

Aprendizaje por refuerzo. - ARef

- En los años **50** se empezó a trabajar
- Los juegos han sido una de las principales aplicaciones de interés. → Backgammon (TD-gammon)
- También se ha utilizado en sistemas de control.

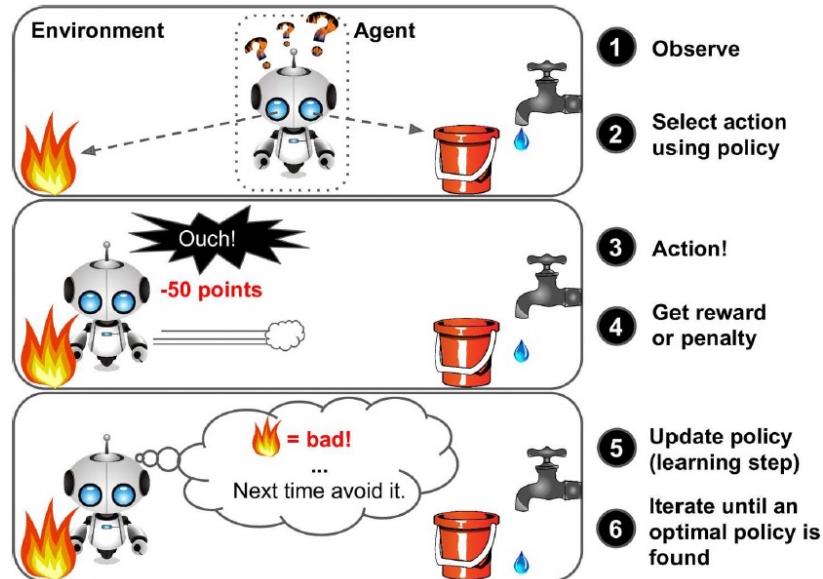
Avance notorio del ARef :

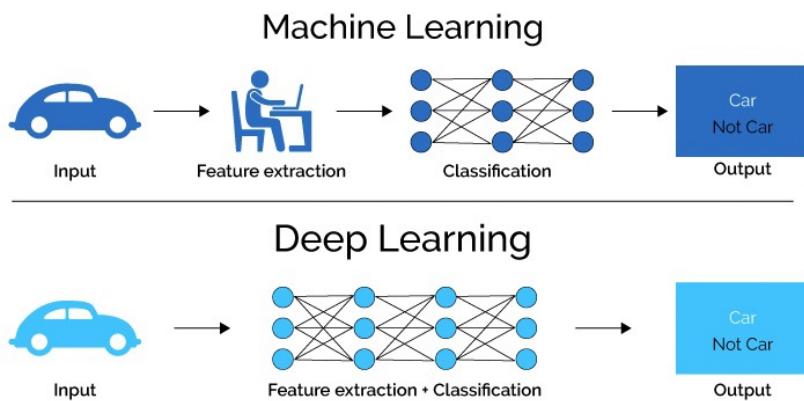
- Start-up Británica Deep Mind (2013)
 - ↳ Crearon sistema que puede aprender a jugar casi cualquier juego de Atari.
 - ↳ El sistema utiliza imágenes para aprender a jugar sin tener conocimiento previo de las reglas del juego.
 - ↳ En 2016 Alpha Go de Deep Mind le gano partidas a un jugador profesional de Go.

En 2014 Google adquiere Deep Mind por US \$ 500 M.

NOTA: El éxito de Deep Mind vs. técnicas convencionales de ARef radica en la **incorporación de Deep learning**

Esquema general →
ARef





Machine Learning vs. Deep learning.

Módulos básicos de estudio en ARef.

- ↳ Optimización de recompensas (Rewards)
- ↳ Procesos de decisión Markovianos (Markov Decision Processes)
- ↳ Deep Q-Networks.

Optimización de recompensas.

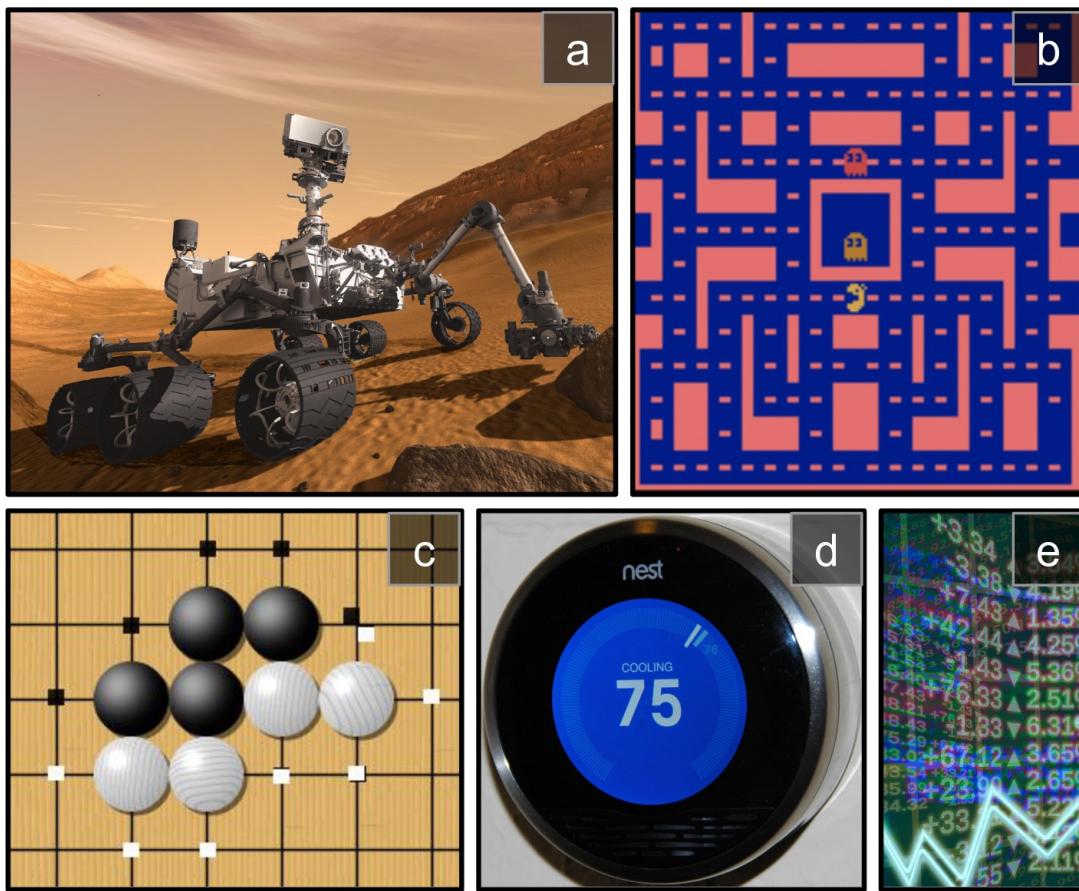
→ Agente → observaciones → acciones } Ambiente.
 Agent → observations → actions } Environment.

NOTA: El objetivo principal en ARef es maximizar las recompensas esperadas a lo largo del tiempo.

+ Rewards → Pleasure

- Rewards → Pain

El agente actúa en el ambiente y aprende por intento y error a maximizar recompensas positivas (placer) y minimizar recompensas negativas (dolor).

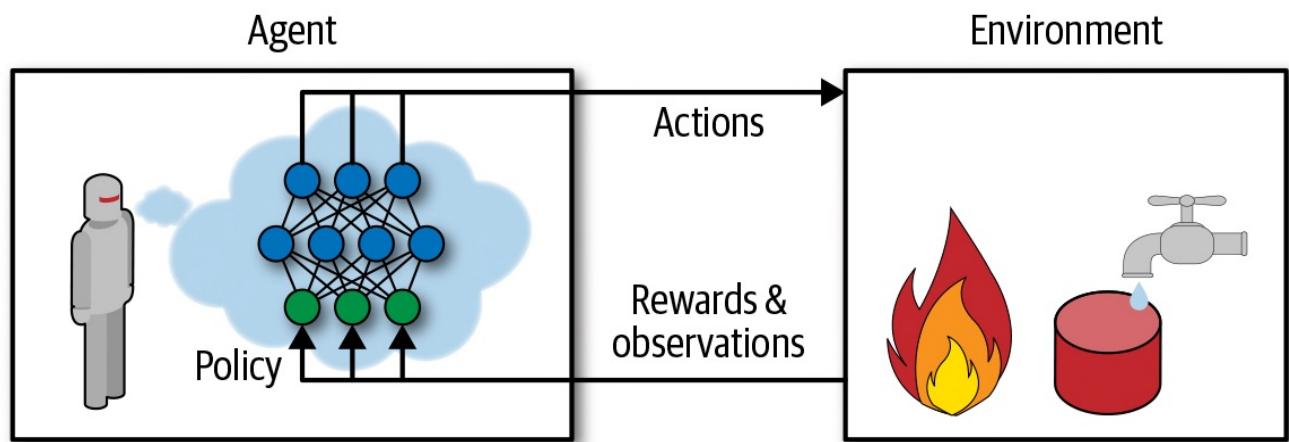


Ejemplos aplicaciones Aprendizaje por Refuerzo

NOTA: En ocasiones no es evidente el concepto de recompensa +
L) Ej: el agente debe encontrar la salida tan rápido como sea posible!

Otros ejemplos de ARf: → Autos que se manejan solos
 → Sistemas de recomendación
 → Ubicación de publicidad en web

→ Búsqueda de estrategia (Policy search)



Policy: algoritmo usado por el agente para determinar sus acciones.

↳ Puede ser determinístico o estocástico

↳ No tiene que observar directamente el ambiente

Ej: Robot aspiradora: → Recompensa → cantidad de polvo almacenado en 30 min
→ Estrategia → movimiento \uparrow /s
(estocástica) $p(\uparrow) = f$
→ movimiento $\leftarrow \circlearrowright$ /s

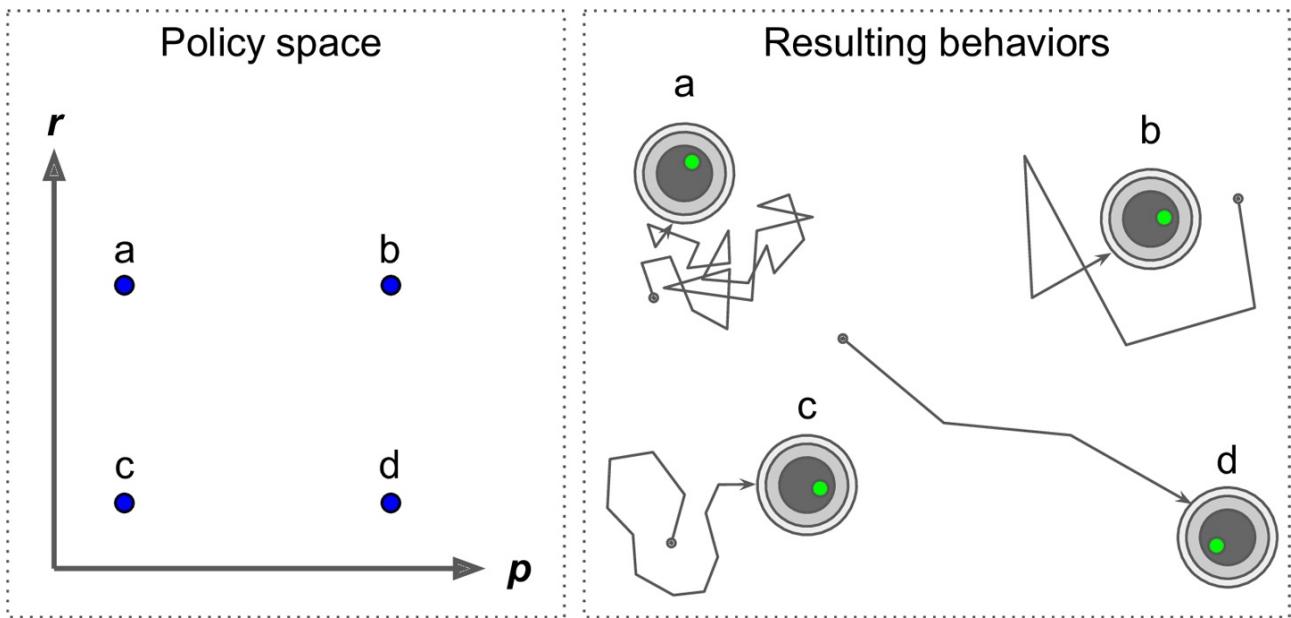
$$p(\leftarrow \circlearrowright) = 1 - f$$

→ rotación $[-\gamma, +\gamma]$
aleatoria

$$a \sim p(a)$$

Cómo entrenar el robot aspiradora?

↳ Parámetros del modelo: f, γ .



Policy Search → Por fuerza bruta.

Muchas opciones de parámetros (combinaciones)!

Alternativas: → Algoritmos genéticos

→ Optimización (Gradientes $\frac{\partial \text{recompensa}}{\partial \text{parámetro}}$)

↳ Policy Gradients (PG)

↳ TensorFlow
(Autodiff) ?

NOTA: Se requiere de un ambiente simulado

Opciones en Python: PyBullet, MuJoCo → 3D física

OpenAI Gym → Juegos de Atari

Juegos de Mesa

2D, 3D física

