Training Language Models on Mathematical Text

Zhangir Azerbayev, Hailey Schoelkopf, Sean Welleck*, Dragomir Radev

*University of Washington

LILY Lab

Introduction

Recent years have seen tremendous progress in applying pre-trained language models to reasoning tasks that require generating natural language or language-like responses. For example, language models pre-trained on code have become popular for code generation tasks, language model-guided tree search has become popular in formal mathematics, and large models pre-trained on math have been successful for natural language quantitative reasoning.

This project seeks to fine-tune language models on large corpora of unstructured mathematical text to support research into the reasoning abilities of language models.

Data

To train our models, we collect the **proof-pile**, a corpus of ~35GB of mathematical text comprising ~12B tokens. The dataset consists of the following sources

- ArXiv.math (35GB)
- Open-source math textbooks (50MB)
- Formal mathematics libraries (500MB)
- Math Overflow and Math Stack Exchange (500MB)
- Wiki-style sources (50MB)
- MATH dataset (6MB)

Models

For our fine-tune, we use a Pythia-1.3B model, an open-source model from EleutherAI similar to GPT-Neo 1.3B. We train for 40,000 steps. We call our fine-tuned model *proofGPT-v0.1*.

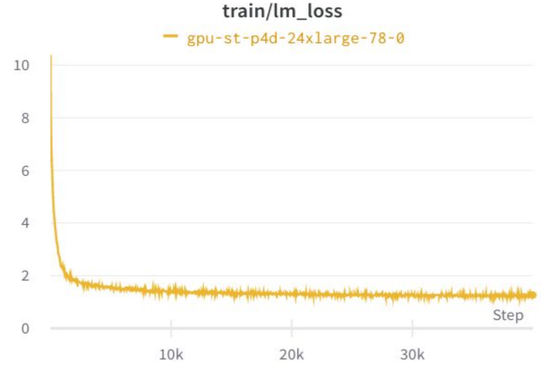

Fig 1. proofGPT-v0.1 training loss curve

NL: Let R be a ring, with M an ideal of R. Suppose that every element of R which is not in M is a unit of R. Prove that M is a maximal ideal and that moreover it is the only maximal ideal of R.

Code-davinci-002 output:

```
theorem exercise_10_7_10 {R : Type*} [ring R] (M : ideal R)
 (hM : ∀ x : R, x ∉ M → is_unit x) :
 is_maximal M ∧ ∀ (N : ideal R), is_maximal N → N = M
```

Fig. 2: Example of ProofNet autoformalization task

	Formalization		
Model	Typecheck rate	BLEU	Accuracy
GPT-J 6B	2.3%	5.9	0%
Code-davinci-002	20.2%	25.7	10.8%
proofGPT*	15.5%	8.7	3.0%

Fig 3. ProofNet autoformalizstion results

Evaluation

A full evaluation suite of quantitative tasks for our model is a work in progress. However, we report performance on the *ProofNet* autoformalization task.

ProofNet autoformalizatoin: ProofNet is a benchmark consisting parallel natural language and formal representations in the Lean proof assistant. We evaluate our models on the task of *statement autoformalization*: converting natural language theorem statements to formal.

To train our models on statement autoformalization, we use the *distilled backtranslation* methodology. First, we use Codex to translate a corpus of 100k Lean formal statements to NL. Then, we train our models in these synthetic pairs in the NL->formal direction.

Future Plans

Next semester, we plan to do further work on the project along the following directions

- Comprehensive evaluation suite, including MMLU, MATH, and identify_math_theorem datasets.
- Expanding the proof-pile, including by a Minerva-style web scrape.
- Investigating pre-training on synthetic data, such as solutions to math problems generated by a CAS.
- Training models with more parameters, up to 20B.