ביה״ס להנדסה ולמדעי המחשב ע״ש רחל וסלים בנין
The Rachel and Selim Benin School of Computer Science and Engineering

Matrix Multiplication, A Little Faster

Elaye Karstadt and Oded Schwartz

האוניברסיטה העברית בירושלים
THE HEBREW UNIVERSITY OF JERUSALEM

Goals

Improve the practical performance of fast matrix multiplication algorithms.

1. We reduce the number of arithmetic operations by a constant factor
2. We reduce the number of data transfers within memory hierarchy by a constant factor

Introduction

Strassen's algorithm (1969), was the first sub-cubic matrix multiplication algorithm. Winograd (1971) improved the leading coefficient of its complexity from 7 to 6. Can we do better?

Theorem (Probert 1976). *Any Strassen-like algorithm with 2x2 base case and 7 multiplications requires at least 15 additions.*

Strassen-Winograd's algorithm was believed to be optimal due to this bound.

Theorem (Karstadt & Schwartz 2017). *There is a Strassen-like algorithm with 2x2 base case and 7 multiplications requires at least 12 additions.*

Our theorem seems to implicitly contradict Probert's lower bound. However, this bound assumes that the input and output are represented in the standard basis. We extend Probert's lower pound to account for alternative bases.

Theorem (Karstadt & Schwartz 2017). *Irrespective of input/output bases, a Strassen-like algorithm with 2x2 base case and 7 multiplications requires at least 12 additions.*

Our generalization of Probert's lower bound shows our algorithm to be optimal for matrix multiplication with 2x2 base case and with 7 multiplications.

Encoding/Decoding matrices

Any bi-linear algorithm which uses t multiplications can be described by encoding/decoding matrices $\langle U, V, W\rangle$:

$$U \in R^{t\times n\cdot m},\ V \in R^{t\times m\cdot k},\ W \in R^{t\times n\cdot k}$$

Such that $\forall A \in R^{n\cdot m},\ B \in R^{m\cdot k}$

$$ALG\left(A, B\right) = W^T\left(\left(U\cdot A\right)\odot\left(V\cdot B\right)\right)$$

where $\cdot$ is matrix multiplication and $\odot$ is element-wise vector product (Hadamard product).

Our Method

Strassen-like $\langle n, m, k; t\rangle$-algorithms are block-recursive algorithms. Defined by $n\times m$, $m\times k$, and $n\times k$ base case for the linear part, and t multiplications.
Bodrato [2] used a method of intermediate representation of 2×2 matrices for repeated squaring and for chain matrix multiplication computations. We extend this method to alternative basis Strassen-like matrix multiplication.

Alternative basis Strassen-like $\langle n, m, k; t\rangle_{\phi,\psi,\upsilon}$-algorithms take input $\phi\left(A\right)$, $\psi\left(B\right)$ and output $\upsilon\left(A\cdot B\right)$.

Lemma (Karstadt & Schwartz 2017). *Let R be a ring, and let ϕ, ψ, υ be automorphisms of $R^{n\cdot m}$, $R^{m\cdot k}$, $R^{n\cdot k}$ (respectively). $\langle U, V, W\rangle$ are encoding/decoding matrices of an $\langle n, m, k; t\rangle_{\phi,\psi,\upsilon}$-algorithm if and only if $\langle U\phi, V\psi, W\upsilon^{-T}\rangle$ are encoding/decoding matrices of an $\langle n, m, k; t\rangle$-algorithm*

We compute over alternative bases where:

- Transformation between the standard and our alternative basis can be done in $O\left(n^2\log n\right)$ time, which is asymptotically negligible.
- A Strassen-like algorithm in our alternative basis uses fewer additions/subtractions.

Optimal $\langle 2,2,2;7\rangle_{\phi,\psi,\upsilon}$-algorithm

$$U_{opt} = \begin{pmatrix} 0 & 0 & 0 & 1\\ 0 & 0 & 1 & 0\\ 0 & 1 & 0 & 0\\ 1 & 0 & 0 & 0\\ 0 & 1 & -1 & 0\\ -1 & 1 & 0 & 0\\ 0 & -1 & 0 & 1 \end{pmatrix} V_{opt} = \begin{pmatrix} 0 & 0 & 0 & 1\\ 0 & 0 & 1 & 0\\ 0 & 1 & 0 & 0\\ 1 & 0 & 0 & 0\\ 0 & -1 & 0 & 1\\ 0 & 1 & -1 & 0\\ -1 & 1 & 0 & 0 \end{pmatrix} W_{opt} = \begin{pmatrix} 0 & 0 & 0 & 1\\ 0 & 0 & 1 & 0\\ 0 & 1 & 0 & 0\\ 1 & 0 & 0 & 0\\ 1 & 1 & 0 & 0\\ 0 & -1 & 0 & -1\\ 0 & 1 & 1 & 0 \end{pmatrix} \quad \{\phi,\psi\}_{opt} = \begin{pmatrix} 1 & 0 & 0 & 0\\ 0 & 1 & -1 & 1\\ 0 & 0 & -1 & 1\\ 0 & 1 & 0 & 1 \end{pmatrix} \upsilon_{opt}^{-1} = \begin{pmatrix} 1 & 0 & 0 & 0\\ 0 & 1 & -1 & 0\\ 0 & -1 & 0 & 1\\ 0 & -1 & 1 & 1 \end{pmatrix}$$

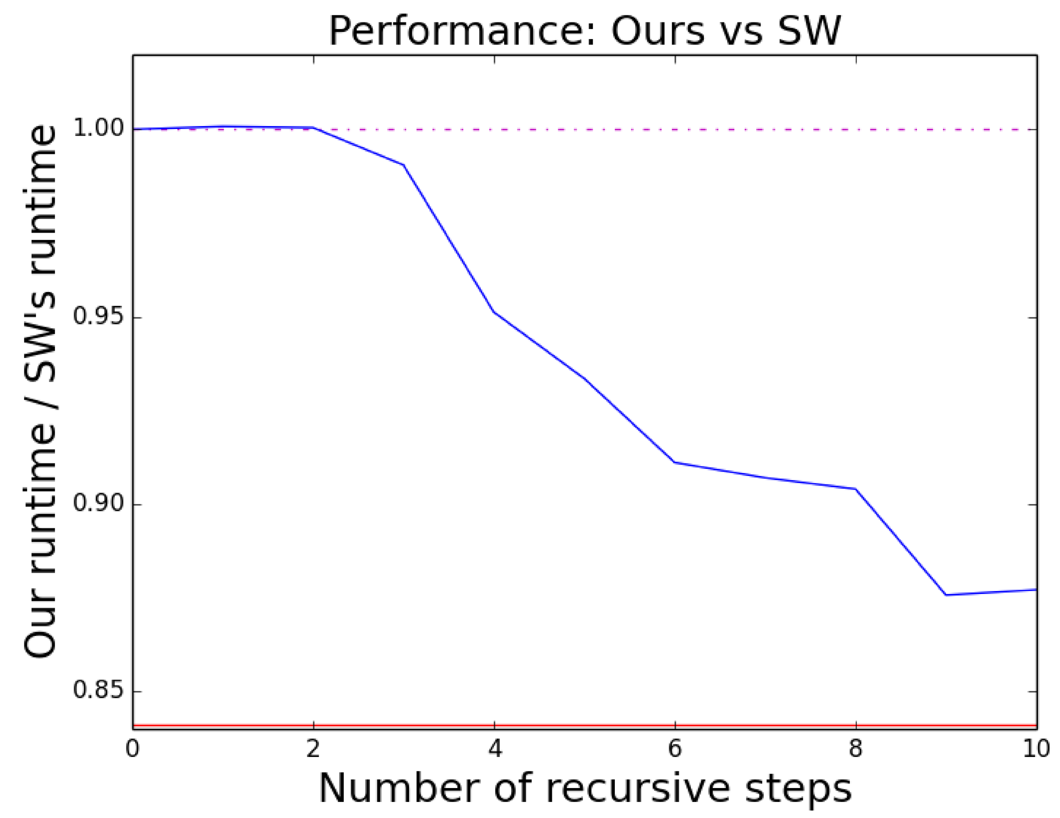
Algorithm 1: Encoding/Decoding and basis transformation matrices for our $\langle 2,2,2;7\rangle_{\phi,\psi,\upsilon} - algorithm$

Comparing our $\langle 2,2,2;7\rangle_{\phi,\psi,\upsilon}$-algorithm to Strassen-Winograd's

Algorithm	Arithmetic Complexity	I/O-Complexity (with a cache of size M)
Strassen [6]	$7n^{\log_2 7} - 6n^2$	$6\cdot\left(\frac{\sqrt{3}\cdot n}{\sqrt{M}}\right)^{\log_2 7}\cdot M - 18n^2 + 3M$
Strassen-Winograd [8]	$6n^{\log_2 7} - 5n^2$	$5\cdot\left(\frac{\sqrt{3}\cdot n}{\sqrt{M}}\right)^{\log_2 7}\cdot M - 15n^2 + 3M$
Ours	$5n^{\log_2 7} - 4n^2 + 3n^2\log_2 n$	$4\cdot\left(\frac{\sqrt{3}\cdot n}{\sqrt{M}}\right)^{\log_2 7}\cdot M - 12n^2 + 3n^2\cdot\log_2\left(\sqrt{2}\cdot\frac{n}{\sqrt{M}}\right) + 5M$

Table 2: Complexity of $\langle 2,2,2;7\rangle$-algorithms

Due to its lower leading coefficient of 5 instead of 6, our $\langle 2,2,2;7\rangle_{\phi,\psi,\upsilon}$-algorithm asymptotically performs 16.6% less arithmetic operations than Strassen-Winograd's. Preliminary benchmark results indicate that our algorithm achieves an improvement close to theory even on modestly sized input ($N = 32768$) with few cores ($P = 6$), and outperforms Strassen-Winograd's algorithm.
We expect the 20% improvement in I/O-complexity to play a greater role on larger inputs with more cores.



Alternative Basis Strassen

Given an $\langle n_0, m_0, k_0; t\rangle_{\phi,\psi,\upsilon}$-algorithm (RBA), and its corresponding $\phi \in GL_{n_0\cdot m_0}\left(R\right)$, $\psi \in GL_{m_0\cdot k_0}\left(R\right)$, $\upsilon \in GL_{n_0\cdot k_0}\left(R\right)$

Input: $A \in R^{n\times m}$, $B \in R^{m\times k}$
Output: $C \in R^{n\times k}$ such that $C = A\cdot B$

1: **function** $ABS(A, B)$
2: $\quad\tilde{A} = \phi(A)$ $\quad\triangleright O\left(nm\cdot\log_{n_0\cdot m_0}(nm)\right)$
3: $\quad\tilde{B} = \psi(B)$ $\quad\triangleright O\left(mk\cdot\log_{m_0\cdot k_0}(mk)\right)$
4: $\quad\tilde{C} = RBA(\tilde{A}, \tilde{B})$
5: $\quad C = \upsilon^{-1}(\tilde{C})$ $\quad\triangleright O\left(nk\cdot\log_{n_0\cdot k_0}(nk)\right)$
6: $\quad$**return** C

Basis transformations are block-recursive, i.e., given ϕ_1 we define:

$$\left(\psi_{k+1}\left(A\right)\right)_{i,j} = \psi_k\left(\psi_1\left(A\right)\right)_{i,j}$$

Further Applications

Algorithm	Prev. Coefficient	New Coefficient	Saves
$\langle 3,2,3;15\rangle$[1]	15.06	7.94	47.2%
$\langle 2,3,4;20\rangle$[1]	9.96	7.46	25.1%
$\langle 2,2,2;7\rangle$[8]	6	5	16.6%
$\langle 6,3,3;40\rangle$[5]	55.63	9.39	83.1%

Table 1: A Sample of Alternative Basis Algorithms

Finding optimal bases for Strassen-like algorithms reduces to the Matrix Sparsification problem, which is NP-Hard [3]. However, our need is for fixed base case sizes.
For algorithms with a base case larger than Strassen-Winograd's, the search space gets quite big. We utilized computer aided search and found several alternative basis variants of known Strassen-like algorithms.

Contact Information

Mail $\{elayeek, odedsc\}@cs.huji.ac.il$
Web $www.cs.huji.ac.il/\sim\{elayeek, odedsc\}$

References

[1] A. R. Benson and G. Ballard. A framework for practical parallel fast matrix multiplication. *ACM SIGPLAN Notices*, 50(8):42–53, 2015.

[2] M. Bodrato. A Strassen-like matrix multiplication suited for squaring and higher power computation. In *Proceedings of the 2010 International Symposium on Symbolic and Algebraic Computation*, pages 273–280. ACM, 2010.

[3] L. Gottlieb and T. Neylon. Matrix sparsification and the sparse null space problem. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, pages 205–218. Springer, 2010.

[4] R. L. Probert. On the additive complexity of matrix multiplication. *SIAM Journal on Computing*, 5(2):187–203, 1976.

[5] A. Smirnov. The bilinear complexity and practical algorithms for matrix multiplication. *Computational Mathematics and Mathematical Physics*, 53(12):1781–1795, 2013.

[6] V. Strassen. Gaussian elimination is not optimal. *Numerische Mathematik*, 13(4):354–356, 1969.

[7] **E. Karstadt and O. Schwartz. Matrix Multiplication, a Little Faster. Submitted to SPAA '17.**

[8] S. Winograd. On multiplication of 2×2 matrices. *Linear algebra and its applications*, 4(4):381–388, 1971.

Acknowledgments

Research is supported by grants 1878/14, and 1901/14 from the Israel Science Foundation (founded by the Israel Academy of Sciences and Humanities) and grant 3-10891 from the Ministry of Science and Technology, Israel. Research is also supported by the Einstein Foundation and the Minerva Foundation. This work was supported by the Peta- Cloud industry-academia consortium. This research was supported by a grant from the United States-Israel Bi-national Science Foundation (BSF), Jerusalem, Israel. This work was supported by the HUJI Cyber Security Research Center in conjunction with the Israel National Cyber Bureau in the Prime Minister's Office. We acknowledge PRACE for awarding us access to Hazel Hen at GCS@HLRS, Germany.