TRAINING 1.58BIT LLMS VIA DISTILLATION

Łukasz Leszko
[uzupełnić]

Filip Mateńko
f.matenko@student.uw.edu.pl

May 29, 2025

ABSTRACT

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

1 Introduction

In recent years, we have observed rapid growth in Large Language Models (LLMs). They have expanded both in their capabilities and in size. Unlike other fields of machine learning, LLMs do not seem to follow the usual rules of overfitting when increasing the number of parameters. When properly trained, more parameters generally lead to better performance in this field.

Unfortunately, the larger these models become, the more sophisticated hardware and computing power they require. Modern LLMs such as ChatGPT or DeepSeek-R1 demand multiple industrial-grade GPU accelerators to run efficiently. This requirement excludes individuals and organizations without access to such infrastructure from running these models locally, limiting full customization and integration.

Moreover, the energy consumption of human technology is considered one of the major issues of the 21st century. Data centers running LLMs consume enormous amounts of energy for both training and inference. One way to address this issue is through quantization—the process of reducing the precision of a model's parameters.Xiao et al. [2024]

Typically, parameters in LLMs are represented in 32-bit precision. The idea is to use lower precisions such as 16-bit, 4-bit, or even 1-bit to reduce the memory required to host and run the model. Many researchers around the world are approaching this task from different angles. Some claim to achieve performance close to that of unquantized LLMs.Wang et al. [2023] We decided to focus on the most extreme forms of quantization—reducing the representation to 1 bit (weights from -1, 1)—and compare it with 1.5-bit representation (weights from -1, 0, 1). The second approach is less common but has already been introduced in BitNet b1.58.Ma et al. [2024]

One possible method is to train the LLM in low precision from scratch. However, this approach, like any full training process, is computationally expensive. Additionally, it poses challenges when computing gradients with respect to discrete-valued parameters. An alternative is to take an existing high-precision model and distill it into a quantized model. Du et al. [2024] The full-precision model serves as a teacher to a smaller, quantized student model. In our work, we aim to explore an approach similar to that used in FBI-LLM.Liqun Ma [2024] In this work, the authors first binarize all linear transformer weights using a signum function—excluding embeddings, layer norms, and the head. They then introduce additional full-precision weights and biases for each binarized linear layer. These parameters, along with the head, become the only learnable components after bit quantization. The model is then distilled using a simple cross-entropy loss to align the responses of the student model with those of the teacher.

In our approach, we plan to explore both 1-bit and 1.5-bit quantizations. Additionally, we aim to experiment with different loss functions, such as KL divergence, Wasserstein distance, and symmetrized KL divergence. These approaches have been investigated in various prior works.Boizard et al. [2025], Du et al. [2024] Using KL divergence could better align the output distributions of the student and teacher models, as opposed to simply learning correct answers, which is the focus of cross-entropy loss. Moreover, Wasserstein distance may allow distillation even when the student and teacher have different output distributions.

For future work, it would also be valuable to compare training from scratch with distillation-based approaches. Some researchers have also explored white-box distillation, which aims to mimic not only the final outputs but also the hidden states of the teacher model.Gu et al. [2024]

2 Headings: first level

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula. See Section 2.

2.1 Headings: second level

Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consectetuer.

$$\xi_{ij}(t) = P(x_t = i, x_{t+1} = j | y, v, w; \theta) = \frac{\alpha_i(t) a_{ij}^{w_t} \beta_j(t+1) b_j^{v_{t+1}}(y_{t+1})}{\sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i(t) a_{ij}^{w_t} \beta_j(t+1) b_j^{v_{t+1}}(y_{t+1})} \tag{1}$$

2.1.1 Headings: third level

Suspendisse vel felis. Ut lorem lorem, interdum eu, tincidunt sit amet, laoreet vitae, arcu. Aenean faucibus pede eu ante. Praesent enim elit, rutrum at, molestie non, nonummy vel, nisl. Ut lectus eros, malesuada sit amet, fermentum eu, sodales cursus, magna. Donec eu purus. Quisque vehicula, urna sed ultricies auctor, pede lorem egestas dui, et convallis elit erat sed nulla. Donec luctus. Curabitur et nunc. Aliquam dolor odio, commodo pretium, ultricies non, pharetra in, velit. Integer arcu est, nonummy in, fermentum faucibus, egestas vel, odio.

Paragraph Sed commodo posuere pede. Mauris ut est. Ut quis purus. Sed ac odio. Sed vehicula hendrerit sem. Duis non odio. Morbi ut dui. Sed accumsan risus eget odio. In hac habitasse platea dictumst. Pellentesque non elit. Fusce sed justo eu urna porta tincidunt. Mauris felis odio, sollicitudin sed, volutpat a, ornare ac, erat. Morbi quis dolor. Donec pellentesque, erat ac sagittis semper, nunc dui lobortis purus, quis congue purus metus ultricies tellus. Proin et quam. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos. Praesent sapien turpis, fermentum vel, eleifend faucibus, vehicula eu, lacus.

3 Examples of citations, figures, tables, references

3.1 Citations

Citations use `natbib`. The documentation may be found at

`http://mirrors.ctan.org/macros/latex/contrib/natbib/natnotes.pdf`

Here is an example usage of the two main commands (`citet` and `citep`): Some people thought a thing [**??**] but other people thought something else [**?**]. Many people have speculated that if we knew exactly why **?** thought this...

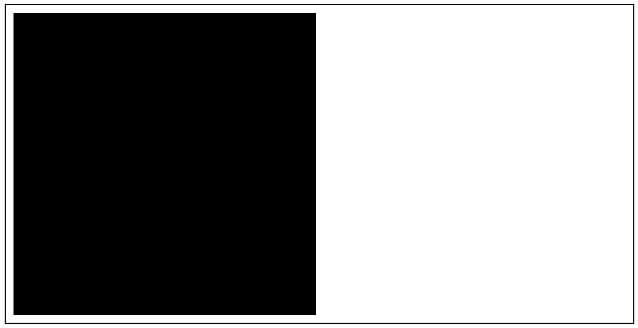

Figure 1: Sample figure caption.

Table 1: Sample table title

Part		
Name	Description	Size (μm)
Dendrite	Input terminal	$\sim$100
Axon	Output terminal	$\sim$10
Soma	Cell body	up to 10^6

3.2 Figures

Suspendisse vitae elit. Aliquam arcu neque, ornare in, ullamcorper quis, commodo eu, libero. Fusce sagittis erat at erat tristique mollis. Maecenas sapien libero, molestie et, lobortis in, sodales eget, dui. Morbi ultrices rutrum lorem. Nam elementum ullamcorper leo. Morbi dui. Aliquam sagittis. Nunc placerat. Pellentesque tristique sodales est. Maecenas imperdiet lacinia velit. Cras non urna. Morbi eros pede, suscipit ac, varius vel, egestas non, eros. Praesent malesuada, diam id pretium elementum, eros sem dictum tortor, vel consectetuer odio sem sed wisi. See Figure 1. Here is how you add footnotes. [1] Sed feugiat. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Ut pellentesque augue sed urna. Vestibulum diam eros, fringilla et, consectetuer eu, nonummy id, sapien. Nullam at lectus. In sagittis ultrices mauris. Curabitur malesuada erat sit amet massa. Fusce blandit. Aliquam erat volutpat. Aliquam euismod. Aenean vel lectus. Nunc imperdiet justo nec dolor.

3.3 Tables

See awesome Table 1.

The documentation for `booktabs` ('Publication quality tables in LaTeX') is available from:

`https://www.ctan.org/pkg/booktabs`

3.4 Lists

- Lorem ipsum dolor sit amet
- consectetur adipiscing elit.
- Aliquam dignissim blandit est, in dictum tortor gravida eget. In ac rutrum magna.

References

Guangxuan Xiao, Ji Lin, Mickael Seznec, Hao Wu, Julien Demouth, and Song Han. Smoothquant: Accurate and efficient post-training quantization for large language models, 2024. URL `https://arxiv.org/abs/2211.10438`.

Hongyu Wang, Shuming Ma, Li Dong, Shaohan Huang, Huaijie Wang, Lingxiao Ma, Fan Yang, Ruiping Wang, Yi Wu, and Furu Wei. Bitnet: Scaling 1-bit transformers for large language models, 2023. URL `https://arxiv.org/abs/2310.11453`.

[1] Sample of the first footnote.

Shuming Ma, Hongyu Wang, Lingxiao Ma, Lei Wang, Wenhui Wang, Shaohan Huang, Li Dong, Ruiping Wang, Jilong Xue, and Furu Wei. The era of 1-bit llms: All large language models are in 1.58 bits, 2024. URL https://arxiv.org/abs/2402.17764.

Dayou Du, Yijia Zhang, Shijie Cao, Jiaqi Guo, Ting Cao, Xiaowen Chu, and Ningyi Xu. Bitdistiller: Unleashing the potential of sub-4-bit llms via self-distillation, 2024. URL https://arxiv.org/abs/2402.10631.

Zhiqiang Shen Liqun Ma, Mingjie Sun. Fbi-llm: Scaling up fully binarized llms from scratch via autoregressive distillation, 2024. URL https://arxiv.org/pdf/2407.07093.

Nicolas Boizard, Kevin El Haddad, Céline Hudelot, and Pierre Colombo. Towards cross-tokenizer distillation: the universal logit distillation loss for llms, 2025. URL https://arxiv.org/abs/2402.12030.

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. Minillm: Knowledge distillation of large language models, 2024. URL https://arxiv.org/abs/2306.08543.