APRENDIZAJE

Jairo Alfaro / María Fernanda Jiménez / Mauricio Pandolfi / Cristian Vargas

¿Qué es aprendizaje ?

Aprendizaje denota cambios en un sistema que permite al sistema hacer la misma tarea más eficientemente la próxima vez.

– Herbert Simon

Aprendizaje es hacer cambios útiles en nuestra mente.

– Marvin Minsky

Existen dos maneras en la que el sistema puede mejorar:

- Adquirir nuevo conocimiento
- Adaptar su comportamiento

¿Por qué el aprendizaje es necesario?

Un sistema que no puede aprender, no es inteligente.

I. Sin aprendizaje, todo es nuevo, un sistema que no pueda aprender no es eficiente porque siempre comete los mismos errores.
II. Para descubrir nuevas estructuras que son desconocidas para los humanos, como Data Mining.
III. Para completar especificaciones incompletas acerca de un dominio. Sistemas complejos de AI no pueden ser derivados a mano completamente, necesitan actualizaciones dinámicas para incorporar nueva información.

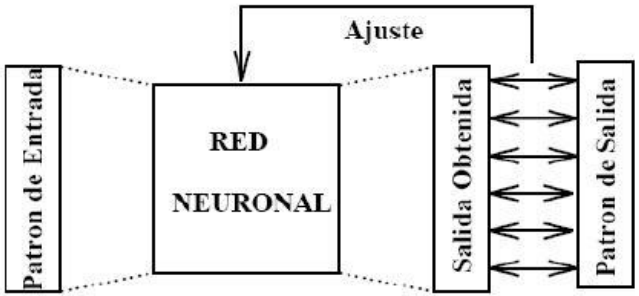
¿Por qué el aprendizaje es posible? Porque hay regularidades en el mundo.

Paradigmas de Aprendizaje

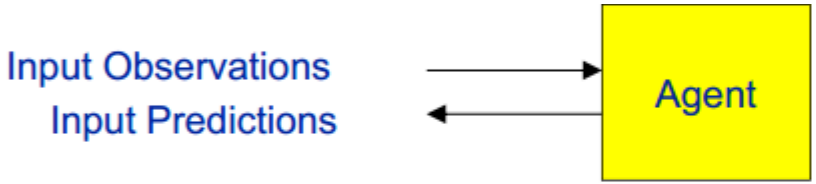
- *Aprendizaje de ruta:* Mapeo uno a uno de una representación guardada.
- *Inducción:* Usa ejemplos específicos para alcanzar conclusiones generales.
- *Refuerzo:* Retroalimentación (positiva y negativa) al final de una serie de pasos.
- *Clustering:* Correspondencia determinada entre dos representaciones diferentes.
- *Algoritmos genéticos.*
- *Descubrimiento:* Metas no supervisadas ni específicas.

Paradigmas de Aprendizaje (cont.)

Aprendizaje supervisado: Para cada entrada hay una salida esperada. El agente genera una salida basada en las observaciones de entrada. También recibe retroalimentación del ambiente para decirle cómo debe responder.



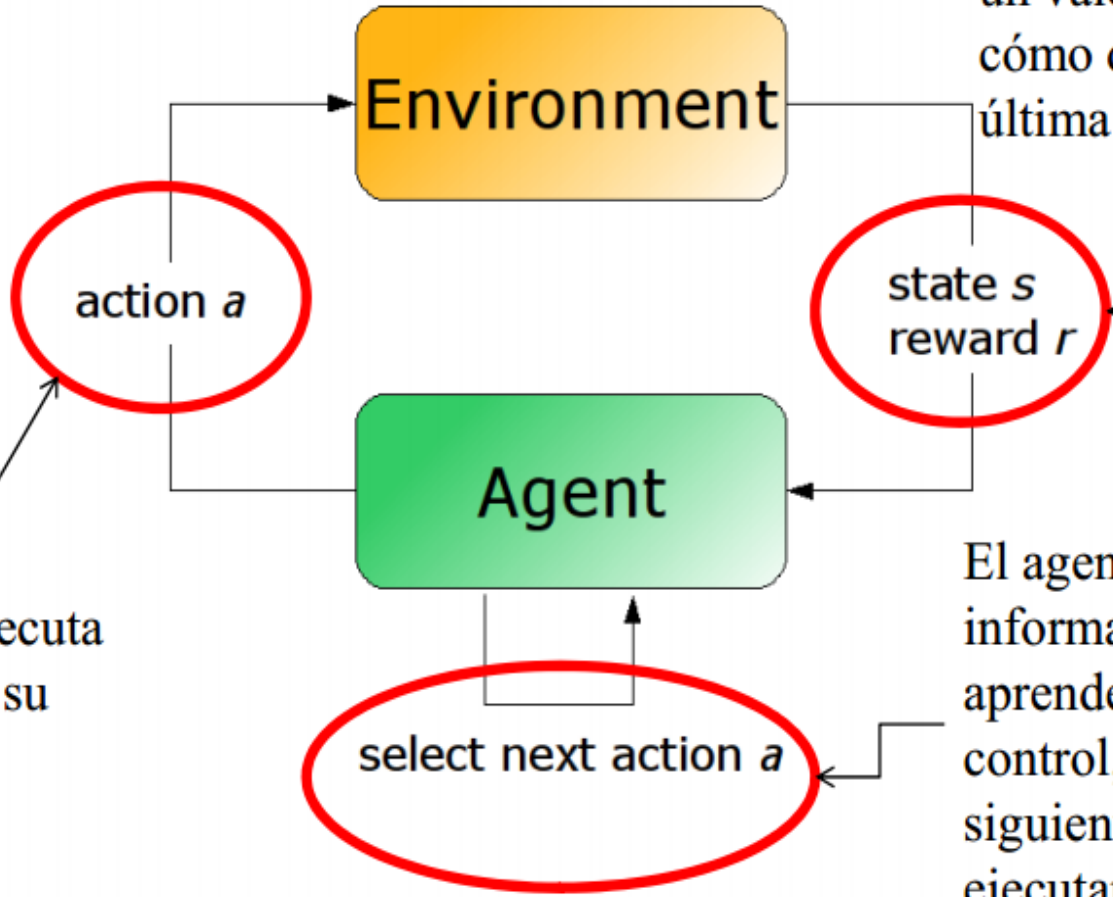
Aprendizaje no Supervisado: Sólo hay entradas. El ambiente no provee retroalimentación. Todo lo que el agente puede hacer es modelar la información de entrada, puede desarrollar representaciones internas eficientes de la entrada.



Paradigmas de Aprendizaje (cont.)

- *Aprendizaje por refuerzo*: Para cada entrada, el agente debe escoger una acción. También recibe retroalimentación del ambiente acerca de qué tan buena fue la respuesta, pero no de lo que tuvo que haber hecho. Se utiliza para mejorar indirectamente los parámetros del agente, por lo que el agente puede aprender un mapeo entre los datos de entrada y las acciones. Sólo hay recompensas después de una larga serie de acciones.

El entorno transita en un nuevo estado y devuelve un valor que define cómo de buena fue la última acción



El agente ejecuta acciones en su entorno

El agente utiliza esta información para aprender la política de control, que determina la siguiente acción a ejecutar

Proceso de Decisión de Markov

El estado actual y la acción nos da toda la información posible respecto al próximo estado al que se llegará, independientemente de cómo se llegó al estado actual.

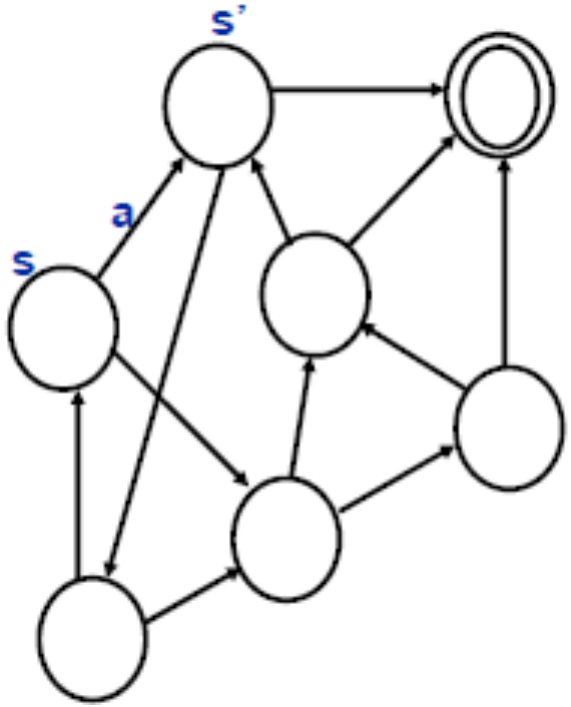
Proceso de Decisión de Markov (cont.)

S: Conjunto finito de estados.

A: Conjunto finito de acciones

T(s, a, s'): probabilidad de ir del estado s al estado s' con la acción a.

R(s, a, s'): recompensa por realizar la transición del estado s al estado s' utilizando la acción a.



Función de valor

- Una función de valor representa el estimado de las sumas esperadas de recompensas futuras.

- Una función Q representa el valor por seleccionar una acción estando en un estado determinado.

Proceso de Decisión de Markov (cont.)

- Si se conocen T y R, se llama basado en modelo y se debe usar programación dinámica (offline)

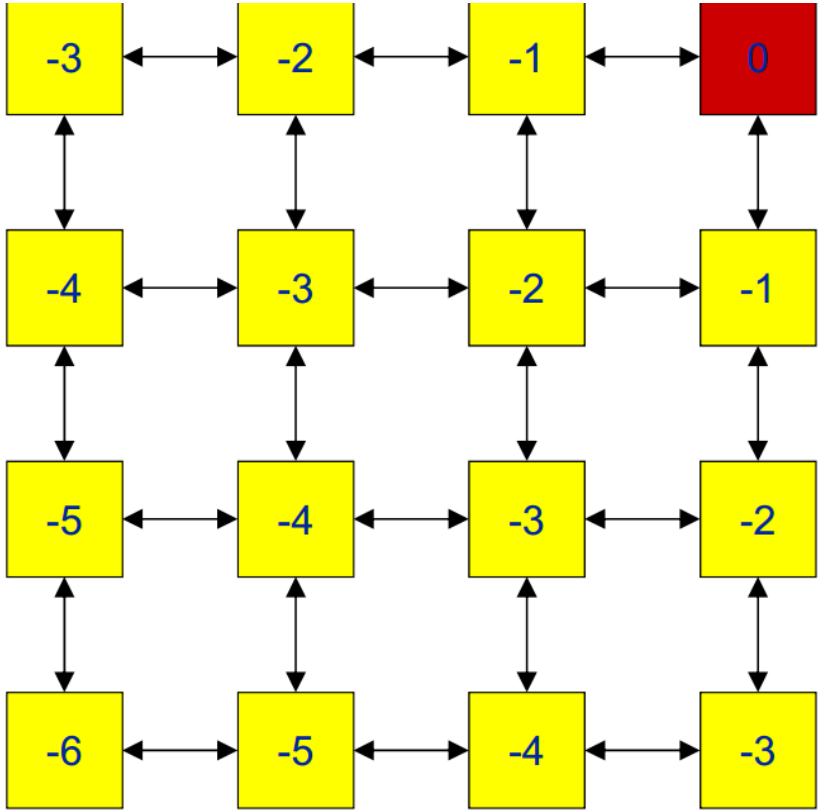
- Si no se conocen T y R, se llama libre de modelo y hay que interactuar con el entorno para aprender de ejemplos.

¿Cómo aprender el comportamiento óptimo?

Si una acción guía a un estado satisfactorio, entonces la tendencia del sistema a producir esa acción en particular es reforzada, en el caso contrario, la tendencia es debilitada.

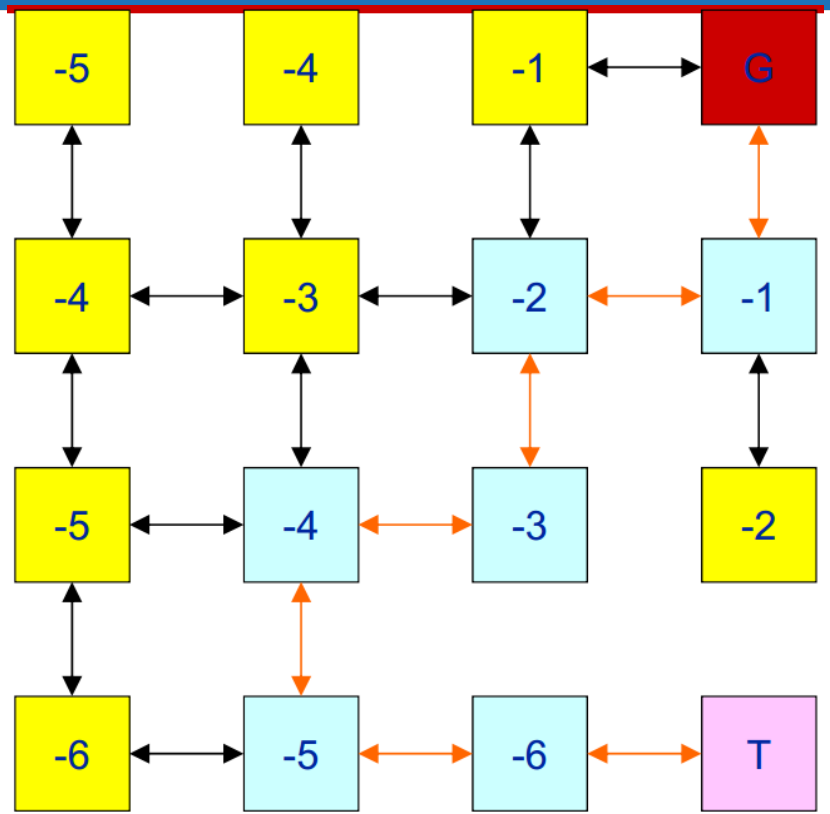
Las recompensas juegan un papel importante en el aprendizaje por refuerzo. De hecho, es la única retroalimentación que el robot recibe por lo que debería ser utilizada para mejorar la selección de acciones.

Ejemplo:



- Cuadrados: estados
- Flechas: acciones permitidas
- Costo de una acción: Función de recompensa (-1)
- Función de valor: acumulado, en este caso es mínima cantidad de pasos multiplicado por -1

Ejemplo:



- Se debe optimizar (maximizar) la recompensa, no funciona hacerlo aleatorio
- Funciones de valor: da el valor global de un estado, tomando en cuenta futuras acciones, da el camino óptimo de un estado T a una meta G

Ejemplo de utilización de RL: Quadruped Locomotion

- Usa 12 parámetros
- Política: se crea una base y se generan 15 más
- Se evalúa cada política en el robot y un gradiente se va optimizando

Ejemplo de utilización de Aprendizaje: Vuelo de Helicóptero

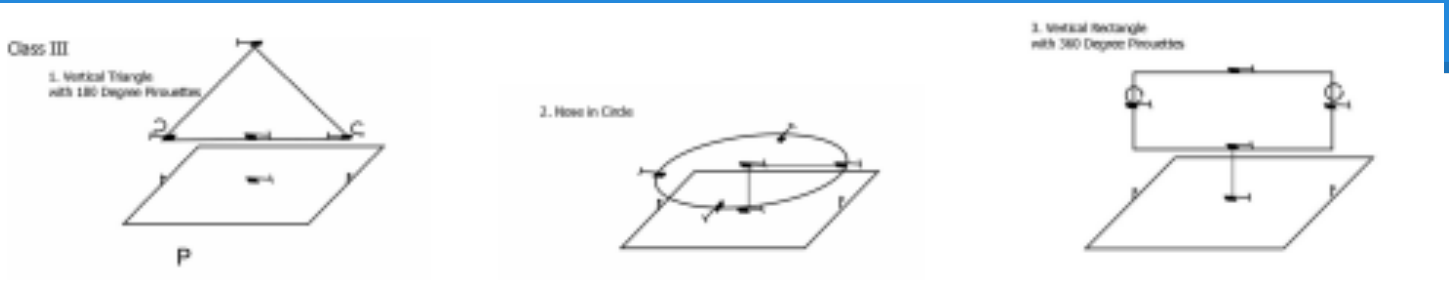
Maniobras de vuelo

Se vuelan 3 maniobras de las más difíciles en competiciones de helicopteros a control remoto

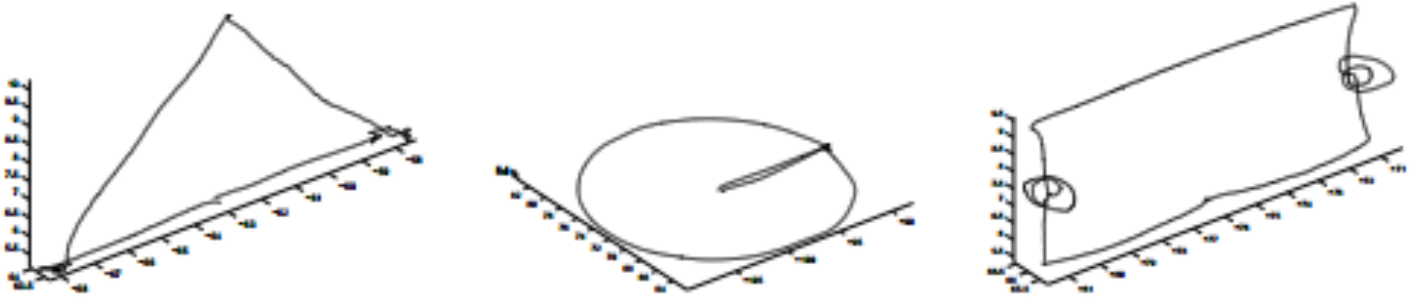
La idea es que sigan una trayectoria

- Se da una recompensa por avanzar en la trayectoria deseada.
- Se castiga si se aleja de la trayectoria deseada.

Trayectoria deseada



Trayectoria obtenida



Vuelo invertido

Muy difícil para los humanos.

Utilizaron el aprendizaje supervisado y también con grabaciones de personas realizando la maniobra.

Video ejemplo: Helicóptero autónomo

https://www.youtube.com/watch?v=ldn10JBsA3Q