Simulation and Scientific Computing

(Simulation und Wissenschaftliches Rechnen - SiWiR)

Winter Term 2015/16

Florian Schornbaum and Christian Kuschel
Chair for System Simulation

FRIEDRICH-ALEXANDER
UNIVERSITÄT
ERLANGEN-NÜRNBERG
TECHNISCHE FAKULTÄT

Assignment 1: Performance Optimization

October 20, 2015 – November 9, 2015

FRIEDRICH-ALEXANDER
UNIVERSITÄT
ERLANGEN-NÜRNBERG
TECHNISCHE FAKULTÄT

Strassen Algorithm

LSS

FAU FRIEDRICH-ALEXANDER UNIVERSITÄT ERLANGEN-NÜRNBERG

TECHNISCHE FAKULTÄT

Strassen Algorithm

Naïve matrix-matrix multiplication:

$$C = A \cdot B \quad ; \quad A = \begin{pmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{nn} \end{pmatrix} \in \mathbb{R}^{n \times n} \quad ; \quad B, C \in \mathbb{R}^{n \times n}$$

$$\begin{array}{cc} & \begin{pmatrix} b_{11} & \cdots & b_{1n} \\ \vdots & \ddots & \vdots \\ b_{n1} & \cdots & b_{nn} \end{pmatrix} \\ \begin{pmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{nn} \end{pmatrix} & \begin{pmatrix} c_{11} & \cdots & c_{1n} \\ \vdots & \ddots & \vdots \\ c_{n1} & \cdots & c_{nn} \end{pmatrix} \end{array}$$

number of multiplications:
$n^2 \cdot n$

number of additions:
$n^2(n-1)$

total number of flops (floating point operations): $2n^3 - n^2 \rightarrow O(n^3)$

Strassen Algorithm

Naïve, recursive scheme:

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

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

$T(n) :=$ total number of flops for $n \times n$ matrices

$$T(n) = 8 \cdot T\left(\frac{n}{2}\right) + 4\left(\frac{n}{2}\right)^2 = 8 \cdot T\left(\frac{n}{2}\right) + n^2$$

Strassen Algorithm

Naïve, recursive scheme:

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

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

$T(n) :=$ total number of flops for $n \times n$ matrices

$$T(n) = 8 \cdot T\left(\frac{n}{2}\right) + 4\left(\frac{n}{2}\right)^2 = 8 \cdot T\left(\frac{n}{2}\right) + n^2$$

Strassen Algorithm

Master theorem (Hauptsatz der Laufzeitfunktionen):

$$T(n) = a \cdot T\left(\frac{n}{b}\right) + f(n) \text{ where } a \geq 1, b > 1$$

$$\text{If } f(n) = O(n^c) \text{ where } c < log_b a \text{ then } T(n) = O(n^{log_b a})$$

Naïve, recursive scheme:

$$T(n) = 8 \cdot T\left(\frac{n}{2}\right) + 4\left(\frac{n}{2}\right)^2 = 8 \cdot T\left(\frac{n}{2}\right) + n^2$$

$$a = 8 \geq 1, b = 2 > 1, f(n) = n^2 = O(n^2), c = 2 < log_2 8 = 3$$

$$\rightarrow \; T(n) = O\left(n^{log_2 8}\right) = O(n^3)$$

Strassen Algorithm

The Strassen algorithm (a different recursive scheme):

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

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

$$T(n) = 7 \cdot T\left(\tfrac{n}{2}\right) + 18\left(\tfrac{n}{2}\right)^2 = 7 \cdot T\left(\tfrac{n}{2}\right) + \tfrac{9}{2}n^2$$

Strassen Algorithm

The Strassen algorithm (a different recursive scheme):

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

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

$$T(n) = 7 \cdot T\left(\frac{n}{2}\right) + 18\left(\frac{n}{2}\right)^2 = 7 \cdot T\left(\frac{n}{2}\right) + \frac{9}{2}n^2$$

Strassen Algorithm

Master theorem (Hauptsatz der Laufzeitfunktionen):

$$T(n) = a \cdot T\left(\frac{n}{b}\right) + f(n) \;\text{ where }\; a \geq 1, b > 1$$

$$\text{If }\; f(n) = O(n^c) \;\text{ where }\; c < \; log_b a \;\text{ then }\; T(n) = O(n^{log_b a})$$

The Strassen algorithm:

$$T(n) = 7 \cdot T\left(\frac{n}{2}\right) + 18\left(\frac{n}{2}\right)^2 = 8 \cdot T\left(\frac{n}{2}\right) + \frac{9}{2}n^2$$

$$a = 7 \geq 1, b = 2 > 1$$

$$f(n) = \frac{9}{2}n^2 = O(n^2), c = 2 < log_2 7 \approx 2.807355$$

$$\rightarrow \;\; T(n) = O\left(n^{log_2 7}\right) = O(n^{2.807355}) < O(n^3)!$$

Strassen Algorithm

The Strassen algorithm (a different recursive scheme):

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

$$\begin{aligned} M_1 &= (A_{11} + A_{22}) \cdot (B_{11} + B_{22}) \\ M_2 &= (A_{21} + A_{22}) \cdot B_{11} \\ M_3 &= A_{11} \cdot (B_{12} - B_{22}) \\ M_4 &= A_{22} \cdot (B_{21} - B_{11}) \\ &\dots \end{aligned}$$

Proof of correctness:

$$\begin{aligned} C_{21} = M_2 + M_4 &= (A_{21} + A_{22}) \cdot B_{11} + A_{22} \cdot (B_{21} - B_{11}) \\ &= A_{21}B_{11} + A_{22}B_{11} + A_{22}B_{21} - A_{22}B_{11} \\ &= A_{21}B_{11} + A_{22}B_{21} \end{aligned}$$

Strassen Algorithm

The Strassen algorithm (a different recursive scheme):

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

$$\begin{aligned} M_1 &= (A_{11} + A_{22}) \cdot (B_{11} + B_{22}) \\ M_2 &= (A_{21} + A_{22}) \cdot B_{11} \\ M_3 &= A_{11} \cdot (B_{12} - B_{22}) \\ M_4 &= A_{22} \cdot (B_{21} - B_{11}) \\ &\ldots \end{aligned}$$

Proof of correctness:

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

Strassen Algorithm

Total number of flops for the Strassen algorithm?

Recursive form: $T(n) = 7 \cdot T\left(\frac{n}{2}\right) + 18\left(\frac{n}{2}\right)^2$

Hypothesis: $T(n) = 7n^{log_2 7} - 6n^2$

Proof by induction:

$$n = 2^k \to k = log_2 n$$

Induction hypothesis:

$$T(k) = 7 \cdot 2^{k log_2 7} - 6 \cdot 2^{k2} = 7 \cdot 2^{log_2 7^k} - 6 \cdot 2^{2k}$$
$$= 7 \cdot 7^k - 6 \cdot 2^{2k} = 7^{k+1} - 6 \cdot 4^k$$

Base case ($k = 0 \to n = 1$ / multiplication of two scalar values):

$$T(0) = 7 - 6 = 1 \text{ (one multiplication)}$$

q.e.d.

Strassen Algorithm

Total number of flops for the Strassen algorithm?

Recursive form: $T(n) = 7 \cdot T\left(\frac{n}{2}\right) + 18\left(\frac{n}{2}\right)^2$

$$\Rightarrow \; T(k) = 7 \cdot T(k-1) + 18 \cdot 4^{k-1}$$

Induction hypothesis:

$$T(k) = 7 \cdot 2^{k log_2 7} - 6 \cdot 2^{k2} = 7 \cdot 2^{log_2 7^k} - 6 \cdot 2^{2k}$$
$$= 7 \cdot 7^k - 6 \cdot 2^{2k} = 7^{k+1} - 6 \cdot 4^k$$

Inductive step ($k \rightarrow k+1$):

$$T(k+1) = 7 \cdot T(k) + 18\left(\frac{2^{k+1}}{2}\right)^2 = 7 \cdot T(k) + 18\frac{2^{2(k+1)}}{2^2}$$
$$= 7 \cdot T(k) + 18 \cdot 4^k = 7\left(7^{k+1} - 6 \cdot 4^k\right) + 18 \cdot$$
$$= 7 \cdot 7^{k+1} - 7 \cdot 6 \cdot 4^k + 18 \cdot 4^k = 7^{(k+1)+1} - 24 \cdot 4^k$$
$$= 7^{(k+1)+1} - 6 \cdot 4 \cdot 4^k = 7^{(k+1)+1} - 6 \cdot 4^{k+1}$$

q.e.d.

Strassen Algorithm

Total number of flops for the Strassen algorithm?

Since $T(k) = 7^{k+1} - 6 \cdot 4^k$ holds for all k (as proven by induction), $T(n) = 7n^{log_2 7} - 6n^2$ is the total number of flops for the Strassen algorithm when multiplying two matrices of size $n \times n$.

When (starting with which n) does the Strassen algorithm need fewer flops than the naïve matrix-matrix multiplication?

$$7n^{log_2 7} - 6n^2 = 2n^3 - n^2$$
$$\Rightarrow 2n^3 - 7n^{log_2 7} + 5n^2 = 0$$
$$\Rightarrow n = 0, n = 1, n \approx 654.031$$

$n = 2^9 \;=\; 512$: $\text{flops}_{\text{naïve}} < \text{flops}_{\text{Strassen}}$

$n = 2^{10} = 1024$: $\text{flops}_{\text{naïve}} > \text{flops}_{\text{Strassen}}$

The End

Questions ?

FRIEDRICH-ALEXANDER
UNIVERSITÄT
ERLANGEN-NÜRNBERG
TECHNISCHE FAKULTÄT