Salisbury UNIVERSITY

Investigation of local quantized LLM performance on desktop and edge devices.

Author: Spencer Presley
Faculty Mentor: Dr. Junyi Tu
Salisbury University
Computer Science Department

Abstract

Since OpenAI's ChatGPT launched on November 30th, 2022 Large Language Models (LLMs) have been subject to countless research as well as attention from the public and political entities. While models like those offered from OpenAI are incredibly powerful they also come with the downside of privacy concerns, this leads some users to turn to open source options such as Meta's Llama. Running Llama models locally requires an immense amount of compute resources, making it out of reach for most people, for example, their smallest model currently is Llama 3.1 8B, which would still require a GPU with 24GB VRAM and 128GB system memory. In an effort to make these models more accessible to run locally quantization is needed. Quantization is the process of compressing a model from a higher-precision to a lower-precision, which lowers it's memory footprint as well as lessens it's computational demands, allowing it to be ran on a larger variety of hardware. In this research I look at one of those models, Bitnet b1.58. Bitnet b1.58 is built using the Llama architecture but with the feed forward layer of the transformer swapped out for a version that quantizes the models parameters. I train the model on the Enwik dataset and then benchmark it in a desktop environment as well as in an emulated android environment. The benchmarks allow for learning the viability of running LLMs locally on desktop as well as edge devices such as smart phones.

Methods

- Bitnet b1.58 implementation used as model.
- Python / PyTorch used for model implementation, training, and benchmarking. Robust training and benchmark scripts were used to train and benchmark the model.
- Enwik8/9 datasets used for training, these are commonly used to train and benchmark smaller models. Enwik8 is the first 100 million characters (~100MB) of English Wikipedia. Enwik9 is the first 1 billion characters (~1GB) of English Wikipedia.
- Both models were benchmarked on 4 datasets: Enwik8, Enwik9, Wikitext-2, Wikitext-103. Enwik8/9 were used to benchmark the model in it's pure form while Wikitext-2/103 were used to benchmark the model on actual text generation.
- Wikitext-2 contains 2 million words derived from English Wikipedia, Wikitext-103 contains 103 million words derived from English Wikipedia.
- Models were trained on an Nvidia RTX 4080 Graphics Processing Unit in Salisbury University's High Performance Computer Lab.

Results

The models were loaded and able to be benchmarked in an emulated android environment, the performance is good in some areas, particularly perplexity, but in others they're quite weak. This is to be expected as while the enwik8/9 datasets are good for benchmarking they are by no means a comprehensive set of training data. Much more data of many types would be needed to train a model capable of performing tasks users have become used to. Nonetheless the perplexity performance of these relatively small models shows it is possible to pack enough information into a model capable of being locally stored and ran on edge devices.

Conclusion

While the models are not in a state suitable to be used in real-world applications some performance metrics such as perplexity show to be quite promising. Applying these same methods to more well-rounded models such as the new Llama 3.1 8B could see incredibly powerful models available to users locally.

Model Information

Model	Training Set	Parameters	Size
SU-Bitnet b1.58 Small	Enwik8	23M	85.33MB
SU-Bitnet b1.58 Large	Enwik9	186M	707.26MB

Enwik8 Desktop Benchmarks

Model	PPL	Latency (ms)	RAM Usage (MB)	BPC	Compression	TCPPL	TCCA	TCCS	ATCP
SU-Bitnet b1.58 Small	11.76	18.97	144.33 (+59.00)	0.44	18.00	102.03	0.03	0.03	0.05
SU-Bitnet b1.58 Large	13.89	54.97	795.27 (+88.01)	0.47	16.87	70.39	0.04	0.04	0.06

Enwik9 Desktop Benchmarks

Model	PPL	Latency (ms)	RAM Usage (MB)	BPC	Compression	TCPPL	TCCA	TCCS	ATCP
SU-Bitnet b1.58 Small	13.36	18.66	144.33 (+59.00)	0.47	17.11	118.97	0.08	0.06	0.05
SU-Bitnet b1.58 Large	17.40	54.63	795.27 (+88.01)	0.52	15.53	73.21	0.10	0.07	0.05

Enwik8 Mobile Benchmarks (Emulated Android Environment)

Model	PPL	Latency (ms)	RAM Usage (MB)	BPC	Compression	TCPPL	TCCA	TCCS	ATCP
SU-Bitnet b1.58 Small	13.36	551.79	105.53 (+22.20)	0.42	19.24	102.04	0.03	0.03	0.05
SU-Bitnet b1.58 Large	15.70	2,798.24	741.48 (+34.22)	0.50	16.11	70.39	0.04	0.04	0.06

Enwik9 Mobile Benchmarks (Emulated Android Environment)

Model	PPL	Latency (ms)	RAM Usage (MB)	BPC	Compression	TCPPL	TCCA	TCCS	ATCP
SU-Bitnet b1.58 Small	11.63	577.55	105.53 (+22.20)	0.44	18.08	118.96	0.08	0.06	0.05
SU-Bitnet b1.58 Large	13.86	2,504.89	741.48 (+34.22)	0.47	16.87	73.21	0.10	0.07	0.05

Wikitext-2 Desktop Benchmarks

Model	TCPPL	TCCA	TCCS	NCPA	ATCP
SU-Bitnet b1.58 Small	118.00	0.01	0.02	0.15	0.08
SU-Bitnet b1.58 Large	71.75	0.02	0.03	0.12	0.08

Wikitext-103 Desktop Benchmarks

Model	TCPPL	TCCA	TCCS	NCPA	ATCP
SU-Bitnet b1.58 Small	118.00	0.01	0.02	0.15	0.08
SU-Bitnet b1.58 Large	71.75	0.02	0.03	0.12	0.08

Wikitext-2 Mobile Benchmarks (Emulated Android Environment)

Model	TCPPL	TCCA	TCCS	NCPA	ATCP
SU-Bitnet b1.58 Small	117.99	0.01	0.02	0.15	0.08
SU-Bitnet b1.58 Large	71.75	0.02	0.03	0.12	0.08

Wikitext-103 Mobile Benchmarks (Emulated Android Environments)

Model	TCPPL	TCCA	TCCS	NCPA	ATCP
SU-Bitnet b1.58 Small	117.99	0.01	0.02	0.15	0.08
SU-Bitnet b1.58 Large	71.75	0.02	0.03	0.12	0.08

Glossary

Parameters: Counted by summing the number of elements in all parameter tensors of the model. Indicates model complexity and capacity.

Perplexity (PPL): Calculated as exp(average cross-entropy loss). Measures how well the model predicts the next character in a sequence. Lower values indicate better predictions.

Latency: Measured by timing the forward pass of the model on a batch of data, averaged over multiple runs. Excludes data loading time.

Memory: Calculated by measuring the difference in system memory usage before and after running the model. Includes model parameters and intermediate activations.

Bits Per Character (BPC): Calculated as log2(Perplexity) / 8. Represents the average number of bits needed to encode each character under the model's predictions.

Compression: Computed as (8 / BPC). Indicates how much the model can theoretically compress the data compared to using 8 bits per character.

Text Continuation PPL (TCPPL): Calculates the mean probability that the model assigns to the correct character at each position when continuing text, even if it's not the top prediction.

Next Character Prediction Accuracy (NCPA): Calculated by having the model predict the next character in a sequence and comparing it to the true next character. Averaged over many samples.

Text Continuation Character Accuracy (TCCA): Computed by having the model continue text from a given prefix and calculating the perplexity of the true continuation under the model's predictions.

Text Continuation Character Similarity (TCCS): Calculates the mean probability that the model assigns to the correct character at each position when continuing text, even if it's not the top prediction.

Average Target Character Probability (ATCP): Computes the mean probability that the model assigns to the correct next character, even when it's not the top prediction. Provides a more nuanced view of model performance.

References

[1] Ma, S., Wang, H., Ma, L., Wang, L., Wang, W., Huang, S., ... & Wei, F. (2024). The Era of 1-bit LLMs: All Large Language Models are in 1.58 Bits. *arXiv preprint arXiv:2402.17764*.

[2] Wang, H., Ma, S., Dong, L., Huang, S., Wang, H., Ma, L., ... & Wei, F. (2023). Bitnet: Scaling 1-bit transformers for large language models. *arXiv preprint arXiv:2310.11453*.

[3] Intergovernmental Panel on Climate Change. (2021, August 9). Climate change widespread, rapid, and intensifying – IPCC. IPCC. https://www.ipcc.ch/2021/08/09/ar6-wg1-20210809-pr/

[6] Nagel, M., Fournarakis, M., Amjad, R. A., Bondarenko, Y., van Baalen, M., & Blankevoort, T. A white paper on neural network quantization. *arXiv 2021. arXiv preprint arXiv:2106.08295*

[7] Strubell, E., Ganesh, A., & McCallum, A. (2019). Energy and policy considerations for deep learning in NLP. *arXiv preprint arXiv:1906.02243*.

[8] MIT License. (2024). Open Source Initiative. Retrieved from https://opensource.org/licenses/MIT

[9] Gomez, K. (2024). BitNet [Software]. GitHub. https://github.com/kyegomez/BitNet