Machine Intelligence Assignment 4

(Papers Summary)

Team 1

Mohamed Shawky Zaky, Sec : **2**, BN : **15**
Remonda Talaat Eskarous, Sec : **1**, BN : **19**
Mohamed Ahmed Mohamed Ahmed, Sec : **2**, BN : **10**
Mohamed Ramzy Helmy, Sec : **2**, BN : **13**

Discovering Reinforcement Learning Algorithms

Advances in Neural Information Processing Systems 33 (NeurIPS 2020), Virtual
17 July 2020
DeepMind
Junhyuk Oh *Matteo Hessel* *Wojciech M. Czarnecki*
Zhongwen Xu *Hado P. van Hasselt* *Satinder Singh, David Silver*

Problem

RL algorithms update the agent's parameters using discovered update rules. Automating the process of update rule discovery from given data can lead to much more efficient algorithms instead of doing so manually. Such automation is known as Learning to Learn. This concept shows that it's possible not only to learn to optimize static well-defined goals but to learn the meta-level information about the goal as well. Learning meta-information is a simple task if the environment domain is static, and the agent will interact with a very close environment to the one it learned to deal with. On the other hand, it's not that simple to generalize on other environments from different domains using the same automated update rule. The problem is to discover general algorithms that are effective for a broader class of agents and environments instead of being adaptive to a particular environment. There have been a few attempts to generalize and meta-learn RL algorithms, such as MetaGenRL. It was proposed to learn a domain-invariant policy update rule, that could generalize from a few MuJoCo environments to other MuJoCo environments. However, no prior work has attempted to discover the full update rule; instead, they all relied on value functions, which is the most fundamental building block of RL bootstrapping. This paper proposes a new methodology called Learned Policy Gradient (LPG) that can generalize to complex Atari games. LPG meta-learns its own mechanism for bootstrapping. This shows the potential to discover general RL algorithms from data in an automated way that can be applied in different domains.

Methodology

The goal is to find the optimal update rule for policy and value functions, parameterized by η, from a distribution of environments $P(\varepsilon)$ and initial agent parameters $P(\theta_0)$: $\eta^* = argmax_\eta(E_{\varepsilon\sim p(\varepsilon)}E_{\theta_0\sim p(\theta_0)}[G])$, where G is the expected return at the end of the lifetime. LPG – the update rule parameterized by η – is proposed to be a backward LSTM network that produces the updates of the policy and the prediction vector – prediction of the state – using the prediction vector of current and previous steps $y_\theta(s_t)$, $y_\theta(s_{t-1})$, and the previous policy $\pi(a_{t-1}, s_{t-1})$. LPG is domain-invariant as it takes neither action space nor observation space as an input, and it's environment-invariant as well, as it meta-trains its parameter η based on the average of the policy gradients generated by different agents in different environments while training the agents using the current belief in an end-to-end manner.

Results

LPG was evaluated on different environments, such as delayed reward, noisy reward, and long-term credit assignment environments, where LPG outperformed A2C on most of them, which shows that LPG is an even better solution than manually designed RL algorithms. Regarding the prediction part, it is implicitly shown that the LPG is asking the agent to predict future rewards and use such information for bootstrapping. This why prediction values converge on a better estimate than in other algorithms.

Comparison to curriculum

Our Curriculum proposes the methodologies in RL in the case that both objective and update rule are static and fully defined. On the other hand, this paper tries to meta-train their parameters to get more efficient and adaptive RL algorithms that generalize to different domains and different environments.

Critique

This paper provides a robust approach to get adaptive RL algorithms. However, it assumes that there are plenty of resources to train multiple agents at the same time and share their gradient information with each other to meta-train the update rule. Another problem is that they assume the samples of environments they take to represent the whole possibilities of environments, which is not the general case for sure.

Robust Multi-Agent Reinforcement Learning with Model Uncertainty

34th Conference on Neural Information Processing Systems (NeurIPS 2020), Vancouver, Canada.

15 December 2020

Amazon Web Services

Kaiqing Zhang Tao Sun Yunzhe Tao Sahika Genc Sunil Mallya Tamer Bas¸ar

Problem

The paper solves the problem of **Multi-Agent Reinforcement Learning** where each model aims to optimize its long-term return while interacting with the environment and the other agents. In such a problem the model of the environment is **not completely known** as each agent neither knows the rewards of all other agents nor the transition probability model. So to build a **robust** Multi-Agent Reinforcement Learning framework, such a framework must be able to handle **uncertainty** due to the not completely known probability model and the other agents' reward functions.

Methodology

The proposed solution to the problem consists of formulating the problem as a **Robust Markov Game problem**. Robust Markov games have the same representation as Markov Games except that it doesn't have the transition model and the reward of each agent. The Robust Markov game model consists of the agents, state-space, actions, and an added uncertainty that models the absence of transition models and other rewards. Such added uncertainty adds robustness to the models. Also, the new formulation contains the value function, action-value function, and a **novel joint policy** that all agents tend to follow which results in maximizing the expected return for each agent. To learn such a joint policy, Agents are represented as **Actor-Critic** agents in which a **Q-Learning algorithm** is used to learn the joint policy by reaching the **Nash Equilibrium** and a **Policy Gradient algorithm** is used to handle the **large** and **infinite** state-action space.

Results

The performance of the novel framework is compared to other frameworks such as MADDPG in which **robustness is not modeled** and M3DDPG in which **robustness is modeled** as weighted components of policies. The novel framework outperforms the other 2 methods in many settings such as cooperative navigation (3 cooperative agents), keep-away (single-agent and single-adversary scenario), physical deception (3 agents that have one adversarial agent), and predator-prey (3 adversarial agents). The proposed framework achieves s**ignificantly more successes** than the other frameworks with different amounts of uncertainty. In environments that contain no uncertainty, the 3 frameworks have similar performance.

Comparison to curriculum

The proposed method handles Multi-Agent Reinforcement Learning problems with joint policies, unlike methods taught in class which mostly deal with single-agent RL problems. Also, the proposed approach further improves dealing with unknown environment models and can combine **Q-Learning** and **Policy Gradient** methods to efficiently learn the new policy and tackle infinite and large state-action spaces.

Critique

The paper is written clearly as it presents the problem and previous literature in a clear way for even new readers. An intensive and fair comparison is made between the previous approaches in which the novel method outperforms the other ones. However, The results would be more powerful if such an approach was experimented on other realistic applications such as **Autonomous driving and Robotics** which can bridge the gap between Research and Practice**.**