A human being would take to solve this maze by researching and trying to learn the maze and its patterns and even learning some tactics in advance before taking on the maze.  
A human being can also try to predict some of the outcomes when taking a step this can help to reach the end of the maze.  
"In animals and humans has increasingly emphasized the potential contribution of episodic memory, with evidence accruing that estimates of state and action value are based on retrieved memories for specific past action-outcome" (Botvinick, M., Ritter, S., Wang J. X., Kurth-Nelson, Z., Blundell, C., & Hassabis, D. 2019).

The intelligent agent is taking on to maze problem Q-Training Algorithm, Deep reinforcement learning.  
The way it works is by letting the agent try on the maze with no previous data on the maze and how to play it, in each run the agent will receive feedback on its choices of moves selected with a reward system, that is the way it is learning to find a better way to win the maze after each run.  
"Reinforcement learning, a set of methods for learning from rewards and punishments rather than from more explicit instruction" (Botvinick, M., Ritter, S., Wang J. X., Kurth-Nelson, Z., Blundell, C., & Hassabis, D. 2019).

The similarities are that human beings can as well learn as they go with no previous experience, but it will take them a lot of time to be proficient in solving the maze, unlike the intelligent agent that can become very proficient in solving the maze in just a few runs and even surpass pro-human beings in maze problems.

Exploitation is about the continuation of a current course of action, unlike exploration which is the pursuit of alternative courses of action.  
The agent learns the optimal path to the treasure by using two techniques referred to as exploitation and exploration.

In that way, the agent determines to either explore the maze or exploit by using the epsilon value.  
In our case As the agent begins to win more games, its exploration factor is reduced because an assumption is made that the agent has explored nearly all possible paths to the treasure.  
The agent uses an epsilon value to determine the next action.  
The if-statement checks if the epsilon value is greater than a randomly generated number and If the epsilon value is greater than the randomly generated number, then the agent selects a random valid action

{

if np.random.rand() < epsilon:

action = random.choice(valid\_actions)

else:

action = np.argmax(experience.predict(envstate))

}

However, if the epsilon value is not greater than the randomly generated number, the agent will rely on choosing an action that has the highest value in the Q-table for the current state.  
The agent continues to evaluate if it should take a random action or an action from experience for every run.  
As the agent begins to win games, its exploration factor is reduced because an assumption is made that the agent has explored nearly all possible paths to the treasure.

The way I implement deep Q-learning is by using envstate, reward, and game status with this the neural network model is trained in every run.

Reference

1. Botvinick, M., Ritter, S., Wang J. X., Kurth-Nelson, Z., Blundell, C., & Hassabis, D. (2019). Reinforcement Learning, Fast and Slow.