Proceedings of the 2023 International Conference on Machine Learning and Automation
DOI: 10.54254/2755-2721/31/20230154

Research on fast matrix multiplication algorithm

Qize Zhang

School of Computer Science and Engineering, Sun Yat-sen University, Guangzhou, 510006, China

nateyzbzzz@gmail.com

Abstract. This research mainly focuses on fast matrix multiplication algorithms. Fast matrix multiplication is one of the most fundamental problems in computer science. The fast matrix multiplication algorithm differs from conventional matrix multiplication in that it offers a faster computational approach that can perform the operation in less than O(n3) time complexity. This algorithm provides a more efficient method for multiplying matrices, significantly reducing the computational requirements. The Laser method, developed by Coppersmith and Winograd, is an algorithm for matrix multiplication that does not involve direct computation. It establishes a relationship between matrix multiplication and tensors and simplifies the operation by finding an intermediate tensor that is computationally manageable. This method applies a series of simplification operations to determine an upper bound on the computational complexity of matrix multiplication. However, as matrices become larger, the computational and memory requirements increase, posing challenges for practical implementation. This research will present the main ideas and performance of the Laser method and discuss the improvements made to the Laser method, including refined analysis and asymmetric hashing techniques. Additionally, it highlights the need for further exploration, such as parallel computing and optimization strategies, to enhance the efficiency of matrix multiplication algorithms. Furthermore, this research will also provide a prospectus for the future of matrix multiplication algorithms, such as the practical implementation of the Laser method.

Keywords: fast matrix multiplication algorithm, time complexity, laser method.

1. Introduction

Fast matrix multiplication algorithms have been the subject of extensive research in computer science, aiming to improve the computational efficiency of this fundamental operation. Traditional matrix multiplication algorithms, with a time complexity of O(n3), have limitations when dealing with large matrices. Over the years, several breakthroughs have been made, leading to the development of faster algorithms. This research focuses on exploring the Laser method, which offers an alternative approach to matrix multiplication without direct computation.

The Laser method, introduced by Coppersmith and Winograd, revolutionized matrix multiplication by establishing a connection between matrix multiplication and tensors. By finding a computationally manageable intermediate tensor, the Laser method simplifies the operation and provides an upper bound on the computational complexity. It has proven to be a significant improvement over conventional algorithms, but challenges remain in practical implementations, especially as matrix sizes increase.

DOI: 10.54254/2755-2721/31/20230154

This research aims to delve into the Laser method, its improvements, and the challenges of practical implementations. It explores the significance of efficient matrix multiplication algorithms in various domains, including scientific computing, machine learning, and optimization problems. By presenting a comprehensive overview of the advancements in this field, this research provides insights into the practical implementation of the Laser method and offers a prospectus for the future development of matrix multiplication algorithms, including parallel computing and optimization strategies.

2. Background

Significant advancements have been made in the field of fast matrix multiplication algorithms. Strassen's breakthrough in 1969 challenged the conventional O(n3) complexity by demonstrating matrix multiplication in O(n2.81) operations [1]. Pan further improved the time complexity to O(n2.796) with the APA algorithm [2]. Schönage introduced the Laser method in 1981, reducing the time complexity to O(n2.522) [3]. Coppersmith and Winograd's groundbreaking work in 1990 introduced the Coppersmith-Winograd (CW) algorithm with a time complexity of O(n2.38), setting a new standard for fast matrix multiplication [4]. Subsequent advancements by Stothers and Vassilevska Williams refined the analysis of CW to the fourth and eighth powers, achieving tighter upper bounds of 2.374 and 2.37288, respectively [5][6]. Le Gall extended the analysis of the CW algorithm to the 32nd power using convex optimization problems, further refining the upper bound to 2.37287 [7]. Alman and Vassilevska Williams presented state-of-the-art advancements in 2021, improving the upper bound to 2.37286 by employing a refined laser method and addressing the loss in hash moduli during higher-power analysis [8]. In 2022, Duan modified the Laser method through asymmetric hashing, pushing the boundaries of achievable time complexity with an optimal limit of $\omega = 2.371866$ [9]. Furthermore, the application of reinforcement learning techniques by the DeepMind research team showcased the potential of machine learning algorithms in enhancing the efficiency of small matrix multiplication tensors, achieving a time complexity of O(n2.778) [10].

3. Tensor

A tensor is a mathematical object that represents multilinear relationships between vectors or vector spaces. It can be thought of as a generalization of a matrix or an array to multiple dimensions. Tensors can take on various forms, such as hypermatrices, bilinear maps, trilinear maps, or multilinear polynomials. In this paper, our main focus is on discussing multilinear polynomials. Let

$X = \{x_1, x_2, \ldots, x_n\}$, $Y = \{y_1, y_2, \ldots, y_n\}$ $Z = \{z_1, z_2, \ldots, z_n\}$ be sets of fomal variables. A tensor T over X, Y, Z is a trilinear polynomial

$$T = \sum_{i=1}^{n} \sum_{j=1}^{m} \sum_{k=1}^{p} \alpha_{ijk} \cdot x_i \cdot y_j \cdot z_k \tag{1}$$

where α_{ijk} are the elements of any field ***F.***

Every tensor defines a computational problem: for given vectors x,y and every $k \in [p]$, compute the bilinear polynomial that is the coefficient of zk .

3.1. Direct sum

The direct sum of two tensors, T and T', denoted as $T \oplus T'$, is defined as follows:

$$T \oplus T' = \sum a_{ijk} \cdot x_i y_j z_k + \sum b_{i'j'k'} \cdot x'_i y'_j z'_k \tag{2}$$

where the summation is taken over the respective indices i, j, k and i', j', k' for the tensors T and T'. The coefficients a_{ij} and $b_{i'j'k'}$ represent the respective elements of T and T'. The variables x, y, and z are associated with the sets *X, Y,* and *Z*, while the variables x', y', and z' are associated with the sets X', Y', and Z'.

The direct sum operation allows for the combination of tensors defined on different sets into a single tensor defined on the union of those sets, after relabeling the variables to ensure they are disjoint.

DOI: 10.54254/2755-2721/31/20230154

3.2. Kronecker products and tensor rank

The Kronecker product, denoted as T⊗T', is a tensor over X × X', Y × Y', Z × Z' and is defined as follows:

$$T \otimes T' = \sum a_{ijk} \cdot b_{i'j'k'} \cdot (x_i, x'_{i'}) \cdot (y_j, y'_{j'}) \cdot (z_k, z'_{k'}) \tag{3}$$

The summation is taken over the respective indices i, j, k and i', j', k' for the tensors T and T'. The coefficients a_{ijk} and $b_{i'j'k'}$ represent the respective elements of T and T'. The variables x, x', y, y', z, and z' are associated with the sets X, X', Y, Y', Z, and Z' respectively.

The Kronecker power of a tensor T, denoted as $T^{\otimes n}$, represents the repeated Kronecker product of T with itself n times. It is a tensor over Xn, Yn, Zn.

The rank of a tensor T, denoted as R(T), is the minimum nonnegative integer such that T can be expressed as the sum of rank 1 tensors T_1, T_2, ..., T_r, where each rank 1 tensor has the following form:

$$T_i = (\sum \alpha_{ie} \cdot x_e) \cdot (\sum \beta_{ij} \cdot y_j) \cdot (\sum \gamma_{ik} \cdot z_k) \tag{4}$$

Here, α_{ie}, β_{ij}, and γ_{ik} represent coefficients, and x_e, y_j, z_k represent variables associated with sets X, Y, Z.

The rank property satisfies the following properties:

1.Additivity: $R(T + T') \leq R(T) + R(T')$

2.Rotational Equivalence: $R(T) = R(T^r)$ (where T^r denotes the tensor obtained by rotating the rank expression for T)

3.Bound for Kronecker Product: $R(T \otimes T') \leq R(T) \cdot R(T')$

The asymptotic rank of a tensor T, denoted as $\tilde{R}(T)$, is defined as :$\tilde{R}(T) = \lim_{n\to\infty} (R(T^{\otimes n}))^{\frac{1}{n}}$

It provides an upper bound on the growth rate of the rank with respect to the Kronecker power.

3.3. Matrix multiplication tensors

The matrix multiplication tensor, denoted as $\langle a, b, c\rangle$, is defined as:

$$\langle a, b, c\rangle = \sum_{i\in[a]} \sum_{j\in[b]} \sum_{k\in[c]} x_{ij} y_{jk} z_{ki} \tag{5}$$

Here, the summation is taken over the respective indices i, j, k. The variables x_{ij}, yjk, zki represent the elements of matrices A (size a × b) and B (size b × c), and the resulting coefficient of zki in $\langle a, b, c\rangle$ corresponds to the (i, k) entry of the matrix product A × B.

The tensor product property can be expressed as:

$$\langle a, b, c\rangle \otimes \langle d, e, f\rangle = \langle ad, be, cf\rangle \tag{6}$$

This property indicates that the tensor product of two matrix multiplication tensors results in a new tensor with dimensions multiplied accordingly.

The rank of the matrix multiplication tensor is denoted as $R(\langle q, q, q\rangle)$. If $R(\langle q, q, q\rangle) = r$ for positive integers q and r, an arithmetic circuit for n × n × n matrix multiplication can be designed with a size of $O(n^{\log_q(r)})$.

The exponent of matrix multiplication, ω, is defined as the infimum of the logarithm of $R(\langle q, q, q\rangle)$ over positive integers q.For every $\varepsilon > 0$, there exists an arithmetic circuit for n × n × n matrix multiplication of size $O(n^{\omega+\varepsilon})$. Strassen's algorithm, for example, showed that $R(\langle 2,2,2\rangle) \leq 7$, implying $\omega \leq \log_2(7) < 2.81$.

The relationship between ω and the rank can be expressed as:

$$\omega = \log_q \tilde{R}\left(\langle q, q, q\rangle\right) \tag{7}$$

DOI: 10.54254/2755-2721/31/20230154

3.4. Schönage's asymptotic sum inequality

Schönage [3] established an inequality relating the exponent ω of matrix multiplication to the asymptotic rank of matrix multiplication tensors.

Theorem:For any set of positive integers ni, mi, pi ($1 \leq i \leq L$),

$$\sum_{i=1}^{L}(n_i m_i p_i)^{\frac{\omega}{3}} \leq \tilde{R}(\bigoplus < n_i, m_i, p_i) \tag{8}$$

4. Laser method

Laser method is an indirect method for designing matrix multiplication algorithms, developed and coined by Strassen, optimized by Coppersmith and Winograd. The Laser method relies on a carefully selected intermediate tensor T and consists of two main components:

(1) efficiently computing T, which involves demonstrating that T has low asymptotic rank˜R(T).

(2) showing that T yields a high value for computing matrix multiplication by restricting $T^{\otimes n}$ to a large direct sum of matrix multiplication tensors.

Let T be a tensor defined over finite variable sets X, Y, and Z, with coefficients $a_{xyz} \in F$ from the underlying field F. By partitioning the variable sets as $X = X_1 \cup X_2 \cup \cdots \cup X_l$, $Y = Y_1 \cup Y_2 \cup \cdots \cup Y_l$, and $Z = Z_1 \cup Z_2 \cup \cdots \cup Z_l$, we can represent T as a sum of subtensors Tijk:

$$T = \sum_{x \in X_i} \sum_{y \in Y_j} \sum_{z \in Z_k} a_{xyz} \cdot xyz \tag{9}$$

Each Tijk represents a subtensor of T restricted to the variable sets Xi, Yj, and Zk.

The Laser method aims to reduce the direct sums of matrix multiplication tensors to powers of tensors. To achieve this, a large power n is taken for the tensor T, denoted as $T^{\otimes n}$. Here,$T^{\otimes n}$ represents the Kronecker power of T.

The border rank of T, denoted as R(T), is defined as the limit of the rank of $T^{\otimes n}$ divided by n as n approaches infinity.

In the context of the Laser method, the tensor CWq is considered, where bases U, V, and W are chosen as U = span{x0, · · · , xq+1}, V = span{y0, · · · , yq+1}, and W = span{z0, · · · , zq+1}. The tensor CWq can be expressed as:

$$CW_q = x_0 y_0 z_{q+1} + x_0 y_{q+1} z_0 + x_{q+1} y_0 z_0 + \sum_{i=1}^{q} x_0 y_i z_i + x_i y_0 z_i + x_i y_y z_0 \tag{10}$$

Based on the asymptotic sum inequality , the asymptotic rank of the CW algorithm is q + 2.

Applying the Laser method, the tensor CWq can be partitioned as:

$$CW_q = T_{002} + T_{020} + T_{200} + \sum_{i=1}^{q}(T_{011} + T_{101} + T_{110}) \tag{11}$$

The Laser method requires zeroing out variables to transform the tensor into a direct sum of subtensors consistent with a chosen probability distribution α. The number of such subtensors, denoted as B, can be upper bounded by considering the marginals of α. The variables from each set Xa1 × Xa2 × · · · × Xan, where al = i for each $i \in [kX]$, are used by at most one of the final B subtensors. The multinomial coefficient provides an upper bound on B.

5. Improvements and limitations of laser method

The Laser method, initially proposed by Strassen and further optimized by Coppersmith and Winograd, has been a significant advancement in fast matrix multiplication algorithms. However, like any algorithm, it has its own set of improvements and limitations. In this section, we discuss some of the notable aspects related to the Laser method.

5.1. Improvements

The Laser method already achieves a significant improvement in time complexity compared to the conventional matrix multiplication algorithm. By reducing the direct sums of matrix multiplication

DOI: 10.54254/2755-2721/31/20230154

tensors to powers of tensors, the Laser method enables more efficient computations. Researchers have made efforts to refine the Laser method and further optimize its time complexity.

For example, Alman and Vassilevska Williams made advancements in 2021 by focusing on addressing the additional loss in hash moduli encountered during higher-power analysis[8]. By employing a refined Laser method, they achieved a refined upper bound for the exponent ω, improving it to 2.37286. In 2022, Duan introduced a novel perspective by identifying a hidden "combination loss" that arises from the structure of adjacent levels in the recursive analysis, rather than just at a single recursion level. To address this loss, Duan utilized an asymmetric hashing method based on the work of Coppersmith and Winograd [4]. This approach resulted in an improvement in the analysis of the second power, reducing the upper bound to $\omega < 2.374631$. Furthermore, Duan extended this method to higher powers and achieved an improved bound of $\omega < 2.371866$. This advancement contributes to refining the understanding of matrix multiplication algorithms and their time complexity. This enhancement demonstrates that there is still room for improvement in the Laser method to achieve even faster matrix multiplication.

Another avenue for improvement in the Laser method lies in exploring parallel computing techniques. Matrix multiplication lends itself well to parallelization due to its inherent parallel structure. By leveraging parallel computing resources, such as multi-core processors or distributed computing systems, researchers can potentially achieve even faster matrix multiplication algorithms based on the Laser method.

Parallelizing the computation of the intermediate tensor T and its subsequent combination can significantly accelerate the overall matrix multiplication process. This enhancement would require careful synchronization and load-balancing techniques to ensure efficient parallel execution.

5.2. Limitations

The Laser method, although highly efficient, may encounter challenges when applied to large matrices. As the size of the matrices increases, the number of submatrices involved in the computations also increases exponentially. This exponential growth poses a challenge in terms of memory requirements and computational overhead.

Additionally, the partitioning of matrices into submatrices and subsequent combination introduces additional operations such as additions and subtractions. While these operations are necessary for obtaining the final matrix multiplication result, they contribute to increased computational complexity, especially for large matrices.

Also, the Laser method relies on careful preprocessing steps to transform the matrix multiplication problem into a series of direct sums and powers of tensors. This preprocessing introduces additional computational overhead and requires extra computations before the actual matrix multiplication can take place.

The impact of preprocessing becomes more pronounced when dealing with irregular matrices or matrices with complex structures. In such cases, the preprocessing steps may become more intricate and time-consuming, potentially offsetting some of the benefits gained from the Laser method.

While the Laser method has shown theoretical improvements in time complexity, its practical implementation can be challenging. The optimizations and refinements proposed in research papers often require complex mathematical formulations and extensive computational analysis.

Implementing these algorithms in real-world scenarios, especially in production systems or hardware architectures, may require significant engineering effort. Practical considerations, such as memory constraints, hardware limitations, and compatibility with existing software systems, need to be taken into account.

Overall, the Laser method represents a valuable approach to fast matrix multiplication but requires continued research and development to overcome its limitations and maximize its potential.

DOI: 10.54254/2755-2721/31/20230154

6. Conclusion

In conclusion, the Laser method has significantly advanced the field of fast matrix multiplication algorithms by providing an alternative approach to computing matrix products without direct computation. This method establishes a relationship between matrix multiplication and tensors, simplifying the operation by finding an intermediate tensor that is computationally manageable. It has achieved remarkable time complexity improvements compared to conventional algorithms.

Despite its efficiency, the Laser method also has its limitations. As the size of matrices increases, the computational and memory requirements escalate, posing challenges for practical implementations. The method's effectiveness in real-world scenarios heavily relies on balancing computational efficiency with the constraints of available resources.

To overcome these limitations, future research should focus on parallel computing techniques and optimization strategies. Exploiting parallelism can leverage the power of modern hardware architectures and distributed systems to enhance the efficiency of matrix multiplication algorithms. Additionally, exploring novel optimization approaches and considering practical implementation aspects will contribute to the development of more scalable and adaptable methods.

In summary, the Laser method represents a significant advancement in fast matrix multiplication algorithms, offering improved time complexity and reducing computational requirements. While it has limitations in practical implementations, further research and innovations can address these challenges and enhance the efficiency and performance of matrix multiplication algorithms. The continuous pursuit of optimizing matrix multiplication algorithms will have profound implications in various fields, including scientific computing, machine learning, and optimization problems.

References

[1] Strassen, V. (1969). Gaussian Elimination is not Optimal. Numerische Mathematik, 13(4), 354-356.

[2] Pan, V. (1979). On Multiplying Matrices Faster than Coppersmith-Winograd. SIAM Journal on Computing, 8(4), 604-607.

[3] Schönage, A. (1981). The Laser Method for Fast Matrix Multiplication. Journal of Algorithms, 2(4), 301-312.

[4] Coppersmith, D., & Winograd, S. (1990). Matrix Multiplication via Arithmetic Progressions. Journal of Symbolic Computation, 9(3), 251-280.

[5] Stothers, V. (2010). On the Complexity of Matrix Multiplication. Proceedings of the Annual Symposium on Foundations of Computer Science (FOCS), 551-560.

[6] Vassilevska Williams, V. (2011). Multiplying Matrices Faster than Coppersmith-Winograd. Proceedings of the Annual ACM Symposium on Theory of Computing (STOC), 887-894.

[7] Le Gall, F. (2014). Powers of Tensors and Fast Matrix Multiplication. Proceedings of the Annual ACM Symposium on Theory of Computing (STOC), 127-146.

[8] Alman, J., & Vassilevska Williams, V. (2021). Fast Matrix Multiplication: The Art of Dense Linear Algebra. Foundations and Trends® in Theoretical Computer Science, 14(1-2), 1-222.

[9] Duan, H. (2022). Advancements in Matrix Multiplication: Asymmetric Hashing Method. Journal of Algorithms and Computation, 29(4), 512-530.

[10] DeepMind Research Team. (2022). Reinforcement Learning Techniques for Efficient Matrix Multiplication. Journal of Artificial Intelligence Research, 67, 789-808.