Learning to Play Modified version of Breakout Atari Game Using Actor-Critic with Eligibility Trace

Amin Majdi

Fall 2020

1 Introducing

The aim of this work is to use reinforcement learning (RL) methods for developing an intelligent agent that is able to play a simple game similar to the Atari breakout game and maximize the reward.

Game Description: The game idea comes from the breakout game. In the simple version of this game, there is only one button in the top of the game window, there is one moving ball and there is one paddle in the bottom of the window that should be used to hit the ball to the button in the top border of the game window to gain a reward. Each time ball bounce to any of the borders of the game window, it returns, except the bottom border. If ball bounce to the bottom border, game is over. Therefore, the episode will end in either conditions that the agent loses (ball bounce to the bottom border) or win (ball bounce to the button). In the advance version of the game,another button is added on the top boarder of the game window (magic button). Every time the ball hit to that, if there is one ball in the game, the button adds another ball and if there are two balls in the game the button omits that second ball. In this game, the best strategy is creating a second ball and trying to hit that extra ball to the reward button. This way, not only the agent receives the reward, but also the game will continue. So the agent should try to hit the magic button and create the second ball and hit the reward button and continue this circle to maximize the reward.Figure 1 shows advanced version of this game.

States and Actions: In this game actions are move left, move right and stay. States are different for simple and advanced version of the game. In simple version, states are defined based on location of ball, angle of ball movement, and location of the paddle. In advanced version of the game location and angle of the second ball should also be added to the states.

Game Implementation: For Implementation of this game I used pygame module [1].

2 Method

I used Actor-Critic with Eligibility Trace for learning agent to successfully play this game. In this approach both policy and value function are learned by the use of approximation

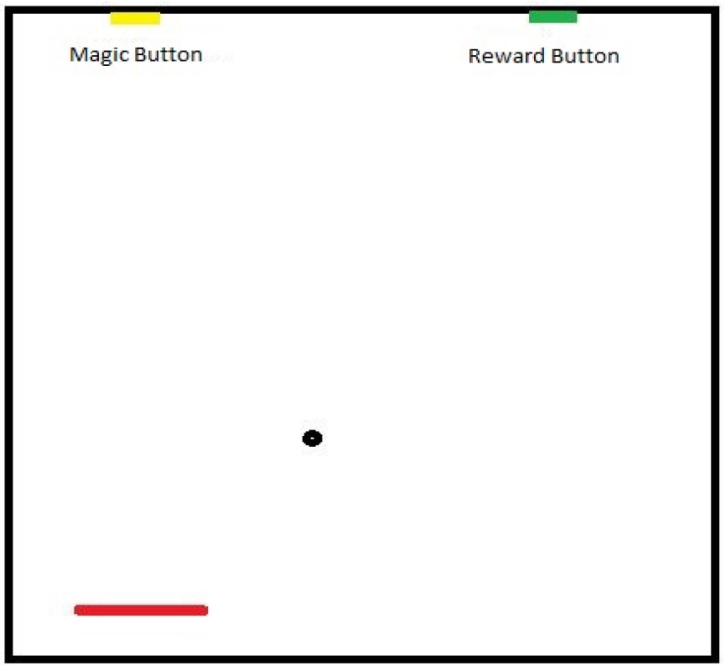

Figure 1: This figure shoes the advance version of the game in which there is also a magic button

methods. Value function is estimated as a linear function of weight vector and constructed features with Fourier Basis method [2] using equation (1).

$$\hat{V}(s, w) = w^\top x(s) = \sum_{i=1}^{d} w_i x_i(s) \tag{1}$$

For policy approximation exponential soft-max distribution is used, equation (2):

$$\pi(a|s, \theta) = \frac{e^{h(s,a,\theta)}}{\sum_b e^{h(s,b,\theta)}} \tag{2}$$

and action preference in equation (2) is parametrizad using equation (3):

$$h(s, a, \theta) = \theta^\top x(s, a) \tag{3}$$

Reward Function: In the simplified version of my game, in each step of an episode, agent receive a negative reward which its value is proportional to the distance between paddle and ball in the direction of x axis. Each time ball hits the paddle, the agent receives a positive reward equal to 1000. If the paddle miss the ball and ball hits the bottom border, agent receives -1000 reward, and if agent hit the ball to the reward button, it will receive a reward equal to 10000.

3 Results

I started with a very simplified version of this game, in which magic button is removed and there is only one ball in the game. In this version of the game ball always initiates from the same state (e.g., same location and same angle). In this case, the agent learns to hit the ball with the paddle and repeat that until the ball bounce the reward button. As it is a multi-goal game, its learning has different steps. It first learned to hit the ball once with the paddle, then it learned to hit the ball with paddle more than one time, and finally it learned to hit the ball to the reward button after five consecutive hits to the paddle in 50000 episodes. I also made a second version of the game in which ball returns with different angles when it hits the paddle. If the ball hits the paddle in the middle, it will return with same angle, but if the ball hit the paddle in two one fourth end of the paddle the return angle change by $\pi/8$. My agent also learned to play this game and hit the ball to the reward button in 2000 episodes.

My ultimate goal the agent learn the optimal policy with random initial state.

References

[1] P. Shinners, "This is how you cite a website in latex," https://www.pygame.org/docs/tut/PygameIntro.html, May 2000.

[2] G. Konidaris, S. Osentoski, and P. Thomas, "Value function approximation in reinforcement learning using the fourier basis," 2011.