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

Presented by:
Angelo Galavotti

Co-supervisor:
Luca Bompani

Sessione I
Anno Accademico 2024/2025

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

Abstract

Pending

Contents

List of Figures

Elenco dei frammenti di codice

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 smartphone. When people contemplate Artificial Intelligence 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.

In this opening chapter we will briefly examine the evolution of LLMs and provide a more technical description of their capabilities. Furthermore, we'll investigate their considerable environmental footprint while highlighting the growing imperative for efficient, locally-deployable models that democratize access without depleting our planet's resources. In this context, we will introduce the *FRANKEN-LLAMA* project, which aims to create a more sustainable and efficient LLM. The future of AI depends not just on what these models can do, but how sustainably they can do it.

1.1 A brief overview of the evolution of LLMs

Fundamentally, at the core of LLMs lies the concept of *transformers*, a neural network architecture introduced in 2017 by Vaswani et al. in their famous paper "Attention is All You Need". Initially designed for translation tasks, transformers have since been adapted for a wide range of natural language processing (NLP) tasks such as summarization and sentiment analysis. A famous example of a transformer model is *BERT* (Bidirectional Encoder Representations from Transformers), which has been widely used for various NLP tasks. BERT's architecture allows it to understand the context of words in a sentence by considering both the left and right context simultaneously, making it particularly effective for classification tasks such as entity named recognition.

However, the biggest impact of transformers has been in the realm of text generation, where they can produce consistent and contextually relevant text based on a given prompt. This is achieved through a mechanism called *self-attention*, which allows the model to weigh the importance of different words in a sentence when generating text. By using self-attention, transformers can capture long-range dependencies and relationships between words. A more technical overview of the transformer architecture is provided in Section 2.2. The GPT (Generative Pre-trained Transformer) series, developed by OpenAI, is a prime example of this capability, with GPT-4 being the most recent version. These models are pre-trained on vast amounts of text data and its performance has become a new benchmark for other models in the field.

1.2 The darker side of LLMs and scope of this project

1.3 Document structure

AAAAAAAA AM I GOING CRAZY?

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 ([3]), 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.

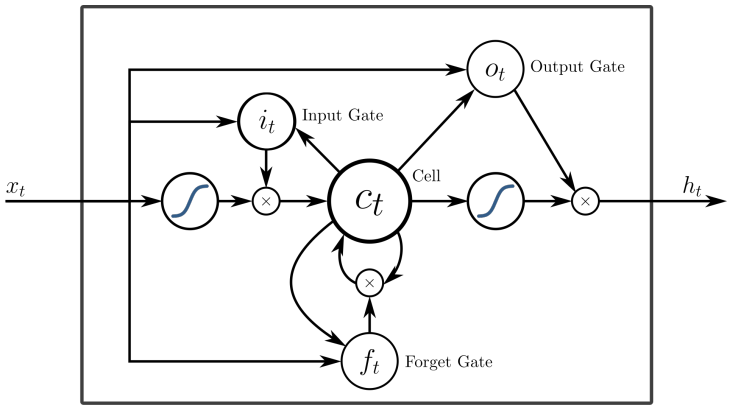

Figure 2.1: A visual representation of the LSTM architecture. The three gates (input, forget, and output) control the flow of information in and out of the cell state, allowing it to retain relevant information over long sequences.

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 Relevant compression techniques

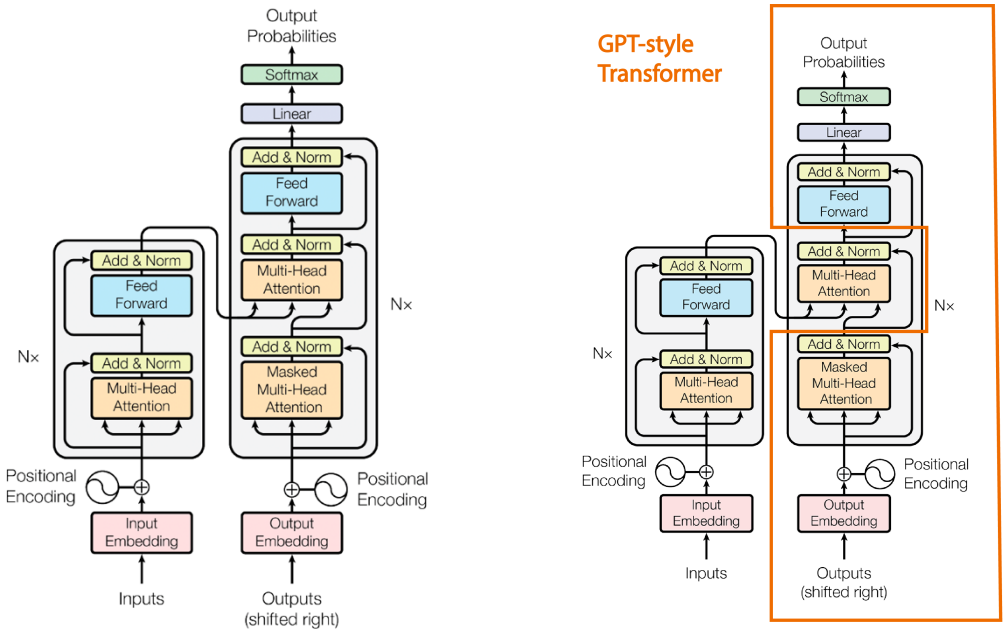

Figure 2.2: On the left, a standard Transformer architecture, which consists of an encoder and a decoder. On the right, we can observe the same architecture, while highlighting the only the layers used by a GPT-style model, which is a decoder-only transformer. The main difference is that the decoder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

Chapter 3

Methodology

nada

Chapter 4

Implementation

nada

Chapter 5

Experimental Results and Analysis

nada

Chapter 6

Conclusion and Future Work

6.1 Future work

ok

6.2 Final remarks

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

Bibliography

[1] 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`.

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

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

[4] 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`.

[5] 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`.

[6] 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`.

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

[8] 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.

[9] 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.

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

[11] 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.

[12] 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.

[13] 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.

[14] , 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.

[15] 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.

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

[17] 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`.

Acknowledgements

Ringrazio il professor Ozalp Babaoglu per la sua disponibilità come relatore, nonostante fosse prossimo alla pensione. Sono molto grato di essere uno dei suoi ultimi tesisti della sua carriera. I suoi contributi nell'ambito dei sistemi operativi sono inestimabili.

Ringrazio il professor Francesco Giacomini, per avermi dato l'accesso a una delle esperienze formative più importanti della mia vita. Lo ringrazio inoltre per avermi assistito attentamente nello sviluppo del progetto e nella scrittura di questa tesi. Grazie anche per la tua pazienza nei miei confronti.

Ringrazio tutto il personale dell'INFN CNAF per aver reso l'esperienza ancora più gradevole e per avermi fatto visitare il centro di calcolo, trattandomi sempre come se fossi un loro collega.

Un ringraziamento speciale va alla mia famiglia, per avermi sempre dato la spinta di andare avanti e per credere in me, senza negarmi mai nulla.

Ringrazio tutti gli amici che ho conosciuto nel corso durante questi tre anni: Leon, Drif, Giaco, Baldo, Adriano, Donnoh, Vir, Matteo, Pino, Denis, Samuele, Alice e tanti altri. Non avrei potuto chiedere dei compagni migliori.

Infine, ringrazio Leti, per essermi stata accanto nei momenti più bui e avermi dato tutto l'affetto che esiste in questo mondo, anche quando non me lo meritavo.