注意力就是你所需要的一切

|  |  |
| --- | --- |
| 阿希什·瓦斯瓦尼∗  谷歌大脑 avaswani@google.com | 诺姆· 沙泽尔∗ 尼基·帕尔马∗ 雅各布 ·乌什科雷特∗  谷歌大脑谷歌研究谷歌研究 noam@google.com nikip@google.com usz@google.com |
| 狮子 琼斯∗  谷歌研究 llion@google.com | 艾丹·戈麦斯∗ † 武卡什 ·凯撒∗  多伦多大学谷歌大脑 aidan@cs.toronto.edu lukaszkaiser@google.com |

伊利亚· 波洛苏欣∗ ‡ illia.polosukhin@gmail.com

# 摘要

占主导地位的序列转导模型基于复杂的反复或卷积神经网络，包括编码器和解码器。性能最佳的模型还通过注意力机制连接编码器和解码器。我们提出了一个新的简单网络架构，外变网络模型，完全基于注意力机制，完全省去了重复和卷积。对两个机器翻译任务的实验表明，这些模型在质量上更胜一筹，同时更具并行化性和可重复性。训练时间明显减少。我们的模型在 2014统计机器翻译研讨会数据集 上实现了 28.4 双语评估替补准确率的英语-德语翻译任务，比现有的最佳结果（包括合奏）提高了 2 个以上蓝.在 2014统计机器翻译研讨会数据集 英语到法语翻译任务中，我们的了模型在八个GPU上训练了3.5天后，建立了一个新的单一模型最先进的双语评估替补准确率分数41.8，这是文献中最佳模型训练成本的一小部分。我们证明了外变网络模型可以通过成功地将其应用于英语成分分析很好地推广到其他任务，无论是大规模还是有限的训练数据。

arXiv：1706.03762v5 [。.cs。 CL] 2017年12月6日

# 介绍

反复神经网络，特别是长短期记忆[13]和门控反复[7]神经网络，已经牢固地确立为 序列模型和

∗

同等贡献。上市顺序是随机的。雅各布提议用自我注意代替反复神经网络，并开始努力评估这个想法。Ashish与Illia一起设计并实施了第一个变形金刚模型，并积极参与了这项工作的各个方面。Noam提出了缩放的点积注意力，多头注意力和无参数位置表示，并成为几乎每个细节都参与的另一个人。Niki在我们原始代码库和tensor2tensor中设计，实现，调整和评估了无数的模式l变体。Llion还尝试了新的模型变体，负责我们最初的代码库，以及高效的推理和可视化。Lukasz 和 Aidan 花了无数天的时间设计各种标准并实现 tensor2tensor，取代了我们早期的代码库，极大地改善了结果并大大加快了我们的研究速度。 †

在Google Brain工作期间完成的工作。

‡

在谷歌研究院期间完成的工作。

第31届神经信息处理系统会议（2017 NIPS），美国加利福尼亚州长滩。

语言模型和机器翻译等转导问题的最先进的方法[35，2，5]。此后，人们不断努力突破反复语言模型和编码器-解码器架构的界限[38，24，15]。

反复模型通常沿输入和输出序列的符号位置考虑计算。将位置与计算时间的步骤对齐，它们生成一系列隐藏状态 h t，作为先前隐藏状态 *ht*−1 和位置 *t* 输入的函数。这种固有的顺序性质排除了训练示例中的并行化，这在较长的序列长度下变得至关重要，因为内存限制约束了跨批处理示例。最近的工作通过因式分解技巧[21]和条件计算[32]在计算效率方面取得了显着的提高，同时也提高了后者的模型性能。然而，序列计算的基本约束仍然存在。

在各种任务中，注意力机制已成为引人注目的序列模型和转导模型的一个组成部分，允许不考虑他们在输入和输出之间距离依赖的模型 [2，19]。然而，除了少数情况[27]外，在所有情况下，这种注意力机制都与反复网络结合使用。

在这项工作中，我们提出了外变网络模型，这是一种模型架构，避免重复，而是完全依靠一种机制来绘制输入和输出之间的全局依赖关系。外变网络模型 允许更多的并行化，并且在八个 P100 GPU 上训练短短 12 小时后，翻译质量达到了一个新的水平。

# 背景

减少顺序计算的目标也构成了扩展神经图形处理单元[16]，字节网络 [18]和卷积序列到序列模型 [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随着位置之间的距离而增长，对于卷积序列到序列模型呈线性增长，对于 字节网络呈对数增长。这使得学习距离远的位置之间的依赖关系变得更加困难[12]。在外变网络模型中，这被减少到恒定数量的操作，尽管代价是由于平均注意力权重d位置而降低了有效分辨率，我们用多头注意力抵消了这种影响，如第3.2节所述。

自我注意，有时称为内注意，是一种将单个序列的不同位置联系起来的注意力机制，以便计算序列的表示 。自我注意已成功用于各种任务，包括阅读理解，摘要总结，文本蕴涵和学习任务-独立的句子表示[4，27，28，22]。

端到端记忆网络 基于反复注意力机制，而不是 序列对齐 的复发，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，外变网络模型 是第一个完全依靠自我注意力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 反复神经网络 或卷积。在以下部分中，我们将描述外变网络模型，激发自我注意力并讨论 其相对于[17，18]和[9]等模型的优势。

# 模型架构

大多数竞争性神经序列转导模型都具有编码器-解码器结构[5，2，35]。在这里，编码器映射符号表示的输入序列 （*x*1*,...，x* n） 到连续表示序列 z = （z 1*,...,***z** *n*）。给定 **z**，解码器然后生成输出序列 （*y*1*,...，ym*） 的符号，一次一个元素。在每一步中，模型都是自回归的 [10]，在生成下一个时将先前生成的符号用作附加输入。

外变网络模型 遵循这一整体架构，编码器和解码器使用堆叠的自身注意力层和 逐点全连接层，分别如图 1 的左半部分和右半部分所示。

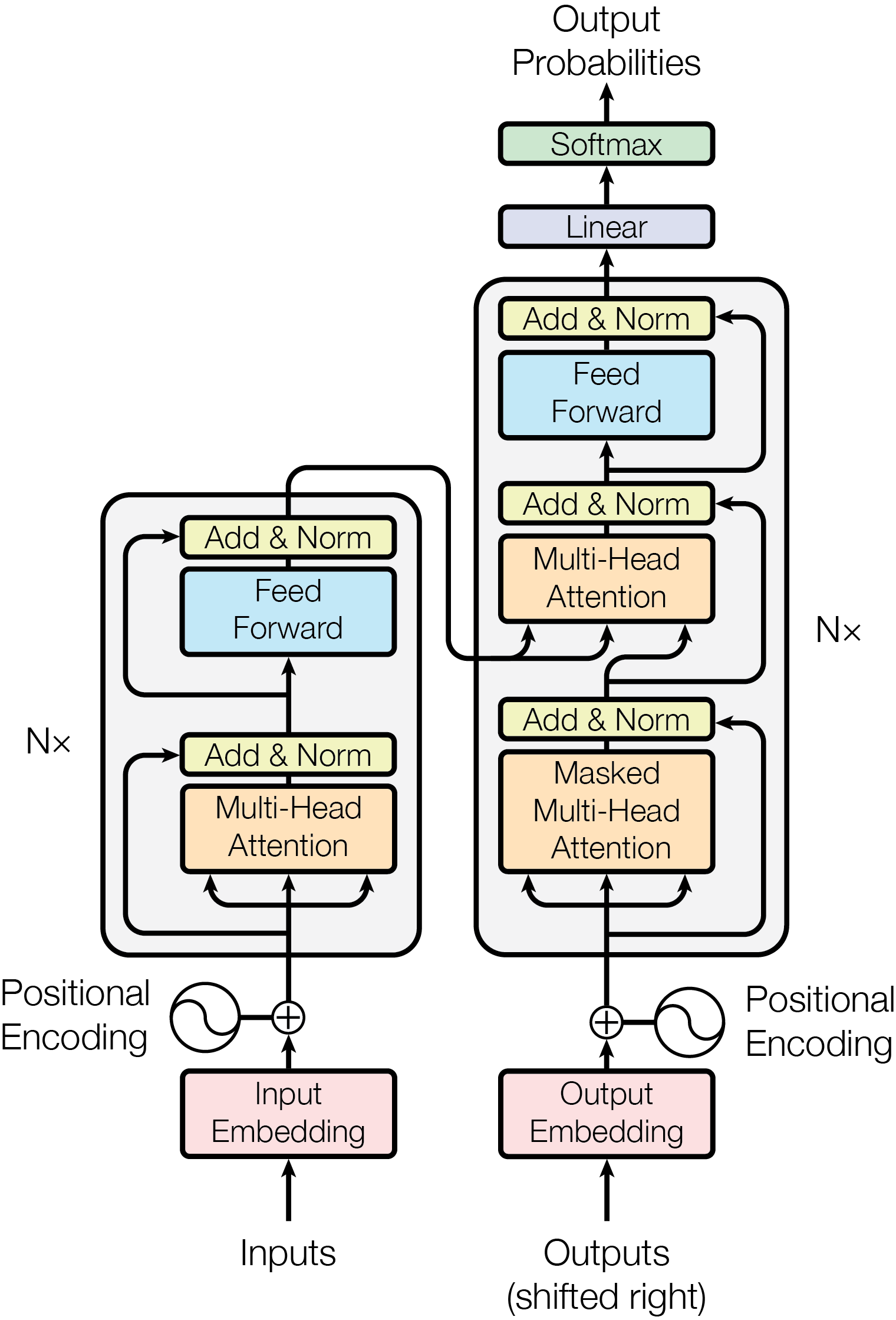


图1：外变 - 模型架构。

3.1 编码器和解码器堆栈

编码器：编码器由 *N* = 6 个相同层的堆叠组成。每个图层有两个子图层。第一种是多头自注意力机制，第二种是简单的、按位置的全连接前馈网络。我们在两个子层中的每一个周围都使用残差连接 [11]，通过层归一化 [1] 。也就是说，每个子层的输出是 LayerNorm（x + Sublayer（x）），其中 Sublayer（*x*） 是子层本身实现的功能。为了促进这些残差连接，模型中的所有子层以及嵌入层都会产生维度 *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外，解码器还插入第三个子层，该子层对编码器堆栈的输出执行多头注意。与编码器类似，我们在子层的每个周围使用残差连接，然后进行层归一化。我们还修改了解码器堆栈中的自注意子层，以防止位置关注后续位置。这种掩码，再加上输出嵌入被一个 position 偏移的事实，确保对位置 i 的预测只能依赖于位置小于 i 的已知输出。

3.2 注意力

注意力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、变量和输出都是向量。输出计算为值的加权总和，其中分配给每个值的权重由查询与相应键的兼容性函数计算。

缩放点积注意力 多头

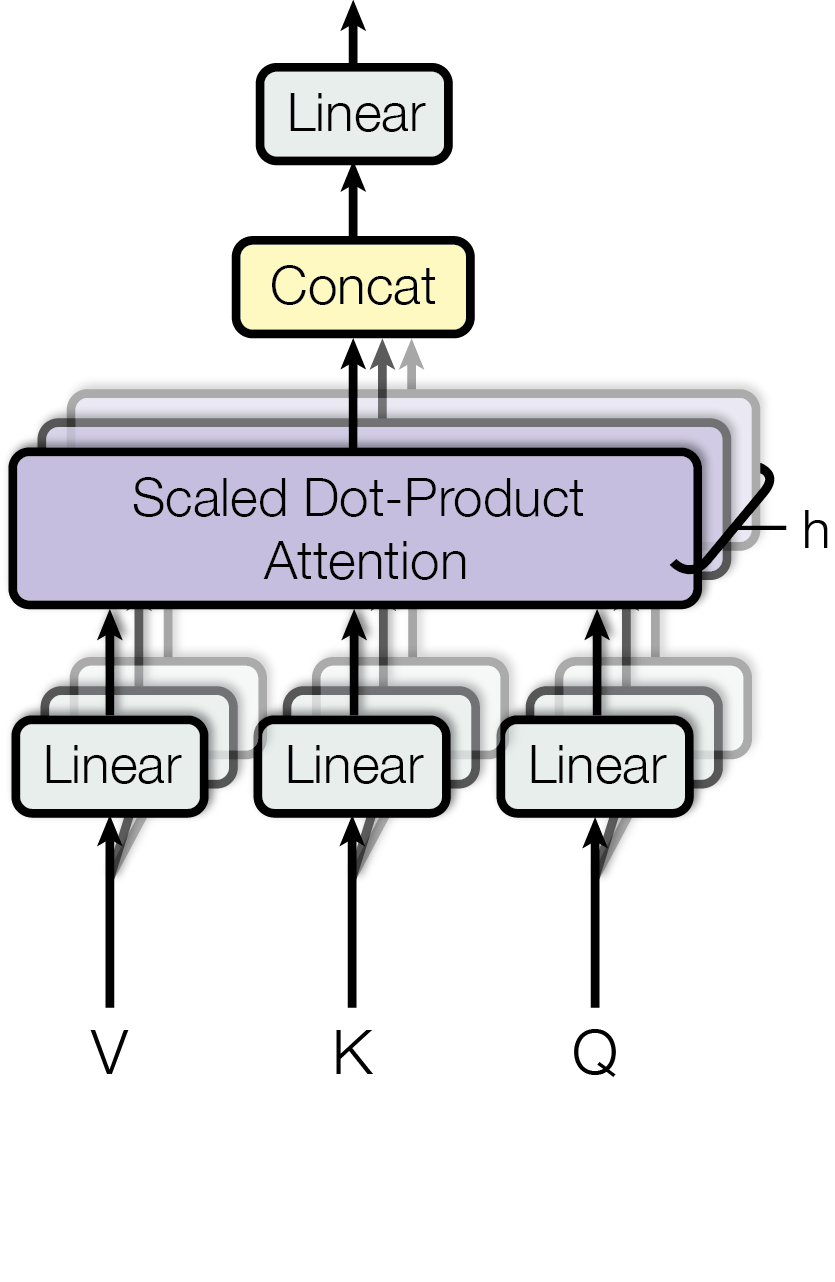
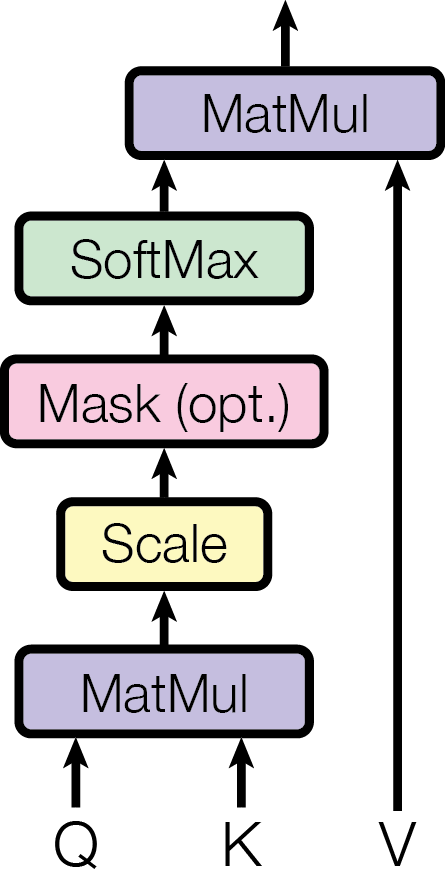
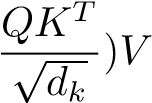


图 2：（左）缩放的点积注意力。（右）多头注意力由几个并行运行的注意力层组成。

3.2.1 缩放的点积注意力

我们将特别注意称为“缩放的点积注意力”（图2）。输入由维度 *dk* 的查询和键以及维度 *dv* 的值组成。我们计算查询与所有键的点积，将每个键除以 根号*dK*，然后应用 软最大函数来获取值的权重。

在实践中，我们同时计算一组查询的注意力函数，打包成一个矩阵 *Q*。键和值也打包到矩阵 *K* 和 *V* 中。 我们将矩阵输出计算为：

注意力（*Q，K，V*  ） = 软最大（ （1）

两个 最常用的 注意力函数是加法注意力[2]和点积（乘法）注意力。点积注意力与我们的算法相同，除了比例因子。 加法注意力使用具有单个隐藏层的前馈网络计算兼容性函数。虽然两者在理论复杂性上相似，但点积注意力在实践中更快，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。

虽然对于*较小的dK*值，这两种机制的表现相似，但加法注意力优于点积注意力，而无需缩放较大的*dk*值[3]。我们怀疑，对于 *dk* 大的值，点积的大小变大，将 软最大 函数推到梯度极小的区域。 [[1]](#footnote-1)为了抵消这种影响，我们缩放了点积。

3.2.2 多头注意力

我们发现，与使用 d 模型维度键、值和查询执行单个注意力函数不同，将查询、键和值 *h* 次线性投影到 d k、 d *k* 和 *dv*  维度是有益的，有效地。在查询、键和值的每个投影版本上，我们并行执行注意力函数，产生 *dv* 维输出值。它们被连接起来并再次投影，产生最终值，如图 2 所示。

多头注意力允许模型共同关注来自不同表示子空间在不同位置的信息。对于单个注意力头，平均会抑制这一点。

多头（*Q，K，V*  ） = Concat（头 1*,...,*头 h）*WO* 其中头i = 注意力（

其中投影是参数矩阵和 .

在这项工作中，我们使用了*h* = 8个平行的注意力层或头部。对于其中的每一个，我们使用。由于每个头部的维数减小，总计算成本与全维单头注意力相似。

3.2.3 衰减在我们的模型中的应用

外变网络模型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，内存键和值来自编码器的输出。这允许解码器中的每个位置覆盖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-解码器注意力机制，例如 [38， 2， 9]。
* 编码器包含自我注意层。在自我注意层中，所有键、值和查询都来自同一个位置，在本例中为编码器中前一层的输出。编码器中的每个位置都可以关注编码器前一层中的所有位置。
* 同样，解码器中的自注意层允许解码器中的每个位置关注解码器中的所有位置，包括该位置。我们需要防止解码器中的信息向左流动，以保留自回归属性。我们通过屏蔽（设置为 −∞）softmax 输入中对应于 非法连接的所有值来暗示这一点。参见图2。
  1. 位置前馈网络

除了注意力子层外，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，该网络分别以相同的方式应用于每个位置。这包括两个线性变换，中间有一个 纠正的线性单元（ReLU） 激活。

FFN（*x*） = 最大值（*0，XW* 1 + b 1）*W* 2 + *b* 2 （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。另一种描述方式是两个内核大小为 1 的卷积。输入和输出的维数为 d 模型 = 512，内层的维数*为 dff* = 2048。

* 1. 嵌入层和 软最大值

与其他序列转导模型类似，我们使用学习嵌入层将输入文字和输出文字转换为*维度 d*模型的向量。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。

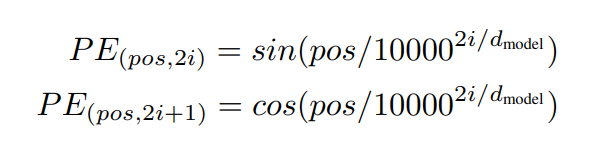
* 1. 位置编码

由于我们的模型不包含反复和卷积，为了使模型利用序列的顺序，我们必须注入有关表 1 的相对或绝对位置的信息：最大路径长度、每层复杂性和不同层类型的最小顺序操作数。 *n*  是序列长度，*d*  是表示维度，k 是卷积的内核大小，*r* 是受限自注意力中邻域的大小。

|  |  |  |  |
| --- | --- | --- | --- |
| 图层类型 | 每层复杂性 | 顺序操作 | 最大路径长度 |
| 自注意力 | *O*（*n*2 · *d*） | *O*（1） | *O*（1） |
| 反复的 | *O*（*n* · *d*2） | *O*（*n*） | *O*（*n*） |
| 卷 积 | *O*（*k* · *n* · *d*2） | *O*（1） | *O*（*logk*（*n*）） |
| 自注意力（受限） | *O*（*r* · *n* · *d*） | *O*（1） | *O*（*n/r）* |

序列中的标记。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。位置编码与嵌入具有相同的维度 *d*模型 ，因此可以将两者相加。 位置编码有很多选择，学习和固定[9]。

在这项工作中，我们使用不同频率的正弦和余弦函数：

**

其中 *POS*  是位置，*i* 是维度。也就是说，位置编码的每个维度对应于一个正弦曲线。波长形成从2*π*到10000·2*π*的几何级数。我们之所以选择这个函数，是因为我们假设它允许模型轻松学习相对位置，因为对于任何固定偏移 *k*，*PEpos*+*k* 可以表示为 *PEpos* 的线性函数。

我们还尝试使用学习的位置嵌入层[9] ， 发现这两个版本产生了完全相同的结果（见表3行（E））。我们选择了正弦版本，因为它可能允许模型推断出比训练期间遇到的序列长度更长的序列长度。

# 为什么要自身注意力

在本节中，我们将自我注意层的各个方面与通常用于映射一个可变长度符号表示序列的反复层和卷积层进行比较。（*x*1*,...，*x n） 到另一个长度相等的序列 （z 1*,...,*z *n*），*x* i，*z* *i* ∈ R*d，*例如典型序列转导编码器或解码器中的隐藏层。为了激励我们使用自我注意力，我们考虑了三个目标。

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，由所需的最小顺序操作数来衡量。

第三个是网络中长期依赖关系之间的路径长度。学习长度依赖性是许多序列转导任务中的关键挑战。影响学习这种依赖关系能力的一个关键因素是网络前向和后向信号必须遍历的路径长度。输入和输出序列中任意位置组合之间的路径越短，就越容易学习长期依赖关系[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，自我注意力层将所有正值与恒定数量的顺序执行操作连接起来，而反复层需要*O*（*n*）顺序操作。在计算复杂度方面，当序列长度*n*小于表示维*数d*时，自我注意力层比反复层更快，这通常是机器翻译中最先进的模型使用的句子表示，例如单词片段[38]和字节对[31]表示。为了提高涉及超长序列的任务的计算性能，可以将自我注意力限制在仅考虑 以相应输出位置为中心的输入序列中大小为r的邻域。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。

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。这样做在连续核的情况下需要一堆 O（*n/k*）卷积层，或者在扩张卷积的情况下需要*O*（*logk*（*n*））[18]，从而增加网络中任何两个位置之间最长路径的长度。卷积层通常比反复层贵，是k的倍数。然而，可分离卷积[6]大大降低了复杂度，达到*O*（*k*· *n* ·*d* + *n* · *d*2）。然而，即使 *k* = *n*，可分离卷积的复杂性也等于自我注意层和逐点前馈层的组合，这是我们在模型中采用的方法。

作为附带好处，自我注意力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，并在附录中介绍和讨论示例。不仅个人注意力头清楚地学会了执行不同的任务， 而且许多 注意力头似乎表现出与句子的句法和语义结构相关的行为。

# 训练

本节介绍模型的训练制度。

5.1 训练数据和批处理

我们使用标准的 2014统计机器翻译研讨会数据集 英语-德语数据集进行训练，该数据集由大约 450 万个句子对组成。句子使用字节对编码[3]进行编码，该编码具有 约37000个标记的共享源目标词汇表。对于英语-法语，我们使用了由36M个句子组成的更大的2014统计机器翻译研讨会数据集 English-French数据集，并将标记拆分为32000个单词的词汇表[38]。句子对按近似序列长度批处理在一起。每个训练批次包含一组句子对，其中包含大约 25000 个酸ce 令牌和 25000 个目标令牌。

5.2 硬件和时间表

我們在一台具有 8 個 NVIDIA P100 GPU 的機器上訓練了模型。对于使用整篇论文中描述的超参数的基本模型，每个训练步骤大约需要 0.4 秒。我们对 基本模型进行了总共 100，000 步或 12 小时的训练。对于我们的大型 模型（如表 3 的底行所述），步进时间为 1.0 秒。大型模型经过300，000步（3.5天）的训练。

5.3 优化器

我们使用了 Adam 优化器 [20]。我们根据以下公式在训练过程中改变了学习率：

*lrate* = *d*−model0*。*5 ·min（*step*\_*num*−0*.*5*，步数*· *预热*步骤−1*.*5） （3）

这对应于第一个 *warmup*\_*steps* 训练步骤的学习率线性增加，然后按比例降低它与步数的平方根反比。我们使用 *warmup*\_*steps* = 4000。

5.4 正则化

我们在训练期间采用三种类型的正则化：

残余辍学 我们将 dropout [33] 应用于每个子层的输出，然后再将其添加到子层输入并进行归一化。此外，我们将 dropout 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 的速率*。*1.

表 2：在英语到德语和英语到法语的 newstest2014 测试中，外变网络模型 在双语评估替补准确率分数方面比以前的最先进模型更高，而培训成本只是其中的一小部分。

字节网[18]

型

蓝

培训 成本 （浮点数）

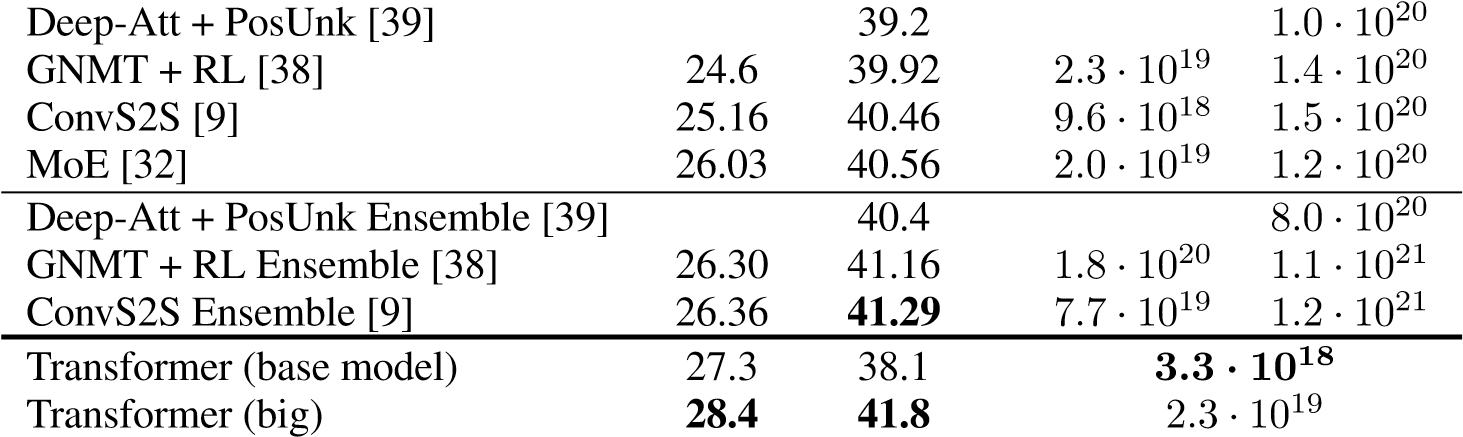
EN-DE

英语-法语

EN-DE

英语-法语

23.75



标签平滑 在训练过程中，我们采用了值的标签平滑 [36]。这伤害了困惑，因为模型学会了更加不确定，但提高了准确性和双语评估替补准确率分数。

# 结果

6.1 机器 翻译

在 2014统计机器翻译研讨会数据集 英语到德语翻译任务中，大变压器模型（表 2 中的变压器（大））的性能比之前报告的最佳模型（包括集成）高出 2 个以上*。*0 双语评估替补准确率，建立了新的最先进的双语评估替补准确率核心28*。*4.表 3 的底行列出了此模型的配置。培训花了3*。*在 8 个 P100 GPU 上使用 5 天。甚至我们的基本模型也超过了所有以前发布的模型和集成，而训练成本只是任何竞争模型的一小部分。

在 2014统计机器翻译研讨会数据集 英语到法语翻译任务中，我们的大模型获得了 41 分的双语评估替补准确率*分。*0，优于 之前发布的所有单一模型，训练成本不到以前最先进的模型的 1*/*4。针对英语到法语训练的 Tra nsformer （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，光束尺寸为 4，长度惩罚 *α* = 0*。*6[38]。这些超参数是在开发集上进行实验后选择的。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。我们通过乘以训练时间、使用的 GPU 数量以及每个 GPU 的持续单精度浮点容量的估计来估计用于训练模型的浮点运算数[[2]](#footnote-2)。

6.2 模型变化

为了评估变形金刚不同组件的重要性，我们以不同的方式改变了我们的基本模型，在开发集newstest2013上测量了英语到德语翻译的性能。我们使用了上一节所述的波束搜索，但没有检查点平均。我们在表3中给出了这些结果。

在表 3 行 （A） 中，我们改变注意力头的数量以及注意力键和值维度，保持计算量恒定，如第 3.2.2 节所述。虽然单头注意力比最佳设置差0.9 双语评估替补准确率，但质量也会随着头部过多而下降。

表 3：变压器架构的变化。未列出的值与基本模型的值相同。所有指标均在英语到德语翻译开发集 newstest2013 上。根据我们的字节对 e ncoding，列出的困惑是每个单词的，不应与每个单词的困惑进行比较。

*d*

*N*

型

*d*

呵呵

*d*

*h*

*k*

*d*

*在*

*P*

*落*

*ls*

火车

蓝

个人利益

参数

步骤

开发

)

(

开发

)

(

×

10

6

基础

0.1

100

K

64

0.1

64

8

2048

512

6

25.8

65

4.92

一个

)

(

1

512

512

24.9

5.29

128

4

128

25.5

5.00

32

16

32

4.91

25.8

16

32

16

25.4

5.01

(

B

)

16

25.1

58

5.16

32

25.4

60

5.01

(

C

)

2

6.11

36

23.7

4

25.3

50

5.19

8

25.5

80

4.88

256

32

32

28

5.75

24.5

128

128

1024

168

26.0

4.66

1024

53

5.12

25.4

4096

26.2

4.75

90

D

(

)

0.0

24.6

5.77

0.2

4.95

25.5

0.0

4.67

25.3

0.2

5.47

25.7

)

(

和

位置 嵌入 而不是 正弦波

4.92

25.7

大

4096

0.3

300

K

6

1024

16

4.33

26.4

213

表 4：转换器很好地推广到英语选区解析（结果见 《华尔街日报》第 23 节）

|  |  |  |
| --- | --- | --- |
| 解析 器 | 训练 | 华尔街日报 23 F1 |
| Vinyals & Kaiser el al. （2014） [37] | 仅限《华尔街日报》，歧视性 | 88.3 |
| 彼得罗夫等人 （2006） [29] | 仅限《华尔街日报》，歧视性 | 90.4 |
| 朱等人 （2013） [40] | 仅限《华尔街日报》，歧视性 | 90.4 |
| 戴尔等人 （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 |
| 朱等人 （2013） [40] | 半监督 | 91.3 |
| 黄与哈珀 （2009） [14] | 半监督 | 91.3 |
| 麦克洛斯基 等人 （2006） [26] | 半监督 | 92.1 |
| 葡萄园 和凯撒 。 （2014年）[37] | 半监督 | 92.1 |
| 变压器（4层） | 半监督 | 92.7 |
| Luong et al. （2015） [23] | 多任务 | 93.0 |
| 戴尔等人 （2016） [8] | 生成 | 93.3 |

在表 3 行 （B） 中，我们观察到减小注意力键大小 *dk* 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函数可能是有益的。我们进一步观察到（C） 和 （D） 行中的 e，正如预期的那样，更大的模型更好，而 dropout 对于避免过度拟合非常有帮助。在第 （E） 行中，我们将正弦位置编码替换为学习的位置嵌入 [9]，并观察到基本模型几乎相同的结果。

6.3 英语选区解析

为了评估转换器是否可以推广到其他任务，我们对英语选区解析进行了实验。这项任务提出了特定的挑战：输出受到强烈的结构应变的影响，并且明显比输入长。此外，反复神经网络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]。

我们在宾夕法尼亚树银行的华尔街日报（WSJ）部分训练了一个4层变压器，*d模型*= 1024[25]，大约40K训练句子。我们还在半监督环境中对其进行了训练，使用了来自大约1700万个句子的更大的高置信度和BerkleyParser语料库[37]。对于仅《华尔街日报》设置，我们使用 16K 的词汇表来表示 kens，在半监督设置中使用了 32K 个令牌的词汇表。

我们只进行了少量的实验来选择辍学，包括注意力和残余

（第 5.4 节），学习率和波束大小 在第 22 节开发集上，所有其他参数与英语到德语基础翻译模型保持不变。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d，光束尺寸为 21 ， *α* = 0*。*3 仅适用于《华尔街日报》和半监督设置。

我们在表4中的结果表明，尽管缺乏特定于任务的调优，但我们的模型表现出奇地好，比以前报告的所有模型都产生了更好的结果，但递归神经网络语法除外[8]。

与反复神经网络序列到序列模型[37]相比，即使仅在40K句子的WSJ训练集上进行训练，外变网络模型的性能也优于BerkeleyParser[29]。

# 结论

在这篇文章中，我们介绍了外变网络模型，这是第一个完全基于注意力的序列转导模型，用多头自注意力取代了编码器-解码器架构中最常用的反复层。

对于转换任务，转换器的 训练速度明显快于基于反复层或卷积层的架构。在 2014统计机器翻译研讨会数据集 英语到德语和 2014统计机器翻译研讨会数据集 英语到法语翻译任务中，我们实现了全新的技术水平。在前一个任务中，我们的最佳模型甚至执行所有先前报告的集成。

我们对基于注意力的模型的未来感到兴奋，并计划将其应用于其他任务。我们计划将转换器扩展到涉及文本以外的输入和输出模式的问题，并引入局部的、有限的注意力机制，以有效地处理图像、音频和视频等大型输入和输出。让世代不那么连续是我们的另一个研究目标。

我们用来训练和评估模型的代码可以在 [https://github.com/ tensorflow/tensor2tensor](https://github.com/tensorflow/tensor2tensor)中使用[。](https://github.com/tensorflow/tensor2tensor)

致谢 我们感谢Nal Kalchbrenner和Stephan Gouws富有成效的评论，更正和启发。

# 引用

1. Jimmy Lei Ba、Jamie Ryan Kiros和Geoffrey E Hinton。层规范化。 arXiv  *预印本*[*arXiv：1607.06450*，](http://arxiv.org/abs/1607.06450)2016。
2. Dzmitry Bahdanau，Kyunghyun Cho和Yoshua Bengio。通过共同学习对齐和翻译来实现神经机器翻译。*CoRR*， abs/1409.0473， 2014.
3. Denny Britz， Anna Goldie， Minh-Thang Luong， and Quoc V. Le.对神经机器翻译离子架构的大规模探索。 *CoRR*， abs/1703.03906， 2017.
4. 程建鹏、李东和米雷拉·拉帕塔。用于机器读取的长短期记忆网络。 arXiv  *预印本*[*arXiv：1601.06733*，](http://arxiv.org/abs/1601.06733)2016。
5. 赵京贤、巴特·范·梅林博尔、卡格拉尔·古尔切尔、费蒂·布加雷斯、霍尔格·施温克和约书亚·本吉奥 。使用 反复神经网络 编码器-dec oder 学习短语表示，用于统计机器翻译。 *CoRR*， abs/1406.1078， 2014.
6. 弗朗索瓦·乔莱特。Xception：具有深度可分离卷积的深度学习。arXiv  *预印本*[*arXiv：1610.02357*，](http://arxiv.org/abs/1610.02357)2016。
7. 郑俊英、恰格拉尔·居尔切尔、赵庆贤和约书亚·本吉奥。门控递归神经网络在序列模型上的实证评估。*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·史密斯。递归神经网络语法。全国*协进会论文集*，2016年。
9. 乔纳斯·盖林、迈克尔·奥利、大卫·格兰吉尔、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：1705.03122v*](http://arxiv.org/abs/1705.03122)*2*，2017。
10. 亚历克斯·格雷夫斯。使用递归神经网络生成序列。arXiv  *预印本*[*arXiv：1308.0850*，](http://arxiv.org/abs/1308.0850)2013。
11. 何凯明， 张翔宇， 任少清， 和孙健. 用于图像识别的深度残差学习.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Jürgen Schmidhuber。反复网络中的梯度流：学习长期依赖关系的困难，2001年。
13. Sepp Hochreiter 和Jürgen Schmidhuber。长短期记忆。 *神经计算*， 9（8）：1735–1780， 1997.
14. 黄忠强和玛丽·哈珀。具有跨语言潜在注释的自我训练 PCFG 语法。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Yonghui Wu。探索语言建模的局限性。 arXiv  *预印本arXiv*[*：1602.02410*，](http://arxiv.org/abs/1602.02410)2016。
16. 乌卡什·凯撒和萨米·本吉奥。主动记忆可以代替注意力吗？神经*信息处理系统进展， （NIPS），* 2016.
17. Łukasz Kaiser 和 Ilya Sutskever。神经 GPU 学习算法。在*学习表征国际会议（ICLR）*上，2016年。
18. 纳尔·卡尔奇布伦纳、拉斯·埃斯佩霍尔特、凯伦·西蒙尼扬、亚伦·范登奥德、亚历克斯·格雷夫斯和科雷·卡武克库格鲁。线性时间的神经机器翻译。 arXiv  *预印本*[*arXiv：1610.10099v*](http://arxiv.org/abs/1610.10099)*2*，2017。
19. Yoon Kim、Carl Denton、Luong Hoang和Alexander M. Rush。结构化的注意力网络。在*学习表征国际会议*上，2017年。
20. 迪德里克·金玛和吉米·巴。亚当：一种优化的方法。在*ICLR*中，2015年。
21. 奥列克西·库恰耶夫和鲍里斯·金斯伯格。LSTM 网络的因式分解技巧。 arXiv  *预印本*[*arXiv：1703.10722*，](http://arxiv.org/abs/1703.10722)2017。
22. 林周涵、冯敏伟、西塞罗·诺盖拉·多斯桑托斯、莫宇、向冰、周博文和约书亚·本吉奥。结构化的自我注意句子嵌入。 arXiv  *预印本arXiv*[*：1703.03130*，](http://arxiv.org/abs/1703.03130)2017。
23. Minh-Thang Luong，Quoc V. Le，Ilya Sutskever，Oriol Vinyals，And Lukasz Kaiser。多任务序列到序列学习。 arXiv  *预印本*[*arXiv：1511.06114*，](http://arxiv.org/abs/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arXiv*[*：1508.04025*，](http://arxiv.org/abs/1508.04025)2015。
25. Mitchell P Marcus，Mary Ann Marcinkiewicz和Beatrice Santorini。建立一个大型带注释的英语语料库：宾夕法尼亚树库。 *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人类语言技术会议记录中，主会议，*第152-159页。ACL，2006 年 6 月。
27. 安库尔·帕里克、奥斯卡·塔克斯特伦、迪潘詹·达斯、雅各布·乌什科雷特。可分解注意力模型。自然*语言处理的经验方法*，2016。
28. 罗曼·保卢斯、熊彩明和理查德·索彻。用于抽象总结的深度强化模型。 arXiv  *预印本*[*arXiv：1705.04304*，](http://arxiv.org/abs/1705.04304)2017。
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、紧凑和可解释的树注释。第 *21 届计算语言学国际会议和第 44 届ACL 周年会议*论文集，第 433-440 页。ACL，2006年7月。
30. 奥菲尔出版社和利奥尔·沃尔夫。使用输出嵌入来改进语言模型。 arXiv  *预印本*[*arXiv：1608.05859*，](http://arxiv.org/abs/1608.05859)2016。
31. Rico Sennrich，Barry Haddow和Alexandra Birch。带有子词单位的生僻词的神经机器翻译 。 arXiv  *预印本*[*arXiv：1508.07909*，](http://arxiv.org/abs/1508.07909)2015。
32. Noam Shazeer、Azalia Mirhoseini、Krzysztof Maziarz、Andy Davis、Quoc Le、Geoffrey Hinton 和 Jeff Dean。 大得离谱的神经网络：稀疏的门控专家混合层。 arXiv  *预印本*[*arXiv：1701.06538*，](http://arxiv.org/abs/1701.06538)2017。
33. 尼蒂什·斯里瓦斯塔瓦、杰弗里·辛顿、亚历克斯·克里热夫斯基、伊利亚·苏茨凯弗和鲁斯兰·萨拉胡迪诺夫。Dropout：一种防止神经网络过度拟合的简单方法。*机器学习研究，* 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，N. D. Lawrence，D. D. Lee，M. Sugiyama和R. Garnett编辑，*神经信息处理系统进展*28，第2440-2448页。Curran Associates， Inc.， 2015.
35. Ilya Sutskever，Oriol Vinyals和Quoc VV Le。使用神经网络进行序列到序列学习。神经*信息处理系统进展*，第3104-3112页，2014年。
36. 克里斯蒂安·塞格迪、文森特·范豪克、谢尔盖·约菲、乔纳斯·舒伦斯和兹比格涅夫·沃伊纳。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， and Hinton.语法作为外语。神经*信息处理系统进展*，2015 年。
38. 吴永辉， 迈克·舒斯特， 陈志峰， 郭 V Le， 穆罕默德·诺鲁兹， 沃尔夫冈·马切里， 马克西姆·克里昆， 曹元， 高勤， 克劳斯·马切里， 等. 谷歌的神经机器翻译系统：弥合人工和机器翻译之间的差距。 *arXiv preprint* [*arXiv：1609.08144*，](http://arxiv.org/abs/1609.08144) 2016.
39. 周杰， 曹颖， 王旭光， 李鹏， 徐伟.Deep 具有快进连接的递归模型，用于神经机器翻译。 *CoRR*， abs/1606.04199， 2016.
40. 朱慕华， 张悦， 陈文亮， 张敏， 朱静波.快速准确的移位-减少成分解析。ACL第*51届年*会论文集*（第1卷：长论文），*第434-443页。ACL，2013 年 8 月。

# 注意力可视化



图 3：第 5 层（共 6 层）中编码器自注意中长距离依赖关系的注意力机制示例。许多头像关注动词“制造”的遥远依赖关系，完成了短语“制造......更难'。这里仅显示“制作”一词的注意事项。不同的颜色代表不同的头部。最好以彩色观看。

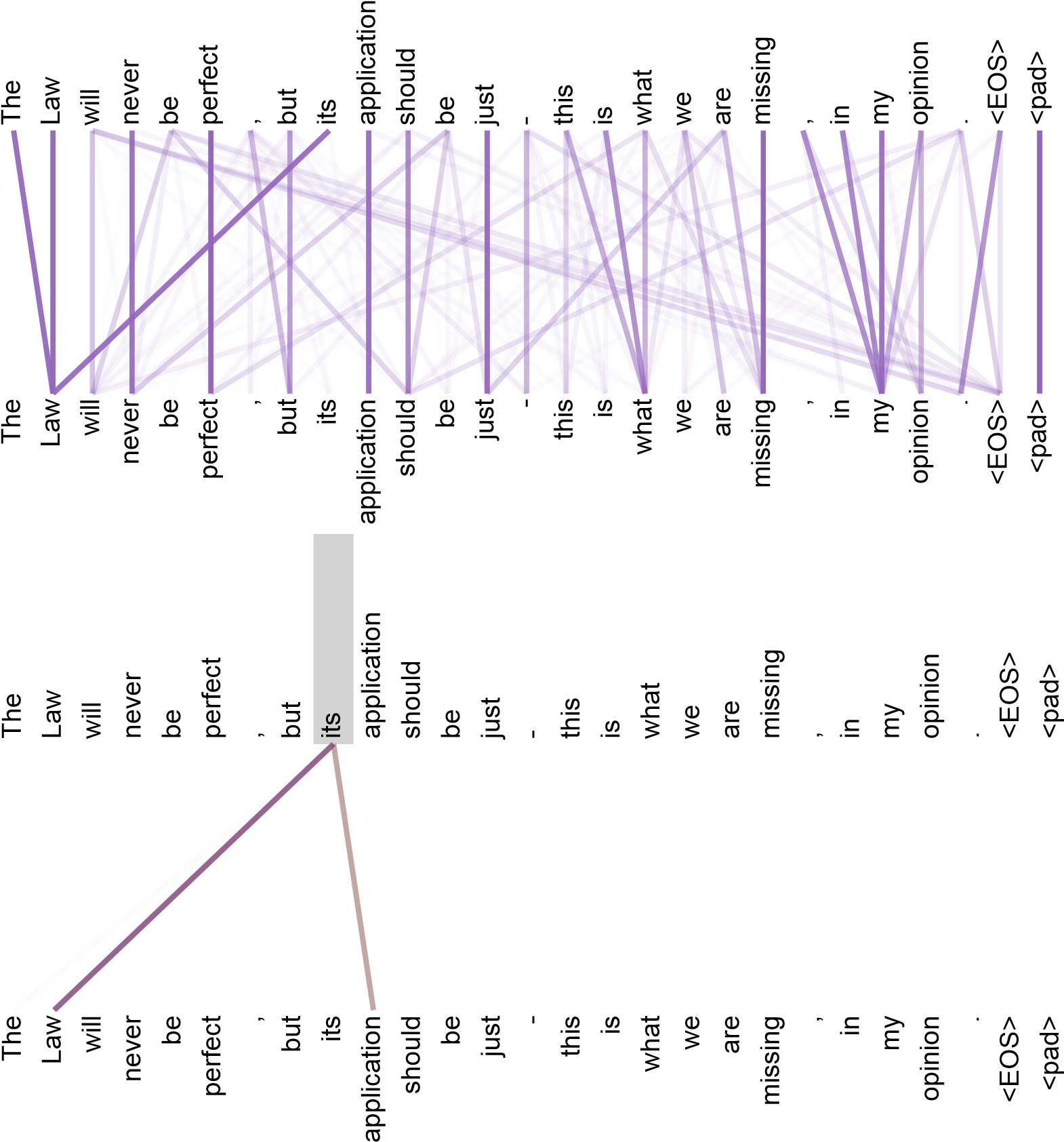


图 4：两个注意力 头，也在 6 层的第 5 层，显然参与了回溯解析。上图：全神贯注头部5。底部：仅从注意力头 5 和 6 的单词“its”中分离出注意力。请注意，这个词的注意力非常敏锐。

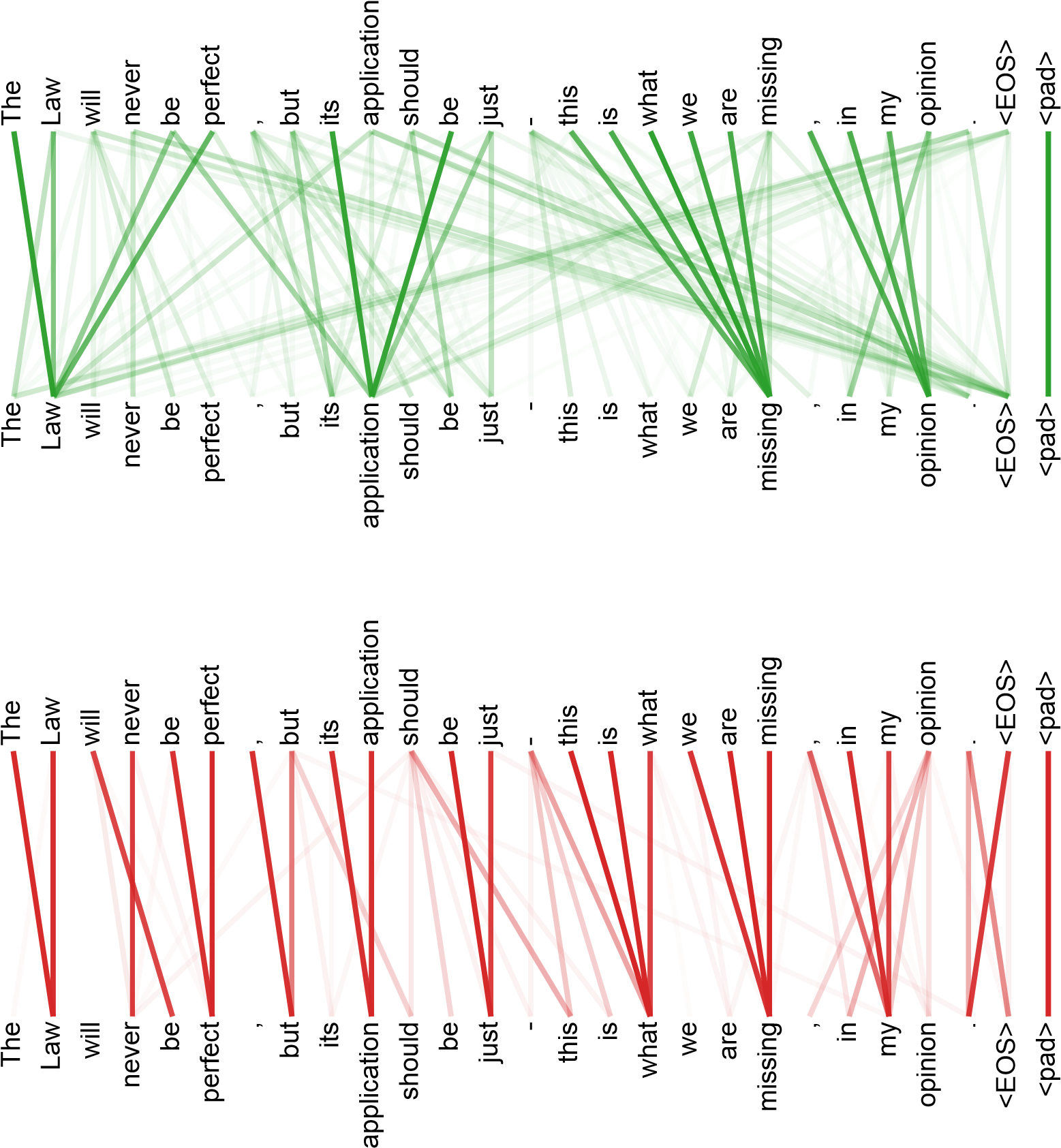


图5：许多注意力头表现出似乎与句子结构有关的行为。我们在上面举了两个这样的例子，分别来自编码器自我注意的第 5 层的第 6 层的两个不同头。负责人显然学会了执行繁重的任务。

1. 为了说明为什么点积变大，假设*q*和*k*是具有平均值的独立随机变量0和方差1.然后他们的点积，，具有平均值0和方差*dk*. [↑](#footnote-ref-1)
2. 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。 [↑](#footnote-ref-2)