7/oct/24
(10)

https://youtu.be/gMOAud7hZg4?list=PLSrTvUm384I9PV10koj_cqit9OfbJXEkq

Stanford, ML Sys

Video 67 Flash Attention (F.A)

Fast and Memory Efficient Exact Attention with IO Awareness

Why this?

3 main reasons for Modelling Longer Sequences

① NLP :- large content is required to understand books, plays, ... etc

② CV :- higher resolution → better insights

③ Time series, Audio, Video, Medical Images → data understanding

↓

these are sequences of Million steps

Challenge: Scaling Transformers to longer Sequences :-

* content length of GPT 3 → 2048
(Seq)

F.A → helps to train things faster and with longer content.

① Tiling

② Recomputation

] — 2 main things in F.A to reduce GPU memory IOs.

Speed → 3 times faster

Memory Efficient → 10 - 20 x for exact attention

Seq length (Content) → up to 16k

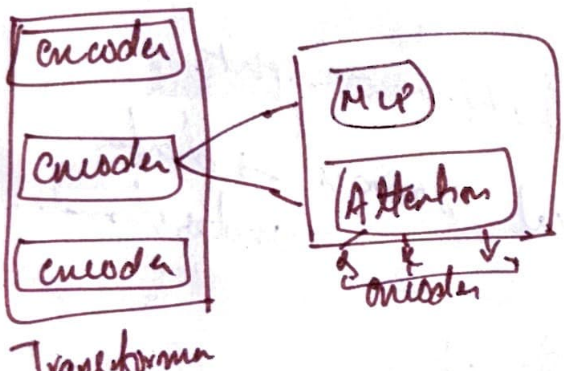
# Table of Content

① Background

② F.A Method

③ Empirical Validation

## ① Background



Q = Query

K = Key

V = Value

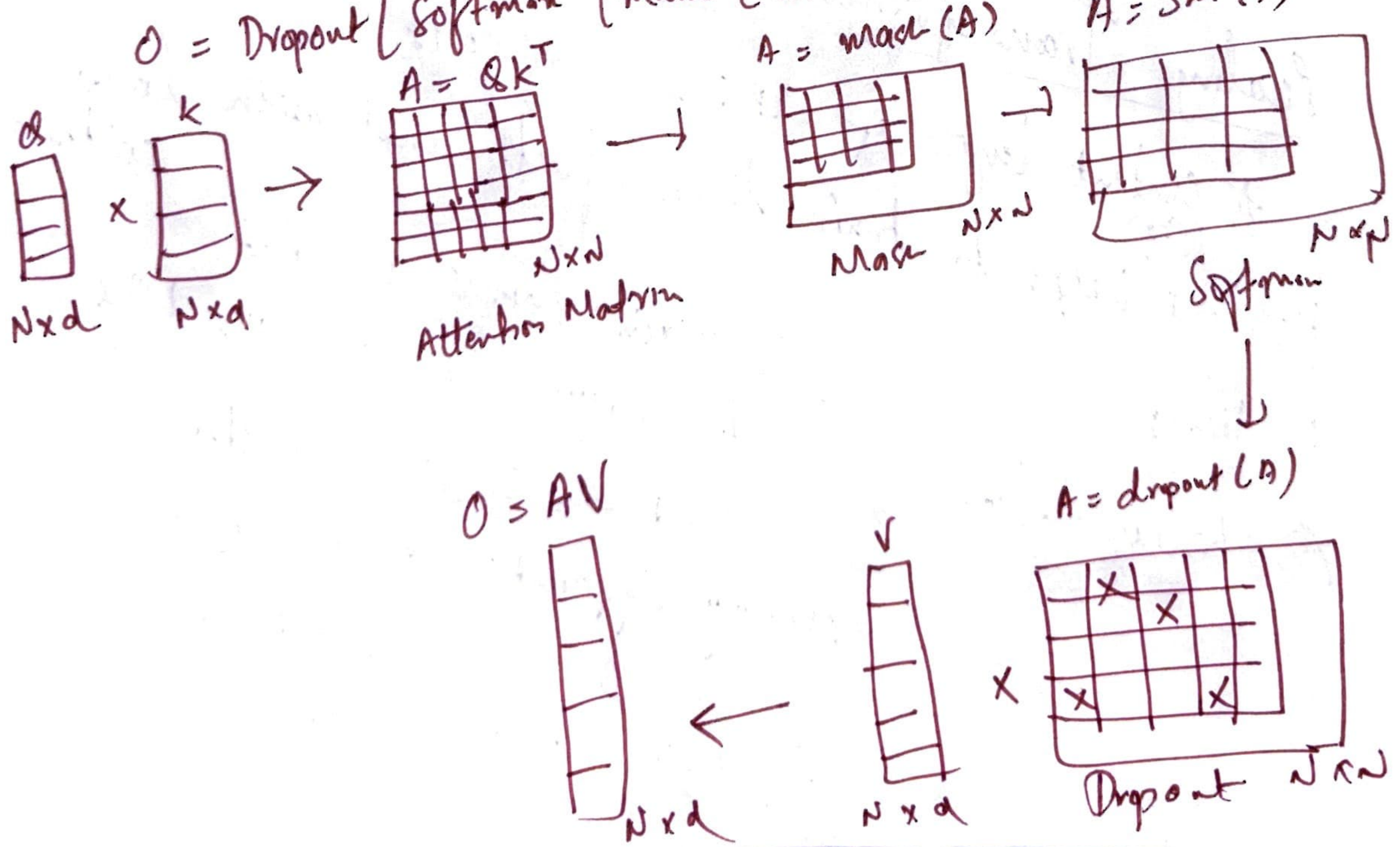
N = Seq length (2k, 8k, 16k)

d = head dimension (64, 128 ...)

→ Attention is heart of Transformer

→ Attention is bottlenecked by Memory Read/writes

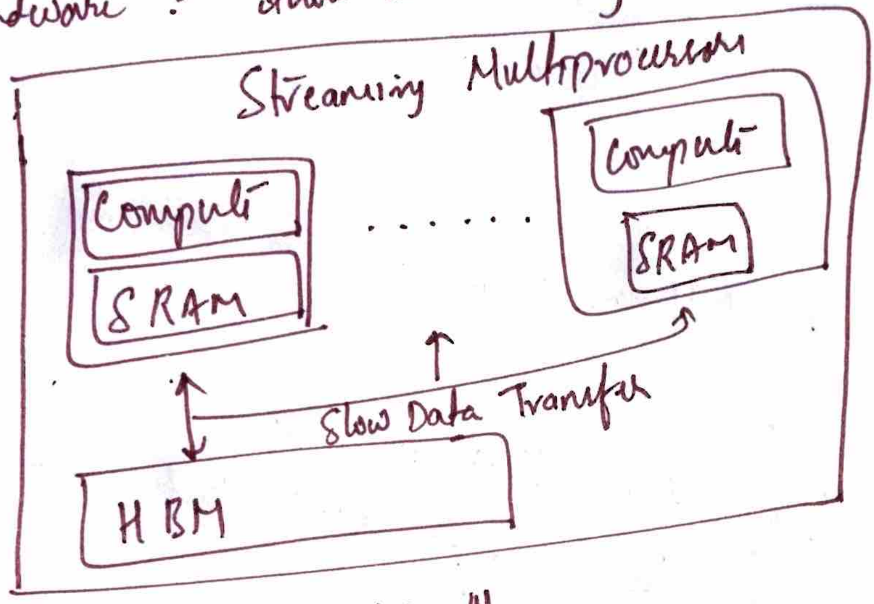
$$O = \text{Dropout}[\text{Softmax}[\text{mask}[QK^T]]]V$$



→ The problem is, lot of time is spent on Reading and writing in GPU instead of calculations.

→ Attention in GPT 2 : Masking, softmax and Dropout takes lot of time than matrix multiplication.

→ hardware :- there is memory hierarchy in hardware



→ SRAM is much much smaller but much much faster

GPU #1

while any computation :-

$$\text{inputs} \rightarrow \text{HBM} \rightarrow \text{SRAM} \Leftrightarrow \text{compute} \rightarrow \text{HBM}$$

↓

(Softmax, Matmul ...etc)

→ data moves back and forth between HBM and ~~SRAM~~ SRAM

② Method . F.A

Challenges :- How to reduce HBM Read/write : compute by blocks :-

① compute softmax reduction without access to full input

② backward without the large attention matrix from forward.

Techniques used to address these issues are:-

① Tiling : Restructure algorithm to load block by block from HBM to SRAM to compute attention

② Recomputation : Dont store attention matrix from forward, recompute it in the backward

Tiling :-

decomposing large softmax into smaller ones by scaling

Steps :-

① load inputs by block from HBM to SRAM

② On chip, compute attention outputs wrt that block

③ Update output in HBM by scaling

Recompute :-

by storing softmax normalization factors from fwd
(size N) quickly recompute attention in bwd from inputs
in SRAM