Chittagong University of Engineering and Technology

Department of Computer Science and Engineering

Research Proposal Paper

Course Title: Algorithm Design and Analysis(Sessional)

Course ID : CSE-244

Student ID : 1804087, 1804083, 1804079

Level : 02

Term : 02

Date : 02.01.22

Remarks

Abstract

The goal of this study is to lower the temporal complexity of a matrix multiplication algorithm. Matrix multiplication is a crucial mathematical technique that can be used to tackle a variety of engineering and scientific challenges. As a result, in order to do various scientific computations, it is critical to calculate the multiplication of matrices. In general, the naïve algorithm takes O(n3) time to calculate the product of matrices of higher order, which is a time-consuming task for a desktop processor. As a result, the goal of this study is to discover an algorithm that can multiply matrices in less time. We're working on a solution to lower the temporal complexity of Strassen's approach. In general, Strassen's algorithm executes in less than a cubic second. According to our findings, it can be done in less time and with fewer arithmetic operations than Strassen's.

Introduction

The multiplication of two matrices is one of the most fundamental operations in linear algebra and scientific computing, and it has become a focal point in the hunt for strategies to accelerate scientific computation. The first sub-cubic matrix multiplication algorithm with temporal complexity O was Strassen's algorithm (n2.81). We offer a new approach for computing A.AT that uses less mathematical computations to calculate the product.

Objective

- To reduce the mathematical operations required to solve the algorithm.
- To reduce the time complexity of matrix multiplication.

Related work

Bodrato introduced a method for chain matrix multiplication which reduces additions and subtractions and seven multiplications in consecutive multiplicatios. In case of repeated squaring, the multiplication can be done using a leading co-efficient of 5. His algorithm works for those matrices whose degree is the power of 2. Cenk and Hassan has shown a clever method for applying Strassen's algorithm in multiplying square matrices of any order. Although their algorithm consumes more memory than Strassen's algorithm as they reuse the variables.

Preliminaries

Strassen's Algorithm for Matrix Multiplication:

To multiply two matrices, we can use neive algorithm which takes $O(n^3)$ time. Strassen came up with an idea to reduce the mathematical calculations. Strassen provided some formulae to calculate the product of two 2x2 matrices with fewer calculations where there are 7 multiplications and 18 additions or subtractions. This leads to a time complexity of $O(n^{2.81})$ in higher order matrices multiplications. That's why it outperforms the neive approch which requires three for loops.

Proposed Methodology

Considered as 2×2 matrices, the matrix product $C = A \cdot B$ could be computed using Strassen algorithm by performing the following computations:

$p1 \leftarrow a11.(b12 - b22)$,

$p2 \leftarrow (a11 + a12).b22$,

$p3 \leftarrow (a21 + a22).b11$,

$p4 \leftarrow (a12 - a22).(b21 + b22)$,

$p5 \leftarrow (a11 + a22).(b11 + b22)$,

$p6 \leftarrow a22.(b21 - b11)$,

$p7 \leftarrow (a21 - a11).(b11 + b12)$,

$$\begin{pmatrix} c1 & c2 \\ c3 & c4 \end{pmatrix} = \begin{pmatrix} p5 + p4 - p2 + p6 & p6 + p3 \\ p2 + p1 & p5 + p7 + p1 - p3 \end{pmatrix}$$

Now, if A be a 2x2 matrix and B is the transpose matrix of A, we can reduce the number of matrix multiplications by following the bellow calculations.

Input: $A = \begin{pmatrix} a11 & a12 \\ a21 & a22 \end{pmatrix}$ $\quad B = \begin{pmatrix} b11 & b12 \\ b21 & b22 \end{pmatrix}$ which is a transpose of matrix A.

s1 ← a11 – a21

s2 ← a21 + a22

s3 ← s2 – a11

s4 ← a12 – s3

t1 ← b22 – b12

t2 ← b12 – b11

t3 ← b11 + t1

t4 ← b21 – t3

p1 ← a11 . b11

p2 ← a12 . b21

p3 ← a22 . t4

p4 ← s1 . t1

p5 ← s3 . t3

p6 ← s4 . b22

p7 ← s2 . t2

C1 ← p1 + p5

C2 ← C1 + p4

C3 ← p1 + p2

C4 ← C2 + p3

C5 ← C2 + p7

C6 ← C1 + p7

C7 ← C6 + p6

return C = $\begin{pmatrix} C3 & C7 \\ C4 & C5 \end{pmatrix}$

The above formula have 7 multiplication but it has less addition and subtarction than Strassen's algorithm. Strassen's algorithm has 18 additions or subtractions but the above algorithm has 15 additions or subtractions.So, if we use the above formula and continue with the divide and conqure approach for higher order matrices, the time taken should be less than that of Strassen's algorithm because of having less addition and subtarction.

References

[1]https://www.researchgate.net/publication/264729252_Comparative_Study_of_Strassen's_Matrix_Multiplication_Algorithm

[2]https://en.wikipedia.org/wiki/Strassen_algorithm

[3]https://www.researchgate.net/publication/2779622_Implementation_of_Strassen's_Algorithm_for_Matrix_Multiplication