CSSE 413 - Project Review

An AI That Can Play Hide-and-Seek

Dr. Wollowski

*Yiyu Ma, Bill Yu*

10/21/2019

1. **Introduction**

OpenAI, the world-famous AI team, recently released an AI project of training its model to play hide-and-seek games. With an easy setup and simple rules, the hide-and-seek agents perform surprisingly complex and intelligent behaviors using current machine learning techniques [1]. For example, during the process, hiders learn to use the boxes to block the doors, meanwhile, seekers learn how to use the ramp to overpass walls. One of the most intelligent yet scary behavior is developed by hiders when they learn to prevent the ramp from seekers using it.

attempts at creating intelligent AI that can solve complex human-relevant tasks have been made by numerous AI researchers. Similarly, the OpenAI team is making progress in the field of physically grounded task by training its hide-and-seek agents. The study is a proof of concept showing that multiagent reinforcement learning can lead to increasingly human-relevant behaviors, Naturally, a much more complex environment and more rules can be introduced in the future, therefore more complex behavior of humans can be explained and observed.

1. **Hide-and-Seek**

**Basic Rules**

The hide-and-seek agents are grouped into team-based hiders and seekers while ramps and blocks are provided for agents to grab and lock in place. Agents are given with basic human capacity; players can move by setting a force on themselves in the x and y directions as well as rotate along the z-axis; can see objects in their line of sight and within a frontal cone; can sense distance to objects, walls, and other agents around them using a lidar-like sensor; can grab and move objects in front of them; and can lock objects in place. Each agent acts independently and knows nothing about invisible objects, where "invisible" means out of the line of sight.

the team that locked an object can unlock it [2]. A random room with doors is provided within the environment for hiding purposes. Before the start of the game, hiders are given ample time to "escape" and to change the environment while seekers are immobilized.

**Training and Evaluation**

The agents are trained with self-play and standard reinforcement learning [2]. At optimization time, Proximal Policy Optimization—the default reinforcement learning algorithm of OpenAI—is used. It has a centralized value function to calculate and evaluate what actions each agent should take. Agents share the same parameters in the function but different information about the state and environment based on their observations and memories [2]. Generalized Advantage Estimation, a policy gradient method in reinforcement learning to optimize the cumulative rewards, is also utilized [3]. Policy gradient methods work by computing an estimator of the policy gradient and plugging it into a stochastic gradient ascent algorithm. the goal of the Proximal Policy Optimization function is to find an optimal behavior strategy for the agent to obtain optimal rewards. The policy gradient methods target modeling and optimizing the policy directly. The policy is usually modeled with a parameterized function of respect to θ. The value of the reward (objective) function depends on this policy and then various algorithms can be applied to optimize θ for the best reward [3].

Researchers compare the multi-agent competition method with another previous algorithm called intrinsic motivation, in which agents are given incentives to interact with objects. For instance, at the very beginning, agents are offered with some 2-D positions of boxes in the environment and encouraged to interact with boxes, but they seem to move these boxes to new places in a confusing way [1]. By contrast, the group trained in a full environment playing hide-and-seek shows more human interpretable behaviors such as building something similar to a defensive shelter [1]. Besides, as the number of sophisticated tools and the complexity of the environment gets more, agents trained by intrinsic motivation perform more and more meaningless behaviors [2].

**Autocurricula and Results**

No strategy taken by the agents is taught or directly rewarded. Multiple interesting behaviors emerged as a result of the so-called, self-supervised *autocurricula* with no direct incentives for agents to interact with objects [1]. Agents are given a team-based reward; hiders are given a reward of +1 if all hiders are hidden and -1 if any hider is seen by a seeker. Seekers are given the opposite reward, -1 if all hiders are hidden and +1 otherwise.

Agents produce a total of six distinct stages of emergence strategy, many of which need the use of sophisticated tools. In a simple environment, six acute phases of the agent strategy appear, each of which can add new pressure for the rival team to adjust immediately: random moving, chasing, door blocking or shelter construction, ramp use, ramp defense, and coordination [1]. In a more complicated environment, two more phases emerge box surfing and surfing defense [1]. The first lesson takes a few million games, and after repeated practicing, agents finally understand the basic rules of the hide-and-seek game: hide and chase. In the second lesson, hiders win many times because they learn to lock themselves in a shelter. Then seekers find out how to climb ramps to finder hiders. Soon hiders come up with new counter-strategies that they lock the ramps in place so that seekers cannot move them. Much to the OpenAI team's surprise, hiders figure out they can move unlocked boxes to the locked ramps, climb onto the boxes, surf themselves to the shelter, and they can see the hiders in their line of sight. Eventually, hiders decide to lock everything in the game environment so that seekers can use nothing to find them if hiders create defensive shelters.

Among more than 480 million repeated games, agents learn emergent strategies and counterstrategies by themselves [1]. In a full environment, the whole process takes more than 30 hours. As batch size increases, the number of training episodes may sometimes increase, but the clock time needed decreases by a lot, which means larger batch size raises efficiency [1].

1. **Contributions**

Reinforcement learning is a subarea of machine learning techniques that these AI agents use to get rewards and evaluate the best behavior to take under specific conditions. AI agents behave surprisingly better than researchers expect.

The main contributions of this work are: 1) clear evidence that multi-agent autocurricula lead to many distinct and compounding phase shifts in agent strategy, 2) evidence that when induced in a physically grounded environment, multi-agent autocurricula can lead to human-relevant skills such as tool use, 3) a proposed framework for evaluating agents in open-ended environments as well as a suite of targeted intelligence tests for our domain, and 4) open-sourced environments and code for environment construction to encourage further research in physically grounded multi-agent Autocurricula [2].

1. **Reference**

[1] Baker, B., Kanitscheider, I., Markov, T., Wu, Y. and Powell, G. (2019). *Emergent Tool Use from Multi-Agent Interaction*. [online] OpenAI. Available at: https://openai.com/blog/emergent-tool-use/ [Accessed 20 Oct. 2019].

[2] Baker, B., Kanitscheider, I., Markov, T., Wu, Y., Powell, G., MeGrew, B. and Mordatch, I. (2019). Emergent Tool Use From Multi-Agent Autocurricula.

[3] Schulman, J., Moritz, P., Levine, S. and Jordan, M. (2016). High-dimentional Continuous Control Using Generalized Advantage Estimation. ICLR.