Fine-Tuning LLMs for Real Analysis Problems

Badnani, Ben
bbadnani@bu.com

Zhang, Cindy
xyz0906@bu.edu

Karimli, Farid
faridkar@bu.edu

Abstract

Large language models usually struggle with complex mathematical questions. Previous research primarily focuses on incorporating supplementary tools or functionality to provide the language models additional layers of context needed to solve such queries. Building upon these existing ideas, we propose an LLM that is capable of providing relevant lemmas, definitions, and other antecedent theorems that correspond to the input mathematical question. Specifically, we fine-tune gpt-3.5-turbo on a corpus of Real Analysis texts, with the objective of providing helpful and pertinent information for Real Analysis questions found on the Harvard Math Entrance Exam.

1. Previous Work

Yiran Wu, Feiran Jia, et al. chained GPT-4 and Python to solve mathematical problems. They used GPT-4 to recognize math problems, then generate related python code. From GPT-4's response, the user proxy agent extracts all code and executes them sequentially. Valid code from previous runs is recorded and will be executed together with the new code to reflect the step-by-step reasoning progress of the model. The results will be returned to GPT-4 and GPT-4 will continue its problem-solving process. [1]

Imani, Du and Shrivastava used zero-shot chain-of-thought prompting technique to generate multiple Algebraic expressions or Python functions [2]. They achieved a high confidence level on their results and improved upon the then-golden standard MultiArith dataset. Wang and Hu used the same technique on more challenging questions and prompts. Their process looked like this:

Stage 1: Input: [input-question] Let's think step by step.

Output: [explanation]

Stage 2: Input: [input-question] Let's think step by step. [explanation] + Therefore, the answer is:

Output: [answer] [3]

2. Problem Statement and Goal

2.1. Problem Statement

State of the art large language models such as GPT-4 have demonstrated these architectures' superb ability at text completion for natural language prompts. However, a common pitfall of these models is their inherent predisposition to hallucinate information or reasoning when it comes to factual based queries. To date, there has been no meaningful progress in the open problem of getting LLMs to produce sound and accurate proofs to solve mathematical questions. Namely, these models are currently unable to pass the Harvard Math entrance exam, which demonstrates mastery of various higher level math fields such as Analysis, Topology, Algebra and more. While the training data for GPT-4 is not available to the general public, GPT-4 does demonstrate an 'understanding' of these concepts and common questions relating to them, suggesting that it has been exposed to the respective fields at a sufficient level in its training process.

Thus, despite having enough exposure to understand these questions, GPT-4 lacks the ability to close the gap between the prompt question and the derivation of the proof.

In this work, we first extract all theorems and definitions from a myriad of real analysis texts [4–8], and then use these to fine tune an instance of gpt-3.5-turbo, that we coin the Lemma Finder, in order to approximate the underlying logical transformation that would allow any theorems that imply the one in question to be derived. Once trained, we then feed the Real Analysis questions from the Harvard Math Entrance exam to this fine-tuned gpt-3.5-turbo model, and use the returned results as additional context in prompting GPT-4 for a solution.

While this pipeline was designed with the restriction to the Real Analysis field in mind, it may seamlessly be extended to other fields to provide for all encompassing model proof derivers.

2.2. Goal

To fine-tune gpt-3.5-turbo on (theorem, lemma) pairs in Real Analysis texts to cultivate a model that is able to accurately provide relevant and supplemental theorems that could be used as a first non-trivial step in deriving solutions for the real analysis questions from the Harvard Math Entrance Exams.

3. Dataset and Method

3.1. Dataset

We curate our dataset using real analysis theorems sourced from mathematics textbooks [4–8]. The dataset will include theorem statements and their corresponding antecedent theorems used in their proofs, as well as definitions necessary for their constructions. Specifically, we cater our collection of textbooks to match the requirements for the Real Analysis portion of the qualifying exam syllabus of the Harvard Math Entrance Exam. [9].

3.1.1 The Lemma Finder

The Lemma Finder is given as input a theorem, and tasked with finding any relevant lemmas, hints, definitions, or other theorems that could be relevant in deriving a solution.

In order to get the training data for the lemma finder, for each digitized real analysis text in our dataset, we will scrape page by page and ask a chatGPT model to extract any such lemmas, definitions, or theorems that are used in proving a specified theorem. Using the response per lemma, we create a sequence of input theorems and target lemmas such that we have our training dataset.

Using this dataset, we will fine-tune the latest available fine-tunable GPT model, gpt-3.5-turbo.

4. Evaluation Criteria

Correctness:

We will feed the real analysis questions from the Harvard Math Entrance Exam to our Lemma Finder, and then with the returned lemmas and definitions it provides, we will feed the original question and these returned lemmas to GPT-4 to generate a solution. We will hand-grade these solutions to assess correctness.

Efficiency: Assess the model's efficiency by measuring the time taken to provide solutions.

Failure Analysis: Analyze failure cases to summarize the reason and types of failure.

Explanation Quality: Evaluate the quality and clarity of the step-by-step explanations provided by the model for the solutions.

References

[1] Y. Wu, F. Jia, S. Zhang, H. Li, E. Zhu, Y. Wang, Y. T. Lee, R. Peng, Q. Wu, and C. Wang, "An empirical study on challenging math problem solving with gpt-4," 2023. 1

[2] S. Imani, L. Du, and H. Shrivastava, "Mathprompter: Mathematical reasoning using large language models," 2023. 1

[3] X. Wang, Z. Hu, P. Lu, Y. Zhu, J. Zhang, S. Subramaniam, A. R. Loomba, S. Zhang, Y. Sun, and W. Wang, "Scibench: Evaluating college-level scientific problem-solving abilities of large language models," 2023. 1

[4] W. Rudin, *Principles of mathematical analysis*. 1953. 1, 2

[5] W. Rudin, *Functional analysis*. 1973.

[6] W. Rudin, *Real and complex analysis*. 1987.

[7] L. C. Evans, *Partial Differential Equations*. 1998.

[8] R. Durrett, *Probability: Theory and Examples*. 1990. 1, 2

[9] "The Harvard Math Qualifying Exam Syllabus." 2