Otimização de Inferência em LLMs na CPU: Análise do Cenário Atual

Pedro Cattai[1], **Alexandro Baldassin**[1], **Allberson Dantas**[2]

[1]Instituto de Geociências e Ciências Exatas – Universidade Estadual Paulista (Unesp)
Rio Claro – SP – Brasil

[2]Universidade da Integração Internacional da Lusofonia Afro-Brasileira (UNILAB)
Redenção – CE – Brasil

{pedro.cattai,alexandro.baldassin}@unesp.br, allberson@unilab.edu.br

Abstract. *Advancements in Artificial Intelligence (AI), particularly in Large Language Models (LLMs), have highlighted the challenges of efficient inference in resource-constrained environments. While GPUs are the preferred hardware for LLM inference, their limited accessibility motivates the exploration of CPU-based alternatives. This work presents a study of optimizations for LLM inference on CPUs, focusing on memory manipulation techniques. By addressing this bottleneck, we propose to enhance inference efficiency without high-end hardware. This paper outlines the current progress of the research, and the proposed methodology sets the ground for future experiments, with the potential to broaden the accessibility of LLMs.*

Resumo. *Os avanços em Inteligência Artificial (IA), especialmente em Large Language Models (LLMs), evidenciaram os desafios da inferência eficiente em ambientes com recursos limitados. Embora GPUs sejam o hardware preferencial para inferência em LLMs, sua acessibilidade restrita motiva a exploração de alternativas baseadas em CPU. Este trabalho apresenta um estudo de otimizações para inferência em LLMs em CPUs, com foco em técnicas de manipulação de memória. Ao abordar este gargalo, propomos melhorar a eficiência da inferência sem hardware avançado. Este artigo descreve o andamento atual da pesquisa e a metodologia proposta estabelece bases para futuros experimentos, com potencial para ampliar a acessibilidade de LLMs.*

1. Introdução

Nos últimos tempos, ferramentas de Inteligência Artificial (IA) têm alcançado níveis de popularidade sem precedentes. Dentre as aplicações que usam tais ferramentas, aquelas que possuem um caráter generativo conversacional obtiveram um destaque maior. Estes produtos têm como base tecnológica os *Large Language Models (LLMs)*, que são modelos de IA treinados em grandes quantidades de dados utilizando a tecnologia dos *transformers* [Shanahan 2024].

Os LLMs têm uma capacidade impressionante de processar linguagem natural, identificando contextos e produzindo respostas à altura. Para atingir estes resultados, é necessário o uso elevado de recursos computacionais para a fase de treinamento do modelo, na qual o processamento dos dados e o cálculo dos pesos da rede neural consomem

muita energia e tempo. Devido a isso, o treinamento de tais modelos não é comumente realizado em trabalhos sem investimentos consideravelmente grandes [Bender et al. 2021].

O uso dos LLMs é possibilitado, entretanto, pela disponibilização de modelos *open source*. Este meio permite o uso livre de tais modelos para execuções de inferência locais, abrindo espaço para estudos de caso, avaliações de performance e adaptações de modelos [Xu et al. 2023]. Outro problema surge, porém, na prática da inferência. Mesmo demandando menos recursos computacionais que o treinamento, nem todo dispositivo é capaz de executar a inferência de maneira satisfatória. Isso ocorre ocorre devido a diferenças significativas entre os modelos disponíveis, como tamanho e complexidade, que podem elevar os requisitos de hardware [Samsi et al. 2023]. Portanto, a importância da GPU no processo de inferência é inegável, pois sua capacidade de processamento paralelo a torna ideal para tarefas intensivas em cálculos. No entanto, nem todos os usuários têm acesso a GPUs poderosas, levando a um cenário onde a inferência em CPUs se torna uma alternativa necessária, ainda que desafiadora [He et al. 2024].

Diante deste contexto, o propósito do nosso trabalho em andamento é realizar otimizações na execução da inferência de LLMs em contextos dependentes da CPU. Estas otimizações serão realizadas principalmente através de manipulações de memória, utilizando técnicas de compressão e adaptação de modelos e reorganizações de seus usos da memória. O intuito é produzir testes que comparem a eficiência entre técnicas diferentes de lidar com o gargalo de memória, além de fazer comparações de performance em arquiteturas alternativas, como memórias persistentes.

2. Fundamentação Teórica

A inferência de *Large Language Models* (LLMs) em CPUs apresenta desafios distintos em relação à execução em GPUs, sendo o principal gargalo de desempenho relacionado à capacidade computacional (*compute-bound*) em vez de limitações de memória (*memory-bound*) [Na et al. 2024]. Este fenômeno ocorre porque as operações matriciais intensivas que compõem os LLMs exigem uma vazão computacional maior do que as CPUs convencionais são capazes de fazer, visto que não possuem a infraestrutura especializada nestas operações como as GPUs.

Os LLMs baseados em arquitetura *decoder-only* operam de forma a gerar um *token* por vez. Durante esse processo, os componentes do modelo que se destacam são as camadas *feed-forward*, que realizam as transformações nos *embeddings*, e o mecanismo de atenção que mantém um *Key-Value (KV) cache* contendo os estados de atenção de todos os tokens anteriores. O *KV cache* permite que o modelo acesse eficientemente o contexto histórico sem recalcular todos os estados a cada novo *token*, porém seu tamanho cresce linearmente com o comprimento da sequência, consumindo rapidamente a memória disponível [He et al. 2024]. No contexto da GPU, isso cria problemas de sobrecarga devido à movimentação de dados, ao demandar o *offloading* para o domínio da CPU. Sem esta limitação da VRAM, o problema passa a ser ligado à capacidade computacional da CPU.

A partir disso, surgem várias possibilidades de otimização voltadas à redução da carga computacional na CPU. Soluções que utilizam quantização trazem novos desafios, como a perda de precisão ou a introdução de sobrecarga para realizar esta conversão [Shen et al. 2023]. Além disso, as técnicas de *pruning* trazem benefícios significativos na redução da escala do processamento, porém com *trade-offs* de complexidade e

acurácia [Ma et al. 2023]. É neste contexto que este trabalho se insere, buscando combinar otimizações para tornar a inferência de LLMs em CPU mais eficiente.

3. Estágio Atual do Trabalho

Considerando a ideia de tornar a inferência de LLMs mais eficiente, nós realizamos uma revisão sistemática da literatura com foco neste assunto. Na condução desta revisão, encontramos trabalhos extremamente diversos, cada um aplicando uma técnica diferente para alcançar suas otimizações. A ampla variação, além do caráter recente desta área de estudos, faz com que em muitos casos os trabalhos deixem lacunas abertas para pesquisas futuras. Dentre estas oportunidades, percebemos que não há literatura extensiva sobre a otimização de LLMs utilizando memória persistente, como a Intel® Optane™ Persistent Memory.

Com isso, demos outro passo na definição do escopo do trabalho, para o contexto da CPU, como mencionado anteriormente. Deste modo, não há mais a restrição baseada no tamanho da VRAM, permitindo a experimentação com contextos maiores, além de mitigar o *overhead* criado pela incongruência da velocidade entre GPU e CPU [Liu et al. 2024]. Uma vez decidido este foco, buscamos trabalhos que lidavam especificamente com isso.

Dentre os poucos artigos que tratam de inferência de LLMs em CPU, alguns se destacaram. Em seu trabalho, [Park and Egger 2024] desenvolveram uma técnica para maximizar a vazão dos modelos usando computações da CPU. Para isso, modificaram o pipeline de operações do processamento do LLM para que o tempo ocioso do CPU e da GPU se minimizem. Por sua vez, [Na et al. 2024] apresentam uma análise de performance das CPUs de última geração para inferência de LLMs. Neste artigo, os autores destacam a vantagem da eliminação da necessidade de *offloading* dos dados, já que o domínio da CPU é suficientemente grande. Além disso, mostram que as novas CPUs com aceleradores de multiplicação de matrizes podem trazer benefícios para processamentos deste tipo.

Outro artigo que se mostrou relevante para nossa revisão foi [He et al. 2024], no qual os autores desenvolveram uma solução usando apenas CPU. Nesta solução, o *KV cache* é reduzido através de quantizações, porém com valores de escala específicos para cada cabeça de atenção, mantendo boa parte da precisão. Além disso, é utilizado um esquema de inferência distribuída, sinalizando o potencial deste conceito para a área.

A partir da revisão sistemática realizada previamente, além da análise dos trabalhos atuais deste nicho, nós pretendemos produzir nossa própria contribuição para a otimização de inferência de LLMs em CPU. Para isso, faremos combinaçõeos de otimizações já estudadas e criaremos comparações de performance entre diferentes cenários de otimização e arquiteturas de memória.

4. Conclusão

Neste texto, apresentamos uma discussão inicial de pesquisa para otimizar a inferência de LLMs em CPUs, destacando a importância de técnicas de manipulação de memória como estratégia para contornar limitações de *hardware*. A revisão da literatura revelou oportunidades significativas, especialmente na exploração de memória persistente, um

campo ainda pouco estudado no contexto de LLMs. O estágio atual do trabalho consistiu na definição do escopo e na identificação de abordagens promissoras, como adaptação de modelos e redução de transferência de dados.

Como próximos passos, planeja-se a implementação prática das técnicas propostas e sua avaliação em diferentes configurações de hardware. Esta pesquisa tem o potencial de contribuir para a democratização do uso de LLMs, tornando-os mais acessíveis em ambientes com restrições computacionais. Os resultados futuros serão fundamentais para validar as hipóteses e consolidar as contribuições deste estudo.

5. Agradecimento

O presente trabalho foi realizado com apoio da Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP), Brasil. Processo nº 2024/02372-7

Referências

Bender, E. M., Gebru, T., McMillan-Major, A., and Shmitchell, S. (2021). On the dangers of stochastic parrots: Can language models be too big?. In *Proceedings of the 2021 ACM conference on fairness, accountability, and transparency*, pages 610–623.

He, P., Zhou, S., Huang, W., Li, C., Wang, D., Guo, B., Meng, C., Gui, S., Yu, W., and Xie, Y. (2024). Inference performance optimization for large language models on cpus.

Liu, N. F., Lin, K., Hewitt, J., Paranjape, A., Bevilacqua, M., Petroni, F., and Liang, P. (2024). Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 12:157–173.

Ma, X., Fang, G., and Wang, X. (2023). Llm-pruner: On the structural pruning of large language models. *Advances in neural information processing systems*, 36:21702–21720.

Na, S., Jeong, G., Ahn, B. H., Young, J., Krishna, T., and Kim, H. (2024). Understanding performance implications of LLM inference on cpus. In *2024 IEEE International Symposium on Workload Characterization (IISWC)*, pages 169–180.

Park, D. and Egger, B. (2024). Improving throughput-oriented LLM inference with cpu computations. In *Proceedings of the 2024 International Conference on Parallel Architectures and Compilation Techniques*, PACT '24, page 233–245, New York, NY, USA. Association for Computing Machinery.

Samsi, S., Zhao, D., McDonald, J., Li, B., Michaleas, A., Jones, M., Bergeron, W., Kepner, J., Tiwari, D., and Gadepally, V. (2023). From words to watts: Benchmarking the energy costs of large language model inference. In *2023 IEEE High Performance Extreme Computing Conference (HPEC)*, pages 1–9. IEEE.

Shanahan, M. (2024). Talking about large language models. *Communications of the ACM*, 67(2):68–79.

Shen, H., Chang, H., Dong, B., Luo, Y., and Meng, H. (2023). Efficient llm inference on cpus.

Xu, Q., Hong, F., Li, B., Hu, C., Chen, Z., and Zhang, J. (2023). On the tool manipulation capability of open-source large language models.