

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«КУБАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»
Факультет компьютерных технологий и прикладной математики

УТВЕРЖДАЮ:

Проректор по учебной работе,
качеству образования – первый
проректор

Хагуров Т.А.

« 29 » августа 2025 г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ
ФТД.02 «Нечёткие нейросетевые модели»

Направление подготовки 02.03.02 Фундаментальная информатика и
информационные технологии

Направленность (профиль) Современные методы машинного обучения и
компьютерного зрения

Форма обучения очная

Квалификация бакалавр

Краснодар 2025

Рабочая программа дисциплины «Нечёткие нейросетевые модели» составлена в соответствии с федеральным государственным образовательным стандартом высшего образования (ФГОС ВО) по направлению подготовки 02.03.02 Фундаментальная информатика и информационные технологии.

Программу составил(и):

Подколзин В.В. канд. физ.-мат. наук, доцент

Рабочая программа дисциплины утверждена на заседании кафедры информационных технологий протокол № 1 от «26» августа 2025г.

Заведующий кафедрой Подколзин В.В.

Утверждена на заседании учебно-методической комиссии факультета компьютерных технологий и прикладной математики протокол № 01 от «28» августа 2025 г.

Председатель УМК факультета

А. В. Коваленко

подпись

Рецензенты:

Мостовой Евгений Викторович, генеральный директор ООО «Портал-Юг», e-mail: mostovoy@portal-yug.ru

Луценко Евгений Вениаминович, доктор экономических наук, кандидат технических наук, профессор кафедры компьютерных технологий и систем Федерального государственного бюджетное образовательное учреждение высшего образования «Кубанский государственный аграрный университет имени И.Т. Трубилина», e-mail: prof.lutsenko@gmail.com

1 Цели и задачи изучения дисциплины (модуля)

1.1 Цель освоения дисциплины

Целью преподавания и изучения дисциплины «Нейросетевые и нечеткие модели» является изучение принципов построения систем нечеткого вывода, определение круга задач, решаемых с помощью построения нейросетевых нечетких моделей, получение практических навыков написания программных продуктов с применением нейронных и нечетких моделей с применением современных платформ и фреймворков.

1.2 Задачи дисциплины

В результате освоения данной компетенции студент должен:

знать принципы работы и типы архитектур нечетких нейронных сетей, структуру систем нечеткого вывода, математические модели обучения нечетких нейронных сетей.

уметь реализовывать модули анализа данных на основе нейросетевых нечетких моделей, внедрять их в комплексные программные решения.

владеть навыками определения архитектуры нечеткой нейронной сети, подходящей для решения конкретной задачи, навыками написания модулей работы с внешними системами (размеченные файлы, базы данных, потоки ввода) и применения нейросетевой модели для анализа данных, полученных из внешних систем.

1.3 Место дисциплины (модуля) в структуре образовательной программы

Дисциплина «Нейросетевые и нечеткие модели» относится к дисциплинам факультативам.

Дисциплина в значительной степени **взаимодействует для формирования компетенций** с дисциплинами:

- Системы искусственного интеллекта;
- Нейросетевые технологии;

1.4 Профессиональные роли в структуре образовательной программы

Роль 1: **Data Engineer (Инженер по данным)**

Задачи:

- Проектирование и построение ETL-процессов
- Создание и оптимизация хранилищ данных
- Обеспечение качества и доступности данных
- Настройка инфраструктуры для обработки больших данных
- Интеграция разрозненных источников данных
- Работа с данными в области природопользования, медицины, связи и телекоммуникаций

Роль 2: **ML Engineer (Инженер МО)**

Задачи:

- Реализация ML-моделей в продуктивных системах
- Оптимизация производительности и масштабирование моделей
- Разработка ML-пайплайнов и автоматизация процессов
- Мониторинг качества моделей в продуктиве
- Интеграция ML-решений с бизнес-приложениями

Роль 3: **MLOps (Специалист по эксплуатации ИИ)**

Задачи:

- Автоматизация процессов обучения и развертывания моделей
- Мониторинг производительности ML-систем
- Управление версиями моделей и данных
- Обеспечение CI/CD для ML-проектов
- Оптимизация вычислительных ресурсов

1.5 Перечень планируемых результатов обучения по дисциплине (модулю), соотнесенных с планируемыми результатами освоения образовательной программы
Изучение данной учебной дисциплины направлено на формирование у обучающихся следующих компетенций: ПК-4.1; ПК-4.2; ПК-4.3

ПК-4	<i>Способен планировать необходимые ресурсы и этапы выполнения работ в области информационно-коммуникационных технологий, составлять соответствующие технические описания и инструкции</i>
ПК-4.1	Использует современные инструментальные средства разработки баз данных, прикладного программного обеспечения и систем различного функционального назначения
	<p>Знать: Знать классификацию, принципы работы и критерии выбора современных инструментов разработки (IDE, СУБД, фреймворки, CI/CD, контейнеризации) для разных типов ПО.</p> <p>Уметь: Уметь выбирать, применять и настраивать инструменты для конкретных задач разработки, тестирования, развертывания и сопровождения ПО, а также интегрировать их в рабочий процесс.</p> <p>Владеть: Владеть практическими навыками работы с инструментальными средствами выбранного технологического стека и способностью быстро осваивать новые инструменты в соответствии с требованиями проекта.</p>
ПК-4.2	Применяет современные приемы работы с инструментальными средствами, поддерживающими создание программных продуктов и программных комплексов на базе языков программирования, баз данных и пакетов прикладных программ
	<p>Знать: Знать современные практики и приёмы профессиональной работы с ключевыми инструментами (отладка, профилирование, рефакторинг, модульное и интеграционное тестирование, сборка и развёртывание), соответствующие выбранному языку программирования, базам данных и пакетам прикладных программ.</p> <p>Уметь: Уметь применять эти приёмы для повышения эффективности и качества разработки: использовать интерактивные отладчики и профилировщики, писать автоматизированные тесты, выполнять рефакторинг кода с помощью IDE, настраивать системы непрерывной интеграции и сборки проектов.</p> <p>Владеть: Владеть комплексным навыком использования инструментальной среды как единого технологического процесса — от написания и оптимизации кода до автоматизированного тестирования, сборки и подготовки промышленного релиза программного продукта.</p>

ПК-4.3	Способен использовать методы эффективного управления командой при разработке, внедрении и сопровождении программных продуктов
	<p>Знать: Знать ключевые методологии, фреймворки и практики управления командой (Agile, Scrum, Kanban), а также принципы командной динамики, мотивации и эффективной коммуникации на разных этапах жизненного цикла ПО.</p> <p>Уметь: Уметь применять эти методы на практике: формировать бэклог, планировать спринты, проводить митинги, управлять рисками и разрешать конфликты для достижения целей проекта по разработке, внедрению и сопровождению ПО.</p> <p>Владеть: Владеть навыками фасилитации командной работы, использования инструментов управления проектами (Jira, Trello) и создания продуктивной рабочей среды, направленной на постоянное улучшение процессов и результатов командной деятельности.</p>

2. Структура и содержание дисциплины

2.1 Распределение трудоёмкости дисциплины по видам работ

Общая трудоёмкость дисциплины составляет 2 зач. ед. (72 часа), их распределение по видам работ представлено в таблице

Вид учебной работы	Всего часов	Семестры (часы)					
		7					
Контактная работа, в том числе:	52,2	52,2					
Аудиторные занятия (всего):	50	50					
Занятия лекционного типа	16	16					
Лабораторные занятия	34	34					
Занятия семинарского типа (семинары, практические занятия)							
Иная контактная работа:	2,2	2,2					
Контроль самостоятельной работы (КСР)	2	2					
Промежуточная аттестация (ИКР)	0,2	0,2					
Самостоятельная работа, в том числе:	19,8	19,8					
Курсовая работа							
Выполнение индивидуальных заданий	16	16					
Реферат	3,8	3,8					
Подготовка к текущему контролю							
Контроль:							
Общая трудоемкость	час.	72	72				
	в том числе контактная работа	52,2	52,2				
	зач. ед	2	2				

2.2 Структура дисциплины

Распределение видов учебной работы и их трудоемкости по разделам дисциплины. Разделы (темы) дисциплины, изучаемые в 7 семестре

№	Наименование разделов (тем)	Количество часов				
		Всего	Аудиторная работа			Внеаудиторная работа
			Л	ПЗ	ЛР	СРС
1	2	3	4	5	6	7
1.	Системы нечеткого вывода	24	6		18	6
2.	Принятие решений. Метод анализа иерархий	24	6		6	6
3.	Нечеткие нейросетевые технологии	18	4		10	7,8
ИТОГО по разделам дисциплины		69,8	16		34	19,8
Контроль самостоятельной работы (КСР)		2				
Промежуточная аттестация (ИКР)		0,2				
Подготовка к текущему контролю						
Общая трудоемкость по дисциплине		72				

Примечание: Л – лекции, ПЗ – практические занятия/семинары, ЛР – лабораторные занятия, СРС – самостоятельная работа студента

2.3 Содержание разделов (тем) дисциплины

2.3.1 Занятия лекционного типа

№	Наименование раздела (темы)	Содержание раздела (темы)	Форма текущего контроля
1	2	3	4
1.	Системы нечеткого вывода	Структура системы нечеткого вывода. Понятие лингвистической переменной. Требования к функции принадлежности термов. Примеры функций принадлежности. Общая схема системы нечеткого вывода. Компоненты системы нечеткого вывода. Фазификация. Правила вывода. Импликация, агрегация и композиция. Дефазификация. Виды систем нечеткого вывода.	ЛР
2.	Принятие решений. Метод анализа иерархий	Метод анализа иерархий Саати для нескольких критериев, нескольких термов. Метод анализа иерархий Саати для нескольких экспертов, групп экспертов. Критерии непротиворечивости эксперта Критерии согласованности экспертов в методе анализа иерархий. Метод попарных сравнений.	ЛР
3.	Нечеткие нейросетевые технологии	Мотивация синтеза: "серые ящики" (нейросети) vs "прозрачные ящики" (нечеткие системы). Гибридные интеллектуальные системы: нейро-нечеткие системы (Neuro-Fuzzy Systems, NFS). Архитектура ННС 0-го типа (Нечеткий перцептрон): представление нечетких множеств в виде весов, нечеткая функция активации. Архитектура ННС 1-го типа (ANFIS - Adaptive Neuro-Fuzzy Inference System): эквивалентность сети Такаги-Сугено-Канга и многослойного перцептрона. Слои ANFIS: фазификация, правило, нормализация, вывод, агрегация. Обучение ANFIS: гибридный алгоритм (градиентный спуск + МНК). Интерпретируемость vs Точность Нечеткие кластеризаторы в НС: Использование FCM (Fuzzy C-Means) для инициализации и интерпретации. Глубокие нечеткие системы (Deep Fuzzy Networks): каскадные и иерархические ANFIS. ННС 2-го и высших типов: работа с нечеткими данными второго порядка (нечеткие числа как входы). Нечеткие	ЛР

№	Наименование раздела (темы)	Содержание раздела (темы)	Форма текущего контроля
1	2	3	4
		автокодировщики (Fuzzy Autoencoders): обучение в условиях неопределенности. Применение: Нечеткие контроллеры для LLM (калибровка уверенности), обработка зашумленных данных в Computer Vision, системы поддержки принятия решений с интерпретируемым выводом. Современные тенденции: интеграция с Transformers (Fuzzy Attention), нечеткие байесовские методы	

Примечание: ЛР – отчет/защита лабораторной работы, КП - выполнение курсового проекта, КР - курсовой работы, РГЗ - расчетно-графического задания, Р - написание реферата, Э - эссе, К - коллоквиум, Т – тестирование, РЗ – решение задач.

2.3.2 Занятия семинарского типа

Не предусмотрены

2.3.3 Лабораторные занятия

№	Наименование раздела (темы)	Наименование лабораторных работ	Форма текущего контроля
1	2	3	4
1.	Системы нечеткого вывода	Нечеткие множества – 4 часа	ЛР1
2.	Системы нечеткого вывода	Нечеткая логика - 4 часа	ЛР2
3.	Системы нечеткого вывода	Построение программного комплекса с применением нечеткой модели 4 часа	ЛР3
4.	Системы нечеткого вывода	Построение нечеткой экспертной системы Мамдани, Такаги-Сугено. 4 часа	ЛР4
5.	Принятие решений. Метод анализа иерархий	Постановка задачи, выбор предметной области, методы попарных сравнений, метод анализа иерархий. 6 часов	ЛР5
6.	Нечеткие нейросетевые технологии	Реализация и обучение системы ANFIS для задачи регрессии 4 часа	ЛР6
7.	Нечеткие нейросетевые технологии	Построение и применение нечеткого кластеризатора для инициализации нейросети 6 часов	ЛР7

Примечание: ЛР – отчет/защита лабораторной работы, КП - выполнение курсового проекта, КР - курсовой работы, РГЗ - расчетно-графического задания, Р - написание реферата, Э - эссе, К - коллоквиум, Т – тестирование, РЗ – решение задач.

2.3.4 Примерная тематика курсовых работ (проектов)

Курсовая работа не предусмотрена.

2.4 Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине (модулю)

№	Вид СРС	Перечень учебно-методического обеспечения дисциплины по выполнению самостоятельной работы
1	2	3
1	Изучение теоретического материала	Методические указания по организации самостоятельной работы студентов, утвержденные кафедрой вычислительных технологий, протокол №7 от 07.05.2025

2	Решение задач	Методические указания по организации самостоятельной работы студентов, утвержденные кафедрой вычислительных технологий, протокол №7 от 07.05.2025
---	---------------	---

Учебно-методические материалы для самостоятельной работы обучающихся из числа инвалидов и лиц с ограниченными возможностями здоровья (ОВЗ) предоставляются в формах, адаптированных к ограничениям их здоровья и восприятия информации:

Для лиц с нарушениями зрения:

- в печатной форме увеличенным шрифтом,
- в форме электронного документа,
- в форме аудиофайла,
- в печатной форме на языке Брайля.

Для лиц с нарушениями слуха:

- в печатной форме,
- в форме электронного документа.

Для лиц с нарушениями опорно-двигательного аппарата:

- в печатной форме,
- в форме электронного документа,
- в форме аудиофайла.

Данный перечень может быть конкретизирован в зависимости от контингента обучающихся.

3. Образовательные технологии

В соответствии с требованиями ФГОС в программа дисциплины предусматривает использование в учебном процессе следующих образовательные технологии: чтение лекций с использованием мультимедийных технологий; метод малых групп, разбор практических задач и кейсов.

При обучении используются следующие образовательные технологии:

- Технология коммуникативного обучения – направлена на формирование коммуникативной компетентности студентов, которая является базовой, необходимой для адаптации к современным условиям межкультурной коммуникации.

- Технология разноуровневого (дифференцированного) обучения – предполагает осуществление познавательной деятельности студентов с учётом их индивидуальных способностей, возможностей и интересов, поощряя их реализовывать свой творческий потенциал. Создание и использование диагностических тестов является неотъемлемой частью данной технологии.

- Технология модульного обучения – предусматривает деление содержания дисциплины на достаточно автономные разделы (модули), интегрированные в общий курс.

- Информационно-коммуникационные технологии (ИКТ) - расширяют рамки образовательного процесса, повышая его практическую направленность, способствуют интенсификации самостоятельной работы учащихся и повышению познавательной активности. В рамках ИКТ выделяются 2 вида технологий:

- Технология использования компьютерных программ – позволяет эффективно дополнить процесс обучения языку на всех уровнях.

- Интернет-технологии – предоставляют широкие возможности для поиска информации, разработки научных проектов, ведения научных исследований.

- Технология индивидуализации обучения – помогает реализовывать личностно-ориентированный подход, учитывая индивидуальные особенности и потребности учащихся.

- Проектная технология – ориентирована на моделирование социального взаимодействия учащихся с целью решения задачи, которая определяется в рамках профессиональной подготовки, выделяя ту или иную предметную область.

- Технология обучения в сотрудничестве – реализует идею взаимного обучения, осуществляя как индивидуальную, так и коллективную ответственность за решение учебных задач.

- Игровая технология – позволяет развивать навыки рассмотрения ряда возможных способов решения проблем, активизируя мышление студентов и раскрывая личностный потенциал каждого учащегося.

- Технология развития критического мышления – способствует формированию разносторонней личности, способной критически относиться к информации, умению отбирать информацию для решения поставленной задачи.

Комплексное использование в учебном процессе всех вышеназванных технологий стимулируют личностную, интеллектуальную активность, развивают познавательные процессы, способствуют формированию компетенций, которыми должен обладать будущий специалист.

Основные виды интерактивных образовательных технологий включают в себя:

- работа в малых группах (команде) - совместная деятельность студентов в группе под руководством лидера, направленная на решение общей задачи путём творческого сложения результатов индивидуальной работы членов команды с делением полномочий и ответственности;

- проектная технология - индивидуальная или коллективная деятельность по отбору, распределению и систематизации материала по определенной теме, в результате которой составляется проект;

- анализ конкретных ситуаций - анализ реальных проблемных ситуаций, имевших место в соответствующей области профессиональной деятельности, и поиск вариантов лучших решений;

- развитие критического мышления – образовательная деятельность, направленная на развитие у студентов разумного, рефлексивного мышления, способного выдвинуть новые идеи и увидеть новые возможности.

Подход разбора конкретных задач и ситуаций широко используется как преподавателем, так и студентами во время лекций, лабораторных занятий и анализа результатов самостоятельной работы. Это обусловлено тем, что при исследовании и решении каждой конкретной задачи имеется, как правило, несколько методов, а это требует разбора и оценки целой совокупности конкретных ситуаций.

Примечание: Л – лекции, ПЗ – практические занятия/семинары, ЛР – лабораторные занятия, СРС – самостоятельная работа студента

Темы, задания и вопросы для самостоятельной работы призваны сформировать навыки поиска информации, умения самостоятельно расширять и углублять знания, полученные в ходе лекционных и практических занятий.

Подход разбора конкретных ситуаций широко используется как преподавателем, так и студентами при проведении анализа результатов самостоятельной работы.

Для лиц с ограниченными возможностями здоровья предусмотрена организация консультаций с использованием электронной почты.

Для лиц с нарушениями зрения:

- в печатной форме увеличенным шрифтом,
- в форме электронного документа.

Для лиц с нарушениями слуха:

- в печатной форме,
- в форме электронного документа.

Для лиц с нарушениями опорно-двигательного аппарата:

- в печатной форме,
- в форме электронного документа.

Для лиц с ограниченными возможностями здоровья предусмотрена организация консультаций с использованием электронной почты.

Данный перечень может быть конкретизирован в зависимости от контингента обучающихся.

4. Оценочные и методические материалы

4.1 Оценочные средства для текущего контроля успеваемости и промежуточной аттестации

Оценочные средства предназначены для контроля и оценки образовательных достижений обучающихся, освоивших программу учебной дисциплины «название дисциплины».

Оценочные средства включает контрольные материалы для проведения **текущего контроля** в форме оценки лабораторных работ к проекта к **зачету**.

Оценочные средства для инвалидов и лиц с ограниченными возможностями здоровья выбираются с учетом их индивидуальных психофизических особенностей.

– при необходимости инвалидам и лицам с ограниченными возможностями здоровья предоставляется дополнительное время для подготовки ответа на экзамене;

– при проведении процедуры оценивания результатов обучения инвалидов и лиц с ограниченными возможностями здоровья предусматривается использование технических средств, необходимых им в связи с их индивидуальными особенностями;

– при необходимости для обучающихся с ограниченными возможностями здоровья и инвалидов процедура оценивания результатов обучения по дисциплине может проводиться в несколько этапов.

Процедура оценивания результатов обучения инвалидов и лиц с ограниченными возможностями здоровья по дисциплине (модулю) предусматривает предоставление информации в формах, адаптированных к ограничениям их здоровья и восприятия информации:

Для лиц с нарушениями зрения:

- в печатной форме увеличенным шрифтом,
- в форме электронного документа.

Для лиц с нарушениями слуха:

- в печатной форме,
- в форме электронного документа.

Для лиц с нарушениями опорно-двигательного аппарата:

- в печатной форме,
- в форме электронного документа.

Данный перечень может быть конкретизирован в зависимости от контингента обучающихся.

Структура оценочных средств для текущей и промежуточной аттестации

№ п/ п	Контролируемые разделы (темы) дисциплины*	Код контролируемой компетенции (или ее части)	Наименование оценочного средства	
			Текущий контроль	Промежуточная аттестация

1	Системы нечеткого вывода	ПК-4.1; ПК-4.2; ПК-4.3	ЛР 1-4	Вопросы на зачет 1-10
2	Принятие решений. Метод анализа иерархий	ПК-4.1; ПК-4.2; ПК-4.3	ЛР 5	Вопросы на зачет 11-15
3	Нечеткие нейросетевые технологии	ПК-4.1; ПК-4.2; ПК-4.3	ЛР 6-7	Вопросы на зачет 15-22

Показатели, критерии и шкала оценки сформированных компетенций

Соответствие **базовому уровню** освоения компетенций планируемым результатам обучения и критериям их оценивания (оценка: **зачтено**):

ПК-4	<i>Способен планировать необходимые ресурсы и этапы выполнения работ в области информационно-коммуникационных технологий, составлять соответствующие технические описания и инструкции</i>
ПК-4.1	Использует современные инструментальные средства разработки баз данных, прикладного программного обеспечения и систем различного функционального назначения Знает программные средства для моделирования нечетких множеств и создания нейронных сетей, инструментальные интегрированные программные среды разработчиков для применения моделей нечетких множеств и нейронных сетей Знает основные определения теории нечетких множеств и нечеткой логики, нейронных сетей, их области применения, технологию создания и использования нейронных сетей, принципы моделирования нечеткой логики и нейронных сетей для решения задач управления Владеет навыками построения и обучения искусственных нейронных сетей; навыками формирования систем нечеткого логического вывода, различных стратегий вывода знаний и объяснения полученных результатов.
ПК-4.2	Применяет современные приемы работы с инструментальными средствами, поддерживающими создание программных продуктов и программных комплексов на базе языков программирования, баз данных и пакетов прикладных программ Умеет пользоваться аппаратными средствами создания нейронных сетей и нечетких логических практических задач Владеет навыками построения и обучения искусственных нейронных сетей; навыками формирования систем нечеткого логического вывода
ПК-4.3	Способен использовать методы эффективного управления командой при разработке, внедрении и сопровождении программных продуктов Умеет разрабатывать программные средства с применением моделей нечеткой логики, нейронных сетей

Типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций в процессе освоения образовательной программы

Факультет компьютерных технологий и прикладной математики
Кафедра вычислительных технологий

01.03.02

Нечеткие нейросетевые модели

Лабораторная работа № 6.

Реализация и обучение системы ANFIS для задачи регрессии

Цель: Реализовать архитектуру ANFIS "с нуля" с использованием NumPy для управления низкоуровневыми операциями и обучить ее на синтетических данных гибридным алгоритмом.

Задания:

1. Генерация синтетических данных.

- Сгенерируйте датасет для задачи регрессии, используя функцию $y = \sin(2x) / e^{(x/3)} + \varepsilon$, где x равномерно распределен на интервале $[0, 10]$, а ε - гауссовский шум с $\mu=0$, $\sigma=0.1$.
- Разбейте данные на обучающую и тестовую выборки.

2. Реализация слоев ANFIS.

- Реализуйте слой фаззификации. Используйте гауссовы функции принадлежности (Gaussian MF). Количество термов на каждую входную переменную задайте параметром ($n_mfs=3$). Функция должна возвращать степень принадлежности для каждого терма.
- Реализуйте слой правил (Rule Layer). Для двух входных переменных x_0 и x_1 и $n_mfs=3$ должно быть сгенерировано 9 правил. Каждое правило вычисляет уровень активации с помощью операции T-нормы (произведение).
- Реализуйте слой нормализации (Normalization Layer), который вычисляет нормализованный уровень активации для каждого правила.
- Реализуйте слой следствий (Consequent Layer). Для каждого правила i реализуйте функцию следствия первого порядка: $f_i = p_i * x_0 + q_i * x_1 + r_i$. Параметры p_i , q_i , r_i должны быть инициализированы случайным образом.
- Реализуйте слой агрегации (Defuzzification Layer), который вычисляет итоговый выход ANFIS как взвешенную сумму: $output = \sum (normalized_firing_strength_i * f_i)$.

3. Реализация гибридного алгоритма обучения.

- Реализуйте прямой проход (forward pass) по всей сети.
- Реализуйте обратный проход (backward pass) с использованием гибридного алгоритма:
- **Для параметров следствия (p , q , r):** Используйте метод наименьших квадратов (МНК) в рамках одного эпохи. Рассчитайте оптимальные параметры для фиксированных параметров предпосылки.
- **Для параметров предпосылки (c , σ гауссовых MFs):** Используйте градиентный спуск (стохастический или mini-batch). Реализуйте вычисление градиентов функции потерь (MSE) по параметрам c и σ через цепное правило.

4. Обучение и валидация.

- Обучите модель на протяжении 500 эпох. Визуализируйте график сходимости (loss по эпохам).
- Постройте на одном графике: исходные данные, предсказания обученной модели и аппроксимацию "идеальной" функции без шума.
- Проведите эксперимент: исследуйте, как изменение количества функций принадлежности (n_mfs от 2 до 5) влияет на точность модели и ее склонность к переобучению. Результаты сведите в таблицу.

Теоретические вопросы для защиты ЛР №1:

1. Объясните, почему архитектура ANFIS является функционально эквивалентной сети Сугено-Тсака-Канга.
2. Опишите, какие именно параметры ANFIS обновляются с помощью МНК, а какие — с помощью градиентного спуска в гибридном алгоритме. В чем преимущество такого подхода?
3. Какую роль играет слой нормализации в ANFIS? Что произойдет с моделью, если этот слой убрать?
4. Как интерпретировать полученные после обучения функции принадлежности? О чем говорит их форма и взаимное расположение?
5. В чем заключается компромисс между интерпретируемостью и точностью при увеличении количества термов в ANFIS?

Лабораторная работа №7: Построение и применение нечеткого кластеризатора для инициализации нейросети

Цель: Использовать алгоритм нечеткой кластеризации (Fuzzy C-Means) для предобработки данных и инициализации простой нейронной сети, сравнив эффективность с случайной инициализацией.

Задания:

1. **Реализация Fuzzy C-Means (FCM).**
 - Реализуйте алгоритм FCM "с нуля" для датасета Iris (используйте только два признака для визуализации). Алгоритм должен принимать на вход данные, количество кластеров C и параметр нечеткости m (обычно $1.5 < m < 2.5$).
 - Реализуйте расчет матрицы нечеткого разбиения U и обновление центров кластеров.
 - Критерием остановки является изменение центроидов меньше заданного порога ϵ или достижение максимального числа итераций.
 - Визуализируйте результат кластеризации для $C=3$. Отобразите центроиды и мягкую принадлежность точек к кластерам (например, с помощью прозрачности или цвета).
2. **Создание нейросети с инициализацией от FCM.**
 - Создайте простую полносвязную нейронную сеть (например, с одним скрытым слоем из 10 нейронов с \tanh и выходным слоем с softmax для 3 классов) для решения задачи классификации на датасете Iris.
 - **Метод инициализации:** Используйте центроиды, полученные FCM (после кластеризации по всем 4 признакам), для инициализации весов между входным и скрытым слоем. Веса от скрытого слоя к выходному инициализируйте случайно.
 - **Подсказка:** Рассмотрите центроиды как "прототипы" в пространстве признаков. Веса можно инициализировать, например, как расстояния от входных данных до этих центроидов или с помощью метода, основанного на RBF-сетях.
3. **Сравнительный эксперимент.**
 - Обучите две идентичные модели: одну с предложенной FCM-инициализацией, вторую — со стандартной случайной инициализацией (например, Xavier/Glorot).
 - Обучайте обе модели на одном и том же разбиении данных (train/test) с одинаковыми параметрами оптимизатора (например, SGD или Adam) и learning rate.
 - Постройте на одном графике кривые обучения (train/validation accuracy) для обеих моделей.
 - Зафиксируйте итоговую точность на тестовой выборке и количество эпох, необходимое для сходимости (достижения 95% от максимальной accuracy). Сделайте вывод об эффективности метода.

Теоретические вопросы для защиты ЛР №2:

1. Чем принципиально отличается результат работы FCM от алгоритма K-Means? Какое преимущество дает "мягкая" кластеризация для последующей инициализации нейросети?
2. Опишите предложенный вами метод использования центроидов FCM для инициализации весов нейронной сети. Каков его интуитивный смысл?
3. Проанализируйте результаты эксперимента. В каких сценариях FCM-инициализация может давать значительное преимущество, а в каких — нет?
4. Как параметр нечеткости m в FCM влияет на результат кластеризации и, как следствие, на инициализацию сети? Что произойдет при $m \rightarrow 1$ и при $m \rightarrow \infty$?
5. Можно ли данный подход отнести к методам обучения с частичным привлечением учителя (semi-supervised learning)? Обоснуйте свой ответ

Студенты должны разместить код, отчет (README.md с описанием, графиками и выводами) и результаты в личных GitHub репозиториях. Ссылки предоставляются для проверки. Защита работ проходит в формате ответов на теоретические вопросы и обсуждения реализованного кода

Зачетно-экзаменационные материалы для промежуточной аттестации (зачёт)

Вопросы к зачёту

Часть 1: Системы нечёткого вывода (10 вопросов)

1. Опишите общую структуру и компоненты системы нечёткого вывода (СНВ). Раскройте роль лингвистической переменной и функции принадлежности в контексте этой структуры.
2. Дайте определение лингвистической переменной. Каким требованиям должны удовлетворять функции принадлежности её термов? Приведите примеры двух различных типов функций принадлежности и обоснуйте их применение.
3. Что такое фазификация? Приведите пример процедуры фазификации для конкретного числового значения в рамках заданной лингвистической переменной.
4. Опишите этап формирования базы правил в СНВ. Что такое аппарат нечётких продукционных правил и какова его роль в системе вывода?
5. Дайте определение нечёткой импликации (операции "если-то"). Какие методы реализации нечёткой импликации (\min , prod) вам известны и в чём их различие?
6. Что такое агрегация нечётких правил и для чего она необходима? Опишите процесс агрегации выходов нескольких правил, использующих один и тот же лингвистический терм.
7. Опишите этап композиции в СНВ. Чем композиция отличается от агрегации? Какой математический аппарат используется для её проведения?
8. Что такое дефазификация и почему она является необходимым этапом в СНВ? Сравните не менее двух методов дефазификации (например, центр тяжести и медиану), указав их достоинства и недостатки.
9. Сравните системы нечёткого вывода Мамдани и Сугено-Такэги-Канга. Укажите ключевые архитектурные отличия, особенности вывода и области применения.

10. Проиллюстрируйте на условном примере полный цикл работы системы Мамдани: от фазификации входных данных до получения чёткого выходного значения через этапы импликации, агрегации и дефазификации.

Часть 2: Принятие решений. Метод анализа иерархий (5 вопросов)

11. Опишите алгоритм метода анализа иерархий (МАИ) Т. Саати для случая принятия решения по нескольким критериям и нескольким альтернативам. Какова роль шкалы относительной важности Саати?
12. Опишите процедуру метода парных сравнений для построения матрицы суждений. Как на основе этой матрицы вычисляются локальные приоритеты и осуществляется проверка на согласованность?
13. Что такое индекс согласованности (ИС) и отношение согласованности (ОС) в МАИ? Сформулируйте критерии непротиворечивости эксперта. Каковы причины возникновения несогласованности в суждениях?
14. Опишите методику обобщения мнений нескольких экспертов в МАИ. Какие методы агрегации индивидуальных суждений (AIJ) и агрегации индивидуальных приоритетов (AIP) вам известны? В чём их различие?
15. Как осуществляется проверка согласованности групповых экспертных оценок? Какие метрики и подходы используются для оценки степени консенсуса внутри группы экспертов?

Часть 3: Нечёткие нейронные сети (7 вопросов)

16. В чём заключается мотивация синтеза нечёткой логики и нейронных сетей? Опишите преимущества и недостатки гибридных нейро-нечётких систем (ННС) по сравнению с "чистыми" нейросетями и "чистыми" нечёткими системами.
17. Опишите архитектуру и принцип работы адаптивной нейро-нечёткой системы вывода (ANFIS) типа Сугено. Детализируйте назначение каждого из пяти слоев этой сети.
18. Опишите гибридный алгоритм обучения ANFIS. Какие параметры сети обновляются с помощью градиентного спуска, а какие — с помощью метода наименьших квадратов (МНК)? Объясните эффективность данного подхода.
19. Что такое "нечёткий персептрон" (ННС 0-го типа)? Чем его архитектура и принцип обучения принципиально отличаются от архитектуры ANFIS?
20. Опишите концепцию ННС 2-го типа. В каких практических задачах оправдано их применение по сравнению с ННС 1-го типа? Какой дополнительный вычислительный overhead они несут?
21. Каковы принципы построения глубоких нечётких систем (Deep Fuzzy Networks)? В чём заключаются преимущества каскадных и иерархических архитектур для задач повышенной сложности?
22. Проанализируйте современные тенденции интеграции нечётких методов и передовых архитектур ИИ. Приведите пример и обоснуйте целесообразность

использования нечёткого подхода в одной из следующих областей: калибровка уверенности LLM, обработка зашумленных данных в Computer Vision, интерпретируемое усиление трансформеров (Fuzzy Attention)

4.2 Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций

Методические рекомендации, определяющие процедуры оценивания на экзамене:

Процедура промежуточной аттестации проходит в соответствии с Положением о текущем контроле и промежуточной аттестации обучающихся ФГБОУ ВО «КубГУ».

Итоговой формой контроля сформированности компетенций у обучающихся по дисциплине является зачет. Студенты обязаны сдать зачет в соответствии с расписанием и учебным планом.

ФОС промежуточной аттестации состоит из заданий и результатов текущего контроля.

Форма проведения зачета: письменно.

Преподавателю предоставляется право задавать студентам дополнительные вопросы по всей учебной программе дисциплины.

Результат сдачи зачета заносится преподавателем в экзаменационную ведомость и зачетную книжку.

Оценивание уровня освоения дисциплины основывается на качестве выполнения студентом заданий текущего контроля и проекта.

Критерии оценки:

Зачтено – выполнено 60% лабораторных работ (на оценку зачтено) и даны полные ответы на два вопроса в билете, или выполнены все лабораторные работы и дан ответ на один вопрос в билете.

Не зачтено – выполнено менее 60% лабораторных работ или не дан ответ ни на один вопрос в билете.

Методические рекомендации, определяющие процедуры оценивания лабораторных работ:

Процедура оценивания лабораторных работ проходит в соответствии с Положением о текущем контроле и промежуточной аттестации обучающихся ФГБОУ ВО «КубГУ».

По каждой лабораторной работе оформляется отчет. Отчеты сдаются на проверку руководителю в течение курса по мере их выполнения, и защищаются студентами в установленном порядке.

При защите отчета студенту могут быть заданы вопросы и дополнительные задания по сути лабораторной работы, в том числе из списка контрольных вопросов к данной лабораторной работе. При неудовлетворительной оценке знаний студента по теме данного отчета, студент возвращается к повторному изучению соответствующих материалов, после чего допускается к повторной защите. Неудовлетворительно выполненный отчет также возвращается на доработку.

Отчет должен содержать заголовок, тему лабораторной работы, цель, задание, индивидуальную тему, описание хода выполнения работы, необходимые прикладные материалы (схемы, макеты документов и т.п.), в соответствии с требованиями к содержанию, и выводы по работе.

Оценочные средства для инвалидов и лиц с ограниченными возможностями здоровья выбираются с учетом их индивидуальных психофизических особенностей.

- при необходимости инвалидам и лицам с ограниченными возможностями здоровья предоставляется дополнительное время для подготовки ответа на экзамене;
- при проведении процедуры оценивания результатов обучения инвалидов и лиц с ограниченными возможностями здоровья предусматривается использование технических средств, необходимых им в связи с их индивидуальными особенностями;
- при необходимости для обучающихся с ограниченными возможностями здоровья и инвалидов процедура оценивания результатов обучения по дисциплине может проводиться в несколько этапов.

Процедура оценивания результатов обучения инвалидов и лиц с ограниченными возможностями здоровья по дисциплине предусматривает предоставление информации в формах, адаптированных к ограничениям их здоровья и восприятия информации:

Для лиц с нарушениями зрения:

- в печатной форме увеличенным шрифтом,
- в форме электронного документа.

Для лиц с нарушениями слуха:

- в печатной форме,
- в форме электронного документа.

Для лиц с нарушениями опорно-двигательного аппарата:

- в печатной форме,
- в форме электронного документа.

Данный перечень может быть конкретизирован в зависимости от контингента обучающихся.

4.3. Методические указания по организации вычислительной инфраструктуры

1. Цель и задачи

Цель: Сформировать у студентов практические навыки развертывания и управления современной инфраструктурой для поддержки полного жизненного цикла коллективной разработки программного обеспечения.

Задачи:

Освоить принципы инфраструктуры как кода (IaC).

Получить опыт работы с системами контейнеризации и оркестрации.

Настроить pipelines непрерывной интеграции и поставки (CI/CD).

Организовать удаленную совместную работу над кодом с использованием Git.

Обеспечить базовый мониторинг и управление конфигурациями.

2. Рекомендуемая архитектура инфраструктуры

Для каждого проектного коллектива (3-5 человек) рекомендуется развернуть следующую схему:

text

[Локальные машины разработчиков]

|
| (Git push/pull)

v

[Удаленный Git-хостинг (GitLab / GitHub)]

|
| (Webhook trigger)

v

[CI/CD Сервер (GitLab Runner / GitHub Actions)]

|
| (Docker build/push)

v

[Контейнерный реестр (Docker Hub / GitLab Registry)]

|
| (Deploy command)
v
[Продакшен-сервер / Хостинг-площадка]
|
v
[Контейнеризованное приложение (Docker Compose)]
3. Технологический стек и инструменты (рекомендуемый)
Обязательный минимум:
Система контроля версий: Git
Удаленный хостинг Git-репозитория: GitHub, GitLab или Bitbucket.
Предпочтение — GitLab, так как он предоставляет встроенный CI/CD, Issue Tracker и Container Registry в одном продукте.
Контейнеризация: Docker
Оркестрация контейнеров (для проектов повышенной сложности): Docker Compose
Инфраструктура как код (IaC):
Базовый уровень: docker-compose.yml

Продвинутый уровень: Ansible playbooks для деплоя или Terraform для управления облачной инфраструктурой (если есть доступ к облачным кредитам).
CI/CD:
GitLab CI/CD (.gitlab-ci.yml) или GitHub Actions (.github/workflows/).
Pipeline должен включать стадии: build, test, deploy.
Опционально (для углубленного изучения):
Веб-сервер / Reverse Proxy: Nginx (для раздачи статики и проксирования запросов к приложению).
Мониторинг: Prometheus + Grafana (для сбора и визуализации метрик), или готовые облачные решения (Datadog, New Relic — по бесплатным тарифам).
Базы данных: Управляемая облачная БД (AWS RDS, DigitalOcean Managed Database) либо контейнеризованная версия (PostgreSQL/MySQL в Docker).

4. Практические задания и этапы внедрения

Этап 1: Базовая настройка репозитория и CI/CD

Создать группу/организацию на выбранном Git-хостинге.
Создать проект, настроить доступ для всех членов команды.
Настроить .gitignore, README.md, LICENSE.
Создать базовый Dockerfile для приложения.
Написать простейший .gitlab-ci.yml (или аналог для GitHub Actions), который выполняет сборку Docker-образа и запуск unit-тестов.

Этап 2: Контейнеризация и оркестрация

Определить все сервисы приложения (frontend, backend, БД, кэш).
Создать docker-compose.yml для локальной разработки и для продакшена.
Настроить переменные окружения для управления конфигурацией.

Этап 3: Настройка автоматического деплоя

Зарегистрировать/настроить сервер для деплоя (рекомендуется VPS от DigitalOcean, VK Cloud, Selectel и др. с ежемесячной оплатой).

Настроить в CI/CD pipeline автоматический деплой на сервер при пуше в ветку main/master.

Способ: Использование Docker Compose на удаленном сервере через ssh.
Альтернативный способ: Развертывание через Ansible-плейбук, запускаемый из CI/CD.

Этап 4: Мониторинг и документация

Настроить логирование приложения.

(Опционально) Развернуть стек мониторинга (Prometheus+Grafana) для сбора базовых метрик (CPU, RAM, нагрузка сети).

Задokumentировать всю инфраструктуру: схему, инструкции по развертыванию, описание pipeline.

5. Критерии оценки инфраструктурной составляющей проекта

Инфраструктура является неотъемлемой частью проектной работы и оценивается в рамках общего критерия по проекту.

На «Зачтено» необходимо:

Работоспособность: Приложение должно быть доступно по URL (или на проверочном сервере) после автоматического деплоя.

Контейнеризация: Приложение и все его зависимости должны быть упакованы в Docker-контейнеры.

CI/CD: Настроен автоматический pipeline, который проходит как минимум стадии build и test. В Git-репозитории видна история запусков pipeline.

Документация: В README.md присутствуют четкие инструкции по запуску проекта локально и описание процесса деплоя.

Работа в команде: Вклад в инфраструктурную часть распределен между членами команды (видно по истории коммитов).

4.4. Методические указания по организации лабораторных работ

1. Цель и задачи

Цель: Сформировать у студентов устойчивые практические навыки работы в распределенной команде разработчиков с использованием современного инструментария и методологий.

Задачи:

Освоить жизненный цикл коллективной разработки от планирования до поставки ПО.

Сформировать компетенции работы с системами контроля версий.

Приобрести опыт настройки процессов непрерывной интеграции и поставки.

Отработать процедуры код-ревью и совместной разработки.

Закрепить навыки контейнеризации приложений.

2. Формат проведения работ

Периодичность: 6 лабораторных работ в течение семестра

Формат: Выполнение в командах по 3-4 человека

Сдача: Защита каждой работы с демонстрацией результата

Ведение: Общий репозиторий команды с историей выполнения всех ЛР

3. Структура и содержание лабораторных работ

Лабораторная работа 1: «Организация рабочего процесса команды»

Цель: Настроить инфраструктуру для коллективной работы и освоить базовые workflow Git.

Задачи:

Создать организацию/группу на GitHub/GitLab
Настроить SSH-ключи для всех участников
Создать проект с README, .gitignore, LICENSE

Отработать Git Flow: fork → clone → feature branch → commit → push → Pull Request

Настроить Issues и Projects для трекинга задач

Результат: Репозиторий с настроенным доступом для всех членов команды, минимум 3 merged PR от разных участников.

Лабораторная работа 2: «Проектирование и начало разработки»

Цель: Спроектировать архитектуру приложения и начать разработку по методологии Scrum.

Задачи:

Провести планирование спринта (2 недели)

Разработать техническое задание и архитектурную схему

Создать milestone и распределить задачи между участниками

Реализовать базовую структуру приложения

Провести код-ревью всех изменений

Результат: ТЗ в wiki, заполненный backlog, работающая заготовка приложения.

Лабораторная работа 3: «Настройка CI/CD пайплайна»

Цель: Настроить автоматическую сборку и тестирование проекта.

Задачи:

Написать Dockerfile для приложения

Настроить CI-пайплайн для автоматического запуска тестов

Добавить статический анализ кода (linters)

Настроить сборку Docker-образов

Добавить бейджи статуса сборки в README

Результат: Рабочий CI-пайплайн, проходящий все стадии, Docker-образ в registry.

Лабораторная работа 4: «Внедрение практик код-ревью»

Цель: Отработать процессы проверки кода и управления качеством.

Задачи:

Настроить protected branches

Создать checklist для код-ревью

Реализовать feature с обязательным ревью 2 участников

Настроить template для Pull Request

Внедрить требование успешной сборки для мержа PR

Результат: Процесс код-ревью документально оформлен и отработан на практике.

Лабораторная работа 5: «Настройка окружений и деплоя»

Цель: Настроить многоступенчатый деплой приложения.

Задачи:

Создать docker-compose.yml для разработки и продакшена

Настроить CD-пайплайн для деплоя на staging-сервер

Реализовать деплой по тегам на production-сервер

Настроить миграции БД при деплое

Добавить health-check для контейнеров

Результат: Приложение автоматически деплоится на два окружения.

Лабораторная работа 6: «Мониторинг и документация»

Цель: Настроить мониторинг работы приложения и завершить документацию.

Задачи:

Настроить логирование в контейнерах

Добавить сбор метрик приложения

Настроить дашборд для мониторинга

Задokumentировать API (Swagger/OpenAPI)

Составить инструкцию по развертыванию и troubleshooting

Результат: Полная документация проекта, работающая система мониторинга.

4. Критерии оценки лабораторных работ

Общие требования для защиты:

Соблюдение сроков сдачи

Участие всех членов команды

Соответствие заданиям ЛР

Качество выполнения и оформления

Критерии оценки каждой ЛР:

Критерий	Вес	Описание
Техническая реализация	40%	Корректность выполнения, работоспособность решения
Работа в команде	30%	Вклад каждого участника, история коммитов, код-ревью
Документация	20%	Описание решения, скриншоты, инструкции
Качество кода	10%	Соблюдение code style, чистота репозитория

Шкала оценки:

«Зачтено»: 70-100% от максимального балла

«Не зачтено»: менее 70% от максимального балла

5. Методические рекомендации

Для студентов:

Распределяйте роли в команде (тимлид, разработчик, тестировщик, DevOps)

Проводите ежедневные стендапы (10-15 минут)

Ведите подробную документацию всех процессов

Своевременно обращайтесь за консультациями

6. Требования к отчетности

Для каждой ЛР команда предоставляет:

Ссылку на репозиторий с выполненной работой

Пулл-реквесты с выполненными задачами

Скриншоты работающего функционала

Краткий отчет в wiki репозитория:

Цель работы

Выполненные задачи

Распределение работы между участниками

Проблемы и пути их решения

Выводы

7. Рекомендуемый инструментарий

Хостинг кода: GitHub Education, GitLab
CI/CD: GitHub Actions, GitLab CI
Контейнеризация: Docker, Docker Compose
Коммуникация: Telegram, Discord, Mattermost
Документация: Wiki GitHub/GitLab, Markdown

4.5. Методические указания по организации проектной деятельности студентов

1. Цель и задачи

Цель: Интегрировать полученные в ходе лабораторных работ знания и навыки в рамках реализации сквозного проекта, максимально приближенного к реальным условиям промышленной разработки.

Задачи:

Сформировать опыт полного жизненного цикла разработки ПО в команде

Закрепить навыки проектного планирования, управления задачами и командной коммуникации

Развить умение принимать архитектурные и технологические решения

Отработать процедуры поставки и сопровождения программного продукта

2. Организация проектной деятельности

Продолжительность: 1 семестр (параллельно с лабораторными работами)

Формат: Сквозной проект с поэтапной сдачей результатов

Команды: 3-4 человека с распределением ролей

Процесс: Гибкая методология разработки (Scrum/Kanban)

Результат: Работающее приложение с полной инфраструктурой

3. Этапы выполнения проекта

Этап 1: Инициация и планирование (Недели 1-3)

Результат: Утвержденное техническое задание и план проекта

Содержание:

Выбор темы проекта из утвержденного перечня или предложение собственной

Формирование команды, распределение ролей (Team Lead, Developer, DevOps, QA)

Разработка технического задания с описанием функциональных требований

Создание дорожной карты (Roadmap) проекта

Планирование первого спринта, создание бэклога продукта

Настройка проектной инфраструктуры (репозиторий, Issue Tracker, CI/CD)

Этап 2: Проектирование архитектуры (Недели 4-5)

Результат: Архитектурная документация и прототип

Содержание:

Выбор технологического стека

Проектирование архитектуры системы (микросервисы/монолит)

Проектирование схемы базы данных

Разработка API-спецификации (OpenAPI/Swagger)

Создание прототипа основного функционала

Презентация архитектурного решения

Этап 3: Активная разработка (Недели 6-12)

Результат: Работающая система с основным функционалом

Содержание:

Проведение регулярных спринтов (2-3 недели каждый)

Ежедневные стендапы, планирование спринтов, ретроспективы

Реализация функциональности согласно бэклогу
Настройка и поддержка CI/CD пайплайна
Написание unit- и integration-тестов
Регулярное код-ревью и мерж Pull Requests
Деплой на тестовые окружения

Этап 4: Тестирование и стабилизация (Недели 13-14)

Результат: Стабильная версия продукта

Содержание:

Интеграционное и системное тестирование

Исправление критических ошибок

Нагрузочное тестирование (опционально)

Подготовка production-окружения

Финальный деплой и проверка безопасности

Этап 5: Защита проекта (Недели 15-16)

Результат: Презентация и демонстрация проекта

Содержание:

Подготовка финальной документации

Создание презентации проекта

Демонстрация работающего продукта

Ответы на вопросы комиссии

4. Роли в команде

Team Lead:

Координация работы команды

Ведение бэклога продукта

Проведение совещаний

Контроль сроков и качества

Developer (2 человека):

Реализация функциональности

Написание тестов

Участие в код-ревью

Рефакторинг кода

DevOps Engineer:

Настройка инфраструктуры

Поддержка CI/CD

Деплой и мониторинг

Контейнеризация приложения

5. Критерии оценки проекта

Общий вес проекта в итоговой оценке: 50%

Критерий	Вес	Показатели оценки
Техническая реализация	30%	Работоспособность, соответствие ТЗ, качество кода, архитектурные решения
Процесс разработки	25%	Следование методологии, история коммитов, код-ревью, управление задачами
Инфраструктура и DevOps	20%	CI/CD пайплайн, контейнеризация, автоматизация деплоя, мониторинг
Документация	15%	Техническое задание, архитектурная документация, руководство пользователя

Презентация и защита	10%	Качество демонстрации, ответы на вопросы, командная работа
----------------------	-----	--

Минимальные требования для защиты:
 Реализован основной заявленный функционал
 Проект запускается по инструкции и работает
 Настроен CI/CD пайплайн
 Все члены команды внесли вклад в разработку
 Предоставлена полная документация

6. Требования к проекту

Обязательные компоненты:

Система контроля версий (Git) с историей разработки
 CI/CD пайплайн с автоматическими тестами
 Контейнеризация приложения (Docker)
 Автоматический деплой на сервер
 Документация в README.md и wiki
 Issue tracking с историей задач

5. Перечень учебной литературы, информационных ресурсов и технологий

5.1. Учебная литература

1. Нечеткие модели и сети : учебное пособие / В. В. Борисов, В. В. Круглов, А. С. Федулов. - 2-е изд., стер. - Москва : Горячая линия-Телеком, 2018. - 284 с. - Режим доступа: для авториз. пользователей. - URL: <https://e.lanbook.com/book/111022> (дата обращения: 20.02.2021). - ISBN 978-5-9912-0283-1. - Текст : электронный.
2. Искусственные нейронные сети : учебник / В. С. Ростовцев. - 3-е изд., стер. - Санкт-Петербург : Лань, 2023. - 216 с. - URL: <https://e.lanbook.com/book/310184> (дата обращения: 28.02.2023.) - Режим доступа: для авториз. пользователей. - ISBN 978-5-507-46446-3. - Текст : электронный.
3. Галушкин, А. И. Нейронные сети: основы теории / А. И. Галушкин. - Москва : Горячая линия-Телеком, 2017. - 496 с. - <https://e.lanbook.com/book/111043> (дата обращения: 29.03.2022). - Режим доступа: для авториз. пользователей. - ISBN 978-5-9912-0082-0. - Текст : электронный.
4. Хайкин С. Нейронные сети. - Издательство: Вильямс ISBN: 5-8459-0890-2, 2006. - 1104с. (20 экз. в библиотеке КубГУ).
5. Нечеткие множества и нейронные сети : учебное пособие / Г. Э. Яхьяева. - 2-е изд., испр. - М. : Интернет-Университет Информационных Технологий : БИНОМ. Лаборатория знаний, 2008. - 315 с. : ил. - (Основы информационных знаний). - Библиогр.: с. 315. - ISBN 9785947748185 : 132.00. - Текст : непосредственный. (23 экз. в библиотеке КубГУ).
6. Нейронные сети: распознавание, управление, принятие решений : [пособие] / А. Б. Барский. - М. : Финансы и статистика, 2007. - 175 с. : ил. - (Прикладные информационные технологии). - Библиогр.: с. 170-173. - ISBN 9785279027576 : 86 р. - Текст : непосредственный. (20 экз. в библиотеке КубГУ).
7. Коваленко, А. В. (КубГУ). Искусственный интеллект в экономике : монография / А. В. Коваленко, Е. В. Казаковцева ; Кубанский государственный университет. - 2-е изд. - Москва : Ай Пи Ар Медиа, 2023. - 361 с. : ил. - Библиогр.: с. 318-327. - ISBN 978-5-4497-1867-9 : 1475 р. - Текст : непосредственный. (40 экз. в библиотеке КубГУ).

5.2. Периодические издания:

- 1 Базы данных компании «Ист Вью» <http://dlib.eastview.com>
- 2 Электронная библиотека GREBENNIKON.RU <https://grebennikon.ru/>

5.3. Интернет-ресурсы, в том числе современные профессиональные базы данных и информационные справочные системы

Электронно-библиотечные системы (ЭБС):

- 1 ЭБС «ЮРАЙТ» <https://urait.ru/>
- 2 ЭБС «УНИВЕРСИТЕТСКАЯ БИБЛИОТЕКА ОНЛАЙН» <http://www.biblioclub.ru/>
- 3 ЭБС «BOOK.ru» <https://www.book.ru>
- 4 ЭБС «ZNANIUM.COM» www.znanium.com
- 5 ЭБС «ЛАНЬ» <https://e.lanbook.com>

Профессиональные базы данных

- 1 Scopus <http://www.scopus.com/>
- 2 ScienceDirect <https://www.sciencedirect.com/>
- 3 Журналы издательства Wiley <https://onlinelibrary.wiley.com/>
- 4 Научная электронная библиотека (НЭБ) <http://www.elibrary.ru/>
- 5 Полнотекстовые архивы ведущих западных научных журналов на Российской платформе научных журналов НЭИКОН <http://archive.neicon.ru>
- 6 Национальная электронная библиотека (доступ к Электронной библиотеке диссертаций Российской государственной библиотеки (РГБ) <https://rusneb.ru/>
- 7 Президентская библиотека им. Б.Н. Ельцина <https://www.prlib.ru/>
- 8 База данных CSD Кембриджского центра кристаллографических данных (CCDC) <https://www.ccdc.cam.ac.uk/structures/>
- 9 Springer Journals: <https://link.springer.com/>
- 10 Springer Journals Archive: <https://link.springer.com/>
- 11 Nature Journals: <https://www.nature.com/>
- 12 Springer Nature Protocols and Methods: <https://experiments.springernature.com/sources/springer-protocols>
- 13 Springer Materials: <http://materials.springer.com/>
- 14 Nano Database: <https://nano.nature.com/>
- 15 Springer eBooks (i.e. 2020 eBook collections): <https://link.springer.com/>
- 16 "Лекториум ТВ" <http://www.lektorium.tv/>
- 17 Университетская информационная система РОССИЯ <http://uisrussia.msu.ru>

Бесплатные образовательные ресурсы

- 1 Jupyter Notebook – интерактивные вычисления
- 2 Visual Studio Code – редактор кода с поддержкой Python
- 3 Google Scholar/arXiv – доступ к научным публикациям

Ресурсы свободного доступа

- 1 КиберЛенинка <http://cyberleninka.ru/>;
- 2 Американская патентная база данных <http://www.uspto.gov/patft/>
- 3 Министерство науки и высшего образования Российской Федерации <https://www.minobrnauki.gov.ru/>;
- 4 Федеральный портал "Российское образование" <http://www.edu.ru/>;
- 5 Информационная система "Единое окно доступа к образовательным ресурсам" <http://window.edu.ru/>;
- 6 Единая коллекция цифровых образовательных ресурсов <http://school-collection.edu.ru/> .
- 7 Проект Государственного института русского языка имени А.С. Пушкина "Образование на русском" <https://pushkininstitute.ru/>;
- 8 Справочно-информационный портал "Русский язык" <http://gramota.ru/>;
- 9 Служба тематических толковых словарей <http://www.glossary.ru/>;

- 10 Словари и энциклопедии <http://dic.academic.ru/>;
- 11 Образовательный портал "Учеба" <http://www.ucheba.com/>;
- 12 Законопроект "Об образовании в Российской Федерации". Вопросы и ответы http://xn--273--84d1f.xn--plai/voprosy_i_otvety

Собственные электронные образовательные и информационные ресурсы КубГУ

- 1 Электронный каталог Научной библиотеки КубГУ
<http://megapro.kubsu.ru/MegaPro/Web>
- 2 Электронная библиотека трудов ученых КубГУ
<http://megapro.kubsu.ru/MegaPro/UserEntry?Action=ToDb&idb=6>
- 3 Среда модульного динамического обучения <http://moodle.kubsu.ru>
- 4 База учебных планов, учебно-методических комплексов, публикаций и конференций
<http://infoneeds.kubsu.ru/>
- 5 Библиотека информационных ресурсов кафедры информационных образовательных технологий <http://mschool.kubsu.ru;>
- 6 Электронный архив документов КубГУ <http://docspace.kubsu.ru/>
- 7 Электронные образовательные ресурсы кафедры информационных систем и технологий в образовании КубГУ и научно-методического журнала "ШКОЛЬНЫЕ ГОДЫ"
<http://icdau.kubsu.ru/>

5.4 Перечень информационно-коммуникационных технологий

1. Облачные платформы и сервисы
cloud.ru, YandexCloud, AWS/GCP/Azure – облачные вычисления
2. Системы управления версиями и коллаборации
Git/GitHub/GitLab – контроль версий кода и совместная разработка
4. Система управления обучением
Moodle – сдача работ

5.5 Перечень лицензионного и свободно распространяемого программного обеспечения

1. Свободное ПО (Open Source)
GitLab, GIT, MLFlow, Docker, Kubernetes, Terraform.

6. Методические указания для обучающихся по освоению дисциплины (модуля)

По курсу предусмотрено проведение лекционных занятий, на которых дается систематизированный материал по java, паттернам проектирования, архитектурным паттернам. В ходе лекций рассматриваются ключевые концепции. После каждой лекции рекомендуется выполнение практических заданий для закрепления ключевых понятий и методов.

Лабораторные занятия курса посвящены практической разработке на Java. На занятиях студенты реализуют основные элементы архитектуры ООП приложения.

При самостоятельной работе студентам необходимо изучать рекомендованную литературу в виде официальной документации к используемым открытым программным продуктам, облачным платформам.

Для студентов с ограниченными возможностями здоровья предусмотрены дополнительные индивидуальные консультации, на которых преподаватель подробно разъясняет сложные аспекты дисциплины, помогает адаптировать практические задания и обеспечивает специальные условия для освоения методов работы с системами искусственного интеллекта. Индивидуальный подход позволяет таким студентам полноценно участвовать в учебном процессе и достигать требуемых результатов обучения.

7. Материально-техническое обеспечение по дисциплине (модулю)

Виртуальные машины, кластер Managed Kubernetes и ресурсы GPU в облаке предоставляется индустриальным партнером ПАО «Сбербанк»:

№	Продукт	Параметры продукта	Кол-во	Кол-во конфигураций	Ед. изм.
1	Виртуальная машина	Виртуальная машина 10% vCPU 2 vCPU 4 RAM	1	60	Шт
		ОС Ubuntu 22.04	1		Шт
		Системный диск SSD	1		Шт
			10		Гб

Дополнительные облачные ресурсы предоставляются технологическим партнером Yandex Cloud.

№	Вид работ	Наименование учебной аудитории, ее оснащенность оборудованием и техническими средствами обучения
1	Лекционные занятия	Аудитория, укомплектованная специализированной мебелью и техническими средствами обучения
2	Лабораторные занятия	Аудитория, укомплектованная специализированной мебелью и техническими средствами обучения, компьютерами, проектором, программным обеспечением
3	Групповые (индивидуальные) консультации	Аудитория, укомплектованная специализированной мебелью и техническими средствами обучения, компьютерами, программным обеспечением
4	Текущий контроль, промежуточная аттестация	Аудитория, укомплектованная специализированной мебелью и техническими средствами обучения, компьютерами, программным обеспечением
5	Самостоятельная работа	Кабинет для самостоятельной работы, оснащенный компьютерной техникой с возможностью подключения к сети «Интернет», программой экранного увеличения и обеспеченный доступом в электронную информационно-образовательную среду университета.

Примечание: Конкретизация аудиторий и их оснащение определяется ОПОП.