Project Two

Steven Anderson

When solving the treasure hunt game there is a very different methods enacted by humans and machines. Assuming that a human player doesn’t know how the maze looks like and is just moving up, down, left or right to solve the maze they would start each turn by making a move in a direction. Then, remembering what moves they have made before, would move in a direction. If a human user got to a point where only one way is available the human would test all three of the other options before moving back. They would also not move back the way they came unless they ruled out other possible avenues of progression. So, a human would in most cases move within the range of a certain number of moves to solve ranging from a perfect game to the max amount of moves a player could get wrong while still progressing forward through the maze.

This is very different from how the intelligent agent works through the maze since the intelligent agent has a level or randomness applied to its learning. The agent may make the mistake of moving back after moving forward based off random chance given to it at the beginning of every move. For example, the agent could be in a position where the only moves that don’t hit a wall are up or down, it could have moved into that position from below and then due to the randomness applied to its moves move back down instead of moving up, backtracking its progression and leading to it taking longer or possibly not solving the maze. This randomness also means that the agent’s possible move count to solve the maze ranges from a perfect game to completely unsolved or infinite moves possibility. There is no guarantee that the agent will solve the maze since it doesn’t focus on remembering all its previous moves before it makes a new one.

While these two methods are very different to each other there is some level of similarity in their moves as well. An agent and a human can both be wrong about which direction to go and start its next move stuck at its previous position. They also both are constrained by their limited move options. They will also have plenty or trail and error before progressing the right direction when moving in areas with limited possible progressing directions.

In pathfinding for our intelligent agent there is two methods of progression. Exploitation, which is the use of known information to prioritize short-term gains in progression, and exploration which is focused more on long-term goals through experimenting (Survey Point Team, 2024). These two are also intertwined when it comes to solving the treasure hunt maze. Without having a level of exploitation, the intelligent agent would constantly be making random moves with no sense of real direction, but without having exploration the program would have a hard time backtracking or figuring out new possibilities for progression. When applied to learning intelligent agents in a broader sense outside of the scope of the maze game having both are important to create a robust system that can learn. A program that doesn’t use enough exploitation methods in its will have a hard time remembering previous inputs or iterations. While not having enough exploration can lead a system into becoming stagnant with little option into finding new possibilities (Survey Point Team, 2024).

In my view the ideal proportion for exploitation and exploration would be more exploitation and less exploration at a factor of maybe 75% to 25% split or 60% to 40% split. This is because be able to process what the system needs to learn should take some level of priority over exploring new possibilities. If a system cannot learn well enough from the known data inputs it can lead to a program that fails at its explicit purpose but without the bit of exploration it wouldn’t be able to adapt with the information it has. Like with the AlphaGo Zero program it relies more on exploitation to remember move combinations and patterns than it does with exploration to develop new moves since it is trying to learn the combinations. And with the small amount of exploration, it is still able to over the thousands of self-play games develop new moves and combinations to add to its dataset.

In the treasure hunt maze game reinforcement learning can be applied to help it reach its goal of finding the treasure. With exploitation it is able to take its immediate moves and process possible paths that it can take. Then using exploration give it enough randomness to avoid getting stuck and able to find possible new paths that its known moves wouldn’t be able to find. Like being able to go in another direction on a branching path after extinguishing one route as a dead end.

In my project I utilized Q-learning techniques in neural networking to teach it how to solve the maze. I used saved episodes for the program to use in order to save that in its experience which can be used to determine its next move. This is a form of exploitation that helps it learn from previous moves. This also involved using observe() to have the program look at the current state of the game board to see where it is. I also utilized exploration when I had the agent use the random choice option when moving to make sure that it wouldn’t get stuck while trying to solve the maze. Together the random action and the saved history and experience values were used when calling the action predict to help the program to generate predictions for its new moves both with a level of randomness and utilizing saved history to make that decision.

**References:**

Survey Point Team. “Exploitation vs Exploration in Machine Learning: All You Need to Know.” *Surveypoint.ai*, Survey Point, 7 Feb. 2024, surveypoint.ai/blog/2024/02/07/exploitation-vs-exploration-in-machine-learning-all-you-need-to-know/.