

最新Transformer模型大盘点，NLP学习必备，Google AI研究员出品



七月在线 ...

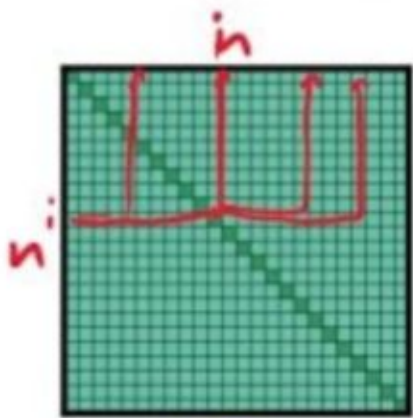
关注公众号：七月在线实验室 领取AI干货大礼包

27 人赞同了该文章

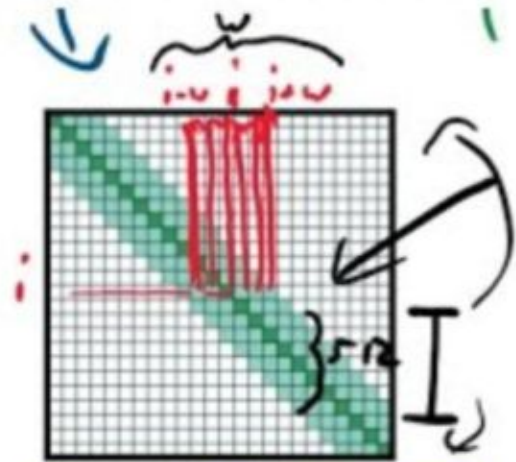
量子位 报道 | 公众号 QbitAI

可高效处理长文本的模型Longformer、和堪称“升级版”Transformer的BigBird模型，到底有什么区别？

Longformer



(a) Full n^2 attention



(b) Sliding window attention

Transformer的其他各种变体（X-former）到底都长什么样、又有哪些新应用？

由于Transformer模型的发展速度日新月异，一天一个样，哪怕是隔段时间回来研究，模型可能也已经多了不少。

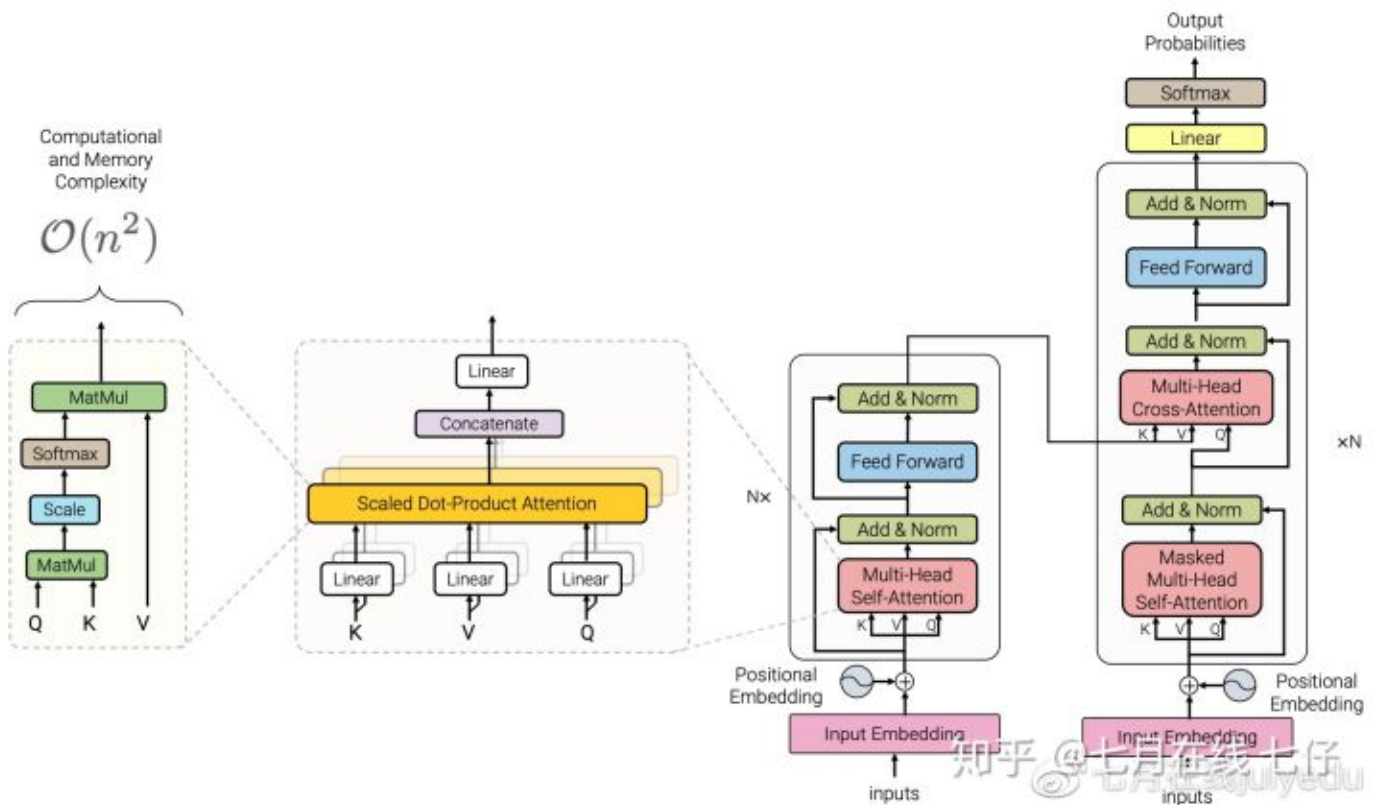
在机器翻译任务上，Transformer表现超过了RNN和CNN，只需要编/解码器就能达到很好的效果，可以高效地并行化。

好消息是，这里有一篇Transformer模型的“最新动向”，它集中探讨Transformer新模型对于自注意力机制（Self-attention）的改进，并对这些模型进行对比。

此外，还有模型在NLP、计算机视觉和强化学习等各个领域的最新应用。

标准Transformer模型

首先来看看，标准的Transformer模型是什么样的。



知乎

(图中玫红色的部分)，解码器通常多一个（交叉）注意力机制。

Transformer最重要的部分，就是注意力机制。

通俗来讲，注意力机制在图像处理中的应用，是让机器“像人一样特别注意图像的某个部分”，就像我们在看图时，通常会“特别关注”图中的某些地方。



这其中，自注意力机制是定义Transformer模型特征的关键，其中一个重点难题就在于它的时间复杂度和空间复杂度上。

由于注意力机制直接将序列（sequence）两两比较，导致计算量巨大（计算量变成 $O(n^2)$ ）。

最近，大量论文提出了新的Transformer“变种”，它们的根本目的都是加速模型的效率，但如果一篇篇去看，可能有点眼花缭乱。

为此，Google AI的研究人员特意整理了一篇Transformer模型的发展论文，仔细讲解它们的出处。

▲ 赞同 27 ▼ ● 1 条评论 ➤ 分享 ♥ 喜欢 ★ 收藏 ...

2种分类方法

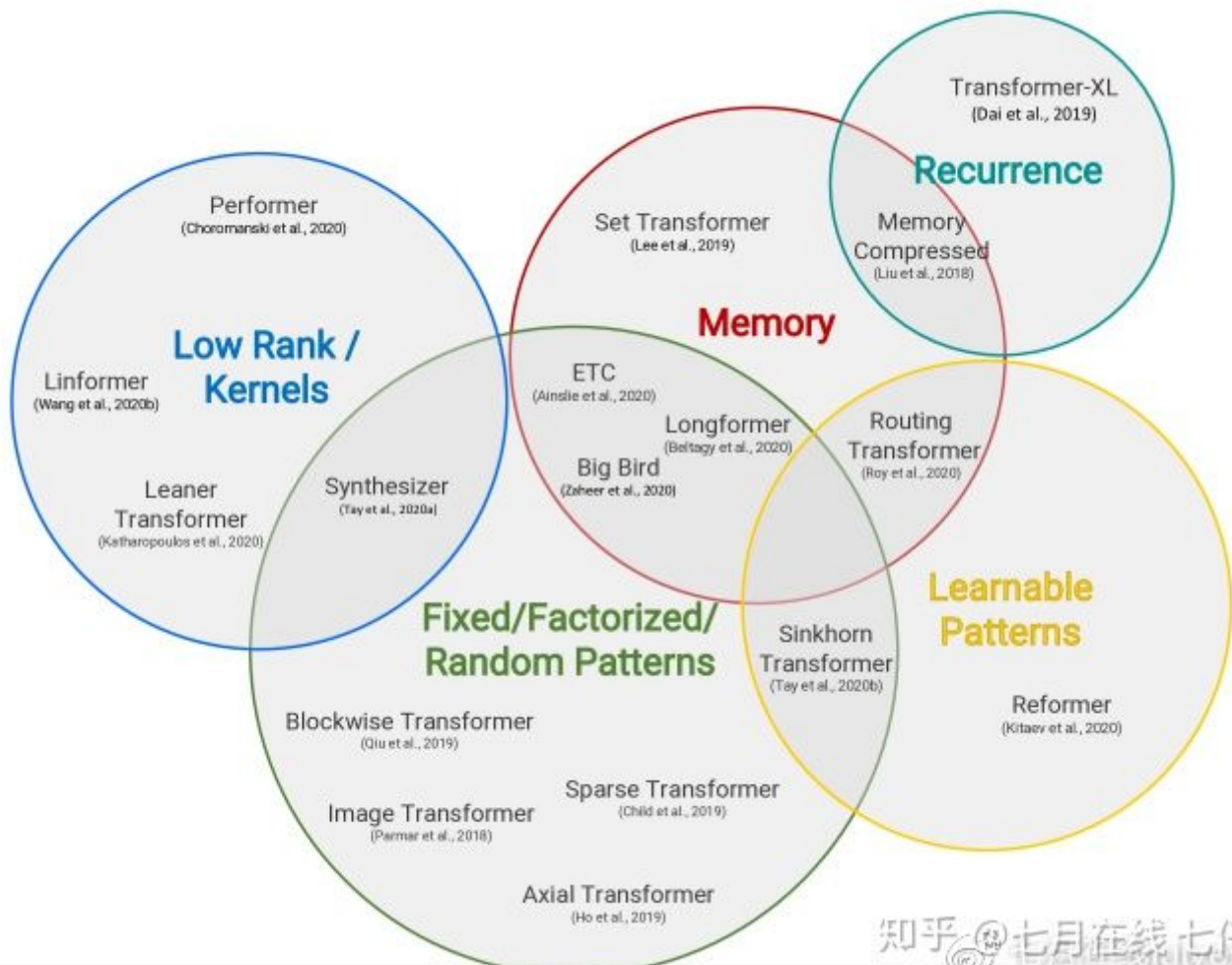
按使用方法来分类的话，Transformer模型可以分成如下3类：

只用编码器：可用于分类

只用解码器：可用于语言建模

编码器-解码器：可用于机器翻译

但如果按这些变种的**提高效率的原理**，也就是“高效方法”来分类，那么Transformer模型的这些“变种”则可以被分成如下几类：



知乎 七月在线 七仔

Learnable Patterns（可学习模式）：以数据驱动的方式学习访问模式，关键在于确定token相关性。

Memory（内存）：利用可以一次访问多个token的内存模块，例如全局存储器。

Low Rank（低秩）：通过利用自注意力矩阵的低秩近似，来提高效率。

Kernels（内核）：通过内核化的方式提高效率，其中核是注意力矩阵的近似，可视为低秩方法的一种。

Recurrence（递归）：利用递归，连接矩阵分块法中的各个块，最终提高效率。

可以看见，近期Transformer相关的研究都被分在上面的图像中了，非常清晰明了。

了解完分类方法后，接下来就是Transformer模型的各种变体了。

17种经典“X-former”

1、Memory Compressed Transformer（2018）

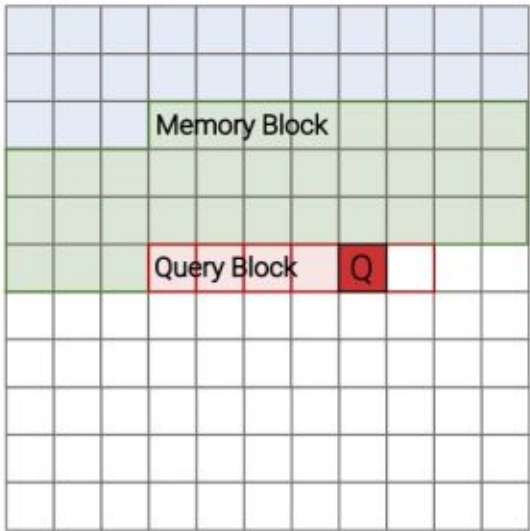
这是让Transformer能更好地处理长序列的早期尝试之一，主要修改了两个部分：定位范围注意、

每个部分的注意力成本不变，激活次数就能根据输入长度线性缩放。

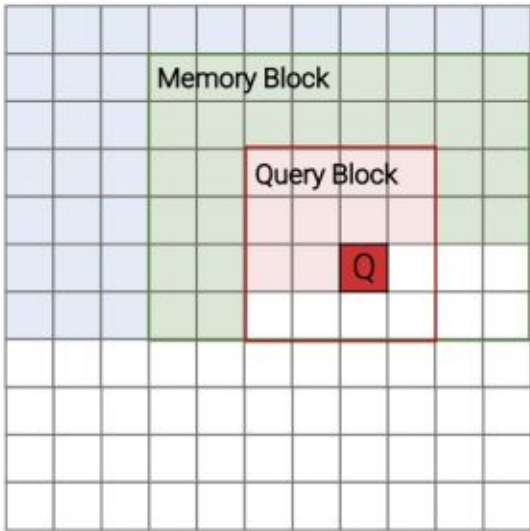
后者则是采用跨步卷积，减少注意力矩阵的大小、以及注意力的计算量，减少的量取决于跨步的步幅。

2、Image Transformer（2018）

这是个受卷积神经网络启发的Transformer变种，重点是局部注意范围，即将接受域限制为局部领域，主要有两种方案：一维局部注意和二维局部注意。



(a) 1-dimensional local attention



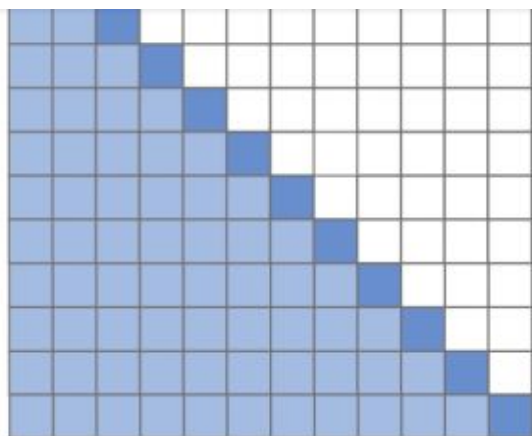
(b) 2-dimensional local attention

不过，这种模型有一个限制条件，即要以失去全局接受域为代价，以降低存储和计算成本。

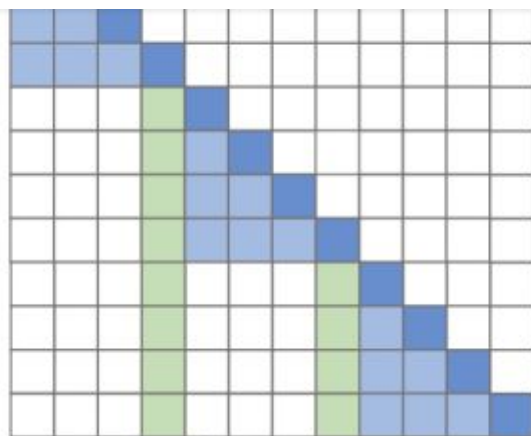
3、Set Transformer（2019）

这个模型是为解决一种特殊应用场景而生的：输入是一组特征，输出是这组特征的函数。

知乎



(a) Transformer



(b) Sparse Transformer

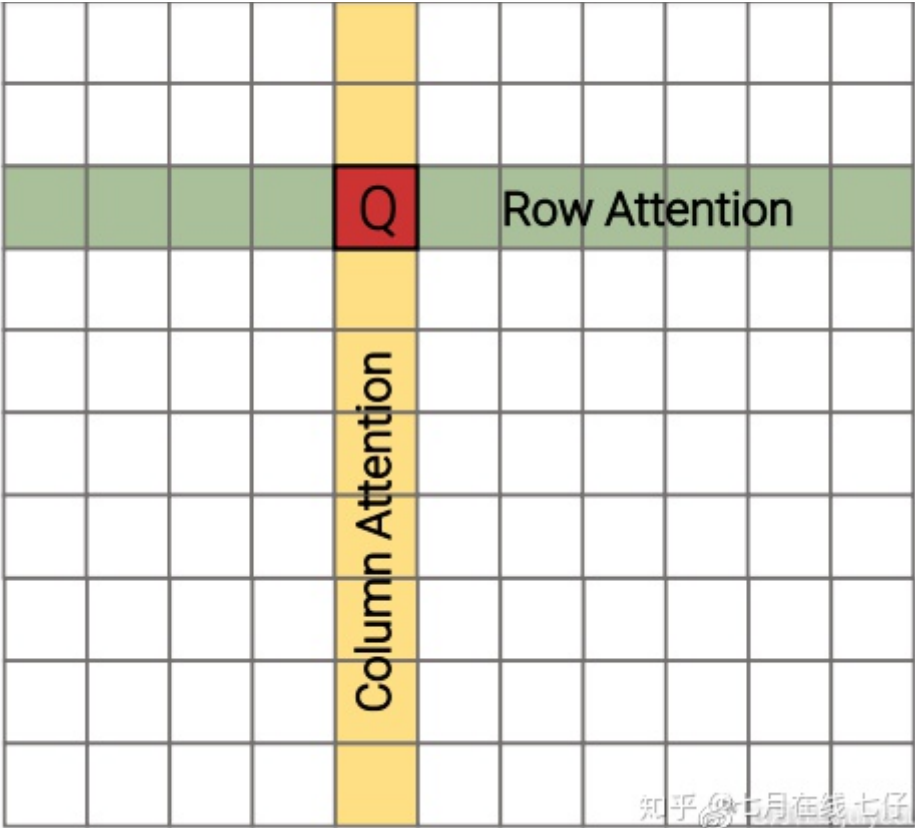
它利用了稀疏高斯过程，将输入集大小的注意复杂度从二次降为线性。

4、Sparse Transformer (2019)

这个模型的关键思想，在于仅在一小部分稀疏的数据对上计算注意力，以将密集注意力矩阵简化为稀疏版本。

不过这个模型对硬件有所要求，需要自定义GPU内核，且无法直接在TPU等其他硬件上使用。

5、Axial Transformer (2019)



这个模型主要沿输入张量的单轴施加多个注意力，每个注意力都沿特定轴混合信息，从而使沿其他轴的信息保持独立。

由于任何单轴的长度通常都比元素总数小得多，因此这个模型可以显著地节省计算和内存。

6、Longformer (2020)

Sparse Transformer的变体，通过在注意力模式中留有空隙、增加感受野来实现更好的远程覆盖。

在分类任务上，Longformer采用可以访问所有输入序列的全局token（例如CLS token）。

7、Extended Transformer Construction (2020)

同样是Sparse Transformer的变体，引入了一种新的全局本地注意力机制，在引入全局token方面与Longformer相似。

8、BigBird (2020)

与Longformer一样，同样使用全局内存，但不同的是，它有独特的“内部变压器构造（ITC）”，即全局内存已扩展为在sequence中包含token，而不是简单的参数化内存。

然而，与ETC一样，BigBird同样不能用于自动回归解码。

9、Routing Transformer (2020)

提出了一种基于聚类的注意力机制，以数据驱动的方式学习注意力稀疏。为了确保集群中的token数量相似，模型会初始化聚类，计算每个token相对于聚类质心的距离。

10、Reformer (2020)

一个基于局部敏感哈希（LSH）的注意力模型，引入了可逆的Transformer层，有助于进一步减少内存占用量。

模型的关键思想，是附近的向量应获得相似的哈希值，而远距离的向量则不应获得相似的哈希值，因此被称为“局部敏感”。

11、Sinkhorn Transformer (2020)

这个模型属于分块模型，以分块的方式对输入键和值进行重新排序，并应用基于块的局部注意力机制来学习稀疏模式。

12、Linformer (2020)

这是基于低秩的自注意力机制的高效Transformer模型，主要在长度维度上进行低秩投影，在单次转换中按维度混合序列信息。

目前，它已经被证明可以在基本保持预测性能的情况下，将推理速度提高多达三个数量级。

14、Performer (2020)

这个模型利用正交随机特征（ORF），采用近似的方法避免存储和计算注意力矩阵。

15、Synthesizer models (2020)

这个模型研究了调节在自注意力机制中的作用，它合成了一个自注意力模块，近似了这个注意权重。

16、Transformer-XL (2020)

这个模型使用递归机制链接相邻的部分。基于块的递归可被视为与其他讨论的技术正交的方法，因为它没有明确稀疏密集的自注意力矩阵。

17、Compressive Transformers (2020)

这个模型是Transformer-XL的扩展，但不同于Transformer-XL，后者在跨段移动时会丢弃过去的激活，而它的关键思想则是保持对过去段激活的细粒度记忆。

整体来说，这些经典模型的数量如下：

知乎

Image Transformer [†] (Parmar et al., 2018)	$\mathcal{O}(n.m)$	✓	FP
Set Transformer [†] (Lee et al., 2019)	$\mathcal{O}(nk)$	✗	M
Transformer-XL [†] (Dai et al., 2019)	$\mathcal{O}(n^2)$	✓	RC
Sparse Transformer (Child et al., 2019)	$\mathcal{O}(n\sqrt{n})$	✓	FP
Reformer [†] (Kitaev et al., 2020)	$\mathcal{O}(n \log n)$	✓	LP
Routing Transformer (Roy et al., 2020)	$\mathcal{O}(n \log n)$	✓	LP
Axial Transformer (Ho et al., 2019)	$\mathcal{O}(n\sqrt{n})$	✓	FP
Compressive Transformer [†] (Rae et al., 2020)	$\mathcal{O}(n^2)$	✓	RC
Sinkhorn Transformer [†] (Tay et al., 2020b)	$\mathcal{O}(b^2)$	✓	LP
Longformer (Beltagy et al., 2020)	$\mathcal{O}(n(k+m))$	✓	FP+M
ETC (Ainslie et al., 2020)	$\mathcal{O}(n_g^2 + nn_g)$	✗	FP+M
Synthesizer (Tay et al., 2020a)	$\mathcal{O}(n^2)$	✓	LR+LP
Performer (Choromanski et al., 2020)	$\mathcal{O}(n)$	✓	KR
Linformer (Wang et al., 2020b)	$\mathcal{O}(n)$	✗	LR
Linear Transformers [†] (Katharopoulos et al., 2020)	$\mathcal{O}(n)$	✓	KR
Big Bird (Zaheer et al., 2020)	$\mathcal{O}(n)$	✗	FP+M

更详细的解读（包括具体的模型参数等），以及对Transformer未来趋势的预测，可以戳下方链接查看整篇论文。

作者介绍



论文一作Yi Tay，硕士和博士均毕业于新加坡国立大学计算机科学。

目前，Yi Tay在Google AI从事研究工作，主要方向是自然语言处理和机器学习。

论文链接：

arxiv-vanity.com/papers...

本文转自：量子位

发布于 09-21

自然语言处理

Transformer

RNN

知乎

Jacob Devlin Ming-Wei Chang Kenton Lee Kristina Toutanova
Google AI Language
jacobdevlin,mingweichang,kentonl,kristout}@google.com


【NLP】Google BERT模型原理详解

李rumor

宅
Tr
不

1 条评论

⇌ 切换为时间排序

写下你的评论... 

 knimet 10-07

哇，yitay开始写综述了啊

 赞