Alma Mater Studiorum · Università di Bologna

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

ARTIFICIAL INTELLIGENCE

An effective strategy for reducing the size of LLaMA-based Language Models

Supervisor:
Prof. Francesco Conti

Co-supervisor:
Luca Bompani

Presented by:
Angelo Galavotti

Session I
Academic Year 2024/2025

Alla migliore madre del mondo,
al miglior padre del mondo.

Abstract

Pending

Contents

1 Introduction **1**

1.1 The social impact of LLMs 2

1.2 The environmental impact of LLMs 3

1.3 Scope 4

1.3.1 The main objective 4

1.3.2 How can optimization techniques help? 6

1.4 Document structure 7

2 Background and Related Work **9**

2.1 Early language models 9

2.2 The architecture of Transformers 10

2.3 The structure of Large Language Models 11

2.4 A look at the target hardware 11

2.5 Relevant compression and optimization techniques 11

3 Methodology **13**

3.1 Preliminary research: Franken-LLaMA 13

3.2 Pruning methods 13

3.2.1 Depth-wise pruning 13

3.2.2 Width-wise pruning (WANDA-based) 13

3.2.3 LoRA 13

3.3 Quantization and EoRA 13

3.4 Evaluation criteria 13

4 Implementation 15

4.1 Implementation of the Pipeline 15

4.1.1 Depth-wise Pruning 15

4.1.2 Width-wise Pruning (WANDA-based) 15

4.1.3 LoRA 15

4.1.4 Quantization and EoRA 15

4.2 Evaluation 15

5 Experimental Results and Analysis 17

5.1 Preliminary observation of the results 17

5.2 Results on TriviaQA 17

5.3 Results on WikiText 17

6 Conclusion and Future Work 19

6.1 Future work 19

6.2 Final remarks 19

References 21

List of Figures

1.1 Comparison of the energy consumption of different LLMs for small and large queries. The data highlights the significant increase in energy requirements as query size grows. These graphs have been sourced from [5]. 5

2.1 A visual representation of the LSTM architecture. The three gates (input, forget, and output) control the flow of information in and out of the cell state, allowing it to retain relevant information over long sequences. 10

2.2 On the left, a standard Transformer architecture, which consists of an encoder and a decoder. On the right, we can observe the same architecture, while highlighting the only the layers used by a GPT-style model, which is a decoder-only transformer. The main difference is that the decoder does not attend to the encoder's output, allowing for auto-regressive generation. 12

Chapter 1

Introduction

It is no secret that, in the last three years, *Large Language Models* (LLMs) have fundamentally transformed our relationship with technology. Their impact rivals the most significant innovations of the past century, such as the internet and the smartphone. When people contemplate *Artificial Intelligence* (AI) today, they immediately think of ChatGPT or Claude, which have seamlessly integrated into our daily routines. Yet these powerful tools come with significant environmental concerns. Their development and operation consume vast amounts of energy and water resource: modern data centers supporting these models require extensive cooling systems and electricity consumption that can rival small cities.

The computational complexity of these systems necessitates cloud-based deployment, which not only amplifies their environmental footprint but also fundamentally restricts user autonomy. This cloud dependency creates a concerning power dynamic where users have limited control over their tools, while simultaneously enabling extensive data collection practices and potential surveillance mechanisms that would be impossible with local, user-controlled alternatives.

In this more conversational opening chapter we will briefly examine the environmental and social impact of LLMs while highlighting the growing imperative for efficient, locally-deployable models that democratize access without depleting our planet's resources. Afterwards, we will outline the scope of this project, which aims to explore the potential of compression techniques to make LLMs more efficient.

The future of AI depends not just on what these models can do, but how sustainably they can do it.

1.1 The social impact of LLMs

The widespread adoption of LLMs has created ripple effects across virtually every sector of society, fundamentally altering how we work, learn, and create. In education, these tools have sparked heated debates about academic integrity while simultaneously offering new possibilities for personalized learning and accessibility for students with disabilities [1]. The workplace has experienced perhaps the most dramatic shifts, with entire professions grappling with automation anxiety while others discover unprecedented productivity gains. Creative industries find themselves in particularly complex territory: writers, artists, and content creators must navigate between leveraging AI as a collaborative tool and protecting their intellectual property from being absorbed into training datasets without consent or compensation [2].

Perhaps what is most striking is how these models have democratized access to sophisticated capabilities that were once the exclusive domain of experts. A small business owner can now generate marketing copy that rivals professional agencies, students can receive tutoring in subjects where human expertise might be scarce, and non-programmers can write functional code with natural language instructions. Yet this democratization comes with a troubling caveat: it's entirely dependent on maintaining access to centralized, corporate-controlled systems. When OpenAI experiences an outage, millions of users worldwide suddenly lose access to tools they've integrated into their daily workflows. When pricing models change, entire business models built around AI assistance can become unsustainable overnight.

This dependency becomes even more concerning when we consider the data these systems collect. Every interaction, every query, every creative prompt may potentially become part of these companies' datasets, raising questions about privacy and intellectual property. As such, the need for transparency and user control over these systems has never been more urgent, and many believe that the future of AI must prioritize local, user-deployable models that empower individuals rather than centralizing power in the hands of a few corporations.

1.2 The environmental impact of LLMs

The computational demands of modern LLMs create an environmental footprint that grows exponentially with model size and usage. Training GPT-3, for instance, consumed an estimated 1,287 MWh of electricity, which is enough to power an average American home for over a century [3]. However, training represents only the tip of the iceberg; the real environmental cost lies in inference, where billions of daily queries across millions of users create a continuous drain on global energy resources. A recent study by Jegham et al. [5] estimates that OpenAI's GPT-4.5 requires 6.7 Wh of energy for a medium sized query (i.e. 100 input tokens, and outputting 300 tokens) to be processed. This figure grows to 20.5 Wh for a larger query (i.e. 1000 input tokens, and outputting 1000 tokens). To put this into perspective, this is equal to charging an average 40 Wh laptop battery to 50%. A graph comparing the energy consumption of different LLMs with different prompt sizes is shown in Figure 1.1.

Data centers housing these models consume approximately 1-2% of global electricity, a figure that's projected to reach 8% by 2030 if current trends continue [4]. The infrastructure supporting a single large-scale LLM requires thousands of high-performance GPUs running 24/7, each consuming as much power as several households.

Water consumption presents an equally pressing concern that receives far less attention. Modern data centers require extensive cooling systems, with some facilities consuming millions of gallons daily. According to their sustainability report [6], Google's water usage increased by 20% between 2021 and 2022, then by 17% from 2022 to 2023, and is largely attributed to AI inference operations [7]. In regions already facing water scarcity, this additional demand creates direct competition with human needs and agricultural requirements.

On the other hand, one has to keep in mind the embedded carbon cost of the hardware itself. Each GPU cluster supporting LLM operations represents significant emissions from manufacturing, shipping, and eventual disposal. The rapid pace of AI advancement drives frequent hardware upgrades, creating electronic waste streams that the recycling industry struggles to process effectively.

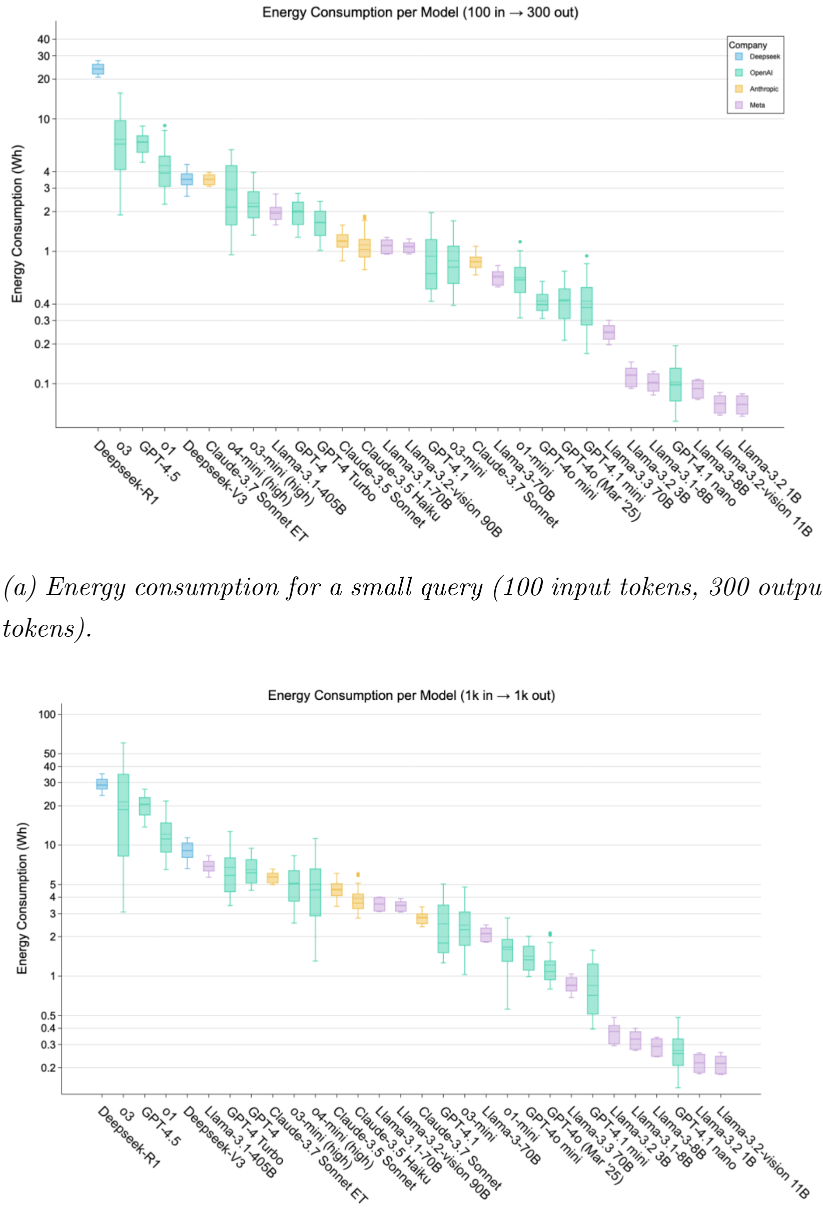
These environmental costs scale directly with model size and usage frequency, creating a fundamental tension between AI capabilities and sustainability.

1.3 Scope

While we've examined several critical challenges facing LLMs, it's worth emphasizing that these models hold significant potential for positive impact. As such, what was outlined in the previous sections points instead toward an urgent need for alternatives to the current paradigm of massive, centralized LLMs. A promising solution lies in compression techniques that can dramatically reduce model size while preserving core functionality. Through compression, billion-parameter server-sized models can be scaled down to more manageable ones that run much more efficiently.

1.3.1 The main objective

In this context, the objective of this project is to push the boundaries of memory efficiency for small-scale LLMs by applying a targeted suite of



(a) Energy consumption for a small query (100 input tokens, 300 output tokens).

(b) Energy consumption for a large query (1000 input tokens, 1000 output tokens).

Figure 1.1: Comparison of the energy consumption of different LLMs for small and large queries. The data highlights the significant increase in energy requirements as query size grows. These graphs have been sourced from [5].

compression techniques. We begin with a distilled model, a 1B parameter variant of LLaMA 3, which already represents a significantly reduced footprint compared to full-scale LLMs. From there, we explore and implement further optimizations, including depth and width pruning, *Low-Rank Adaptation* (LoRA), and quantization.

Pruning allows us to remove redundant layers or neurons from the network architecture, trimming excess capacity without substantial loss of capability. Quantization reduces the bit-width of model weights and activations, decreasing both memory usage and compute requirements. LoRA, meanwhile, introduces a lightweight, parameter-efficient training mechanism that reduces the cost of fine-tuning and adaptation without (necessarily) modifying the base model weights. In this way, the model can be adapted to new tasks or domains with minimal overhead.

The reason behind the focus on maximizing memory efficiency is related to the fact that memory represents the most expensive constraint in our target deployment scenario, where the intended hardware platform consists of a low-power RISC-V SoC with severely constrained memory resources (further details on the target hardware can be found in Section 2.4). Crucially, optimizations that reduce memory footprint also translate directly into computational complexity improvements, creating a dual benefit for resource-constrained environments.

1.3.2 How can optimization techniques help?

Optimization techniques tackle the environmental, social, and infrastructure problems that come with large-scale LLMs. When models run more efficiently, they need far less power for each query. Smaller models can actually run on edge devices and other energy-efficient hardware, cutting down on electricity usage substantially. This efficiency boost also extends the useful life of older devices: hardware that might otherwise be considered obsolete can suddenly run modern LLMs, which helps reduce electronic waste and makes better use of existing resources.

There's also the autonomy angle. Compact models that run locally allows users to be independent from centralized servers when using AI. People can run LLMs right on their own devices without any internet connection, which means no data gets sent to third parties and there's no risk of surveillance. This approach supports decentralization and opens up AI access to areas with poor connectivity or limited economic resources.

In other words, optimization is a pathway toward environmentally sustainable, privacy-respecting, and widely accessible AI.

1.4 Document structure

Will write this when the document is finished.

Chapter 2

Background and Related Work

Before explaining the details and implementation of the methodology used in this project, it is essential to provide an overview of the evolution of the inner workings of the Transformer architecture as well as Language Models in general. In addition, we will also discuss relevant compression techniques that have been developed in this context, and how they influenced this work. Finally, we will also shed some light on the target hardware, whose limitations have been a driving force behind the design choices made in this project.

2.1 Early language models

Before the Transformer architecture, language models were typically based on recurrent neural networks (RNNs). They consists in an extension of the Multi-Layer Perceptron (MLP) in which by integrating cycles, allowing for the network to take into account the previous inputs when making a prediction. For these reason, they can learn long-range dependencies in sequential data, such as text.

However, RNNs can be very much prone to the vanishing gradient problem ([14]), which makes it difficult for them to train in the long run, as during backpropagration the contribution of the gradients from earlier layer diminishes exponentially.

This problem motivated the introduction of the Long Short-Term Memory (LSTM) units, which uses gated cells in order to store more information about the input. In particular, there are three types gate cells:

- *Input gate*: controls how much of the new input should be added to the cell state.
- *Forget gate*: determines how much of the previous cell state should be retained.
- *Output gate*: decides how much of the cell state should be outputted to the next layer.

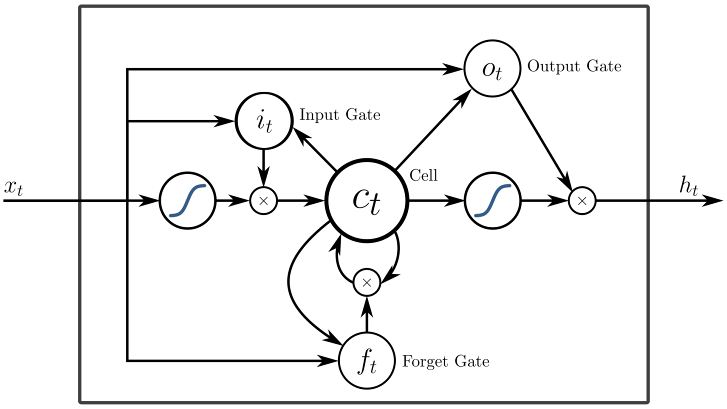


Figure 2.1: A visual representation of the LSTM architecture. The three gates (input, forget, and output) control the flow of information in and out of the cell state, allowing it to retain relevant information over long sequences.

2.2 The architecture of Transformers

The Transformer architecture, introduced by Vaswani et al. (2017) [TODO ADD REF HERE], represents a paradigm shift in sequence modeling, moving away from recurrent and convolutional approaches toward a purely attention-based mechanism. This innovation addressed fundamental limitations of previous architectures, particularly the sequential processing bottleneck that hindered parallelization during training. Attention Mechanism Evolution The development of attention mechanisms began with Bahdanau et al. (2014), who introduced additive attention to improve neural machine translation by allowing the decoder to focus on relevant parts of the input sequence. This was refined by Luong et al. (2015) with multiplicative attention, which offered computational advantages. However, the breakthrough came with Vaswani et al. (2017), who demonstrated that attention mechanisms alone, without recurrence or convolution, could achieve state-of-the-art results across multiple tasks. The core innovation lies in the scaled dot-product attention mechanism: where Q, K, and V represent queries, keys, and values respectively, and d_k is the dimension of the key vectors. This formulation enables efficient computation while capturing long-range dependencies without the sequential constraints of RNNs.

2.3 The structure of Large Language Models

2.4 A look at the target hardware

2.5 Relevant compression and optimization techniques

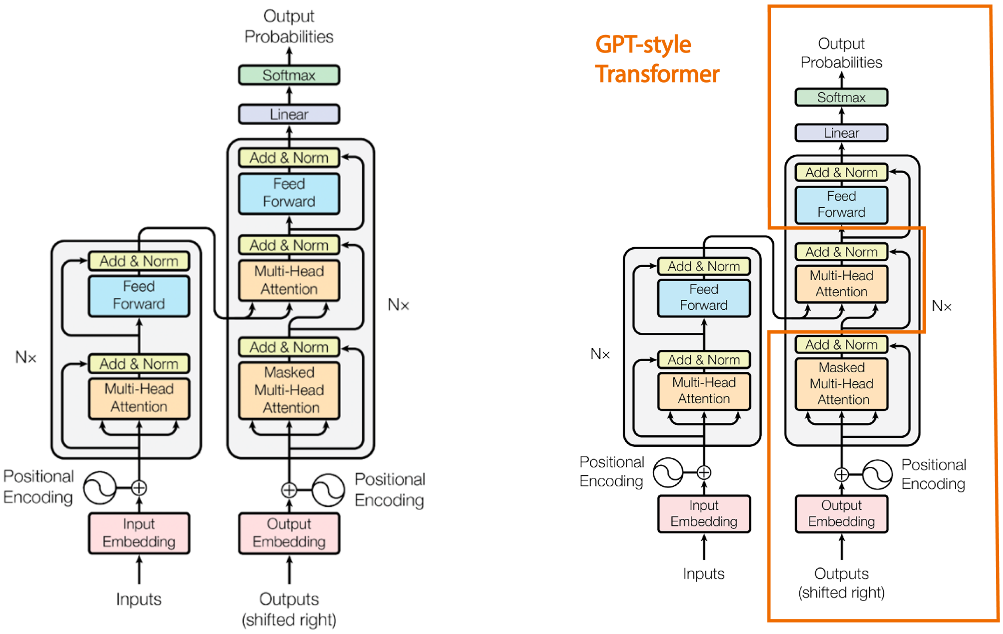


Figure 2.2: On the left, a standard Transformer architecture, which consists of an encoder and a decoder. On the right, we can observe the same architecture, while highlighting the only the layers used by a GPT-style model, which is a decoder-only transformer. The main difference is that the decoder does not attend to the encoder's output, allowing for auto-regressive generation.

Chapter 3

Methodology

Intro here

3.1 Preliminary research: Franken-LLaMA

3.2 Pruning methods

3.2.1 Depth-wise pruning

3.2.2 Width-wise pruning (WANDA-based)

3.2.3 LoRA

3.3 Quantization and EoRA

3.4 Evaluation criteria

Chapter 4

Implementation

nada

4.1 Implementation of the Pipeline

4.1.1 Depth-wise Pruning

4.1.2 Width-wise Pruning (WANDA-based)

ù

4.1.3 LoRA

4.1.4 Quantization and EoRA

4.2 Evaluation

How were the models evaluated?

Chapter 5

Experimental Results and Analysis

intro here

5.1 Preliminary observation of the results

Before examining the results based on In this section, we will present and comment on some examples of text generated by the

5.2 Results on TriviaQA

5.3 Results on WikiText

Chapter 6

Conclusion and Future Work

6.1 Future work

- speak about Distillation
- speak about experimenting with other quantization methods
- speak about adding support for other llms
- kV cache compression

6.2 Final remarks

The objective of this project was to research and implement a methodology for compressing LLaMA based LLMs,

Bibliography

[1] Mike Perkins, *Academic Integrity considerations of AI Large Language Models in the post-pandemic era: ChatGPT and beyond*, `https://www.researchgate.net/publication/368775737_Academic_integrity_considerations_of_AI_Large_Language_Models_in_the_post-pandemic_era_ChatGPT_and_beyond`.

[2] Daniel Mügge, *AI Is Threatening More Than Just Creative Jobs—It's Undermining Our Humanity*, `https://www.socialeurope.eu/ai-is-threatening-more-than-just-creative-jobs-its-undermining-our-humanity`.

[3] David Patterson, Joseph Gonzalez, Quoc Le, Chen Liang, Lluis-Miquel Munguia, Daniel Rothchild, David So, Maud Texier, Jeff Dean, *Carbon Emissions and Large Neural Network Training*, `https://arxiv.org/abs/2104.10350`.

[4] Thomas Spencer, Siddharth Singh, *What the data centre and AI boom could mean for the energy sector*, `https://www.iea.org/commentaries/what-the-data-centre-and-ai-boom-could-mean-for-the-energy-sector`

[5] Nidhal Jegham, Marwen Abdelatti, Lassad Elmoubarki, Abdeltawab Hendawi *How Hungry is AI? Benchmarking Energy, Water, and Carbon Footprint of LLM Inference* `https://arxiv.org/pdf/2505.09598v1`

[6] Google, *Google Environmental Report 2023*, `https://sustainability.google/reports/google-2023-environmental-report-executive-summary/`

[7] Pengfei Li, Jianyi Yang, Mohammad A. Islam, Shaolei Ren, *Making AI Less "Thirsty": Uncovering and Addressing the Secret Water Footprint of AI Models*, https://arxiv.org/abs/2304.03271.

[8] Arpan Suravi Prasad, Moritz Scherer, Francesco Conti, Davide Rossi, Alfio Di Mauro, Manuel Eggimann, Jorge Tómas Gómez, Ziyun Li, Syed Shakib Sarwar, Zhao Wang, Barbara De Salvo, Luca Benini *Siracusa: A 16 nm Heterogenous RISC-V SoC for Extended Reality with At-MRAM Neural Engine*, https://arxiv.org/abs/2312.14750.

[9] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, Illia Polosukhin, *Attention is All You Need*, https://arxiv.org/abs/1706.03762.

[10] Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova, *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*, https://arxiv.org/abs/1810.04805.

[11] Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, *Improving Language Understanding by Generative Pre-Training*, https://cdn.openai.com/research-covers/language-unsupervised/language_understanding_paper.pdf.

[12] OpenAI, *GPT-4 is OpenAI's most advanced system, producing safer and more useful responses*, https://openai.com/index/gpt-4/.

[13] Robin M. Schmidt, *Recurrent Neural Networks (RNNs): A gentle Introduction and Overview*, https://arxiv.org/pdf/1912.05911

[14] Chris Nicholson, *A Beginner's Guide to LSTMs and Recurrent Neural Networks*. https://skymind.ai/wiki/lstm.

[15] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, et al., *PyTorch: An Imperative Style, High-Performance Deep Learning Library*, https://arxiv.org/abs/1912.01703.

[16] Angelo Galavotti, *FRANKEN-LLAMA code repository*, https://github.com/AngeloGalav/franken-llama

[17] Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, Yejin Choi *HellaSwag: Can a Machine Really Finish Your Sentence?* https://arxiv.org/abs/1905.07830.

[18] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, et al., *LLaMA: Open and Efficient Foundation Language Models*, https://arxiv.org/abs/2302.13971.

[19] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, et al., *Llama 2: Open Foundation and Fine-Tuned Chat Models*, https://arxiv.org/abs/2307.09288.

[20] Llama Team, AI @ Meta, *The Llama 3 Herd of Models*, https://arxiv.org/abs/2407.21783.

[21] Hanjuan Huang, Hao-Jia Song, Hsing-Kuo Pao, *Large Language Model Pruning*, https://arxiv.org/abs/2406.00030.

[22] Bo-Kyeong Kim, Geonmin Kim, Tae-Ho Kim, Thibault Castells, Shinkook Choi, Junho Shin, Hyoung-Kyu Song, *Shortened LLaMA: Depth Pruning for Large Language Models with Comparison of Retraining Methods*, https://arxiv.org/abs/2402.02834.

[23] Mengzhou Xia, Tianyu Gao, Zhiyuan Zeng, Danqi Chen *Sheared LLaMA: Accelerating Language Model Pre-training via Structured Pruning*, https://arxiv.org/abs/2310.06694.

[24] ModelCloud, *GPTQModel*, https://github.com/ModelCloud/GPTQModel.

[25] Elias Frantar, Saleh Ashkboos, Torsten Hoefler, Dan Alistarh *GPTQ: Accurate Post-Training Quantization for Generative Pre-trained Transformers*, https://arxiv.org/abs/2210.17323

[26] Mingjie Sun, Zhuang Liu, Anna Bair, J. Zico Kolter, *A Simple and Effective Pruning Approach for Large Language Models*, https://arxiv.org/abs/2306.11695.

[27] Shih-Yang Liu, Maksim Khadkevich, Nai Chit Fung, Charbel Sakr, Chao-Han Huck Yang, Chien-Yi Wang, Saurav Muralidharan, Hongxu Yin, Kwang-Ting Cheng, et al., *EoRA: Fine-tuning-free Compensation for Compressed LLM with Eigenspace Low-Rank Approximation*, https://arxiv.org/abs/2410.21271.

[28] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, *LoRA: Low-Rank Adaptation of Large Language Models*, https://arxiv.org/abs/2106.09685.

[29] Mandar Joshi, Eunsol Choi, Daniel S. Weld, Luke Zettlemoyer, *TriviaQA: A Large Scale Distantly Supervised Challenge Dataset for Reading Comprehension*, https://arxiv.org/abs/1705.03551.

[30] Stephen Merity, Caiming Xiong, James Bradbury, Richard Socher, *Pointer Sentinel Mixture Models*, https://arxiv.org/abs/1609.07843.

[31] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, Peter J. Liu, *Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer*, https://arxiv.org/pdf/1910.10683.

[32] Chen Liang, Haoming Jiang, Zheng Li, Xianfeng Tang, Bin Yin, Tuo Zhao, *HomoDistil: Homotopic Task-Agnostic Distillation of Pre-trained Transformers*, https://arxiv.org/abs/2302.09632.

Acknowledgements

Pending