Matthew Trembley

10/16/2022

Southern New Hampshire University

CS-370

Project Two: Design Defense

The overall goal of this game was to create an AI agent that would find “treasure” within a maze at a certain coordinate at a certain success rate. A human would solve this situation in a trial-and-error basis. If they hit a dead end, turn around and try again until they find the reward or exit. Depending on the complexity of the maze, it would seem that a human could either get out rather easily or be stuck in the maze indefinitely. An AI on the other hand would have a very methodical or algorithmic way of solving the way. AI would set a value to each step closer to the treasure with reward, while taking away from the reward severely for impossible moves (like jumping straight to the treasure). In this way, the agent will continue to take moves that provide a greater reward than those that don’t. This will allow the agent to understand their position and how great their next reward will be depending on the next move. “…an agent takes current state, picks best (based on model prediction) action and executes it on an environment. Subsequently, environment returns a reward for a given action, a new state and an information if the new state is terminal. The process repeats until termination” (Cartpole – Introduction to Reinforcement Learning, 2018). The main difference between human and machine processes can simply be put as emotional reward. If a human were in a maze, each dead end would be detrimental to the emotional wellbeing of the person. A machine will just see it as a number, which in a certain way may be more productive. This difference does touch base with the same similarity: reward. A human that is quickly progressing through the maze will feel more confident in their surroundings and sense of direction, an emotional reward so to speak. Although AI does not emote, the AI agent will continually gravitate towards high reward moves until the maze is solved.

The purpose of an intelligent agent in pathfinding is to quickly find the most efficient path to the end. An intelligent agent will use the methods of exploration and exploitation to help it in its journey to the end of a maze. Exploration is the use of trial and error to learn about the world. Exploitation is the act of explicitly choosing the best-known action at a state (Deep Q-Learning Tutorial, 2020). In this particular problem, there was a balance between the trial and error needed to learn the maze, and an amount of exploitation to find the best route to the treasure. At the beginning of the algorithm, a lot of exploration is needed for the agent to learn about the maze it is in. As the algorithm progresses in output, that exploration will dissipate as the agent will have an understanding of its surroundings, and exploitation will increase to take the maximum reward. Reinforcement learning can help determine the path of the goal by the agent by training the agent repetitively on the world it is set in and rewarding the agent for each positive move. As the agent is rewarded for each move towards the goal, it will exploit the best possible next move to continually receive a reward.

The implementation of deep Q-learning using neural networks was tricky at first, but once the building blocks were there it just took minor tweaks to obtain correct output. First, a loop was created for the game to train the agent, which started at a random space in the maze. For each move the agent needed to make, the reward was compared to *epsilon*. If the action reward was less than epsilon, the next action to take was random. If it was greater than epsilon, the next action to take was highest yielding reward move based on past experiences. This refers to exploration/exploitation. The agent was coded to “remember” past experiences to help train faster in the maze. This allowed the agent to quickly find its way to the treasure at a high percentage success rate. The Q value was continually updated as the future maximum reward would change depending on the action the agent took. The use of neural networks in this treasure hunting problem was ideal, and within a short amount of time the agent was able to win at a near 100% win rate.

References

Surma, G. (2019, November 10). *Cartpole - introduction to reinforcement learning (DQN - deep Q-learning)*. Medium. Retrieved October 11, 2022, from https://gsurma.medium.com/cartpole-introduction-to-reinforcement-learning-ed0eb5b58288

Wang, M. (2021, October 3). *Deep Q-learning tutorial: Mindqn*. Medium. Retrieved October 11, 2022, from https://towardsdatascience.com/deep-q-learning-tutorial-mindqn-2a4c855abffc#:~:text=Deep%20Q%2DLearning%20uses%20Experience,to%20train%20after%20each%20step.