Overview

- Attention is a key component in the highly successful Transformer architecture.
- However, it has a quadratic computational cost, limiting the input (context) length.
- Previous approaches to alleviate this cost sacrifice Attention's random-access flexibility.
- We propose Landmark Attention to allow the attention itself to be used for retrieval, maintaining its random-access flexibility.

Landmark Attention

- Given the vector of dot products $v$ and grouping $g$, the grouped softmax is computed as:
  \[
  \sigma(v, g) := \text{GroupedSoftmax}(v, g) := \sum_{i} e^{v_i/g_i} / \sum_{j} e^{v_j/g_j}
  \]
- For each block, we create a separate group and put its regular (non-landmark) tokens in that group.
- When computing the attention weights for the $i$-th token, landmark tokens for other blocks are placed in the same group as the $i$-th token.
- The landmark token for the $i$-th token's block is ignored when computing the attention weights for the $i$-th token.
- Using the above grouping, the attention weight for each token can be computed as the product of the token's attention weight and its corresponding landmark token's attention weight.
- Under this scheme, the attention weights sum to one as in the standard Softmax function.

Inference

- First, break the input into chunks and augment the sequence by landmark tokens.
- Each chunk contains multiple blocks.
- The chunks are iteratively fed to the model from the beginning to the end.
- When processing each chunk at each layer, for each token we first compute the attention score to the landmark tokens currently in the cache.
- We only compute the attention score to tokens in the blocks of the top k highest scoring landmarks.
- This is a close approximation to training as tokens in blocks with low scoring landmarks will not receive a high weight anyway.
- The performance can be further improved by using nearest neighbor data structures.
- Computation cost is immediately improved by a factor of the block size.

Stingy Position Mapping

- Challenge: Transformers are unable to extrapolate to position indices not observed during training.
- Previous methods proposed to alleviate this issue: usually penalize or pre-atten-tion to long distance tokens.
- Instead we propose a special approximate position mapping scheme called stingy position mapping.
- The position mapping preserves the position of the last $k$ blocks but maps all earlier blocks to the same position.
- The retrieved blocks are prepend-ed in order to the current chunk with an empty block separating retrieved blocks in the last $k$ blocks from earlier ones.

Training

- Mostly standard training procedure is used except for the following changes:
  - Landmark tokens are added to the vocabulary, increasing the size by one.
  - Landmark tokens are inserted every after every $h$ tokens.
  - GroupSoftmax is used to compute Softmax in groups.
- A special grouping scheme imposes a hierarchy leading to the gating mechanism that can be used for retrieval.
- We train a 12-layer decoder-only Transformer with 8 heads and 128 hidden dimension on language modeling tasks.
- In particular, we consider perplexity on PG-19 dataset and math papers from arXiv.
- Results demonstrate that Landmark Attention allows inference at much larger context lengths than training context length.

Language Modeling

- The performance at the larger context length is comparable to a Transformer-XL trained directly at the larger context lengths on PG-19.

Retrieval Granularity

- Block retrieval can be performed on different levels of granularity:
  - At the most granular level the set of retrieved blocks can be different for each head and each token.
  - It is possible to further limit this granularity at inference, for increased system throughput.
  - Experiments in Section 4.2.1 show that reducing granularity hurts performance but most of the lost performance can be re-gained by retrieving more landmark tokens.

Combination with Flash Attention

- Using Flash Attention's block size to that of Landmark Attention, the two can be naturally combined.
- We provide open source implementation of this combination in Triton.

Context Miss Token

- Landmark Attention's grouping scheme can be adapted to introduce additional functionalities.
  - For example, we can use a Context Miss Token can be triggered to signal a need for accessing memory.
  - Experiments show that using this token around 50% of the retrievals can be dropped with minor effect on perplexity.

Extrapolating Positional Encoding

- Experiments on PG-19 (without landmarks) show adding random index increases after each landmark token help generalization to unseen position indices but a fully extrapolating positional encoding is yet to be built.
- Such encoding removes the need for stingy position mapping but Landmark Attention is still needed to reduce the computation cost.

Landmark Attention
Random-Access Infinite Context Length for Transformers

Amirkeivan Mohtashami, Martin Jaggi
Machine Learning and Optimization Laboratory

Reinforcement

- The chunks are iteratively fed to the model from the beginning to the end.
- We provide an open source implementation of this combination in Triton.

Pass Key Retrieval Task

- The task is to find a pass phrase hidden in a long piece of text.
- Instances of this task are generated by randomly choosing the pass phrase (a number between 1 and 50000) as well as its location.
- Prefixes and suffixes filled with a repetitive text are added to the pass key to ensure the desired length.
- We use the success rate as the evaluation metric.

Evaluation

- The original model fails to retrieve the pass key and even runs out of memory as the prompt gets longer (marked with a red cross).
- In contrast, the fine-tuned model using landmarks can successfully identify the pass key with a high success rate for contexts with over 32k tokens.
- The results demonstrate that our method can be successfully used even during fine-tuning to extend the context length limit to arbitrary large values.
- For very large context lengths, e.g. 32k, to avoid running out of memory, we use the capability offered by landmark attention allowing us to offload the majority of the key-value cache to CPU.

Retrieval

- Block retrieval can be performed on different levels of granularity.
- All the most granular level the set of retrieved blocks can be different for each head and each token.

- It is possible to further limit this granularity at inference, for increased system throughput.
- Experiments in Section 4.2.1 show that reducing granularity hurts performance but most of the lost performance can be re-gained by retrieving more landmark tokens.

Combination with Flash Attention

- Using Flash Attention's block size to that of Landmark Attention, the two can be naturally combined.
- We provide open source implementation of this combination in Triton.

Context Miss Token

- Landmark Attention's grouping scheme can be adapted to introduce additional functionalities.
  - For example, we can use a Context Miss Token can be triggered to signal a need for accessing memory.
  - Experiments show that using this token around 50% of the retrievals can be dropped with minor effect on perplexity.

Extrapolating Positional Encoding

- Experiments on PG-19 (without landmarks) show adding random index increases after each landmark token help generalization to unseen position indices but a fully extrapolating positional encoding is yet to be built.
- Such encoding removes the need for stingy position mapping but Landmark Attention is still needed to reduce the computation cost.

Retrieval

- Block retrieval can be performed on different levels of granularity.
- All the most granular level the set of retrieved blocks can be different for each head and each token.

- It is possible to further limit this granularity at inference, for increased system throughput.
- Experiments in Section 4.2.1 show that reducing granularity hurts performance but most of the lost performance can be re-gained by retrieving more landmark tokens.

Combination with Flash Attention

- Using Flash Attention's block size to that of Landmark Attention, the two can be naturally combined.
- We provide open source implementation of this combination in Triton.

Context Miss Token

- Landmark Attention's grouping scheme can be adapted to introduce additional functionalities.
  - For example, we can use a Context Miss Token can be triggered to signal a need for accessing memory.
  - Experiments show that using this token around 50% of the retrievals can be dropped with minor effect on perplexity.

Extrapolating Positional Encoding

- Experiments on PG-19 (without landmarks) show adding random index increases after each landmark token help generalization to unseen position indices but a fully extrapolating positional encoding is yet to be built.
- Such encoding removes the need for stingy position mapping but Landmark Attention is still needed to reduce the computation cost.