Madison Fleitas

Cs370

Project Two

Design Defense

When I think about how a person would solve the maze, it’s all about using intuition, visual cues, and memory. I’d start by looking for a route that seems to head toward the treasure, steering clear of obvious obstacles. If I hit a dead end, I’d remember the paths I’ve tried and backtrack, relying on my spatial sense to decide which direction to try next. Humans naturally use these skills to adapt quickly and avoid repeating mistakes; when playing a video game, constantly going back and forth between what works and what doesn’t work by going to the save point and trying again is a repetitive nature that allows us to learn and adapt. In contrast, the AI agent doesn’t have this kind of awareness or problem-solving intuition. Instead, it learns through reinforcement learning, which means a lot of trial and error. The agent begins with no idea about the maze and figures things out by interacting with it over many training episodes. It gets rewarded for positive outcomes, like reaching the treasure, and penalized for negative ones, like hitting a wall or wandering aimlessly. These rewards and penalties guide the agent to update its understanding of what actions are valuable, helping it find the most efficient way through the maze. While humans can use spatial reasoning and past experiences, the AI relies on learning patterns from repeated training.

For the pirate agent in this project, its job was to solve a pathfinding problem: navigate from the starting point to the treasure as efficiently as possible. To do this, it had to balance exploring new paths and exploiting what it already knew about successful routes. At first, the agent needs to focus more on exploration because it doesn’t know much about the maze. As training goes on, the agent shifts more toward exploitation, where it uses what it’s learned to make decisions that are more likely to get it to the goal. I controlled this balance using the exploration factor. I started with a higher value to encourage trying new paths, then gradually decreased it so the agent would focus more on the actions it knew worked well. Reinforcement learning played a big role in this, as it allowed the agent to associate certain actions with rewards or penalties, helping it learn which paths to follow and which to avoid.

The deep Q-learning approach builds on regular Q-learning by using a neural network to approximate Q-values, which makes it possible for the agent to generalize its learning across the larger state space of the 8x8 maze. This means that instead of keeping a basic Q-table for every possible state-action pair, the agent uses the neural network to predict expected rewards, even for states it hasn’t encountered before. To keep the training stable, I used experience replay and a target network. Experience replay stores the agent’s experiences in memory and uses them to train the model in batches, which helps break the pattern of training on consecutive actions and stabilizes learning. The target network adds another layer of stability by keeping the target Q-values fixed for a set number of steps, so the training isn’t too volatile.

In the end, deep Q-learning provided a strong framework for solving the pathfinding problem, allowing the pirate agent to learn efficient strategies from its experiences. However, the results heavily relied on how well I managed the balance between exploration and exploitation and fine-tuned the training process. With careful adjustments, the agent consistently found the treasure, demonstrating how reinforcement learning can tackle complex problems like navigating a maze.

Resources:

Sutton, R. S., & Barto, A. G. (2018). *Reinforcement learning: An introduction* (2nd ed.). MIT Press

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., ... & Hassabis, D. (2015). Human-level control through deep reinforcement learning. *Nature*, *518*(7540), 529-533. https://doi.org/10.1038/nature14236