



Deep Learning

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

Длительность курса: 108 академических часов

1 Обзорное занятие

Цели занятия:

познакомиться;
разобрать, как устроен процесс обучения и какими инструментами будем пользоваться.

Краткое содержание:

о чем этот курс;
преподаватели;
инструменты: докер/колаб.

Домашние задания

1 Установка и настройка рабочего окружения

Цель: Собрать свой docker контейнер для работы с моделями глубокого обучения.

Необходимо установить и настроить рабочее окружение:

- 1) Установить и настроить Docker.
- 2) Если используете NVIDIA GPU, то также установить nvidia-docker.
- 3) Собрать и запустить контейнер с Ubuntu 18.04 (или выше).
- 4) Развернуть в контейнере PyTorch, TensorFlow, JupyterLab, OpenCV.
- 5) Зайдите в Google Colab и запросите GPU.
- 6) В качестве результатов отправьте в личный кабинет следующее:
 - a) Конфигурацию вашего компьютера: ОС, модель GPU и CPU.
 - b) Скриншот или лог, который будет содержать версию PyTorch и TensorFlow, запущенные из под Docker.
 - c) Скриншот или лог, который будет содержать версию PyTorch и TensorFlow, запущенные в Google Colab.

Пример Dockerfile (для GPU) и Dockerfile_cpu (для CPU) можно найти в материалах к лекции.

2 **Градиентный
спуск.
Математика**

Цели занятия:

рассмотреть базовую математику необходимую для обучения нейронных сетей.

Краткое содержание:

производная/градиент;
матричные операции.

3 Градиентный спуск. Вывод

Цели занятия:

разобрать процесс обучения простейшей нейронной сети на практике;
обсудить, как его можно обобщить на более сложные случаи.

Краткое содержание:

полносвязный перцептрон;
дифференцируемые функции и граф вычислений.

Домашние задания

1 Знакомство с оптимизаторами в PyTorch

Цель: Цель текущего задания познакомиться с тем как ведут себя на практике различные оптимизаторы в PyTorch и насколько сильно они могут влиять на тренинг даже в самом простом случае.

1. Написать код простой подготовки данных.
2. Собрать простенькую NN модель.
3. Обучить модель с разными оптимизаторами (Adam, SGD, RMSprop).
4. Сравнить результаты разных оптимизаторов.

1. Открываем colab по [ссылке] (<https://colab.research.google.com/drive/1Qyomr7jxliujhvwBDnHBXlIxQ1gez8la?usp=sharing>) и делаем себе его копию;
 2. Делаем задания описанные внутри ноутбука(попутно делаем git всем ячейкам и изучаем результаты, если есть вопросы то не стесняемся их задавать);
 3. Отправляем проверяющему результаты.
-

4 NumPy

Цели занятия:

разобрать базовые математические операции в техническом коде и обсудить библиотеку numpy.

Краткое содержание:

повторяем матричные операции;
numpy-style code.

Домашние задания

1 Градиентный спуск на NumPy

Цель: Реализовать метод стохастического градиентного спуска на примере задачи логистической регрессии.

0. Используя `np.random` научиться сэмплировать случайные точки из двумерного нормального распределения.

В качестве параметров двух разных нормальных распределений взять $\mu_0 = (-1, -1)$ и $\mu_1 = (1, 1)$, с единичными дисперсиями.

1. Заменить функцию ошибки MSE на LogLoss.

2. Обучить модель на примерах из этих распределений "предсказывать" 0 и 1 соответственно для первого и второго распределений.

3. Используя библиотеку `matplotlib` нарисовать разделяющую поверхность.

4. Заменить однослойную нейронную сеть двухслойной.

5 Распределения и информация

Цели занятия:

разобраться с основами теории информации и ее применением к машинному обучению.

Краткое содержание:

Бернулли, Гаусс, равномерное распределение; информация, энтропия, кросс-энтропия.

Цели занятия:

разделить на трейн, тест и валидацию;
рассмотреть PyTorch и torchvision;
научиться классифицировать MNIST.

Краткое содержание:

PyTorch;
PyTorch - Первый классификатор;
переобучение.

7 Логрессия на pytorch

Цели занятия:

познакомиться с одним из основных фреймворков для работы с нейронными сетями на примере решения задачи логистической регрессии в нейросетевом стиле.

Краткое содержание:

логистический сигмоид;
вывод логосса через правдоподобие;
Pytorch.

Домашние задания

1 Практика по PyTorch

Цель: Получить навыки практической работы с PyTorch.

Используя PyTorch:

- 1) Создать и обучить регрессионную модель, которая аппроксимирует значение функции $\sin(x + 2*y) * \exp(-(2*x + y)^2)$ на диапазоне значений $[-10; 10]$ по x и y .
- 2) В качестве обучающей выборки необходимо сгенерировать 20000 точек случайным образом.
- 3) Разделить получившийся датасет на train / test / val в отношениях 70% / 15% / 15%, соответственно.

Результаты:

- a) Посчитать метрику Mean Square Error(MSE) на test.
- b) Нарисовать график, в котором сравнивается истинная функция и ее аппроксимированный вариант (предсказываемый моделью).

1 Переобучение и регуляризация

Цели занятия:

разобраться с проблемой переобучения и ключевыми методами ее решения.

Краткое содержание:

L1/L2;
dropout;
batchnorm.

Домашние задания

1 Переобучение

Цель: Добиться от классификатора на MNIST переобучения за счет изменения архитектуры и гиперпараметров.

1. Возьмите классификатор из ноутбука занятия или создайте свой.
 2. В качестве датасета используйте MNIST или Fashion MNIST.
 3. Постройте кривые обучения train/test.
 4. Путем изменения архитектуры нейросети, гиперпараметров или исходных данных добейтесь переобучения.
 5. Сделайте вывод, какие действия привели к переобучению.
-

2 Взрыв и затухание градиентов

Цели занятия:

повторить градиентный спуск;
создать свой класс модели.

Краткое содержание:

Попробуем создать свой класс модели на PyTorch и повторим градиентный спуск

Домашние задания

1 Подбор гиперпараметров

Цель: Используя тестовое множество подобрать оптимальные гиперпараметры нейросети, попробовать различные способы инициализации.

1. Взять тестовое множество.
 2. Собрать свою нейронную сеть.
 3. Подобрать наиболее оптимальные параметры для этой архитектуры.
 4. Попробовать разные варианты инициализации
 5. Следить за процессом обучения в wandb
 4. Добиться значения accuracy на тестовом множестве не ниже 95%.
-

Цели занятия:

объяснить, что такое автокодировщик;
объяснить для чего нужны автокодировщики и какими они бывают.

Краткое содержание:

РСА;
автокодировщик;
разреженный автокодировщик;
практика: автокодировщики.

Домашние задания

1 Автокодировщик

Цель: Разберемся с тем, как работает автокодировщик и что содержится в латентном представлении данных.

1. Обучить многослойный автокодировщик:

- а. отрисовать классы в латентном представлении автокодировщика с двумерным латентным слоем;
- б. сделать tSNE над 32-мерным латентным слоем, отрисовать представление.

2. Обучить шумный автокодировщик:

- а. Добавить ко входным данным нормальный шум;
 - б. Сделать отражение части изображения;
 - в. Отрисовать карты активаций нейронов первого слоя.
-

4 Вариационный автокодировщик

Цели занятия:

повторить KL;
рассмотреть VAE.

Краткое содержание:

VAE Theory: Reparametrization trick.

Домашние задания

1 Вариационный автокодировщик и латентное пространство

Цель: В этом ДЗ вы обучите вариационный автокодировщик и исследуете его латентное пространство.

Ссылка на ноутбук с заданием:

<https://colab.research.google.com/drive/1h-bepWhR1IBhNjALTJTnJ-iZ5M8-Dtc-?usp=sharing>

5 Адаптивные методы градиентного спуска

Цели занятия:

применить современные модификации метода градиентного спуска для ускорения обучения нейронных сетей.

Краткое содержание:

метод моментов и метод Нестерова;
Adagrad/RMSProp.

поговорим о задаче Metric Learning - группе подходов и техник, при помощи которых решается задача измерения степени схожести изображений между собой (или объектов на них).

Краткое содержание:

начнем с основ - Mahalanobis Distance Metric;
перейдем к первым DL подходам, таким как Contrastive Loss, Triplet Loss;
далее углубимся в DL методы: Center Loss и SphereFace;
и наконец перейдем к топовым техникам: CosFace, ArcFace, AdaCos и Sub-Center ArcFace.

1 Сверточные сети. Классифицируем MNIST

Цели занятия:

собрать сверточную сеть из базовых элементов и сравнить ее с полносвязной на примере классификации.

Краткое содержание:

операции: свертка и пуллинг;
страйд, паддинг, разреженная свертка;
ReLU.

2 Сверточные сети, fine-tuning

Цели занятия:

переиспользовать нейросеть на новой задаче.

Краткое содержание:

Approaches to finetuning;
Deep dive into code;
Transfer learning with multi-target training.

Домашние задания

1 Катастрофическое забывание

Цель: Проверить влияние fine-tuning на исходную модель.

Для домашней работы использовать приложенный ноутбук Homework.ipynb. В нем будут содержаться вспомогательные функции.

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. (*) Сделать все тоже самое наоборот (то есть обучиться на CIFAR10, а дообучаться на подвыборку imagenette2).

В качестве результата нужно прислать ссылку на Colab.

3 AutoML

Цели занятия:

объяснить, что происходит в области AutoML;
научиться использовать набор инструментов для
новых задач;
поработать с NAS.

Краткое содержание:

AutoML;
"классический AutoML";
Neural Architecture Search.

4 Методы оптимизации сетей: pruning, mixint, quantization

Цели занятия:

рассмотрим методы ускорения вывода (inference)
нейронных сетей.

Краткое содержание:

оптимизация архитектуры модели (дистилляция
знаний, architecture search);
алгоритмическая оптимизация (pruning, factorization,
quantization);
вычислительная оптимизация (lookup table,
computation reusing).

5 Обучение с подкреплением. Q-learning для ТicTacToe

Цели занятия:

разобраться с основами обучения с подкреплением и
решить простую игру с его помощью.

Краткое содержание:

многорукие бандиты;
value, quality и policy;
Q-learning, TD-learning.

6 Рекуррентные сети

Цели занятия:

классифицировать и сгенерировать текст с RNN.

Краткое содержание:

Applications;
Basics of nlp;
Rnn types;
Vanilla RNN;
LSTM;
GRU;
Tutorials;
Classification;
Generation.

7 Transformers

Цели занятия:

обсудить типы обработки последовательностей;
проанализировать проблемы архитектуры Encoder-Decoder;
рассмотреть механизм внимания;
объяснить как вычислить веса.

Краткое содержание:

типы обработки;
Encoder-Decoder;
механизм внимания;
веса.

8 **Generative
Adversarial
Networks (GANs)**

Цели занятия:

познакомиться с фреймворком генеративных состязательных сетей и его применениями.

Краткое содержание:

генерация объектов;
условная генерация;
суперразрешение.

9 **Графовые
модели**

Цели занятия:

поговорим о графовых нейронных сетях (GNN).

Краткое содержание:

повторим что такое граф;
рассмотрим как работает Graph Convolution;
поймем что же такое графовые нейронные сети;
также затронем SpinConv и анализ молекул.

1 Выбор темы и организация проектной работы

Цели занятия:

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

Краткое содержание:

правила работы над проектом и специфика проведения итоговой защиты;
требования к результату проекта и итоговой документации.

Домашние задания

1 Проектная работа

Цель: Вместе с преподавателями выбрать тему итогового проекта;
Закрепить тему проекта в чате по ДЗ.

1. Выбрать тему
 2. Согласовать с преподавателем;
 3. Закрепить тему;
 4. Сделать и защитить проект.
-

2 Консультация по проектам и домашним заданиям

Цели занятия:

получить ответы на вопросы по проекту, ДЗ и по курсу.

Краткое содержание:

вопросы по улучшению и оптимизации работы над проектом;
затруднения при выполнении ДЗ;
вопросы по программе.

**3 Защита
проектных
работ**

Цели занятия:

защитить проект и получить рекомендации экспертов.

Краткое содержание:

презентация проектов перед комиссией;
вопросы и комментарии по проектам.

1 Введение в обучение с подкреплением

Цели занятия:

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.

2 Глубокие сверточные сети. Сегментация

Цели занятия:

разобраться, как при помощи нейронной сети сегментировать изображение, какие для этого есть архитектуры, что за принципы лежат в основе UNet и DeepLab.

Краткое содержание:

архитектуры: UNet, DeepLab.

3 Генеративные модели для текстов

Цели занятия:

работать с методом Professor forcing;
использовать идеи обучения с подкреплением для генерации текстов, а также пропускать градиент через дискретные объекты с помощью репараметризации.

Краткое содержание:

архитектуры: Professor forcing, ORGAN, Gumbel trick.