MathLLaMA: A Specialized Language Model for Mathematical Reasoning and Problem-Solving

Yu Zhang[§], **Changsong Lei**[§]
[§]Tsinghua University, China;
{yu-zhang23,leics23}@mails.tsinghua.edu.cn;

Abstract

As recent advancements in AI and natural language processing have made it possible for language models to understand and generate human-like text, the field of mathematical language processing remains uniquely challenging. Mathematical texts often involve intricate symbolic notations, specialized terminology, and formal structures, necessitating tailored approaches for training models to handle such content effectively. Addressing these challenges, MathLLaMA leverages the LLaMA-Factory framework to create a model optimized for a variety of mathematical tasks, ranging from algebraic manipulation and calculus problem-solving to higher-level areas like discrete mathematics and number theory. This paper introduces MathLLaMA, a fine-tuned version of the LLaMA model, designed explicitly for mathematical problem-solving and reasoning. MathLLaMA leverages the LLaMA-Factory framework, which provides a comprehensive toolkit for training and fine-tuning LLMs. The primary objective of MathLLaMA is to extend the capabilities of LLaMA for use in mathematical domains by equipping it with the ability to understand formal mathematical language, reason through multi-step solutions, and generate accurate mathematical expressions. Our approach involves fine-tuning the model on a diverse set of mathematical datasets and using specialized techniques to address the unique challenges posed by mathematical texts.

1 Introduction

Mathematical reasoning and problem-solving are fundamental components of human intelligence and have broad applications in fields ranging from physics and engineering to economics and computer science [1]. Despite recent advancements in natural language processing (NLP), enabling large language models (LLMs) to understand and generate mathematical content remains a significant challenge. Mathematical language is distinct from natural language due to its symbolic notations, formalized expressions, and specialized terminology, all of which require an approach tailored to mathematical texts [2]. The successful application of language models in this domain has the potential to revolutionize mathematics education, automated theorem proving, and computational research.

Recent progress in LLMs, particularly architectures like GPT-3, T5, and LLaMA, has shown the capacity of these models to understand a wide range of natural language tasks [3]. However, their ability to perform well on mathematical tasks remains limited due to the distinct nature of mathematical language. LLMs often struggle with symbolic manipulation, equation solving, and step-by-step reasoning required to solve complex mathematical problems [4]. Addressing these limitations necessitates the development of specialized models that can handle mathematical language with greater accuracy and reliability.

To fine-tune MathLLaMA, we employed a multi-stage approach designed to maximize the model's performance on mathematical language. Initially, a curriculum learning strategy was applied, gradu-

37th Conference on Neural Information Processing Systems (NeurIPS 2023).

ally increasing the complexity of training examples—from basic arithmetic to advanced mathematical proofs [5]. Data augmentation techniques were utilized to expand the dataset, including generating varied problem-solving approaches and incorporating different representations of mathematical expressions. Prompt engineering was applied to guide the model's training on solving problems, explaining solutions, and understanding specialized mathematical queries.

MathLLaMA will be extensively fine-tuned on the latest high-quality mathematical datasets, including a combination of academic textbooks, research papers, structured problem sets, and crowdsourced solutions from platforms like MathOverflow and Stack Exchange. The model's performance will be rigorously evaluated on a variety of mainstream benchmark datasets for mathematical reasoning, including the MATH dataset [4], GSM8K, and ARQMath [6]. Our results demonstrate that MathLLaMA achieves superior performance across several key metrics, including problem-solving accuracy, symbolic reasoning, and equation generation, outperforming existing models in these tasks. This work highlights the potential of MathLLaMA to advance AI-driven mathematics education, research, and automated problem-solving.

1.1 Contributions

- Curriculum-Based Fine-Tuning: We employ a curriculum learning strategy, starting with simpler mathematical tasks and progressively introducing more complex problems, such as calculus derivations, discrete mathematics, and number theory. This gradual increase in difficulty allows the model to build a strong foundation in basic mathematical reasoning before tackling advanced topics.
- Extensive Dataset Preparation and Training: MathLLaMA is fine-tuned on the latest high-quality mathematical datasets, including textbooks, research papers, problem sets, and real-world question-answering forums like Stack Exchange and MathOverflow. This comprehensive dataset covers various mathematical fields and problem types, ensuring broad coverage and relevance.
- Benchmark Performance Evaluation: We evaluate MathLLaMA on mainstream datasets such as MATH, GSM8K, and ARQMath to assess its capabilities in mathematical reasoning, symbolic manipulation, and equation generation. Our results indicate that MathLLaMA outperforms existing language models across several metrics, demonstrating its effectiveness in solving mathematical tasks and generating accurate solutions.

MathLLaMA aims to push the boundaries of what LLMs can achieve in the realm of mathematical problem-solving, offering a valuable tool for educators, researchers, and practitioners seeking AI-driven solutions to mathematical challenges. This work represents a step forward in bridging the gap between natural language understanding and formal mathematical reasoning.

2 Related Work

Recent advancements in large language models (LLMs) such as GPT-3, T5, and LLaMA have significantly improved natural language understanding and generation tasks. However, mathematical reasoning presents unique challenges, as it involves structured formal expressions and multistep logical deductions that go beyond simple language comprehension. This section reviews prior research in language modeling for mathematical tasks, specialized models, fine-tuning techniques, and benchmark datasets, highlighting the progress and limitations of current approaches.

2.1 Language Models for Mathematical Reasoning

LLMs like GPT-3 have shown some promise in mathematical problem-solving but still struggle with complex reasoning tasks. For example, while GPT-3 can solve basic math word problems, its performance decreases significantly on tasks that require deeper symbolic manipulation or multistep reasoning. This limitation is attributed to the autoregressive nature of LLMs, which generate tokens sequentially and can easily propagate errors in reasoning across multiple steps. Research using the GSM8K dataset, which contains grade-school math word problems, shows that even state-of-the-art models struggle with multistep reasoning accuracy. Fine-tuning techniques and verification methods have been explored to improve performance, where verifiers evaluate generated solutions to filter out incorrect responses, leading to better accuracy in mathematical tasks.

2.2 Specialized Models for Mathematical Tasks

To address the limitations of general-purpose LLMs, specialized approaches have been developed. For instance, MathematicaGPT integrates symbolic computation engines to solve algebraic problems, combining language modeling with symbolic reasoning. However, this hybrid approach still depends heavily on the capabilities of external tools. Other models, such as GeoSolver and MathQA, focus on specific domains like geometry or math question-answering, using domain-specific datasets and tailored architectures. While these models perform well in their respective areas, their narrow focus limits generalization to broader mathematical tasks.

2.3 Fine-Tuning Techniques for Mathematical Reasoning

Effective fine-tuning methods are crucial for adapting LLMs to specialized domains like mathematics. Key techniques include:

- Curriculum Learning: This involves starting with simpler problems and gradually increasing difficulty, allowing the model to learn foundational skills before tackling more advanced tasks. Curriculum learning has been shown to enhance performance on reasoning-heavy tasks by building a structured learning pathway.
- Prompt Engineering: Prompt-based training has proven effective in guiding models to produce more accurate solutions. For mathematical tasks, prompts can include step-by-step instructions or intermediate steps that break down the reasoning process. This approach helps models better understand problem-solving sequences and improve accuracy.

2.4 Benchmark Datasets for Mathematical Reasoning

Several datasets are used to evaluate the capabilities of LLMs on mathematical tasks, including:

- MATH Dataset: A collection of high school and competition-level problems covering various topics, such as algebra, calculus, and geometry. It is a standard benchmark for evaluating the ability of models to handle complex mathematics.
- GSM8K [7]: Contains grade-school math problems that require basic arithmetic and multi-step reasoning. This dataset has been used extensively to assess fine-tuning techniques and verifier-based improvements for models like GPT-3.
- ARQMath [8]: Derived from Stack Exchange discussions, ARQMath focuses on mathematical question answering, offering a real-world benchmark for evaluating models' problem-solving capabilities in diverse mathematical domains.

2.5 Recent Advances in Mathematical Language Modeling

Recent studies have explored innovative approaches to enhance LLMs' mathematical reasoning abilities. MathScale, for example, uses a pipeline to generate large-scale mathematical reasoning datasets by extracting concepts from existing problems and constructing new questions based on topic relationships. Such instruction-tuning approaches aim to teach models specific problem-solving skills by providing diverse training examples. Other works integrate techniques like process supervision, where models are trained to generate intermediate steps rather than just final answers, helping improve multistep reasoning accuracy.

2.6 Our Approach: MathLLaMA

Building upon these insights, MathLLaMA seeks to overcome the limitations identified in prior works by fine-tuning a large LLaMA model using curriculum learning, prompt engineering techniques. It is evaluated across multiple benchmark datasets, including MATH, GSM8K, and ARQMath, achieving state-of-the-art results in various metrics. This work demonstrates that with specialized training and fine-tuning methods, LLMs can significantly improve their ability to solve complex mathematical problems.

3 method

In the context of complex reasoning tasks, such as multi-step mathematical word problems, individuals often engage in a cognitive process that involves decomposing the problem into manageable intermediate steps.

For instance, consider the math problem: "Bob purchased 3 boxes of apples, with 10 apples in each box, and he also bought 4 boxes of pears, with 12 pears in each box. How many pieces of fruit did he buy in total?" When humans address this problem, they typically decompose it into intermediate steps: "Bob bought $3 \times 10 = 30$ apples and $4 \times 12 = 48$ pears. therefore, he purchased a total of 78 pieces of fruit."

In this paper, we expect LLMs can achieve comparable CoT(Chain of Thought)[9] performance through prompt engineering and fine-tuning. They will first generate intermediate thought processes, then produce the corresponding Python code, and ultimately provide the solution to the problem.

3.1 Data Preparation

As most of the datasets only have an evaluation split, we need to generate a dataset that includes intermediate thinking processes and the corresponding Python code.

In this stage, we now propose three ways to construct our dataset.

- Use Prompt Engineering to have GPT-4 generate math problems directly.
- Manually collect data, use GPT-4 to solve the problems, and then conduct manual verification..
- Using GPT-4 and Few-Shot learning, automatically generate the dataset based on provided math problems and their corresponding answers.

In the future, we will further explore ways to construct an ideal dataset.

3.2 Curriculum Learning

Our main goal is to enhance the ability of large models to address a variety of mathematical problems through curriculum learning, gradually advancing from easier to more challenging datasets. There are numerous datasets available in the field of mathematics to support this endeavor.

- GSM8K: contains 8,500 elementary mathematics problems, with each problem accompanied by a complete solution process. The problems involve basic arithmetic operations and typically require between 2 to 8 steps to resolve. The dataset is partitioned into 7,500 training samples and 1,000 testing samples.
- GAOKAO(Math)[10]: a collection of mathematics questions from the Chinese national college entrance examination (Gaokao) spanning the years 2010 to 2022. It comprises six datasets, including multiple-choice, fill-in-the-blank, and open-ended questions from both National Paper 1 and National Paper 2. Each question in GAOKAO(Math) is accompanied by a detailed solution process, facilitating CoT training.
- MATH: contains 12,500 high school mathematics competition problems, with 7,500 designated for training and 5,000 for testing. This dataset is presented in text format using LaTeX and provides a detailed, step-by-step solution for each problem.

These datasets cover a wide array of mathematical problems at different stages and difficulty levels. By selecting a subset of problems that progress from easier to more challenging, we can effectively enhance our model's curriculum learning experience.

3.3 Fine-Tuning

In the future, we will choose a LLM with an appropriate number of parameters based on our available computing resources. We anticipate using LoRA in conjunction with the LLaMA-Factory[11] framework for fine-tuning.

References

[1] Pan Lu, Liang Qiu, Wenhao Yu, et al. A survey of deep learning for mathematical reasoning. *ArXiv*, 2022.

[2] Wenhu Chen, Xueguang Ma, Xinyi Wang, and William W. Cohen. Program of thoughts prompting: Disentangling computation from reasoning for numerical reasoning tasks. *ArXiv*, 2022.

[3] Junjie Ye, Xuanting Chen, Nuo Xu, et al. A comprehensive capability analysis of gpt-3 and gpt-3.5 series models. *ArXiv*, 2023.

[4] Dan Hendrycks, Collin Burns, et al. Measuring mathematical problem solving with the math dataset. *ArXiv*, 2021.

[5] Xin Wang, Yudong Chen, and Wenwu Zhu. A survey on curriculum learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44:4555–4576, 2021.

[6] Swaroop Mishra, Matthew Finlayson, et al. Lila: A unified benchmark for mathematical reasoning. *ArXiv*, 2022.

[7] Ye Tian, Baolin Peng, Linfeng Song, Lifeng Jin, Dian Yu, Haitao Mi, and Dong Yu. Toward self-improvement of llms via imagination, searching, and criticizing. *CoRR*, abs/2404.12253, 2024.

[8] Wei Zhong, Jheng-Hong Yang, Yuqing Xie, and Jimmy Lin. Evaluating token-level and passage-level dense retrieval models for math information retrieval. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang, editors, *Findings of the Association for Computational Linguistics: EMNLP 2022*, December 2022.

[9] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models, 2023.

[10] Xiaotian Zhang, Chunyang Li, Yi Zong, Zhengyu Ying, Liang He, and Xipeng Qiu. Evaluating the performance of large language models on gaokao benchmark. 2023.

[11] Yaowei Zheng, Richong Zhang, Junhao Zhang, Yanhan Ye, Zheyan Luo, Zhangchi Feng, and Yongqiang Ma. Llamafactory: Unified efficient fine-tuning of 100+ language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, Bangkok, Thailand, 2024. Association for Computational Linguistics.