Day96_AgenticAI_Reasoning_and_Building_Simple_Agents

October 13, 2025

Agentic AI – Complete Guide

Today's Learning: Agentic AI – Theory and Concepts

Today we are learning **all the theory and concepts of Agentic AI** from scratch.
We will cover:

- What Agentic AI is and its key characteristics
- Basic concepts: Agents, Agency, Tools, and Reasoning
- How to build a **Simple AI Agent**
- How to extend it into a **Reasoning Agent** with tools
- Introduction to **Multimodal Agents**
- Popular **frameworks and APIs** for building agents
- Recommended **learning path** to master agentic AI

By the end of this notebook, you will have a **complete understanding of agentic AI concepts** and be able to start building your own AI agents.

1 Introduction

Agentic AI refers to AI systems that act autonomously to achieve goals by perceiving their environment, making decisions, and executing actions. Unlike simple AI that responds passively, agentic AI can **plan, reason, and take sequential actions** to complete tasks.

1.1 Key characteristics of agentic AI:

- **Autonomy:** Works independently to achieve goals.
- **Reactivity:** Perceives the environment and reacts to changes.
- **Proactiveness:** Plans future actions to achieve objectives.
- **Goal-Oriented:** Focused on achieving predefined or dynamically set goals.

1.2 Example:

- A robot vacuum planning a cleaning path dynamically.
- AI financial assistant recommending and executing trades automatically.

2 Key Concepts

2.1 Agent

An agent is an AI entity that perceives its environment through sensors and acts through actuators. Agents can be: - **Simple reflex agents:** Take actions based on current inputs.
- **Model-based agents:** Maintain internal state to handle partially observable environments.
- **Goal-based agents:** Plan actions to achieve a specific goal.

2.2 Agency

Agency is the **capacity of an AI system to act independently**, take initiatives, and make decisions that affect outcomes.

2.3 Tools

Tools are **external functions or modules** that an agent can use to perform tasks (e.g., calculator, web search, translation).

2.4 Reasoning

Reasoning is the ability to **analyze information, plan steps, and draw conclusions** to achieve goals.

2.4.1 Types of reasoning:

- **Deductive:** From general rules to specific conclusions.
- **Inductive:** From specific examples to general rules.
- **Abductive:** Inferring the best explanation for observed facts.

3 Simple AI Agent

A **simple agent** is a program that can take an input and perform actions without complex reasoning.

3.1 Example: Calculator Agent

Concept:
- Takes a mathematical expression as input.
- Computes the result.
- Returns output.

Python Example:

```python
# Simple Calculator Agent
def simple_calculator(expression):
    try:
        return eval(expression)
    except Exception as e:
        return f"Error: {e}"

result = simple_calculator("15*7 + 10")
print("Simple Agent Result:", result)
```

4 Reasoning Agent

A **reasoning agent** can plan multiple steps to achieve a goal, often using tools and external knowledge sources.

4.1 Example Goal:

- "Calculate 15 * 7 and then add 10"

4.2 Steps:

1. Use calculator tool to compute `15*7`.
2. Add `10` to the result.
3. Return final answer.

4.3 Python Example:

```python
# Reasoning Agent with Tool
class Tool:
    def __init__(self, name, func):
        self.name = name
        self.func = func

class ReasoningAgent:
    def __init__(self, name):
        self.name = name
        self.tools = {}

    def add_tool(self, tool):
        self.tools[tool.name] = tool.func

    def run(self, goal):
        if "calculate" in goal.lower():
            expression = "15*7 + 10"
            return self.tools["Calculator"](expression)
        return "Goal not recognized."
```

```python
# Define calculator tool
calculator = Tool("Calculator", lambda x: eval(x))

# Initialize agent
agent = ReasoningAgent("LocalReasoningAgent")
agent.add_tool(calculator)

goal = "Calculate 15 * 7 and then add 10"
response = agent.run(goal)
print("Reasoning Agent Response:", response)
```

5 Multimodal Agent

A **multimodal agent** can handle **different types of input and output**, e.g., text, images, audio.

5.1 Example:

- Input: Text query or image of a handwritten math problem.
- Output: Text answer or solved equation.

5.2 Python Concept Example:

```python
# Example conceptual pseudo-code
class MultimodalAgent:
    def __init__(self):
        pass

    def handle_input(self, input_data):
        if isinstance(input_data, str):
            return f"Text processed: {input_data}"
        elif isinstance(input_data, bytes):
            return "Image/audio processed"
        else:
            return "Unknown format"

agent = MultimodalAgent()
print(agent.handle_input("Solve 15*7"))
```

6 Agent Frameworks & APIs

6.1 Popular Agent Frameworks:

1. **AGNO:** Easy multi-model support, can connect Groq, OpenAI, custom tools.
2. **LangChain:** Tool-based agent building for LLMs.
3. **AutoGPT / BabyAGI:** Fully autonomous AI agents.
4. **Hugging Face Transformers:** LLMs for custom reasoning and multimodal AI.

6.2 Using AGNO:

- **Installation:**

```
pip install agno python-dotenv
```

- **Initialize an agent:**

```
from agno.agent import Agent
from agno.models.groq import Groq
from dotenv import load_dotenv
load_dotenv()

agent = Agent(
    model=Groq(id="llama-3.3-70b-versatile"),
    description="You are a news reporter summarizing news in 3 lines with URLs",
    markdown=True
)

agent.print_response("Tell me the most latest breaking political news from the USA")
```

6.3 AGNO Advantages:

- Supports **multiple backends** (Groq, OpenAI).
- Can **add custom tools**.
- Provides **online reasoning** capabilities.

7 Learning Path for Agentic AI

1. **Foundations:**
    - Python programming
    - Basics of AI and ML
    - LLM fundamentals
2. **Agents:**
    - Simple reflex and reasoning agents
    - Using external tools
3. **Frameworks & APIs:**
    - AGNO, LangChain, Hugging Face
    - Tool integration and multimodal inputs
4. **Advanced Topics:**
    - Autonomous multi-step agents
    - Planning and goal-oriented reasoning
    - Real-world deployment

8 Summary

- Agentic AI focuses on autonomy, reasoning, and goal-oriented actions.
- **Simple agents** execute tasks; **reasoning agents** plan multi-step actions.
- **Multimodal agents** handle different types of inputs.
- **Frameworks like AGNO** simplify agent building with custom tools and reasoning support.
- Learning path: Foundations → Agents → Frameworks → Advanced deployment.

“‘

[ ]: