Implementación de Aprendizaje por refuerzo - Pong

Santiago Daniel Ribot
UTN - FRBA
Buenos Aires, Argentina
ribot@frba.utn.edu.ar

***Abstract*—El presente trabajo tiene por fin realizar una implementación de la técnica Q-Learning de aprendizaje por refuerzo. Dicha implementación se realizará a través del clásico juego Pong del Atari, donde se le enseñará a un agente las reglas del juego y será el quien decida cuáles son las acciones que debe elegir en cada paso.**

***Index Terms*—Q-Learning, Aprendizaje por Refuerzo, Pong, Atari, Inteligencia Artificial.**

I. Introducción

El Aprendizaje por Refuerzo es un área del aprendizaje automático centrada en descubrir qué acciones debe tomar un agente para maximizar la recompensa recibida. De esta forma, al agente no se le dice qué acciones tomar, sino que es él quien debe experimentar para encontrar las acciones que llevan a maximizar la retribución, ya sea inmediata o no [1].

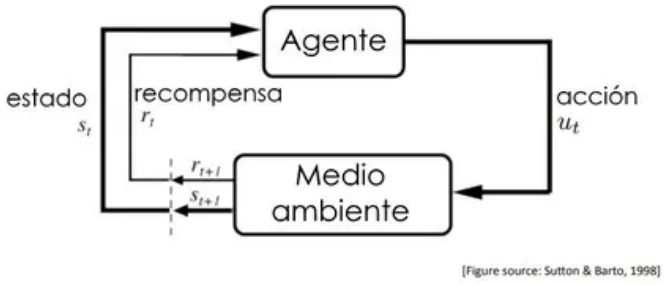

Fig. 1. Modelo de Aprendizaje por Refuerzo

Uno de los desafíos más grandes que existen en el aprendizaje por refuerzo es el paradigma de explotación y exploración: El agente, para encontrar la máxima recompensa, debe explotar aquella que funcionó en el pasado. Pero también tiene que explorar nuevas acciones para asegurarse de que es la máxima recompensa. Tiene que existir un balance, puesto que tomar únicamente explotación o exploración exclusivamente lleva al error.

Además del Agente y las acciones, se pueden identificar otros cuatro elementos claves en el sistema de aprendizaje por refuerzo [2]:

- **Política**: Define el modo en que el agente debe comportarse en un momento definido. Es el conjunto de acciones que va a seguir según el estado que se encuentra. Representa el núcleo del agente y es suficiente para determinar el comportamiento del mismo.
- **Recompensa**: Define el objetivo final del problema que se quiere solucionar. Cada paso de tiempo (cada acción que cambia el estado del medio) da una recompensa (retribución positiva) o un castigo (retribución negativa).
- **Función valor**: La recompensa señalará cual es la mejor acción a seguir en el corto plazo, pero la función valor nos va a representar a mas largo plazo cuales son las acciones para maximizar la recompensa. Es una ”promesa” de recompensa que nos permite tomar acciones más allá del siguiente paso.
- **Medio ambiente**: Es el marco donde se desarrolla la acción. Es un modelo que define cual va a ser la respuesta del medio ante un estado y una acción.

En el caso de estudio se utilizó Q-Learning para enseñarle a un agente cómo jugar Pong. Se seleccionó este juego por su sencillez de reglas y posibilidad de implementarlo con una matriz Q relativamente sencilla. El juego, desarrollado en JavaScript y HTML, fue configurado para que otorgue una recompensa positiva cada vez que le da a la pelota y una negativa cada vez que pierde una vida.

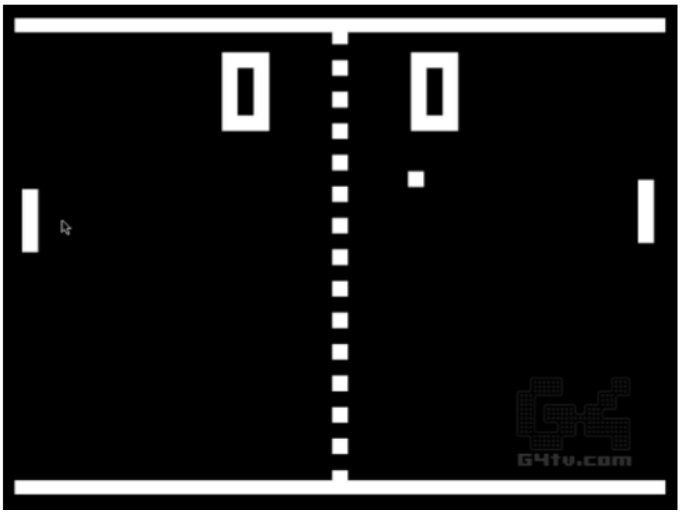

Fig. 2. Pong de Atari

El algoritmo de Q-Learning se basa en resolver el problema a través de una tabla de doble entrada definida por los estados y las acciones posibles. Cada uno de los casilleros de la tabla se va a llenar con un valor que depende de la recompensa recibida y el Agente debe elegir cuál es la siguiente acción a tomar según tenga el valor Q más grande. Los valores Q se van obteniendo a través de la ecuación de Bellman.

El valor Q del estado (s) al aplicar la acción (a) es igual a

$$Q(s,a) = r + \gamma \max_{a'} Q(s',a')$$

Fig. 3. Ecuacion de Bellman

la recompensa obtenida (r) sumado al valor Q correspondiente a ejecutar la mejor acción posible en el siguiente estado, multiplicado por un factor de descuento. Con el factor de descuento podemos definir si le queremos dar más peso a las recompensas en corto o largo plazo [3].

II. Desarrollo

A. Medio ambiente

La implementación de la IA se desarrollará en un Pong de un solo jugador y las acciones posibles del agente son dos: Mover arriba o Mover abajo. Las reglas del juego son simples:

- El jugador (Agente) tiene 3 vidas.
- Si pierde se le dará un castigo (-10 puntos).
- Cada vez que le da a la pelota se le dará una recompensa positiva (10 puntos).
- El juego termina de dos maneras: al perder todas las vidas o al alcanzar 200 puntos.

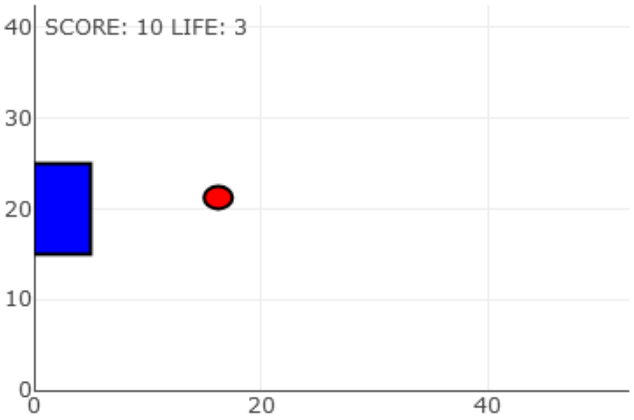

Fig. 4. Juego Pong

Además, y solo durante la etapa de aprendizaje, al perder las 3 vidas se le dará un castigo más fuerte (-20 puntos) para reforzar el castigo.

B. Matriz Q

Al tener un juego que se desarrolla en un plano y con dos elementos, la matriz Q se vuelve un poco más complicada. Para realizarla, se pensó en una pelota definida por dos valores: posición en X y en Y. Por otro lado, la paleta está definida por un solo valor: posición solo en Y, nombrada p. De esta forma, el estado en el que se encuentra el juego nos queda definido por una matriz de tres dimensiones: Estado[x][y][p].

Además, las acciones que se pueden realizar en cada uno de los estados son Arriba o Abajo, así que la Matriz de valores Q nos queda definida de la siguiente manera:

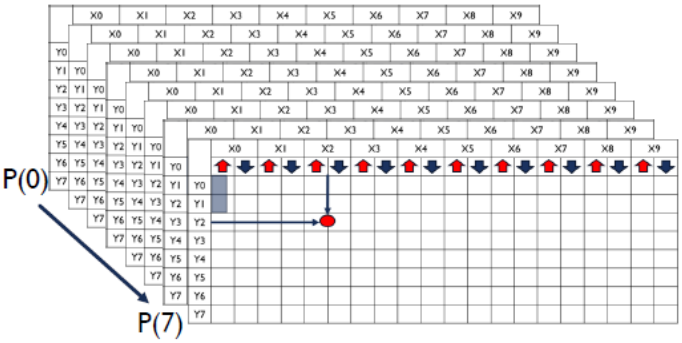

Fig. 5. Ejemplo de la matriz propuesta en el estado [2][2][7]

Cada una de las tablas representa uno de los 7 estados posibles de la paleta, mientras que dentro de cada tabla se define el valor de la pelota. De esta forma, para una determinada posición de la pelota y de la paleta, nos quedamos únicamente con un par de valores Q correspondientes a las acciones posibles. De estos valores, el agente puede elegir el que tenga el mayor Q (explotación) o una acción aleatoria (exploración).

C. Clase Agente

Es quien tiene la Matriz Q, la modifica a través de la ecuación de Bellman y decide cuál es la siguiente acción a tomar. Ademas tiene definidas el factor de descuento, la tasa de aprendizaje y el ratio de exploración.

La tasa de aprendizaje es un modificador que le damos a los valores Q que se van sumando a la tabla. De esta forma el valor Q actual es igual al pasado más el obtenido multiplicado por la tasa de aprendizaje.

$$Q'(s,a) := Q(s,a) + \alpha(r + \gamma \max_{a'} Q(s',a') - Q(s,a))$$

Fig. 6. Función valor

El ratio de exploración es la probabilidad de elegir la mejor acción de la matriz o una acción aleatoria. Nos permite solventar el paradigma de explotación vs exploración.

D. Clase Ambiente

Es quien implementa la lógica y reglas del juego. Actualiza la pantalla para que se vea la animación del juego y aplica las acciones que son enviadas por el agente. Ademas, es quien cambia de estado al sistema y devuelve una recompensa.

III. Funcionamiento

Llenar la matriz con los valores Q es el método de aprendizaje de la máquina. Para ello se debe realizar un algoritmo cíclico hasta obtener una buena cantidad de muestras.

Este algoritmo lo repetimos durante una cantidad determinada de partidas, donde cada partida puede terminar por tres motivos: llegar al máximo puntaje (200), perder todas las vidas (3) o llegar a la cantidad máxima de pasos permitida (3000).

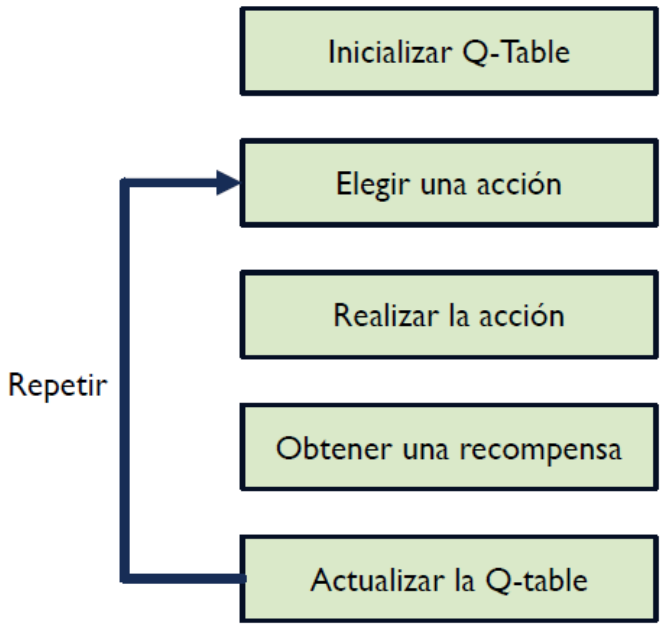

Fig. 7. Algoritmo de Q-Learning

Esta cantidad máxima de pasos se puso para evitar que el agente se quede jugando indefinidamente.

A lo largo del aprendizaje se va variando el ratio de exploración de menor (más acciones aleatorias, más exploración) a mayor (más greedy, más explotación) para hacerlo más efectivo.

IV. Conclusiones

El presente trabajo demuestra cómo el aprendizaje por refuerzo, específicamente el algoritmo Q-Learning, puede aplicarse exitosamente para enseñar a un agente a jugar un juego sencillo como Pong. Mediante el uso de una matriz Q y la ecuación de Bellman, el agente es capaz de aprender estrategias efectivas para maximizar la recompensa y minimizar las penalidades, balanceando adecuadamente la exploración de nuevas estrategias y la explotación de las ya conocidas.

A través del desarrollo y la implementación, se observó la importancia de los parámetros como la tasa de aprendizaje y el ratio de exploración para lograr un aprendizaje eficiente. Estos factores permiten un entrenamiento progresivo, pasando de un enfoque exploratorio a uno más basado en políticas óptimas.

Si bien los resultados obtenidos son buenos para un primer intento, el enfoque elegido tiene ciertas limitaciones. Por ejemplo, el espacio de estados crece rápidamente con problemas más complejos, lo cual puede hacer que la matriz Q consuma grandes cantidades de memoria y sea difícil de gestionar. En futuras iteraciones, sería interesante explorar métodos más avanzados como Redes Neuronales Profundas en combinación con aprendizaje por refuerzo, como ocurre en Deep Q-Learning, para abordar estos desafíos.

Por último, cabe destacar que esta implementación se realizó con JavaScript en un entorno de desarrollo web, que no es la forma más adecuada de hacerlo. En su lugar, es conveniente realizarlo con Python por su mejor uso de matrices y tuplas, además de su activa comunidad y constante desarrollo de nuevas librerías.

References

[1] Miguel Silva (2019). Aprendizaje por Refuerzo: Introducción al mundo del RL. Publicación en la web www.medium.com.
[2] Richard S. Sutton and Andrew G. Barto (2018). Reinforcement Learning: An introduction.
[3] Markel Sanz Ausin (2020). Introduccion al aprendizaje por refuerzo parte 2: Q-Learning. Publicacion en la web www.medium.com.