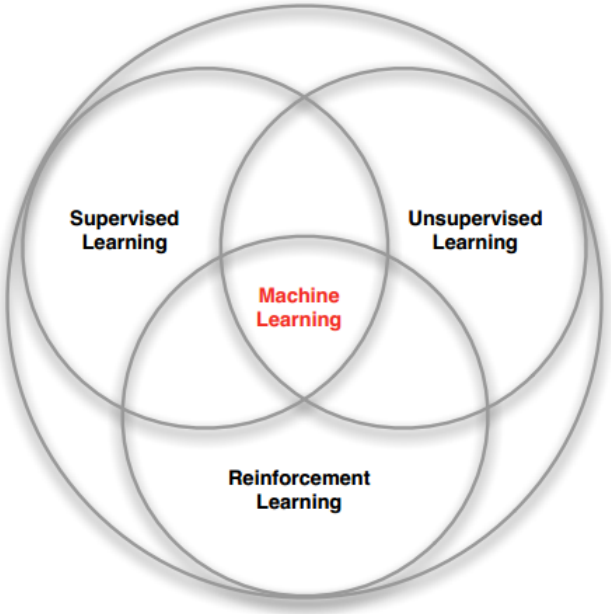
Aprendizaje por refuerzo

César Olivares

Pontificia Universidad Católica del Perú
Maestría en Informática
INF659 - Técnicas avanzadas de data mining y sistemas inteligentes

2017

Ramas del aprendizaje de máquina



(Silver 2015)

Aprendizaje por Refuerzo

- El **Aprendizaje por Refuerzo** es el problema abordado por un agente que tiene la tarea de aprender una conducta por medio de interacciones de prueba y error con un entorno dinámico. (Kaelbling 1996)
- Como en el caso de aprendizaje supervisado y no supervisado, es más una clase de problemas que un conjunto de técnicas.
- En cada interacción, el agente percibe una indicación del estado actual de su entorno y elige un acción.
- Esta acción altera el estado del entorno, cuyo valor es transmitido al agente como una señal de refuerzo.
- El agente tiene como tarea aprender las acciones que maximicen la suma total de las señales de refuerzo.

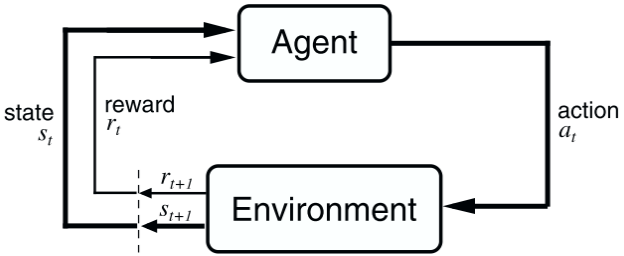
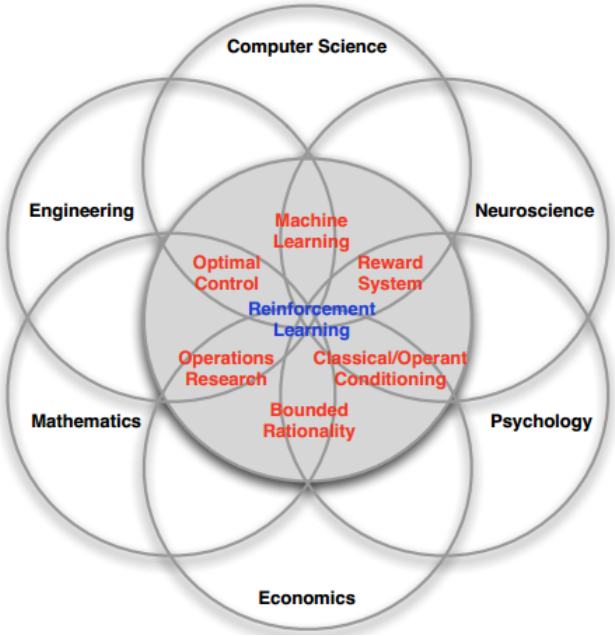


Figura 1: La interacción agente-entorno en el aprendizaje por refuerzo. (Sutton 1998)

Diversos rostros del aprendizaje por refuerzo



(Silver 2015)

Características del aprendizaje por refuerzo

- No tenemos etiquetas, sino recompensas.
- La retroalimentación es diferida, no es instantánea
- El orden de los datos es importante (datos secuenciales)
- Las acciones del agente afectan lo que ocurre en el entorno y por lo tanto también los siguiente datos que recibirá.

Ejemplos de aprendizaje por refuerzo

- Helicóptero autónomo con capacidad de hacer maniobras acrobáticas.
- Vencer al campión mundial en ajedrez
- Manejar un portafolio de inversiones
- Robot humanoide que aprende a caminar
- Jugar diferentes juegos de Atari mejor que los seres humanos

Ejemplos de aprendizaje por refuerzo

Abbeel, P., Coates, A., Quigley, M., & Ng, A. Y. (2007). An application of reinforcement learning to aerobatic helicopter flight.

Ejemplos de aprendizaje por refuerzo

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., ... others. (2015). Human-level control through deep reinforcement learning.

Recompensas

- Una *recompensa* R_t es una señal escalar de retroalimentación.
- Indica cuán bien o mal está desempeñándose un agente en el paso de tiempo t.
- El objetivo del agente es maximizar la recompensa acumulada.

El aprendizaje por refuerzo se basa en la *hipótesis de la recompensa*:

- *Cualquier* objetivo puede ser descrito como la maximización de la recompensa acumulada esperada.

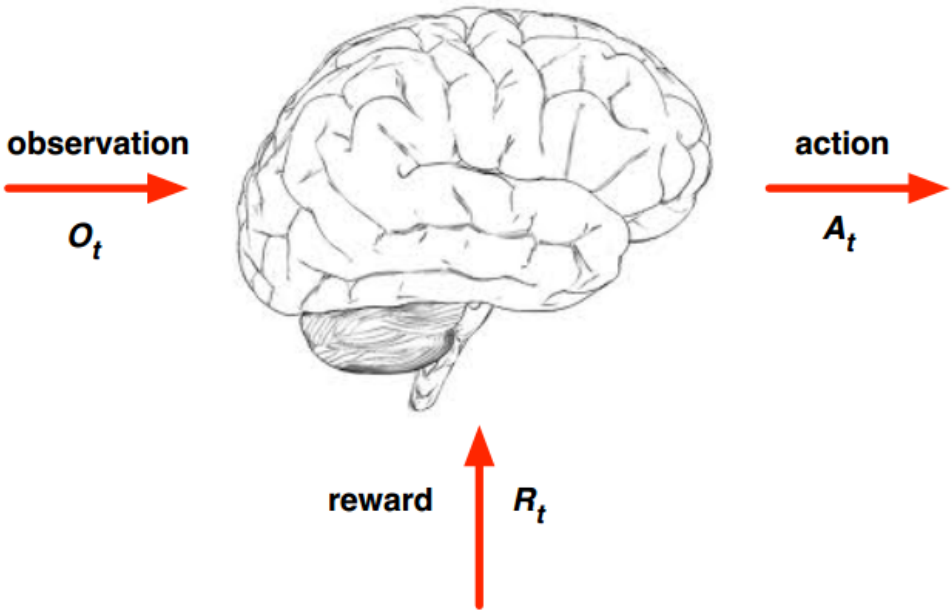
Ejemplos de recompensas

- Helicóptero autónomo
  - $+1$ por seguir la trayectoria deseada
  - -10 por estrellarse
- Ajedrez
  - $+1$/-1 por ganar/perder un juego.
- Robot
  - $+1$ por moverse hacia adelante
  - -5 por caerse
- Atari
  - $+1$/-1 por cada aumento/disminución del puntaje.

Toma de decisiones secuenciales

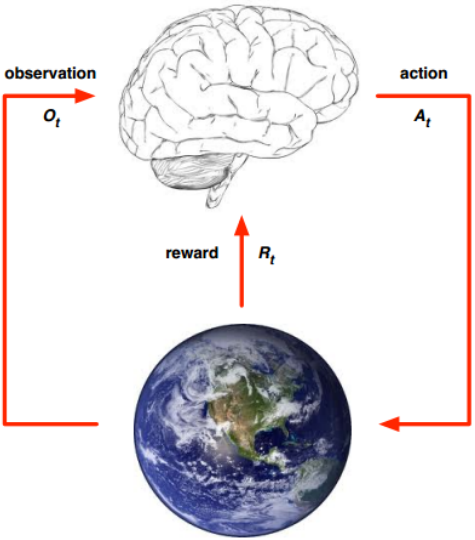
- Objetivo: *elegir las acciones que maximicen la recompensa futura total*
- Las acciones pueden tener consecuencias de largo plazo.
- La recompensa puede ser diferida.
- Podría ser mejor sacrificar una recompensa inmediata para obtener una mayor recompensa en el largo plazo.
- Ejemplos:
  - Una inversión financiera puede tomar meses en madurar.
  - Llenar el combustible de un helicóptero puede evitar una accidente horas después.
  - Bloquear movidas del oponente puede incrementar las oportunidades de victoria muchas movidas más adelante.

Agente y entorno



(Silver 2015)

Agente y entorno



(Silver 2015)

- En cada paso t el agente:
  - Ejecuta la acción A_t
  - Recibe la observación O_t
  - Recibe la recompensa escalar R_t
- El entorno:
  - Recibe la acción A_t
  - Emite la observación O_{t+1}
  - Emite la recompensa escalar R_{t+1}
- t se incrementa en cada paso del entorno.

Historia y estado

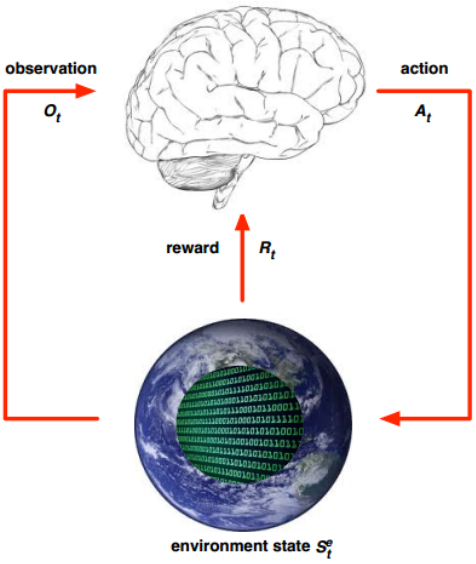
- La *historia* es la secuencia de observaciones, acciones y recompensas

$$H_t = O_1, R_1, A_1, \ldots, A_{t-1}, O_t, R_t$$

- Contiene todas las variables observables hasta t.
- Lo que ocurra después depende de la historia:
  - El agente elige acciones
  - El entorno emite observaciones/recompensas
- El *estado* es la información empleada para determinar qué ocurrirá después.
- Formalmente, el estado es una función de la historia:

$$S_t = f(H_t)$$

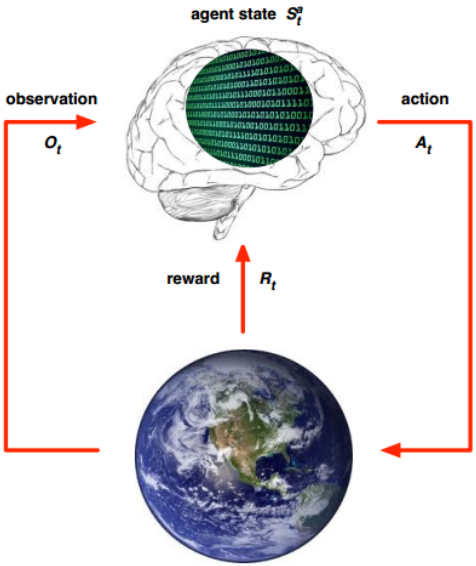
Estado del entorno



(Silver 2015)

- El *estado del entorno* S_t^e es la representación privada del entorno.
- Contiene la información que el entorno emplea para emitir la siguiente observación/recompensa.
- Usualmente este estado interno del entorno no es visible para el agente.
- Aún si S_t^e es visible, puede contener información irrelevante.

Estado del agente



(Silver 2015)

- El *estado del entorno* S_t^a es la representación interna del agente.
- Contiene la información que el agente emplea para elegir la siguiente acción.
- Esta es la información usada por los algoritmos de aprendizaje por refuerzo
- Puede ser cualquier función de la historia:

$$S_t^a = f(H_t)$$

- Cuando el agente observa directamente el estado del entorno *por completo*, entonces

$$O_t = S_t^a = S_t^e$$

Estado de información

- Un *estado de información* (llamado también *estado de Markov*) contiene tota la información relevante de la historia.
- Un estado S_t es *de Markov* si y sólo si

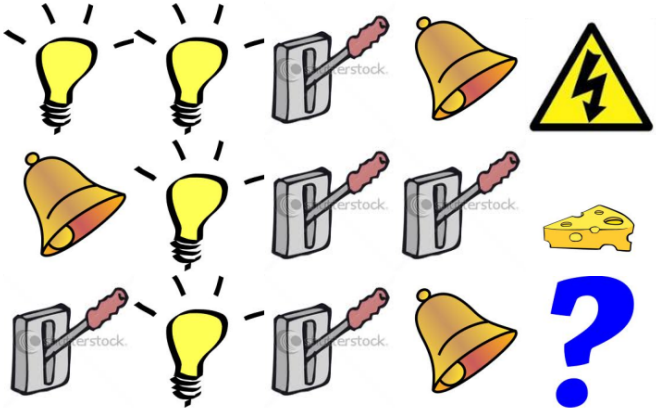
$$\mathbb{P}[S_{t+1}|S_t] = \mathbb{P}[S_{t+1}|S_1, \ldots, S_t]$$

- «El futuro es independiente del pasado dado el presente»

$$H_{1:t} \to S_t \to H_{t+1:}$$

- Una vez conocido el estado, se puede descartar la historia.

Ejemplo de la rata



(Silver 2015)

- El comportamiento del agente depende de su representación del estado.

Principales componentes de un agente de RL

Un agente de aprendizaje por refuerzo puede incluir uno o más de los siguientes componentes:

- **Política:** Una función que determina el comportamiento del agente.
- **Función de valor:** Una función que determina la calidad de cada estado y/o acción.
- **Modelo:** Una representación explícita del entorno por parte del agente.

Política

- Una *política* es una representación del comportamiento del agente.
- Es un mapa de estados a acciones, p.ej.:
- Política determinista: $a = \pi(s)$
- Política estocástica: $\pi(a|s) = \mathbb{P}[A_t = a|S_t = s]$

Función de valor

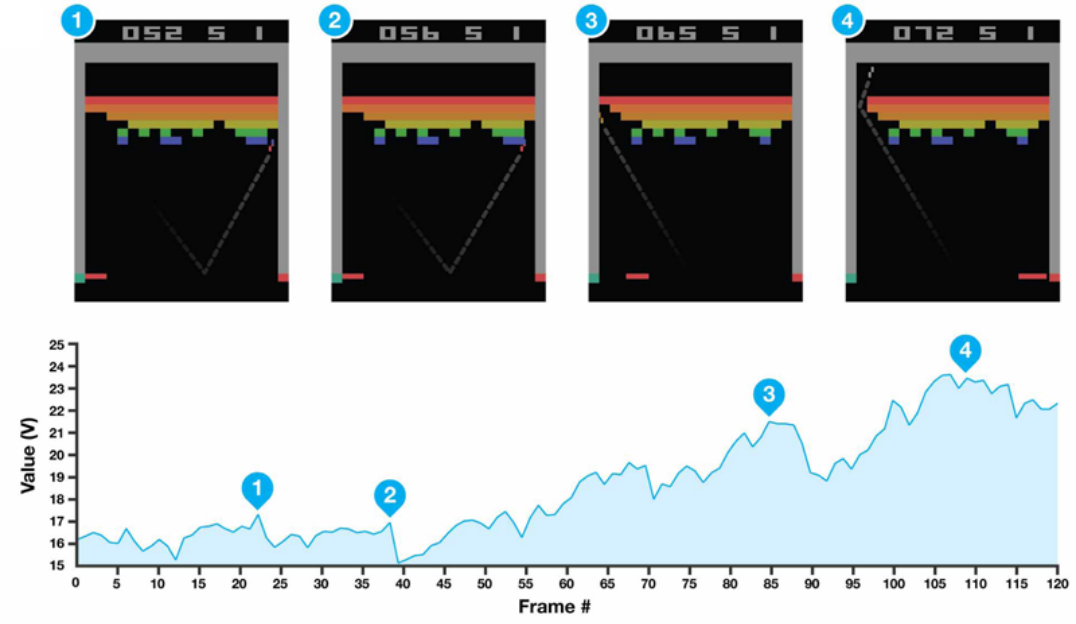
- Una *función de valor* es una predicción de una recompensa futura.
- Se la emplea para evaluar la calidad buena o mala de los estados o de los pares estado-acción.
- Sirve para elegir entre diversas acciones.
- Ejemplos:
  - Ejemplo de función de valor de un estado:

$$V_\pi(s) = \mathbb{E}_\pi[R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \ldots | S_t = s]$$

  - Ejemplo de función de valor de un par estado-acción:

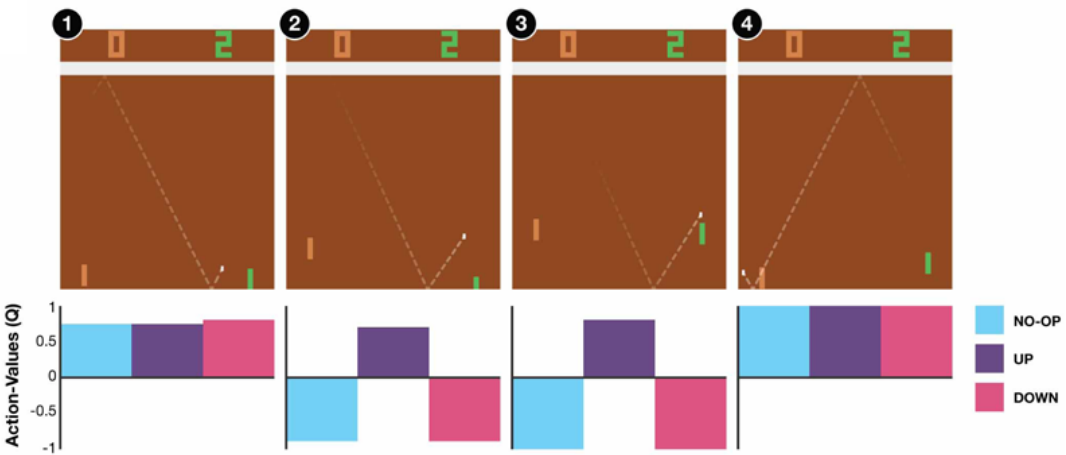
$$Q_\pi(s, a) = \mathbb{E}_\pi[R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \ldots | S_t = s, A_t = a]$$

Función de valor / Ejemplo



(Mnih 2015)

Función de valor / Ejemplo



(Mnih 2015)

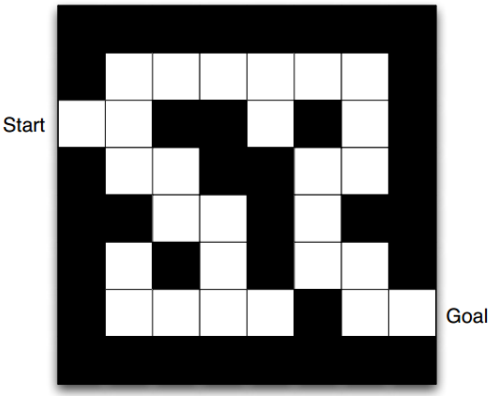
Modelo

- Un *modelo* predice lo que el entorno hará luego.
- $\mathcal{P}$ predice el siguiente estado.
- $\mathcal{R}$ predice la recompensa (inmediatamente) siguiente.

$$\mathcal{P}^a_{ss'} = \mathbb{P}[S_{t+1} = s' | S_t = s, A_t = a]$$

$$\mathcal{R}^a_s = \mathbb{E}[E_{t+1} | S_t = s, A_t = a]$$

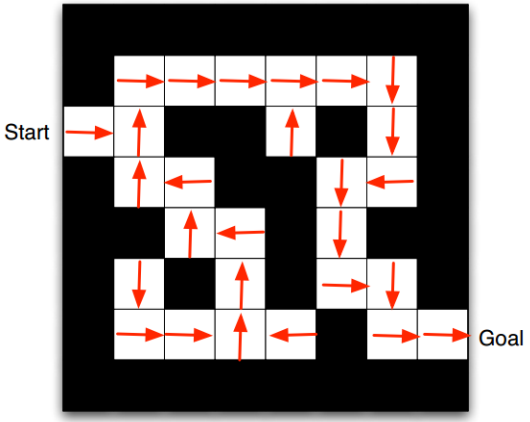
Ejemplo del laberinto



(Silver 2015)

- Recompensas: -1 en cada paso de tiempo
- Acciones posibles: N, E, S, O
- Estados: ubicación del agente

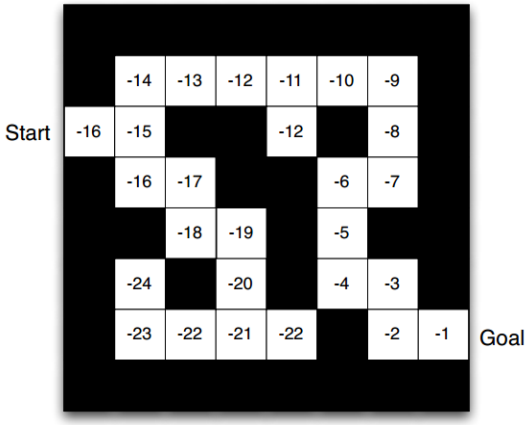
Ejemplo del laberinto / Política



(Silver 2015)

- Las flechas representan la política $\pi(s)$ para cada estado s

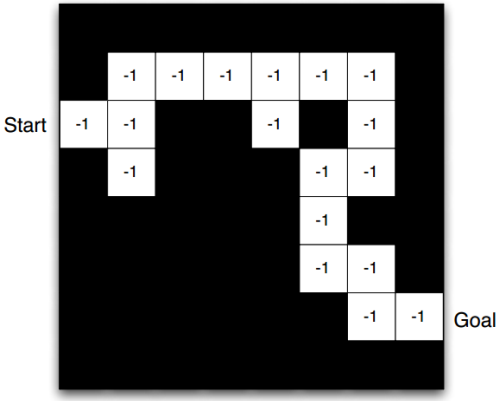
Ejemplo del laberinto / Función de valor



(Silver 2015)

- Los números representan el valor $v_\pi(s)$ para cada estado s

Ejemplo del laberinto / Modelo



- El agente puede contar con un modelo interno del entorno
- Dinámica: cómo las acciones alteran el estado
- Recompensas: qué recompensa se recibe en cada estado
- El modelo puede ser imperfecto

(Silver 2015)

- La configuración de la grilla representa el modelo de transición $\mathcal{P}^a_{ss'}$
- Los números representan la recompensa inmediata $\mathcal{R}^a_s$ para cada estado s (en este ejemplo es la misma para todas las acciones a)

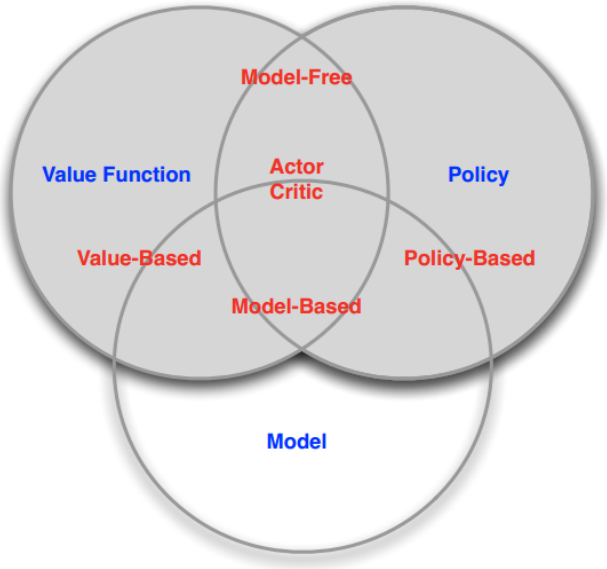
Taxonomía de los agentes de RL / Por componente

- Basados en el valor
  - *Sin política (queda implícita)*
  - Función de valor
- Basados en políticas
  - Política
  - *Sin función de valor (queda implícita)*
- Actor/Crítico
  - Política
  - Función de valor

Taxonomía de los agentes de RL / Por modelo

- Sin modelo
  - Política y/o función de valor
  - *Sin modelo*
- Basados en modelos
  - Política y/o función de valor
  - Modelo

Taxonomía de los agentes de RL



(Silver 2015)

Exploración y explotación

- El aprendizaje por refuerzo se basa en el ensayo y error.
- El agente debe descubrir una buena política
- ...a partir de sus experiencias con el entorno
- ...sin perder mucha recompensa en el camino.
- *Exploración* es buscar más información sobre el entorno.
- *Explotación* es aprovechar la información conocida para maximizar la recompensa.
- Usualmente es importante explorar **y** explotar.
- Ejemplos:
  - *Explotación:* Ir a tu restaurante favorito.
  - *Exploración:* Probar un nuevo restaurante.
  - *Explotación:* Mostrar el anuncio comercial más exitoso.
  - *Exploración:* Mostrar un anuncio nuevo.
  - *Explotación:* Hacer la mejor movida que conoces.
  - *Exploración:* Probar una movida experimental.

Aprendizaje por refuerzo **profundo**

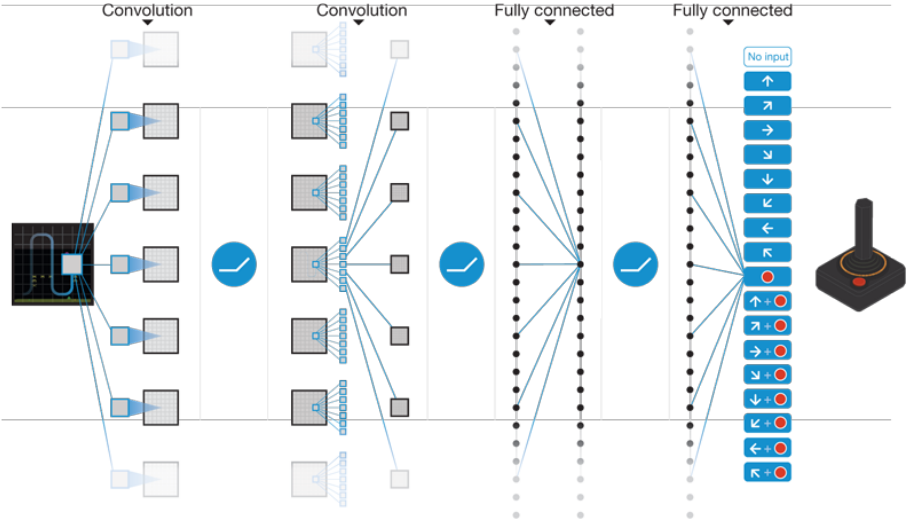
- Se habla de aprendizaje por refuerzo *profundo* cuando se emplea *redes neuronales profundas* para aproximar cualquiera de los siguientes componentes del aprendizaje por refuerzo:
  - La función de valor $V(s;\theta)$ o $Q(s,a;\theta)$
  - La política $\pi(a|s;\theta)$
  - El modelo
    - La función de transición de estados

$$\mathcal{P}^a_{ss'} = \mathbb{P}[S_{t+1} = s' | S_t = s, A_t = a, \theta]$$

    - La función de recompensa

$$\mathcal{R}^a_s = \mathbb{E}[E_{t+1} | S_t = s, A_t = a, \theta]$$

Deep Q Network (DQN)



Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., ... others. (2015). Human-level control through deep reinforcement learning.

Bibliografía

- Abbeel, P., Coates, A., Quigley, M., & Ng, A. Y. (2007). An application of reinforcement learning to aerobatic helicopter flight. In Advances in neural information processing systems (pp. 1–8).
- Géron, A. (2017). Hands-On Machine Learning with Scikit-Learn and TensorFlow Concepts, Tools, and Techniques to Build Intelligent Systems. O'Reilly Media.
- Kaelbling, L. P., Littman, M. L., & Moore, A. W. (1996). Reinforcement learning: A survey. Journal of Artificial Intelligence Research, 4, 237–285.
- Li, Y. (2017). Deep Reinforcement Learning: An Overview.
- Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., ... others. (2015). Human-level control through deep reinforcement learning. Nature, 518(7540), 529–533.
- Silver, D. (2015). UCL Course on RL. Retrieved from http://www0.cs.ucl.ac.uk/staff/d.silver/web/Teaching.html
- Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., van den Driessche, G., ... Hassabis, D. (2016). Mastering the game of Go with deep neural networks and tree search. Nature, 529(7587).
- Tesauro, G. (1995). Temporal difference learning and TD-Gammon. Communications of the ACM, 38(3), 58–68.
- Sutton, R. S., & Barto, A. G. (1998). Reinforcement learning: An introduction (Vol. 1). MIT press Cambridge.
- Szepesvári, C. (2010). Algorithms for reinforcement learning. Synthesis Lectures on Artificial Intelligence and Machine Learning, 4(1), 1–103.