# 

# CS 370 7-3: Design Defense

Raymond Aponte

raymond.aponte@snhu.edu

Southern New Hampshire University

**Human vs. Machine Intelligence**

To describe how a human would solve this maze as opposed to my intelligent agent, it is important to note that humans and machines learn in different ways. One difference is that artificial neural networks need a lot of data to learn whereas humans can pick up on things a lot easier and a lot faster (Fernandez, 2019). For instance, a human might take a glance at the maze, try to determine the right path, and start navigating from start to finish. If that person believes they made a mistake, they can trace back where they came from and go down another route. A person would not even think about minuscule rewards, just completing the maze. For the agent to solve the pathfinding problem, its moves are reward-based. Positive or negative points are rewarded based on optimal or suboptimal moves. For example, the greatest reward is achieved by reaching the treasure, and penalties are given if the agent moves backward, hits an occupied cell, or attempts to move outside the maze boundaries. For a human, going outside the maze boundaries would most likely not even be considered and going backwards, although it could be considered “cheating,” would not be as detrimental. For the intelligent agent, the goal is to achieve the greatest discounted reward which means every move must have a purpose to win as fast as possible achieving the greatest award with the least mistakes. Also, the intelligent agent will train on the maze multiple times in “episodes” and save each episode to learn from its past moves. Humans, on the other hand, would most likely finish the maze in very few attempts, if not one.

**Purpose of Intelligent Agent**

Exploration revolves around finding out new information about the environment while exploitation is about going back to current/known information to maximize those rewards. However, there may be larger rewards that could be missed if the agent is programmed to exploit current rewards much more than explore for future rewards (ADL, 2018). For this pathfinding problem, the exploration factor, or epsilon, was set to 0.1 meaning that 90% of the time, the intelligent agent would exploit known paths/reward values. However, it was still able to explore 10% of the time to discover new paths that could bring better rewards. This is important because each episode consists of states that exist between an initial state and terminal state; these episodes are stored meaning the agent can exploit known values of high reward. Also, if the win rate of the agent was over 90%, the epsilon would drop to 0.05 as the agent would have much less need to explore. If the exploration factor were higher from the start, the agent would need much more time to complete the game and it may not have achieved a 100%-win rate. As a matter of fact, the agent may have not been able to win at all because it would not take advantage of the most beneficial rewards/paths. An epsilon of 0.1 and 15,000 epochs were optimal for this algorithm.

Reinforcement learning can help to determine the path to the goal by the agent. Reinforcement learning can be defined as “an aspect of Machine learning where an agent learns to behave in an environment, by performing certain actions and observing the rewards (…) from those actions,” (ADL, 2018). Within a reinforcement learning, there is a type of task called an episodic task that has a starting state (initial state) and terminal state. In each episode, there are states, actions, and rewards. In addition to this, there is the exploitation and exploration trade-off (ADL, 2018). The agent in this pathfinding problem randomly selects a cell at the start of each epoch and after playing many games/episodes within that epoch, the agent was able to improve itself through recording each episode and varying exploitation and exploration (exploration is 0.1 and tapers off 0.05 as the win rate exceeds 0.9) as it builds off each episode’s results.

**Evaluating Algorithms**

For this game, I implemented deep Q-learning using neural networks. The neural net was already provided and included three dense layers, two PReLU layers, an MSE loss function, and Adam optimizer. I was tasked with creating the Q-training algorithm. The goal of deep Q-learning is for the agent to choose the best action given current circumstances/observation and each action for each observation has a Q value; an accurate Q value is determined by remembering states and performing an experience replay (Surma, 2018). Experience replay uniformly samples experiences that are remembered and updates the Q value per entry (Surma, 2018). In this game, the agent randomly selected a free cell to start per epoch. For each episode within the epoch, the epsilon value determined whether the agent should make an action based on exploitation or exploration. All states, actions, and rewards were stored, the neural network was trained using this data, loss was evaluated, and Q values were updated until the agent could maximize rewards and achieve a win rate of 100%. If the agent/pirate won the game, the win rate was updated. Otherwise, the loss was recorded and updated. The test was completed when the win rate was greater than the epsilon.

**References**

ADL. (2018, April 7). A brief introduction to reinforcement learning. freeCodeCamp.org. Retrieved August 20, 2022, from https://www.freecodecamp.org/news/a-brief-introduction-to-reinforcement-learning-7799af5840db

Fernandez, E. (2019, November 30). *Ai is not similar to human intelligence. thinking so could be dangerous*. Forbes. Retrieved July 8, 2022, from https://www.forbes.com/sites/fernandezelizabeth/2019/11/30/ai-is-not-similar-to-human-intelligence-thinking-so-could-be-dangerous/?sh=3647169b6c22

Surma, G. (2018, September 6). *Cartpole - Introduction to reinforcement learning (DQN - deep Q-learning)*. Medium. Retrieved August 6, 2022, from https://gsurma.medium.com/cartpole-introduction-to-reinforcement-learning-ed0eb5b58288