Software Impacts 13 (2022) 100325

Contents lists available at ScienceDirect

Software Impacts

journal homepage: www.journals.elsevier.com/software-impacts

Original software publication

On the applicability of the Hadamard as an input modulator for problems of classification

J. de Curtò [a,c,*], I. de Zarzà [a,c], Hong Yan [a,b], Carlos T. Calafate [c]

[a] Centre for Intelligent Multidimensional Data Analysis, HK Science Park, Shatin, Hong Kong
[b] Department of Electrical Engineering, City University of Hong Kong, Kowloon, Hong Kong
[c] Department of Computer Engineering, Universitat Politècnica de València, València, Espanya

ARTICLE INFO

Keywords:
Hadamard
Learning
Classification
SIMD
SGD
Drones

ABSTRACT

The library explores the applicability of the Hadamard as an input modulator for problems of classification. It introduces a framework in C++ to use kernel approximates in the mini-batch setting with Stochastic Gradient Descent. The algorithm requires to compute the product of matrices Walsh Hadamard. A free-standing cache friendly Fast Walsh Hadamard that achieves compelling speed is provided, as well as a lightweight efficient CPU implementation of the method for research and practical purposes alike.

Code metadata

Current code version	v2.2
Permanent link to code/repository used for this code version	https://github.com/SoftwareImpacts/SIMPAC-2022-69
Permanent link to Reproducible Capsule	https://codeocean.com/capsule/4330307/tree/v1
Legal Code License	MIT License
Code versioning system used	git
Software code languages, tools, and services used	C++
Compilation requirements, operating environments & dependencies	
If available Link to developer documentation/manual	
Support email for questions	decurto@doctor.upv.es

1. Introduction

The applicability of the Hadamard as a modulator of the input samples for the purpose of classification has proven to be an effective technique for learning [1]. In the more general case, the Hadamard can be substituted for Fourier [2,3] or other transforms, such as the Signature Transform on the ongoing research work in [4]. The library provides a framework to use Logistic Regression along with the Hadamard, with iterative updates provided by SGD, see Fig. 1. Examples on several datasets are provided, achieving considerably good performance. A lightweight, fast, CPU optimized implementation is the key to a complete C++ library that also integrates a more general DL framework.

2. Scope

Our goal was to build a clear, concise and very well written code. Inspired by the principles of MapReduce [5]: few lines of code and reusability at its core.

3. McKernel

McKernel involves the following computation

$$\hat{Z} := \frac{1}{\sigma\sqrt{n}} C H G \Pi H B. \tag{1}$$

Here C,G and B are diagonal matrices, Π is a random permutation matrix and H is the Hadamard. Whenever the number of rows in W

The code (and data) in this article has been certified as Reproducible by Code Ocean: (https://codeocean.com/). More information on the Reproducibility Badge Initiative is available at https://www.elsevier.com/physical-sciences-and-engineering/computer-science/journals.

* Corresponding author at: Centre for Intelligent Multidimensional Data Analysis, HK Science Park, Shatin, Hong Kong.
E-mail addresses: decurto@doctor.upv.es (J. de Curtò), dezarza@doctor.upv.es (I. de Zarzà), h.yan@cityu.edu.hk (H. Yan), calafate@disca.upv.es (C.T. Calafate).

https://doi.org/10.1016/j.simpa.2022.100325
Received 18 May 2022; Accepted 26 May 2022

2665-9638/© 2022 The Author(s). Published by Elsevier B.V. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

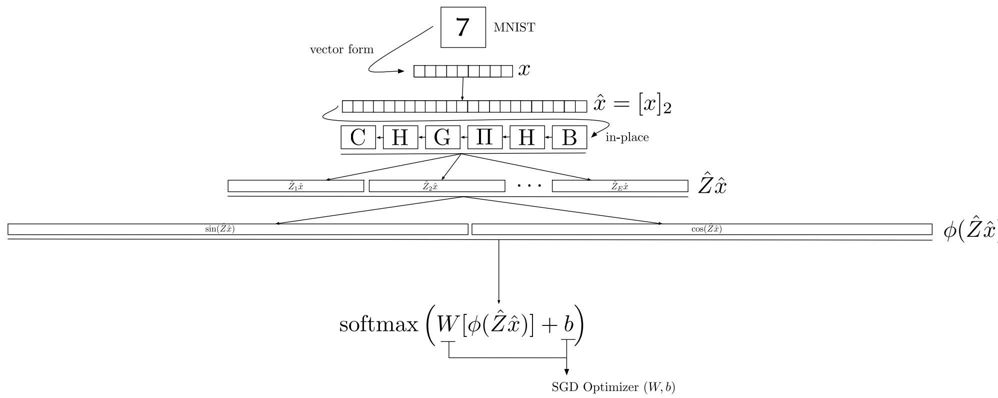

Fig. 1. Diagram of McKernel. We visually describe softmax($W\tilde{x}+b$) where $\tilde{x}$ = mckernel(x). The original image is padded in form of long vector to the nearest power of 2, mapping $\hat{Z}$ is applied in-place. Calibration C defines the choice of Kernel. The tensor is expanded by the number of Kernel Expansions E building a network with high compositionality. Finally, use real feature map ϕ, Eq. (2). SGD Optimizer finds appropriate weights W and bias b. Compute $\hat{Z}$ on-the-fly keeping same seed both for training and testing.

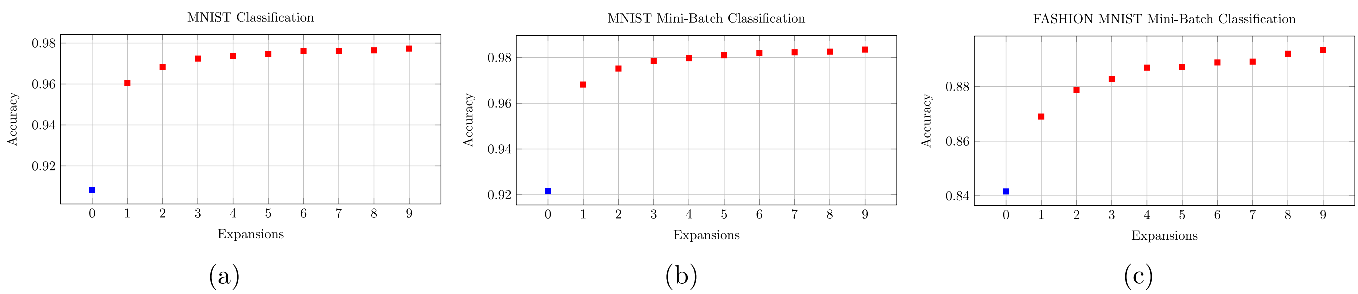

Fig. 2. Logistic Regression (blue) and RBF MATÉRN (red) with increasing number of Kernel Expansions. MNIST (a: full batch, b: mini-batch) and FASHION-MNIST (c: mini-batch). σ = 1.0, t = 40, seed 1398239763, learning rate γ = 0.001, lr 0.01, # epochs 20. For (b, c) batch size 10. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

exceeds the dimensionality of the data, we generate multiple instances of $\hat{Z}$, drawn i.i.d., until the required number of dimensions is obtained.

Binary B**.** This is a matrix with entries $B_{cc} \in \{\pm 1\}$, drawn from the uniform distribution. To avoid memory footprint, we use Murmurhash and extract bits from $h(c, x)$ with $x \in \{0, \ldots, N\}$.

Hadamard H**.** This matrix is iteratively composed of $H_n = \begin{bmatrix} H_{n-1} & H_{n-1} \\ H_{n-1} & -H_{n-1} \end{bmatrix}$. It is fixed and matrix–vector products are carried out efficiently in $O(n \log n)$ time using the Fast Walsh Hadamard, see Section 4.

Permutation Π**.** We generate a random permutation using the FISHER–YATES shuffle. The algorithm runs in linear time and its coefficients can be stored in $O(n)$ space. Moreover, to obtain a deterministic mapping, we replace the generator of random numbers with calls to the function of hashing.

Gaussian G**.** This is a matrix diagonal with entries i.i.d. Gaussian. We generate the random variates using the BOX–MULLER transform while substituting the generator of random numbers by calls to the function of hashing to allow us to recompute the values at any time without the need to store random numbers.

Calibration C**.** This is a random scaling operator whose behavior depends on the type of kernel chosen, such as the RBF MATÉRN Kernel, the RBF Kernel or any other radial spectral distribution.

Ultimately, the feature pairs are computed by assigning

$$[\cos(\hat{Z}x), \sin(\hat{Z}x)]. \tag{2}$$

In particular, McKernel assigns the features by using the real version of the complex feature map ϕ in [2]. SIMD vectorized instructions and cache locality are used to increase speed performance.

4. Fast Walsh Hadamard

A key part of the library is an efficient C++ implementation of Fast Walsh Hadamard. In particular, McKernel offers considerable improvement over other packages, due to automatic code generation, the use of SIMD intrinsics (SSE2 using 128 bit registers) and loop unrolling. This decreases the memory overhead. McKernel proceeds with vectorized sums and subtractions iteratively for the first $\frac{n}{2^z}$ input vector positions (where n is the length of the input vector and z the iteration starting from 1), computing the intermediate operations of the COOLEY–TUKEY algorithm till a small routine Hadamard that fits in cache. Then the algorithm continues in the same way but starting from the smallest length and doubling on each iteration the input dimension until the whole Fast Walsh Hadamard is done in-place.

5. Classification

SGD Optimizer finds W and b in

$$\mathrm{softmax}\left(W[\phi(\hat{Z}\hat{x})] + b\right), \tag{3}$$

where $\phi = (\sin(\cdot), \cos(\cdot))$, $\hat{x} = [x]_2$.

See Fig. 2 for a comparison between Logistic Regression and RBF MATÉRN on MNIST, full-batch and mini-batch, and FASHION-MNIST.

6. Impact overview

- C++ fast open-source Hadamard that works for any input size.
  Useful in any application where FOURIER Transform is applicable, such as compression (MPEG-4 AVC), encryption or quantum computing (Grover, Shor). Widely deployed in Signal Processing, Communications and Compressed Sensing.
- Approximate kernel expansions in log-linear time [1,6,7].
  Useful in large-scale setting to turn any linear classifier into non-linear, wherever SVM is still useful over DL methods. For instance, in robotics, unmanned aerial vehicles and ML for healthcare when the number of samples to train on is relatively bounded.
- As a stepping stone to integrate DL methods and kernel expansions [6,7].
  Useful to foster new DL architectures with better human-induced/mathematical priors [4,8].
- As a DL research framework.
  It offers multiple open questions: learning C and G adapts the type of kernel. Learning B acts as a mechanism of attention. Domains that could be further explored are: end-to-end training, self-supervised learning, meta-learning, integration with evolution strategies, NAS reducing substantially the search space and many others.

The library has been utilized both in industry (e.g. CIMDA, a private research center at City University of Hong Kong) and in academia. It is currently maintained and developed at the research group on Unmanned Aerial Vehicles at the Universitat Politècnica de València and at CIMDA.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work is supported by HK Innovation and Technology Commission, Hong Kong (InnoHK Project CIMDA) and HK Research Grants Council, Hong Kong (Project CityU 11204821).

References

[1] Q. Le, T. Sarlós, A. Smola, Approximating kernel expansions in loglinear time, in: ICML, 2013.
[2] A. Rahimi, B. Recht, Random features for large-scale kernel machines, in: NIPS, Vol. 20, 2007.
[3] A. Rahimi, B. Recht, Weighted sums of random kitchen sinks: Replacing minimization with randomization in learning, in: NIPS, Vol. 21, 2008.
[4] J. de Curtò, I. de Zarzà, H. Yan, C.T. Calafate, Learning with signatures, 2022, arXiv:2204.07953.
[5] J. Dean, S. Ghemawat, MapReduce: Simplified data processing on large clusters, in: OSDI'04: Sixth Symposium on Operating System Design and Implementation, San Francisco, CA, 2004, pp. 137–150.
[6] I. de Zarzà, A Unifying Theory of Learning: DL Meets Kernel Methods (Ph.D. thesis), ETH Zürich, 2021.
[7] J. de Curtò, Vision and Learning in the Context of Exploratory Rovers (Ph.D. thesis), ETH Zürich, 2021.
[8] J. de Curtò, I. de Zarzà, H. Yan, Signature and log-signature for the study of empirical distributions generated with GANs, 2022, arXiv:2203.03226.