Jordan McNairy

SNHU CS 370

06/23/2024

Design Defense

This paper investigates the application of Deep Q-learning (DQL) to solve a pathfinding problem within a treasure hunt game. I propose a framework for an intelligent agent, modeled as a pirate, that utilizes DQL to navigate a game environment and locate a treasure before a human player. This document details the implemented approach, explains the decision-making processes employed by the intelligent agent, and evaluates the effectiveness of the chosen DQL algorithm in achieving the objective.

Human navigation within a maze environment relies heavily on visuospatial processing and a combination of cognitive processes to inform decision-making. Initially, individuals may employ an exploratory strategy, traversing various paths and utilizing trial-and-error to locate the goal. Encountering dead ends necessitates backtracking, a behavior that reinforces spatial memory of unproductive routes and guides subsequent exploration. Furthermore, humans engage in logical reasoning to optimize their search process. This involves retaining information about previously explored paths and dead ends, thereby eliminating unproductive choices and facilitating more efficient navigation. Additionally, some individuals may utilize heuristic strategies, such as the "right-hand rule." This strategy involves consistently following a specific wall (e.g., the right wall) to ensure complete exploration of the maze while avoiding revisiting previously explored areas. The implementation of heuristic strategies demonstrates a systematic approach to navigating complex maze environments.

In contrast to human navigation, which relies on visuospatial cues, the intelligent agent uses reinforcement learning to solve the pathfinding issue. This strategy begins with the agent having no past knowledge of the maze environment. Through engagement, the agent iteratively learns the best navigation tactics. The basic decision-making process consists of multiple steps: initialization, exploration, learning, and exploitation. During startup, the agent creates state and action spaces that reflect its view of the environment and possible actions. Furthermore, Q-values, which estimate the future benefit of performing a certain action in a given condition, are set. Exploration motivates agents to find new routes by randomly picking actions with a probability of epsilon (ε). Exploitation prioritizes actions with higher Q-values (1-ε probability) to take use of previously gained knowledge. This balance of inquiry and exploitation is critical for successful learning. Finally, the agent changes its Q-values as it interacts with the environment (performing actions, getting rewards, and seeing state transitions), allowing it to develop its understanding of the maze and find the best pathways to the objective.

Both humans and intelligent agents strive to discover the most efficient way to a goal. However, their techniques are different. Humans use their cognitive talents and heuristics to swiftly adapt to unfamiliar circumstances. Reinforcement learning agents, on the other hand, require training data to learn optimum behaviors in each environment, giving up some flexibility in exchange for the ability to handle complicated circumstances.

Within reinforcement learning frameworks, the intelligent agent makes a vital decision between exploitation and exploration. Exploitation is the agent's prioritizing of activities with high expected rewards based on its present understanding of the environment. Exploration, on the other hand, entails taking unique acts in order to identify possibly more rewarding avenues that were not previously evident.

In the context of the treasure hunt game, the agent initially gains by prioritizing exploration. This allows it to collect information about the surroundings, such as the maze's pattern and potential reward locations. As the agent interacts with its surroundings and learns from the rewards it gets, the optimal balance between exploration and exploitation evolves. This shift enables the agent to use its learned knowledge by selecting behaviors with larger predicted rewards, resulting in a more efficient path to the objective (i.e., the treasure).

Reinforcement learning (RL) provides a framework for the agent to learn optimal paths within the treasure hunt game. This approach leverages a reward mechanism, where actions leading the agent closer to the goal are positively reinforced, while those leading it astray are penalized. Through interaction with the environment, the agent receives these rewards and utilizes them to update its internal estimates of future rewards, often represented by Q-values. This iterative process of trial-and-error allows the agent to gradually learn which actions within specific states (e.g., different locations in the maze) contribute most significantly to the cumulative reward (i.e., reaching the treasure efficiently). Over time, the agent converges on a policy that prioritizes these high-rewarding actions, resulting in the discovery of the most efficient path to the treasure.

Deep Q-learning is implemented by utilizing neural networks to estimate the Q-values of state-action pairings. The procedure starts with creating a neural network with input layers representing state, hidden layers for processing, and output layers expressing Q-values for actions. A training loop is then developed, in which the agent interacts with the environment, chooses actions based on an epsilon-greedy strategy, earns rewards, and changes the neural network weights via backpropagation. To improve training stability, events are saved in a replay buffer and sampled in batches to disrupt correlations. Furthermore, a target network is employed to offer constant Q-value targets, which helps to stabilize the training process.

One problem was ensuring that the agent maintained a balance between exploration and exploitation. This was handled by progressively reducing epsilon to ensure early exploration followed by exploitation. Another problem was training stability, which was addressed utilizing experience replay and target networks to smooth out learning updates.

Deep Q-learning enables the agent to operate in enormous state spaces and learn sophisticated behaviors. The neural network's capacity to estimate Q-values allows the agent to generalize learning to comparable conditions, increasing its efficiency in locating the prize.

References

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

Zai, C. (2019). *Hands-On Deep Learning for Games: Leverage the power of neural networks and reinforcement learning to build intelligent games*. Packt Publishing Ltd.

Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., Van Den Driessche, G., ... & Hassabis, D. (2016). Mastering the game of Go with deep neural networks and tree search. *Nature*, 529(7587), 484-489.

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.