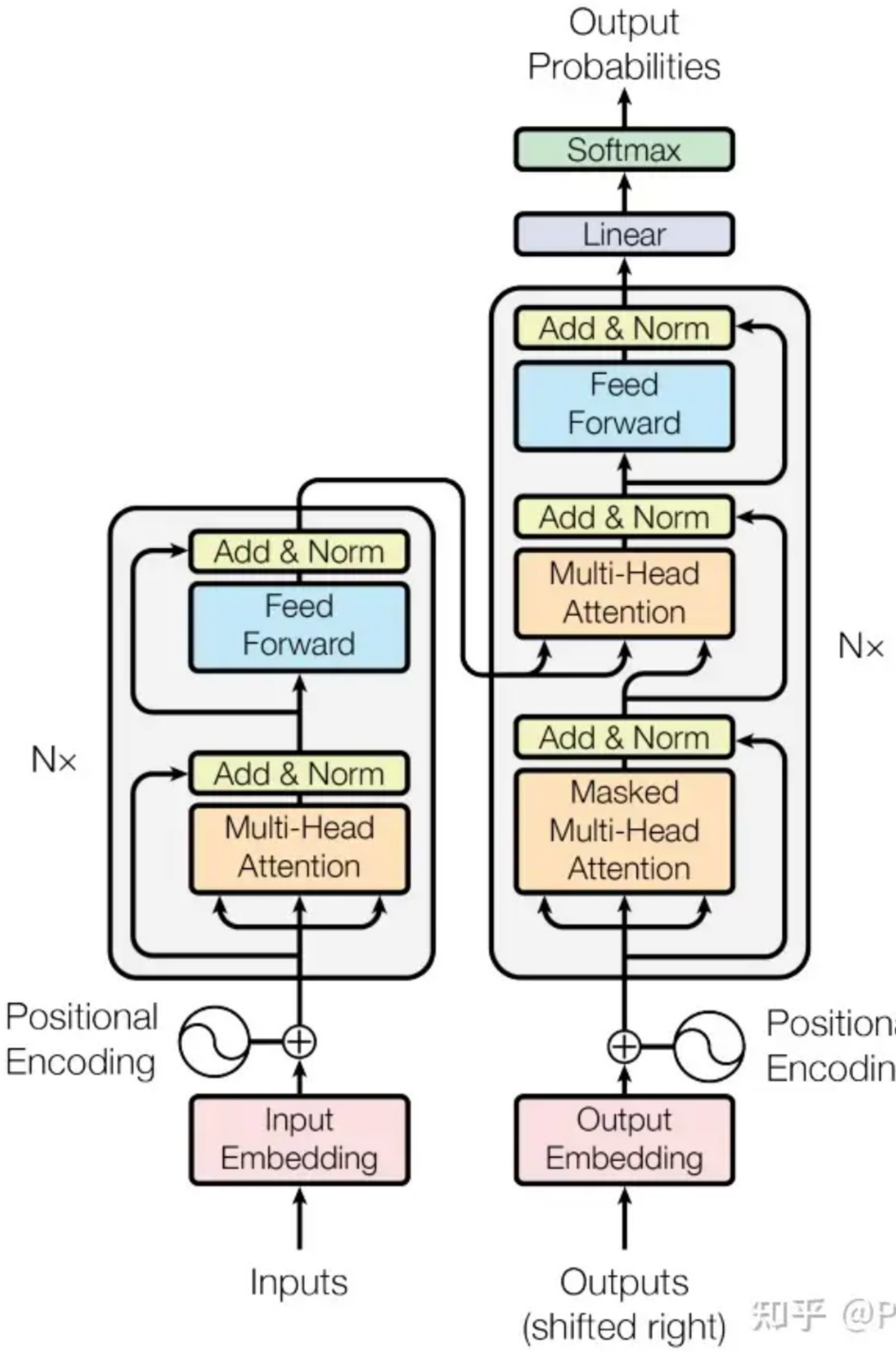
论文地址：arxiv.org/pdf/1706.0376...

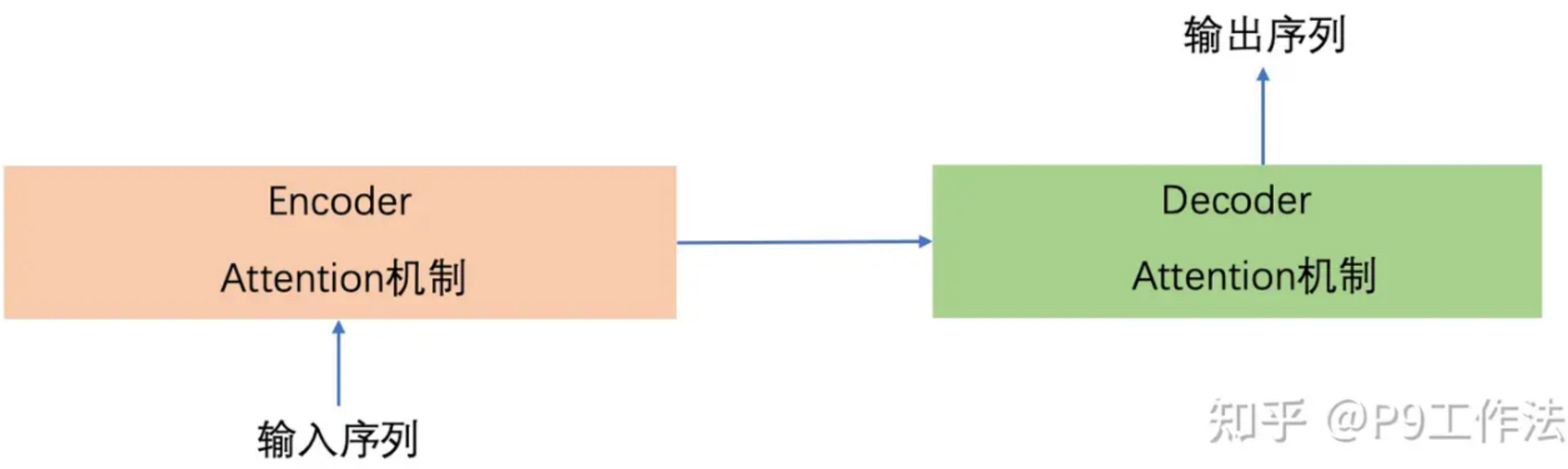
模型结构

这是从论文中截取的transformer模型✦的结构图，看起来是有点复杂的，按照架构思维再进行简化。



从左右看

把transformer结构按照左右视角来看，可以简化为下图的 encoder与decoder结构，只是这里的encoder和decoder是使用了attention机制。



encoder（编码器）负责将输入的信息编码到一个空间，然后传递给decoder✦（解码器），解码器将其解码为目标内容。这个结构是极其有魅力，用这个结构可以做很多任务。比如：

1、输入为中文，输出为英文，这就是文本翻译。

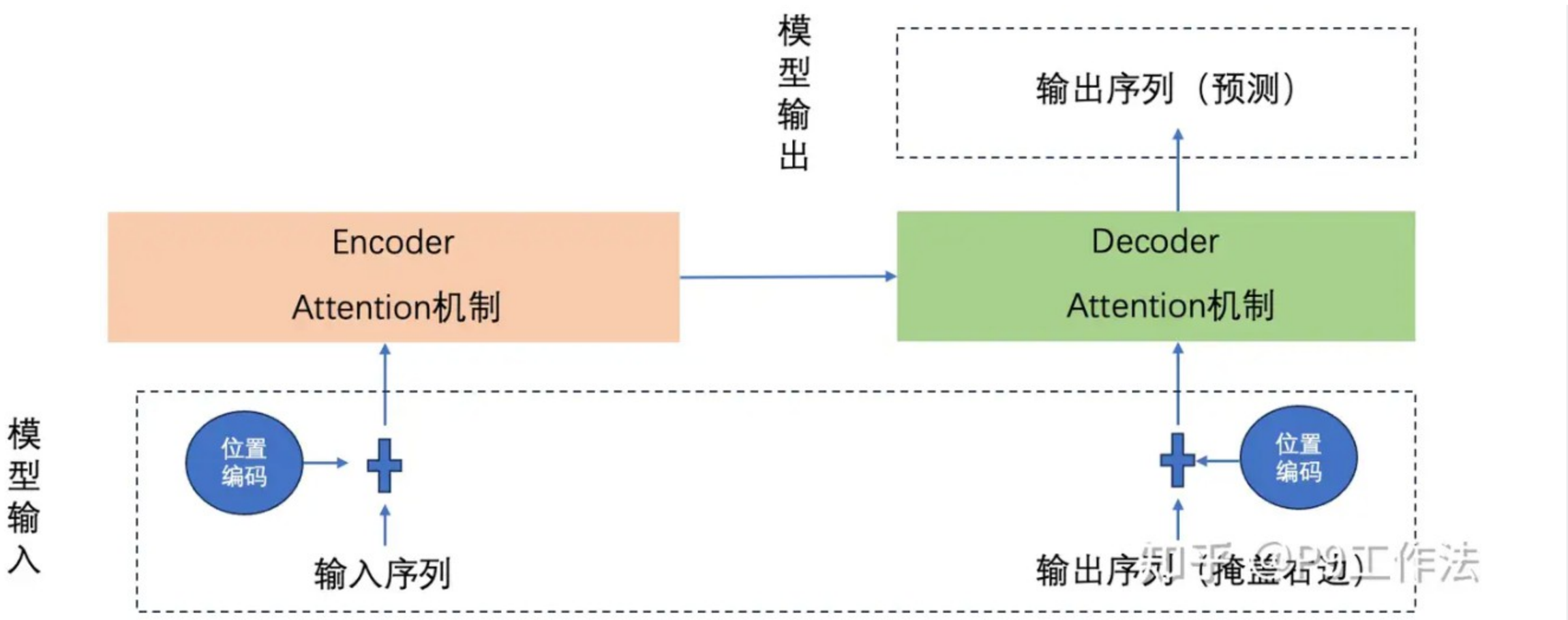
2、输入为语音，输出为文本，这就是语音识别。

3、输入为文字，输出为图片，这就是图片生成。

这个结构也可以说是seq2seq，输入一个序列得到一个序列，输入序列和输出序列都不拘泥于长度。

从上下看

把transformer结构按照上下视角来看，可以简化为下图的输入和输出结构。



可以看到输入其实是encoder，decoder都有，而输出只是在decoder部分。注意这里的表述是模型的输入，就是喂给模型的数据。

以文本翻译为例（“我爱你”翻译为“I love you ”），给encoder输入的就是“我爱你”的embedding向量✦。但是在训练的时候，我们是要把 “我爱你” ----> “I love you ” 这个数据都给模型，那么喂给decoder的就会是“I love you ”的embedding向量。让decoder产出预测的输出序列。预测的输出序列和给定的输出训练去做损失函数，由此去训练模型。

其次模型输入中有一个位置编码✦信息，这个的含义就是让输入的信息包含了位置信息。否则你输入的是“我爱你“的向量，模型有可能理解错为“你爱我”。

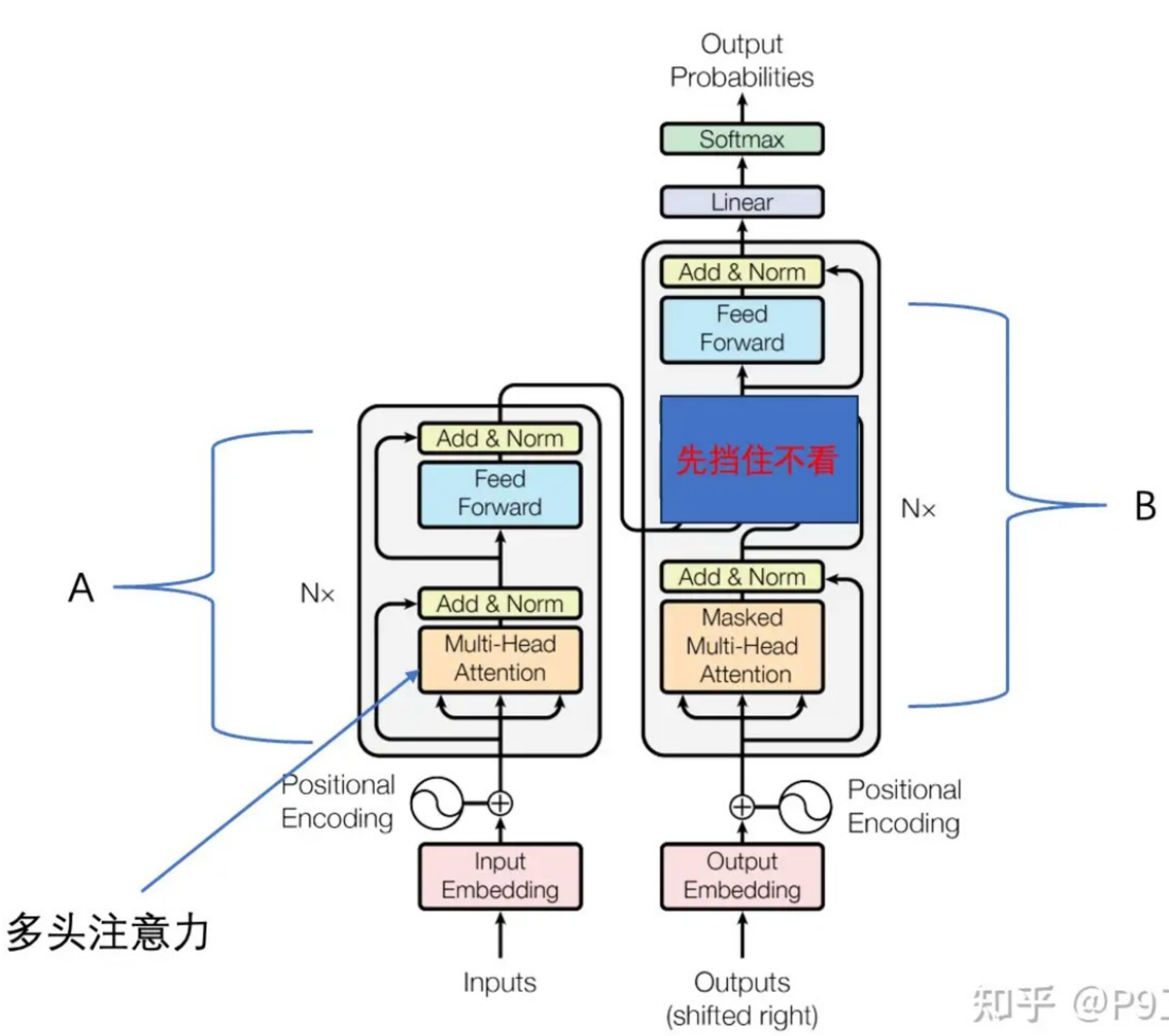
从对称看

我们再来看这个模型的对称性，首先是左右两边来看（A与B），如果把右边decoder的中间部分挡住，就会发现两边极具对称性。

A中的结构依次是：
多头注意力模块，残差模块，归一化模块，前馈神经网络✦，残差模块，归一化模块，这一串叫一个block。

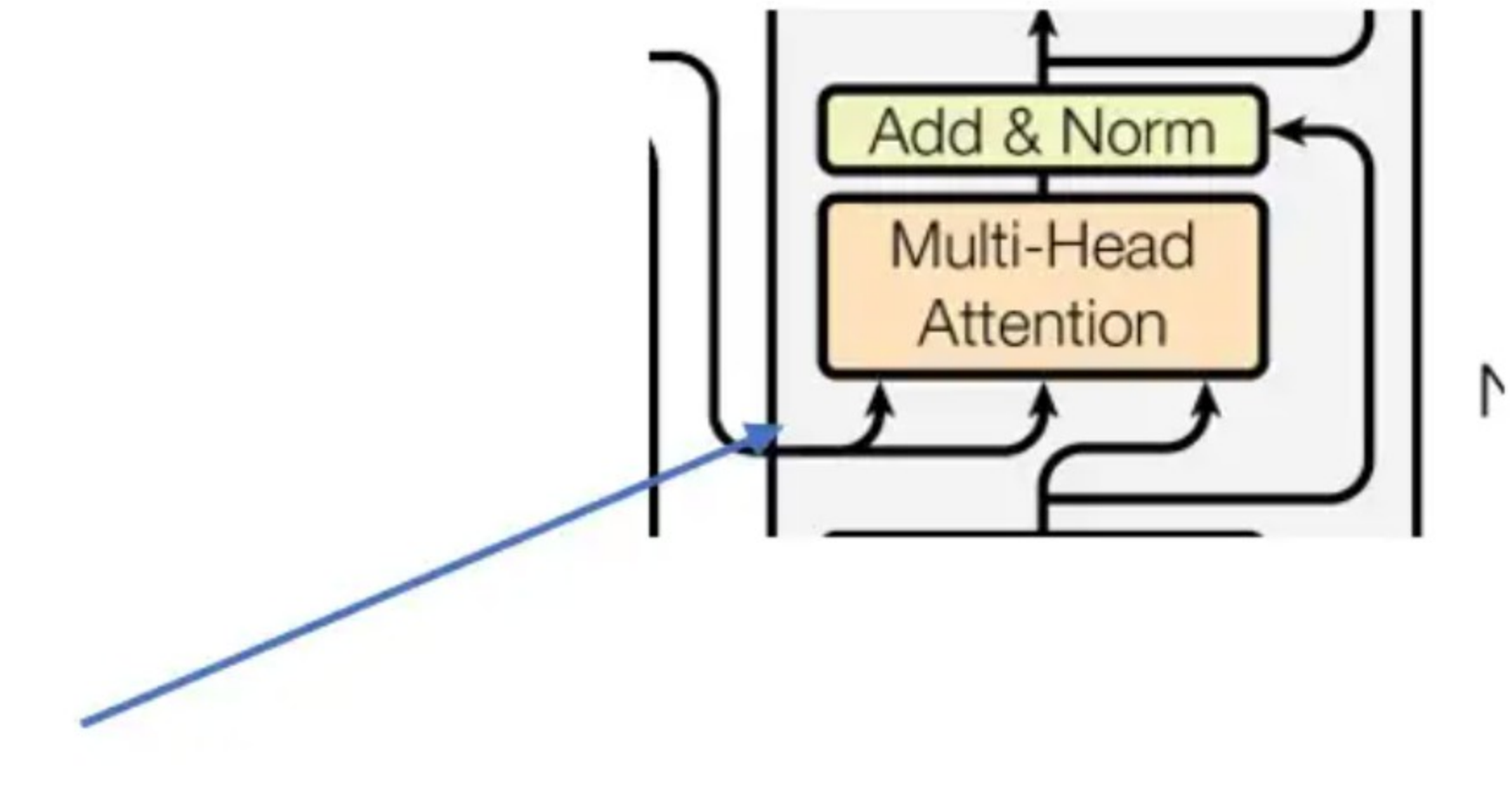
而A中有这样的6个block。

而多头注意力✦模板就是多个注意力模块。



而B的结构与A的架构高度类似，差异的点就是多头注意力变成了masked 多头注意力机制✦。

而挡住的那部分再单独拎出来，大的结构也是多头注意力模块，残差模块，归一化层。唯一多的就是下面的连接线，这叫交叉注意力机制✦。



小结

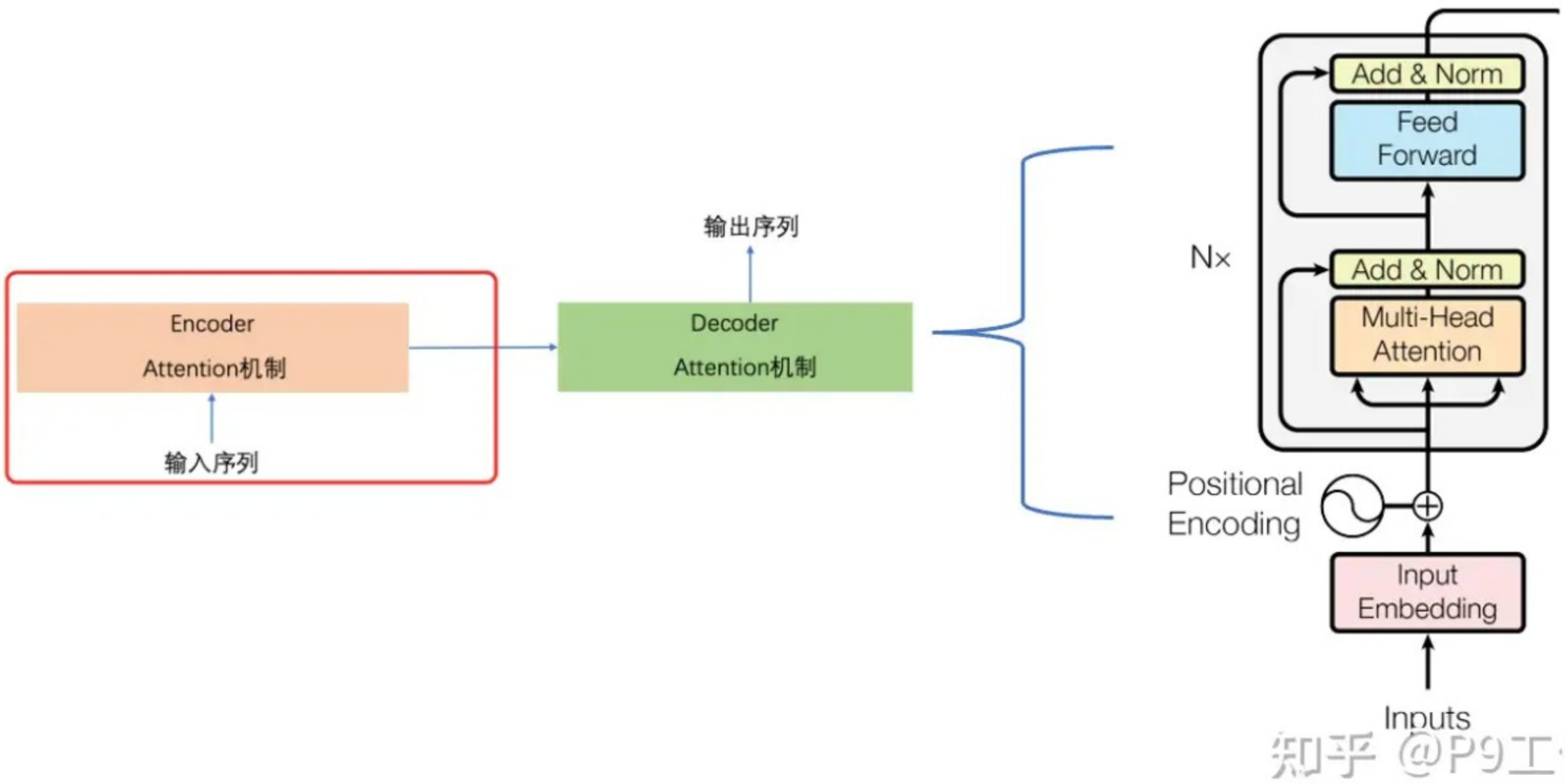
整个transformer模型，可以拆解为如下组件去理解：

模块	含义
位置编码模块	讲序列的位置信息编入到数据中。
多头注意力机制模块	对序列提取有效的信息。
残差模块	将信息跨层连接，直接传递信息。
归一化层	会对一个样本的所有特征进行归一化处理，有助于加速训练过程并提高模型的稳定性。
前馈神经网络	普通的全连接神经网络✦。
掩码注意力机制	训练时不能让模型看到后面的内容，比如翻译“I”时，不能把love you 给模型看到。
交叉注意力	就是让decoder去提取什么encoder的什么信息对它预测最用。
Linear	高维向量转换成一个与词汇表大小相匹配的向量。
SoftMax	将输出的信息转换成概率。

实例计算

Encoder

把encoder与transformer的对应起来，如下图所示：



输入序列处理

1、首先将输入序列“我爱你”进行词嵌入，得到词向量

$$\begin{bmatrix} 0.1 & 0.2 & 0.3 \\ 0.2 & 0.4 & 0.6 \\ 0.4 & 0.5 & 0.7 \end{bmatrix}$$

2、用论文的公式，对词嵌入向量进行位置编码。分为两步，计算得到位置编码信息，然后与词嵌入向量相加。

这两个过程如下图所示：

$$\begin{bmatrix} 0.1 & 1.2 & 0.3 \\ 1.0414709848078965 & 0.969650086931313 & 1.401961795214785 1 \\ 1.3092974268256818 & 0.1491048058597337 4 & 1.6581443762382828 \end{bmatrix}$$

↑ 嵌入了位置信息的词向量

位置编码

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

论文公式

模型输入:
我爱你

$$\begin{bmatrix} 0.1 & 0.2 & 0.3 \\ 0.2 & 0.4 & 0.6 \\ 0.4 & 0.5 & 0.7 \end{bmatrix}$$

我爱你的词向量

注意力机制计算

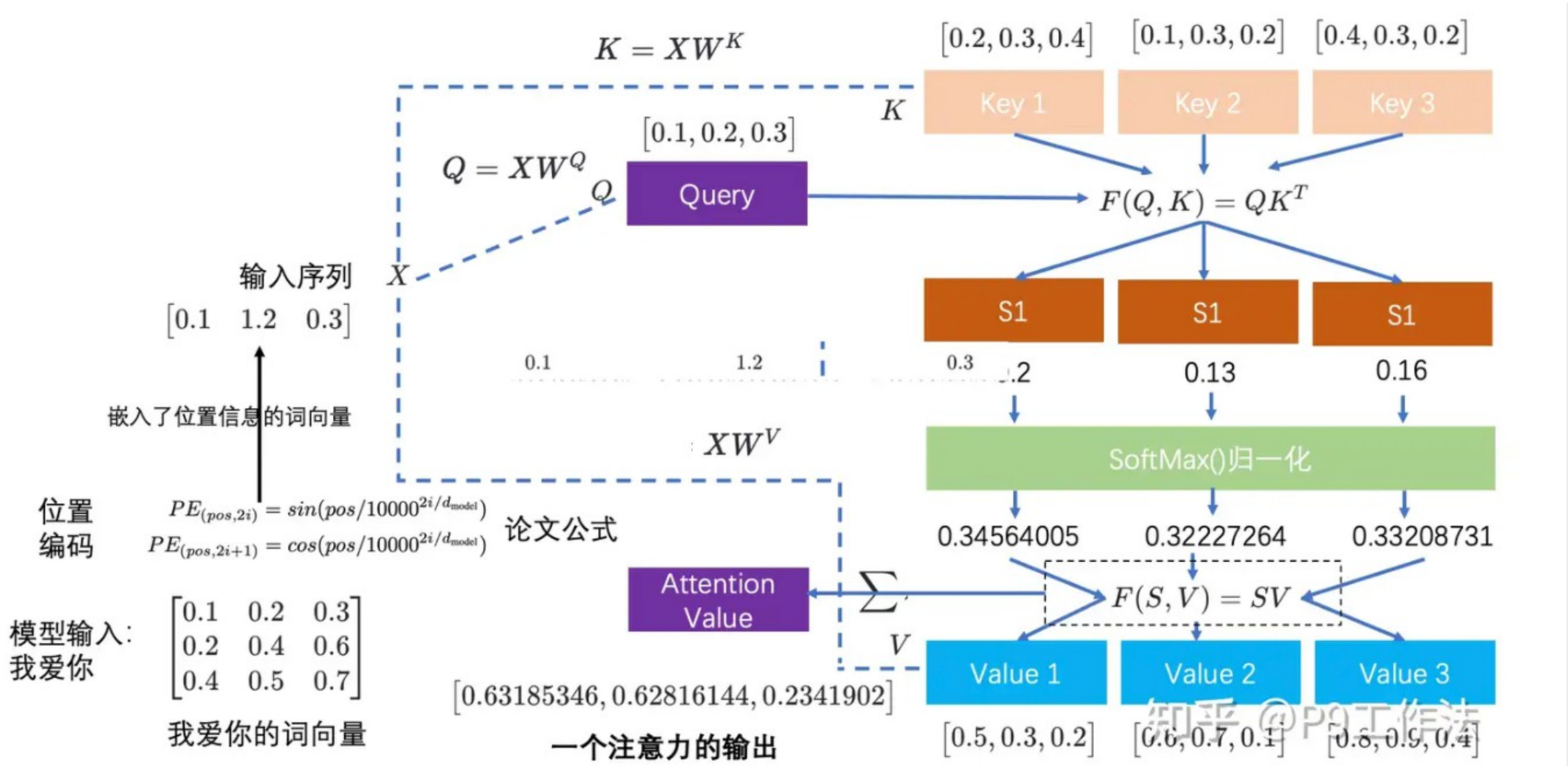
1、将嵌入了位置信息的向量去做注意力机制的计算，这里显然是自注意力机制✦的计算。

首先是“我”，与“爱”，“你”去做自注意力机制的计算。

然后是“爱”，与“我”，“你”去做自注意力机制的计算。

最后是“你”，与“我”，“爱”去自注意力机制的计算。

每一次自注意力机制的计算如下图所示（如先计算“我”）



得到的attention 分数为：

$$[0.63185346, 0.62816144, 0.2341902]$$

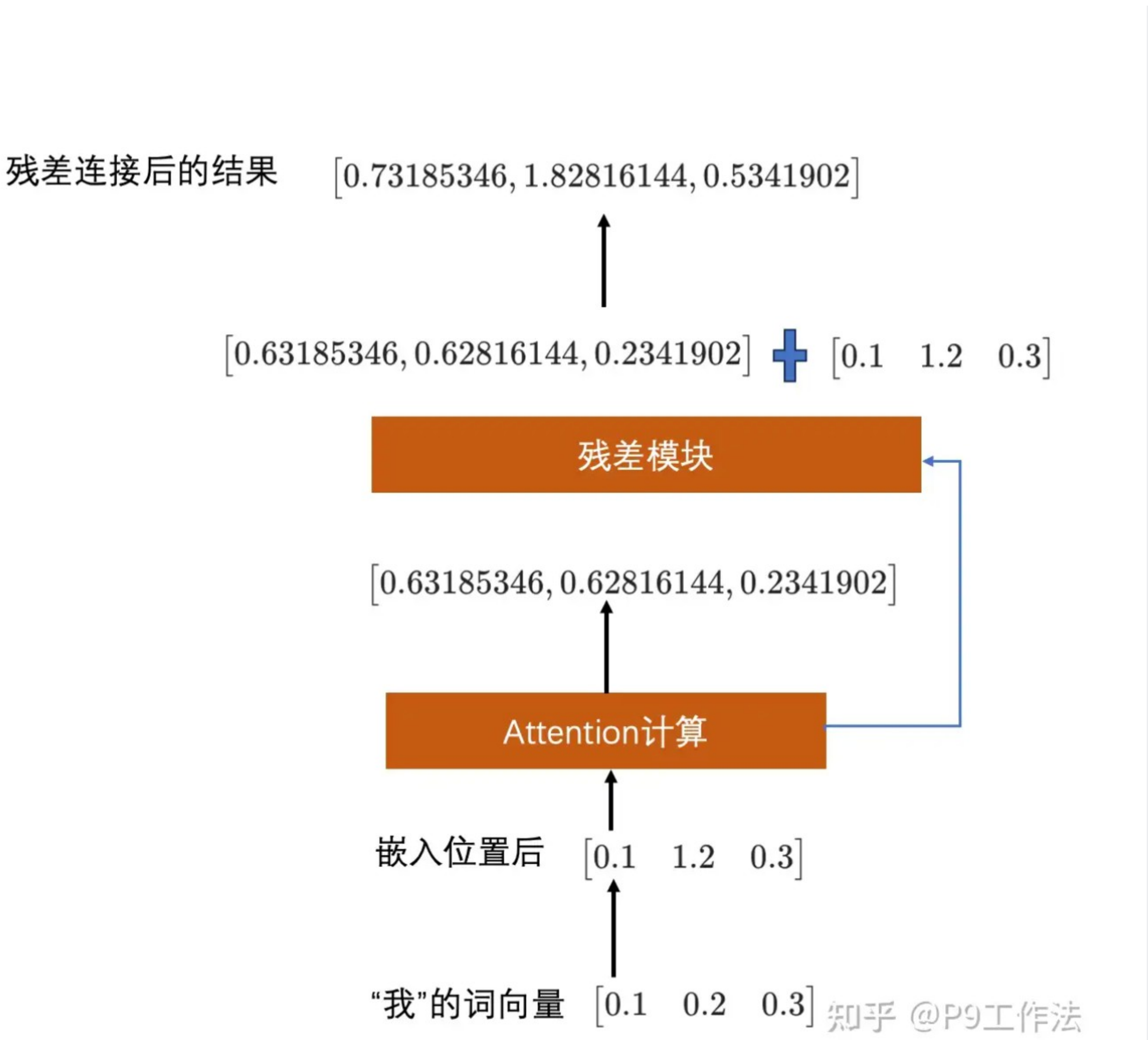
2、而多头注意力机制就是有多个这样的这样Q,K,V矩阵，去计算多次，在transformer中是计算8次。

残差链接

1、残差连接✦其实就是跨层连接，不管中间做了多少计算，都把原始信息传递下去。

这有点像团队中的跨级管理，TL A给下级TL B派了一些活，TL B各种拆解折腾，然后交给下面的一线员工C去执行，但TL A不放心，还是要把原始活带给C，然C既知道B的决策，也知道A的决策，这样能够更好去执行任务。

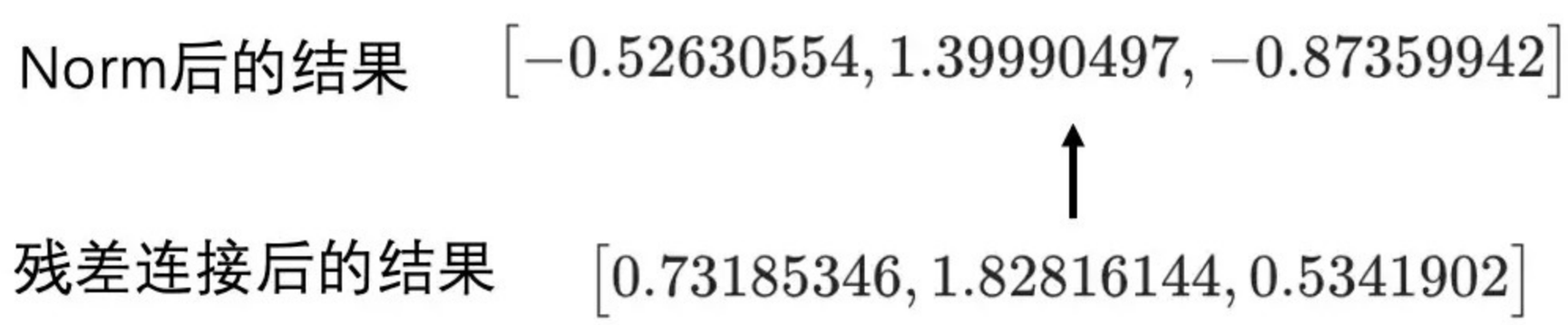
2、而多头注意力机制相当于，TL A为了保险起见，把活不仅给到了TL B，还给了TL C,D,E等，让他们都去拆解折腾，但最终执行时候汇总起来一份给到C去执行。

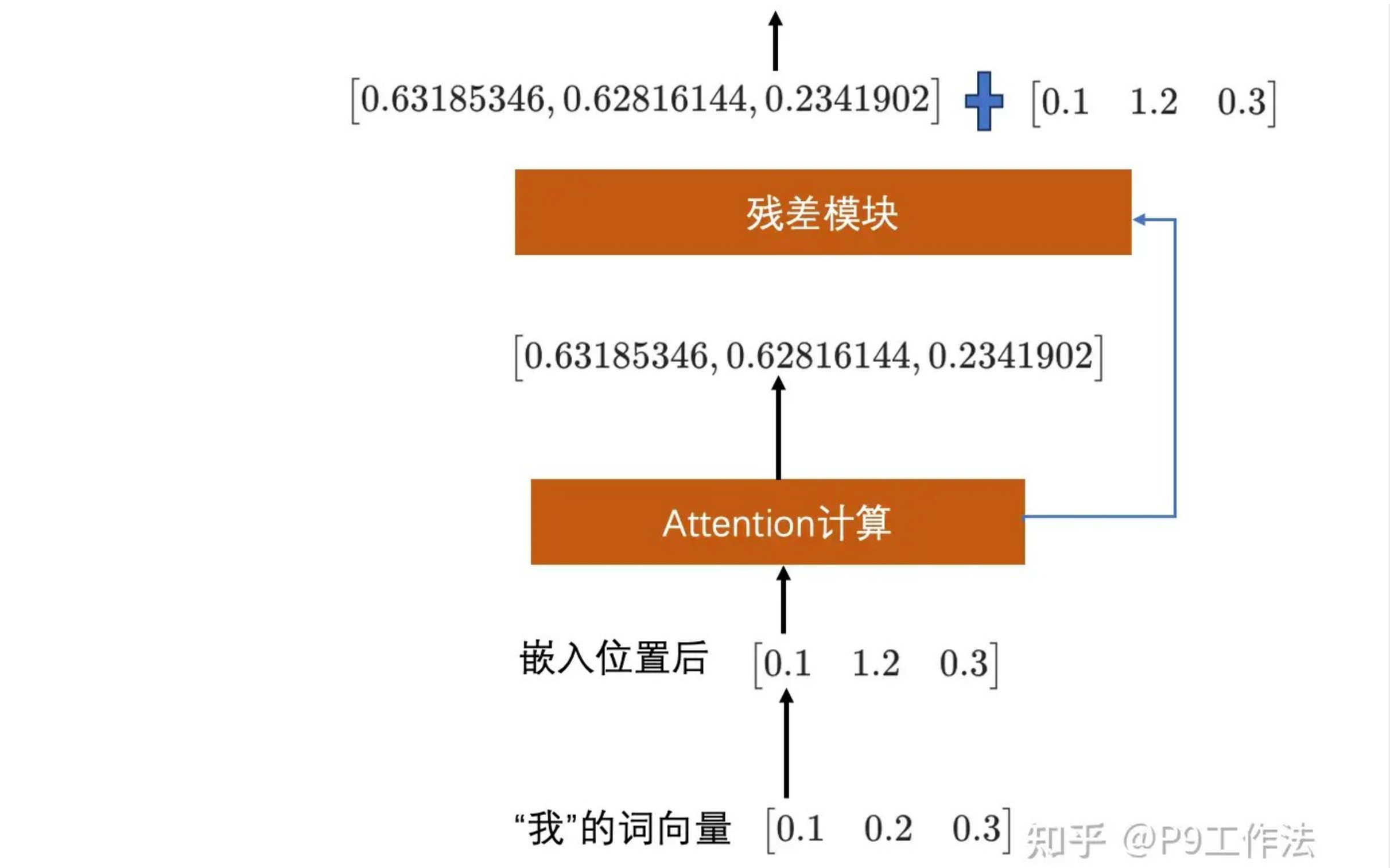


层标准化

一种标准化技术，简单来说就是值减去均值后除以方差。目的是减少梯度，加快训练。公式如下：

$$x_i' = \frac{x_i - m}{\sigma}$$



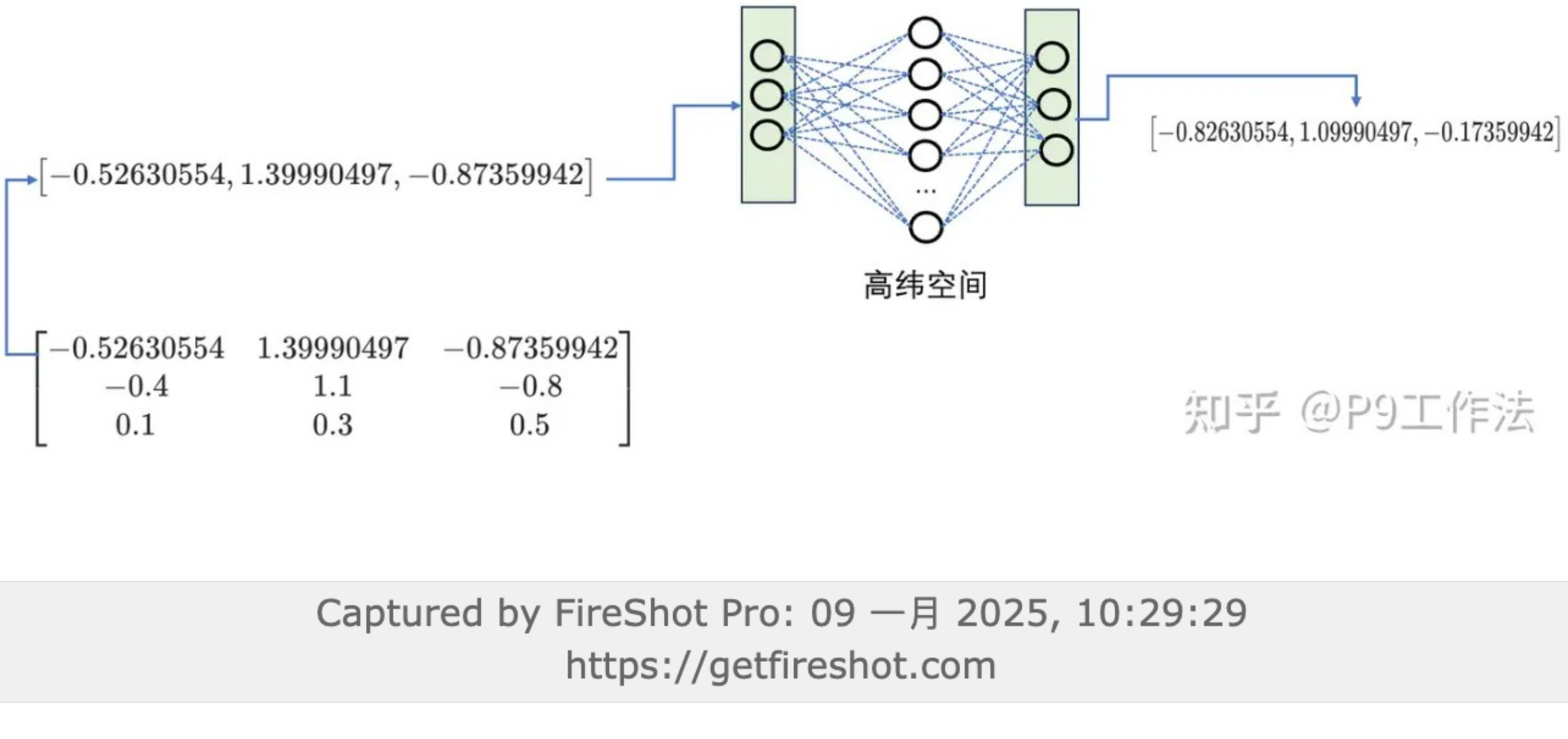


前馈神经网络

将“我”“爱”“你”句话的三个词都用自注意机制，残差连接，层标准化后得到一个完整的向量：

1、将“我”的词向量丢给前馈神经网络，前馈神经网络的维度应该是要高于输入的维度。

2、输入是三维，输出还是三维。



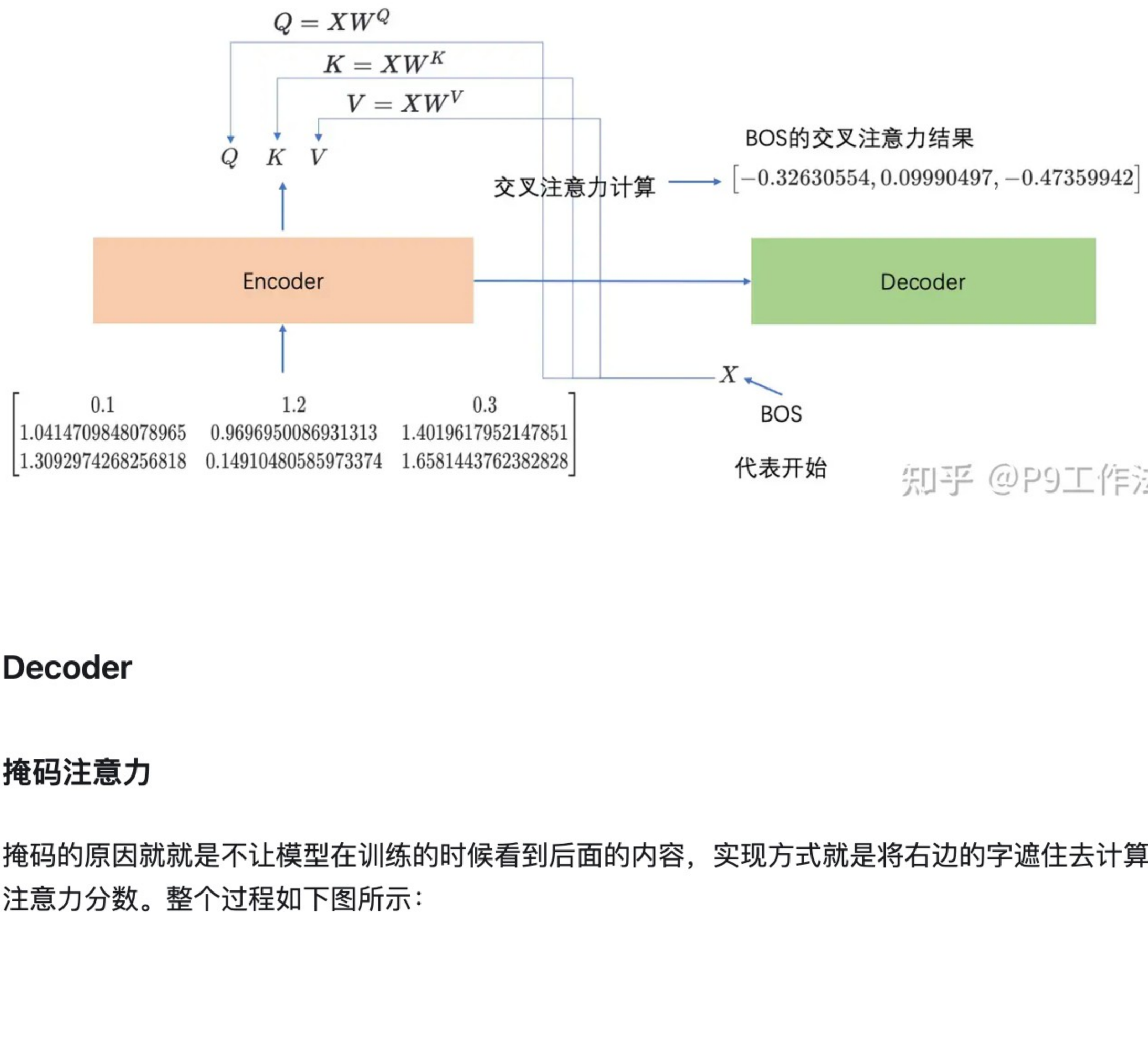
3、前馈神经网络输出后，继续经过层标准化和残差，就不再赘述了。

交叉注意力

1、交叉注意力机制是解码器与编码器之间的桥梁，解码器通过它把特征信息传递给解码器。

2、解码器第一个是开始符号，BOS。按照词嵌入和位置编码后，得到一个向量。

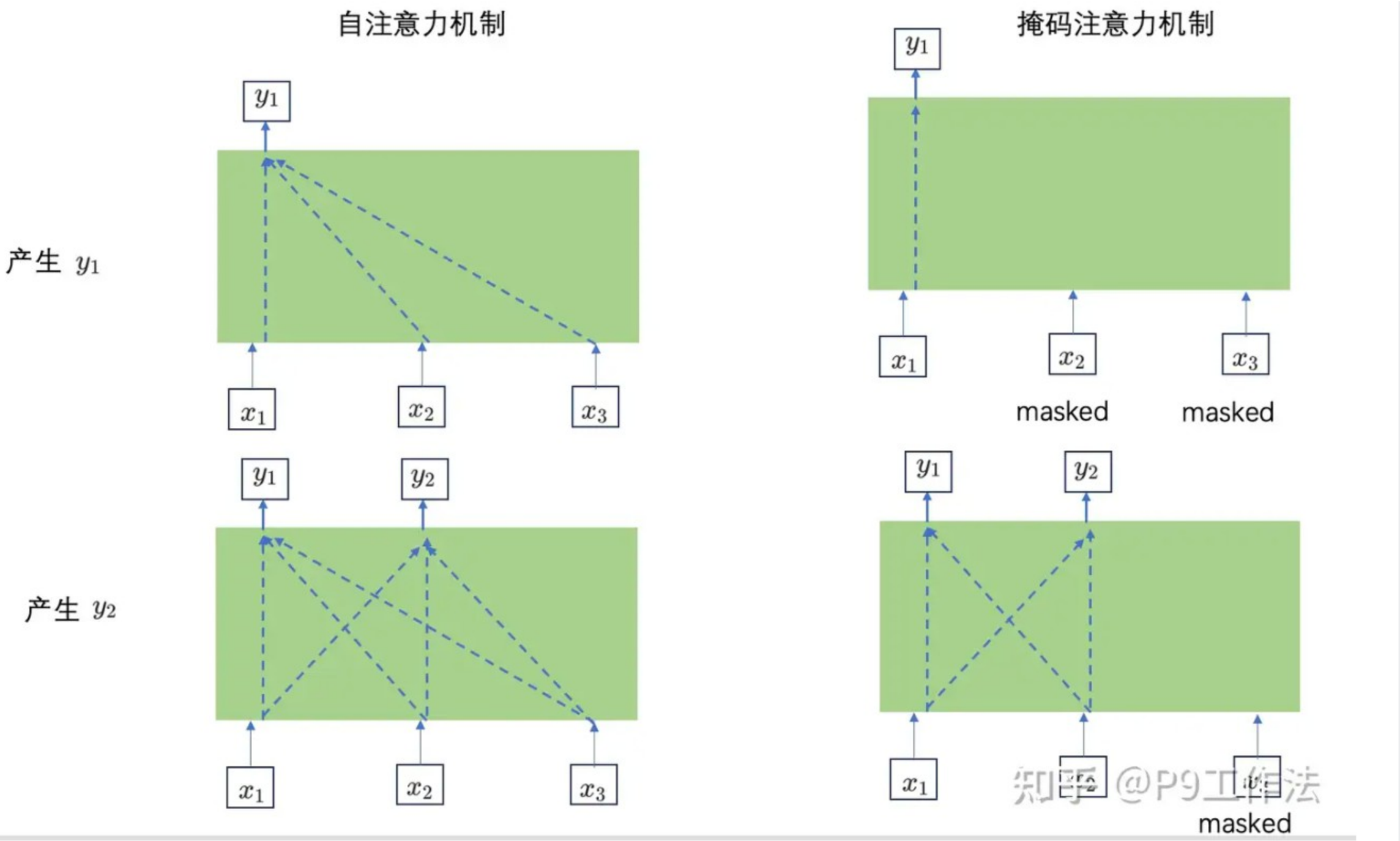
3、该向量与编码器的K,V矩阵求注意力，并得到解码的 W^Q, W^K, W^V 。



Decoder

掩码注意力

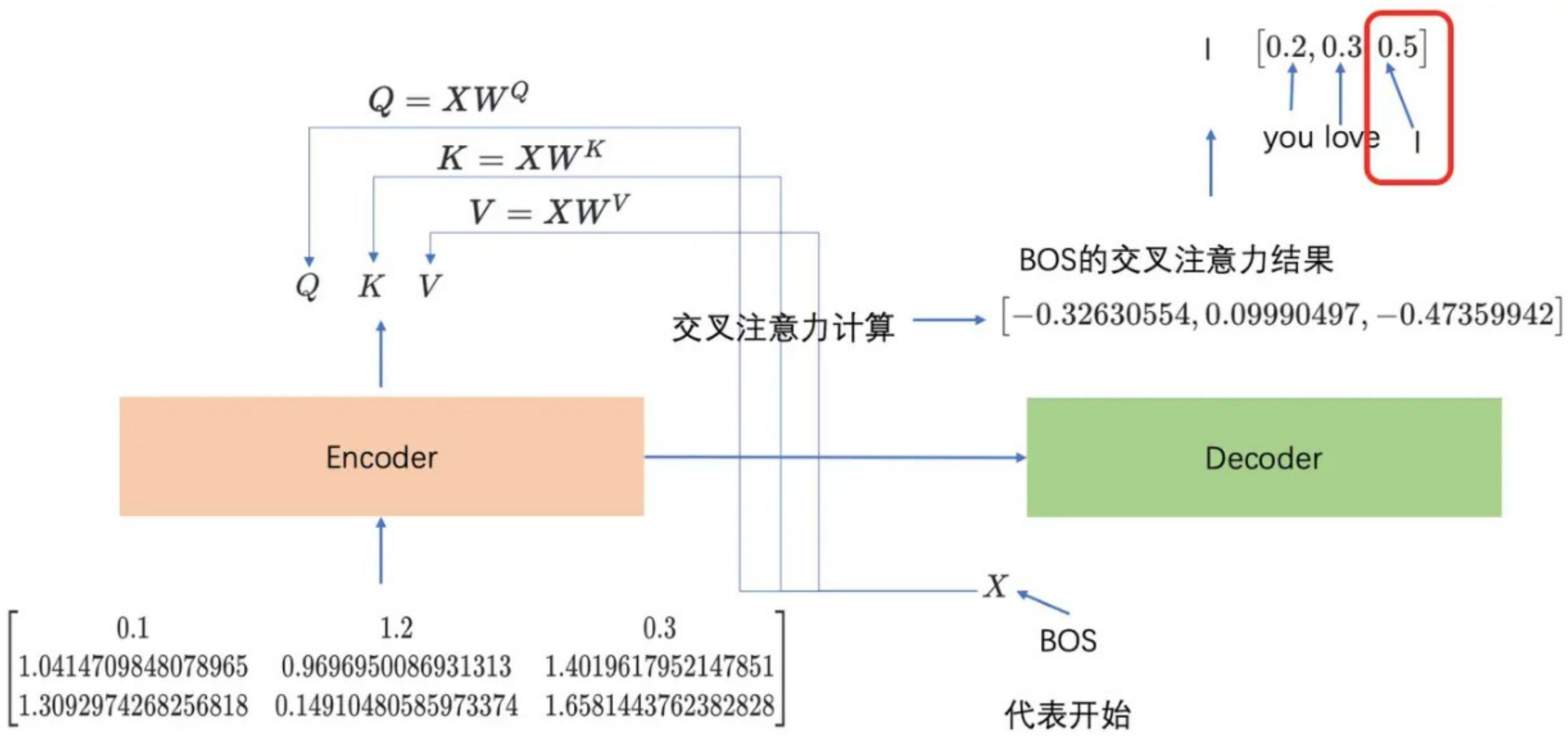
掩码的原因就就是不让模型在训练的时候看到后面的内容，实现方式就是将右边的字遮住去计算自注意力分数。整个过程如下图所示：



Linear&SoftMax

这里的线性层其实是一个全连接层✦。线性层接收前面的隐状态作为输入，并通过一个简单的线性变换（即权重矩阵乘法加上偏置），将每个隐状态映射到一个长度等于词汇表大小的新向量。这个新向量中的每个元素对应于词汇表中的一个单词或标记，并且这个值被称为logit。

假设词汇表里面就三个单词“I”，“love”，“you”。



得到的logits向量接着会传递给Softmax函数（或其他类似的归一化指数函数）。Softmax函数会将logits转换成一个概率分布，其中每个元素代表相应词汇项被选为下一个输出词的概率。这样做的结果是一个由0到1之间的数值组成的向量，所有的数值加起来等于1。

也就是实现了预测单词的功能，上面就预测出来了“I”。

总结

在transformer架构中可以看到有9个组件，编码器有6个block，解码器有6个block，多头注意力有8个，位置编码设计等。这些都是精心设计不能更改的吗？其实不是，神经网络中没有一个结构是绝对的真理，减少一些或者增加一些都是可以的，最终都是看效果，用效果去反调这些结构（增加或减少组件）。

这也是为什么要用架构思维来理解AI的原因，架构就是看模块与模块之间的关系，这在AI领域也是同样适用。