TRAINING 1.58BIT LLMS VIA DISTILLATION

PROPOSAL

Łukasz Leszko
ll438580@student.mimuw.edu.pl

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

May 30, 2025

ABSTRACT

In this work, we propose an evaluation of LLMs distilled from a full-precision model down to 1.58-bit precision. Our evaluation will focus on the impact of different loss functions and quantization methods on the performance of the distilled model. We will try to validate claims that 1.58-bit quantization can achieve performance close to that of the full-precision models.

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 [1].

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 [2]. We decided to focus on the most extreme forms of quantization - reducing the representation to 1 bit (weights from set $\{-1, 1\}$) - and compare it with 1.58-bit representation (weights from set $\{-1, 0, 1\}$). The second approach is less common but has already been introduced in BitNet b1.58 [3].

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 [4]. 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 [5], namely distillation with Quantization Aware Training. 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 1.58-bit quantizations. Additionally, we aim to experiment with different loss functions, such as KL divergence, Confidence-Aware KL divergence, Wasserstein distance. These approaches have been investigated in various prior works [4, 6]. 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 use different tokenizers.

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 [7].

2 Experimental setup

Choosing model size is crucial for our experiments. Results from [3] show that starting from 3B parameters, a 1.58-bit distilled student was able to match its teacher model. Taking that into account, as well as the fact that we are limited by computational power, we decided to use a model in a similar size range (2-3B parameters). The choice of the actual architecture will be made later, based on empirical results.

We follow the quantization process proposed in BitNet and BitNet b1.58, which replaces linear layers with a custom linear layer (BiLinear) that performs 1-bit or 1.58-bit quantization on weights during training. This approach allows us to easily add 1-bit and 1.58-bit quantization capabilities to any model with linear layers (not just transformers).

To perform the distillation, we extend the FBI-LLM approach to support 1.58-bit quantization simply by replacing the quantization function with the one proposed in BitNet b1.58. While FBI-LLM uses cross-entropy loss, we will also experiment with the following loss functions:

- (Forward) KL Divergence - a measure of how the teacher's output distribution differs from the student's output distribution
- Confidence-Aware KL divergence - a weighted sum of forward and reverse KL divergence with weights based on the confidence of the teacher model's predictions
- Wasserstein Distance - a metric for quantifying dissimilarities between distributions stemming from optimal transport theory

The architecture of the BiLinear layer proposed in FBI-LLM closely follows the one used in OneBit [8] but differs from the one used in BitNet b1.58 by adding small full-precision learnable parameters. We will investigate how different implementations of the BiLinear layer affect the performance of the distilled model. To that end, we will compare the BiLinear implementations from FBI-LLM, OneBit, and BitNet b1.58.

The baseline for our experiments will be a 1-bit LLM, which has been trained using the exact FBI-LLM approach on the Amber dataset [9]. The Amber dataset is an agglomeration of RefinedWeb [10], StarCoder [11], and RedPajama-v1 [12] and contains 1.26 trillion tokens. We will also utilize parts of this dataset to train our 1.58-bit LLMs.

To evaluate our models, we will use EleutherAI's evaluation suite, which provides a set of benchmarks for LLMs. We will focus on measuring perplexity and accuracy on various datasets, like WikiText, MMLU, and others. In addition, as noted in the FBI-LLM paper, training extremely quantized models with distillation may be unstable and result in models that fail to converge. We will monitor this stability by tracking the flip-flop rate [13], which is the percentage of quantized weights that changed sign during a training step.

One thing to note is that while the quantized weights, on disk, can be stored in a packed format, during training, we still need to store them in float16 format to compute gradients. Similarly, during inference, multiplying by a 1.58-bit quantized matrix does not require the `MUL` operator; however, to benefit from that, we need to utilize custom hardware and kernels, which is outside of our expertise.

References

[1] Jiedong Lang, Zhehao Guo, and Shuyu Huang. A comprehensive study on quantization techniques for large language models, 2024.

[2] 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.

[3] 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.

[4] 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.

[5] Zhiqiang Shen Liqun Ma, Mingjie Sun. Fbi-llm: Scaling up fully binarized llms from scratch via autoregressive distillation, 2024.

[6] Nicolas Boizard, Kevin El Haddad, Céline Hudelot, and Pierre Colombo. Towards cross-tokenizer distillation: the universal logit distillation loss for llms, 2025.

[7] Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. Minillm: Knowledge distillation of large language models, 2024.

[8] Yuzhuang Xu, Xu Han, Zonghan Yang, Shuo Wang, Qingfu Zhu, Zhiyuan Liu, Weidong Liu, and Wanxiang Che. Onebit: Towards extremely low-bit large language models, 2024.

[9] Zhengzhong Liu, Aurick Qiao, Willie Neiswanger, Hongyi Wang, Bowen Tan, Tianhua Tao, Junbo Li, Yuqi Wang, Suqi Sun, Omkar Pangarkar, Richard Fan, Yi Gu, Victor Miller, Yonghao Zhuang, Guowei He, Haonan Li, Fajri Koto, Liping Tang, Nikhil Ranjan, Zhiqiang Shen, Xuguang Ren, Roberto Iriondo, Cun Mu, Zhiting Hu, Mark Schulze, Preslav Nakov, Tim Baldwin, and Eric P. Xing. Llm360: Towards fully transparent open-source llms, 2023.

[10] Guilherme Penedo, Quentin Malartic, Daniel Hesslow, Ruxandra Cojocaru, Alessandro Cappelli, Hamza Alobeidli, Baptiste Pannier, Ebtesam Almazrouei, and Julien Launa. The refinedweb dataset for falcon llm: Outperforming curated corpora with web data, and web data only, 2023.

[11] Raymond Li, Loubna Ben Allal, Yangtian Zi, Niklas Muennighoff, Denis Kocetkov, Chenghao Mou, Marc Marone, Christopher Akiki, Jia Li, Jenny Chim, Qian Liu, Evgenii Zheltonozhskii, Terry Yue Zhuo, Thomas Wang, Olivier Dehaene, Mishig Davaadorj, Joel Lamy-Poirier, João Monteiro, Oleh Shliazhko, Nicolas Gontier, Nicholas Meade, Armel Zebaze, Ming-Ho Yee, Logesh Kumar Umapathi, Jian Zhu, Benjamin Lipkin, Muhtasham Oblokulov, Zhiruo Wang, Rudra Murthy, Jason Stillerman, Siva Sankalp Patel, Dmitry Abulkhanov, Marco Zocca, Manan Dey, Zhihan Zhang, Nour Fahmy, Urvashi Bhattacharyya, Wenhao Yu, Swayam Singh, Sasha Luccioni, Paulo Villegas, Maxim Kunakov, Fedor Zhdanov, Manuel Romero, Tony Lee, Nadav Timor, Jennifer Ding, Claire Schlesinger, Hailey Schoelkopf, Jan Ebert, Tri Dao, Mayank Mishra, Alex Gu, Jennifer Robinson, Carolyn Jane Anderson, Brendan Dolan-Gavitt, Danish Contractor, Siva Reddy, Daniel Fried, Dzmitry Bahdanau, Yacine Jernite, Carlos Muñoz Ferrandis, Sean Hughes, Thomas Wolf, Arjun Guha, Leandro von Werra, and Harm de Vries. Starcoder: may the source be with you!, 2023.

[12] Maurice Weber, Daniel Fu, Quentin Anthony, Yonatan Oren, Shane Adams, Anton Alexandrov, Xiaozhong Lyu, Huu Nguyen, Xiaozhe Yao, Virginia Adams, Ben Athiwaratkun, Rahul Chalamala, Kezhen Chen, Max Ryabinin, Tri Dao, Percy Liang, Christopher Ré, Irina Rish, and Ce Zhang. Redpajama: an open dataset for training large language models, 2024.

[13] Zechun Liu, Zhiqiang Shen, Shichao Li, Koen Helwegen, Dong Huang, and Kwang-Ting Cheng. How do adam and training strategies help bnns optimization?, 2021.