Scaling is all you need: quantization of butterfly matrix products via optimal rank-one quantization

Rémi GRIBONVAL[1] Theo MARY[2] Elisa RICCIETTI[1]

[1]Univ Lyon, ENS de Lyon, UCBL, CNRS, Inria, LIP, F-69342, LYON Cedex 07, France

[2]Sorbonne Université, CNRS, LIP6, F-75005 Paris, France

Résumé – On s'intéresse à la quantification des facteurs dans un produit de matrices creuses à structure papillon. On montre que les propriétés particulières es supports des facteurs papillon permettent d'aborder le problème comme une séquence de problème de quantification de rang un. On propose un algorithme optimal de quantification de rang un utilisant les invariances du problème par remise à l'échelle. Cet algorithme sert de brique de base dans une approche heuristique pour la quantification de produits papillon, qui s'avère bien plus précis qu'une approche naive basée sur la quantification de chaque facteur au plus proche voisin indépendamment.

Abstract – We consider the problem of quantizing the factors of butterfly factorizations. We show how the properties of butterfly supports can be exploited to reduce the problem to the solution of a series of rank-one quantization problems. We thus propose an algorithm that, exploiting the intrinsic scaling invariance of the problem, gives the optimal solution of the rank-one quantization problem. We employ this algorithm as a building block in a heuristic procedure to solve the quantization problem of butterfly factors, which we show can be much more accurate that the naive round-to-nearest startegy of quantizing each factor independently.

1 Introduction

Approximating a large dense matrix as a product of two or more sparse factors is a fundamental problem in many tasks, such as in signal processing and machine learning, see e.g., [6, 8, 2]. This amounts to finding, given a matrix $\mathbf{Z}$, J sparse factors $\mathbf{X}_1, \dots, \mathbf{X}_J$ such that $\mathbf{Z} \approx \mathbf{X}_1 \dots \mathbf{X}_J$. This is often referred to as sparse matrix factorization (SMF). The sparse factors usually belong to structured family of matrices, so that the sparsity pattern can be easily exploited to reduce the computational cost of linear operations involving the matrix $\mathbf{Z}$. 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.

Due to the growing size of matrices in applications, optimizing the memory usage of SMF algorithms is important. Quantization methods have been applied in many fields to deal with the always growing scale of models and datasets, for instance in the training and inference of large deep neural networks (DNN), see e.g., [7] but, to our knowledge, quantization in SMF has never been addressed before. The interest for this problem, especially when butterfly factorizations are considered, is further motivated by the recent works on the approximation of weight matrices in DNN with sparse structured matrices [2, 3].

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 [5] to design algorithms able to approximate a given matrix $\mathbf{Z}$ with a butterfly product. In this paper we also exploit this property to quantize butterfly factors $\mathbf{X}_1 \dots \mathbf{X}_J$, in order to approximate their product with low error $\|\mathbf{X}_1 \dots \mathbf{X}_J - \hat{\mathbf{X}}_1 \dots \hat{\mathbf{X}}_J\|$ – here, $\|\cdot\|$ is the Frobenius norm. The key ingredient that our approach relies on is optimal rank-one quantization: given $\mathbb{F}_t$ a finite set of t-bit numbers and unquantized (or high-precision) vectors $x \in \mathbb{R}^m$, $y \in \mathbb{R}^n$, we wish to solve

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

We thus first propose in Section 2 an optimal algorithm with bounded complexity for the solution of (1.1), which differs from the naive round-to-nearest strategy by exploiting the intrinsic scaling invariances of the problem. We demonstrate (empirically and with theoretical upper/lower bounds on the worst case error) in Section 3 that using a reduced number of bits to quantize x and y can preserve the precision of their product. This is used in Section 4 to propose an approach to quantize butterfly factors, and we demonstrate how much savings on precision can be achieved on each factor when the number of factors increases. Due to lack of space all formal proofs are omitted.

2 Optimal rank-one quantization

We consider rank-one quantization: given $x \in \mathbb{R}^m$, $y \in \mathbb{R}^n$ where $m, n \geq 1$, we seek quantized $\widehat{x} \in \mathbb{F}_t^m$, $\widehat{y} \in \mathbb{F}_t^n$ that solve

$$\min_{\widehat{x}\in\mathbb{F}_t^m, \widehat{y}\in\mathbb{F}_t^n} C_{x,y}(\widehat{x}, \widehat{y}), \text{ with } C_{x,y}(\widehat{x}, \widehat{y}) \coloneqq \|xy^\top - \widehat{x}\widehat{y}^\top\|^2. \tag{2.1}$$

A naive approach is to map each element of x and y to their nearest neighbor in $\mathbb{F}_t$. This yields quantized $\widehat{x}$ and $\widehat{y}$ satisfying

$$\widehat{x} = \mathrm{round}(x) = x + \Delta x, \quad \|\Delta x\| \leq v_t \|x\|, \tag{2.2a}$$

$$\widehat{y} = \mathrm{round}(y) = y + \Delta y, \quad \|\Delta y\| \leq v_t \|y\|, \tag{2.2b}$$

where $\mathrm{round}(\cdot)$ is the function that maps any $a \in \mathbb{R}$ to its nearest neighbor in $\mathbb{F}_t$ (we ignore ties for the sake of simplicity), and is applied elementwise. The quantity $v_t \coloneqq \frac{2^{-t}}{1+2^{-t}}$ bounds the maximum relative distance between any element in the representable range and its nearest neighbor in $\mathbb{F}_t$ [4]: for every $a \in \mathbb{R}$, we have $|\mathrm{round}(a) - a| \leq v_t|a|$. We will refer to this "naive" approach as the round-to-nearest (RTN) approach. It is worth noting that, by definition, $\widehat{x}$ and $\widehat{y}$ are optimal quantizations of x and y, in the sense that they minimize the errors $\|x - \widehat{x}\|$ and $\|y - \widehat{y}\|$. However, our goal is to minimize instead the overall quantization error $\|xy^\top - xy^\top\|$, and the RTN approach is far from optimal, as it does not take into account the intrinsic scaling invariances of the problem. The following result bounds the worst case quantization error obtained with RTN.

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

$$\|xy^\top - \mathrm{round}(x)\mathrm{round}(y)^\top\| \leq (2v_t + v_t^2)\|x\|\|y\|. \quad (2.3)$$

As we will see, optimal quantization can achieve much lower errors. However, finding an optimal quantization is a harder problem. For example, for low precision quantization (small values of t), and by restricting the set $\mathbb{F}_t$ to a finite interval such as $[1, 2]$, $\mathbb{F}_t \cap [1, 2]$ only has a small number of elements and so one may think of using a brute-force algorithm that enumerates all possible solutions. However, the challenge is that this brute force approach requires to test each of these values for each element of $\widehat{x}$, $\widehat{y}$, yielding a $O(2^{t(m+n)})$ complexity which is exponential in $m + n$ and thus clearly intractable except for very small problems. One of the main contributions of this work is to characterize the optimal solution of this problem and to use this characterization to devise a tractable solution method with complexity $O(mn2^t)$.

2.1 Existence and characterization of an optimal quantization

Our first result is the key technical ingredient to prove that an optimal quantization exists and to characterize it. It allows us to reduce the problem to a scalar one: it suffices to find an optimum λ^* of a scalar function $f(\lambda)$, assuming it exists.

Lemma 2.2. *Let $x \in \mathbb{R}^m$, $y \in \mathbb{R}^n$. We have*

$$\inf_{\widehat{x}\in\mathbb{F}_t^m, \widehat{y}\in\mathbb{F}_t^n} C_{x,y}(\widehat{x}, \widehat{y}) = \inf_{\lambda\in\mathbb{R}} f(\lambda), \quad (2.4)$$

$$\textit{with } f(\lambda) \coloneqq \max_{\widehat{x}\in\mathrm{round}(\lambda x)} C_{x,y}\big(\widehat{x}, \mathrm{round}(\mu(\widehat{x})y)\big) \quad (2.5)$$

where $\mu(\widehat{x}) \coloneqq \frac{x^\top\widehat{x}}{\|\widehat{x}\|^2}$ if $\widehat{x} \neq 0$ and 0 otherwise.

We also shw that the function $f(\lambda)$ takes a finite number of values in $[1, 2)$, therefore proving the existence of an optimum. Formally, we need to define the *breakpoints* of the function $\lambda \mapsto \mathrm{round}(\lambda x)$ for a given x. The breakpoints correspond to the values of λ for which there exists at least one coordinate i such that $\mathrm{round}(\lambda x_i)$ corresponds to a tie. For any $x \in \mathbb{R}^m$, the function $\lambda \mapsto \mathrm{round}(\lambda x)$ is piecewise constant with finitely many breakpoints in the open interval $(1, 2)$. We denote them λ_j, $1 \leq j \leq B$ in increasing order and establish the following result.

Theorem 2.3. *Consider nonzero $x \in \mathbb{R}^m$, $y \in \mathbb{R}^n$, with $m, n \geq 1$, and $t \geq 1$. Denote $\lambda_0 \coloneqq 1 < \lambda_1 < \ldots < \lambda_B < 2 =: \lambda_{B+1}$ with λ_j, $1 \leq j \leq B$ the breakpoints of $\lambda \in (1, 2) \mapsto \mathrm{round}(\lambda x)$, and $\lambda_{j+1/2} \coloneqq (\lambda_j + \lambda_{j+1})/2$, $0 \leq j \leq B$. Problem (2.1) admits an optimum $\widehat{x}, \widehat{y}$ such that*

$$\widehat{x} = \mathrm{round}(\lambda^* x) \quad \textit{with } \lambda^* = \lambda_{j^*+1/2} \textit{ for some } 0 \leq j^* \leq B,$$

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

This theorem tells us that to find an optimum of problem (2.1) we just need to find the optimal scaling parameter λ^*. Once λ^* has been determined, both $\widehat{x}$ and $\widehat{y}$ can easily be found. We thus have reduced a problem with mn variables to a problem with a single variable λ, whose feasible values belong to a finite set.

2.2 Optimal quantization algorithm

There remains to develop a tractable algorithm to actually compute the optimal scalar λ^* for each instance (x, y). To do so, we can leverage the characterization above by explicitly enumerating the breakpoints λ_j, $j = 1\colon B$. We outline in Algorithm 1 a method to solve (2.1). The algorithm builds the set of breakpoints $\mathbb{B}(x)$, sorts it in increasing order, and finally enumerates it to test each midpoint and find the optimal one.

Algorithm 1: An algorithm to solve (2.1).

Input: $x \in \mathbb{R}^m, y \in \mathbb{R}^n, t \in \mathbb{N}^*$;
Output: $\widehat{x}^* \in \mathbb{F}_t^m, \widehat{y}^* \in \mathbb{F}_t^n$ solutions to (2.1);
Initialize $\widehat{x}^* \leftarrow 0$, $\widehat{y}^* \leftarrow 0$;
if $x = 0$ *or* $y = 0$ **then**
  exit;
end
Sort breakpoints in increasing order to obtain λ_j, $1 \leq j \leq B$, $\lambda_0 \leftarrow 1$, $\lambda_{B+1} \leftarrow 2$;
for $j = 1$ ***to*** $B + 1$ **do**
  $\lambda \leftarrow (\lambda_{j-1} + \lambda_j)/2$;
  $\widehat{x} \leftarrow \mathrm{round}(\lambda x)$;
  $\mu \leftarrow x^\top\widehat{x}/\|\widehat{x}\|^2$;
  $\widehat{y} \leftarrow \mathrm{round}(\mu y)$;
  if $C_{x,y}(\widehat{x}, \widehat{y}) < C_{x,y}(\widehat{x}^*, \widehat{y}^*)$ **then**
    $\widehat{x}^* \leftarrow \widehat{x}$, $\widehat{y}^* \leftarrow \widehat{y}$;
  end
end

By Theorem 2.3, this algorithm computes the optimal quantization. Moreover, we can bound its complexity by showing that the number of breakpoints $\#\mathbb{B}(x)$ is bounded by $m2^t$. Building and sorting $\mathbb{B}(x)$ has a space cost in $O(m2^t)$ and a time cost in $O(m2^t \log m)$. The operations performed at each iteration of Algorithm 1 have a cost in $O(m + n)$, including the evaluation of $C_{x,y}(\widehat{x}, \widehat{y})$, which is efficiently performed by exploiting the fact that $xy^\top - \widehat{x}\widehat{y}^\top$ is a rank-two matrix. Taking into account the loop over $\#\mathbb{B}(x)$ breakpoints, Algorithm 1 has a total time cost in $O((m + n)\#\mathbb{B}(x)) = O((m + n)m2^t)$. Since we can also reverse the respective roles of x and y by looping on

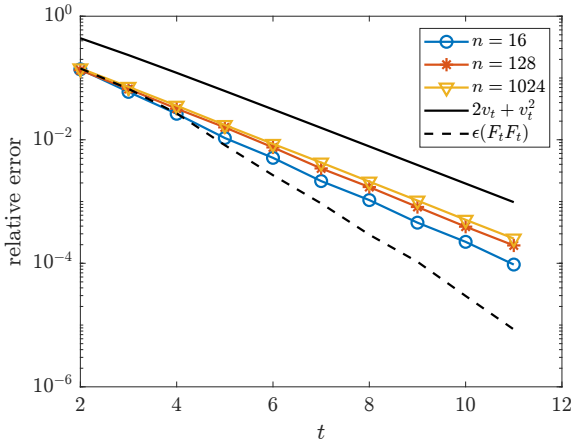

Figure 3.1 – Worst relative error $\sqrt{C_{x,y}(\widehat{x},\widehat{y})}/\|xy^T\|$ over 100 random pairs $(x, y) \in \mathbb{R}^{n\times n}$ and its lower/upper bounds of (3.1).

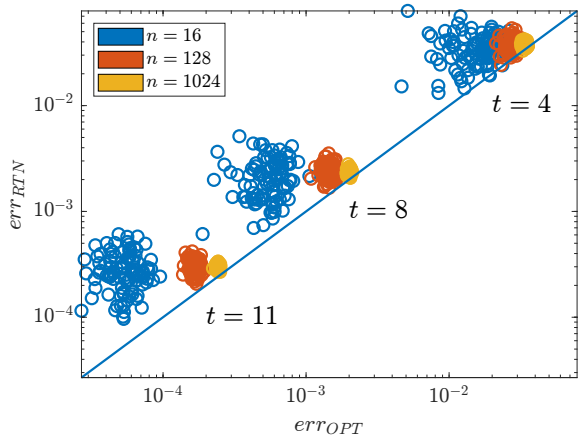

Figure 3.2 – Scatter plot of err_{OPT}, err_{RTN} for 100 randomly chosen couples $(x, y) \in \mathbb{R}^{n\times n}$ for different values of n and t.

$\mathbb{B}(y)$, a time cost in $O((m + n)n2^t)$ can also be achieved, so choosing the best of both yields $O((m + n)\min(m, n)2^t) = O(\max(m, n)\min(m, n)2^t) = O(mn2^t)$. Overall we get:

Theorem 2.4. *Algorithm 1 solves problem* (2.1) *in* $O(mn2^t)$ *time and in* $O(\min(m, n)2^t)$ *space.*

3 Rank-one quantization error

We can prove the following bounds on the worst-case optimal rank-one quantization error, denoted $\epsilon(\Sigma_1^{m\times n}(\mathbb{F}_t))$:

$$\epsilon(\mathbb{F}_t\mathbb{F}_t) \le \epsilon(\Sigma_1^{m\times n}(\mathbb{F}_t)) \le 2v_t + v_t^2 \approx 2^{1-t}, \tag{3.1}$$

$$\text{where} \quad \epsilon(\mathbb{F}_t\mathbb{F}_t) := \sup_{z\in\mathbb{R}\setminus\{0\}} \frac{\inf_{\widehat{z}\in\mathbb{F}_t\mathbb{F}_t} |\widehat{z} - z|}{|z|} \tag{3.2}$$

is the *worst case relative error* of quantizing a scalar on the set $\mathbb{F}_t\mathbb{F}_t := \{\widehat{x}\widehat{y},\ \widehat{x} \in \mathbb{F}_t,\ \widehat{y} \in \mathbb{F}_t\}$. The upper bound in (3.1) is simply the worst-case bound of the naive RTN approach (cf. Lemma 2.1). While we do not have an explicit expression for the lower bound $\epsilon(\mathbb{F}_t\mathbb{F}_t)$, we can exhibit an algorithm to compute it given t. We observed that it approximately behaves as $2^{-1.6t}$.

We illustrate the theoretical bounds (3.1) in Figure 3.1 which compares the upper and lower bounds $\epsilon(\mathbb{F}_t\mathbb{F}_t)$ and $2v_t + v_t^2$ with the empirical worst-case quantization errors obtained over 100 randomly generated couples $x, y \in \mathbb{R}^n$ for different values of n and t. The empirical worst case error falls somewhere between its lower and upper bounds. Interestingly, while both bounds do not depend on n, the empirical worst-case error clearly increases with n, getting closer and closer to its upper bound.

We also compare the optimal quantization error $err_{OPT} := \|xy^\top - \widehat{x}\widehat{y}^\top\|/\|xy^\top\|$ with the RTN baseline error $err_{RTN} := \|xy^\top - \text{round}(x)\text{round}(y)^\top\|/\|xy^\top\|$. Figure 3.2 displays scatter plots of both errors for various values of n and t. The plot reveals two trends as n increases: the points become less dispersed, and closer to the diagonal (where $err_{OPT} = err_{RTN}$, 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.

4 Quantizing butterflies

We now consider the application of the optimal rank-one quantization developed in the previous section to the quantization of butterfly factorizations.

For this, observe first that Algorithm 1 can be adapted to also output optimal scalars $\lambda^*, \mu^* \in \mathbb{R}$ (cf. Theorem 2.3) such that $\widehat{x} = \text{round}(\lambda^* x)$ and $\widehat{y} = \text{round}(\mu^* y)$ are optimal solutions to (1.1). Minor adaptations of the algorithm also yield λ, μ such that $\widehat{x} := \text{round}(\lambda x)$ and $\widehat{y} := \mu y$ are optimal solutions to a variant of (1.1) where $\widehat{y} \in \mathbb{R}^n$ is not constrained to be quantized.

Now, given vectors $x_i, y_i \in \mathbb{R}^n, 1 \le i \le r$ such that the rank-one matrices $x_i y_i^\top$ have disjoint supports $T_i := \{(k, \ell) : (x_i)_k \neq 0 \text{ and } (y_i)_\ell \neq 0\}$, and considering $\mathbf{X}, \mathbf{Y} \in \mathbb{R}^{n\times r}$ the matrices with the corresponding columns, it is not difficult to check that the following optimal quantization problem

$$\widehat{\mathbf{X}}, \widehat{\mathbf{Y}} \in \arg\min_{\widehat{\mathbf{X}},\widehat{\mathbf{Y}}\in\mathcal{F}_t} \|\mathbf{X}\mathbf{Y}^\top - \widehat{\mathbf{X}}\widehat{\mathbf{Y}}^\top\|^2, \tag{4.1}$$

where $\mathcal{F}_t$ is the set of matrices with coefficient in $\mathbb{F}_t$ and the same supports as $\mathbf{X}, \mathbf{Y}$, decouples into r independent optimal rank-one quantization problems. Its solution can thus be computed by r independent applications of Algorithm 1 and yields diagonal matrices $\Lambda^*, M^* \in \mathbb{R}^{r\times r}$ such that $\widehat{\mathbf{X}} = \text{round}(\mathbf{X}\Lambda^*)$ and $\widehat{\mathbf{Y}} = \text{round}(\mathbf{Y}M^*)$ (resp. $\widehat{\mathbf{X}} = \text{round}(\mathbf{X}\Lambda)$, $\widehat{\mathbf{Y}} = \mathbf{Y}M$ when $\widehat{\mathbf{Y}}$ is not constrained to be quantized). According to Theorem 2.4 this is achievable with time complexity $O(2^t \sum_{i=1}^r \|x_i\|_0 \times \|y_i\|_0)$.

Finally, a property of butterfly products $\mathbf{Z} = \mathbf{X}_1 \dots \mathbf{X}_J \in \mathbb{R}^{n\times n}$, $J = \log_2(n)$, is that every pair of partial products $\mathbf{X} := \mathbf{X}_1 \dots \mathbf{X}_\ell$, $\mathbf{Y}^\top := \mathbf{X}_{\ell+1} \dots \mathbf{X}_J$ is such that the $r = n$ rank-one matrices $x_i y_i^\top$ associated to the columns of $\mathbf{X}, \mathbf{Y}$ have disjoint support [5, Lemma 2] and $\|x_i\|_0 \times \|y_i\|_0 = n$. This leads to the heuristic outlined in Algorithm 2, which quantizes the factors one by one in J steps. At step k, the factor $\mathbf{X}_k$ is quantized using optimal two-factor quantization on $\mathbf{X} = M\mathbf{X}_k$ and $\mathbf{Y}^\top = \mathbf{X}_{k+1} \dots \mathbf{X}_J$, where the diagonal scaling M comes from the quantization of the pair at step $k - 1$. Thus, the quantization of each individual factor is optimal but there is no guarantee that the entire process is globally optimal. Notice that in Algorithm 2,

each time the optimal algorithm is called, it is used to quantize only one of the factors, while the other is then used in full precision, except for the last iteration. A variant quantizing first the rightmost factors is straightforward, and extensions based on other factor-bracketing trees [9] are left to future work. Algorithm 2 has time complexity $O(J2^t rn) = O(2^t n^2 \log(n))$.

Algorithm 2: J-factor butterfly quantization.

$\mathbf{X} \leftarrow \mathbf{X}_1$;
$\mathbf{Y}^\top \leftarrow \mathbf{X}_2 \dots \mathbf{X}_J$;
for $k = 1$ ***to*** $J - 1$ **do**
  Quantize the two-factor decomposition $\mathbf{X}\mathbf{Y}^\top$ via the optimal algorithm to solve (4.1) with no quantization constraint on $\widehat{\mathbf{Y}}$, yielding $\widehat{\mathbf{X}}_k = \widehat{\mathbf{X}} = \text{round}(\mathbf{X}\Lambda)$ and $\widehat{\mathbf{Y}}^\top = M\mathbf{Y}^\top$ for suitable diagonal scalings Λ and M;
  if $k < J - 1$ **then**
    $\mathbf{X} \leftarrow M\mathbf{X}_{k+1}$;
    $\mathbf{Y}^\top \leftarrow \mathbf{X}_{k+2} \dots \mathbf{X}_J$;
  else
    $\widehat{\mathbf{X}}_J = \text{round}(M\mathbf{X})$;
  end
end

5 Experiments

To validate the approach, we consider a matrix $\mathbf{Z} \in \mathbb{R}^{n \times n}$ with $n = 8192$, written as the product of 13 orthonormal random butterfly factors. Figure 5.1 plots the quantization error obtained on average for 10 random instances, for t varying from 2 to 11 (we are mainly interested in low precision quantization). We compare the naive RTN baseline approach to our Algorithm 2. We find that our algorithm significantly reduces the quantization error: its accuracy approximately behaves as $O(2^{-1.4t})$, which is consistent with our previous analysis for rank-one matrices. Thanks to this $O(2^{-1.4t})$ accuracy, our algorithm can achieve an accuracy equivalent to the RTN baseline (which behaves as $O(2^{-t})$ as expected) with a lower precision of $t' = t/1.4$, which represents a $1 - 1/1.4 \approx 30\%$ reduction of storage.

6 Conclusion and perspectives

We proposed an optimal algorithm to quantize rank-one matrices and used it to quantize butterfly matrix products. The algorithms show promising gains with respect to the naive RTN strategy, and a natural perspective is their complex-valued extension, to deal with, e.g., quantized Fast Fourier Transforms. Another problem is to directly include such quantization within an algorithm to approximate a "dense" matrix $\mathbf{Z}$ as a butterfly product [5]. This is likely to require optimizing $\|\mathbf{B} - \hat{x}\hat{y}^\top\|_F^2$ when $\mathbf{B}$ is not necessarily of rank one, a problem that could be adressed via heuristics relying on our optimal quantization algorithm. Finally, since scaling invariance – which is at the heart of our optimal rank-one quantization algorithm – has also led to heuristic schemes ot quantize deep ReLU networks [7], an exciting challenge is to extend our principled approach in such a setting, possibly up to extreme one-bit quantization [1].

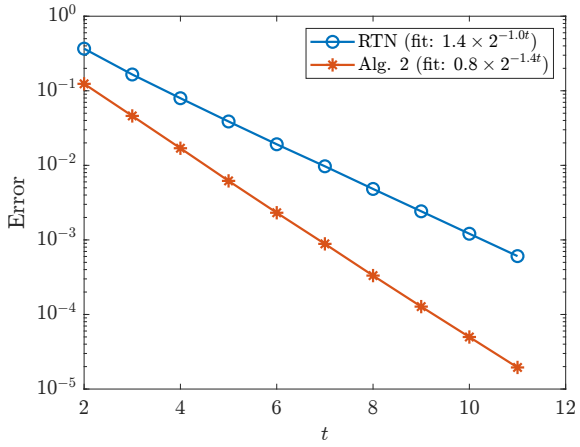

Figure 5.1 – Average quantization error for 10 random matrices of order $n = 8192$ decomposed as 13 butterfly factors, as a function of t, for the RTN baseline and our Algorithm 2 based on the optimal quantization of rank-one components.

7 Acknowledgements

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

References

[1] M. Courbariaux, Y. Bengio, and J.-P. David. BinaryConnect: Training Deep Neural Networks with binary weights during propagations. In *NeurIPS*, 2015.

[2] T. Dao and al. Learning Fast Algorithms for Linear Transforms Using Butterfly Factorizations. In *ICML*, 2019.

[3] T. Dao and al. Monarch: Expressive structured matrices for efficient and accurate training. In *ICML*, 2022.

[4] C.-P. Jeannerod and S. Rump. On relative errors of floating-point operations: optimal bounds and applications. *Math. Comp.*, 87(310):803–819, 2018.

[5] Q.-T. Le, L. Zheng, E. Riccietti, and R. Gribonval. Fast learning of fast transforms, with guarantees. In *ICASSP 2022*, 2022.

[6] L. Le Magoarou and R. Gribonval. Flexible multilayer sparse approximations of matrices and applications. *IEEE JSTSP*, 10(4):688–700, 2016.

[7] M. Nagel, M. van Baalen, T. Blankevoort, and M. Welling. Data-Free Quantization Through Weight Equalization and Bias Correction. In *ICCV*, 2019.

[8] H. Sun and al. Approximate DCT Design for Video Encoding Based on Novel Truncation Scheme. *IEEE Trans. on Circuits and Systems I: Regular Papers*, 66(4):1517–1530, April 2019.

[9] L. Zheng, E. Riccietti, and R. Gribonval. Efficient Identification of Butterfly Sparse Matrix Factorizations. *SIAM J. on Math. of Data Science*, 2022.