Extending Brainwave's BERT Implementation to Support Sparsity

Shruti Misra

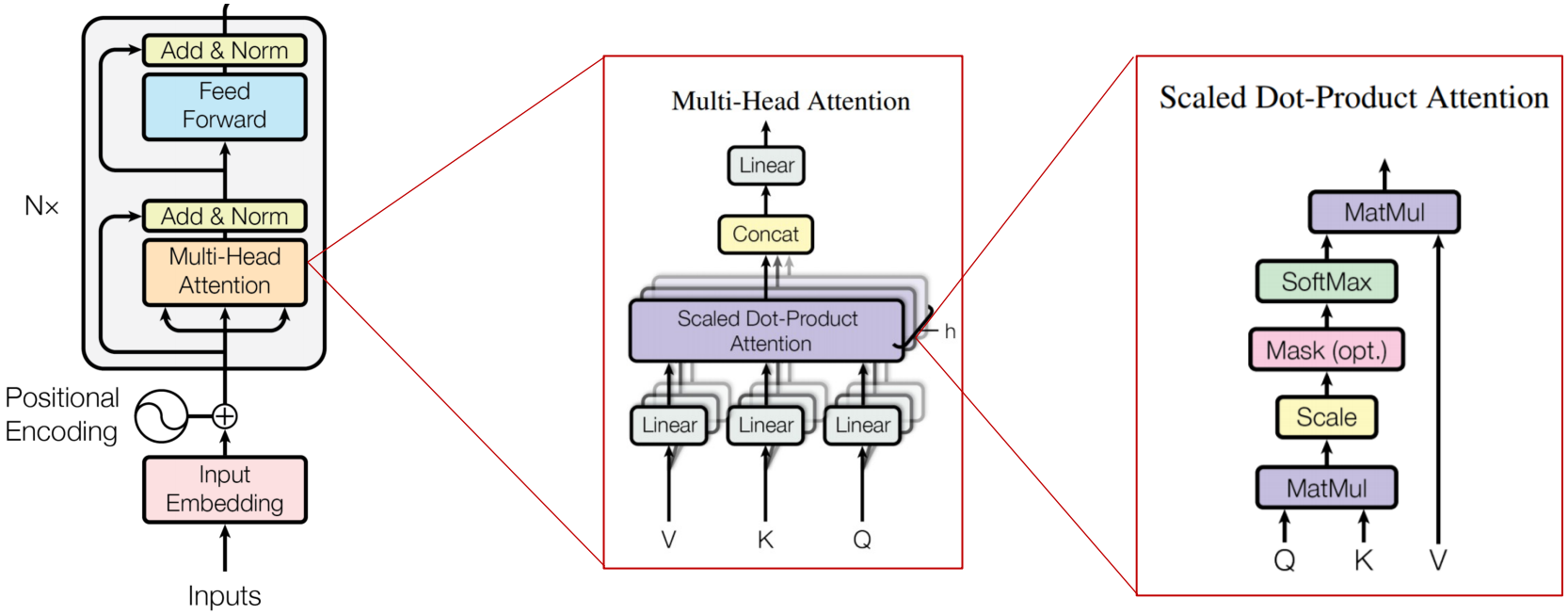
Goals

- Analyze the feasibility of supporting sparse computations within Brainwave's BERT framework.
- Design, implement/modify and test **firmware** to implement support for sparse computations in BERT.
- Identify/suggest ways the architecture could be extended to better support sparsity

Note: Many results have been redacted to protect IP

Transformer

A pre-trained language model for general NLP task such as question answering, next sentence prediction, sequence classification, etc.



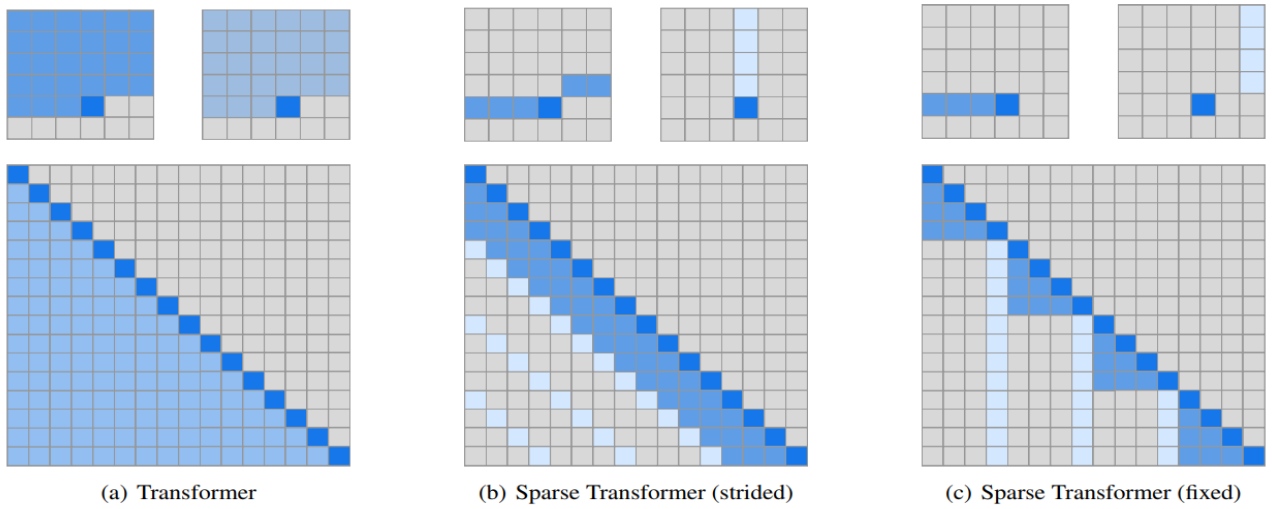
Sparse Self Attention

- Limitations of the full Transformer

  Memory and computational requirements of the full transformer grows quadratically with sequence length.

- Sparse Transformer model

  Sparse factorization of the attention matrix scales as O($n\sqrt{n}$) with sequence length (n).



(a) Transformer (b) Sparse Transformer (strided) (c) Sparse Transformer (fixed)

Child, Rewon, et al. "Generating long sequences with sparse transformers." *arXiv:1904.10509* (2019).

Performance Results Summary

- Significant decrease in latency is observed with added support for sparse computations.
- % Latency decrease scales with sequence length and sparsity.

References

https://ai.googleblog.com/2017/08/transformer-novel-neural-network.html

https://www.analyticsvidhya.com/blog/2019/06/understanding-transformers-nlp-state-of-the-art-models/

https://openai.com/blog/sparse-transformer/