

ローカルLLMを用いたADASの多目的最適化への対応

ルネサンス大阪高等学校 1年 大江 修聖

1. 背景

Automated Design of Agentic Systems(ADAS)という、AIエージェントの自動設計を行う技術がShengran Hu et al.(2024)によって発表され、ADASは読解や数学のタスクにおいて、従来の手動設計されたAIエージェントのパフォーマンスを上回ったが、OpenAI APIのコストに関する問題や、多目的なADASの開発という展望が残されていた。^{*1}

そして、ローカルLLMを用いてコストの削減を行うということは、Bilgeworthによって既に行われていた^{*2}ので、本研究では、実行速度と性能の面で、多目的なADASの開発、すなわちADASの多目的最適化への対応を試みた。

2. 手法

ADASの多目的最適化を行うために用意したソフトウェアは以下の通りである。

- **Python3.13(64-bit)**
- **ADAS-ollama^{*2}**
(GitHubからクローン)
- **ADAS-ollamaのRequirements.txt上のAPI**
- **Llama-3.2-1B-Instruct-GGUF^{*3}**
(QuantTypeが異なるものを複数使用)
- **Anaconda**
(ADAS-ollamaのセットアップに使用)

3. 研究

実行速度と性能の多目的最適化を行いパレートフロンティアを得るため、まずbartowskiによる、複数の程度に量子化されたLlama3.2^{*3}をOllamaで使用するためにインストールした。

参考文献

^{*1} <https://arxiv.org/abs/2408.08435> Shengran Hu et al.(2024)

^{*2} <https://github.com/Bilgeworth/ADAS-ollama> Bilgeworth

^{*3} <https://huggingface.co/bartowski/Llama-3.2-1B-Instruct-GGUF>
bartowski

Anaconda PowerShell PromptでADAS-ollamaリポジトリの_arcフォルダ内のsearch.pyを実行するとエラーが発生した。しかし、同フォルダ内のarc_prompt.pyのOutside_the_boxの定義の最初の行のインデントを削除することで解決した。そして、動作テストのためにsearch.pyのモデルを指定するparser.add_argumentにおいて、モデルをllama3.1からllama3.2(通常のLlama3.2)に変更した。以上の変更を加え、Windows11の環境でsearch.pyを実行したところ、実行中に複数のエラーが発生した。(例外処理は行われている) このエラーは2024/11/17時点で未解決であり、原因を究明している。

4. 結果・展望

本研究では、ADASの多目的最適化のため、必要なソフトウェアを明らかにし、ADAS-ollamaのバグを一部修正した。2024/11/17時点、ADASの多目的最適化は未完である。それを達成し、ADASのパラメータ間のトレードオフが明らかになれば、ADASはより強力になるだろう。