**Analyze the differences between human and machine approaches to solving problems.**

Humans, speaking from experience, start by considering possible solutions, and evaluating what the correct approach might be based on intuition and past experience. By laying out all the options, we can infer the correct path forwards by ruling out bad options, and weighing the choices against each other, akin to balancing a scale. We can see similarities in this process in machine learning- as after all, it is largely intended to mimic the process that we use. Humans didn’t get to be the apex of the earth by making the wrong decisions, at least most of the time; as such, machine learning also follows a step by step breakdown approach to problem solving.

There are a number of interweaving systems involved in a machine’s problem solving approach, relying on tracking past experiences, exploration, and experimentation (effectively similar to us weighing pros and cons), which eventually can lead to a hyper efficient predictive / learned algorithm of steps to take towards completion. Trial and error is a common element, progressively iterating on previous generations in order to reach a better result over time, learning from past mistakes.

**Describe the steps a human being would take to solve this maze.**

A human being might start with a previously learned strategy, such as trying to move in one specific direction, navigating around obstacles. They might also try random pathing, or perhaps they’re more clever, and have a method such as following along the sides of a wall to traverse the entire maze. Depending on the maze’s construction and layout, any one of these strategies may or may not work. Whatever the case, we have the intuition to try again, or to try something new and different each time. Eventually, they will rule out (supposing they don’t forget), all the options for pathing and find the end of the maze.

**Describe the steps your intelligent agent is taking to solve this pathfinding problem / What are the similarities and differences between these two approaches?**

In the case of this assignment, the intelligent agent begins by selecting free cells, evaluating the state of its environment, and choosing either to explore or to exploit its existing knowledge from past iterations. It then moves depending on this status and repeats this process until it succeeds in finding the end. This is similar to the approach that a human might take if they were to re-trace their steps, however, they might have to evaluate how close they are to the end, or how “successful” their past attempts were. One might hope that the human does not get stuck in an infinite loop banging their head against a wall. All though I did not have time to tweak the agent so that it always succeeds, the agent did progressively get closer each iteration to the end, but perhaps I will need to adjust its exploration and exploitation rates so as to encourage it to not get stuck in corners / loops.

**Assess the purpose of the intelligent agent in pathfinding.**

The intelligent agent is the part of the machine that navigates the path, being fed data and then performing actions based on its environment and conditions. Depending on its parameters, it may prioritize new strategies / pathways, or continue on the same road that it has walked before. Supposing that its parameters are balanced well, it should eventually come up with a path forwards that leads to the end efficiently.

**What is the difference between exploitation and exploration? What is the ideal proportion of exploitation and exploration for this pathfinding problem? Explain your reasoning.**

Exploration and exploitation are both important aspects of the pathfinding- one is akin to trying new things, ie, exploring, and the other is the process by which the intelligent agent learns from its past experiences, ie “exploiting” its knowledge. As I have come to realize through trial and error, there is an optimal amount of exploration and exploitation. If you have too much of exploration, then your agent might not ever capitalize on rewards, or it may get stuck, making silly decisions without learning from its mistakes. With too much exploitation, the agent may not be willing to lose any space, nor backtrack in order to progress further. Perhaps the agent is stuck in a corner due to a lack of exploration / hyperfixation on exploitation, or maybe it has no direction or greater purpose due to a lack of exploitation.

**How can reinforcement learning help to determine the path to the goal (the treasure) by the agent (the pirate)? Evaluate the use of algorithms to solve complex problems.**

Reinforcement learning, keeping in line with its namesake, attempts to reinforce neural pathways that lead to better future results. By experimenting with many different paths, and figuring out which of them are good or bad, it eventually begins to build a network of strong and weak paths. In doing so, it helps to suggest to the agent “stronger” decisions / choices in navigating a maze, or solving a complicated problem. An algorithm is simply a set of steps to follow, generally intended to maximize or guarantee successful execution of a process. With our machine learning agent, we give it a couple of steps to repeat, generating an iteratively improving navigator. This can be extrapolated to lots of different situations and more complicated problems. The same process of iterative learning is then applied and reinforced in order to generate an optimal pathing.

**How did you implement deep Q-learning using neural networks for this game?**

Deep q-learning is a form of reinforcement learning. In the python code, we make use of a neural network, (the model). This model is then used to strengthen the chances of future rewards / success for the agent, by recursively learning from its experience, and balancing levels of exploration and exploitation to achieve a strong neural pathway.

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. <https://www.turing.com/kb/how-are-neural-networks-used-in-deep-q-learning>