Question1

When the agent doesn’t know the states it can visit and doesn’t know the transition function from each state

Question2

The expression for TD learning:

TD(s, a) = r + gamma \* V(s') - V(s)

And the expression for Q-learning is:

Q(s, a) = r + gamma \* max(Q(s', a'))

The essential difference between these two expressions is that TD learning uses the estimated value of the next state s' to update the value of the current state s, while Q-learning uses the maximum estimated value of all possible actions in the next state s' to update the value of the current state-action pair (s, a). This allows Q-learning to take into account the potential long-term consequences of different actions and better approximate the optimal policy for the game.

Question3

Because in Q learning you can directly extract the policy while in TD learning you have to do so by knowing the transition model T

Question4

Gamma quantifies how much importance we give for future rewards. It’s also handy to approximate the noise in future rewards. Gamma varies from 0 to 1. If Gamma is closer to zero, the agent will tend to consider only immediate rewards. If Gamma is closer to one, the agent will consider future rewards with greater weight, willing to delay the reward.

Question5

Epsilon is related to the epsilon-greedy action selection procedure in the Q-learning algorithm. In the action selection step, we select the specific action based on the Q-values we already have. The epsilon parameter introduces randomness into the algorithm, forcing us to try different actions. This helps not getting stuck in a local optimum.

If epsilon is set to 0, we never explore but always exploit the knowledge we already have. On the contrary, having the epsilon set to 1 force the algorithm to always take random actions and never use past knowledge. Usually, epsilon is selected as a small number close to 0.

Question6

A picture containing chart

Description automatically generated

Based on this plot, we can see that the value of the adaptive epsilon decreases over the episodes, starting at a value of 1 and reaching a value of 0 after approximately 700 episodes. This means that the agent will start by exploring the Gridworld aggressively, taking random actions with high probability, and gradually shift towards exploitation of its learned policy as training progresses. As a result, we would expect the agent to initially make suboptimal decisions and receive lower rewards, but eventually learn to take actions that maximize its reward and perform well in the Gridworld.

Question7

Question8

Approximate Q-learning is more scalable and flexible than the naive Q-learning algorithm, as it can handle large or continuous state spaces and use different types of function approximators to estimate the Q-values of the state-action pairs. It is also more sample efficient, as it can learn from a smaller number of samples and generalize the Q-values of the state-action pairs using function approximation.

Question9

New features for improvement according to the given hint:

1. the number of scared ghosts within one step of Pacman in each of the four cardinal directions

2. the number of scared ghosts within two steps of Pacman in each of the four cardinal directions

3. the number of power pellets within one step of Pacman in each of the four cardinal directions

4. a binary feature indicating whether Pacman has eaten a power pellet in the current state

The function would be:

Q(s, a) = W1 \* (number of scared ghosts within one step of Pacman) + W2 \* (number of scared ghosts within two steps of Pacman) + W3 \* (number of power pellets within one step of Pacman) + W4 \* (binary indicator of whether Pacman has eaten a power pellet)