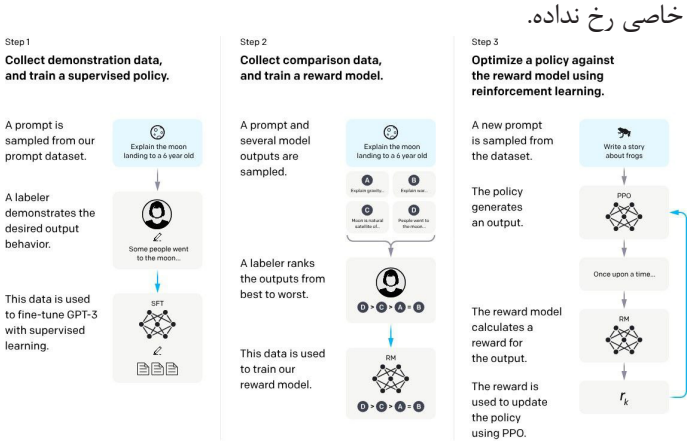
مکانیزم RLHF؛ نمایش زیبایی از تعامل انسان و ماشین

محمدرضا اسماعیلیان
مهندس ارشد هوش مصنوعی
شرکت باسلام استاد قم
mrz.esmln@gmail.com

احتمالا برای شما هم این سوال بوجود اومده باشه که ChatGPT چطوری بوجود اومده و با بقیه مدل‌های زبانی قبلی چه تفاوتی داره؟ برای پاسخ به این سوال باید اول به این سوال پاسخ داد که Open-AI دقیقا «با چه سوال و چه نیازی» به GPT-۳ راضی نشد و به chatGPT رسید؟

بطور خلاصه پاسخ اینه که پاسخ‌ها و تولیدات GPT-۳ با ترجیحات انسانی هم‌راستا (Align) نبود. در واقع GPT-۳ از روی Text موجود در اینترنت آموزش دیده بود. برای ساخت معماری این شبکه از قسمت Decoder معماری Transformer استفاده کردند؛ پس تسک این بود که وقتی یه جمله بهش میدی کلمه یا کلمات بعدی رو حدس بزنه. اما با این ساختار آموزشی و این نوع دیتا هیچ تضمینی وجود نداشت که اون جملاتی که در ادامه Predict می‌کنه لزوما دارای حقیقت باشه (Truthfulness) یا جملات سمی و توهمی یا حتی توهین آمیز نباشه. این اولین نیاز بود. نیاز دوم این بود که بتونه دستور و خواسته‌ای که یوزر از طریق ورودی میده رو متوجه بشه و چیزی رو که یوزر میخواد رو تولید کنه. یعنی ساختار آموزش بجای «بقیه‌اش رو تو بگو» به ساختار ارباب رعیتی «این کاری که میگم رو بکن» تبدیل بشه. بطورکلی این دو نیاز ذیل تسک Alignment تعریف می‌شه و برای هم‌راستا کردن تولیدات مدل با ترجیحات انسانی تسک Alignment رو انجام می‌دن.

برخلاف تصور، ChatGPT مستقیم از روی GPT-۳ ایجاد نشده. بلکه از نظر Open-AI راه ChatGPT از fine-tune کردن InstructGPT می‌گذشته. در واقع GPT-۳ رو با اصلاح ساختار آموزش و ارائه یک روش آموزشی نوظهور، fine-tune کردند و InstructGPT به دنیا اومده. و بعد از این مدل به ChatGPT رسیدند. جالب اینجاست که اصل زیبایی‌های خلقت توی InstructGPT جمع شده. و از InstructGPT تا ChatGPT خیلی مسایل فنی خاصی رخ نداده.



برای ساخت InstructGPT در اولین مرحله اومدن روی مدل GPT-۳ تسک Supervised fine tuning یا (SFT) رو انجام دادند. یعنی اومدن تمام Promptهایی که ملت روی GPT-۳ داشتند رو به آدم‌ها دادند و ازشون خواستند پاسخ این Prompt رو بنویسند. و بعد با کمک این سوال و جواب‌ها مدل رو ذیل تسک SFT فاین‌تیون کردند. این اولین مرحله‌ی اون مکانیزمی است که در عنوان بهش اشاره کردم. یعنی مکانیزم RLHF. «یادگیری تقویتی با بازخورد انسان» یا همون Reinforcement Learning Human Feedback. بعد از انجام مرحله‌ی اول، مرحله‌ی دوم به این صورت انجام می‌شه:

مرحله دوم: اول به ازای هر Prompt، از مدلی که در مرحله‌ی اول فاین‌تیون شده چندین خروجی می‌گیریم. این خروجی‌ها را با تنظیمات مختلف می‌گیریم که خروجی‌های متفاوتی داشته باشیم. و یا در مرحله‌ی اول چندین مدل با معماری متفاوت فاین‌تیون می‌کنیم و در این مرحله از همه این مدل‌ها خروجی می‌گیریم تا خروجی‌های متفاوت داشته باشیم. خروجی‌ها رو به انسان می‌دیم تا برامون از بهترین تا بدترین جواب Sort کنه. (در اینجا Promptها، Requestهایی هستند که مردم روی API مدل GPT-۳ زدند). و بعد از طریق این دیتای باارزش (ترتیب بندیِ نتایج مدل‌ها بر اساس ترجیح انسان) یک Reward Model توسعه می‌دیم. در واقع اینجا با این مدل داریم اون Functionی رو مدل می کنیم که معمولا یا Rule Based بود یا انسان.

مرحله سوم: در این مرحله می‌آیم مدل فاین‌تیون شده در مرحله‌ی اول رو تبدیل به یک مدل RL می کنیم. و به ازای هر Prompt در دیتابیس ازش خروجی می گیریم. خروجی رو میدیم به Reward Model و از Reward محاسبه شده برای آپدیت Policyهای مدل استفاده می کنیم.

حقیقتا روش زیبایی است. بنظرم یک نکته مهم این روش در اینه که کار یدی و کار علمی-مهندسی در یک تعادل جذابی قرار داره. از یه طرف تبدیل کردن یه مدل زبانی به یک مدل RL بنظر خیلی خفن میاد و احتمالا بیشتر در آینده شاهدش باشیم. از طرفی، جایی که تصور نمی شد انسان حضور داشته باشه، از انسان استفاده شد (یعنی مرحله‌ی دوم). و در آخر هم با Reward Model زیبایی رو بر ما تمام کردند و در جایی که حضور انسان یا Rules پذیرفته شده بود اثبات کردند میشه مدلی ساخت که ترجیحات انسان ها رو مدل کرد و خلاصه که با RLHF نمایش زیبایی از تعامل انسان و ماشین رقم زدند.

در ChatGPT این مساله رو حل کردند که چطوری نحوه ارتباط کاربر با مدل حالت تعاملی بگیره (اصلاح پاسخ های قبلی و..) که بطور کلی تغییرات ریزی روی Reward Model دادند. ضمن اینکه چالش های Iterative Deployment رو برای تصحیح اشتباهات و همچنین ایمن کردن مدل از ورودی های سمی کاربران تاحدی حل کردند.

برای مطالعه بیشتر:

https://openai.com/blog/instruction-following/#fn۱

https://arxiv.org/abs/۲۲۰۳.۰۲۱۵۵

https://openai.com/blog/deep-reinforcement-learning-from-/human-preferences