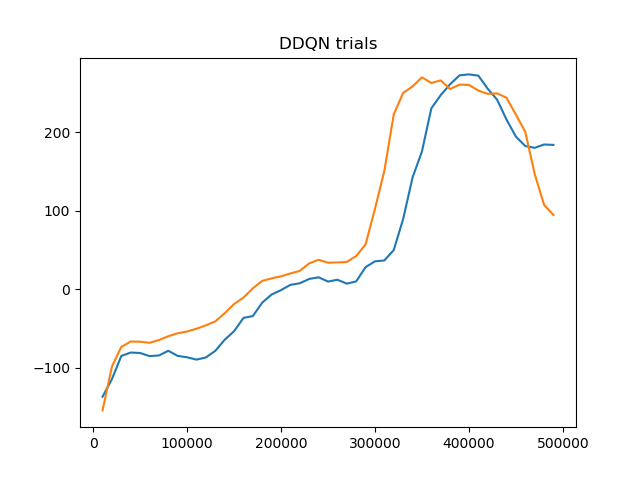
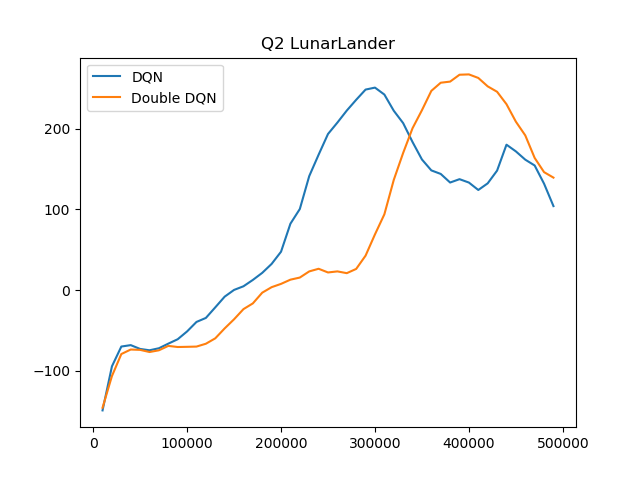


graph of train\_avg\_return

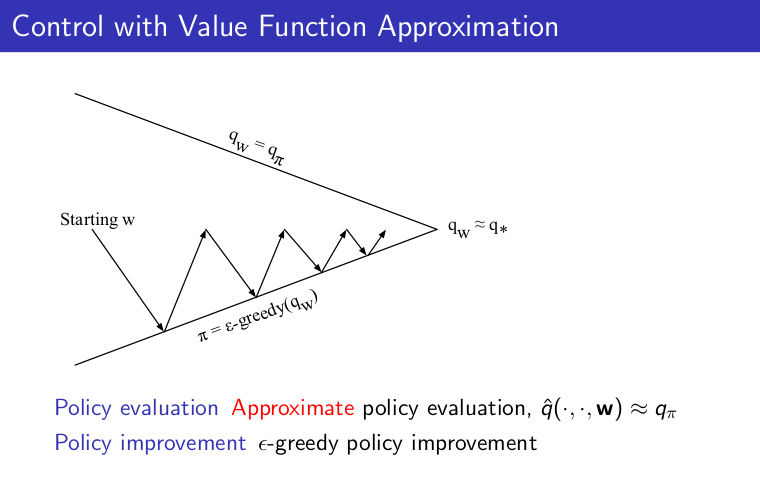
DQN trails with random seed on lander environment (observe high variance between graphs)

graph of train\_avg\_return

DDQN trails with random seed on lander environment (observe low variance between graphs)

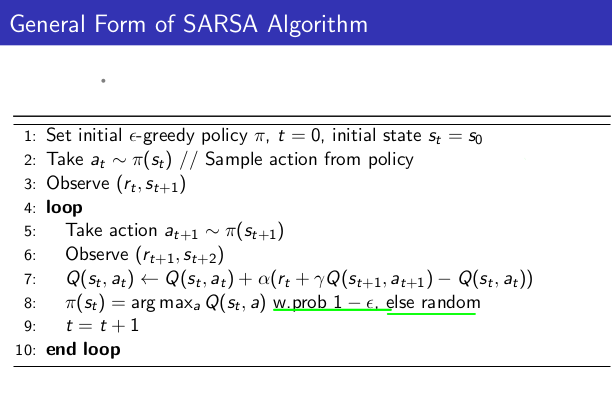
 graph of train\_avg\_return compairing DQN VS DDQN

**Model free Policy Iteration to DQN**



Example of a very model free Policy-Iteration Algorithm:

SARSA ALGORITHM:



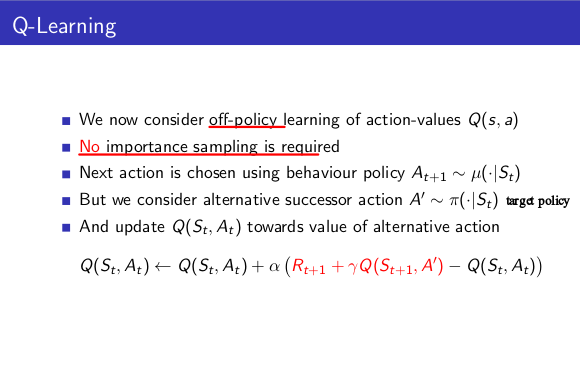
SARSA is ON policy learning , estimates the value of current behavior policy and then updates the policy trying to estimate.

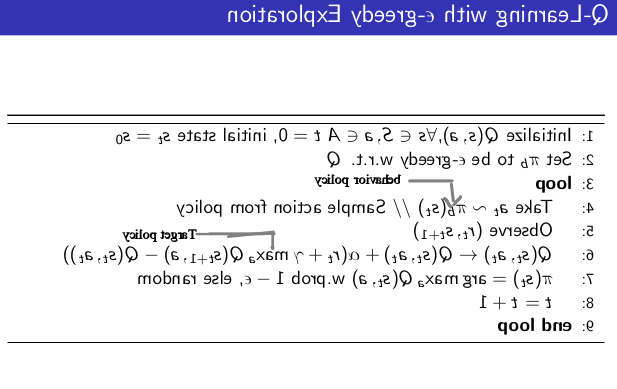
Other Option is of OFF-POLICY LEARNING where we can directly estimate the value

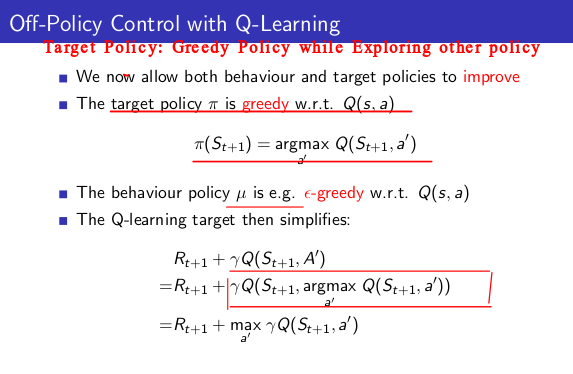
of optimal\_policy while acting another behavior policy pi\_b which can also explore all possible states .

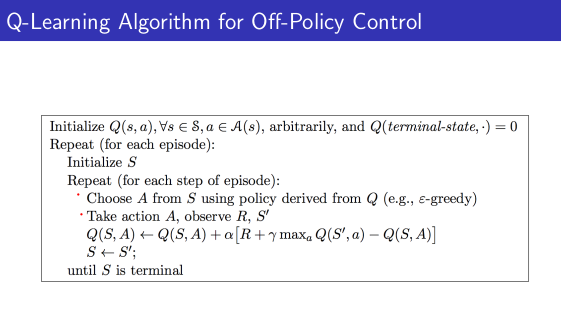
Can we omit the Policy improvement step? YES

**Q-LEARNING**



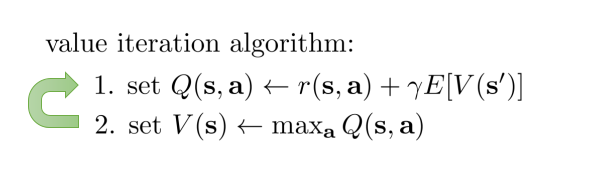


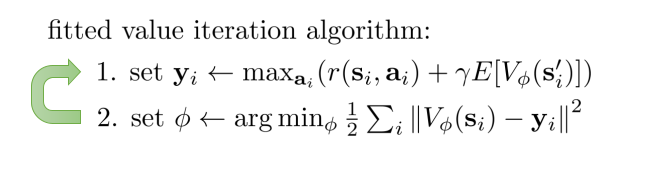


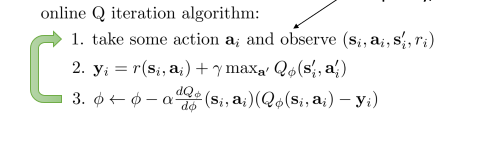


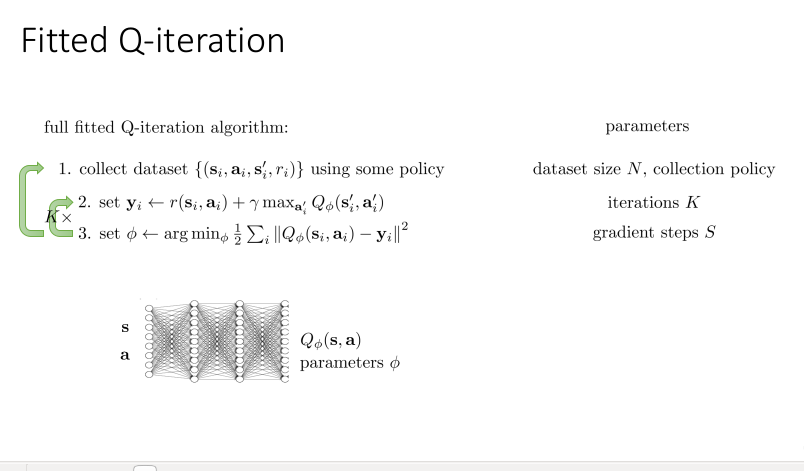
Essence : Updating the Q values in the direction of best possible next Q values

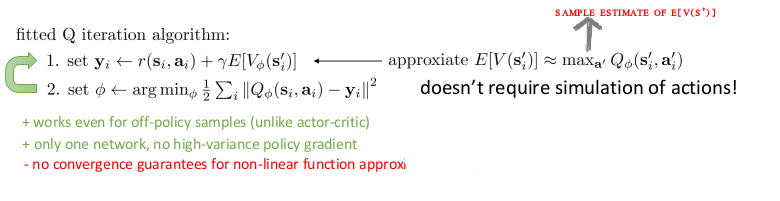
Value Iteration->Fitted Value Iteration ->Fitted Q Value Iteration( for unkown dynamics)





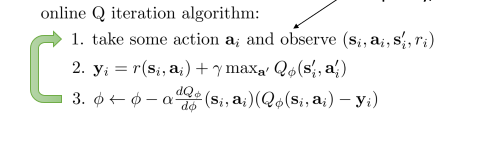






Fitted Q learning is off policy learning , where data can be come from any policy and the final policy that we are trying to learn is arg max policy .

**Q-learning** which is an online policy :is a special case of Fitted-Q iteration where K=1



This Q learning algorithm does not converge in practice , there is two problem with this

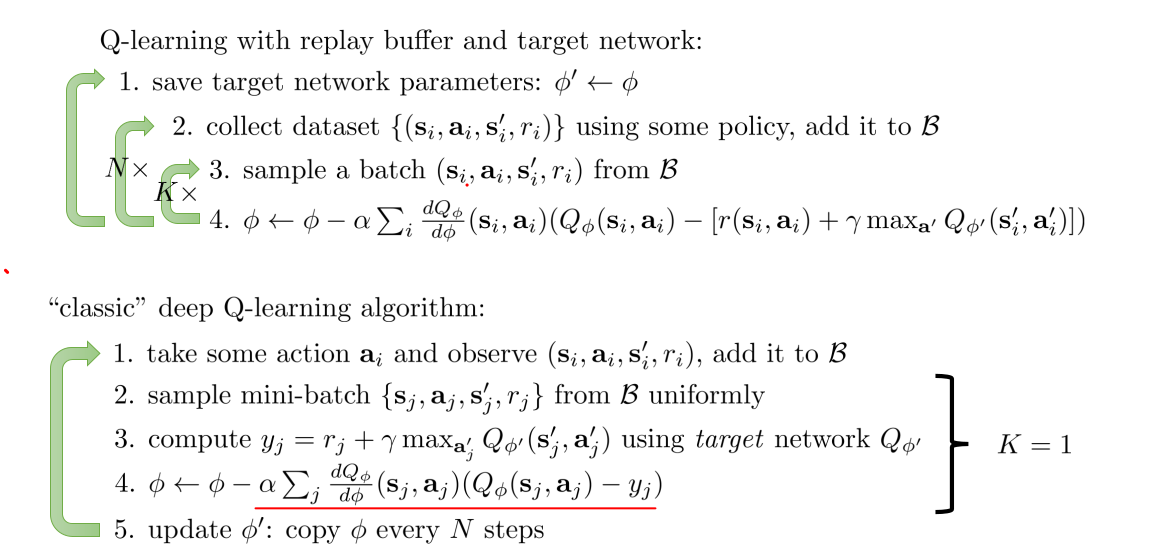
a. Correlated data

b.non stationary target value

These are fixed by replay buffers, using mini – batch gradient descent and using target networks.

DQN algorithm is improved , stable algorithm for fitted Q learning algorithm consist of above changes:

DEEP Q- LEARNING ALGORITHM (DQN):



Double DEEP Q-LEARNING ALGORITHM:

In some stochastic environments the well-known reinforcement learning algorithm Q-learning performs very poorly. This poor performance is caused by large overestimations of action values. These overestimations result from a positive bias that is introduced because Q-learning uses the maximum action value as an approximation for the maximum expected action value.introducing an alternative way to approximate the maximum expected value for any set of random variables. The obtained double estimator method is shown to sometimes underestimate rather than overestimate the maximum expected value. We apply the double estimator to Q-learning to construct Double Q-learning, a new off-policy reinforcement learning algorithm.New algorithm converges to the optimal policy and that it performs well in some settings in which Q-learning performs poorly due to its overestimation.

To modify the DQN one way to use the current and target networks:

