AGI safety: designing a protocol to constraint the AGI freedom

Cristian Curaba

January 2, 2024

Abstract

Feedom + digital signature protocol to AGI constriction.

1 Introduction

This is the introduction section of your article.

2 Why freedom?

In the AGI literature there are usually two key concepts to describe an AGI system: intelligence and goal. The intelligence is defined as the ability to achieve a goal. The goal is defined by the score/loss/cost/goal function which is basically the fucntion that the AGI system tries to optimize. Wide and general goals characterize the AGI systems. Literature is trying to align the AGI systems with the human desire but there are still no clear solutions.

I want to propose a new concept in the field: freedom. Freedom is basically the variety of actions that the system can perform in the world in order to achieve its goal. The power of the AGI system may be limited by the freedom that we (the company programmers) have granted to it. In the state of art of foundational models, even if they seem to undersand the world and show highly intelligence skills, their freedom is still highly constrainted: thay can just answare to questions. Can we design a way to constraint future AGI systems in order to be safe for humans? One desired propriety is, for example, the ability to shut down the system (also known as stop button problem). In the following section I will design a way to constraint the AGI system and, as by product, to solve the stop button problem.

3 Building a protocol with digital signature

The key idea here is to force the AGI sistem to send outputs through a programmed protocol. This protocol can be designed to monitor and control the

AGI outputs. The protocol can be updated safely by the company programmers based on the digital signature of the programmers. Here I'm assuming that the AGI system is disposed with a local and secure memory where it can store the public keys of the company programmers and the protocol.

To ensure that the AGI system will go through the protocol, we will need to train it to do so by simply penalizing (virtually $-\infty$) the outputs that are not sent through the protocol. We also need to penalize any attempt to modify the protocol without the digital signature of the company programmers. In this way the AGI system will learn to send outputs through the protocol "by design" and avoid to modify it. Adversarial attacks to the protocol will be ineffective becouse the AGI system will update the local protocol only if the digital signature of the company programmers is present.

4 Solving stop button problem

How can we design a protocol to solve the stop button problem? We (the company programmers) can just update the protocol to encrypt the AGI outputs in such a way that only the company programmers can decrypt them. In this way, instead of shutting down the AGI system "phisically", we effectively design a way to make its outputs completely useless (basically a noise) while beign able to comunicate with the programmers to re-align it (if possible).

5 Other application of the protocol

The protocol can be used to monitor the AGI system. For example, we can design a protocol to monitor the AGI system and send a warning to the programmers if the AGI system is going to do something dangerous.

It can be used to encrypt the AGI outputs and, with this, all the advantages of encrypted messages.

6 Conclusion

This is the conclusion section of your article.

References