FlashAttention : Fast and Memory-Efficient Exact Attention with IO-Awareness

https://arxiv.org/pdf/2205.14135

0. Introduction

- Transformer attention는 높은 메모리 사용과 느린 속도가 대규모 모델 학습의 병목이 됨
- 기존 attention은 GPU 메모리 접근 비용을 충분히 고려하지 않음
- 본 논문은 **정확한 attention 계산을 유지하면서 메모리 사용을 크게 줄이는 방법** 제안
- 핵심 기여는 IO 병목을 최소화하는 새로운 attention 구현 방식 제시

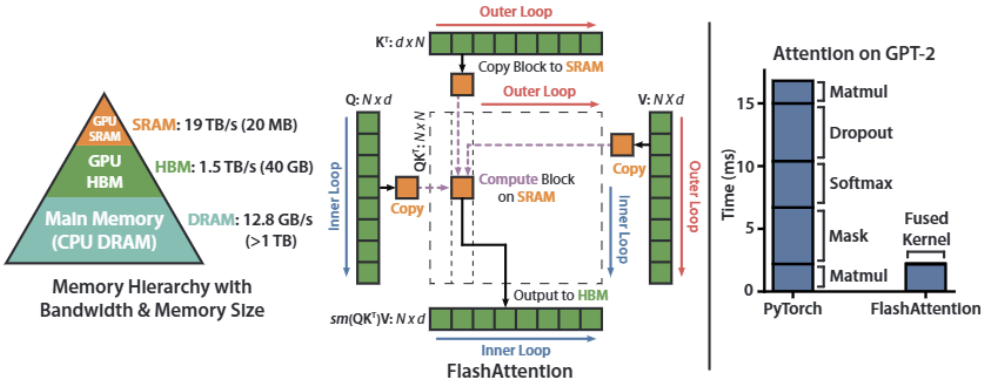
1. Overview

- FlashAttention은 attention 계산 순서를 재구성하여 GPU 메모리 접근을 최소화
- 중간 attention matrix를 저장하지 않고 블록 단위로 계산 수행
- 정확한 attention 결과를 유지하면서 속도와 메모리 사용 개선
- 대규모 Transformer 및 LLM 학습 효율 향상 목적

2. Challenges

- Attention 계산 시 $O(n^2)$ 메모리 요구 발생
- GPU 계산보다 메모리 접근이 병목이 되는 문제 존재
- 기존 방법은 근사 계산으로 정확도를 희생하는 경우 많음
- 긴 시퀀스 학습 시 메모리 한계로 batch 크기 축소 필요

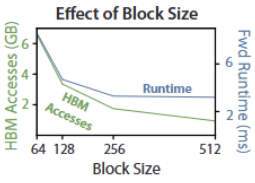
3. Method

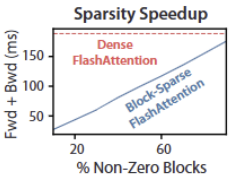


- Query, Key, Value를 블록 단위로 나누어 계산 수행
- Softmax 계산을 블록 단위 누적 방식으로 처리하여 정확도 유지
- 중간 attention matrix를 저장하지 않고 즉시 연산 후 폐기
- GPU SRAM과 HBM 간 데이터 이동 최소화 설계
- 결과적으로 메모리 사용량 O(n) 수준으로 감소

4. Experiments

Attention	Standard	FlashAttention
GFLOPs	66.6	75.2
HBM R/W (GB)	40.3	4.4
Runtime (ms)	41.7	7.3





BERT Implementation	Training time (minutes)
Nvidia MLPerf 1.1 [58]	20.0 ± 1.5
FlashAttention (ours)	**17.4** ± 1.4

Model implementations	OpenWebText (ppl)	Training time (speedup)
GPT-2 small - Huggingface [87]	18.2	9.5 days (1.0×)
GPT-2 small - Megatron-LM [77]	18.2	4.7 days (2.0×)
GPT-2 small - FlashAttention	18.2	**2.7 days (3.5×)**
GPT-2 medium - Huggingface [87]	14.2	21.0 days (1.0×)
GPT-2 medium - Megatron-LM [77]	14.3	11.5 days (1.8×)
GPT-2 medium - FlashAttention	14.3	**6.9 days (3.0×)**

Models	ListOps	Text	Retrieval	Image	Pathfinder	Avg	Speedup
Transformer	36.0	63.6	81.6	42.3	72.7	59.3	-
FlashAttention	37.6	63.9	81.4	43.5	72.7	59.8	2.4×
Block-sparse FlashAttention	37.0	63.0	81.3	43.6	73.3	59.6	**2.8×**
Linformer [84]	35.6	55.9	77.7	37.8	67.6	54.9	2.5×
Linear Attention [50]	38.8	63.2	80.7	42.6	72.5	59.6	2.3×
Performer [12]	36.8	63.6	82.2	42.1	69.9	58.9	1.8×
Local Attention [80]	36.1	60.2	76.7	40.6	66.6	56.0	1.7×
Reformer [51]	36.5	63.8	78.5	39.6	69.4	57.6	1.3×
Smyrf [19]	36.1	64.1	79.0	39.6	70.5	57.9	1.7×

Model implementations	Context length	OpenWebText (ppl)	Training time (speedup)
GPT-2 small - Megatron-LM	1k	18.2	4.7 days (1.0×)
GPT-2 small - FlashAttention	1k	18.2	**2.7 days (1.7×)**
GPT-2 small - FlashAttention	2k	17.6	3.0 days (1.6×)
GPT-2 small - FlashAttention	4k	**17.5**	3.6 days (1.3×)

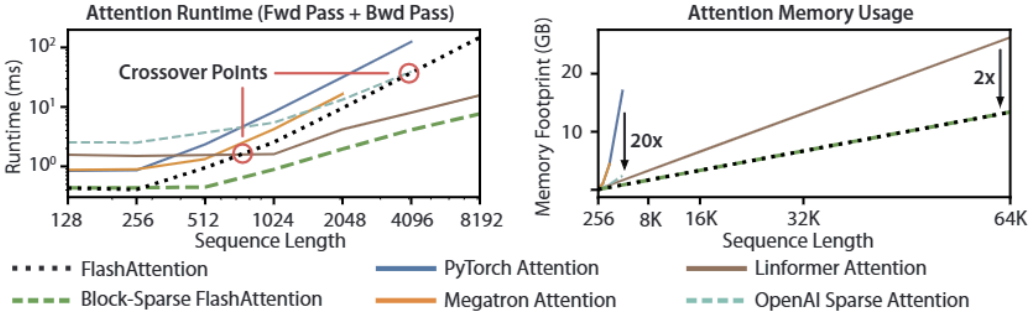


Figure 3: **Left:** runtime of forward pass + backward pass. **Right:** attention memory usage.

Table 5: Long Document performance (micro F_1) at different sequence lengths using FlashAttention.

	512	1024	2048	4096	8192	16384
MIMIC-III [47]	52.8	50.7	51.7	54.6	56.4	**57.1**
ECtHR [6]	72.2	74.3	77.1	78.6	**80.7**	79.2

Table 6: We report the first Transformer model that can achieve non-random performance on Path-X and Path-256.

Model	Path-X	Path-256
Transformer	✗	✗
Linformer [84]	✗	✗
Linear Attention [50]	✗	✗
Performer [12]	✗	✗
Local Attention [80]	✗	✗
Reformer [51]	✗	✗
SMYRF [19]	✗	✗
FlashAttention	**61.4**	✗
Block-sparse FlashAttention	56.0	**63.1**

- GPT 및 Transformer 기반 모델 학습 환경에서 테스트 수행
- 긴 시퀀스 길이 조건에서 기존 attention 대비 성능 비교
- 학습 속도와 메모리 사용량 중심 평가
- 다양한 sequence length 조건에서 실험 수행

5. Results

- 기존 attention 대비 최대 2~3배 속도 향상 확인
- GPU 메모리 사용량 크게 감소
- 긴 시퀀스 처리 시 batch size 증가 가능
- 정확도 손실 없이 동일 결과 유지
- 대규모 모델 학습 비용 감소 효과 확인

6. Insight

- Attention 병목은 계산이 아니라 메모리 접근 문제임을 명확히 보여줌
- 알고리즘 개선뿐 아니라 하드웨어 IO 구조 고려가 중요함을 시사
- LLM 학습 효율 개선의 핵심 기술로 자리잡는 중
- 이후 FlashAttention-2 등 후속 연구로 계속 발전 중
- 대규모 모델 최적화 연구의 방향성을 제시한 중요한 논문