Project Two

Jeremia Faust

Doctor Obafemi Balogun

August 13, 2022

**Treasure Hunt Eight by Eight Maze**

A black and white checkered background

Description automatically generated with low confidence

**Differences between Human and Machine approaches to Solving Problems**

The treasure hunt game is an eight-by-eight matrix where with a path that the agent has to find without previous knowledge. If a human were to solve this maze, they would first take a look three hundred degrees around to get their bearing. They would notice that there is one way out of the corner which is down. Human would continue to look around as they are working. Depending on conditions a human can look beyond the immediate area and could see that down leads to a dead end thus from the viewpoint of the human would turn left. A human can do this process quickly and complete the maze in a few seconds.

An intelligent agent using the Q-learning algorithm looks at each step with the ability to move up, down, left, and right. At each step it looks around itself to find the best path based on a reward system and is rewarded based on if was a good move or not. This is mostly done randomly and after each episode it can make better choices. In the case of the treasure hunt it took 35 episodes, 262 epochs, and 393 seconds to solve the maze at 100%.

A human can solve this maze fairly quickly because they are not limited and have faster reasoning skills over an agent. An agent has to methodically go through each step to find the treasure. While each takes in data of their own and make choices, a human can look much further and can decide what looks like a dead end lets go this way. Yes, a human can make a wrong choice but can recover faster. An agent has to go back to the beginning and make better choices based on the data it received.

**Purpose of the Intelligent Agent in Pathfinding**

Reinforcement learning is the act of taking actions to maximize learning. This is done with a method called exploration and exploitation. This is a method called epsilon greedy strategy which according to Baeldung is “a well-known reinforcement learning algorithm.” There are a few state values that programs the parameters that the agent follows. Gamma is the value where the agent considers the reward. With a higher value the agent looks for higher rewards. The learning rate or Alpha is the value that “defines the learning rate or step size”( Baeldung,2021). The lower the value the less the agent learns per step. On the other hand, if the value is too high the agent will not use prior knowledge and only use current information. The exploration, min max, and decay are the values use to balance exploration to exploitation. As the agent learns there is less needed to explore so it exploits that knowledge. That data set is a summary of agents experience at a certain time. It stores the data in what’s called replay memory and is randomly pulled from the replay memory. This breaks any correlation thus providing better learning. The discount factor scales down the rewards after each step. According to Sajil C. K it “adjusts the agent’s behavior to achieve long term goals”. Neural networks are the key component for Q deep learning, it helps the agent to recognize patterns. It provides a “Neural network model that can predict the action for a given state in a game with better accuracy”(Chandrakant,2020). In the case of the treasure game it maximizes performance by predicting the best path with a much higher accuracy. One thing to note is that too much reinforcement will diminish the result, so make sure you dial in the values for greater results.

**Use of Algorithms to Solve Complex Problems**

For the treasure can we had to implement Q-Learning using the given code and pseudocode. The first step was to study the code. watch the walkthrough and read all the documentation. The pseudocode was the key to the entire assignment. It gave us a path to complete the deep Q- Learning. The previous assignment of the deep learning with the cartpole was also used to guide me in how to implement the code in the treasure hunt. The first thing that was important was to figure out all the variables that were used with the rest of the program. This allowed me to understand how everything worked making it easier to implement deep learning. I set up everything I could just using the pseudocode. I was able to get ideas from other methods such as the play game method. This was probably the most difficult to figure out but is how I implemented the if else statements and append the win history and game state. There was a lot if trial and error and was able to get working code and match the sample output.

References

Baeldung. (2021, January 15). *Epsilon-Greedy Q-Learning*. Baeldung on Computer Science. Retrieved August 13, 2022, from https://www.baeldung.com/cs/epsilon-greedy-q-learning

Chandrakant, K. (2022, June 20). *Reinforcement learning with neural network*. Baeldung on Computer Science. Retrieved August 13, 2022, from https://www.baeldung.com/cs/reinforcement-learning-neural-network

Düğmeci, D. (2021, November 10). *Finding shortest path using Q-learning algorithm*. Medium. Retrieved August 13, 2022, from https://towardsdatascience.com/finding-shortest-path-using-q-learning-algorithm-1c1f39e89505

freeCodeCamp.org. (2020, April 7). *A brief introduction to reinforcement learning*. freeCodeCamp.org. Retrieved August 13, 2022, from https://www.freecodecamp.org/news/a-brief-introduction-to-reinforcement-learning-7799af5840db

Lamba, A. (2018, September 3). *An introduction to Q-learning: Reinforcement learning*. Medium. Retrieved August 13, 2022, from https://medium.com/free-code-camp/an-introduction-to-q-learning-reinforcement-learning-14ac0b4493cc

NancyJemimah. (2020, April 24). *Level up - understanding Q learning*. Medium. Retrieved August 13, 2022, from https://medium.com/@nancyjemi/level-up-understanding-q-learning-cf739867eb1d

.