Lite-transformer

Lite-Transformer原理分析：

Transformer模型因其训练效率高、捕获长距离依赖能力强等特点，已经在自然语言处理中得到广泛应用。在此基础上，现代最先进的模型，如BERT，能够从未标注的文本中学习强大的language representation，甚至在一些很有挑战性的问答任务上超越人类。但它需要大量计算去实现高性能，比如一个Transformer模型翻译一个长度不超过30个单词的句子需要大约10G 的Mult-Adds。而这不适合受限于硬件资源和电池严格限制的移动应用，比如智能手机，手环，物联网设备等。那么如何减少Transformer的计算量呢？看了上面的HAT我们知道一种办法是通过减少Embedding size 。但是这存在的一个问题是：这样做在减少计算量的同时也削弱了Transformer捕获长距离和短距离关系的能力。

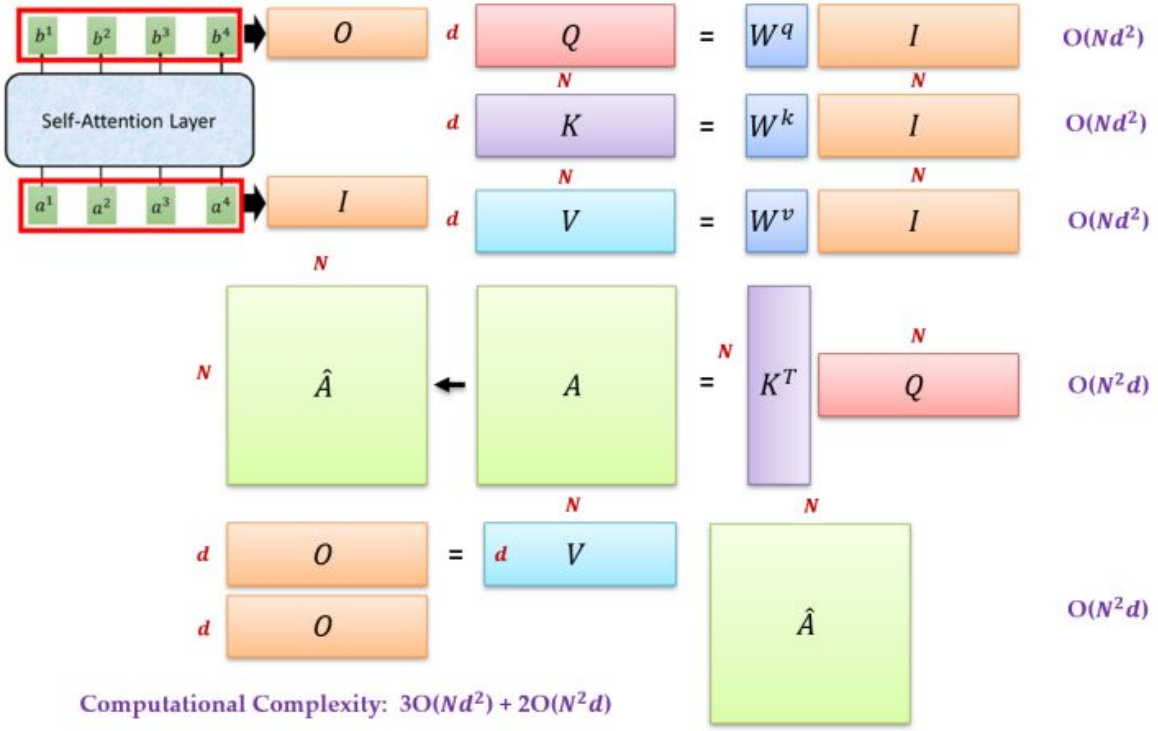
Lite-Transformer这项研究提出了一种高效的模块 —— LSRA，其核心是长短距离注意力（Long-Short Range Attention，LSRA），其中一组注意力头（通过卷积）负责局部上下文建模，而另一组则（依靠注意力）执行长距离关系建模。

这样的专门化配置使得模型在机器翻译、文本摘要和语言建模这3个语言任务上都比原版transformer 有所提升，基于LSRA所构建的Lite Transformer达到了移动设备计算量所要求的500M Mult-Adds。以WMT 2014 English-German任务为例，在计算量限制为500M Mult-Adds或者100M Mult-Adds时，Lite Transformer的性能比原版 Transformer 的 BLEU 值比分别比 transformer 高 1.2或1.7。结合剪枝和量化技术，研究者进一步将 Lite Transformer 模型的大小压缩到原来的 5%。

对于语言建模任务，在大约 500M MACs 上，Lite Transformer 比 transformer 的困惑度低 1.8。值得注意的是，对于移动 NLP 设置，Lite Transformer 的 BLEU 值比基于 AutoML 的 Evolved Transformer 高 0.5，而且AutoML方法所需要的搜索算力超过了250 GPU years，这相当于5辆汽车的终身碳排放量。

Lite-Transformer具体方法：

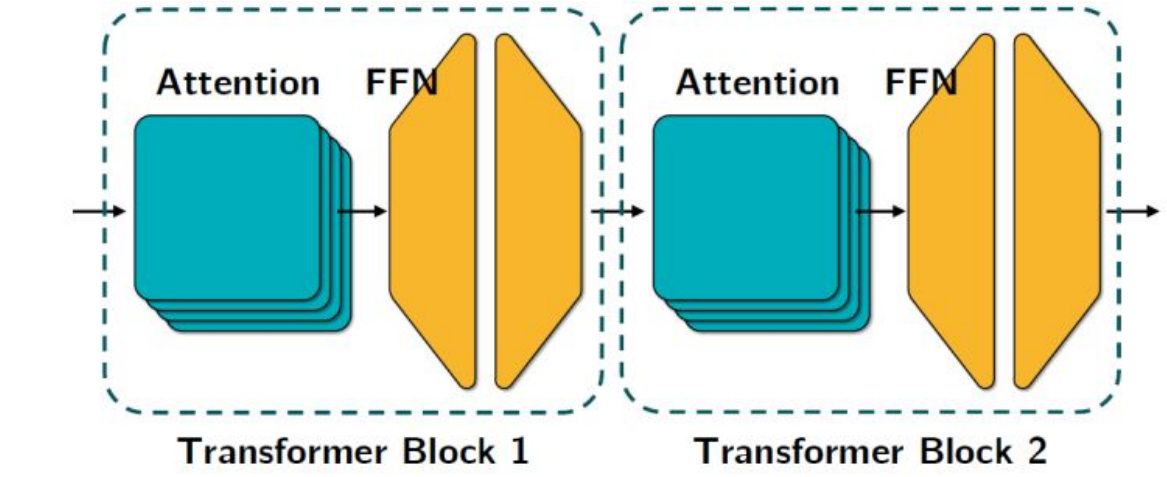
我们将用于文本处理的Self-attention称为1-D attention，用于图像识别的Self-attention称为2-D attention，用于视频处理的Self-attention称为3-D attention。首先看看Self-attention的计算复杂度，如下图所示：

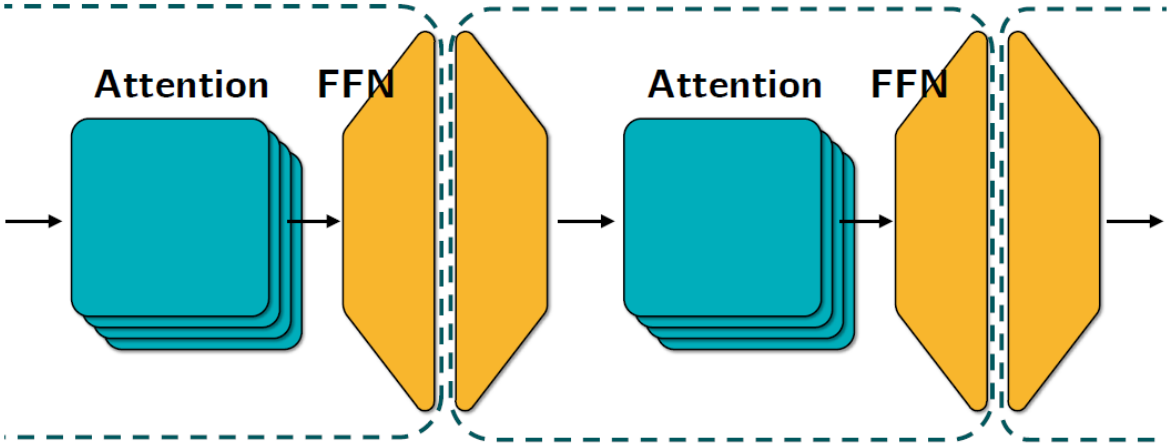


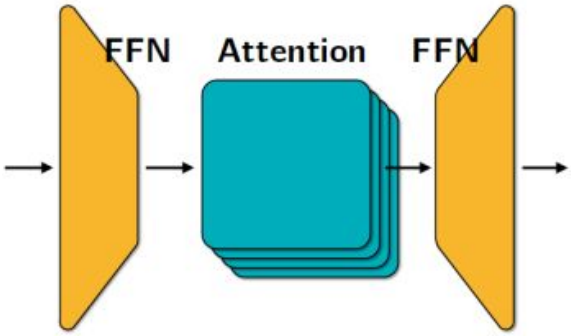
而这样的计算复杂度下就会产生一个问题：当N增大时整个模型的计算量同样也会变得巨大

如何解决这个问题：

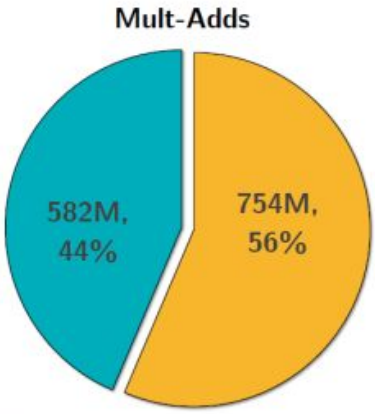
1. 减少Embedding dim来降低计算量——会严重影响Self-attention layer的性能，使得我们无法在保证性能的前提下大幅减少计算量。
2. 设计一种Flattened Transformer Block，它使得特征在进入Self-attention layer之前不进行降维，使得attention layer占据了绝大部分计算量。







Base Transformer Block



Computation Proportion

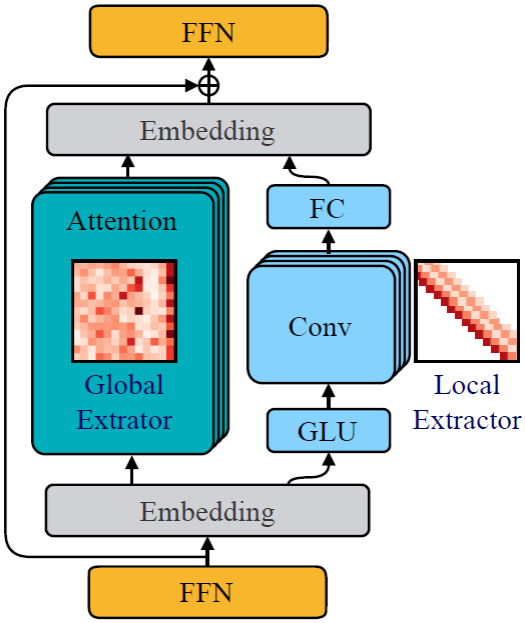
对比一下之前的方法：

之前想通过减少Embedding dim来降低计算量，但是由于 bottleneck design 的缺点，使得Self-attention受到了严重的影响，影响了模型的性能。

现在通过减少LSRA来降低计算量，由于 Flattened Transformer Block，使得计算量可以通过LSRA进行大幅降低而不影响性能。

让Self-attention这个模块更加专门化：

长短距离注意力 (LSRA)哪里专门化呢？在翻译任务中，注意力模块必须捕获全局和局部上下文信息。LSRA 模块遵循两分支设计，如下图所示。左侧注意力分支负责捕获全局上下文，右侧卷积分支则建模局部上下文。研究者没有将整个输入馈送到两个分支，而是将其沿通道维度分为两部分，然后由后面的 FFN 层进行混合。这种做法将整体计算量减少了 50%。



左侧分支处理全局信息：是正常的Self-attention模块，不过通道维度减少了一半。

右侧分支处理局部关系：一个自然的想法是对序列应用卷积。为了进一步减少计算量，研究者将普通卷积替换为轻量级的版本，该版本由线性层linear layers和Depth-wise convolution组成。

实验及评估结果

IWSLT 实验结果：

下图为Lite Transformer 在 IWSLT’14 De-En 数据集上的定量结果。并与 transformer 基线方法和 LightConv 做了对比。在大约 100M Mult-Adds 时，Lite Transformer 模型的 BLEU 值比 transformer 高出 1.6

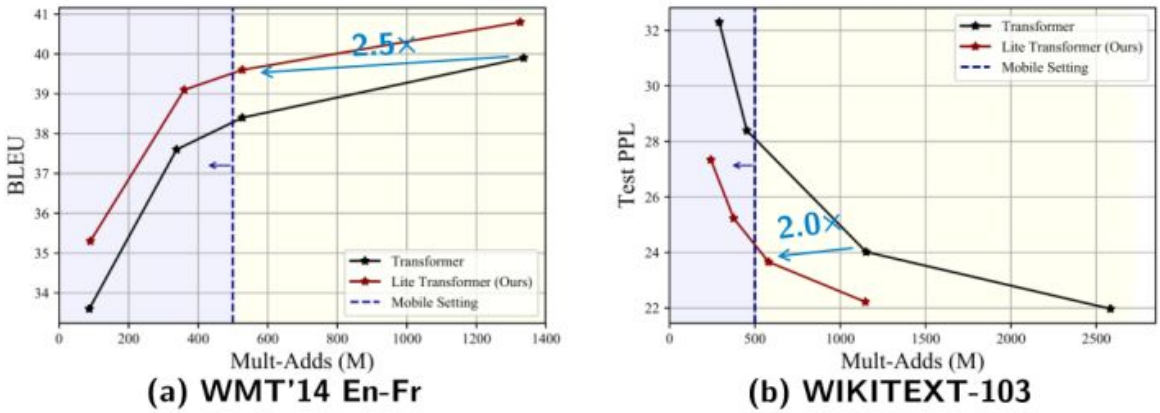
	#Parameters	#Mult-Adds	BLEU	ΔBLEU
Transformer (Vaswani et al., 2017)	2.8M	63M	27.8	–
LightConv (Wu et al., 2019b)	2.5M	52M	28.5	+0.7
Lite Transformer (Ours)	2.8M	54M	**30.9**	**+3.1**
Transformer (Vaswani et al., 2017)	5.7M	139M	31.3	–
LightConv (Wu et al., 2019b)	5.1M	115M	31.6	+0.3
Lite Transformer (Ours)	5.4M	119M	**32.9**	**+1.6**
Transformer (Vaswani et al., 2017)	8.5M	215M	32.7	–
LightConv (Wu et al., 2019b)	8.4M	204M	32.9	+0.2
Lite Transformer (Ours)	8.9M	209M	**33.6**	**+0.9**

WMT 实验结果：

下图为Lite Transformer 在 WMT’14 En-De and WMT’14 En-Fr 数据集上的定量结果。并与 transformer 基线方法做了对比。 Lite Transformer在总计算量和模型参数量之间实现了更好的平衡。在大约 100M Mult-Adds 时，Lite Transformer 模型的 BLEU 值比 transformer 分别高出了 1.2和1.7；在大约 300M Mult-Adds 时，Lite Transformer 模型的 BLEU 值比 transformer 分别高出了 0.5和1.5

			WMT’14 En-De		WMT’14 En-Fr	
	#Parameters	#Mult-Adds	BLEU	ΔBLEU	BLEU	ΔBLEU
Transformer (Vaswani et al., 2017)	2.8M	87M	21.3	–	33.6	–
Lite Transformer (Ours)	2.9M	90M	**22.5**	**+1.2**	**35.3**	**+1.7**
Transformer (Vaswani et al., 2017)	11.1M	338M	25.1	–	37.6	–
Lite Transformer (Ours)	11.7M	360M	**25.6**	**+0.5**	**39.1**	**+1.5**
Transformer (Vaswani et al., 2017)	17.3M	527M	26.1	–	38.4	–
Lite Transformer (Ours)	17.3M	527M	**26.5**	**+0.4**	**39.6**	**+1.2**

WMT En-Fr数据集实验结果的trade-off曲线如下图所示



(a) WMT’14 En-Fr　　(b) WIKITEXT-103

与 Evolved Transformer 对比：

相比 Evolved Transformer，在大约 100M Mult-Adds 时，Lite Transformer 模型的 BLEU 值比 Evolved transformer 高出了 0.5；在大约 300M Mult-Adds 时，Lite Transformer 模型的 BLEU 值比 Evolved transformer 高出了 0.2

	#Params	#Mult-Adds	BLEU	GPU Hours	CO_2e (lbs)	Cloud Computation Cost
Transformer (Vaswani et al., 2017)	2.8M	87M	21.3	8×12	26	$68 - $227
Evolved Transformer (So et al., 2019)	3.0M	94M	22.0	8×274K	626K	$1.6M - $5.5M
Lite Transformer (Ours)	2.9M	90M	**22.5**	8×14	32	$83 - $278
Transformer (Vaswani et al., 2017)	11.1M	338M	25.1	8×16	36	$93.9 - $315
Evolved Transformer (So et al., 2019)	11.8M	364M	25.4	8×274K	626K	$1.6M - $5.5M
Lite Transformer (Ours)	11.7M	360M	**25.6**	8×19	43	$112 - $376

本文的贡献

- 发现bottleneck design的结构对于1-D attention (文本处理) 来说不是最优的
- 提出一种多分支的特征提取器 Long-Short Range Attention (LSRA)，其中卷积操作帮助捕捉局部上下文，而attention用来捕捉全局上下文
- 基于LSRA所构建的Lite Transformer达到了移动设备计算量所要求的500M Mult-Adds，在3种任务上获得了一致的性能提升，与AutoML-based方法Evolved Transformer相比也获得了性能的提升，并大大减少训练成本