Design Defense for the Pirate Intelligent Agent Pathfinding Problem

By: Nicholas Kreuziger

**Approach to Solving the Problem**

In designing the pirate intelligent agent to navigate the maze and find the treasure, I utilized a deep Q-learning algorithm. This approach leverages reinforcement learning where the agent learns to make decisions by receiving rewards for its actions, progressively improving its policy for navigating the maze. The goal was to enable the pirate to consistently find the treasure before the human player.

**Human vs. Machine Approaches to Problem Solving**

**Human Approach:**

A human solving the maze would typically use the following steps:

1. **Visual Inspection:** Look at the entire maze to identify the start and end points.
2. **Path Planning:** Mentally map out potential paths to the treasure, avoiding obstacles.
3. **Trial and Error:** Move step-by-step, adjusting the path when encountering obstacles.
4. **Memory and Learning:** Remember successful paths or dead ends to improve subsequent attempts.

**Machine Approach:**

The intelligent agent, however, takes the following steps:

1. **Initialize Environment:** The agent starts with no knowledge of the maze.
2. **Random Exploration:** Initially, it explores the maze randomly to gather information about the environment.
3. **Experience Replay:** Stores past experiences (state, action, reward, next state) in memory.
4. **Learning from Experiences:** Uses experiences to train a neural network, updating Q-values (expected future rewards) for state-action pairs.
5. **Balancing Exploration and Exploitation:** Uses ε-greedy policy to balance between exploring new paths and exploiting known paths for better rewards.

**Similarities and Differences:**

* **Similarities:** Both approaches involve exploration, learning from past experiences, and progressively improving their strategies.
* **Differences:** A human relies on visual and cognitive skills, whereas the machine uses mathematical models and iterative learning processes. The machine can explore and learn from a much larger number of trials compared to a human.

**Purpose of the Intelligent Agent in Pathfinding**

The primary purpose of the intelligent agent in this pathfinding problem is to autonomously navigate the maze and locate the treasure efficiently. This involves making decisions at each step to maximize the cumulative reward, ultimately finding the shortest or most efficient path to the goal.

**Exploitation vs. Exploration**

**Exploration:** The agent tries out new actions to discover their effects and gather more information about the environment. **Exploitation:** The agent uses the knowledge it has already gained to make the best possible decisions to maximize rewards.

**Ideal Proportion of Exploitation and Exploration:**

The ideal proportion of exploration and exploitation changes over time. Initially, more exploration is needed to gather sufficient information about the environment. Over time, as the agent's knowledge improves, the focus should shift towards exploitation. A common strategy is to start with a high exploration rate (ε) and gradually reduce it, as described by Gulli and Pal (2017). For instance, the ε value decreases from 0.1 to 0.001 over the training period, ensuring a balance between exploration and exploitation early on and favoring exploitation as the agent becomes more knowledgeable.

"In case of ε-greedy exploration, the agent chooses the action suggested by the network with probability 1-ε or an action uniformly at random otherwise... ε-greedy exploration ensures that in the beginning the system balances the unreliable predictions made from the Q-network with completely random moves to explore the state space, and then settles down to less aggressive exploration (and more aggressive exploitation) as the predictions made by the Q-network improve." (Gulli & Pal, 2017)

**Reinforcement Learning for Pathfinding**

Reinforcement learning helps the agent determine the path to the treasure by continuously improving its decision-making policy based on rewards received from the environment. The agent learns which actions lead to higher rewards (finding the treasure) and which lead to penalties (hitting obstacles), refining its strategy over time through iterative learning.

**Implementation of Deep Q-Learning Using Neural Networks**

Deep Q-learning combines Q-learning with deep neural networks to handle large state spaces and complex problems. The steps in implementing deep Q-learning for the pirate pathfinding problem are:

1. **State Representation:** The maze is represented as a matrix where each cell corresponds to a state.
2. **Action Space:** The possible movements (up, down, left, right) are the actions the agent can take.
3. **Reward Function:** The agent receives rewards for finding the treasure and penalties for hitting obstacles or taking unnecessary steps.
4. **Neural Network:** A deep neural network approximates the Q-values for each state-action pair. It predicts the expected future rewards for actions taken from a given state.
5. **Experience Replay:** Stores past experiences in a replay buffer to break the correlation between consecutive experiences and improve learning stability.
6. **Training Process:** The agent samples batches of experiences from the replay buffer to train the neural network. The loss function measures the difference between predicted Q-values and target Q-values.
7. **Exploration vs. Exploitation:** Uses an ε-greedy policy to balance exploration of new actions and exploitation of known rewarding actions. Initially, the agent explores more to gather information, and gradually shifts to exploiting the learned policy.

The deep Q-learning algorithm uses a neural network to approximate the Q-values for state-action pairs, which helps the agent make informed decisions. By balancing exploration and exploitation, the agent progressively learns the optimal path to the treasure, demonstrating the power of reinforcement learning in solving complex pathfinding problems.

**Citation:**

Gulli, Antonio, and Sujit Pal. *Deep Learning with Keras: Implementing Deep Learning Models and Neural Networks with the Power of Python*, Packt Publishing, Limited, 2017. ProQuest Ebook Central, <http://ebookcentral.proquest.com/lib/snhu-ebooks/detail.action?docID=4850536>.