For Project Two, I was tasked with designing a solution for a pirate maze problem. It was expected to develop a Deep Q-Network (DWN) in Python, which had code provided for the GameExperience & TreasureMaze classes. The pirate’s direction was determined by the four cardinal directions tied to numerical. The pirate would earn a reward for reaching the treasure & be penalized for hitting obstacles, repeat trips to a square, & attempting invalid move. Through reinforcement learning, the agent began to optimize its actions via trial & error while training its strategies.

Exploration involves selecting unfamiliar actions to gather new information, improving long-term performance. **The beginning of the learning process involves exploring different paths & learning of the environment. As experience is gained and it “learns”, its balance begins to shift to make use of the knowledge & alter its path before reaching the obstacle. This is known as exploitation or maximizing immediate rewards based on existing knowledge.**

*Α =1 + epoch × decayRate1 ​< ε*

This formula is applied to the agent to train its rate of decay, shifting from exploration to experienced based decision making. Unlike human brains & our learning, agents do not have foresight & must rely upon DQN to learn optimal paths.

At the start of the learning process, exploration is prioritized to allow the agent to discover various paths and understand the environment. As the agent gains experience, the focus shifts toward exploitation to maximize performance using the knowledge acquired. To achieve this balance, the decay rate was adjusted, ultimately settling on a value of 0.1 for an epsilon (ε) of 0.1. This allowed for 90 epochs of random exploration before transitioning to experience-based decision-making. While a human might solve a simple maze in fewer attempts due to their ability to plan ahead, the agent’s restricted visibility (limited to one move at a time) makes this comparison less straightforward.

While the agent trains by exploring the environment & updating its action-values function, it gains considerable feedback. The experience helps to build optimal values, & the new learned policies become more accurate & efficient at finding the optimal path to the treasure.

The DWN was enforced map out action-state pairs. Each action is either a reward or a punishment suing the code below.

*def get\_reward(self):*

*pirate\_row, pirate\_col, mode = self.state*

*nrows, ncols = self.maze.shape*

*if pirate\_row == nrows-1 and pirate\_col == ncols-1:*

*return 1.0*

*if mode == 'blocked':*

*return self.min\_reward - 1*

*if (pirate\_row, pirate\_col) in self.visited:*

*return -0.25*

*if mode == 'invalid':*

*return -0.75*

*if mode == 'valid':*

*return -0.04*

As the agent made the transition between exploration & exploitation, these rewards guide the agent on the next best possible move. The decision for the agent is based on a decaying learning function. The formula for learning rate decay function is listed below.

*if 1 / (1 + epoch \* decayRate) < epsilon:*

*Action = np.argmax(experience.predict(previous\_envstate)) #Exploitation*

*else:*

*Action = random.choice(validAction )#randomily pick from the validActions #Exploration*

My agent begins winning in its 2nd epoch, reaching 100% by epoch 419. It’s final path is pictured below.

A black and white crossword puzzle

AI-generated content may be incorrect.

*Model-free (reinforcement learning). (2023, March 3). In Wikipedia. https://en.wikipedia.org/w/index.php?title=Model-free\_(reinforcement\_learning)&oldid =1142602733*

*Haswani, V. (2020, September 3). Learning rate decay and methods in deep learning. Medium. https://medium.com/analytics-vidhya/learning-rate-decay-and-methods-in-deep-learning-2cee564f910b*

*Yang, A. (2022, July 24). What is exploration vs. exploitation in reinforcement learning? Medium. https://angelina-yang.medium.com/what-is-exploration-vs-exploitation-in-reinforcement-learning-a3b96dcc9503*