Ihab Elrayah
CS370

7-3 Project Two

Analysis of Human and Machine Approaches to Solving Problems, Humans solve problems using intuition, experience, and adaptive thinking, often throough trial and error. We often rely on visual assessment, memory, and strategic planning. “*Machines, on the other hand, utilize algorithms and structured learning processes*”(Human and Machine Problem Solving). They process vast amounts of data, perform complex calculations, and systematically update their strategies based on feedback, achieving efficiency and consistency in problem-solving.

Human Approach, Humans approach problem-solving by leveraging perception, planning, memory, and adaptive thinking. For instance, when solving a maze, a human would examine the layout visually, identifying the starting point (pirate) and the endpoint (treasure). They would plan a route by considering various paths, using trial and error to find the shortest. As they navigate the maze, they remember previously tried paths to avoid redundancy and adapt their strategy based on new information, such as encountering dead ends. For example, a human might start at the initial position, evaluate the possible moves, and choose one based on visual inspection and intuition. If a dead end is encountered, the human would backtrack and try a different route, learning from the failed attempt.

Machine Approach, In contrast, machines use algorithms and structured learning processes to solve problems. In the given code, the pirate agent uses deep Q-learning to find the optimal path to the treasure. The maze is represented as a grid, and the agent learns by performing actions (left, right, up, down) and receiving rewards. For instance, the agent uses an exploration factor (epsilon) to balance exploring new paths and exploiting known information. The Q-values, which estimate the expected rewards for actions, are updated based on the rewards received and the maximum Q-value of the next state.

```
# Initialize the maze and agent state
qmaze = TreasureMaze(maze)
qmaze.reset((0, 0))  # Start at the initial position

# Perform an action (e.g., move DOWN) and get the resulting state and reward
canvas, reward, game_over = qmaze.act('DOWN')
print("reward=", reward)
show(qmaze)  # Visualize the updated environment

# Example of exploration vs exploitation decision-making
if np.random.rand() < epsilon:
    action = random.choice(valid_actions)  # Exploration
else:
    action = np.argmax(model.predict(state))  # Exploitation
```

Steps a Human Would Take:

- Examine the maze layout, identify the start (pirate) and end points (treasure).
- Visualize possible paths, considering the shortest or easiest routes.
- Move step-by-step, re-evaluating the position and path based on information.
- Remember to try paths and adjust the strategy based on success or failure.

Steps the Intelligent Agent Takes:

- Represent the maze as a grid with states and actions.
- Use deep Q-learning to estimate Q-values for actions.
- Balance exploration of new paths actions and exploitation of known information.
- Perform actions, receive rewards, and update Q-values

Similarities and Differences, Both humans and machines aim to find the optimal path through exploration and learning from experience. Humans use intuition, memory, and adaptive thinking, relying on visual assessment and strategic planning. On the other hand, machines rely on structured algorithms and mathematical models, such as deep Q-learning, to estimate the best actions based on rewards and state transitions. While humans adapt in real-time and remember past attempts, machines use a systematic approach to update their policies based on cumulative rewards, balancing exploration and exploitation with factors like epsilon.

For example, while a human would visually inspect and backtrack based on memory, the machine uses code below to decide whether to explore new paths or exploit known information, updating its Q-values and policy accordingly.

```
if np.random.rand() < epsilon:
    action = random.choice(valid_actions)  # Exploration
else:
    action = np.argmax(model.predict(previous_envstate))  # Exploitation
```

Purpose of the Intelligent Agent in Pathfinding, The purpose of the intelligent agent in pathfinding is to autonomously navigate through a maze to locate a specified goal, such as the treasure, by determining the most efficient route. Using techniques like deep Q-learning, the agent learns from its interactions with the environment to optimize its movements, improving its performance overtime without human intervention.

Exploitation vs. Exploration, Exploitation refers to the agent using existing knowledge to choose the best known action to maximize rewards. Exploration, on the other hand, involves trying new actions to discover potentially better paths. The ideal proportion of exploitation and exploration balances learning new information and utilizing current knowledge effectively.

Ideal Proportion of Exploitation and Exploration, The ideal proportion of exploitation and exploration in this pathfinding problem should ensure that the agent can efficiently learn the maze while adapting to new possibilities. "*A 90% exploitation and 10% exploration ratio is reasonable because it allows the agent to primarily rely on its learned experience, ensuring quicker convergence to optimal paths*"(/towardsdatascience.com), while still maintaining some level of exploration to discover new paths or adapt to changes in the environment.

Role of Reinforcement Learning, Reinforcement learning helps the agent determine the path to the goal by "*continuously updating its knowledge based on the rewards received from different actions."*(aws.amazon.com) The agent learns to associate actions with positive or negative outcomes through trial and error. In this problem, the pirate (agent) receives rewards or penalties for each move, gradually refining its policy to maximize cumulative rewards. This learning process enables the agent to identify and follow the optimal path to the treasure over time.

Evaluating the Use of Algorithms to Solve Complex Problems, Algorithms efficiently solve complex problems by providing structured methods to process large amounts of data and perform intricate calculations. They are crucial in areas like pathfinding and optimization, enabling solutions that are often beyond human capabilities.

Implementing Deep Q-Learning Using Neural Networks for This Game, Deep Q-learning was implemented by approximating Q-values with a neural network, which guides the pirate agent to the treasure. The agent learns by performing actions, receiving rewards, and updating its Q-values. The training loop involved randomly selecting start positions, choosing actions based on a balance of exploration and exploitation, and updating the neural network with experiences stored in memory. This process allows the agent to refine its pathfinding strategy over time.

Sources:

https://ebookcentral-proquest-com.ezproxy.snhu.edu/lib/snhu-ebooks/reader.action?docID=4850536

https://www.khanacademy.org/science/biology/human-biology/neuron-nervous-system/a/overview-of-neuron-structure-and-function

https://link.springer.com/book/10.1007/978-1-4684-8015-3

https://www.youtube.com/watch?v=59bMh59JQDo

https://towardsdatascience.com/striking-a-balance-between-exploring-and-exploiting-5475d9c1e66e

https://aws.amazon.com/what-is/reinforcement-learning/#:~:text=Reinforcement%20learning%20(RL)%20is%20a,use%20to%20achieve%20their%20goals.