

Нейронные сети на Python

Курс о нейронных сетях, глубоком обучении и их применении для решения задач CV, NLP и обучения с подкреплением

Длительность курса: 128 академических часов

1 Первые шаги

1 Искусственный интеллект и другие задачи которые можно решать с помощью нейронных сетей

объяснить, что такое искусственные нейроны для того, чтобы понимать из чего состоят искусственные нейронные сети, познакомиться с историей развития искусственного интеллекта и актуальными задачами, которые стоят перед учеными в области Data Science.

Домашние задания

1 Docker

Цель: Собрать свой докер-образ для работы с нейросетями.

В докерфайле из материалов нужно заменить версию Tensorflow на последнюю и добавить установку Keras. Запустить контейнер из своего образа, импортировать TF и Keras и вывести версии.
Выложить скриншот с версиями

2 Стохастический градиентный спуск и метод обратного распространения

разобрать математическую основу обучения нейронной сети и базовые идеи работы с данными.

3 Первая нейронная сеть на PyTorch

работать с фреймворком PyTorch; реализовывать простейшие нейронные сети.

Домашние задания

1 Логистическая регрессия и переобучение

1. В файле `log_reg.ipynb` изменить код таким образом, чтобы решить ту же самую задачу, а именно обучить логистическую регрессию, на двумерных данных

2. Добиться от классификатора на `mnist` переобучения за счет изменения архитектуры и гиперпараметров.

4	Переобучение и регуляризация нейронных сетей	объяснить, что такое переобучение нейронной сети, и как с ним бороться для повышения эффективности своих моделей.
5	Первая нейронная сеть на Tensorflow	<p>реализовывать нейронные сети с помощью одного из самых популярных фреймворков.</p> <p>Домашние задания</p> <p>1 Основы <code>tensorflow</code> и <code>tf.keras</code></p> <p>Цель: Скачайте ноутбук с домашним заданием по ссылке. Более подробная инструкция в ноутбуке, в том числе и в какой форме его можно сдать.</p> <p>Цель домашнего задания - научиться создавать собственные слои <code>tf.keras</code> с заданной логикой и применять их для создания нейронных сетей.</p> <p>Скачайте ноутбук с домашним заданием по ссылке.</p> <p>Более подробная инструкция в ноутбуке, в том числе и в какой форме его можно сдать.</p>
6	Взрыв и затухание градиентов	проанализировать проблемы взрыва и затухания градиентов и познакомиться с методами борьбы позволяющими обучать более глубокие нейронные сети.
7	Погружение в Tensorflow	разобрать продвинутые возможности фреймворка <code>Tensorflow</code> .
8	Адаптивные методы градиентного спуска	применять современные модификации метода градиентного спуска для ускорения обучения нейронных сетей.

2 Основные архитектуры и методы обучения нейронных сетей

1	Основные архитектуры нейронных сетей: Автокодировщики	<p>создавать автокодировщики; модифицировать основную структуру под конкретную задачу.</p> <p>Домашние задания</p> <p>1 Автокодировщик</p> <p>1. Обучить многослойный автокодировщик: а. отрисовать классы в латентном представлении автокодировщика с двумерным латентным слоем, б. сделать tSNE над 32-мерным латентным слоем.</p> <p>1. Обучить автокодировщик с шумом: а. Добавить ко входным данным нормальный шум, б. Сделать отражение части изображения, в. Отрисовать карты активаций нейронов внутреннего слоя.</p>
2	Практическое занятие на TensorFlow. Разреженный Автокодировщик	<p>решать задачи удаления шума с изображения и разреживания представления данных.</p>
3	Основные архитектуры нейронных сетей: Сверточные сети	<p>использовать наиболее популярную архитектуру нейронной сети для решения задач компьютерного зрения, а также операцию обратную свертке для реализации сверточных автокодировщиков.</p> <p>Домашние задания</p> <p>1 Сверточные сети</p> <p>Необязательное домашнее задание</p> <p>1. Реализовать сверточный автокодировщик 2. Сделать оптимизацию изображения под активацию сверточных нейронов классификатора 3. Сделать оптимизацию изображения под активацию сверточных нейронов автокодировщика</p>
4	Практическое занятие на Keras. Transfer Learning	<p>разобрать на практике один из ключевых методов решения задач компьютерного зрения.</p>
5	Основные архитектуры нейронных сетей: Рекуррентные сети	<p>на примере задачи классификации коротких текстов, познакомиться еще с одной важнейшей архитектурой — рекуррентной нейронной сетью; помимо обработки естественных языков рассмотреть общий подход к работе с последовательными данными с помощью рекуррентной сети.</p>

<p>6</p> <p>Практическое занятие на PyTorch. Генерация Википедии</p>	<p>перейти от предсказания свойств текста к порождению собственных текстов.</p> <p>Домашние задания</p> <p>1 Создаем Википедию</p> <p>1. Используя подход аналогичный torchvision, сделать свой класс датасета.</p> <p>Необязательное д/з:</p> <p>1. Поэкспериментировать с разными архитектурами рекурренток: тип ячеек, слои, нормализация, методы оптимизации</p>
<p>7</p> <p>Введение в обучение с подкреплением</p>	<p>применять методы классического обучения с подкреплением для таких задач как A/B тестирование или игра в шахматы; разобраться с тем, кто такие многорукие бандиты и что такое TD-обучение.</p>
<p>8</p> <p>Практическое занятие. Крестики-нолики</p>	<p>на примере игры в крестики-нолики сравнить решение с помощью нейронной сети с "табличными" методами обучения с подкреплением.</p> <p>Домашние задания</p> <p>1 Deep-Tac-Toe</p> <p>Цель: Опираясь на табличное обучение с подкреплением реализовать и обучить нейронную Q-функцию.</p>

3 Современные нейронные сети для задач компьютерного зрения и обработки естественных языков

- | | | |
|---|---|---|
| 1 | Современные сверточные сети. Блоки и шорткаты | проанализировать историю развития современных сверточных сетей;
объяснить, что позволило сделать их по-настоящему глубокими и как это повлияло на область компьютерного зрения. |
| 2 | Современные рекуррентные сети. Внимание | “научить” нейросеть обращать внимание на наиболее важную информацию на изображении или в тексте;
использовать эту способность, чтобы искать логически связанные блоки информации, чем это поможет при машинном переводе, поиске ответов на вопросы, аннотации изображений. |
| 3 | Современные рекуррентные сети. Трансформеры | использовать только attention-слои для Natural Language Inference, а также разобраться, как трансформер позволяет создать эмбединги, учитывающими контекст при кодировании многозначных слов. |
| 4 | Metric-learning и обучение без примеров | отображать объекты в пространство с метрикой и использовать такое представление для поиска похожих объектов и решения задачи классификации при отсутствии данных. |
| 5 | Современные сверточные сети. Transfer learning и fine-tuning | объяснить, что делать, если данных недостаточно для обучения глубокой нейронной сети;
использовать знания из уже обученной нейронной сети и подстраивать такую сеть к конкретной задаче. |

Домашние задания

- 1 Катастрофическое забывание (необязательное)

Цель: Проверить влияние fine-tuning на исходную модель

1. Дообучить готовую модель на imagenette2 и проверить качество
2. Сохранить последний слой обученной модели и заменить его на новый для CIFAR10
3. Дообучить модель решать датасет CIFAR10 и проверить качество
4. Вернуть оригинальный последний слой модели и проверить качество на imagenette2
5. Дообучить только последний слой на imagenette2 и проверить удалось ли добиться исходного качества.

- | | | |
|---|---|--|
| 6 | Генеративные состязательные сети | объяснить, как нейронная сеть учится обманывать другую нейронную сеть, и как от этого они обе становятся лучше, а мы получаем генератор качественных данных. |
|---|---|--|

1 GANs

1. Сделать сверточный GANs

- | | | |
|-------|--|---|
| 7 | Глубокие
сверточные сети.
Сегментация | разобраться, как при помощи нейронной сети сегментировать изображение, какие для этого есть архитектуры, что за принципы лежат в основе UNet и DeepLab. |
| <hr/> | | |
| 8 | Глубокие
сверточные сети.
Детекция | проанализировать принципы работы нейронных сетей для детекции объектов: от эволюцию от первых “чисто” нейросетевых алгоритмов до современных архитектур, работающих за “один взгляд” - YOLO, SSD. |

1	Вариационный и состязательный автокодировщики	<p>Домашние задания</p> <p>1 Вариационный автокодировщик</p> <p>Реализовать VAE в pytorch для MNIST датасета. 1.Обучить VAE с 32- и 64-мерным латентным слоем и сравнить качество автокодирования и качество генерации новых изображений 2.Обучить VAE с 2-мерным латентным слоем и сгенерировать изображения из решетки в латентном пространстве</p>
2	Доменная адаптация и условная генерация	<p>разобрать подходы к обучению нейронных сетей на многомодальных данных, способы целенаправленного забывания информации и применение состязательного фреймворка для "честной" условной генерации.</p> <p>Домашние задания</p> <p>1 Условная генерация</p> <p>Обучить две версии CAAE — "частную" и "нечестную", преобразовать с их помощью исходные данные в латентное представление и обучить 3 классификатора — один на исходных данных и два на кодах, полученных из "честного" и "нечестного" CAAE. Сравнить качество классификации. Необязательная часть: Сделать перенос стиля с одной цифры на другие с помощью "честного" CAAE</p>
3	Генеративные модели для текстов	<p>работать с методом Professor forcing; использовать идеи обучения с подкреплением для генерации текстов, а также пропускать градиент через дискретные объекты с помощью репараметризации.</p>
4	Катастрофическое забывание	
5	Глубокое обучение с подкреплением	<p>проанализировать как же Google удалось победить чемпиона мира в игру го и что еще можно сделать таким же способом.</p>
6	Обратное обучение с подкреплением	<p>имитировать поведение эксперта; объяснить, что делать, если награды неизвестны и можно ли назначать награды нейронной сетью.</p>
7	Сферические	<p>применять непрерывные свертки для данных имеющих</p>

**свертки и
множества точек**

пространственную структуру.

8

**Сверточные и
рекуррентные сети
на графах**

проанализировать архитектуры, позволяющие работать с
данными имеющими графовую структуру.

1 Выбор темы и организация проектной работы / Консультация по домашним заданиям

выбрать и обсудить тему проектной работы;
спланировать работу над проектом;
ознакомиться с регламентом работы над проектом.

Домашние задания

1 Выпускная работа

Цель: В рамках курса предусмотрена защита проекта. Он представляет собой генеративную модель для порождения текста заданного стиля или изображений заданной тематики.

Работа над проектом проходит поэтапно:

1. Выбор тематики.
2. Сбор и подготовка соответствующих данных.
3. Построение и обучение генеративной модели

Проект выполняется программистом в течение двух недель после прохождения курса. Готовое решение должно включать в себя код модели и ее обучения на Python, описание архитектуры модели, отчет об обучении модели и примеры генерации.

Примеры тем проекта:

имплементация универсального класса оптимизатора, обобщающего различные методы градиентного спуска для PyTorch;
состязательная нейронная сеть для порождения изображений рукописного текста с заданным стилем;
генеративная модель повышения разрешения изображений;
рекуррентная состязательная seq2seq-архитектура для порождения стихотворений на основе обычного текста.

Преподаватели



Артур Кадури
Chief AI Officer Insilico Medicine

В Mail.Ru Group был первым сотрудником в группе по анализу данных департамента рекламных технологий. После того, как группа выросла в отдел, возглавил группу сегментирования аудитории. Руководил и участвовал в проектах, связанных с анализом пользовательского поведения в рекламе, соцсетях, играх и т. д.

С 2016 года применяет методы глубокого обучения для научных исследований в области медицинской химии совместно с командой Insilico Medicine, где после успешного проекта занял должность директора по AI. В конце 2017 г. возглавил дочернюю компанию Insilico Taiwan в качестве исполнительного директора.

В 2008 году получил специальность математика и системного программиста в Кубанском Государственном Университете на Факультете Компьютерных Наук и Прикладной Математики, с 2013 года занимается машинным обучением.

Соавтор книги [Глубокое обучение](#)

Соавтор многочисленных научных публикаций: [Google Scholar](#)



Артур
Кадури



Михаил
Степанов



Артем
Васильев



Евгения
Ческидова



Антон
Витвицкий