Large Language Models

Homework 4, Theory

Javad Razi - 401204354

First Problem

A) Preventing Excessive Score Difference

The loss function in the question aims to prevent the scoring model from assigning excessively large score differences between responses. This is achieved by using a sigmoid function, which squashes the score difference between responses into the interval [0, 1]. The loss function then encourages the score difference to be close to 0, which indicates that the scoring model is able to distinguish between good and bad responses without assigning unreasonably large scores to either.

This is advantageous because it helps to ensure that the scoring model is making fair and accurate comparisons between responses. If the scoring model were to assign excessively large score differences, this could lead to the model being biased towards selecting responses with high scores, even if those responses are not actually the best ones.

B) Objective Function Explanation

The objective function for learning the model consists of three terms:

1. **Reward term:** $r\big(\theta(x,y)\big)$ measures the quality of the response y generated by the model θ for the input x. This term encourages the model to generate responses that are relevant, informative, and coherent.
2. **Policy gradient term:** $-\beta \log\left(\frac{\pi_{\phi}^{RL}(y|x)}{\pi^{SFT}(y|x)}\right)$ encourages the model to generate responses that are more likely to be preferred by humans than responses generated by a standard language model π^{SFT}. This term helps the model learn to align its responses with human preferences.
3. **Entropy regularization term:** $\gamma \mathbb{E}_{x\sim D_{pretrain}}\left[\log\left(\pi_{\phi}^{RL}(x)\right)\right]$ encourages the model to generate diverse responses by penalizing responses that are too predictable. This term helps prevent the model from becoming overly repetitive or biased towards certain types of responses.

C) Relationship Between $r_\theta(x, y)$ and ϕ

In the objective function:

$$\text{objective}(\phi) = \mathbb{E}_{(x,y)\sim\pi_\phi^{RL}}\left[r\big(\theta(x,y)\big) - \beta\log\left(\frac{\pi_\phi^{RL}(y|x)}{\pi^{SFT}(y|x)}\right)\right] + \gamma\mathbb{E}_{x\sim D_{pretrain}}\left[\log\left(\pi_\phi^{RL}(x)\right)\right]$$

the derivative is taken with respect to ϕ, yet the term $r_\theta(x, y)$ is present. The presence of $r_\theta(x, y)$ within the objective function, even though it is a function of θ and not ϕ, can be explained by considering the role of r_θ as the reward signal in reinforcement learning.

The reward model r_θ provides an evaluation of the quality of the outputs (actions) produced by the policy π_ϕ^{RL}. During the optimization process, we are not directly altering the parameters θ of the reward model; instead, we are adjusting the policy parameters ϕ to maximize the expected reward as assessed by r_θ. The derivative of the objective function with respect to ϕ is not zero because ϕ directly influences the probability of selecting actions that are being evaluated by r_θ. Hence, changes in ϕ will affect the expected reward, even though r_θ itself remains constant with respect to ϕ.

D) Implementing Derivatives of Target Function

To derive the gradient of the expected return with respect to the policy parameters (θ), we can use the Policy Gradient Theorem. This theorem provides a method to compute policy gradients without explicitly computing the derivative of the expected return, which can be intractable.

Given the target function:

$$L_\theta = \mathbb{E}_{\pi_\theta}[G_t]$$

where G_t represents the total return from time t, the theorem states that the gradient of the expected return with respect to the policy parameters is:

$$\nabla_\theta L_\theta = \mathbb{E}_{\pi_\theta}[\nabla_\theta \log\pi_\theta(y|x) \cdot G_t]$$

This equation is derived using the log-derivative trick, which leverages the identity:

$$\nabla_\theta \pi_\theta(y|x) = \pi_\theta(y|x)\nabla_\theta \log\pi_\theta(y|x)$$

By rewriting the derivative of π_θ in terms of $\log\pi_\theta$, we can pull the gradient operator inside the expectation. This results in an expectation over trajectories (τ) of the product of the gradient of the log-probability of the trajectory under the policy and the total return, which can be estimated using sample trajectories:

$$\nabla_\theta L_\theta = \mathbb{E}_{\pi_\theta,\tau}[\nabla_\theta \log\pi_\theta(\tau) \cdot G_\tau]$$

This formula allows for the estimation of the gradient by sampling trajectories and computing the gradient of the log-policy weighted by the returns.

E) KL Divergence Estimation

Why Direct Computation of KL Divergence is Challenging:

The expression for KL divergence is given by:

$$D_{KL}\left(\pi_{\phi}^{RL} \parallel \pi^{SFT}\right) = \mathbb{E}_{y\sim\pi_{\phi}^{RL}(y|x)}\left[\log\left(\frac{\pi_{\phi}^{RL}(y|x)}{\pi^{SFT}(y|x)}\right)\right]$$

Directly computing this expression is challenging due to the following reasons:

Intractability of Expectation: The expectation over $\pi_{\phi}^{RL}(y|x)$ is typically intractable, especially in high-dimensional spaces or when the policy's distribution is complex.

Sampling Issues: Even if we resort to sampling to approximate this expectation, it requires samples from π_{ϕ}^{RL}, which might be difficult or inefficient to obtain.

The proposed estimator is:

$$D_{KL}(q \parallel p) \approx \frac{1}{N}\sum_{i=1}^{N}\frac{1}{2}\left[\log\left(q(x_i)\right) - \log\left(p(x_i)\right)\right]^2, \quad x_i \sim q(x)$$

This estimator is advantageous because:

Variance Reduction: Squaring the log ratio reduces the impact of extreme values (which might be due to sampling anomalies or regions where the policy ratio is highly skewed), leading to a reduction in variance.

Symmetry in Error: Squaring emphasizes both underestimation and overestimation equally, leading to a more balanced error profile.

F) Reducing Bias in the Estimator

The proposed estimator, while having lower variance, introduces bias. This happens because the expectation of the square of a quantity is not the same as the square of the expectation of that quantity (a consequence of Jensen's inequality).

To reduce the bias, one approach is to include correction terms that approximate the higher-order moments of the distribution. One such corrected estimator can be formulated as:

$$D_{KL}(q \parallel p) \approx \frac{1}{N}\sum_{i=1}^{N}\left[\log\left(\frac{q(x_i)}{p(x_i)}\right)\right] - \frac{\lambda}{2N^2}\sum_{i=1}^{N}\left[\log\left(\frac{q(x_i)}{p(x_i)}\right) - \mu\right]^2$$

Where μ is the sample mean of the log ratios and λ is a correction factor that can be tuned based on the distribution properties or empirically optimized. This correction term essentially tries to account for the bias introduced by the square operation by adjusting for the spread around the mean log ratio.

Problem Two

Most of the mathematical proofs, and methods used in this problem were either directly, or indirectly derived from the original paper "DPO: Your Language Model is Secretly a Reward Model"

A) Solution to DPO Objective Function

To prove the equivalence, we need to show that the solution to the maximization problem is indeed the policy given by the $softmax$ distribution over the exponentiated rewards adjusted by the reference policy and the temperature parameter β.

Let's consider the optimization problem:

$$\max_{\pi_\theta} \mathbb{E}_{x\sim D, y\sim\pi_\theta(y|x)}\left[r_\phi(x,y) - \beta D_{KL}\left(\pi_\theta(y|x) \parallel \pi_{ref}(y|x)\right)\right]$$

We want to find a policy $\pi_\theta(y|x)$ that maximizes the expected reward minus a penalty term which is the KL divergence between $\pi_\theta(y|x)$ and a reference policy $\pi_{ref}(y|x)$, scaled by a temperature parameter β.

We define the Lagrange multiplier $Z(x)$ to enforce the constraint that $\pi_\theta(y|x)$ is a valid probability distribution:

$$L(\pi_\theta, Z) = \mathbb{E}_{x\sim D, y\sim\pi_\theta(y|x)}[r(x,y)] - \beta\mathbb{E}_{x\sim D, y\sim\pi_\theta(y|x)}\left[\log\frac{\pi_\theta(y|x)}{\pi_{ref}(y|x)}\right] - \mathbb{E}_{x\sim D}\left[Z(x)\left(\sum_y \pi_\theta(y|x) - 1\right)\right]$$

Taking the derivative of L with respect to $\pi_\theta(y|x)$ and setting it to zero gives us the condition for optimality:

$$\frac{\partial L}{\partial \pi_\theta(y|x)} = r(x,y) - \beta\left(1 + \log\frac{\pi_\theta(y|x)}{\pi_{ref}(y|x)}\right) - Z(x) = 0$$

Solving for $\pi_\theta(y|x)$, we get:

$$\pi_\theta(y|x) = \pi_{ref}(y|x)\exp\left(\frac{r(x,y)}{\beta} - 1\right)$$

To make $\pi_\theta(y|x)$ a valid probability distribution, we normalize it by dividing by the sum over all possible y, which gives us the partition function $Z(x)$:

$$Z(x) = \sum_y \pi_{ref}\,(y|x)\exp\left(\frac{r(x,y)}{\beta}\right)$$

Substituting $Z(x)$ back into the expression for $\pi_\theta(y|x)$, we get the softmax policy:

$$\pi_r(y|x) = \frac{1}{Z(x)}\pi_{ref}(y|x)\exp\left(\frac{r(x,y)}{\beta}\right)$$

This policy $\pi_r(y|x)$ is the solution to our original optimization problem, and it is equivalent to the softmax distribution over the exponentiated rewards adjusted by the reference policy and the temperature parameter β.

Therefore, we have proved that the solution to the maximization problem is equivalent to the softmax policy given in the statement.

B) Cost Function for DPO

To derive the cost function for DPO, we substitute the optimal policy distribution πθ(y|x) into the loss function of the scoring model. The loss function is given by:

$$\text{loss}(\theta) = \mathbb{E}_{x,y_w,y_l\sim D}\left[r_\phi(x,y_w) - r_\phi(x,y_l)\right]$$

where (x,yw,yl) is a triple of context, ground truth, and rejected response.

Substituting the optimal policy distribution πθ(y|x), we get:

$$\text{loss}(\theta) = \mathbb{E}_{x,y_w,y_l\sim D}\left[\left(r_\phi(x,y_w) - r_\phi(x,y_l)\right) - \beta D_{KL}\left(\pi_{ref}(y|x) \parallel \pi_{ref}(y|x)\right)\right]$$

This is the cost function for DPO, which we can use to train the reward model. The first term encourages the scoring model to distinguish between good and bad responses, while the second term encourages the scoring model to make consistent judgments.

C) Calculating Gradient of Cost Function

The gradient of the cost function for DPO is given by:

$$\nabla_\theta \text{Cost}(\pi_r) = -\mathbb{E}_{x\sim D, y\sim\pi_r(y|x)}\left[\nabla_\theta \log\pi_\theta(y|x) \cdot \left(r_\phi(x,y) - \beta\left(1 + \log\pi_\theta(y|x) - \log\pi_{ref}(y|x)\right)\right)\right]$$

This gradient consists of three terms:

Reward Term Gradient: This term encourages the policy to increase the expected reward. A positive gradient indicates that the policy should increase the probability of actions that lead to higher rewards, while a negative gradient indicates that the policy should decrease the probability of actions that lead to lower rewards.

Entropy Regularization Gradient: This term encourages the policy to explore different actions and avoid becoming deterministic. A positive gradient indicates that the policy should increase the entropy of its distribution, while a negative gradient indicates that the policy should decrease the entropy of its distribution.

KL Divergence Gradient: This term encourages the policy to stay close to the reference policy. A positive gradient indicates that the policy should increase the probability of actions that are similar to those of the reference policy, while a negative gradient indicates that the policy should decrease the probability of actions that are different from those of the reference policy.

The combination of these three terms helps the policy to learn effectively by balancing the goals of maximizing rewards, exploring different actions, and staying close to the reference policy.