courserd

Generative AI Advance Fine-Tuning × for LLMs

Module 1

Different Approaches to Fine-Tuning

Fine-Tuning Causal LLMs with Human
Feedback and Direct Preference

- Large Language Models (LLMs) as Distributions Video
- From Distributions to Policies
- Reinforcement Learning from Human Feedback
- (RLHF)
 Video
- Proximal Policy Optimization (PPO)
 Video
- PPO with Hugging Face Video
- PPO Trainer
 Video
- Log-derivative Trick
 Ungraded Plugin
- Lab: Reinforcement Learning from Human
 Feedback using PPO
 Ungraded App Item
- Summary and Highlights
- Practice Quiz: Proximal Policy Optimization (PPO)
 Practice Assignment
- DPO: Partition Function Video
- DPO: Optimal Solution Video
- From Optimal Policy to DPO
- DPO with Hugging Fac
- Lab: Direct Preference Optimization (DPO) using
 Hugging Face
 Ungraded App Item
- Fine-tune LLMs Locally with InstructLab
 Ungraded Plugin
- Summary and Highlights
- Summary and Highlights
 Reading
- Practice Quiz: Direct Preference Optimization

- Congratulations! You have completed this module. At this point in the course, you know the following:
- The reward function provides human feedback for the inserted query.
- Rollouts help queries and responses to review the sampling process. The rollout libraries, such as Hugging Face, differ from the reinforcement learning.
- The expected rewards use an empirical formula to understand how an agent performs in the language model.
- Reinforcement learning from human feedback (RLHF) uses response distribution as an input query to finetune the pre-trained LLMs.
- Pre-trained reward model evaluates and generates a reward for the query plus response.
- Proximal policy optimization (PPO) provides feedback on the quality of actions taken by the policy.
- The sentiment analysis pipeline scores evaluate the generated responses' quality. Pipe_outputs list generates scores for the responses.
- The LengthSampler varies text lengths for data processing, enhances model robustness, and simulates realistic training conditions.
- The sample query questions may provide various random responses based on the probability distribution.
- The transformer model generates probabilities for different words using the softmax function.
- You can select words at various timestamps and change the probabilities for those words.
- The generation parameters, such as temperature, top-k sampling, beam search, top-p sampling, repetition penalty, and max and min tokens, help change the sequences generated using LLMs.
- Objective functions coordinate algorithms and data to reveal patterns, trends, and insights to produce
 accurate predictions. They measure the difference between an ML model's predicted outcomes and target
 values.
- The Kullback-Leibler, or KL, divergence measures the difference between two probability distributions, the desired and the arbitrary policy.
- The optimal solution scales the reference model to the reward function, with the beta parameter controlling the constant.
- Following the policy distribution, the language model generates responses based on the inserted query.
- The policy gradient method maximizes the objective function, and PPO helps to achieve this maximization.
- To optimize the policy, derive the sample response, estimate the reward, and extend the dataset.
- You can calculate the log derivative by identifying a policy that maximizes the objective function by simplifying the expression and converting it into analytical distributions.
- Use a toy gradient ascent example using stochastic gradient ascent or SGA to maximize the objective function compared to a standard optimization problem with maximum likelihood.
- A positive update occurs when a reward is positive, and a negative update occurs when a reward is negative.
- To train the model, regularly evaluate it using human feedback, use the moderate beta value, and increase the temperature.