Report - Navigation

1. Environment Description

The state space has 37 dimensions and contains the agent's velocity, along with ray-based perception of objects around agent's forward direction. Four discrete actions are available, corresponding to:
- **`0`** - move forward.
- **`1`** - move backward.
- **`2`** - turn left.
- **`3`** - turn right.

2. Learning Algorithm

The solution in this submission adopts DQN as the learning model, the neural network of which takes 2 hidden layers and 1 output layer, with 64,64 and 4 nodes respectively. The number of nodes in output layer corresponds to the number of actions which has been described in the first section and each hidden layer takes “relu” as the activation function.

The details of the neural net can be found in “model.py”.

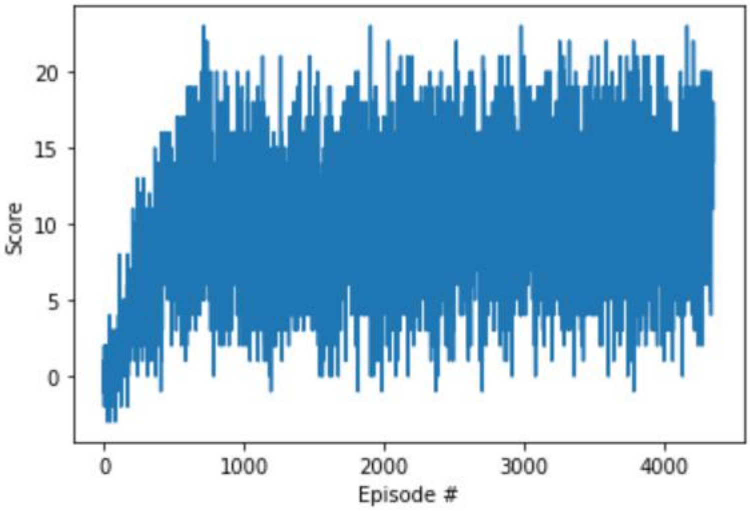
DQN takes two neural networks with identical structures, one local DQN and one target DQN. It also sets a replay memory to replay the experiences. In the submission, soft update is adopted, which updates both target and local network at the same time but uses a parameter tau to determine the parameters of the updated target network

The hyper parameters of the model is set as follows:

Replay Buffer Size: 100000
Sampled number for relay: 64
Discount factor - Gamma: 0.99
Learning rate: 0.005
Number of episodes to update target network: 4
Tau: 0.001

3. Result

The environment is solved in 4354 episodes and the plot of rewards is as below:



4. Future Work

Since this work only tries Vanilla DQN and there are a few things that can be tried in future work.

1) Prioritized Replay
2) Double DQN
3) Dueling DQN