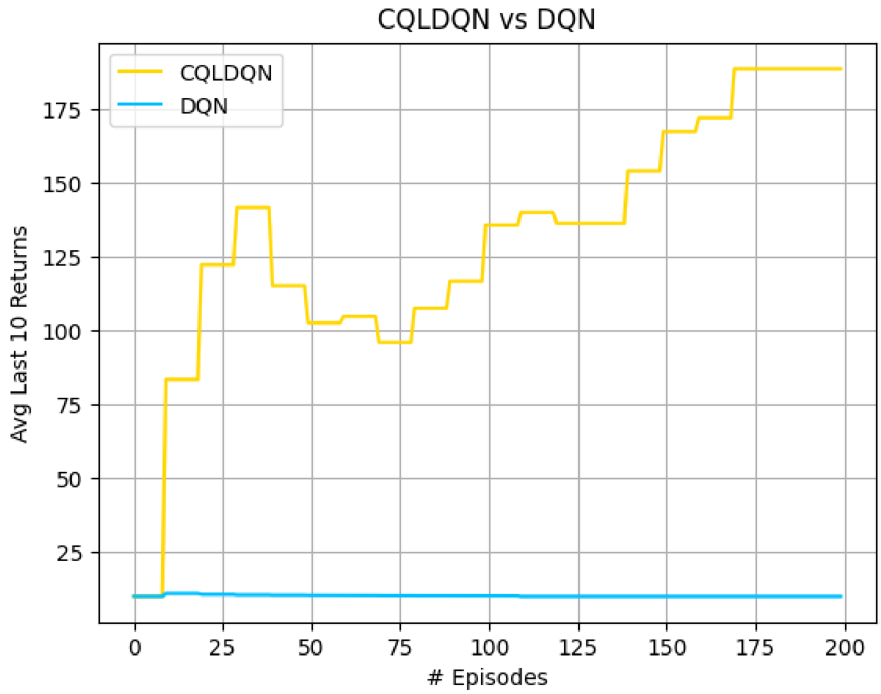

From the graph, we see that Conservative Q-learning (CQL) algorithm for offline Reinforcement Learning greatly outperforms offline Deep Q-Learning (DQN). Both approaches use experience replay and two network (Q and Target network) for mitigating divergence. In both approaches – the Q and the target network are deep neural networks, where every c steps, the target network is updated with the weights of the Q network.

When we use only DQN, we use the current Q estimate, and the Q estimate of the next step (and reward of executing current action a) to perform the policy update:

$$Q^{\pi} = argmin_Q E_{(s,a,r,s')\sim D}\left[\left(r + \gamma E_{a'\sim\pi(a'|s')}[Q(s',a')] - Q(s,a)\right)^2\right]$$

Some predicted Q(s,a) values are underestimated and some are overestimated. The above equation results in selecting more overestimated values due to the greedy policy improvement rule. So, instead of selecting good actions – some bad actions with overestimated Q values are selected (Because using argmin, we are trying to reduce the temporal difference between current Q estimate, and target).

A penalty term in CQL is introduced to mitigate this Q-value overestimation. Using sufficiently large penalty term – we can ensure that the Estimated Q-values are lower bounded by true underlying Q values. This ensures the reduction of temporal difference avoids overestimation. This results in a significant avoidance of overestimated Q-values, and due to the nature of greedy policy – the underestimated values are mostly avoided. This is the reason why the CQL performance is significantly better than DQN.