Team Member: Lingjie Kong, Ruixuan Ren
Mentor: Rahul Sarkar

Problem

Reinforcement Learning (RL) is an area regarding how agents act to an unknown environment for maximizing its rewards. Unlike Markov Decision Process (MDP) in which agent has full knowledge of its state, rewards, and transitional probability, reinforcement learning utilizes exploration and exploitation to cover model uncertainty. Due to the model usually has a large input feature space, a neural network (NN) is often used to summarize the correlation between input feature and output state action value. Our goal is to improve existing algorithms or potentially develop new algorithms, specifically double A3C. We will implement DQN, double DQN, dueling DQN and A3C (Asynchronous Advantage Actor-Critic) to play OpenAI Gym Atari 2600 games to obtain benchmark performance. Then we will propose our implementation on double A3C, an improved version of state-of-the-art A3C algorithm. We will compare its performance, data efficiency and computation efficiency to the other methods.

Literature and Background

Computer Vision leads to breakthroughs on how to extract the feature representation more efficiently [1]. All these methods utilize ideas of neural network structures such as Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN), Multilayer Perception, Boltzmann Machine Graphic Model, and so on.

Q-Learning algorithm [2] with stochastic gradient descent (SGD) is often used to train reinforcement learning model. However, we need to store and update a Q value estimate Q(*s, a*) for each (*s, a*) pair. Therefore, if we have a large state or action space, it will be expensive to store Q values for all (*s, a*) pairs. One of the common solutions to this issue is to use function approximation, where we extract features from (*s, a*) and define a function to approximate Q(*s, a*). Then optimizing the estimation of Q values turns into optimizing the parameters in.

Deep Reinforcement Learning [3] uses Deep Q-Network (DQN) to replace Q values. In most cases, research showed that the agents trained by DQN achieve high performances in playing Atari 2600 games. Further studies of more advanced DQN structures such as Double DQN [4] and Dueling DQN [5] proposed methods to enahnce the convergence speed and final performance.

Recently, asynchronous method has been proposed for Deep Reinforcement Learning [6]. The study showed that Asynchronous Advantage Actor-Critic (A3C), can achieved 2x faster training speed compared to DQN even if it uses a multi-core CPU instead of GPU. Moreover, agents trained by A3C can achieve higher performances in most of the Atari 2600 games than DQN models.

Approach

Our approach will be based on the Double DQN algorithm [4] and the state-of-the-art A3C algorithm [6]. The key technique in Double DQN is to use two deep neural networks with the same structure but different parameters as two approximate functions of Q values. This technique can reduce the overestimations of action values under certain conditions and improve the agent performance. We believe the same technique can also be applied to A3C algorithm. To be more concrete, we can use two sets of parameters of the same neural network to approximate the values of states. We hope to achieve higher or at least the same performance compared to the vanilla A3C algorithm.

We will train and evaluate our approach using the environment of OpenAI Gym Atari 2600 games. The input will be the screen images in each game and the output will be the optimal policy. We will compare the average performances of agents trained by existing algorithms (especially vanilla A3C) and by double A3C. Moreover, we will analyze whether our method will benefit or harm the convergence speed of A3C algorithm.

References

[1] A. Krizhevsky, I. Sutskever, and H. Geoffrey E., “ImageNet Classification with Deep Convolutional Neural Networks,” *Adv. Neural Inf. Process. Syst. 25*, pp. 1–9, 2012.

[2] C. J. C. H. Watkins and P. Dayan, “Q-Learning,” *Mach. Learn.*, vol. 8, no. 3–4, pp. 279–292, 1992.

[3] V. Mnih *et al.*, “Playing Atari with Deep Reinforcement Learning,” *arXiv*, vol. 32, no. Ijcai, pp. 1–9, 2016.

[4] H. Van Hasselt, A. Guez, D. Silver. “Deep Reinforcement Learning with Double Q-learning,” in proceedings of AAAI 2016.

[5] Z. Wang *et al.*, “Dueling Network Architectures for Deep Reinforcement Learning,” in proceedings of ICML 2016.

[6] V. Mnih *et al.*, “Asynchronous Methods for Deep Reinforcement Learning,” in proceedings of ICML 2016.