LLM Alignment Week 1 Progression

Yuanqi Zhao

University of Oxford

17.July.2024

LLM Alignment Procedure

- Pre-training;
- Supervised-Fine-Tuning (SFT);
- RLHF/DPO.

Pre-Training and Fine-Tuning

In Pre-Traning Stage, the objective function is normally:

$$\text{Max } L_{GPT} = \sum_{i=1}^{n} logP(x_i|x_{i-1}, ..., x_{i-k})$$

In Fine-Tuning Stage, the objective function is normally:

$$\alpha L_{GPT} + L_{FT},\ L_{FT} = \sum_{(x,y)} log(P(y|x^1, ..., x^m))$$

$$\text{where } P(y|x^1, ..., x^m) = softmax(h_l^m W_y)$$

$$h_l^m : \text{final transformer blocks activation}, W_y : Parameter$$

Reward Model

Objective Function:

$$loss(\theta) = -\frac{1}{\binom{K}{2}} E_{(x,y_w,y_l)\sim D}[log(\sigma(r_\theta(x,y_w) - r_\theta(x,y_l)))]$$

Reward Model normally take the same structure as our language model, but with smaller size, and the final layer is replaced by a fully connected layer to output a scalar(or say reward).

RLHF

Objective Function:

$$Obj(\phi) = E_{(x,y)\sim D_{\pi_{\phi}^{RL}}}[r_\theta(x,y) - \beta log(\pi_{\phi}^{RL}(y|x)/\pi^{SFT}(y|x))]$$

$$+\gamma E_{x\sim D_{pretrain}}[log(\pi_{\phi}^{RL}(x))]$$

$r_\theta(x,y)$: Reward Model parameterised by θ;
π_{ϕ}^{RL} : learned RL Policy;
π^{SFT} : Supervised trained model;
$D_{pretrain}$: pretraining distribution;
β : KL reward coefficient;
γ : control the strength of the KL penalty and pretraining gradients.

RLHF Objective Function

$$\begin{aligned}
objective(\phi) &= E_{(x,y)\sim D_{\pi_\phi^{RL}}}[r_\theta(x,y) - \beta log(\pi_\phi^{RL}(y|x)/\pi^{SFT}(y|x))] + \gamma E_{x\sim D_{pretrain}}[log(\pi_\phi^{RL})] \\
&= E_{(x,y)\sim D_{\pi^{RL'}}}\left[\frac{\pi_\phi^{RL}(y|x)}{\pi^{RL'}(y|x)} r_{\theta'}(x,y) - \beta log(\pi^{RL'}(y|x)/\pi^{SFT}(y|x))\right] + \gamma E_{x\sim D_{pretrain}}[log(\pi_\phi^{RL})] \\
&= E_{(x,y)\sim D_{\pi^{RL'}}}\left[\min\left(\frac{\pi_\phi^{RL}(y|x)}{\pi^{RL'}(y|x)} r_{\theta'}(x,y), clip\left(\frac{\pi_\phi^{RL}(y|x)}{\pi^{RL'}(y|x)}, 1-\varepsilon, 1+\varepsilon\right) r_{\theta'}(x,y)\right) - \beta log(\pi^{RL'}(y|x)/\pi^{SFT}(y|x))\right] + \gamma E_{x\sim D_{pretrain}}[log(\pi_\phi^{RL})] \\
&= E_{(x,y)\sim D_{\pi^{RL'}}}\left[\min\left(\frac{\pi_\phi^{RL}(y|x)}{\pi^{RL'}(y|x)} A^{\theta^{RL'}}(x,y), clip\left(\frac{\pi_\phi^{RL}(y|x)}{\pi^{RL'}(y|x)}, 1-\varepsilon, 1+\varepsilon\right) A^{\theta^{RL'}}(x,y)\right)\right] + \gamma E_{x\sim D_{pretrain}}[log(\pi_\phi^{RL})]
\end{aligned}$$

Figure 1: Simplification of RLHF Obj

DPO - Alternative method of RLHF

- DPO is Equivelent to RLHF, but it's much simpler to train!
- DPO aims to increase the relative log prob. of preferred to dis-preferred responses.

$$L_{DPO}(\pi_\theta; \pi_{ref}) = -E_{(X,Y_w,Y_l)\sim D}[\log \sigma(\beta \log \frac{\pi_\theta(y_w|x)}{\pi_{ref}(y_w|x)} - \beta \log \frac{\pi_\theta(y_l|x)}{\pi_{ref}(y_l|x)})]$$

L_{DPO} is differentiable, so that we can use backward propagation!

To-do list

- Code implementation for the theoretical part above.
- Most NLP labs are currently working on modifications to the DPO algorithm.
- Simpo, Orpo is a direction worth to mention.
- Pay attention to Engineering aspect. 1