Design Defense

Derricko Swink

Solving problems involves very different strategies when comparing human cognition and machine learning. Humans approach problem-solving by using logic, intuition, pattern recognition, and spatial awareness. They can visualize the entire maze, estimate possible paths, and adjust their strategy after only a few trials. This rapid adaptability, combined with the ability to infer shortcuts and apply abstract reasoning, allows humans to solve mazes efficiently. In contrast, machines rely on numerical computations, structured rules, and experience gathered through repetition. In the context of reinforcement learning, machines learn through a cycle of trial and error, receiving feedback in the form of rewards or penalties and gradually improving performance over many iterations. As discussed by Sutton and Barto (2018), reinforcement learning enables machines to learn optimal policies from interaction with an environment without explicit instructions.

A human solving a maze typically starts by analyzing the layout visually and identifying potential paths. The person then chooses a direction based on heuristics, such as following the right-hand wall. If they encounter a dead end, they backtrack and try another path. Throughout the process, they rely on memory to avoid previously failed routes and adjust their decisions based on learned experiences. This method allows them to adapt quickly and find an efficient path to the goal.

In contrast, the intelligent agent in Project Two uses a Deep Q-Learning algorithm to solve the pathfinding problem. It starts in a predefined state and must choose actions from a finite set, such as moving up, down, left, or right. At each step, it uses an epsilon-greedy strategy to decide whether to explore new actions or exploit known ones. After taking an action, the agent receives

a reward depending on the outcome—positive for reaching the treasure, negative for falling into a trap or hitting a wall. It stores this experience in a memory buffer and later samples mini-batches of these experiences to train a neural network. Over time, the agent's Q-value estimates improve, guiding it to make better decisions and eventually learn the optimal path to the goal.

There are both similarities and differences between human and machine problem-solving in this context. Both the human and the agent rely on learning from experience and adjusting future actions accordingly. However, humans can adapt to new environments quickly and make logical inferences even with incomplete information. In contrast, the agent requires many iterations and structured feedback to learn effectively. While humans use intuition and visual memory, the agent uses numerical optimization and value estimation to determine the best course of action.

The primary purpose of the intelligent agent in this project is to autonomously navigate the environment and find the most efficient path to the treasure. This mirrors real-world applications in robotics, autonomous vehicles, and game development, where intelligent agents must operate in dynamic and often unknown environments. The reinforcement learning approach allows the agent to learn from interactions and improve over time without being explicitly programmed for every scenario.

Exploration and exploitation are fundamental aspects of reinforcement learning. Exploration involves trying new or random actions to discover potentially better outcomes, while exploitation leverages known strategies to maximize rewards. Striking a balance between these two is crucial. At the beginning of the training process, more exploration is needed—typically around 70%—to allow the agent to learn about the environment. Over time, the exploration rate should decay to around 5–10%, enabling the agent to exploit its learned knowledge effectively. Mnih et al.

(2015) highlight that this decaying exploration strategy is essential for achieving human-level performance in reinforcement learning tasks.

Reinforcement learning is particularly effective in helping the pirate agent find the path to the treasure. As the agent receives feedback on its decisions through rewards and penalties, it gradually builds an understanding of which actions are most beneficial in different states. Using Q-learning, it estimates the expected value of taking specific actions from each state and continuously refines this estimate through updates. This process enables the agent to avoid traps, learn from past mistakes, and develop a policy that consistently leads to the treasure.

In this project, deep Q-learning was implemented using a neural network to approximate the Q-function. The neural network takes the current state as input and outputs the estimated Q-values for all possible actions. The Q-values are updated based on the Bellman equation, incorporating immediate rewards and the discounted value of future rewards. The agent stores its experiences in a replay buffer and samples mini-batches to train the network, which helps stabilize learning by breaking the temporal correlation between experiences. This method allows the model to generalize across different states and learn effective strategies even in complex environments. As discussed by Francois-Lavet et al. (2018), the combination of Q-learning and deep learning enables agents to handle large state spaces and learn sophisticated behaviors over time.

References

- Sutton, R. S., & Barto, A. G. (2018). *Reinforcement Learning: An Introduction*. MIT Press.
- Mnih, V., Kavukcuoglu, K., Silver, D., et al. (2015). "Human-level control through deep reinforcement learning." *Nature*, 518(7540), 529–533.

- Francois-Lavet, V., Henderson, P., Islam, R., Bellemare, M. G., & Pineau, J. (2018). "An Introduction to Deep Reinforcement Learning." *Foundations and Trends® in Machine Learning*, 11(3–4), 219–354.
- Kaelbling, L. P., Littman, M. L., & Moore, A. W. (1996). "Reinforcement learning: A survey." *Journal of Artificial Intelligence Research*, 4, 237–285.