Clase 165 — RL moderno: A3C, PPO, SAC (vista general)

Parte: 2 — Deep Learning · Fuente: papers A3C (Mnih 2016), PPO (Schulman 2017), SAC (Haarnoja 2018) + docs Stable-Baselines3. Duración estimada: 65 min.

Clase 165 — RL moderno: A3C, PPO, SAC (vista general)

> *Parte: 2 — Deep Learning · Fuente: papers A3C (Mnih 2016), PPO (Schulman 2017), SAC (Haarnoja 2018) + docs Stable-Baselines3. Duración estimada: 65 min.*

Objetivo

Vista general (sin implementación desde cero) de los 3 algoritmos modernos de RL: A3C (Asynchronous Advantage Actor-Critic), PPO (Proximal Policy Optimization — el default industrial), y SAC (Soft Actor-Critic — off-policy, continuous actions). Saber cuál elegir y cómo usarlos con Stable-Baselines3.

Resultados de aprendizaje

Al finalizar, el estudiante podrá:

- Diferenciar on-policy (A3C, PPO) de off-policy (SAC, DQN).
- Reconocer la idea de Actor-Critic: dos redes — actor (policy) + critic (value).
- Entender el clipped objective de PPO: limita updates a [1-ε, 1+ε] veces el old policy → estabilidad.
- Usar Stable-Baselines3: PPO('MlpPolicy', env).learn(total_timesteps=100_000).
- Elegir: PPO para discreto/continuo, on-policy. SAC para continuo, off-policy, sample-efficient.

Temas

- Actor-Critic: actor da policy, critic da V(s). Advantage = G - V(s).
- A3C: async + advantage actor-critic. Paralelización con multiples workers.
- A2C: variante sincrónica (más simple).
- PPO: clipped surrogate objective.
- SAC: actor-critic off-policy + entropy regularization.
- Stable-Baselines3 como librería estándar.

Definiciones y características

- Actor: red de policy $\pi_\theta(a|s)$.
- Critic: red de valor $V_\varphi(s)$ o $Q_\varphi(s,a)$.
- Advantage A(s, a): cuán mejor es a que el promedio.
- GAE (Generalized Advantage Estimation): estimador robusto del advantage usado en PPO.
- PPO clip: min(ratio · A, clip(ratio, 1-ε, 1+ε) · A) con ratio = π_{new}/π_{old}.
- SAC entropy bonus: $H(\pi)$ se maximiza junto al return → exploración natural.

Dataset / recursos

- LunarLander, CartPole, Pendulum.
- Librerías: gymnasium, stable-baselines3 (pip install stable-baselines3[extra]).

Ejercicios

1. PPO con SB3: from stable_baselines3 import PPO; model = PPO('MlpPolicy', 'CartPole-v1', verbose=1); model.learn(50_000). Evaluar.
2. SAC en Pendulum: SAC('MlpPolicy', 'Pendulum-v1').learn(20_000).
3. Comparar: PPO vs SAC en LunarLander; reportar return y sample efficiency.
4. TensorBoard: PPO(..., tensorboard_log='./tb/') y ver curvas.
5. Custom callback: EvalCallback para evaluar cada N steps y guardar mejor modelo.

Homework verificable

Entrenar PPO en LunarLander-v3:

1. PPO('MlpPolicy', 'LunarLander-v3', verbose=1, tensorboard_log='./tb/').
2. model.learn(total_timesteps=500_000).
3. Evaluar 100 episodios; reportar mean reward.

Criterio de aceptación: mean reward ≥ 200 (problema resuelto); TensorBoard muestra curva ascendente.

Errores comunes

Síntoma / mensaje	Causa y cómo arreglar
PPO no converge	LR alto o env muy complejo. Fix: learning_
SAC en discrete actions	SAC es continuous. Fix: PPO o DQN para dis
Reward "stuck" en ~0	El env tal vez no es resoluble con MLP. Fi
Training muy lento	Pocos parallel envs. Fix: make_vec_env('Ca
Modelo guardado no carga	Versión de SB3 incompatible. Fix: model =

Preguntas frecuentes

¿PPO o SAC?

- PPO: discreto o continuo, on-policy, más estable, default industrial.
- SAC: continuo, off-policy, sample-efficient, mejor performance pico.
- Estándar: PPO primero; SAC si necesitás más sample efficiency.

¿Cuántos steps?

CartPole: 50k. LunarLander: 500k. Atari: 10M+. Robótica MuJoCo: 1M-10M.

¿RL para LLM (RLHF)?

PPO es el algo estándar de RLHF. DPO (clase 128) lo reemplazó por simplicidad.

¿Cuándo NO usar RL?

Si el problema es supervised (clasificación, regresión), no uses RL. RL es sample-inefficient y caro.

¿Hyperparameter tuning?

Optuna sobre SB3 o rl_zoo3 que tiene hyperparams pre-tuneados para muchos envs.

Referencias

- Mnih et al. (2016), Asynchronous Methods for Deep Reinforcement Learning (A3C), ICML.
- Schulman et al. (2017), PPO.
- Haarnoja et al. (2018), Soft Actor-Critic, ICML.
- Stable-Baselines3 docs.
- Sutton & Barto, capítulos avanzados.

Siguiente clase

Clase 166 — TF Serving + gRPC (+ ONNX, TensorRT, vLLM/TGI)

Apéndice: notebook (primer bloque)

Primera celda ejecutable del notebook de la clase.

```
# Imports y configuración inicial
```

Archivos complementarios

- notebook.ipynb