

LLM-Reasoner: 让任何大模型都能像DeepSeek R1一样深入思考

原创 南七无名氏 PyTorch研习社 2025年03月10日 08:01 安徽

LLM-Reasoner 是一个库，它让任何 LLM（大模型）都能像 OpenAI o1 和 DeepSeek R1 一样深入思考。

🤖 LLM-Reasoner: Make any LLM to think deeper like OpenAI o1 and deepseek R1!

Make any LLM to think deeper like OpenAI o1 and deepseek R1!

```
C:\Users\haris>llm-reasoner reason "count number of r's in word strawberry" --min-steps 10 --model gpt-4
Thinking... 0% -:--:--
Step 1: Step 1: Understand the Problem
<thinking> The task is to count the number of occurrences of the letter 'r' in the word 'strawberry'. This involves
going through each character in the word and checking if it is 'r'. If it is, we increment a counter. </thinking>
Confidence: 0.95
Thinking time: 6.29s
Thinking... 5% -:--:--
Step 2: Counting 'r' in 'strawberry'
<thinking> The word 'strawberry' has 10 characters. I will start from the first character and move towards the last,
checking each character. If the character is 'r', I will increment a counter. At the end, the counter will hold the
number of 'r's in 'strawberry'. </thinking>
Confidence: 0.95
Thinking time: 4.39s
Thinking... 10% 0:01:20
Final Answer:

<thinking> The first character is 's', so the counter remains at 0. The second character is 't', so the counter
remains at 0. The third character is 'r', so the counter is incremented to 1. The fourth character is 'a', so the
counter remains at 1. The fifth character is 'w', so the counter remains at 1. The sixth character is 'b', so the
counter remains at 1. The seventh character is 'e', so the counter remains at 1. The eighth character is 'r', so the
counter is incremented to 2. The ninth character is 'r', so the counter is incremented to 3. The tenth character is
'y', so the counter remains at 3. </thinking>
```

✨ 主要特点

- 🧠 循序渐进的推理：不再有黑箱答案！准确了解 LLM 是如何思考的，类似于 O1 的系统方法
- 📺 实时进度：通过流畅的动画观看推理的展开
- 🎯 多提供商支持：与 LiteLLM 支持的所有提供商兼容
- 🎮 精美的 UI：一个漂亮的 Streamlit 界面可供使用
- 🔧 高级用户 CLI：无缝嵌入你的代码
- 📊 信心跟踪：了解 LLM 对每个步骤的确定程度



🚀 快速开始

首先安装：

```
1 pip install llm-reasoner
```

设置 OpenAI key:

```
1 export OPENAI_API_KEY="sk-your-key"
```

对于国内用户可以选择提供了与 OpenAI 接口兼容的模型或者是使用 llama_cpp_python 启动一个本地的 LLM 服务，这个服务接口与 OpenAI 接口兼容。

以下是一些简单的用法：

```
1 # 列当前所有可用模型
2 llm-reasoner models
3
4 # 生成一个推理链
5 llm-reasoner reason "How do planes fly?" --min-steps 5
6
7 #启动 UI 界面
8 llm-reasoner ui
```



或者直接在你的代码中使用：

```
1 from llm_reasoner import ReasonChain
2 import asyncio
3
4 async def main():
5     # Create a chain with your preferred settings
6     chain = ReasonChain(
7         model="gpt-4",           # Choose your model
8         min_steps=3,             # Minimum reasoning steps
9         temperature=0.2,        # Control creativity
10        timeout=30.0             # Set your timeout
11    )
12    # Watch it think step by step!
13    async for step in chain.generate_with_metadata("Why is the sky blue?"):
14        print(f"\nStep {step.number}: {step.title}")
15        print(f"Thinking Time: {step.thinking_time:.2f}s")
16        print(f"Confidence: {step.confidence:.2f}")
17        print(step.content)
```



```
18 asyncio.run(main())
```

<https://github.com/harishsg993010/LLM-Reasoner>



PyTorch研习社

打破知识壁垒，做一名知识的传播者

711篇原创内容

公众号

