P1 Navigation

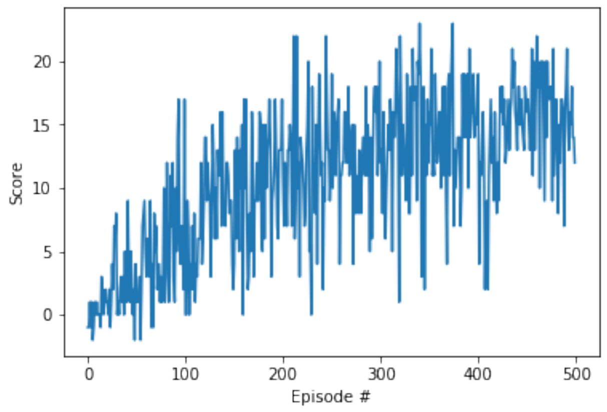
Learning Algorithm

The Agent makes use of a Double DQN learning algorithm with a Replay Buffer and an epsilon-greedy strategy. The neural-network model has two hidden layers with 64 nodes each, activated with a ReLU function.

The Replay Buffer size used is 10,000 and it is sampled during our learning step once there are at least 64 memories within it. Our Agent learns every 4 steps, discounting the future with a Gamma=0.99, optimizing our Local Model with an Adam Optimizer using an LR=5e-4, and soft updating the Double DQN target model with a Tau=1e-3. We found this Gamma value to work well as a balance between future and present rewards given the number of steps we expect the agent to take in solving this environment. We also found this intermittent Double DQN learning strategy to stabilize our Agent.

Plot of Rewards

Our per episode rewards are expectantly erratic but clearly trending upwards, stabilizing with an average score above 13. It is clear the agent is able to achieve a score greater than 13 within 100 episodes, and able to achieve a consistent average score greater than 13 sometime around episode 300 ~ 350. The agent continues to improve beyond this point, but not much so. It is entirely possible if we continue to train this agent that its performance will rise until such point where it begins to overfit.



Future Work

Future work would include implementing other RL agent types, such as Rainbow DQN and Actor-Critic.

We would also look into solving this environment from pixel data, rather than from a codified environment.