

Project 1 Navigation

Let me describe my background first. Basically, I am a civil engineer from Bangladesh who doesn't know civil engineering much let alone RL. RL is like Hebrew, not only RL the whole CS things seem like Hebrew to me- top to bottom I don't understand anything. I did only a Python course in my undergrad. That's it. Yet I tried this Nanodegree. Why? Firstly, I got one month free access to Udacity which I cannot afford otherwise. Secondly, I got an admission in an Erasmus Joint Master Degree Program of Hydroinformatics in Europe. My program arranges a conference named 'Symhydro'. The accepted papers of that conference get published in collaboration with Springer Water in a journal namely 'Advances in Hydroinformatics'. I read one of the papers from there- 'Large Markov Decision Processes Based Management Strategy of Inland Waterways in Uncertain Context.' I read the paper and didn't understand anything. That's how I got interest in RL and took the resolution to decode RL, though in reality I didn't understand many things while reading and doing so.

Thanks to the solution notebooks by Udacity and also the alumni of this Nanodegree. The solution codes I use here are managed through visiting their GitHub Repositories. Now, by miracle if this project get passed and I get the certificate, I will be the happiest man in the world.

The solution code uses:

- Vanilla Deep Q-Network;
- Double Deep Q-Network to avoid oscillations caused by overestimated Q-values;
- Experience Replay in order to keep training the Q-Network with past experiences;
- Epsilon Greedy with geometric decay of 0.995, in order to balance exploration at the beginning (epsilon=1) versus exploitation at the end (epsilon=0.01);
- Adam optimizer with learning rate of $5e-4$;
- BATCH_SIZE=64 (the number of experience tuples per training iteration);
- GAMMA=0.99 (the Q-Network is aware of the intermediate future, but not the far future);
- A deep neural network to represent complex continuous states.

The deep neural network to represent complex continuous states has:

- A linear fully-connected layer of dimensions state_size=37 and fc1_units=64;
- A linear fully-connected layer of dimensions fc1_units=64 and fc2_units=64;
- A linear fully-connected layer of dimensions fc2_units=64 and action_size=4.

Plot of Rewards

The Q-Network was trained for 706 episodes. In each episode, the agent is trained from the beginning to the end of the simulation. Some episodes are larger and some episodes are shorter, depending when the ending condition of each episode appears. Each episode has many iterations. In each iteration, the Q-Network is trained with BATCH_SIZE=64 experience tuples (SARS).

Episode 100 Average Score: 0.96

Episode 200 Average Score: 4.39

Episode 300 Average Score: 7.77

Episode 400 Average Score: 10.41

Episode 500 Average Score: 12.49

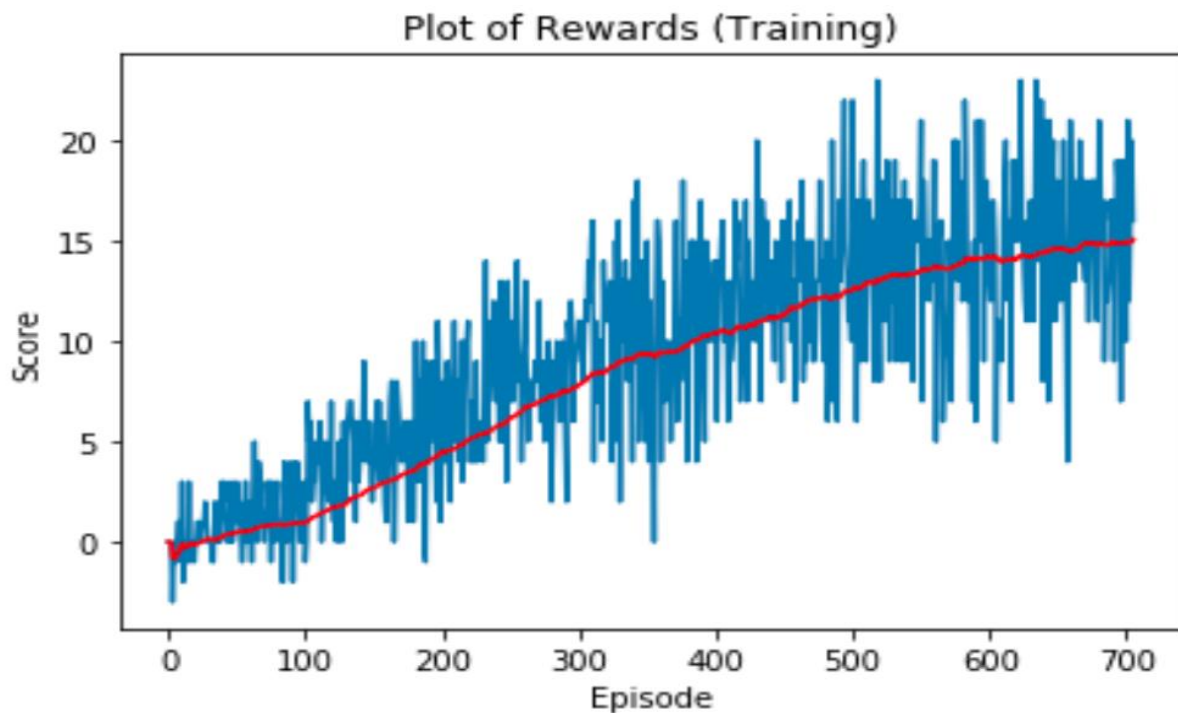
Episode 600 Average Score: 14.25

Episode 700 Average Score: 14.91

Episode 706 Average Score: 15.07

Environment solved in 606 episodes! Average Score: 15.07

The rubric asks to obtain an average score of 13 for 100 episodes. I increased that value to 15. As a result, the Q-Network achieved an average score greater than 15 in 706 episodes of training. The best model was saved. In the graph, the blue lines connect the scores in each episode. Whereas the red lines connect the average scores in each episode.



After training, the saved model was loaded and tested for 20 episodes. Here are the results of such testing. You can see that on average, the scores are greater than 15. In the graph, the blue lines connect the scores in each episode.



Ideas about Future Works:

1. More experiments can be done to increase the performance of agent by applying different extensions of DQN:

- Double DQN (DDQN)
- Prioritized experience replay
- Dueling DQN A3C
- Distributional DQN
- Noisy DQN

2. Personally, I want to apply DeepReinforcement Learning in the field of Hydroinformatics and publish my researches in Springer Water. I want to be the best RL Research Assistant my professor will ever have.