

**Правительство Российской Федерации
Федеральное государственное автономное
образовательное учреждение высшего образования
«Национальный исследовательский университет «Высшая школа экономики»**

Факультет компьютерных наук
Образовательная программа бакалавриата 01.03.02 «Прикладная математика и информатика»

**ОТЧЕТ
по учебной практике**

на факультете компьютерных наук НИУ ВШЭ
(название организации, предприятия)

Выполнила студентка
группы БПМИ215

А. Е. Биран

(инициалы, фамилия)



(подпись)

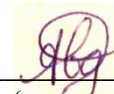
Руководитель практики

_____ Департамент больших данных и информационного поиска, доцент _____
(подразделение ФКН, должность)

_____ Авдеев Роман Сергеевич _____
(ФИО руководителя практики)

Дата 05.09.2022

_____ 10 _____
(оценка)



(подпись)

Аннотация

Практика была пройдена в департаменте больших данных и информационного поиска ФКН НИУ ВШЭ. В процессе данной практики была изучена статья Ивана Оселедца (I. Oseledets) "Tensor-train decomposition", в том числе разобраны принципы работы алгоритмов, использующихся для работы с ТТ-форматом тензоров, а основные из них реализованы на языке Python. В качестве численных экспериментов было проведено исследование эффективности ТТ разложения и сравнение результатов работы алгоритмов, представленных в библиотеках языка Python, и алгоритмов, реализованных самостоятельно.

Оглавление

Аннотация.....	2
Цель практики	2
Задачи практики	2
Изученные материалы	2
Методы решения задач	3
Полученные результаты.....	4
Заключение	7
Список использованных источников	8
Рабочий план-график прохождения практики с отметками о выполнении	8
Приложения	8

Цель практики

Изучение методов работы с тензорами в ТТ-формате. Исследование эффективности различных реализаций ТТ-разложения в зависимости от параметров с помощью численных экспериментов.

Задачи практики

1. Освоение базовых понятий теории тензоров
2. Реализация алгоритмов ТТ-разложения и ТТ-округления с помощью языка Python
3. Овладение навыками работы с библиотеками `tensorly` и `t3f` языка Python, использующихся для работы с тензорами в ТТ-формате

Изученные материалы

Основным источником материала является статья [1], где и были изложены основные идеи ТТ-разложения, а также описаны принципы работы алгоритмов с тензорами в ТТ-формате. Для получения информации в более сжатом виде была проанализирована презентация [2].

Для реализации алгоритмов были использованы библиотеки `tensorly` и `t3f`, в которых представлены основные инструменты для работы с тензорами в ТТ-формате. В частности, была изучена документация данных библиотек [3] и [4].

Методы решения задач

Основной задачей ТТ-разложения является решение проблемы проклятия размерности, т.е. экспоненциального возрастания количества данных с увеличением размерности. Эта задача решается путем представления каждого элемента тензора в виде произведения матриц, количество которых совпадает с размерностью тензора(d). Набор данных матриц, соответствующих определенному индексу, образует трехмерный тензор, называемый ядром ТТ-разложения.

Нахождение ядер(алгоритм ТТ-SVD) сводится к последовательному вычислению SVD разложения d вспомогательных матриц, начиная с матрицы развертки начального тензора. Так как низкоранговые матрицы встречаются достаточно редко, данный алгоритм может быть адаптирован для приближенного случая, т.е. вычисления наилучшего разложения с некоторой заданной относительной погрешностью. Это достигается путем отбрасывания всех сингулярных значений, меньших, чем δ (параметр, зависящий от требуемой погрешности и размерности тензора). Таким образом, нахождение рангов разложение равносильно нахождению δ -рангов матриц развертки, в отличие от канонического разложения, где данная задача является NP-трудной.

Еще одним важным алгоритмом является алгоритм ТТ-округления, который позволяет найти оптимальное (т.е. с меньшими рангами) представление тензора в ТТ-формате с некоторой заданной относительной погрешностью, если дано ТТ-разложение тензора. Этот алгоритм основан на вычислении SVD разложения с помощью QR разложения. А именно, структурированного вычисления QR разложения для матрицы, представленной в ТТ-формате, путем развертки ядер справа налево и ортогонализации строк. После этого процедура снова сводится к последовательному вычислению усеченного SVD (δ -truncated SVD). Алгоритм ТТ-округления эффективен в случае осуществления арифметических операций в ТТ-формате, ведь многие базовые операции с ТТ-тензорами приводят к тензору также в ТТ-формате, но с неоптимальными рангами.

Преимуществом ТТ-формата является то, что с помощью него можно не только хранить тензор в сжатом виде, но и осуществлять различные базовые арифметические операции намного эффективнее. Так, например, сложение, умножение на скаляр, многомерная свертка, покомпонентное и скалярное произведение, норма Фробениуса сводятся к простейшим операциям с ядрами тензоров.

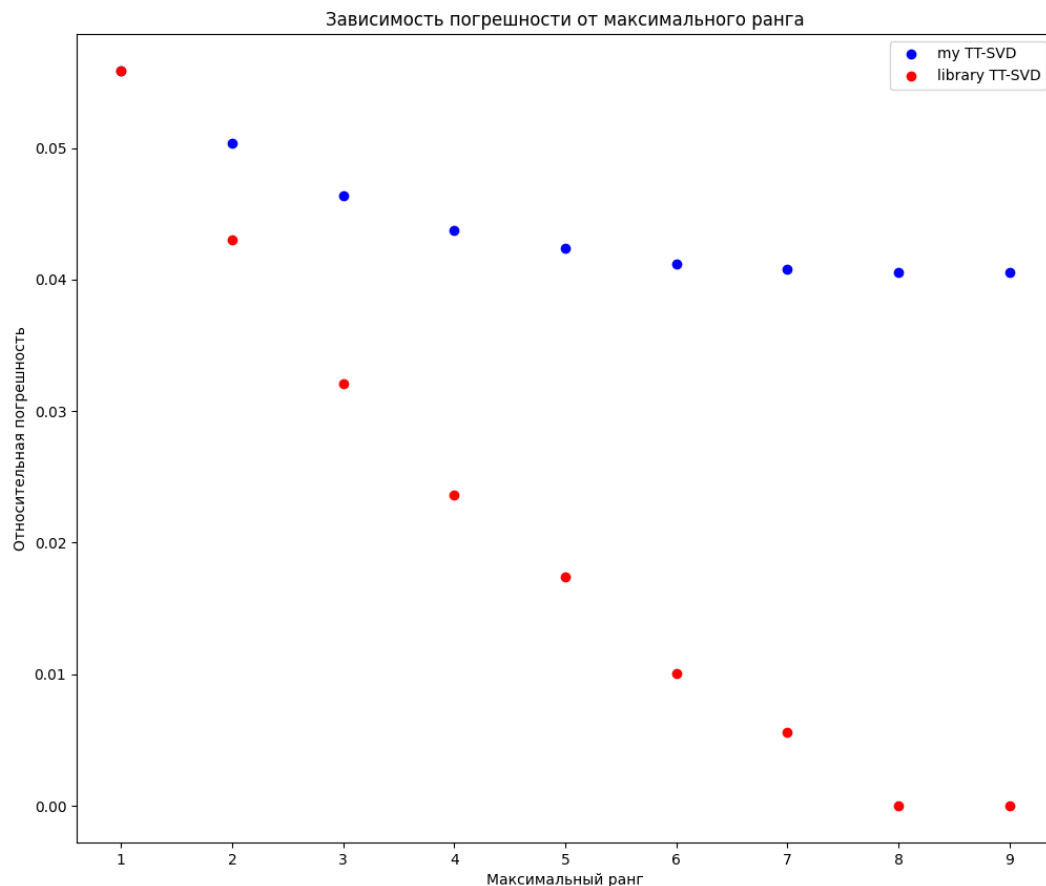
Для реализации алгоритмов ТТ-SVD и ТТ-rounding применялись стандартные функции библиотеки `numpy`, в частности для получения SVD и QR разложений. Результаты работы сравнивались с результатами работы аналогичных функций библиотеки `t3f`(`to_tt_tensor` и `round`). В случае ТТ-SVD численные эксперименты проводились на случайных тензорах, в то время как для ТТ-rounding для получения тензора неоптимального ранга сначала создавались два случайных тензора одинаковых размерностей, а затем находилась их сумма. Визуализация в виде графиков проводилась с помощью библиотеки `matplotlib` языка Python.

Полученные результаты

Главным результатом практики является реализация двух основных алгоритмов (TT-SVD и TT-round) для работы с тензорами в TT-формате, а также сравнение их эффективности с библиотечными функциями.

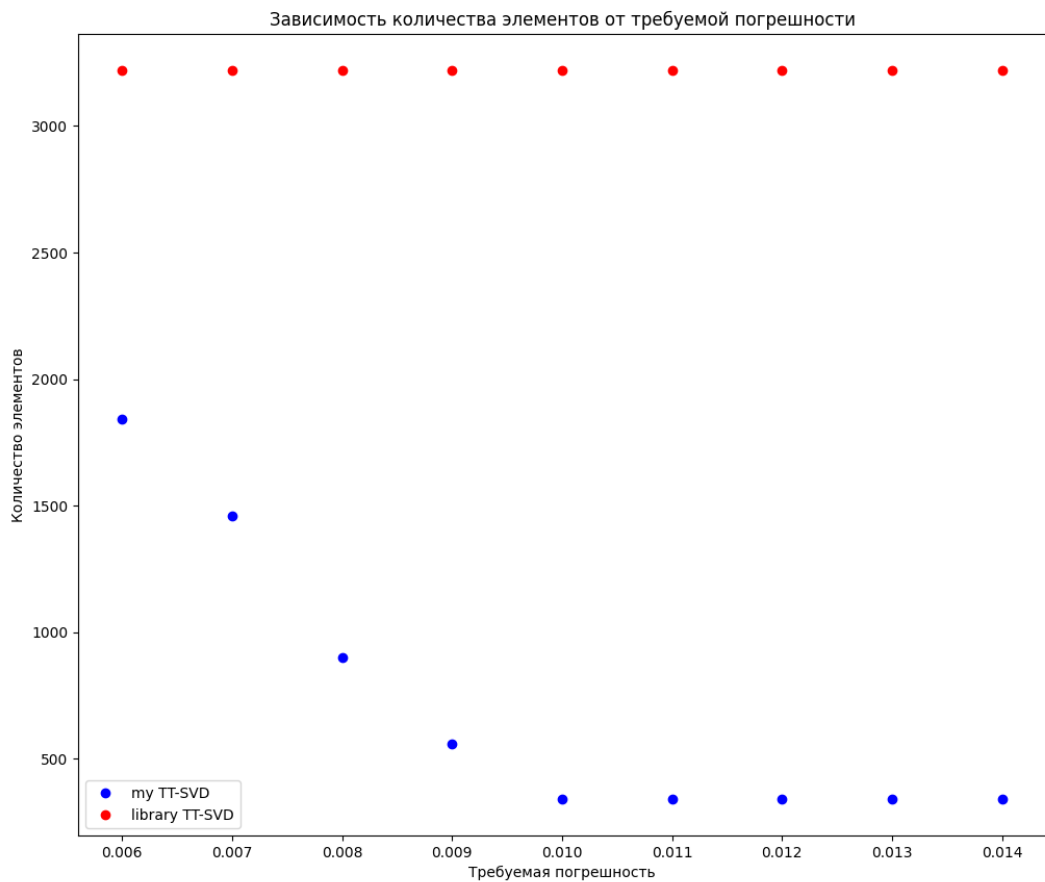
Первые два графика показывают сравнение результатов работы самостоятельно реализованного алгоритма TT-SVD и аналогичной библиотечной функции `t3f.to_tt_tensor`.

На этом графике отображена зависимость точности сжатия двух алгоритмов TT-SVD от максимального ранга сжатого тензора.



Судя по данному графику можно сказать, что при фиксации остальных параметров и изменении максимального ранга, у самостоятельно реализованного алгоритма точность хуже, но все еще остается в рамках нескольких процентов.

Второй график, относящийся к сравнению двух реализаций алгоритма TT-SVD, иллюстрирует зависимость размера сжатого тензора от заданной точности.

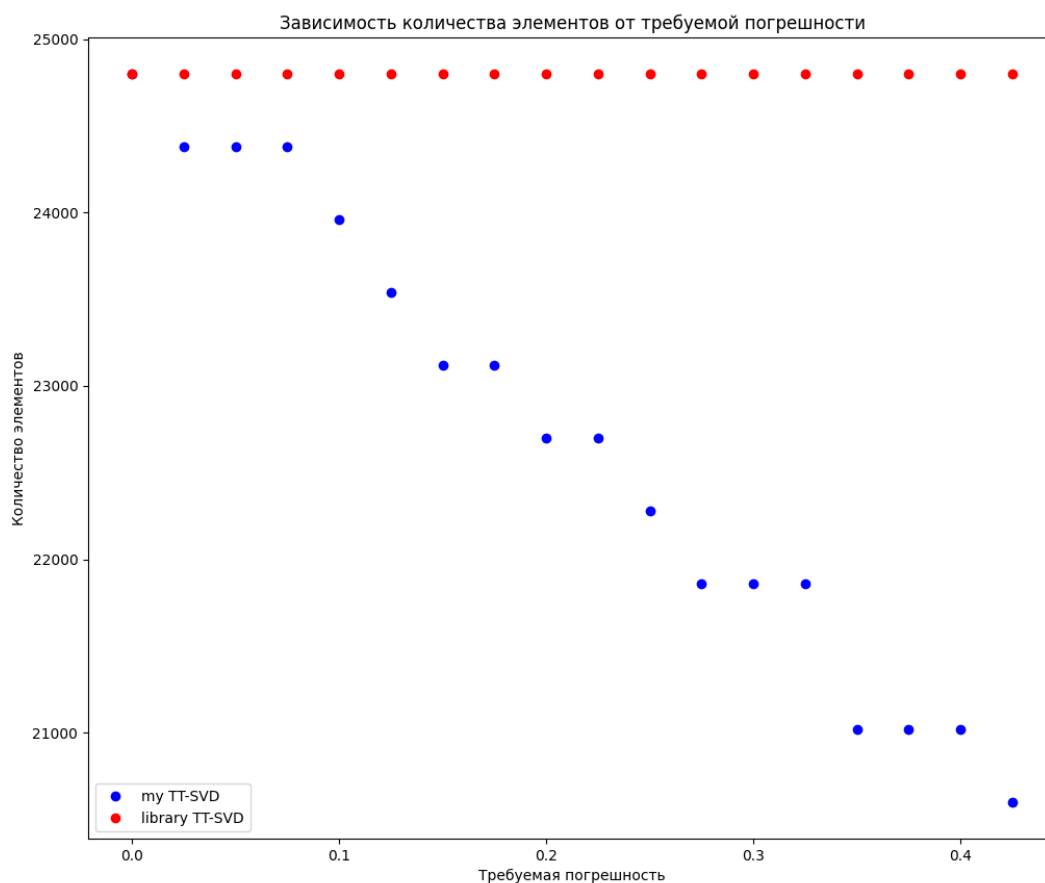


Здесь видно, что в этом случае результаты полностью противоположные. В то время как размер тензора в сжатом виде у библиотечного алгоритма остается константным, размер сжатого самостоятельно реализованным алгоритмом тензора значительно убывает, причем самостоятельно реализованный алгоритм вполне эффективен (сжимает тензор лучше в 6 раз).

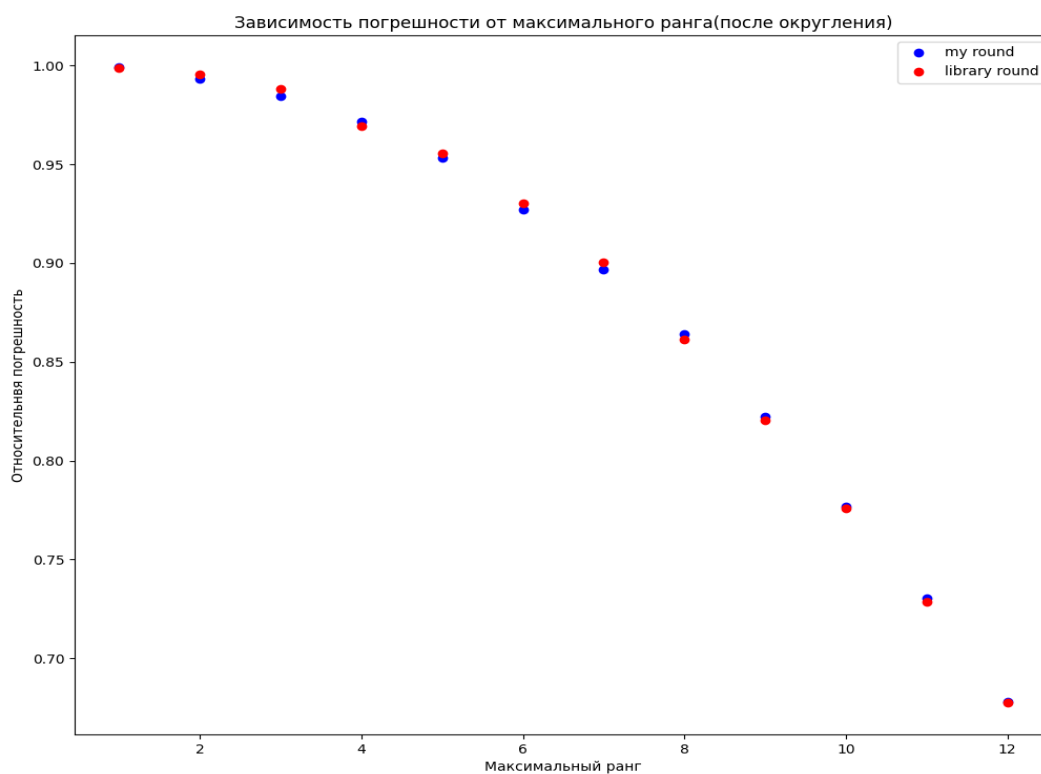
Таким образом, выбор алгоритма TT-SVD зависит от того, что представляет большую важность: точность, либо память, которую занимает сжатый тензор.

Также было проведено сравнение эффективности самостоятельно реализованного алгоритма TT-rounding и аналогичной библиотечной функции `t3f.round`, что проиллюстрировано на следующих трех графиках.

Первый из них изображает зависимость размера тензора после округления от требуемой погрешности.

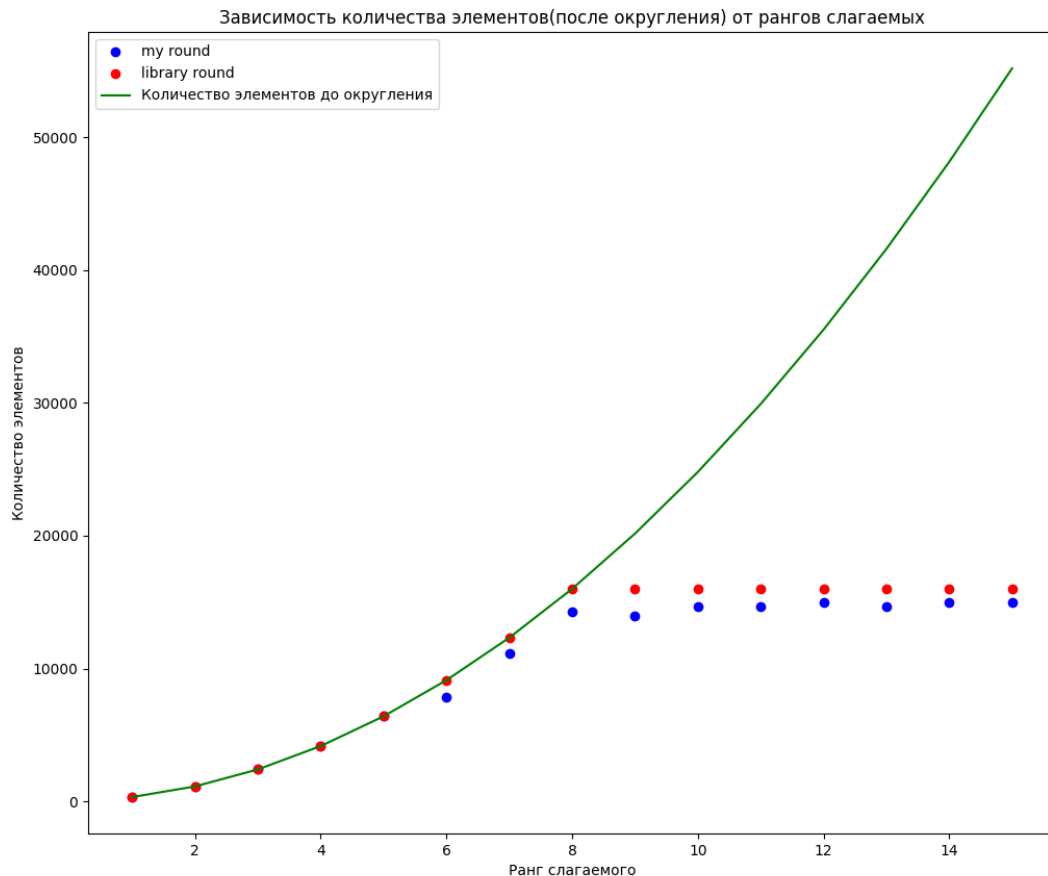


Можно заметить, что этот график очень похож на аналогичный для алгоритма TT-SVD. Пусть и не так эффективно, как в случае TT-SVD, самостоятельно реализованный алгоритм округления уменьшает количество элементов при увеличении желаемой погрешности, в то время как библиотечная функция не дает никакого сжатия. Следующий график показывает зависимость погрешности округления от максимального ранга.



Судя по данному графику, оба алгоритма одинаково эффективны и при увеличении максимального ранга погрешность стремительно падает.

Последний график иллюстрирует зависимость количества элементов после округления от начальных рангов слагаемых.



Оба алгоритма также показывают примерно одинаковые результаты. На данном графике отображено, насколько в принципе эффективен алгоритм округления (при некоторых данных тензор сжимается в 4 и более раза).

Итак, самостоятельно реализованный алгоритм TT-rounding является более эффективным, чем библиотечная функция `t3f.round`, так как намного значительно сжимает тензор в зависимости от точности, в то время как в остальных экспериментах алгоритмы имели одинаковые результаты.

Заключение



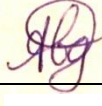

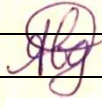

Таким образом, в рамках практики был проведен анализ статьи [1], в ходе которого были получены базовые навыки работы с тензорами и изучены основные алгоритмы для работы с тензорами в TT-формате. Более того, эти алгоритмы были успешно реализованы на языке Python для сравнения с аналогичными алгоритмами библиотек `tensorly` и `t3f`. Результаты сравнения показали, что по некоторым параметрам самостоятельно

реализованный алгоритм эффективнее, чем библиотечный, однако данные результаты я не считаю достаточными для использования в ВКР.

Список использованных источников

- [1] I. V. Oseledets: Tensor-train decomposition. SIAM, 2011
https://sites.pitt.edu/~sjh95/related_papers/tensor_train_decomposition.pdf
- [2] Родоманов А.О: ТТ-разложение для компактного представления тензоров. 2013
<http://www.machinelearning.ru/wiki/images/f/fe/TT-decomposition.pdf>
- [3] Tensorly documentation
<http://tensorly.org/stable/modules/api.html#>
- [4] t3f documentation
<https://t3f.readthedocs.io/en/latest/api.html#>

Рабочий план-график прохождения практики с отметками о выполнении

№ п/п	Сроки проведения	Планируемые работы	Отметка руководителя о выполнении (подпись)
1	01.07.2022	1. Инструктаж по ознакомлению с требованиями охраны труда, техники безопасности, пожарной безопасности, а также правилами внутреннего трудового распорядка	
2	02-04.07.2022	2. Изучение основных понятий теории тензоров	
3	05-08.07.2022	3. Анализ статьи Ивана Оселедта (I. Oseledets) "Tensor-train decomposition"	
4	09-13.07.2022	4. Реализация базовых алгоритмов для работы с тензорами в ТТ-формате 5. Проведение численных экспериментов	
5	14.07.2022	6. Подготовка отчета о прохождении практики	
6	15.07.2022	7. Обсуждение результатов с руководителем практики	

Приложения

- 1. Графики
https://drive.google.com/drive/folders/1rK_BDBNOJQSdQth2fQVkJNrt5z5kkSneJ?usp=sharing
- 2. Репозиторий с кодом
<https://github.com/nastya-biran/practice-TT-decompostion/tree/main>