Part 3: Automated Discovery of Efficient Software Design

Implementation

I got rid of the JSON structure as I thought this assignment was meant to optimize to the max, and I found the JSON structure was hindering GPT from giving me correct answers. As the objective here was just to multiply matrices, I thought it safe to just ask GPT to provide Rust code. My program first reads system information, then asks GPT to provide an initial rust code. The code is executed and latency is measured, and then a loop runs where GPT is continuously asked to generate a hypothesis to better the latency and then generate code based on that hypothesis. An error correction method is also in the loop to ensure any incorrect code is fixed.

Difficulties

GPT kept outputting incorrect code over and over again, so it was difficult to get better latency quickly. However, once I played with my prompts a little bit I was able to get it to work a little better. The program as a whole works well when run for longer periods of time, but it is inconsistent as sometimes GPT gets trapped in an error ridden code loop and is just trying to fix those for a long time. Sometimes it executes the task quickly the first time and sometimes it doesn't, leading to different variations of latency at the end – the model is very inconsistent so it is hard to get accurate results.

AI

AI wrote almost all of this code!