Lab6 FlashAttention

Nov, 2025 Parallel Programming

Outline

- Attention
- FlashAttention
- Lab6 Assignment
- HW4 Assignment

Attention

- Q : What we're focusing on.
- K : What features are available.
- V : What content is retrieved based on focus.

$$\mathrm{Attention}(Q, K, V) = \mathrm{softmax}(\frac{QK^T}{\sqrt{d_k}})V \qquad \mathbf{Q}, \mathbf{K}, \mathbf{V} \in \mathbb{R}^{N \times d}$$

Attention

$$\boxed{\mathbf{S} = \mathbf{Q}\mathbf{K}^\top \in \mathbb{R}^{N \times N}}, \quad \mathbf{P} = \mathrm{softmax}(\mathbf{S}) \in \mathbb{R}^{N \times N}, \quad \mathbf{O} = \mathbf{P}\mathbf{V} \in \mathbb{R}^{N \times d},$$

$$\begin{bmatrix} \boxed{s_{11}} & s_{12} & s_{13} & s_{14} \\ s_{21} & s_{22} & s_{23} & s_{24} \\ s_{31} & s_{32} & s_{33} & s_{34} \\ s_{41} & s_{42} & s_{43} & s_{44} \end{bmatrix} = \begin{bmatrix} q_{11} & q_{12} & q_{13} \\ q_{21} & q_{22} & q_{23} \\ q_{31} & q_{32} & q_{33} \\ q_{41} & q_{42} & q_{43} \end{bmatrix} \cdot \begin{bmatrix} k_{11} & k_{21} & k_{31} & k_{41} \\ k_{12} & k_{22} & k_{32} & k_{42} \\ k_{13} & k_{23} & k_{33} & k_{43} \end{bmatrix}$$

Attention

$$\mathbf{S} = \mathbf{Q}\mathbf{K}^\top \in \mathbb{R}^{N\times N}, \quad \boxed{\mathbf{P} = \mathrm{softmax}(\mathbf{S}) \in \mathbb{R}^{N\times N}}, \quad \mathbf{O} = \mathbf{P}\mathbf{V} \in \mathbb{R}^{N\times d},$$

$$\begin{bmatrix} \boxed{p_{11} \quad p_{12} \quad p_{13} \quad p_{14}} \\ p_{21} \quad p_{22} \quad p_{23} \quad p_{24} \\ p_{31} \quad p_{32} \quad p_{33} \quad p_{34} \\ p_{41} \quad p_{42} \quad p_{43} \quad p_{44} \end{bmatrix} = \mathrm{softmax}\left(\begin{bmatrix} \boxed{s_{11} \quad s_{12} \quad s_{13} \quad s_{14}} \\ s_{21} \quad s_{22} \quad s_{23} \quad s_{24} \\ s_{31} \quad s_{32} \quad s_{33} \quad s_{34} \\ s_{41} \quad s_{42} \quad s_{43} \quad s_{44} \end{bmatrix}\right)$$

Attention

$$\mathbf{S} = \mathbf{Q}\mathbf{K}^\top \in \mathbb{R}^{N\times N}, \quad \mathbf{P} = \mathrm{softmax}(\mathbf{S}) \in \mathbb{R}^{N\times N}, \quad \boxed{\mathbf{O} = \mathbf{P}\mathbf{V} \in \mathbb{R}^{N\times d}},$$

$$\begin{bmatrix} \boxed{o_{11}} & o_{12} & o_{13} \\ o_{21} & o_{22} & o_{23} \\ o_{31} & o_{32} & o_{33} \\ o_{41} & o_{42} & o_{43} \end{bmatrix} = \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \\ p_{41} & p_{42} & p_{43} & p_{44} \end{bmatrix} \cdot \begin{bmatrix} v_{11} & v_{12} & v_{13} \\ v_{21} & v_{22} & v_{23} \\ v_{31} & v_{32} & v_{33} \\ v_{41} & v_{42} & v_{43} \end{bmatrix}$$

Multi-Head Attention

- Rich Representations
- Efficient Parallelization
- E.g. emb_dim = 4096 -> num_heads = 32, head_size = 128

$$\mathrm{MultiHead}(Q, K, V) = \mathrm{Concat}(\mathrm{head_1}, ..., \mathrm{head_h})W^O$$
$$\text{where } \mathrm{head_i} = \mathrm{Attention}(QW_i^Q, KW_i^K, VW_i^V)$$

Causal Attention

- If you're predicting the next word in a sentence, the model shouldn't have access to future words beyond the current position.

$$\begin{bmatrix} s_{11} & s_{12} & s_{13} & s_{14} \\ s_{21} & s_{22} & s_{23} & s_{24} \\ s_{31} & s_{32} & s_{33} & s_{34} \\ s_{41} & s_{42} & s_{43} & s_{44} \end{bmatrix}$$

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

$$\text{Masked Scores}_{i,j} = \begin{cases} \frac{(QK^T)_{i,j}}{\sqrt{d_k}}, & \text{if } j \leq i \\ -\infty, & \text{if } j > i \end{cases}$$

$$\text{Masked Attention Weights}_{i,j} = \text{softmax}(\text{Masked Scores}_{i,j})$$

$$\text{Causal Attention}(Q, K, V) = \text{softmax}\left(\text{Mask}\left(\frac{QK^T}{\sqrt{d_k}}\right)\right)V$$

FlashAttention - Overview

- Goal: avoid reading and writing the attention matrix to and from HBM.
  - Computing the softmax without access to the whole input.
  - Not storing the large intermediate attention matrix for the backward pass.

Algorithm 0 Standard Attention Implementation

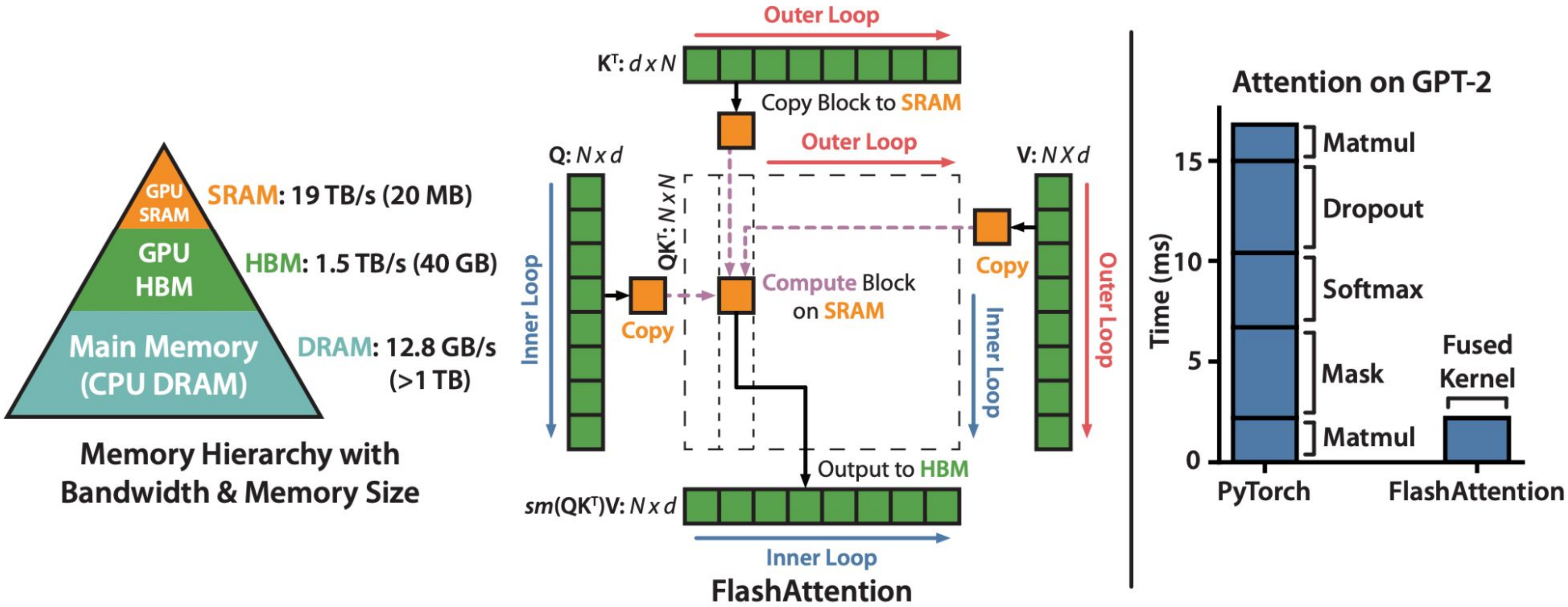
Require: Matrices $\mathbf{Q}, \mathbf{K}, \mathbf{V} \in \mathbb{R}^{N \times d}$ in HBM.

Load $\mathbf{Q}, \mathbf{K}$ by blocks from HBM, compute $\mathbf{S} = \mathbf{Q}\mathbf{K}^\top$, write $\mathbf{S}$ to HBM.
Read $\mathbf{S}$ from HBM, compute $\mathbf{P} = \mathrm{softmax}(\mathbf{S})$, write $\mathbf{P}$ to HBM.
Load $\mathbf{P}$ and $\mathbf{V}$ by blocks from HBM, compute $\mathbf{O} = \mathbf{P}\mathbf{V}$, write $\mathbf{O}$ to HBM.
Return $\mathbf{O}$.

FlashAttention - Method

- Tiling: split the input into blocks and make several passes over input blocks.
  - Matrix multiplication and pointwise operations are easy to handle.
  - SoftMax: need to maintain $m(x)$, $l(x)$.
- Recompute: store the softmax normalization factor in order to quickly recompute in the backward pass.

FlashAttention



FlashAttention - SoftMax

$$\operatorname{softmax}(x_i) = \frac{e^{x_i}}{\sum_{j=1}^{d} e^{x_j}}$$

$$m = \max_i (x_i); \quad \operatorname{softmax}(x_i) = \frac{e^{x_i - m}}{\sum_{j=1}^{d} e^{x_j - m}}$$

FlashAttention - SoftMax

$$m(x) := \max_i \quad x_i, \quad f(x) := \begin{bmatrix} e^{x_1 - m(x)} & \dots & e^{x_B - m(x)} \end{bmatrix}, \quad \ell(x) := \sum_i f(x)_i, \quad \text{softmax}(x) := \frac{f(x)}{\ell(x)}.$$

For vectors $x^{(1)}, x^{(2)} \in \mathbb{R}^B$, we can decompose the softmax of the concatenated $x = \begin{bmatrix} x^{(1)} & x^{(2)} \end{bmatrix} \in \mathbb{R}^{2B}$ as:

$$m(x) = m(\begin{bmatrix} x^{(1)} & x^{(2)} \end{bmatrix}) = \max(m(x^{(1)}), m(x^{(2)})), \quad f(x) = \begin{bmatrix} e^{m(x^{(1)}) - m(x)} f(x^{(1)}) & e^{m(x^{(2)}) - m(x)} f(x^{(2)}) \end{bmatrix},$$

$$\ell(x) = \ell(\begin{bmatrix} x^{(1)} & x^{(2)} \end{bmatrix}) = e^{m(x^{(1)}) - m(x)} \ell(x^{(1)}) + e^{m(x^{(2)}) - m(x)} \ell(x^{(2)}), \quad \text{softmax}(x) = \frac{f(x)}{\ell(x)}.$$

FlashAttention - SoftMax

$$m_1 = max([1,2]) = 2 \qquad m_2 = max([3,4]) = 4 \qquad m = max(m_1, m_2) = 4$$

$$f_1 = [e^{1-2}, e^{2-2}] = [e^{-1}, e^0] \qquad f_2 = [e^{3-4}, e^{4-4}] = [e^{-1}, e^0] \qquad f = [e^{m_1 - m} f_1, e^{m_2 - m} f_2] = [e^{-3}, e^{-2}, e^{-1}, e^0]$$

$$l_1 = \sum f_1 = e^{-1} + e^0 \qquad l_2 = \sum f_2 = e^{-1} + e^0 \qquad l = e^{m_1 - m} l_1 + e^{m_2 - m} l_2 = e^{-3} + e^{-2} + e^{-1} + e^0$$

$$o_1 = \frac{f_1}{l_1} = \frac{[e^{-1}, e^0]}{e^{-1} + e^0} \qquad o_2 = \frac{f_2}{l_2} = \frac{[e^{-1}, e^0]}{e^{-1} + e^0} \qquad o = \frac{f}{l} = \frac{[e^{-3}, e^{-2}, e^{-1}, e^0]}{e^{-3} + e^{-2} + e^{-1} + e^0}$$

FlashAttention - Algorithm

Algorithm 1 FlashAttention

Require: Matrices $\mathbf{Q}, \mathbf{K}, \mathbf{V} \in \mathbb{R}^{N \times d}$ in HBM, on-chip SRAM of size M.

Set block sizes $B_c = \left\lceil \frac{M}{4d} \right\rceil, B_r = \min\left(\left\lceil \frac{M}{4d} \right\rceil, d\right)$.

Initialize $\mathbf{O} = (0)_{N \times d} \in \mathbb{R}^{N \times d}, \ell = (0)_N \in \mathbb{R}^N, m = (-\infty)_N \in \mathbb{R}^N$ in HBM.

Divide $\mathbf{Q}$ into $T_r = \left\lceil \frac{N}{B_r} \right\rceil$ blocks $\mathbf{Q}_1, \dots, \mathbf{Q}_{T_r}$ of size $B_r \times d$ each, and divide $\mathbf{K}, \mathbf{V}$ in to $T_c = \left\lceil \frac{N}{B_c} \right\rceil$ blocks $\mathbf{K}_1, \dots, \mathbf{K}_{T_c}$ and $\mathbf{V}_1, \dots, \mathbf{V}_{T_c}$, of size $B_c \times d$ each.

Divide $\mathbf{O}$ into T_r blocks $\mathbf{O}_i, \dots, \mathbf{O}_{T_r}$ of size $B_r \times d$ each, divide ℓ into T_r blocks $\ell_i, \dots, \ell_{T_r}$ of size B_r each, divide m into T_r blocks $m_1, \dots, m_{T_r}$ of size B_r each.

for $1 \le j \le T_c$ **do**

  Load $\mathbf{K}_j, \mathbf{V}_j$ from HBM to on-chip SRAM.

  for $1 \le i \le T_r$ **do**

    Load $\mathbf{Q}_i, \mathbf{O}_i, \ell_i, m_i$ from HBM to on-chip SRAM.

    On chip, compute $\mathbf{S}_{ij} = \mathbf{Q}_i \mathbf{K}_j^T \in \mathbb{R}^{B_r \times B_c}$.

    On chip, compute $\tilde{m}_{ij} = \mathrm{rowmax}(\mathbf{S}_{ij}) \in \mathbb{R}^{B_r}$, $\tilde{\mathbf{P}}_{ij} = \exp(\mathbf{S}_{ij} - \tilde{m}_{ij}) \in \mathbb{R}^{B_r \times B_c}$ (pointwise), $\tilde{\ell}_{ij} = \mathrm{rowsum}(\tilde{\mathbf{P}}_{ij}) \in \mathbb{R}^{B_r}$.

    On chip, compute $m_i^{\mathrm{new}} = \max(m_i, \tilde{m}_{ij}) \in \mathbb{R}^{B_r}$, $\ell_i^{\mathrm{new}} = e^{m_i - m_i^{\mathrm{new}}} \ell_i + e^{\tilde{m}_{ij} - m_i^{\mathrm{new}}} \tilde{\ell}_{ij} \in \mathbb{R}^{B_r}$.

    Write $\mathbf{O}_i \leftarrow \mathrm{diag}(\ell_i^{\mathrm{new}})^{-1}(\mathrm{diag}(\ell_i) e^{m_i - m_i^{\mathrm{new}}} \mathbf{O}_i + e^{\tilde{m}_{ij} - m_i^{\mathrm{new}}} \tilde{\mathbf{P}}_{ij} \mathbf{V}_j)$ to HBM.

    Write $\ell_i \leftarrow \ell_i^{\mathrm{new}}$, $m_i \leftarrow m_i^{\mathrm{new}}$ to HBM.

  end for

end for

Return $\mathbf{O}$.

Reference

- https://arxiv.org/pdf/2205.14135
- https://www.cvmart.net/community/detail/7943
- https://www.youtube.com/watch?v=eMlx5fFNoYc

Lab6 - Objective

- Evaluate the performance of attention mechanisms by comparing:
  - The original PyTorch implementation
  - The FlashAttention v1 implementation
- Analyze the benefits of FlashAttention and explore its advantages over the standard approach.
- Conduct benchmarking with varying parameters and compare the results to gain deeper insights.

Lab6 - Benchmark Script

- TA provide the benchmark script that does not require any modifications.
- Your task is to adjust only the parameters within the benchmark.
- The result will be outputted to a JSON file.
    - Execution time
    - FLOPs
    - Peak memory usage

benchmark_result.json

lab5 > benchmark_result.json > ...


```json
{
  "forward": {
    "time(s)": 0.007095515976349513,
    "FLOPS(TFLOPs/s)": 38.739664297876566
  },
  "backward": {
    "time(s)": 0.018877355754375456,
    "FLOPS(TFLOPs/s)": 36.40312638599925
  },
  "forward_backward": {
    "time(s)": 0.025972871730724968,
    "FLOPS(TFLOPs/s)": 37.04144402199095
  },
  "peak_memory_usage(MB)": 1288.00048828125
}
```

Lab6 - Benchmark Script

- Test following parameters and compare the results.
  - **batch_size** : int
  - **seq_len (N)** : int
  - **num_heads** : int, (must be divisible by emb_dim)
  - **emb_dim (d)** : int
  - **impl** : str, (choose between Pytorch and Flash1)
  - **causal** : bool

Lab6 - Benchmark Script

- [Colab download link]

Connect to T4 GPU

Lab6 - Submission

- Plot the experimental data in a chart for better visualization.
- Analyze and explain your observations based on the collected data.
- Submit your report as a `lab6.pdf` file to eeclass before 11/27 23:59.

HW4 - FlashAttention

- Spec
- Deadline : 12/7 23:59