Design Defense

Human vs. Machine Problem-Solving

Humans and machines approach problem-solving differently. A human solving the maze would most likely analyze it visually, identifying potential paths and obstacles. They might plan a route based on intuition and experience. They can adjust their strategy dynamically as they encounter dead ends or new opportunities. In contrast, the intelligent agent employs a systematic approach using deep Q-learning, which relies on reinforcement learning and neural networks to evaluate actions. The agent uses trial-and-error to explore the maze, gathering information and improving its decision-making over time.

Steps a Human Would Take:

1. Visually inspect the maze to locate the starting position and the treasure.
2. Identify possible paths and obstacles.
3. Plan an initial route to the treasure.
4. Adjust the route dynamically based on the outcome of each movement.
5. Repeat until the treasure is reached.

Steps the Intelligent Agent Takes:

1. Observe the current state (position in the maze).
2. Choose an action (move left, right, up, or down) based on exploration (random moves) or exploitation (using learned data).
3. Execute the action and observe the result (new state, reward, and game status).
4. Store the episode in memory for future learning.
5. Train the neural network using past experiences to improve decision-making.

Similarities and Differences:

Both methods involve trial-and-error and learning from previous attempts. The human approach is guided by intuition and prior experience, while the intelligent agent relies on data and algorithms (DeepMind 2024). The agent’s ability to systematically explore all options allows it to find optimal solutions over time, unlike humans, who may stop after finding a working path. This can lead humans to settle for suboptimal paths.

From a paper published by Google’s DeepMind (2024**)**, we see parallels in how both systems benefit from combining "external search" (systematic exploration of options) and "internal reasoning" (policy refinement). For instance, the TreasureMaze intelligent agent balances exploration and exploitation in ways like Monte Carlo Tree Search (MCTS) strategies used in chess and other games.

The intelligent agent’s purpose in pathfinding is to efficiently navigate the maze and reach the treasure. It must balance exploitation (choosing the best-known action based on prior knowledge) and exploration (testing new actions to discover better paths). For this problem, an ideal proportion might involve a higher exploration rate during initial epochs to gather data and gradually shifting towards exploitation as the agent learns the maze (Amine, 2020). This ensures the agent can discover optimal strategies while avoiding over-reliance on incomplete knowledge. Reinforcement learning enables the agent to assign values to actions based on rewards. Positive rewards guide the agent toward desirable outcomes, while penalties discourage poor decisions. Over time, the agent learns the most efficient path to the goal by maximizing cumulative rewards (DeepMind 2024).

Evaluating the Algorithm

Deep Q-learning uses neural networks to approximate the Q-value function, which predicts the expected cumulative reward for each possible action in each state. This game implements deep Q-learning through the following steps:

* Input State Representation: The maze’s state is encoded as a feature vector.
* Action Selection: The agent selects actions based on the epsilon-greedy policy, balancing exploration and exploitation.
* Experience Replay: Episodes (state, action, reward, next state) are stored in memory and sampled for training.
* Neural Network Training: The model minimizes the mean squared error between predicted and target Q-values, enabling it to improve its predictions.
* Policy Update: The agent’s policy is updated after each training step to reflect learned knowledge.

Deep Q-learning is well-suited for this problem because it allows the agent to learn complex strategies in environments with large state spaces. The use of neural networks enables the agent to generalize and adapt to different situations, ensuring efficient pathfinding.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generatedModel never reached 100%. Plateaued around 0.98

References:

Amine, Amrani. (Dec 2020). *Q-Learning Algorithm: From Explanation to Implementation.* Towards Data Science. https://towardsdatascience.com/q-learning-algorithm-from- explanation-to-implementation-cdbeda2ea187

Geeks for Geeks. (Aug 2024). *Q-Learning.* Geeks for Geeks. https://www.geeksforgeeks.org/q- learning-in-python/

DeepMind. Multiple Authors. (Dec 2024). *Mastering Board Games by External and Internal Planning with Language Models.* Google DeepMind. https://deepmind.google/research/publications/139455/