Project 2 Essay

The Pirate Treasure hunt game is a pathfinding reinforcement learning mechanism that gives clear evidence to the progress of training within a machine. While humans and machines can have similarities in the way that we solve things, like this pirate pathfinder game, there can be distinct differences in strategies and adaptability. For example, a human would be more prone to rely on their visual and prior knowledge or memory throughout this game. We would try to remember what paths we had already gone down and remember visually where to avoid, on a trial-and-error basis. Machines, on the other hand, would rely more on algorithms aiming to achieve the best results and gain the rewards. Using the Deep Q-Learning method, the machine is very quickly able to obtain perfect results consistently by using exploitation, which allows it to use knowledge for its moves instead of them being random. By remembering every move it makes, it is quickly able to perfect the game. The example I created for this project shows that the machine was able to reach a one hundred percent win rate at epoch 432, which took only eighteen minutes to perfect the game.

The purpose of the intelligent agent in pathfinding is dependent on how you want it to act. If you want a quick result that is solely based on trial and error and remembering moves to achieve the maximum number of rewards, then that is exploitation. This is the equivalent of a human gamer trying to “speedrun” a game. If you want the agent to take random actions and explore the environment and discover new areas, then that is exploration. The agent in this has the objective of trying to explore all paths possible. Therefore, the time this can take to train can be longer. In the terms of this game, the pirate is trying to gain all the treasure on the map regardless of the time it takes to do so. I believe that a good mix of these would be to start with exploration, where the machine is looking for the treasure and exploring the map a little, and then lean more towards an exploitation stage where he starts to try to win quicker. This would ensure that while most of the places on the map have been explored, the time is still relatively short in terms of the length of the game. The Q-Learning method is a lot quicker than I thought it would be, as this is my first time doing work with machine learning. I expected even things like this to take many hours, but sometimes they can take 10 minutes. Giving it more of a exploration side at the beginning also allows it to act more human, as we take random routes and use our memory to avoid those areas. This can make the results more interesting and human-like.

I implemented deep Q-Learning using neural networks for this game by allowing the agent to find the Q-value for each action in each state. This consisted of input, hidden, and output layers. The input layer was the maze state. The hidden layers were processing the states and learning patterns. Lastly, the output layer is the one doing the actions of right, up, left or down for the agent. By being able to remember every action it makes, the machine using exploitation can achieve victory very quickly. This makes it a viable option for a quick and effective reward system training method. Every reward, or treasure, the agent gets it stored in its memory, and it will use that knowledge to perfect the game quickly. It was able to achieve consistent one hundred percent games within 10 minutes of it first learning the game.

This shows that the Deep Q-Learning algorithm is highly effective in achieving quick results. It can do this by rewarding the agent and allowing it to remember every move that it makes. While this can be fun, using exploration can also be interesting too. It can achieve must more precisely results, but at a much slower rate. Either option, the results this technique can achieve it a very interesting thing to see.

Citations

*Deep Q-Learning*. (2019, June 13). GeeksforGeeks. https://www.geeksforgeeks.org/deep-q-learning/

‌

Wang, M. (2021, February 1). *Deep Q-Learning Tutorial: minDQN*. Medium. https://towardsdatascience.com/deep-q-learning-tutorial-mindqn-2a4c855abffc

‌