Model-Based or Model-Free, a Review of Approaches in Reinforcement Learning

Qingyan Huang
School of Software
Xinjiang University
Urumqi, China
761217831@qq.com

***Abstract*—Reinforcement learning (RL) algorithms can successfully solve a wide range of problems that we faced. Because of the Alpha Go against KeJie in 2017, the topic of RL has reached the completed new level of public opinion. Usually, reinforcement learning includes two categories, model-based method and model-free method, each of which shows unique advantages. Model-free RL can successfully solve various tasks, which can play video games and solve robotic tasks, but requires many samples to realize good performance. Model-based RL can quickly obtain near-optimal control by learning the model in a rather limited class of dynamics. In this situation, knowledge about the environment can be acquired in an unsupervised setting, even in trajectories where no rewards are available. However, its disadvantages lie in that most model-based algorithms learn local models over-fitting several samples by depending on simple functional approximators, usually one mini-batch. The main body of this paper is going to summarize the classic algorithms in RL. Also, in the discussion part, new approaches are discussed, to keep the strength in model-based and model-free algorithms.**

Keywords—review, reinforcement learning, model-based RL, model-free RL, Q-learning

I. Introduction

Being a branch of machine learning, reinforcement learning (RL) enables agents to maximize learning rewards by experimenting in an interactive environment and learning reinforcement based on their own actions and errors in empirical feedback. Reinforcement learning is usually divided into model-free RL and model-based RL according to whether the environment is understood first or not. The biggest advantage of having a model lies in that it allows agents to plan by thinking ahead, seeing what happens to a range of possible options, and deciding on their plan. Model-free RL can be further divided into Q-learning and policy optimization according to different learning objects. And model-based RL can be divided into learn the model and given the model. In the area of robot design, the application of RL includes the algorithm of controlling robot arm motion proposed by Christopher and the learning algorithm of robot football learning by Peter Stone [1]. There are also applications of RL in the game, such as Samuel's checkers system [2], which uses a lot of self-learning to defeat human players. In this paper, we will summarize the research status of RL in the world. Firstly, this paper introduces two major categories of reinforcement learning. Then it introduces the principle, structure and implementation method of the examples below these two categories. Finally, a brief conclusion will be made.

II. Model-free RL

As one part of RL, model-free algorithm does not use the transition probability distribution and reward function related to markov decision process (MDP) since the transition probability distribution and reward function are generally referred to as the model of environment (or MDP), which express why it is called “model-free RL”. Q-learning is a typical example of model-free algorithms.

Model-Free RL does not try to understand the environment. It will give actions according to the environment changes and feedbacks. There are two main ways in the model-free RL.

A. Q-learning

Learning the policy is the goal of Q-learning, which tells the agent what action to take under different conditions. Requiring no environment models, it can deal with problems with stochastic transitions and rewards without adaptation.

Q is $Q(s,a)$, which expects that the agent takes actions $a\ (a \in A)$ in the $s\ (s \in S)$ state at some time. And the agent will be rewarded by the environment based on those actions. Therefore the main thought of this algorithm is building action and state into a Q-table to store Q value. The Bellman equation is employed to define the updated target of Q value, and then Q value is used to choose the action which achieves the maximum reward. But today, we are confronting with increasingly complicated problems. The possible state of state is very large (such as playing go game). If we store them all in a table, our computers may not have enough memory. And it is a waste of time that we search the corresponding state in such a large table every time, which becomes an obstacle to the traditional form of RL. Therefore, people proposed a variant of Q learning, deep Q-learning (DQN). Instead of recording the Q value in the Q table, DQN predicts the Q value by use of neural network. Besides, it is committed to learning the optimal action path by continuously updating the neural network. At present, DQN’s excellent performance in image processing has enabled it to beat human experts in some domains of Atari games [3].

In 2017, Deep Mind and Cambridge University proposed a new method of RL, distributional reinforcement learning with quantile regression (QR-DRL) [4], which points out a more promising direction for the future development of RL. This method replaces the expectation value of learning reward with the probability of learning reward. Deep Mind’s paper proved the powerful performance of QR-DRL through experiments in Atari games and achieved state-of-the-art

978-1-7281-7106-7/20/$31.00 ©2020 IEEE
DOI 10.1109/CDS49703.2020.00051

techniques in many games. QR-DRL restricts to a discrete set of quantiles, automatically adapting return quantiles to minimize the Wasserstein distance between the Bellman updated and current return distributions.

B. *Policy Optimization*

A series of methods for policy optimization is the policy-based approach to RL algorithms. These methods explicitly identify policies as $\pi\theta(a|s)$. Usually, the parameter θ will be optimized by the gradient ascent of the performance index $J(\pi\theta)$. Or the optimization is conducted by maximization of the local approximation. All of these optimizations are undergone with the on-police. It means that each update only uses the data that was collected when it was executed based on the latest version of the policy.

Policy gradient [5] updates the policy parameter θ by optimizing the target function $J(\theta)$ (like accumulated sum of rewards) in the stochastic policy $\pi(\theta)$. Apart from approximating a value function and using it to calculate a deterministic policy, Richard S. Sutton et al. [5] used an independent function approximator with its own parameters directly to approximate a stochastic policy so as to improve its performance.

However, in the policy gradient method, it is difficulty to choose a proper learning step to obtain an ideal learning direction. An improper learning step will lead to inefficient learning rate or even cause learning collapse. John Schulman et al. made several approximations to the theoretically-justified procedure to propose a new method named trust region policy optimization (TRPO) [6], to select proper learning steps. The reward function in TRPO keeps monotone increasing or non-decreasing, to obtain better values. TRPO has been proved good performance in playing Atari games, learning simulated robotic swimming, walking gaits and hopping. In 2017, John Schulman et al. proposed an improved TRPO method, proximal policy optimization (PPO) [7]. It updates the policy by use of multiple epochs of stochastic gradient ascent. In addition to having the stability and reliability of TRPO, PPO is much simpler to implement, which is applicable in more general settings and has better overall performance.

III. MODEL-BSAED RL

Unlike model-free RL, model-based RL has a model in RL which enables agents to think in advance, seeing what will happen due to a series of possible choices, and then making choices. The agent can extract the results from advance planning into the learning strategy. AlphaZero is a typical application of a model-based RL algorithm. Currently, AlphaZero go have achieved skills with an unprecedented height beyond human beings in this domain.

Model-free deep RL methods have shown the ability to learn a variety of tasks, from playing video games and images [3, 8] to learning complicated locomotion skills. However, the sample complexity of model-free algorithms tends to be limited to physical systems particularly when high-dimensional function approximators are used.

Fortunately, model-based RL can quickly obtain near-optimal control under fairly restricted dynamics classes with learned models. With this advantage of model-based RL, different teams propose different approaches to overcome the shortcomings of model-free RL.

A. *Learn The Model*

Learn the model uses the sampling data to train the policy of model-free RL or the Q function. It trains the agent with virtual experience. The training data can be obtained by the model sampling, or the combination of virtual data or real data.

In 2018, by integrating model-base and model-free learning techniques through disciplined model use for value estimation, Vladimir Feinberg et al. attempted to reduce sample complexity and remain complex nonlinear dynamics. They proposed model-based value expansion (MBVE) [9] by enabling wider use of learned dynamics models within a model-free reinforcement learning algorithm， which is a hybrid algorithm adopting a dynamics model to simulate Q-learning and the short-term horizon so as to calculate the long-term value beyond the simulation horizon.

Compared with most model-based reinforcement learning and planning methods which direct how a model should be used to reach a policy, an original architecture for deep reinforcement learning proposed by Théophane et al. [10] combines model-based and model-free strengths, which is named imagination-augmented agents (I2A) and established by employing the predictions as additional context in deep policy networks. I2As demonstrate better performance, data efficiency, and robustness to model misspecification than some baselines. When predictions are used as additional context in deep policy networks, I2A learns to interpret predictions from a learned environment model to create implicit plans arbitrarily. Moreover, I2A shows better performance than model-free baselines in many domains such as Sokoban. It can be said that with less data it performs better.

Anusha Nagabandi et al. [11] prove that excellent sample complexity actually can be achieved in a model-based RL algorithm by combining medium-sized neural network models with model predictive control (MPC) . In addition,, they propose initializing a model-free learner by employing deep neural network dynamics models. It has turned out that their hybrid algorithm can speed up model-free learning on high-speed benchmark tasks, thus achieving sample efficiency gains triple to quintuple efficiency on swimmer, cheetah, hopper, ant agents, and so on.

David Ha et al. put forward world models [12] in 2018, which can be easily trained to learn a compressed spatial and temporal representation of the environment in an unsupervised way. To be specific, we can use features extracted from the world model as inputs to an agent in order to train a very compact and simple policy that can solve the required task. Specially, the agent can be trained completely inside of its own hallucinated dream generated by its world model; and then the policy can be transferred back into the actual environment. The training process in the world models is called "training in the dream" because all the data used for training is virtual.

B. *Given the Model*

Differing from learn the model created by the agent, given the model gives the entire model to agent. AlphaGo is one of the classical given the models in the model-based RL.

AlphaGo utilizes Monte-Carlo tree search, with value network and policy network. Value network is used to evaluating massive possible site selection, and policy network is used for positioning. Initially, AlphaGo attempted to match past games of professional players by emulating them, which contains about 30 million moves in the database. After reaching a certain level, it used RL to get improved and started playing numerous chess games against itself. With this design, the computer can integrate the long-term inferences of a tree graph with the intuitive training that spontaneously learned by the human brain to strengthen its chess skills.

As AlphaGo's successor, AlphaZero [13] demonstrated extraordinary performance in chess and shogi (Japanese chess) games within 24 hours like Go, forcefully beat a world championship in each case. Even when applied to the more challenging game of shogi without modification, the same algorithm again beat the best in just several hours.

IV. Discussion

Although reinforcement learning has reached an unprecedented height, we still face many challenges in the process of applying the algorithm in practice. In general, reinforcement learning includes model-free and model-based algorithms. Model-free deep reinforcement learning algorithms can successfully deal with a variety of continuous control tasks, from playing video games from raw pixels [3, 8] to solving locomotion and robotic tasks [14]. However, plenty of samples are required to achieve good performance. Model-based algorithms can swiftly reach near-optimal control with learned models under fairly restricted dynamics classes. The essential advantage of model-based RL lies in that knowledge about the environment can be acquired in an unsupervised setting, even in trajectories with no rewards. Nevertheless, the most successful model-based algorithms existing learn local models over-fitting a few samples by use of simple functional approximators [15], usually one mini-batch. Therefore, ideas that depending on a global neural network model for planning often perform unsatisfactory [16, 17].

In order to keep the strength in both model-free and model-based methods, researchers have made great efforts. Tong Che et al. [18] leveraged multi-step neural network based predictive models by embedding real trajectories into imaginary rollouts of the model and used the imaginary cumulative rewards as control variates for model-free algorithms. In this way, they derived an estimator which is not only sample-efficient, but also unbiased and of very low variance. In 2016, Gu et al. [17] derived a continuous variant of the Q-learning algorithm (NAF), which can be used to replace the more general policy gradient and actor-critic methods. To further improve the efficiency of NAF, they tried to accelerate model-free reinforcement learning by use of model-based algorithms. In 2017, Nagabandi et al. [11] attempted to train a neural network based global model and employed model predictive control to initialize the policy for model free fine tuning so as to construct an algorithm being drastically more sample-efficient than purely model-free designs.

V. Conclusion

In this paper, we discuss the current development of RL by its classifications and describe some classical methods of reinforcement learning and their implementation principles. Meanwhile, considering the current challenges facing model-based and model-free methods, we explore how to combine them to achieve better performance. Because of the Alpha Go against KeJie in 2017, the topic of RL has reached an unprecedented level of public opinion. At present, RL has made great achievements not only in the game, but also in robot control and decision making. In the future, we believe that with the many advantages of RL, automatic driving, market forecast, and more intelligent robots, will become a reality in our daily life.

References

[1] Peter Stone. Layered Learning in Multi-Agent Systems: A Winning Approach to Robotic Soccer, 2000

[2] Samuel A L. Some studies in machine learning using the game of checkers. IBM Journal of Research and Development, 1959

[3] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning, 2013

[4] Will Dabney, Mark Rowland, Marc G. Bellemare, Rémi Munos Distributional Reinforcement Learning with Quantile Regression, 2017

[5] Richard S. Sutton, David McAllester, Satinder Singh, Yishay MansourPolicy Gradient Methods for Reinforcement Learning with Function Approximation, 2000

[6] John Schulman, Sergey Levine, Philipp Moritz, Michael Jordan, PieterAbbeel, Trust Region Policy Optimization, 2017

[7] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, Oleg Klimov OpenAI, Proximal Policy Optimization Algorithms, 2017

[8] Junhyuk Oh, Valliappa Chockalingam, Satinder Singh, and Honglak Lee.Control of memory, active perception, and action in minecraft, 2016

[9] Vladimir Feinberg, Alvin Wan, Ion Stoica, Michael I. Jordan, Joseph E. Gonzalez, Sergey Levine, Model-Based Value Expansion for Efficient Model-Free Reinforcement Learning, 2018

[10] Théophane Weber∗Sébastien Racanière∗David P. Reichert∗Lars Buesing Arthur Guez Danilo Rezende Adria Puigdomènech Badia Oriol Vinyals Nicolas Heess Yujia Li Razvan Pascanu Peter Battaglia Demis Hassabis David Silver Daan Wierstra DeepMind, Imagination-Augmented Agents for Deep Reinforcement Learning, 2018

[11] Anusha Nagabandi, Gregory Kahn, Ronald S. Fearing, Sergey Levine University of California, Berkeley, Neural Network Dynamics for Model-Based Deep Reinforcement Learning with Model-Free Fine-Tuning, 2017

[12] David Ha, Jürgen Schmidhuber, World Models, 2018

[13] David Silver,Thomas Hubert,Julian Schrittwieser, Ioannis Antonoglou,Matthew Lai,Arthur Guez,Marc Lanctot,Laurent Sifre, Dharshan Kumaran,Thore Graepel, Timothy Lillicrap,Karen Simonyan,Demis Hassabis, Mastering Chess and Shogi by Self-Play with a General Reinforcement Learning Algorithm, 2017

[14] John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In Proceedings of the 32nd International Conference on Machine Learning, 2015

[15] Rudolf Lioutikov, Alexandros Paraschos, Jan Peters, and Gerhard Neumann.Sample-based informationl-theoretic stochastic optimal control. In Robotics and Automation (ICRA), 2014 IEEE International Conference on, 2014

[16] Nikhil Mishra, Pieter Abbeel, and Igor Mordatch. Prediction and control with temporal segment models. arXiv preprint arXiv:1703.04070, 2017

[17] Gu, Shixiang, Lillicrap, Timothy, Sutskever, Ilya, and Levine, Sergey. Continuous deep q-learning with model- based acceleration. In International Conference on Ma- chine Learning, 2016

[18] Tong Che, Yuchen Lu, George Tucker, Surya Bhupatiraju, Shane Gu, Sergey Levine, Yoshua Bengio, Combining Model-based and Model-free RL via Multi-step Control Variates, 2018.