

Lecture 10: Introduction to LangChain

Learning Objectives

By the end of this lecture, you should be able to:

- Understand what LangChain is and why it's used for building LLM applications.
 - Learn core LangChain concepts: chains, agents, tools, and memory.
 - Build a basic LangChain-powered agent that uses tools and memory.
 - Navigate LangChain's modular architecture for agentic development.
-

Key Concepts

What is LangChain?

- A Python framework designed to **orchestrate LLMs** in complex workflows.
- Abstracts away the boilerplate for chaining calls, using tools, and managing memory.
- Supports multiple model providers (OpenAI, Cohere, Hugging Face, etc.).

LangChain Components

- **LLM**: Interface to call a language model.
 - **PromptTemplate**: Formats user input into a model-friendly prompt.
 - **Chains**: A sequence of calls (e.g., input → LLM → output).
 - **Agents**: Decision-making loop that uses tools and memory.
 - **Tools**: External functions (e.g., search, calculator) that the agent can call.
 - **Memory**: Stores context and conversation history.
-

Required Tools/Libraries

- Python
- OpenAI API key (or other model provider)
- LangChain

```
pip install langchain openai
```

Hands-on Exercise: Build a LangChain Agent

Goal: Create a basic agent that answers questions using reasoning and tool calls.

Step 1: Import LangChain components

```
from langchain.agents import initialize_agent, load_tools
from langchain.llms import OpenAI

llm = OpenAI(temperature=0)
tools = load_tools(["serpapi", "llm-math"], llm=llm)
```

Step 2: Initialize the agent

```
agent = initialize_agent(
    tools=tools,
    llm=llm,
    agent="zero-shot-react-description",
    verbose=True
)
```

Step 3: Run a task

```
agent.run("What is the square root of 2025, and who is the current president of the USA?")
```

Step 4: Observe outputs

- Look at the intermediate thoughts, actions, and observations.
- How does the agent choose which tool to use?

Bonus:

- Add memory using [ConversationBufferMemory](#).
 - Try chaining multiple tools or customizing tool behavior.
 - Build your own tool (e.g., weather lookup or file reader) and register it with the agent.
-