Project Proposal - A Practical LLM Pruning Framework

Michal Kuchar

1 Introduction

The rising popularity of Large Language Models (LLMs) has drawn attention to the vast resource requirements of practically deploying them. The largest of LLMs can consist of up to hundreds of billions floating-point weights, demanding hundreds of gigabytes of memory. In practice, LLMs are usually deployed on accelerator hardware with relatively limited memory, requiring large numbers of accelerators for deployment at scale. This creates a prohibitively large demand for energy, which has prompted the development of a variety of techniques to reduce the post-training footprint of LLMs.

Chief among these techniques are quantization and pruning. Quantization refers to reducing the precision of in-memory representations of model weights by reducing the number of bits. Pruning refers to removing components of the network by removing individual weights (unstructured pruning), or removing groups of weights (structured pruning). [5] [8] While research around quantization techniques seems to be converging with certain techniques becoming ubiquitous, there appear to be many novel developments around pruning, leveraging the structure of LLMs.

The purpose of this project is to synthesize these emerging pruning techniques into a unified "toolkit" and use it to better understand the pruning vs. quantization tradeoff.

2 The LLM Quantization Landscape

Quantization often comes in two forms - Post-Training Quantization (PTQ) and Quantization Aware Training (QAT). PTQ compresses a model's weights and/or activations to a lower bit precision without re-training. QAT optimizes the weights such that once quantized, they minimize the output difference between the quantized and full-precision versions. It seems that many models can be quantized to a precision as low as 4 bits, with remarkably small performance degradation relative to the reduction in memory usage. [8] Even more remarkably, recently [3] released a 1.58-bit model which appears to rival open-weight models of similar parameter counts at a fraction of the memory and energy usage. Overall, pruning seems to offer compelling cost savings. The savings can be further enhanced with pruning, but one does need to balance the two because they do affect one another. [2]

3 The LLM Pruning Landscape

In its most basic form, an LLM is built out of multiple Transformer blocks (or layers) in sequence, where the output of a given layer becomes an input for the following layer. For cases like this, [4] propose structured pruning techniques that involve removing entire Transformer blocks or their constituent components - the Attention or the subsequent Feed-Forward Network (FFN). The pruning is executed based on thresholds over a pruning metric - a measure of how much the component impacts the outcome of the network. More advanced LLM architectures introduce residual connections that combine the input into a given layer with the transformed output produced by that same layer. As demonstrated by [1], such architectures can be pruned by removing Transformer blocks and re-directing the affected residual connection to the following Transformer block.

In a slight deviation from the typical Transformer-based architecture, some LLMs replace the FFN in Transformer blocks with a Mixture-of-Experts (MoE) layer. An MoE layer consists of multiple smaller "expert" sub-networks along with a router component. The router is trained in concert with the experts to route tokens to them. For MoE models, it makes most sense to prune the expert networks because they make up most of the parameters. This can be achieved using a pruning metric defined in terms of the router output - the least frequently utilized expert

weights are pruned. [6]

In addition, Multimodal Large Language Models (MLLMs) introduce an extra layer of complexity by creating an infrastructure of multiple deep learning components around an LLM. Modality interfaces are introduced to pre-process data that are not of a text-based nature, and learnable "connector" networks map the extracted features to LLM-readable token space. For example, to pre-process images, Osprey uses the convolutional network ConvNext-L, while MiniGPT-4 uses a Transformer-based ViT-G/14 encoder. [7] This means that the modality interfaces themselves can be candidates for pruning when looking at the MLLM as a whole, and the same goes for the connector networks.

4 The Goal of the Project

LLM and MLLM architectures are developing rapidly, and so are the corresponding pruning techniques. Quantization is popular, and it seems useful to further investigate its relationship with pruning. There already exist attempts at comprehensive quantization toolkits, and we aim to create something similar for pruning. These tools can then be combined to study the quantization vs. pruning tradeoff. More precisely, we aim to:

1. Create a comprehensive pruning toolkit that aggregates the latest developments in techniques.
2. Measure the performance of models in varying pruned and quantized configurations to find ones that maximize performance and minimize resource usage.

References

[1] Andrey Gromov et al. *The Unreasonable Ineffectiveness of the Deeper Layers*. Version Number: 2. 2024. DOI: 10.48550/ARXIV.2403.17887. URL: https://arxiv.org/abs/2403.17887 (visited on 04/23/2025).

[2] Andrey Kuzmin et al. *Pruning vs Quantization: Which is Better?* Version Number: 2. 2023. DOI: 10.48550/ARXIV.2307.02973. URL: https://arxiv.org/abs/2307.02973 (visited on 04/23/2025).

[3] Shuming Ma et al. *BitNet b1.58 2B4T Technical Report*. Version Number: 1. 2025. DOI: 10.48550/ARXIV.2504.12285. URL: https://arxiv.org/abs/2504.12285 (visited on 04/23/2025).

[4] Xin Men et al. *ShortGPT: Layers in Large Language Models are More Redundant Than You Expect*. Version Number: 3. 2024. DOI: 10.48550/ARXIV.2403.03853. URL: https://arxiv.org/abs/2403.03853 (visited on 04/23/2025).

[5] Zhongwei Wan et al. *Efficient Large Language Models: A Survey*. Version Number: 4. 2023. DOI: 10.48550/ARXIV.2312.03863. URL: https://arxiv.org/abs/2312.03863 (visited on 04/23/2025).

[6] Yanyue Xie et al. *MoE-Pruner: Pruning Mixture-of-Experts Large Language Model using the Hints from Its Router*. Version Number: 1. 2024. DOI: 10.48550/ARXIV.2410.12013. URL: https://arxiv.org/abs/2410.12013 (visited on 04/23/2025).

[7] Shukang Yin et al. "A Survey on Multimodal Large Language Models". In: (2023). Publisher: arXiv Version Number: 4. DOI: 10.48550/ARXIV.2306.13549. URL: https://arxiv.org/abs/2306.13549 (visited on 04/23/2025).

[8] Xunyu Zhu et al. *A Survey on Model Compression for Large Language Models*. Version Number: 4. 2023. DOI: 10.48550/ARXIV.2308.07633. URL: https://arxiv.org/abs/2308.07633 (visited on 04/23/2025).