Simplex LoRA: Expand important adapters

Давыденко Григорий, Шалыгин Игорь

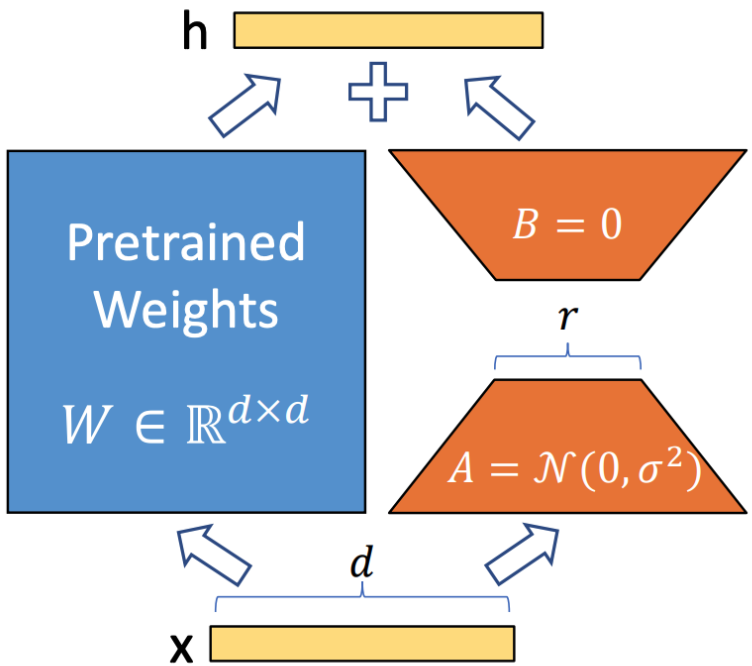
Научный руководитель: Безносиков А. Н.
Научный консультант: Веприков А. С.

Московский физико-технический институт

18 мая 2025 г.

Содержание доклада

Введение: Low-Rank Adaptation

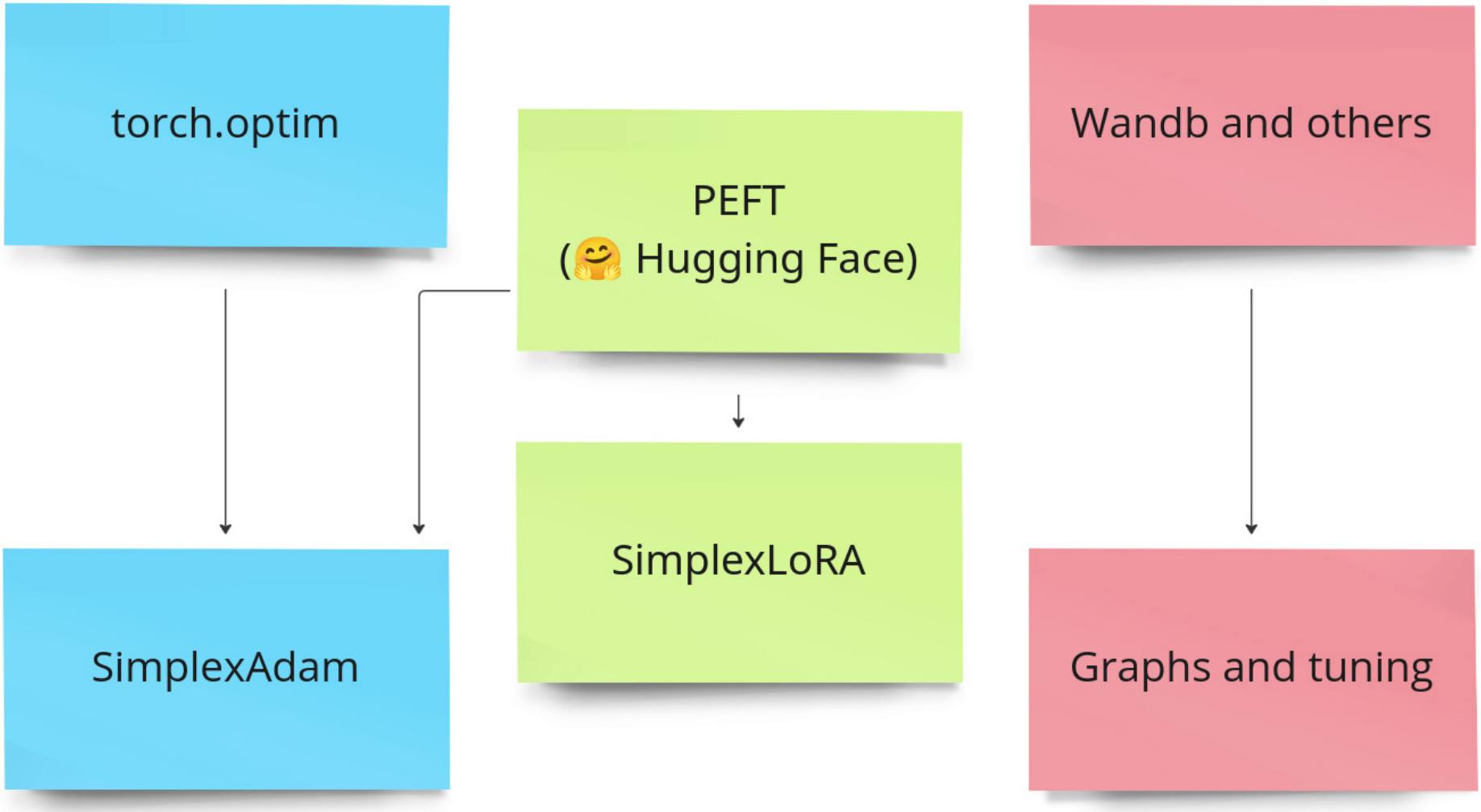


- Тюнинг линейных слоев LLM через низкоранговые матрицы $A \in \mathbb{R}^{d \times r}$ и $B \in \mathbb{R}^{r \times d}$
- $r \ll d$ (например, $r = 8$ для моделей с $d = 1024$)

Мотивация: как оценить важность слоя?

1. AdaLoRA, LoHA, DoRA, VERA и др.
2. Есть потребность в улучшении эффективности существующего алгоритма
3. Градиент слоя, его тип, зависимости, которые он отлавливает — показатели важности
4. Будем оценивать эту важность при помощи дополнительного параметра $\omega = (\omega_1, \ldots, \omega_n)^\top$
5. Новый вид адаптера: $W^i + A^i B^i \to W^i + \omega_i A^i B^i$, $\omega \in \Delta_{n-1}$
6. Тьюнер и оптимизатор для него

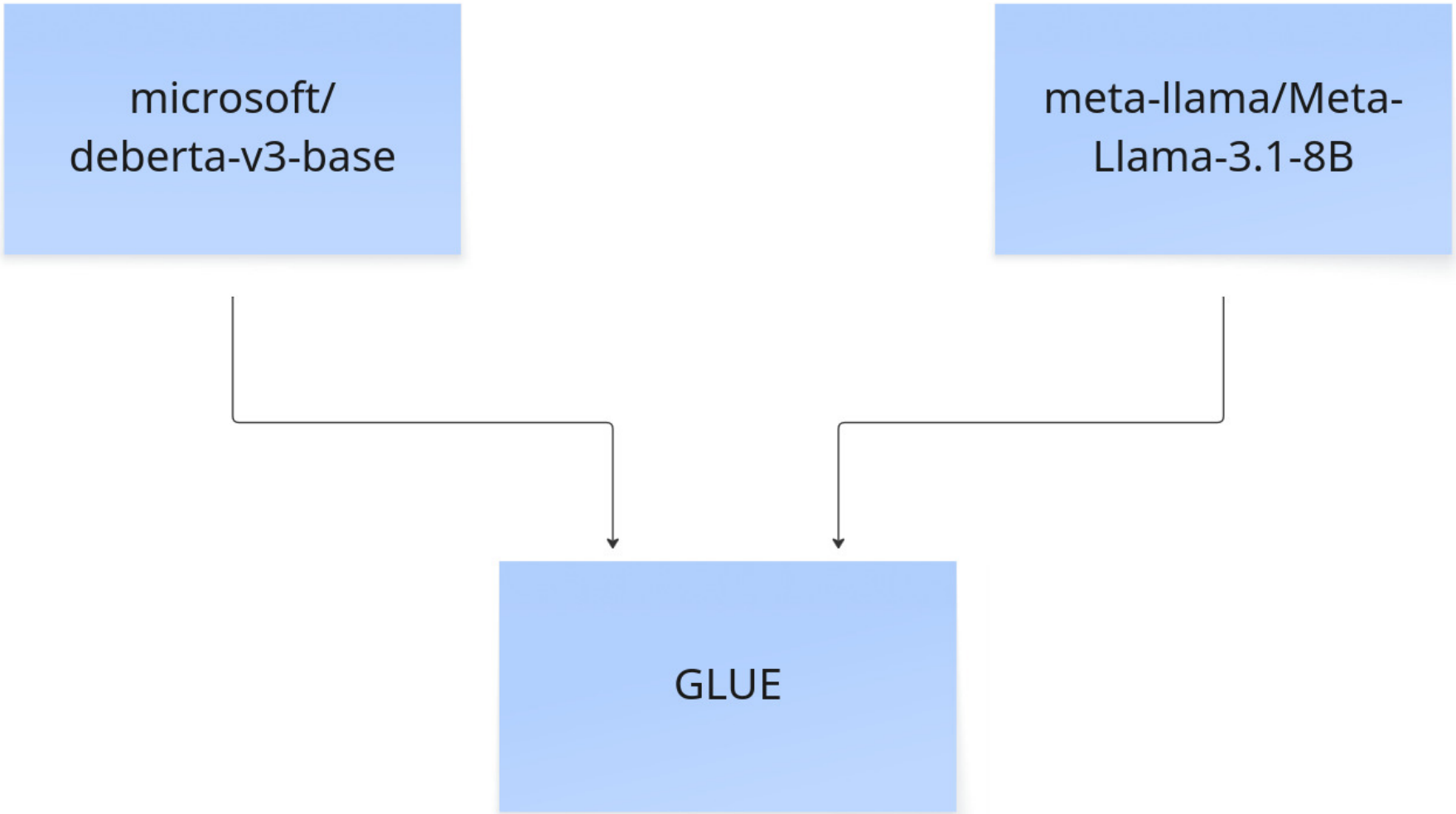
SimplexLoRA Framework



SimplexLoRA Framework

- **Softmax**
  - Параметр:
    - Температура
  - Особенности:
    - Гладкая
    - Дифференцируемая
    - Быстрая
  - Формула:
    - $x_i = \frac{\exp(\omega_i)}{\sum_{k=1}^{n} \exp(\omega_i)}$
- **Евклидова проекция**
  - Параметров нет
  - Особенности:
    - Итеративная
    - Точная
- **Взвешенный Softmax**
  - Параметр:
    - Температура
  - Особенности:
    - Гладкая
    - Дифференцируемая
    - Быстрая
    - Учитывает исходные веса
  - Формула:
    - $x_i = \frac{\exp(\omega_i) \cdot \omega_i}{\sum_{k=1}^{n} \exp(\omega_i) \cdot \omega_i}$

Модели, датасеты, эксперименты



Модели, датасеты, эксперименты

Method	# Params	MNLI Acc	SST-2 Acc	CoLA Mcc	QQP Acc/F1	QNLI Acc	RTE Acc	MRPC Acc/F1	ALL Avg
Full Fine-Tuning	184M (100%)	0.8910	0.9541	0.6806	0.8962/0.8644	**0.9383**	0.8376	0.8848/0.9165	**0.8742**
LoRA$_{r=8}$	442K (0.24%)	0.8797	0.9450	**0.6913**	0.8802/0.8437	0.9301	**0.8448**	0.8897/0.9220	**0.8693**
LoHA$_{r=8}$	884K (0.48%)	0.8560	0.9392	0.6295	0.8674/0.8269	0.9085	0.8051	0.8628/0.9007	0.8439
LoHA$_{r=32}$	**57.6K (0.03%)**	0.7587	0.9289	0.5907	0.8275/0.7816	0.8605	0.7509	0.7500/0.8416	0.7861
AdaLoRA$_{r=8}$	664K (0.36%)	0.8390	0.9392	0.6222	0.8534/0.8083	0.8999	0.7834	0.7059/0.8225	0.8099
VERA$_{r=1024}$	1.64M (0.88%)	0.8355	0.9404	0.6252	0.8592/0.8203	0.8975	0.7581	0.8529/0.8962	0.8351
rsLoRA$_{r=8}$	442K (0.24%)	0.8814	0.9564	0.6414	0.8853/0.8506	0.9325	0.8267	0.8750/0.9122	0.8620
DoRA$_{r=8}$	442K (0.24%)	0.8811	0.9473	0.6748	0.8809/0.8451	0.9295	0.7978	0.8799/0.9139	0.8656
SimplexLoRA$_{r=8}$	442K (0.24%)	**0.8984**	**0.9587**	0.6621	**0.9126/0.8846**	0.9381	0.8099	**0.8987/0.9289**	0.8683

Таблица: microsoft/deberta-v3-base on GLUE

- NVIDIA GeForce RTX 2080 Ti

Научный и технический продукт

Научный продукт

- Эффективный метод выбора наиболее важных лор
- Балансировка между эффективностью алгоритма и затратами по памяти и FLOP

Технический продукт

- Реализация SimplexLoRA в библиотеке PEFT
- Оптимизатор SimplexAdam в библиотеке PEFT

Приложение: related work

- AdaLoRA: Adaptive Budget Allocation for Parameter-Efficient Fine-Tuning
- DoRA: Weight-Decomposed Low-Rank Adaptation
- Large-scale Multiclass Support Vector Machine Training via Euclidean Projection onto the Simplex
- LoRA: Low-Rank Adaptation of Large Language Models
- VeRA: Vector-based Random Matrix Adaptation