OPTIMAL QUANTIZATION OF RANK-ONE MATRICES IN FLOATING-POINT ARITHMETIC—WITH APPLICATIONS TO BUTTERFLY FACTORIZATIONS[*]

RÉMI GRIBONVAL[†], THEO MARY[‡], AND ELISA RICCIETTI[†]

Abstract. We consider the problem of optimally quantizing rank-one matrices to low precision floating-point arithmetic. We first explain that the naive strategy of separately quantizing the two rank-one factors can be far from optimal, and we provide worst case error bounds to support this observation. We characterize the optimal solution as the quantization of suitably scaled factors of the rank-one matrix and we develop an algorithm of tractable complexity to find the optimal scaling parameters. Using random rank-one matrices, we show experimentally that our algorithm can significantly reduce the quantization error. We then apply this algorithm to the quantization of butterfly factorizations, a fundamental tool that appears in many fast linear transforms. We show how the properties of butterfly supports can be exploited to approach the problem via a series of rank-one quantization problems and we employ our algorithm as a building block in a heuristic procedure to quantize a product of butterfly factors. We show that, despite being only heuristic, this strategy can be much more accurate than quantizing each factor independently or, equivalently, can achieve storage reductions of up to 30% with no loss of accuracy.

Key words. numerical linear algebra, low-rank approximations, rank-one matrices, floating-point arihmetic, low precision quantization, butterfly factorizations, sparse matrices

AMS subject classifications. 65F55,65Y04,65Y20,15B99

1. Introduction.

We consider the problem of optimally quantizing rank-one matrices: given $\mathbb{F}_t$ a finite set of floating-point numbers with t-bit significand and unquantized (or high precision) vectors $x \in \mathbb{R}^m$, $y \in \mathbb{R}^n$, we wish to solve

$$\min_{\widehat{x}\in\mathbb{F}_t^m,\widehat{y}\in\mathbb{F}_t^n} \|xy^\top - \widehat{x}\widehat{y}^\top\|^2. \tag{1.1}$$

The problem of factorizing a rank-one matrix arises in many applications in linear algebra, signal processing, and machine learning. It is for instance a building block in the generic problem of rank-r decomposition, which can indeed be solved by singular value decomposition (SVD), a process that can be decomposed as a series of r rank-one approximations [10].

The rank-one problem is also interesting on its own. For instance, it is a key ingredient in the efficient solution of specific instances of sparse matrix factorization (SMF), which seeks to approximate a large dense matrix Z as a product of two or more sparse factors $B_1, \dots, B_L$, $L \geq 2$ [12]. The sparse factors usually belong to a structured family of matrices, so that the sparsity pattern can be easily exploited to reduce the computational cost of linear operations involving the matrix Z. A particularly useful family is that of *butterfly matrices*, widely used for their strong expressivity and extreme sparsity pattern: they only have two nonzeros per row and per column and appear for instance in the factorizations of the Hadamard and of the Fourier matrices [1]. Due to the structure of butterfly matrices, certain partial products of their factors can be decomposed into blocks that admit an exact representation as rank-one matrices. This property has been used in the literature [11, 10] to design

[*]Version of June 7, 2022. Code available at: https://perso.ens-lyon.fr/elisa.riccietti/code.php, https://inria.hal.science/hal-04124171

[†]Univ Lyon, ENS de Lyon, UCBL, CNRS, Inria, LIP, F-69342, LYON Cedex 07, France (remi.gribonval@ens-lyon.fr; elisa.riccietti@ens-lyon.fr)

[‡]Sorbonne Université, CNRS, LIP6, Paris, F-75005, France (theo.mary@lip6.fr)

algorithms to approximate a given matrix Z with a butterfly product by solving a series of rank-one problems.

Due to the growing size of matrices in such applications, optimizing the memory usage and computational cost of algorithms involving them is important. Low precision quantization methods have been applied in many fields to deal with the always growing scale of models and datasets. For example, low precision floating-point arithmetic has been exploited in linear algebra to reduce the cost of many computational tasks, often while preserving a high accuracy thanks to mixed precision algorithms; see [8] for a recent survey. Low precision quantization is also a key tool in the training and inference of large deep neural networks (DNN), see for example [6]. The interest for problem (1.1) is thus further motivated by recent work on the approximation of weight matrices in DNN with sparse structured matrices [4, 3], especially when butterfly factorizations are involved.

The natural strategy to attack problem (1.1) is to separately quantize the two rank-one factors x and y by a round-to-nearest (RTN) strategy: each coefficient of x and y is mapped to its closest floating-point neighbor. Indeed, this strategy minimizes separately the error on the quantization of each of the two factors: see (2.7) below. We derive an upper bound on the worst case quantization error obtained with this RTN strategy, which is of order 2^{-t}. Somewhat surprisingly, and importantly, we prove in this work that the RTN strategy is not necessarily optimal, in the sense that (potentially much) lower quantization errors on the overall product can be achieved for some other choice of $\widehat{x}$ and $\widehat{y}$.

The optimal quantization method that we develop in this paper is based on the observation that the problem has a scaling invariance: for any scaling parameter $\lambda \neq 0$, the unquantized product is independent of λ: $(\lambda x)\left(\frac{1}{\lambda}y\right)^\top = xy^\top$. However, the product of the quantized versions of λx and $\frac{1}{\lambda}y^\top$ does depend on λ, and a well-chosen λ may yield a more accurate approximation. We develop an analysis that characterizes the optimal solution as the quantization of scaled vectors λx and $\mu y^\top$, where, crucially, μ is close, but not equal to, $1/\lambda$.

Despite the combinatorial nature of the problem—the number of possible quantizations is exponential in both t, the number of significand bits of the target floating-point format, and the dimensions m, n—we show that the optimal scaling parameters can be found with tractable complexity. We develop an algorithm that, by enumerating a finite number of values for λ, achieves a complexity in $O(2^t mn)$ in time and in $O(2^t \min(m, n))$ in space, which is thus only polynomial in the dimensions m, n. The exponential dependence in t is tractable in a context of coarse to moderate quantization and we tested the algorithm with success up to $t = 11$. For the applications that we consider, higher values of t are not of interest.

We demonstrate both empirically and theoretically that the proposed optimal algorithm can indeed be much more accurate than the simple strategy based on separately quantizing x and y with RTN. Theoretically, we derive upper and lower bounds on the worst case optimal quantization error. In particular, a theoretical lower bound, whose numerical evaluation for $t \leq 11$ is of order $2^{-1.6t}$, is obtained by studying the properties of $\mathbb{F}_t\mathbb{F}_t$, the set of elements that can be written as the product of two t-bit floating-point numbers, to which the elements of $\widehat{x}\widehat{y}^\top$ belong. Empirically, we investigate the behavior of the proposed algorithm with random rank-one matrices. We show that it preserves a high accuracy in the product $xy^\top$ despite using a reduced number of bits to quantize x and y.

Finally, we apply the proposed algorithm to the quantization of butterfly factorizations $B_1 \dots B_L$. As mentioned before, due to the special structure of butterfly matrices, certain partial products of their factors can be decomposed into rank-one blocks. We prove that in the case of two factors ($L = 2$), we can employ the optimal rank-one quantization to obtain an optimally quantized butterfly factorization, whereas optimality is no longer guaranteed for more than two factors. In this case, we propose two heuristics that use the optimal rank-one quantization algorithm as a building block to quantize different sets of partial products of the factors. We show experimentally that, despite being only heuristic, this strategy can be much more accurate than quantizing each butterfly factor independently.

The rest of this paper is organized as follows. In Section 2 we present the notations and technical ingredients necessary for the rest of the paper, and we introduce the RTN strategy and its corresponding error bounds. Then in Section 3 we first derive lower and upper bounds on the worst case quantization error of a rank-one matrix; the lower bound motivates the interest of finding an optimal solution to (1.1). In Section 4, we characterize such an optimal solution as a rank-one matrix with suitably scaled factors. In Section 5, we develop an algorithm to find the optimal scaling parameters with tractable complexity. We illustrate the empirical behavior of this algorithm for random rank-one matrices in Section 6. We then apply it to the quantization of butterfly factorizations in Section 7. Finally, we provide our concluding remarks in Section 8.

2. Technical preliminaries and notations.

We consider the problem of quantizing a rank-one matrix $xy^\top$, $x \in \mathbb{R}^m$, $y \in \mathbb{R}^n$ with t bits of precision, that is, to map the elements of x and x from $\mathbb{R}$ to a finite set $\mathbb{F}_t$ of t-bit numbers. Denoting the quantized x and y as $\widehat{x}$ and $\widehat{y}$, our goal is to minimize the quantization error, that it to solve (1.1).

We will work with floating-point arithmetic, and define $\mathbb{F}_t$ to be the set of floating-point numbers with $t \geq 1$ bits of significand:

$$\mathbb{F}_t \coloneqq \{\pm k 2^{e-t},\ k \in [\![2^{t-1}, 2^t - 1]\!],\ e \in \mathbb{Z}\}. \tag{2.1}$$

The significand $\pm k$ can indeed be encoded with t bits: one bit for the sign, and $t-1$ bits to describe k (since the most significant bit of k is implicitly fixed to one). Note that in practice, the exponent e of floating-point numbers is restricted to a finite range $[\![e_{\min}, e_{\max}]\!]$ that depends on how many bits are used to encode the exponent. Throughout this paper, we ignore issues related to the exponent part of the floating-point representation: we assume that a fixed number of additional bits is used for encoding the exponent, and that it is sufficient to prevent overflow and underflow, so that all elements in x and y belong to the representable range of the floating-point arithmetic. We also ignore subnormal numbers (numbers with $e = 0$ and k smaller than 2^{t-1}).

We define the $\mathrm{round}_t(\cdot)$ function that maps any $a \in \mathbb{R}$ to its (set of) nearest neighbor(s) in $\mathbb{F}_t$:

$$\mathrm{round}_t(a) \in \arg\min_{\widehat{a} \in \mathbb{F}_t} |a - \widehat{a}|. \tag{2.2}$$

Note that in most cases there is a unique nearest neighbor, except in the case of a tie, in which case there are two of them: for this technical reason, we consider $\mathrm{round}_t(\cdot)$ to be set-valued. The unit roundoff of $\mathbb{F}_t$ is defined [7, Thm. 2.2] as

$$u = u_t \coloneqq 2^{-t}, \tag{2.3}$$

and the closely related quantity

$$v = v_t \coloneqq \frac{u_t}{1 + u_t}, \tag{2.4}$$

bounds the maximum relative distance between any element in the representable range and its nearest neighbor in $\mathbb{F}_t$ [9]:

$$\forall a \in \mathbb{R}, \quad |\mathrm{round}_t(a) - a| \le v_t |a|. \tag{2.5}$$

In case of a tie, observe that $|\mathrm{round}_t(a) - a|$ is equal for both values of $\mathrm{round}_t(a)$ so that the expression is well defined. The bound is sharp [9]: there exists $a > 0$ such that $\mathrm{round}_t(a) - a = v_t a$. For brevity we will use $\mathrm{round}(\cdot)$, u, and v instead of $\mathrm{round}_t(\cdot)$, u_t, and v_t when t is clear from context.

The round-to-nearest (RTN) strategy to quantize the product $xy^\top$ consists in mapping each element of x and y to their nearest neighbor in $\mathbb{F}_t$, with some suitable tie-breaking rule. This yields quantized $\widehat{x}$ and $\widehat{y}$ satisfying

$$\widehat{x} = \mathrm{round}(x) = x + \Delta x, \quad \|\Delta x\| \le v\|x\|, \tag{2.6a}$$
$$\widehat{y} = \mathrm{round}(y) = y + \Delta y, \quad \|\Delta y\| \le v\|y\|, \tag{2.6b}$$

where $\mathrm{round}(\cdot)$ is applied elementwise. It is worth noting that $\widehat{x}$ and $\widehat{y}$ are, by definition, optimal quantizations of x and y, in the sense that they are solutions of the problems

$$\min_{\widehat{x}} \|x - \widehat{x}\|, \quad \widehat{x} \in \mathbb{F}_t^m, \tag{2.7a}$$
$$\min_{\widehat{y}} \|y - \widehat{y}\|, \quad \widehat{y} \in \mathbb{F}_t^n. \tag{2.7b}$$

However, our goal is not to solve these problems but rather to solve (1.1), which seeks to minimize the quantization error on the overall product $xy^\top$. The following result bounds the worst case quantization error obtained with this RTN strategy.

Lemma 2.1. *For any $x \in \mathbb{R}^m$ and $y \in \mathbb{R}^n$ we have*

$$\|xy^\top - \mathrm{round}(x)\mathrm{round}(y)^\top\| \le (2v + v^2)\|x\|\|y\|. \tag{2.8}$$

This bound is sharp: there are vectors x, y for which the round-to-nearest strategy achieves exactly *an error of $2v + v^2$.*

Proof. We prove the result in the more general case of matrices $X \in \mathbb{R}^{m\times r}$ and $Y \in \mathbb{R}^{n\times r}$, with $r \ge 1$. Recall that for any matrices $\|MN\| \le \|M\|_{2\to 2} \cdot \|N\| \le \|M\| \cdot \|N\|$ where $\|M\|_{2\to 2} := \sup_{x\neq 0} \|Mx\|_2/\|x\|_2$. As a result

$$\begin{aligned}
\|XY^\top - \widehat{X}\widehat{Y}^\top\| &\le \|X(Y^\top - \widehat{Y}^\top)\| + \|(X - \widehat{X})\widehat{Y}^\top\| \\
&\le \|X\| \cdot \|Y^\top - \widehat{Y}^\top\| + \|X - \widehat{X}\| \cdot \|\widehat{Y}^\top\| \\
&= \|X\| \cdot \|\Delta Y\| + \|\Delta X\| \cdot \|Y + \Delta Y\| \\
&\le v\|X\|\|Y\| + v\|X\| \cdot (\|Y\| + \|\Delta Y\|) \le (2v + v^2)\|X\|\|Y\|.
\end{aligned}$$

For the converse result, given the sharpness of (2.5), consider $a > 0$ a real number reaching maximal *positive* relative error $(\mathrm{round}(a) - a)/a = v$ and X, Y two matrices filled with zeros except the upper left entry which is set to a. We have

$$\begin{aligned}
\frac{\|XY^\top - \widehat{X}\widehat{Y}^\top\|}{\|XY^\top\|} &= \frac{|a^2 - (\mathrm{round}(a))^2|}{a^2} = \frac{|a - \mathrm{round}(a)|}{a}\frac{a + \mathrm{round}(a)}{a} \\
&= v\frac{\mathrm{round}(a) - a + 2a}{a} = 2v + v^2.
\end{aligned}$$ ∎

3. Worst case quantization error bounds for rank-one matrices.

The RTN strategy described in the previous section is not necessarily optimal, in the sense that lower quantization errors can be achieved for some other choice of $\widehat{x}$ and $\widehat{y}$. In this section, we study this problem from a theoretical perspective, by computing worst case error bounds on the quantization of rank-one matrices. To do so, we define for any non-empty subset $\mathbb{S} \subset \mathbb{R}$ the set of rank-one matrices that can be decomposed into two factors with coefficients in $\mathbb{S}$,

$$\Sigma(\mathbb{S}) := \Sigma^{m\times n}(\mathbb{S}) = \{\widehat{x}\widehat{y}^\top : \widehat{x} \in \mathbb{S}^m, \widehat{y} \in \mathbb{S}^n\} \subseteq \mathbb{R}^{m\times n}. \tag{3.1}$$

The distance from any matrix $M \in \mathbb{R}^{m\times n}$ to any non-empty subset $\Sigma(\mathbb{S}) \subseteq \mathbb{R}^{m\times n}$ is

$$d(M, \Sigma(\mathbb{S})) := \inf_{\widehat{M}\in\Sigma(\mathbb{S})} \|M - \widehat{M}\|. \tag{3.2}$$

We are thus particularly interested in studying $d(M, \Sigma(\mathbb{F}_t))$ when $M \in \Sigma(\mathbb{R})$.

Note that the coefficients of matrices $\widehat{M} \in \Sigma(\mathbb{F}_t)$ belong to the set

$$\mathbb{F}_t\mathbb{F}_t := \{\widehat{x}\widehat{y},\ \widehat{x} \in \mathbb{F}_t,\ \widehat{y} \in \mathbb{F}_t\}, \tag{3.3}$$

which is the set of numbers that can be written as the product of two t-bit floating-point numbers. Therefore, in order to derive a lower bound on $d(M, \Sigma(\mathbb{F}_t))$, we can study the properties of this set and more specifically the worst case error when quantizing a real number z by a number of the form $\widehat{z} = \widehat{x}\widehat{y} \in \mathbb{F}_t\mathbb{F}_t$.

We consider the distance of $z \in \mathbb{R}$ to a non-empty set $\mathbb{S} \subseteq \mathbb{R}$:

$$d(z, \mathbb{S}) = \inf_{\widehat{z}\in\mathbb{S}} |\widehat{z} - z|, \tag{3.4}$$

which is the instantiation of (3.2) for $m = n = 1$ and $\Sigma = \mathbb{S}$. Then, the *worst case relative error* of quantizing an element z on a non-empty set $\mathbb{S}$ is given by

$$\epsilon(\mathbb{S}) := \sup_{z\in\mathbb{R}\setminus\{0\}} \frac{d(z, \mathbb{S})}{|z|}. \tag{3.5}$$

For example, for any $t > 0$ we have [7, Thm. 2.2].

$$\epsilon(\mathbb{F}_t) = v_t = \frac{2^{-t}}{1 + 2^{-t}}. \tag{3.6}$$

With this formalism, we can provide the following bounds on the worst case relative error of optimal quantization over rank-one matrices.

Lemma 3.1. *Consider integers $m, n, t \geq 1$. With the notation $\Sigma(\mathbb{S}) \subseteq \mathbb{R}^{m\times n}$ from (3.1), we denote*

$$\epsilon(\Sigma(\mathbb{R}), \Sigma(\mathbb{F}_t)) := \sup_{0\neq xy^\top\in\Sigma(\mathbb{R})} \frac{d(xy^\top, \Sigma(\mathbb{F}_t))}{\|xy^\top\|}. \tag{3.7}$$

It satisfies

$$\epsilon(\mathbb{F}_t\mathbb{F}_t) \leq \epsilon(\Sigma(\mathbb{R}), \Sigma(\mathbb{F}_t)) \leq 2v_t + v_t^2. \tag{3.8}$$

Proof. The upper bound is a direct consequence of Lemma 2.1, since an optimal quantization will always be at least as accurate as the RTN quantization. By the very definition of $\epsilon(\mathbb{F}_t\mathbb{F}_t)$, the lower bound is achieved by considering a worst case over pairs of the form $x = x_1e_1 \in \mathbb{R}^m$, $y = y_1f_1$ where $x_1, y_1 \in \mathbb{R}$ and e_1 (resp. f_1) is the first canonical basis vector in $\mathbb{R}^m$ (resp. in $\mathbb{R}^n$). $\square$

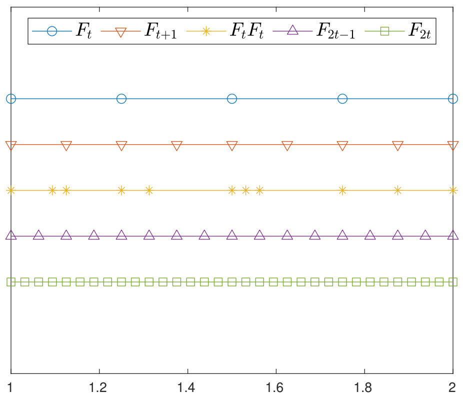


Fig. 3.1: Elements of $\mathbb{F}_t$, $\mathbb{F}_{t+1}$, $\mathbb{F}_t\mathbb{F}_t$, $\mathbb{F}_{2t-1}$, and $\mathbb{F}_{2t}$ in the interval $[1,2]$, for $t=3$. This figure illustrates Lemma 3.2.

To the best of our knowledge, $\epsilon(\mathbb{F}_t\mathbb{F}_t)$ does not have a known simple closed-form expression as a function of t. Nevertheless, we may gain insight on its size by noting that if $S_1 \subseteq S_2$, then $\epsilon(S_1) \geq \epsilon(S_2)$. We are thus interested in finding sets S_1 and S_2 such that $S_1 \subseteq \mathbb{F}_t\mathbb{F}_t \subseteq S_2$. The following lemma accomplishes this by using the sets of floating-point numbers $\mathbb{F}_p$.

Lemma 3.2. *For any $t \geq 1$, $\mathbb{F}_t \subseteq \mathbb{F}_t\mathbb{F}_t \subseteq \mathbb{F}_{2t}$. These inclusions are sharp, in the sense that $\mathbb{F}_{t+1} \nsubseteq \mathbb{F}_t\mathbb{F}_t$ and $\mathbb{F}_t\mathbb{F}_t \nsubseteq \mathbb{F}_{2t-1}$ (the latter only holds for $t \geq 2$).*

Proof. The $\mathbb{F}_t \subseteq \mathbb{F}_t\mathbb{F}_t$ inclusion is trivial since $1 \in \mathbb{F}_t$.

To prove that $\mathbb{F}_t\mathbb{F}_t \subseteq \mathbb{F}_{2t}$, let $x, y \in \mathbb{F}_t$ such that $x = \pm m_x 2^{e_x - t}$ and $y = \pm m_y 2^{e_y - t}$. Then

$$xy = \pm m_x m_y 2^{e_x + e_y - 2t} := \pm m 2^{e-2t}.$$

Clearly, since $m_x, m_y \leq 2^t - 1$, m satisfies the upper bound $m \leq 2^{2t} - 1$. We now distinguish two cases. If $m \geq 2^{2t-1}$ then by definition of $\mathbb{F}_{2t}$ we have $xy \in \mathbb{F}_{2t}$. Otherwise, if $m < 2^{2t-1}$, then letting $m' = 2m$ we have $m' < 2^{2t}$, i.e. $m' \leq 2^{2t} - 1$. Moreover since $m_x, m_y \geq 2^{t-1}$, m' also satisfies the lower bound $m' \geq 2^{2t-1} \geq 2^{t-1}$. Therefore $xy = m' 2^{e-1-2t} \in \mathbb{F}_{2t}$.

To prove that $\mathbb{F}_{t+1} \nsubseteq \mathbb{F}_t\mathbb{F}_t$, observe that by Bertrand's postulate [14, p. 371-382], there exists a prime p such that $2^t < p < 2^{t+1}$. Since $p \in [\![2^t, 2^{t+1} - 1]\!]$ we have $p \in \mathbb{F}_{t+1}$ by (2.1). However p does not belong to $\mathbb{F}_t\mathbb{F}_t$, since it cannot be written as $m_1 m_2 2^{f'}$, with $m_1, m_2 \leq 2^t - 1$.

Finally, to prove that $\mathbb{F}_t\mathbb{F}_t \nsubseteq \mathbb{F}_{2t-1}$ assuming that $t \geq 2$, it suffices to consider the example $x = (2^{t-1} + 1)2^{e-t} \in \mathbb{F}_t$ and $y = (2^t - 1)2^{f-t} \in \mathbb{F}_t$. Then $xy \in \mathbb{F}_t\mathbb{F}_t$ is given by $xy = m2^{e+f-2t}$ with $m = 2^{2t-1} + 2^{t-1} - 1$. Observe that m is whole but exceeds the required upper bound $2^{2t-1} - 1$; and that since $t \geq 2$, m is odd hence for any $s \geq 1$, $m/2^s$ cannot be an integer. Thus $xy \notin \mathbb{F}_{2t-1}$. ∎

We illustrate Lemma 3.2 in Figure 3.1 for $t = 3$. Lemma 3.2 shows that $\epsilon(\mathbb{F}_t\mathbb{F}_t)$ lies in between $\epsilon(\mathbb{F}_t) = v_t \approx 2^{-t}$ and $\epsilon(\mathbb{F}_{2t}) = v_{2t} \approx 2^{-2t}$. Moreover, we can numerically compute $\epsilon(\mathbb{F}_t\mathbb{F}_t)$ exactly by using the following observation, which applies to $\mathbb{S} = \mathbb{F}_t$ as well as to $\mathbb{S} = \mathbb{F}_t\mathbb{F}_t$.

Lemma 3.3. *For any set $\mathbb{S} \subseteq \mathbb{R}$ and $a \in \mathbb{R}$ denote $aS := \{ax : x \in \mathbb{S}\}$. If*

$\mathbb{S} = -\mathbb{S} = 2\mathbb{S}$ *is non-empty then for any* $z' \in \mathbb{R}\backslash\{0\}$ *there is* $z \in [1,2]$ *such that*

$$\frac{d(z',\mathbb{S})}{|z'|} = \frac{d(z,\mathbb{S})}{|z|} \tag{3.9}$$

Proof. Since $z' \neq 0$, there exists $p \in \mathbb{Z}$ such that $z := 2^p|z'| \in [1,2]$. Since $\mathbb{S} = -\mathbb{S}$ and $\mathbb{S} = 2\mathbb{S}$ we have $\mathbb{S} = \mathtt{sign}(z')2^{-p}\mathbb{S}$ hence

$$\begin{aligned}\frac{d(z,\mathbb{S})}{|z|} &= \min_{\widehat{z}\in\mathbb{S}} \frac{|z-\widehat{z}|}{|z|} = \min_{\widehat{z}\in\mathbb{S}} \frac{|2^p|z'| - 2^p(\widehat{z}2^{-p})|}{|2^p z'|} = \min_{\widehat{z}'\in\mathbb{S}} \frac{||z'| - \widehat{z}'2^{-p}|}{|z'|} \\ &= \min_{\widehat{z}'\in\mathbb{S}} \frac{|z' - \mathtt{sign}(z')\widehat{z}'2^{-p}|}{|z'|} = \min_{\widehat{z}'\in\mathbb{S}} \frac{|z'-\widehat{z}'|}{|z'|} = \frac{d(z',\mathbb{S})}{|z'|}.\end{aligned}$$

□

Corollary 3.4. *Consider a non-empty* $\mathbb{S} \subset \mathbb{R}$ *such that* $\mathbb{S} = -\mathbb{S}$, $\mathbb{S} = 2\mathbb{S}$, *and* $\mathbb{S} \cap [1,2]$ *is finite. Denote* $\widehat{z}_i$ *the elements of* $\mathbb{S}\cap[1,2]$ *in increasing order. We have*

$$\epsilon(\mathbb{S}) = \max_i \frac{\widehat{z}_{i+1} - \widehat{z}_i}{\widehat{z}_{i+1} + \widehat{z}_i}. \tag{3.10}$$

Proof. By Lemma 3.3, since $\mathbb{S} = -\mathbb{S} = 2\mathbb{S}$, we have

$$\epsilon(\mathbb{S}) = \sup_{z\in[1,2]} \frac{d(z,\mathbb{S})}{|z|} = \max_i \sup_{z\in[\widehat{z}_i,\widehat{z}_{i+1}]} \frac{d(z,\mathbb{S})}{|z|}.$$

Now, observe that, in each interval $[\widehat{z}_i, \widehat{z}_{i+1}]$, the relative error is expressed as

$$\frac{d(z,\mathbb{S})}{|z|} = \begin{cases} 1 - \frac{\widehat{z}_i}{z}, & \text{if } z \in [\widehat{z}_i, m_i] \\ \frac{\widehat{z}_{i+1}}{z} - 1 & \text{if } z \in [m_i, \widehat{z}_{i+1}] \end{cases} \qquad \text{where } m_i := (\widehat{z}_{i+1} + \widehat{z}_i)/2 \tag{3.11}$$

and is thus maximized at the midpoint m_i, where it is equal to

$$\frac{d(m_i,\mathbb{S})}{|m_i|} = \frac{m_i - \widehat{z}_i}{m_i} = \frac{\widehat{z}_{i+1} - \widehat{z}_i}{\widehat{z}_{i+1} + \widehat{z}_i}.$$

□

Since $\mathbb{F}_t\mathbb{F}_t$ is stable by sign flip and by multiplication by two, Corollary 3.4 shows that we can numerically compute $\epsilon(\mathbb{F}_t\mathbb{F}_t)$ simply by evaluating (3.10) for all $\widehat{z}_i$, of which there are at most 2^{2t} since $\mathbb{F}_t\mathbb{F}_t \subseteq \mathbb{F}_{2t}$. For modest values of t, this is tractable. We have done this for $t \leq 11$ and report the result in Figure 3.2 (left). Using a linear fit, we observe $\epsilon(\mathbb{F}_t\mathbb{F}_t)$ to behave approximately as $2^{-1.6t}$. The gap between $\epsilon(\mathbb{F}_t)$ and $\epsilon(\mathbb{F}_t\mathbb{F}_t)$ thus increases with t.

The proof of Lemma 3.3 also suggests a simple $O(2^{2t})$ algorithm to find an optimal quantized pair $\widehat{x}$, $\widehat{y}$ given x, y. A first step is to select an integer p such that $2^p xy \in [1,2]$. Then, we can simply enumerate all elements of $\mathbb{F}_t\mathbb{F}_t \cap [1,2]$ and select one minimizing the distance to $2^p xy$. Denote such an element as $\widehat{z} = ab$, $a \in \mathbb{F}_t$, $b \in \mathbb{F}_t$; then we obtain (for example) $\widehat{x} = a$ and $\widehat{y} = 2^{-p}b$. Moreover, if we assume that the elements $\widehat{z}_i$ of $\mathbb{F}_t\mathbb{F}_t \cap [1,2]$ are precomputed and stored in increasing order (together with their factors $a_i, b_i \in \mathbb{F}_t$, with a memory complexity $O(2^{2t})$, then a simple dichotomy procedure allows to find $\widehat{x}$, $\widehat{y}$ with a time complexity $O(t)$. Refinements of these ideas will be at the core of the algorithm proposed in the next sections to optimally quantize $xy^\top$, $x \in \mathbb{R}^m, y \in \mathbb{R}^n$.

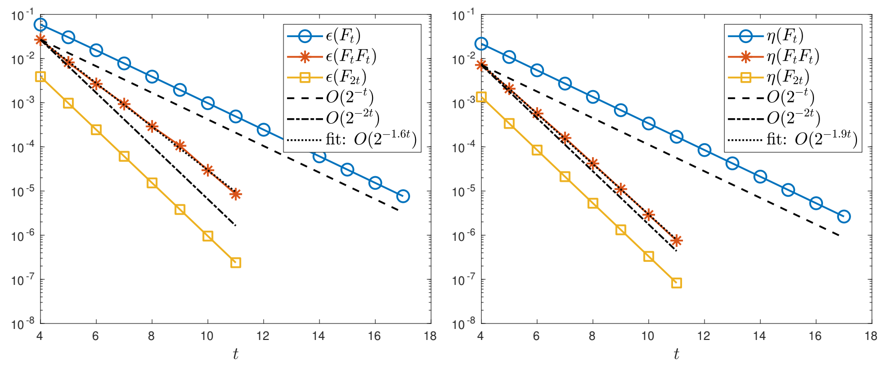


Fig. 3.2: Left: Values of the worst case relative errors $\epsilon(\mathbb{F}_t) = v_t \approx 2^{-t}$, $\epsilon(\mathbb{F}_t\mathbb{F}_t)$ (computed by (3.10)), and $\epsilon(\mathbb{F}_{2t}) = v_{2t} \approx 2^{-2t}$, as a function of t. From a linear fit $\epsilon(\mathbb{F}_t\mathbb{F}_t) \approx 2^{-1.6t}$. Right: Values of average relative errors $\eta(\mathbb{F}_t)$, $\eta(\mathbb{F}_t\mathbb{F}_t)$ and $\eta(\mathbb{F}_{2t})$ (defined in (3.12), with $\Omega = [1,2]$), as a function of t. From a linear fit $\eta(\mathbb{F}_t\mathbb{F}_t) \sim 2^{-1.9t}$.

Having analyzed the behavior of the worst case quantization error, we now explain why the error may be even better on average. There is no standard definition of *average relative error*, but we may for example define it as

$$\eta(\mathbb{S}|\Omega) = \frac{1}{|\Omega|}\int_\Omega \frac{d(z,\mathbb{S})}{|z|}\mathrm{d}z$$

for any non-empty subset $\mathbb{S} \subseteq \mathbb{R}$ and any measurable domain $\Omega \subseteq \mathbb{R}\backslash\{0\}$ with Lebesgue measure $0 < |\Omega| < +\infty$. Observe that we always have $\eta(\mathbb{S}|\Omega) \le \epsilon(\mathbb{S})$. For $\mathbb{S} = \mathbb{F}_t$ or $\mathbb{S} = \mathbb{F}_t\mathbb{F}_t$, in light of Lemma 3.3 it is natural to choose $\Omega = [1,2]$, so we define

$$\eta(\mathbb{S}) = \int_1^2 \frac{d(z,\mathbb{S})}{|z|}\mathrm{d}z \text{ for } \mathbb{S} = \mathbb{F}_t \text{ and for } \mathbb{S} = \mathbb{F}_t\mathbb{F}_t. \tag{3.12}$$

For the standard set $\mathbb{F}_t$ of floating-point numbers, the average relative error $\eta(\mathbb{F}_t)$ is not a very interesting metric, as it is within a constant factor of the worst case relative error $\epsilon(\mathbb{F}_t) = u/(1+u)$. Indeed, because of the uniform spacing of the elements of $\mathbb{F}_t$ within intervals of consecutive powers of two, and in particular in $[1,2]$, we have

$$\int_1^2 d(z,\mathbb{F}_t) = \sum_{i=1}^{2^{t-1}} \int_{1+2(i-1)u}^{1+2iu} d(z,\mathbb{F}_t)$$

where, in any interval $[1+2(i-1)u, 1+2iu]$, $d(z,\mathbb{F}_t)$ has the shape of a triangle of height u and base $2u$. Hence

$$\int_1^2 d(z,\mathbb{F}_t) = \sum_{i=1}^{2^{t-1}} u^2 = \frac{u}{2}$$

and since $d(z,\mathbb{F}_t)/2 \le d(z,\mathbb{F}_t)/|z| \le d(z,\mathbb{F}_t)$ for $z \in [1,2]$ we obtain

$$\frac{u}{4} = \int_1^2 \frac{d(z,\mathbb{F}_t)}{2}\mathrm{d}z \le \eta(\mathbb{F}_t) \le \int_1^2 d(z,\mathbb{F}_t)\mathrm{d}z = \frac{u}{2}. \tag{3.13}$$

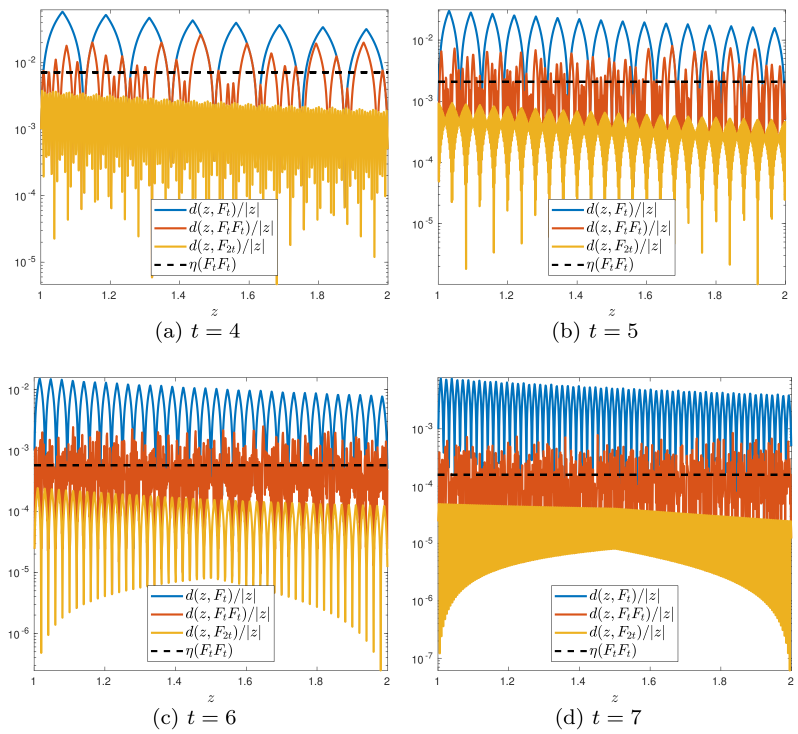


Fig. 3.3: Relative errors $d(z, \mathbb{F}_t)/|z|$ and $d(z, \mathbb{F}_t\mathbb{F}_t)/|z|$ for $t = 4, 5, 6, 7$.

As illustrated in Figure 3.1, the elements of $\mathbb{F}_t\mathbb{F}_t$ are *not* uniformly spaced, and so the concept of average relative error takes its full meaning as it can be quite different from the worst case one. We illustrate this by plotting in Figure 3.3 the relative errors $d(z, \mathbb{F}_t)/|z|$ and $d(z, \mathbb{F}_t\mathbb{F}_t)/|z|$ for $z \in [1, 2]$ and $t = 4$. The average relative error, given by the area under the curves, is visibly much better for $\mathbb{F}_t\mathbb{F}_t$ than for $\mathbb{F}_t$. We now quantify this more precisely.

Lemma 3.5. *Consider a set $\mathbb{S} \subset \mathbb{R}$ such that $\mathbb{S} \cap [1, 2]$ is finite and non-empty. Denoting $\widehat{z}_i$ the elements of $\mathbb{S} \cap [1, 2]$ in increasing order, we have*

$$\eta(\mathbb{S}) = \sum_i \left(\widehat{z}_i \log \widehat{z}_i + \widehat{z}_{i+1} \log \widehat{z}_{i+1} - 2m_i \log m_i\right). \tag{3.14}$$

Proof. We can reuse the expression (3.11) from the proof of Corollary 3.4 to get

$$\begin{aligned}\int_{\widehat{z}_i}^{\widehat{z}_{i+1}} \frac{d(z,\cdot)}{|z|} &= m_i - \widehat{z}_i - \widehat{z}_i[\log z]_{\widehat{z}_i}^{m_i} + \widehat{z}_{i+1}[\log z]_{m_i}^{\widehat{z}_{i+1}} - (\widehat{z}_{i+1} - m_i)\\ &= \widehat{z}_i \log \widehat{z}_i - \widehat{z}_i \log m_i + \widehat{z}_{i+1} \log \widehat{z}_{i+1} - \widehat{z}_{i+1} \log m_i\\ &= \widehat{z}_i \log \widehat{z}_i + \widehat{z}_{i+1} \log \widehat{z}_{i+1} - 2m_i \log m_i. \qquad \square\end{aligned}$$

We use this formula for computing $\eta(\mathbb{S})$ for varying t and plot the result in Figure 3.2 (right). Using a linear fit, we observe the average relative error $\eta(\mathbb{F}_t\mathbb{F}_t)$ to behave approximately as $O(2^{-1.9t})$, which is even better than the $O(2^{-1.6t})$ behavior of the worst case one $\epsilon(\mathbb{F}_t\mathbb{F}_t)$. This confirms that the non-uniform spacing of the elements of $\mathbb{F}_t\mathbb{F}_t \cap [1,2]$ can have a significant impact on the average relative error when quantizing on this set.

In conclusion, we have derived lower and upper bounds on the worst case quantization accuracy over rank-one matrices $d(\Sigma(\mathbb{R}), \Sigma(\mathbb{F}_t))$ in Lemma 3.1. We observe that these bounds are independent of m, n. Moreover, the lower bound $\epsilon(\mathbb{F}_t\mathbb{F}_t)$ empirically behaves as $O(2^{-1.6t})$ (see Figure 3.2) while the upper bound behaves as $O(2^{-t})$. We may thus hope to achieve an optimal quantization error that behaves as $\epsilon(\mathbb{F}_t\mathbb{F}_t)$, rather than the worst case RTN error $2v_t + v_t^2$. There is no guarantee that this is possible in general. Indeed, we will show in Subsection 6.2 that the empirical worst case quantization error obtained with the optimal algorithm is close to the lower bound for small values of m, n, whereas it approaches the upper bound for larger values. Theoretically characterizing $\epsilon(\Sigma_1^{m\times n}(\mathbb{R}), \Sigma_1^{m\times n}(\mathbb{F}_t))$ is an interesting challenge left to future work. However, we will further investigate whether one of this bounds is sharp (or at least has a sharp behavior with respect to t) empirically in Subsection 6.2.

4. Existence and characterization of an optimal quantization.

In this section, we focus on the optimal solution of problem (1.1). Assuming the problem has a solution, finding it is not straightforward. First of all, we can note that there is no obvious way to separate the global problem into smaller independent subproblems: for example, at first sight we may think of using the equality

$$\|xy^\top - \widehat{x}\widehat{y}^\top\|^2 = \sum_{i=1}^{n} \|x_i y - \widehat{x}_i \widehat{y}\|^2$$

and exploiting the analysis of the previous section to optimally quantize the elements of $x_i y$ in $\mathbb{F}_t\mathbb{F}_t$, but the issue is that the optimal $\widehat{y}^*$ for a given optimal $\widehat{x}_i^*$ is in general different from the optimal $\widehat{y}^*$ for an optimal $\widehat{x}_j^*$, $j \neq i$. Thus we have no choice but to tackle the global problem.

A first possibility is to use a brute-force algorithm that enumerates all possible solutions. Even assuming that the search space could be restricted to a finite set, e.g., $\mathbb{F}_t \cap [1,2]$, the main challenge remains to enumerate all possible solutions. This set contains 2^t elements, and we need to test each of these values for each element of $\widehat{x}$, $\widehat{y}$, which yields a $O(2^{t(m+n)})$ complexity, which is exponential in $m+n$ and thus clearly intractable except for very small problems. We will then study the properties of this problem to devise a more clever solution method.

When x or y is zero the existence of a minimizing pair is trivial. The extension of this result to the case of nonzero x, y is based on the following key observation.

LEMMA 4.1. *Let $x \in \mathbb{R}^m$, $y \in \mathbb{R}^n$. Considering any $\widehat{x} \in \mathbb{F}_t^m, \widehat{y} \in \mathbb{F}_t^n$, we have*

$$C_{x,y}(\widehat{x}, \widehat{y}) \geq C_{x,y}(\widehat{x}', \widehat{y}), \quad \forall \widehat{x}' \in \operatorname{round}(\lambda x) \subset \mathbb{F}_t^m \tag{4.1}$$

where the right hand side is constant over all possible $\widehat{x}' \in \operatorname{round}(\lambda x)$ and

$$\lambda = \lambda_y(\widehat{y}) := \begin{cases} \frac{y^\top \widehat{y}}{\|\widehat{y}\|^2}, & \text{if } \widehat{y} \neq 0 \\ 0, & \text{otherwise.} \end{cases}$$

Proof. If $\widehat{y} = 0$ then $\lambda = 0$ thus $\operatorname{round}(\lambda x) = \{0_m\} =: \{\widehat{x}'\} \subset \mathbb{F}_t^m$, hence the result as $C_{x,y}(\widehat{x},\widehat{y}) = \|xy^\top\|^2 = C_{x,y}(\widehat{x}',\widehat{y})$. When $\widehat{y} \neq 0$, for $i \in [\![m]\!]$

$$\begin{aligned}\|x_i y - \widehat{x}_i \widehat{y}\|^2 \geq \inf_{w\in\mathbb{F}_t} \|x_i y - w\widehat{y}\|^2 &= \inf_{w\in\mathbb{F}_t} \left\{\|x_i y\|^2 + w^2\|\widehat{y}\|^2 - 2w x_i y^\top \widehat{y}\right\} \\ &= \|x_i y\|^2 + \|\widehat{y}\|^2 \inf_{w\in\mathbb{F}_t} \left\{w^2 - 2\lambda x_i w\right\} \\ &= \|x_i y\|^2 - \|\widehat{y}\|^2\lambda^2 + \|\widehat{y}\|^2 \underbrace{\inf_{w\in\mathbb{F}_t} (w-\lambda x_i)^2}_{d^2(\lambda x_i,\mathbb{F}_t)}.\end{aligned}$$

The infimum in the right hand side is achieved at each $w \in \text{round}(\lambda x_i)$, hence each $\widehat{x}' \in \text{round}(\lambda x) \subset \mathbb{F}_t^m$ satisfies $\|x_i y - \widehat{x}_i \widehat{y}\|^2 \geq \|x_i y - \widehat{x}'_i \widehat{y}\|^2$, $i \in [\![m]\!]$, and

$$C_{x,y}(\widehat{x},\widehat{y}) = \sum_{i=1}^m \|x_i y^\top - \widehat{x}_i \widehat{y}^\top\|^2 \geq \sum_{i=1}^m \|x_i y^\top - \widehat{x}'_i \widehat{y}^\top\|^2 = C_{x,y}(\widehat{x}',\widehat{y}).$$

The right hand side $C_{x,y}(\widehat{x}',\widehat{y})$ is constant over all possible $\widehat{x}' \in \text{round}(\lambda x)$. ∎

This result tells us that once $\widehat{y}$ is fixed, a class of vectors $\widehat{x}$ that minimize the function $C_{x,y}$ can be identified simply by an optimal scaling parameter λ, which is determined by $y, \widehat{y}$. Note that Lemma 4.1 remains true if we exchange the roles of x and y. This observation yields the following key result.

Lemma 4.2. *Let $x \in \mathbb{R}^m$, $y \in \mathbb{R}^n$. We have*

$$\inf_{\widehat{x}\in\mathbb{F}_t^m, \widehat{y}\in\mathbb{F}_t^n} C_{x,y}(\widehat{x},\widehat{y}) = \inf_{\lambda\in\mathbb{R}} f(\lambda), \tag{4.2}$$

with

$$f(\lambda) := \max_{\widehat{x}\in\text{round}(\lambda x)} C_{x,y}\big(\widehat{x}, \text{round}(\mu(\widehat{x})y)\big) \tag{4.3}$$

where

$$\mu(\widehat{x}) := \begin{cases} \frac{x^\top \widehat{x}}{\|\widehat{x}\|^2}, & \text{if } \widehat{x} \neq 0 \\ 0, & \text{otherwise}\end{cases} \tag{4.4}$$

and $C_{x,y}\big(\widehat{x}, \text{round}(\mu(\widehat{x})y)\big)$ denotes the value of $C_{x,y}\big(\widehat{x},\widehat{y}\big)$ for $\widehat{y} \in \text{round}(\mu(\widehat{x})y)$, which does not depend on the choice of $\widehat{y}$ among possible ties.

Proof. Consider any $\widehat{x} \in \mathbb{F}_t^m, \widehat{y} \in \mathbb{F}_t^n$, $\lambda = \lambda_y(\widehat{y})$ defined in (4.1), and any $\widehat{x}' \in \text{round}(\lambda x)$. By Lemma 4.1 we have

$$C_{x,y}(\widehat{x},\widehat{y}) \geq C_{x,y}(\widehat{x}',\widehat{y}). \tag{4.5}$$

Since $\widehat{x}' \in \mathbb{F}_t^m$, by the analog of Lemma 4.1 where the role of rows/columns is exchanged, we obtain with $\mu(\widehat{x}') := \lambda_x(\widehat{x}')$ (this matches definition (4.4))

$$C_{x,y}(\widehat{x}',\widehat{y}) \geq C_{x,y}(\widehat{x}',\widehat{y}'), \quad \forall \widehat{y}' \in \text{round}(\mu(\widehat{x}')y) \subset \mathbb{F}_t^n. \tag{4.6}$$

By (4.5) and (4.6) we get $C_{x,y}(\widehat{x},\widehat{y}) \geq C_{x,y}(\widehat{x}',\widehat{y}')$ for every $\widehat{x}' \in \text{round}(\lambda x)$ and $\widehat{y}' \in \text{round}(\mu(\widehat{x}')y)$, hence $C_{x,y}(\widehat{x},\widehat{y}) \geq f(\lambda)$. This holds for each $\widehat{x} \in \mathbb{F}_t^m, \widehat{y} \in \mathbb{F}_t^n$, hence we obtain

$$\inf_{\widehat{x}\in\mathbb{F}_t^m, \widehat{y}\in\mathbb{F}_t^n} C_{x,y}(\widehat{x},\widehat{y}) \geq \inf_{\lambda\in\mathbb{R}} f(\lambda). \tag{4.7}$$

For any λ, since $\text{round}(\lambda x) \subset \mathbb{F}_t^m$ and $\text{round}(\mu(\widehat{x})y) \subset \mathbb{F}_t^n$, $f(\lambda)$ is lower-bounded by the left hand side in (4.7). Thus (4.7) is indeed an equality. ∎

Lemma 4.2 is a key result because it allows us to reduce the problem to a scalar one: it suffices to find an optimum λ^* of $f(\lambda)$, assuming it exists. It therefore only remains to prove that such an optimum exists. To do so, we first show that f is invariant by multiplication by 2 and sign flip in Lemma 4.3. This allows us to restrict the search for an optimum to the interval $[1, 2)$.

Lemma 4.3. *Let $f(\lambda)$ be defined as in (4.3); f is invariant by multiplication by ± 2: $f(\pm 2\lambda) = f(\lambda)$ for each $\lambda \in \mathbb{R}$.*

Proof. For any $w \in \mathbb{F}_t^m$ we have $\mu(\pm 2w) = \pm\mu(w)/2$ and $\text{round}(\pm 2w) = \pm 2\text{round}(w)$. Defining

$$g(w) = C_{x,y}\big(w, \text{round}(\mu(w)y)\big) \tag{4.8}$$

we have

$$\begin{aligned} g(\pm 2w) &= \|xy^\top - (\pm 2w)\text{round}\big(\mu(\pm 2w)y\big)\|^2 \\ &= \|xy^\top - (\pm 2w)\text{round}\big(\pm\mu(w)y/2\big)\|^2 \\ &= \|xy^\top - w\text{round}\big(\mu(w)y\big)\|^2 = g(w). \end{aligned}$$

Therefore,

$$f(\pm 2\lambda) = \max_{\widehat{x} \in \pm 2\text{round}(\lambda x)} g(\widehat{x}) = \max_{\widehat{x}' \in \text{round}(\lambda x)} g(\pm 2\widehat{x}') = \max_{\widehat{x}' \in \text{round}(\lambda x)} g(\widehat{x}') = f(\lambda). \qquad \square$$

Finally we show that, as illustrated on Figure 4.1 (left), the function $f(\lambda)$ takes a finite number of values in $[1, 2)$, therefore proving the existence of an optimum. To do so, we need to study the function $\lambda \mapsto \text{round}(\lambda x)$ for a given x and to characterize its *breakpoints*.

Definition 4.4 (Breakpoints). *Let $x \in \mathbb{R}^m$. A scalar $\lambda \in \mathbb{R}$ is a breakpoint of the function $\lambda \mapsto \text{round}(\lambda x)$ if there does not exist a neighborhood in which $\text{round}(\lambda x)$ is constant.*

Thus, breakpoints are characterized by the existence of at least one coordinate i such that $\text{round}(\lambda x_i)$ corresponds to a tie. For any $x \in \mathbb{R}^m$, the function $\lambda \mapsto \text{round}(\lambda x)$ is piecewise constant, and has finitely many breakpoints in the open interval $(1, 2)$. We denote them λ_j, $1 \le j \le J$ in increasing order.

We are now ready to state our main result.

Theorem 4.5. *Consider nonzero $x \in \mathbb{R}^m$, $y \in \mathbb{R}^n$, with $m, n \ge 1$, and $t \ge 1$. Denote $\lambda_0 := 1 < \lambda_1 < \ldots < \lambda_J < 2 =: \lambda_{J+1}$ with λ_j, $1 \le j \le J$ the breakpoints of $\lambda \in (1, 2) \mapsto \text{round}(\lambda x)$, and $\lambda_{j+1/2} := (\lambda_j + \lambda_{j+1})/2$, $0 \le j \le J$. Problem (1.1) admits an optimum $\widehat{x}, \widehat{y}$ such that*

$$\widehat{x} = \text{round}(\lambda^* x) \quad \textit{with } \lambda^* = \lambda_{j^*+1/2} \textit{ for some } 0 \le j^* \le J, \tag{4.9}$$

$$\widehat{y} \in \text{round}(\mu^* y), \quad \textit{with } \mu^* = \begin{cases} \frac{x^\top \widehat{x}}{\|\widehat{x}\|^2}, & \textit{if } \widehat{x} \ne 0 \\ 0, & \textit{otherwise.} \end{cases} \tag{4.10}$$

Moreover, any $\lambda \in (\lambda_{j^*}, \lambda_{j^*+1})$ yields the same $\widehat{x}, \widehat{y}$ and is therefore also optimal.

Proof. By Lemma 4.2, it is sufficient to show that the optimum of $f(\lambda)$ is achieved at some $\lambda^* \in (1, 2)$. To do so we progressively restrict the search space. To begin with, we can exclude $\lambda = 0$ from the search space: since $xy^\top \ne 0$ there are indices i, j such that $|x_i y_j| > 0$ and $\widehat{x} \in \mathbb{F}_t^n$, $\widehat{y} \in \mathbb{F}_t^m$ with all entries set to zero except $\widehat{x}_i = \texttt{sign}(x_i)\tilde{x}_i$,

$\widehat{y}_j = \texttt{sign}(y_j)\tilde{y}_j$, $0 < \tilde{x}_i \le |x_i|$ and $0 < \tilde{y}_j \le |y_j|$; we have $|x_i y_j - \tilde{x}_i \tilde{y}_j| = |x_i y_j| - \tilde{x}_i \tilde{y}_j < |x_i y_j|$ while for all pairs $(i', j') \ne (i, j)$ $|x_i y_j - \tilde{x}_i \tilde{y}_j| = |x_i y_j|$, hence

$$\inf_{\lambda \in \mathbb{R}} f(\lambda) \overset{(4.2)}{=} C_{x,y}(\widehat{x}, \widehat{y}) = \|xy^\top - \widehat{x}\widehat{y}^\top\|^2 < \|xy^\top\|^2 = f(0).$$

Moreover, by Lemma 4.3 we can restrict the search to $[1, 2)$ and so

$$\inf_{\lambda \in \mathbb{R}} f(\lambda) = \inf_{\lambda \in [1,2)} f(\lambda). \tag{4.11}$$

Finally, the function $\lambda \mapsto \text{round}(\lambda x)$ (and therefore $\lambda \mapsto f(\lambda)$) is piecewise constant, with finitely many breakpoints λ_j, $1 \le j \le J$ within the interval $(1, 2)$. Setting $\lambda_0 = 1$, $\lambda_{J+1} = 2$ we consider the partition $[1, 2) = \cup_{j=0}^{J} [\lambda_j, \lambda_{j+1})$ and vectors $\widehat{x}_j$, $0 \le j \le J$ such that $\text{round}(\lambda x) = \{\widehat{x}_j\}$ for every $\lambda \in (\lambda_j, \lambda_{j+1})$. Considering arbitrary interior points $\lambda_{j+1/2} \in (\lambda_j, \lambda_{j+1})$ in each interval (for example $\lambda_{j+1/2} := (\lambda_j + \lambda_{j+1})/2$), this yields finitely many values $f(\lambda_{j+1/2}) = g(\widehat{x}_j)$, with g defined in (4.8). For $1 \le j \le J$ the breakpoint $\lambda_j \in (1, 2)$ corresponds to a tie hence $\{\widehat{x}_{j-1}, \widehat{x}_j\} \subset \text{round}(\lambda_j x)$ and therefore $f(\lambda_j) \ge \max(g(\widehat{x}_{j-1}), g(\widehat{x}_j)) = \max(f(\lambda_{j-1/2}), f(\lambda_{j+1/2}))$. Since we are looking for a minimizer of $f(\lambda)$ this allows us to exclude the breakpoints from the search space. Similarly, $\lambda_0 = 1$ can be excluded from the search space if it corresponds to a breakpoint, and we can also exclude it if it not a breakpoint since we then have $f(\lambda_0) = f(\lambda_{1/2})$. Overall we obtain $\inf_{\lambda \in [1,2)} f(\lambda) = \min_{0 \le j \le J} f(\lambda_{j+1/2})$. As a last step, observe that for every j, $\text{round}(\lambda_{j+1/2} x)$ is a singleton hence we can indeed write an optimal $\widehat{x}$ as in (4.9). An optimal $\widehat{y}$ can be deduced from Lemma 4.1 and is thus expressed as in (4.10). ∎

This theorem tells us that to find the optimum of problem (1.1) we just need to find the optimal scaling parameter λ^*, that can be found through an extensive search based on the finitely many breakpoints of the function $\lambda \in (1, 2) \mapsto \text{round}(\lambda x)$. Once λ^* has been determined, both $\widehat{x}$ and $\widehat{y}$ can easily be found. We have thus reduced a problem with $m + n$ variables to a problem with a single variable λ, whose feasible values belong to a finite set.

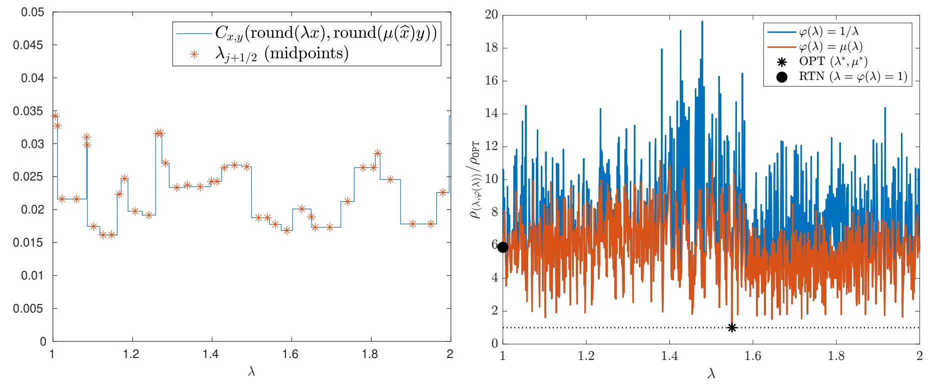


Fig. 4.1: Left: typical shape of $f(\lambda) = C_{x,y}(\widehat{x}(\lambda), \widehat{y}(\lambda))$, for $x, y \in \mathbb{R}^n$ $(n = 5)$ drawn with random uniform $[0, 1]$ entries and $t = 4$, where $\widehat{x}(\lambda) := \text{round}(\lambda x)$, $\widehat{y}(\lambda) := \text{round}(\mu(\widehat{x}(\lambda))y)$ and $\mu(\cdot)$ as in (4.4). Right: plot of $\rho_{(\lambda,\varphi(\lambda))}/\rho_{\text{OPT}}$ (see (4.12), (4.13)) for $\varphi(\lambda) = 1/\lambda$ and $\varphi(\lambda) = \mu(\lambda) := \mu(\widehat{x}(\lambda))$, for $t = 11$ and $n = 16$.

Importantly, while in exact arithmetic we have the scaling invariance $xy^\top = \lambda x(\frac{1}{\lambda}y)^\top$, in finite precision arithmetic, the quantization $\text{round}(\lambda x)\text{round}(\mu y)^\top$ has an accuracy that strongly depends on the two scalings λ and μ, and, crucially, the optimal μ^* is close to, *but in general not equal to*, $1/\lambda^*$. This is illustrated numerically in Figure 4.1 (right), which plots $\rho_{(\lambda,\varphi(\lambda))}/\rho_{\mathrm{OPT}}$ with

$$\rho_{\mathrm{OPT}} \coloneqq \|xy^\top - \widehat{x}\widehat{y}^\top\|/\|xy^\top\|, \tag{4.12}$$

$$\rho_{(\lambda,\varphi(\lambda))} \coloneqq \frac{\|xy^\top - \mathrm{round}(\lambda x)\mathrm{round}(\varphi(\lambda)y)^\top\|}{\|xy^\top\|}, \tag{4.13}$$

for $\varphi(\lambda) = 1/\lambda$ and $\varphi(\lambda) = \mu(\lambda) \coloneqq \frac{x^\top \widehat{x}(\lambda)}{\|\widehat{x}(\lambda)\|^2}$. The figure shows that using $\varphi(\lambda) = \mu(\lambda)$ provides much better quantization errors. Moreover, the figure also suggests that finding the optimal λ^* by simply sampling the interval $[1, 2]$ would require a very fine sampling, because the relative error displays large variations even in the neighborhood of the optimum.

5. Optimal quantization algorithm.

There remains to understand whether one can find a tractable algorithm to actually compute the optimal scalar λ^* for each instance (x, y).

As shown on Figure 4.1 (left), the behavior of $\lambda \mapsto f(\lambda)$ on $[1, 2)$ is not regular, and since it is piecewise constant one cannot expect to rely on gradient descent to find its minimizer. Fortunately, leveraging the analysis of the previous section, we can design an algorithm of controlled complexity, based on the explicit enumeration of the breakpoints of $\lambda \in (1, 2) \mapsto \mathrm{round}(\lambda x)$, rather than on enumerating all possible $\widehat{x}, \widehat{y}$. This requires an explicit characterization of the breakpoints, which the following lemma provides.

Lemma 5.1. *Given $x \in \mathbb{R}^m$ the set of breakpoints of $\lambda \in (1, 2) \mapsto \mathrm{round}(\lambda x)$ is exactly $\mathbb{B}(x) \coloneqq \cup_{i:x_i \neq 0} B(\bar{x}_i)$ where for any scalar $z \in [1, 2)$ we define*

$$B(z) \coloneqq \left\{ (k + 1/2)2^{e-t}\frac{1}{z} : k \in [\![2^{t-1}, 2^t - 1]\!],\ e \in \{1, 2\} \right\} \tag{5.1}$$

and if $x_i \neq 0$ we define $\bar{x}_i \coloneqq 2^{p_i}|x_i|$ with $p_i \in \mathbb{Z}$ such that $\bar{x}_i \in [1, 2)$.

Proof. First we show that $B(z)$ is the set of breakpoints of $\lambda \in (1, 2) \mapsto \mathrm{round}(\lambda z)$. For this, observe that a breakpoint is characterized by the fact that λz is equidistant to two points of $\mathbb{F}_t$. Moreover, for $1 < \lambda < 2$ and $1 \leq z < 2$ we have $1 < \lambda z < 4$, and if $\lambda z = 2$ then λ is not a breakpoint, hence all considered breakpoints satisfy $\lambda z \in (1, 2) \cup (2, 4)$. If λ is breakpoint such that $\lambda z \in (1, 2)$ then the points surrounding λz in $\mathbb{F}_t$ (cf (2.1)) read $k2^{1-t}$ and $(k + 1)2^{1-t}$ with $k \in [\![2^{t-1}, 2^t - 1]\!]$, hence $\lambda z = (k + 1/2)2^{1-t}$ and $\lambda = (k + 1/2)2^{1-t}/z$. Vice-versa, this expression yields a breakpoint such that $\lambda z \in (1, 2)$. When $\lambda z \in (2, 4)$, we repeat the argument with $\lambda' = \lambda/2$ to obtain $\lambda = (k + 1/2)2^{2-t}/z$.

Now, if $x_i \neq 0$, observe that $\lambda \in (1, 2)$ is a breakpoint of $\lambda \mapsto \mathrm{round}(\lambda x_i)$ if, and only if, it is a breakpoint of $\lambda \mapsto \mathrm{round}(\lambda \bar{x}_i)$; if $x_i = 0$, $\lambda \mapsto \mathrm{round}(\lambda x_i)$ has no breakpoint. The set of breakpoints of $\lambda \mapsto \mathrm{round}(\lambda x)$ is thus $\mathbb{B}(x)$. ∎

As an immediate corollary we can bound the number of breakpoints (in the result below, $\|x\|_0$ denotes the number of nonzero elements of x).

Corollary 5.2. *The number of breakpoints $\#\mathbb{B}(x)$ of the function $\lambda \in (1, 2) \mapsto \mathrm{round}(\lambda x)$ is bounded by $\|x\|_0 2^t \leq m2^t$.*

Using this characterization of breakpoints, we outline in Algorithm 5.1 a method to solve (1.1). The algorithm builds the set of breakpoints $\mathbb{B}(x)$, sorts it in increasing

Algorithm 5.1 An algorithm to solve (1.1).

```
Input: $t \geq 1$ an integer, $x \in \mathbb{R}^m, y \in \mathbb{R}^n$.
Output: $\widehat{x}^* \in \mathbb{F}_t^m, \widehat{y}^* \in \mathbb{F}_t^n$ solutions to (1.1).
 1: Initialize $\widehat{x}^* \leftarrow 0$, $\widehat{y}^* \leftarrow 0$
 2: if $x = 0$ or $y = 0$ then
 3:     exit
 4: end if
 5: $\mathbb{B} \leftarrow \mathbb{B}(x)$ as defined by Lemma 5.1.
 6: Sort $\mathbb{B}$ in increasing order to obtain $\lambda_j$, $1 \leq j \leq J := \#\mathbb{B}$, $\lambda_0 \leftarrow 1$, $\lambda_{J+1} \leftarrow 2$.
 7: for $j = 1$ to $J + 1$ do
 8:     $\lambda \leftarrow (\lambda_{j-1} + \lambda_j)/2$
 9:     $\widehat{x} \leftarrow \mathrm{round}(\lambda x)$          ▷ There is never a tie here, and $\widehat{x} \neq 0$
10:     $\mu \leftarrow x^\top \widehat{x}/\|\widehat{x}\|^2$
11:     $\widehat{y} \leftarrow \mathrm{round}(\mu y)$          ▷ In case of tie, choose arbitrarily
12:     if $C_{x,y}(\widehat{x}, \widehat{y}) < C_{x,y}(\widehat{x}^*, \widehat{y}^*)$ then
13:         $\widehat{x}^* \leftarrow \widehat{x}$, $\widehat{y}^* \leftarrow \widehat{y}$
14:     end if
15: end for
```

order, and finally enumerates it to test each midpoint and find the optimal one. Some comments are in order:

- line 8 ensures that λ is *not* a breakpoint, hence $\mathrm{round}(\lambda x)$ is a singleton and $\widehat{x}$ is well-defined in line 9;
- since line 9 can only be reached when $x \neq 0$ and with $\lambda > 1$, this ensures that it also yields $\widehat{x} \neq 0$; hence the expression in line 10 is well-defined;
- an arbitrary choice of $\widehat{y} \in \mathrm{round}(\mu y)$, $\mu = x^\top \widehat{x}/\|\widehat{x}\|^2$ in line 11 is possible as the value of $C_{x,y}(\widehat{x}, \widehat{y}')$ is the same for every $\widehat{y}' \in \mathrm{round}(\mu y)$ (Lemma 4.2).

We next discuss the space and time cost of Algorithm 5.1. Building and sorting $\mathbb{B}(x)$ has a space cost in $O(m2^t)$ and a time cost in $O(m2^t \log m)$. A naive implementation to compute $C_{x,y}(\widehat{x}, \widehat{y})$ in line 12 at each iteration would be to explicitly build the $m \times n$ matrix $xy^\top - \widehat{x}\widehat{y}^\top$, which would cost $O(mn)$. However since we are dealing with rank-one matrices, denoting $\langle A, B\rangle_F := \mathtt{trace}(A^\top B)$ the Frobenius inner-product between $m \times n$ matrices, we have

$$\begin{aligned} C_{x,y}(\widehat{x}, \widehat{y}) &= \|xy^\top - \widehat{x}\widehat{y}^\top\|^2 = \|xy^\top\|^2 + \|\widehat{x}\widehat{y}^\top\|^2 - 2\langle xy^\top, \widehat{x}\widehat{y}^\top\rangle_F && (5.2)\\ &= \|x\|^2\|y\|^2 + \|\widehat{x}\|^2\|\widehat{y}\|^2 - 2(x^\top\widehat{x})(y^\top\widehat{y}). && (5.3)\end{aligned}$$

This allows us to compute $C_{x,y}(\widehat{x}, \widehat{y})$ with cost $O(m + n)$. It is easy to check that the rest of the operations performed at each iteration of Algorithm 5.1 also have a cost in $O(m + n)$. Taking into account the loop over $J = \#\mathbb{B}(x)$ breakpoints, Algorithm 5.1 therefore has a total time cost in $O((m+n)\#\mathbb{B}(x)) = O((m+n)m2^t)$ (the initial sorting of $\mathbb{B}(x)$ has negligible cost). Since we can also reverse the respective roles of x and y by looping on $\mathbb{B}(y)$, a time cost in $O((m + n)n2^t)$ can also be achieved, so choosing the best of both yields $O((m + n)\min(m, n)2^t) = O(\max(m, n)\min(m, n)2^t) = O(mn2^t)$, and so we have proved the following result.

THEOREM 5.3. *Algorithm 5.1 solves problem (1.1) in $O(\min(m, n)2^t)$ space and in $O(mn2^t)$ time.*

Note that additional optimizations could reduce the constant in the $O(mn2^t)$

time cost. In particular, it is likely that from one breakpoint to the next, only a few (possibly only one) coordinates of x change. With some bookkeeping it is thus possible to cheaply compute $\widehat{x}$ and μ in lines 9 and 10, although $\widehat{y}$ must still be computed from scratch.

Since it explicitly builds and stores the set of breakpoints, Algorithm 5.1 requires $O(\min(m,n)2^t)$ space. If space complexity is an issue, this can be avoided by *implicitly* enumerating the breakpoints instead, using their characterization in terms of triplets $(k_i, e_i, \bar{x}_i)$ from Lemma 5.1. At each iteration, the next breakpoint can simply be found by finding, among the m coordinates of x, which one is associated with the smallest remaining breakpoint. This is feasible with a cost in $O(m \log m)$ at the first iteration via sorting, and with a cost in $O(1)$ at subsequent iterations using an adequate data structure to maintain a list of m sorted numbers through removal/insertion. Alternatively this can be done with a cost in $O(m)$ at each iteration (via a simple loop to find the minimum over m unsorted numbers), a cost that remains dominated by the cost in $O(m+n)$ to compute $C_{x,y}$ at each iteration. Overall, if space complexity is an issue (typically for large t), an implementation of Algorithm 5.1 with reduced space complexity $O(\min(m,n)t)$ and maintained time complexity $O(mn2^t)$ can be achieved.

Remark 5.1. *Given the rising interest in mixed precision arithmetic [8], one may wonder whether Algorithm 5.1 can be adapted in such a context. Considering the variant of (1.1) where we seek $\widehat{x} \in \mathbb{F}_{t_x}^m$, $\widehat{y} \in \mathbb{F}_{t_y}^n$, with[1] $t_x, t_y \in \mathbb{N} \cup \{\infty\}$ two precision levels, it is easy to check that an adaptation of the whole analysis leads to a variant of Algorithm 5.1, where $\mathbb{B}(x)$ is computed in line 5 from Lemma 5.1 with $t = t_x$, and the rounding operation of line 9 (resp. of line 11) is performed with round_{t_x} (resp. with round_{t_y}). With the convention $\mathbb{F}_\infty := \mathbb{R}$ this enables a variant without quantization constraint on one of the factors. This will be used in Algorithms 7.1 and 7.3.*

6. Numerical validation of the optimal quantization algorithm.

In this section we validate numerically the performance of the optimal quantization algorithm outlined in Algorithm 5.1, both in terms of accuracy of the quantization and in terms of computational cost. We compare the solution of the optimal algorithm $\widehat{x}, \widehat{y}$ with the RTN baseline solution $\mathrm{round}(x), \mathrm{round}(y)$. The code to reproduce the experiments in this section is available online[2].

6.1. Experimental protocol.

In all the experiments presented in this section we consider 100 randomly generated couples $x, y \in \mathbb{R}^n$. The entries of x and y are drawn from the uniform $[0,1]$ distribution and multiplied by random exponents values ranging from 10^{-2} to 10^2. We define

$$\rho_{\mathrm{OPT}} := \|xy^\top - \widehat{x}\widehat{y}^\top\| / \|xy^\top\|,$$
$$\rho_{\mathrm{RTN}} := \|xy^\top - \mathrm{round}(x)\mathrm{round}(y)^\top\| / \|xy^\top\|.$$

We consider different values of $n \le 1024$ and $t \le 11$. We do not test any larger values of t (such as 32-bit) as our main interest is in low precision quantization applications.

6.2. Worst case behavior.

We begin by analyzing the *empirical* worst case quantization error obtained with the optimal algorithm. The worst case behavior is interesting in light of inequality (3.8) from Lemma 3.1. Indeed, to elucidate the behavior of the true worst case quantization error $\epsilon(\Sigma(\mathbb{R}), \Sigma(\mathbb{F}_t))$, one idea is to investigate empirically the worst case error obtained over many random inputs. Figure 6.1 shows

[1] to avoid any ambiguity, in this paper we denote $\mathbb{N} = \{1, 2, \dots\}$ the set of positive integers.
[2] https://perso.ens-lyon.fr/elisa.riccietti/code.php, https://inria.hal.science/hal-04124171

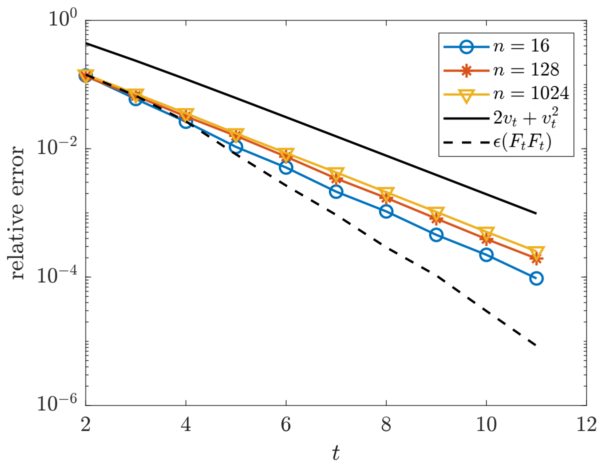


Fig. 6.1: Worst relative error ρ_{OPT} over 100 randomly chosen couples $(x, y) \in \mathbb{R}^{n\times n}$ for $n = 16, 128, 1024$ and lower and upper bound of (3.7).

this experiment for different values of n and t. We recall from (3.8) that the worst case error falls between $\epsilon(\mathbb{F}_t\mathbb{F}_t)$, which behaves as $O(2^{-1.6t})$, and $2v_t + v_t^2$, which behaves as $O(2^{-t})$. Empirically, we confirm these bounds and find the empirical worst case error to be somewhere between them. Interestingly, while both the lower and upper bounds do not depend on m, n, the empirical worst case error clearly increases with m, n, getting closer and closer to its upper bound while still remaining an order of magnitude smaller for $m = n = 1024$. Moreover, its behavior as a function of t, for m, n fixed, follows that of its upper bound, $O(2^{-t})$.

6.3. Average case behavior. We now turn to the empirical average case optimal quantization error, which is more representative of the typical accuracy gains that can be achieved by the optimal algorithm over the RTN baseline.

To gain further insight on the behavior of optimal quantization error ρ_{OPT} compared with the RTN baseline error ρ_{RTN}, we perform a scatter plot of both errors in Figure 6.2 for various values of n and t. The plot reveals two interesting trends as n increases: the points become less dispersed, and closer to the diagonal (where $\rho_{\mathrm{OPT}} = \rho_{\mathrm{RTN}}$, that is, when no accuracy gain is achieved by the optimal algorithm). These trends seem to hold regardless of t, although the average accuracy gains seem larger as t increases.

To quantify more precisely the gains that can be achieved with the optimal algorithm, we define the accuracy gain (the larger, the better) as the percentage

$$100 \times \left(1 - \frac{\rho_{\mathrm{OPT}}}{\rho_{\mathrm{RTN}}}\right) \tag{6.1}$$

and we plot in Figure 6.3 some boxplots of this measure for different values of t and n. We remind that the red line marks the median of the distribution, the bottom and top edges of the boxes indicate the 25th and the 75th percentiles, respectively, and the whiskers extend to the most extreme data points. As expected, the figure shows that the gains decrease as the dimension n increases, and also seem to slightly increase as the number of bits t increases. Overall, significant gains can be achieved; for example, for $t = 11$ and $n = 128$, the gain is concentrated around 40% reduction of the error in

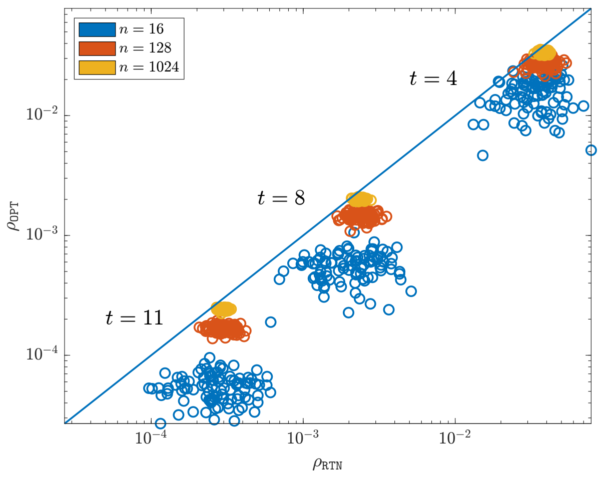


Fig. 6.2: Scatter plot of ρ_{OPT}, ρ_{RTN} for 100 randomly chosen couples $(x, y) \in \mathbb{R}^{n \times n}$ for different values of n and t.

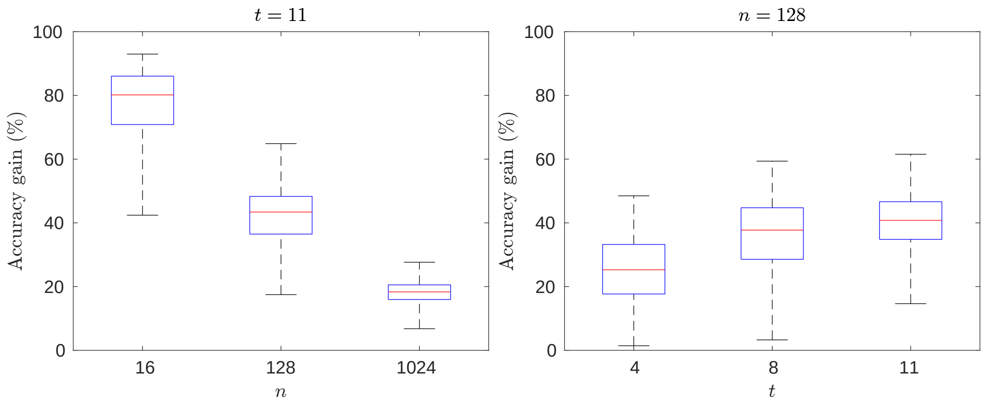


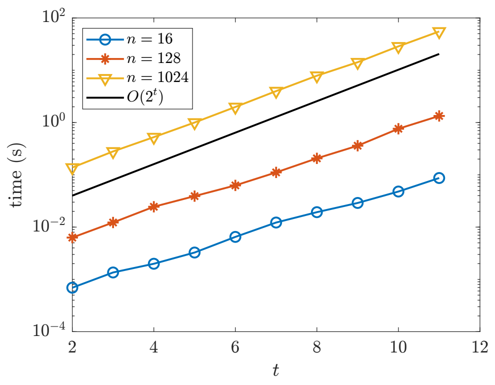
Fig. 6.3: Boxplots of the accuracy gain (6.1) achieved by the optimal algorithm with respect to the RTN baseline for 100 randomly chosen couples $(x, y) \in \mathbb{R}^{n \times n}$, for different values of t and n: $n = 16, 128, 1024$ for fixed $t = 8$ (left), and $t = 4, 8, 11$ for fixed $n = 128$ (right).

half the cases.

6.4. Time cost. Finally we conclude these numerical experiments on the optimal algorithm by measuring its time cost. Figure 6.4 reports the execution time for different values of n and t. The figure confirms that the time cost of the algorithm is in $O(n^2 2^t)$, as predicted by Theorem 5.3. Thus, the cost of the optimal algorithm remains tractable for a wide range of t and n values of interest.

7. Application to butterfly factorization. We now consider the application of the optimal rank-one quantization developed in the previous section to the quantization of butterfly factors. The code to reproduce the experiments in this section is available online[3].

[3]https://perso.ens-lyon.fr/elisa.riccietti/code.php, https://inria.hal.science/hal-04124171



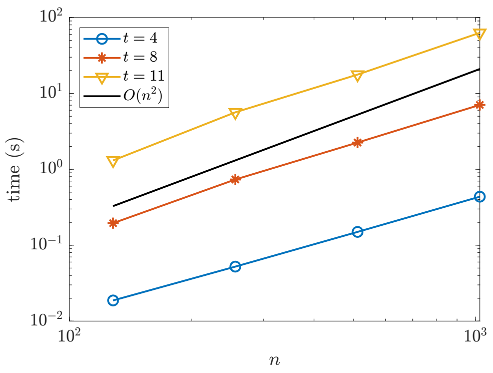


Fig. 6.4: Time cost of the optimal algorithm as a function of t and n.

The problem of butterfly factorization amounts to writing an $n \times n$ matrix Z, with $n = 2^L$, as a product $B_1 \dots B_L$, where each *butterfly factor* B_ℓ is an $n \times n$ sparse matrix with a fixed sparsity pattern [12]. Such a factorization is desired to reduce time and memory complexity for numerical methods involving the linear operator associated to Z, for example in large-scale linear inverse problems [5]. The butterfly factors appear in particular in fast transforms such as the discrete Fourier transform or the Hadamard transform [1], and, thanks to their strong expressivity, find a wide range of applications in various domains at the interface of signal processing and machine learning [4, 3], for instance in the approximation of weight matrices in deep neural networks.

7.1. Preliminaries on butterfly factors. Each butterfly factor B_ℓ satisfies the fixed support constraint $\mathrm{supp}(B_\ell) \subseteq \mathrm{supp}(S_\ell)$ for $1 \leq \ell \leq L := \log_2(n)$, where $\mathrm{supp}(B_\ell)$ denotes the support (that is, the set of indices associated with the nonzero elements, which can be interpreted both as a binary matrix and a set of indices) of B_ℓ and

$$S_\ell := I_{2^{\ell-1}} \otimes \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix} \otimes I_{n/2^\ell},$$

with I_k the identity matrix of order k and $\otimes$ the Kronecker product. This leads to highly sparse factors: each B_ℓ has exactly two nonzero elements on each row and on each column. The support of the butterfly factors for $n = 16$ are depicted in Figure 7.1.

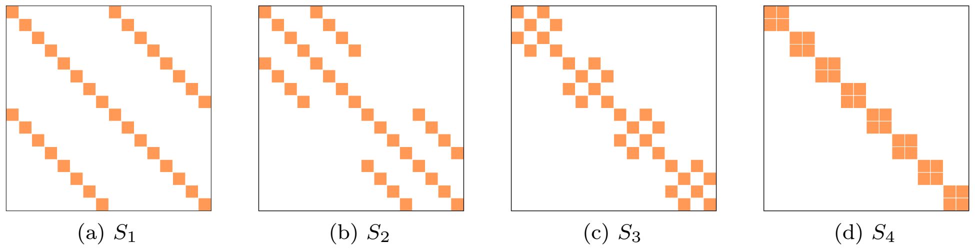


Fig. 7.1: Support of the butterfly factors, $n = 16$.

The butterfly supports have the interesting property that for any subset of consecutive factors, the product between $X := B_{\ell_0} \dots B_{\ell_1} \in \mathbb{R}^{n\times n}$ and $Y^\top := B_{\ell_1+1} \dots B_{\ell_2} \in \mathbb{R}^{n\times n}$, with $1 \le \ell_0 \le \ell_1 < \ell_2 \le L$, can be written as

$$XY^\top = \sum_{i=1}^{n} x_i y_i^\top =: \sum_{i=1}^{n} C_i, \tag{7.1}$$

where the n rank-one matrices $C_i := x_i y_i^\top \in \mathbb{R}^{n\times n}$ (the so-called rank-one components) associated with the columns of X and Y have disjoint supports [11, Lemma 2], that is,

$$\operatorname{supp}(C_i) \cap \operatorname{supp}(C_j) = \emptyset, \quad i \neq j. \tag{7.2}$$

Moreover the matrices C_i partially retain the sparsity of the butterfly factors: a product of p consecutive support factors has exactly 2^p nonzero elements on each row and on each column, so that in (7.1) x_i and y_i satisfy $\|x_i\|_0 \le 2^p$ and $\|y_i\|_0 \le 2^q$ with $p := \ell_1 - \ell_0 + 1$ and $q := \ell_2 - \ell_1$.

These are crucial properties that can be exploited to design of an efficient factorization algorithm, relying on a non-trivial application of singular value decomposition (SVD) to compute best rank-one approximations of specific submatrices, which has bounded complexity and is endowed with exact recovery guarantees [11, 10, 15]. We will see next how to exploit it also for the quantization of butterfly factors.

7.2. Quantization of butterfly factors. Given $L := \log_2(n)$ butterfly factors $B_1, \dots, B_L \in \mathbb{R}^{n\times n}$ (that may have been obtained using an existing algorithm to approximate some target dense matrix Z) we now seek to develop a procedure to quantize the factors $B_1, \dots, B_L$ while minimizing the error

$$\|B_1 \dots B_L - \widehat{B}_1 \dots \widehat{B}_L\|.$$

The quantized factors $\widehat{B}_1, \dots, \widehat{B}_L$ must have coefficients in $\mathbb{F}_t$ and retain the same support as the unquantized factors: $\operatorname{supp}(\widehat{B}_\ell) \subseteq \operatorname{supp}(B_\ell)$, $\ell = 1\colon L$.

We first discuss the case of a two-factor decomposition $XY^\top$, in which case an optimal quantization $\widehat{X}\widehat{Y}^\top$ can be found by solving a series of rank-one problems through our optimal Algorithm 5.1. Indeed, exploiting the decomposition (7.1) into rank-one components *with disjoint supports*, it is not difficult to check that the following optimal quantization problem

$$\widehat{X}, \widehat{Y} \in \arg \min_{\widehat{X}, \widehat{Y} \in \mathcal{F}_t} \|XY^\top - \widehat{X}\widehat{Y}^\top\|^2, \tag{7.3}$$

where $\mathcal{F}_t$ is the set of matrices with coefficient in $\mathbb{F}_t$ and the same supports as X, Y, decouples into n independent optimal rank-one quantization problems. Its solution can thus be computed by n independent applications of Algorithm 5.1 and yields diagonal matrices $\Lambda, M \in \mathbb{R}^{n\times n}$ such that $\widehat{X} = \operatorname{round}(X\Lambda)$ and $\widehat{Y} = \operatorname{round}(YM)$. We will also need a version with asymmetric quantization constraints, where $t_x, t_y \in \mathbb{N} \cup \{\infty\}$ and

$$\widehat{X}, \widehat{Y} \in \arg \min_{\widehat{X} \in \mathcal{F}_{t_x}, \widehat{Y} \in \mathcal{F}_{t_y}} \|XY^\top - \widehat{X}\widehat{Y}^\top\|^2, \tag{7.4}$$

with the convention $\mathbb{F}_\infty = \mathbb{R}$. The resulting process is outlined in Algorithm 7.1.

We can then use this optimal two-factor quantization algorithm as a building block in a heuristic procedure to quantize a decomposition $B_1 \dots B_L$ with more than

Algorithm 7.1 Optimal two-factor structured quantization.

```
Input: $t_x, t_y \in \mathbb{N} \cup \{\infty\}$, $X, Y \in \mathbb{R}^{n\times n}$ satisfying (7.1),(7.2).
Output: $\widehat{X} \in \mathbb{F}_{t_x}^{n\times n}, \widehat{Y} \in \mathbb{F}_{t_y}^{n\times n}$, solution to (7.4).
1: Initialize $\Lambda = M = 0$.
2: for $i = 1$ to $n$ do
3:     Set $x_i$ and $y_i$ to the $i$th columns of $X$ and $Y$, respectively.
4:     Quantize the rank-one matrix $x_i y_i^\top$ via Algorithm 5.1 (see Remark 5.1 when
       $t_x \neq t_y$), yielding $\widehat{x}_i = \mathrm{round}_{t_x}(\lambda_i x_i)$ and $\widehat{y}_i = \mathrm{round}_{t_y}(\mu_i y_i)$.
5:     Set $\Lambda_{ii} = \lambda_i$ and $M_{ii} = \mu_i$.
6: end for
7: $\widehat{X} = \mathrm{round}_{t_x}(X\Lambda)$, $\widehat{Y} = \mathrm{round}_{t_y}(YM)$.
```

two factors ($L > 2$). We can use different parenthesizations corresponding to so-called factor-bracketing trees [15]) of the product to sequentially divide it in a series of two-factor products with disjoint rank-one components, to which we can apply Algorithm 7.1. Note that the quantizations of each individual product will be optimal, but there is no guarantee that the entire process is globally optimal.

We consider two heuristics based on different parenthesizations. The first heuristic uses the parenthesization $(B_1B_2)(B_3B_4)\dots(B_{L-1}B_L)$ (assuming that L is even), and quantizations are performed on each pair of consecutive factors $(B_\ell B_{\ell+1})$ separately. If the number of factors L is odd, the last factor B_L is simply quantized by RTN, $\widehat{B}_L = \mathrm{round}(B_L)$. This "pairwise" heuristic is outlined in Algorithm 7.2.

Algorithm 7.2 L-factor butterfly quantization: pairwise heuristic.

```
Input: $t \geq 1$ an integer, $B_1, \dots, B_L \in \mathbb{R}^{n\times n}$  ($L = \log_2(n)$, $B_\ell$ are butterfly factors).
Output: $\widehat{B}_1, \dots, \widehat{B}_L \in \mathbb{F}_t^{n\times n}$ (with unchanged support).
1: for $\ell = 1$ to $\lfloor L/2 \rfloor$ do
2:     Quantize $B_{2\ell-1}B_{2\ell}$ via Algorithm 7.1 with $t_x = t_y = t$.
3: end for
4: if $L$ is odd then
5:     $\widehat{B}_L = \mathrm{round}(B_L)$
6: end if
```

As the pairwise heuristic acts very locally, we also propose a second one intuitively expected to provide more accuracy (this will indeed be confirmed experimentally). This second heuristic proceeds to a parenthesization from left to right (a variant from right to left is immediate) and performs a quantization at each level of the tree by quantizing the factors one by one in L steps. At step ℓ, the first $\ell - 1$ factors have already been quantized, and the remaining factors are parenthesized as $(MB_\ell)(B_{\ell+1}\dots B_L)$, where the current diagonal scaling M comes from the quantization at step $\ell - 1$. The quantized factor $\widehat{B}_\ell$ is obtained using the two-factor quantization in Algorithm 7.1 with $X = MB_\ell$ and $Y^\top = B_{\ell+1}\dots B_L$. Notice that Algorithm 7.1 does not alter the structure of the factors X, Y: the outputs $\widehat{X}, \widehat{Y}$ still satisfy (7.1),(7.2). This is a crucial observation to be able to employ this algorithm in a sequential manner.

Importantly, at every step except the last, we only need to quantize the left factor $X = MB_\ell$, since the factors $B_{\ell+1}, \dots B_L$ will be quantized at later steps. This observation can be taken into account by *not imposing* any quantization constraint

Algorithm 7.3 *L-factor butterfly quantization: left-to-right heuristic.*

```
Input: $t \geq 1$ an integer, $B_1, \dots, B_L \in \mathbb{R}^{n \times n}$   ($L = \log_2(n)$, $B_\ell$ are butterfly factors)
Output: $\widehat{B}_1, \dots, \widehat{B}_L \in \mathbb{F}_t^{n \times n}$ (with unchanged support).
1: $X \leftarrow B_1$
2: $Y^\top \leftarrow B_2 \dots B_L$
3: for $\ell = 1$ to $L - 2$ do
4:     Quantize $XY^\top$ via Algorithm 7.1 with $t_x = t$, $t_y = \infty$ yielding $\widehat{B}_\ell = \widehat{X} = \mathrm{round}(X\Lambda)$ and $\widehat{Y} = YM$ for suitable diagonal scalings $\Lambda$ and $M$.
5:     $X \leftarrow MB_{\ell+1}$
6:     $Y^\top \leftarrow B_{\ell+2} \dots B_L$
7: end for
8: Quantize $XY^\top$ via Algorithm 7.1 with $t_x = t_y = t$, yielding $\widehat{B}_{L-1} = \widehat{X} = \mathrm{round}(X\Lambda)$ and $\widehat{B}_L = \widehat{Y}^\top = \mathrm{round}(YM)^\top$ for diagonal scalings $\Lambda$ and $M$.
```

on the right factor Y to obtain $\widehat{X} = \mathrm{round}(X\Lambda)$ and $\widehat{Y} = YM$ with M the new scaling factor that will serve at the next step. This left-to-right heuristic is outlined in Algorithm 7.3. The choice made at line 4 of Algorithm 7.3 to not quantize the right factor ($t_y = \infty$), compared with a variant where both factors would be constrained to be quantized ($t_x = t_y = t$), was observed to significantly improve the accuracy of the global quantization. We note that we have also tested a similar heuristic with a right-to-left parenthesization and obtained equivalent results.

Complexity. The left-to-right heuristic is more expensive than the pairwise one. Indeed, the pairwise heuristic preserves the extreme sparsity of the butterfly matrices; in fact, it does not require forming any explicit intermediate matrix product and can directly work in-place by replacing the butterfly factors by their quantized version. In contrast, the left-to-right heuristic requires forming the explicit product of up to $L - 1$ consecutive factors into $Y^\top$, which is an almost dense matrix. For large n, forming this entire matrix would be untractable but, fortunately, we only need to access one row of $Y^\top$ at a time. Therefore, we can form y_i (which is almost dense), use it to quantize x_i (which is sparse), and discard y_i before forming y_{i+1}. In practice, to attain high performance, we form a block of multiple consecutive rows at a time. Note that for both heuristics, the left factor X remains completely sparse, with only two nonzero coefficients for each column x_i; therefore, in Theorem 5.3 we have $\min(m, n) = m = 2$. As a result of these observations, both heuristics only require $O(nL + 2^t) = O(n \log n + 2^t)$ space. In terms of time, the pairwise heuristic requires $O(nL)$ calls to Algorithm 5.1 with vectors x_i, y_i with at most 2 nonzero elements, which by Theorem 5.3 yields a $O(2^t nL)$ complexity. As for the left-to-right heuristic, we need to take into account both the cost of calling Algorithm 5.1 and that of forming the matrices X and Y on which it is called. At step ℓ of Algorithm 7.3, each row $y_i^\top$ of $Y^\top$, that is, each column y_i of Y, is formed by the products

$$\begin{aligned} y_i &= B_L^\top \dots B_{\ell+2}^\top v_i^{(\ell+1)} \\ &= B_L^\top \dots B_{\ell+3}^\top v_i^{(\ell+2)} \\ &\dots \\ &= B_L^\top v_i^{(L-1)}, \end{aligned}$$

where $v_i^{(\ell+1)}$ is initialized to the ith column of $B_{\ell+1}^\top$ and $v_i^{(\ell+2)}, \dots, v_i^{(L-1)}$ are the vectors formed during the intermediate computations, which satisfy $\|v_i^{(k)}\|_0 \le 2^{k-\ell}$ for $\ell+1 \le k \le L-1$. Since B_{k+1} has at most two nonzero elements per row, the cost of computing $v_i^{(k+1)}$ from $v_i^{(k)}$ is in $O(\|v_i^{(k)}\|_0)$. Therefore, the cost of forming y_i at step ℓ is in $O(\sum_{k=\ell+1}^{L-1} 2^{k-\ell}) = O(n2^{-\ell})$, and by Theorem 5.3 the cost of applying Algorithm 5.1 on x_i, y_i is in $O(2^t n 2^{-\ell})$. Summing over all rows $i = 1\colon n$ and all steps $\ell = 1\colon L-1$ thus yields a total time complexity in $O(2^t n^2)$.

We summarize this complexity analysis in the following theorem.

THEOREM 7.1. *Algorithm 7.2 and Algorithm 7.3 have time complexities in $O(2^t n \log n)$ and in $O(2^t n^2)$, respectively. Both algorithms have a space complexity in $O(2^t + n \log n)$.*

7.3. Experiments. We conclude with some numerical experiments. We consider a range of matrices $Z \in \mathbb{R}^{n\times n}$ for several values of $n = 2^L$, with $L := \log_2(n)$ up to 16, where each of the L butterfly factors is generated with random uniform values in $[-1, 1]$.

Figure 7.2 plots the quantization error obtained for varying t (left) or n (right). We compare three approaches: the naive RTN baseline where each factor is separately quantized with RTN, and our two heuristic approaches (Algorithm 7.2 and Algorithm 7.3) that use as a building block the optimal two-factor quantization. We find that both heuristics significantly reduce the quantization error with respect to the RTN baseline and that, as expected, the more expensive left-to-right heuristic is more accurate than the pairwise one. Importantly, both heuristics achieve an accuracy that behaves significantly better than the $O(2^{-t})$ RTN accuracy: the pairwise and left-to-right heuristics approximately behave as $O(2^{-1.3t})$ and $O(2^{-1.4t})$, respectively. Thanks to this improved accuracy, our algorithms can achieve an accuracy equivalent to the RTN baseline with a lower precision of $t' = t/1.4$ for the left-to-right heuristic, which represents a $1 - 1/1.4 \approx 30\%$ reduction of storage (and similarly, the pairwise heuristic achieves a $1 - 1/1.3 \approx 23\%$ reduction) for the same accuracy.

Figure 7.3 compares the time cost of both heuristics. As expected, the pairwise one is (slightly) faster than the left-to-right one. However, both heuristics have much closer and much better time complexities than predicted by the theoretical bounds discussed in the previous section, and both are tractable for values of t of interest (here up to $t = 11$) and for reasonably large values of n (here up to $n = 2^{18} = 262144$). It remains open whether better bounds and/or more efficient implementations can be achieved.

8. Conclusions.

We tackled the problem of optimally quantizing a rank-one matrix $xy^\top$ in a floating-point arithmetic with t bits of precision. We showed that the apparent combinatorial nature of the problem can be overcome by characterizing the optimal solution $\widehat{x}\widehat{y}^\top$ as the quantization of suitably scaled factors, $\widehat{x} = \mathrm{round}(\lambda x)$ and $\widehat{y} = \mathrm{round}(\mu y)$, see Theorem 4.5. We then devised an algorithm, see Algorithm 5.1, that can find these optimal scaling parameters with a time complexity in $O(mn2^t)$, which is tractable for the values of t of interest corresponding to low precisions. We showed both theoretically and experimentally that this optimal algorithm can achieve much more accurate quantizations than by simply quantizing x and y separately. Finally, we explained how this optimal rank-one matrix quantization can be applied to the quantization of butterfly factors—an important tool appearing in many areas of scientific computing. We proposed two heuristics that employ the proposed rank-one matrix quantization algorithm as building block, and we demonstrated their effectiveness compared with quantizing each factor separately: we can improve the

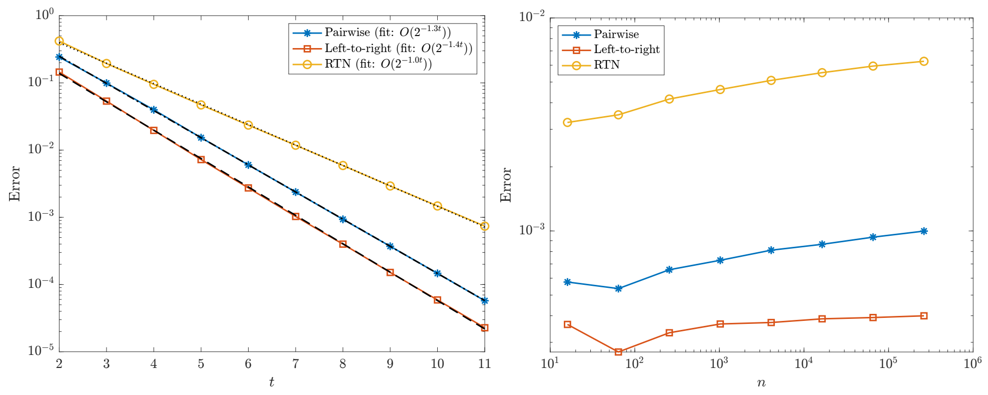


Fig. 7.2: Quantization error for butterfly factors with either the RTN baseline or pairwise or left-to-right algorithms, as a function of t and n. Left: $n = 2^{16}$, t varies; right: $t = 8$, n varies.

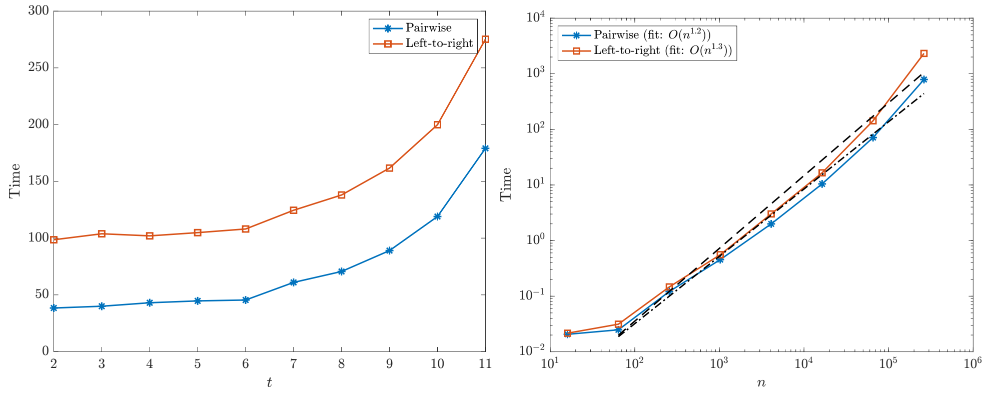


Fig. 7.3: Time cost (in seconds) for quantizing butterfly factors with either the pairwise or left-to-right algorithms, as a function of t and n. Left: $n = 2^{16}$, t varies; right: $t = 8$, n varies.

accuracy by several orders of magnitude for the same storage budget or, equivalently, we can reduce the storage by up to 30% with no accuracy loss.

This work opens several research questions regarding its extension to various higher dimensional settings: the complex-valued case (where both the real and imaginary parts are quantized in $\mathbb{F}_t$); the case of rank-r matrices (with $r > 1$); and the case of rank-one tensors. These problems can be decoupled into individual rank-one matrix quantization problems that could be tackled by using the approach proposed here as a building block; however better results are likely to be obtained by studying these higher dimensional problems globally.

Finally, since scaling invariance—which is at the heart of our optimal rank-one quantization algorithm—has also led to heuristic schemes to quantize deep ReLU networks [13], an exciting challenge is to extend our principled approach in such a setting, possibly up to extreme one-bit quantization [2].

Acknowledgments. This project was supported by the AllegroAssai ANR project ANR-19-CHIA-0009 and the GdR ISIS project MOMIGS.

REFERENCES

[1] J. W. Cooley and J. W. Tukey, *An algorithm for the machine computation of the complex Fourier series*, Mathematics of Computation, 19 (1965), pp. 297–301, https://doi.org/10.1090/S0025-5718-1965-0178586-1.

[2] M. Courbariaux, Y. Bengio, and J.-P. David, *BinaryConnect: Training Deep Neural Networks with binary weights during propagations*, in NeurIPS, 2015, http://papers.nips.cc/paper/5647-binaryconnect-training-deep-neural-networks-with-b.

[3] T. Dao, B. Chen, N. S. Sohoni, A. Desai, M. Poli, J. Grogan, A. Liu, A. Rao, A. Rudra, and C. Ré, *Monarch: Expressive structured matrices for efficient and accurate training*, in International Conference on Machine Learning (ICML), PMLR, 2022, pp. 4690–4721, https://proceedings.mlr.press/v162/dao22a/dao22a.pdf.

[4] T. Dao, A. Gu, M. Eichhorn, A. Rudra, and C. Ré, *Learning fast algorithms for linear transforms using butterfly factorizations*, in International conference on machine learning (ICML), PMLR, 2019, pp. 1517–1527, http://proceedings.mlr.press/v97/dao19a/dao19a.pdf.

[5] M. Elad, *Sparse and redundant representations: from theory to applications in signal and image processing*, Springer Science & Business Media, (2010), https://doi.org/10.1007/978-1-4419-7011-4.

[6] A. Gholami, S. Kim, Z. Dong, Z. Yao, M. W. Mahoney, and K. Keutzer, *A survey of quantization methods for efficient neural network inference*, arXiv preprint 2103.13630, (2021), https://arxiv.org/abs/2103.13630.

[7] N. J. Higham, *Accuracy and Stability of Numerical Algorithms*, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, second ed., 2002, https://doi.org/10.1137/1.9780898718027.

[8] N. J. Higham and T. Mary, *Mixed precision algorithms in numerical linear algebra*, Acta Numerica, 31 (2022), pp. 347–414, https://doi.org/10.1017/s0962492922000022.

[9] C.-P. Jeannerod and S. M. Rump, *On relative errors of floating-point operations: optimal bounds and applications*, Math. Comp., 87 (2018), pp. 803–819, https://www.ams.org/journals/mcom/2018-87-310/S0025-5718-2017-03234-8/S0025-5718-2017-03234-8.pdf.

[10] Q.-T. Le, E. Riccietti, and R. Gribonval, *Spurious valleys, NP-hardness, and tractability of sparse matrix factorization with fixed support*, SIAM J. Matrix Anal. Appl., 44 (2023), pp. 503–529, https://doi.org/10.1137/22M1496657.

[11] Q.-T. Le, L. Zheng, E. Riccietti, and R. Gribonval, *Fast learning of fast transforms, with guarantees*, in ICASSP 2022, 2022, https://doi.org/10.1109/ICASSP43922.2022.9747791.

[12] L. Le Magoarou and R. Gribonval, *Flexible multilayer sparse approximations of matrices and applications*, IEEE JSTSP, 10 (2016), pp. 688–700, https://doi.org/10.1109/JSTSP.2016.2543461.

[13] M. Nagel, M. van Baalen, T. Blankevoort, and M. Welling, *Data-Free Quantization Through Weight Equalization and Bias Correction*, in ICCV, 2019, http://arxiv.org/abs/1906.04721 (accessed 2021-03-27).

[14] Tchebyshev, *Mémoire sur les nombres premiers.*, Journal de mathématiques pures et appliquées 1re série, 17 (1852), pp. 366–390, http://www.numdam.org/item/JMPA_1852_1_17__366_0.pdf.

[15] L. Zheng, E. Riccietti, and R. Gribonval, *Efficient identification of butterfly sparse matrix factorizations*, SIAM Journal on Mathematics of Data Science, 5 (2023), pp. 22–49, https://doi.org/10.1137/22M1488727.