位置编码

相对位置编码

首先区分下绝对位置编码和相对位置编码：

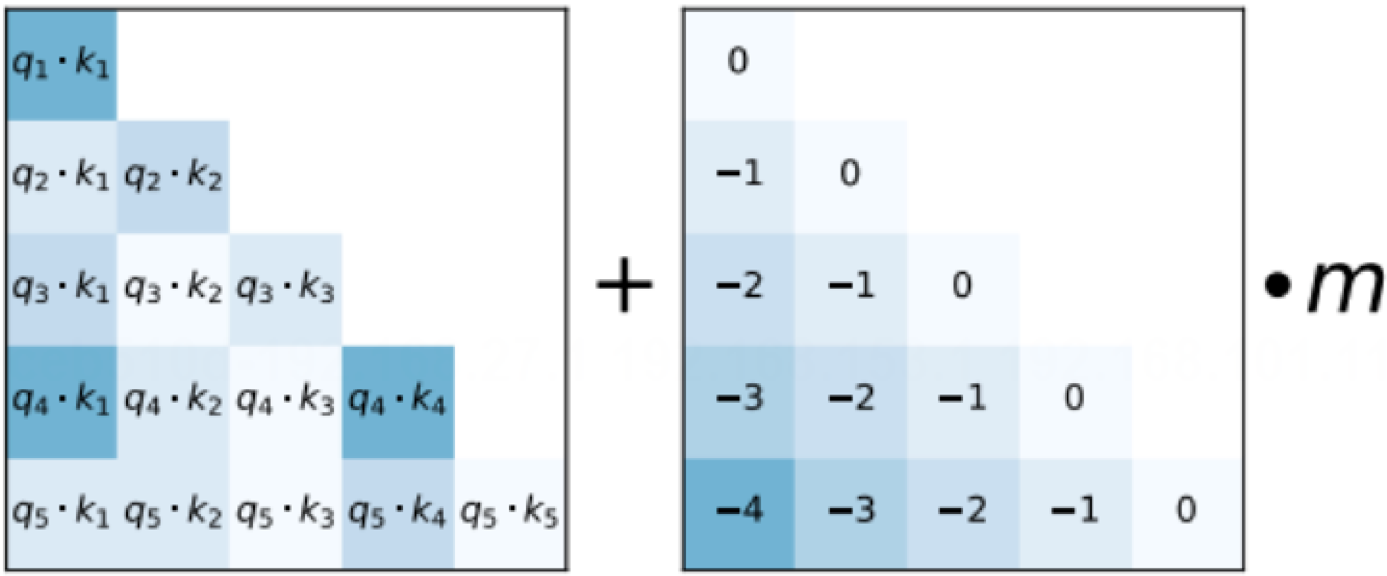
- 绝对位置编码直接将token的绝对位置信息添加到对应的 q_t , k_s 中,例如循环位置编码。

$$
\begin{aligned}
\boldsymbol{A}_{t,s} &= \boldsymbol{q}_t^T \boldsymbol{k}_s = (\boldsymbol{x}_t + \boldsymbol{p}_t)^T \boldsymbol{W}_Q^T \boldsymbol{W}_K (\boldsymbol{x}_s + \boldsymbol{p}_s) \\
\boldsymbol{A}_{t,s} &= \boldsymbol{x}_t^T \boldsymbol{W}_Q^T \boldsymbol{W}_K \boldsymbol{x}_s + \boldsymbol{x}_t^T \boldsymbol{W}_Q^T \boldsymbol{W}_K \boldsymbol{p}_s + \boldsymbol{p}_t^T \boldsymbol{W}_Q^T \boldsymbol{W}_K \boldsymbol{x}_s \\
&\quad + \boldsymbol{p}_t^T \boldsymbol{W}_Q^T \boldsymbol{W}_K \boldsymbol{p}_s \qquad (2) \\
\boldsymbol{p}_t &= \left[\cdots, \sin \frac{t}{10000^{2n/d}}, \cos \frac{t}{10000^{2n/d}}, \cdots\right]^T \in \mathbb{R}^d
\end{aligned}
$$

- 相对位置编码则将两个token的相对位置信息添加到对应的attention值中。

1. Attention with linear biases enables input length extrapolation（ALiBi）

在计算attention时，对前边位置的分数进行惩罚，如图所示：



- 传统的绝对位置编码（如正弦/余弦编码）在训练时会为每个位置分配一个固定的向量，模型可能会过度拟合这些特定长度的模式。而ALiBi通过在注意力分数计算中直接使用线性偏置，减少了模型对特定序列长度的依赖，从而提高了对未见过的序列长度的泛化能力。

- ALiBi的位置偏差随距离线性增长，这种设计让模型在处理不同长度的序列时，可以自然地根据距离调整注意力权重，无需显式学习位置编码的复杂周期性结构。
- 由于线性偏置的引入直接与序列中元素的位置相关，没有固定大小的编码矩阵限制，理论上模型可以更容易地处理任意长度的序列，从而展现出良好的长度外推性能。
- 通过直接在注意力分数上施加与距离相关的线性惩罚，ALiBi鼓励模型关注更近的位置，同时不完全排除远处的依赖，从而在一定程度上平衡了局部和全局依赖的学习，这对于处理长序列尤其有利。

2. XLENT

从绝对位置编码出发：

$$\boldsymbol{q}_i\boldsymbol{k}_j^\top = \boldsymbol{x}_i\boldsymbol{W}_Q\boldsymbol{W}_K^\top\boldsymbol{x}_j^\top + \boldsymbol{x}_i\boldsymbol{W}_Q\boldsymbol{W}_K^\top\boldsymbol{p}_j^\top + \boldsymbol{p}_i\boldsymbol{W}_Q\boldsymbol{W}_K^\top\boldsymbol{x}_j^\top + \boldsymbol{p}_i\boldsymbol{W}_Q\boldsymbol{W}_K^\top\boldsymbol{p}_j^\top \tag{7}$$

XLENT将 p_j 替换成相对位置向量 R_{i-j} ， p_i 替换成可训练的向量u和v，

$$\boldsymbol{x}_i\boldsymbol{W}_Q\boldsymbol{W}_K^\top\boldsymbol{x}_j^\top + \boldsymbol{x}_i\boldsymbol{W}_Q\boldsymbol{W}_K^\top\boldsymbol{R}_{i-j}^\top + \boldsymbol{u}\boldsymbol{W}_Q\boldsymbol{W}_K^\top\boldsymbol{x}_j^\top + \boldsymbol{v}\boldsymbol{W}_Q\boldsymbol{W}_K^\top\boldsymbol{R}_{i-j}^\top \tag{8}$$

$$\boldsymbol{r}_{t-s} = \left[\cdots, \sin\frac{t-s}{10000^{2n/d}}, \cos\frac{t-s}{10000^{2n/d}}, \cdots\right]^T \in \mathbb{R}^d$$

3. T5

（7）式中的每一项可以理解为“输入-输入”、“输入-位置”、“位置-输入”、“位置-位置”四项注意力的组合，由于输入信息与位置信息应该是独立（解耦）的，它们不应该有过多的交互，所以“输入-位置”、“位置-输入”两项Attention可以删掉，“位置-位置”实际上是一个依赖于（s,）的一个标量。

此外，通过固定的桶函数 b(t-s)，将 t-s 从 [−128,128] 压缩至 [0,31]，再对每个 b(t-s) 训练对应的偏移量 $r_{b(t-s)}$

$$\boldsymbol{A}_{t,s} = \boldsymbol{x}_t^T\boldsymbol{W}_Q^T\boldsymbol{W}_K\boldsymbol{x}_s + r_{b(t-s)} \tag{4}$$

4. DeBERTa

与T5相反，扔掉“位置-位置”一项只保留剩下三项，通过通过 $\delta(t,s)$ 将 t−s 直接截断在区间 (−k,k] 内

$$\boldsymbol{A}_{t,s} = \boldsymbol{x}_t^T\boldsymbol{W}_Q^T\boldsymbol{W}_K\boldsymbol{x}_s + \boldsymbol{x}_t^T\boldsymbol{W}_Q^T\boldsymbol{W}_K\boldsymbol{P}_{\delta(t,s)} + \boldsymbol{x}_s^T\boldsymbol{W}_K^T\boldsymbol{W}_Q\boldsymbol{P}_{\delta(s,t)} \tag{7}$$

DeBERTa在softmax时校正系数为 $\sqrt{3d}$ ，不是默认的 $\sqrt{d}$ 。此外，指出NLP的大多数任务可能都只需要相对位置信息，但确实有些场景下绝对位置信息更有帮助，于是它将整个模型分为两部分来理

解。以Base版的MLM预训练模型为例，它一共有13层，前11层只是用相对位置编码，这部分称为Encoder，后面2层加入绝对位置信息，这部分它称之为Decoder；至于下游任务的微调截断，则是使用前11层的Encoder加上1层的Decoder来进行。

RoPE

RoPE实现了绝对位置编码和相对位置编码的统一，它通过绝对位置编码的形式，实现了相对位置编码的效果。

RoPE将输入序列的位置信息通过旋转操作嵌入到self-attention的计算中，不同位置的 token 可以有不同的旋转角度，从而嵌入位置信息，从而增强模型对长序列和相对位置的处理能力。RoPE是可学习的，在自注意力公式中结合了明确的相对位置依赖性。

RoPE保持了序列长度的灵活性、随相对距离的增加而衰减的token间依赖性。其原理如下图，$x_m^{'}$ 表示经过RoPE后的结果：

从内积角度

证明目标:在x_m表达式为 (1) 的情况下， 内积能够表示成位置的函数，即表达式（3）成立

$$x_m^{'} = W_q x_m e^{im\theta} = (W_q x_m) e^{im\theta} = q_m e^{im\theta} \qquad (1)$$

$$x_n^{'} = W_k x_n e^{in\theta} = (W_k x_n) e^{in\theta} = k_n e^{in\theta} \qquad (2)$$

假设词向量只有2维

$$x_m^{'T} x_n^{'} = \begin{pmatrix} q_m^1 & q_m^2 \end{pmatrix} \begin{pmatrix} cos((m-n)\theta) & -sin((m-n)\theta) \\ sin((m-n)\theta) & cos((m-n)\theta) \end{pmatrix} \begin{pmatrix} k_n^1 \\ k_n^2 \end{pmatrix} \qquad (3)$$

从内积的角度推导：

$$x_m^{'} = W_q x_m e^{im\theta} = (W_q x_m) e^{im\theta} = \boxed{q_m} \boxed{e^{im\theta}}$$

$$q_m = \begin{pmatrix} W_q^{11} & W_q^{12} \\ W_q^{21} & W_q^{22} \end{pmatrix} \begin{pmatrix} x_m^1 \\ x_m^2 \end{pmatrix} = \begin{pmatrix} q_m^1 \\ q_m^2 \end{pmatrix}$$

二维向量可以表示成虚数形式
应为向量的一个维度表示一个坐标
类比到实数坐标和虚数坐标

$$q_m = q_m^1 + \mathrm{i} q_m^2$$

$$e^{im\theta} = cos(m\theta) + \mathrm{i}sin(m\theta) \quad \text{欧拉公式}$$

$$\mathrm{i}^2 = -1$$

$$= q_m e^{im\theta} = (q_m^1 + \mathrm{i} q_m^2)(cos(m\theta) + \mathrm{i}sin(m\theta))$$

$$= (q_m^1 cos(m\theta) - q_m^2 sin(m\theta)) + \mathrm{i}(q_m^2 cos(m\theta) + q_m^1 sin(m\theta))$$

再写成向量形式

$$= [q_m^1 cos(m\theta) - q_m^2 sin(m\theta), q_m^2 cos(m\theta) + q_m^1 sin(m\theta)] = \begin{pmatrix} cos(m\theta) & -sin(m\theta) \\ sin(m\theta) & cos(m\theta) \end{pmatrix} \begin{pmatrix} q_m^1 \\ q_m^2 \end{pmatrix}$$

带入到内积中

$$x_m^{'T} x_n^{'} = \left(\begin{pmatrix} cos(m\theta) & -sin(m\theta) \\ sin(m\theta) & cos(m\theta) \end{pmatrix} \begin{pmatrix} q_m^1 \\ q_m^2 \end{pmatrix} \right)^T \begin{pmatrix} cos(n\theta) & -sin(n\theta) \\ sin(n\theta) & cos(n\theta) \end{pmatrix} \begin{pmatrix} k_n^1 \\ k_n^2 \end{pmatrix}$$

$$= \begin{pmatrix} q_m^1 & q_m^2 \end{pmatrix} \begin{pmatrix} cos(m\theta) & sin(m\theta) \\ -sin(m\theta) & cos(m\theta) \end{pmatrix} \begin{pmatrix} cos(n\theta) & -sin(n\theta) \\ sin(n\theta) & cos(n\theta) \end{pmatrix} \begin{pmatrix} k_n^1 \\ k_n^2 \end{pmatrix}$$

$$
\begin{aligned}
x_m^{'T} x_n^{'} &= \left(\begin{pmatrix} cos(m\theta) & -sin(m\theta) \\ sin(m\theta) & cos(m\theta) \end{pmatrix} \begin{pmatrix} q_m^1 \\ q_m^2 \end{pmatrix} \right)^T \begin{pmatrix} cos(n\theta) & -sin(n\theta) \\ sin(n\theta) & cos(n\theta) \end{pmatrix} \begin{pmatrix} k_n^1 \\ k_n^2 \end{pmatrix} \\
&= \begin{pmatrix} q_m^1 & q_m^2 \end{pmatrix} \begin{pmatrix} cos(m\theta) & sin(m\theta) \\ -sin(m\theta) & cos(m\theta) \end{pmatrix} \begin{pmatrix} cos(n\theta) & -sin(n\theta) \\ sin(n\theta) & cos(n\theta) \end{pmatrix} \begin{pmatrix} k_n^1 \\ k_n^2 \end{pmatrix} \\
&= \begin{pmatrix} q_m^1 & q_m^2 \end{pmatrix} \begin{pmatrix} cos(m\theta)cos(n\theta) + sin(m\theta)sin(n\theta) & -cos(m\theta)sin(n\theta) + sin(m\theta)cos(n\theta) \\ -sin(m\theta)cos(n\theta) + cos(m\theta)sin(n\theta) & sin(m\theta)sin(n\theta) + cos(m\theta)cos(n\theta) \end{pmatrix} \begin{pmatrix} k_n^1 \\ k_n^2 \end{pmatrix} \\
&= \begin{pmatrix} q_m^1 & q_m^2 \end{pmatrix} \begin{pmatrix} cos((m-n)\theta) & -sin((m-n)\theta) \\ sin((m-n)\theta) & cos((m-n)\theta) \end{pmatrix} \begin{pmatrix} k_n^1 \\ k_n^2 \end{pmatrix}
\end{aligned}
$$

$$
\begin{aligned}
sin(m\theta + n\theta) &= sin(m\theta)cos(n\theta) + cos(m\theta)sin(n\theta) \\
sin(m\theta - n\theta) &= sin(m\theta)cos(n\theta) - cos(m\theta)sin(n\theta) \\
cos(m\theta + n\theta) &= cos(m\theta)cos(n\theta) - sin(m\theta)sin(n\theta) \\
cos(m\theta - n\theta) &= cos(m\theta)cos(n\theta) + sin(m\theta)sin(n\theta)
\end{aligned}
$$

对于多维的旋转位置编码，可以简化为以下形式：

$$
\begin{pmatrix}
\cos m\theta_0 & -\sin m\theta_0 & 0 & 0 & \cdots & 0 & 0 \\
\sin m\theta_0 & \cos m\theta_0 & 0 & 0 & \cdots & 0 & 0 \\
0 & 0 & \cos m\theta_1 & -\sin m\theta_1 & \cdots & 0 & 0 \\
0 & 0 & \sin m\theta_1 & \cos m\theta_1 & \cdots & 0 & 0 \\
\vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\
0 & 0 & 0 & 0 & \cdots & \cos m\theta_{d/2-1} & -\sin m\theta_{d/2-1} \\
0 & 0 & 0 & 0 & \cdots & \sin m\theta_{d/2-1} & \cos m\theta_{d/2-1}
\end{pmatrix}
\begin{pmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \\ \vdots \\ q_{d-2} \\ q_{d-1} \end{pmatrix}
$$

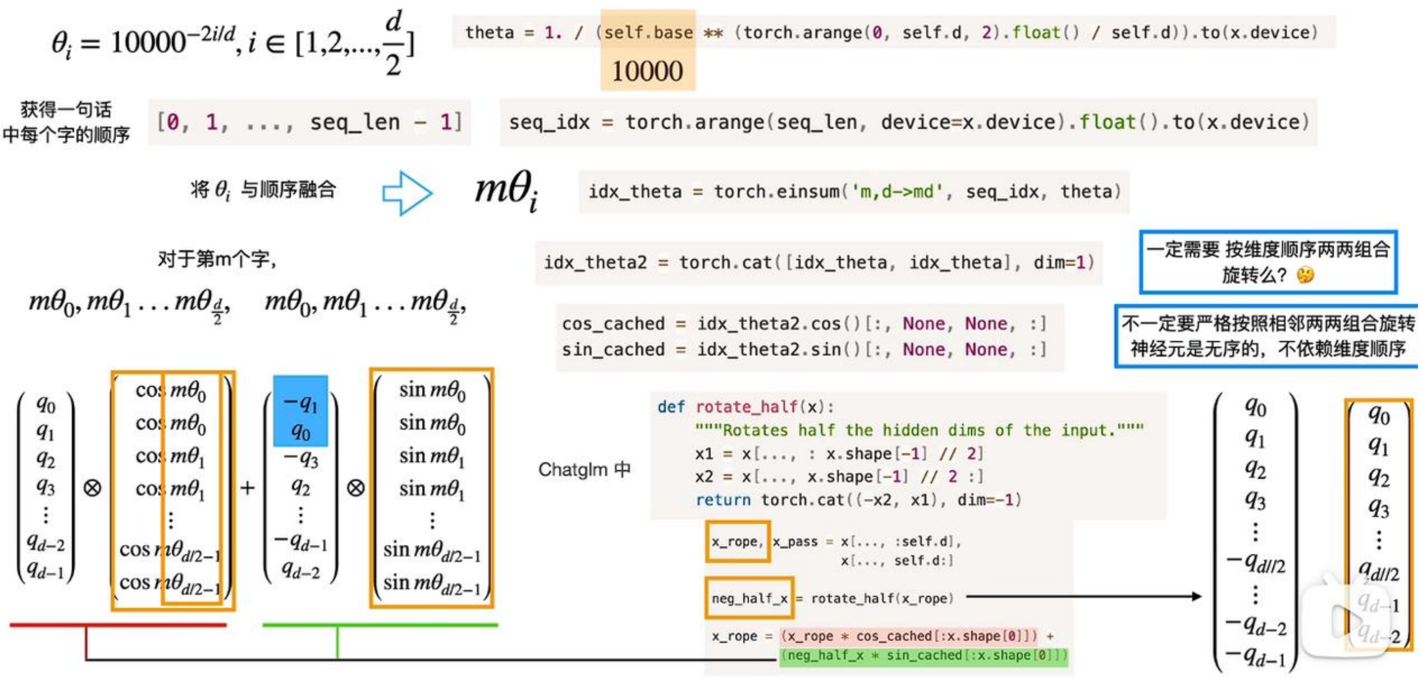
可以看到矩阵中有很多元素是0，
如果用矩阵乘法，其实有很多计算是无用的 🤔

$$
\begin{pmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \\ \vdots \\ q_{d-2} \\ q_{d-1} \end{pmatrix} \otimes
\begin{pmatrix} \cos m\theta_0 \\ \cos m\theta_0 \\ \cos m\theta_1 \\ \cos m\theta_1 \\ \vdots \\ \cos m\theta_{d/2-1} \\ \cos m\theta_{d/2-1} \end{pmatrix} +
\begin{pmatrix} -q_1 \\ q_0 \\ -q_3 \\ q_2 \\ \vdots \\ -q_{d-1} \\ q_{d-2} \end{pmatrix} \otimes
\begin{pmatrix} \sin m\theta_0 \\ \sin m\theta_0 \\ \sin m\theta_1 \\ \sin m\theta_1 \\ \vdots \\ \sin m\theta_{d/2-1} \\ \sin m\theta_{d/2-1} \end{pmatrix}
$$

$$\theta_i = 10000^{-2i/d}, i \in [1,2,...,\frac{d}{2}]$$

d 一定是偶数

结合代码来看，在chatglm中，因为内积计算与顺序无关，巧妙地将所有负数和正数分开



从公式的角度推理，要计算m和n之间的距离：

$$g(x'_m, x'_n, m-n) = Re[(W_q x_m)(W_k x_n)^* e^{i(m-n)\theta}]$$

Re 表示，取实数部分的结果

* 表示，共轭 复数 $k_n = k_n^1 + ik_n^2$ $k_n^* = k_n^1 - ik_n^2$

$$= Re[q_m k_n^* e^{i(m-n)\theta}]$$

$$= Re[(q_m^1 + iq_m^2)(k_n^1 - ik_n^2)(cos((m-n)\theta) + isin((m-n)\theta))]$$

欧拉公式

$$e^{i(m-n)\theta} = cos((m-n)\theta) + isin((m-n)\theta)$$

$$= Re[((q_m^1 k_n^1 + q_m^2 k_n^2) + i(q_m^2 k_n^1 - q_m^1 k_n^2))(cos((m-n)\theta) + isin((m-n)\theta))]$$

$$= Re[((q_m^1 k_n^1 + q_m^2 k_n^2)cos((m-n)\theta) - (q_m^2 k_n^1 - q_m^1 k_n^2)sin((m-n)\theta)) + (q_m^1 k_n^1 + q_m^2 k_n^2)isin((m-n)\theta)) + i(q_m^2 k_n^1 - q_m^1 k_n^2)cos((m-n)\theta)]$$

不要复数，只保留实数部分

$$= (q_m^1 k_n^1 + q_m^2 k_n^2)cos((m-n)\theta) - (q_m^2 k_n^1 - q_m^1 k_n^2)sin((m-n)\theta)$$

总结下RoPE的流程：首先计算得到q和k矩阵，然后对q和k向量的元素按照两两一组应用RoPE，例如：

$$\mathbf{f}(\mathbf{x}, m) = [(x_0 + \mathrm{i}x_1)e^{\mathrm{i}m\theta_0}, (x_2 + \mathrm{i}x_3)e^{\mathrm{i}m\theta_1}, \dots, (x_{d-2} + \mathrm{i}x_{d-1})e^{\mathrm{i}m\theta_{d/2-1}}]^\top$$

那么对于m和n之间的考虑位置距离的注意力就是：

$$\begin{aligned} a(m,n) &= \mathrm{Re}\langle \mathbf{f}(\mathbf{q}, m), \mathbf{f}(\mathbf{k}, n) \rangle \\ &= \mathrm{Re}\left[\sum_{j=0}^{d/2-1} (q_{2j} + \mathrm{i}q_{2j+1})(k_{2j} - \mathrm{i}k_{2j+1}) e^{\mathrm{i}(m-n)\theta_j} \right] \\ &= \sum_{j=0}^{d/2-1} (q_{2j}k_{2j} + q_{2j+1}k_{2j+1}) \cos((m-n)\theta_j) + (q_{2j}k_{2j+1} - q_{2j+1}k_{2j}) \sin((m-n)\theta_j) \end{aligned}$$

得到的注意力随m与n之间的距离增大而减小，注意对v矩阵不需要应用RoPE。

xPOS

RoPE能扩展到任意长度，但其外推性能较差：虽然RoPE可以拓展到任意长度，但对于语言建模等生成任务，无法在测试长序列性能时，维持其在训练长度序列上的表现。xPos在旋转的基础上，在旋转角度向量的每个维度上都包含了独特的指数衰减因子，以及blockwise causal attention，让模型忽略相距较远的语义：

$$
\begin{aligned}
\boldsymbol{A}_{t,s} &= \mathrm{Re}\left[\sum_{n=1}^{d/2}\left(\zeta_n^{-t} e^{it\theta_n}\,\tilde{q}_t^{(n)}\right)^{*}\zeta_n^{s} e^{is\theta_n}\,\tilde{k}_s^{(n)}\right] = \sum_{n=1}^{d/2}\mathrm{Re}\left[\tilde{q}_t^{(n)*}\,\tilde{k}_s^{(n)}\,\zeta_n^{s-t} e^{i(s-t)\theta_n}\right] \\
\zeta_n &= \frac{1+\gamma}{2n/d+\gamma},\ \gamma = 0.4, \quad \theta_n = 10000^{-2n/d}
\end{aligned} \tag{16}
$$

参考：[通俗易懂-大模型的关键技术之一：旋转位置编码rope(https://www.bilibili.com/video/BV12x42127Pb/?spm_id_from=333.788&vd_source=4804e44ea59abe1f4c2b7dde30651898)]

[Long-Context LLM综述(https://blog.csdn.net/Cyril_KI/article/details/139573263)]

[Transformer位置编码（基础）(https://zhuanlan.zhihu.com/p/631363482)]

[Transformer升级之路：7、长度外推性与局部注意力(https://zhuanlan.zhihu.com/p/602487515)]