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.
подпись
« 29 » августа 2025 г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ (МОДУЛЯ)

Б1. В.08 Методы обучения с подкреплением

Направление подготовки 02.03.03 Математическое обеспечение и администрирование информационных систем

Профиль Искусственный интеллект и аналитика данных

Форма обучения очная

Квалификация бакалавр

Краснодар 2025

Рабочая программа дисциплины «Методы обучения с подкреплением» составлена в соответствии с федеральным государственным образовательным стандартом высшего образования (ФГОС ВО) по направлению подготовки 02.03.03 Математическое обеспечение и администрирование информационных систем.

Программу составил(и):

А. А. Кадурин, ст. преподаватель кафедры

Е.В. Казаковцева, доцент кафедры анализа данных и искусственного интеллекта, к. ф.-м. н.

Рабочая программа дисциплины утверждена на заседании центра искусственного интеллекта
протокол № 01 «28» августа 2025 г.
Руководитель центра ИИ Коваленко А.В.

подпись

Утверждена на заседании учебно-методической комиссии факультета компьютерных технологий и прикладной математики
протокол № 01 «28» августа 2025 г.
Председатель УМК факультета Коваленко А.В.

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

Цели дисциплины

− познакомить студентов с основными разделами, концепциями и принципами обучения с подкреплением, включая ключевые термины, модели и алгоритмы.;

− обеспечить глубокое понимание теоретических основ, таких как марковские процессы принятия решений (MDP), функции ценности, политики и стратегии;

− научить студентов применять методы обучения с подкреплением на практике, включая разработку, реализацию и тестирование алгоритмов на реальных и синтетических задачах.

1.2 Задачи дисциплины

− изучить основные задачи, решаемые при помощи методов обучения с подкреплением;

− развить навыки анализа и оценки эффективности различных алгоритмов обучения с подкреплением, а также умение выбирать подходящий метод в зависимости от задач;

− изучить библиотеки, необходимые при работе с методами обучения с подкреплением на Python (TF-Agents/ PyTorch RL/ OpenAi Gym)

1.3 Место дисциплины (модуля) в структуре образовательной программы

Дисциплина «Методы обучения с подкреплением» относится к части, формируемой участниками образовательных отношений Блока 1 «Дисциплины (модули)» учебного плана. Для успешного освоения данной дисциплины необходимы знания следующих дисциплин: Математический анализ, Векторная алгебра, Обработка данных на Python, Машинное обучение, Нейросетевые технологии, A/B-тестирование и Uplift-моделирование, Разработка ИИ-решений для индустрии.

1.4 Профессиональные роли в структуре образовательной программы

Роль 1: Data Analyst (Аналитик данных)

Задачи:

1. Статистический анализ, визуализация данных, предварительная обработка.

2. Создание прогнозных моделей

3. Построение аналитических моделей для поддержки бизнес-решений.

Роль 2: MLOps (Специалист по эксплуатации ИИ)

Задачи:

1. DevOps для ML.

2. Автоматизация, мониторинг ML-систем.

3. Операционное управление жизненным циклом ML-моделей.

Роль 3: AI PM (Менеджер проектов ИИ)

Задачи:

1. Управление ИИ-проектами от идеи до внедрения

2. Анализ бизнес-требований и постановка задач

3. Оценка эффективности и ROI ИИ-решений

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:

Код, уровень и формулировка компетенции	**Индикаторы**	**Уровни освоения индикаторов компетенции**
ПК-2 Способен участвовать в исследовании новых	ПК-2.1 Умеет анализировать и адаптировать существующие математические модели для	Выполняет анализ и адаптацию существующих математических моделей в

математических моделей в прикладных областях	решения прикладных задач в конкретной предметной области	задачах обучения с подкреплением
	ПК-2.2 Способен предлагать и обосновывать новые математические подходы для моделирования процессов в прикладных исследованиях	Предлагает новые математические подходы для моделирования процессов в обучении с подкреплением
ML-6 Способен применять алгоритмы обучения с подкреплением	ML-6.1 Обосновывает способы и варианты применения алгоритмов обучения с подкреплением в задачах ИИ, включая их преобразование и адаптацию к специфике задачи	(Б) Описывает основные принципы обучения с подкреплением (агент, среда, награда) и обосновывает выбор простейших алгоритмов (Q-Learning, SARSA) для решения типовых задач
О-2 Способен применять и (или) разрабатывать мультиагентные алгоритмы	О-2.4 Оценивает результативность применения мультиагентных алгоритмов в задачах ИИ на основе сопоставления с аналогами	(П) Создает метрики качества решения задач ИИ, в которых учитывается эффект самоорганизации агентов
LLM-4 Проектирует, разрабатывает и интегрирует интеллектуальных агентов на базе генеративных моделей	LLM-4.1 Умеет применять и разрабатывать интеллектуальных агентов	(Б) Использует простейших агентов в пайплайнах

2. Структура и содержание дисциплины

2.1 Распределение трудоёмкости дисциплины по видам работ

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

Вид учебной работы	**Всего часов**	**Семестры (часы)**					
		7					
Контактная работа, в том числе:	**36.2**	**36.2**					
Аудиторные занятия (всего):	**34**	**34**					
Занятия лекционного типа	**18**	18					
Лабораторные занятия	**16**	16					
Занятия семинарского типа (семинары, практические занятия)							
Иная контактная работа:	**0.2**	**0.2**					
Контроль самостоятельной работы (КСР)	**2**	2					
Промежуточная аттестация (ИКР)							
Самостоятельная работа, в том числе:	**35.8**	**35.8**					
Курсовая работа	**-**	-					
Проработка учебного (теоретического) материала							
Выполнение индивидуальных заданий (подготовка сообщений, презентаций)							
Реферат							
Подготовка к текущему контролю							
Контроль:							
Подготовка к экзамену	**-**	-					
Общая трудоемкость **час.**	**72**	**72**					

	в том числе контактная работа	**36.2**	**36.2**					
	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.	Введение в обучение с подкреплением (RL)	11	4		2	5
2.	Табличные методы обучения с подкреплением	8,8	2		2	4,8
3.	Приближённые методы RL	12	4		2	6
4.	Политико-ориентированное обучение	10	2		2	6
5.	Усовершенствованные методы обучения с подкреплением	10	2		2	6
6.	Применения глубокого RL	14	2		6	6
7.	Тренды в RL	6	2			4
ИТОГО по разделам дисциплины		**69,8**	**18**		**16**	**35.8**
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.	Введение в обучение с подкреплением (RL)	Основные понятия RL: агент, среда, политика, вознаграждение, марковская цепь принятия решений (MDP). Политики агентов: жадные политики, ε-жадные политики, softmax-политики. Классификация задач RL: эпизодические и непрерывные задачи, разреженные вознаграждения	ЛР
2.	Табличные методы обучения с подкреплением	Value iteration и Policy Iteration: определение значений состояний и политик. Q-learning: обновление Q-функции, offline и online обучение. SARSA: On-policy vs off-policy методы	ЛР

№	Наименование раздела (темы)	Содержание раздела (темы)	Форма текущего контроля
1	2	3	4
3.	Приближённые методы RL	Линейные аппроксимационные методы: представление состояний и действий, функции приближения. Глубокое обучение с подкреплением (DRL): введение в нейронные сети для RL. Q-networks и Deep Q-Learning (DQN): опыт воспроизведения, target networks, Double DQN.	ЛР
4.	Политико-ориентированное обучение	REINFORCE: максимизация ожиданий наград через градиент политики. Actor-Critic методы: сочетание политики и ценности. Advantage Actor-Critic (A2C/A3C): advantage function и асинхронное обучение.	ЛР
5.	Усовершенствованные методы обучения с подкреплением	Динамическая оптимизация траекторий (Trajectory optimization): методы TRPO и PPO. Обобщённая экстраполяция политик (Generalized Policy Improvement). Imitation Learning и Inverse Reinforcement Learning (IRL).	ЛР
6.	Применения глубокого RL	Применение DRL в играх (Atari, AlphaGo, StarCraft). Robotics и автономные транспортные средства. RL в экономических системах и финансовом трейдинге.	ЛР
7.	Тренды в RL	Multitask и Meta-Reinforcement Learning. Эффективность обучения в условиях неопределённости и частичной наблюдаемости. Социально-экономические и этические аспекты RL.	Т

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

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.

2.3.3 Лабораторные занятия

№	Наименование раздела (темы)	Наименование лабораторных работ	Форма текущего контроля
1	2	3	4
1.	Введение в обучение с подкреплением (RL)	Лабораторная работа №1: «Простейший агент для игры GridWorld»	ЛР
2.	Табличные методы обучения с подкреплением	Лабораторная работа №2: Value Iteration и Policy Iteration в среде Taxi-v3	ЛР
3.	Приближённые методы RL	Лабораторная работа №3: Q-Learning с линейной аппроксимацией для Cartpole	ЛР
4.	Политико-ориентированное обучение	Лабораторная работа №4: REINFORCE для MountainCarContinuous	ЛР
5.	Усовершенствованные методы обучения с подкреплением	Лабораторная работа №5: Trust Region Policy Optimization (TRPO) на примере LunarLander	ЛР
6	Применения глубокого RL	Лабораторная работа №6: AlphaGo-style Agent для настольной игры (Connect Four) Лабораторная работа №7: Управление машиной с помощью RL с использованием pytorch-rl	ЛР

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)

Не предусмотрены

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 УСФ, протокол №1 от 30.06.2025
2	Решение задач	Методические указания по организации самостоятельной работы студентов, утвержденные УСФ, протокол №1 от 30.06.2025

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.

Семестр	**Вид занятия**	**Используемые интерактивные образовательные технологии**	**количество интерактивных часов**
7	ЛР	Практические занятия в режимах взаимодействия «преподаватель – студент» и «студент – студент»	12
Итого			**12**

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 «Методы обучения с подкреплением».

Оценочные средства включает контрольные материалы для проведения **текущего контроля** в форме тестовых заданий, лабораторных работ и **промежуточной аттестации** в форме вопросов и заданий 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 зачете;

– 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	Введение в обучение с подкреплением (RL)	ПК-2.1, ПК-2.2, О-2.4	*Тестирование, Лабораторная работа №1*	*Вопросы к зачету 1-8*
2	Табличные методы обучения с подкреплением	ПК-2.1, ПК-2.2, ML-6.1	*Тестирование, Лабораторная работа №2*	*Вопросы к зачету 9-15*
3	Приближённые методы RL	ПК-2.1, ПК-2.2, ML-6.1	*Тестирование, Лабораторная работа №3*	*Вопросы к зачету 16-20*

4	Политико-ориентированное обучение	ПК-2.1, ПК-2.2, ML-6.1	*Тестирование, Лабораторные работы №4*	*Вопросы к зачету 21-25*
5	Усовершенствованные методы обучения с подкреплением	ПК-2.1, ПК-2.2, ML-6.1	*Тестирование, Лабораторная работа №5*	*Вопросы к зачету 26-30*
6	Применения глубокого RL	ПК-2.1, ПК-2.2, ML-6.1	*Тестирование, Лабораторные работы №6,7*	*Вопросы к зачету 31-34*
7	Тренды в RL	LLM-4.1, О-2.4	*Тестирование*	*Вопросы к зачету 35-40*

Показатели, критерии и шкала оценки сформированных компетенций

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

ПК-2 ***Способен участвовать в исследовании новых математических моделей в прикладных областях***

Выполняет анализ и адаптацию существующих математических моделей в задачах обучения с подкреплением

Предлагает новые математические подходы для моделирования процессов в обучении с подкреплением

ML-6 Б ***Способен применять алгоритмы обучения с подкреплением***

Описывает основные принципы обучения с подкреплением (агент, среда, награда) и обосновывает выбор простейших алгоритмов (Q-Learning, SARSA) для решения типовых задач

О-2 П ***Способен применять и (или) разрабатывать мультиагентные алгоритмы***

Создает метрики качества решения задач ИИ, в которых учитывается эффект самоорганизации агентов

Использует метрики эффективности для сопоставления мультиагентных алгоритмов с традиционными методами решения задач ИИ

LLM-4 Б ***Проектирует, разрабатывает и интегрирует интеллектуальных агентов на базе генеративных моделей***

Использует простейших агентов в пайплайнах

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

Пример тестирования

1. Что такое агент в контексте RL?
   - a) Программа, принимающая внешние команды.
   - b) Элемент среды, контролирующий состояние мира.
   - c) Алгоритм, выбирающий действия для достижения целей.
   - d) Внешнее устройство ввода данных.
2. Выберите правильное утверждение относительно политики агента:
   - a) Политика определяет стратегию агента по выбору действий.
   - b) Политика описывает структуру окружения.
   - c) Политика устанавливает правила остановки эпизода.

- d) Политика регулирует награду агента.

3. В каком виде представлена функция вознаграждения в RL?
   - a) Только числовое значение.
   - b) Число или вектор чисел.
   - c) Всегда положительное число.
   - d) Матрица вероятностей.
4. Что такое эпизодическая задача в RL?
   - a) Повторяющаяся последовательность эпизодов с одинаковыми начальными условиями.
   - b) Задача с фиксированным числом шагов.
   - c) Задача с конечным состоянием завершения.
   - d) Постоянная циклическая активность.
5. Чем отличается жадная политика от εε-жадной?
   - a) εε-жадная политика допускает случайные действия.
   - b) Жадная политика игнорирует будущие вознаграждения.
   - c) εε-жадная политика гарантирует оптимальное поведение.
   - d) Нет разницы между ними.
6. Что значит фраза "value iteration"?
   - a) Процесс повторения одного и того же действия.
   - b) Последовательное изменение оценок стоимости состояний.
   - c) Метод присвоения весов действиям.
   - d) Получение постоянных значений вознаграждения.
7. Как называется ситуация, когда обновляется таблица Q-значений непосредственно во время взаимодействия с окружающей средой?
   - a) Offline обучение.
   - b) Online обучение.
   - c) Batch обучение.
   - d) Reinforce обучение.
8. Какой алгоритм относится к категории on-policy?
   - a) Q-Learning.
   - b) SARSA.
   - c) Monte Carlo Control.
   - d) TD($\lambda\lambda$).
9. В чём основное различие между Q-Learning и SARSA?
   - a) Используют разные функции вознаграждения.
   - b) Q-Learning учитывает последующие действия, а SARSA — нет.
   - c) SARSA выбирает следующее действие по полученной политике, тогда как Q-Learning делает это независимо.
   - d) Q-Learning поддерживает жадную политику, а SARSA — nope.
10. Что такое линейная аппроксимация в RL?
    - a) Замена таблицы состояний набором уравнений.
    - b) Упрощённый способ расчета Q-значений.
    - c) Процесс подбора коэффициентов, наилучшим образом отображающих зависимость между входами и выходами.
    - d) Метод группировки похожих состояний.
11. Каким способом достигается масштабируемость RL при увеличении числа состояний?
    - a) Увеличением размера таблицы Q-значений.
    - b) Линейной аппроксимацией Q-функции.
    - c) Использованием большего количества данных.
    - d) Снижением размеров сети.
12. Какую основную проблему позволяет решить механизм "experience replay" в DQN?

- a) Преодолевает коррелированность данных.
- b) Позволяет сохранять историю действий.
- c) Предсказывает следующую позицию агента.
- d) Регулирует размер памяти.

13. Что такое target network в DQN?
    - a) Это сеть, используемая для предсказания следующего состояния.
    - b) Это копия основной сети, которая периодически обновляется.
    - c) Это вспомогательная сеть для обучения основного агента.
    - d) Это специальная структура для сбора данных.
14. В чём главная особенность Double DQN?
    - a) Используется две отдельные сети для повышения надёжности.
    - b) Отдельная сеть оценивает ценность будущих действий.
    - c) Повышается точность оценки текущего состояния.
    - d) Она обучается вдвое быстрее.
15. Что такое градиент политики в REINFORCE?
    - a) Значение оптимальной стратегии агента.
    - b) Скорость адаптации параметров агента.
    - c) Производная среднего вознаграждения по параметрам политики.
    - d) Вероятность выбора конкретного действия.
16. В чём заключается подход actor-critic?
    - a) Актор выбирает действия, критик оценивает ситуацию.
    - b) Оба компонента совместно выбирают лучшие действия.
    - c) Один компонент формирует действия, другой их проверяет.
    - d) Они работают отдельно друг от друга.
17. Чем принципиально отличается A2C от REINFORCE?
    - a) A2C использует отдельный модуль для оценки ценности состояний.
    - b) REINFORCE обучается значительно быстрее.
    - c) A2C имеет меньшую вариативность оценки вознаграждения.
    - d) REINFORCE больше подходит для больших задач.
18. Что обозначает сокращение TRPO?
    - a) Temporal Regularization Policy Optimization.
    - b) Trust Region Policy Optimization.
    - c) Task-based Random Policy Optimization.
    - d) Time-series Regression Predictor Optimization.
19. В чём преимущество метода PPO перед TRPO?
    - a) Более стабильное обучение.
    - b) Менее требовательный к ресурсам.
    - c) Быстрая адаптация к изменениям среды.
    - d) Возможность параллельного обучения.
20. Что такое imitation learning?
    - a) Моделирование поведения экспертов.
    - b) Простой перенос существующих стратегий.
    - c) Стремление достичь идеальных действий.
    - d) Самостоятельное формирование стратегии.
21. В какой игре впервые продемонстрировала свою силу технология DRL?
    - a) Pac-Man.
    - b) Chess.
    - c) Go.
    - d) Starcraft.
22. Какое применение находит DRL в финансовой сфере?
    - a) Оптимизация портфеля активов.
    - b) Прогноз цен акций.

- c) Быстрое принятие торговых решений.
- d) Все вышеперечисленные пункты.

23. В чём особенность использования RL в робототехнике?
    - a) Необходимость физического присутствия робота.
    - b) Высокие затраты на разработку.
    - c) Сложность восприятия внешней среды.
    - d) Требование реального времени реакции.
24. Какая задача особенно сложна для RL в беспилотных автомобилях?
    - a) Нахождение парковочного места.
    - b) Ориентация в городской среде.
    - c) Распознавание пешеходов.
    - d) Переход на нужную полосу движения.
25. Какие особенности отличают Meta-Reinforcement Learning?
    - a) Решение одной конкретной задачи.
    - b) Поддерживает быстрое обучение новым задачам.
    - c) Используется исключительно в игровых приложениях.
    - d) Высокая сложность реализации.

Перечень компетенций (части компетенции), проверяемых оценочным средством

ML-6.1

Практические кейсы по тематике лабораторных работ

Лабораторная работа №1: Простейший агент для игры GridWorld

Практический кейс: Разработать простого агента, который учится перемещаться по лабиринту GridWorld, избегая преград и находя выход за минимальное количество шагов. Агент может двигаться в четырёх направлениях: вверх, вниз, влево и вправо. В лабиринте расставлены награды (+10 очков за переход к выходу, −1 за каждый сделанный шаг, −10 за столкновение с препятствием).

Задачи:

Реализовать лабиринт GridWorld с простыми правилами (награды, наказания).

Обучить агента избегать стен и достигать выхода.

Провести эксперименты с разными параметрами обучения (rate, epsilon-greedy, размер сетки).

Лабораторная работа №2: Value Iteration и Policy Iteration в среде Taxi-v3

Практический кейс: Реализовать таксиста-агента, который перевозит пассажиров из одной точки города в другую, минимизируя количество поездок и штрафов за неправильное перемещение.

Задачи:

Реализовать агент на основе Value Iteration и Policy Iteration.

Провести сравнительное исследование эффективности Value Iteration и Policy Iteration.

Оценить устойчивость полученного решения и зависимость от начальных условий.

Лабораторная работа №3: Q-Learning с линейной аппроксимацией для Cartpole

Практический кейс: Создать агента, управляющего тележкой с палкой, используя Q-Learning с линейной аппроксимацией. Палка должна оставаться строго вертикальной, а тележка — двигаться горизонтально.

Задачи:

Реализовать Q-Learning с линейной аппроксимацией Q-функции.

Настроить гиперпараметры (epsilon, learning rate, discount factor).

Оценить качество баланса палки и устойчивость найденной стратегии.

Лабораторная работа №4: REINFORCE для MountainCarContinuous
Практический кейс: Построить агента, который управляет автомобилем, находящимся в долине, с задачей подъёма на вершину ближайшей горы. Автомобиль ограничен низкой мощностью двигателя и должен накопить достаточную кинетическую энергию для восхождения.
Задачи:
Реализовать агента на основе метода REINFORCE.
Исследовать влияние количества шагов на качество обучения.
Сравнить результаты с Q-Learning и другими методами RL.

Лабораторная работа №5: Trust Region Policy Optimization (TRPO) на примере LunarLander
Практический кейс: Разработать агента, осуществляющего мягкую посадку лунохода на поверхности Луны, используя метод Trust Region Policy Optimization (TRPO).
Задачи:
Реализовать TRPO-агента для задачи посадки.
Минимизировать топливные расходы и повреждения при посадке.
Исследовать чувствительность к параметрам алгоритма (step-size, trust region radius).

Лабораторная работа №6: AlphaGo-style Agent для настольной игры (Connect Four)
Практический кейс: Создать агента, способного играть в настольную игру «Four-in-a-row» (Connect Four), используя подход, аналогичный AlphaGo.
Задачи:
Реализовать агента, комбинирующий RL и Monte Carlo Tree Search (MCTS).
Обучить агента обыгрывать стандартный Minimax-агента с ограниченной глубиной дерева.
Провести турнир между разными версиями агентов, сравнивая их эффективность.
Лабораторная работа №7: Управление машиной с помощью RL с использованием pytorch-rl
Практический кейс: Создать агента, управляющегося автомобилем в ралли-симуляторе (например, в CarRacing-v0), используя PyTorch и библиотеку pytorch-rl.
Задачи:
Реализовать агента с использованием RL (PPO, DQN или другого метода).
Оценить эффективность различных подходов к управлению.
Добиться максимальной средней дистанции, проходимой автомобилем за раунд гонки.

Пример лабораторной работы

Лабораторная работа №1: Управление автомобилем с помощью RL с использованием pytorch-rl
Цель:
Освоить реализацию алгоритмов reinforcement learning (RL) с использованием библиотеки pytorch-rl для управления виртуальным автомобилем.

Для демонстрации возьмем среду автономного вождения на примере платформы CARLA.
Ход работы:
1. Установка зависимостей и библиотек
pip install carla pytorch-rl numpy
import carla
import torch
import torch.nn as nn
import torch.optim as optim

2. Запуск сервера Carla
Запустите сервер игры Carla перед началом эксперимента:./CarlaUE4.sh

3. Написание скрипта клиента для подключения к среде
Создаем простой клиентский сценарий для связи с Carla:
import carlaclient = carla.
Client('localhost', 2000)
world = client.get_world()

4. Добавляем автомобиль и окружение
Создаем объект автомобиля и размещаем его в мире:
blueprint_library = world.get_blueprint_library()
vehicle_bp = blueprint_library.find('vehicle.tesla.model3')
spawn_point = random.choice(world.get_map().get_spawn_points())
vehicle = world.spawn_actor(vehicle_bp, spawn_point)

5. Алгоритм обучения RL
Применяем базовую версию Deep Q-learning (DQN) для обучения управлению автомобилем:

```
import torch
import torch.nn as nn
import torch.optim as optim
class SimpleDQN(nn.Module):
    def init(self, input_shape, n_actions):
        super(SimpleDQN, self).init()
        self.fc = nn.Sequential(
            nn.Linear(input_shape, 128),
            nn.ReLU(),
            nn.Linear(128, n_actions)
        )
    def forward(self, x):
        return self.fc(x)
# Параметры обучения
input_shape = vehicle.get_state().shape
n_actions = 3  # вперед, вправо, влево
dqn_model = SimpleDQN(input_shape, n_actions)
optimizer = optim.Adam(dqn_model.parameters(), lr=0.001)
loss_fn = nn.MSELoss()
```

6. Тренировка модели
Начинаем тренировочный цикл:

```
def train_dqn(episodes=100):
    for episode in range(episodes):
        state = vehicle.get_state()
        action = dqn_model(torch.tensor(state)).argmax(dim=-1)
        next_state, reward, done = step(action)
        target_q_value = reward + gamma * max(q_values(next_state))
        current_q_value = q_values(state)[action]
        loss = loss_fn(current_q_value, target_q_value)
        optimizer.zero_grad()
        loss.backward()
        optimizer.step()
```

7. Оценка результатов

Тестируем модель в условиях реальной дороги:
```
test_episodes = 10
total_reward = 0
for _ in range(test_episodes):
   state = vehicle.get_state()
   while not done:
      action = dqn_model(torch.tensor(state)).argmax(dim=-1)
      next_state, reward, done = step(action)
      total_reward += reward
average_reward = total_reward / test_episodesprint(f"Средний доход на этапе теста: {average_reward}")
```

Требования к отчету
1. Титульный лист
Название работы, ФИО студента, группа, дата.

2. Введение
Цель и задачи работы.
Описание библиотек и сред обучения.

3. Алгоритм обучения с подкреплением
Подготовка данных.
Описание алгоритма обучения.
Описание параметров обучения, нейронной сети и функции потерь.

4. Реализация
Код на Python с комментариями.
Примеры до/после обработки.

5. Результаты
Метрики

6. Выводы
Проблемы, возникшие при обработке.
Предложения по улучшению.

7. Приложения
Исходный код.
Словарь сленга.

Критерии оценки:
- **Зачтено:** Полное выполнение всех шагов, демонстрация качественного понимания поведения модели, адекватный анализ полученного результата.
- **Не зачтено:** Ключевые этапы выполнения пропущены или содержат критические ошибки.

Проверяемые компетенции комплексом практических заданий: ML-6.1

Зачетно-экзаменационные материалы для промежуточной аттестации (зачет)

Вопросы для подготовки к зачету

1. Назовите основные компоненты модели RL и дайте определения каждому элементу.

2. Что такое политика (policy)? Какие виды политик существуют?
3. Охарактеризуйте различия между жадными политиками и $\varepsilon\varepsilon$-жадными политиками.
4. Для чего используется функция полезности (reward function) в RL?
5. В чём разница между эпизодическими и непрерывными задачами в RL?
6. Что означает термин «разреженное вознаграждение» (sparse rewards)?
7. Определите понятие Марковской цепи принятия решений (MDP).
8. В чём состоит проблема оптимального планирования в пространстве состояний?
9. Что представляет собой метод value iteration? Когда он применяется?
10. Что такое policy iteration и в чём заключается его преимущество над value iteration?
11. Расскажите о принципе обновления Q-значений в Q-learning.
12. Чем отличаются on-line и off-line режимы обучения в Q-learning?
13. В чём принципиальное различие между Q-learning и SARSA?
14. Почему SARSA называют методом on-policy, а Q-learning — off-policy?
15. Опишите, каким образом строится стратегия обучения с помощью временных разниц (temporal difference learning).
16. Зачем используются линейные аппроксимационные методы в RL?
17. Какие преимущества имеют функции приближения (approximation functions)?
18. Какова цель представления пространств состояний и действий в приближённых методах RL?
19. В чём суть метода feature engineering применительно к RL?
20. Опишите процесс аппроксимации функций полезности с помощью линейных моделей.
21. Что представляют собой Q-networks и какую проблему они решают?
22. В чём смысл введения механизма experience replay в Deep Q-Learning (DQN)?
23. Объясните концепцию target network в DQN и её влияние на устойчивость обучения.
24. В чём заключаются отличия обычного DQN от Double DQN?
25. Какие ещё механизмы применяются для стабилизации обучения в DQN?
26. Как работает метод REINFORCE и зачем нужны градиенты политики?
27. В чём заключается идея actor-critic методов и как они улучшают обучение?
28. Что такое advantage function и как он улучшает training process в A2C/A3C?
29. Какие плюсы даёт асинхронное обучение в A3C?
30. В чём особенность trust region оптимизации в политике (TRPO)?
31. В чём специфика метода Proximal Policy Optimization (PPO)?
32. Как метод imitation learning решает проблему переноса экспертных знаний в RL?
33. Что такое inverse reinforcement learning (IRL) и как он используется в RL?
34. Что подразумевают под generalized policy improvement?
35. Где применяется DRL в компьютерных играх (примеры Atari, AlphaGo, StarCraft)?
36. Какие задачи решают RL-методы в робототехнике и самоуправляемых транспортных средствах?
37. Приведите примеры применения RL в экономике и финансовых рынках.
38. Какие направления развиваются в multitask и meta-reinforcement learning?
39. Как влияет неполная наблюдаемость среды на эффективность обучения RL?
40. Какие социально-экономические и этические вопросы возникают при применении RL?

Перечень компетенций (части компетенции), проверяемых оценочным средством
ПК-2.1, ПК-2.2, ML-6.1, O-2.4, LLM-4.1

Практические задания к зачету

1. Создание простейшей среды RL.

Разработайте свою собственную минимальную окружающую среду (environment) с двумя действиями («налево» и «направо»). Покажите, как создать простого агента с случайной стратегией, вычислить значение вознаграждения и завершить эпизод.

2. Жадные и ε-жадные политики.

Реализуйте два типа агентов с разными политиками: один действует согласно жадному принципу, второй — согласно ε-жадному. Продемонстрируйте разницу в поведении агентов в одном и том же окружении.

3. Пространство состояний и переходы в Markovian Environment.

Определите пространство состояний для заданной игровой среды (например, GridWorld). Рассчитайте вероятности перехода между состояниями и создайте матрицу переходов для простых движений (слева направо, вверх-вниз).

4. Реализация Value Iteration и Policy Iteration.

Для небольшой карты размером 4×4 решите задачу поиска пути от начального положения до цели с помощью двух классических табличных методов: Value Iteration и Policy Iteration. Сравните полученные результаты и обсудите их сходства и различия.

5. Алгоритм Q-Learning.

Решите задачу оптимизации маршрута для GridWorld с помощью Q-Learning. Используйте разные варианты обучения (offline и online) и сравните их эффективность.

6. On-policy и off-policy обучение.

Покажите разницу между этими двумя подходами, выполнив одно задание дважды: сначала с помощью SARSA (on-policy), затем с помощью Q-Learning (off-policy). Проанализируйте результаты и сделайте выводы.

7. Линейная аппроксимация Q-values.

Модифицируйте Q-Learning таким образом, чтобы заменить таблицу Q-values функцией линейной регрессии. Решите задачу навигации в произвольной среде (например, лабиринте), демонстрируя эффективность нового подхода.

8. Feature Engineering в RL.

Представьте себя инженером по данным. Разработайте набор признаков (features) для задачи обхода препятствий в двумерном пространстве и покажите, как эти признаки помогают улучшить обучение агента.

9. Real-time игра с использованием DQN.

Воспользуйтесь платформой OpenAI Gym и классическим примером CartPole-v1. Создайте и натренируйте агента на основе DQN с использованием target network и replay memory. Показать динамику обучения.

10. Double DQN против стандартного DQN.

Продемонстрируйте эффект двойного DQN (Double DQN) на задаче из OpenAI Gym. Постройте графики качества обучения и продемонстрируйте улучшение показателя по сравнению с обычным DQN.

11. REINFORCE Algorithm.

Реализуйте алгоритм REINFORCE для задачи Atari Breakout из OpenAI Gym. Измерьте среднее количество очков, набранных агентом, и проведите эксперименты с различными значениями параметра скорости обучения.

12. Actor-Critic Approach.

Создайте собственного агента с архитектурой Actor-Critic для простой задачи (например, MountainCar-v0) и оцените его способность достигать цели быстрее, чем стандартный DQN.

13. Преимущества асинхронного обучения (A3C).

Покажите, как асинхронное обучение ускоряет и стабилизирует процесс обучения в сложных средах (например, LunarLander-v2). Посмотрите на динамику изменения награды и сопоставьте результаты с обычными последовательными подходами.

14. Proximal Policy Optimization (PPO).

Реализуйте PPO на простой задаче (например, Ant-v2 из MuJoCo) и сравните с результатом A2C. Обсудите преимущества и ограничения каждого подхода.

15. Inverse Reinforcement Learning (IRL).

Решите задачу обучения с помощью примеров эксперта (expert demonstration). Воспользуйтесь методом Maximum Entropy Inverse Reinforcement Learning (MaxEnt IRL) и воссоздайте политику эксперта на небольшом примере (GridWorld или аналогичном).

Перечень компетенций (части компетенции), проверяемых оценочным средством

ПК-2.1, ПК-2.2, ML-6.1, O-2.4, LLM-4.1

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

Методические рекомендации, определяющие процедуры оценивания на зачете:

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 вопросов к зачету и результатов текущего контроля.

Зачет по дисциплине преследует цель оценить работу студента за курс, получение теоретических знаний, их прочность, развитие творческого мышления, приобретение навыков самостоятельной работы, умение применять полученные знания для решения практических задач.

Форма проведения зачета: устно.

Результат сдачи зачета заноситься преподавателем в зачетную ведомость и зачетную книжку.

Оценивание уровня освоения дисциплины основывается на качестве выполнения студентом заданий текущего контроля и ответов на вопросы зачета.

Критерии оценки:

1. Оценка ответов на вопросы к зачету (40% итоговой оценки)

Зачет

Полные, развернутые ответы с демонстрацией глубокого понимания темы.
Ответы содержат основные идеи, но без углубленного анализа.
Использование примеров, формул, корректных терминов.
Возможны небольшие ошибки в деталях или формулировках.
Умение анализировать и сравнивать методы.
% выполнения: 60–100% (допускаются незначительные неточности).

Незачет

Отсутствие понимания ключевых концепций.
Грубые ошибки или неспособность ответить на большую часть вопросов.

% выполнения: <60%.

2. Оценка выполнения практических кейсов и лабораторных работ (40% итоговой оценки)

Зачет

Полное выполнение всех этапов кейса с инновационными решениями.
Достижение целевых метрик (например, F1 > 0.9).
Четкая документация кода и анализ результатов.
% выполнения: 60–100%.

или

Выполнены основные задачи, но без дополнительной оптимизации.
Незначительные отклонения от целевых метрик (например, F1 = 0.85).

или

Решены базовые задачи, но с критическими ошибками.
Низкое качество кода или отсутствие анализа.

Незачет

Невыполнение ключевых этапов.
Код нерабочий или отсутствует.
% выполнения: <60%.

3. Оценка тестовых вопросов (20% итоговой оценки)

Зачет

18–25 правильных ответов (70–100%).
Демонстрация уверенного владения терминологией и методами.

Незачет

Менее 18 правильных ответов (<70%).
Неспособность отличить архитектуры (например, многослойный персептрон от спайковой сети).

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 зачете;

– 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

Условия применения:

- Курс рассчитан на студентов 4-го года обучения.
- Наличие доступа к вычислительным ресурсам (GitLab, Google Colab или Yandex DataSphere, JupyterHub).
- Разработаны лабораторные работы;
- Инфраструктура для приёма задач (gitlab, CI/CD) согласована с лабораторными работами.

Цели, задачи и ожидаемые результаты

Цели организации вычислительной инфраструктуры:

- дать представление о работе в IT инфраструктуре (приучить пользоваться гитом, jupyter-ноутбуками).

Задачи преподавателя:

- Организация регистрации студентов в Google Colab и Yandex DataSphere
- Создание учетных записей студентов в gitlab вуза;
- Настройка GitLab Runner для автоматического тестирования кода.
- Разработка шаблонного репозитория для лабораторных работ с предустановленными зависимостями (TF-Agents/ PyTorch RL/ OpenAi Gym).
- Написание автотестов для проверки корректности выполнения заданий.
- Визуализация результатов тестирования через HTML-отчеты.
- Подготовка инструкций по работе с Git и облачными ресурсами.

Ожидаемые результаты студентов:

- начальное представление о работе в IT инфраструктуре (гит, нейминг).
- Навыки запуска и тестирования моделей обучения с подкреплением в облачных средах.
- Понимание CI/CD-процессов в контексте разработки методов обучения с подкреплением.

Порядок реализации

Задача №1: Организация регистрации студентов в Google Colab и Yandex DataSphere

Задача №2: Создание учетных записей студентов в gitlab вуза

Задача №3: Настройка GitLab Runner:

Для автоматического тестирования кода используется Docker-образ с предустановленными библиотеками (TF-Agents/ PyTorch RL/ OpenAi Gym).

Для выполнения CI/CD пайплайна был настроен GitLab Runner на удалённой виртуальной машине с ОС Ubuntu 24.04.
Последовательность настройки включала следующие шаги:

- Настройка системы – установка необходимых компонентов, таких, как Docker.
- Установка GitLab Runner по официальной инструкции.
- Регистрация Runner для частного сервера GitLab.

Задача №4: Шаблонный репозиторий:
Включает:
.gitlab-ci.yml для CI/CD.
Скрипты для обучения моделей.
Примеры кода для работы с ИИ-агентами.
Задача №5: Автотесты:
Проверяют корректность моделей обучения с подкреплением.
Задача №6: Визуализация результатов:
Генерация HTML-отчетов с результатами тестирования, включая метрики качества моделей.
Порядок проверки корректности:

- Наличие Git-репозиториев у всех студентов.
- Шаблонный репозиторий с подключенными автотестами.
- Инструкция по работе с Git и CI/CD в формате README.md.

Вся структура максимально адаптирована для копирования студентами и минимизации порога входа при выполнении лабораторных

4.4. Методические указания по организации лабораторных работ

Условия применения:

- Курс рассчитан на студентов 4-го года обучения.
- Наличие доступа к вычислительным ресурсам (GitLab) и к GPU/CPU (Kaggle, локальные серверы).
- Разработана инфраструктура для приёма задач (Gitlab, CI/CD) и согласована с лабораторными работами и настроена на всех студентов образовательной программы;
- Использование открытых датасетов и библиотек.

Цели, задачи и ожидаемые результаты
Цели организации лабораторных работ:

- Закрепление теоретических знаний на практике.
- Развитие навыков работы с методами обучения с подкреплением.
- Подготовка к решению реальных задач в индустрии.

Задачи преподавателя:

- Обеспечить студентов структурированными лабораторными работами.
- Предоставить доступ к необходимым вычислительным ресурсам.
- Организовать проверку и обратную связь по выполненным работам.

Ожидаемые результаты студентов:

- Умение применять методы обучения с подкреплением на практике.
- Владение библиотеками и фреймворками (TF-Agents/ PyTorch RL/ OpenAi Gym).
- Опыт работы с алгоритмами обучения с подкреплением, такими как Q-Learning, SARSA, а также их модификации и расширенные версии (DQN, A2C, A3C, TRPO, PPO)

Порядок реализации
Задача №1: Подготовка лабораторных работ (в соответствии с п. 2.3.3 РПД)
1) Определение тем:

- Простейший агент для игры GridWorld

- Value Iteration и Policy Iteration в среде Taxi-v3
- Q-Learning с линейной аппроксимацией для Cartpole
- REINFORCE для MountainCarContinuous
- Trust Region Policy Optimization (TRPO) на примере LunarLander
- AlphaGo-style Agent для настольной игры (Connect Four)
- Управление машиной с помощью RL с использованием pytorch-rl

Наименование Лабораторной работы	**Содержание Лабораторной работы**	**Распределение часов**
2	**3**	
Простейший агент для игры GridWorld	Реализовать среду GridWorld (сеточная среда). Создать базового агента, способного перемещаться по сетке, выбирать доступные действия и получать вознаграждение. Применить простую стратегию поведения (например, случайный выбор направлений движения). Изучить основы взаимодействия агента с окружающей средой, представление состояния и получение вознаграждения.	3 часа (2 часа ЛР, 1 час СР)
Value Iteration и Policy Iteration в среде Taxi-v3	Использовать встроенную среду OpenAI Gym Taxi-v3. Реализовать алгоритм Value Iteration для нахождения оптимальной стратегии путём последовательного обновления значений состояний. Реализовать алгоритм Policy Iteration для постепенного улучшения существующей стратегии и вычисления её ценности. Провести сравнение производительности обоих подходов и проанализировать полученные решения.	3 часа (2 часа ЛР, 1 час СР)
Q-Learning с линейной аппроксимацией для Cartpole	Работа с средой OpenAI Gym Cartpole, имеющей непрерывные переменные состояния. Создание модели, использующей линейную аппроксимацию значения функции Q. Обучение агента балансировать палочку на тележке, минимизируя потери и максимизируя продолжительность эпизода. Анализ влияния гиперпараметров на качество обучения и устойчивость модели.	3 часа (2 часа ЛР, 1 час СР)
REINFORCE для MountainCarContinuous	Использование среды OpenAI Gym MountainCarContinuous с непрерывными действиями. Реализация алгоритма REINFORCE с целью обучения агента управлять автомобилем таким образом, чтобы достичь вершины горы. Оценка качества полученных результатов и интерпретация итоговых траекторий обучения.	3 часа (2 часа ЛР, 1 час СР)
Trust Region Policy Optimization (TRPO) на	Применение метода TRPO в среде OpenAI Gym LunarLander. Разработка агента, способного эффективно приземляться на поверхность Луны, учитывая ограничения ресурсов и сложность управления.	3 часа (2 часа ЛР, 1 час СР)

Наименование Лабораторной работы	Содержание Лабораторной работы	Распределение часов
2	3	
примере LunarLander	Изучение особенностей метода TRPO: доверительные области, ограничение изменения политики, константа KL-дивергенции. Исследование преимуществ TRPO перед обычными методами Policy Gradients.	
AlphaGo-style Agent для настольной игры (Connect Four)	Разработка собственной версии игрового движка для Connect Four. Реализация дерева Монте-Карло с выбором и расширением узлов, оценкой положения и возвратом результата. Подключение глубокого обучения (нейронных сетей) для предсказания вероятностей хода и оценки позиций. Тщательная настройка параметров и тестирование обученной системы против простого противника или человеческого игрока.	4 часа (3 часа ЛР, 1 час СР)
Управление машиной с помощью RL с использованием pytorch-rl	Создание виртуальной среды вождения автомобиля (например, с использованием Carla или Unity ML-Agents). Реализация глубоконейронной сети для восприятия окружающего мира и выбора оптимальных управляющих воздействий. Настройка и запуск экспериментов с различными конфигурациями алгоритмов обучения с подкреплением. Оптимизация работы агента и оценка надёжности полученной модели управления транспортом.	4 часа (3 часа ЛР, 1 час СР)

2) Разработка заданий:
- Пошаговые инструкции.
- Примеры кода.

Контрольные вопросы.

Лабораторная работа №7: Управление автомобилем с помощью RL с использованием pytorch-rl
Цель:
Освоить реализацию алгоритмов reinforcement learning (RL) с использованием библиотеки pytorch-rl для управления виртуальным автомобилем.

2. Пошаговые инструкции
Для демонстрации возьмем среду автономного вождения на примере платформы CARLA.
Шаг 1: Установка зависимостей
pip install carla pytorch-rl numpy
Шаг 2: Запуск сервера Carla
Запустите сервер игры Carla перед началом эксперимента:./CarlaUE4.sh
Шаг 3: Написание скрипта клиента для подключения к среде
Создаем простой клиентский сценарий для связи с Carla:
import carlaclient = carla.
Client('localhost', 2000)

```
world = client.get_world()
```
Шаг 4: Добавляем автомобиль и окружение
Создаем объект автомобиля и размещаем его в мире:
```
blueprint_library = world.get_blueprint_library()
vehicle_bp = blueprint_library.find('vehicle.tesla.model3')
spawn_point = random.choice(world.get_map().get_spawn_points())
vehicle = world.spawn_actor(vehicle_bp, spawn_point)
```
Шаг 5: Алгоритм обучения RL
Применяем базовую версию Deep Q-learning (DQN) для обучения управлению автомобилем:
```
import torch
import torch.nn as nn
import torch.optim as optim
class SimpleDQN(nn.Module):
    def init(self, input_shape, n_actions):
        super(SimpleDQN, self).init()
        self.fc = nn.Sequential(
            nn.Linear(input_shape, 128),
            nn.ReLU(),
            nn.Linear(128, n_actions)
        )
    def forward(self, x):
        return self.fc(x)
# Параметры обучения
input_shape = vehicle.get_state().shape
n_actions = 3  # вперед, вправо, влево
dqn_model = SimpleDQN(input_shape, n_actions)
optimizer = optim.Adam(dqn_model.parameters(), lr=0.001)
loss_fn = nn.MSELoss()
```
Шаг 6: Тренировка модели
Начинаем тренировочный цикл:
```
def train_dqn(episodes=100):
    for episode in range(episodes):
        state = vehicle.get_state()
        action = dqn_model(torch.tensor(state)).argmax(dim=-1)
        next_state, reward, done = step(action)
        target_q_value = reward + gamma * max(q_values(next_state))
        current_q_value = q_values(state)[action]
        loss = loss_fn(current_q_value, target_q_value)
        optimizer.zero_grad()
        loss.backward()
        optimizer.step()
```
Шаг 7: Оценка результатов
Тестируем модель в условиях реальной дороги:
```
test_episodes = 10
total_reward = 0
for _ in range(test_episodes):
    state = vehicle.get_state()
    while not done:
        action = dqn_model(torch.tensor(state)).argmax(dim=-1)
        next_state, reward, done = step(action)
        total_reward += reward
```

average_reward = total_reward / test_episodesprint(f"Средний доход на этапе теста: {average_reward}")

3. Примеры кода на Python
Полный рабочий скрипт для запуска обучения и оценки модели:

```
import carla
import torch
import torch.nn as nn
import torch.optim as optim

# Создаем экземпляр клиента и подключаемся к серверу Carla
client = carla.Client('localhost', 2000)
world = client.get_world()

# Генерируем машину Tesla Model 3
blueprint_library = world.get_blueprint_library()
vehicle_bp = blueprint_library.find('vehicle.tesla.model3')
spawn_point = random.choice(world.get_map().get_spawn_points())
vehicle = world.spawn_actor(vehicle_bp, spawn_point)

# Архитектура DQN-модели
class SimpleDQN(nn.Module):
    def __init__(self, input_shape, n_actions):
        super(SimpleDQN, self).__init__()
        self.fc = nn.Sequential(
            nn.Linear(input_shape, 128),
            nn.ReLU(),
            nn.Linear(128, n_actions)
        )

    def forward(self, x):
        return self.fc(x)

# Начальная конфигурация
input_shape = vehicle.get_state().shape
n_actions = 3  # вперед, вправо, влево
dqn_model = SimpleDQN(input_shape, n_actions)
optimizer = optim.Adam(dqn_model.parameters(), lr=0.001)
loss_fn = nn.MSELoss()
gamma = 0.99

# Функция шага окружения
def step(action):
    # Действие выполнено в среде
    pass  # реализация зависит от конкретной задачи

# Основной цикл обучения
episodes = 100
for episode in range(episodes):
    state = vehicle.get_state()
    action = dqn_model(torch.tensor(state)).argmax(dim=-1)
    next_state, reward, done = step(action)
```

```
    target_q_value = reward + gamma * max(q_values(next_state))
    current_q_value = q_values(state)[action]
    loss = loss_fn(current_q_value, target_q_value)
    optimizer.zero_grad()
    loss.backward()
    optimizer.step()

# Оцениваем работу
test_episodes = 10
total_reward = 0
for _ in range(test_episodes):
    state = vehicle.get_state()
    while not done:
        action = dqn_model(torch.tensor(state)).argmax(dim=-1)
        next_state, reward, done = step(action)
        total_reward += reward
average_reward = total_reward / test_episodes
print(f"Средний доход на этапе теста: {average_reward}")
```

4. Контрольные вопросы

Теоретические вопросы:

Что такое Q-value и почему важно выбрать правильную политику (policy)?

Какие факторы влияют на скорость и качество обучения RL?

Чем отличается модельный подход от бесмодельного подхода в RL?

Практические вопросы:

Приведите пример реализации метода epsilon-greedy для выбора действий.

Опишите процесс обучения DQN-модель с примерами на Python.

Аналитические вопросы:

- Почему агент RL часто сталкивается с проблемой стабильности обучения?
- Объясните принцип использования опыта replay buffer и его роль в улучшении обучения.
- Какой показатель является основным критерием успеха агента RL?

Критерии оценки:

- **Зачтено:** Полное выполнение всех шагов, демонстрация качественного понимания поведения модели, адекватный анализ полученного результата.
- **Незачтено:** Ключевые этапы выполнения пропущены или содержат критические ошибки.

3) Подготовка датасетов:

- Использование открытых данных Kaggle, paperswithcode и Hugging Face.
- Генерация синтетических данных при необходимости.

Задача №2: доступ к необходимым вычислительным ресурсам (в п.4.3 РПД)
Задача №3: Организация проверки и обратной связи по выполненным работам.

Порядок проверки корректности

Чек-лист для проверки лабораторных работ:

1. Выполнение заданий:
   Код запускается без ошибок.
   Достигнуто целевое качество модели.
2. Качество кода:
   Соблюдение PEP-8.

Наличие комментариев.

3. Отчет:

   инструкция по работе с гитом с подробным описанием именования методов и коммитов;

   Описание хода работы.

   Анализ результатов.

4. Своевременность:

   Работа сдана в установленный срок.

Критерии оценки:

Зачтено: Полное выполнение всех заданий, качественный код и отчет.

Незачтено: Критические ошибки или невыполнение работы.

4.5. Методические указания по организации проектной деятельности студентов

Условия применения:

Курс рассчитан на студентов 4-го года обучения,

Общее время на проект – не более 12 часов на каждого студента.

Имеется доступ к кейсам индустриальных партнеров.

Цели, задачи и ожидаемые результаты

Цели организации вычислительной инфраструктуры:

дать начальное представление о реальных задачах, решаемых с помощью методов обучения с подкреплением и возникающих проблемах.

Задачи преподавателя:

- сбор кейсов индустриальных партнеров;
- сбор кейсов преподавателей практиков и лабораторий в вузе;
- формирование ТЗ на зачетный проект на основе кейсов;
- разработка системы учёта результатов проекта в итоговой оценке зачета

Ожидаемые результаты студентов:

начальное представление о реальных задачах, решаемых с помощью методов обучения с подкреплением и возникающих проблемах.

Порядок реализации

Задача №1: сбор кейсов индустриальных партнеров

1. Оптимизация кредитного скоринга с использованием обучения с подкреплением

Описание: Сбербанк активно развивает кредитные продукты и стремится улучшить процесс оценки кредитоспособности клиентов. Задача – разработать модель, которая будет анализировать широкий спектр данных (финансовые показатели, поведенческие данные, социальные сети и т.д.) для более точной оценки риска.

Цель: Создать модель, которая сможет предсказывать вероятность дефолта с высокой точностью, используя различные типы данных.

Ожидаемый результат: Нейронная сеть, способная обрабатывать разнородные данные и выдавать точные прогнозы по кредитоспособности клиентов.

2. Мультимодальный агент для анализа строительных площадок

Описание: ООО «АВА ЛАБ» разрабатывает систему для мониторинга строительных объектов. Требуется создать прототип мультимодального ИИ-агента, способного анализировать изображения со стройплощадки (видео/фото), а также принимать голосовые и текстовые запросы (например, «проверь монтаж перекрытия на 5 этаже»).

Цель: Объединить возможности компьютерного зрения (распознавание стадии строительства, техники, нарушений) и НЛП (понимание запросов, отчётов).

Ожидаемый результат: Интерактивный агент, который на запрос специалиста может показать нужный участок, прокомментировать прогресс, зафиксировать нарушения.

Задача № 2: кейсов преподавателей практиков и лабораторий в вузе.

1. Простейший агент для игры GridWorld

Постановка задачи: Разработать простого агента, который учится перемещаться по лабиринту GridWorld, избегая препятствий и достигая цели за минимальное число шагов. Задача состоит в том, чтобы научить агента находить оптимальный путь от начальной точки до выхода, следуя правилам игры.

Требования:

– Создание среды GridWorld: реализовать лабиринт с фиксированными размерами, препятствиями и наградами (за попадание в цель +10 очков, штраф за каждый ход −1 очко, удар о препятствие −10 очков).

– Алгоритм обучения: разработать агента, использующего случайное поведение или простое правило выбора действий (например, єє-greedy). Агент должен учиться избегать стены и быстрее доходить до выхода.

– Экспериментальная проверка: провести серию экспериментов с варьируемыми параметрами обучения (темпы обучения (αα), коэффициент жадности (єє) и размеры сетки лабиринта). Необходимо оценить влияние параметров на скорость обучения и стабильность найденного маршрута.

2. Value Iteration и Policy Iteration в среде Taxi-v3

Постановка задачи: Реализовать агента-таксиста, задача которого заключается в эффективной доставке пассажира из одной точки города в другую. Важно минимизировать общее количество шагов, необходимых для перевозки, и избежать штрафа за неправильные перемещения.

Требования:

– Использование среды Taxi-v3: создать агента на основе стандартной среды OpenAI Gym Taxi-v3. Эта среда представляет собой симуляцию такси, которое должно доставлять пассажиров, соблюдая правила движения.

– Применение Value Iteration и Policy Iteration: сравнить два фундаментальных подхода в динамическом программировании (Value Iteration и Policy Iteration). Оба метода направлены на формирование оптимального пути для достижения целей с минимальным количеством ошибок.

– Анализ результатов: исследовать различия в скорости сходимости и качестве найденных решений между двумя методами. Определить наиболее подходящий подход в зависимости от размера пространства состояний и уровня сложности задачи.

Задача №3: формирование ТЗ на зачетный проект на основе кейсов

1. Inverse Reinforcement Learning (IRL).

Решите задачу обучения с помощью примеров эксперта (expert demonstration). Воспользуйтесь методом Maximum Entropy Inverse Reinforcement Learning (MaxEnt IRL) и воссоздайте политику эксперта на небольшом примере (GridWorld или аналогичном).

Проект выполняется в командах от 1 до 3 человек. Оценивается вся команда одной оценкой.

Индивидуальное задание состоит в анализе качества разработанных моделей. Для выполнения задания необходимо выполнить несколько задач:

- подготовить набор данных для проведения тестирования;
- протестировать разработанные модели,
- посчитать метрики
- сделать выводы о качестве модели

В зависимости от качества теоретического ответа и количества реализованного самостоятельно кода преподаватель выставляет зачтено или незачтено.

Критерии оценки:

Зачтено: Полное выполнение всех шагов, анализ полученных результатов с построением функции потерь и необходимых метрик, ответил практически на все теоретические вопросы.

Незачтено: Невыполнение ключевых этапов. Не ответил на теоретические вопросы.

Оценку можно повысить, реализовав требуемый функционал или ответив дополнительно или заново на необходимые вопросы.

Требования для повышения оценки и итоговую оценку формирует преподаватель.

Задача №4: разработка системы учёта результатов проекта в итоговой отметке за зачет

Выполнено в РПД, п 4.2

Порядок проверки корректности

Чек-лист для проверки лабораторных работ:

Набор кейсов индустриальных партнеров – 13 шт;

Набор кейсов преподавателей практиков и лабораторий ВУЗа – 7 шт;

Набор ТЗ в количестве 15 штук.

5. Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины (модуля)

5.1 Основная литература:

1. Бессмертный, И. А. Искусственный интеллект. Введение в многоагентные системы : учебник для вузов / И. А. Бессмертный. — Москва : Издательство Юрайт, 2025. — 148 с. — (Высшее образование). — ISBN 978-5-534-20348-6. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: https://urait.ru/bcode/569279 (дата обращения: 12.08.2025).
2. Хливненко, Л. В. Практика нейросетевого моделирования : учебное пособие для вузов / Л. В. Хливненко, Ф. А. Пятакович. — 4-е изд., стер. — Санкт-Петербург : Лань, 2024. — 200 с. — ISBN 978-5-507-47590-2. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/393482 (дата обращения: 21.07.2025). — Режим доступа: для авториз. пользователей.
3. Митяков, Е. С. Искусственный интеллект и машинное обучение : учебное пособие для вузов / Е. С. Митяков, А. Г. Шмелева, А. И. Ладынин. — Санкт-Петербург : Лань, 2025. — 252 с. — ISBN 978-5-507-51465-6. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/450827 (дата обращения: 21.07.2025). — Режим доступа: для авториз. пользователей.
4. Ростовцев, В. С. Искусственные нейронные сети : учебник для вузов / В. С. Ростовцев. — 5-е изд., стер. — Санкт-Петербург : Лань, 2025. — 216 с. — ISBN 978-5-507-50568-5. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/447392 (дата обращения: 21.07.2025). — Режим доступа: для авториз. пользователей.
5. Платонов, А. В. Машинное обучение : учебное пособие для вузов / А. В. Платонов. — Москва : Издательство Юрайт, 2022. — 85 с. — (Высшее образование). — ISBN 978-5-534-15561-7. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: https://urait.ru/bcode/508804 (дата обращения: 19.07.2025).

5.2 Дополнительная литература:

1. Sun, X., Li, J., Kovalenko, A.V., Feng, W., Ou, Y. Integrating Reinforcement Learning and Learning From Demonstrations to Learn Nonprehensile Manipulation //IEEE Transactions on Automation Science and Engineering, 2023, 20(3), 1735–1744, DOI: 10.1109/TASE.2022.3185071, Q1
2. Petukhova, A.V.; Kovalenko, A.V.; Ovsyannikova, A.V. Algorithm for Optimization of Inverse Problem Modeling in Fuzzy Cognitive Maps. Mathematics 2022, 10, 3452. DOI: 10.3390/math10193452, Q1

3. Kirillova, E.; Kovalenko, A.; Urtenov, M. Study of the Current–Voltage Characteristics of Membrane Systems Using Neural Networks. AppliedMath 2025, 5, 10. https://doi.org/10.3390/appliedmath5010010.
4. Kadurin, Artur, et al. "The cornucopia of meaningful leads: Applying deep adversarial autoencoders for new molecule development in oncology." Oncotarget 8.7 (2016): 10883.
5. Kadurin, Artur, et al. "druGAN: an advanced generative adversarial autoencoder model for de novo generation of new molecules with desired molecular properties in silico." Molecular pharmaceutics 14.9 (2017): 3098-3104.
6. Polykovskiy, Daniil, et al. "Molecular sets (MOSES): a benchmarking platform for molecular generation models." Frontiers in pharmacology 11 (2020): 565644.
7. Khrabrov, Kuzma, et al. "$\nabla^ 2$ DFT: A Universal Quantum Chemistry Dataset of Drug-Like Molecules and a Benchmark for Neural Network Potentials." Advances in Neural Information Processing Systems 37 (2024): 36869-36889.
8. Polykovskiy, Daniil, et al. "Entangled conditional adversarial autoencoder for de novo drug discovery." Molecular pharmaceutics 15.10 (2018): 4398-4405.
9. Николенко, Сергей, Кадурин, Артур и Архангельская Екатерина. Глубокое обучение. " Издательский дом"" Питер""", 2017.
10. Рабчевский, А. Н. Синтетические данные и развитие нейросетевых технологий : учебное пособие для вузов / А. Н. Рабчевский. — Москва : Издательство Юрайт, 2024. — 187 с. — (Высшее образование). — ISBN 978-5-534-17716-9. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: https://urait.ru/bcode/545036 (дата обращения: 19.07.2025).
11. Елисеев А. И., Минин Ю. В. Разработка программных интерфейсов веб-приложений с использованием фреймворка FastAPI : учебное пособие. Тамбов: ТГТУ, 2024. 81 с. https://e.lanbook.com/book/472310 (дата обращения: 19.07.2025).
12. Чернышев, С. А. Основы программирования на Python : учебное пособие для вузов / С. А. Чернышев. — Москва : Издательство Юрайт, 2022. — 286 с. — (Высшее образование). — ISBN 978-5-534-14350-8. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: https://urait.ru/bcode/496893. (дата обращения: 19.07.2025).
13. Златопольский Д. М. Основы программирования на языке Python. 2-е изд. Москва: ДМК Пресс, 2018.

5.3. Периодические издания:

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

Конференции A*:

1. https://openreview.net/forum?id=FMMF1a9ifL
2. https://openreview.net/forum?id=ElUrNM9U8c#discussion
3. https://openreview.net/forum?id=JoO6mtCLHD
4. https://aclanthology.org/2024.findings-emnlp.760/
5. https://aclanthology.org/2020.coling-main.588/
6. https://link.springer.com/chapter/10.1007/978-3-030-72113-8_30
7. https://link.springer.com/chapter/10.1007/978-3-031-42448-9_10
8. https://aclanthology.org/2024.findings-naacl.288/

5.4. 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. Официальная документация по TensorFlow https://www.tensorflow.org/?hl=ru
2. Официальная документация по pyTorch https://docs.pytorch.org/docs/stable/index.html
3. КиберЛенинка http://cyberleninka.ru/;
4. Американская патентная база данных http://www.uspto.gov/patft/
5. Министерство науки и высшего образования Российской Федерации https://www.minobrnauki.gov.ru/;
6. Федеральный портал "Российское образование" http://www.edu.ru/;
7. Информационная система "Единое окно доступа к образовательным ресурсам" http://window.edu.ru/;
8. Единая коллекция цифровых образовательных ресурсов http://school-collection.edu.ru/ .
9. Проект Государственного института русского языка имени А.С. Пушкина "Образование на русском" https://pushkininstitute.ru/;
10. Справочно-информационный портал "Русский язык" http://gramota.ru/;
11. Служба тематических толковых словарей http://www.glossary.ru/;
12. Словари и энциклопедии http://dic.academic.ru/;
13. Образовательный портал "Учеба" http://www.ucheba.com/;
14. Законопроект "Об образовании в Российской Федерации". Вопросы и ответы http://xn--273--84d1f.xn--p1ai/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/

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

По дисциплине «Методы обучения с подкреплением» предусмотрено проведение лекционных занятий, на которых даётся систематизированное представление о методах обучения с подкреплением, их классификации и трендах. В ходе лекций студенты знакомятся с основными концепциями и методами обучения с подкреплением (Reinforcement Learning, RL). Основное внимание уделяется пониманию принципов взаимодействия агента с окружающей средой, формирования оптимальной политики поведения, а также методов, позволяющих находить эффективные стратегии действий в динамических ситуациях. Изучаются классические алгоритмы обучения с подкреплением, такие как Q-Learning, SARSA, а также их модификации и расширенные версии (DQN, A2C, A3C, TRPO, PPO). Особое внимание уделяется следующим важным вопросам: различиям между greedy-, є-greedy- и softmax-политиками, возможностям прямого обучения по поведению экспертов (imitation learning) и обратной индукции политики (inverse reinforcement learning), особенностям работы с нестационарными и частично наблюдаемыми средами. Демонстрируются актуальные приложения RL в различных областях, включая игровые среды (AlphaGo, Atari games), автономные транспортные системы, робототехнику и управление экономическими системами.

Лабораторные занятия направлены на развитие практических навыков работы с современными инструментами и моделями RL. Студенты выполняют проекты, включающие создание собственных агентов для обучения в среде OpenAI Gym, применение популярных алгоритмов (DQN, A2C, PPO) для задач, связанных с играми, управлением транспортными средствами и работой с робототехникой. Помимо этого, осваиваются техники обработки сигналов и данных для задач синтеза текста и распознавания речи.

Самостоятельная работа предполагает изучение дополнительной литературы, статей и документации библиотек, чтобы глубже разобраться в теориях и новейших практиках RL. Среди обязательных навыков — умение подбирать подходящий алгоритм RL для конкретной задачи, понимать нюансы их настройки и обучения, а также интерпретировать результаты и выбирать подходящие критерии оценки качества работы моделей.

Важнейший этап обучения — самостоятельная проектная работа, в рамках которой студенты самостоятельно конструируют систему на основе выбранных методик RL. Такие проекты могут включать, например, построение интеллектуального агента для настольной игры, разработку автопилотируемых систем для автомобилей или роботов, проектирование систем финансового анализа или поддержки принятия решений. Этот вид деятельности позволяет интегрировать изученную теорию и приобрести уверенные навыки практической реализации решений в области обучения с подкреплением.

Для студентов с ограниченными возможностями здоровья предусмотрены индивидуальные консультации и адаптированные материалы. Преподаватель помогает осваивать интерфейсы взаимодействия с ИИ, объясняет ключевые понятия в доступной

форме, предоставляет инструкции с альтернативным форматированием. При необходимости используются голосовые интерфейсы, увеличенный масштаб экрана, сопровождение при выполнении заданий. Индивидуальный подход обеспечивает равные условия участия в образовательном процессе и достижения запланированных результатов обучения.

Кейсы ПАО «Сбербанк»

1. **Мультимодальный ассистент для банковских отделений**

Описание: Физические отделения Сбербанка внедряют интерактивных консультантов. Предполагается создание мультимодального ИИ-ассистента, который воспринимает речь и визуально ориентируется в пространстве (распознаёт клиента, документы, банкоматы), а также отвечает голосом.

Цель: Разработать базовый прототип, имитирующий функциональность помощника: ответы на типовые запросы, визуальные подсказки, навигация по отделению.

Ожидаемый результат: Интерактивная модель, объединяющая голосовой ввод, зрительное восприятие (например, QR-код паспорта), текстовый вывод и жестовую реакцию.

2. **Анализ поведения пользователей в экосистеме цифрового рубля**

Описание: Сбербанк участвует в пилотных проектах по внедрению цифрового рубля. Интерес представляет исследование пользовательских паттернов: как изменяются модели потребления, скорости операций, уровень доверия, сравнение с классическим безналом.

Цель: Построить модель анализа поведения клиентов, участвующих в транзакциях с цифровым рублем: частота, средний чек, контексты.

Ожидаемый результат: Отчёт и ML-модель, классифицирующая типы пользователей и выявляющая ключевые различия в предпочтениях и барьерах цифровой валюты.

3. **Оптимизация кредитного скоринга с использованием обучения с подкреплением**

Описание: Сбербанк активно развивает кредитные продукты и стремится улучшить процесс оценки кредитоспособности клиентов. Задача – разработать модель, которая будет анализировать широкий спектр данных (финансовые показатели, поведенческие данные, социальные сети и т.д.) для более точной оценки риска.

Цель: Создать модель, которая сможет предсказывать вероятность дефолта с высокой точностью, используя различные типы данных.

Ожидаемый результат: Нейронная сеть, способная обрабатывать разнородные данные и выдавать точные прогнозы по кредитоспособности клиентов.

4. **Прогнозирование оттока клиентов с использованием обучения с подкреплением**

Описание: Сбербанк стремится минимизировать отток клиентов, предлагая им персонализированные услуги и продукты. Задача – разработать модель, которая будет предсказывать вероятность ухода клиента на основе его поведения и истории взаимодействия с банком.

Цель: Создать модель, которая сможет предсказывать отток клиентов за несколько месяцев до фактического ухода, что позволит банку предпринять превентивные меры.

Ожидаемый результат: Модель, способная анализировать временные ряды данных и предсказывать отток клиентов с высокой точностью.

5. **Оптимизация работы чат-бота с использованием обучения с подкреплением**

Описание: Сбербанк использует чат-боты для обслуживания клиентов и обработки запросов. Задача – разработать модель, которая будет оптимизировать работу чат-бота, улучшая его способность понимать запросы клиентов и давать точные ответы.

Цель: Создать модель, которая сможет обучаться на основе обратной связи от клиентов и улучшать свои ответы с течением времени.

Ожидаемый результат: Чат-бот, способный давать более точные и релевантные ответы, что повысит удовлетворенность клиентов.

6. **Оптимизация маршрутов доставки с использованием обучения с подкреплением**

Описание: Сбербанк предоставляет услуги по доставке документов и других материалов. Задача – разработать модель, которая будет оптимизировать маршруты доставки, учитывая различные факторы, такие как время, расстояние и загруженность дорог.

Цель: Создать модель, которая сможет находить оптимальные маршруты для доставки, что сократит время и затраты на логистику.

Ожидаемый результат: Система, способная автоматически оптимизировать маршруты доставки, что повысит эффективность логистики.

7. **Оптимизация работы колл-центра с использованием обучения с подкреплением**

Описание: Сбербанк имеет большой колл-центр, который обрабатывает множество звонков от клиентов. Задача – разработать модель, которая будет оптимизировать работу колл-центра, улучшая распределение звонков и сокращая время ожидания клиентов.

Цель: Создать модель, которая сможет обучаться на основе данных о звонках и улучшать распределение звонков между операторами.

Ожидаемый результат: Система, способная оптимизировать работу колл-центра, что повысит удовлетворенность клиентов и снизит нагрузку на операторов.

Кейсы от «АВАЛАБ»

1. **Мультимодальный агент для анализа строительных площадок**

Описание: ООО «АВА ЛАБ» разрабатывает систему для мониторинга строительных объектов. Требуется создать прототип мультимодального ИИ-агента, способного анализировать изображения со стройплощадки (видео/фото), а также принимать голосовые и текстовые запросы (например, «проверь монтаж перекрытия на 5 этаже»).

Цель: Объединить возможности компьютерного зрения (распознавание стадии строительства, техники, нарушений) и НЛП (понимание запросов, отчётов).

Ожидаемый результат: Интерактивный агент, который на запрос специалиста может показать нужный участок, прокомментировать прогресс, зафиксировать нарушения.

2. **Модель прогнозирования сроков сдачи объектов на основе текстовых и визуальных данных**

Описание: Девелоперская компания ведёт аналитический архив по срокам строительства. С помощью мультимодальных моделей (текстовые отчёты + фото стройки) можно прогнозировать вероятность отклонения от графика сдачи.

Цель: Разработать модель, которая по текущему статусу объекта (фото, отчёт СМР) оценивает риски задержек.

Ожидаемый результат: Прототип, который показывает вероятность отклонений и даёт текстовые пояснения (основанные на распознанных признаках — «не завершены фасадные работы», «монтаж инженерии не начат»).

3. Оптимизация работы с клиентами в CRM-системе

Описание: Ава Лаб разрабатывает CRM-систему для управления взаимоотношениями с клиентами. Задача – разработать модель, которая будет оптимизировать работу с клиентами, улучшая их опыт и удовлетворенность.

Цель: Создать модель, которая сможет обучаться на основе данных о поведении клиентов в CRM-системе и улучшать их взаимодействие с компанией.

Ожидаемый результат: Система, способная оптимизировать работу с клиентами в CRM-системе, что повысит их удовлетворенность и лояльность.

4. Оптимизация планирования строительных проектов

Описание: Строительная компания сталкивается с необходимостью эффективного планирования строительных проектов, учитывая множество факторов, таких как сроки, ресурсы и бюджет. Задача — разработать модель, которая будет оптимизировать планирование проектов, улучшая их выполнение и сокращая затраты.

Цель: Создать модель, которая сможет обучаться на основе данных о предыдущих проектах и улучшать планирование будущих проектов.

Ожидаемый результат: Система, способная оптимизировать планирование строительных проектов, что повысит эффективность работы компании и снизит затраты.

5. Оптимизация управления ресурсами на стройплощадке

Описание: Строительная компания управляет множеством ресурсов на стройплощадке, включая материалы, оборудование и персонал. Задача — разработать модель, которая будет оптимизировать управление ресурсами, улучшая их использование и сокращая простои.

Цель: Создать модель, которая сможет обучаться на основе данных о ресурсах и улучшать их распределение и использование.

Ожидаемый результат: Система, способная оптимизировать управление ресурсами на стройплощадке, что повысит эффективность работы компании и снизит затраты.

6. Оптимизация работы с клиентами в системе управления проектами

Описание: Строительная компания использует систему управления проектами для координации работы команды и выполнения задач. Задача — разработать модель, которая будет оптимизировать работу с клиентами в системе управления проектами, улучшая их опыт и удовлетворенность.

Цель: Создать модель, которая сможет обучаться на основе данных о поведении клиентов в системе управления проектами и улучшать их взаимодействие с компанией.

Ожидаемый результат: Система, способная оптимизировать работу с клиентами в системе управления проектами, что повысит их удовлетворенность и лояльность.

7. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине (модулю)

7.1 Перечень информационно-коммуникационных технологий

1. Облачные платформы и сервисы

Google Colab – облачная среда для выполнения кода на Python с GPU/TPU
cloud.ru,YandexCloud,AWS/GCP/Azure– облачные вычисления
Kaggle – платформа для работы с датасетами и соревнований по ML
Hugging Face Spaces – развертывание CV-моделей в виде демо

2. Системы управления версиями и коллаборации

Git/GitHub/GitLab – контроль версий кода и совместная разработка
Notion/Trello – организация проектной деятельности

3. Инструменты для работы с данными

DVC (Data Version Control) – управление версиями данных
Apache Spark – обработка больших текстовых корпусов

4. Система управления обучением

Moodle – сдача работ

7.2 Перечень лицензионного и свободно распространяемого программного обеспечения

1. Лицензионное ПО

VSCode – IDE для Python (свободнораспространяемое)

LibreOffice– оформление отчетов (свободнораспространяемое)

2. Свободное ПО (Open Source)
Hugging Face Optimum – оптимизация и развёртывание моделей

Среды для RL:

PyTorch-RL/TF-agents/ OpenAi Gym – обучение с подкреплением

Инструменты для визуализации:

Streamlit/Gradio – создание веб-интерфейсов для моделей
Matplotlib/Seaborn – графики и анализ данных

СУБД:

SQLite/PostgreSQL – хранение структурированных данных
FAISS/Annoy – векторный поиск

8. Материально-техническое обеспечение по дисциплине (модулю)

Виртуальные машины 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
		Аренда публичного IP	1		Шт
2	Виртуальная машина с GPU	Виртуальная машина с GPU NVIDIA® Tesla® V100 2 GPU 8 vCPU 128 ГБ RAM	1	1	Шт
		ОС Ubuntu_24.04	1		Шт
		Системный диск SSD	1		Шт
			2000		Гб
		Диск SSD	1		Шт
			4096		Гб
		Диск SSD	1		Шт
			4096		Гб
		Аренда публичного IP	1		Шт
3	K8S	Master node 8 vCPU 16 RAM	1	1	Шт
		Worker node 10% доля 4 vCPU 32 RAM	5		Шт
		Worker node SSD-NVME	64		Гб

		Аренда публичного IP	1		Шт
4	ML Inference Instance Type GPU	Время работы в месяц	40	1	Ч
		Инстанс 8 x NVIDIA® H100 NVLink PCIe 160 vCPU 1520 GB RAM	1		Шт
		Количество запросов к ML-моделям	1		Млн. Шт
		Кэш ML-моделей	160		Гб
5	LLM	Токены GigaChat 2 Max	50		Млн. Шт
		Токены Embeddings	400		Млн. Шт

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.	Практические занятия	Аудитория, укомплектованная специализированной мебелью и техническими средствами обучения
4.	Групповые (индивидуальные) консультации	Аудитория, укомплектованная специализированной мебелью и техническими средствами обучения, компьютерами, программным обеспечением
5.	Текущий контроль, промежуточная аттестация	Аудитория, укомплектованная специализированной мебелью и техническими средствами обучения, компьютерами, программным обеспечением
6.	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.