Exact Strassen-Type Solutions for 2 x 2 Matrix Multiplication from Neural Network Learning

Christian Löffeld[*]

December 10, 2018

Abstract

We implemented a neural network aimed specifically at learning fast matrix multiplication. For 2x2 matrices, the network learned the original Strassen Algorithm [1] with 7 multiplications and 18 additions. Additionally, it learned an exact algorithm with 7 multiplications and 16 additions, merely one more addition than the arithmetically optimal Strassen-Winograd algorithm [2].

Introduction

Since Volker Strassen in 1969 discovered a subcubic algorithm for matrix multiplication [1], there have been intense research efforts to devise algorithms that reduce the computational complexity of matrix multiplication. Winograd [2] found the arithmetically optimal algorithm for 2x2 matrix multiplication in 1971. Coppersmith and Winograd [3] developed an algorithm for general nxn matrix multiplication with a complexity of $O(n^{2.3728639})$. This complexity was consecutively improved upon by various researchers [4, 5, 6].

More recently, neural networks have been used to investigate the complexity space of matrix multiplication. Elser [7] used a particular neural network architecture and learning methodology in order to learn nxn matrix multiplication for $n = 2, 3$. Tschannen *et al.* [8] used a neural network approach to find approximate solutions to matrix multiplication and employed these approximate algorithms successfully for image recognition applications.

Result

We implemented a neural network with $2n^2$ input nodes, one hidden layer containing k nodes representing the k allowed multiplications in a nxn matrix multiplication, and n^2 output nodes. In order to train the network, we utilized stochastic gradient descent and regularization, together with the element-wise objective function

$$e_{i,j} = (M_{i,j} - C_{i,j})^2 \tag{1}$$

where M is the learned matrix product, and $C = AB$ is defined according to the rules of 2x2 matrix multiplication as described below.

In the case, where $n = 2$ and $k = 7$, we found two multiplicatively optimal algorithms. Firstly, the very well known algorithm discovered by Strassen in 1969 [1], and secondly an algorithm that requires two fewer additions than Strassen's original solution. Here, we present the second algorithm learned by the neural network, specifically the one requiring 7 multiplications and 16 additions.

[*]Email: christian.loeffeld@gmail.com

We define 2x2 matrix multiplication as follows,

$$A = \begin{pmatrix} A_{1,1} & A_{1,2} \\ A_{2,1} & A_{2,2} \end{pmatrix} \qquad B = \begin{pmatrix} B_{1,1} & B_{1,2} \\ B_{2,1} & B_{2,2} \end{pmatrix} \qquad AB = \begin{pmatrix} C_{1,1} & C_{1,2} \\ C_{2,1} & C_{2,2} \end{pmatrix}$$

In matrix element notation, we define allowed $k = 7$ products as follows,

$$\begin{aligned}
I &= -A_{1,2}(B_{1,2} + B_{2,2}) \\
II &= A_{2,2}B_{2,1} \\
III &= (A_{1,1} - A_{1,2} - A_{2,1} + A_{2,2})(B_{2,1} + B_{2,2}) \\
IV &= A_{2,1}B_{1,1} \\
V &= (A_{1,1} - A_{2,1})(B_{1,1} + B_{1,2} + B_{2,1} + B_{2,2}) \\
VI &= (A_{1,1} - A_{1,2} - A_{2,1})(B_{1,2} + B_{2,1} + B_{2,2}) \\
VII &= (A_{1,1} - A_{1,2})B_{1,2}
\end{aligned}$$

From the terms in the products $I - VII$, we extract the following 8 simple identities,

$$\begin{aligned}
x_0 &= A_{1,1} - A_{2,1} & y_0 &= B_{1,2} + B_{2,2} \\
x_1 &= A_{1,1} - A_{1,2} & y_1 &= B_{2,1} + B_{2,2} \\
x_2 &= x_1 - A_{2,1} & y_2 &= y_1 + B_{1,2} \\
x_3 &= x_2 + A_{2,2} & y_3 &= y_2 + B_{1,1}
\end{aligned}$$

which in total, require 8 additions/subtractions. The products $I - VII$, may thus be simplified as follows,

$$\begin{aligned}
I &= -A_{1,2}y_0 \\
II &= A_{2,2}B_{2,1} \\
III &= x_3y_1 \\
IV &= A_{2,1}B_{1,1} \\
V &= x_0y_3 \\
VI &= x_2y_2 \\
VII &= x_1B_{1,2}
\end{aligned}$$

Using these simplified products, we can now construct, analogously as in the original Strassen case [1], the 4 elements $C_{i,j}$ of the matrix AB. For this particular Strassen-type algorithm, the sum of additions/subtractions is thus 16 in total.

$$\begin{aligned}
C_{1,1} &= I + IV + V - VI \\
C_{2,1} &= VII - I \\
C_{1,2} &= II + IV \\
C_{2,2} &= III - II - VI + VII
\end{aligned}$$

Conclusion

Neural Networks can be trained to learn exact subcubic algorithms for matrix multiplication. Given sufficient amounts of memory and computing power, it is very likely that practical and exact algorithms for higher-rank matrix multiplication can be discovered.

References

[1] V. Strassen Gaussian Elimination is not Optimal, Numerische Mathematik, 13, 354-356, 1969

[2] S. Winograd On multiplication of 2 x 2 matrices, Linear Algebra and its Applications, 4, 381-388, 1971

[3] D. Coppersmith, S. Winograd Matrix multiplication via arithmetic progressions, Journal of Symbolic Computation, 9, 251-280, 1990

[4] V. Vassilevska Williams Breaking the Coppersmith-Winograd barrier, https://people.csail.mit.edu/virgi/matrixmult-f.pdf, 2014

[5] A.M Davie, A.J. Stothers Improved bound for complexity of matrix multiplication, Proceedings of the Royal Society of Edinburgh, 143A, 351-370, 2013

[6] F. Le Gall Powers of tensors and fast matrix multiplication, Proceedings of the 39th International Symposium on Symbolic and Algebraic Computation (ISSAC 2014) arXiv:1401.7714, 2014

[7] V. Elser A network that learns Strassen multiplication, arXiv:1601.07227, 2016

[8] M. Tschannen, A. Khanna, A. Anandkumar StrassenNets: Deep Learning with a Multiplication Budget, arXiv:1712.03942, 2017