RL_Methods_Week_5

November 18, 2020

1 Approaches

According to Vincent François-Lavet, Peter Henderson, Riashat Islam, Marc G. Bellemare and Joelle Pineau (2018), there are three major approaches to solve reinforcement learning problem.

1.1 Value Funtion

In naive Q learning, we can memorize Q∗(.) for all state-action pairs in Q-learning. However, the dictonary grows fast, so we use functions with parameter to approximate Q values and this is called function approximation.

- Q-Learning Network in CartPole https://www.tensorflow.org/agents/tutorials/1_dqn_tutorial

1.2 Policy gradient methods

Always combine with Value Function. Optimize policy parameters using stochastic gradient descent.

- Monte-Carlo Policy Gradient in CartPole https://lilianweng.github.io/lil-log/2018/05/05/implementing-deep-reinforcement-learning-models.html#monte-carlo-policy-gradient
- Temporal Difference Learning in backgammon https://github.com/fomorians/td-gammon

1.3 Model-based

Best when MDP can't be learned.

- Monte Carlo Tree Search https://github.com/DylanSnyder31/AlphaZero-Chess

1.3.1 Markov decision process(MDP)

The Markov property means that the future of the process only depends on the current observation, and the agent has no interest in looking at the full history.

2 Code Source

Most projects are coded in tensorflow==1.X.X, which is no longer available in Python 3.8. To run these code, either build a vertual environemnt in python 3.6-3.7 or upgrade the code. https://www.tensorflow.org/guide/upgrade