Multi-Agent Reinforcement Learning: The Gist

As if one robot learning everything wasn't enough already

Austin Nguyen Follow

Apr 15, 2020 · 7 min read ★

Photo by Daniel Cheung on Unsplash

Reinforcement learning (RL) studies the classic "monkey see monkey do" shenanigan or the "robot that learns as it goes." Like how infants learn and grow with positive reinforcement from adults, RL teaches agents to learn by maximizing cumulative rewards in their environment. Ultimately, they then acquire a desired skill or behavior. In the past decade, we've witnessed RL agents beat top-tier E-Sport athletes at their own games, teach quadruped robots how to run, and even optimize chemical reactions. The field is stacking accolades for itself, but what's next?

Multi-Agent Reinforcement Learning (MARL) studies how *multiple* agents can collectively learn, collaborate, and interact with each other in an environment. It's one of those things that makes people imagine the *possibilities:* teams of robots playing soccer, building houses, or managing farms. With strength in numbers, MARL could make tasks previously infeasible for robots, possible. However, it has some big, big hurdles to cross. As a MARL researcher, this is the part that I love: designing algorithms and methods to help overcome those obstacles. Here I present a brief overview: the inspiration, formulations, and, most importantly, difficulties of Multi-Agent Reinforcement Learning.

Photo by Maksim Shutov on Unsplash

Why It Matters

Ants are peculiar creatures and have fascinated researchers for the longest time. They have this innate ability to communicate, coordinate, and collaborate effortlessly. As a result, their behavior has baffled scientists and has remained one of nature's most mysterious beauties. Often, this is what inspires multi-agent learning for robotics. Since colonies of ants have seamlessly constructed shelter, gathered food, or even built bridges with their bodies, why can't robots learn to do the same? If we could teach robots to achieve the same dexterity, fluidity, and coordination as ants, the potential for multi-robot systems would have no bounds.

The Background

Reinforcement learning represents the environment as a Markov Decision Process (MDP) with specified state space, action space, reward function, and probabilistic

transition function. The agent's goal is to learn a policy that maximizes the expected discounted reward characterized by the MDP:

$$\eta(\pi) = \mathbb{E}_{s_0, a_0, \ldots} \left[\sum_{t=0}^{\infty} \gamma^t r(s_t) \right]$$

where gamma represents the discount factor. The agent adjusts its policy to maximize the above value. In other words, the agent tries to find the best sequence of actions to get as much reward as possible, putting less priority on rewards farther in the future.

In MARL, however, we tweak this formulation.

MARL typically represents the environment as a Stochastic Game. While the name is different, it's fairly similar to an MDP. States become joint states of all the agents, with different rewards corresponding to each possible joint action. Transition functions remain analogous to the single-agent case, replacing states and actions accordingly. Assuming we have two agents, we can represent a *single state* as a two-dimensional chart.

Representation of a state in MARL

Let one agent's action space be represented by the rows and the second agent's by the columns. After both agents choose their actions, we look at the cell corresponding to the respective row and column. Each agent then receives a reward denoted by the values in the cell.

The general framework of MARL is similar to its single-agent counterpart, but what makes MARL a whole new beast? Why can't we just use the same single-agent algorithms on two agents? Groups of ten? Hundreds?

At first glance, I thought of a straightforward solution: have each robot use single-agent algorithms, learn for themselves, and voilà. We have them razing mountains together. Of course, that was wrong, and here's why.

The Hurdles

Non-Stationary Transitions

This is one of the biggest challenges in MARL. In MDPs (the framework used for single-agent RL), we assume stationary transitions. We assume that given a distinct state and action pair, the transition probabilities to other states **remain constant** throughout. For example, let's say we have a robot named Wall-E at *Location A* (his state). Then, say he chose to move to the *right* (his action), transitioning himself to *Location B* with some probability. A stationary MDP assumes that this probability value stays the same, as long as Wall-E chooses to move *right* again next time he's in *Location A*.

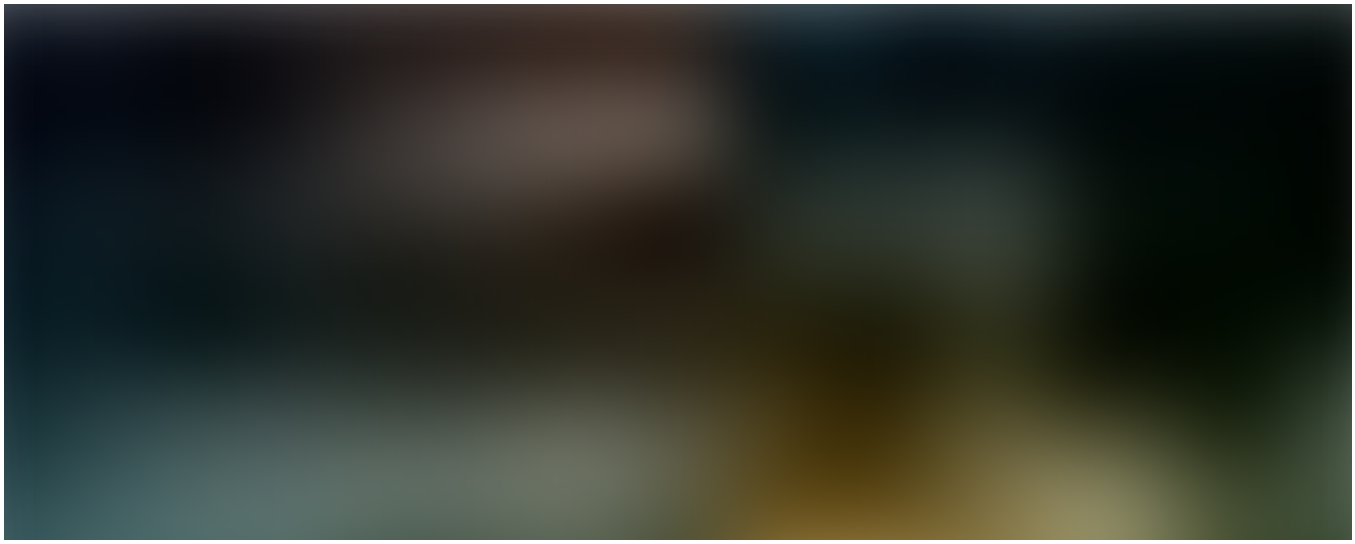

Photo by Lenin Estrada on Unsplash

In MARL, however, we have multiple agents. We denote the "state" in MARL as the joint state of all the agents. As a result, state transitions become dependent on the joint actions of all the agents, ie. dependent on how all of them act, not just one. To prove this point about non-stationary transitions, let's say Wall-E has some friends he wants to work with.

From Wall-E's point of view, we assume he doesn't know the other agent's actions. Through his eyes, if Wall-E, again at *Location A*, went to the *right* on two occasions, the transition probabilities may be different. This is caused by other agents possibly **changing their policies.** Even if Wall-E always chooses to go *right* at *Location A*, the other agents might choose different actions from before. As a result, the environment becomes non-stationary from Wall-E's point of view. Him taking the same action in a state does not always give the same results.

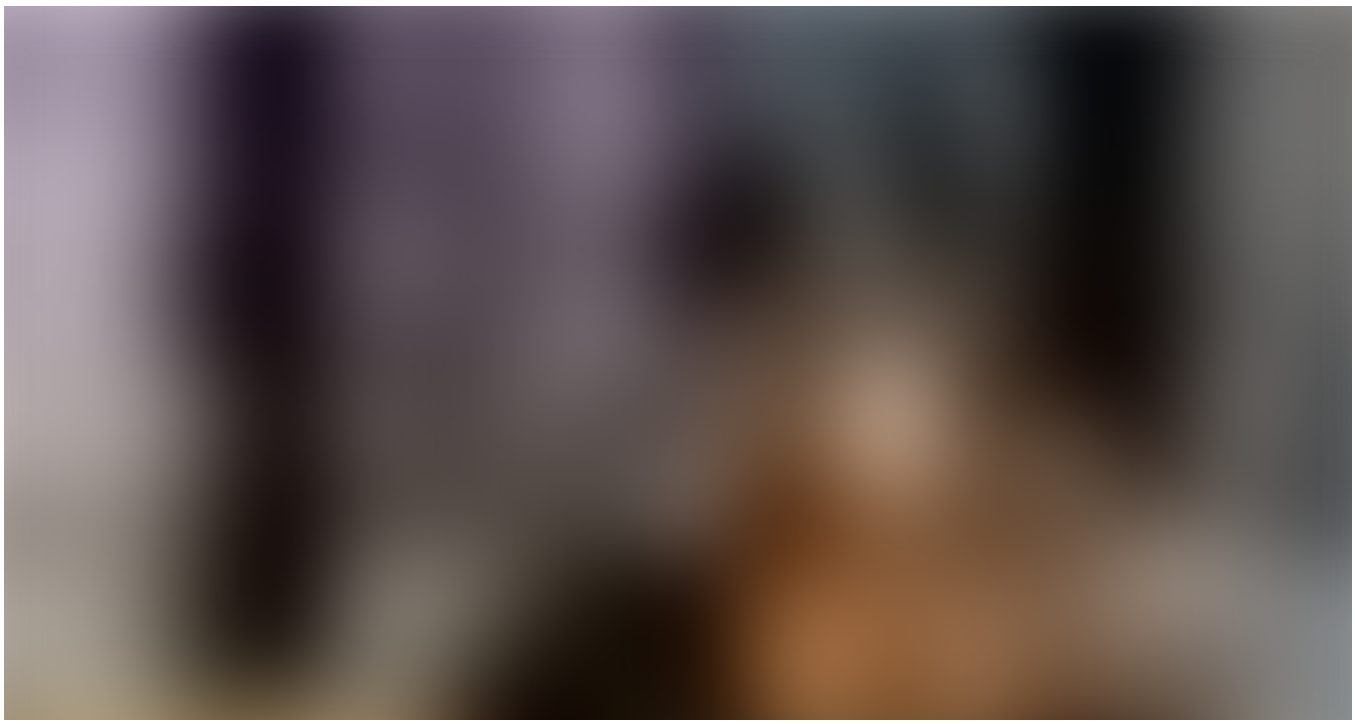

Photo by ray rui on Unsplash

So…why does it matter?

RL agents learn to distinguish which actions are preferable over others. They reason about this using an intuitive principle: good actions have good consequences and vice versa. In other words, rewards for actions are dependent on the state it transitions to. Since the transition probabilities, given a state and action, change over time, it makes training MARL agents that much more difficult. They don't have as much information about the environment relative to single-agent problems.

Exponentially Increasing State and Action Space

After encountering this issue, I gravitated towards an easy fix: just have a single policy dictate the actions of all the agents. That way, in the policy's perspective, executing that joint action in this state will always have the same result afterward.

Photo by Jehyun Sung on Unsplash

Again, that was wrong. State and action space sizes increase **exponentially** with respect to the number of agents. This can make learning intractable due to things like the curse of dimensionality. Sometimes, the problem may get too large and make convergence take too long. Intuitively, it makes sense. The more decisions our policy has to make for our agents, the longer it will take to learn how to make correct decisions.

The Tug of War

Multi-agent RL attempts to reconcile these two opposing forces: decentralization vs. centralization. As said before, the former suffers from non-stationarity with good scalability while the latter has the opposite.

Photo by Anna Samoylova on Unsplash

There are many other difficulties with multi-agent systems like the unstableness of experience replay, increased variance in policy gradients, and the question of data efficiency. However, the two competing issues discussed above remain the crux of what makes MARL hard.

Well, Shoot...what now?

Well, yes. While there are issues inherent to multi-agent settings not as present in its single-agent counterpart, plenty of algorithms have adapted to these conditions. These difficulties haven't been fully resolved (of course), but we have seen significant strides for MARL in the past two decades.

I believe the hurdles in MARL are just like any other scientific barrier in the past: overcome-able. Do we need newer, more clever algorithms? Do we need crazier robot designs? Do we need to completely shift our formulation of the problem itself? I don't know, but the mystery is what makes this so captivating. The field's potential is what pushes us to help find an answer. Even though the idea of "perfectly collaborative robot-ants" seems pretty far-fetched as of now, researchers, scientists, and fanatics, including myself, are inching our way to making it just a bit less science fiction.

> *From the classic to state-of-the-art, here are related articles discussing both multi-agent and single-agent reinforcement learning:*

Introduction to Nash Equilibria: Friend or Foe Q-Learning

Making robots tip the scales

towardsdatascience.com

Unpacking Stigmergic Independent Reinforcement Learning and Why It Matters

Because 1+1=3

towardsdatascience.com

Reinforcement Learning Machine Learning Artificial Intelligence Beginners Guide

Multi Agent Systems