LangChain Academy: Long-Term Memory in LangGraph

Introduction

This module explores memory concepts in LangGraph, focusing on long-term memory. Memory is a cognitive function that allows agents to store, retrieve, and utilize information across interactions. This document will provide an overview of short-term and long-term memory in LangGraph, their use cases, and implementation techniques.

---

1. Memory in LangGraph

LangGraph provides structured memory management, categorized as:

- **Short-term memory:** Scoped within a session (thread), useful for conversational history and chat interruptions.
- **Long-term memory:** Persists across multiple sessions, allowing information retention about users and previous interactions.

Feature	Short-Term Memory	Long-Term Memory
Scope	Within a session	Across sessions
Use Case	Maintaining conversation history within a single chat	Remembering user preferences over multiple interactions
LangGraph Usage	Checkpointer	Store

---

2. Short-Term Memory

- Used for temporary storage within a single session.
- Implemented via thread-scoped memory.
- Supports conversational context retention and human-in-the-loop interactions.

---

3. Long-Term Memory

Long-term memory in LangGraph follows different structuring methodologies:

3.1 Semantic Memory

- Stores factual knowledge and structured information.
- Organized into various formats:
    - **Profile Memory:** Single document representation of user data.
    - **List Memory:** Maintains a collection of documents, allowing incremental updates.

Type	Pros	Cons
Profile	Easy retrieval	Hard to maintain as it grows
List	Scalable, modular	Retrieval complexity increases with size

3.2 Episodic Memory

- Stores past actions and reasoning paths.
- Useful for agents requiring prior knowledge of interactions.
- Techniques:
    - Few-shot prompting to recall previous decisions.
    - Contextual retrieval of past trajectories.

3.3 Procedural Memory

- Stores learned behaviors and workflows.
- Helps agents improve responses over time by updating instructions dynamically.
- Updated using:
    - Prior reasoning trajectories.
    - Tool-based learning frameworks.

---

4. Updating Long-Term Memory

- Memory can be updated at different stages:
    - **Hot-path Updates:** During runtime (e.g., ChatGPT), offering real-time learning but increasing latency.
    - **Background Updates:** Processes memory separately, reducing UX impact but requiring optimal frequency tuning.

Type	Pros	Cons
Hot-path	Real-time updates	Can slow response time
Background	Lower UX impact	Needs optimized frequency

Conclusion

Understanding and utilizing memory in LangGraph enhances AI capabilities by retaining important information across interactions. Implementing structured memory strategies like episodic, semantic, and procedural memory helps improve agent intelligence and responsiveness.