Machine Learning Reinforcement Learning Assignment

The focus of thsiassignment is to explore the use of modules in reinforcement learning. You ahould create a driving environment that consists of a wide sidewalk. Your environment should have:

1. Many different litter objects to pick up, each of which needs a specialized module.
2. Many different litter objects to pick up, each of which needs a specialized module.
3. A module with reward for getting to the end of the sidewalk.

You should solve this problem by giving reinforcement modules that learn by experience.

Each module that you choose should use a version of Q-Learning. SARSA is a popular option where you follow the best policy, with a schedule of picking a random alternative.

A. First train up your modules individually and demonstrate their performance.

B. Next construct a protocol to allow them to work together when they are simultaneously active. The weighted average of Q values is one alternative. The highest Q value is another.

C. Next see if you can construct a protocol that will allow the mix of active modules to vary according to the environmental demand.

See helpful RL note on next page ...

Some of you have asked for a Reinforcement Text. This is a fairly good one.

From **"Statistical Reinforcement Learning" by Sugiyama, CRC Press 2015**

Then, in the same way as $V^\pi(\boldsymbol{s})$, $Q^\pi(\boldsymbol{s}, a)$ can be expressed by recursion as

$$Q^\pi(\boldsymbol{s}, a) = r(\boldsymbol{s}, a) + \gamma \mathbb{E}_{\pi(a'|\boldsymbol{s}')p(\boldsymbol{s}'|\boldsymbol{s},a)} \Big[Q^\pi(\boldsymbol{s}', a') \Big], \tag{2.1}$$

where $\mathbb{E}_{\pi(a'|\boldsymbol{s}')p(\boldsymbol{s}'|\boldsymbol{s},a)}$ denotes the conditional expectation over $\boldsymbol{s}'$ and a' drawn from $\pi(a'|\boldsymbol{s}')p(\boldsymbol{s}'|\boldsymbol{s}, a)$ given $\boldsymbol{s}$ and a. This recursive expression is called the *Bellman equation for state-action values*.

Based on the Bellman equation, the optimal policy may be obtained by iterating the following two steps:

$$\text{Policy evaluation:} \quad Q^\pi(\boldsymbol{s}, a) \longleftarrow r(\boldsymbol{s}, a) + \gamma \mathbb{E}_{\pi(a'|\boldsymbol{s}')p(\boldsymbol{s}'|\boldsymbol{s},a)} \Big[Q^\pi(\boldsymbol{s}', a') \Big].$$

$$\text{Policy improvement:} \quad \pi(a|\boldsymbol{s}) \longleftarrow \delta \left(a - \operatorname*{argmax}_{a' \in \mathcal{A}} Q^\pi(\boldsymbol{s}, a') \right).$$

In practice, it is sometimes preferable to use an explorative policy. For example, *Gibbs policy improvement* is given by

$$\pi(a|\boldsymbol{s}) \longleftarrow \frac{\exp(Q^\pi(\boldsymbol{s}, a)/\tau)}{\int_{\mathcal{A}} \exp(Q^\pi(\boldsymbol{s}, a')/\tau) \mathrm{d}a'},$$

where $\tau > 0$ determines the degree of exploration. When the action space $\mathcal{A}$ is discrete, *ϵ-greedy policy improvement* is also used:

$$\pi(a|\boldsymbol{s}) \longleftarrow \begin{cases} 1 - \epsilon + \epsilon/|\mathcal{A}| & \text{if } a = \operatorname*{argmax}_{a' \in \mathcal{A}} Q^\pi(\boldsymbol{s}, a'), \\ \epsilon/|\mathcal{A}| & \text{otherwise,} \end{cases}$$

where $\epsilon \in (0, 1]$ determines the randomness of the new policy.