注意力就是你所需要的

Ashish Vaswani*谷歌大脑
avaswani@google.com

诺阿姆·沙泽尔*
谷歌大脑
noam@google.com

Niki Parmar*谷歌研究
nikip@google.com

雅各布·乌斯科雷特*谷歌研究
usz@google.com

Llion Jones*谷歌研究
llion@google.com

艾丹·N·戈麦斯*，多伦多大学，电子邮件：aidan@cs .toronto.edu

卢卡什·凯泽*谷歌大脑
lukaszkaiser@google.com

伊利亚·波洛苏欣*?
illia.polosukhin@gmail.com

摘要

主流的序列转导模型基于复杂的递归或卷积神经网络，这些网络包含编码器和解码器。表现最佳的模型还通过注意力机制连接了编码器和解码器。我们提出了一种新的简单网络架构——Transformer，该架构完全基于注意力机制，无需递归或卷积。在两个机器翻译任务上的实验表明，这些模型不仅质量更优，而且更加并行化，训练时间显著减少。我们的模型在WMT 2014英德互译任务中达到了28.4的BLEU分数，比现有的最佳结果（包括集成结果）提高了超过2个BLEU分数。在WMT 2014英法翻译任务中，我们的模型经过3.5天的训练，在八个GPU上达到了41.8的最新单模型BLEU分数，这仅占文献中最佳模型训练成本的一小部分。我们展示了Transformer模型在其他任务上的良好泛化能力，成功应用于英语成分句法分析，无论是在大量还是有限的训练数据上。

贡献平等。列表顺序随机。Jakob提议用自注意力机制替代循环神经网络，并开始评估这一想法。Ashish与Illia共同设计并实现了首个Transformer模型，并在这一工作的各个方面发挥了关键作用。Noam提出了缩放点积注意力机制、多头注意力机制和无参数位置表示，几乎参与了每一个细节。Niki在我们的原始代码库和tensor2tensor中设计、实现、调优并评估了无数模型变体。Llion也尝试了新的模型变体，负责我们最初的代码库、高效的推理和可视化。Lukasz和Aidan花了无数个漫长的日子设计和实现tenser2tenser的各个部分，取代了我们早期的代码库，极大地提高了结果，并极大地加速了我们的研究。

t在谷歌Brain工作期间完成的工作。

在谷歌研究期间完成的工作。

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。

1 介绍

循环神经网络，尤其是长短期记忆[13]和门控循环[7]神经网络，已被确立为序列建模和转换问题（如语言模型和机器翻译）的最先进方法。此后，许多研究继续拓展循环语言模型和编码器-解码器架构的边界[38,24,15]。

循环模型通常将计算过程与输入和输出序列中的符号位置相关联。通过将这些位置与计算时间的步骤对齐，它们生成一系列隐藏状态ht，作为前一个隐藏状态ht- 1和位置t的输入的函数。这种固有的顺序性使得在训练样本中难以实现并行化，这在较长的序列长度下尤为重要，因为内存限制了跨样本的批量处理。最近的研究通过使用因子化技巧[21]和条件计算[32]，在计算效率上取得了显著提升，同时在后者的情况下也提高了模型性能。然而，顺序计算的基本限制依然存在。

注意力机制已成为各种任务中强制序列建模和转换模型不可或缺的一部分，允许在不考虑输入或输出序列中距离的情况下建模依赖关系[2,19]。然而，除了少数情况[27]之外，这种注意力机制通常与循环网络结合使用。

在本研究中，我们提出了一种名为Transformer的模型架构，该架构摒弃了传统的递归方法，转而完全依赖注意力机制来捕捉输入与输出之间的全局关联。通过这种架构，Transformer能够实现显著的并行化效果，并且在仅需12小时的训练时间，使用8个P100 GPU后，就能达到翻译质量的新高度。

2 背景

减少顺序计算的目标也构成了扩展神经GPU [16]、ByteNet [18]和ConvS2S [9]的基础，所有这些都使用卷积神经网络作为基本构建块，并为所有输入和输出位置并行计算隐藏表示。在这些模型中，从任意输入或输出位置到另一个位置的信号关联所需的操作数量随着位置间距离的增加而线性增长，对于ConvS2S模型而言，这种增长是线性的；而对于ByteNet模型，则是呈对数增长。这使得学习远距离位置之间的依赖关系变得更加困难[12]。在Transformer模型中，这一问题被简化为常数级别的操作量，尽管由于平均注意力加权位置导致的有效分辨率有所下降，但通过多头注意力机制，如第3.2节所述，这一问题得到了缓解。

自注意力机制，有时也称为内部注意力，是一种通过关联单个序列中不同位置来计算该序列表示的方法。自注意力机制在多种任务中取得了成功应用，包括阅读理解、抽象摘要、文本蕴含分析以及学习任务无关的句子表示[4,27,28,22]。

端到端记忆网络基于循环注意力机制，而非序列对齐循环，并在简单语言问答和语言建模任务中表现出色[34]。

据我们所知，Transformer是首个完全依赖自注意力机制来计算输入和输出表示的转换模型，无需使用序列对齐的RNN或卷积。在接下来的部分中，我们将介绍Transformer，解释自注意力机制，并探讨其相对于[17,18]和[9]等模型的优势。

3 模型架构

大多数竞争性的神经序列转导模型都采用了编码器-解码器结构[5,2,35]。在这个结构中，编码器将输入的符号表示序列（x1，...，xn）转换为连续表示序列z =（z1，...，zn）。解码器根据给定的z，逐个生成符号序列（y1，...，ym）。在每一步中，模型采用自回归[10]，利用之前生成的符号作为生成下一个符号的额外输入。

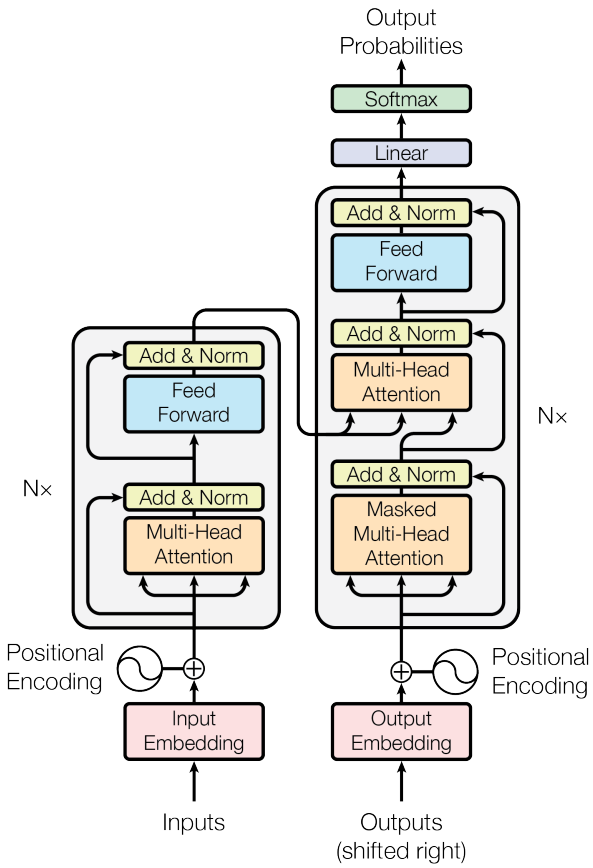

图1：变压器-模型架构。

Transformer遵循这种总体架构，使用堆叠的自注意力和点对点全连接层作为编码器和解码器，分别如图1的左半部分和右半部分所示。

3.1 编码器和解码器堆栈

编码器由N = 6个相同的层构成，每个层包含两个子层。第一个子层采用多头自注意力机制，第二个子层则是一个简单的全连接前馈网络。每个子层周围都使用了残差连接[11]，并进行了层归一化[1]。具体来说，每个子层的输出为LayerNorm（x + Sublayer(x)），其中Sublayer(x)是该子层实现的具体功能。为了便于这些残差连接，模型中的所有子层以及嵌入层都生成了维度为dmodel = 512的输出。

解码器同样由N = 6个相同的层构成。除了每个编码器层中的两个子层外，解码器还额外添加了一个第三子层，该子层会对编码器堆栈的输出进行多头注意力机制处理。类似于编码器，我们在每个子层周围采用了残差连接，并进行了层归一化。此外，我们还修改了解码器堆栈中的自注意力子层，以防止位置关注后续位置。这种掩码机制，加上输出嵌入偏移一个位置的事实，确保了位置i的预测仅依赖于位置小于i的已知输出。

3.2 注意

注意力函数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、值和输出均为向量。输出是通过加权求和计算得出的。

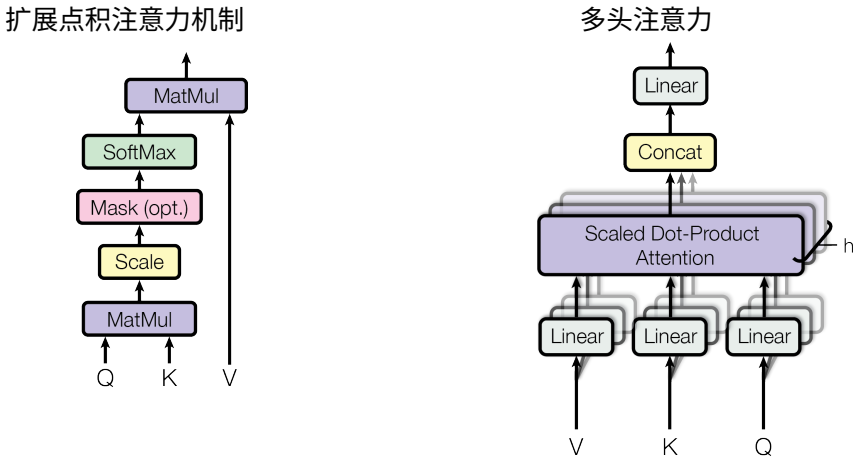

图2：（左）缩放点积注意力机制。（右）多头注意力机制由多个并行运行的注意力层构成。

其中，每个值的权重是通过查询与相应键之间的兼容性函数计算得出的。

3.2.1 扩展点积注意力机制

我们特别关注的注意力机制被称为“缩放点积注意力”（图2）。该机制的输入包括维度为dk的查询和键，以及维度为dv的值。我们首先计算查询与所有键的点积，然后将每个点积除以$\sqrt{}$dk，并通过应用softmax函数来确定值的权重。

在实际操作中，我们同时对一组查询执行注意力函数计算，这些查询被整合到矩阵Q中。同时，键和值也被分别整合到矩阵K和V中。输出矩阵的计算方式如下：

$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V \tag{1}$$

最常用的两种注意力函数是加性注意力[2]和点积（乘法）注意力。点积注意力与我们的算法相同，只是缩放因子不同。of $\frac{1}{\sqrt{d_k}}$加性注意力机制通过单层隐藏层的前馈网络来计算兼容性函数。虽然两者在理论复杂性上是相似的，但点积注意力在实践中要快得多，也更节省空间，因为它可以使用高度优化的矩阵乘法代码来实现。

当dk取较小值时，两种机制的表现相似；然而，对于较大的dk [3]值，加性注意力机制在不进行缩放的情况下优于点积注意力机制。我们推测，当dk值较大时，点积的数值会显著增大，导致softmax函数被推入其作用范围之外。

极小的梯度4.为了抵消这种效应，我们通过缩放点积来实现。$\frac{1}{\sqrt{d_k}}$.

3.2.2 多头注意力

我们发现，将查询、键和值分别线性投影到dk、dk和dv维度，而不是使用d模型维度的键、值和查询执行单一注意力函数，这种方法更为有效。在这些投影后的查询、键和值上并行执行注意力函数，从而生成dv维的

4为了说明为什么点积会变大，假设q和k的分量是均值为0、方差为1的独立随机变量。接着计算它们的点积，q · k=Σ di。$\overset{k}{=}$1个随机变量Qi的均值为0，方差为dk。

输出值被连接并再次投影，最终形成如图2所示的最终值。

多头注意力机制允许模型在不同位置同时关注来自不同表示子空间的信息。而单个注意力头则会通过平均化抑制这种能力。

多头注意力机制（Q、K、V）=Concat（头1，...，头h）WO，其中头i=Attention（QWiQ，KWiK，VWiV）

预测值由参数矩阵构成，具体形式为：WiQ∈Rdmodel×dk、WiK∈Rdmodel×dk、WiV∈Rdmodel×dv以及WO∈Rhdv×dmodel。

在本研究中，我们采用了h = 8个并行注意力层（或头）。每个层的维度dk = dv = dmodel /h = 64。由于每个头的维度较小，因此总计算成本与全维度单头注意力机制相当。

3.2.3 注意力在我们模型中的应用

Transformer以三种不同的方式使用多头注意力：

- ? 在“编码器-解码器注意力”层中，查询来自前一个解码器层，记忆键和值来自编码器的输出。这样，解码器中的每个位置都能关注输入序列的所有位置。这模仿了序列到序列模型中典型的编码器-解码器注意力机制，例如[38,2,9]。
- ? 编码器包含自注意力层。在自注意力层中，所有的键、值和查询都来自同一个地方，即编码器前一层的输出。编码器中的每个位置都可以关注前一层的所有位置。
- ? 同样，解码器中的自注意力层使得每个位置都能关注到包括该位置在内的所有位置。为了保持自回归特性，我们需要防止解码器中出现向左的信息流动。我们通过在softmax的输入中屏蔽（设为?∞）所有非法连接的值来实现这一点。参见图2。

3.3 位置性前馈网络

除了注意力子层之外，我们的编码器和解码器中的每一层都包含一个全连接的前馈网络，该网络分别且一致地应用于每个位置。这一过程由两个线性变换组成，并在中间加入了ReLU激活函数。

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

虽然线性变换在不同位置上保持一致，但各层使用的参数却有所不同。另一种描述方式是两个卷积核大小均为1的卷积操作。输入和输出的维度均为dmodel = 512，而中间层的维度为dff = 2048。

3.4 嵌入和Softmax

与其它序列转导模型类似，我们使用学习到的嵌入来将输入和输出的标记转换为维度为dmodel的向量。此外，我们还采用常规的学习线性变换和softmax函数，将解码器的输出转换为预测的下一个标记的概率。在我们的模型中，两个嵌入层和预softmax线性变换之间共享相同的权重矩阵，这与[30]模型相似。在嵌入层中，我们将这些权重乘以$\sqrt{\text{dmodel}}$。

表1：不同层类型的最大路径长度、各层复杂度及最小顺序操作次数。其中，n表示序列长度，d代表表示维度，k为卷积核大小，r为受限自注意力机制中的邻域大小。

层类型	每层的复杂性	顺序操作	最大路径长度
自注意力	$O(n^2 \cdot d)$	$O(1)$	$O(1)$
循环	$O(n \cdot d^2)$	$O(n)$	$O(n)$
卷积	$O(k \cdot n \cdot d^2)$	$O(1)$	$O(log_k(n))$
自注意力（受限）	$O(r \cdot n \cdot d)$	$O(1)$	$O(n/r)$

3.5 位置编码

由于我们的模型不包含循环和卷积，为了使模型能够利用序列的顺序，我们必须在输入嵌入中加入关于标记在序列中的相对或绝对位置的信息。为此，我们在编码器和解码器堆栈的底部添加了“位置编码”。这些位置编码与嵌入具有相同的维度dmodel，因此可以将两者相加。位置编码有多种选择，包括学习的和固定的[9]。

在本研究中，我们使用了不同频率的正弦和余弦函数：

PE（pos，2i）=sin（pos/100002i/dmodel），PE（pos，2i+1）=cos（pos/100002i/dmodel）

其中，pos表示位置，i表示维度。也就是说，每个位置编码的维度对应一个正弦波。这些波长从2π到10000 · 2π构成几何级数。我们选择这个函数是因为假设它能让模型通过相对位置轻松学习注意力机制，因为对于任何固定的偏移量k，PEpos+k可以表示为PEpos的线性函数。

我们还尝试使用学习到的位置嵌入[9]，发现两个版本的结果几乎相同（见表3第(E)行）。我们选择了正弦波版本，因为它可能允许模型对超出训练过程中遇到的序列长度进行外推。

4 为什么需要自我关注

在本节中，我们将自注意力层与常用的循环和卷积层进行对比，这些层通常用于将一个可变长度的符号表示序列（x1，...，xn）映射到另一个等长的序列（z1，...，zn），其中xi和zi∈Rd，例如，在典型的序列传输编码器或解码器中的隐藏层。为了说明我们使用自注意力层的理由，我们考虑了三个目标。

一是每层的总计算复杂度。二是可以并行化的计算量，这通过所需的最少顺序操作来衡量。三是网络中长距离依赖之间的路径长度。学习长距离依赖是许多序列转换任务中的关键挑战。影响学习这些依赖能力的一个重要因素是信号在网络中前向和后向传播的路径长度。输入和输出序列中任何位置组合之间的路径越短，学习长距离依赖就越容易[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1所示，自注意力层通过固定数量的顺序执行操作连接所有位置，而循环层则需要O(n)次顺序操作。在计算复杂度方面，当序列长度较短时，自注意力层比循环层运行更快。

长度n小于表示的维度d，这种情况在机器翻译中常用的句子表示中尤为常见，例如词片[38]和字节对[31]表示。为了提高处理超长序列任务的计算效率，可以将自注意力机制限制在以输出位置为中心、大小为r的输入序列邻域内。这样可以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。

单个卷积层，其内核宽度k小于n，并不能连接所有输入和输出位置。若使用连续内核，则需要O（n/k）层卷积层；若采用扩张卷积，则需O（logk（n））层卷积层，这会增加网络中任意两个位置之间最长路径的长度。通常情况下，卷积层的成本比递归层高k倍。然而，可分离卷积[6]显著降低了复杂度，使其降至O（k · n · d + n · d²）。即使k = n，可分离卷积的复杂度也等于自注意力层和逐点前馈层的组合，这是我们模型采用的方法。

作为附带的好处，自注意力机制能够生成更具可解释性的模型。我们检查了模型中的注意力分布，并在附录中展示了相关示例进行讨论。不仅各个注意力头明确地学习执行不同的任务，许多注意力头还表现出与句子的句法和语义结构相关的行为。

5 训练

本节描述了我们模型的训练方案。

5.1 训练数据和批处理

我们使用了标准的WMT 2014英德语数据集进行训练，该数据集包含约450万对句子。句子采用字节对编码[3]进行编码，共享源目标词汇表包含约37000个标记。对于英法语对，我们使用了规模更大的WMT 2014英法语数据集，该数据集包含3600万句子，并将标记拆分为一个包含32000个词片的词汇表[38]。句子对根据近似序列长度进行分批处理。每个训练批次包含一组句子对，包含大约25000个源标记和25000个目标标记。

5.2 硬件和时间表

我们在一台配备8个NVIDIA P100 GPU的机器上训练了模型。对于使用文中所述超参数的基本模型，每个训练步骤大约需要0.4秒。我们总共训练了100,000步，即12小时。对于大模型（见表3底部），每步时间约为1.0秒。这些大模型总共训练了300,000步，即3.5天。

5.3 优化器

我们采用了Adam优化器[20]，其参数设置为 β ?= 0 .9、 β ?= 0 .98和 ε =10??。在训练过程中，我们根据以下公式调整学习率：

$$lrate = d_{\text{mode}}^{-0.5} \text{l} \cdot \text{min}\left(\text{step\_num-0.5}, \text{step\_num} \cdot \text{warmup\_steps-1.5}\right) \tag{3}$$

这相当于在前warmup_steps个训练步骤中线性增加学习率，之后则按步数的平方根倒数比例递减。我们设warmup_steps为4000。

5.4 正则化

我们在训练过程中采用了三种类型的正则化：

表2：在英语到德语和英语到法语的newstest2014测试中，Transformer在训练成本仅为前一最先进模型的一小部分的情况下，实现了更好的BLEU分数。

模型	有蓝色霉菌花纹的		训练成本（FLOPs）	
	EN-DE	EN-FR	EN-DE	EN-FR
ByteNet [18]	23.75			
深部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乐团[39]		40.4		$8.0 \cdot 10^{20}$
GNMT + RL集合[38]	26.30	41.16	$1.8 \cdot 10^{20}$	$1.1 \cdot 10^{21}$
ConvS2S集合[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	$\mathbf{3.3 \cdot 10^{18}}$	
变压器（大）	**28.4**	**41.8**	$2.3 \cdot 10^{19}$	

残差Dropout：我们对每个子层的输出应用dropout [33]，然后将其添加到子层的输入中并进行归一化。此外，我们还对编码器和解码器堆栈中的嵌入和位置编码的总和应用Dropout。对于基础模型，我们采用的Dropout率为Pdrop = 0 .1。

在训练过程中，我们采用了标签平滑技术，其参数设置为?ls=0.1[36]。这一方法虽然增加了模型的困惑度，使其学习到更加不确定的状态，但显著提升了准确率和BLEU分数。

6 结果

6.1 机器翻译

在WMT 2014英德翻译任务中，大型变压器模型（表2中的Transformer（big））的表现比之前报道的最佳模型（包括集成模型）高出2倍以上。BLEU评分达到了28.4，创下了新的最高纪录。该模型的配置详见表3的底部。训练过程在8个P100 GPU上耗时3.5天。即使我们的基础模型也超越了所有已发表的模型和集成，而其训练成本仅为任何竞争模型的一小部分。

在WMT 2014英法互译任务中，我们的大型模型达到了41.0的BLEU分数，超越了所有先前发布的单模型，且训练成本仅为之前最先进模型的四分之一。用于英法互译的Transformer（大型）模型使用了0. 1的丢弃率，而非0.3。

对于基础模型，我们采用了通过平均最后5个检查点得到的单一模型，这些检查点每10分钟记录一次。对于大型模型，我们则取了最后20个检查点的平均值。在推理过程中，我们使用了束搜索，束大小为4，长度惩罚系数α = 0 .6 [38]。这些超参数是在开发集上通过实验确定的。我们设定了最大输出长度为输入长度+ 50，但尽可能提前终止[38]。

表2总结了我们的研究结果，并将我们的翻译质量和训练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数量以及每个GPU的持续单精度浮点运算能力来估算训练模型所需的浮点运算次数。

6.2 型号变更

为了评估Transformer不同组件的重要性，我们以不同的方式调整了基础模型，测量了在英语到德语翻译上的性能变化。

5我们分别使用了K80、K40、M40和P100的2.8、3.7、6.0和9.5 TFLOPS值。

表3：变压器架构的变体。未列出的值与基础模型相同。所有指标均基于英语到德语的翻译开发集newstest2013。所列困惑度是按字块计算的，根据我们的字节对编码方法，不应与单词层面的困惑度进行比较。

	N	d_{model}	d_{ff}	h	d_k	d_v	P_{drop}	ϵ_{ls}	火车步骤	私人飞行执照 (dev)	有蓝色霉菌花纹的 (dev)	参数 $\times 10^6$
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	
(B)				16						5.16	25.1	58
				32						5.01	25.4	60
(C)	2									6.11	23.7	36
	4	256			32	32				5.19	25.3	50
	8	1024			128	128				4.88	25.5	80
			1024							5.75	24.5	28
			4096							4.66	26.0	168
										5.12	25.4	53
										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						正弦波			4.92	25.7	
大的	6	1024	4096	16			0.3		300K	**4.33**	**26.4**	213

开发集，newtest2013。我们采用了前一节所述的束搜索方法，但未采用检查点平均法。这些结果见表3。

在表3的行(A)中，我们调整了注意力头的数量以及注意力键和值的维度，同时保持计算量不变，如第3.2.2节所述。尽管单头注意力比最佳设置低0.9 BLEU，但过多的注意力头也会导致质量下降。在表3的行(B)中，我们发现减少注意力键的大小dk会损害模型的质量。这表明确定兼容性并非易事，可能需要使用比点积更复杂的兼容性函数。在第(C)和(D)行中，我们进一步观察到，正如预期的那样，更大的模型表现更佳，而dropout技术在避免过拟合方面非常有效。在第(E)行中，我们将原本使用的正弦波位置编码替换为学习得到的位置嵌入[9]，结果与基础模型几乎完全一致。

6.3 英语成分句法分析

为了评估Transformer是否能应用于其他任务，我们在英语成分句法分析上进行了实验。这项任务面临特定的挑战：输出受到严格的结构限制，并且显著长于输入。此外，RNN序列到序列模型在小数据集上未能达到最佳结果[37]。

我们使用Penn Treebank [25]的《华尔街日报》（WSJ）部分，训练了一个dmodel = 1024的四层变压器模型，该数据集包含约40,000个训练句子。此外，我们还在半监督设置下进行了训练，使用了包含大约1700万个句子的较大高置信度和伯克利解析器语料库[37]。对于仅WSJ的设置，我们使用了16,000个词汇量；而在半监督设置中，则使用了32,000个词汇量。

我们仅在第22节开发集上进行了少量实验，以选择注意力机制和残差机制（第5.4节）、学习率及束大小，其他所有参数均与英德基础翻译模型保持一致。在推理阶段，我们

表4：Transformer在英语成分句法分析中的应用效果良好（结果见《华尔街日报》第23节）

画笔	训练	WSJ 23 F1
Vinyals & Kaiser等（2014）[37]	仅限《华尔街日报》，歧视性	88.3
Petrov等（2006）[29]	仅限《华尔街日报》，歧视性	90.4
Zhu等（2013）[40]	仅限《华尔街日报》，歧视性	90.4
Dyer等（2016）[8]	仅限《华尔街日报》，歧视性	91.7
变压器（4层）	仅限《华尔街日报》，歧视性	91.3
朱等（2013）[40]	半监督	91.3
黄与哈珀（2009）[14]	半监督	91.3
麦克洛斯基等（2006）[26]	半监督	92.1
维纳尔斯与凯撒等（2014）[37]	半监督	92.1
变压器（4层）	半监督	92.7
龙等（2015）[23]	多任务	93.0
戴尔等（2016）[8]	能生产的	93.3

我们将最大输出长度提升至输入长度+ 300。在仅使用WSJ数据集和半监督设置的场景下，我们采用了21的光束尺寸和α = 0 .3的参数配置。

表4中的结果显示，尽管我们的模型缺乏特定任务的调整，但其表现却出人意料地好，比所有先前报道的模型都取得了更好的结果，除了循环神经网络语法[8]。

与RNN序列到序列模型[37]相比，Transformer即使仅在包含4万句的WSJ训练集上进行训练，其性能也优于BerkeleyParser [29]。

7 结论

在本工作中，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，用多头自注意力替换了编码器-解码器架构中最常用的循环层。

在翻译任务中，Transformer的训练速度显著快于基于循环或卷积层的架构。在WMT 2014英德和英法翻译任务中，我们取得了新的最佳成绩。在英德任务中，我们的最佳模型甚至超越了所有先前报告的集成模型。

我们对基于注意力机制的模型的未来充满期待，并计划将其应用于其他任务。我们打算将Transformer扩展到处理非文本输入和输出的问题，并研究局部、受限注意力机制，以高效处理如图像、音频和视频等大型输入和输出。减少生成过程的顺序性也是我们的研究目标之一。我们用于训练和评估模型的代码可在https://github.com/ tensorflow/tensor2tensor获取。

致谢我们感谢Nal Kalchbrenner和Stephan Gouws的有益评论、纠正和启发。

参考文献

[1] Jimmy Lei Ba、Jamie Ryan Kiros和Geoffrey E Hinton.层归一化. arXiv预印本arXiv：1607.06450,2016.

[2] Dzmitry Bahdanau、Kyunghyun Cho和Yoshua Bengio.通过联合学习对齐与翻译实现神经机器翻译. CoRR，abs/1409.0473,2014.

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

[4]程建鹏、李东和米雷拉·拉帕塔.长短期记忆网络在机器阅读中的应用. arXiv预印本arXiv：1601.06733,2016.

[5]乔坤贤、巴特·范·梅里恩博尔、卡格拉尔·古尔切赫雷、费蒂·布加雷斯、霍尔格·施文克和约书亚·本吉奥。《利用rnn编码器-解码器学习短语表示以实现统计机器翻译》。CoRR，abs/1406.1078,2014。

[6]弗朗索瓦·肖莱特.《Xception：基于深度可分离卷积的深度学习》. arXiv预印本arXiv：1610.02357,2016.

[7]钟俊英、居尔切赫雷·卡格拉尔、赵京贤和本吉奥·约书亚。《门控循环神经网络在序列建模中的实证评估》。CoRR，abs/1412.3555,2014。

[8]克里斯·戴尔、阿迪古纳·昆科罗、米格尔·巴列斯特罗斯和诺亚·A·史密斯。《循环神经网络语法》。载于《NAACL会议论文集》，2016年。

[9]乔纳斯·格林、迈克尔·奥利、大卫·格朗热、丹尼斯·亚拉茨和扬·N·多芬。卷积序列到序列学习。arXiv预印本arXiv：1705.03122v2,2017年。

[10] Alex Graves.使用循环神经网络生成序列。arXiv预印本arXiv：1308.0850,2013年。

[11]韩明何、张翔宇、任少青、孙健.深度残差学习在图像识别中的应用.收录于《IEEE计算机视觉与模式识别会议论文集》，第770-778页，2016年。

[12]塞普·霍赫赖特、约书亚·本吉奥、保罗·弗拉斯科尼和尤尔根·施密德胡贝尔。《递归网络中的梯度流：学习长期依赖的难度》，2001年。

[13]塞普·霍赫赖特和尤尔根·施密德胡伯。《长短期记忆》。《神经计算》，第9卷第8期：1735－1780页，1997年。

[14]黄中强和玛丽·哈珀。跨语言的自训练PCFG语法与潜在注释。收录于《2009年自然语言处理经验方法会议论文集》，第832至841页，ACL，2009年8月。

[15] Rafal Jozefowicz、Oriol Vinyals、Mike Schuster、Noam Shazeer和吴永辉。探索语言模型的极限。arXiv预印本arXiv：1602.02410,2016。

[16]卢卡斯·凯泽和萨米·本吉奥。主动记忆能否替代注意力？载于《神经信息处理系统进展》（NIPS），2016年。

[17]卢卡斯·凯泽和伊利亚·苏特克弗。神经GPU学习算法。收录于《国际学习表示会议》（ICLR），2016年。

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

[19] Yoon Kim、Carl Denton、Luong Hoang和Alexander M. Rush。结构化注意力网络。*在2017年国际学习表征会议上。*

[20] Diederik Kingma和Jimmy Ba. Adam：一种随机优化方法。发表于《ICLR》2015年。

[21]奥列克西·库恰耶夫和鲍里斯·金斯堡。LSTM网络的因子分解技巧。arXiv预印本arXiv：1703.10722,2017年。

[22]周汉林、冯敏伟、西塞罗·诺盖拉·多斯桑托斯、莫宇、邢冰、周博文和约书亚·本吉奥.结构化自注意力句嵌入. arXiv预印本arXiv：1703.03130,2017.

[23]麦明-唐·龙，国V。李、伊利亚·苏茨克弗、奥里奥尔·维尼亚尔斯和卢卡什·凯撒。多任务序列到序列学习。arXiv预印本arXiv：1511.06114,2015。

[24]郑明唐、黄辉和克里斯托弗·D·曼宁.基于注意力机制的神经机器翻译的有效方法.arXiv预印本arXiv：1508.04025,2015.

[25]米切尔·P·马库斯、玛丽·安·马辛凯维奇和比阿特丽斯·桑托里尼。构建大型英语标注语料库：宾夕法尼亚树库。《计算语言学》，19(2)：313–330,1993年。

[26]大卫·麦考斯基、尤金·查尼阿克和马克·约翰逊.有效的自训练解析方法.收录于《美国语言学协会人类语言技术会议论文集》主会场，第152-159页.ACL，2006年6月。

[27]安库尔·帕里克、奥斯卡·塔克斯特罗姆、迪潘詹·达斯和雅各布·乌斯科雷特。《可分解注意力模型》。收录于《自然语言处理中的实证方法》，2016年。

[28]罗曼·保罗斯、熊凯明和理查德·索彻。一种用于抽象摘要的深度强化模型。arXiv预印本arXiv：1705.04304,2017年。

[29]斯拉夫·彼得罗夫、莱昂·巴雷特、罗曼·蒂博和丹·克莱因。《学习准确、紧凑且可解释的树注释》。收录于第21届国际计算语言学会议及第44届ACL年会论文集，第433-440页。ACL，2006年7月。

[30] Ofir Press和Lior Wolf.利用输出嵌入提升语言模型性能。arXiv预印本arXiv：1608.05859,2016。

[31] Rico Sennrich，Barry Haddow和Alexandra Birch。基于子词单元的稀有词汇神经机器翻译。arXiv预印本arXiv：1508.07909,2015年。

[32]诺亚·沙泽尔、阿扎莉亚·米尔霍塞尼、克日什托夫·马齐亚兹、安迪·戴维斯、郭乐、杰弗里·欣顿和杰夫·迪恩。《异常庞大的神经网络：稀疏门控的专家混合层》。arXiv预印本arXiv：1701.06538,2017年。

[33]尼提什·斯里瓦斯塔瓦、杰弗里·E·欣顿、亚历克斯·克里热夫斯基、伊利亚·苏特克韦尔和鲁斯兰·萨拉赫丁诺夫。《Dropout：一种简单防止神经网络过拟合的方法》。《机器学习研究杂志》，15(1)：1929–1958,2014年。

[34]萨因巴亚尔·苏赫巴塔尔、阿瑟·斯拉姆、杰森·韦斯顿和罗布·弗格斯。端到端记忆网络。收录于C.科特斯、N. D.劳伦斯、D. D.李、M.苏吉亚马和R.加内特编著的《神经信息处理系统第28届会议论文集》，第2440-2448页，Curran Associates，Inc.，2015年。

[35]伊利亚·苏茨克弗、奥里奥尔·维尼亚尔斯和国VV·李.神经网络的序列到序列学习.收录于《神经信息处理系统进展》第3104-3112页，2014年。

[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》.收录于《神经信息处理系统进展》，2015年。

[38] Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, 谷歌的神经机器翻译系统：弥合人类与机器翻译之间的差距.arXiv预印本arXiv:1609.08144,2016年.

[39]周杰、曹颖、王旭光、李鹏和徐伟.带有快速前向连接的深度循环模型在神经机器翻译中的应用. CoRR，abs/1606.04199,2016.

[40]朱木华、张月、陈文良、张敏和朱景波.快速准确的移位减少成分分析.收录于《ACL第51届年会论文集》（第一卷：长篇论文），第434-443页，ACL，2013年8月。