Taller

Deep Reinforcement Learning

ACTUMLOGOS

DESARROLLANDO HABILIDADES TECNOLÓGICAS

Objetivo

- Taller de introducción a aprendizaje por refuerzo.

- El objetivo es entender de forma intuitiva los conceptos básicos que existen detrás de este paradigma de machine learning.

- Está dirigido a todos, preferiblemente con conocimientos en:
  - Machine Learning
  - Deep Learning
  - Programación en Python
  - TensorFlow2
  - Matemáticas de bachillerato

Agenda

- Motivación
- Introducción
- Aprendizaje Reforzado
- OpenAI Gym
- Deep Learning
- Deep Reinforcement Learning
- Deep Q-Network

Motivación

ACTUMLOGOS

DESARROLLANDO HABILIDADES TECNOLÓGICAS

Jugando Atari. DeepMind, 2013

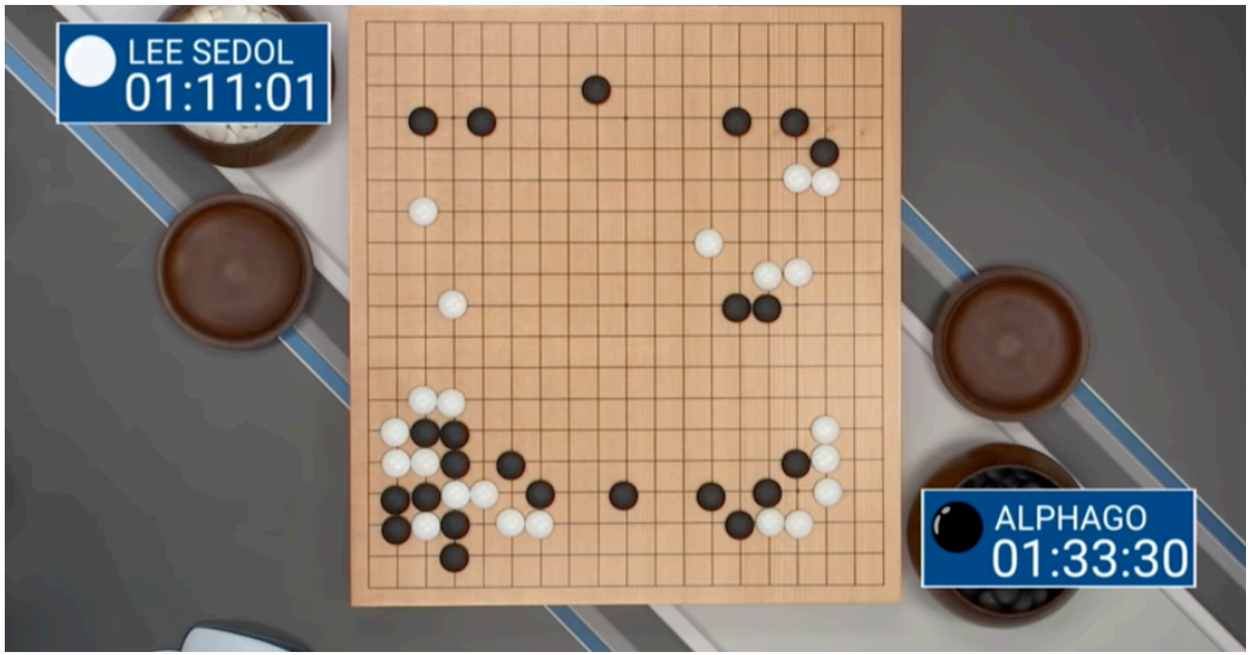
Unos minutos después

Un par de horas después

Unas cuantas horas después

AlphaGo. DeepMind, 2016

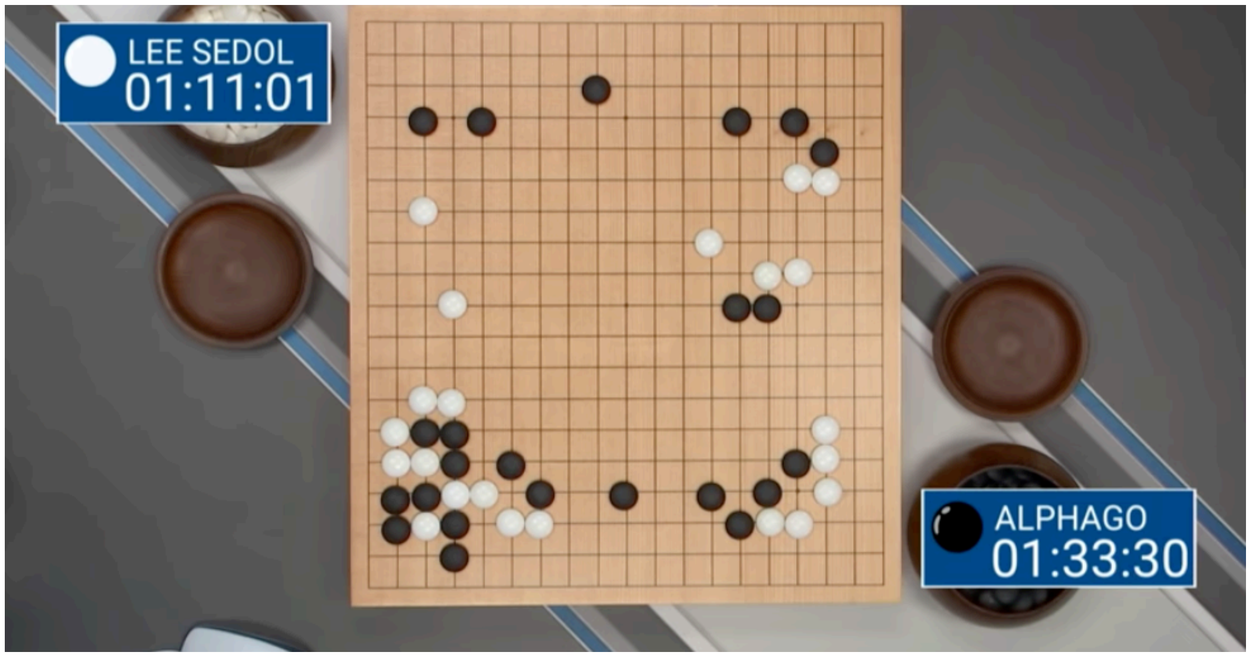
- 10^{170} posibles configuraciones de juego.
- AlphaGo - The Movie



Imágenes: https://www.youtube.com/watch?v=WXuK6gekU1Y

AlphaGo. DeepMind, 2016

- AlphaGo Zero
- Alpha Zero



Imágenes: https://www.youtube.com/watch?v=WXuK6gekU1Y

OpenAi Five. OpenAi, 2018

Dota 2

- 180 años de juegos contra sí mismo todos los días usando 256 GPUs y 128000 CPU núcleos.

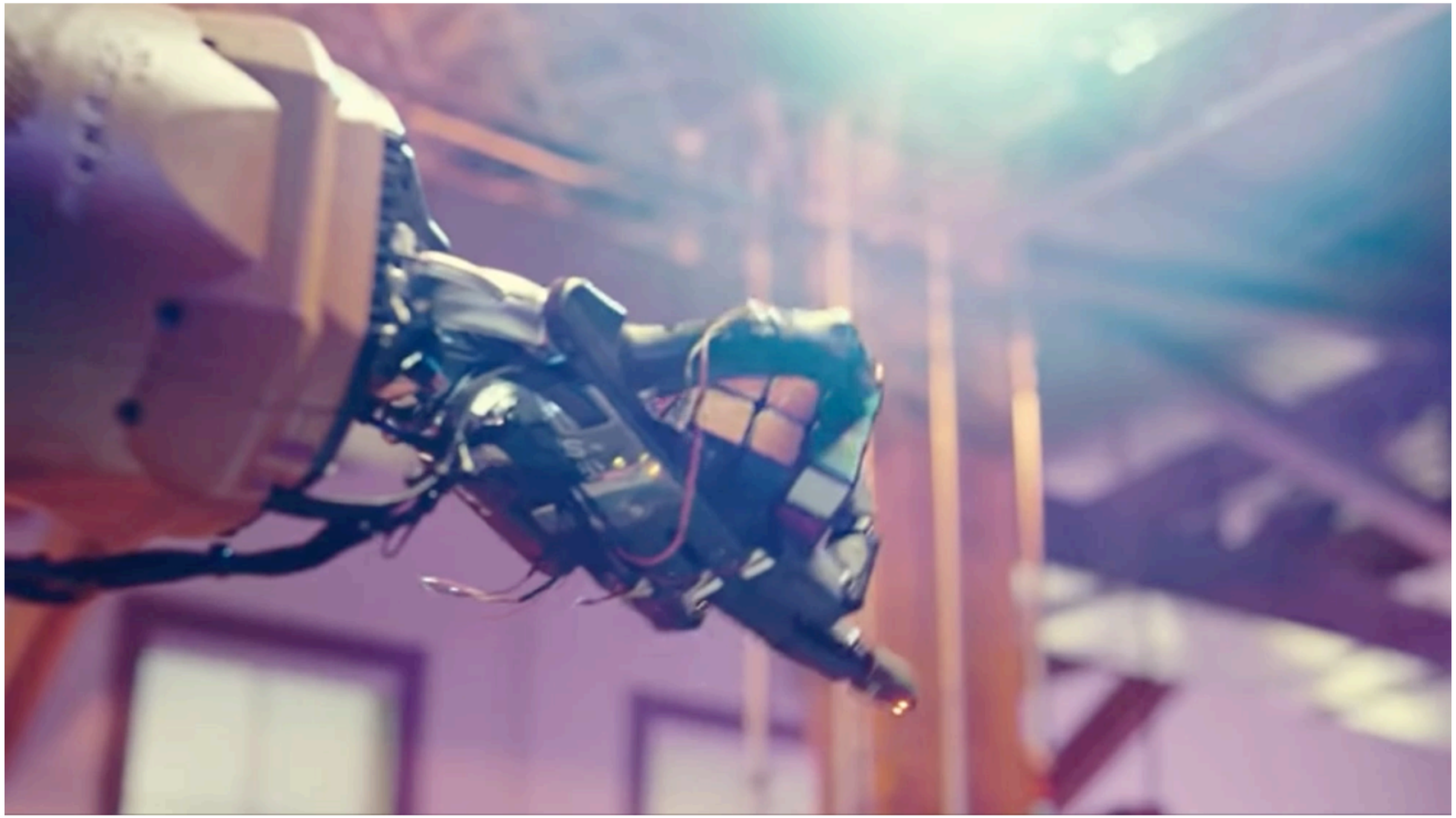
Video: https://openai.com/blog/openai-five/

AlphaStar. DeepMind, 2019

StarCraft II

Video: https://deepmind.com/blog/article/alphastar-mastering-real-time-strategy-game-starcraft-ii

Resolviendo Cubo Rubik. OpenAi, 2019



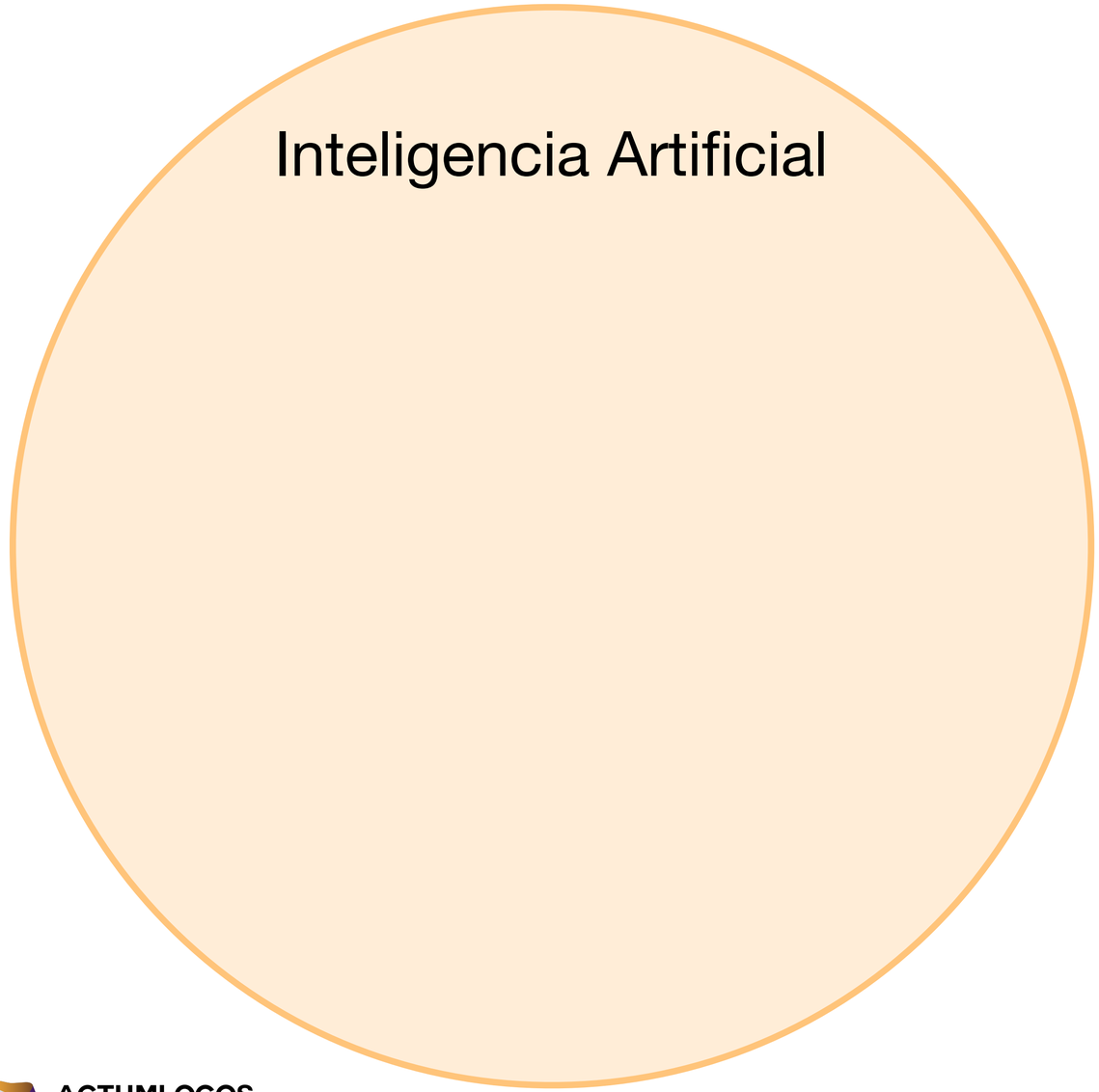
Video: https://openai.com/blog/solving-rubiks-cube/

Introducción

ACTUMLOGOS

DESARROLLANDO HABILIDADES TECNOLÓGICAS

Introducción

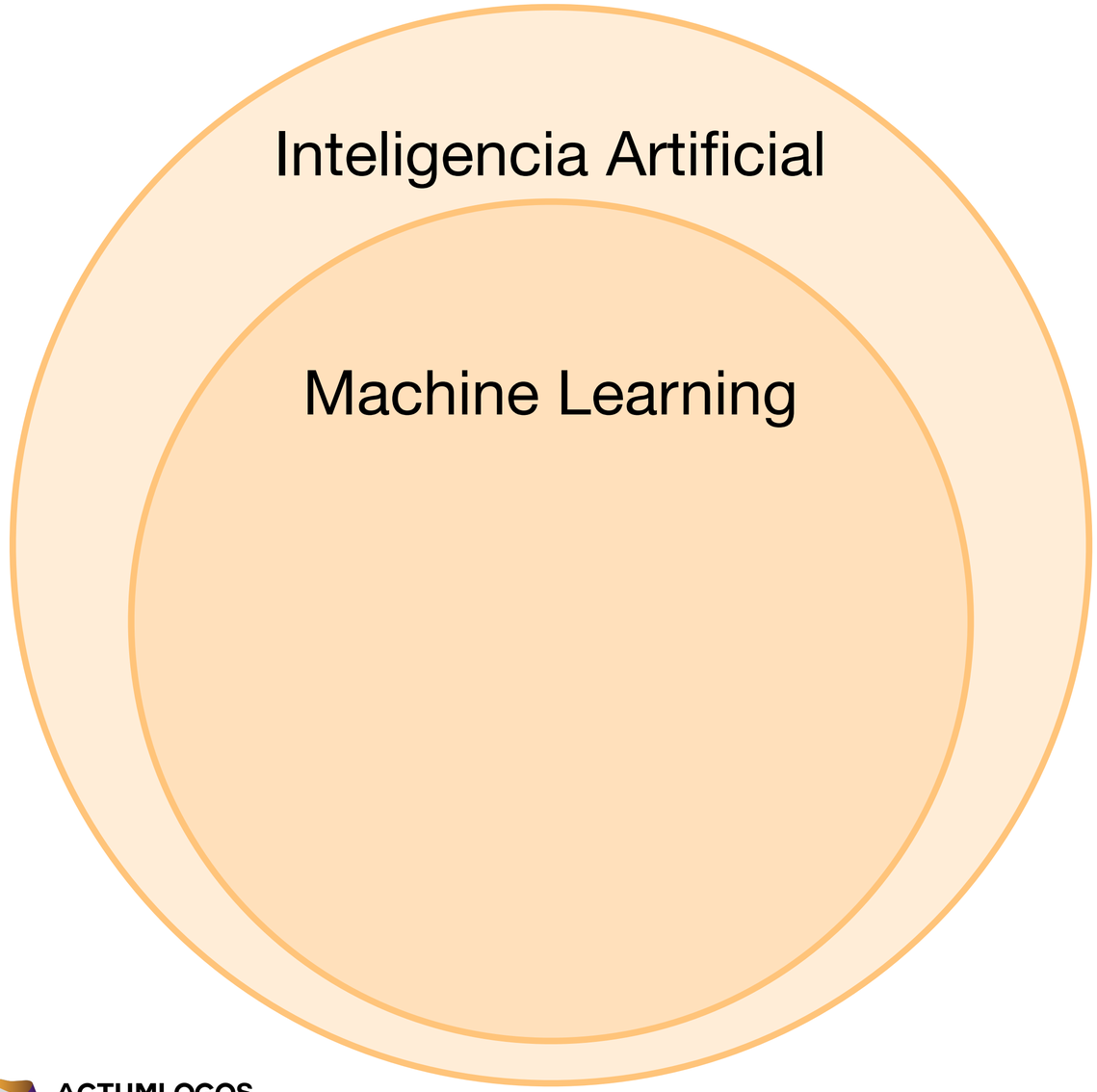


Inteligencia Artificial

John McCarthy en 1955 la definió como "la ciencia y la ingeniería de hacer máquinas inteligentes". Muchos de los sistemas computacionales considerados como IA requieren humanos que programen las máquinas para que se comporten de una manera inteligente, como jugar ajedrez, pero hoy, hacemos hincapié en las máquinas que pueden aprender, al menos algo parecido a lo que hacen los seres humanos

La IA débil son sistemas inteligentes que resuelven problemas específicos (reconocimiento facial o de voz). La IA a nivel humano o IA general (AGI), busca ampliar la inteligencia a máquinas adaptables al contexto. Esto es necesario para que los chatbots sociales o la interacción humano-robot sean más eficaces.

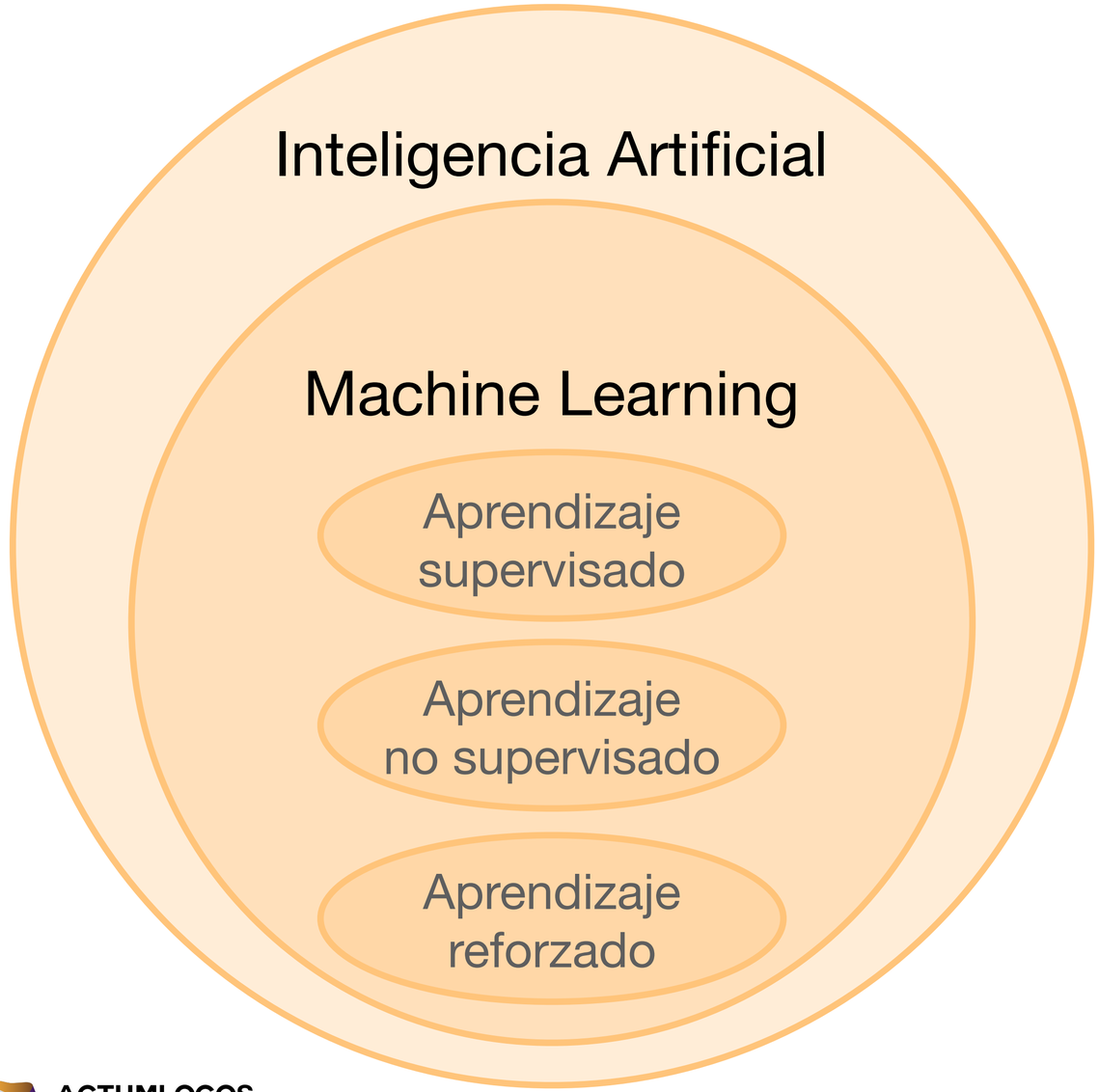
Introducción



Machine learning (**ML**) is the study of computer algorithms that can improve automatically through experience and by the use of data.[1] It is seen as a part of artificial intelligence. Machine learning algorithms build a model based on sample data, known as "training data", in order to make predictions or decisions without being explicitly programmed to do so.[2] Machine learning algorithms are used in a wide variety of applications, such as in medicine, email filtering, speech recognition, and computer vision, where it is difficult or unfeasible to develop conventional algorithms to perform the needed tasks.[3]

WIKIPEDIA
The Free Encyclopedia

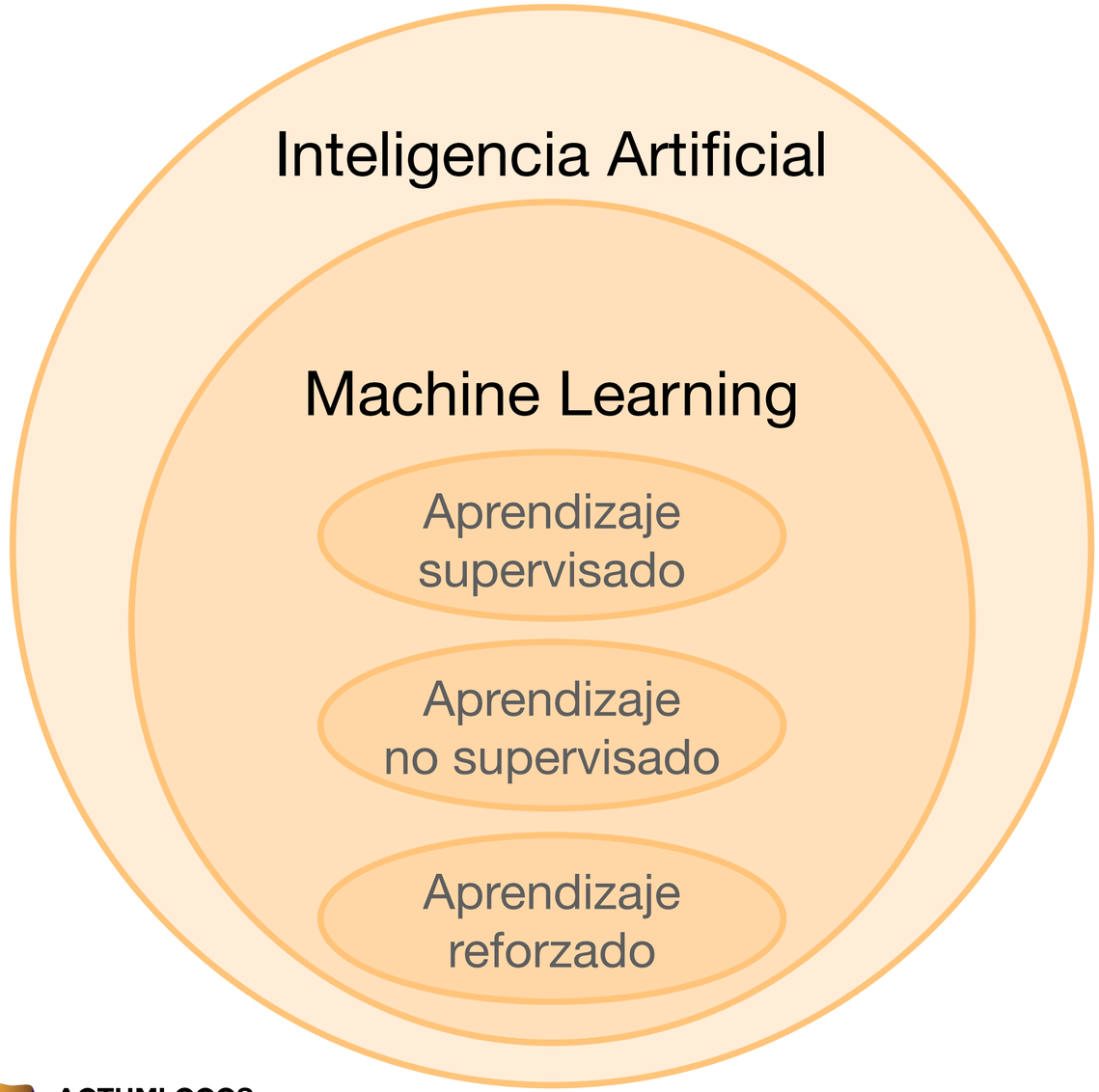
Introducción



Aprendizaje supervisado

Usamos datos etiquetados para hacer predicciones.

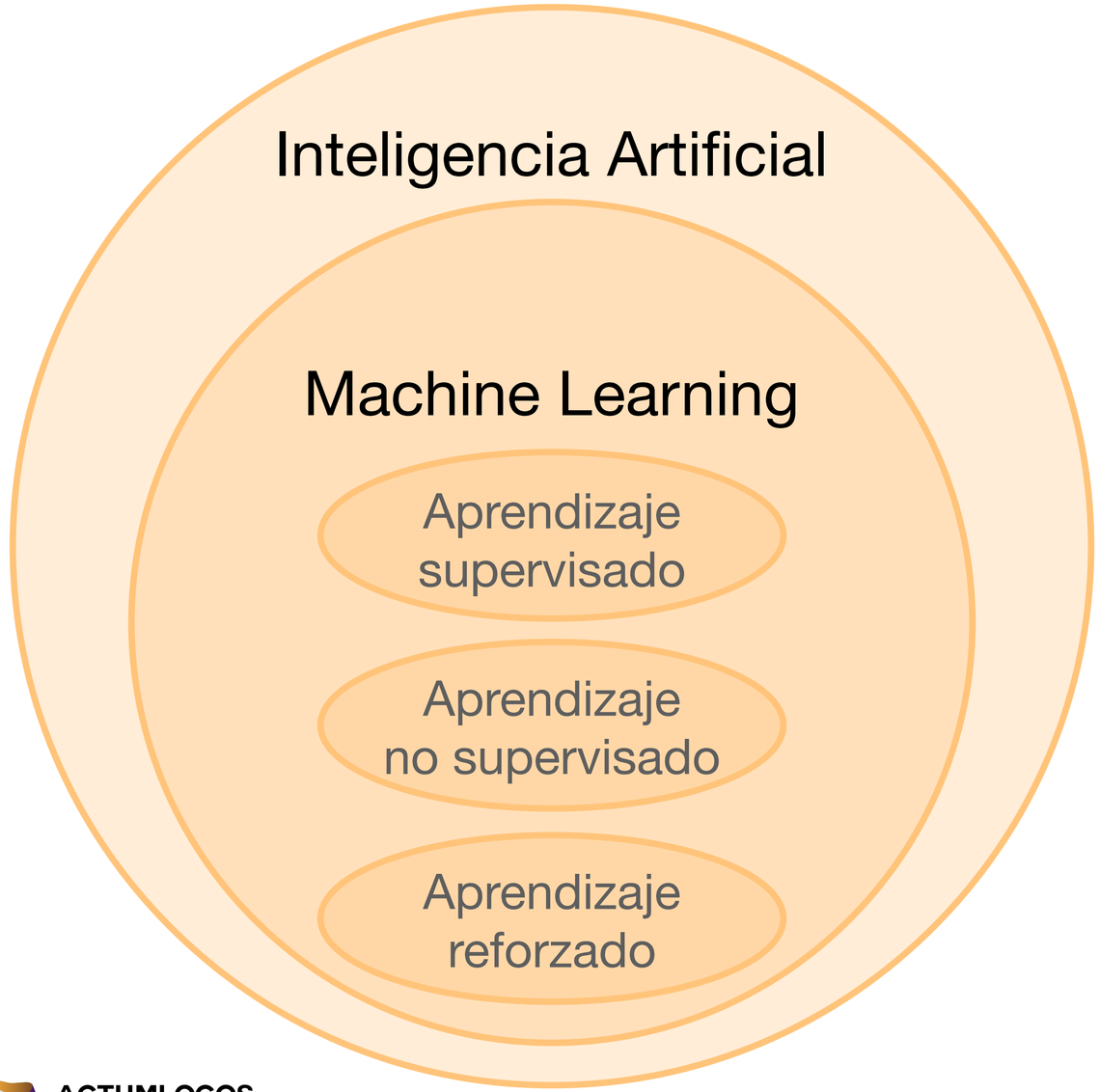
Introducción



Aprendizaje no supervisado

Usamos datos no etiquetados para buscar patrones en ellos.

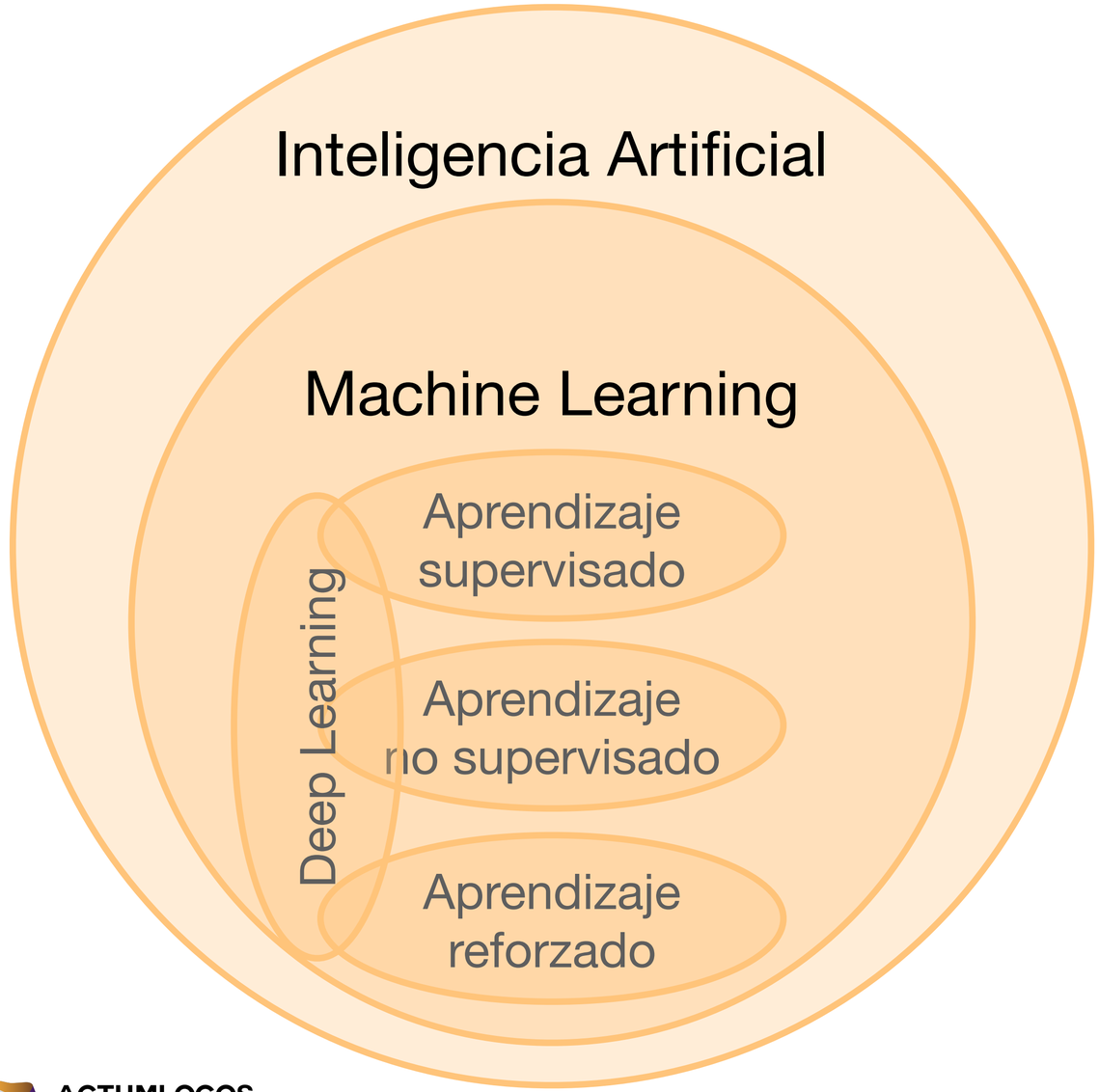
Introducción



Aprendizaje reforzado

No usamos un conjunto de datos como tal, en vez de ello tenemos un sistema con un agente y un ambiente, donde el agente se entrena a sí mismo interactuando con el ambiente para resolver la tarea en cuestión.

Introducción



Deep Learning

Familia de métodos basados en redes neuronales.

Aprendizaje Reforzado

ACTUMLOGOS

DESARROLLANDO HABILIDADES TECNOLÓGICAS

Introducción

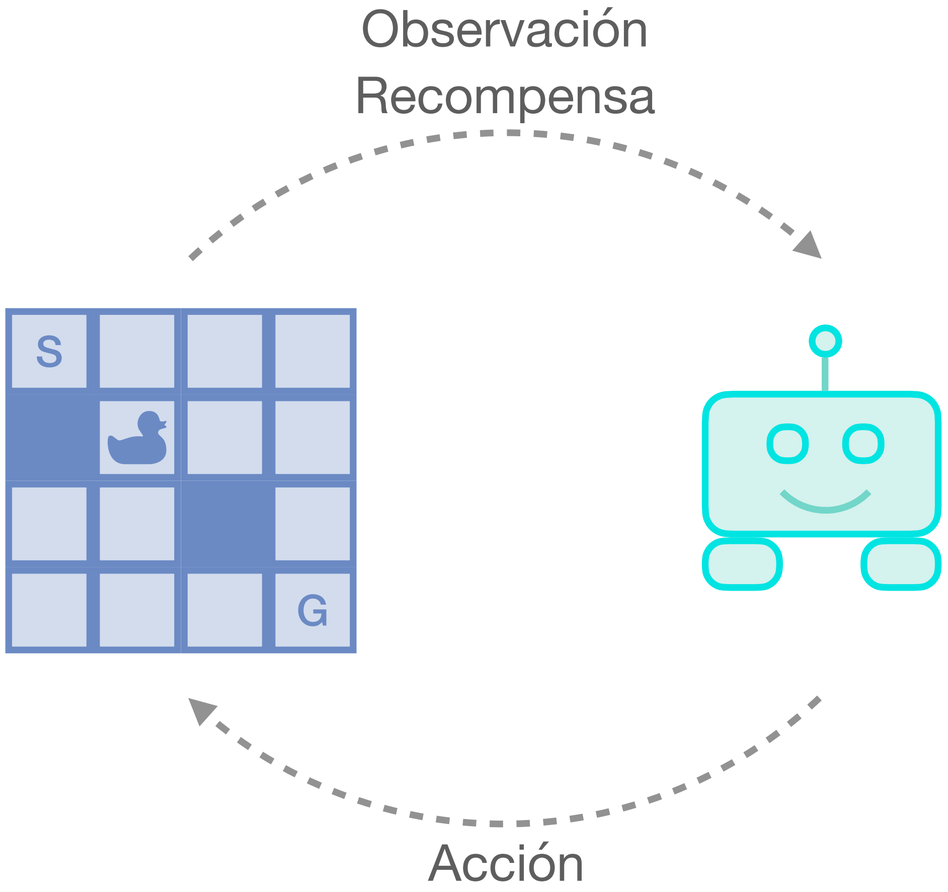
- Reinforcement learning opera de forma diferente a aprendizaje supervisado o no supervisado.

- En vez de depender de una conjunto de datos etiquetados o no etiquetados, RL usa un agente para interactuar con un ambiente donde, a través de repetidos intentos recompensados, el agente logre aprender un determinado comportamiento dentro del ambiente.

Objetivo de RL

- Entrenar un agente para que se capaz de llevar a cabo una tarea dentro de un ambiente.
  - Ganar StarCraft II
  - Controlar una mano robótica para resolver un cubo Rubik

Ciclo de RL



Agente

Es el algoritmo que interactúa con el ambiente:

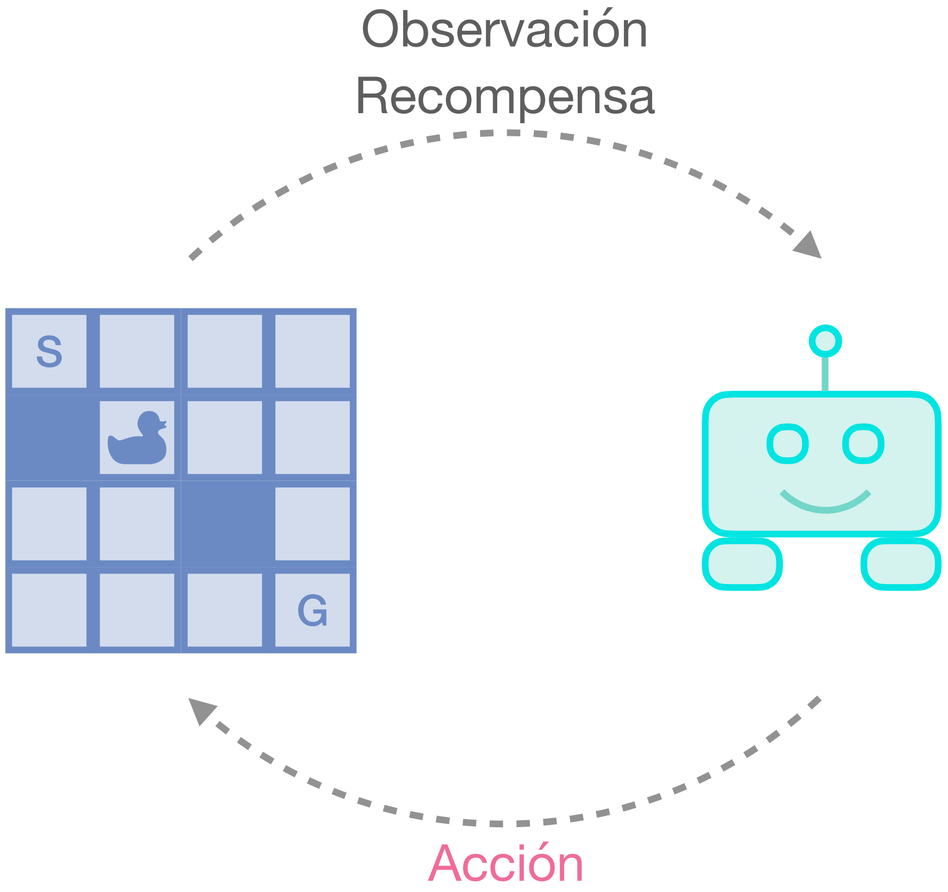
- Observa al ambiente.
- Toma acciones dentro del ambiente.

Ambiente

Todo lo que no es el agente:

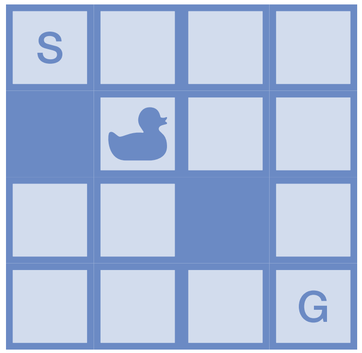
- Es afectado por las acciones del agente.
- Cambia su estado y genera una recompensa y observación.

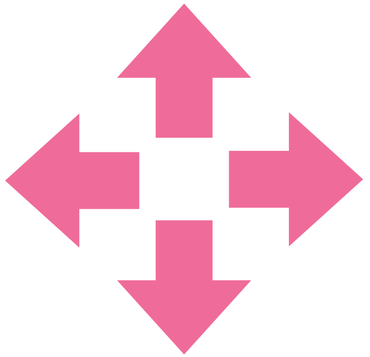
Ciclo de RL



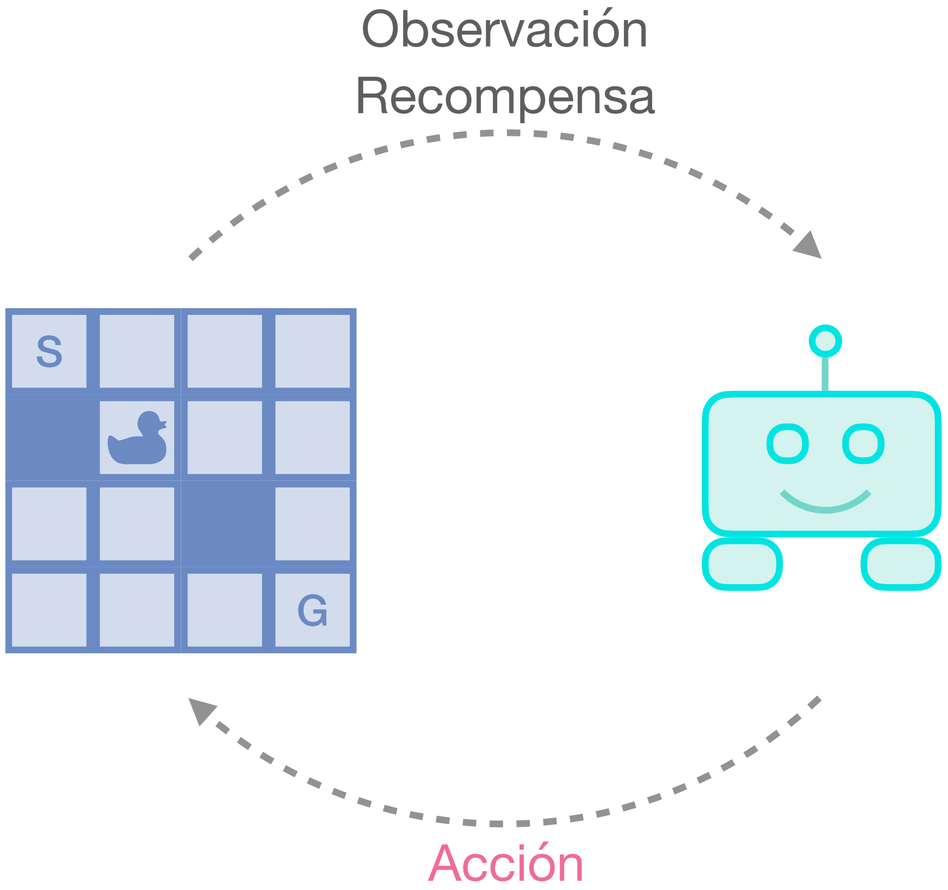
Acción (a)

Todos los ambientes tienen un conjunto de acciones disponibles que el agente puede tomar.



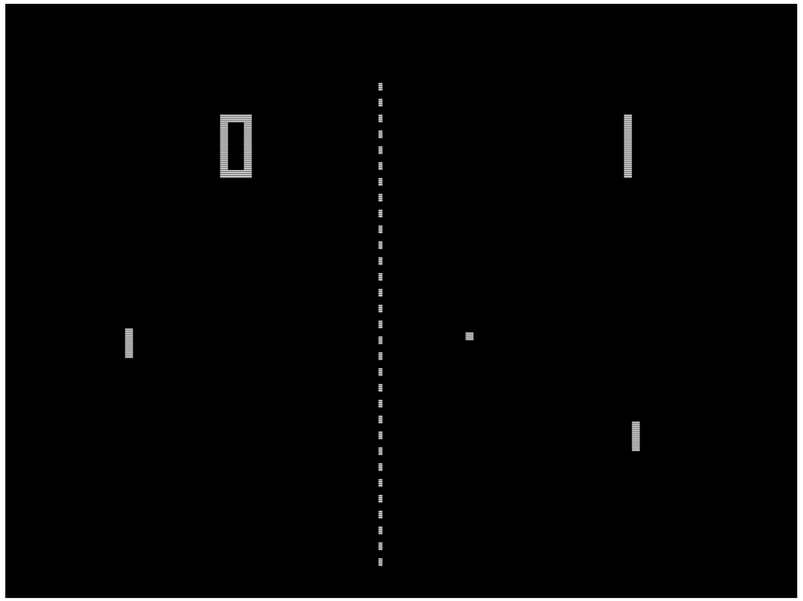


Ciclo de RL

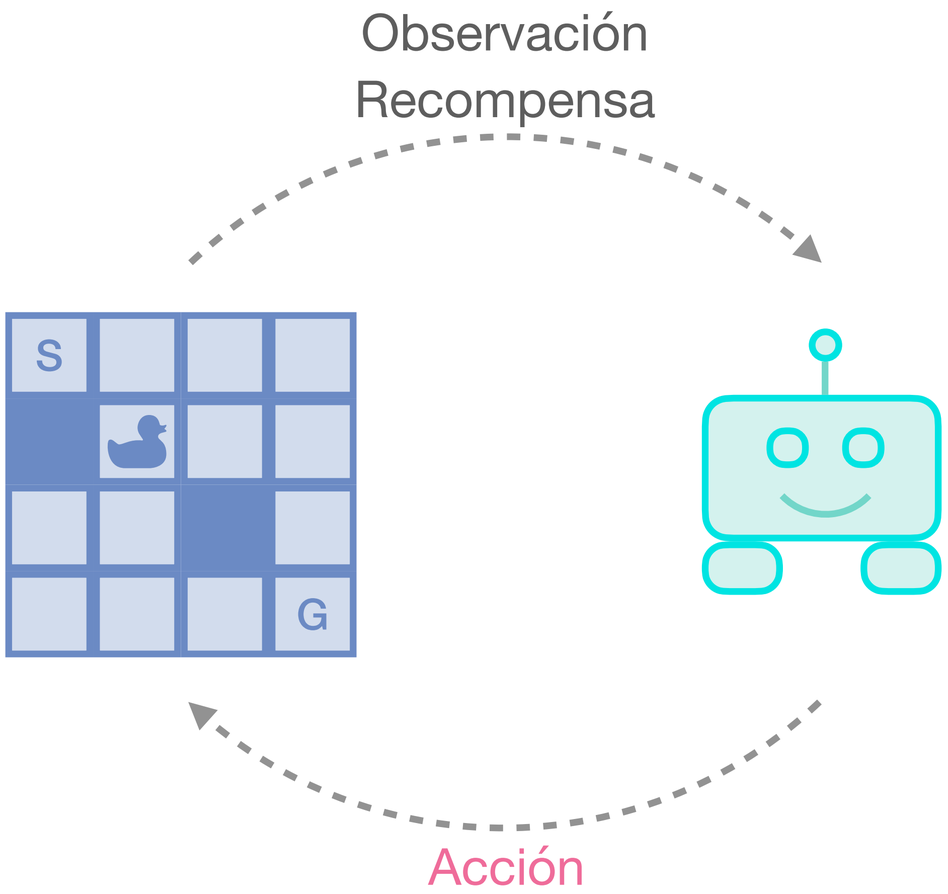


Acción (a)

Todos los ambientes tienen un conjunto de acciones disponibles que el agente puede tomar.



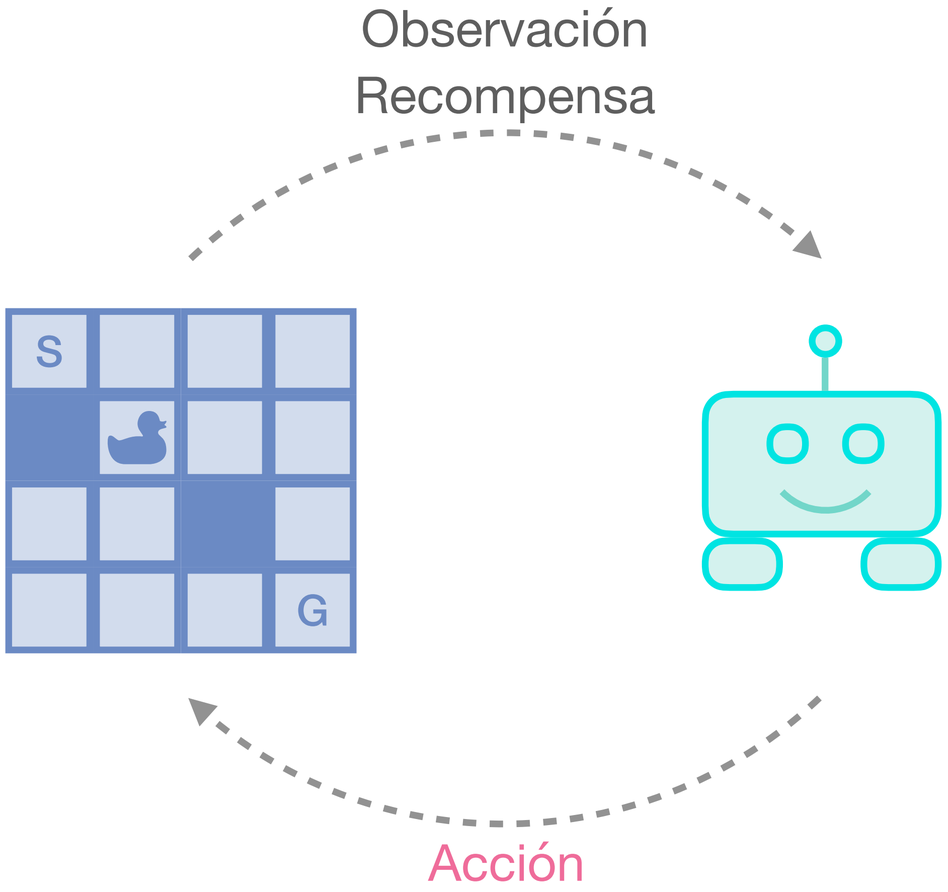
Ciclo de RL



Acción (a)

Todos los ambientes tienen un conjunto de acciones disponibles que el agente puede tomar.

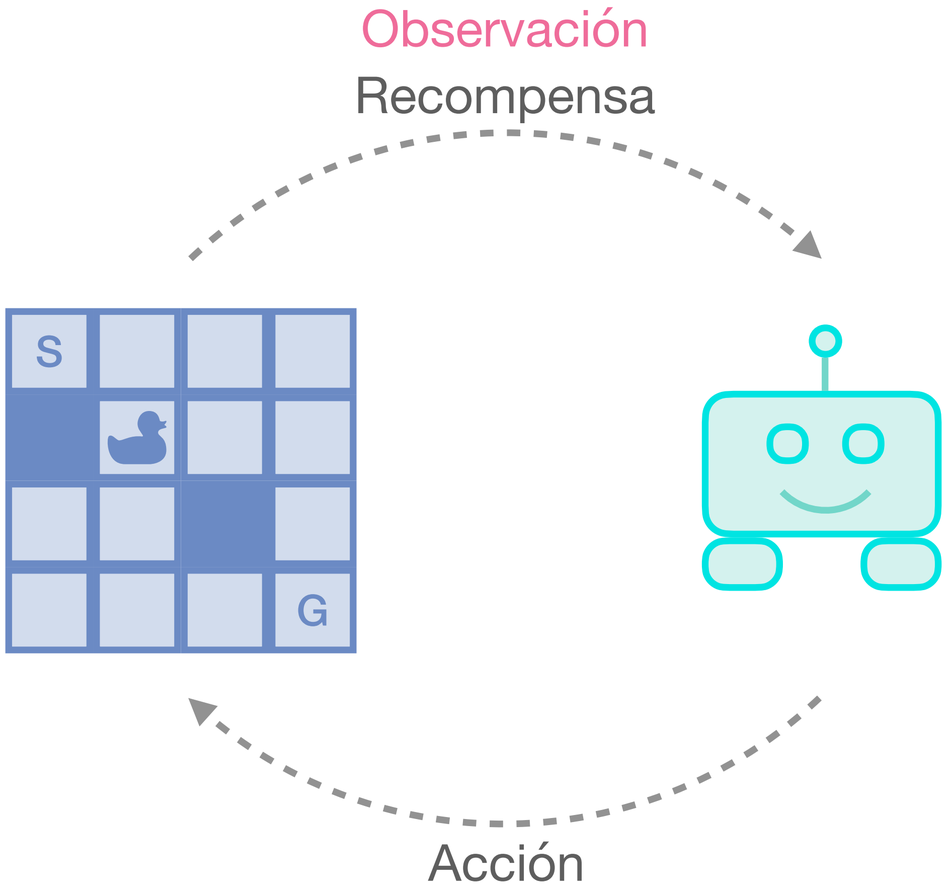
Ciclo de RL



Acción (a)

Todos los ambientes tienen un conjunto de acciones disponibles que el agente puede tomar.

Ciclo de RL

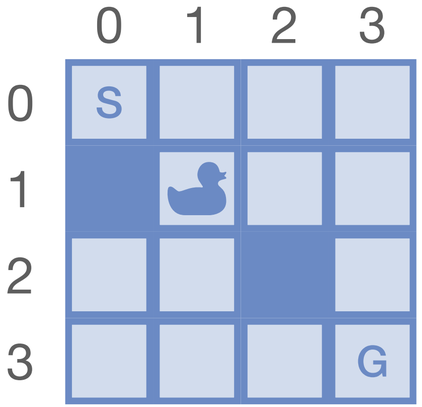


Estado (s)

Es una descripción completa del ambiente en un determinado paso de tiempo t.

Observación

Es la descripción del ambiente a la que el agente tiene acceso. Puede ser parcial.

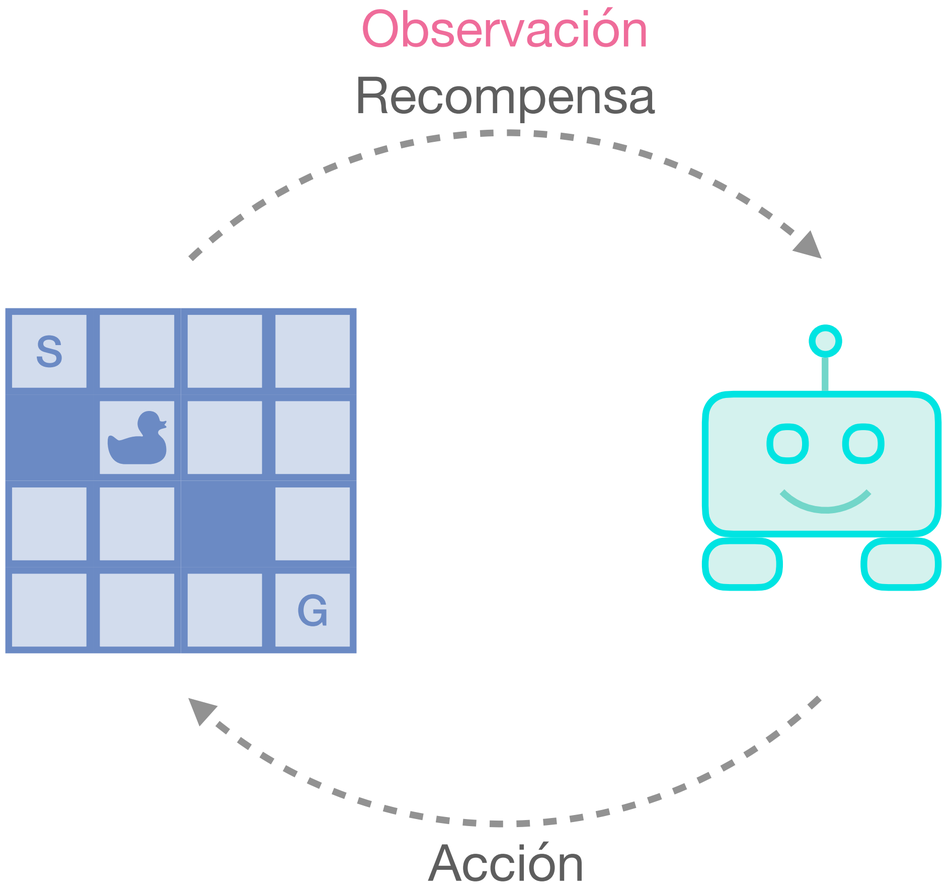


(1)

(1,1)

Estado Observación Observación

Ciclo de RL

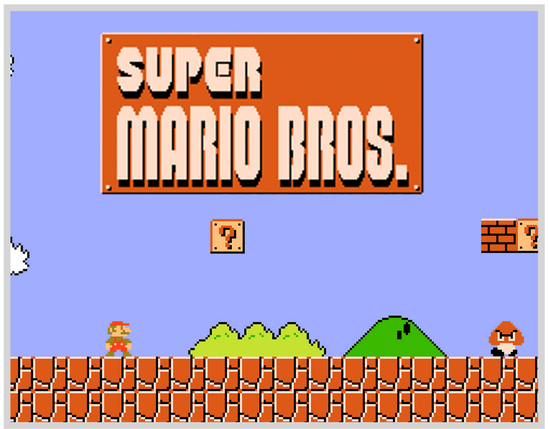


Estado (s)

Es una descripción completa del ambiente en un determinado paso de tiempo t.

Observación

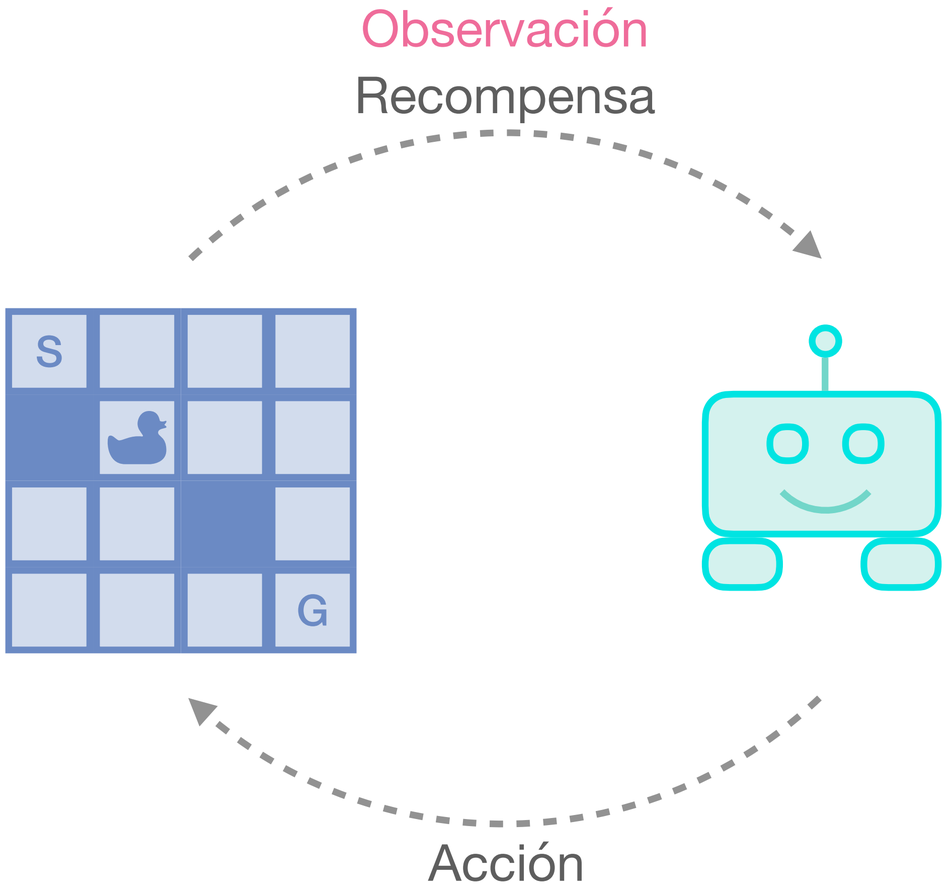
Es la descripción del ambiente a la que el agente tiene acceso. Puede ser parcial.



Estado

Observación

Ciclo de RL

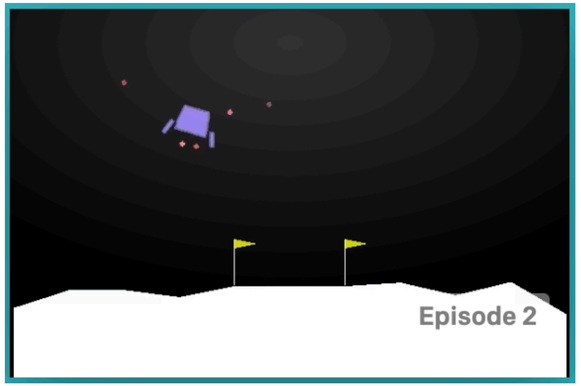


Estado (s)

Es una descripción completa del estado en que se encuentra el ambiente en un determinado paso de tiempo t.

Observación

Es la descripción del ambiente a la que el agente tiene acceso. Puede ser parcial.

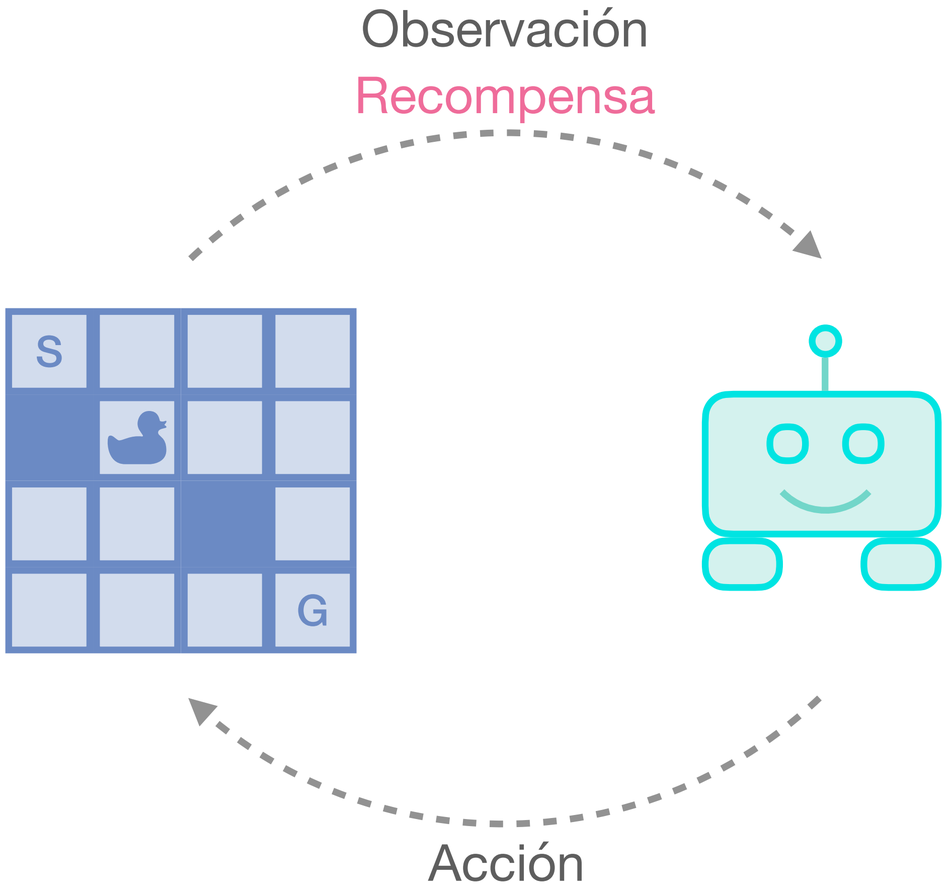


[-1.3, 2.4, 1.2, 0.7, 81, 0.23]

Estado

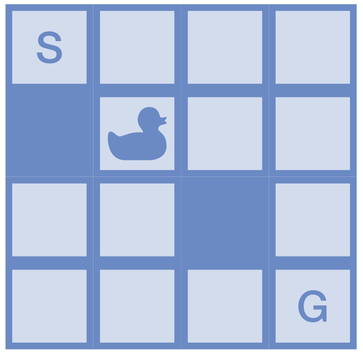
Observación

Ciclo de RL



Recompensa (r)

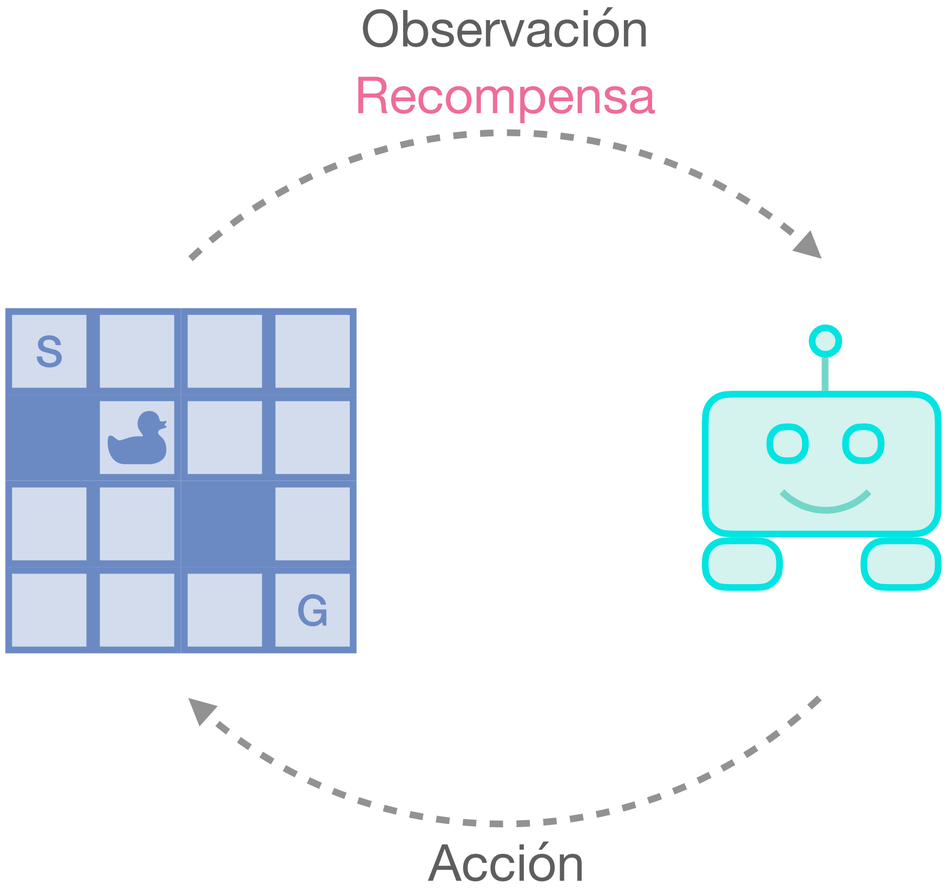
Retroalimentación numérica que el ambiente proporciona como consecuencia de las acciones tomadas por el agente.



r=0 en todos los estados menos en Goal donde r=1

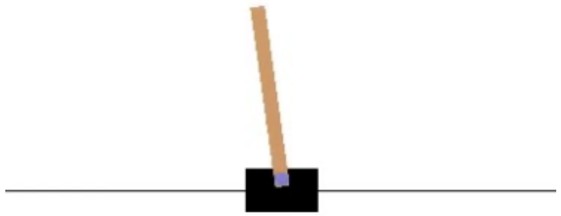
Recompensa

Ciclo de RL



Recompensa (r)

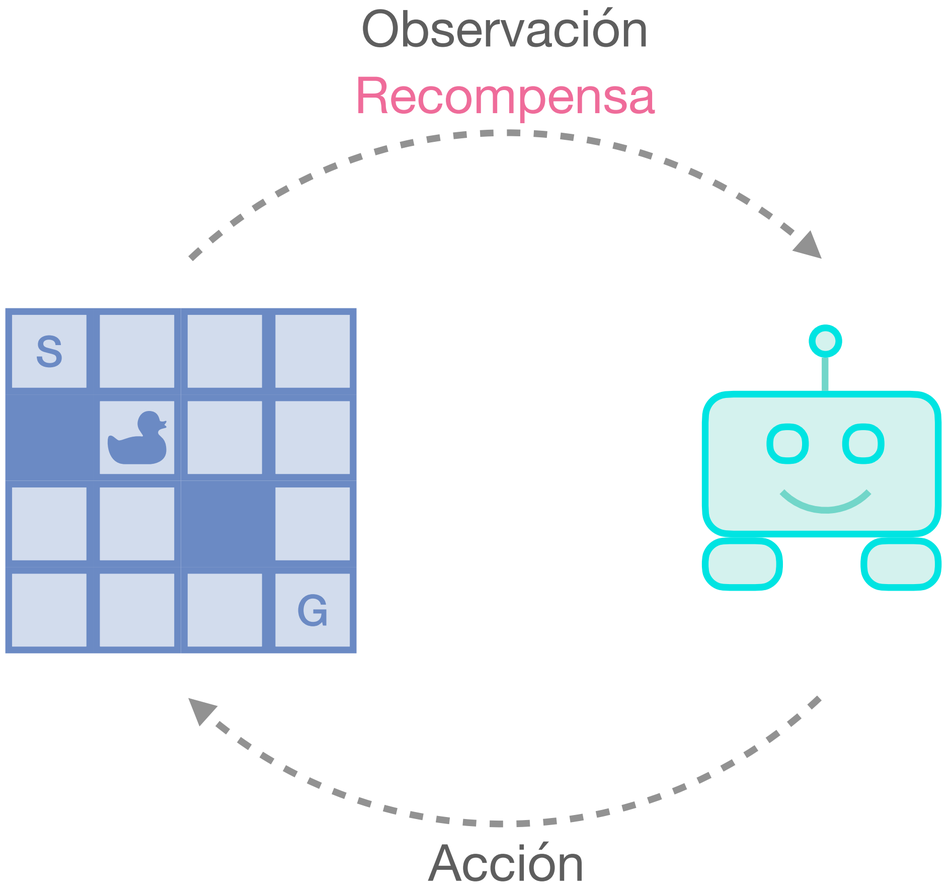
Retroalimentación numérica que el ambiente proporciona como consecuencia de las acciones tomadas por el agente.



r=+1 cada paso de tiempo, siempre que el péndulo no rebase los 15º

Recompensa

Ciclo de RL



Recompensa (r)

Retroalimentación numérica que el ambiente proporciona como consecuencia de las acciones tomadas por el agente.

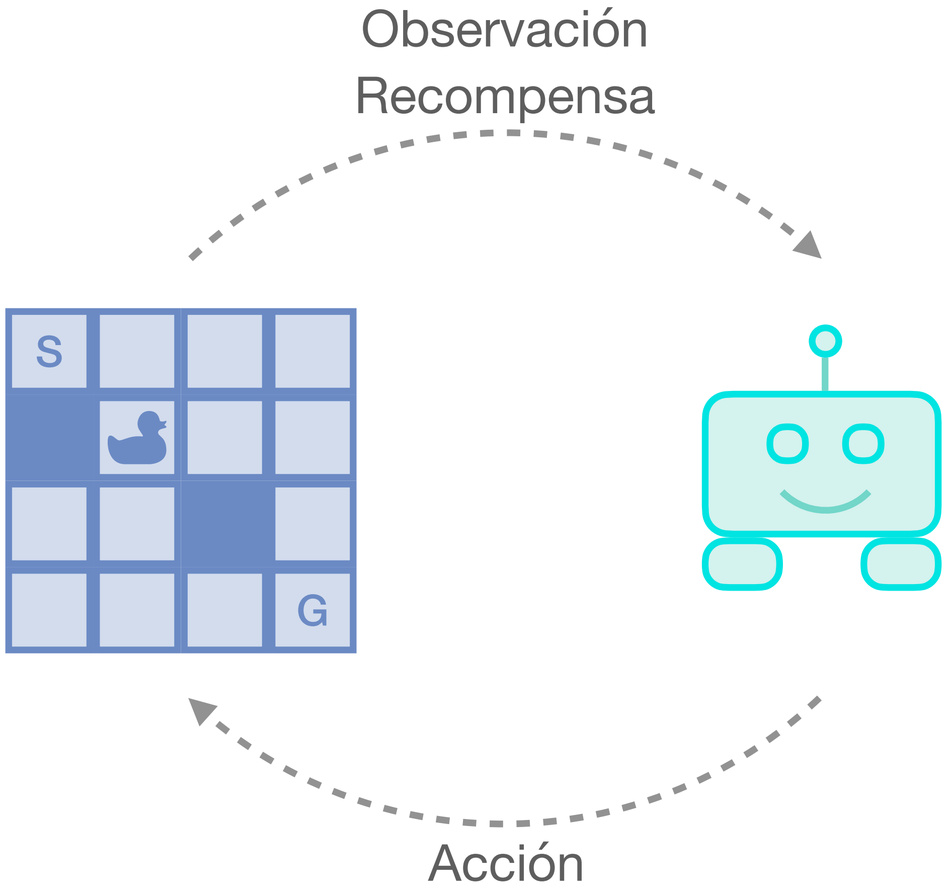
1. *v*: the difference in agent *x* values between states
   - in this case this is instantaneous velocity for the given step
   - *v = x1 - x0*
     - *x0* is the x position before the step
     - *x1* is the x position after the step
   - moving right ⇔ *v > 0*
   - moving left ⇔ *v < 0*
   - not moving ⇔ *v = 0*
2. *c*: the difference in the game clock between frames
   - the penalty prevents the agent from standing still
   - *c = c0 - c1*
     - *c0* is the clock reading before the step
     - *c1* is the clock reading after the step
   - no clock tick ⇔ *c = 0*
   - clock tick ⇔ *c < 0*
3. *d*: a death penalty that penalizes the agent for dying in a state
   - this penalty encourages the agent to avoid death
   - alive ⇔ *d = 0*
   - dead ⇔ *d = -15*

r = v + c + d

The reward is clipped into the range *(-15, 15)*.

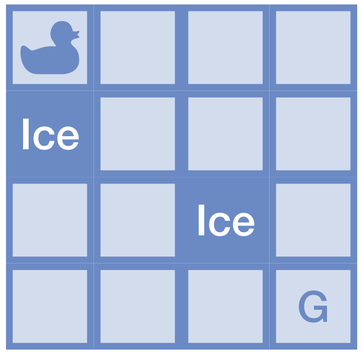
Recompensa

¿Cómo aprende el agente?

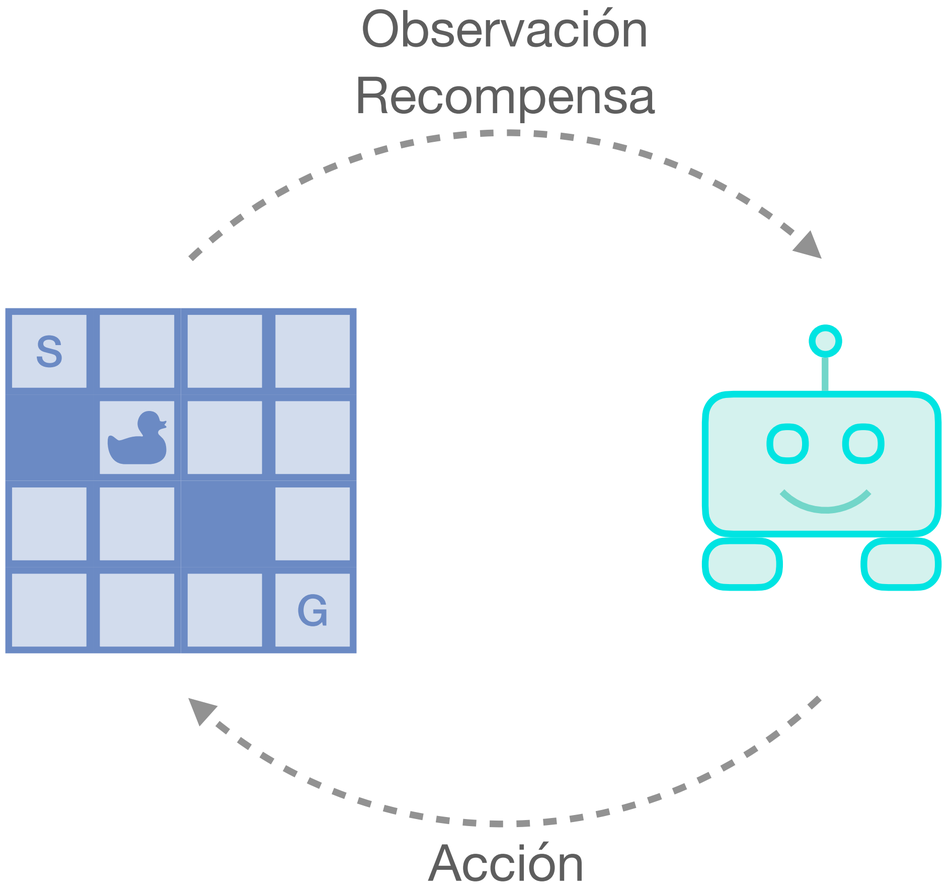


Política

Es la regla usada por el agente para seleccionar una acción. Indica qué acción tomar en cada estado.

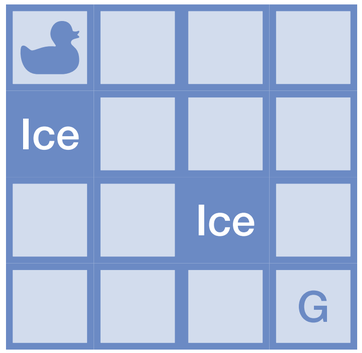


¿Cómo aprende el agente?



Política

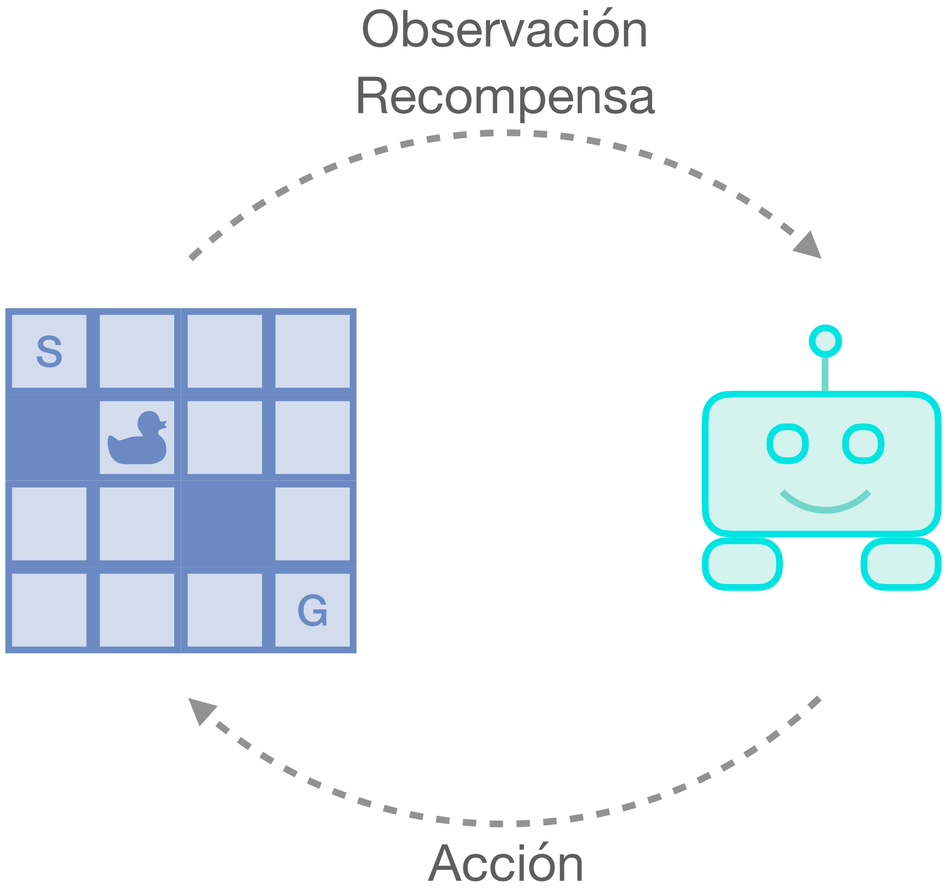
Es la regla usada por el agente para seleccionar una acción. Indica qué acción tomar en cada estado.



Ir siempre a la derecha

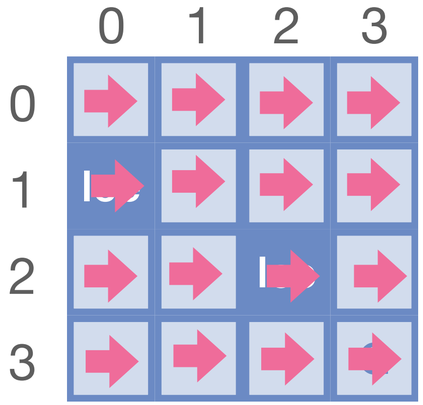
Política

¿Cómo aprende el agente?



Política

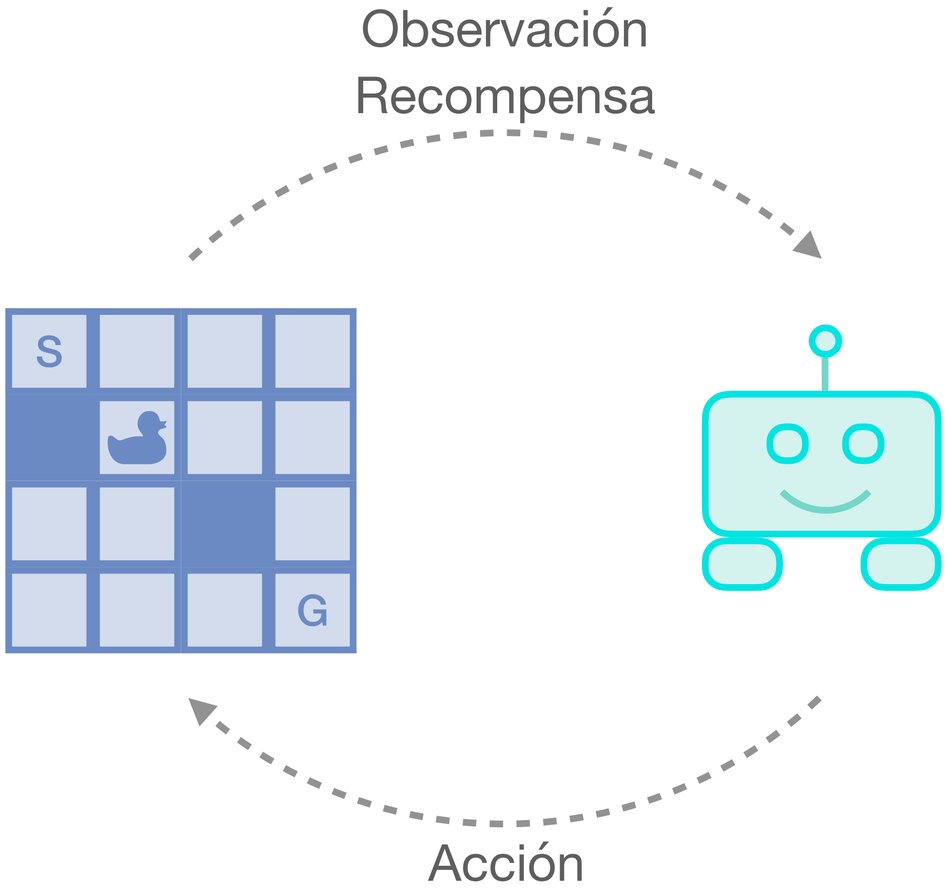
Es la regla usada por el agente para seleccionar una acción. Indica qué acción tomar en cada estado.



Ir siempre a la derecha

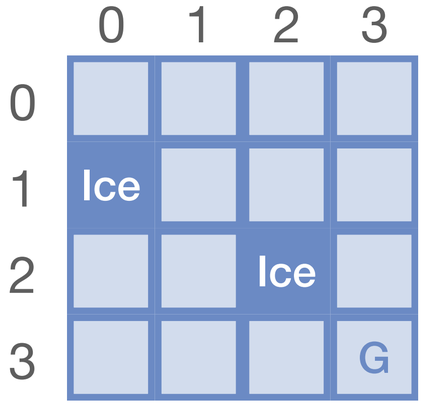
Política

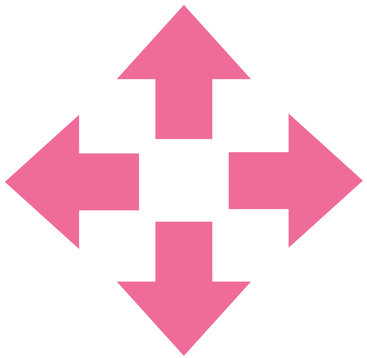
¿Cómo aprende el agente?



Política

Es la regla usada por el agente para seleccionar una acción. Indica qué acción tomar en cada estado.

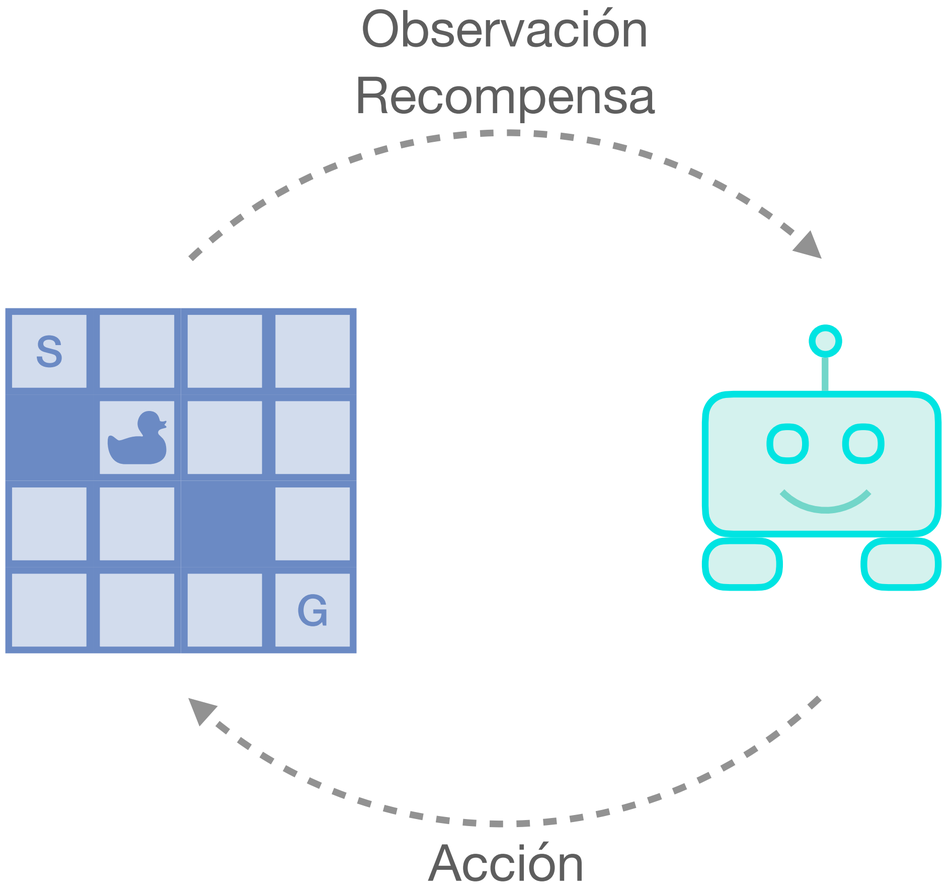




Política random

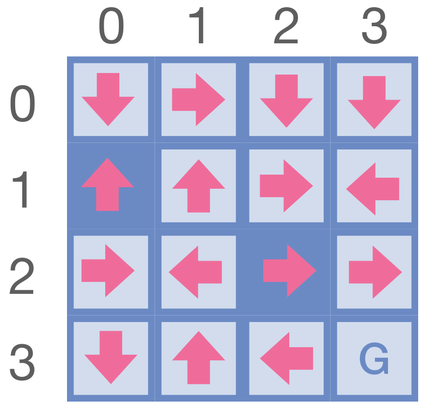
Política

¿Cómo aprende el agente?



Política

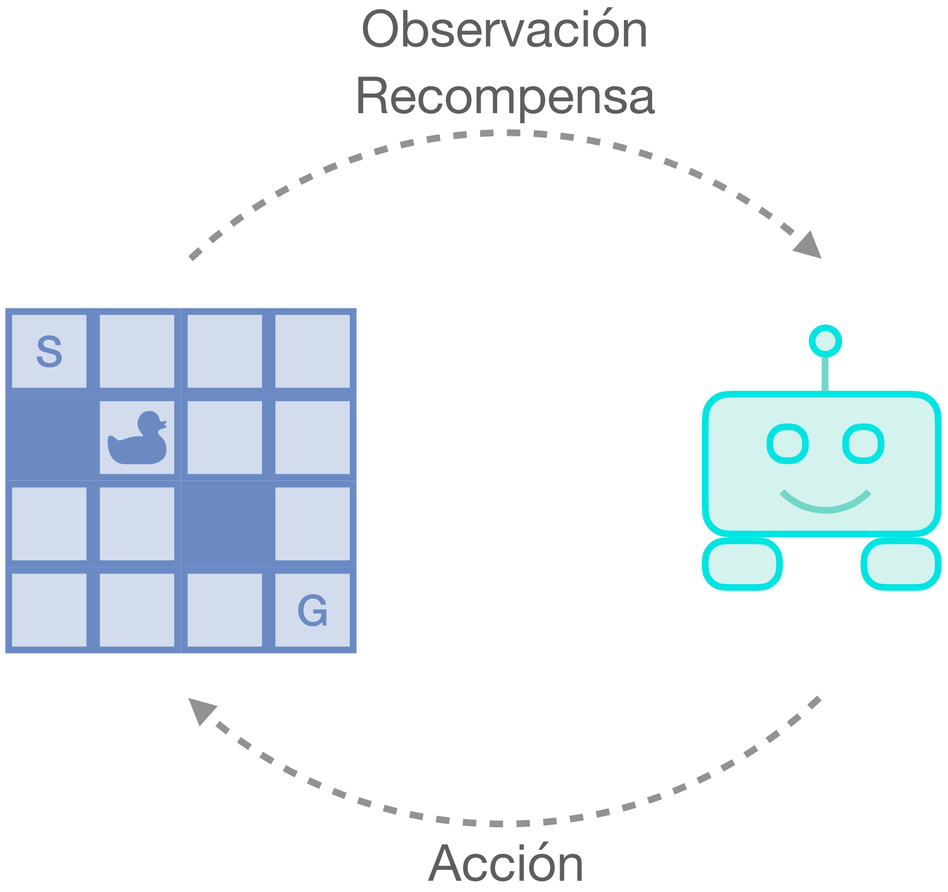
Es la regla usada por el agente para seleccionar una acción. Indica qué acción tomar en cada estado.



En general, podemos comenzar con una política random y la idea es que el agente vaya mejorando la política a través de la experiencia

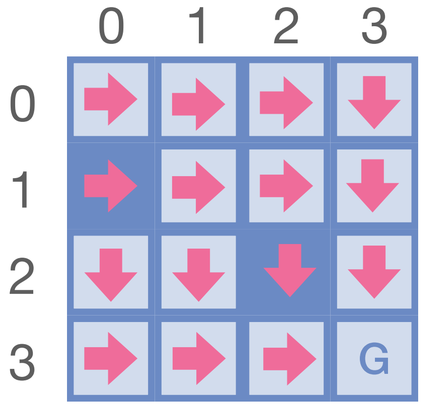
Política

¿Cómo aprende el agente?



Política

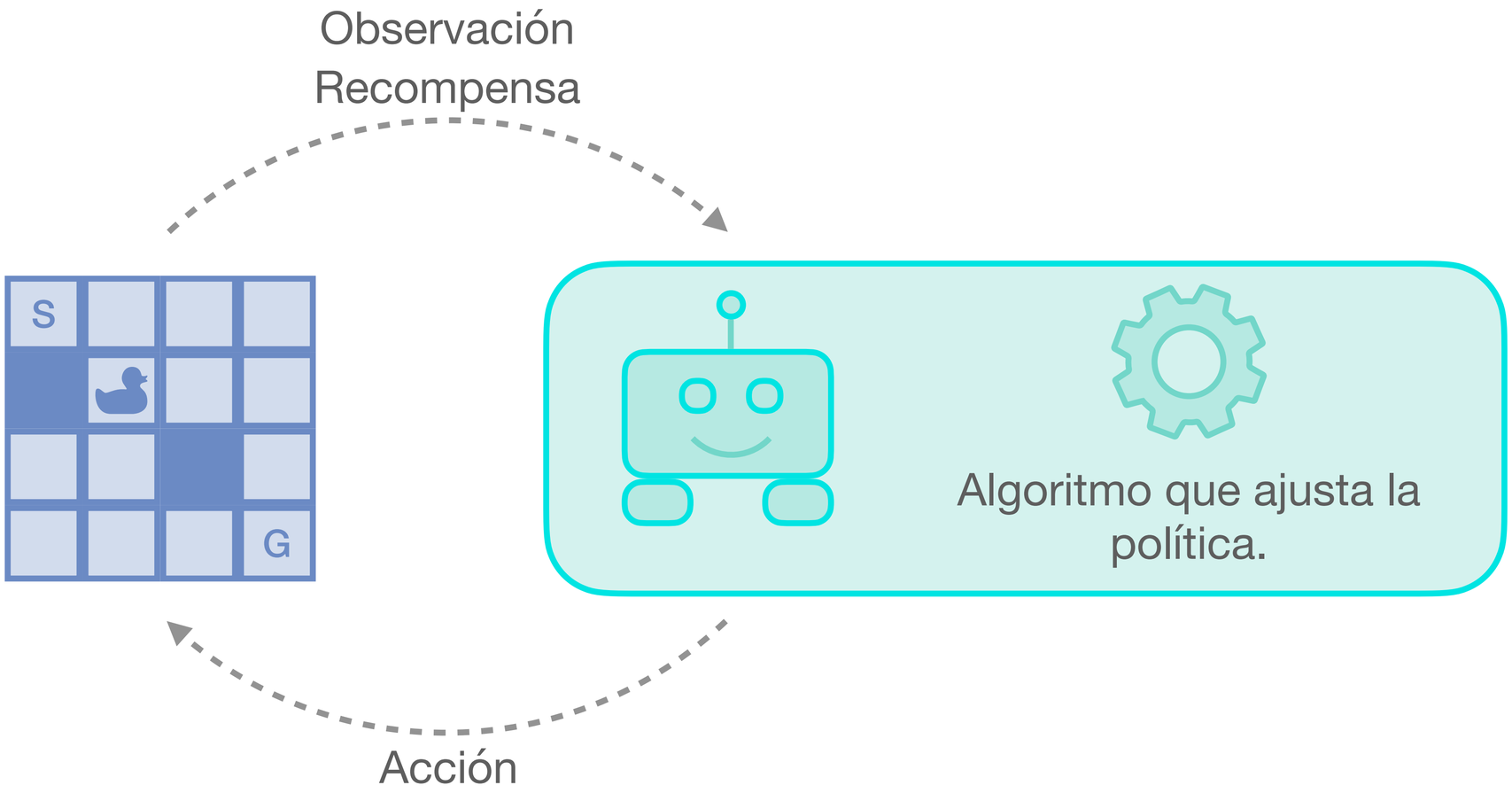
Es la regla usada por el agente para seleccionar una acción. Indica qué acción tomar en cada estado.



En general, podemos comenzar con una política random y la idea es que el agente vaya mejorando la política a través de la experiencia

Política

¿Cómo aprende el agente?



OpenAI Gym

ACTUMLOGOS

DESARROLLANDO HABILIDADES TECNOLÓGICAS

Tour por Gym

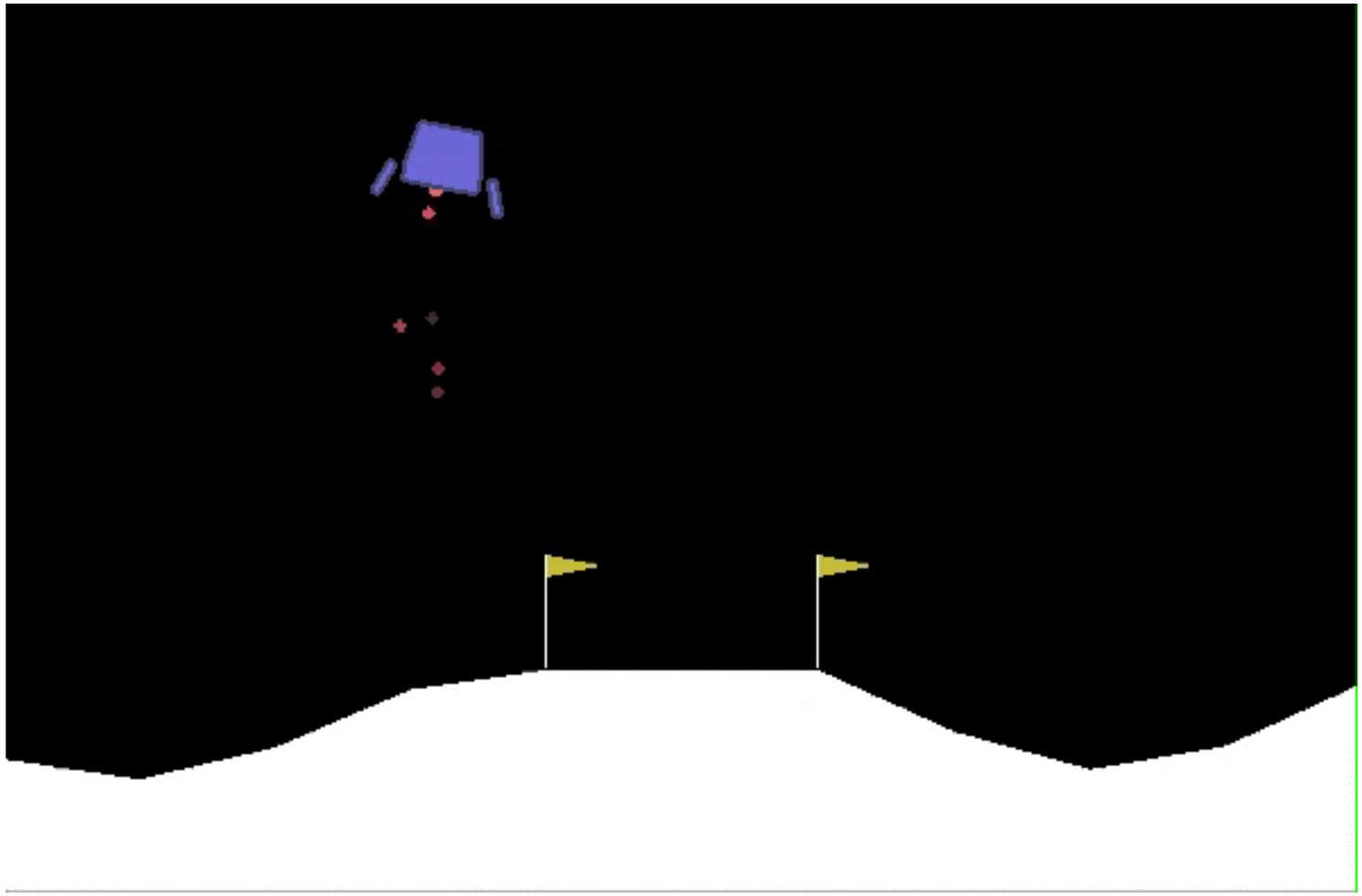
https://arxiv.org/pdf/1606.01540.pdf

https://gym.openai.com

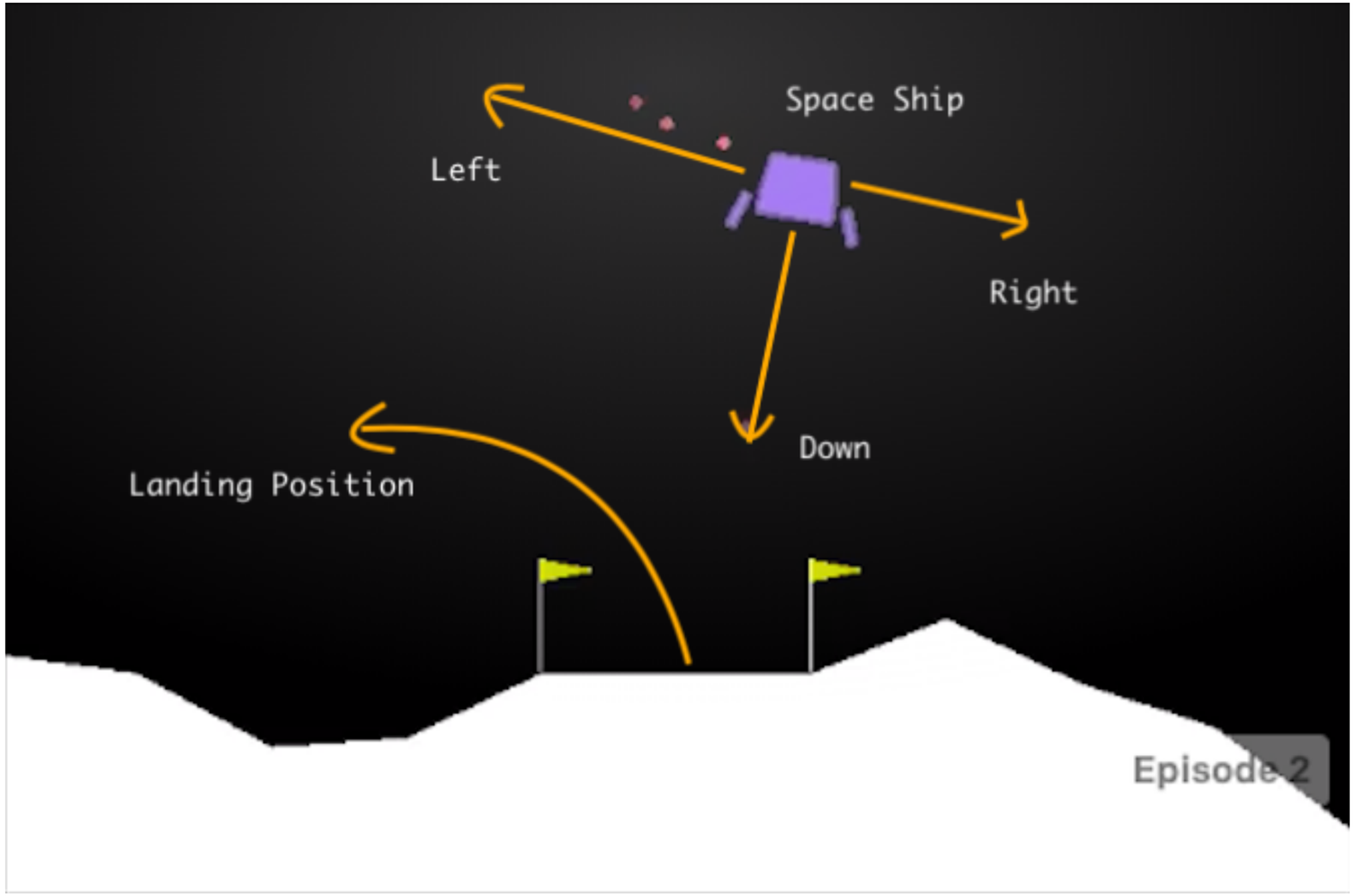
https://github.com/openai/gym

https://www.gymlibrary.ml

LunarLander



LunarLander



Objetivo

- Aterrizar entre las banderas
- Episodio termina si choca o aterriza correctamente

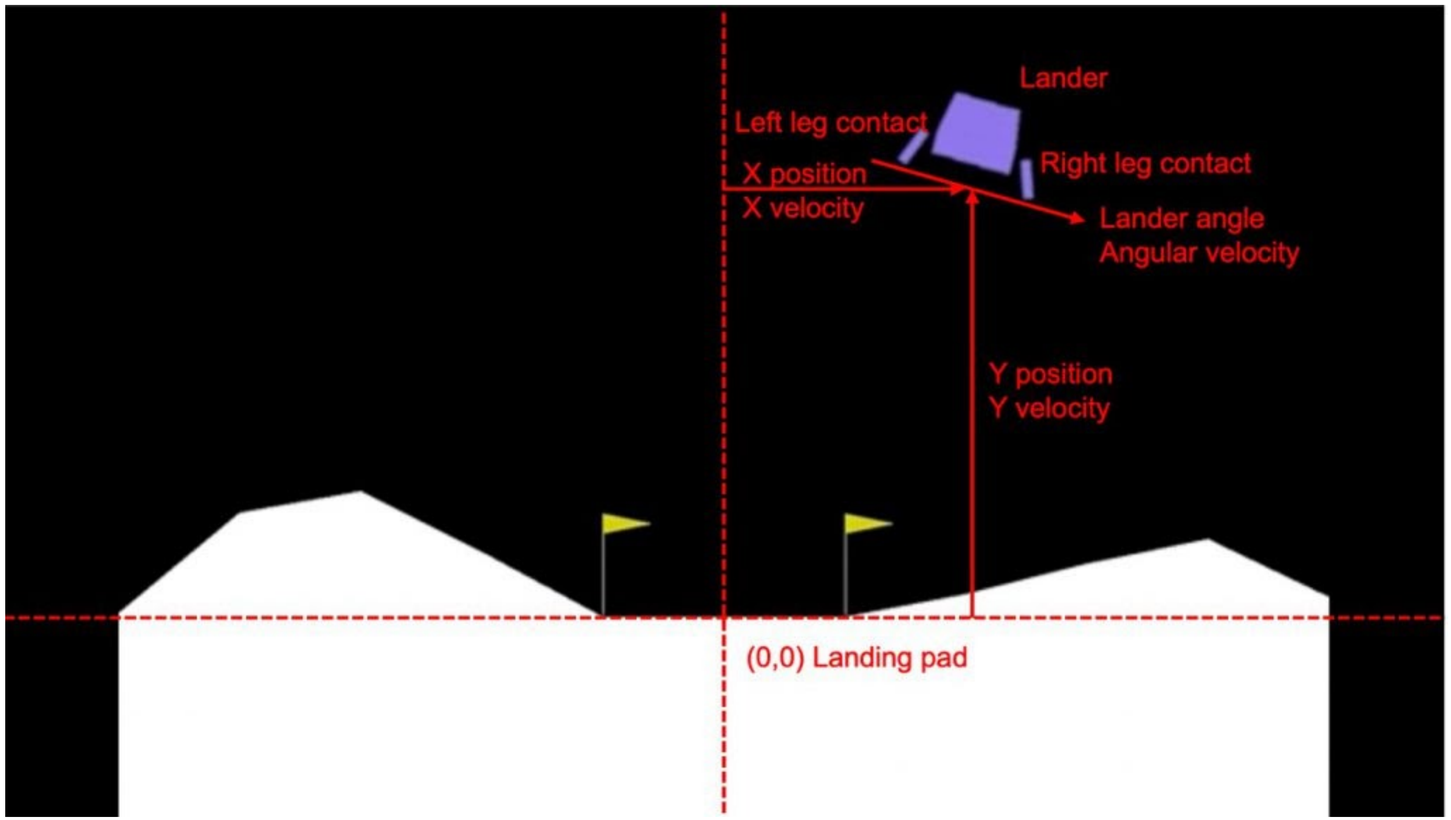
Acciones

- Activar propulsor derecho
- Activar propulsor izquierdo
- Activar propulsor principal (abajo)
- No activar propulsores

Recompensas

- Chocar -100, aterrizar +100, activar propulsor principal -3 (por paso de tiempo), etc.

LunarLander



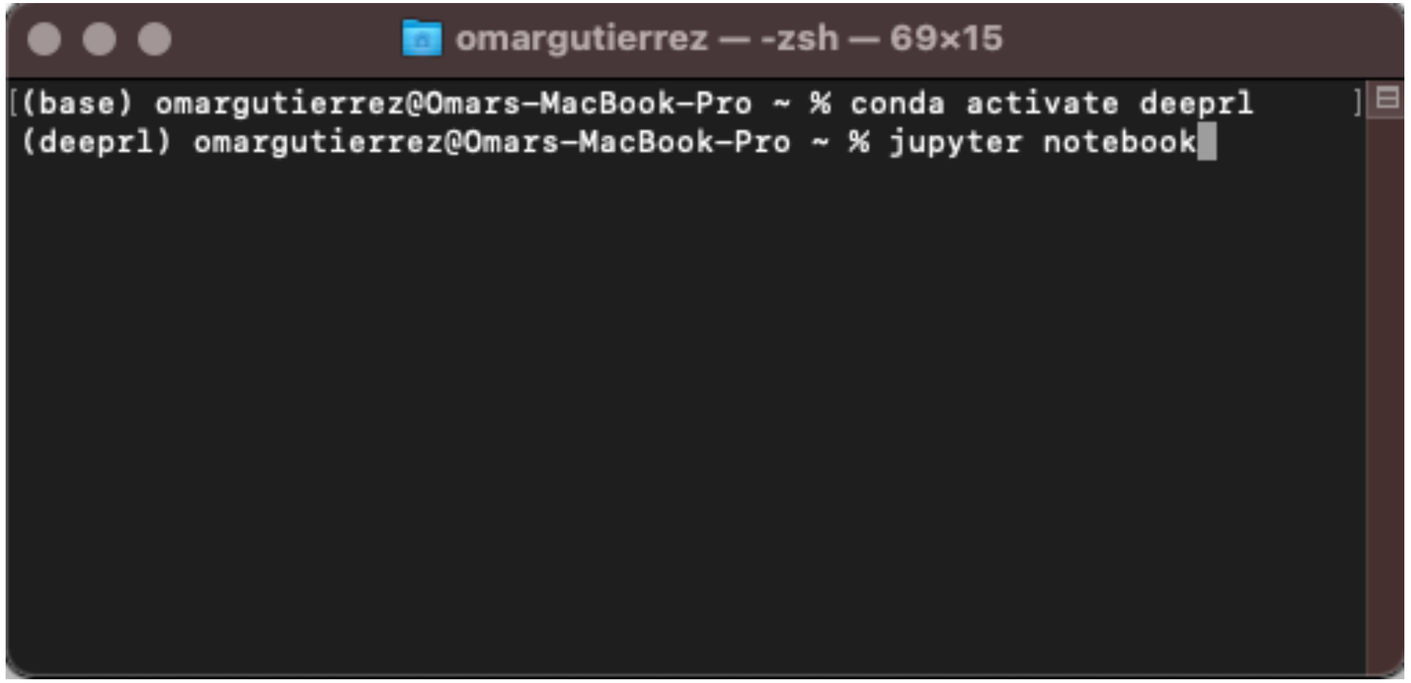
Observación

- Posición en x
- Posición en y
- Velocidad en x
- Velocidad en y
- Ángulo de aterrizaje
- Velocidad angular
- Indicador de contacto en pata izquierda
- Indicador de contacto en pata derecha

```
[-0.00619364  1.4174144  -0.6273631   0.28862926  0.00718367  0.14210723   0.          0.        ]
```

Entendiendo Gym

- Descargar y abrir el notebook taller_rl.ipynb en el ambiente deeprl

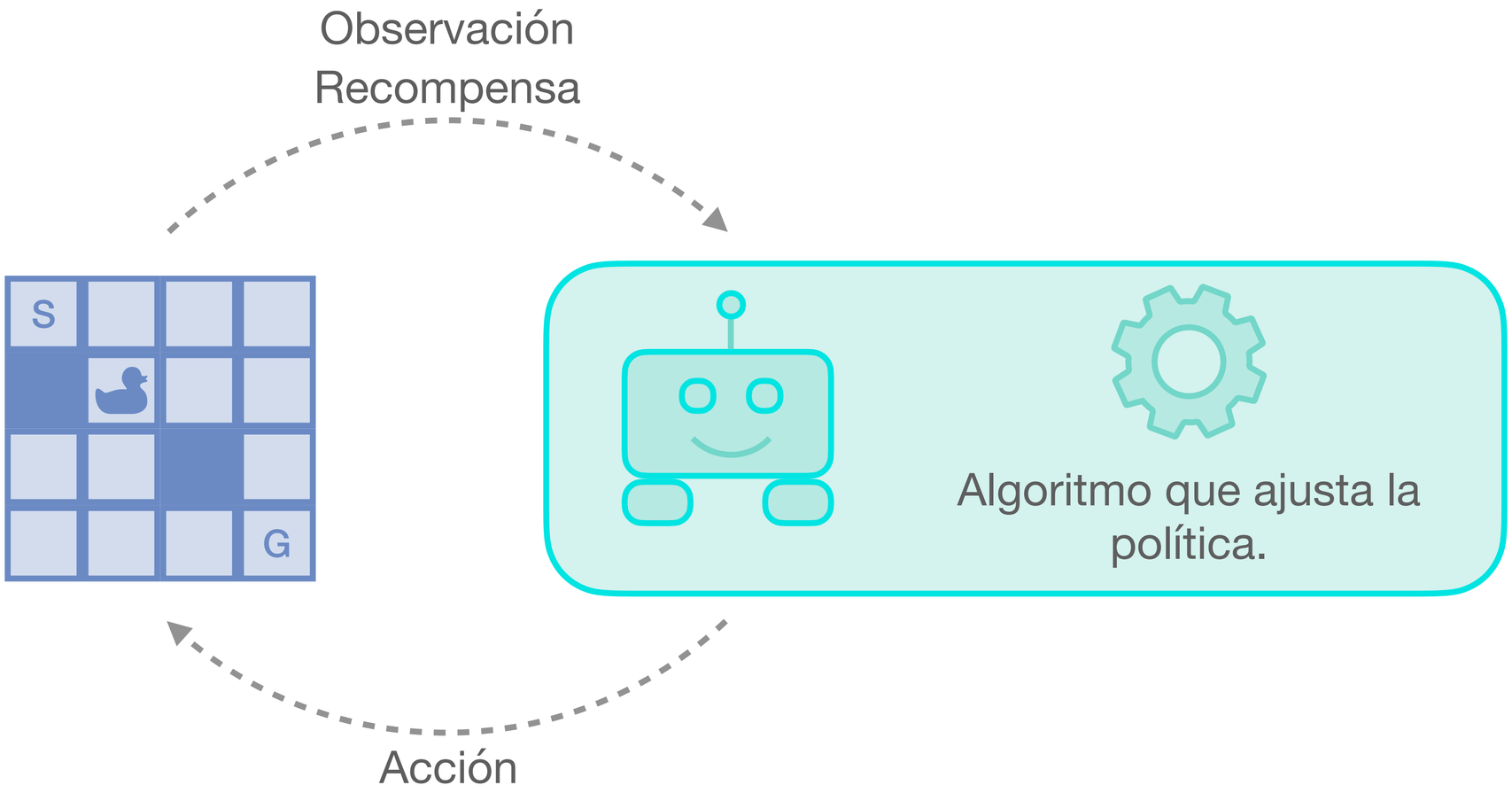


¿Cómo aprenden los agentes?

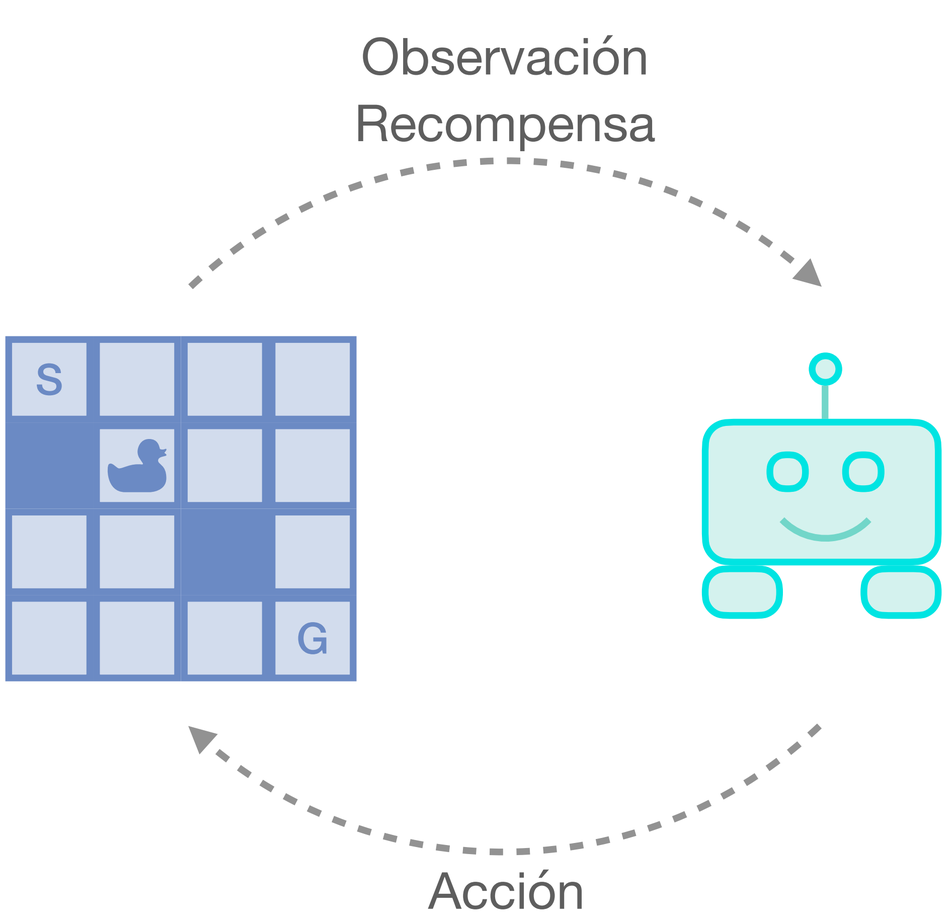
ACTUMLOGOS

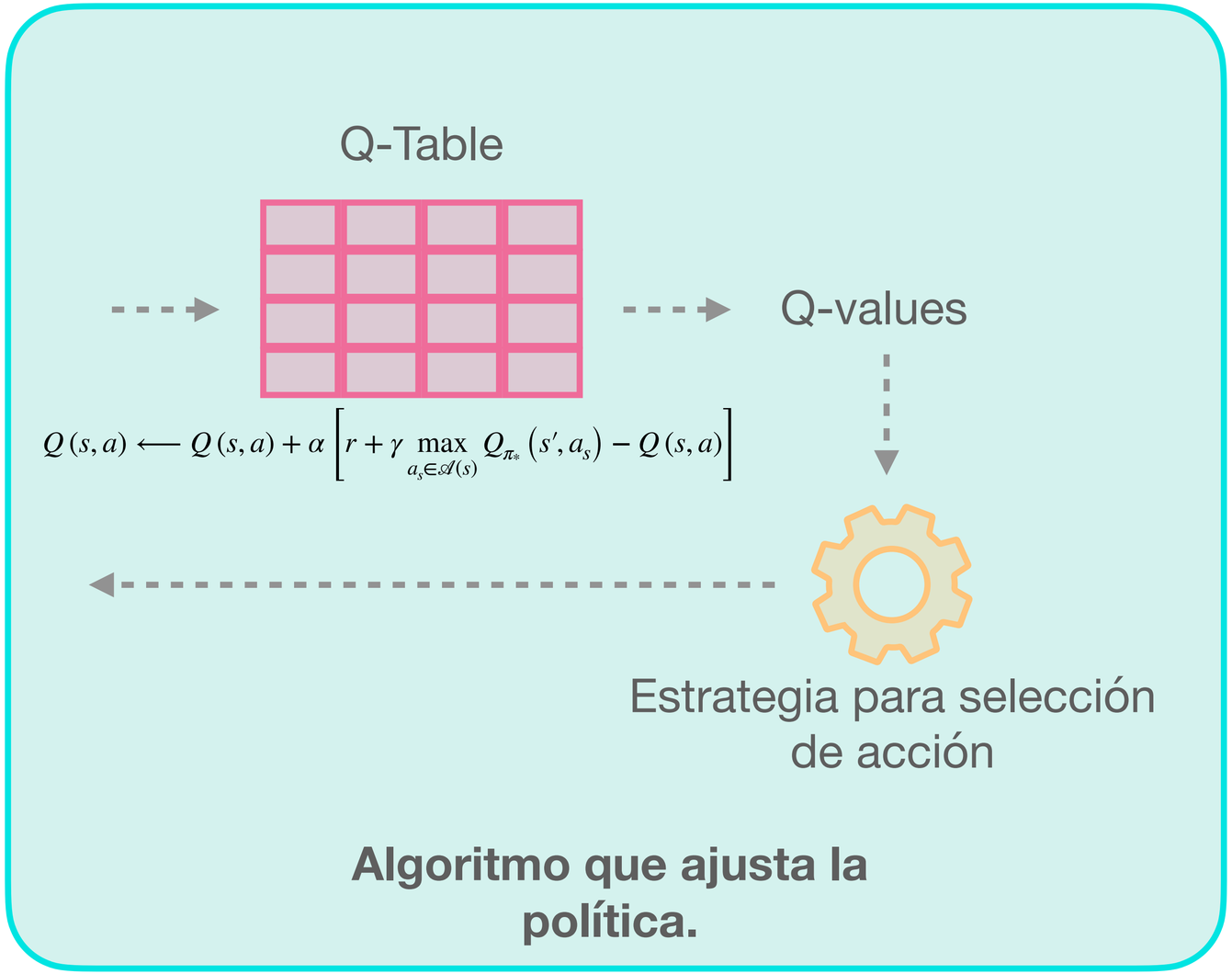
DESARROLLANDO HABILIDADES TECNOLÓGICAS

¡Ajustando la política!



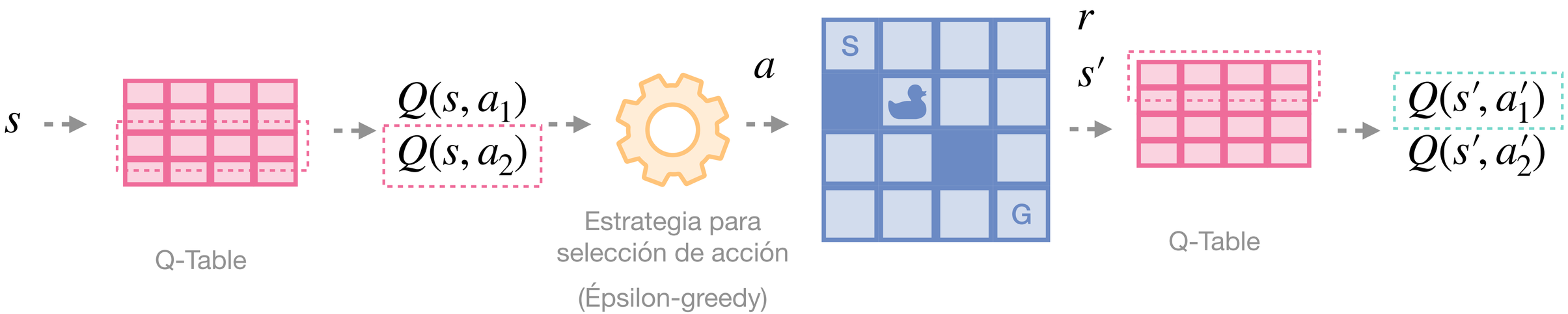
Q-Learning





¿Cómo aprende el agente?

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



Deep Learning

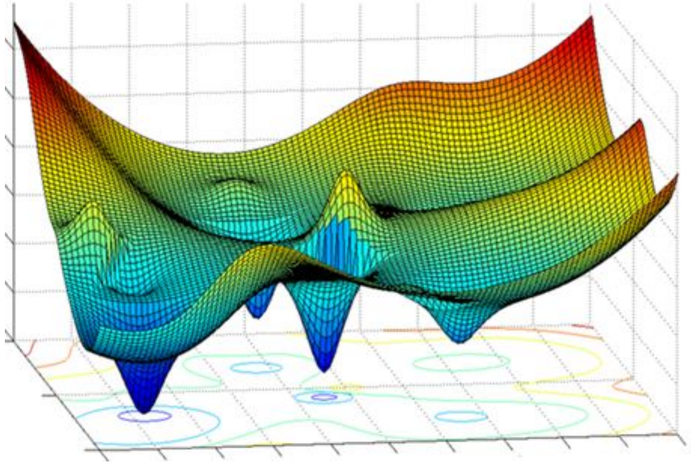
ACTUMLOGOS

DESARROLLANDO HABILIDADES TECNOLÓGICAS

Redes neuronales

$$w_t = w_{t-1} - \alpha \nabla L(w_{t-1})$$

Regla de actualización
de descenso por gradiente



Dataset

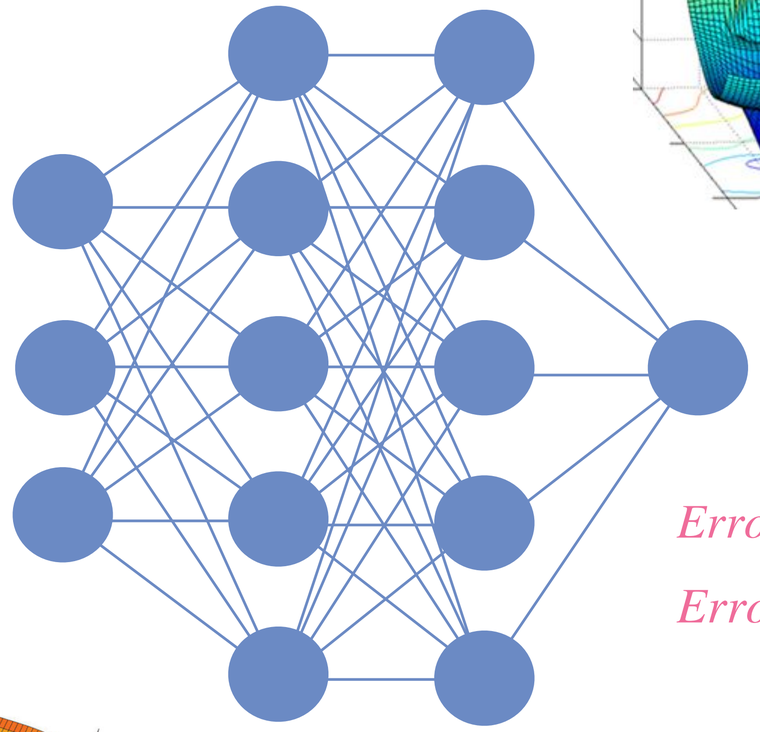
Asignación de crédito

Features

Sexo	Edad	Salario
M	25	16,000
H	48	25,000
H	75	38,000
M	21	27,000
...		
H	25	16,000

Target

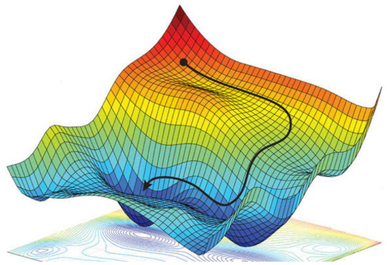
$
10,000
27,000
20,000
35,000
...
9,300



$$\hat{y} = 11{,}528$$

$$Error\left(\hat{y}, y\right) = \left(y - \hat{y}\right)^2$$

$$Error\left(\hat{y}, y\right) = \left(10000 - 11528\right)^2$$



Deep Reinforcement Learning

ACTUMLOGOS

DESARROLLANDO HABILIDADES TECNOLÓGICAS

Problema

- Los métodos tabulares como Q-Learning, iterativamente aproximan el valor de $Q(s, a)$ y lo almacenan en una tabla (Q-Table).
- Problemas que presentan los métodos tabulares:
  - Requiere que el espacio de estados y acciones sean discretos y no muy grandes.
  - Si el espacio de estados o de acciones es continuo, la Q-Table crece infinitamente.

Q-Table

State	a_1	a_2	a_3	a_4
s_1				
s_2				
s_3				
...				
s_n				

Problema

- Estamos usando una tabla para mapear duplas (s, a) a un valor real $Q(s, a)$.
- En vez de eso, podemos usar una función que aproxime dicha table.
- De esta forma, a cada paso de tiempo, el agente toma la dupla (s, a) y predice $Q(s, a)$.
- Este nuevo planteamiento es un problema de regresión.

Q-Table

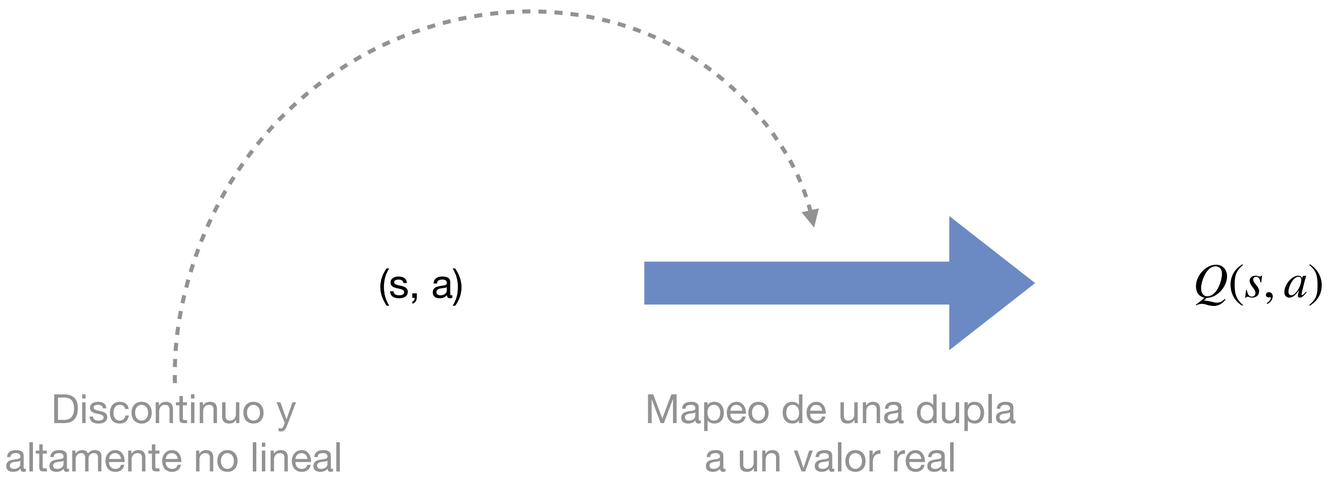
State	a_1	a_2	a_3	a_4
s_1				
s_2				
s_3				
...				
s_n				

(s, a) → $Q(s, a)$

Mapeo de una dupla a un valor real

Problema

$$(s, a) \dashrightarrow f \dashrightarrow Q(s, a)$$



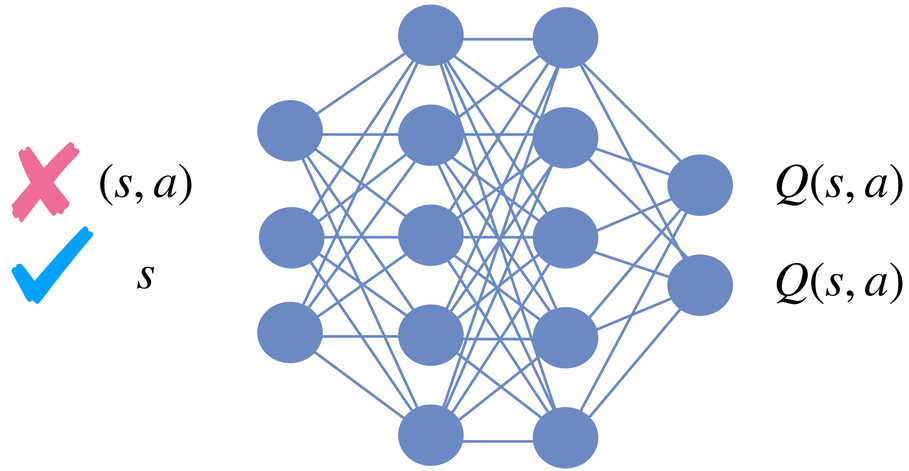
Solución

Frame

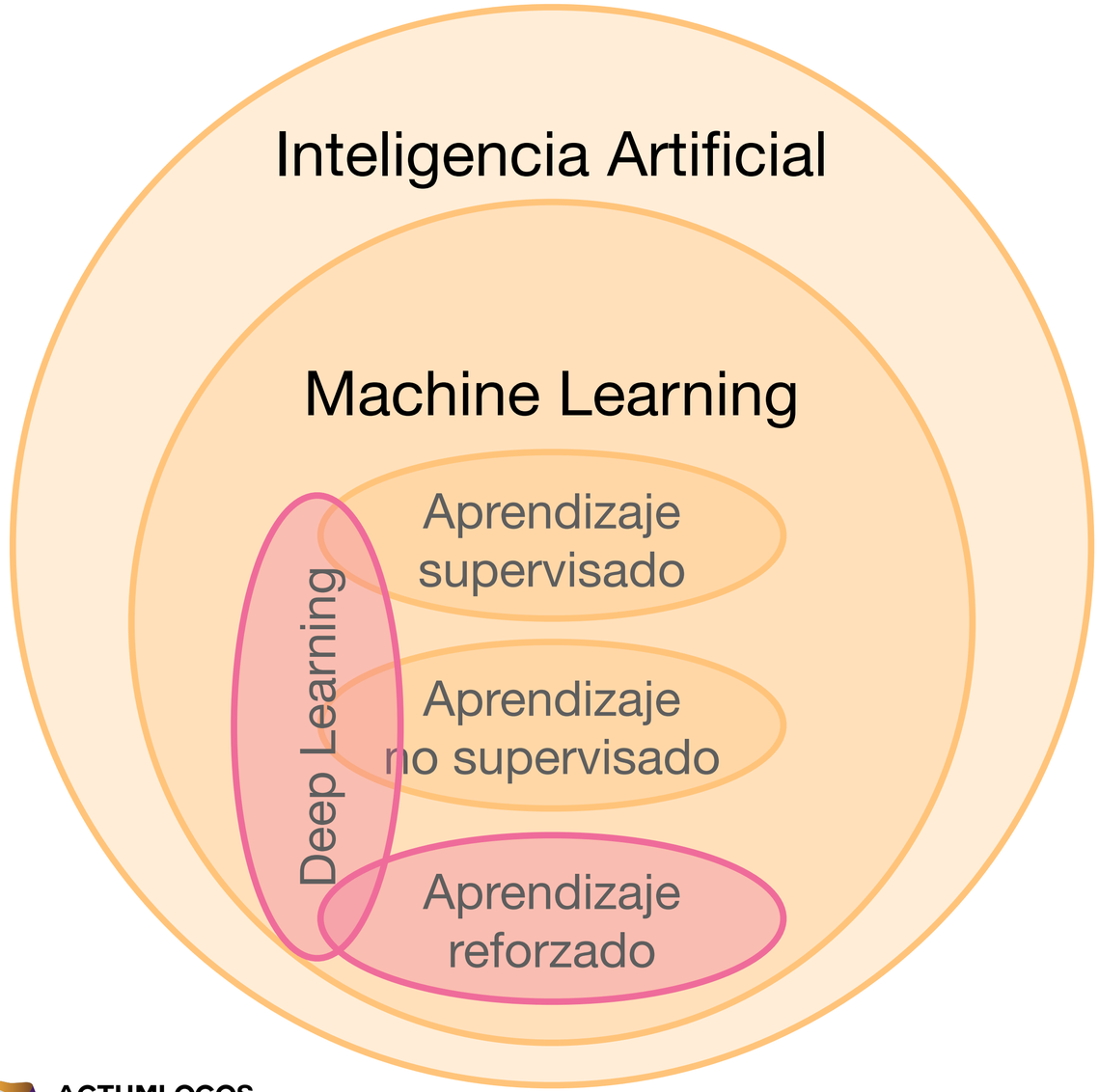
[0.1, 0.04, 0.0, 0.23, …, 0.07]

Vector

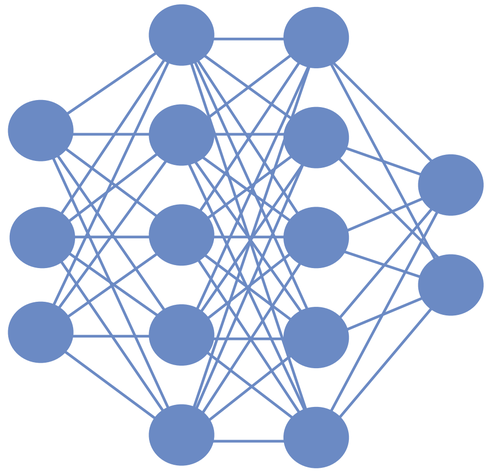
Estado s



DRL



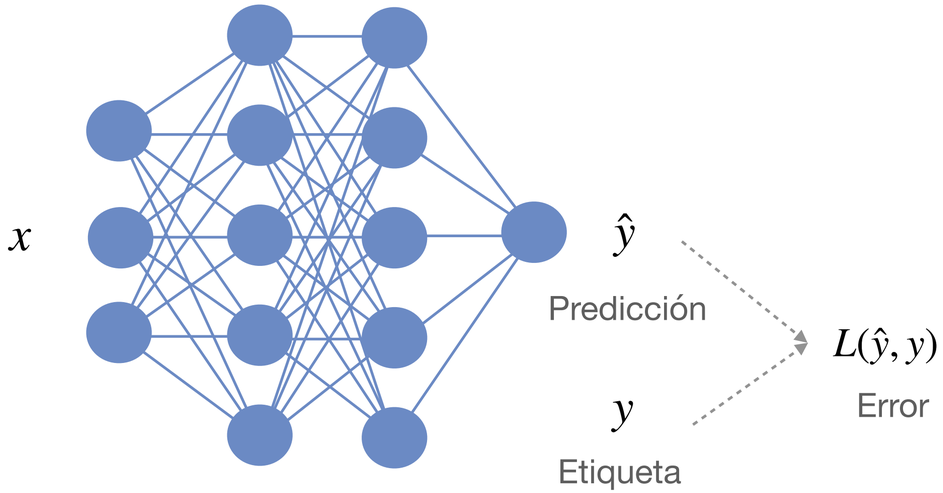
RL +

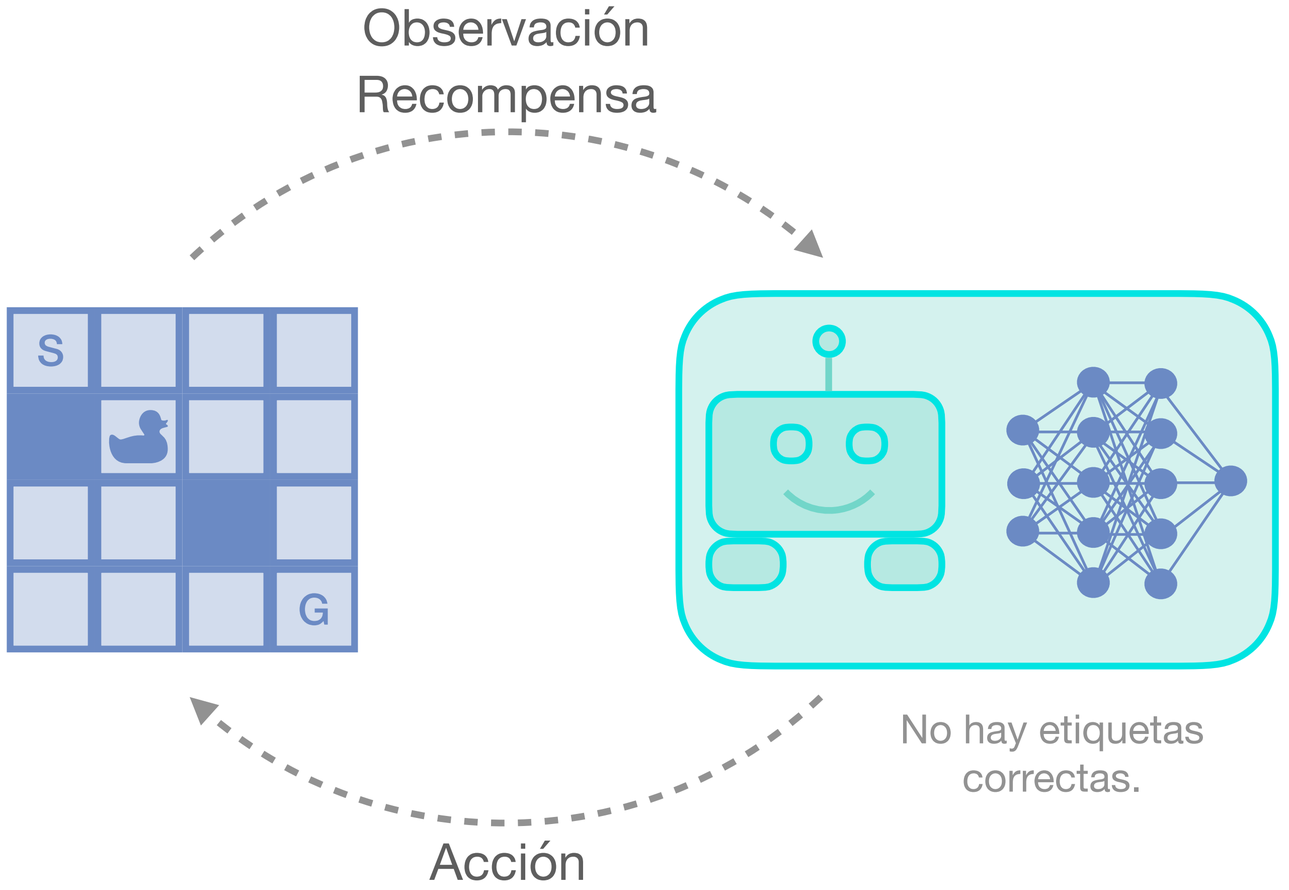


Otro problema

- Usar DL en RL es más difícil que usarlo para ML supervisado

Supervisado





Reforzado

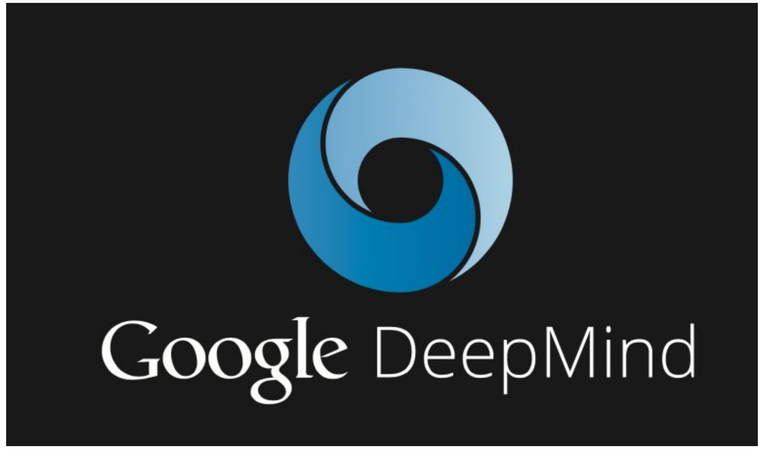
Deep Q-Learing

ACTUMLOGOS

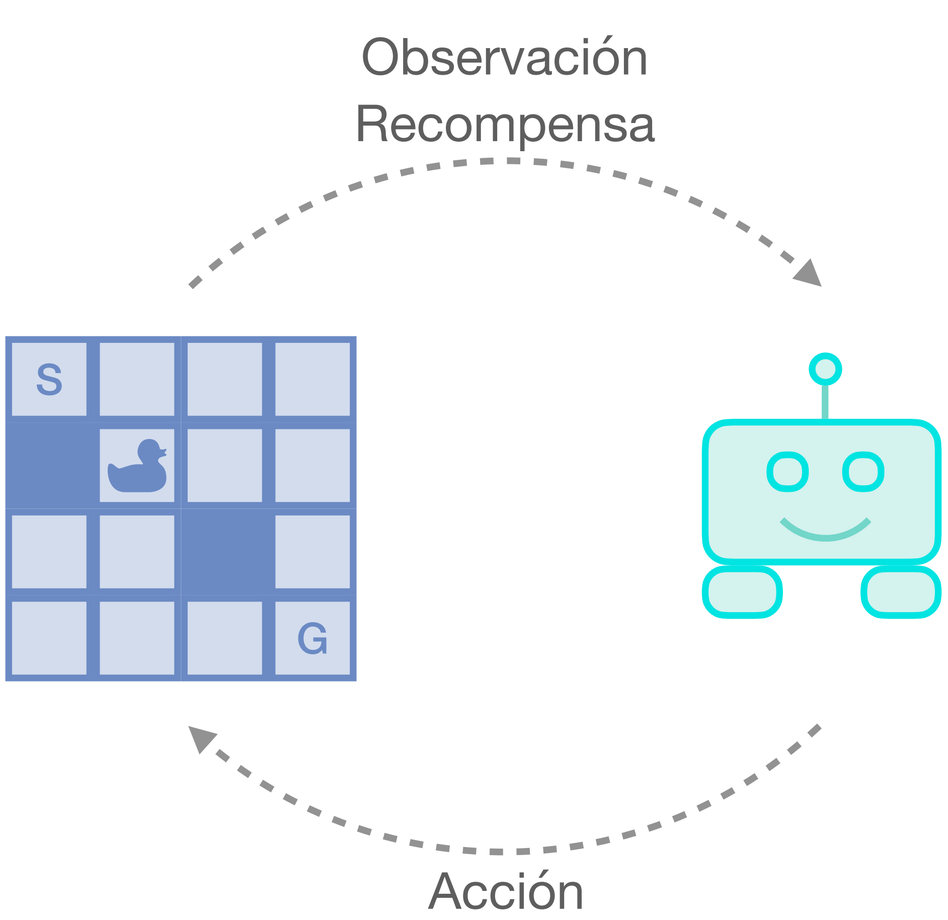
DESARROLLANDO HABILIDADES TECNOLÓGICAS

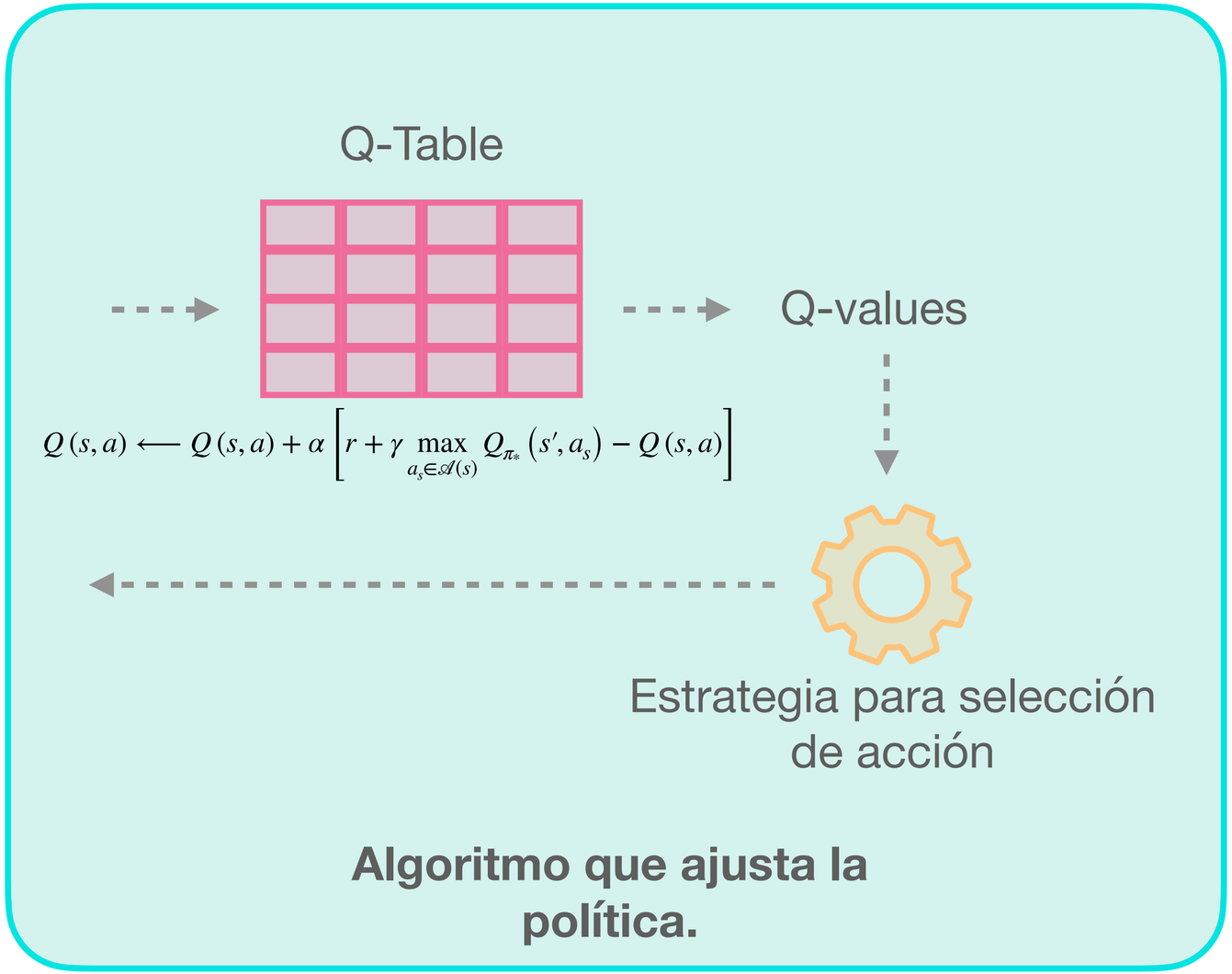
Introducción

- El algoritmo de Deep Q-Network (DQN) desarrollado por DeepMind comenzó la revolución de RL in 2013.
- Logran entrenar un modelo usando el viejo algoritmo de Q-Learning pero con redes neuronales, obteniendo resultados record en 6 juegos.
- El éxito del modelo fue en gran parte debido los trucos realizados para lidiar con problemas derivados de entrar un algoritmo de RL con NN.

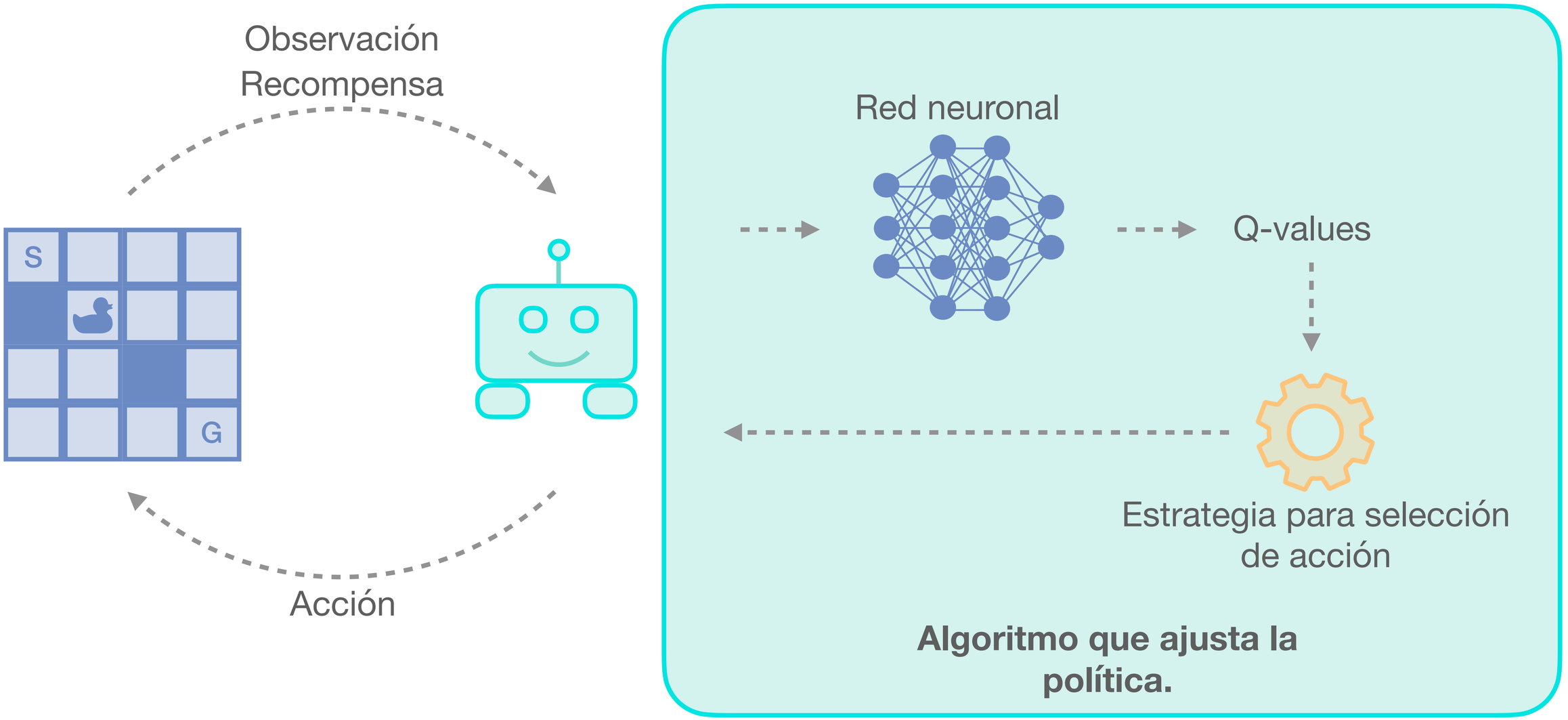


Q-Learning





Deep Q-Learning (idea general)



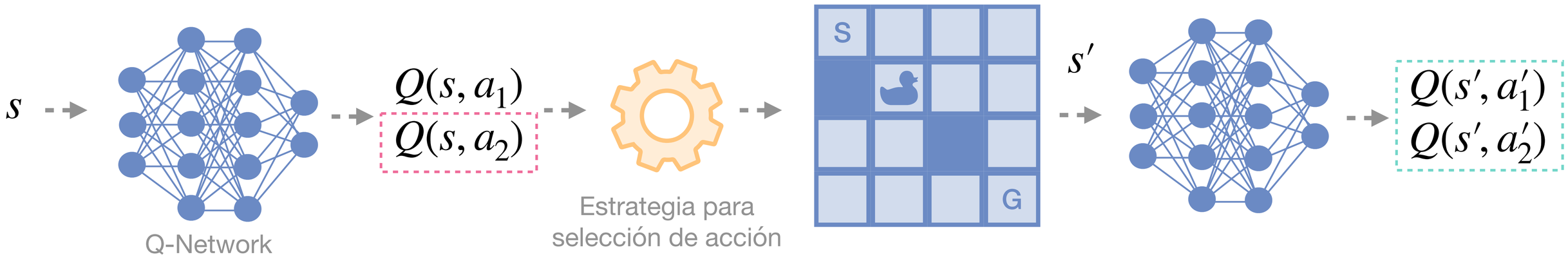
Target y función de error

Q-Learning update rule

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

Target

Prediction



Target y función de error

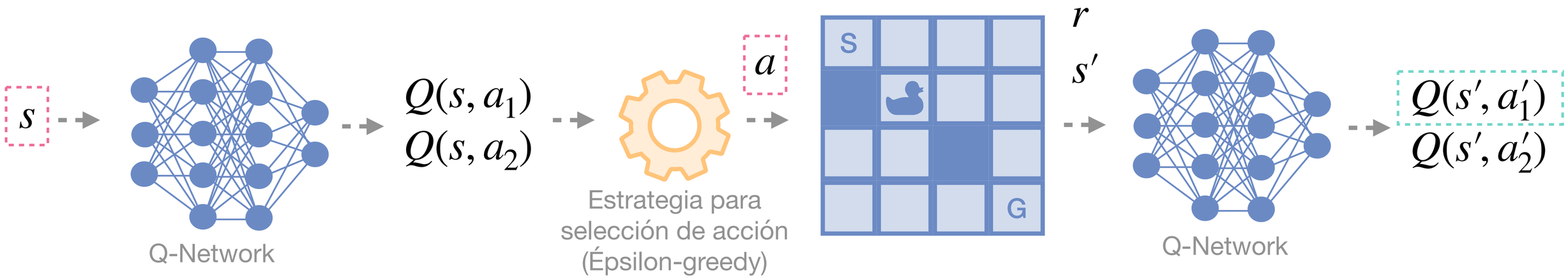
DQN update rule

$$w \longleftarrow w + \alpha \nabla \left[r + \gamma \max_a Q(s', a) - Q(s, a) \right]^2$$

Target — Prediction

$$w_t = w_{t-1} - \alpha \nabla L(w_{t-1})$$

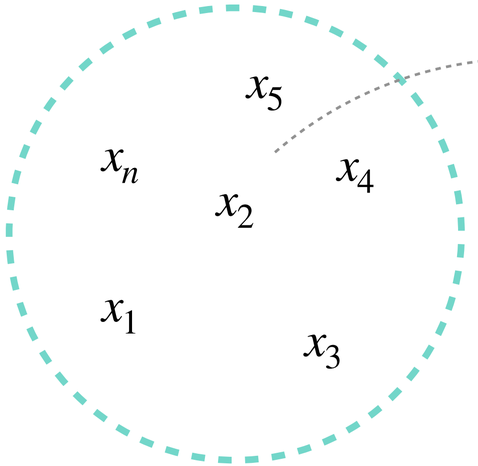
Regla de actualización
de descenso por gradiente



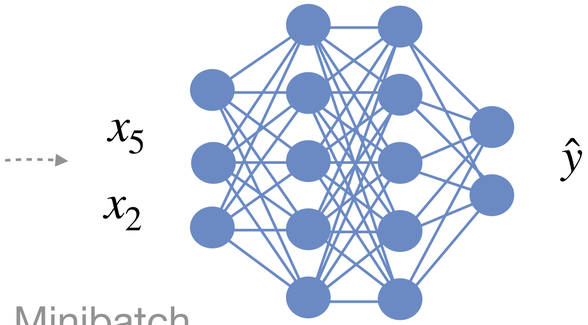
Experience Replay

Supervisado

Cada instancia
es iid

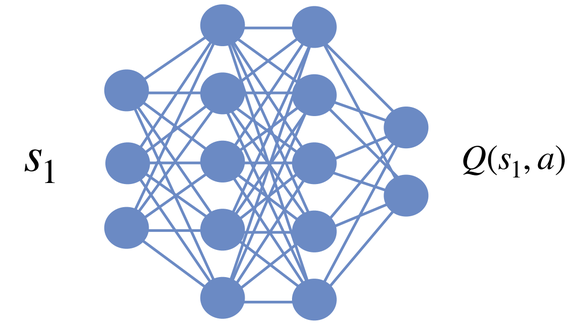


Dataset



Minibatch

Reforzado



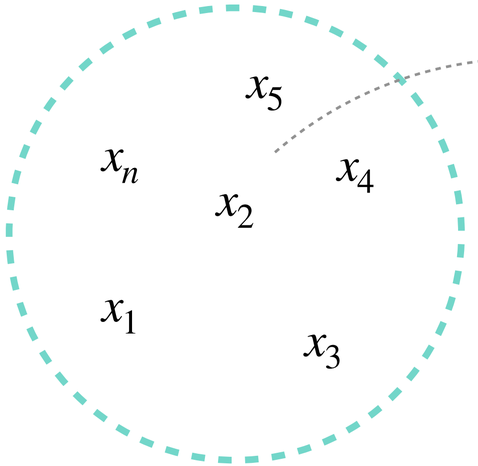
$s_1, a_1, r_2, s_2, \ldots, s_n, a_n, r_{n+1}, s_{n+1}, \ldots$

Hay
dependencia

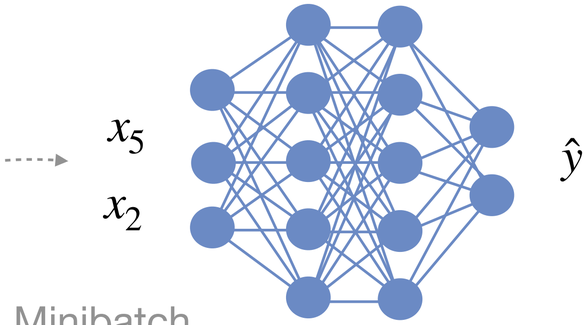
Experience Replay

Supervisado

Cada instancia es idd

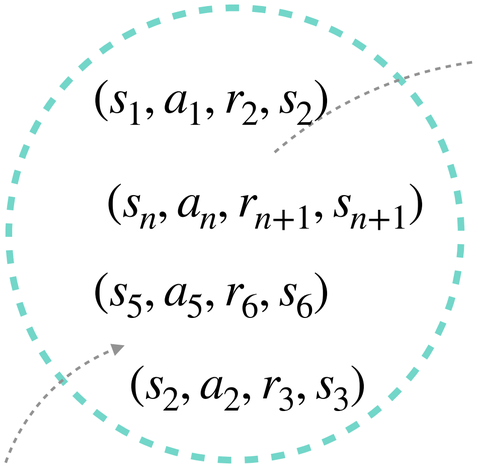


Dataset

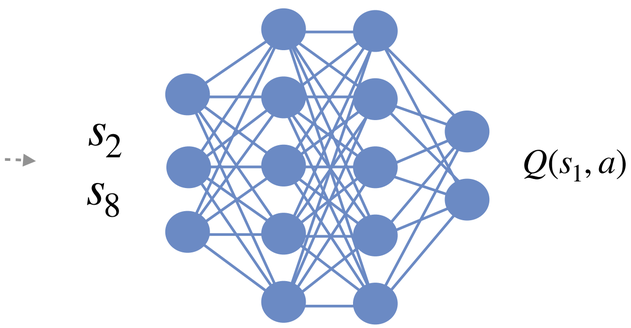


Minibatch

Reforzado



Buffer



Minibatch

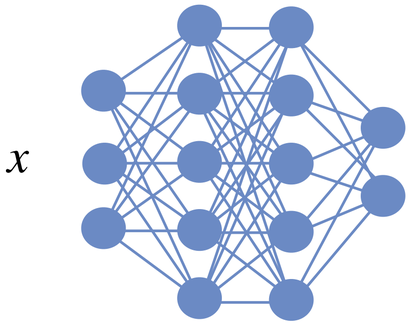
$s_1, a_1, r_2, s_2, \ldots, s_n, a_n, r_{n+1}, s_{n+1}, \ldots$

$s_1, a_1, r_2, s_2, \ldots, s_n, a_n, r_{n+1}, s_{n+1}, \ldots$

$s_1, a_1, r_2, s_2, \ldots, s_n, a_n, r_{n+1}, s_{n+1}, \ldots$

Target en movimiento

Supervisado



x

$\hat{y}$

Predicción

$L(\hat{y}, y)$

Error

y

Etiqueta

Target fijo

Reforzado

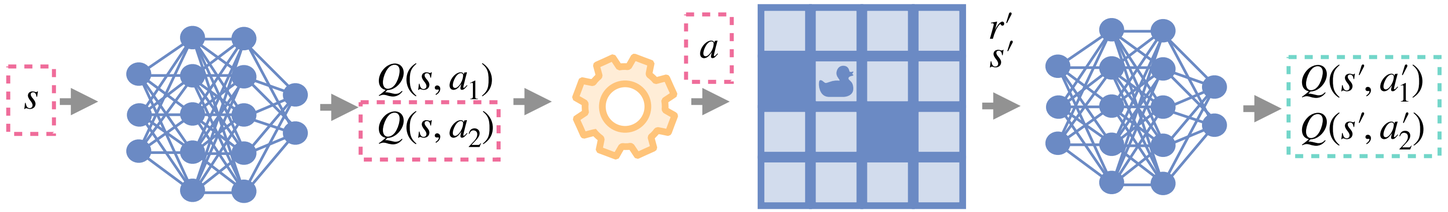
Target en movimiento

DQN update rule

$$w \longleftarrow w + \alpha \nabla \left[r + \gamma \max_{a} Q(s', a) - Q(s, a) \right]^2$$

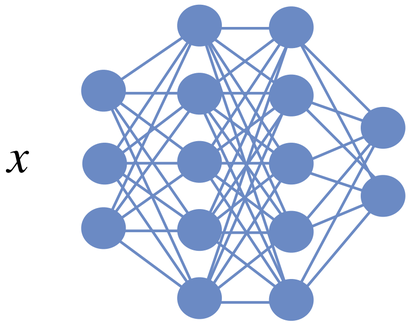
Target

Prediction



Target en movimiento

Supervisado



x

$\hat{y}$

Predicción

$L(\hat{y}, y)$

Error

y

Etiqueta

Target fijo

Target en movimiento

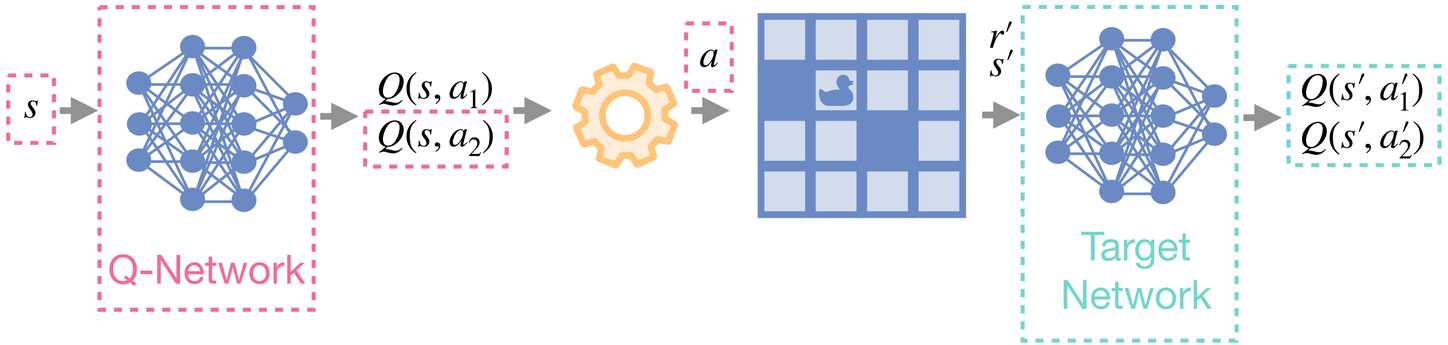
Reforzado

DQN update rule

$$w \longleftarrow w + \alpha \nabla \left[r + \gamma \max_{a} Q(s', a) - Q(s, a) \right]^2$$

Target

Prediction

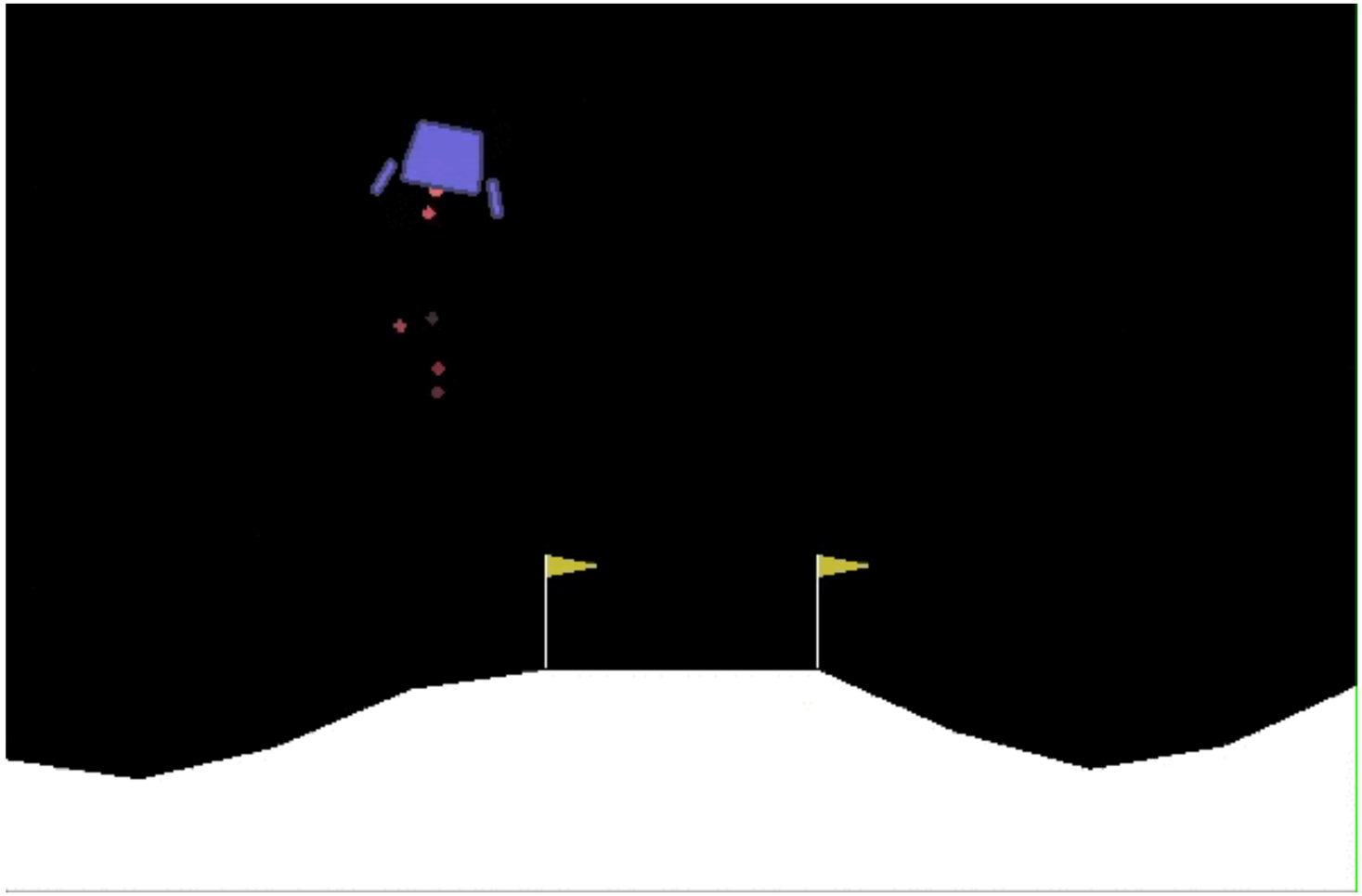


LunarLander

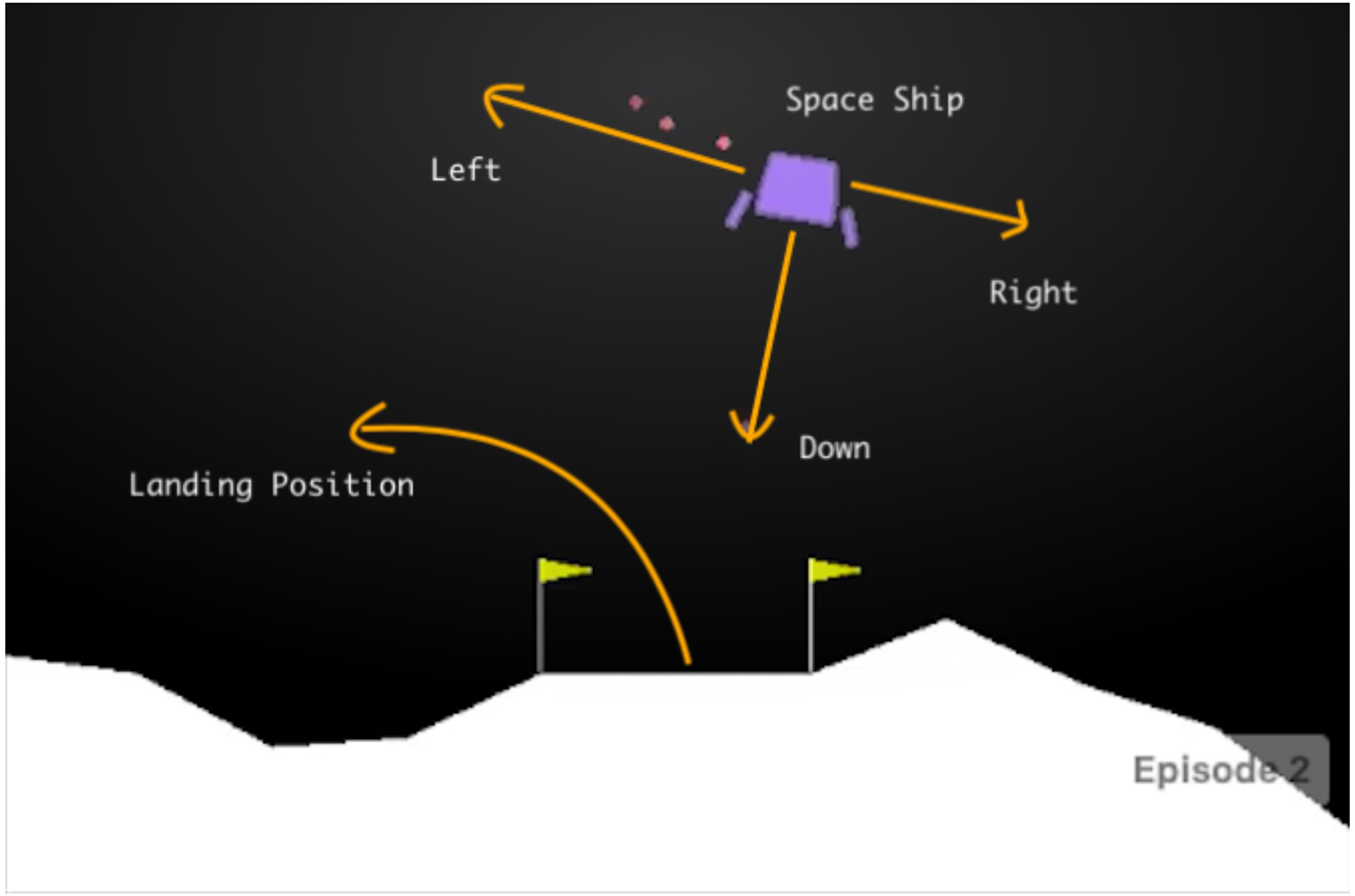
ACTUMLOGOS

DESARROLLANDO HABILIDADES TECNOLÓGICAS

LunarLander



LunarLander



Objetivo

- Aterrizar entre las banderas
- Episodio termina si choca o aterriza correctamente

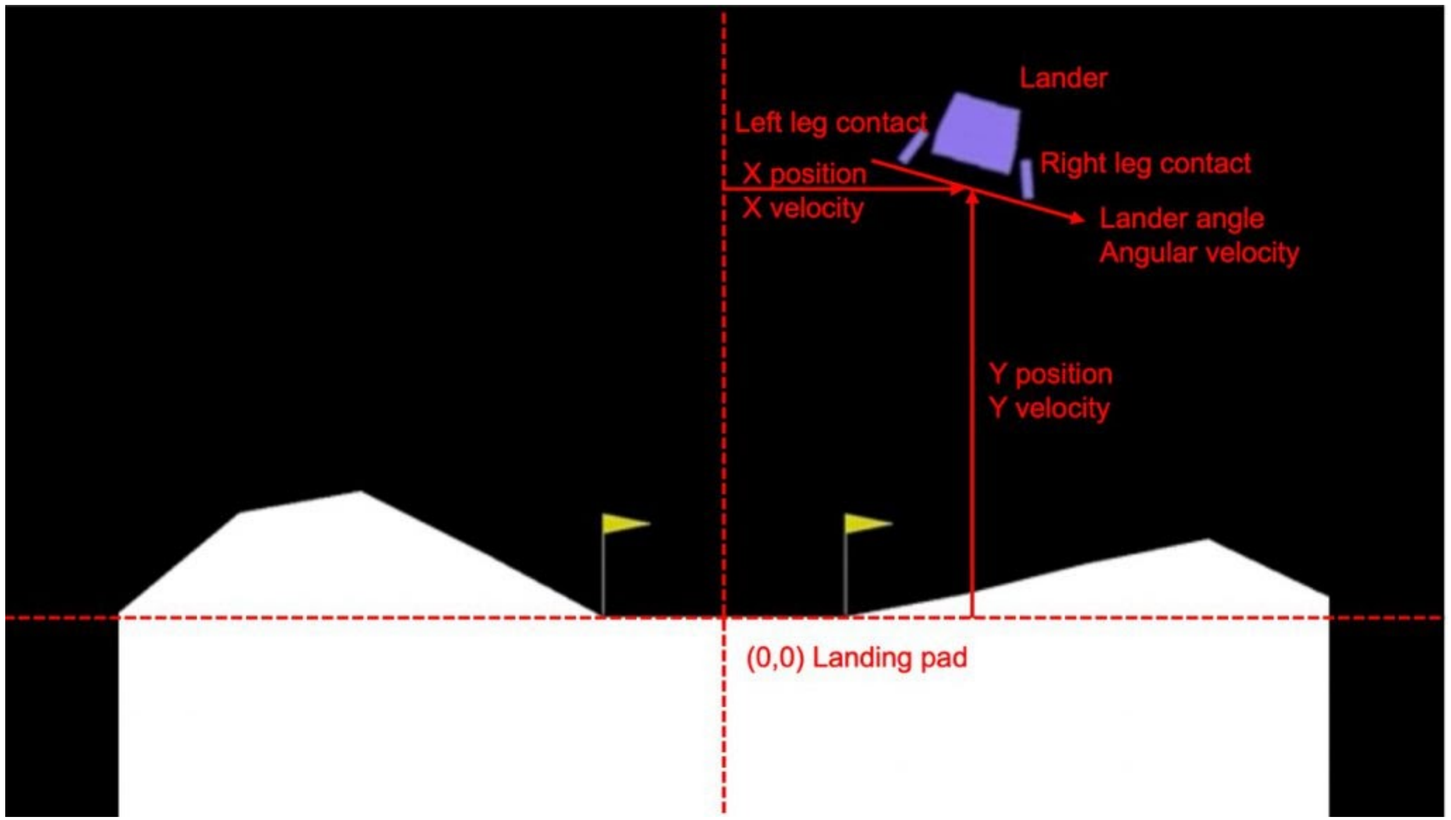
Acciones

- Activar propulsor derecho
- Activar propulsor izquierdo
- Activar propulsor principal (abajo)
- No activar propulsores

Recompensas

- Chocar -100, aterrizar +100, activar propulsor principal -3 (por paso de tiempo), etc.

LunarLander



Observación

- Posición en x
- Posición en y
- Velocidad en x
- Velocidad en y
- Ángulo de aterrizaje
- Velocidad angular
- Indicador de contacto en pata izquierda
- Indicador de contacto en pata derecha

```
[-0.00619364  1.4174144  -0.6273631   0.28862926  0.00718367  0.14210723   0.          0.        ]
```

LunarLander

- Descargar y abrir el notebook taller_rl.ipynb en el ambiente deeprl