

МЕЖГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«БЕЛОРУССКО-РОССИЙСКИЙ УНИВЕРСИТЕТ»

Кафедра «Физические методы контроля»

ОСНОВЫ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА

*Методические рекомендации к лабораторным работам
для студентов направления подготовки
12.03.01 «Приборостроение»
очной формы обучения*



Могилев 2023

УДК 004.8
ББК 32.813
О-75

Рекомендовано к изданию
учебно-методическим отделом
Белорусско-Российского университета

Одобрено кафедрой «Физические методы контроля» «1» сентября 2023 г.,
протокол № 1

Составитель канд. техн. наук, доц. А. В. Кушнер

Рецензент канд. техн. наук, доц. М. Н. Миронова

В методических рекомендациях кратко изложены теоретические сведения, необходимые для выполнения лабораторных работ, и требования к оформлению. Методические рекомендации составлены в соответствии с рабочей программой по дисциплине «Основы искусственного интеллекта».

Учебное издание

ОСНОВЫ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА

Ответственный за выпуск	А. В. Хомченко
Корректор	А. А. Подошевка
Компьютерная верстка	Е. В. Ковалевская

Подписано в печать . Формат 60×84/16. Бумага офсетная. Гарнитура Таймс.
Печать трафаретная. Усл. печ. л. . Уч.-изд. л. . Тираж 26 экз. Заказ №

Издатель и полиграфическое исполнение:
Межгосударственное образовательное учреждение высшего образования
«Белорусско-Российский университет».
Свидетельство о государственной регистрации издателя,
изготовителя, распространителя печатных изданий
№ 1/156 от 07.03.2019.
Пр-т Мира, 43, 212022, г. Могилев.

© Белорусско-Российский
университет, 2023

Содержание

1 Лабораторная работа № 1. Формирование базы знаний с помощью экспертов.....	4
2 Лабораторная работа № 2. Формирование базы знаний с помощью таблицы решений.....	7
3 Лабораторная работа № 3. Формирование базы знаний с помощью продукционных правил.....	11
4 Лабораторная работа № 4. Формирование базы знаний с помощью семантических сетей.....	14
5 Лабораторная работа № 5. Формирование базы знаний с помощью фреймов.....	17
6 Лабораторная работа № 6. Экспертные системы с неопределенными знаниями и байесовские сети доверия.....	19
7 Лабораторная работа № 7. Экспертные системы извлечения знаний на основе генетических алгоритмов.....	26
8 Лабораторная работа № 8. Разработка искусственного нейрона.....	37
Список литературы.....	44

1 Лабораторная работа № 1. Формирование базы знаний с помощью экспертов

Цель работы: разработка программного обеспечения (ПО) для формирования базы знаний, основанной на опросе экспертов.

1.1 Теоретические сведения

Эффективность начальных этапов разработки экспертных систем (этапов идентификации и концептуализации) во многом определяется успешным формированием авторитетной группы экспертов и получением от них качественных знаний, составляющих основу любой экспертной системы.

1.1.1 Экспертное оценивание как процесс измерения.

Суть процесса выявления знаний заключается в организации проведения экспертами интуитивно-логического анализа проблемной области с количественной оценкой формулируемых ими суждений.

На этом этапе эксперты:

- формируют объекты и понятия предметной области (цели, решения, альтернативные ситуации и т. д.);
- производят измерение характеристик (вероятности свершения событий, коэффициенты значимости целей, предпочтение решений и т. д.).

Экспертное оценивание представляет собой процесс измерения, который можно определить как процедуру сравнения объектов по выбранным показателям (признакам). В этом определении фигурируют три понятия: объект, показатель (признак) и процедура сравнения.

Объекты – предметы, явления, решения.

Показатели сравнения – пространственно-временные, физические, психические и другие свойства и характеристики объектов.

Процедура сравнения включает в себя:

- определение причинно-следственной связи между объектами;
- установление степени влияния одних объектов на другие.

Введение конкретных показателей сравнения позволяет экспертам устанавливать отношения между объектами, например, «больше», «лучше», «более чем», «хуже», «одинаковы», «предпочтительнее» и т. д.

Эмпирическая система – это совокупность интересующих нас объектов вместе с системой связывающих их отношений.

При экспертном оценивании предметной области важным является возможность использования для эмпирической системы с отношениями построения числовой системы с отношениями, описывающими влияние объектов и отношения между ними с помощью чисел.

1.1.2 Методы измерения степени влияния объектов.

Типовыми методами измерения степени влияния объектов являются:

- ранжирование;
- парное сравнение;
- непосредственная оценка.

Ранжирование – процедура упорядочения объектов по степени их влияния на результат. На основе своих знаний и опыта эксперт располагает объекты в порядке предпочтения, руководствуясь одним или несколькими показателями сравнения. В зависимости от вида отношений между объектами возможны различные варианты упорядочения объектов.

Парное сравнение представляет собой процедуру установления предпочтения объектов при сравнении всех возможных пар. В отличие от ранжирования, при котором осуществляется упорядочение всех объектов сразу, парное сравнение является для экспертов более простой задачей.

Непосредственная оценка представляет собой процедуру приписывания объектам числовых значений в шкале интервалов. Эти значение соответствует степени влияния того или иного объекта на наблюдаемый результат. В процессе выявления знаний эксперт должен сопоставить каждому объекту значение, например, из диапазона $[0, 1]$. Естественно потребовать, чтобы сходным объектам приписывалось бы одно и то же значение. Измерение предпочтения в шкале интервалов можно выполнить с высокой степенью доверия только при хорошей информированности экспертов о свойствах объектов и предметной области. В ряде случаев, для ослабления этих условий за счет уменьшения точности измерения, рассматривают оценку, которая использует 5-, 10-, 100-балльные шкалы. Однако непосредственная оценка не всегда должна использовать числовые шкалы. Например, цвет объекта невозможно представить в виде какого-либо числового значения, а переход к значениям частот спектра во многих случаях затруднителен для эксперта.

1.1.3 Характеристики экспертов и группы экспертов.

Основными характеристиками экспертов являются:

- компетентность;
- креативность;
- отношение к экспертизе;
- конформизм;
- коллективизм и самокритичность.

Основными характеристиками группы экспертов является:

- достоверность экспертизы;
- затраты на экспертизу.

Обе эти характеристики определяют количество экспертов в группе и их качество (рисунок 1.1).

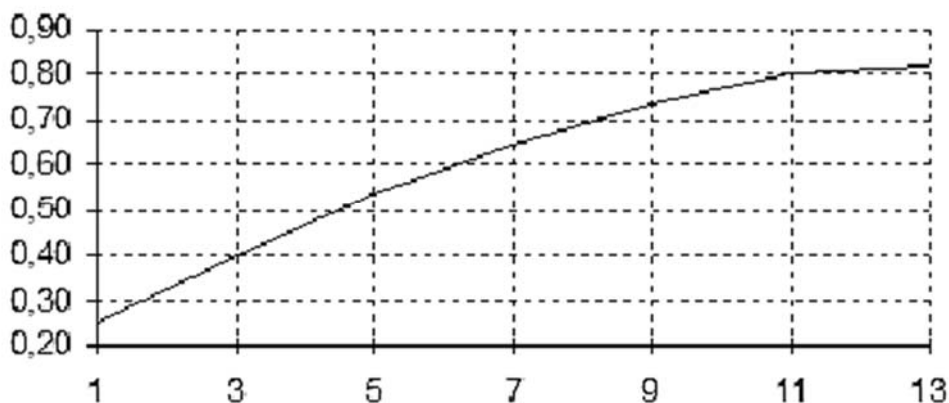


Рисунок 1.1 – Зависимость достоверности экспертизы от числа экспертов

Наряду с компетентностью, каждый из экспертов может характеризоваться достоверностью суждений. Эта характеристика определяется на основе информации о прошлом опыте его участия в решении проблем.

Для решения проблем с высоким уровнем информационного потенциала знаний, увеличение количества экспертов в группе приводит, как это следует из теории обработки наблюдений, к монотонному возрастанию достоверности экспертизы.

Опрос – это основной этап совместной работы группы экспертов.

При коллективной экспертизе используются следующие виды опроса:

- дискуссия;
- анкетирование и интервьюирование;
- метод коллективной генерации идей;
- мозговой штурм.

Результат опроса – информация, выражающая предпочтение экспертов и содержательное обоснование этих предпочтений.

Наличие как числовых данных, так и содержательных высказываний экспертов, приводит к необходимости применения качественных и количественных методов обработки результатов группового экспертного оценивания.

1.2 Задания для самостоятельного выполнения

1 В соответствии с выданным заданием разработать ПО для экспертного оценивания.

2 Одним из типовых методов провести оценивание степени влияния объектов.

3 Провести оценку достоверности экспертизы.

Контрольные вопросы

1 Что такое экспертное оценивание? Для чего оно необходимо?

2 Что включает в себя процедура сравнения?

3 Дайте определение эмпирической системы.

- 4 Какие существуют методы для измерения степени влияния объектов?
- 5 В чем заключается процедура ранжирования объектов?
- 6 Что такое непосредственная оценка объектов?
- 7 Назовите основные характеристики экспертов.
- 8 Какие виды опросов используются при коллективной экспертизе?

2 Лабораторная работа № 2. Формирование базы знаний с помощью таблицы решений

Цель работы: разработка программного обеспечения для формирования базы знаний, с использованием таблицы решений.

2.1 Теоретические сведения

Для представления знаний была предложена табличная форма, получившая название таблицы решений. Идея этого подхода очень проста и заключается в использовании табличной формы описания проблемы и способа ее решения (таблица 2.1).

Таблица 2.1 – Структура таблицы решений

Условие	Объект			
	O_1	O_2	...	O_K
U_1	1	0	...	1
...
U_l	1	1	...	0
Действие				
D_l	0	1	...	1
...
D_j	1	0	...	1

В этой таблице предполагается наличие двух основных частей: строк условий U_l и строк действий D_j . Каждое условие и действие соотносится с набором определенных объектов O_K .

Если некоторое условие или действие имеет место для данного объекта, то в соответствующую ячейку таблицы заносится 1, в противном случае используется символ 0.

В общем случае в ячейках таблицы решений могут использоваться любые символы, отражающие суть условия или действия: значения и диапазоны значений, коды и номера, символьные данные и т. п. В качестве действия может выступать также набор инструкций, подлежащих выполнению в случае выполнения условий.

Структура таблицы тоже может быть модифицирована под конкретную задачу. Например, может добавиться столбец с кодом операции, которую

необходимо выполнять для проверки заданного условия. Возможно использование упрощенного варианта таблицы, состоящей из пар «условие – действие» без указания каких-либо объектов.

Не представляет труда и расширение таблицы на новые классы объектов – необходимо лишь добавить в конец таблицы недостающие столбцы. А добавление новых строк, представляется несколько более сложным, т. к. потребуются внести изменения для всех существующих записей (столбцов).

Алгоритм поиска решения по таблице решений тривиален и прост в реализации. Интерпретирующая программа получает на вход конкретизированный набор условий для заданной ситуации и осуществляет последовательный перебор, подыскивая столбцы, где этот набор выполняется. Если такие столбцы находятся, то выполняется действие, указанное для этой ситуации.

Основное достоинство таблиц решений – высокая степень формализации, наглядности процесса принятия решений. Они строятся регулярным образом и могут наращиваться практически до бесконечности, т. е. являются универсальным средством решения задач, для которых возможно описание ситуаций с помощью ограниченного набора условий, например, проектирование деталей, представляющих собой поверхности вращения.

Недостаток – если различные ситуации характеризуются разными условиями, то таблицы решений становятся сильно разреженными и делают данный подход малоэффективным.

Рассмотрим пример использования таблицы решений, реализованный в системе STRIPS (Stanford Research Institute Problem Solver). Программа предназначалась для решения проблемы формирования плана поведения робота, перемещающего предметы через множество комнат.

Текущее состояние окружающей среды – помещений и предметов в них – представляется набором выражений предикат-аргумент, которые в совокупности образуют модель мира. Общепринятая конструкция предикат-аргумент, определяется следующим образом:

<предикат> ::= <предикатный символ> (<аргумент₁>, ..., <аргумент_n>).

Различают одноместные или n-местные предикаты. В случае одноместного предиката считается, что аргумент обладает свойством, выраженным предикатным символом. N-местный предикат описывает отношение между объектами, которые заданы аргументами.

Примером ситуации, описываемой с помощью двухместного предиката, может быть нахождение робота в определенной комнате:

at(robot, roomA).

Данный предикат означает, что объект robot находится в комнате roomA.

Порядок следования аргументов в предикате определяется разработчиком, но должен быть неизменным везде, где этот предикат используется.

Можно было бы написать $at(roomA, robot)$ и придать этой записи аналогичный смысл, но использовать одновременно в пределах одной базы знаний записи $at(robot, roomA)$ и $at(roomA, robot)$ нельзя.

С помощью множества предикатов можно описать текущую модель мира, т. е. набор конкретных объектов, их свойств и отношений.

Например, исходная ситуация может описываться множеством

$$W1 = \{at(robot, roomA), at(box1, roomB), at(box2, roomC)\}.$$

Конечная ситуация также задается множеством предикатов

$$WK = \{at(box1, roomA), at(box2, roomB)\}.$$

Множество допустимых операций, таких как перемещение робота или перенос предметов, кодируются в таблице операторов, которая близка по структуре к таблице решений, но предполагает обязательное наличие модели мира (таблица 2.2).

Таблица 2.2 – Пример таблицы операторов

Оператор	$move(X, Y)$	$push(X, Y, Z)$
Предварительное условие	$\{at(robot, X)\}$	$\{at(robot, Y), at(X, Y)\}$
Список удалений	$\{at(robot, X)\}$	$\{at(robot, Y), at(X, Y)\}$
Список добавлений	$\{at(robot, Y)\}$	$\{at(robot, Z), at(X, Z)\}$

Задачей системы STRIPS является формирование плана действий робота для достижения цели. Таким образом, результатом работы системы должна быть последовательность операторов, применение которых к исходной модели мира позволяет достичь целевой модели. Зная целевое состояние среды, можно было бы перебирать последовательно или случайно комбинации операторов, пока цель не будет достигнута. Но экспоненциальный рост количества вариантов при каждой новой проверке делает такой подход неприемлемым на практике.

Для предотвращения экспоненциального роста вариантов возможных решений в качестве основы для работы системы предложен метод анализа целей и средств, идея которого состоит в том, чтобы с каждой новой операцией отличие между текущим состоянием и целевым уменьшалось. Это предполагает наличие меры оценки «расстояния» в пространстве решений.

Например, если очередная подцель сформулирована в виде предиката $at(box1, roomA)$, а ящик $box1$ находится в комнате $roomB$, то перемещение робота из комнаты $roomA$ в комнату $roomC$ не приблизит текущее состояние к целевому (для модели $W1$). А перемещение из комнаты $roomA$ в комнату $roomB$, наоборот, уменьшит «расстояние» между текущим и целевым состоянием, т. к. позволит на следующем шаге переместить ящик $box1$ в комнату $roomA$.

Алгоритм поиска требуемых операторов системы STRIPS основан на сопоставлении очередной подцели и списков добавления в таблице операторов. Новые подцели выбираются из списка предварительного условия найденного

оператора. Так, например, цель $at(box1, roomA)$ соответствует элементу $at(X, Z)$ в списке добавлений оператора $push(X, Y, Z)$.

Сопоставление этих двух предикатов (X соответствует $box1$, а Z – $roomA$) позволяет выбрать из предварительного условия оператора новые подцели: $at(robot, Y)$, $at(box1, Y)$.

Далее необходимо найти в модели мира предикат, содержащий объект (в данном случае это комната), который можно сопоставить с символом Y .

Таким предикатом может быть, например, $at(box1, roomB)$. Поставив в соответствие с символом Y объект $roomB$ можно окончательно сформулировать очередные подцели: $at(robot, roomB)$ и $at(box1, roomB)$.

Теперь первый элемент в этом списке указывает желаемое (целевое) положение робота, а второй элемент уже присутствует в модели мира $W1$.

Следовательно, после применения оператора $push$ к модели мира $W1$, необходимо добавить $at(robot, roomB)$ и удалить $at(robot, roomA)$. В результате получится новая модель

$$W2 = \{at(robot, roomB), at(box1, roomB), at(box2, roomC)\}.$$

Описанная процедура повторяется до тех пор, пока очередная модель не будет соответствовать целевой.

Так как таблица операторов, модель мира и цели представлены с помощью одного и того же синтаксиса в виде конструкций предикат-аргумент, то, применяя описанную выше схему сопоставления, программа довольно просто находит, какие именно операции нужно выполнить для достижения поставленной цели. Всё, что нужно для этого сделать, – просмотреть списки добавлений в описании операторов и найти в них элемент, соответствующий заданной цели.

2.2 Задания для самостоятельного выполнения

1 В соответствии с выданным заданием разработать ПО для формирования таблицы решений.

2 С помощью множества предикатов описать исходную и конечную ситуацию.

3 Разработать процедуру перехода из начального в конечное состояние.

Контрольные вопросы

1 В чем заключается основная идея таблицы решений?

2 Назовите основное достоинство алгоритма поиска решений.

3 В чем заключается недостаток алгоритма поиска решений?

4 Для чего применяется система STRIPS?

5 Что является основной задачей системы STRIPS?

3 Лабораторная работа № 3. Формирование базы знаний с помощью продукционных правил

Цель работы: разработка программного обеспечения для формирования базы знаний с использованием продукционных правил.

3.1 Теоретические сведения

Системы представления знаний, использующие выражения вида «ЕСЛИ условие, ТО действие», получили название систем продукций или систем, основанных на правилах.

Правила обеспечивают формальный способ представления рекомендаций, указаний или стратегий. Они хорошо подходят в тех случаях, когда знания предметной области возникают из эмпирических ассоциаций, накопленных за годы работы по решению задач в данной области.

Представления знаний в виде продукций наиболее распространено в экспертных системах, т. к. запись знаний фактически ведется на подмножестве естественного языка.

Следствием этого является то, что правила легко читаются, их просто понять и модифицировать, эксперты без труда могут сформулировать новое правило или указать на ошибочность какого-либо существующего.

В качестве условия и действия в правилах может быть, например, предположение о наличии того или иного свойства, принимающее значение истина или ложь. При этом термин действие следует трактовать широко: это может быть как директива к выполнению какой-либо операции, рекомендация, или модификация базы знаний – предположение о наличии какого-либо производного свойства. Примером продукции может служить следующее выражение:

ЕСЛИ клиент работает на одном месте более двух лет,
ТО клиент имеет постоянную работу.

Как условие, так и действие правила, могут учитывать несколько выражений, объединенных логическими связками И, ИЛИ, НЕ, например:

ЕСЛИ клиент имеет постоянную работу **И** клиенту более 18 лет
И клиент **НЕ** имеет финансовых обязательств,
ТО клиент может претендовать на получение кредита.

Помимо продукционных правил база знаний должна включать и простые факты, поступающие в систему через интерфейс пользователя или выводимые в процессе поиска решения задачи. Факты являются простыми утверждениями типа «клиент работает на одном месте более двух лет». И когда в процессе

интерпретации правил машиной вывода какой-либо факт согласуется с частью правила ЕСЛИ, то выполняется действие, определяемое частью ТО этого правила.

Новые факты, добавляемые в базу знаний в результате действий, описанных в правилах, также могут быть использованы для сопоставления с частями ЕСЛИ других правил. Последовательное сопоставление частей правил ЕСЛИ фактами порождает цепочку вывода (рисунок 3.1).



Рисунок 3.1 – Пример цепочки вывода

Эта цепочка вывода показывает, как на основании правил и исходных фактов выводит заключение о возможности получения кредита. Цепочки вывода экспертной системы могут быть предъявлены пользователю и помогают понять, как было получено решение.

Существуют два основных способа выполнения правил в системе: прямая цепочка рассуждений или прямой вывод и обратная цепочка рассуждений или обратный вывод. В приведенном выше примере использовалась прямая цепочка рассуждений – от исходных фактов к цели (заключению).

Этот вид вывода применим в задачах, где на основании имеющихся фактов необходимо определить тип (класс) объекта или явления, выдать рекомендацию, определить диагноз и т. п.

Если же задача формулируется иначе – для заданной ситуации необходимо определить условия к ней приводящие, то используется обратный вывод.

Подобная задача может формулироваться, например, следующим образом: «Определить условия, при которых клиент может получить кредит».

Обратная цепочка применима также, если требуется объяснить, как было получено решение.

В процессе вывода решения все правила системы равнозначны и самодостаточны, т. е. все необходимое для активизации правила содержится в его

условии, и одни правила не могут непосредственно вызывать другие. Но иногда для получения решения требуется вмешательство в стандартный процесс вывода. Для этих целей некоторые продукционные системы позволяют вводить в базу знаний специальные правила для управления процессом вывода – метаправила.

Метаправила не принимают непосредственного участия в процессе формирования рассуждений, а определяют приоритет выполнения или исключают из рассмотрения обычные правила и выполняются в первую очередь. Ниже приведен пример метаправила, сокращающего цепочку вывода:

ЕСЛИ кредитный рейтинг клиента высокий

И клиент является клиентом банка,

ТО сначала применить правила для льготных условий предоставления кредита.

Можно также сформулировать пример метаправила, касающегося общей стратегии вывода и не связанного с какой-либо конкретной предметной областью:

ЕСЛИ существуют правила, в условиях которых не упоминается текущая цель,

И существуют правила, в условиях которых упоминается текущая цель,

ТО сначала следует активизировать первые из перечисленных правил.

Последний пример не связан с предметной областью и поэтому подобные метаправила можно применять в системах различного назначения.

Интерес к такого рода обобщенным формулировкам знаний достаточно высок. Идея использования метаправил является весьма продуктивной, но, тем не менее, метаправилами следует пользоваться осмотрительно, учитывая возможные исключительные ситуации.

Продукционные правила обеспечивают естественный способ описания процессов, управляемый сложной и изменяющейся средой. Через правила можно описать ход решения задачи, не имея заранее алгоритма этого решения. Более того, можно корректировать способ решения, добавляя новые правила, не изменяя существующих, что обеспечивает высокую модульность базы знаний.

Однако, несмотря на то, что с помощью продукционных правил можно представить решение любой задачи, при большом количестве правил становится сложно отслеживать непротиворечивость базы знаний.

3.2 Задания для самостоятельного выполнения

1 В соответствии с выданным заданием разработать ПО для формирования продукционных правил.

2 Разработать алгоритм формирования цепочки выводов.

Контрольные вопросы

1 Какие системы представления знаний получили название «системы продукций»?

2 Что представляет собой цепочка вывода?

- 3 Назовите основные способы выполнения правил в экспертной системе.
- 4 Что такое метаправила?
- 5 Для чего необходимы продукционные правила?

4 Лабораторная работа № 4. Формирование базы знаний с помощью семантических сетей

Цель работы: разработка программного обеспечения для формирования базы знаний с использованием семантических сетей.

4.1 Теоретические сведения

Семантика устанавливает отношения между символами и объектами, которые они обозначают, т. е. определяет смысл знаков.

Семантические сети впервые были предложены Р. Квиллианом в 1968 г. в качестве способа описания феноменов человеческой памяти.

Первоначально семантические сети использовались для анализа значения слов в предложениях. В последующем они успешно применялись для решения задач, связанных с представлением и извлечением знаний за счет возможности учитывать не только формы утверждений, но и их семантику.

Такие знания позволяют экспертным системам получить приемлемый вывод при работе с неоднозначными фактами.

Семантическая сеть – это ориентированный граф, вершины которого представляют понятия, а дуги – различные отношения между ними (рисунок 4.1).

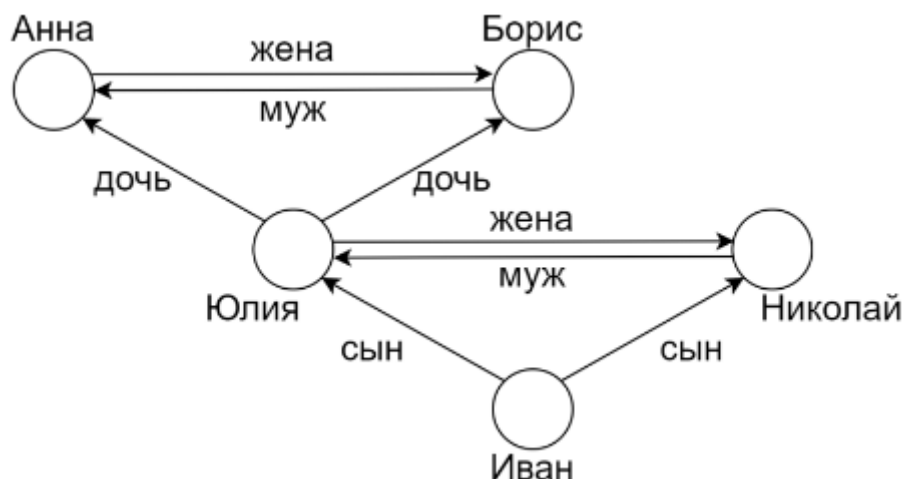


Рисунок 4.1 – Пример семантической сети

Понятия семантической сети – это абстрактные или конкретные объекты, а отношения – связи типа: is-a, has-a, a-kind-of, cause.

Связь is-a – означает, что отдельный объект является экземпляром определенного класса. Примером такого типа связей может быть отнесения клиентов банковских учреждений к определенному классу (рисунок 4.2).

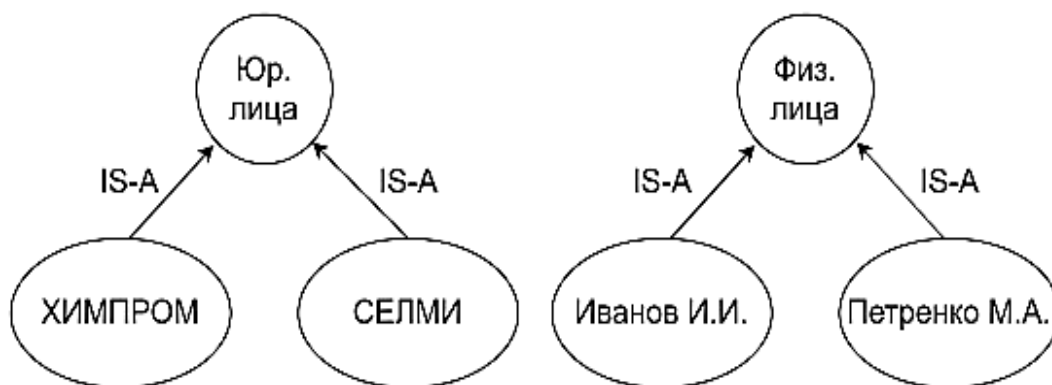


Рисунок 4.2 – Связь is-a

Связь is-a используется для обозначения отношений между отдельными объектами через принадлежность их к общему классу, благодаря тождественности атрибутов.

Связь a-kind-of определяет отношение между самими родовыми классами. Следует отметить, что общий класс, на который указывает стрелка, называется суперклассом (рисунок 4.3).

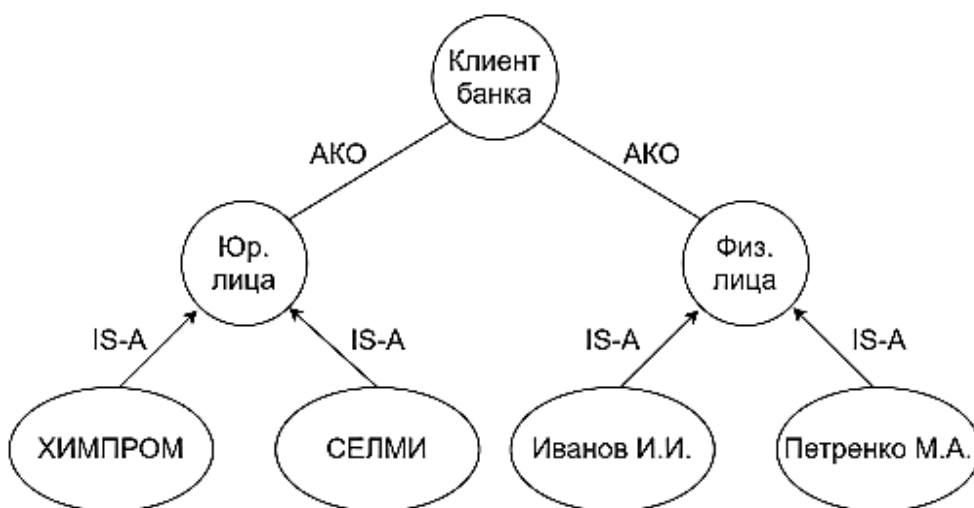


Рисунок 4.3 – Связь a-kind-of

В случае если суперкласс имеет связь a-kind-of, что указывает на другой узел, то он, вместе с тем, является классом суперкласса.

Таким образом, в семантических сетях есть три основных типа отношений:

- 1) класс – элемент класса;
- 2) свойство – значение;
- 3) пример элемента класса.

По типу отношений семантические сети бывают:

- однородные (с единственным типом отношений);
- неоднородные (с различными типами отношений).

По типам отношений:

- бинарные (в которых отношения связывают два объекта);
- n-арные (в которых есть специальные отношения, связывающие более двух понятий).

Часто используемыми отношениями являются:

- связи типа «часть-целое» («класс-подкласс», «элемент-множество» и т. п.);
- функциональные связи («производит», «влияет» и т. п.);
- количественные («больше», «меньше», «равно» и т. п.);
- пространственные («далеко от», «близко от», «за», «над» и т. п.);
- временные («раньше», «позже», «в течение» и т. п.);
- атрибутивные связи («иметь свойство», «иметь значение» и т. п.);
- логические связи (И, ИЛИ, НЕ) и др.

На рисунке 4.4 приведен пример семантической сети.

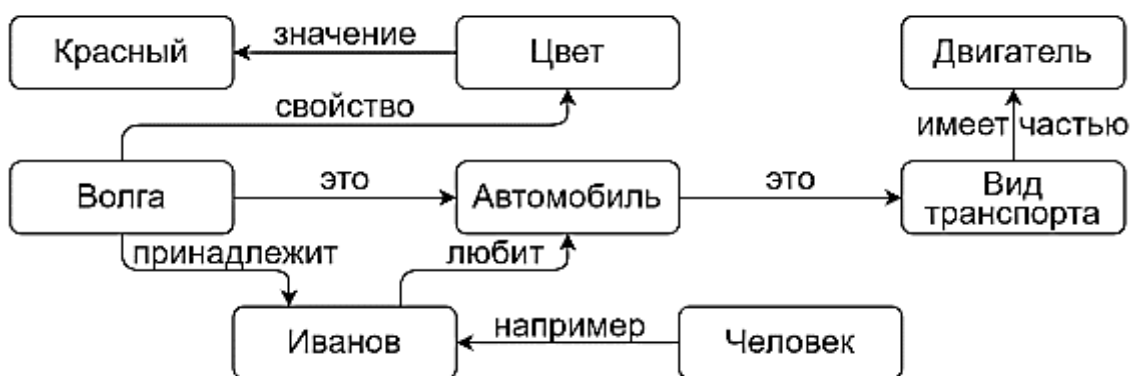


Рисунок 4.4 – Пример семантической сети

Поиск решения на основе знаний, представленных семантическими сетями, типа семантической сети сводится к задаче поиска фрагмента сети (подсети), соответствующей поставленному вопросу.

Достоинство семантических сетей – соответствие представлениям об организации долговременной памяти человека, а недостаток – сложность вывода.

4.2 Задания для самостоятельного выполнения

1 В соответствии с выданным заданием разработать ПО для формирования семантических сетей.

2 Разработать алгоритм формирования выводов на основе семантических сетей.

Контрольные вопросы

- 1 Дайте определения семантики.
- 2 Что такое семантическая сеть?
- 3 Что представляет собой тип связи is-a?
- 4 Что представляет собой тип связи a-kind-of?
- 5 Для чего используется связь is-a?
- 6 Назовите типы отношений в семантических сетях.

7 Какие бывают семантические сети по типу отношений?

8 Что представляют собой n-арные семантические сети?

9 Назовите основные типы отношений, которые используются в семантических сетях.

5 Лабораторная работа № 5. Формирование базы знаний с помощью фреймов

Цель работы: разработка программного обеспечения для формирования базы знаний с использованием фреймов.

5.1 Теоретические сведения

Фреймы предложены М. Минским в 70-е гг. XX в. как структура знаний для восприятия пространственных сцен. Эта модель, как и семантическая сеть, имеет глубокое психологическое обоснование. Под фреймом понимается абстрактный образ или ситуация.

В психологии и философии известно понятие абстрактного образа.

Например, слово «комната» представляет образ «жилого помещения с четырьмя стенами, полом, потолком, окнами и дверью, площадью до 30 кв. м.», из описания которого ничего нельзя убрать и «слоты» которого заполняются значениями атрибутов – количеством окон, цветом стен, высотой потолка, покрытием пола и др.

Такой образ, а точнее его формализованная модель, и называется фреймом. В таблице 5.1 показана структура фрейма.

Таблица 5.1 – Структура фрейма

Имя фрейма			
Имя слота	Тип слота	Значение слота	Присоединенная процедура
...

Дополнительные столбцы предназначены для описания типа слота и возможного присоединения к тому или иному слоту специальных процедур.

В качестве значения слота может выступать имя другого фрейма, так образуют сети фреймов.

Различают фреймы-образцы, или прототипы, хранящиеся в базе знаний, и фреймы-экземпляры, которые создаются для отображения реальных ситуаций на основе поступающих данных.

Модель фрейма позволяет отобразить знания через:

- фреймы-структуры, для обозначения объектов и понятий (например, заем, залог, вексель);
- фреймы-роли (например, менеджер, кассир, клиент);
- фреймы-сценарии (например, банкротство, собрание акционеров, празднование именин);
- фреймы-ситуации (например, тревога, авария, рабочий режим устройства) и др.

Важным свойством фреймов является заимствованное из теории семантических сетей наследование свойств. И во фреймах, и в семантических сетях наследование происходит по АКО-связям (a-kind-of). Слот АКО указывает на фрейм более высокого уровня иерархии, откуда неявно наследуются, т. е. переносятся, значения аналогичных слотов.

Основным достоинством фреймов, как модели представления знаний, является ее гибкость и наглядность. Так, на вопрос: «Любят ли ученики сладкое?» Следует ответ: «Да», т. к. этим свойством обладают все дети, что указано во фрейме «ребенок». Наследование свойств может быть частичным. Так, возраст для учеников не наследуется из фрейма «ребенок», поскольку указан явно в своем собственном фрейме (рисунок 5.1).

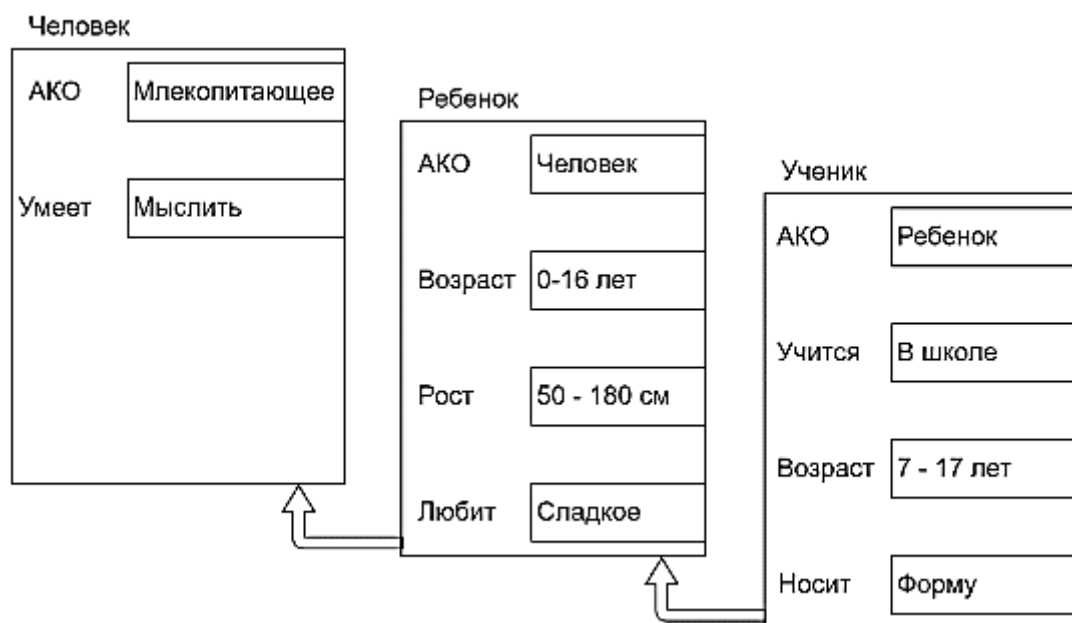


Рисунок 5.1 – Пример фрейма

5.2 Задания для самостоятельного выполнения

1 В соответствии с выданным заданием разработать ПО для формирования базы знаний на основе фреймов.

2 Разработать алгоритм формирования выводов на основе фреймов.

Контрольные вопросы

- 1 Что такое фрейм?
- 2 Опишите структуру фрейма.
- 3 Для чего создаются фреймы-образцы и фреймы-экземпляры?
- 4 Какие преимущества дает создание фреймов?
- 5 Каким свойством обладает теория фреймов?
- 6 Приведите пример фрейма-сценария.
- 7 Приведите пример фрейма-ситуации.

6 Лабораторная работа № 6. Экспертные системы с неопределенными знаниями и байесовские сети доверия

Цель работы: разработка программного обеспечения для формирования базы знаний в условиях неопределенности.

6.1 Теоретические сведения

6.1.1 Неопределенность в экспертных системах и проблемы, порождаемые ими.

В практических задачах часто приходится оценивать гипотезы, относительно которых имеется неполная или недостаточная информация. Но, несмотря на сложность точных оценок в условиях неопределенности, человеком принимаются разумные решения. Чтобы экспертные системы были полезными, они тоже должны уметь это делать.

Классическим примером таких задач является медицинская диагностика. Всегда существуют некоторые сомнения в четкости проявления симптомов того или иного заболевания. Сомнения в наличии у пациента конкретного заболевания сохраняются даже в том случае, когда все его симптомы отчетливо выражены.

Как проявляется и учитывается неопределенность экспертной системой?

Пусть используется правило:

Если (A), То (B),

и предположим, никакие другие правила и посылки не имеют отношения к рассматриваемой ситуации.

При этом неопределенность может быть двух типов:

1) неопределенность в истинности самой предпосылки (например, если степень уверенности в том, что A истинно составляет 90 %, то какие значения примет B ?);

2) неопределенность самого правила (например, можно сказать, что в большинстве случаев, но не всегда, если есть A , то есть также и B).

Еще более сложная ситуация возникает в случае, если правило имеет вид

Если (A И B), То (C),

где можно с некоторой степенью быть уверенными как в истинности каждой из посылок (A , B), а тем более их совместного проявления, так и в истинности самого вывода.

Типовыми проблемами при проектировании и создании экспертных систем являются следующие.

1 Как количественно выразить степень определенности при установлении истинности (или ложности) некоторой части данных?

2 Как выразить степень поддержки заключения конкретной посылкой?

3 Как использовать совместно две (или более) посылки, независимо влияющие на заключение?

4 Как быть в ситуации, когда нужно обсудить цепочку вывода для подтверждения заключения в условиях неопределенности?

Вероятность события определяется как отношение случаев, в которых данное событие происходит, к общему числу наблюдений.

Объективистский подход рассматривает вероятность отношения исходов ко всем наблюдениям в течение длительного времени. Этот подход основан на законе больших чисел, гарантирующим то, что при наличии достаточно большого количества наблюдений частота исходов интересующего события будет стремиться к объективной вероятности.

Персонифицированный, субъективистский подход рассматривает вероятностную меру как степень доверия того, как отдельная личность судит об истинности некоторого высказывания (событию). При этом постулируют, что данная личность имеет в некотором смысле отношение к этому событию. Но это не отрицает возможности того, что две личности могут иметь различные степени доверия для одного и того же суждения.

Необходимый или логический подход расширяет вероятностную меру на множество утверждений, имеющих логическую связь такую, что истинность одного из них может выводиться из другого. Вероятность здесь характеризует степень доказуемости логически выверенного заключения. Такой взгляд можно рассматривать как расширение обычной логики.

Для вероятностных расчетов используются два направления:

- 1) по Паскалю – байесовские правила для проверки и обработки мер доверия;
- 2) по Бэкону – правила логики для доказательства или опровержения гипотез.

Общепринятые вероятности (по Паскалю) не могут быть получены из индуктивных вероятностей (по Бэкону) и, наоборот. Объективистский и субъективный взгляды используют расчеты по Паскалю. Те же, кто поддерживают логические выводы, используют расчеты по Бэкону.

Существуют экспертные системы, построенные в рамках каждого из этих направлений. Однако большинство современных экспертных систем, использующих теорию вероятностей, являются «байесовскими».

6.1.2 Основные понятия теории вероятностей.

Пусть A – некоторое событие.

Совокупность всех элементарных событий называется выборочным пространством или пространство событий (Ω).

Вероятность события A , обозначается $p(A)$.

Вероятность события A , обозначается $p(A)$ и каждая вероятностная функция p должна удовлетворять трем аксиомам.

1 Вероятность любого события A является неотрицательной:

$$p(A) \geq 0 \text{ для } \forall A \in \Omega.$$

2 Вероятность всех событий выборочного пространства равна 1:

$$p(\Omega) = 1.$$

3 Если k событий являются взаимно независимыми (т. е. не могут произойти одновременно), то вероятность, по крайней мере, одного из этих событий равна сумме отдельных вероятностей:

$$p(A_1 \cup A_2 \cup \dots \cup A_k) = \sum_{i=1}^k p(A_i).$$

Аксиомы 1 и 2 можно объединить:

$$1 \geq p(A) \geq 0 \text{ для } \forall A \in \Omega.$$

Это утверждение показывает, что вероятность любого события – между 0 и 1. По определению, когда $p(A) = 0$, то событие A никогда не произойдет.

В том случае и когда $p(A) = 1$, то событие A должно произойти обязательно.

Дополнение к A , обозначаемое $(\neg A)$, содержит совокупность всех событий за исключением A .

Так как A и $\neg A$ являются взаимонезависимыми, то из аксиомы следует:

$$p(A) + p(\neg A) = p(A \cup \neg A) = p(\Omega) = 1.$$

Переписывая это равенство в виде

$$p(\neg A) = 1 - p(A),$$

определяем способ для получения

$$p(\neg A) \text{ из } p(A).$$

Предположим теперь, что B некоторое другое событие. Тогда вероятность того, что произойдет A при условии, что произошло B записывается в виде $p(A|B)$ и называется условной вероятностью события A при заданном событии B .

Вероятность того, что оба события A и B произойдут, называется совместной вероятностью событий A и B :

$$p(A \cap B).$$

Условная вероятность $p(A|B)$ равна отношению совместной вероятности к вероятности события B при условии, что она не равна 0:

$$p(A|B) = \frac{p(A \cap B)}{p(B)}.$$

Аналогично условная вероятность события B при условии A :

$$p(B|A) = \frac{p(B \cap A)}{p(A)}.$$

Таким образом,

$$p(B \cap A) = p(B | A) \times p(A).$$

Так как совместная вероятность коммутативна, то

$$p(A \cap B) = p(B \cap A) = p(B | A) \times p(A).$$

Подставляя это равенство в ранее полученное выражение для условной вероятности $p(A|B)$ получим правило Байеса:

$$p(A | B) = \frac{p(B | A) \times p(A)}{p(B)}.$$

В ряде случаев знание того, что произошло событие B , не влияет на вероятность события A (или наоборот A на B). Другими словами, вероятность события A не зависит от того, что произошло или нет событие B , так что

$$p(A | B) = p(A) \text{ и } p(B | A) = p(B).$$

В этом случае говорят, что события A и B являются независимыми.

6.1.3 Теорема Байеса.

Приведенные выше соотношения предполагают определенную связь между теорией вероятностей и теорией множеств. Если A и B являются непересекающимися множествами, то объединение множеств соответствует сумме вероятностей, а пересечение – произведению вероятностей

$$p(A \cup B) = p(A) + p(B) \text{ и } p(A \cap B) = p(A) \times p(B).$$

Без предположения независимости эта связь является неточной и формулы должны содержать дополнительные члены включения.

Продолжая теоретико-множественную интерпретацию B , можно записать:

$$B = (B \cap A) \cup (B \cap \neg A).$$

Так как это объединение явно непересекающееся, то

$$\begin{aligned} p(B) &= p((B \cap A) \cup (B \cap \neg A)) = p(B \cap A) + p(B \cap \neg A) = \\ &= p(B | A) p(A) + p(B | \neg A) p(\neg A). \end{aligned}$$

Возвращаясь к обозначению событий, а не множеств, последнее равенство может быть подставлено в правило Байеса:

$$p(A | B) = \frac{p(B | A) \times p(A)}{p(B | A) \times p(A) + p(B | \neg A) \times p(\neg A)}.$$

Это выражение является основой для использования положений теории вероятности при учете неопределенности. Оно определяет способ получения условной вероятности события B при условии A . Это соотношение позволяет экспертным системам «делать вывод вперед и назад».

6.1.4 Байесовские сети доверия.

Байесовские сети доверия (Bayesian Belief Network) используются в задачах, которые характеризуются неопределенностью вследствие: неполного понимания задачи; неполных знаний; наличия случайных событий.

Для байесовских сетей доверия (БСД) используется ещё одно название – причинно-следственные сети, если случайные события в них соединены причинно-следственными связями. Такие связи наиболее часто делаются в направлении от «наблюдателя» к «наблюдению», или от «эффекта» к «следствию» (рисунок 6.1).

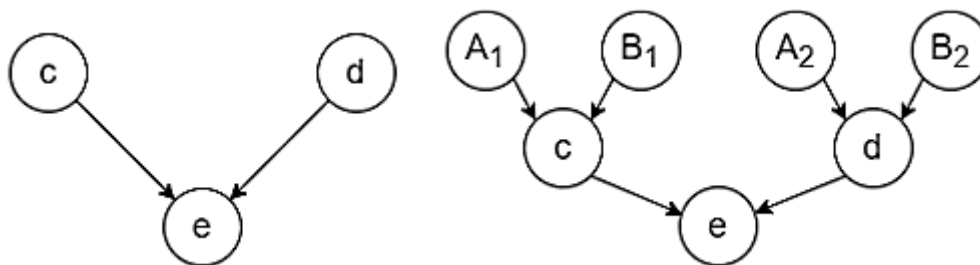


Рисунок 6.1 – Примеры байесовских сетей доверия

Вероятность пребывания вершины «e» в различных состояниях (e_k) зависит от состояний (c_i, d_j) вершин «c» и «d».

Байесовская сеть доверия представляется в виде направленного ациклического графа, обладающего следующими свойствами:

- каждая вершина представляет собой событие, описываемое случайной величиной, которая может иметь несколько состояний;
- все вершины, связанные с «родительскими», определяются таблицей условных вероятностей (ТУВ) или функцией условных вероятностей (ФУВ);
- для вершин без «родителей» вероятности их состояний являются безусловными (маргинальными).

Другими словами, в БСД вершины представляют собой случайные переменные, а дуги – вероятностные зависимости, которые определяются через ТУВ. Таблица условных вероятностей каждой вершины содержит вероятности состояний этой вершины при условии состояний её «родителей».

Рассмотрим пример построения простейшей БСД.

Однажды дерево лишилось листвы. Выясним, почему это случилось. Знаем, что листва часто опадает, если:

- дерево засыхает в результате недостатка влаги;
- или дерево болеет.

Данная ситуация может быть смоделирована байесовской сетью доверия, содержащей три вершины: «Болеет», «Засохло» и «Облетело» (рисунок 6.2).



Рисунок 6.2 – Пример простейшей БСД

Рассмотрим ситуацию, при которой каждая вершина может принимать всего лишь два возможных состояний и, как следствие, находится в одном из них (таблица 6.1).

Таблица 6.1 – Табличная иллюстрация БСД

Вершина (событие) БСД	Состояние 1	Состояние 2
«Болеет»	«да»	«нет»
«Засохло»	«да»	«нет»
«Облетело»	«да»	«нет»

Вершина «Болеет» говорит о том, что дерево заболело, будучи в состоянии «болеет», в противном случае она находится в состоянии «нет». Аналогично для других двух вершин. Рассматриваемая БСД, моделирует тот факт, что имеется причинно-следственная зависимость от события «Болеет» к событию «Облетело» и от события «Засохло» к событию «Облетело». Это отображено стрелками на БСД.

Когда имеется причинно-следственная зависимость от вершины A к другой вершине B , то ожидаем, что когда A находится в некотором определённом состоянии, это оказывает влияние на состояние B . Следует быть внимательным, когда моделируется зависимость в БСД. Иногда совсем не очевидно, какое направление должна иметь стрелка.

В рассматриваемом примере, говорим, что имеется зависимость от «Болеет» к «Облетело», т. к. когда дерево болеет, это может вызывать опадание его листья. Опадание листья является следствием болезни, а не болезнь – следствием опадания листья.

На рисунке дано графическое представление байесовской сети доверия.

Однако это только качественное представление байесовской сети доверия.

Перед тем, как назвать это полностью байесовской сетью доверия, необходимо определить количественное представление, т. е. множество таблиц условных вероятностей (таблицы 6.2 и 6.3).

Таблица 6.2 – Таблицы условных вероятностей для родительских вершин

Априорная вероятность $p(\text{«Болеет»})$		Априорная вероятность $p(\text{«Засохло»})$	
Болеет = «да»	Болеет = «нет»	Засохло = «да»	Засохло = «нет»
0,1	0,9	0,1	0,9

Таблица 6.3 – Таблица условных вероятностей для вершины-потомка

Таблица условных вероятностей $p(\text{«Облетело»} \text{«Болеет», «Засохло»})$				
Вероятность	Засохло = «да»		Засохло = «нет»	
	Болеет = «да»	Болеет = «нет»	Болеет = «да»	Болеет = «нет»
Облетело = «да»	0,95	0,85	0,90	0,02
Облетело = «нет»	0,05	0,15	0,10	0,98

Заметим, что все три таблицы показывают вероятность пребывания некоторой вершины в определённом состоянии, обусловленным состоянием её родительских вершин. Но так как вершины «Болеет» и «Засохло» не имеют родительских вершин, то их вероятности являются маргинальными, т. е. не зависят (не обусловлены) ни от чего.

В примере пусть известно, что дерево сбросило листву. Это свидетельство вводится выбором состояния «да» в вершине «Облетело». После этого можно узнать вероятности того, что дерево засохло. Для приведенных выше исходных данных:

$$p(\text{«Болеет»} = \text{«да»} | \text{«Облетело»} = \text{«да»}) = 0,47;$$

$$p(\text{«Засохло»} = \text{«да»} | \text{«Облетело»} = \text{«да»}) = 0,49.$$

6.2 Задания для самостоятельного выполнения

- 1 Формирование экспертных систем на основе законов распределения случайных величин.
- 2 Разработка экспертных систем по оценке вероятности поломки оборудования.
- 3 Разработка экспертных систем по диагностике оборудования.

Контрольные вопросы

- 1 Какие типы неопределенностей встречаются в экспертных системах?
- 2 Какие проблемы необходимо решать при проектировании и создании экспертных систем?
- 3 В чем заключается объективистский подход теории субъективных вероятностей?
- 4 Какие направления вероятностных расчетов существуют?
- 5 Что характерно для байесовских сетей доверия?
- 6 Назовите свойства направленного ациклического графа в байесовской сети доверия.
- 7 Что представляют собой вершины в байесовских сетях доверия?
- 8 Что представляют собой дуги в байесовских сетях доверия?

7 Лабораторная работа № 7. Экспертные системы извлечения знаний на основе генетических алгоритмов

Цель работы: разработка программного обеспечения для извлечения знаний на основе генетических алгоритмов.

7.1 Теоретические сведения

Интеллектуальные системы извлечения новых знаний (обучения и самообучения) ориентированы на автоматическое накопление и формирование знаний с использованием процедур анализа и обобщения данных. К ним относятся системы символьного, нейросетевого и эволюционного обучения.

Системы символьного обучения ориентированы на интеллектуальный анализ данных (Data Mining), поиск скрытых правил и закономерностей (Knowledge Discovery), автоматические рассуждения, доказательство теорем.

При поиске закономерностей задача и относящаяся к ней информация описывается в виде логических аксиом. В дальнейшем система рассматривает различные варианты задачи как теоремы, которые следует доказать.

Нейросетевые системы используют методы обучения, направленные на модификацию собственной структуры (структуры сети) и весовых коэффициентов связей между элементами.

Эволюционные системы построены на принципах генетических и эволюционных процессов (алгоритмов), когда из набора кандидатов (популяции), получаемого посредством операций скрещивания и мутаций, по принятому критерию отбираются лучшие, наиболее приспособленные для решения «особи». Они реализуют один из эффективных подходов к решению многомодальных (имеющих несколько локальных экстремумов) оптимизационных задач большой размерности. Эволюционные вычисления не гарантируют обнаружения глобального экстремума целевой функции (оптимального решения) за определенное время, однако позволяют найти «хорошие» решения очень трудных задач.

Основными направлениями исследований в области эволюционных вычислений являются следующие:

- эволюционное программирование;
- эволюционные стратегии;
- генетические алгоритмы.

В основе эволюционного программирования лежит идея представления альтернатив решения задачи в виде универсальных конечных автоматов, которые способны реагировать на стимулы, поступающие из окружающей среды. Абстрактный автомат в теории алгоритмов (Л. Дж. Фогель, 1960 г.) – модель дискретного устройства, имеющего один вход, один выход и в каждый момент времени находящегося в одном состоянии из множества возможных. На вход поступают символы одного алфавита, на выходе формируются символы

(в общем случае) другого алфавита. Автомат может быть представлен в виде графа, вершинами которого являются состояния, а дуги – переходами между ними. Как правило, автоматы используются для описания поведения программного обеспечения, аппаратных и технических средств.

Эволюционные стратегии (И. Рехенберг, 1964 г.) – это эвристические методы оптимизации, основанные на принципах адаптации и эволюции. Альтернативы представляются единым массивом, воздействие стратегией на которые осуществляется с учетом семантики альтернатив и направлено на улучшение значений, входящих в них параметров. При поиске решения вначале происходит скрещивание особей для получения потомков и их мутация, а затем выполняется детерминированный отбор без повторений лучших особей из родителей и порожденных потомков.

Отличительной особенностью генетических алгоритмов (Дж. Холланд, 1975 г.) является представление любой альтернативы решения в виде кодовой (как правило, битовой) строки фиксированной длины, манипуляции с которой проводятся в отсутствие всякой связи с ее смысловой интерпретацией.

Рассмотрим более подробно основные вопросы построения и использования генетических алгоритмов. На рисунке 7.1 проиллюстрирована схема естественного отбора в природе.



Рисунок 7.1 – Схема естественного отбора

Суть естественного отбора, как основного механизма эволюции, состоит в том, что более приспособленные особи имеют больше возможностей для выживания и размножения и, следовательно, приносят больше потомства, чем плохо приспособленные особи. При этом, благодаря передаче генетической информации (генетическому наследованию), потомки наследуют от родителей основные их качества. Таким образом, потомки сильных индивидуумов также будут относительно хорошо приспособленными, а их доля в общей массе особей будет возрастать. После смены нескольких десятков или сотен поколений средняя приспособленность особей данного вида заметно возрастает.

Механизмы генетического наследования в природе основаны на том, что в каждой клетке любого животного содержится вся генетическая информация этой

особи. Эта информация записана в виде набора очень длинных молекул ДНК. Каждая молекула ДНК – это цепочка, состоящая из молекул нуклеотидов четырех типов, обозначаемых А, Т, С и G. Собственно, информацию несет порядок следования нуклеотидов в ДНК.

Генетический код индивидуума – это просто очень длинная строка символов, где используются всего четыре буквы. В животной клетке каждая молекула.

ДНК окружена оболочкой – такое образование называется хромосомой.

Каждое врожденное качество особи (цвет глаз, наследственные болезни, тип волос и т. д.) кодируется определенной частью хромосомы, которая называется геном этого свойства. Так, ген цвета глаз содержит информацию, которая кодирует определенный цвет глаз. Различные значения гена называются его аллелями.

При размножении животных происходит слияние двух родительских половых клеток и их ДНК взаимодействуют, образуя ДНК потомка. Основным способом взаимодействия – кроссовер (crossover, скрещивание). При кроссовере ДНК предков делятся на две части, а затем обмениваются своими половинками.

При наследовании возможны мутации из-за радиоактивности или других влияний, в результате которых могут измениться некоторые гены в половых клетках одного из родителей. Измененные гены передаются потомку и придают ему новые свойства. Если эти новые свойства полезны, они, скорее всего, сохранятся в данном виде – при этом произойдет скачкообразное повышение приспособленности вида.

Рассмотрим основные понятия генетических алгоритмов.

Ген (свойство) – атомарный элемент хромосомы, может быть битом, числом или неким другим объектом.

Аллель – значение конкретного гена.

Локус – положение конкретного гена в хромосоме.

Хромосома (цепочка) – упорядоченная последовательность генов.

Генотип (код) – упорядоченная последовательность хромосом.

Особь (индивидуум) – конкретный экземпляр генотипа.

Фенотип – набор значений, соответствующих генотипу (представляет собой интерпретацию генотипа с точки зрения решаемой задачи).

Распространен случай, когда генотип состоит всего из одной хромосомы и представляется в виде битовой строки. Таким образом, ген – это бит; генотип (хромосома) – битовая строка заданной размерности и с определенным положением битов; особь – конкретный набор битов (0 и 1).

На каждом шаге работы генетический алгоритм использует несколько точек поиска одновременно. Совокупность этих точек является набором особей, который называется популяцией. Количество особей в популяции называют размером популяции.

На каждом шаге генетический алгоритм обновляет популяцию путем создания новых особей и уничтожения ненужных. Чтобы отличать популяции на каждом из шагов и сами эти шаги, их называют поколениями и обычно идентифицируют по номеру. Например, популяция, полученная из исходной

популяции после первого шага работы алгоритма, будет первым поколением, после следующего шага вторым и т. д.

При работе алгоритма генерация новых особей происходит на основе моделирования процесса размножения. При этом, естественно, порождающие особи называются родителями, а порожденные – потомками.

Выбор родителей для скрещивания выполняется случайным образом с помощью оператора отбора. При этом, один родитель может участвовать в нескольких скрещиваниях и необязательно, чтобы все родители участвовали в размножении.

На рисунке 7.2 показана типовая схема генетического алгоритма.

Родительская пара, как правило, порождает пару потомков. Непосредственная генерация новых кодовых строк из двух выбранных происходит за счет работы оператора скрещивания, который также называют кроссинговером (англ. crossover).

Моделирование процесса мутации новых особей осуществляется за счет работы оператора мутации, применяемого к случайно выбранным потомкам за счет изменения случайного выбранного гена.

Поскольку размер популяции фиксирован, то порождение потомков должно сопровождаться уничтожением особей. Выбор лучших («жизнеспособных») особей из числа родителей и потомков выполняет оператор редукции, который уничтожает худшие («малоприспособленные») особи.

Основным правилом отбора является закон эволюции – «выживает сильнейший», который обеспечивает улучшение искомого решения. В некоторых источниках процесс приведения расширенной популяции к исходному размеру рассматривается не с точки зрения уничтожения худших особей (редукции), а с точки зрения выбора лучших (отбора).

Операторы отбора, скрещивания, мутации и редукции называют генетическими операторами. Отбор и мутация выполняются с использованием элементов случайности, скрещивание и редукция по строго детерминированным правилам.

Остановка генетического алгоритма происходит в случаях, когда:

- сформировано заданное число поколений;
- исчерпано время, отведенное на эволюцию;
- популяция достигла заданного качества – значение критерия одной, нескольких или всех особей превысило заданный порог;
- достигнут некоторый уровень сходимости – особи в популяции стали настолько подобными, что дальнейшее их улучшение происходит чрезвычайно медленно.



Рисунок 7.2 – Схема генетического алгоритма

Основные отличия генетических алгоритмов от традиционных методов поиска решений.

1 Генетические алгоритмы работают с кодовыми строками, от которых зависят значения аргументов целевой функции и, соответственно, значение самой целевой функции. Интерпретация этих кодов выполняется только в операторе редукции. В остальном работа алгоритма не зависит от смысловой интерпретации кодов.

2 Для поиска лучшего решения генетический алгоритм на отдельном шаге использует сразу несколько точек поискового пространства (несколько вариантов решения задачи) одновременно, а не переходит от точки к точке, как это делается в традиционных методах. Это позволяет преодолеть один из их недостатков опасность попадания в локальный экстремум целевой функции, если она не является унимодальной (т. е. имеет несколько экстремумов). Использование нескольких точек одновременно значительно снижает такую возможность.

3 Генетический алгоритм использует как вероятностные правила для порождения новых точек для анализа, так и детерминированные правила для перехода от одних точек к другим. Одновременное использование элементов случайности и детерминированности дает значительно больший эффект, чем раздельное.

Рассмотрим пример реализации генетического алгоритма.

Пусть требуется найти глобальный минимум функции

$$y(x) = 5 - 24x + 17x^2 - \frac{11}{3}x^3 + \frac{1}{4}x^4$$

на отрезке $[0, 7]$.

На этом отрезке функция принимает минимальное значение в точке $x = 1$. Очевидно, что в точке $x = 6$ функция попадает в локальный минимум.

Если для нахождения глобального минимума использовать градиентные методы, то в зависимости от начального приближения можно попасть в данный локальный минимум.

Для простоты положим, что x принимает лишь целые значения, т. е. $x \in \{0, 1, 2, 3, 4, 5, 6, 7\}$.

Выберем случайным образом несколько чисел на отрезке $[0, 7]$: $\{2, 3, 5, 4\}$. Будем рассматривать эти числа в качестве пробных решений задачи.

Основной идеей генетических алгоритмов является организация «борьбы за существование» и «естественного отбора» среди этих пробных решений.

Запишем пробные решения в двоичной форме: $\{010, 011, 101, 100\}$.

Генетические алгоритмы используют биологические аналогии и принцип естественного отбора – в конкурентной борьбе выживает наиболее приспособленный. В нашем случае приспособленность особи определяется целевой функцией: чем меньше значение целевой функции, тем более приспособленной является особь, т. е. пробное решение, использовавшееся в качестве аргумента целевой функции (рисунок 7.3).

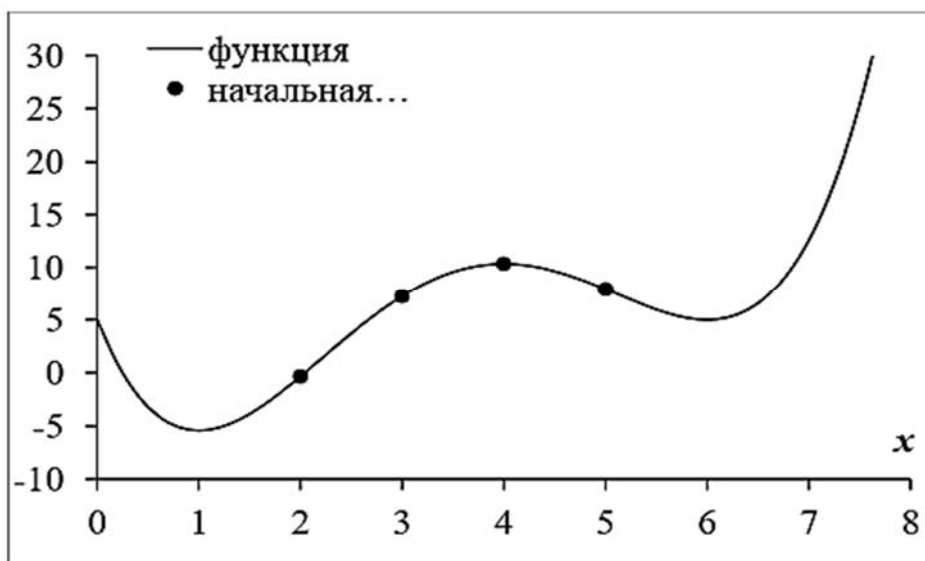


Рисунок 7.3 – График целевой функции с выбранными значениями пробных решений

Попробуем на основе исходной популяции создать новую, чтобы пробные решения в новой популяции были бы ближе к искомому глобальному минимуму целевой функции. Для этого сформируем из исходной популяции пары для скрещивания (таблица 7.1).

Таблица 7.1 – Исходная популяция

Номер особи	Особь		
	Целое число	Двоичное число	Приспособленность
1	2	010	-0,33
2	3	011	37,25
3	5	101	7,92
4	4	100	10,33

На этапе скрещивания поставим в соответствие каждой особи исходной популяции случайное целое число из диапазона от 1 до 4. Будем рассматривать эти числа как номера особей популяции. При таком выборе какие-то из особей популяции не будут участвовать в процессе скрещивания, т. к. образуют пару сами с собой. Какие-то особи популяции примут участие в процессе скрещивания неоднократно с различными особями популяции.

Процесс скрещивания (рекомбинация) заключается в обмене участками хромосом между родителями. Например, пусть скрещиваются две хромосомы 111111 и 000000. Определяем случайным образом точку разрыва хромосомы, пусть это будет 3: 111|111 000|000. Теперь хромосомы обмениваются частями, стоящими после точки разрыва, и образуют двух новых потомков: 111000 и 000111 (таблица 7.2).

Таблица 7.2 – Одноточечный кроссинговер

Номер опыта	Особь популяции	Выбранный номер	Вторая особь-родитель	Точка кроссинговера	Особь-потомки
1	010	1	010 \leftrightarrow 0 10	1	000
2	011	4	100 \leftrightarrow 1 00		110
3	101	3	101 \leftrightarrow 1 01	2	100
4	100	1	010 \leftrightarrow 0 10		011

Следующим этапом генетического алгоритма являются мутации, т. е. случайные изменения полученных в результате скрещивания хромосом.

Пусть вероятность мутации равна 0,3. Для каждого потомка возьмем случайное число на отрезке $[0, 1]$, и если это число меньше 0,3, то инвертируем случайно выбранный ген (заменяем 0 на 1 или наоборот) (таблица 7.3, рисунок 7.4).

Таблица 7.3 – Мутация потомков

Номер опыта	Особь-потомки	Случайное число	Выбранный ген для мутации	Потомок после мутации	Приспособленность потомка до мутации	Приспособленность потомка после мутации
1	000	0,1	3	001	5	-5,42
2	110	0,6	–	110	5	5
3	100	0,5	–	100	10,33	10,33
4	011	0,2	1	111	7,25	12,58

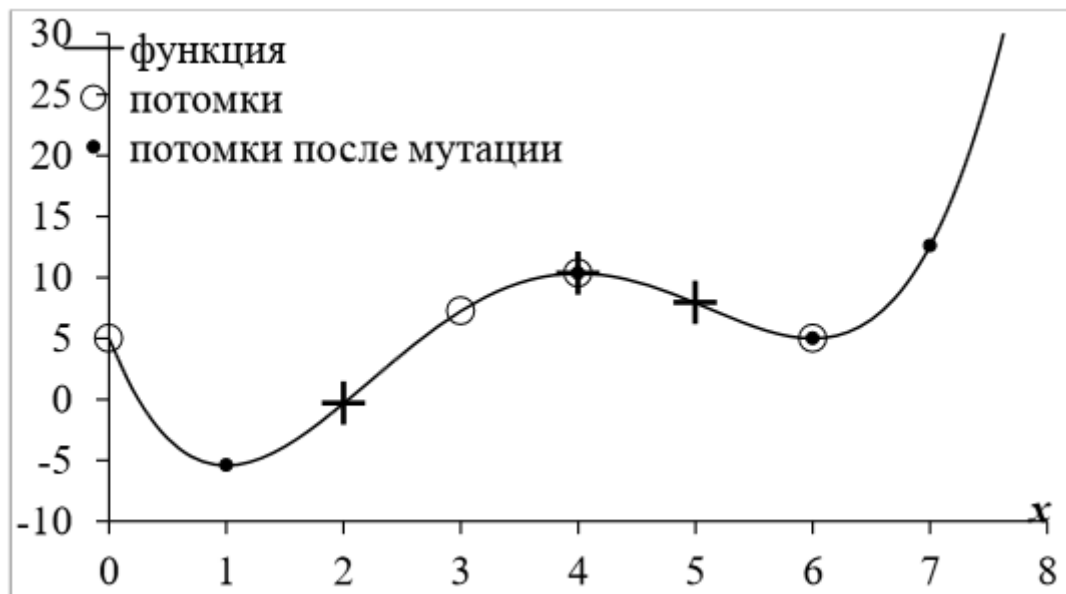


Рисунок 7.4 – График целевой функции с особями после мутации

Как видно, мутации способны улучшить (первый потомок) или ухудшить (четвертый потомок) приспособленность особи-потомка.

В результате скрещивания хромосомы обмениваются «хвостами», т. е. младшими разрядами в двоичном представлении числа.

В результате мутаций изменению может подвергнуться любой разряд, в том числе старший. Таким образом, если скрещивание приводит к относительно небольшим изменениям пробных решений, то мутации могут привести к существенным изменениям значений пробных решений.

Теперь из четырех особей-родителей и четырех полученных особей потомков необходимо сформировать новую популяцию. В новую популяцию отберем четыре наиболее приспособленных особи из числа «старых» особей и особей-потомков. Получившуюся популяцию можно будет вновь подвергнуть кроссинговеру, мутации и отбору особей в новое поколение. Таким образом, через несколько поколений мы получим популяцию из похожих и наиболее приспособленных особей. Значение приспособленности наиболее «хорошей» особи (или средняя приспособленность по популяции) и будет являться решением нашей задачи.

Следуя этому, в данном случае, взяв наиболее приспособленную особь 001 во втором поколении, можно сказать, что минимумом целевой функции является значение $-5,42$, соответствующее аргументу $x = 1$ (таблица 7.4, рисунок 7.5). Тем самым попадания в локальный минимум удалось избежать!

Таблица 7.4 – Формирование новой популяции из особей-родителей и особей-потомков

Номер опыта	Особь	Приспособленность	Новая популяция	Приспособленность особей в новой популяции
1	010	-0,33	001	-5,42
2	011	7,25	010	-0,33
3	101	7,92	110	5
4	100	10,33	011	7,25
5	001	-5,42		
6	110	5		
7	100	10,33		
8	111	12,58		

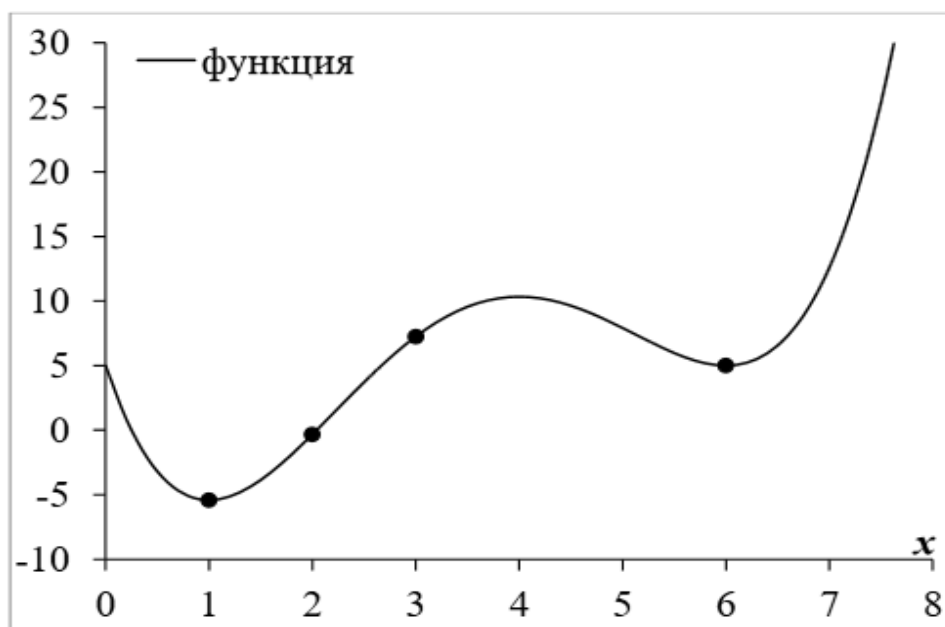


Рисунок 7.5 – График целевой функции после завершения работы генетического алгоритма

Рассмотрим пример работы генетического алгоритма при решении задачи коммивояжера.

Задача. Коммивояжеру требуется посетить N городов. Для каждой пары городов по маршруту следования установлена стоимость (расстояние, время) проезда. Требуется найти путь минимальной стоимости, который начинается из некоторого города, обеспечивает посещение всех остальных городов ровно по одному разу и возврат в точку отправления (рисунок 7.6).

Задача коммивояжера относится к категории NP-полных задач, т. е. задач, решаемых методом полного перебора всех вариантов.

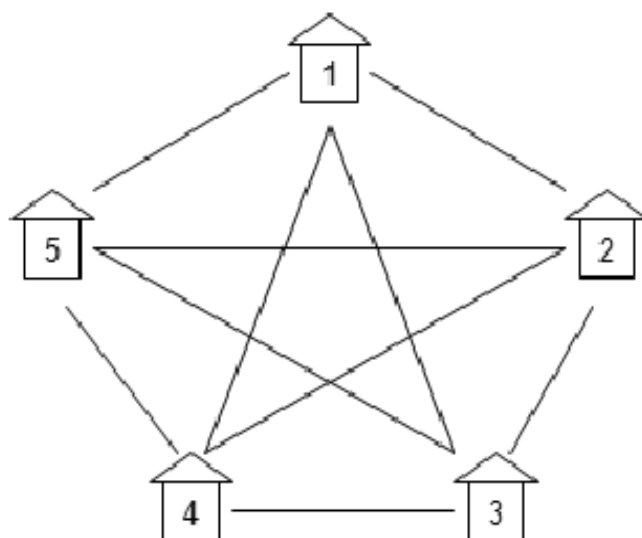


Рисунок 7.6 – Пример графического представления задачи коммивояжера

Ген – здесь число, характеризующее номер посещаемого города. Генотип – строка из чисел длиной N , описывающая порядок посещения городов.

Особь – конкретная строка из чисел (допустимый вариант решения задачи).

Предположим, $N = 9$ и особи «231586479» и «147523869» – примеры допустимых вариантов решения задачи. Классическое скрещивание приведет к генерации недопустимых вариантов, например, Родители Потомки:

23158 6479	23158 3869
14752 3879	14158 6479

т. к. в потомках посещение некоторых городов будет дублироваться или проигнорировано.

Предложены различные варианты решения данной проблемы, в частности, Л. Девис предлагает следующую модификацию оператора скрещивания.

Шаг 1. Случайным образом выбираются два сечения генотипа:

$$P1 = 231 | 568 | 479$$

$$P2 = 147 | 523 | 869.$$

Шаг 2. Для потомков копируются участки кода, расположенные между сечениями:

$$P1 = xxx | 586 | xxx,$$

$$P2 = xxx | 523 | xxx.$$

Шаг 3. Из родителей генерируются вспомогательные строки, у которых участки года после второго сечения перемещаются в начало:

$$B1 = 479 231 586,$$

$$B2 = 869 147 523.$$

Шаг 4. Свободные гены потомков последовательно заполняются генами из перекрестных вспомогательных строк с пропуском уже имеющихся в потомке генов:

$$P1 = 914 | 586 | 723,$$

$$P2 = 479 | 523 | 186.$$

Оператор мутации также может быть реализован различными способами, например:

– перестановка пары, случайным образом выбранных генов местами:

$$479523186 \rightarrow 473529186;$$

– инверсия случайным образом выбранной последовательности генов:

$$479 | 523 | 186 \rightarrow 479 | 325 | 186.$$

7.2 Задания для самостоятельного выполнения

Реализация генетического алгоритма для поиска минимума целевой функции в заданной предметной области.

Контрольные вопросы

- 1 Какие типы систем извлечения новых знаний существуют?
- 2 На каких принципах построены нейросетевые системы?
- 3 Что представляют собой системы символьного обучения?
- 4 На каких принципах базируются эволюционные системы?
- 5 Назовите основные направления эволюционных вычислений.
- 6 Сформулируйте определение абстрактного автомата в теории алгоритмов.
- 7 Назовите основные направления эволюционных вычислений и их авторов.
- 8 В чем заключается основной механизм эволюции?
- 9 Что представляет собой генетический код индивидуума?
- 10 Назовите основные понятия, используемые в генетических алгоритмах.
- 11 Сформулируйте определения популяции и ее размера.
- 12 Какие операторы используются в генетических алгоритмах?
- 13 В каких случаях происходит остановка генетического алгоритма?
- 14 Назовите основные отличия генетических алгоритмов от традиционных методов поиска решений.

8 Лабораторная работа № 8. Разработка искусственного нейрона

Цель работы: изучение назначения, особенности построения, обучения и функционирования искусственного нейрона.

8.1 Теоретические сведения

8.1.1 Краткая история исследований.

1943 г. У. Маккалох (W. McCulloch) и У. Питтс (W. Pitts) предложили модель формального нейрона.

1949 г. Д. Хебб (D. Hebb) высказал идеи о характере соединений нейронов мозга и их взаимодействии (клеточные ансамбли, синаптическая пластичность). Впервые предложил правила обучения нейронной сети.

1957 г. Ф. Розенблатт (F. Rosenblatt) разработал принципы организации и функционирования перцептронов, предложил вариант технической реализации первого в мире нейрокомпьютера Mark.

1959 г. Д. Хьюбел (D. Hubel) и Т. Визель (T. Wiesel) показали распределенный и параллельный характер хранения и обработки информации в биологических нейронных сетях.

1960–1968 гг. Активные исследования в области искусственных нейронных сетей, АДАЛИНА и МАДАЛИНА В. Уидроу (W. Widrow) (1960–1962 гг.), ассоциативные матрицы К. Штайнбуха (K. Steinbuch) (1961 г.).

1969 г. Вышла книга М. Минского (M. Minsky) и С. Пейперта (S. Papert) «Перцептроны», в которой доказывается ограниченность возможностей перцептронов.

1974 г. П. Вербос разработал алгоритм обратного распространения ошибки (backpropagation) для обучения многослойных перцептронов. Одновременно А. И. Галушкин опубликовал книгу, в которой обобщил подходы к обучению многослойных нейронных сетей. Заново алгоритм обратного распространения ошибки был «переоткрыт» в 1982 г. Д. Паркером.

1970–1976 гг. Активные разработки в области перцептронов в СССР (основные заказчики – военные ведомства).

1975 г. К. Фукушима (K. Fukushima) предложил когнитрон – самоорганизующуюся сеть для инвариантного распознавания образов.

1980 г. К. Фукушима представил концепцию неокогнитрона.

1982–1985 гг. Дж. Хопфилд (J. Hopfield) предложил семейство оптимизирующих нейронных сетей, моделирующих автоассоциативную память.

1982 г. Т. Кохонен (T. Kohonen) разработал обучающиеся без учителя самоорганизующиеся карты (Self-Organizing Maps).

1985 г. Первые коммерческие нейрокомпьютеры Mark III (TRW, США).

1987 г. С. Гроссберг (S. Grossberg) создал адаптивную резонансную теорию (ART) и модели нейронных сетей на ее основе.

1989 г. Я. Лекун (Y. LeCun) предложил сверточные нейронные сети для инвариантного распознавания изображений.

1990-е гг. Начало широкомасштабного финансирования разработок в области нейронных сетей в США, Японии и Западной Европе. Агентством DARPA (США) начато финансирование программы по созданию сверхбыстродействующих образцов нейрокомпьютеров.

1990-е гг. Активизация СССР в области искусственных нейронных сетей: Институт кибернетики им. Глушкова (Киев), Институт многопроцессорных вычислительных систем (Таганрог), Институт нейрокибернетики (Ростов-на-Дону), центры нейрокомпьютеров в Москве, Новосибирске, Санкт-Петербурге.

1997 г. З. Хохрайтер (S. Hochreiter) и Ю. Шмидхубер (J. Schmidhuber) предложили новый подход к созданию рекуррентных нейронных сетей – LSTM-сетей (Long Short-Term Memory).

2000-е гг. Решена проблема попадания искусственной нейронной сети в локальный минимум с применением стохастических методов обучения (больцмановское обучение, обучение Коши).

2007 г. Дж. Хинтон создал алгоритмы глубокого обучения многослойных нейронных сетей с использованием больцмановского обучения (RBM – Restricted Boltzmann Machine) для «нижних» слоев сети.

Искусственная нейронная сеть (ИНС) – структурированная совокупность искусственных нейронов, определенным образом соединенных друг с другом и с внешней средой с помощью связей, задаваемых весовыми коэффициентами.

Особенности искусственных ИНС:

- обучение на основе прецедентов (примеров);
- обобщение предыдущего опыта;
- извлечение значимой информации и закономерностей из избыточных и зашумленных данных;
- адаптивность к изменению условий функционирования.

Использование ИНС является целесообразным, если:

- отсутствует алгоритм решения задачи или неизвестен принцип ее решения, но имеются экспериментальные данные ее решения;
- задача характеризуется большими объемами информации;
- данные неполны, зашумлены, избыточны или противоречивы.

Постановка задачи для искусственных нейронных сетей: необходимо построить такое отображение $X \rightarrow Y$, чтобы на каждый входной сигнал X формировался правильный выходной сигнал Y .

Области применения ИНС:

- классификация и распознавание образов (X – входной образ; Y – номер класса, к которому принадлежит входной образ);
- кластеризация/категоризация (X – входной вектор; Y – кластер, к которому относится входной вектор);
- аппроксимация функции (X – вектор входных переменных; Y – аппроксимированная выходная переменная);
- предсказание/прогноз (X – временные ряды на некотором интервале времени; Y – подмножество переменных входного сигнала)
- идентификация (X и Y представляют собой входные и выходные сигналы идентифицируемой системы, процесса)
- оптимизация;

- ассоциативная память.
- управление.

8.1.2 Искусственный нейрон.

На рисунке 8.1 представлена типовая структура искусственного нейрона.

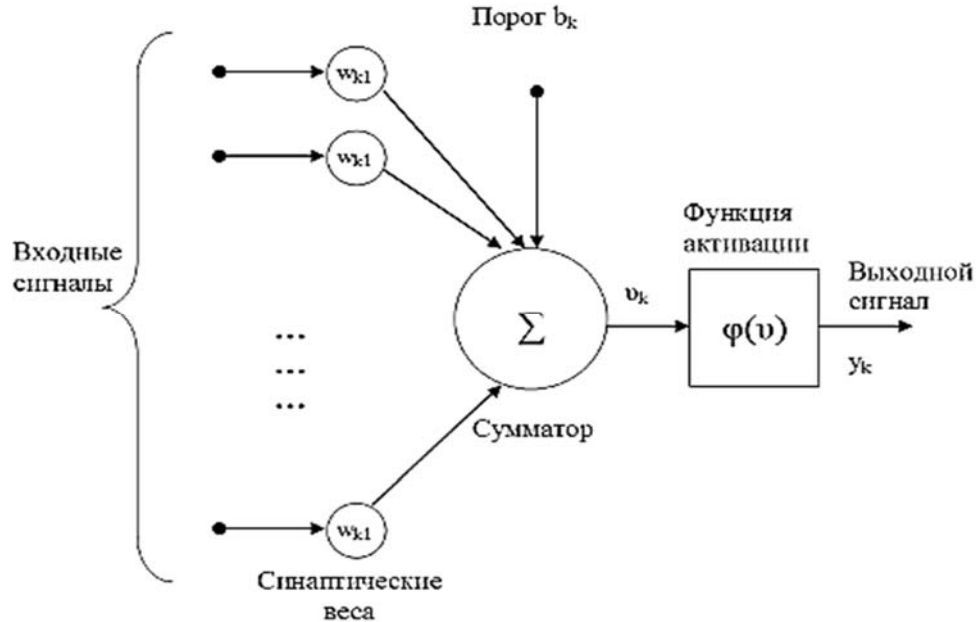


Рисунок 8.1 – Схема искусственного нейрона

Основные элементы нейрона:

– набор синапсов/связей, каждый из которых характеризуется; сигнал x_j на входе синапса j , связанного с нейроном k , умножается на вес w_{kj} (первый индекс относится к рассматриваемому нейрону, а второй – ко входному окончанию синапса, с которым связан данный вес); синаптический вес может быть как положительным, так и отрицательным;

– сумматор складывает входные сигналы, взвешенные относительно соответствующих синапсов нейрона;

– функция активации ограничивает амплитуду выходного сигнала нейрона; нормированный диапазон амплитуд выхода нейрона задается в диапазоне в интервале $[0, 1]$ или $[-1, 1]$.

Функционирование нейрона k описывается следующими выражениями:

$$u_k = \sum_{j=1}^m w_{kj} x_j;$$

$$y_k = \varphi(u_k + b_k),$$

Где x_1, x_2, \dots, x_m – входные сигналы;

$w_{k1}, w_{k2}, \dots, w_{km}$ – синаптические веса k -го нейрона;

u_k – линейная комбинация входных воздействий;

b_k – пороговое смещение (обеспечивает эффект аффинного преобразования выхода сумматора);

φ – функция активации;

y_k – выходной сигнал нейрона.

b_k – использование порога.

В нейроне постсинаптический потенциал вычисляется следующим образом:

$$v_k = u_k + b_k.$$

В частности, в зависимости от того, какое значение принимает порог b_k , положительное или отрицательное, индуцированное локальное поле или потенциал активации u_k k -го нейрона изменяется так, как показано на рисунке 8.2.

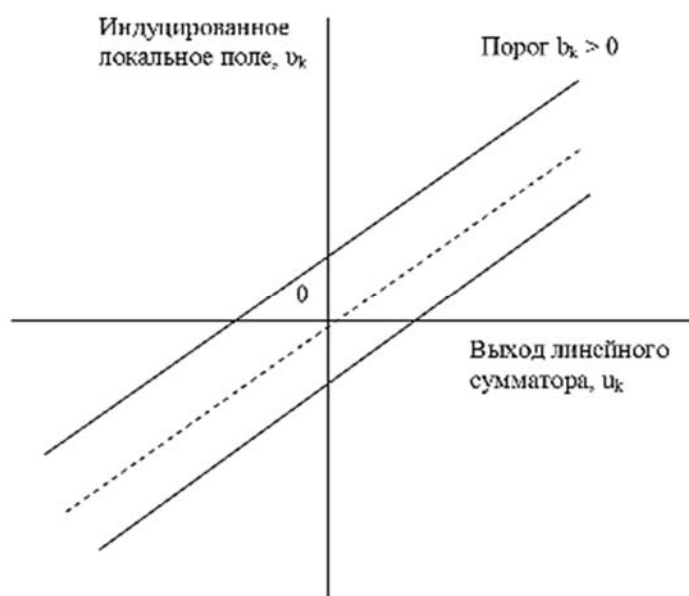


Рисунок 8.2 – Влияние порога на выход нейрона

На рисунке 8.3 проиллюстрированы примеры активационных функций искусственного нейрона:

– функция единичного скачка

$$f(s) = \begin{cases} 0, s < \Theta; \\ 1, s \geq \Theta; \end{cases}$$

– кусочно-линейная функция

$$f(s) = \begin{cases} 0, s < -a; \\ \frac{1}{2a}(s + a), -a < s < a; \\ 1, s > a; \end{cases}$$

– логистическая (сигмоидальная)

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

– гиперболический тангенс (сигмоидальная)

$$f(s) = \frac{e^{as} - e^{-as}}{e^{as} + e^{-as}}.$$

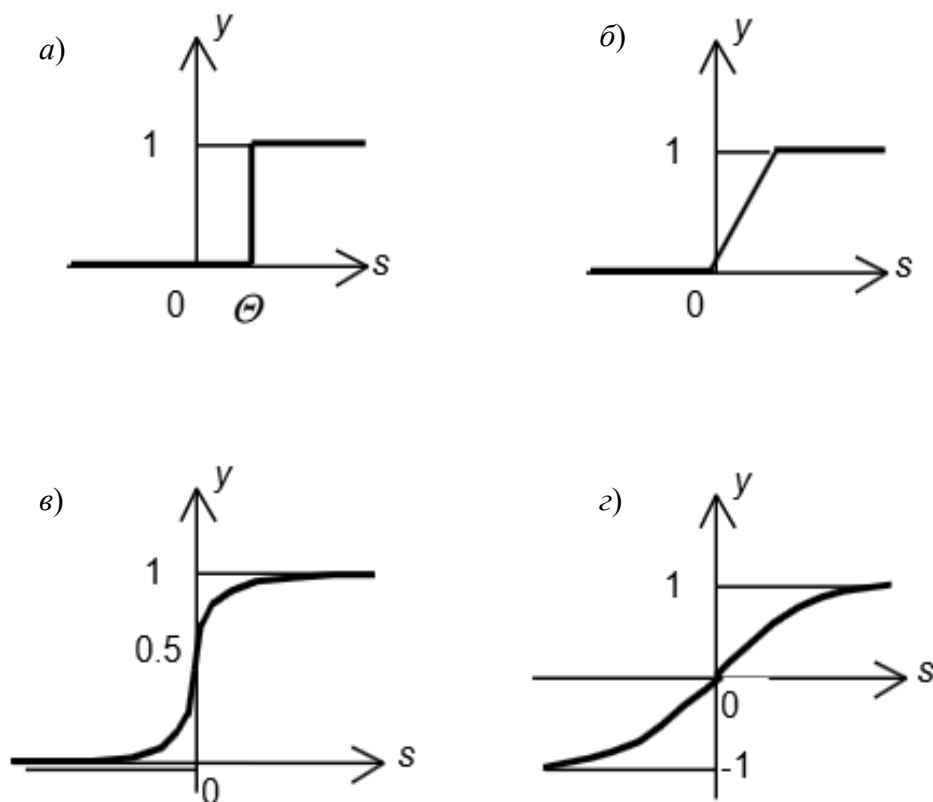


Рисунок 8.3 – Примеры активационных функций нейрона

Типы нейронов ИНС:

- входные нейроны, кодируют входные воздействия и передают «взвешенные» данные на промежуточные нейроны нейронной сети;
- промежуточные нейроны выполняют основные преобразования данных в нейронной сети;
- выходные нейроны, выполняют результирующие преобразования; их выходные значения являются выходами нейронной сети.

8.1.3 Классификация искусственных нейронных сетей и их свойства.

В общем случае можно выделить три фундаментальных класса архитектур ИНС:

- 1) однослойные сети прямого распространения;

- 2) многослойные сети прямого распространения;
- 3) рекуррентные сеть.

В однослойных сетях прямого распространения (ациклических сетях) имеется входной слой элементов, информация от которого передается на выходной слой нейронов (рисунок 8.4).

Многослойные сети прямого распространения характеризуются наличием одного или нескольких скрытых слоев нейронов (рисунок 8.5). Такая сеть позволяет выделять глобальные свойства данных с помощью локальных соединений за счет наличия дополнительных синаптических связей и повышения уровня взаимодействия нейронов. Способность скрытых нейронов выделять статистические зависимости высокого порядка особенно существенна, когда размерность признакового пространства (число нейронов входного слоя) достаточно велика.

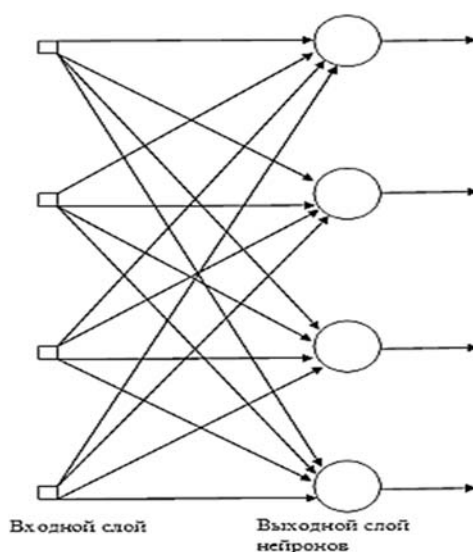


Рисунок 8.4 – Однослойная сеть прямого распространения

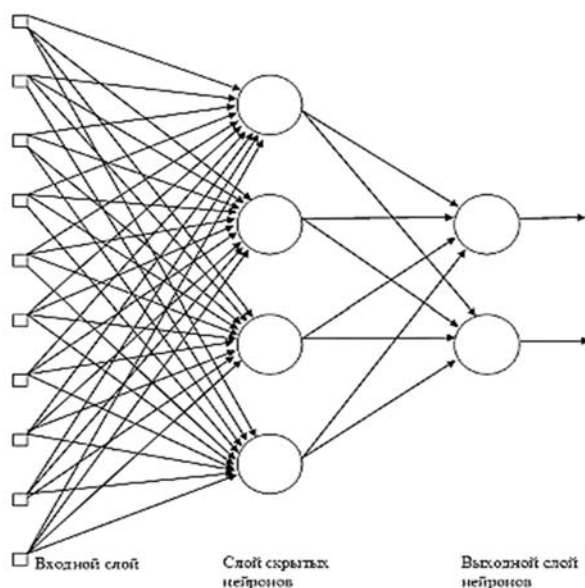


Рисунок 8.5 – Многослойная сеть прямого распространения

Элементы входного слоя сети формируют входные сигналы, поступающие на нейроны второго (скрытого) слоя и т. д., вплоть до выходного слоя. Обычно нейроны каждого из последующих слоев ИНС используют в качестве входных сигналов, выходные сигналы нейронов только предыдущего слоя. Набор выходных сигналов нейронов выходного (последнего) слоя сети определяет общий отклик ИНС на вектор входных сигналов.

Нейронная сеть, показанная на рисунке 8.5, считается полносвязной в том смысле, что все нейроны каждого слоя соединены со всеми нейронами смежных слоев. Если отдельные синаптические связи отсутствуют, такая сеть считается неполносвязной.

Рекуррентная нейронная сеть отличается от сети прямого распространения наличием, по крайней мере, одной обратной связи. Например, рекуррентная сеть может состоять из единственного слоя нейронов, каждый из которых направляет свой выходной сигнал на входы всех остальных нейронов слоя (рисунок 8.6). На рисунке 8.7 показана рекуррентная сеть со слоем скрытых нейронов. Здесь обратные связи исходят как из скрытых, так и из выходных нейронов.

Наличие обратных связей в сетях оказывает непосредственное влияние на способность таких сетей к обучению и на их производительность. Более того, обратная связь подразумевает использование элементов единичной задержки (они обозначены как z^{-1}), что приводит к нелинейному динамическому поведению ИНС, если, конечно, в сети содержатся нелинейные нейроны.

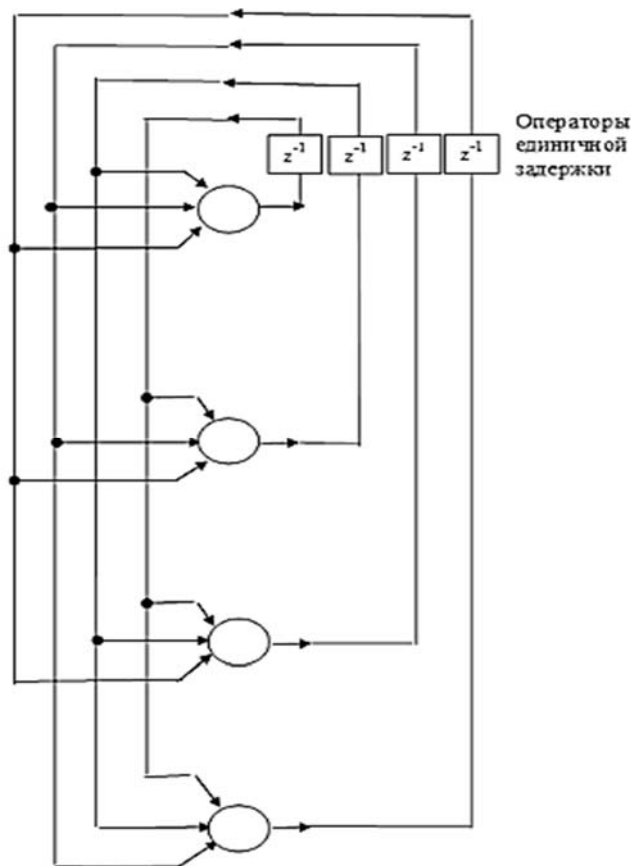


Рисунок 8.6 – Рекуррентная сеть без слоя скрытых нейронов

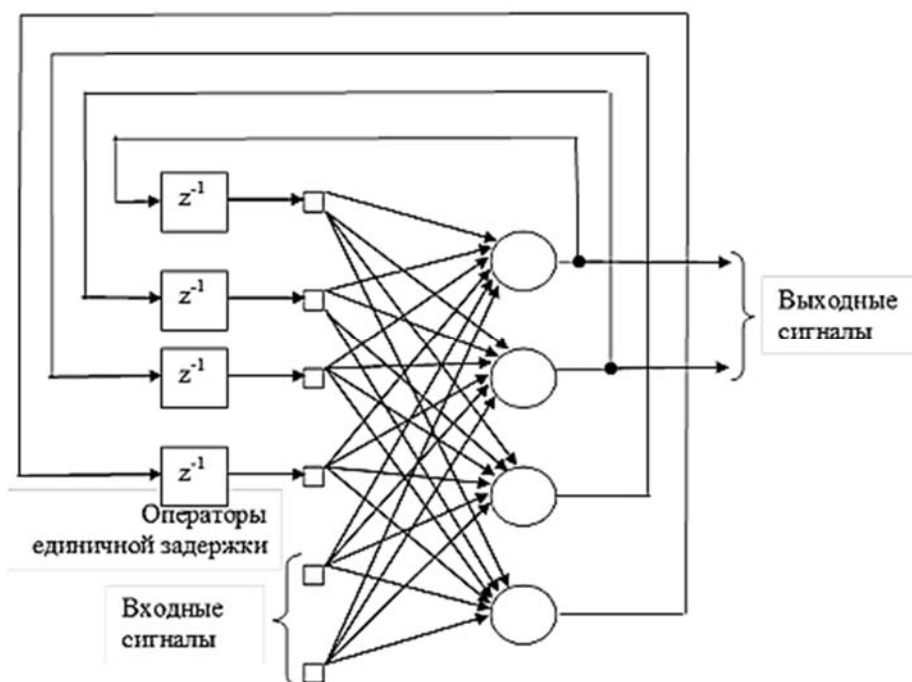


Рисунок 8.7 – Рекуррентная сеть со слоем скрытых нейронов

8.2 Задания для самостоятельного выполнения

Реализация искусственного нейрона с различными активационными функциями.

Контрольные вопросы

- 1 Из каких элементов состоит формальный нейрон?
- 2 Назовите функции активации нейрона.
- 3 Какие типы нейронов в искусственной нейронной сети можно выделить в зависимости от выполняемых ими функций?

Список литературы

- 1 **Борисов, В. В.** Экспертные системы: учебное пособие / В. В. Борисов, А. В. Бобряков, А. Е. Мисник. – Смоленск: Универсум, 2021. – 110 с.
- 2 **Сосинская, С. С.** Представление знаний в информационной системе. Методы искусственного интеллекта и представления знаний : учебное пособие / С. С. Сосинская. – Старый Оскол : ТНТ, 2019. – 216 с.