Matrix Multiplication Using Divide-&-Conquer

Textbook Reading:

- Section 7.4 Multiplication of Matrices, pp. 292-295
- Section 7.5 Closing Remarks, first two paragraphs, pp. 320-321.

Matrix Multiplication is Composition of Linear Transformations

$$\begin{bmatrix} y_0 \\ y_1 \end{bmatrix} = \begin{bmatrix} b_{00} & b_{01} \\ b_{10} & b_{11} \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \end{bmatrix} = \begin{bmatrix} b_{00}x_0 + b_{01}x_1 \\ b_{10}x_0 + b_{11}x_1 \end{bmatrix}$$

$$\begin{bmatrix} z_0 \\ z_1 \end{bmatrix} = \begin{bmatrix} a_{00} & a_{01} \\ a_{10} & a_{11} \end{bmatrix} \begin{bmatrix} y_0 \\ y_1 \end{bmatrix} = \begin{bmatrix} a_{00}y_0 + a_{01}y_1 \\ a_{10}y_0 + a_{11}y_1 \end{bmatrix}$$

Substitute for y_0 and y_1 from first equation into second equation

$$\begin{bmatrix} z_0 \\ z_1 \end{bmatrix} = \begin{bmatrix} a_{00}(b_{00}x_0 + b_{01}x_1) + a_{01}(b_{10}x_0 + b_{11}x_1) \\ a_{10}(b_{00}x_0 + b_{01}x_1) + a_{11}(b_{10}x_0 + b_{11}x_1) \end{bmatrix}$$

$$= \begin{bmatrix} a_{00}b_{00} + a_{01}b_{10} & a_{00}b_{01} + a_{01}b_{11} \\ a_{10}b_{00} + a_{11}b_{10} & a_{10}b_{01} + a_{11}b_{11} \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \end{bmatrix}$$

Matrix Multiplication General Formula for Square Matrices

Consider two $n \times n$ matrices

$$A = (a_{ij}) \text{ and } B = (b_{ij}),\ 0 \le i, j \le n - 1$$

The product AB is defined to be the $n \times n$ matrix $C = (c_{ij})$, where

$$c_{ij} = a_{i1} * b_{1j} + a_{i2} * b_{2j} + \ldots + a_{in} * b_{nj}\ ,\ 0 \le i, j \le n - 1$$

Straightforward Algorithm

We will take the input size to be n for the problem of multiplying two $n \times n$ matrices. The straightforward algorithm for solving this problem is to simply perform the n multiplications to compute each c_{ij} , $0 \le i, j \le n - 1$, resulting in a total number of multiplications of

$$W(n) = n^3.$$

2×2 matrices

Multiplying two 2×2 matrix involves 8 multiplications to obtain product matrix

$$C = AB = \begin{bmatrix} a_{00} & a_{01} \\ a_{10} & a_{11} \end{bmatrix} \begin{bmatrix} b_{00} & b_{01} \\ b_{10} & b_{11} \end{bmatrix}$$

$$\begin{bmatrix} c_{00} & c_{01} \\ c_{10} & c_{11} \end{bmatrix} = \begin{bmatrix} a_{00}b_{00} + a_{01}b_{10} & a_{00}b_{01} + a_{01}b_{11} \\ a_{10}b_{00} + a_{11}b_{10} & a_{10}b_{01} + a_{11}b_{11} \end{bmatrix}$$

Strassen's method for 2×2 matrices

In 1970 Strassen discovered a way to multiply two 2×2 matrices using only 7 multiplications!

$$m_1 = (a_{00} + a_{11})*(b_{00} + b_{11})$$
$$m_2 = (a_{10} + a_{11})*b_{00}$$
$$m_3 = a_{00}*(b_{01} - b_{11})$$
$$m_4 = a_{11}*(b_{10} - b_{00})$$
$$m_5 = (a_{00} + a_{01})*b_{11}$$
$$m_6 = (a_{10} - a_{00})*(b_{00} + b_{01})$$
$$m_7 = (a_{01} - a_{11})*(b_{10} + b_{11})$$

Combining 7 Multiplications

$C = AB =$

$$\begin{bmatrix} m_1 + m_4 - m_5 + m_7 & m_3 + m_5 \\ m_2 + m_4 & m_1 + m_3 - m_2 + m_6 \end{bmatrix}$$

Verification of Correctness

Consider entry in first row and first column

$$m_1 + m_4 - m_5 + m_7 = (a_{00} + \mathrm{a}_{11})*(b_{00} + b_{11})$$
$$+ a_{11}*(b_{10} - b_{00}) - (a_{00} + a_{01})*b_{11} + (a_{01} - a_{11})*(b_{10} + b_{11})$$

$$= (a_{00}b_{00} + a_{00}b_{11} + \mathrm{a}_{11}b_{00} + \mathrm{a}_{11}b_{11}) + (a_{11}b_{10} - a_{11}b_{00})$$
$$- (a_{00}b_{11} + a_{01}b_{11}) + (a_{01}b_{10} + a_{01}b_{11} - a_{11}b_{10} - a_{11}b_{11})$$

$$= a_{00}b_{00} + a_{01}b_{10} = c_{00}$$

This verifies that the 00 entry is correct. In the homework you will verify that all four entries are correct.

PSN. Verify correctness for c_{01}.

Now Apply Divide-&-Conquer

We now can multiply two 2$\times$2 matrix using only 7 multiplications!

What about general matrices? This is where Divide-&-Conquer comes in.

Divide Step

Consider now the case of two $n \times n$ matrices, where we assume that $n = 2^k$. The Divide Step involves partitioning each of the matrices A and B into four $\frac{n}{2} \times \frac{n}{2}$ submatrices:

$$A = \left[\begin{array}{c|c} A_{00} & A_{01} \\ \hline A_{10} & A_{11} \end{array}\right], \quad B = \left[\begin{array}{c|c} B_{00} & B_{01} \\ \hline B_{10} & B_{11} \end{array}\right]$$

Combine Step – Inefficient

The four $\frac{n}{2} \times \frac{n}{2}$ matrix blocks act as the four entries A_{01}, A_{02}, A_{10}, A_{11} of a 2×2 matrix A. Similarly for B. We obtain the matrix product of the two $n \times n$ matrices A and B by multiplying these two 2×2 matrices, each consisting of four $\frac{n}{2} \times \frac{n}{2}$ matrix blocks

$$\begin{bmatrix} A_{00}B_{00} + A_{01}B_{10} & A_{00}B_{01} + A_{01}B_{11} \\ A_{10}B_{00} + A_{11}B_{10} & A_{10}B_{01} + A_{11}B_{11} \end{bmatrix}$$

However, this will not give us any improvement over the straightforward algorithm.

Combine Step – Efficient

A more efficient way to combine is using Strassen's method for multiplying 2×2 matrices. We first perform the 7 matrix multiplications

$$
\begin{aligned}
M_1 &= (A_{00} + A_{11})(B_{00} + B_{11}) \\
M_2 &= (A_{10} + A_{11})B_{00} \\
M_3 &= A_{00}(B_{01} - B_{11}) \\
M_4 &= A_{11}(B_{10} - B_{00}) \\
M_5 &= (A_{00} + A_{01})B_{11} \\
M_6 &= (A_{10} - A_{00})(B_{00} + B_{01}) \\
M_7 &= (A_{01} - A_{11})(B_{10} + B_{11})
\end{aligned}
$$

Combining 7 Multiplications

$C = AB$

$$= \begin{bmatrix} M_1 + M_4 - M_5 + M_7 & M_3 + M_5 \\ M_2 + M_4 & M_1 + M_3 - M_2 + M_6 \end{bmatrix}$$

Summary of Strassen's Algorithm

Input: Two $n \times n$ matrices A and B, where $n = 2^k$.
Output: The matrix product $C = AB$.

Divide Step. Divide each of the matrices A and B into four $\frac{n}{2} \times \frac{n}{2}$ submatrices $A_{00}, A_{01}, A_{10}, A_{11}$ and $B_{00}, B_{01}, B_{10}, B_{11}$.

Conquer Step. Perform 7 recursive calls to compute 7 matrices $M_1, M_2, \ldots, M_7$ using Strassen's Formula for 2×2 matrices.

Combine Step.

$$C = \begin{bmatrix} M_1 + M_4 - M_5 + M_7 & M_3 + M_5 \\ M_2 + M_4 & M_1 + M_3 - M_2 + M_6 \end{bmatrix}$$

Recurrence Relation for Worst-Case Complexity

Algorithm based on this Divide-&-Conquer Strategy involves 7 recursive calls with matrices of half the size. The initial condition involves multiplying matrices of size 1, i.e.,

$[a_{00}][b_{00}] = [a_{00} * b_{00}]$, which involves a single scalar multiplications. Thus, assuming $n = 2^k$

$$W(n) = 7W(n/2), \text{ initial condition } W(1) = 1.$$

Using Repeated Substitution to Solve

$W(n) = 7W(n/2) = 7(7W(n/4)) = 7^2W(n/4)$

$= 7^2(7W(n/8)) = 7^3W(n/8)$

.

.

.

$= 7^kW(n/2^k)$

$= 7^kW(1)$

$= 7^k$

$=(2^{\log_2 7})^k$

$=(2^k)^{\log_2 7} = n^{\log_2 7}$

Complexity of Improved Divide-&-Conquer Algorithm

The worst-case complexity is

$$W(n) = n^{\log_2 7} \approx n^{2.81}$$

This is a significant improvement over the cubic complexity of the straightforward algorithm.

Matrix Product for General $n \times n$ Matrices

The design of Strassen's algorithm depended on the size n of the input matrices A and B being a power of 2, otherwise they could not be divided into four square matrix blocks each of size $\frac{n}{2} \times \frac{n}{2}$. However, in practice it is important to be able to handle matrices of sizes different from a power of 2.

Solution: If n is not a power of 2, then letting k be the smallest exponent, such that $n < 2^k$, add $n - 2^k$ rows and columns consisting of all zeros. After applying Strassen's algorithm remove these rows and columns of all zeros.

Improvements over Strassen's Algorithm

Nearly ten years after Strassen discovered his identities, Pan found a way to multiply two 70×70 matrices that involves only 143,640 multiplications (compared to over 150,000 multiplications used by Strassen's method), yielding an algorithm that performs $O(n^{2.795})$ multiplications.

PSN. Verify $O(n^{2.795})$ result.

Best Known Algorithm – Coppersmith and Winograd

Improvements over Pan's algorithm have been discovered, and the best result currently known (due to Coppersmith and Winograd) can multiply two $n \times n$ matrices using $O(n^{2.376})$ multiplications.

Theory vs. Practice

Both Pan’s and Coppersmith and Winograd’s require the size *n* of the matrices to be quite large before improvements over Strassen’s method are significant. Moreover, they are very complicated due to the large number of identities required to achieve the savings in the number of multiplications. Hence, the currently known order-of-complexity improvements over Strassen’s algorithm are mostly of theoretical rather than of practical interest.

Bagel Matrices

1. What is holding the bagel back?

2. Who did the bagel call when it was locked out of the house?

Answer

$$\begin{bmatrix} \text{lox} \\ \text{loxsmith} \end{bmatrix}$$