

Deep Reinforcement Learning Nanodegree Program

Navigation

Atauro Chow

Deep Reinforcement Learning Nanodegree Program

Navigation

Introduction.

For this project, I will train an agent to navigate (and collect bananas!) in a large, square world. A reward of +1 is provided for collecting a yellow banana, and a reward of -1 is provided for collecting a blue banana. Thus, the goal of your agent is to collect as many yellow bananas as possible while avoiding blue bananas.

The state space has 37 dimensions and contains the agent's velocity, along with ray-based perception of objects around the agent's forward direction.

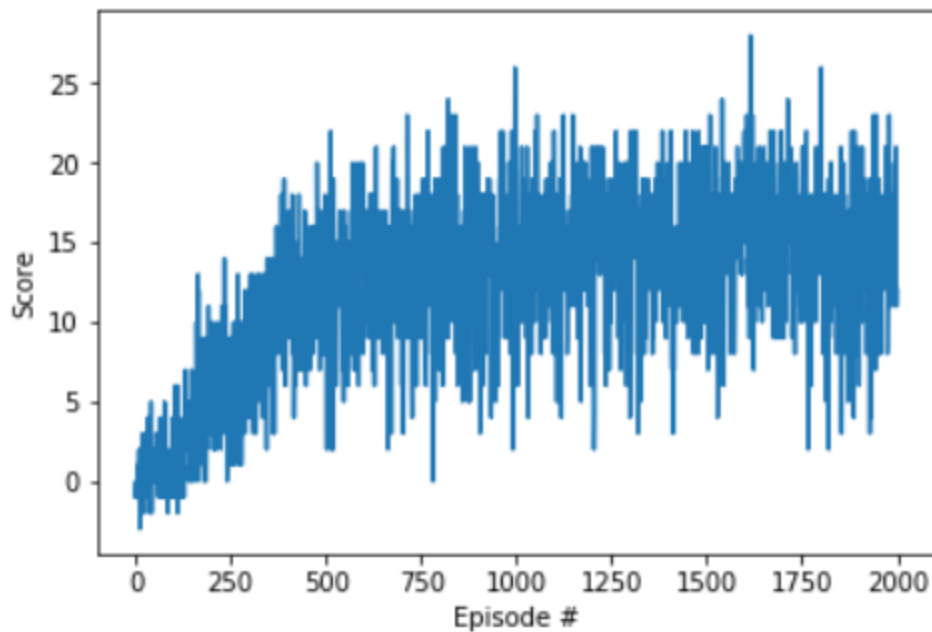
The task is episodic, and in order to solve the environment, the agent must get an average score of +13 over 100 consecutive episodes.

Algorithm

We use Deep Q-Network (DQN) as our fundamental model. DQN is widely use in solving the problems with discrete observation spaces. The model applies one neutral network and have two hidden layers with 400 and 300 units respectively. They are full-connected layers. An experience replay is implemented, and its buffer size is 10,000 with mini-batch size is 64. The agent will pick 64 entries from buffer randomly for training. The training update network every 4 steps. The learning rate is 0.0005.

Results

The average score was 13.68 at 900 episodes. For the agent training, the problem was solved (over score 13) after 900 episodes.



Idea for future work

We believed the following implementation can improve the current result.

- Prioritized Experience Replay: Currently, the agent picked records from replay buffer randomly. It would be benefit if good behavior will be picked in more frequently.
- Dynamic Frame Skip Deep Q-Network (DFDQN): The paper proved that the performance has above 70% improvement compared with DQN.

References

1. Playing Atari with Deep Reinforcement Learning (<https://arxiv.org/pdf/1312.5602.pdf>)
2. Sample code of Deep Q Network in Nanodegree program.