I/O Complexity of Attention, or How Optimal is FlashAttention?

Barna Saha & Christopher Ye

University of California San Diego

The Attention Mechanism

central to the popular Transformer architecture.

[Attention Is All You Need]
Vaswani, Shazeer, Parmar, Uszkoreit, Jones, Gomez, Kaiser, Polosukhin.

many applications:

- text-to-text translation
- voice-to-text & text-to-voice
- text-to-image (Dall-E)
- ChatGPT & more!

The Attention Mechanism

inputs $Q, K, V \in \mathbb{R}^{N \times d}$ compute $\text{softmax}(QK^T) \cdot V = O$.

Q – Query, K – Key, V – Value

N – context length
d – head dimension

$$\text{softmax}(QK^T)_{ij} = \frac{\exp(QK^T_{ij})}{\sum_j \exp(QK^T_{ij})}$$

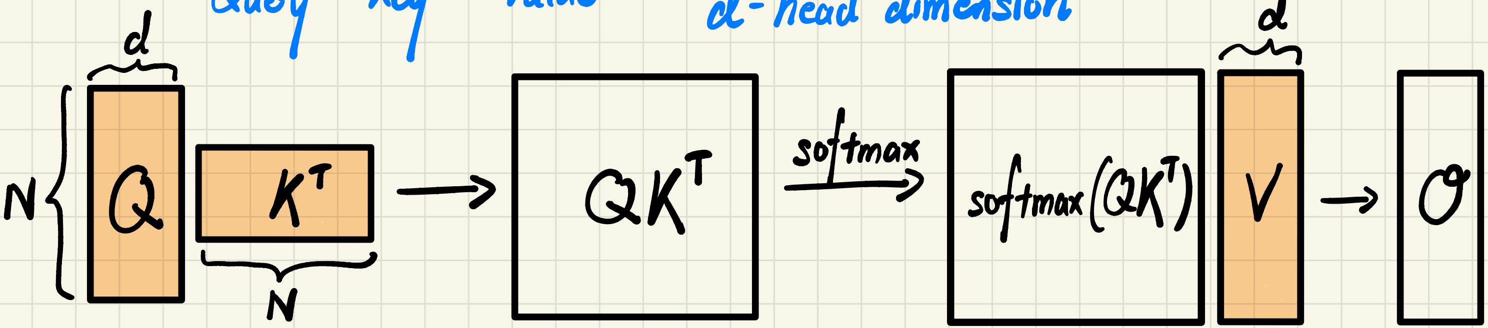
$A = \text{softmax}(QK^T)$ is the "Attention" matrix.

The Attention Mechanism

inputs $Q, K, V \in \mathbb{R}^{N \times d}$ compute $\text{softmax}(QK^T) \cdot V = O$

Query Key Value

N - context length
d - head dimension

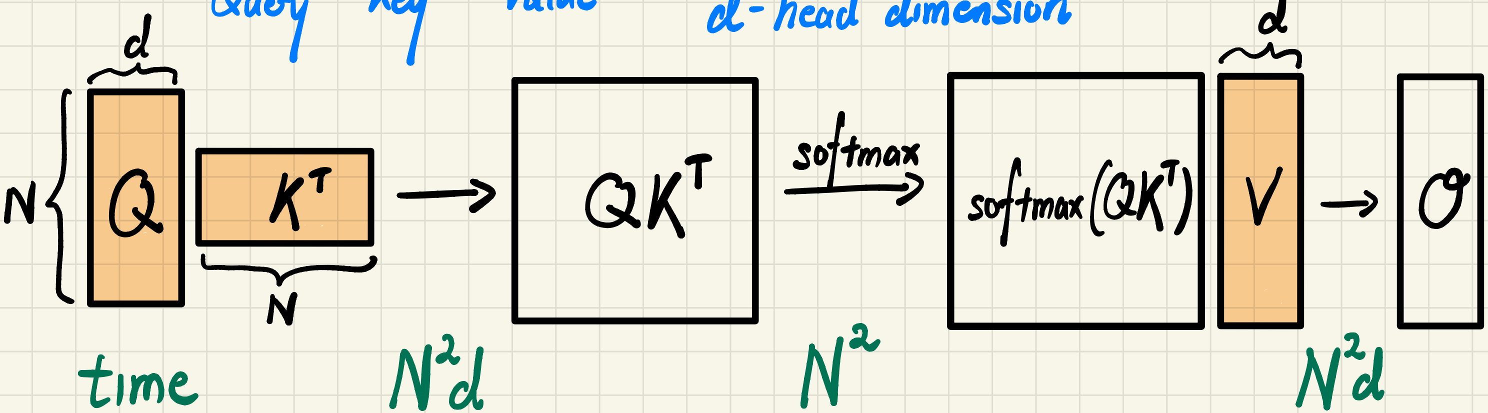


The Attention Mechanism

inputs $Q, K, V \in \mathbb{R}^{N \times d}$ compute $\text{softmax}(QK^T) \cdot V = O$.

Query, Key, Value

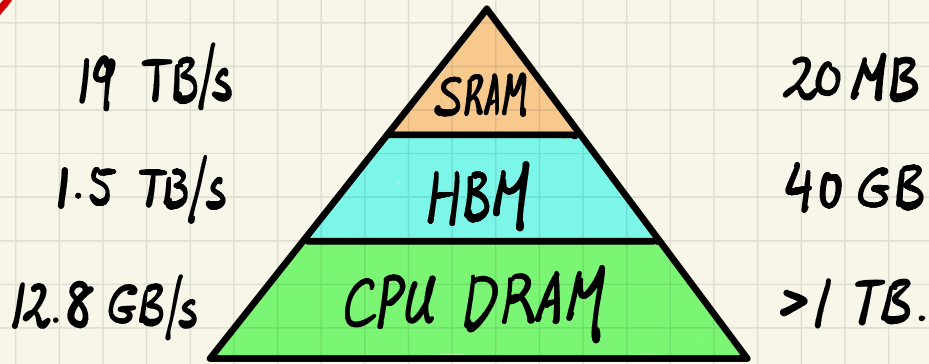
N - context length
d - head dimension



even w/ optimal matrix multiplication [$\omega = 2$], requires N^2-time.
typically context length $N \gg d \Rightarrow$ quadratic time!

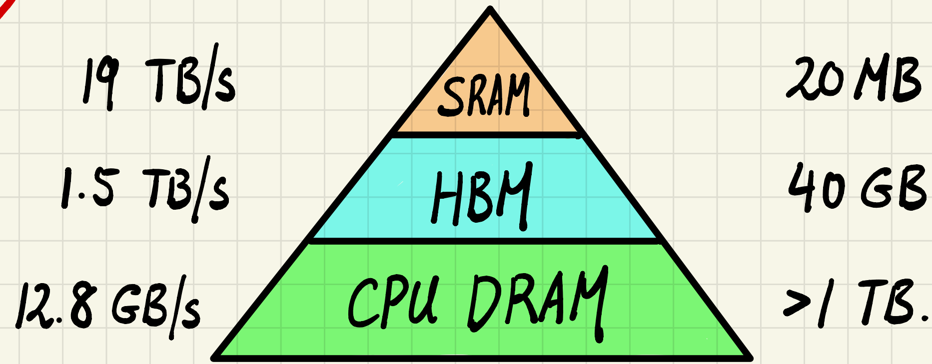
Where is the Bottleneck?

The Memory Hierarchy.



Where is the Bottleneck?

The Memory Hierarchy.



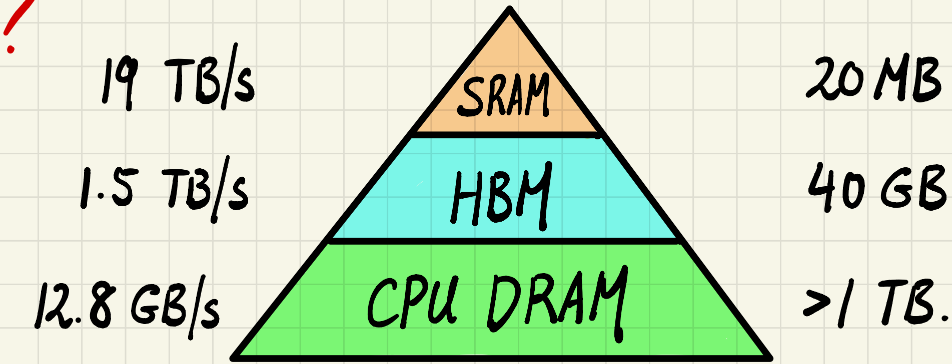
model w/ 2 levels:

cache - small (<M entries), fast, where computation occurs.

memory - large, slow, where data resides.

Where is the Bottleneck?

The Memory Hierarchy.



model w/ 2 levels:

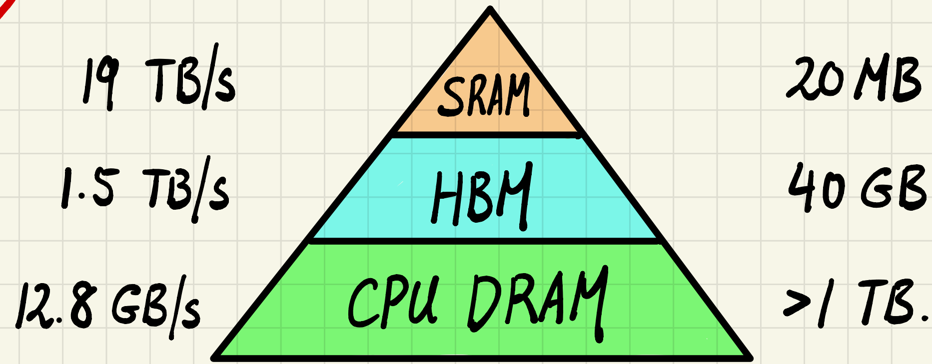
cache - small (<M entries), fast, where computation occurs.

memory - large, slow, where data resides.

I/O complexity: how many entries moved b/w cache & memory?

Where is the Bottleneck?

The Memory Hierarchy.



model w/ 2 levels:

cache - small (<M entries), fast, where computation occurs.

memory - large, slow, where data resides.

I/O complexity: how many entries moved b/w cache & memory?

FlashAttention: attention can be computed w/ <u>linear</u> I/O

Dao, Fu, Ermon, Rudra, Ré

The FlashAttention Algorithm

standard algorithm has $\geq N^2$ I/O complexity.

<u>Thm</u> FlashAttention has $O\left(\frac{N^2 d^2}{M}\right)$ I/O complexity.

The FlashAttention Algorithm

standard algorithm has $\geq N^2$ I/O complexity.

Thm FlashAttention has $O\left(\frac{N^2 d^2}{M}\right)$ I/O complexity.

Rmk when $M = \Theta(Nd)$, FlashAttention is optimal as O has size $\Omega(Nd)$.

Q are there better algorithms when $M < Nd$?

Is FlashAttention Optimal?

__Thm 1__ Attention (w/ standard matrix multiplication) requires $\Theta\left(\min\left(\frac{N^2d^2}{M}, \frac{N^2d}{\sqrt{M}}\right)\right)$- I/O.

large cache regime $M \geq d^2$: $\frac{N^2d^2}{M} \leq \frac{N^2d}{\sqrt{M}}$

FlashAttention is __optimal__

small cache regime $M < d^2$: $\frac{N^2d}{\sqrt{M}} \leq \frac{N^2d^2}{M}$. [note $\frac{N^2d}{\sqrt{M}} \geq N^2$].

FlashAttention is __not__ optimal, but quadratic I/O is necessary.

we give a better algorithm w/ lower I/O complexity.

__Thm 2__ There is an algorithm computing Attention w/ $O\left(\frac{N^2d}{\sqrt{M}}\right)$- I/O.

Is FlashAttention Optimal?

Reminder

using standard MM, Attention has I/O Complexity $\Theta\left(\min\left(\frac{N^2 d^2}{M}, \frac{N^2 d}{\sqrt{M}}\right)\right)$.

__Thm 3__ Attention (w/ any matrix multiplication algorithm) requires $\Omega\left(\min\left(\frac{N^2 d^2}{M}, N^2\right)\right)$ - I/O.

large cache regime $M \geq d^2$: $\frac{N^2 d^2}{M} \leq N^2$

FlashAttention is __optimal__ even w/ FMM.

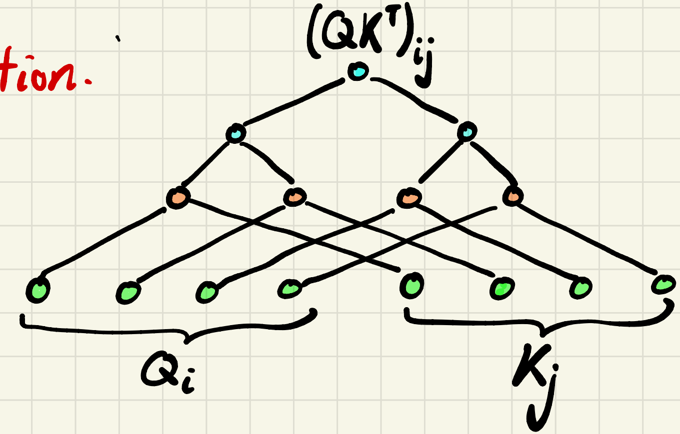
small cache regime $M < d^2$: $N^2 \leq \frac{N^2 d^2}{M}$ => quadratic I/O optimal.

we obtain algorithms w/ I/O $\ll N^2 d / \sqrt{M}$.

__Thm 4__ when $M < d^2$, Attention & matrix multiplication have same I/O complexity.

Attention w/ Standard Matrix Multiplication.

consider computational graph G of Attention.



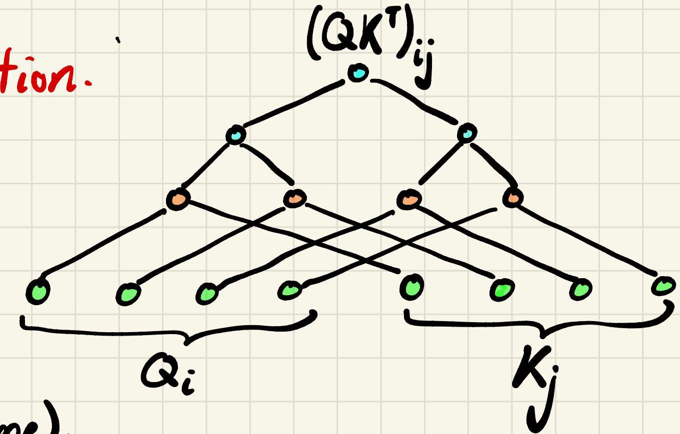
Attention w/ Standard Matrix Multiplication.

consider computational graph G of Attention.

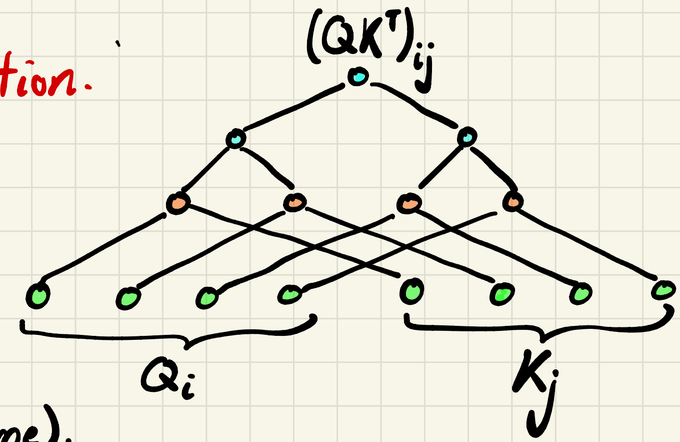
Red-Blue Pebble Game [Hong, Kung]

intuition: blue pebbles on memory (unlimited)

red pebbles on cache ($\leq M$ at a time).



Attention w/ Standard Matrix Multiplication.



consider computational graph G of Attention.

Red-Blue Pebble Game [Hong, Kung]

intuition: blue pebbles on memory (unlimited)

red pebbles on cache ($\leq M$ at a time).

initial : blue pebbles on inputs ⟶ final: blue pebbles on outputs.

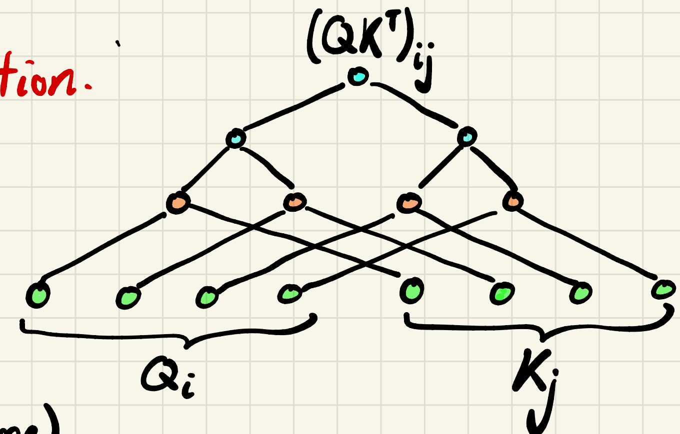
(R1) add blue pebble to red (write)

(R2) add red pebble to blue (read)

(R3) if all parents red, add red to child (compute)

Defn I/O Complexity is # of (R1) & (R2) to compute initial → final.

Attention w/ Standard Matrix Multiplication.



consider computational graph G of Attention.

Red-Blue Pebble Game [Hong, Kung]

intuition: blue pebbles on memory (unlimited)

red pebbles on cache ($\leq M$ at a time).

initial: blue pebbles on inputs $\longrightarrow$ final: blue pebbles on outputs.

(R1) add blue pebble to red (write)

(R2) add red pebble to blue (read)

(R3) if all parents red, add red to child (compute)

Defn I/O Complexity is # of (R1) & (R2) to compute initial $\rightarrow$ final.

Q. How to lower bound I/O Complexity of G?

Attention w/ Standard Matrix Multiplication.

Defn an M-partition is $V_1, \ldots, V_T$ s.t.

1) $V_1, \ldots, V_T$ partition V
2) Each V_i has dominator set D_i of size $\leq M$ — input $\rightsquigarrow V_i$ paths dominated
3) Each V_i has minimum set M_i of size $\leq M$ — M_i has no children in V_i

Attention w/ Standard Matrix Multiplication.

<u>Defn</u> an M-partition is $V_1, \ldots, V_T$ s.t.

1) $V_1, \ldots, V_T$ partition V
2) Each V_i has dominator set D_i of size $\leq M$ — input $\rightsquigarrow V_i$ paths dominated
3) Each V_i has minimum set M_i of size $\leq M$ — M_i has no children in V_i

<u>intuition</u> each V_i represents a set of nodes computed w/o using I/O.

$D_i \sim$ set of nodes in cache at start

$M_i \sim$ set of nodes in cache at end

Attention w/ Standard Matrix Multiplication.

<u>Defn</u> an M-partition is $V_1, \dots, V_T$ s.t.

1) $V_1, \dots, V_T$ partition V
2) Each V_i has dominator set D_i of size $\leq M$ — input $\rightsquigarrow V_i$ paths dominated
3) Each V_i has minimum set M_i of size $\leq M$ — M_i has no children in V_i

<u>intuition</u> each V_i represents a set of nodes computed w/o using I/O.

$D_i \sim$ set of nodes in cache at start

$M_i \sim$ set of nodes in cache at end

<u>Thm</u> (Hong, Kung) I/O Complexity of $G = \Omega(T \cdot M)$.

we show $|V_i| = O\left(\frac{M^2}{d}\right) \Rightarrow T = \Omega\left(\frac{N^2 d^2}{M^2}\right)$.

Attention w/ Fast Matrix Multiplication

<u>idea</u> bound how many entries of QK^T computed w/ M I/O.

<u>Defn</u> [Matrix Compression] Alice is given $Q, K \in \mathbb{F}_q^{N \times d}$ & $B \geq 0$.

How many bits to send Bob so Bob can compute B entries of QK^T?

↑ one-way communication complexity (streaming, data structures & more!)

Attention w/ Fast Matrix Multiplication

<u>idea</u> bound how many entries of QK^T computed w/ M I/O.

<u>Defn</u> [Matrix Compression] Alice is given $Q, K \in \mathbb{F}_q^{N \times d}$ & $B \geq 0$.

How many bits to send Bob so Bob can compute B entries of QK^T?

one-way communication complexity (streaming, data structures & more!)

<u>Thm</u> One-way-communication of matrix compression $\geq \min(B, d\sqrt{B}) \log q$

send B entries

send $\sqrt{B}$ rows of Q, K

Attention w/ Fast Matrix Multiplication

<u>idea</u> bound how many entries of QK^T computed w/ M I/O.

<u>Defn</u> [Matrix Compression] Alice is given $Q, K \in \mathbb{F}_q^{N \times d}$ & $B \geq 0$.

How many bits to send Bob so Bob can compute B entries of QK^T?

one-way communication complexity (streaming, data structures & more!)

<u>Thm</u> One-way-communication of matrix compression $\geq \min(B, d\sqrt{B}) \log q$

send B entries

send $\sqrt{B}$ rows of Q, K

$\Rightarrow$ w/ cache of size $M \geq \min(B, d\sqrt{B})$

$\Rightarrow B \leq \max(M, \frac{M^2}{d^2})$.

$\Rightarrow$ I/O complexity $\geq M \cdot \frac{N^2}{B} \geq \min\left(N^2, \frac{N^2 d^2}{M}\right)$.

Future Directions

1) I/O complexity of Fast Matrix Multiplication?
2) I/O complexity (& other bottlenecks) of Multi-Head Attention?
3) Approximate Attention Mechanisms - better I/O complexity?

   HyperAttention Han, Jayaram, Karbasi, Mirrokni, Woodruff, Zandieh

   Fast Attention requires Bounded Entries Alman, Song

   & many more [our results rule out $\|\cdot\|_\infty$ - additive error]

4) Beyond the Worst-Case

   are there structures to exploit in Attention?

   can output be represented succinctly? is $\Theta(Nd)$ necessary?