On Limitations of the Transformer Architecture - YouTube -- Transcript

Summary

The text explores the limitations of the Transformer architecture in artificial intelligence, focusing on its tendency to produce 'hallucinations' where answers generated do not align with input or questions. It discusses theoretical limitations of Transformers related to function composition and complexity theory. The text introduces the Chain of Thought approach to simplify complex problems for Transformers. It also delves into the challenges Transformers face in handling compositional tasks, such as logic puzzles and circuit evaluation. The discussion is grounded in concepts of communication and computational complexity and highlights limitations in Transformer performance, particularly in handling tasks requiring deeper levels of compositionality. The text concludes by discussing the implications of complexity arguments on the future development of Transformer architectures.

Bullet Points

- Transformers in artificial intelligence can produce hallucinations where the answers generated do not align with training data or questions asked.
- Previous studies have identified theoretical limitations of Transformers, such as struggles with simple patterns and function composition.
- Transformers fall into a relatively weak complexity class, struggling with tasks requiring sequential composition.
- Transformers struggle with tasks requiring deep function composition, leading to unreliable outputs and potential hallucinations.
- Communication and computational complexity arguments reveal fundamental limitations of Transformers in handling compositional tasks, suggesting a mismatch between problems and the architecture.

Formatted Transcript

Section Introduction: In this section, we delve into the intriguing yet challenging aspect of the Transformer architecture in artificial intelligence, which is its tendency to produce hallucinations. This means that sometimes the answers generated by these models don't align with the information they were trained on or the question asked. There's a lot of research exploring the nature types and solutions to these hallucinations.

One might wonder if the root of this issue lies within the Transformer architecture itself. Previous studies, including a notable one from 2020, have highlighted theoretical limitations of Transformers, such as their inability to recognize simple patterns like whether a sentence has an even number of negations or if parentheses are balanced. These limitations, while proven through complex theoretical models, seem to only emerge with very large inputs which are not common in practical scenarios. In fact, it has been shown that Transformers can handle these tasks well within realistic settings.

Our exploration also touches on how Transformers are viewed through the lens of complexity theory, revealing that they fall into a relatively weak complexity class. This perspective helps us understand the computational boundaries of Transformers. Additionally, recent research has

pinpointed specific mathematical challenges that Transformers struggle with, such as identifying three numbers in a sequence that sum up to zero modulo a large number. This finding contrasts with their ability to solve simpler versions of the problem and highlights a unique advantage of Transformers over other machine learning models. Even though these challenges are not directly related to their primary functions, we are particularly interested in a fundamental limitation of the Transformer architecture – its difficulty with function composition, a basic yet crucial semantic operation.

For instance, when asked about the birthday of Frederick Chan's father based on given facts, a Transformer might fail to connect the dots correctly. This is a clear example of function composition where the model needs to combine relational information effectively. Some have suggested integrating knowledge graphs with Transformers to improve their ability to perform function composition and reduce hallucinations. Function composition is not only vital for combining relational data but also plays a key role in language understanding, which is a core strength of Transformers. The handling of indexicals or words that refer to entities within a specific context further illustrates the importance of function composition and understanding language. However, Transformers sometimes struggle with this, indicating a potential area for improvement.

In our paper, we demonstrate that Transformers inherently struggle with function composition. Specifically, we show that a single Transformer attention layer is unlikely to compute function composition queries correctly if the size of the function's domain and certain other parameters exceed a particular threshold. This limitation seems to stem from the softmax computation within the Transformer, which relies on limited non-local information to compute the next token's embedding. While this issue is significant, we also discuss how the Chain of Thought cut approach can mitigate the problem by breaking down tasks into smaller steps. However, we also present a theorem suggesting that even with Cot, a Transformer layer would require a significantly larger prompt to handle a series of function compositions. Furthermore, we propose another argument that applies to any number of layers, focusing on a different type of hallucination related to compositionality tasks.

Through extensive experimentation, it has been shown that Transformers struggle with tasks requiring sequential composition, and this difficulty increases with the depth of composition required. This limitation is tied to a widely accepted complex assumption suggesting that multi-layer Transformers may not be capable of performing several crucial computations for compositionality tasks. Lastly, we discuss how the probabilistic nature of Transformers as language generators might contribute to their tendency to veer off course. The models aim to maximize the probability of generating the next token can lead to unreliable outputs especially when dealing with low probability inputs tasks or outputs. This issue is exacerbated in situations where the input context is ambiguous or underspecified, increasing the likelihood of generating incorrect tokens and potentially leading to hallucinations. This phenomenon has been studied more formally in recent research, which examines the statistics of rare patterns in the training data.

Section Summary: In this section, we explore the limitations of the Transformer architecture, particularly in its ability to perform function composition tasks. Relying on the foundational Definitions Necessary to understand our discussion on Transformers' function composition and information theory.

Preliminary Definitions: In this section, we begin by laying out some foundational concepts necessary to understand our discussion on Transformers' function composition and information theory. Firstly, we delve into the concept of a transformer to mathematically describe how Transformers operate. We slightly modify an existing formal model. At the heart of a Transformer is the self-attention unit, which is essentially a function that transforms a sequence of input vectors into a sequence of output vectors, maintaining the sequence length but potentially altering

the dimensionality of the vectors. This transformation is achieved through the use of three matrices known as the key, query, and value matrices, which are all of the same dimension for simplicity in our explanation.

When a sequence of vectors is fed into the self-attention unit, it computes a new sequence where each element is a weighted sum of the original vectors, with the weights determined by the attention mechanism. We also introduce the concept of an h-headed Transformer layer, which consists of multiple self-attention units working in parallel on the same input but potentially capturing different aspects of the input data. The outputs of these units are then combined to produce a single output vector for each input vector in the sequence. A Transformer model is built by stacking several of these layers on top of each other. Moving on, we define the problem of function composition.

Imagine we have two functions, where one maps elements from one domain to another and the second function maps elements of the second domain to a third domain. We describe these functions in a structured prompt divided into three parts. The first part describes the first function, the second part describes the second function, and the third part poses a query about the composition of these two functions. We say that a Transformer layer correctly computes the function composition if it can provide the correct answer to the query based on the descriptions of the two functions.

In the realm of information theory, we stick to standard terminology. We discuss concepts like entropy, which measures the uncertainty of a random variable, and mutual information, which quantifies the amount of information shared between two random variables. These concepts play a crucial role in our analysis. Lastly, we tackle the challenge of proving the impossibility of certain computations by Transformers under specific conditions. We demonstrate that if a Transformer layer does not have sufficient capacity, measured in terms of the number of heads, the dimensionality of the embeddings, and the precision of computation, it cannot correctly solve the function composition problem for arbitrary functions.

This conclusion is drawn from principles in communication complexity, which studies the minimum amount of information exchange required for distributed agents to compute a function. Our proof shows that if the Transformer layer's capacity is below a certain threshold, it will inevitably fail to compute the function composition correctly with a non-negligible probability. In summary, this section sets the stage for our exploration of the limitations of Transformers in computing function compositions grounded in the mathematical framework of communication complexity and information theory.

Section Summary: In this section, we introduce preliminary definitions related to Transformers' self-attention units and function composition. Transformers consist of self-attention units defined by key, query, and value matrices and are organized into layers with combining functions. The function composition problem involves combining two functions through a prompt where an h-headed Transformer layer computes the composition correctly. The section also delves into the impossibility of solving function composition problems with certain constraints, leveraging concepts from information theory and communication complexity.

Section Chain of Thought: In this section, we explore whether the Chain of Thought cut approach can assist in solving complex problems by breaking them down into simpler steps. We believe that the answer is yes. For instance, when faced with a question about the birthplace of Turing, we can simplify the problem by creating a Cot prompt that divides the question into easier parts, such as determining where Turing was born, identifying the country of that place, and then concluding Turing's country of birth. However, we demonstrate that solving more complex problems, which involve applying multiple functions in sequence, may require an indefinitely large number of Cot steps.

Specifically, we look at problems where we have to apply a series of functions one after the other to a starting value. Imagine we have a set of functions and we apply the first function to a number, then apply the second function to the result of the first function, and so on until we have applied all functions. We then delve into the technical requirements for a Transformer layer, a type of neural network architecture, to solve these iterated function composition problems using Cot. We consider factors such as the number of attention heads in the Transformer, the dimension of the embeddings, the precision of the computations, and the size of the problem domain.

We find that the number of Cot steps needed for a Transformer layer to correctly answer these problems is proportional to the square root of the ratio of the problem's domain size to the product of the number of attention heads, the embedding dimension, and the computation precision. To support our findings, we draw parallels with a known problem in communication complexity called pointer chasing. In this problem, two parties, Alice and Bob, each know a part of a sequence of functions, and they need to communicate to find out the result of applying these functions in sequence. We show that if a Transformer layer can solve the iterated function composition problem within a certain number of Cot steps, then Alice and Bob can use a similar strategy to solve the pointer chasing problem by exchanging information about the results of applying their functions.

We outline a communication protocol where Alice and Bob take turns simulating the steps of Cot, exchanging information about the application of their functions. This process continues until they have simulated all required Cot steps, allowing Bob to compute the final result. The total amount of information exchanged depends on the number of Cot steps, the number of attention heads, the embedding dimension, and the computation precision. By combining our observations and calculations, we conclude that a Transformer layer needs a specific number of Cot steps to solve iterated function composition problems, which also enables solving the pointer chasing problem within a certain communication complexity.

This conclusion helps us understand the capabilities and limitations of the Cot approach in solving complex problems.

Section Summary: In this section, we explore the potential of Chain of Thought aau in addressing composition problems, demonstrating its effectiveness in breaking down complex questions into simpler ones for language model llm responses. However, we establish that a significant number of Cot steps are necessary to solve generalized composition challenges involving multiple function applications, by leveraging insights from communication complexity problems like pointer chasing. We reveal the intricate relationship between Cot steps and the efficiency of solving iterated function composition tasks using Transformer layers.

Section Compositionality and Logarithmic Space: In this section, we delve into the concept of compositionality and its challenges within the realm of logarithmic space. Recently, a study uncovered a new type of error made by Transformer models – they start with compositional tasks. These tasks involve performing basic operations repeatedly, such as multiplying multi-digit numbers, solving a sum maximization problem with specific constraints, and tackling logic puzzles like Einstein's riddle. These errors are intriguing because they highlight a potential limitation in how these models process information.

When we talk about evaluating a circuit, we're referring to the process of determining the output based on the inputs and the operations found within the circuit. This could be as simple as multiplying numbers or as complex as solving optimization problems with certain rules. Another concept we explore is derivability, which involves finding a sequence of steps from an initial state to a final state within a given set of rules. This is akin to solving a puzzle where each move must be legal to reach the end goal.

Logical reasoning, particularly with logic puzzles, is typically framed as a satisfiability problem, which is known to be quite challenging. Despite this, there are simpler cases like 2 sat, horn sat, and mod 2 sat, which are more manageable and form the basis of much common sense reasoning. However, we question whether Transformer models can handle even these simpler cases effectively. We observe that the computation performed by a multi-layer Transformer on a given input can be done using a logarithmic amount of memory space. This isn't a new discovery, but it's important because it suggests that Transformers operate within a certain computational complexity class.

This limitation in memory usage is crucial when we consider the ability of these models to handle tasks that require more complex reasoning or deeper levels of compositionality. We further discuss that if we accept certain widely held beliefs in computational complexity, it becomes clear that multi-layer Transformers cannot solve certain problems unless some of these beliefs are proven wrong. Specifically, problems like derivability, 2 sat, horn sat, and circuit evaluation are out of reach unless we find that deterministic and non-deterministic logarithmic memory are equivalent or even that deterministic logarithmic memory is as powerful as polynomial time, which are significant shifts from current understanding.

This insight helps explain why Transformers might falter on tasks that increase in complexity or depth as demonstrated in experiments. The performance of these models drops significantly as the task depth exceeds a certain threshold, aligning with our discussion on computational complexity. This connection between the theoretical limitations and practical performance challenges sheds light on the nature of the errors observed in compositional tasks and suggests avenues for further investigation into the capabilities and limitations of Transformer models.

Section Summary: In this section, we identify a novel type of hallucinations where Transformers struggle with compositional tasks such as multiplying multi-digit integers or solving logic puzzles like Einstein's riddle. By analyzing the memory requirements of multi-layer Transformers, we show that they cannot reliably handle tasks like derivability, 2 sat, horn sat, and circuit evaluation unless certain complexity conjectures are violated, shedding light on the limitations observed in their performance on such tasks.

Section Discussion: In this section, we delve into how we've applied two types of complexity arguments – communication complexity and computational complexity – to highlight certain limitations of the Transformer architecture which lead to various kinds of errors in output, often referred to as hallucinations. We've demonstrated that a single layer of a Transformer can't solve the basic function composition problem. Moreover, we found that Chain of Thought not can tackle the iterated composition problem only if it generates a prompt that's significantly long, specifically the length must be proportional to the square root of n.

These mathematical insights have their limitations. Firstly, they apply when the size of the function's domain is larger than the Transformers dimension parameters, which are usually in the hundreds. Secondly, these insights don't hold when considering multiple layers of Transformers. We also explored evidence from the realm of complexity theory, specifically looking at classical Turing machines to show that the tasks which are empirically challenging for Transformers involve computational elements that are inherently difficult for Transformers to process.

It's important to note that the complexity arguments we use come with certain caveats. The claim that composition is impossible for a single layer holds in a probabilistic sense – the error probability is not zero but also not certain – and only when the function's domain exceeds the Transformer layer's parameters. The arguments based on computational complexity come with their own set of conditions – they depend on certain conjectures that, while widely accepted, remain unproven. And these results are asymptotic, meaning they apply to instances beyond a certain unknown size, where these conjectures are assumed to be relevant.

These complexity results serve as cautionary tales, suggesting a fundamental mismatch between these problems and the Transformer architecture, indicating that we shouldn't expect these issues to be solvable indefinitely in practice. However, similar to other complexity conjectures like P not equal to NP, computational challenges often arise even with reasonably small instances. For instance, we've observed that Transformers struggle with relatively small compositionality tasks, and we've provided anecdotal evidence in the appendix showing that large language models frequently make errors in response to small prompts related to function composition.

Interestingly, the opposite is also true for some NP-complete problems like 3 sat, which seems solvable in practice for large instances thanks to decades of exploring a wide range of algorithmic techniques tailored to practical instances rather than a fixed algorithmic architecture. The real issue with complexity lower bounds isn't their provisional and asymptotic nature – it's that they are rare, difficult to establish, come in limited forms, and tend to overestimate the capabilities of computational agents. For example, Lemma 1 is designed to hold even under sophisticated mathematical encoding by an agent named Grace.

However, when applied in Theorem 1, Grace's message to Xavier is not a complex coding of her tokens but rather two simple numerical expressions. This simplistic approach to encoding the values of a function highlights an open problem and the potential to develop a more sophisticated version of communication complexity for computationally limited agents, aiming to better understand the limitations of architectures like the Transformer. Our exploration of these two genres of negative results poses an interesting challenge – what would it take to design an attention layer immune to these lower bound techniques.

While keeping the architecture's efficiency and effectiveness, our findings suggest a version of the softmax computation in the attention layer that is either not commutative or not associative or one that requires more than logarithmic space. However, evading a lower bound technique doesn't automatically translate to improved performance.

Acknowledgment: We are thankful to Fernando Pereira for his valuable insights, engagement, and constructive criticism throughout this project. We also appreciate Olivier Bousquet for his insightful comments on an earlier draft. This work was conducted while the third author was a visiting scientist at Google DeepMind in Zurich, and the efforts of the first and third authors were partially supported by an NS Grant.