Taylor Expansion of LogSumExp and Softmax

Su Jianlin

2026-07-13

Recently, I saw the paper *The Key to Going Linear: Analysis-Driven Transformer Linearization*, which directly performs an approximate expansion on Softmax to linearize Attention. I gained some insights from it and will record them here.

Some Notations

First, we introduce the following notations:

$$\boldsymbol{x} = (x_1, x_2, \cdots, x_n) \in \mathbb{R}^n, \qquad \overline{\boldsymbol{x}} = \frac{1}{n}\sum_{i=1}^n x_i \tag{1}$$

$$\text{logsumexp}(\boldsymbol{x}) = \log \sum_{i=1}^n e^{x_i} = \log n + \log \overline{e^{\boldsymbol{x}}} \tag{2}$$

$$\text{softmax}(\boldsymbol{x}) = \frac{e^{\boldsymbol{x}}}{\sum_{i=1}^n e^{x_i}} = e^{\boldsymbol{x} - \text{logsumexp}(\boldsymbol{x})} \tag{3}$$

We agree that function operations on vectors are in the Hadamard sense, for example

$$e^{\boldsymbol{x}} = (e^{x_1}, e^{x_2}, \cdots, e^{x_n}), \qquad \boldsymbol{x}^2 = (x_1^2, x_2^2, \cdots, x_n^2) \tag{4}$$

In particular, carefully note the difference between $\overline{e^{\boldsymbol{x}}}$ and $e^{\overline{\boldsymbol{x}}}$, as well as $\overline{\boldsymbol{x}^2}$ and $\overline{\boldsymbol{x}}^2$: the former applies the function to each component first and then averages, while the latter averages first and then applies the function. The two are generally not equal.

In addition to the above definitions, logsumexp and softmax can also be connected through their gradients:

$$\text{softmax}(\boldsymbol{x}) = \nabla_{\boldsymbol{x}} \text{logsumexp}(\boldsymbol{x}) \tag{5}$$

Basic Expansion

Next, we expand logsumexp($\boldsymbol{x}$). Introducing $\boldsymbol{y} = \boldsymbol{x} - \overline{\boldsymbol{x}}$, we have

$$\begin{aligned}
\text{logsumexp}(\boldsymbol{x}) &= \log n + \overline{\boldsymbol{x}} + \log \overline{e^{\boldsymbol{y}}} \\
&= \log n + \overline{\boldsymbol{x}} + \log \left(\overline{1 + \boldsymbol{y} + \frac{\boldsymbol{y}^2}{2} + \frac{\boldsymbol{y}^3}{6} + \frac{\boldsymbol{y}^4}{24} + \cdots} \right) \\
&= \log n + \overline{\boldsymbol{x}} + \log \left(1 + \frac{\overline{\boldsymbol{y}^2}}{2} + \frac{\overline{\boldsymbol{y}^3}}{6} + \frac{\overline{\boldsymbol{y}^4}}{24} + \cdots \right) \\
&= \log n + \overline{\boldsymbol{x}} + \frac{\overline{\boldsymbol{y}^2}}{2} + \frac{\overline{\boldsymbol{y}^3}}{6} + \left(\frac{\overline{\boldsymbol{y}^4}}{24} - \frac{(\overline{\boldsymbol{y}^2})^2}{8} \right) + \cdots
\end{aligned} \tag{6}$$

The last step uses $\log(1+t) = t - t^2/2 + t^3/3 - \cdots$. If needed, one can continue to expand further. If the reader is not familiar with this, they can also leave it to Kimi to complete.

Introducing the bias of $\overline{\boldsymbol{x}}$ here is a small trick for simplification. If it is not introduced, an equivalent result can still be obtained, but the various terms of $\boldsymbol{y}$ will be explicitly expanded:

$$\text{logsumexp}(\boldsymbol{x}) = \log n + \overline{\boldsymbol{x}} + \underbrace{\left(\frac{\overline{\boldsymbol{x}^2}}{2} - \frac{\overline{\boldsymbol{x}}^2}{2} \right)}_{\overline{\boldsymbol{y}^2}/2} + \underbrace{\left(\frac{\overline{\boldsymbol{x}^3}}{6} - \frac{\overline{\boldsymbol{x}}\,\overline{\boldsymbol{x}^2}}{2} + \frac{\overline{\boldsymbol{x}}^3}{3} \right)}_{\overline{\boldsymbol{y}^3}/6} + \cdots \tag{7}$$

Finding the Gradient

For softmax($\boldsymbol{x}$), we directly use the identity (5) and differentiate both sides of Equation (6) to obtain the expansion of softmax($\boldsymbol{x}$). To do this, we utilize

$$\nabla_{\boldsymbol{x}} \overline{\boldsymbol{x}} = \frac{\mathbf{1}}{n}, \qquad \nabla_{\boldsymbol{x}} \overline{\boldsymbol{y}^m} = \frac{m}{n} \left(\boldsymbol{y}^{m-1} - \overline{\boldsymbol{y}^{m-1}} \right) \tag{8}$$

to get

$$\text{softmax}(\boldsymbol{x}) = \frac{1}{n} \left(1 + \boldsymbol{y} + \left(\frac{\boldsymbol{y}^2}{2} - \frac{\boldsymbol{y}^2}{2} \right) + \left(\frac{\boldsymbol{y}^3}{6} - \frac{\overline{\boldsymbol{y}^3}}{6} - \frac{\overline{\boldsymbol{y}^2}\,\boldsymbol{y}}{2} \right) + \cdots \right) \tag{9}$$

Note that if a finite number of terms are truncated, the right-hand side can still maintain the property that the sum of all components is 1 (which can be seen from the gradient form of $\nabla_{\boldsymbol{x}} \overline{\boldsymbol{y}^m}$), but it cannot guarantee the non-negativity of each component. This point needs to be noted in practical applications (for example, when taking the logarithm to calculate entropy).

Besides finding the gradient, we can also expand using $\text{softmax}(\boldsymbol{x}) = e^{\boldsymbol{x}-\text{logsumexp}(\boldsymbol{x})} = e^{\boldsymbol{y}-\text{logsumexp}(\boldsymbol{y})}$. Combining this with Equation (6), we get:

$$\begin{aligned} e^{\boldsymbol{y}}e^{-\text{logsumexp}(\boldsymbol{y})} &= \left(1 + \boldsymbol{y} + \frac{\boldsymbol{y}^2}{2} + \frac{\boldsymbol{y}^3}{6} + \cdots\right)\exp\left(-\log n - \frac{\overline{\boldsymbol{y}^2}}{2} - \frac{\overline{\boldsymbol{y}^3}}{6} - \cdots\right) \\ &= \frac{1}{n}\left(1 + \boldsymbol{y} + \left(\frac{\boldsymbol{y}^2}{2} - \frac{\boldsymbol{y}^2}{2}\right) + \left(\frac{\boldsymbol{y}^3}{6} - \frac{\overline{\boldsymbol{y}^3}}{6} - \frac{\overline{\boldsymbol{y}^2}\boldsymbol{y}}{2}\right) + \cdots\right) \end{aligned} \tag{10}$$

Sparse Attention

So what are the applications of these two expansions? First is the one for $\text{logsumexp}(\boldsymbol{x})$, which is related to the block scoring of Block Sparse Attention like MoBA. We know that Full Attention scores each Token according to $e^{\boldsymbol{q}\cdot\boldsymbol{k}}$, so for a certain block $\mathcal{B}$, its score can be reasonably defined as the sum of the scores of each Token, which is equivalent to

$$s_{\mathcal{B}} = \log\sum_{t\in\mathcal{B}} e^{\boldsymbol{q}\cdot\boldsymbol{k}_t} \tag{11}$$

Its first-order approximation is $\log|\mathcal{B}| + \boldsymbol{q}\cdot\overline{\boldsymbol{k}}$, where $\overline{\boldsymbol{k}} = \frac{1}{|\mathcal{B}|}\sum_{t\in\mathcal{B}}\boldsymbol{k}_t$. This exactly corresponds to the empirical practice in MoBA of using AvgPooling as the intra-block Landmark vector. If the first-order approximation feels too coarse, one can consider higher-order corrections. According to Equation (6), the second-order term is

$$\frac{1}{2|\mathcal{B}|}\sum_{t\in\mathcal{B}}(\boldsymbol{q}\cdot(\boldsymbol{k}_t - \overline{\boldsymbol{k}}))^2 = \frac{1}{2}\boldsymbol{q}^\top\underbrace{\left(\frac{1}{|\mathcal{B}|}\sum_{t\in\mathcal{B}}(\boldsymbol{k}_t - \overline{\boldsymbol{k}})(\boldsymbol{k}_t - \overline{\boldsymbol{k}})^\top\right)}_{\boldsymbol{\Sigma}}\boldsymbol{q} \tag{12}$$

That is, the second-order approximation is $\log|\mathcal{B}| + \boldsymbol{q}\cdot\overline{\boldsymbol{k}} + \boldsymbol{q}^\top\boldsymbol{\Sigma}\boldsymbol{q}/2$, where $\boldsymbol{\Sigma}$ is exactly the covariance matrix of $\boldsymbol{k}_t$ within the block. Furthermore, we can consider a diagonal approximation to reduce computational costs. The idea of this higher-order correction is essentially the same as SPLA. Subsequent related work also includes HiLS, which additionally learns a non-equally weighted average central vector to replace $\overline{\boldsymbol{k}}$.

Linear Attention

As for the application of the $\text{softmax}(\boldsymbol{x})$ expansion, it is naturally the linearization of Attention. In *Transformer Upgrade Path: 5, Linear Attention as Infinite Dimensions*, we summarized three approaches to linearization, all of which approximate $e^{\boldsymbol{q}\cdot\boldsymbol{k}}$. However, after approximating $e^{\boldsymbol{q}\cdot\boldsymbol{k}}$, it still needs to be normalized. It might be better to directly approximate the normalized $\text{softmax}(\boldsymbol{x})$.

Let $\boldsymbol{x} = (\boldsymbol{q}\cdot\boldsymbol{k}_1, \cdots, \boldsymbol{q}\cdot\boldsymbol{k}_n)$, then we have the first-order approximation:

$$\text{softmax}(\boldsymbol{x})_i \approx \frac{1}{n} + \boldsymbol{q}\cdot(\boldsymbol{k}_i - \overline{\boldsymbol{k}}) \tag{13}$$

This is exactly the scheme discussed in the paper at the beginning, and it is also very intuitive itself: the average vector $\overline{\boldsymbol{k}}$ provides a baseline for the relative magnitude of attention. Tokens with a similarity greater than $\overline{\boldsymbol{k}}$ should have increased attention, otherwise it is decreased. To improve accuracy, Based expands $e^{\boldsymbol{q}\cdot\boldsymbol{k}}$ to the second order (which just ensures non-negativity, refer here), but from our perspective, considering the second-order approximation of $\text{softmax}(\boldsymbol{x})$ is more scientific. The second-order term is

$$\frac{1}{2n}\left[(\boldsymbol{q}\cdot(\boldsymbol{k}_i - \overline{\boldsymbol{k}}))^2 - \frac{1}{n}\sum_{j=1}^{n}(\boldsymbol{q}\cdot(\boldsymbol{k}_j - \overline{\boldsymbol{k}}))^2\right] \tag{14}$$

where $(\boldsymbol{q}\cdot(\boldsymbol{k}_i - \overline{\boldsymbol{k}}))^2$ can be transformed into an inner product by introducing an outer product:

$$(\boldsymbol{q}\cdot(\boldsymbol{k}_i - \overline{\boldsymbol{k}}))^2 = \boldsymbol{q}^\top(\boldsymbol{k}_i - \overline{\boldsymbol{k}})(\boldsymbol{k}_i - \overline{\boldsymbol{k}})^\top\boldsymbol{q} = \left\langle \boldsymbol{q}\boldsymbol{q}^\top, (\boldsymbol{k}_i - \overline{\boldsymbol{k}})(\boldsymbol{k}_i - \overline{\boldsymbol{k}})^\top\right\rangle_F \tag{15}$$

So, similar to Based, truncating the Softmax in Softmax Attention to the second-order approximation also results in linear attention.

Article Summary

This article derived the Taylor expansions of LogSumExp and Softmax respectively, and discussed two of their potential applications.