Subcubic Parallelized Matrix Inversion

Through Strassen's Algorithm

Andres Valdes

April, 2018

Abstract

Matrix inversion is one of the most computationally complex matrix operations. This is unfortunate, as matrix inversion has many practical applications, including encryption, where the contents of a matrix are obfuscated by multiplying by a square, invertible cypher matrix and can only be retrieved my multiplying it by the inverse of that cypher.

1 Introduction

A matrix is a rectangular array of numbers or expressions, for example:

$$\begin{bmatrix} a & b & c & d \\ e & f & g & h \end{bmatrix}$$

This is a what is referred to as a 2×4 matrix, as it has two rows and four columns. In general, we speak about matrices as $m \times n$, where m is the number of rows and n is the number of rows. The types of matrices we will deal with are called "square" matrices, this is the case that we have an $m \times n$ matrix such that $m = n$, we refer to these also as $n \times n$ matrices. Only square, non-singular[1] matrices are invertible.

The inverse of a matrix A, is a matrix A^{-1} that when multiplied by A, yields the identity matrix, I. I_n is the square, $n \times n$ matrix with the top-leftmost to the bottom-rightmost diagonal filled in with ones, and where every other element is a zero.

$$I_n = \begin{bmatrix} 1 & 0 & 0 & \dots & 0 \\ 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & 0 \\ 0 & 0 & 0 & \dots & 1 \end{bmatrix}_{n \times n}$$

So $AA^{-1} = I$, this is the linear algebra equivalent of $a\frac{1}{a} = 1$.

Traditionally, the inversion of a matrix involves the use of the formula

$$A^{-1} = \frac{1}{det(A)} adj(A), \; adj(A) = (C_A)^T$$

Where C_A is the coefficient matrix of A, that is, a matrix of the dimension of A with determinants filled in for each of the elements.

This is a particularly heavy computation, as the common algorithm for determinant computation is of the computation complexity $O(n!)$, with the best algorithm being $O(n^3)$, forcing us to do $O(n^5)$ computations for the coefficient matrix alone, as there are n^2 elements in A.

[1] A matrix whose determinant is nonzero.

Thankfully, we do have a way of reducing the problem for the calculation of a matrix inverse via an analytic inversion formula known as blockwise inversion. In blockwise inversion, we split our matrix M into blocks, such that

$$M^{-1} = \begin{bmatrix} A & B \\ C & D \end{bmatrix}^{-1} = \begin{bmatrix} A^{-1} + A^{-1}B(D - CA^{-1}B)^{-1}CA^{-1} & -A^{-1}B(D - CA^{-1}B)^{-1} \\ -(D - CA^{-1}B)^{-1}CA^{-1} & (D - CA^{-1}B)^{-1} \end{bmatrix}$$

The advantage to this divide-and-conquer algorithm is that it can be shown that the computational complexity of the entire inversion is the same as the computational complexity of the multiplication algorithm used. This is where Strassen's algorithm, with a complexity of $O(n^{log_2 7})$, comes into play.

Strassen describes an algorithm to multiply two matrices by splitting each into four matrix blocks of equal dimensions and performing seven multiplications between those blocks as opposed to eight.

$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix},\ B = \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{bmatrix},\ AB = C = \begin{bmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{bmatrix}$$

We define seven new matrices

$$\begin{aligned} M_1 &\coloneqq (A_{11} + A_{22})(B_{11} + B_{22}) \\ M_2 &\coloneqq (A_{21} + A_{22})B_{11} \\ M_3 &\coloneqq A_{11}(B_{12} - B_{22}) \\ M_4 &\coloneqq A_{22}(B_{21} - B_{11}) \\ M_5 &\coloneqq (A_{11} + A_{12})B_{22} \\ M_6 &\coloneqq (A_{21} - A_{11})(B_{11} + B_{12}) \\ M_7 &\coloneqq (A_{12} - A_{22})(B_{21} + B_{22}) \end{aligned}$$

Note that for each matrix, we multiply only once. We can now compose these matrices into C, or AB.

$$\begin{aligned} C_{11} &= M_1 + M_4 - M_5 + M_7 \\ C_{12} &= M_3 + M_5 \\ C_{21} &= M_2 + M_4 \\ C_{22} &= M_1 - M_2 + M_3 + M_6 \end{aligned}$$

And thus, we've achieved matrix multiplication in subcubic time, and by extension, we can achieve matrix inversion in subcubic time.