🤖 Aprendizaje por Refuerzo

Reinforcement Learning

Autor: Jesús Emmanuel Martínez García

Fecha: Octubre 2025

📚 Contenido

1. Introducción e historia
2. Fundamentos del Aprendizaje por Refuerzo
3. Conceptos técnicos y matemáticos
4. Principales algoritmos
5. Aplicaciones reales
6. Ejemplos de proyectos
7. Conclusiones y futuro

🧠 ¿Qué es el Aprendizaje por Refuerzo?

El aprendizaje por refuerzo (RL) es un paradigma del aprendizaje automático en el que un **agente aprende a tomar decisiones óptimas** mediante **interacción con un entorno**, **recibiendo recompensas o castigos** según sus acciones.

www.aprendemachinelearning.com

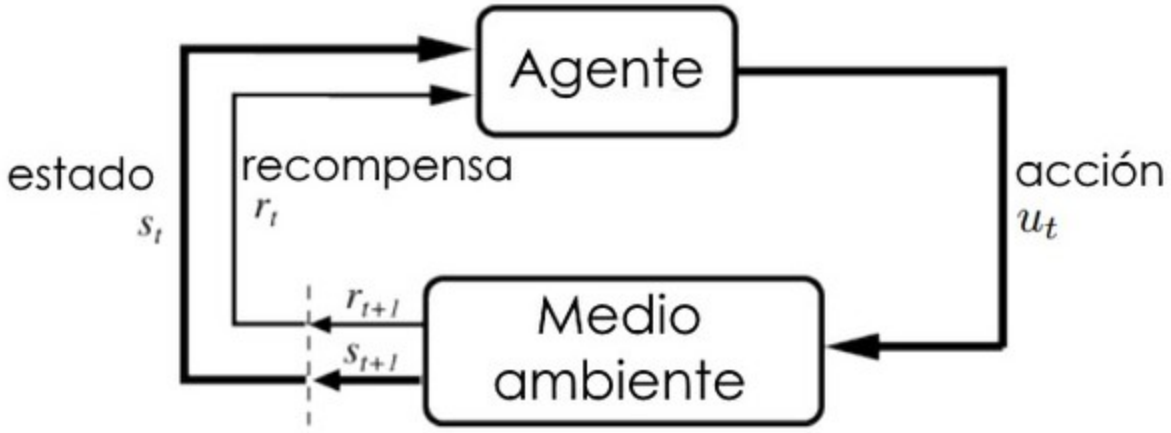
🕰️ Breve Historia del RL

Año	Acontecimiento	Descripción
1950s	**Alan Turing**	Propuso la idea de máquinas que aprendieran mediante recompensas.
1952	**Arthur Samuel**	Creó un programa de ajedrez que aprendía por refuerzo.
1980s	**Richard Sutton & Andrew Barto**	Formalizan el RL moderno.
1990s	**Q-learning (Watkins)**	Primer algoritmo RL ampliamente usado.
2010s	**DeepMind (Google)**	Fusión de RL con Deep Learning → *Deep Reinforcement Learning*.
2016	**AlphaGo**	RL logra vencer a campeones humanos en Go.

⚙ Fundamentos del Aprendizaje por Refuerzo

El RL se basa en tres elementos principales:

- **Agente:** quien aprende y actúa.
- **Entorno:** con el que interactúa.
- **Recompensa:** señal que indica qué tan buena fue una acción.



[Figure source: Sutton & Barto, 1998]

🔁 Ciclo de Aprendizaje

1. El **agente** observa el **estado** del entorno.
2. Toma una **acción** según su **política (π)**.
3. El entorno responde con una **recompensa (R)** y un **nuevo estado (S')**.
4. El agente **ajusta su política** para maximizar la recompensa acumulada.

🧮 Marco Matemático

El aprendizaje por refuerzo se modela con un **Proceso de Decisión de Markov (MDP)**:

$$MDP = (S, A, P, R, \gamma)$$

Donde:

- (S): conjunto de estados
- (A): conjunto de acciones
- (P(s'|s,a)): probabilidad de transición
- (R(s,a)): recompensa esperada
- (\gamma): factor de descuento

🎯 Objetivo del Agente

Maximizar la **recompensa acumulada esperada**:

$$G_t = E\left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1}\right]$$

El agente busca la **política óptima** π* que maximiza este valor.

🔍 Algoritmos Principales

1. **Métodos basados en valor**
    - Q-Learning
    - SARSA
2. **Métodos basados en política**
    - REINFORCE
    - Actor-Critic

3. **Métodos avanzados**
    - Deep Q-Networks (DQN)
    - Proximal Policy Optimization (PPO)
    - AlphaZero (autoaprendizaje con simulaciones)

Q-Learning

Aprende una **tabla Q(s,a)** que estima el valor esperado de tomar una acción en un estado.

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

	x0		x1		x2		x3		x4		x5		x6		x7		x8		x9	
	Subir	Bajar	Subir	Bajar	Subir	Bajar	Subir	Bajar	Subir	Bajar	Subir	Bajar	Subir	Bajar	Subir	Bajar	Subir	Bajar	Subir	Bajar
y7	-2.4	-0.2	-0.3	-0.	-0.1	-0.	-0.	-0.	-0.	-0.	-0.1	-0.1	-0.3	-0.2	-39.6	-27.3	-16.1	-24.4	-21.2	-16.6
y6	-4.	-0.9	-0.6	-0.1	-0.	-0.	-0.	-0.	-0.	-0.	-0.	-0.	-0.1	-0.3	-9.7	-0.7	-15.7	-14.	-20.7	-45.9
y5	-5.4	-3.4	-0.4	-0.1	-0.	-0.	-0.	-0.	-0.	-0.	-0.1	-0.	-0.3	-0.	-0.7	-0.7	-17.4	-1.1	-53.2	-28.5
y4	-3.4	-1.4	-0.1	-0.4	-0.	0.	-0.	-0.	-0.	-0.	-0.	-0.	-0.1	-0.	-1.9	-0.6	-8.5	-19.5	-46.	-16.7
y3	-0.4	-3.4	-0.	-0.1	-0.	-0.	-0.	0.	-0.	-0.	-0.	-0.	-0.	-0.1	-23.3	-5.3	-12.4	-4.1	-2.4	-28.6
y2	0.4	-1.1	0.	-0.1	-0.	-0.	-0.	-0.	-0.	-0.	-0.	-0.	-0.1	-0.4	-2.7	-14.4	1.9	-9.	3.8	-16.6
y1	-1.8	-2.2	-0.3	-0.1	-0.	-0.	-0.	0.	-0.	-0.	-0.	-0.1	-0.	-0.1	-17.6	-29.	-9.8	-4.1	-39.3	5.7
y0	-2.8	-0.7	-0.3	-0.1	-0.	0.	-0.	-0.	-0.	-0.	-0.1	-0.	-0.	-0.2	-5.7	-14.4	-16.5	0.1	-35.9	2.7

🤖 Deep Reinforcement Learning (DRL)

Combina **redes neuronales profundas** con aprendizaje por refuerzo.

Ejemplo:

- Entrada: imagen del entorno (píxeles).
- Salida: acción a tomar.

🌍 Aplicaciones Reales

- **Robótica:** navegación, manipulación de objetos.
- **Juegos:** AlphaGo, Atari, StarCraft.
- **Finanzas:** trading algorítmico, gestión de portafolios.
- **Vehículos autónomos:** toma de decisiones en tiempo real.
- **Energía:** optimización de consumo en redes eléctricas.

🧪 Ejemplos de Proyectos para Aprender

1. **CartPole (OpenAI Gym)**
   → Equilibrar una varilla sobre un carrito.

2. **MountainCar**
   → Empujar un auto hasta una colina.

3. **Atari Pong con DQN**
   → Entrenar una red neuronal a jugar Pong desde píxeles.

4. **Robot simulador (PyBullet o Mujoco)**
   → Aprendizaje de locomoción o manipulación.

🚀 Implementaciones Populares

- OpenAI Gym
- Stable Baselines3
- TensorFlow Agents / PyTorch RL
- Unity ML-Agents

🔮 Futuro del RL

- Integración con **modelos generativos y LLMs**
- Aplicaciones en **energía, salud y clima**
- Enfoque en **aprendizaje con menos datos y más eficiencia**
- Desarrollo de **agentes generalistas (AGI-oriented)**

🧾 Conclusiones

✅ El aprendizaje por refuerzo permite **descubrir estrategias óptimas** sin supervisión directa.

✅ Ha revolucionado campos como los videojuegos, robótica y finanzas.

✅ Aún presenta retos: **eficiencia, exploración y estabilidad**.

✅ Es un área con **gran proyección futura** en IA avanzada.

📚 Referencias

- Sutton, R. & Barto, A. (2018). *Reinforcement Learning: An Introduction*.
- Silver, D. et al. (2016). *Mastering the game of Go with deep neural networks and tree search*. *Nature*.
- OpenAI Gym Documentation.
- DeepMind Research Publications.