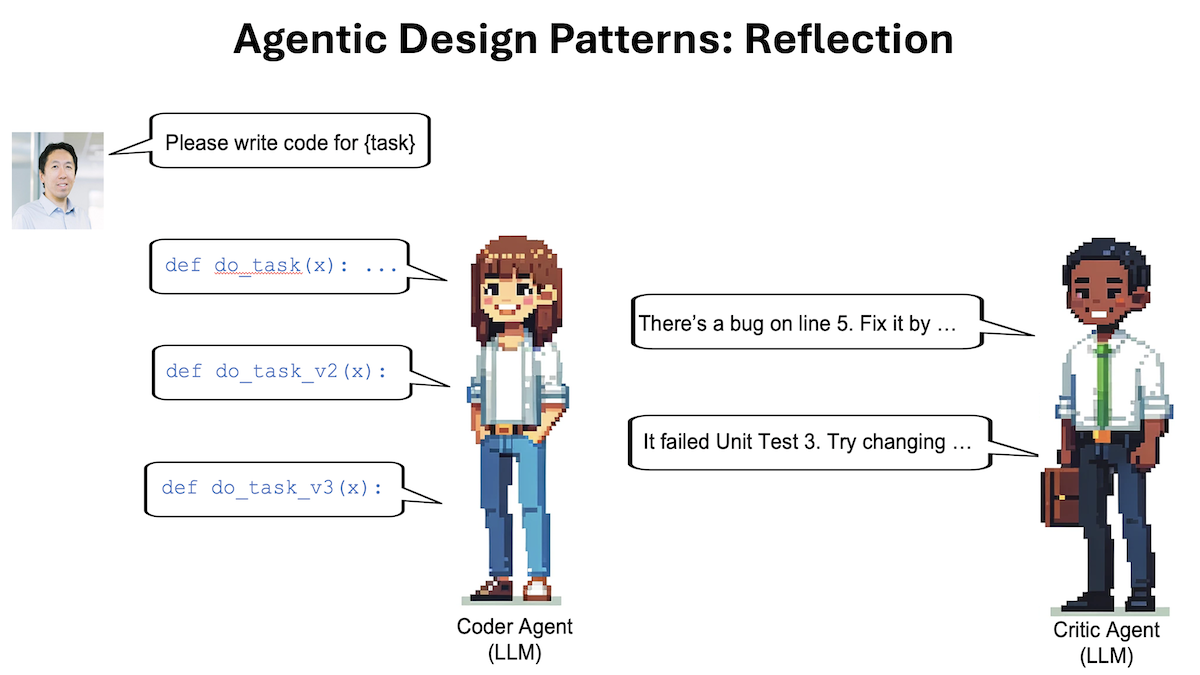
Dear friends, 亲爱的朋友们，

Last week, I described four design patterns for AI agentic workflows that I believe will drive significant progress this year: Reflection, Tool use, Planning and Multi-agent collaboration. Instead of having an LLM generate its final output directly, an agentic workflow prompts the LLM multiple times, giving it opportunities to build step by step to higher-quality output. In this letter, I'd like to discuss Reflection. For a design pattern that’s relatively quick to implement, I've seen it lead to surprising performance gains.   
上周，我描述了四种我认为今年将推动重大进展的AI代理工作流设计模式：反思、工具使用、规划和多智能体协作。与直接由LLM生成最终输出不同，代理工作流多次提示LLM，给予其逐步提高输出质量的机会。在这封信中，我想讨论反思。对于一个相对快速实施的设计模式，我看到它带来了令人惊讶的性能提升。  
  
You may have had the experience of prompting ChatGPT/Claude/Gemini, receiving unsatisfactory output, delivering critical feedback to help the LLM improve its response, and then getting a better response. What if you automate the step of delivering critical feedback, so the model automatically criticizes its own output and improves its response? This is the crux of Reflection.   
您可能有过这样的经历：提示ChatGPT/Claude/Gemini，收到不满意的输出，提供关键反馈以帮助LLM改进其回答，然后得到更好的回答。如果您自动化关键反馈的步骤，让模型自动批评自己的输出并改进回答，会怎样呢？这就是Reflection的关键所在。  
  
Take the task of asking an LLM to write code. We can prompt it to generate the desired code directly to carry out some task X. After that, we can prompt it to reflect on its own output, perhaps as follows:  
请要求一个LLM编写代码的任务。我们可以提示它直接生成所需的代码来执行某个任务X。之后，我们可以提示它反思自己的输出，可能如下所示：

Here’s code intended for task X: [previously generated code]      
这是用于任务X的代码：[之前生成的代码]  
Check the code carefully for correctness, style, and efficiency, and give constructive criticism for how to improve it.  
仔细检查代码的正确性、风格和效率，并提出建设性的批评以改进它。  
  
Sometimes this causes the LLM to spot problems and come up with constructive suggestions. Next, we can prompt the LLM with context including (i) the previously generated code and (ii) the constructive feedback and (iii) ask it to use the feedback to rewrite the code. This can lead to a better response. Repeating the criticism/rewrite process might yield further improvements. This self-reflection process allows the LLM to spot gaps and improve its output on a variety of tasks including producing code, writing text, and answering questions.   
有时这会导致LLM发现问题并提出建设性的建议。接下来，我们可以提供上下文给LLM，包括（i）之前生成的代码和（ii）建设性的反馈，然后（iii）要求它使用反馈来重写代码。这可能会导致更好的回应。重复批评/重写的过程可能会带来进一步的改进。这个自我反思的过程使LLM能够发现差距，并在各种任务中改进其输出，包括生成代码、写作文本和回答问题。



And we can go beyond self-reflection by giving the LLM tools that help evaluate its output; for example, running its code through a few unit tests to check whether it generates correct results on test cases or searching the web to double-check text output. Then it can reflect on any errors it found and come up with ideas for improvement.  
我们可以通过给予LLM工具来评估其输出来超越自我反思；例如，通过运行其代码通过一些单元测试来检查它是否在测试案例上生成正确的结果，或者搜索网络来双重检查文本输出。然后它可以反思任何发现的错误，并提出改进的想法。

Further, we can implement Reflection using a multi-agent framework. I've found it convenient to create two different agents, one prompted to generate good outputs and the other prompted to give constructive criticism of the first agent's output. The resulting discussion between the two agents leads to improved responses.  
此外，我们可以使用多代理框架来实现反思。我发现创建两个不同的代理很方便，一个被促使生成良好的输出，另一个被促使对第一个代理的输出提出建设性的批评。两个代理之间的讨论会导致改进的回应。

Reflection is a relatively basic type of agentic workflow, but I've been delighted by how much it improved my applications’ results in a few cases. I hope you will try it in your own work. If you’re interested in learning more about reflection, I recommend these papers:  
反射是一种相对基础的代理工作流程类型，但在一些情况下，我对它如何改善我的应用结果感到非常高兴。我希望你在自己的工作中尝试一下。如果你对学习更多关于反射的内容感兴趣，我推荐阅读以下论文：

* “[Self-Refine: Iterative Refinement with Self-Feedback](https://arxiv.org/abs/2303.17651?utm_campaign=The Batch&utm_source=hs_email&utm_medium=email&_hsenc=p2ANqtz-9dHVnW1I1bA3sPBbsikjT165Qez3QiiAssknCERwgki818YHG7PyHOQSgg-nxKDa0BuE7B" \t "https://www.deeplearning.ai/the-batch/issue-242/_blank),” Madaan et al., 2023  
  “自我完善：带有自我反馈的迭代改进”，Madaan等人，2023年
* “[Reflexion: Language Agents with Verbal Reinforcement Learning](https://arxiv.org/abs/2303.11366?utm_campaign=The Batch&utm_source=hs_email&utm_medium=email&_hsenc=p2ANqtz-9dHVnW1I1bA3sPBbsikjT165Qez3QiiAssknCERwgki818YHG7PyHOQSgg-nxKDa0BuE7B" \t "https://www.deeplearning.ai/the-batch/issue-242/_blank),” Shinn et al., 2023  
  “反思：具有口头强化学习的语言代理”，Shinn等人，2023年
* “[CRITIC: Large Language Models Can Self-Correct with Tool-Interactive Critiquing](https://arxiv.org/abs/2305.11738?utm_campaign=The Batch&utm_source=hs_email&utm_medium=email&_hsenc=p2ANqtz-9dHVnW1I1bA3sPBbsikjT165Qez3QiiAssknCERwgki818YHG7PyHOQSgg-nxKDa0BuE7B" \t "https://www.deeplearning.ai/the-batch/issue-242/_blank),” Gou et al., 2024  
  “ CRITIC: 大型语言模型可以通过工具交互式批评进行自我纠正，” Gou等人，2024年

I’ll discuss the other agentic design patterns in future letters.  
我将在未来的信件中讨论其他的主动设计模式。

Keep learning! 不断学习！