Aprendizaje Automático No Supervisado

# Tema 9. Introducción al aprendizaje por refuerzo

# Índice

# Esquema

## 9.1. Introducción y objetivos

E l **aprendizaje por refuerzo** (*reinforcement learning,* RL) es una rama del aprendizaje automático que se centra en cómo los agentes deben tomar decisiones para maximizar alguna noción de recompensa acumulada a lo largo del tiempo. A diferencia de otros enfoques de aprendizaje, el RL permite a los agentes aprender a través de la interacción directa con su entorno, ajustando sus estrategias en función de las consecuencias de sus acciones (Kaelbling, Littman y Moore, 1996; Szepesvári, 2022).

En este contexto, los agentes se definen como programas de *software* capaces de operar de manera autónoma, percibiendo su entorno mediante sensores y actuando sobre él mediante actuadores. Estos agentes son fundamentales en RL, ya que son los responsables de experimentar, aprender y mejorar continuamente sus decisiones para maximizar las recompensas.

El concepto de agente ha evolucionado desde una entidad aislada que opera de forma independiente a un componente integral de sistemas más amplios, donde múltiples agentes autónomos interactúan para formar un sistema global. Esta evolución ha llevado a que el diseño de agentes sea visto tanto desde la perspectiva de la ingeniería de *software* como de la inteligencia artificial.

Los **objetivos** de este tema son:

- **Definir y comprender los agentes inteligentes:** entender qué son los agentes inteligentes, sus características principales y cómo interactúan con su entorno para tomar decisiones autónomas.
- **Explorar los componentes de los procesos de decisión de Markov (MDP):** analizar los componentes esenciales de los MDP, tales como estados, acciones, recompensas y transiciones, y cómo estos proporcionan una estructura matemática

para modelar problemas de decisión en RL.

- **Aplicar las ecuaciones de Bellman:** comprender y aplicar las ecuaciones de Bellman para descomponer problemas de optimización a largo plazo en subproblemas más manejables, facilitando la solución de MDPs y el desarrollo de políticas óptimas.

- **Explorar aplicaciones prácticas de RL:** examinar diversas aplicaciones del aprendizaje por refuerzo en áreas como conducción autónoma, automatización industrial, comercio y finanzas, procesamiento del lenguaje natural, atención sanitaria e ingeniería, demostrando su versatilidad y potencial en la solución de problemas complejos en el mundo real.

## 9.2. Definición de agente

Los **agentes** son fundamentales en el RL, ya que son los encargados de interactuar con el entorno, aprender de las consecuencias de sus acciones y mejorar su comportamiento para maximizar las recompensas a lo largo del tiempo.

Un agente es un tipo de programa *software* cuya función es observar el entorno y reaccionar ante él, operando de forma autónoma. El proceso parte de la respuesta a partir de unos datos de entrada, de tal manera que se puede interpretar lo que se observa, razonar una respuesta, y luego aplicarla. De esta manera, es posible considerar lo siguiente:

> «Los agentes representan el nuevo paradigma más importante para el desarrollo de *software* desde la orientación a objetos» (McBurney, 2004).

El concepto de agente ha pasado de ser un ente aislado que opera de forma autónoma a un elemento más de un sistema en el que diferentes sistemas autónomos conforman un sistema global. El concepto de agente desde la perspectiva de la ingeniería de *software* se plantea más como un elemento de diseño de *software* que como un elemento propio de la inteligencia artificial.

El agente percibe el entorno mediante sensores (reales o virtuales) y actúa sobre él mediante actuadores (de nuevo, reales o virtuales) (ver Figura 1).

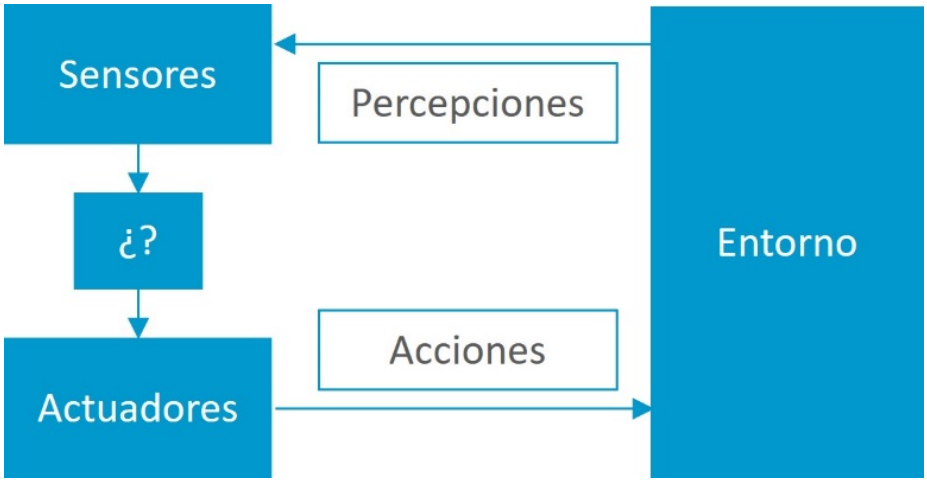



Figura 1. Componentes que conforman un agente. Fuente: elaboración propia.

Las **características** de un agente inteligente son:

- **Autonomía:** capacidad para operar sin intervención humana directa y controlar sus acciones y estado internos.
- **Percepción y acción:** utiliza sensores para obtener información del entorno y actuadores para interactuar con él.
- **Objetivo o función de recompensa:** posee un criterio claro de éxito o recompensa que guía sus decisiones y acciones.
- **Adaptabilidad:** capacidad de modificar su comportamiento basado en experiencias pasadas o nueva información recibida del entorno.
- **Persistencia:** opera de manera continua a lo largo del tiempo.

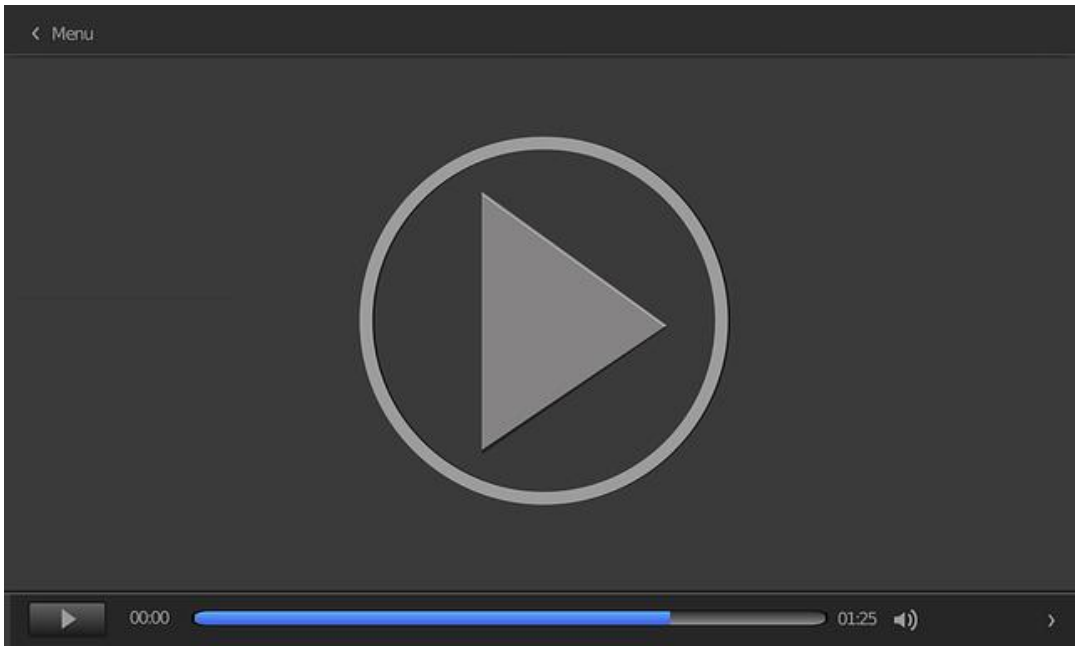
Existen diferentes **tipos** de agentes inteligentes:

- **Agente reactivo simple:** toma decisiones basadas únicamente en la percepción actual del entorno. Ejemplo: un termostato.
- **Agente basado en modelo:** mantiene un modelo interno del entorno para tomar

decisiones más informadas. Ejemplo: un robot de limpieza que recuerda los obstáculos.

- **Agente basado en objetivos:** actúa para cumplir ciertos objetivos específicos, no solo para reaccionar. Ejemplo: un agente de búsqueda en un juego.
- **Agente basado en utilidad:** toma decisiones basadas en la maximización de una función de utilidad que puede incorporar múltiples objetivos y prioridades. Ejemplo: sistemas de recomendación personalizados.

A continuación, veremos el vídeo ***Implementación de aprendizaje por refuerzo a través de un ejemplo.***




Accede al vídeo:

https://unir.cloud.panopto.eu/Panopto/Pages/Embed.aspx?id=4937a871-3bf9-43b2-af16-b1c500f89a95

## 9.3. Procesos de decisión de Markov (MDP)

Un MDP es un marco matemático utilizado para modelar la toma de decisiones en situaciones donde los resultados son en parte aleatorios y en parte bajo el control de un agente (Puterman, 1990; Wei, Xu, Lan, Guo y Cheng, 2017).

Los MDPs se componen de estados, acciones, recompensas y transiciones de estado.

Son esenciales en el RL porque proporcionan la estructura matemática para modelar y resolver problemas en los que un agente debe aprender a tomar decisiones óptimas a través de la interacción con su entorno.

### Componentes

- **Estados (S):** representan todas las posibles situaciones en las que puede encontrarse el agente.
- **Acciones (A):** conjunto de todas las acciones posibles que el agente puede realizar.
- **Transiciones (T):** probabilidades de pasar de un estado a otro dado una acción específica. Formalmente, $T(s,a,s') = P(s' \vee s,a)$ , donde $s$ es el estado actual, $a$ es la acción tomada y $s'$ es el estado siguiente.
- **Recompensas (R):** recompensa recibida después de transitar de un estado a otro, dado una acción. Formalmente, $R(s,a,s')$ .
- **Política (π):** estrategia que el agente sigue para decidir qué acción tomar en cada estado.

El paso a paso en los procesos de decisión de Markov se muestra en la Figura 2.


Figura 2. Proceso de solución en MDP. Fuente: elaboración propia.

## Política óptima y política estacionaria

### Política óptima

En el contexto de la teoría de decisiones y el aprendizaje por refuerzo, una política (denotada usualmente como π) define una estrategia para elegir acciones en cada estado con el fin de maximizar alguna medida de rendimiento a largo plazo, generalmente la utilidad.

La política óptima, π*, es aquella que maximiza el rendimiento esperado desde cualquier estado inicial. Es decir, es la mejor estrategia posible que un agente puede seguir para obtener el mayor beneficio esperado.

### Política estacionaria

Una política es estacionaria si las decisiones que se toman no dependen del tiempo, sino únicamente del estado actual. Esto significa que, independientemente de cuándo se encuentra el agente en un determinado estado, la acción que se tomará será siempre la misma.

Las políticas estacionarias son importantes porque simplifican el análisis y la implementación de algoritmos en el aprendizaje por refuerzo.

## Utilidad

La utilidad es una medida del valor o satisfacción que un agente obtiene de un

determinado resultado. Juega un papel crucial en la definición y evaluación de políticas, especialmente en contextos de decisión secuencial.

**Utilidad aditiva**

La utilidad aditiva asume que el valor total de una secuencia de decisiones es simplemente la suma de las utilidades de cada decisión individual.

Esta suposición simplifica mucho los cálculos y es adecuada en situaciones donde los efectos de las decisiones no se interrelacionan de manera compleja. La fórmula de la utilidad aditiva está expresada en la siguiente ecuación.

$$U\left[\left(S_0,S_1,S_2,S_3,\ldots\right)\right]=r_0+r_1+r_2+r_3+\ldots=\sum_{t=0}^{\infty}r_t$$

**Utilidad ponderada**

En la utilidad ponderada, cada utilidad individual se multiplica por un factor de ponderación antes de ser sumada.

Esta técnica permite dar más importancia a ciertas decisiones o resultados sobre otros, reflejando prioridades o preferencias específicas. La fórmula de la utilidad ponderada está expresada en la siguiente ecuación.

$$U\left[\left(S_0,S_1,S_2,S_3,\ldots\right)\right]=p_o r_0+p_1 r_1+p_2 r_2+p_3 r_3+\ldots=\sum_{t=0}^{\infty}p_t r_t$$

**Utilidad descontada**

La utilidad descontada introduce la idea de que las recompensas futuras valen menos que las recompensas inmediatas. Esto se representa con un factor de descuento γ ($0 \leq \gamma < 1$). La fórmula de la utilidad ponderada está expresada en la siguiente ecuación.

# Ideas clave

$$U\left[\left(S_0,S_1,S_2,S_3,\ldots\right)\right]=r_0+\gamma r_1+\gamma^2 r_2+\gamma^3 r_3+\ldots=\sum_{t=0}^{\infty}\gamma^t r_t:\gamma\in[0,1]$$

La **utilidad total** en un proceso de decisión es la suma de las utilidades descontadas de todas las decisiones futuras. Formalmente, si $r_t$ es la recompensa en el tiempo $t$, la utilidad total es $\sum_{t=0}^{\infty}\gamma^t r_t$. Este concepto es fundamental en muchos algoritmos de aprendizaje por refuerzo, como el Q-learning y el algoritmo de política de valor, porque ayuda a equilibrar la explotación de recompensas inmediatas con la exploración de recompensas futuras.

## 9.4. Aprendizaje por refuerzo y sus aplicaciones

E l **aprendizaje por refuerzo** (*reinforcement learning,* RL) es una rama del aprendizaje automático que se centra en cómo los agentes deben tomar decisiones para maximizar alguna noción de recompensa acumulada a lo largo del tiempo. En este contexto, los agentes aprenden a interactuar con un entorno para descubrir las mejores estrategias o políticas a seguir. Para entender mejor el aprendizaje por refuerzo, es esencial familiarizarse con los procesos de decisión de Markov (MDP), que proporcionan el marco matemático subyacente y que vimos anteriormente.

Las ecuaciones de Bellman juegan un papel fundamental en la solución de MDPs y, por ende, en el aprendizaje por refuerzo. Estas ecuaciones descomponen el problema de encontrar la política óptima en subproblemas más pequeños y manejables.

### Ecuaciones de Bellman

Las **ecuaciones de Bellman,** nombradas en honor al matemático Richard Bellman, son una formulación matemática que descompone un problema de decisión secuencial en subproblemas más pequeños y manejables. Estas ecuaciones se aplican tanto en la optimización estática como dinámica, siendo especialmente útiles en contextos del aprendizaje por refuerzo.

Antes de abordar las ecuaciones, es útil entender algunos términos clave. Al igual que los MDP, en las ecuaciones de Bellman existe el concepto de estado, acción, recompensa y política. Acá se agrega el concepto de valor o utilidad esperada.

- **Estado (s):** la situación actual en la que se encuentra el agente.
- **Acción (a):** una decisión que el agente puede tomar desde un estado.
- **Recompensa (r):** el beneficio inmediato obtenido al tomar una acción en un estado.

- **Valor (V):** la utilidad esperada total que se puede obtener a partir de un estado, siguiendo una política.
- **Política (π):** regla que define qué acción tomar en cada estado.

$$V^{\pi}(s) = E[r_t + \gamma V^{\pi}(s_{t+1}) \vee s_t = s, \pi]$$

Donde:

- $V^{\pi}(s)$ es el valor del estado s siguiendo la política π.
- $r_t$ es la recompensa inmediata al tomar la acción a en el estado s.
- $\gamma$ es el factor de descuento.
- $s_{t+1}$ es el estado siguiente.
- La expectativa E se toma sobre todas las acciones y estados posibles, dado que se sigue la política π.

Las ecuaciones de Bellman descomponen el problema de optimización a largo plazo en una serie de decisiones más pequeñas, facilitando su resolución. Son la base para varios algoritmos de aprendizaje por refuerzo, como el v*alue iteration, policy iteration* y Q-learning. Ayudan a identificar la política óptima que maximiza la recompensa total esperada en problemas de decisión secuencial.

Imaginemos un agente en un entorno simple con tres estados (A, B, C) y dos acciones posibles (X, Y). La ecuación de Bellman se utilizaría para actualizar los valores estimados de cada estado o estado-acción basado en las recompensas inmediatas y los valores futuros esperados.

Veamos un ejemplo de agente en una cuadrícula.

**Ejemplo de agente cuadrícula**

# Ideas clave

Supongamos un agente en una cuadrícula de 3x3 que puede moverse en cuatro direcciones: arriba, abajo, izquierda y derecha. El objetivo del agente es llegar a una celda de objetivo (recompensa +1) y evitar una celda peligrosa (recompensa -1). Todas las demás celdas tienen una recompensa de 0. El factor de descuento, γ, es 0.9.

Configuración del problema:

- Estados (s): cada celda en la cuadrícula es un estado.
- Acciones (a): {Arriba, Abajo, Izquierda, Derecha}.
- Recompensas (r): +1 en la celda objetivo y -1 en la celda peligrosa.
- 0 en todas las demás celdas.
- Factor de descuento (γ): 0.9.

La cuadrícula se ve así:

| | | |
|---|---|---|
| S | 0 | G |
| 0 | -1 | 0 |
| 0 | 0 | 0 |

Tabla 1. Cuadrícula. Fuente: elaboración propia.

Donde S es el estado inicial, G es la celda objetivo y -1 es la celda peligrosa.

Aplicamos las ecuaciones de Bellman.

Primero, inicializamos los valores de estado V(s) a 0 para todas las celdas, excepto las celdas objetivo y peligrosa, que se inicializan con sus recompensas directas.

| | | |
|---|---|---|
| 0 | 0 | +1 |
| 0 | -1 | 0 |
| 0 | 0 | 0 |

Tabla 2. Cuadrícula. Fuente: elaboración propia.

Iteración de valores.

Utilizamos la ecuación de Bellman para actualizar los valores de estado iterativamente hasta que converjan. La ecuación de Bellman para el valor del estado es:

$$V(s) = max_a \sum_{s'} P\left(s' \vee s,a\right)\left[R\left(s,a,s'\right) + \gamma V(s')\right]$$

Donde $P\left(s' \vee s,a\right)$ es la probabilidad de transición al estado $s'$ dado el estado actual $s$ y la acción $a$. Asumiremos que las transiciones son determinísticas para este ejemplo.

Primera iteración.

Tomemos un estado no terminal, por ejemplo, la celda (0,1). Evaluamos el valor del estado V(0,1) considerando las posibles acciones:

- Arriba: no es posible (borde de la cuadrícula).
- Abajo: transición a (1,1) con V(1,1) = -1.
- Izquierda: transición a (0,0) con V(0,0) =0.
- Derecha: transición a (0,2) con V(0,2) =1.

Usamos la ecuación de Bellman.

V(0,1)=max{0+0,0+0.9·(−1),0+0.9·0,0+0.9·1}

$V(0,1)=\max\{0,-0.9,0,0.9\}$

V(0,1)=0.9

Actualizamos la tabla:

| | | |
|---|---|---|
| 0 | 0.9 | +1 |
| 0 | -1 | 0 |
| 0 | 0 | 0 |

Tabla 3. Cuadrícula. Fuente: elaboración propia.

Repetimos este proceso para todas las celdas hasta que los valores converjan.

Después de suficientes iteraciones, los valores convergerán a los valores óptimos. Para nuestro ejemplo, los valores convergentes pueden verse así:

| | | |
|---|---|---|
| 0.81 | 0.9 | +1 |
| 0.729 | -1 | 0.81 |
| 0.6561 | 0.729 | 0.81 |

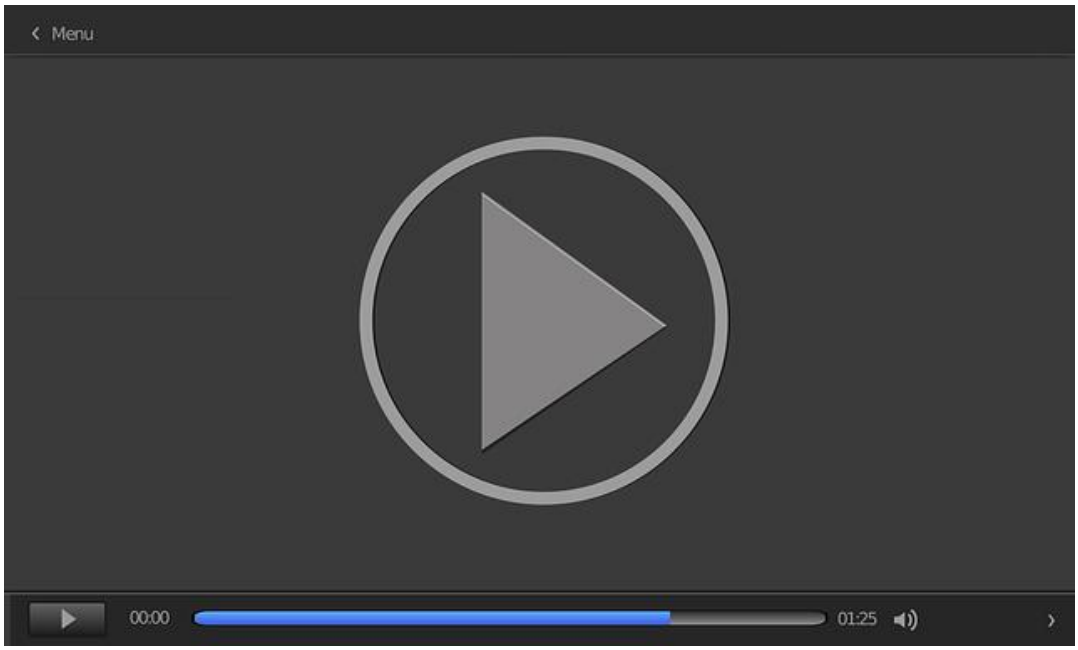
Tabla 4. Cuadrícula. Fuente: elaboración propia.

Una vez que los valores han convergido, podemos derivar la política óptima seleccionando la acción que maximiza el valor esperado en cada estado. Para cada estado $s$ .

En este ejemplo partimos de la celda (0,0), seguimos con la celda (0,1) y finalizamos

con la celda (0,3) que es la celda objetivo y el camino es el que mayor utilidad nos proporciona.

A continuación, veremos el vídeo ***Aplicaciones prácticas del aprendizaje por refuerzo.***




Accede al vídeo:

https://unir.cloud.panopto.eu/Panopto/Pages/Embed.aspx?id=85569be3-3884-4736-841a-b1c501149703

## Aplicaciones del aprendizaje por refuerzo (Mwiti, 2023)

**Aplicaciones en conducción autónoma**

Varios artículos han propuesto el aprendizaje por refuerzo profundo para la conducción autónoma. En los vehículos autónomos hay varios aspectos a considerar, como los límites de velocidad en varios lugares, las zonas de conducción y evitar colisiones, solo por mencionar algunos.

Algunas de las tareas de conducción autónoma en las que se podría aplicar el

aprendizaje por refuerzo incluyen la optimización de trayectorias, la planificación del movimiento, la trayectoria dinámica, la optimización del controlador y las políticas de aprendizaje basadas en escenarios para carreteras.

Por ejemplo, el estacionamiento se puede lograr aprendiendo políticas de estacionamiento automático. El cambio de carril se puede lograr utilizando Q-learning, mientras que los adelantamientos se pueden implementar aprendiendo una política de adelantamiento evitando colisiones y manteniendo una velocidad constante a partir de entonces.

AWS DeepRacer es un coche de carreras autónomo que ha sido diseñado para probar RL en una pista física. Utiliza cámaras para visualizar la pista y un modelo de aprendizaje de refuerzo para controlar el acelerador y la dirección (Mwiti, 2023).

**Automatización de la industria con aprendizaje por refuerzo**

En el refuerzo industrial, se utilizan robots basados en el aprendizaje por refuerzo para realizar diversas tareas. Los robots pueden realizar tareas que serían peligrosas para las personas.

Un gran ejemplo es el uso de agentes de inteligencia artificial por parte de Deepmind para enfriar los centros de datos de Google. Esto llevó a una reducción del 40 % en el gasto energético. Los centros ahora están totalmente controlados con el sistema de IA sin necesidad de intervención humana. Obviamente, todavía hay supervisión por parte de expertos en centros de datos (Olaoye y Potter, 2024). El sistema funciona de la siguiente manera:

- Tomar instantáneas de los datos de los centros de datos cada cinco minutos y enviarlas a redes neuronales profundas.
- Luego predice cómo las diferentes combinaciones afectarán los consumos de energía futuros.

- Identificar acciones que conduzcan a un consumo mínimo de energía manteniendo al mismo tiempo un estándar establecido de criterios de seguridad.
- Enviar e implementar estas acciones en el centro de datos.
- Las acciones son verificadas por el sistema de control local.

**Aplicaciones de aprendizaje por refuerzo en comercio y finanzas**

Los modelos de series de tiempo supervisados se pueden utilizar para predecir ventas futuras, así como para predecir los precios de las acciones. Sin embargo, estos modelos no determinan la acción a tomar ante un precio de acción en particular. Un agente de RL puede decidir sobre dicha tarea; si mantener, comprar o vender. El modelo RL se evalúa utilizando estándares de referencia del mercado para garantizar que funcione de manera óptima.

Esta automatización aporta coherencia al proceso, a diferencia de los métodos anteriores en los que los analistas tenían que tomar todas las decisiones. IBM, por ejemplo, tiene una sofisticada plataforma basada en el aprendizaje por refuerzo que tiene la capacidad de realizar transacciones financieras. Calcula la función de recompensa en función de la pérdida o ganancia de cada transacción financiera (Olaoye y Potter, 2024).

**Aprendizaje por refuerzo en PLN (procesamiento del lenguaje natural)**

En PLN, RL se puede utilizar para resumir textos, responder preguntas y traducir automáticamente, solo por mencionar algunos.

Investigadores de la Universidad de Stanford, la Universidad Estatal de Ohio y Microsoft Research han desarrollado *deep* RL para su uso en la generación de diálogos. El RL profundo se puede utilizar para modelar recompensas futuras en un diálogo de *chatbot.* Las conversaciones se simulan mediante dos agentes virtuales. Los métodos de gradiente de políticas se utilizan para recompensar secuencias que

contienen atributos de conversación importantes, como coherencia, informatividad y facilidad de respuesta (Mwiti, 2023).

**Aprendizaje por refuerzo en atención sanitaria**

En el sector sanitario, los pacientes pueden recibir tratamiento a partir de políticas aprendidas de los sistemas RL. RL es capaz de encontrar políticas óptimas utilizando experiencias previas sin necesidad de información previa sobre el modelo matemático de sistemas biológicos. Hace que este enfoque sea más aplicable que otros sistemas basados en control en la atención sanitaria.

La RL en atención médica se clasifica como regímenes de tratamiento dinámico (DTR) en enfermedades crónicas o cuidados críticos, diagnóstico médico automatizado y otros dominios generales.

En las DTR, la entrada es un conjunto de observaciones y evaluaciones clínicas de un paciente. Los resultados son las opciones de tratamiento para cada etapa. Estos son similares a los estados de RL. La aplicación de RL en DTR es ventajosa porque es capaz de determinar decisiones dependientes del tiempo para el mejor tratamiento para un paciente en un momento específico.

El uso de RL en la atención sanitaria también permite mejorar los resultados a largo plazo al tener en cuenta los efectos retardados de los tratamientos.

RL también se ha utilizado para el descubrimiento y generación de DTR óptimas para enfermedades crónicas (Olaoye y Potter, 2024).

**Aplicaciones del aprendizaje por refuerzo en ingeniería**

Facebook ha desarrollado una plataforma de aprendizaje por refuerzo de código abierto: Horizon. La plataforma utiliza el aprendizaje por refuerzo para optimizar los sistemas de producción a gran escala. Facebook ha utilizado Horizon internamente para personalizar sugerencias, entregar notificaciones más significativas a los

usuarios y optimizar la calidad de la transmisión de vídeo.

Otra aplicación del RL es en la visualización de vídeo. Ofrece al usuario un vídeo con una velocidad de *bits* baja o alta según el estado de los *buffers* de vídeo y las estimaciones de otros sistemas de aprendizaje automático (Mwiti, 2023).

**Aprendizaje por refuerzo en recomendación de noticias**

Las preferencias de los usuarios pueden cambiar con frecuencia, por lo que recomendar noticias a los usuarios basándose en reseñas y me gusta podría quedar obsoleto rápidamente. Con el aprendizaje por refuerzo, el sistema RL puede rastrear los comportamientos de retorno del lector.

La construcción de dicho sistema implicaría obtener características de noticias, características de lector, características de contexto y características de noticias de lector. Las características de las noticias incluyen, entre otras, el contenido, el titular y el editor. Las características del lector se refieren a cómo el lector interactúa con el contenido, por ejemplo, hace clic y comparte. Las características de contexto incluyen aspectos de las noticias como el momento y la actualidad de las noticias. Luego se define una recompensa en función de estos comportamientos de los usuarios (Mwiti, 2023).

## 9.5. Cuaderno de ejercicios

### Ejercicio 1. Definición de agentes

Define un agente inteligente y describe las características que lo diferencian de un programa de *software* tradicional. Luego, proporciona un ejemplo de un agente inteligente en un entorno real.

**Solución**

Un agente inteligente es un programa de *software* que opera de forma autónoma, percibe su entorno mediante sensores y actúa sobre él mediante actuadores para alcanzar ciertos objetivos. Las características que lo diferencian de un programa de *software* tradicional incluyen autonomía, percepción y acción, objetivo o función de recompensa, adaptabilidad y persistencia.

Ejemplo: un robot aspirador en un hogar es un agente inteligente. Utiliza sensores para detectar obstáculos y suciedad (percepción), y actuadores para moverse y limpiar (acción). Su objetivo es maximizar la limpieza del área (recompensa). Puede adaptarse a diferentes configuraciones de muebles y aprender las mejores rutas para limpiar eficientemente (adaptabilidad y persistencia).

### Ejercicio 2. Componentes de los MDP

Describe los componentes de un proceso de decisión de Markov (MDP). Proporciona un ejemplo simple que ilustre cada uno de estos componentes.

**Solución**

Un agente en una cuadrícula 3x3 con el objetivo de alcanzar la celda superior derecha (G):

- **Estados (S):** cada celda en la cuadrícula.

- **Acciones (A):** {Arriba, Abajo, Izquierda, Derecha}.
- **Transiciones (T):** probabilidad de moverse a una celda adyacente según la acción tomada.
- **Recompensas (R):** +1 en la celda objetivo (G), -1 en celdas peligrosas y 0 en otras celdas.
- **Política (π):** plan para moverse hacia la celda objetivo evitando las celdas peligrosas.

## Ejercicio 3. Ecuaciones de Bellman

Explica la ecuación de Bellman para la utilidad esperada de un estado V(s) y cómo se aplica en el contexto del aprendizaje por refuerzo. Luego, aplica la ecuación de Bellman a un estado s en una cuadrícula 2x2 con una celda objetivo (G) y una celda peligrosa (-1).

**Solución**

La ecuación de Bellman para la utilidad esperada de un estado V(s) se define como: $V(s) = max_a \sum_{s'} P\left(s' \vee s,a\right)\left[R\left(s,a,s'\right) + \gamma V(s')\right]$, donde γ es el factor de descuento.

Aplicación (considera una cuadrícula 2x2):

- Estados: (0,0), (0,1), (1,0), (1,1).
- Recompensas: +1 en (0,1), -1 en (1,0) y 0 en otros.
- Factor de descuento γ=0.9.
- Para el estado (0,0), evaluamos las posibles acciones (suponiendo transiciones determinísticas):

Arriba: no aplicable.

- Abajo: (1,0) con V(1,0)=−1.
- Izquierda: no aplicable.
- Derecha: (0,1) con V(0,1)=1.

- Usamos la ecuación de Bellman:
  V(0,0)=max{0+0.9·(−1),0+0.9·1}=max{−0.9,0.9}=0.9.

## Ejercicio 4. Seudocódigo

Proporciona el seudocódigo para un agente que debe aprender a moverse en una cuadrícula de 2x2 con una celda objetivo (+1) y una celda peligrosa (-1).

**Solución**

- **Inicialización:** establecer los valores para todas las parejas estado-acción.
- **Episodios:** repetir el proceso para varios episodios.
- **Selección de acción:** elegir una acción basada en la mejor política.
- **Actualización:** actualizar el valor usando la fórmula de Bellman.
- **Transición:** moverse al nuevo estado y repetir hasta alcanzar un estado terminal.

Este algoritmo permite al agente aprender la mejor acción a tomar en cada estado para maximizar la recompensa total esperada.

## 9.6. Referencias bibliográficas

Kaelbling, L. P., Littman, M. L. y Moore, A. W. (1996). Reinforcement learning: A survey. *Journal of artificial intelligence research, 4,* 237-285.

Mwiti, D. (2023, septiembre 1). *10 Real-life applications of reinforcement learning.* Neptune. https://neptune.ai/blog/reinforcement-learning-applications

Olaoye F. y Potter, K. (2024). *Reinforcement Learning and its Real-World Applications.* ReserarchGate. https://www.researchgate.net/publication/379025393

Puterman, M. L. (1990). Markov decision processes. *Handbooks in operations research and management science, 2,* 331-434.

Szepesvári, C. (2022). *Algorithms for reinforcement learning.* Springer nature.

Wei, Z., Xu, J., Lan, Y., Guo, J. y Cheng, X. (2017, August). Reinforcement learning to rank with Markov decision process. *Proceedings of the 40th international ACM SIGIR conference on research and development in information retrieval,* 945-948.

## ¿Qué es el aprendizaje por refuerzo y cómo trabaja?

Kadari, P. (2024, mayo 26). *What is reinforcement learning and how does it work.* Analytics Vidhya. https://www.analyticsvidhya.com/blog/2021/02/introduction-to-reinforcement-learning-for-beginners/

El aprendizaje por refuerzo *(reinforcement learning)* es una técnica de *machine learning* en la que un agente aprende a tomar decisiones óptimas mediante la interacción con su entorno, buscando maximizar recompensas acumuladas. Este proceso implica experimentar acciones, recibir recompensas o penalizaciones, y ajustar estrategias para mejorar el rendimiento futuro sin intervención humana directa.

# Aprendizaje por refuerzo a partir de la retroalimentación humana (RLHF).

AWS (s. f.). *¿Qué es el RLHF?* https://aws.amazon.com/es/what-is/reinforcement-learning-from-human-feedback/

El aprendizaje por refuerzo a partir de la retroalimentación humana (RLHF) es una técnica de *machine learning* que mejora la precisión de los modelos al incorporar *feedback* humano en la función de recompensas. Esto permite que los modelos realicen tareas de manera más alineada con los objetivos humanos, optimizando su desempeño y generando respuestas más naturales y satisfactorias en aplicaciones como *chatbots,* generación de imágenes y música.

# Test

**1.** ¿Qué es un agente en el contexto del aprendizaje por refuerzo?

A. Un algoritmo que siempre sigue instrucciones predefinidas.

B. Un programa que interactúa con el entorno y aprende de sus acciones.

C. Un dispositivo de *hardware* que ejecuta comandos.

D. Un conjunto de datos utilizado para entrenar modelos de aprendizaje automáticos.

**2.** ¿Cuál de las siguientes características no es propia de un agente inteligente?

A. Autonomía.

B. Percepción y acción.

C. Almacenamiento de datos en la nube.

D. Adaptabilidad.

**3.** ¿Qué representan los estados (S) en un proceso de decisión de Markov (MDP)?

A. Todas las posibles acciones que un agente puede realizar.

B. Todos los posibles resultados de una acción.

C. Todas las posibles situaciones en las que puede encontrarse el agente.

D. Las recompensas acumuladas en el tiempo.

**4.** ¿Cuál es el propósito principal de la ecuación de Bellman en aprendizaje por refuerzo?

A. Predecir las acciones futuras de otros agentes.

B. Descomponer problemas de optimización a largo plazo en subproblemas manejables.

C. Almacenar datos históricos de las recompensas.

D. Aumentar la velocidad de procesamiento de los algoritmos.

**5.** ¿Qué se entiende por política (π) en el contexto de un MDP?

A. Una medida de la recompensa inmediata de una acción.

B. Una estrategia que el agente sigue para decidir qué acción tomar en cada estado.

C. La probabilidad de transición entre estados.

D. El valor esperado de todas las recompensas futuras.

**6.** ¿Qué tipo de agente utiliza un modelo interno del entorno para tomar decisiones más informadas?

A. Agente reactivo simple.

B. Agente basado en objetivos.

C. Agente basado en modelo.

D. Agente basado en utilidad.

**7.** ¿Cuál es el factor de descuento (γ) en el contexto de la utilidad descontada?

A. La tasa a la que se actualizan los estados.

B. La probabilidad de transitar a un nuevo estado.

C. La frecuencia con la que se recompensan las acciones.

D. Un valor que refleja la importancia relativa de las recompensas futuras frente a las inmediatas.

**8.** En el aprendizaje por refuerzo, ¿qué función cumple una recompensa (R)?

A. Determina la probabilidad de una transición entre estados.

B. Proporciona una medida inmediata de la bondad de una acción tomada.

C. Define la política que un agente debe seguir.

D. Establece el conjunto de posibles estados.

**9.** ¿Qué significa que una política es estacionaria?

A. Las decisiones del agente dependen del tiempo.

B. La política no se puede modificar una vez establecida.

C. Las decisiones del agente dependen únicamente del estado actual, no del tiempo.

D. La política cambia en cada iteración.

**10.** ¿Cuál de las siguientes es una aplicación del aprendizaje por refuerzo en la vida real?

A. Reconocimiento facial.

B. Optimización de motores de búsqueda.

C. Conducción autónoma.

D. Clasificación de correos electrónicos.