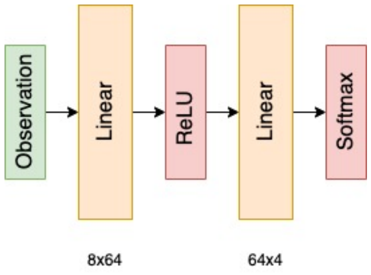
Applied Deep Learning: Assignment 3 - Deep Reinforcement Learning

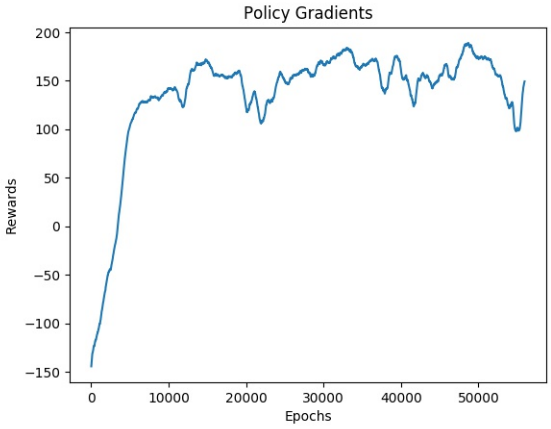
b07902047 羅啓帆

May 2020

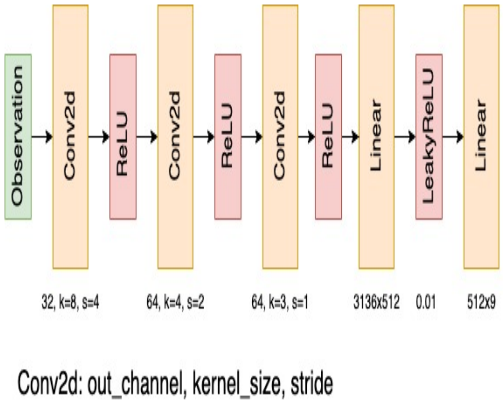
Q1: Models

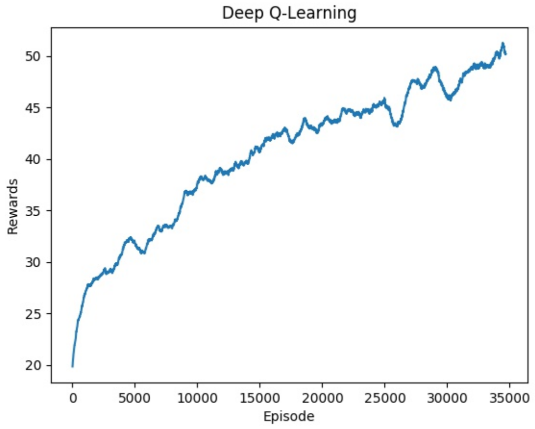
Policy Gradient



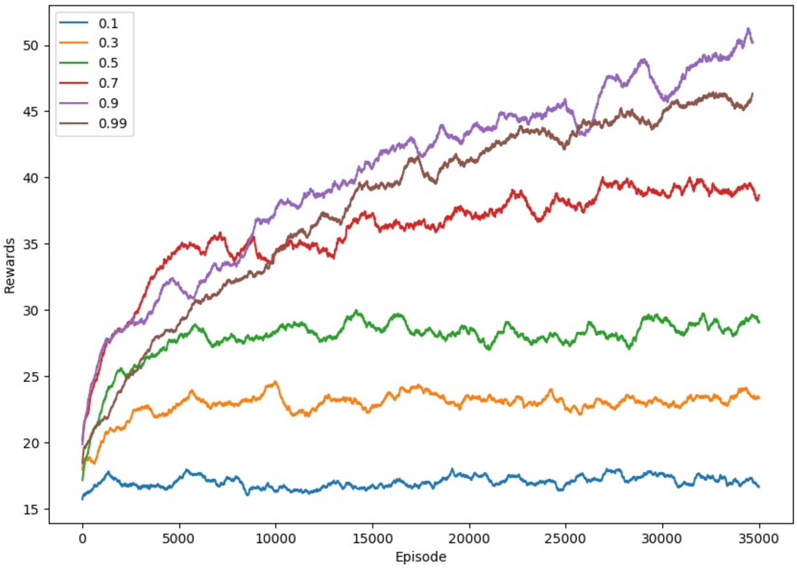


DQN





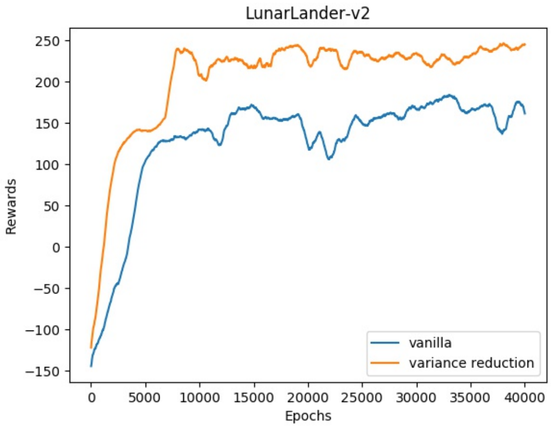
Q2: Hyperparameters of DQN

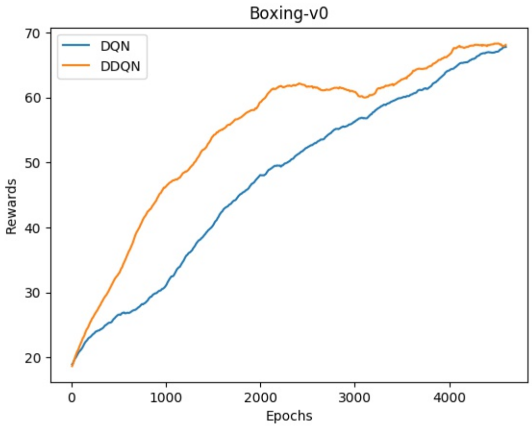


I choose GAMMA as the hyperparameter and play on MsPacman-v0. We can observe that if we discount the weight too servere, the model will not perform well as small GAMMA makes the model short-sighted. We can also observe that a little discount might work better than nearly no discount (0.9 vs 0.99).

Q3: Impromendts of Policy Gradient / DQN

Variance Reduction & Double DQN





Variance reduction shift the rewards by minus a ”baseline” b (I used average reward) from the discounted reward. This might be helpful because if all rewards are positive, some bad actions might be chosen first and because rewards are positive, it’s probablity being chosen will raise, further preventing the model from choosing good actions (which brings more rewards). By shifting the reward, we can decrease the probablity of ”not so good” actions and solve the problem mentioned above. From the left figure above, we can observe that variance reduction converges fatser and performs better on the environment LunarLander-v2.

Double DQN reduced the problem that nature DQN tends to overestimate Q values by changing

$$\mathcal{L}(w) = \mathbb{E}_{s,a,r,s'\sim D}[(r + \gamma \max_{a'} \hat{Q}(s', a', w^-) - Q(s,a,w))^2]$$

to

$$\mathcal{L}(w) = \mathbb{E}_{s,a,r,s'\sim D}[(r + \gamma \hat{Q}(s', \arg\max_{a'} Q(s', a', w), w^-) - Q(s,a,w))^2]$$

In other words, we choose the next state action using online network(current) instead of target network(older). This may be helpful because it prevent the overestimation of target network, which is older and more biased. From the right figure above, we can observe that DDQN converges faster and performs better most of the time on the environment Boxing-v0.