Neon: Nuclear Norm to Beat Muon

Alexey Kravatskiy
kravtskii.aiu@phystech.edu

Ivan Kozyrev
kozyrev.in@phystech.edu

Nikolay Kozlov
kozlov.na@phystech.edu

Alexander Vinogradov
vinogradov.am@phystech.edu

April 22, 2025

In this paper, we develop a new algorithm for optimization of functions of weight matrices, which are typical for training large language models. Changing spectral norm, which was used to derive Muon, to nuclear norm, we pose a new optimization problem for an update matrix, solution of which defines a novel method we name Neon. After providing theoretical guarantees of Neon convergence, we compare performances of Neon, Muon, and Adam on training multilayer perceptron and BERT vectorizer.

1 Idea

The goal of the project is to make variations on Muon to speed it up. Recently, authors of [1] have proposed to look at different optimizers as the solution of the optimization problem for an update. This approach can be utilized to derive Muon [2], a novel algorithm for fast training of neural networks. Instead of using spectral norm in the problem, we use nuclear norm to produce a new problem.

1.1 Problem

In this subsection, we provide a more detailed description of our idea and formulate it as a mathematical problem. The authors of [1] suggest obtaining the update step as a solution to the optimization problem:

$$\langle g, \delta w \rangle + \lambda \|\delta w\|^2 \to \min_{\delta w}, \tag{1}$$

where w is the weight vector, g is a gradient-like vector (e.g., obtained via momentum SGD), and $\|\cdot\|$ represents a certain norm. Many popular optimizers, such as Adam (with exponential moving average disabled) and vanilla SGD, can be cast within this framework [1].

In large language models, most weights are structured as matrices, which offers additional opportunities for optimization. Let W be the weight matrix of a linear layer, and G be a gradient-like matrix. Then, the update step δW can be obtained as a solution to the optimization problem:

$$\langle G, \delta W \rangle + \lambda \|\delta W\|^2 \to \min_{\delta W}, \tag{2}$$

where $\|\cdot\|$ denotes a certain matrix norm. By setting this norm to the RMS-to-RMS norm (a scaled version of the spectral norm), we recover the Muon optimizer [3, 1] with an update step defined by:

$$\delta W = -\frac{1}{\lambda}\sqrt{\frac{n}{m}} U V^T, \tag{3}$$

where m is the input dimension of the layer, n is the output dimension, and U and V are obtained from the singular value decomposition of the gradient matrix $G = U\Sigma V$.

Motivated by the recent achievements of the Muon optimizer (e.g., [4]), we consider alternative choices of norms, specifically the kernel norm $\|\cdot\|_*$ and a custom $F*$ norm, given by $\|X\|_{F*}^2 = (\|X\|_F + \|X\|_*)/2$, where $\|\cdot\|_F$ denotes the Frobenius norm.

Using the kernel norm in (2) leads to a rank-one update of the weight matrices:

$$\delta W = \frac{1}{\lambda} u_1 \sigma_1 v_1^T \,, \tag{4}$$

where σ_1 is the largest singular value, and u_1 and v_1 are the corresponding singular vectors. We expect one iteration of this method to be significantly faster than one iteration of Muon.

Another choice is the $F*$ norm. With this choice, (2) yields

$$\delta W = UDV^T \tag{5}$$

with $D = \mathrm{diag}(d_i)$, where $d_i = [\sigma_i - \tau]_+$ and τ is given by:

$$\sum_{i=1}^{n} [\sigma_i - \tau]_+ = \tau \,. \tag{6}$$

We anticipate that the method with this update step will perform well with large batch sizes.

In this article we show how one can quickly compute weight updates defined by (4) and (5). Then we finilize the methods by adding momentum and test their performance against those of Muon ant training multilayer perceptron and transformer. The results will be fast algorithms, which we will convert into a new optimizer classes for PyTorch, as was done with Muon.

2 Outcomes

3 Literature review

4 Quality metrics

5 Preliminary plan

6 Prototyping phase report

References

[1] Jeremy Bernstein and Laker Newhouse. Old optimizer, new norm: An anthology. *arXiv preprint arXiv:2409.20325*, 2024.

[2] Keller Jordan, Yuchen Jin, Vlado Boza, Jiacheng You, Franz Cesista, Laker Newhouse, and Jeremy Bernstein. Muon: An optimizer for hidden layers in neural networks, 2024.

[3] Jeremy Bernstein. Deriving muon, 2025.

[4] Jingyuan Liu, Jianlin Su, Xingcheng Yao, Zhejun Jiang, Guokun Lai, Yulun Du, Yidao Qin, Weixin Xu, Enzhe Lu, Junjie Yan, et al. Muon is scalable for llm training. *arXiv preprint arXiv:2502.16982*, 2025.