1. INTRODUCCIÓN

Los seres vivos muestran siempre un tipo de comportamiento del modo que realizan una acción como respuesta a las señales que reciben del entorno en donde viven. Algunos, además, cambian su comportamiento con el tiempo, por lo que se puede decir que han aprendido del entorno. La idea de aprender a partir de interactuar con el entorno es probablemente la primera que ocurre cuando se piensa sobre la naturaleza del aprendizaje. La realización de estas actividades produce una riqueza de información sobre las relaciones causa y efecto, sobre las consecuencias de las acciones, y sobre qué hacer con el fin de alcanzar objetivos.

Aprender de la interacción con el entorno es una idea fundamental que subyace a casi todas las teorías del aprendizaje dentro del ámbito de la Inteligencia Artificial. Por lo que la utilización de un enfoque capaz de aprender del entorno, como es el Aprendizaje por Refuerzo (ApR), es un problema cuya importancia ha ido aumentando de forma progresiva en los últimos años, y mucho más ahora con el auge de los sistemas cambiantes, online, interactivos con grandes capacidades de cálculo y almacenamiento.

En un sistema estándar, en el ámbito del ApR, existe un agente situado en un entorno que puede percibir, mediante un conjunto de sensores, y transformarlo con las acciones que pueden realizar un conjunto de actuadores. Si consideramos un entorno discreto, cada interacción del agente recibe como entrada una colección de valores desde sus sensores que configurarán su estado actual ($s \in S$), y selecciona una acción ($a \in A$) de todas las posibles en dicha situación. La ejecución de una determinada acción (mediante sus actuadores) cambia el estado S', recibiendo una señal de refuerzo o recompensa ($r \in R$). En Fig. 1.1 se ilustra el ciclo de este aprendizaje.

1.1 MOTIVACIÓN

Una política óptima es una política que maximiza la recompensa total esperada. La tarea del ApR es utilizar las recompensas observadas para aprender una política óptima para el entorno donde el agente se encuentra. En Fig. 1.2 se ilustra la evolución de la recompensa total por episodio (o experiencia), mostrando como converge en la política óptima a partir del episodio 35.

En una gran cantidad de entornos complejos, el aprendizaje por refuerzo es el único enfoque posible para entrenar un agente para que funcione de forma adecuada.

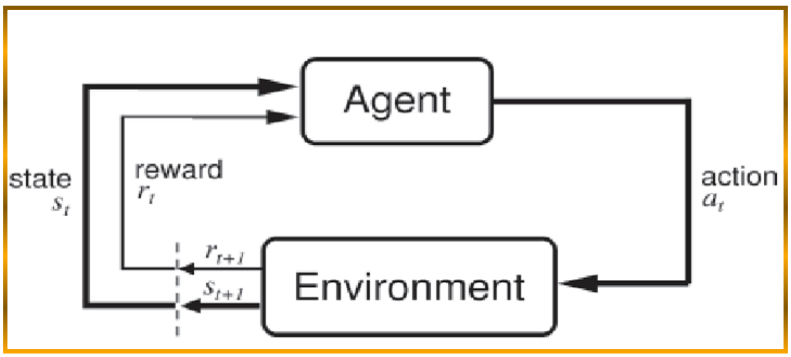

Figura 1.1: Ciclo del aprendizaje

Hay tres diseños de agentes en este enfoque:

Agente basado en la utilidad

Este agente aprende una función de utilidad que la usará para seleccionar aquellas acciones que maximizarán la utilidad.

Agente de QL

Aprende una función acción-valor, la cual le da la utilidad esperada al realizar una acción determinada en un estado dado.

Agente reactivo

Aprende una política que da lugar a una relación directa entre los estados y las acciones.

El principal inconveniente del agente basado en la utilidad es que necesita un modelo del entorno. Sin embargo, el agente de QL tiene la posibilidad de comparar los valores de sus posibles acciones sin tener que saber sus resultados, por lo que es una política que no necesita un modelo del entorno. En el aprendizaje pasivo, la política del agente está fijada y la tarea es aprender las utilidades de los estados, mientras que en el aprendizaje activo el agente aprenderá más de su entorno cuantas más experiencias tenga en él.

Por lo tanto, en el presente Trabajo Fin de Grado (TFG) se desarrolla la técnica para implementar un agente de QL, así como la visualización de su aprendizaje.

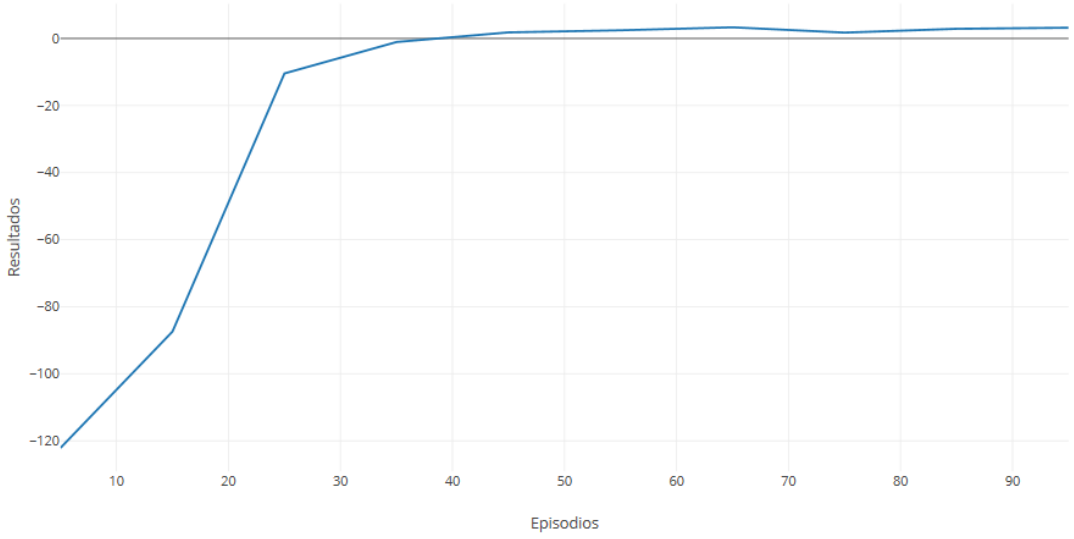

Figura 1.2: Evolución del aprendizaje

1.2 PROBLEMA

El problema de aprender una política óptima para un entorno determinado donde se encuentre el agente pasa por calcular una matriz-Q a partir de la experiencia. Esta matriz-Q representa nada más y nada menos que a la función acción-valor que aprende el algoritmo QL.

El algoritmo QL es el principal partícipe de este cálculo con la llamada ecuación de Bellman:

$$Q(a,s) \leftarrow Q(a,s) + \alpha\big(R(s) + \gamma max Q_{a'}\,(a',s') - Q(a,s)\big) \qquad (1.1)$$

Siendo (a,s) la función-Q en un determinado estado s con una determinada acción a, $R(s)$ es la función de recompensa, que especifica la recompensa de cada estado, α la velocidad de aprendizaje y γ el factor de descuento. Los parámetros α y γ se explicarán con mayor extensión más adelante, y ajustándolos podemos determinar la velocidad a la que el agente aprende y la importancia de cada acción que realiza el agente, respectivamente.

Para mostrar el aprendizaje de este agente en un entorno determinado, se propone desarrollar un entorno 2D donde se visualice a un agente aprendiendo la política óptima con el fin de encontrar un tesoro o premio que se encontrará en un punto aleatorio del mapa. Como se ha explicado antes, la principal característica brillante de este algoritmo es que se adapta a cualquier entorno introducido por el usuario, por lo que se podrá observar la gran maleabilidad de éste.

1.3 ESTRUCTURA DEL DOCUMENTO

Este TFG está estructurado con los siguientes capítulos:

Capítulo 1: Introducción

Se muestra una breve presentación del ApR, así como su motivación para su uso en este proyecto y problema general que se intenta solucionar.

Capítulo 2: Objetivos

En este apartado se muestran los objetivos principales y específicos para este TFG para solucionar de forma óptima la problemática de este TFG. También se mostrarán los medios hardware y software que han sido utilizados para el desarrollo del presente TFG.

Capítulo 3: Antecedentes

Se muestra un análisis de la situación actual o estado del arte de todos los elementos del proyecto para entender su autonomía y funcionamiento. Entre los elementos se expone: ApR, algoritmo QL, etc.

Capítulo 4: Metodología

Se explica el método de desarrollo utilizado para el desarrollo del TFG y las distintas fases llevadas a cabo para la resolución de los problemas.

Capítulo 5: Resultados

En esta sección se explica la aplicación del método de desarrollo presentado en el capítulo 4 con elementos como modelos, diagramas, especificaciones, etc. Se muestra el algoritmo QL implementado, el servidor y el cliente integrados, así como sus resultados experimentales.

Capítulo 6: Conclusiones

Se analizan y presentan todas las conclusiones obtenidas tras la completa realización del TFG. También se incluyen posibles mejoras y extensiones del mismo.

Capítulo 7: Bibliografía

Se muestran todas las referencias usadas para el desarrollo del TFG ordenadas por orden alfabético del primer apellido del primer autor.