빠른 3×3 행렬 곱셈법을 자동으로 찾아내기

Automatic Discovery of Fast 3×3 Matrix Multiplication

김진
최적화 및 금융공학 연구실
서울대학교

2011년 1월 6일

단순한 복소수 곱셈

$$(a+bi)(c+di) = ac + adi + bci - bd$$

$$a \times c$$
$$a \times d$$
$$b \times c$$
$$b \times d$$

곱셈 네 번

가우스의 복소수 곱셈법

$$(a+bi)(c+di) = ac - bd + adi + bci$$

$$\begin{aligned} k_1 &= c \times (a+b) \\ k_2 &= a \times (d-c) \\ k_3 &= b \times (c+d) \end{aligned}$$

$$\begin{aligned} \text{실수부} &= k_1 - k_3 \\ \text{허수부} &= k_1 + k_2 \end{aligned}$$

곱셈 세 번

자연수 곱셈

		1	2
×		3	4
		4	8
	3	6	
	4	0	8

곱셈 네 번 (자릿수별로)

카라슈바 알고리즘

$$x = x_1 \cdot 10 + x_0$$
$$y = y_1 \cdot 10 + y_0$$

$$z_2 = x_1 \times y_1$$
$$z_1 = x_1 \times y_0 + x_0 \times y_1$$
$$z_0 = x_0 \times y_0$$

$$xy = (x_1 \cdot 10 + x_0)(y_1 \cdot 10 + y_0) = z_2 \cdot 100 + z_1 \cdot 10 + z_0$$

여전히 곱셈 네 번? 하지만,

$$z_1 = (x_1 + x_0) \times (y_1 + y_0) - z_2 - z_0$$

세 번

일반적인 카라슈바 알고리즘

x와 y가 n자리 B진수이고, $m < n$일 때:

$$x = x_1 \cdot B^m + x_0$$
$$y = y_1 \cdot B^m + y_0$$

$$z_2 = x_1 \times y_1$$
$$z_1 = (x_1 + x_0) \times (y_1 + y_0) - z_2 - z_0$$
$$z_0 = x_0 \times y_0$$

$$xy = z_2 \cdot B^{2m} + z_1 \cdot B^m + z_0$$

z들 계산에 다시 카라슈바 알고리즘을 적용

2×2 행렬 곱셈

$$\begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix} \times \begin{pmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{pmatrix} = \begin{pmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{pmatrix}$$

$$\begin{aligned} C_{11} &= A_{11} \times B_{11} + A_{12} \times B_{21} \\ C_{12} &= A_{11} \times B_{12} + A_{12} \times B_{22} \\ C_{21} &= A_{21} \times B_{11} + A_{22} \times B_{21} \\ C_{22} &= A_{21} \times B_{12} + A_{22} \times B_{22} \end{aligned}$$

곱셈 여덟 번

스트라센 알고리즘

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

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

곱셈 일곱 번

래더만 알고리즘

3×3 행렬 곱셈

$$
\begin{aligned}
M_1 &= (A_{11} + A_{12} + A_{13} - A_{21} - A_{22} - A_{32} - A_{33}) \times B_{22} \\
M_2 &= (A_{11} - A_{21}) \times (-B_{12} + B_{22}) \\
M_3 &= A_{22} \times (B_{11} + B_{12} + B_{21} - B_{22} - B_{23} - B_{31} + B_{33}) \\
M_4 &= (-A_{11} + A_{21} + A_{22}) \times (B_{11} - B_{12} + B_{22}) \\
M_5 &= (A_{21} + A_{22}) \times (-B_{11} + B_{12}) \\
M_6 &= A_{11} \times B_{11} \\
M_7 &= (-A_{11} + A_{31} + A_{32}) \times (B_{11} - B_{13} + B_{23}) \\
M_8 &= (-A_{11} + A_{31}) \times (B_{13} - B_{23}) \\
M_9 &= (A_{31} + A_{32}) \times (-B_{11} + B_{13}) \\
M_{10} &= (A_{11} + A_{12} + A_{13} - A_{22} - A_{23} - A_{31} - A_{32}) \times B_{23} \\
M_{11} &= A_{32} \times (-B_{11} + B_{13} + B_{21} - B_{22} - B_{23} - B_{31} + B_{32}) \\
M_{12} &= (-A_{13} + A_{32} + A_{33}) \times (B_{22} + B_{31} - B_{32}) \\
M_{13} &= (A_{13} - A_{33}) \times (B_{22} - B_{32}) \\
M_{14} &= A_{13} \times B_{31} \\
M_{15} &= (A_{32} + A_{33}) \times (-B_{31} + B_{32}) \\
M16 &= (-A_{13} + A_{22} + A_{32}) \times (B_{23} + B_{31} - B_{33}) \\
M_{17} &= (A_{13} - A_{23}) \times (B_{23} - B_{33}) \\
M_{18} &= (A_{22} + A_{23}) \times (-B_{31} + B_{33}) \\
M_{19} &= A_{12} \times B_{21} \\
M_{20} &= A_{23} \times B_{32} \\
M_{21} &= A_{21} \times B_{13} \\
M_{22} &= A_{31} \times B_{12} \\
M_{23} &= A_{33} \times B_{33}
\end{aligned}
$$

$$
\begin{aligned}
C_{11} &= M_6 + M_{14} + M_{19} \\
C_{12} &= M_1 + M_4 + M_5 + M_6 + M_{12} \\
C_{13} &= M_6 - M_7 - M_9 + M_{10} - M_{14} \\
C_{21} &= M_2 + M_3 + M_4 - M_6 + M_{14} \\
C_{22} &= M_2 + M_4 + M_5 + M_6 + M_{20} \\
C_{23} &= M_{14} + M_{16} + M_{17} + M_{18} + M \\
C_{31} &= M_6 + M_7 + M_8 - M_{11} + M_{12} \\
C_{32} &= M_{12} + M_{13} + M_{14} + M_{15} + M \\
C_{33} &= M_6 + M_7 + M_8 + M_9 - M_{23}
\end{aligned}
$$

곱셈 27번 $\rightarrow$ 23번

행렬 곱셈 방정식

$$M_t = \left(\sum \alpha_{ij}^t A_{ij}\right)\left(\sum \beta_{kl}^t B_{kl}\right) \tag{1}$$

$$C_{mn} = \sum_{t=1}^{T} \gamma_{mn}^t M_t \tag{2}$$

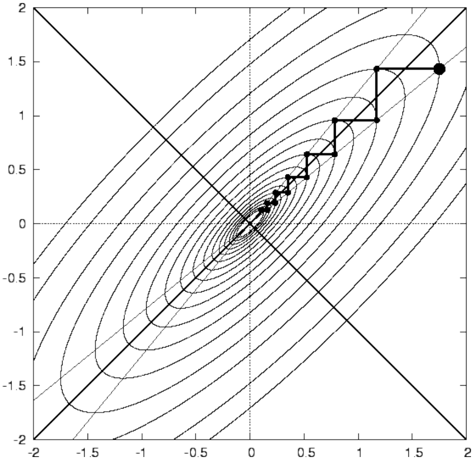
$$C_{mn} = \sum_{t,i,j,k,l} \gamma_{mn}^t \alpha_{ij}^t A_{ij} \beta_{kl}^t B_{kl} \tag{3}$$

$$\sum_t \gamma_{mn}^t \alpha_{ij}^t \beta_{kl}^t = \begin{cases} 1 & \text{if } m = i,\ j = k,\ l = n \\ 0 & \text{o.w.} \end{cases} \tag{4}$$

$$\sum_t \gamma_{mn}^t \alpha_{ij}^t \beta_{kl}^t = \delta_{mi}\delta_{jk}\delta_{ln} \tag{5}$$

$$\alpha, \beta, \gamma \in \{-1, 0, 1\}$$

최급강하법



찾아낸 곱셈 알고리즘

Group 1 (Strassen's Solution)	Group 2	Group 3
$P_1 = A_1(B_3 - B_4)$	$P_1 = A_1(B_3 - B_4)$	$P_1 = A_1(B_3 - B_4)$
$P_2 = (A_1 + A_2)B_4$	$P_2 = (A_1 + A_2)B_4$	$P_2 = (A_1 + A_2)B_4$
$P_3 = (A_3 + A_4)B_1$	$P_3 = A_4(B_2 + B_4)$	$P_3 = (A_3 - A_4)B_2$
$P_4 = A_4(-B_1 + B_2)$	$P_4 = A_3(B_1 + B_3)$	$P_4 = A_3(B_1 + B_2)$
$P_5 = (A_1 + A_4)(B_1 + B_4)$	$P_5 = (A_2 + A_4)(B_1 - B_2)$	$P_5 = (A_1 + A_2 + A_3 + A_4)(B_1 + B_2 + B_3 + B_4)$
$P_6 = (A_2 - A_4)(B_2 + B_4)$	$P_6 = (A_1 + A_2 + A_4)(B_1 + B_4)$	$P_6 = (A_1 + A_2 - A_3 + A_4)(B_1 + B_2 + B_3 - B_4)$
$P_7 = (-A_1 + A_3)(B_1 + B_3)$	$P_7 = (A_1 + A_2 + A_3 + A_4)B_1$	$P_7 = (A_1 - A_2 + A_3 - A_4)(B_1 - B_2 + B_3 - B_4)$
$C_1 = -P_2 + P_4 + P_5 + P_6$	$C_1 = -P_2 - P_3 - P_5 + P_6$	$C_1 = -P_1 - P_3 + 0.5P_6 + 0.5P_7$
$C_2 = P_1 + P_2$	$C_2 = P_1 + P_2$	$C_2 = P_1 + P_2$
$C_3 = P_3 + P_4$	$C_3 = P_2 + P_3 - P_6 + P_7$	$C_3 = -P_3 + P_4$
$C_4 = P_1 - P_3 + P_5 + P_7$	$C_4 = -P_2 + P_4 + P_6 - P_7$	$C_4 = -P_2 - P_4 + 0.5P_5 - 0.5P_6$
Group 4	Group 5	Group 6
$P_1 = A_4(-B_1 + B_2 - B_3 + B_4)$	$P_1 = (A_1 + A_2)(B_1 + B_3)$	$P_1 = (A_1 + A_2)(B_3 + B_4)$
$P_2 = A_1(B_1 - B_2 - B_3 + B_4)$	$P_2 = (A_1 + A_2 - A_3 + A_4)(B_2 - B_3)$	$P_2 = (A_1 - A_2)(B_3 - B_4)$
$P_3 = (A_1 + A_4)(B_1 - B_2 + B_3 + B_4)$	$P_3 = (-A_3 + A_4)(B_1 - B_3)$	$P_3 = (A_2 - A_4)(B_1 - B_2 - B_3 + B_4)$
$P_4 = (A_1 - A_3)B_3$	$P_4 = (A_1 + A_2 - A_3 - A_4)(B_1 + B_2)$	$P_4 = (A_2 + A_4)(B_1 + B_2 + B_3 + B_4)$
$P_5 = (A_3 + A_4)(B_1 + B_3)$	$P_5 = (A_1 - A_2 - A_3 + A_4)(B_1 - B_2)$	$P_5 = (A_1 - A_4)(B_1 + B_4)$
$P_6 = (A_1 + A_2)(B_2 - B_4)$	$P_6 = A_2(B_1 - B_2 + B_3 - B_4)$	$P_6 = (A_1 + A_2 - A_3 - A_4)(B_1 - B_3)$
$P_7 = (A_2 - A_4)B_4$	$P_7 = A_4(B_1 + B_2 - B_3 - B_4)$	$P_7 = (A_1 - A_2 + A_3 - A_4)(B_1 + B_3)$
$C_1 = 0.5P_1 + 0.5P_2 + 0.5P_3 + P_6 + P_7$	$C_1 = 0.5P_1 + 0.5P_2 - 0.5P_3 + 0.5P_5$	$C_1 = -0.5P_1 + 0.5P_2 - 0.5P_3 + 0.5P_4 + P_5$
$C_2 = 0.5P_1 - 0.5P_2 + 0.5P_3 + P_7$	$C_2 = 0.5P_1 - 0.5P_2 + 0.5P_3 - 0.5P_5 - P_6$	$C_2 = 0.5P_1 + 0.5P_2$
$C_3 = 0.5P_1 + 0.5P_2 - 0.5P_3 + P_4 + P_5$	$C_3 = 0.5P_1 + 0.5P_2 - 0.5P_3 - 0.5P_4$	$C_3 = -0.5P_1 - 0.5P_2 + 0.5P_3 + 0.5P_4 - 0.5P_6 + 0.5P_7$
$C_4 = 0.5P_1 - 0.5P_2 + 0.5P_3 - P_4$	$C_4 = 0.5P_1 + 0.5P_2 + 0.5P_3 - 0.5P_4 - P_7$	$C_4 = 0.5P_1 - 0.5P_2 - P_5 + 0.5P_6 + 0.5P_7$
Group 7	Group 8	Group 9 (Winograd's Solution)
$P_1 = A_1B_1$	$P_1 = A_1B_1$	$P_1 = A_1B_1$
$P_2 = A_2B_2$	$P_2 = A_2B_2$	$P_2 = A_2B_2$
$P_3 = A_3(B_1 + B_2 + B_3 + B_4)$	$P_3 = A_3(B_3 + B_4)$	$P_3 = A_3(B_1 + B_2 + B_3 + B_4)$
$P_4 = (A_2 + A_4)(B_1 + B_2 - B_3 - B_4)$	$P_4 = (A_2 + A_4)(B_1 + B_2 + B_3 + B_4)$	$P_4 = (A_2 + A_4)(B_3 + B_4)$
$P_5 = (A_3 - A_4)(B_2 + B_4)$	$P_5 = (A_3 - A_4)B_4$	$P_5 = (A_3 - A_4)(B_2 + B_4)$
$P_6 = (A_2 - A_3 + A_4)(B_1 - B_2 - B_3 - B_4)$	$P_6 = (A_2 - A_3 + A_4)(B_1 + B_3 + B_4)$	$P_6 = (A_2 - A_3 + A_4)(B_2 - B_2 - B_3 - B_4)$
$P_7 = (A_1 - A_2 + A_3 - A_4)(B_1 - B_4)$	$P_7 = (A_1 - A_2 + A_3 - A_4)(B_1 + B_3)$	$P_7 = (A_1 - A_2 + A_3 - A_4)B_3$
$C_1 = P_1 + P_2$	$C_1 = P_1 + P_2$	$C_1 = P_1 + P_2$
$C_2 = P_1 - P_2 + P_3 - P_6 - P_7$	$C_2 = -P_1 + P_5 + P_6 + P_7$	$C_2 = -P_2 + P_5 + P_6 + P_7$
$C_3 = -P_2 + 0.5P_3 + 0.5P_4 + 0.5P_6$	$C_3 = -P_2 - P_3 + P_4 - P_6$	$C_3 = -P_2 + P_3 - P_4 + P_6$
$C_4 = P_2 + 0.5P_3 - 0.5P_4 - P_5 + 0.5P_6$	$C_4 = P_3 - P_5$	$C_4 = P_2 + P_4 - P_5 - P_6$

유전 알고리즘과 결합

- 2×2 곱셈법은 재현해서 진화 연산 학술지에 게재
- 3×3 곱셈 23개를 사용하는 방법은 다른 연구자들이 찾아냄
- 유전 알고리즘과 최급강하법의 결합으로 22개에 도전

래더만 알고리즘 응용

- 3×3 행렬 곱셈은 2×2 행렬 곱셈 여러 개로 이루어지고,
- 2×2 쌍마다 곱셈 수를 하나씩 줄인다 (스트라센 알고리즘)
- 2×2 네 개로 이루어진 래더만 알고리즘은 곱셈 23개
- 2×2 행렬 곱셈 다섯 개를 조합할 수 있다면?

$$\begin{pmatrix} \odot & \odot & \cdot \\ \odot & \odot & \cdot \\ \cdot & \cdot & \cdot \end{pmatrix} \times \begin{pmatrix} \odot & \odot & \cdot \\ \odot & \odot & \cdot \\ \cdot & \cdot & \cdot \end{pmatrix} \quad \begin{pmatrix} \odot & \odot & \cdot \\ \cdot & \cdot & \cdot \\ \odot & \odot & \cdot \end{pmatrix} \times \begin{pmatrix} \odot & \cdot & \odot \\ \odot & \cdot & \odot \\ \cdot & \cdot & \cdot \end{pmatrix}$$

$$\begin{pmatrix} \cdot & \odot & \odot \\ \cdot & \cdot & \cdot \\ \cdot & \odot & \odot \end{pmatrix} \times \begin{pmatrix} \cdot & \cdot & \cdot \\ \odot & \odot & \cdot \\ \odot & \odot & \cdot \end{pmatrix} \quad \begin{pmatrix} \cdot & \odot & \odot \\ \cdot & \odot & \odot \\ \cdot & \cdot & \cdot \end{pmatrix} \times \begin{pmatrix} \cdot & \cdot & \cdot \\ \odot & \cdot & \odot \\ \odot & \cdot & \odot \end{pmatrix}$$