

Why is Adam's Update RMS 0.2?

Su Jianlin

September 2, 2025

As is well known, we began experimenting with using Muon for large-scale LLM training quite early on. Specifically, in the post "Muon Sequel: Why We Chose to Try Muon?", we proposed the "Match Adam Update RMS" trick to facilitate a quick migration from Adam to Muon. This technique was also applied in the training of Kimi K2. The trick involves unifying Muon's Update RMS to 0.2, which allows us to reuse Adam's learning rate and weight decay rate.

Behind this trick is our observation that Adam's Update RMS is approximately equal to 0.2, and this phenomenon is stable and reproducible. This raises an interesting question: Why is Adam's Update RMS 0.2? Can we explain it theoretically?

1 Introduction to the Problem

First, let's describe the phenomenon: from experiments, we observe that roughly after the warmup ends and the model enters formal training, Adam's Update RMS almost always stays between 0.2 and 0.3. This pattern is consistent across models of different sizes. The commonality among these models is that they are all trained with Adam using parameters $\beta_1 = 0.9, \beta_2 = 0.95$. Since this commonality is so obvious, it is likely not a coincidence. Therefore, I attempted to analyze the underlying principle.

Let's review the form of the Adam optimizer:

$$\text{Adam}^{\text{W}} := \begin{cases} \mathbf{m}_t = \beta_1 \mathbf{m}_{t-1} + (1 - \beta_1) \mathbf{g}_t \\ \mathbf{v}_t = \beta_2 \mathbf{v}_{t-1} + (1 - \beta_2) \mathbf{g}_t^2 \\ \hat{\mathbf{m}}_t = \mathbf{m}_t / (1 - \beta_1^t) \\ \hat{\mathbf{v}}_t = \mathbf{v}_t / (1 - \beta_2^t) \\ \mathbf{u}_t = \hat{\mathbf{m}}_t / (\sqrt{\hat{\mathbf{v}}_t} + \epsilon) \\ \boldsymbol{\theta}_t = \boldsymbol{\theta}_{t-1} - \eta_t (\mathbf{u}_t + \lambda_t \boldsymbol{\theta}_{t-1}) \end{cases} \quad (1)$$

Note: In this article, all vector multiplications and divisions, including squares, refer to the Hadamard product/quotient (element-wise operations).

Our goal is to prove that $\|\mathbf{u}_t\|_{RMS} \approx 0.2$, at least under the setting $\beta_1 = 0.9, \beta_2 = 0.95$. We assume ϵ is small enough to be ignored, and we consider the steady state as $t \rightarrow \infty$. In this state, β_1^t and β_2^t are close to zero, so we do not need to distinguish between $\mathbf{m}_t$ and $\hat{\mathbf{m}}_t$, or $\mathbf{v}_t$ and $\hat{\mathbf{v}}_t$. Thus, we have $\mathbf{u}_t = \mathbf{m}_t / \sqrt{\mathbf{v}_t}$.

For $\mathbf{m}_t$ and $\mathbf{v}_t$, we can obtain the expansion:

$$\mathbf{m}_t = (1 - \beta_1) \sum_{i=1}^t \beta_1^{t-i} \mathbf{g}_i, \quad \mathbf{v}_t = (1 - \beta_2) \sum_{i=1}^t \beta_2^{t-i} \mathbf{g}_i^2 \quad (2)$$

2 Numerical Simulation

If we assume that $\mathbf{g}_1, \mathbf{g}_2, \dots, \mathbf{g}_t$ are sampled from the same distribution, we can directly use numerical simulation to estimate $\|\mathbf{u}_t\|_{RMS}$. Let's try the simplest case using the standard normal distribution $\mathcal{N}(\mathbf{0}, \mathbf{I})$. The reference code is as follows:

```
1 import numpy as np
2
3 N, T = 10000, 2000
4 beta1, beta2 = 0.9, 0.95
5 m, v = 0, 0
6 for t in range(1, T + 1):
7     g = np.random.randn(N)
8     m = beta1 * m + (1 - beta1) * g
9     v = beta2 * v + (1 - beta2) * g**2
10    u = m / v**0.5
11
12 rms = (u**2).mean()*0.5
13 print(rms)
```

Can you guess the result? The answer is approximately 0.225, which is strikingly similar to the experimental results! This suggests that our simulation assumptions align well with reality. Some readers might think this is incorrect—isn't $\mathbf{g} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ just pure noise? How can it match? Actual training is certainly not pure noise, but we can say that the signal-to-noise ratio (SNR) of a single gradient is extremely small, so it can be modeled as pure noise.

Readers can experiment with the code above to observe the variables affecting Update RMS. The general conclusions are: Update RMS is positively correlated with β_1 , seems largely independent of β_2 , and if the distribution of $\mathbf{g}$ has a non-zero mean (equivalent to increasing the SNR of the gradient), the Update RMS will increase.

3 Mean-Field Approximation

In this section, I will attempt to derive an approximate analytical solution for the simulation results. First, from the definition of RMS, to find $\|\mathbf{u}_t\|_{RMS}$, we first need $\mathbf{u}_t^2 = \mathbf{m}_t^2 / \mathbf{v}_t$. My idea is to use the expectation of $\mathbf{u}_t^2$ as an approximation and further transform it into a mean-field approximation:

$$\mathbb{E}[\mathbf{u}_t^2] = \mathbb{E}\left[\frac{\mathbf{m}_t^2}{\mathbf{v}_t}\right] \approx \frac{\mathbb{E}[\mathbf{m}_t^2]}{\mathbb{E}[\mathbf{v}_t]} \quad (3)$$

Some readers might question the validity of the last approximation step. My suggestion is to ignore these details for now—much like assuming $\mathbf{g} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ in the previous section—and calculate it first. If the result is reasonable, then the process must be reasonable to some extent. Now we calculate the numerator and denominator separately. We generally set $\mathbb{E}[\mathbf{g}] = \boldsymbol{\mu}$ and $\mathbb{E}[\mathbf{g}^2] = \boldsymbol{\mu}^2 + \boldsymbol{\sigma}^2$. The denominator is simpler:

$$\begin{aligned} \mathbb{E}[\mathbf{v}_t] &= (1 - \beta_2) \sum_{i=1}^t \beta_2^{t-i} \mathbb{E}[\mathbf{g}_i^2] \\ &= (1 - \beta_2) \sum_{i=1}^t \beta_2^{t-i} (\boldsymbol{\mu}^2 + \boldsymbol{\sigma}^2) \\ &= (1 - \beta_2^t) (\boldsymbol{\mu}^2 + \boldsymbol{\sigma}^2) \\ &\approx \boldsymbol{\mu}^2 + \boldsymbol{\sigma}^2 \quad (t \rightarrow \infty) \end{aligned} \quad (4)$$

As for the numerator, we can expand the square directly or take a shortcut: we need the second moment of $\mathbf{m}_t$, $\mathbb{E}[\mathbf{m}_t^2]$, which equals $\mathbb{E}[\mathbf{m}_t]^2 + \text{Var}[\mathbf{m}_t]$. The calculation for $\mathbb{E}[\mathbf{m}_t]$ is similar to

$\mathbb{E}[\mathbf{v}_t]$, resulting in $(1 - \beta_1^t)\boldsymbol{\mu} \approx \boldsymbol{\mu}$. As for the variance, it possesses square additivity, so:

$$\text{Var}[\mathbf{m}_t] = (1 - \beta_1)^2 \sum_{i=1}^t \beta_1^{2(t-i)} \boldsymbol{\sigma}^2 = \frac{(1 - \beta_1)^2 (1 - \beta_1^{2t})}{1 - \beta_1^2} \boldsymbol{\sigma}^2 \approx \frac{1 - \beta_1}{1 + \beta_1} \boldsymbol{\sigma}^2 \quad (t \rightarrow \infty) \quad (5)$$

Therefore:

$$\mathbb{E}[\mathbf{u}_t^2] \approx \frac{\boldsymbol{\mu}^2 + \frac{1-\beta_1}{1+\beta_1} \boldsymbol{\sigma}^2}{\boldsymbol{\mu}^2 + \boldsymbol{\sigma}^2} \quad (6)$$

4 Analysis of Results

Since $\mathbb{E}[\mathbf{u}_t^2]$ is already a squared vector, to estimate $\|\mathbf{u}_t\|_{RMS}$, we only need to average the components and take the square root. For the averaging step, let's apply the mean-field approximation again (averaging the numerator and denominator separately), which yields:

$$\|\mathbf{u}_t\|_{RMS} \approx \sqrt{\frac{\|\boldsymbol{\mu}\|^2 + \frac{1-\beta_1}{1+\beta_1} \|\boldsymbol{\sigma}\|^2}{\|\boldsymbol{\mu}\|^2 + \|\boldsymbol{\sigma}\|^2}} = \sqrt{\frac{\|\boldsymbol{\mu}\|^2 / \|\boldsymbol{\sigma}\|^2 + \frac{1-\beta_1}{1+\beta_1}}{\|\boldsymbol{\mu}\|^2 / \|\boldsymbol{\sigma}\|^2 + 1}} \quad (7)$$

There are two influencing factors: first, $\|\boldsymbol{\mu}\|^2 / \|\boldsymbol{\sigma}\|^2$, which can be viewed as the Signal-to-Noise Ratio (SNR) of the gradient; second, β_1 , which is one of Adam's hyperparameters. Notably, the result does not depend on β_2 , which matches our previous simulation. How good is this approximation? Let's consider the simplest case where $\boldsymbol{\mu} = \mathbf{0}$:

$$\|\mathbf{u}_t\|_{RMS} \approx \sqrt{\frac{1 - \beta_1}{1 + \beta_1}} \quad (8)$$

Substituting $\beta_1 = 0.9$, the result is $0.2294\dots$, which matches both simulation results and practical performance very well! Furthermore, here are several comparisons with simulation results:

[Click to view original SVG: Simulation results vs Mean-field approximation (different beta1, beta2)]

Figure 1: Simulation results vs Mean-field approximation (different β_1, β_2)

Overall, the degree of approximation is quite good, especially for $\beta_2 \geq 0.9$, where the results almost overlap with the mean-field approximation (as pointed out by @EIFY, the paper "Rotational Equilibrium: How Weight Decay Balances Learning Across Neural Networks" also obtained the same calculation results).

The comparison considering SNR is as follows:

[Click to view original SVG: Simulation results vs Mean-field approximation (different beta1, SNR)]

Figure 2: Simulation results vs Mean-field approximation (different β_1 , SNR)

As the SNR increases, the error of the mean-field approximation begins to grow, but it still predicts the overall trend. In fact, in actual training, the gradient SNR rarely reaches as high as 1, so the mean-field approximation remains a good estimate.

5 Inverse Prediction

If we accept the mean-field approximation (7), we can use it in reverse to estimate the gradient SNR:

$$\frac{\|\boldsymbol{\mu}\|^2}{\|\boldsymbol{\sigma}\|^2} \approx \frac{\|\mathbf{u}_t\|_{RMS}^2 - \frac{1-\beta_1}{1+\beta_1}}{1 - \|\mathbf{u}_t\|_{RMS}^2} \quad (9)$$

In actual training, β_1 is given, and $\|\mathbf{u}_t\|_{RMS}$ (Adam's Update RMS) can be directly estimated, so the above formula is computable. Of course, this formula only applies to Adam. Is there a more general estimation approach? Yes! Recall our earlier estimate:

$$\mathbb{E}[\mathbf{m}_t^2] \approx \boldsymbol{\mu}^2 + \frac{1-\beta_1}{1+\beta_1} \boldsymbol{\sigma}^2 \quad (10)$$

Summing its components and taking the square root, we consider it an approximation of $\|\mathbf{m}_t\|$:

$$\|\mathbf{m}_t\| \approx \sqrt{\|\boldsymbol{\mu}\|^2 + \frac{1-\beta_1}{1+\beta_1} \|\boldsymbol{\sigma}\|^2} \quad (11)$$

The second moment is $\mathbb{E}[\mathbf{v}_t] \approx \boldsymbol{\mu}^2 + \boldsymbol{\sigma}^2$. While optimizers like Muon do not have a second moment available, we notice that the second moment result is independent of β_2 . Thus, we can consider the simplest case— $\beta_2 = 0$ —where $\mathbf{v}_t = \mathbf{g}_t^2$. This might be a bit forced, but for estimation, we choose the most convenient path. This "approximation" implies $\|\mathbf{g}_t\|^2 \approx \|\boldsymbol{\mu}\|^2 + \|\boldsymbol{\sigma}\|^2$, leading to:

$$\frac{\|\mathbf{m}_t\|}{\|\mathbf{g}_t\|} \approx \sqrt{\frac{\|\boldsymbol{\mu}\|^2 + \frac{1-\beta_1}{1+\beta_1} \|\boldsymbol{\sigma}\|^2}{\|\boldsymbol{\mu}\|^2 + \|\boldsymbol{\sigma}\|^2}} \quad (12)$$

The right side is identical in form to equation (7), so we can write:

$$\frac{\|\boldsymbol{\mu}\|^2}{\|\boldsymbol{\sigma}\|^2} \approx \frac{\|\mathbf{m}_t\|^2 / \|\mathbf{g}_t\|^2 - \frac{1-\beta_1}{1+\beta_1}}{1 - \|\mathbf{m}_t\|^2 / \|\mathbf{g}_t\|^2} \quad (13)$$

In other words, using $\|\mathbf{m}_t\|/\|\mathbf{g}_t\|$ to replace $\|\mathbf{u}_t\|_{RMS}$ provides a general way to estimate $\|\boldsymbol{\mu}\|^2/\|\boldsymbol{\sigma}\|^2$ for optimizers with momentum. Some might ask what to do if there is no momentum? In that case, there is truly no way, because $\|\boldsymbol{\mu}\|^2/\|\boldsymbol{\sigma}\|^2$ is a statistic across optimization trajectories; we must have some cross-trajectory statistical information to estimate it.

6 Summary

This article explored Adam's Update RMS from both simulation experiments and theoretical approximations. This serves as one of the theoretical foundations for aligning the Update RMS to 0.2 in the Muon optimizer.

Reprinting: Please include the original address of this article: <https://kexue.fm/archives/11267>

For more details on reprinting, please refer to: "Scientific Space FAQ"