

МЕЖГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«БЕЛОРУССКО-РОССИЙСКИЙ УНИВЕРСИТЕТ»

Кафедра «Автоматизированные системы управления»

ИНТЕЛЛЕКТУАЛЬНЫЕ ИНФОРМАЦИОННЫЕ СИСТЕМЫ

*Методические рекомендации к лабораторным работам
для студентов специальности
6-05-0612-03 «Системы управления информацией»
дневной и заочной форм обучения*



Могилев 2026

УДК 004:519.25
ББК 32.937-018:22.18
И23

Рекомендовано к изданию
учебно-методическим отделом
Белорусско-Российского университета

Одобрено кафедрой «Автоматизированные системы управления»
«24» марта 2026 г., протокол № 8

Составитель канд. техн. наук, доц. Е. М. Борчик

Рецензент канд. техн. наук, доц. В. В. Кутузов

Методические рекомендации к выполнению лабораторных работ по дисциплине «Интеллектуальные информационные системы» предназначены для студентов специальности 6-05-0612-03 «Системы управления информацией» дневной и заочной форм обучения.

Учебное издание

ИНТЕЛЛЕКТУАЛЬНЫЕ ИНФОРМАЦИОННЫЕ СИСТЕМЫ

Ответственный за выпуск	А. И. Якимов
Корректор	А. А. Подошевки
Компьютерная верстка	Н. П. Полевничая

Подписано в печать . Формат 60×84/16. Бумага офсетная. Гарнитура Таймс.
Печать трафаретная. Усл. печ. л. . Уч.-изд. л. . Тираж 21 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, 2026

Содержание

Введение.....	4
1 Лабораторная работа № 1. Формирование базы знаний с помощью экспертов.....	5
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.....	15
5 Лабораторная работа № 5. Формирование базы знаний с помощью фреймов.....	19
6 Лабораторная работа № 6. Экспертные системы с неопределенными знаниями и байесовские сети доверия.....	23
7 Лабораторная работа № 7. Экспертные системы извлечения знаний на основе генетических алгоритмов.....	26
8 Лабораторная работа № 8. Разработка искусственного нейрона.....	27
9 Лабораторная работа № 9. Нейронные сети и их свойства.....	29
10 Лабораторная работа № 10. Разработка алгоритма обучения нейронных сетей.....	30
11 Лабораторная работа № 11. Искусственные нейронные сети встречного распространения.....	32
12 Лабораторная работа № 12. Искусственные нейронные сети радиальных базисных функций.....	33
13 Лабораторная работа № 13. Искусственные нейронные сети с анализом главных компонент.....	34
14 Лабораторная работа № 14. Каскадные искусственные нейронные сети.....	36
Список литературы.....	37

Введение

Целью учебной дисциплины является формирование у специалистов умений обоснованно применять методы, приёмы и технологии построения интеллектуальных информационных систем.

Если говорить более конкретно, изучение данной дисциплины преследует следующие ключевые цели.

1 Формирование фундаментальных знаний.

Понимание принципов ИИ: изучение основ искусственного интеллекта, истории его развития и базовых подходов (логический подход, нейронные сети, эволюционные алгоритмы, нечеткая логика и др.).

Изучение архитектуры ИИС: понимание того, из чего состоит интеллектуальная система (база знаний, механизм логического вывода, подсистема объяснений, интерфейс пользователя).

2 Освоение методов представления знаний.

Обучение студентов тому, как формализовать человеческие знания для компьютера.

Изучение моделей представления знаний: продукционных моделей, фреймов, семантических сетей, онтологий.

3 Развитие навыков проектирования и разработки.

Выбор инструментов: освоение языков программирования и специализированных сред разработки (например, Python с библиотеками Scikit-learn, TensorFlow, PyTorch или специализированных оболочек экспертных систем).

Решение прикладных задач: приобретение опыта в разработке систем, способных к обучению, поиску закономерностей в данных (Data Mining), принятию решений в условиях неопределенности и обработке естественного языка (NLP).

4 Понимание специфики интеллектуальных систем.

Принятие решений: обучение методам обработки нечеткой информации и работе с неполными данными.

Интерпретируемость: понимание важности объяснимости результатов работы системы («почему система приняла такое решение?»), что критически важно в медицине, финансах и управлении.

5 Аналитическая подготовка.

Развитие умения анализировать предметную область, оценивать целесообразность применения методов ИИ для решения конкретных бизнес- или технических задач.

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

Программное обеспечение: Microsoft Word.

Цель работы: освоить методику извлечения знаний из экспертных описаний; выполнить первичную структуризацию знаний в виде текстового документа; подготовить глоссарий, правила и факты для будущей базы знаний.

Теоретические сведения.

Инженер по знаниям взаимодействует с экспертами (не менее трех экспертов) для выявления ключевых факторов, влияющих на решение. Результаты фиксируются в документе Word: глоссарий, текстовые правила, таблицы. Важно отделить условия (посылки) от заключений и унифицировать терминологию.

Пример 1

Предметная область – решение о выдаче потребительского кредита.

Эксперты: специалист по кредитованию, аналитик рисков, сотрудник отдела обслуживания.

Опросник.

Какие параметры клиента наиболее важны?

Какие значения дохода и стажа минимально допустимы?

Как просрочки влияют на решение?

В каких случаях требуется обеспечение?

Выявленные факторы: доход, стаж, кредитная история, сумма кредита, обеспечение.

Классы знаний.

Финансовые характеристики клиента.

Кредитная история клиента.

Параметры кредитного договора.

Способы обеспечения кредита.

Правила (естественный язык).

1 Если доход ниже минимального порога, то требуется дополнительная проверка.

2 Если есть просрочки, то банк может отказать или предложить худшие условия.

3 Если сумма кредита велика, то требуется залог или поручитель.

Факты: конкретный клиент – доход 1600 BYN, стаж 14 месяцев, нет просрочек, запрос 400000 BYN на 720 месяцев.

Результат: документ Word *.docx с глоссарием, таблицей факторов, правилами и фактами.

Пример 2

Предметная область – «Авторазборка» (прием, разборка и утилизация бывших в употреблении автомобилей с целью продажи запчастей и металлолома).

Эксперты: в демонстрационных целях в роли эксперта выступил специализированный чат-бот, подключённый к поисковым системам (Google Web Search), модели компьютерного зрения (GPT-4V), репозиториям кода (GitHub) и научным базам (Google Scholar). Взаимодействие велось через мессенджер Telegram. Несмотря на нестандартный источник, методика извлечения знаний полностью сохранена: формулировались вопросы, анализировались ответы, выявлялись ключевые факторы и правила.

Опросник (фрагмент переписки).

Какие обязательные этапы проходит автомобиль на авторазборке?

Какие документы требуются для законного приёма машины?

По каким критериям оценивается экономическая целесообразность приёма автомобиля?

Что является главным показателем ликвидности кузова?

Где разрешено утилизировать остатки после разборки?

Выявленные факторы: наличие правоустанавливающих документов, соотношение стоимости приёма и оценочной прибыли, масса кузова как ключевой параметр ликвидности, состояние автомобиля (пригодность к восстановлению), доступ к лицензированным полигонам ТБО.

Классы знаний: приём авто, проверка авто, демонтаж, продажа, утилизация.

Правила (естественный язык).

1 Если приём автомобиля не сопровождается договором купли-продажи или доверительной распиской, то такой автомобиль не принимается.

2 Если суммарные затраты на демонтаж и транспортировку превышают ожидаемую выручку от продажи запчастей и металлолома, то приём экономически нецелесообразен.

Основной критерий ликвидности – вес автомобиля: если масса кузова позволяет получить стоимость металлолома выше цены приёма и сопутствующих расходов, то автомобиль считается ликвидным.

Утилизация неметаллических остатков должна выполняться исключительно на официальных полигонах ТБО.

Факты (конкретный экземпляр):

– автомобиль Ford Mondeo, масса кузова 890 кг, стоимость приёма 600 р;

– оценочная стоимость запчастей и металла превышает общие расходы (приём + демонтаж);

– при приёме владельцем оформлена доверительная расписка;

– состояние автомобиля признано не подлежащим восстановлению.

Результат: документ MS Word KB_Expert_Autorecycling.docx, содержащий глоссарий терминов «ликвидность», «доверительная расписка», «оценочная прибыль», «полигон ТБО», таблицу выявленных классов знаний, пронумерованные правила и фактографическую базу для выбранного автомобиля. Этот документ

служит основой для последующей формализации знаний в табличной, продукционной или фреймовой форме.

Практические задания.

1 Проведите интервью с экспертами по теме, связанной с промышленным производством, например – «Диагностика неисправностей станка с ЧПУ и принятие решения о виде ремонта». Роль экспертов могут выполнять приглашённые специалисты (инженер по эксплуатации, мастер цеха, технолог), либо, в учебных целях, чат-боты, обученные на технической документации. Разработайте опросник не менее чем из пяти вопросов, направленных на выявление ключевых факторов, влияющих на решение о немедленной остановке станка или допустимости продолжения работы.

2 На основе полученных ответов сформируйте в документе Word:

- глоссарий (не менее восьми терминов);
- четыре-пять продукционных правил в формате «Если ... , То...».

3 Опишите один конкретный производственный инцидент (объект исследования) – реальный или смоделированный набор фактов, характеризующих состояние конкретного станка в определённый момент.

Укажите, какие из сформулированных вами правил срабатывают для данного инцидента, и какое итоговое решение (действие) должна принять экспертная система (например, «немедленный останов и замена шпиндельного узла»).

Приведите выдержки из правил, которые активировались, и обоснуйте выбор.

Контрольные вопросы

- 1 Какие этапы включает инженерия знаний при работе с экспертом?
- 2 Почему необходимо фиксировать глоссарий до начала формализации?
- 3 Какие инструменты Word удобны для структурирования правил?
- 4 Чем активный метод извлечения знаний отличается от пассивного?

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

Программное обеспечение: Microsoft Excel.

Цель работы: представить знания в виде классической таблицы решений; освоить разделение условий и решений в табличной форме; реализовать поиск подходящего столбца по заданным фактам.

Теоретические сведения.

Для представления знаний предложена табличная форма (таблица 2.1), получившая название таблицы решений. Идея этого подхода очень проста и

заключается в использовании табличной формы описания проблемы и способа ее решения.

Таблица 2.1 – Структура таблицы решений

Условие	Объект			
	O_1	O_2	...	O_k
U_1	1	0	...	1
...	...	...	...	...
U_i	1	1	...	0
Действие				
D_1	0	1	...	1
...	...	...	...	...
D_j	1	0	...	1

В этой таблице предполагается наличие двух основных частей: строк условий U_i 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.

Использование табличной формы позволяет наглядно представить взаимосвязь между параметрами задачи и возможными решениями. Такой способ удобен для анализа различных сценариев и последующего формирования правил принятия решений.

Таблица решений содержит верхнюю секцию условий и нижнюю – действий (решений). Каждый столбец описывает одно правило. Прочерк «—» означает безразличное значение. Такая форма удобна для проверки полноты и непротиворечивости знаний.

Пример 1

В качестве предметной области была выбрана система принятия решения о выдаче банковского кредита физическим лицам.

На основе данных, полученных от экспертов в предыдущем этапе исследования, были выделены основные параметры, влияющие на решение банка: уровень дохода клиента, стаж работы, наличие просрочек по кредитам, размер запрашиваемого кредита, наличие обеспечения (таблицы 2.2 и 2.3).

Таблица 2.2 – Таблица исходных данных

Параметр	Объект 1	Объект 2	Объект 3	Объект 4	Объект 5
Доход, р.	1600	1400	1100	900	1700
Стаж работы, мес.	14	10	8	4	18
Просрочки	0	0	0	1	0

Окончание таблицы 2.2

Параметр	Объект 1	Объект 2	Объект 3	Объект 4	Объект 5
Сумма кредита, р.	20000	25000	15000	10000	30000
Наличие налога	Да	Нет	Нет	Нет	Да
Решение	Выдать кредит	Поручитель	Потребовать залог	Отказать	Выдать кредит

Таблица 2.3 – Ход решения

Объект	Ход решения
1	Хороший доход, достаточный стаж, нет просрочек, поэтому – выдать кредит
2	Средний доход и нет залога, поэтому нужен поручитель
3	Доход ниже среднего, поэтому требуется обеспечение – потребовать залог
4	Есть просрочки и маленький стаж, поэтому – отказать
5	Высокий доход и есть залог, поэтому кредит одобрен – выдать кредит

На основе экспертных данных создана таблица 2.4.

Таблица 2.4 – Таблица решений

Условие	Параметр	Объект 1	Объект 2	Объект 3	Объект 4	Объект 5
Доход	Высокий	1	0	0	0	1
	Средний	0	1	0	0	0
	Ниже среднего	0	0	1	0	0
	Низкий	0	0	0	1	0
Стаж	Достаточный	1	–	–	0	–
	Маленький	0	–	–	1	–
Обеспечение	Просрочки есть	0	–	–	1	–
	Нет залога	–	1	0	–	0
	Есть залог	–	0	0	–	1
	Требуется	–	0	1	–	0
Решение	Выдать кредит	1	0	0	0	1
	Нужен поручитель	0	1	0	0	0
	Потребовать залог	0	0	1	0	0
	Отказать	0	0	0	1	0

Пример 2

Предметная область – «Авторазборка» (прием и разборка бывших в употреблении автомобилей для продажи запасных частей и металлолома).

На основе знаний, полученных от экспертов в лабораторной работе № 1, были выделены три основных фактора, определяющих решение о приёме автомобиля:

- 1) ликвидность автомобиля (определяется по весу кузова относительно стоимости приёма);
- 2) рентабельность запчастей (возможность выгодно продать узлы и детали);
- 3) наличие правоустанавливающего документа (договор купли-продажи или доверительная расписка).

Экспертные оценки для нескольких характерных ситуаций были сведены в неформальную таблицу (таблица 2.5), напоминающую протокол анализа «ходов решения».

Таблица 2.5 – Таблица решений

Объект	Ход решения
O1	Авто ликвидное, запчасти рентабельны, договор есть, поэтому оформление не требуется, автомобиль принимается
O2	Авто неликвидное, запчасти нерентабельны, договор есть, поэтому оформление не требуется, автомобиль не принимается
O3	Авто неликвидное, запчасти рентабельны, договора нет, поэтому требуется оформление, автомобиль принимается
O4	Авто неликвидное, запчасти рентабельны, договор есть, поэтому оформление не требуется, автомобиль принимается

Формализованная таблица решений, построенная в MS Excel, представлена в таблице 2.6. В верхней части перечислены условия (ликвидность, рентабельность, наличие договора), в нижней – действия (приём автомобиля, необходимость оформления договора). Каждый столбец соответствует одному правилу. Единица означает выполнение условия или действия, ноль – невыполнение. Прочерки отсутствуют, поскольку все четыре правила чётко заданы.

Таблица 2.6 – Таблица решений для авторазборки

Параметр		Объект 1	Объект 2	Объект 3	Объект 4
Условие	Авто ликвидное	1	0	0	0
	Запчасти рентабельны	1	0	1	1
	Договор имеется	1	1	0	1
Действие	Принять автомобиль	1	0	1	1
	Оформить договор	0	0	1	0

С помощью условного форматирования Excel можно подсветить столбец, условия которого полностью совпадают с введёнными в отдельной области фактами. Так, например, при вводе фактов «неликвидное авто, рентабельные запчасти, договор отсутствует» автоматически выделяется столбец объекта 3, а решение гласит: принять автомобиль и оформить договор.

Таким образом, табличная форма решений позволила компактно и наглядно представить логику принятия решений в предметной области авторазборки,

а её реализация в среде MS Excel обеспечила возможность автоматизированного поиска подходящего правила.

Практические задания.

1 В Excel постройте таблицу решений для кредитной задачи, используя параметры: доход, стаж, просрочки, сумма кредита, обеспечение. Создайте не менее шести столбцов-правил (строк-правил).

2 В отдельной области «Ввод фактов» организуйте поля для значений параметров. С помощью формул (например, СЧЁТЕСЛИ) и условного форматирования выделите цветом столбец, условия которого полностью выполняются.

3 Добавьте в таблицу новое правило, например, «если доход средний и есть залог – выдать кредит с повышенной ставкой», и проверьте его работу.

Контрольные вопросы

1 Какова структура классической таблицы решений?

2 Почему прочерк может приводить к неоднозначности при поиске решения?

3 Как с помощью формул Excel (без VBA) найти первый подходящий столбец?

4 Какие преимущества табличная форма имеет перед текстовым списком правил?

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

Программное обеспечение: Microsoft Excel (формулы, без макросов).

Цель работы: представить знания в виде системы продукций «Если ..., То ...»; реализовать логический вывод через формулы Excel; ознакомиться с расчётными показателями, необходимыми для правил.

Теоретические сведения.

Продукционное правило состоит из условия (логического выражения над фактами) и действия. В Excel условия можно записать в отдельных ячейках, а решение вычислять цепочкой функций ЕСЛИ, И, ИЛИ, НЕ, ИНДЕКС, ПОИСКПОЗ. Для сложных расчётов вводятся промежуточные ячейки (ежемесячный платёж, долговая нагрузка, коэффициент обеспечения).

Пример 1

Исходные параметры: доход, стаж, просрочки, сумма кредита, процентная ставка, срок, залог.

Расчетные показатели.

1 Ежемесячный платеж = (Сумма $\times$ (1 + ставка)) / Срок.

2 Долговая нагрузка = Платёж / Доход.

3 Коэффициент обеспечения = Залог / Сумма.

4 Кредитный коэффициент = (1 – Долговая нагрузка) + Коэффициент обеспечения.

Правила (реализованы в Excel).

1 Выдать кредит = стаж ≥ 6 И долговая нагрузка $\leq 0,45$ И кредитный коэффициент $\geq 0,55$ И просрочки = 0.

2 Потребовать поручителя = стаж < 6 ИЛИ просрочки > 0 .

3 Потребовать дополнительный залог = коэффициент обеспечения $< 0,4$ И долговая нагрузка $> 0,45$.

4 Уменьшить сумму кредита = коэффициент обеспечения $< 0,4$.

Формула для выдачи рекомендации (упрощённо).

= ЕСЛИ (И (стаж ≥ 6 ; долговая нагрузка $\leq 0,45$; кредитный коэффициент $\geq 0,55$; просрочки = 0); «Выдать кредит»;

ЕСЛИ (ИЛИ (стаж < 6 ; просрочки > 0); «Потребовать поручителя»; ...))

На отдельных листах Excel представлены: таблица исходных данных, блок расчетов, блок правил и итоговое решение.

Продукционные правила представлены на рисунке 3.1.

Решение представлено на рисунке 3.2.

	А	В
1	Параметр	Действие
2	(Стаж ≥ 6) $\wedge$ (Долговая нагрузка ≤ 0.45) $\wedge$ (Кредитный коэффициент ≥ 0.55) $\wedge$ (Просрочки = 0)	Выдать кредит
3	(Стаж < 6) $\wedge$ (Просрочки = 1)	Потребовать поручителя
4	(Коэффициент обеспечения < 0.4) $\wedge$ (Долговая нагрузка > 0.45)	Потребовать дополнительный залог
5	(Коэффициент обеспечения < 0.4)	Уменьшить сумму кредита
6		
7		
8	Ежемесячный платёж = (Сумма кредита * (1 + процентная ставка)) / срок	
9	Долговая нагрузка = платёж / доход	
10	Коэффициент обеспечения = залог / сумма кредита	
11	Кредитный коэффициент = (1 - долговая нагрузка) + коэффициент обеспечения	

Рисунок 3.1 – Продукционные правила

	A	B	C	D
1	Параметр	Объект 1	Объект 2	Объект 3
2	Доход (BYN)	1600	1500	1200
3	Стаж (месяцы)	14	10	4
4	Просрочки	0	0	0
5	Долговая нагрузка	0,388888889	0,373333333	0,388888889
6	Сумма кредита (BYN)	400000	300000	200000
7	Залог (BYN)	200000	150000	80000
8	Процентная ставка (%)	12	12	12
9	Срок кредита (месяцы)	720	600	480
10	Ежемесячный платёж	622,2222222	560	466,6666667
11	Коэффициент обеспечения	0,5	0,5	0,4
12	Кредитный коэффициент	1,111111111	1,126666667	1,011111111
13				
14	Условие	Объект 1	Объект 2	Объект 3
15	Стаж ≥ 6	1	1	0
16	Долговая нагрузка ≤ 0.45	1	1	1
17	Кредитный коэффициент ≥ 0.55	1	1	1
18	Серьёзные просрочки = 0	0	0	0
19	Долговая нагрузка > 0.45	0	0	0
20	Коэффициент обеспечения < 0.4	0	0	0
21				
22	Результат	Объект 1	Объект 2	Объект 3
23	Выдать кредит	1	1	0
24	Потребовать поручителя	0	0	1
25	Потребовать дополнительный залог	0	0	0
26	Уменьшить сумму кредита	0	0	0

Рисунок 3.2 – Табличное решение

Пример 2

Предметная область – управление мобильным роботом, предназначенным для транспортировки грузов между тремя комнатами А, В и С. Робот может находиться в любой из комнат, поднимать и класть коробку, перемещаться в соседнюю комнату.

Целевая задача: перенести коробку из комнаты А в комнату С, последовательно выполняя элементарные действия.

На основе анализа задачи экспертами были выделены ключевые факторы, описывающие текущее состояние системы:

- местоположение робота (А, В или С);
- местоположение коробки (в одной из комнат или у робота);
- состояние захвата (держит ли робот коробку).

Для каждого характерного сочетания условий определено одно рекомендуемое действие, приближающее систему к цели. Неформальная таблица «ходов решения» приведена в таблице 3.1.

Формализованная таблица решений, построенная в MS Excel, представлена в таблице 3.2. В верхней секции перечислены условия (семь предикатов), в нижней – возможные действия. Каждый столбец соответствует одному правилу (объекту). Единица означает истинность условия или необходимость выполнения действия, ноль – ложь или отсутствие действия.

Таблица 3.1 – Данные экспертов в табличной форме

Объект	Ход решения
O1	Робот в А, коробка в А, не держит → поднять коробку
O2	Робот в А, держит коробку → переместиться в В
O3	Робот в В, держит коробку → переместиться в С
O4	Робот в С, держит коробку → положить коробку
O5	Робот в С, коробка в С, не держит → цель достигнута

Таблица 3.2 – Таблица решений для перемещения коробки роботом

Параметр		Объект 1	Объект 2	Объект 3	Объект 4	Объект 5
Условие	Робот в А	1	1	0	0	0
	Робот в В	0	0	1	0	0
	Робот в С	0	0	0	1	1
	Коробка в А	1	0	0	0	0
	Коробка в В	0	0	0	0	0
	Коробка в С	0	0	0	0	1
	Робот держит коробку	0	1	1	1	0
Действие	Поднять коробку	1	0	0	0	0
	Переместиться в В	0	1	0	0	0
	Переместиться в С	0	0	1	0	0
	Положить коробку	0	0	0	1	0
	Завершить	0	0	0	0	1

С помощью условного форматирования в Excel можно автоматически выделить столбец, условия которого полностью совпадают с текущим состоянием, введенным в отдельной области листа. Например, при фактах «робот в А, коробка в А, не держит» система подсветит столбец объекта O1 и укажет действие «Поднять коробку». После выполнения действия состояние изменится, и для следующего шага будет активирован другой столбец, что позволяет реализовать пошаговое управление роботом.

Практические задания.

1 Создайте новую книгу Excel. На листе «Таблица Решений» постройте таблицу, аналогичную таблице 3.2. Предусмотрите не менее пяти столбцов-правил для различных комбинаций состояний робота и коробки. Четко отделите секцию условий от секции действий.

2 Реализуйте вспомогательную область «Ввод фактов», куда пользователь заносит текущее состояние (комнату робота, комнату с коробкой, наличие захвата). С помощью формул (например, СЧЕТЕСЛИ и И) и условного форматирования добейтесь, чтобы столбец, удовлетворяющий всем условиям, автоматически выделялся цветом. В отдельной ячейке выведите описание действия, которое робот должен выполнить (можно использовать функцию ЕСЛИ

и ИНДЕКС/ПОИСКПОЗ для возврата текста из заранее подготовленного списка).

3 Добавьте в таблицу новый сценарий: робот находится в В, коробка в С, захват пуст. Определите, какое действие должен выполнить робот (например, переместиться в С). Дополните таблицу решений соответствующим столбцом. Проверьте, что после добавления правила система корректно обрабатывает все введенные факты, включая новый. Опишите в отчете, как изменилась таблица и как вы обеспечили непротиворечивость (отсутствие двух срабатывающих столбцов).

Контрольные вопросы

- 1 Из каких компонентов состоит производственная система?
- 2 Каким образом в Excel можно смоделировать прямой логический вывод?
- 3 Зачем вводятся расчетные показатели и почему их неудобно хранить непосредственно в правилах?
- 4 Каковы ограничения чисто формульного подхода при увеличении числа правил?

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

Программное обеспечение: MS Visio / Draw.io; допускается предварительное описание в Excel.

Цель работы: представить знания о кредитном скоринге в виде семантической сети; освоить основные типы отношений a kind of, is-a; научиться анализировать сеть для ответа на информационные запросы.

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 – ориентированный граф, узлы которого соответствуют понятиям, а дуги – отношениям. Сеть позволяет наглядно отобразить взаимосвязи и использовать наследование (АКО, IS-A). В кредитной задаче можно

выделить объекты «Заёмщик», «Кредит», «Решение банка» и множество параметров и расчётных показателей.

Пример 1

Узлы: Заёмщик, Доход, Стаж, Просрочки, Залог, Поручитель, Кредит, Сумма кредита, Процентная ставка, Срок кредита, Ежемесячный платёж, Долговая нагрузка, Коэффициент обеспечения, Кредитный коэффициент, Решение банка, Выдать кредит, Потребовать поручителя, Дополнительный залог, Уменьшить сумму кредита.

Семантическая сеть представлена в табличной форме на рисунке 4.1 в программе Excel, на рисунках 4.2 и 4.3 – в графической форме, выполненной в программе draw.io.

	A	B	C	D
1	Объект 1	Связь	Объект 2	
2	Заёмщик	имеет	доход	
3	Заёмщик	имеет	стаж работы	
4	Заёмщик	имеет	залог	
5	Заёмщик	имеет	поручитель	
6	Заёмщик	имеет	просрочки по другим кредитам	
7	Кредит	имеет	сумма кредита	
8	Кредит	имеет	процентная ставка	
9	Кредит	имеет	срок кредита	
10	Процентная ставка + Сумма кредита + Срок кредита	формирует	ежемесячный платёж	
11	Ежемесячный платёж + Доход	формирует	долговая нагрузка	
12	Залог + сумма кредита	формирует	Коэффициент обеспечения	
13	Долговая нагрузка + Коэффициент обеспечения	формирует	Кредитный коэффициент	
14	Кредитный коэффициент + Просрочки + Стаж работы + поручитель	влияют на	Решение банка	
15				
16	Решение банка			
17	Выдать кредит			
18	Потребовать поручителя у заёмщика			
19	Потребовать дополнительный залог			
20	Уменьшить сумму кредита			

Рисунок 4.1 – Семантическая сеть в табличной форме

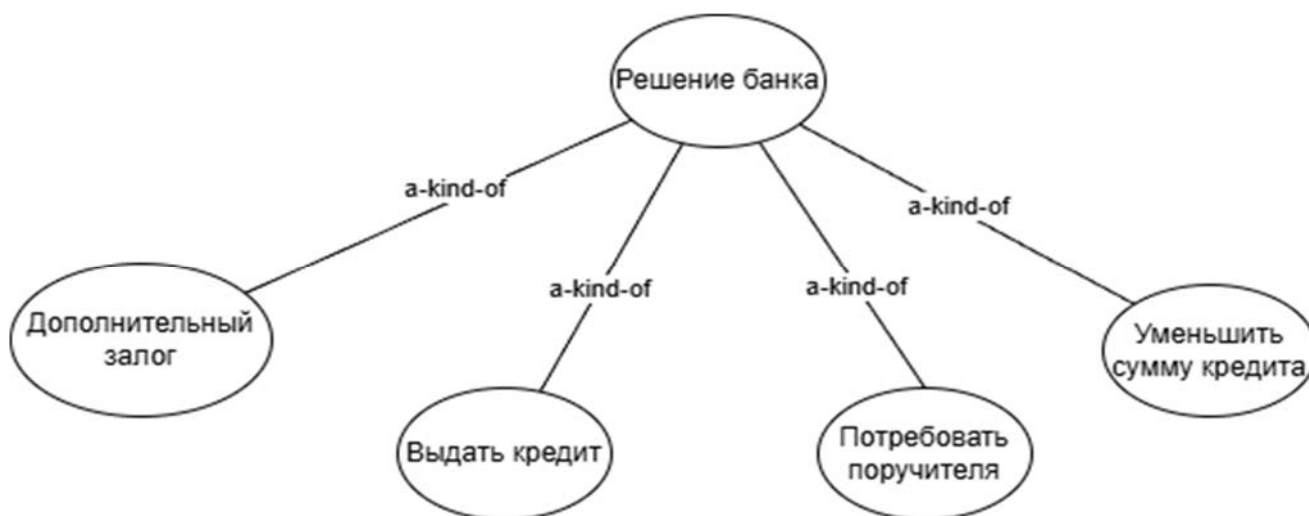


Рисунок 4.2 – Семантическая сеть в графической форме № 1

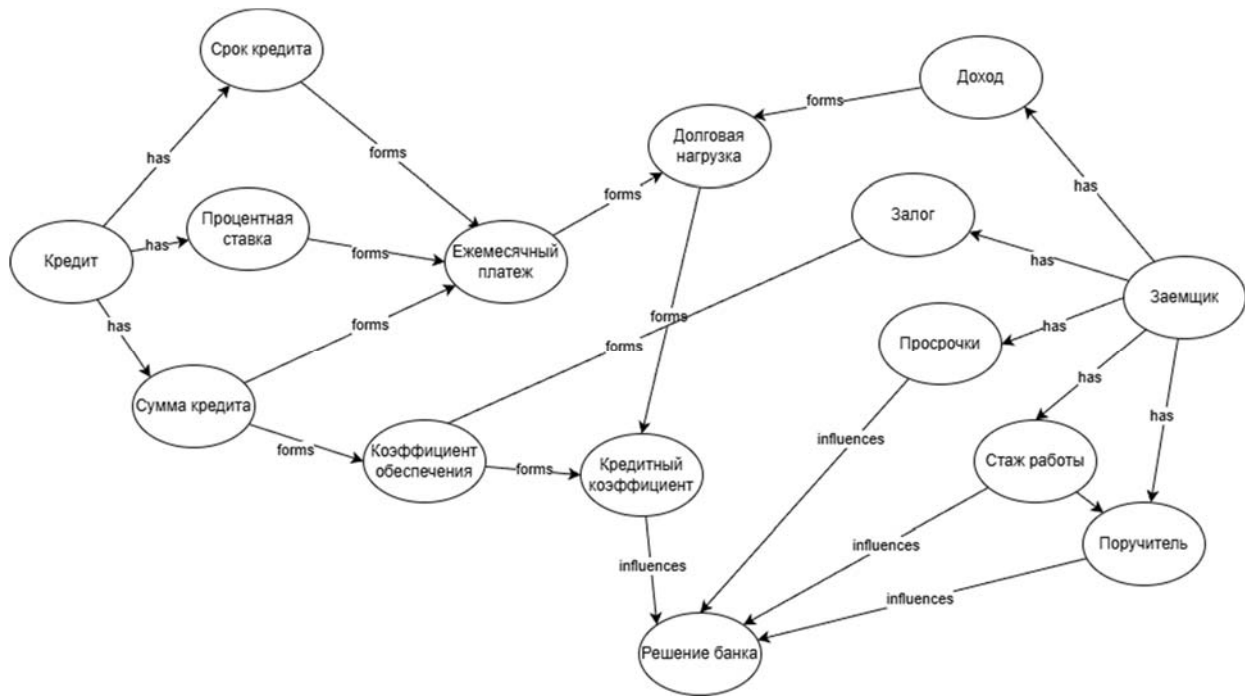


Рисунок 4.3 – Семантическая сеть в графической форме № 2

Связи между узлами

Связи показывают отношения между понятиями. Ниже будут описаны в виде Объект 1 - > (тип_связи) - > Объект 2.

Заемщик → имеет (has) → Доход.

Заемщик → имеет (has) → Стаж работы.

Заемщик → имеет (has) → Залог.

Заемщик → имеет (has) → Поручитель.

Заемщик → имеет (has) → Просрочки по другим кредитам.

Кредит → имеет (has) → Сумма кредита.

Кредит → имеет (has) → Процентная ставка.

Кредит → имеет (has) → Срок кредита.

Процентная ставка + Сумма кредита + Срок кредита → формирует (forms) → Ежемесячный платеж.

Ежемесячный платеж + Доход → формирует (forms) → Долговая нагрузка.

Залог + Сумма кредита → формирует (forms) → Коэффициент обеспечения.

Долговая нагрузка + Коэффициент обеспечения → формируют (forms) → Кредитный коэффициент.

Кредитный коэффициент, Просрочки, Стаж работы, поручитель → влияют на (influences) → Решение банка.

Решения банка в сети.

Связи принятия решения.

Выдать кредит → а kind of → Решение банка.

Потребовать поручителя → а kind of → Решение банка.

Потребовать дополнительный → а kind of → Решение банка залог.

Уменьшить сумму кредита → а kind of → Решение банка.

Пример 2

На рисунке 4.4 приведен пример семантической сети.

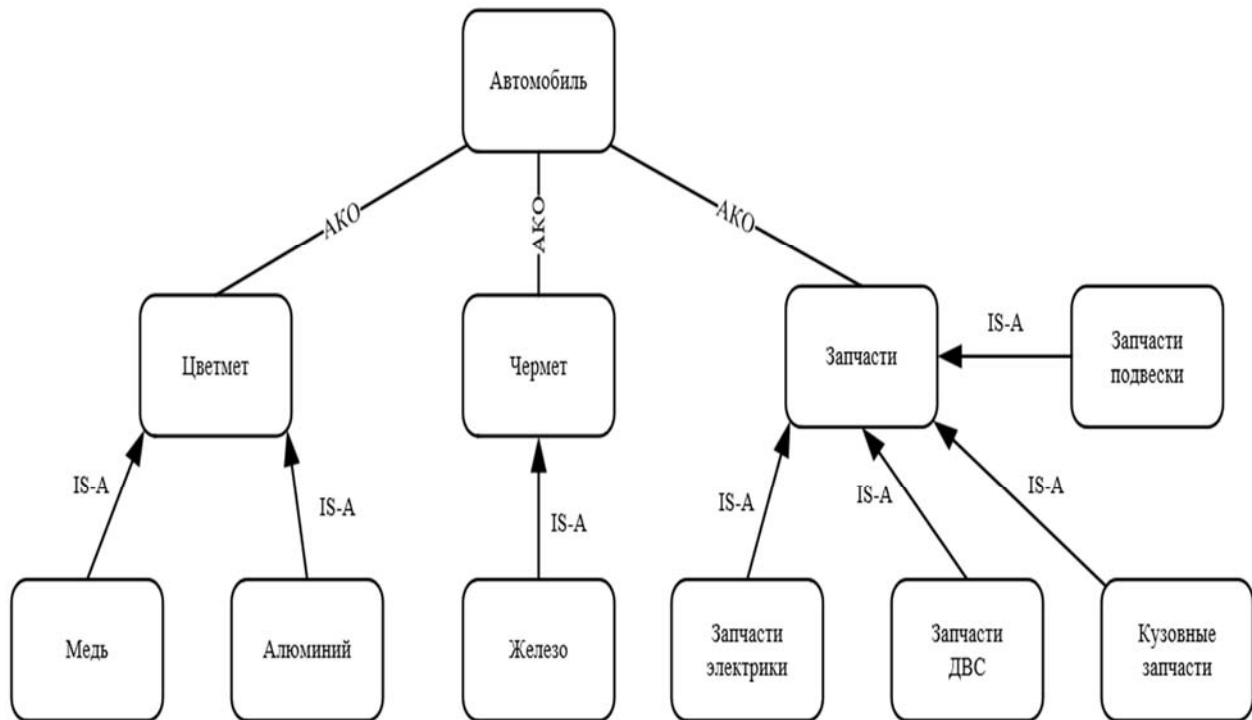


Рисунок 4.4 – Семантическая сеть в графической форме № 1

Практические задания.

1 В draw.io (или Visio) постройте семантическую сеть кредитного скоринга, включив все перечисленные узлы и связи (обязательно использовать связи АКО, IS-A). Подпишите каждую дугу.

2 Создайте в Excel таблицу, описывающую все связи в формате «Узел1 → тип связи → Узел2». Проверьте, что все связи, указанные в графике, присутствуют в таблице.

3 Используя граф, ответьте на вопрос: «Какие факторы непосредственно влияют на решение банка?». Покажите путь от этих факторов к узлу «Решение банка».

Контрольные вопросы

- 1 Что такое семантическая сеть?
- 2 Можно ли по построенной сети автоматически получить список всех параметров заёмщика? Если да, то как?
- 3 Почему семантические сети удобны для проверки полноты знаний?
- 4 Какие ограничения имеет данная модель по сравнению с продукционными правилами?

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

Программное обеспечение: Enterprise Architect для создания ER-диаграмм, MS Access / SQL Server для реализации БД.

Цель работы: представить знания в виде иерархии фреймов-прототипов; спроектировать реляционную базу данных, отражающую фреймовую структуру; реализовать запросы, демонстрирующие наследование свойств и принятие решений.

Теоретические сведения.

Фрейм – структура данных с именем и набором слотов. Фреймы могут наследовать слоты от родительских фреймов. При реляционной реализации каждому фрейму соответствует таблица, а связь между фреймами задаётся внешними ключами. Это позволяет хранить как прототипы (правила), так и экземпляры (конкретные заявки).

Пример 1

Фреймы и таблицы.

1 Заёмщик BorrowerFrame (Id, Name, Income, WorkExperienceMonths, LatePayments, HasGuarantor, CollateralValue).

2 Кредит CreditFrame (Id, BorrowerId (FK), LoanAmount, InterestRate, LoanTermMonths).

3 Финансовые расчёты CreditAnalysisFrame (Id, CreditId (FK), MonthlyPayment, DebtLoad, CollateralCoefficient, CreditCoefficient).

4 Решение банка BankDecisionFrame (Id, CreditId (FK), CreditCoefficient, LatePayments, WorkExperienceMonths, HasGuarantor, DecisionType).

Пример 2

В примере разработана система фреймов, описывающая основные сущности авторазборки: автомобиль, получаемые из него запчасти и металл, а также участники процесса – клиенты и юридические лица. Каждый фрейм представлен как таблица со слотами.

Фрейм «Человек» является родительским для фреймов «Клиент» и «Юр. лицо», что реализует иерархию наследования: общие свойства (умения, ресурсы) хранятся в родительской таблице, а специфические – в дочерних, связанных внешним ключом.

Фрейм «Авто» содержит ключевые характеристики автомобиля (цена приёма, вес). Связь с ним из фреймов «Запчасть» и «Металл» через внешний ключ FK_Авто отражает отношение «часть–целое»: детали и металлолом относятся к конкретному автомобилю.

Присоединённые процедуры в описании фреймов указаны декларативно (РК, FK). В реляционной реализации они заменяются ограничениями первичных и внешних ключей, а при необходимости – триггерами или хранимыми процедурами.

На рисунках 5.1 и 5.3 приведены примеры фреймовых структур.

На рисунке 5.2 приведен пример ER-диаграммы базы данных со связями М : N.

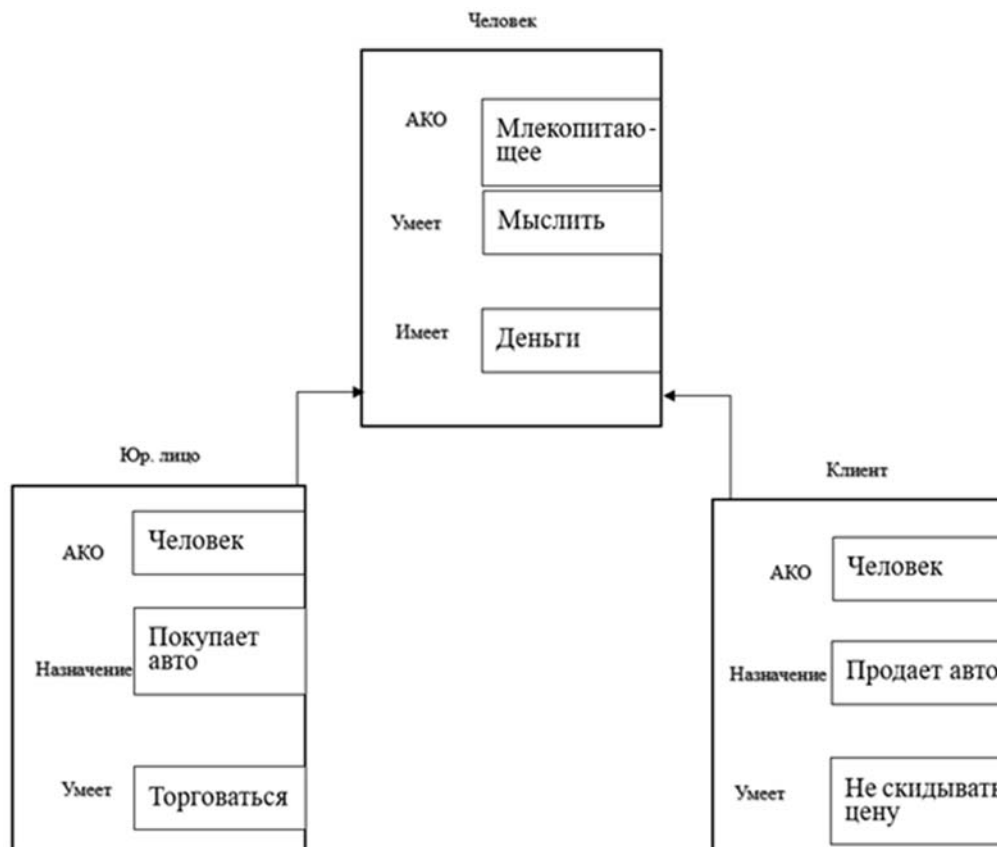


Рисунок 5.1 – Пример фреймовой структуры «Человек – Юр. лицо – Клиент»

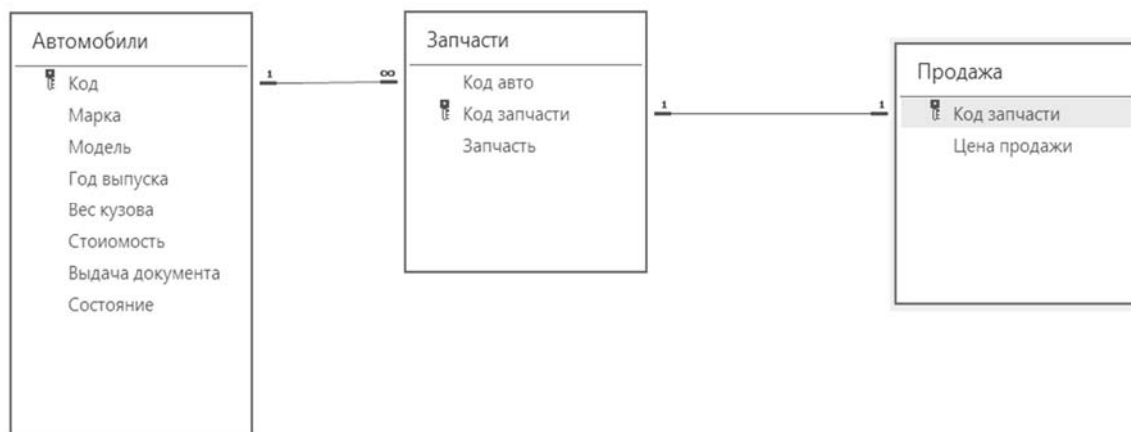


Рисунок 5.2 – Пример ER-диаграммы базы данных со связями М : N

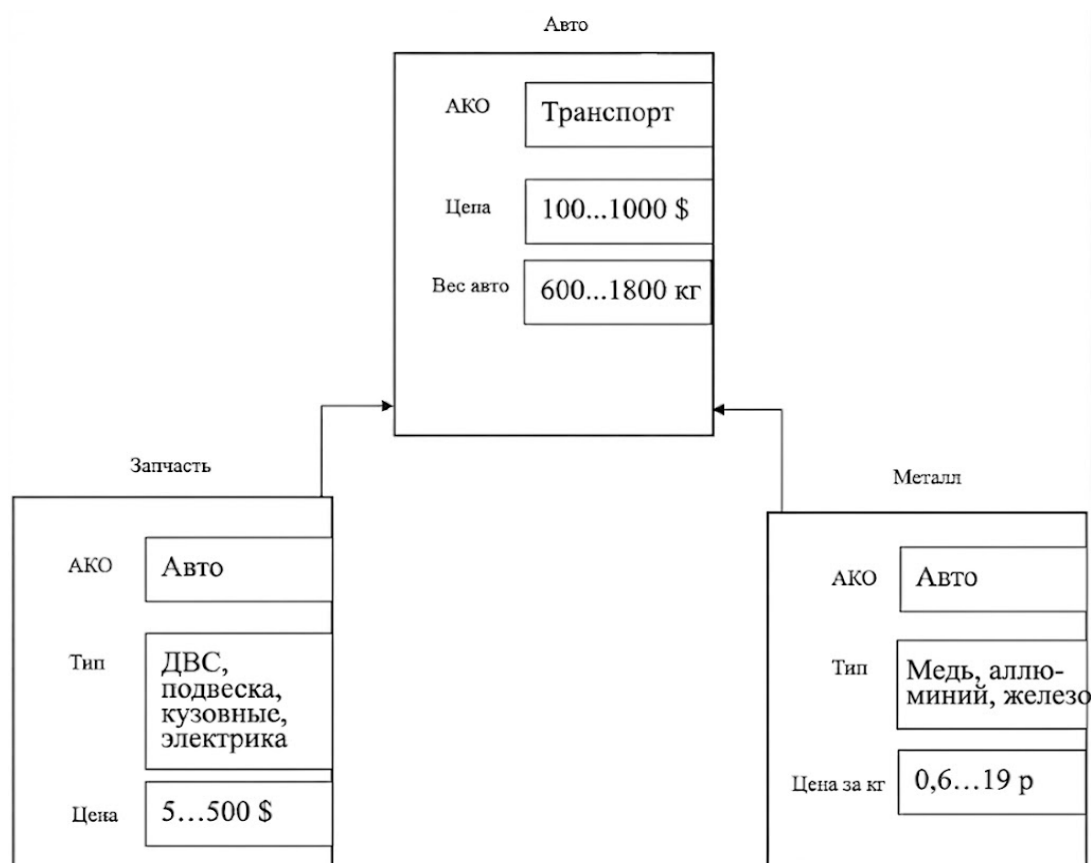


Рисунок 5.3 – Пример фреймовой структуры «Авто – Запчасть – Металл»

Примеры структур таблиц базы знаний исследуемой предметной области представлены в таблицах 5.1–5.6. На рисунке 5.4 приведен пример реализации интерфейса интеллектуальной информационной системы.

Таблица 5.1 – Структура таблицы «Запчасть»

Имя слота	Тип слота	Значение слота	Присоединенная процедура
Тип	Текстовый (50)	Тип запчастей	FK Авто
Цена	Денежный	Цена запчастей	FK Авто

Таблица 5.2 – Структура таблицы «Металл»

Имя слота	Тип слота	Значение слота	Присоединенная процедура
Тип	Текстовый (20)	Тип металла	FK Авто
Цена за кг	Денежный	Цена металла за 1 кг	FK Авто

Таблица 5.3 – Структура таблицы «Авто»

Имя слота	Тип слота	Значение слота	Присоединенная процедура
Цена	Денежный	Цена автомобиля	PK Авто
Вес авто	Числовой	Вес автомобиля	PK Авто

Таблица 5.4 – Структура таблицы «Юр. лицо»

Имя слота	Тип слота	Значение слота	Присоединенная процедура
Назначение	Текстовый (100)	Предназначение юридического лица	FK Человек
Умеет	Текстовый (100)	Умение юридического лица	FK Человек

Таблица 5.5 – Структура таблицы «Клиент»

Имя слота	Тип слота	Значение слота	Присоединенная процедура
Назначение	Текстовый (100)	Предназначение клиента	FK Человек
Умеет	Текстовый (100)	Умение клиента	FK Человек

Таблица 5.6 – Структура таблицы «Человек»

Имя слота	Тип слота	Значение слота	Присоединенная процедура
Умеет	Текстовый (70)	Умение человека	РК Человек
Имеет	Текстовый (70)	Имеющиеся ресурсы человека	РК Человек

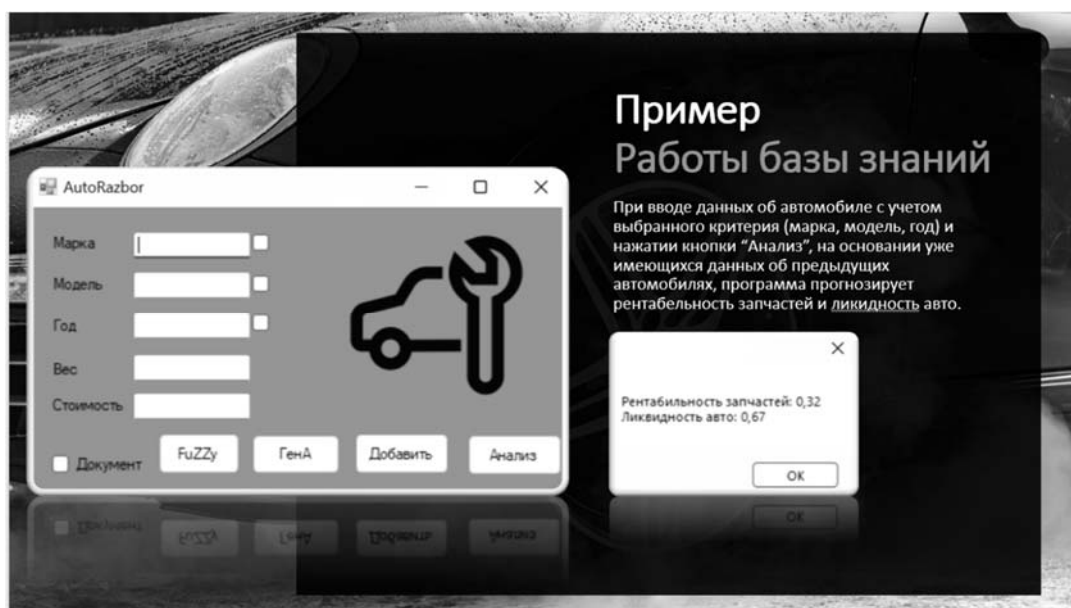


Рисунок 5.4 – Пример реализации интерфейса интеллектуальной информационной системы

Практические задания.

1 Создайте в MS Access (или SQL Server) базу данных CreditKnowledgeBase. Выполните скрипты для создания не менее трех-четырех таблиц в соответствии с фреймами.

2 Заполните таблицы тестовыми данными не менее чем для 10 заёмщиков с разными параметрами. В таблицу решений BankDecisionFrame внесите соответствующие решения (можно использовать правила из ЛР3).

3 Напишите запрос, который для заданного BorrowerId возвращает все его параметры, расчётные показатели и решение банка. Проверьте его работу.

4 Дополнительно: создайте форму в Access для ввода нового заёмщика и автоматического расчёта показателей с помощью VBA.

Контрольные вопросы

1 Что такое фреймы и для чего они используются?

2 Как в реляционной БД реализуется наследование слотов между фреймами?

3 Каким образом можно модифицировать схему, чтобы хранить не только конкретные решения, но и правила вывода?

4 В чём преимущество использования SQL-запросов при работе с фреймовой базой знаний?

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

Программное обеспечение: Microsoft Word, MS Visio / draw.io для построения байесовской сети доверия, разработанная ИИС (MS Access + VBA) для вывода вероятностей.

Цель работы: освоить применение формулы Байеса в условиях неопределённости; построить простейшую байесовскую сеть доверия для оценки кредитоспособности; реализовать в ИИС форму, вычисляющую вероятность дефолта при заданных свидетельствах.

Теоретические сведения.

Байесовская сеть состоит из узлов-переменных и ориентированных дуг. Для каждого узла задана таблица условных вероятностей. Наличие просрочек (А) и высокой долговой нагрузки (В) могут влиять на вероятность дефолта (С). Зная априорные вероятности и СРТ, можно вычислить апостериорную вероятность $P(A | C, B)$ и т. д.

Пример 1 (адаптирован под кредит).

Переменные:

1) А – «Есть просрочки» (да/нет), $P(A=\text{да}) = 0,2$;

2) В – «Долговая нагрузка $> 0,5$ » (да/нет), $P(B=\text{да}) = 0,3$ (независимы).

3) С – «Дефолт» (да/нет), зависит от А и В.

Таблица СРТ представлена на рисунке 6.1.

A	B	C	$P(C A, B)$
Да	Да	Да	0,9
Да	Нет	Да	0,6
Нет	Да	Да	0,4
Нет	Нет	Да	0,05

Рисунок 6.1 – Таблица условных вероятностей

Расчёт в Word: формула Байеса для $P(A = \text{да} | C = \text{да}, B = \text{да})$.

Реализация в Access: форма frmBayes с флажками «Просрочки», «Высокая нагрузка», кнопкой «Рассчитать». На VBA вычисляется апостериорная вероятность дефолта и выводится в процентах. Все вероятности хранятся в таблице BayesCPT.

Пример 2

Приведен пример поиска книги в базе знаний по ключевому слову, либо жанру. Вершины C и D на рисунке 6.2 – не зависимые вершины, E – вершина, зависимая от C и D.

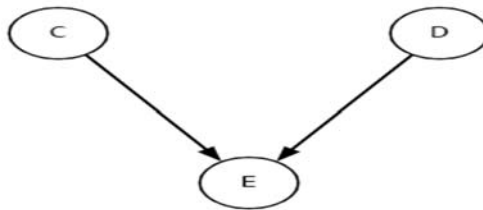


Рисунок 6.2 – Байесовская сеть доверия

C – совпадение ключевого слова; D – совпадение жанра; E – получение нужной книги.

Таблица условных вероятностей для вершин C и D – количественные представления вероятностей совпадений представлена на рисунке 6.3, таблица перекрестных совпадений для событий C и D – на рисунке 6.4, таблица условных вероятностей – на рисунке 6.5.

Совпало ключевое слово		Совпал жанр	
да	нет	да	нет
0,2	0,8	0,25	0,75

Рисунок 6.3 – Таблица условных вероятностей для вершин C и D – количественные представления вероятностей совпадений

Вероятность того, что совпало ключевое слово, если выдало нужную книгу и совпал жанр.

$$P(C | E, D) = P(E | C) \cdot P(C) / (P(E | C) \cdot P(C) + P(E | \bar{C}) \cdot P(\bar{C})) =$$

$$= 0,05 \cdot 0,2 / (0,05 \cdot 0,2 + 0,2 \cdot 0,8) = 0,06.$$

Перекрестные совпадения		Совпал жанр	
		да	нет
Совпало ключевое слово	да	0,05	0,15
	нет	0,2	0,6

Рисунок 6.4 – Таблица перекрестных совпадений для событий C и D

Условные вероятности		Совпал жанр			
		да		нет	
		Совпало ключевое слово		Совпало ключевое слово	
		да	нет	да	нет
Выдало нужную книгу	да	0,05	0,2	0,15	0,6
	нет	0,95	0,8	0,85	0,4
Σ		1	1	1	1

Рисунок 6.5 – Таблица условных вероятностей

Практические задания.

1 В документе Word (с использованием MS Visio, или draw.io) постройте байесовскую сеть из трёх узлов (Просрочки, Долговая нагрузка, Дефолт). Задайте априорные и условные вероятности. Выполните ручной расчёт вероятности дефолта при наличии просрочек и высокой нагрузки.

2 В ИИС (Access), созданной в ЛР5, добавьте таблицу BayesProbabilities и форму «Вероятностная оценка». Реализуйте код VBA, который считывает введённые свидетельства и вычисляет $P(\text{Дефолт} | \text{свидетельства})$. Выведите результат на форму.

3 Добавьте ещё один узел «Наличие залога» и модифицируйте сеть. Рассчитайте вручную и в программе вероятность дефолта при трёх свидетельствах.

Контрольные вопросы

- 1 Как формулируется теорема Байеса для двух событий?
- 2 Почему байесовские сети удобны для представления неопределённых знаний?
- 3 Как влияет добавление новой вершины на сложность точного вывода?
- 4 Где в вашей БД целесообразно хранить таблицы условных вероятностей?

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

Программное обеспечение: Python (или VBA в Excel) с подключением к базе данных Access.

Цель работы: изучить принципы эволюционного поиска правил; реализовать генетический алгоритм для автоматической оптимизации порогов кредитного правила; сравнить полученное правило с экспертными правилами из ЛР3.

Теоретические сведения.

Генетический алгоритм (ГА) кодирует решение в виде хромосомы. Популяция развивается под действием отбора, скрещивания и мутации. Функция приспособленности оценивает качество правила на обучающей выборке (например, процент верных решений). В задаче кредитного скоринга ГА может подбирать оптимальные пороги для стажа, долговой нагрузки и кредитного коэффициента.

Пример (адаптирован под кредит).

Обучающая выборка: таблица TrainingCases в Access с полями: Стаж, Долговая нагрузка, Кредитный коэффициент, Просрочки, Эталонное решение (выдать/отказать).

Цель: найти правило «ЕСЛИ стаж $\geq S$ AND долговая нагрузка $\leq D$ AND кредитный коэффициент $\geq K$ AND просрочки = 0 ТО выдать».

Хромосома: три вещественных числа $[S, D, K]$.

Приспособленность: доля правильно классифицированных примеров из обучающей таблицы.

Операторы: турнирный отбор, арифметический кроссовер, гауссова мутация.

После выполнения ГА (50 поколений) лучшая особь даёт пороги, близкие к экспертным, что подтверждает качество извлечённых знаний.

Реализация возможна на Python с библиотекой pyodbc для чтения Access, либо на VBA в Excel, где популяция и расчёты ведутся на листе.

Практические задания.

1 Подготовьте в Access таблицу с 50–100 учебными примерами (сгенерируйте на основе экспертных правил с небольшим шумом). Экспортируйте данные в CSV.

2 На Python реализуйте ГА для оптимизации трёх числовых порогов. Постройте график роста максимальной приспособленности по поколениям. Выведите найденные пороги.

3 Сравните точность полученного правила с исходным экспертным на тестовой выборке. Опишите, удалось ли ГА улучшить правило или оно совпало с

экспертным.

Контрольные вопросы

- 1 В чём отличие генетического алгоритма от метода полного перебора при поиске правил?
- 2 Почему для задач извлечения знаний важно использовать отдельную тестовую выборку?
- 3 Какие способы кодирования правил, помимо вещественных порогов, вы можете предложить?
- 4 Как можно применить ГА для отбора наиболее информативных факторов?

8 Лабораторная работа № 8. Разработка искусственного нейрона

Программное обеспечение: Python, NumPy, Microsoft Excel (для визуализации).

Цель работы: изучить математическую модель искусственного нейрона; реализовать нейрон с пороговой и сигмоидальной функцией активации; вручную подобрать веса и смещение для задачи классификации состояния станка по двум диагностическим параметрам.

Теоретические сведения.

Искусственный нейрон выполняет операцию:

$$y = \varphi(w_1 \cdot x_1 + w_2 \cdot x_2 + \dots + w_n \cdot x_n + b),$$

где φ – функция активации.

Пороговая функция выдаёт 1, если $net \geq 0$, иначе 0; сигмоида: $\sigma(z) = 1/(1+e^{-z})$.

Один нейрон способен решать задачи линейной классификации. В данной работе нейрон должен разделить состояния «Норма» (продолжать работу) и «Требуется вмешательство» по двум параметрам: температура подшипника и виброскорость.

Пример

Создать нейрон, который выдаёт сигнал «Требуется остановка» (1), если температура $t > 75$ °С И виброскорость $v > 3,5$ мм/с.

Подбираем веса и смещение для сигмоиды так, чтобы при $t = 80$, $v = 4$ получался выход $> 0,8$, а при $t = 60$, $v = 2$ – выход $< 0,2$.

Код на Python:

```
python
import numpy as np

class Neuron:
    def __init__(self, w, b):
        self.w = np.array(w)
        self.b = b
    def sigmoid(self, z):
        return 1 / (1 + np.exp(-z))
    def predict(self, x):
        net = np.dot(self.w, x) + self.b
        return self.sigmoid(net)

# Нейрон-детектор опасного состояния
neuron = Neuron([0.1, 0.3], -10) #  $t \times 0.1 + v \times 0.3 - 10$ 
# Проверка
print(neuron.predict([80, 4])) # ~0.86
print(neuron.predict([60, 2])) # ~0.047
```

Выход нейрона интерпретируется как вероятность необходимости вмешательства.

Практические задания.

1 Создайте в среде Python класс нейрона с сигмоидальной и пороговой функциями активации. Вручную подберите веса и смещение так, чтобы нейрон правильно классифицировал пять заранее заданных комбинаций (t, v) из таблицы состояний станка.

2 Реализуйте на листе Excel модель того же нейрона (без VBA), где в отдельных ячейках задаются веса, смещение и входные параметры, а выход вычисляется по формуле сигмоиды. Проверьте совпадение результатов с Python.

3 Исследуйте, как влияет значение смещения b на положение разделяющей прямой в координатах (t, v) . Постройте график прямой для нескольких значений b .

Контрольные вопросы

- 1 Какие компоненты входят в модель искусственного нейрона?
- 2 Почему пороговая функция не применяется при градиентном обучении?
- 3 Как геометрически интерпретировать веса нейрона?
- 4 Сколько нейронов потребуется, чтобы разделить три класса состояний?

9 Лабораторная работа № 9. Нейронные сети и их свойства

Программное обеспечение: Python, Scikit-learn, Matplotlib; данные подготовлены в Excel/CSV.

Цель работы: изучить многослойный персептрон (MLP); освоить обучение нейросети с помощью библиотеки Scikit-learn; исследовать влияние архитектуры и гиперпараметров на точность классификации состояния станка.

Теоретические сведения.

Многослойный персептрон состоит из входного, одного или нескольких скрытых слоёв и выходного. Благодаря нелинейным функциям активации MLP способен аппроксимировать любую непрерывную функцию. Основные гиперпараметры: число скрытых слоёв и нейронов в них, функция активации (relu, tanh, logistic), параметр регуляризации α . Переобучение диагностируется по большому разрыву между точностью на обучении и на тесте.

Пример

Обучим MLP на наборе данных «Диагностика шпинделя» с признаками: температура (t), виброскорость (v), наработка после ТО, биение инструмента, шум; целевая переменная – тип ремонта (0 – нет, 1 – внеочередное ТО, 2 – замена подшипника). Используем MLPClassifier.

```
python
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.neural_network import MLPClassifier
from sklearn.preprocessing import StandardScaler

data = pd.read_csv('spindle_data.csv')
X = data[['temp', 'vibro', 'hours', 'runout', 'noise']]
y = data['action']

scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)
X_train, X_test, y_train, y_test = train_test_split(X_scaled, y, test_size=0.2)

mlp = MLPClassifier(hidden_layer_sizes=(6,), activation='relu', max_iter=500)
mlp.fit(X_train, y_train)
print('Train accuracy:', mlp.score(X_train, y_train))
print('Test accuracy:', mlp.score(X_test, y_test))
Строим график зависимости точности от числа нейронов в скрытом слое.
```

Практические задания.

1 Сгенерируйте или загрузите таблицу с 200 примерами состояния станка (см. пример). Обучите MLP с одним скрытым слоем (5–8 нейронов). Оцените

точность на тестовой выборке.

2 Проведите эксперимент: изменяйте количество нейронов в скрытом слое от 2 до 20, фиксируя точность обучения и теста. Постройте график, объясните, где начинается переобучение.

3 Зафиксируйте архитектуру и сравните функции активации 'relu', 'tanh', 'logistic'. Сравните скорость сходимости (число итераций) и итоговую точность. Сделайте вывод о предпочтительном выборе для данной задачи.

Контрольные вопросы

1 Какие преимущества даёт добавление скрытых слоёв по сравнению с одиночным нейроном?

2 Как обнаружить переобучение в процессе обучения MLP?

3 Почему функция ReLU сегодня используется чаще, чем сигмоида?

4 Для чего нужна нормализация входных данных перед обучением?

10 Лабораторная работа № 10. Разработка алгоритма обучения нейронных сетей

Программное обеспечение: Python, NumPy (без высокоуровневых библиотек для нейросетей).

Цель работы: освоить алгоритм обратного распространения ошибки; реализовать его самостоятельно для полносвязной сети с одним скрытым слоем; обучить сеть на задаче классификации состояний станка.

Теоретические сведения.

Обратное распространение вычисляет градиенты функции потерь по всем весам сети, последовательно передавая ошибку от выходного слоя к входному. Для сигмоиды производная $\sigma'(z) = \sigma(z)(1-\sigma(z))$. Обновление весов $w := w - \eta \cdot \nabla w$, где η – скорость обучения.

Пример

Создадим класс MLP с одним скрытым слоем (три входа, четыре нейрона в скрытом, два выхода). Обучим на тех же данных, что в ЛР9, с использованием MSE.

```
python
class SimpleMLP:
    def __init__(self, sizes, lr=0.01):
        self.lr = lr
        self.W1 = np.random.randn(sizes[0], sizes[1]) * 0.1
        self.b1 = np.zeros(sizes[1])
        self.W2 = np.random.randn(sizes[1], sizes[2]) * 0.1
        self.b2 = np.zeros(sizes[2])
```

```

def sigmoid(self, z):
    return 1 / (1 + np.exp(-z))
def forward(self, X):
    self.z1 = X @ self.W1 + self.b1
    self.a1 = self.sigmoid(self.z1)
    self.z2 = self.a1 @ self.W2 + self.b2
    self.a2 = self.sigmoid(self.z2) # для бинарной классификации
    return self.a2
def backward(self, X, y, out):
    m = X.shape[0]
    d2 = (out - y) * (out * (1 - out)) / m
    dW2 = self.a1.T @ d2
    db2 = np.sum(d2, axis=0)
    d1 = (d2 @ self.W2.T) * (self.a1 * (1 - self.a1))
    dW1 = X.T @ d1
    db1 = np.sum(d1, axis=0)
    self.W2 -= self.lr * dW2
    self.b2 -= self.lr * db2
    self.W1 -= self.lr * dW1
    self.b1 -= self.lr * db1

```

Обучение 1000 эпох, вывод графика ошибки.

Практические задания.

1 Используя предоставленный набор данных с тремя признаками (t , v , $runout$) и бинарной меткой «Требуется остановка», реализуйте обучение нейросети с нуля. Постройте кривую среднеквадратичной ошибки по эпохам.

2 Модифицируйте функцию активации на ReLU и $\tanh$, реализовав их производные. Сравните скорость сходимости и итоговую точность для трёх вариантов.

3 Добавьте $L2$ -регуляризацию (штраф суммы квадратов весов) в функцию потерь. Проведите обучение с разными коэффициентами регуляризации (0; 0,001, 0,01). Прокомментируйте изменение величины весов и обобщающей способности.

Контрольные вопросы

- 1 В чём заключается идея алгоритма обратного распространения?
- 2 Какую роль играет скорость обучения?
- 3 Почему возникает проблема «исчезающего градиента» и какие функции активации её смягчают?
- 4 Зачем нужна регуляризация при обучении нейросети?

11 Лабораторная работа № 11. Искусственные нейронные сети встречного распространения

Программное обеспечение: Python, NumPy; результаты визуализируются в Matplotlib.

Цель работы: изучить архитектуру сети встречного распространения (Counterpropagation); реализовать слой Кохонена для кластеризации эксплуатационных состояний станка и слой Гроссберга для предсказания ремонтного воздействия.

Теоретические сведения.

Сеть состоит из слоя Кохонена (конкурентное обучение без учителя, самоорганизующаяся карта) и слоя Гроссберга (линейный, обучение с учителем). Сеть быстро обучается и позволяет одновременно решать задачи кластеризации и ассоциативного вывода. В нашей задаче слой Кохонена разбивает векторы показателей на кластеры (например, «норма», «допустимое отклонение», «критическое состояние»), а слой Гроссберга сопоставляет каждому кластеру рекомендуемое действие.

Пример

Обучаем карту Кохонена размером 3×3 на нормированных векторах (t , v , runout). После фиксации центров для каждого примера определяем нейрон-победитель. Затем обучаем слой Гроссберга: для каждого примера по известному действию сдвигаем веса победителя к целевому вектору действия.

Код фрагмента:

```
python
class KohonenLayer:
    def __init__(self, map_size, input_dim, lr=0.5):
        self.weights = np.random.rand(map_size[0], map_size[1], input_dim)
        self.lr = lr
    def find_winner(self, x):
        dist = np.linalg.norm(self.weights - x, axis=2)
        return np.unravel_index(np.argmin(dist), dist.shape)
    def train(self, x):
        win = self.find_winner(x)
        self.weights[win] += self.lr * (x - self.weights[win])
```

Слой Гроссберга обучается по правилу $\Delta w = \eta \cdot (y - w)$.

Практические задания.

1 Сгенерируйте 150 векторов состояния станка с тремя признаками и метками действий (0 – продолжать, 1 – внеочередное ТО, 2 – останов). Реализуйте слой Кохонена с 9 нейронами (3×3). Обучите карту и визуализируйте её, раскрасив нейроны по доминирующему классу.

2 Добавьте слой Гроссберга, который обучается предсказывать номер действия по активированному нейрону Кохонена. Оцените точность предсказания на тестовой выборке.

3 Экспериментально подберите размер карты (2×2 , 3×3 , 4×4) и сравните точность работы слоя Гроссберга. Объясните, почему слишком большой размер ухудшает обобщение.

Контрольные вопросы

1 В чём отличие обучения слоя Кохонена от слоя Гроссберга?

2 Почему сеть встречного распространения обучается быстрее многослойного персептрона?

3 Что такое «топографическое упорядочивание» в карте Кохонена?

4 Можно ли с помощью CPN решать задачи, где выходов больше одного?

12 Лабораторная работа № 12. Искусственные нейронные сети радиальных базисных функций

Программное обеспечение: Python, NumPy, Matplotlib; для кластеризации – Scikit-learn (k-means).

Цель работы: освоить RBF-сеть; понять принцип интерполяции радиальными базисными функциями; применить RBF-сеть для восстановления функциональной зависимости вибрации от температуры и наработки.

Теоретические сведения.

RBF-сеть имеет скрытый слой, каждый нейрон которого вычисляет гауссовскую функцию $\phi_i(x) = \exp(-\|x - c_i\|^2 / (2\sigma^2))$. Выходной слой – линейная комбинация $y = \sum w_i \phi_i(x) + b$. Центры c_i часто определяются алгоритмом k-means, σ подбирается эвристически. Веса находятся методом наименьших квадратов (псевдообратная матрица). Сеть хороша для аппроксимации гладких функций.

Пример

По историческим данным (температура, наработка) предсказать ожидаемую вибрацию. Разместим шесть центров методом k-means.

```
python
from sklearn.cluster import KMeans
from scipy.linalg import pinv

# X – входные векторы, y – целевая вибрация
kmeans = KMeans(n_clusters=6, random_state=0).fit(X)
centers = kmeans.cluster_centers_
sigma = np.mean([np.linalg.norm(c1 - c2) for c1 in centers for c2 in centers]) / np.sqrt(6)
```

```
def rbf(x, c, s):
    return np.exp(-np.linalg.norm(x - c)**2 / (2*s**2))

Phi = np.array([[rbf(x, c, sigma) for c in centers] for x in X])
Phi = np.hstack((Phi, np.ones((X.shape[0],1)))) # bias
w = pinv(Phi) @ y
```

Строим поверхность предсказанной вибрации в координатах «температура – наработка».

Практические задания.

1 Сгенерируйте 100 точек со случайными значениями температуры и наработки, вычислите вибрацию по формуле с шумом. Реализуйте RBF-сеть с шестью центрами, выбранными случайно из обучающих примеров. Вычислите веса через псевдоинверсию. Постройте график предсказанной и фактической вибрации.

2 Используйте k-means для определения центров и эмпирическую формулу для σ (среднее расстояние между центрами, делённое на $\sqrt{2}$ ·число_центров). Сравните качество аппроксимации (MSE) со случайным выбором центров.

3 Исследуйте влияние количества центров (2, 4, 6, 8, 10) на точность. Для каждого числа вычислите ошибку на обучающей и тестовой выборках. Постройте график, определите оптимальное число центров.

Контрольные вопросы

1 Чем RBF-сеть принципиально отличается от многослойного персептрона?

2 Как выбрать ширину гауссиана σ ? Что будет при слишком малом и слишком большом σ ?

3 Почему для нахождения весов выходного слоя RBF-сети можно использовать псевдообратную матрицу?

4 В каких практических задачах RBF-сети предпочтительнее MLP?

13 Лабораторная работа № 13. Искусственные нейронные сети с анализом главных компонент

Программное обеспечение: Python, Scikit-learn, Matplotlib; данные из Access/CSV.

Цель работы: изучить метод главных компонент (PCA) как инструмент сокращения размерности входных данных; обучить нейронную сеть на исходных и сжатых признаках; оценить влияние PCA на точность и скорость обучения в задаче диагностики станка.

Теоретические сведения.

РСА преобразует исходные коррелированные признаки в ортогональные главные компоненты, упорядоченные по убыванию дисперсии. Снижение размерности уменьшает шум и мультиколлинеарность, ускоряет обучение, но может привести к потере информации. В задачах диагностики часто используется что для визуализации многомерных данных и предобработки перед нейросетью.

Пример

Исходный набор имеет 10 признаков (температуры в нескольких точках, вибрации по осям, токи двигателей). Применим РСА, оставим три компонента, объясняющие 95 % дисперсии. Затем обучим MLP на полном и сокращённом наборах, сравним точность и время обучения.

```
python
from sklearn.decomposition import PCA
from sklearn.neural_network import MLPClassifier
```

```
pca = PCA(n_components=0.95)
X_reduced = pca.fit_transform(X_scaled)
```

```
mlp_full = MLPClassifier(hidden_layer_sizes=(10,)).fit(X_train_full, y_train)
mlp_pca = MLPClassifier(hidden_layer_sizes=(10,)).fit(X_train_pca, y_train)
```

Выводим число компонент, график кумулятивной дисперсии, метрики.

Практические задания.

1 Загрузите многомерный набор данных (минимум шесть признаков) о состоянии станка. Проведите РСА, постройте график доли объяснённой дисперсии. Определите минимальное число компонент, сохраняющих не менее 90 % дисперсии.

2 Обучите два классификатора MLP одинаковой архитектуры – на исходных данных и на главных компонентах. Сравните точность на тесте и время обучения. Объясните результат.

3 Визуализируйте первые две главные компоненты на плоскости, раскрасив точки по типу технического воздействия. Проанализируйте, насколько хорошо классы разделяются в новом пространстве.

Контрольные вопросы

- 1 Какую проблему многомерных данных решает РСА?
- 2 Почему перед применением РСА данные необходимо стандартизировать?
- 3 Всегда ли сокращение размерности улучшает точность нейросети?
- 4 Можно ли использовать РСА для отбора наиболее информативных исходных признаков?

14 Лабораторная работа № 14. Каскадные искусственные нейронные сети

Программное обеспечение: Python, NumPy; возможно использование готовых библиотек для алгоритма Cascade Correlation.

Цель работы: изучить принцип каскадного наращивания архитектуры; реализовать упрощённый алгоритм Cascade Correlation; применить его для построения классификатора состояний станка без ручного подбора топологии.

Теоретические сведения.

Каскадная корреляционная сеть (Cascade Correlation) начинает обучение с пустого скрытого слоя, добавляя нейроны по одному. Каждый новый нейрон (кандидат) тренируется максимизировать корреляцию своего выхода с остаточной ошибкой сети. После обучения веса кандидата замораживаются, и он добавляется в сеть. Процесс продолжается до достижения требуемой точности или предела числа нейронов. Это автоматизирует выбор архитектуры.

Пример

Реализуем упрощённый вариант: на каждом шаге добавляем один нейрон с сигмоидой, обучаем его веса градиентным спуском, максимизируя абсолютную корреляцию с ошибкой. После фиксации весов пересчитываем выходной слой через псевдоинверсию. Демонстрируем на задаче бинарной классификации «требуется ли останов».

```
python
def candidate_correlation(x, errors, weights, lr=0.01, epochs=200):
    w = np.random.randn(x.shape[1]) * 0.1
    b = 0.0
    for _ in range(epochs):
        net = x @ w + b
        out = 1 / (1 + np.exp(-net))
        corr = np.corrcoef(out, errors)[0,1]
        # градиент корреляции по out, затем по w,b (упрощённо)
        # ... (реализация в коде студента)
    return w, b
```

После добавления нескольких нейронов сеть достигает высокой точности.

Практические задания.

1 Используя данные с тремя признаками и бинарной целью, реализуйте алгоритм добавления одного скрытого нейрона, обучая его максимизировать корреляцию с ошибкой текущей сети. Выведите график зависимости ошибки от числа добавленных нейронов.

2 Модифицируйте алгоритм так, чтобы на каждом шаге генерировался пул из четырех кандидатов с различной инициализацией, и выбирался лучший по

корреляции. Сравните скорость уменьшения ошибки с вариантом одиночного кандидата.

3 Примените каскадную сеть к многоклассовой задаче (три типа ремонта), преобразовав её в несколько бинарных классификаторов (one-vs-rest). Оцените итоговое число нейронов и точность, сравните с MLP из ЛР9.

Контрольные вопросы

1 В чём отличие каскадной сети от обычного MLP с фиксированной архитектурой?

2 Как вычисляется корреляция, которую максимизирует кандидат?

3 Какой недостаток связан с замораживанием весов добавленных нейронов?

4 В каких ситуациях каскадные сети могут быть предпочтительнее глубокого обучения?

Список литературы

1 **Гаврилова, Т. А.** Базы знаний интеллектуальных систем : учеб. пособие / Т. А. Гаврилова, В. Ф. Хорошевский. – СПб. : Питер, 2000. – 384 с.

2 **Джексон, П.** Введение в экспертные системы / П. Джексон. – М. : Вильямс, 2001. – 624 с.

3 **Рассел, С.** Искусственный интеллект: современный подход / С. Рассел, П. Норвиг. – 2-е изд. – М. : Вильямс, 2006. – 1408 с.

4 **Хайкин, С.** Нейронные сети. Полный курс / С. Хайкин. – 2-е изд. – М. : Вильямс, 2006. – 1104 с.

5 **Осовский, С.** Нейронные сети для обработки информации / С. Осовский. – М. : Финансы и статистика, 2002. – 344 с.

6 **Ярушкина, Н. Г.** Интеллектуальные информационные системы : учеб. пособие / Н. Г. Ярушкина. – Ульяновск : УлГТУ, 2010. – 153 с.

7 **Люгер, Дж. Ф.** Искусственный интеллект: стратегии и методы решения сложных проблем / Дж. Ф. Люгер. – М. : Вильямс, 2003. – 864 с.

8 **Уотермен, Д.** Руководство по экспертным системам / Д. Уотермен. – М. : Мир, 1989. – 388 с.

9 **Дюк, В. А.** Интеллектуальный анализ данных : учеб. пособие / В. А. Дюк, А. А. Самойленко. – СПб. : Питер, 2001. – 368 с.

10 **Рутковская, Д.** Нейронные сети, генетические алгоритмы и нечеткие системы / Д. Рутковская, М. Пилиньский, Л. Рутковский. – М. : Горячая линия – Телеком, 2004. – 452 с.

11 **Берк, Э.** Современные методы интеллектуального анализа данных : учеб. пособие / Э. Берк, Р. И. Кузьмич, С. В. Мельников. – Мн.: БГУИР, 2008. – 168 с.

12 **Рассел, С.** Искусственный интеллект: современный подход / С. Рассел, П. Норвиг. – 4-е изд. – М. : Вильямс, 2021. – 1408 с.

13 **Шолле, Ф.** Глубокое обучение на Python / Ф. Шолле. – 2-е изд. – СПб. : Питер, 2022. – 576 с.

14 **Рашка, С.** Python и машинное обучение: машинное и глубокое обучение с использованием Python, scikit-learn и TensorFlow / С. Рашка, В. Мирджалили. – 3-е изд. – М. : ДМК-Пресс, 2021. – 848 с.

15 **Вьюгин, В. В.** Машинное обучение : учеб. пособие / В. В. Вьюгин. – М. : Физматлит, 2022. – 384 с.

16 **Флах, П.** Машинное обучение. Наука и искусство построения алгоритмов, которые извлекают знания из данных / П. Флах. – М. : ДМК-Пресс, 2021. – 400 с.

17 **Келлехер, Дж. Д.** Наука о данных: базовый курс / Дж. Д. Келлехер, Б. Тирни. – М. : Альпина Паблишер, 2023. – 256 с.

18 **Саттон, Р. С.** Обучение с подкреплением: введение / Р. С. Саттон, Э. Г. Барто. – 2-е изд. – М. : ДМК-Пресс, 2021. – 456 с.

19 Scikit-learn: Machine Learning in Python. – URL: <https://scikit-learn.org> (date of access: 20.03.2026).

20 Python 3 documentation. – URL: <https://docs.python.org> (date of access: 20.03.2026).

21 Microsoft Access SQL reference. – URL: <https://docs.microsoft.com/en-us/office/client-developer/access/desktop-database-reference/> (date of access: 20.03.2026).