AI 3000 / CS5500 : Reinforcement Learning
Assignment № 4

Course Instructor : Easwar Subramanian 08/11/2024

Problem 1 : Deep Deterministic Policy Gradients

The problem involves coding DDPG algorithm and testing in **MountainCarContinuous-v0** and **LunarLanderContinuous-v2** environments. Assignment 4 had descriptions of similar environments where the action space was discrete. On the contrary, the above environments have continuous action spaces and hence these are continuous control problems. Recall that DDPG algorithm is mainly used for solving continuous control problems.

(a) Develop a small code snippet to load the corresponding Gym environment(s) and print out the respective state and action space. Develop a random agent to understand the reward function of the environment. Record your observations. (3 Points)

(b) Implement the DDPG algorithm to solve the tasks envisioned by the two environments. Provide corresponding learning graphs in your Jupyter notebooks. (12 Points)

(c) The standard DDPG implementation involves using Ornstein Ulhenbeck (OU) noise for exploration. For this sub question study the implication of using Gaussian noise for exploration and report (with reasons) if solving the task takes similar number of episodes. (5 Points)

Problem 2 : Bandit Algorithms

Consider a multi-arm bandit problem consisting of K arms. Every time an arm is pulled, the agent receives a reward r_a^t such that $r_a^t \sim \mathcal{N}(\mu_a, 1)$ where $a \in \{1, 2, \cdots, K\}$ is the arm that was pulled at time t.

(a) Develop a bandit environment where the reward distribution of each bandit is normally distributed around a given mean (μ) and unit variance. One can use choose appropriate mean reward values for each bandit. The mean reward values for each arm could be well spaced out so that the best arm can be distinctly identified. (2 Points)

(b) Implement greedy, ϵ-greedy, ϵ-greedy with decay and UCB algorithm to solve the above bandit problem in N rounds. For the ϵ-greedy algorithm, consider $\epsilon = 0.01$ and $\epsilon = 0.1$. (Recall that $\epsilon = 0$ is the greedy strategy) (8 Points)

(c) For the case of $K = 10$ and number of rounds $N = 1000$, plot the average reward obtained using each of the above strategies (in same graph). The x-axis of the plot is the number of rounds (1 to 1000) and y-axis is the average reward obtained in that round. The average reward is computed by running a bandit algorithm multiple times (say 100 times).

(2 Points)

[For example, let's say we run a bandit algorithm three times. This means that, each time, the agent gets 1000 rounds to pull any of the 10 arms. Corresponding to each time, the agent gets a reward at each round. The average reward is the average values of the reward obtained in each round over multiple times.The bandit environment should be reset every time.]

(d) Plot a histogram consisting of average number of times each of the 10 arms is pulled for each of the algorithm in sub-question (b) (2 Points)

(e) Plot the average regret as a function of time for each algorithm (2 Points)

(f) From the above plots, reason out which of the above bandit algorithm is better (2 Points)

(g) Repeat the above analysis for the scenario when the mean rewards of the second best arm and the best arm are close (2 Points)