Daviel Machet

CS-370

Design Defense

When comparing how a human and a machine solve this problem, the process is very different. A human would look at the maze, identify open paths, and use logic to find the shortest route while avoiding dead ends. If they reach a dead end, they will backtrack and try a different path, repeating this process until they reach the goal. A machine, on the other hand, doesn’t have intuition. Instead, it starts by moving randomly through the maze. As it moves, it collects data and uses reinforcement learning to gradually improve its decision-making. Over time, it learns which paths lead to success and which ones don’t. Unlike a human, the machine does not "see" the maze but instead relies on numerical representations and trial-and-error learning to optimize its movements.

The purpose of the intelligent agent in pathfinding is to automatically find the best route from the starting position to the treasure while avoiding obstacles. It does this by recognizing which actions lead to a win and which result in failure. A key part of this learning process is balancing two approaches: exploration and exploitation. Exploration means the agent tries new paths, even if they are not always the best, to discover better options. Exploitation means the agent follows what it has already learned to make the best possible move. At the beginning of training, the agent explores more to gather information. As it learns, it shifts toward exploitation to move more efficiently. In my implementation, I used an exploration decay rate (decayRate = 0.1) to help the agent smoothly transition from trying random paths to making smarter choices based on experience. This balance is crucial because too much exploration can make learning inefficient, while too much exploitation can cause the agent to miss better routes.

A technique called experience replay helps the agent store past experiences and learn from them over time, rather than just reacting to the most recent actions. This method allows the agent to reuse past experiences to improve learning efficiency and avoid forgetting previously successful paths (Neves,2024). Without experience replay, the agent might get stuck in repetitive loops or rely too heavily on recent experiences rather than optimizing for long-term success. Reinforcement learning also works by using rewards and penalties to guide the agent’s decisions. If the pirate moves toward the treasure, it earns positive rewards. If it hits a wall or gets stuck, it receives negative rewards. By repeating this process, the pirate learns the best way to reach the treasure.

To train the pirate using deep Q-learning, I built a neural network that takes in the maze’s current state and predicts the best move based on previous learning. The model includes multiple layers that process this information and adjust as it gets feedback. I used the Adam optimizer, which is widely used in reinforcement learning because of its ability to adjust learning rates dynamically and improve convergence speed (GeeksforGeeks, 2024a). Additionally, I used the Mean Squared Error (MSE) loss function to minimize mistakes in predictions, as it helps measure how well the model is learning to approximate the best possible actions (GeeksforGeeks, 2024b). To make the learning process more effective, I used experience replay, which helps the model learn from multiple past moves rather than just the most recent one. I also included exploration decay so that at first, the agent focuses on discovering different paths, but later, it relies on what it has learned to make better decisions. The model was trained over 15,000 epochs (n\_epoch = 15000) to ensure it had enough time to learn, and I used batch training (batch\_size = 16) to make the learning process faster and more efficient. This method ensures that the pirate does not just memorize a single solution but can adapt to different maze configurations if necessary.

References:

GeeksforGeeks. (2024a). *What is Adam Optimizer?* GeeksforGeeks. https://www.geeksforgeeks.org/adam-optimizer/

GeeksforGeeks. (2024b). *Mean squared error*. GeeksforGeeks. https://www.geeksforgeeks.org/mean-squared-error/

Neves, D. E., Ishitani, L., & Júnior, Z. K. G. D. P. (2024). Advances and challenges in learning from experience replay. *Artificial Intelligence Review*, *58*(2). https://doi.org/10.1007/s10462-024-11062-0