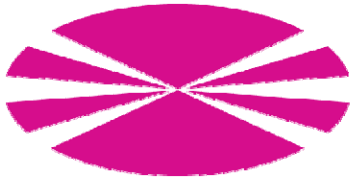


UNIVERSIDADE DA CORUÑA

# APRENDIZAJE AUTOMÁTICO

## TEMA 4:

## APRENDIZAJE POR REFUERZO

# TEMA 4

## APRENDIZAJE POR REFUERZO

Fig. 4

# TEMA 4

## APRENDIZAJE POR REFUERZO

- Introducción al Aprendizaje por Refuerzo
- Elementos del AR
- Aplicaciones del AR
- Procesos de Decisión de Markov
- Taxonomía de los métodos de AR
- Q-Learning
- Ejemplos de Aplicación
- AR y Robots Autónomos
- Conclusiones

# Tema 4. Aprendizaje por Refuerzo
## Bibliografía


- D. Borrajo, J. González y P. Isasi. Aprendizaje Automático. Ed. Sanz y Torres S.L. 2006
- R. Sutton, A. Barto. Reinforcement Learning. An Introduction. The MIT Press. 1998.
- T. Mitchell. Machine Learning. McGraw-Hill. 1997.

# APRENDIZAJE AUTOMÁTICO

## DIFERENTES APRENDIZAJES

- **Aprendizaje supervisado:** el comportamiento deseado se representa por medio de un conjunto de ejemplos. Estos ejemplos permiten definir un criterio para evaluar el comportamiento real del sistema.
- **Aprendizaje no supervisado:** sin ningún tipo de señal que indique un comportamiento deseado para el sistema. El criterio de evaluación se basa en la regularidad de los grupos de datos identificados.

- **Aprendizaje por refuerzo:** el comportamiento deseado no se representa mediante ejemplos, sino mediante una cierta evaluación sobre los resultados que genera el sistema en su entorno. ***Consiste en aprender a decidir mediante prueba y error, ante una situación determinada, qué acción es la más adecuada para lograr un objetivo.***
  - Ej. Jugar a las damas; invertir en bolsa; conducir un vehículo.

# APRENDIZAJE AUTOMÁTICO

## APRENDIZAJE POR REFUERZO

- Ej.: Se quiere construir un sistema que **aprenda a jugar a las damas**.
  - Cada **INSTANCIA** de aprendizaje (ESTADO, ACCION) viene dada por: nº piezas propias y del contrario, nº damas propias y del contrario, nº de diagonales controladas por nuestras piezas.
  - La tarea que se quiere mejorar es la de **predecir** si, dada una determinada situación o ESTADO (configuración concreta de un tablero) una determinada ACCION puede llevar a que el ordenador gane.
  - La "clase" sería: una estimación de la probabilidad de ganar en el futuro si llevásemos a cabo esa ACCION, en esa situación.
  - Para conocer la "clase" a la que pertenece cada INSTANCIA, el ordenador debe esperar a que termine el juego para saber si ha ganado, perdido o empatado, la "clase" **se conoce una vez que se han tomado una secuencia de decisiones**.
  - Además, la "clase" no se conoce de forma exacta, porque desde una instancia se llega a una situación ganadora o no, no siempre se llega a la misma situación, depende de cómo se actúe posteriormente y sobre todo, de CÓMO actúe el adversario en cada caso.
- Por eso se le llama **REFUERZO** y no, clase (esto deriva de ciencias como la Psicología, Etología o Biología)

# APRENDIZAJE AUTOMÁTICO

## APRENDIZAJE POR REFUERZO

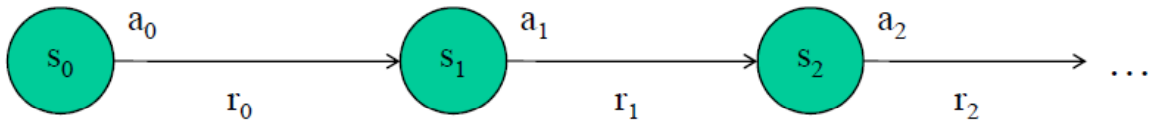
- Dentro del amplio abanico de sistemas con un comportamiento inteligente se encuentran, por tanto, algunos donde **la interacción con el entorno** es muy activa y dinámica:
  - Un robot que aprende a navegar en su entorno; un coche que conduzca autónomamente.
- A la hora de proporcionar inteligencia a estos sistemas, aparecen ciertas características que los definen:
  - El aprendizaje de una tarea por parte del sistema o agente se realiza mediante un proceso iterativo de prueba y error en el entorno con el que interactúa.
  - La forma en que el entorno informa al agente sobre si está haciendo bien o mal la tarea que está aprendiendo, se realiza a través de una ***señal de REFUERZO***, que puede recibirse retardada en el tiempo.
    - La idea es que los agentes aprendan a comportarse de manera cuasi-óptima solamente guiados por su afán de maximizar una señal de refuerzo, pero sin un experto que les indica qué acciones tomar en cada momento.
    - Ej. Al entrenar un agente para jugar a un juego, se le asigna una recompensa positiva al ganar el juego, una negativa al perder, y cero en cualquier otro estado.
  - El sistema realiza una tarea repetidamente para adquirir experiencia y mejorar su comportamiento.

# APRENDIZAJE AUTOMÁTICO

## APRENDIZAJE POR REFUERZO

- Ejemplo 1: un agente software que aprende a jugar al tres en raya
  - El agente percibe la configuración del tablero (estado)
  - El agente puede colocar sus fichas (acciones)
  - El agente quiere aprender, para cada configuración del tablero, dónde colocar su próxima ficha (política de control)

- Ejemplo 2: un robot que aprende a navegar en su entorno
  - El robot tiene sensores para observar las características de su entorno (estado)
  - El robot puede moverse, coger objetos, etc., (acciones) modificando así el estado de su entorno
  - El robot quiere aprender un mapeado estado-acción para alcanzar sus objetivos (política de control)



# APRENDIZAJE AUTOMÁTICO

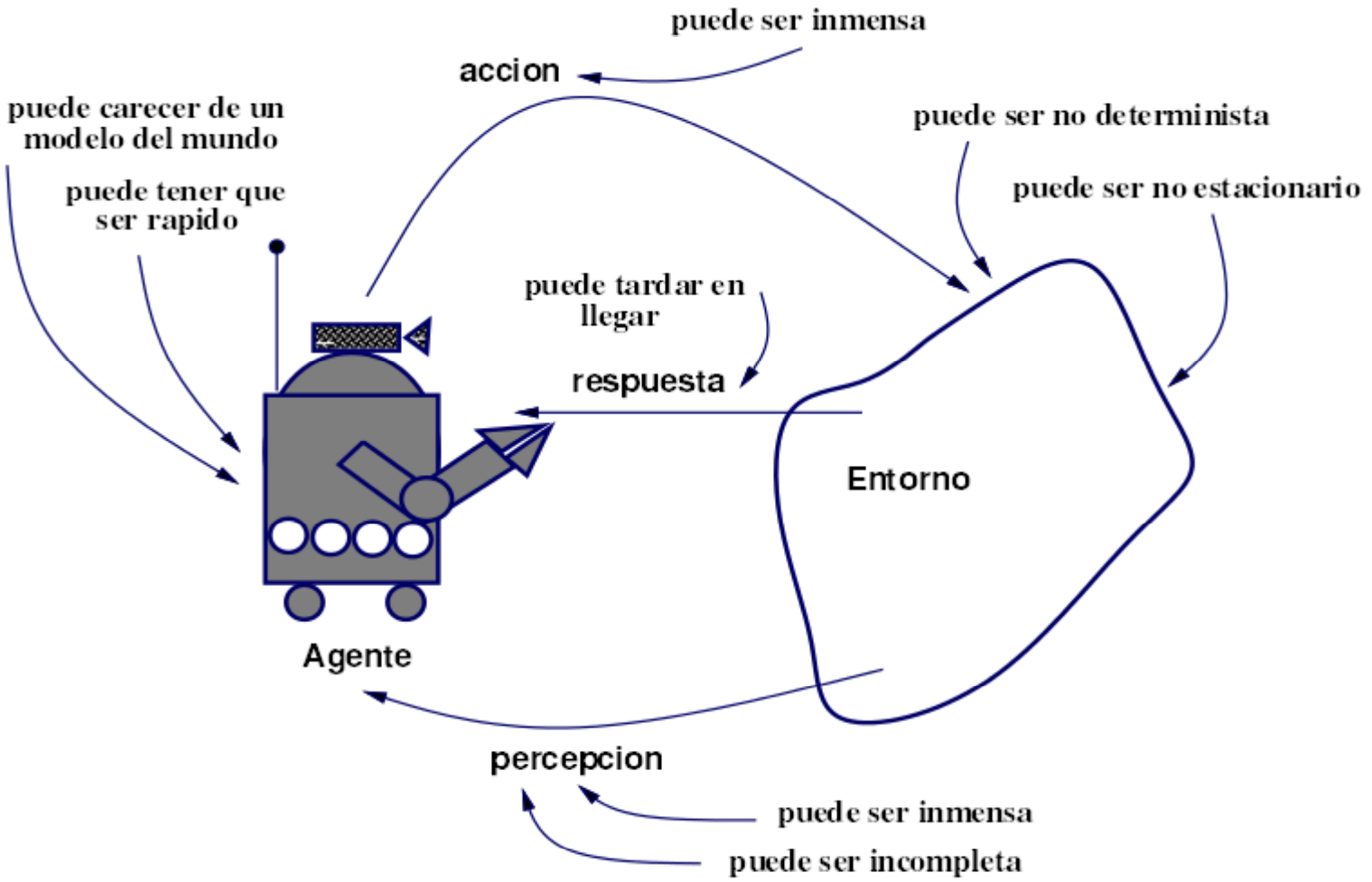
## APRENDIZAJE POR REFUERZO

**Ejemplo del robot:**

- Que un robot aprenda la mejor secuencia de movimientos que le permita avanzar hacia un objeto.
- Debe aprender mediante interacción directa con el entorno, siguiendo un algoritmo iterativo, maximizando su refuerzo:
  - El robot recibe información de la distancia que avanza después de ejecutar una acción, y con base en esta información, obtiene un refuerzo según la distancia que logre avanzar tras cada intento.
  - Para aplicar el algoritmo de aprendizaje, se define primero el conjunto de estados y acciones que el robot puede realizar (puede ser necesaria una discretización, cuando son muchos).
  - Para calcular **el refuerzo**, se emplea un sonar del robot: el sonar realiza inicialmente una medición y obtiene un valor de distancia hacia el objeto de referencia.
  - A partir de ese valor inicial, después de la ejecución de cada acción, el sonar realiza otra medición. Con los 2 valores, el anterior y el actual, realiza una resta y compara si la distancia es mayor que un umbral definido.
    - Si es mayor, entonces se obtiene un refuerzo de 10, en caso contrario se obtiene un refuerzo de 0 (se van anotando los refuerzos que se obtienen en cada caso).
    - Pero pueden variar aspectos del entorno inesperadamente (moverse el objeto).

# APRENDIZAJE AUTOMÁTICO

## APRENDIZAJE POR REFUERZO



# APRENDIZAJE AUTOMÁTICO

## APRENDIZAJE POR REFUERZO

- Cómo realizar ese proceso de prueba y error y cómo tratar esta señal de refuerzo para que el sistema aprenda de forma eficiente un comportamiento, deseablemente óptimo, ha sido objeto de estudio y se ha unificado bajo el término de **aprendizaje por refuerzo.**

- La aparición de nuevos campos como la robótica, dieron un auge importante a este tipo de técnicas, definiendo como objetivo la búsqueda de **políticas** de comportamiento óptimas para realizar determinadas tareas.
  - Un agente que utiliza aprendizaje por refuerzo aprende al interactuar con su entorno observando los resultados de esas interacciones.
  - No se conoce la salida adecuada, solo que el efecto de esta salida sobre el entorno tiene que ser tal que se maximice la recompensa a largo plazo.

- Esto imita al modo fundamental en el que los humanos y algunos animales aprenden.
  - Estudios en aprendizaje animal muestran cómo los animales pueden aprender secuencias de acciones arbitrarias solamente maximizando refuerzos recibidos.

# APRENDIZAJE AUTOMÁTICO

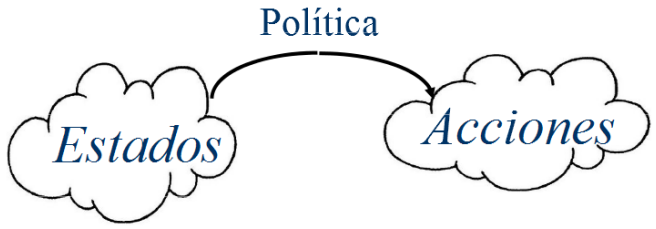
## APRENDIZAJE POR REFUERZO

- Especialmente orientado a agentes que ***interaccionan con el entorno***
  - El entorno ha de **cuantificar** el éxito o fracaso de las acciones

- **No hay ejemplos:** a diferencia de otros métodos de aprendizaje, los ejemplos los obtiene el agente interactuando con el entorno.

- El sistema aprende mediante prueba y error.

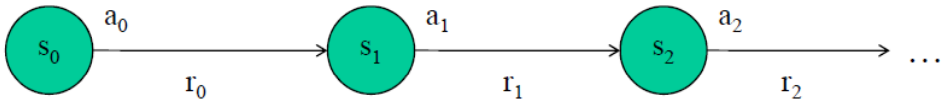
- EXPLORACIÓN del ENTORNO para obtener MODELO/POLÍTICA de COMPORTAMIENTO que maximice alguna recompensa a largo plazo.




- Deben existir: percepción, acción y objetivo.

# APRENDIZAJE AUTOMÁTICO

## APRENDIZAJE POR REFUERZO




- Formalización del problema:
  - El agente quiere aprender la política **π : S --> A** que produzca el mayor refuerzo acumulado en el tiempo, a partir de cualquier estado **st**

$$r_t + \gamma\, r_{t+1} + \gamma^2\, r_{t+2} + \ldots = \sum_{i=0}^{\infty} \gamma^i\, r_{t+i} = V^{\pi}(s_t)$$

  - $V^{\pi}(s_t)$ : función de valor que indica el **refuerzo acumulado** que se consigue siguiendo la política **π** a partir de un estado inicial **st** (i.e., "utilidad" de **st**)
  - $\gamma$ es una constante ($0 < \gamma < 1$) que determina la importancia relativa de los refuerzos inmediatos respecto a los refuerzos futuros.
    - Dado que el refuerzo esperado en el futuro es más inseguro cuanto mayor sea la distancia desde el momento actual, se eleva al n° de instantes de tiempo.
    - Si $\gamma = 0$, el agente es "miope" y solo maximiza los refuerzos inmediatos
    - Si $\gamma = 1$, el agente es más precavido, (el refuerzo futuro también le importa mucho)

# APRENDIZAJE AUTOMÁTICO

## APRENDIZAJE POR REFUERZO

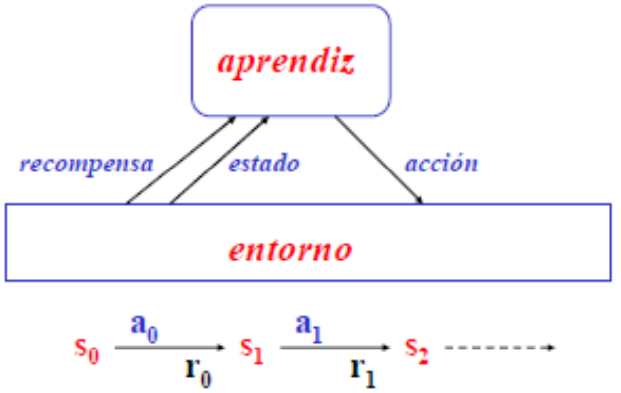
- Diferencias respecto a aprendizaje supervisado:
  - El agente no dispone de información de entrenamiento tipo **< s, π *(s) >,** donde **π*(s)** es la ***política óptima que se quiere aprender***, sino de información del tipo **<s, a, r>,** donde **r** es el refuerzo inmediato que se recibe al ejecutar la acción **a** en el estado **s**.
  - El entorno no dice al agente "en el estado s6 deberías haber ejecutado la acción a3 en vez que la acción a9", sino le dice: "en el estado s6 la ejecución de la acción a9 vale 34.5"
  - El problema de la **exploración-explotación**: la información disponible depende de las acciones ejecutadas, por lo tanto el agente tiene que explorar el espacio de acciones, balanceando la ejecución de acciones que se sabe son buenas (**explotación**) y de acciones que nunca se han probado (**exploración**).

# APRENDIZAJE AUTOMÁTICO

## APRENDIZAJE POR REFUERZO

- Los pasos básicos de un agente/aprendiz que emplea aprendizaje por refuerzo son:

1. El agente observa un estado
2. Una acción es determinada por una función de toma de decisiones (política)
3. Se ejecuta la acción
4. El agente recibe un refuerzo/recompensa escalar desde el entorno
5. La información sobre el refuerzo dado por ese par estado/acción se graba.




- OBJETIVO: Realizando acciones y observando la recompensa resultante, puede **optimizarse la política usada para determinar la mejor acción a realizar para un estado**.
- Si se observan suficientes estados, se generará una ***política de decisiones*** óptima y el agente actuará perfectamente en ese entorno.
  - Ej. estados: situación del robot en un laberinto; situación concreta de las piezas en un tablero.
  - Ej. acciones: movimientos robot (izq, drcha, arriba, abajo); movimientos válidos de fichas en las damas.

# APRENDIZAJE POR REFUERZO

## Elementos del aprendizaje por refuerzo

- **Elementos del aprendizaje por refuerzo:**
  - *Agente*
  - *Entorno*
  - La *política,* define el comportamiento del aprendiz en cada momento
  - La *función de refuerzo,* define el refuerzo para cada acción
  - La *función de acción-valor,* permite establecer la recompensa a largo plazo estimada a partir de cada posible acción
  - Opcionalmente un *modelo del entorno*

- **Agente**: Es el sujeto del aprendizaje por refuerzo. Lee el estado del entorno, realiza acciones sobre el entorno y lee las recompensas que producen estas acciones.
- **Entorno:** Es el "mundo" sobre el que opera el agente. El entorno recibe las acciones del agente y evoluciona. Su comportamiento suele ser desconocido y estocástico. Es el responsable de generar las recompensas asociadas a las acciones y cambios de estado.
- **Política:** Define el comportamiento del agente. Puede verse como un mapeo de estados en acciones si es determinista (determinista: dado un estado y una acción siempre se transita al mismo estado) y acciones en probabilidades si es estocástica.

# APRENDIZAJE POR REFUERZO

## Elementos del aprendizaje por refuerzo

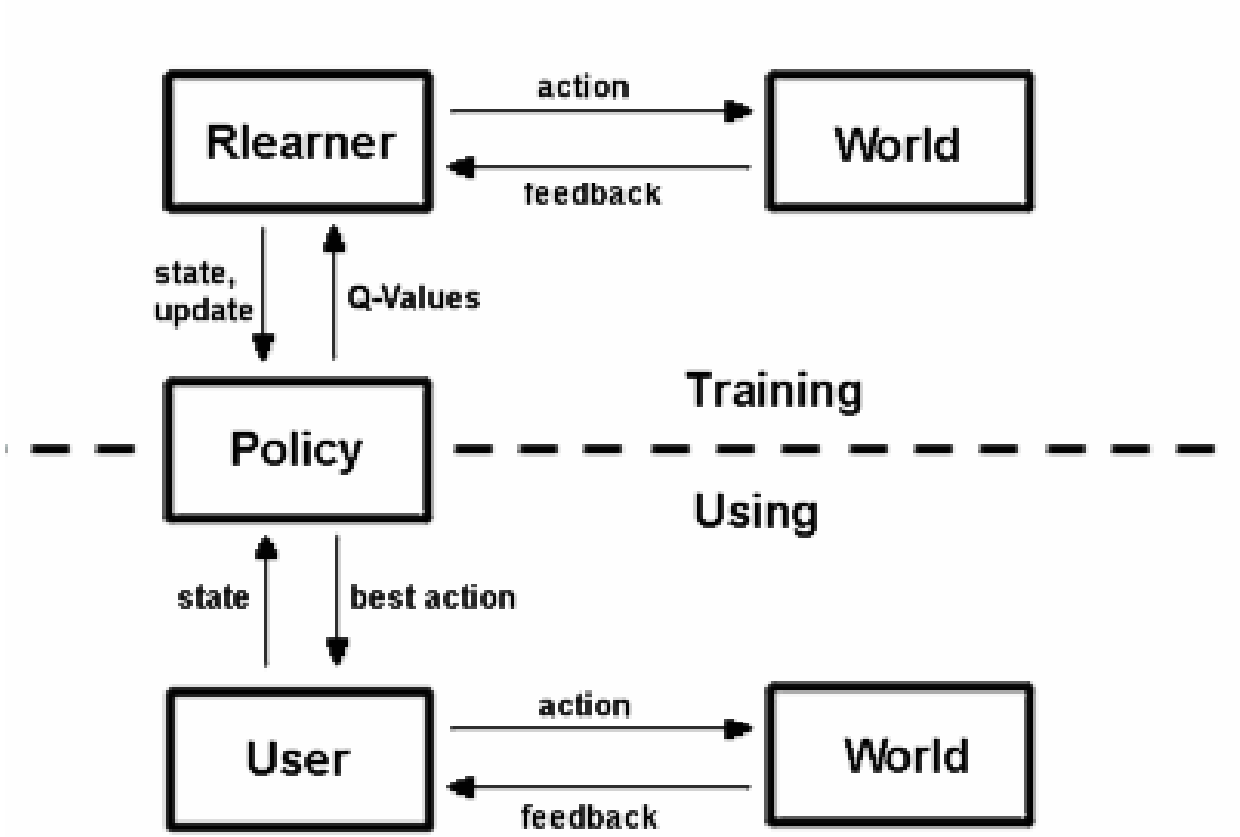
- **Función de refuerzo R(s, a)**: Indica si ***una acción*** realizada fue buena o mala estableciendo el valor de la recompensa en función del estado del entorno y la acción realizada sobre él. Puede ser determinista o estocástica.

- **Función de Acción-Valor Q(s, a):** El objetivo del aprendizaje por refuerzo es maximizar la recompensa total obtenida a largo plazo. Esta función refleja ***una estimación de la recompensa a largo plazo*** que se va a recibir partiendo de un cierto estado **s**, ejecutando la acción **a** y siguiendo una cierta política. El objetivo de los algoritmos de aprendizaje por refuerzo es construir esta función.

- **Modelo del entorno**: permite saber cuál será el próximo estado (o cuál es la probabilidad de ser, si es estocástico), si se realiza una acción concreta en el estado actual. ***Permite predecir el comportamiento del entorno y aprovechar esta información para resolver el problema***.
  - Cuando se conoce el modelo, se acelera el aprendizaje.
  - Si no, **el agente debe ir descubriéndolo a medida que cursa su aprendizaje, explorando el espacio de políticas sin saber de antemano cuán bueno o malo es un estado.**

# APRENDIZAJE POR REFUERZO

## Elementos del aprendizaje por refuerzo



# APRENDIZAJE POR REFUERZO

## Aplicaciones del aprendizaje por refuerzo

- Dado que los agentes puede aprender sin la supervisión de expertos, el tipo de problemas que se adaptan mejor al AR son problemas complejos en los que no parece haber ninguna solución clara, ni una solución fácilmente programable.
- Dos de las principales aplicaciones son:
  - **Jugar a Juegos** - determinar el mejor movimiento en un juego a menudo depende de una serie de factores diferentes, de ahí que el nº de estados posibles que pueden existir en un determinado juego suele ser muy grande. Para cubrir tantos estados utilizando un enfoque basado en normas estándar, significaría también especificar un nº grande de reglas codificadas. AR reduce la necesidad de especificar manualmente las reglas, ***los agentes aprenden jugando al juego***. Los agentes pueden ser entrenados mientras juegan contra otros jugadores humanos o incluso otros agentes de AR.
  - **Problemas de control** - como la programación de un ascensor. Una vez más, no está claro cuáles son las estrategias que proporcionan el mejor servicio de ascensor. Para los problemas de control de este tipo, agentes pueden aprender en un entorno simulado y con el tiempo se van planteando buenas políticas de control.
    - Algunas de las ventajas de la utilización de AR para los problemas de control es que un agente se pueda formar fácilmente para adaptarse a los cambios del entorno, y entrena continuamente mientras el sistema está on-line, mejorando el rendimiento todo el tiempo.

# APRENDIZAJE POR REFUERZO

## Aplicaciones del aprendizaje por refuerzo

- Un jugador de damas, ajedrez: aprender la mejor secuencia de movimientos para ganar.
- Robots que deben seguir cierta formación.
  - Robots móviles: aprendizaje de la forma de escapar de un laberinto.
  - Brazo robot: aprendizaje de la secuencia a aplicar a las articulaciones para conseguir un cierto movimiento.
  - Un robot que tiene que cargar su batería.
  - Exploración de planetas
  - Limpieza de sitios peligrosos
  - Reconocimiento de terreno en sitios difíciles o bajo el agua
- Cadena de producción en industria.
- La compra de acciones de un inversor. Evaluación de productos financieros.
- La selección de un nodo en una red para direccionar un paquete.
- Optimización en controladores de memoria.
- Agentes autónomos en oficinas, industrias o casas.

# APRENDIZAJE POR REFUERZO

## Procesos de Decisión de Markov

- **Aprendizaje por refuerzo:** aprender a decidir mediante prueba y error, ante una situación determinada, qué acción es la más adecuada para lograr un objetivo.
  - Se basa en experiencias del tipo "*para el estado S la acción A del aprendiz/agente ha producido la recompensa R*".
  - No se sabe si la acción tomada es la mejor posible, requiere explorar las diferentes acciones para aprender cual es la mejor
  - **El objetivo** es encontrar la **política** de acción óptima que permita seleccionar en cada estado la acción que maximice en el futuro el refuerzo (Ej. Ganar a las damas)

*Esquema general del aprendizaje por refuerzo*


$S = \{ s_0, s_1, \ldots \}$ es el conjunto de estados del entorno

$A = \{ a_0, a_1, \ldots \}$ es el conjunto de acciones que puede realizar el aprendiz

$r_i$ es la recompensa que recibe el aprendiz tras haber realizado la acción $a_i$ en el estado $s_i$

$$s_0 \xrightarrow[r_0]{a_0} s_1 \xrightarrow[r_1]{a_1} s_2 \cdots\cdots$$

# APRENDIZAJE POR REFUERZO

## Procesos de Decisión de Markov

- La mayoría de los algoritmos de aprendizaje por refuerzo están sujetos a que la señal de estado satisfaga la propiedad de Markov.
- **Propiedad de Markov**: consiste en que la evolución del entorno dependa exclusivamente de su estado y de la acción realizada.
  - Es decir, la evolución del entorno no depende de los estados anteriores, ni de las acciones anteriores. No importa cómo llegó a ese estado.
  - Ese estado contiene la información completa de las percepciones pasadas de una manera compacta.
- **Proceso de Decisión de Markov (MDP)**: problema de aprendizaje por refuerzo en el que el entorno cumple la propiedad de Markov.
- **Proceso de Decisión de Markov finito**: es un MDP en el que el espacio de estados y de acciones es finito.
  - En un MDP finito, se conocen los conjuntos de estados y acciones y las probabilidades que definen su dinámica.
- Existen problemas de observación parcial, ya que en determinadas circunstancias no es posible observar toda la información relevante para tomar decisiones óptimas, o que la señal de estado contenga toda la información necesaria para cumplir la propiedad de Markov.

# APRENDIZAJE POR REFUERZO

## Procesos de Decisión de Markov

- Un Proceso de Decisión de Markov (MDP) se define mediante la tupla <S,A,T,R> donde:
  - El entorno se puede encontrar en un conjunto de posibles estados **S**
  - Existe una función de transición **T(s, a)** entre ellos (normalmente desconocida y no determinista), de forma que considerando un estado **s** y la realización en ese estado de una acción **a**, les asocia el estado siguiente, denominado **s'**
  - Una función de refuerzo **R(s, a)** (también desconocida normalmente) que el entorno puede devolver al realizar la acción **a** en el estado **s**.

- OBJETIVO: encontrar una política de acción óptima que permita seleccionar en cada estado aquella acción que maximice en el futuro el refuerzo (p.e. ganar en el caso de las damas)

- Se cumple la propiedad de Markov: *la política (qué acción realizar en cada estado) se basa solo en el estado actual y no en la historia previa (el estado actual ya tiene toda la información necesaria)*

# APRENDIZAJE POR REFUERZO

## Taxonomía de los métodos de aprendizaje por refuerzo

- Los métodos de resolución en un problema de aprendizaje por refuerzo (considerando que sea un Proceso de Decisión de Markov) se pueden clasificar en función del conocimiento que se tenga del modelo:
- **Resolución a partir de un conocimiento completo**
  - Se conocen el conjunto de estados y acciones, la función de transición de estados y la función de refuerzo.
  - La resolución se basa en que el agente aprenda el modelo conocido según la información de las funciones existentes y luego aplique la mejor política de acciones.
- **Resolución a partir de un conocimiento incompleto**
  - No se conoce el modelo del entorno (es decir, los efectos de las acciones sobre el entorno, no se conoce la función de transición T ni de refuerzo R). Es lo más habitual.
    - Técnicas como: **Q-Learning, SARSA, Dyna-Q, etc.**
  - Ventaja de poder ser empleados de manera incremental y permitir que el agente aprenda mientras se encuentra on-line.
  - Además, utilizan algoritmos iterativos muy sencillos, cosa que favorece su implantación.

# APRENDIZAJE POR REFUERZO

## Resolución a partir de un conocimiento incompleto

- Objetivo del aprendizaje: inferir la función Acción-Valor óptima **Q(s, a)** s=estado, a=acción.

- **Q(s, a) tabla/matriz** que contiene para cada estado **s** y acción **a**, el refuerzo esperado en el tiempo por aplicación de la acción **a** desde el estado **s**:
  - La acción a elegir será la que maximice el esfuerzo esperado que aparecerá retardado en el tiempo.
  - La tabla se irá construyendo progresivamente, aprendiendo el esfuerzo esperado.
- Conocido **Q(s, a)**, el sistema sabrá qué acción tiene que ejecutar en cada estado.
- El agente necesita una **política de selección de acciones** a ejecutar en un determinado estado **s.**
  - Dado que el comportamiento del entorno es desconocido, el aprendizaje por refuerzo conlleva una fuerte carga de ensayo y error.
  - Obtener el ***balance exploración-explotación***. Se trata de evaluar si es mejor **explorar** el entorno para mejorar el conocimiento del problema (a costa de empeorar a corto plazo la recompensa obtenida) o **explotar** el conocimiento acumulado (intentando maximizar la recompensa).

# APRENDIZAJE POR REFUERZO

## Exploración y Explotación

- Problema interesante que surgen al utilizar el aprendizaje por refuerzo: equilibrio entre la ***exploración vs explotación***.
  - Si un agente ha tratado una acción determinada en el pasado y recibió una recompensa decente, repitiendo esta acción va a reproducir la recompensa. De este modo, el agente está explotando lo que conoce para recibir una recompensa.
  - Por otro lado, explorar el entorno intentando otras posibilidades puede producir una mejor recompensa, por lo que es una buena táctica a veces.
  - Sin un equilibrio de la exploración y explotación del agente AR no va a aprender con éxito.
  - La manera más común para lograr un buen equilibrio es intentar una variedad de acciones, mientras que progresivamente se favorecen aquellas que destacan por producir la mayor recompensa.

# APRENDIZAJE POR REFUERZO

## Exploración y Explotación

- Existen estrategias que se utilizan para la selección de acciones: el objetivo de estas estrategias es equilibrar la explotación y la exploración, al no siempre explotar lo que se ha aprendido hasta el momento actual.

- No está claro qué estrategia produce los mejores resultados, depende de la naturaleza de la tarea. Si el problema es un juego en el que una máquina juega contra un oponente humano, los factores humanos pueden ser también influyentes.
  - $\epsilon$ -greedy – el agente selecciona la mayoría de las veces la acción con la mejor recompensa estimada (explotación), y alguna vez, según cierta probabilidad $\epsilon$, se elige una acción aleatoria cualquiera (exploración). Suelen descubrirse acciones óptimas.
  - softmax – impide que se seleccionen las peores acciones en el paso de elección aleatoria, al otorgar un peso a las acciones según su refuerzo estimado. Se eligen semi-aleatoriamente, teniendo en cuenta su peso asociado para que no sean las peores.

# APRENDIZAJE POR REFUERZO

## Resolución a partir de un conocimiento incompleto

¿Cómo inferir Q?

1. Inicialmente Q(s, a) toma valores aleatorios (a cada estado y a cada acción posible en ese estado se les asigna un valor Q aleatorio).
2. Partiendo del estado inicial, se realizan acciones aleatoriamente hasta que se cumple una de estas condiciones:
   - El sistema alcanza el objetivo buscado.
   - El nº de acciones realizadas alcanza un límite máximo.
3. Se actualizan los valores de Q en función del resultado obtenido:
   - Si se ha alcanzado el objetivo, se recompensan las acciones proporcionalmente a la distancia al objetivo final.
   - Si se ha realizado el número límite de acciones sin alcanzar el objetivo final, no hay recompensa.

- Las etapas 2 y 3 se repiten hasta cumplirse una de estas dos condiciones:
  - Q converge (sin cambios significativos).
  - Se alcanza un número máximo de repeticiones o episodios.
- ***Episodio*:** secuencia de acciones-estados completa, es decir, desde el inicio hasta un estado objetivo o sumidero. El aprendizaje se divide en episodios de forma natural (p.ej. partidas de un juego, secuencias de movimientos hasta llegar a una meta, etc.)
- Normalmente se requiere un nº de episodios elevado para obtener una buena Q.

# APRENDIZAJE POR REFUERZO

## Resolución a partir de un conocimiento incompleto: Q-Learning

Inicialización: Q(s,a) arbitrariamente
Método:
Repetir (para cada episodio)

Inicializar s
Repetir para cada paso del episodio

*Estrategia de selección de acciones: exploración vs explotación*

Seleccionar una acción *a* a partir de s utilizando la política derivada de Q
Ejecutar la acción *a*. Observar el refuerzo recibido r y el estado siguiente s'
Actualizar la tabla Q(s,a)

$$Q(s,a) \leftarrow Q(s,a) + \alpha[r + \gamma \max_a Q(s',a') - Q(s,a)]$$

$s \leftarrow s'$

hasta que s' sea un estado terminal

($\alpha$ es un parámetro que pondera las actualizaciones con respecto a los valores anteriores.
Para dominios deterministas $\alpha$=1)

Algoritmo de Aprendizaje-Q

# APRENDIZAJE POR REFUERZO

## Resolución a partir de un conocimiento incompleto

- ¿Que pasa si el entorno no es determinista?
  - No se puede actualizar Q(s,a) con *r + max(Q(s', a'))*, por que sería como confiar que ejecutando la acción **a** en el estado **s** siempre se transita en el estado **s'**
  - Hay que actualizar Q(s,a) con una parte de esta información nueva que se acaba de descubrir, considerar que *(a'= b)*:

$$Q(s, a) \leftarrow Q(s, a) + \alpha \cdot [\, r + \gamma \cdot \max_{b} Q(s', b) - Q(s, a) \,]$$

estimación actual — vieja estimación — nueva información — vieja estimación

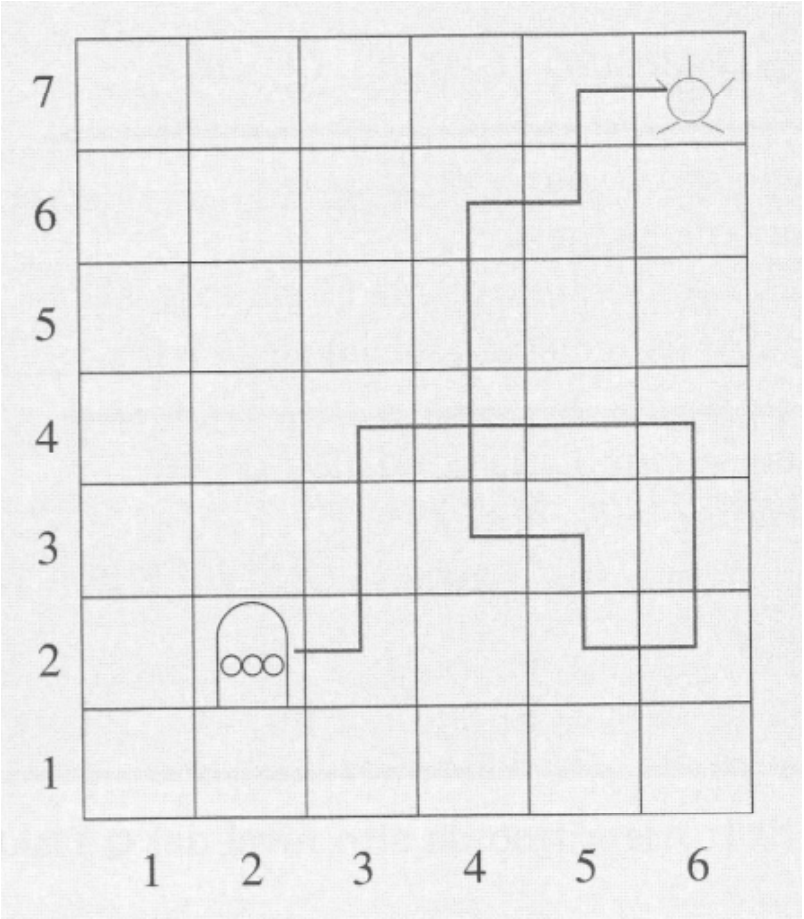
  - **alfa** representa la porción de la diferencia entre la nueva información y la vieja estimación que se añade a la vieja estimación
- Se irá construyendo progresivamente una tabla Q en el aprendizaje.
- Q-learning garantiza la convergencia de *Q(s, a) a Q*(s, a) (tabla óptima)*
  - Si el entorno se puede modelar como un MDP (Proceso de Decisión de Markov)
  - El entorno es estacionario (es decir, la probabilidad de transitar de **s** a **s'** ejecutando la acción ***a***, no varía en el tiempo)

# APRENDIZAJE POR REFUERZO

## Ejemplo Q-Learning

- Recorrido de un robot hacia la meta:
  - Cumple MDP
  - Determinista: alfa = 1
  - Se crea la tabla Q mediante prueba y error.

La ***condición de fin*** del algoritmo suele ser ejecutar N ciclos o comprobar si no ha habido cambios en la tabla Q entre 2 iteraciones consecutivas.



# APRENDIZAJE POR REFUERZO

## Ejemplo Q-Learning

- Robot que se mueve en un entorno bidimensional y que debe aprender a llegar a una posición determinada (x, y) desde cualquier posición inicial del entorno.
  - 4 movimientos (acciones): hacia arriba (Ar), hacia abajo (Ab), izquierda (I), derecha (D).
  - No se conoce "a priori" la consecuencia de cada acción.
  - Al robot se le proporcionará un refuerzo de 1 cuando llegue a la meta y un 0 en todos los demás casos.
- El refuerzo o "clase" (éxito o fracaso en llegar a la meta) lo recibe retardado en el tiempo (después de haberse movido semi-aleatoriamente por el entorno durante un tiempo -> es un problema de AR

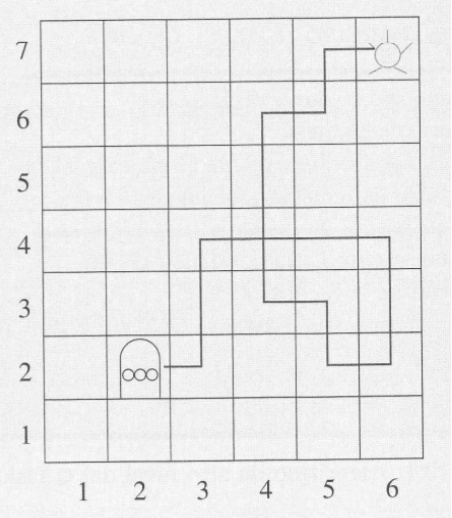
Función Q-LEARNING $(S, A, E)$: $Q(s, a)$

---

$S$: conjunto de estados o situaciones
$A$: conjunto de acciones
$E$: conjunto de instancias de la forma $(s, a, s', r)$
$Q(s, a)$: tabla de refuerzos, inicialmente a 0

---

Hasta que se cumpla la condición de fin
  Para cada $e = (s, a, s', r) \in E$
    $Q(s, a) := \alpha[r + \gamma \text{máx}_{b \in A} Q(s', b)] + (1 - \alpha)Q(s, a)$
Devolver $Q$



# APRENDIZAJE POR REFUERZO

## Ejemplo Q-Learning

- Lo primero es fijar una estrategia de explotación-exploración, decidir cómo obtener las tuplas (instancias) de entrenamiento (s, a, s', r).
  - Una forma sencilla consiste en empezar ejecutando acciones seleccionadas aleatoriamente hasta que llegue al objetivo (zona de meta) o no llegue (después de k acciones). A medida que se crea la Q, se puede ir explotando alguna acción "buena".
- Si el robot ha realizado las acciones de la figura, el conjunto de tuplas de entrenamiento generadas serían las que aparecen en la tabla.

| $s$ | $a$ | $s'$ | $r$ |
|---|---|---|---|
| (2,2) | D | (3,2) | 0 |
| (3,2) | Ar | (3,3) | 0 |
| (3,3) | Ar | (3,4) | 0 |
| (3,4) | D | (4,4) | 0 |
| (4,4) | Ab | (4,3) | 0 |
| (4,3) | D | (5,3) | 0 |
| (5,3) | Ab | (5,2) | 0 |
| (5,2) | D | (6,2) | 0 |
| (6,2) | Ar | (6,3) | 0 |
| (6,3) | Ar | (6,4) | 0 |
| (6,4) | I | (5,4) | 0 |
| (5,4) | I | (4,4) | 0 |
| (4,4) | Ar | (4,5) | 0 |
| (4,5) | Ar | (4,6) | 0 |
| (4,6) | D | (5,6) | 0 |
| (5,6) | Ar | (5,7) | 0 |
| (5,7) | D | (6,7) | 1 |

  - En la inicialización, se hace $Q_0(s, a) = 0$ para cada par (s, a).
  - Si suponemos que $\gamma = 0{,}8$ y $\alpha = 1$ (caso determinista), en el primer ciclo se harían las siguientes actualizaciones:

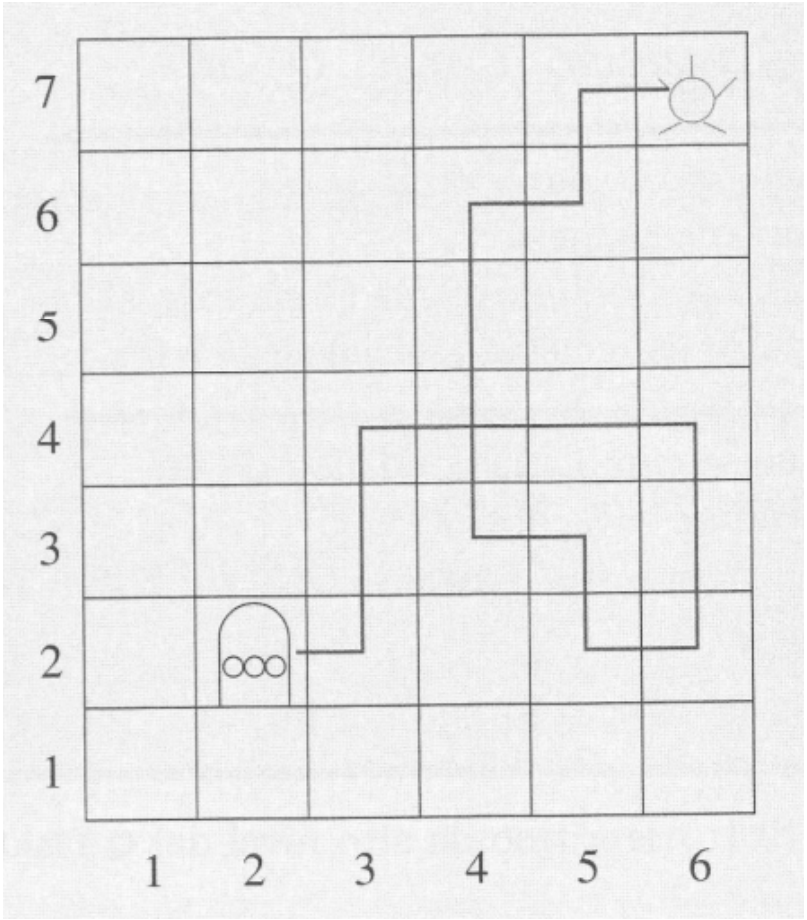
$$Q(s,a) \leftarrow Q(s,a) + \alpha[r + \gamma \max_{a'} Q(s',a') - Q(s,a)]$$

    - $Q_1((2,2), D) = r((2, 2), D) + 0{,}8 \max_{a'} Q_0((3,2), a') = 0 + 0{,}8 \times 0 = 0$
  - Para cada una de las tuplas, salvo para la última, sucederá lo mismo, porque todos los refuerzos inmediatos son 0 y las $Q_0(s, a)$ son 0.
  - En el caso de la última tupla:
    - $Q_1((5, 7), D) = r((5,7), D) + 0{,}8 \max_{a'} Q_0((6,7), a') = 1 + 0{,}8 \times 0 = 1$

# Aprendizaje Por Refuerzo

## Ejemplo Q-Learning

| $s$ | $a$ | $s'$ | $r$ |
|---|---|---|---|
| (2,2) | D | (3,2) | 0 |
| (3,2) | Ar | (3,3) | 0 |
| (3,3) | Ar | (3,4) | 0 |
| (3,4) | D | (4,4) | 0 |
| (4,4) | Ab | (4,3) | 0 |
| (4,3) | D | (5,3) | 0 |
| (5,3) | Ab | (5,2) | 0 |
| (5,2) | D | (6,2) | 0 |
| (6,2) | Ar | (6,3) | 0 |
| (6,3) | Ar | (6,4) | 0 |
| (6,4) | I | (5,4) | 0 |
| (5,4) | I | (4,4) | 0 |
| (4,4) | Ar | (4,5) | 0 |
| (4,5) | Ar | (4,6) | 0 |
| (4,6) | D | (5,6) | 0 |
| (5,6) | Ar | (5,7) | 0 |
| (5,7) | D | (6,7) | 1 |



# APRENDIZAJE POR REFUERZO

## Ejemplo Q-Learning

- En el siguiente ciclo, volverán a ser 0 todas las actualizaciones (los refuerzos inmediatos son 0, así como los $Q_1(s, a)$), salvo el caso de la penúltima tupla que será:

  $Q_2(5, 6), Ar) = r((5, 6), Ar) + 0{,}8 \max_{a'} Q_1((5,7), a') = 0+0{,}8 \times 1 = 0{,}8$

  - Además,

  $Q_2(5,7), D) = r((5,7), D) + 0{,}8 \max_{a'} Q_1((6, 7), a') = 1 + 0{,}8 \times 0 = 1$

  - es decir, se conserva el valor de Q((5,7), D). Como se puede comprobar, en las siguientes iteraciones se irían propagando los valores de refuerzo a las casillas previas, descontadas por el valor de $\gamma$ =0,8. Por ejemplo, en el siguiente ciclo (3), le tocaría el turno de actualización a la siguiente tupla:

  $Q_3(4, 6), Ar) = r((4, 6), D) + 0{,}8 \max_{a'} Q_2((5, 6), a') = 0 + 0{,}8 \times 0{,}8 = 0{,}64$

  - Al final de k ciclos (que, en este caso, sería igual al n° acciones en esta ejecución), la tabla Q(s, a) no cambiaría entre dos ciclos consecutivas, con lo que se podría terminar el procedimiento.
  - La salida sería la tabla Q (s, a) generada.

| $s$ | $a$ | $s'$ | $r$ |
|---|---|---|---|
| (2,2) | D | (3,2) | 0 |
| (3,2) | Ar | (3,3) | 0 |
| (3,3) | Ar | (3,4) | 0 |
| (3,4) | D | (4,4) | 0 |
| (4,4) | Ab | (4,3) | 0 |
| (4,3) | D | (5,3) | 0 |
| (5,3) | Ab | (5,2) | 0 |
| (5,2) | D | (6,2) | 0 |
| (6,2) | Ar | (6,3) | 0 |
| (6,3) | Ar | (6,4) | 0 |
| (6,4) | I | (5,4) | 0 |
| (5,4) | I | (4,4) | 0 |
| (4,4) | Ar | (4,5) | 0 |
| (4,5) | Ar | (4,6) | 0 |
| (4,6) | D | (5,6) | 0 |
| (5,6) | Ar | (5,7) | 0 |
| (5,7) | D | (6,7) | 1 |

# APRENDIZAJE POR REFUERZO

## Ejemplo Q-Learning

- Si se tuvieran más tuplas de entrenamiento provenientes de otras ejecuciones-episodios del robot (volverlo a poner en otra posición inicial e intentar alcanzar la meta), se añadirían a las tuplas anteriores, y se tratarían todas en cada ciclo.
- En el caso no determinista, habría que decidir el valor de alfa y utilizar la ecuación completa.
- Se asume que el n° de estados y acciones es finito y no muy grandes.
- Si no se cumple esta restricción, la tabla Q podría hacerse incluso infinita, por lo que habría que discretizar tanto los estados (espacio de estados) como las acciones:
  - Discretizar mediante otros métodos (RNA, etc.).
- El cálculo de la tabla/matriz de refuerzos es proporcional al n° de instancias de entrenamiento y al n° de ciclos que se desee ejecutar.

# APRENDIZAJE POR REFUERZO

## Ejemplo Q-Learning

- Utilización de lo aprendido: cuando está la tabla Q construida y el sistema se encuentra en un estado s, el sistema tendría que realizar aquella acción que maximice el esfuerzo esperado en el tiempo.
- Como eso es lo que calcula la posición Q (s, a) para cada **a** perteneciente a A, habrá que seleccionar la acción **a** de acuerdo a la política π (s) tal que:

$$\pi(s) = \arg\max_{a} [Q(s, a)]$$

- Es decir, dada una situación en la que se pueden tomar acciones diferentes, se elige aquella acción con la que se espera obtener un máximo refuerzo en el futuro. Ej. En la casilla (5, 7) dado que la fila con D tiene mayor refuerzo, se elegirá la acción mover a la derecha.

- Permite resolver eficientemente la adquisición de conocimiento para la toma de decisiones cuando no se dispone de un modelo del entorno, éste es no determinista y el refuerzo llega retardado en el tiempo.
  - Juegos, robots.

# Aprendizaje Por Refuerzo

## Ejemplo Q-Learning

- El aprendizaje puede ocurrir tanto de manera online como de manera offline.
  - A ser posible, aprender la política óptima de manera offline y aplicarla en el problema real

# APRENDIZAJE POR REFUERZO

## Ejemplo: Applet Gato y Ratón

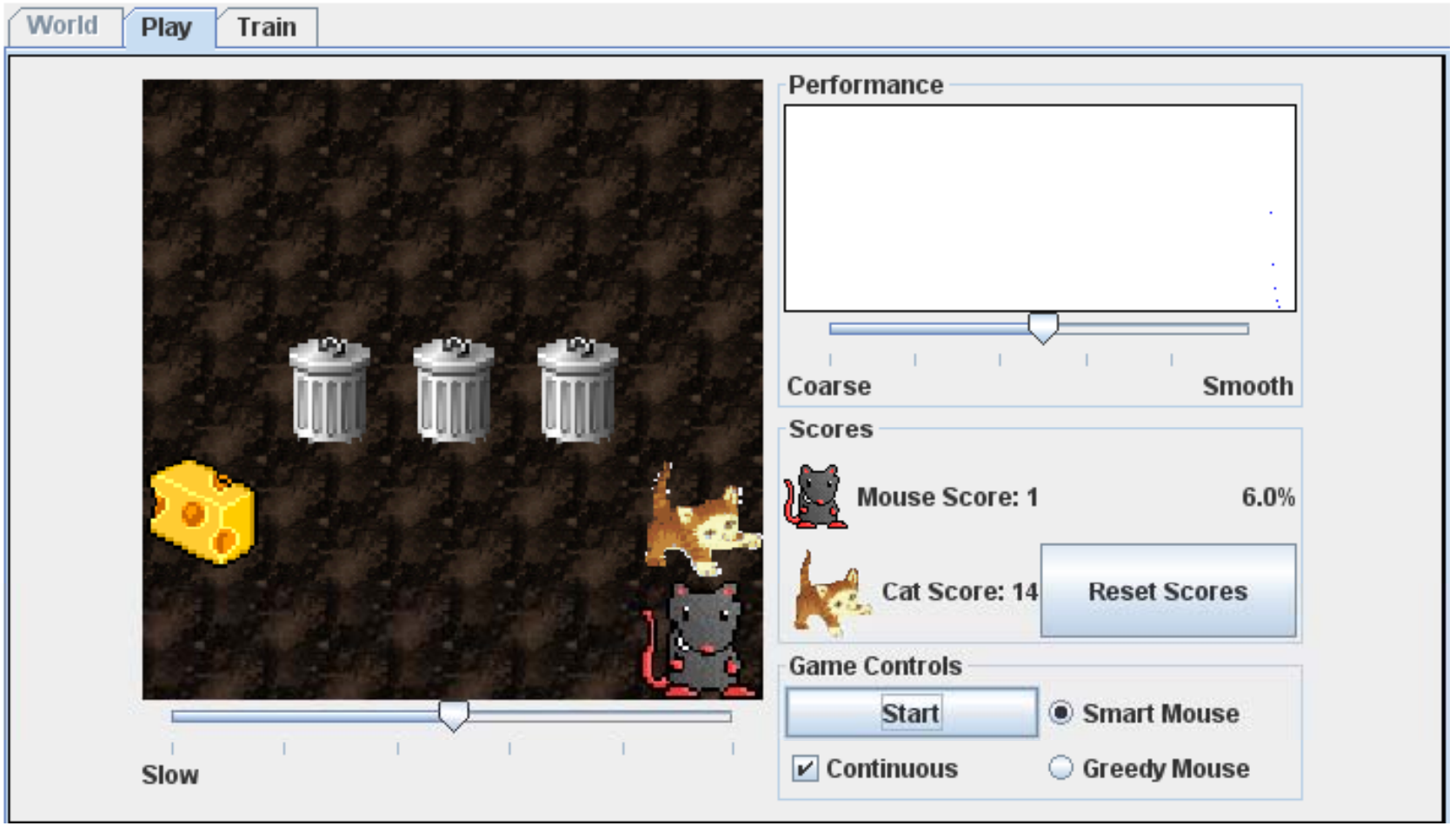
- APPLET: http://www.cse.unsw.edu.au/~cs9417ml/RL1/applet.html

- En este applet, el entrenamiento y la ejecución (utilizando la información de entrenamiento) se han separado.
- En algunas situaciones de aprendizaje por refuerzo, entrenamiento y ejecución se mezclan, por lo que el agente aprende de su experiencia en el entorno.
- Debido a la naturaleza del problema y para explicar el algoritmo, se han separado las dos fases.
  - Hay un gato, un ratón, un trozo de queso, así como algunos obstáculos en el entorno del gato y el ratón.
  - El ratón trata de evitar ser atrapado por el gato, al mismo tiempo, tratando de llegar al queso para comer.
  - El ratón es el único que aprende en el applet, el gato ya está programado para ir siempre a por el ratón.

# APRENDIZAJE POR REFUERZO

## Ejemplo: Applet Gato y Ratón

- APPLET: http://www.cse.unsw.edu.au/~cs9417ml/RL1/applet.html



# Aprendizaje Por Refuerzo

## Ejemplo: Applet Gato y Ratón

- Las reglas del juego del gato y del ratón son:
  - Tanto el gato como el ratón tienen 8 grados de movimiento: arriba, abajo, izquierda y derecha, así como las cuatro diagonales.
  - El ratón anota un punto al conseguir el queso (cuando está en la misma celda que el queso).
  - El gato anota un punto al coger al ratón, al moverse a la misma plaza que el ratón.
  - Si el ratón consigue el queso, un nuevo trozo se coloca al azar, mientras que el gato y el ratón mantienen sus posiciones.
  - El juego termina cuando el gato caza al ratón. Los marcadores se actualizan y un nuevo juego puede comenzar.
- Cuando el ratón se entrena mediante el algoritmo de aprendizaje por refuerzo, se puede apreciar que ha aprendido a usar la pared como una barricada y valora su vida por encima de conseguir un pedazo de queso.
  - Ocasionalmente, sin embargo, el ratón queda atrapado por el gato.
- En función de los parámetros de aprendizaje (algoritmo, exploración, explotación, refuerzo del queso, etc.), se observa como mejora o empeora el marcador del ratón.

# APRENDIZAJE POR REFUERZO

## AR y Robots Autónomos

- Los ***robots autónomos*** son entidades físicas con capacidad de percepción sobre un entorno y que actúan sobre él en base a dichas percepciones, sin supervisión directa de otros agentes.
  - En ocasiones se asimila el término "robot autónomo" con el de "robot móvil", pero son términos diferentes.
  - Un robot autónomo suele ser móvil, entendiendo por móvil que no se encuentra fijado a una posición y puede desplazarse por su entorno, pero no hay nada que en principio obligue a ello.
  - A la inversa, un robot móvil no es necesariamente autónomo: existen multitud de robots móviles que son teledirigidos.
- Un robot autónomo percibe un entorno y actúa sobre él, está literalmente inmerso en el entorno, esto es, el robot no actúa sobre modelos, sino directamente sobre la realidad material.
- Los robots autónomos operan sobre el mundo físico, su experiencia del mundo y sus acciones sobre él se producen de forma directa haciendo uso de sus propias capacidades físicas.

# APRENDIZAJE POR REFUERZO

## AR y Robots Autónomos

- Importancia de poder proporcionar a los robots mecanismos de aprendizaje
  - Un robot sin capacidades de aprendizaje puede funcionar adecuadamente en un entorno controlado, pero al enfrentarse a un entorno cambiante, las acciones que antes eran adecuadas pueden convertirse en inútiles.
  - Si el robot tuviese la capacidad de calcular su propio consumo y aprender cuáles son las acciones que consiguen reducir dicho consumo al mínimo para cumplir un cierto objetivo, esto podría suponer grandes ahorros en tiempo y energía en la vida útil del robot.
- Estas capacidades pueden obtenerse con las técnicas de aprendizaje por refuerzo.
- El aprendizaje por refuerzo permitirá conseguir que un robot actúe en un entorno de manera que maximice la recompensa que obtiene por sus acciones.
- Que los entornos sean complejos y dinámicos, puede suponer que al efectuar una misma acción en el mismo estado en dos ocasiones distintas obtengamos consecuencias diferentes cada vez.
  - Tener en cuenta que en el entorno podría incluir a otros agentes.
  - En muchas ocasiones los agentes no tienen la capacidad de percibir completamente dicho entorno, y esto puede añadir dificultades a la hora de llevar a cabo ciertas tareas.

# APRENDIZAJE POR REFUERZO

## AR y Robots Autónomos

- Proyectos Ingeniería Informática:
  - Automatización de la obtención de comportamientos mediante aprendizaje por refuerzo "Q" en un robot Pioneer 2-AT.
  - Sistema de aprendizaje por refuerzo supervisado de un robot móvil en aplicaciones de control visual.
  - Búsqueda evolutiva y aprendizaje por refuerzo complementario para el control neuronal en robótica autónoma.

# Aprendizaje Por Refuerzo

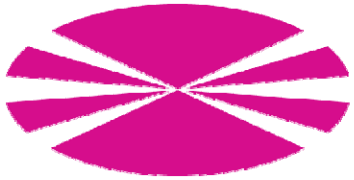
## Conclusiones

- El aprendizaje por refuerzo es un modelo de aprendizaje que permite implementar comportamientos inteligentes de forma automática, sin que el diseñador tenga que incorporar conocimiento o modelos del dominio en el que está trabajando.
- La mayor parte de la teoría del aprendizaje por refuerzo tiene su fundamento en la programación dinámica, y por tanto, en lo que se denominan funciones de valor.
  - Estas funciones dan información sobre lo valioso que es para el desarrollo de una tarea, el encontrarse en una determinada situación o estado, e incluso lo valioso que es ejecutar una determinada acción, dando por hecho que el sistema se encuentra en un determinado estado.
  - Estas funciones, generalmente implementadas como tablas, son utilizadas para obtener de forma directa la política de acción que debe guiar el comportamiento del sistema.
  - Sin embargo, la implementación tradicional de estas funciones en forma tabular no es práctica cuando el espacio de estados es muy grande, o incluso infinito. Cuando se produce esta situación, se deben aplicar métodos de generalización que permitan extrapolar la experiencia adquirida para un conjunto limitado de estados, a la totalidad del espacio, de forma que se consigan comportamientos óptimos en cualquier situación, aunque ésta no hubiera sido explorada con anterioridad.

# Aprendizaje Por Refuerzo

## Conclusiones

- Es interactivo (recibe información del entorno y puede modificar el mismo)
  - El comportamiento del entorno es, en general, desconocido y puede ser estocástico, es decir, que la evolución del entorno y la recompensa generada pueden obedecer a una cierta función de probabilidad.
- Es dirigido por objetivos (el fin del aprendizaje es alcanzar un máximo)
  - Está dirigido por objetivos. Este objetivo se expresa por una recompensa que devuelve el entorno al realizar una acción sobre él. No se conoce cual es la salida adecuada para el sistema. Tan solo que el efecto que debe producir esta salida sobre el entorno sea tal que se maximice la recompensa recibida a largo plazo.
- El aprendiz debe explotar las acciones que le beneficien y explorar nuevas acciones
  - La recompensa puede tener un cierto retardo. Es decir, la bondad de una acción tomada por el sistema puede que no se refleje hasta un cierto número de evaluaciones posteriores.



UNIVERSIDADE DA CORUÑA

# APRENDIZAJE AUTOMÁTICO

## TEMA 4:

## APRENDIZAJE POR REFUERZO