7-3 Project 2:

Design Defense

Richard Buono

Southern New Hampshire University

CS370 Current/Emerging Trends in Computer Science

Travis Smith

12/5/2023

7-3 Project 2: Design Defense

Solving an algorithm for a treasure hunt game that allows the pirate agent to travel through a maze to find treasure was completed by creating a Deep Q Network (DQN) in Python that drives the classes “*GameExperience*” and ”*TreasureMaze”*. In these classes, the pirate agent will be able to move in the basic directions of right, left, backward, and forward. The pirate agent is rewarded for reaching the treasure and penalized for being blocked, vising the same square, and trying an invalid move. Through reinforcement learning, the pirate agent was trained in optimal action for each state. The two categories that the strategies that were adopted for learning and fell into are exploration and exploitation.

# Exploration and Exploitation

Exploration in reinforcement learning denotes an approach of indicating actions that the agent has not yet exasperated, or that it has confidence in will lead to added information or knowledge that can improve its long-term performance. Exploration focuses on acquiring fresh information and knowledge that can contribute to enhanced decision-making and improved performance in the future. ( Yang, 2022)

Exploitation involves selecting actions that the agent anticipates will yield the highest immediate reward, drawing upon the knowledge it has acquired up to that point. Exploitation centers on optimizing the short-term performance of the agent through decision-making rooted in its existing knowledge.

Having the agent explore in the early attempts but gain more knowledge in later attempts using an algorithm to help with randomized wandering, and taking out the rate decay formula might decay too quickly or slowly depending on the number of epochs and the decay rate. Therefore, changing the algorithm to using the rand function. (Yang, 2022)

**Differences Between Human and Machine Approaches to Solving Problems**

**Since Artificial Intelligence (AI) relies on gained knowledge similar to how a human solves a maze, they start off slowly wandering around early to discover paths and then stringing that gained knowledge into useful ideas. Where the pirate agent and a human differ is forward more on visibility and intuition. The pirate agent cannot see beyond the current position and could walk right into an obstacle as a human would not, this is because reinforcement learning (RL) with DQN is model-free. (Botvinick, 2019)**

**Since the agent can be an “explicit” trial-and-error algorithm, the agent cannot think beyond or plot ahead of the current move while in the learning manner. This, however, does not pertain to a human, who could easily look ahead at many steps and decide that moving forward would result in a dead end and change the path before reaching such obstacle.**

**At the beginning of the learning process, exploration should be given more priority to allow the agent to explore different paths and learn about the environment. As the agent gains more experience and knowledge, the balance can shift towards exploitation to make use of the knowledge and achieve a much better performance. To achieve the balance between the strategies, decay rate formula was taken out of the equation. This allowed for a much lower epoch of random learning (trial-and-error) before experience-based predictions were used. Humans did not use more than 50 attempts, but if that visibility were restricted to a single move, which would be so hard to say in the end.**

***How can reinforcement learning help to determine the path to the goal (the treasure) by the agent (the pirate)?***

The pirate agent learns through actions, at each state the agent learned a policy that mapped that state to an action. The states correspond to the various locations on the map, and the actions correspond to the movements that the pirate can make to navigate the map. After each action, the agent is rewarded or punished with a goal of maximizing the total expected reward over time. As stated before, exploration and exploitation were used to define the actions available to choose within the states.

***How did you implement deep Q-learning using neural networks for this game?***

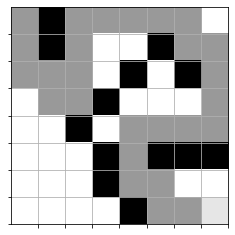
DQN was used to map the action-state pairs along with using equations to update the Q-values in the neural network based on the sampled mini batch of 20. With each action a reward or a punishment was determined for the agent. As the agent transitioned from exploration to exploitation, the rewards guided the agent on the next best move throughout the maze. The code below to the learning function and the predictions made by the agent:

***if*** *np****.****random****.****rand()* ***<*** *epsilon:*

*action* ***=*** *random****.****choice(valid\_actions)*

***else****:*

*action* ***=*** *np****.****argmax(experience****.****predict(prev\_envstate))*

The pirate agent started winning in the third epoch and had a 100%-win rate by epoch 58. The figure below is the path the pirate took to find the treasure:

**Figure 1: Final Path (80% shading) Taken by the Pirate Agent to Find the Treasure (50% shading)**

References

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>

Botvinick, M., Ritter, S., Wang, X. J., Kurth-Nelson, Z., Blundell, C., & Hassabis, D. (2019, May). *Reinforcement Learning, Fast and Slow* ScienceDirect. <https://www-sciencedirect-com.ezproxy.snhu.edu/science/article/pii/S1364661319300610?via%3Dihub>