Solving the Banana Collector Environment with Deep Q-Networks

Jason Kutarnia

December 1, 2018

1 Learning Algorithm

This repository implements the basic Deep Q-Network (DQN) described in the the work by Mnih et al., which was published in Nature. Key features of this architecture include the definition of local and target networks as well as a replay memory which is uniformly sampled before each learning step. The dual networks aid in the optimization of the weights by preventing them from changing erratically. The replay memory addresses the correlation problem between states occurring close together in time which keeps the network from reinforcing undesirable behaviors.

To solve the banana collector problem a fully connected network was constructed with 2 hiddens layers containing 64 neurons each. Since there are 4 actions in the game, the output layer contained 4 neurons as well. The replay memory stored the previous 10,000 states which was uniformly sampled to produce a mini-batch of 64. Using the Adam variant of gradient descent, the agent took an optimization step after every 4 time steps in the banana environment. The learning rate was 1e-3 and the discount factor was .99. The target network used a soft update rate of .001 to smooth out the optimization.

2 Plot of Rewards

Figure 1 shows that the DQN agent successfully solved the environment after 490 episodes, which occurs when the agent's average performance over the last 100 episodes is at least 13 bananas.

3 Ideas for Future Work

Some improvements to Deep Q-Networks have been proposed in the literature and include Double DQN, Dueling DQN, and Prioritization Replay. These additions may improve the performance of the banana collector but remain to be tested. There also appears to be a hard limit at 16 bananas where

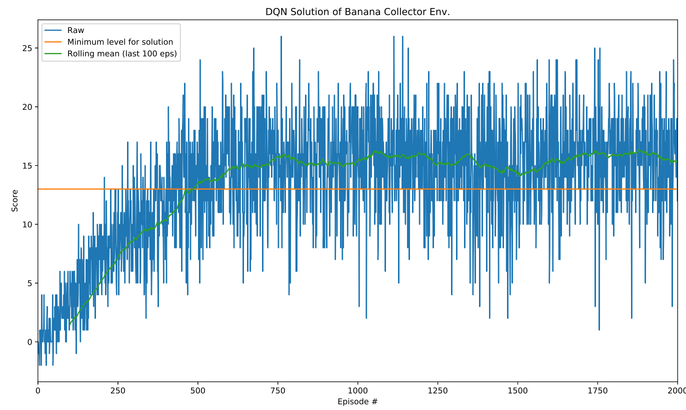

Figure 1: Plot of rewards for DQN agent

additional training doesn't improve the agent's ability. A more complicated network architecture may help the agent surpass this limit.