



研讨/分享/感悟

仓颉社区中的灵感碰撞 社区先行者的干货分享

CangChain架构与Agent

主讲人：杨海龙



CangChain的进展汇报和2024计划



2023年CangChain的开发总结:

- 与大语言模型的交互, 包括Openai的chatgpt
- Onnx模型推理, 实现在CPU和GPU上的运行onnx模型
- Embedding计算
- 本地向量数据库的访问
- google search tool

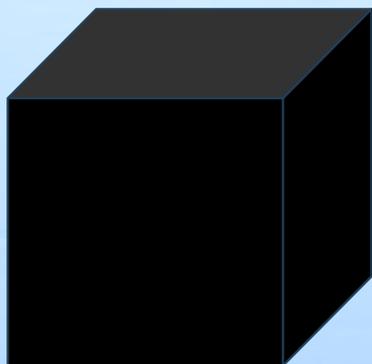
2024年CangChain的开发计划:

- Onnx模型的微调, 提供C语言接口方便仓颉调用;
- 向量数据库封装为tool
- 内置更多的tool, arxiv tool, wiki tool, Mathematica tool;
- 内置更多的智能体, 基于规则的智能体, 多智能体;
- 智能体领域专用语言, Agent DSL;
- 带有隐私保护的协同机制;

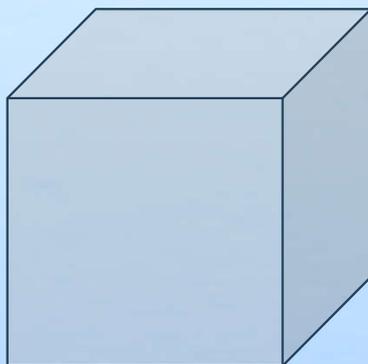
WORKSHOP

黑盒系统

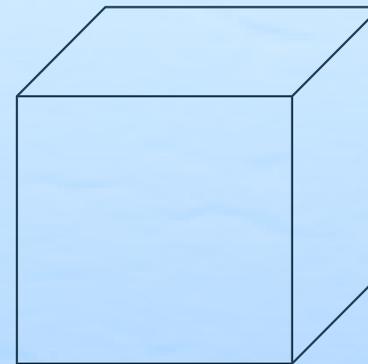
数据驱动的黑盒模型



数据驱动和知识驱动的灰盒模型



完全由知识驱动的黑盒模型

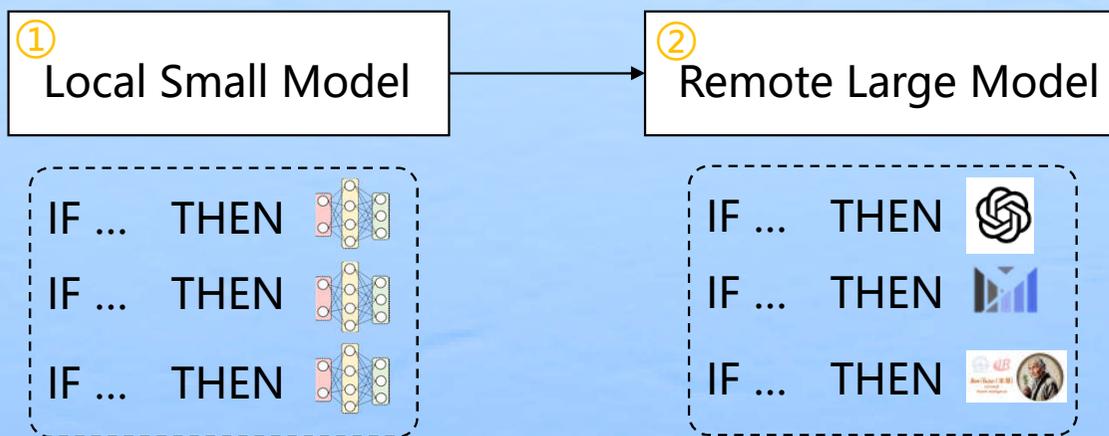
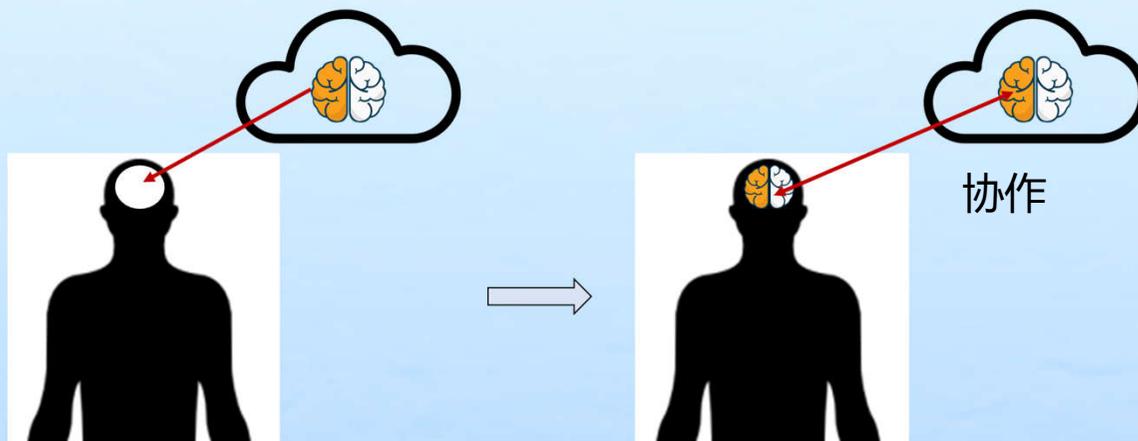


```
num = float(input("输入一个特征: "))  
if num > 0:  
    ■  
elif num == 0:  
    ■  
else:  
    ■
```

```
num = float(input("输入一个特征: "))  
if num > 0:  
    print("正样本")  
elif num == 0:  
    print("零")  
else:  
    print("负样本")
```

WORKSHOP

CangChain的基于规则的智能体



CangChain的Agent DSL

链式调用

```
let agent = Agent()  
agent.setLLM("GPT3.5")  
agent.addTool("googlesearch")  
agent.addTool("kgtool")  
agent.addTool("formulatool")  
agent.setPrompt(".....")  
agent.setCallback()
```



```
Agent().setLLM("GPT3.5").addTool("googlesearch").addTool("kgtool").addTool("formulatool").setPrompt(".....").setCallback()
```

多智能、planner以及自定义的思考范式?

```
agent.copy().setPrompt("...") //新的agent  
Planner().t1().t2().t3() // 带有时序的链式调用  
agent.reason().action().finishWhen()
```

CangChain的Agent DSL

写一个结构化的优质的Prompt是一件非常困难的事。

@role() //指定负责执行任务的个体或团队的角色或职责。这有助于明确谁将执行任务，他们的专业背景或角色是什么。

@background()//提供任务的背景信息或上下文。这有助于理解为什么要执行这个任务，以及它与更广泛的目标或项目的关系。

@attention() //指定在执行任务时需要特别关注的领域或细节。这有助于确保重要的方面不会被忽视。

@profile() //描述执行任务的个体或团队的特征，可能包括经验、技能或其他相关特质。

@skills() //确定完成任务所需的具体技能或能力。这有助于明确是否具备执行任务所需的资源和能力。

@goals() //描述任务的最终目标或预期结果。这有助于保持工作的聚焦和方向。

@constrains() //说明在执行任务过程中可能遇到的限制或挑战，如时间、预算或资源的限制。

@workflow() //描述完成任务的步骤或过程。这有助于组织任务的执行，确保效率和一致性。

@outputFormat() //指定任务完成后的结果应该呈现的格式或样式，例如报告、演示文稿或数据表。

@suggestions() //提供有关如何改进或最优化任务执行的建议。这可能基于以往的经验或最佳实践。

@initialization() //定义任务开始时的初始设置或条件。这可能包括任务的起始状态、初始资源分配或首要步骤。

通过元编程的宏函数来生成提示词模板代码

WORKSHOP



研讨/分享/感悟

仓颉社区中的灵感碰撞 社区先行者的干货分享

Thanks!



添加“编程语言小助手”，
获取仓颉相关资讯!

