Renaldo Musto

Project 2 - Design Defense

The purpose of this maze is to be able to find the hidden treasure starting from one point in the map and traversing through the free cells to get to the treasure. If I were playing the game I would first look at the whole board and see what potential routes were available to me. I would imagine the path I would take and run a few simulations quickly in my mind to see which one should work best then I would actually make my move through the maze once I visualized the correct path and tested it in my own mind. This on a fundamental level is very similar to how the AI trains and decides what paths to take. When using a Q-learning algorithm like we have here the AI responds to the game through observing various states and rewards then updating an overall policy based on those results in order to inform the AI on what actions to take (Dugmeci, 2021). Before the AI starts to play the game things like discount factor, reward value and learning rate are set so that the AI has base values to work with and we also need to be able to tell the AI what is a good and bad move based on the reward value. As the AI plays the game it will initially make seemingly random choices and then see the rewards it got for playing that way. Based on these rewards it will adjust its policy or strategy and then play again hopefully getting a higher reward each time. The goal of the AI is to get the highest reward value possible and it will keep playing and adjusting based on that value until it reaches an acceptable win rate (Dugmeci, 2021).

I see quite a bit of similarity between the human and AI approach to the issue, the main difference I see is the amount of background knowledge I have versus the AI. I have done things like this before so I can call on my previous knowledge to help me but the AI must learn fundamentally what it means to play the game and what even hitting a wall will do to the player. Overall though I think that this works effectively the same way as a human if this were a game I had no familiarity with at all I would likely take a similar approach to the AI. Taking guess actions at first then seeing how that went and then slowly refining my guesses as I get better results and notice patterns throughout the game.

The difference between exploration and exploitation is how the AI decides what move it will make next. Exploration is when the AI makes a random move and exploitation is when the AI makes a move based on its past experience. These are both necessary because if we only take random actions we will never be able to apply what we have learned through playing the game and if we never take random action we could get stuck doing the same thing over and over and never trying to find a new solution to the problem. The key with exploration and exploitation is to find a balance between trying out new things and working with what we already know (Yang, 2022). In this project for the treasure hunt game I chose to have the exploration to exploitation ratio be 1 to 10 so for every 10 exploitative actions we take one exploration action. I think that this is ideal because I was able to achieve a 100% win rate this way in about 150 epochs while any slight change I seemed to make dramatically changed my results for the worse. It seems like if we have more exploration the AI can never really use what it has learned properly because its strategies get cut short by the random action and if we have more exploitation the AI has a hard time completing the maze because it gets stuck in loops of trying the same thing over and over. Reinforcement learning is great at helping the AI solve the maze because it is able to learn from past gameplay and therefore improve over time with each new game (Lamba, 2018).

Deep Q-learning and neural networks worked together in a fundamental way to make the solving of this maze by the AI possible. The way this was implemented was by having the neural network replace the Q-table that is typically present; this is what separates Q-learning from deep Q-learning. We use an epsilon-greedy exploration strategy to find the next move for the AI and then pass this to the neural network so it can update weights based on the rewards that were given that turn. We also use two neural networks in deep Q-learning to help stabilize the results in the learning process (Singh, 2022).

References

Dugmeci, D. (2021, November 10). *Finding shortest path using Q-learning algorithm*. Medium. Retrieved December 11, 2022, from https://towardsdatascience.com/finding-shortest-path-using-q-learning-algorithm-1c1f39e89505

Lamba, A. (2018, September 3). *An introduction to Q-learning: Reinforcement learning*. Medium. Retrieved December 11, 2022, from https://medium.com/free-code-camp/an-introduction-to-q-learning-reinforcement-learning-14ac0b4493cc

Singh, S. (2022, October 7). *A comprehensive guide to neural networks in Deep Q-Learning*. A Comprehensive Guide to Neural Networks in Deep Q-learning. Retrieved December 11, 2022, from https://www.turing.com/kb/how-are-neural-networks-used-in-deep-q-learning

Yang, A. (2022, July 25). *What is exploration vs. exploitation in reinforcement learning?* Medium. Retrieved December 11, 2022, from https://medium.com/@angelina.yang/what-is-exploration-vs-exploitation-in-reinforcement-learning-a3b96dcc9503