比较 AI 系统设计

预计时间：5 分钟

在设计 AI 系统时，正确的方法取决于任务的复杂性、适应性需求和操作要求。让我们比较三种范式：**单一 LLM 特性**、**结构化工作流**和**自主 AI 代理**。

完成本阅读后，您将能够：

- 区分单一 LLM 特性、结构化工作流和自主 AI 代理。
- 确定每种 AI 系统设计的适用用例和局限性。
- 评估哪种范式最适合特定任务的复杂性和适应性需求。
- 识别混合 AI 系统设计的现实世界趋势。

单一 LLM 特性：简单的一次性任务

想象一下，您想快速总结一篇新闻文章或翻译客户评价。您只需将文本输入到一个单一的 LLM 中，即可立即获得总结或翻译的结果——无需进一步步骤。在最基本的层面上，您可以使用 LLM 处理简单的一次性任务，而无需在调用之间保持记忆或上下文。

关键特征

单一 LLM 特性具有以下关键特征：

- **无状态处理**：在交互中不保留信息或上下文。
- **直接输入输出流**：简单的请求–响应机制。
- **预定义任务**：仅适合于明确的单步操作。

最佳用途

此范例最适合：

- 简单、定义明确的任务，无需记忆或多步骤逻辑

示例

该范式适用于：

- 文本摘要
- 情感分类
- 信息提取
- 翻译

优势

使用单一 LLM 特性提供：

- **速度和简单性**：构建和运行速度最快
- **确定性输出**：相同输入，相同输出
- **低成本**：计算和协调开销最小

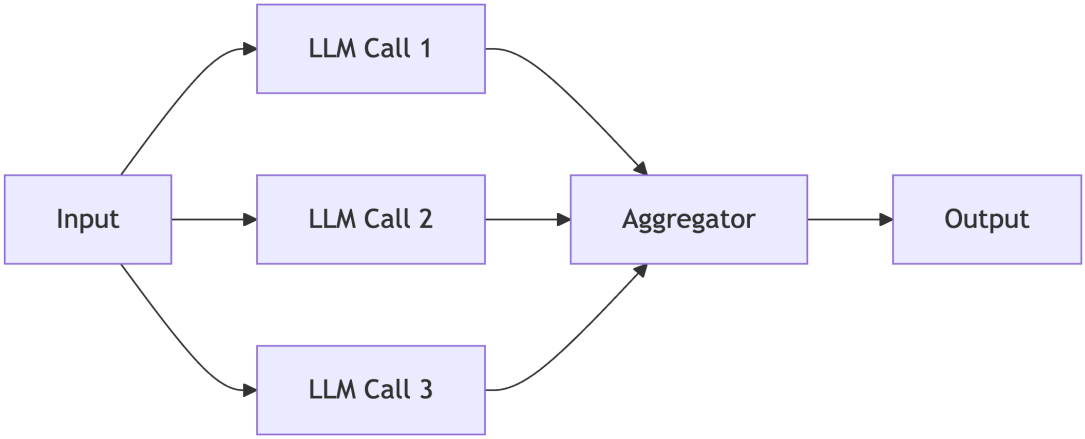
限制

使用单一的 LLM 时，您可能会遇到以下限制：

- **没有适应性**：无法处理上下文或动态决策
- **没有记忆**：每个输入都是独立处理的

结构化工作流程：多步骤、可预测的过程

结构化工作流程 通过明确的、确定性的代码路径协调 LLM 和工具调用。它们非常适合重复的、多步骤的或合规性要求高的任务。考虑处理保险索赔的过程，其中每个文档都被扫描、信息被提取、验证并存储。每个步骤必须遵循精确、可预测的顺序，这使得结构化工作流程成为理想选择。



关键特征

结构化工作流具有以下关键特征：

- **确定性执行**：输入产生一致的输出。
- **明确的控制流**：所有步骤和决策都是预定义的。
- **预定义的工具链**：工具的使用是固定且透明的。

最佳用途

结构化工作流程适用于以下需求：

- 具有明确逻辑和最小模糊性的重复性多步骤任务
- 受监管或合规驱动的应用程序
- 需要一致性、可追溯性和可审计性的场景

示例

您会发现结构化工作流程在以下场景中效果很好：

- 文档和数据管道（光学字符识别（OCR）→ 提取 → 验证 → 存储）
- 批量报告生成
- 财务和医疗交易处理

优势

结构化工作流程提供以下优势：

- **可预测和可靠**：易于监控、调试和审计
- **成本效益**：没有不必要的探索
- **合规准备**：支持版本控制、错误处理和审计跟踪

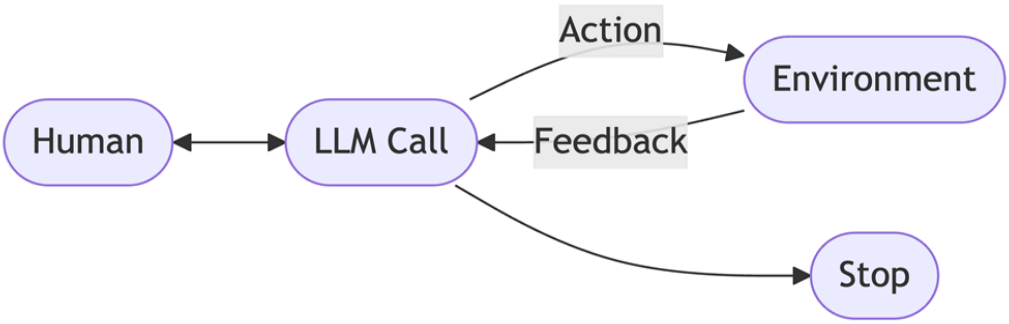
限制

在使用结构化工作流时，您可能会遇到以下限制：

- **僵化性**：难以适应新的或模糊的场景
- **开发开销**：需要为每个例外或变体编写代码

自主代理：灵活的、上下文感知的推理

自主代理 使大型语言模型能够规划一系列行动并根据条件变化进行调整。代理根据实时上下文和反馈选择使用哪些工具以及如何实现目标。想象一下，一个由人工智能驱动的虚拟助手帮助用户规划假期。它动态地收集用户偏好，研究目的地，建议住宿，并根据反馈调整建议。这需要一个能够进行规划、具备上下文感知能力和迭代改进的自主代理。



核心能力

自主代理具有以下核心能力：

- **动态规划**：分解目标并根据需要调整步骤
- **情境意识**：记住过去的步骤，并根据用户和环境反馈进行调整
- **工具协调**：动态选择工具并改变策略

最佳用途

您可以将自主代理用于以下需求：

- 复杂的、开放式的任务，解决路径不明确
- 需要实时适应和推理的场景
- 具有高度变异性或需要个性化的环境

示例

考虑为以下用途实施自主代理：

- 研究代理合成新信息
- 自适应客户支持和故障排除
- 基于反馈迭代优化结果的自动化

优势

自主代理提供以下优势：

- **高度适应性**：处理不可预见的情况
- **动态决策**：随着时间的推移进行迭代和改进
- **减少人类干预**：自主管理复杂性

限制

自主代理也可能面临以下挑战：

- **不可预测的结果**：需要强有力的监控和保障措施
- **更高的复杂性和成本**：更难以调试和保证合规性

现在您已经熟悉了这三种范式，请查看以下总结表，以便并排比较这些范式。

摘要表

下表比较了三种人工智能系统，它们的过程、使用案例以及优缺点。

人工智能系统类型	过程	使用案例	优点	缺点
单一大型语言模型	输入 → 大型语言模型 → 输出	摘要、分类	简单、快速、低成本	不可适应，缺乏上下文
工作流	并行大型语言模型 → 聚合 → 输出	结构化多步骤任务	可预测，易于审计	僵化，不够动态
代理	计划 → 行动 → 观察 → （重复代理循环）	复杂、自适应自动化	灵活，从反馈中学习	不可预测，复杂，成本较高

现实世界的实施实践

在实践中，混合架构是常见的。它们结合了工作流的可靠性和代理的灵活性，以实现最佳结果。

最近的标准，包括来自Anthropic的模型上下文协议（Model Context Protocol，简称MCP）和来自IBM的代理通信协议（Agent Communication Protocol，简称ACP），简化了两种方法在大规模下的集成、监控和治理。

关键要点

在选择 AI 代理时，请考虑以下因素：

- **从简单开始**：使用最直接的解决方案来满足您的需求。例如，您可以使用单一 LLM 功能来满足基本需求。
- **利用工作流程**：当可预测性、合规性和效率很重要时
- **选择性部署代理**：仅在需要适应性、复杂推理或开放式问题解决时

作者

Faranak Heidari