**Transformer**

**介绍**

**背景**

**模型结构**

**Why self-attention**

**训练**

****

本文提出了一种新的简单神经网络结构:Transformer,这种网络结构**仅基于注意机制(attention mechanisms)，完全摒弃了循环和卷积的结构**。

循环神经网络(RNN)，LSTM，GRU等循环结构的神经网络已经在NLP应用任务中取得了卓越的表现，循环模型通常沿输入和输出序列的符号位置考虑计算，产生对应位置的隐藏状态ht，ht是前一状态ht-1和位置t的函数，这种顺序序列特性使得网络无法进行并行计算，对于较长的输入序列而言，其弊端就更加明显。

注意机制已成为各种NLP任务中引人注目的序列建模和转导模型的组成部分，其允许对依赖项进行建模，而无需考虑它们在输入或输出序列中的距离。

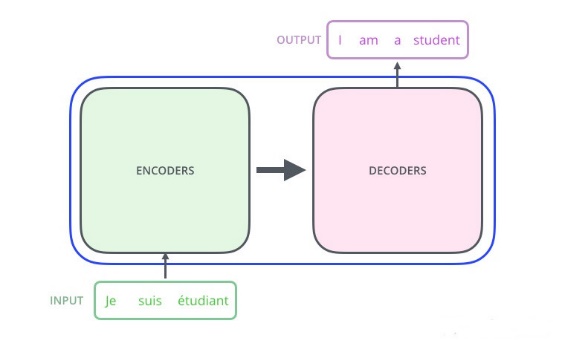
**背景：**

**自我注意（self-attention），有时也称为内部注意，是一种与单个序列的不同位置相关的注意力机制，目的是计算序列的表示形式**。End-end memory network基于循环注意力机制，而不是序列比对的复发，已证明在简单语言问答和语言建模任务中表现良好。

**Transformer是第一个完全依靠自我注意来计算其输入和输出表示的转导模型，而无需使用序列对齐的RNN或卷积。**

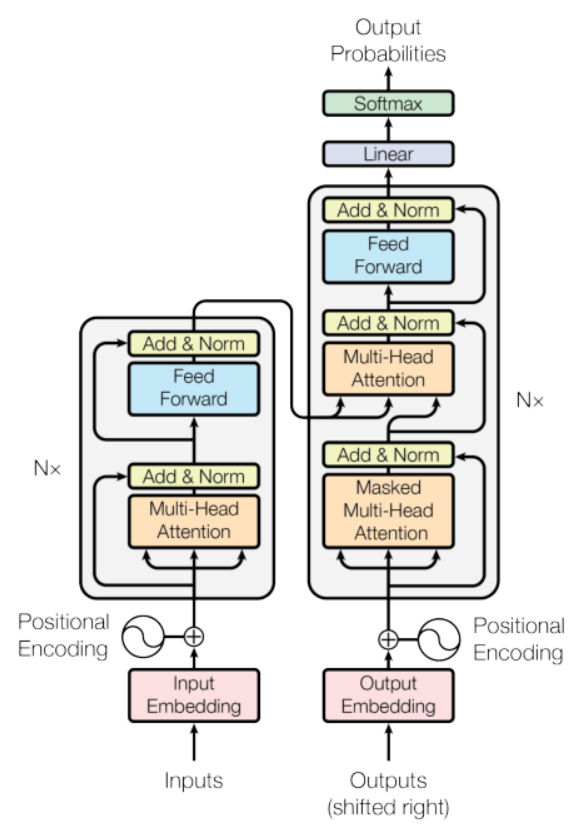
**模型结构：**

**Transformer的整体结构为Encoder-decoder structure**



Encoder 将输入序列表达（x1,x2,x3,,,xn）映射为另一连续序列表达Z=（z1,z2,z3…zn），给定**Z**，解码器decoder然后一次生成一个元素的符号的输出序列（y1，...，ym）。模型的每一步都是自动回归的，在生成下一个输出时，会将先前生成的符号用作附加输入。

Transformer 的结构遵循堆叠自我注意和点对点机制，然后对encoder和decoder使用全连接层，如图所示



**Encoder：**

Encoder由**6**个identical layer堆叠而成，每一个identical layer由2个sub-layers组成：

**第一个部分是multi-head self-attention机制**

**第二个部分是一个简单的全连接前馈网络(position-wise fully connected feed-forward network)**

在每个sub-layers中加入了残差连接（residual connection），并归一化，即每个sub-layer的输出为LayerNorm(x + Sublayer(x))。所有sub-layers中向量的维度均为d model = 512。

**Decoder：**

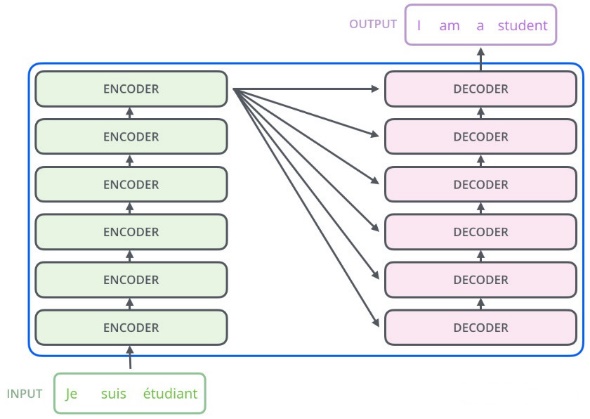
Decoder也是由6个identity layer堆叠而成:

**第一个部分是 multi-head self-attention mechanism**

**第二部分是 multi-head context-attention mechanism**

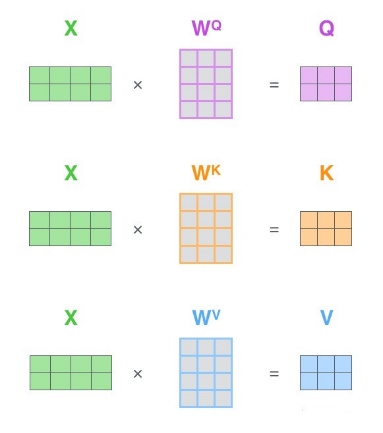
**第三部分是一个 position-wise feed-forward network**

不过，与encoder单层2个sub-layer不同的是，decoder中单层中插入了1个sub-layer：**Encoder-Decoder attention**，该层对encoder的输出进行multi-head attention 机制处理。与encoder结构相似，每个sub-layer都进行了残差连接并归一化。本文还修改了解码器堆栈中的自注意力子层，以防止位置出现在后续位置上。

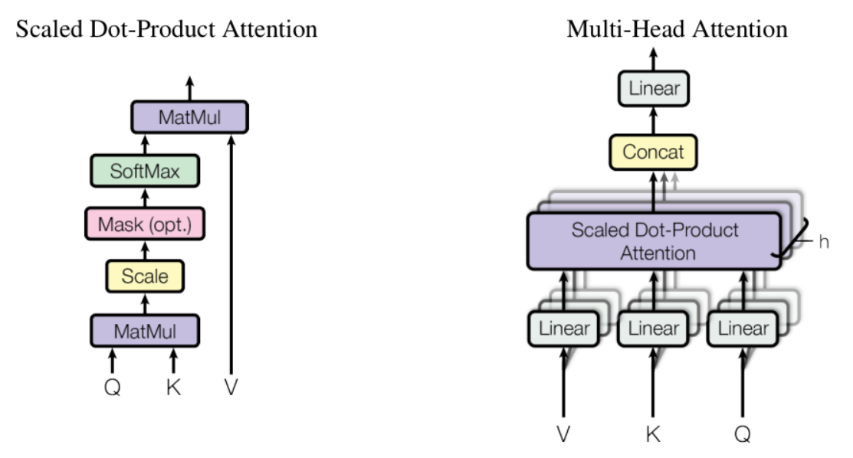


**Attention：**

注意函数可以描述为关于Query，Key-Value对的映射，论文中并没有对Query，Key，Value进行说明。它们是通过embedding向量 X乘以三个不同的权值矩阵 ，， 得到，其中三个矩阵的尺寸也是相同的。



Attention的输出为values的加权和，其中每个value对应的权重（weight）由Q和对应的K经过一系列函数运算得到，如下图左所示。



**Scaled Dot-Product Attention:**

本文把这一注意函数称为Scaled Dot-Product Attention，其输入为上文提到的Q，K, V。Q和K向量维度均为dk，V向量维度为dv。整个注意函数的表达为：

Attention(Q,K,V) = softmax()V

其中，先对Q和K进行点乘，然后除以，再经过softmax处理，与V向量相乘。除以操作的目的是**因为点乘值过大，使得softmax函数值位于梯度很小的区域**。

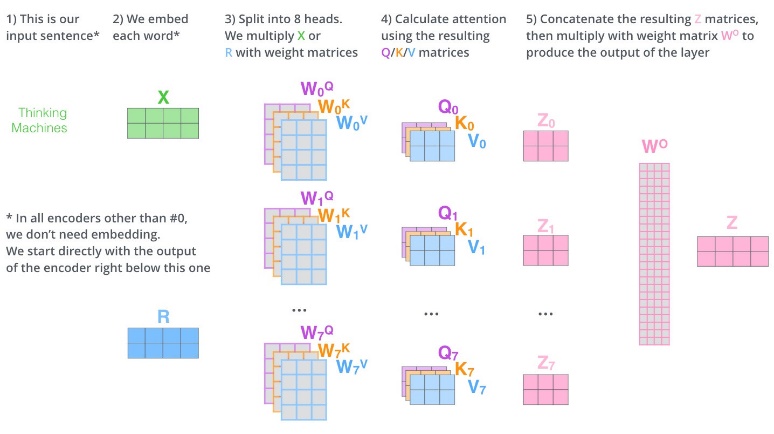
**Multi-Head Attention：**

Multi-head attention机制能让模型考虑到不同位置的Attention，另外Multi-head Attention可以在不同的子空间表示不一样的关联关系，使用单个Head的Attention一般达不到这种效果。

MultiHead(Q,K,V) = Concat(head1 ,...,headh )

where head i = Attention(Q, K, V)

本文中使用8个并行的head，对于每一个head，dk=dv =dmodel/h=64，整个多头注意运行过程如下图所示：



**Attention application：**

transformer中以三种不同的方式使用了“多头”Attention：

1) 在"Encoder-Decoder Attention"层，Query来自先前的decoder层，并且Key和Value来自Encoder的输出。Decoder中的每个位置Attend输入序列中的所有位置。

2) Encoder中的Self-attention层。在Self-attention层中，所有的Key、Value和Query都来同一个地方，这里都是来自Encoder中前一层的输出。Encoder中当前层的每个位置都能Attend到前一层的所有位置。

3) 类似的，decoder中的Self-attention层允许解码器中的每个位置Attend当前解码位置和它前面的所有位置。这里需要屏蔽解码器中向左的信息流以保持自回归属性。具体的实现方式是在缩放后的点积Attention中，屏蔽（设为负无穷）Softmax的输入中所有对应着非法连接的Value。

**Position-wise前馈网络：**

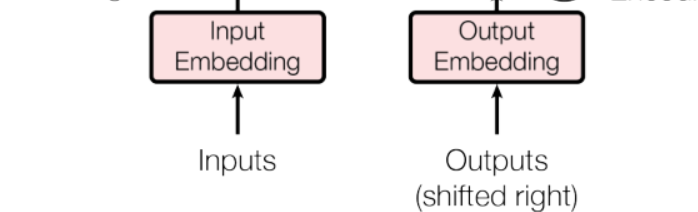
除了attention机制子层以外，encoder和decoder每层还包含一个全连接前馈网络，分别应于每个位置，其中包含了中间带有RELU作为激活函数的两个线性变换。

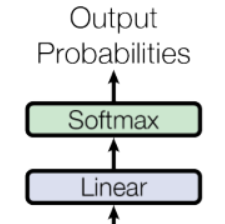
FFN(z) = max(0,zW1 + b1 )W2 + b2

尽管线性变换在不同位置是相同的，但层与层之间的参数不同，这相当于在层与层之间使用两个尺寸为1的卷积核。

**Embedding and Softmax:**

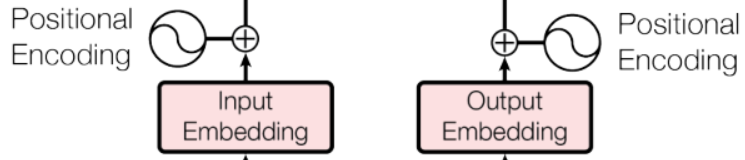
与其他序列转换模型一样，我们也使用预学习的embeddings将输入tokens和输出tokens转换为d model维向量。本文还使用预训练的线性变换和softmax函数将decoder的输出转换为预测下一个token的概率。在本文的模型中，在两个embedding 层和pre-softmax线性变换中共享相同的权值矩阵。





**Positional encoding：**

**由于模型中不包含循环和卷积结构，为了能够利用序列的顺序特征，本文采用了在序列中加入不同tokens之间相对或绝对位置的信息**：在encoder和decoder输入embeddings中加入positional encoding，positional encoding和input embeddings 的维度相同，所以可以直接相加。



在本文的工作中，使用了不同频率的正弦和余弦函数：

PE (pos,2i) = sin(pos/10000^2i/d model )

PE (pos,2i+1) = cos(pos/10000^2i/d model )

之所以选择正弦和余弦函数作为位置编码，是因为对于任意固定的偏移量k，都可以表示为的线性函数。本文也测试了使用预学习的positional embeddings作为位置编码，这两种方法测试结果相近，不过本文为了使模型推断出比训练期间遇到的序列长的序列长度，选择正弦函数作为位置编码。

总结：

Transformer不同于循环和卷积网络结构，完全依赖注意力机制，具有更快的训练速度和更好的翻译任务表现。Transformer也是后来一个著名预训练模型BERT的重要组成部分。