УДК 004.852

# СРАВНЕНИЕ МЕТОДОВ СЖАТИЯ ТЕНЗОРОВ МОМЕНТОВ ОПТИМИЗАТОРА ПРИ РАСПРЕДЕЛЕННОМ ОБУЧЕНИИ БОЛЬШИХ ЯЗЫКОВЫХ МОДЕЛЕЙ

**Маслов И. А.[1] , Поляков И. В.[1]**
**Научный руководитель – доктор техн. наук, доцент Духанов А. В.[1]**
[1]Университет ИТМО
iamaslov@itmo.ru

## Введение

Один из классических подходов к распределенному обучению – параллелизм модели – подразумевает хранение тензоров моментов оптимизатора на разных хостах. При таком подходе, передача тензоров для синхронизации становится узким местом. В данной работе рассматривается ряд подходов к сжатию тензоров, сравнивается их скорость и вызываемые ими погрешности.

## Основная часть

Спарсификация – наиболее алгоритмически простой подход к сжатию, основанный на эвристике разреженности градиента. В процессе спарсификации, доля частных производных обнуляется, таким образом потери касаются в основном статистического шума, а содержательные веса сохраняются без потерь[1]. Как правило, используется динамическая спарсификация, при которой доля обнуляемых весов постепенно уменьшается на протяжении трейна.

Один из наиболее популярных современных подходов к сжатию – гомоморфное сжатие[2]. Алгоритм сопоставляет тензору тензор меньшей размерности, причем это сопоставление является гомоморфизмом векторных пространств, что позволяет не разжимать тензоры на этапе синхронизации. Также, существует алгоритм, комбинирующий гомоморфное сжатие и спарсификацию, позволяющий значительно снизить потери при сжатии.

Тензорная факторизация – другое активно исследуемое направление в тензорном сжатии[3]. Основной алгоритма тензорного разложения является сингулярное разложение матриц, получаемых изменением формы тензора; дополнительное сжатие достигается за счет отбрасывания части сингулярных чисел[4].

Все описанные выше подходы имеют сильные и слабые стороны, выбор зависит непосредственно от АО и архитектуры модели: тензорная факторизация дает наименьшие потери при высоком сжатии среди рассматриваемых алгоритмов, однако вычислительно сложно, в связи с чем на не специализированном оборудовании может даже усугубить узкое место. Спарсификация, за счет высокой скорости, хорошо подходит в качестве вспомогательного алгоритма сжатия, однако при повышении степени сжатия вызывает резкое увеличение погрешности, в связи с чем сама по себе не способна значительно ускорить обучение. Гомоморфное сжатие дает среднюю среди рассматриваемых алгоритмов погрешность и имеет среднюю сложность, однако, при использовании небольшого числа хостов, его эффективность мала.

## Выводы

В ходе исследования были рассмотрены классические алгоритмы сжатия (равномерное квантование, спарсификация), а также фронтирные методы, в числе которых гомоморфное сжатие и тензорная факторизация, также были рассмотрены комбинированные подходы. Для каждого подхода были выявлены особенности и условия, в которых их применение предпочтительно, были проведены эксперименты.

**Литература**


1. Lin J., Tang J., AWQ: Activation-aware Weight Quantization for On-Device LLM Compression and Acceleration // Proceedings of Machine Learning and Systems 6, (MLsys) 2024. – Режим доступа: https://proceedings.mlsys.org/paper_files/paper/2024/hash/42a452cbafa9dd64e9ba4aa95cc1ef21-Abstract-Conference.html
2. Haoyu Li, Yuchen Xu, Accelerating Distributed Deep Learning using Lossless Homomorphic Compression // Cornell University, 2024. – Режим доступа: https://dl.acm.org/doi/abs/10.1145/3721146.3721946 (дата обращения: 20.01.2026).
3. Kolda T., Bader B., Tensor Decompositions and Applications // Society for Industrial and Applied Mathematics, 2009. Vol. 51, No. 3, pp. 455–500
4. Bigoni D., Engsig-Karup A., Spectral Tensor-Train Decomposition // SIAM, 2016. – Режим доступа: https://arxiv.org/pdf/1405.5713 (дата обращения: 20.01.2026).


Маслов И. А. (автор) Подпись

Духанов А. В, (научный руководитель) Подпись