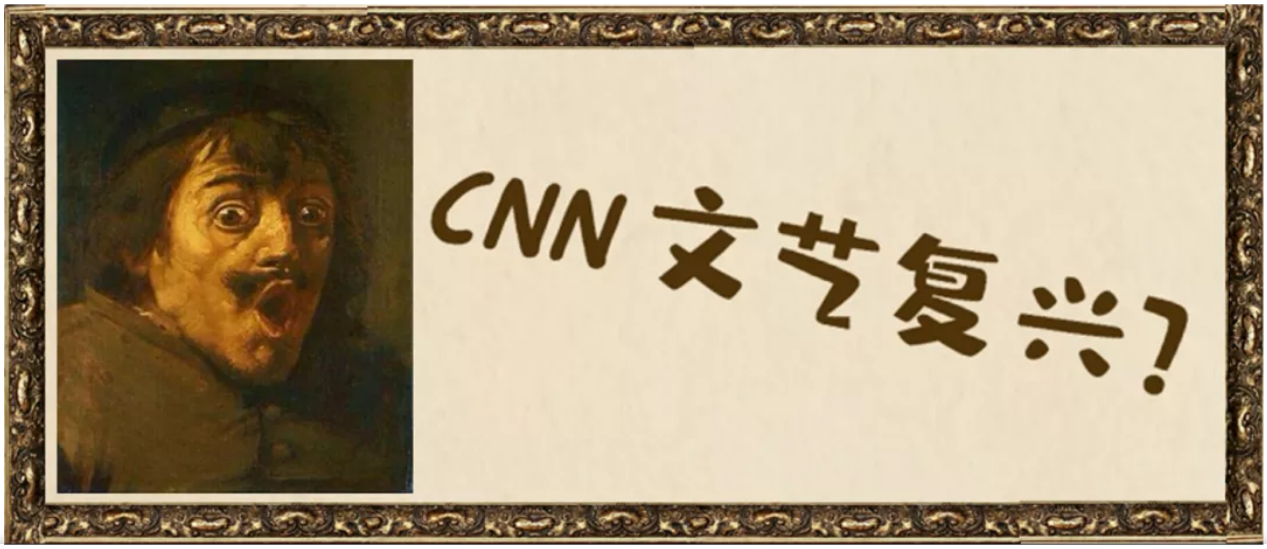
谷歌：CNN击败Transformer，有望成为预训练界新霸主！LeCun却沉默了...

原创 ivem 夕小瑶的卖萌屋 2021-05-11 15:00



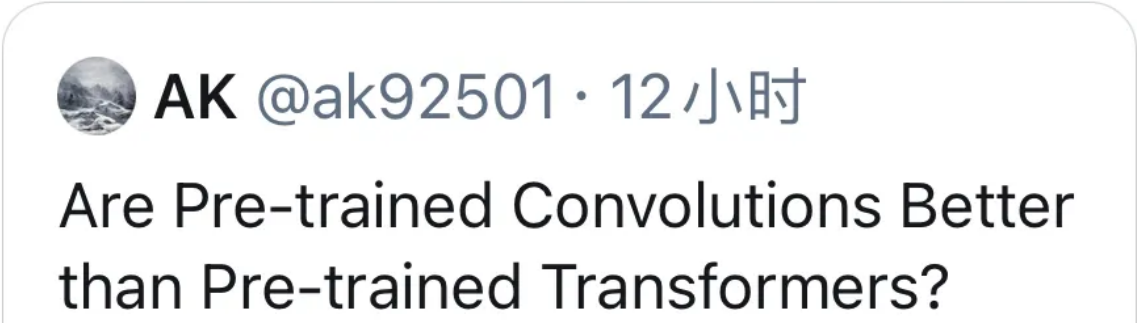
文 | ivem

这几年，大家都说深度学习进入了预训练时代。作为一个入行不久的小白，我一直以为各类基于 Transformers 结构的预训练模型是 NLP 的巨大里程碑，CNN、RNN 老矣，只配作为手下败将。大家的文章似乎也是这样：把 BERT 作为 encoder 的归为一组来对比性能，把 CNN、RNN encoder 归为另一组，且总是要比基于 BERT 的差一些。

但是，我们有没有想过一个问题？当今所有预训练模型都是基于 transformers 结构的模型，**我们使用预训练模型提升下游任务性能，是因为使用海量数据预训练，还是因为 transformers 的结构呢？**

今天这篇文章就使用卷积模型进行预训练，并且在几个下游任务 fine-tune，性能和基于 transformers 的预训练模型相当（甚至更高）。作者认为，这样的好结果加上卷积操作本身更小的复杂度，pre-trained convolutions 简直是在性能和效率上将 transformers 完爆！

然而，Yann LeCun 对这篇文章却做出了很暧昧的评价：



相信这两天，大家也都被这篇 Google 的 ACL 和 LeCun 的评价刷屏，但 LeCun 为什么会这样评价？这是正面评价还是负面评价？

笔者看完这篇文章之后，也有一种意犹未尽的感觉：这个问题确实有待进一步研究。下面就容我细细道来。

论文题目:

Are Pre-trained Convolutions Better than Pre-trained Transformers?

论文链接:

https://arxiv.org/pdf/2105.03322.pdf

模型

这部分将详细介绍整体的卷积预训练模型。这篇文章并没有直接采用最原始的卷积操作，而是采用了 [1] 中改进的卷积。因此，让我们先了解一下这里的卷积操作。为严谨起见，下文中的 CNN 均特指在文本序列上的一维卷积。

卷积模块

CNN 与 self-attention 都可以理解为对 token 的聚合。self-attention 在以下方面比 CNN 更好：

1. CNN 与 self-attention 相比，CNN 在单层的感受野大小是有限且固定的，只能通过堆叠层数来增大感受野；self-attention 在一层就可以捕捉所有 token 之间的关系，这对于捕捉长距离依赖非常关键。
2. self-attention 聚合的权重是与输入 token 相关的，而 CNN 的聚合权重是与输入 token 无关的。

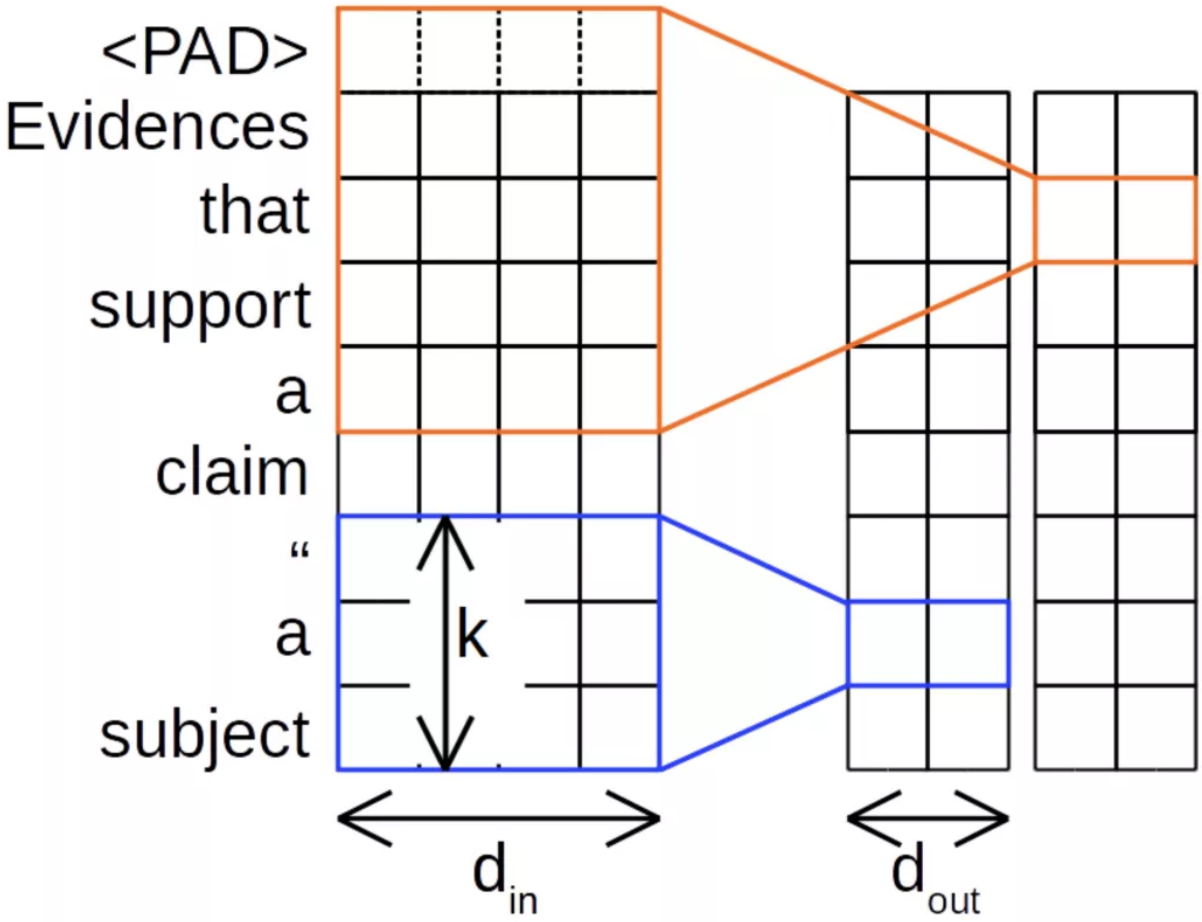
反过来，CNN 也有自己的优势：

1. CNN 比 self-attention 快得多：CNN 是线性复杂度，self-attention 是平方复杂度（甚至因此诞生了《轻量 transformers》这个分支领域）。
2. attention 中的位置编码不断在被改进和完善 [3]；甚至最近有人发现，输入顺序对 transformers 影响很小 [4]，因此位置编码还有待研究。而 CNN 是按顺序进行的，不需要额外的位置编码。

怎样融合二者的优点呢？请看我下面一步步推出 Dynamic Convolution。

Convolutions

我们先来回忆一下传统的 CNN 结构：

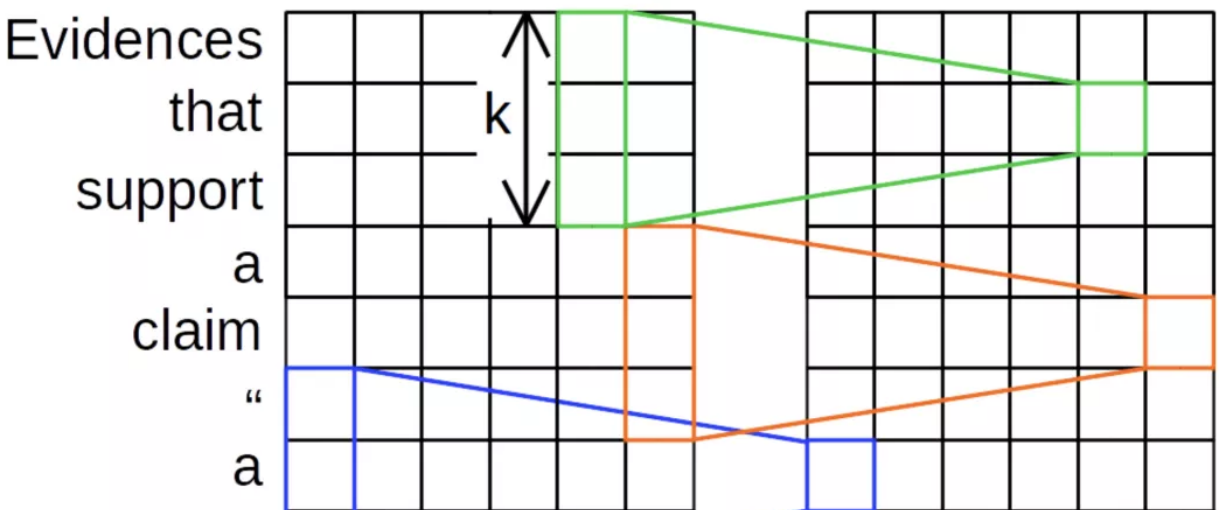


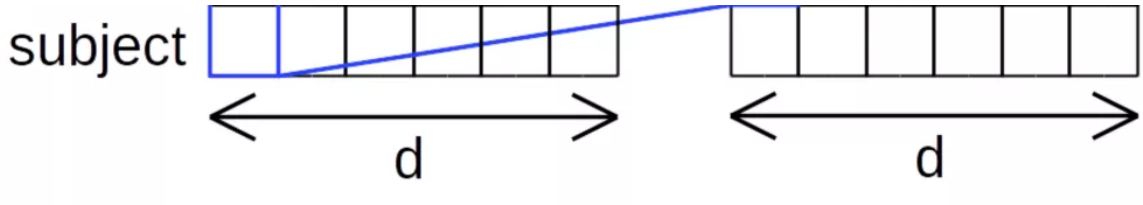
▲传统CNN，图源[2]

传统 CNN 结构如上图所示，不同的卷积核有不同的尺寸，**一个卷积核对输入序列的所有通道进行卷积计算**。

Depthwise Convolutions

深度可分离卷积中，**每个通道只被一个卷积核所卷积**：



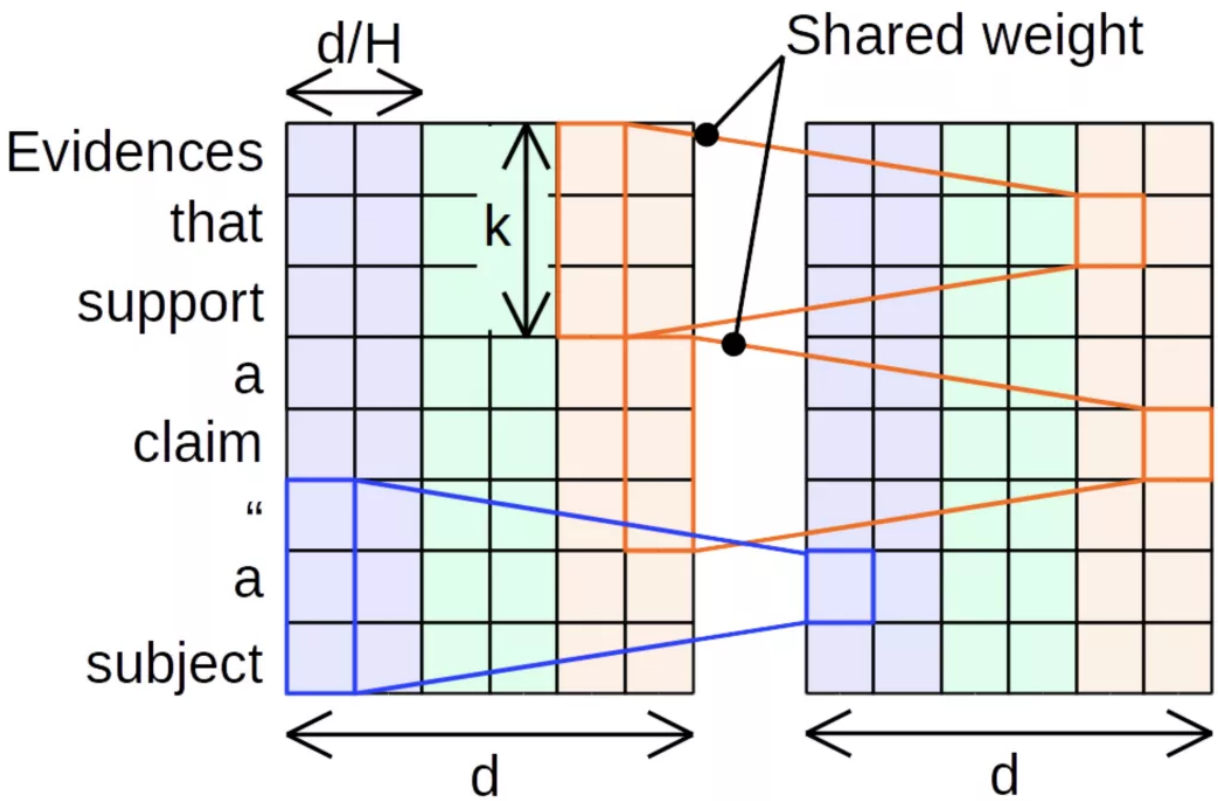


▲Depthwise，图源[2]

例如在上图中，原始序列的第一个通道只与蓝色的卷积核交互，得到输出序列中的第一个维度，其它通道也是同理。这样一来，卷积的计算量将大大减少。

Lightweight Convolutions

轻量化卷积对深度可分离卷积做了进一步地简化：



▲Lightweight，图源[2]

$$\text{LightConv}(X, W_{\lceil \frac{cH}{d} \rceil,:}, i, c) = \text{DepthwiseConv}(X, \text{softmax}(W_{\lceil \frac{cH}{d} \rceil,:}), i, c)$$

首先，**相邻通道的卷积核可进行参数共享**：例如图中相同颜色的通道，其卷积核参数是共享的。

另外，**卷积核参数在其长度的维度上被 softmax 归一化**：

$$\text{softmax}(W)_{h,j} = \frac{\exp W_{h,j}}{\sum_{j=1}^{k} \exp W_{h,j'}}$$

其中，卷积核参数 $W \in \mathbb{R}^{H \times k}$。里面的 H, k 分别是卷积核的数量，以及卷积核的长度。

看到这里，是不是突然发现，这里的归一化和 attention map 的归一化简直一模一样？都是对加权聚合的权重进行归一化！另外，attention 的 multi-head 也可以理解为多个通道的卷积核。这样一来，

self-attention 中的 attention map 归一化和 multi-head 都在卷积中有所体现。

Dynamic Convolutions

动态卷积是对轻量化卷积的进一步改进：

$$\text{DynamicConv}(X, i, c) = \text{L}$$

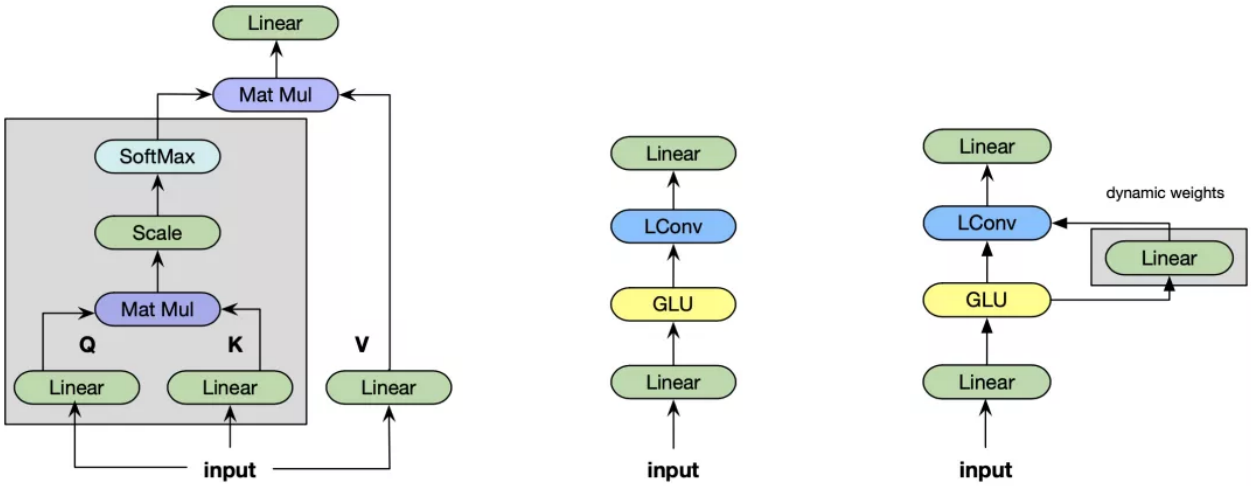
动态卷积通过一个线性映射 $f: \mathbb{R}^d \to \mathbb{R}^{H \times k}$ 使得**卷积核 W 的生成与其顺序输入的不同 token X 有关**，而不是对整个文本序列固定的卷积核。而且，**这里的卷积核参数只与当前被卷积核覆盖的几个 token 相关**，而不像 self-attention 那样，需要与全部 token 交互计算。因此整体上，动态卷积还是线性复杂度。

综上所述，**动态卷积于是很好地模拟了 self-attention 中 attention map 归一化、multi-head，以及权重与输入相关**。本文就分别基于上述的三种卷积操作，搭建卷积预训练模型结构。

卷积预训练模型结构

写到这里实在忍不住吐槽：本文的卷积预训练模型结构依然在模仿基于 transformers 的预训练模型结构，只不过是将其中的 multi-head self-attention 换成了上面说的卷积操作，query-key-value 的结构换成了类似的线性门控（Gated Linear Units[5]）结构。

首先，每个 convolution block 的结构如下图所示：



(a) Self-attention (b) Lightweight convolution (c) Dynamic convolution

Figure 2: Illustration of self-attention, lightweight convolutions and dynamic convolutions.

这里没有使用类似 self-attention 的 query-key-value 的结构（上图的 a），而是使用了一种“线性门控 + 卷积 + 线性映射”的结构（上图的 bc）：

$$X^1 = W^I X \odot \text{sigmoid}(W^S X)$$
$$X^2 = \text{ConvBlock}(X^2)$$

$$X^3 = W^O(X^2)$$

这里 W^I, W^S, W^O 都是可以学习的参数。实验中的卷积使用了上文说的轻量化卷积、动态卷积，以及空洞卷积 [6]。

对于整体的卷积预训练模型，本文也是使用类似 transformers 的方式将 convolution blocks 进行堆叠：

$$X_A = \text{LayerNorm}(\text{ConvBlock}(X)) + X$$
$$X_B = \text{LayerNorm}(\text{FFN}(X_A)) + X_A$$

其中 $\text{ConvBlock}(\cdot)$ 是上文提到的 convolution block，$\text{FFN}(\cdot)$ 是两层全连接网络，后面接一个 ReLU。

实验

模型在 Colossal Cleaned CommonCrawl Corpus (C4) 数据集上进行了预训练。预训练时，模型的 seq2seq 的结构、MLM 任务依然是模拟 transformers；层数、序列长度等参数也与 BART-base 保持了一致。

在实验部分，这篇文章希望探究如下五个问题：

1. 卷积也能在预训练中获益，学到丰富的先验知识吗？
2. 卷积预训练和 transformers 相比，性能怎么样？
3. 卷积预训练和 transformers 相比，有什么优点？会更快吗？
4. 什么情景下，卷积预训练会失败？
5. 不同的卷积模块之间，有很大的差别吗？

下游任务

这篇文章在非常多下游任务上进行了实验，在一些任务上性能追平了基于 transformers 的 BART 或 T5：

Model	CIVILCOMMENT Acc	CIVILCOMMENT F1	WIKITOXIC Acc	WIKITOXIC F1	IMDb Acc	SST-2 Acc	S140 Acc	TREC Acc	News Acc
No pre-training									
Trans.	77.22	85.09	91.93	95.45	84.81	78.44	58.84	78.00	84.25
Light	78.58	85.82	91.05	94.65	**85.88**	81.65	60.64	**82.20**	**87.22**
Dilat.	**79.94**	**86.50**	**92.29**	94.91	85.84	79.01	55.62	79.60	81.24
Dyna.	78.49	84.71	90.06	**95.66**	85.69	**82.80**	**60.84**	80.20	85.13
With pre-training									
Trans.	81.16	86.56	91.46	95.12	**94.16**	92.09	61.65	93.60	93.54
Light	81.47	87.58	93.61	96.48	93.60	**92.20**	61.65	93.60	93.63
Dilat.	81.67	**87.78**	**93.84**	96.21	93.92	92.09	**62.85**	**94.20**	93.26
Dyna.	**81.83**	87.71	93.76	**96.53**	93.35	91.59	62.45	92.40	**93.93**

	Gain from pre-training								
Trans.	+5.1%	+1.7%	-0.6%	-0.4%	+11.0%	+17.4%	+4.7%	+20.0%	+11.0%
Light	+3.7%	+2.1%	+2.8%	+1.9%	+9.0%	+13.0%	+1.7%	+14.0%	+7.3%
Dilat.	+2.1%	+1.5%	+1.7%	+1.4%	+9.4%	+17.0%	+13.0%	+18.0%	+14.8%
Dyn.	+4.3%	+3.5%	+4.1%	+1.0%	+8.9%	+10.6%	+2.6%	+15.2%	+10.4%

Table 2: Comparison of pre-trained Convolutions and pre-trained Transformers on toxicity detection, sentiment classification, question classification and news classification. All models have approximately $230M$ parameters and are 12 layered seq2seq architectures. Our findings show that convolutions (1) also benefit from pretraining and (2) are consistently competitive to transformer models with and without pretraining.

1. 在攻击性言论检测任务中（CivilComment 和 WikiToxic 数据集），卷积预训练网络均优于 transformers，但是 Lightweight 从预训练得到的提升更高。
2. 在情感分类任务中（IMDb，SST-2 和 S140 数据集），卷积预训练不敌 transformers，但是非常接近。
3. 在问题分类任务中（TREC 数据集），卷积预训练网络大体上优于 transformers，transformers 从预训练得到的提升更高一点。
4. 在新闻分类任务中（News 数据集），卷积预训练网络均优于 transformers，空洞卷积受预训练增益最大。

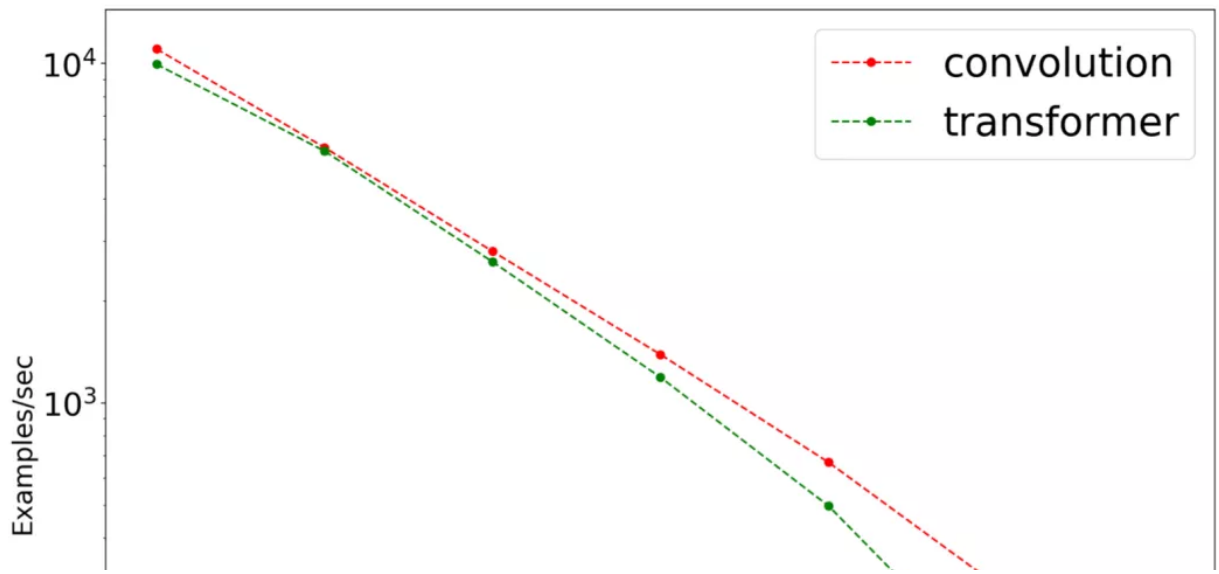
上面的实验**可以回答提出的几个问题**：

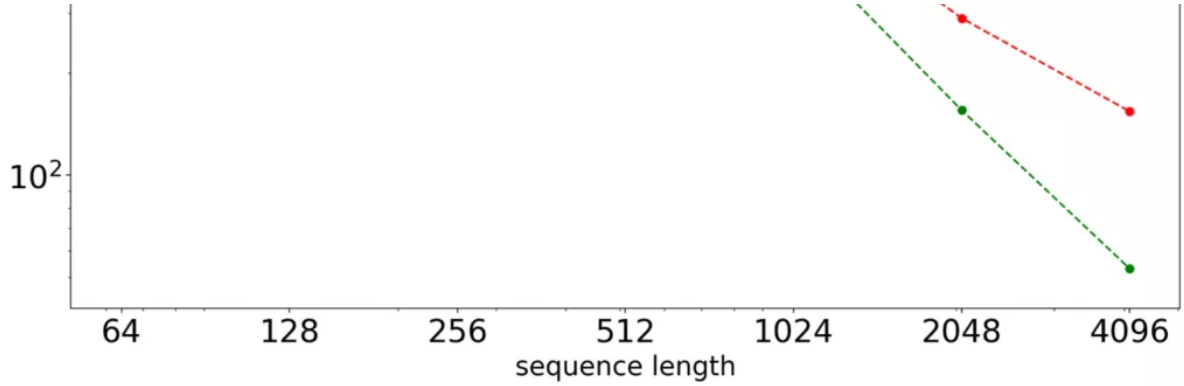
- 问题1：卷积网络也能在预训练中获益，只是不如 transformers 大。
- 问题2：无论是否与训练，卷积的性能优于或与 transformers 一致。
- 问题5：空洞卷积和动态卷积似乎好于轻量化卷积。

其它对比

作者在实验中发现，与训练卷积结构缺少相互的 attention 结构，因此在需要构建多个事物之间关系的任务上，卷积预训练结构似乎并不适合。

另外，卷积预训练模型更快，因此能被运用到更长的序列。随着序列长度的增加，卷积预训练模型的速度优势将更加显著：





总结

现在的预训练是和 transformers 绑定的。因此，BERT、transformers、大规模预训练模型，这些概念似乎被混为了一谈。这篇文章就将 transformers 结构和预训练解耦，希望唤起学术界的注意：是不是其它结构也能在预训练时代大放光彩呢？

个人认为，在某种意义上来讲，这篇文章的卷积操作相当于在模拟 multi-head self-attention；整体的卷积预训练模型也可以说是在模拟 transformers。用这样的预训练模型与基于 transformers 的预训练模型相比，就能得出“transformers 结构不重要，预训练才重要”的结论吗？这是不是还需要进一步研究？

参考文献

[1] Felix Wu, et al., "Pay Less Attention with Lightweight and Dynamic Convolutions", ICLR 2019, https://arxiv-download.xixiaoyao.cn/pdf/1901.10430.pdf

[2] 論文紹介: Pay Less Attention with Lightweight and Dynamic Convolutions, https://qiita.com/koreyou/items/328fa92a1d3a7e680376

[3] Jianlin Su, et al., "RoFormer: Enhanced Transformer with Rotary Position Embedding", arXiv:2104.09864, https://arxiv-download.xixiaoyao.cn/pdf/2104.09864.pdf

[4] Koustuv Sinha, et al., "Masked Language Modeling and the Distributional Hypothesis: Order Word Matters Pre-training for Little", ACL 2021, https://arxiv-download.xixiaoyao.cn/pdf/2104.06644.pdf

[5] Yann N. Dauphin, et al., "Language Modeling with Gated Convolutional Networks", ICML 2017, https://arxiv-download.xixiaoyao.cn/pdf/1612.08083.pdf

[6] Fisher Yu and Vladlen Koltun, "Multi-Scale Context Aggregation by Dilated Convolutions", ICLR 2016, https://arxiv-download.xixiaoyao.cn/pdf/1511.07122.pdf