Clase 164 — TD Learning, Q-Learning, Deep Q-Networks

Parte: 2 — Deep Learning · Fuente: Géron, cap. 18 § Q-Learning y § Deep Q-Learning.
Duración estimada: 80 min.

Clase 164 — TD Learning, Q-Learning, Deep Q-Networks

> *Parte: 2 — Deep Learning · Fuente: Géron, cap. 18 § Q-Learning y § Deep Q-Learning. Duración estimada: 80 min.*

Objetivo

Implementar Q-Learning clásico (Watkins 1989) y su versión moderna DQN (Mnih et al. 2015, Nature paper que aprendió Atari desde pixels). Off-policy, model-free, bootstrap. Conocer los 2 trucos que hicieron a DQN funcionar: experience replay y target network.

Resultados de aprendizaje

Al finalizar, el estudiante podrá:

- Explicar la update Q-learning: Q(s,a) ← Q(s,a) + α[r + γ max_a' Q(s',a') - Q(s,a)].
- Implementar Q-learning tabular en FrozenLake.
- Construir un DQN: red state → Q(a) para todas las actions, MSE entre Q predicted y r + γ max_a' Q'(s',a').
- Implementar replay buffer (almacenar transitions, samplear batch para training).
- Implementar target network (copia frozen actualizada cada N steps).

Temas

- TD (Temporal Difference) error: δ = r + γ V(s') - V(s).
- Q-learning: off-policy, sigue greedy de Q*.
- ε-greedy exploration: con prob ε explora random, sino greedy.
- Replay buffer: deque de (s, a, r, s', done).
- Target network: estabilidad (sino, target se mueve mientras estimás).
- DQN sobre CartPole y Atari.

Definiciones y características

- Q-value Q(s, a): expected return tras tomar a desde s.
- Bootstrap: update usando estimación actual (sin esperar fin de episodio).
- Off-policy: aprende sobre policy óptima aunque exploración sea ε-greedy.
- Replay buffer: almacena experiencias para sampleo iid.
- Target network Q': copia de Q usada para calcular el target. Actualizada cada N steps.
- ε-greedy: balance exploration vs exploitation.

Dataset / recursos

- FrozenLake (tabular).
- CartPole (DQN).
- Librerías: gymnasium, tensorflow, keras, numpy.

Ejercicios

1. Q-learning tabular: en FrozenLake, mantener Q[s, a] numpy array. Update con ε-greedy. Reportar success rate tras 5000 episodios.
2. DQN básico: red Dense(64) → Dense(64) → Dense(2) para CartPole. Sin replay buffer ni target network → ver inestabilidad.
3. Replay buffer: from collections import deque; buffer = deque(maxlen=10_000). Sample batch=32 random.
4. Target network: copiar Q.weights cada 100 steps a Q_target.
5. ε decay: empezar ε=1.0, decaer linealmente a 0.01 en 10 000 steps.

Homework verificable

DQN sobre CartPole-v1:

1. Red Dense(64, relu) → Dense(64, relu) → Dense(2).
2. Replay buffer 50 000, batch 64.
3. Target network sync cada 100 steps.
4. ε: linear decay 1.0 → 0.05 sobre 10 000 steps.
5. Train hasta mean_reward(100 episodios) ≥ 195.

Criterio de aceptación: alcanza ≥ 195 en ≤ 500 episodios; gráfico de return promedio muestra convergencia.

Errores comunes

Síntoma / mensaje	Causa y cómo arreglar
Q values explotan	Sin target network, bootstrap inestable. F
Convergencia lenta	Replay buffer muy chico o batch muy chico.
No explora → policy subóptima	ε muy chico desde el inicio. Fix: ε=1.0 al
done confundido con truncated	Gymnasium 5-tuple. Fix: tratar terminated,
OOM con Atari	Stacks de 4 frames + buffer enorme. Fix: r

Preguntas frecuentes

¿DQN supera Q-learning tabular cuándo?

Cuando state space es enorme (continuous o pixels). Para FrozenLake (16 states), tabular gana.

¿Variantes de DQN?

- Double DQN (van Hasselt 2016): reduce overestimación.
- Dueling DQN (Wang 2016): separa V y advantage.
- Prioritized Experience Replay: sampleo no uniforme.
- Rainbow (Hessel 2018): combina todas.

¿DQN vs Policy Gradient?

DQN: off-policy, sample-efficient, action space discreto. PG (PPO): on-policy, más estable, action space continuo. Ambos en el zoo moderno.

¿Por qué replay buffer rompe la correlación temporal?

Sin él, transitions consecutivas están correlacionadas → batches no iid → gradiente sesgado.

¿Cuál env empezar?

CartPole para entender. LunarLander para algo más complejo. Atari para serio (requiere días).

Referencias

- Géron, cap. 18 — Q-Learning y Deep Q-Learning.
- Watkins (1989), Learning from Delayed Rewards (PhD thesis).
- Mnih et al. (2015), Human-level control through deep reinforcement learning, Nature — DQN paper.
- van Hasselt et al. (2016), Deep Reinforcement Learning with Double Q-learning.

Siguiente clase

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

Apéndice: notebook (primer bloque)

Primera celda ejecutable del notebook de la clase.

```
# Imports y configuración inicial
```

Archivos complementarios

- notebook.ipynb