Transformer 模型为什么需要位置编码?

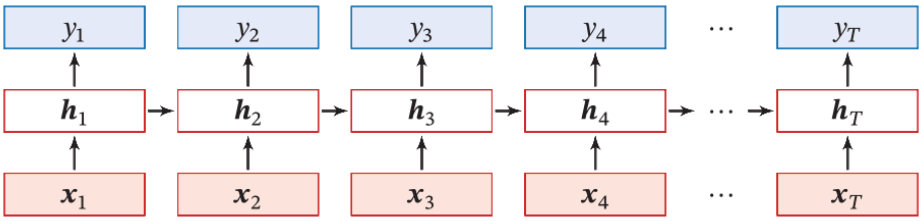
2017 年，Google 在《Attention is All You Need》中提 Transformer 模型。

Transformer 通过自注意力计算得到 Attention 函数(相关性的加权和)。

$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V$$

这种方法不像 RNN 依靠前面的或者后面的序列来累计，所以缺失了位置信息。

$$\boldsymbol{h}_t = f(\boldsymbol{U}\boldsymbol{h}_{t-1} + \boldsymbol{W}\boldsymbol{x}_t + \boldsymbol{b}).$$



按时间展开的循环神经网络

因此，Attention 模块无法获取输入向量的位置顺序，即无法区分不同位置的 Token，这就导致了 Transformer 会产生错误理解。**但实际上，交换句子中两个单词的位置，语义会发生变化。因此需要加入位置编码来标注词的位置信息。**

例子:

杰克送给茉莉一张照片。　　Jack gave a photo to Molly.

茉莉送给杰克一张照片。　　Molly gave a photo to Jack.

Transformer 模型中为什么采用正弦位置编码?

好的位置编码需要满足以下要求：

(1) 位置编码可以描述 Transformers 模型一个 token 在序列中的位置，为每个位

置**分配唯一确定的表示**。

(2)**位置偏移量相同(相对位置不变)的 tokens 的位置编码差值也相同**。即对于任何固定的偏移量k，$\mathrm{p}_{\mathrm{pos+k}}$表示成 $\mathrm{p}_{\mathrm{pos}}$ 的线性函数,而与pos的绝对位置无关。

例子：**我爱中国**

”国“与”中“相对位置为 1，国与中的相关性程度取决于相对位置值 1。而如果这句话前面还有其他字符，那国和中两个字的绝对位置会变化，这个变化不会影响到中华这两个字的相关程度。

(3)模型应该容易泛化到更长的句子，即**模型具有较高的外推能力**。当测试集的输入序列长度大于训练集的输入序列长度时，模型仍较好地适用。

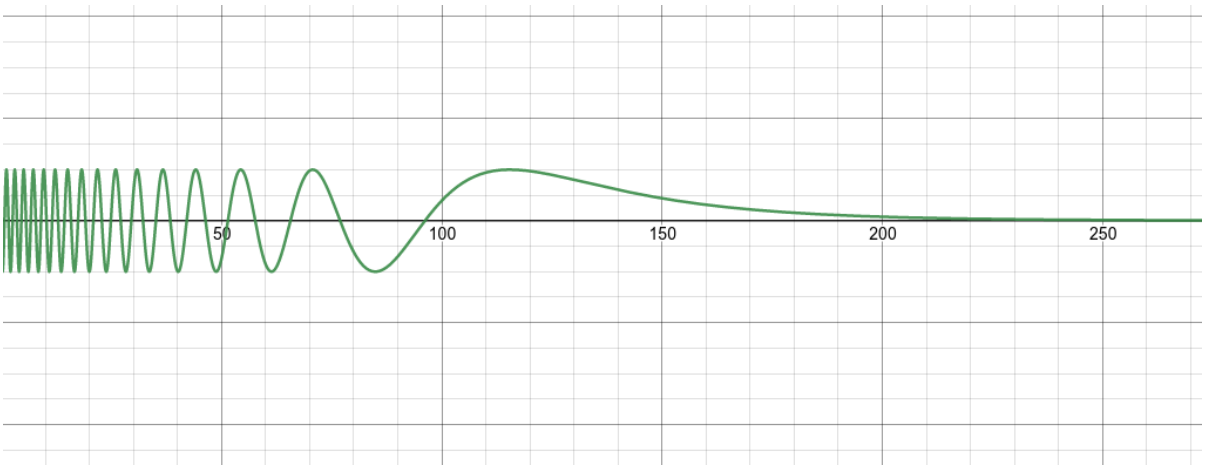
最简单的位置编码是 17 年 Transformer 架构，使用正弦位置编码。计算公式如下：

$$PE_{(pos,2i)} = sin(pos/10000^{2i/d_{\text{model}}})$$
$$PE_{(pos,2i+1)} = cos(pos/10000^{2i/d_{\text{model}}})$$

其中，pos 表示单词在句子中的位置，d 表示位置编码的维度 (与词 Embedding 一样)，2i 表示偶数的维度，2i+1 表示奇数维度 (即 2i≤d, 2i+1≤d)。

Transformer 模型中正弦位置编码的缺点？

(1)对于正弦位置编码，当 embedding 的维度较小时，不同位置的词的位置编码差异明显，当 embedding 维度很大的时候，后面的维度位置编码差异很小。这会导致 **embedding 维度显著上升后，Transformer 模型捕捉词与词之间关系的能力大大下降**。



sin 位置编码函数的图像

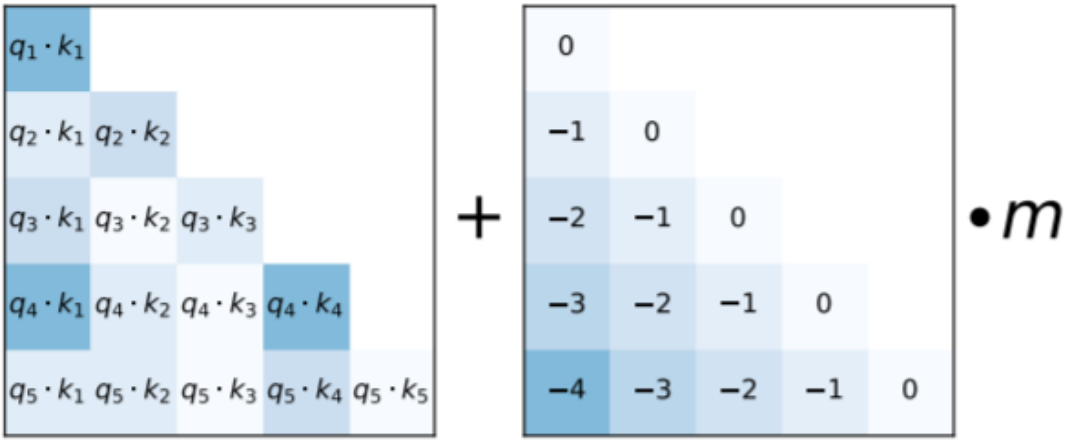
(2)外推能力弱（extrapolation），如果训练集输入序列最长是 512，那么测试集输入序列相差较小。

Sinusoidal Position Embeddings Sinusoidal position embeddings (Vaswani et al., 2017; §3.5) are constant, non-learned vectors that are added to token embeddings on input to the first layer of the transformer. They are frequently used in transformer language modeling (Baevski & Auli, 2018; Lewis et al., 2021) and machine translation (Vaswani et al., 2017; Ott et al., 2018) models. We first consider the unmodified model of Baevski & Auli (2018), which uses sinusoidal position embeddings, and train it on $L = 512$ tokens; we then run inference with it on the validation set on $L + k$ tokens, with k ranging from 0 to 15,000. Figure 1 (left) and the corresponding Table 2 (in the appendix) show that while the model improves perplexity up to $k = 20$, performance stops improving and stays steady from $k = 20$ to $k = 50$ and then begins degrading. Similar results are obtained for a model trained with $L = 1024$ tokens (Figure 1 (right) and Table 3 in the appendix). That model improves for up to $L_{valid} = L + 50$ tokens, after which performance declines.

正弦位置编码怎么改进？

Attention with Linear Biases (ALiBi)位置编码

在 Attention with Linear Biases (ALiBi)位置编码方法中，位置 embedding 并没有加在词 embedding 上，而是乘以比例系数 m 后加在 QK 点积运算后的注意力得分上。



上图中左边是计算出来的其中一个 Head 的 QK 注意力得分矩阵，右边是一个常量矩阵，矩阵中每个位置的值对应 QK 对在序列中的距离，距离越远值越小，该矩阵再乘以常数 m，常数 m 与 Head 相关，其计算公式如下：

$$m_h = \left(\frac{1}{2^{2^{\frac{1}{log_2 H+3}}}}\right)^h$$

其中，H 表示多头自注意力层中的 Head 个数，h 表示 Head 的索引。这种操作，相当于q_i和 k_j相对位置差 1 就加上一个 -1 的偏置*比例系数 m,相当于假设两个 tokens 距离越远那么相互贡献也就越低。

实验设置:

采用 GPT2 模型在 **OpenWebText** 数据集上做实验。实验结果表明，**采用 AliBi 方法可以获得更低的 Perplexity 指标，且训练阶段耗时和内存占用都更低。**

Transformer 可以很好地学习这种位置编码，并与类似大小的等效模型相比执行起来具有竞争力。论文在一个 13 亿参数量的模型上做了实验，**使用 ALiBi 方法，训练阶段输入序列长度是 1024， 推理(测试)阶段输入序列长度是 2048，**

和**训练与预测阶段序列长度都是 2048 且使用位置编码的设置相比，AliBi 具有同样的 perplexity 指标，且速度上快 11% 同时也能节省 11% 的内存。**

和其它模型相比，**ALiBil 模型外推能力较强，即随着推断输入序列长度的增加，ALiBil 模型的 perplexity 没有显著增加，但是 sinusdal,rotary,T5 模型的 perplexity 显著增加。**如下图所示：

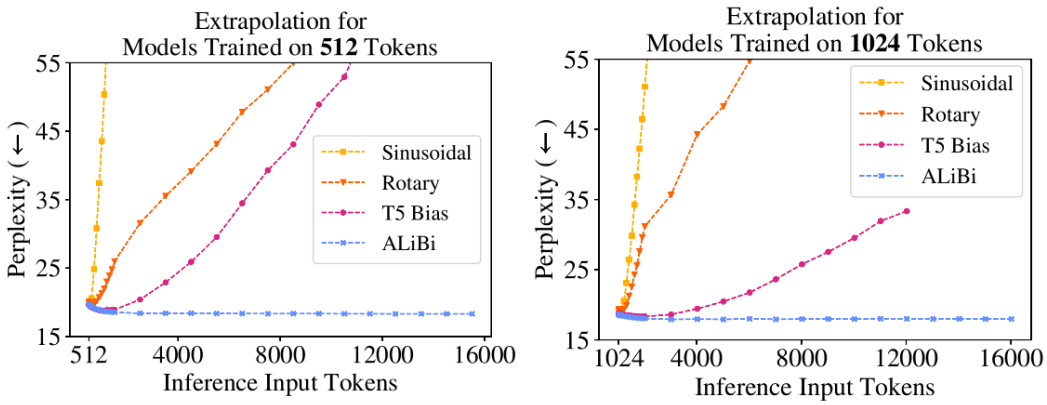


Figure 1: Extrapolation: as the (validation-set's) input sequence gets longer (x-axis), current position methods (sinusoidal, rotary, and T5) show degraded perplexity (y-axis, lower is better), but our method (§3) does not. Models were trained on WikiText-103 with sequences of $L = 512$ (left) or $L = 1{,}024$ (right) tokens. T5 ran out of memory on our 32GB GPU. For more detail on exact perplexities and runtimes, see Tables 2 and 3 in the appendix.

模型使用比训练时更长的序列进行推理(测试)时，虽然之前的几个模型中，随着序列的加长，困惑度急剧增加。但使用 ALiBi 模型，随着序列的加长困惑度不增反而降低。不过**实验中也观察到一个现象：当训练阶段序列长度小于 128 的时候，推理(测试)阶段拓展到更长的序列，AliBi 方法的性能下降严重。**所以建议采用 AliBi 方法训练模型的时候，训练序列长度应大于 256 或者训练阶段序列长度应该不小于推理(测试)阶段序列长度的 0.03125 倍。

总结：

Transformer 位置编码的设置本身有缺点： Transforme 不能利用单词的顺序信息，而这部分信息对于自然语言处理来说非常重要。在单词的表示向量中加入位置编码是一个权宜之计，并没有改变 Transformer 结构上的固有缺陷。