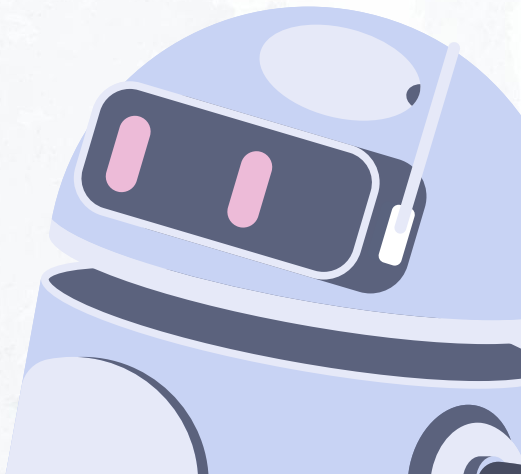


# Reinforcement Learning



(RL)

# Table of contents

- 01 → Que es el aprendizaje automático?
- 02 → Fundamentos teoricos
- 03 → Presentación del proyecto
- 04 → Demo del proyecto

01 →

# Que es el aprendizaje automático?

(AI)

# Aprendizaje Automático

En el aprendizaje de máquinas un computador observa datos, construye un modelo basado en esos datos y utiliza ese modelo a la vez como una hipótesis acerca del mundo y una pieza de software que puede resolver problemas



# Tipos de aprendizaje automatico

(AI)



# Tipos de Aprendizaje Automático

## (a) Aprendizaje no supervisado

Un modelo que intenta identificar patrones y segmentar un conjunto de datos sin tener etiquetas previamente



## (b) Aprendizaje por refuerzo

Un modelo que, mediante ensayo y error, intenta generar una concepción de modelo del mundo para poder interactuar con el



02 →

# Fundamentos teóricos

# Partes del RL

## (a) States →

Representación del environment con el que está interactuando el agente

## (b) Actions →

Decisiones que puede tomar el agente según las reglas de juego o comportamiento que se quiere adquirir

## (c) Agents →

Modelo que se encarga de estar al tanto del state actual y predecir que actions tomar para ir acercándose a la recompensa que fue seteada por un usuario



# Partes del RL

## (a) Short-term memory →

Esta es la memoria que tiene el modelo para recordar las jugadas que hizo en la iteración(o partida) actual

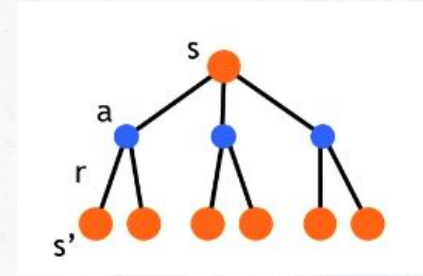
## (b) Long-term memory →

Esta es la memoria que almacena las iteraciones anteriores y las acciones que se tomaron.

## (c) Rewards →

Es la recompensa que se le da al agente cuando toma un acción deseada. Lo opuesto también aplica al generar acciones indeseadas. Se genera con la Value Function

# Value Function



Discount Factor

$$Q_{t+1}(s_t, a_t) = Q_t(s_t, a_t) + \alpha \left( R_{t+1} + \gamma \max_a Q_t(s_{t+1}, a) - Q_t(s_t, a_t) \right)$$

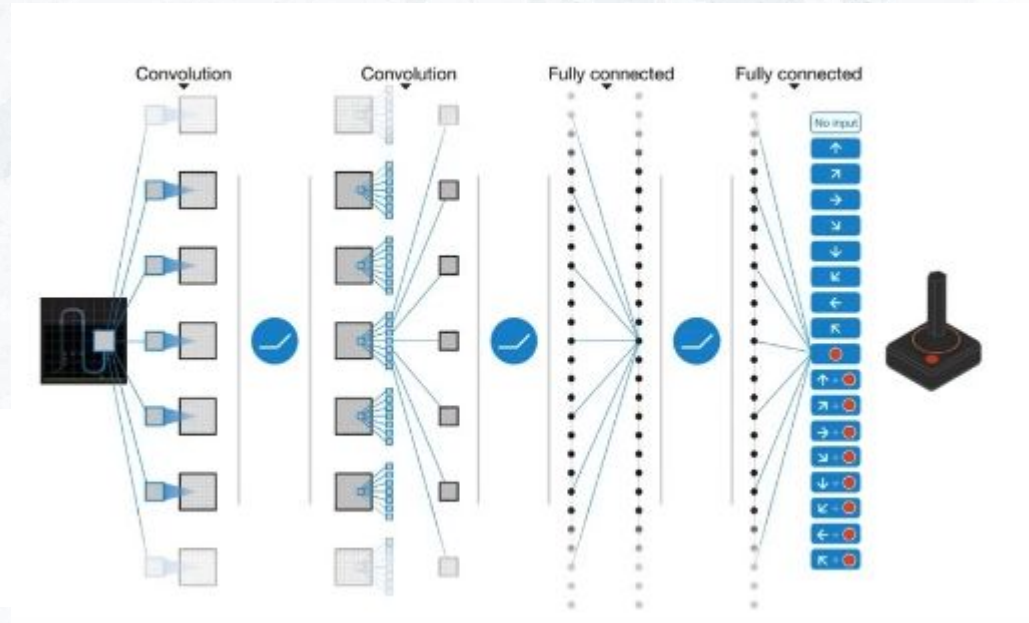
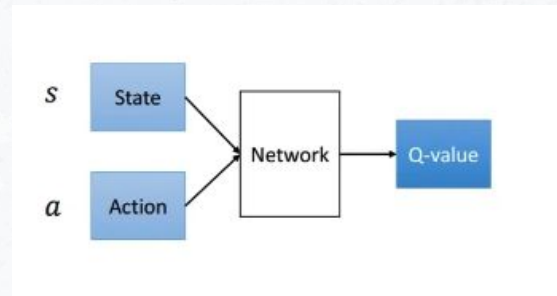
New State

Old State

Reward

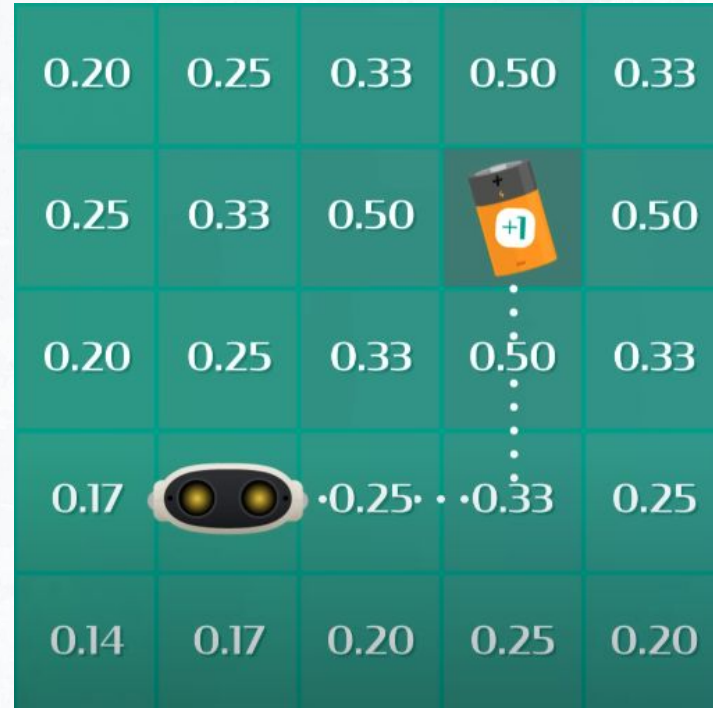
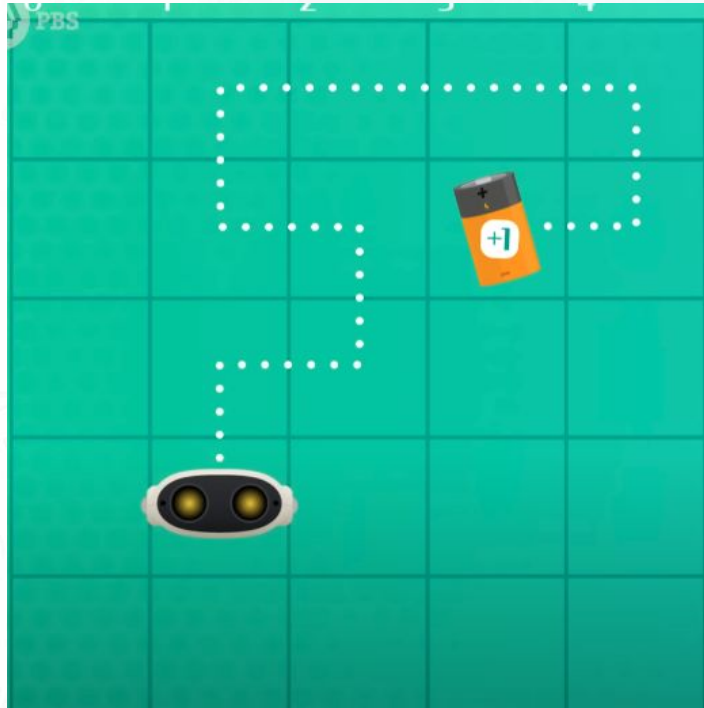
# Value Function

	A1	A2	A3	A4
S1	+1	+2	-1	0
S2	+2	0	+1	-2
S3	-1	+1	0	-2
S4	-2	0	+1	+1



$$L = \mathbb{E}[\underbrace{(r + \gamma \max_{a'} Q(s', a'))}_{\text{target}} - \underbrace{Q(s, a)}_{\text{prediction}})^2]$$

# Value Function



## Agent

- game
- model

### Training:

- state = get\_state(game)
- action = get\_move(state):
  - model.predict()
- reward, game\_over, score = game.play\_step(action)
- new\_state = get\_state(game)
- remember
- model.train()

## Game (Pygame)

- play\_step(action)
  - > reward, game\_over, score

## Model (PyTorch)

### Linear\_QNet (DQN)

- model.predict(state)
  - > action