Title: Toward Autonomous Learning in AI: Foundations in Visual Recognition, Memory Formation, and Resource Management in OAI

Abstract

This paper presents the foundational architecture and methodologies behind OAI, an autonomous learning model that mimics human-like interaction and learning within a desktop environment. OAI is designed to autonomously recognize, categorize, and understand visual elements through recursive memory processing, semantic association, and region-based visual focus. This work details the core structures and multi-threaded processes that support OAI's learning algorithm, emphasizing its self-organizing memory formation, efficient resource management through CPU-constrained Region of Interest (ROI), and semantic association with visual data. By enabling autonomous parameter adjustment and multi-threaded feedback loops, OAI aims to scale its recognition capabilities in an efficient and adaptable manner.

1. Introduction

AI learning models typically rely on supervised data to classify and recognize patterns. OAI, however, is designed to operate with minimal initial knowledge, incrementally developing understanding through direct interaction with visual data from a desktop environment. By capturing and processing screenshots as compressed JSON representations, OAI can observe pixel-based changes and autonomously identify editable versus static elements, forming primitive concepts of desktop interaction.

2. System Architecture and Multi-threaded Processing

The OAI architecture consists of a multi-threaded recursive feedback system that autonomously develops semantic connections and operational preferences. OAI operates on a structured, layered feedback loop using multiple parallel threads. Each thread manages a specific task, such as parsing input data, generating semantics, and scanning tables for updates. This setup ensures efficient parallel processing of tasks critical to OAI's self-learning algorithm, with core processes including:

- **Vision Processing Thread**: A thread dedicated to capturing screenshots every 30 seconds, compressing these images, and saving their file paths and metadata to an SQL database. This thread supports OAI's "visual memory" by feeding it time-stamped image data in compressed JSON format.
- **Core Memory Formation Thread**: Another thread continuously queries the `user_input` and `vision_input` tables in the SQL database, matching records based on timestamps. This thread forms "core memories," creating associations between visual inputs and user interactions (such as mouse movements or keystrokes) for subsequent pattern recognition.
- **Continuous Semantic Processing**: A third thread continuously scans new words and visual data, updating OAI's `word_matrix` and `semantics` tables. It autonomously generates semantic equations and records low-level associations, which serve as the building blocks for higher-level recognition patterns.

Each thread operates as a daemon, ensuring minimal memory usage while remaining active as long as the main process is alive.

3. Memory Structures and Core Memory Formation

Core memories are the primary units of learning in OAI's architecture. The system establishes core memories by aligning inputs from two SQL tables, `user_input` and `vision_input`, based on their timestamps. This alignment is achieved through recursive comparison algorithms that select the closest visual representation for each user action, establishing associative links between action-based and visual data.

Each core memory contains foreign keys that link back to the original `user_input` and `vision_input` records. This modular structure allows OAI to build a consolidated "experience" from raw input data, enabling later pattern recognition and categorization. The process functions as follows:

1. **Data Capture**: Screenshots are periodically captured, compressed, and saved as JSON objects within a shared directory. Metadata, including file path, timestamp, and file size, are logged in the `vision_input` table.
2. **Memory Formation Algorithm**: A recursive algorithm matches entries between `user_input` and `vision_input` tables, comparing timestamps within a predefined threshold (e.g., 1-second proximity). Once matched, a new entry is created in the `core_memories` table, linking the user action with its corresponding visual input.
3. **Core Memory Storage**: The core memory is stored in the SQL database with foreign keys referencing the original input records, ensuring traceability and modularity in memory formation.

These core memories provide OAI with a "history" of user interactions and environmental changes, creating an early model for understanding cause and effect on the desktop.

4. Semantic Association and Word Matrix Integration

To support the growth of OAI's vocabulary, each word in the `word_matrix` table is associated with relevant images or areas on the desktop. For example, the term "Desktop" in the `word_matrix` is linked to screenshots tagged as "desktop" in the `vision_input` table, establishing semantic connections between visual data and language.

OAI uses a recursive semantic process to generate associations for each word, which are stored in the `semantics` table. This table includes fields for semantic equations, such as `chatgpt_response`, and low-level word associations, enabling OAI to build context and meaning from repeated visual and linguistic inputs.

5. Region of Interest (ROI) and CPU Processing Management

To optimize CPU utilization, the concept of a Region of Interest (ROI) is introduced, initially defined by a static parameter to limit processing scope. The ROI represents a constrained area within the image where processing resources are focused. This allows OAI to narrow its attention to high-priority regions rather than analyzing entire images.

In the future, OAI will autonomously adjust the ROI based on task requirements. By analyzing past core memories, OAI could “learn” to identify regions where change frequently occurs and adapt its focus accordingly. Allowing OAI to control ROI size and set up multiple ROIs will introduce parallel processing capabilities similar to human multitasking, enabling simultaneous analysis of multiple areas.

6. Data Compression and Storage Management

The vision processing thread compresses screenshots using a custom JSON formula, which captures and condenses RGB values to reduce file size. These JSON objects are stored in the `vision_input` directory with only metadata recorded in the SQL database (file path, timestamp, and file size) to manage storage costs effectively.

Compression follows a two-step process:

1. **Condensed RGB Representation**: Screenshots are transformed into condensed RGB matrices, capturing only unique color patterns and their run-lengths to reduce data size.
2. **Unique RGB Sorting**: The matrices are sorted and further condensed by removing duplicates, creating a minimalist representation of each screenshot.

This approach allows OAI to “recall” previous images and compare differences across core memories, identifying static versus dynamic areas on the desktop and recognizing patterns based on visual changes.

7. Future Work

Future developments will include extending OAI’s control over ROI, enabling it to adjust parameters autonomously based on processing needs. Additionally, OAI will be equipped with mechanisms to prioritize frequently changing areas, potentially scaling up to recognize editable elements and adaptively manage CPU resources. Semantic association will continue to grow, allowing OAI to refine its understanding of desktop elements by analyzing the cumulative impact of user interactions and visual changes.

8. Conclusion

OAI’s design represents a shift from traditional supervised AI to an autonomous learning model that interacts with visual data as a human would. Through multi-threaded processing, recursive memory formation, and ROI-based resource management, OAI incrementally develops an understanding of desktop environments, linking visual changes to semantic knowledge. This foundational framework opens avenues for self-adaptive and highly efficient AI capable of autonomous learning in complex digital environments.