Spring 2025 | DATA 266 | Homework -4
Deadline – 11.59 PM – May 10th, 2025
20 Points

Complete RLHF Pipeline — Reward Modeling, PPO Fine-Tuning, and LoRA Comparison

In this assignment, you will build a complete Reinforcement Learning from Human Feedback (RLHF) training pipeline for language models. The process consists of two major phases:

- (1) Train a Reward Model from human preference data.
- (2) Fine-Tune a Policy Model using Proximal Policy Optimization (PPO) with the trained reward model.

You will implement both full model fine-tuning and parameter-efficient fine-tuning using Low-Rank Adaptation (LoRA).

🛠 Datasets

You will use two datasets, each designated for a specific phase:

- Reward Model Training Dataset:
  ➔ Anthropic HH-RLHF (Anthropic/hh-rlhf) from Hugging Face.
  ➔ Focus on the columns: prompt, chosen, rejected.
  ➔ Use a subset of 1000 examples for manageable training.
- PPO Fine-Tuning Dataset:
  ➔ OpenAssistant Conversations Dataset (OpenAssistant/oasst1) from Hugging Face.
  ➔ Extract only the prompter texts (user questions).
  ➔ Drop all assister (assistant response) texts.
  ➔ Randomly sample 1000 prompts for PPO training.

🧠 Part 1: Reward Model Training

You will train a reward model to predict human preferences between two responses to a given prompt.

- Load and preprocess the Anthropic HH-RLHF dataset.
- For each sample, you have a prompt, a chosen response, and a rejected response.
- Use a causal language model (e.g., gpt2-medium) as the base reward model.
- Set up a training procedure to teach the reward model to assign:
  - Higher reward to the chosen response
  - Lower reward to the rejected response
- Training Details:
  - Loss: Pairwise Margin Ranking Loss
  - Optimizer: AdamW
  - Learning Rate: 5e-5
  - Batch Size: 8
  - Epochs: 3
  - Tokenize input carefully (prompt + response concatenation).

- Save the best reward model based on validation loss for use in PPO fine-tuning.

🧠 Part 2: PPO Fine-Tuning of Policy Model

You will fine-tune a **new copy** of gpt2-medium as the policy model using PPO, guided by the reward model you just trained.

- Sample 1000 prompts from the **OpenAssistant Conversations** dataset.
- Only use prompter texts (drop assister).
- For each prompt:
  - Generate a response from the policy model.
  - Evaluate the response using the **frozen reward model**.
  - Apply PPO optimization to improve reward scores.
- **Training Details**:
  - Optimizer: PPOTrainer (trl library recommended)
  - Learning Rate: 1e-5
  - Batch Size: 64
  - Forward Batch Size: 16
  - Epochs: **3**
  - Max Prompt Length: 256 tokens
  - Max Response Length: 128 tokens
  - Target moderate KL divergence (~0.02–0.1).

Train two separate models:

1. **Full Fine-Tuning**: Update all parameters of the policy model.
2. **LoRA Fine-Tuning**: Insert LoRA adapters into attention layers (q_proj and v_proj) with:
   - Rank (r): 8
   - Alpha: 32
   - Dropout: 0.1
   - Only LoRA parameters are updated.

Save both fine-tuned model checkpoints separately.

📊 Evaluation and Reporting

You must evaluate your results based on **three aspects**:

1. Reward Score Evaluation (Quantitative)

- On a held-out set (minimum 50) of unseen prompts (not used during PPO training):
  - Generate responses from:
    - (a) Original pre-trained model
    - (b) Full fine-tuned model
    - (c) LoRA fine-tuned model
  - Score each response using the reward model.
- Metrics:
  - Average reward score before PPO vs. after PPO.
    - (a) vs. (b) and (a) vs. (c)

- Reward gain = Final Avg Reward − Initial Avg Reward.

2. KL Divergence Monitoring (Quantitative)

- When using PPO libraries like trl, the KL divergence between the fine-tuned policy model and the original base model is automatically approximated and logged during training.
  - You must track and plot: KL divergence vs. PPO steps.
- Interpretation:
  - If KL grows too quickly, the model is diverging too much.
  - If KL stays very low, learning may be slow.

Target maintaining a moderate KL divergence (~0.02–0.1).

3. Manual Human Evaluation (Qualitative): Write a comparative analysis on the manual rating.

- Select 10 unseen prompts.
- For each prompt:
  - Generate responses from:
    - (a) Original pre-trained model
    - (b) Full fine-tuned PPO model
    - (c) LoRA fine-tuned PPO model.
- Manual Rating:
  - Rate each generated response on a scale of 1–5 based on:
    - Coherence
    - Relevance
    - Helpfulness
    - Completeness
    - Hallucination/toxicity (if any)

You are required to follow:

1. Submit **one** MS/PDF/Scanned document:
   - Include all the steps of your calculations.
   - Include the summary of the model.
   - Attach screenshots of your code.
   - Attach screenshots – showing first few epochs of model training.
   - Attach screenshots of the important code outputs such as confusion matrices, learning curves, and classification reports.
2. Source code:
   a. Python (Jupyter Notebook)
   b. Ensure it is well-organized with comments and proper indentation.

- **Failure to submit the source code will result in a deduction of full/partial points.**
- Format your filenames as follows: "your_last_name_HW1.pdf" for the document and "your_last_name_HW1_source_code.ipynb" for the source code.
- Before submitting the source code, please double-check that it runs without any errors.
- Must submit the files separately.
- Do not compress into a zip file.
- HW submitted more than 24 hours late will not be accepted for credit.