Making Alchemy More Scientific (Part 9): Classic Adaptive Gradient Algorithms

Su Jianlin

2026-09-05

The previous eight articles in this series have all revolved around SGD and its learning rate. Starting from this article, we will formally enter the world of adaptive gradient algorithms. It can be said that all current adaptive gradient algorithms share a common origin, which is the 2011 classic work *Adaptive Subgradient Methods for Online Learning and Stochastic Optimization*—namely the famous AdaGrad paper.

This paper extends the classic convergence conclusions of SGD to a general preconditioning matrix form, and then derives that the optimal preconditioning matrix is exactly the second moment of the gradient (to the power of $-1/2$), and then takes the diagonal approximation to obtain AdaGrad. Although AdaGrad itself is not considered highly popular, the subsequent works it inspired, such as RMSProp and Adam, are undeniably mainstream optimizers. Moreover, subsequent optimizers like Shampoo and KL-Shampoo are also considered its inheritances.

Let us revisit this classic work together in this article.

1 Difficulty Analysis

Actually, in previous articles, we briefly discussed the idea of adaptive learning rates, such as in *Making Alchemy More Scientific (Part 5): Fine-tuning Learning Rate Based on Gradients*, but that was still about tuning a scalar learning rate within the SGD framework. This time we consider the general update rule

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta_t \boldsymbol{H}_t^{-1} \boldsymbol{g}(\boldsymbol{x}_t, \boldsymbol{\theta}_t) \tag{1}$$

where $\boldsymbol{H}_t$ is a positive definite symmetric matrix, and $\boldsymbol{H}_t^{-1}$ is usually called the "Preconditioner". We can view it as a matrix version of the learning rate; obviously, when $\boldsymbol{H}_t = \boldsymbol{I}$, it degenerates into SGD.

Why extend it to a matrix learning rate? Just from the perspective of input and output, we hope to input a gradient vector and output an update vector. The most general linear transformation between vectors is matrix multiplication, so a matrix learning rate is the ultimate generalization. As for the positive definiteness constraint, it is to ensure that $\boldsymbol{g}^\top \boldsymbol{H}_t^{-1} \boldsymbol{g} \geq 0$ holds for any $\boldsymbol{g}$, so that $-\boldsymbol{H}_t^{-1}\boldsymbol{g}$ remains a direction that can decrease the loss.

In this section, let us briefly analyze what difficulties we will face for a general $\boldsymbol{H}_t$ if we directly apply the proof process of SGD. The starting point of all previous proofs is the following identity

$$\begin{aligned} \|\boldsymbol{\theta}_{t+1} - \boldsymbol{\varphi}\|^2 &= \|\boldsymbol{\theta}_t - \eta_t \boldsymbol{g}(\boldsymbol{x}_t, \boldsymbol{\theta}_t) - \boldsymbol{\varphi}\|^2 \\ &= \|\boldsymbol{\theta}_t - \boldsymbol{\varphi}\|^2 - 2\eta_t (\boldsymbol{\theta}_t - \boldsymbol{\varphi}) \cdot \boldsymbol{g}(\boldsymbol{x}_t, \boldsymbol{\theta}_t) + \eta_t^2 \|\boldsymbol{g}(\boldsymbol{x}_t, \boldsymbol{\theta}_t)\|^2 \end{aligned} \tag{2}$$

Then we try to separate $(\boldsymbol{\theta}_t - \boldsymbol{\varphi}) \cdot \boldsymbol{g}(\boldsymbol{x}_t, \boldsymbol{\theta}_t)$ to one side, and connect it with the loss value through the convexity assumption:

$$L(\boldsymbol{x}_t, \boldsymbol{\theta}_t) - L(\boldsymbol{x}_t, \boldsymbol{\varphi}) \leq (\boldsymbol{\theta}_t - \boldsymbol{\varphi}) \cdot \boldsymbol{g}(\boldsymbol{x}_t, \boldsymbol{\theta}_t) \tag{3}$$

However, when we consider the general update rule (1), if we still use the same identity, then $(\boldsymbol{\theta}_t - \boldsymbol{\varphi}) \cdot \boldsymbol{H}_t^{-1}\boldsymbol{g}(\boldsymbol{x}_t, \boldsymbol{\theta}_t)$ will appear. Due to the extra $\boldsymbol{H}_t^{-1}$, even if we assume the convexity of the loss function, we cannot connect this term with the loss function—the right end of the convexity inequality (3) can only be the clean $(\boldsymbol{\theta}_t - \boldsymbol{\varphi}) \cdot \boldsymbol{g}(\boldsymbol{x}_t, \boldsymbol{\theta}_t)$; inserting a $\boldsymbol{H}_t^{-1}$ in the middle makes it inapplicable.

2 Generalized Identity

Therefore, we must simultaneously generalize this identity. The generalization idea uses the identity $\boldsymbol{x} \cdot \boldsymbol{y} = \boldsymbol{x}^\top \boldsymbol{y} = \text{tr}(\boldsymbol{y}\boldsymbol{x}^\top)$. Note that $\boldsymbol{y}\boldsymbol{x}^\top$ is the vector outer product, and the result is a matrix. This means we can first establish a matrix version of the identity at the outer product level, and then take the tr on both sides at the appropriate time to restore the scalar form. In the matrix version of the identity, $\boldsymbol{H}_t^{-1}$ can be successfully moved out.

Specifically, we have (for simplicity, $\boldsymbol{g}(\boldsymbol{x}_t, \boldsymbol{\theta}_t)$ will be denoted as $\boldsymbol{g}_t$ hereafter):

$$\begin{aligned}
&(\boldsymbol{\theta}_{t+1} - \boldsymbol{\varphi})(\boldsymbol{\theta}_{t+1} - \boldsymbol{\varphi})^\top \\
=&(\boldsymbol{\theta}_t - \eta_t \boldsymbol{H}_t^{-1}\boldsymbol{g}_t - \boldsymbol{\varphi})(\boldsymbol{\theta}_t - \eta_t \boldsymbol{H}_t^{-1}\boldsymbol{g}_t - \boldsymbol{\varphi})^\top \\
=&(\boldsymbol{\theta}_t - \boldsymbol{\varphi})(\boldsymbol{\theta}_t - \boldsymbol{\varphi})^\top - \eta_t(\boldsymbol{\theta}_t - \boldsymbol{\varphi})\boldsymbol{g}_t^\top \boldsymbol{H}_t^{-1} - \eta_t \boldsymbol{H}_t^{-1}\boldsymbol{g}_t(\boldsymbol{\theta}_t - \boldsymbol{\varphi})^\top + \eta_t^2 \boldsymbol{H}_t^{-1}\boldsymbol{g}_t\boldsymbol{g}_t^\top \boldsymbol{H}_t^{-1}
\end{aligned} \tag{4}$$

Right-multiplying both ends by $\boldsymbol{H}_t$ and rearranging, we get

$$\begin{aligned}
&\eta_t(\boldsymbol{\theta}_t - \boldsymbol{\varphi})\boldsymbol{g}_t^\top + \eta_t \boldsymbol{H}_t^{-1}\boldsymbol{g}_t(\boldsymbol{\theta}_t - \boldsymbol{\varphi})^\top \boldsymbol{H}_t \\
=&(\boldsymbol{\theta}_t - \boldsymbol{\varphi})(\boldsymbol{\theta}_t - \boldsymbol{\varphi})^\top \boldsymbol{H}_t - (\boldsymbol{\theta}_{t+1} - \boldsymbol{\varphi})(\boldsymbol{\theta}_{t+1} - \boldsymbol{\varphi})^\top \boldsymbol{H}_t + \eta_t^2 \boldsymbol{H}_t^{-1}\boldsymbol{g}_t\boldsymbol{g}_t^\top
\end{aligned} \tag{5}$$

Taking the tr on both ends, and then repeatedly using $\text{tr}(\boldsymbol{A}\boldsymbol{B}) = \text{tr}(\boldsymbol{B}\boldsymbol{A})$, we get

$$2\eta_t(\boldsymbol{\theta}_t - \boldsymbol{\varphi}) \cdot \boldsymbol{g}_t = \|\boldsymbol{\theta}_t - \boldsymbol{\varphi}\|_{\boldsymbol{H}_t}^2 - \|\boldsymbol{\theta}_{t+1} - \boldsymbol{\varphi}\|_{\boldsymbol{H}_t}^2 + \eta_t^2 \|\boldsymbol{g}_t\|_{\boldsymbol{H}_t^{-1}}^2 \tag{6}$$

This is the identity we expected, where $\|\boldsymbol{x}\|_{\boldsymbol{H}} \triangleq \sqrt{\boldsymbol{x}^\top \boldsymbol{H}\boldsymbol{x}}$, which we call the "Mahalanobis norm". The Euclidean norm is just $\|\boldsymbol{x}\|_{\boldsymbol{I}}$, which we simply denote as $\|\boldsymbol{x}\|$. Now the left end restores $(\boldsymbol{\theta}_t - \boldsymbol{\varphi}) \cdot \boldsymbol{g}_t$, allowing us to connect it with the convexity inequality (3). The price is that the distance metric on the right end changes from the Euclidean norm to a time-varying Mahalanobis norm.

3 Classic Scaling

The next steps are very similar to those in *Making Alchemy More Scientific (Part 1): Convergence of SGD's Average Loss* and *Making Alchemy More Scientific (Part 2): Generalizing the Conclusion to Unbounded Domains*. On the left end, we similarly use inequality (3) to connect it with the loss, and then sum over $t = 1, 2, \cdots, T$ to get

$$2\sum_{t=1}^T \eta_t[L(\boldsymbol{x}_t, \boldsymbol{\theta}_t) - L(\boldsymbol{x}_t, \boldsymbol{\varphi})] \leq \sum_{t=1}^T (\|\boldsymbol{\theta}_t - \boldsymbol{\varphi}\|_{\boldsymbol{H}_t}^2 - \|\boldsymbol{\theta}_{t+1} - \boldsymbol{\varphi}\|_{\boldsymbol{H}_t}^2) + \sum_{t=1}^T \eta_t^2 \|\boldsymbol{g}_t\|_{\boldsymbol{H}_t^{-1}}^2 \tag{7}$$

However, since $\|\boldsymbol{\theta}_{t+1} - \boldsymbol{\varphi}\|_{\boldsymbol{H}_t}^2$ in the first summation term on the right end is not $\|\boldsymbol{\theta}_{t+1} - \boldsymbol{\varphi}\|_{\boldsymbol{H}_{t+1}}^2$, the summation cannot cancel out. Here we will use a classic technique from *Making Alchemy*

More Scientific (Part 1): Convergence of SGD's Average Loss:

$$\begin{aligned}
&\sum_{t=1}^{T}(\|\boldsymbol{\theta}_t - \boldsymbol{\varphi}\|_{\boldsymbol{H}_t}^2 - \|\boldsymbol{\theta}_{t+1} - \boldsymbol{\varphi}\|_{\boldsymbol{H}_t}^2) \\
=& \|\boldsymbol{\theta}_1 - \boldsymbol{\varphi}\|_{\boldsymbol{H}_1}^2 - \|\boldsymbol{\theta}_{T+1} - \boldsymbol{\varphi}\|_{\boldsymbol{H}_T}^2 + \sum_{t=2}^{T}(\|\boldsymbol{\theta}_t - \boldsymbol{\varphi}\|_{\boldsymbol{H}_t}^2 - \|\boldsymbol{\theta}_t - \boldsymbol{\varphi}\|_{\boldsymbol{H}_{t-1}}^2) \\
=& \|\boldsymbol{\theta}_1 - \boldsymbol{\varphi}\|_{\boldsymbol{H}_1}^2 - \|\boldsymbol{\theta}_{T+1} - \boldsymbol{\varphi}\|_{\boldsymbol{H}_T}^2 + \sum_{t=2}^{T}\|\boldsymbol{\theta}_t - \boldsymbol{\varphi}\|_{\boldsymbol{H}_t - \boldsymbol{H}_{t-1}}^2
\end{aligned} \tag{8}$$

To continue the scaling, we need to add an assumption to $\boldsymbol{H}_t$: for any t, $\boldsymbol{H}_t - \boldsymbol{H}_{t-1}$ is positive semi-definite, denoted simply as $\boldsymbol{H}_t \succeq \boldsymbol{H}_{t-1}$. This is equivalent to the previous non-increasing learning rate assumption. In addition, for any positive semi-definite matrix $\boldsymbol{H}$, $\boldsymbol{H} \preceq \operatorname{tr}(\boldsymbol{H})\boldsymbol{I}$ also holds, so $\|\boldsymbol{x}\|_{\boldsymbol{H}} \leq \sqrt{\operatorname{tr}(\boldsymbol{H})}\|\boldsymbol{x}\|$. This scaling is actually quite loose, but it is sufficient for qualitative conclusions.

Using the above assumptions and scaling, and discarding the non-positive term $-\|\boldsymbol{\theta}_{T+1} - \boldsymbol{\varphi}\|_{\boldsymbol{H}_T}^2$, we have

$$\begin{aligned}
&\sum_{t=1}^{T}(\|\boldsymbol{\theta}_t - \boldsymbol{\varphi}\|_{\boldsymbol{H}_t}^2 - \|\boldsymbol{\theta}_{t+1} - \boldsymbol{\varphi}\|_{\boldsymbol{H}_t}^2) \\
\leq& \|\boldsymbol{\theta}_1 - \boldsymbol{\varphi}\|^2 \operatorname{tr}(\boldsymbol{H}_1) \cancel{- \|\boldsymbol{\theta}_{T+1} - \boldsymbol{\varphi}\|_{\boldsymbol{H}_T}^2} + \sum_{t=2}^{T}\|\boldsymbol{\theta}_t - \boldsymbol{\varphi}\|^2 \operatorname{tr}(\boldsymbol{H}_t - \boldsymbol{H}_{t-1}) \\
\leq& R^2 \operatorname{tr}(\boldsymbol{H}_T)
\end{aligned} \tag{9}$$

Here $R = \max(\|\boldsymbol{\theta}_1 - \boldsymbol{\varphi}\|, \cdots, \|\boldsymbol{\theta}_T - \boldsymbol{\varphi}\|)$, i.e., the common upper bound of all $\|\boldsymbol{\theta}_t - \boldsymbol{\varphi}\|$. If you have doubts about this step, we can simply assume, like in *Making Alchemy More Scientific (Part 1): Convergence of SGD's Average Loss*, that the optimization is performed within a bounded domain, so all $\|\boldsymbol{\theta}_t - \boldsymbol{\varphi}\|$ naturally have a common upper bound R.

4 Initial Results

Substituting the scaling from the previous section back into equation (7), and setting $\boldsymbol{\varphi}$ to the theoretical optimum $\boldsymbol{\theta}^*$, we obtain

$$2\sum_{t=1}^{T}\eta_t[L(\boldsymbol{x}_t, \boldsymbol{\theta}_t) - L(\boldsymbol{x}_t, \boldsymbol{\theta}^*)] \leq R^2 \operatorname{tr}(\boldsymbol{H}_T) + \sum_{t=1}^{T}\eta_t^2\|\boldsymbol{g}_t\|_{\boldsymbol{H}_t^{-1}}^2 \tag{10}$$

For simplicity, let's first consider the case where η_t is a constant. Rearranging, we get

$$\frac{1}{T}\sum_{t=1}^{T}L(\boldsymbol{x}_t, \boldsymbol{\theta}_t) - L(\boldsymbol{x}_t, \boldsymbol{\theta}^*) \leq \frac{R^2}{2T\eta}\operatorname{tr}(\boldsymbol{H}_T) + \frac{\eta}{2T}\sum_{t=1}^{T}\|\boldsymbol{g}_t\|_{\boldsymbol{H}_t^{-1}}^2 \tag{11}$$

This is the general convergence conclusion for $\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta\boldsymbol{H}_t^{-1}\boldsymbol{g}_t$. Now let's think in reverse: for a good optimizer, the right end of the above equation should be as small as possible. Therefore, minimizing the upper bound might yield a better optimizer, which aligns with the idea in *Making Alchemy More Scientific (Part 5): Fine-tuning Learning Rate Based on Gradients*. To this end, we first use the assumption $\boldsymbol{H}_1 \preceq \boldsymbol{H}_2 \preceq \cdots \preceq \boldsymbol{H}_T$ to get $\|\boldsymbol{g}_t\|_{\boldsymbol{H}_t^{-1}}^2 \geq \|\boldsymbol{g}_t\|_{\boldsymbol{H}_T^{-1}}^2$, and thus

$$\frac{R^2}{2T\eta}\operatorname{tr}(\boldsymbol{H}_T) + \frac{\eta}{2T}\sum_{t=1}^{T}\|\boldsymbol{g}_t\|_{\boldsymbol{H}_t^{-1}}^2 \geq \frac{R^2}{2T\eta}\operatorname{tr}(\boldsymbol{H}_T) + \frac{\eta}{2T}\sum_{t=1}^{T}\|\boldsymbol{g}_t\|_{\boldsymbol{H}_T^{-1}}^2 \tag{12}$$

That is to say, taking the same $\boldsymbol{H}$ for all $\boldsymbol{H}_t$ can achieve a smaller upper bound. Next, we rewrite the right end of the above equation as

$$\frac{R^2}{2T\eta}\operatorname{tr}(\boldsymbol{H}) + \frac{\eta}{2T}\sum_{t=1}^{T}\|\boldsymbol{g}_t\|_{\boldsymbol{H}^{-1}}^2 = \frac{R^2}{2T\eta}\operatorname{tr}(\boldsymbol{H}) + \frac{\eta}{2T}\operatorname{tr}(\boldsymbol{\Sigma}\boldsymbol{H}^{-1}) \tag{13}$$

where $\boldsymbol{\Sigma} = \sum_{t=1}^{T}\boldsymbol{g}_t\boldsymbol{g}_t^\top$ is the (unnormalized) second moment matrix. It is not difficult to see that η is actually a redundant parameter; the true independent variable is $\boldsymbol{H}/\eta$, which tells us that η can be any positive number. For simplicity, we take $\eta = R$, which makes $\frac{R^2}{2T\eta} = \frac{\eta}{2T}$, so minimizing the upper bound is equivalent to

$$\min_{\boldsymbol{H}\succeq\boldsymbol{0}} \operatorname{tr}(\boldsymbol{H}) + \operatorname{tr}(\boldsymbol{\Sigma}\boldsymbol{H}^{-1}) \tag{14}$$

Interestingly, this is essentially the optimization objective of PSGD!

5 Objective Solution

To solve the above objective, we again use the cyclic property of the trace to rewrite the two terms respectively as

$$\begin{aligned}\operatorname{tr}(\boldsymbol{H}) &= \operatorname{tr}(\boldsymbol{H}^{1/2}\boldsymbol{H}^{1/2}) = \|\boldsymbol{H}^{1/2}\|_F^2 \\ \operatorname{tr}(\boldsymbol{\Sigma}\boldsymbol{H}^{-1}) &= \operatorname{tr}(\boldsymbol{\Sigma}^{1/2}\boldsymbol{H}^{-1/2}\boldsymbol{H}^{-1/2}\boldsymbol{\Sigma}^{1/2}) = \|\boldsymbol{\Sigma}^{1/2}\boldsymbol{H}^{-1/2}\|_F^2\end{aligned} \tag{15}$$

Thus, using the mean inequality, we get

$$\operatorname{tr}(\boldsymbol{H}) + \operatorname{tr}(\boldsymbol{\Sigma}\boldsymbol{H}^{-1}) = \|\boldsymbol{H}^{1/2}\|_F^2 + \|\boldsymbol{\Sigma}^{1/2}\boldsymbol{H}^{-1/2}\|_F^2 \geq 2\,\|\boldsymbol{H}^{1/2}\|_F\,\|\boldsymbol{\Sigma}^{1/2}\boldsymbol{H}^{-1/2}\|_F \tag{16}$$

Then, using the Cauchy-Schwarz inequality, we get

$$\|\boldsymbol{H}^{1/2}\|_F\,\|\boldsymbol{\Sigma}^{1/2}\boldsymbol{H}^{-1/2}\|_F \geq \operatorname{tr}(\boldsymbol{H}^{1/2}\boldsymbol{\Sigma}^{1/2}\boldsymbol{H}^{-1/2}) = \operatorname{tr}(\boldsymbol{\Sigma}^{1/2}) \tag{17}$$

The condition for the equality in the Cauchy-Schwarz inequality to hold is that there exists $\lambda \geq 0$ such that $\boldsymbol{H}^{1/2} = \lambda\boldsymbol{\Sigma}^{1/2}\boldsymbol{H}^{-1/2}$, i.e., $\boldsymbol{H} = \lambda\boldsymbol{\Sigma}^{1/2}$. The condition for the mean inequality to hold is $\|\boldsymbol{H}^{1/2}\|_F = \|\boldsymbol{\Sigma}^{1/2}\boldsymbol{H}^{-1/2}\|_F$. Substituting this further determines $\lambda = 1$, so the optimal solution is

$$\boldsymbol{H}^* = \boldsymbol{\Sigma}^{1/2} = \left(\sum_{t=1}^{T}\boldsymbol{g}_t\boldsymbol{g}_t^\top\right)^{1/2} \tag{18}$$

This is the most classic result of the AdaGrad paper, and it can be regarded as the "primogenitor" of adaptive learning rate optimizers based on the second moment. From this, we can see that adjusting the learning rate based on the second moment is not an ad-hoc heuristic trick, but a natural product of the principle of "minimizing the convergence upper bound". When $\boldsymbol{\Sigma}$ is not invertible, we can add a $\kappa\boldsymbol{I}$ to it. This is an implementation detail and will not be expanded upon in the theoretical derivation.

Intuitively, $\boldsymbol{H}^{-1}\boldsymbol{g}_t = \boldsymbol{\Sigma}^{-1/2}\boldsymbol{g}_t$ is simply a "whitening" operation on the gradient, making the updates more isotropic. This is exactly the same idea as the current Muon optimizer.

6 Various Variants

However, although $\boldsymbol{H}^* = \boldsymbol{\Sigma}^{1/2}$ is theoretically optimal, it violates the law of causality because it assumes we foresee the gradients at all time steps from the very beginning. We should be familiar with this episode; we already encountered it in *Making Alchemy More Scientific (Part*

5): Fine-tuning Learning Rate Based on Gradients. The solution is to truncate the summation to the current time step:

$$\boldsymbol{\Sigma}_t = \sum_{\tau=1}^{t} \boldsymbol{g}_\tau \boldsymbol{g}_\tau^\top, \qquad \boldsymbol{H}_t = \boldsymbol{\Sigma}_t^{1/2} \tag{19}$$

It is worth pointing out that $\boldsymbol{H}_t$ defined this way still satisfies $\boldsymbol{H}_t \succeq \boldsymbol{H}_{t-1}$, so the convergence conclusion up to (11) still holds. However, retaining the full matrix $\boldsymbol{H}_t$ is extremely expensive in terms of both storage and computational costs, which is impractical. A naive solution is to consider a diagonal approximation, i.e., only taking the diagonal part of $\boldsymbol{\Sigma}_t$ and then taking the square root:

$$\boldsymbol{H}_t \approx \operatorname{diag}(\boldsymbol{\Sigma}_t)^{1/2} \qquad \Rightarrow \qquad \boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta \frac{\boldsymbol{g}_t}{\sqrt{\sum_{\tau=1}^t \boldsymbol{g}_\tau \odot \boldsymbol{g}_\tau}} \tag{20}$$

This is what we call the AdaGrad optimizer, where $\odot$ represents the Hadamard product, and the division of two vectors is also component-wise division in the Hadamard sense. Note that the learning rate of each component in AdaGrad is monotonically decreasing, which is usually considered a bad property in actual training (it is easy to "stop learning" in the later stages of training). Therefore, RMSProp replaces the summation with an Exponential Moving Average (EMA):

$$\boldsymbol{v}_t = \beta \boldsymbol{v}_{t-1} + (1-\beta)\boldsymbol{g}_t \odot \boldsymbol{g}_t, \qquad \boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta \frac{\boldsymbol{g}_t}{\sqrt{\boldsymbol{v}_t}} \tag{21}$$

Note that after switching to EMA, the practical performance is often better, but it can no longer guarantee $\boldsymbol{H}_t \succeq \boldsymbol{H}_{t-1}$. This is the beginning of the "divergence" between theory and practice. If we add momentum, it becomes the classic of classics—the Adam optimizer (for simplicity, Bias Correction is omitted):

$$\begin{aligned} \boldsymbol{m}_t &= \beta_1 \boldsymbol{m}_{t-1} + (1-\beta_1)\boldsymbol{g}_t \\ \boldsymbol{v}_t &= \beta_2 \boldsymbol{v}_{t-1} + (1-\beta_2)\boldsymbol{g}_t \odot \boldsymbol{g}_t \end{aligned}, \qquad \boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta \frac{\boldsymbol{m}_t}{\sqrt{\boldsymbol{v}_t}} \tag{22}$$

Before Muon, Adam had always been the most mainstream optimizer, bar none. Even with Muon, Adam remains an important part of mainstream optimizers (for Embeddings, LM Heads, etc.). As for RMSProp, it is usually used for training generative models, especially GAN models, because for such models, adding momentum can easily cause Mode Collapse.

In addition to these classic optimizers, new optimizers like PSGD, Shampoo, KL-Shampoo, and AdaBK are essentially just another structured approximation of $\boldsymbol{\Sigma}$ or $\boldsymbol{\Sigma}^{-1/2}$ for matrix multiplication parameters. In other words, even today, the preconditioning matrix $\boldsymbol{\Sigma}^{-1/2}$ derived from AdaGrad remains one of the important guiding principles for optimizers.

7 Article Summary

This article reviewed the foundational work of adaptive gradient algorithms—AdaGrad—and reproduced its complete derivation along the route of "convergence analysis → minimizing the upper bound → optimal preconditioning matrix". It can be said that "adjusting the learning rate based on the second moment of the gradient" proposed by AdaGrad is almost the common underlying principle of all subsequent adaptive learning rate optimizers.