Deep Reinforcement Learning for Atari Games

Liyang Xie Xiao Zeng Kaixiang Lin

Abstract

Deep reinforcement learning (DRL) has achieved unprecedented success in many challenging domains [5, 6], by combining the power of modeling complex functions of deep learning and the fairly general-purpose framework of reinforcement learning. In this project, we will propose a new methods in DRL for the purpose of playing games such as Atari. The experiments will be conducted on OpenAI Gym [1] and performance of our method will be compared with competing state-of-arts such as A3C algorithm.

1. Introduction

Deep reinforcement learning has been attracted widely interests from both industry and academia. The reason is that the combination of deep learning and reinforcement learning has shown the effectiveness on lots of applications and more surprisingly, the good performances can be achieved without extensive problem-specific engineering [5, 6].

In the center of most reinforcement learning algorithms is how the agent interacts with the environment. At each step, the agent takes the corresponding *action* based on the *observation* and *reward* received, which determine a internal agent *state* that is used by reinforcement learning algorithms. The mapping function from agent state to an action is designed to maximize the cumulative reward [7].

Lots of advanced algorithms were proposed to learn the function that project state to an action. One track is to formulate an action-value function $Q(s, a; \theta)$ and always choose the action gives you the maximum Q-value, which is named as value-based model-free reinforcement learning method. In contrast to above methods, policy-based model-free methods directly parameterize the policy $(\pi(a|s; \theta))$ and update the parameters θ by performing policy gradient. In this project, we will first review the existing techniques in deep reinforcement learning and proposed a method to outperform the state-of-arts on a specific problem, such as Atari 2600 on OpenAI Gym.

2. Related Work

The milestone of the deep reinforcement learning is deep Q-learning(DQN)[4], a variant of Q-learning, proposed by DeepMind. It is the first deep learning model trying to learn control policies with reinforcement learning. In 2015, DeepMind presented an improved version of DQN[5]. Their work outperformed previous algorithms and achieved a capability comparable to that of professional human-being.

While DQN performs well in fully-observable environments, it achieves poor result in partially observable environments. To address this problem, Hausknecht and Stone *et al.* [2] introduced the Deep Recurrent Q-Networks(DRQN). The idea is to build a recurrent neural network such as LSTM on top of the DQN model.

One drawback of DQN is that it needs to aggregate over time to overcome data non-stationarity. To reduce the overhead caused by experience replay, DeepMind [3] proposed a another paradigm for deep reinforcement learning: multiple agents are running in parallel asynchronously on multiple instances of the environment. Using the paradigm makes Q-learning both efficient and compatible with deep neural network at the same time. They named their best method *asynchronous advantage actor-critic* (A3C). Experiments showed that A3C not only achieved better result but also required less computational cost.

3. Time Line

10/19-10/26: Review some related literatures about deep reinforcement learning-based games and related deep reinforcement learning methods. Figure out possible suitable methodologies and games to implement.

10/27-11/02: Review paper "Asynchronous Methods for Deep Reinforcement Learning". Learn to use OpenAI Gym, Keras software package.

11/09-11/16: Use Keras to define the deep q network. Write midterm paper.

11/16-11/23: OpenAI's gym library to interact with the game Learning Environment.

11/23-11/30: Use Tensorflow to optimization the network

12/01-12/13: Test the game. Write final report.

References

[1] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba. Openai gym. *arXiv preprint arXiv:1606.01540*, 2016.

[2] M. Hausknecht and P. Stone. Deep recurrent q-learning for partially observable mdps. *arXiv preprint arXiv:1507.06527*, 2015.

[3] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. P. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu. Asynchronous methods for deep reinforcement learning. *arXiv preprint arXiv:1602.01783*, 2016.

[4] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

[5] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015.

[6] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *Nature*, 529(7587):484–489, 2016.

[7] R. S. Sutton and A. G. Barto. *Introduction to reinforcement learning*, volume 135. MIT Press Cambridge, 1998.