高效的transformer

2021年1月23日 9:58

1.标准transformer结构

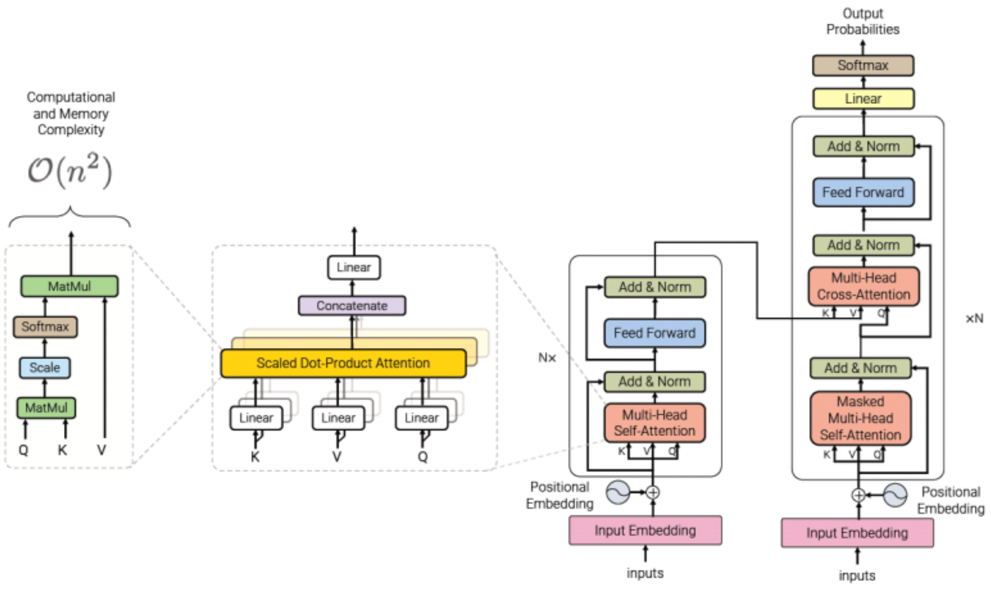


Figure 1: Architecture of the standard Transformer (Vaswani et al., 2017)

ST

2.不同种类的高效transformer

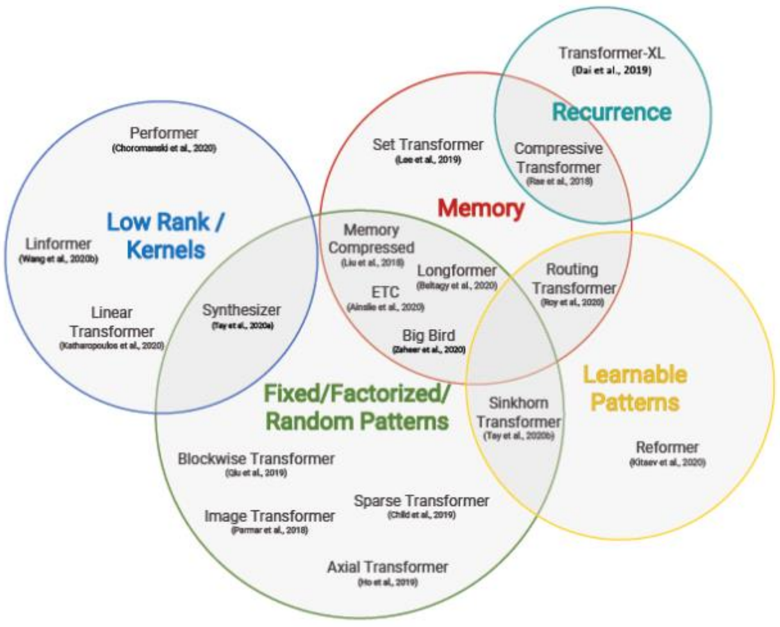
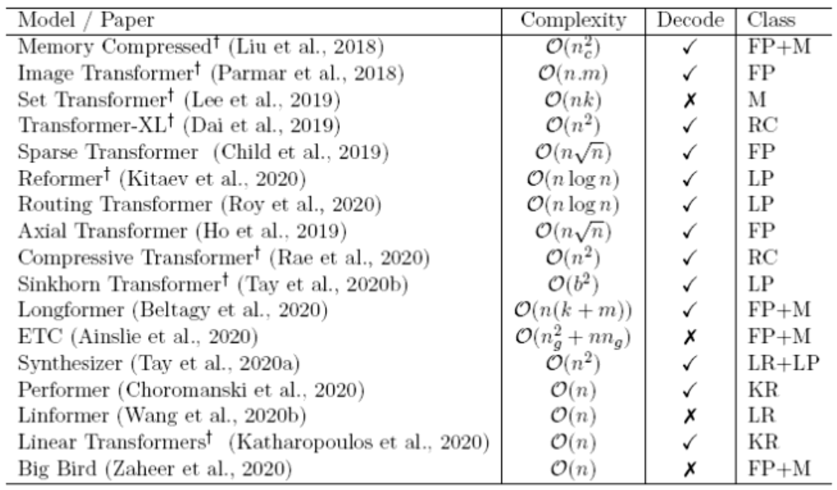

Transformer-XL (Dai et al., 2019)
Recurrence
Compressive Transformer (Rae et al., 2018)
Set Transformer (Lee et al., 2019)
Memory
Performer (Choromanski et al., 2020)
Low Rank / Kernels
Linformer (Wang et al., 2020b)
Linear Transformer (Katharopoulos et al., 2020)
Synthesizer (Tay et al., 2020a)
Memory Compressed (Liu et al., 2018)
Longformer (Beltagy et al., 2020)
ETC (Ainslie et al., 2020)
Big Bird (Zaheer et al., 2020)
Routing Transformer (Roy et al., 2020)
Learnable Patterns
Sinkhorn Transformer (Tay et al., 2020b)
Reformer (Kitaev et al., 2020)
Fixed/Factorized/ Random Patterns
Blockwise Transformer (Qiu et al., 2019)
Sparse Transformer (Child et al., 2019)
Image Transformer (Parmar et al., 2018)
Axial Transformer (Ho et al., 2019)

Figure 2: Taxonomy of Efficient Transformer Architectures.

3.不同种类的transformer复杂度比较



Model / Paper	Complexity	Decode	Class
Memory Compressed† (Liu et al., 2018)	$\mathcal{O}(n_c^2)$	✓	FP+M
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

Table 1: Summary of Efficient Transformer Models presented in chronological order of their first public disclosure. Some papers presented sequentially may first appear at the same time, e.g., as an ICLR submission. Papers annotated with a superscript † are peer-reviewed papers. Class abbreviations include: FP = Fixed Patterns or Combinations of Fixed Patterns, M = Memory, LP = Learnable Pattern, LR = Low Rank, KR = Kernel and RC = Recurrence. Furthermore, n generally refers to the sequence length and b is the local window (or block) size. We use subscript g on n to denote global memory length and n_c to denote convolutionally compressed sequence lengths.

(1) Fixed Patterns (FP)：

a. Blockwise Patterns:EMNLP2019《Blockwise Self-Attention for Long Document Understanding》引入稀疏mask矩阵，将输入文本分块，只考虑块的attention
b. Strided Patterns：ICLR2020《Sparse Transformer: Concentrated Attention Through Explicit Selection》将attention过小的value置0（注意力排序topK个进行保留）arXiv2004《The long-document transformer》滑动窗口（只对相邻的几个token做attention（实验中设置的竟然是512)，空洞滑动窗口（计算机视觉的技术，attention的token之间存在空隙），融合全局信息的空洞滑动窗口（例如[cls]可以attend到所有序列信息)
c. Compressed Patterns:使用池化或者下采样等方式降低序列长度和复杂度:ICLR2018《Generating wikipedia by summarizing long sequences》

(2)Combination of Patterns (CP)主要思想是组合

上面提到的Sparse Transformer将strided和local注意力结合在了一起。

《Axial attention in multidimensional transformers》

（3）Learnable Patterns (LP)：

ICLR2020《Reformer: The efficient transformer》：基于哈希相似度度量的方法,将相似的token放入一个桶中。

《Efficient contentbased sparse attention with routing transformers》:使用一种类似于on-line k-means的方法将token进行聚类。

（4）memory：

ICML2019《Set transformer: A framework for attention-based permutation-invariant neural networks》

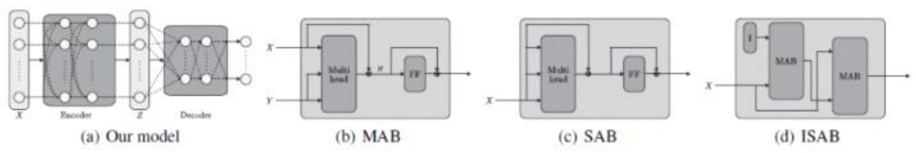


Figure 1. Diagrams of our attention-based set operations.

《The long-document transformer》

（5）Low-Rank Methods：从NxN的矩阵中寻找低秩的有用信息。

arXiv2020 Lin-former《Synthesizer:Rethinking self-attention in transformer models》:将k和v线性映射到低维空间，降低复杂度

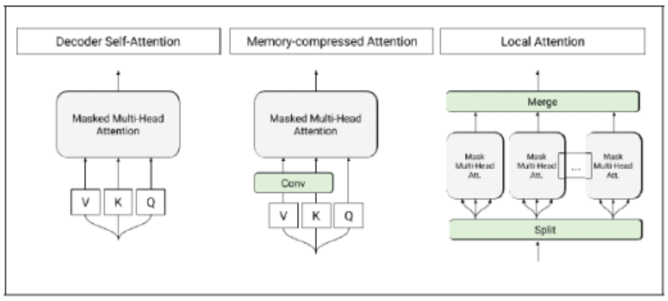
（6）Kernels：将NxN的注意力矩阵进行数学化的近似，方便计算，相当于一种低秩分解操作。

（7）Recurrence：可以视作一种fixed pattern，利用注意力重现的方式克服长期依赖。

4、经典的transformer高效变种

(1) Memory compressed transformer

Peter J Liu, Mohammad Saleh, Etienne Pot, Ben Goodrich, Ryan Sepassi, Lukasz Kaiser, and Noam Shazeer. Generating wikipedia by summarizing long sequences.ICLR2018



Decoder Self-Attention
Memory-compressed Attention
Local Attention
Masked Multi-Head Attention
Masked Multi-Head Attention
Conv
V
K
Q
Merge
Mask Multi-Head Att.
Split

Figure 1: The architecture of the self-attention layers used in the T-DMCA model. Every attention layer takes a sequence of tokens as input and produces a sequence of similar length as the output. **Left:** Original self-attention as used in the transformer-decoder. **Middle:** Memory-compressed attention which reduce the number of keys/values. **Right:** Local attention which splits the sequence into individual smaller sub-sequences. The sub-sequences are then merged together to get the final output sequence.

主要是两点贡献：

a. Local Attention Span:一种直观的想法，将transformer的注意力锁定在一个相对固定长度的span中，每个span相互独立（这种方式虽然能降低复杂度，但对长期依赖来说是致命的缺点）实验中设置的是256个token为一个block
b. Memory-compressed Attention：想法是使用一定步长的卷积操作来降低keys和values的size，模型压缩大小取决于卷积步长，并且相对于局部注意力获得全局的信息。实验中设置的步长为3，卷积核大小3
c. 实验中设置的堆叠方式是LMLML（L代表局部注意力，M代表内存压缩注意力）

(2) Set transformer

Juho Lee, Yoonho Lee, Jungtaek Kim, Adam Kosiorek, Seungjin Choi, and Yee Whye Teh.
Set transformer: A framework for attention-based permutation-invariant neural networks.
In International Conference on Machine Learning, pages 3744{3753, 2019.

a. 适用于一些无序的输入包括集合，特征，排列组合等形式
b. Attention Blocks：

$$
\begin{aligned}
\mathbf{MAB}(\mathbf{X},\mathbf{Y}) &:= \text{LayerNorm}\left(H + \text{rFF}(H)\right),\\
H &:= \text{LayerNorm}\left(X + \text{Multihead}(X,Y,Y)\right),\\
\mathbf{SAB}(\mathbf{X}) &:= \text{MAB}(X,X),\\
\mathbf{ISAB_m}(\mathbf{X}) &:= \text{MAB}\left(X, \text{MAB}(I_m, X)\right).\\
\mathbf{PMA_k}(\mathbf{X}) &:= \text{MAB}\left(S_k, \text{rFF}(X)\right).
\end{aligned}
$$

Multihead Attention Block(MAB), Set Attention Block (SAB),Induced Set Attention Block (ISAB), Pooling by Multihead Attention(PMA).

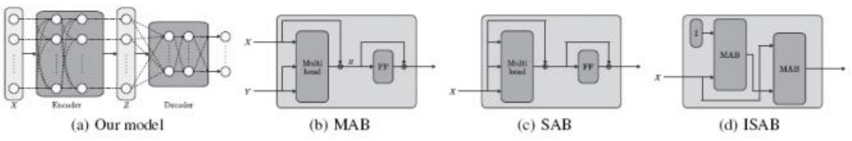


Figure 1. Diagrams of our attention-based set operations.

$$
\text{Encoder}(X) = \text{SAB}(\text{SAB}(X))
$$

$$
\text{Encoder}(X) = \text{ISAB}_m(\text{ISAB}_m(X)).
$$

$$
\text{Decoder}(Z;\lambda) = \text{rFF}(\text{SAB}(\text{PMA}_k(Z))) \in \mathbb{R}^{k\times d} \quad (15)
$$

$$
\text{where } \text{PMA}_k(Z) = \text{MAB}(S, \text{rFF}(Z)) \in \mathbb{R}^{k\times d}, \quad (16)
$$

c. 直接使用SAB复杂度为O(n^2）因此使用Induced Set Attention Block，其中I的维度为m低于X，X相当于先降维，保留重要的特征，再重建X，具有AE的思想，复杂度为O（mn）。
在最大值回归、计数不同字符、混合高斯、集合异常检测和点云分类五个任务上有较好表现

(3) Sparse transformer

Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever. Generating long sequences
with sparse transformers. arXiv preprint arXiv:1904.10509, 2019.

主要思想是将原有的transformer的O(n^2)复杂度通过计算稀疏元素的注意力来降低。主要有两种方式：

a. Local Attention Heads

$$
\hat{A}_{ij} = \begin{cases} Q_i(K)_j^\top), & \text{if } \lfloor j/N \rfloor = \lfloor i/N \rfloor \\ 0 & \text{otherwise} \end{cases}
$$

where A_{ij} is the attention weight of q_i, k_j and $\lfloor\ \rfloor$ denote the floor operation. In this case, we only compute the attention if $\lfloor j/N \rfloor = \lfloor i/N \rfloor$ (within the same block).

b. Strided Attention Heads

$$
\hat{A}_{ij} = \begin{cases} Q_i(K)_j^\top), & \text{if } (i-j) \mod N = 0 \\ 0 & \text{otherwise} \end{cases}
$$

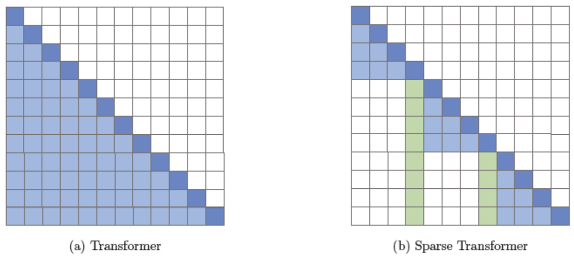


Figure 4: Illustration of patterns of the attention matrix for dense self-attention in Transformers and sparse fixed attention in Sparse Transformers.

(4) Axial transformer

Jonathan Ho, Nal Kalchbrenner, Dirk Weissenborn, and Tim Salimans. Axial attention in multidimensional transformers. arXiv preprint arXiv:1912.12180, 2019.

主要思想是对于高维的向量，只应用注意力到轴上，通常一个轴上的元素数量远远小于所有元素的数量。Axial Transformer实现了编码器和解码器

a. unmask row attention + unmask col attention，分别对行和对列进行attention操作后进行信息整合；
b. unmask row attention + mask col attention，通过整合后获得操作点以上部分所有信息；
c. mask row attention，通过整合获得操作点所在行且在操作点之前的像素点信息。

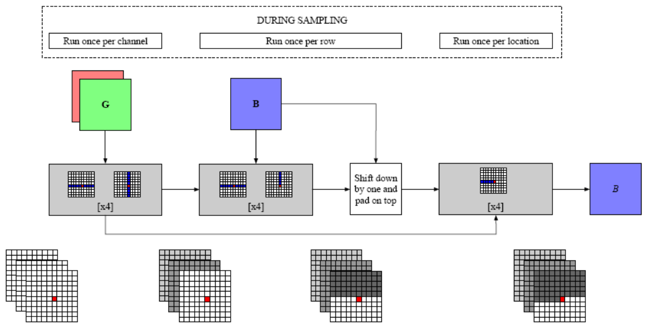


Figure 1: The Axial Transformer model for 2-dimensional tensors. Before sampling a channel we encode all previous channels and frames with 8 blocks of unmasked row and unmasked column attention (left). Then, for each row, we apply 4 blocks of unmasked row and masked column attention to integrate the previously sampled rows for the active channels into our encoded representation (middle). Finally, we shift the encoded representation up to make sure the conditioning information satisfies causality, and we run the inner decoder consisting of 4 blocks of masked row attention to sample a new row in the image (right).

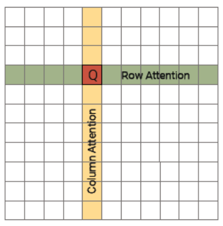


Figure 5: Attention span in Axial Transformer on a two-dimensional input.

(1) Longformer:主要思想和sparse transformer类似，主要的区别是带有空隙的滑动窗口，降低了空间复杂度，同时使用CUDA编程将时间复杂度和普通transformer相同。

Iz Beltagy, Matthew E Peters, and Arman Cohan. Longformer: The long-document transformer. arXiv preprint arXiv:2004.05150, 2020.

a. Global Attention：使用[cls]获得全局的注意力信息。
b. Sliding window attention：就是围绕每一个token采用固定大小的窗口计算局部注意力。
c. Dilated Sliding Window：窗口的缝隙大小是d，假设window size是w，transformer的层数是l，那么窗口能覆盖到的接受范围就是l*d*w。

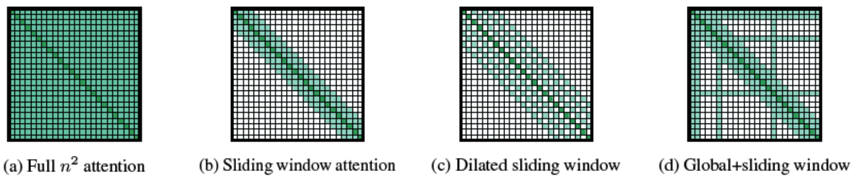

(a) Full n^2 attention
(b) Sliding window attention
(c) Dilated sliding window
(d) Global+sliding window

Figure 2: Comparing the full self-attention pattern and the configuration of attention patterns in our Longformer.

(2) Extended transformer construction(ETC)

Joshua Ainslie, Santiago Ontanon, Chris Alberti, Philip Pham, Anirudh Ravula, and Sumit Sanghai. Etc: Encoding long and structured data in transformers. arXiv preprint arXiv:2004.08483, 2020.

主要思想仍然类似于稀疏注意力，不过结合了多种注意力信息global-to-global (g2g), global-to-local (g2l), local-toglobal(l2g) and local-to-local (l2l).

局限性是无法应用到ar模型中，因为无法计算全局注意力信息g2l

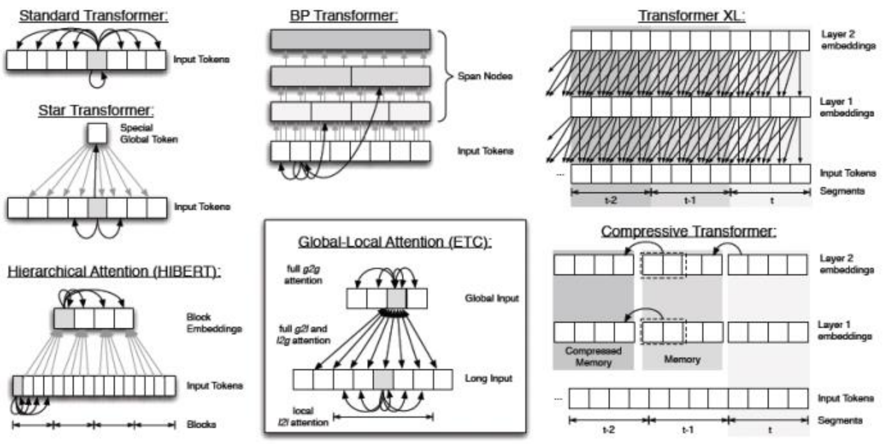
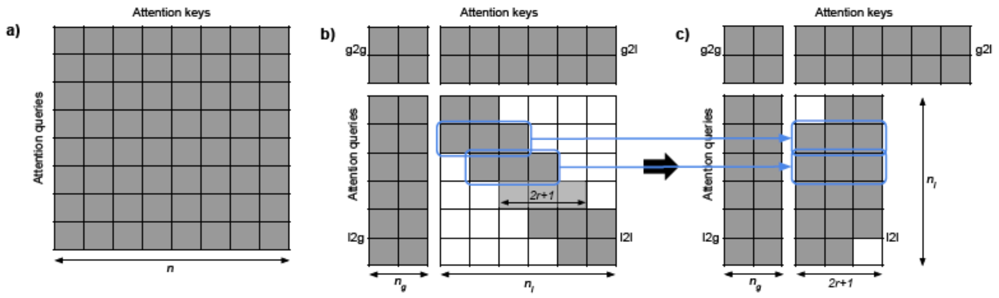


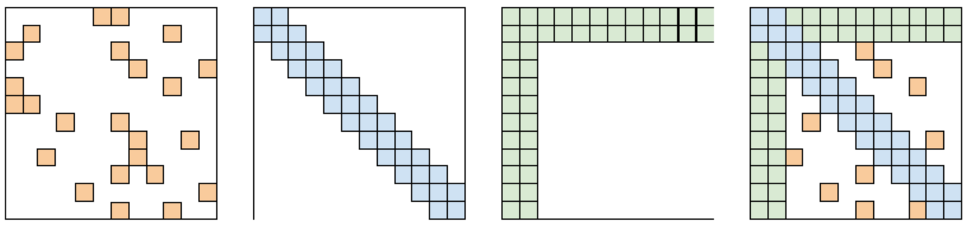
Figure 1: An illustration of mechanisms to scale attention to long inputs, including our proposed model, ETC.



(3) Big Bird

Manzil Zaheer, Guru Guruganesh, Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, et al. Big bird: Transformers for longer sequences. arXiv preprint arXiv:2007.14062, 2020.

建立在etc的基础之上，主要有三个部分组成：(1) global tokens, (2) random attention (queries attend to random keys) and (3) fixed patterns (local sliding windows).



(a) Random attention (b) Window attention (c) Global Attention (d) BIGBIRD

Figure 1: Building blocks of the attention mechanism used in BIGBIRD. White color indicates absence of attention. (a) random attention with $r = 2$, (b) sliding window attention with $w = 3$ (c) global attention with $g = 2$. (d) the combined BIGBIRD model.

NLP问答和摘要任务中超越了SOTA

(4) Reformer

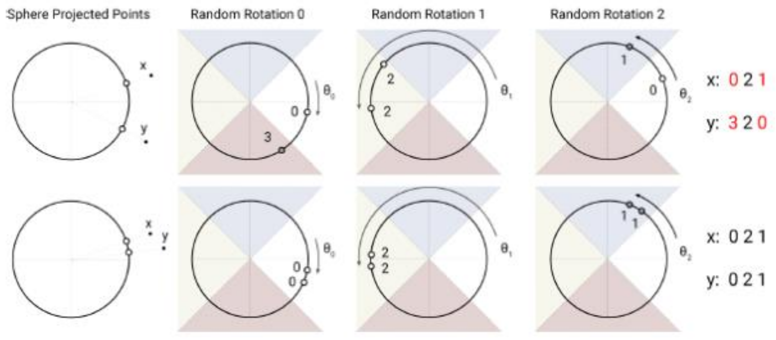


Figure 1: An angular locality sensitive hash uses random rotations of spherically projected points to establish buckets by an argmax over signed axes projections. In this highly simplified 2D depiction, two points x and y are unlikely to share the same hash buckets (above) for the three different angular hashes unless their spherical projections are close to one another (below).

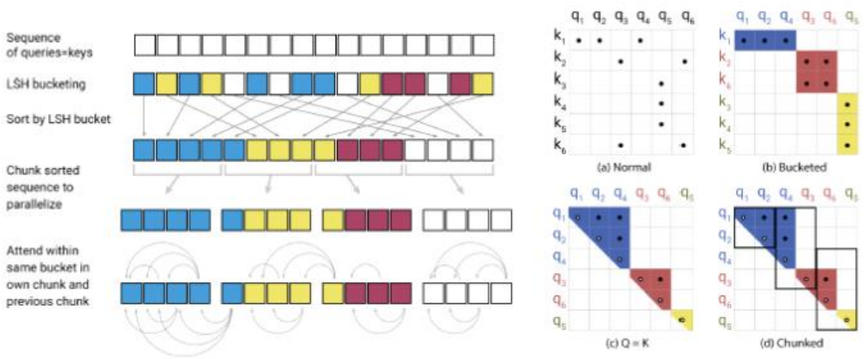


Figure 2: Simplified depiction of LSH Attention showing the hash-bucketing, sorting, and chunking steps and the resulting causal attentions. (a-d) Attention matrices for these varieties of attention.

(5) Sinkhorn transformer

(6) Linformer

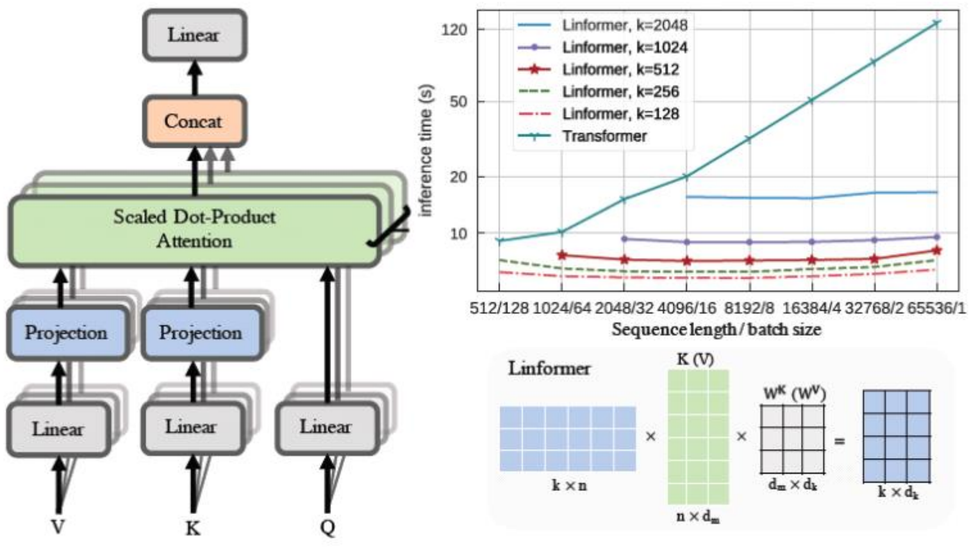


Figure 2: Left and bottom-right show architecture and example of our proposed multihead linear self-attention. Top right shows inference time vs. sequence length for various Linformer models.

(7) Linear transformer

(8) Performer

(9) Synthesizers

(10) Transformer-XL

(11) Compressive transformer