Why does DeltaNet need L2 Normalization?

Jianlin Su

December 23, 2025

In the article "A Brief History of Linear Attention: From Imitation and Innovation to Feedback", we introduced DeltaNet, which brought the Delta Rule into linear attention, becoming one of its powerful tools and forming the basis for subsequent works such as GDN and KDA. However, that article focused primarily on the overall idea of DeltaNet and did not delve into many technical details. In this post, we will discuss one of them: why do DeltaNet and its subsequent works apply L2 Normalization to $\boldsymbol{Q}$ and $\boldsymbol{K}$?

Of course, it is not difficult to explain this operation directly from the perspective of eigenvalues, but I always felt it was missing something. A few days ago, I learned a new interpretation from the paper "Error-Free Linear Attention is a Free Lunch: Exact Solution from Continuous-Time Dynamics", which I find quite valuable and would like to share.

1 Basic Analysis

The recursive format of DeltaNet is:

$$\boldsymbol{S}_t = \boldsymbol{S}_{t-1} - \eta_t(\boldsymbol{S}_{t-1}\boldsymbol{k}_t - \boldsymbol{v}_t)\boldsymbol{k}_t^\top = \boldsymbol{S}_{t-1}(\boldsymbol{I} - \eta_t\boldsymbol{k}_t\boldsymbol{k}_t^\top) + \eta_t\boldsymbol{v}_t\boldsymbol{k}_t^\top \tag{1}$$

From the perspective of TTT (Test-Time Training), this is using the SGD optimizer with a learning rate η_t to perform online optimization on the loss $\frac{1}{2}\|\boldsymbol{S}\boldsymbol{k} - \boldsymbol{v}\|^2$ (where the trainable parameter is $\boldsymbol{S}$). We know that optimizers are often sensitive to the learning rate, especially non-adaptive optimizers like SGD. In DeltaNet, this manifests as additional requirements for the transition matrix $\boldsymbol{I} - \eta_t\boldsymbol{k}_t\boldsymbol{k}_t^\top$.

Specifically, since transition matrices at different time steps are multiplied together during the recursion, to avoid numerical explosion, the transition matrix cannot have eigenvalues with a magnitude greater than 1. For the matrix $\boldsymbol{I} - \eta_t\boldsymbol{k}_t\boldsymbol{k}_t^\top$, one of its eigenvalues is $1 - \eta_t\|\boldsymbol{k}_t\|^2$, and the rest are all 1 (please prove this). From this, we obtain the constraint:

$$-1 \leq 1 - \eta_t\|\boldsymbol{k}_t\|^2 \leq 1 \tag{2}$$

To satisfy this constraint, a common practice is to apply L2 Normalization to $\boldsymbol{k}_t$ and a Sigmoid function to η_t, so that all eigenvalues fall within $(0, 1]$. This is the origin of L2 Normalization for $\boldsymbol{K}$. As for the L2 Normalization of $\boldsymbol{Q}$, it is not strictly necessary and is added more for the sake of symmetry, similar to the case of Short Conv, where applying Short Conv to $\boldsymbol{K}$ is the most critical part [Reference].

2 Supplementary Notes

As a side note, for a long time, people were accustomed to keeping eigenvalues within $(0, 1]$ and thus chose to apply Sigmoid to η_t. Later, "Unlocking State-Tracking in Linear RNNs Through Negative Eigenvalues" pointed out that negative eigenvalues can enhance the state-tracking capability of DeltaNet. They proposed modifying DeltaNet to:

$$\boldsymbol{S}_t = \boldsymbol{S}_{t-1}(\boldsymbol{I} - 2\eta_t\boldsymbol{k}_t\boldsymbol{k}_t^\top) + \eta_t\boldsymbol{v}_t\boldsymbol{k}_t^\top \tag{3}$$

Then, by still applying L2 Normalization to $\boldsymbol{k}_t$ and Sigmoid to η_t, the range of eigenvalues for the transition matrix $\boldsymbol{I}-2\eta_t\boldsymbol{k}_t\boldsymbol{k}_t^\top$ is expanded to $(-1,1]$. However, state-tracking is a capability biased towards specific syntax (such as code). Therefore, if we only train and test on natural language after this modification, we might not observe significant changes.

Another detail to note is that when $\eta_t=1$, the transition matrix $\boldsymbol{I}-2\boldsymbol{k}_t\boldsymbol{k}_t^\top$ is an orthogonal matrix. Theoretically, this is fine, but in practice, it fails. Because for efficiency, we usually use at least BF16 precision in implementation. BF16 has lower precision, which makes it possible for the eigenvalues of $\boldsymbol{I}-2\boldsymbol{k}_t\boldsymbol{k}_t^\top$ to fall below -1. Under long-term cumulative multiplication, there is still a risk of explosion. Therefore, it is necessary to ensure η_t does not get too close to 1.

In fact, the above explanation is already quite complete and not complex. My critique of it stems mainly from personal aesthetic preference: the method of implementing condition (2) is not unique. For example, one could introduce a Squash operation similar to Capsule as in Longhorn. Thus, we cannot naturally derive L2 Normalization; we can only say it is one viable solution.

3 Continuous Perspective

Next, we introduce the approach from the paper "Error-Free Linear Attention is a Free Lunch: Exact Solution from Continuous-Time Dynamics". I believe this is also an elegant derivation path, though this depends on one's aesthetic. It views Equation (1) as the Euler discretization of the following differential equation over the interval $[t-\eta_t, t]$:

$$\frac{d}{dt}\boldsymbol{S}_t = \boldsymbol{S}_t\underbrace{(-\boldsymbol{k}_t\boldsymbol{k}_t^\top)}_{\boldsymbol{A}_t} + \underbrace{\boldsymbol{v}_t\boldsymbol{k}_t^\top}_{\boldsymbol{B}_t} \tag{4}$$

The paper points out that numerical explosion occurs because the precision of the discretization format is not high enough. Therefore, it proposes constructing the recursion by directly solving the differential equation rather than using approximate discretization. Since $\boldsymbol{A}_t$ and $\boldsymbol{B}_t$ are constants within the interval $[t-\eta_t, t]$, solving the recursion from $t-\eta_t$ to t is equivalent to solving a linear differential equation with constant coefficients. The general result is:

$$\boldsymbol{S}_t = \boldsymbol{S}_{t-\eta_t}e^{\eta_t\boldsymbol{A}_t} + \boldsymbol{B}_t\boldsymbol{A}_t^{-1}(e^{\eta_t\boldsymbol{A}_t}-\boldsymbol{I}) \tag{5}$$

Replacing the notation $\boldsymbol{S}_{t-\eta_t}$ back with $\boldsymbol{S}_{t-1}$ and substituting the expressions for $\boldsymbol{A}_t$ and $\boldsymbol{B}_t$, we simplify to get:

$$\boldsymbol{S}_t = \boldsymbol{S}_{t-1}\left(\boldsymbol{I} - \frac{1-e^{-\eta_t\|\boldsymbol{k}_t\|^2}}{\|\boldsymbol{k}_t\|^2}\boldsymbol{k}_t\boldsymbol{k}_t^\top\right) + \frac{1-e^{-\eta_t\|\boldsymbol{k}_t\|^2}}{\|\boldsymbol{k}_t\|^2}\boldsymbol{v}_t\boldsymbol{k}_t^\top \tag{6}$$

This is the final result we want to derive. The original paper calls this "EFLA (Error-Free Linear Attention)". It is equivalent to replacing η_t with $\frac{1-e^{-\eta_t\|\boldsymbol{k}_t\|^2}}{\|\boldsymbol{k}_t\|^2}$. Here, $\|\boldsymbol{k}_t\|^2$ naturally appears in the denominator, and when multiplied by $\boldsymbol{k}_t\boldsymbol{k}_t^\top$, it manifests exactly as L2 Normalization on $\boldsymbol{K}$.

4 Mathematical Details

In the previous section, we quickly introduced the results of EFLA, omitting many mathematical details. In this section, we supplement some discussions. Due to space limitations, we can only briefly mention the key points of the derivation.

The core result of the previous section is Equation (5), which is the solution to the differential equation $d\boldsymbol{S}_t/dt = \boldsymbol{S}_t\boldsymbol{A}+\boldsymbol{B}$. To avoid confusion, we omit the subscripts for $\boldsymbol{A}$ and $\boldsymbol{B}$ here,

as they are indeed constants within the solution interval. If $\boldsymbol{B} = \boldsymbol{0}$, we can directly write $\boldsymbol{S}_t = \boldsymbol{S}_0 e^{t\boldsymbol{A}}$, where $e^{t\boldsymbol{A}}$ is the matrix exponential. When $\boldsymbol{B} \neq \boldsymbol{0}$, we rewrite the equation as $d(\boldsymbol{S}_t + \boldsymbol{B}\boldsymbol{A}^{-1})/dt = (\boldsymbol{S}_t + \boldsymbol{B}\boldsymbol{A}^{-1})\boldsymbol{A}$. Using the solution for the $\boldsymbol{B} = \boldsymbol{0}$ case, we get:

$$\boldsymbol{S}_t = (\boldsymbol{S}_0 + \boldsymbol{B}\boldsymbol{A}^{-1})e^{t\boldsymbol{A}} - \boldsymbol{B}\boldsymbol{A}^{-1} = \boldsymbol{S}_0 e^{t\boldsymbol{A}} + \boldsymbol{B}\boldsymbol{A}^{-1}(e^{t\boldsymbol{A}} - \boldsymbol{I}) \tag{7}$$

Finally, by changing the starting point to $t - \eta_t$, the end point to t, and restoring the subscripts t for $\boldsymbol{A}$ and $\boldsymbol{B}$, we obtain Equation (5). Note that the last term involves the inverse matrix $\boldsymbol{A}^{-1}$, but in practice, $\boldsymbol{A}$ does not need to be invertible. It is understood by expanding $(e^x - 1)/x$ as a power series and substituting $x = \boldsymbol{A}$. Now focusing again on Equation (5), for DeltaNet, $\boldsymbol{A}_t = -\boldsymbol{k}_t\boldsymbol{k}_t^\top$ is a rank-1 matrix, which allows for further simplification:

$$f(\boldsymbol{x}\boldsymbol{y}^\top) = \sum_{n=0}^{\infty} a_n(\boldsymbol{x}\boldsymbol{y}^\top)^n = a_0\boldsymbol{I} + \sum_{n=1}^{\infty} a_n(\boldsymbol{x}\boldsymbol{y}^\top)^n = f(0)\boldsymbol{I} + \boldsymbol{x}\underbrace{\left(\sum_{n=1}^{\infty} a_n(\boldsymbol{y}^\top\boldsymbol{x})^{n-1}\right)}_{\frac{f(\boldsymbol{y}^\top\boldsymbol{x}) - f(0)}{\boldsymbol{y}^\top\boldsymbol{x}}}\boldsymbol{y}^\top \tag{8}$$

Since $\boldsymbol{y}^\top\boldsymbol{x}$ is a scalar, the essence of the simplification is converting a matrix function into a scalar function. From this, we obtain:

$$e^{\eta_t\boldsymbol{A}_t} = \boldsymbol{I} - \frac{1 - e^{-\eta_t\|\boldsymbol{k}_t\|^2}}{\|\boldsymbol{k}_t\|^2}\boldsymbol{k}_t\boldsymbol{k}_t^\top, \qquad \boldsymbol{B}_t\boldsymbol{A}_t^{-1}(e^{\eta_t\boldsymbol{A}_t} - \boldsymbol{I}) = \frac{1 - e^{-\eta_t\|\boldsymbol{k}_t\|^2}}{\|\boldsymbol{k}_t\|^2}\boldsymbol{v}_t\boldsymbol{k}_t^\top \tag{9}$$

5 Personal Thinking

This concludes our introduction to EFLA. The original paper also includes experimental results showing that EFLA has some advantages over the original DeltaNet. However, as seen from Equation (6), EFLA still maintains the DeltaNet form, so one should not expect "revolutionary" improvements. Why does EFLA generally perform slightly better? DeltaNet directly discards the magnitude of $\boldsymbol{K}$ through L2 Normalization, whereas the $\boldsymbol{v}_t\boldsymbol{k}_t^\top$ term in Equation (6) depends on $\|\boldsymbol{k}_t\|$. Thus, EFLA actually possesses an extra degree of freedom, leading to a higher theoretical upper bound.

Furthermore, the practice of constructing recursion using exact solutions of differential equations is not new. We mentioned it when introducing SSMs in "Revisiting SSM (II): Some Legacy Issues of HiPPO". The key result, Equation (5), already appeared in HiPPO. EFLA specifically expands the calculation for the special case of DeltaNet to obtain a simplified and usable result.

A more profound question is: what is the benefit of using differential equations as a starting point? It is easy to see that the eigenvalues of the transition matrix in Equation (6) are automatically within $(0, 1]$. In other words, the recursive form derived from solving the differential equation (4) is naturally more stable. Because differential equations come with continuity constraints, and the matrix $-\boldsymbol{k}_t\boldsymbol{k}_t^\top$ is semi-negative definite, according to differential equation theory, its solution is stable.

In mathematical modeling, a classic example is the Logistic equation $dx/dt = \alpha x - \beta x^2$. Its solution is simple—the Logistic function. However, the corresponding difference equation $x_{t+1} - x_t = \alpha x_t - \beta x_t^2$ can exhibit chaotic behavior (extreme sensitivity to initial conditions) under certain settings. Therefore, starting from a differential equation can automatically avoid some abnormal behaviors.

6 Summary

This article discussed the L2 Normalization of DeltaNet, primarily introducing the idea of reparameterizing DeltaNet from the perspective of differential equations. This can also be viewed as an explanation for the L2 Normalization of $\boldsymbol{K}$ in DeltaNet.