你只需要注意力

ashish Vaswani*Google 大脑 avaswani@google.com

诺姆 Shazeer*Google 大脑 noam@google.com

尼基 Parmar*Google 研究 nikip@google.com

雅各布 Uszkoreit*Google 研究 usz@google.com

llion Jones*Google 研究 llion@google.com

Aidan N. Gomez*†多伦多大学 aidan@cs.toronto.edu

卢卡什 Kaiser* 谷歌大脑 lukaszkaiser@google.com

伊利亚 Polosukhin*‡ illia.polosukhin@gmail.com

摘要

主要序列转导模型基于包括编码器和解码器的复杂递归或卷积神经网络。性能最佳的模型还通过注意机制连接编码器和解码器。我们提出了一种新的简单网络架构，变压器，完全基于注意机制，完全免除重现和卷积。两个机器翻译任务的实验表明，这些模型质量优越，同时可以更加并行化，并且需要更少的时间进行训练。我们的模型在WMT 2014英语 - 德语翻译任务中达到28.4 BLEU，超过现有的最佳成绩，包括合奏，超过2 BLEU。在WMT 2014英语到法语翻译任务中，我们的模型在8个GPU上训练3.5天后，建立了一个新的单模型最新BLEU分数41.8，这是最好的培训成本的一小部分文献中的模型。我们通过将其成功应用于英语选区解析大型和有限的训练数据，表明Transformer可以很好地概括其他任务。

1 介绍

循环神经网络，长期短期记忆[13]和门控反复[7特别是神经网络已被确立为最先进的序列建模方法

*平等贡献。上市订单是随机的。雅各布提议用自我关注取代RNN，并开始努力评估这一想法。Ashish与Illia一起设计并实施了第一批变压器模型，并且在这项工作的各个方面都非常重要。Noam提出了缩放点产品注意力，多头注意力和无参数位置表示，并成为几乎涉及每个细节的另一个人。Niki在我们的原始代码库和tensor2tensor中设计，实现，调整和评估了无数的模型变体。Llion还尝试了新的模型变体，负责我们的初始代码库，以及有效的推理和可视化。Lukasz和Aidan花了无数长的时间来设计和实现tensor2tensor，取代我们早期的代码库，大大改善了结果并大大加速了我们的研究。

†在Google Brain进行的工作。

‡在Google Research上完成的工作。

第31届神经信息处理系统会议（NIPS 2017），美国加利福尼亚州长滩。

转换问题，如语言建模和机器翻译[35, 2, 5]. 自那以后，许多努力继续推动循环语言模型和编码器 – 解码器架构的界限[38, 24, 15].

递归模型通常考虑沿输入和输出序列的符号位置的计算。将位置与计算时间中的步骤对齐，它们产生一系列隐藏状态h_t，作为先前隐藏状态h_{t-1}和位置t的输入的函数。这种固有的顺序性质排除了训练样本中的并行化，这在较长的序列长度中变得至关重要，因为内存约束限制了跨越示例的批处理。最近的工作通过分解技巧实现了计算效率的显着提高[21]和条件计算[32]，同时也改善后者的模型性能。然而，顺序计算的基本约束仍然存在。

注意机制已成为各种任务中引人注目的序列建模和转换模型的组成部分，允许对依赖关系进行建模，而不考虑它们在输入或输出序列中的距离[2, 19]. 除了少数情况以外[27然而，这种注意机制与循环网络结合使用。

在这项工作中，我们提出了Transformer，一种避免重现的模型架构，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。变压器允许显着更多的并行化，并且在8个P100 GPU上经过长达12小时的培训后，可以达到翻译质量的最新技术水平。

2 背景

减少顺序计算的目标也构成了扩展神经GPU的基础[16], ByteNet [18]和ConvS2S [9]，所有这些都使用卷积神经网络作为基本构建块，并行计算所有输入和输出位置的隐藏表示。在这些模型中，关联来自两个任意输入或输出位置的信号所需的操作数量在位置之间的距离上增长，对于ConvS2S呈线性增长，对于ByteNet呈线对数。这使得学习远程位置之间的依赖关系变得更加困难[12]. 在变压器中，这被减少到一定数量的操作，虽然由于平均注意力加权位置而降低了有效分辨率，但是我们在部分中描述了多头注意力的影响。3.2.

自我关注，有时称为内部关注是关联机制，其关联单个序列的不同位置以计算序列的表示。自我关注已经成功地用于各种任务，包括阅读理解，抽象概括，文本蕴涵和学习任务独立的句子表示[4, 27, 28, 22].

端到端内存网络基于反复关注机制，而不是序列对齐的重复，并且已被证明在简单语言问答和语言建模任务中表现良好[34].

然而，据我们所知，变压器是第一个完全依靠自我关注的转换模型来计算其输入和输出的表示，而不使用序列对齐的RNN或卷积。在接下来的部分中，我们将描述变形金刚，激发自我关注并讨论其优于模型的优势[17, 18] 和[9].

3 模型架构

大多数竞争神经序列转导模型具有编码器 – 解码器结构[5, 2, 35]. 这里，编码器将符号表示的输入序列（x_1，...，x_n）映射到连续表示序列$z = (z_1, ..., z_n)$。给定z，解码器然后一次一个元素地生成符号的输出序列（y_1，...，y_m）。在每个步骤中，模型都是自回归的[10], 在生成下一个时消耗先前生成的符号作为附加输入。

Transformer遵循这种整体架构，使用堆叠的自注意和逐点，完全连接的层，用于编码器和解码器，如图的左半部分和右半部分所示。1, 分别。

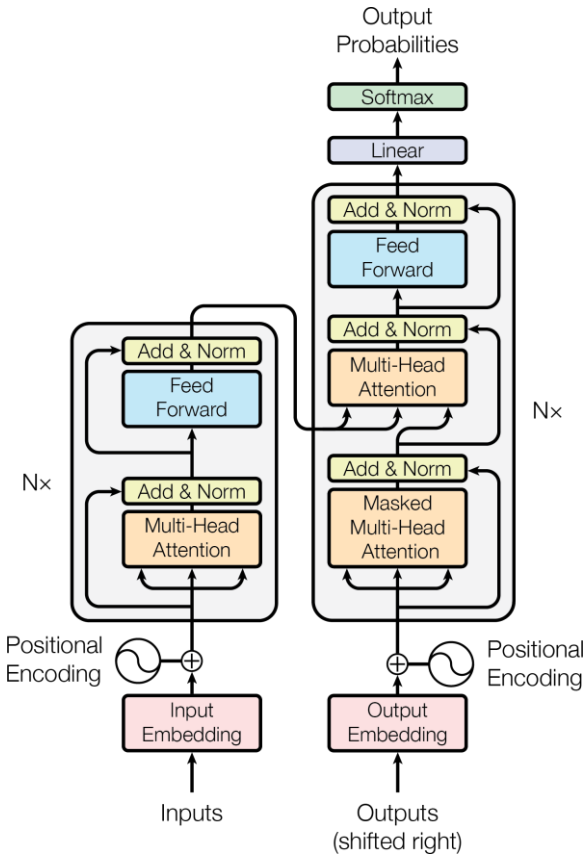

图1：变压器 - 模型架构。

3.1 编码器和解码器堆栈

编码器：编码器由一堆N = 6个相同的层组成。每层有两个子层。第一种是多头自我关注机制，第二种是简单的，位置完全连接的前馈网络。我们使用剩余连接[11]围绕两个子层中的每一个，然后进行层标准化[1].也就是说，每个子层的输出是LayerNorm（x + Sublayer（x）），其中Sublayer（x）是由子层本身实现的功能。为了促进这些残余连接，模型中的所有子层以及嵌入层产生维度$d_{模型}$ = 512的输出。

解码器：解码器也由N = 6个相同层的堆栈组成。除了每个编码器层中的两个子层之外，解码器还插入第三子层，其对编码器堆栈的输出执行多头注意。与编码器类似，我们在每个子层周围使用残余连接，然后进行层规范化。我们还修改解码器堆栈中的自注意子层以防止位置出现在后续位置。这种掩蔽与输出嵌入偏移一个位置的事实相结合，确保了位置i的预测仅依赖于小于i的位置处的已知输出。

3.2 注意

注意功能可以被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。输出被计算为值的加权和，其中分配给每个值的权重由查询与对应密钥的兼容性函数计算。

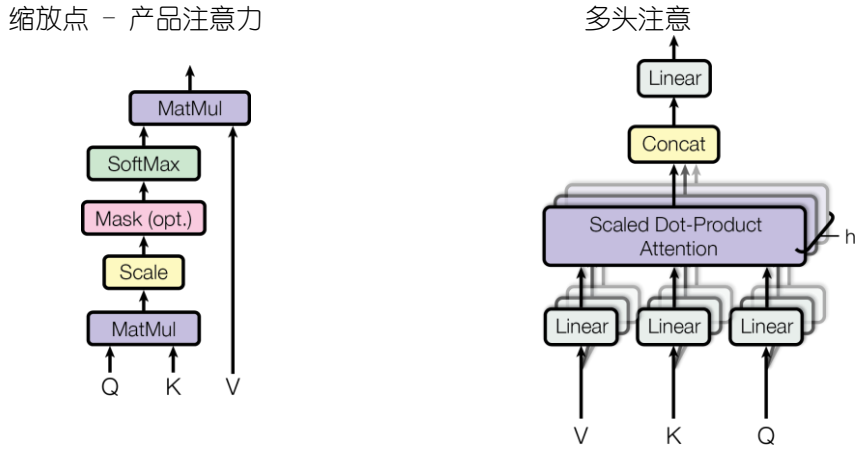

图2 :(左）标度点 - 产品注意。 （右）多头注意由几个并行运行的注意层组成。

3.2.1 缩放点 - 产品注意力

我们特别关注“Scaled Dot-Product Attention”（图2).输入包括
维度d_k的查询和密钥，维度d的$\sqrt{}$和nd值$_v$。我们计算的点积
使用所有键进行查询，将每个键 d_k，并应用softmax函数来获得权重
除以值。

在实践中，我们同时在一组查询上计算注意函数，将它们打包在一起形成矩阵Q.键和值也一起打包成矩阵K和V.我们计算输出矩阵为：

$$\text{注意}（Q，K，V）= \text{softmax}\left(\frac{qk^t}{\sqrt{d}}\right)V \quad (1)$$

两个最常用的注意功能是加分注意[2]和点积（乘法）注意。除了缩放因子$\sqrt{1}$之外，点产品注意力与我们的算法相同。附加注意使用具有单个隐藏层的前馈网络来计算兼容性功能。虽然两者在理论复杂性上相似，但是产品的关注点是
因为它可以使用高度优化的矩阵乘法代码实现，所以在实践中更快，更节省空间。

虽然对于较小的d值$_k$ ，这两种机制的表现相似，但是附加注意力优于网络产品注意力而不会缩放更大的d值$_k$ [3].我们怀疑对于大的d值$_k$，点积大幅增大，将softmax函数推向具有极小梯度的区域[4].为了抵消这种影响，我们将点积缩放$\sqrt{1}$。

$\sqrt{d_k}$

3.2.2 多头注意

我们发现用d_k，d_k 将不同的学习线性投影线性投影查询，关键和值h次，而不是使用$d_{模型}$ -维，值和查询执行单个注意函数。 ）和d_v 尺寸。在这些投影查询，键和值的每个投影版本中，我们然后并行执行注意功能，产生d$_v$维度输出值。将它们连接起来并再次投影，得到最终值，如图所示2.

[4]为了说明点积变大的原因，假设q和k的分量是独立随机的
然后，它们的点积q k = $\sum_{i=1}^{dk}$ q$_i$k$_i$具有平均值0和方差d_k。

多头注意允许模型共同关注来自不同位置的不同表示子空间的信息。只需一个注意头，平均就可以抑制这种情况。

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{头部}_1, \ldots, \text{头部}_h) W^o$$
$$\text{其中头部}_i = \text{注意力}(QW_i^q, KW_i^k, VW_i^v)$$

其中投影是参数矩阵$W_i^q \in \mathbb{R}^{d_{model} \times d_k}$，$W_i^k \in \mathbb{R}^{d_{model} \times d_k}$，$W_i^v \in \mathbb{R}^{d_{model} \times d_v}$

和$W^o \in \mathbb{R}^{hd_v \times d_{模型}}$。

在这项工作中，我们采用$h = 8$个平行注意力层或头部。对于这些中的每一个，我们使用$d_k = d_v = d_{模型}/h = 64$。由于每个头的尺寸减小，总计算成本类似于具有全维度的单头注意力。

3.2.3 注意力在我们的模型中的应用

Transformer以三种不同的方式使用多头注意力：

- 在“编码器-解码器关注”层中，查询来自先前的解码器层，并且存储器键和值来自编码器的输出。这允许解码器中的每个位置都参与输入序列中的所有位置。这模仿了序列到序列模型中的典型编码器-解码器注意机制，例如[38, 2, 9]。
- 编码器包含自我关注层。在自我关注层中，所有键，值和查询来自相同的位置，在这种情况下，是编码器中前一层的输出。编码器中的每个位置都可以处理编码器前一层中的所有位置。
- 类似地，解码器中的自注意层允许解码器中的每个位置参与解码器中的所有位置直到并包括该位置。我们需要防止解码器中的向左信息流以保持自回归属性。我们通过屏蔽（设置）softmax输入中与非法连接相对应的所有值来实现缩放点产品注意内部。见图2。$-\infty$

3.3 位置前馈网络

除了关注子层之外，我们的编码器和解码器中的每个层都包含一个完全连接的前馈网络，该网络分别和相同地应用于每个位置。这包括两个线性变换，其间有ReLU激活。

$$\text{FFN}(x) = \max(0, xW_1 + b_1)W_2 + b_2 \quad (2)$$

虽然线性变换在不同位置上是相同的，但它们在层与层之间使用不同的参数。另一种描述这种情况的方法是两个内核大小为1的卷积。输入和输出的维数为$d_{模型} = 512$，内层的维数$d_{ff} = 2048$。

3.4 嵌入和Softmax

与其他序列转导模型类似，我们使用学习嵌入将输入标记和输出标记转换为维度d的向量$_{模型}$。我们还使用通常学习的线性变换和softmax函数将解码器输出转换为预测的下一个令牌概率。在
在我们的模型中，我们在两个嵌入层和pre-softmax之间共享相同的权重矩阵
线性变换，类似于[30]。在嵌入层中，我们将这些权重乘以 d_{model}。

3.5 位置编码

由于我们的模型不包含重复和没有卷积，为了使模型能够利用序列的顺序，我们必须注入一些关于相对或绝对位置的信息。

表1：不同层类型的最大路径长度，每层复杂性和最小顺序操作数。n是序列长度，d是表示维度，k是卷积的核心大小，r是受限制的自我关注中邻域的大小。

图层类型	每层的复杂性	顺序操作	最大路径长度
自我关注	$O(n^2 \cdot d)$	$O(1)$	$O(1)$
复发	$O(n \cdot d^2)$	$O(n)$	$O(n)$
卷积	$O(k \cdot n \cdot d^2)$	$O(1)$	$O(\log_k(n))$
自我注意（受限制）	$O(r \cdot n \cdot d)$	$O(1)$	$O(n/r)$

令牌中的令牌。为此，我们将“位置编码”添加到编码器和解码器堆栈底部的输入嵌入中。位置编码具有与嵌入相同的尺寸$d_{模型}$，因此可以将两者相加。有许多位置编码选择，学习和修复[9].

在这项工作中，我们使用不同频率的正弦和余弦函数：

$$PE_{(pos,2i)} = sin(pos/10000^{2i/d_{model}})$$

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

pos是位置，我是维度。也就是说，位置编码的每个维度对应于正弦曲线。波长形成从2π到10000·2π的几何级数。我们之所以选择这个函数，是因为我们假设它可以让模型很容易地通过相对位置来学习，因为对于任何固定的偏移k，PE_{pos+k} 可以表示为PE的线性函数*销售时点情报系统*。

我们还尝试使用学习的位置嵌入[9相反，发现这两个版本产生了几乎相同的结果（见表3 第（E）行。我们选择了正弦曲线版本，因为它可以允许模型外推到比训练期间遇到的序列长度更长的序列长度。

4 为什么要自我注意

在本节中，我们将自注意层的各个方面与通常用于将一个可变长度符号表示序列（x_1，...，x_n）映射到另一个相等序列的循环和卷积层进行比较。长度（z_1，...，z_n），具有x_i，$z_i \in \mathbb{R}^d$，例如典型序列转导编码器或解码器中的隐藏层。激励我们使用自我关注，我们考虑三个需求。

一个是每层的总计算复杂度。另一个是可以并行化的计算量，通过所需的最小顺序操作数来衡量。

第三个是网络中远程依赖之间的路径长度。学习远程依赖性是许多序列转导任务中的关键挑战。影响学习这种依赖性的能力的一个关键因素是前向和后向信号必须在网络中传播的路径的长度。输入和输出序列中任何位置组合之间的这些路径越短，学习远程依赖性就越容易[12]. 因此，我们还比较了由不同层类型组成的网络中任意两个输入和输出位置之间的最大路径长度。

如表中所示1，自我关注层使用恒定数量的顺序执行的操作连接所有位置，而循环层需要0（n）个顺序操作。就计算复杂性而言，当序列长度n小于表示维度d时，自注意层比复现层更快，这通常是机器翻译中最先进模型使用的句子表示的情况。 ，如文字
[38]和字节对[31表达。为了提高涉及很长序列的任务的计算性能，可以将自我关注限制为仅考虑大小为r的邻域

输入序列以相应的输出位置为中心。这会将最大路径长度增加到O（n / r）。我们计划在未来的工作中进一步研究这种方法。

内核宽度为k <n的单个卷积层不会连接所有输入和输出位置对。这样做需要在连续内核的情况下堆叠O（n / k）卷积层，或者在扩张卷积的情况下需要O（$\log_k$（n））[18]，增加网络中任意两个位置之间最长路径的长度。卷积层通常比复发层更昂贵，为k倍。可分离的卷积[6然而，将复杂性大大降低到O（knd + nd^2）。然而，即使k = n，可分离卷积的复杂性也等于自注意层和逐点前馈层的组合，我们在模型中采用的方法。

作为附带利益，自我关注可以产生更多可解释的模型。我们检查模型中的注意力分布，并在附录中展示和讨论示例。个别注意头不仅清楚地学会执行不同的任务，许多人似乎表现出与句子的句法和语义结构相关的行为。

5 训练

本节介绍了我们模型的培训制度。

5.1 培训数据和批处理

我们使用标准的WMT 2014英语 - 德语数据集进行了培训，该数据集包含大约450万个句子对。使用字节对编码对句子进行编码[3]，具有大约37000个令牌的共享源 - 目标词汇表。对于英语 - 法语，我们使用了大得多的WMT 2014英语 - 法语数据集，该数据集由36M个句子组成，并将令牌分成32000个单词词汇[38].句子对按照近似的序列长度进行批处理。每个训练批包含一组句子对，其包含大约25000个源令牌和25000个目标令牌。

5.2 硬件和时间表

我们使用8个NVIDIA P100 GPU在一台机器上训练我们的模型。对于使用本文所述超参数的基本模型，每个训练步骤大约需要0.4秒。我们对基础模型进行了总共100,000步或12小时的培训。对于我们的大型模型，（在表格底部描述3），步时间为1.0秒。大型模型经过300,000步（3.5天）的培训。

5.3 优化

我们使用了Adam优化器[20 β_1 = 0.9， β_2 = 0.98，E = 10^{-9}。根据以下公式，我们在培训过程中改变了学习率：

$$lrate = d_{模型}^{-0.5} \cdot min(step\_num^{-0.5}, 步骤\_num \bullet num \bullet n \bullet)) \quad (3)$$

这对应于为第一个warmup_steps训练步骤线性地增加学习速率，然后与步数的反平方根成比例地减小它。我们使用了warmup_steps = 4000。

5.4 正则

我们在培训期间采用三种正规化：

剩余辍学我们申请辍学[33]在每个子层的输出被添加到子层输入并归一化之前。此外，我们将丢包应用于编码器和解码器堆栈中的嵌入和位置编码的总和。对于基本模型，我们使用$P_{下降}$ = 0.1的速率。

表2：变压器在英语 - 德语和英语 - 法语新闻测试2014测试中获得了比以前最先进模型更好的BLEU分数，只需培训成本的一小部分。

模型	布鲁		培训成本（FLOP）	
	埃内代	恩-弗尔	埃内代	恩-弗尔
ByteNet [18]	23.75			
深度 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	$9.6 \cdot 10^{18}$	$1.5 \cdot 10^{20}$
教育部[32]	26.03	40.56	$2.0 \cdot 10^{19}$	$1.2 \cdot 10^{20}$
深度 att + posunk 合奏[39]		40.4		$8.0 \cdot 10^{20}$
GNMT + RL合奏[38]	26.30	41.16	$1.8 \cdot 10^{20}$	$1.1 \cdot 10^{21}$
convs 2s 合奏[9]	26.36	**41.29**	$7.7 \cdot 10^{19}$	$1.2 \cdot 10^{21}$
变压器（基础型号）	27.3	38.1	**$3.3 \cdot 10^{18}$**	
变压器（大）	**28.4**	**41.8**	$2.3 \cdot 10^{19}$	

标签平滑在训练期间，我们使用值$\epsilon_{ls} = 0.1$ [36]. 这会伤害困惑，因为模型学会更加不确定，但提高了准确性和BLEU分数。

6 结果

6.1 机器翻译

关于WMT 2014英语 - 德语翻译任务，大变压器模型（表中的变压器（大））2) 优于之前报道的最佳模型（包括合奏）超过2.0 BLEU，建立了最新的BLEU得分28.4。该模型的配置列于表格的最后一行3. 8个P100 GPU上的培训需要3.5天。甚至我们的基础模型也超过了之前发布的所有模型和合奏，而且只占培训成本的一小部分。

在WMT 2014英语到法语翻译任务中，我们的大型模型获得了41.0的BLEU分数，优于以前发布的所有单一模型，不到以前最先进技术培训成本的1/4模型。使用英语到法语训练的变形金刚（大）模型使用辍学率$P_{下降} = 0.1$，而不是0.3。

对于基本模型，我们使用通过平均最后5个检查点获得的单个模型，这些检查点以10分钟的间隔写入。对于大型模型，我们平均了最后20个检查点。我们使用光束搜索，光束大小为4，长度罚分$\alpha = 0.6$ [38]. 在开发集上进行实验后选择这些超参数。我们将推理期间的最大输出长度设置为输入长度+50，但尽可能提前终止[38].

表2 总结了我们的结果，并将我们的翻译质量和培训成本与文献中的其他模型架构进行了比较。我们通过将训练时间，使用的GPU数量和每个GPU的持续单精度浮点容量的估计相乘来估计用于训练模型的浮点运算的数量[5].

6.2 型号变化

为了评估变压器不同组件的重要性，我们以不同的方式改变我们的基本模型，测量开发集上的英语 - 德语翻译的性能变化，newstest2013。我们使用了上一节中描述的波束搜索，但没有检查点平均值。我们将这些结果呈现在表中3.

在表中3 行（A），我们改变注意头的数量和注意键和值的维度，保持计算量不变，如章节所述3.2.2. 虽然单头注意力比最佳设置低0.3 BLEU，但是头部太多也会使质量下降。

[5]我们分别对K80，K40，M40和P100使用了2.8,3.7,6.0和9.5 TFLOPS的值。

表3：变压器架构的变化。未列出的值与基本模型的值相同。所有指标均在英语 - 德语翻译开发集newstest2013上。根据我们的字节对编码，列出的困惑是每个词，并且不应该与每个词的困惑进行比较。

	N	d_{model}	*达夫*	h	d_k	d_v	*普洛普*	*埃尔斯*	火车步骤	PPL（开发）	布鲁（开发）	params $\times 10^6$
基础	6	512	2048	8	64	64	0.1	0.1	100K	4.92	25.8	65
(A)				1	512	512				5.29	24.9	
				4	128	128				5.00	25.5	
				16	32	32				4.91	25.8	
				32	16	16				5.01	25.4	
					16					5.16	25.1	58
					32					5.01	25.4	60
(C)	2									6.11	23.7	36
	4									5.19	25.3	50
	8									4.88	25.5	80
		256			32	32				5.75	24.5	28
		1024			128	128				4.66	26.0	168
			1024							5.12	25.4	53
			4096							4.75	26.2	90
(D)							0.0			5.77	24.6	
							0.2			4.95	25.5	
								0.0		4.67	25.3	
								0.2		5.47	25.7	
(E)	位置嵌入而不是正弦曲线									4.92	25.7	
大	6	1024	4096	16			0.3		300K	**4.33**	**26.4**	213

表4：Transformer很好地概括了英语选区解析（结果在WSJ第23节）

分析器	训练	WSJ 23 F1
Vinyals&Kaiser el al。（2014）[37]	仅限WSJ，具有歧视性	88.3
Petrov等人。（2006年）[29]	仅限WSJ，具有歧视性	90.4
朱等人。（2013）[40]	仅限WSJ，具有歧视性	90.4
戴尔等人。（2016）[8]	仅限WSJ，具有歧视性	91.7
变压器（4层）	仅限WSJ，具有歧视性	91.3
朱等人。（2013）[40]	半监督	91.3
黄和哈珀（2009年）[14]	半监督	91.3
McClosky等。（2006年）[26]	半监督	92.1
Vinyals&Kaiser el al。（2014）[37]	半监督	92.1
变压器（4层）	半监督	92.7
Luong等。（2015）[23]	多任务	93.0
戴尔等人。（2016）[8]	生成的	93.3

在表中3 行（B），我们观察到减少注意键大小d_k会损害模型质量。这表明确定兼容性并不容易，并且比点积更复杂的兼容性功能可能是有益的。我们在行（C）和（D）中进一步观察到，正如预期的那样，更大的模型更好，并且辍学对于避免过度拟合非常有帮助。在行（E）中，我们用学习的位置嵌入替换我们的正弦位置编码[9]，并观察基本模型几乎相同的结果。

6.3 英国选区解析

为了评估变形金刚是否可以推广到其他任务，我们进行了英语选区分析的实验。这项任务提出了具体的挑战：产出受到强烈的结构性影响

约束并且明显长于输入。此外，RNN序列到序列模型无法在小数据体系中获得最先进的结果[37].

我们在Penn Treebank的华尔街日报（WSJ）部分训练了一个4层变压器，其中$d_{模型}$ = 1024 [25]，约40K训练句。我们还在一个半监督的环境中训练它，使用更大的高可信度和BerkleyParser语料库，大约17M句子[37].我们使用16K令牌的词汇表仅用于WSJ设置，并使用32K令牌的词汇表用于半监督设置。

我们只进行了少量实验来选择辍学，包括注意力和残留（部分5.4),第22节开发集中的学习率和波束大小，所有其他参数与英语 - 德语基本翻译模型保持不变。在推理过程中，我们将最大输出长度增加到输入长度+ 300.我们使用的光束尺寸为21， α = 0.3仅用于WSJ和半监督设置。

我们的结果见表4 表明，尽管缺乏任务特定的调整，我们的模型表现出色得多，产生的结果比之前报道的所有模型都要好，但回归神经网络语法除外[8].

与RNN序列到序列模型相比[37]，Transformer的表现优于Berkeley-Parser[29] 即使仅在WSJ训练集上训练40K句子时也是如此。

7 结论

在这项工作中，我们提出了Transformer，这是第一个完全基于注意力的序列转换模型，用多头自我关注取代了编码器 - 解码器架构中最常用的复现层。

对于转换任务，Transformer的训练速度明显快于基于循环或卷积层的架构。在WMT 2014英语到德语和WMT 2014英语到法语翻译任务中，我们实现了最新的技术水平。在之前的任务中，我们最好的模型甚至超过了之前报道过的所有合奏。

我们对基于注意力的模型的未来感到兴奋，并计划将它们应用于其他任务。我们计划将变换器扩展到涉及文本以外的输入和输出模态的问题，并研究局部的，受限制的注意机制，以有效地处理大型输入和输出，如图像，音频和视频。让生成顺序更少是我们的另一个研究目标。

我们用于训练和评估模型的代码可在以下网站获得https://github.com/ 张量.

致谢我们感谢Nal Kalchbrenner和Stephan Gouws的富有成效的评论，更正和灵感。

参考

[1] Jimmy Lei Ba，Jamie Ryan Kiros和Geoffrey E Hinton。图层规范化。arXiv预印本*的arXiv：1607.06450*,2016.

[2] Dzmitry Bahdanau，Kyunghyun Cho和Yoshua Bengio。神经机器翻译通过联合学习对齐和翻译。CoRR，abs / 1409.0473,2014。

[3] Denny Britz，Anna Goldie，Minh-Thang Luong和Quoc V. Le。大规模探索神经机器翻译架构。CoRR，abs / 1703.03906,2017。

[4] Jianpeng Cheng，Li Dong和Mirella Lapata。用于机器读取的长期短期记忆网络。arXiv预印本*的arXiv：1601.06733*,2016.

[5] Kyunghyun Cho，Bart van Merrienboer，Caglar Gulcehre，Fethi Bougares，Holger Schwenk和Yoshua Bengio。学习使用rnn编码器 - 解码器进行统计机器翻译的短语表示。CoRR，abs / 1406.1078,2014。

[6] Francois Chollet。Xception：深度学习与深度可分离的卷积。arXiv预印本*的arXiv：1610.02357*,2016.

[7] Junyoung Chung, ÇaglarGülçehre, Kyunghyun Cho和Yoshua Bengio。门控递归神经网络在序列建模中的实证评价。CoRR, abs / 1412.3555,2014。

[8] Chris Dyer, Adhiguna Kuncoro, Miguel Ballesteros和Noah A. Smith。循环神经网络语法。在Proc。NAACL, 2016年。

[9] Jonas Gehring, Michael Auli, David Grangier, Denis Yarats和Yann N. Dauphin。卷积序列到序列学习。arXiv预印本*的arXiv: 1705.03122v2*,2017.

[10] Alex Graves。用递归神经网络生成序列。arXiv预印本*的arXiv: 1308.0850*,2013.

[11] 何开明，张翔宇，任少卿，孙健。深度残差学习用于图像识别。在IEEE计算机视觉和模式识别会议论文集，第770-778页，2016年。

[12] Sepp Hochreiter, Yoshua Bengio, Paolo Frasconi和JürgenSchmidhuber。复发网中的梯度流：学习长期依赖性的难度，2001。

[13] Sepp Hochreiter和JürgenSchmidhuber。长期短暂记忆。神经计算，9（8）：1735-1780,1997。

[14] 黄忠强和玛丽哈珀。自我训练PCFG语法，具有跨语言的潜在注释。在2009年自然语言处理经验方法会议论文集，第832-841页。ACL，2009年8月。

[15] Rafal Jozefowicz, Oriol Vinyals, Mike Schuster, Noam Shazeer和Wu Yonghui Wu。探索语言建模的局限性。arXiv预印本*的arXiv: 1602.02410*,2016.

[16] Ł ukaszKaiser和Samy Bengio。活动内存可以取代注意吗？“神经信息处理系统进展”（NIPS），2016年。

[17] Ł ukaszKaiser和Ilya Sutskever。神经GPU学习算法。在2016年国际学习代表会议（ICLR）上。

[18] Nal Kalchbrenner, Lasse Espeholt, Karen Simonyan, Aaron van den Oord, Alex Graves和Koray Kavukcuoglu。线性时间的神经机器翻译。arXiv预印本*的arXiv: 1610.10099v2*,2017.

[19] Yoon Kim, Carl Denton, Luong Hoang和Alexander M. Rush。结构化的关注网络。2017年国际学习代表会议。

[20] Diederik Kingma和Jimmy Ba。亚当：随机优化的一种方法。在ICLR，2015年。

[21] Oleksii Kuchaiev和Boris Ginsburg。LSTM网络的分解技巧。arXiv预印本*的arXiv: 1703.10722*,2017.

[22] Linhan Han, Feng Minwei, Cicero Nogueira dos Santos, Mo Yu, Bing Xiang, Bowen Zhou和Yoshua Bengio。结构化的自我痴迷句子嵌入。arXiv预印本*的arXiv: 1703.03130*,2017.

[23] Minh-Thang Luong, Quoc V. Le, Ilya Sutskever, Oriol Vinyals和Lukasz Kaiser。多任务序列到序列学习。arXiv预印本*的arXiv: 1511.06114*,2015.

[24] Minh-Thang Luong, Hieu Pham和Christopher D Manning。基于注意力的神经机器翻译的有效方法。arXiv预印本*的arXiv: 1508.04025*,2015.

[25] Mitchell P Marcus, Mary Ann Marcinkiewicz和Beatrice Santorini。建立一个大的注释英语语料库：penn treebank。计算语言学，19（2）：313-330,1993。

[26] David McClosky, Eugene Charniak和Mark Johnson。有效的自我训练解析。在NAACL人类语言技术会议论文集，主要会议，第152-159页。ACL，2006年6月。

[27] Ankur Parikh, OscarTäckström, Dipanjan Das和Jakob Uszkoreit。可分解的注意力模型。在自然语言处理的经验方法，2016年。

[28] Romain Paulus, Caiming Xiong和Richard Socher。抽象概括的深度强化模型。arXiv预印本*的arXiv：1705.04304*,2017.

[29] Slav Petrov, Leon Barrett, Romain Thibaux和Dan Klein。学习准确，紧凑和可解释的树注释。在第21届计算语言学国际会议论文集和第44届ACL年会上，第433-440页。ACL，2006年7月。

[30] Ofir Press和Lior Wolf。使用输出嵌入来改进语言模型。arXiv预印本*的arXiv：1608.05859*, 2016.

[31] Rico Sennrich, Barry Haddow和Alexandra Birch。具有子词单元的罕见词的神经机器翻译。arXiv预印本*的arXiv：1508.07909*, 2015.

[32] Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton和Jeff Dean。令人难以置信的大型神经网络：稀疏门控的专家混合层。arXiv预印本*的arXiv：1701.06538*,2017.

[33] Nitish Srivastava，Geoffrey E Hinton，Alex Krizhevsky，Ilya Sutskever和Ruslan Salakhutdi-nov。辍学：一种防止神经网络过度拟合的简单方法。机器学习研究期刊，15（1）：1929-1958,2014。

[34] Sainbayar Sukhbaatar, Arthur Szlam, Jason Weston和Rob Fergus。端到端内存网络。在C.Cortes，ND Lawrence，DD Lee，M.Sugiyama和R.Garnett，编辑，Advances in Neural Information Processing Systems 28，第2440-2448页。Curran Associates，Inc.，2015。

[35] Ilya Sutskever, Oriol Vinyals和Quoc VV Le。用神经网络进行序列学习的序列。在神经信息处理系统的进展中，第3104-3112页，2014。

[36] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens和Zbigniew Wojna。重新思考计算机视觉的初始架构。CoRR，abs / 1512.00567,2015。

[37] Vinyals&Kaiser, Koo, Petrov, Sutskever和Hinton。语法作为外语。在*神经信息处理系统的进展，2015。*

[38] Wu Yonghui Wu，Mike Schuster，Zhifeng Chen，Quoc V Le，Mohammad Norouzi，Wolfgang Macherey，Maxim Krikun，Yuan Cao，Qin Gao，Klaus Macherey，et al。谷歌的神经机器翻译系统：缩小人机翻译的差距。arXiv预印本*的arXiv：1609.08144*, 2016.

[39] 周杰，曹莹，王旭光，李鹏，魏旭。具有快速连接的深度递归模型，用于神经机器转换。CoRR，abs / 1606.04199,2016。

[40] 朱木华，张悦，陈文亮，张敏，朱静波。快速准确的移位 - 减少成分解析。在ACL第51届年会的会议记录（第1卷：长篇论文），第434-443页。ACL，2013年8月。

注意可视化

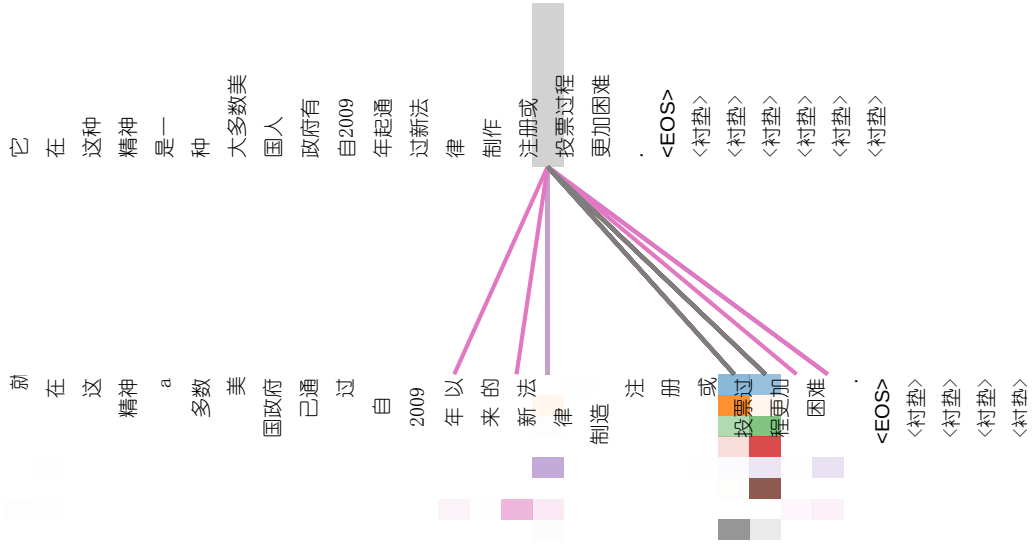

图3：在第5层中的编码器自我关注中跟随长距离依赖关系的注意机制的一个例子。许多注意力的人注意到动词'制作'的远距离依赖，完成短语'制作.....。更加困难'。此处的注意事项仅显示“制作”一词。不同的颜色代表不同的头。最好看的颜色。

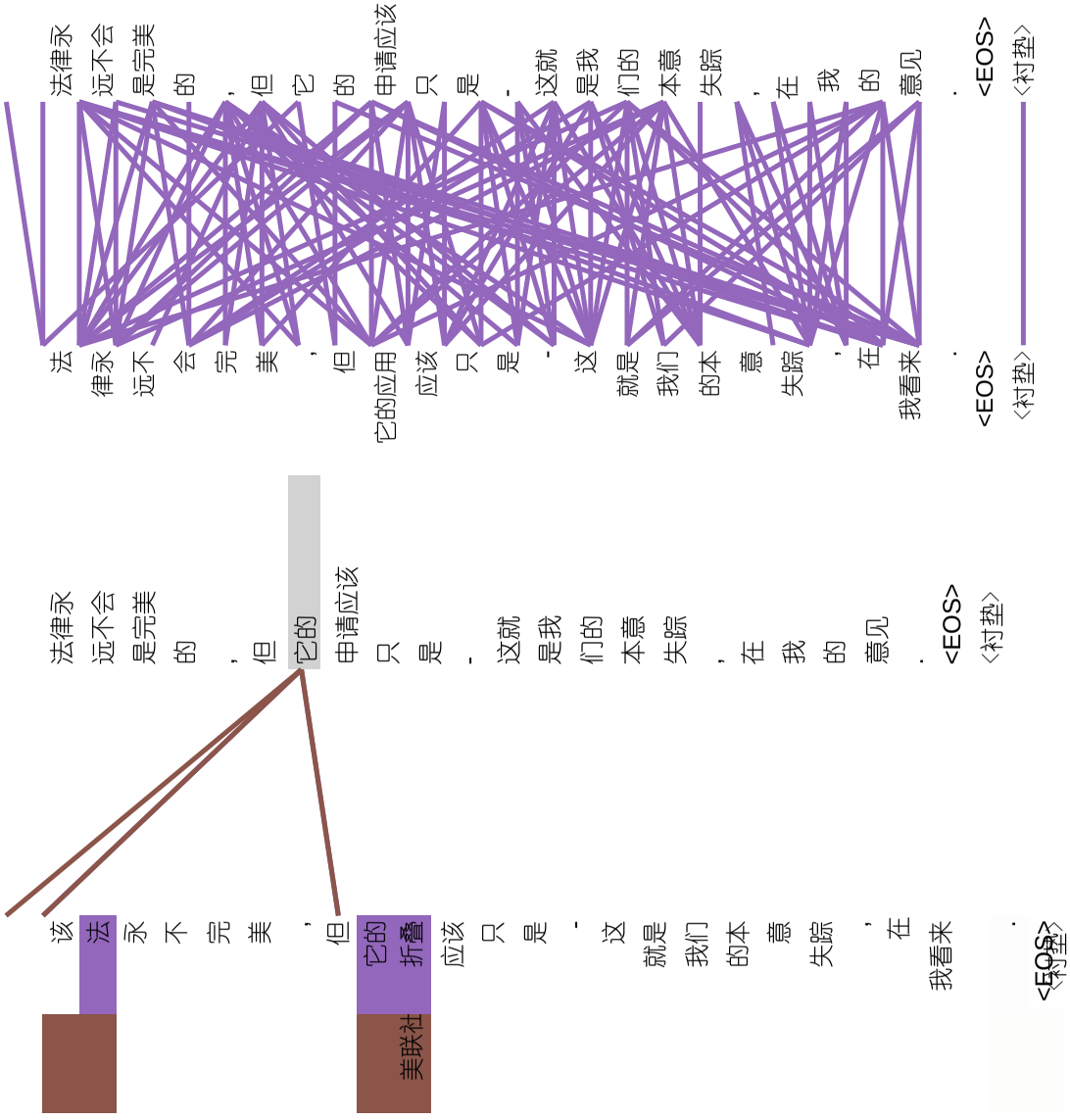

图4：两个注意头，也在6层的第5层，显然涉及回指分辨率。顶部：头部的完全注意力5. 底部：注意力集中在5和6的单词“它”的孤立注意事项。请注意，这个单词的注意力非常明确。

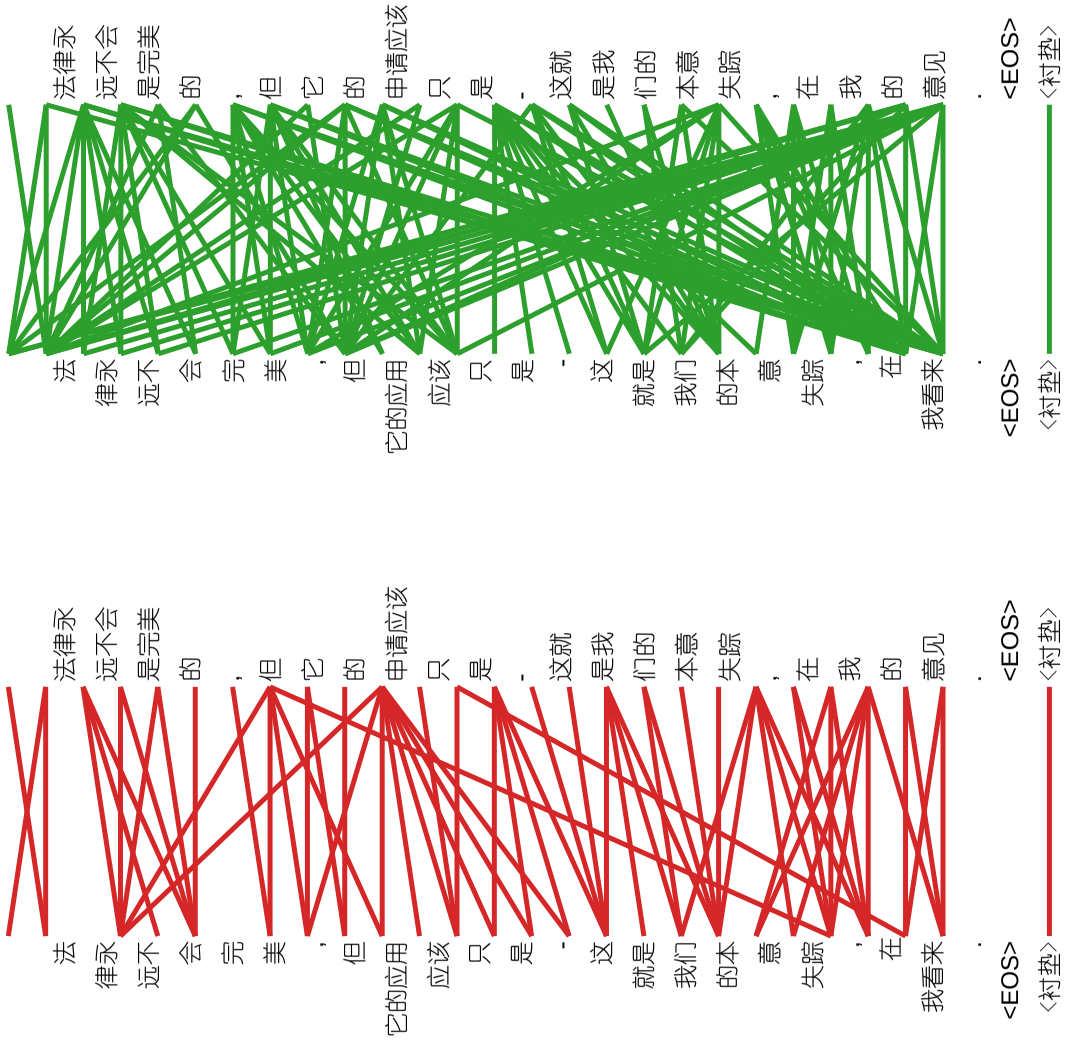

图5：许多注意头表现出与句子结构相关的行为。我们在上面给出了两个这样的例子，来自编码器自我关注的两个不同的头部，位于第5层。这些头部清楚地学会了执行不同的任务。