**Дисциплина: Глубокое обучение (осенний семестр 2023-2024 уч. г.)**

**ПИ, лектор С.В. Макрушин**

**Теоретические вопросы. 1 вопрос в билете (1й вопрос – максимум 20 баллов).**

**Основы глубокого обучения**

1. Модель перцептрона. Проблема линейно неразделимых множеств и ее решение. Логика построения многослойных ИНС. Линейные слои (Linear Layers) в PyTorch.

Описать модель, в системе базовая пороговая функция активации, а также сигмоиду

Изображение выглядит как текст, снимок экрана, линия, диаграмма

Автоматически созданное описание

Изображение выглядит как текст, снимок экрана, Шрифт, диаграмма

Автоматически созданное описание

Изображение выглядит как снимок экрана, текст, линия, диаграмма

Автоматически созданное описание

Объяснить что почему и как работает.

Изображение выглядит как текст, линия, Шрифт, диаграмма

Автоматически созданное описание

При помощи однослойного перцептрона невозможно решить задачу линейного разделения линейнонеразделимого множества.

Решение:  
Изображение выглядит как текст, линия, диаграмма, Шрифт

Автоматически созданное описание

Проблема состояла в том, что тогда не умели обучать двухслойные перцептроны. Тогда не был сплетенным подход обратного распространения ошибки. Вся проблема решается с помощью обратного распространения ошибки.

Изображение выглядит как текст, снимок экрана, линия, диаграмма

Автоматически созданное описание

Пояснить каждый слой

Объявление сигнатуры (параметров функции) в pytorch (имя функции, ее параметры с объяснением, что каждый из параметров обозначает, какие возвращаемые значения), пример объявления и примения.

1. Функции активации. Требования к функциям активации Популярные функции активации. Слои нелинейной активации (Non Linear Activations) в PyTorch.

Изображение выглядит как текст, Шрифт, программное обеспечение, снимок экрана

Автоматически созданное описание

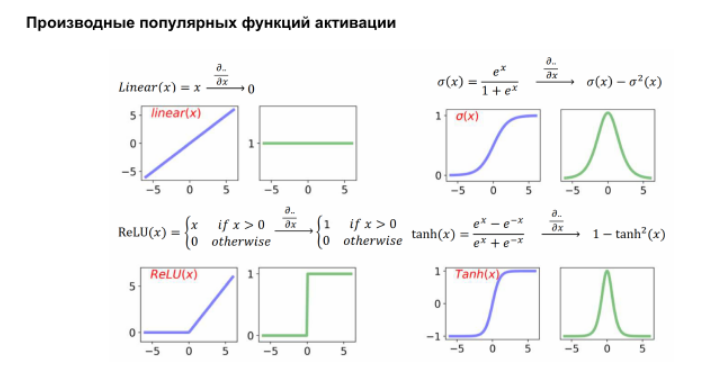
Объяснить, почему плоха обычная пороговая функция

Изображение выглядит как текст, линия, диаграмма, Шрифт

Автоматически созданное описание

+ желательно графики производных

Объяснить, какие функции наиболее популярны (ReLU), почему



1. Глубокое обучение. «Вторая весна искусственного интеллекта» и ее причины.

Что это такое и причины перехода к глубокому обучению.

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Зима ИИ – проблема линейно неразделимых множеств.

Первая весна ИИ – обратное распространение ошибки.

Изображение выглядит как текст, снимок экрана, Шрифт, диаграмма

Автоматически созданное описание

В 2012г AlexNet стал яркой причиной к тому, что всё сообщество ML повернулось лицом к глубоким нейронным сетям

1. Линейное отображение (операции с линейными операторами). Векторно-матричное дифференцирование.

Для того, чтобы выполнять операции с градиентами и тд, мы должны уметь брать производные сложных функций.

Изображение выглядит как текст, снимок экрана, Шрифт, письмо

Автоматически созданное описание

1. Проблема поиска градиента в общей логике обучения нейронной сети. Градиент функции многих переменных. Методы вычисления.

Градиент нужен для того, чтобы при помощи обратного распространения ошибки обучать нейронную сеть. Берем градиент ошибки и делаем шаг оптимизации для того, чтобы обновить веса.

Общая логика обучения нейронной сети:



Проблема: как так получается, что, имея функцию ошибки, используя градиент для преобразований в многослойной ИНС, мы можем распространить эту ошибку на места всех слоев глубокой нейронной сети.

Эта проблема в свое время остановила развитие многослойных ИНС, потому что не знали, как их обучать.

Проблема: как рассчитать градиент от функции нескольких переменных с учетом того, что функция ошибки – скалярная функция.

С помощью математики расписать ядро решения задачи поиска градиента и метода обратного распространения ошибки.

Решение: придумали, что надо получать loss и рассчитывать градиент, а далее протаскивать его до весов любого из слоев.

Изображение выглядит как текст, снимок экрана, Шрифт, документ

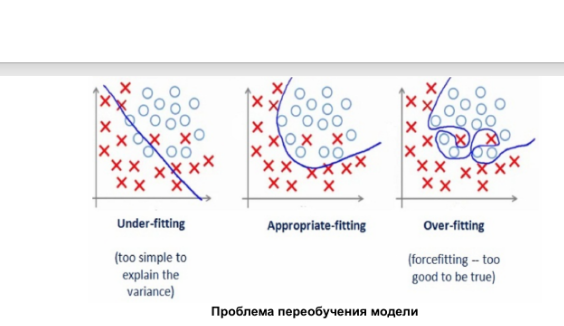
Автоматически созданное описание

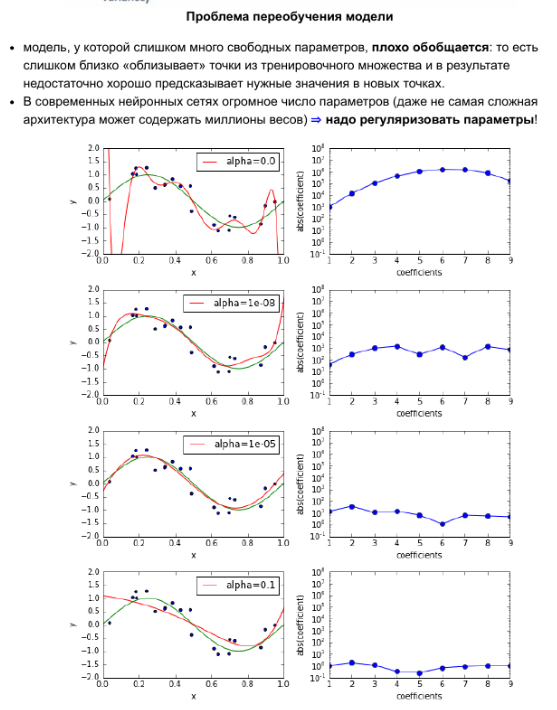
Изображение выглядит как текст, снимок экрана, Шрифт, документ

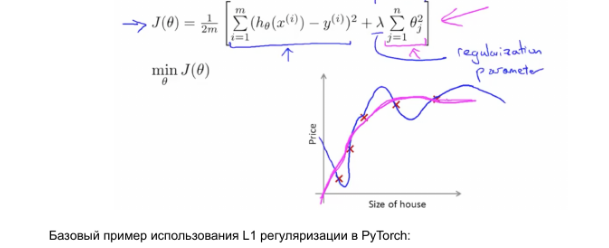
Автоматически созданное описание

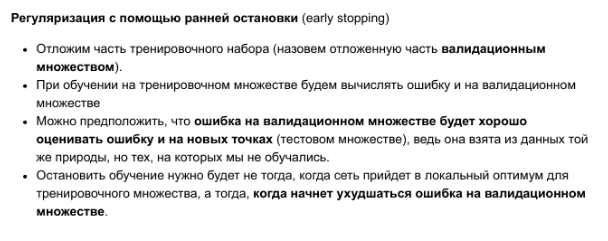
1. Кросс-валидация. Выборки train, validation, test. Проблема переобучения. Ранняя остановка.

3 лекция.









1. Преобразование Softmax и функция потерь Cross Entropy loss.

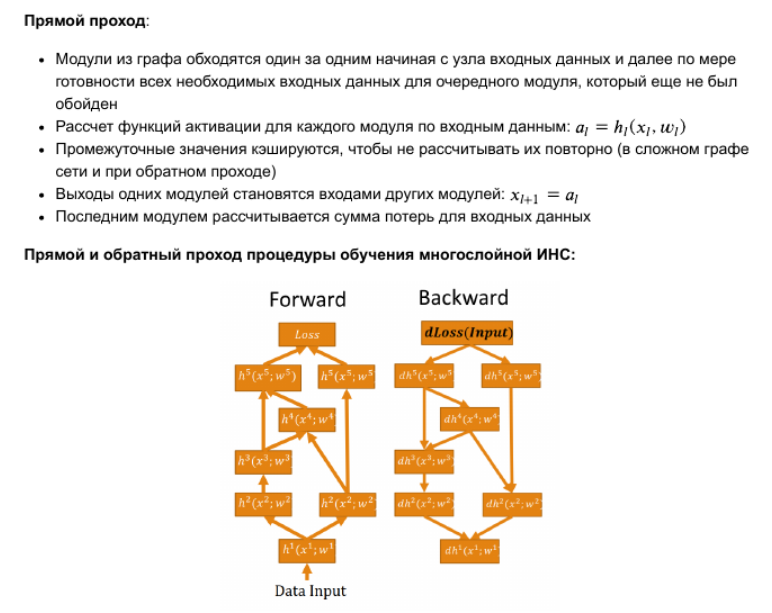
Softmax – формула, в каких случаях применяется (случай многоклассовой классификации например), какими свойствами обладает, что выглядит как вероятность и в чем преимущество софтакс перед обычным максимумом (дифференцируемость). + pytorch

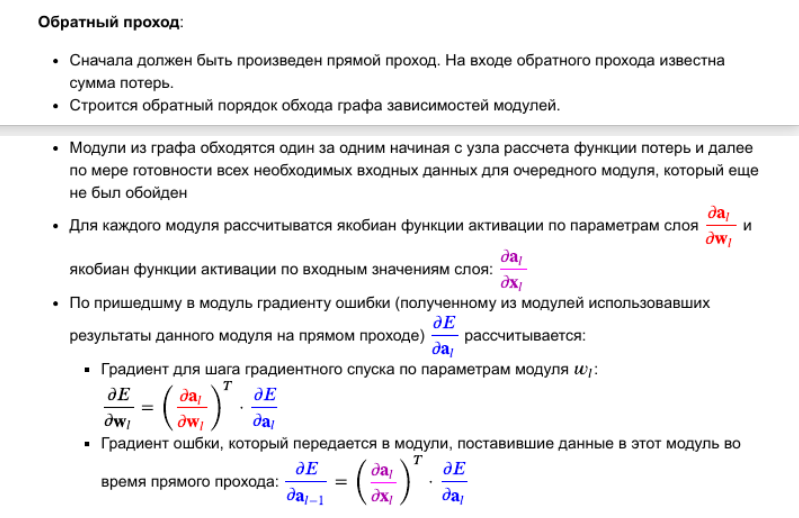
Cross Entropy loss – как она выглядит, почему она такая

1. Механизм обратного распространения ошибки. Принципиальная логика основного цикла обучения нейронной сети в PyTorch.  Слои функций потерь (Loss Functions) в PyTorch (не меньше трёх).



Рассказать общую логику (есть прямой проход (получили предсказания, ошибку), обратный проход. После прямого прохода помодульное выполнение обратного распространения ошибки и в каждом из модулей (слоев нейронной сети) обучать параметры этого слоя и передавать градиент ошибки более ранним слоям.





1. Дифференцируемое программирование и реализация обратного распространения ошибки. Автоматическое дифференцирование в PyTorch. Пример и применение в обучении ИНС.

2 часть 1 лекция

Как можно дифференциросовать с помощью pytorch

Определение

Автоматическое дифференцирование в pytorch

Как можно им управлять включая и выключая расчет градиента

Пример в виде простого основного цикла обучения pytorch

1. Стохастический градиентный спуск. Батчи обучающей выборки.

Оптимизация

Общий вид – loss по всей обучающей выборке

Стохастический – шаг оптимизации по одному наблюдению из обучающей выборки

В жизни используются минибатчи. Рассказать что это. В чем преимущества использования минибатчей.

Более оптимально.

1. Адаптивные методы градиентного спуска. Метод импульсов. Метод Нестерова.

Есть стохастическиё градиентный спуск, его можно улучшить.

Изображение выглядит как текст, снимок экрана, Шрифт, документ

Автоматически созданное описание

Изображение выглядит как текст, документ, снимок экрана, Шрифт

Автоматически созданное описание

Изображение выглядит как текст, снимок экрана, Шрифт, Параллельный

Автоматически созданное описание

1. Проблема инициализации весов при обучении ИНС. Инициализация Ксавье.

Что происходит, когда мы инициализируем нулями или случайными числами.

Дать формулу, объяснить, какие преимущества, технически показать какие параметры есть, что они означает, что эта инициализация даёт в практическом плане.

1. Гиперпараметры. Скорость обучения и размер батча.

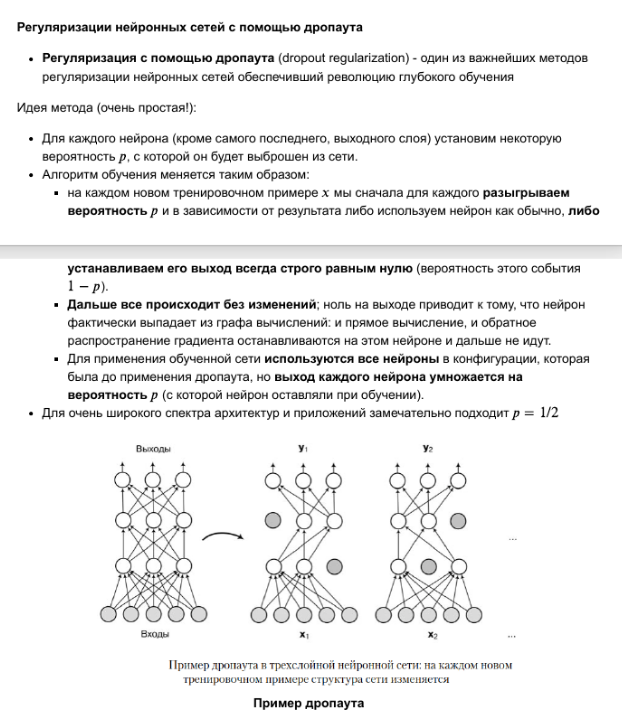
Определение что такое гиперпараметры

Какие параметры могут быть у ИНС

Что такое скорость обучения, где она используется

Что такое батчи, как они влияют на обучение и как можно управлять размером батчей, что влияет на выбор размера батчей

1. Переобучение модели и регуляризация. Принцип механизма Dropout. Слои регуляризации (Dropout Layers) в PyTorch.





1. Минбатчи – причина использования. Нормализация по мини-батчам. Слои нормализации (Normalization Layers) в PyTorch.

Рассказать про минибатчи.

Рассказать про нормализацию, что происходит без неё, к чему приводит отсутствие такой нормализации

В процессе обучения входы очередного слоя могут очень сильно поплыть

В чём проблема, с чем мы боремся, как работает нормализация по мини батчам

Как устроена в Pytorch

Изображение выглядит как текст, снимок экрана, Шрифт, документ

Автоматически созданное описание

Изображение выглядит как текст, снимок экрана, Шрифт, чек

Автоматически созданное описание

Изображение выглядит как текст, снимок экрана, Шрифт, линия

Автоматически созданное описание

Изображение выглядит как текст, Шрифт, чек, снимок экрана

Автоматически созданное описание

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

1. Многослойные сети. Граф потока вычислений. Класс nn.Module в PyTorch: назначение, основные поля и методы.

Полносвязанные сети (из 1 лекции)

Изображение выглядит как текст, снимок экрана, диаграмма, линия

Автоматически созданное описание

Граф потока вычислений

Изображение выглядит как текст, Шрифт, диаграмма, линия

Автоматически созданное описание

Изображение выглядит как текст, Шрифт, снимок экрана, алгебра

Автоматически созданное описание

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

Что такое nn.Module, как пользоваться, пример построения многослойной ИНС, как определить и что с ней делать в реальных вычислениях (в основном цикле обучения)

**Cверточные нейронные сети**

1. Специфика задач машинного обучения на изображениях. Принцип работы сверточных сетей. Преимущества сверточных сетей при решении этих задач.

Что такое изображение, что это матрица, обычно трехканальная, высокая корреляция между пикселями

Изображение выглядит как кот, домашняя кошка, текст, Мелкие и средние кошки

Автоматически созданное описание

Изображение выглядит как текст, снимок экрана, дизайн

Автоматически созданное описание

Изображение выглядит как текст, снимок экрана, графический дизайн, Графика

Автоматически созданное описание

Определение сверт НС

Изображение выглядит как текст, снимок экрана, Шрифт, чек

Автоматически созданное описание

Показать к чему приводит

Изображение выглядит как линия, дизайн, искусство

Автоматически созданное описание

Пример свертки

1. Архитектура многослойной ИНС распознавания изображений на основе сверточных сетей. Сверточные слои (Convolution Layers) и сжимающие слои (Pooling Layers) в PyTorch.

Изображение выглядит как диаграмма, снимок экрана, текст, План

Автоматически созданное описание

Изображение выглядит как текст, снимок экрана, Шрифт, алгебра

Автоматически созданное описание

Изображение выглядит как текст, снимок экрана, Шрифт, документ

Автоматически созданное описание

1. Приемы для глубокого обучения на небольших наборах изображений.

Меняем классификатор, оставляя текущий сверточный слой или когда мы замораживаем веса, кроме последних слоев (почему оставлем последние слои, что такое заморозка весов)

Почему на небольшом наборе изображений нельзя нормально обучить св ИНС

Изображение выглядит как текст, снимок экрана, Шрифт, документ

Автоматически созданное описание

Изображение выглядит как текст, Шрифт, снимок экрана, документ

Автоматически созданное описание

Изображение выглядит как текст, снимок экрана, диаграмма, Шрифт

Автоматически созданное описание

Изображение выглядит как текст, Шрифт, снимок экрана, алгебра

Автоматически созданное описание

1. Схема работы сверточной сети. Операции свертки, пулинга, общий вид сверточной сети для решения задачи классификации изображения.

Операции свертки (как в 17)

18 вопрос – схема, акцент не на описание конв и пул, а на реализации полного конвейера на PyTorch

**Рекуррентные нейронные сети**

1. Рекуррентная нейронная сеть, принципы ее обучения. Сложности применения рекуррентных нейронных сетей.

Что означает, как применяется, трудности

Изображение выглядит как текст, снимок экрана, Шрифт, диаграмма

Автоматически созданное описание

Изображение выглядит как текст, снимок экрана, Шрифт, алгебра

Автоматически созданное описание

Изображение выглядит как текст, снимок экрана, диаграмма, Шрифт

Автоматически созданное описание

Изображение выглядит как текст, Шрифт, снимок экрана, письмо

Автоматически созданное описание

Зачем нужны RNN: Входная последовательность произвольной длины и нужно либо классифицировать либо выполнить перевод

Изображение выглядит как снимок экрана, диаграмма, текст, дизайн

Автоматически созданное описание

1. Модуль LSTM.

Это рекурентный модуль, зачем нужен, чем плох обычный RNN модуль, что LSTM дает, чем принципиалььно отличается, какую проблему RNN решает

Изображение выглядит как текст, снимок экрана, письмо

Автоматически созданное описание

Изображение выглядит как текст, диаграмма, План, Параллельный

Автоматически созданное описание

1. Модуль GRU.

Отличие от RNN и LSTM, там меньше параметров (за счет чего) и с точки зрения применения (отличается кол-во входов и выходов)

Изображение выглядит как диаграмма, снимок экрана, линия, мультфильм

Автоматически созданное описание

Изображение выглядит как текст, документ, черно-белый, Шрифт

Автоматически созданное описание

1. Класс torch.nn.LSTM и torch.nn.GRU в PyTorch.

Как устроены и как применять

Изображение выглядит как текст, письмо, Шрифт, снимок экрана

Автоматически созданное описание

1. Продвинутые RNN-архитектуры: BiLSTM, улучшенные seq2seq модели машинного перевода на основе RNN. (что такое, в чем преимуществоИзображение выглядит как текст, Шрифт, линия, снимок экрана

   Автоматически созданное описаниеИзображение выглядит как текст, снимок экрана, письмо, Шрифт

   Автоматически созданное описание

**Механизм внимания и модуль Transformer и BERT**

1. Механизм Attention. Пример использования Attentinon.

Как решалась классически и в чем ее проблема

Проблема в том, что вход и выход могут оказываться очень далеко друг от друга и информации чоень сложно пройти на такое расстояние

1. Модуль Transformer: общая архитектура Transformer, работа self-attention и multi-head attention.
2. Модуль Transformer: общая архитектура Transformer, специфика работы декодера и его увязки с энкодером, позиционное кодирование в Transformer.
3. Модель BERT.

**Автоэнкодеры и генеративные модели**

1. Векторная интерпретация весов скрытого слоя многослойного перцептрона. Задача восстановления входного сигнала после сжатия в скрытом слое ИНС.
2. Автоэнкодеры. Линейный автоэнкодер и его аналог в статистических методах. Нелинейные и глубокие автоэнкодеры. Области значений латентного пространства. Примеры интерполяции и экстраполяции на многообразиях области значений латентных пространств автоэнкодеров.
3. Задачи дискриминативных и генеративных моделей, сравнение моделей. Классификация генеративных моделей. Прикладные задачи для генеративных моделей в области компьютерного зрения.
4. Архитектура GAN: описание общей архитектуры, модели обучения и архитектур глубоких моделей в GAN. Принцип подхода к обучению в GAN на примере обучения одномерной функции распределения.
5. Модель вариационного автоэнкодера (VAE): общие и специфические цели модели, общая архитектура модели и баейсовское моделирование. Функция потерь VAE и специфика получаемых скрытых представлений.
6. Модель вариационного автоэнкодера (VAE): баейсовское моделирование, функция ошибки на основе метода максимального правдоподобия, причины построения ELBO и метод построения этой оценки.
7. Denoising diffusion models: общий принцип работы, описание прямого процесса и основные принципы описания обратного процесса.