Tanner Hunt

October 15, 2025

CS370 Current/ Emerging Trends in CS

Project Two Design Defense

This is a neural network designed for a pirate game that navigates a maze towards a treasure. The player then tries to navigate towards the treasure before the pirate can reach it. For small maze sizes, a human might solve this problem using a “flood fill” -like algorithm. In this approach, the player takes the most direct approach to the end of the maze until an obstacle is reached, then they update their intended path. In this way, they can quickly trace out the path they are going to take with just their eyes. Human players may also identify important intermediary points they need to pass through to reach the end of the maze. In that case, navigating the maze becomes a recursive problem. The neural network designed for this problem, in contrast, uses its past experiences to shape its decisions. After every move, the intelligent agent looks at the maze, remembers what choices it made from a similar position, then decides based on how well previous decisions played out. These two approaches to navigating a maze are significantly different. The human approach to small mazes is based on an algorithm – a sequence of steps to take based on a rule set. The agent’s approach is based on past experiences it's had navigating a maze. Humans may take this experiential approach in a larger maze that can’t easily be navigated. This would entail exploring different avenues within the maze, then deciding based on what pathways seem the most promising.

This neural network uses past experiences to make judgements about what actions to take next. This raises an important question. When should the neural network experiment – and therefore gain new experiences – and when should it make decisions from its past experiences. This question is answered through the exploration-exploitation algorithm. In this algorithm, the AI agent makes random decisions early in training to gain a breadth of knowledge. This is the exploration phase of training. Then, after the agent has gained an understanding of the environment, it makes decisions on what it thinks is best to fine tune its decision making. This is the exploitation phase of training (Exploration and Exploitation, 2025). A well-designed approach to training this neural network would then involve making more random decisions early in training, then slowly transitioning to exploiting the learned patterns in the neural net. Specifically, the neural network should have an exploration coefficient of near 100% early in training, because every possible action the network could take explores new state space. Then, reduce the number of random decisions the network makes by a small amount – say 5% -- after each training epoch. This is done because it is unfeasible to explore the entire state-action space of an 8x8 grid. If each grid piece has 5 different states it can be in (empty, wall, explored, pirate, treasure), and each tile has 4 possible moves (left, right, up, down), then there are (64\*5)! \* 4 different possible tuples that could be explored. This decay rate should allow the neural network to fine-tune its approach based on its trained data. Reinforcement learning can help the agent determine the path to the goal because it is rewarded when it makes actions that avoid obstacles and advances it towards the goal. The agent then indirectly learns a policy about how to navigate mazes by predicting how it will be rewarded based on its current position, available actions, and possible future positions and actions.

Because the state space for this problem is so large, this neural network implements deep-Q learning. In this algorithms implementation, rather than recording a verbose table of rewards associated with very action the neural network has taken, a neural network was trained on the experiences (states, actions, and rewards) of the neural network. This allowed the network to predict the reward from future actions without having directly experienced it. The training algorithm takes the following steps: choose a random starting point in the maze. Then, the agent is given 150 actions to complete the maze. If the agent is within the exploration phase of training, it is more prone to choosing random actions. If it is in the exploitation phase of training, it will predict the best action to take next from the current state of the board. Each experience (action, state, and reward) is stored in a buffer. At the end of the game, the neural network is updated based on the experiences in the buffer. As the neural network is trained, it makes better predictions about the reward of the actions it takes. When the model can correctly solve the maze 32 times in a row, the model has completed its training.

References:

Exploitation and Exploration in Machine Learning. (2025). Geeks for Geeks. <https://www.geeksforgeeks.org/machine-learning/exploitation-and-exploration-in-machine-learning/>