A Simple Derivation of Rotary Position Encoding for Large Language Models

Authors: He Cangping, Xu Tao, He Cangping

Date: 2023-07-12T00:00:00+00:00

Abstract

Open-source large language models exemplified by LLaMA have widely adopted rotary position embedding, with the original paper employing complex function derivations. This paper instead utilizes a linear algebra-based derivation, aiming to better understand this encoding method; it also identifies a potential issue with this approach and proposes recommendations for improvement.

Full Text

Preamble

Easy Derivation of Rotary Position Embeddings for Large Language Models

He Cangping, Xu Tao
cangping@staff.weibo.com, xutao@sugon.com
Sugon Information Industry (Beijing) Co., Ltd.

Rotary Position Embeddings (RoPE) is widely used in open-source large language models such as LLAMA. While the original paper derives the method using complex functions, this article presents an alternative derivation using linear algebra to facilitate better understanding of the encoding technique. We also identify a potential issue with the method and propose an improvement.

Keywords: Large Language Model (LLM), Rotary Position Embeddings (RoPE), LLAMA

Abstract

The release of ChatGPT in November 2022 ignited a new wave of technological innovation. Following the open-sourcing of the LLAMA model [2], numerous large language models have been released, such as Baichuan-7B, which adopts the same architecture as LLAMA. Rotary Position Embeddings (RoPE) [1] is

a crucial component of the LLAMA model. The original paper derives RoPE using complex function theory, which can be difficult to understand for those unfamiliar with complex analysis. This paper attempts to derive the method using more familiar linear algebra, hoping to make this excellent encoding technique accessible to a broader audience.
Completion date: July 10, 2023

2. Function Definitions

As preliminary groundwork, this section defines several functions. Since PyTorch organizes arrays in row-major order with zero-based indexing, we adopt the same conventions for vectors and matrices in this paper, using zero-based indices for matrix elements.

For any given positive integers m and n, row vectors are denoted by bold lowercase letters in the form $x = (x_0, x_1, \dots, x_{n-1})$. Matrices are denoted by uppercase letters. The softmax function (softmax) is defined as follows:

For a row vector x, $\text{smax}(x) = \frac{(e^{x_0}, e^{x_1}, \dots, e^{x_{n-1}})}{\sum_{i=0}^{n-1} e^{x_i}}$. For a matrix X, $\text{smax}(X)$ applies the softmax operation to each row: $\text{smax}(X) = (\text{smax}(x_{0:}); \text{smax}(x_{1:}); \dots; \text{smax}(x_{m-1:}))$, where $x_{i:} = (x_{i0}, x_{i1}, \dots, x_{i,n-1})$ and the semicolon in parentheses denotes a new row.

3. Rotary Position Embeddings

In the LLAMA model, the core operation of self-attention involves computing the softmax function using given query matrix Q and key matrix K:

$$R = QK^T$$

Here, Q has dimensions $n_3 \times n_6$, where n_3 is the current sequence length (incremented by 1 after each token generation) and n_6 is the width of a single attention head ($n_6 = 128$ in LLAMA-7B). Matrix K has the same dimensions as Q but with different element values. Let $R = QK^T$, where matrix R has dimensions $n_3 \times n_3$. We represent matrices Q, K, and R as:

$$Q = \begin{pmatrix} q_{0:} \\ q_{1:} \\ \vdots \\ q_{n_3-1:} \end{pmatrix}, \quad K = \begin{pmatrix} k_{0:} \\ k_{1:} \\ \vdots \\ k_{n_3-1:} \end{pmatrix}, \quad R = \begin{pmatrix} r_{0,0} & r_{0,1} & \cdots & r_{0,n_3-1} \\ r_{1,0} & r_{1,1} & \cdots & r_{1,n_3-1} \\ \vdots & \vdots & \ddots & \vdots \\ r_{n_3-1,0} & r_{n_3-1,1} & \cdots & r_{n_3-1,n_3-1} \end{pmatrix}$$

Vectors $q_{i:}$ and $k_{j:}$ correspond to the i-th and j-th tokens in the current sequence, where $i, j = 0, 1, \dots, n_3 - 1$. The scalar r_{ij} is the inner product of $q_{i:}$ and $k_{j:}$, i.e., $r_{ij} = q_{i:}k_{j:}^T$.

The goal of position encoding is to apply a transformation f to vectors $q_{i:}$ and $k_{j:}$ that incorporates absolute position information i and j, respectively:

$$\hat{r}_{ij} = f(q_{i:})f(k_{j:})^T$$

This transformation must satisfy the following requirements:

- **(c1)** When $i = j$, the inner product remains unchanged after transformation: $\hat{r}_{ii} = r_{ii}$. In the token sequence, this means a token's self-inner product at any position is unaffected by f.
- **(c2)** After transformation, the inner product $\hat{r}_{ij}$ contains only relative position information $i - j$, not absolute position information.
- **(c3)** For fixed vectors $q_{i:}$ and $k_{j:}$, the larger $|i-j|$ is, the smaller $|\hat{r}_{ij} - r_{ij}|$ becomes. In the token sequence, this means tokens farther apart influence each other less.

Following standard research methodology, we first examine the simplest case where $n_6 = 2$. The vectors are $q_{i:} = (q_{i0}, q_{i1})^T$ and $k_{j:} = (k_{j0}, k_{j1})^T$. Assume f is a linear transformation: $f(q_{i:}) = q_{i:}A_i$, where A_i is a 2×2 matrix. Then:

$$\hat{r}_{ij} = f(q_{i:})f(k_{j:})^T = q_{i:}A_i(k_{j:}A_j)^T = q_{i:}A_iA_j^Tk_{j:}^T$$

Applying requirement **(c1)** to this equation yields $A_iA_i^T = I$ and $A_i \neq I$. Equation (3) requires A_i to be an orthogonal matrix. A common orthogonal matrix in linear algebra is the rotation matrix:

$$I_\theta = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix}$$

It is easy to verify that $I_\theta I_\theta^T = I$. Geometrically, $q_{i:}I_\theta$ rotates vector $q_{i:}$ by angle θ. The expression $q_{i:}I_\theta I_\theta^T$ rotates $q_{i:}$ by θ and then by $-\theta$, with the two rotations canceling out, leaving the vector unchanged. If the two rotation angles differ, the net effect is rotation by the difference angle. Therefore, letting the rotation angles at positions i and j be θ_i and θ_j, respectively, we can verify that $I_{\theta_i}I_{\theta_j}^T = I_{\theta_i - \theta_j}$.

Let $\theta_{ij} = \theta_i - \theta_j$. Substituting $A_i = I_{\theta_i}$ and $A_j = I_{\theta_j}$ into equation (2) gives:

$$\hat{r}_{ij} = q_{i:}(I_{\theta_i}I_{\theta_j}^T)k_{j:}^T = q_{i:}I_{\theta_{ij}}k_{j:}^T$$

Clearly, this expression contains only relative position information θ_{ij}, satisfying requirement **(c2)**.

For each position $i = 0, 1, \ldots, n_3 - 1$ in the token sequence, there must be a corresponding θ_i. The original paper [1] chooses an arithmetic sequence: select a base value $\bar{\theta}_0$, then set $\theta_i = i\bar{\theta}_0$.

When n_6 is an even number greater than 2, we can process dimensions in pairs. For a row vector $q_{i:} = (q_{i0}, q_{i1}, \ldots, q_{i,n_6-1})$ of length n_6, the rotation transformation $q_{i:}A_i$ uses a block-diagonal matrix:

$$A_i = \begin{pmatrix} \cos i\bar{\theta}_0 & -\sin i\bar{\theta}_0 & 0 & 0 & \cdots \\ \sin i\bar{\theta}_0 & \cos i\bar{\theta}_0 & 0 & 0 & \cdots \\ 0 & 0 & \cos i\bar{\theta}_2 & -\sin i\bar{\theta}_2 & \cdots \\ 0 & 0 & \sin i\bar{\theta}_2 & \cos i\bar{\theta}_2 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{pmatrix}$$

For the rotation angles $\bar{\theta}_t$ where $t = 0, 2, 4, \ldots, n_6 - 2$, the original paper [1] uses fixed values $\bar{\theta}_t = 10000^{-t/n_6}$.

[Figure 1: see original paper] Rotation angle in inner product when $\bar{\theta}_0 = 1$
[Figure 2: see original paper] Rotation angle in inner product when $\bar{\theta}_0 = \frac{2\times 2048}{10000}$

4. Concerns and Recommendations

According to equation (4) with $\bar{\theta}_0 = 1$, for token position differences $j - i = 1, 2, \ldots, 50$ and the first two dimensions of the vectors, the angular difference in the inner product operation is $\theta_{ij} = (j - i)\bar{\theta}_1$ modulo 2π. The variation trend is shown in Figure 1. When $j - i = 6$, the angular difference is $\theta_{ij} = 6$; when $j - i = 7$, it suddenly drops to $\theta_{ij} = 0.7168$. This means that tokens farther apart can have smaller rotation angle differences, violating requirement **(c3)**.

This paper proposes modifying equation (4) to incorporate the maximum sequence length n_9 (e.g., $n_9 = 2048$). The modified formulation ensures that rotation angle differences increase monotonically with token distance, as demonstrated in Figure 2, thereby satisfying requirement **(c3)**.

References

[1] Jianlin Su. *RoFormer: Transformer with Rotary Position Embeddings - ZhuiyiAI*. Tech. rep. 2021. URL: https://github.com/ZhuiyiTechnology/roformer.

[2] Hugo Touvron et al. "LLaMA: Open and Efficient Foundation Language Models". In: (2023). arXiv: 2302.13971 [cs.CL].

Note: Figure translations are in progress. See original paper for figures.