DeepSeek-V3.2-Exp技术解读
从**稀疏注意力**到**DSA**
DeepSeek Sparse Attention

东川路第一可爱猫猫虫

DeepSeek-V3.2-Exp: Boosting Long-Context Efficiency with DeepSeek Sparse Attention

DeepSeek-AI

research@deepseek.com

主要内容

- 我们为什么需要稀疏注意力
- 稀疏注意力（Openai2019）
- 稀疏注意力的变体（CVPR2024）
- DeepSeek Sparse Attention （DeepSeek2025）

我们为什么需要稀疏注意力

- Self attention的时间复杂度

  对序列中任意两个token要计算attention得分

  时间复杂度

  $$O(N^2D)$$

  空间复杂度

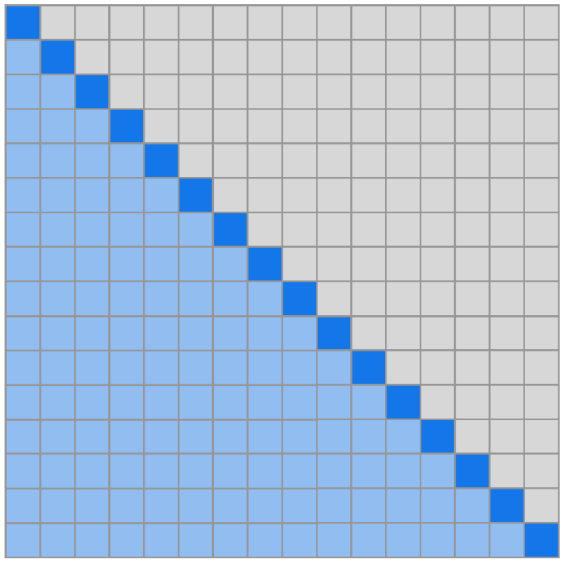
  $$O(N^2)$$

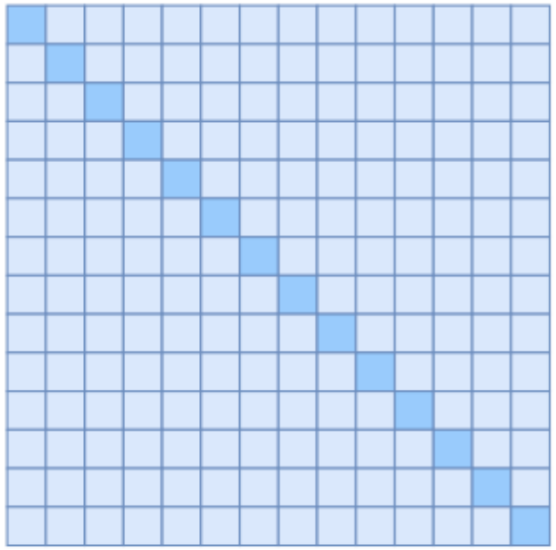
- 显存和算力需求大

  无法用于长序列

稀疏注意力（Openai 2019）

- 《Generating Long Sequences with Sparse Transformers》
- 两种：strided跨步稀疏注意力和fixed固定稀疏注意力
- 跨步稀疏注意力=跳跃Atrous（局部头）+前后Local（稀疏头）
- Self Attention

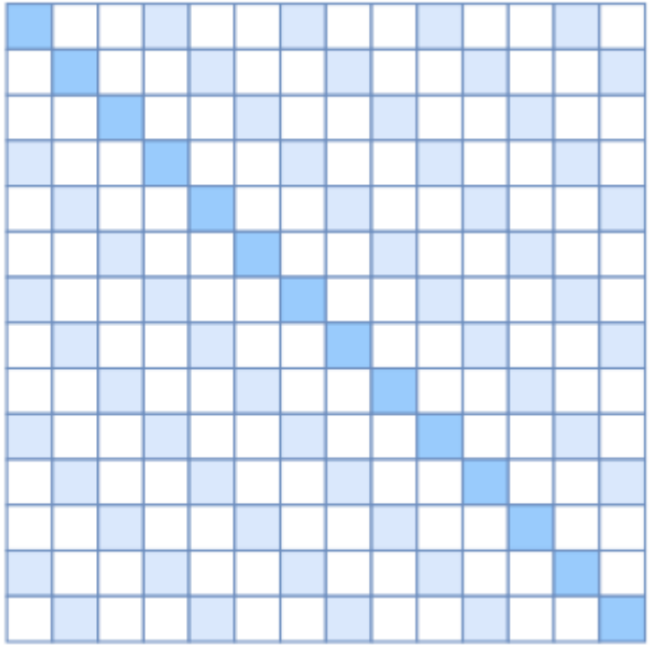




Atrous Self Attention （苏剑林 2019）

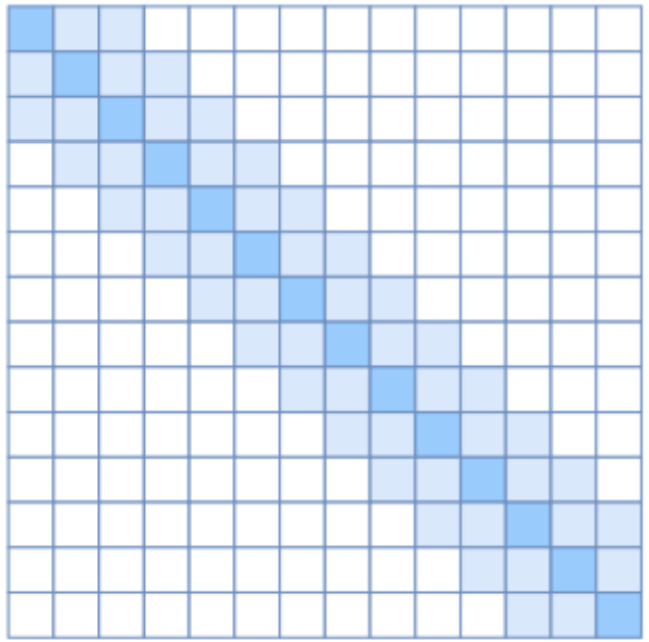
- 每个元素只跟它相对距离为k,2k,3k,…的元素关联
- 至于其他的token，我们不再计算注意力
- 每个元素只与$\frac{n}{k}$个算相关性
- 这样，时间和空间复杂度

  都降低到原来的$\frac{1}{k}$



Local Self Attention （苏剑林 2019）

- 每个元素只与前后k个元素以及自身有关联
- 超出这个范围的token不计算注意力
- 本质是加了权重的卷积



跨步稀疏注意力
=跳跃Atrous（局部头）+前后Local（稀疏头）

Formally, $A_i^{(1)} = \{t, t+1, ..., i\}$ for $t = \max(0, i-l)$ and $A_i^{(2)} = \{j : (i-j) \bmod l = 0\}$. This pattern can be visualized in Figure 3(b).

- 局部头

  关注$(i-l, i)$里的token　l就是跨步的步长

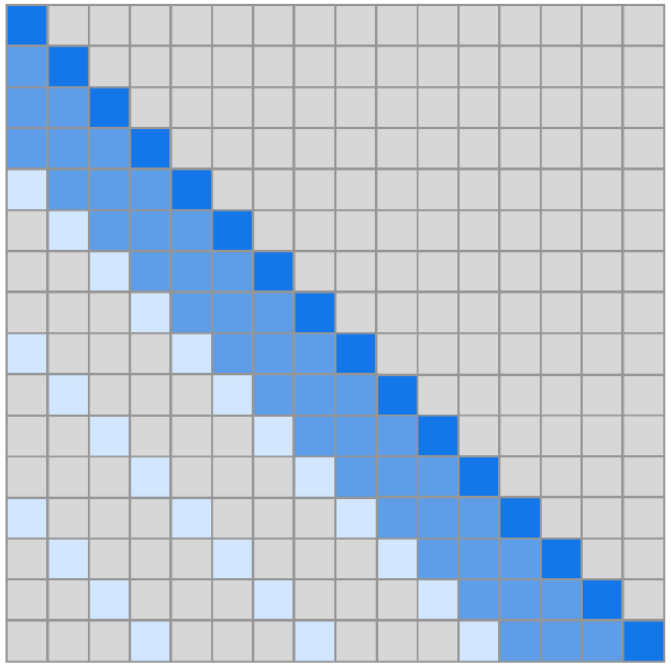
  若$i-l<0$，则从序列位置0开始

- 稀疏头

  关注 “与i的间隔为l的整数倍” 的历史 token

  每隔l个位置选一个 token

  跳跃 固定跳跃间隔



固定稀疏注意力

Formally, $A_i^{(1)} = \{j : (\lfloor j/l \rfloor = \lfloor i/l \rfloor)\}$, where the brackets denote the floor operation, and $A_i^{(2)} = \{j : j \bmod l \in \{t, t+1, ..., l\}\}$, where $t = l - c$ and c is a hyperparameter.

- 局部头

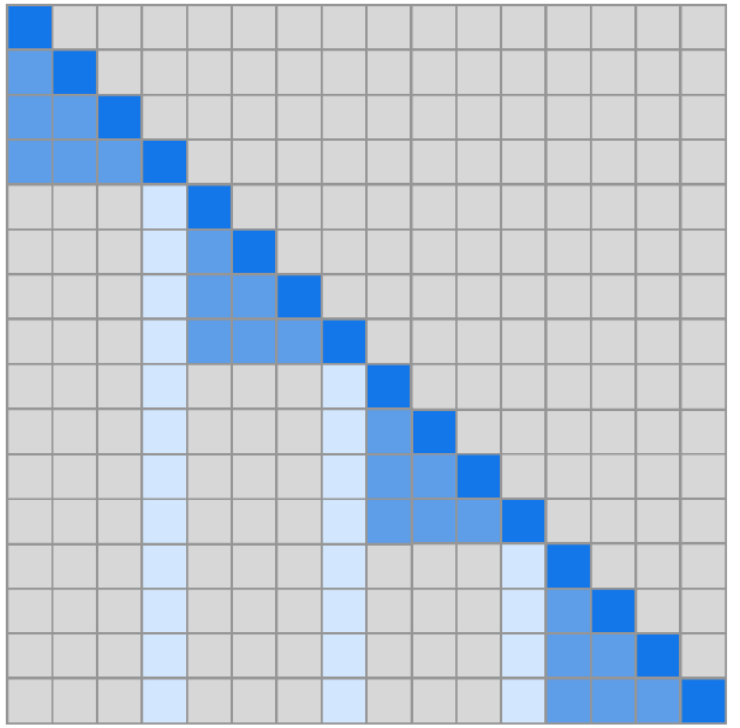
  关注区块内的所有token

  向下取整求出i所在的区块

- 稀疏头

  不关注每个区块的所有 token

  仅保留 最后c个关键 token



ACC-ViT （CVPR 2024）

- 成倍跳跃的稀疏注意力
- Atrous Convolution
- 空洞卷积

titioning of the featuremaps. For a given input featuremap $x \in \mathcal{R}^{c,h,w}$, where c, h, w refers to the number of channels, height, and width, respectively, we can compute windows at different levels of dilation as,

$$x_{atr-k}[c, i, j] = x[c, i + 2k - 1, j + 2k - 1] \quad (1)$$

For example,

$$x_{atr-1}[c, i, j] = x[c, i + 1, j + 1] \quad (2)$$

$$x_{atr-2}[c, i, j] = x[c, i + 3, j + 3] \quad (3)$$

$$x_{atr-3}[c, i, j] = x[c, i + 7, j + 7] \quad (4)$$

DeepSeek Sparse Attention

- Lightning Indexer 闪电索引器

  轻量级计算token间的索引分数，筛选关键token

$$I_{t,s} = \sum_{j=1}^{H^I} w_{t,j}^I \cdot \mathrm{ReLU}\left(\mathbf{q}_{t,j}^I \cdot \mathbf{k}_s^I\right)$$

- 对每个查询tokenh_t与他之前的所有token

- 用FP8精度

DeepSeek Sparse Attention

- fine-grained token selection细粒度token选择

  仅用选中的k个token来计算注意力

  $O(L^2) \rightarrow O(Lk)$

- 计算注意力

  $$\mathbf{u}_t = \mathrm{Attn}\left(\mathbf{h}_t, \left\{\mathbf{c}_s \,\middle|\, I_{t,s} \in \text{Top-k}\left(I_{t,:}\right)\right\}\right)$$

  把选中的top-k个分数对应的c_s与h_t求注意力分数

  这里是MQA 查询头c_s共享同一组 KV 条目

- 成功降低了计算复杂度