

Генеративные модели в машинаном обучении

Лекция 8
Оптимизация моделей

Михаил Гущин

mhushchyn@hse.ru

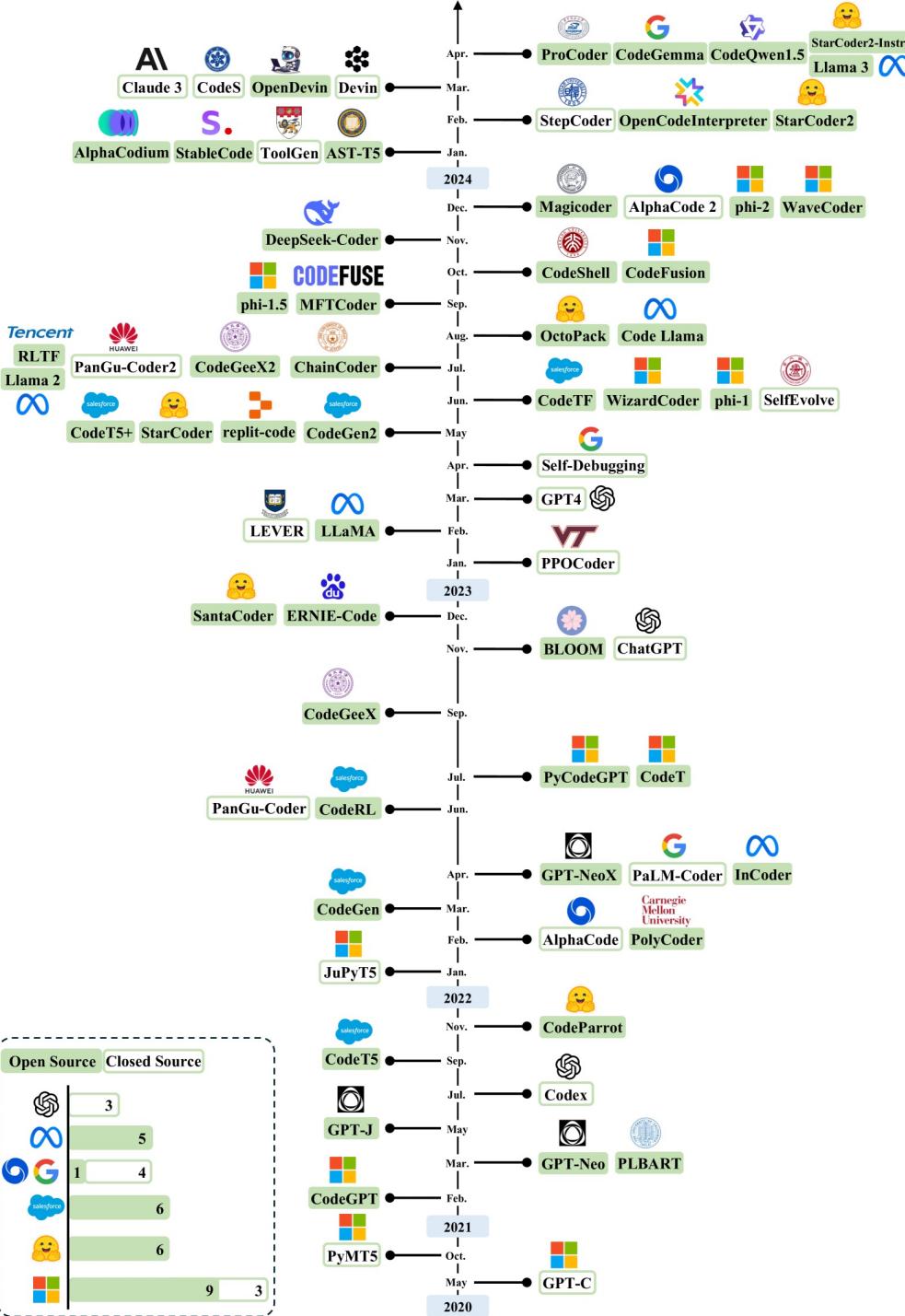
НИУ ВШЭ, 2024



НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
УНИВЕРСИТЕТ

Зачем нужна оптимизация

Обзор LLM



Источник: <https://arxiv.org/html/2406.00515v1>

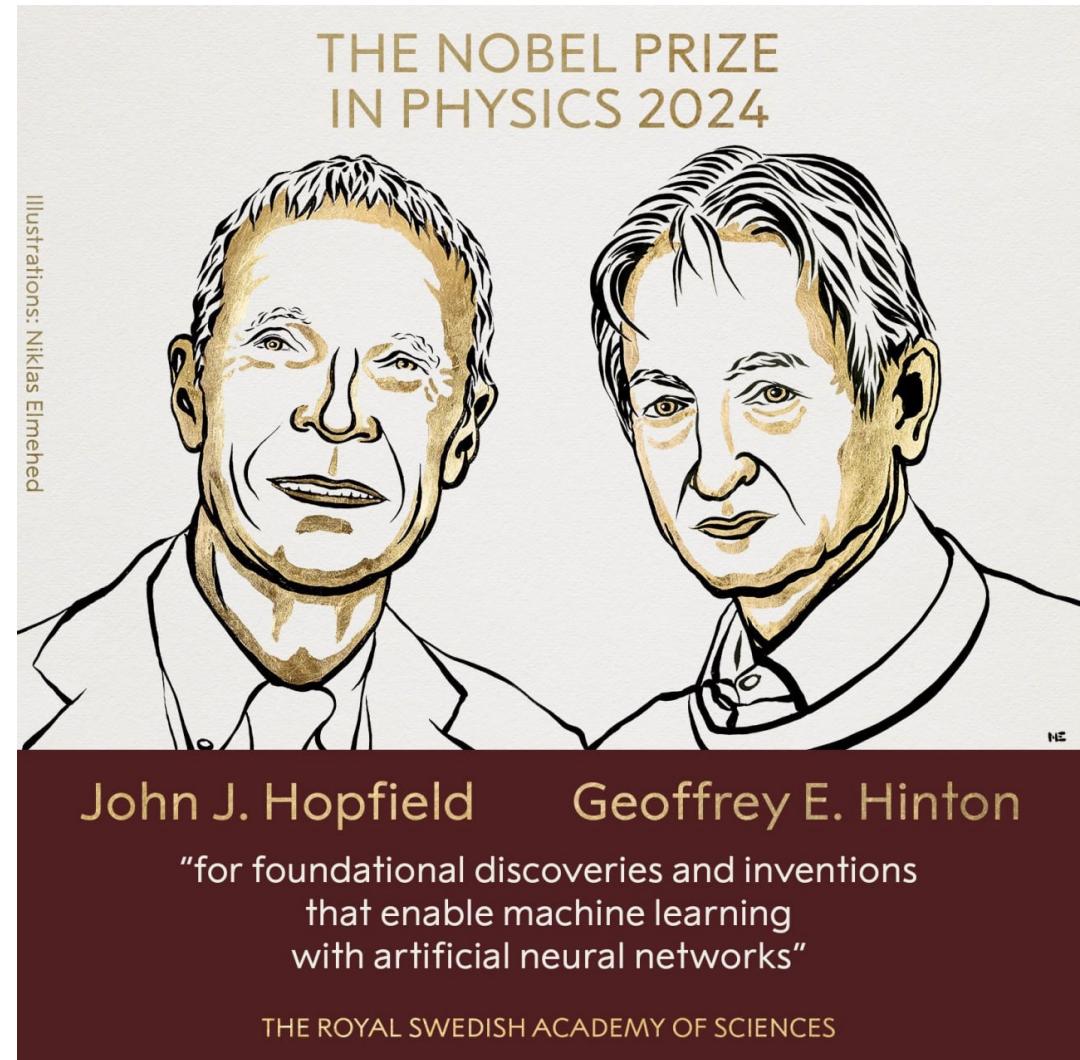
Нобелевская премия по физике 2024

ИИ – один из основных инструментов анализа данных в физике высоких энергий и астрономии.

ИИ используется в физике с конца 1980х.

Пример списка применений ИИ (около 1000) на Большом адронном коллайдере:

<https://github.com/iml-wg/HEPML-LivingReview>

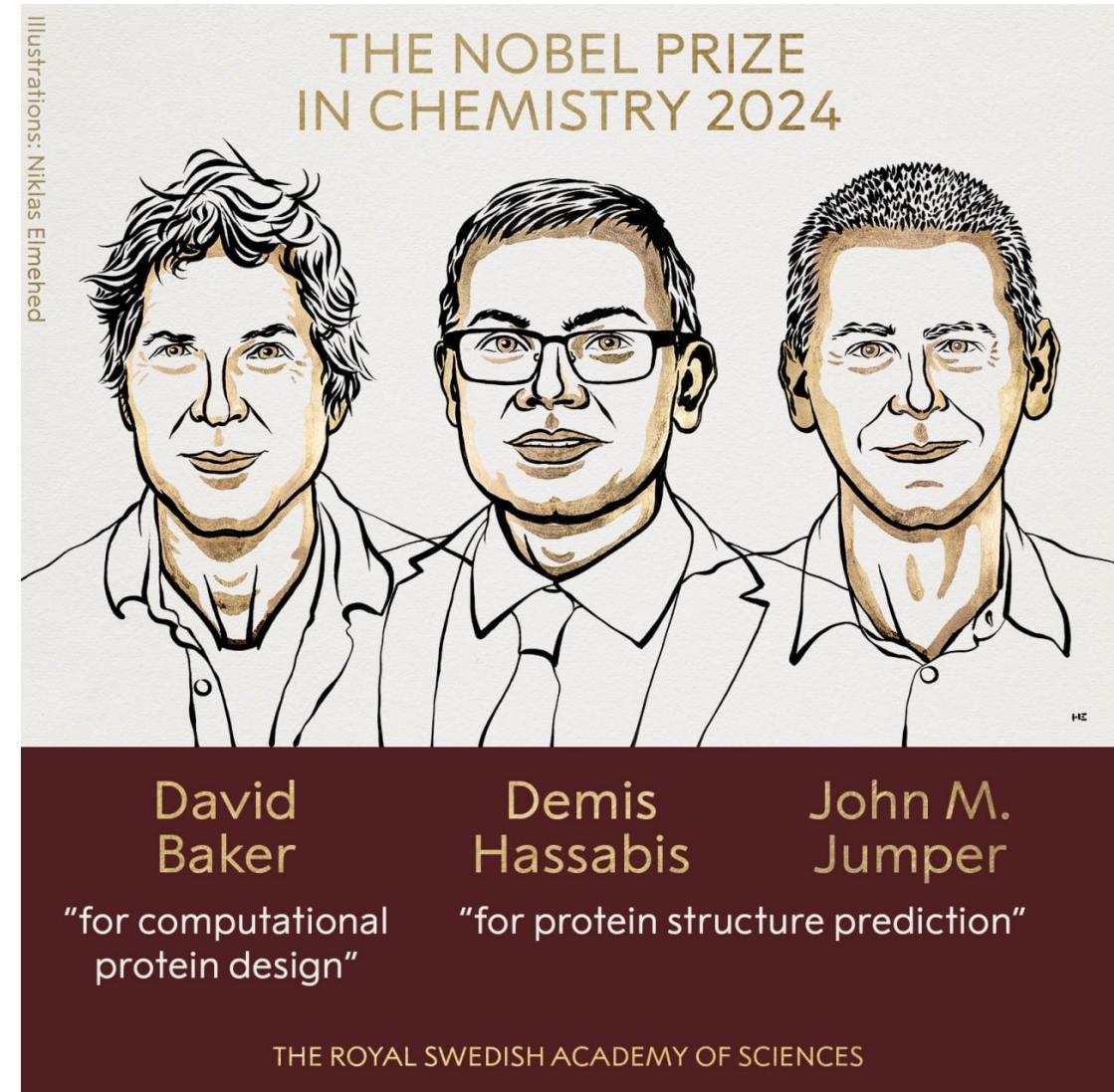


Нобелевская премия по химии 2024

«Лауреатами Нобелевской премии по химии за 2024 год стали Дэвид Бейкер — «за вычислительный дизайн белков», а также Демис Хассабис и Джон Джампер из дочерней компании Google — Google DeepMind — «за предсказание структуры белков», сообщает пресс-служба Нобелевского комитета.»

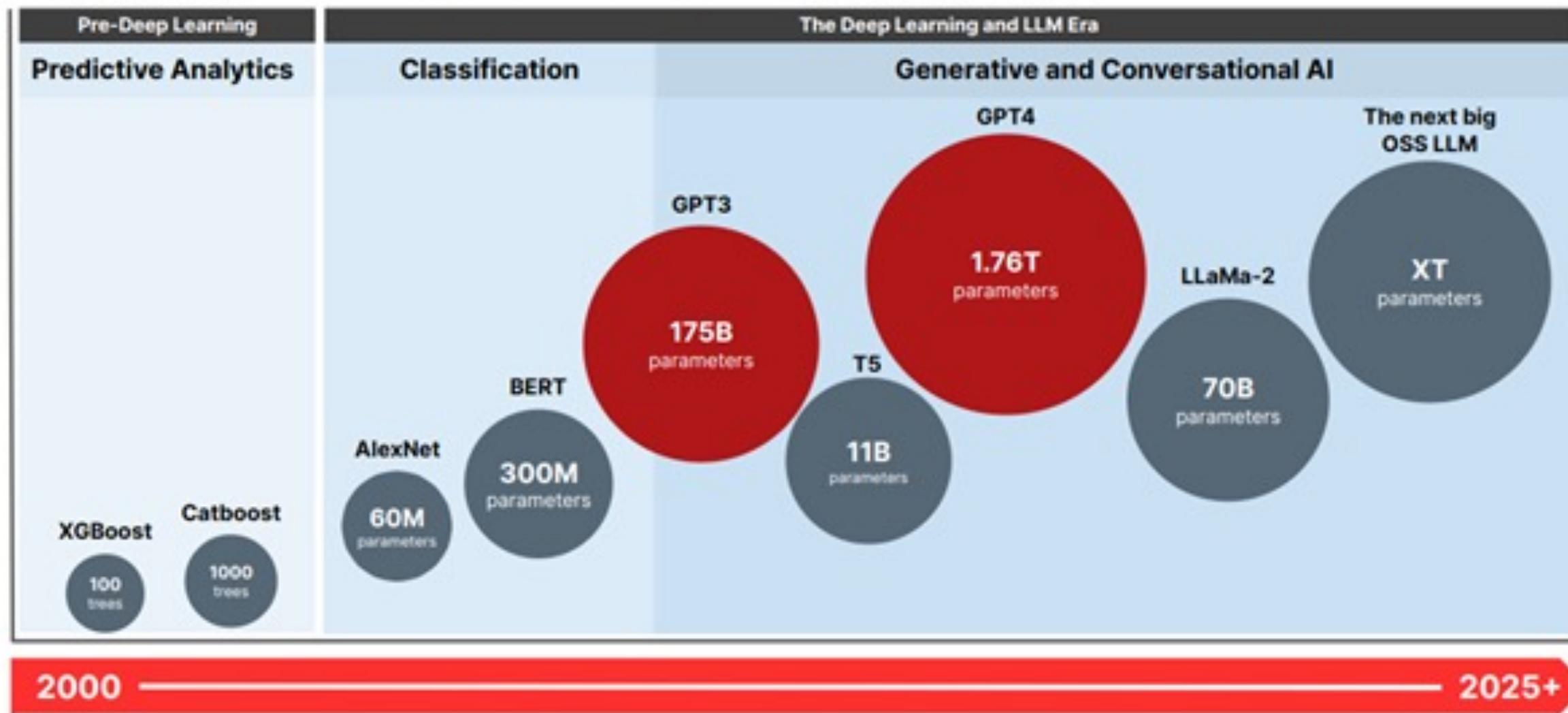
«Хассабис и Джампер разработали модель ИИ для решения 50-летней проблемы — предсказания сложных структур белков. Их модель под названием AlphaFold2 смогла предсказать структуру практически всех 200 млн белков, которые идентифицировали исследователи.»

Ссылка: <https://www.rbc.ru>

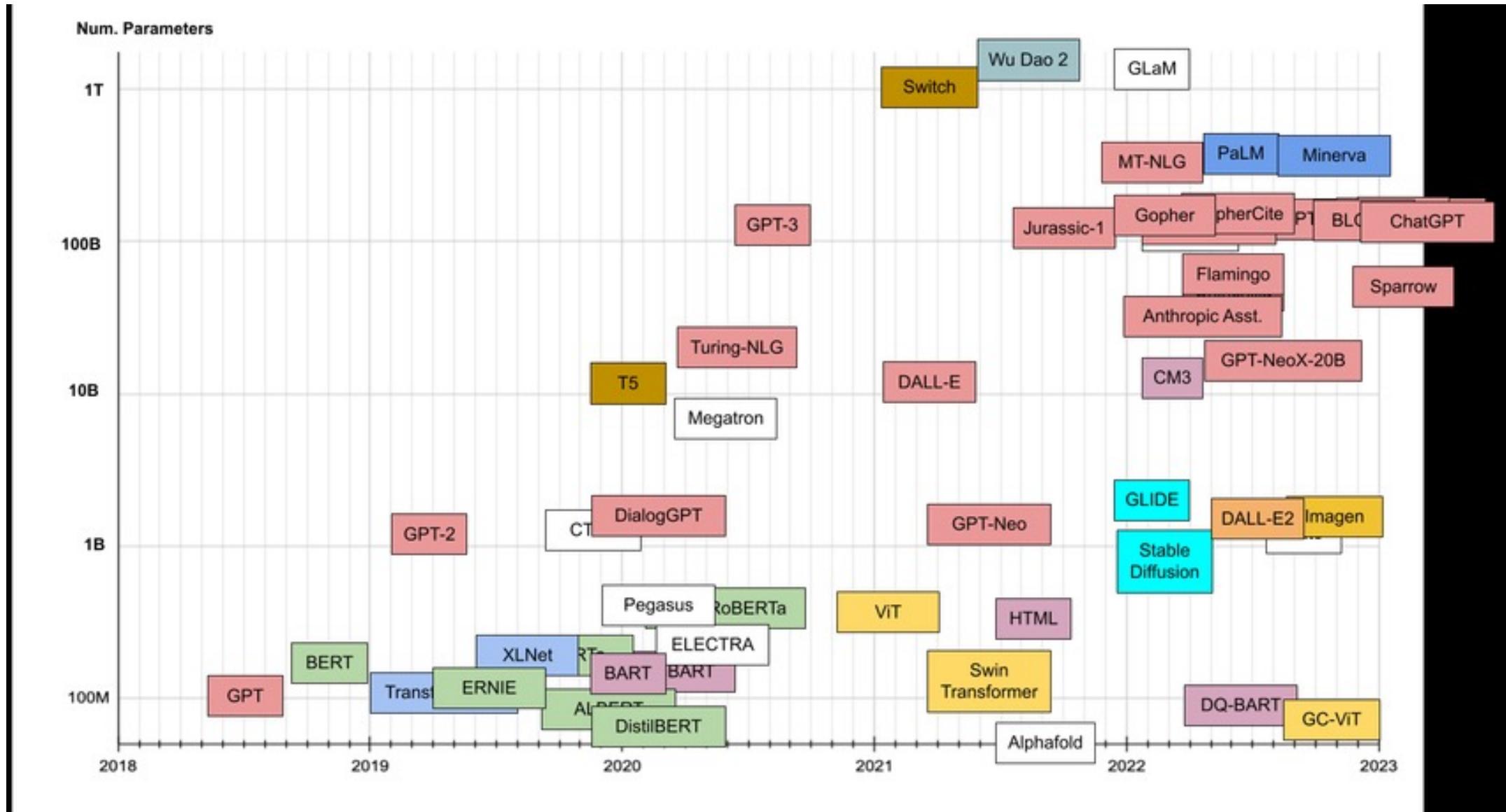


Размер моделей ИИ

Rapid advancements in general intelligence



Размер моделей ИИ



СКОЛЬКО СТОИТ ОБУЧЕНИЕ

LLM Training Costs on MosaicML Cloud

Model	Billions of Tokens (Compute-optimal)	Days of Train on MosaicML Cloud	Approx.Cost on MosaicML Cloud
GPT-1.3B	26B	0.14	\$2,000
GPT-2.7B	54B	0.48	\$6,000
GPT-6.7B	134B	2.32	\$30,000
GPT-13B	260B	7.43	\$100,000
GPT-30B*	610B	35.98	\$450,000
GPT-70B**	1400B	176.55	\$2,500,000

Costs and benefits of your own LLM | by Maciej Tatarek | Medium

Источник: <https://www.metadialog.com/>

СКОЛЬКО СТОИТ ОБУЧЕНИЕ

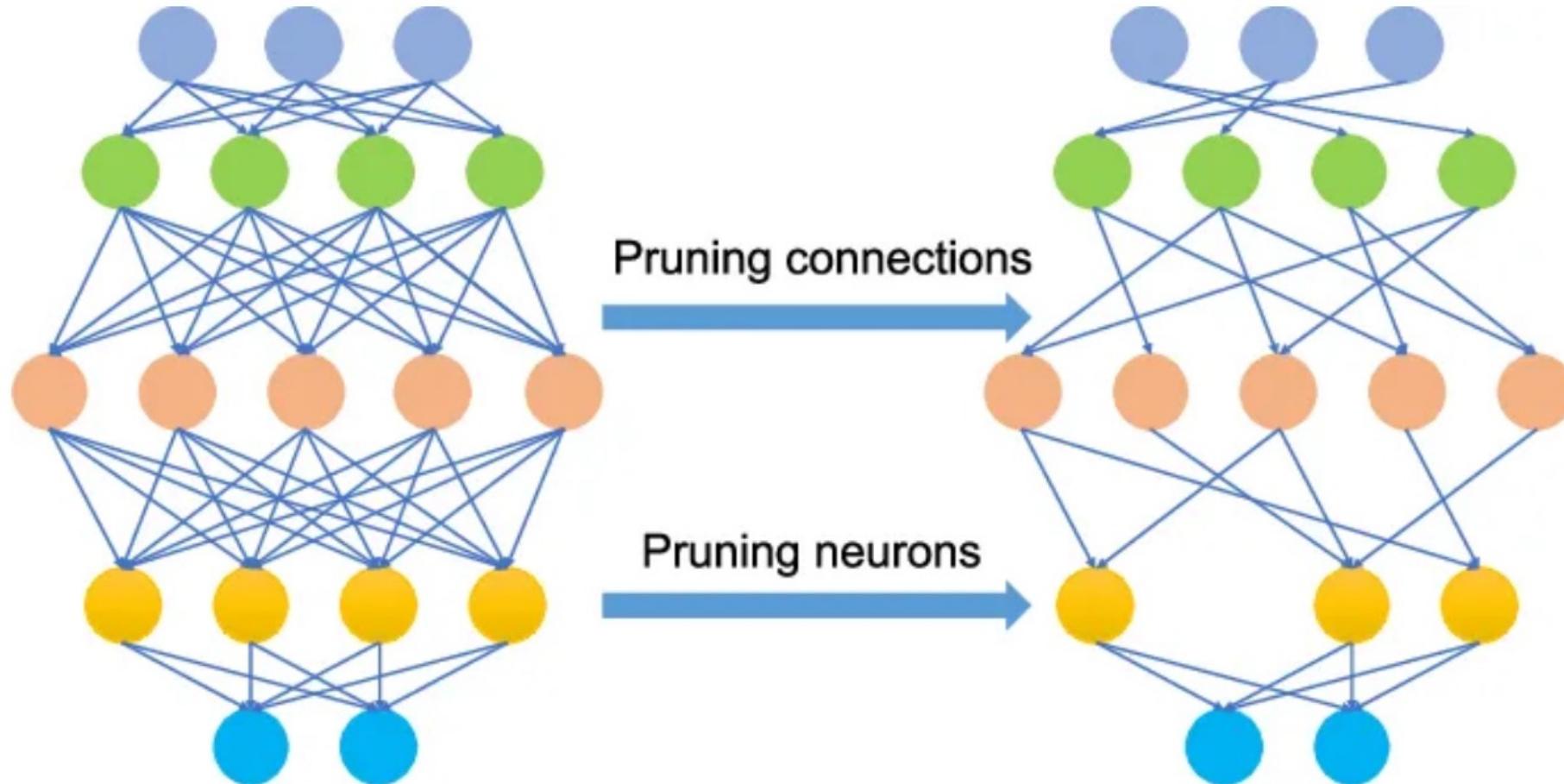
Optimal LLM Training Cost				
Model	Size (# Parameters)	Tokens	GPU	Optimal Training Compute Cost
MosaicML GPT-30B	30 Billion	610 Billion	A100	\$ 325,855
Google LaMDA	137 Billion	168 Billion	A100	\$ 368,846
Yandex YaLM	100 Billion	300 Billion	A100	\$ 480,769
Tsinghua University Zhipu.AI GLM	130 Billion	400 Billion	A100	\$ 833,333
Open AI GPT-3	175 Billion	300 Billion	A100	\$ 841,346
AI21 Jurassic	178 Billion	300 Billion	A100	\$ 855,769
Bloom	176 Billion	366 Billion	A100	\$ 1,033,756
DeepMind Gopher	280 Billion	300 Billion	A100	\$ 1,346,154
DeepMind Chinchilla	70 Billion	1,400 Billion	A100	\$ 1,745,014
MosaicML GPT-70B	70 Billion	1,400 Billion	A100	\$ 1,745,014
Nvidia Microsoft MT-NLG	530 Billion	270 Billion	A100	\$ 2,293,269
Google PaLM	540 Billion	780 Billion	A100	\$ 6,750,000

Оптимизация

- ▶ Методы повышения эффективности обучения нейронных сетей преследуют несколько целей:
 - ускорение обучения
 - сокращение объемов вычислительных ресурсов
 - уменьшение потребности в объемах обучающих данных
 - сохранение, а иногда и улучшение качества результатов модели



Прунинг нейронных сетей



Прунинг весов

При прунинге весов некоторым параметрам устанавливается значение в ноль, тем самым создается разреженная сеть.

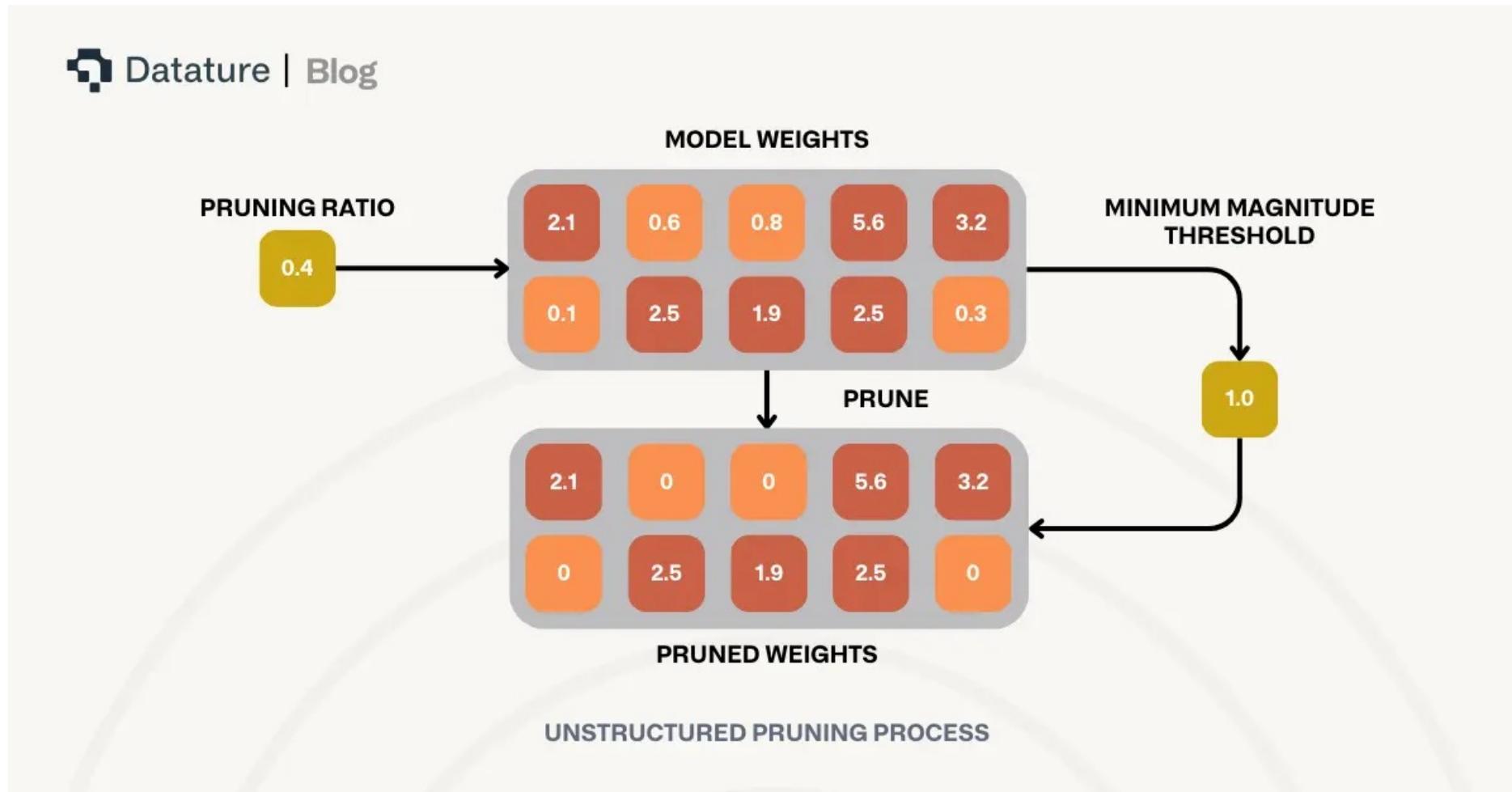
- ▶ Это уменьшает количество параметров модели, при этом сохраняется целостность архитектуры. Мы получаем сеть с меньшим количеством параметров.
- ▶ Однако, для эффективности обучения такой сети требуются разреженные вычисления, для которых необходима поддержка оборудования (то есть специальные инструменты, которые не всегда в наличии).

Прунинг весов

Варианты решения:

- ▶ Можно обрезать (занулить) те веса модели, которые и так имеют достаточно низкое значение по модулю
- ▶ Для преднамеренного обучения модели уменьшать или загулять незначимые веса используют L1 или L2 регуляризацию

Пример



Прунинг нейронов

При удалении из сети нейрона удаляются также все входные и выходные ребра, то есть целый набор весов.

- ▶ Такой способ уменьшает архитектуру сети и позволяет выполнять плотные, более оптимизированные вычисления. Можно работать без разреженных вычислений, и такие вычисления лучше поддерживаются на оборудовании.
- ▶ Однако такой прунинг способен навредить нейронной сети — удалить важные нейроны.

Прунинг нейронов

Варианты решения:

- ▶ При запуске обучения мы можем собрать некоторую статистику активаций. Те нейроны, которые выдают низкие значения, редко используются сетью и следовательно могут быть удалены.
- ▶ Кроме величины весов можно смотреть на схожесть с другими выходами текущего слоя: если значения двух выходов статистически повторяются, то можно предположить, что они делают одно и то же. Следовательно, можно удалить один из них, и функциональность при этом не изменится.

Типы прунинга



PRUNING APPROACH

PROS

CONS

TRAIN-TIME PRUNING

More efficient models since they are trained with the objective of sparsity in mind

Pruning decisions are made alongside the consideration of model parameter optimization

Makes training process more complex

Change in pruning parameters may require retraining of the whole model

POST-TRAINING PRUNING

Simpler to implement

Pruning parameters can be easily adjusted based on inference requirements

May require fine-tuning to regain performance in the case of accuracy degradation

Pruning decisions are user-defined and may not be optimal

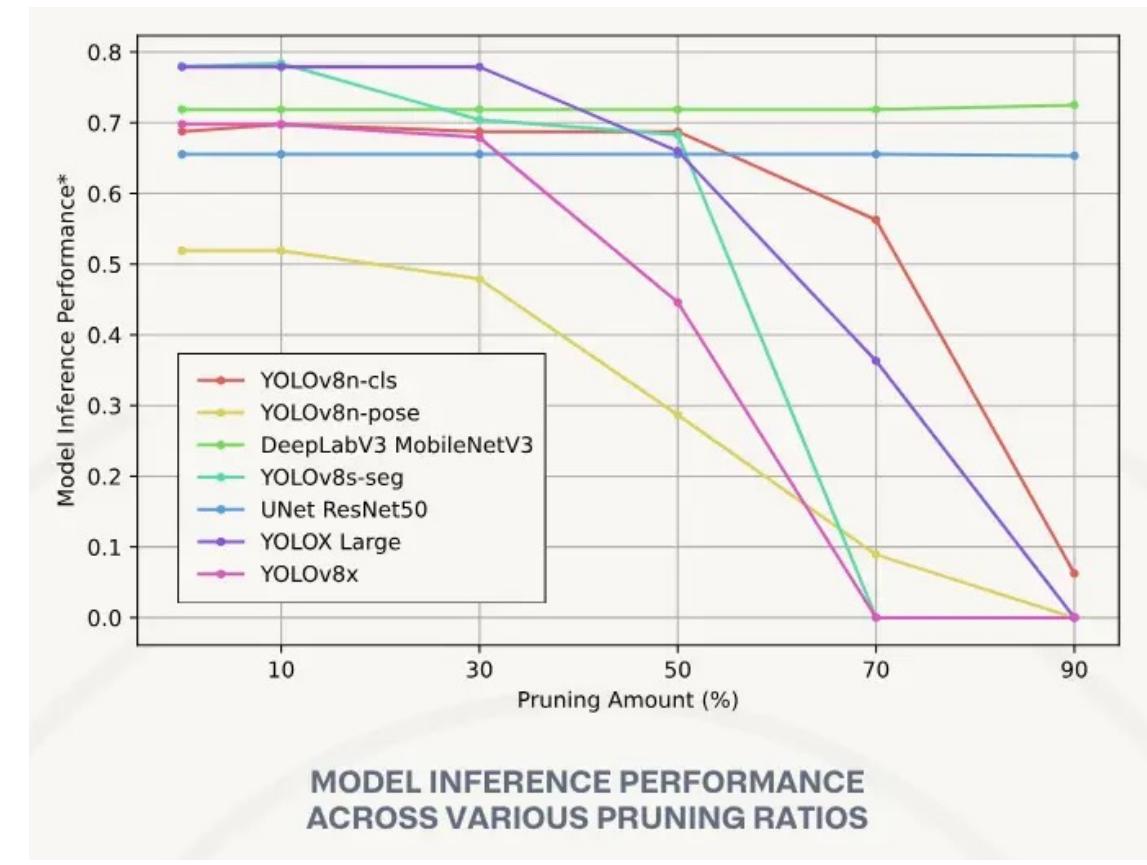
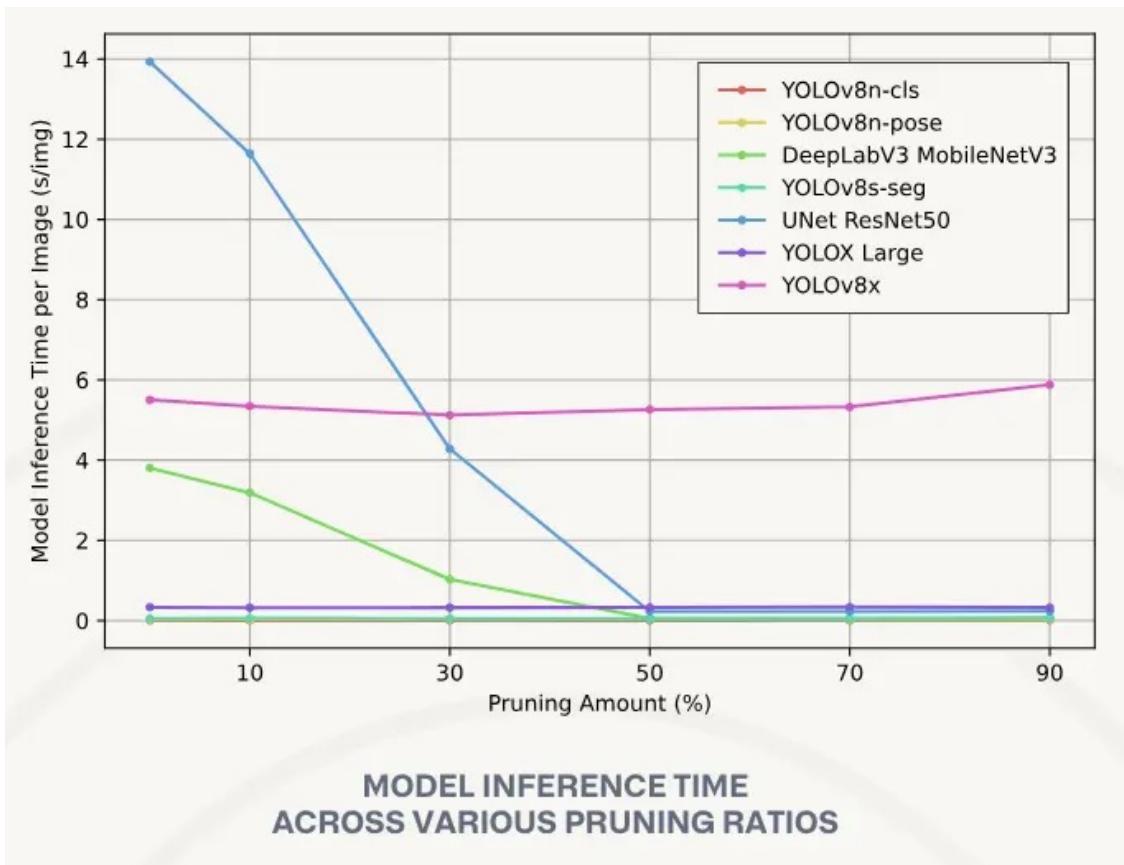
TRADEOFFS OF DIFFERENT PRUNING APPROACHES

Типы прунинга

PRUNING SCOPE		
EXECUTION	LOCAL	GLOBAL
PROS	Focuses on individual weights at a more fine-grained level Simpler to implement Typically faster to execute More measured approach	Considers entire network with a more big-picture approach Accounts for more context during pruning Potentially higher levels of compression
CONS	More likely to result in degraded model performance	Requires more computational resources

TRADEOFFS OF DIFFERENT PRUNING SCOPES

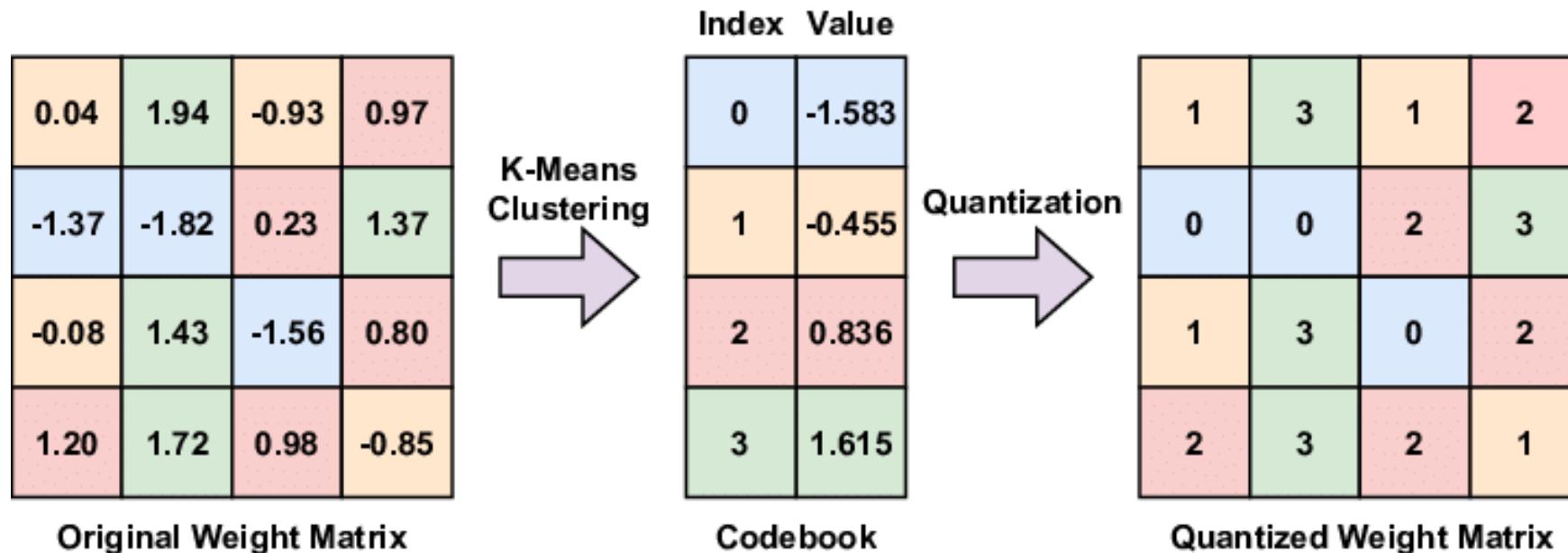
Примеры



Квантизация

Квантизация

Квантизация (квантование) модели — это метод оптимизации нейронных сетей, при котором данные модели — как параметры сети, так и функции активации — преобразуются из представления с плавающей точкой в представление с более низкой точностью.



Преимущества квантизации

- ▶ При обработке 8-битных целочисленных данных графические процессоры NVIDIA используют более быстрые и дешевые 8-битные тензорные ядра для вычисления операций свертки и умножения матриц. Это дает большую пропускную способность вычислений.
- ▶ Уменьшение объема памяти означает, что модель требует меньше места для хранения, меньше параметров для обновления, использование кэша выше и т.д.

Как делать квантизацию

Снижение точности параметров может легко навредить точности модели, поэтому квантизацию нужно делать аккуратно.

- ▶ 32-битный тип с плавающей точкой хранит около 4 миллиардов чисел в интервале $[a_{\min}, a_{\max}] = [-3.4 \cdot 10^{38}, 3.40 \cdot 10^{38}]$
- ▶ Мы хотим перевести числа из этого интервала в 8-битные целые числа, то есть в диапазон $[-128, 127]$.

Как делать квантизацию

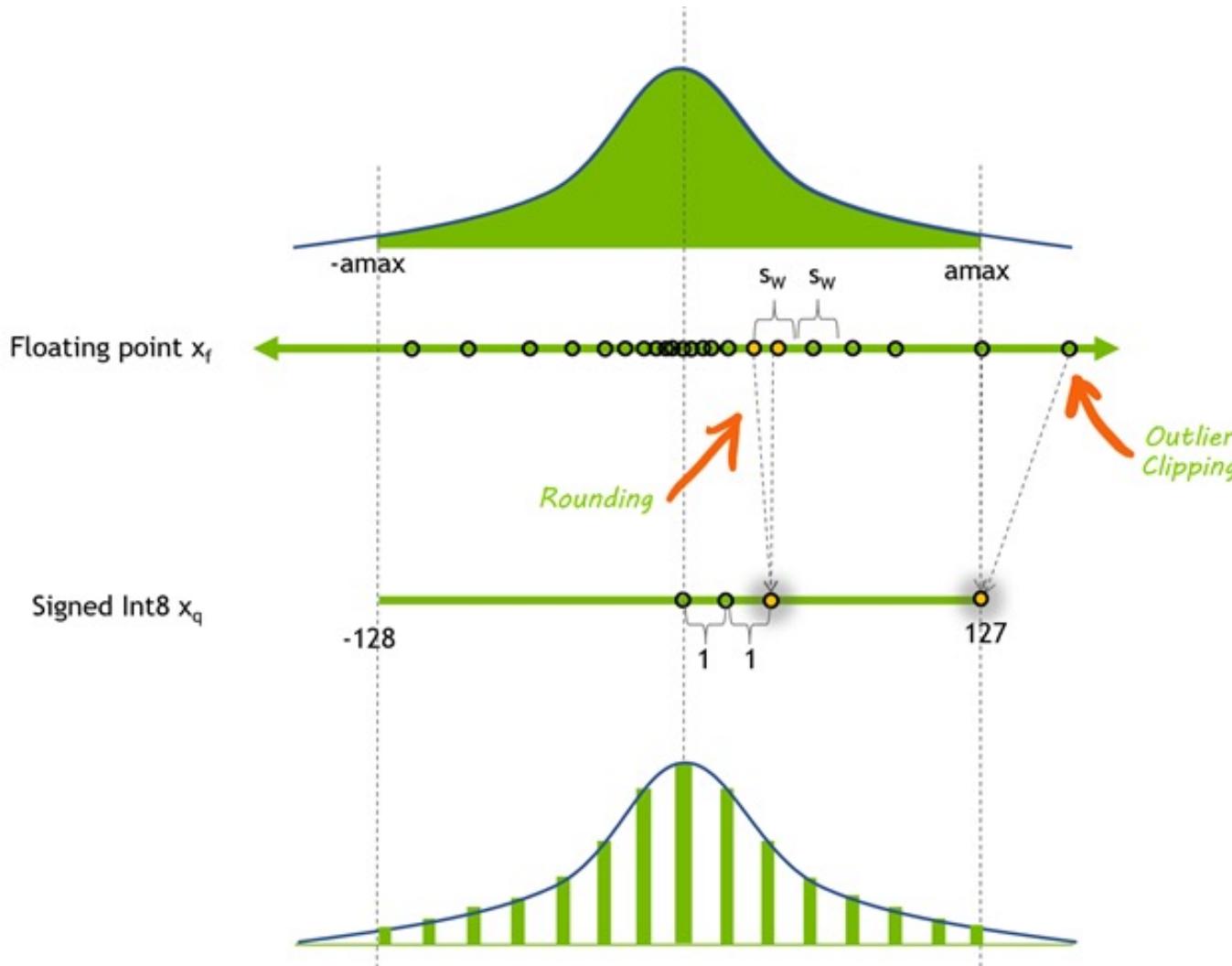
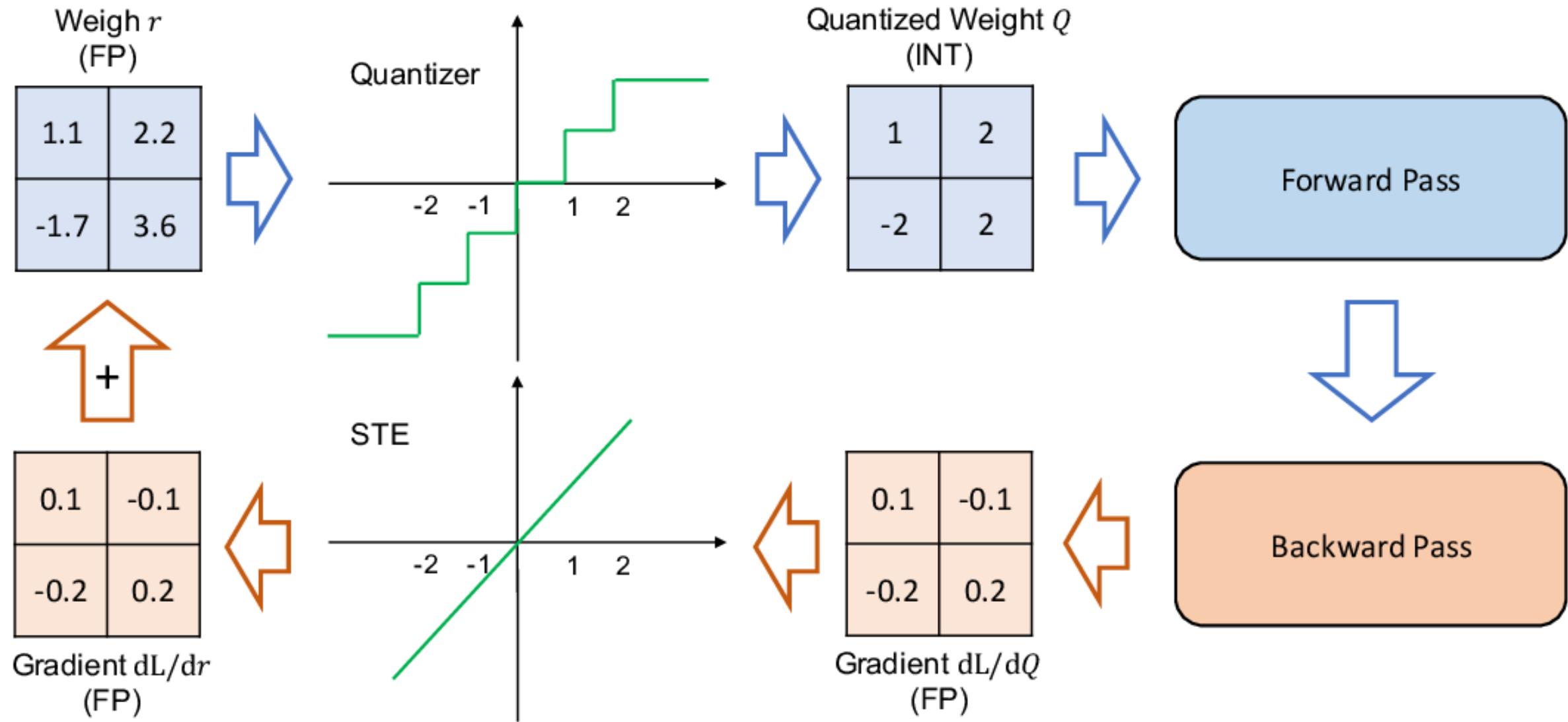


Figure 1. 8-bit signed integer quantization of a floating-point tensor x_f . The symmetric dynamic range of x_f $[-amax, amax]$ is mapped through quantization to $[-128, 127]$.

Квантизация и обучение

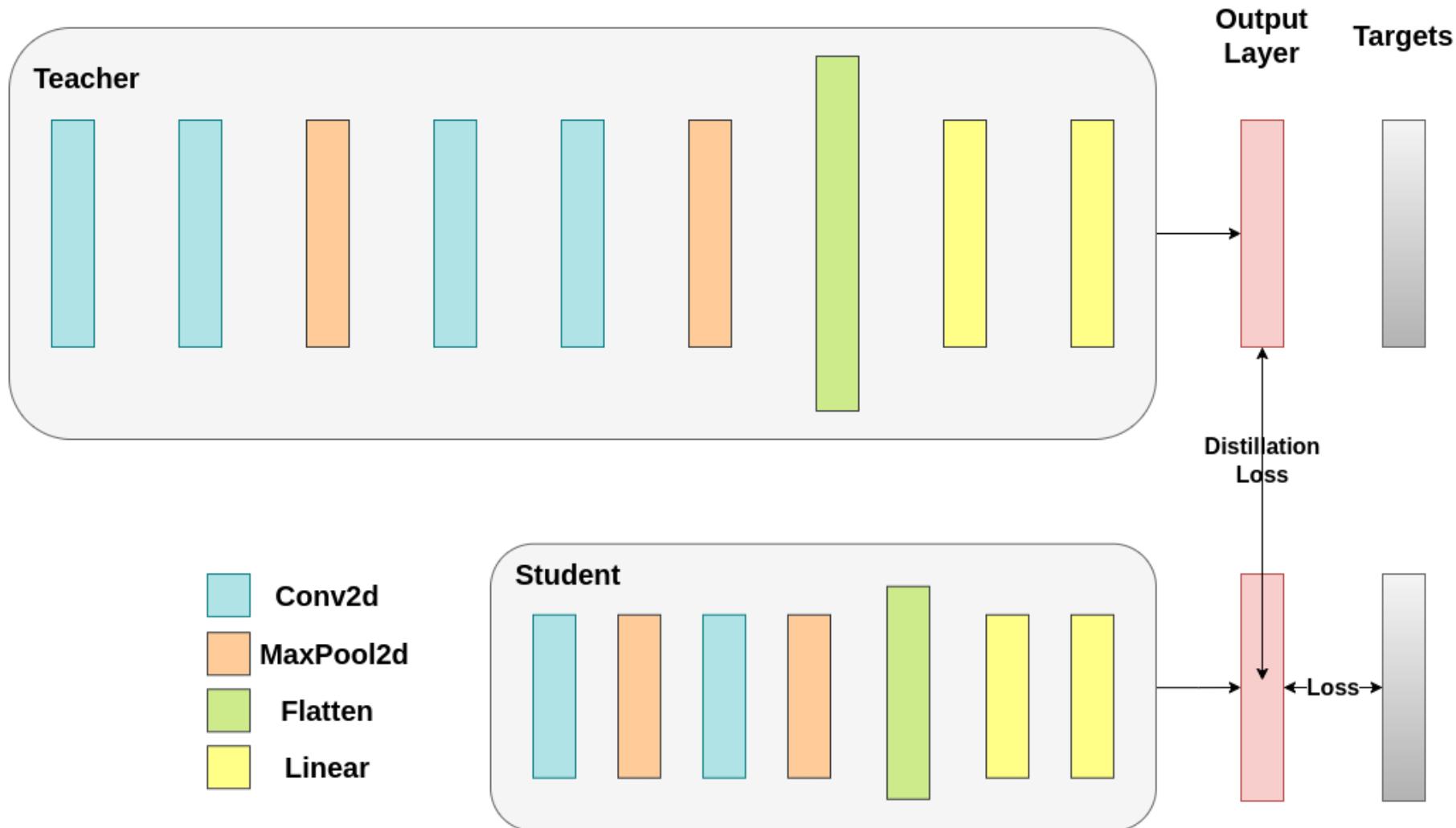


Дистилляция

Дистилляция нейронных сетей

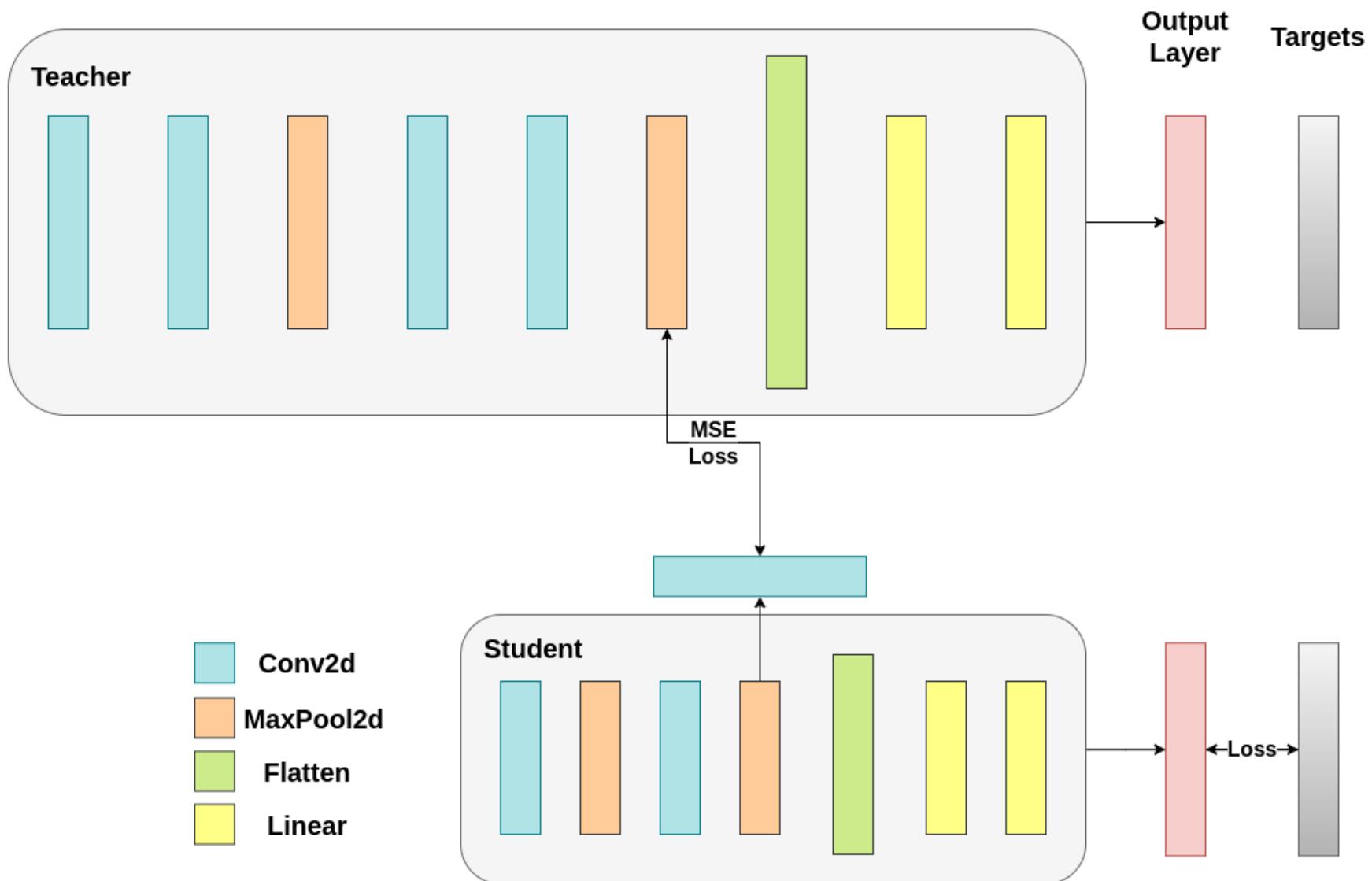
- ▶ **Дистилляция (или Knowledge Distillation)** - это процесс, при котором большая, мощная модель (teacher) используется для обучения меньшей модели (student). Меньшая модель пытается воспроизвести поведение большой, включая вероятностные распределения ответов.
- ▶ Дистилляция позволяет создавать меньшие и более быстрые модели с похожей производительностью, что снижает требования к вычислительным ресурсам и позволяет использовать модели в реальном времени, например, на мобильных устройствах.

Общий подход



Источник: https://pytorch.org/tutorials/beginner/knowledge_distillation_tutorial.html

Общий подход



Источник: https://pytorch.org/tutorials/beginner/knowledge_distillation_tutorial.html

Distillation loss

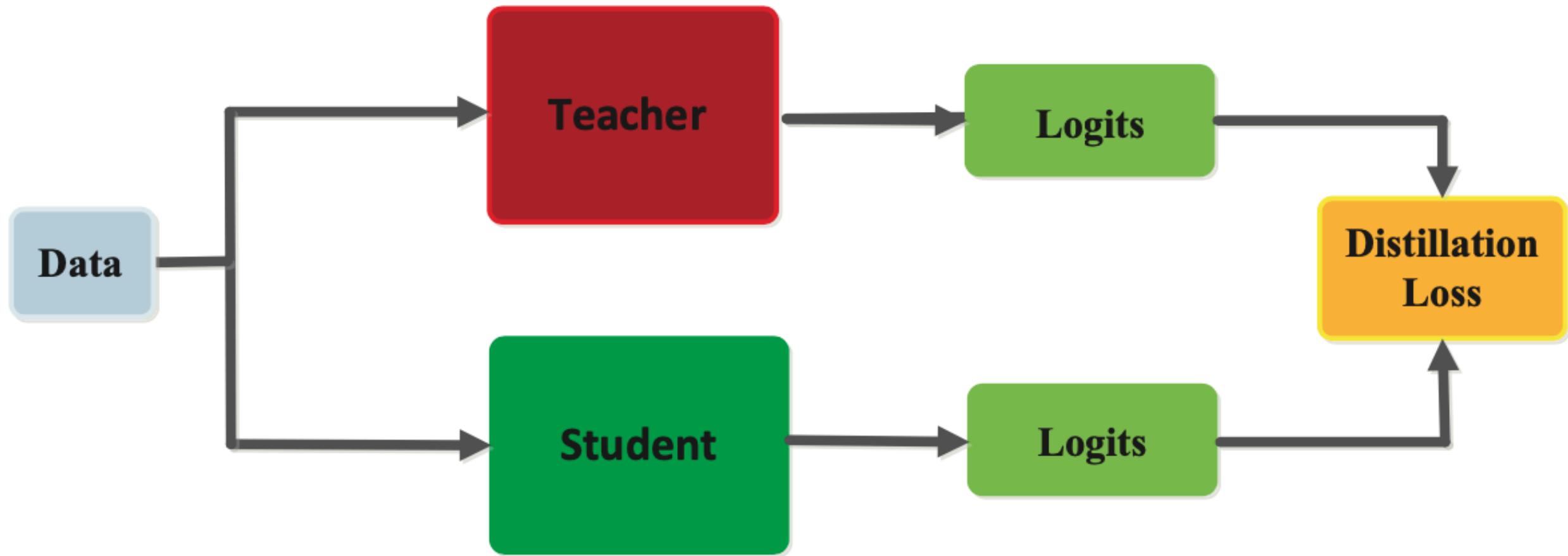
- ▶ Distillation loss – измеряет сходство между двумя выходами.
- ▶ Для softmax в случае классификации
 - считаем дивергенцию Кульбака-Лейблера (KL как в VAE)
- ▶ Для выхода регрессора или эмбеддингов
 - MSE
 - косинусное расстояние

Offline-distillation

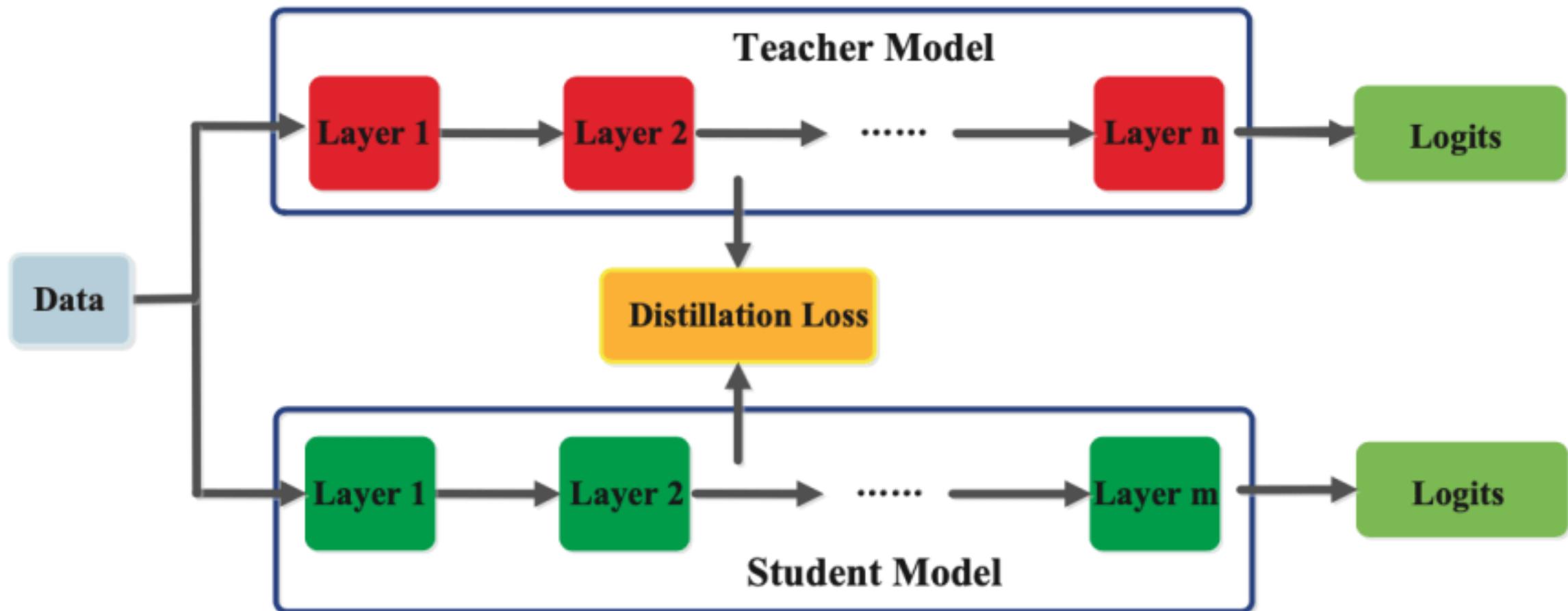
В этом подходе мы сначала обучаем сеть-учителя, а потом на основе нее обучаем сеть-ученика:

- ▶ Обучаем сеть-учителя.
- ▶ Устанавливаем соответствие. При проектировании сети-ученика необходимо установить соответствие между промежуточными выходами сети-ученика и сети-учителя.
- ▶ Прямой проход через сеть-учителя. Пропускаем данные через сеть учителя, чтобы получить все промежуточные результаты.
- ▶ Обратный проход через сеть-ученика. Используем выходные данные из сети-учителя и отношение соответствия для обратного распространения ошибки в ученической сети, чтобы она могла научиться воспроизводить поведение учительской сети.

Offline-distillation (response-based)



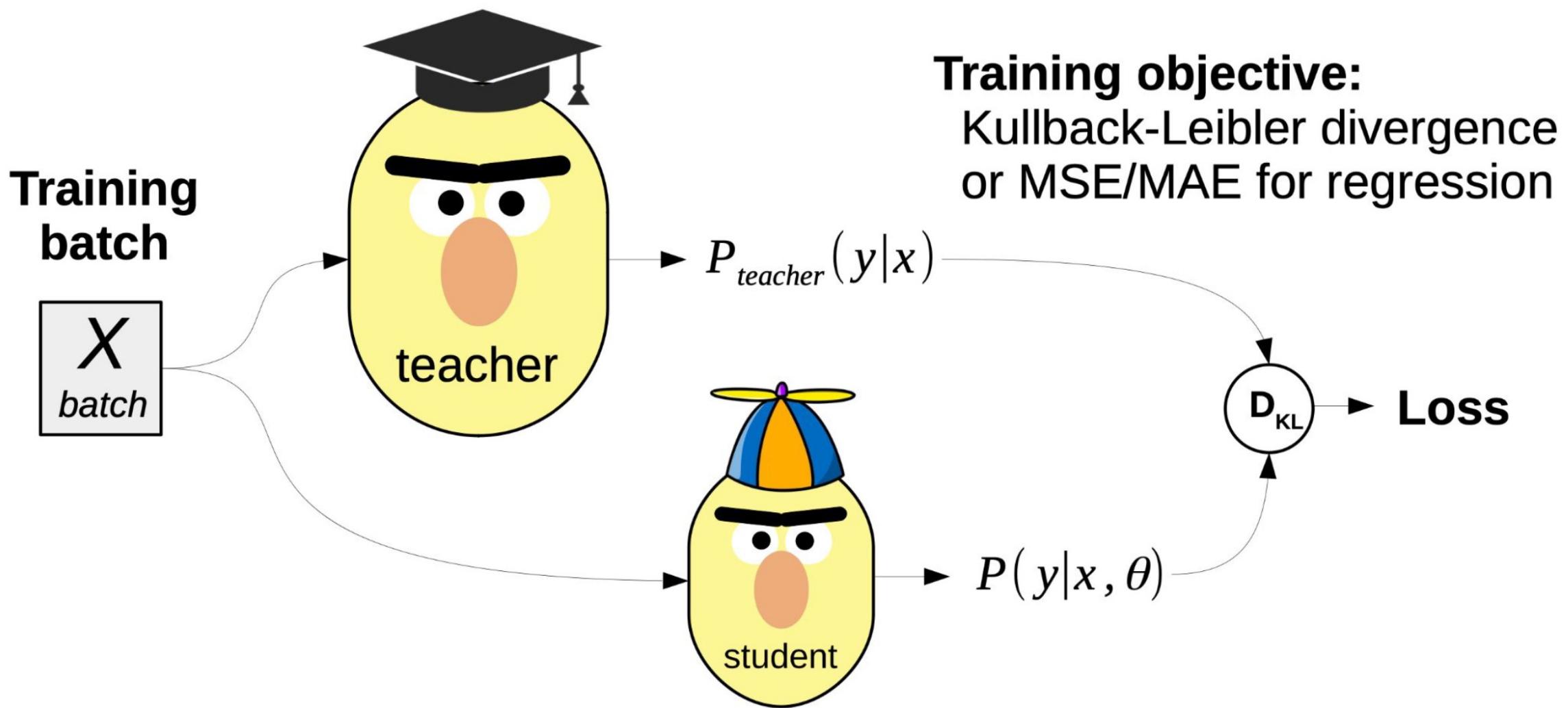
Offline-distillation (feature-based)



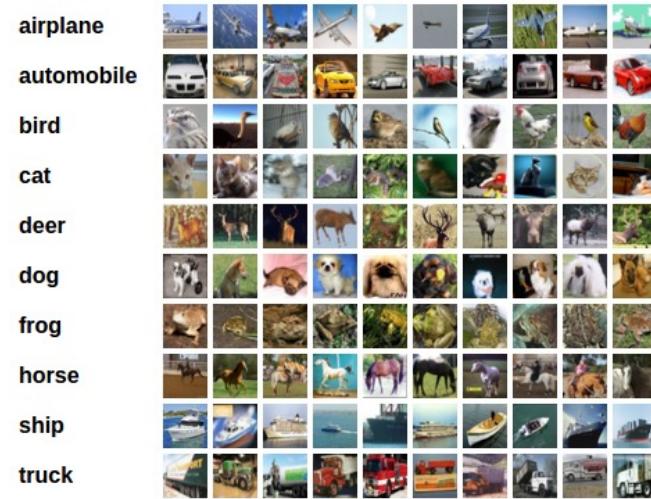
Online-distillation

В этом подходе мы обучаем сеть-учителя и сеть-ученика одновременно. Например, мы можем одновременно обучать две сети с целью получить одинаковое вероятностное распределение ответов. Этого можно достичь, используя в качестве функции потерь дивергенцию Кульбака-Лейблера между распределениями ответов сети-учителя и сети-ученика как distillation loss

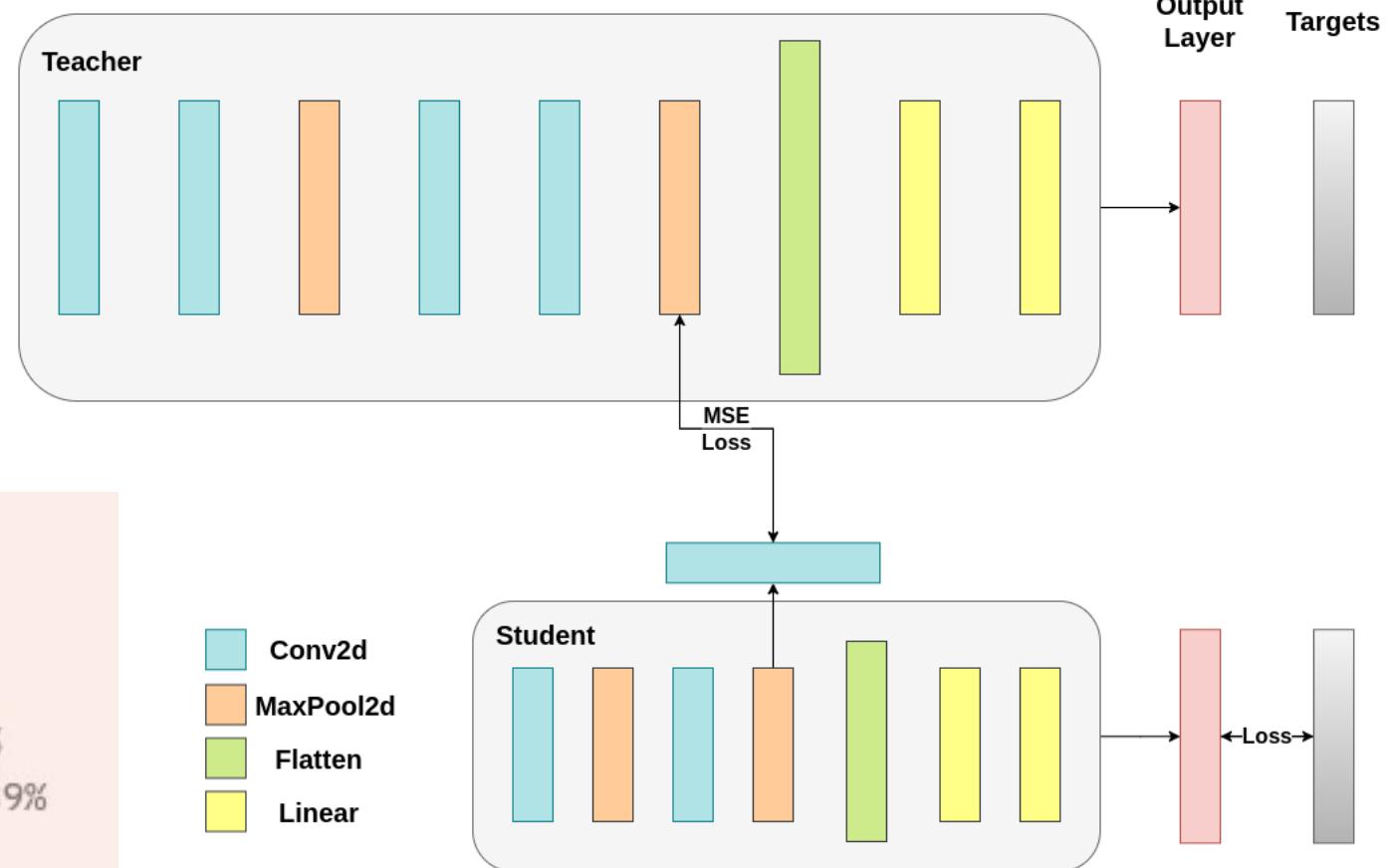
Online-distillation



Пример: дистилляция на CIFAR-10



Teacher accuracy: 75.49%
Student accuracy without teacher: 70.61%
Student accuracy with CE + KD: 70.48%
Student accuracy with CE + CosineLoss: 71.35%
Student accuracy with CE + RegressorMSE: 71.39%



Ссылка: https://pytorch.org/tutorials/beginner/knowledge_distillation_tutorial.html