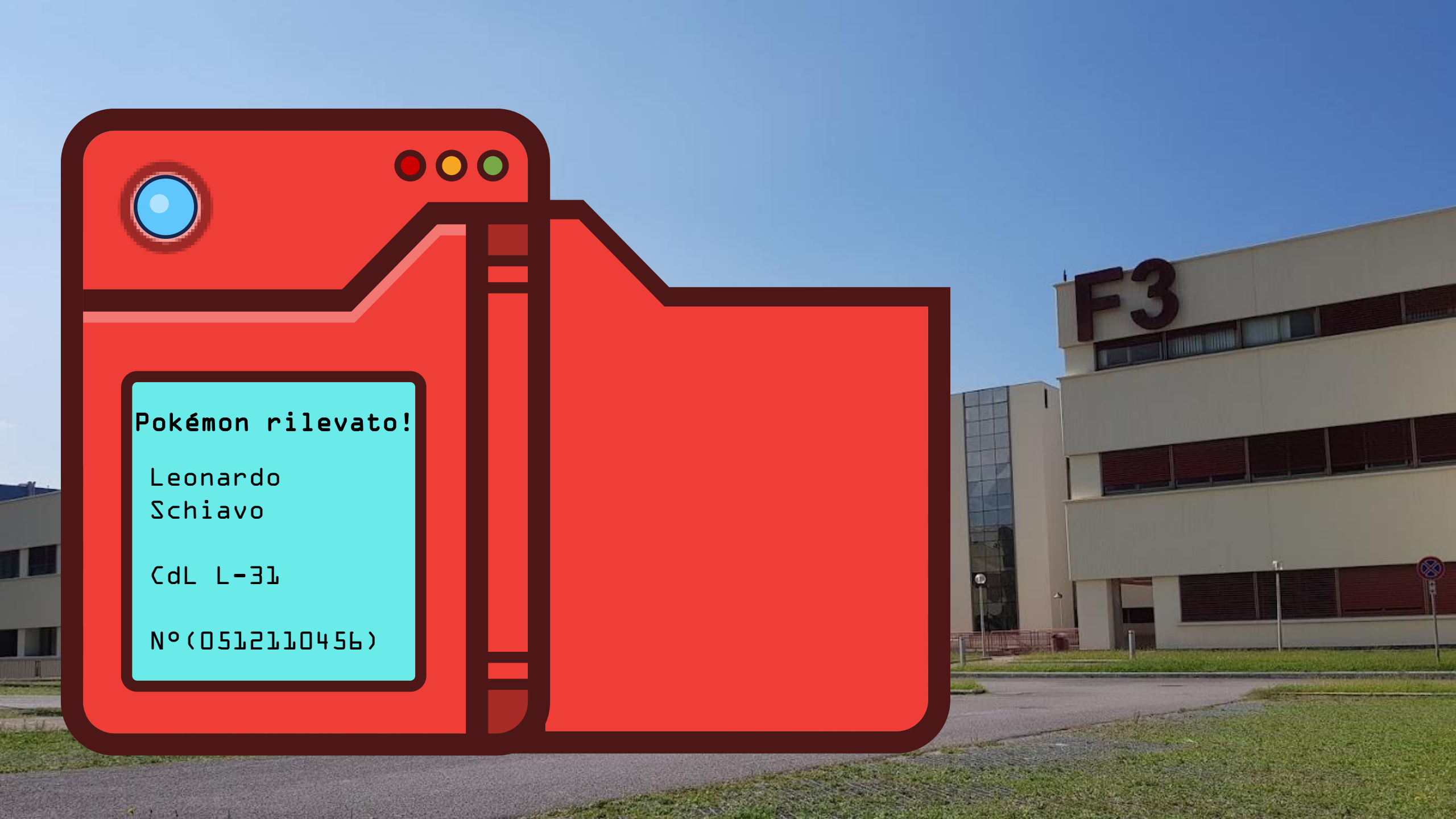
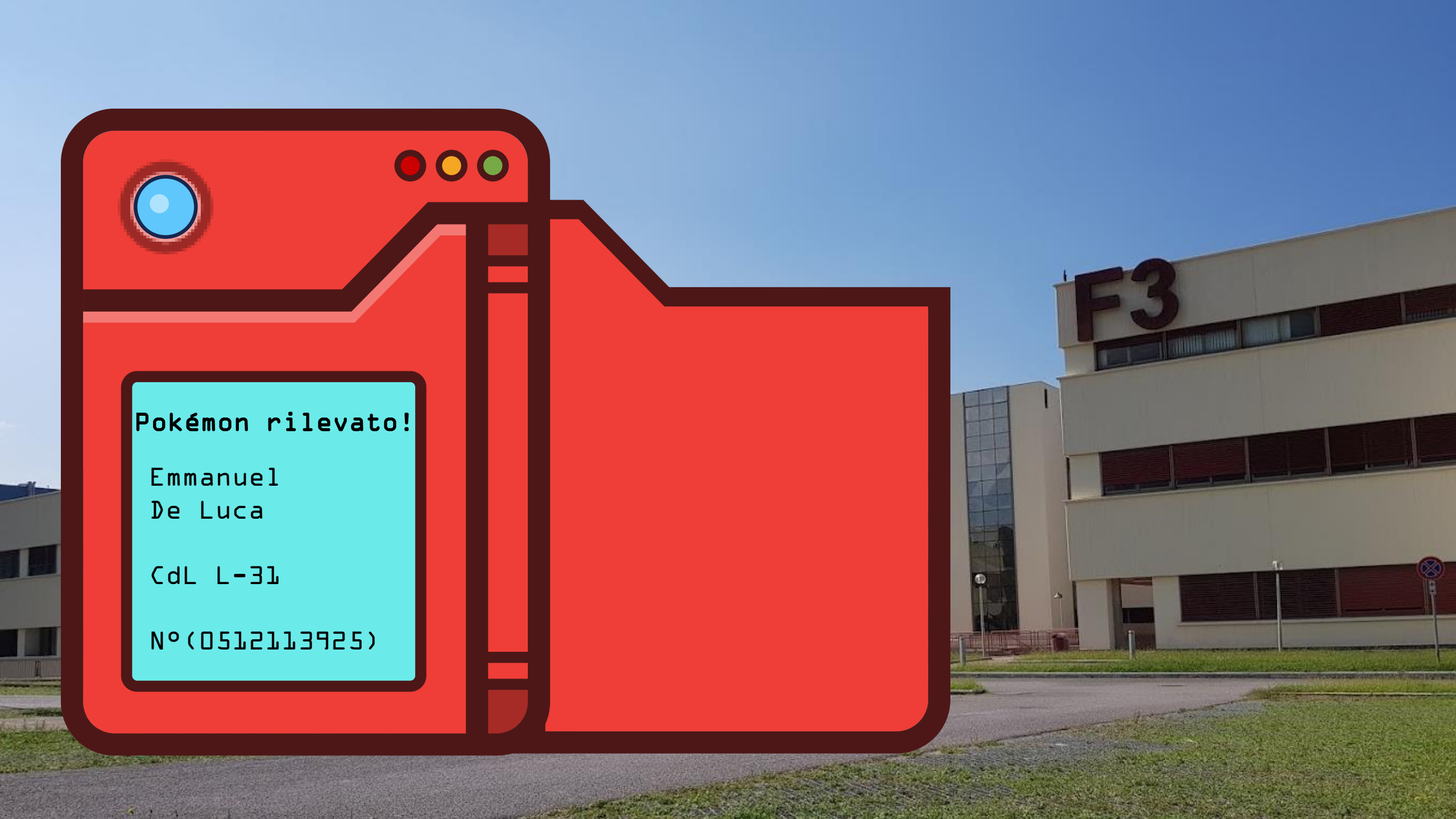
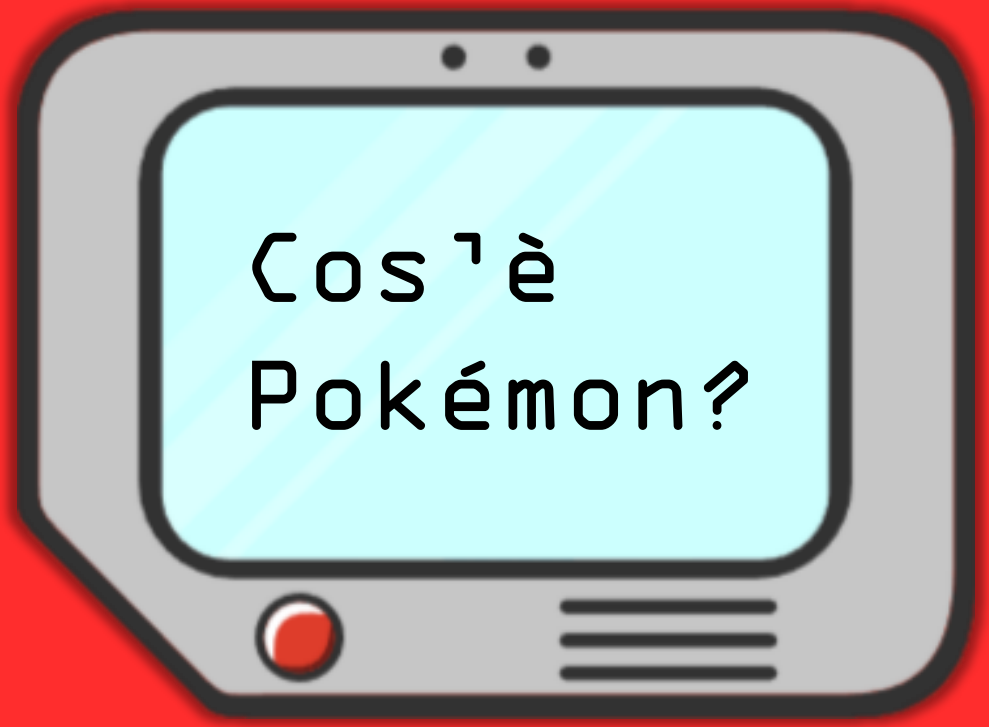
Cydon-IA


F3
Pokémon rilevato!
Leonardo
Schiavo
CdL L-31
N°(0512110456)


F3
Pokémon rilevato!
Emmanuel
De Luca
CdL L-31
N°(0512113925)



Pokémon è un franchise giapponese di proprietà di **The Pokémon Company** creato nel 1996 da **Satoshi Tajiri.**

La sua nascita coincide con la pubblicazione di 2 videogiochi:

- **Pokémon Versione Rossa**
- **Pokémon Versione Verde**

Meccanica principale del gioco:

Lotte Pokémon

Ma cosa sono?

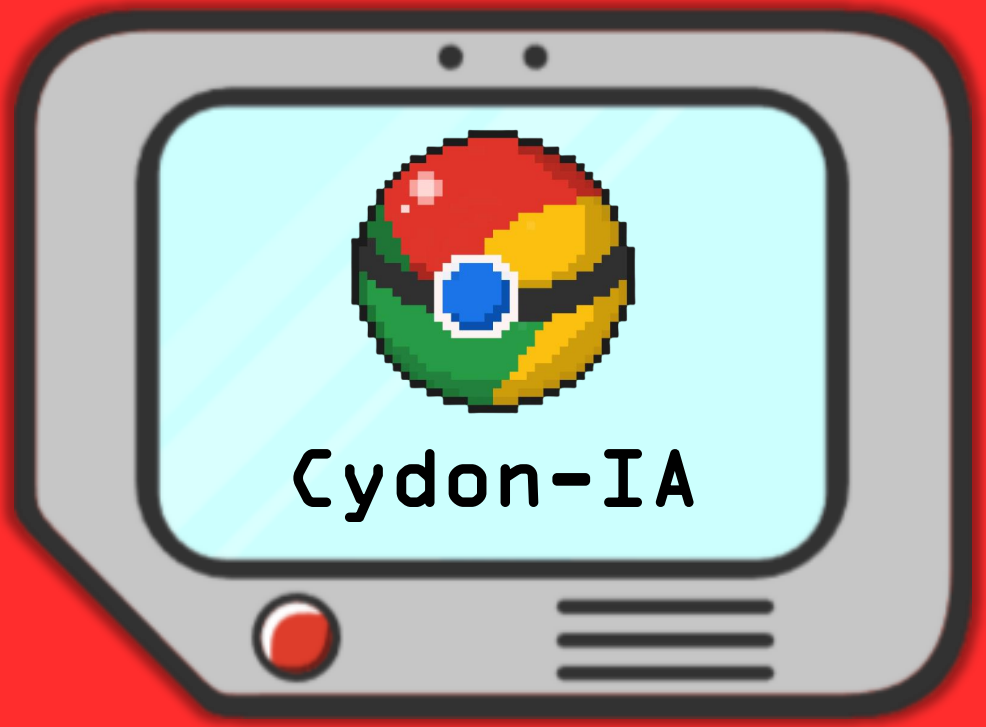
Sono l'elemento centrale dell'intero franchise.

2 allenatori si sfidano facendo combattere i loro Pokémon.
Le battaglie sono influenzate da diversi fattori,
quali le mosse, le abilità, le statistiche e i tipi dei Pokémon.
Possono avvenire tra 2 giocatori reali e tra 1 giocatore e 1 NPC (Non Playable Character)

Criticità delle lotte Pokémon

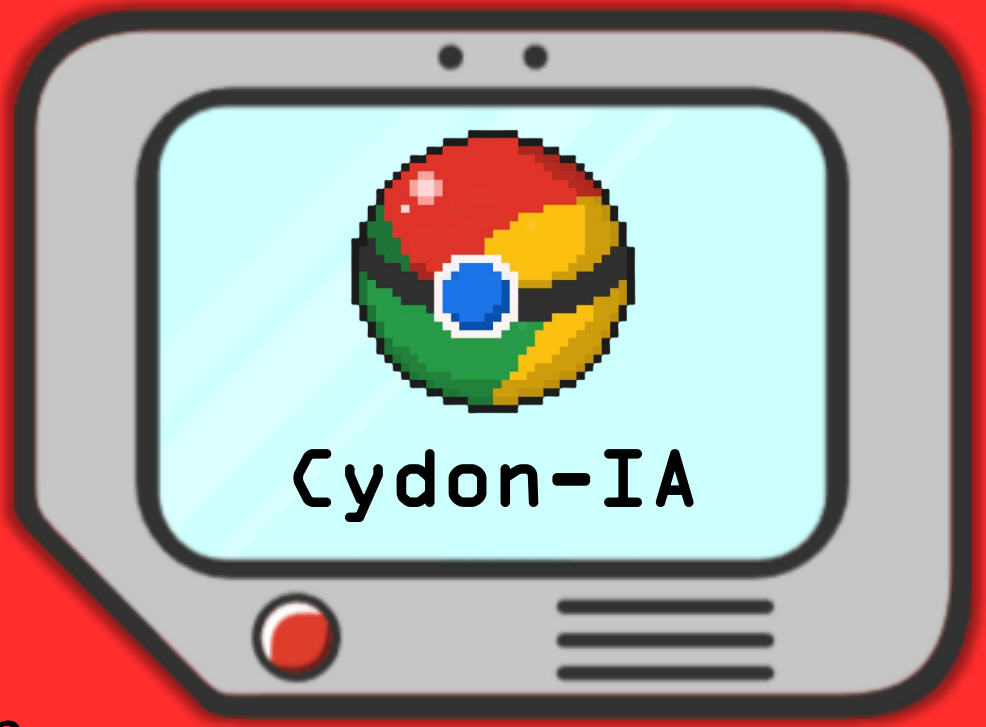
Le battaglie Pokémon sono complesse e la varietà di scenari in cui ci si potrebbe trovare all'interno di un singolo turno potrebbe mettere in difficoltà un giocatore inesperto, portandolo a rinviare le battaglie più difficili aspettando che i suoi Pokémon diventino più forti anziché adottare sin da subito strategie efficaci.

Basti pensare che un singolo turno, seguendo la nostra astrazione, conta più di 27 diramazioni possibili per ciascun Pokémon (questo senza contare le varie combinazioni di statistiche, abilità e mosse possibili per ogni Pokémon).



Cydon-IA

Volendo creare un'AI che permetta anche ad un giocatore inesperto di approcciarsi a questo complesso mondo e di comprendere quali siano le «best practices» del gioco, nasce l'idea di Cydon-IA.



Cosa significa fa apprendere a giocatori novizi come approcciarsi al gioco?

Vuol dire fa comprendere al giocatore, tramite l'intelligenza artificiale, come azioni che sembrano dare uno svantaggio iniziale, protratte nel lungo termine possano effettivamente portare alla vittoria.

Pokémon non è un gioco in cui la mossa più «potente» vince, ma è un gioco in cui **la mossa giusta al momento giusto** ti porta la vittoria.

Performance: le misure di prestazione scelte sono:

- **Tasso di vittoria**: indica la percentuale di battaglie vinte dall'agente rispetto al totale delle battaglie affrontate.
- **Efficacia delle mosse suggerite**: indica quanto le mosse suggerite dall'agente portano a risultati positivi.
- **Adattabilità**: indica quanto bene l'agente si comporta in scenari variabili.

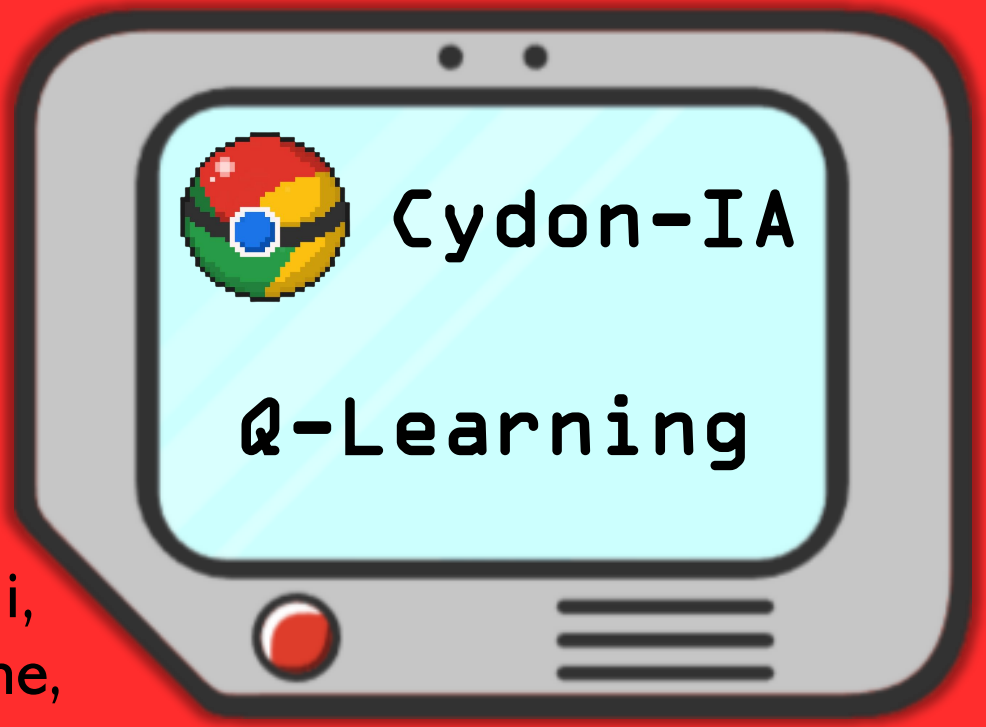
Enviroment: l'ambiente in cui opera l'agente è rappresentato dallo stato attuale della battaglia (nome dei Pokémon in campo, salute, status, malus).

Actuators: l'agente interviene nell'ambiente suggerendo la mossa più conveniente per vincere la battaglia.

Sensors: la percezione dell'ambiente da parte dell'agente è data da un insieme di dati che costituiscono una rappresentazione dello stato corrente della battaglia.

Specifica dell'ambiente:

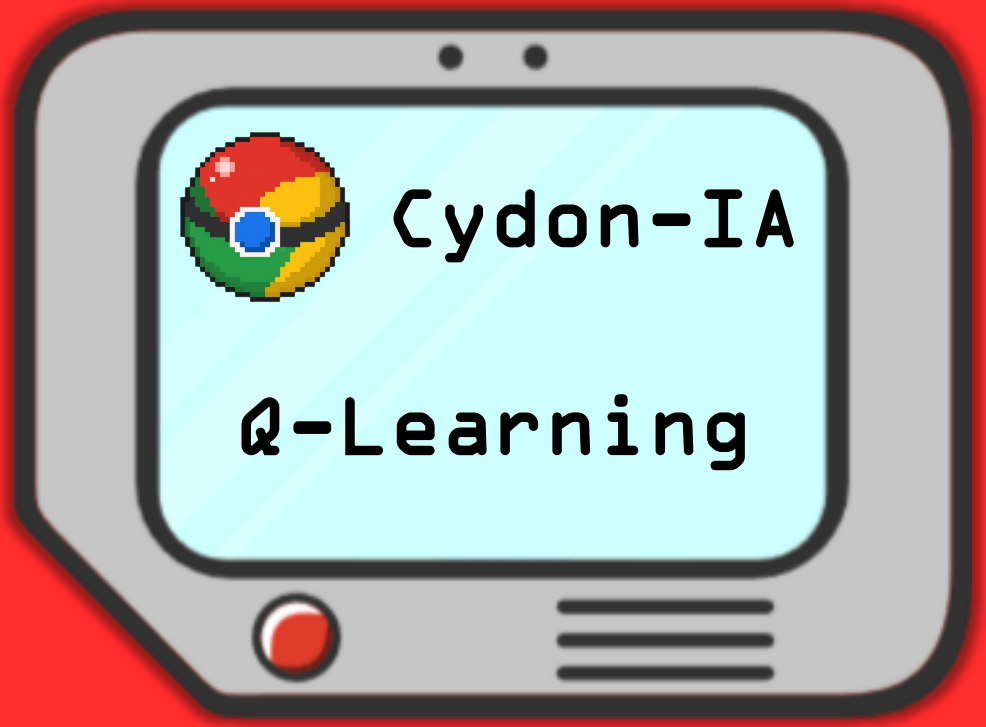
- Singolo agente
- Completamente osservabile
- Stocastico e simil black box
- Sequenziale
- Statico
- Discreto



Usare il Reinforcement Learning nella sua variante del Q-Learning, perché?

Perchè permette all'AI di imparare strategie possibili, esplorarle e migliorarle, e non solo di dare quella che, dopo una sola computazione, sembra essere quella migliore.

Usare un approccio basato sull'apprendimento e sullo sviluppo di strategie adattive, interagendo direttamente con l'ambiente stesso, risulta essere l'approccio migliore.

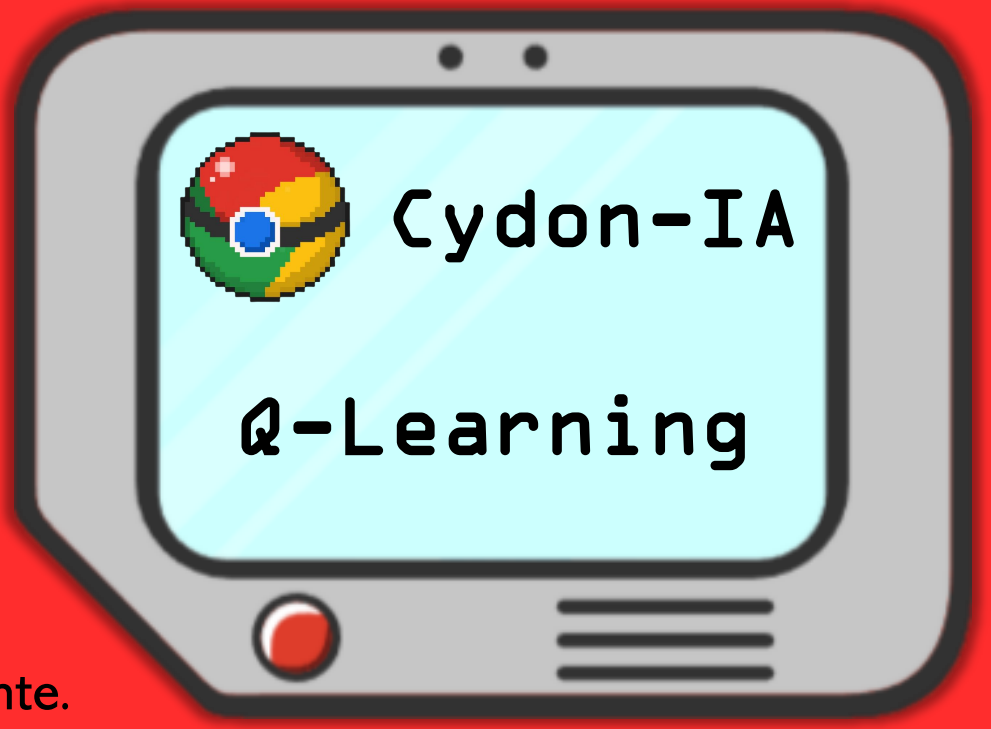


Cos'è il Reinforcement Learning?

È un paradigma nell'ambito dell'apprendimento automatico in cui un agente impara a compiere azioni ottimali per raggiungere determinati obiettivi interagendo direttamente con un ambiente.

Cos'è il Q – Learning?

È una variante del RL che mira a trovare la migliore azione per lo stato corrente, basandosi su una funzione Q , che associa ad ogni coppia stato-azione un valore indice di qualità della mossa in quel determinato stato.



Come agisce il q-learning?

1. Crea una matrice Q_0 (composta da tutti zero).
2. Sceglie uno stato casuale dallo spazio degli stati in caso di apprendimento batch, oppure usa lo stato corrente passato al modello in caso di apprendimento online.
3. Sceglie un'azione casuale tra quelle disponibili nell'ambiente.
4. Esegue l'azione, osserva l'evoluzione dell'ambiente in base ad essa e ne calcola il reward associato.
5. Normalizza il valore usando la formula tipica del q-learning:

$$Q(s,a) \leftarrow (1-\alpha) \cdot Q(s,a) + \alpha \cdot (r + \gamma \cdot \max a' Q(s',a'))$$

6. Inserisce il reward nella matrice Q, composta da N righe ed M colonne, nella posizione $[i][j]$, dove sulle righe vengono rappresentati gli stati e sulle colonne vengono rappresentate le azioni e i rappresenta la i-esima riga e la j rappresenta la j-esima colonna.

Come funziona il reward in Cydon-IA?

Il reward di Cydon-IA è deciso in funzione del delta del turno e in base all'azione associata.

Al turno stesso, quindi la valutazione parte dal delta e in base ad esso e all'azione si assegna un valore possibile tra 1, -1, 100.

Calcolo del delta relativo al turno

Per definire se un turno risulta vantaggioso per il Pokémon attuale, viene usata una funzione euristica che valuta la situazione attuale tramite una formula basata sui danni inflitti da noi
e dal Pokémon avversario, valutando se in quel turno il delta indica un ambiente favorevole o meno per il Pokémon attualmente in campo.
Il delta viene rappresentato da un booleano.

$$\textbf{damage\_dealt\_by\_me} = \frac{(atk_1 + spatk_1) \cdot first\_mon\_type\_advantage \cdot my\_buff \cdot opponent\_status}{de_2 \cdot hp_2}$$

$$\textbf{damage\_dealt\_by\_me} = \frac{(atk_2 + spatk_2) \cdot second\_mon\_type\_advantage \cdot opponent\_buff \cdot my\_status}{de_1 \cdot hp_1}$$

$$\textbf{damage\_advantage} = damage\_dealt\_by\_me \geq damage\_dealt\_to\_me$$

Assegnamento del reward

Se osserviamo il suggerimento di una mossa aggressiva, allora il modello prima di valutare il reward stesso andrà a valutare se il delta risulta positivo.
Se osserviamo il suggerimento di una mossa difensiva, si andrà a valutare il delta come negativo.

Successivamente dev'essere valutata l'azione compiuta, dando un valore arbitrario pari a:

- 1 nel caso in cui l'azione viene valutata coerente al delta
- -1 nel caso di una mossa non coerente al delta
- 100 nel caso di mossa che porta allo stato finale

Qual è lo stato finale?

Lo stato in cui il Pokémon avversario raggiunge 0 HP.

Come avviene la computazione del valore migliore?

Il risultato finale sarà dato dal massimo valore che la funzione Q riesce ad assumere nelle varie epoche, per le varie azioni, per quel dato stato.

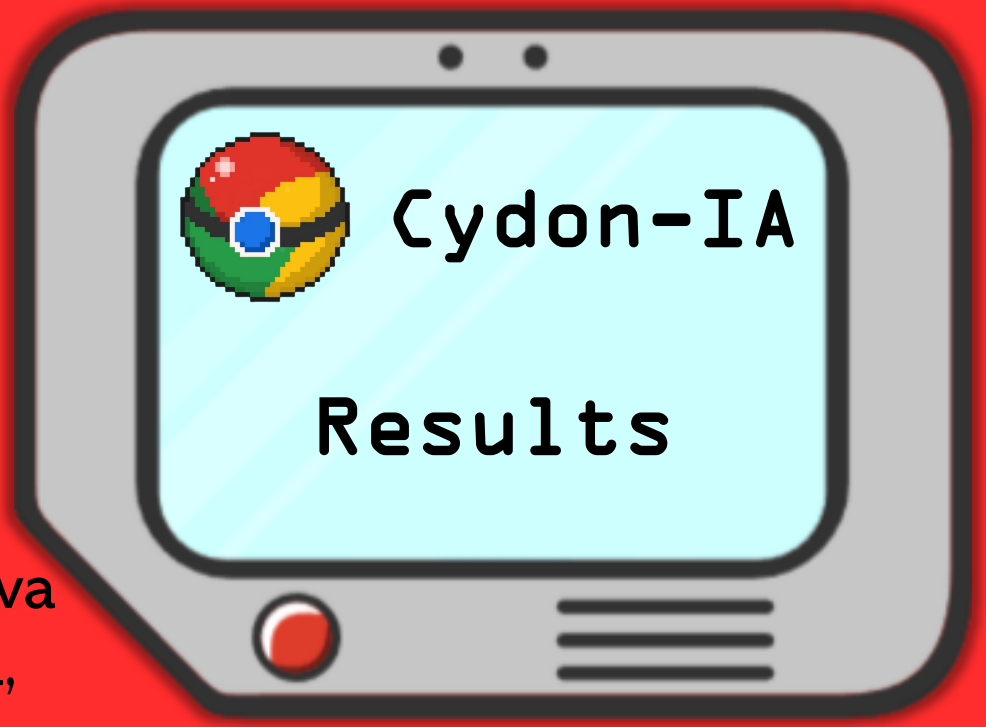
Quindi, per uno stato X, sarà associato a una generale azione Y un valore Q che equivale alla computazione della formula Q basata, nel nostro caso, sulla formula standard del Q-Learning e sulla strategia di esplorazione «Epsilon Greedy».

Il valore finale, ossia l'azione ottimale, sarà quella in cui la funzione Q avrà assunto il $Max(State\_Index)$ massimo per lo stato nella nostra matrice Q.

Risultati del modello

Il modello:

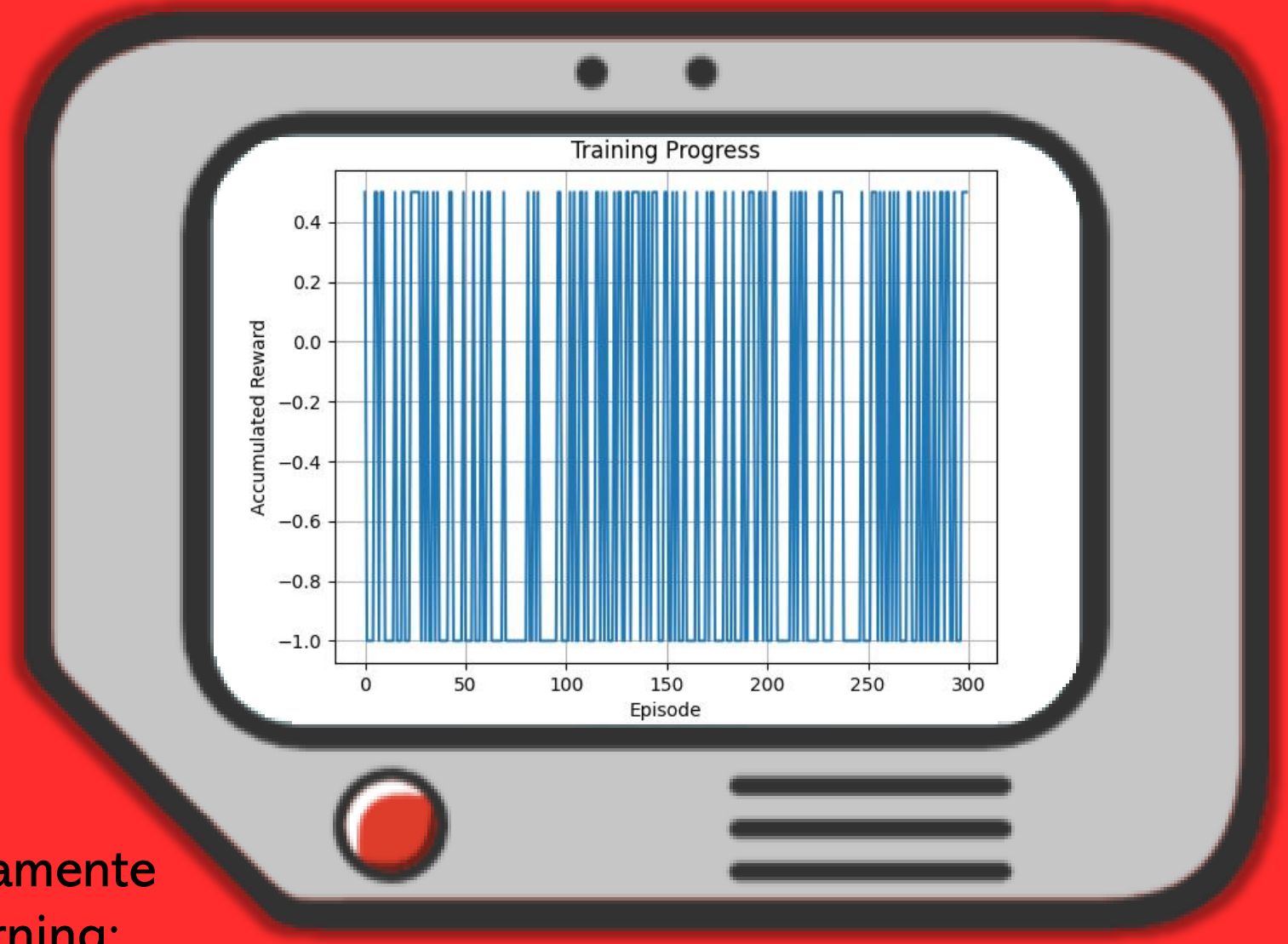
- per situazioni già affrontate più volte in passato, risulta essere efficiente.
- per situazioni sporadiche, di cui l'agente non aveva già delle epoche di addestramento ed esperienza, risulta suggerire mosse poco ottimali ed efficienti.



Quindi è tutto da scartare?

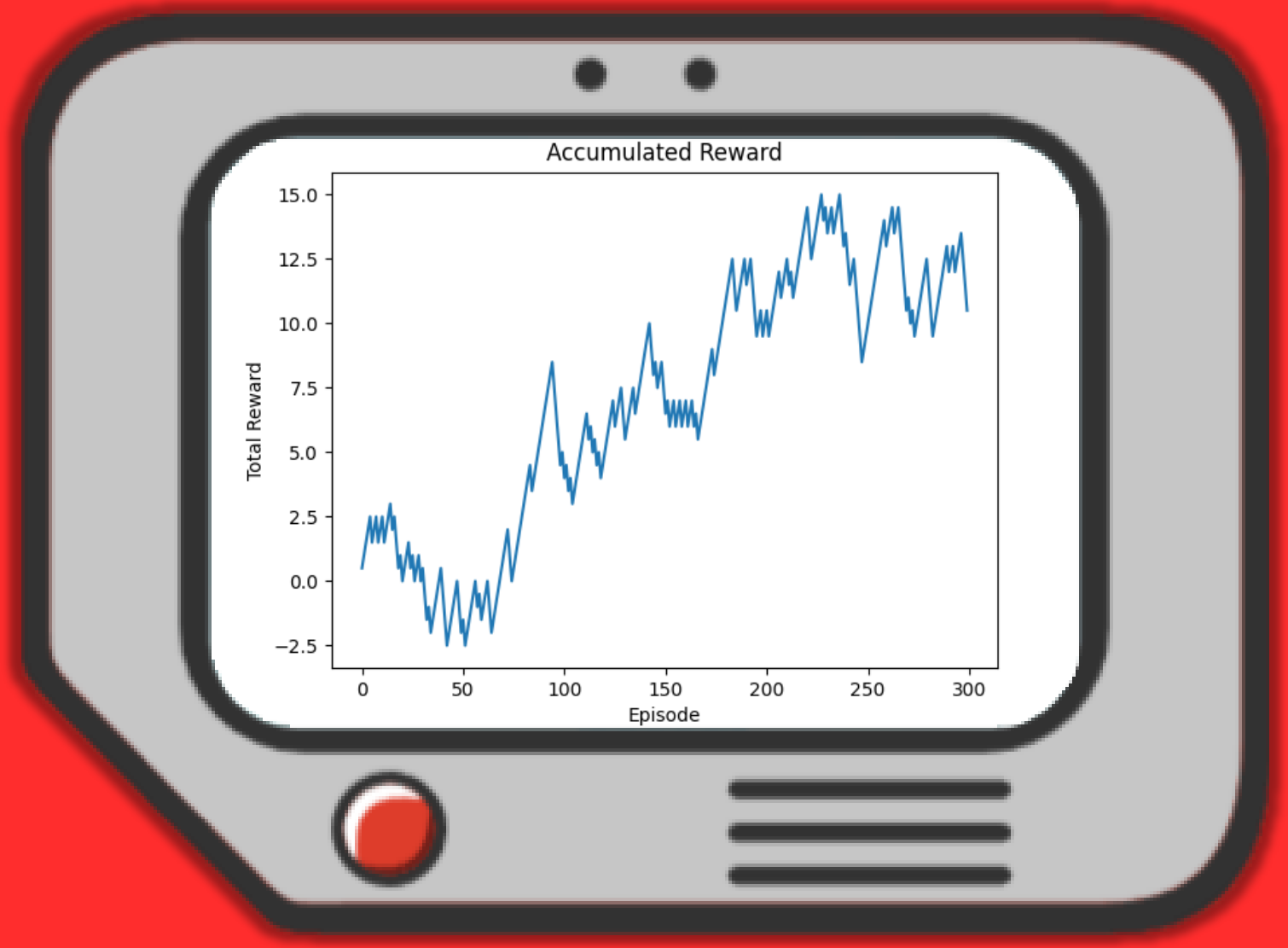
No! L'apprendimento per rinforzo è un processo **stocastico**: il comportamento dell'agente e le ricompense ottenute possono variare casualmente da un episodio all'altro.
Un modello prima di dare mosse buone deve **addestrarsi per moltissime epoche**!

Grafici rilevanti



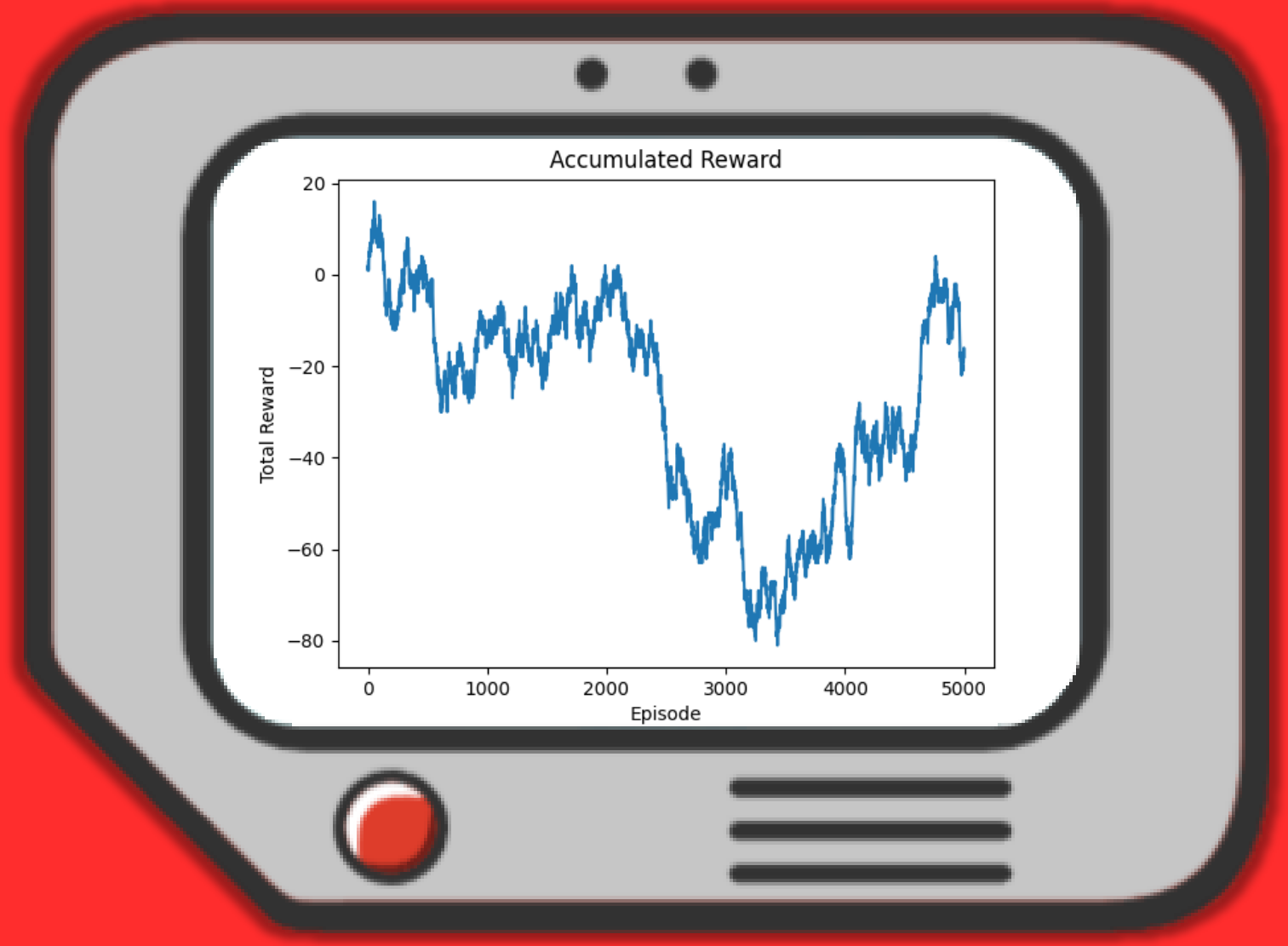
Questo grafico mostra perfettamente la natura stocastica del Q-Learning: dopo solo 300 epoche avevamo una distribuzione delle ricompense casuale e randomica.

Grafici rilevanti



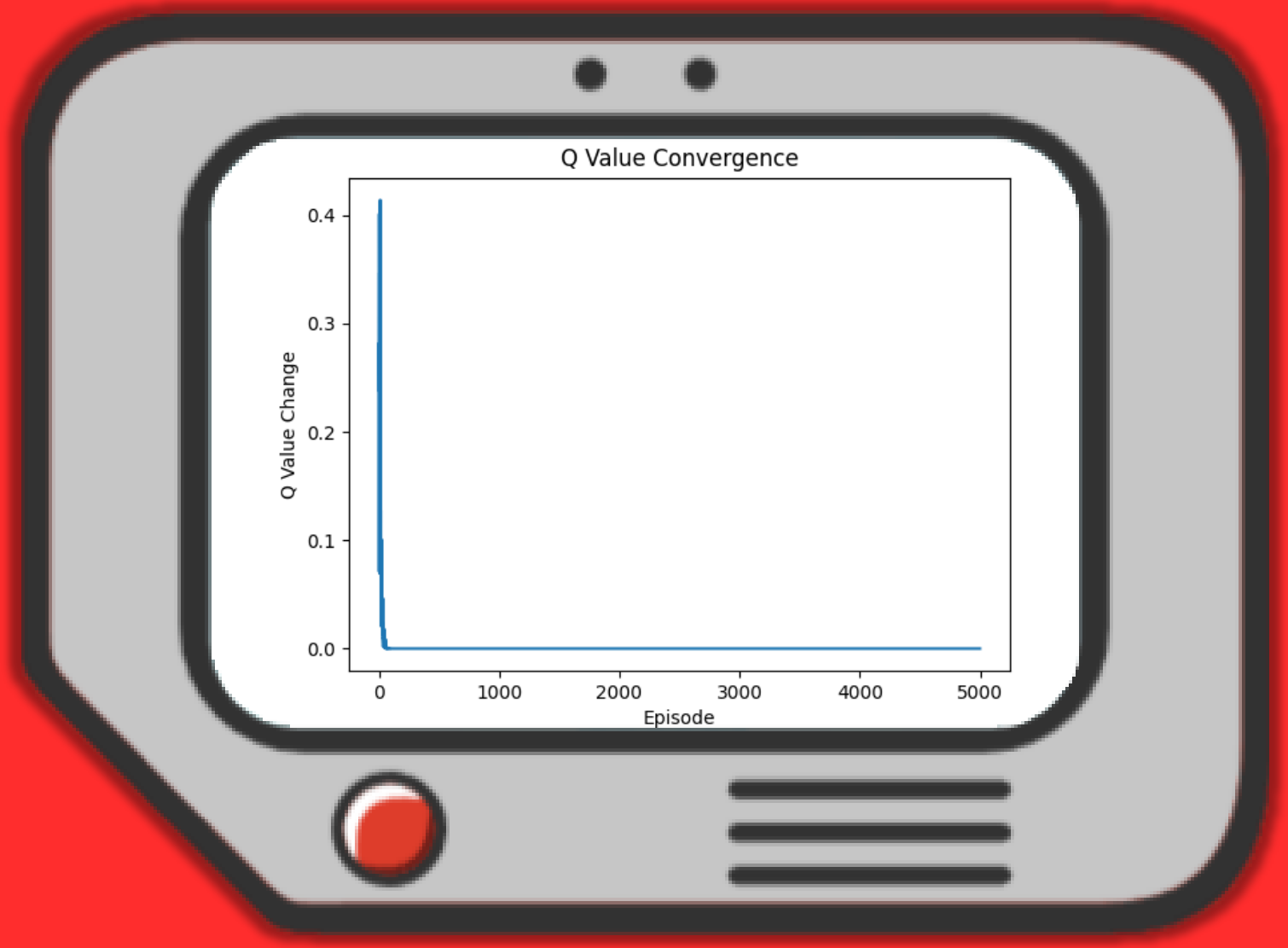
Dopo quasi 20 mila epoche
il grafico ha cominciato a stabilizzarsi.

Grafici rilevanti



Dopo quasi 20 mila epoche
il grafico ha cominciato a stabilizzarsi.

Grafici rilevanti

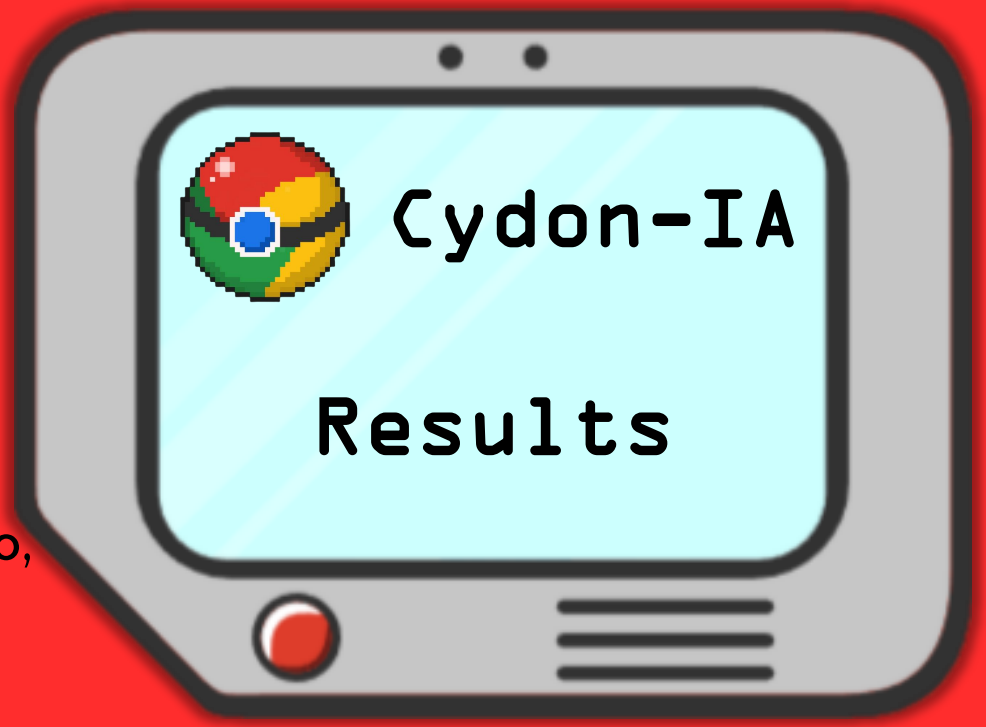


Dopo un bel po’ di epoche,
i valori hanno cominciato a convergere sullo 0.

Il modello è saturo e per alcuni valori già ha trovato i valori migliori.

Risultati

Date le valutazioni effettuate,
compresa la complessità del problema e la dinamicità stessa
che il problema ci pone, assimilato e definito l'approccio,
implementato il modello e valutato i risultati del modello stesso,
nonostante un tasso di precisione del 75%,
i risultati sono **ottimi ma migliorabili**.



Quindi è tutto da scartare?

No! L'apprendimento per rinforzo è un processo **stocastico**: il comportamento dell'agente e le ricompense ottenute possono variare casualmente da un episodio all'altro.
Un modello prima di dare mosse buone deve **addestrarsi per moltissime epoche**!

Cydon-IA

Le Nostre Conclusioni