

Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Лельчицкий Игорь Давыдович
Должность: и.о. проректора по образовательной деятельности
Дата подписания: 16.06.2026 14:06:11
Уникальный программный ключ: aa5b5ee17d97a2e4d94e98e995320af94f043ce2

Министерство науки и высшего образования Российской Федерации

ФГБОУ ВО «Тверской государственный университет»

Утверждаю:
Руководитель ООП


/ С.М. Дудаков /
16.06.2026 года



Тверской государственный университет
ФАКУЛЬТЕТ ПРИКЛАДНОЙ МАТЕМАТИКИ И КИБЕРНЕТИКИ

Рабочая программа дисциплины (с аннотацией)

МЕТОДЫ МАШИННОГО ОБУЧЕНИЯ

Направление подготовки
02.03.02 ФУНДАМЕНТАЛЬНАЯ ИНФОРМАТИКА И
ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

Профиль подготовки
Программная инженерия в искусственном интеллекте

Для студентов 3-го курса

Очная форма

Составитель: к.ф.-м.н., доцент И.С. Солдатенко

Тверь, 2026

I. Аннотация

1. Цель и задачи дисциплины:

Целью освоения дисциплины является:

Формирование у студентов теоретических знаний и практических навыков в области машинного обучения, включая классические методы (линейные модели, деревья решений, ансамбли, обучение без учителя) и современные подходы на основе глубоких нейронных сетей (свёрточные, рекуррентные, трансформеры), достаточных для решения прикладных задач анализа данных, компьютерного зрения и обработки естественного языка (NLP).

Задачами освоения дисциплины являются:

1. Изучение фундаментальных концепций машинного обучения: принципы обучения с учителем и без учителя, компромисс смещения и разброса, методы регуляризации, оценка качества моделей.
2. Освоение классических методов машинного обучения: линейная и логистическая регрессии, метод опорных векторов, деревья решений, ансамблевые методы, методы кластеризации, методы понижения размерности.
3. Освоение основ глубокого обучения: архитектуры полносвязных и свёрточных нейронных сетей, метод обратного распространения ошибки, алгоритмы оптимизации, методы регуляризации и нормализации.
4. Изучение современных архитектур для компьютерного зрения.
5. Освоение методов обработки естественного языка и механизмов внимания: рекуррентные нейронные сети, механизм внимания, архитектура трансформера, основы языковых моделей.
6. Ознакомление с современными направлениями: графовые нейронные сети, нормализующие потоки, методы сжатия и развертывания моделей, экосистема фундаментальных моделей.
7. Приобретение практических навыков программной реализации методов машинного обучения и глубокого обучения с использованием современных библиотек (NumPy, PyTorch / TensorFlow) для решения задач классификации, регрессии, кластеризации, компьютерного зрения, прогнозирования временных рядов и других.

2. Место дисциплины в структуре ООП

Данная дисциплина относится к разделу «Дисциплины профиля подготовки» части, формируемой участниками образовательных отношений Блока 1.

Для освоения дисциплины, обучающиеся должны иметь навыки программирования, работы с основными структурами данных, анализа сложности алгоритмов, а также иметь базовые знания математического анализа, теории вероятностей, алгебры и геометрии. Эти знания могут быть получены в ходе изучения таких дисциплин как «Алгоритмы и структуры данных», «Программирование», «Программирование для искусственного интеллекта», «Объектно-ориентированное программирование», «Практикум на ЭВМ», «Математический анализ», «Теория вероятностей и

математическая статистика», «Алгебра и геометрия», «Методы оптимизации и исследование операций», «Функциональный анализ».

Полученные в ходе изучения дисциплины знания могут быть востребованы при обучении в магистратуре и профессиональной деятельности, а также при освоении дисциплины «Генеративные модели в машинном обучении».

3. Объем дисциплины: 6 зачетных единиц, 216 академических часов, в том числе:

контактная аудиторная работа: лекции 62 часа, в т.ч. практическая подготовка 7 часов; практические занятия 62 часа, в т.ч. практическая подготовка 7 часов;

контактная внеаудиторная работа: контроль самостоятельной работы 0 часов, в том числе курсовая работа 0 часов;

самостоятельная работа: 92 часа, в том числе контроль 68 часов.

4. Планируемые результаты обучения по дисциплине, соотнесенные с планируемыми результатами освоения образовательной программы

Планируемые результаты освоения образовательной программы (формируемые компетенции)	Планируемые результаты обучения по дисциплине
ПК-4. Способен разрабатывать и применять методы машинного обучения для решения задач	ПК-4.1. Проводит анализ требований и определяет необходимые классы задач машинного обучения ПК-4.2. Определяет метрики оценки результатов моделирования и критерии качества построенных моделей ПК-4.3. Принимает участие в оценке, выборе и при необходимости разработке методов машинного обучения
ПК-6. Способен создавать и поддерживать системы искусственного интеллекта на основе нейросетевых моделей и методов	ПК-6.1. Осуществляет оценку и выбор моделей искусственных нейронных сетей и инструментальных средств для решения поставленной задачи ПК-6.2. Разрабатывает системы искусственного интеллекта на основе моделей искусственных нейронных сетей и инструментальных средств
ПК-9. Способен создавать и внедрять одну или несколько сквозных цифровых субтехнологий искусственного интеллекта	ПК-9.1. Участвует в реализации проектов в области сквозной цифровой субтехнологии «Компьютерное зрение» ПК-9.2. Участвует в реализации проектов в области сквозной цифровой субтехнологии «Обработка естественного языка»

5. Форма промежуточной аттестации – экзамен (5, 6 семестры).

6. Язык преподавания русский.

II. Содержание дисциплины, структурированное по темам (разделам) с указанием отведенного на них количества академических часов и видов учебных занятий

Учебная программа – наименование разделов и тем	Всего (час.)	Контактная работа (час.)				Сам. работа, в том числе контр. (час.)
		Лекции		Практические занятия		
		всего	В т.ч. практ. подготовка	всего	В т.ч. практ. подготовка	
5 семестр						
1. Линейные модели	34	10	0	10	0	14
2. Непараметрические модели	20	6	1	6	1	8
3. Обучение без учителя	20	6	2	6	2	8
4. Дополнительные темы	28	8	0	8	0	12
6 семестр						
5. Основы глубокого обучения	26	8	1	8	1	10
6. Компьютерное зрение	28	8	1	8	1	12
7. Последовательности и внимание	30	8	1	8	1	14
8. Современные направления	30	8	1	8	1	14
ИТОГО	216	62	7	62	7	92

Содержание разделов (тем) дисциплины

№ п/п	Наименование разделов (тем) дисциплины	Содержание разделов (тем) дисциплин
5 семестр		
1.	Линейные модели	<ul style="list-style-type: none"> • Краткий обзор математики в машинном обучении: MLE, MAP, выборочные оценки, смещённость, основные распределения вероятностей, многомерные гауссианы • Линейная регрессия. Регуляризация Ridge, LASSO • Логистическая регрессия • Обобщённые линейные модели. Softmax-регрессия • Генеративные модели (GDA, NB с категориальной и бернуллиевской моделями)
2.	Непараметрические модели	<ul style="list-style-type: none"> • Ядерные методы. Машина опорных векторов

		<ul style="list-style-type: none"> • Деревья принятия решений • Ансамбли
3.	Обучение без учителя	<ul style="list-style-type: none"> • Bias-Variance Decomposition • ICA, LDA, t-SNE • K-means. GMM
4.	Дополнительные темы	<ul style="list-style-type: none"> • EM-алгоритм. Факторный анализ • Рекомендательные системы • Временные ряды. Модель ARIMA • Методы ранжирования
6 семестр		
5.	Основы глубокого обучения	<ul style="list-style-type: none"> • Введение в глубокое обучение • Многослойные сети и метод обратного распространения ошибки (backpropagation) • Оптимизация в нейросетях • Обучение нейронных сетей
6.	Компьютерное зрение	<ul style="list-style-type: none"> • Свёрточные нейронные сети (CNN) • Современные CNN-архитектуры и трансферное обучение • Обнаружение объектов • Семантическая сегментация
7.	Последовательности и внимание	<ul style="list-style-type: none"> • Рекуррентные нейросети • Механизм внимания • Трансформеры • Языковые модели
8.	Современные направления	<ul style="list-style-type: none"> • Графовые нейронные сети (GNN) • Нормализующие потоки • Эффективность моделей и их развертывание • Современная экосистема глубокого обучения

III. Образовательные технологии

Учебная программа – наименование разделов и тем	Вид занятия	Образовательные технологии
Линейные модели Непараметрические модели Обучение без учителя Дополнительные темы Основы глубокого обучения Компьютерное зрение Последовательности и внимание	<ul style="list-style-type: none"> • лекция, • практическая работа, • выполнение домашнего проектного задания 	<ul style="list-style-type: none"> • традиционные (фронтальная лекция, решение упражнений), • компьютерные (показ презентаций) • проектный подход, • исследовательская работа

Современные направления		
-------------------------	--	--

IV. Оценочные материалы для проведения текущей и промежуточной аттестации

ПК-4. Способен разрабатывать и применять методы машинного обучения для решения задач

ПК-6. Способен создавать и поддерживать системы искусственного интеллекта на основе нейросетевых моделей и методов

№	Результат (индикатор)	Примерная формулировка заданий	Вид/способ	Критерии оценивания
1	ПК-4.1 ПК-4.2 ПК-6.1 ПК-6.2	<p>Проектное задание</p> <ol style="list-style-type: none"> 1. Реализовать классификатор K-ближайших соседей на датасете CIFAR-10. 2. Используя перекрестную проверку, найти наилучшее значение гиперпараметра K. 3. Визуализировать результат <p>Проект выполняется в питоновском ноутбуке.</p>	<p>вид: творческое, проектное задание</p> <p>результаты: отчет, компьютерная программа</p>	<ul style="list-style-type: none"> • все задания выполнены полностью: 5 баллов, • не выполнена визуализация: 4 балла, • не найдено оптимальное значение гиперпараметра или найдено без использования перекрестной проверки: 3 балла
2	ПК-4.3 ПК-6.1 ПК-6.2	<p>Примеры вопросов для устного ответа:</p> <ol style="list-style-type: none"> 1. Что из следующего верно в отношении дропаута? <ul style="list-style-type: none"> • Во время тестирования (вывода) дропаут применяется с обратной вероятностью сохранения нейрона p • Ни один из вышеперечисленных вариантов • Чем больше вероятность сохранения нейрона, тем сильнее регуляризация весов • Дропаут приводит к разреженности обученной матрицы весов (много нулей) 2. Во время обратного распространения, когда градиент течет назад через сигмовидную нелинейность, градиент всегда будет: <ul style="list-style-type: none"> • уменьшаться по модулю, сохранять знак • увеличиваться по модулю, менять знак • увеличиваться по модулю, сохранять знак • уменьшаться по модулю, менять знак 	<p>вид: ответы на вопросы</p> <p>способ: устно</p>	<p>Правильный ответ на каждый вопрос – 0.5 балла.</p>

		<p>3. Выберите методы, которые могут ослабить взрыв градиента:</p> <ul style="list-style-type: none"> ○ использование обходных соединений (residual connections) ○ использование обрезки градиента (gradient clipping) ○ добавление пакетной нормализации ○ использование функции активации ReLU вместо сигмоиды 		
--	--	--	--	--

ПК-9. Способен создавать и внедрять одну или несколько сквозных цифровых субтехнологий искусственного интеллекта

№	Результат (индикатор)	Примерная формулировка заданий	Вид/способ	Критерии оценивания
1	ПК-9.1 ПК-9.2	<p>Проектное задание</p> <ol style="list-style-type: none"> 1. Реализовать полностью векторизованную функцию потерь для SVM 2. Реализовать полностью векторизованное выражение для его аналитического градиента 3. Проверить свою реализацию градиента с помощью числового градиента 4. Использовать валидационную выборку, чтобы подобрать скорость обучения и силу регуляризации 5. Оптимизировать функцию потерь с помощью SGD 6. Визуализировать веса обученной модели <p>Проект выполняется в питоновском ноутбуке.</p>	<p>вид: творческое, проектное задание</p> <p>результаты: отчет, компьютерная программа</p>	<ul style="list-style-type: none"> • все задания выполнены полностью: 5 баллов, • не выполнена визуализация: 4 балла, • не найдено оптимальное значение гиперпараметров или найдено без использования валидационной выборки: 3 балла
2	ПК-9.1 ПК-9.2	<p>Примеры вопросов для письменного ответа:</p> <p>Вы создаете глубокую сверточную нейронную сеть, имеющую модульную архитектуру. Пусть ваша сеть состоит из трех идентичных модулей, каждый из которых включает в себя:</p> <ul style="list-style-type: none"> • сверточный слой, • слой max-пулинга, • слой ReLU. <p>Все слои пулинга будут иметь шаг 2 и высоту/ширину 2.</p> <p>На основе данных ниже гиперпараметров сверточных слоев определите форму выходных данных, получаемых после прохождения одного изображения формы $126 \times 126 \times 3$</p>	<p>вид: ответы на вопросы</p> <p>способ: письменно</p>	<p>Правильный ответ на каждый вопрос – 0.5 балла</p>

	<p>(формат HxWxC) через всю сеть, а также количество параметров во всей сети.</p> <p>Сверточные слои имеют 64 фильтра размером 3x3, шагом 1, паддингом 0.</p> <p>Укажите форму выходных данных сети: H x W x C и количество параметров сети.</p>		
--	--	--	--

V. Учебно-методическое и информационное обеспечение дисциплины

1) Рекомендуемая литература

а) Основная литература

1. Яхьяева, Г.Э. Основы теории нейронных сетей / Г.Э. Яхьяева. - 2-е изд., испр. - Москва: Национальный Открытый Университет «ИНТУИТ», 2016. - 200 с.: ил. - (Основы информационных технологий). - ISBN 978-5-94774-818-5; То же [Электронный ресурс]. - URL: <http://biblioclub.ru/index.php?page=book&id=429110>.
2. Гудфеллоу, Я. Глубокое обучение / Я. Гудфеллоу, И. Бенджио, А. Курвилль; перевод с английского А. А. Слинкина. — 2-е изд. — Москва: ДМК Пресс, 2018. — 652 с. — ISBN 978-5-97060-618-6. — Текст: электронный // Лань: электронно-библиотечная система. — URL: <https://e.lanbook.com/book/107901>

б) Дополнительная литература

1. Goodfellow I., Bengio Y., Courville A. Deep Learning. – MIT Press, 2016. [Electronic resource]. – URL: <https://www.deeplearningbook.org>
2. Дэвис Р., Терк М. Компьютерное зрение. Передовые методы и глубокое обучение. – М.: ДМК-Пресс, 2022. – 690 с.
3. Шапиро, Л. Компьютерное зрение: учебное пособие / Л. Шапиро, Д. Стокман; под редакцией С. М. Соколова; перевод с английского А. А. Богуславского. — 4-е изд. — Москва: Лаборатория знаний, 2020. — 763 с. — ISBN 978-5-00101-696-0. — Текст: электронный // Лань: электронно-библиотечная система. — URL: <https://e.lanbook.com/book/135496>

2) Программное обеспечение

Компьютерный класс факультета прикладной математики и кибернетики № 46 (170002, Тверская обл., г.Тверь, Садовый переулок, д.35)	
Adobe Acrobat Reader DC - Russian	бесплатно
Apache Tomcat 8.0.27	бесплатно
Cadence SPB/OrCAD 16.6	Государственный контракт на поставку лицензионных программных продуктов 103 - ГК/09 от 15.06.2009

GlassFish Server Open Source Edition 4.1.1	бесплатно
Google Chrome	бесплатно
Java SE Development Kit 8 Update 45 (64-bit)	бесплатно
JetBrains PyCharm Community Edition 4.5.3	бесплатно
JetBrains PyCharm Edu 3.0	бесплатно
Kaspersky Endpoint Security 10 для Windows	Акт на передачу прав ПК545 от 16.12.2022
Lazarus 1.4.0	бесплатно
Mathcad 15 M010	Акт предоставления прав ИС00000027 от 16.09.2011
MATLAB R2012b	Акт предоставления прав № Us000311 от 25.09.2012
Многофункциональный редактор ONLYOFFICE бесплатное ПО	бесплатно
ОС Linux Ubuntu бесплатное ПО	бесплатно
MiKTeX 2.9	бесплатно
MSXML 4.0 SP2 Parser and SDK	бесплатно
NetBeans IDE 8.0.2	бесплатно
NetBeans IDE 8.2	бесплатно
Notepad++	бесплатно
Oracle VM VirtualBox 5.0.2	бесплатно
Origin 8.1 Sr2	договор №13918/M41 от 24.09.2009 с ЗАО «СофтЛайн Трейд»
Python 3.1 pygame-1.9.1	бесплатно
Python 3.4 numpy-1.9.2	бесплатно
Python 3.4.3	бесплатно
Python 3.5.1 (Anaconda3 2.5.0 64-bit)	бесплатно
WCF RIA Services V1.0 SP2	бесплатно
WinDjView 2.1	бесплатно
R Studio	бесплатно
Anaconda3 2019.07 (Python 3.7.3 64-bit)	бесплатно

Компьютерный класс факультета прикладной математики и кибернетики № 249 (170002, Тверская обл., г.Тверь, Садовый переулок, д.35)	
Cadence SPB/OrCAD 16.6	Государственный контракт на поставку лицензионных программных продуктов 103 - ГК/09 от 15.06.2009
FidesysBundle 1.4.43 x64	Акт приема передачи по договору №02/12-13 от 16.12.2013
Google Chrome	бесплатно
JetBrains PyCharm Community Edition 4.5.3	бесплатно
Kaspersky Endpoint Security 10 для Windows	Акт на передачу прав ПК545 от 16.12.2022
Lazarus 1.4.0	бесплатно
Mathcad 15 M010	Акт предоставления прав ИС00000027 от 16.09.2011
MATLAB R2012b	Акт предоставления прав № Us000311 от

	25.09.2012
MiKTeX 2.9	бесплатно
NetBeans IDE 8.0.2	бесплатно
Notepad++	бесплатно
OpenOffice	бесплатно
Origin 8.1 Sr2	договор №13918/M41 от 24.09.2009 с ЗАО «СофтЛайн Трейд»
Python 3.4.3	бесплатно
Python 3.5.1 (Anaconda3 2.5.0 64 bit)	бесплатно
R for Windows 3.3.2	бесплатно
STATGRAPHICS Centurion XVI.И	Акт приема-передачи № Tr024185 от 08.07.2010
Многофункциональный редактор ONLYOFFICE бесплатное ПО	бесплатно
ОС Linux Ubuntu бесплатное ПО	бесплатно

3) Современные профессиональные базы данных и информационные справочные системы

- ЭБС «ZNANIUM.COM» www.znanium.com;
- ЭБС «Университетская библиотека онлайн» <https://biblioclub.ru/>;
- ЭБС «Лань» <http://e.lanbook.com>.

4) Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины

- Электронная образовательная среда ТвГУ: <http://lms.tversu.ru>

VI. Методические материалы для обучающихся по освоению дисциплины

Электронные презентации по материалам курса размещаются на сайте поддержки учебного процесса по дисциплине: <http://lms.tversu.ru>

Важной составляющей данного раздела РПД являются требования к рейтинг-контролю с указанием баллов, распределенных между модулями и видами работы обучающихся.

Максимальная сумма баллов по учебной дисциплине, заканчивающейся экзаменом, по итогам семестра составляет 60 баллов (30 баллов - 1-й модуль и 30 баллов - 2-й модуль).

Обучающемуся, набравшему 40–54 балла, при подведении итогов семестра (на последнем занятии по дисциплине) в рейтинговой ведомости учета успеваемости и зачетной книжке может быть выставлена оценка «удовлетворительно».

Обучающемуся, набравшему 55–57 баллов, при подведении итогов семестра (на последнем занятии по дисциплине) в графе рейтинговой ведомости учета успеваемости «Премиальные баллы» может быть добавлено 15 баллов и выставлена экзаменационная оценка «хорошо».

Обучающемуся, набравшему 58–60 баллов, при подведении итогов семестра (на последнем занятии по дисциплине) в графе рейтинговой

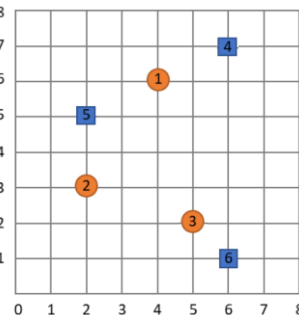
ведомости учета успеваемости «Премияльные баллы» может быть добавлено 27 баллов и выставлена экзаменационная оценка «отлично». В каких-либо иных случаях добавление премиальных баллов не допускается.

Обучающийся, набравший до 39 баллов включительно, сдает экзамен.

Распределение баллов по модулям устанавливается преподавателем и может корректироваться.

Типовые задания для рейтингового контроля (5-й семестр)

1) Задана двумерная задача с вещественными признаками (все множество примеров представлено в виде точек на плоскости). Какие-то точки помечены квадратом, какие-то кружком. Используем только горизонтальные и вертикальные решающие границы.



Подзадачи:

1. Для границ $L(x = 3)$ и $L(y = 4)$ подсчитать их потери на приведенном множестве (потеря – взвешенная сумма индексов Джини получаемых с помощью границы разбиений) и оценить, какая граница лучше.
2. Для исходного множества примеров построить по алгоритму эффективное множество решающих границ.
3. Построить на основе получившихся на предыдущем шаге границ дерево принятия решений. Если на каком-то шаге две границы имеют одинаковую эффективность, выбирайте горизонтальную. Если и в этом случае границ остается несколько, выбирайте наименьшую.

2) Пусть у нас есть небольшая выборка из 5 наблюдений: $x = \{1.5; 2.0; 3.0; 4.5; 5.0\}$. Предполагается, что данные порождены смесью двух нормальных распределений с плотностями: $p(x) = \pi_1 \mathcal{N}(x; \mu_1, \sigma_1^2) + \pi_2 \mathcal{N}(x; \mu_2, \sigma_2^2)$. Примените EM-алгоритм, чтобы оценить параметры $(\pi_1, \pi_2, \mu_1, \mu_2, \sigma_1^2, \sigma_2^2)$ по выборке. Начальные приближения: $\pi_1^{(0)} = 0.5, \pi_2^{(0)} = 0.5, \mu_1^{(0)} = 2.5, \mu_2^{(0)} = 4.5, \sigma_1^{2(0)} = 1.0, \sigma_2^{2(0)} = 1.0$.

Выполните один шаг алгоритма (один E-шаг для вычисления апостериорных вероятностей и один M-шаг для пересчёта параметров). Найдите $\pi_1^{(1)}, \pi_2^{(1)}, \mu_1^{(1)}, \mu_2^{(1)}, \sigma_1^{2(1)}, \sigma_2^{2(1)}$.

Типовые задания для рейтингового контроля (6-й семестр)

1) Постройте вычислительный граф для функции $y(x,y,z) = \sqrt{(x+y)/(2z^2)}$, нарисуйте над стрелками прямой проход по графу со значениями $x=2, y=5, z=1$, а затем и обратное распространение.

2) Дана полносвязная сеть прямого распространения следующей конфигурации: размер изображений – 3072 пикселя (уже вытянуто в вектор),

количество нейронов на первом скрытом слое – 10000, количество нейронов на втором скрытом слое – 5000, количество нейронов на выходном слое – 10. Рассчитайте:

- Размеры всех матриц и векторов параметров
- Количество параметров нейронной сети (не забудьте смещения!)
- Во время обучения сеть должна хранить активации на всех слоях.

Предположим, обучение происходит пакетами по 100 изображений. Рассчитайте объем памяти, требуемый для хранения всех активаций.

Типовые вопросы для подготовки к экзамену (5-й семестр)

- Линейная регрессия: нормальные уравнения, геометрическая интерпретация, предположения модели.
- Регуляризация: Ridge-регрессия (L2), LASSO-регрессия (L1) — сравнение, геометрическая и аналитическая интерпретация, понятие разреженности (sparsity).
- Логистическая регрессия: вывод функции потерь (log loss) через MLE, сигмоида, градиентный спуск для обучения.
- Обобщённые линейные модели (GLM): концепция, примеры: линейная и логистическая регрессия как частные случаи.
- Softmax-регрессия: обобщение логистической регрессии на многоклассовый случай, функция softmax, перекрёстная энтропия.
- Разведочный анализ данных: назначение и методы.
- Что такое спрямляющие пространства для линейных моделей и для чего они используются.
- Задача оценки плотности вероятности: как это работает, для каких задач бывает полезно.
- Генеративные модели классификации: Гауссовский дискриминантный анализ (GDA), его предположения и сравнение с логистической регрессией.
- Наивный Байес: «наивное» предположение о независимости признаков, варианты для категориальных (Multinomial NB) и бинарных (Bernoulli NB) данных.
- Метод опорных векторов (SVM): идея максимального зазора (hard margin), метод мягкого зазора (soft margin).
- Ядерный трюк (Kernel Trick): переход в пространство большей размерности, определение ядра, популярные ядра (линейное, полиномиальное, RBF).
- Деревья решений: алгоритм построения (CART), критерии ветвления (энтропия, индекс Джини, MSE), проблемы переобучения и важность обрезки.
- Компромисс смещения и разброса (Bias-Variance Tradeoff): разложение ошибки на составляющие, интерпретация для моделей разной сложности.
- Бэггинг (Bootstrap Aggregating): принцип, снижение дисперсии, алгоритм Random Forest (дополнительная случайность признаков).

- Бустинг: основная идея последовательного исправления ошибок. Gradient Boosting: представление обучения как градиентного спуска в пространстве функций. AdaBoost – основная идея, вывод формулы для пересчёта весов. Простой способ расчёта новых значений весов. Построение результирующей модели.
- ООВ-валидация для бэггинга. Принцип, почему это работает, когда, как и где используется.
- Кластеризация K-Means: алгоритм, ограничения (форма кластеров, выбор K), метрики оценки кластеризации (инерция, силуэт).
- EM-алгоритм: общая схема (E-шаг, M-шаг) на интуитивном уровне, применение для Гауссовских смесей (GMM).
- Метод главных компонент (PCA): две формулировки (максимизация дисперсии проекции / минимизация ошибки реконструкции), геометрическая интерпретация.
- t-SNE: цель (визуализация), принцип работы (сохранение локальных расстояний), сравнение с PCA.
- Независимый компонентный анализ (ICA): цель (поиск независимых источников), отличие от PCA.
- Коллаборативная и контентная фильтрация: принципы работы, проблемы (холодный старт, разреженность матрицы), гибридные подходы.
- Моделирование временных рядов: компоненты ряда (тренд, сезонность, шум), модель ARIMA(p,d,q) — смысл параметров и этапы построения.
- Метрики качества рекомендаций и общий пайплайн рекомендательных систем (генерация кандидатов, ранжирование, реранкинг).

Типовые вопросы для подготовки к экзамену (6-й семестр)

- В чем принципиальное отличие нейронных сетей от линейных моделей и GLM? Когда deep learning оправдан, а когда лучше использовать классические методы?
- Что такое «нейросеть как универсальный аппроксиматор»? Какие драйверы прогресса (данные, GPU, фреймворки) сделали глубокое обучение практически применимым?
- Объясните устройство feedforward-сети: как проходят прямой и обратный проход? Что такое вычислительный граф и как с его помощью работает автоматическое дифференцирование (autodiff)?
- Сравните функции активации (sigmoid, tanh, ReLU, GELU). Почему ReLU стала стандартом? В чем ее недостатки и как они решаются?
- С какими проблемами сталкивается обычный SGD? Как методы с инерцией (Momentum) и адаптивные методы (RMSProp, Adam) решают эти проблемы? В чем компромисс между Adam и SGD?
- Что такое исчезающие и взрывающиеся градиенты? Как правильная инициализация весов (Xavier, He) и learning rate schedules помогают с этим бороться?

- Как работает dropout во время обучения и инференса? Почему он помогает бороться с переобучением? Как L2-регуляризация и early stopping дополняют его?
- Объясните принцип работы Batch Normalization и Layer Normalization. В чем их ключевое различие? Когда применяют каждый из методов? Что такое data augmentation и почему это тоже регуляризация?
- Зачем нужны сверточные слои вместо полносвязных? Что такое локальное рецептивное поле, разделяемые веса, padding, stride и pooling? Какую инвариантность они обеспечивают?
- Как устроена residual connection и почему она позволила обучать очень глубокие сети (ResNet)? Что происходит с градиентами в такой архитектуре?
- Что такое трансферное обучение и fine-tuning? В чем разница между «замороженными» и «обучаемыми» слоями? Как методы интерпретации (Grad-CAM) помогают понять, на что смотрит сеть?
- Назовите ключевые архитектуры CNN (AlexNet, VGG, ResNet, EfficientNet). Как менялись идеи о масштабировании (depth, width, resolution) и почему EfficientNet считается более эффективным?
- В чем задача обнаружения объектов? Чем отличаются двухэтапные подходы (R-CNN, Fast/Faster R-CNN) от одноэтапных (YOLO)? Какие метрики используются (IoU, mAP)?
- В чем задача семантической сегментации? Как устроены архитектуры encoder–decoder (FCN, U-Net)? Где они применяются (медицина, спутники)?
- Как устроены рекуррентные нейросети (RNN)? В чем их проблемы при обучении на длинных последовательностях? Как LSTM и GRU решают проблему исчезающего градиента?
- Сравните RNN и методы из классического анализа временных рядов (ARIMA). Когда имеет смысл использовать RNN/LSTM, а когда — классические модели?
- В чем идея механизма внимания (attention)? Чем он лучше рекуррентности для обработки последовательностей? Что такое self-attention?
- Как устроен механизм внимания в трансформерах? Что такое multi-head attention и positional encoding? Чем отличается encoder от decoder?
- Что такое языковое моделирование? В чем разница между авторегрессивными (GPT) и маскированными (BERT) языковыми моделями? Как transfer learning применяется в NLP?
- Какие задачи решаются с помощью BERT и GPT? В чем их принципиальные архитектурные различия (encoder-only vs decoder-only)?
- Какие данные называют графовыми? Как устроен принцип message passing в графовых нейронных сетях? Приведите примеры применения GNN.

- В чем разница между GCN и GraphSAGE? Как адаптировать идеи свёртки для нерегулярных графовых структур?
- Зачем нужны нормализующие потоки (normalizing flows)? Какое ключевое свойство (обратимость) делает их полезными для генеративных задач?
- В чем идея flow-based моделей? Как они связаны с подготовкой к изучению VAE, GAN и диффузионных моделей?
- Что такое knowledge distillation? Как обучить маленькую модель (student) на предсказаниях большой (teacher)? Зачем это нужно?
- Какие методы сжатия моделей существуют (pruning, quantization, distillation)? С какими требованиями (latency, throughput) сталкиваются при деплое моделей?
- Что такое foundation models? Приведите примеры. В чем особенности и риски использования мультимодальных моделей?
- Каковы ограничения современных моделей глубокого обучения (data hunger, отсутствие рассуждения, дыры в безопасности)? Какой мост эти идеи дают к VAE, GAN, диффузионным моделям?
- Как устроены U-Net и FCN для сегментации? Почему в медицинской диагностике и спутниковых снимках эти архитектуры особенно популярны?
- Как работает механизм внимания в деталях (Q, K, V)? Почему softmax применяется именно так, а не иначе? Зачем нужен scaling?
- Как происходит масштабирование (scaling laws) в CNN и трансформерах? Что такое EfficientNet и как она находит баланс между depth, width и resolution?
- Каковы практические trade-offs при выборе архитектуры модели (скорость, точность, память)? Как domain adaptation помогает переносить модели на новые данные без полного переобучения?

Примеры заданий для самостоятельной работы (5-й семестр)

Задание 1. Линейная и логистическая регрессия с нуля

Цели задания:

- Реализовать линейную регрессию двумя способами: через нормальные уравнения и через градиентный спуск.
- Реализовать логистическую регрессию с L2-регуляризацией (Ridge) и L1-регуляризацией (LASSO).
- Сравнить поведение L1 и L2 регуляризации на синтетических данных с избыточными признаками.
- Реализовать функцию кросс-валидации для подбора гиперпараметров (скорость обучения, коэффициент регуляризации).

Что нужно сделать:

1. Сгенерировать синтетические данные: 1000 объектов, 10 признаков, из которых только 3 действительно влияют на целевую переменную (остальные — шум).
2. Реализовать класс LinearRegression с методами `fit_normal()` и `fit_gd()`.
3. Реализовать класс LogisticRegression с методами `fit_gd()`, `predict()`, `predict_proba()`.
4. Для LASSO реализовать проксимальный градиентный спуск (proximal gradient descent).
5. На графиках показать, как L1-регуляризация обнуляет неважные веса, а L2 — нет.
6. Сравнить ассигасу и время обучения при разных методах.

Задание 2. Деревья решений и ансамбли с нуля

Цели задания:

- Реализовать дерево решений для классификации и регрессии без готовых библиотек (только NumPy).
- Реализовать Random Forest (бэггинг + случайный подбор признаков) поверх своего дерева.
- Реализовать градиентный бустинг (Gradient Boosting) поверх своего дерева.
- Сравнить поведение одного дерева, случайного леса и бустинга на данных с разным уровнем шума.

Что нужно сделать:

1. Реализовать класс DecisionTree с поддержкой:
 - Критериев ветвления: энтропия, Джини (для классификации), MSE (для регрессии).
 - Механизма обрезки (pruning) для борьбы с переобучением.
2. Реализовать класс RandomForest:
 - Бутстрэп-выборки для каждого дерева.
 - Случайное подмножество признаков для каждого разбиения.
3. Реализовать класс GradientBoosting:
 - Последовательное добавление деревьев, каждое из которых предсказывает псевдо-остатки (антиградиент).
 - Параметры: количество деревьев, скорость обучения (learning rate), максимальная глубина.
4. Сравнить модели на реальном датасете (например, Breast Cancer из sklearn или Titanic). Построить графики зависимости ошибки от сложности модели.

Задание 3. Кластеризация, PCA и EM для Gaussian Mixture

Цели задания:

- Реализовать K-Means с нуля и исследовать его ограничения.
- Реализовать метод главных компонент (PCA) двумя способами: через SVD и через ковариационную матрицу.
- Реализовать EM-алгоритм для обучения гауссовской смеси (GMM).

- Сравнить K-Means и GMM на синтетических данных с разной формой кластеров.

Что нужно сделать:

1. Сгенерировать три набора синтетических данных:
 - Кластеры в форме сфер (хорошо разделимые).
 - Кластеры в форме эллипсов (разная ориентация).
 - Кластеры, где один содержит другой (вложенные).
2. Реализовать K-Means:
 - Инициализация центроидов (random, k-means++).
 - Критерий остановки (изменение центроидов или инерции).
 - Выбор K методом локтя и силуэтом.
3. Реализовать PCA:
 - Уменьшение размерности с 3D до 2D.
 - Визуализация данных до и после проекции.
4. Реализовать EM для GMM:
 - E-шаг: вычисление soft-принадлежностей (posterior probabilities).
 - M-шаг: пересчёт параметров (средние, ковариации, веса компонент).
 - Визуализация доверительных эллипсов компонент.
5. Показать, почему K-Means не работает на вложенных кластерах, а GMM — работает.

Примеры заданий для самостоятельной работы (6-й семестр)

Задание 1. Полносвязная и свёрточная сеть с нуля (NumPy + autograd)

Цели задания:

- Реализовать полносвязную нейронную сеть (MLP) с нуля, используя только NumPy и собственный вычислительный граф.
- Реализовать обратное распространение (backpropagation) вручную для всех слоёв.
- Реализовать свёрточный слой (Conv2D) и max-pooling с нуля.
- Сравнить эффективность MLP и CNN на задаче классификации изображений (MNIST / Fashion-MNIST).

Что нужно сделать:

1. Реализовать модульную систему с классами Linear, ReLU, Sigmoid, CrossEntropyLoss, MSELoss.
2. Реализовать класс Conv2D:
 - Прямой проход: свёртка с ядром, padding, stride.
 - Обратный проход: градиенты относительно входа и относительно ядра (через im2col или циклы).
3. Реализовать класс MaxPool2d с обратным проходом (запоминать позиции максимумов).
4. Реализовать оптимизаторы: SGD, Momentum, Adam.
5. Собрать MLP (2-3 слоя) и CNN (Conv → Pool → Conv → Pool → Linear) и сравнить accuracy на MNIST.

Задание 2. Обучение CNN для классификации и визуализация через Grad-CAM

Цели задания:

- Реализовать обучение современной CNN-архитектуры (ResNet или EfficientNet) на датасете изображений (например, CIFAR-10).
- Применить трансферное обучение: дообучить (fine-tune) предобученную на ImageNet модель.
- Реализовать Grad-CAM для интерпретации решений модели.
- Исследовать влияние аугментации данных и dropout на качество.

Что нужно сделать:

1. Загрузить предобученную модель (ResNet18 / EfficientNet-B0) из PyTorch/TensorFlow.
2. Заморозить все слои, кроме последнего, и обучить классификатор (linear head).
3. Разморозить последние 2-3 блока и выполнить fine-tuning с маленькой learning rate.
4. Реализовать Grad-CAM:
 - Получить градиенты относительно карт признаков последнего свёрточного слоя.
 - Усреднить градиенты по каналам для получения весов.
 - Наложить взвешенную сумму карт признаков на исходное изображение.
5. Сравнить ассурасу при обучении с нуля, с frozen backbone и с fine-tuning.
6. Визуализировать Grad-CAM для правильно и неправильно классифицированных примеров.

Задание 3. Рекуррентная сеть и трансформер для предсказания временных рядов

Цели задания:

- Реализовать LSTM/GRU с нуля (или используя фреймворк) для задачи прогнозирования временного ряда.
- Реализовать простой трансформер (encoder-only) с механизмом внимания для той же задачи.
- Сравнить RNN/LSTM, трансформер и классический ARIMA на реальных данных.
- Исследовать влияние длины контекста (sequence length) на качество предсказания.

Что нужно сделать:

1. Взять реальный временной ряд (например, температура в городе, цена акций, число пассажиров авиакомпании — данные AirPassengers).
2. Реализовать модель LSTM (или GRU) с одним или двумя слоями.
3. Реализовать трансформер для временных рядов:
 - Positional encoding (синус-косинус или обучаемый).
 - Multi-head self-attention.
 - Слой FeedForward.

4. Реализовать функцию скользящего окна для преобразования ряда в обучение с учителем.
5. Сравнить метрики (MSE, MAE) на тестовой выборке для:
 - ARIMA (использовать готовую библиотеку statsmodels).
 - LSTM (обученный с нуля).
 - Трансформера.
6. Построить графики реальных vs предсказанных значений для всех моделей.
7. Сделать вывод: когда RNN лучше трансформера, а когда — наоборот (в зависимости от длины последовательности и объема данных).

VII. Материально-техническое обеспечение

Для аудиторной работы.

Компьютерный класс факультета ПМиК № 4в (170002, Тверская обл., г.Тверь, Садовый переулок, д.35)	Компьютер, экран, маркерная доска, проектор, кондиционер.
Учебная аудитория № 206 (170002, Тверская обл., г.Тверь, Садовый переулок, д.35)	Набор учебной мебели, экран, проектор.

Для самостоятельной работы

Помещение для самостоятельной работы обучающихся: Компьютерный класс факультета прикладной математики и кибернетики № 4б (170002, Тверская обл., г.Тверь, Садовый переулок, д.35)	Компьютер, экран, проектор, кондиционер.
--	--

VIII. Сведения об обновлении рабочей программы дисциплины

№ п.п.	Обновленный раздел рабочей программы дисциплины	Описание внесенных изменений	Дата и протокол заседания кафедры, утвердившего изменения
1.			
2.			
3.			
4.			
5.			