如果提供了适当的归属，谷歌特此授权复制本文中的表格和数字，仅供新闻或学术作品使用。

**你只需要关注**

**Ashish Vaswani**∗**谷歌大脑**[avaswani@google.com](mailto:avaswani@google.com)

**诺姆·沙泽**∗**谷歌大脑**[noam@google.com](mailto:noam@google.com)

尼基·帕尔马∗谷歌研究[nikip@google.com](mailto:nikip@google.com)

**雅各布·乌什科雷特**∗**谷歌研究**[usz@google.com](mailto:usz@google.com)

Llion Jones∗谷歌研究[llion@google.com](mailto:llion@google.com)

Aidan N.Gomez∗ †多伦多大学[aidan@cs.toronto.edu](mailto:aidan@cs.toronto.edu)

## 卢卡什·凯泽∗

谷歌大脑

[lukaszkaiser@google.com](mailto:lukaszkaiser@google.com)

## 伊利亚·波洛苏欣∗ ‡

[illia.polosukhin@gmail.com](mailto:illia.polosukhin@gmail.com)

# 摘要

主要的序列转导模型基于复杂的递归或卷积神经网络，该网络包括编码器和解码器。性能最好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结构，即变压器，它完全基于注意机制，完全消除了重复和卷积。在两个机器翻译任务上的实验表明，这些模型在质量上是优越的，同时具有更好的并行性，并且训练时间明显减少。我们的模型在WMT 2014英德翻译任务中获得了28.4个BLEU，比现有的最佳结果（包括合奏）提高了超过2个BLEU。在WMT 2014英法翻译任务中，我们的模型在8个GPU上训练3.5天后，新的单模型State-of-Art-Bleu得分为41.8，这是文献中最好模型训练成本的一小部分。通过成功地将变压器应用于大型和有限训练数据的英语选区分析，我们表明变压器很好地推广到其他任务。

∗同等贡献。上市顺序是随机的。雅各布提议用自我关注取代RNN，并开始努力评估这个想法。Ashish与Illia一起设计并实施了第一批变压器模型，并参与了这项工作的各个方面。Noam提出了一个多头的点产品关注

关注度和无参数的位置表示，成为对方几乎参与到每一个细节中。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高效的推理和可视化。Lukasz和Aidan花了无数漫长的时间设计和实现tensor2tensor的各个部分，取代了我们以前的代码库，极大地改进了结果，并大大加快了我们的研究速度。

†在谷歌大脑工作时完成的工作。

‡在谷歌研究公司工作。

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。

# 导言

循环神经网络，长短期记忆[[13](#_bookmark23)[和门控复发][7](#_bookmark17)[特别是神经网络，已被牢固确立为序列建模和转换问题（如语言建模和机器翻译）的最先进方法[[35](#_bookmark45), [2](#_bookmark12), [5](#_bookmark15)]。此后，许多努力不断地推动了递归语言模型和编码器——解码器体系结构的边界[[38,](#_bookmark48) [24,](#_bookmark34) [15].](#_bookmark25)

递归模型通常沿输入和输出序列的符号位置进行因子计算。将位置与计算时间中的步骤对齐，它们生成隐藏状态h*t*的序列，该序列是先前隐藏状态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实例内的并行化，这在较长的序列长度下变得至关重要，因为内存约束限制了跨实例的批处理。最近的工作通过因式分解技巧在计算效率方面取得了显著的提高[[21](#_bookmark31)]和条件计算[[32](#_bookmark42)]，同时也提高了后者的模型性能。然而，顺序计算的基本限制仍然存在。

注意机制已经成为各种任务中引人注目的序列建模和转导模型的一个组成部分，允许对依赖关系进行建模，而不管它们在输入或输出序列中的距离如何[[2](#_bookmark12), [19](#_bookmark29)]。除少数情况外，所有情况下[[27](#_bookmark37)]然而，这种注意机制是与经常性网络结合使用的。

在这项工作中，我们提出了变压器，一个模型架构，避免重复，而是完全依赖于一个注意机制，以绘制输入和输出之间的全局依赖关系。变压器允许显著增加的并行化，并且在8个P100 GPU上训练了12小时后，可以达到新的翻译质量。

# 背景

减少顺序计算的目标也构成了扩展神经GPU的基础[[16](#_bookmark26)]、字节网[[18](#_bookmark28)]和ConvS2S[[9](#_bookmark19)]，都是以卷积神经网络为基本构建块，对所有输入输出位置并行计算隐藏表示。在这些模型中，将来自两个任意输入或输出位置的信号关联起来所需的操作数随着位置之间的距离而增加，对于ConvS2S，线性增加，对于字节网，则对数增加。这使得学习远距离位置之间的依赖关系变得更加困难[[12](#_bookmark22)]。在变压器中，这被减少到恒定的操作次数，尽管由于平均注意加权位置而降低了有效分辨率，我们用第1节中描述的多头注意抵消了这种影响[3.2.](#_bookmark1)

自我注意，有时称为内注意，是一种注意机制，它将单个序列的不同位置联系起来，以计算序列的表示。自我注意已成功地应用于阅读理解、抽象概括、文本推理和学习任务无关的句子表征等多种任务中[[4,](#_bookmark14) [27,](#_bookmark37) [28,](#_bookmark38) [22].](#_bookmark32)

端到端记忆网络基于重复注意机制而不是序列对齐重复，已被证明在简单语言问答和语言建模任务中表现良好[[34].](#_bookmark44)

然而，据我们所知，变压器是第一个完全依赖于自我关注的转换模型，无需使用序列对准的RNN或卷积来计算其输入和输出的表示。在下面的部分中，我们将描述变压器，激发自我关注，并讨论它相对于诸如[[17,](#_bookmark27) [18]](#_bookmark28) 和[[9].](#_bookmark19)

# 模型体系结构

大多数竞争性的神经序列转导模型都具有编码器——解码器结构[[5](#_bookmark15), [2](#_bookmark12), [35](#_bookmark45)]。这里，编码器将符号表示的输入序列(x1、…、x*n*映射到连续表示的序列z=(z1、…、z*n*。给定z，解码器然后一次生成一个元素的符号的输出序列(y1、…、y*m*)。在每一步中，模型都是自回归的[[10],](#_bookmark20) 在生成下一个符号时，使用先前生成的符号作为附加输入。

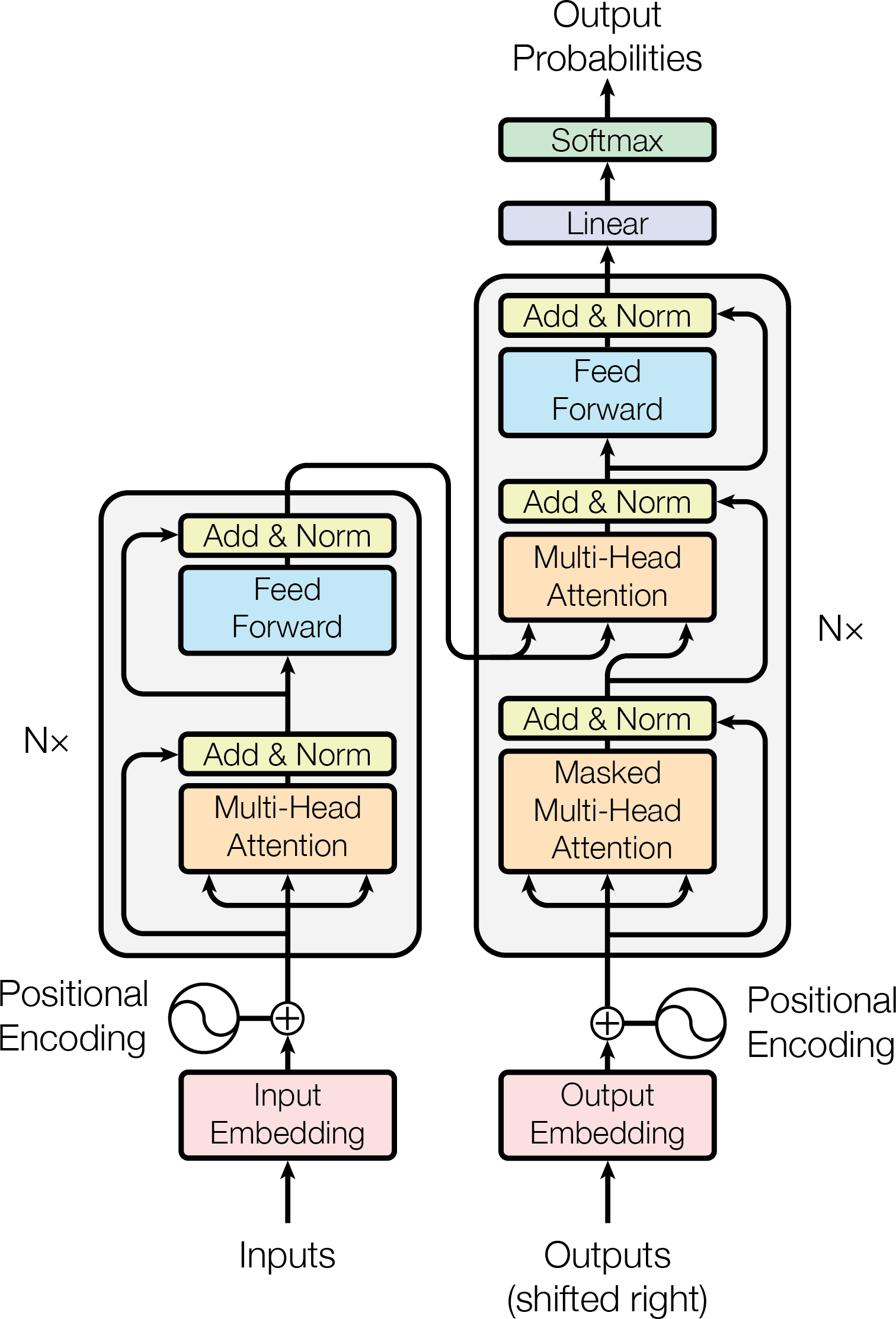


图1：变压器模型架构。

变压器遵循这种整体架构，使用堆叠的自聚焦层和点向的、完全连接的编码器和解码器层，如图的左右两半所示[1,](#_bookmark0) 分别。

## 编码器和解码器堆栈

**编码器：**编码器由N=6个相同层的堆栈组成。每层有两个子层。第一种是多头的自我关注机制，第二种是简单的、位置上完全连接的前馈网络。我们采用了一个剩余连接[[11](#_bookmark21)[围绕两个子层，然后是层归一化[[1](#_bookmark11)]。即每个子层的输出为LayerNorm（x+子层（x）），其中子层（x）是子层本身实现的功能。为了便于这些剩余连接，模型中的所有子层以及嵌入层都会产生尺寸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进行多头关注。与编码器类似，我们在每个子层周围使用剩余连接，然后进行层规范化。我们还修改了解码器栈中的自关注子层，以防止位置关注后续位置。这种屏蔽，再加上输出嵌入偏移一个位置，确保位置i的预测只能依赖于小于i的位置处的已知输出。

## 注意

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。输出计算为加权和

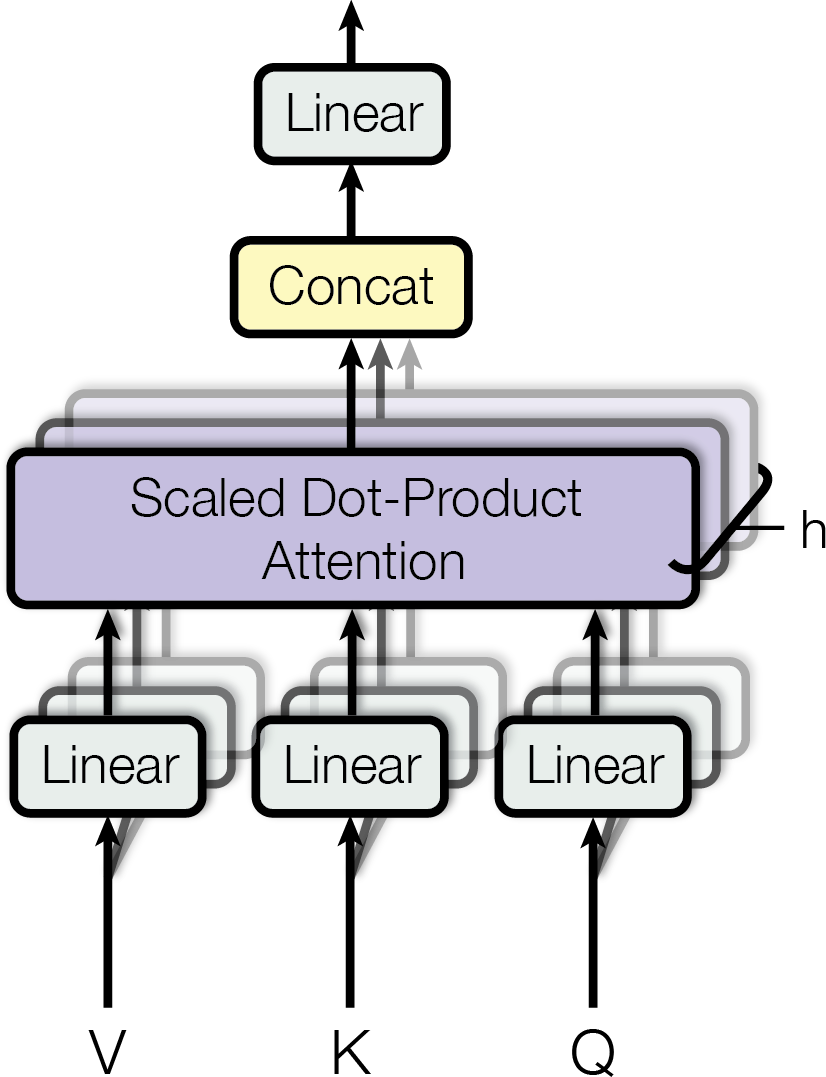
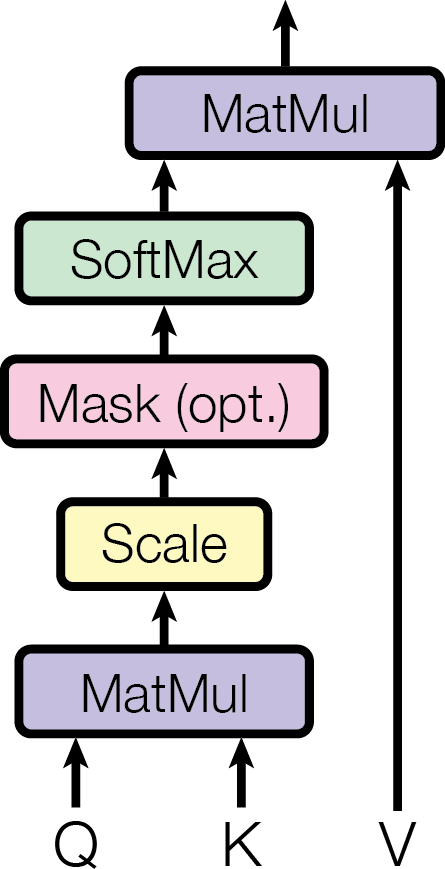
标度点——产品关注度 多头注意力

图2：（左）缩放的点——产品关注度。（右）多头注意力由几个并行运行的注意力层组成。

其中分配给每个值的权重由具有相应键的查询的兼容性函数计算。

## 标度点——产品关注度

我们把我们的特别关注称为“缩放点——产品关注”（图[2).](#_bookmark2) 输入包括维度d*k*的查询和键、维度d*v*的A√和d值。我们计算了

使用所有键进行查询，将每个键除以值。

d*k*，并应用softmax函数来获得

在实践中，我们同时计算一组查询的注意函数，将它们打包成一个矩阵Q。键和值也被打包在矩阵K和V中。我们计算输出矩阵为：

*QKT*

注意(Q,K,V)=软最大值(√*d*

)*V* (1)

最常用的两种注意力功能是加性注意力[[2](#_bookmark12)]，和点积（多重）注意力。点积注意力与我们的算法相同，只是√的缩放因子1。附加注意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要快得多，空间效率更高，因为它可以使用高度优化的矩阵乘法代码来实现。

虽然对于小值的d*k*，两种机制的表现相似，但对于较大值的d*k*，加性注意优于点积注意，而不缩放[[3](#_bookmark13)]。我们猜想，对于d*k*的大值，点积的幅度会增大，从而将softmax函数推入梯度极小的区域[4](#_bookmark4).为了抵消这种影响，我们将点积按√1缩放。

## 多头注意力

我们发现，与使用d模型维度的键、值和查询执行单一的注意力功能相比，使用不同的学习线性投影将查询、键和值线性投影h次到d*k*、d*k*和d*v*维度是有益的。然后，在这些查询、键和值的每个投影版本上，我们并行执行注意函数，生成d*v*-维

4为了说明为什么点积变大，假设q和k的分量是均值为0、方差为1的独立随机变量。然后，它们的点积q·k=.dkq*i*k*i*具有均值0和方差d*k*。

输出值。这些值被连接并再次投影，产生最终值，如图所示[2.](#_bookmark2)

多头关注使模型能够在不同位置共同关注来自不同表示子空间的信息。对于单个注意力头，平均可以抑制这种情况。

多头(Q，K，V)=凹形(头1，...，头h)W*O*

其中头部i=注意力(QW*Q*、KW*K*、V W*V*)

*i i* *i*

其中投影为参数矩阵W*Q*∈R*d*模型×dk，W*K*∈R*d*模型×dk，W*V*∈R*d*模型×dv

和W O∈Rhdv × dmodel。

*i i* *i*

在这项工作中，我们使用了h=8个平行的注意层，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。由于每个头部的维数降低，总的计算成本与全维度的单头部注意力相似。

## 注意在我们模型中的应用

变压器以三种不同的方式使用多头注意力：

* + - * 在“编码器——解码器注意”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都参与输入序列中的所有位置。这模仿了序列对序列模型中典型的编码器——解码器注意机制，例如[[38,](#_bookmark48) [2,](#_bookmark12) [9].](#_bookmark19)
      * 编码器包含自我关注层。在一个自我关注层中，所有的键、值和查询都来自同一个地方，在本例中，是编码器中前一层的输出。编码器中的每个位置都可以照顾编码器上一层中的所有位置。
      * 类似地，解码器中的自我关注层允许解码器中的每个位置关注解码器中的所有位置，直到该位置并包括该位置。我们需要防止解码器中的信息向左流动，以保持自回归特性。我们通过屏蔽（设置为）softmax输入中与非法连接相对应的所有值来实现这种缩放点产品关注。见图[2.](#_bookmark2)

## 位置前馈网络

除了注意子层，我们的编码器和解码器中的每一层都包含一个完全连接的前馈网络，该网络分别且相同地应用于每个位置。这包括两个线性变换，中间有一个ReLU激活。

FFN(x)=max(0,xW1+b1)W2+b2 (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。另一种描述方法是两个卷积，核大小为1。输入和输出的尺寸为d模型=512，内层的尺寸为d*ff*=2048。

## 嵌入和Softmax

与其他序列转导模型类似，我们使用学习的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](#_bookmark40)]。在嵌入层中，我们将这些权重乘以

D模型。

表1：不同层类型的最大路径长度、每层复杂度和最小顺序操作数。n为序列长度，d为表示维数，k为卷积的核大小，r为受限自关注下邻域的大小。

层类型 每层复杂性 顺序的 最大路径长度

业务

自我关注 *O*(*n*2 · *d*) *O*(1) *O*(1)

复发性 *O*(*n* · *d*2) *O*(*n*) *O*(*n*)

卷积 *O*(*k* · *n* · *d*2) *O*(1) *O（对数k（n））*

自我关注（受限） *O*(*r* · *n* · *d*) *O*(1) *O*(*n/r*)

## 位置编码

由于我们的模型不包含递归和卷积，为了使模型利用序列的顺序，我们必须注入一些关于序列中标记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底部的输入嵌入中添加“位置编码”。位置编码具有与嵌入相同的维度d模型，因此可以将两者求和。有许多位置编码的选择，学习和固定[[9].](#_bookmark19)

在这项工作中，我们使用了不同频率的正弦和余弦函数：

*PE*（pos，2i）*=sin(pos/10000*2i/D模型*PE*（pos，2i+1）*=cos(pos/10000*2i/D模型

其中pos是位置，i是尺寸。即，位置编码的每个维度对应于正弦。波长从2 π到10000 2 π形成几何级数。我们选择这个函数是因为我们假设它可以让模型很容易地学会按相对位置来处理，因为对于任何固定的偏移k，PE*pos+k*可以表示为PE*pos*的线性函数。

我们还尝试使用学习的位置嵌入[[9](#_bookmark19)]相反，发现两个版本产生的结果几乎相同（见表[3](#_bookmark9) 第(E)行）。我们选择正弦版本，因为它可能允许模型外推到比训练期间遇到的更长的序列长度。

# 为什么要自我关注

在本节中，我们将自我注意层的各个方面与通常用于将一个可变长度的符号表示序列(x1、…、x*n*映射到另一个等长度的序列(z1、…、z*n*的递归和卷积层进行比较，其中x*i*、z*i*R*d*，例如典型序列转换编码器或解码器中的隐藏层。为了激励我们使用自我关注，我们考虑了三个愿望。

一是每层的总计算复杂度。另一个是可以并行化的计算量，这是通过所需的最少顺序操作数来衡量的。

第三个是网络中远程依赖项之间的路径长度。学习长程依赖性是许多序列转导任务中的一个关键挑战。影响学习这种依赖关系能力的一个关键因素是网络中前向和后向信号所要经过的路径长度。输入和输出序列中任意位置组合之间的这些路径越短，就越容易学习远程依赖关系[[12](#_bookmark22)]。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。

如表所示[1,](#_bookmark5) 自我关注层用连续执行的操作数恒定的数量连接所有位置，而循环层需要O（n）个顺序操作。在计算复杂度方面，当序列被序列化时，自关注层比递归层快

长度n小于表示维数d，这是机器翻译中最先进的模型使用的句子表示的最常见情况，例如单词片[[38](#_bookmark48)]和字节对[[31](#_bookmark41)]陈述。为了提高涉及非常长序列的任务的计算性能，可以将自我注意限制为只考虑输入序列中以各自输出位置为中心的大小r的邻域。这将使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的单个卷积层并不连接所有的输入和输出位置对。这样做需要在连续核的情况下堆叠O(n/k)卷积层，或者在扩展卷积的情况下堆叠O(log*k*(n))[[18](#_bookmark28)]，增加网络中任意两个位置之间最长路径的长度。卷积层通常比循环层贵，贵了k倍。可分离卷积[[6](#_bookmark16)]，但是，将复杂性大大降低到O(KnD+ND2)。然而，即使有k=n，可分离卷积的复杂性也等于我们在模型中采用的方法——自我关注层和点向前馈层的组合。

作为一个附带利益，自我关注可以产生更多可解释的模型。我们从模型中检查注意力分布，并在附录中介绍和讨论示例。不仅单个注意头明显地学会了执行不同的任务，许多注意头似乎表现出与句子的句法和语义结构相关的行为。

# 培训

本节介绍了我们模型的训练方案。

## 培训数据和批处理

我们使用标准的WMT 2014英德数据集进行培训，该数据集包含约450万个句子对。使用字节对编码对句子进行编码[[3](#_bookmark13)]，其拥有约37000个令牌的共享源目标词汇。对于英语法语，我们使用了规模更大的WMT 2014英语法语数据集，由3600万个句子组成，并将标记拆分为32000个单词的词汇[[38](#_bookmark48)]。句子对按大致的序列长度分批在一起。每个训练批次包含一组句子对，其中包含大约25000个源标记和25000个目标标记。

## 硬件和时间表

我们在一台配备8个NVIDIA P100 GPU的机器上训练模型。对于使用本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。我们训练了基本模型，总共100,000步，即12小时。对于我们的大模型，（在表的底部描述[3),](#_bookmark9) 步进时间为1.0秒。大模型接受了30万步（3.5天）的训练。

## 优化器

我们使用了Adam优化器[[20](#_bookmark30)]β1=0.9，β2=0.98和ε=10−9。我们根据以下公式在整个训练过程中改变学习率：

*lrate=d*−0.5*·min(step\_num*−0.5*,step\_num·warmup\_steps*−1.5*)* (3)

这对应于第一个warmup\_steps训练步骤的学习速率线性增加，之后与步骤数的逆平方根成比例地降低。我们使用warmup\_steps=4000。

## 正则化

我们在培训中采用三种类型的正规化：

表2：在2014年的英语到德语和英语到法语的测试中，变压器的BLEU成绩比以前的最先进的模型更好，只需一小部分的培训成本。

蓝色 培训成本（FLOPs）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 模型 | 恩-德 | EN-FR |  | 恩-德 EN-FR |
| 字节网[[18]](#_bookmark28)  深-Att+俯冲[[39]](#_bookmark49) | 23.75 | 39.2 |  | 20  1*.*0 · 10 |
| GNMT + RL [[38]](#_bookmark48) | 24.6 | 39.92 |  | 19 20  2*.*3 · 10 1*.*4 · 10 |
| ConvS2S [[9]](#_bookmark19) | 25.16 | 40.46 |  | 18 20  9*.*6 · 10 1*.*5 · 10 |
| MoE [[32]](#_bookmark42) | 26.03 | 40.56 |  | 19 20  2*.*0 · 10 1*.*2 · 10 |
| 深-Att+后沉合奏[[39]](#_bookmark49) |  | 40.4 |  | 20  8*.*0 · 10 |
| GNMT+RL合奏[[38]](#_bookmark48) | 26.30 | 41.16 |  | 20 21  1*.*8 · 10 1*.*1 · 10 |
| Convs 2 S套装[[9]](#_bookmark19) | 26.36 | **41.29** |  | 19 21  7*.*7 · 10 1*.*2 · 10 |
| 变压器（基本型号） | 27.3 | 38.1 |  | **3*.*3** · **1018** |
| 变压器（大） | **28.4** | **41.8** |  | 19  2*.*3 · 10 |

**剩余辍学我们适用辍学[**[33](#_bookmark43)]到每个子层的输出，然后再将其添加到子层输入并进行规范化。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，我们使用速率P*下降*=0.1。

标签平滑在训练期间，我们采用了值ε*ls*=0.1的标签平滑[[36](#_bookmark46)]。这会伤害困惑，因为模型学会了更不确定，但提高了准确性和BLEU评分。

# 结果

## 机器翻译

WMT 2014英德翻译任务中，表格中的大变压器模型（Transformer（big））[2)](#_bookmark7) 比之前报道的最佳模型（包括合奏）的BLEU性能超过2.0，创造了28.4的新的最先进的BLEU评分。该模型的配置列于表的底部[3.](#_bookmark9) 在8个P100 GPU上进行培训需要3.5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合，其培训成本只是任何竞争模型的一小部分。

在WMT 2014英法翻译任务中，我们的大模型取得了41.0的BLEU评分，超过了之前发布的所有单一模型，仅用不到之前最先进模型培训成本的1/4。针对英语到法语训练的变压器(big)模型使用的是辍学率P*下降*=0.1，而不是0.3。

对于基本模型，我们使用了一个通过平均最后5个检查点获得的单个模型，这些检查点以10分钟的间隔编写。对于大模型，我们平均了最后20个检查点。我们使用了光束搜索，光束大小为4，长度惩罚α=0.6[[38](#_bookmark48)]。这些超参数是在开发集上进行实验后选择的。我们在推理过程中将最大输出长度设置为输入长度+50，但尽可能提前终止[[38].](#_bookmark48)

表[2](#_bookmark7) 总结我们的结果，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比较。我们通过将训练时间、所用GPU的数量和每个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操作的数量[5](#_bookmark8).

## 模型变化

为了评估变压器不同部件的重要性，我们以不同的方式改变了我们的基本模型，测量了变压器的英德翻译性能的变化

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

表3：变压器结构的变化。未列出的值与基本模型中的值相同。所有指标均在newstest2013的英德翻译开发集上。根据我们的字节对编码，列出的困惑是每个字段的，不应该与每个字的困惑进行比较。

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | *N* | *d*model | *d*ff | *h* | *dk* | *dv* | *Pdrop* | *L* | 列车步骤 | PPL  （德夫） | 蓝色  （德夫） | 参数  6  ×10 |
| 基地 | 6 | 512 | 2048 | 8 | 64 | 64 | 0.1 | 0.1 | 100K | 4.92 | 25.8 | 65 |
|  |  |  |  | 1 | 512 | 512 |  |  |  | 5.29 | 24.9 |  |
| (A) |  |  |  | 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 |
|  | 2 |  |  |  |  |  |  |  |  | 6.11 | 23.7 | 36 |
|  | 4 |  |  |  |  |  |  |  |  | 5.19 | 25.3 | 50 |
|  | 8 |  |  |  |  |  |  |  |  | 4.88 | 25.5 | 80 |
| (C) |  | 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 |
|  |  |  |  |  |  |  | 0.0 |  |  | 5.77 | 24.6 |  |
| (D) |  |  |  |  |  |  | 0.2 | 0.0 |  | 4.95 | 25.5 |  |
| 4.67 | 25.3 |
|  |  |  |  |  |  |  |  | 0.2 |  | 5.47 | 25.7 |  |
| (E) | 位置嵌入代替正弦 | | | | | | | | | 4.92 | 25.7 |  |
| 大 | 6 | 1024 | 4096 | 16 |  |  | 0.3 |  | 300K | **4.33** | **26.4** | 213 |

开发集，newstest2013。我们使用了上一节中所述的波束搜索，但没有检查点平均。我们在表中给出了这些结果[3.](#_bookmark9)

表中[3](#_bookmark9) 在(a)行中，我们改变注意头的数量以及注意键和值维度，保持计算量不变，如第1节所述[3.2.2.](#_bookmark3) 虽然单头注意力比最佳设置差0.9倍，但由于头部过多，质量也会下降。

表中[3](#_bookmark9) 在行(B)中，我们观察到减小注意键大小d*k*会损害模型质量。这表明确定兼容性并不容易，比dot产品更复杂的兼容性函数可能是有益的。我们在(C)和(D)行进一步观察到，不出所料，较大的模型更好，而dropout对于避免过度拟合非常有帮助。在(E)行中，我们用学习的位置嵌入代替正弦位置编码[[9](#_bookmark19)]，并观察到与基本模型几乎相同的结果。

## 英语选区解析

为了评估变压器是否能推广到其他任务，我们进行了英语选区解析实验。这项任务提出了具体的挑战：产出受到很强的结构约束，而且比投入要长得多。此外，RNN序列对序列模型在小数据条件下还不能达到最先进的结果[[37].](#_bookmark47)

我们在宾夕法尼亚树库的华尔街日报(WSJ)部分训练了一个d*模型*=1024的4层变压器[[25](#_bookmark35)]，约4万个训练句子。我们还在半监督环境中使用更大的高自信和BerkleyParser语料库对其进行了训练，该语料库包含大约1700万个句子[[37](#_bookmark47)]。我们对仅适用于《华尔街日报》的设置使用了16K个令牌的词汇，对半监督设置使用了32K个令牌的词汇。

我们只进行了少量的实验来选择注意力和剩余的辍学者（第[5.4),](#_bookmark6) 在第22节开发集上，所有其他参数从英德基础翻译模型保持不变。在推理过程中，我们

表4：Transformer很好地推广到了英语选区分析（结果见《华尔街日报》第23节）

|  |  |  |
| --- | --- | --- |
| **解析器** | **培训** | **WSJ 23 F1** |
| Vinyals & Kaiser el al. (2014) [[37]](#_bookmark47) | 仅限《华尔街日报》，具有歧视性 | 88.3 |
| 彼得罗夫等人（2006年）[[29]](#_bookmark39) | 仅限《华尔街日报》，具有歧视性 | 90.4 |
| 朱等人（2013年）[[40]](#_bookmark50) | 仅限《华尔街日报》，具有歧视性 | 90.4 |
| 戴尔等人（2016年）[[8]](#_bookmark18) | 仅限《华尔街日报》，具有歧视性 | 91.7 |
| 变压器（4层） | 仅限《华尔街日报》，具有歧视性 | 91.3 |
| 朱等人（2013年）[[40]](#_bookmark50) | 半监督 | 91.3 |
| 黄&哈珀（2009）[[14]](#_bookmark24) | 半监督 | 91.3 |
| 麦克洛斯基等人（2006年）[[26]](#_bookmark36) | 半监督 | 92.1 |
| Vinyals & Kaiser el al. (2014) [[37]](#_bookmark47) | 半监督 | 92.1 |
| 变压器（4层） | 半监督 | 92.7 |
| Luong等人（2015年）[[23]](#_bookmark33) | 多任务 | 93.0 |
| 戴尔等人（2016年）[[8]](#_bookmark18) | 生成的 | 93.3 |

将最大输出长度增加到输入长度+300。我们仅对WSJ和半监督设置使用了21和α=0.3的光束大小。

我们的结果在表中[4](#_bookmark10) 表明尽管缺乏特定于任务的调整，我们的模型表现出奇的好，产生了比以前报道的所有模型更好的结果，除了递归神经网络文法[[8].](#_bookmark18)

与RNN序列对序列模型相比[[37](#_bookmark47)]，变压器的性能优于Berkeley解析器[[29]](#_bookmark39) 即使只在《华尔街日报》的40K个句子的训练集上训练。

# 结论

在本研究中，我们提出了第一个完全基于注意力的序列转换模型Transformer，它用多头自我注意力来取代编码器——解码器架构中最常用的递归层。

对于翻译任务，变压器的训练速度可以明显快于基于递归层或卷积层的架构。在WMT 2014英德翻译和WMT 2014英法翻译任务中，我们都达到了新的技术水平。在前一个任务中，我们的最佳模型甚至超过了之前报道的所有集合。

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。我们计划将变压器扩展到涉及文本以外的输入和输出模式的问题，并研究本地、受限的注意力机制，以有效地处理图像、音频和视频等大型输入和输出。使代的顺序性降低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/ tensorflow/tensor2tensor](https://github.com/tensorflow/tensor2tensor).

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

# 参考文献

1. 吉米·雷巴、杰米·瑞安·基罗斯和杰弗里·辛顿。层规范化。arXiv预印[*存档：1607.06450*](http://arxiv.org/abs/1607.06450), 2016.
2. Dzmitry Bahdanau、Kyunghyun Cho和Yoshua Bengio。通过联合学习对齐和翻译来实现神经机器翻译。CoRR，abs/1409.0473，2014。
3. 丹尼·布里茨、安娜·戈尔迪、明胜良和郭诉勒。神经机器翻译体系结构的大规模探索。CoRR，abs/1703.03906，2017。
4. 程建鹏、李东、米雷拉·拉帕塔。长时间的短期记忆——机器阅读的网络。arXiv预印[*存档：1601.06733*](http://arxiv.org/abs/1601.06733), 2016.
5. 赵京贤、巴特·范·梅里恩波尔、卡格拉尔·古尔切雷、费蒂·布加雷斯、霍尔格·施文克和约书亚·本吉奥。使用rnn编码器——解码器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。例外：具有深度可分离卷积的深度学习。arXiv预印[*存档：1610.02357*](http://arxiv.org/abs/1610.02357), 2016.
7. Chung Junyoung，Caglar Gülzehre，Kyunghyun Cho和Yoshua Bengio。门控递归神经网络对序列建模的实证评价。CoRR，abs/1412.3555，2014。
8. 克里斯·戴尔、阿迪古娜·昆科罗、米格尔·巴列斯特罗斯和诺亚·a·史密斯。递归神经网络语法。正在进行中。美国国家广播公司，2016年。
9. 乔纳斯·格林、迈克尔·奥利、大卫·格兰杰、丹尼斯·亚拉茨和扬·N·多芬。卷积序列到序列学习。arXiv预印[*存档：1705.03122 v2*](http://arxiv.org/abs/1705.03122), 2017.
10. 亚历克斯·格雷夫斯。用递归神经网络生成序列。arXiv预印[*存档：1308.0850*](http://arxiv.org/abs/1308.0850), 2013.
11. 何开明、张翔宇、任少卿、孙健。图像年龄识别的深度剩余学习。IEEE计算机视觉和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·弗拉斯科尼和于尔根·施密德胡伯。循环网络中的梯度流：学习长期依赖的困难，2001年。
13. 塞普·霍赫赖特和于尔根·施密德胡伯。长短时记忆。Neural computation,9(8)：1735-1780,1997.
14. 黄忠强和玛丽·哈珀。自训练PCFG语法，具有跨语言的潜在注释。载于《2009年自然语言处理经验方法会议录》，第832-841页。ACL，2009年8月。
15. 拉法尔·约泽福维奇、奥里奥尔·维尼亚尔斯、迈克·舒斯特、诺姆·沙泽尔和吴永辉。探索语言建模的局限性。arXiv预印[*存档：1602.02410*](http://arxiv.org/abs/1602.02410), 2016.
16. 卢卡斯·凯泽和萨米·本吉奥。主动记忆能代替注意力吗？《神经信息处理系统进展》，NIPS，2016年。
17. 卢卡斯·凯泽和伊利亚·苏茨凯弗。神经GPU学习算法。2016年国际学习表征会议。
18. 纳尔·卡尔奇布伦纳、拉塞·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·卡武克库奥卢。线性时间中的神经机器翻译。arXiv预印[*存档：1610.10099 v2*](http://arxiv.org/abs/1610.10099), 2017.
19. 尹金、卡尔·丹顿、黄隆和亚历山大·拉什。结构化注意力网络。2017年国际学习表征会议。
20. 迪德里克·金玛和吉米·巴。亚当：随机优化的一种方法。载于ICLR，2015年。
21. Oleksii Kuchaiev和Boris Ginsburg。LSTM网络的分解技巧。arXiv预印[*存档：1703.10722*](http://arxiv.org/abs/1703.10722), 2017.
22. 林周翰、冯敏伟、西塞罗·诺盖拉·多斯桑托斯、莫宇、炳翔、周博文、约书亚·本吉奥。结构化的自我注意句子嵌入。arXiv预印[*存档：1703.03130*](http://arxiv.org/abs/1703.03130), 2017.
23. 明胜良，郭诉乐，伊利亚·苏茨凯弗，奥里奥尔·维尼亚尔斯和卢卡斯·凯泽。多任务序列到序列学习。arXiv预印[*存档：1511.06114*](http://arxiv.org/abs/1511.06114), 2015.
24. 梁明胜、范晓和克里斯托弗·D·曼宁。基于注意力的神经机器翻译的有效方法。arXiv预印[*存档：1508.04025*](http://arxiv.org/abs/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·圣托里尼。建立一个大型的注释英语语料库：宾夕法尼亚树银行。计算语言学，19（2）：313-330，1993。
26. 大卫·麦克洛斯基、尤金·查尼亚克和马克·约翰逊。有效的自我训练，用于解析。载于NAACL人类语言技术会议记录，主要会议，第152-159页。ACL，2006年6月。
27. Ankur Parikh，Oscar Täckström，Dipanjan Das和Jakob Uszkoreit。一个可分解的注意力模型。《自然语言处理中的经验方法》，2016年。
28. 罗曼·保卢斯、熊才明和理查德·索彻。一个用于抽象总结的深度强化模型。arXiv预印[*存档：1705.04304*](http://arxiv.org/abs/1705.04304), 2017.
29. 斯拉夫·彼得罗夫、利昂·巴雷特、罗曼·蒂博和丹·克莱恩。学习准确、紧凑、可解释的树注释。载于第21届国际计算语言学会议和ACL第44届年会论文集，第433-440页。ACL，2006年7月。
30. 奥菲尔出版社和利奥·沃尔夫。使用输出嵌入来改进语言模型。arXiv预印[*档案：1608.05859*](http://arxiv.org/abs/1608.05859), 2016.
31. 里科·森里奇、巴里·哈多和亚历山德拉·伯奇。用子词单元进行生僻词的神经机器翻译。arXiv预印[*存档：1508.07909*](http://arxiv.org/abs/1508.07909), 2015.
32. 诺姆·沙泽尔、阿扎莉亚·米罗塞尼、克日什托夫·马齐亚兹、安迪·戴维斯、郭乐、杰弗里·辛顿和杰夫·迪恩。大得离谱的神经网络：稀疏的专家层。arXiv预印[*存档：1701.06538*](http://arxiv.org/abs/1701.06538), 2017.
33. Nitish Srivastava、Geoffrey E Hinton、Alex Krizhevsky、Ilya Sutskever和Ruslan Salakhutdi——11月辍学：防止神经网络过度拟合的简单方法。机器学习研究杂志，15（1）：1929-1958，2014。
34. 塞恩巴亚尔·苏赫巴塔尔、阿瑟·斯拉姆、杰森·韦斯顿和罗布·弗格斯。端到端内存网络。载于C.Cortes、N.D.Lawrence、D.D.Lee、M.Sugiyama和R.Garnett，编辑，神经信息处理系统的进展28，第2440-2448页。柯伦联合公司，2015年。
35. 伊利亚·苏茨凯弗，奥里奥尔·维尼亚尔斯，和Quoc等。用神经网络进行序列到序列学习。《神经信息处理系统的进展》，第3104-3112页，2014年。
36. 克里斯蒂安·塞格迪、文森特·范胡克、谢尔盖·约菲、乔纳森·什伦斯和兹比格涅夫·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和Hinton。语法作为一门外语。《神经信息处理系统进展》，2015年。
38. 吴永辉、迈克·舒斯特、陈志锋、郭诉乐、穆罕默德·诺鲁齐、沃尔夫冈·马切里、马克西姆·克里昆、袁曹、秦高、克劳斯·马切里等谷歌的神经机器翻译系统：架起人与机器翻译的桥梁。arXiv预印[*存档：1609.08144*](http://arxiv.org/abs/1609.08144), 2016.
39. 周杰、曹颖、王旭光、李鹏、徐伟。神经机器翻译的快速前向连接深度递归模型。更正，abs/1606.04199，2016。
40. 朱慕华、张悦、陈文亮、张敏、朱敬波。快速准确的移位减少成分解析。载于ACL第51届年会会议记录（第1卷：长篇论文），第434-443页。ACL，2013年8月。

# 注意查看

图3：6层5中编码器自我关注中的远程依赖引起的注意机制的一个例子。许多注意力集中在动词“making”的一个遥远的依赖上，完成了短语“making...more discurit”。这里只显示了“制造”这个词的注意事项。不同的颜色代表不同的头部。最好用颜色来看。

永远不会

完美

,

但是

应该

公正

-

我们就是这样

失踪

,

在我看来

<pad>

法律永远不会完美

,

但是

申请应

公正

-

我们就是这样

失踪

,

在我的

意见

.

<EOS>

<pad>

法律永远不会

完美

,

但它的应用

应该

公正

-

我们就是这样

失踪

,

在我看来

.

<EOS>

<pad>

法律永远不会完美

,

但它

申请应

公正

-

我们就是这样

失踪

,

在我的

意见

.

<EOS>

<pad>

法律永远不会

完美

,

但它的应用

应该

公正

-

我们就是这样

失踪

,

在我看来

.

<EOS>

<pad>

法律永远不会完美

,

但它

申请应

公正

-

我们就是这样

失踪

,

在我的

意见

.

<EOS>

<pad>

法律永远不会

完美

,

但它的应用

应该

公正

-

我们就是这样

失踪

,

在我看来

.

<EOS>

<pad>

法律永远不会完美

,

但它

申请应

公正

-

我们就是这样

失踪

,

在我的

意见

.

<EOS>

<pad>