**《Attention Is All You Need》**

**摘要：**

1. **本文的背景：**2017年之前，主流序列转换模型都是基于RNN or CNN的，而表现较为出色的RNN架构在解码器上不能有效进行并行计算，导致绝大多数模型有着算法效率低下的问题。
2. **所提出的解决方法：**一个新的简单网络结构——**Transformer**，其核心结构仅仅是基于注意力机制，而不完全是RNN 或CNN。
3. **模型表现：**在WMT 2014英语到德语翻译任务以及英语到法语翻译任务上，分别取得了28.4BLEU和41.8BLEU的分数，是当时该项目的最好结果。
4. **其他研究分析：**在大型、有限的训练数据中，通过将Transformer成功应用于英语句法解析，表明了其可以很好地适用于其他任务，即具有一定的泛用性。

**问题背景：**

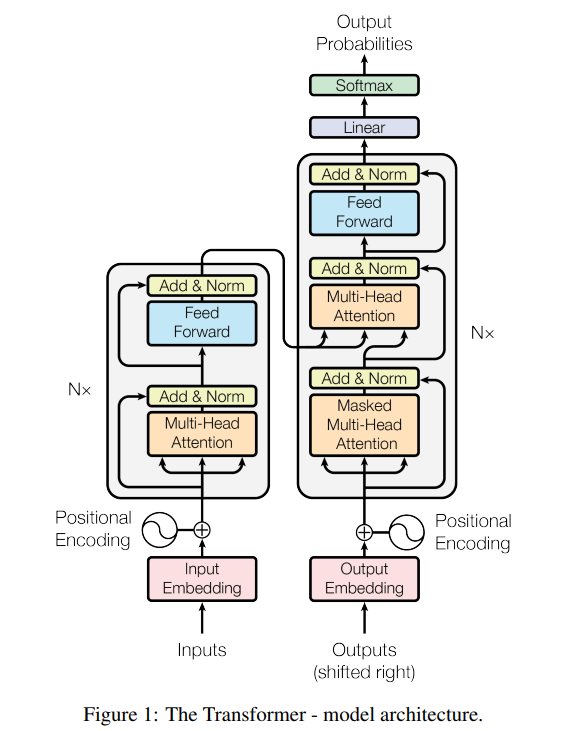
1. 长短期记忆（LSTM）和门控循环神经网络（GRU）作为序列建模和转换问题的主流方法，使得许多相关研究以循环神经网络和**编码器-解码器体系结构**进行。
2. 由于RNN需要将序列中的每个位置数据与计算时刻的步骤对齐，即所产生的一系列隐藏状态，需要用上一时刻隐藏状态和当前时刻的信息作为输入来获得，而这种固有的顺序性导致了训练样本难以进行并行化计算。
3. 本论文的研究者们发现**注意力机制**能在不考虑输入或输出序列中的距离的情况下，对依赖关系进行建模，但是当时现行的这些注意力机制却通常都与RNN一起绑定使用。
4. 如何避免顺序计算的网络架构，构建并行计算的模块，是一个能有效提高算法效率的热门问题。

**Transformer的提出：**

Transformer是第一个完全依靠self-attention（自注意力机制）来计算其输入和输出的表示，而不使用RNN或卷积的模型。因此，相比传统的RNN模型记忆长度有限且无法并行化（只有计算完时刻后的数据才能计算​时刻的数据），Transformer可以完全做到并行化计算。【这也是Transformer的主要优势】

**Transformer的模型架构：**

（1）整体架构；



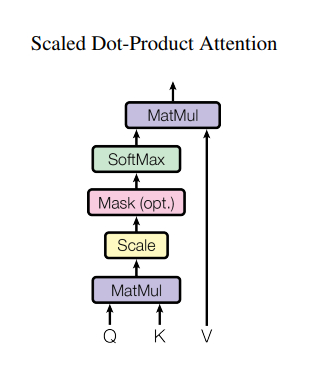
如上图所示，Transformer主要遵循编码器（图中左边）—解码器（图中右边）结构。其中，每个编码器由个相同的模块组成。每个模块又主要由两个子层构成，分别是多头自注意力机制（Multi-Head Attention）和前馈网络（Feed Forward）。此外，对两个子层使用残差连接（Add），并加入层归一化（Norm）。

同样，解码器也由个相同的模块组成。除了与编码器模块中一样的两个子层外，解码器还加入了第三个子层，该子层对编码器的输出执行带掩蔽的多头注意力机制（Masked Multi-Head Attention），即确保对位置的预测只依赖于小于位置的已知输出来计算。

（2）注意力机制；

**缩放点积注意力：（细节模块）**

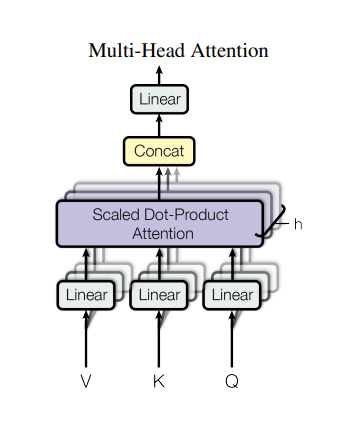
上述结构中的attention函数可以被简单描述为将query和一组key-value映射到输出outputs，其中query、key、value和outputs都是向量。outputs被计算为value的加权求和，其中分配给每个value的权重由query与对应的key的兼容函数来计算得出。



如图所示，这是attention函数的详细结构，也可用公式表达为：

其中，为缩放因子，由key的维度决定。此外，之所以采用点积是因为它能利用高度优化的矩阵乘法来实现计算（有效并行化）。

**多头注意力：（整体模块）**



如图所示，将query、key和value向量分别用不同的、所学到的线性映射层以倍降维到、和维，而不是直接用维的query、key和value向量来计算单个attention函数，具体公式如下：

其中，；

如公式所示，基于每个映射模块的query、key和value，通过并行执行attention函数，产生维的输出值，然后将它们连接（Concat）并再次映射，来产生最终值。【实际上，就是多了一个能调参的超参数】

**多头注意力模块的应用；**

在编码器中：

包含了self-attention层，即在self-attention层中，所有的key、value和query来自相同的位置（即将输入复制成三份）。因此，在编码器中的每个位置都可以注意到编码器上一层的所有位置。

在解码器中：

在Masked Multi-Head Attention模块中，使用mask层来屏蔽所在计算的位置之后的位置信息，同时也采用self-attention层。

而在Multi-Head Attention层中，query来自前一个解码器层，而key和value来自编码器的输出。

（3）位置前馈网络；

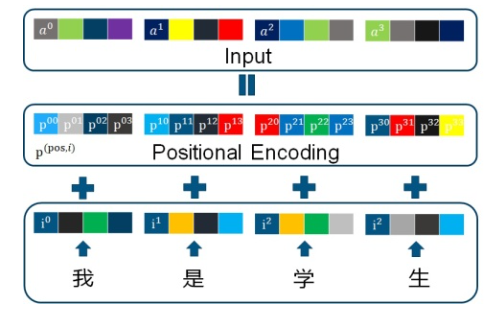
在上图中，可以注意到在每个编码器和解码器模块中都包含了一个完全连接的前馈网络。该前馈网络单独且相同地应用于每个位置，而它主要包括两个线性变换以及中间的一个激活，具体计算公式如下：

实际上，它的主要目的是把输出结果映射到我们想要的语义空间中。

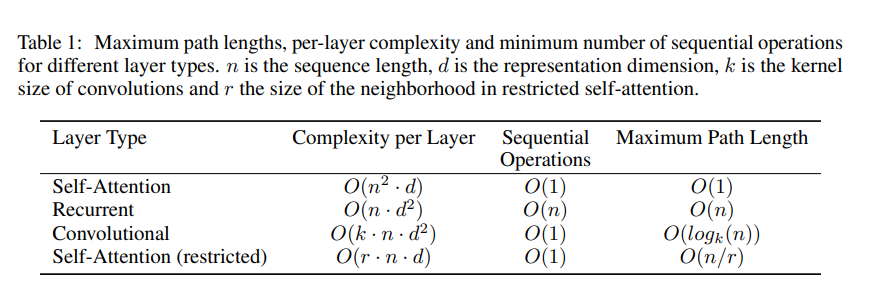
（4）位置编码；

由于该模型不包含循环和卷积，因此为了让模型能够利用序列的顺序，各模块的输入就必须加入序列中关于词符相对或者绝对位置的一些信息。为此，采用将“位置编码”添加到编码器和解码器堆栈底部的输入的方式。

且由于“位置编码”和输入的维度相同，所以它们可以相加，如下图所示：



**关于自注意力机制的分析：**



如上表所示，通过比较self-attention与循环层和卷积层的各个方面，可以清楚地得出，self-attention不仅在计算的总复杂度上优于其他结构，还能有效进行并行计算。（这也是自注意力机制的优势所在，因此如今受到研究者的追捧。）

**实验细节与结果：**

（1）优化算法；

使用Adam算法优化，其中，并根据下述公式在训练过程中改变学习率：

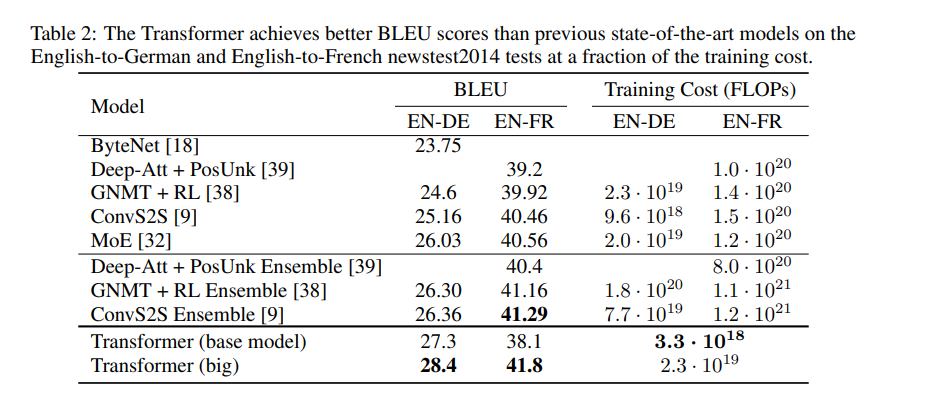
其中，。

（2）正则化；

训练主要采用两种正则化：

1. Residual Dropout：将dropout应用到每个子层的输出中，。
2. Label Smoothing：在训练过程中，使用label smoothing的值为。尽管这让模型不易理解，但这有效提高了准确性和BLEU得分。

（3）结果；



从图中可以看到，Transformer模型在WMT 2014的任务上取得了最优的分数。

**总结：**

Transformer作为一个既CNN和RNN之后的新模型架构，有效解决了当时RNN所存在的无法有效并行计算的问题，同时它也在翻译任务上取得了更优异的结果。总之，这给当时的深度学习研究带来了新的可能。而如今，注意力机制也已被应用于多项其他研究之中。