arxiv.org/abs/2401.08565

Tuning Language Models by Proxy

Proxy-Tuning：用小模型引导大模型的零参数调优方法

Alisa Liu[♡] **Xiaochuang Han**[♡] **Yizhong Wang**[♡♣] **Yulia Tsvetkov**[♡]
Yejin Choi[♡♣] **Noah A. Smith**[♡♣]

[♡]University of Washington [♣]Allen Institute for AI
alisaliu@cs.washington.edu

开源代码：github.com/alisawuffles/proxy-tuning

简介

本文提出Proxy-Tuning (代理微调)，一种作用于开源llm的解码(decoding)方法，本方法不需要对目标llm参数进行tuning，而是tuning一个更小规模的llm (称为expert, 专家)，然后在解码时，利用expert与其未tuning版本(称为anti-expert，反专家)之间的logit差值，对目标llm的logit做修正，从而近似出tuning后的效果。比如对于Llama2-70B，tuning成本太高，那就去tuning Llama2-7B或13B，然后指导Llama2-70B做解码，能够做到效果比较近似Llama2-70B-Chat。

$$p_{\tilde{\mathcal{M}}}(X_t \mid x_{<t}) = \text{softmax}\left[s_{\mathcal{M}}(X_t \mid x_{<t}) + s_{\mathcal{M}^+}(X_t \mid x_{<t}) - s_{\mathcal{M}^-}(X_t \mid x_{<t})\right]$$

背景

虽然现在开源llm的效果越来越好，但是相应的规模也越来越大，比如最近kimi k2已经达到了1T参数量，这使得对这些开源llm进行任务或领域适配的tuning成本越来越高甚至是难以接受。目前已经有诸如lora等参数高效调优(parameter-efficient tuning)方法，能够做到仅tuning少量参数而无需全部参数。本文作者更进一步提出了proxy-tuning(代理微调)，干脆不tuning 目标llm，而是找一个小规模的llm进行tuning，然后在解码时引导目标llm去做生成，效果竟然也不错。

Proxy-Tuning

其实全文讲的就是右边那个公式

$$p_{\tilde{\mathcal{M}}}(X_t \mid x_{<t}) = \text{softmax}\left[s_{\mathcal{M}}(X_t \mid x_{<t}) + s_{\mathcal{M}^+}(X_t \mid x_{<t}) - s_{\mathcal{M}^-}(X_t \mid x_{<t})\right]$$

思考

论文公式能成立，我觉得是基于假设：小规模的专家和反专家的logit差异能够捕捉tuning的“方向”，并且这种方向适用于其他llm。 那么这个假设是否成立呢?
即使假设成立，应用proxy-tuning也是有代价的，先不说效果会降低多少，首先是在inference时涉及3个llm，效率会降低，其次本文暗含目标llm和专家/反专家llm的vocabulary要相同，这样才能修正logits。