ARTICLE IN PRESS JLAMP:126

Journal of Logical and Algebraic Methods in Programming ••• (••••) •••–•••

Contents lists available at ScienceDirect

Journal of Logical and Algebraic Methods in Programming

www.elsevier.com/locate/jlamp

Gaussian elimination is not optimal, revisited

Hugo Daniel Macedo [a,b,*,1]

[a] DTU Compute, Technical University of Denmark, Kongens Lyngby, Denmark
[b] Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro, Brazil

ARTICLE INFO

Article history:
Received 14 June 2015
Received in revised form 1 June 2016
Accepted 14 June 2016
Available online xxxx

Keywords:
Tensor product
Matrix multiplication
Category theory

ABSTRACT

We refactor the universal law for the tensor product to express matrix multiplication as the product $M \cdot N$ of two matrices M and N thus making possible to use such matrix product to encode and transform algorithms performing matrix multiplication using techniques from linear algebra. We explore such possibility and show two stepwise refinements transforming the composition $M \cdot N$ into the Naïve and Strassen's matrix multiplication algorithms.

The inspection of the stepwise transformation of the composition of matrices $M \cdot N$ into the Naïve matrix multiplication algorithm evidences that the steps of the transformation correspond to apply Gaussian elimination to the columns of M and to the lines of N therefore providing explicit evidence on why "Gaussian elimination is not optimal", the aphorism serving as the title to the succinct paper introducing Strassen's matrix multiplication algorithm.

Although the end results are equations involving matrix products, our exposition builds upon previous works on the category of matrices (and the related category of finite vector spaces) which we extend by showing: why the direct sum $(\oplus, 0)$ monoid is not closed, a biproduct encoding of Gaussian elimination, and how to further apply it in the derivation of linear algebra algorithms.

1. Introduction

A universal law is a central concept within the framework of category theory assuring the uniqueness and existence of certain mathematical constructs. In particular, the tensor product $\otimes$ universal law [1, Theorem 22] states that every bilinear map $\beta : V \times W \to X$ is factorizable as the composition $\beta = \lceil \sigma \rceil \cdot \rho$ of a linear map $\lceil \sigma \rceil : V \otimes W \to X$ after the tensor embedding $\rho : V \times W \to V \otimes W$ which combines a pair of inputs $V \times W$, two spaces V and W, into a unique input, the tensor space $V \otimes W$.

Matrix–matrix multiplication ($\mathbb{MMM}$), the linear algebra operation at the heart of many other operations, which when applied to suitable matrices A and B, that is $\mathbb{MMM}(A, B)$, results in the matrix $C = A \times B$, is a bilinear map and therefore is suitable for the application of the universal law for the tensor product. In other words, it is possible to factorize the operation $\mathbb{MMM} : \mathbb{K}^{m\times q} \times \mathbb{K}^{q\times n} \to \mathbb{K}^{m\times n}$ as the composition of a unique linear map $\lceil \sigma \rceil : \mathbb{K}^{m\times q} \otimes \mathbb{K}^{q\times n} \to \mathbb{K}^{m\times n}$ after $\rho : \mathbb{K}^{m\times q} \times \mathbb{K}^{q\times n} \to \mathbb{K}^{m\times q} \otimes \mathbb{K}^{q\times n}$, the tensor embedding.

* Correspondence to: DTU Compute, Asmussens Alle, Building 303B, DK-2800 Kongens Lyngby, Denmark.
E-mail address: hudo@dtu.dk.

[1] This work was supported by a CNPq (Conselho Nacional de Desenvolvimento Científico e Tecnológico – Brasil) grant.

http://dx.doi.org/10.1016/j.jlamp.2016.06.003
2352-2208/

Please cite this article in press as: H.D. Macedo, Gaussian elimination is not optimal, revisited, J. Log. Algebraic Methods Program. (2016), http://dx.doi.org/10.1016/j.jlamp.2016.06.003

As it is folklore, linear maps on finite dimensional vector spaces correspond to matrices [2], thus one could question: What is the matrix involved in matrix multiplication? In this paper we address such question by reformulating it as: What is a concrete matrix representation of the linear map $\lceil\sigma\rceil$? Furthermore, given that in the context of functions between sets one can define a function that composes functions: Could one define a matrix σ that multiplies matrices?

This paper shows how an answer may be given by refactoring the tensor product universal law which is synthetically encoded as the categorical diagram (21). The generic spaces V, W, and X, interpreted as types/objects of the maps/morphisms β, $\lceil\sigma\rceil$, and ρ are instantiated with the $\mathbb{MMM}$ spaces: $\mathbb{K}^{m\times q}$, $\mathbb{K}^{q\times n}$, and $\mathbb{K}^{m\times n}$ respectively. Furthermore, β is instantiated with $\mathbb{MMM}$ in the diagram (21) and we solve the equation encoded in it to calculate a matrix σ that represents the linear map $\lceil\sigma\rceil$.

Furthermore, the quest for such matricial representation of the linear map $\lceil\sigma\rceil$ results in the factorization of $\mathbb{MMM}$ as the product of two matrices $M \cdot N$, where $M = \sigma$ and $N = (\mathbf{vec}\, A \otimes \mathbf{vec}\, B)$. Such representation makes it possible to encode and transform matrix algorithms applying basic linear algebra identities to the matrices σ and $(\mathbf{vec}\, A \otimes \mathbf{vec}\, B)$. In section 5 we explore such possibility and show matricial encodings and stepwise derivations of the algorithms performing Naïve (28) and Strassen's $\mathbb{MMM}$ (30).

The outcome of such derivations is beyond a pen and paper exercise on the derivation of matrix multiplication algorithms because through an inspection of the stepwise derivation of the Naïve algorithm performing $\mathbb{MMM}$ we explicitly find that such algorithm is the result of applying Gaussian Elimination ($\mathbb{GE}$) to the columns of σ and to the lines of $(\mathbf{vec}\, A \otimes \mathbf{vec}\, B)$. Thus the derivation leads us into a fruitful insight into a cornerstone paper [3] on matrix multiplication algorithms.

Such paper was written by Volker Strassen, a famous German mathematician known for his work on the analysis of algorithms and for introducing a $\mathbb{MMM}$ algorithm on square matrices with n lines and columns performing better — Strassen's algorithm runs in $\mathcal{O}(n^{2.807})$ time — than its Naïve version — running in $\mathcal{O}(n^3)$ time. Although the title "Gaussian elimination is not optimal" one does not find why, or an explicit connection between Gaussian elimination and the Naïve $\mathbb{MMM}$ algorithm. The derivation in section 5 explicitly shows that the Naïve algorithm for $\mathbb{MMM}$ is obtained by applying Gaussian elimination, and as the Naïve $\mathbb{MMM}$ is not an optimal algorithm then the conclusion: $\mathbb{GE}$ is not optimal.

Our full exposition expands and relies on previous research effort [4–6] on the connection between the domain of category theory, linear algebra, and computer science. Regardless, we encourage the reader to translate the sequence denoted by $\bowtie$ in equation (28) into a product of elementary matrices and to first understand how such matrix product is performing $\mathbb{GE}$ on the $(\mathbf{vec}\, A \otimes \mathbf{vec}\, B)$ lines. Only afterwards and after exercising the matrix product formulation of Strassen's algorithm (30) using plain linear algebra we recommend delving into the full exposition expanding the research connecting category theory, linear algebra, and $\mathbb{MMM}$ algorithm derivation.

The full exposition is structured as follows: In section 2 we set up essential background and identities that are needed to understand and justify in a categorical setting the stepwise justifications. The text is mainly an adaptation of results from [4,5] that we develop showing why when equipped with the direct sum $(\oplus, 0)$ monoid the category defined lacks closing and thus not Cartesian closed, thus not Turing complete. In section 3 we improve the translation of the traditional specification of the $\mathbb{GE}$ in terms of elementary matrices into the biproduct framework, derive new lemmas (18), and novel notation to specify the column and line application of $\mathbb{GE}$, equations (19) and (20), used in achieving the main results. In section 4 the tensor product universal law diagram is translated into an equation (27) encoding matrix multiplication in terms of a composition of two matrices. In section 5 we present a stepwise derivation of matrix multiplication algorithms, and as a byproduct we show how $\mathbb{GE}$ is related to the Naïve $\mathbb{MMM}$ algorithm. Then in section 6 we finish the exposition with some concluding remarks and possibilities for future work.

2. Background on the category of matrices

When dealing with matrices in a computational context we use a category where matrices are the morphisms and the objects transformed are natural numbers. We build upon the work in [5], but we chose not to depict matrices as arrows with right to left orientation. Although it makes sense to draw arrows backwards, due to the flow of matrix calculations, in a setting where we want to mingle matrices (linear functions) with common (non-linear) functions we opted to make morphism direction evolve.

The **Mat**$_{\mathbb{K}}$ *category*. The category of matrices has as objects natural numbers and morphisms the linear maps, we write an arrow $n \xrightarrow{R} m$ to denote a matrix which is traditionally denoted as an object in $\mathbb{K}^{m\times n}$ for a field $\mathbb{K}$.

$$R = \begin{bmatrix} r_{11} & \dots & r_{1n} \\ \vdots & \ddots & \vdots \\ r_{m1} & \dots & r_{mn} \end{bmatrix}_{m\times n} \qquad\qquad n \xrightarrow{R} m$$

Mat$_{\mathbb{K}}$ enables to define the type of a matrix at the correct abstract level when dealing with the computational structure that does not depend on the r_{ij} elements.[2]

[2] Assuming the elements are dealt with atomic machine operations, the programmer's task is to organize how to structure such operations.