Analytics Vidhya

Introduction to LangGraph

Instructor

Dipanjan Sarkar

Head of Community & Principal AI Scientist at Analytics Vidhya

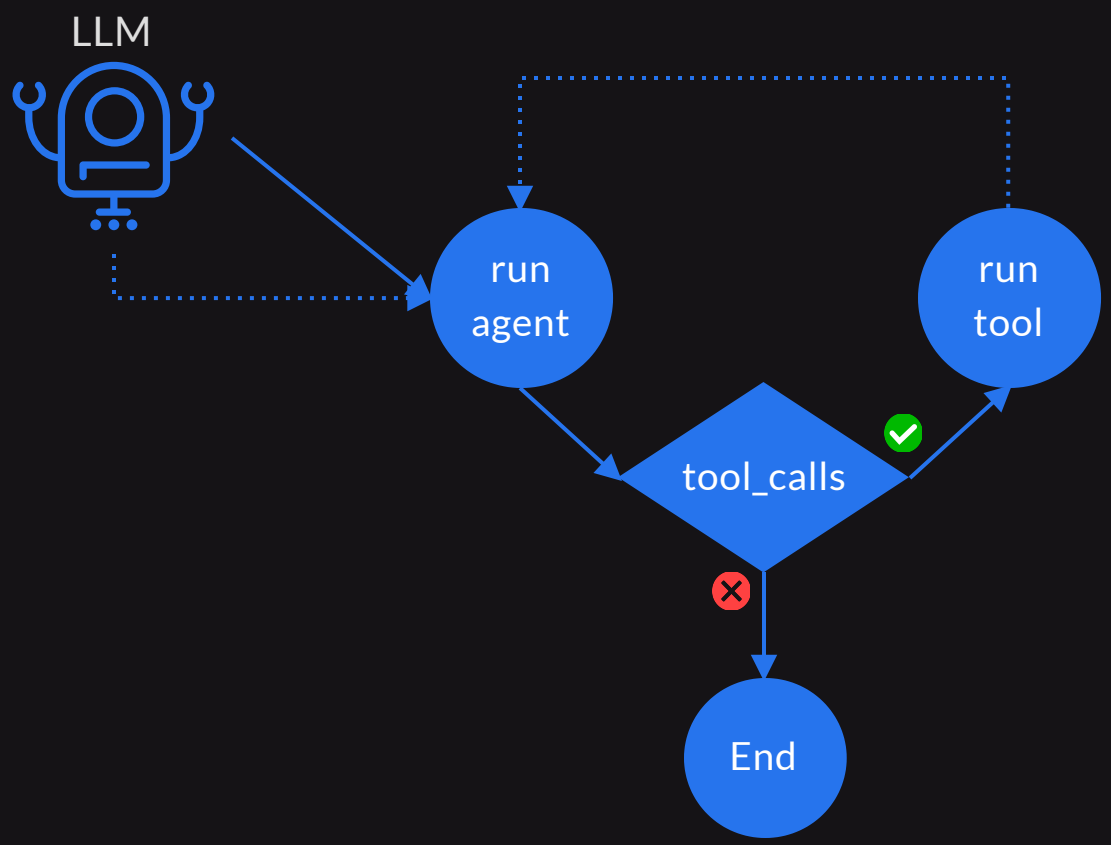
Google Developer Expert - ML & Cloud Champion Innovator

Published Author

Outline

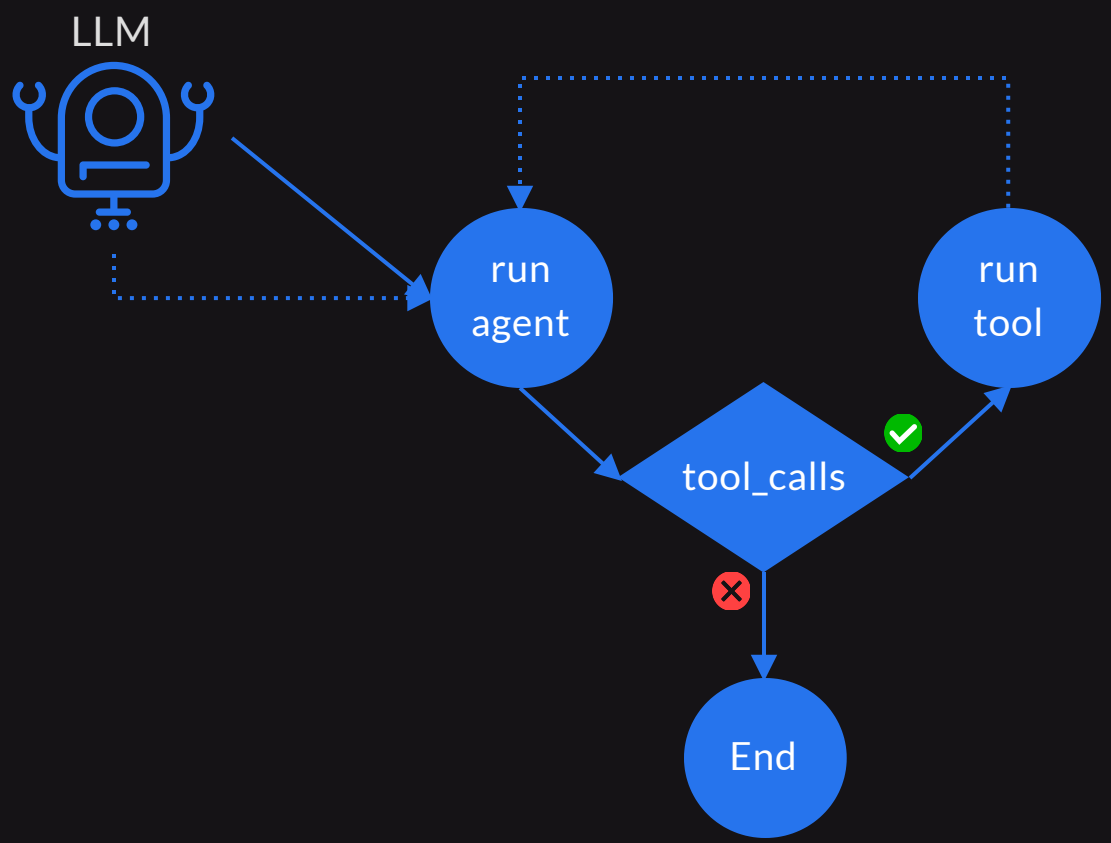
- Understanding LangGraph
- LangGraph Components
- LangGraph and AI Agents
- LangGraph - Agents and State Machines
- LangGraph for Multi-Agent Workflows

Understanding LangGraph



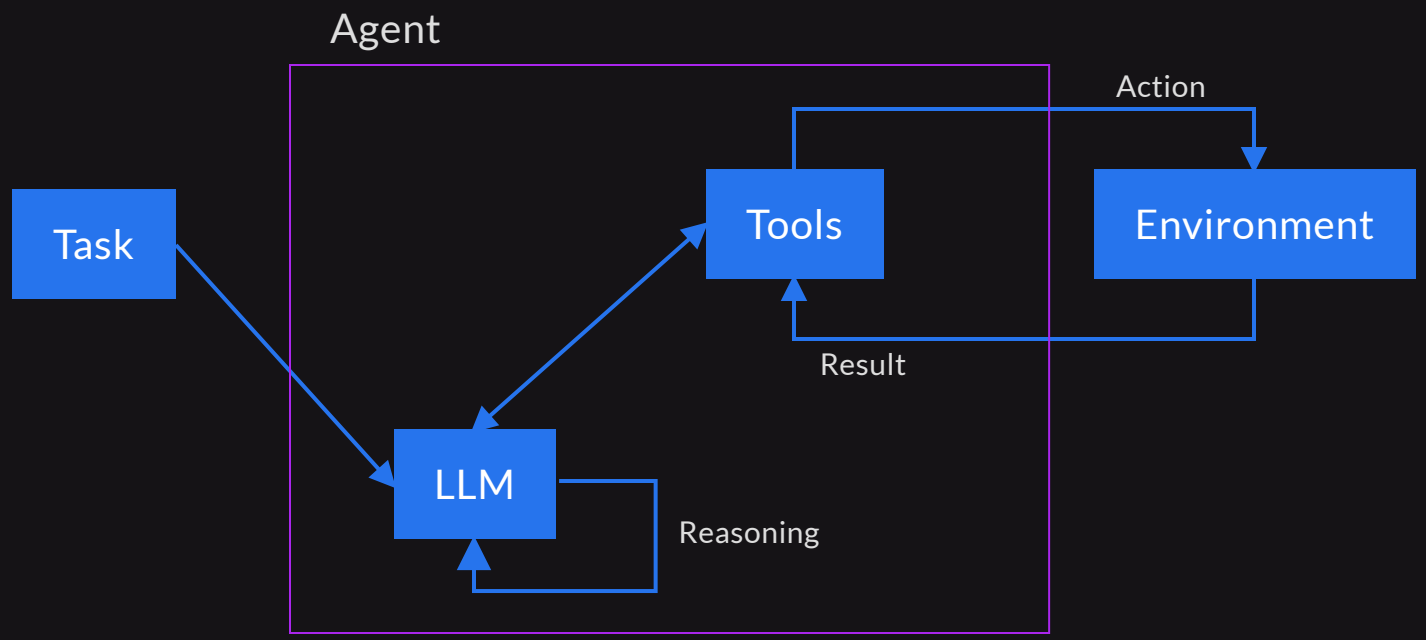
- LangGraph, built on top of LangChain, facilitates the creation of cyclical graphs essential for developing AI agents powered by LLMs.
- Its interface is inspired by the widely-used NetworkX library.
- It enables coordination and checkpointing of multiple chains (or actors) through cyclic computational steps.

LangGraph Components



- LangGraph treats Agent workflows as a cyclical Graph structure
- Main features
  - Nodes: Functions or LangChain Runnable objects such as tools.
  - Edges: Specify directional paths between nodes
  - Stateful Graphs: Manage and update state objects while processing data through nodes.
- LangGraph leverages this to facilitate cyclical LLM call executions with state persistence which is often required for AI Agents

LangGraph and AI Agents



- Complex LLM applications often use cycles during execution
- These cycles often use the LLM to reason about what to do next in the cycle like Chain of Thought
- This can essentially be thought of as running an LLM in a for-loop
- These types of systems are commonly called AI agents
- LangGraph is the perfect framework to model these agents as Graphs

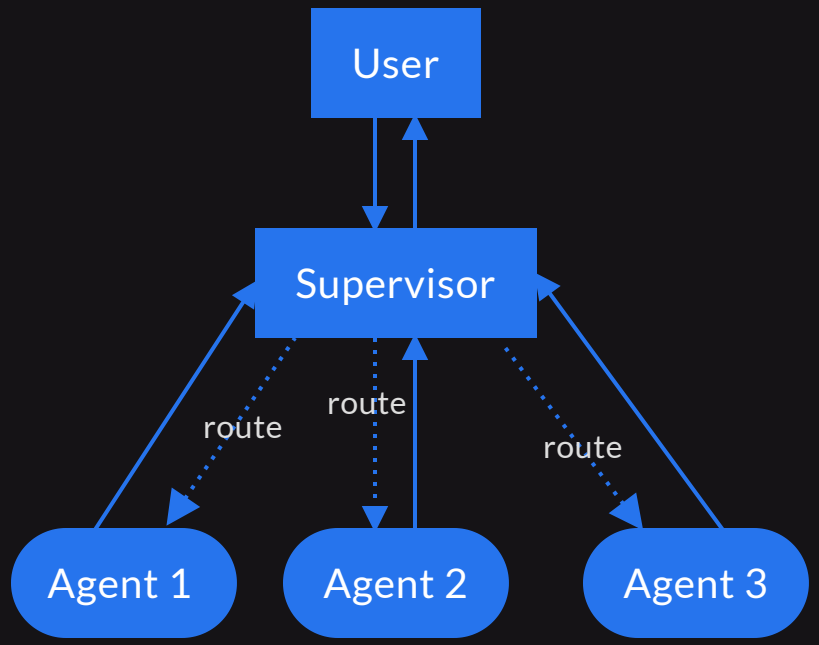
LangGraph - Agents and State Machines

- Agents rely entirely on the LLM, offering limited user control.
- State machines, on the other hand, provide greater control over the tools an agent can access, acting as enhanced versions of agents.

Both agents and state machines can be modeled as graphs within LangGraph.

Analytics Vidhya

LangGraph for Multi-Agent Workflows



- Multi-Agent workflows involve multiple independent agents powered by LLMs connected in a specific way
- Each agent can have its prompt, LLM, tools, and other custom code to collaborate with the other agents if needed
- LangGraph makes each agent a node in the graph. The control flow is managed by edges, and they communicate by adding to the graph's state

Thank You

Analytics Vidhya