



МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ  
Федеральное государственное автономное образовательное учреждение  
высшего образования  
«Дальневосточный федеральный университет»  
(ДФУ)

**ШКОЛА ЦИФРОВОЙ ЭКОНОМИКИ**

СОГЛАСОВАНО  
Руководитель ОП

 Р.И. Дремлюга

« 24 » июня 2018 г.



**РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ**

**«НЕЙРОННЫЕ СЕТИ И ГЛУБОКОЕ ОБУЧЕНИЕ»**

направления 09.04.03 Прикладная информатика

Магистерская программа «Искусственный интеллект и большие данные»

Форма подготовки очная

курс 1 семестр 2  
лекции 18 час.  
практические занятия 18 час.  
лабораторные работы 18 час.  
всего часов аудиторной нагрузки 54 час.  
самостоятельная работа 54 час.  
в том числе на подготовку к экзамену – 36 час.  
контрольные работы программой не предусмотрены  
курсовая работа/проект – не предусмотрено  
зачет - не предусмотрено учебным планом  
экзамен – 2 семестр

Рабочая программа составлена в соответствии с требованиями федерального государственного образовательного стандарта высшего образования по направлению подготовки 09.04.03 – Прикладная информатика, утвержденного приказом Министерства образования и науки Российской Федерации от 30.10.2014 № 1404

Рабочая программа рассмотрена и утверждена на заседании Дирекции Школы цифровой экономики 24 июня 2018 г., протокол №2

Составитель(и): к.т.н. Д.А. Колесов, ст.пр. Кленин А.С.

**Оборотная сторона титульного листа РПД**

**I. Рабочая программа пересмотрена на заседании Дирекции Школы цифровой экономики:**

Протокол от «\_\_\_\_\_» \_\_\_\_\_ 20 г. № \_\_\_\_\_

Зам. директора по

учебной и воспитательной работе \_\_\_\_\_

(подпись)

(И.О. Фамилия)

**II. Рабочая программа пересмотрена на заседании Дирекции Школы цифровой экономики:**

Протокол от «\_\_\_\_\_» \_\_\_\_\_ 20 г. № \_\_\_\_\_

Зам. директора по

учебной и воспитательной работе \_\_\_\_\_

(подпись)

(И.О. Фамилия)

## АННОТАЦИЯ

### Б1.Б.02.03 Нейронные сети и глубинное обучение

Рабочая программа учебной дисциплины «Нейронные сети и глубинное обучение» предназначена для студентов, обучающихся по направлению подготовки для студентов 1 курса магистратуры, обучающихся по направлению подготовки 09.04.03 «Прикладная информатика» магистерской программы «Искусственный интеллект и большие данные».

Дисциплина «Нейронные сети и глубинное обучение» входит в базовую часть блока «Дисциплины (модули) Б.1» (Б1.Б.02.03) учебного плана подготовки магистров, модуль математических методов анализа данных.

Общая трудоемкость освоения дисциплины составляет 3 зачетных единиц, 108 часа. Дисциплина реализуется на 1 курсе в 1 семестре.

Семестр	Аудиторные занятия			Самостоятельная работа	Контроль	Форма контроля	Всего по дисциплине	
	Лекции	Лабораторные занятия	Практические занятия				Часы	з.е.
1 семестр	18	18	18	27	27	Экзамен	108	3

Предлагаемый курс посвящён методам «глубокого обучения» - нового поколения нейросетевых методов машинного обучения. В первую очередь курс направлен на формирования у студентов навыков решения прикладных задач при помощи глубоких нейронных сетей.

В результате освоения курса студент должен:

- изучить базовые понятия по следующим разделам курса: нейронные сети, их функциональные свойства, алгоритмы обучения, распознавания образов, прогнозирование временных рядов, оптимальные статистические решения;
- знать основные типы нейронных сетей и задач, для решения которых эти нейронные сети предназначены;
- иметь представление об основных сферах применения методов глубокого обучения, о концепциях и идеях, на которых основано многообразие современных нейросетевых технологий обработки информации;
- научиться использовать современные методы статистического анализа, прогнозирования и принятия решений на основе алгоритмов адаптации нейронных сетей.

- уметь использовать программные комплексы для создания и моделирования нейронподобных сетей, их применения в конкретных прикладных задачах.

- уметь разработать программную реализацию выбранного типа нейронной сети, произвести ее обучение и испытание.

Дисциплина логически и содержательно связана с такими курсами как «Математические методы машинного обучения», «Методы статистического анализа и прогнозирования», «Методы и системы обработки больших данных» и др.

Для успешного изучения дисциплины «Математические методы машинного обучения» необходимы знания базовой программы курса «Высшая математика» и основ программирования (желательно Python).

В результате данной дисциплины у обучающихся формируются следующие общекультурные, общепрофессиональные и профессиональные компетенции (элементы компетенций):

Код и формулировка компетенции	Этапы формирования компетенции	
ОК-1 – способность к абстрактному мышлению, анализу, синтезу	Знает	о новых методах исследований и необходимости их изучения; основные особенности и методологические основы научного метода познания и творчества, задачи и инструментарий математического моделирования
	Умеет	самостоятельно обучаться новым методам исследования; применять современные методы создания и анализа моделей, позволяющих прогнозировать свойства и поведение объектов профессиональной деятельности
	Владеет	навыками самостоятельного обучения новым методам исследования; навыками изменения научного и научно-производственного профиля своей профессиональной деятельности современными методами создания и анализа моделей, позволяющих прогнозировать свойства и поведение объектов профессиональной деятельности
ОПК-4 – способность исследовать закономерности становления и развития информационного общества в конкретной прикладной области	Знает	основные положения современных теорий информационного общества; предпосылки и факторы формирования информационного общества; содержание, объекты и субъекты информационного общества, критерии эффективности его функционирования

	Умеет	оценивать и анализировать различные точки зрения на особенности информационного общества и пути его развития; понимать и правильно использовать терминологию современных теорий информационного общества; исследовать закономерности развития и использования информационно-коммуникационных технологий в экономике, технике и других прикладных областях
	Владеет	оценивать и анализировать различные точки зрения на особенности информационного общества и пути его развития; понимать и правильно использовать терминологию современных теорий информационного общества; исследовать закономерности развития и использования информационно-коммуникационных технологий в экономике, технике и других прикладных областях
ПК-13 – способность проектировать информационные процессы и системы с использованием инновационных инструментальных средств, адаптировать современные ИКТ к задачам прикладных ИС	Знает	основные математические методы анализа данных и методы компьютерного моделирования
	Умеет	анализировать данные и оценивать требуемые знания для решения нестандартных задач с использованием инновационных инструментальных средств
	Владеет	математическими методами и методами компьютерного моделирования для анализа данных и оценки требуемых знаний для решения нестандартных задач

# СТРУКТУРА И СОДЕРЖАНИЕ ТЕОРЕТИЧЕСКОЙ ЧАСТИ КУРСА

## Лекционные занятия (18 часов)

### Тема 1. Нейронные сети (2 часа)

Перцептрон, многослойный перцептрон.

Функции активации.

Методы оптимизации.

Метод обратного распространения ошибки.

Теория сложности и машинное обучение.

Probably Approximately Correct learning.

### Тема 2. Тематическое моделирование (2 часа)

Задача [тематического моделирования](#) коллекции текстовых документов.

[Вероятностный латентный семантический анализ](#) PLSA. [Метод максимума правдоподобия](#). [EM-алгоритм](#). Элементарная интерпретация EM-алгоритма.

[Латентное размещение Дирихле](#) LDA. [Метод максимума апостериорной вероятности](#). Сглаженная частотная оценка условной вероятности.

Небайесовская интерпретация LDA и её преимущества. Регуляризаторы разреживания, сглаживания, частичного обучения.

Аддитивная регуляризация тематических моделей. Регуляризованный EM-алгоритм, теорема о стационарной точке (применение условий Каруша–Куна–Таккера).

Рациональный EM-алгоритм. Онлайнный EM-алгоритм и его распараллеливание.

Мультимодальная тематическая модель.

Регуляризаторы классификации и регрессии.

Регуляризаторы декоррелирования и отбора тем.

Внутренние и внешние критерии качества тематических моделей.

### Тема 3. Обучение с подкреплением (2 часа)

Задача о многоруком бандите.

Жадные и эpsilon-жадные стратегии.

Метод UCB (upper confidence bound). Стратегия Softmax.

Среда для экспериментов.

Адаптивные стратегии на основе скользящих средних.

Метод сравнения с подкреплением.

Метод преследования.

Постановка задачи в случае, когда агент влияет на среду. Ценность состояния среды. Ценность действия.

Жадные стратегии максимизации ценности. Уравнения оптимальности Беллмана.

Метод временных разностей TD. Метод Q-обучения.

Градиентная оптимизация стратегии (policy gradient). Связь с максимизацией log-правдоподобия.

Постановка задачи при наличии информации о среде в случае выбора действия. Контекстный многорукий бандит.

Линейная регрессионная модель с верхней доверительной оценкой LinUCB.

Оценивание новой стратегии по большим историческим данным.

#### **Тема 4. Активное обучение (2 часа)**

Постановка задачи машинного обучения. Основные стратегии: отбор объектов из выборки и из потока, синтез объектов. Почему активное обучение быстрее пассивного. Сокращение пространства решений. Взвешивание по плотности.

Оценивание качества активного обучения.

Синтез объектов по критерию сокращения дисперсии.

Применение обучения с подкреплением для активного обучения.

Активное томпсоновское сэмплирование.

#### **Тема 5. Сверточные нейронные сети. Рекуррентные нейронные сети (2 часа)**

Сверточные нейронные сети (CNN). Сверточный нейрон. Pooling нейрон. Выборка размеченных изображений ImageNet.

Идея обобщения CNN на любые структурированные данные.

Рекуррентные нейронные сети (RNN). Обучение рекуррентных сетей: Backpropagation Through Time (BPTT).

Сети долгой кратковременной памяти (Long short-term memory, LSTM).

Автокодировщики. Векторные представления дискретных данных.

## **Тема 6. Генеративные модели (2 часа)**

Моделирование случайных данных

Неслучайные модельные данные

Ранговая классификация, ОС-SVM.

Попарный подход: RankingSVM, RankNet, LambdaRank.

## **Тема 7. Кластеризация и частичное обучение (2 часа)**

Постановка задачи [кластеризации](#). Примеры прикладных задач. Типы кластерных структур.

Постановка задачи Semisupervised Learning, примеры приложений.

Оптимизационные постановки задач кластеризации и частичного обучения.

[Алгоритм k-средних](#) и [EM-алгоритм](#) для разделения гауссовской смеси.

[Графовые алгоритмы кластеризации](#). Выделение связных компонент. [Кратчайший незамкнутый путь](#).

[Алгоритм ФОРЭЛ](#).

[Алгоритм DBSCAN](#).

[Агломеративная кластеризация](#), [Алгоритм Ланса-Вильямса](#) и его частные случаи.

Алгоритм построения [дендрограммы](#). Определение числа кластеров.

Свойства сжатия/растяжения, монотонности и редуktivности.

Псевдокод редуktivной версии алгоритма.

Простые эвристические методы частичного обучения: self-training, co-training, co-learning.

Трансдуктивный метод опорных векторов TSVM.

Алгоритм Expectation-Regularization на основе многоклассовой регуляризованной логистической регрессии.

## **Тема 8. Основы статистики для машинного обучения (2 часа)**

Статистический критерий информативности.

[Точный Тест Фишера.](#)

Сравнение областей эвристических и статистических закономерностей.

Асимптотическая эквивалентность статистического и энтропийного критерия информативности.

Разнообразие критериев информативности в  $(p,n)$ -пространстве.

## **Тема 9. Автоматическое обучение машин (AutoML) (2 часа)**

Автоматическая подготовка данных и сбор и сохранение данных (из сырых данных и разнообразных форматов)

Автоматическое конструирование признаков

Автоматический выбор модели

Оптимизация гиперпараметров алгоритма обучения и характеристики

Автоматический выбор каналов по времени, памяти и ограничений сложности

Автоматический выбор метрик оценки / процедур валидации

Автоматическая проверка задач

Автоматический анализ полученных результатов

Пользовательские результаты и визуализация для автоматического обучения машин

# 1. СТРУКТУРА И СОДЕРЖАНИЕ ПРАКТИЧЕСКОЙ ЧАСТИ КУРСА

## Лабораторные работы (18 часов)

Наименование темы и содержание	Часы
<b>Лабораторная работа №1. Методы активного обучения.</b> Сэмплирование по неуверенности. Сэмплирование по несогласию в комитете. Сэмплирование по ожидаемому изменению модели. Сэмплирование по ожидаемому сокращению ошибки. Введение изучающих действий в стратегию активного обучения. Алгоритмы $\epsilon$ -active и EG-active.	2
<b>Лабораторная работа №2. Методы селекции.</b> Селекция: пропорционально качеству, универсальная выборка (stochastic universal sampling), с наследием (reward-based), турнир. Стратегия элитизма.	2
<b>Лабораторная работа №3. Методы кроссовера.</b> Двух и многоточечный кроссовер, равномерный кроссовер (по подмножествам), для перестановок. Мутация. Влияние на скорость обучения.	2
<b>Лабораторная работа №4. Управление популяцией.</b> Сегрегация, старение, распараллеливание. Генетическое программирование.	2
<b>Лабораторная работа №5. Наивный классификатор.</b> Оценка плотности распределения для числовых признаков. Алгоритмические оптимизации. Алгоритм EM.	2
<b>Лабораторная работа №6. Оптимизации с ограничениями</b> Двойственная задача Лагранжа. Условия Каруша-Куна-Такера. Функция Лагранжа для линейного SVM. Опорный вектор. Типы опорных векторов. Kernel trick. Полиномиальное ядро. Радиально-базисное ядро (RBF).	2
<b>Лабораторная работа №7. Алгоритм преобразования Хафа</b> Написание алгоритма преобразования Хафа для прямых, кругов. Добавление оптимизации, требующих вычисления градиента на изображении.	2
<b>Лабораторная работа №8. Алгоритмы нейронной сети (обработка изображений)</b> Использование классических методов обработки изображений совместно с нейросетевым подходом (свертки, фильтры, описания текстур и пр.).	2
<b>Лабораторная работа №9. Алгоритмы нейронной сети</b> Реализация “ванильной” рекуррентной нейронной сети в TensorFlow. Реализация сети LSTM в TensorFlow.	2

## Практические занятия (18 часов)

Наименование темы и содержание	Часы
<b>Практическая работа №1.</b>	3

Использование классических методов обработки изображений совместно с нейросетевым подходом (свертки, фильтры, описания текстур и пр.).	
<b>Практическая работа №2.</b> Построение сверточной сети средствами TensorFlow.	3
<b>Практическое задание №3.</b> Анализ последовательностей изображений в Google EarthEngine. Реализация классификатора на базе многослойного перцептрона для обработки последовательности изображений.	3
<b>Практическое задание №4.</b> Частотный анализ, представление bag-of-words, TF-IDF и его варианты. N-граммы, byte-pair encoding.	3
<b>Практическое задание №5.</b> Построение многослойного перцептрона для классификации изображений. Построение небольшой архитектуры сверточной нейронной сети для классификации. Построение одной из известных архитектур нейронных сетей (AlexNet, VGG, ResNet, Inception).	4
<b>Практическое задание №6.</b> Построение сверточной нейронной сети для задачи сегментации. Построение одной из известных архитектур (UNet).	2

### **III. УЧЕБНО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ САМОСТОЯТЕЛЬНОЙ РАБОТЫ ОБУЧАЮЩИХСЯ**

Учебно-методическое обеспечение самостоятельной работы обучающихся по дисциплине «Нейронные сети и глубокое обучение» представлено в Приложении 1 и включает в себя:

- план-график выполнения самостоятельной работы по дисциплине, в том числе примерные нормы времени на выполнение по каждому заданию;
- характеристика заданий для самостоятельной работы обучающихся и методические рекомендации по их выполнению;
- требования к представлению и оформлению результатов самостоятельной работы;
- критерии оценки выполнения самостоятельной работы.

### **IV. КОНТРОЛЬ ДОСТИЖЕНИЯ ЦЕЛЕЙ КУРСА**

Изучение дисциплины «Нейронные сети и глубокое обучение» предусматривает:

- изучение теоретического материала в соответствии с программой, с использованием материала из списка литературы и информационно-методического обеспечения дисциплины;
- выполнение лабораторных работ;
- выполнение практических работ;

- *текущий контроль* – учет посещения студентами занятий в течение периода обучения и оценка своевременности и качества изучения студентами темы и выполнения лабораторных работ.

- *итоговый контроль* – выведение итоговой оценки за семестр по результатам рейтинга без обязательной сдачи экзамена.

№ п/п	Контролируемые разделы / темы дисциплины	Коды и этапы формирования компетенций		Оценочные средства	
				текущий контроль	промежуточная аттестация
1	Нейронные сети. Основные понятия.	ОК-1 ОПК-4 ПК-13	знает	УО	Экзамен
			умеет	ТС	
			владеет	ПР-4	
2	Тематическое моделирование	ОК-1 ОПК-4 ПК-13	знает	УО-1	Экзамен
			умеет	ТС	
			владеет	ПР-11	
3	Обучение с подкреплением	ОК-1 ОПК-4 ПК-13	знает	УО-1	Экзамен
			умеет	ТС	
			владеет	ПР-11	
4	Активное обучение	ОК-1 ОПК-4 ПК-13	знает	УО-1	Экзамен
			умеет	ТС	
			владеет	ПР-4	
5	Сверточные нейронные сети. Рекуррентные нейронные сети	ОК-1 ОПК-4 ПК-13	знает	УО-1	Экзамен
			умеет	ТС	Экзамен
			владеет	ПР-4	Экзамен
6	Генеративные модели	ОК-1 ОПК-4 ПК-13	знает	УО-2	Экзамен
			умеет	ТС	Экзамен
			владеет	ПР-11	Экзамен
7	Кластеризация и частичное обучение	ОК-1 ОПК-4 ПК-13	знает	УО-2	Экзамен
			умеет	ТС	Экзамен
			владеет	ПР	Экзамен

8	Основы статистики для машинного обучения	ОК-1 ОПК-4 ПК-13	знает	УО-1	Экзамен
			умеет	ТС	Экзамен
			владеет	ПР-4	Экзамен
9	Автоматическое обучение машин	ОК-1 ОПК-4 ПК-13	знает	УО-1	Экзамен
			умеет	ТС	Экзамен
			владеет	ПР-11	Экзамен

1. устный опрос (УО): собеседование (УО-1), коллоквиум (УО-2); итоговая презентация (УО-3); круглый стол (УО-4);
2. технические средства контроля (ТС);
3. письменные работы (ПР): тесты (ПР-1), контрольные работы (ПР-2), эссе (ПР-3), рефераты (ПР-4), курсовые работы (ПР-5), научно-учебные отчеты по практикам (ПР-6), конспект (ПР-7), проект (ПР-9). Разноуровневые задачи и задания (ПР-11) и т.п.

Типовые контрольные задания, методические материалы, определяющие процедуры оценивания знаний, умений и навыков и (или) опыта деятельности, а также критерии и показатели, необходимые для оценки знаний, умений, навыков и характеризующие этапы формирования компетенций в процессе освоения образовательной программы, представлены в Приложении 2.

## **V. СПИСОК УЧЕБНОЙ ЛИТЕРАТУРЫ И ИНФОРМАЦИОННО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ**

### **Основная литература**

*(электронные и печатные издания)*

1. Горожанина Е.И. Нейронные сети [Электронный ресурс]: учебное пособие/ Горожанина Е.И.— Электрон. текстовые данные.— Самара: Поволжский государственный университет телекоммуникаций и информатики, 2017.— 84 с.— Режим доступа: <http://www.iprbookshop.ru/75391.html>.— ЭБС «IPRbooks»
2. Седов В.А. Введение в нейронные сети [Электронный ресурс]: методические указания к лабораторным работам по дисциплине «Нейроинформатика» для студентов специальности 09.03.02 «Информационные системы и технологии»/ Седов В.А., Седова Н.А.— Электрон. текстовые данные.— Саратов: Ай Пи Эр Медиа, 2018.— 30 с.— Режим доступа: <http://www.iprbookshop.ru/69319.html>.— ЭБС «IPRbooks»
3. Барский А.Б. Введение в нейронные сети [Электронный ресурс]/ Барский А.Б.— Электрон. текстовые данные.— М.: Интернет-Университет Информационных Технологий (ИНТУИТ), 2016.— 358 с.— Режим доступа: <http://www.iprbookshop.ru/52144.html>.— ЭБС «IPRbooks»
4. Гудфеллоу, Я. Глубокое обучение [Электронный ресурс] / Я. Гудфеллоу, И. Бенджио, А. Курвилль ; пер. с англ. А. А. Слинкина. — Электрон. дан. — Москва : ДМК Пресс, 2018. — 652 с. — Режим доступа: <https://e.lanbook.com/book/107901>. — Загл. с экрана.
5. Неделько В.М. Основы статистических методов машинного обучения [Электронный ресурс]: учебное пособие/ Неделько В.М.— Электрон. текстовые данные. — Новосибирск: Новосибирский государственный технический университет, 2010. — 72 с.— Режим доступа: <http://www.iprbookshop.ru/45418.html>. — ЭБС «IPRbooks»

6. Коэльо, Л.П. Построение систем машинного обучения на языке Python [Электронный ресурс] / Л.П. Коэльо, В. Ричарт; пер. с англ. Слинкин А. А.. — Электрон. дан. — Москва: ДМК Пресс, 2016. — 302 с. — Режим доступа: <https://e.lanbook.com/book/82818>. — Загл. с экрана.
7. Флах, П. Машинное обучение. Наука и искусство построения алгоритмов, которые извлекают знания из данных [Электронный ресурс] / П. Флах. — Электрон. дан. — Москва: ДМК Пресс, 2015. — 400 с. — Режим доступа: <https://e.lanbook.com/book/69955>. — Загл. с экрана.
8. Рашка, С. Python и машинное обучение: крайне необходимое пособие по новейшей предсказательной аналитике, обязательное для более глубокого понимания методологии машинного обучения [Электронный ресурс]: руководство / С. Рашка; пер. с англ. Логунова А.В. — Электрон. дан. — Москва: ДМК Пресс, 2017. — 418 с. — Режим доступа: <https://e.lanbook.com/book/100905>. — Загл. с экрана.
9. Шарден, Б. Крупномасштабное машинное обучение вместе с Python [Электронный ресурс]: учебное пособие / Б. Шарден, Л. Массарон, А. Боскетти; пер. с англ. А. В. Логунова. — Электрон. дан. — Москва: ДМК Пресс, 2018. — 358 с. — Режим доступа: <https://e.lanbook.com/book/105836>. — Загл. с экрана.

### **Дополнительная литература**

*(печатные и электронные издания)*

1. Нейронные сети : основы теории / А. И. Галушкин. Москва : Горячая линия - Телеком, 2010. - 496 с. -Г 168 32.818.э
2. Кук, Д. Машинное обучение с использованием библиотеки H2O [Электронный ресурс] / Д. Кук; пер. с англ. Огурцова А.Б.. — Электрон. дан. — Москва: ДМК Пресс, 2018. — 250 с. — Режим доступа: <https://e.lanbook.com/book/97353>. — Загл. с экрана.
3. Информационные аналитические системы [Электронный ресурс]: учебник / Т. В. Алексеева, Ю. В. Амириди, В. В. Дик и др.; под ред. В. В.

Дика. - М.: МФПУ Синергия, 2013. - 384 с. - (Университетская серия). - ISBN 978-5-4257-0092-6,  
<http://www.znaniyum.com/bookread.php?book=451186>

4. Домингос, П. Верховный алгоритм: как машинное обучение изменит наш мир [Электронный ресурс] Москва: Манн, Иванов и Фербер, 2016. 336 с. <https://e.lanbook.com/book/91645>.
5. Гаврилова, И.В. Основы искусственного интеллекта [Электронный ресурс]: учеб. пособие / И.В. Гаврилова, О.Е. Масленникова. Москва: ФЛИНТА, 2013. 282 с. <https://e.lanbook.com/book/44749>.
6. Ясницкий, Л.Н. Интеллектуальные системы [Электронный ресурс]: учеб. пособ./ Москва : Издательство 'Лаборатория знаний', 2016. 224 с. Режим доступа: <https://e.lanbook.com/book/90254>.

### **Перечень ресурсов**

#### **информационно-телекоммуникационной сети «Интернет»**

1. Байесовские\_методы\_машинного\_обучения\_(курс\_лекций)\_/\_2017 Д.П. Ветров - [http://www.machinelearning.ru/wiki/index.php?title=Байесовские\\_методы\\_машинного\\_обучения\\_\(курс\\_лекций\)\\_/\\_2017\\_Д.П.Ветров](http://www.machinelearning.ru/wiki/index.php?title=Байесовские_методы_машинного_обучения_(курс_лекций)_/_2017_Д.П.Ветров)
2. Машинное обучение (курс лекций, Н.Ю. Золотых) - <http://www.uic.unn.ru/~zny/ml/>
3. Машинное\_обучение\_(курс\_лекций\_С.К.Воронцов). - [http://www.machinelearning.ru/wiki/index.php?title=Машинное\\_обучение\\_\(курс\\_лекций%2C\\_К.В.Воронцов\)](http://www.machinelearning.ru/wiki/index.php?title=Машинное_обучение_(курс_лекций%2C_К.В.Воронцов))
4. [Курс «Введение в машинное обучение», К.В.Воронцов \(ВШЭ и Яндекс\).Хабр об этом курсе.](#)
5. [Специализация «Машинное обучение и анализ данных» \(МФТИ и Яндекс\). Хабр об этом курсе.](#)
6. [Машинное обучение \(семинары,ФУПМ МФТИ\)](#)
7. [Машинное обучение \(семинары, ВМК МГУ\)](#)
8. [Машинное обучение \(курс лекций, Н.Ю.Золотых\)](#)
9. [Машинное обучение \(курс лекций, СГАУ, С.Лисицын\)](#)

## VI. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ

Для эффективного изучения теоретической части дисциплины необходимо:

- построить работу по освоению дисциплины в порядке, отвечающем изучению основных этапов, согласно приведенным темам лекционного материала;
- систематически проверять свои знания по контрольным вопросам и тестам;
- усвоить содержание ключевых понятий;
- активно работать с основной и дополнительной литературой по соответствующим темам;
- регулярно консультироваться с преподавателем, ведущим изучаемую дисциплину.

Для эффективного изучения практической части дисциплины настоятельно рекомендуется:

систематически выполнять подготовку к лабораторным работам по предложенным преподавателем темам;

своевременно выполнять лабораторные работы.

Варианты лабораторных работ подобраны так, что их разбор и решение способствуют пониманию теоретических положений, излагаемых лектором. Задания предлагаются по мере изучения теоретических разделов дисциплины. Студент должен ответить на любой вопрос преподавателя, касающийся выполнения лабораторных работ и контрольные вопросы по изучаемой теме.

## VII. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

<p><b>Компьютерный класс:</b> Проектор DLP, 3000 ANSI Lm, WXGA 1280x800, 2000:1 EW330U Mitsubishi,; Системный блок с монитором. Процессор: Intel I5-8600k 3.6Ghz, оперативная память: 32gb, жесткий диск: 1ТБ, графический ускоритель: Nvidia GTX 1080 Беспроводные ЛВС для обучающихся обеспечены системой на базе точек доступа 802.11a/b/g/n 2x2 MIMO(2SS).</p> <p><b>Специализированное ПО:</b> Visual Studio 2019, Anaconda, Tensorflow, Keras</p>	<p>690922, Приморский край, г. Владивосток, о. Русский, п. Аякс, 10, г. Владивосток, о. Русский, п. Аякс, корпус G, ауд. G468</p>
---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-----------------------------------------------------------------------------------------------------------------------------------------------------------



МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ  
Федеральное государственное автономное образовательное учреждение  
высшего образования  
«Дальневосточный федеральный университет»  
(ДВФУ)

---

**ШКОЛА ЦИФРОВОЙ ЭКОНОМИКИ**

**УЧЕБНО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ  
САМОСТОЯТЕЛЬНОЙ РАБОТЫ ОБУЧАЮЩИХСЯ**  
по дисциплине «Нейронные сети и глубокое обучение»  
Направление подготовки – 09.04.03 Прикладная информатика  
магистерская программа «Искусственный интеллект и большие данные»  
**Форма подготовки очная**

**Владивосток  
2018**

**План-график выполнения самостоятельной работы по дисциплине**

№ п/п	Дата/сроки выполнения	Вид самостоятельной работы	Примерные нормы времени на выполнение	Форма контроля
<b>2 семестр</b>			<b>54</b>	
1	Февраль-май	Проработка лекционного материала по конспектам и учебной литературе	10	Собеседование, опрос
2	Февраль-май	Подготовка и выполнение практических заданий №1 - 6	34	Письменный отчет
3	Май	Выполнение и подготовка к защите лабораторных работ №1 - 9		Защита лабораторных работ
4	Апрель	Самостоятельная проработка тем		Семинар
5	Июнь	Подготовка к экзамену	10	Экзамен
<b>Итого</b>			<b>180</b>	

**Перечень тем для самостоятельного изучения по дисциплине**

**Тема 1. Метрики**

Примеси Джини (Gini impurity), добавленная информация (information gain). Деревья регрессии. Метрика вариации. Непрерывные признаки. Использование главных компонент вместо признаков. Сокращение дерева (pruning). Метрики, понятие центроида и представителя класса. Центроидные алгоритмы: k-means, k-medoid. Алгоритмы, основанные на плотности: DBSCAN, OPTICS. Алгоритмы, основанные на распределении: сумма гауссиан. Нечеткая кластеризация, алгоритм c-means. Метрики качества: leave-one-out, силуэт, индекс Дэвиса-Болдина (Davies-Bouldin), индекс Данна (Dunn).

**Тема 2. Последовательности изображений**

Анализ последовательностей изображений в Google EarthEngine. Классификатора на базе многослойного перцептрона для обработки последовательности изображений. Рекуррентные нейронные сети.

**Тема 3. Алгоритмы компьютерного зрения**

Классические алгоритмы обработки изображений.

**Тема 4. Алгоритмы машинного обучения**

Сверточные нейронные сети. Классификация изображений. Сегментация. Локализация.

### *Вопросы к темам для самостоятельного изучения*

1. Задачи обработки текста: извлечение, поиск, классификация (тематическая, эмоциональная), перевод
2. Разбиение на слова, пунктуация, лексический и морфологический анализ
3. Определение частей речи, имён, основ слов
4. Частотный анализ, представление bag-of-words, TF-IDF и его варианты
5. N-граммы, byte-pair encoding.
6. Векторные представления, семантическая интерпретация алгебраических операций
7. Унитарный код (One-hot encoding).
8. Алгоритмы Word2Vec и FastText.
9. Алгоритм GloVe\*.
10. Постановка задачи, причины и цели снижения размерности.
11. Выбор и извлечение признаков.
12. Подходы к выбору признаков: filtering, wrapping, embedding.
13. Расстояние между распределениями. Расстояние Кульбака-Лейблера. Взаимная информация.
14. Алгоритмы выбора признаков: на основе корреляции (CFS), взаимной информации, Relief.
15. Метод главных компонент (PCA).
16. Нелинейные обобщения метода главных компонент. Kernel PCA.\*
17. Неотрицательное матричное разложение (NMF).\*
18. Стохастическое вложение соседей с t-распределением (t-SNE).

### **Вопросы для самостоятельной работы по дисциплине**

1. Дайте определение объекта, образа и прецедента.
2. Приведите структурную схему системы распознавания образов.
3. Охарактеризуйте 3 способа минимизации среднего риска.
4. Дайте определение функционала риска.
5. Дайте определение функционала эмпирического риска.

6. Охарактеризуйте принцип минимизации эмпирического риска.
7. Дайте определение VC-измерения.
8. Охарактеризуйте понятие минимизации структурного риска.
9. Охарактеризуйте понятие вероятностно-корректной в смысле аппроксимации модели обучения.
10. Охарактеризуйте понятие байесовского классификатора.
11. Приведите структурные схемы байесовского классификатора на основе отношения правдоподобия и его логарифма.
12. Охарактеризуйте байесовский классификатор для Гауссовского распределения.
13. В чем заключаются сходство и различие перцептрона и байесовского классификатора при решении задач классификации объектов.
14. Оптимальная гиперплоскость для линейно-разделимых образов.
15. Квадратичная оптимизация и поиск оптимальной гиперплоскости. Применение множителей Лагранжа.
16. Статистические свойства оптимальной гиперплоскости для линейно-разделимых образов.
17. Оптимальная гиперплоскость для неразделимых образов. Фиктивные переменные и множители Лагранжа.
18. В чем заключается идея машины опорных векторов для решения задачи распознавания образов?
19. Охарактеризуйте понятие ядра скалярного произведения.
20. Сформулируйте теорему Мерсера. Собственные функции и собственные значения.
21. Сформулируйте двойственную задачу условной оптимизации для машины опорных векторов.
22. Оптимальная селекция признаков.
23. Оптимальная селекция признаков на основе нейронной сети.

**Рекомендации по самостоятельной работе студентов**

Самостоятельная работа студента, безусловно – один из важнейших этапов в подготовке магистров. Она приобщает студентов к исследовательской работе, обогащает опытом и знаниями, необходимыми для дальнейшего их становления как специалистов, прививает навыки работы с литературой.

Цель самостоятельной работы – систематизация, закрепление и расширение теоретических и практических знаний с использованием современных информационных технологий и литературных источников. Данная цель может быть достигнута при решении следующего круга задач:

- изучение лекционного материала;
- изучение дополнительных источников информации;
- выполнение лабораторных работ;
- выполнение практических заданий.

Теоретическое обучение предполагает самостоятельную работу с литературными источниками. Вопросы, вынесенные на самостоятельное изучение, дополняют сведения, полученные на лекциях, и также являются актуальными для будущей специальности. Некоторые из вопросов традиционно изучаются в рамках других дисциплин, поэтому они вынесены на самостоятельное изучение. В этом случае цель самостоятельного изучения заключается в том, чтобы студент получил на данном этапе общее представление о вопросе. Форма отчетности по проделанной работе – включение этих вопросов в экзаменационные билеты. Другая часть самостоятельной работы сводится к подготовке и защите в течение семестра лабораторных работ.

Руководство и контроль за самостоятельной работой студента осуществляется в форме индивидуальных консультаций.

При затруднении изучения отдельных тем, вопросов следует обращаться за консультациями к преподавателю.



МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ  
Федеральное государственное автономное образовательное учреждение  
высшего образования  
«Дальневосточный федеральный университет»  
(ДВФУ)

---

**ШКОЛА ЦИФРОВОЙ ЭКОНОМИКИ**

**ФОНД ОЦЕНОЧНЫХ СРЕДСТВ**  
**по дисциплине «Нейронные сети и глубокое обучение»**  
**Направление подготовки – 09.04.03 Прикладная информатика**  
**магистерская программа «Искусственный интеллект и большие данные»**  
**Форма подготовки очная**

**Владивосток**  
**2018**

Фонд оценочных средств по дисциплине «Распознавание образов и машинное обучение» включает в себя:

- критерии и показатели, необходимые для оценки знаний, умений, навыков и характеризующие этапы формирования компетенций в процессе освоения образовательной программы
- типовые контрольные задания,
- методические материалы, определяющие процедуры оценивания знаний, умений и навыков и (или) опыта деятельности,

Сформированность каждой компетенции в рамках освоения данной дисциплины оценивается по трехуровневой шкале:

- пороговый уровень является обязательным для всех обучающихся по завершении освоения дисциплины;
- продвинутый уровень характеризуется превышением минимальных характеристик сформированности компетенции по завершении освоения дисциплины;
- эталонный уровень характеризуется максимально возможной выраженностью компетенции и является важным качественным ориентиром для самосовершенствования.

Уровень сформированности каждой компетенции на различных этапах ее формирования в процессе освоения данной дисциплины оценивается в ходе текущего контроля успеваемости представлен различными видами оценочных средств.

Результаты обучения (компетенции из ФГОС)	Знает	Умеет	Владеет
ОК-1 ОПК-4 ПК-13	Методы построения нейронных сетей и технологии глубокого обучения в различных системах; задачи, для решения которых применяются методы машинного обучения	Ставить задачи и разрабатывать алгоритмы их решения, использовать необходимые методы построения нейронных сетей и технологии глубокого обучения, реализовывать выбранные или разработанные алгоритмы	Математическим и алгоритмическим аппаратом, применяемым при решении задач построения нейронных сетей и технологии глубокого обучения
Эталонный	Основной и дополнительный материал, предусмотренный компетенцией, без	<b>Умеет</b> в полном объеме ставить задачи и разрабатывать алгоритмы их решения, использовать	всеми навыками, демонстрируя их не только в стандартных ситуациях, но и при решении

	ошибок и погрешностей	необходимые методы построения нейронных сетей и технологии глубокого обучения, реализовывать выбранные или разработанные алгоритмы	нестандартных задач
Продвинутый	основной материал, предусмотренный компетенцией, без ошибок и погрешностей	<b>Умеет</b> с незначительными погрешностями ставить задачи и разрабатывать алгоритмы их решения, использовать необходимые методы построения нейронных сетей и технологии глубокого обучения, реализовывать выбранные или разработанные алгоритмы	основными навыками, демонстрируя их в стандартных ситуациях, в том числе при решении дополнительных задач
Пороговый	большинство основных понятий, изучаемых в рамках дисциплины	<b>Умеет</b> с погрешностями ставить задачи и разрабатывать алгоритмы их решения, использовать необходимые методы построения нейронных сетей и технологии глубокого обучения, реализовывать выбранные или разработанные алгоритмы	некоторыми основными навыками, демонстрируя их в стандартных ситуациях

### **Методические материалы, определяющие процедуры оценивания результатов освоения дисциплины**

Текущая аттестация студентов проводится в соответствии с локальными нормативными актами ДВФУ и является обязательной. Текущая аттестация проводится в форме контрольных мероприятий: собеседование, защита лабораторных работ.

Объектами оценивания выступают:

- учебная дисциплина (активность на занятиях, своевременность выполнения различных видов заданий, посещаемость всех видов занятий по аттестуемой дисциплине);
- степень усвоения теоретических знаний;
- уровень овладения практическими умениями и навыками по всем видам учебной работы;

- результаты самостоятельной работы.

Во время выполнения лабораторных работ преподаватель на основе серии контрольных вопросов проверяет теоретические знания студента по теме лабораторной работы. Для экзамена подготовлены 12 билетов. Оценка качества освоения дисциплины производится по результатам следующих контролируемых мероприятий:

### **Список вопросов к экзамену**

1. Обзор основных моделей нейросетей.
2. Сверточные нейронные сети (свертки и сверточные сети; часто используемые архитектуры).
3. Рекуррентные нейросети. Обработка последовательностей и “ванильная” рекуррентная сеть;
4. Использование Earthengine google для анализа спутниковой съемки.
5. Основы работы с TensorFlow.
6. Построение многослойного перцептрона в Google EarthEngine.
7. Построение сверточной сети средствами TensorFlow.
8. Анализ последовательностей изображений в Google EarthEngine.
9. Реализация “ванильной” рекуррентной нейронной сети в TensorFlow.
1. Методы оптимизации. Градиентный спуск.
2. Линейная регрессия.
3. Глобальная оптимизация. Генетический алгоритм.
4. Метод ближайших соседей (k-NN)
5. Наивный байесов классификатор
6. Логистическая регрессия
7. Сигмоид. Метод наибольшего правдоподобия.
8. Деревья решений
9. Кластеризация
10. Снижение размерности
11. Метод опорных векторов (SVM)
12. Кросс-валидация. Подбор гиперпараметров. Визуализация данных.
13. Методы селекции: пропорционально качеству, универсальная выборка (stochastic universal sampling), с наследием (reward-based), турнир. Стратегия элитизма.
14. Методы кроссовера. Двух и много-точечный, равномерный (по подмножествам), для перестановок.
15. Управление популяцией. Сегрегация, старение, распараллеливание. Генетическое программирование.

16. Наивный классификатор, предположение о независимости признаков. Оценка плотности распределения для числовых признаков. Алгоритмические оптимизации. Алгоритм EM.
17. Задача оптимизации с ограничениями. Двойственная задача Лагранжа. Условия Каруша-Куна-Такера.
18. Преобразование Хафа
19. Алгоритм Кэнни
20. Задача классификации изображений. Основные архитектуры нейронных сетей
21. Задача сегментации. Архитектура Encoder-Decoder. Основные виды нейронных сетей для задачи сегментации
22. Задача локализации. Алгоритм SSD, Алгоритм RCNN, Fast RCNN, Faster RCNN, YOLO.
23. Задача Instance Segmentation. Архитектура Mask RCNN
24. Автоэнкодеры
25. Генеративные модели.

#### *Оценка предварительной подготовки*

1. Стек вызова, работа с динамической памятью, время жизни переменных
2. Абстрактные типы данных (стек, очередь)
3. Работа с динамической памятью, разбор и реализация типа данных "Очередь". Работа с многофайловым проектом, применение и практика.
4. Введение в алгоритмы и структуры данных (сложность, классификации итд). Разбор процесса компиляции приложения (препроцессор, компиляция, линковщик). Разбор стандартной библиотеки Си.
5. Наследование и полиморфизм.
6. Шаблоны в языке C++, вывод типов, шаблонные функции, шаблонные классы, итераторы.
7. Обзор стандартной библиотеки C++. Практикум студентов на языке C++. Разбор задачи
8. Объектно-ориентированное программирование. Инкапсуляция, наследование, полиморфизм. Перегрузка операторов. Декораторы.

#### **Критерии выставления оценки студенту на экзамене**

Порядок начисления рейтинговых баллов по предмету

Выполнение лабораторных работ – 100 баллов

Баллы (рейтинг)	Оценка экзамена	Требования к сформированным компетенциям
-----------------	-----------------	------------------------------------------

говой оценки )		
85-100	«отлично»	Оценка «отлично» выставляется студенту, если он глубоко и прочно усвоил программный материал, исчерпывающе, последовательно, четко и логически стройно его излагает, умеет тесно связывать теорию с практикой, свободно справляется с задачами, вопросами и другими видами применения знаний, причем не затрудняется с ответом при видоизменении заданий, использует в ответе материал монографической литературы, владеет разносторонними навыками и приемами выполнения лабораторных работ.
70-84	«хорошо»	Оценка «хорошо» выставляется студенту, если он твердо знает материал, грамотно и по существу излагает его, не допуская существенных неточностей в ответе на вопрос, правильно применяет теоретические положения при решении лабораторных работ вопросов и задач, владеет необходимыми навыками и приемами их выполнения.
50-69	«удовлетворительно»	Оценка «удовлетворительно» выставляется студенту, если он имеет знания только основного материала, но не усвоил его деталей, допускает неточности, недостаточно правильные формулировки, нарушения логической последовательности в изложении программного материала, испытывает затруднения при выполнении лабораторных работ.
0-49	«неудовлетворительно»	Оценка «неудовлетворительно» выставляется студенту, который не знает значительной части программного материала, допускает существенные ошибки, неуверенно, с большими затруднениями выполняет практические работы. Как правило, оценка «неудовлетворительно» ставится студентам, которые не могут продолжить обучение без дополнительных занятий по соответствующей дисциплине.

Правила аттестации для студентов, не набравших необходимый минимум баллов по дисциплине

Если студент, в ходе изучения дисциплины набрал 70 и более баллов, то он имеет право на выставление соответствующей оценки по экзамену без его сдачи.

Если студент набрал менее 70 баллов, то он должен сдавать экзамен (экзаменационный тест). Данный тест оценивается в диапазоне от 0 до 30 баллов. Полученные баллы суммируются к уже набранным и студенту выставляется итоговая оценка.

## Типовые лабораторные работы

### Задача А. Градиентный спуск

Входной файл: Стандартный вход  
Выходной файл: Стандартный выход

Ограничение времени: 1 сек  
Ограничение памяти: 512 Мб

- **Условие**

Требуется реализовать класс на языке Python, который соответствует следующему интерфейсу.

```
class GradientOptimizer:
    def __init__(self, oracle, x0):
        self.oracle = oracle
        self.x0 = x0

    def optimize(self, iterations, eps, alpha):
        pass
```

В конструктор принимаются два аргумента — оракул, с помощью которого можно получить градиент оптимизируемой функции, а также точку, с которой необходимо начать градиентный спуск.

Метод `optimize` принимает максимальное число итераций для критерия остановки, L2-норму градиента, которую можно считать оптимальной, а также learning rate. Метод возвращает оптимальную точку.

Оракул имеет следующий интерфейс:

```
class Oracle:
    def get_func(self, x)
    def get_grad(self, x)
```

`x` имеет тип `np.array` вещественных чисел.

- **Формат выходных данных**

Код должен содержать только класс и его реализацию. Он не должен ничего выводить на экран.

### Задача В. Линейная регрессия. Основы

Входной файл: Стандартный вход  
Выходной файл: Стандартный выход

Ограничение времени: 1 сек  
Ограничение памяти: 512 Мб

- **Условие**

Требуется реализовать следующие функции на языке Python.

```
def linear_func(theta, x) # function value
def linear_func_all(theta, X) # 1-d np.array of function values of
all rows of the matrix X
def mean_squared_error(theta, X, y) # MSE value of current regression
def grad_mean_squared_error(theta, X, y) # 1-d array of gradient by theta
```

`theta` — одномерный `np.array`

`x` — одномерный `np.array`

`X` — двумерный `np.array`. Каждая строка соответствует по размерности вектору `theta`

$y$  — реальные значения предсказываемой величины

Матрица  $XX$  имеет размер  $M \times NM \times N$ .  $MM$  строк и  $NN$  столбцов.

Используется линейная функция вида:  $h_{\theta}(x) = \theta_1 x_1 + \theta_2 x_2 + \dots + \theta_n x_n$

Mean squared error (MSE) как функция от  $\theta$ :  $J(\theta) = \frac{1}{M} \sum_{i=1}^M (y_i - h_{\theta}(x^{(i)}))^2$

Где  $x^{(i)}$  —  $i$ -я строка матрицы  $XX$

Градиент функции MSE:  $\nabla J(\theta) = \{\partial J \partial \theta_1, \partial J \partial \theta_2, \dots, \partial J \partial \theta_N\}$

- **Пример**

```
X = np.array([[1, 2], [3, 4], [4, 5]])
theta = np.array([5, 6])
y = np.array([1, 2, 1])
linear_func_all(theta, X) # -> array([17, 39, 50])
mean_squared_error(theta, X, y) # -> 1342.0
grad_mean_squared_error(theta, X, y) # -> array([215.33333333, 283.33333333])
```

- **Формат выходных данных**

Код должен содержать только реализацию функций.

---

## Задача С. Найти линейную регрессию

Входной файл: Стандартный вход

Ограничение времени: 10 сек

Выходной файл: Стандартный выход

Ограничение памяти: 512 Мб

- **Условие**

Требуется реализовать функцию на языке Python, которая находит линейную регрессию заданных векторов, используя метрику MSE.

```
def fit_linear_regression(X, y) # np.array of linear regression coeffs
```

$X$  — двумерный `np.array`. Каждая строка соответствует отдельному примеру.

$y$  — реальные значения предсказываемой величины

- **Формат выходных данных**

Код должен содержать только реализацию функций.

---

## Распределение задач

Входной файл: input.txt

Ограничение времени: 1 сек

Выходной файл: output.txt

Ограничение памяти: 256 Мб

- **Условие**

Группа разработчиков работает над проектом. Весь проект разбит на задачи, для каждой задачи указывается ее категория сложности (1, 2, 3 или 4), а также оценочное время выполнения задачи в часах. Проект считается выполненным, если выполнены все задачи. Для каждого разработчика и для каждой категории сложности задачи указывается коэффициент, с которым, как ожидается, будет соотноситься реальное время выполнения задачи данным разработчиком к оценочному времени. Считается, что все разработчики начинают работать с проектом в одно и то же время и выделяют для работы одинаковое время. Необходимо реализовать программу, распределяющую задачи по разработчикам, с целью минимизировать время выполнения проекта (получить готовый проект за минимальный промежуток времени). Поиск решения необходимо реализовать с помощью генетического алгоритма.

- **Отправка решения и тестирование**

Данная задача будет проверяться на *ОДНОМ* входном файле. Этот файл можно скачать [ЗДЕСЬ](#).

В качестве решения принимается текстовый файл, содержащий ответ к задаче в требуемом формате (при его отправке следует выбрать в тестирующей системе среду разработки "Answer text").

Решение набирает количество баллов, вычисляемое по следующей формуле:  $Score = 10 \cdot T_{max} \cdot \frac{Score}{T_{max}}$ .  $T_{max}$  — наибольшее среди всех разработчиков время, затраченное на выполнение выданных соответствующему разработчику задач.

- **Формат входного файла**

Первая строка входного файла содержит целое число  $NN$  количество задач.

Вторая строка —  $NN$  целых чисел от 1 до 4 категорий сложности задач.

Третья строка —  $NN$  вещественных положительных чисел оценочного времени для задач.

Четвертая строка — целое число  $MM$ , количество разработчиков.

Следующие  $MM$  строк содержат по 4 вещественных положительных числа — коэффициенты каждого разработчика.

- **Формат выходного файла**

Первая и единственная строка выходного файла содержит  $NN$  целых чисел  $w_i$  — номер разработчика, назначенного на  $i$ -ю задачу.

- **Ограничения**

- **Примеры тестов**

№	Входной файл (input.txt)	Выходной файл (output.txt)
1	3 1 1 4 5.2 3.4 4 2 1 1 2 5 0.7 1 1.2 1.5	1 2 2

---

## Логистическая регрессия. Основы

Входной файл: Стандартный вход

Ограничение времени: 1 сек

Выходной файл: Стандартный выход

Ограничение памяти: 512 Мб

- **Условие**

Требуется реализовать следующие функции на языке Python.

```
def logistic_func(theta, x) # function value
def logistic_func_all(theta, X) # 1-d np.array of function values
of all rows of the matrix X
def cross_entropy_loss(theta, X, y) # cross entropy loss value of
current regression
def grad_cross_entropy_loss(theta, X, y) # 1-d array of gradient by theta
```

$\theta$  — одномерный `np.array`

$x$  — одномерный `np.array`

$X$  — двумерный `np.array`. Каждая строка соответствует по размерности вектору  $\theta$

$y$  — реальные значения предсказываемой величины

Матрица  $XX$  имеет размер  $M \times NM \times N$ .  $MM$  строк и  $NN$  столбцов.

Используется линейная функция вида:  $h_\theta(x) = \theta_1 x_1 + \theta_2 x_2 + \dots + \theta_n x_n$

- **Формат выходных данных**

Код должен содержать только реализацию функций.

---

Найти логистическую регрессию

Входной файл: Стандартный вход  
Выходной файл: Стандартный выход

Ограничение времени: 10 сек  
Ограничение памяти: 512 Мб

- **Условие**

Требуется реализовать функцию на языке Python, которая находит логистическую регрессию заданных векторов, используя метрику cross entropy loss.

```
def fit_logistic_regression(X, y) # np.array of logistic regression coeffs
```

X — двумерный `np.array`. Каждая строка соответствует отдельному примеру.

y — реальные значения предсказываемой величины

- **Формат выходных данных**

Код должен содержать только реализацию функций.

---

## News category

Входной файл: input.txt  
Выходной файл: output.txt

Ограничение времени: 1 сек  
Ограничение памяти: 256 Мб

- **Условие**

Требуется обучить модель определения категории новости. Обучающую выборку можно скачать [ЗДЕСЬ](#). Категория новости в обучающей выборке представлена столбцом `CAT`.

- `HEADER` — заголовок новости
- `MEDIA_NAME` — название СМИ
- `WEBSITE` — вебсайт СМИ
- `PTIME` — время публикации

Для определения качества модели будет использоваться тестовая выборка, доступная [ЗДЕСЬ](#).

В тестовой выборке требуется предсказать значения столбца `CAT`, соответствующие каждому тестовому примеру. Категории новостей кодируются одним символом, аналогично данным в обучающей выборке.

- **Отправка решения и тестирование**

Данная задача будет проверяться на *ОДНОМ* входном файле.

В качестве решения принимается текстовый файл, содержащий ответ к задаче в требуемом формате (при его отправке следует выбрать в тестирующей системе среду разработки "Answer text").

Решение набирает количество баллов, вычисляемое по следующей формуле:  $Score = 105 \cdot AccuracyScore$ . `AccuracyScore` — доля верно классифицированных новостей относительно всех новостей в тестовой выборке.

- **Формат выходного файла**

Каждая строка выходного файла должна содержать единственный символ, задающий категорию соответствующего тестового примера.

---

## Качество вина

Входной файл: input.txt  
Выходной файл: output.txt

Ограничение времени: 1 сек  
Ограничение памяти: 256 Мб

- **Условие**

Требуется обучить модель определения качества вина. Качество вина определяется по 1010-балльной шкале. В данной задаче будем использовать бинарную модель и предсказывать,

"хорошее" вино или "плохое". Хорошим будем считать вино с качеством строго выше 66. Обучающую выборку можно скачать [ЗДЕСЬ](#). Качество вина представлено столбцом `quality`. Для определения качества модели будет использоваться тестовая выборка, доступная [ЗДЕСЬ](#). В тестовой выборке требуется предсказать значения 11 или 00, "хорошее" вино или "плохое" соответственно, для каждого примера. Оценку качества по 1010-балльной шкале предсказывать не требуется.

- **Отправка решения и тестирование**

Данная задача будет проверяться на *ОДНОМ* входном файле.

В качестве решения принимается текстовый файл, содержащий ответ к задаче в требуемом формате (при его отправке следует выбрать в тестирующей системе среду разработки "Answer text").

Решение набирает количество баллов, вычисляемое по следующей формуле:  $Score=105 \cdot F1$   $Score=105 \cdot F1$ .

- **Формат выходного файла**

Каждая строка выходного файла должна содержать целое число 11 или 00. Количество строк должно быть равно количеству эл