为什么要用这种正弦/余弦函数来表示位置?

一、问题背景：Transformer 没有“顺序感”

传统模型：

- **RNN/LSTM**：通过递归一步步读词，天然知道“前后顺序”；
- **CNN**：通过卷积核的局部连接，也隐含了一定的顺序结构。

但 Transformer 的 **Self-Attention** 是“全局的”：

> 它在一个步骤里就能看到整个句子，没有顺序输入的结构。

所以，我们必须显式地告诉模型：

> “这是第 1 个词、这是第 2 个词……这些词之间的距离是多少。”

这就是 **位置编码（Positional Encoding）** 存在的原因。

二、设计目标：我们想要的位置表示要满足什么特性?

Vaswani 等人希望位置编码具有以下几个性质：

1. **唯一性**
   每个位置 pos 对应一个唯一的向量。
2. **可加性**
   可以直接和词向量相加（不改变维度）。
3. **可推广性**
   能对**比训练时更长的序列**也有效。
4. **能表达相对位置信息**
   模型不仅知道“这是第 7 个词”，还可以通过内积等运算**学到两个位置的相对距离。**

三、为什么用正弦 + 余弦?

论文的设计是这样的：

$$PE_{(pos,2i)} = \sin\left(\frac{pos}{10000^{2i/d_{model}}}\right)$$

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

这是一个**连续的、周期性的、平滑变化**的编码方式，具有几个重要优点 👇

---

1. 不依赖学习，能“泛化到更长序列”

有些模型也用“可学习的位置向量”（learned positional embeddings），
但那种做法只能记住训练时看到的序列长度，
对更长的句子就无能为力。

而 sin/cos 是**解析函数**，可以外推：
即使你输入 `pos = 10000`，也能得到一个合理的编码。

---

2. 能通过线性变换表达“相对位置”

这是这个设计最优雅的地方。

设 (k) 是一个偏移量：

$$\sin(a+b) = \sin(a)\cos(b) + \cos(a)\sin(b)$$

$$\cos(a+b) = \cos(a)\cos(b) - \sin(a)\sin(b)$$

也就是说：

> (PE(pos + k)) 可以用 (PE(pos)) 的线性组合表示。

这意味着模型在注意力机制中，可以通过简单的线性操作“推断”相对距离 (k)！
不需要显式告诉它两个词相隔几个位置。

为什么正弦/余弦形式的位置编码能让模型“线性地”表示相对位置信息?

1、回顾公式

位置编码定义为：

$$PE(pos) = \left[\sin(\omega_0 \cdot pos)\ \cos(\omega_0 \cdot pos)\ \sin(\omega_1 \cdot pos)\ \cos(\omega_1 \cdot pos)\ \vdots\right]$$

其中每一对 $\sin, \cos$ 对应一个频率：

$$\omega_i = \frac{1}{10000^{2i/d_{\text{model}}}}$$

2、关键的三角恒等式

对每个频率分量 ω_i，
假设我们想表达从位置 pos 到 $pos + k$ 的“相对位移 k”：

$$\sin(\omega_i(pos+k)) = \sin(\omega_i pos)\cos(\omega_i k) + \cos(\omega_i pos)\sin(\omega_i k)$$

$$\cos(\omega_i(pos+k)) = \cos(\omega_i pos)\cos(\omega_i k) - \sin(\omega_i pos)\sin(\omega_i k)$$

3、用矩阵形式写出来

对每个频率 ω_i，我们可以写成一个二维线性变换：

$$\begin{bmatrix}\sin(\omega_i(pos+k)) \\ \cos(\omega_i(pos+k))\end{bmatrix} = \underbrace{\begin{bmatrix}\cos(\omega_i k) & \sin(\omega_i k) \\ -\sin(\omega_i k) & \cos(\omega_i k)\end{bmatrix}}_{\text{线性变换矩阵 } R_i(k)} \begin{bmatrix}\sin(\omega_i pos) \\ \cos(\omega_i pos)\end{bmatrix}.$$

这说明：

> 对于每个频率维度，**位移** k 对应一个**旋转矩阵** $R_i(k)$。

也就是说：

- 从位置 `pos` 到位置 `pos + k` 等价于在 $(\sin, \cos)$ 二维平面上旋转一个角度 $\omega_i k$。

4、几何直观理解

每对 $(\sin, \cos)$ 相当于二维平面上的一个单位向量。
随着位置 `pos` 增大，这个向量在单位圆上“转动”。

- 每个维度对应该频率的旋转速度；
- 位移 k 就是额外转过的角度。

所以：

- 相对位置差 k 对应一个旋转角度；
- “旋转”是线性变换；
- 因此 Transformer 可以通过线性操作（比如注意力的加权和）学会相对位置信息！

5、直观的数值例子

设只有一个频率 $\omega = 0.5$。

位置 3 的编码：

$$PE(3) = \begin{bmatrix} \sin(0.5 \times 3) \\ \cos(0.5 \times 3) \end{bmatrix} = \begin{bmatrix} \sin(1.5) \\ \cos(1.5) \end{bmatrix} \approx \begin{bmatrix} 0.99749499 \\ 0.07073720 \end{bmatrix}.$$

位置 4 的编码：

$$PE(4) = \begin{bmatrix} \sin(0.5 \times 4) \\ \cos(0.5 \times 4) \end{bmatrix} = \begin{bmatrix} \sin(2.0) \\ \cos(2.0) \end{bmatrix} \approx \begin{bmatrix} 0.90929743 \\ -0.41614684 \end{bmatrix}.$$

位移 $k = 1$ 对应的线性变换矩阵为：

$$R(1) = \begin{bmatrix} \cos(0.5) & \sin(0.5) \\ -\sin(0.5) & \cos(0.5) \end{bmatrix} \approx \begin{bmatrix} 0.87758256 & 0.47942554 \\ -0.47942554 & 0.87758256 \end{bmatrix}.$$

验证一下：

$$R(1) \cdot PE(3) \approx \begin{bmatrix} 0.87758256 & 0.47942554 \\ -0.47942554 & 0.87758256 \end{bmatrix} \begin{bmatrix} 0.99749499 \\ 0.07073720 \end{bmatrix} \approx \begin{bmatrix} 0.90929743 \\ -0.41614684 \end{bmatrix} = PE(4).$$

完全吻合！

这说明：

> 从位置 3 到位置 4，只是一个线性旋转操作。

6、为什么这很重要?

这意味着 Transformer 的注意力机制可以**直接从绝对位置编码中“线性推断”相对位置差。**
它不需要显式存储所有相对距离，只要学到这种线性模式即可。

这正是 sin/cos 编码的一个重大优势：**相对位置 → 线性可表示。**

---

3. 多频率编码 = 多尺度位置感知

由于不同维度使用不同频率的正弦波：

- 高频维度 → 捕捉局部相邻词之间的微小位置差；
- 低频维度 → 捕捉全局的长程位置信息。

这种多尺度特性让模型能同时感知：

> “这是在第几个词” + “它距离句首/句尾多远”。

---

4. 向量维度与词向量对齐

sin/cos 位置编码的维度与 embedding 相同 ((d_{model}))，
可以直接与词向量 **相加**：

$$X_{input} = X_{word} + X_{position}$$

这样模型在输入层就同时拥有：

- 词义（word embedding）
- 位置信息（positional encoding）

四、与其他位置编码方法的对比

方法	特点	优缺点
固定 sin/cos	不可训练，可外推	泛化好，但灵活性有限
可学习 embedding	每个位置一个参数	学习力强，但不能外推
相对位置编码（Transformer-XL, T5)	编码距离关系	对长序列建模更强，但复杂度更高

五、总结一句话

> Transformer 用正弦/余弦位置编码，是为了让模型在没有顺序结构的前提下，
> 通过一种 **连续、可外推、能表达相对距离的方式** 给每个位置一个唯一的、可学习的空间表示。