



دانشگاه تهران

دانشکده علوم و فنون نوین

گزارش هفته پنجم و ششم

نام و نام خانوادگی	فاطمه چیت ساز
شماره دانشجویی	830402092
تاریخ ارسال گزارش	25 دی 1402

Contents

1.....	week to stronge learner	برسی مقاله
9.....	lets verify step by step	برسی مقاله

برسی مقاله week to strong learner

قضیه این مقاله از اینجا شروع میشه که هوش مصنوعی دارن زیادی باهوش میشن و خب الان reward model ها اینطورین که ادم ها بر آنها نظاره میکنن اما حالا فکر کنید که مثلا چهار روز دیگه هوش مصنوعی اونقدر قوی شد که برامون هزار خط کد نوشت الان کدوم ادمی حال داره دقیق نگا کنه کده اوکیه یا نه یا حتی نگا کنه آیا ادم میتونه مثلا یک باگ امنیتی رو تو کد پیدا کنه؟ حرفم اینه کم کم ادم دیگه نمیتونن روی مدلا نظارت کنن و خب ی چیزی باید بسازیم که بتونه به یوزر امنیت بده که اره من قرار نیست کارای خطرناک بکنم

خلاصه دوتا هدف داریم یک این که با ما هوش خودمون نیایم اون هوش مصنوعیه رو سطحشو بیاریم پایین و خب در واقع میشه دقت هوش مصنوعی

نکته دوم اینه که ی چی برای امنیت قضیه داشته باشیم و ی جور یایی بتونیم اون هوش مصنوعیه رو align کنیم که دیگه اونقدر وحشناک نشه

خلاصه اینطور که بوش میاد ما ی reward model جدید نیاز داریم

راه حل open ai برای این قضیه چیه اینکه فرض کنید ما دوتا مدل داریم یکی قوی یکی ضعیف مدل ضعیفه مثلا چیزیه که ما بهش اعتماد داریم و قویه اونیه ک ما میخوایم براش align تعیین کنیم اگر که ما بیایم label های تولید شده توسط مدل ضعیف رو بدیم به مدل قوی شاید بتونیم ی align بذاریم روی قضیه در واقع ما finetuning میکنیم مدل قوی رو با label های مدل ضعیف

در واقع ما اینطوری میتونیم اون چیزایی که برامون مهمه رو برای مدل بزرگ مشخص کنیم و بگیم اره اقا اینا حد و حدود ماعه

Open ai بزرگوار اومده اینا رو روی چندتا چیز تست کرده

(NLP) benchmarks, chess puzzles, and our internal ChatGPT reward modeling dataset

اولی که یک مشت تسک nlp معروفه دومی یک مسئله شطرنجه که ی موقعیت از وضعیت مهره های شطرنج میدی میگو بهترین حرکت بعدی چیه

بعدیم اینطوریه ک ما بیایم ی reward model برای chat gpt بسازیم بر این اساس

حالا بسم الله چندتا کار باید بکنیم

1. مدل قویمونو آموزش بدیم و تست کنیم و دقت به دست بیاریم
2. مدل ضعیفو آموزش بدیم و دقت به دست بیاریم
3. مدل قوی رو با لیبل های مدل ضعیف آموزش بدیم و تست کنیم و دقت بدست بیاریم
4. حالا ایا ی گپی بدست میاد بین مدل قوی با لیبل های اصلی و مدل قوی با لیبل های مدل ضعیف
5. متر ما میشه این گپه باید کمش کنیم

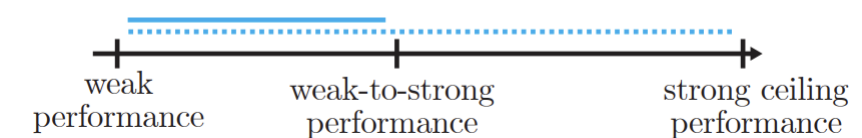
این دوستان ایزی تونستن تو مرحله اول نصف گپ رو پر کنن
 برای تسک های nlp هم با مشقت فراوان تونستن هشتاد درصد گپ رو پر کنن
 اینجا مدل قویشون جی پی تی چهاره مدل قوی جی پی تی دو
 ی گرفتاری که موجوده اینه که اینجا برای هر مسئله ی سری تنظیمات انجام دادن یعنی مثلا به
 یک روش کلی نرسیدن که بگن به به برای همه گل و بلبله
 نه اینطوری نی برای هر کدوم از این سه مرحله برای بهتر شدنش ی سری تنظیمات دادن
 و اینکه هنوزم نتونستن همه گپ رو پر کنن و وضعیت زیاد فرح بخش نی
 کارهای مشابه این تحقیق اونقدر زیاد نیست ولی ی کارهایی هست که شاید کمک کنه مثل مسئله
 لیبل هایی که همیشه بهشون اعتماد زیاد کرد (trained using unreliable labels)
 چون مام اینجا ممکنه این مدل ضعیفمون اوسکول بشه و جواب غلط بده به مدل قوی و مدل قوی
 بنده خدا گیج بشه
 بحث بعدی مسئله نویز و از بین بردن نویزه راه حل زیبای بعدی bootstrapping عه که باهاش کار
 داریم
 نگا لیبل های ما الان یک سریشون برای مدل ضعیف ایزی هستن یک سریشون نه براش سختن در
 واقع پس میشه به مسئله ی جورایی به چشم semi-supervised نگا کرد
 داستان بعدی کارایی هست که در مورد مدل های معلم دانش آموز طور هست که البته اکثرشون
 معلمه باهوش تره و دانش آموزه کم علم تر

مسئله بعدی اینه ک وقتی مسئله ما حال semi-supervised میگیره میشه از confidence auxiliary loss استفاده کرد

البته ی گرفتاری داریم باز اونم اینه که در حذف نویز معمولا میدونن نویز کجاش ما حتی نمیدونیم نویز کجا هست

موضوع جالب بعدی Eliciting Latent Knowledge (ELK) عه که میاد دانش نهفته در یک شبکه عصبی رو میکشه بیرون در واقع اگ بدونیم توش چ خبره شاید بتونیم ی align بذاریم روش

خب حالا یک دیدی گرفتیم مترمون هم که مشخص شد همون گپه هست که میخوایم کمش کنیم حالا اگر یکم ریاضی طوری به قضیه نگا کنیم مترمون اینطوری میشه :

$$PGR = \frac{\text{weak-to-strong} - \text{weak}}{\text{strong ceiling} - \text{weak}} = \frac{\text{---}}{\text{.....}}$$


در واقع وقتی pgr یک باشه یعنی گل کاشتیم

البته باز گرفتاری هایی هست که ما الان تنظیمات درست و قشنگ نداریم و اینکه الان نمیدونیم این مدلای سوپر هیومن قراره چه ریختی باشن قراره چه کنن و قراره چه حرکاتی از خودشون داشته باشن

حالا بیاید یکم مسائلمون رو خوشگل تر بشناسیم

مسئله nlp ها که بیست و دو تا تسک معروف nlp است که تبدیل میکننش به فیچر های باینری و خروجی هم ی حال باینری ای داره

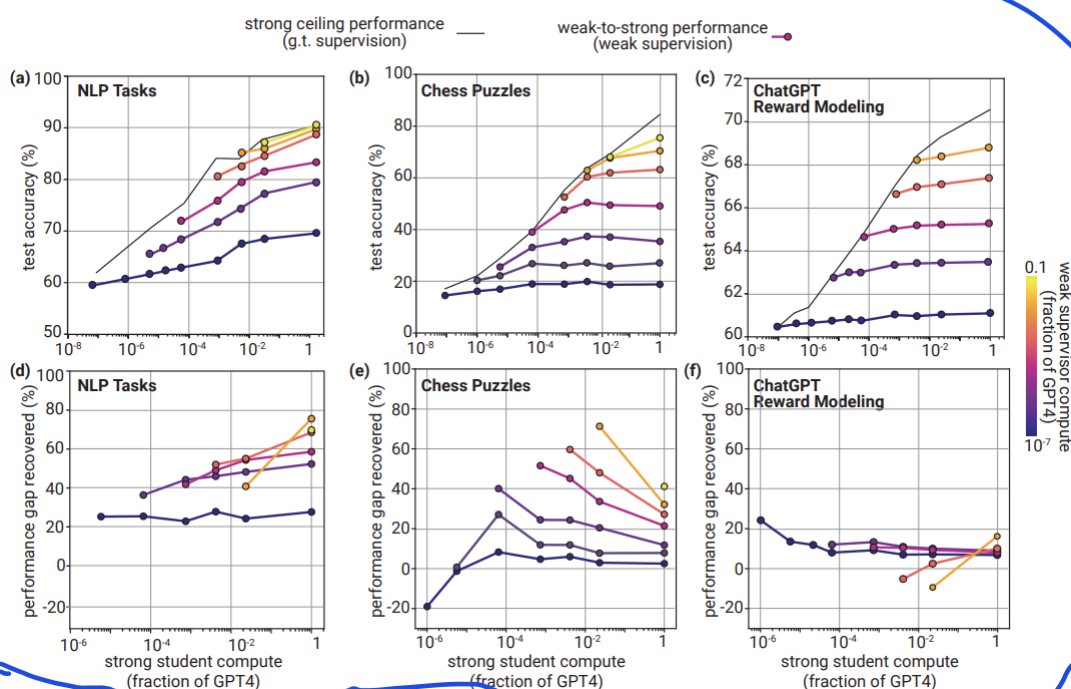
مسئله شطرنجه هم بهت ی صفحه شطرنج میدن با ی سری مهره و میگن اقا بهترین حرکت چیه اینجا گفته با 0 temperature انجام دادیم قضیه رو که من هیچ ایده ای ندارم چیه

مسئله بعدی ChatGPT reward modeling که ما باید بیایم دقت reward model مون رو ببریم بالا و ترجیحات یوزر رو بفهمیم

حالا چندتا قضیه وجود داره یک اینکه همیطوری ساده بیایم لیبل های رو بدیم به مدل قوی که برای تسک های nlp این کارو انجام دادیم بود ک pgr ها معمولا خوب بود

اگر مدل ضعیفمون رو قوی یا مدل قویمونو بزرگ کنیم pgr میره بالا برای این nlp تسک ها برای شطرنجه اولش که pgr خیلی داغونه و صفره برای این اینطوریه که هر چیه فاصله بین معلم و دانش آموز زیاد میشه وضعیت داغون تر میشه

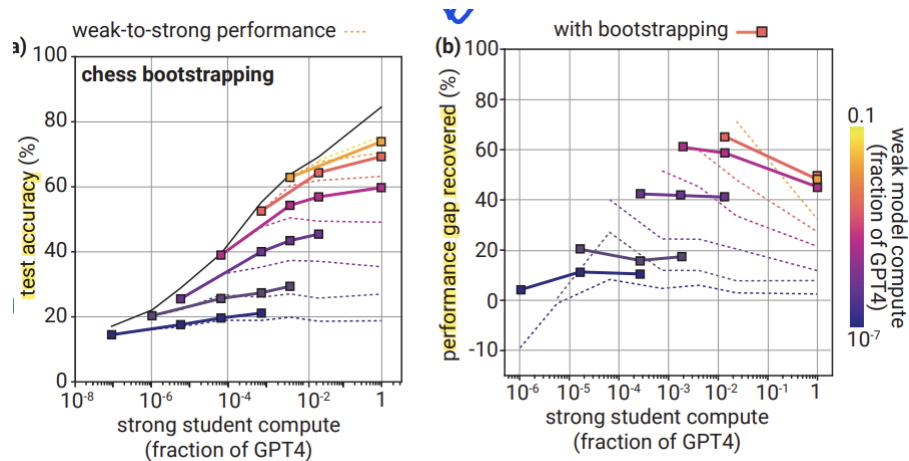
یعنی وقتی بیایم مدل قوی رو بزرگ تر کنیم و مدل ضعیف رو نگه داریم pgr کمتر میشه



حالا این روش ساده بود اگر بخوایم قضیه رو خفن کنیم و pgr رو برسونیم به یک چندتا روش وجود داره

یک اینکه BOOTSTRAPPING انجام بدیم به این صورت که به جای یک مدل ضعیف و یک کدل قوی مجموعه ای از مدل ها داشته باشیم مثلاً ی مدل ضعیف بیاد به مدل ضعیف قوی تر لیبل بده ضعیف قوی به ضعیف قوی تر تا مدل قوی خدا

این برای اون مسئله شطرنجه جوابه ولی بقیه نه

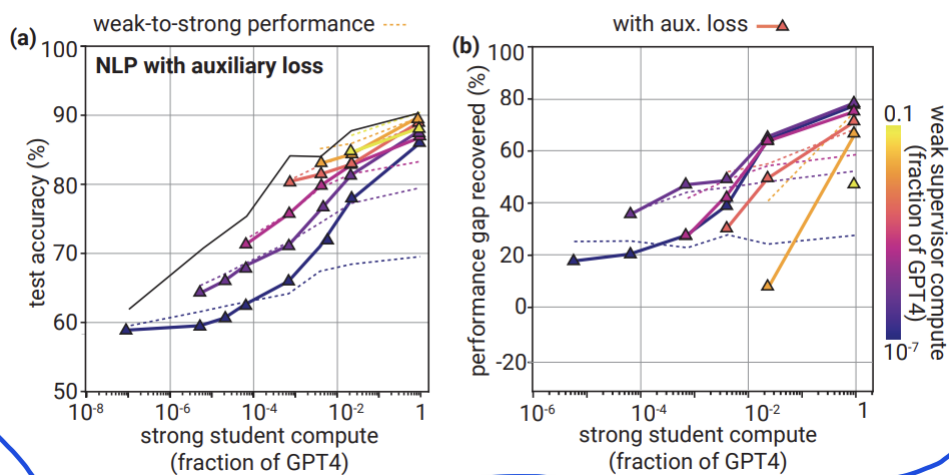


راه بعدی AUXILIARY CONFIDENCE LOSS است یعنی مدل ضعیف ما ممکنه ارور تولید کنه و این اروره بره به مدل قوی هم منتقل شه پس این مدل قوی همه جا هم نباید حرف گوش کن باشه

میان این confidence loss رو اضافه میکنن به cross entropy

گرفتاری: وقتی گپ بین مدل قوی و ضعیف کمه یا inverse scaling جواب نی

برای تسک های nlp جوابه این روش



گرفتاری ها:

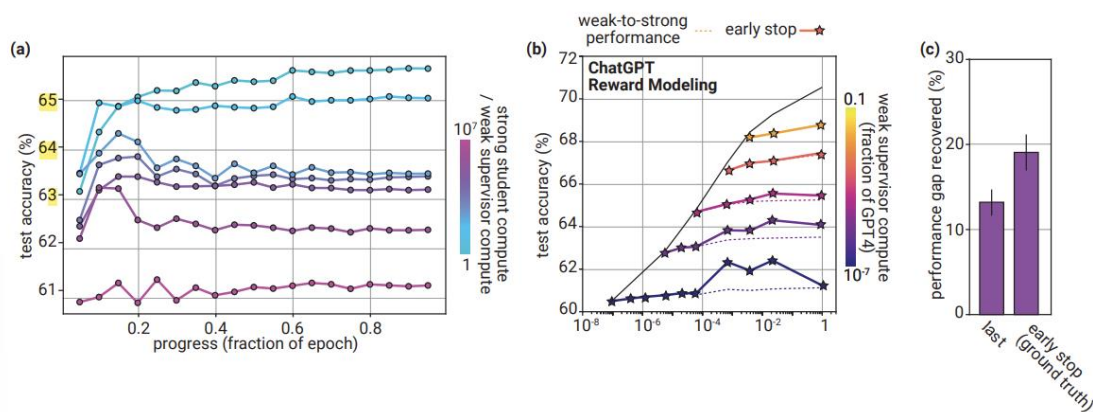
یک OVERFITTING TO WEAK SUPERVISION

وقتی مدل قوی میاد ارور های مدل ضعیف رو بر میداره و سعی میکنه خودش هم ضعیف شه
راه حل :

یک اینکه آموزش رو زود تموم کنیم (early stop)

حالا میشه early stop را بر اساس لیبل های درست انجام داد که بهش میگن cheats

اما خب وضعیت رو بهتر میکنه برای gpt reward model



ی متر دیگه میتونه این باشه مدل قوی چقدر با مدل ضعیف موافقه (AGREEMENT) در واقع وقتی
کاملا موفق باشه pgr

میشه صفر

در واقع confidence loss میاد این AGREEMENT رو کم میکنه

حالا ای چیز جالب فهمیدن که مدلای قوی اونقدر حرف گوش کن نیستن یعنی AGREEMENT شون
کمتره

ی نکته ای هم هست ما باید ببینم نوع ارور در weak supervision چیه و بر اساس اون سعی کنیم
مشکلات رو حل کنیم

موضوع بعدی SALIENCY است

قضیه اینه که مدلای قوی ما انقدر قوی هستند که بتونن zero-shot عمل کنن یعنی در واقع شاید این لیبل های مدل ضعیف اونقد به کارشون نیاد

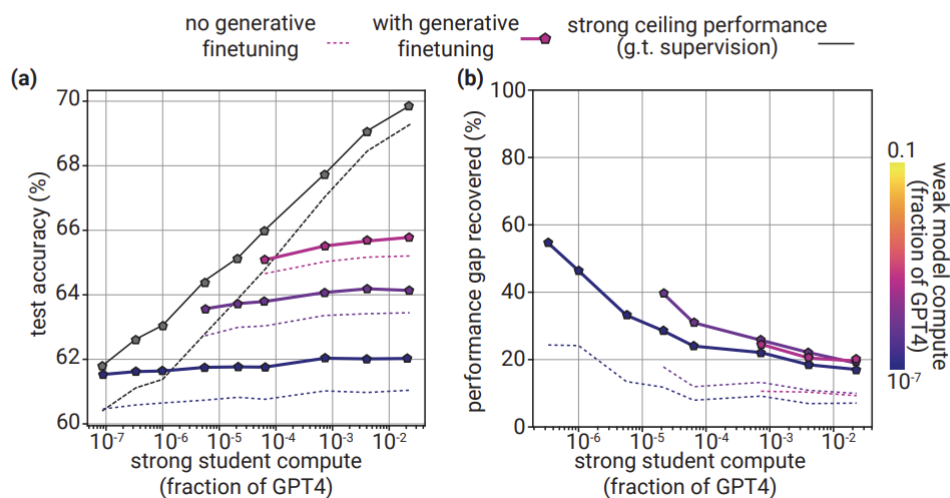
به نظر من این میتونه چیز خوبی باشه وقتی که لیبل های مدل ضعیف خرابن ولی چیز بدی باشه برای اینکه ما چطورییی اعتماد کنیم به این مدل قویه پس: /

در واقع اینجا یک بحثی مطرح میشه که به جای اینکه لیبل مدل ضعیف رو بدیم به مدل قوی بیایم به عنوان prompt

یعنی مثلا فرض کن ما بیایم مدل زبانیمون رو با ریویو ها انلاین finetuning کنیم خب مدل بیشتر میره سمت احساسات

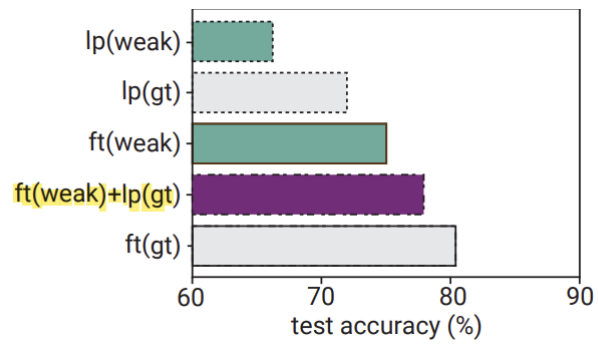
البته open ai گفته این موضوع براش فعلا خطرناکه و نرفته سمش

اما به نظر من خیلی جالب ترههه



در واقع کلا سمت generative finetuning رفتن میتونه قضیه رو باحال کنه

ی موضوع جالب دیگه هم هست اینکه اوکی با لیبل های مدل ضعیف ترین کنیم مدل قوی رو ولی بعدش با لیبل های اصلی و یک مدل خطی بیایم مدل رو بهتر کنیم



حالا ریپو اصلیش اینا رو به ما نداده که ولی این مدلا رو داده

لیست مدلا:

gpt2

gpt2-medium

gpt2-large

gpt2-xl

Qwen/Qwen-1_8B

Qwen/Qwen-7B

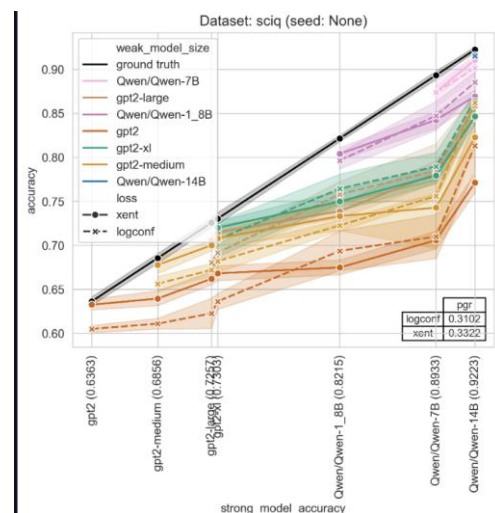
Qwen/Qwen-14B

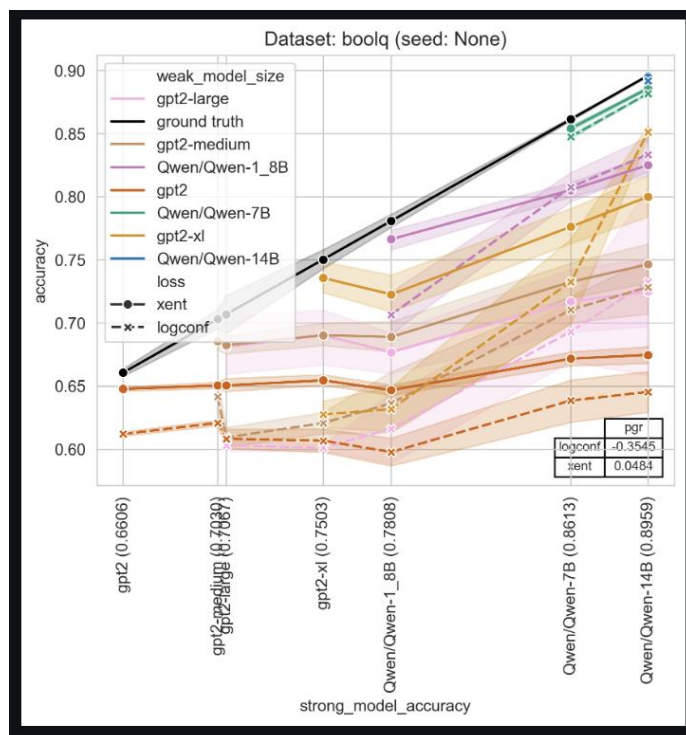
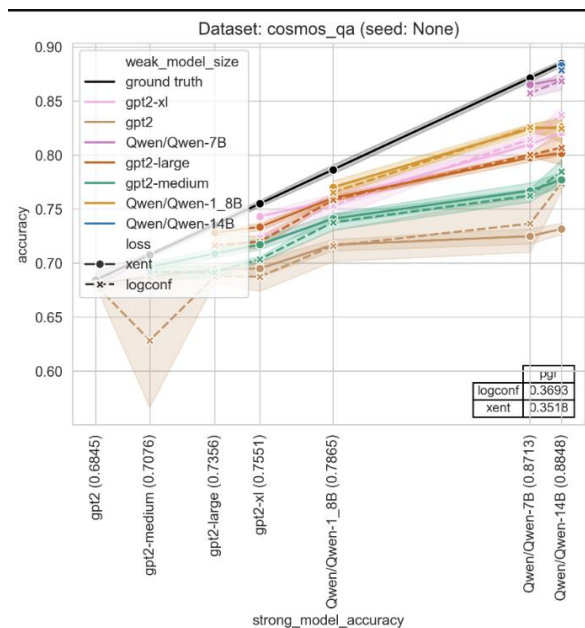
Qwen/Qwen-72B

لیست دیتاست ها:

"amazon_polarity," "sciq," "anthropic_hh," "cosmos_qa," and "boolq"

نتایج :





بررسی مقاله step by step lets verify

تو این مقاله میخوایم به نگاه عمیق تر به مدل های یادگیری تقویتی (Reinforcement Learning) بپردازیم و ببینیم چطور با دو روش مختلف آموزششون بهتره: روش اول نگاه به نتیجه ی نهاییه (Outcome Supervision)، روش دوم هم چک کردن تک تک قدم های حل مساله (Process Supervision).

تو این مقاله به لیست کار مرحله به مرحله هم به شکل کد شبه (Pseudocode) برای انجام این تحلیل بهتون میدیم.

روش‌ها:

دو محیط داریم: مقیاس بزرگ و مقیاس کوچک. هر کدومش به جور خوبه و اطلاعات بیشتری بهمون میده.

مقیاس بزرگ:

یه مدل قوی (GPT-4 از OpenAI، سال 2023) داریم که روش اول (نگاه به نتیجه) و دوم (چک کردن تک‌تک قدم‌ها) رو روش پیاده می‌کنیم تا بهترین مدل‌های ممکنه رو بسازیم.

مقیاس کوچک:

مدل‌هایی رو اینجا آموزش میدیم که بشه راحت‌تر مقایسه‌شون کرد. برای این کار از مدل‌های بزرگتر کمک می‌گیریم تا نحوه‌ی یادگیری مدل‌های کوچیک‌تر رو کنترل کنیم و بتونیم آزمایش‌های بیشتری انجام بدیم. حوزه بررسی:

یه مدل ثابت به اسم ژنراتور داریم که تو هر دو مقیاس، جواب‌های مختلف به مسائل ریاضی میده. هدف اصلی ما آموزش دقیق‌ترین مدل پاداشه که بتونه جواب‌های درست رو از بین جواب‌هایی که ژنراتور به صورت تصادفی بهش میده (Best-of-N Search) پیدا کنه.

مدل‌های پایه:

مدل‌های مقیاس بزرگ همگی از مدل OpenAI GPT-4، 2023) بهتر شده‌اند.

مدل‌های پایه مقیاس کوچک شبیه GPT-4 طراحی شدن ولی با قدرت محاسباتی خیلی کمتر (حدود 200 برابر کمتر) آموزش دیدن.

همه مدل‌ها رو روی یه مجموعه داده حدود 1.5 میلیارد توکن مرتبط با ریاضی (MathMix) آموزش میدیم.

شبه‌کد الگوریتمی: لیست کار گام به گام

محیط‌های مقیاس بزرگ و کوچک رو آماده کن.

برای هر محیط:

مدل‌ها رو از GPT-4 (OpenAI), 2023) تنظیم دقیق کن.

بهترین مدل‌های پاداش (ORM و PRM) رو با هر دو روش آموزش بده.

از مدل مقیاس بزرگ برای کنترل آموزش مدل مقیاس کوچک استفاده کن.

آزمایش‌های مختلف رو تو مقیاس کوچک انجام بده.

برای هر مقیاس مدل:

از ژنراتور برای درست کردن همه‌ی جواب‌ها استفاده کن.

بهترین مدل پاداش رو با جستجوی بهترین گزینه از بین جواب‌های تصادفی ژنراتور پیدا کن.

عملکرد مدل پاداش رو با این روش ارزیابی کن.

همه مدل‌ها رو روی مجموعه داده MathMix تنظیم دقیق کن.