# Aprendizaje por Refuerzo

Materia: Redes Neuronales.

Abril 2020

# Bibliografía


- Hands-on ML with Sk-learn & TensorFlow. (Géron)
- Reinforcement Learning: an introduction. (Sutton & Barto)
- Foundations of ML. (Mohri)
- Algorithms for RL. (Szepesvari)
- https://storage.googleapis.com/deepmind-media/dqn/DQNNaturePaper.pdf


- Link:

https://drive.google.com/drive/folders/1HD-ty2o09S4SpappK4LVK8rPdpxA5Zud?usp=sharing

# Aprendizaje por refuerzo (RL)

1. Campo de Machine Learning (ML) – 1950

# Aprendizaje por refuerzo (RL)

1. Campo de Machine Learning (ML) – 1950
2. Objetivo: Mapear situaciones a acciones (¿Qué debo hacer?) para maximizar una recompensa.

# Aprendizaje por refuerzo (RL)

1. Campo de Machine Learning (ML) – 1950
2. Objetivo: Mapear situaciones a acciones (¿Qué debo hacer?) para maximizar una recompensa.
3. Estudio de planificación y aprendizaje en un escenario.

# Aprendizaje por refuerzo (RL)

1. Campo de Machine Learning (ML) – 1950

2. Objetivo: Mapear situaciones a acciones (¿Qué debo hacer?) para maximizar una recompensa.

3. Estudio de planificación y aprendizaje en un escenario.

4. Elementos: Un aprendiz (**agente)** interactúa con un **ambiente** para llegar a un objetivo medido en **recompensa.**

# Aprendizaje por refuerzo (RL)

1. Campo de Machine Learning (ML) – 1950

2. Objetivo: Mapear situaciones a acciones (¿Qué debo hacer?) para maximizar una recompensa.

3. Estudio de planificación y aprendizaje en un escenario.

4. Elementos: Un aprendiz (**agente)** interactúa con un **ambiente** para llegar a un objetivo medido en **recompensa.**

5. Es un enfoque distinto a los aprendizajes supervisado y no supervisados (SL, UL).

   Se recibe un data set pasivamente desde un supervisor externo. **La interacción provee datos activamente.**

   SL: (Descripción X, Etiqueta Y) => Extrapolar.
   UL: (Datos X) => Estructura oculta.
   RL: Interacción => Maximizar una recompensa.
   No hay representantes a todas las situaciones. Es necesaria, la experiencia.

# Aprendizaje por refuerzo (RL)

1. Campo de Machine Learning (ML) – 1950

2. Objetivo: Mapear situaciones a acciones (¿Qué debo hacer?) para maximizar una recompensa.

3. Estudio de planificación y aprendizaje en un escenario.

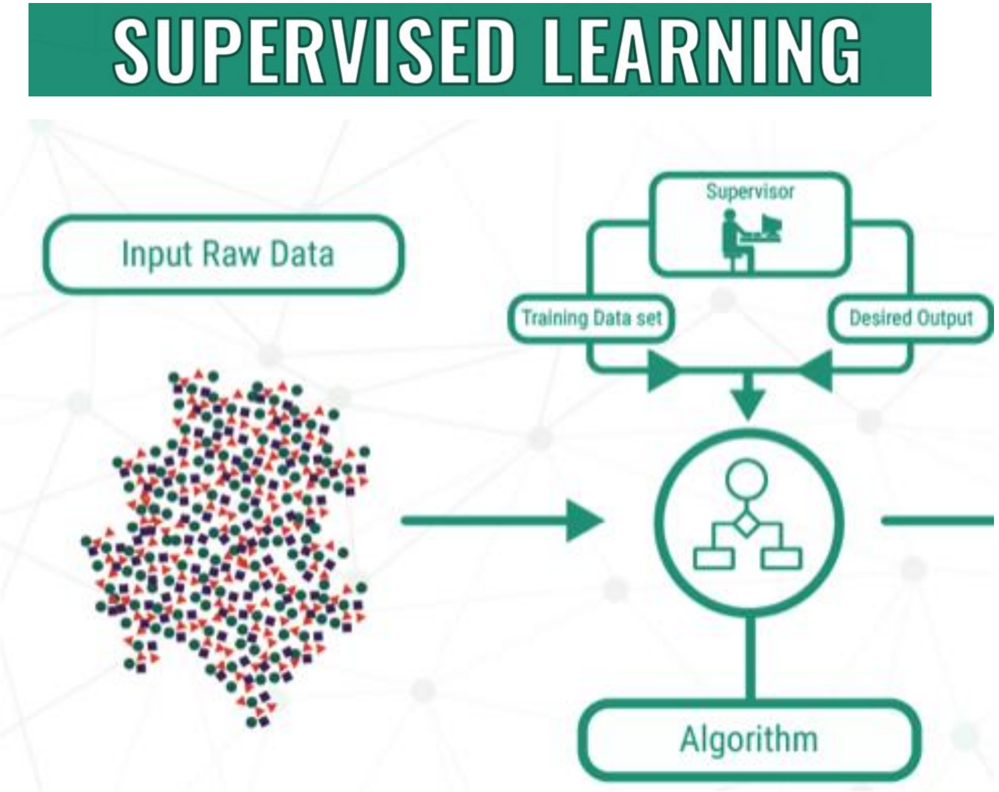
4. Elementos: Un aprendiz (**agente)** interactúa con un **ambiente** para llegar a un objetivo medido en **recompensa.**

5. Es un enfoque distinto a los aprendizajes supervisado y no supervisados (SL, UL).

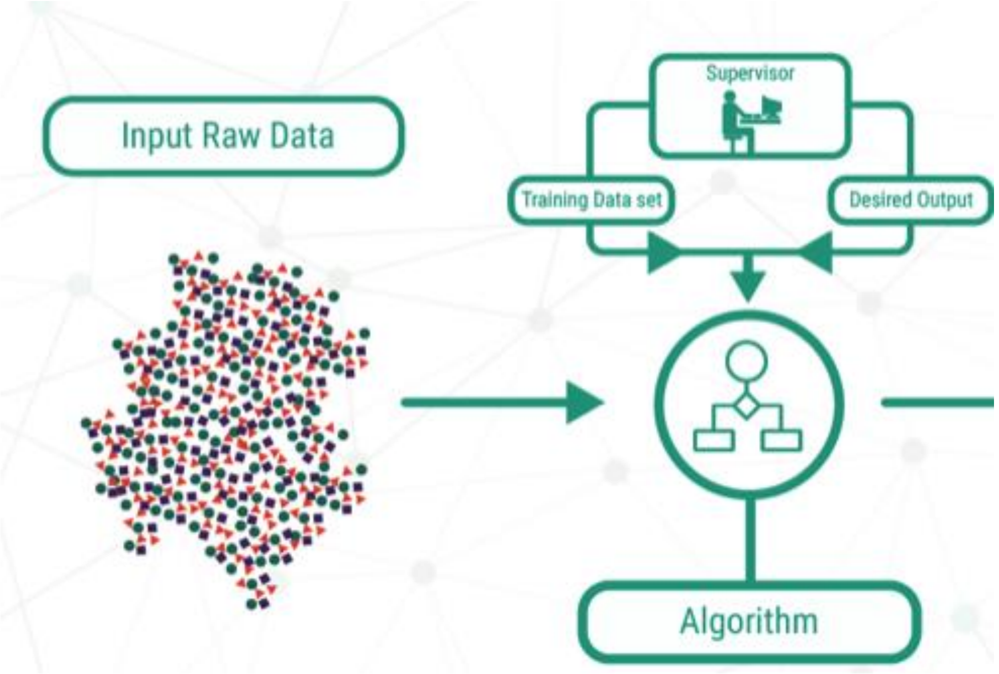
   Se recibe un data set pasivamente desde un supervisor externo. **La interacción provee datos activamente.**

   SL: (Descripción X, Etiqueta Y) => Extrapolar.
   UL: (Datos X) => Estructura oculta.
   RL: Interacción => Maximizar una recompensa.
   No hay representantes a todas las situaciones. Es necesaria, la experiencia.

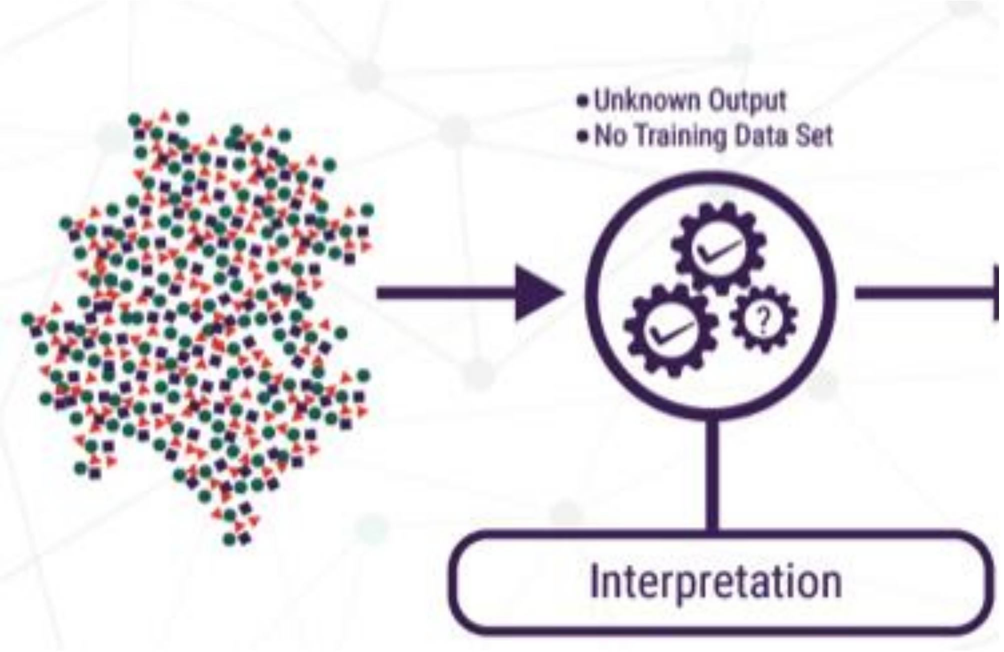
6. Aplicaciones en teoría de control, optimización, ciencia cognitivas. Games: Ataris, Alpha-Go


SUPERVISED LEARNING
Input Raw Data
Supervisor
Training Data set
Desired Output
Algorithm

# SUPERVISED LEARNING

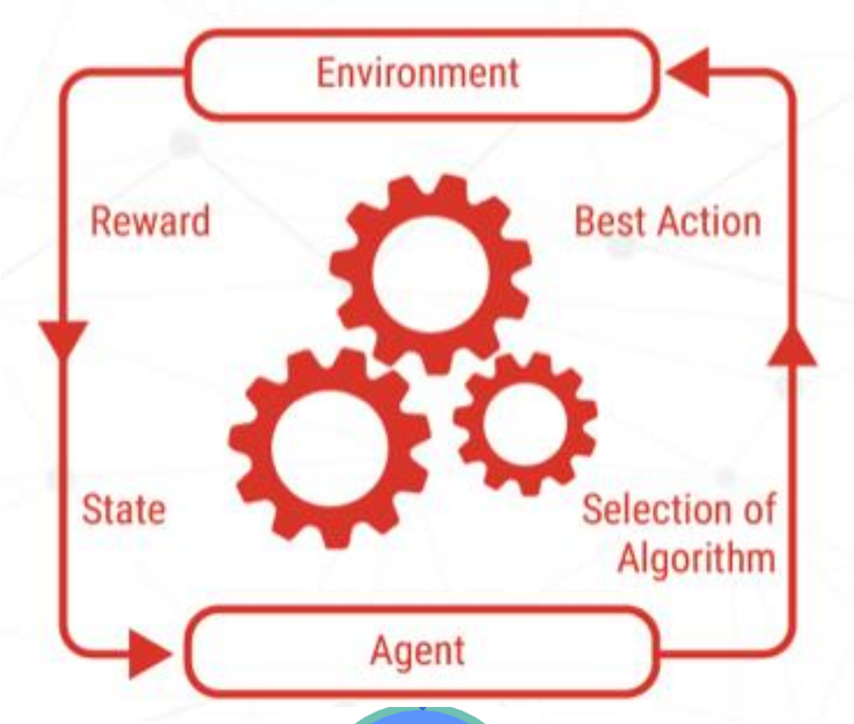



# UNSUPERVISED LEARNING

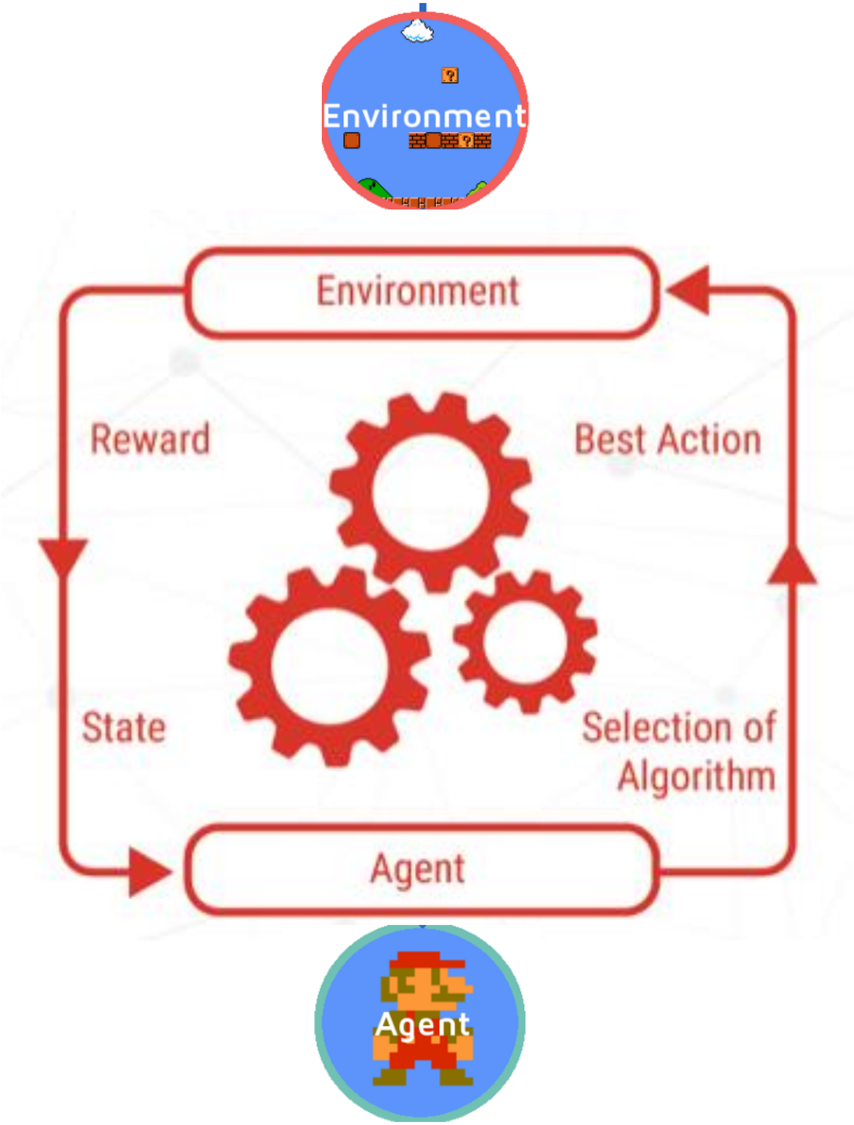


# REINFORCEMENT LEARNING

# REINFORCEMENT LEARNING

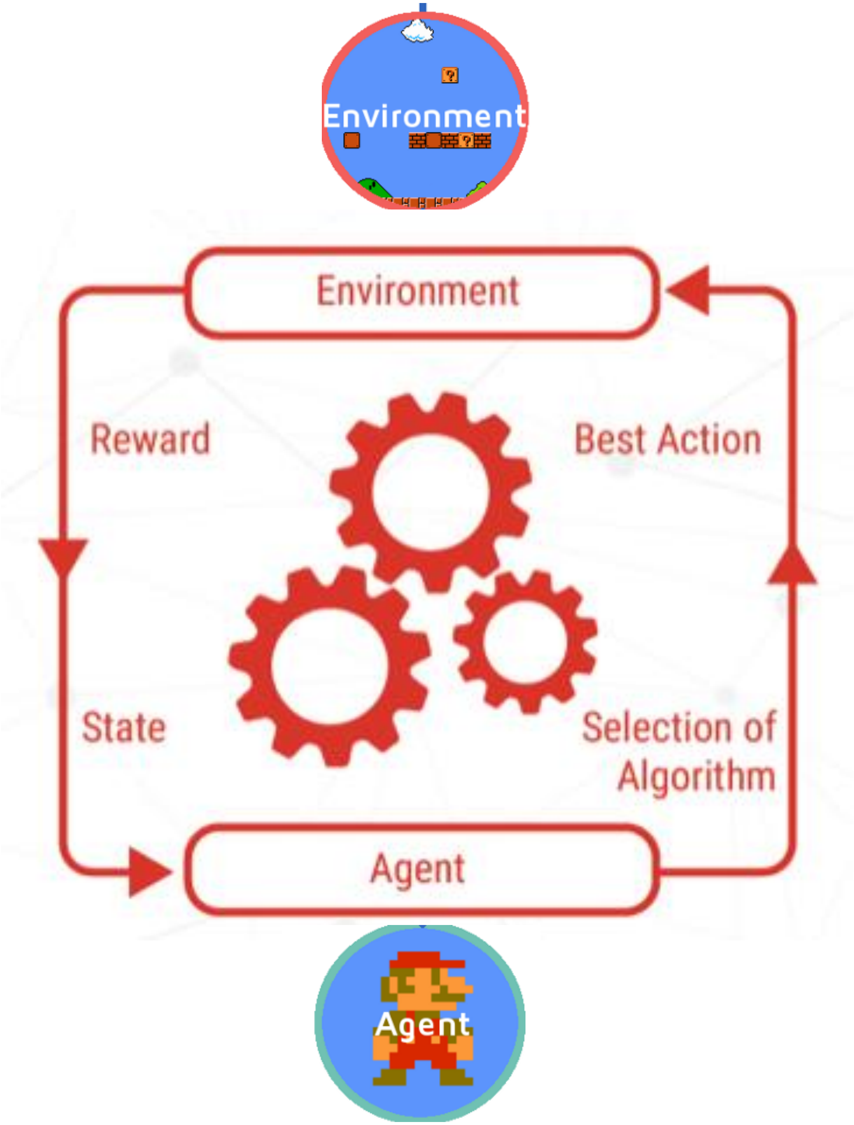




# REINFORCEMENT LEARNING




Agente:

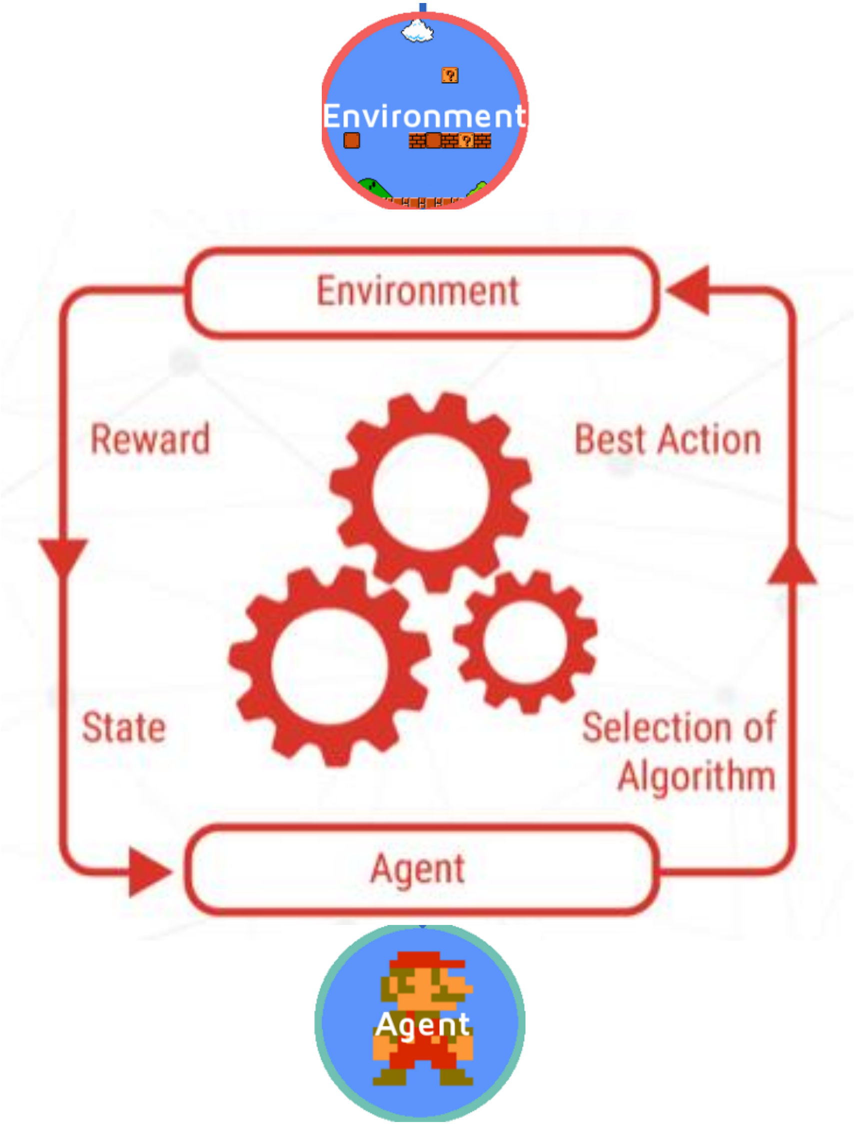
# REINFORCEMENT LEARNING




**Agente:** Controlador del personaje Mario.

**Ambiente:**

# REINFORCEMENT LEARNING

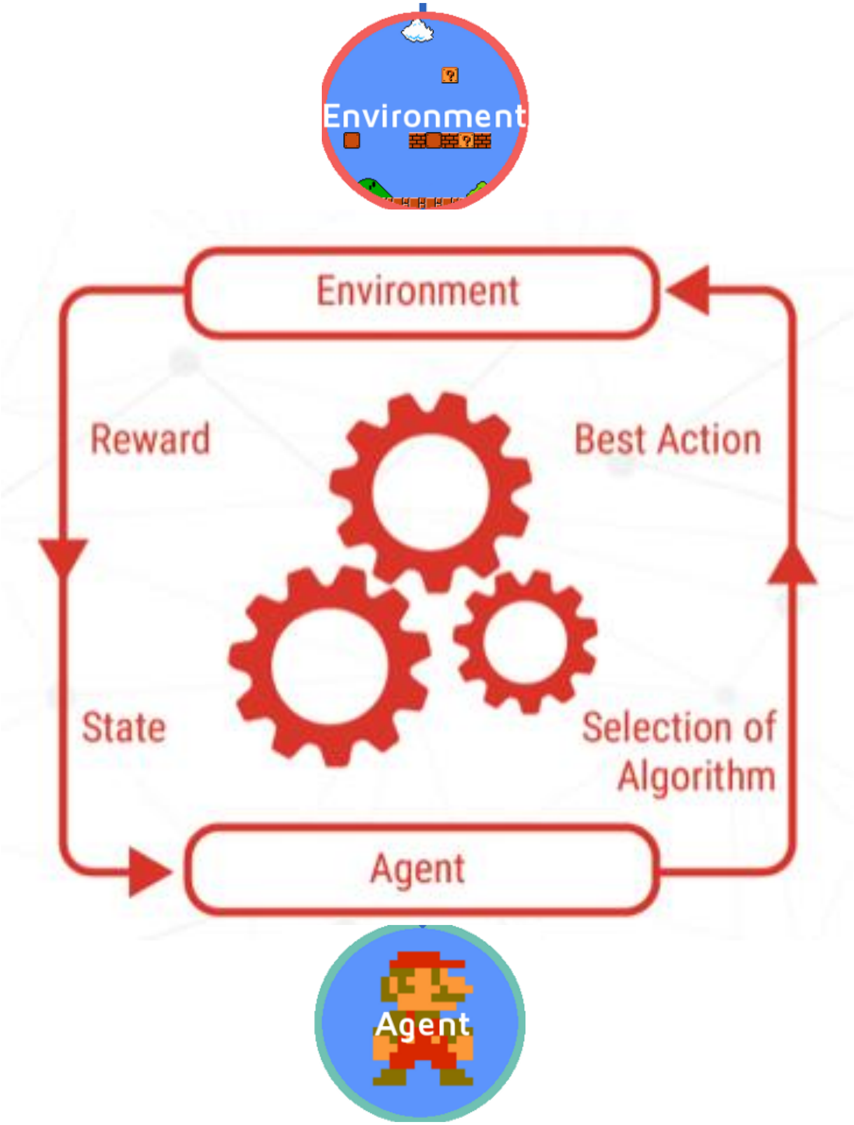



**Agente:** Controlador del personaje Mario.

**Ambiente:** Simulación del entorno del juego Mario Bros.

**Observación:**

# REINFORCEMENT LEARNING



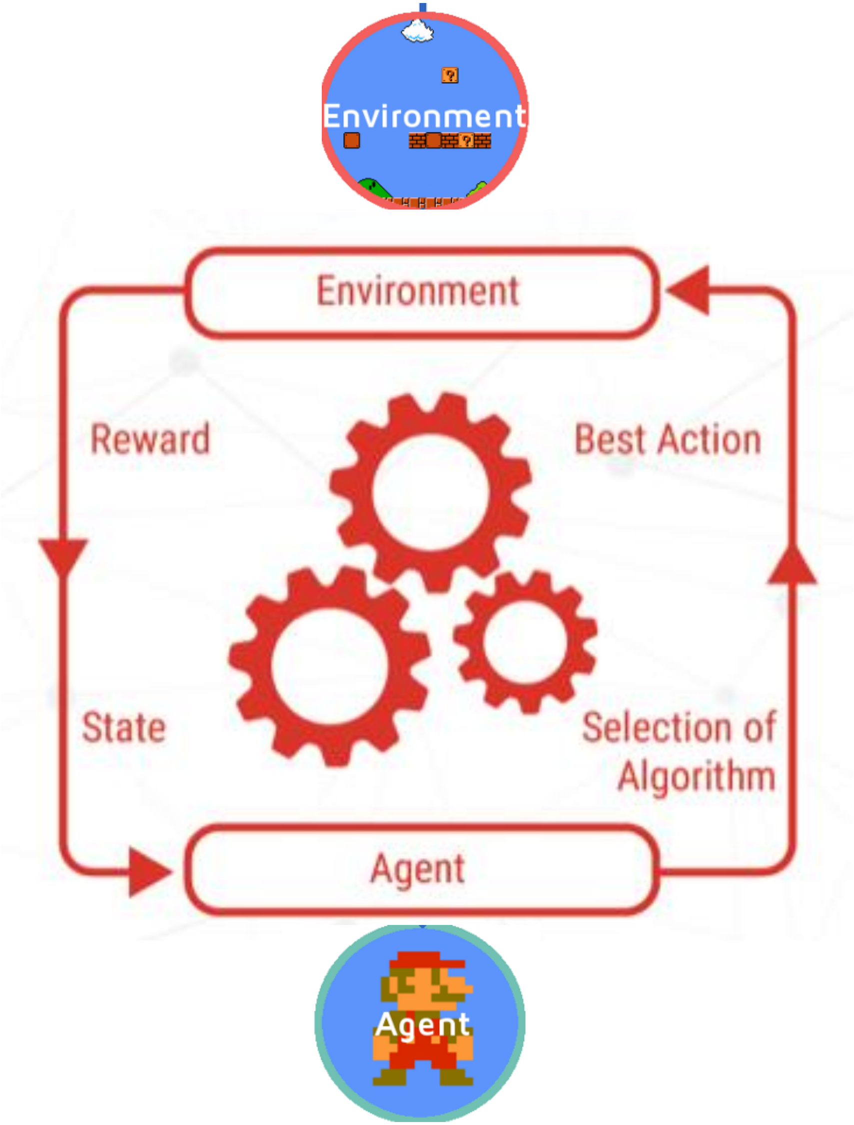

**Agente:** Controlador del personaje Mario.

**Ambiente:** Simulación del entorno del juego Mario Bros.

**Observación**: Screenshot – Imagen

**Acción**:

# REINFORCEMENT LEARNING




**Agente:** Controlador del personaje Mario.

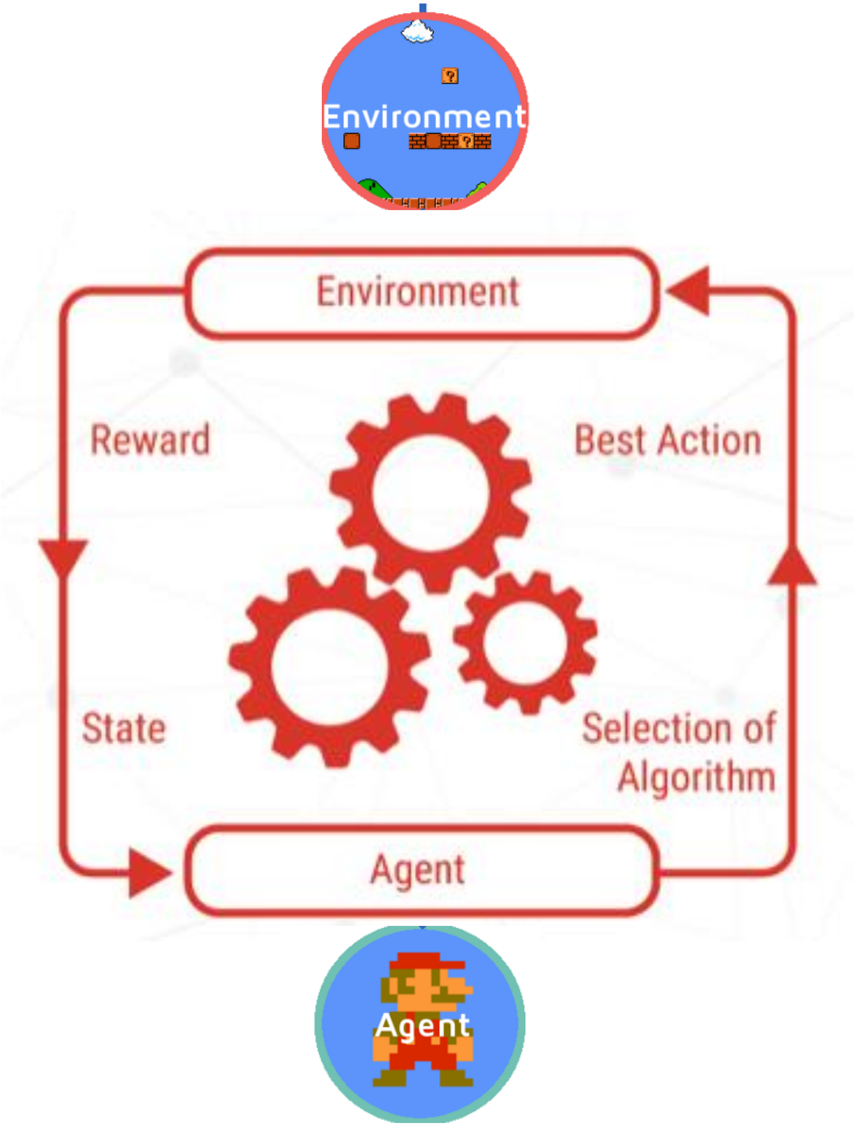
**Ambiente:** Simulación del entorno del juego Mario Bros.

**Observación**: Screenshot – Imagen

**Acción**: Posiciones del Joystick ( △ ▽ ▷ ◁)

**Recompensa:**

# REINFORCEMENT LEARNING




**Agente:** Controlador del personaje Mario.

**Ambiente:** Simulación del entorno del juego Mario Bros.

**Observación**: Screenshot – Imagen

**Acción**: Posiciones del Joystick ( △ ▽ ▷ ◁)

**Recompensa:** Puntos del juego (Cantidad de vidas, Monedas)

# Política

# Política

$\pi$

Es el algoritmo usado por el agente para las acciones a tomar.

Mapeo entre observación y la acción elegida.

# Política

$\pi$

Es el algoritmo usado por el agente para las acciones a tomar.

Mapeo entre observación y la acción elegida.

**PROBLEMA 1:**

Un programa controla la caminata de un robot (agente) y observa su alrededor (ambiente) mediante cámaras (observación). Las acciones tomadas por el agente son señales que activan el sistema motor del robot.

# Política

$\pi$

Es el algoritmo usado por el agente para las acciones a tomar.

Mapeo entre observación y la acción elegida.

**PROBLEMA 1:**

Un programa controla la caminata de un robot (agente) y observa su alrededor (ambiente) mediante cámaras (observación). Las acciones tomadas por el agente son señales que activan el sistema motor del robot.

Ejemplos de Políticas:

**Determinista, sin considerar la observación:**

**Determinista, considerando la observación:**

**Estocástica, sin considerar la observación:**

**Estocástica, considerando la observación:**

# Política

$\pi$

Es el algoritmo usado por el agente para las acciones a tomar.

Mapeo entre observación y la acción elegida.

PROBLEMA 1:

Un programa controla la caminata de un robot (agente) y observa su alrededor (ambiente) mediante cámaras (observación). Las acciones tomadas por el agente son señales que activan el sistema motor del robot.

Ejemplos de Políticas:

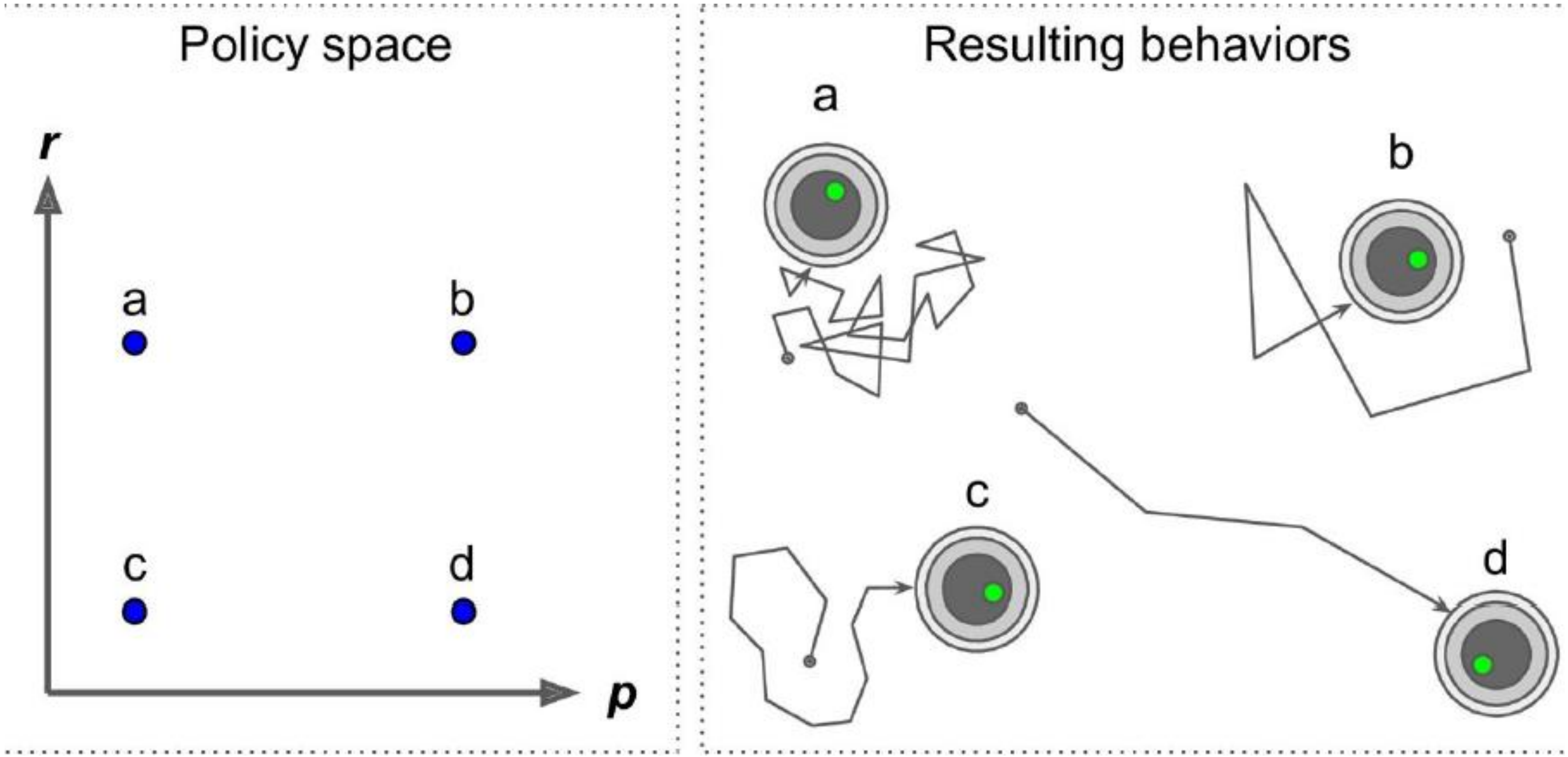
**Determinista, sin considerar la observación:**
A todo tiempo, el robot camina derecho (no dobla).

**Determinista, considerando la observación:**
El robot caminará derecho si no hay obstáculos al frente. En caso contario, girará 90º a la izquierda.

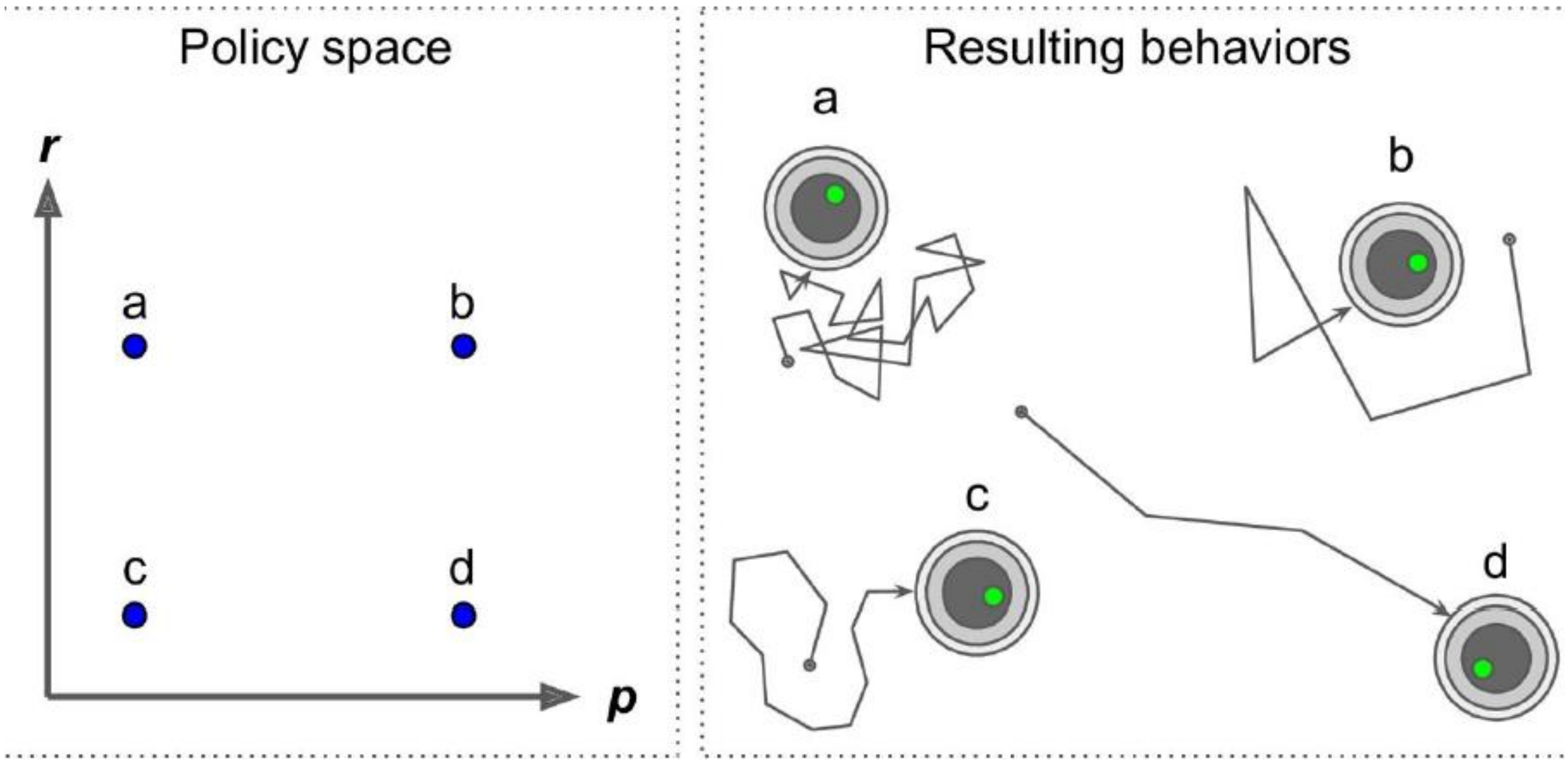
**Estocástica, sin considerar la observación:**
El robot caminará derecho con probabilidad **p** ó girará **r** $\in$**[π,-π]** con probabilidad **1-p**

**Estocástica, considerando la observación:**
El robot caminará derecho si no hay obstáculos al frente. En otro caso, girará un ángulo random.

# Política

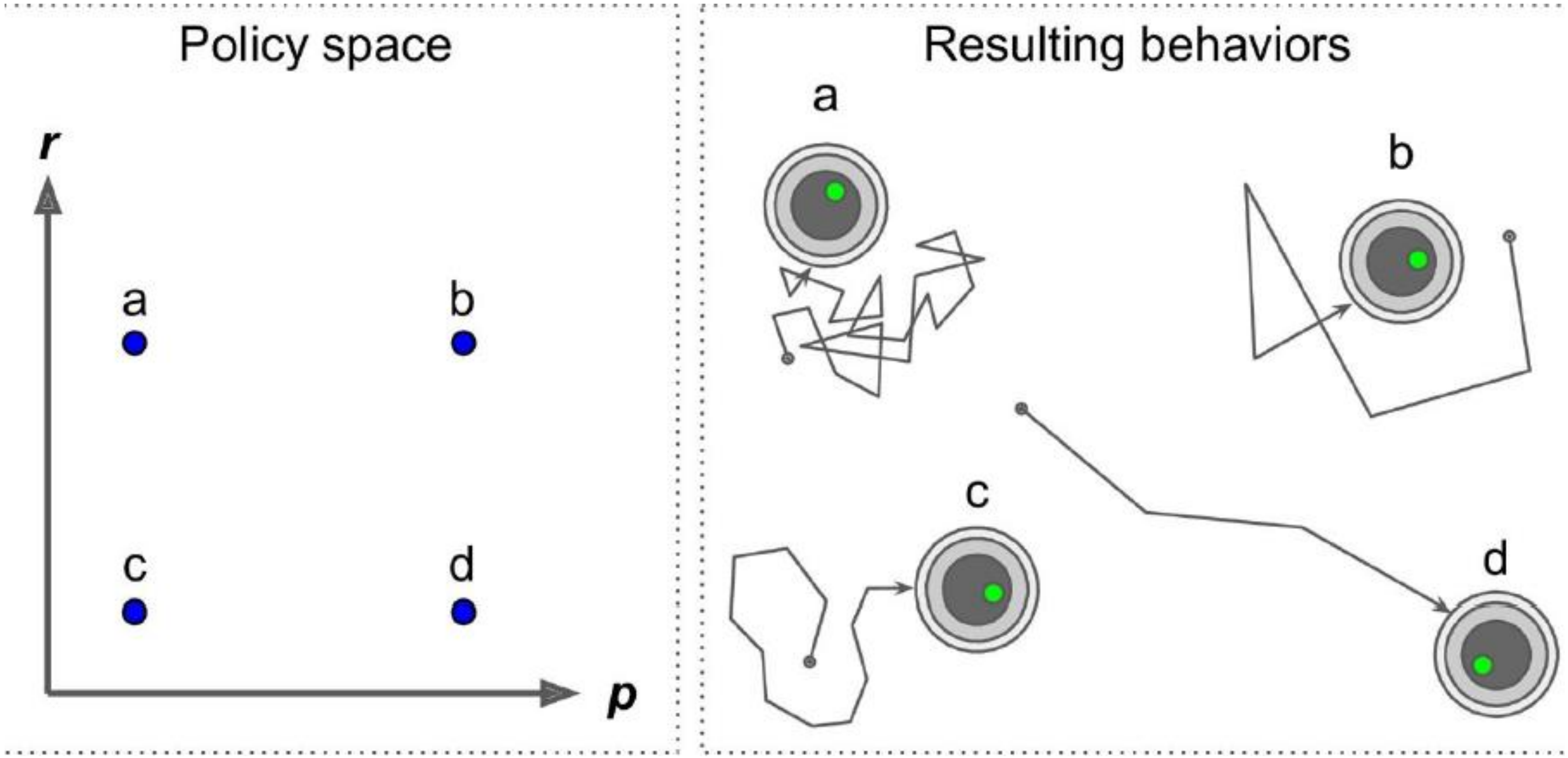


# Política




PROBLEMA 2: Si la recompensa disminuye con el tiempo de llegada a un punto fijo desde otro dado, ¿Cuáles son los valores de (r,p) óptimos?

# Política




PROBLEMA 2: Si la recompensa disminuye con el tiempo de llegada a un punto fijo desde otro dado, ¿Cuáles son los valores de **(r,p)** óptimos?

$$(r,p)^* = argmax_{(r,p)}\, R = argmin_{(r,p)}\, T$$

# Política

$\pi$

$$(r,p)^* = argmax_{(r,p)}\, R = argmin_{(r,p)}\, T$$

# Política

$$(r,p)^* = argmax_{(r,p)}\, R = argmin_{(r,p)}\, T$$

PROBLEMA 3:
* ¿Cómo se resuelve analíticamente este problema?

# Política

$$(r,p)^* = argmax_{(r,p)}\, R = argmin_{(r,p)}\, T$$

**PROBLEMA 3:**
*** ¿Cómo se resuelve analíticamente este problema?**

La búsqueda de máximos de funciones multivariables se basa en resolver el gradiente nulo y hessiana definida negativa.

# Política

$$(r,p)^* = argmax_{(r,p)}\, R = argmin_{(r,p)}\, T$$

PROBLEMA 3:

*** ¿Cómo se resuelve analíticamente este problema?**

La búsqueda de máximos de funciones multivariables se basa en resolver el gradiente nulo y hessiana definida negativa.

*** ¿Qué dificultad se presenta con el aumento de la dimensiones N de parámetros de la función de recompensa? ¿Qué métodos permiten solucionarlo?**

# Política

$$(r,p)^* = argmax_{(r,p)}\, R = argmin_{(r,p)}\, T$$

PROBLEMA 3:

*** ¿Cómo se resuelve analíticamente este problema?**

La búsqueda de máximos de funciones multivariables se basa en resolver el gradiente nulo y hessiana definida negativa.

*** ¿Qué dificultad se presenta con el aumento de la dimensiones N de parámetros de la función de recompensa? ¿Qué métodos permiten solucionarlo?**

El numero de ecuaciones aumenta orden N para el gradiente y N^2 para la hessiana.

Los algoritmos genéticos permiten ir explorando los valores del espacio de parámetros inteligentemente.

# Política

$$(r,p)^* = argmax_{(r,p)}\ R = argmin_{(r,p)}\ T$$

PROBLEMA 3:
*** ¿Cómo se resuelve analíticamente este problema?**

La búsqueda de máximos de funciones multivariables se basa en resolver el gradiente nulo y hessiana definida negativa.

*** ¿Qué dificultad se presenta con el aumento de la dimensiones N de parámetros de la función de recompensa? ¿Qué métodos permiten solucionarlo?**

El numero de ecuaciones aumenta orden N para el gradiente y N^2 para la hessiana.

Los algoritmos genéticos permiten ir explorando los valores del espacio de parámetros inteligentemente.

*** Suponga que es capaz, de encontrar los parámetros que cumplen la optimización. ¿Puede asegurar que es la política óptima (es la que maximiza la recompensa)?**

# Política

$\pi$

$$(r,p)^* = argmax_{(r,p)}\, R = argmin_{(r,p)}\, T$$

PROBLEMA 3:

*** ¿Cómo se resuelve analíticamente este problema?**

La búsqueda de máximos de funciones multivariables se basa en resolver el gradiente nulo y hessiana definida negativa.

*** ¿Qué dificultad se presenta con el aumento de la dimensiones N de parámetros de la función de recompensa? ¿Qué métodos permiten solucionarlo?**

El numero de ecuaciones aumenta orden N para el gradiente y N^2 para la hessiana.

Los algoritmos genéticos permiten ir explorando los valores del espacio de parámetros inteligentemente.

*** Suponga que es capaz, de encontrar los parámetros que cumplen la optimización. ¿Puede asegurar que es la política óptima (es la que maximiza la recompensa)?**

No, pues solo nos estamos centrando en un subconjunto del espacio de políticas.

POLÍTICA es una función que para la observación actual del ambiente devuelve la acción a tomar.

$\pi$

PROBLEMA 4: ¿Puede la Política depender de las observaciones previas? ¿En qué caso pasa?

$\pi$

POLÍTICA es una función que para la observación actual del ambiente devuelve la acción a tomar.

PROBLEMA 4: ¿Puede la Política depender de las observaciones previas? ¿En qué caso pasa?

Cuando una observación no tenga toda la información del sistema, existen variables ocultas. En este caso, necesitaremos hallar las variables ocultas usando información de otras observaciones.

**Ejemplo:** Sea un sistema caracterizado por la posición y velocidad **(x,v)** de un objeto; pero, cada observación solo es sobre la posición. En este caso, la variable oculta velocidad $v_t$podemos calcularla como $x_t - x_{t-1}$.

La función política dependerá de las observaciones actual y anterior.

π

POLÍTICA es una función que para la observación actual del ambiente devuelve la acción a tomar.

**PROBLEMA 4: ¿Puede la Política depender de las observaciones previas? ¿En qué caso pasa?**

Cuando una observación no tenga toda la información del sistema, existen variables ocultas. En este caso, necesitaremos hallar las variables ocultas usando información de otras observaciones.

**Ejemplo:** Sea un sistema caracterizado por la posición y velocidad **(x,v)** de un objeto; pero, cada observación solo es sobre la posición. En este caso, la variable oculta velocidad $v_t$podemos calcularla como $x_t - x_{t-1}$.

La función política dependerá de las observaciones actual y anterior.

*** ¿Qué pasa cuando la observación es ruidosa?**

POLÍTICA es una función que para la observación actual del ambiente devuelve la acción a tomar.

$\pi$

**PROBLEMA 4: ¿Puede la Política depender de las observaciones previas? ¿En qué caso pasa?**

Cuando una observación no tenga toda la información del sistema, existen variables ocultas. En este caso, necesitaremos hallar las variables ocultas usando información de otras observaciones.

**Ejemplo:** Sea un sistema caracterizado por la posición y velocidad **(x,v)** de un objeto; pero, cada observación solo es sobre la posición. En este caso, la variable oculta velocidad $v_t$podemos calcularla como $x_t - x_{t-1}$.

La función política dependerá de las observaciones actual y anterior.

*** ¿Qué pasa cuando la observación es ruidosa?**

Se utiliza métodos de inferencia para la determinación del estado del sistema.

$\pi$

POLÍTICA es una función que para la observación actual del ambiente devuelve la acción a tomar.

**PROBLEMA 4: ¿Puede la Política depender de las observaciones previas? ¿En qué caso pasa?**

Cuando una observación no tenga toda la información del sistema, existen variables ocultas. En este caso, necesitaremos hallar las variables ocultas usando información de otras observaciones.

**Ejemplo:** Sea un sistema caracterizado por la posición y velocidad **(x,v)** de un objeto; pero, cada observación solo es sobre la posición. En este caso, la variable oculta velocidad $v_t$ podemos calcularla como $x_t - x_{t-1}$.

La función política dependerá de las observaciones actual y anterior.

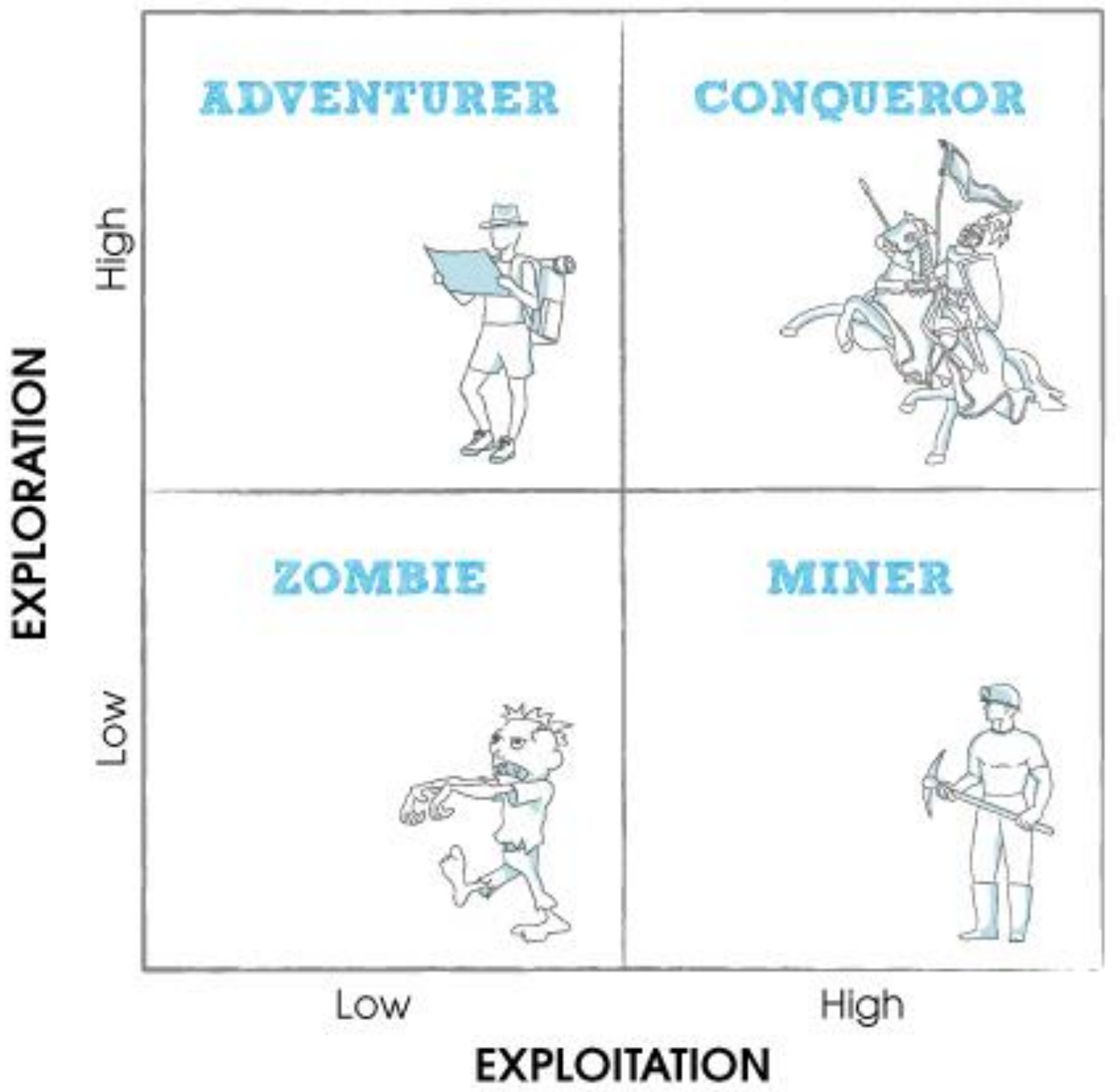
*** ¿Qué pasa cuando la observación es ruidosa?**

Se utiliza métodos de inferencia para la determinación del estado del sistema.

# OBSERVACIÓN = ESTADO COMPLETO DEL SISTEMA

$\pi$

POLÍTICA es una función que para la observación actual del ambiente devuelve la acción a tomar.



# Procesos de decisión markoviana.

# Procesos de decisión markoviana.

Un MDP está definido por:

# Procesos de decisión markoviana.

**Un MDP está definido por:**

- Un conjunto de estados S.

# Procesos de decisión markoviana.

**Un MDP está definido por:**

- Un conjunto de estados S.
- Una condición inicial $s_{t=0} \in S$.

# Procesos de decisión markoviana.

**Un MDP está definido por:**

- Un conjunto de estados S.
- Una condición inicial $s_{t=0} \in S$.
- Un conjunto de acciones A.

# Procesos de decisión markoviana.

**Un MDP está definido por:**

- Un conjunto de estados S.
- Una condición inicial $s_{t=0} \in S$.
- Un conjunto de acciones A.
- Una dinámica de transición P(s'|s,a)

# Procesos de decisión markoviana.

**Un MDP está definido por:**

- Un conjunto de estados S.
- Una condición inicial $s_{t=0} \in S$.
- Un conjunto de acciones A.
- Una dinámica de transición P(s’|s,a)
- Una función de recompensa R = R(s,a,s’)

# Procesos de decisión markoviana.

**Un MDP está definido por:**

- Un conjunto de estados S.
- Una condición inicial $s_{t=0} \in S$.
- Un conjunto de acciones A.
- Una dinámica de transición P(s’|s,a)
- Una función de recompensa R = R(s,a,s’)

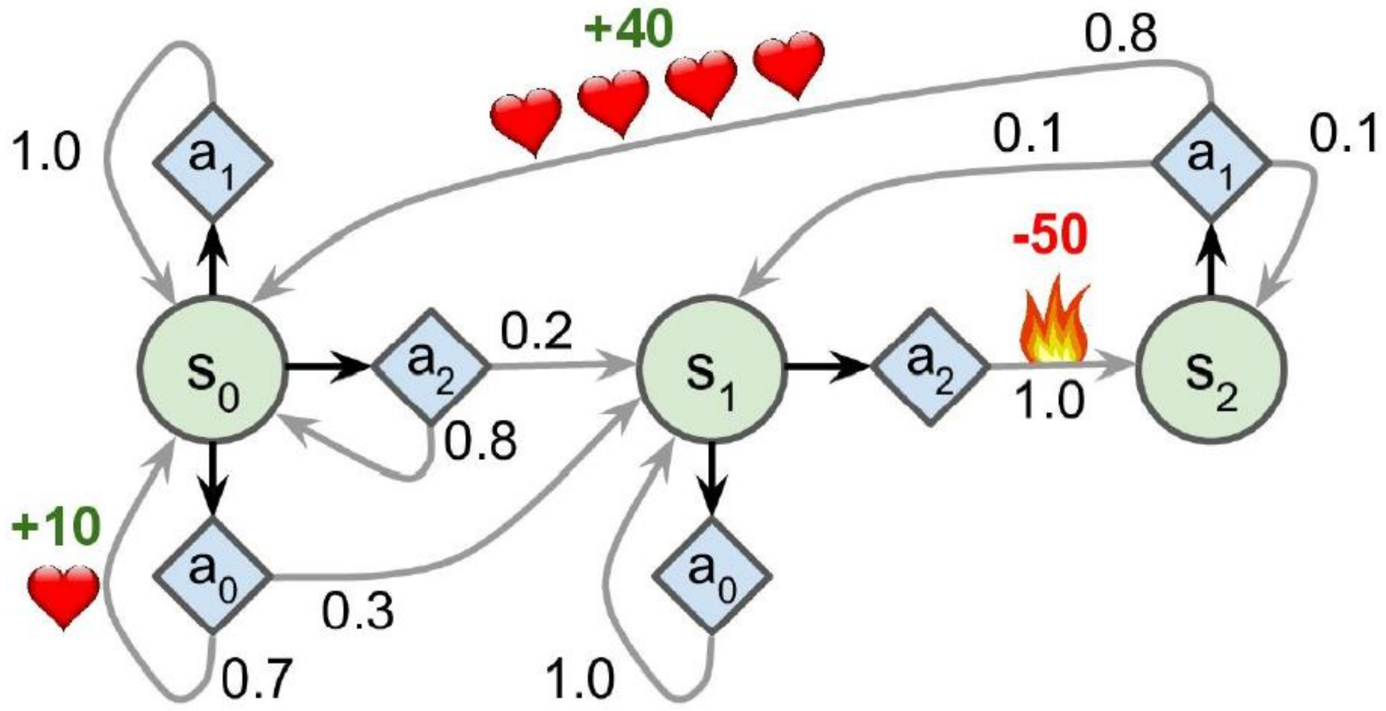
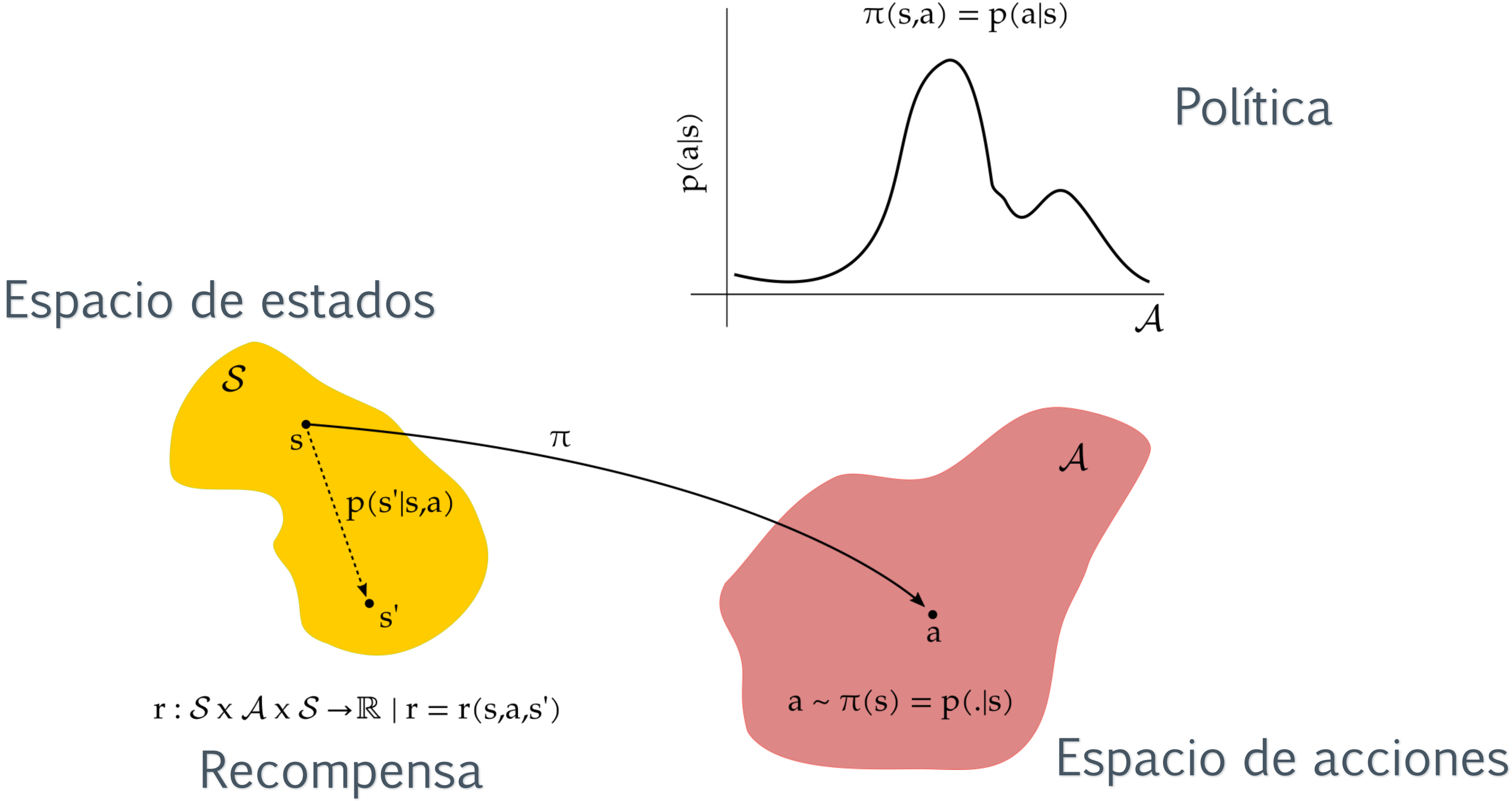


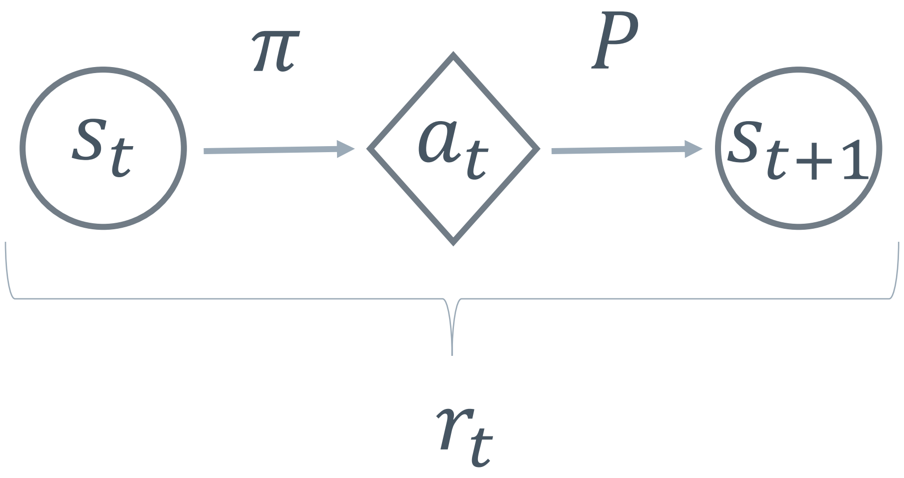

Figure 16-8. Example of a Markov decision process

# Procesos de decisión markoviana.



# Procesos de decisión markoviana.

# Procesos de decisión markoviana.



# Procesos de decisión markoviana.

$$\underbrace{s_t \xrightarrow{\pi} a_t \xrightarrow{P} s_{t+1}}_{r_t}$$

Recompensa acumulada
(variable aleatoria)

$$\boxed{R_t^{\pi} = \sum_{n=0}^{\infty} \gamma^n \, r_{t+n}}$$

# Procesos de decisión markoviana.

$$\underbrace{s_t \xrightarrow{\pi} a_t \xrightarrow{P} s_{t+1}}_{r_t}$$

Recompensa acumulada
(variable aleatoria)

$$\boxed{R_t^\pi = \sum_{n=0}^{\infty} \gamma^n \, r_{t+n}}$$

Value function

$$\boxed{V^\pi(s) = \mathrm{E}[R_t^\pi | s_t = s]}$$

# Procesos de decisión markoviana.

$$\underbrace{s_t \xrightarrow{\pi} a_t \xrightarrow{P} s_{t+1}}_{r_t}$$

Recompensa acumulada
(variable aleatoria)

$$\boxed{R_t^\pi = \sum_{n=0}^{\infty} \gamma^n \, r_{t+n}}$$

Value function

$$\boxed{V^\pi(s) = \mathrm{E}[R_t^\pi | s_t = s]}$$

<u>PROBLEMA</u>: Deducir la ecuación de Bellman (1)

$$V^\pi(s) = E[r(s, \pi(s), s')] + \gamma \sum_{s'} p(s' | s, \pi(s)) V^\pi(s')$$

# Procesos de decisión markoviana.

Q function

$$\boxed{Q^{\pi}(s,a) = \mathrm{E}[R_t^{\pi} | s_t = s, a_t = a]}$$

# Procesos de decisión markoviana.

Q function

$$\boxed{Q^{\pi}\,(s,a) = \mathrm{E}[R_t^{\pi} | s_t = s, a_t = a]}$$

<u>PROBLEMA</u>: Deducir la ecuación de Bellman (2)

$$Q^{\pi}(s,a) = E[r(s,a,s')] + \gamma \sum_{s'} p(s'|s,a) V^{\pi}(s')$$

# Procesos de decisión markoviana.

Q function

$$\boxed{Q^{\pi}\,(s,a) = \mathrm{E}[R_t^{\pi}|s_t = s, a_t = a]}$$

<u>PROBLEMA</u>: Deducir la ecuación de Bellman (2)

$$Q^{\pi}(s,a) = E[r(s,a,s')] + \gamma \sum_{s'} p(s'|s,a)V^{\pi}(s')$$

<u>PROBLEMA</u>: Deducir la relación

$$Q^{\pi}(s,\pi(s)) = V^{\pi}(s)$$

# Política óptimas

Una política **π** es óptima si maximiza V

$$\boxed{V^*(s) = V^{\pi*}(s) = max_{\pi}\, V^{\pi}(s)}$$

# Política óptimas

Una política **π** es óptima si maximiza V

$$\boxed{V^*(s) = V^{\pi *}(s) = max_{\pi}\, V^{\pi}(s)}$$

$$Q^*\ (s, \pi^*(s)) = V^*(s)$$

# Política óptimas

Una política **π** es <u>óptima</u> si maximiza V

$$\boxed{V^*(s) = V^{\pi*}(s) = max_{\pi}\, V^{\pi}(s)}$$

$$Q^*\,(s, \pi^*(s)) = V^*(s)$$

Equivalentemente,

$$max_a\, Q^*(s, a) = V^*(s)$$

# Política óptimas

**Una política π es <u>óptima</u> si maximiza V**

$$\boxed{V^*(s) = V^{\pi *}(s) = max_{\pi}\, V^{\pi}(s)}$$

$$Q^*\,(s, \pi^*(s)) = V^*(s)$$

Equivalentemente,

$$max_a\, Q^*(s, a) = V^*(s)$$

Entonces, la política óptima se puede calcular

$$\pi^*(s) = argmax_a\, Q^*(s, a)$$

# Política óptimas

PROBLEMA: Desde (2) deducir la ecuación de Bellman (3)

$$Q^*(s,a) = E[r(s,a,s')] + \gamma \sum_{s'} p(s'|s,a)\, max_{a'} Q^*(s',a')$$

# Política óptimas

<u>PROBLEMA</u>: Desde (2) deducir la ecuación de Bellman (3)

$$Q^*(s,a) = E[r(s,a,s')] + \gamma \sum_{s'} p(s'|s,a)\, max_{a'} Q^*(s',a')$$

Si resolvemos la ecuacĭón anterior (3), la política óptima es

$$\pi^*(s) = argmax_a\, Q^*(s,a)$$

# Algoritmos del RL

$$\boldsymbol{Q^*(s,a) = E[r(s,a,s')] + \gamma \sum_{s'} p(s'|s,a)\, max_{a'} Q^*(s',a')}$$

<u>Programación dinámica.</u>

(Value Iteration, Q Iteration)

Necesito conocer el MDP completo

<u>Monte Carlo</u>

Estimación de la dinámica de transición P y de la Recompensa acumulada.

No explota la propiedad de Markov

<u>Temporal Difference</u>

(Q-learning, SARSA)

Estimación de P

Explota la propiedad de Markov

# Algoritmos del RL: Temporal Difference.

SARSA (On Policy)

$$Q(S,A) \leftarrow Q(S,A) + \alpha[\underbrace{R + \gamma \underline{Q(S',A')} - Q(S,A)}]$$

Q-Learning (Off Policy)

$$Q(S,A) \leftarrow Q(S,A) + \alpha[\underbrace{R + \gamma \underline{\max_a Q(S',a)} - Q(S,A)}]$$

# Algoritmos del RL y Deep Learning.

$$\boxed{Q: S \times A \rightarrow R}$$

| | $a_0$ | $a_1$ | $a_2$ | … | | | | $a_M$ |
|---|---|---|---|---|---|---|---|---|
| $s_0$ | | | | | | | | |
| $s_1$ | | | | | | | | |
| $s_2$ | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |
| $s_N$ | | | | | | | | |

$$Q \in R^{|S| \times |A|}$$

$$|A| = M$$
$$|S| = N$$

# Algoritmos del RL y Deep Learning.

$$|A| = M$$
$$|S| \to \infty$$

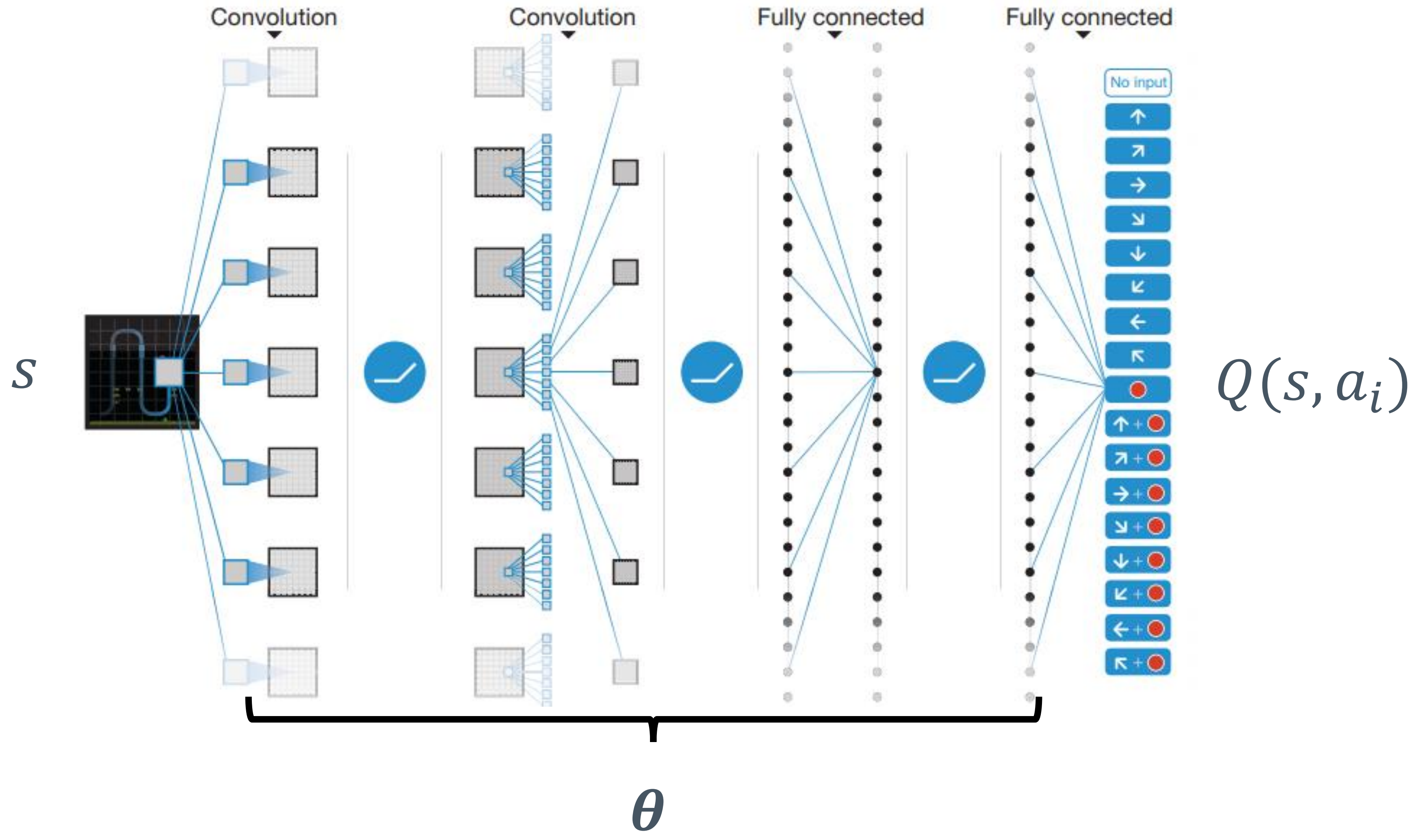
$$Q = \begin{bmatrix} Q(s, a_1) \\ \dots \\ Q(s, a_M) \end{bmatrix}$$

$$s \longrightarrow \boxed{\text{Black Box}} \longrightarrow \begin{bmatrix} Q(s, a_1) \\ \dots \\ Q(s, a_M) \end{bmatrix}$$

$\pi$

$|A| = M$

$|S| \to \infty$




$$L_i(\theta_i) = \mathbb{E}_{(s,a,r,s') \sim \mathrm{U}(D)}\left[\left(r + \gamma \max_{a'} Q(s',a';\theta_i^-) - Q(s,a;\theta_i)\right)^2\right]$$

Deep Q-Learning