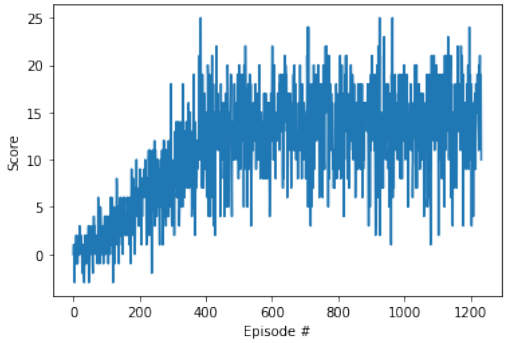
I. Double DQN

a. Learning Algorithm

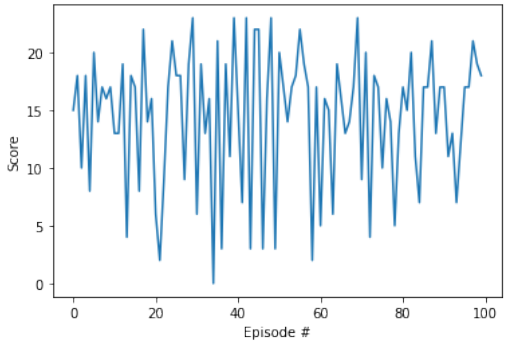
i. The model uses 3 fully-connected layers as hidden layers, and each layer has 128 units.
ii. Discount rate (γ) = 0.99
iii. Stochastic randomness (ϵ): initial = 1.0; final = 0.01; decay rate = 0.995
iv. Learning rate = 0.0005
v. Training max episode = 2500
vi. Early Stopping tolerance = 50 episodes

b. Result

i. Solved in 1067 episodes with reward condition over 14.5.
ii. Training plot



iii. Testing performance: 14.47 average reward (over 100 episodes)
iv. Testing plot



II. Duel DQN

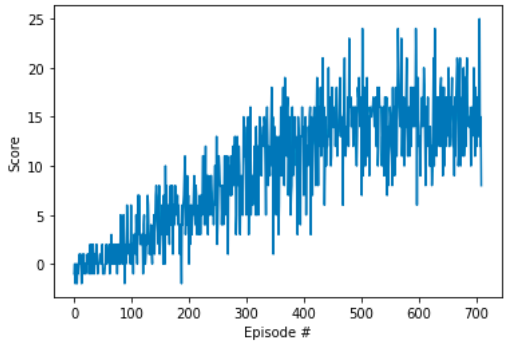
a. Learning Algorithm

i. The model uses 2 fully-connected 128-unit layers as feature extraction layers, an 128-unit layer as value layer, and an 128-unit layer as advantage layer.
ii. Discount rate (γ) = 0.99
iii. Stochastic randomness (ϵ): initial = 1.0; final = 0.01; decay rate = 0.995

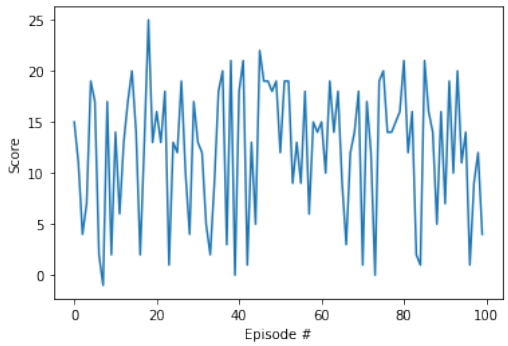
iv. Learning rate = 0.0005
v. Training max episode = 2500
vi. Early Stopping tolerance = 50 episodes

b. Result

i. Solved in 553 episodes with reward condition over 14.75.
ii. Training plot



iii. Testing performance: 12.36 average reward (over 100 episodes)
iv. Testing plot



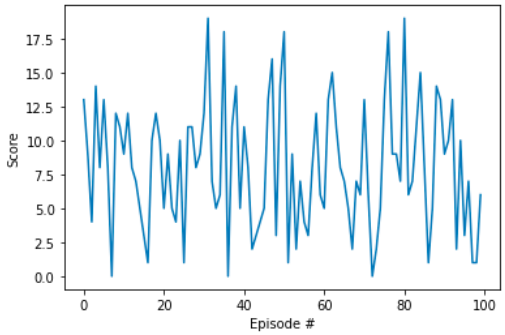
III. Prioritized Experience Replay DQN

a. Learning Algorithm

i. The model is the same as the Double DQN model.
ii. Discount rate (γ) = 0.99
iii. Stochastic randomness (ϵ): initial = 1.0; final = 0.01; decay rate = 0.995
iv. Learning rate = 0.0001
v. Compensate weight (β): initial = 0.6; final = 1.0; time to saturate = 2000
vi. Training max episode = 20000
vii. Early Stopping tolerance = 50 episodes

b. Result

i. Never solved…
ii. Testing performance: 8.1 average reward (over 100 episodes)
iii. Testing plot



Ideas for Future Work

I still don’t understand why the DDQN I trained couldn’t perform well in the evaluation phase. With environment reset every training episode, it doesn’t make sense the agent would prone to overfitting. A further investigation would be a good start for future work.

On the other hand, I assume the hyperparameters of using Prioritized Experience Replay might lead to the poor performance of the agent. Therefore, exploration of different sets of hyperparameters could also be a potential topic.

Last but not the least, besides rainbow DQN, N-step training also seems very fascinating. I think adding N-step training on top of the rainbow DQN would yield pretty decent results.