МИНОБРНАУКИ РОССИИ
Ярославский государственный университет им. П.Г. Демидова

Кафедра цифровых технологий и машинного обучения

УТВЕРЖДАЮ

Декан физического факультета
_____ И.С. Огнев
(подпись)

«17» мая 2022 г.

Рабочая программа дисциплины
«Машинное обучение»

Направление подготовки
03.03.03 Радиофизика

Направленность (профиль)
Телекоммуникационные системы и технологии

Форма обучения
очная

Программа одобрена
на заседании кафедры
от «28» апреля 2022 года, протокол № 9

Программа одобрена НМК
физического факультета
протокол № 5 от «11» мая 2022 года

Ярославль

1. Цели освоения дисциплины

Целью освоения дисциплины является изучение студентами эффективных алгоритмов систем искусственного интеллекта и получение опыта их практического применения.

В процессе преподавания дисциплины решаются следующие задачи:

- ознакомление с методами обучения с учителем;
- ознакомление с методами обучения без учителя;
- изучение алгоритмов глубокого обучения;
- практическое использование систем искусственного интеллекта.

2. Место дисциплины в структуре образовательной программы

Дисциплина относится к части, формируемой участниками образовательного процесса, и является дисциплиной по выбору.

Дисциплина тесно связана с курсами «Математический анализ», «Теория вероятностей и математическая статистика». Формируемые навыки в ходе освоения курса на этапах дальнейшего обучения могут являться средством выполнения научных работ. Следует отметить динамику постоянного совершенствования систем на базе методов машинного обучения, что требует от процесса преподавания постоянной доработки и переработки некоторых разделов.

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

Процесс изучения дисциплины направлен на формирование следующих элементов компетенций в соответствии с ФГОС ВО, ОП ВО и приобретения следующих знаний, умений, навыков и (или) опыта деятельности:

Формируемая компетенция (код и формулировка)	**Индикатор достижения компетенции (код и формулировка)**	**Перечень планируемых результатов обучения**
Профессиональные компетенции		
ПК-2 Способен применять современные теоретические и (или) экспериментальные методы исследования с целью анализа текущего состояния телекоммуникационных устройств, систем и сетей	ИД_ПК-2.1 Знает основные характеристики телекоммуникационных устройств, систем и сетей	**знает:** – подходы, позволяющие выполнять настройку параметров алгоритмов машинного обучения; **умеет:** – реализовывать алгоритмы машинного обучения с использованием средств компьютерного моделирования
	ИД_ПК-2.3 Проводит теоретические исследования телекоммуникационных устройств, систем и сетей	**знает:** – методы машинного обучения с учителем и без учителя; – примеры практических задач, в которых возможно эффективное использование методов машинного обучения **умеет:** – выполнять настройку параметров алгоритмов машинного обучения; – анализировать правильность работы алгоритмов машинного обучения. **владеет:** – навыками построения, анализа и компьютерного моделирования систем машинного обучения.

4. Объем, структура и содержание дисциплины

Общая трудоемкость дисциплины составляет 2 зачетные единицы, 72 акад. часа.

№ п/п	Темы (разделы) дисциплины, их содержание	Семестр	Виды учебных занятий, включая самостоятельную работу студентов, и их трудоемкость (в академических часах)						Формы текущего контроля успеваемости Форма промежуточной аттестации
			Контактная работа					Самостоятельная работа	
			лекции	практические	лабораторные	консультации	аттестационные испытания		
1	Введение и обзор материала курса. Введение в язык Python	7	3		1,5			2	
2	Линейная регрессия с одной переменной. Линейная регрессия со множеством переменных. Классификация. Логистическая регрессия	7	5		2,5	0,25		2	Тестирование, задание для самостоятельной работы
3	Искусственные нейронные сети (представление)	7	4		2			2	Тестирование, задание для сам. работы
4	Искусственные нейронные сети (обучение)	7	3		1,5	0,25		2	Тестирование, задание для сам. работы
5	Рекомендации по применению алгоритмов машинного обучения. Построение систем машинного обучения. Оптическое распознавание символов. Формирование базы данных	7	3		1,5	0,25		2	Тестирование, задание для самостоятельной работы
6	Кластеризация	7	3		1,5	0,25		2	Тестирование, задание для сам. работы
7	Анализ главных компонент	7	3		1,5	0,25		2	Тестирование, задание для сам. работы
8	Детектирование лиц на основе алгоритма Виола/Джонса	7	3		1,5	0,25		1	
9	Машинное обучение на больших базах данных	7	3		1,5	0,25		1	
10	Глубокое обучение. Свёрточные нейронные сети	7	4		2	0,25		1	
							0,3	1,7	Зачет
	ИТОГО		**34**		**17**	**2**	**0,3**	**18,7**	

Содержание разделов дисциплины

Тема № 1
Введение и обзор материала курса. Введение в язык Python

Что такое машинное обучение? Примеры задач, решаемые в области машинного обучения (обучение с учителем и без учителя, стимулированное обучение, эволюционное обучение). Обзор основных инструментов языка Python.

Тема № 2
Линейная регрессия с одной переменной. Линейная регрессия со множеством переменных. Классификация. Логистическая регрессия

Общая постановка задачи регрессии. Что такое гипотеза, параметры модели и стоимостная функция на примере задачи линейной регрессии? Минимизация стоимостной функции (нормальные уравнения и численная оптимизация). Метод градиентного спуска. Масштабирование признаков и настройка скорости обучения алгоритма.

Общая постановка задачи классификации. Логистическая регрессия и ее стоимостная функция. Граница принятия решения. Многоклассовая классификация на основе логистической регрессии (подходы «один против всех» и «один против одного»).

Тема № 3
Искусственные нейронные сети (представление)

Проблема нелинейной классификации. Биологические нейроны и мозг. Модель нейрона и искусственные нейронные сети. Нейронные сети прямого распространения. Примеры реализации логических операций на основе искусственных нейронных сетей. Классификация рукописных цифр. Классификация объектов в сложных сценах. Многоклассовая классификация для нейронных сетей.

Тема № 4
Искусственные нейронные сети (обучение)

Регуляризация и проблема переобучения. Регуляризованная стоимостная функция для линейной и логистической регрессии, нейронной сети прямого распространения. Алгоритм обратного распространения ошибки. Градиентная проверка. Проблема симметричности весов нейронной сети. Автономное вождение (проект ALVINN).

Тема № 5
Рекомендации по применению алгоритмов машинного обучения. Построение систем машинного обучения. Оптическое распознавание символов. Формирование базы данных.

Тема № 6
Кластеризация

Что такое кластеризация? Разновидности алгоритмов кластеризации данных. Алгоритм K-средних (стоимостная функция, случайная инициализация, выбор числа кластеров). Примеры использования алгоритма K-средних для обработки цифровых изображений.

Тема № 7

Анализ главных компонент

Задача сокращения размерности данных. Общая формулировка анализа главных компонент и его реализация на практике. Выбор числа главных компонент. Рекомендации по применению анализа главных компонент. Распознавание лиц с использованием анализа главных компонент (пространство лиц, собственные лица, структура алгоритма распознавания лиц).

Тема № 8

Детектирование лиц на основе алгоритма Виола/Джонса

Общее описание проблемы детектирования лиц. Признаки, используемые в алгоритме детектирования лиц Виола/Джонса. Каскад классификаторов. Интегральные изображения. Слабые классификаторы. Комитет классификаторов и бустинг. Данные для обучения каскада классификаторов и тестирование алгоритма Виола/Джонса.

Тема № 9

Машинное обучение на больших базах данных

Общая постановка задачи обучения на больших базах данных. Стохастический градиентный спуск и минигрупповой градиентный спуск. Онлайн-обучение. Технология MapReduce и распараллеливание данных.

Тема № 10

Глубокое обучение. Свёрточные нейронные сети

Проблема глубокого обучения. Архитектура свёрточной нейронной сети (свёрточный слой и слой прореживания, полносвязные слои). Обучение сверточной нейронной сети. Предварительное обучение и автоэнкодеры. Примеры использования свёрточных нейронных сетей для решения различных практических задач.

5. Образовательные технологии, в том числе технологии электронного обучения и дистанционные образовательные технологии, используемые при осуществлении образовательного процесса по дисциплине

В процессе обучения используются следующие образовательные технологии:

Вводная лекция – дает первое целостное представление о дисциплине и ориентирует студента в системе изучения данной дисциплины. Студенты знакомятся с назначением и задачами курса, его ролью и местом в системе учебных дисциплин и в системе подготовки в целом. Дается краткий обзор курса, история развития науки и практики, достижения в этой сфере, имена известных ученых, излагаются перспективные направления исследований. На этой лекции высказываются методические и организационные особенности работы в рамках данной дисциплины, а также дается анализ рекомендуемой учебно-методической литературы.

Лабораторная работа – организация учебной работы с реальными материальными и информационными объектами, экспериментальная работа с аналоговыми моделями реальных объектов.

Консультации – вид учебных занятий, являющийся одной из форм контроля самостоятельной работы студентов. На консультациях по просьбе студентов рассматриваются наиболее сложные моменты при освоении материала дисциплины, преподаватель отвечает на вопросы студентов, которые возникают у них в процессе самостоятельной работы.

6. Перечень лицензионного и (или) свободно распространяемого программного обеспечения, используемого при осуществлении образовательного процесса по дисциплине

В процессе осуществления образовательного процесса по дисциплине используются:

для формирования материалов для текущего контроля успеваемости и проведения промежуточной аттестации, для формирования методических материалов по дисциплине:

– программы Microsoft Office;

– Adobe Acrobat Reader.

7. Перечень современных профессиональных баз данных и информационных справочных систем, используемых при осуществлении образовательного процесса по дисциплине (при необходимости)

В процессе осуществления образовательного процесса по дисциплине используются:

Автоматизированная библиотечно-информационная система «БУКИ-NEXT» http://www.lib.uniyar.ac.ru/opac/bk_cat_find.php.

8. Перечень основной и дополнительной учебной литературы, ресурсов информационно-телекоммуникационной сети «Интернет» (при необходимости), рекомендуемых для освоения дисциплины

а) основная литература

1. Осипов, Г. С. Методы искусственного интеллекта / Осипов Г. С. - Москва : ФИЗМАТЛИТ, 2011. - 296 с. - URL : https://www.studentlibrary.ru/book/ISBN9785922113236.html

б) дополнительная литература

1. Местецкий, Л. М. Математические методы распознавания образов / Местецкий Л. М. - Москва : Национальный Открытый Университет "ИНТУИТ", 2016. - URL : https://www.studentlibrary.ru/book/intuit_134.html

в) ресурсы сети «Интернет»:

1. Электронная библиотека учебных материалов ЯрГУ (http://www.lib.uniyar.ac.ru/opac/bk_cat_find.php).

9. Материально-техническая база, необходимая для осуществления образовательного процесса по дисциплине

Материально-техническая база, необходимая для осуществления образовательного процесса по дисциплине включает в свой состав специальные помещения:

- учебные аудитории для проведения занятий лекционного типа;
- учебные аудитории для проведения лабораторных работ;
- учебные аудитории для проведения групповых и индивидуальных консультаций,
- учебные аудитории для проведения текущего контроля и промежуточной аттестации;
- помещения для самостоятельной работы;
- помещения для хранения и профилактического обслуживания технических средств обучения.

Специальные помещения укомплектованы средствами обучения, служащими для представления учебной информации большой аудитории.

Число посадочных мест в лекционной аудитории больше либо равно списочному составу потока, а в аудитории для практических занятий (семинаров) – списочному составу

группы обучающихся. (Для проведения лабораторных работ группа обучающихся делится на две подгруппы).

Автор:

Доцент кафедры инфокоммуникаций и радиофизики , к.т.н. Хрящев В.В.
(должность, ученая степень) *(Фамилия И.О.)*

Приложение №1 к рабочей программе дисциплины
«Машинное обучение»

Фонд оценочных средств
для проведения текущего контроля успеваемости
и промежуточной аттестации студентов
по дисциплине

. Типовые контрольные задания и иные материалы,
используемые в процессе текущего контроля успеваемости

Примерные варианты заданий для самостоятельной работы

Тема: Линейная регрессия

1. Ознакомиться с материалами лекций № 1 и № 2.
2. Установить необходимое программное обеспечение. При выполнении задания наверняка понадобятся **Python 3**, **NumPy** и **Matplotlib**.
3. Ознакомиться с содержимым папки с заданием, которая включает в себя файлы, представленные ниже.

`main_one.py` – «основной» модуль, необходимый для выполнения первой части задания, который поможет выполнить его поэтапно. Настоящий программный код не требует какой-либо коррекции!

`main_multi.py` – «основной» модуль, необходимый для выполнения второй части задания, который поможет выполнить его поэтапно. Настоящий программный код не требует какой-либо коррекции!

`data1.txt` – база данных для выполнения первой части задания.

`data2.txt` – база данных для выполнения второй части задания.

`plotData.py` – модуль, содержащий функцию `plotData`, которая необходима для визуализации данных.

`computeCost.py` – модуль, содержащий функцию `computeCost`, которая необходима для вычисления значения стоимостной функции линейной регрессии.

`gradientDescent.py` – модуль, содержащий функцию `gradientDescent`, которая необходима для выполнения градиентного спуска с целью поиска параметров модели линейной регрессии.

`featureNormalize.py` – модуль, содержащий функцию `featureNormalize`, которая необходима для нормализации признаков.

`normalEqn.py` – модуль, содержащий функцию `normalEqn`, которая необходима для поиска параметров модели линейной регрессии с использованием нормальных уравнений.

4. Выполнить первую часть задания, связанную с реализацией и исследованием линейной регрессии с одной переменной.
5. Выполнить вторую часть задания, связанную с реализацией и исследованием линейной регрессии со множеством переменных.
6. Ответить на вопросы, необходимые для составления отчета по данному практическому заданию. Отчет сдается на проверку в печатной или письменной форме в указанные сроки.

При выполнении данного задания требуется заполнить пустые места программного кода в блоках с комментарием «Ваш код здесь». Данную процедуру необходимо выполнить для следующих функций: `plotData`, `computeCost`, `gradientDescent`.

1. При решении любой задачи с использованием инструментов машинного обучения важным является понимание структуры анализируемых данных и их визуализация в случае возможности. В первой части задания предлагается использовать базу данных из файла **data1.txt**. Данные представляют собой множество объектов, описываемых одним признаком (численность населения в некотором городе) и меткой (прибыль, которую можно получить при продаже определенного товара в городе с соответствующей численностью населения). Завершите программный код в модуле **plotData.py**, который позволит выполнять визуализацию данных. Завершение модуля подразумевает под собой написание строчек программного кода, которые позволят вызвать функцию из соответствующего модуля в файле **main_one.py**, позволяя решить определенный кусок настоящего задания. Например, в данном случае завершенный программный код будет выглядеть так, как представлено на рис. 1.\

```
import matplotlib.pyplot as plt
import numpy as np

def plotData(data):
    """
        Функция позволяет выполнить визуализацию данных в декартовой
        системе координат с подписанным осями (численность населения
        и прибыль)
    """

    # ====================== Ваш код здесь ======================
    # Инструкция: визуализируйте данные с использованием функций
    # figure и plot. Подпишите оси с использованием функций xlabel
    # и ylabel, предполагая, что аргументами этих функций являются
    # численность населения по x и прибыль по y

    plt.figure()                                                                    # создание нового окна
    plt.plot(data[:, 0], data[:, 1], 'rx', markersize = 5, label = 'Тренировочные данные') # построение точек данных
    plt.legend(loc = 'upper right', shadow = True, fontsize = 12, numpoints = 1)   # построение легенды
    plt.xlabel('Численность населения в 10,000')                                    # обозначение оси x
    plt.ylabel('Прибыль в $10,000')                                                 # обозначение оси y
    plt.grid()                                                                      # создание координатной сетки

    # ==========================================================
```

Рис. 1. Завершенный программный код для функции `plotData`

После завершения каждого блока кода интерпретируйте файл **main_one.py** с целью проверки правильности работы соответствующей части задания. Результат визуализации данных с использованием функции `plotData` представлен на рис. 2. В случае успешной интерпретации программного кода разрешается перейти к следующему пункту задания.

2. Завершите программный код в модуле **computeCost.py**, который позволит вычислить значение стоимостной функции для линейной регрессии. Формулы, описывающие ее вычисление, представлены в лекции № 1. При выполнении данной части задания могут понадобиться функции из библиотеки **NumPy**, представленные ниже.

`dot` – позволяет вычислить матричное произведение для двумерных массивов и скалярное произведение для одномерных массивов (без комплексного сопряжения).

`sum` – позволяет вычислить сумму элементов вдоль определенной размерности двумерного массива и сумму всех элементов для одномерного массива.

Также полезным может оказаться оператор поэлементное возведение компонентов двумерного и одномерного массивов в квадрат: `** 2`.

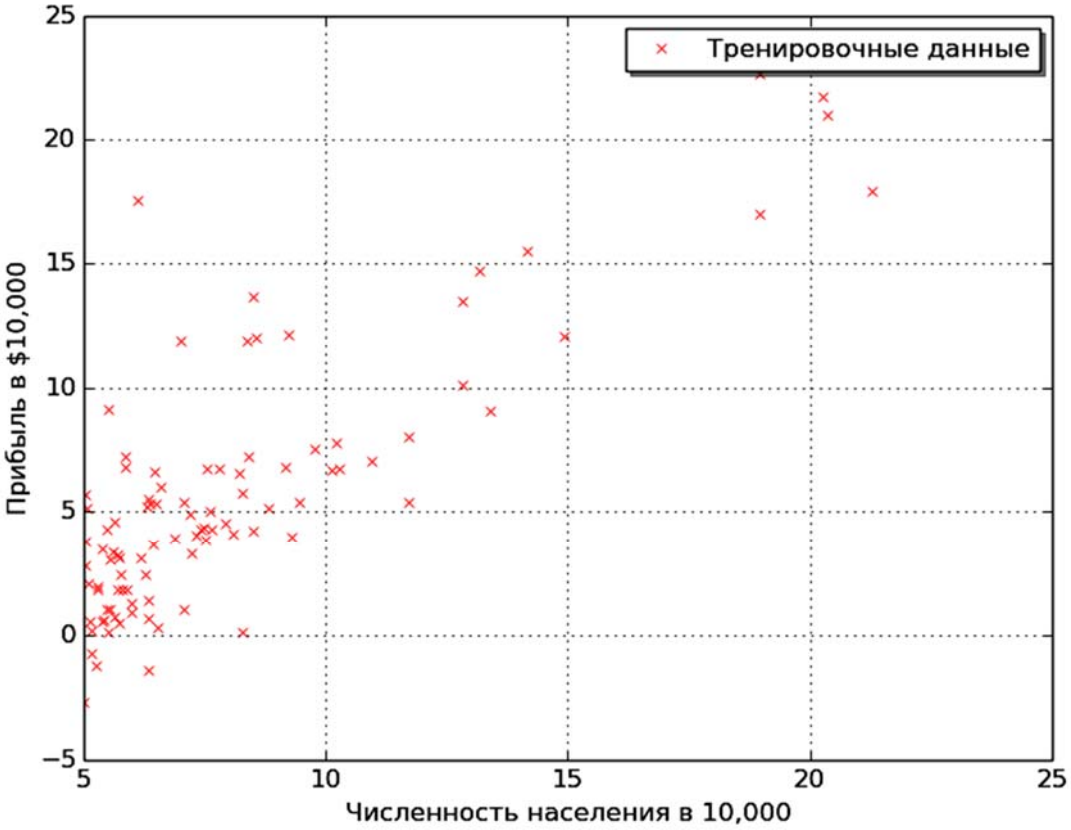


Рис. 2. Результат визуализации тренировочных данных

3. Завершите программный код в модуле **gradientDescent.py**, который позволит выполнить алгоритм градиентного спуска с целью обучения параметров модели линейной регрессии. Формулы, описывающие реализацию градиентного спуска, представлены в лекции № 1. При выполнении данной части задания могут понадобиться следующие функции из библиотеки **NumPy**: `dot` и `transpose`.

`transpose` – позволяет выполнить транспонирование массива. Для одномерного массива данная функция не оказывает никакого действия, а для двумерного массива использование функции соответствует обычному матричному транспонированию.

После обучения параметров модели линейной регрессии с одной переменной с настройками градиентного спуска, заданными по умолчанию в файле **main_one.py** должен получиться результат, представленный на рис. 3.

4. После завершения предыдущих пунктов выполните предсказание прибыли от продажи товара в городах с численностью населения 35,000 и 70,000. При выполнении задания обратите внимание на то, что в матрице объекты-признаки, сформированной в файле **main_one.py** после загрузки базы данных из **data1.txt**, единственный признак объекта, описывающий численность населения в городе, является нормированным на значение 10,000.

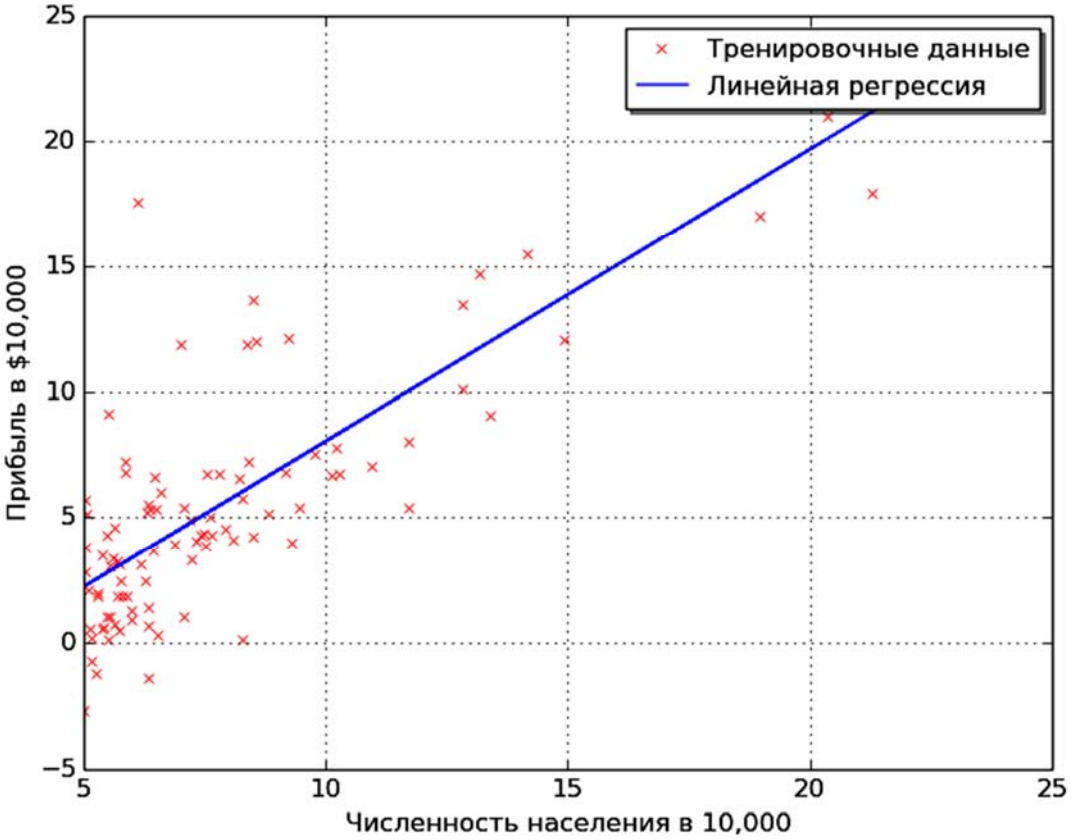


Рис. 3. Результат визуализации тренировочных данных и линии регрессии (гипотезы для линейной регрессии с одной переменной)

Используя файл **main_one.py** ответьте на следующие вопросы по первой части практического задания.

1. Чему равно значение стоимостной функции для случая, когда все параметры модели равны нулю (**35 баллов**)?

2. Чему равны значения параметров обученной модели линейной регрессии с одной переменной для случая, когда параметр сходимости равен 0.01, а число итераций градиентного спуска равно 1500 (**40 баллов**)?

3. Чему равна прибыль от продажи товара в городах с численностью населения 35,000 и 70,000 для обученной в вопросе 2 модели (**5 баллов**)?

Тема: Логистическая регрессия

1. Ознакомиться с материалами лекции № 2.

2. Установить необходимое программное обеспечение. При выполнении задания наверняка понадобятся **Python 3**, **NumPy** и **Matplotlib**.

3. Ознакомиться с содержимым папки с заданием, которая включает в себя файлы, представленные ниже.

main.py – «основной» модуль, необходимый для выполнения задания, который поможет выполнить его поэтапно. Настоящий программный код не требует какой-либо коррекции!

data.txt – база данных для выполнения задания.

plotData.py – модуль, содержащий функцию `plotData`, которая необходима для визуализации данных.

plotDecisionBoundary.py – модуль, содержащий функцию plotDecisionBoundary, которая необходима для визуализации данных с границей решения для заданного множества параметров модели логистической регрессии. Данный модуль не требует коррекции!

computeCost.py – модуль, содержащий функцию computeCost, которая необходима для вычисления значения стоимостной функции логистической регрессии.

gradientDescent.py – модуль, содержащий функцию gradientDescent, которая необходима для выполнения градиентного спуска с целью поиска параметров модели логистической регрессии.

featureNormalize.py – модуль, содержащий функцию featureNormalize, которая необходима для нормализации признаков. Данный модуль не требует коррекции!

sigmoid.py – модуль, содержащий функцию sigmoid, которая позволяет вычислить значение сигмоидной функции.

predict.py – модуль, содержащий функцию predict, которая необходима для предсказания метки класса (0 или 1).

4. Поэтапно выполнить задание, связанное с реализацией и исследованием логистической регрессии.

5. Ответить на вопросы, необходимые для составления отчета по данному практическому заданию. Отчет сдается на проверку в печатной или письменной форме в указанные сроки.

При выполнении задания требуется заполнить пустые места программного кода в блоках с комментарием «Ваш код здесь». Данную процедуру необходимо выполнить для следующих функций: plotData, computeCost, gradientDescent, sigmoid, predict.

1. При решении любой задачи с использованием инструментов машинного обучения важным является понимание структуры анализируемых данных и их визуализация в случае возможности. В настоящем задании предлагается использовать базу данных из файла **data.txt**. Данные представляют собой множество объектов, описываемых двумя признаками (оценка студента за первый экзамен и оценка студента за второй экзамен) и меткой (аттестован или не аттестован студент по итогам двух экзаменов). Необходимо обратить внимание на то, что база данных в настоящем задании размечена, а метка принимает дискретный набор из двух значений (0 – не аттестован, 1 – аттестован). Поэтому в рамках настоящего задания рассматривается решение задачи бинарной классификации, а не регрессии, как это было в практическом задании № 1. Завершите программный код в модуле **plotData.py**, который позволит выполнять визуализацию данных. Завершение модуля подразумевает под собой написание строчек программного кода, которые позволят вызвать функцию из соответствующего модуля в файле **main.py**, позволяя решить определенный кусок настоящего задания. Например, в данном случае завершенный программный код будет выглядеть так, как представлено на рис. 1.

После завершения каждого блока кода интерпретируйте файл **main.py** с целью проверки правильности работы соответствующей части задания. Результат визуализации данных с использованием функции plotData представлен на рис. 2. В случае успешной интерпретации программного кода разрешается перейти к следующему пункту задания.

2. Завершите программный код в модуле **computeCost.py**, который позволит вычислить значение стоимостной функции для логистической регрессии. Формулы, описывающие ее вычисление, представлены в лекции № 2. При выполнении данной части задания могут понадобиться функции из библиотеки **NumPy**, представленные ниже.

dot – позволяет вычислить матричное произведение для двумерных массивов и скалярное произведение для одномерных массивов (без комплексного сопряжения).

`sum` – позволяет вычислить сумму элементов вдоль определенной размерности двумерного массива и сумму всех элементов для одномерного массива.
`log` – позволяет вычислить натуральный логарифм от элементов массива.

```
import matplotlib.pyplot as plt
import numpy as np

def plotData(X, y):
    """
        Функция позволяет выполнить визуализацию данных с маркером
        + для положительных примеров и маркером o для отрицательных
        примеров. X - матрица объекты-признаки размера mx2,
        а y - вектор меток размера mx1, где m - размер базы данных
    """

    # ===================== Ваш код здесь =====================
    # Инструкция: визуализируйте положительные и отрицатедьные
    # примеры на двумерной плоскости, используя маркер + для
    # обозначения положительных примеров и маркер o для обозначения
    # отрицательных примеров

    plt.figure()                                                                    # создание нового окна
    pos = np.where(y == 1)[0]                                                       # извлечение индексов положительных примеров
    neg = np.where(y == 0)[0]                                                       # извлечение индексов отрицательных примеров

    plt.plot(X[pos, 0], X[pos, 1], '+', markersize = 7, markeredgecolor = 'black', markeredgewidth = 2)   # построение положительных примеров
    plt.plot(X[neg, 0], X[neg, 1], 'o', markersize = 7, markeredgecolor = 'black', markerfacecolor = 'yellow') # построение отрицательных примеров
    plt.grid()                                                                      # создание координатной сетки

    # ==========================================================
```

Рис. 1. Завершенный программный код для функции `plotData`

При заполнении программного кода в модуле **`computeCost.py`** потребуется вычисление значений сигмоидной функции. Реализуйте ее вычисление в модуле **`sigmoid.py`**. При реализации сигмоидной функции может понадобиться функция `exp` из библиотеки **NumPy**, которая позволяет вычислить значения экспоненциальной функции от элементов массива.

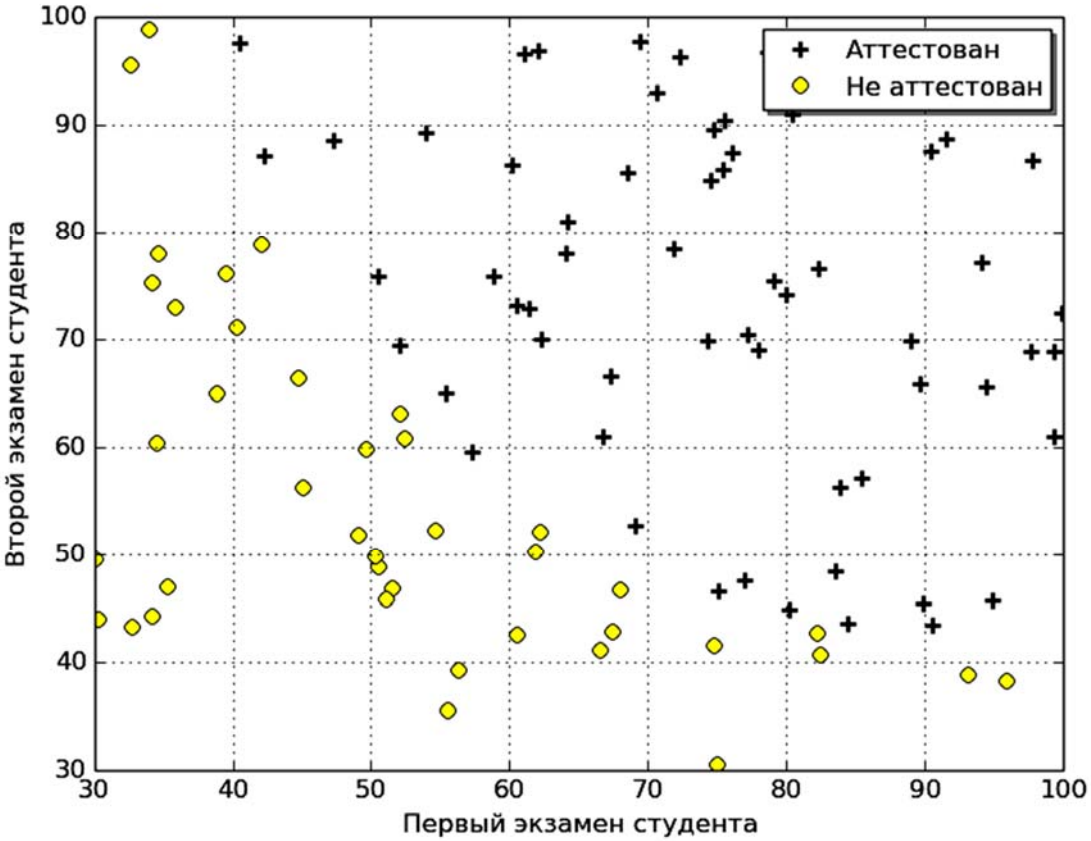


Рис. 2. Результат визуализации тренировочных данных

3. Завершите программный код в модуле **gradientDescent.py**, который позволит выполнить алгоритм градиентного спуска с целью обучения параметров модели логистической регрессии. Формулы, описывающие реализацию градиентного спуска для логистической регрессии, представлены в лекции № 2. При выполнении данной части задания могут понадобиться следующие функции из библиотеки **NumPy**: `dot` и `transpose`.

`transpose` – позволяет выполнить транспонирование массива. Для одномерного массива данная функция не оказывает никакого действия, а для двумерного массива использование функции соответствует обычному матричному транспонированию.

Так же совершенно будет необходима функция `sigmoid`, реализованная в модуле **sigmoid.py**.

Обратите внимание на то, что при обучении параметров модели логистической регрессии в файле **main.py** используется нормализация признаков, позволяющая выполнить качественную сходимость градиентного спуска к единственному в данном случае минимуму стоимостной функции. Нормализация признаков полностью реализована в модуле **featureNormalize.py**, который завершать не требуется.

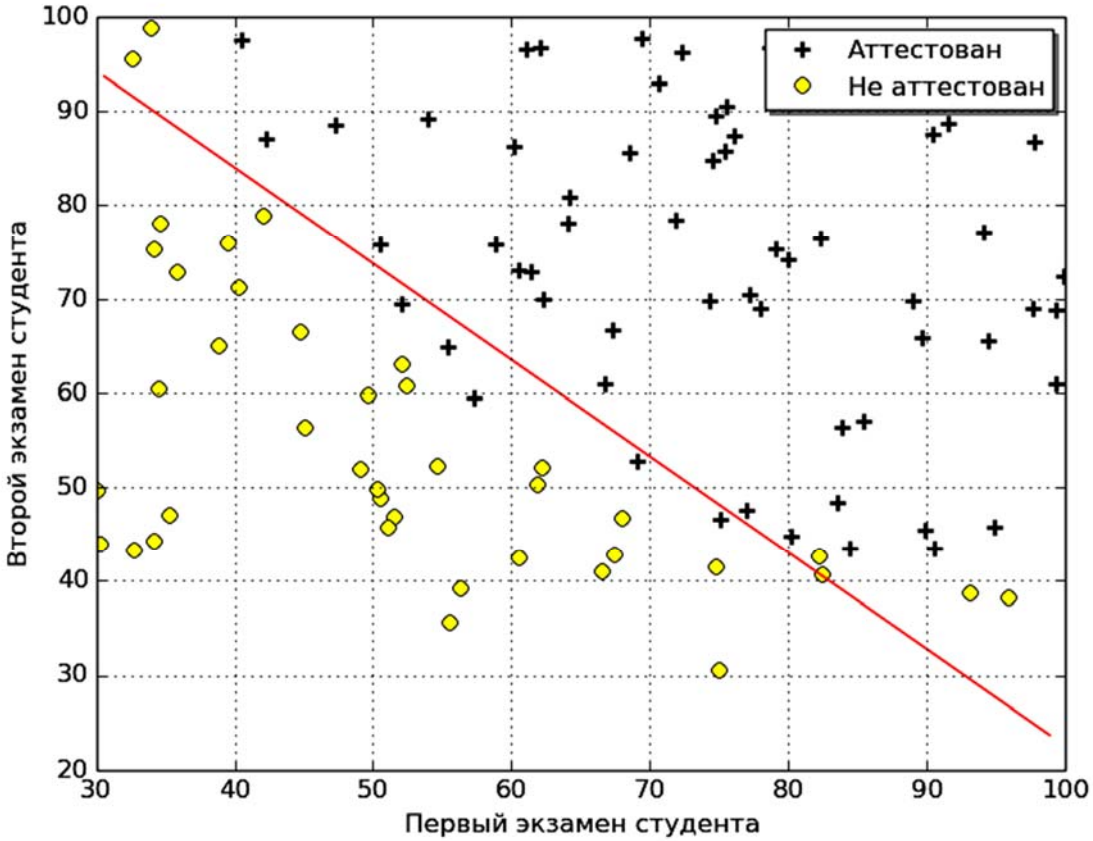


Рис. 3. Результат визуализации тренировочных данных с границей решения для логистической регрессии

После обучения параметров модели логистической регрессии с настройками градиентного спуска, заданными по умолчанию в файле **main.py**, должен получиться результат, представленный на рис. 3, на котором помимо тренировочных данных изображена найденная граница решения для модели на основе логистической регрессии. Построение границы решения полностью реализовано в функции `plotDecisionBoundary` модуля **plotDecisionBoundary.py**. Объекты, которые описываются точками на рис. 3, лежащими выше границы решения, будут отнесены алгоритмом к классу 1 (аттестован), иначе к классу 0 (не аттестован). Необходимо обратить внимание на то, что в процессе принятия

решения алгоритмом на тренировочной базе данных возникают ошибки. Иногда точки, которые относятся к классу 1, классифицируются, как точки класса 0 и наоборот.

Как и в практическом задании № 1, в данном случае возможно провести исследование сходимости градиентного спуска при различных настройках с использованием зависимости представленной на рис. 4.

4. Завершите программный код в модуле **`predict.py`**, который позволит выполнить предсказание метки класса для обученной модели логистической регрессии. В ходе предсказания порог классификатора необходимо выставить равным значению 0.5. При выполнении данной части задания могут понадобиться следующие функции из библиотеки **NumPy**: `dot` и `astype`.

`astype` – позволяет выполнить приведение элементов массива к определенному типу данных.

Так же совершенно будет необходима функция `sigmoid`, реализованная в модуле **`sigmoid.py`**.

5. После завершения предыдущих пунктов вычислите значение вероятности, с которой студент будет аттестован в случае, если его оценка за первый экзамен равна 45, а оценка за второй экзамен равна 85. Обратите внимание на то, что перед выполнением процедуры предсказания требуется провести нормализацию признаков на соответствующие им математическое ожидание и среднеквадратическое отклонение.

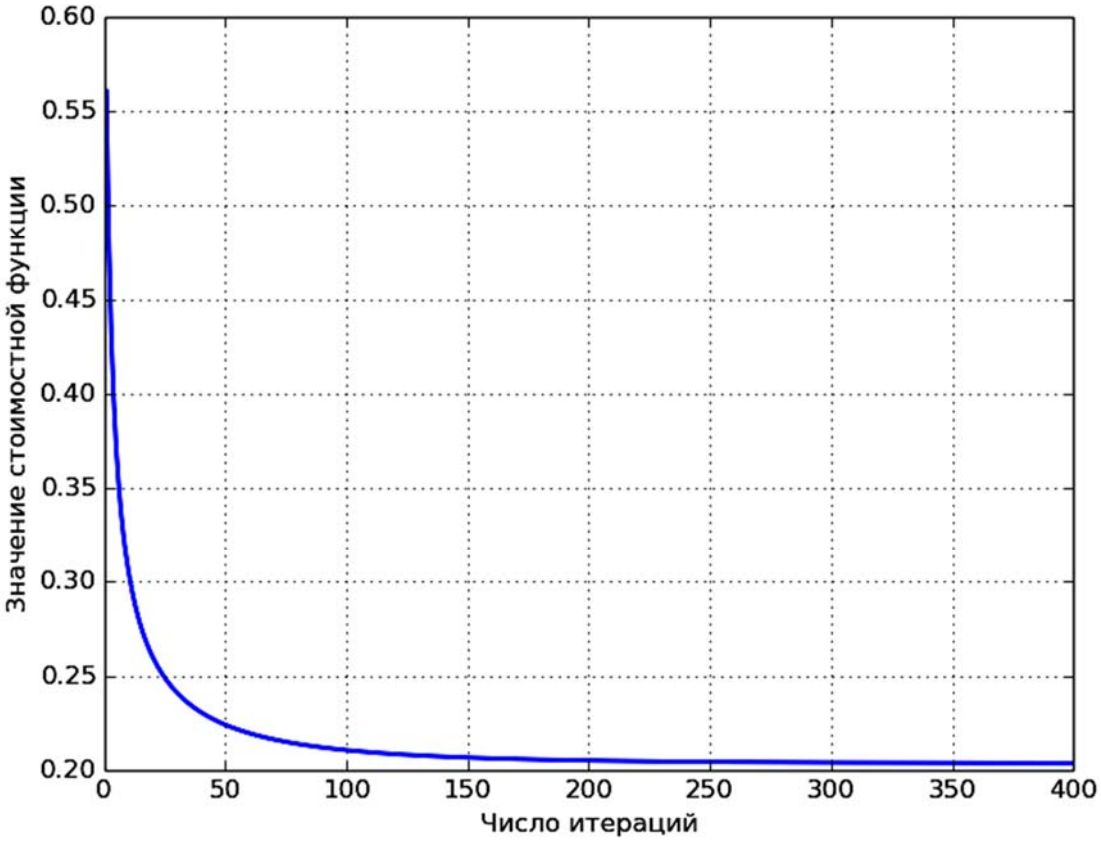


Рис. 4. Пример сходимости градиентного спуска для удачно подобранной скорости сходимости

6. Оцените долю правильных ответов обученной модели логистической регрессии. Необходимо обратить внимание на то, что в русскоязычной литературе доля правильных ответов иногда обозначается словом точность (от англ. accuracy). Последний перевод не всегда является удачным, так как существует другая метрика оценки качества работы

классификатора – precision, которая на русский язык так же переводится, как точность. Однако смысл accuracy и precision является абсолютно разным. **Accuracy** равна отношению числа ответов, для которых алгоритм выполнил предсказание метки класса верно, к общему числу объектов в базе данных. **Precision** равна отношению числа объектов, для которых алгоритм верно предсказал метку класса 1, к числу объектов, для которых алгоритм предсказал метку класса 1.

Используя файл `main.py` ответьте на следующие вопросы.

1. Чему равно значение стоимостной функции для случая, когда все параметры модели равны нулю (**35 баллов**)?

2. Чему равны значения параметров обученной модели логистической регрессии для случая, когда параметр сходимости равен 1, а число итераций градиентного спуска равно 400 (**40 баллов**)?

3. Чему равна вероятность аттестации студента в случае, если его оценка за первый экзамен равна 45, а оценка за второй экзамен равна 85 (**15 баллов**) для обученной в вопросе 2 модели?

4. Чему равна доля правильных ответов обученной в вопросе 2 модели логистической регрессии (**10 баллов**)?

5. Опираясь на материал лекции № 2, опишите возможное решение (решения), которые позволят увеличить долю правильных ответов классификатора на основе логистической регрессии применительно к рассматриваемой в настоящем задании базе данных (**20 баллов**). Настоящий вопрос является необязательным, но позволяет заработать дополнительные баллы!

Примерные варианты тестовых заданий

1. Говорят, что компьютерная программа способна к обучению из опыта E по отношению к некоторой задаче T и критерию качества работы P, если ее производительность на T, как мера P, улучшается с опытом E. Допустим, что алгоритм машинного обучения обучен (на множестве исторических данных о погоде) предсказывать погоду. Чем в этом случае является E?

а) Среди ответов нет правильного.

б) Вероятность правильно предсказывать погоду в будущем.

в) Процедура исследования алгоритмом большого количества исторических данных о погоде.

г) Задача предсказания погоды.

2. Допустим нам необходимо предсказать будет или нет дождь завтра в 17.00. Для этой цели мы желаем использовать алгоритм машинного обучения. Как в данном случае будет трактоваться задача машинного обучения?

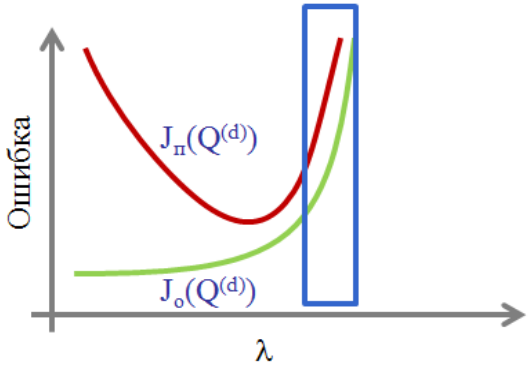
а) Задача классификации.

б) Задача регрессии.

3. В чем проблема алгоритма машинного обучения при выбранных параметрах регуляризации в области, отмеченной на рисунке?

а) Недообучение.

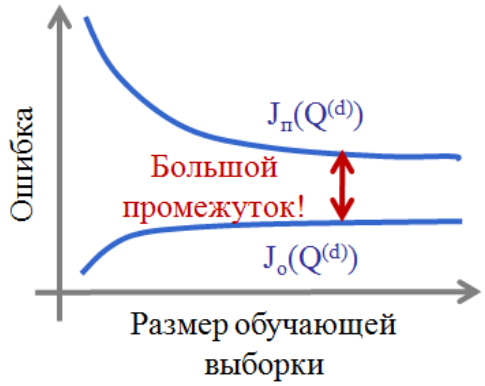
б) Переобучение.



4. После обучения классификатора получены следующие кривые обучения для обучающего и проверочного множеств. В чем проблема обученного алгоритма?

а) Недообучение.

б) Переобучение.



5. Какие действия можно предпринять в случае переобучения алгоритма на базе логистической регрессии?

а) Получить больше тренировочных примеров.

б) Попытаться уменьшить число свойств.

в) Попытаться получить дополнительные свойства.

г) Попытаться уменьшить параметр регуляризации.

д) Попытаться увеличить параметр регуляризации.

6. При решении задачи бинарной классификации для 1000 тестовых примеров получены результаты, представленные в таблице ниже. Чему равны точность (Precision) и полнота выборки (Recall) классификатора?

Действительный класс

Предсказанный класс	1	0
1	85	890
0	15	10

7. Анализ главных компонент и линейная регрессия – это одно и тоже?

а) Да.
б) Нет.

8. Каким свойствам удовлетворяет базис главных компонент?
а) Ортогональность.
б) Базисные вектора являются собственными вектора ковариационной матрицы данных.
в) Представление данных в базисе главных компонент является оптимальным в смысле среднеквадратической ошибки.
г) Все ответы неверны.

9. Как называются изображения лиц, формирующие базис главных компонент, рассчитанный для базы данных лиц?

10. Как называется алгоритм классификации данных, в котором неизвестный вектор признаков относится к тому классу, к отдельному эталонному образцу которого этот вектор наиболее близок.
а) Логистическая регрессия.
б) Нейронная сеть прямого распространения.
в) Метод ближайшего соседа.
г) Метод K-ближайших соседей.

2. Список вопросов и (или) заданий для проведения промежуточной аттестации

Список вопросов к зачету:
(проверка сформированности ПК-2, индикаторы ИД-ПК-2_1, ИД-ПК-2_3)
(Билет содержит два вопроса)

1. Что такое машинное обучение?
2. Примеры задач, решаемые в области машинного обучения (обучение с учителем и без учителя, стимулированное обучение, эволюционное обучение).
3. Обзор основных инструментов языка Python.
4. Общая постановка задачи регрессии.
5. Что такое гипотеза, параметры модели и стоимостная функция на примере задачи линейной регрессии?
6. Минимизация стоимостной функции (нормальные уравнения и численная оптимизация).
7. Метод градиентного спуска.
8. Масштабирование признаков и настройка скорости обучения алгоритма.
9. Общая постановка задачи классификации.
10. Логистическая регрессия и ее стоимостная функция.
11. Граница принятия решения.
12. Многоклассовая классификация на основе логистической регрессии (подходы «один против всех» и «один против одного»).
13. Проблема нелинейной классификации.
14. Биологические нейроны и мозг.
15. Модель нейрона и искусственные нейронные сети.
16. Нейронные сети прямого распространения.
17. Примеры реализации логических операций на основе искусственных нейронных сетей.
18. Классификация рукописных цифр.
19. Классификация объектов в сложных сценах.

20. Многоклассовая классификация для нейронных сетей.
21. Регуляризация и проблема переобучения.
22. Регуляризованная стоимостная функция для линейной и логистической регрессии, нейронной сети прямого распространения.
23. Алгоритм обратного распространения ошибки.
24. Градиентная проверка.
25. Проблема симметричности весов нейронной сети.
26. Автономное вождение (проект ALVINN).
27. Отладка алгоритмов машинного обучения.
28. Что такое диагностика?
29. Оценка работоспособности гипотезы.
30. Обучающее, проверочное и тестовое множества данных.
31. Алгоритм выбора модели.
32. Подбор параметра регуляризации.
33. Кривые обучения.
34. Анализ ошибок.
35. Метрики ошибки для ассиметричных классов.
36. Машинное обучение в задаче оптического распознавания символов (детектирование текста, сегментация символов, классификация символов).
37. Что такое скользящее окно?
38. Формирование большого количества данных для решения задачи машинного обучения.
39. Анализ производительности конвейерной системы.
40. Что такое кластеризация?
41. Разновидности алгоритмов кластеризации данных.
42. Алгоритм K-средних (стоимостная функция, случайная инициализация, выбор числа кластеров).
43. Примеры использования алгоритма K-средних для обработки цифровых изображений.
44. Задача сокращения размерности данных.
45. Общая формулировка анализа главных компонент и его реализация на практике.
46. Выбор числа главных компонент.
47. Рекомендации по применению анализа главных компонент.
48. Распознавание лиц с использованием анализа главных компонент (пространство лиц, собственные лица, структура алгоритма распознавания лиц).
49. Общее описание проблемы детектирования лиц.
50. Признаки, используемые в алгоритме детектирования лиц Виола/Джонса.
51. Каскад классификаторов.
52. Интегральные изображения.
53. Слабые классификаторы.
54. Комитет классификаторов и бустинг.
55. Данные для обучения каскада классификаторов и тестирование алгоритма Виола/Джонса.
56. Общая постановка задачи обучения на больших базах данных.
57. Стохастический градиентный спуск и минигрупповой градиентный спуск.
58. Онлайн-обучение.
59. Технология MapReduce и распараллеливание данных.
60. Проблема глубокого обучения.
61. Архитектура свёрточной нейронной сети (свёрточный слой и слой прореживания, полносвязные слои).
62. Обучение сверточной нейронной сети.
63. Предварительное обучение и автоэнкодеры.

64. Примеры использования свёрточных нейронных сетей для решения различных практических задач.

Критерии оценивания ответов на вопросы билета

Критерий	Пороговый уровень (на «удовлетворительно»)	Продвинутый уровень (на «хорошо»)	Высокий уровень (на «отлично»)
Соответствие ответа вопросу	Хотя бы частичное (*не относящееся к вопросу не подлежит проверке*)	Полное	Полное
Наличие примеров	Имеются отдельные примеры	Много примеров	Есть практически ко всем утверждениям
Содержание ответа	Понятийные вопросы изложены с классификациями, проблемные с постановкой проблемы и изложением различных точек зрения. Имеются ошибки или пробелы.	Ответ почти полный, без ошибок, не хватает отдельных элементов и тонкостей	Исчерпывающий полный ответ

3. Описание процедуры выставления зачета

В зависимости от уровня сформированности компетенции по окончании освоения дисциплины студенту выставляется оценка.

Оценка «зачтено» выставляется студенту, ответ которого на вопросы билета соответствует уровню не ниже порогового.

Оценка «не зачтено» выставляется студенту, ответ которого на вопросы билета соответствует уровню ниже порогового.

Приложение №2 к рабочей программе дисциплины «Машинное обучение»

Методические указания для студентов по освоению дисциплины

Одной из основных форм усвоения учебного материала по дисциплине является самостоятельная работа студента, причем в достаточно большом объеме. По всем темам предусмотрены задания самостоятельной работы, на которых происходит закрепление изученного материала и отработка навыков анализа и синтеза систем на базе методов машинного обучения.

Освоить вопросы дисциплины самостоятельно студенту достаточно сложно. Посещение всех предусмотренных лекционных занятий является совершенно необходимым. Без упорных и регулярных самостоятельных занятий в течение семестра сдать зачет практически невозможно.

Для самостоятельной работы рекомендуется использовать учебную литературу, указанную в разделе № 8 данной рабочей программы, и электронно-библиотечные системы, подписка на которые предоставлена через ЯрГУ, список и инструкцию по использованию которых можно найти по адресу: http://www.lib.uniyar.ac.ru/content/resource/net_res(1).php .