Whether Large Language Models Learn at the Inference Stage?

No Institute Given

Abstract. In-context learning (zero, one and few-shot learning), chain-of-thoughts and prompt engineering have been more and more widely researched and discussed recently, in connection with the discovered possibility of Large Language Models (LLMs) based on examples (and in particular demonstrations of instructions, some actions with examples, including demonstration of reasoning) to solve certain types of problems at Inference, without the need for fine-tuning, and in general without any additional training of LLMs on these tasks.
In our work, due to the lack of a general term, we refer to this emergent ability, observed in LLMs, as the L&R effect (Learning and Reasoning at the Inference Stage effect). This effect, along with the opportunities it creates, leaves open the question of what exactly is the reason for the emergence of such a surprising ability in LLMs.
In this paper we formulate a hypothesis, consistent with both the literature and our experiments, that explains the cause of the L&R effect and answers the main question: Do LLMs learn at the Inference or not?

Keywords: Learning · Reasoning · Inference · Large Language Models

1 Introduction

Oddly enough, however in NLP(Natural Language Processing), there is no single term that characterizes the emergent ability being studied, while the terms "learning" or "reasoning" are somewhat misleading, as it will be seen in further.

However, the words "learning" and "reasoning" have firmly entered the terminology, so we will use the term L&R effect (Learning and Reasoning at the Inference Stage effect) and we will understand by it the ability of LLMs at the Inference stage based on:

1) demonstration of examples, instructions describing the task, or

2) demonstrating both examples and instructions, and any actions with them (for example, logical or arithmetic operations, including chains of reasoning)

give the correct answer, without any additional training of the model.

The L&R effect is emergent in nature[1] (emergence is a qualitative property that occurs spontaneously in a system when it reaches a certain threshold of complexity). The terminology used in this paper systematically follows the terminology used in the [2], [3] for the terms *in-context, few-shot, one-shot, zero-shot learning* and the term *chain-of-thought(CoT)* is understood according to [4] as

a series of intermediate reasoning steps.

Main questions in this work:

I. L&R effect - What is it?
II. Do LLMs learn and reason at the Inference?

To answer for this questions, was formulated the following hypothesis, the validity of which is consistent with the facts and observations in Part 2 and will be tested experimentally in Part 3.

Hypothesis:

Teaching LLMs based on languages data leads to the creation of inner language spaces, that contain not only the language itself linguistically, but also some of the patterns of rules of reasoning implicitly embedded in the structures of human languages.

LLMs "learn" at inference without actually learning:

1. LLMs use (at the Inference Stage) language spaces of reasoning that LLMs have previously created at the Train stage.

2. Demonstrating examples of reasoning with data(or just instructions with and without data) focuses the LLM's "attention" on those areas of the language space of reasoning where already there are similar "rules" that allow models to work with this and similar reasoning patterns or(and) data.

2 Related works

First, consider a special case of few-shot learning for machine translation where a reasonable explanation of the phenomena is internal language spaces.

In the paper – [5], the authors "demonstrate the potential of few-shot translation systems trained on unpaired language data, and it turns out that with only 5 few-shot examples of high-quality translations, the quality of the resulting translations can match state-of-the-art models as well as more general commercial translation systems. That is, 5 few-shot examples of high-quality inference translation, for initially unpaired language data for the model, is enough to match the two languages very well.

And this special case, which we emphasized, is particularly revealing, because it makes it easy to see that without pre-formed language constructs at the training stage, the model cannot learn how to make high-quality translations in initially unpaired languages, just on the basis of 5 inference examples.

We think that LLMs create, at the training stage, an internal language spaces for each language. Since we are dealing with human languages, these internal language spaces have a common basic structure and can be matched by several n-dimensional points. These "calibration" points are fed into the model at inference as a few-shot examples.

At the inference stage, these language spaces created at the train stage, are matched, allowing LLMs to translate between languages that were not originally

paired.

Now let's look at the facts and observation about reasoning abilities of LLMs.

The first thing to note is what the authors in [6] call rule-like generalization. According the article, LLMs receive in-context examples and draw conclusions based on "rules", i.e., using some "reasoning", the ability to create and apply which appears when they learn languages and according to [7] - if the LLM was trained on more "weak" data, "weaker" than the language - then the L&R effect is not observed.

Consider [6] in more detail. The researchers observed distinct patterns of generalization in transformers when they processed information in-context versus in-weight. The results indicate that generalization based on in-weight information is entirely rule-based when the model is trained on synthetic data, whereas generalization based on in-context information is primarily exemplar-based. Furthermore, research has shown that it is feasible to prompt the transformer to make generalizations based on rules by pre-training it on a classification problem that relies solely on rules. Remarkably, as the size of the language model increases, it becomes more proficient at making rule-based generalizations from contextual information. These findings suggest that natural language data may have a significant impact on the development of rule-like generalization, with the effectiveness being determined by the scale of the model.

The second point will be about chain-of-thought. A series of intermediate reasoning steps improves the ability of LLMs to reproduce complex reasoning at the inference stage [4] and from same authors - paper [8] about self-consistency, to replace the naive greedy decoding used in chain-of-thought prompting.

In details, after previous article, about rule-like generalization, we have an understanding that LLMs pre-trained in languages, receiving examples at the Inference, are surprisingly rule-based when generalizing on these examples. Based on this understanding, we can quite logically assume that the addition of an example-only prompt with rules for working with these examples would be more "understandable" by LLMs and could probably give better results. Indeed, what we might assume from our analysis corresponds to what was called in [4] chain-of-thoughts – a series of intermediate reasoning steps, which, as expected, as we now understand, improves the ability of LLMs to reproduce complex reasoning on the inference stage.

In [8], the authors proposed to modify the chain-of-thoughts method, which led to a further improvement in the results. The self-consistency technique is composed of three main steps. Firstly, a language model is prompted through the chain-of-thought (CoT) method. Next, instead of using the "greedy decode" approach, the CoT prompting is modified to randomly sample from the language model's decoder, resulting in the generation of various reasoning paths. Finally, these reasoning paths are marginalized, and the most consistent answer is selected from the resulting answer set.

It first samples a diverse set of reasoning paths instead of only taking the greedy one, and then selects the most consistent answer by marginalizing out the

sampled reasoning paths. Self-consistency leverages the intuition that a complex reasoning problem typically admits multiple different ways of thinking leading to its unique correct answer.

However, in addition to the last two articles, which seem to indicate that LLMs learn, there are following facts and observations that contradict this assumption.

In research [9] was found that validity of reasoning matters only a small portion to the performance (it remains 80-90% of the original). While relevance to the input query and following the order along the reasoning steps are the key to the effectiveness of chain-of-thought prompting

In articles [10] explored a chain-of-thoughts with formal logic. LLMs are quite capable of doing correct single steps of deduction and are generally able to reason even in fictional contexts. However, they have difficulty planning proofs - when there are multiple valid inference steps available, they cannot systematically explore different options .

In [11] was studied 12 LLMs including GPT-3 and it was found that the models can use prior from pretraining and at the same time ignore the task defined by the demonstrations. Genuine demo values are not really required - randomly swapping labels in demos has almost no performance impact on a number of classification and multiple-choice tasks. While other aspects of the demonstration are key - namely, providing some examples: "the label space, the distribution of the input text, the overall format of the sequence".

It the work [12] was observed that models often learn just as well with misleading and irrelevant templates as well as instructive ones, and the choice of target words is more important than the meaning of the overall prompts . In general, the results obtained contradict the widely accepted hypothesis in the literature that prompts serve as semantically meaningful instructions for performing a task and that writing highly effective prompts requires domain knowledge .

A lot of work today is devoted to ways to improve the prompt. This is called prompt engineering, for example: [13], [14], [15], [16], [17] - it doesn't matter what the prompt is, only the result matters. By successfully setting up the prompt, including training, supplementing with web search, etc. - possible to improve the quality (prompt engineering). That is, it is no longer about what is right and in what form it seems right to us to feed at the inference into the model as a demonstration of examples, reasoning and instructions, but what exactly will work with the best quality.

3 Methodology

The main questions for experiments was:

Does the LLMs learn to reason at the Inference or only use the reasoning abilities learned at the Train stage?

Since we were primarily interested in identifying patterns rather than detailed assessments, we manually ran a small number of the same tests on very different models (a total of 400 tests were performed, which were also visually checked manually).

The work, the methodology and tests of which we follow [9], was performed using InstructGPT-175B[18] (text-davinci-002 and addition text-davinci-003). Therefore, it was especially interesting to compare the results of this work with:

1. The OpenSource line of LLMs Bloom[19] and Bloomz[20]: bloomz-560M, bloomz-1 B1, bloomz-1.7B, bloomz-3B, bloomz-7B1, bloom 176B + mt0-xxl 13B;

2. Quanaco 33B[21] which is LLAMA33B [22] fine-tuned with Quantized LoRA(Low Rank Adapters);

3. The latest version of GPT - GPT 4[23] at the moment.

For arithmetic reasoning, was used GSM-8K[24], mathematical reasoning benchmarks and for Q&A reasoning was used Bamboogle[25], a dataset of compositional questions. With this tests were verified facts and observations regarding chain-of-thought (CoT) - as a form of L&R effect, the most complex and demanding to the size of the model. In general, the test results correlate well with the results from [9] and other facts from the literature overview [1], [2].

Chain-of-thought was tested on the full line of available LLMs, in a variant with 9 few examples of question-answer, followed by a question whose answer is expected from the model, for example - 1 test out of 400 in Figure 1.

In addition, the 9 examples demonstrated by each model were in 4 variants (in our tests in 4, in [9] - 9 variants), just one example from each variant to illustrate the type of test data:

STD (Standard prompting) Q: There were nine computers in the server room. Five more computers were installed each day, from monday to thursday. How many computers are now in the server room?

A: 29

CoT (Chain-of-Thought prompting) Q: There were nine computers in the server room. Five more computers were installed each day, from monday to thursday. How many computers are now in the server room?

A: There were originally 9 computers. For each day from monday to thursday, 5 more computers were installed. So 4 * 5 = 20 computers were added. Now 9 + 20 = 29 computers are now in the server room. The answer is 29.

Invalid Reasoning Q: There were nine computers in the server room. Five more computers were installed each day, from monday to thursday. How many computers are now in the server room?

A: There were originally 9 computers. For each day from monday to thursday, 5 more computers were installed. Now 9 * 5 = 45 computers. Since 4 * 4 = 16, now 45 - 16 = 29 computers are now in the server room. The answer is 29.

No relevance Q: There were nine computers in the server room. Five more computers were installed each day, from monday to thursday. How many computers are now in the server room?

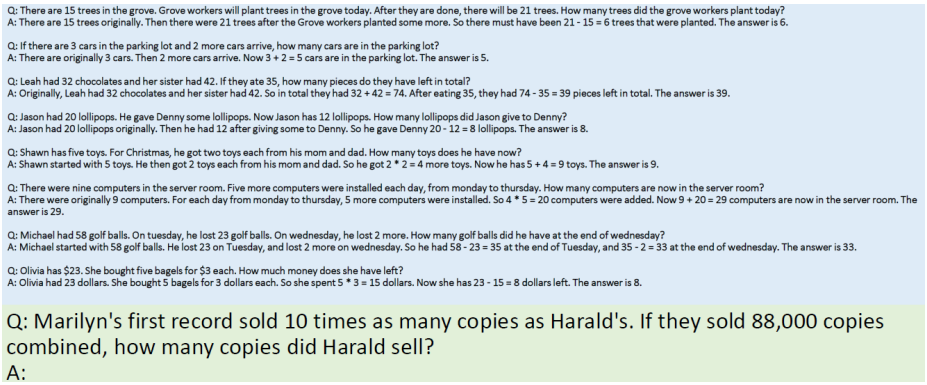

Fig. 1. This all is only 1 prompt, 1 test from 400, in one form from 4 settings (standard, chain-of-thought, invalid, no relevant). Blue - 8 (few shot) demonstration (Q&A) how need reason and solve task in this setting. Green - question from GSM8K test.

A: Haley is currently 23 inches tall. She grows at the rate of 10 inches every year for 4 years. So she will have grown by 10 * 4 = 40 inches. Her height after 4 years will be 23 + 40 = 63 inches. The answer is 63."

4 Experiment results

First, the results of observing the L&R effect for each of the models:

1) bloomz-560M, bloomz-1 B1, bloomz-1.7B, bloomz-3B - the CoT effect was not observed. The answer of the model was just numbers, without chain-of-thought, and the answer was not correct.

2) bloomz-7B1 – the model's response was short fragments of reasoning and (or) incorrect answers. Probably, this allows us to state the first signs of the CoT effect, but on a model of this size - with a result of zero quality.

3) mt0-xxl 13B – the effect was not observed, but the model itself was from another line. The model's answer was only numbers or facts, without chain-of-thought, the answers were not correct.

4) bloom 176B – the CoT effect was observed in full. Reasoning reaching the answer - but often the answer was not correct, and the reasoning is sometimes inadequate.

5) Quanaco 33B - the CoT effect was observed in full. Model demonstrates very good results, despite the fact that all other models in the tables are superior in size. Definitely better than Quanaco 33B only GPT4 and comparable only text-davinci-003,

6) ChatGPT Plus with GPT 4 (the size of the model is not known, most likely not less than 175B as it was in the previous version) – the CoT effect was observed in full, correct reasoning and the correct result in almost all cases.

Second, here are the summary tables with tests – Table 1 and 2 [1].

The omission of data in Tables 1, 2 in the row № 1 for the models text-davinci-2 and text-davinci-3 is explained by the fact that the data were taken for comparative analysis from [9], and the authors did not conduct such an experiment.

Table 1. Percentage of correct answers on the GSM-8K (arithmetic reasoning). Right reasoning (CoT) and Invalid reasoning give the same or almost the same result in all models. This means that models don't learn at the Inference but use prior reasoning patterns from the Train.

№	Test setting	Bloom 176B	Quanaco 33B LLAMA QLoRA	text davinci 002	text davinci 003	GPT4 ChatGPT Plus
1	Without any examples of resoning (without demo)	0%	40%	-	-	100%
2	STD (Standard prompting) without thoughts	0%	0%	0%	10%	80%
3	CoT: Chain-of-Thought right reasoning	10%	50%	40%	60%	90%
4	Invalid Reasoning	10%	50%	40%	80%	100%
5	No relevance	0%	0%	10%	10%	100%

And now the detailed results and conclusions from the experiments will be formulated:

In full agreement with the literature [1], [2] the L&R effect is emergent and is not observed in models up to a certain model size.

The lower limit of the observation of L&R effect signs depends not only on the size of the model, but also on the model itself. Signs of the effect, although weak, observed in bloomz-7B1, do not appear in mt0-xxl 13B. As it was described in the studied sources[1], [2].

As we assumed earlier, in the section on hypotheses, if there is no reasoning pattern already formed on train in the model, the effect will emerge weakly (in case of falling into a similar area of reasoning within the language space of reasoning), or it will not be observed at all – which emerges in the fact that the answers and reasoning will be inadequate to the question (we see all this in the case of Bloom on GSM8K).

The most advanced model, ChatGPT4 – most often simply ignores the demonstrated examples, and the worse the example (that is, from the no relevance category), the better accuracy. Accordingly, the model explicitly uses a priori knowledge and reasoning methods that initially already reach the limit level (without the need for CoT on test examples).

[1] Tests are available here

Table 2. Percentage of correct answers on the Bamboogle Test (Q&A reasoning about facts). The same patterns Right reasoning (CoT) and Invalid reasoning are observed as in the previous table.

№	Test setting	Bloom 176B	Quanaco 33B LLAMA QLoRA	text davinci 002	text davinci 003	GPT4 ChatGPT Plus
1	Without any examples of resoning (without demo)	40%	90%	-	-	100%
2	STD (Standard prompting) without thoughts	30%	50%	30%	50%	80%
3	CoT: Chain-of-Thought right reasoning	40%	100%	50%	90%	100%
4	Invalid Reasoning	40%	80%	50%	90%	100%
5	No relevance	30%	20%	50%	60%	100%

The prohibition of explicit reasoning, which is "standard" prompting, significantly harms all the models.

Completely inappropriate examples of "no relevance" - damage all models except ChatGPT4.

ChatGPT4, although it ignores complete nonsense ("no relevance"), but it can worsen the result with correct CoT, which probably correspond less to the task than its own chains of reasoning. That is, the correct CoT from the examples – position ChatGPT4 (in 10% of cases) on less suitable patterns of reasoning than its own, from prior. As a result, the correct CoT examples sometimes interfere with ChatGPT4 on GSM8K.

As can be seen from lines 3 and 4 in each of tables 1 and 2 and in full agreement with observations in [9] - Right reasoning and Invalid reasoning give the same or almost the same result in all models, but as tests have shown, often on a different questions That is, the demos are only a calibration points for pre-train patterns of reasoning. This means that models don't learn at the Inference but use prior reasoning patterns from the Train.

Based on this, we think that the our findings and other available observations sufficiently support our hypothesis that LLMs do not learn at Inference, but use the reasoning patterns they extracted from the languages at the Train stage.

5 Conclusion

After analyzing the facts and observations in other works and conducting experiments, the original questions were answered:

Question I. L&R effect - What is it?

Hypothesis was formulated that LLMs form "Inner Language spaces of reasoning" during the Training stage. It's prior from Train stage and this hypothesis

aligns well with all facts and observations found in the literature (referenced in Part 2) and is supported by our experiments (discussed in Part 3).

Question II. Do LLMs learn and reason at the Inference?

Various hypotheses regarding the learning of LLMs at the Inference stage, including those [26], [27], [28], [29] and other studies, have been explored. However, to date, we have not found any evidence in either the literature or our experiments that confirms the actual realization of the theoretical cases discussed in these papers. Following references [9], [11], [30], we state, perhaps somewhat categorically, that LLMs in the Inference stage do not learn in the classical sense of the word. Rather, their ability to learn and reason is constrained by the capabilities acquired during the Training stage.

6 Future directions

It would be desirable to approach the questions considered in this work from the perspective of building a general theory of language spaces and patterns of reasoning in LLMs formed on languages. It want to note that it's necessary to distinguish between two aspects of this phenomenon:

1. The formation, which occurs during the Training stage,
2. The usage, which occurs during the Inference stage,

of language spaces of reasoning in LLMs.

These aspects can either be part of a single, integral description or independent descriptions.

An interesting and important topic for further research would be to investigate the actual nature of the dependence of the ability to learn and reason on the number of model parameters. The emergent properties have been studied previously[1], but due to the lack of a consistent series of models, questions remain both about the presence of exact thresholds for emergence and about the nature of the dependence of emergent properties on the number of model parameters. The unresolved nature of these questions leaves too much room for assumptions, e.g. [31].

It seems important to understand the true dynamics of emergence in the ascending series of the same model, but with a different number of parameters:

1, 2, 3, ... , 174, 175, 176, ... ,999, 1000B

Unfortunately, such a series of models does not exist today. This is the only way to really understand how the emergent properties depend on the number of model parameters. Ideally, it would be good to do this on several independent model lines of LLMs and compare the results.

References

1. Jason Wei et al. Emergent abilities of large language models, 2022.
2. Tom B. Brown et al. Language models are few-shot learners, 2020.

3. Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Zhiyong Wu, Baobao Chang, Xu Sun, Jingjing Xu, Lei Li, and Zhifang Sui. A survey on in-context learning, 2023.
4. Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models, 2023.
5. Xavier Garcia, Yamini Bansal, Colin Cherry, George Foster, Maxim Krikun, Fangxiaoyu Feng, Melvin Johnson, and Orhan Firat. The unreasonable effectiveness of few-shot learning for machine translation, 2023.
6. Stephanie C. Y. Chan, Ishita Dasgupta, Junkyung Kim, Dharshan Kumaran, Andrew K. Lampinen, and Felix Hill. Transformers generalize differently from information stored in context vs in weights, 2022.
7. Stephanie C. Y. Chan, Adam Santoro, Andrew K. Lampinen, Jane X. Wang, Aaditya Singh, Pierre H. Richemond, Jay McClelland, and Felix Hill. Data distributional properties drive emergent in-context learning in transformers, 2022.
8. Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models, 2023.
9. Boshi Wang, Sewon Min, Xiang Deng, Jiaming Shen, You Wu, Luke Zettlemoyer, and Huan Sun. Towards understanding chain-of-thought prompting: An empirical study of what matters, 2023.
10. Abulhair Saparov and He He. Language models are greedy reasoners: A systematic formal analysis of chain-of-thought, 2023.
11. Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. Rethinking the role of demonstrations: What makes in-context learning work?, 2022.
12. Albert Webson and Ellie Pavlick. Do prompt-based models really understand the meaning of their prompts?, 2022.
13. Laria Reynolds and Kyle McDonell. Prompt programming for large language models: Beyond the few-shot paradigm, 2021.
14. Xiao Liu, Yanan Zheng, Zhengxiao Du, Ming Ding, Yujie Qian, Zhilin Yang, and Jie Tang. Gpt understands, too, 2021.
15. Yongchao Zhou, Andrei Ioan Muresanu, Ziwen Han, Keiran Paster, Silviu Pitis, Harris Chan, and Jimmy Ba. Large language models are human-level prompt engineers, 2023.
16. Omar Khattab, Keshav Santhanam, Xiang Lisa Li, David Hall, Percy Liang, Christopher Potts, and Matei Zaharia. Demonstrate-search-predict: Composing retrieval and language models for knowledge-intensive nlp, 2023.
17. Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models, 2023.
18. Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback, 2022.
19. BigScience Workshop et al. Bloom: A 176b-parameter open-access multilingual language model, 2023.
20. Niklas Muennighoff, Thomas Wang, Lintang Sutawika, Adam Roberts, Stella Biderman, Teven Le Scao, M Saiful Bari, Sheng Shen, Zheng-Xin Yong, Hailey Schoelkopf, Xiangru Tang, Dragomir Radev, Alham Fikri Aji, Khalid Almubarak,

Samuel Albanie, Zaid Alyafeai, Albert Webson, Edward Raff, and Colin Raffel. Crosslingual generalization through multitask finetuning, 2023.
21. Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. Qlora: Efficient finetuning of quantized llms, 2023.
22. Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. Llama: Open and efficient foundation language models, 2023.
23. OpenAI. Gpt-4 technical report, 2023.
24. Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems, 2021.
25. Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah A. Smith, and Mike Lewis. Measuring and narrowing the compositionality gap in language models, 2023.
26. Ekin Akyürek, Dale Schuurmans, Jacob Andreas, Tengyu Ma, and Denny Zhou. What learning algorithm is in-context learning? investigations with linear models, 2023.
27. Johannes von Oswald, Eyvind Niklasson, Ettore Randazzo, João Sacramento, Alexander Mordvintsev, Andrey Zhmoginov, and Max Vladymyrov. Transformers learn in-context by gradient descent, 2023.
28. Shivam Garg, Dimitris Tsipras, Percy Liang, and Gregory Valiant. What can transformers learn in-context? a case study of simple function classes, 2023.
29. Damai Dai, Yutao Sun, Li Dong, Yaru Hao, Shuming Ma, Zhifang Sui, and Furu Wei. Why can gpt learn in-context? language models implicitly perform gradient descent as meta-optimizers, 2023.
30. Yasaman Razeghi, Robert L. Logan IV au2, Matt Gardner, and Sameer Singh. Impact of pretraining term frequencies on few-shot reasoning, 2022.
31. Rylan Schaeffer, Brando Miranda, and Sanmi Koyejo. Are emergent abilities of large language models a mirage? *arXiv preprint arXiv:2304.15004*, 2023.