Project 1. Navigation

Introduction

In this project, a DQN agent is trained to navigate and collect bananas in a large, square world.

The state space has 37 dimensions and contains the agent's velocity, along with ray-based perception of objects around agent's forward direction. Given this information, the agent needs to learn how to best select actions. Four discrete actions are available, corresponding to:

- 0 – move forward
- 1 – move backward
- 2 – turn left
- 3 – turn right

The task is episodic. 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 the agent is to collect as many yellow bananas as possible while avoiding blue bananas. The environment is considered solved if the agent gets an average score of +13 over 100 consecutive episodes.

Algorithm

Deep reinforcement learning uses nonlinear function approximators to calculate the value actions based directly on observation from the environment, which is represented using a deep neural network.

To address the instabilities from reinforcement learning, the following two key features are used in the Deep Q-Learning algorithm:

Experience Replay – a replay buffer is used to contain a collection of experience tuples. By sampling a small batch of tuples from the replay buffer, it can break harmful correlations, learn more from individual tuples multiple times, and recall rare occurrences.

Fixed Q-Targets- to avoid harmful correlations by updating guess based on another guess, the parameters of the network, w, are updated with the following update rule:

$$\Delta w = \alpha \cdot \overbrace{(\underbrace{R + \gamma \max_a \hat{q}(S', a, w^-)}_{\text{TD target}} - \underbrace{\hat{q}(S, A, w)}_{\text{old value}})}^{\text{TD error}} \nabla_w \hat{q}(S, A, w)$$

where w^- are parameters of a target network that are not changed during the learning step.

For the detailed algorithm, please refer to reference [1].

Experimental Evaluation

This environment is solved with 429 number of episodes to reach an average score of +13.04 with 100 consecutive episodes. The plot of rewards per episode is displayed in Figure 1 below.

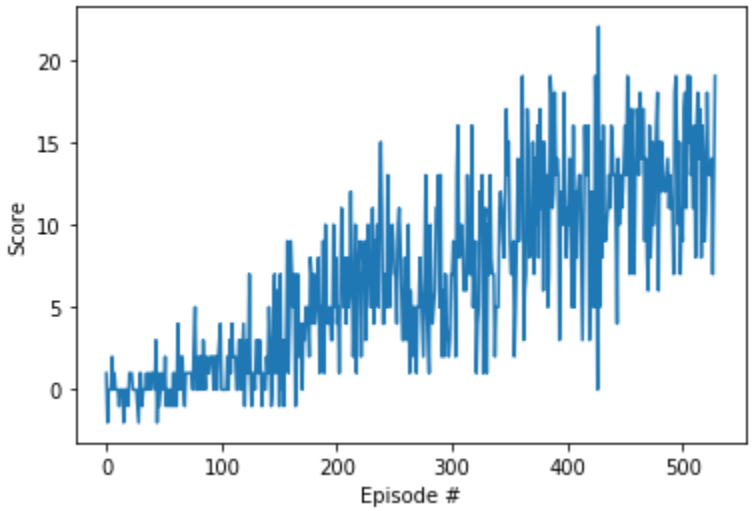

Figure 1. Episode # vs Score

Future Work

To further improve the performance of the Deep Q-Learning algorithm described above, the following improvements should be considered to avoid overestimation of Q-values and prioritize experience that are important but infrequent:

- Double DQN
- Dueling DQN
- Prioritized Experience Replay

Reference

[1]. Mnih, V. et al. Human-level control through deep reinforcement learning. Nature 518, 529–533 (2015)