Why Does the Official Muon Have an Extra $\max(1, \cdot)$ Compared to the MuP Version?

Su Jianlin

2026-06-03

Why Does the Official Muon Have an Extra $\max(1, \cdot)$ Compared to the MuP Version?

In the article *Muon Optimizer Guide: Quick Start and Key Details*, we listed several versions of Muon, which differ in scaling factors related to the matrix shape of the learning rate. Among them, the "official version (KellerJordan version)" differs from the "MuP version" only by an additional $\max(1, \cdot)$ truncation operation. This article will specifically discuss where this truncation comes from.

1 Several Versions

The update rule for Muon can be uniformly written as

$$\begin{aligned} \boldsymbol{M}_t &= \beta \boldsymbol{M}_{t-1} + \boldsymbol{G}_t \\ \boldsymbol{W}_t &= \boldsymbol{W}_{t-1} - \eta_t(\alpha\, \mathrm{msign}(\boldsymbol{M}_t) + \lambda \boldsymbol{W}_{t-1}) \end{aligned} \tag{1}$$

The differences among the versions lie in α, which are:

$$\alpha = \begin{cases} 1 & \text{(Naive Version)} \\ \sqrt{\max(1, d_{out}/d_{in})} & \text{(KellerJordan Version)} \\ \sqrt{d_{out}/d_{in}} & \text{(MuP Version)} \\ 0.2 \times \sqrt{\max(d_{out}, d_{in})} & \text{(Moonlight Version)} \end{cases}$$

Here, matrix $\boldsymbol{W} \in \mathbb{R}^{d_{in} \times d_{out}}$ represents the trainable parameters of a linear layer $\boldsymbol{y} = \boldsymbol{x}\boldsymbol{W}$, with input $\boldsymbol{x} \in \mathbb{R}^{d_{in}}$ being a row vector.

We mainly focus on the "KellerJordan version" and the "MuP version". The former adds a $\max(1, \cdot)$ on top of the latter. According to our analysis in articles such as *Higher-Order MuP: Simpler but Smarter Spectral Condition Scaling* and *Beyond MuP: 2. Linear Layers and Steepest Descent*, under spectral condition constraints related to MuP, the steepest descent should correspond exactly to the MuP version of Muon. How then should we interpret the extra $\max(1, \cdot)$?

2 Feature Increment

For simplicity, we omit the subscript t in the following discussion. Without loss of generality, assume the momentum $\boldsymbol{M}$ is full-rank. Then $\boldsymbol{\Phi} = \text{msign}(\boldsymbol{M})$ has all singular values equal to 1. Thus, when $d_{in} \leq d_{out}$, $\boldsymbol{\Phi}\boldsymbol{\Phi}^\top = \boldsymbol{I}_{d_{in}}$; when $d_{in} > d_{out}$, $\boldsymbol{\Phi}^\top\boldsymbol{\Phi} = \boldsymbol{I}_{d_{out}}$.

Let $\Delta\boldsymbol{W} = \eta\alpha\boldsymbol{\Phi}$. Our task is to find the relationship between α and d_{in}, d_{out}. As discussed in *Why Do We Prefer Isotropy? A Steepest Descent Perspective*, the parameters are actually by-products of the model; changes at the feature level might be more fundamental. Translating $\Delta\boldsymbol{W}$ to the feature level gives $\Delta\boldsymbol{y} = \boldsymbol{x}\Delta\boldsymbol{W} = \eta\alpha\boldsymbol{x}\boldsymbol{\Phi}$, so $\|\Delta\boldsymbol{y}\|_{RMS} = \eta\alpha\|\boldsymbol{x}\boldsymbol{\Phi}\|_{RMS}$.

Now we consider cases separately. First, when $d_{in} \leq d_{out}$, $\boldsymbol{\Phi}$ can be written as $\boldsymbol{U}[\boldsymbol{I}_{d_{in}}, \boldsymbol{0}_{d_{in}\times(d_{out}-d_{in})}]\boldsymbol{V}^\top$, where $\boldsymbol{U} \in \mathbb{R}^{d_{in}\times d_{in}}$, $\boldsymbol{V} \in \mathbb{R}^{d_{out}\times d_{out}}$ are orthogonal matrices. Then

$$\begin{aligned}
\|\Delta\boldsymbol{y}\|_{RMS} &= \eta\alpha\big\|\boldsymbol{x}\boldsymbol{U}[\boldsymbol{I}_{d_{in}}, \boldsymbol{0}_{d_{in}\times(d_{out}-d_{in})}]\boldsymbol{V}^\top\big\|_{RMS} \\
&= \eta\alpha\big\|\boldsymbol{x}\boldsymbol{U}[\boldsymbol{I}_{d_{in}}, \boldsymbol{0}_{d_{in}\times(d_{out}-d_{in})}]\big\|_{RMS} \\
&= \eta\alpha\big\|[\boldsymbol{x}\boldsymbol{U}, \boldsymbol{0}_{d_{out}-d_{in}}]\big\|_{RMS} \\
&= \eta\alpha\sqrt{\frac{d_{in}}{d_{out}}}\|\boldsymbol{x}\boldsymbol{U}\|_{RMS} \\
&= \eta\alpha\sqrt{\frac{d_{in}}{d_{out}}}\|\boldsymbol{x}\|_{RMS}
\end{aligned} \tag{2}$$

Note all steps are equalities. Therefore, by setting $\alpha = \sqrt{d_{out}/d_{in}}$, we ensure that the RMS of *every* $\Delta\boldsymbol{y}$ is $\eta\|\boldsymbol{x}\|_{RMS}$, i.e., the relative update magnitude is consistent across all tokens.

3 Isotropy

Unfortunately, for the second case $d_{in} > d_{out}$, the above goal of "perfect consistency" cannot be achieved. Specifically, in this case, the SVD of $\boldsymbol{\Phi}$ is written as $\boldsymbol{U}\begin{bmatrix}\boldsymbol{I}_{d_{out}} \\ \boldsymbol{0}_{(d_{in}-d_{out})\times d_{out}}\end{bmatrix}\boldsymbol{V}^\top$.

Thus,

$$\begin{aligned}\|\Delta \boldsymbol{y}\|_{RMS} &= \eta\alpha \left\| \boldsymbol{x}\boldsymbol{U} \begin{bmatrix} \boldsymbol{I}_{d_{out}} \\ \boldsymbol{0}_{(d_{in}-d_{out})\times d_{out}} \end{bmatrix} \boldsymbol{V}^\top \right\|_{RMS} \\ &= \eta\alpha \left\| \boldsymbol{x}\boldsymbol{U} \begin{bmatrix} \boldsymbol{I}_{d_{out}} \\ \boldsymbol{0}_{(d_{in}-d_{out})\times d_{out}} \end{bmatrix} \right\|_{RMS} \\ &= \eta\alpha \left\| (\boldsymbol{x}\boldsymbol{U})_{[:d_{out}]} \right\|_{RMS}\end{aligned} \tag{3}$$

Here, $\boldsymbol{x}\boldsymbol{U}$ is a d_{in}-dimensional vector with $d_{in} > d_{out}$, so $(\boldsymbol{x}\boldsymbol{U})_{[:d_{out}]}$ only takes the first d_{out} dimensions to compute the RMS. Its RMS is not fixedit can be as large as $\sqrt{d_{in}/d_{out}}\|\boldsymbol{x}\|_{RMS}$ (worst case) and as small as 0.

Since orthogonal matrices preserve RMS, $\|\boldsymbol{x}\boldsymbol{U}\|_{RMS} = \|\boldsymbol{x}\|_{RMS}$. When the distribution of $\boldsymbol{x}$ is sufficiently isotropic, we can assume that each component of $\boldsymbol{x}\boldsymbol{U}$ has an average magnitude of $\|\boldsymbol{x}\|_{RMS}$. Thus, taking the RMS of the first d_{out} components would also be approximately $\|\boldsymbol{x}\|_{RMS}$, meaning $\|\Delta \boldsymbol{y}\|_{RMS} \approx \eta\alpha\|\boldsymbol{x}\|_{RMS}$. Therefore, setting $\alpha = 1$ achieves an effect similar to the previous section.

4 Anisotropy

Combining the results from the previous two sections, we obtain

$$\alpha = \sqrt{\max\left(1, \frac{d_{out}}{d_{in}}\right)} \tag{4}$$

This is precisely the $\max(1, \cdot)$ that appears in the KellerJordan version of Muon.

However, the conclusion in the previous section relies on the assumption that the input $\boldsymbol{x}$ is sufficiently isotropic. This assumption may approximately hold in the early stages of training, but as training progresses, the feature distribution becomes increasingly anisotropic, tending toward the "worst case" that maximizes $\|\Delta \boldsymbol{y}\|_{RMS}$. In such cases, the average approximation $\|\Delta \boldsymbol{y}\|_{RMS} \approx \eta\alpha\|\boldsymbol{x}\|_{RMS}$ becomes less accurate, while the maximum value $\eta\alpha\sqrt{d_{in}/d_{out}}\|\boldsymbol{x}\|_{RMS}$ becomes more accurate.

Under these conditions, the α that maintains $\|\Delta \boldsymbol{y}\|_{RMS} \approx \eta\|\boldsymbol{x}\|_{RMS}$ is $\sqrt{d_{out}/d_{in}}$, which is consistent with the result when $d_{in} \leq d_{out}$, yielding the MuP version again. That is, during the middle and later stages of training, the MuP version of Muon is more appropriate.

To resolve this inconsistency, we have two strategies:

1. Use the MuP version of Muon throughout training. This may slightly reduce convergence speed in the early phase, but the middle and later stages are the "most critical" part of training.

2. Modify the scaling factor to

$$\alpha = \sqrt{\max\left(\tau_t, \frac{d_{out}}{d_{in}}\right)} \tag{5}$$

where τ_t monotonically decays from 1 to 0. This achieves a smooth transition from the KellerJordan version to the MuP version, at the cost of introducing an additional schedule to tune.

5 Summary

This article mainly explains the origin of the $\max(1, \cdot)$ in the KellerJordan version from the perspective of uniformity in "feature increments".