

Clasificadores Probabilísticos en Aprendizaje Automático

Martin Nievas

DISCLAIMER: Algunos slides no son míos

Los



míos

van a poder identificar



COMIC SAAAAAAANNS!!



DRL

a.k.a. Deep Reinforcement Learning

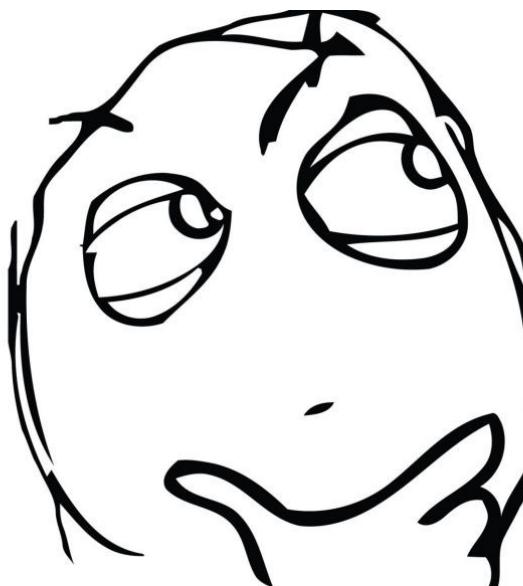
La Revolución del Deep Learning



Topics

The Poker-Playing AI That Beat the World's Best Players

By Edd Gent - Jul 15, 2019 • 3416



ultrad.com.br

n of strategy and intuition, something that's games and devilishly difficult for machines to ebook and Carnegie Mellon University has nals in a multiplayer version of the game for the

Stig Petersen¹, Charles Beattie¹, Amir Sadik¹, Ioannis Antonoglou¹,
g¹ & Demis Hassabis¹

| 529



AI4KI

ING MAGAZINE [82] NOVEMBER 2012

Necesitan financiamiento
para proyectos

Porqué no usarlo en todos lados?



NO Aplicaciones

Dónde No Podemos Usar (Solo) DL

- Problemas en los que sea necesario interpretar los modelos
 - DL tiende a generar “cajas negras” poco interpretables

Dónde No Podemos Usar (Solo) DL

- Problemas en los que tengamos pocos datos
 - O porque no tenemos datos suficientes...
 - Ejemplo: algunas aplicaciones médicas y forenses

Dónde No Podemos Usar (Solo) DL

- Problemas en los que sea necesario extrapolar porque su espacio de características es poco conocido
 - Ejemplo: generar imágenes nuevas a partir de conocidas Generative Adversarial Networks, Variational Autoencoders
- ... o porque el espacio es tan complejo que no hay suficientes datos (aún)
 - Ejemplo: RL en tareas poco definidas (coche autónomo)

Su salida es una probabilidad entre clases

Objetivos del Curso

- Entender los fundamentos de los clasificadores probabilísticos
 - Probabilidad, estadística bayesiana
- Definir y entender el funcionamiento de un clasificador probabilístico
 - Entendido en sentido amplio Entran datos → salen probabilidades
- Medir cuándo un clasificador probabilístico está funcionando “bien”
 - Es decir, cuándo tomamos buenas decisiones con él
 - Término clave: calibración ← IMPORTANTE (y poco conocido)
- Aprender métodos para mejorar el funcionamiento de un clasificador probabilístico
 - Calibración intrínseca
 - Calibración extrínseca

← Como solucionar el problema

¿Hay Algo Más Allá del DL/DRL?

- “The Bitter Lesson”, de Rich Sutton



- “1) AI researchers have often tried to build knowledge into their agents
 - 2) this always helps in the short term, and is personally satisfying to the researcher, but

<http://www.incompleteideas.net/Inclideas/BitterLesson.html>

¿Hay Algo Más Allá del DL/DRL?

- “The Bitter Lesson”, de Rich Sutton



- “1) AI researchers have often tried to build knowledge into their agents
 - 2) this always helps in the short term, and is personally satisfying to the researcher, but
 - 3) in the long run it plateaus and even inhibits further progress, and

<http://www.incompleteideas.net/Incldeas/BitterLesson.html>

¿Hay Algo Más Allá del DL/DRL?

- “The Bitter Lesson”, de Rich Sutton



- “1) AI researchers have often tried to build knowledge into their agents
 - 2) this always helps in the short term, and is personally satisfying to the researcher, but
 - 3) in the long run it plateaus and even inhibits further progress, and
 - 4) breakthrough progress eventually arrives by an opposing approach based on **scaling computation by search and learning.**”

<http://www.incompleteideas.net/Incldeas/BitterLesson.html>

¿Hay Algo Más Allá del DL/DRL?

- “¿Modelos o solo datos y cómputo?”, de Max Welling



- “The most fundamental lesson of ML is the **bias-variance tradeoff**: when you have **sufficient data**, you do not need to impose a lot of human generated inductive bias on your model. You can *let the data speak*.
 - “However, when you **do not have sufficient data** available you will need to use **human-knowledge** to **fill the gaps**.”
 - This is precisely what happens when you extrapolate: you enter a new input domain where you have very sparse data and your trained model will start to fail.”

<https://staff.fnwi.uva.nl/m.welling/wp-content/uploads/Model-versus-Data-AI.pdf>

¿Hay Algo Más Allá del DL/DRL?

- “¿Modelos o solo datos y cómputo?”, de Max Welling



- “For narrowly defined domains with enough data or a really accurate simulator, you can train a discriminative model and do very well.”

<https://staff.fnwi.uva.nl/m.welling/wp-content/uploads/Model-versus-Data-AI.pdf>

Recordemos:

- Clasificador probabilístico

Clasificador Probabilístico: Modelos

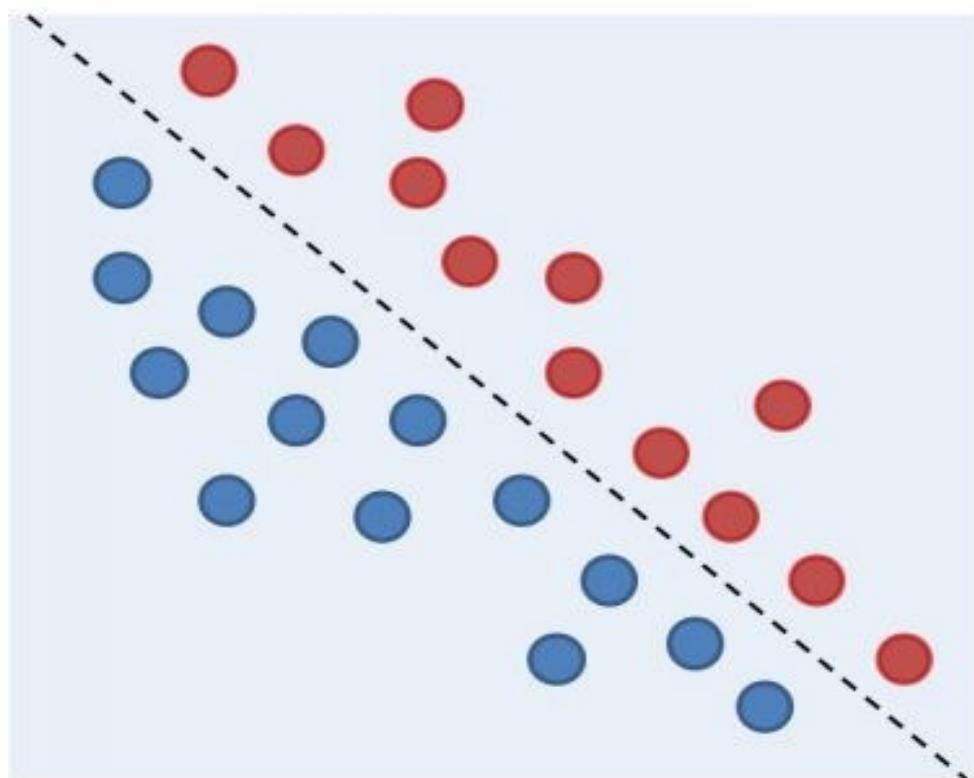
- Modelos probabilísticos **discriminativos**
 - Intentan obtener la probabilidad de interés directamente
 - $p(\mathbf{Z}|\mathbf{X})$
- Ventajas:
 - Obtengo la información que necesito para mi decisión
 - Sin preocuparme de nada más
 - Ej: DNNs
- Inconvenientes
 - Se pierde la información del modelo generador $p(\mathbf{X}|\mathbf{Z})$
 - Tiene ventajas conocerlo (ver discusión Sutton-Welling)
 - No permiten generar datos nuevos (*data augmentation*)

Clasificador Probabilístico: Modelos

- Modelos probabilísticos **generativos**
 - Se intenta obtener la representación probabilística completa
 - O bien buscando la probabilidad generadora $p(\mathbf{X}|\mathbf{Z})$
 - O bien modelando el problema completo $p(\mathbf{X}, \mathbf{Z})$
 - Y obteniendo más adelante $p(\mathbf{X}|\mathbf{Z})$, $p(\mathbf{Z}|\mathbf{X})$, $p(\mathbf{X})$
- Ventaja: problema probabilístico definido completamente
 - $p(\mathbf{X})$: *marginal likelihood*: ajuste del modelo a los datos
 - Selección de modelos
 - Detección de *outliers*
 - ...
- Inconvenientes
 - Es el problema más complejo de todos

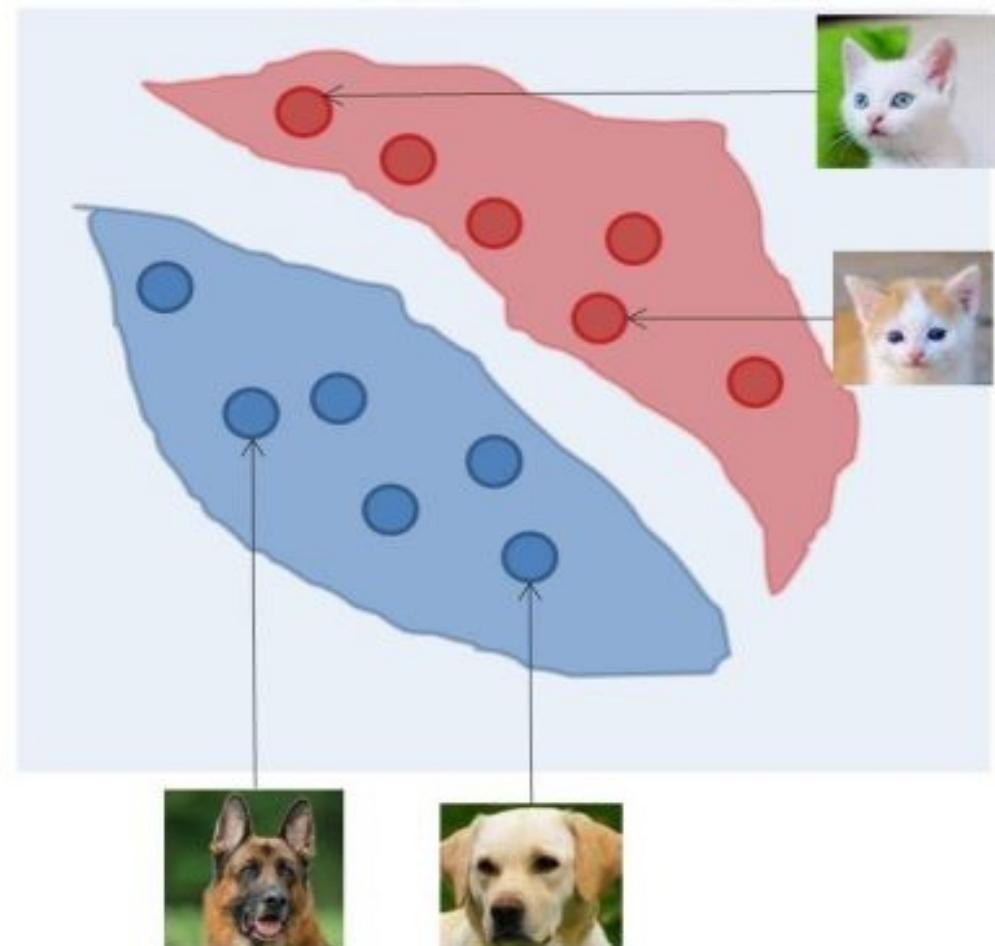
Discriminativo

Estimar directamente $P(y/x)$

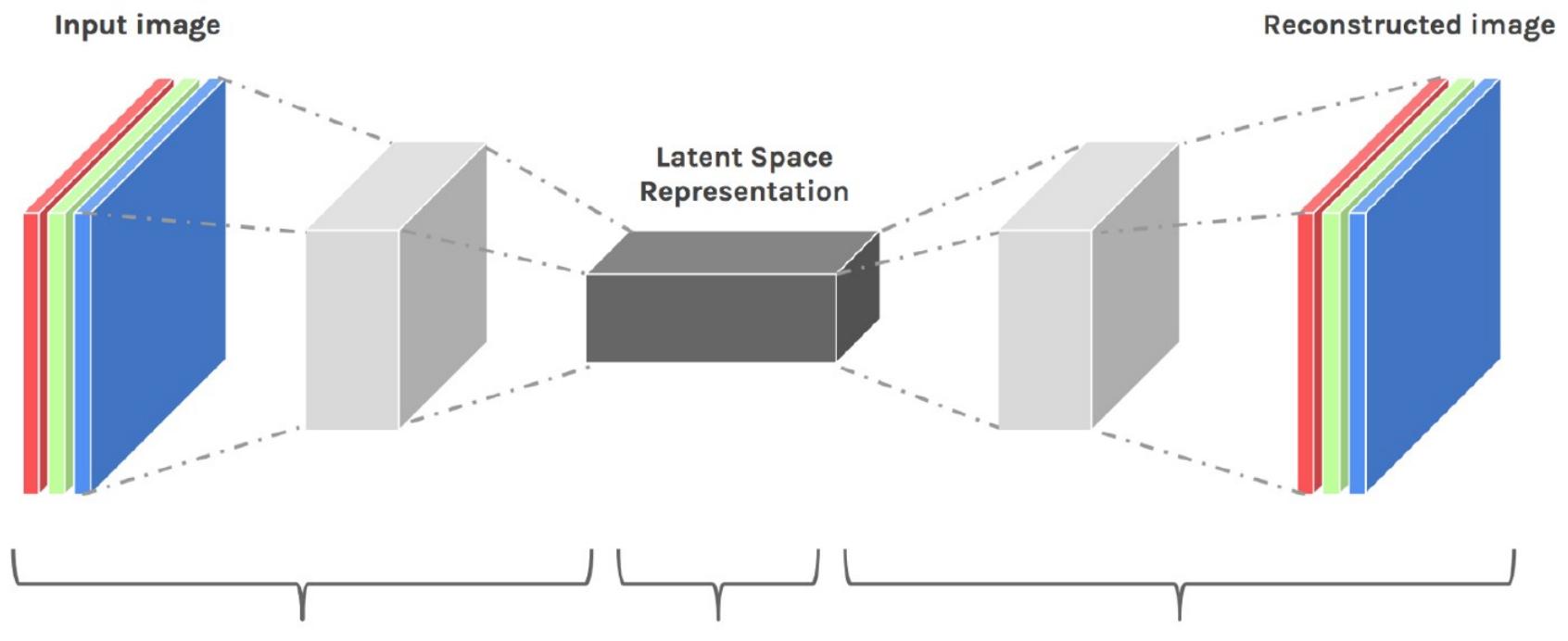


Generativo

Estima $P(x/y)$ y deduce $P(y/x)$



Autoencoder Variacional: Interpretación Neuronal

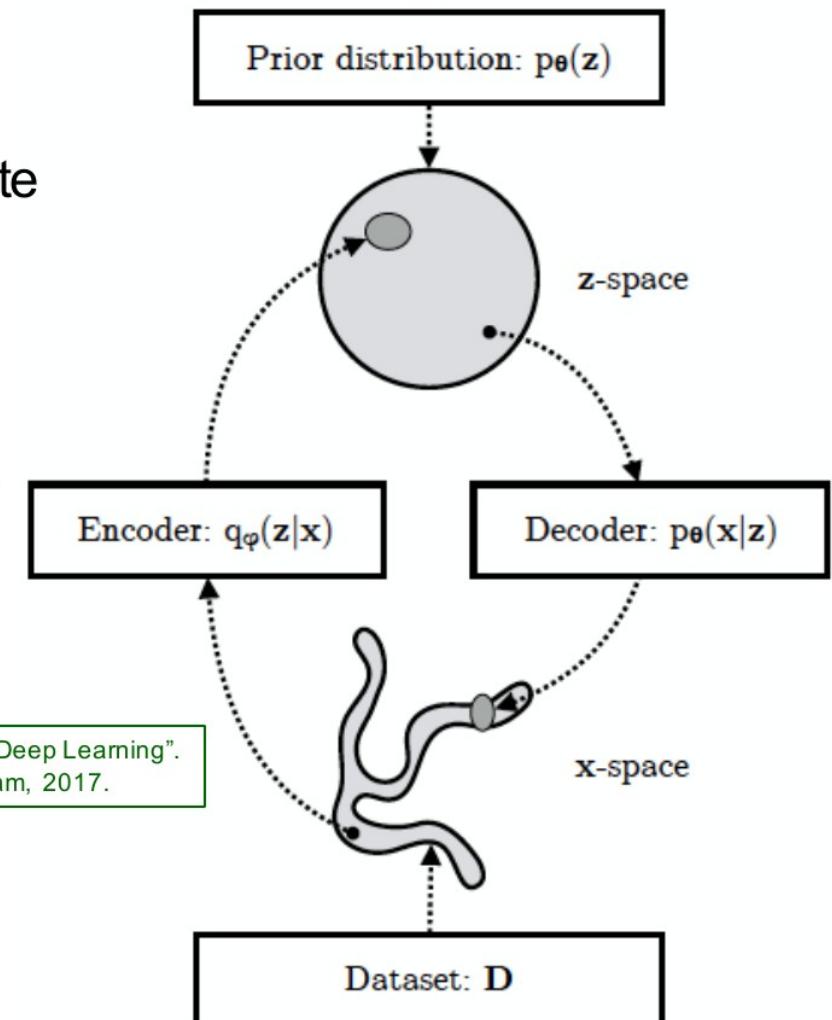


<https://datasciencevision.com/autoencoders/>

Autoencoder Variacional (VAE)

- Modelo probabilístico generativo
- Interpretación neuronal: 2 pasos:
 - Transformar datos a un espacio latente
 - Encoder (NN con pesos φ)
 - Devuelve parámetros
 - Transformar del espacio latente de nuevo al espacio observado de forma probabilística
 - Decoder (NN con pesos θ)
 - Devuelve parámetros
- Objetivo
 - Reducir al mínimo el error de reconstrucción
 - Para que la salida del VAE se parezca al máximo a su entrada

D. Kingma. "Variational Inference and Deep Learning".
PhD Thesis, Univ. Van Amsterdam, 2017.



0000000000000000
1111111111111111
2222222222222222
3333333333333333
4444444444444444
5555555555555555
6666666666666666
7777777777777777
8888888888888888
9999999999999999

The MNIST database (Modified National Institute of Standards and Technology database)

The MNIST database contains 60,000 training images and 10,000 testing images

28x28 pixel bounding box

Autoencoder Variacional

■ ¿Para qué?

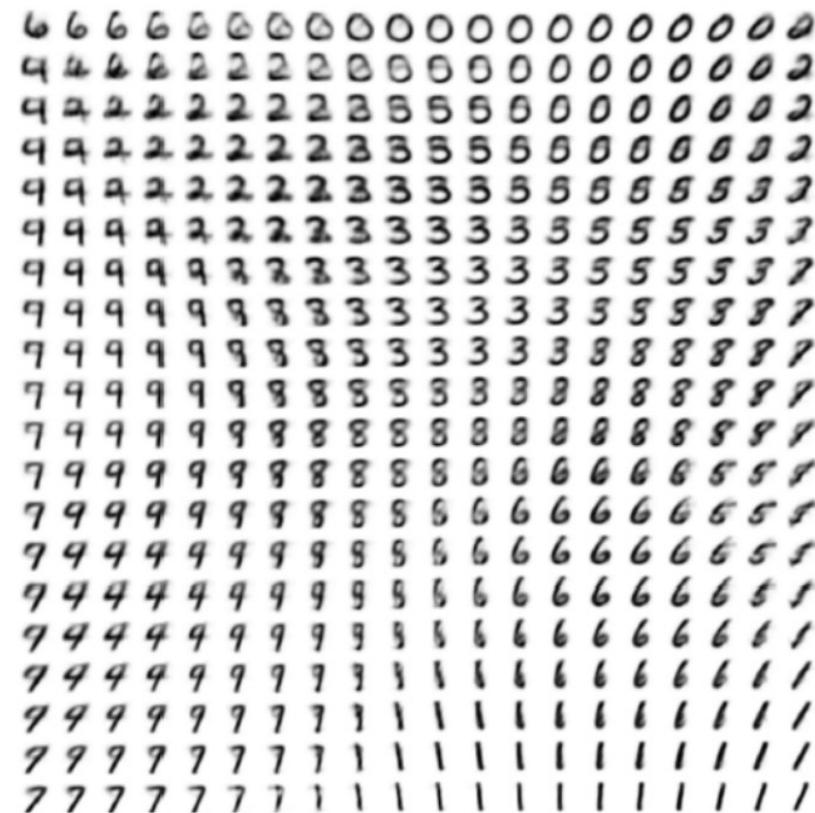
- ❑ Representación probabilística simplificada de los datos
- ❑ A partir de dicha representación, generar nuevos datos nunca vistos antes

D. Kingma, M. Welling. "Auto-Encoding Variational Bayes". Arxiv, 2014.



< aud

(a) Learned Frey Face manifold



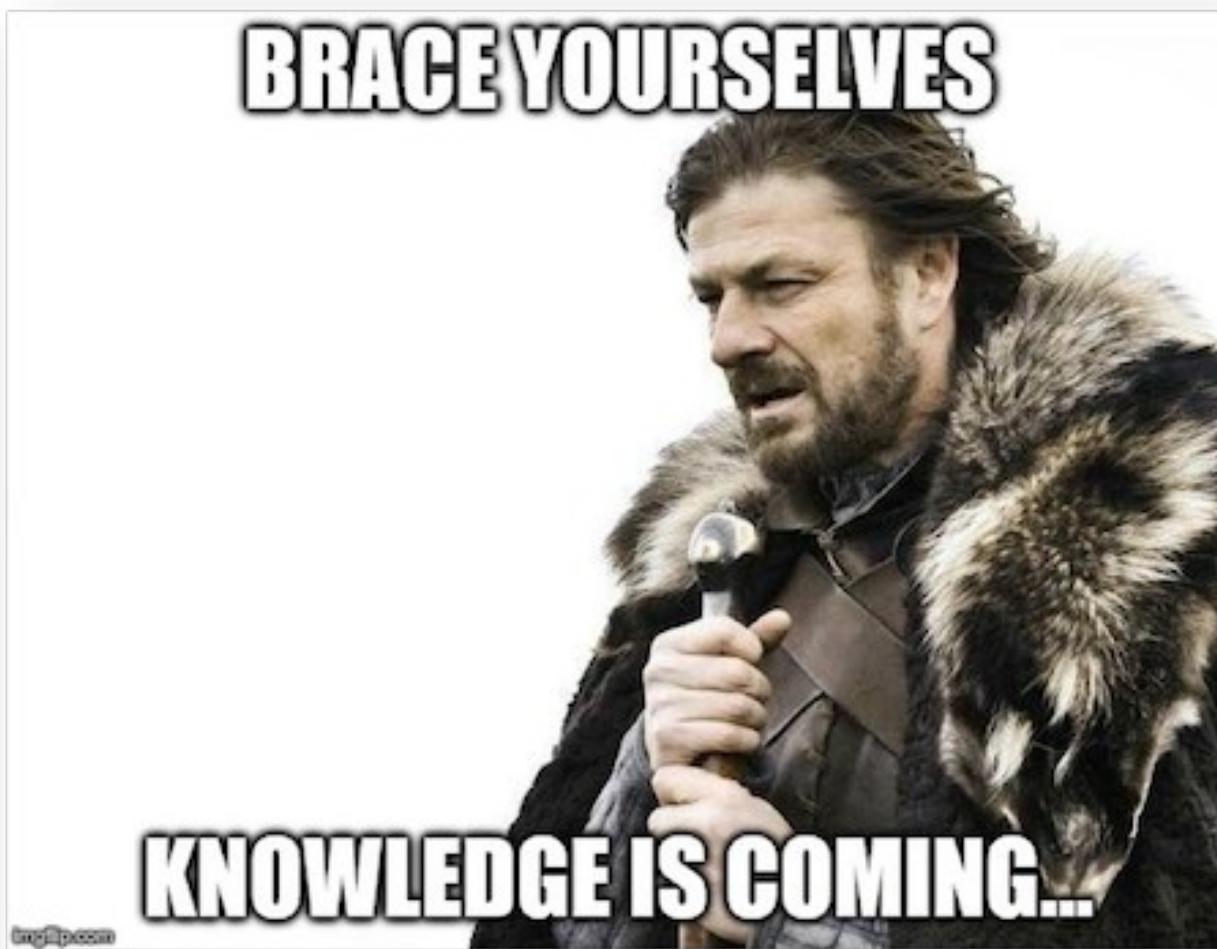
(b) Learned MNIST manifold

UAM

Autoencoder Variacional

- Variables observadas: x
 - Por ejemplo, vector de píxeles de dígitos MNIST
- Variables latentes: z
 - Un espacio de dimensión mucho menor
- Objetivo: modelo generativo (θ : pesos de una NN)
 - $p_{\theta}(x, z) = p_{\theta}(x|z)p_{\theta}(z)$
- Generación de datos
 - Primero muestreo un dato latente
 - $z^{(i)} \sim p_{\theta}(z)$
 - Luego obtengo el dato observado
 - $x^{(i)} \sim p_{\theta}(x|z^{(i)})$

Calibración?

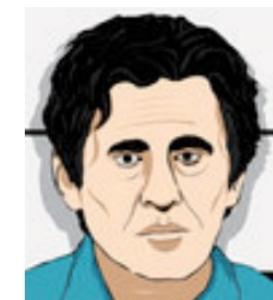


El problema

- Grabación incriminatoria (**dubitada**)
 - Pinchazo telefónico
 - Llamada anónima
 - Micrófono oculto
 - ...
- La policía arresta a un sospechoso
- Se realiza una toma de voz del sospechoso (**indubitada**)
 - En dependencias policiales
 - Pinchazos cuya autoría se reconoce
 - ...
- El contenido lingüístico no se conoce a priori en ambos casos
 - **Independiente de texto**



**Criminal
(Identidad C)**



**Sospechoso
(Identidad S)**

Evidencia

- La evidencia es la relación entre la toma dubitada y la toma indubitada
 - La evidencia nos da información sobre la relación de ambas fuentes
 - Ambas fuentes están relacionadas
 - Ambas fuentes no están relacionadas
- Valorar la evidencia es evaluar esa información



Planteamiento

- Hipótesis que se manejan:
 - Hipótesis del fiscal: H_p
 - Ejemplo: “ambas tomas pertenecen a la misma fuente”
(ventana en la escena del crimen)
 - Hipótesis del defensor: H_d
 - Ejemplo: “ambas tomas pertenecen a fuentes diferentes”
(ventanas diferentes)
- Pregunta sobre la que se basa la decisión del juez
 - ¿Cuál es la probabilidad de que, a la luz de la evidencia (E) y del resto de información acerca del caso, el sospechoso sea el autor del robo?

$$\mathcal{P}(H_p | E, I)$$

Solución: Teorema de Bayes



$$P(H_p | E, I) = \frac{P(E | H_p, I) P(H_p | I)}{P(E | I)}$$

$$P(H_d | E, I) = \frac{P(E | H_d, I) P(H_d | I)}{P(E | I)}$$

$$\frac{P(H_p | E, I)}{P(H_d | E, I)} = \frac{P(E | H_p, I) P(H_p | I)}{P(E | H_d, I) P(H_d | I)}$$

Separación de Roles

$$\frac{P(H_p | E, I)}{P(H_d | E, I)} = \frac{P(E | H_p, I)}{P(E | H_d, I)} \cdot \frac{P(H_p | I)}{P(H_d | I)}$$

$$\frac{P(H_p | E, I)}{P(H_d | E, I)}$$



$$\frac{P(E | H_p, I)}{P(E | H_d, I)}$$



$$\frac{P(H_p | I)}{P(H_d | I)}$$



¿Rol del científico forense?

- Calcular el *likelihood ratio (LR)*

$$LR = \frac{P(E|H_p, I)}{P(E|H_d, I)}$$



LR>1: apoyo la hipótesis del fiscal

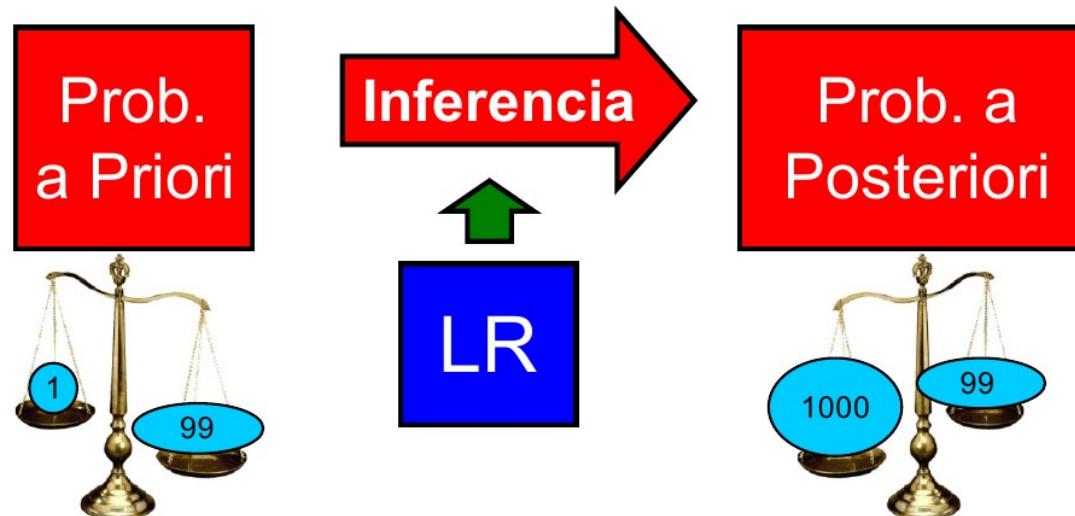
LR<1: Apoyo la hipótesis del defensor

LR=1: No apoyo a nadie

- Cuanto mayor (menor) el valor del LR, más apoyo a la hipótesis del fiscal (de la defensa)
- Clave: ¿cómo calcular el LR?

Inferencia en Ciencia Forense

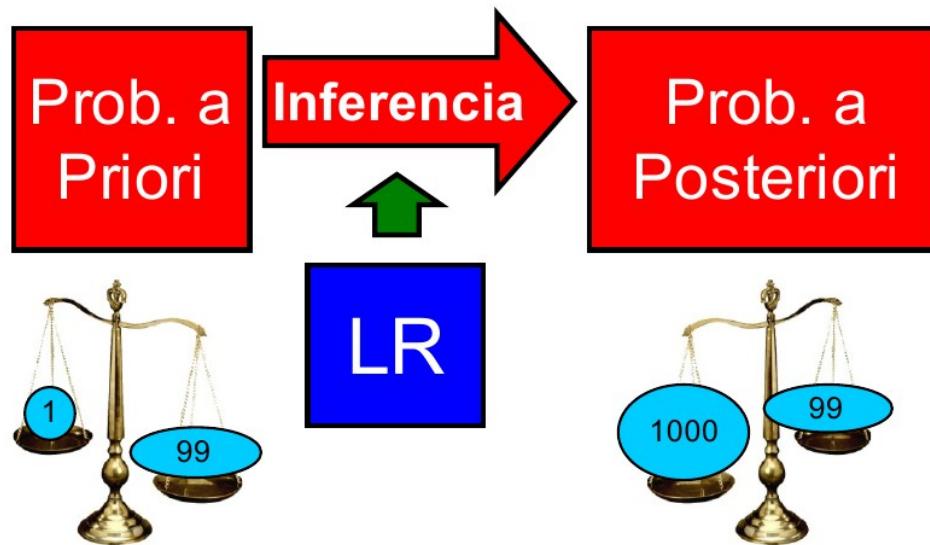
- Razón de Verosimilitud: valor probabilístico de la evidencia



LR

$$\frac{P(H_p | E, I)}{P(H_d | E, I)} = \frac{P(E | H_p, I) P(H_p | I)}{P(E | H_d, I) P(H_d | I)}$$

Decisión en un Caso: Elementos



■ Inferencia

- Probabilidad (apuesta) a priori, sin conocer la prueba
- Probabilidad (apuesta) a posteriori, una vez conocida la prueba
- LR: valor de la prueba

Calibración Extrínseca: Referencias

■ Robustez frente a falta de datos

D. Ramos-Castro, J. Gonzalez-Rodriguez, A. Montero-Asenjo and J. Ortega-Garcia, "Suspect-adapted MAP estimation of within-source distributions in generative likelihood ratio estimation", IEEE Odyssey 2006.

■ Calibración de scores (extrínseca) no supervisada

Niko Brummer and Daniel Garcia-Romero, "Generative Modelling for Unsupervised Score Calibration", ICASSP 2014.

■ Calibración extrínseca bayesiana

Niko Brummer and Albert Swart, 'Bayesian calibration for forensic evidence reporting', Interspeech 2014.

D. Ramos et al., ' Bayesian strategies for Likelihood Ratio computation in forensic voice comparison with automatic systems.', Subsidia 2017.

■ Análisis de la distribución de scores calibrados

David van Leeuwen, Niko Brummer, "The distribution of calibrated likelihood-ratios in speaker recognition", Interspeech 2013.

■ NNs para calibración de LRs

W. Campbell et al., "Estimating and Evaluating Confidence for Forensic Speaker Recognition", ICASSP 2005.

DRL



Fuente: Code Bullet (YouTube) videos de ML

SUSCRIBIRSE 1,6 M

DRL

Aprendizaje por refuerzo



Fuente: Siraj Raval (Suscríbete al canal de Youtube)

SUSCRIBIRSE 684 MIL

Introducción

Curso



UNIVERSIDAD
DE GRANADA

Aprendizaje profundo por refuerzo

(*deep reinforcement learning*)

Profesor: Juan Gómez Romero (PhD)

jgomez@ugr.es <http://decsai.ugr.es/~jgomez>

Material <https://github.com/jgromero/eci2019-DRL>

Organización del curso 3 horas / día (30' Q&A, 2h teoría + prácticas, 30' lab) x 5 días

1. Introducción al aprendizaje profundo (días 1, 2)
2. Aprendizaje por refuerzo (día 3)
3. Aprendizaje profundo por refuerzo (días 4, 5)

Introducción

Aprendizaje profundo por refuerzo

Prerrequisitos

Se recomienda contar con conocimientos sobre Álgebra, Inteligencia Artificial y Redes Neuronales.

El lenguaje de programación que se utilizará en el curso es Python.



UNIVERSIDAD
DE GRANADA

Conceptos general de AI, ML y ANN

- Del curso "AI for Everyone", de Coursera, Week 1:

Vídeo: Machine Learning

Vídeo: The Terminology of AI

Vídeo: Non-technical explanation of Deep Learning (Part 1, Part 2)

- Del libro de Chollet "Deep Learning with Python":

Capítulo 1: What is Deep Learning

Formalización de modelos de ANN

- Del libro de Russell & Norvig "Artificial Intelligence: A Modern Approach":

Sección 18.7: Artificial Neural Networks (20.5 en la versión en español)

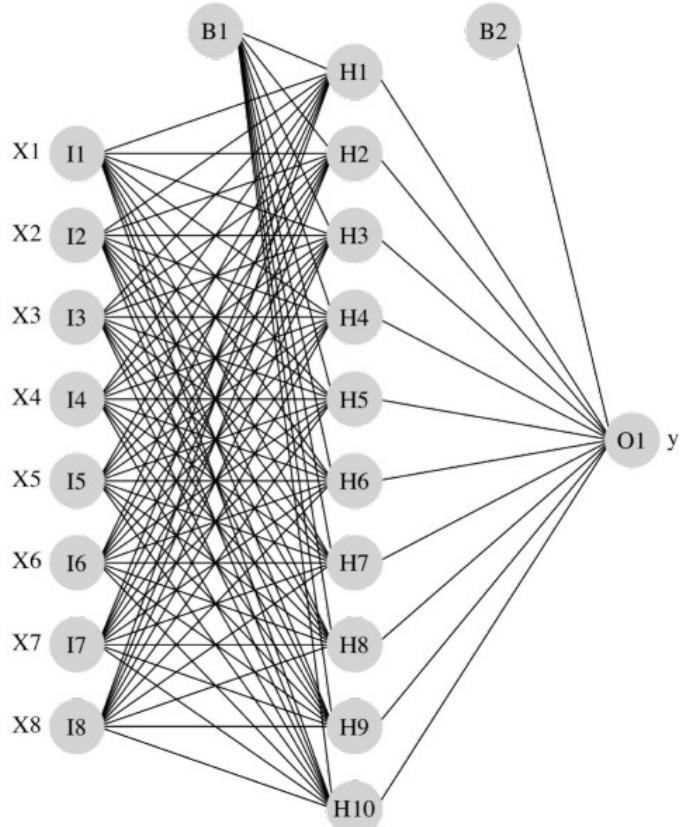
Primeros pasos en Deep Learning & PyTorch

- Del libro de Stevens & Antiga "Deep Learning with PyTorch":

Capítulo 1: PyTorch from 1 Mile Away

Redes neuronales

Modelo



<https://youtu.be/MRlv2IwFTPg>
Partes 1, 2



UNIVERSIDAD
DE GRANADA

Entrada (n)

$$x = (x_1, \dots, x_n)$$

Salidas capa oculta (m)

$$H_j = h\left(\sum_{i=1}^n x_i w_{ij} + B_1\right)$$



producto escalar

$$x \cdot w = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} [w_1 \quad \dots \quad w_n]$$

Salidas finales (k)

$$O_k = h'\left(\sum_{j=1}^m H_j w_{jk} + B_2\right)$$

5

Introducción

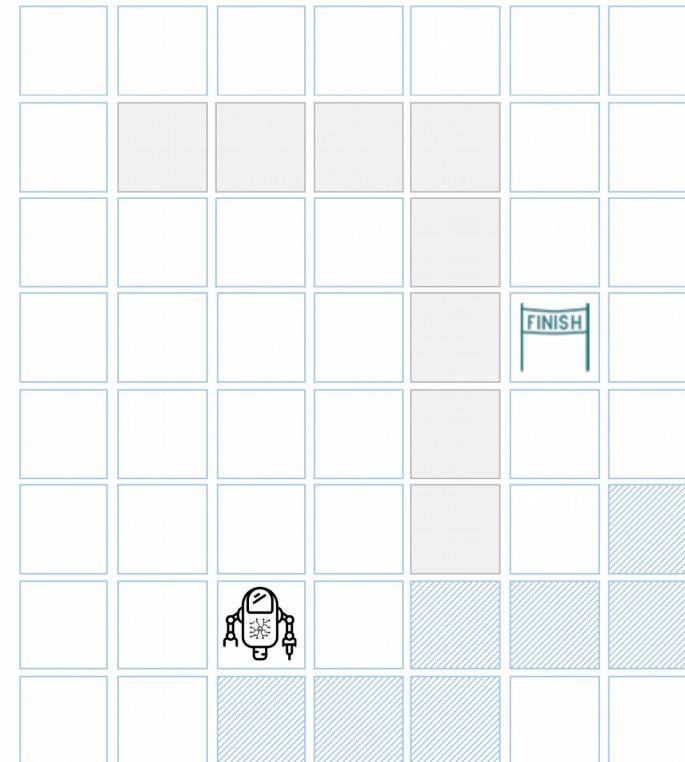
Aprendizaje por refuerzo



UNIVERSIDAD
DE GRANADA

Objetivo

Encontrar la meta en el laberinto en el menor tiempo posible



Introducción

Aprendizaje por refuerzo

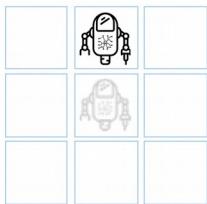


UNIVERSIDAD
DE GRANADA

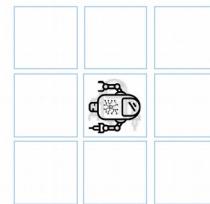
Objetivo

Encontrar la meta en el laberinto en el menor tiempo posible

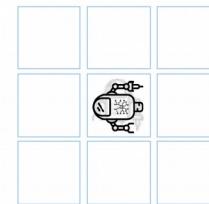
Acciones



avanzar



giro derecha



giro izquierda

Tiempo



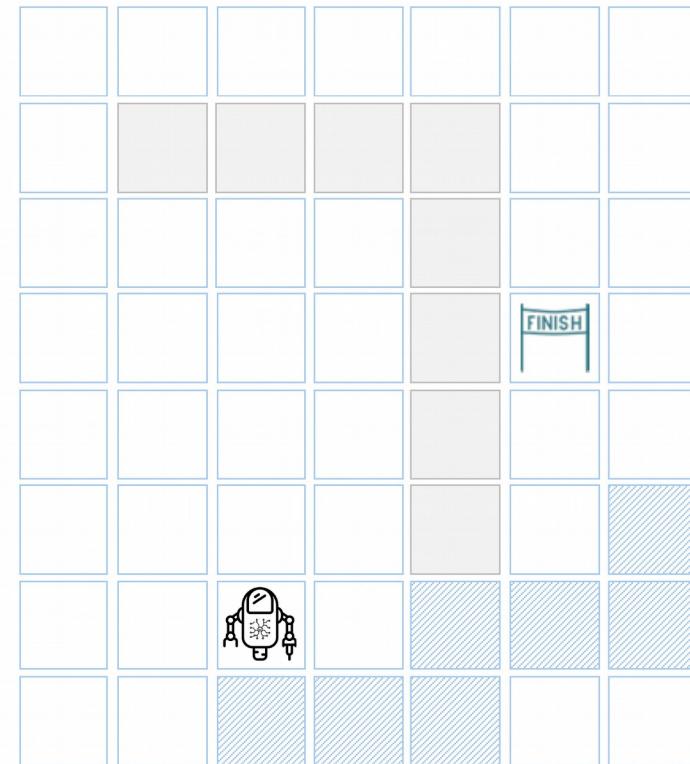
+1



+5



no autorizado



Introducción

Aprendizaje por refuerzo

Objetivo

Encontrar la meta en el laberinto en el menor tiempo posible

Camino óptimo: 20s

Algoritmos de búsqueda óptimos

+ Búsqueda con coste

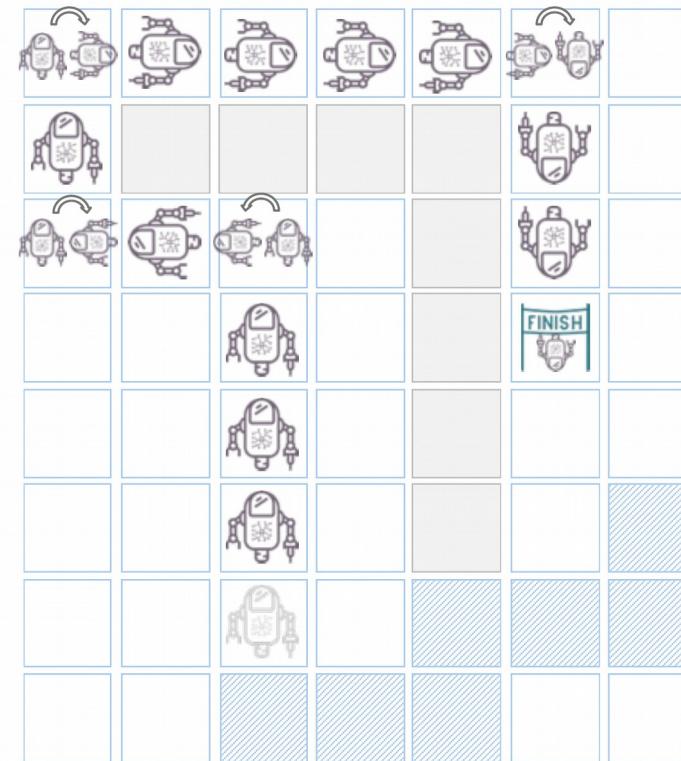
Explora todos los posibles caminos, asignando un coste según el tiempo que se tarda en realizar una acción

+ Búsqueda con función de potencial

+ Búsqueda con función heurística (e.g A*)

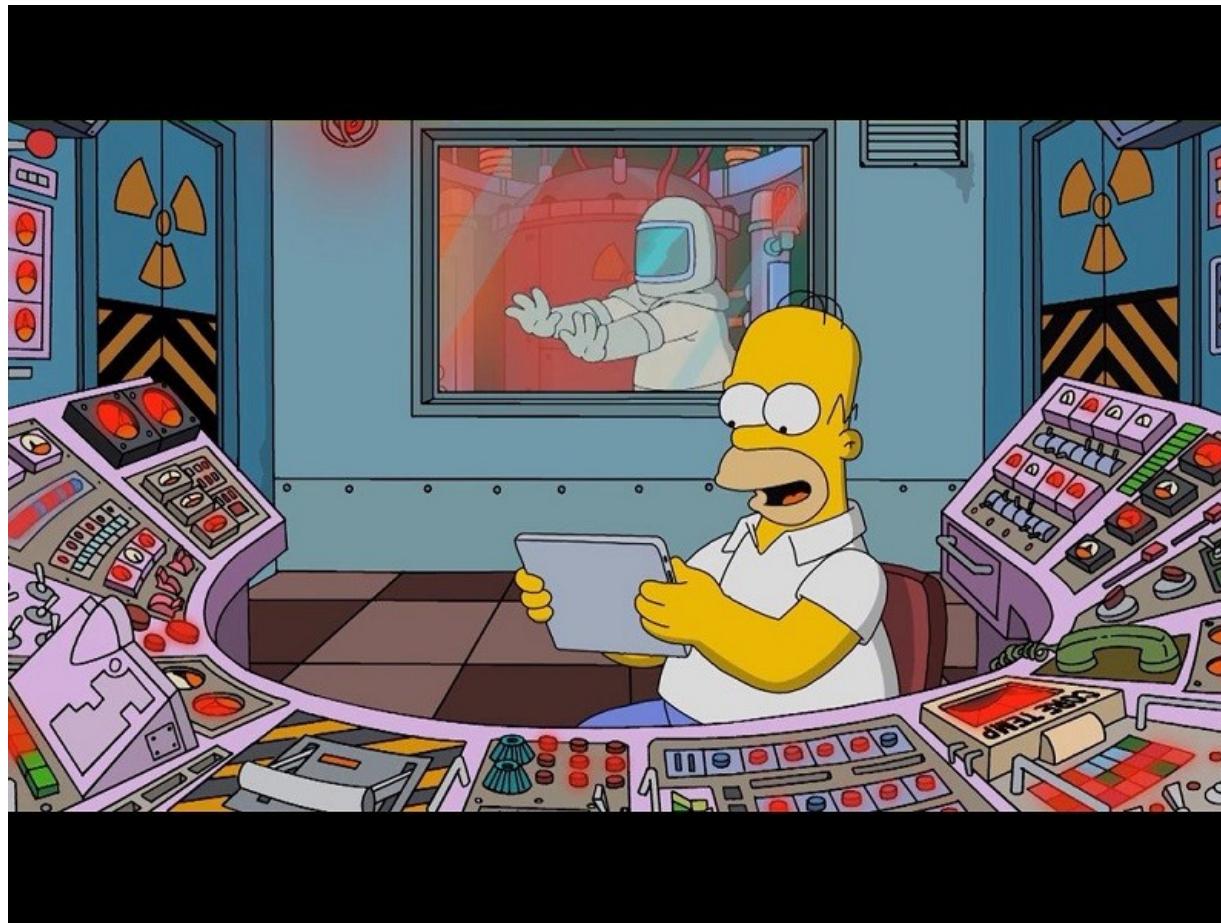
Necesitan conocer el mapa y la posición objetivo

No son viables en problemas de tamaño moderado



DRL

El agente realiza una acción



Si la realiza correctamente



Si no...



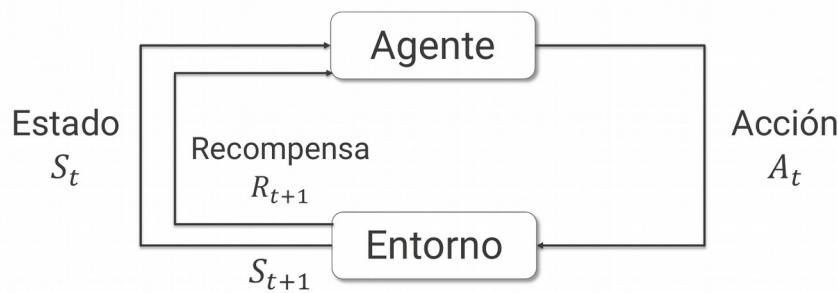
Introducción

Aprendizaje por refuerzo



UNIVERSIDAD
DE GRANADA

Aprendizaje por refuerzo

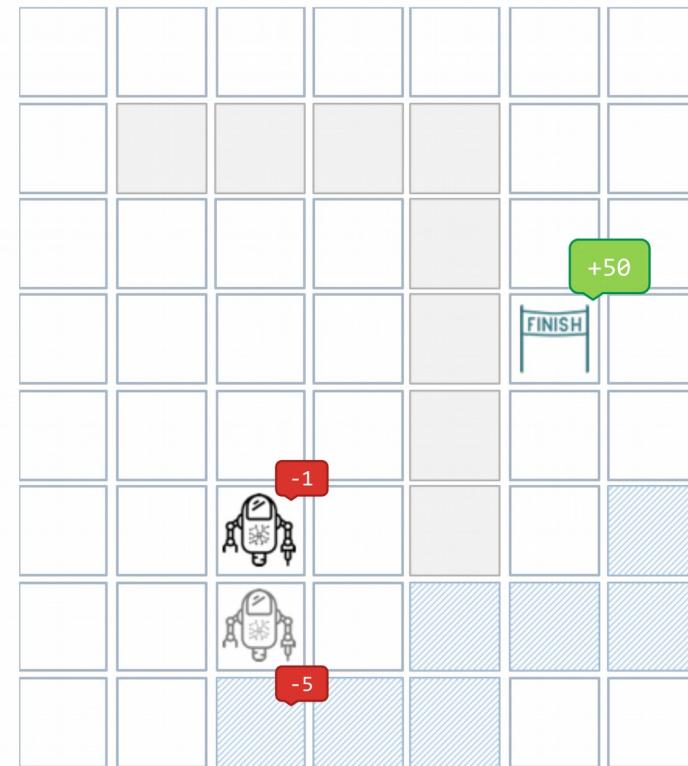


R.S. Sutton, A.G. Barto (2018) **Reinforcement Learning**. MIT Press.

$$S_t = \{(6, 2), \uparrow\} \longrightarrow S_{t+1} = \{(5, 2), \uparrow\}$$

A_t = AVANZAR

$$R_{t+1} = -1$$



Introducción

Aprendizaje por refuerzo

Política de actuación: π

Si normal: mover arriba

Si rayas: mover derecha

Secuencia o episodio:

$(2, 1) \uparrow -5 (1, 1)$

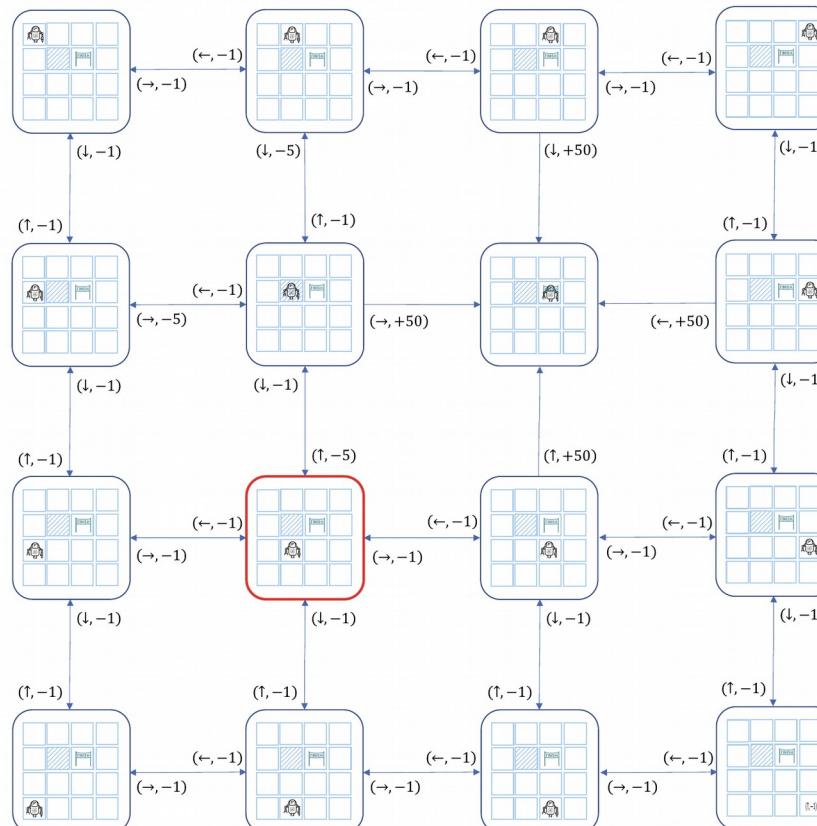
$(1, 1) \rightarrow +50 (1, 2)$

Recompensa acumulada:

$$-5 + 50 = -45$$

Política óptima π_* :

Estado	Acción
$(2, 1)$	\rightarrow
$(2, 2)$	\uparrow
$(2, 3)$	\leftarrow
...	



UNIVERSIDAD
DE GRANADA

Aprendizaje



Introducción

Aprendizaje profundo por refuerzo



UNIVERSIDAD
DE GRANADA

Aprendizaje profundo por refuerzo (deep reinforcement learning)

Utilizar redes neuronales profundas para optimizar la recompensa acumulada



La red neuronal:

- devuelve la próxima acción que se debe realizar...
- ... en función de una estimación de cómo de buena es un acción en el estado actual

Entrenar la red para encontrar π_* :

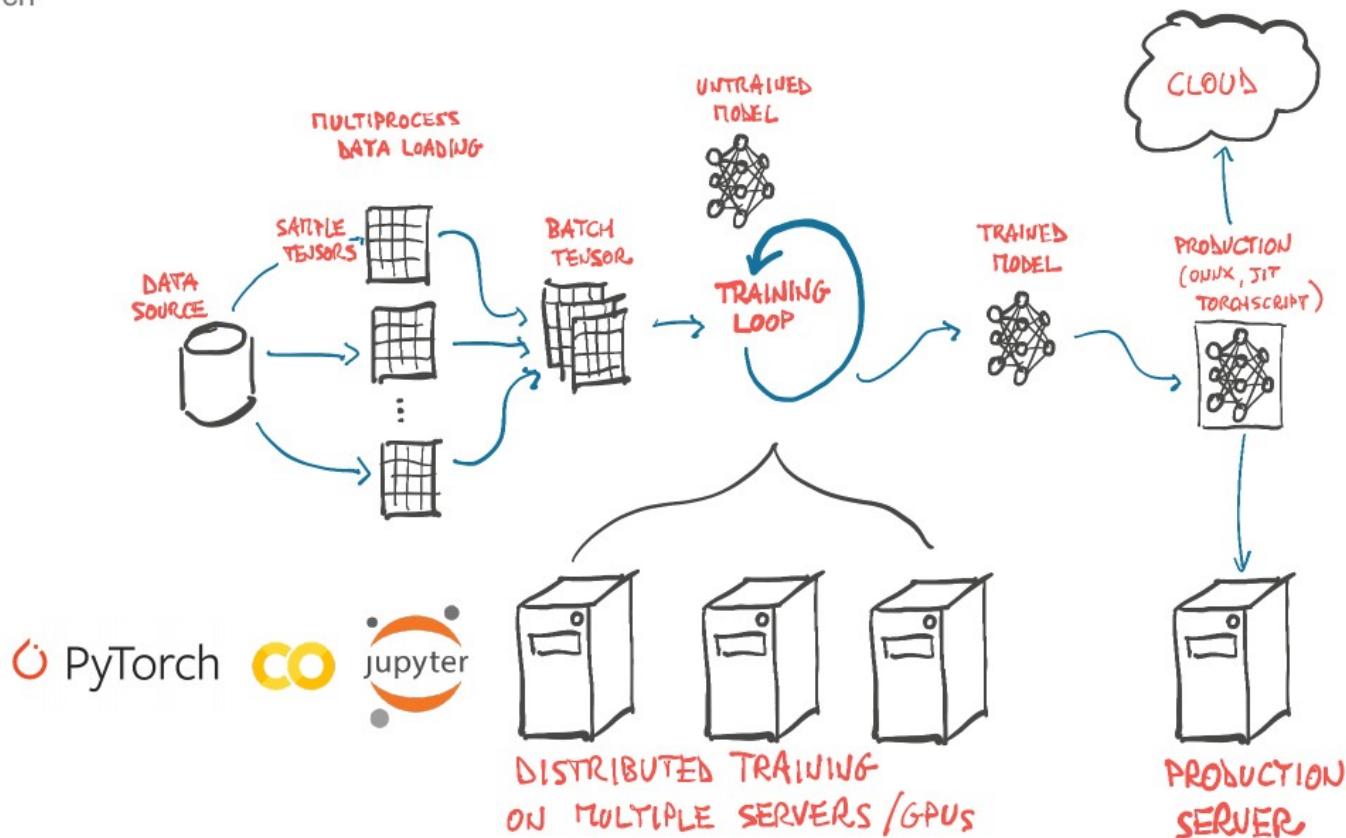
- A partir de ejemplos de secuencias de acciones y recompensas asociadas

Implementación

PyTorch



UNIVERSIDAD
DE GRANADA



E. Stevens, L. Antiga (2019) Deep Learning with PyTorch. Manning.

24

Implementación

Google Colaboratory



UNIVERSIDAD
DE GRANADA



<https://colab.research.google.com>

Entorno gratuito de *Jupyter Notebook* que no requiere configuración y que se ejecuta completamente en la nube.

Permite escribir, ejecutar y guardar código Python utilizando GPUs, de forma gratuita desde el navegador.

El código se escribe en un cuaderno. Un cuaderno puede contener celdas de código (en Python) y celdas de texto (en Markdown).

El cuaderno puede ejecutarse completamente o por celdas.

<https://colab.research.google.com/notebooks/welcome.ipynb>

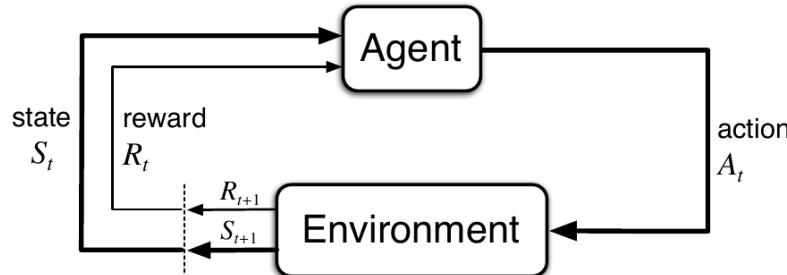
Google Colaboratory ya tiene preinstalado PyTorch.

```
!pip3 install torch torchvision
```

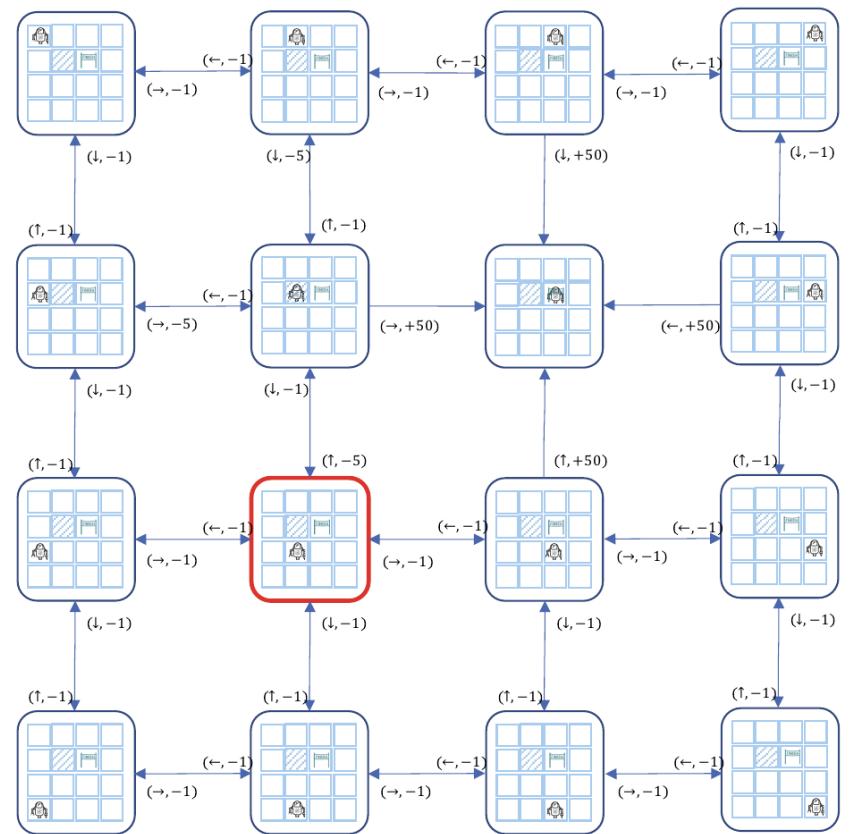
Aprendizaje por refuerzo

Problema de búsqueda en el grafo de estados y transiciones.

Optimizar la función de recompensa acumulada (*max*)



R.S. Sutton, A.G. Barto (2018) *Reinforcement Learning*. MIT Press.

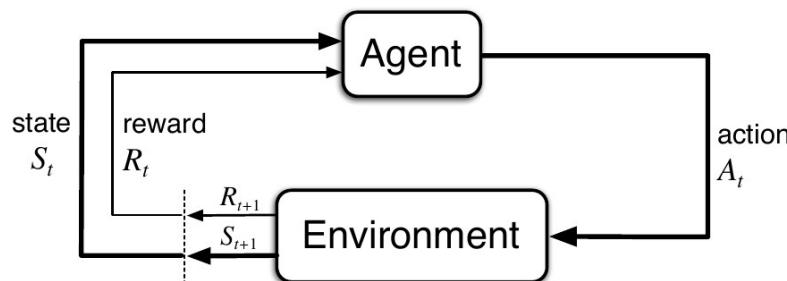


MDP

Formulación



UNIVERSIDAD
DE GRANADA



R.S. Sutton, A.G. Barto (2018) *Reinforcement Learning*. MIT Press.

Markov Decision Process - MDP

tiempo: $0, 1, 2, \dots$
estado: $S_t \in \mathcal{S}$
acción: $A_t \in \mathcal{A}(S_t)$
recompensa: $R_{t+1} \in \mathcal{R} \subset \mathbb{R}$
estado siguiente: $S_{t+1} \in \mathcal{S}$

secuencia:
 $S_0, A_0, R_1, S_1, A_1, R_2, S_2, A_2, R_3, \dots$

dinámica del MDP:

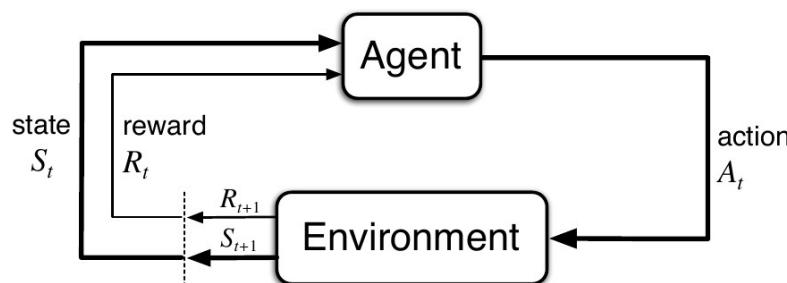
$$p(s', r | s, a) \doteq Pr\{S_t = s', R_t = r | S_{t-1} = s, A_{t-1} = a\}$$

MDP

Formulación



UNIVERSIDAD
DE GRANADA



R.S. Sutton, A.G. Barto (2018) *Reinforcement Learning*. MIT Press.

Markov Decision Process - MDP

Recompensa acumulada:

$$G_t \doteq R_{t+1} + R_{t+2} + R_{t+3} + \dots + R_T$$

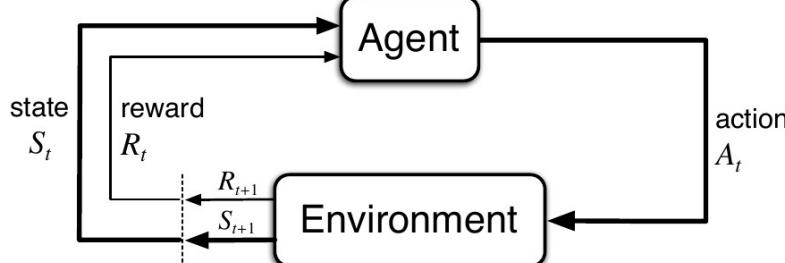
Recompensa con descuento:

$$G_t \doteq R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \dots = \sum_{k=0}^T \gamma^k R_{t+k+1}$$

Política de actuación: π

Acción según π : $\pi(a|s)$

Política de actuación óptima: π_*



R.S. Sutton, A.G. Barto (2018) *Reinforcement Learning*. MIT Press.

Markov Decision Process - MDP

Función de estado-valor [*state value*] (para todo s):

$$v_\pi(s) \doteq \mathbb{E}_\pi[G_t \mid S_t = s] = \mathbb{E}_\pi \left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \mid S_t = s \right]$$

Función de acción-valor [*action value*] (para todo (s, a)):

$$q_\pi(s, a) \doteq \mathbb{E}_\pi[G_t \mid S_t = s, A_t = a] = \mathbb{E}_\pi \left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \mid S_t = s, A_t = a \right]$$

Comparación de políticas de actuación y política óptima

$$\pi \geq \pi' \Leftrightarrow v_\pi(s) \geq v_{\pi'}(s)$$

$$\pi_* \geq \pi' \quad \forall \pi' \quad \text{(garantizada)}$$

MDP

Formulación

$$v_{\pi}(s) \doteq \mathbb{E}_{\pi} \left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \mid S_t = s \right]$$

Política π_1 ($\gamma = 1$) :

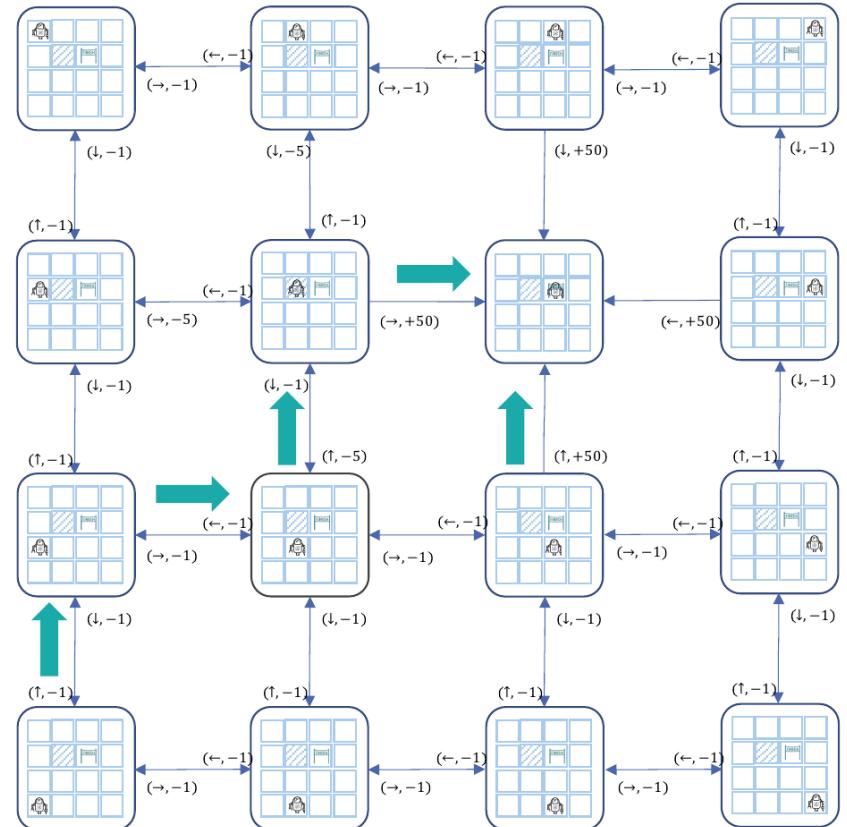
Estado	Acción
(3, 0)	\uparrow
(2, 0)	\rightarrow
(2, 1)	\uparrow
(1, 1)	\rightarrow
(2, 2)	\uparrow
...	

$$v_{\pi_1}(\langle 2, 1 \rangle) = -5 + 50 = 45$$

$$v_{\pi_1}(\langle 2, 2 \rangle) = 50$$

$$v_{\pi_1}(\langle 3, 0 \rangle) = -1 - 1 - 5 + 50 = 43$$

$$v_{\pi_1}(\langle 2, 0 \rangle) = -1 - 5 + 50 = 42$$



$$v_{\pi}(s) \doteq \mathbb{E}_{\pi} \left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \mid S_t = s \right]$$

Política π_2 ($\gamma = 1$) :

Estado	Acción
(3, 0)	↑
(2, 0)	→
(2, 1)	→
(1, 1)	→
(2, 2)	↑
...	

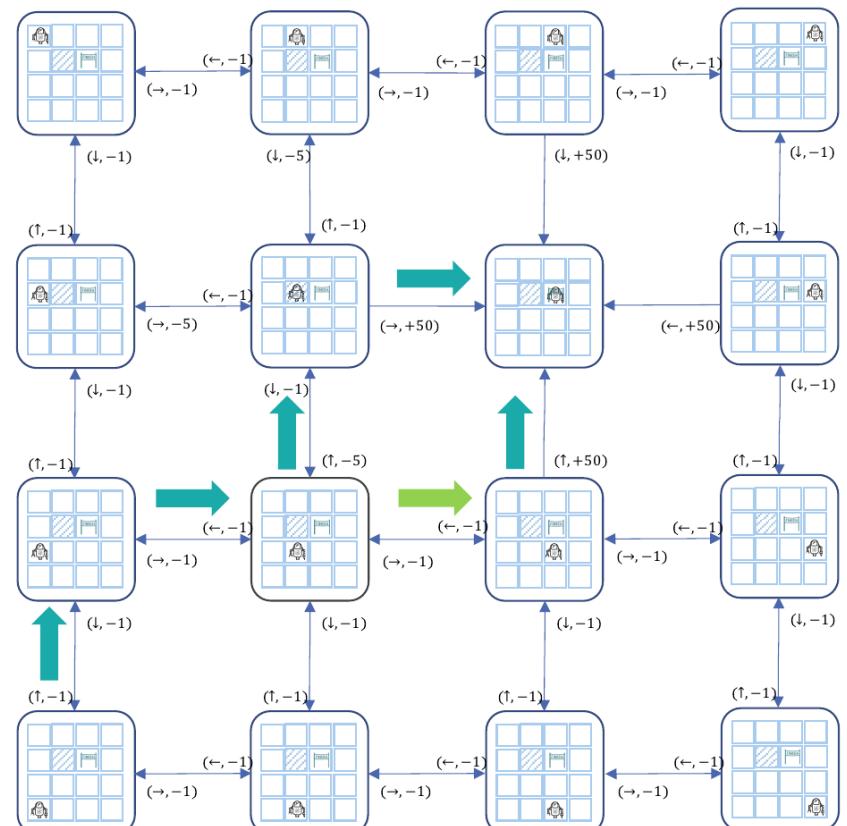


$$v_{\pi_2}(\langle 2, 1 \rangle) = -1 + 50 = 49 \geq 45$$

$$v_{\pi_2}(\langle 2, 2 \rangle) = 50 \geq 50$$

$$v_{\pi_2}(\langle 3, 0 \rangle) = -1 - 1 - 1 + 50 = 47 \geq 43$$

$$v_{\pi_2}(\langle 2, 0 \rangle) = -1 - 1 + 50 = 48 \geq 42$$



MDP

Formulación

$$q_{\pi}(s, a) \doteq \mathbb{E}_{\pi} \left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \mid S_t = s, A_t = a \right]$$

Política π_2 ($\gamma = 1$) :

Estado	Acción
(3, 0)	\uparrow
(2, 0)	\rightarrow
(2, 1)	\rightarrow
(1, 1)	\rightarrow
(2, 2)	\uparrow
(3, 1)	\leftarrow
...	

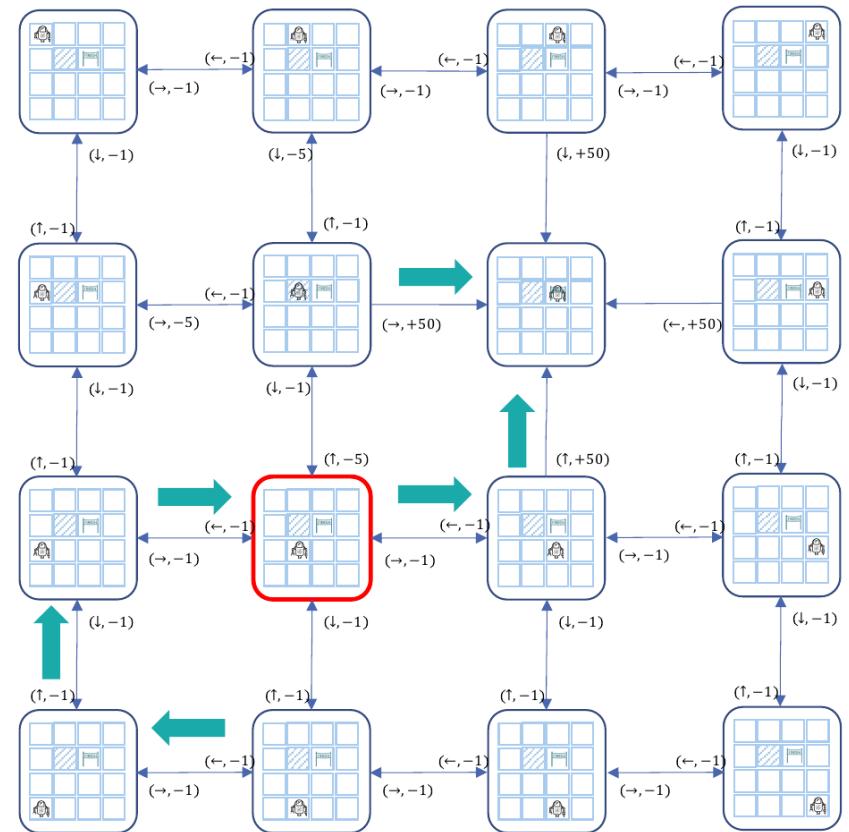
$$v_{\pi}(s) = q_{\pi}(s, \pi(a|s))$$

$$q_{\pi_2}(\langle 2, 1 \rangle, \uparrow) = -5 + 50 = 45$$

$$q_{\pi_2}(\langle 2, 1 \rangle, \rightarrow) = -1 + 50 = 49$$

$$q_{\pi_2}(\langle 2, 1 \rangle, \leftarrow) = -1 - 1 - 1 + 50 = 47$$

$$q_{\pi_2}(\langle 2, 1 \rangle, \downarrow) = -1 - 1 - 1 - 1 - 1 + 50 = 45$$



Ecuación de Bellman (recursividad de v_π)

$$\begin{aligned}
 v_\pi(s) &\doteq \mathbb{E}_\pi[G_t \mid S_t = s] \\
 &= \mathbb{E}_\pi[R_{t+1} + \gamma G_{t+1} \mid S_t = s] \\
 &= \mathbb{E}_\pi[R_{t+1} + \gamma v_\pi(S_{t+1}) \mid S_t = s]
 \end{aligned}$$

_____ *recompensa*
 _____ *valor del*
al realizar acción *siguiente estado*

Ecuación de Bellman (optimalidad de v_*)

$$\underline{v_{\pi_*}(s)} = \max_{a \in \mathcal{A}(s)} q_{\pi_*}(s, a)$$

_____ *valoración*
de un estado
(política óptima) *valoración de la*
mejor acción desde ese estado
(política óptima)

$$Q(s, a) = r + \gamma \max_{a'} Q(s', a')$$

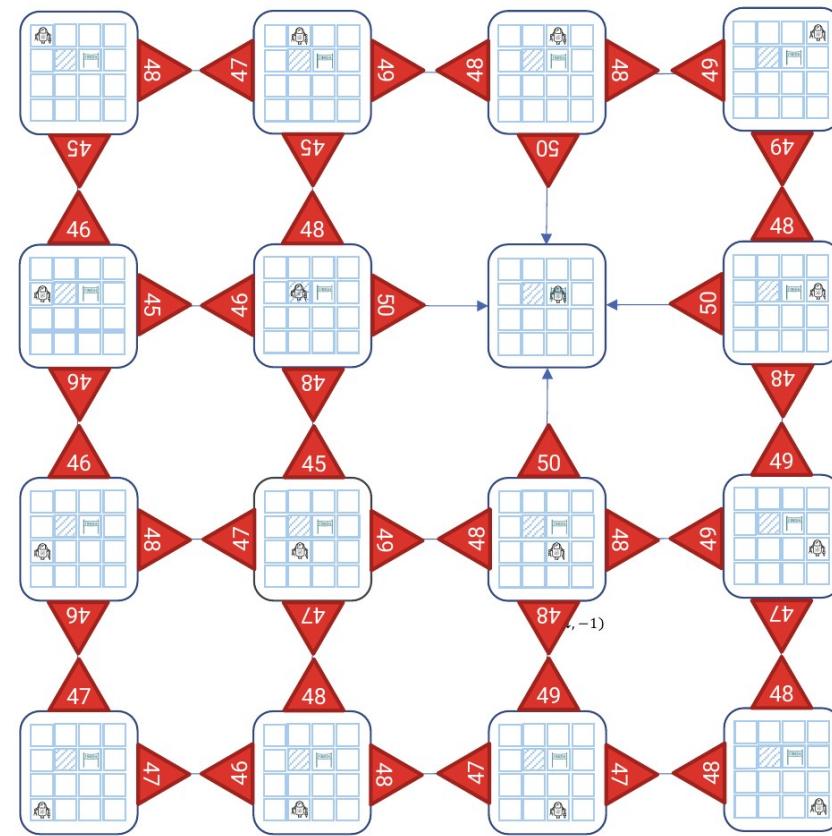
MDP

Formulación

$$q_{\pi_*}(s, a)$$



UNIVERSIDAD
DE GRANADA



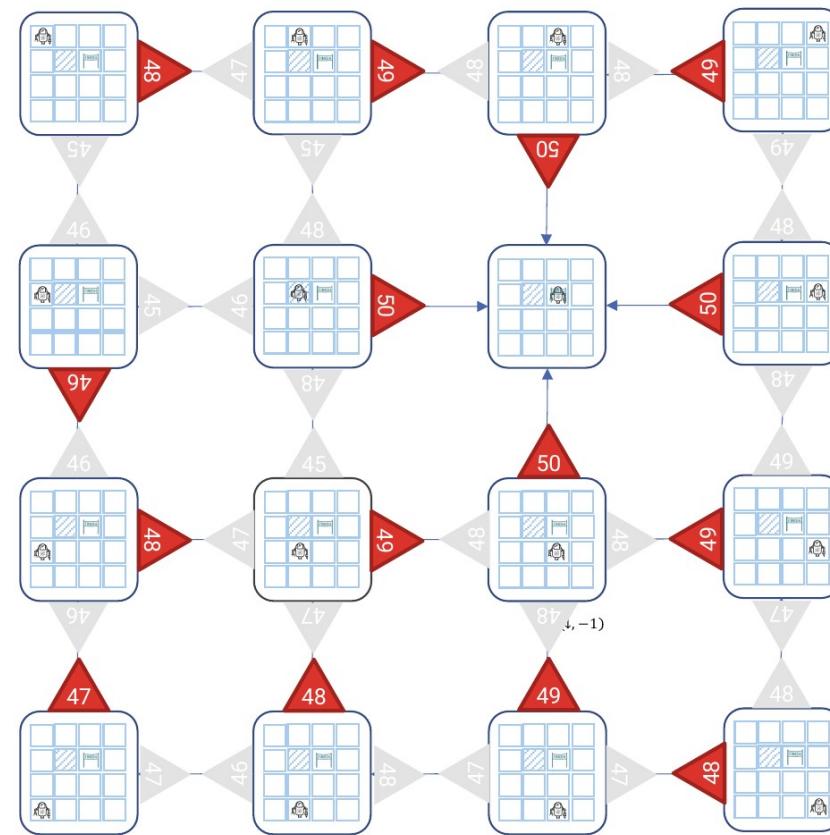
MDP

Formulación

$$q_{\pi_*}(s, a)$$

Resaltar valor máximo de cada estado:

$$v_*(s) = \max_{a \in \mathcal{A}(s)} q_{\pi_*}(s, a)$$



UNIVERSIDAD
DE GRANADA

MDP

Formulación

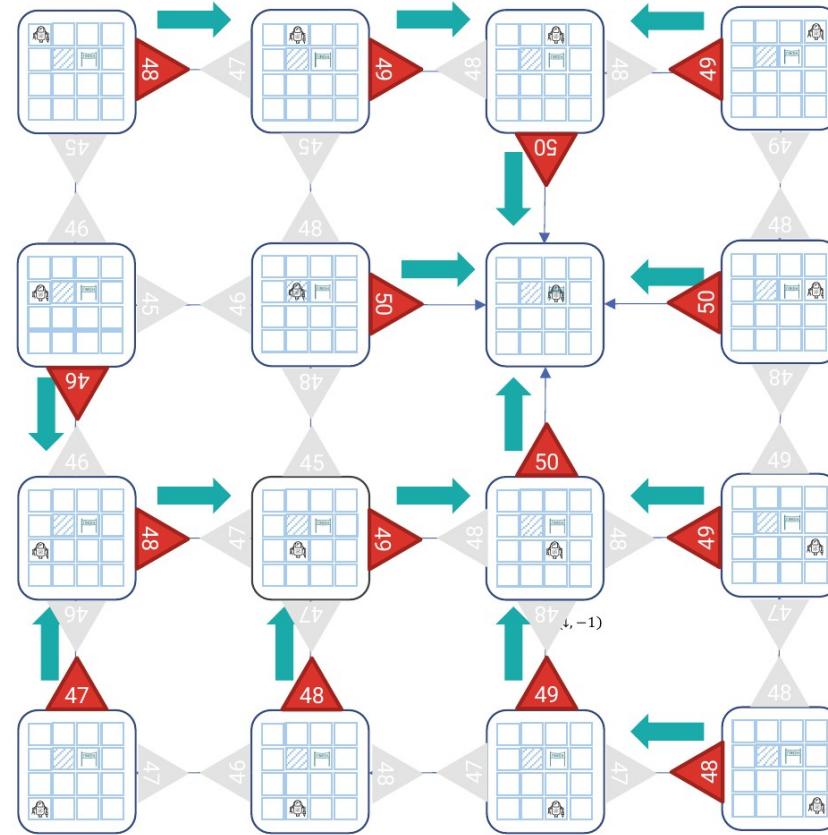
$$q_{\pi_*}(s, a)$$

Resaltar valor máximo de cada estado:

$$v_*(s) = \max_{a \in \mathcal{A}(s)} q_{\pi_*}(s, a)$$

Reconstruir π_*

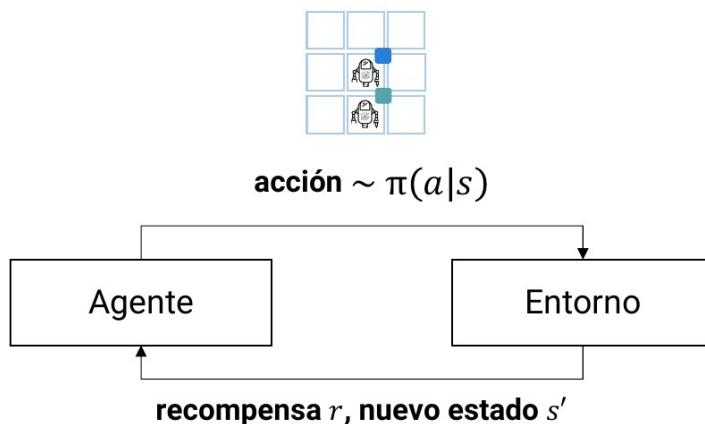
Estado	Acción
(3, 0)	→
(2, 0)	↑
...	



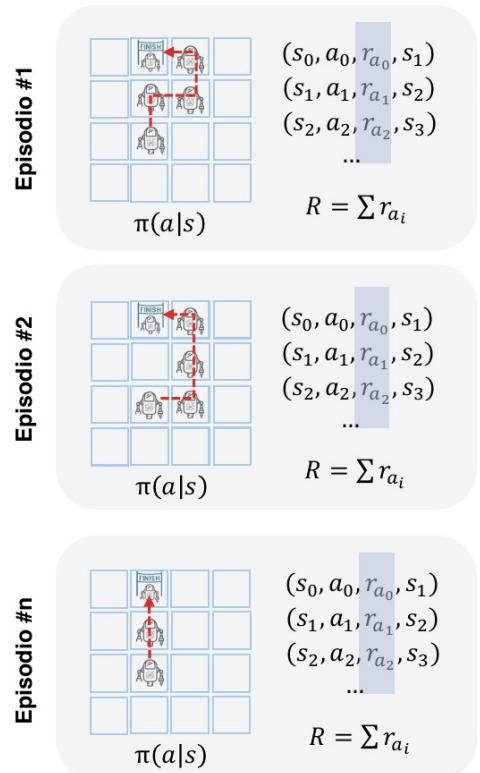
UNIVERSIDAD
DE GRANADA

MDP

Aprendizaje por refuerzo



Episodios de entrenamiento



Estado	Acción	q
(3, 0)	↑	10
(2, 0)	→	12
...		...



training

$$\pi' = \arg \max_{\pi} \sum_{t=0}^{\infty} \gamma^t * r_{at}(s_t, s_{t+1})$$

con $a_t \sim \pi(a|s)$

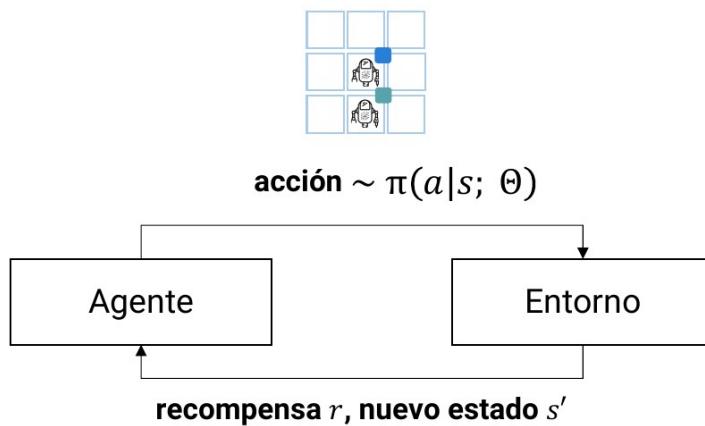
$\gamma \in [0, 1]$: tasa de descuento

MDP

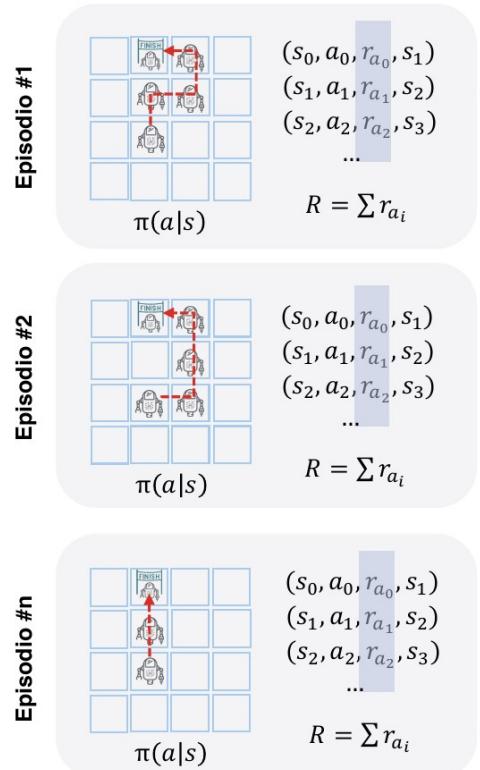
Aprendizaje por refuerzo



UNIVERSIDAD
DE GRANADA



Episodios de entrenamiento



training

$$\Theta' = \arg \max_{\Theta} \sum_{t=0}^T \gamma^t * r_{a_t}(s_t, s_{t+1})$$

con $a_t \sim \pi(s; \Theta)$

$\gamma \in [0, 1]$: tasa de descuento

Q-Learning

Formulación



UNIVERSIDAD
DE GRANADA

Q-Learning (SARSA-max)

params: Q inicial, n_episodios, α

for i = {0, ..., n_episodios}

while *episodio no terminado*

elegir acción A_t con política ϵ -greedy(Q)

aplicar A_t y obtener R_{t+1}, S_{t+1}

Actualizar Q

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha \left(R_{t+1} + \gamma \max_{a \in \mathcal{A}} Q(S_{t+1}, a) - Q(S_t, A_t) \right)$$

Deep Q-Learning

Deep Q-Learning

Q-Learning



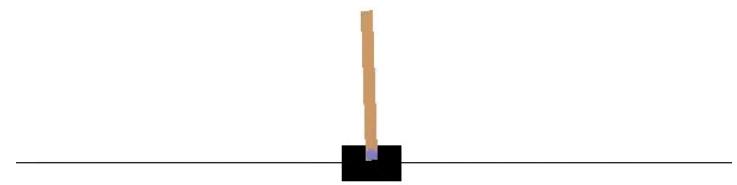
UNIVERSIDAD
DE GRANADA

Estimación $Q(s, a)$

Estado	Acción		
	A_1	\dots	A^m
S^1			
\dots			
S^n			

Problemas con espacio de estados o acciones continuas

No se pueden representar en una tabla Q



A partir de secuencias de episodios

$S_0 \ A_0 \ R_1 \ S_1 \quad | \quad A_1 \ R_2 \ S_2$

$$\rightarrow Q(S_0, A_0) \leftarrow Q(S_0, A_0) + \alpha \left(R_1 + \gamma \max_{a \in \mathcal{A}} Q(S_1, a) - Q(S_0, A_0) \right)$$
$$\pi \leftarrow \epsilon_0\text{-greedy}(Q)$$

$$Q(S_1, A_1) \leftarrow Q(S_1, A_1) + \alpha \left(R_2 + \gamma \max_{a \in \mathcal{A}} Q(S_2, a) - Q(S_1, A_1) \right)$$
$$\pi \leftarrow \epsilon_0\text{-greedy}(Q)$$

Experiencias se generan de forma secuencial

Dependencia entre *experiencias*
Algunas zonas son difíciles de explorar

Experiencias se descartan

Si aparece una *experiencia* interesante solo se utiliza una vez en el proceso de aprendizaje

Deep Q-Learning

Q-Learning



UNIVERSIDAD
DE GRANADA

Estimación $Q(s, a)$

$$f(s, a) \rightarrow \mathbb{R}$$

Modelar Q como una función no lineal

Se puede trabajar con estados y acciones continuos

A partir de pasos no secuenciales

$S_0 A_0 R_1 S_1$

$S_1 A_1 R_2 S_2$

...

Guardar una base de datos o memoria de *experiencias (rolling)*

Se puede utilizar en cualquier orden

Se puede utilizar más de una vez

Deep Q-Learning

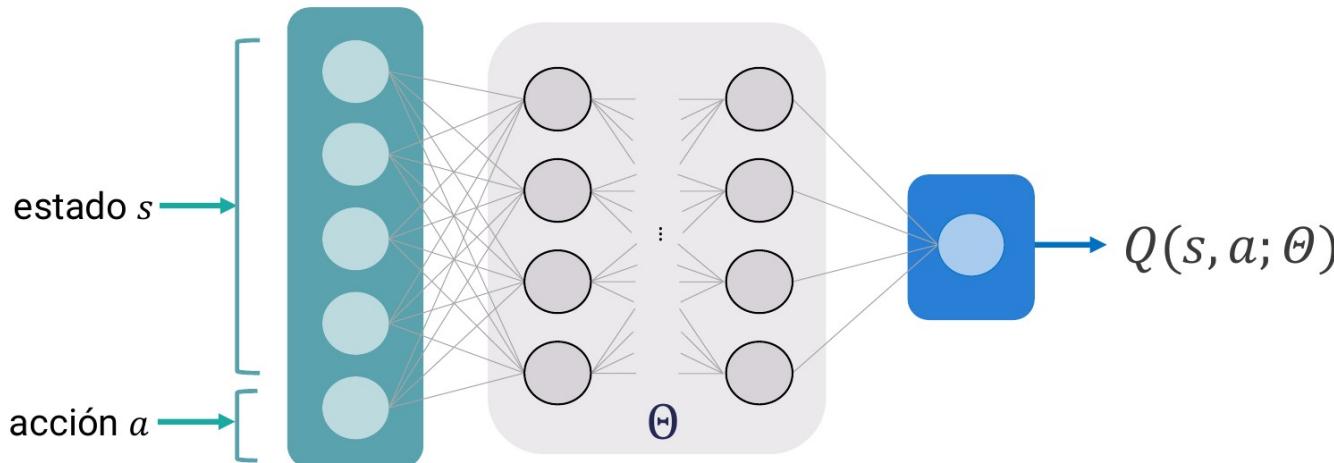
Estimación de Q con red neuronal

Estimación $Q(s, a)$

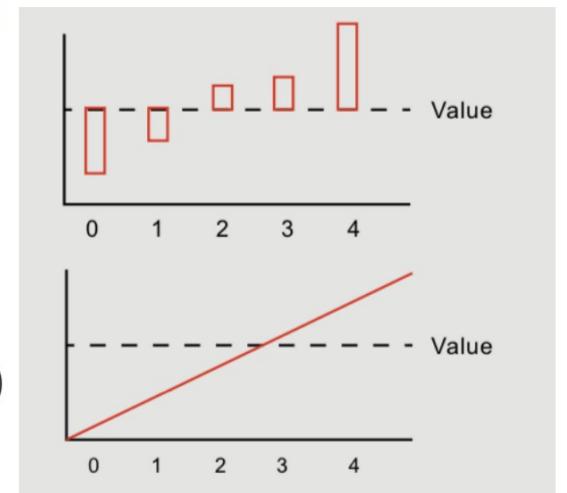
$$f(s, a) \rightarrow \mathbb{R}$$

Modelar Q como una función no lineal

Se puede trabajar con estados y acciones continuos



$$Q(s_0, a) = [-2.0, -1.0, 0.5, 1.0, 3.0]$$



$$Q(s_0, a; \theta)$$

Deep Q-Learning

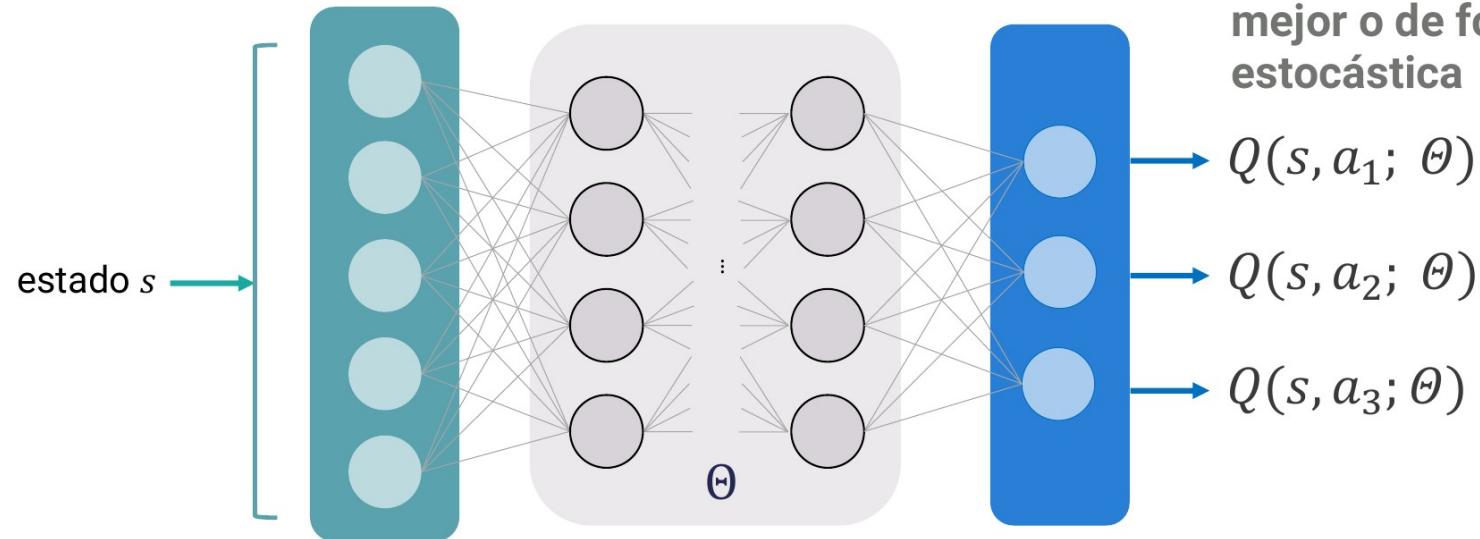
Estimación de Q con red neuronal

Estimación $Q(s, a)$

$$f(s, a) \rightarrow \mathbb{R}$$

Modelar Q como una función no lineal

Se puede trabajar con estados y acciones continuas



Seleccionar la
mejor o de forma
estocástica

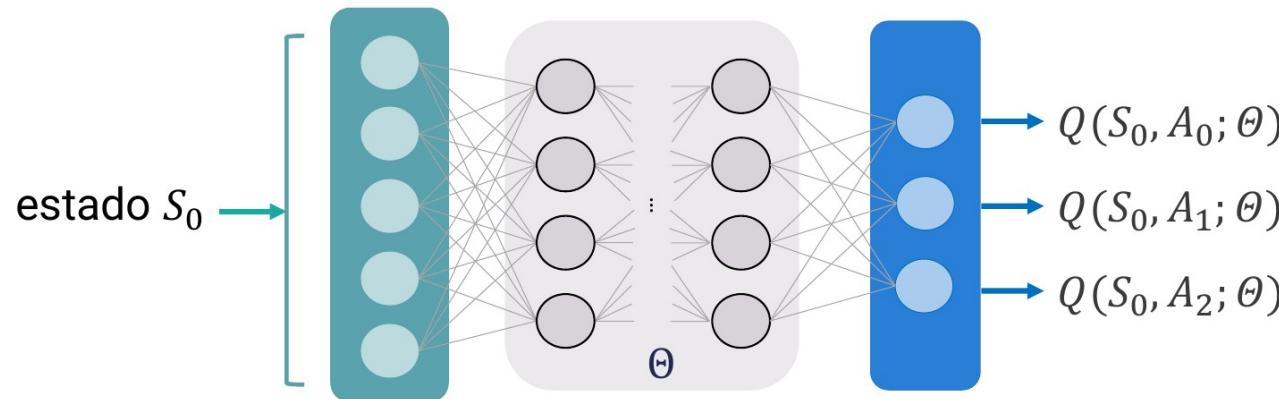
$$Q(s, a_1; \Theta)$$

$$Q(s, a_2; \Theta)$$

$$Q(s, a_3; \Theta)$$

Deep Q-Learning

DQN – Deep Q-Network



Entrenamiento (*learning*)

Entrada: S

Salida: $Q(S; \theta)$

Función de error: $\mathcal{L}(Q(S), Q(S; \theta))$

*¡No disponemos
de esta salida!*

Guardar una base de datos o memoria de *experiencias*

Se puede utilizar en cualquier orden y más de una vez

$$Q(S_0, A_0) \leftarrow Q(S_0, A_0) + \alpha \left(R_1 + \gamma \max_{a \in \mathcal{A}} Q(S_1, a) - Q(S_0, A_0) \right)$$

*Pero podemos estimarla a partir de:
 $S_0 \ A_0 \ R_1 \ S_1$*

Deep Q-Learning

Formulación DQN

$$\ell(x^{(i)}, \Theta) = \mathcal{L}(Q(S), Q(S, A; \Theta))$$

Con MSE: $E = \mathbb{E}_\pi \left[(Q(S, A) - Q(S, A; \Theta))^2 \right]$

Recordemos que: $\Delta\Theta = \Delta w = -\eta \frac{\partial E}{\partial w}$

Desarrollamos: $\frac{\partial E}{\partial w} = 2(Q(S, A) - Q(S, A; \Theta)) \frac{\partial Q(S, A; \Theta)}{\partial w}$ gradiente de los pesos red ∇_Θ

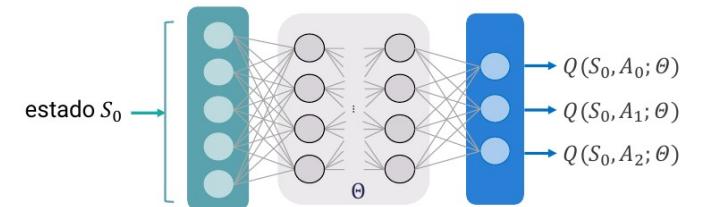
Sustituimos: $\Delta\Theta = -2\eta (Q(S, A) - Q(S, A; \Theta)) \nabla_\Theta Q(S, A; \Theta)$

Simplificamos y aplicamos la expresión de Q-Learning: $Q(S_0, A_0) \leftarrow Q(S_0, A_0) + \alpha \left(R_1 + \gamma \max_{a \in \mathcal{A}} Q(S_1, a) - Q(S_0, A_0) \right)$

$$\Delta\Theta = \alpha \left(R + \gamma \max_{a \in \mathcal{A}} Q(S', a; \Theta) - Q(S, A; \Theta) \right) \nabla_\Theta Q(S, A; \Theta)$$



UNIVERSIDAD
DE GRANADA



Deep Q-Learning

Formulación DQN



UNIVERSIDAD
DE GRANADA

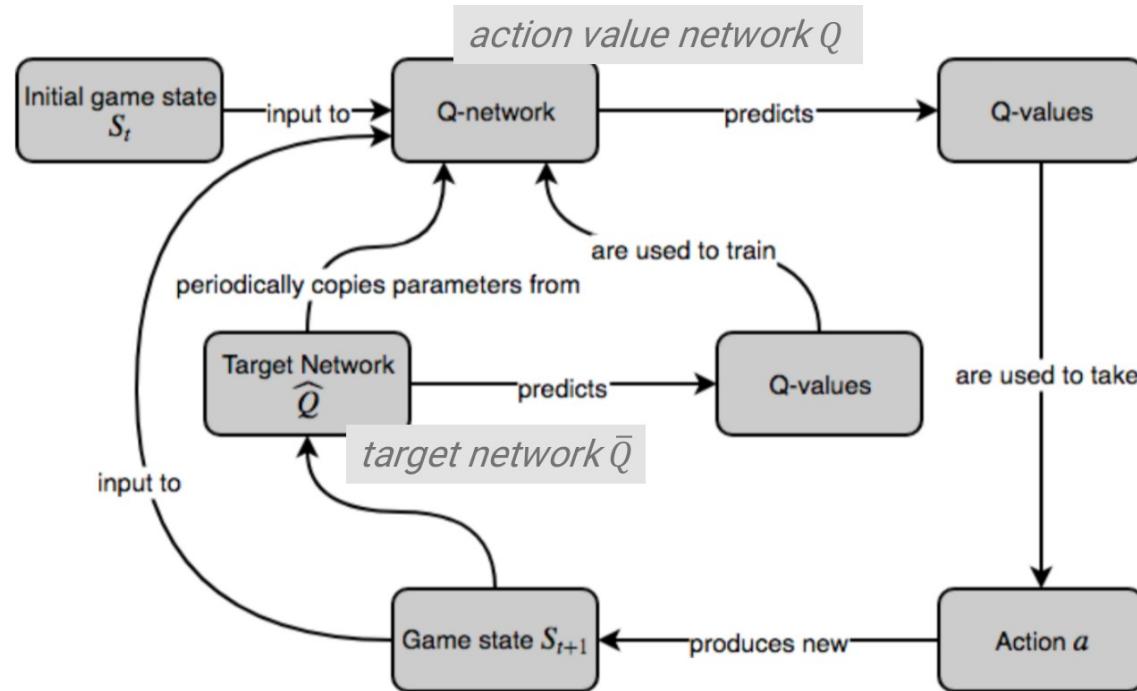
¿Qué significa esto?

- La expresión para **actualizar los pesos de la red es diferente** al algoritmo de gradiente descendente que vimos en el Tema 01
- Para actualizar los pesos de la red Q usando gradientes **es posible utilizar un *mini-batch*** extraido de la memoria de experiencias; esto es, un sub-conjunto $S_t \ A_t \ R_{t+1} \ S_{t+1}$
- Estamos actualizando una aproximación con una aproximación. Por ello, **se suelen utilizar dos redes Q** (misma topología; los pesos Θ^- de la red *target* \bar{Q} se actualizan con menor frecuencia a partir de la red *action value* Q)

$$\Delta\Theta = \alpha \left(R + \gamma \max_{a \in \mathcal{A}} \bar{Q}(S', a; \Theta^-) - Q(S, A; \Theta) \right) \nabla_\Theta Q(S, A; \Theta)$$

Deep Q-Learning

Algoritmo DQN



A. Zai, B. Brown (2018) *Deep Reinforcement Learning in Action*. Manning.



V. Mnih et al. (2015) *Human-level control through deep reinforcement learning*. Nature 518, 529-533

Deep Q-Learning

Algoritmo DQN



UNIVERSIDAD
DE GRANADA



Deep Mind DQN (2016). Más: <https://deepmind.com/research/dqn/>

Deep Q-Learning

Algoritmo DQN



UNIVERSIDAD
DE GRANADA

Deep Q-Learning

params: Q y \bar{Q} iniciales, D inicial, $n_{\text{episodios}}$, $n_{\text{entrenamiento}}$, n_{batch} , γ , C , τ

for $i = \{0, \dots, n_{\text{episodios}}\}$

 for $t = \{0, \dots, n_{\text{entrenamiento}}\}$

 elegir acción A_t con política ϵ -greedy(Q)

 aplicar A_t y obtener R_{t+1}, S_{t+1}

 almacenar $S_t A_t R_{t+1} S_{t+1}$ en memoria D

 muestrear un mini-batch de experiencias $\langle S_j A_j R_{j+1} S_{j+1} \rangle$ de tamaño n_{batch} de la memoria D

 obtener salida de predicción de cada experiencia:

$$y_j = \begin{cases} R_j & \text{(si el episodio termina en } j+1, \text{ o en otro caso)} \\ R_j + \gamma \max_{a \in \mathcal{A}} \bar{Q}(S_{j+1}, a; \theta^-) & \end{cases}$$

 optimizar mediante gradiente descendente sobre $(y_j - Q(S_j, A_j; \theta))^2$

 update \bar{Q} | cada C iteraciones, actualizar \bar{Q} a partir de Q ; por ejemplo, $\theta^- = \tau \theta^- + (1 - \tau) \theta$

Deep Q-Learning

Avanzado

Convergencia



- Z. Yang, Y. Xie, Z. Wang (2019) *A Theoretical Analysis of Deep Q-Learning*. <https://arxiv.org/abs/1901.00137v2>
S.J. Lee (2019) *The Deep Q-Network Book*. <https://www.dqnbook.com>

Double DQN

DQN tiende a sobreestimar los valores de Q , sobre todo al principio del algoritmo >> Evaluar cómo de buena es A_t con θ^-

Memoria de experiencias con prioridad

La memoria tiene un tamaño limitado >> Mantener experiencias antiguas valiosas
La memoria es muestreada de forma uniforme >> Definir probabilidad no uniforme

Dueling networks

Utilizar una arquitectura diferente, de forma que: $V(s) = Adv(s) + Q(s, a)$, donde $Adv(s)$ es la función de ventaja

Otras técnicas

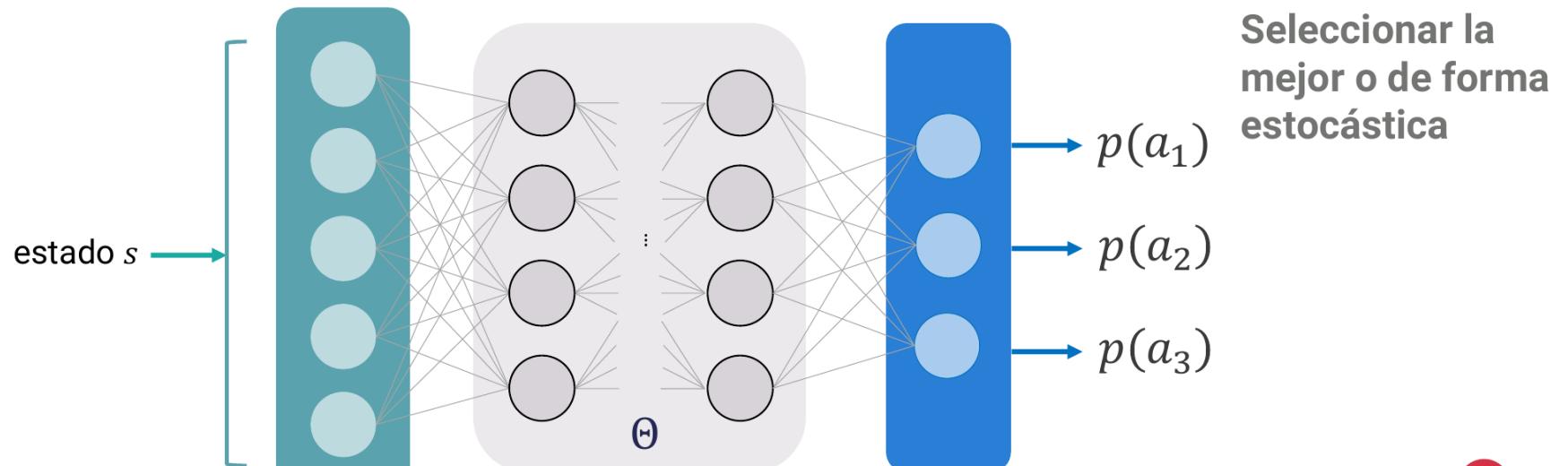
Distributional DQN, Noisy DQN, Rainbow

Métodos basados en política

Concepto

Hasta ahora, hemos intentado estimar Q , y a partir de la ésta, conseguir la política óptima

¿Por qué no intentar calcular directamente qué acción es mejor, sin necesidad de estimar Q ? **Métodos basados en optimización de política** (vs Métodos basados en estimación de valor)



Actor critic



Métodos *actor-critic*

Concepto



UNIVERSIDAD
DE GRANADA

Método basados en política (REINFORCE)

calcular probabilidad de que una acción conduzca a un buen resultado final (ACTUAR)

Problema: una trayectoria que conduzca a un mal resultado final puede contener buenas acciones

+

Métodos basados en valor (DQN)

estimar el valor de recompensa obtenido (ESTIMAR)

Problema: se estima a partir de estimaciones, lo cual introduce sesgo en el aprendizaje

=

Métodos *actor-critic*

Combinan ambas aproximaciones

Utilizan dos redes: *actor* (similar a la red de REINFORCE - π), *critic* (similar a la red de DQN - V)

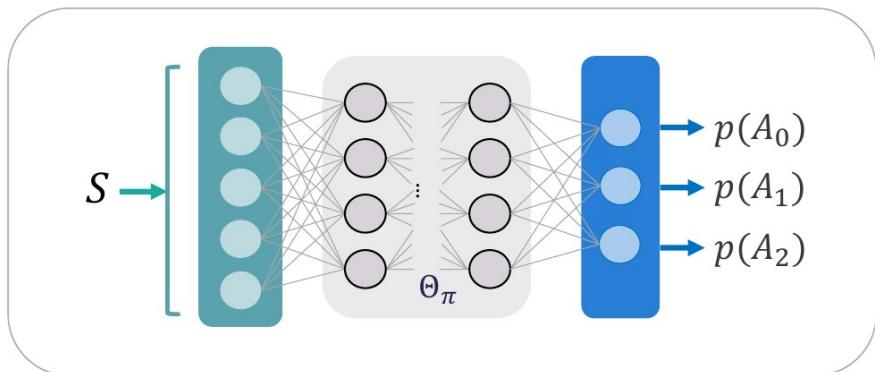
Ventajas: acelerar el aprendizaje (más rápidos), evitar sesgos (más estable)

Métodos *actor-critic*

Concepto

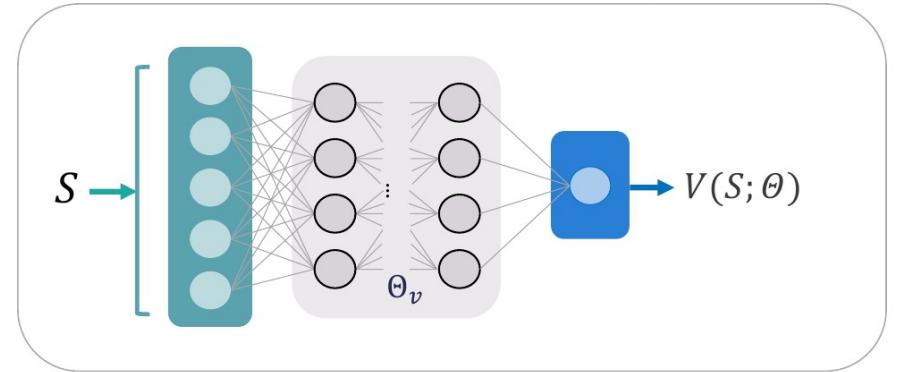
actor

$$\pi(A|S; \Theta_\pi)$$



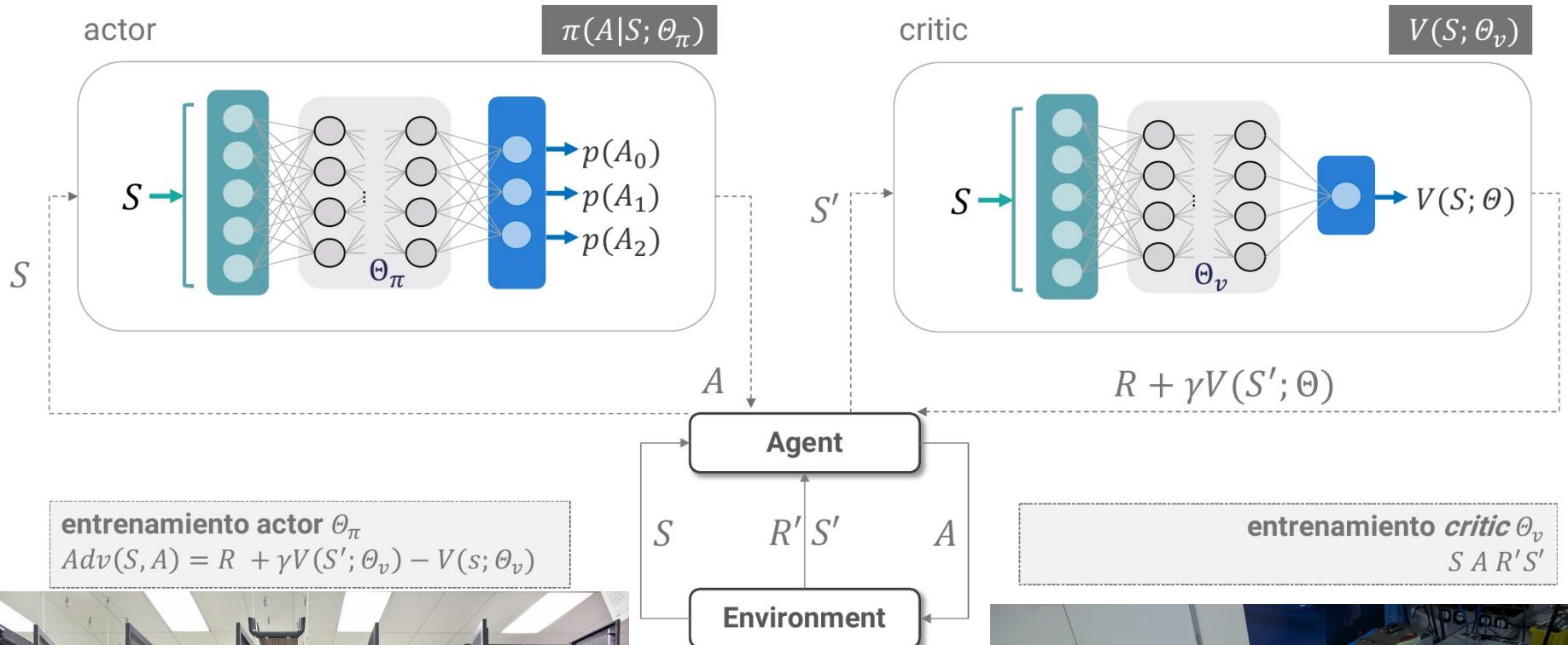
critic

$$V(S; \Theta_v)$$



Métodos *actor-critic*

Algoritmo



Métodos *actor-critic*

Avanzado



UNIVERSIDAD
DE GRANADA

Mejoras

Arquitectura paralela con múltiples agentes que aprenden simultáneamente, en lugar de implementar memoria de experiencias y mini-batches

- *A3C : Asynchronous Advantage Actor-Critic (2016)*
Cada agente utiliza una copia local de la red *actor-critic*
Periódicamente, cada agente actualiza de forma asíncrona la red *actor-critic* global
- *A2C : Advantage Actor-Critic (2016)*
Los agentes actualizan de forma síncrona la red *actor-critic*



V. Mnih et al. (2016) *Asynchronous Methods for Deep Reinforcement Learning*. <https://arxiv.org/abs/1602.01783>

Métodos *actor-critic*

Avanzado

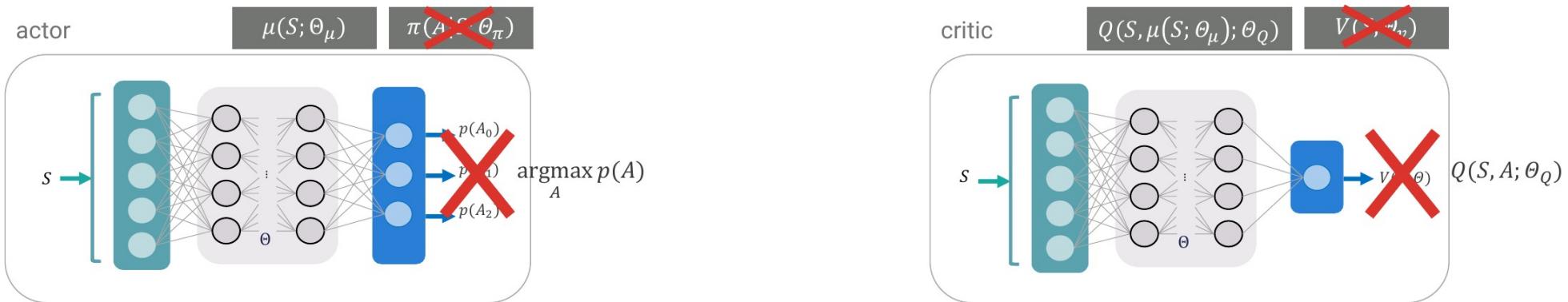


UNIVERSIDAD
DE GRANADA

Mejoras

Aproximaciones para espacios de acciones continuos

- *DDPG : Deep Deterministic Policy Gradient Continuous Action Space* (2015)



T. Lillicrap et al. (2015) *Continuous control with Deep Reinforcement Learning*. <https://arxiv.org/abs/1509.02971>

Últimos slides

Que hacen los picantes?



AlphaGo

Conceptos fundamentales



UNIVERSIDAD
DE GRANADA

Agente capaz de jugar al *Go* mejor que los humanos

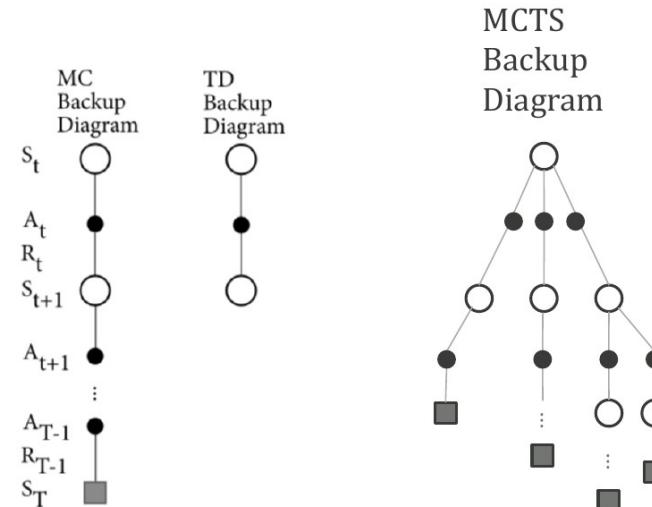
Aproximación

Resolución de juegos mediante árboles de búsqueda

Aprendizaje profundo por refuerzo

Método *actor-critic*

Rollout utilizando Monte Carlo Tree Search (MCTS)



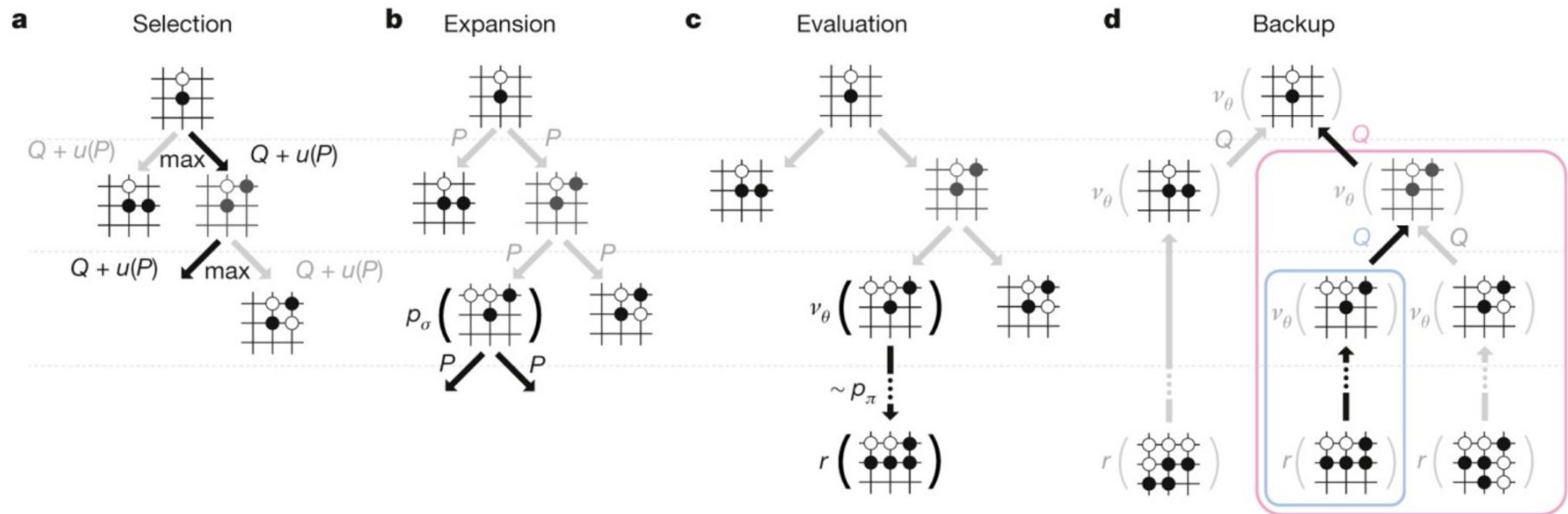
D. Silver et al. (2016) *Mastering the game of Go with Deep Neural Networks & Tree Search*. Nature 529, 484-489

AlphaGo

Conceptos fundamentales



UNIVERSIDAD
DE GRANADA



D. Silver et al. (2016) *Mastering the game of Go with Deep Neural Networks & Tree Search*. Nature 529, 484-489
M. Pumperla, K. Ferguson (2019) *Deep Learning and the Game of Go*. Manning.

AlphaGo

Versiones avanzadas



UNIVERSIDAD
DE GRANADA

AlphaZero (2017)

Aprende jugando contra sí mismo

AlphaStar (2019)

Es capaz de vencer a humanos en StarCraft 2.0

OpenAI DOTA 2 (2019)

Es capaz de vencer a humanos en DOTA 2

Pluribus (2019)

Es capaz de vencer a humanos en Poker



D. Silver et al. (2017) *A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play*. Science 362(6419), 1140-1144



O. Vinyals et al. (2019) *AlphaStar: Mastering the Real-Time Strategy Game StarCraft II*. <https://deepmind.com/blog/alphastar-mastering-real-time-strategy-game-starcraft-ii/>



J. Pachocki et al. (2019) *OpenAI Five*. <https://openai.com/five/>



N. Brown, T. Sandholm (2019) *Superhuman AI for multiplayer poker*. Science 11 July 2019.



Implementación de algoritmo DQN en colab

A close-up photograph of a baby with light brown hair and blue eyes. The baby has a slightly grumpy or unimpressed expression, with a small frown. They are wearing a green and white long-sleeved shirt. In their hands, they are holding a light-colored, textured rock. The background is a soft-focus outdoor scene, possibly a beach or a park.

End of presentation

Thank You

Gracias por su atención