* Analyze the differences between human and machine approaches to solving problems.
* Describe the steps a human being would take to solve this maze.

The steps that a human being would take to solve this maze would be the following:

1. Visualize – in this step, a human being would most likely conduct what I call a sort of a visual assessment where they would first look at the maze, identify the pathways that are available, then analyze by mentally mapping out what potential routes could lead to the treasure and which ones don’t.
2. Trial and error – in this step, a human being most likely recalls their attempts when using different paths, so they learn from past mistakes, what works and what doesn’t. This helps them avoid obstacles and dead ends in future attempts.
3. Heuristic Navigation – in this step, human beings use their intuition and reasoning in order to prioritize the pathways that seem more direct and take less time and effort.
4. Pattern recognition – in this step, human beings recognized all the familiar patterns from previous attempts (experience) and apply this information and experience to all of their future attempts.
5. Goal-oriented decision making – in this step, human beings adapt by re-evaluating the path needed to take to get to the treasure based on all of the new information and experience that they have gained from previous attempts and acquired knowledge.

* Describe the steps your intelligent agent is taking to solve this pathfinding problem.

The steps that my intelligent agent is taking to solve this pathfinding problem are the following:

1. Exploration and Experience Replay – in this step the intelligent agent randomly explores different paths and then stores these experiences in memory.
2. Learning through rewards (Reinforcement learning) – in this step the intelligent agent assigns both rewards and penalties based on the actions that it has taken.
3. Neural Network Predictions – in this step the intelligent agent uses the Deep Q-Network (DQN) in order to predict the best possible action and this is based on its learned experience throughout their exploration (game playing).
4. Iterative thinking – in this step the intelligent agent is trained over multiple epochs which improves its decision-making over time.
5. Optimal Policy Formation – in this step the intelligent agent’s final trained model develops an optimal strategy that will help it reach its goal – efficiently reaching the treasure.

* What are similarities and differences between these two approaches?

A table with text on it

AI-generated content may be incorrect.

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

The difference between exploitation and exploration is that First of all the intelligent agent balances these two strategies – exploitation and exploration, the difference between them is that exploitation uses what it has learned in order to choose what it deems to be the best-known path to its goal, in this case reaching the treasure. While exploration tries new and unexplored paths (hence the name – it “explores” more) in order to learn more about its environment. I believe that a high exploration rate is useful during the early stages of training in order to discover optimal paths. However, more exploitation is preferred during the later stages of training in order to help with efficient decision making.

As for finding the ideal proportion (balance) for this pathfinding problem I would suggest the following after running four experimental runs for this project:

* Higher exploration when having a low batch size and more epochs.
* Even though this run is slower and takes its time, in the end, it finds better long-term solutions.
* Higher exploitation when having a high batch size and less epochs.
* This is much faster training, but it also risks missing optimal paths.

In the end, the model I chose (epochs=18, batch\_size=64) achieved a 100%-win rate in 95 epochs, which I believe shows a well-balanced mix of exploitation and exploration.

* How can reinforcement learning help to determine the path to the goal (the treasure) by the agent (the pirate)?

Reinforcement learning can help to determine the path to the goal (the treasure) by the agent (the pirate) by:

1. Trial and Error Learning – here the intelligent agent (pirate) tries out different paths and remembers what the penalties are for not reaching the treasure or making wrong moves (decisions) and what the rewards are for reaching the treasure or making correct moves (decisions).
2. Discounted Rewards (GAMMA) – here the intelligent agent (pirate)’s decisions are influenced by the future rewards that lie ahead. This is what allows the intelligent agent (pirate) to plan ahead rather than just making immediate decisions for the sake of it.
3. Neural Network Predictions – here the DQN predicts what the best action to take is from any given position.
4. Optimal Path Formation – it is here that over time, Q-learning refines the intelligent agent’s choices in order to create the most efficient route.

This shows that by balancing both exploitation and exploration, reinforcement learning can successfully teach the pirate (intelligent agent) to reach the treasure more efficiently.

* Evaluate the use of algorithms to solve complex problems.
* How did you implement deep Q-learning using neural networks for this game?

I implemented deep Q-learning using neural networks for this game by using:

1. Experience Replay – this is where past actions are stored, allowing the intelligent agent (pirate) to learn from its previous mistakes.
2. Neural Network Updates – this refers to the model.fit() function in the code, which trains the neural network to improve its decision making.
3. Q-Value Optimization – this refers to the model.evaluate() function in the code, which ensures that the model is indeed updating correctly.
4. Exploitation and Exploration Balance – by taking the liberty to make four experimental runs with different values, I was able to see hoe the model efficiently learned over time.

A table with numbers and time

AI-generated content may be incorrect.

The Key Takeaways from my Experimental Runs:

1. Higher Batch Size leads to faster training
   * + My best result came from having a batch\_size=64, which significantly reduced the training time.
2. Lower Epochs can still be effective
   * + My best result came from having Epochs=18, which in my opinion provided the best trade-off between fast training and optimal learning.
3. Fine-Tuning hyperparameters matters
   * + Finding the right batch size and epochs required multiple runs (trials).
     + The final model effectively balances speed and accuracy.

In conclusion, through Deep Q-learning, I successfully trained an intelligent agent (pirate) to find the treasure in the shortest amount of time. I was able to do this by adjusting the hyperparameters such as batch\_size and Epochs. This allowed me to optimize the model for both speed and learning efficiency. In the end, my final model achieved a 100%-win rate in just 6 minutes and 22 seconds. This, as you can see from the chart above, is a significant improvement from the original 20-minute runtime (the one obtained during the project one milestone) or the third run with a longer runtime of 24 minutes and 30 seconds. This in my opinion reinforces the power of deep reinforcement learning in solving complex pathfinding problems. It also both shows and highlights how AI can efficiently learn from experience in order to optimize its decision-making.

Images of my notes while performing the experimental runs:

A close-up of a notebook

AI-generated content may be incorrect.

References

Sutton, R., Barto, A. (2014, 2015). Reinforcement Learning: An Introduction. MIT Press. <https://web.stanford.edu/class/psych209/Readings/SuttonBartoIPRLBook2ndEd.pdf>

Mnih, V., Kavukcuoglu, K. (2015, February 25). Human-level control through deep reinforcement learning. Nature, 518, 529-533. <https://www.nature.com/articles/nature14236>

Silver,D., Schrittwieser, J., et al. (2017, October 19). Mastering the game of Go without human knowledge. Nature, 550, 354-359. <https://www.nature.com/articles/nature24270>

OpenAI. (2018). Welcome to Spinning Up in Deep RL! OpenAI Spinning Up. <https://spinningup.openai.com/en/latest/>