Space Exploration with Deep Reinforcement Learning

Kwasi Debrah-Pinamang
School of Computer, Data, and Information Sciences
University of Wisconsin-Madison
Madison, WI, USA
kdebrahpinam@wisc.edu

Abstract:

One of the main challenges in space robotics is getting robots to perform a complex task completely autonomously. An example of something considered as a complex task would be "assembling a structure using provided components" or "survey this plot of land and report anything interesting" [2]. The purpose of this paper is to investigate the second of these two tasks. In other words, what is a method that we can use in order to have a robot go on an exploration mission autonomously for the sake of discovery? Using deep reinforcement learning, we can theorize a way to have a robot autonomously survey a plot of land in order to ensure that we would obtain many interesting observations from the land.

Introduction:

There are three main ways that robots learn: under complete supervision, under no supervision, and under some supervision. Without any supervision, robots are forced to learn everything themselves. Likewise, with complete supervision, the robot is almost told explicitly what to do. Having a robot learn with some supervision is a process known as reinforcement learning [3]. In reinforcement learning, robots are put in an unfamiliar environment and are rewarded or punished based on their actions. The main goal of the robot is to maximize the reward amount while traversing the environment.

There are several terms that must be defined in order to understand the process of reinforcement learning. For starters, we have the state and the action. The state is the robot's current position in the land we have set, while the action is simply what the robot will do at each opportunity to make a choice. Next, we have the agent and the environment. The agent is what is being trained. In our case, this will be the robot doing the exploring. The environment is the location where the learning takes place. The robot is incapable of changing the environment, it can only modify its actions to deal with the environment's hazards. Additionally, there is the reward function, the state transition model, and the policy, all of which are modeled as $r(s_t,a_t)$, $P(s_{t+1} \mid s_t, a_t)$, and $\pi(s)$, respectively. The reward function dictates how much reward an action will provide, the state transition model models changes to the state, and the policy is the action taken at a certain state. The goal of reinforcement learning is to find a policy from states to actions to maximize rewards. Lastly, we have the value function, which is simply the sum of the reward for each policy. The value function is modeled as $V^{\pi}(s_0)$, with s_0 being the starting state. All of these functions and models are what determine a reinforcement learning model and how the agent will behave [1,8].

Deep learning is a subset of machine learning that revolves around attempting to mimic the human brain in order to make predictions and cluster data. Deep learning works by using neural networks that act similarly to the neurons in our brains. These neural networks consist of multiple layers of connected nodes, with each layer building on the last in order to refine and optimize the task the neural network is performing [4]. As you would guess, deep reinforcement learning is a combination of deep learning and reinforcement learning. More specifically, the process of deep reinforcement learning involves using deep neural networks to approximate any of the components of reinforcement learning, including the value function, policy, the state transition function, or the reward function [6]. Generally speaking, deep reinforcement learning is ideal for environments with high dimensional features or large amounts of data that needs to be collected.

Related Work:

Two researchers affiliated with the University of Hong Kong, Tai Lei and Liu Ming, wrote a paper in 2016 investigating the use of a Deep Q-Network (DQN) for exploring an unknown corridor. In their experiment, they separated the DQN into a supervised deep learning structure and a Q-learning network, then simulated a turtlebot in Gazebo to examine how effective their reinforcement learning method was. [11]

The method proposed in this paper works similarly. Both methods involve the use of deep learning and reinforcement learning in order to achieve the goal, but my proposed method adds an extra step to the deep learning portion in order to achieve a different goal in the process of exploration.

Method:

My proposal for a method of autonomous space exploration revolves around deep reinforcement learning. For this method, we will use a slightly modified version of Q-Learning known as deep Q-Learning, which is one of the algorithms for deep reinforcement learning. Q-Learning is a model-free reinforcement learning algorithm, meaning that the algorithm doesn't learn a model in order to make predictions of future states and rewards. The base algorithm can be written as such:

```
def QLearn(int step_size, int greedy_par):
        Q[] = 0
        for each episode
                draw initial state s
                while (state s is not terminal):
                        perform action a =
ε-greedy(Q), receive r, s'
```

```
            $Q(s,a) = (1-\alpha)Q(s,a) + \alpha(r + \gamma \max Q(s',b))$
            $s \leftarrow s'$
        endwhile
    endfor
```

In words, the base Q-Learning algorithm works in three steps: initialize the Q-Table, choose an action using the Epsilon-Greedy Exploration Strategy, update the Q-table using the Bellman Equation. This process should continue until the Q-table converges. The Q-Table is a data structure used to keep track of the states, actions, and their expected rewards. The Q-table starts out with all values initialized to zero as the agent knows nothing about the environment [1, 5].

The Epsilon-Greedy Exploration Strategy is a strategy for tackling the tradeoff between exploration and exploitation. Exploration means to take an action with unknown consequences, while exploitation means to go with the best strategy found so far. Exploration usually gets a better grasp of the environment and may come across a greater reward, but utility is not always maximized. In exploitation, the rewards are usually maximized but the most optimal strategy might not be discovered. Like mentioned earlier, the Epsilon-Greedy Exploration Strategy is a method for balancing this tradeoff. This strategy works by first selecting an epsilon value that will steadily decrease for each iteration of the for loop. Next, a random value between 0 and 1 is chosen. If epsilon is greater than the random value, the agent will take a random action, otherwise the agent will take the action with the predicted highest reward [1, 5, 7].

The last step is updating the Q-table using the Bellman Equation. The Bellman Equation is as follows:

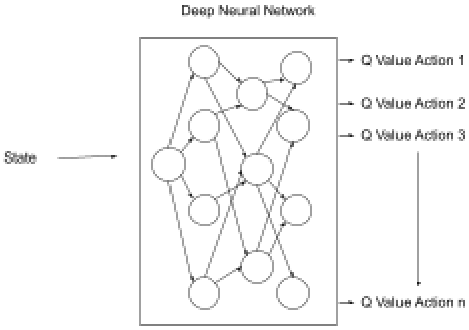
$$Q(S_t, A_t) = (1-\alpha)Q(S_t, A_t) + \alpha * (R_t + \lambda * max_a Q(S_{t+1}, a))$$

The Bellman Equation

with S being the state, A being the action, R being the reward from that action, t being the time step, α being the learning rate, and λ being the discount factor. The time step is the current step the reinforcement learning process is at. Adding 1 to the time step simply takes us a step further in the learning process. The learning rate and discount factor are simply constants defined before the iteration that help in determining the new Q-value like shown. To put it simply, the Bellman Equation updates the current Q-value with the perceived maximum future reward under the assumption that the best actions will be taken. This is done to maximize the value of the policy [1, 5, 7].

With this, we have described how the Q-Learning algorithm works. In Deep Q-Learning, the forming of the Q-table happens a little bit differently. To build the Q-table, a neural network maps input states to Q-table pairs. The neural network is then used to approximate a Q-value. Putting it simply, the Q-table is formed using deep learning. Aside from the construction of the Q-table, the remainder of Deep Q-Learning works the same as regular Q-Learning. Deep Q-Learning works on three important steps: initialize the neural networks, choose an action using the Epsilon-Greedy Exploration

Strategy, and update the network weights using the Bellman Equation [5, 7, 9].



Deep Q-Learning Q-Table

After describing the means of solving the problem, we can now go into greater detail in answering how Deep Q-Learning will help in robotic space exploration. To put it simply, Deep Q-Learning is ideal for more complex environments, which the surface of a celestial body would qualify as. The original problem we were trying to solve is "how can we get a robot to autonomously survey a plot of land and report something interesting"? We can use deep learning in order to measure the level of interest of an area in order to have our agent move in that direction and possibly record its findings. Level of interest could be measured in a variety of ways, including measuring based off of other interesting landmarks found on other celestial bodies or by comparing the scene being viewed to what is considered the default state of the surface. The level of interest of a scene would be the value in this case, which is what would be used in making moves and updating the Q-table. This process would be done in order to potentially find the most interesting landmarks in a shorter amount of time compared to surveying the entire land.

An example of a specific application of this method could be in the investigation of Mars by NASA's rovers back in 2004. In January of this year, NASA sent two rovers, Spirit and Opportunity, to Mars in order to investigate the surface of Mars and report any interesting findings. In the process of the mission, the rovers are usually left to roam autonomously for about 20 hours a day, periodically sending images and carrying out other commands. The method described within this paper could possibly be used in similar expeditions in the future in order to keep the quality of findings the same while likely reducing the time needed to be spent searching.

Besides the applications for space exploration, a similar method could be used for exploration of other uncharted areas, most notably in deep sea exploration or locations on Earth that are difficult for humans to traverse. The method theorized in this paper describes a way of finding interesting landmarks in an unfamiliar environment in a generally more efficient manner, it could definitely be applied beyond space applications, though while the general idea will remain the same, changes must be made in both the software and body of the agent to adapt to its environment.

While this process does sound efficient, it is not flawless. The glaring issue with this potential method is that like mentioned, the whole land would likely not be surveyed. This would mean that it is entirely possible that landmarks aside from the most significant ones are missed.

Despite this weakness, however, this method could prove to be effective for finding landmarks in unknown environments with some experimentation.

References:

[1] Russell, S. J., & Norvig, P. (2010). *Artificial Intelligence: A modern approach*. Prentice-Hall.

[2] L. Pedersen, D. Kortenkamp, D. Wettergreen, I. Nourbakhsh. A Survey of Space Robotics. 2003. Retrieved May 31, 2022 from https://ntrs.nasa.gov/citations/20030054507

[3] Dr. Robert Babushka. Robots that learn like humans. Retrieved June 6, 2022 from https://www.tudelft.nl/en/3me/research/check-out-our-science/robots-that-learn-like-humans#:~:text=Roughly%20speaking%2C%20robots%20can%20learn,one%20in%20a%20given%20situation

[4] IBM Cloud Education. 2020. Deep Learning. Retrieved June 6, 2022 from https://www.ibm.com/cloud/learn/deep-learning

[5] Mike Wang. 2020. Deep Q-Learning Tutorial: miniDQN. Retrieved June 6, 2022 from https://towardsdatascience.com/deep-q-learning-tutorial-mindqn-2a4c855abffc

[6] Li, X. 2018. Deep Reinforcement Learning (Oct 2018), 14-15. DOI: https://doi.org/10.48550/arXiv.1810.06339

[7] Peter Foy. 2021. Deep Reinforcement Learning: Guide to Deep Q-Learning. Retrieved June 10, 2022 from https://www.mlq.ai/deep-reinforcement-learning-q-learning/

[8] Rohan Jagtap. 2020. Understanding Markov Decision Process (MDP). Retrieved June 12, 2022 from https://towardsdatascience.com/understanding-the-markov-decision-process-mdp-8f838510f150

[9] Alind Gupta. 2022. Deep Q-Learning. Retrieved June 12, 2022 from https://www.geeksforgeeks.org/deep-q-learning/

[10] Marshall Brain & Kate Kershner 2004. How the Mars Exploration Rovers Work. Retrieved August 2, 2002 from https://science.howstuffworks.com/mars-rover.htm

[11] L. Tai and M. Liu, "A robot exploration strategy based on Q-learning network," *2016 IEEE International Conference on Real-time Computing and Robotics (RCAR)*, 2016. DOI: 10.1109/RCAR.2016.7784001.