Q-LEARNING

NOMBRE: HUARACHI NAVARRO SERGIO ALEJANDRO

CARRERA: CIENCIAS DE LA COMPUTACION

NOMBRE APP: Gym-Solution (juego)

Descripción detallada de la Funcionalidad e Implementación del Algoritmo Q-learning:

El código implementa el algoritmo Q-learning, una técnica de aprendizaje por refuerzo, en el contexto de un entorno específico proporcionado por OpenAI Gym, el juego FrozenLake-v1.

Funcionalidad:

1. Entorno y Juego:
    - Utiliza la biblioteca OpenAI Gym para crear un entorno del juego FrozenLake-v1, un problema de navegación en un lago con hielo.
    - El objetivo del juego es que un agente navegue desde el inicio hasta la meta, evitando agujeros en el hielo.
2. Modo de Entrenamiento y Evaluación:
    - El código se adapta según si está en modo de entrenamiento o evaluación.
    - En modo de entrenamiento (is_training = True), el agente realiza exploración y actualiza su tabla Q mediante el algoritmo Q-learning.
    - En modo de evaluación (is_training = False), la tabla Q se carga desde un archivo previamente guardado y el agente toma decisiones basadas en la información aprendida.
3. Exploración y Explotación:
    - Implementa la estrategia de exploración y explotación utilizando la política epsilon-greedy. Durante la exploración, el agente elige acciones aleatorias con probabilidad epsilon, y de lo contrario, elige la acción con el mayor valor Q.
4. Actualización de la Tabla Q:

- La tabla Q se actualiza utilizando la fórmula de actualización del Q-learning, que tiene en cuenta la recompensa inmediata, el factor de descuento y el valor máximo de Q para el próximo estado.

5. Guardado y Carga de la Tabla Q:
    - En modo de entrenamiento, la tabla Q se guarda en un archivo usando la biblioteca pickle.
    - En modo de evaluación, la tabla Q se carga desde el archivo para utilizar el conocimiento previamente adquirido.

Implementación:

1. Inicialización de la Tabla Q:
    - La tabla Q se inicializa como una matriz de ceros si está en modo de entrenamiento.
    - En modo de evaluación, la tabla Q se carga desde un archivo previamente guardado.
2. Bucle Principal de Episodios:
    - Utiliza un bucle principal para ejecutar episodios del juego.
    - En cada episodio, el agente toma decisiones basadas en la estrategia epsilon-greedy y actualiza su tabla Q en modo de entrenamiento.
3. Estrategia de Exploración y Explotación:
    - La probabilidad de exploración (epsilon) se degrada con el tiempo según una tasa específica (epsilon_decay_rate).
    - Permite que el agente explore aleatoriamente al principio y luego se centre más en la explotación de conocimiento adquirido.
4. Guardado de Resultados y Gráficos:
    - Guarda la tabla Q en un archivo al final del entrenamiento (si está en modo de entrenamiento).
    - Grafica y guarda la suma acumulativa de recompensas por episodio para evaluar el rendimiento del agente a lo largo del tiempo.

Elementos Principales utilizados para la Implementación:

1. OpenAI Gym (gym):
    - OpenAI Gym es una biblioteca que proporciona entornos para probar y desarrollar algoritmos de aprendizaje por refuerzo. En este caso, el entorno utilizado es FrozenLake-v1, que representa el juego de navegación en un lago con hielo.
2. NumPy (np):
    - NumPy es una biblioteca para la programación en Python que proporciona soporte para arrays y matrices, así como funciones matemáticas para operar con estos datos. Se utiliza para inicializar y manipular eficientemente matrices, en este caso, para la tabla Q.
3. Pickle (pickle):
    - Pickle es un módulo en la biblioteca estándar de Python que se utiliza para la serialización y deserialización de objetos de Python. En el código, se usa para guardar y cargar la tabla Q en/desde un archivo.
4. Algoritmo Q-Learning:
    - Q-learning es un algoritmo de aprendizaje por refuerzo que permite a un agente aprender a tomar decisiones secuenciales en un entorno desconocido. El agente mantiene una función *Q* que estima la recompensa esperada para cada par estado-acción. A través de iteraciones, el agente ajusta *Q* utilizando la ecuación de Bellman.