MLA
KVCache进阶优化

东川路第一可爱猫猫虫

Multi-head Latent Attention有多好

economical training and efficient inference. It comprises 236B total parameters, of which 21B are activated for each token, and supports a context length of 128K tokens. DeepSeek-V2 adopts innovative architectures including Multi-head Latent Attention (MLA) and DeepSeekMoE. MLA guarantees efficient inference through significantly compressing the Key-Value (KV) cache into a latent vector, while DeepSeekMoE enables training strong models at an economical cost through sparse computation. Compared with DeepSeek 67B, DeepSeek-V2 achieves

(MQA) (Shazeer, 2019). However, these methods often compromise performance in their attempt to reduce the KV cache. In order to achieve the best of both worlds, we introduce MLA, an attention mechanism equipped with low-rank key-value joint compression. Empirically, MLA achieves superior performance compared with MHA, and meanwhile significantly reduces the KV cache during inference, thus boosting the inference efficiency. (2) For Feed-Forward Networks (FFNs), we follow the DeepSeekMoE architecture (Dai et al., 2024), which adopts

However, for both the attention module and the FFN, we design and employ innovative architectures. For attention, we design MLA, which utilizes low-rank key-value joint compression to eliminate the bottleneck of inference-time key-value cache, thus supporting efficient inference. For FFNs, we adopt the DeepSeekMoE architecture (Dai et al., 2024), a high-performance MoE

主要内容

- 从GQA到MLA的低秩投影
- MLA做的改进：结合dot-attention的恒等变换
- MLA遇到的困扰及解决：如何兼容位置编码RoPE
- MLA的小细节：对Q的低秩投影与显存优化

GQA可以看作低秩投影

- 设输入向量$x_i \in R^d$，分组为g，每个组里有$\frac{h}{g}$个头

  $q_i^{(s,t)} = x_i W_q^{(s,t)},\ k_i^{(s)} = x_i W_k^{(s)},\ v_i^{(s)} = x_i W_v^{(s)}$

  如果把所有组的k和v堆叠起来，有：

  $c_i = \left[k_i^{(1)}, k_i^{(2)}, \ldots, k_i^{(g)}, v_i^{(1)}, v_i^{(2)}, \ldots, v_i^{(g)}\right]$

  它实际上等于$x_i \left[W_k^{(1)}, W_k^{(2)}, \ldots, W_k^{(g)}, W_v^{(1)}, W_v^{(2)}, \ldots, W_v^{(g)}\right]$

  隐藏层维度d很大，如5120，而c_i的维度$=g(d_k + d_v) \ll d$

MLA的优化初尝试

- 投影后的GQA
- MLA把简单的线性变换换成一般的线性变换

  GQA的KV是x_i直接投影再复制、分割

  MLA的KV都基于c_i生成
- 好处：

  增加模型的能力
- 背离主题：

  每个头的KV又不一样了，违背了GQA的初衷，增大了KV Cache

结合dot-attention的恒等变换

- 训练阶段照常
- 推理阶段

$$q_t^{(s)} k_i^{(s)T} = \left(x_t W_q(s)\right)\left(c_i W_k^{(s)}\right)^T = x_t \left(W_q^{(s)} W_k^{(s)T}\right) c_i^T$$

MLA把$W_q^{(s)} W_k^{(s)T}$合并，作为新的Query投影阵

ki可被ci替代

V的投影阵$W_v^{(s)}$同理可以吸收到输出层

vi也可被ci替代

- KV Cache 仅需储存共享的c_i

MLA遇到了困扰

- 如果加入RoPE

$$q_t^{(s)} = x_t W_q^{(s)} R_{\mathrm{t}}, k_i^{(s)} = x_i W_k^{(s)} R_i$$

$$q_t^{(s)} k_i^{(s)\mathrm{T}} = x_t \left(W_q^{(s)} R_{t-i} W_k^{(s)T} \right) c_i^T$$

  而我们需要一个固定不变的融合投影矩阵

- 解决

  新增维度

  Q、K新增一些维度用于RoPE

  无RoPE的维度就可以沿用刚才的恒等变换和投影矩阵

As a solution, we propose the decoupled RoPE strategy that uses additional multi-head queries $\mathbf{q}_{t,i}^R \in \mathbb{R}^{d_h^R}$ and a shared key $\mathbf{k}_t^R \in \mathbb{R}^{d_h^R}$ to carry RoPE, where d_h^R denotes the per-head dimension of the decoupled queries and key. Equipped with the decoupled RoPE strategy, MLA performs the following computation:

$$[\mathbf{q}_{t,1}^R; \mathbf{q}_{t,2}^R; ...; \mathbf{q}_{t,n_h}^R] = \mathbf{q}_t^R = \text{RoPE}(W^{QR}\mathbf{c}_t^Q), \tag{14}$$

$$\mathbf{k}_t^R = \text{RoPE}(W^{KR}\mathbf{h}_t), \tag{15}$$

$$\mathbf{q}_{t,i} = [\mathbf{q}_{t,i}^C; \mathbf{q}_{t,i}^R], \tag{16}$$

$$\mathbf{k}_{t,i} = [\mathbf{k}_{t,i}^C; \mathbf{k}_t^R], \tag{17}$$

$$\mathbf{o}_{t,i} = \sum_{j=1}^{t} \text{Softmax}_j\left(\frac{\mathbf{q}_{t,i}^T \mathbf{k}_{j,i}}{\sqrt{d_h + d_h^R}}\right)\mathbf{v}_{j,i}^C, \tag{18}$$

$$\mathbf{u}_t = W^O[\mathbf{o}_{t,1}; \mathbf{o}_{t,2}; ...; \mathbf{o}_{t,n_h}], \tag{19}$$

一个小细节

- 模型训练时的显存占用主要来自两部分

  模型参数（静态）和激活值（动态）
- 激活值的占用是长序列训练的一个重要显存瓶颈

  为了减小激活值显存，对Q做低秩投影

  “哪怕这并不能减小KV Cache”

举个例子看看效果

- 结合 DeepSeek-V2 的具体参数（隐藏维度$d = 5120$、注意力头数$n_h = 128$、单头维度$d_h = 128$、Q 压缩维度$d'_c = 1536$）
- 通过下投影矩阵W^{DQ} 将高维的h_t压缩为低秩的 latent 向量c_t^Q

  单个 Token 的低秩 latent 向量维度为 $d'_c = 1536$

  这样就把输入维度5120压缩到了1536
- MLA 通过上投影矩阵W^{UQ} 将低秩c_t^Q恢复为高维的 Q 向量拼接结果q_t^C，维度为$n_h \times d_h = 16384$
- MLA 在训练时，前向传播仅需存储下投影后的c_t^Q，无需存储上投影后的q_t^C

回到原论文

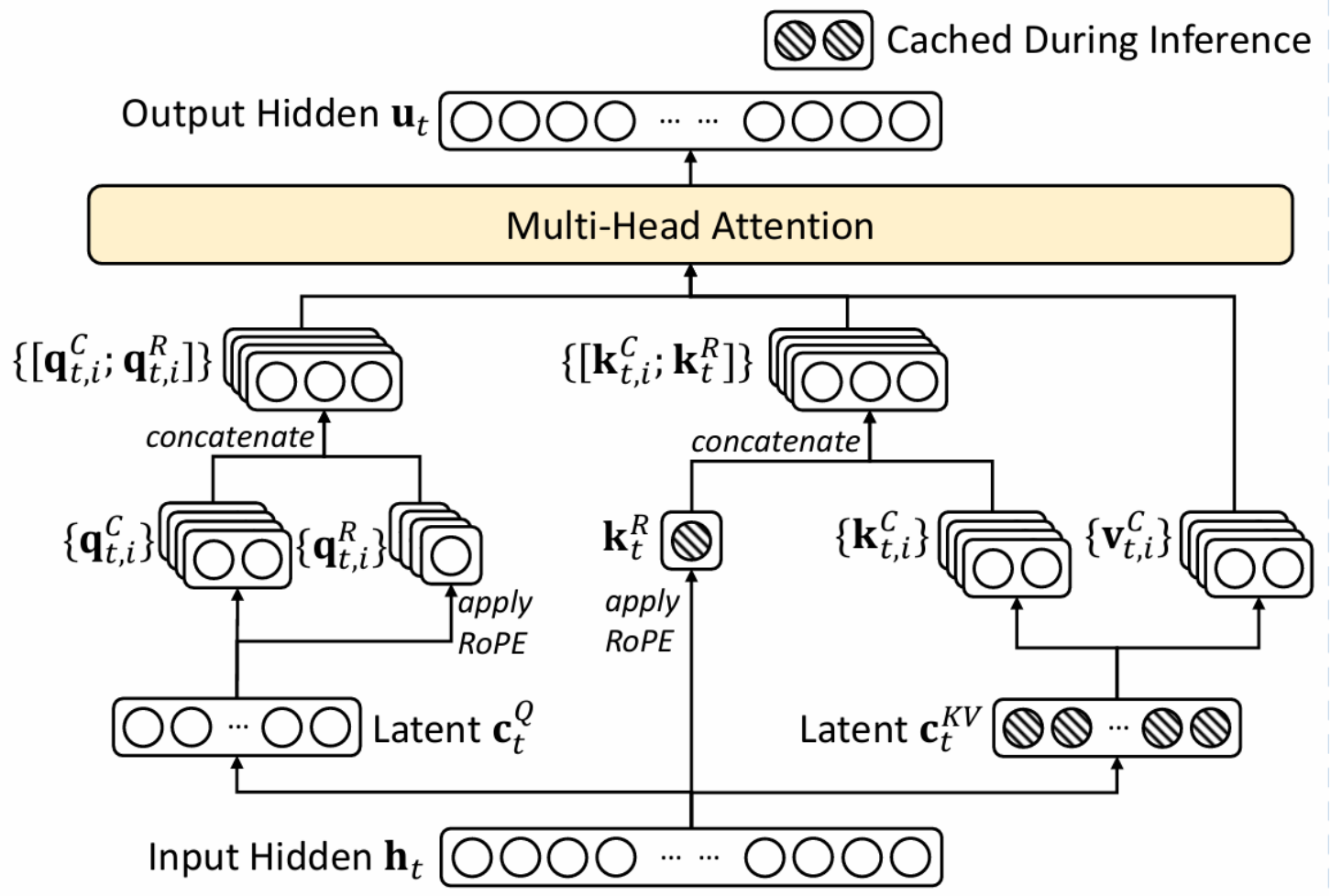
$$\mathbf{c}_t^Q = W^{DQ}\mathbf{h}_t, \tag{12}$$

$$\mathbf{q}_t^C = W^{UQ}\mathbf{c}_t^Q, \tag{13}$$

where $\mathbf{c}_t^Q \in \mathbb{R}^{d_c'}$ is the compressed latent vector for queries; $d_c' (\ll d_h n_h)$ denotes the query compression dimension; and $W^{DQ} \in \mathbb{R}^{d_c' \times d}, W^{UQ} \in \mathbb{R}^{d_h n_h \times d_c'}$ are the down-projection and up-projection matrices for queries, respectively.

dimension to 5120. All learnable parameters are randomly initialized with a standard deviation of 0.006. In MLA, we set the number of attention heads n_h to 128 and the per-head dimension d_h to 128. The KV compression dimension d_c is set to 512, and the query compression dimension d_c' is set to 1536. For the decoupled queries and key, we set the per-head dimension d_h^R to 64. Following Dai et al. (2024), we substitute all FFNs except for the first layer with MoE layers.

回到原论文


Cached During Inference
Output Hidden $\mathbf{u}_t$
Multi-Head Attention
$\{[\mathbf{q}_{t,i}^C; \mathbf{q}_{t,i}^R]\}$
$\{[\mathbf{k}_{t,i}^C; \mathbf{k}_t^R]\}$
concatenate
concatenate
$\{\mathbf{q}_{t,i}^C\}$
$\{\mathbf{q}_{t,i}^R\}$
apply RoPE
$\mathbf{k}_t^R$
apply RoPE
$\{\mathbf{k}_{t,i}^C\}$
$\{\mathbf{v}_{t,i}^C\}$
Latent $\mathbf{c}_t^Q$
Latent $\mathbf{c}_t^{KV}$
Input Hidden $\mathbf{h}_t$