алгебраический способ.



20) Дано 3 нечетких множества A, B, C (заданы их функции принадлежности). Построить функцию принадлежности нечеткого множества $D = \bar{A} \cup \bar{B} \cap \bar{C}$ и определить степень принадлежности одного элемента множеству D, используя метод ограничений.



Нечеткий вывод

Нечеткая импликация выражается в следующем виде:

$$A \rightarrow B = \overline{A \cup B} \text{ и } \mu_{A \rightarrow B}(x) = \max\{1 - \mu_A(x), \mu_B(x)\}.$$

Основой для проведения операции нечеткого логического вывода является база правил, содержащая нечеткие высказывания в форме "Если-то" и функции принадлежности для соответствующих лингвистических термов. При этом должны соблюдаться следующие условия:

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

В противном случае имеет место неполная база нечетких правил.

Нечеткими высказываниями называются:

1. Высказывание $\langle \beta \text{ есть } \beta' \rangle$, где β - наименование лингвистической переменной, β' - ее значение, которому соответствует нечеткое множество на универсальном множестве X .
2. Высказывание $\langle \beta \text{ есть } m\beta' \rangle$, где m - модификатор, которому соответствуют слова «очень», «более или менее», «много больше» и др.
3. Составные высказывания, образованные из высказываний видов 1. и 2. и союзов "И", "ИЛИ", "ЕСЛИ.., ТО...", "ЕСЛИ.., ТО.., ИНАЧЕ".

Лингвистической переменной называется набор $\langle \beta, T, X, G, M \rangle$, где β - наименование лингвистической переменной, T - множество ее значений (терм-множество), представляющих собой наименования нечетких переменных, областью определения каждой из которых является множество X (множество T называется базовым терм-множеством лингвистической переменной), G - синтаксическая процедура, позволяющая оперировать элементами терм-множества T , в частности, генерировать новые термы (значения), M - семантическая процедура, позволяющая превратить каждое новое значение лингвистической переменной, образуемое процедурой G , в нечеткую переменную, т.е. сформировать соответствующее нечеткое множество.

Нечеткая переменная характеризуется тройкой $\langle \alpha, X, A \rangle$, где α - наименование переменной, X - универсальное множество (область определения α), A - нечеткое множество на X , описывающее ограничения (т.е. $\mu_A(x)$) на значения нечеткой переменной α .

Пример решения задачи

Задача. Построить нечеткую базу знаний (использовать не менее 3 лингвистических переменных) для задачи определения временных затрат для решения студентом задач данного пособия (учитывать успеваемость студента и количество решаемых вариантов), проверить ее на полноту и произвести нечеткий вывод для конкретных значений (выбрать случайным образом).

Описание процесса решения. Для реализации логического вывода необходимо выполнить следующее:

- 1) Сформулировать на естественном языке в виде предложений «Если..., то» закономерности предметной области.
- 2) Выделить из этих предложений лингвистические переменные, их значения (построить их функции принадлежности), высказывания различных видов, формализовать формализовать нечеткие правила.
- 3) Проверить полученную базу знаний на полноту.
- 4) Провести фаззификацию (входные данные выбираем случайным образом).
- 5) Провести аккумуляцию.
- 6) Провести дефаззификацию.

Решение.

- 1) Предложения, описывающие задачу следующие:
 - Если успеваемость студента высокая или хорошая и он прорешивает малое количество вариантов, то ему требуется немного времени.
 - Если успеваемость студента высокая или хорошая и он прорешивает много вариантов, то ему требуется достаточно большой промежуток времени.
 - Если успеваемость студента низкая и он прорешивает много вариантов, то ему требуется много времени.
 - Если успеваемость студента средняя и он прорешивает достаточно большое количество вариантов, то ему требуется достаточно большой промежуток времени.

Выделить из этих предложений лингвистические переменные (определим их через формальную запись $\langle \beta, T, X, G, M \rangle$):

1. β =успеваемость студента, T =(«высокая», «средняя», «низкая»), X =[2,5] (используется пятибалльная система), G =(«очень низкая», «высокая или

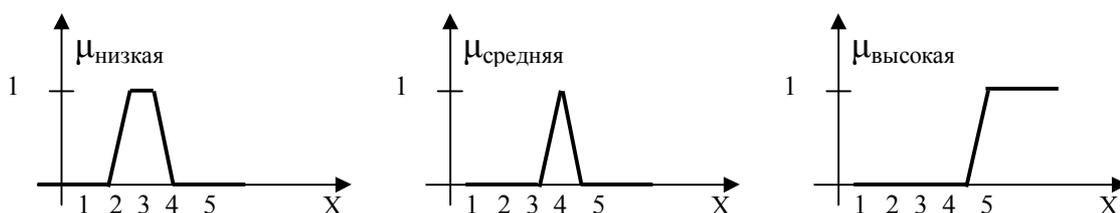
средняя)), M – уменьшение на единицу степени принадлежности нечеткой переменной «высокая», операция объединения нечетких множеств;

2. β = количество вариантов, T = («мало», «достаточно», «много»), $X = [1, 20]$ (количество вариантов 20 в каждой теме), G = («очень много», «достаточно или мало»), M – увеличение на единицу степени принадлежности нечеткой переменной «много», операция объединения нечетких множеств;

3. β = количество времени, T = («мало», «достаточно», «много»), $X = [1, 7]$ (количество часов в неделю, уделенных предмету изучения), G = («очень много», «достаточно или мало»), M – увеличение на единицу степени принадлежности нечеткой переменной «много», операция объединения нечетких множеств;

Для полного задания лингвистической переменной необходимо определить нечеткие переменные, входящие в T :

Успеваемость:



Количество вариантов:



Количество времени:



С учетом выделенных лингвистических переменных, нечеткие правила следующие:

1. Если Успеваемость = «высокая» или Успеваемость = «средняя» и Количество вариантов = «мало», то Количество времени = «мало».
2. Если Успеваемость = «высокая» или Успеваемость = «средняя» и Количество вариантов = «много», то Количество времени = «достаточно».
3. Если Успеваемость = «низкая» и Количество вариантов = «много», то Количество времени = «много».
4. Если Успеваемость = «средняя» и Количество вариантов = «достаточно», то Количество времени = «достаточно».

2) Проверим полученную базу на полноту:

- существует хотя бы одно правило для каждого лингвистического термина выходной переменной – выходная переменная «Количество времени» имеет 3 термина: «мало» используется в 1 правиле, «достаточно» - в 2 и 4, «много» - в третьем;
- для любого термина входной переменной имеется хотя бы одно правило, в котором этот терм используется в качестве предпосылки - есть две входных переменных «Успеваемость» и «Количество вариантов» у каждой из них 3 термина: «высокая» используется в 1 и 2 правиле, «средняя» - 1, 2 и 4, «низкая» - в 3, «мало» - в 1, «достаточно» - 4, «много» - 3 и 2.

Значит, полученная база нечетких правил полная.

- 3) Пусть имеется студент Иванов А.А., имеющий среднюю оценку 3,5 и решивший прорешать 9 вариантов, нужно определить сколько ему понадобится времени.

Определим степени уверенности простейших утверждений:

Успеваемость = «высокая» - 0;

Успеваемость = «средняя» - 0.5;

Успеваемость = «низкая» - 1;

Количество вариантов = «мало» - 0;

Количество вариантов = «достаточно» - 0.5;

Количество вариантов = «много» - 0.125.

Определим степени уверенности посылок правил:

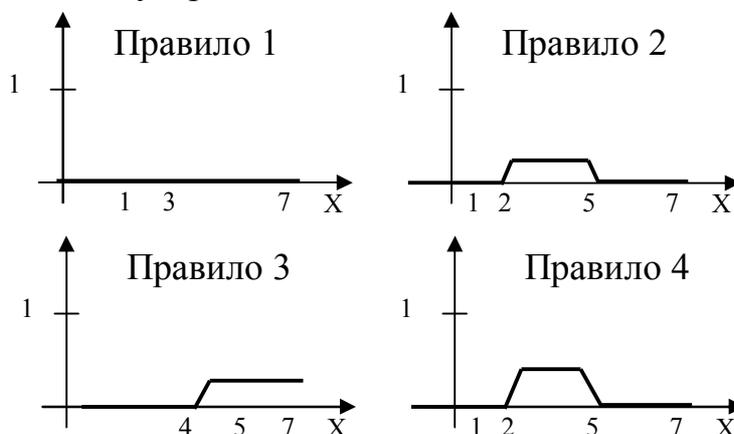
Правило 1: $\min(\max(0, 0.5), 0) = 0$;

Правило 2: $\min(\max(0, 0.5), 0.125) = 0.125$;

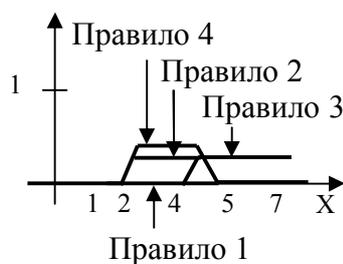
Правило 3: $\min(1, 0.125) = 0.125$;

Правило 4: $\min(0.5, 0.5) = 0.5$.

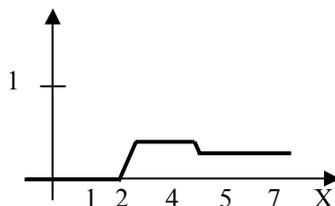
Построим новую выходную нечеткую переменную, используя полученные степени уверенности:



- 4) Аккумуляция:



Новый терм выходной переменной Количество часов:



- 5) Исходя из полученного графика степени принадлежности выходного термина, можно сказать, что Иванову А.А., имеющему среднюю оценку 3.5, на решение 9 вариантов заданий понадобится не менее 2.75 часа (степень уверенности данного утверждения 0.5).

Задачи

- 1) Построить нечеткую базу знаний (использовать не менее 3 лингвистических переменных) для задачи закупок (соотношения цены, качества, объема закупок и т.д.), проверить ее на полноту и произвести нечеткий вывод для конкретных значений (выбрать случайным образом).
- 2) Построить нечеткую базу знаний (использовать не менее 3 лингвистических переменных) для задачи распределения нагрузок спортсмена (соотношение нагрузок, физического состояния, потребляемых калорий и т.д.), проверить ее на полноту и произвести нечеткий вывод для конкретных значений (выбрать случайным образом).
- 3) Построить нечеткую базу знаний (использовать не менее 3 лингвистических переменных) для задачи управления транспортным средством (регулировка скорости с учетом передачи, погодных условий, интенсивности потока и т.д.), проверить ее на полноту и произвести нечеткий вывод для конкретных значений (выбрать случайным образом).
- 4) Построить нечеткую базу знаний (использовать не менее 3 лингвистических переменных) для задачи управления транспортным средством (управление рулем, газом, тормозом при въезде в гараж),

- проверить ее на полноту и произвести нечеткий вывод для конкретных значений (выбрать случайным образом).
- 5) Построить нечеткую базу знаний (использовать не менее 3 лингвистических переменных) для задачи регулирования теплоснабжения (соотношение среднесуточной температуры, ветра, размера здания и т.д.), проверить ее на полноту и произвести нечеткий вывод для конкретных значений (выбрать случайным образом).
 - 6) Построить нечеткую базу знаний (использовать не менее 3 лингвистических переменных) для задачи регулирования реверсного движения на волжском мосту (учитывать время, интенсивность потока, день недели и т.д.), проверить ее на полноту и произвести нечеткий вывод для конкретных значений (выбрать случайным образом).
 - 7) Построить нечеткую базу знаний (использовать не менее 3 лингвистических переменных) для задачи подбора специй для блюда (соотношение количества и остроты специй, рецептуры, предпочтений едока, объема пищи и т.д.), проверить ее на полноту и произвести нечеткий вывод для конкретных значений (выбрать случайным образом).
 - 8) Построить нечеткую базу знаний (использовать не менее 3 лингвистических переменных) для задачи подбора объема блюд (учитывать калорийность, вкусовые предпочтения, количество едоков и т.д.), проверить ее на полноту и произвести нечеткий вывод для конкретных значений (выбрать случайным образом).
 - 9) Построить нечеткую базу знаний (использовать не менее 3 лингвистических переменных) для задачи подачи электроэнергии в условиях экономии (учет времени суток, типа помещений, количества людей, типа оборудования и т.д.), проверить ее на полноту и произвести нечеткий вывод для конкретных значений (выбрать случайным образом).
 - 10) Построить нечеткую базу знаний (использовать не менее 3 лингвистических переменных) для задачи подбора интенсивности занятий (учитывать начальный уровень подготовки, объем учебного материала, количество человек в группе, необходимый уровень усвоения и т.д.), проверить ее на полноту и произвести нечеткий вывод для конкретных значений (выбрать случайным образом).
 - 11) Построить нечеткую базу знаний (использовать не менее 3 лингвистических переменных) для задачи расчета потребления бензина

- (учитывать тип совершаемых маневров, уровень подготовки водителя, состояние автомобиля, тип автомобиля и т.д.), проверить ее на полноту и произвести нечеткий вывод для конкретных значений (выбрать случайным образом).
- 12) Построить нечеткую базу знаний (использовать не менее 3 лингвистических переменных) для задачи регулирования системы орошения (учитывать время года, количество выпадающих осадков, вид орошаемой культуры и т.д.), проверить ее на полноту и произвести нечеткий вывод для конкретных значений (выбрать случайным образом).
 - 13) Построить нечеткую базу знаний (использовать не менее 3 лингвистических переменных) для задачи настройки аудиосистемы (мощность колонок, их количество, размер помещения, назначение установки и т.д.), проверить ее на полноту и произвести нечеткий вывод для конкретных значений (выбрать случайным образом).
 - 14) Построить нечеткую базу знаний (использовать не менее 3 лингвистических переменных) для задачи выбора дозы снотворного (количество препарата, действие препарата, восприимчивость к выбранному препарату, цель и т.д.), проверить ее на полноту и произвести нечеткий вывод для конкретных значений (выбрать случайным образом).
 - 15) Построить нечеткую базу знаний (использовать не менее 3 лингвистических переменных) для задачи планирования объема производства продукции (с учетом возможной прибыли, необходимых ресурсов, платежеспособности населения, рынка сбыта и т.д.), проверить ее на полноту и произвести нечеткий вывод для конкретных значений (выбрать случайным образом).
 - 16) Построить нечеткую базу знаний (использовать не менее 3 лингвистических переменных) для задачи регулирования кондиционера (учитывать его мощность, объем помещения, температуру окружающей среды, необходимую температуру в помещении и т.д.), проверить ее на полноту и произвести нечеткий вывод для конкретных значений (выбрать случайным образом).
 - 17) Построить нечеткую базу знаний (использовать не менее 3 лингвистических переменных) для задачи распределения нагрузки между компьютерами при использовании их в кластерах (учитывать характеристики компьютеров, их количество, количество

параллельного кода, характеристики сети и т.д.), проверить ее на полноту и произвести нечеткий вывод для конкретных значений (выбрать случайным образом).

- 18) Построить нечеткую базу знаний (использовать не менее 3 лингвистических переменных) для задачи выбора складского помещения (учитывать площадь склада, количество и размеры продукции, удаленность от места производства и точек реализации, свойства продукции и характеристики помещений и т.д.), проверить ее на полноту и произвести нечеткий вывод для конкретных значений (выбрать случайным образом).
- 19) Построить нечеткую базу знаний (использовать не менее 3 лингвистических переменных) для задачи выбора комплектующих для компьютера (учитывать цену, потребности пользователя, совместимость, сроки использования и т.д.), проверить ее на полноту и произвести нечеткий вывод для конкретных значений (выбрать случайным образом).
- 20) Построить нечеткую базу знаний (использовать не менее 3 лингвистических переменных) для задачи определения количества линий в службе поддержки (учитывать количество обслуживаемых клиентов, среднюю частоту обращения в службу одного клиента, среднее время обслуживания одной заявки, квалификацию персонала и т.д.), проверить ее на полноту и произвести нечеткий вывод для конкретных значений (выбрать случайным образом).

Литература

- 1) Барсегян А.А., Куприянов М.С., Степаненко В.В., Холод И.И. Методы и модели анализа данных: OLAP и Data Mining. – СПб.: БХВ-Петербург, 2004. – 336 с.
- 2) Гаврилова Т.А., Хорошевский В.Ф. Базы знаний интеллектуальных систем. СПб.: Питер, 2001. с. 384.
- 3) Джарратано Д., Райлт Г. Экспертные системы: принципы разработки и программирование, 4-е издание. / Пер. с англ. – М.: ООО «И.Д. Вильямс», 2007. – 1152 с.
- 4) Интеллектуальные информационные системы: учебное пособие / А.А. Смагин, С.В. Липатова А.С., Мельниченко. – Ульяновск: УлГУ, 2009. - 123 с.
- 5) Круглов В.В., Дли М.И., Голунов Р.Ю. Нечеткая логика и искусственные нейронные сети. М.: ФИЗМАТЛИТ, 2001. - с. 224.
- 6) Минский М.Л. Фреймы для представления знаний. М.: Энергия, 1979. с. 152.
- 7) Прикладные нечеткие системы: пер. с япон./ К.Асаи, Д. Ватада, С. Иваи и др.; под редакцией Т. Тэрано, К. Асаи, М. Сугэно. – М.:Мир, 1993. – 368 с.
- 8) Рассел С., Норвиг П. Искусственный интеллект: современный подход, 2-е изд. – М.: Вильямс, 2006. - 1408 с.
- 9) Уоссермен Ф. Нейрокомпьютерная техника: Теория и практика. - Пер. с англ.,1992. - 118 с.
- 10) Фогель Л., Оуэнс А., Уолш М. Искусственный интеллект и эволюционное моделирование. М.: Мир, 1969. - 230 с.
- 11) Частиков А.П., Гаврилова Т.А., Белов Д.Л. Разработка экспертных систем. Среда CLIPS. – СПб.: БХВ-Петербург, 2003. – 608 с.