基于注意的transformer

abstract

提出了一个新的简单的网络架构，**transformer**，完全基于注意力机制，完全摒弃**递归和卷积**

introduction

提出了Transformer，这是一种避免重复的模型架构，而是**完全依赖于注意机制**来绘制输入和输出之间的全局依赖关系

background

Transformer是第一个**完全依赖于自关注**来计算其输入和输出表示的转导模型，而**不使用序列对齐rnn或卷积**

Model Architecture

Transformer模型：提出了一种全新的序列到序列的模型架构，完全摒弃了**循环神经网络（RNN）和卷积神经网络（CNN）**，仅依赖注意力机制来建立输入和输出之间的全局依赖关系。

编码器-解码器结构：模型由编码器和解码器两部分组成，均使用堆叠的自注意力层和逐点全连接层。

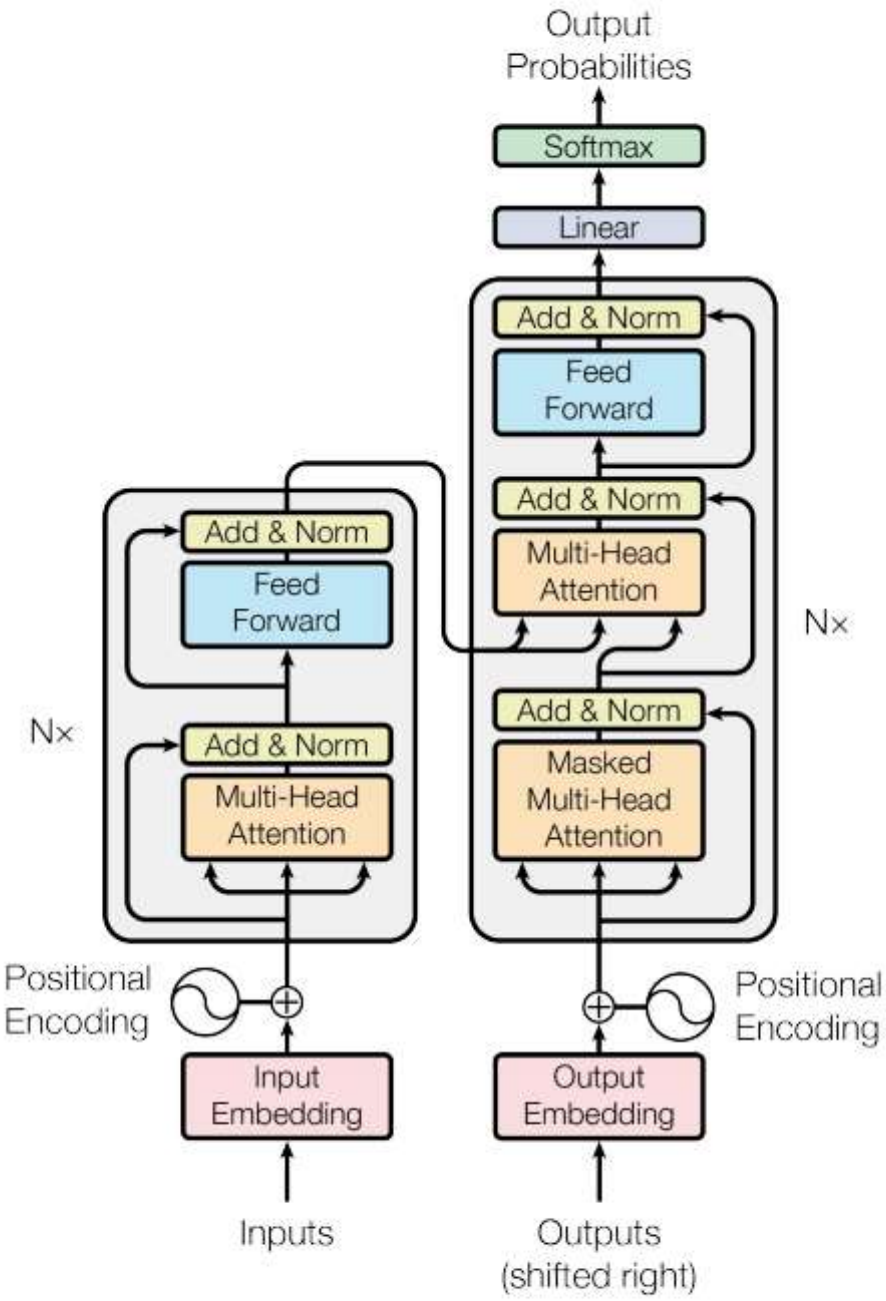


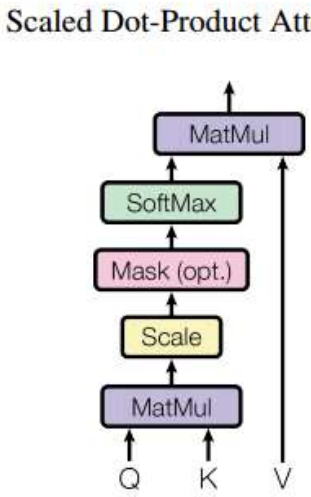
Figure 1: The Transformer - model architecture.

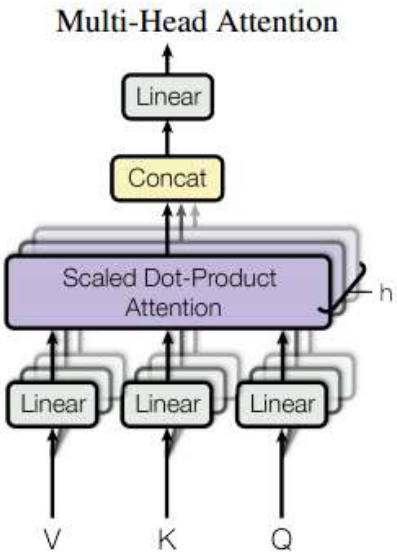
编码器，解码器

注意力机制

多头注意机制，有点点像NN神经网络里面的W，b矩阵 注意函数可以描述为将查询和一组键值对映射到输出，其中查询、键、值和输出都是向量。输出以加权和的形式计算

Scaled Dot-Product Attention（缩放的点积注意力）





这个是注意力函数：

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

自注意力层：在编码器和解码器中均使用了自注意力层，允许模型在处理序列时，每个位置都能关注到序列中的其他所有位置

Multi-Head Attention（多头注意）

$$\begin{aligned}\text{MultiHead}(Q, K, V) &= \text{Concat}(\text{head}_1, ..., \text{head}_\text{h})W^O \\ \text{where head}_\text{i} &= \text{Attention}(QW_i^Q, KW_i^K, VW_i^V)\end{aligned}$$

多头注意力：通过并行地运行多个自注意力层，模型能够同时从不同的子空间表示中学习信息

位置前馈网络

正弦位置编码：由于Transformer模型没有循环或卷积结构来捕获序列的顺序信息，因此引入了正弦和余弦函数的位置编码，以**提供关于序列中单词位置的信息**

Positional Encoding

编入时序信息

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

这个就是transformer输入时带入时序信息的方法

实验

评论

我们用来训练和评估模型的代码可以在https://github.com/ tensorflow/tensor2tensor上找到。