Report

This project DQN model, agent implement project.

State and Action

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 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.

Learning algorithm

rewards + (gamma * Q_targets_next * (1 - dones))

this is important algorithm, and I am use two Qnetwork (target, main)

Hyper Parameters

n_episodes (int): maximum number of training episodes

max_t (int): maximum number of timesteps per episode

eps_start (float): starting value of epsilon, for epsilon-greedy action selection

eps_end (float): minimum value of epsilon

eps_decay (float): multiplicative factor (per episode) for decreasing epsilon

n_episodes=2000

max_t=1000

eps_start=1.0

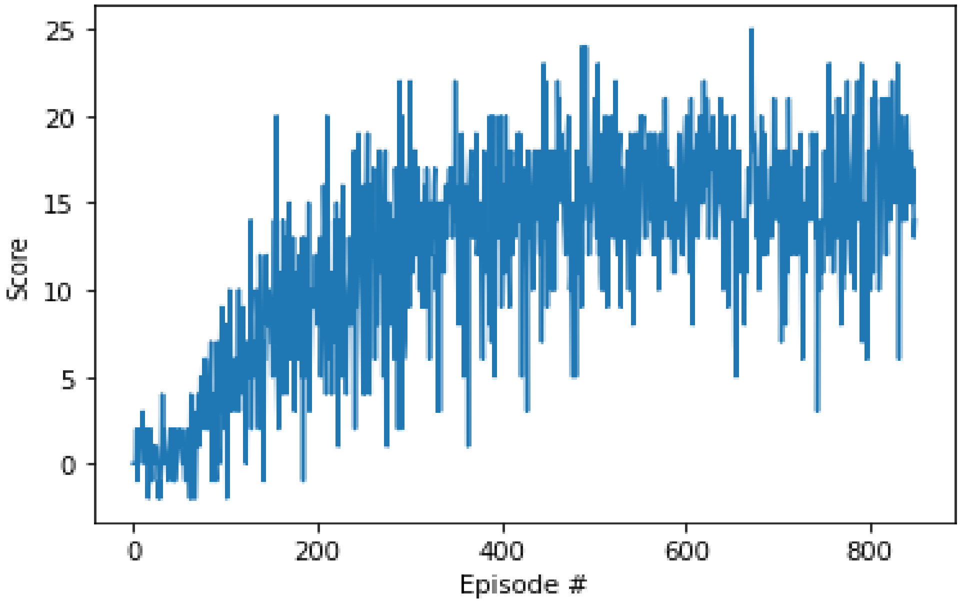
eps_end=0.01

eps_decay=0.995

Agent Hyper Parameters

```
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 Reward



```
Episode 100	Average Score: 1.35
Episode 200	Average Score: 7.36
Episode 300	Average Score: 11.06
Episode 400	Average Score: 13.69
Episode 500	Average Score: 14.86
Episode 600	Average Score: 15.38
Episode 700	Average Score: 15.66
Episode 800	Average Score: 14.66
Episode 850	Average Score: 16.01
Environment solved in 750 episodes!	Average Score: 16.01
```