Deep Q-Network Demo

Rishikesh Vaishnav

July 2, 2018

Basic Implementation

Code

- The code for this project is available at: `https://github.com/rish987/Reinforcement-Learning/blob/master/demos/deep_q_network/code/deep_q_network.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.
- The model is a simple vanilla neural network with one hidden layer:
  - Let M be the number of nodes in the hidden layer.
  - Let K be the number of output nodes (i.e., number of actions).
  - The hidden layer is calculated as:

$$Z_m = \sigma(\alpha_{0m} + \alpha_m^T x(s)), m = 1, \ldots, M$$

  - The output layer is calculated as:

$$\hat{q}(s, a_i) = \beta_{0i} + \beta_i^T Z, i = 1, \ldots, K$$

- Solving for the

Results

- The results can be summarized as follows:
  -