**POLICY GRADIENT METHODS**

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.

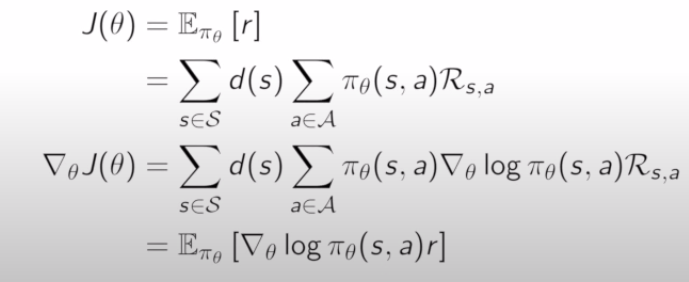
Now we come to talk about Policy based learning. Direct improvement of policy rather than the sequence of value functions and control. By parameterize-sing the policy distribution itself i.e. a the function approximator will be used to approximate the policy, preferred stochastic policy over deterministic 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. The value of “r” indicates the value to be an immediate reward, also it is a one step MDP, so upon replacing the value by action value approximate, we can calculate the gradient of policy “Policy Gradient Theorem”.

  
Figure 2: Policy Gradient Theorem

With the introduction of the action value function, sampling of the states occurs over a number of sequences and the concept is called the Monte Carlo Policy gradient (slow and high variance). The use of the return or reward from a sequence in gradient estimators gives high variance so,

Actor Critic method: Instead of using the returned as the action value, we now use a critic to estimate the value of the action value. So an actor critic method focuses on the two parameters to update:

* Critic : Updates action value function parameters,w
* Actor : Updates the policy parameters, u, in the direction suggested by the critic





Here the critic is actually solving the problem of policy evaluation which was previously solves using the Monte Carlo and TD. The critic is updated normally by the TD method where as the actor is updated by the policy gradient hence calculated. **Till now intuitively we are picking up a policy, which is then evaluated using the critic and instead of moving towards more greedy policy we are moving in the direction pointed by the gradient towards better policy.**

Now, the concept of actor-critic introduces bias as the policy is approximated using a linear approximate (basically the problem of local optimum). This is solved using **advantage function** in place of the action value function. Furthermore the action and state value function can both be estimated using the TD learning approaches via TD error concept.

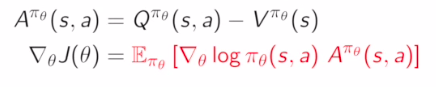
  
Figure 3: Gradient calculated on the basis of advantage function

  
Figure 4: Gradient calculated on the basis of TD error

**Takeaways**:

* Evaluate the policy itself than going through the concept of the value function
* Find out gradients to direct the agent towards the optimum
* On calculating gradients prefer the critic method solved using the TD error concept