Deep Q-Learning for Navigation Report

Dalton J. Schutte

Background

Deep Q-Learning was first proposed by Minh et al. in a 2015 Nature paper, Human-Level Control Through Deep Reinforcement Learning, that utilized a combination of Q-Learning, neural networks (deep Q-Networks or DQNs), and experience replay to teach an agent to play Atari games. Since that initial paper several improvements have been made to Deep Q-Learning methods. Double DQNs, dueling network architectures, distributional DQNs, prioritized experience replay, noisy nets, and others have resulted in various improvements to the basic DQL algorithms.

Problem

In this project, the goal was to train an agent to navigate in space to collect yellow bananas while avoiding blue bananas. Each yellow banana was worth +1 point and each blue banana was worth -1 point. The task was considered solved if the agent could obtain an average score of +13 over 100 consecutive episodes, an episode being 300 time steps in a new environment. The environment states have 37 values and the agent can choose one of four actions at each time step, forward, left, right, or backward.

Solution

A double deep Q-Learning approach was used, and the network architecture was a dueling network architecture with experience replay. The specific architecture is outlined in the table below. The output from the second fully connected layer was fed to each of the advantage and value arms. The output from the advantage arm and the value arm were added and the mean of the output of the advantage arm was subtracted as van Hasselt et al. 2015 found that this improved stability.

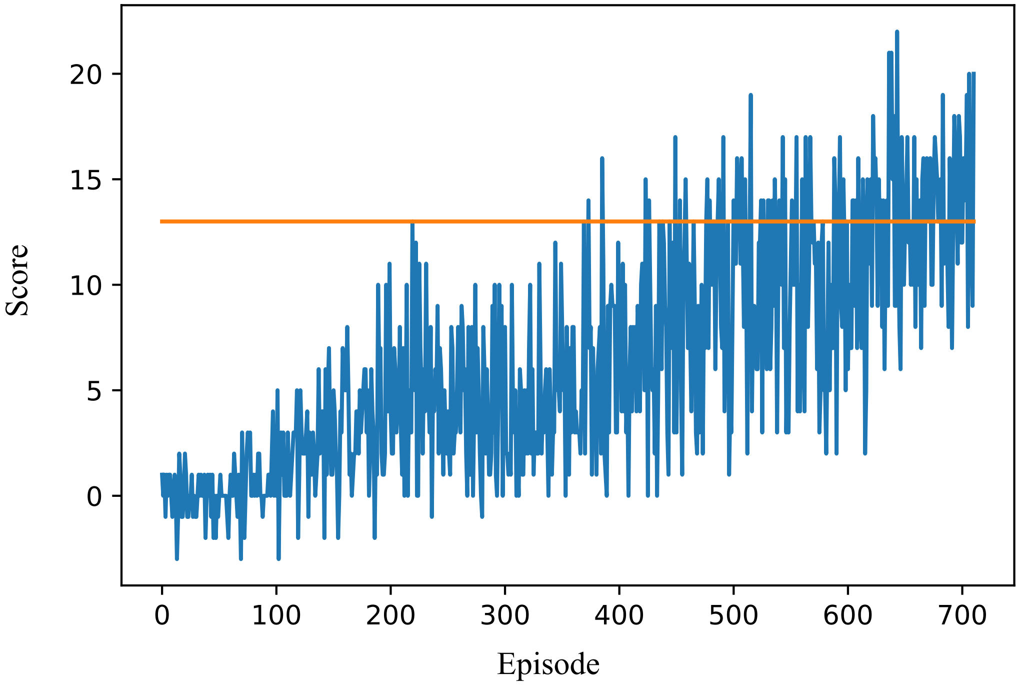
Hyperparameters

The replay memory was initialized to a capacity of 10,000 of the most recent experiences and would sample 64 experiences uniformly without replacement. The learning rate for the Adam optimizer was set to 0.0005, the discount factor to 0.99, the update interpolation rate to 0.0004, the update frequency to every four time steps, epsilon minimum to 0.05, and epsilon was linearly decreased from 1 to 0.05 over 50,000 time steps. The random seed was set to 1.

Layer (Activation)	Input Dimension	Output Dimension
Fully Connected (ReLU)	37	64
Fully Connected (ReLU)	64	32
Advantage FC (ReLU)	32	32
Advantage Out (None)	32	4
Value FC (ReLU)	32	32
Value Out (None)	32	1

Results

The agent solved the task in 711 episodes and attained an average of 13.02 over the last 100 episodes. The orange line is at a score of 13.



Conclusion

To improve the training, various improvements could be used. Based on the results in the ablation studies of the RAINBOW paper in Hessel et al. 2017, prioritized experience replay, multi-step learning, and distributional learning would likely yield the most return as the studies suggest they contribute most to the performance of the RAINBOW algorithm. The two improvements used in this implementation, dueling and double DQN, showed almost no change in RAINBOW's performance when they were removed, suggesting that their contribution is minimal when combined with other improvements.