ML Assignment 3: GRPO Training for Mathematical Reasoning

Overview

The goal of this assignment is to implement **GRPO (Group Relative Policy Optimization)**, a reinforcement learning algorithm designed for large language models that performs well on mathematical reasoning tasks. GRPO is a simplified variant of PPO (Proximal Policy Optimization) and has achieved strong results in several state-of-the-art reasoning models (e.g., DeepSeek-R1).

Background

GRPO Algorithm Summary

Key features of GRPO:

1. **Within-group advantage computation**: no value network is used; advantages are computed by comparing each response to the group's average reward.
2. **Simple and efficient**: no critic network required, making computation and implementation lightweight.
3. **Effective**: validated on several top LLM reasoning models.

Algorithm Flow

Each training step includes:

1. **Sample multiple responses per prompt** (e.g., `group_size=4`)
2. **Compute rewards for all responses**
3. **Compute group-relative advantages**
4. **Update the policy using a PPO-style objective**

Assignment Tasks

You need to complete **5 TODOs** in `grpo_homework.py`:

TODO 1: Tokenization

Location: `GSM8KDataset.__getitem__`
Task: Use the tokenizer to convert the input prompt into tensors that the model can process.

TODO 2: GRPO Advantage Computation

Location: `compute_advantages_grpo`
Task: Implement the core group-relative advantage calculation.

TODO 3: PPO Policy Loss

Location: `compute_policy_loss`
Task: Implement the PPO clipped objective loss for the policy.

TODO 4: Compute Model Log-Probs

Location: `compute_logprobs_from_model`
Task: Compute log probabilities for generated sequences from the model.

TODO 5: GRPO Training Step Implementation

Location: `train_grpo`
Task: Implement the complete GRPO training flow.

Environment Setup

1. **Install dependencies**:

```
pip install -r requirements.txt
```

2. **Model path**: you can download and use the Qwen-2.5-1.5B model:

```
https://hf-mirror.com/Qwen/Qwen2.5-1.5B-Instruct
```

3. **GPU configuration**: default to GPU:

```
device = torch.device("cuda")
```

Run the Code

Test your implementation with:

```
python grpo_homework.py {YOUR_MODEL_PATH}
```

Example output:

```
Loading tokenizer and model...
Loading dataset...
Setting up optimizer...
Starting GRPO training...
...
Training completed!
```

Submission Instructions

Submit:

1. A report describing:
    - Algorithm and implementation overview
    - Training hyperparameters
    - RL training logs / progress
    - Before/after answer comparisons
2. The completed `grpo_homework.py` file
   Filename format:

```
studentID_name_hw2.py
```

Example: `12345678_zhangsan_hw2.py`

Grading Rubric

Total 100 points:

- **TODO 1** (15 pts): Correct tokenization
- **TODO 2** (15 pts): Correct advantage computation
- **TODO 3** (15 pts): Correct PPO policy loss
- **TODO 4** (15 pts): Correct log-prob computation
- **TODO 5** (20 pts): Complete training flow and measurable improvement vs. baseline
- **Report** (20 pts)

Resources

- GRPO original paper
- PPO original paper
- AReaL repository