**Solving a Pathfinding Problem with Deep Q-Learning: Design Defense**

Connie M. Knupp

Department of Computer Science, Southern New Hampshire University

CS370: Current / Emerging Trends in CS

Dr. Ashima Yadav

October 19, 2025

Key elements of many video games are non-player characters with specific roles to play or tasks to complete.  To create a challenging and engaging treasure hunt game, developers have designed an intelligent agent, a pirate, to serve as the player's adversary. The pirate will compete directly against the human player to be the first to find the treasure. A deep Q learning algorithm was developed to train the pirate agent to solve this pathfinding problem. This competition provides a great opportunity to examine the many differences between human learning and machine learning; each stemming from fundamentally different types of intelligence.

 Human intellect is based on past experiences and driven by creativity and abstract thinking. Humans “make connections based on our overall understanding and experience, rather than just a specific context.” (Ricotta, 2024). If a human were solving the pathfinding problem, he/she would analyze the entire picture and plan a path that intuitively avoids obstacles. Humans would see the border of the maze and grasp the concept that the player must stay within the boundaries, even if the instruction is not explicitly stated. If a human pathfinder took a path that ended in a dead end, they would backtrack to another open path, often looking ahead several turns to see which path was the best option. Humans learn quickly, often from just one experience, and apply the knowledge to future scenarios. The explanation for this is that “humans excel at learning from experience, social interactions, and observation. Our ability to adapt, generalize across contexts, and apply abstract thinking allows us to learn from relatively small amounts of data.” (Zhang, n.d.). The methods used by an intelligent agent to solve the same pathfinding problem are vastly different due to the nature of machine intelligence.

The intelligent agent created by the development team, the pirate, learns to solve the pathfinding problems in an entirely different manner. First, an intelligent agent requires training, which can be accomplished in many ways. Common methods used in machine learning are supervised learning, unsupervised learning, and reinforcement learning. (Ricotta, 2024). The team selected deep Q-learning, a reinforcement learning method, to train the intelligent agent for this pathfinding problem. In deep Q-learning, “the agent learns a policy by updating a neural network (the “online network”) to predict the expected future rewards (Q-values) for actions in a given state.” (*What are Target,* 2025). According to specifications, the agent is given the ability to move right, left, up, or down and is rewarded with a point for reaching the treasure. The agent has a small fraction of a point deducted for moving to discourage wandering and a larger fraction of a point deducted for moving outside of the maze or to an occupied square. (Southern New Hampshire University, n.d.). In the developed algorithm, the agent takes an action, choosing to exploit the environment using the developed policy 90% of the time, and explores 10% of the time, then records the state, action, and rewards before choosing another action. After a specified number of actions are recorded, the policy is updated so that the agent improves its decision-making policy over time, maximizing rewards, until the agent has mastered the pathfinding problem. The repetitions needed to train the intelligent agent are just one of the factors that distinguish machine learning from human learning.

An intelligent agent may require more repetitions to solve pathfinding problems than a human would, but the approaches are not entirely different. The approach used has both similarities and differences to how humans approach the same problem. A similarity between the two approaches is initial exploration; if the path is unknown, both a human and an intelligent agent may choose to randomly strike out in a direction and explore the branches of the maze. Another similarity is obstacle avoidance or adaptability; after observing the negative reward for moving to an occupied square, the intelligent agent avoids the obstacle just as a human would. Observation of the environment is also used by both intelligent agents and humans, although the modes of observation are different; for example, humans use their senses to observe the environment, while the intelligent agent uses numeric representations of the current state.

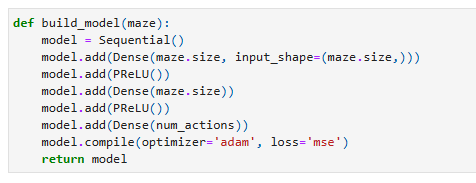
Additional differences include planning; humans can look ahead to plan the best path, often using intuitive guesses, like exploring the area of the maze that seems more open first. In contrast, intelligent agents initially explore randomly, considering each action on its own merit until more about the environment is learned. The intelligent agent makes analytical decisions based on data, while the human uses creativity and intuition to interpret the environment. (Zhang, 2024). It is the purpose of an intelligent agent to navigate an environment autonomously, finding the most efficient route to a goal. To do that effectively in a pathfinding problem, one must manage exploitation and exploration.

GeeksforGeeks explained the two strategies with the following statement, “One of the most critical aspects of machine learning that people must keep in mind is the proper balance for exploitation and exploration.” (*Exploitation and Exploration*, 2025). Exploitation is the use of compiled knowledge to guide decisions to maximum expected reward. Exploration is choosing actions with uncertain outcomes to gather information about an environment or model. (*Exploitation and Exploration*, 2025). During exploration, the intelligent agent would choose to move in a direction where the state and reward are unknown. When exploiting, the intelligent agent would choose actions that it knows lead to the highest rewards using the information obtained from prior experience. The ideal proportion of exploitation and exploration for a pathfinding problem is an epsilon-greedy algorithm.

“Epsilon-greedy algorithms manage to unify those two characteristics (exploitation and exploration) by sometimes choosing completely random actions with probability epsilon while continuing to use the current best-known action with probability (1 - epsilon).” (*Exploitation and Exploration*, 2025). Ideally, the intelligent agent would start with a high epsilon, exploring a majority of the time, while little is known about the environment. The epsilon would gradually decay, leading to more exploitation as the environment is learned and the agent has a good base of knowledge to leverage. Maintaining a minimum epsilon ensures the agent will continue to explore randomly, which helps the agent find the best route possible. While random exploration allows the intelligent agent to explore the maze, the states, actions, and experiences are recorded. In deep Q-learning, the stored information is then used to periodically retrain the neural network, with experience replay.

Experience replay is a technique in reinforcement learning that stores past experiences in a replay buffer and then uses a random sampling of these experiences to update the policy. (Beysolow, 2019). The policy is the guide the intelligent agent is using to decide on which action to take, and in deep Q-learning, the policy is learned indirectly. It is indirectly improved as the neural networks update estimated rewards based on current states and actions, which are known as Q-values. Managing an appropriate balance between exploitation and exploration allows the agent to learn how to maximize the reward. Combining this balance with the use of experience replay to train neural networks is how reinforcement learning is used to teach the pirate to find the optimal path to the goal.

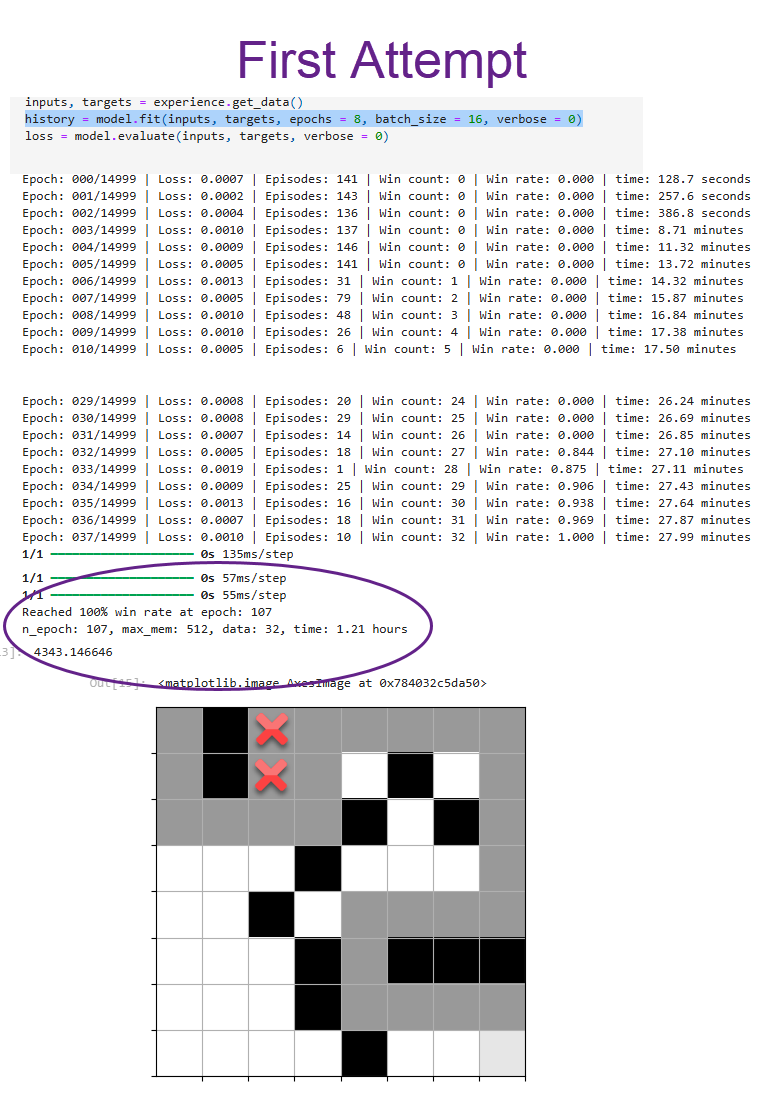
To implement deep Q-learning for this treasure hunt game, the development team created a neural network with one input layer, one hidden layer, and an output layer. Both the input layer and the hidden layer have as many neurons as the maze size; the number of neurons in the output layer is equal to the number of available actions. This allows the neural network to accept the current state of the maze as input, perform complex operations using the Bellman equation, and link the probability of future rewards to each action as the output. (Beysolow, 2019). The layers of the neural network are followed by the activation of a Parametric Rectified Linear Unit (PreLU) function, which introduces non-linearity, allowing the network to learn complex relationships. “PReLU improves upon Leaky ReLU by making the slope a learnable parameter, enhancing model accuracy and convergence.” (Olamendy, 2023). As you can see in the code block below, the model is compiled using “adam” as the optimizer and the mean squared error (mse) as the loss function. This setup built a neural network capable of training the pirate to traverse the maze.

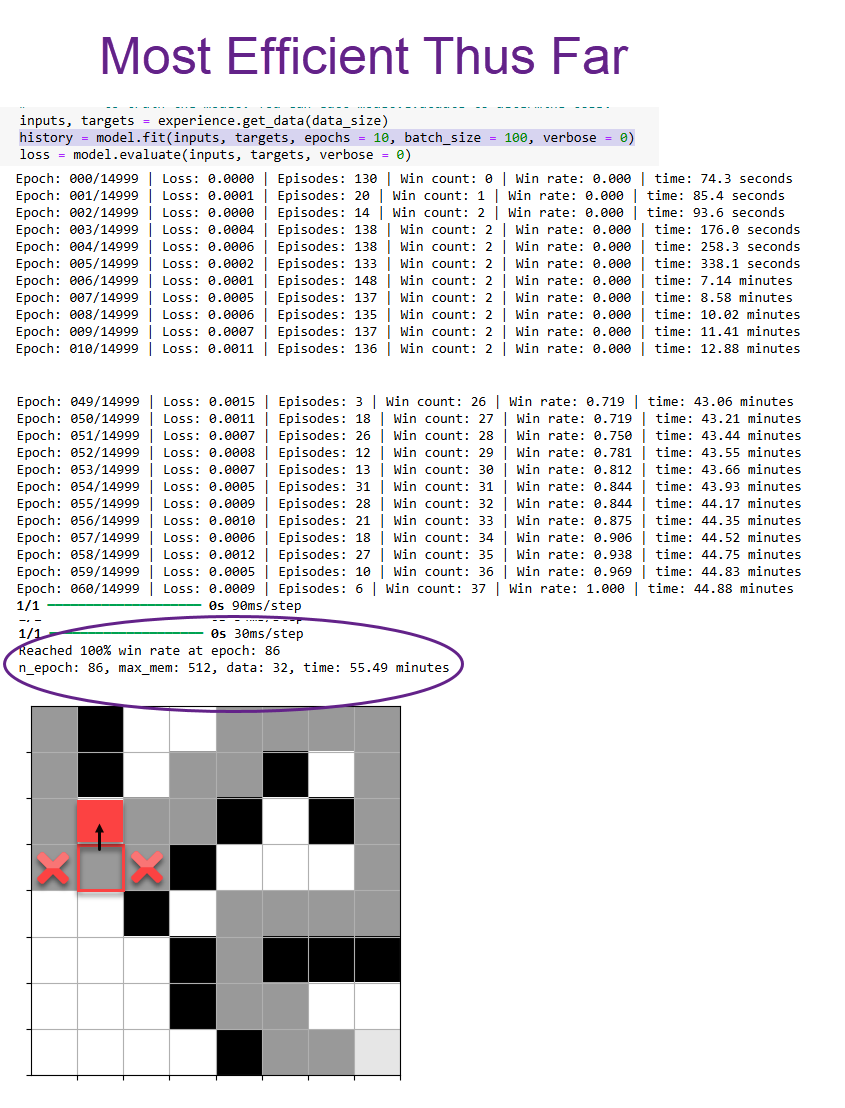


The online network, created by the code above, was cloned to create a target network. The target network is a slower copy of the online network created for stability and to reduce volatility that can occur from frequently updating Q-value targets. (*What are Target,* 2025). The epsilon was set at 0.1 and reduced to 0.05 when the win-rate was above 90%, meaning the agent explored 10% of the time and exploited the environment 90% of the time in the beginning. Once the agent knew enough about the environment, the exploration decreased to 5% of the time; maintaining some exploration helps to ensure the agent does not get stuck in exploitation mode before the optimal route is found.

The first attempt at training the intelligent pirate set the number of passes through the data (epochs) at 8 and the batch size to 16, meaning it updated the neural network after every 16 samples. During this training, the win rate reached 100% at epoch 37 and passed the completion check at epoch 107, which took 1.21 hours. Several iterations of training were completed with varying training parameters, some of which exceeded resources and stopped execution, some leading to less effective training. The most effective training that has been found to date had epochs set at 10 and the batch size set at 100. During this training, the win rate reached 100% later, at epoch 60, but passed the completion check earlier at epoch 86, taking only 55.49 minutes. Meaning slower updates to the neural network with more passes through the complete data set lead to an overall improvement in the efficiency of the training. In both instances, in the post-training testing, the agent reached the treasure with only 2 extra steps.

Key screenshots from these two training and testing experiments are displayed on subsequent pages. Additional experiments with training variables may be able to increase training efficiency further. Adjusting the epsilon or expanding parameters in the completion check may lead to the agent finding a better path and removing the 2 extra steps that were identified in both cases. However, even without those optimizations, the neural network created for deep Q-learning used by the intelligent pirate successfully met specifications and solved the pathfinding problem.





References

Beysolow, I. T. (2019). *Applied reinforcement learning with python: With openai gym, tensorflow, and keras.* Apress L. P..

GeeksforGeeks. (2025, July 23). *Exploitation and exploration in machine learning*. GeeksforGeeks. <https://www.geeksforgeeks.org/machine-learning/exploitation-and-exploration-in-machine-learning/>

Lake, B. M., Ullman, T. D., Tenenbaum, J. B., & Gershman, S. J. (2017). Building machines that learn and think like people. *The Behavioral and brain sciences*, *40*, e253. <https://doi.org/10.1017/S0140525X16001837>

Milvus. (2025). *What are target networks in DQN?*. Milvus. <https://milvus.io/ai-quick-reference/what-are-target-networks-in-dqn>

Olamendy, J. (2023, December 4). *Understanding ReLU, LeakyReLU, and PReLU: A Comprehensive Guide*. Medium. <https://medium.com/@juanc.olamendy/understanding-relu-leakyrelu-and-prelu-a-comprehensive-guide-20f2775d3d64>+

Ricotta, F. (2024, February 4). *How humans learn, how machines learn, and how we learn together*. BurstIQ. https://burstiq.com/blog-how-humans-and-machines-learn-together/

Southern New Hampshire University. (n.d.). *CS 370: Pirate Intelligent Agent Specifications.* [PDF Course Handout]. LearnSNHU. <https://learn.snhu.edu/content/enforced/2019739-CS-370-10416.202581-1/course_documents/CS%20370%20Pirate%20Intelligent%20Agent%20Specifications.pdf?ou=2019739>

Zhang, J. (n.d.). *Artificial Intelligence Versus Human Intelligence: Which excels where and what will never be matched*. UTHealth Houston. https://sbmi.uth.edu/blog/2024/artificial-intelligence-versus-human-intelligence.htm

Zhao, Min. *Understanding humans’ strategies in maze solving from eye-hand coordination***.**Retrieved from <https://doi.org/doi:10.7282/T38W3FX5>