Proceedings of Machine Learning Research tbd:??–??, 2025 International Conference on Computational Optimization

The Ky Fan Norms and Beyond: Dual Norms and Combinations for Matrix Optimization

Alexey Kravatskiy KRAVTSKII.AIU@PHYSTECH.EDU
Moscow Institute of Physics and Technology (MIPT)

Ivan Kozyrev KOZYREV.IN@PHYSTECH.EDU
Moscow Institute of Physics and Technology (MIPT)
Marchuk Institute of Numerical Mathematics

Nikolai Kozlov KOZLOV.NA@PHYSTECH.EDU
Moscow Institute of Physics and Technology (MIPT)

Alexander Vinogradov VINOGRADOV.AM@PHYSTECH.EDU
Moscow Institute of Physics and Technology (MIPT)

Daniil Merkulov DANIIL.MERKULOV@PHYSTECH.EDU
Moscow Institute of Physics and Technology (MIPT)
Skoltech, HSE, AI4Science

Ivan Oseledets I.OSELEDETS@SKOLTECH.RU
AIRI, Skoltech
Marchuk Institute of Numerical Mathematics

Abstract

In this article, we explore the use of various matrix norms for optimizing functions of weight matrices, a crucial problem in training large language models. Moving beyond the spectral norm that underlies the Muon update, we leverage the duals to the Ky Fan k-norms to introduce a family of Muon-like algorithms we name *Fanions*, which happen to be similar to Dion. Then working with the duals of convex combinations of the Ky Fan k-norms and the Frobenius norm or the l_∞ norm, we construct the families of *F-Fanions* and *S-Fanions* respectively. Their most prominent members are *F-Muon* and *S-Muon*. We complement our theoretical analysis with an extensive empirical study of the algorithms across a wide range of tasks and settings, from which it follows that F-Muon and S-Muon are always on par with Muon, while on fine-tuning of NanoGPT and synthetic linear least squares they are even better than vanilla Muon optimizer.

1. Introduction

Minimizing loss functions in unprecedentedly high-dimensional spaces has recently become an integral and crucial part in training large language models. Hence, new scalable, time- and memory-efficient algorithms have been demanded. Besides well-known Adam (?) and AdamW (?), recently proposed Muon (?) has shown promising results on training very large models (?). Its key difference from Adam and AdamW is that it has been constructed specifically for optimizing functions of weight matrices, which are common in deep learning.

That is what can be said from a practical point of view. From a theoretical perspective, a key innovation of Muon was its principled derivation of the update rule, which emerged as the solution to an optimization problem constrained by the RMS-to-RMS norm (scaled version of the spectral norm) (**?**).

Motivated by the success of Muon, many generalizations and variations of it were proposed. Among the notable ones are Scion (**?**), Dion (**?**) and Gluon (**?**). Those works try to explain Muon's efficiency and establish convergence bounds. One central question, however, remains unanswered:

In deriving Muon's update step, why should one constrain by the spectral or any other operator norm? How would alternative norms affect performance and computational cost?

In this article, we tackle this question by actually showing that there are many viable non-operator norms. We leverage the family of norms dual to Ky Fan k-norms to derive a new family of **Fanions**, algorithms with low-rank updates. This approach theoretically explains the backbone of Dion's update (**?**) and generalizes the memory-motivated application of the nuclear norm to Sharpness-Aware Minimization (**?**). As it was done with Muon, we come up with an effective procedure for computing Fanions' updates. Lanczos algorithm, which is described and compared with its competitors in Section **??**, is the most operation-effective algorithm, which, however, for now lacks an effective GPU- and PyTorch-friendly implementation.

Working with duals to conic combinations of dual norms, we construct the families of **F-Fanions** and **S-Fanions**, which are hybrids of Muon and NormalizedSGD and SignSGD, respectively.

Then we compare the performances of the algorithm families on various model and real-world problems:

- Synthetic least squares experiment Section **??**
- CIFAR-10 airbench (**?**)
- Pre-training NanoGPT on FineWeb dataset (**?**)
- Fine-tuning NanoGPT on Tiny Stories (**?**)

Our experiments reveal important insights into the role of matrix norms in optimization. First, we show on the example of Neon, the one-rank Fanion, that not every LMO-based algorithm is effective, despite the same asymptotics in the general bounds of **?** and **?**. This suggests that existing theoretical guarantees should be reworked to explain empirical performance.

Most notably, our experiments on real-world tasks demonstrate that the choice of underlying matrix norm is remarkably flexible. On CIFAR-10 airbench, properly-tuned F-Muon and S-Muon achieve $94.02 \pm 0.13\%$ and $94.03 \pm 0.13\%$ accuracy, matching Muon's $94.01 \pm 0.13\%$ performance. On NanoGPT pre-training, F-Muon achieves 3.281 cross-entropy loss, while fully-tuned Muon achieves 3.279. Finally, S-Muon matches Muon on fine-tuning of NanoGPT on Tiny Stories, while F-Muon is far more resistant to the learning rate choice than Muon. These results show that Muon-like algorithms can maintain competitive performance even when the underlying norm constraint is significantly modified, answering affirmatively the central question posed above. Moreover, the tools from Section **??** give the researchers an unheard-of flexibility in designing algorithms that do not have to be modifications of Muon.

2. Preliminaries: Linear Minimization Oracle Framework

Training a neural network is essentially an optimization of a function of several weight matrices and a few vectors. Let us start by considering the problem of minimizing a differentiable function of a *single* matrix:

$$F(\cdot)\colon \mathbb{R}^{m\times n} \to \mathbb{R}\,, \qquad F(\boldsymbol{X}) \to \min_{\boldsymbol{X}\in\mathbb{R}^{m\times n}} \tag{1}$$

We equip the matrix space $\mathbb{R}^{m\times n}$ with a standard dot product $\langle\cdot,\cdot\; > \to \mathbb{R}$ and a norm $\|\cdot\|\colon \mathbb{R}^{m\times n} \to \mathbb{R}_+$, which does not have to coincide with the Frobenius norm $\|\cdot\|_{\mathrm{F}}$. The dual norm $\|\cdot\|^\dagger\colon \mathbb{R}^{m\times n} \to \mathbb{R}_+$ that is associated with $\|\cdot\|$ is defined as

$$\|\boldsymbol{G}\|^\dagger = \sup_{\boldsymbol{D}\in\mathbb{R}^{m\times n}\,:\|\boldsymbol{D}\|\le 1} \langle \boldsymbol{G},\boldsymbol{D} > \;. \tag{2}$$

Such problems can be solved with an iterative algorithm based on the Linear Minimization Oracle (LMO):

$$\mathrm{LMO}(\boldsymbol{M}^k) \in \arg\min_{\boldsymbol{D}\in\mathcal{S}} \langle \boldsymbol{M}^k,\boldsymbol{D} >\,, \tag{3}$$

where $\boldsymbol{M}^k$ is a gradient (or a momentum) of F in $\boldsymbol{X}^k$ and $\mathcal{S} \subset \mathbb{R}^{m\times n}$ is some set. The update of the algorithm is defined as follows:

$$\boldsymbol{X}^{k+1} = \boldsymbol{X}^k + \gamma_k \mathrm{LMO}\left(\boldsymbol{M}^k\right)\,. \tag{4}$$

We are particularly interested in the case when $\mathcal{S}$ is a unit ball in the norm $\|\cdot\|$:

$$\mathcal{S} = \mathcal{B}_{\|\cdot\|} = \{\boldsymbol{D}\in\mathbb{R}^{m\times n} \mid \|\boldsymbol{D}\| \le 1\}\,.$$

In this case,

$$\arg\min_{\boldsymbol{D}\in\mathcal{S}} \langle \boldsymbol{M}^k,\boldsymbol{D} >= -\{\boldsymbol{D}\in\mathcal{B}_1 \mid \langle \boldsymbol{M}^k,\boldsymbol{D} >= \|\boldsymbol{M}^k\|^\dagger\}\,,$$

and update for $\boldsymbol{X}^k$ in eq:simple$_{u}pdatesimplifiesto\boldsymbol{X}^{k+1} = \boldsymbol{X}^k - \gamma_k\{\boldsymbol{D}\in\mathcal{B}_1 \mid \langle \boldsymbol{M}^k,\boldsymbol{D} >= \|\boldsymbol{M}^k\|^\dagger\}$.(5)

Using this formula it is easy to compute updates for algorithms induced by various norms $\|\cdot\|$.

For example, when the norm $\|\cdot\|$ is the Frobenius norm $\|\cdot\|_{\mathrm{F}}$, eq:our$_{u}pdateturnsinto\boldsymbol{X}^{k+1} = \boldsymbol{X}^k - \eta_k \frac{\boldsymbol{M}^k}{\|\boldsymbol{M}^k\|_{\mathrm{F}}}$,(6) which recovers Normalized SGD (NSGD).

And when the norm is the spectral norm $\|\cdot\|_2$, we get

$$\boldsymbol{X}^{k+1} = \boldsymbol{X}^k - \eta_k U V^\top\,, \tag{7}$$

which is Muon without the $\sqrt{m/n}$ factor. Here, $\boldsymbol{M}^k = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\top$ is the Singular Value Decomposition (SVD) of $\boldsymbol{M}^k$ ($\boldsymbol{U} = [u_1,u_2,\ldots,u_r]$, $\boldsymbol{\Sigma} = \mathrm{diag}(\sigma_1,\sigma_2,\ldots,\sigma_r)$, and $\boldsymbol{V} = [v_1,v_2,\ldots,v_r]$). Muon can be recovered by taking the RMS-to-RMS operator norm $\sqrt{\frac{n}{m}}\|\cdot\|_2$.

When the norm is the Chebyshev norm $\|\cdot\|_{\mathrm{C}}$, we get

$$\boldsymbol{X}^{k+1} = \boldsymbol{X}^k - \eta_k\,\mathrm{sign}(\boldsymbol{M}^k)\,, \tag{8}$$

which recovers SignSGD (**?**). Here, $\mathrm{sign}(\boldsymbol{M}^k)$ denotes the element-wise sign function. SignSGD is particularly notable for its communication efficiency in distributed training, as it compresses gradients to 1-bit per parameter.