

PROJECT 1: NAVIGATION

P.M. Lucaci

DEEP REINFORCEMENT LEARNING NANODEGREE, Udacity

Contents

| | |
|----------------------------|---|
| Learning Algorithm | 2 |
| Hyperparameters | 2 |
| NN Model Architecture..... | 3 |
| Plot of Rewards | 4 |
| Ideas for Future Work..... | 5 |

Learning Algorithm

DQN stands for “Deep Q-Network”, which is an extension to the Reinforcement Learning algorithm “Q-Learning”.

Q-Learning uses tuples (S,A,R,S') (i.e. State, Action, Reward, Next State) to estimate the optimal (or nearly optimal) state-action value (also known as a Q-Function). In turn, the Q-Function maximises the agent's expected cumulative reward.

Therefore, we can say that DQN is using deep neural nets to estimate the expected cumulative reward by computing the optimal action-value function

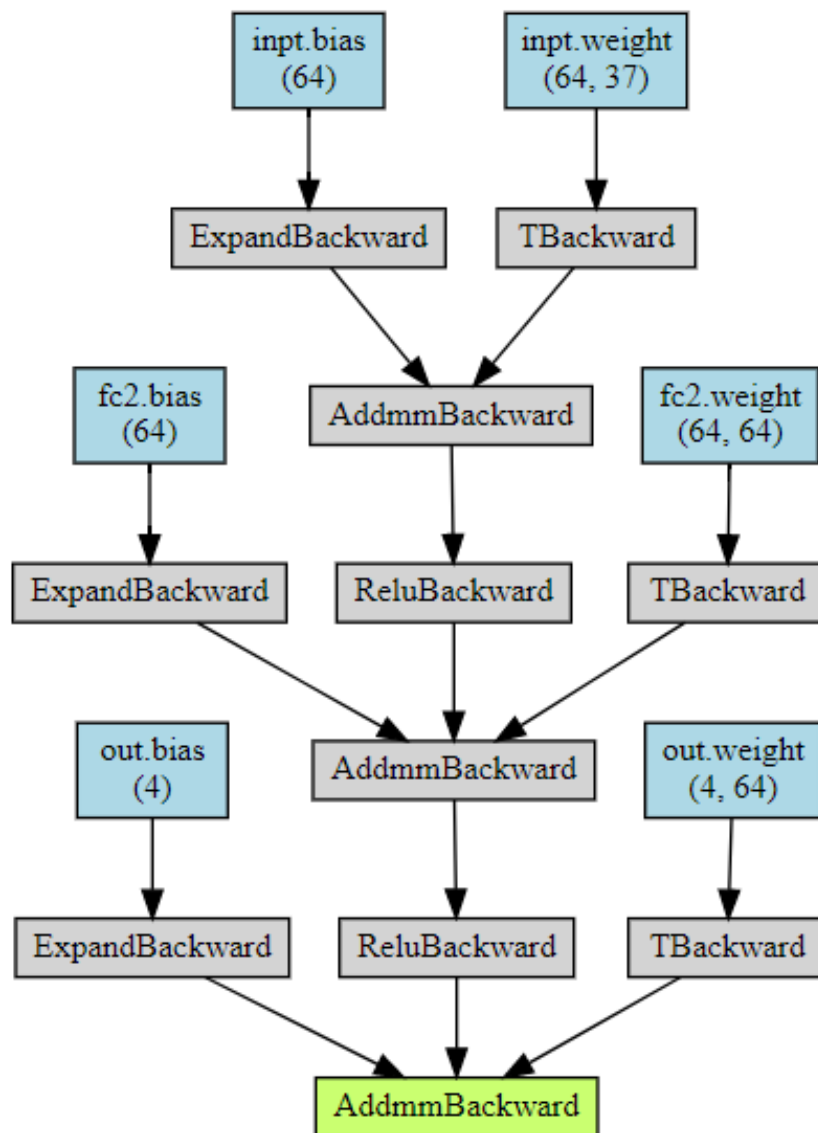
$$Q^*(s, a) = \max_a \mathbb{E}[r_0 + \gamma r_1 + \gamma^2 r_2 + \dots | s = s, a = a, \pi],$$

where Q^* is the maximum sum of expected rewards r_0 , discounted at each time step, t , by factor γ , based on taking action, a , given state observation, s , and the behavioural policy $\pi = P(a | s)$.

Hyperparameters

| | |
|------------------------|--|
| n_episodes = 2000 | # no. of episodes to train |
| eps_start = 1.0 | # epsilon upper limit (before any decay) |
| eps_end = 0.01 | # epsilon lower limit (minimum value) |
| eps_decay = 0.995 | # epsilon decay rate |
| 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 = 0.001 | # learning rate |
| UPDATE_EVERY = 4 | # how often to update the network |

NN Model Architecture



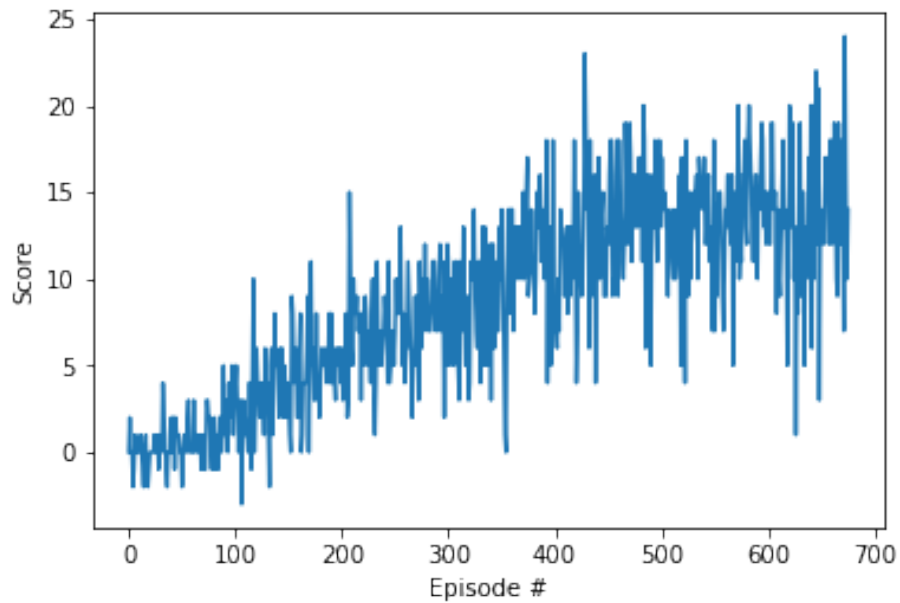
LINEARLAYER(IN = 37, OUT = 64) - > RELU - > LINEARLAYER(OUT = 64) - > RELU - > LINEARLAYER(OUT = 4)

N.B. SAME ARCHITECTURE HAS BEEN USED FOR BOTH NETWORKS: QTARGET AND QLOCAL

Plot of Rewards

| | |
|-------------|----------------------|
| Episode 100 | Average Score: 0.64 |
| Episode 200 | Average Score: 3.95 |
| Episode 300 | Average Score: 7.25 |
| Episode 400 | Average Score: 9.90 |
| Episode 500 | Average Score: 12.95 |
| Episode 600 | Average Score: 13.29 |
| Episode 676 | Average Score: 14.00 |

Environment solved in 576 episodes! Average Score: 14.00



Ideas for Future Work