Reinforcement Learning with Verified Rewards (RLVR)

The Goal

- Fine-tune LLM performance on mathematical reasoning
- Specifically, we want an LLM to produce answers that we can easily verify, which means we need to train it to produce an answer in a specific format

What is RLVR?

- This is where RLVR comes in
- RLVR was introduced in a report on a family of models called Tulu 3 by Allen Institute for AI. I largely followed their approach to RLVR for this project
- RLVR is similar to RLHF, but uses deterministic functions to evaluate LLM responses rather than a (computation and labor expensive) reward model trained by human responses

Setup

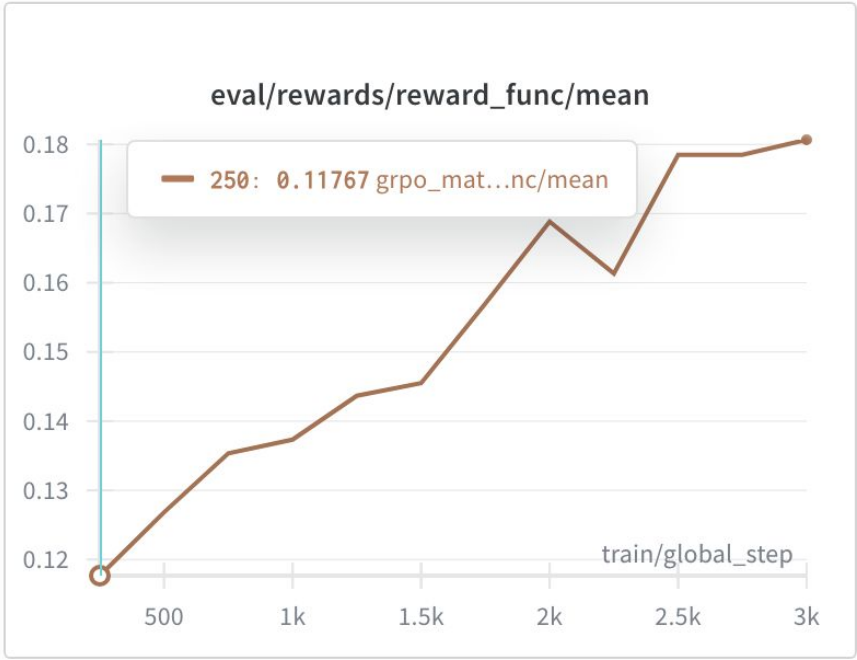
- I used the open-source HuggingFace dataset AllenAI used in the Tulu 3 report, which included many-shot prompting and a specific format (placing final answers in “\boxed{ __ }”) for a large number of math problems
- I also used HuggingFace for the training pipeline, including the underlying model (an 8B parameter version of Llama 3.1*)
- I hosted my script on RunPod, using LoRA optimization to save memory. I saved results to W&B

Algorithm

- I used GRPO, a policy optimization algorithm that works very well with RLVR
- Intuitively, GRPO produces a group of responses at a time, and updates the parameters of the model being fine-tuned in the direction of the best responses of the group (evaluated with the deterministic reward function)
- GRPO also uses methods common in other policy optimization algorithms (like PPO) to ensure the parameters do not change too much

Results

- The model clearly improved - due to the cost of compute I only trained for about ¼ the amount of time as in the original report
  - My results would project to accuracy of about 0.36 after 12 epochs, which is shy of their 0.44 likely due to LoRA
- Responses were largely cogent even for incorrect answers



Questions?