AGI logic tutorial

YKY

May 16, 2021

Summary

describes a kind of univeral logic that can completely solve AGI

Contents

0 Background

We want to **training** an intelligent system. Training is a process of **machine learning** and also a problem of **optimization**. The goal is to maximize **long-term reward sum**:

$$\text{maximize: } \int_0^\infty R\,dt \tag{0.0.1}$$

where $R(t)$ = reward at time t. $\int_0^\infty$ represents the **time horizon** for calculating the cumulative reward. (I used the differential form, the actual application is usually the discrete form , But the two are basically the same, so you don't have to go into it)

As the saying goes, "Chess shit is greedy for pawns." It may lead to checkmates after N moves. This is **stupid** behavior. Therefore, the (0.0.1) formula requires the system to take into account long-term interests, and forces it to learn **wisdom**.

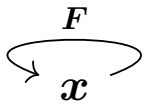
Architecturally, the AI is a **dynamical system** that constantly updates its "state" $\boldsymbol{x}$ via: *

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}) \tag{0.0.2}$$

Or expressed in discrete form:

$$\boldsymbol{x}_{t+1} = \boldsymbol{F}(\boldsymbol{x}_t) \tag{0.0.3}$$

$\boldsymbol{F}$ is called transition function. Or more vividly:



(0.0.4)

Our goal is to **learn** the function $\boldsymbol{F}$, implemented as a **deep neural network**. $\boldsymbol{F}$ contains all the **knowledge** in the intelligent system.

* Part of the state $\boldsymbol{x}$ contains **sensory input** and **action output** that allow the AI to interact with the external environment.