Eyoel Tesfu

SNHU

CS-370 Current Emerging Trends in Computer Science

**Project Two: Design Defense**

**How the Intelligent Agent Works**

The intelligent agent works by interacting with its environment (the maze), making decisions (actions) based on either exploration (random directional choice) or exploitation (using predictions from the trained model). It uses the Q-learning algorithm to maximize the expected cumulative reward over time by choosing the most optimal actions that lead to the most rewarding outcomes (Watkins, 1989).

**Taking a look at the Algorithm**

Q-learning is a type of reinforcement learning algorithm that is used to find the most optimum action selection policy for a given finite Markov decision process (Watkins, 1989). It learns an action-value function which leads to the agent taking a given action in a particular state and then following the optimal policy.

It can handle problems with stochastic transitions and rewards without requiring adaptations. It is off policy which means that it learns the value of the optimal policy independent of the agent’s actions.

A basic Q-learning algorithm may be slow to converge. This is the reason that deep Q-learning is used with neural networks as it can generalize across states to speed up the learning process (Mnih et al., 2013).

**Human vs. Machine Approach to Solving the Problem**

Humans would first observe and visually process the maze. They might recognize patterns in the maze based on prior knowledge or experience. They will try out different paths, fail, remember dead-ends, and try to avoid these in their future attempts. Over time, they might develop a strategy to solve the problem (like always turning left or using wall following methods).

A machine learning agent will start at a random free cell and observe the state of the maze. It will then choose an action based on either exploration (random choice) or exploitation (model’s prediction). After taking the action, the agent will receive a reward or penalty and updates its knowledge (Q-values). It trains a neural network to approximate the Q-function. This process will be repeated over numerous episodes to refine the agent’s policy.

Both humans and the agent would use trial and error, and both will refine their strategies overtime based on given feedback. However, humans rely on intuition, prior knowledge, and visual interpretation to solve the problem, while the agent uses a learned policy from interacting with the environment. Additionally, the agent’s decisions are automatically optimized to maximize the expected rewards (Mnih et al., 2013).

**The Purpose of the Intelligent Agent in Pathfinding**

Pathfinding is the act of navigating through a space to find the optimal path from a start to an end point. The purpose of the agent in pathfinding is to autonomously learn the best strategy to navigate the maze and reach the treasure in the fewest possible steps.

When navigating the maze, the agent uses either exploitation or exploration to choose its next action. Exploitation involves using the agent’s current knowledge to make the best decision, and exploration involves taking random actions to discover new strategies (Watkins, 1989).

The ideal proportion to use between these varies based on the problem to be solved and the stage where the learning is at. Initially, high exploration will be more beneficial since there is almost no prior knowledge. Over time, as the agent learns, exploitation should be increased to make use of the acquired knowledge. In the code, the variable **epsilon** controls the balance between the two.

Reinforcement learning here involves the agent interacting with the environment to maximize its cumulative rewards (Watkins, 1989). During pathfinding, the agent (pirate) learns by taking actions (steps in certain directions), receiving rewards (or penalties), and updating its policy to optimize the path to the treasure. Over time, the agent learns the best actions to take in different states to reach the treasure efficiently, all to maximize its cumulative rewards.

**Use of Algorithm in Solving Complex Problems**

AI and Machine Learning Algorithms can autonomously handle and solve complex problems by analyzing large amounts of data, identifying the patterns, and optimizing their decisions based on perceived feedback.

In our code, to solve the complex problem of finding the most efficient pathway through the maze, we implemented a deep Q-learning algorithm. The algorithm includes a neural network that approximates the Q-function, and an agent that interacts with the maze and stores its experiences (state, action, reward, next state) in a ‘experience’ replay buffer. The experiences are then randomly sampled to train the neural network to update its weights to better predict Q-values. I also implemented the Bellman equation to calculate the target Q-values based on the current rewards and the maximum predicted Q-values of upcoming states (Mnih et al., 2013). The agent then learns an optimal policy to navigate the maze by iteratively updating the neural network via the agent’s experience.

**References**

Mnih, V., Badia, A. P., Mirza, M., Graves, A., Lillicrap, T., Harley, T., ... & Kavukcuoglu, K. (2016). Asynchronous methods for deep reinforcement learning. *In International conference on machine learning* (pp. 1928-1937).

Sutton, R. S., & Barto, A. G. (2018). Reinforcement learning: An introduction. *MIT press*.

Watkins, C. J. C. H. (1989). Learning from delayed rewards (Ph.D. thesis). *Cambridge University*.