# Rethinking Learning Rate and Batch Size (Part 3): Muon

Jianlin Su

September 15, 2025

In the previous two articles, "Rethinking Learning Rate and Batch Size (Part 1): Status Quo" and "Rethinking Learning Rate and Batch Size (Part 2): Mean Field", we primarily proposed the mean-field method to simplify calculations related to learning rate and batch size. At that time, the optimizers we analyzed were SGD, SignSGD, and SoftSignSGD, and our main goal was simplification; essentially, no new conclusions were drawn.

However, in today's "feast of optimizers," how could Muon be left out? Therefore, in this article, we will attempt to calculate the relevant conclusions for Muon to see if the relationship between its learning rate and batch size exhibits any new patterns.

## 1 Basic Notation

As is well known, the main characteristic of Muon is its non-element-wise update rule. Consequently, the element-wise calculation methods used previously in "How Should the Learning Rate Change as the Batch Size Increases?" and "How Does Adam's Epsilon Affect the Scaling Law of Learning Rate?" are completely inapplicable. Fortunately, the mean-field method introduced in the previous article remains effective, requiring only a slight adjustment of details.

First, let us introduce some notation. Let the loss function be $\mathcal{L}(\boldsymbol{W})$, where $\boldsymbol{W} \in \mathbb{R}^{n\times m}$ is a matrix (assume $n \geq m$). Let $\boldsymbol{G}$ be its gradient. The gradient of a single sample is denoted as $\tilde{\boldsymbol{G}}$, its mean is $\boldsymbol{G}$, and its variance is $\sigma^2$. When the batch size is $B$, the gradient is denoted as $\tilde{\boldsymbol{G}}_B$; its mean remains $\boldsymbol{G}$, but its variance becomes $\sigma^2/B$. Note that the variance here is treated as a scalar $\sigma^2$, rather than considering the full covariance matrix as done previously.

The core reason for this simplification is that the random variable itself is already a matrix, so its corresponding covariance matrix would actually be a 4th-order tensor, which is cumbersome to discuss. Does simplifying it to a single scalar significantly sacrifice accuracy? In fact, it does not. Although we considered the full covariance matrix $\boldsymbol{\Sigma}$ in the previous two articles, a closer look reveals that the final results only depend on $\mathrm{tr}(\boldsymbol{\Sigma})$, which is equivalent to simplifying it to a scalar from the beginning.

## 2 Hessian Matrix

Similarly, let the update amount be $-\eta\tilde{\boldsymbol{\Phi}}_B$. Consider the second-order expansion of the loss function:

$$\mathcal{L}(\boldsymbol{W} - \eta\tilde{\boldsymbol{\Phi}}_B) \approx \mathcal{L}(\boldsymbol{W}) - \eta\,\mathrm{tr}(\tilde{\boldsymbol{\Phi}}_B^{\top}\boldsymbol{G}) + \frac{1}{2}\eta^2\,\mathrm{tr}(\tilde{\boldsymbol{\Phi}}_B^{\top}\boldsymbol{H}\tilde{\boldsymbol{\Phi}}_B) \tag{1}$$

The first two terms should be straightforward; the third term is more difficult to understand. Like the covariance matrix, the Hessian matrix $\boldsymbol{H}$ here is a 4th-order tensor, which is complex to interpret.

The simplest entry point here is the linear operator perspective, i.e., treating $\boldsymbol{H}$ as a linear operator where both input and output are matrices. We do not need to know what $\boldsymbol{H}$ looks like or how $\boldsymbol{H}$ operates with $\tilde{\boldsymbol{\Phi}}_B$; we only need to know that $\boldsymbol{H}\tilde{\boldsymbol{\Phi}}_B$ is linear with respect to

$\tilde{\boldsymbol{\Phi}}_B$. In this way, the objects we handle remain matrices, avoiding additional cognitive load. Any linear operator that satisfies the conditions can serve as an approximation of the Hessian matrix, without needing to write out the specific high-order tensor form.

The protagonist of this article is Muon. we take $\tilde{\boldsymbol{\Phi}}_B = \operatorname{msign}(\tilde{\boldsymbol{G}}_B)$ as its approximation for calculation. By definition, we write $\operatorname{msign}(\tilde{\boldsymbol{G}}_B) = \tilde{\boldsymbol{G}}_B(\tilde{\boldsymbol{G}}_B^\top \tilde{\boldsymbol{G}}_B)^{-1/2}$. From a Newton's method perspective, this is equivalent to assuming $\boldsymbol{H}^{-1}\boldsymbol{X} = \eta_{\max}\boldsymbol{X}(\boldsymbol{G}^\top\boldsymbol{G})^{-1/2}$, which implies $\boldsymbol{H}\boldsymbol{X} = \eta_{\max}^{-1}\boldsymbol{X}(\boldsymbol{G}^\top\boldsymbol{G})^{1/2}$. This will be used in subsequent calculations.

## 3 Calculating Expectation

Taking the expectation of both sides of Eq. (1), we get:

$$\mathbb{E}[\mathcal{L}(\boldsymbol{W} - \eta\tilde{\boldsymbol{\Phi}}_B)] \approx \mathcal{L}(\boldsymbol{W}) - \eta \operatorname{tr}(\mathbb{E}[\tilde{\boldsymbol{\Phi}}_B]^\top \boldsymbol{G}) + \frac{1}{2}\eta^2 \mathbb{E}[\operatorname{tr}(\tilde{\boldsymbol{\Phi}}_B^\top \boldsymbol{H}\tilde{\boldsymbol{\Phi}}_B)] \tag{2}$$

First, calculate $\mathbb{E}[\tilde{\boldsymbol{\Phi}}_B]$:

$$\mathbb{E}[\tilde{\boldsymbol{\Phi}}_B] = \mathbb{E}[\tilde{\boldsymbol{G}}_B(\tilde{\boldsymbol{G}}_B^\top\tilde{\boldsymbol{G}}_B)^{-1/2}] \approx \mathbb{E}[\tilde{\boldsymbol{G}}_B](\mathbb{E}[\tilde{\boldsymbol{G}}_B^\top\tilde{\boldsymbol{G}}_B])^{-1/2} = \boldsymbol{G}(\mathbb{E}[\tilde{\boldsymbol{G}}_B^\top\tilde{\boldsymbol{G}}_B])^{-1/2} \tag{3}$$

We write out $\mathbb{E}[\tilde{\boldsymbol{G}}_B^\top\tilde{\boldsymbol{G}}_B]$ by components and assume independence between different components:

$$\mathbb{E}[\tilde{\boldsymbol{G}}_B^\top\tilde{\boldsymbol{G}}_B]_{i,j} = \mathbb{E}\left[\sum_{k=1}^n (\tilde{G}_B)_{k,i}(\tilde{G}_B)_{k,j}\right] = \begin{cases} \mathbb{E}\left[\sum_{k=1}^n (\tilde{G}_B)_{k,i}^2\right] = \left(\sum_{k=1}^n G_{k,i}^2\right) + n\sigma^2/B, & (i = j) \\ \sum_{k=1}^n \mathbb{E}[(\tilde{G}_B)_{k,i}]\mathbb{E}[(\tilde{G}_B)_{k,j}] = \sum_{k=1}^n G_{k,i}G_{k,j}, & (i \neq j) \end{cases} \tag{4}$$

Combining these, we have $\mathbb{E}[\tilde{\boldsymbol{G}}_B^\top\tilde{\boldsymbol{G}}_B] = \boldsymbol{G}^\top\boldsymbol{G} + (n\sigma^2/B)\boldsymbol{I}$, so:

$$\mathbb{E}[\tilde{\boldsymbol{\Phi}}_B] \approx \boldsymbol{G}(\boldsymbol{G}^\top\boldsymbol{G} + (n\sigma^2/B)\boldsymbol{I})^{-1/2} = \operatorname{msign}(\boldsymbol{G})(\boldsymbol{I} + (n\sigma^2/B)(\boldsymbol{G}^\top\boldsymbol{G})^{-1})^{-1/2} \tag{5}$$

To further simplify the dependency on $B$, we approximate $\boldsymbol{G}^\top\boldsymbol{G}$ with $\operatorname{tr}(\boldsymbol{G}^\top\boldsymbol{G})\boldsymbol{I}/m$, which means keeping only the diagonal part of $\boldsymbol{G}^\top\boldsymbol{G}$ and then replacing the diagonal elements with their average. Thus, we obtain:

$$\mathbb{E}[\tilde{\boldsymbol{\Phi}}_B] \approx \operatorname{msign}(\boldsymbol{G})(1 + \mathcal{B}_{\text{simple}}/B)^{-1/2} \tag{6}$$

where $\mathcal{B}_{\text{simple}} = mn\sigma^2/\operatorname{tr}(\boldsymbol{G}^\top\boldsymbol{G}) = mn\sigma^2/\|\boldsymbol{G}\|_F$. This is actually the same as treating $\boldsymbol{G}$ as a vector and calculating $\mathcal{B}_{\text{simple}}$ as in the previous two articles. The form of the above equation is identical to that of SignSGD. From this, we can guess that Muon will not present many new results regarding the relationship between learning rate and batch size.

## 4 Same Patterns

As for $\mathbb{E}[\operatorname{tr}(\tilde{\boldsymbol{\Phi}}_B^\top\boldsymbol{H}\tilde{\boldsymbol{\Phi}}_B)]$, we only calculate the assumption corresponding to Muon derived earlier, namely $\boldsymbol{H}\boldsymbol{X} = \eta_{\max}^{-1}\boldsymbol{X}(\boldsymbol{G}^\top\boldsymbol{G})^{1/2}$. Then:

$$\operatorname{tr}(\tilde{\boldsymbol{\Phi}}_B^\top\boldsymbol{H}\tilde{\boldsymbol{\Phi}}_B) = \eta_{\max}^{-1}\operatorname{tr}(\tilde{\boldsymbol{\Phi}}_B^\top\tilde{\boldsymbol{\Phi}}_B(\boldsymbol{G}^\top\boldsymbol{G})^{1/2}) \tag{7}$$

Note that $\tilde{\boldsymbol{\Phi}}_B$ is the result of msign, so it must be an orthogonal matrix (full rank), which means $\tilde{\boldsymbol{\Phi}}_B^\top\tilde{\boldsymbol{\Phi}}_B = \boldsymbol{I}$. In this case, $\operatorname{tr}(\tilde{\boldsymbol{\Phi}}_B^\top\boldsymbol{H}\tilde{\boldsymbol{\Phi}}_B)$ is a fixed constant $\eta_{\max}^{-1}\operatorname{tr}((\boldsymbol{G}^\top\boldsymbol{G})^{1/2}) = \eta_{\max}^{-1}\operatorname{msign}(\boldsymbol{G})^\top\boldsymbol{G}$. Thus, we can obtain:

$$\eta^* \approx \frac{\operatorname{tr}(\mathbb{E}[\tilde{\boldsymbol{\Phi}}_B]^\top\boldsymbol{G})}{\mathbb{E}[\operatorname{tr}(\tilde{\boldsymbol{\Phi}}_B^\top\boldsymbol{H}\tilde{\boldsymbol{\Phi}}_B)]} \approx \frac{\eta_{\max}}{\sqrt{1 + \mathcal{B}_{\text{simple}}/B}} \tag{8}$$

As expected, the form is exactly the same as the result for SignSGD, with no new patterns.

Actually, upon reflection, this is reasonable. SignSGD directly applies sign to the gradient, while Muon's msign applies sign to the singular values. Intuitively, it is equivalent to applying sign in a different coordinate system. It brings a new matrix update rule, but the learning rate $\eta^*$ and batch size $B$ are just scalars. Given that the core of both is sign, it is highly likely that the asymptotic relationship of these scalars will not undergo significant changes.

Of course, we have only calculated for a specific $\boldsymbol{H}$. If a more general $\boldsymbol{H}$ is considered, it is possible that, like SignSGD, a "Surge" phenomenon might occur where "as the batch size increases, the learning rate should instead decrease." However, as mentioned in the "Reflections on Causes" section of the previous article, if a Surge phenomenon is truly observed, it might be more appropriate to change the optimizer rather than correcting the relationship between $\eta^*$ and $B$.

## 5 Summary

In this article, we attempted a simple analysis of Muon using the mean-field approximation. The conclusion is that its relationship between learning rate and batch size is consistent with SignSGD, with no new patterns discovered.