Clase 161 — RL: aprendizaje por recompensa, Gymnasium (Farama)

Parte: 2 — Deep Learning · Fuente: Géron, cap. 18 § Introduction to Reinforcement Learning + docs Gymnasium. Duración estimada: 75 min.

Clase 161 — RL: aprendizaje por recompensa, Gymnasium (Farama)

> *Parte: 2 — Deep Learning · Fuente: Géron, cap. 18 § Introduction to Reinforcement Learning + docs Gymnasium. Duración estimada: 75 min.*

Objetivo

Entender el paradigma de Reinforcement Learning — un agente interactúa con un environment, observa states, toma actions, recibe rewards, y aprende una policy que maximiza recompensa acumulada. Conocer Gymnasium (Farama Foundation, fork del antiguo OpenAI Gym que dejó de mantenerse en 2022) — la librería estándar de environments para benchmarks.

> *Nota: el nombre "OpenAI Gym" es histórico. Desde 2022, OpenAI dejó de mantenerlo y la Farama Foundation tomó el relevo creando Gymnasium — drop-in replacement con mejor mantenimiento.*

Resultados de aprendizaje

Al finalizar, el estudiante podrá:

- Definir los 5 componentes RL: agent, environment, state, action, reward.
- Usar gymnasium: env = gym.make('CartPole-v1'); obs, _ = env.reset(); obs, reward, terminated, truncated, info = env.step(action).
- Implementar un policy random como baseline.
- Reconocer la diferencia entre on-policy (PPO, A2C) y off-policy (DQN, SAC).
- Saber dónde RL es apropiado (juegos, robótica, navegación) vs donde no (clasificación, regresión típicas).

Temas

- Bloque básico: state → action → reward → next_state.
- Episodic vs continuous tasks.
- Discrete vs continuous action spaces.
- Discount factor γ: balance entre recompensa inmediata y futura.
- Gymnasium API estándar (reset, step).
- Environments populares: CartPole, MountainCar, Atari, MuJoCo, BipedalWalker.

Definiciones y características

- Policy π(a|s): la estrategia — distribución sobre acciones dado state.
- Trajectory / episode: secuencia (s_0, a_0, r_0, s_1, a_1, ...) hasta done.
- Return: suma de recompensas futuras G_t = Σ γ^k r_{t+k}.
- Value function V(s): expected return desde s siguiendo π.
- Q-function Q(s, a): expected return tras tomar a desde s.
- Gymnasium step: devuelve 5-tuple (obs, reward, terminated, truncated, info) (Gym original devolvía 4 — done combinaba terminated/truncated).

- Action space: Discrete(N) o Box(low, high, shape).

Dataset / recursos

- gymnasium library (pip install gymnasium).
- Environments built-in: CartPole-v1, LunarLander-v3, MountainCar-v0.
- Librerías: gymnasium, numpy, matplotlib.

Ejercicios

1. Entorno básico: env = gym.make('CartPole-v1', render_mode='human'). Correr 10 episodios con acciones random; reportar duración promedio.
2. Estructura del state: imprimir env.observation_space y env.action_space para CartPole, MountainCar, LunarLander.
3. Policy heurística: para CartPole, action = 0 if pole_angle < 0 else 1 (push opuesto al tilt). Comparar contra random.
4. Return discounted: calcular G_t = Σ γ^k r_{t+k} con γ=0.99 sobre un episodio.
5. Render: usar render_mode='rgb_array' y guardar frames para crear un gif.

Homework verificable

Sobre CartPole-v1:

1. Implementar 3 policies: random, heurística simple, "siempre derecha".
2. Correr 100 episodios de cada; reportar return promedio y desviación.
3. Verificar que heurística supera random.

Criterio de aceptación: random ≈ 20 steps; heurística ≥ 80 steps; "siempre derecha" muere rápido.

Errores comunes

Síntoma / mensaje	Causa y cómo arreglar
gym.make(...) deprecated warning	Estás usando el viejo OpenAI Gym. Fix: pip
env.step(action) devuelve 4 valores	Gym viejo. Gymnasium devuelve 5. Fix: actu
env.reset() devuelve un solo valor en tuto	Gym pre-2022 devolvía obs. Gymnasium devue
Render no muestra ventana	render_mode no especificado al crear env.
Atari environments fail	Requieren pip install gymnasium[atari,acce

Preguntas frecuentes

¿RL vs supervised?

Supervised: aprende f(x) = y con labels. RL: aprende π(s) = a con feedback indirecto (rewards). Más cercano a humanos pero más difícil.

¿Cuándo RL?

Juegos (AlphaGo), robótica, navegación, sistemas de recomendación con feedback de usuarios. NO para clasificación / regresión típicas.

¿Gymnasium o RLlib o Stable-Baselines?

Gymnasium = environments. Stable-Baselines3 (PyTorch) o RLlib (Ray) = algoritmos. Estándar 2026: SB3 para experimentos, RLlib para producción a escala.

¿RL con LLMs?

Sí — RLHF y DPO (clase 128) son aplicaciones de RL a LLMs.

¿Cuánto cuesta entrenar?

CartPole: 1 minuto. Atari: horas-días. AlphaGo: semanas en cluster. RL es muy costoso en datos (interacciones).

Referencias

- Géron, cap. 18 — Reinforcement Learning.
- Sutton & Barto (2018), Reinforcement Learning: An Introduction (2nd ed.) — biblia del RL.
- Gymnasium docs.
- Stable-Baselines3 docs.

Siguiente clase

Clase 162 — Policy gradients

Apéndice: notebook (primer bloque)

Primera celda ejecutable del notebook de la clase.

```
# Imports y configuración inicial
```

Archivos complementarios

- notebook.ipynb