Formal AGI

Tom Rochette <tom.rochette@coreteks.org>

December 12, 2016 — a6ee0e5d

0.1 Context

0.2 Learned in this study

0.3 Things to explore

1 Overview

[1] introduces the idea of an environment $\mathcal{E}$ where our agent lives. At each time step t, the agent observes state s_t and select an action a_t. Therefore, a sequence $S_{1\to t} = s_1, a_1, ..., a_{t-1}, s_t$ represents the states and the actions that were taken at every time step by the agent.

2 Optimizing the sequence

One of our goals is to reduce the length of the sequence required to get from any state s_x to a desired state s_y.

Given a sequence $S_{x\to y} = s_x, a_x, ..., a_{y-1}, s_y$, we want to know if it is possible to reduce the length of this sequence. In graph theory, this would be comparable to searching for the shortest path between s_x and s_y.

This question has a couple of interesting challenges:

- If we have s_x and s_y but no path between these two states, how can we build such path? This would be comparable to you being in state s_x where you don't know graph theory and wanting to be in state s_y where you know graph theory.
  - In this case, I think that there are two ways to learn, either through self-learning/exploration or through observation/repetition.
  - Let's start by looking at the second case. You first need to find a source you can use as a good model to base yourself on. Once you have acquired that source, you need to extract the critical components that will need to be reproduced. Then you need to reproduce those components as closely as possible to the actual reference model.
  - Once you have been able to establish a path between s_x and s_y, you can now attempt to improve/optimize it even further through random exploration. Basically, you'll be trying to "feel aroud" to see if anything can be done better. In this case, it means that you'll need to have an evaluation function that will let you know if what you are doing is better than what you did before.
  - When you have acquired the first part of a behavior, that is, how to get from s_x to s_y, it is possible to trim the parts of the graph are not the shortest path between these two points.
  - [2] introduces a system where both SL (supervised learning) and RL (reinforcement learning) are used to train an agent. First, the SL is used in order to train the agent to have the same decisions

[1] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, Martin Riedmiller: "Playing Atari with Deep Reinforcement Learning", 2013; arXiv:1312.5602.

[2] David Silver, Aja Huang, Chris J. Maddison, Arthur Guez, Laurent Sifre, George van den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, Sander Dieleman, Dominik Grewe, John Nham, Nal Kalchbrenner, Ilya Sutskever, Timothy Lillicrap, Madeleine Leach, Koray Kavukcuoglu, Thore Graepel, Demis Hassabis: "Mastering the game of Go with deep neural networks and tree search", 2016; doi:10.1038/nature16961

as the training set. Once the agent reproduces the actions of the training set more frequently than by chance (50%), RL takes over to improve the quality of the results produced by the agent.
- How can we automatically develop an intuition for an appropriate evaluation function?

3 Evaluation functions

Previously we touched on the fact that during exploration, you will need to be able to evaluate if you're moving in the right direction or not (figuratively speaking).

Here, we look at a couple of generic evaluation functions which may have guided non-organic components to turn into organic components and then evolve into living organism and finally into the humans being that we are nowadays.

It is very likely that the evaluation function, like us, has evolved over time, but it means that it was also built on previous generations of evaluation functions, keeping some traits and evolving others.

- Random
  - Anything can be rewarded/punished, thus no possible convergence
- Reward on repetition
  - Should stay stuck on a local minima based on the length of the repetition string
- Reward for repetition of previous behavior of previous agents
  - The reward would always be increasing, rewarding reptition of behavior
  - Not necessarily rewarding "good"/"goal" behavior, just rewarding repetition
- Reward conservation of energy/accumulation of energy (energy maximization)
  - Will do nothing except if forced to reduce/stop energy consumption
- Reward spending of energy (energy minimization)
  - How should energy expenditure be directed?
  - Reward steps when agent does not have any energy -> Concept of energy/consumable/budget
  - In order to prevent the agent from doing nothing in order to save its energy, not spending energy would still imply some energy being spent anyway (force the agent to act and not stay idle)
- Transition to society/social-based competition
  - Reward being better than other agents using a commonly agreed metric
- Multi-level system of rewards, similar to Maslow pyramid (basic needs being more critical than high-level/intellectual needs)

4 See also

5 References