Mohamed Babiker

CS-370

Module 7

Module 7: Design Defense

The goal of solving the pirate maze problem was to develop an intelligent agent that could discover the treasure and navigate the maze with ease. This design defense describes the methodology, the intelligent agent's operation, and an analysis of the selected algorithm, Deep Q Network (DQN).

Exploration and exploitation would probably be combined by a person navigating the maze. They would begin by aimlessly meandering to find potential routes, then use the knowledge they acquired to make wise choices and effectively accomplish the objective. In contrast to the intelligent agent, a person has the advantage of intuition and forward vision, which enables them to plan several movements ahead of time and avoid blind barriers.

The pirate intelligent agent employs a similar tactic and is powered by a Python DQN. It discovers the best course of action for every state by exploring several options (exploration) through reinforcement learning. Based on a learning rate degradation, the agent moves from aimless meandering to making use of acquired information (exploitation). The algorithm finds the best route by balancing exploration and exploitation, a technique inspired by human-like problem-solving.

The intelligent agent's main goal is to navigate the maze and find the prize before the human player does. This entails becoming familiar with a policy that associates states—that is, places on the map—with movements. The agent seeks to balance exploration and exploitation in order to maximize the overall predicted payoff over time.

While exploitation makes decisions about actions based on information already known, exploration makes decisions about actions based on fresh information. After gaining experience, the intelligent agent switches from exploration to exploitation. The learning process determines the optimal ratio. Before depending only on experience-based forecasts, a declining learning rate encourages further investigation in the early stages (90 epochs of trial and error).

Reinforcement learning learns a policy at each state, assisting the agent in figuring out the best route to the objective. Positions on the map correlate with states, and movements with activities. The agent updates its action-value function through exploration and exploitation, convergent to optimal values and a precise strategy for determining the best route to the prize.

Decisions are guided by the agent's rewards and penalties, which are implemented in Python using DQN. Achieving the treasure (1.0), receiving a penalty for getting blocked or going to the same square again, and knowing which movements are legal and illegal are among the rewards.

The shift from exploration to exploitation is governed by a learning rate decline, which draws inspiration from Yang (2022). Upon reaching a certain learning rate, the agent ceases its aimless meandering, guaranteeing that the knowledge it has acquired is put to use.
By the 248th epoch, the agent has a 100% success rate in discovering the prize, demonstrating excellent learning. The pirate agent's ultimate route is depicted in Figure 1.

In summary, the intelligent agent uses a DQN algorithm to effectively solve the pathfinding issue. A learning rate decline combined with a balance between exploration and exploitation resembles how humans solve problems. The design guarantees that the agent will successfully navigate the maze and accomplish its objective.

Cited

Model-free (reinforcement learning). (2023, March 3). In Wikipedia. https://en.wikipedia.org/w/index.php?title=Model-free_(reinforcement_learning)&oldid=1142602733

Yang, A. (2022, July 24). What is exploration vs. exploitation in reinforcement learning? Medium. https://angelina-yang.medium.com/what-is-exploration-vs-exploitation-in-reinforcement-learning-a3b96dcc9503