

Deep Learning. Basic

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

Длительность курса: 88 академических часов

1 Пререквизиты

- 1 Обзорное занятие**
 - о чем этот курс
 - преподаватели
 - инструменты: докер/колаб

Домашние задания

 - 1 Docker**

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

Для тех кто будет использовать Docker:
В докерфайле из материалов нужно заменить версии Pytorch и Tensorflow на последние, добавить установку Keras и запустить контейнер из своего образа.

Для всех:
Импортировать Pytorch и TF и вывести версии. Выложить скриншот с версиями в ЛК.
- 2 Градиентный спуск. Математика**
 - производная/градиент
 - матричные операции
- 3 Градиентный спуск. Вывод**
 - полносвязный перцептрон
 - дифференцируемые функции и граф вычислений
- 4 Numpy**
 - повторяем матричные операции
 - numpy-style code

Домашние задания

1 Градиентный спуск

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

Опционально:

Заменить однослойную нейронную сеть двуслойной.

5 **Распределения и информация**

- Бернулли, Гаусс, равномерное распределение
- информация, энтропия, кросс-энтропия, правдоподобие

6 **Линейная регрессия на TensorFlow**

- повторяем вывод лосса через правдоподобие
- повторяем слои, активации, ошибки на примере MSE
- TF

7 **Логрессия на pytorch**

- Логистический сигмоид
- вывод логлосса через правдоподобие
- Pytorch

2 Первая нейронная сеть

- 1 TensorFlow**
 - дашборд
 - обзор TF
 - функции активаций
 - Классифицируем MNIST

- 2 PyTorch**
 - разделение на трейн, тест и валидацию
 - Обзор PyTorch и torchvision
 - Классифицируем MNIST

Домашние задания

 - 1 Переобучение

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

- 3 Переобучение и регуляризация**
 - L1/L2
 - dropout
 - batchnorm

- 4 Взрыв и затухание градиентов**
 - повторяем градиентный спуск
 - инициализация

Домашние задания

 - 1 Подбор гиперпараметров

Цель: Используя валидационное множество подобрать оптимальные гиперпараметры нейросети.

- 5 Автокодирование**
 - PCA vs NN
 - разреженный автокодировщик

- 6 Вариационный автокодировщик**
 - повторяем KL
 - VAE

- 7 Решение домашних заданий**

получить ответы на вопросы.

1	Сверточные сети. Классифицируем MNIST	<ul style="list-style-type: none">- операции: свертка и пуллинг- страйд, паддинг, разреженная свертка- ReLU <hr/>
2	Сверточные сети. fine-tuning	<p>переиспользуем нейросеть на новой задаче</p> <p>Домашние задания</p> <ol style="list-style-type: none">1 Катастрофическое забывание <p>Цель: Проверить влияние fine-tuning на исходную модель</p> <ol style="list-style-type: none">1. Дообучить готовую модель на imagenette2 и проверить качество2. Сохранить последний слой обученной модели и заменить его на новый для CIFAR102. Дообучить модель решать датасет CIFAR10 и проверить качество3. Вернуть оригинальный последний слой модели и проверить качество на imagenette24. Дообучить только последний слой на imagenette2 и проверить удалось ли добиться исходного качества. <hr/>
3	Обучение с подкреплением. Q-learning для TicTacToe	<hr/>
4	GANs	<p>генерация объектов</p> <p>условная генерация</p> <hr/>
5	Рекуррентные сети	<ul style="list-style-type: none">- анализ тональности- посимвольная генерация <hr/>
6	Автокодировщики 2	<ul style="list-style-type: none">- Деконволюция- AAE <p>Домашние задания</p> <ol style="list-style-type: none">1 Условный состязательный автокодировщик <p>Цель: Реализовать условный AAE.</p> <p>Используя архитектуру с занятия, добавить на вход декодеру и дискриминатору вектор класса. Обучить полученную нейросеть. Сделать отчет об автокодировании, переносе "стиля" и условной генерации.</p> <hr/>

- | | | |
|---|---|---|
| 1 | Выбор темы и организация проектной работы | выбрать и обсудить тему проектной работы;
спланировать работу над проектом;
ознакомиться с регламентом работы над проектом.

Домашние задания

1 Проектная работа |
| 2 | Консультация по проектам и домашним заданиям | получить ответы на вопросы по проекту, ДЗ и по курсу. |
| 3 | Защита проектных работ | защитить проект и получить рекомендации экспертов. |