IOSR Journal of Engineering (IOSRJEN) **www.iosrjen.org**
ISSN (e): 2250-3021, ISSN (p): 2278-8719
Vol. 11, Issue 1, January 2021, ||Series -I|| PP 01-03

Optimizing Divide and Conquer Method for Matrix Multiplication Algorithm

Tancy V S
Department of Computer Science, Mary Matha Arts and Science College Mananthavady, Kannur University, Wayanad, Kerala
Received 06 January2021; Accepted 21 January2021

Abstract: There are many different algorithms for matrix multiplication. Commonly used matrix multiplications are conventional matrix multiplication and matrix multiplication using divide and conquer method including Strassen's matrix multiplication. In this paper I optimize the divide and conquer method for matrix multiplication by avoiding recomputation of the matrix elements. This new method is useful when multiplicands biggest and nearest digits.
Keywords: Algorithm, Matrix multiplication optimization, divide and conquer

I. INTRODUCTION

In ancient time's pictures in various areas such as architecture, cartoons etc. were performed by human by hand. The entry of computer graphics changed these all. Matrix multiplications are widely used in computer graphics for transformations such as rotation and scaling.

Matrix multiplication of two matrices A and B is defined as if A is an $n \times m$ matrix and B is an $m \times p$ matrix, their matrix product C is an $n \times p$ matrix, in which the m entries across a row of A are multiplied with the m entries down a column of B and summed to produce an entry of C.
Ex.

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix} * \begin{bmatrix} E & F \\ G & H \end{bmatrix} = \begin{bmatrix} AE+BG & AF+BH \\ CE+DG & CF+DH \end{bmatrix}$$

As I mentioned before there are many methods to calculate the multiplication of matrices. All of them give same result but each one consumes different space in memory and takes different processor time.

II. CONVENTIONAL MATRIX MULTIPLICATION

The matrix multiplication can be applied only if the number of columns of the first matrix is equal to the number of rows of second matrix and resultant matrix will have the row size same as first matrix and column size same as second matrix.
The conventional matrix multiplication algorithm is given below.

Input: matrices A_{nm} and B_{mp}
Output: C_{np} which is the multiplication result of A and B

For i from 1 to n:
For j from 1 to p:
  set $C_{ij} = 0$
  For k from 1 to m:
    Set $C_{ij} \leftarrow C_{ij} + A_{ik} \times B_{kj}$
Return C

This algorithm takes time O(nmp). A common simplification for the purpose of algorithms analysis is to assume that the inputs are all square matrices of size $n \times n$, in which case the running time is $O(n^3)$, i.e., cubic.

III. DIVIDE AND CONQUER METHOD

In divide and conquer method as we know that the matrix is divided into sub matrixes. Division continues until we get a 2 by 2 matrix.
Here if A and B are two matrices. And the product is stored in C
Then product AB is as follows

$$\begin{bmatrix} A11 & A12 \\ A21 & A22 \end{bmatrix} * \begin{bmatrix} B11 & B12 \\ B21 & B22 \end{bmatrix} = \begin{bmatrix} C11 & C12 \\ C21 & C22 \end{bmatrix}$$

Here $C_{11}=A_{11}B_{11}+A_{12}B_{21}$
$C_{12}=A_{11}B_{12}+A_{12}B_{22}$
$C_{21}=A_{21}B_{11}+A_{22}B_{21}$
$C_{22}=A_{21}B_{12}+A_{22}B_{22}$

IV. PROPOSED ALGORITHM

We can see that in divide and conquer A_{11} is multiplied with B_{11} and B_{12}, A_{12} is multiplied with B_{21} and B_{22}, A_{21} is multiplied with B_{11} and B_{12}, and A_{22} is multiplied with B_{21} and B_{22}. That is A_{11} is multiplier for both B_{11} and B_{12} and A_{12} is multiplier for both B_{21} and B_{22} and so on. So in this proposed method, if A_{11} is multiplied with B_{11} then we can reapply this result to get $A_{11}*B_{12}$ thereby we can save the multiplication time. This new method is useful when multiplicands biggest and nearest digits.

Algorithm can be written as follows.

- Input matrix A and B
- Divide both matrices until we get matrices of size 2*2
- For each 2*2 matrices perform following

If ($B_{11} < B_{12}$)
{
$D= B_{12}-B_{11}$
$X_1=A_{11}*B_{11}$
$X_2=X_1+A_{11}*D$
$X_3=A_{21}*B_{11}$
$X_4=X_3+A_{21}*D$
}
Else
{
$D= B_{11}-B_{12}$
$X_2=A_{21}*B_{12}$
$X_1=X_2+A_{21}*D$
$X_4=A_{21}*B_{12}$
$X_3=X_4+A_{21}*D$
}
If ($B_{21}<B_{22}$)
{
$D= B_{22}-B_{21}$
$Y_1=A_{12}*B_{21}$
$Y_2=Y_1+A_{12}*D$
$Y_3=A_{22}*B_{21}$
$Y_4=Y_3+A_{22}*D$
}
Else
{
$D= B_{21}-B_{22}$
$Y_2=A_{12}*B_{22}$
$Y_1= Y_2+A_{12}*D$
$Y_4=A_{22}*B_{22}$
$Y_3=Y_4+A_{22}*D$
}
$C_{11}=X_1+Y_1$
$C_{12}=X_2+Y_2$

$C_{21}=X_3+Y_3$
$C_{22}=X_4+Y_4$

- Join the matrices

Here there is no change in the number of multiplication, but through this method we can reduce the time for multiplication.

V. CONCLUSION

As we know now a day's matrix plays an important role in computer science too. So there are different types of algorithm available for matrix manipulation. Here I also try to find out the new matrix multiplication which reduces the time for multiplication.

REFERENCES

[1]. Alfred, V, Aho., John, E, Hopcroft., Jeffrey, D, Ullman., Design & Analysis of Computer Algorithms,Pearson
[2]. Khaled, Thabet., King, Abdulaziz. Matrix Multiplication Algorithms., International Journal of Computer Science and Network Security, VOL.12 No.2, February 2012
[3]. Pawan, Rathod., Ankit, Vartak., Neha, Kunte., Optimizing the Complexity of Matrix Multiplication Algorithm., 2017 International Conference on Intelligent Computing and Control (I2C2'17)

Tancy V S. "Optimizing Divide and Conquer Method for Matrix Multiplication Algorithm." *IOSR Journal of Engineering (IOSRJEN),* 11(01), 2021, pp. 01-03.