Deep Q-Network Demo

Rishikesh Vaishnav

July 1, 2018

Monte Carlo Implementation

Code

- The code for this project is available at: `https://github.com/rish987/Reinforcement-Learning/blob/master/demos/policy_gradient/code/policy_gradient.py`.

Implementation Details

- Unlike the Atari gameplay environment described by Mnih et. al., the pole-cart environment is not perceptually aliased. That is, the current observation of the state is theoretically all that is needed to determine an optimal value. Therefore, the current state can be equated with the current observation, without taking into account past observations and actions.
- Because the observation space of the Atari gameplay environment is much larger than the pole-cart environment, it should suffice to use a smaller ANN model.
- Because the observation space of the pole-cart environment is small and not spatially correlated, it is not helpful to use a convolutional neural network.

Results

- The results can be summarized as follows:
  - The largest learning rate initiates learning quickly but fails to converge to an optimal policy, likely because it overshoots the mark at each parameter update.
  - The smallest learning rate learns the policy slowly because of its smaller updates but does reach a near-optimal policy.
  - The middle learning rate finds a near-optimal policy relatively quickly.