In the previous study, we talked about function approximation that optimize the value of a parameterized vector, w, that predicts the value function which is then compared with a target value from the output of incremental mean methods, Monte Carlo or TD Learning. Of course, the target is partially fixed, to avoid hitting a moving target. The output of the value function from the approximation was then used to determine a policy, either greedily or E-greedily. That was the extent of Deep Q Learning.

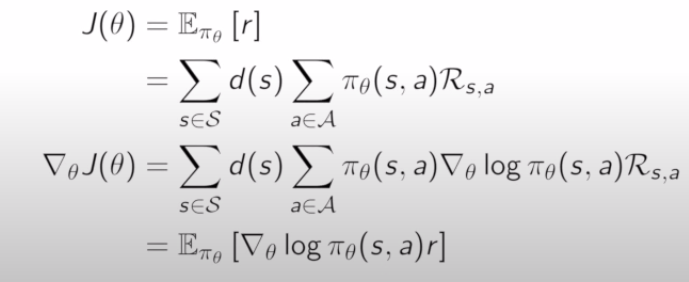
Now we come to talk about Policy based learning. Direct improvement of policy rather than the sequence of value functions and control. By parameterize-sing the policy distribution itself i.e. a the function approximator will be used to approximate the policy, preferred stochastic policy over deterministic because of state aliasing and equilibrium, and the gradient will be followed up towards the maximum, hopefully global.

Π θ (s,a) = g (f(s,a), θ)

Some of the advantages of the policy gradient methods include better convergence, effectiveness in high dimensional or continuous action spaces and even on stochastic policies. Where as the disadvantages includes converging to a local maximum and high variance.

Apart from different gradient based methods like gradient descent, conjugate gradient or quasi-newton, we will be focusing on gradient descent for policy optimization of the objective function.

Policy gradient involves gradient ascent, to maximize the value of returned reward. The optimization of the policy takes based on the concept of the log likelihood or soft max combination with the gradient being calculated on the basis of score function. A score function calculates the difference between the feature (state , action pair) and the expected feature based on the policy given meaning it gives the intuition to the agent about how much more than usual am I doing something?

  
Figure 1: Based on One Step MDP, no sequences

Clearly here the final equation tell us that if we are expecting more reward, we should move in the direction of gradient pointed out by the score function.