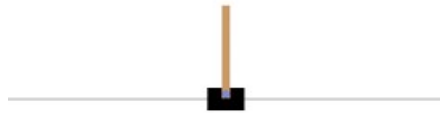


**TAREA CARPOLE:**

En el proceso de actualización de los valores Q en Deep Q-Learning, se utiliza la segunda parte de la ecuación de Bellman, o sea en la fórmula para estimar el nuevo valor  $Q(s, a)$ , no se utiliza directamente el propio valor  $Q(s, a)$ .

La fórmula utilizada es:

$$Q(s, a) = r + \gamma \cdot \text{Max}_{a'} Q(s', a')$$

Esta fórmula actualiza el valor  $Q(s, a)$  hacia el objetivo calculado, que es la suma de la recompensa actual y el valor Q máximo predicho para el siguiente estado  $s'$ , ponderado por el factor de descuento.

Así, aunque no se use directamente el propio valor  $Q(s, a)$  en la fórmula, el proceso de aprendizaje, a través del descenso del gradiente, se encarga de actualizar los valores Q de manera adecuada para converger hacia los valores óptimos.

En el siguiente link puedes descargar el código en Python del problema:

<https://www.kaggle.com/code/mehmetkasap/reinforcement-learning-deep-q-learning-cartpole>

Puedes ver una explicación del código en el link: **Deep Q Learning Networks**

<https://www.youtube.com/watch?v=OYhFoMySoVs>

La tarea consiste en la implementación, entrenamiento y explicación del código.

Se subirá un archivo PDF con el notebook, ejecutado y celdas markdown con explicaciones.

**CALIFICACIÓN:**

- (4 Puntos) El sistema funciona correctamente (se ve la animación y se sostiene el palo).
- (3 Puntos) Se incluyen gráficas de entrenamiento (pueden usar TensorBoard).
- (3 Puntos) Se documenta correctamente lo realizado (recomentable usar LaTeX). Pueden sacar capturas del funcionamiento del algoritmo viendo los enlaces de interés (especialmente el último).

**OTROS ENLACES DE INTERÉS:**

Training & Testing Deep reinforcement learning (DQN) Agent - Reinforcement Learning p.6

<https://www.youtube.com/watch?v=qfovbG84EBg>

Deep Q-Learning (DQL) / Deep Q-Network (DQN) Explained | Python+Pytorch Deep Reinforcement Learning

<https://youtu.be/EUrWGTCGzIA>

Teoría e implementación de DQN (Deep Q-Network). Usando DQN con ROS y Gazebo.

<https://www.youtube.com/watch?v=9GLVB6Trn10>

A friendly introduction to deep reinforcement learning, Q-networks and policy gradients

<https://youtu.be/SgC6AZss478>