Produto de matrizes: uma abordagem mais eficiente

*Pedro Soares da Silva Neto**

RESUMO

O produto de duas matrizes nxn exige $\Theta(n^3)$ passos com o uso do algoritmo tradicional. Será apresentado o algoritmo de Strassen que reduz essa complexidade para $\Theta(n^{2.81})$.

1. INTRODUÇÃO

Sejam A e B duas matrizes nxn a serem multiplicadas, e seja C o seu produto. O algoritmo clássico de produto de matrizes é obtido diretamente da definição:

$$C_{ij} = \sum_{k=l}^{n} A_{ik} B_{kj}.$$

Cada elemento da matriz C é calculado em um tempo $\Theta(n)$ (gasta n passos), ao todo, n produtos e n somas. Assume-se que adição escalar e multiplicação são operações elementares. Desde que há n^2 elementos a serem calculados para obtenção de C, o produto de A e B pode ser calculado em $\Theta(n^3)$.

No final da década de 60, Strassem apresentou uma melhora para esse algoritmo. A idéia básica é baseada na estratégia dividir para conquistar. Deseja-se obter o produto C = AB, onde A, B e C são matrizes nxn. Assumindo-se que n é uma potência de 2, pode-se dividir cada uma das matrizes em quatro submatrizes n/2 x n/2 :

$$A = \begin{bmatrix} A_{11} A_{12} \\ A_{21} A_{22} \end{bmatrix} \text{ e } B = \begin{bmatrix} B_{11} B_{12} \\ B_{21} B_{22} \end{bmatrix}$$

O produto será obtido mediante as seguinte operações:

*Maj QEM – Instituto Militar de Engenharia – Departmento de Sistemas e Computação.

– $M_1 = (A_{12} - A_{22})(B_{21} + B_{22})$

– $M_2 = (A_{11} + A_{22})(B_{11} + B_{22})$

– $M_3 = (A_{11} - A_{21})(B_{11} + B_{12})$

– $M_4 = (A_{11} + A_{12})B_{22}$

– $M_5 = A_{11}(B_{12} - B_{22})$

– $M_6 = A_{22}(B_{21} - B_{11})$

– $M_7 = (A_{21} + A_{22})B_{11}$

A matriz é dada por C = $\begin{vmatrix} (M_1 + M_2 - M_4 + M_6)..(M_4 + M_5) \\ (M_6 + M_7) \;................. \;(M_2 - M_3 + M_{45} - M_5) \end{vmatrix}$;

A descoberta de Strassen é uma abordagem recursiva que exige apenas 7 multiplicações de matrizes que gastam $\Theta(n^3)$ passos cada, e 24 adições. de matrizes com $\Theta(n^2)$ passos cada uma.

2. O ALGORITMO DE STRASSEN

Algoritmo *Strassen* (A,B,n)

– /*Retoma M = A*B, onde A e B são matrizes nxn e n é assumido como potência de 2*/

– **início**
– **se** n = 1 **faça então**
– **retornar** M = A*B/*multiplicação escalar*/;
– **senão**

– Particionar A = $\begin{vmatrix} A_{12} & A_{12} \\ A_{12} & A_{12} \end{vmatrix}$ e B= $\begin{vmatrix} B_{12} & B_{12} \\ B_{12} & B_{12} \end{vmatrix}$, onde Aij e Bij são matrizes de tamanho (n/2) x (n/2);

– M_1 = Strassen($A_{12} - A_{22}$, B21+ B22,n/2);

– M_2 = Strassen($A_{11} + A_{22}$, B_{11} + B22, n/2);

– M_3 = Strassen(A_{11} - A_{21}, $B_{11} + B_{12}$, n/2);

– M_4 = Strassen($A_{11} + A_{12}$, B_{22}, n/2);

– M_5 = Strassen(A_{11}, B_{12} - B_{22}, n/2);

– M_6 = Strassen(A_{22}, $B_{21} - B_{11}$, n/2);

– M_7 = Strassen($A_{21} + A_{22}$, B_{11}, n/2);

– $C_{11} = M_1 + M_2 - M_4 + M_6$;

– $C_{12} = M_4 + M_5$;

– $C_{21} = M_6 + M_7$;

– $C_{22=}$ M_2 M_{3+} M_{5-} M7;

– **Retorne** C = $\begin{vmatrix} C_{11} + C_{12} \\ C_{21} + C_{22} \end{vmatrix}$

– **Fim**

3. COMPLEXIDADE DO ALGORITMO

Passaremos a analisar a complexidade do algoritmo apresentado.

Seja T(n) o número de passos necessários para calcular o produto de duas matrizes de dimensão n. A partir do algoritmo de Strassen pode-se concluir que:

$T(n) = 7T(n/2) + 18(n/2)^2$ pois são necessárias 7 chamadas recursivas ao algoritmo com matrizes de tamanho n/2 e 18 operações de soma e subtração entre matrizes de dimensão n/2.

Substituindo os valores de n por n/2, obtém-se:

$$T(n) = 7\left(7T\left(\frac{n}{4}\right) + 18\left(\frac{n}{4}\right)^2\right) + 18\left(\frac{n}{2}\right)^2$$

que é equivalente à expressão a seguir.

$$T(n) = 7^2 T\left(\frac{n}{2^2}\right) + 7x18\left(\frac{n}{2^2}\right)^2 + 18\left(\frac{n}{2}\right)^2 \tag{1}$$

Substituindo sucessivamente até atingir T(1), tem-se:

$$T(n) = 7^k T(1) + 7^{k-1}.18\left(\frac{n}{2^k}\right)^2 + 7^{k-2}.18\left(\frac{n}{2^{k-1}}\right)^2 + ... + 7^0.18\left(\frac{n}{2}\right)^2 \tag{2}$$

onde k= $\log_2 n$

$$T(n) = 7^k + 18n^2 \sum_{j=1}^{k} \frac{7^{j-1}}{2^{2j}} = 7^k + 18n^2 \frac{1}{7} \sum_{j=1}^{k} \left(\frac{7}{4}\right)^j$$

$$T(n) = n^{\log_2 7} + 18n^2 \frac{1}{7}\frac{7}{4} \frac{\left[\left(\frac{7}{4}\right)^k - 1\right]}{\frac{7}{4} - 1}$$

$$T(n) = n^{2.81} + \frac{18}{3} n^2 \left[\left(\frac{7}{4}\right)^k - 1\right] = n^{2.81} + 6n^2 n^{\log_2 1.75} - 6n^2$$

$$T(n) = 7.n^{2.81} - 6n^2 \tag{3}$$

Dessa forma, o algoritmo de Strassen tem complexidade $\Theta(n^{2.81})$, para algum valor $n > n_0$, o que o torna superior ao algoritmo tradicional. A expressão (3) não considera outras operações que não sejam soma e subtração, havendo constantes multiplicativas não reveladas. A expressão (3) considera que a recursão foi aprofundada até obter matrizes de dimensão lxl, mas sabe-se que o algoritmo gasta 7 produtos e 18 somas para calcular uma matriz 2x2 e o algoritmo tradicional gasta 8 multiplicações e 4 somas. Usando esse raciocínio, seria interessante suspender a recursão para alguma constante n_o, usando o algoritmo tradicional para matrizes de dimensões menores que $n_0 x n_0$. O valor dessa constante depende muito da implementação e pode ser determinada experimentalmente. Admite-se que o valor no deve ser n_o mínimo 45[3].

O limite atual para produto de matrizes é $\Theta(n^{2.376})$, mas o algoritmo utilizado não é tão prático comparado com o algoritmo de Strassen.

4. CONCLUSÕES

O algoritmo apresentado pode ser generalizado para matrizes com dimensões arbitrárias, não necessariamente potência de 2. Basta modificar o algoritmo para acrescentar uma linha ou uma coluna de zeros quando o número de linhas ou de colunas for impar, possibilitando-o particionamento da matriz ao meio. Esse problema não existe para matrizes com ambas dimensões menores do que n_0, a constante abaixo da qual usa-se o algoritmo tradicional.

A literatura tem explorado outros esquemas semelhantes ao apresentado, como o esquema de matrizes 3x3 proposto em[2], que requer apenas 23 multiplicações ao invés de 27.

O esquema proposto por Strassen pode ser melhorado com a redução das operações de adição/subtração de 18 para 15 operações. Essa melhoria não é muito grande, mas ajuda a reduzir um pouco o número de operações, com um gasto maior de memória.

Como caráter ilustrativo, um produto entre matrizes densas com dimensões da ordem de 1000x1000, o algoritmo de Strassen é cerca de três vezes mais rápido do que o algoritmo tradicional, o que não pode ser desprezado em aplicações práticas.

REFERÊNCIAS BIBLIOGRAFIAS

1. Gilles Brassard and Paul Bratley, Fundamentals of Algorithmics, Prentice Hall, 1996

2. J.D. Laderman, A non-commutative algorithm for multiplying 3x3 matrices using 23 multiplications, Bull. Am. Math.Soc., 82 (1976) 126-128.

3. Cormen, T.H., et. al., Introduction to Algorithms, McGraw-Hill, 1990.