Transformers- what’s next?

Sukanya Nath
Data Science Lab (DSL)
University of Bern

What are some limitations of the Transformers?

Limitations

1. High computational and infrastructure costs of training transformers.
2. Deployment of transformers in a service can have high latency.
3. Self attention layers generate pairwise comparisons of all the tokens in a sequence leading to $O(n^2)$ in computational time complexity.
4. Transformers often have a harsh limit on the length of the context that can be given to the model, making it difficult to allow long contexts.
5. Transformers cannot generalize to out of distribution data.
6. It is difficult to understand or explain the decisions taken by transformers.

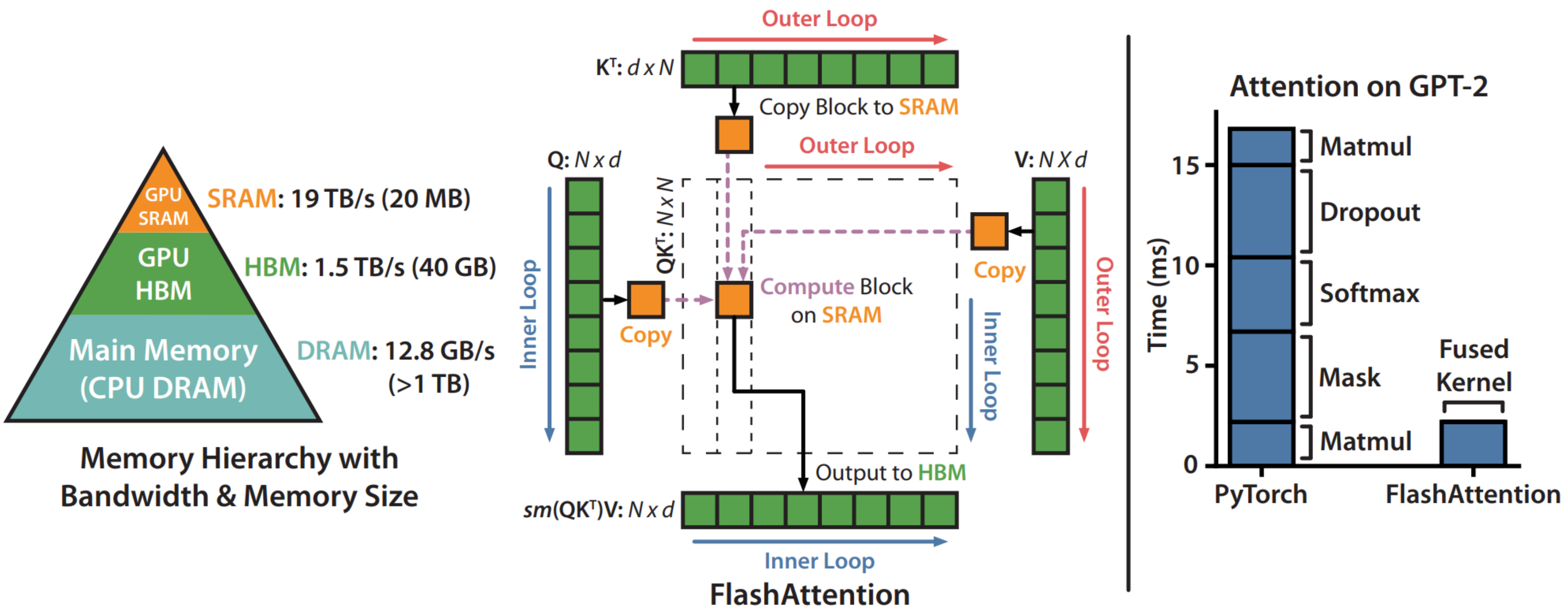
Flash Attention

How to reduce time consuming operations in self attention?

1. The time-consuming operations in Transformers can be attributed to the number of FLOPs and IO reads.
2. IO reads from High Bandwidth Memory (HBM) are much slower than SRAM.
3. Flash attention (Dao et al 2022) optimizes the number of IO reads with an IO-aware exact attention algorithm.
4. Instead of multiplying two large matrices in the HBM, the attention computation is restructured to to split the input into blocks loaded into SRAM and incrementally performing the softmax reduction (also known as tiling).
5. Finally, the output is written down to the HBM.
6. Using Flash attention BERT-large could be trained 15% faster and GPT2 3× faster than baseline implementations from HuggingFace.

Dao, T., Fu, D., Ermon, S., Rudra, A., & Ré, C. (2022). Flashattention: Fast and memory-efficient exact attention with io-awareness. *Advances in Neural Information Processing Systems*, *35*, 16344-16359.

Flash Attention



Dao, T., Fu, D., Ermon, S., Rudra, A., & Ré, C. (2022). Flashattention: Fast and memory-efficient exact attention with io-awareness. *Advances in Neural Information Processing Systems*, *35*, 16344-16359.

Quantization

How to reduce storage requirements of model weights?

Quantization is the process of reducing the memory required to store the model weights by reducing precision of the model from 32 bits to lower precision values.

0 10000000 11010101000100100110101

Quantization Level	Sign bit	Exponent	Precision/ Mantissa	Maximum value	Memory required
FP32	1	8	23	$\sim 3.4 \times 10^{38}$	4 bytes
FP16	1	5	10	$\sim 6.5 \times 10^{4}$	2 bytes
BFLOAT16	1	8	7	$\sim 3 \times 10^{38}$	2 bytes
INT8	1	0	7	127	1 byte

Quantization

Quantization is the process of reducing the memory required to store the model weights by reducing precision of the model from 32 bits to lower precision values.

$\underbrace{0}_{\text{Sign bit}}\ \underbrace{10000}_{\text{Exponent}}\ \underbrace{1101010100}_{\text{Precision/Mantissa}}$

Quantization Level	Sign bit	Exponent	Precision/ Mantissa	Maximum value	Memory required
FP32	1	8	23	$\sim 3.4 \times 10^{38}$	4 bytes
FP16	1	5	10	$\sim 6.5 \times 10^{4}$	2 bytes
BFLOAT16	1	8	7	$\sim 3 \times 10^{38}$	2 bytes
INT8	1	0	7	127	1 byte

Quantization

Quantization is the process of reducing the memory required to store the model weights by reducing precision of the model from 32 bits to lower precision values.

0 10000000 1101010

Quantization Level	Sign bit	Exponent	Precision/ Mantissa	Maximum value	Memory required
FP32	1	8	23	$\sim 3.4\times10^{38}$	4 bytes
FP16	1	5	10	$\sim 6.5\times10^{4}$	2 bytes
BFLOAT16	1	8	7	$\sim 3\times10^{38}$	2 bytes
INT8	1	0	7	127	1 byte

Quantization

Quantization is the process of reducing the memory required to store the model weights by reducing precision of the model from 32 bits to lower precision values.

0

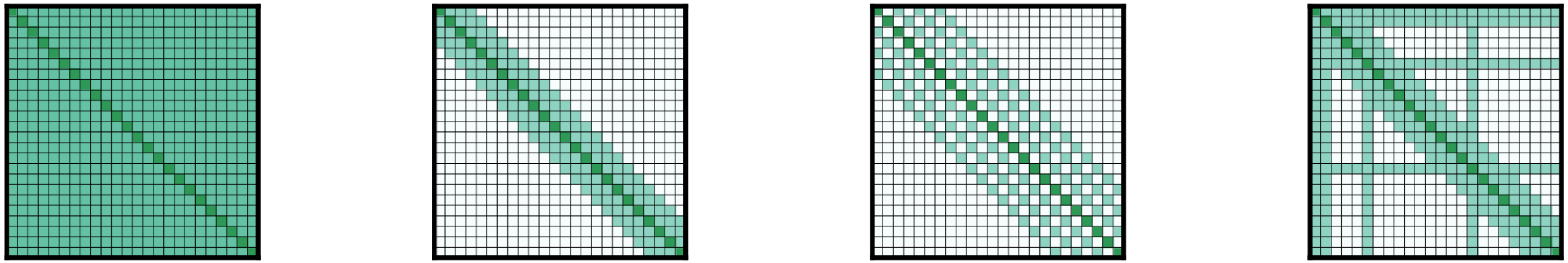
1101010

Quantization Level	Sign bit	Exponent	Precision/ Mantissa	Maximum value	Memory required
FP32	1	8	23	$\sim 3.4\times 10^{38}$	4 bytes
FP16	1	5	10	$\sim 6.5\times 10^{4}$	2 bytes
BFLOAT16	1	8	7	$\sim 3 \times 10^{38}$	2 bytes
INT8	1	0	7	127	1 byte

Longformer

How to allow longer contexts?

1. Goal is to allow longer contexts by allowing attention to scale linearly instead of quadratically by using a combination of sliding window, dilation and global attention for selected tokens.
2. Sliding window of size w = 512 tokens and a sequence length of n = 4096 could be accommodated.
3. Facilitates further pretraining of pretrained models.



(a) Full n^2 attention

(b) Sliding window attention $O(n \times w)$

(c) Dilated sliding window $O(n \times w)$

(d) Global+sliding window $O(n \times w)$

Beltagy, I., Peters, M. E., & Cohan, A. (2020). Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*.

How well do Transformers manage out of domain data?

Yadlowsky et al. 2023 show that

1. Transformers (notably LLMs) can well distinguish task families and learn in context when these task families were present in the pretraining data.
2. When Transformers encounter data outside of these known task families, very little evidence of out-of-distribution generalization was found.

Yadlowsky, S., Doshi, L., & Tripuraneni, N. (2023). Pretraining Data Mixtures Enable Narrow Model Selection Capabilities in Transformer Models. *arXiv preprint arXiv:2311.00871*.

Security Risks

1. Deployment of models without awareness can impact security in privacy-critical domains such as healthcare and finance.
2. Train a student/attack model to copy the original/ victim model by to gain sensitive information or to extract the model.
3. Pan et al. 2020 show that text embeddings can be reverse engineered to disclose sensitive information under certain conditions
   – A corpus of sensitive data with associated labels is available/ can be created.
   – The victim model can be accessed on demand.

Pan, X., Zhang, M., Ji, S., & Yang, M. (2020, May). Privacy risks of general-purpose language models. In *2020 IEEE Symposium on Security and Privacy (SP)* (pp. 1314-1331). IEEE.

Security Risks

1. Krishna et al 2019 show that it is possible to train an adversary model by feeding random sequences of words or task-specific queries for model extraction on a diverse set of NLP tasks.
2. Defense strategies against model extraction—membership classification and API watermarking—were ineffective against more sophisticated attacks.
3. Backdoors [Liu et al 2022] can be injected to NLP models such that they misbehave when the trigger words or sentences appear in an input sample.

Krishna, K., Tomar, G. S., Parikh, A. P., Papernot, N., & Iyyer, M. (2019). Thieves on sesame street! model extraction of bert-based apis. *arXiv preprint arXiv:1910.12366*.

Liu, Y., Shen, G., Tao, G., An, S., Ma, S., & Zhang, X. (2022, May). Piccolo: Exposing complex backdoors in nlp transformer models. In *2022 IEEE Symposium on Security and Privacy (SP)* (pp. 2025-2042). IEEE.