

**Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего
образования
«ИРКУТСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ТЕХНИЧЕСКИЙ
УНИВЕРСИТЕТ»**

Структурное подразделение «Брикс кафедры»

УТВЕРЖДЕНА:
на заседании кафедры
Протокол №15 от 18 марта 2025 г.

Рабочая программа дисциплины

«МАШИННОЕ ТВОРЧЕСТВО / COMPUTATIONAL CREATIVITY»

Направление: 09.03.01 Информатика и вычислительная техника

Искусственный интеллект и компьютерные науки /Artificial Intelligence and Computer
Science

Квалификация: Бакалавр

Форма обучения: очная

Документ подписан простой
электронной подписью
Составитель программы:
Чжан Цзюньвэй
Дата подписания: 16.06.2025

Документ подписан простой
электронной подписью
Утвердил: Киреенко Анна
Павловна
Дата подписания: 17.06.2025

Документ подписан простой
электронной подписью
Согласовал: Афанасьев
Александр Диомидович
Дата подписания: 17.06.2025

Год набора – 2025

Иркутск, 2025 г.

1 Перечень планируемых результатов обучения по дисциплине, соотнесённых с планируемыми результатами освоения образовательной программы

1.1 Дисциплина «Машинное творчество / Computational Creativity» обеспечивает формирование следующих компетенций с учётом индикаторов их достижения

Код, наименование компетенции	Код индикатора компетенции
ПКС-2 Способен приобретать систематические знания в области искусственного интеллекта, понимая специфику научного знания, основанного на научно-обоснованных фактах	ПКС-2.4
ПКС-3 Способность разрабатывать программные продукты, используя современные технологии в области искусственного интеллекта	ПКС-3.12

1.2 В результате освоения дисциплины у обучающихся должны быть сформированы

Код индикатора	Содержание индикатора	Результат обучения
ПКС-2.4	Знает специфику научного знания, его отличия от художественного, ненаучного и ложного	Знать Современный уровень и направления развития моделей, методов и программного обеспечения генеративных систем. Уметь Работать с различными моделями представления знаний и обосновывать выбор той или иной модели в зависимости от характера предметной области. Владеть Способностью применять методы искусственного интеллекта для решения задач профессиональной деятельности, включая выполнение мини-проекта по генеративным ИИ-системам.
ПКС-3.12	Применяет лучшие практики глубокого обучения в области искусственного интеллекта для решения задач профессиональной деятельности	Знать Основные принципы глубокого обучения и алгоритмы обучения генеративно-сопоставительных нейронных сетей (GAN), включая архитектуры VAE, Diffusion. Уметь Проектировать архитектуру генеративных моделей (GAN, VAE и др.), обучать их на примерах, адаптировать к творческим задачам и интерпретировать результаты. Владеть Навыками применения современных инструментов (PyTorch, TensorFlow, Google Colab и др.) для реализации проектов по генеративному ИИ и создания креативных интеллектуальных

		систем.
--	--	---------

2 Место дисциплины в структуре ООП

Изучение дисциплины «Машинное творчество / Computational Creativity» базируется на результатах освоения следующих дисциплин/практик: «Программирование / Programming», «Большие данные, методы обработки / Big Data, Processing Methods», «Дискретная математика / Discrete Mathematics», «Математика / Mathematics», «Введение в специальность / Introduction to the specialty», «Теория вероятностей, математическая статистика и случайные процессы / Probability Theory, Mathematical Statistics, and Random Processes», «Глубокое обучение в технологиях обработки естественного языка (NLP) / Deep Learning in Natural Language Processing Technologies (NLP)», «Глубокое обучение в технологиях компьютерного зрения / Deep Learning in Computer Vision Technology», «Алгоритмы ML / Machine Learning Algorithms», «Теоретические основы искусственного интеллекта / Theoretical Foundations of Artificial Intelligence»

Дисциплина является предшествующей для дисциплин/практик: «Производственная практика: преддипломная практика / Manufacturing Practice: Undergraduate Practice», «Машинное творчество / Computational Creativity»

3 Объем дисциплины

Объем дисциплины составляет – 4 ЗЕТ

Вид учебной работы	Трудоемкость в академических часах (Один академический час соответствует 45 минутам астрономического часа)	
	Всего	Семестр № 7
Общая трудоемкость дисциплины	144	144
Аудиторные занятия, в том числе:	60	60
лекции	30	30
лабораторные работы	0	0
практические/семинарские занятия	30	30
Самостоятельная работа (в т.ч. курсовое проектирование)	57	57
Трудоемкость промежуточной аттестации	27	27
Вид промежуточной аттестации (итогового контроля по дисциплине)	Экзамен, Курсовой проект	Экзамен, Курсовой проект

4 Структура и содержание дисциплины

4.1 Сводные данные по содержанию дисциплины

Семестр № 7

№ п/п	Наименование раздела и темы дисциплины	Виды контактной работы						СРС		Форма текущего контроля
		Лекции		ЛР		ПЗ(СЕМ)		№	Кол. Час.	
		№	Кол. Час.	№	Кол. Час.	№	Кол. Час.			
1	2	3	4	5	6	7	8	9	10	11
1	Введение в генеративные системы и модели	1	2			1	2	1, 5	2	Тест

	искусственного интеллекта									
2	Обучение нейронных сетей и платформы для генеративного ИИ	2, 3	6			2, 3	6	1, 2, 4, 5	10	Тест, Письменная работа
3	Генерация текста и семантические модели	4	2			4	2	1, 4, 5	6	Тест, Проект, Творческое задание
4	Создание изображений и перенос стиля	5, 6, 7, 8, 9	12			5, 6, 7, 8	10	1, 2, 4, 5	15	Тест, Творческое задание, Проект
5	Эволюционные и креативные алгоритмы	10	2			9	2	1, 4, 5	4	Тест, Проект, Творческое задание
6	Генеративные состязательные сети (GAN)	11, 12	6			10, 11	8	1, 2, 3, 4, 5	20	Творческое задание, Проект, Тест, Отчет
	Промежуточная аттестация								27	Экзамен, Курсовой проект
	Всего		30				30		84	

4.2 Краткое содержание разделов и тем занятий

Семестр № 7

№	Тема	Краткое содержание
1	Введение в генеративные системы и модели искусственного интеллекта	Основные понятия и история ИИ. Эвристика, правила, факты. Особое внимание — генеративным ИИ.
2	Обучение нейронных сетей и платформы для генеративного ИИ	Повторение основ нейросетей. Сравнение фреймворков (TensorFlow, PyTorch). Примеры генеративных моделей на практике.
3	Генерация текста и семантические модели	Роль семантических сетей, работа с LLM API. Текстовая генерация с помощью RNN, LSTM, GPT.
4	Создание изображений и перенос стиля	Принципы работы DeepDream, нейронного переноса стиля (Style Transfer), VAE, Diffusion, GAN. Визуальные эксперименты.
5	Эволюционные и креативные алгоритмы	Генетические алгоритмы, эволюционное программирование, применение к задачам креативности.
6	Генеративные состязательные сети (GAN)	Структура и принципы работы GAN. Примеры: CycleGAN, StyleGAN. Генерация изображений и аудио.

4.3 Перечень лабораторных работ

Лабораторных работ не предусмотрено

4.4 Перечень практических занятий

Семестр № 7

№	Темы практических (семинарских) занятий	Кол-во академических часов
1	Реализация нейросетей на Python	2
2	Обучение нейросетей вручную в Excel	4
3	Тестирование Colab + импорт моделей PyTorch	2
4	Использование LLM API (например, ChatGPT/OpenAI) для генерации текстов	2
5	Генерация изображений с помощью DeepDream и VGG19	4
6	Нейронный перенос стиля: художественные эксперименты	2
7	Работа с VAE: латентное кодирование и визуализация	2
8	Запуск модели Stable Diffusion в Google Colab	2
9	Эволюция изображений с помощью генетического алгоритма	2
10	Тренировка простого GAN для генерации чисел MNIST	2
11	Проект по CycleGAN — преобразование изображений	6

4.5 Самостоятельная работа

Семестр № 7

№	Вид СРС	Кол-во академических часов
1	Итоговый тест	6
2	Написание курсового проекта (работы)	12
3	Написание отчета	6
4	Подготовка к практическим занятиям (лабораторным работам)	27
5	Подготовка к экзамену	6

В ходе проведения занятий по дисциплине используются следующие интерактивные методы обучения: Лекция-дискуссия, метод проектов, групповая работа

5 Перечень учебно-методического обеспечения дисциплины

5.1 Методические указания для обучающихся по освоению дисциплины

5.1.1 Методические указания для обучающихся по курсовому проектированию/работе:

Курсовой проект посвящён практическому применению модели CycleGAN для преобразования изображений между двумя доменами.

Каждый студент выбирает индивидуальный набор данных (например, день ↔ ночь,

фото ↔ аниме и т.д.) и выполняет полный цикл проекта.

Проект включает:

- изучение архитектуры и принципов работы CycleGAN;
- освоение основных команд работы с GitHub (fork, commit, push);
- реализация и обучение модели на индивидуальном датасете (Google Colab / PyTorch);
- визуализация результатов генерации;
- оформление итогового Jupyter/Colab-ноутбука;
- написание отчёта по структуре: постановка задачи, описание данных, реализация, результаты, выводы.

Итоговые материалы (отчёт и ссылка на репозиторий) предоставляются в электронном виде через систему LMS (MOODLE) или преподавателю.

Подробные методические материалы, инструкции по запуску моделей и ссылки на Colab-ноутбуки размещены в курсе на платформе:
<https://el.istu.edu/course/view.php?id=9355>

5.1.2 Методические указания для обучающихся по практическим занятиям

Практические занятия построены на основе курса и включают следующие этапы:

- 1) изучение теоретических основ используемой модели (например, GAN, VAE, Diffusion);
- 2) практическая реализация и запуск модели с использованием предоставленных кодов (Google Colab / Jupyter);
- 3) получение и интерпретация результатов генерации;
- 4) документирование процесса и выводов.

Методические материалы включают пояснительные инструкции, ссылки на репозитории и видео-примеры. Все занятия направлены на формирование практических навыков в области генеративного ИИ.

Подробные методические материалы, инструкции по запуску моделей и ссылки на Colab-ноутбуки размещены в курсе на платформе:
<https://el.istu.edu/course/view.php?id=9355>

5.1.3 Методические указания для обучающихся по самостоятельной работе:

Самостоятельные работы направлены на закрепление практических навыков, полученных в ходе занятий.

Студенты выполняют индивидуальные задания, основанные на практических темах курса (CycleGAN, VAE, Diffusion и др.), с обязательной реализацией модели в среде Google Colab или Jupyter Notebook.

Ожидается, что обучающиеся:

- 1) самостоятельно проведут обучение модели на выбранном датасете;
- 2) сохранят полученные результаты (изображения, метрики и т.д.);
- 3) оформят итоговый ноутбук в соответствии с методическими требованиями.

Готовые ноутбуки и отчёты проверяются преподавателем и могут быть представлены к защите.

Подробные методические материалы, инструкции по запуску моделей и ссылки на Colab-ноутбуки размещены в курсе на платформе:
<https://el.istu.edu/course/view.php?id=9355>

6 Фонд оценочных средств для контроля текущей успеваемости и проведения промежуточной аттестации по дисциплине

6.1 Оценочные средства для проведения текущего контроля

6.1.1 семестр 7 | Тест

Описание процедуры.

Тест проводится в системе LMS (el.istu.edu) в онлайн-режиме.

Он включает 5~30 вопросов (множественный выбор и открытые вопросы) по темам 7.1–7.6.

Время на выполнение: 5~60 минут.

Каждому студенту предоставляется индивидуальный порядок вопросов.

Критерии оценивания.

Оценка формируется на основе количества правильных ответов:

90–100% — отлично (5)

75–89% — хорошо (4)

60–74% — удовлетворительно (3)

менее 60% — неудовлетворительно (2, не зачтено)

6.1.2 семестр 7 | Письменная работа

Описание процедуры.

Письменная работа проводится в формате практического теста.

Студентам необходимо самостоятельно реализовать простую архитектуру нейронной сети (например, MLP или CNN) в одной из доступных платформ (PyTorch, Keras и др.) и обучить её на предоставленном или собственном датасете.

Результатом является Jupyter/Colab-ноутбук с реализованной моделью, визуализацией результатов и краткими комментариями.

Сдача осуществляется через LMS или ссылкой на GitHub.

Критерии оценивания.

— Корректность структуры модели и кода (55%)

— Способность обучить модель и получить рабочий результат (25%)

— Качество визуализации (например, loss-графики, accuracy и т.п.) (10%)

— Чистота кода и пояснения в ноутбуке (10%)

Итоговая оценка выставляется по 5-балльной шкале.

6.1.3 семестр 7 | Проект

Описание процедуры.

Проектная работа выполняется студентом индивидуально.

Каждому студенту необходимо выбрать одну из технологий, изученных в курсе (CycleGAN, Diffusion, GPT, VAE, StyleTransfer и т.д.) и реализовать мини-проект с практической частью.

Допускается использование Google Colab или Jupyter Notebook.
Результаты (ноутбук + отчёт) оформляются в виде GitHub-репозитория.
Предоставление осуществляется ссылкой в LMS или по e-mail преподавателю.
Доклад и презентация не требуются.

Критерии оценивания.

- Корректность и сложность реализованной технологии (30%)
- Полный рабочий код и обоснованные результаты (30%)
- Структура репозитория (ноутбук, README, данные и т.д.) (20%)
- Качество документации и отчёта (20%)

Общая оценка выставляется по 5-балльной шкале.

6.1.4 семестр 7 | Творческое задание

Описание процедуры.

Творческое задание выполняется индивидуально.

Студентам предлагается разработать оригинальное применение одной из генеративных моделей, изученных на курсе.

Это может быть смешение стилей (StyleMix), межмодальные эксперименты (текст → изображение, аудио → текст и т.п.), обучение нестандартных данных и др.

Результат оформляется в виде Jupyter/Colab-ноутбука с описанием идеи, реализацией и визуализацией.

Файлы размещаются в GitHub и прикрепляются ссылкой через LMS.

Критерии оценивания.

- Оригинальность идеи (30%)
- Реализация рабочей модели (30%)
- Наличие визуальных/творческих результатов (20%)
- Структура ноутбука и пояснения (20%)

Оценка выставляется по 5-балльной шкале.

6.1.5 семестр 7 | Отчет

Описание процедуры.

Отчет является итоговым индивидуальным заданием по курсу.

Каждый студент должен выбрать генеративную модель (например, CycleGAN), изучить её структуру и реализовать проект на основе самостоятельного датасета.

Результаты включают:

- реализованную модель в виде Colab-ноутбука,
- краткий отчёт в формате PDF (10~15 стр.)

Работы размещаются на GitHub / Google Drive и сдаются через LMS или по e-mail преподавателю.

Защита (доклад) не требуется.

Критерии оценивания.

- Полнота и корректность реализации модели (40%)
- Адекватность и оригинальность выбранного датасета (20%)
- Оформление репозитория (структура, читаемость кода) (20%)
- Качество отчёта (обоснование, визуализация, анализ) (20%)

Оценка выставляется по 5-балльной шкале.

6.2 Оценочные средства для проведения промежуточной аттестации

6.2.1 Критерии и средства (методы) оценивания индикаторов достижения компетенции в рамках промежуточной аттестации

Индикатор достижения компетенции	Критерии оценивания	Средства (методы) оценивания промежуточной аттестации
ПКС-2.4	Квалифицированно определена стратегия действий на основе критического анализа и системного подхода; продемонстрирована способность применить знания на практике через выполнение мини-проекта.	Устный ответ по билетам, зачет, экзамен, выполнение мини-проекта (например, реализация простого генеративного ИИ-приложения на базе открытого кода), защита курсового проекта.
ПКС-3.12	Квалифицированно определена архитектура модели и стратегия её обучения; демонстрируется умение доводить проект до рабочего прототипа с обоснованием технических решений.	Устный ответ по билетам, зачет, экзамен, выполнение практико-ориентированного проекта (например, реализация генеративной модели для создания изображений, музыки или текста), защита курсовой работы.

6.2.2 Типовые оценочные средства промежуточной аттестации

6.2.2.1 Семестр 7, Типовые оценочные средства для курсовой работы/курсового проектирования по дисциплине

6.2.2.1.1 Описание процедуры

Курсовой проект выполняется на основе существующего открытого проекта (например, CycleGAN), размещённого на платформе GitHub. Целью проекта является не изменение кода, а глубокое понимание архитектуры модели и этапов её функционирования.

Студент должен:

- Ознакомиться с документацией проекта;
- Подготовить и загрузить собственный набор данных;
- Выполнить обучение и тестирование модели;
- Проанализировать полученные результаты (например, по качеству генерации);
- Сформировать подробный отчёт (Google Colab / PDF), содержащий описание, выводы и скриншоты работы модели.

Главный акцент делается на понимание принципов работы модели и грамотную подачу результатов в отчёте, а не на программирование “с нуля”.

Пример задания:

Тема: Исследование модели CycleGAN для преобразования изображений без парных данных.

Задачи:

1. Ознакомьтесь с репозиторием <https://github.com/junyanz/pytorch-CycleGAN-and-pix2pix>;
2. Подготовьте уникальный набор изображений (например, зима ↔ лето, стиль ↔ реальность);
3. Обучите модель на своём датасете и протестируйте её работу;
4. Подготовьте отчёт, включающий: цели, методику, результаты, трудности и выводы.

6.2.2.1.2 Критерии оценивания

Отлично	Хорошо	Удовлетворительно	Неудовлетворительно
Отлично (5 / 90–100%) Студент полностью выполнил все этапы проекта: загрузка данных, обучение, тестирование, визуализация результатов; Отчёт оформлен в структуре научного отчёта (введение, цели,	Хорошо (4 / 75–89%) Проект выполнен в целом, но отчёт содержит незначительные упрощения или не полностью раскрывает понимание архитектуры; Есть результаты обучения и тестирования, но не все этапы	Удовлетворительно (3 / 60–74%) Студент запустил модель на чужих данных или с незначительным изменением; Отчёт содержит только описание этапов без анализа; Низкая оригинальность задания; Недостаточно скриншотов,	Неудовлетворительно (2 / 0–59%) Проект не выполнен, код не запускается, результаты не представлены; Отчёт отсутствует или сделан формально (без кода, без объяснений); Явное копирование чужих работ, отсутствие индивидуальной деятельности.

методика, результаты, выводы); Показано глубокое понимание архитектуры модели и её компонентов (циклы, дискриминаторы, генераторы и пр.); Оформлен в Google Colab или PDF, содержит код, графики и пояснения;	проанализированы ; Формат отчёта соблюден частично, отсутствуют некоторые выводы или комментарии.	визуализации или выводов.	
---	--	---------------------------	--

6.2.2.2 Семестр 7, Типовые оценочные средства для проведения экзамена по дисциплине

6.2.2.2.1 Описание процедуры

Экзамен проводится в письменной форме с использованием LMS-системы или Google Colab.

Студенты обязаны:

- 1) Выполнить все лабораторные работы, самостоятельные задания и домашние проекты в течение семестра;
- 2) Участвовать в промежуточных тестированиях и финальном экзамене;
- 3) Подготовить итоговый отчёт или презентацию по выбранной теме.

Оценка итоговой работы основывается на качестве выполнения проекта, оригинальности, понимании архитектуры модели и уровне самостоятельности.

Пример задания:

Разработать генеративную модель (например, на основе CycleGAN) по выбранной теме.

Провести обучение и тестирование на индивидуальном датасете.

Результаты представить в виде оформленного Colab-ноутбука и краткого аналитического отчета.

Предусмотреть демонстрацию работы модели и интерпретацию полученных результатов.

Вопросы к билету (примеры):

1. Объясните архитектуру модели VAE и роль латентного пространства.
2. Какова цель обучения модели GAN и чем отличается генератор от дискриминатора?
3. В чём заключается суть алгоритма Diffusion?
4. Опишите основные этапы процесса обучения модели StyleGAN.
5. Какие метрики можно использовать для оценки качества генерации изображений?
6. Что такое Zero-Shot и Few-Shot обучение и как они применяются в LLM?
7. Объясните, как работают механизмы внимания (Attention) в генеративных моделях.
8. В чём особенность использования Noise в генеративных моделях (например, GAN,

Diffusion)?

9. В чём разница между автоэнкодером и вариационным автоэнкодером?

10. ...

6.2.2.2.2 Критерии оценивания

Отлично	Хорошо	Удовлетворительно	Неудовлетворительно
Отлично (5, 90–100%) Студент продемонстрировал глубокое понимание темы, выполнил все лабораторные и самостоятельные работы, успешно защитил курсовой проект. Проект содержит оригинальную идею, корректную реализацию и полное обоснование. Все этапы задания (подготовка данных, обучение модели, анализ результатов) выполнены на высоком уровне. Тесты и экзамен пройдены с отличными результатами.	Хорошо (4, 75–89%) Студент выполнил все основные задания курса. Курсовой проект завершён, присутствует логичная структура и достаточный уровень реализации. Отчёт содержит незначительные недочёты, но в целом демонстрирует понимание ключевых аспектов темы. Результаты тестов и экзамена находятся в пределах 75–89%.	Удовлетворительно (3, 60–74%) Задания выполнены частично, проект содержит ошибки в реализации или обосновании. Анализ результатов ограничен, отчёт фрагментарен. Студент проявил базовое понимание темы, но имеются значительные пробелы. Тесты и экзамен сданы на уровне 60–74%.	Неудовлетворительно (2, 0–59%) Работы не представлены или выполнены с грубыми ошибками. Проект не завершён, отчёт отсутствует или не соответствует требованиям. Студент не продемонстрировал понимания изучаемого материала. Результаты итоговой аттестации ниже 60%.

7 Основная учебная литература

1. Шоле Ф. Глубокое обучение на Python / Ф. Шоле. // М. : ДМК Пресс, 2018. — 386 с.
2. Шоле Ф. Глубокое обучение на Python. 2-е изд. / Ф. Шоле. // М. : ДМК Пресс, 2021. — 504 с.

8 Дополнительная учебная литература и справочная

1. Гудфеллоу И. Глубокое обучение / И. Гудфеллоу, Й. Бенджио, А. Курвиль // пер. с англ. — М. : Диалектика, 2018. — 720 с.
2. Руттото Л., Хабер Э. Введение в глубокие генеративные модели / Л. Руттото, Э. Хабер // arXiv preprint. – 2021. – arXiv:2101.03582.

3. Фостер Д. Генеративное глубокое обучение: обучение машин рисовать, писать, сочинять и играть / Д. Фостер. // O'Reilly Media, 2019. – 352 с.

4. Далал А. Генеративный искусственный интеллект / А. Далал. // ResearchGate, 2023. – 284 с.

5. Чжан А. Погружение в глубокое обучение / А. Чжан, З. К. Липтон, М. Ли, К. Смолл. // arXiv, 2023. – 1145 с.

9 Ресурсы сети Интернет

1. <http://library.istu.edu/>

2. <https://e.lanbook.com/>

3. Онлайн курс «Нейронные сети и обработка текста» от экспертов Samsung AI Center – Режим доступа: <https://stepik.org/course/54098/syllabus>

4. Kaggle (образовательные ресурсы крупнейшего в мире сообщества в сфере искусственного интеллекта, науки о данных (data science)) – Режим доступа: <https://www.kaggle.com/learn/overview>

5. PyTorch (официальный сайт и документация по одной из самых популярных библиотек для глубокого обучения) – Режим доступа: <https://docs.pytorch.org/tutorials/>

6. Hugging Face (платформа с открытым исходным кодом и библиотеками для работы с моделями машинного обучения, особенно в области обработки естественного языка) – Режим доступа: <https://huggingface.co/learn>

7. arXiv (онлайн-архив научных публикаций по физике, математике, информатике и другим дисциплинам, включая искусственный интеллект и обработку языка) – Режим доступа: <https://arxiv.org/>

10 Профессиональные базы данных

1. <http://new.fips.ru/>

2. <http://www1.fips.ru/>

3. Datasets на Kaggle – Режим доступа: <https://www.kaggle.com/datasets>

4. Keras Datasets – Режим доступа: <https://keras.io/api/datasets/>

5. Наборы данных TensorFlow: набор готовых наборов данных – Режим доступа: <https://www.tensorflow.org/datasets>

6. PyTorch Datasets – Режим доступа: <https://docs.pytorch.org/vision/stable/datasets.html>

7. Caffe Datasets – Режим доступа: <https://caffe2.ai/docs/datasets.html>

11 Перечень информационных технологий, лицензионных и свободно распространяемых специализированных программных средств, информационных справочных систем

1. Свободно распространяемое программное обеспечение Официальный сайт Python <http://www.python.org/>

2. Свободно распространяемое программное обеспечение Anaconda — дистрибутив языков программирования Python и R, включающий набор популярных свободных библиотек, объединённых проблематиками науки о данных и машинного обучения. <https://www.anaconda.com/>

3. Свободно распространяемое программное обеспечение Краткое руководство по работе с сервисом Google Colaboratory <https://colab.research.google.com/notebooks/welcome.ipynb#scrollTo=-Rh3-Vt9Nev9>

4. Свободно распространяемое программное обеспечение Официальный сайт Keras
<https://keras.io/>
5. Свободно распространяемое программное обеспечение Официальный сайт TensorFlow
<https://www.tensorflow.org/>
6. Свободно распространяемое программное обеспечение Официальный сайт PyTorch
<https://pytorch.org/>

12 Материально-техническое обеспечение дисциплины

1. Компьютерный класс мультимедиа-проектором
2. Стабильный доступ в Internet для работы с облачными технологиями
3. Наличие браузера Google Chrome на компьютерах студентов и преподавателя
4. Установка пакета Anaconda на компьютеры студентов и преподавателя