deep_reinforcement_learning_project1

Overview

This project trains 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. Given this information, the agent has 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, my agent get an score of +15 over 100 consecutive episodes which is larger than +13 which the project requires.

Code Architecture

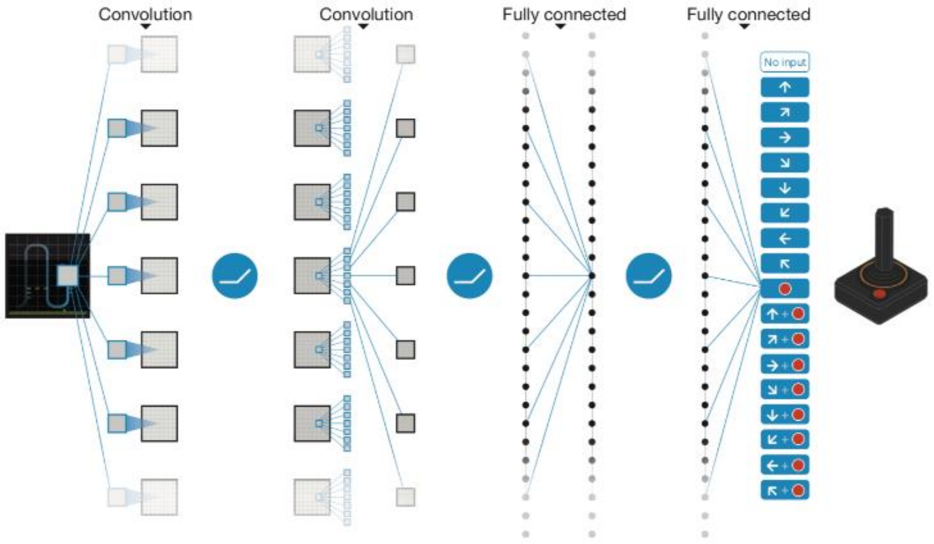
Navigation.ipynb: jupyter notebook based solution

dqn_agent.py: DQN agent code

model.py: Q-Network based model

checkpoint.pth: weights of the DQN model

Learning Algorithm



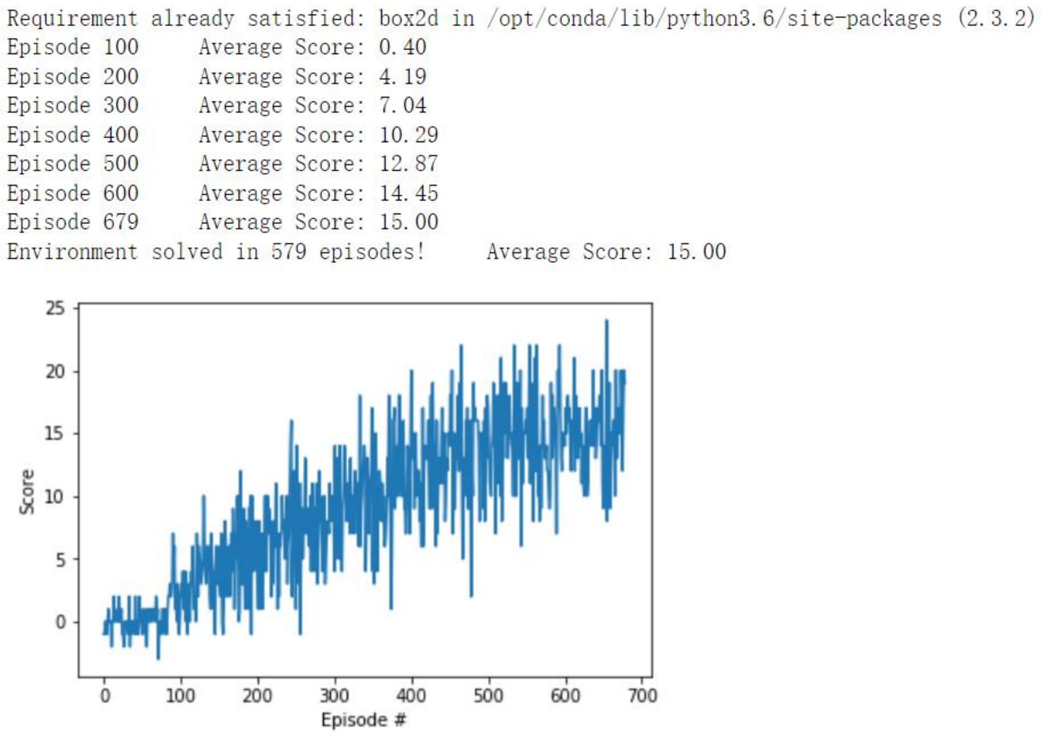
*) Picture from Udacity

DQN is implemented as the deep reinforcement learning algorithm with parameter as below:
n_episodes=2000, max_t=1000, eps_start=1.0, eps_end=0.01, eps_decay=0.995

QNetwork and ReplayBuffer is used to build the agent class with the parameter as below.
BUFFER_SIZE = int(1e5)

```
BATCH_SIZE = 64
GAMMA = 0.99
TAU = 1e-3
LR = 5e-4
UPDATE_EVERY = 4
```

Plot the score



```
Requirement already satisfied: box2d in /opt/conda/lib/python3.6/site-packages (2.3.2)
Episode 100	Average Score: 0.40
Episode 200	Average Score: 4.19
Episode 300	Average Score: 7.04
Episode 400	Average Score: 10.29
Episode 500	Average Score: 12.87
Episode 600	Average Score: 14.45
Episode 679	Average Score: 15.00
Environment solved in 579 episodes!	Average Score: 15.00
```

Ideas for future work

Double DQN and Dueling DQN could be further investigated if they have better results.