Agentic Architecture

Robert Cunningham

12/2/2024

Introduction

This paper presents a mathematical framework for modeling interactions between a user and an AI agent equipped with a set of tools. This framework formalizes the agent's internal processes, tool selections, and responses over multiple interaction cycles. By representing the communication and tool usage in terms of matrices and tensors, the paper introduces a proposition stating that the entire interaction is uniquely determined by four artifacts: the chat history h, the tool queries array $\hat{q}$, the tool indices matrix σ, and the tool results array $\hat{r}$. These artifacts are both necessary and sufficient to fully reconstruct the interaction, effectively serving as a hash of the agent's behavior.

Definitions

Let A be an **agent** with **tools**, $T = \{T_1, T_2, \ldots, T_n\}$. Suppose an initial **user input**, u_1, is sent to A through *chat*, and that A, having been instructed by its **system prompt**, s_A, determines that T_i should be called. Then A will construct its first query, q_1, and call T_i with it. The result,

$$r_1^{(i)} = T_i(q_1)$$

will then be sent back to the agent to be interpreted via the agent's internal chat, *chat** (unless the tool is configured to return its output directly to the user). The **value** of this output, v_1 (where either $v_1 = r_1^{(i)}$ or $v_1 = \mathit{chat}^*(r_1^{(i)})$) is considered the agent's response to u_1, and is thus added to the **chat history**, h, in the form of a cycle, $c_1 = (u_1, v_1)$. Thus,

$$h = [c_1] = [(u_1, v_1)]$$

And throughout k such cycles,

$$h = [c_i]_{i=1}^{k} = [(u_1, v_1), (u_2, v_2), \ldots, (u_k, v_k)]$$

Supposing m tool calls are made, queries and results for each tool call can be represented in matrix form R, where $r_{i,j}$ is the result from the j^{th} tool call, built for the i^{th} tool, T_i:

$$[r_{i,j}]_{\substack{i\in\{1,2,\ldots,n\}\\ j=[1,2,\ldots,m]}}$$

Where the m rows of R represent the m tool calls made, and the n columns of R are aligned with the n different tools that can be called.

$$R = \begin{bmatrix} r_{1,1} & r_{1,2} & \cdots & r_{1,n} \\ r_{2,1} & r_{2,2} & \cdots & r_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ r_{m,1} & r_{m,2} & \cdots & r_{m,n} \end{bmatrix}$$

However, since any given tool call is made to exactly one tool, each row of R can only contain one entry. Since tools can be called in any order, a more accurate representation of R might look like:

$$R = \begin{bmatrix} 0 & 0 & r_{1,j_1} & 0 & \cdots & 0 \\ r_{2,j_2} & 0 & 0 & 0 & \cdots & 0 \\ 0 & r_{3,j_3} & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & r_{m,j_m} \end{bmatrix}$$

where the column index $(j_l)_{l=1}^{m}$ is a sequence of values from $\{1, 2, \ldots, m\}$ in no particular order, signifying that tools can be called in any order. As another example, that tools can be called an arbitrary number of times (including not at all), let

$$R = \begin{bmatrix} 0 & r_{1,2} & 0 & 0 & 0 \\ r_{2,1} & 0 & 0 & 0 & 0 \\ 0 & 0 & r_{3,3} & 0 & 0 \\ 0 & 0 & 0 & 0 & r_{4,5} \\ 0 & r_{5,2} & 0 & 0 & 0 \\ 0 & r_{6,2} & 0 & 0 & 0 \end{bmatrix}$$

In this example, the sequence $(j_l)_{l=1}^{6} = (2, 1, 3, 5, 2, 2)$ indicates the order in which tools 1 through 5 were called throughout the interaction ($m = 6$ and $n = 5$). In this example, T_2 has been called three times, whereas T_4 was never called at all. Tools 1, 3, and 5 were each called once. Of course:

$$R = \begin{bmatrix} 0 & T_2(q_1) & 0 & 0 & 0 \\ T_1(q_2) & 0 & 0 & 0 & 0 \\ 0 & 0 & T_3(q_3) & 0 & 0 \\ 0 & 0 & 0 & 0 & T_5(q_4) \\ 0 & T_2(q_5) & 0 & 0 & 0 \\ 0 & T_2(q_6) & 0 & 0 & 0 \end{bmatrix}$$

So that where:

$$Q = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \\ q_5 \\ q_6 \end{bmatrix}, \quad T = [T_1 \; T_2 \; T_3 \; T_4 \; T_5]$$

we have:

$$R = Q \otimes_\circ T \cdot \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix} \overset{\text{def}}{=} S$$

where "·" denotes the Hadamard (element-wise) product, $\otimes_\circ$ denotes the Kronecker product over function composition, and we define the binary matrix S, the **selection mask for R**, containing 1s in the i, j^{th} position to signify that T_i was called with q_j and 0s elsewhere. S thus encodes which tools were called and in which order by the agent throughout its engagement by the user.

The Algebra

Consider the matrix S from above. There is no reason why an agent couldn't make all 6 of those tool calls within the first cycle of a user-agent interaction. Suppose it did, and that during cycle 2 the agent makes the following 3 tool calls:

$$S_2 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

This gives rise to k selection masks S_i, one for each cycle within a user-agent interaction. Whereas each S_i will always have 5 columns (since there are 5 tools to call from in this example), the row count of S_i will vary according to the number of tool calls made inside a given cycle. Let $m = [m_l]_{l=1}^k$ be the sequence of such row counts.

Let's consider another example over a few more cycles ($k = 5$), but with a smaller toolset $\{T_1, T_2, T_3\}$ ($n = 3$):

$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}, \quad \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \end{bmatrix}, \quad \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}, \quad \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}, \quad \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

Setting and defining μ as the maximum number of tool calls made across each of the 5 cycles, $\mu = 4$, i.e.,

$$\begin{aligned}\mu &= \max\{m_i \mid i \in \{1, \ldots, 5\}\} \\ &= \max\{2, 4, 3, 0, 2\} \\ &= 4\end{aligned}$$

we can construct a tensor of shape $k \times \mu \times n$ to encode which tools were called, the order in which they were called, and within which cycle each tool call was made throughout the entire user-agent interaction. Using the example from above,

$$\Sigma = \left[\begin{bmatrix}0&1&0\\1&0&0\\0&0&0\\0&0&0\end{bmatrix}, \begin{bmatrix}1&0&0\\0&0&1\\0&1&0\\0&1&0\end{bmatrix}, \begin{bmatrix}0&0&1\\1&0&0\\0&0&0\\0&1&0\end{bmatrix}, \begin{bmatrix}0&0&0\\0&0&0\\0&0&0\\0&0&0\end{bmatrix}, \begin{bmatrix}1&0&0\\0&0&0\\0&0&0\\0&1&0\end{bmatrix}\right]$$

This encoding allows us to represent the entire structure of the user-agent interaction over multiple cycles. The process continues similarly for generating J, σ, θ, and ρ, as demonstrated previously.

Proposition

Let A be an agent with a system prompt s_A and a toolset $T = \{T_1, T_2, \ldots, T_n\}$. Suppose a user interacts with A over k cycles, resulting in:

- **Chat History:** h, where u_i is the user input and v_i is the agent's response at cycle i.
$$h = [(u_1, v_1), (u_2, v_2), \ldots, (u_k, v_k)]$$
- **Tool Indices Matrix:** σ, where $\sigma_{i,j}$ indicates which tool $T_{\sigma_{i,j}}$ was called with query $\theta_{i,j}$.
$$\sigma \in \{1, 2, \ldots, n\}^{k \times \mu}$$
- **Queries Array:** $\hat{q}$, where $\hat{q}_p \implies \theta_{i,j}$, the j^{th} query made by A in cycle i.
$$\hat{q} \in \{q\}^m, \quad \theta \in \{q\}^{k \times \mu}$$
- **Results Array:** $\hat{r}$, where $\hat{r} \implies \rho_{i,j} = T_{\sigma_{i,j}}(\theta_{i,j})$.
$$\hat{r} \in \{r\}^m, \quad \rho \in \{r\}^{k \times \mu}$$

Then, the entire interaction between the user and the agent, including all tool calls and results, is uniquely determined by the artifacts h, $\hat{q}$, σ, and $\hat{r}$. Moreover, these artifacts are necessary and sufficient to fully reconstruct the interaction.