Attention is all you need

汇报人：李方

目录

相关介绍

相关介绍

Transformer中抛弃了传统的CNN和RNN，由且仅由self-Attenion和Feed Forward Neural Network组成。

在2014年WMT英德翻译任务中单模达到28.4BLEU,比当时的方案(包括集成模型)高了2BLEU以上。 (BLEU，全称是bilingual evaluation understudy，是评价翻译质量的指标，越大越好)

动机

动机

在 Transformer 之前，多数基于神经网络的机器翻译方法依赖于循环神经网络(RNN)。

RNN 在建模序列方面非常强大，但其序列性意味着该网络在训练时非常缓慢(无法并行计算)，因为长句需要的训练步骤更多，其循环结构也加大了训练难度。

Transformer eschewing recurrence and instead relying entirely on an attention mechanism to draw global dependencies between input and output.

Transformer避免了递归，而是完全依赖注意机制去描绘输入和输出之间的全局依赖关系。

动机

RNN虽然对于长期依赖可用LSTM等门机制缓解，但对于特别长期的依赖无法解决。

Transformer可以同时利用自注意力机制将上下文与较远的单词结合起来。通过并行处理所有单词，并让每个单词在多个处理步骤中注意到句子中的其他单词。

Self-attention is an attention mechanism relating different positions of a single sequence in order to compute a representation of the sequence.

自注意是一种将一个序列的不同位置联系起来的注意机制，用来计算序列的表示。

Attention mechanisms allowing modeling of dependencies without regard to their distance in the input or output sequences.

注意力机制允许对词语建模而不用考虑它们在输入或输出间的距离。

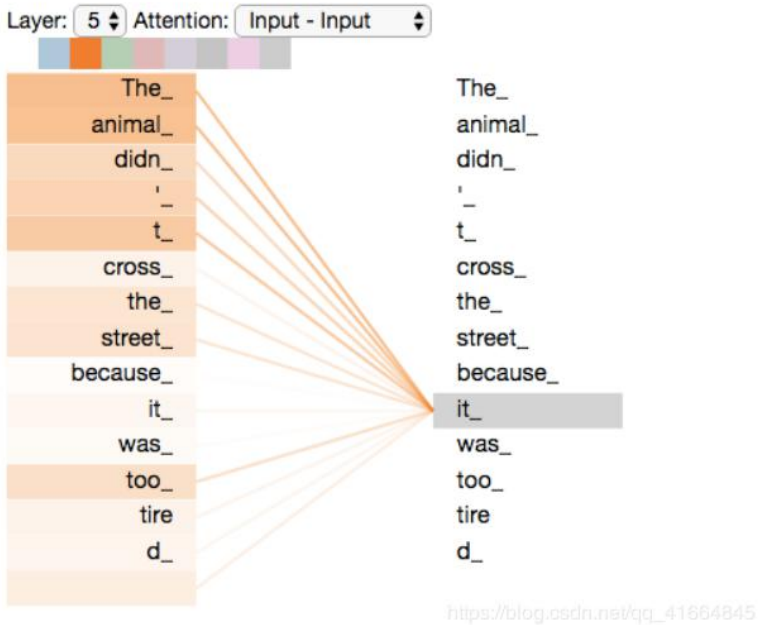
论文思路

论文思路

翻译：The animal didn't cross the street because it was too tired

在这句话中it表示什么，让机器来处理是一件难事。

self-attention的出现就是为了解决这个问题，通过self attention可以将it与animal联系起来。当模型处理单词的时候，self attention层可以通过当前单词去查看其输入序列中的其他单词，以此来寻找编码这个单词更好的线索。



可以发现Encoder在编码it的时候，部分注意力机制集中在了animal上，这部分的注意力会通过权值传递的方式影响到it的编码

论文思路

Attention mechanisms allowing modeling of dependencies without regard to their distance in the input or output sequences.

注意力机制允许对词建模而不用考虑输入输出的序列

we propoes the Transformer, a model architecture eschewing recurrence and instead relying entirely on an attention mechanism to draw global dependencies between input and output.

提出了Transformer，避免了递归，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。

模型思路

模型介绍

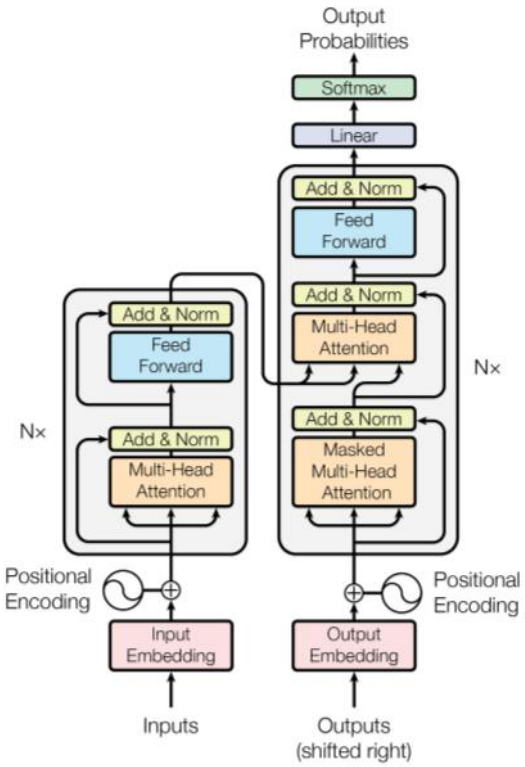
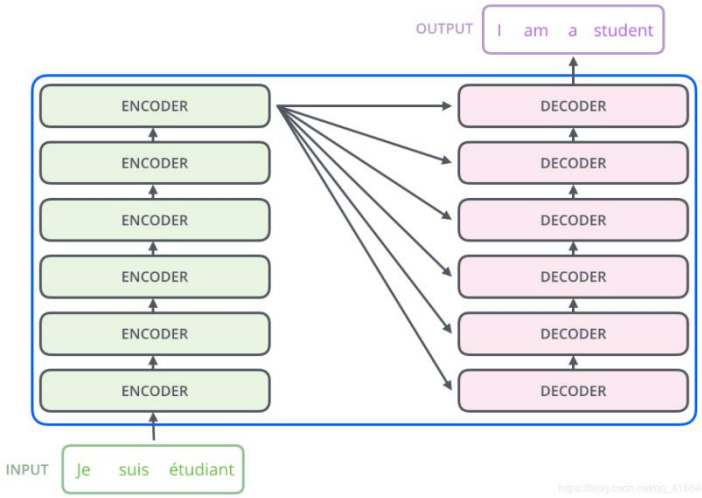


Figure 1: The Transformer - model architecture.

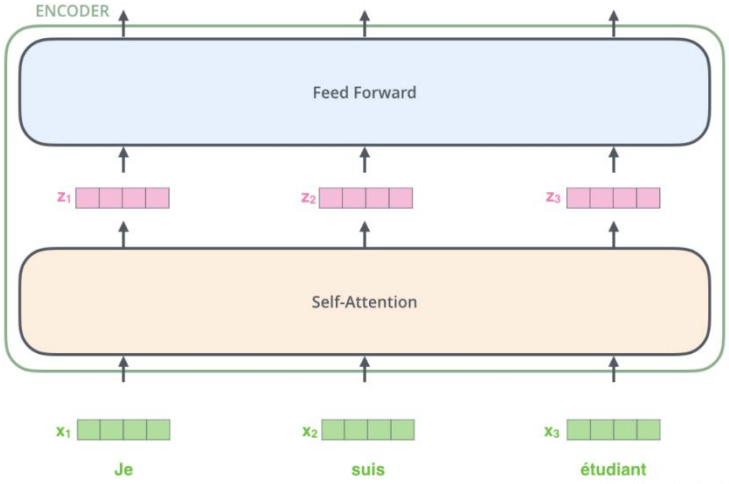
Transformer的本质上是一个Encoder-Decoder的结构。（左边是Encoders，右边是Decoders）其中每个sub-layer都加了residual connection（残差连接）和normalisation（归一化）

Decoder block比Encoder block多了一个Masked Multi-Head Attention是防止在训练的时候 使用未来的输出的单词。比如训练时，第一个单词是不能参考第二个单词的生成结果的。Masking就会把这个信息变成0，用来保证预测位置 i 的信息只能基于比 i 小的输出。

模型介绍



Encoders由N=6个Encoder block组成，Decoders也是6个Decoder block组成。与所有的生成模型相同的是，编码器的输入会作为解码器的输入。

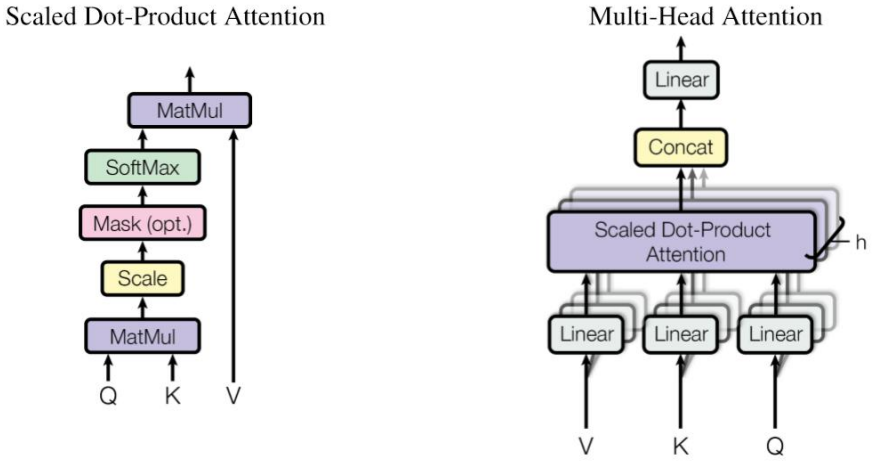


每个单词进入Self-Attention层后都会有一个对应的输出。Self-Attention层中的输入和输出是存在依赖关系的，而前馈层则没有依赖，所以在前馈层可以用到并行化来提升速度。除了attention还有一个前馈神经网络，数学公式描述如下

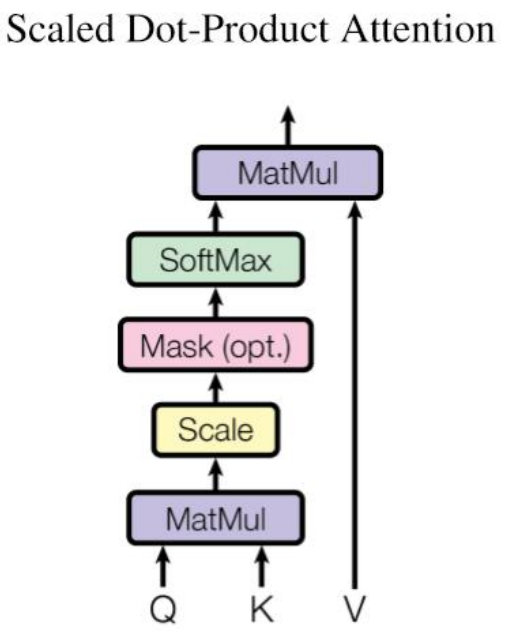
$$FFN(x) = max(0, xW_1 + b_1)W_2 + b_2$$

模型介绍

attention function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。论文里介绍了两个attention机制，一个是Scaled dot-product attention，另一个是Multi-Head Attention



模型介绍



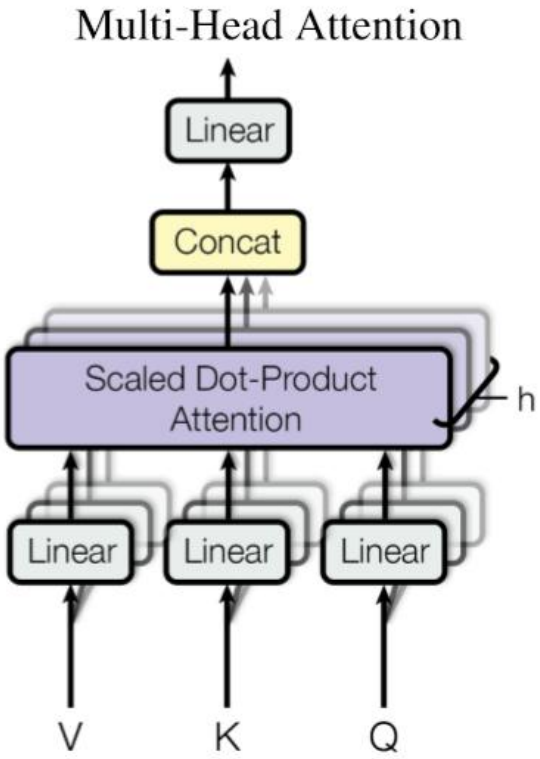
Scaled dot-product attention
其输入由维度为d_k的查询（Q）和键（K）以及维度为dv的值（V）组成，所有键计算查询的点积，并应用softmax函数获得值的权重。

具体的操作有三个步骤：
1、每个query-key 会做出一个MatMul运算过程，同时为了防止值过大除以维度的常数
2、会使用softmax 把他们归一化
3、最后会乘以V (values) 用来当做attention vector

数学公式表达：

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

模型介绍



Multi-Head Attention

Multi-Head attention则是通过h个不同的线性变换对Q，K，V进行投影，最后将不同的attention结果拼接起来，self-attention则是取Q，K，V相同。在这篇论文中，作者采用了h=8层的Scale Dot-Product Attention,然后使得$d_k = d_v = d_{model}/h = 64$

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, ..., \text{head}_h)W^O$$
$$\text{where head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V)$$

Where the projections are parameter matrices $W_i^Q \in \mathbb{R}^{d_{\text{model}} \times d_k}$, $W_i^K \in \mathbb{R}^{d_{\text{model}} \times d_k}$, $W_i^V \in \mathbb{R}^{d_{\text{model}} \times d_v}$ and $W^O \in \mathbb{R}^{hd_v \times d_{\text{model}}}$.

模型介绍

由于Transformer没有使用递归和卷积，无法获得词语的顺序信息。因此论文作者采用了Positional Encoding的方法来缓解这个问题。将encoding后的数据与embedding数据求和，加入了相对位置信息。

we add "positional encodings" to the input embeddings at the bottoms of the encoder and decoder stacks.
我们将positional encodings添加到input embeddings中

The positional encodings have the same dimension as the embeddings, so that the two can be summed.
positional encodings与embeddings具有相同的维度，因此可以对二者进行求和。

模型介绍

在本文中采用的是sin(正弦)和cos(余弦)函数来计算Positional Encoding

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

pos是位置，i是维度，位置编码的每个维度对应于一个正弦曲线。将这些信息也添加到词嵌入中，然后与Q/K/V向量点积，获得的attention就有了距离的信息了

实验结果及分析

实验结果及分析

Model	BLEU		Training Cost (FLOPs)	
	EN-DE	EN-FR	EN-DE	EN-FR
ByteNet [18]	23.75			
Deep-Att + PosUnk [39]		39.2		$1.0 \cdot 10^{20}$
GNMT + RL [38]	24.6	39.92	$2.3 \cdot 10^{19}$	$1.4 \cdot 10^{20}$
ConvS2S [9]	25.16	40.46	$9.6 \cdot 10^{18}$	$1.5 \cdot 10^{20}$
MoE [32]	26.03	40.56	$2.0 \cdot 10^{19}$	$1.2 \cdot 10^{20}$
Deep-Att + PosUnk Ensemble [39]		40.4		$8.0 \cdot 10^{20}$
GNMT + RL Ensemble [38]	26.30	41.16	$1.8 \cdot 10^{20}$	$1.1 \cdot 10^{21}$
ConvS2S Ensemble [9]	26.36	**41.29**	$7.7 \cdot 10^{19}$	$1.2 \cdot 10^{21}$
Transformer (base model)	27.3	38.1	**$3.3 \cdot 10^{18}$**	
Transformer (big)	**28.4**	**41.8**	$2.3 \cdot 10^{19}$	

结论

结论

作者总结：

we presented the Transformer, the first sequence transduction model based entirely on attention, replacing the recurrent layers most commonly used in encoder-decoder architectures with multi-headed self-attention.

我们提出了Transformer，第一个sequence transduction model完全基于注意，用multi-headed self-attention取代了recurrent layers最常用的编码器和解码器架构。

For translation tasks, the Transformer can be trained significantly faster than architectures based on recurrent or convolutional layers.

对于翻译任务， Transformer可以比基于循环网络或卷积网络的体系结构训练得快得多。

个人点评

个人点评

优点：

Transformer是一个全连接（或者是一维卷积）加Attention的结合体，抛弃了在NLP中最根本的RNN或者CNN并且取得了非常不错的效果。

Transformer的设计最大的带来性能提升的关键是将任意两个单词的距离是1，这对解决NLP中棘手的长期依赖问题是非常有效的。

算法可并行性，符合目前的环境。

缺点：

抛弃RNN和CNN，使模型丧失了捕捉局部特征的能力，RNN + CNN + Transformer的结合可能会带来更好的效果。

Transformer失去的位置信息在NLP中非常重要，即使加入Position Embedding也不能改变其失去位置信息带来的缺陷。

敬请各位大佬批评指正

Attention is all you need