Design Defense

Eskinder Kassahun

Design Defense

As a AI developer, I used a deep Q-learning approach to learn an 8x8 maze-navigating pirate smart agent that will be able to beat the human player to the treasure. In contrast to the human condition, wherein humans use logic, spatial reasoning, and intuition to solve mazes, the agent learns through trial and error. It begins with knowing nothing about the maze and slowly learning what action gives the greatest reward in the long term through world experience and reward. This is the essence of reinforcement learning where actions are taken based on either exploitation (following the best known so far path) or exploration (exploring along new paths).

To train the agent, I employed a neural network constructed using Keras that translates observed maze states into optimal actions. The pirate starts at a random location, searches, chooses an action, and gets a reward. These are stored and utilized to train the model in batches. As time progresses, the agent learns what sequences of actions bring it closer to the treasure. I used an epsilon-greedy policy to strike a balance between exploitation and exploration, which began at 0.1 to permit the occasional random action and generally stick to learned policies.

The primary distinction between a human and this kind of intelligent agent is how each sees and solves the problem. The human is able to view the entire maze and anticipate routes using vision and rational analysis, whereas the agent learns solely from experience and feedback. However, both methods have the ultimate goal of arriving at the treasure without obstacles and using the fewest number of decisions possible. Gleaning experience through repeated experimentation by the agent makes it an environment-friendly method for environments too complicated to utilize manual logic or planning.

Overall, deep Q-learning was a great algorithm for this pathfinding problem. It enabled the agent to learn an optimal navigation policy without the use of hardcoded rules. With the use of neural networks, experience replay, and a reward-based learning system, the agent was effectively able to have a high win rate. This project showcases the strength of reinforcement learning in resolving dynamic and intricate problems in game development and elsewhere.

Reference

Li, Y. (2017). Deep reinforcement learning: An overview. arXiv preprint arXiv:1701.07274. <https://arxiv.org/abs/1701.07274>

Sutton, R. S., & Barto, A. G. (2018). *Reinforcement learning: An introduction* (2nd ed.). MIT Press. https://www.andrew.cmu.edu/course/10-703/textbook/BartoSutton.pdf