LLM as Optimizers and Mesa Optimizers Introduction

Nasy

Oct 13, 2023

Outline

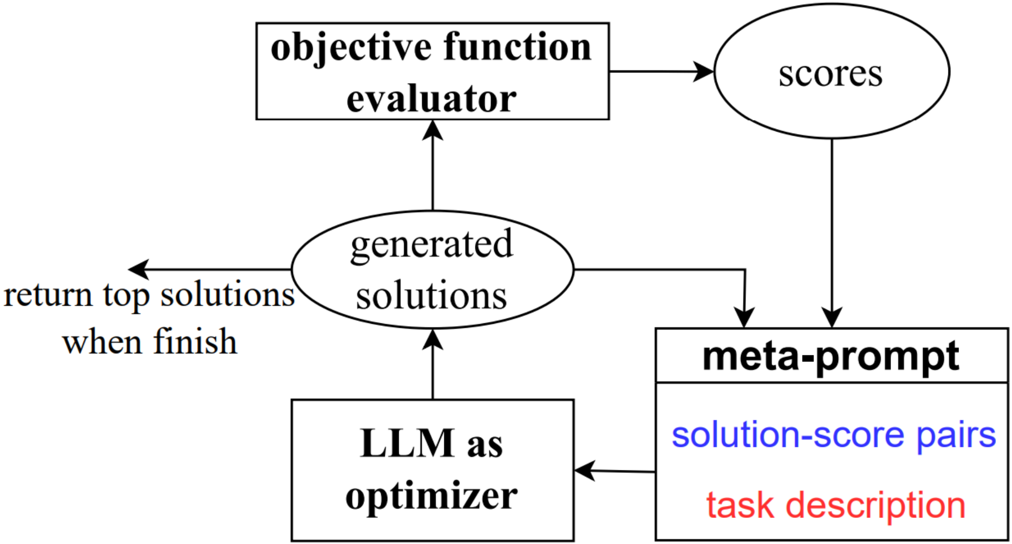
Section

Introduction

- What is LLM as Optimizers?
  - Optimization by PROmpting (OPRO)
  - The goal is to find instructions that maximize the task accuracy.
- With a variety of LLMs, we demonstrate that the best prompts optimized by OPRO outperform human-designed prompts by up to 8% on GSM8K, and by up to 50% on Big-Bench Hard tasks.

Section

Optimization by PROmpting (OPRO)



Example

You are given a list of points with coordinates below: (0): (-4, 5), (1): (17, 76), (2): (-9, 0), (3): (-31, -86), (4): (53, -35), (5): (26, 91), (6): (65, -33), (7): (26, 86), (8): (-13, -70), (9): (13, 79), (10): (-73, -86), (11): (-45, 93), (12): (74, 24), (13): (67, -42), (14): (87, 51), (15): (83, 94), (16): (-7, 52), (17): (-89, 47), (18): (0, -38), (19): (61, 58).
Below are some previous traces and their lengths. The traces are arranged in descending order based on their lengths, where lower values are better.

<trace> 0,13,3,16,19,2,17,5,4,7,18,8,1,9,6,14,11,15,10,12 </trace>
length:
2254

<trace> 0,18,4,11,9,7,14,17,12,15,10,5,19,3,13,16,1,6,8,2 </trace>
length:
2017

<trace> 0,11,4,13,6,10,8,17,12,15,3,5,19,2,1,18,14,7,16,9 </trace>
length:
1953

<trace> 0,10,4,18,6,8,7,16,14,11,2,15,9,1,5,19,13,12,17,3 </trace>
length:
1840

Give me a new trace that is different from all traces above, and has a length lower than any of the above. The trace should traverse all points exactly once. The trace should start with <trace> and end with </trace>.

Motivation

- Making use of natural language descriptions
    - The main advantage of LLMs for optimization is their ability of understanding natural language, which allows people to describe their optimization tasks without formal specifications.
- Trading off exploration and exploitation
    - LLM should be able to exploit promising areas of the search space where good solutions are already found, while also exploring new regions of the search space so as to not miss potentially better solutions.

Meta-Prompt Design

- Optimization problem description
  - e.g., generate a new instruction that achieves a higher accuracy
  - e.g., the instruction should be concise and generally applicable
- Optimization trajectory
  - The optimization trajectory includes past solutions paired with their optimization scores, sorted in the ascending order.

Solution Generation

- Optimization stability.
  - Prompt the LLM to generate multiple solutions at each optimization step, allowing the LLM to simultaneously explore multiple possibilities and quickly discover promising directions to move forward.
- Exploration-exploitation trade-off
  - Tune the LLM sampling temperature to balance between exploration and exploitation. A lower temperature encourages the LLM to exploit the solution space around the previously found solutions and make small adaptations, while a high temperature allows the LLM to more aggressively explore solutions that can be notably different.

Section

Mathematical Optimization

Now you will help me minimize a function with two input variables w, b. I have some (w, b) pairs and the function values at those points. The pairs are arranged in descending order based on their function values, where lower values are better.

input:
w=18, b=15
value:
10386334

input:
w=17, b=18
value:
9204724

Give me a new (w, b) pair that is different from all pairs above, and has a function value lower than any of the above. Do not write code. The output must end with a pair [w, b], where w and b are numerical values.

Mathematical Optimization

Table 2: Linear regression by optimizer LLMs: the mean $\pm$ standard deviation of the number of steps and the number of unique (w, b) pairs explored before reaching the global optima. Both w and b start from 5 random starting points in $[10, 20]$. We use temperature 1.0 for all models. We run each setting 5 times. The starting points are the same across optimizer LLMs but are different across 5 runs, and are grouped by: within the starting region, outside and close to the starting region, and outside and farther from the starting region. Bold numbers indicate the best among three LLMs in each setting.

w_{true}	b_{true}	number of steps			number of unique (w, b) pairs explored		
		`text-bison`	`gpt-3.5-turbo`	`gpt-4`	`text-bison`	`gpt-3.5-turbo`	`gpt-4`
15	14	5.8 ± 2.6	7.6 ± 4.5	**4.0** ± 1.5	40.0 ± 12.4	36.0 ± 15.2	**17.2** ± 5.1
17	17	**4.0** ± 1.8	12.6 ± 6.0	6.0 ± 3.7	33.4 ± 11.7	53.8 ± 16.9	**26.0** ± 10.6
16	10	**3.8** ± 2.2	10.4 ± 5.4	6.2 ± 3.1	30.2 ± 13.4	42.8 ± 16.3	**24.2** ± 8.2
3	5	**9.8** ± 2.8	10.8 ± 2.7	12.2 ± 2.0	55.8 ± 16.1	39.6 ± 10.1	**33.0** ± 4.0
25	23	19.6 ± 11.4	26.4 ± 18.3	**12.2** ± 3.7	104.0 ± 52.3	78.6 ± 26.2	**44.2** ± 8.3
2	30	**31.4** ± 6.3	42.8 ± 9.7	38.0 ± 15.9	126.4 ± 17.7	125.6 ± 21.7	**99.0** ± 24.6
36	-1	**35.8** ± 6.4	45.4 ± 16.9	50.4 ± 18.8	174.0 ± 28.2	142.2 ± 31.2	**116.4** ± 32.7

Limitations

Limitations. We would like to note that OPRO is designed for neither outperforming the stateof-the-art gradient-based optimization algorithms for continuous mathematical optimization, nor surpassing the performance of specialized solvers for classical combinatorial optimization problems such as TSP. Instead, the goal is to demonstrate that LLMs are able to optimize different kinds of objective functions simply through prompting, and reach the global optimum for some smallscale problems.

Section

Prompt Optimization

I have some texts along with their corresponding scores. The texts are arranged in ascending order based on their scores, where higher scores indicate better quality.

text:
Let's figure it out!
score:
61

text:
Let's solve the problem.
score:
63

(... more instructions and scores ...)

The following exemplars show how to apply your text: you replace <INS> in each input with your text, then read the input and give an output. We say your output is wrong if your output is different from the given output, and we say your output is correct if they are the same.

input:
Q: Alannah, Beatrix, and Queen are preparing for the new school year and have been given books by their parents. Alannah has 20 more books than Beatrix. Queen has 1/5 times more books than Alannah. If Beatrix has 30 books, how many books do the three have together?
A: <INS>
output:
140

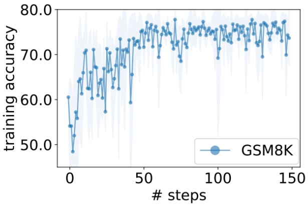
(... more exemplars ...)

Write your new text that is different from the old ones and has a score as high as possible. Write the text in square brackets.

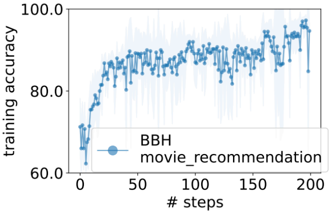
Prompt Optimization Design

- Optimization problem examples.
- Optimization trajectory
- Meta-instructions

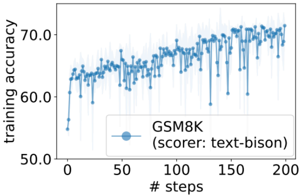
Results



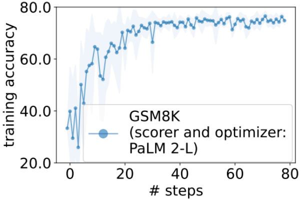
(a) GSM8K



(b) BBH movie_recommendation



(a) `PaLM 2-L-IT` optimizer



(b) pre-trained `PaLM 2-L` optimizer

Results

Table 4: Test accuracies on GSM8K. We show the instruction with the highest test accuracy for each scorer-optimizer pair.

Scorer	Optimizer / Source	Instruction position	Top instruction	Acc
Baselines				
PaLM 2-L	(Kojima et al., 2022)	A_begin	Let's think step by step.	71.8
PaLM 2-L	(Zhou et al., 2022b)	A_begin	Let's work this out in a step by step way to be sure we have the right answer.	58.8
PaLM 2-L		A_begin	Let's solve the problem.	60.8
PaLM 2-L		A_begin	(empty string)	34.0
text-bison	(Kojima et al., 2022)	Q_begin	Let's think step by step.	64.4
text-bison	(Zhou et al., 2022b)	Q_begin	Let's work this out in a step by step way to be sure we have the right answer.	65.6
text-bison		Q_begin	Let's solve the problem.	59.1
text-bison		Q_begin	(empty string)	56.8
Ours				
PaLM 2-L	PaLM 2-L-IT	A_begin	Take a deep breath and work on this problem step-by-step.	**80.2**
PaLM 2-L	PaLM 2-L	A_begin	Break this down.	79.9
PaLM 2-L	gpt-3.5-turbo	A_begin	A little bit of arithmetic and a logical approach will help us quickly arrive at the solution to this problem.	78.5
PaLM 2-L	gpt-4	A_begin	Let's combine our numerical command and clear thinking to quickly and accurately decipher the answer.	74.5
text-bison	PaLM 2-L-IT	Q_begin	Let's work together to solve math word problems! First, we will read and discuss the problem together to make sure we understand it. Then, we will work together to find the solution. I will give you hints and help you work through the problem if you get stuck.	64.4
text-bison	text-bison	Q_end	Let's work through this problem step-by-step:	**68.5**
text-bison	gpt-3.5-turbo	Q_end	Analyze the given information, break down the problem into manageable steps, apply suitable mathematical operations, and provide a clear, accurate, and concise solution, ensuring precise rounding if necessary. Consider all variables and carefully consider the problem's context for an efficient solution.	66.5
text-bison	gpt-4	Q_begin	Start by dissecting the problem to highlight important numbers and their relations. Decide on the necessary mathematical operations like addition, subtraction, multiplication, or division, required for resolution. Implement these operations, keeping in mind any units or conditions. Round off by ensuring your solution fits the context of the problem to ensure accuracy.	62.7

Results BBH

See paper p15.

Suggestions

https://github.com/AGI-Edgerunners/LLM-Optimizers-Papers

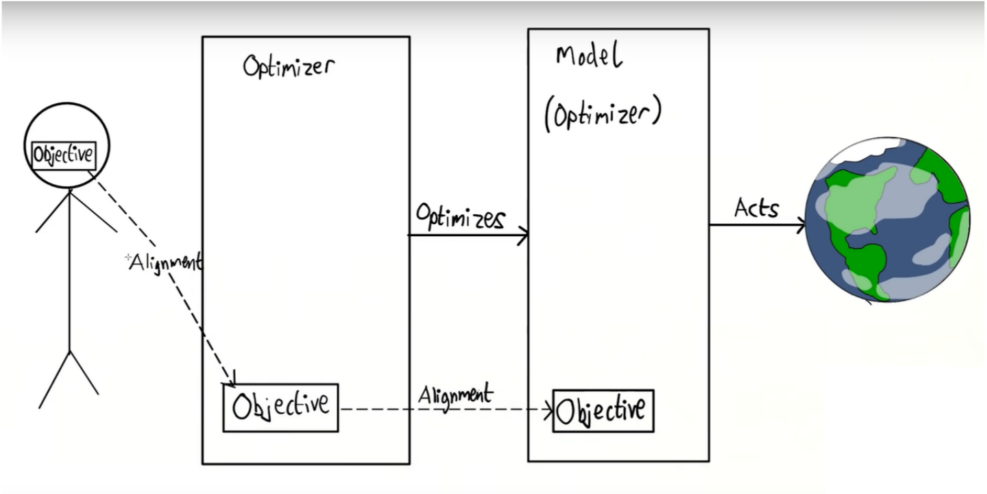
Conclusion

- Optimization by PROmpting (OPRO)
  - The goal is to find instructions that maximize the task accuracy.
- Motivation
  - Making use of natural language descriptions
  - Trading off exploration and exploitation
- Design
  - Optimization problem description
  - Optimization trajectory
  - Meta-instructions
- Results

Introduction

- Why?
  - Transformers have become the dominant model in deep learning, but the reason for their superior performance is poorly understood
- How?
  - By reverse engineering a series of autoregressive Transformers trained on simple sequence modeling tasks, the authors reveal the gradient-based mesa-optimization algorithm that drives prediction generation.
- New?
  - Propose a novel self-attention layer, the mesa-layer, that explicitly and efficiently solves optimization problems specified in context

Mesa-Optimization



Mesa-Optimization

- An optimizer (like gradient descent, or evolution) produces another optimizer (like complex AIs, or humans). When this happens, the second optimizer is called a mesa-optimizer.
- Recently, This phenomenon has been recently termed mesa-optimization: minimizing a generic autoregressive loss gives rise to a subsidiary gradient-based optimization algorithm running inside the forward pass of a Transformer.

Mesa-Optimization

- We might train a neural network to play a game using a gradient descent algorithm as our base optimizer. However, after many iterations, the neural network might develop some strategy or heuristic for playing the game. This strategy or heuristic can be thought of as a secondary optimization process or mesa optimizer.
- An essential consideration is that the objective of the mesa optimizer might not perfectly align with that of the base optimizer. For instance, while the base optimizer's objective might be to have the neural network perform well on training data, the mesa optimizer might be more concerned with quickly achieving rewards without considering long-term consequences. This misalignment can lead to unpredictable or undesired behaviors.

Mesa-Optimization

p10

Reference I