Flash Attention 2

A Preprint

In this paper authors work on parallelism and work partitioning to address the problem or reaching only 30-40% of theoretical maximum

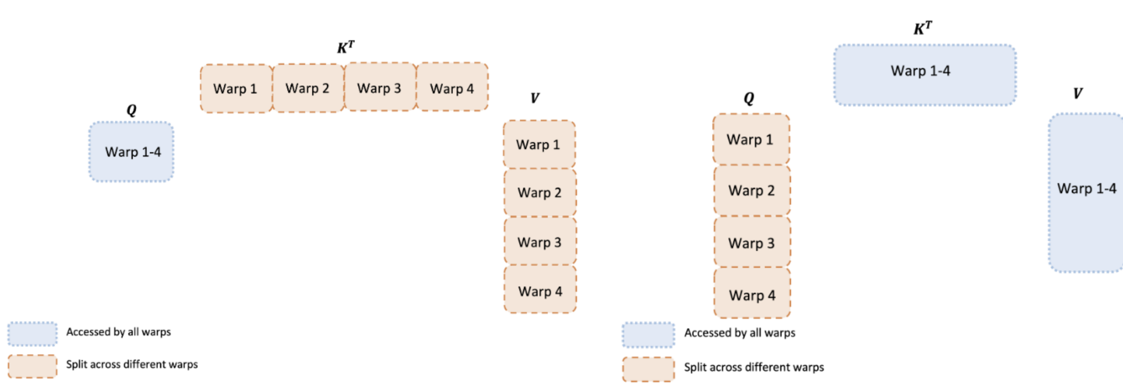
$Q, K, V \in \mathbb{R}^{N \times d}$ where N is length of input sequence and d is head dimension; $O \in \mathbb{R}^{\mathbb{N}\times}$ output. $S = QK^T \quad P = softmax(S) \quad O = PV$

Then backprop:

$$dV = P^T dO \tag{1}$$
$$dP = dOV^T \tag{2}$$
$$dS = dsoftmax(dP) \tag{3}$$
$$dQ = dSK \tag{4}$$
$$dK = QdS^T \tag{5}$$

Flash Attention 2 presents new ideas above first paper. More specifically authors elaborate on removing scaling term and saving statistics. As for removing scaling term $diag(l^{(2)})^{-1}$ is mentioned. In every pass while calculating output it was done next way: $O^{(2)} = diag(l^{(1)}/l^{(2)})^{-1}O^{(1)} + diag(l^{(2)})^{-1}e^{S^{(2)}-m^{(2)}}V^{(2)}$. The new way is presented: maintaining unscaled version of calculations and scaling only in the end of the loop: $O^{(2)} = diag(l^{(1)})^{-1}O^{(1)} + e^{S^{(2)}-m^{(2)}}V^{(2)}$. Also authors mention that it is not neccessary to store max m and sum of exponentials l as logsumexp can be stored: $L^{(j)} = m^{(j)} + \log(l^{(j)})$

Authors also work on parallelization suggesting splitting rows into blocks which results in GPU load. Also instead of splitting K and V authors split Q across 4 warps while keeping K and V accessible by all warps. After each warp performs matrix multiply to get a slice of QK^T, they just need to multiply with their shared slice of V to get their corresponding slice of the output.



(a) FlashAttention (b) FlashAttention-2

Algorithm 1 FlashAttention-2 forward pass

Require: Matrices $Q, K, V \in \mathbb{R}^{N \times d}$ in HBM, block sizes B_c, B_r.

Divide Q into $T_r = \lceil N/B_r \rceil$ blocks $Q_1, \ldots, Q_{T_r}$ of size $B_r \times d$, and divide K, V into $T_c = \lceil N/B_c \rceil$ blocks $K_1, \ldots, K_{T_c}$, $V_1, \ldots, V_{T_c}$ of size $B_c \times d$.
Divide output $O \in \mathbb{R}^{N \times d}$ into T_r blocks $O_1, \ldots, O_{T_r}$ of size $B_r \times d$, and logsumexp L into T_r blocks $L_1, \ldots, L_{T_r}$ of size B_r.
for $i = 1, \ldots, T_r$ **do**
    Load Q_i from HBM to on-chip SRAM.
    Initialize

$$O_i^{(0)} = 0^{B_r \times d}, \quad \ell_i^{(0)} = 0^{B_r}, \quad m_i^{(0)} = (-\infty)^{B_r}.$$

    for $j = 1, \ldots, T_c$ **do**
        Load K_j, V_j from HBM to on-chip SRAM.
        $S_i^{(j)} = Q_i \, K_j^\top \in \mathbb{R}^{B_r \times B_c}$.
        $m_i^{(j)} = \max\big(m_i^{(j-1)}, \mathrm{rowmax}(S_i^{(j)})\big)$.
        $\tilde{P}_i^{(j)} = \exp\big(S_i^{(j)} - m_i^{(j)}\big) \in \mathbb{R}^{B_r \times B_c}$.
        $\ell_i^{(j)} = e^{m_i^{(j-1)} - m_i^{(j)}} \, \ell_i^{(j-1)} + \mathrm{rowsum}\big(\tilde{P}_i^{(j)}\big)$.
        $O_i^{(j)} = e^{m_i^{(j-1)} - m_i^{(j)}} \, O_i^{(j-1)} + \tilde{P}_i^{(j)} \, V_j$.
    end for
    $O_i = \mathrm{diag}\big(\ell_i^{(T_c)}\big)^{-1} \, O_i^{(T_c)}$.
    $L_i = m_i^{(T_c)} + \log\big(\ell_i^{(T_c)}\big)$.
    Write O_i and L_i back to HBM.
end for
return O, L