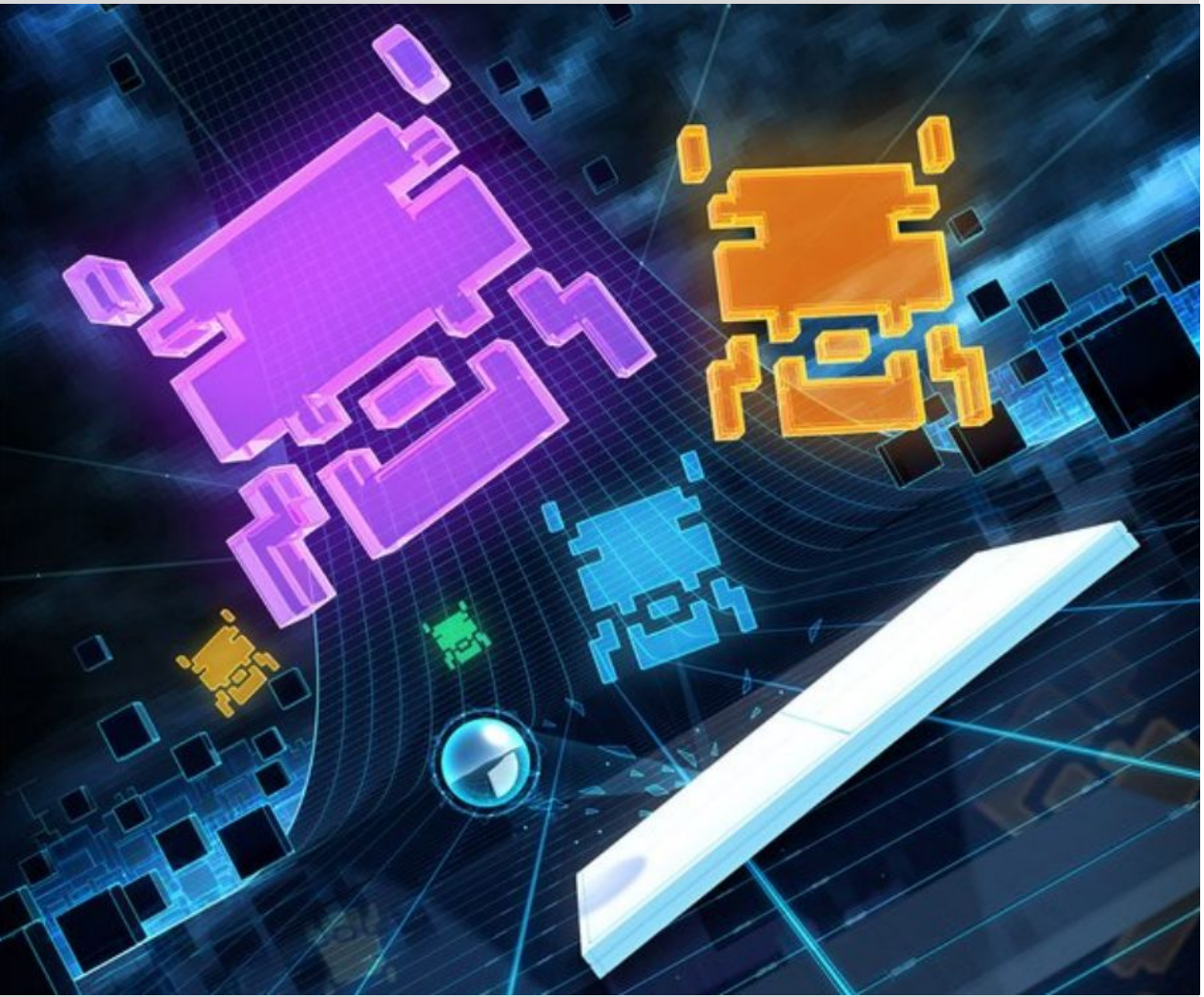


DigitalHouse > Coding School

INTELIGENCIA ARTIFICIAL

Aprendizaje por Refuerzo
2da Parte

OBJETIVOS DE LA CLASE

1. Solución usando el método Q-Learning
2. Solución usando el método Deep Q-Network (DQN)
3. Aplicaciones del Aprendizaje por Refuerzo

Función de Valor (value-function)

- El Aprendizaje por Refuerzo esta basado en las estimaciones de la funciones de valor.
- Data una política π y un estado *s*, denotamos como el **Retorno Esperado** cuando comienza en el estado *s* y sigue la política π.
- Se puede definir formalmente como:

$$V^{\pi}(s) = E_{\pi}\{ R_t \mid s_t = s\}$$

- La **función acción-valor Q(s,a)** es una función de valor.

Retorno descontado

- Vimos que para tareas episódicas, el retorno podía ser calculado como:

$$G_t = r_{t+1} + G_{t+1}$$

- Sin embargo cuando las tareas son continuas, podríamos usar la fórmula del retorno descontado (discounted return):

$$G_t = r_{t+1} + \gamma G_{t+1}$$

Con **gamma(γ)** entre 0 y 1. Observar que dependiendo el valor de gamma, los retornos futuros van a tener mayor o menor peso.

Monte Carlo Constante-alpha

- Se le llama así a una variante de Monte Carlo que es útil en problemas de baja predictibilidad, o sea, con alta variación,

$$Q(s,a) = Q(s,a) + \alpha(G_t - Q(s,a))$$

- Para calcular el nuevo valor del retorno estimado, compara el valor estimado actual con el valor de retorno estimado futuro. Lo suma usando el **parámetro alfa (α)** para indicar el peso que va a tener, es decir, cuánto se va a parecer a G_t
- **alfa (α)** puede valer entre 0 y 1

Uniendo las fórmulas

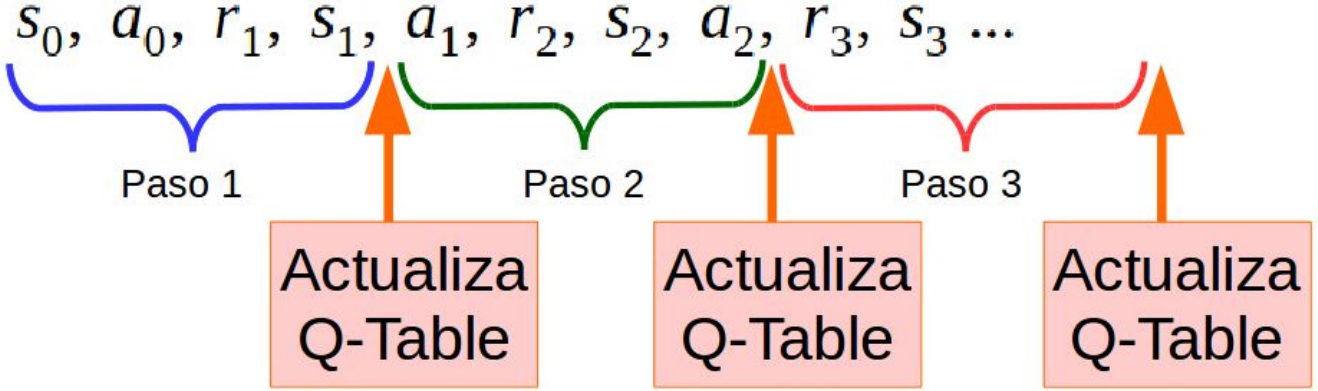
- Podemos entonces estimar el valor de retorno futuro para un estado y una acción dada como:

$$Q(s,a) = Q(s,a) + \alpha((r_{t+1} + \gamma\, Q(s_{t+1}, a')) - Q(s\,, a))$$

- Donde **alpha(α)** nos va a indicar que tan gradual va a ser la actualización
- Y **gamma(γ)** va a indicar cuánto usamos de las experiencias futuras, el peso de los futuros retornos.
- Falta ver cómo elegimos la acción a'

Q-Learning

- Pertenece a la clase de algoritmos de **Diferencia Temporal** (Temporal Difference - TD Learning)
- Utiliza una Q-table que va a ser actualizada después de cada paso (step) de todos los episodios *(bootstrap)*
- Útil para tareas continuas (no episódicas) y episódicas.



Q-Learning Algoritmo

DigitalHouse > Coding School

Q-Learning (Sarsamax)

Inicializa tabla Q(s,a) con valores arbitrarios

Repite durante varios episodios e:

Por cada paso t del episodio e:

Elige una acción a con s y una política que use Q (ε-greedy)

Toma la acción a, observa r, s'

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

s = s'

Actividad 1:

Q-Learning

Aprendizaje por Refuerzo Profundo

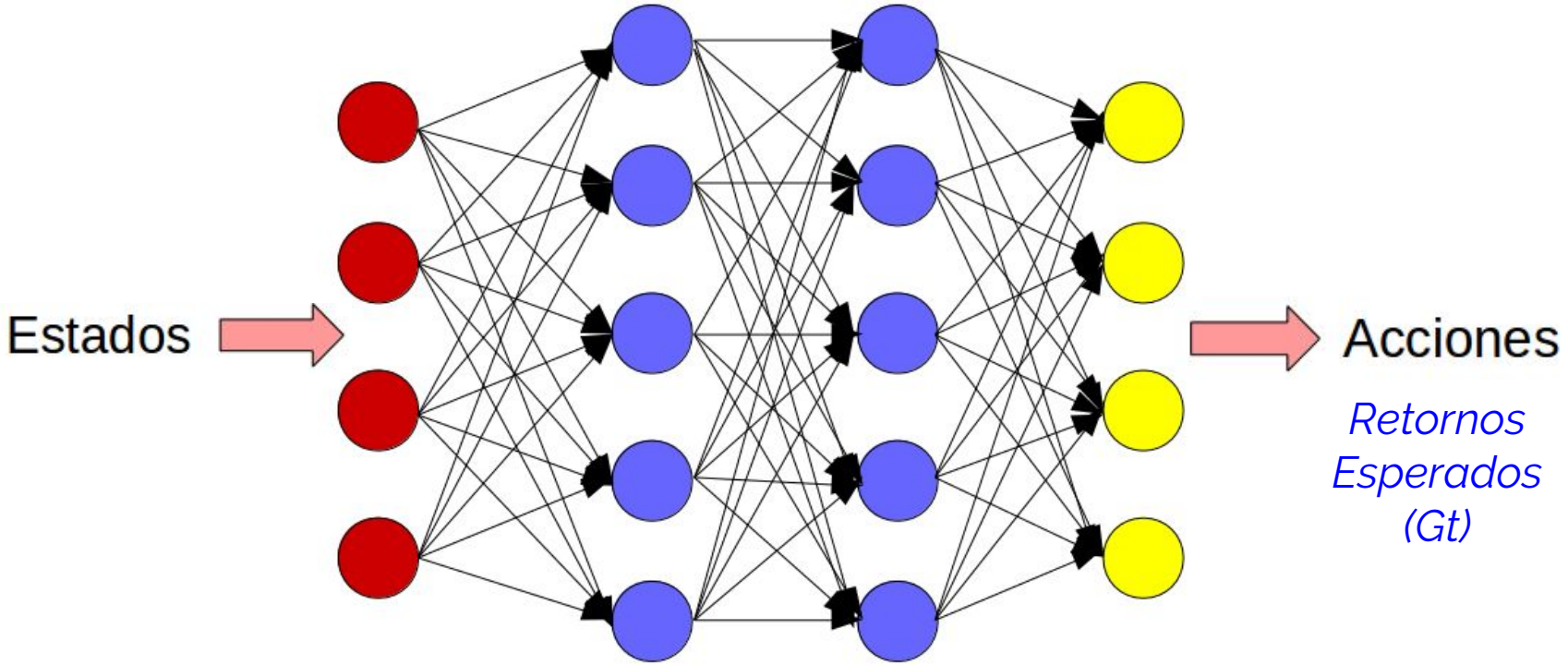
- Área del Aprendizaje por Refuerzo que utilizar Redes Neuronales para determinar la acción a tomar.
- Tapa de la revista Nature sobre la publicación: Playing Atari with Deep Reinforcement Learning (2013 DeepMind)
- AI juega con nivel humano o superior a 49 juegos de la Atari 2600

https://storage.googleapis.com/deepmind-media/dqn/DQNNaturePaper.pdf

Deep Q-Network DQN

- Utiliza una Red Neuronal para implementar la función de valor
- Es inestable. Hay dos estragias para garantizar estabilidad:
    - Repetición de Experiencia (Experience Replay)
    - Q-Targets Fijos (Fixed Q-Targets)

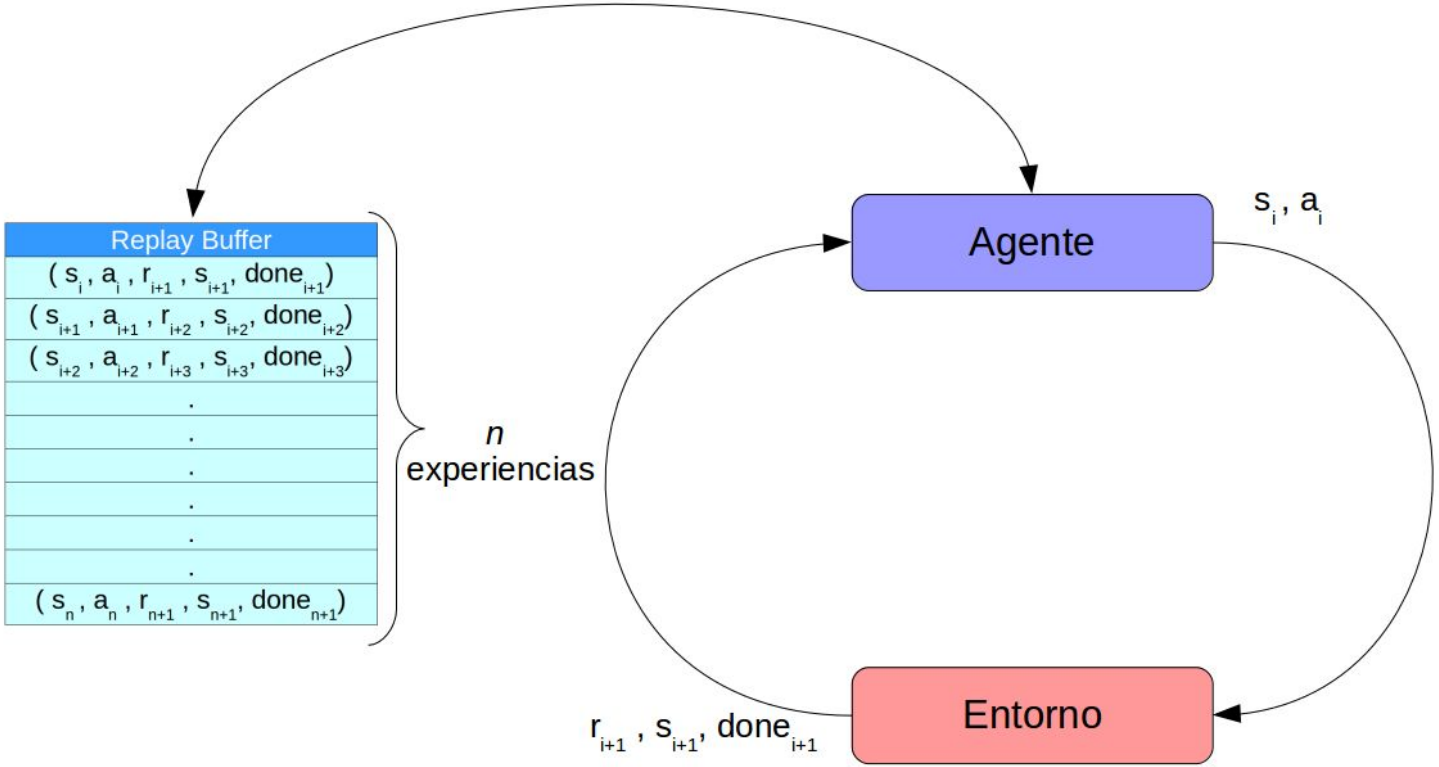
Deep Q-Network (DQN)



Repetición de Experiencia (Experience Replay)

- Consiste en ir **recolectando las experiencias pasadas**: (estado, acción, retorno, estado siguiente, terminó) para ser usadas durante el entrenamiento.
- Este buffer es llamado “Replay Buffer” o “Replay Memory”.
- Para entrenar, se toma una **muestra aleatoria de este buffer** con la idea es tener una variedad de casos y de **sucesos** que fueron ocurriendo de **manera no correlacionada**.

Repetición de Experiencia



DQN Algoritmo

Inicializa la memoria ReplayBuffer vacio

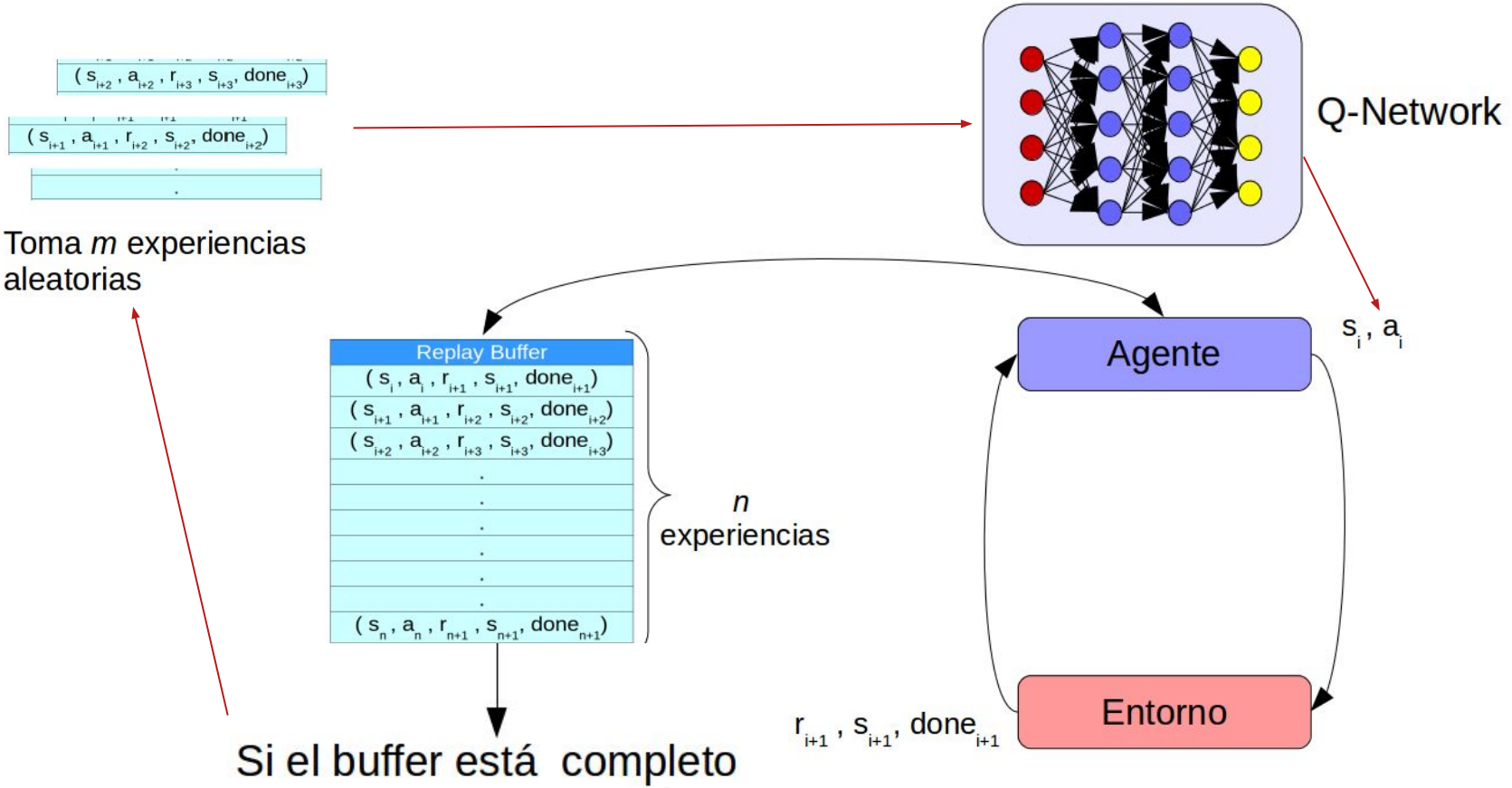
Inicializa la Q-Network

Por cada episodio:

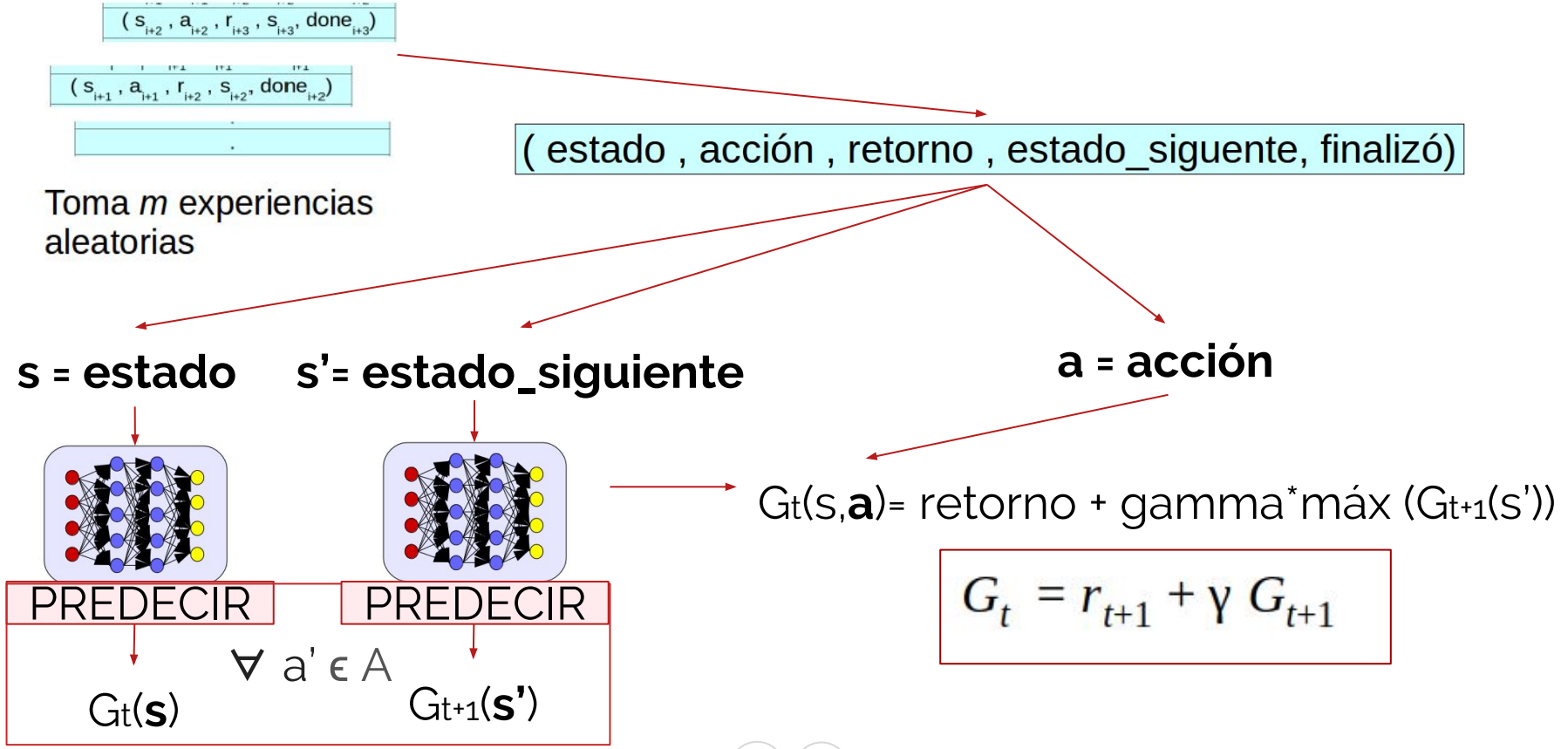
Por cada paso:

1) Toma acciones acorde a la política π y recolecta experiencias (s,a,r+1,s+1,done) en el ReplayBuffer
2) Si tiene suficientes experiencias (>= batch_size)
   Selecciona aleatoriamente batch_size experiencias.
   Por cada experiencia:
   1. Gt esperada <- Recalcula Gt para la acción
   2. Re-entrena la Q-Network usando Gt esperada

DQN - Repetición de Experiencias



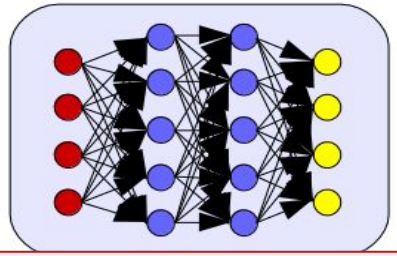
DQN - Cálculo del Retorno Estimado



DQN - Actualización de Q-Network

DigitalHouse > Coding School

s = estado



PREDECIR

Gt(**s**) ∀ a'ϵA

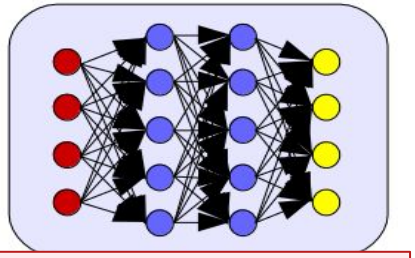
y(a')=Gt(s,a') ∀ a'ϵA: **a' ≠ a**

y(a)=retorno+γ*máx (Gt+1(s'))

Input: x

Label: y

Re-entrena la Q-Network para acercarse a los nuevos valores de Gt



FIT(x,y)

Actividad 2:

DQN

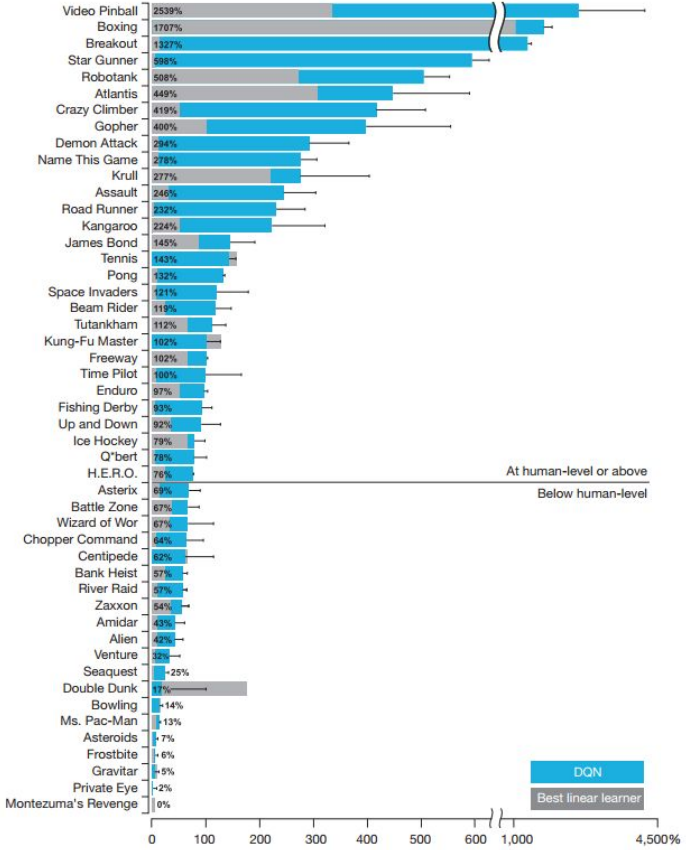
Fixed Q-Targets

- Consiste en tener una segunda Q-Network target que se va a ir actualizando periódicamente con la Q-Network original.
- La Q-Network target se va a usar como lectura. Cada vez que ese quieran consultar los valores de la función Q.
- Esto hace que la función Q no esté cambiando continuamente con la variación de las muestras, volviéndose más estable al consultarla.

Aplicaciones del Aprendizaje por Refuerzo

Investigación

- Máquinas que juegan a videos juegos con el mismo nivel de juego que los humanos
- Atari 2600 videogames



Investigación

- Juegos en los cuales las máquinas vencen a los humanos

Investigación

- Aprendizaje en Robots
- Manejo de Drones
- Predecir valores de las acciones y ganancias en tiempo real
- Control de los semáforos
- Recomendaciones de noticias

Recursos

- DQN Paper
- Publicación en Nature DQN: Human-level control through deep reinforcement learning
- AlphaGo Zero: Starting from scratch
- Video - David Silver: Deepmind AlphaZero - Mastering Games Without Human Knowledge
- Video - Siraj Raval: Reinforcement Learning for Stock Prediction (principiante)