Kalee Li

Professor: Timothy Alexander

CS 370: Current and Emerging Trends in Computer Science

**Project Two – Design Defense**

# **Analyze Human vs. Machine Intelligence.**

## The steps a human being would take to solve this maze.

Humans learn from intuitive understanding, experience, sensory input, and perception to activate our cognitive process. The maze in this project is not complicated; therefore, humans can take a quick look and find the route almost instantly. However, for humans to take steps in a more complex maze, the steps as follow:

1. Observe the environment – to understand the maze, look at the starting point, ending point, and possible routes.
2. Mental plan or written plan – create a strategy to navigate the maze. (If the maze were to be more complicated, a map may be created)
3. Trial and error –attempt to traverse or backtrack the maze.
4. Memory and patterns – use memory of previous positive or negative attempts to avoid incorrect paths.

## The steps of an intelligent agent took to solve this maze.

The intelligent agent learns to find the path using a neural network training DQL algorithm, which comprises iterative exploration, exploitation, and storing experience.

Training loops start

1. For each epoch, the agent randomly selects a free cell in the maze as the starting point, and then the agent observes.
2. The agent interacts with the environments until the game is over
3. The agent selects and executes an action using an exploration and exploitation strategy.
4. Each win or loss gets recorded in the win history list
5. Each episode, such as previous environment state, action, reward, current environment state, and game status, gets added to the episode list and experience replay memory, experience.remember().
6. Retrieve training data, train the model, and evaluate to determine loss.
7. If the win rate is above the threshold and passes the completion check, the model has achieved satisfactory performance, and the training can be terminated early.

## The similarities and differences between human and AI approach

The human and AI approaches are similar in that we both use a combination of exploration and exploitation, meaning trying new paths and using previous experience to determine a good path. As mentioned, the agent randomly selects a free cell (exploring), each experience appended to the episode list, and records game status, replay experience, and replay memory (experience) to choose an action. Equivalently, humans use trial and error, memory, and pattern recognition to make attempts, backtrack, and recall previous feedback to make decisions.

The main difference between the human and AI approaches is that the human approach is less structured because we heavily rely on intuition pattern recognition and memory. However, it is more adaptive than AI because the AI algorithmic approach often has very well-predefined rules, and parameters (rules and boundaries) are set rigidly. Therefore, the human problem-solving skill is more flexible and adaptive, whereas machine approaches are more systematic.

# **The purpose of the intelligent agent in pathfinding.**

## Exploitation and exploration and the ideal proportion of exploitation and exploration.

This pathfinding problem is implemented using a Deep Q-network (DQN) in Reinforcement Learning (RL). The agent learns to navigate the environment through exploration and exploitation without human input or labeled data. **Exploration** is when the agent randomly tries out new action with different approaches to see what works the best; in other words, it is used to experiment with new action to see which action has a greater Q value. **Exploitation** is when the agent selects actions based on experience to maximize immediate reward, meaning it uses existing data, remember(episode), to select the highest estimated value action to replay. This project's exploration factor (epsilon) is set to 0.1, meaning the agent will select an action by experience 90% of the time and randomly explore a new path 10%. The ideal proportion is gradually decreasing the exploration factor rate as the agent learns from each epoch because the agent has accumulated sufficient experience for faster convergence.

## How does RL help to determine the path to the goal (the treasure) by the agent (the pirate)?

The reinforcement learning provided an iterative learning framework for the agent to determine the path to the goal in a maze. This approach allowed the agent to learn from interacting with the environment, balancing the exploration and exploitation factors and choosing the best action.

1. Environment
   1. State - The maze object contained an 8x8 matrix and visual representation.
   2. Action- The pirate agent can select to move in four directions: left, right, up, and down
2. Deep Q Network training
   1. Using a neural network to estimate Q-value
3. Exploration and Exploitation
   1. Balancing the exploration and exploitation to maximize Q-value
4. Iteration
   1. Environment
   2. DQN training
   3. Exploration and Exploitation

# **Evaluate the use of algorithms.**

## The implementation of deep Q-learning using neural networks for this game.

The breakdown of how I implemented deep Q-learning using neural networks for this game is below:

1. Initialize the necessary value to a variable.

A screenshot of a computer program

Description automatically generated

1. Training loop

A white background with black and purple text

Description automatically generated

1. For each epoch, the agent starts randomly selecting a free cell in the maze random.choice(qmaze.free\_cells), then reset the agent cell to the current position reset(agent\_cell) call observe() in the TreasureMaze.py class, observe() returns the current state of the environment.

A screenshot of a computer code

Description automatically generated

1. Agent interacts with the environment until the game is over. Initialize the previous environment state to the current environment state.

A close up of text

Description automatically generated

1. Action: The agent will choose action either by exploration or by exploitation. Epsilon is the exploration factor, and it was set to .1, then np.random.rand() generates a random number. If this number is less than epsilon, then the agent will explore, choosing a random action. Otherwise, the agent will exploit, choosing the action based on the Q-value. qmaze.act(action) – action execution

A white background with black text

Description automatically generated

1. Adding win or lose to the list

A computer screen shot of a computer code

Description automatically generated

1. Initializing the episode to a list. The elements are the single experience the agent encountered for each interaction with the environment. Then, call remember() from the GameExperience.py to store the episodes in the memory list.

A screenshot of a computer screen

Description automatically generated

1. The build\_model code was included in the starter code. The build model comprises an input layer, two hidden dense layers, two activation layers, and an output layer.

A computer code with black text

Description automatically generated

Call get\_data from GameExperience.py, which returns input and target data from memory for training the model, data\_size: The number of episodes to sample from memory.

A close-up of a text

Description automatically generated

1. If the length of win\_history list is greater than history window size, then the win rate is the sum of the last hsize elements from the win-history list then divided by the history window size. In other words, it calculates the average win rate based on the last hsize games in win\_history

If win\_rate greater than 0.9, then epsilon is 0.05. If the sum of the last hsize elements from the win-history list and completion check is true, then the agent reached the treasure.

A screenshot of a computer code

Description automatically generated

# **Reference**

TreasureEnvironment.py & GameExperience.py. (n.d.). CS 370 Pirate Intelligent Agent Specifications. In *TreasureHuntGame Jupyter notebook*. <https://learn.snhu.edu/content/enforced/1610115-CS-370-11333.202451-1/course_documents/CS%20370%20Pirate%20Intelligent%20Agent%20Specifications.pdf?ou=1610115>

SNHU Media. (2020, April 30). *CS 370: Project Two Walkthrough* [Video]. YouTube. <https://www.youtube.com/watch?v=snRe-zLzEy4>

Yoon, C. (2021a, December 7). Deriving Policy Gradients and Implementing REINFORCE. *Medium*. <https://medium.com/@thechrisyoon/deriving-policy-gradients-and-implementing-reinforce-f887949bd63>

JanardhananPS. (2020, April 2). *Human Learning and Machine Learning – How they differ ?* Data Science Central. <https://www.datasciencecentral.com/human-learning-and-machine-learning-how-they-differ/>

Fernandez, E. (2023, October 5). AI Is Not Similar To Human Intelligence. Thinking So Could Be Dangerous. *Forbes*. <https://www.forbes.com/sites/fernandezelizabeth/2019/11/30/ai-is-not-similar-to-human-intelligence-thinking-so-could-be-dangerous/?sh=bb80bc16c22f>