Mark Eilers

CS 370: Current/Emerging Trends

7-3 Project Two

April 18, 2024

7-3 Project Two

Human and machine approaches to problem-solving exhibit notable differences, particularly shown in maze-solving scenarios. A human's strategy for navigating a maze typically begins with visual inspection and mental mapping of the maze's layout. They rely on spatial reasoning to recognize obstacles like walls and dead ends, using landmarks or memory to guide their decisions. Through trial and error, humans explore different paths, learning from past experiences and adapting their strategy accordingly. In contrast, the intelligent agent implemented in the code snippet employs a systematic approach to maze-solving. It perceives the maze through input data and utilizes decision-making algorithms like Q-learning to choose actions that maximize its chances of reaching the goal efficiently. The agent starts by randomly selecting a starting point and exploring the maze while balancing exploitation (choosing actions based on past experiences) and exploration (trying new actions to discover potentially better strategies or paths). It learns from past experiences stored in its memory and adjusts its strategy over time to optimize its pathfinding efficiency.

Despite these differences, there are similarities between human and machine approaches to maze-solving. Both involve perception of the environment, decision-making based on available information, and learning from past experiences. Both aim to reach the goal by navigating through the maze efficiently, although through different mechanisms and strategies. However, humans may incorporate emotions, preferences, and subjective judgment into their decision-making process, while machines operate based on objective algorithms and predefined rules. Additionally, humans rely on visual inspection and spatial reasoning, while machines interpret data algorithmically to make decisions.

The intelligent agent deployed in pathfinding tasks, such as navigating a maze to locate the treasure as depicted in the project, serves the fundamental purpose of efficiently charting a course through complex environments to reach predefined objectives. Central to the agent's success is its ability to strike a delicate balance between exploitation and exploration. Exploitation entails leveraging existing knowledge gained from past interactions with the environment to select actions that are known to yield favorable outcomes (Sutton 2018). In contrast, exploration involves venturing into uncharted territory by trying new actions, thereby uncovering potentially superior strategies or pathways (Sutton 2018). Through a dynamic interplay between exploitation and exploration, the agent aims to optimize its decision-making process and ultimately achieve its goal of efficiently navigating the maze. While exploitation exploits known strategies for immediate gains, exploration seeks to broaden the agent's understanding of the environment and discover better long-term solutions.

Determining the ideal proportion of exploitation and exploration poses a challenge, depending upon various factors such as the maze's complexity and the agent's learning rate. In the stages of exploration, a higher emphasis on exploration is necessary to familiarize the agent with that of the environment. By probing various paths and experimenting with different actions, the agent can construct a comprehensive mental map of the maze, identifying potential obstacles and optimal routes. However, as the agent accumulates experience and acquires a deeper understanding of the maze, a gradual transition towards exploitation becomes beneficial. Exploitation capitalizes on the agent's refined knowledge, allowing it to exploit known high-reward actions and navigate the maze more efficiently.

Balancing exploitation and exploration often entails employing an experiential approach, such as an epsilon-greedy strategy. This strategy enables the agent to dynamically adjust its exploration-exploitation tradeoff by selecting actions based on the highest expected reward with a probability of (1 - epsilon) while exploring random actions with a probability of epsilon (Mnih V. et al, 2015). By fine-tuning the value of epsilon over time, the agent can adapt its strategy to changing environmental conditions and optimize its pathfinding efficiency accordingly.

Reinforcement learning plays a crucial role in determining the path to the goal, represented by the treasure, for the agent, which in this scenario, embodies the pirate navigating through the maze. Through reinforcement learning, the agent learns to make decisions by interacting with the environment, receiving feedback in the form of rewards or penalties based on its actions. By iteratively exploring the maze and updating its strategy based on received rewards, the agent gradually learns to navigate the maze efficiently towards the treasure. Reinforcement learning algorithms, such as Q-learning, enable the agent to estimate the expected future rewards associated with each action in a given state (Watkins C. 1992). By maximizing the cumulative reward over time, the agent learns an optimal policy or strategy for reaching the goal.

The implementation of algorithms, particularly deep Q-learning using neural networks, offers a powerful approach to solving complex problems like maze navigation. Deep Q-learning combines reinforcement learning with neural networks to approximate the Q-function, which estimates the expected future rewards for each action in a given state (Osband, I, et al. 2019). In the context of this maze-solving game, deep Q-learning involves several key steps. First, a neural network architecture is defined, typically consisting of multiple layers of neurons. The neural network takes the maze's state as input and outputs Q-values for each possible action, representing the expected future rewards.

During training, the agent interacts with the environment, collecting experiences in the form of state-action-reward-next state tuples. These experiences are stored in a replay memory buffer to break correlations between consecutive experiences and stabilize training. Periodically, the agent samples batches of experiences from the replay memory and uses them to update the neural network parameters using gradient descent to minimize the temporal difference error between predicted and target Q-values. This process iterates over multiple epochs, gradually improving the neural network's ability to approximate the Q-function and learn an optimal policy for maze navigation (Hasselt, H.V. 2010).

The use of deep Q-learning with neural networks offers several advantages for solving complex problems like maze navigation. Neural networks can approximate complex functions and capture intricate relationships between inputs and outputs, enabling the agent to learn sophisticated strategies for navigating the maze. Additionally, deep Q-learning allows for end-to-end learning, where the agent directly learns from raw sensory input without requiring handcrafted features or domain-specific knowledge. However, implementing deep Q-learning also entails challenges such as selecting an appropriate neural network architecture, tuning hyperparameters, and managing the trade-off between exploration and exploitation. Despite these challenges, deep Q-learning with neural networks has demonstrated remarkable success in solving a wide range of complex problems, including maze navigation.

Works Cited:

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

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., ... & Petersen,

S. (2015). Human-level control through deep reinforcement learning. Nature, 518(7540), 529-533.

Watkins, C. J. C. H., & Dayan, P. (1992). Q-learning. Machine learning, 8(3-4), 279-292.

Osband, I., Blundell, C., Pritzel, A., & Van Roy, B. (2016). Deep exploration via

bootstrapped DQN. In Advances in neural information processing systems (pp. 4026-4034).

Hasselt, H. V. (2010). Double Q-learning. In Advances in neural information processing systems

(pp. 2613-2621).