FRIEDRICH-SCHILLER-
UNIVERSITÄT
JENA

A Comparison of Different Algorithms for Einsum

BACHELOR THESIS

to be Awarded the Academic Degree

of

Bachelor of Science (B.Sc.)

in Informatics

FRIEDRICH-SCHILLER-UNIVERSITY JENA

Faculty for Mathematics and Informatics

Submitted by Leon Manthey

born on 31.10.1999 in Berlin

Supervisor: Mark Blacher

Jena, 15.05.2024

Abstract

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Fusce eleifend orci et venenatis cursus. Nullam eget ornare lacus. Donec non dolor non tellus eleifend vehicula. Sed et lorem lectus. Vestibulum sagittis sed nisi ac interdum. Duis nec accumsan velit, hendrerit malesuada magna. Aliquam erat volutpat. Cras eu ante nec est malesuada volutpat. Proin quis posuere quam. Etiam aliquam eros quis dui sagittis, a fermentum lectus rutrum. Nunc tempor mauris vel tellus facilisis rhoncus. Aliquam ut leo eget metus volutpat vestibulum. Nam non consequat ante. In rutrum felis in enim fringilla lacinia. Phasellus ut imperdiet risus. Curabitur tincidunt libero sed urna dignissim, eget rutrum felis scelerisque. Nunc ut convallis neque, non tincidunt nulla. Curabitur quis condimentum leo. Phasellus laoreet ligula vel mi commodo, id accumsan diam tristique. Maecenas euismod lorem in tempor iaculis. Orci varius natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Ut eget purus sem. Suspendisse venenatis aliquet dignissim. Integer turpis lorem, tempus non turpis et, gravida aliquet erat. Sed vel neque non ex ultrices vestibulum. Aliquam purus quam, rhoncus non ante at, convallis sagittis erat. Sed justo elit, vulputate vel accumsan non, porta eget turpis. Proin eget ultrices sem. Nunc eu velit.

Contents

1 Introduction

Einstein summation notation is a powerful and compact notation used in mathematics and physics to represent sums over tensor indices. It was introduced by Albert Einstein in the early 20th century as a means to simplify expressions in the theory of relativity. [1] The notation is both elegant and efficient, making it an essential tool in various fields such as theoretical physics, computational mathematics, and data science.

The fundamental operation in Einstein summation notation, often referred to simply as "einsum," is the Einstein summation. This operation allows for the concise expression of various tensor operations, including element-wise multiplication, dot products, outer products, and matrix multiplications. The computational efficiency and expressiveness of einsum have led to its adoption in numerous applications, ranging from machine learning to scientific computing.

In many practical applications, especially in machine learning and scientific computing, the data involved is often sparse. Sparse matrices are matrices in which most of the elements are zero. Handling sparse matrices efficiently requires specialized algorithms and data structures to avoid unnecessary computations and to save memory. Traditional libraries like NumPy [2] and other machine learning frameworks typically support Einstein summation (einsum) for dense matrices, but not for sparse matrices. The only known library that aims to support einsum operations on sparse tensors is Sparse [3]. However, Sparse only allows the same symbols as indices for the einsum notation that NumPy supports which limits the number of dimensions a tensor can have by only allowing for the latin alphabet in capital and small letters. Furthermore, real Einstein summation problems often inculde expressions with hundreds or even thousand of higher order tensors. In order to express those kind of operations we require a large set of symbols. This is why our approaches is capable of handling all symbols in the UTF-8 encoding.

This thesis explores the implementation and performance of Einstein summation across different computing paradigms, with a particular focus on sparse tensors. Specifically, it focuses on comparing two distinct implementations to multiple libraries:

- SQL-based Implementation: This implementation constructs SQL queries dynamically using Python and executes them via SQLite. While SQL is traditionally

used for database operations, this approach demonstrates the versatility of SQL in performing tensor operations.

- C++ Implementation: The final implementation is written in C++, with multiple versions ranging from naive to optimized approaches. The C++ implementations aim to explore the performance trade-offs between simplicity and optimization, offering insights into how different coding strategies affect computational efficiency.

By comparing these implementations, this thesis aims to provide a comprehensive analysis of the performance and scalability of Einstein summation in various computing environments. The SQL-based implementation serves as a baseline, showcasing the potential of database query languages for tensor operations. The Sparse library implementation highlights the advantages of using specialized libraries for sparse data structures. Finally, the C++ implementations demonstrate the impact of low-level optimizations on computational performance.
Through this comparative study, we seek to identify the strengths and weaknesses of each approach, providing valuable guidelines for selecting the appropriate method based on specific use cases and computational requirements. This work contributes to the broader understanding of tensor operations and their efficient implementation, offering practical insights for researchers and practitioners in fields that rely heavily on tensor computations.

Bibliography

[1] Albert Einstein. “Die Grundlage der allgemeinen Relativitätstheorie”. In: *Annalen der Physik*. Vierte Folge Band 49 (1916). pp. 781-782, pp. 769–822.

[2] Charles R. Harris et al. “Array programming with NumPy”. In: *Nature* 585.7825 (2020), pp. 357–362.

[3] Hameer Rocklin Matthew Abbasi. *Sparse 0.15.4*. `https://github.com/pydata/sparse`. 2024.