# Making Alchemy More Scientific (Part 8): Learning Rates for Multi-Stage Training

Su Jianlin

2026-08-31

In the previous article *Making Alchemy More Scientific (Part 7): Step Size Scheduling and Weight Averaging*, we briefly introduced the work on Schedule-Free learning rates. It attempts to replace learning rate scheduling with some form of weight averaging, achieving the effect of training an optimal model with a constant learning rate. However, as mentioned in the previous article, without introducing additional assumptions, this optimal constant learning rate also depends on the number of training steps, so true schedule-free training cannot be achieved.

In this article, we rethink this problem from the perspective of multi-stage training. The main idea is to compromise the scheduling objective to "being close to optimal at the end of each stage," making it more simplified and feasible in practice.

## 1 Classical Conclusion

We first start from a classical convergence conclusion to understand why the optimal learning rate depends on the number of training steps. We first introduced this convergence conclusion in *Making Alchemy More Scientific (Part 2): Generalizing Conclusions to Unbounded Domains*:

$$\frac{\sum_{t=1}^T \eta_t \mathbb{E}[L(\boldsymbol{\theta}_t) - L(\boldsymbol{\theta}^*)]}{\sum_{t=1}^T \eta_t} \leq \frac{R^2}{2\sum_{t=1}^T \eta_t} + \frac{G^2}{2}\frac{\sum_{t=1}^T \eta_t^2}{\sum_{t=1}^T \eta_t} \triangleq f_T(\eta) \tag{1}$$

We will not repeat the specific meanings of the notations. This convergence conclusion is representative because the subsequent *Making Alchemy More Scientific (Part 6): Top-Down Ingenious Construction* and the previous article *Making Alchemy More Scientific (Part 7): Step Size Scheduling and Weight Averaging* are both based on it.

Now what we need to do is, given $R, G, T > 0$, find $\eta_1, \cdots, \eta_T$ such that $f_T(\eta)$ is minimized, which can be solved through a simple inequality. First, it is easy to prove that $\sum_{t=1}^T \eta_t^2 \geq (\sum_{t=1}^T \eta_t)^2/T$, thus

$$f_T(\eta) \geq \frac{R^2}{2\sum_{t=1}^T \eta_t} + \frac{G^2}{2T}\sum_{t=1}^T \eta_t \geq \frac{RG}{\sqrt{T}} \triangleq f_T^* \tag{2}$$

The condition for the equality in both parts to hold simultaneously is $\eta_1 = \cdots = \eta_T = \frac{R}{G\sqrt{T}}$, which depends on $T$. If we switch to a learning rate schedule that is truly independent of $T$, such as $\eta_t \propto 1/\sqrt{t}$, then at most we can achieve $f_T(\eta) = \mathcal{O}(\log T/\sqrt{T})$, and it can also be proved that no matter how it is improved, a learning rate independent of $T$ cannot achieve a better $f$ than $\Theta(\sqrt{\log T/T})$.

## 2 Changing the Perspective

Now let us return to think about the motivation of Schedule-Free. Ideally, Schedule-Free hopes to continue training through some method that does not require knowing the total number of training steps $T$ in advance, so that no matter at which step we stop training, we can obtain nearly optimal model weights up to the current step.

However, this perfect goal seems almost impossible. Because even under the convex optimization assumption of Schedule-Free, we can equate LR Decay with weight averaging to achieve constant learning rate training, but as analyzed in the previous section, the optimal value of this constant learning rate also depends on $T$. It cannot achieve obtaining the optimal solution up to the current step at any time, unless we further introduce additional assumptions.

Since this is the case, we might as well reflect on what we really want from Schedule-Free. First, a constant learning rate is not a rigid requirement; if a certain time-varying learning rate can train a good result, then we can also accept it. Secondly, "optimal at any stopping time" is of course a good property, but in fact, it is not a strong demand. We mainly hope that during multi-stage training, the endpoint of each stage can approach the optimum as closely as possible.

In other words, when our training is divided into multiple stages, we hope that at the end of each stage, we can get a model as close to optimal as possible. The number of stages will not be too large, so for simplicity, let us first consider two-stage training: the first stage trains for $T_1$ steps, and the second stage trains for $T_2 - T_1$ steps, totaling $T_2$ steps of training.

Here, it can be further divided into two situations: "unexpected" and "planned". "Unexpected" means there was originally no plan to continue training, so the first stage will be tuned to be as optimal as possible. After deciding to continue training, the second stage then tries to tune to the optimum based on the existing foundation. This yields a greedy solution, with the drawback that no matter how the second stage is tuned, the result is relatively suboptimal. As for the "planned" scenario, the multi-stage training is planned in advance, and theoretically, we can better balance the effects of the two stages.

## 3 Greedy Decision

Let us first look at the greedy version. The first stage takes the optimum, which is naturally a constant learning rate $\eta^*_{(1)} = \frac{R}{G\sqrt{T_1}}$. At this time, the $f$ of the first stage reaches the theoretical optimum $f^*_{T_1}$. In the second stage, we need to find $\eta_{T_1+1}, \cdots, \eta_{T_2}$ on this basis to minimize

$$f_{T_2}(\eta) = \frac{R^2 + G^2(T_1(\eta^*_{(1)})^2 + \sum_{t=T_1+1}^{T_2}\eta_t^2)}{2(T_1\eta^*_{(1)} + \sum_{t=T_1+1}^{T_2}\eta_t)} \tag{3}$$

Based on the same inequality $\sum_{t=T_1+1}^{T_2}\eta_t^2 \geq (\sum_{t=T_1+1}^{T_2}\eta_t)^2/(T_2 - T_1)$, we know that the minimum is still achieved at a constant learning rate, so it simplifies to

$$f_{T_2}(\eta) = \frac{R^2 + G^2(T_1(\eta^*_{(1)})^2 + (T_2 - T_1)(\eta_{(2)})^2)}{2(T_1\eta^*_{(1)} + (T_2 - T_1)\eta_{(2)})} \tag{4}$$

This can be done by matching inequalities or directly taking derivatives. The minimum point is

$$\eta^*_{(2)} = \frac{R/G}{\sqrt{T_2^\#}}, \qquad f(\eta^*_{(2)}) = \frac{RG}{\sqrt{T_2^\#}} \tag{5}$$

where

$$T_2^\# = \frac{1}{2}T_2 + \frac{1}{2}\sqrt{T_1(2T_2 - T_1)} \tag{6}$$

## 4 Equivalent Steps

Here we introduce the notation $T_2^\#$, its meaning is "equivalent steps".

If there were no restriction in the first stage, we could have used a constant learning rate $\frac{R}{G\sqrt{T_2}}$ from the very beginning, and the minimum at the end of training would reach $\frac{RG}{\sqrt{T_2}}$. But now we can only achieve $\frac{RG}{\sqrt{T_2^\#}}$, and the learning rate for the second stage is also in the form of $\frac{R}{G\sqrt{T_2^\#}}$. This is equivalent to saying that the effective number of steps has changed from $T_2$ to $T_2^\#$. It is not difficult to verify that

$$T_2^\# \leq T_2 \tag{7}$$

That is to say, two-stage greedy training is equivalent to losing some steps in terms of effect. Taking $T_2 = 2T_1$ as an example, $T_2^\# \approx 1.866T_1$, compared to $2T_1$, it roughly loses 6.7% of the steps. Note that if $T_2 \to \infty$, then $T_2^\#/T_2 \to 1/2$, which means that if training continues indefinitely, the proportion lost will increase, up to a maximum of half.

Why abstract this concept? Because the concept of "steps" is universal, it makes the conclusions easier to transfer. The above conclusion is derived based on SGD, but

actual training usually uses non-SGD optimizers such as Adam and Muon. The setting of hyperparameters such as learning rate does not directly use the above formulas, but is usually determined based on a searched Scaling Law.

Therefore, the various conclusions given by SGD cannot be used directly. But if we can extract the relative amount of change through the concept of "equivalent steps", then the step input in the Scaling Law used in practice can also be similarly replaced with equivalent steps, thereby obtaining a heuristic correction result. This is the main bridge from the theoretical conclusions in this article to practice.

## 5 Planned Scenario

Next, let us look at the "planned" scenario. At this time, $T_1, T_2$ are both known, and the learning rates of the two stages can be jointly tuned. The core problem here is that we have two delivery targets—the model at the end of the first stage and the final model—and we need to find a way to balance the effects of these two stages. To this end, we introduce the minimax optimization objective:

$$\min_{\eta\geq 0} \max_{T\in\{T_1,T_2\}} \frac{f_T(\eta)}{f_T^*} \tag{8}$$

It hopes that the ratio of the actual effect of each stage to its ideal value is as small as possible. The advantage of this optimization objective is that it does not introduce additional hyperparameters and is easy to generalize to multi-stage training. Introducing a new variable $m$, we can transform it into a nonlinear programming problem

$$\min\left\{m \;\middle|\; f_{T_1}(\eta)/f_{T_1}^* \leq m,\ f_{T_2}(\eta)/f_{T_2}^* \leq m,\ m\geq 1,\ \eta\geq 0\right\} \tag{9}$$

Dimensionalize the problem: let $x_t = G\eta_t/R$, then for any $T$ we can complete the square to get

$$\frac{f_T(\eta)}{f_T^*} = \frac{\sqrt{T}(1+\sum_{t=1}^T x_t^2)}{2\sum_{t=1}^T x_t} \leq m \iff \sum_{t=1}^T \left(x_t - \frac{m}{\sqrt{T}}\right)^2 \leq m^2 - 1 \tag{10}$$

That is, the feasible set of $x_t$ forms a hyperball centered at $\frac{m}{\sqrt{T}}\mathbf{1}$ with a radius of $\sqrt{m^2-1}$. It can be seen that the original problem is equivalent to a quadratic programming problem with $m$ as the objective.

## 6 Solution Process

For the two-stage scenario, this problem can be solved analytically. Note that the first constraint only involves $x_1, \cdots, x_{T_1}$, and the second constraint involves all $T_2$ variables. For $x_t$ where $t > T_1$, they only appear in the second constraint, so we can directly set

$x_t = m/\sqrt{T_2}$ to make its contribution zero. Thus the problem is reduced to: the first $T_1$ $x_t$ must simultaneously fall within two balls

$$\sum_{t=1}^{T_1}\left(x_t - \frac{m}{\sqrt{T_1}}\right)^2 \leq m^2 - 1, \qquad \sum_{t=1}^{T_1}\left(x_t - \frac{m}{\sqrt{T_2}}\right)^2 \leq m^2 - 1 \tag{11}$$

The radii of both balls are $\sqrt{m^2-1}$, and the centers are both in the direction of the all-ones vector. The squared distance between the centers is

$$T_1\left(\frac{m}{\sqrt{T_1}} - \frac{m}{\sqrt{T_2}}\right)^2 = m^2(1-\sqrt{\tau})^2, \qquad \tau = T_1/T_2 \tag{12}$$

The two balls have an intersection if and only if the distance between the centers does not exceed twice the radius, i.e., $m^2(1-\sqrt{\tau})^2 \leq 4(m^2-1)$. From this, the optimal $m$ is solved as:

$$m^* = \frac{2}{\sqrt{4-(1-\sqrt{\tau})^2}} \tag{13}$$

The corresponding solution can be taken as the midpoint of the line connecting the centers of the two balls, i.e.,

$$x_t = \begin{cases} \dfrac{m^*}{2}\left(\dfrac{1}{\sqrt{T_1}} + \dfrac{1}{\sqrt{T_2}}\right), & t \leq T_1 \\ \dfrac{m^*}{\sqrt{T_2}}, & t > T_1 \end{cases} \tag{14}$$

## 7 Result Analysis

Restoring $\eta_t = Rx_t/G$, we get

$$\eta^*_{(1)} = \frac{m^*R}{2G}\left(\frac{1}{\sqrt{T_1}} + \frac{1}{\sqrt{T_2}}\right), \qquad \eta^*_{(2)} = \frac{m^*R}{G\sqrt{T_2}} \tag{15}$$

Interestingly, $\eta^*_{(1)}$ is exactly the average (multiplied by $m^*$) of the two single-stage optimal constant learning rates for "only training for $T_1$ steps" and "training from scratch for $T_2$ steps". From the perspective of equivalent steps, $\eta^*_{(1)}$ is equivalent to presetting a larger number of training steps, while $\eta^*_{(2)}$ is equivalent to presetting a smaller number of training steps

$$T_1^{\#} = \left(\frac{R}{G\eta^*_{(1)}}\right)^2 = T_1\left(\frac{3-\sqrt{\tau}}{1+\sqrt{\tau}}\right), \qquad T_2^{\#} = \left(\frac{R}{G\eta^*_{(2)}}\right)^2 = T_2\left(1 - \frac{(1-\sqrt{\tau})^2}{4}\right) \tag{16}$$

That is to say, the first stage only needs to train for $T_1$ steps, but when setting the learning rate, it is assumed that it will train for $T_1^{\#}$ steps; the two stages train for a total of $T_2$ steps,

but when setting the learning rate for the second stage, it is assumed that it will train for $T_2^{\#}$ steps. As for the effect, at this time $f_{T_1}/f_{T_1}^* = f_{T_2}/f_{T_2}^* = m^*$, that is, the relative loss of the two stages is completely equal—this is exactly the equilibrium characteristic of the minimax objective: it does not favor either delivery point.

Still taking $T_2 = 2T_1$ as an example, at this time $\tau = 1/2$. Substituting this, we get $T_2^{\#} \approx 0.979T_2$, so in terms of effect, both stages are equivalent to a loss of about 2.1% of the steps compared to the optimal solution. From the perspective of the average effect of the two stages, it is better than the greedy solution. In terms of learning rate setting, the learning rate of the first stage should be set according to the equivalent steps of $T_1^{\#} \approx 1.343T_1$, and the learning rate of the second stage should be set according to the equivalent steps of $T_2^{\#} \approx 0.979T_2$.

Another limit is when $\tau \to 0$, $T_2^{\#} \to 0.75T_2$, which means that when the number of steps in the $T_2$ stage is large enough, the minimax solution loses at most 25% of the steps at the endpoint, which is somewhat better than the 50% of the greedy solution.

## 8 Multi-Stage Generalization

Whether it is the greedy solution for the "unexpected" scenario or the minimax solution for the "planned" scenario, theoretically, both can easily be generalized to multi-stage training. The greedy solution goes without saying; we will mainly expand on the minimax solution. Suppose there are $K$ delivery points $T_1 < T_2 < \cdots < T_K$, the minimax objective becomes

$$\min_{\eta\geq 0} \max_{T\in\{T_1,\cdots,T_K\}} \frac{f_T(\eta)}{f_T^*} \tag{17}$$

The same dimensionalization gives $K$ ball constraints

$$\sum_{t=1}^{T_k} \left(x_t - \frac{m}{\sqrt{T_k}}\right)^2 \leq m^2 - 1, \qquad k = 1, 2, \cdots, K \tag{18}$$

Although there are theoretically $T_K$ variables, it can still be proved that the optimal solution can always be taken in the form of "piecewise constant", so the problem is reduced to a small-scale programming problem with only $K + 1$ variables. When $K \geq 3$, there is generally no concise closed-form solution, but numerical solving is not difficult.

## 9 Article Summary

This article re-examines "schedule-free learning rates" from the perspective of multi-stage training. By changing the scheduling objective to "being close to optimal at the end of each stage", it makes practice more simplified and feasible. Furthermore, we extract the concept

of “equivalent steps”, making it possible to transfer the conclusions to the optimizers and Scaling Laws used in practice.