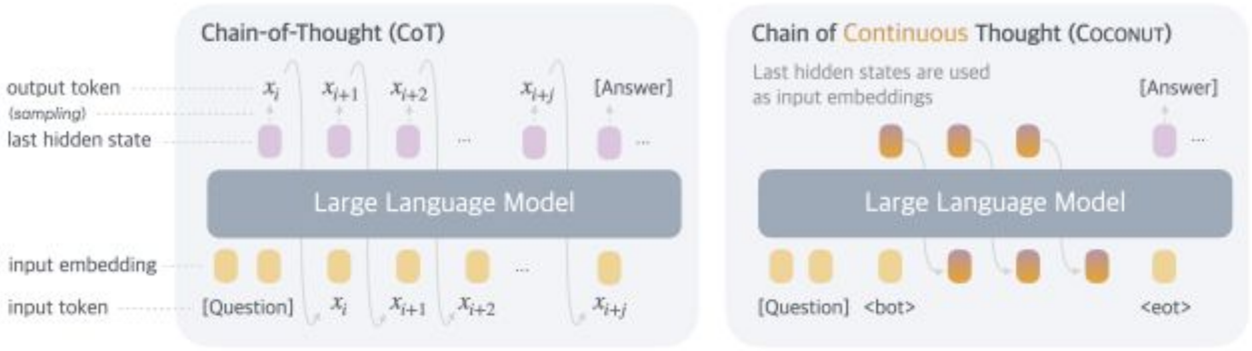
Digesting “COCONUT: Training LLMs to Reason in a Continuous Latent Space”

Roger Lam
Dec 22 2024

Chain of Thought (CoT) vs Change of Continuous Thought (COCONUT)



In CoT, we feed words back into the LLM to generate the next word, repeatedly, until we get to the end.

In COCONUT, we feed the last layer of calculations (numbers) back into the LLM to generate the next numbers, repeatedly, until we get to the end and then translate into words.

That way we retain richer representation (but in numbers) and better performance.

More gradient descent and new training strategy

In this work we instead explore LLM reasoning in a latent space by introducing a novel paradigm, Coconut (Chain of Continuous Thought). It involves a simple modification to the traditional CoT process: instead of mapping between hidden states and language tokens using the language model head and embedding layer, Coconut directly feeds the last hidden state (a continuous thought) as the input embedding for the next token (Figure 1). This modification frees the reasoning from being within the language space, and the system can be optimized end-to-end by gradient descent, as continuous thoughts are fully differentiable. To enhance the training of latent reasoning, we employ a multi-stage training strategy inspired by Deng et al. (2024), which effectively utilizes language reasoning chains to guide the training process.

By keeping our numbers, we can continue to do gradient descent and improve on this transition between input and output.

We also use different training data to better suit this new method.

We replace “steps” with “thoughts”

Figure 2 Training procedure of Chain of Continuous Thought (Coconut). Given training data with language reasoning steps, at each training stage we integrate c additional continuous thoughts ($c = 1$ in this example), and remove one language reasoning step. The cross-entropy loss is then used on the remaining tokens after continuous thoughts.

We start with regular CoT training data. Then train in stages, progressively replacing “steps” with “thought” representations, until we have all the steps represented.

Staying in latent space gives us better perspective

Like a horse race, it's better to know where all the horses are than to know just that #4 is in the lead.

Number of thoughts could map to edges of traversal

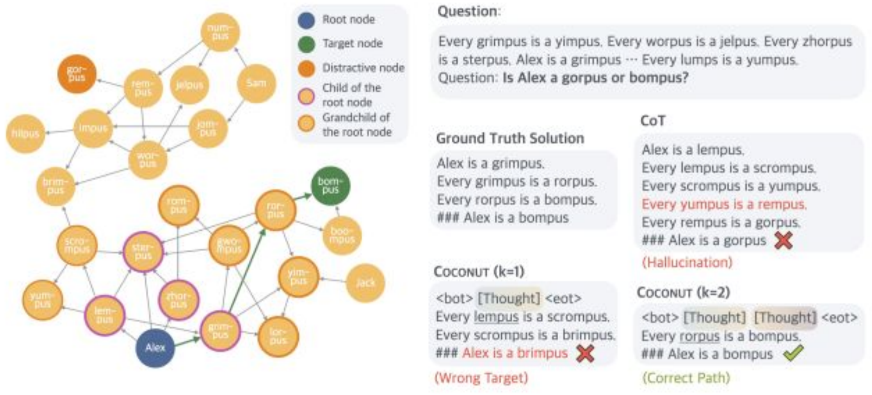


Figure 6 A case study of ProsQA. The model trained with *CoT* hallucinates an edge (*Every yumpus is a rempus*) after getting stuck in a dead end. Coconut (k=1) outputs a path that ends with an irrelevant node. Coconut (k=2) solves the problem correctly.

Take a closer look. The correct answer is two nodes away.

When trained with k=1 (single thought), it gets the answer wrong.

When trained with k=2 (two thoughts), it gets the answer correctly.

Very competitive results with 3-10x fewer tokens

Method	GSM8k		ProntoQA		ProsQA	
	Acc. (%)	# Tokens	Acc. (%)	# Tokens	Acc. (%)	# Tokens
CoT	42.9 ± 0.2	25.0	98.8 ± 0.8	92.5	77.5 ± 1.9	49.4
No-CoT	16.5 ± 0.5	2.2	93.8 ± 0.7	3.0	76.7 ± 1.0	8.2
iCoT	30.0*	2.2	99.8 ± 0.3	3.0	98.2 ± 0.3	8.2
Pause Token	16.4 ± 1.8	2.2	77.7 ± 21.0	3.0	75.9 ± 0.7	8.2
Coconut (Ours)	34.1 ± 1.5	8.2	99.8 ± 0.2	9.0	97.0 ± 0.3	14.2
- *w/o curriculum*	14.4 ± 0.8	8.2	52.4 ± 0.4	9.0	76.1 ± 0.2	14.2
- *w/o thought*	21.6 ± 0.5	2.3	99.9 ± 0.1	3.0	95.5 ± 1.1	8.2
- *pause as thought*	24.1 ± 0.7	2.2	100.0 ± 0.1	3.0	96.6 ± 0.8	8.2

Table 1 Results on three datasets: GSM8l, ProntoQA and ProsQA. Higher accuracy indicates stronger reasoning ability, while generating fewer tokens indicates better efficiency. *The result is from Deng et al. (2024).

Performance across the board is pretty good compared to CoT with much fewer tokens.

Remember, this is just the beginning.

This may be the GPT-2 of reasoning models

- Researchers are looking to biology for areas of research.
- This encodes a more powerful System-1 thinking system vs OpenAI o-series models power a more powerful System-2 thinking system.
- Likely both are valuable - much like how humans use both.

- Also, Meta FAIR is proving to be a researcher's playground. Just take a look: https://ai.meta.com/results/?content_types%5B0%5D=publication

Thanks!

Thanks and follow for more breakdowns.

https://www.lamroger.com

https://www.linkedin.com/in/lam-roger/