Lecture 14: Agents with Memory and Goals

🎯 Learning Objectives

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

- Understand how to design agents that pursue long-term goals.
- Implement episodic and persistent memory for agents.
- Track agent state and goal progression over time.
- Build an agent that remembers context and acts toward objectives.

🧩 Key Concepts

Goal-Oriented Agents

- Operate with a defined **objective** or set of **subgoals**.
- Make decisions not just based on the current prompt, but on **overall intent**.
- Maintain a memory of past thoughts, actions, and observations.

Memory Types in Goal-driven Agents

- **Episodic Memory**: Captures the sequence of interactions or tasks within a session.
- **Long-Term Memory**: Persists across sessions using a vector database.
- **Working Memory**: Context window of the current reasoning step.

Agent State Representation

- **Current Goal**: What the agent is actively pursuing.
- **Completed Tasks**: Progress tracking.
- **Context Memory**: Reference of previously retrieved or created content.

🛠️ Required Tools/Libraries

- LangChain
- ChromaDB or FAISS
- OpenAI API or LLM of choice
- Python

🔬 Hands-on Exercise: Build a Goal-Driven Agent with Memory

Goal: Create an assistant that helps a user draft a report over multiple steps and remembers what was done.

Step 1: Initialize a memory buffer

```
from langchain.memory import ConversationBufferMemory
memory = ConversationBufferMemory()
```

Step 2: Define agent with memory

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

llm = OpenAI()
tools = load_tools(["llm-math"], llm=llm)

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

Step 3: Run multi-step conversation

```
agent.run("Let's start drafting a research report on climate change.")
agent.run("Add a section on recent temperature trends.")
agent.run("What sections have we written so far?")
```

Step 4: Observe memory in action

```
print(memory.buffer)
```

Bonus:

- Save long-term memory to FAISS for later sessions.
- Introduce goal check-ins: the agent re-evaluates its strategy after each major step.
- Visualize memory as a timeline of actions and decisions.
