Joan Estepan

CS-370 Current and Emerging Trends in CS

Project Two: Design Defense

February 15th, 2023

The main objective in the Treasure Hunt Game is for the pirate to find the treasure which is placed at the end of a maze. The pirate must find its way through the maze to reach the treasure successfully. For a human to solve this maze, most users will first scan the maze in search of creating a mental path that they can get the pirate to the maze, this method may be useful with simple mazes, but it may not be an optimal strategy if the maze reaches a certain level of complexity. Instead, the user might try different paths, discard the paths that lead to a dead end, and keep trying the next paths until they make it to the goal, basically a trial and error approach.

Our AI agent in the treasure hunt game learns the game by experience mainly through exploitation, but also by exploration. The AI agent will start by trying random paths to try to get the treasure, every try by the AI is what is considered an episode. The agent will run the episode, and either win or lose the game, then it will save the state of the episode and remember it for the next episode. The agent will learn from every episode and take the steps from the path that he knows from experience that will lead to a good outcome, this is what we call exploitation. However, every once in a while, the agent will try a new path to see if it can find an even more efficient path than before, this is what we call exploration. The number of times that the agent will try a new path is decided by the epsilon, which we can set in our code.

The way the agent and the humans solve the issues are very similar. As mentioned earlier, the second approach that humans take to solve the problem. Involves trying different paths and remembering which paths lead to a good outcome. Which is the same approach that the AI agent takes in the pathfinding algorithm.

Exploitation is when the agent utilizes the tactic of carrying out known to be beneficial acts based on the information that has previously been obtained. Instead of doing new things, exploitative behavior focuses on increasing short-term rewards. For example, a robot that has previously discovered a good route to its destination may continue to follow that route instead of trying a new path that could be more efficient.

Exploration, on the other hand, is when the agent uses an approach that consists of performing random acts to learn more about the environment around it. Though this approach might not look ideal, in the long run, this data could be extremely helpful for the agent to find its path. Both of these approaches are very beneficial when used the right way. To get the most efficient route, the algorithm must have the right balance of exploration and exploitation. For this pathfinding algorithm, I would say the ideal exploration rate would be around 15%. This would allow the agent to search for alternative routes while also taking determined steps toward the goal most of the time.

Reinforcement learning is a potent method for teaching agents to find their way through complex settings. The agent can identify the best route to the objective by learning from experience and tailoring its behavior to the particulars of the environment. This is one of the best methods for pathfinding algorithms.

To implement the Q learning in this game, first, we define our variables such as the number of episodes (n\_epoch), the max memory, the data size, the start\_time, etc. Then, we constructed the environment from our numpy array in the “qmaze” variable, initialized an experience replay object, and set the win history array, the size of the maze, and the win rate. After all, these variables were declared, we set a loop that runs according to the number of episodes, in this loop the agent is placed on the maze and will make random moves looking for the treasure unless it had previously learned a viable move before. At the bottom of the loop, the algorithm will save the experience data from the current episode, increase the number of played episodes and repeat the loop until either it reaches the number of episodes or the win percentage rate reaches 100%.

References

Jing), J. H. (2020, May 4). *Striking a Balance between Exploring and Exploiting*. Medium. https://towardsdatascience.com/striking-a-balance-between-exploring-and-exploiting-5475d9c1e66e

Makone, A. (2021, August 22). *Reinforcement Learning 6: Exploration vs Exploitation*. Medium. https://ashutoshmakone.medium.com/reinforcement-learning-5-exploration-vs-exploitation-c1bae5a2ea42

*Exploitation and Exploration in Machine Learning - Javatpoint*. (n.d.). www.javatpoint.com. https://www.javatpoint.com/exploitation-and-exploration-in-machine-learning