Introducción al aprendizaje por refuerzo

PID_00275578

Luis Esteve Elfau

Tiempo mínimo de dedicación recomendado: 3 horas

Universitat Oberta de Catalunya

Luis Esteve Elfau

Ingeniero Superior de Telecomunicaciones por la UPC - ETSETB TelecomBCN (2013) y emprendedor experimentado con un amplio historial laboral en la industria de la educación superior. Experto en Inferencia Estadística, Procesado de señal, GNSS, Matlab, Emprendimiento y Gestión de Equipos. Actualmente sus intereses se centran en la aplicación de las técnicas del Aprendizaje por Refuerzo (RL) en diversos campos de la ciencia y de la vida real.

El encargo y la creación de este recurso de aprendizaje UOC han sido coordinados por el profesor: Jordi Casas Roma

Primera edición: septiembre 2021

Av. Tibidabo, 39-43, 08035 Barcelona
Autoría: Luis Esteve Elfau
Producción: FUOC

Índice

Introducción

En este módulo veremos una introducción al aprendizaje por refuerzo (que se suele abreviar como RL, *reinforcement learning* en inglés). El aprendizaje por refuerzo es una rama del aprendizaje automático (abreviado como ML, *machine learning* en inglés) que se caracteriza por ser una aproximación computacional al aprendizaje por interacción.

El aprendizaje por interacción es un elemento presente en la naturaleza, ya que tanto los animales como los humanos en sus primeros meses de vida aprenden mediante un proceso de prueba y error en busca de unos objetivos, unas recompensas (comida, calor, ...) que pueden no llegar de inmediato, sino después de una secuencia de acciones que deben ser aprendidas. Estas dos características, el proceso de prueba y error y la existencia de recompensas retrasadas en el tiempo, son las dos características principales que diferencian al aprendizaje por refuerzo del resto de métodos de aprendizaje.

En este curso abordaremos este tipo de aprendizaje desde el punto de vista del diseño de un agente (que básicamente es el ente abstracto que pretende aprender, el algoritmo que queremos diseñar) que debe interactuar con el entorno para lograr un objetivo.

El área del aprendizaje por refuerzo ha ganado muchos adeptos estos últimos años debido, en parte, a los logros obtenidos en algunas disciplinas. Casos famosos como el de Deepmind (empresa comprada por Google en 2014) y su agente AlphaGo (primer programa de ordenador en ganar a un jugador profesional de Go y posteriormente al campeón mundial de dicha disciplina) han tenido repercusión en los medios de comunicación.

Aunque pueda parecer lo contrario, el aprendizaje por refuerzo no es una disciplina nueva, sus orígenes se remontan a la década de 1980. Incluso algunas de sus ideas ya se habían aplicado en otras áreas de la ciencia mucho antes, como las teorías sobre control óptimo desarrolladas a finales de la década de 1950. Pero es en los últimos años, junto con la explosión del aprendizaje profundo (abreviado como DL, *Deep Learning* en inglés), cuando se han producido los mayores avances en este campo.

De esta forma, iniciaremos el módulo dando una visión histórica sobre la motivación que ha llevado al desarrollo del aprendizaje automático y del aprendizaje por refuerzo. A continuación, veremos cómo partes de los fundamentos y características del aprendizaje por refuerzo aparecen en diversas disciplinas de muchas de las áreas más importantes de la ciencia. Seguiremos analizando

cuáles son las características básicas que lo caracterizan y cómo encaja entre las diferentes ramas del aprendizaje automático.

En el siguiente apartado nos centraremos en definir los elementos que forman cualquier problema de aprendizaje por refuerzo. De esta forma estudiaremos la señal de recompensa, el entorno y definiremos el concepto de estado, conceptos primordiales todos ellos a la hora de diseñar un agente de aprendizaje por refuerzo.

En el tercer apartado definiremos los elementos que puede contener un agente: la política (*policy* en inglés), la función de valor (*value function* en inglés) y el modelo; y haremos hincapié en la taxonomía de los agentes dependiendo de si disponen de uno o de varios de estos elementos.

En el apartado siguiente, daremos una visión general de algunos de los puntos clave que aparecen al diseñar un agente de aprendizaje por refuerzo y que iremos analizando con mayor profundidad a lo largo del curso. Entre dichos puntos cabe resaltar: la diferencia entre aprendizaje y planificación (*planning* en inglés), la importancia del balance entre las fases de exploración y explotación en el diseño de cualquier agente, y la diferencia entre los conceptos de predicción y control, así como la manera en que se interrelacionan.

Objetivos

En este módulo encontraremos las herramientas necesarias para ser capaces de asimilar los siguientes objetivos:

1. Entender qué es el aprendizaje por refuerzo, su motivación, cómo encaja dentro de la rama del aprendizaje automático y su conexión con disciplinas de otras áreas de la ciencia.
2. Comprender cuál es la tipología de los problemas que pretendemos resolver con el aprendizaje por refuerzo. Saber identificar los elementos que forman este tipo de problemas y poder identificarlos en un problema real.
3. Conocer los principales componentes de un agente. Distinguir si un agente dispone o no de cada uno de esos componentes.
4. Conocer los tipos de agentes que existen y saber clasificarlos.
5. Saber identificar los problemas clave que nos encontraremos al afrontar un problema de aprendizaje por refuerzo y que iremos resolviendo a lo largo del curso.

1. Sobre el aprendizaje por refuerzo

1.1. Motivación

Antes de definir qué es el aprendizaje por refuerzo, empezaremos analizando la motivación histórica que ha llevado al desarrollo de este tipo de aprendizaje. Si lo analizamos desde un punto de vista de alto nivel, una forma de ver esta evolución es compararla con el resto de automatizaciones de tareas que se han realizado a lo largo de la historia.

Hace ya muchos años que se inició el proceso de automatizar tareas físicas repetitivas utilizando máquinas, fue la denominada *Revolución Industrial*, que comprende tanto la Primera Revolución Industrial (1750-1850), donde las máquinas empezaron a realizar las tareas que hasta entonces se reservaban para personas o animales (por ejemplo, el ferrocarril reemplazó a los caballos en el transporte de mercancías) como la Segunda Revolución Industrial o era de las máquinas (*machine age* en inglés) (1870-1940), donde la aparición de nuevos combustibles (por ejemplo, el petróleo) y energías (la electricidad) facilitó la evolución de la maquinaria existente y la aparición de nuevos modelos de máquinas que propiciaron un gran aumento de la productividad.

Cronología de la automatización

- 1ª Revolución Industrial (1750-1850)
- 2ª Revolución Industrial (1870-1940)
- Revolución Digital (1950-2010)
- ¿Revolución AI?

La segunda ola de automatización, que empezó entre las décadas de 1950 y 1960 y que se prolonga hasta la segunda década del siglo XXI, es lo que podríamos llamar la *Revolución Digital* (también denominada Tercera Revolución Industrial) en la que se produjo un proceso de automatización similar pero en lugar de automatizar tareas físicas, se realizó una automatización de tareas mentales repetitivas. Un ejemplo muy ilustrativo es la aparición de la calculadora. Sabemos cómo hacer todo tipo de operaciones aritméticas (sumas, restas, multiplicaciones, etc.), pero al programar estas operaciones en una calculadora podemos delegar en un futuro la realización de estas tareas repetitivas de índole puramente mental. En ambos casos (el ferrocarril en la Revolución Industrial y la calculadora en la Revolución Digital) fuimos los humanos los que nos planteamos un problema (qué queríamos hacer), su solución (cómo hacerlo) y luego la implementamos en una máquina.

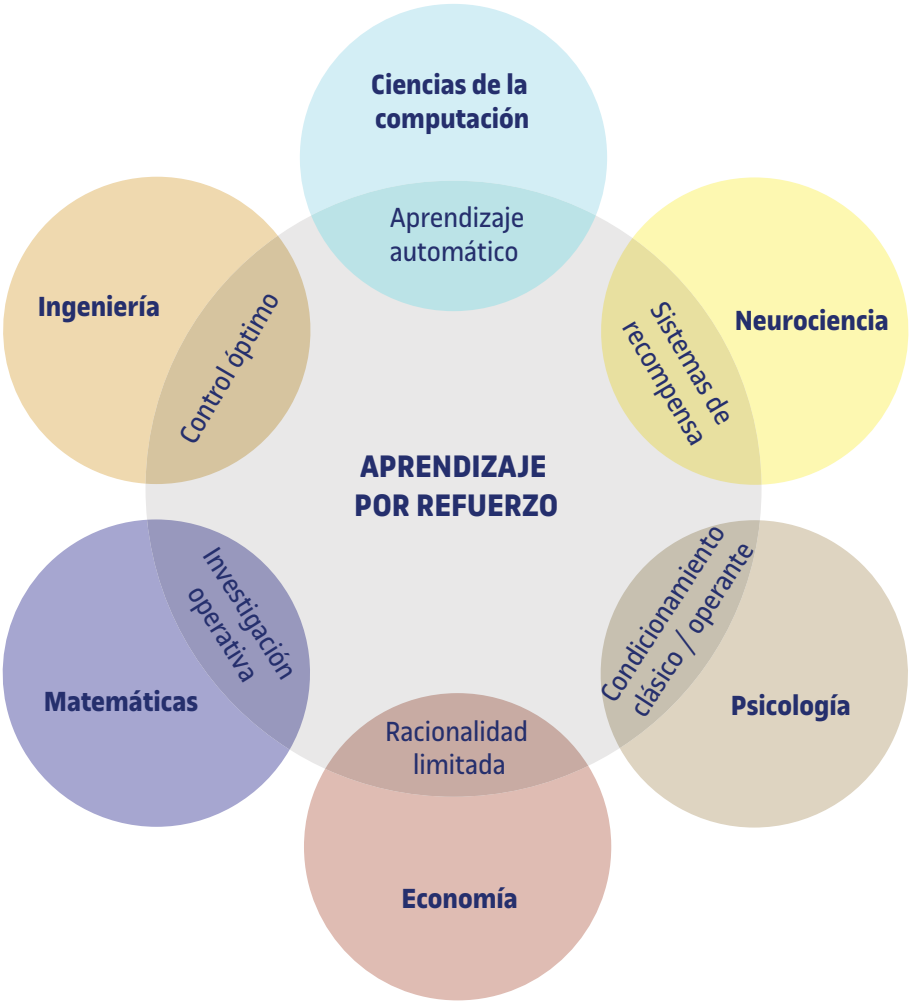
El siguiente paso es definir un problema y que sea la máquina quien lo resuelva por sí misma. Para conseguir dicho objetivo la máquina necesita aprender, pero para ello ¿qué información hay que introducirle? Podemos seguir introduciendo el propio conocimiento, ya sean soluciones mentales o físicas, tal y como se hizo en las dos primeras revoluciones, pero además tenemos que añadir algún tipo de conocimiento sobre cómo aprender, así como propor-

cionar los datos para que la máquina aprenda por sí sola. Esta es la base del aprendizaje automático y, por ende, del aprendizaje por refuerzo.

1.2. Las diferentes caras del aprendizaje por refuerzo

Otra forma de acercarnos al aprendizaje por refuerzo es ubicarlo dentro de las diversas áreas de la ciencia. Si analizamos el diagrama de la figura 1 podemos ver que una de las características del aprendizaje por refuerzo es que se encuentra en la intersección de diversas áreas científicas. El diagrama ilustra el hecho de que en cada una de estas diferentes áreas o campos de la ciencia hay una disciplina perteneciente a ese campo que trata de estudiar el mismo problema que veremos en el aprendizaje por refuerzo.

Figura 1. Las diferentes caras del aprendizaje por refuerzo. Diagrama de Venn que ilustra como el aprendizaje por refuerzo se da en muchos campos de la ciencia



Fuente: Adaptación de la gráfica de David Silver, 2015

¿Cuál es ese problema? Esencialmente es el problema de la toma de decisiones. Tratar de entender la forma óptima de tomar decisiones es un problema recurrente que surge una y otra vez en la mayoría de las áreas científicas, y eso hace del aprendizaje por refuerzo una disciplina tan amplia e interesante.

En **ciencias de la computación** analizaremos este problema bajo el paraguas del aprendizaje automático, concretamente dentro del aprendizaje por refuerzo, que es la materia que estudiaremos en este curso. Pero si nos centramos en el mundo de la **ingeniería**, existe una rama, llamada control óptimo (*optimal control* en inglés), que esencialmente estudia el mismo tipo de problemas y utiliza los mismos métodos de resolución con nombres distintos, pero la finalidad es la misma, cómo decidir de manera óptima una secuencia de acciones para obtener los mejores resultados al final de un periodo de tiempo.

En **neurociencia**, uno de los principales estudios en las últimas décadas trata de comprender cómo el cerebro humano realmente toma decisiones complejas. Esta toma de decisiones se basa en el sistema de recompensas (*reward system* en inglés) del cerebro humano. En dicho sistema la dopamina, un neurotransmisor que se encuentra repartido por varias zonas del cerebro, tiene un papel fundamental en la señalización del placer esperado de esos posibles eventos futuros y por tanto influye activamente en la toma de estas decisiones. Así pues, los métodos de aprendizaje por refuerzo de los que hablaremos en este curso también son la base para la toma de decisiones realizadas por humanos.

En **psicología**, dentro de la rama del conductismo, se han realizado múltiples estudios a lo largo de la historia para tratar de entender cómo los animales toman decisiones (sistemas de recompensas y castigos que dan lugar a la teoría del condicionamiento simple de Pavlov y el condicionamiento operante de Skinner) y la teoría que subyace es esencialmente la misma que en el aprendizaje por refuerzo.

En **matemáticas**, existe una rama que se encarga de estudiar las herramientas matemáticas utilizadas en el control óptimo en ingeniería, dicha rama se conoce con el nombre de investigación operativa (*operations research* en inglés).

Finalmente, en **economía** tenemos la teoría de juegos, la teoría de la utilidad y la racionalidad limitada (*bounded rationality* en inglés), en las que nuevamente se estudia cómo y por qué las personas toman decisiones para tratar de optimizar su utilidad.

Vemos, por tanto, que los conceptos que estudiaremos en este curso son de interés general para muchas áreas diferentes de la ciencia.

1.3. Características básicas del aprendizaje por refuerzo

Los seres humanos, al igual que otros seres del reino animal, tenemos la capacidad de aprender al interactuar con el entorno, que es una de las características básicas del aprendizaje por refuerzo, y esto difiere de otros tipos de aprendizaje.

Una diferencia es que este tipo de aprendizaje es activo en lugar de pasivo, ya que el sujeto (la persona en el caso de los seres humanos, o el agente en el caso de los algoritmos de aprendizaje por refuerzo) interactúa con el entorno y el entorno responde a la interacción. De esta forma, en el aprendizaje por refuerzo tenemos un agente, que es nuestro sistema de aprendizaje, que toma decisiones y realiza ciertas acciones. Estas acciones son absorbidas por el entorno que es, básicamente, todo lo que es externo al agente y dicho entorno responde enviando una observación y luego este ciclo continúa repetidamente.

El agente puede tomar nuevas decisiones y realizar nuevas acciones y el entorno puede o no cambiar dependiendo de estas acciones y en consecuencia las observaciones que emite también pueden o no cambiar. Este ciclo interactivo, e iterativo, es lo que permite al agente ir aprendiendo paulatinamente.

Además, al igual que el aprendizaje humano, el aprendizaje por refuerzo está dirigido a la consecución de objetivos, es decir, las acciones del agente deben encaminarse a la consecución de un objetivo final aunque no conozca exactamente cómo alcanzar ese objetivo por adelantado, aprendiendo mediante el procedimiento de prueba y error sin la necesidad de tener ejemplos del comportamiento óptimo.

Existen básicamente dos tipos de objetivos que se pueden diferenciar. Un objetivo es encontrar soluciones desconocidas anteriormente, por ejemplo al diseñar un agente que pueda jugar mejor que cualquier ser humano a un determinado juego (Go, ajedrez, etc.). Para poder conseguir este objetivo el agente debe encontrar nuevas formas de jugar nunca antes vistas por cualquier humano. El otro objetivo es poder aprender rápidamente en un nuevo entorno, como por ejemplo un robot que camina y ha sido entrenado en un tipo de terreno y se le cambia de terreno. En este caso el robot debe adaptarse al nuevo entorno y aprender en línea cómo caminar en el nuevo terreno. El aprendizaje por refuerzo busca proporcionar algoritmos que puedan manejar ambos tipos de objetivos.

Así pues, podríamos definir el aprendizaje por refuerzo de la siguiente forma.

El **aprendizaje por refuerzo** es la ciencia de tomar decisiones a partir de la interacción con el entorno.

Esto requiere, como veremos más adelante, que analicemos muchos conceptos tales como el tiempo y las consecuencias a largo plazo de las acciones tomadas, la predicción del futuro para tratar de adelantarnos a las consecuencias de dichas acciones, la recopilación activa de experiencia a partir de la interacción y cómo hacer frente a la incertidumbre inherente en problemas ruidosos o parcialmente ocultos.

1.4. Ramas del aprendizaje automático

Dentro del campo de la informática (o ciencias de la computación), si nos centramos en la rama del aprendizaje automático, la mayoría de textos en la literatura actual dividen el aprendizaje automático en tres disciplinas:

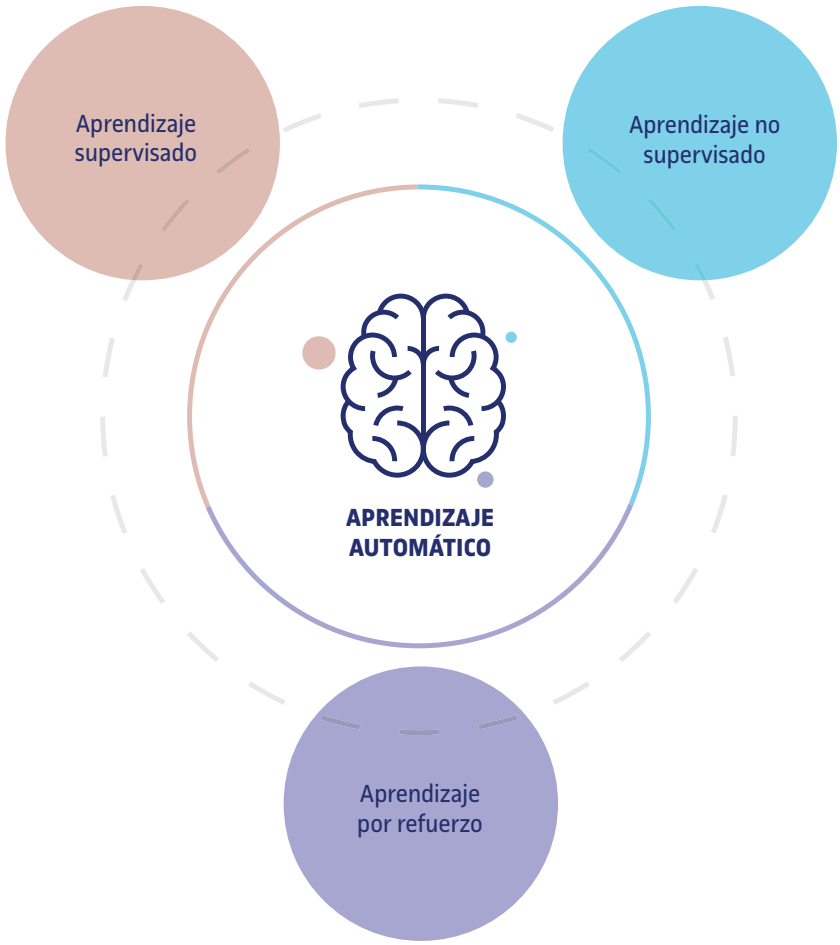
1) el aprendizaje supervisado (*supervised learning* en inglés),

2) el aprendizaje no supervisado (*unsupervised learning* en inglés),

3) el aprendizaje por refuerzo.

Podemos ver esta clasificación en el diagrama de la figura 2.

Figura 2. Las ramas del aprendizaje automático



En el aprendizaje supervisado, una de las disciplinas que más se han desarrollado en parte gracias al enorme trabajo desarrollado en aprendizaje profundo, el objetivo es, en pocas palabras, encontrar un mapeo entre unos datos de entrada (ejemplos) y unos datos de salida (etiquetas de esos ejemplos) de manera que ese mapeo aprendido a partir del entrenamiento con los datos de ejemplo también pueda generalizar nuevas entradas que nunca se hayan visto antes.

Por otro lado, en el aprendizaje no supervisado no se tienen ejemplos etiquetados, por lo que no se puede realizar un mapeo entre estos datos y unas etiquetas. Aquí el objetivo es aprender a estructurar los datos (encontrar algún tipo de estructura inherente a dichos datos, pero no evidente a simple vista) para poder comprenderlos mejor (por ejemplo, clasificándolos en grupos).

Históricamente el aprendizaje por refuerzo ha sido a veces clasificado como parte de una de estas dos disciplinas (aprendizaje supervisado o no supervisado) o incluso una mezcla de ambas, pero este tipo de aprendizaje tiene una serie de características que finalmente han hecho que se considere como una tercera disciplina separada dentro del aprendizaje automático.

La primera característica que se debe remarcar es que no existe un supervisor que nos diga cuál es la acción correcta en cada caso. En cada estado (situación) el agente obtiene, tras cada acción (interacción con el entorno), una señal que llamamos recompensa. La señal de recompensa proporciona una idea de lo bueno o malo que es algo (sea una acción o un estado) cuando se compara con otra cosa, pero no nos dice exactamente qué hacer, no nos proporciona una verdad absoluta como lo hacen las etiquetas en el aprendizaje supervisado.

La segunda característica del aprendizaje por refuerzo que lo diferencia del resto de paradigmas de aprendizaje automático es que dicha recompensa puede no ser instantánea, puede obtenerse retrasada en el tiempo, por lo que a veces al escoger una acción, esta acción no produce sus frutos inmediatamente sino que mucho más tarde conduce a la recompensa deseada. Es decir, la mayoría de veces no podemos ver si la acción elegida es buena o mala hasta cierto tiempo después. De hecho puede pasar que una acción obtenga un beneficio o recompensa inmediatos, pero al cabo de un tiempo nos lleve a una situación catastrófica, con lo que el beneficio final (también llamado retorno) será negativo. Por tanto, el tiempo y la secuencialidad de las acciones son muy importantes, ya que si se elige una acción ahora, podría ser imposible deshacer esa acción más adelante.

Esto nos lleva a la tercera característica y es que, al tratarse de un sistema de toma de decisiones secuencial, donde el agente recibe unos datos (observación del entorno), toma una decisión y ejecuta una acción a la que el entorno responde con nuevos datos, y así sucesivamente, estos datos secuenciales no pueden ser considerados independientes e idénticamente distribuidos (i.i.d. *independent and identically distributed* en inglés) tal y como sucede en la mayo-

Datos i.i.d.

Datos i.i.d. es la abreviatura de datos independientes e idénticamente distribuidos (*independent and identically distributed* en inglés)

ría de problemas que se estudian en el aprendizaje supervisado y no supervisado. Un ejemplo típico de esta situación serían los datos que recibe un agente que juega a un emulador de videojuegos. Después de cada acción (un movimiento del mando de la consola u ordenador) los datos que recibe del entorno (las imágenes de la pantalla) son muy parecidos a los que recibió en la acción anterior y por tanto están altamente correlacionados.

Por último, la cuarta característica que diferencia al aprendizaje por refuerzo del resto de los paradigmas del aprendizaje automático es que las acciones del agente afectan a la interacción posterior y por tanto a los datos que recibe del entorno. Es decir, el agente influye directamente en los datos que recibe del entorno. Esta característica, no presente en el aprendizaje supervisado ni en el no supervisado, es tal vez la más diferencial y genuina del aprendizaje por refuerzo. Siguiendo con el ejemplo del agente que juega a videojuegos, dependiendo de si mueve el mando hacia un lado o hacia otro los datos que recibe son distintos. Por tanto, la acción (el movimiento del mando en este caso) condiciona completamente los datos que recibe (la imagen de la pantalla).

Este conjunto combinado de características es lo que diferencia el aprendizaje por refuerzo del resto de aprendizajes.

2. Definiendo el problema

Como veremos en este apartado, y en apartados posteriores, los elementos que caracterizan a cualquier problema de aprendizaje por refuerzo son:

- la señal de recompensa (*reward signal* en inglés),
- el entorno (*environment* en inglés),
- los estados (*states* en inglés),
- el agente (*agent* en inglés).

De estos cuatro elementos, la señal de recompensa, el entorno y los estados definen el problema que queremos resolver, mientras que el agente es la solución a dicho problema, ya que es el algoritmo que tenemos que diseñar para tomar decisiones.

Como el objetivo de este apartado es definir el problema que se debe resolver, nos centraremos en analizar los tres primeros elementos y dejaremos el estudio del agente para el siguiente apartado de este módulo.

2.1. La señal de recompensa

Como hemos comentado en el apartado anterior, los problemas que pretende resolver el aprendizaje por refuerzo están orientados a la consecución de un objetivo final. De esta forma, una de las señales más importantes que nos ayudará a diseñar algoritmos capaces de alcanzar estos objetivos es la señal de recompensa R_t.

La **señal recompensa** R_t es un valor escalar que el agente recibe en el instante *'t'* después de cada acción que ejecuta.

Aunque pueda parecer que un escalar no es suficiente para definir los objetivos en problemas complejos, se ha comprobado que en la mayoría de casos se puede llevar a cabo una reducción de la complejidad del problema y encontrar una recompensa escalar que caracterice completamente la consecución del objetivo.

De esta forma se establece el siguiente teorema:

Hipótesis de la recompensa (*reward hypothesis* en inglés)

Cualquier objetivo puede describirse mediante la maximización de la esperanza (media estadística) de la recompensa acumulada.

Dicha recompensa acumulada, también llamada retorno (*return* en inglés), puede definirse para el instante *'t'* como:

$$G_t = R_{t+1} + R_{t+2} + R_{t+3} + ... \quad (1)$$

Es decir que se define la señal de **retorno** G_t como el **valor acumulado de todas las recompensas futuras**. Remarcar que como dichas recompensas son aleatorias el retorno también lo es y por eso el objetivo del agente es maximizar su valor esperado. Por otro lado, dado que el número de episodios (cada vez que se recibe una recompensa después de una acción) puede no ser finito, el sumatorio de recompensas podría no converger y obtener un valor de retorno infinito. En este caso conviene redefinir el retorno como:

$$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + ... \quad (2)$$

En la ecuación anterior se introduce un parámetro γ llamado factor de descuento cuyo valor está acotado (concretamente $0 \leq \gamma \leq 1$) y permite la convergencia del valor del retorno en los problemas donde el número de recompensas futuras es infinito. El factor de descuento, además, tiene una influencia en el proceso de aprendizaje tal y como veremos en un módulo posterior dedicado a los procesos de decisión de Markov.

Para poder satisfacer la *hipótesis de recompensa* definida anteriormente debemos ser muy cuidadosos a la hora de diseñar la señal de recompensa. Veamos unos ejemplos representativos.

Ejemplo 1. Laberinto

En el caso de tener que diseñar un algoritmo que encuentre el camino de salida de un laberinto en el menor tiempo posible, una forma válida de diseñar la señal de recompensa podría ser la siguiente:

$$\begin{cases} R_t = -1 & \text{para cualquier movimiento excepto si se alcanza la salida} \\ R_t = 0 & \text{si alcanza la salida} \end{cases}$$

De esta forma el valor del retorno siempre será negativo, y contra menos movimientos realice el agente para salir del laberinto mayor será su valor.

Ejemplo 2. Jugador de ajedrez

En el caso de tener que diseñar un agente que juegue al ajedrez, una posible definición de la señal de recompensa para conseguir dicho objetivo podría ser la siguiente:

$$\begin{cases} R_t = 0 & \text{para cualquier movimiento excepto si se gana la partida} \\ R_t = 1 & \text{si se gana la partida} \end{cases}$$

En este caso (al igual que en el diseño de agentes que jueguen a otros juegos como las damas, el backgamon o el Go) es muy importante, cuando se define la señal de recompensa, tener claro el objetivo final y evitar recompensas intermedias que puedan desviarnos de dicho objetivo. Por ejemplo, si establecemos una recompensa por conseguir que los peones lleguen al final del tablero y se conviertan en reinas, el algoritmo puede centrarse en conseguir ese objetivo intermedio y dejar desprotegido al rey, con la consecuente pérdida de la partida, que era nuestro objetivo final.

Por otro lado, el marco de referencia que podemos establecer para todos los ejemplos anteriores es conocido como **toma de decisiones secuenciales** y consta de las siguientes características:

- El objetivo es seleccionar acciones que maximicen el retorno.
- Las acciones que tomemos pueden tener consecuencias a largo plazo.
- La recompensa puede llegar retrasada en el tiempo.
- Puede ser necesario sacrificar las recompensas inmediatas para obtener una mayor recompensa a largo plazo.

Por tanto, está claro que debemos planificar con antelación las consecuencias de nuestras acciones.

Algunos ejemplos ilustrativos de la presencia de estos elementos pueden ser la realización de inversiones en bolsa de un agente financiero (donde el beneficio económico puede verse demorado durante meses) o en el caso de un agente que juegue al ajedrez, la realización de movimientos para bloquear al oponente (que no reportan una recompensa inmediata pero pueden proporcionarnos más posibilidades de obtener la victoria final).

Por último hay que remarcar que debemos considerar la señal de recompensa como un elemento externo al agente. Aunque debemos diseñarla de forma conveniente junto con el agente para conseguir los objetivos deseados (tal y como hemos visto en los ejemplos anteriores), el agente no puede manipular su valor y es el entorno el que irá proporcionando nuevos valores conforme reciba las acciones del agente, tal y como veremos en el siguiente subapartado.

2.2. El entorno

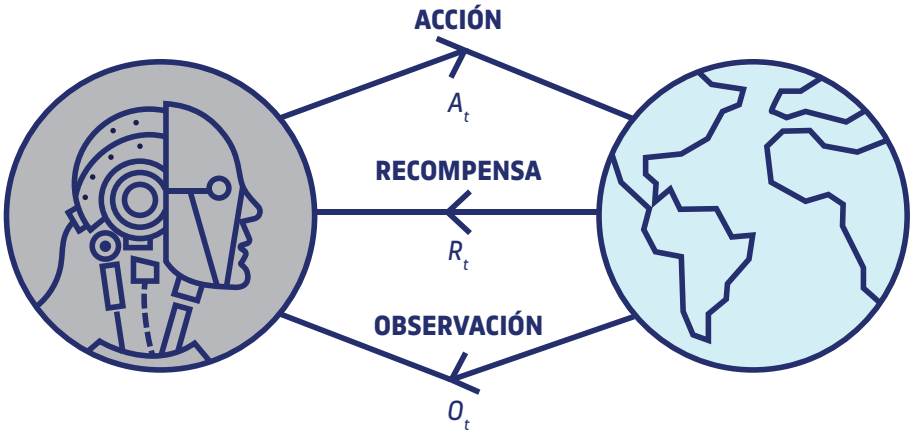
El siguiente elemento a tener en cuenta en el problema del aprendizaje automático es el entorno. Por definición el entorno incluye todo aquello que es

externo al agente. En el caso del juego de ajedrez es el tablero, las fichas y el jugador contrario. En el caso de un emulador de videojuegos es el propio emulador, el mando, la pantalla, etc. Por tanto, el entorno es todo aquello que no sea propiamente el ente que toma las decisiones (el agente).

Una vez tenemos esto en cuenta debemos analizar la interacción entre el agente y el entorno.

La figura 3 es una representación gráfica de dicha interacción. El autómata de la figura representa al agente y el mundo representa al entorno.

Figura 3. Interacción entre el agente y el entorno



Nuestro objetivo es diseñar algoritmos, dentro del agente, con el objetivo de tomar decisiones o, lo que es lo mismo, elegir acciones (qué valores escoger para comprar o vender en el caso de un inversor financiero, qué movimiento ejecutar en el caso de un jugador de ajedrez, etc.) sobre la base de la información que recibe el agente en cada instante de tiempo 't'.

En la cara opuesta se encuentra el entorno (representado por el mundo), que es todo aquello exterior al agente, pero que interacciona con él.

La comunicación bidireccional entre el agente y el entorno se lleva a cabo mediante tres señales. El agente solo puede influir en el entorno por medio de las acciones A_t elegidas en cada instante 't', y el entorno responde a estas acciones proporcionando una observación O_{t+1} y una señal de recompensa R_{t+1}. La señal de recompensa ya ha sido analizada en subapartados anteriores, no así la observación que proporciona el entorno. Dicha observación puede entenderse como una fotografía del entorno en ese instante (por ejemplo, en el caso de un agente que juega con un emulador de videojuegos, la observación sería la pantalla que proporciona el emulador después de cada acción).

A modo de resumen:

- En cada instante 't' el agente:
  - Recibe la observación O_t.
  - Recibe la recompensa R_t.
  - Ejecuta la acción A_t.
- Por su parte, el entorno:
  - Recibe la acción A_t.
  - Emite la observación O_{t+1}.
  - Emite la recompensa R_{t+1}.

Como podemos observar, la actualización del instante de tiempo se produce en el entorno (cada vez que genera una nueva observación y emite una nueva recompensa). Esta puntualización es importante para que podamos definir formalmente el problema en módulos posteriores.

Esta interacción, basada en la terna acción, observación y recompensa, se va repitiendo en un bucle (a veces infinito) generando una secuencia de datos que conforma la experiencia a partir de la cual el agente, mediante el procedimiento de prueba y error, irá aprendiendo.

2.3. Los estados

2.3.1. Historia y estado

Una definición más formal de la secuencia de datos que conforman la experiencia a partir de la cual el agente irá aprendiendo, tal y como hemos visto en el subapartado anterior, es lo que podríamos llamar historia H_t:

La **historia** es la secuencia de todas las acciones, observaciones y recompensas desde el inicio hasta el instante 't':

$$H_t = \{O_0,A_0,R_1,O_1,A_1,R_2,O_2,A_2,...,A_{t-1},R_t,O_t\}$$

Por tanto, la historia está conformada por todas las variables que pueden ser observadas, tanto por el agente como por el entorno, a lo largo del tiempo. De esta forma el agente elige las acciones basándose en la historia. De hecho, se puede entender un algoritmo de aprendizaje automático como un mapeo entre la historia y el conjunto de acciones a escoger. De la misma manera el entorno genera las nuevas observaciones y las recompensas basándose en la misma historia.

El problema que tiene la historia es que el volumen de variables que almacena se va incrementando con el tiempo, lo que hace que en la mayoría de los casos sea inmanejable y por tanto poco útil para los algoritmos de aprendizaje que tenemos que diseñar. Por ejemplo, en el caso de un agente que juega a un emulador de videojuegos la historia debería almacenar todos los movimientos ejecutados por el agente, así como todas las puntuaciones y las pantallas del videojuego generadas en cada instante durante el juego, por lo que el volumen de datos que habría que almacenar sería enorme.

Para solucionar este problema definimos los estados.

El **estado** S_t es la información que se usa para determinar qué sucederá en el próximo instante de tiempo. Formalmente es una función de la historia:

$$S_t = f(H_t)$$

Podemos ver el estado S_t como un resumen conciso de la historia H_t pero que debe incluir toda la información necesaria para determinar qué sucederá a continuación, tanto desde el punto de vista del agente (es necesario que el agente pueda tomar decisiones y ejecutar acciones según ese estado), como del entorno (el entorno debe generar las próximas observaciones y las próximas recompensas según el estado). De esta forma, dependiendo del problema que hay que resolver, el estado puede incluir, por ejemplo, la última observación, o las tres últimas, o cualquier función de ellas.

Existen varias definiciones y clases de estados. Además de la definición que ya hemos visto, especificaremos tres más:

1) estado del entorno (*environment state* en inglés),

2) estado del agente (*agent state* en inglés),

3) estado de información (*information state* en inglés).

2.3.2. Estado del entorno y del agente

El estado del entorno S_t^e es una representación privada del entorno. Está formado por todos los datos que utiliza el entorno para generar la próxima observación y recompensa. Normalmente esta información no es visible para el agente. Por ejemplo, en una partida de ajedrez incluye el tablero, las fichas, pero también lo que piensa el contrincante que juega contra el agente. En un

emulador de videojuegos incluye el historial de las instrucciones que se han ejecutado en el software del emulador. Los algoritmos que tenemos que diseñar no dependen de este estado, ya que la mayoría de las veces para el agente este estado está oculto. El algoritmo solo ve las observaciones y las recompensas que le llegan del entorno, pero no su estado. Además, incluso aunque el estado S_t^e sea observable por el agente, puede incluir mucha información irrelevante para dicho agente.

Por contra, el estado del agente S_t^a es la representación interna que hace el agente de la historia. Es cualquier información que usa el agente para elegir la próxima acción que debe realizar, y por tanto constituye la información que utilizaremos para construir los algoritmos de aprendizaje por refuerzo. El estado del agente S_t^a puede ser cualquier función de la historia ($S_t^a = f(H_t)$). Somos nosotros, como diseñadores del algoritmo de aprendizaje por refuerzo, quienes decidiremos cómo debe ser este estado S_t^a, qué observaciones y recompensas lo deben formar, cuáles descartar y cómo combinarlas para condensar toda la información necesaria que nos ayude a elegir las mejores acciones para conseguir nuestro objetivo.

2.3.3. Estado de información

Existe una definición más matemática del estado conocida como estado de información (o estado de Markov). Un estado de información es aquel que contiene toda la información útil de la historia H_t.

Propiedad de Markov

Definición: un estado S_t es denominado de Markov (o markoviano) si y solo si:

$$Pr\{S_{t+1}|S_t\} = Pr\{S_{t+1}|S_1,...,S_t\}$$

La definición anterior establece que la probabilidad del siguiente estado S_{t+1} condicionada por el estado actual es la misma que si está condicionada por el estado actual y todos los anteriores. De esta forma, si conocemos el estado actual, podemos desestimar toda la historia pasada para determinar el siguiente estado.

En otras palabras: «El futuro es independiente del pasado dado el presente». Es decir, toda la historia futura depende del estado actual S_t y no de los estados previos, que por tanto se pueden descartar.

Otra manera de verlo es que el estado S_t es un estadístico suficiente del futuro (se basta plenamente para caracterizar estadísticamente el futuro).

Sobre la base de esta definición se puede afirmar que tanto el estado del entorno S_t^e como la historia H_t cumplen la propiedad de Markov. No así el estado del agente S_t^a, que puede serlo o no serlo, depende de cómo lo diseñemos. Por tanto, ya que un estado de Markov tiene la información para caracterizar todas las recompensas futuras (que es la información que necesitamos para diseñar nuestros algoritmos) debemos intentar diseñar los estados del agente de forma acorde para que cumplan esta propiedad y no caer en representaciones imperfectas. Por ejemplo, en el caso de querer diseñar un algoritmo que conduzca un coche por una carretera, un estado que incluyera la posición del coche en la carretera, la dirección de las ruedas, la velocidad angular, la velocidad y fuerza del viento, etc., podría considerarse un estado de Markov, ya que determina plenamente el movimiento que hay que hacer con el volante en el próximo instante y no depende de estados anteriores (no nos sirve de nada conocer todas esas variables de hace unos minutos). Por contra, un estado que solo incluyera la posición del coche no podría considerarse de Markov, ya que deberíamos conocer estados anteriores para determinar la trayectoria y decidir qué movimiento realizar con el volante. Sería por tanto un estado que representa la historia parcialmente, de forma imperfecta.

Nuestro objetivo es evitar situaciones como la anterior y diseñar estados del agente S_t^a que nos permitan predecir qué sucederá a continuación con la mayor fiabilidad posible.

2.3.4. Entornos plenamente observables y parcialmente observables

Para acabar con este subapartado vamos a comentar dos casos particulares:

- **Entornos plenamente observables** (*fully observable environments* en inglés). Es el caso ideal en que el agente puede observar el estado del entorno y utilizarlo para la elección de la acción que debe realizar. En este caso la observación, el estado del entorno y el estado del agente coinciden ($O_t = S_t^e = S_t^a$). Cuando trabajamos con este tipo de entornos, podemos definir uno de los pilares sobre los que se describen formalmente los problemas de aprendizaje por refuerzo: los procesos de decisión de Markov (abreviados como MDP, *Markov decision processes* en inglés).

- **Entornos parcialmente observables** (*partially observable environments* en inglés). El agente observa el entorno de forma indirecta, no tiene una visión completa del entorno y de su estado. Por ejemplo, un jugador de dominó no puede ver las fichas de su oponente. Un inversor puede ver los precios de los activos, pero no de dónde provienen esos precios. En este ca-

so el estado del agente no coincide con el estado del entorno ($S_t^e \neq S_t^a$). Formalmente, este tipo de problemas se caracterizan matemáticamente por medio de los procesos de Markov parcialmente observables (abreviado POMDP, *partially observable Markov decision processes* en inglés).

En este caso es el agente el que debe construir una representación del estado S_t^a adecuada. Para ello existen varias posibilidades entre las que remarcamos:

– Utilizar como estado la historia completa, es decir $S_t^a = H_t$. El problema de esta solución, como ya hemos comentado anteriormente, es que la mayoría de las veces es inviable debido a la extensión de la historia.

– Realizar una aproximación estadística, también denominada bayesiana, al proceso de generación del estado y aplicar nuestras propias creencias sobre lo que ocurre en el entorno definiendo ciertas probabilidades que caractericen las diferentes posibilidades de que el entorno se encuentre en un estado concreto o en otro. Es decir $S_t^a = (Pr\{S_t^e = s_1\},...,P\{S_t^e = s_n\})$.

– Aplicar una red neuronal recurrente (abreviada RNN, *recurrent neural network* en inglés), que nos permita combinar linealmente tanto las acciones pasadas del agente como las observaciones y aplicarles finalmente una no linealidad. $S_t^a = \sigma(S_{t-1}^a W_s + O_t W_o)$

3. Los agentes en el aprendizaje por refuerzo

Hasta ahora, en el apartado anterior, lo que hemos hecho es definir el problema, pero no la manera de solucionarlo. Esa es la función del agente.

En este apartado analizaremos con más detenimiento la composición de un agente, definiendo los componentes principales que lo caracterizan, así como una clasificación de los tipos de agente en función de la presencia o ausencia de estos componentes.

3.1. Principales componentes de un agente

Un agente de aprendizaje por refuerzo puede disponer de uno o más de los siguientes elementos:

- **Política** (*policy* en inglés), que rige la manera como el agente selecciona las acciones.
- **Función de valor** (*value function* en inglés), que es una función que nos indica cuán bueno (en términos del posible retorno) es estar en un determinado estado o llevar a cabo una determinada acción.
- **Modelo** (*model* en inglés), que es la visión del entorno que tiene el agente, cómo piensa el agente que el entorno evoluciona.

3.1.1. Política

La política $\pi(\bullet)$, es una función que rige la manera en que el agente selecciona las acciones. Es una función que hace un mapeo entre los estados S_t y las acciones A_t. Explica por tanto el comportamiento del agente.

Existen dos tipos de políticas:

- **Determinista.** En este caso la acción a tomar 'a' viene determinada unívocamente por el estado 's' en que se encuentra el agente. Es por tanto un mapeo directo entre el estado y la acción:

$$a = \pi(s) \tag{3}$$

- **Estocástica.** En este caso puede haber más de una acción a escoger para cada estado, y lo que hace la política es asignar una probabilidad a cada una de esas opciones:

$$\pi(a|s) = Pr\{A_t = a|S_t = s\} \quad (4)$$

donde $\pi(a|s)$ es la probabilidad de seleccionar la acción 'a' si el agente se encuentra en el estado 's'. Este suele ser el caso habitual en la mayoría de los problemas de la vida real. Una política estocástica permite la exploración del entorno, ya que propicia que, si pasamos dos veces por el mismo estado, el agente puede tomar acciones distintas.

3.1.2. Función de valor

Como hemos comentado con anterioridad, la función de valor $v_\pi(s)$ es una función que nos indica cuán bueno (en términos del posible retorno) es estar en un determinado estado.

$$v_\pi(s) = \mathbb{E}_\pi[G_t|S_t = s] = \mathbb{E}_\pi[R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + ...|S_t = s] \quad (5)$$

Esta función calcula el valor esperado de todas las posibles recompensas acumuladas futuras (el retorno futuro) si el agente se encuentra en el estado 's' y sigue una política $\pi(\bullet)$ a partir de este instante. Es, por tanto, una predicción de las recompensas futuras.

Es importante darse cuenta de la dependencia de la función de valor $v_\pi(s)$ con la política $\pi(\bullet)$, ya que, dependiendo de qué política siga el agente, se tomarán unas acciones u otras que nos llevarán a distintos futuros estados y recompensas.

Por otro lado, es interesante notar que esta función de valor también puede ser utilizada para seleccionar acciones, ya que podríamos realizar la elección de dichas acciones basándose en el valor de los estados futuros. Es decir, si conseguimos calcular la función de valor para todos los posibles estados a los que puede llegar el agente en el siguiente instante de tiempo, tendremos una clasificación de cuán deseables son estos futuros estados, y podemos seleccionar la acción que nos lleve al estado de mayor valor.

Además de la función de valor de un estado $v_\pi(s)$, podemos definir la función de valor de una acción (o un par estado-acción) $q_\pi(s,a)$:

$$q_\pi(s,a) = \mathbb{E}_\pi[G_t|S_t = s,A_t = a] = \mathbb{E}_\pi[R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + ...|S_t = s,A_t = a] \quad (6)$$

Esta ecuación calcula el valor esperado de todas las posibles recompensas acumuladas futuras (el retorno futuro) si el agente se encuentra en el estado *'s'*, toma la acción *'a'* en ese instante y sigue una política $\pi(\bullet)$ a partir del instante siguiente. Como esta función relaciona la predicción de las recompensas futuras con la acción inmediata que selecciona el agente, una posible política consistiría en seleccionar aquella acción que maximice dicha función. Esta es la base de algunos de los algoritmos más relevantes del aprendizaje por refuerzo.

3.1.3. Modelo

Un modelo es una visión del entorno que tiene el agente, cómo piensa el agente que el entorno evoluciona, cómo funciona. No es el entorno en sí, sino que pretende ser una predicción de su comportamiento. Cuando un agente dispone de un modelo, puede predecir lo que el entorno hará en el siguiente instante de tiempo y actuar en consecuencia seleccionando las acciones oportunas.

Formalmente existen dos tipos de modelos o predicciones:

- Modelo de transiciones: $\mathcal{P}$. Es una predicción del próximo estado que generará el entorno, de su dinámica. Por ejemplo:

$$\mathcal{P}^a_{ss'} = p(s'|s,a) \approx Pr\{S_{t+1} = s' \mid S_t = s, A_t = a\} \tag{7}$$

- Modelo de recompensas: $\mathcal{R}$. Es una predicción del valor esperado de la próxima recompensa inmediata que generará el entorno.

$$\mathcal{R}^a_s = r(s,a) \approx \mathbb{E}[R_{t+1} \mid S_t = s, A_t = a] \tag{8}$$

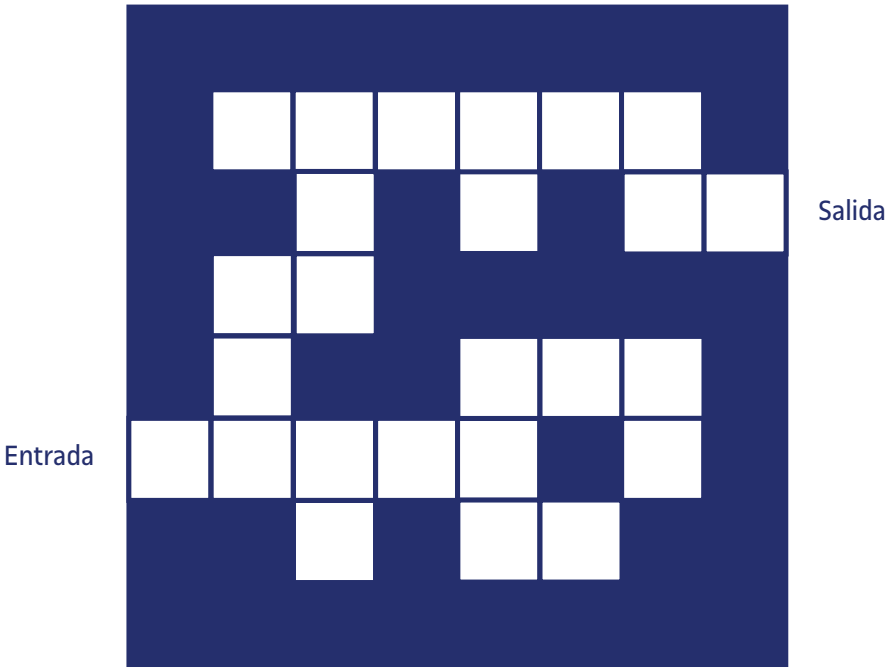
Es importante remarcar que un buen modelo no nos proporciona una buena política de por sí, pero nos facilita que podamos desarrollarla. Para ello, a partir del modelo debemos realizar la predicción del comportamiento futuro del entorno para incorporarlo en nuestra política; es lo que se conoce como planificación (*planning* en inglés).

Por último, cabe mencionar que un agente puede tener un modelo o no, es algo opcional. A lo largo del curso la mayoría de los agentes que veremos carecen de modelo, pero es importante tener en mente que esa posibilidad existe.

3.2. Ejemplo: el laberinto

En este subapartado analizaremos un ejemplo que nos servirá para clarificar los conceptos explicados a lo largo de este apartado. El problema consiste en diseñar un agente que sea capaz de aprender cómo cruzar el laberinto de la figura 4 de la manera más rápida posible.

Figura 4. Laberinto



Con el propósito de que el agente aprenda a cruzar el laberinto con la máxima celeridad posible, se establecen las siguientes recompensas (tal y como ya vimos con anterioridad):

$$\begin{cases} R_t = -1 & \text{para cualquier movimiento, excepto si alcanza la salida} \\ R_t = 0 & \text{si alcanza la salida} \end{cases}$$

Las posibles acciones que se pueden seleccionar consisten en realizar un movimiento en la dirección de los cuatro puntos cardinales:

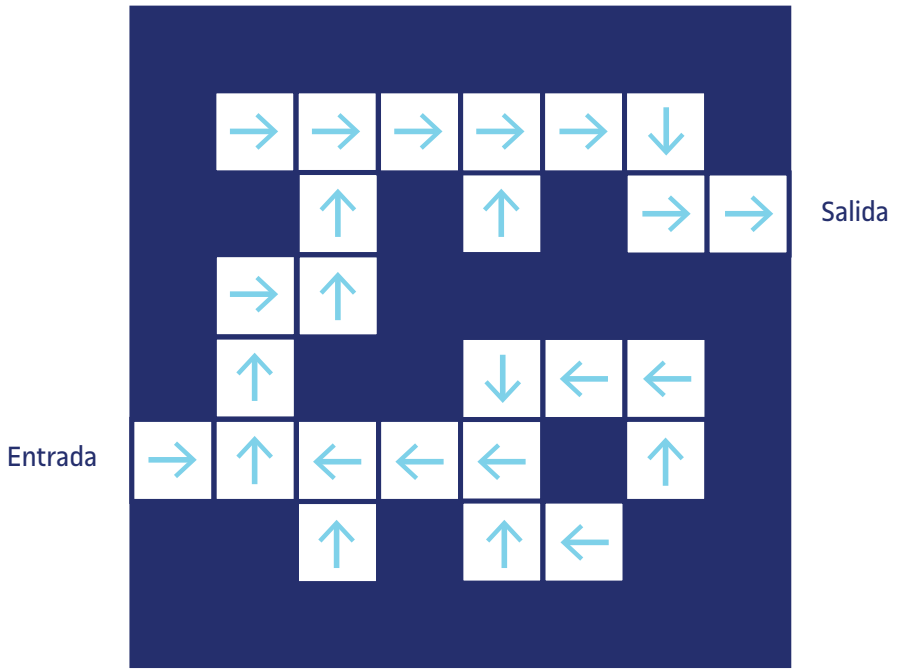
$$A_t \in \mathcal{A} = \{N = norte, S = sur, E = este, O = oeste\}$$

Por último, los estados S_t incluyen tan solo la posición del agente dentro del laberinto (las casillas del laberinto).

3.2.1. Política

Dado que solo existe un camino que permita atravesar el laberinto en el menor tiempo posible, la política a utilizar es determinista ($a = \pi(s)$). Podemos observar dicha política en la figura 5, donde las flechas representan la política ($a = \pi(s)$) para cada estado 's'.

Figura 5. Laberinto: representación de la política determinista $a = \pi(s)$



Se trata de un mapeo entre los estados y las acciones, ya que nos indica qué acción 'a' debemos tomar en cada estado 's'.

Remarcar que si hubiera más de un camino posible que nos permitiera llegar a la salida en el menor tiempo posible (por ejemplo, por medio de una bifurcación) habría que establecer una política estocástica $\pi(a|s)$.

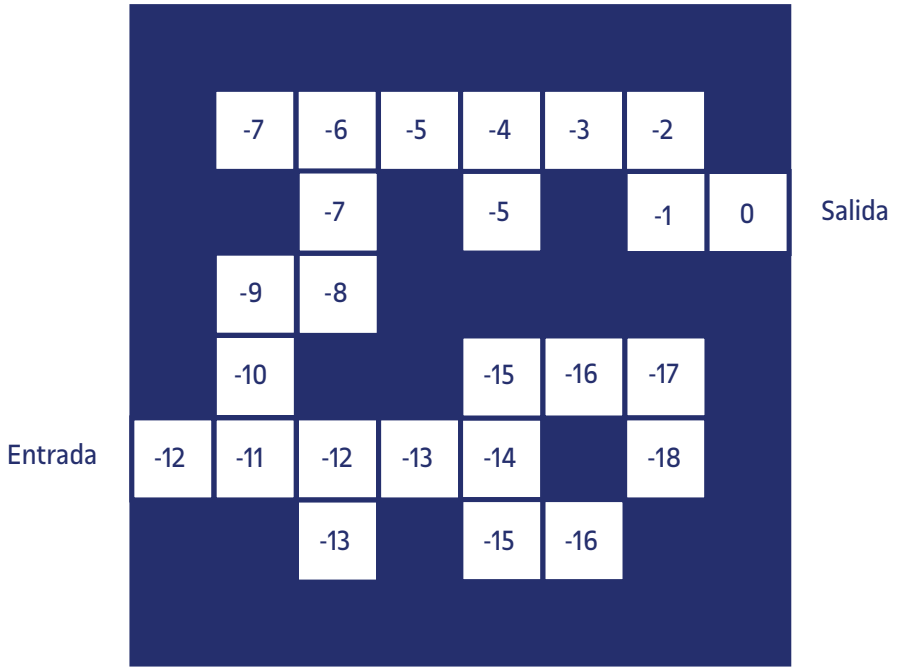
3.2.2. Función de valor

La figura 6 representa los valores de la función de valor de estado para el ejemplo del laberinto. Así, en cada casilla nos encontramos con un número que representa el valor esperado del retorno $v_\pi(s)$ en ese estado 's' siguiendo la política determinista explicada en el subapartado anterior. Los valores se han calculado con un factor de descuento $\gamma = 1$.

Si empezamos por el último estado (la casilla que se encuentra antes de la salida) vemos que su valor es 0, ya que si ejecutamos la acción determinada por la política establecida (moverse en dirección este en este caso) llegaremos a la salida y la recompensa será cero. Si miramos el estado inmediatamente

a la izquierda (hacia el oeste) vemos que su valor es de menos uno, ya que si ejecutamos la política mencionada, primero nos moveremos hacia el este (llegando al estado anterior) y luego volveremos a ejecutar la acción Este para salir del laberinto, con lo que la función de valor será el retorno que obtendremos siguiendo esta política, concretamente serán las recompensas obtenidas en estas dos acciones $R_{t+1} + R_{t+2} = -1 + 0 = -1$. Siguiendo este razonamiento es fácil calcular la función de valor para todos los estados.

Figura 6. Laberinto: función de valor $v_\pi(s)$



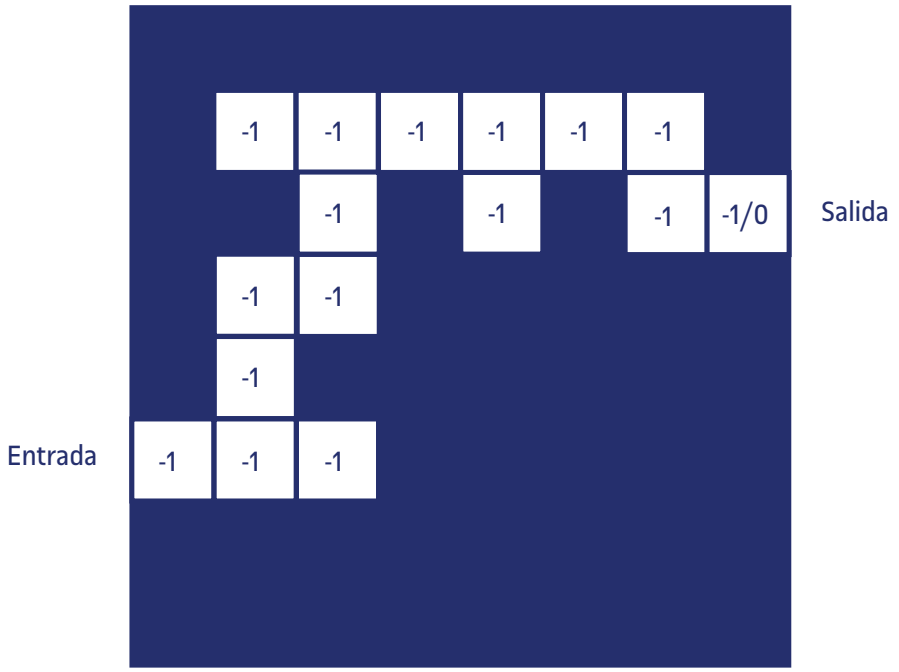
Merece la pena observar que con estos valores podríamos construir una política óptima basada en moverse en la dirección que nos lleva a un estado de mayor valor y que dicha política coincide con la política determinista del subapartado anterior.

3.2.3. Modelo

En la figura 7 tenemos la representación de un modelo del laberinto. Imaginemos que el agente ha observado una trayectoria a través del laberinto (el camino que se ve en la figura), ha alcanzado la salida y pretende construir un modelo de cómo funciona el entorno, cómo es la dinámica del entorno.

De esta forma la cuadrícula en sí misma representa parcialmente el modelo de transiciones $\mathcal{P}^a_{ss'}$, mientras que los números que aparecen en cada casilla representan el modelo de recompensa inmediata $\mathcal{R}^a_s$ para cada estado 's'. Si nos fijamos, vemos que es el mismo para cada acción 'a' y cada estado futuro 's'', exceptuando en la casilla final, donde la acción Oeste provocaría una recompensa inmediata de menos uno y la acción Este provocaría una recompensa de valor cero.

Figura 7. Laberinto: modelo del entorno



3.3. Taxonomía de los tipos de agentes

Existen principalmente dos formas de clasificar los tipos de agentes según los elementos clave que contienen (política, función de valor y modelo).

3.3.1. Basado en la función de valor, en la política o ambas

La primera clasificación hace referencia al hecho de si el agente utiliza la política, la función de valor o ambas para determinar las acciones que se deben elegir en cada momento. De esta forma podemos definir los siguientes tres tipos de agentes:

- **Agentes basados en la función de valor**. Son aquellos que contienen de forma explícita una función de valor mientras que la política está de forma implícita. En el ejemplo del laberinto que hemos visto en el subapartado anterior, un agente basado en la función de valor solo almacenaría los datos de la figura 6, y seleccionaría la acción que le llevara al estado siguiente de más valor. No necesita por tanto una representación explícita de la política a seguir, aunque realmente estaría ejecutando una política (óptima en este caso) pero no estaría almacenada en el agente de forma explícita, tan solo estaría almacenada la función de valor.

- **Agentes basados en la política**. Son aquellos que contienen de forma explícita una representación de la política a seguir, pero no de la función de valor (si está, está de forma implícita). En el ejemplo del laberinto, un agen-

te basado en la política tendría almacenadas las acciones de cada estado tal y como aparecen en la figura 5. De esta forma elegiría las acciones en cada estado sobre la base de esta política almacenada (las flechas de la figura 5) y conseguiría el máximo retorno al seguirlas, pero sin tener almacenados los valores de la función de valor para cada estado.

- **Actor crítico**. Es una combinación de los dos tipos de agentes anteriores. Un actor crítico es un tipo de agente que almacena explícitamente tanto la función de valor como la política, y usa ambas (las combina) para determinar la acción que hay que escoger en cada situación.

En la tabla 1 se muestra un resumen de las principales características de estos tres tipos de agentes.

Tabla 1. Clasificación según la política y/o la función de valor

Descripción	Función de valor	Política
Basado en la función de valor	Tiene función de valor explícita	Sin política explícita (solo implícita)
Basado en la política	Sin función de valor explícita (solo implícita)	Tiene política explícita
Actor crítico	Tiene función de valor explícita	Tiene política explícita

3.3.2. Basado en un modelo o libre de modelo

Otra de las principales distinciones en el aprendizaje por refuerzo es si los agentes disponen o no de un modelo del entorno. Basándonos en esto podemos clasificar a un agente como:

- **Libre de modelo** (*model free* en inglés). En este caso el agente no intenta entender explícitamente cómo es el entorno, cómo son sus dinámicas, y tampoco pretende aprender una representación del entorno (un modelo). Su aprendizaje se basa, única y exclusivamente, en una política y/o en una función de valor.

- **Basado en un modelo** (*model based* en inglés). En este caso lo primero que hace el agente es intentar aprender un modelo del entorno y después hacer predicciones según este modelo. En este caso también puede existir una política y/o una función de valor, pero lo primordial es la existencia del modelo.

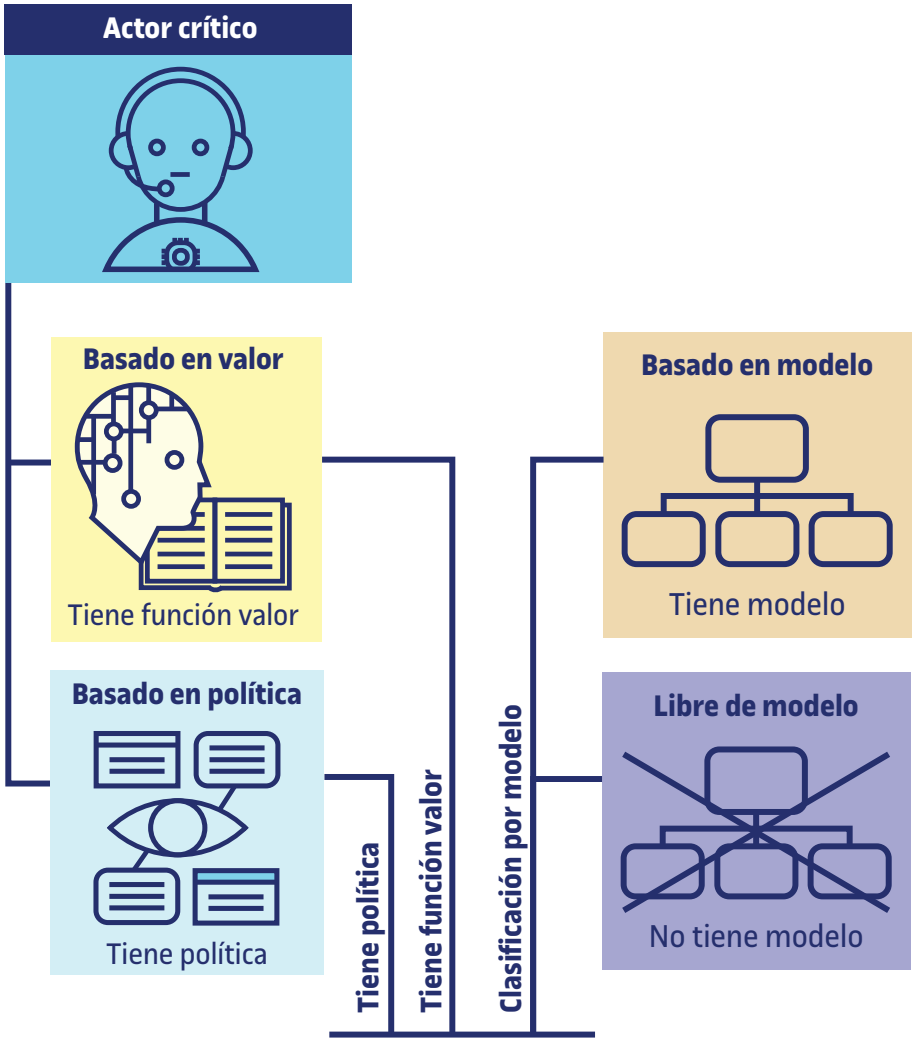
En la tabla 2 se muestra un resumen de las principales características de estos dos tipos de agentes.

Tabla 2. Clasificación según el modelo

Descripción	Modelo	Función de valor y/o política
Libre de modelo	No tiene un modelo explícito	Puede tener política y/o función de valor.
Basado en un modelo	Tiene un modelo explícito	Puede tener política y/o función de valor.

Una versión gráfica de la taxonomía de los diferentes tipos de agentes puede verse en la figura 8.

Figura 8. Taxonomía de los agentes de aprendizaje automático



4. Problemas principales en el aprendizaje por refuerzo

En este apartado final veremos algunos problemas clave que nos encontraremos al diseñar un agente de aprendizaje por refuerzo:

- Aprendizaje y planificación
- Exploración frente a explotación
- Predicción y control

No presentaremos aún las soluciones a estos problemas (esto lo iremos haciendo a lo largo del curso), tan solo los introduciremos para tenerlos presentes cuando llegue el momento de afrontarlos.

4.1. Aprendizaje y planificación

Existen fundamentalmente dos configuraciones de problemas en la toma secuencial de decisiones:

- **Aprendizaje por refuerzo** puramente dicho, donde el entorno es inicialmente desconocido, el agente interactúa con el entorno y mediante el proceso de prueba y error consigue ir mejorando su política según la experiencia que recoge de esa interacción. Esta es la configuración básica de la mayoría de los problemas de aprendizaje por refuerzo.

- **Planificación** (*planning* en inglés). En este tipo de problemas el agente tiene un modelo perfecto del entorno que le viene dado de base (no tiene que aprenderlo). Entonces el agente realiza operaciones sobre este modelo (sin ningún otro tipo de interacción) y basándose en los resultados de esos cálculos va mejorando su política.

Ambos tipos de problemas son distintos aunque están ligados, ya que podemos partir de un problema de aprendizaje por refuerzo puro, cuyo objetivo sea aprender un modelo, y luego realizar planificación a partir de este modelo aprendido.

4.2. Exploración y explotación

Como ya hemos visto con anterioridad, el aprendizaje por refuerzo es un problema de aprendizaje mediante prueba y error en que el agente debe descubrir

una buena política a partir de la experiencia que recoge de su interacción con el entorno. El proceso mediante el cual el agente busca nuevos caminos que le permitan aprender nuevas soluciones se conoce como proceso de exploración (*exploration* en inglés). El problema de la exploración es que normalmente se producen pérdidas (en lo que a recompensas se refiere) hasta que no se encuentra una solución mejor.

Por contra, el proceso que trata de rentabilizar la experiencia adquirida maximizando el retorno se conoce como proceso de explotación (*explotation* en inglés).

Existe por tanto un compromiso entre la parte del tiempo que el algoritmo dedica a explorar nuevas acciones y el tiempo que dedica a explotarlas. Es de suma importancia saber balancear ambos procesos en el diseño de un agente, ya que ambos suelen ser igual de importantes a la hora de resolver un problema de aprendizaje por refuerzo.

De esta forma, si por ejemplo queremos diseñar un agente para recomendar canciones a un usuario, la fase de explotación consistiría en recomendar al usuario canciones de su cantante o estilo favoritos, mientras que la fase de exploración consistiría en probar nuevos artistas y estilos. En el caso de un agente que juegue al ajedrez, la fase de explotación sería realizar el movimiento que tenga un mejor retorno sobre la base de la experiencia acumulada en ese momento, y la fase de exploración consistiría en realizar un movimiento experimental. En el caso del laberinto, si hubiese más de un camino que nos llevara a la salida y ya hemos descubierto uno de ellos (aunque tal vez no sea el más corto), la fase de explotación sería seguir siempre ese camino, y la fase de exploración sería probar otros caminos para ver si encontramos un camino más corto.

4.3. Predicción y control

Por último vamos a introducir una distinción entre dos términos que también encontraremos en los problemas de aprendizaje por refuerzo: predicción y control.

Hacer una predicción es fundamentalmente evaluar el futuro a partir de una política dada. Por ejemplo, la función de valor $v_\pi(s)$ es una predicción del retorno, y un modelo es una predicción de las dinámicas del entorno.

Por otro lado, control significa optimizar el futuro. Por ejemplo, encontrar la política óptima $\pi_*(s)$ que nos permita obtener el máximo valor.

Por tanto, ambos conceptos están relacionados. Matemáticamente esta relación se puede expresar de la siguiente manera:

$$\pi_*(s) = \underset{\pi}{argmax} v_\pi(s) \tag{9}$$

La ecuación anterior nos dice que la política óptima es aquella política que consigue obtener una función de valor máxima. Lo que suele pasar en los problemas de aprendizaje por refuerzo es que muchas veces tenemos que resolver el problema de predicción (estimar el valor de $v_\pi(s)$) para resolver a continuación el problema de control, ya que debemos evaluar diferentes políticas para encontrar el valor de la política óptima $\pi_*(s)$.

Resumen

A lo largo de este módulo hemos realizado una introducción al aprendizaje por refuerzo.

Hemos iniciado el módulo dando una visión histórica sobre la motivación que ha llevado al desarrollo del aprendizaje automático y del aprendizaje por refuerzo. A continuación, hemos visto cómo partes de los fundamentos y características del aprendizaje por refuerzo aparecen en diversas disciplinas de muchas áreas de la ciencia. Hemos seguido analizando cuáles son las características básicas que lo caracterizan y cómo encaja entre las diferentes ramas del aprendizaje automático.

Posteriormente nos hemos centrado en definir los elementos que forman cualquier problema de aprendizaje por refuerzo. De esta forma hemos estudiado la señal de recompensa, el entorno y hemos definido el concepto de estado, conceptos primordiales todos ellos a la hora de diseñar un agente de aprendizaje por refuerzo.

A continuación, hemos definido los elementos que puede contener un agente: la política, la función de valor y el modelo, y hemos hecho hincapié en la taxonomía de los agentes dependiendo de si disponen de uno o varios de estos elementos.

Finalmente hemos visto una descripción de algunos problemas clave que aparecen al diseñar un agente de aprendizaje por refuerzo y que iremos resolviendo a lo largo del curso, tales como la diferencia entre aprendizaje y planificación, la importancia del balance entre las fases de exploración y explotación en el diseño de cualquier agente, así como la diferencia entre los conceptos de predicción y control y cómo se relacionan entre sí.

Glosario

aprendizaje automático *m* disciplina de las ciencias de computación que da a los ordenadores la capacidad de aprender sin ser explícitamente programados.
sigla **ML**
en machine learning

aprendizaje profundo *m* conjunto de algoritmos de aprendizaje automático que intenta modelar abstracciones de alto nivel utilizando redes neuronales de múltiples capas o niveles.
sigla **DL**
en deep learning

deep learning *m* Véase **aprendizaje profundo**.

DL *m* Véase **aprendizaje profundo**.

i.i.d *m* Véase **independientes e idénticamente distribuidos**.

independent and identically distributed *m* Véase **independientes e idénticamente distribuidos**.

independientes e idénticamente distribuidos *m* conjunto de datos o variables aleatorias que son mutuamente independientes entre sí y además cada variable tiene la misma distribución de probabilidad.
sigla **i.i.d.**
en independent and identically distributed

machine learning *m* Véase **aprendizaje automático**.

ML *m* Véase **aprendizaje automático**.

taxonomía *f* clasificación u ordenación en grupos de cosas que tienen características comunes.

Bibliografía

Lapan, Maxim (2020). *Deep Reinforcement Learning Hands-On (Second Edition)*. Packt Publishing. ISBN: 9781838826994

Silver, David (2015). *UCL Course on RL*. [Fecha de consulta: 12 de abril de 2021]. <https://www.davidsilver.uk/teaching/>

Sutton, Richard S.; Barto, Andrew G. (2018). *Reinforcement Learning: An Introduction (Second Edition)*. Cambridge, MA: MIT Press.

Van Hasselt, Hado (2018). *Deepmind Videolecture: Introduction to Reinforcement Learning*. [Fecha de consulta: 12 de abril de 2021]. <https://youtu.be/ISk80iLhdfU>