Eliciting Reasoning in LLMs using Logprob-based Rewards

Supervisors: Auguste Poiroux, Eric Chen, Nicolò De Sabbata

Overview

This project aims to explore reinforcement learning (RL) techniques to enhance reasoning capabilities in language models, focusing on reproducing and extending the DeepSeek-R1 approach. Students will work with open-source tools to implement and compare different reward mechanisms, including rule-based and log-probability-based rewards.

Log-Probability-Based Rewards

In reinforcement learning (RL) for language models, log-probability-based rewards offer a general and continuous feedback mechanism, especially useful in tasks lacking explicit correctness criteria.

What Are Log-Probability-Based Rewards?

Language models assign probabilities to sequences of tokens. The log-probability (logprob) of a token reflects the model's confidence in predicting that token given its preceding context. By summing the logprobs of all tokens in a generated sequence, we obtain the total log-likelihood of that sequence.

In RL settings, this log-likelihood can serve as a reward signal. For instance, if a model generates a reasoning trace leading to an answer, the reward can be defined as the log-likelihood the model assigns to the **correct** answer given its own reasoning (irrespective of the answer generated by the LLM). This approach encourages the model to produce reasoning that increases its confidence in the correct answer.

Advantages

- Domain Generality: Unlike rule-based rewards that require task-specific evaluators (e.g., test cases for code, symbolic equivalence for maths, …), logprob-based rewards can be applied across various tasks without additional infrastructure.
- Continuous Signal: Provides nuanced reward signals, reflecting degrees of confidence rather than binary correctness.

Challenges

- Length Normalization: Longer sequences naturally accumulate lower total log-likelihoods. Solution: Normalization techniques are necessary to prevent biases and have a uniform reward across all answers in a dataset.
- Reward Misalignment (hypothetical): The model might learn to exploit the reward signal in some way. Solution: Tweaking the KL divergence penalty term.

- Training Stability (hypothetical): RL training with logprob-based rewards might become unstable. Solution: Tweaking parameters and/or reshaping the reward signal.

By integrating log-probability-based rewards, we can train language models to develop reasoning capabilities even in the absence of explicit supervision, broadening their applicability to a wider range of tasks.

Training Algorithm

In DeepSeek-R1, the Group Relative Policy Optimization (GRPO) algorithm is used to train the model. For each input, the model generates several reasoning traces and corresponding answers. Each output is evaluated with a reward function, and the advantage is computed by comparing an individual reward to the average reward of the group. This relative advantage helps stabilize training across problems of varying difficulty.
This algorithm is also perfectly suited for logprob-based rewards. *Intuitively, the model will be trained to prefer reasoning traces increasing its confidence in correct answers and, therefore, in generating them at test time.*

Some concise equations capturing key components of logprob-based training in this context:

1. **Reward Computation:** For an input prompt *p*, a generated reasoning trace *r*, and the **correct** answer *a* (i.e. the ground-truth), define the reward as the log-probability of the correct answer given the reasoning:

   $$R = log\, P(a \mid p, r)$$

   In other words, we discard the answer generated by the LLM and replace it with the official answer *a*. The reward is then the probability for the LLM to generate the correct answer *a* given the prompt *p* and the reasoning *r*.

2. **Baseline (Group Average):** For a batch of N generated outputs (reasoning traces and answers) for the same prompt, compute the baseline as the average log-probability:

   $$b = \frac{1}{N}\sum_{i=1}^{N} log\, P(a_i \mid p,\ r_i)$$

3. **Advantage Calculation:** The advantage of a given output is the difference between its log-probability and the batch average:

   $$A = R - b$$

4. **Policy Gradient Update:** With the advantage, the gradient of the objective with respect to the model parameters θ is computed. For more details, I invite you to read the DeepSeekMath paper referenced at the end of this document.

Project Objectives

- Reproduce DeepSeek-R1 with rule-based rewards on a small scale
  - Utilize resources such as Mini-R1, Open-R1, TinyZero, SimpleRL-reason, … to reproduce the DeepSeek-R1 approach on a smaller scale (e.g., 1B parameter model).
- Implement and evaluate logprob-based rewards
  - Replace the rule-based reward mechanism with a log-probability-based reward, where the model is rewarded based on the likelihood it assigns to correct answers, given its reasoning steps.
  - Experiment with different normalization techniques for the log-probability reward to ensure stable and meaningful learning signals.
  - Compare the performance and reasoning quality between rule-based and log-probability-based rewards.
- Apply logprob-based rewards to open-ended tasks
  - Extend the log-probability reward approach to tasks lacking clear-cut evaluation metrics, such as creative writing or poetry.
  - Investigate how the model's reasoning and output quality are influenced in these subjective domains.

References

1. DeepSeek-AI. (2025). *DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning*. Link
2. Kimi Team. (2025). *Kimi k1.5: Scaling Reinforcement Learning with LLMs*. Link
3. Shao et al. (2024). *DeepSeekMath: Pushing the Limits of Mathematical Reasoning in Open Language Models.* Link