CS6700 : Reinforcement Learning
Written Assignment #2

Deadline: ?

- This is an individual assignment. Collaborations and discussions are strictly prohibited.
- Be precise with your explanations. Unnecessary verbosity will be penalized.
- Check the Moodle discussion forums regularly for updates regarding the assignment.
- **Please start early.**

AUTHOR : Rajan kumar soni.
ROLL NUMBER : CS18S038

1. (3 points) Consider a bandit problem in which the policy parameters are mean μ and variance σ of normal distribution according to which actions are selected. Policy is defined as $\pi(a;\mu,\sigma^2) = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{(a-\mu)^2}{2\sigma^2}}$. Derive the parameter update conditions according to the REINFORCE procedure (assume baseline is zero).

Solution: Using Reinforce update rule $\Delta\theta_t = \alpha_t \cdot r_t \cdot \frac{\nabla\pi(\Theta,a_t)}{\pi(\Theta,a_t)}$
We have two dimensional parameter.Here μ and $\sigma$are two parameters. Differentiating policy by parameters one by one. Writing separate update equation for μ and σ
$\Delta\mu = \alpha_t \cdot r_t \cdot \frac{(a-\mu)}{\sigma^2}$
$\Delta\sigma = \alpha_t \cdot r_t \cdot \frac{(a-\mu)^2-\sigma}{\sigma^3}$
$\mu_{t+1} = \mu_t + \alpha r_t \left(\frac{(a-\mu_t)}{\sigma^2}\right)$
$\sigma_{t+1} = \sigma_t + \alpha r_t \left(\frac{-1}{\sigma} + \frac{(e-\mu_0)^2}{\sigma^3}\right)$

2. (6 points) Let us consider the effect of approximation on policy search and value function based methods. Suppose that a policy gradient method uses a class of policies that do not contain the optimal policy; and a value function based method uses a function approximator that can represent the values of the policies of this class, but not that of the optimal policy.

   (a) (2 points) Why would you consider the policy gradient approach to be better than the value function based approach?

   Solution: Since we have approximation, we know that value function based methods might not converge to best policy availabe in that class but policy gradient to converge on of the local policy.

(b) (2 points) Under what circumstances would the value function based approach be better than the policy gradient approach?

Solution:

- When the value function is not that much complex.
- If we need less parameter to represent value function.
- If we only need linear parameterization because there only it is guaranteed to converge.
- Where finding hand-crafted feature is difficult because it is very easy with DQN.
- If the optimal policy is deterministic then it is good to go with value function

(c) (2 points) Is there some circumstance under which either of the method can find the optimal policy?

Solution: If we talk about policy gradient, here we are directly searching for optimal policy although we might stuck in local optima. We know that given class of policy does not contain optimal policy so it is impossible to reach the optimal policy in this case.
As we know that there always exist deterministic optimal policy. In case of value function based methods we use value function to find the policy. If the relative ordering of the state-action value or q value is same so we can still find optimal policy. Although we would not be sure it optimal policy or not because we don't have optimal value.
For example- for a given state actions a1 , a2, a3 have optimal q1, q2, q3 values where $q1 > q2$ and $q1 > q3$ then a1 will be optimal action for optimal policy but if we have $q1^{'}$, $q2^{'}$, $q3^{'}$ but still they satisfy $q1^{'>q2^{'}}$, $q1^{'>q3^{'}}$.

3. (4 points) Answer the following questions with respect to the DQN algorithm:

- (2 points) When using one-step TD backup, the TD target is $R_{t+1} + \gamma V(S_{t+1}, \theta)$ and the update to the neural network parameter is as follows:

$$\Delta\theta = \alpha(R_{t+1} + \gamma V(S_{t+1}, \theta) - V(S_t, \theta))\nabla_\theta V(S_t, \theta) \tag{1}$$

Is the update correct ? Is any term missing ? Justify your answer

Solution: It depends how we are treating target constant or variable. But we generally treat target constant so it is correct.If we will do gradient descent we will get

$$\Delta\theta = \alpha(R_{t+1} + \gamma V(S_{t+1}, \theta) - V(S_t, \theta))\nabla_\theta V(S_t, \theta) \tag{2}$$

- (2 points) Describe the two ways discussed in class to update the parameters of target network. Which one is better and why?

Solution: Two ways to update the target networks are Full update and Soft update.
Full update:- Here parameters of the target network is fully replaced by the new vales of the current network.
Soft update:-Here parameters of the target network is combination of the current network parameter and target network parameter. Here frequency of updating is high.

Soft one is better. If we see loss curve of DQN we see spikes there it is because we suddenly change the parameter of the target network which cause increase in loss. It is like every change in target networks create new supervised learning model and it start reducing loss.It seems like model was training towards some target and suddenly target got change. And we know that no target network represent true values, So its better to move slowly and don't blindly follow new target or old target
$\theta^- = \theta \times \tau + \theta^- \times (1 - \tau)$

4. (4 points) Experience replay is vital for stable training of DQN.

   (a) (2 points) What is the role of the experience replay in DQN?

Solution:

- It helps in using samples efficiently.
- In case of continuous space updates are strongly co-related which can lead to problem in convergence. So they put current sample in replay memory and then picks randomly to update the network.
- It also eliminates biased sampling(skewed sampling) of input space introduced by online update.

   (b) (2 points) Consequent works in literature sample transitions from the experience replay, in proportion to the TD-error. Hence, instead of sampling transitions using a uniform-random strategy, higher TD-error transitions are sampled at a higher frequency. Why would such a modification help?

Solution: Large TD error tells that state value is far from approx optimal v value . It shows that network parameter is not trained in such way that it can reduce the error for that state.it might be rare state but important to learn the good approximate for that state. Transitions may be more or less surprising, redundant, or task-relevant. Some transitions may not be immediately useful to the agent, but might become so when the agent competence increases. In case of uniform sampling it might be washed out when new transition come but in case of Prioritized Experience Replay we can save it for long time and use for training.

5. (3 points) We discussed two different motivations for actor-critic algorithms: the original motivation was as an extension of reinforcement comparison, and the modern motivation is as a variance reduction mechanism for policy gradient algorithms. Why is the original version of actor-critic not a policy gradient method?

Solution: In extension reinforcement comparison we use v function as critique while in variance reduction mechanism for policy gradient algorithm we use q function as critique.there is relation between parameterisation of actor and critique. One controls the other. Here gradient of both the parameter is related to each other consistently. But it is not the case in original motivation of actor critique where

the actor-critique algorithm generalises from reinforce and the baseline of the critique is described by a value function based method. The actor is evaluated by TD ERROR. Value function of the critique gets updates regardless of the chosen action on that state.Therefore updates step of these parameters does not follow policy gradient. Where actor and critique are updated by same amount but at different time instant.

6. (4 points) This question requires you to do some additional reading. Dietterich specifies certain conditions for safe-state abstraction for the MaxQ framework. I had mentioned in class that even if we do not use the MaxQ value function decomposition, the hierarchy provided is still useful. So, which of the safe-state abstraction conditions are still necessary when we do not use value function decomposition?

Solution:

Leaf irrelevance
Max Node Irrelevance
result irrelevance

Termination
Shielding

Max node irrelevance and leaf irrelevance, They are the necessary. They are based on one step transition probability and reward formulation. If only these conditions are holding we can have safe-state abstraction.

7. (3 points) Consider the problem of solving continuous control tasks using Deep Reinforcement Learning.
   (a) (2 points) Why can simple discretization of the action space not be used to solve the problem? In which exact step of the DQN training algorithm is there a problem and why?

   Solution: simple discritization wold not work because each action has its own effect. It migt be possible that two actions are close in action space but effects are very different vice versa. If we discretize the continuous action spaces, then the memory required to hold the actions, will shoot up.i.e. for an action space with M dimensions, discretizing K atomic actions per dimension leads to M^Kcombinations of joint atomic actions. The problem is when we trying to pick the argmax action.

   (b) (1 point) How is exploration ensured in the DDPG algorithm?

   Solution: Here we introduce noise process like ϵ-greedy to ensure exploration
   $a_t = \mu\left(s_t \mid \theta^\mu\right) + \mathcal{N}_t$
   In original paper author uses the Ornstein-Uhlenbeck Process generates noise that is correlated with the previous noise, as to prevent the noise from canceling out or "freezing" the overall dynamics

8. (3 points) Option discovery has entailed using heuristics, the most popular of which is to identify bottlenecks. Justify why bottlenecks are useful sub-goals. Describe scenarios in which a such a heuristic could fail.

Solution: Bottleneck divides the MDP in different components.Bottleneck states are also called access states.Bottlenecks are those states which act as bridge between different parts of MDP. If we have to go other part we must have to cross it. Then its is better to learn the way to bridge first.Bottlenecks are simply graph cuts where are graph is minimally connected.
It can fail in dynamic environment where bottleneck keep changing.