Report:

CRITERIA:

Learning Algorithm

The report clearly describes the learning algorithm, along with the chosen hyperparameters. It also describes the model architectures for any neural networks.

1) Learning algorithm

Vanilla Q-learning – initializing our Q-table, choosing an action based on the epsilon-greedy exploration strategy, and updating the Q-table using the Bellman Equation

2) Q-Network's architecture

Fully connected layer 1 with ReLU (input: 37 units (state_size), output: 64 units)

Fully connected layer 2 with ReLU (input: 64 units, output 64 units)

Fully connected layer 3 (input: 64 units, output: 4 units (action_size))

3) Parameters used in DQN algorithm

Maximum steps per episode: 1000

Epsilon-greedy decay: 0.995 (eps_start: 1.0, eps_end: 0.01)

(The following added by the reviewer's comments)

4) Parameters used in DQN Agent (left as the basic settings introduced in 2.2.7.)

Required

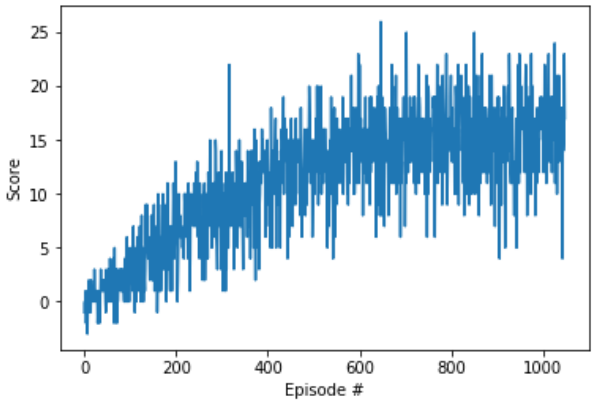
- Thank you for providing the hyperparameter values in the report.
- However, it is requested to also provide the values for the following hyperparameters as well:
  `BUFFER_SIZE, BATCH_SIZE, GAMMA, TAU, LR, UPDATE_EVERY`

```
BUFFER_SIZE = int(1e5)  # replay buffer size
BATCH_SIZE = 64         # minibatch size
GAMMA = 0.99            # discount factor
TAU = 1e-3              # for soft update of target parameters
LR = 5e-4               # learning rate
UPDATE_EVERY = 4        # how often to update the network
```

Plot of Rewards

A plot of rewards per episode is included to illustrate that the agent is able to receive an average reward (over 100 episodes) of at least +13. The submission reports the number of episodes needed to solve the environment.

```
Episode 100     Average Score: 1.00
Episode 200     Average Score: 4.42
Episode 300     Average Score: 7.87
Episode 400     Average Score: 9.45
Episode 500     Average Score: 12.42
Episode 600     Average Score: 13.88
Episode 700     Average Score: 14.67
Episode 800     Average Score: 15.75
Episode 900     Average Score: 15.23
Episode 1000    Average Score: 15.13
Episode 1048    Average Score: 16.03
Environment solved in 948 episodes!     Average Score: 16.03
```



Ideas for Future Work

The submission has concrete future ideas for improving the agent's performance.

1) **Implementing a more advanced form of DQN such as Double/Dueling DQN and Rainbow DQN.**
2) **Utilizing a CNN structure as well to directly learn from raw pixel values.**
3) **Using prioritized experience replay (PER) to make the agent able to learn from experience transitions more effectively – more important and rare experience vectors sampled with higher probability. (The reviewer's additional comments: PER should also help to significantly reduce the training time also. Using Sum Tree, a special data structure, a fast implementation of PER is possible.)**
4) **Optimizing hyperparameters more extensively**
5) **Adding parametric noise to the weights in order to induce stochasticity to the policy of the agent – more efficient exploration yielded.**