 注意力是一切你需要的 Ashish Vaswani Noam Shazeer Niki Parmar Jakob Uszkoreit GoogleBrain GoogleBrain GoogleResearch GoogleResearch avaswani@google.com noam@google.com nikip@google.com usz@google.com Llion Jones Aidan N. Gomez † ukasz Kaiser GoogleResearch University of Toronto GoogleBrain llion@google.com aidan@cs.toronto.edu lukaszkaiser@google.com Illia Polosukhin ‡ illia.polosukhin@gmail.com 摘要
目前主流的序列转换模型基于复杂的递归或卷积神经网络，其中包括编码器和解码器。性能最好的模型还通过注意力机制连接编码器和解码器。我们提出了一种新的简单网络架构，Transformer，仅基于注意力机制，完全放弃了循环和卷积。在两个机器翻译任务上的实验证明，这些模型在质量上优于其他模型，而且并行化效果更好，训练时间明显缩短。我们的模型在WMT 2014年英德翻译任务上获得28.4
BLEU的成绩，相比已有的最佳结果（包括集成模型），提高了超过2个BLEU。在WMT 2014年英法翻译任务上，我们的模型在单一模型状态下将BLEU得分提高到了41.8，经过了仅在八个GPU上进行了3.5天的训练，成本远低于现有文献中最佳模型的训练成本。我们还证明了Transformer在其他任务上的泛化能力，通过成功将其应用于英文成分解析，无论是使用大量还是有限的训练数据都能取得良好效果。

1引言 循环神经网络，长短期记忆以及门控循环神经网络，在序列建模和转导问题中，如语言建模和机器翻译方面已经被确定为艺术水平的方法[13,7,35,2,5]。许多工作继续推动循环语言模型和编码器-解码器架构的边界[38,24,15]。 循环模型通常是根据输入和输出序列的符号位置进行计算。将位置与计算时间步骤对齐，它们生成一个隐藏状态序列h，作为先前隐藏状态h和位置t的输入的函数。然而，这种顺序性质在训练示例内部无法并行化，这在较长的序列长度下变得关键，因为内存限制限制了跨示例的批处理。最近的研究通过因式分解技巧和条件计算[21,32]在计算效率方面取得了显著的改进，同时在后一种情况下提高了模型性能。然而，顺序计算的基本约束仍然存在。 注意机制已成为引人注目的序列建模和转导模型的一个重要组成部分，它允许对依赖关系进行建模，而不考虑其在输入或输出序列中的距离。然而，除了少数情况[27]之外，这些注意机制通常与循环网络结合使用。 在这项工作中，我们提出了变压器（Transformer），这是一种模型体系结构，它摒弃了循环性，而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。变压器可以实现更高的并行化，并在仅经过12小时在8个P100GPU上训练后，达到了翻译质量的最新水平。 2背景
减少顺序计算的目标还构成了ExtendedNeuralGPU [16]，ByteNet[18]和ConvS2S[9]的基础，它们都使用卷积神经网络作为基本构建块，为所有输入和输出位置同时计算隐藏表示。在这些模型中，从两个任意输入或输出位置之间相关信号所需的操作数量随着位置之间的距离增大而线性增长（对于ConvS2S）或对数增长（对于ByteNet），这使得学习远距离位置之间的依赖关系更加困难。在变压器中，这被减少为恒定数量的操作，尽管这会以平均注意权重的位置来降低效果分辨率，这种效果可以通过Multi-Head Attention来抵消，如第3.2节所述。 自注意力，有时称为自内注意力，是一种关联同一序列中不同位置的注意机制，以计算序列的表示。自注意力已成功地应用于各种任务，包括阅读理解、抽象摘要、文本蕴含和学习任务无关的句子表示[4,27,28,22]。 端到端内存网络是基于逐步关注机制而不是序列对齐重复的注意机制，已经在简单语言问答和语言建模任务上表现良好[34]。 据我们所知，然而，变压器是第一个完全依赖于自注意力来计算输入和输出表示而不使用序列对齐的RNN或卷积的转导模型。在接下来的章节中，我们将描述变压器，阐明自注意力并讨论它相对于模型[17,18]和[9]的优势。 3模型架构 大多数竞争性神经序列转导模型具有编码器-解码器结构[5,2,35]。在这里，编码器将表示符号（x1, ..., xn）的输入序列映射到连续表示序列z = (z1, ..., zn)。给定z，解码器然后逐个元素地生成输出序列(y1, ..., ym)的符号。在每个步骤中，模型是自回归[10]的，当生成下一个符号时，它消耗先前生成的符号作为额外的输入。