PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO GRANDE DO SUL
ESCOLA POLITÉCNICA
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

TREINAMENTO OTIMIZADO, AVALIAÇÃO E INSTRUMENTAÇÃO DE LLMS: UMA ABORDAGEM INTEGRADA PARA AMBIENTES COM RECURSOS COMPUTACIONAIS LIMITADOS

Lucas Treviso Bandeira
Vermon João de Aguiar Neto

Proposta de Trabalho de Conclusão apresentada como requisito parcial à obtenção do grau de Bacharel em Ciência da Computação na Pontifícia Universidade Católica do Rio Grande do Sul.

Orientador(a): Cesar Augusto Fonticielha De Rose

Porto Alegre
2025

1. INTRODUÇÃO

O rápido avanço da inteligência artificial tem transformado diversas áreas do conhecimento, impulsionado especialmente pelo desempenho excepcional dos modelos de linguagem de larga escala (LLMs) em uma variedade de tarefas – como evidenciado pelos trabalhos de Brown et al. [1] e Touvron et al. [2], que demonstram a capacidade desses modelos de realizar inferência eficaz mesmo com poucos exemplos de treinamento. No entanto, a implementação desses sistemas enfrenta desafios significativos, decorrentes do elevado número de parâmetros e da necessidade de conjuntos de dados extensos, o que se traduz em alta demanda por memória e processamento. Adicionalmente, a execução e o ajuste fino (fine-tuning) desses modelos costumam requerer infraestruturas robustas, equipadas com GPUs de alto desempenho, limitando seu acesso a usuários e aplicações que dispõem apenas de hardware modesto.

Em paralelo ao desenvolvimento dos chamados Foundation Models, a comunidade científica tem buscado métodos eficientes para adaptar esses modelos a tarefas específicas, de forma a ajustar suas estruturas linguísticas, padrões e relações conceituais sem incorrer nos altos custos computacionais de um ajuste completo. Nesse contexto, técnicas como a quantização [3] e o Low-Rank Adaptation (LoRA) [4] emergem como estratégias promissoras, pois reduzem os requisitos computacionais e possibilitam a execução de LLMs em hardware convencional.

Apesar dos avanços recentes, persiste uma lacuna na literatura quanto à aplicação prática dessas técnicas em ambientes com recursos computacionais limitados. A complexidade inerente à implantação de sistemas de aprendizado de máquina em larga escala revela desafios técnicos e operacionais – como a necessidade de configurações específicas, o gerenciamento de feedbacks e a dependência de dados externos – que podem resultar em custos elevados a longo prazo [5]. Tal cenário evidencia a necessidade de uma abordagem integrada que combine ajuste fino e monitoramento de desempenho, facilitando a adaptação e a operação de LLMs com a utilização ótima dos recursos em hardware restrito.

Além das limitações de infraestrutura, outra motivação relevante para a realização de treinamento e ajuste fino localmente diz respeito à segurança e à confidencialidade dos dados utilizados. Em muitos cenários, como aplicações médicas, jurídicas ou corporativas, os dados são sensíveis e não podem ser compartilhados com serviços externos ou armazenados em nuvem sem comprometer requisitos éticos ou

regulatórios. A possibilidade de executar o treinamento localmente, mesmo em máquinas modestas, representa uma vantagem significativa nesse contexto, pois permite preservar a privacidade das informações, reduzir a exposição a riscos de vazamento e manter o controle total sobre o fluxo de dados durante o processo de ajuste dos modelos.

Diante disso, o presente projeto propõe o desenvolvimento de uma ferramenta integrada que una técnicas de fine-tuning a módulos de infraestrutura e monitoramento de desempenho, viabilizando a execução local de LLMs em ambientes com recursos limitados. Especificamente, o objetivo é adaptar estratégias de ajuste fino para modelos compactos e implementar módulos de instrumentação capazes de monitorar, em tempo real, o uso da infraestrutura, permitindo a identificação precoce de regressões no desempenho e gargalos operacionais [6].

Este estudo apresenta, inicialmente, uma revisão abrangente da literatura sobre estratégias de otimização para o ajuste fino de LLMs em ambientes restritos e a instrumentação do processo (Seção 2). Em seguida, detalha a proposta metodológica para o desenvolvimento da ferramenta integrada, que visa não apenas aprimorar o desempenho dos modelos, mas também contribuir para a democratização do acesso à tecnologia em contextos de hardware limitado (Seção 3). Após, o trabalho apresenta um cronograma das etapas previstas, abrangendo desde a revisão bibliográfica até o desenvolvimento prático da solução (Seção 4). Por fim, são discutidos os recursos a serem utilizados durante o projeto para desenvolvimento e teste da proposta (Seção 5).

2. FUNDAMENTAÇÃO TEÓRICA

Os modelos de linguagem de larga escala (LLMs) têm avançado rapidamente e transformado diversas áreas. Pesquisas recentes destacam suas capacidades excepcionais de compreensão, geração e raciocínio de linguagem natural, o que levou a avanços em diversas tarefas como resumo de documentos [7], reformulação de texto [8] e resposta a perguntas [9]. Esses progressos tiveram forte impacto, viabilizando ferramentas amplamente usadas (por exemplo, assistentes de código como o Copilot e agentes conversacionais como o ChatGPT) e gerando um efeito transformador.

Pesquisas recentes têm mapeado as diversas aplicações dos LLMs – como evidenciado pelo trabalho de Kumar, P [10] – que abrangem a geração de texto, imagens, assistência para codificação e aplicações em bioinformática. Dessa forma, a literatura atual confirma que esses modelos alcançaram resultados de ponta em múltiplos domínios, porém, apesar de seu sucesso, a execução e o ajuste fino dos LLMs apresentam desafios computacionais significativos.

Modelos com dezenas ou centenas de bilhões de parâmetros demandam uma quantidade substancial de memória de GPU para a inferência, além de requererem um volume massivo de dados para o pré-treinamento (RAM e VRAM) [11]. Nesse cenário, destaca-se o exemplo citado por Edward J. Hu et al. [3], que evidencia o custo computacional proibitivo do fine-tuning completo, especialmente em modelos de escala como o GPT-3, com seus 175 bilhões de parâmetros, que requerem grandes datacenters equipados com inúmeras GPUs de altíssima capacidade.

Mesmo em cenários onde o ajuste fino completo dos LLMs é dispensável, a inferência desses modelos em hardware convencional continua sendo problemática. De acordo com Dettmers et al. [11], é comum que os LLMs não caibam na memória das GPUs, o que dificulta sua utilização por pesquisadores, usuários comuns e aplicações em geral. Por outro lado, Zheng et al. [12] propõem uma revisão dos LLMs na borda (Edge LLMs) e destacam desafios relevantes para pesquisas que envolvem hardware limitado, como o aumento da latência devido ao baixo throughput em CPUs ou GPUs modestas. Assim, a literatura atual evidencia que as elevadas exigências de GPU, memória e dados constituem barreiras significativas para a implementação local de LLMs.

Diante desses desafios, diversas técnicas otimizadas para a adaptação de modelos emergiram como soluções viáveis, ganhando destaque na literatura. Entre as principais estratégias exploradas para o presente projeto, destacam-se a quantização de pesos (redução da precisão numérica) e a Low-Rank Adaptation (LoRA). A quantização, por sua vez, tem como objetivo diminuir o tamanho do modelo e a carga computacional necessária sem comprometer significativamente seu desempenho, possibilitando a execução de modelos de larga escala em hardware menos especializado. Um exemplo seria o método LLM.int8, que consegue rodar um modelo de 175B parâmetros em 8 bits sem perda de acurácia, cortando pela metade a memória de inferência necessária [11].

A outra estratégia explorada para otimização de LLMs é o método LoRA, proposto por Hu et al. [4], que congela os pesos originais do LLM e treina apenas matrizes de menor dimensão (low-rank) inseridas em cada camada, reduzindo significativamente o número de parâmetros ajustados em comparação com um ajuste completo.

Ambas as técnicas de otimização podem ser combinadas para maximizar a adaptação dos LLMs a ambientes com hardware convencional, conforme explorado por Dettmers et al. [13] ao apresentar o método QLoRA. Esse método híbrido utiliza quantização em 4 bits e LoRA simultaneamente, possibilitando o fine-tuning de um modelo com 65B parâmetros em uma única GPU de 48GB sem perda de desempenho em relação à precisão original do modelo. Assim, conseguimos inferir que métodos de

quantização e Low-Rank Adaptation são abordagens que podem viabilizar a adaptação e inferência de LLMs em cenários com recursos limitados.

Além das estratégias de otimização apresentadas, a utilização de ferramentas de monitoramento é fundamental para o sucesso no ajuste e inferência de LLMs em ambientes com recursos limitados. Nesse contexto, a ferramenta MLflow destaca-se como uma plataforma open-source robusta para o gerenciamento do ciclo de vida de experimentos em machine learning, permitindo o rastreamento de hiperparâmetros, métricas de desempenho e o versionamento de modelos. Assim, é uma ferramenta que pode ser explorada no contexto de hardware convencional para contribuir na identificação de gargalos e regressões e melhorar a reprodutibilidade dos experimentos.

Diante do exposto, embora os modelos de linguagem de larga escala tenham transformado diversas áreas e demonstrado capacidades excepcionais, sua aplicação em ambientes com recursos computacionais limitados continua sendo um grande desafio. Técnicas de otimização emergem como estratégias promissoras para reduzir os requisitos computacionais sem comprometer a performance dos modelos e, paralelamente, a integração de ferramentas de monitoramento, como o MLflow, torna-se indispensável para rastrear experimentos e gerenciar o ciclo de vida dos modelos.

Assim, o resultado da análise exploratória da literatura converge para a necessidade de uma abordagem integrada que combine essas técnicas de otimização com soluções robustas de instrumentação, reforçando a viabilidade de se adaptar LLMs para execução local em hardware convencional. De forma mais específica, deve-se buscar viabilizar a instrumentação e o monitoramento local com baixo overhead, ou seja, com comprometimento mínimo dos recursos para não gerar sobrecarga durante tarefas de ajuste parcial e inferência.

3. PROPOSTA

Este projeto busca desenvolver uma ferramenta integrada que combine instrumentação, otimização e ajuste fino de modelos de linguagem (LLMs) para ambientes com recursos computacionais restritos – caracterizados pelo uso exclusivo de CPUs, GPUs de consumo doméstico e memória RAM limitada. Ao democratizar o acesso às tecnologias de inteligência artificial, a proposta fundamenta-se na aplicação de técnicas avançadas, como o ajuste fino via LoRA, a modelos compactos exemplificados pelas versões "tiny" ou "baby" do Llama. Dessa forma, a solução não só viabiliza a execução local de LLMs em hardware modesto, mas também garante a privacidade e segurança dos dados processados, contribuindo para uma adoção mais ampla e sustentável dessas tecnologias.

Para alcançar esses objetivos, o projeto integrará práticas de MLOps com estratégias de monitoramento e instrumentação em tempo real, permitindo a análise detalhada da utilização de CPU, gerenciamento da memória e identificação de eventuais gargalos computacionais. Técnicas de paralelização, quantização e balanceamento de carga serão incorporadas para maximizar o desempenho mesmo em plataformas desprovidas de GPUs especializadas para tarefas de machine learning.

Além disso, a plataforma oferecerá suporte para visualização e ajuste fino dos modelos, possibilitando aos usuários realizarem análises que inferirão a melhor configuração para suas máquinas, de acordo com as especificidades do hardware disponível. Com foco em sistemas com restrições computacionais, o ambiente não só permite a implementação eficiente de modelos compactos, mas também fornece insights estratégicos que otimizam a alocação de recursos.

4. CRONOGRAMA

O cronograma proposto detalha as principais atividades a serem realizadas ao longo de nove meses em 2025, conforme ilustrado na Tabela 1. As etapas foram organizadas de forma sequencial, garantindo a conclusão de cada fase dentro dos prazos estabelecidos e atendendo às dependências entre as tarefas, o que viabiliza o desenvolvimento integral da metodologia e a análise aprofundada dos resultados.

Tabela 1 – Cronograma

Atividades	Mar	Abr	Mai	Jun	Jul	Ago	Set	Out	Nov	Dez
Revisão da Literatura										
Escrita da proposta										
Estudo de ferramentas										
Desenvolvimento										
Análise de resultados										
Produção textual										

5. RECURSOS A SEREM UTILIZADOS

Para o desenvolvimento da proposta, serão utilizadas bibliotecas consolidadas do ecossistema Python voltadas ao treinamento, ajuste e instrumentação de modelos de linguagem. Essas ferramentas serão selecionadas com base em sua compatibilidade com técnicas de otimização leve e sua facilidade de integração em diferentes ambientes. Em relação à infraestrutura, os testes e implementações serão realizados em computadores

com e sem GPU dedicada, sendo os dispositivos sem GPU (apenas CPUs tradicionais) caracterizados por configurações modestas quando comparados a servidores ou máquinas de alto desempenho.

A escolha de ambientes com diversas configurações de hardware busca simular diferentes cenários de uso, refletindo tanto contextos com recursos limitados quanto ambientes um pouco mais robustos, possibilitando a avaliação da versatilidade e da eficiência da ferramenta desenvolvida. Também, para fins de comparação e análise de desempenho, poderá ser utilizado o Laboratório de Alto Desempenho (LAD) da PUCRS, permitindo observar diferenças na execução entre ambientes restritos e sistemas com maior capacidade computacional.

6. CONSIDERAÇÕES FINAIS

Ao propor uma ferramenta capaz de viabilizar o uso local de modelos de linguagem em ambientes com recursos computacionais limitados, este trabalho busca ampliar o acesso às tecnologias de inteligência artificial e fomentar a autonomia no desenvolvimento de soluções baseadas em LLMs. A adaptação de estratégias de ajuste fino e o monitoramento em tempo real da infraestrutura contribui para tornar esses sistemas mais acessíveis, sustentáveis e alinhados às restrições técnicas enfrentadas por muitos usuários e instituições. A implementação local também reforça aspectos importantes como a privacidade e o controle sobre os dados processados, promovendo uma IA mais segura e ética. Além disso, serão conduzidas análises aprofundadas sobre as ferramentas existentes que viabilizam o ajuste e a execução de LLMs em hardware limitado, bem como estudos para avaliar a eficácia da solução proposta em cenários reais de uso, permitindo a identificação de seus benefícios, limitações e potenciais melhorias.

REFERÊNCIAS

[1] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open Foundation and Fine-Tuned Chat Models, 2023.

[2] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language Models are Few-Shot Learners, 2020.

[3] Benoit Jacob, Skirmantas Kligys, Bo Chen, Menglong Zhu, Matthew Tang, Andrew Howard, Hartwig Adam, and Dmitry Kalenichenko. Quantization and Training of Neural Networks for Efficient Integer-Arithmetic-Only Inference, 2017.

[4] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-Rank Adaptation of Large Language Models, 2021.

[5] D. Sculley, Gary Holt, Daniel Golovin, Eugene Davydov, Todd Phillips, Dietmar Ebner, Vinay Chaudhary, Michael Young, Jean-François Crespo, and Dan Dennison. Hidden Technical Debt in Machine Learning Systems, 2018.

[6] Eric Breck, Shanqing Cai, Eric Nielsen, Michael Salib, and D. Sculley. The ML Test Score: A Rubric for ML Production Readiness and Technical Debt Reduction, 2017.

[7] Huan Yee Koh, Jiaxin Ju, Ming Liu, and Shirui Pan. An Empirical Survey on Long Document Summarization: Datasets, Models, and Metrics, 2022.

[8] Huiyuan Lai and Malvina Nissim. A Survey on Automatic Generation of Figurative Language: From Rule-Based Systems to Large Language Models, 2024.

[9] Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. Pretrain, Prompt, and Predict: A Systematic Survey of Prompting Methods in Natural Language Processing, 2023.

[10] Kumar, P. Large language models (LLMs): survey, technical frameworks, and future challenges, 2024.

[11] Tim Dettmers, Mike Lewis, Younes Belkada, and Luke Zettlemoyer. LLM.int8(): 8-bit Matrix Multiplication for Transformers at Scale, 2022.

[12] Yue Zheng, Yuhao Chen, Bin Qian, Xiufang Shi, Yuanchao Shu, and Jiming Chen. A Review on Edge Large Language Models: Design, Execution, and Applications, 2025.

[13] Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. QLoRA: Efficient Finetuning of Quantized LLMs, 2023.