Thomas Cheek

Prof. Kraya

CS 370

2/20/2025

Design Defense: Learning Deep Q-Learning for Pathfinding

How a Human Solves a Maze:

When I think about how a human would solve a maze, the process seems straightforward. First, we look at the maze, spotting any walls and possible routes. Then, we try different paths, remembering where we got stuck, and gradually figure out the fastest way to reach the goal. We can use logic and intuition to speed things up.

How My AI Agent Solves the Maze:

My AI agent uses Deep Q-Learning to navigate the maze. It breaks the maze into a grid where each square is either open, blocked, or the target. The agent can move left, right, up, or down. It learns by receiving rewards or penalties for its moves. Over time, it stores this information in its memory and uses a neural network to predict the best moves to maximize rewards.

Similarities and Differences:

| * Similarities | * Differences |
| --- | --- |
| * Both explore different paths to find a solution. | * Humans rely on logic and intuition, while the AI uses numbers. |
| * Both learn from mistakes to avoid dead ends. | * The AI needs thousands of attempts to learn effectively. |
| * Both aim to reach the goal as quickly as possible. | * Humans can adapt faster, but AI becomes more consistent. |

The main goal of my AI agent is to find the fastest and most efficient route to reach the treasure while avoiding obstacles. Using reinforcement learning, the agent improves its performance by learning from both rewards and penalties.

* Exploration vs. Exploitation:
  + Exploration: The agent tests different paths, even if they don’t seem optimal, to discover new possibilities.
  + Exploitation: Once the agent knows the best path, it uses that knowledge to maximize rewards.
* Balancing Exploration and Exploitation: I found that starting with 90% exploration and 10% exploitation helped the agent explore thoroughly. As it learned, switching to 90% exploitation and 10% exploration allowed it to apply what it had learned and navigate more efficiently.

How does Reinforcement Learning help the agent find the treasure?

Reinforcement learning teaches the agent by rewarding good actions and penalizing bad ones. Each move results in a specific score:

* +1.0 for reaching the treasure.
* -0.75 for hitting a wall.
* -0.8 for trying to move outside the maze.
* -0.04 for each step to discourage wandering.

By repeating this process, the agent figures out which sequences of moves lead to the highest rewards.

Neural Network Structure:

* Input Layer: Represents the current state of the maze.
* Hidden Layers: Two layers with ReLU activation functions to learn complex patterns.
* Output Layer: Predicts the Q-values for each possible move.

Training Process:

* Experience Replay: The agent saves its experiences and uses random samples to train, which helps break the correlation between consecutive actions.
* Bellman Equation: The neural network is updated using this equation: Q(s,a)=Q(s,a)+α[R+γmaxQ(s′,a′)−Q(s,a)] where α is the learning rate and γ is the discount factor.

Key Settings:

* Learning Rate (α): 0.001
* Discount Factor (γ): 0.95
* Exploration Rate (epsilon): Starts at 1.0 and decreases to 0.01

Results:

By adjusting the number of training cycles and batch sizes, my AI achieved 95.4% accuracy in just 15 minutes, proving that deep Q-learning is a fast and effective way to solve pathfinding problems.

Reflection:

This project helped me see how reinforcement learning allows machines to improve through trial and error. By adjusting hyperparameters like the learning rate and exploration rate, I was able to reduce training time while still achieving high accuracy. Seeing my AI agent successfully navigate the maze felt like proof that neural networks can tackle complex problems just like humans only with a lot more patience!

References

* Sutton, R. S., & Barto, A. G. (2018). *Reinforcement Learning: An Introduction*. MIT Press.
* Mnih, V., et al. (2015). *Human-level control through deep reinforcement learning*. Nature, 518, 529–533.
* TensorFlow Documentation. (n.d.). *Building Neural Networks with Keras*. Retrieved from [tensorflow.org](https://www.tensorflow.org/)