UNICA
UNIVERSITÀ DEGLI STUDI
DI CAGLIARI

By Sanna Massimo, Luigi Mario Contu, Edoardo Lodo

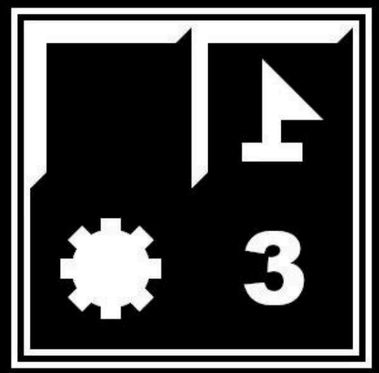
Reinforcement Learning– Prato Fiorito

Indice

Panoramica

Torna all'indice

Sviluppare un'intelligenza artificiale, allenata tramite tecnica di Apprendimento per Rinforzo, che sfrutta un ambiente creato da noi, capace di apprendere come giocare a Campo Minato.



Obbiettivi

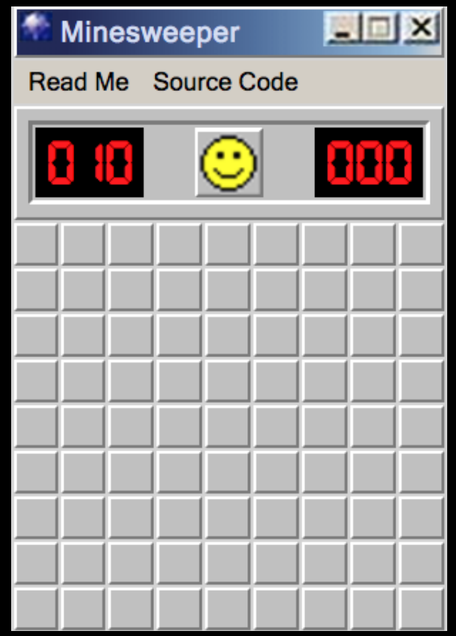
- Creare un ambiente fedele a prato fiorito;
- Ideare diversi modelli per un confronto ottimale;

Ambiente

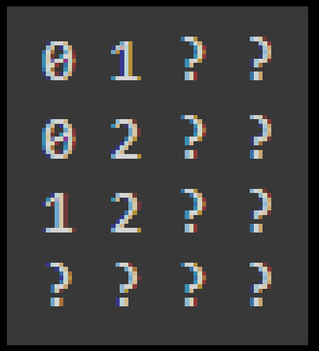
Torna all'indice

Abbiamo cominciato ricreando la struttura del gioco con diverse funzionalità di base, come la possibilità di regolare la difficoltà (numero di bombe nella griglia) e le dimensioni della griglia. Il gioco è stato successivamente integrato in Gym.

Obbiettivo



Realtà



I numeri indicano la quantità di bombe attorno alla cella. Abbiamo scelto di far fare all'ambiente il primo passo casuale, per evitare che il modello imparasse da un passo non contenente alcuna informazione utile al training.

Modelli e Tecniche

Torna all'indice

Policy Based

Una policy è una funzione che mappa gli stati alle azioni. La rete neurale prende come input lo stato del gioco (disposizione della griglia) e produce una distribuzione di probabilità sulle possibili azioni (celle da selezionare).
L’obiettivo del policy based è di massimizzare il rendimento atteso totale.
La rete viene addestrata utilizzando tecniche di gradient ascent per aumentare la probabilità di azioni che danno buoni risultati.
L’algoritmo scelto da noi è il PPO, poichè rende più stabile l’addestramento, permettendo di modificare il ratio clipping.
L'esplorazione è intrinseca nella natura stocastica della policy perchè seleziona le azioni in base a una distribuzione di probabilità.
Vantaggi del policy based:

- Spesso converge più velocemente a una soluzione ottimale;
- Non richiede un replay buffer di grandi dimensioni;
- Può facilmente apprendere politiche stocastiche.

Svantaggi:

- Rischio di convergenza in minimi locali;
- L’esplorazione è guidata dalla casualità, che potrebbe non esplorare sistematicamente tutti gli stati.

DQN (Deep Q-Network)

Il DQN è un'estensione del Q-learning che utilizza reti neurali profonde per approssimare la funzione Q, è un metodo value-based, cioè, cerca di stimare il valore di ogni azione possibile in ogni stato. L'agente sceglie l'azione con il valore Q più alto in un dato stato.
Abbiamo implementato una strategia ε-greedy, per incentivare l’esplorazione di nuove strategie, ovvero, far compiere azioni casuali all'agente. Il training tende a stabilizzarsi grazie alla memorizzazione di stati, azioni e ricompense, che vengono usate come punto di riferimento dall’IA stessa. Il replay buffer permette di campionare batch di esperienze casuali: questo è molto importante nel DQN in quanto aiuta a rompere le correlazioni temporali tra esperienze consecutive.

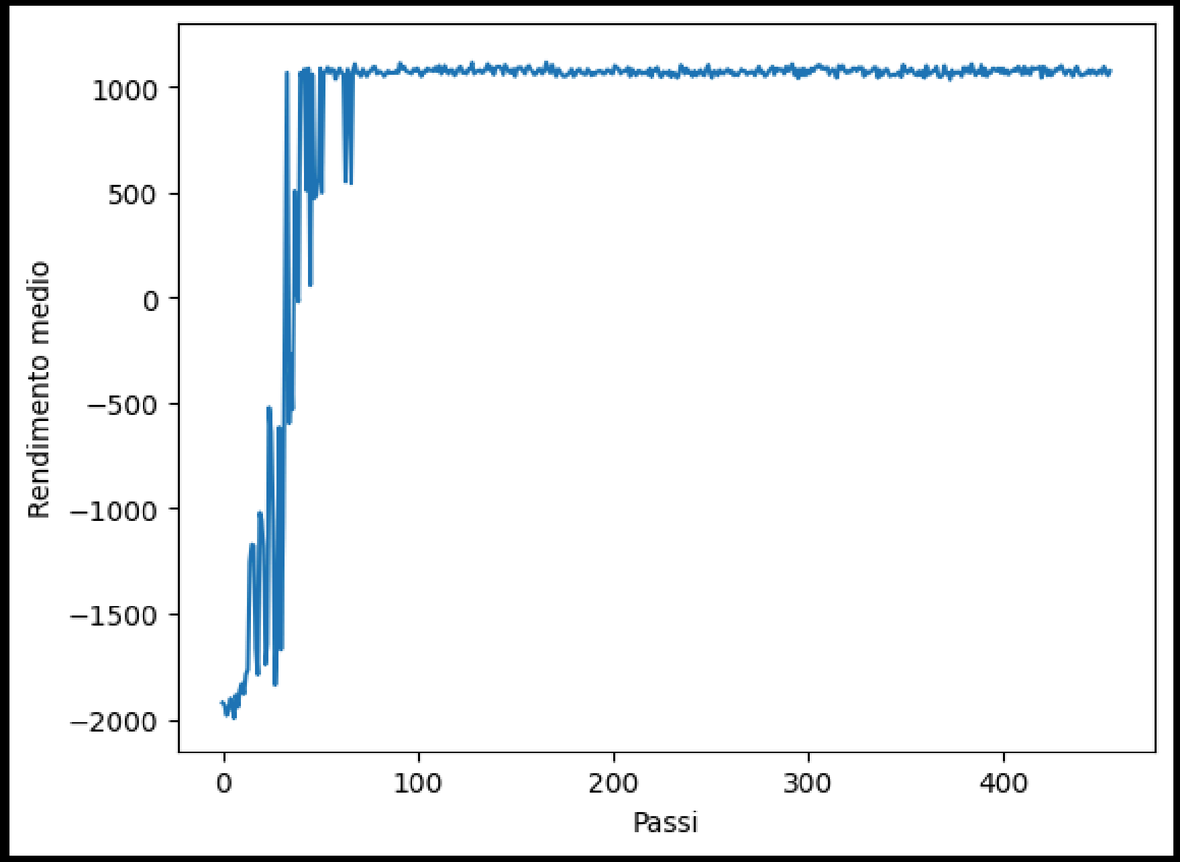
Modelli e Tecniche

Torna all'indice

Storico dello sviluppo

Per via dell'assenza di documentazione, abbiamo avuto molte difficoltà nell'approcio iniziale. Siamo partiti testando diversi modelli e algoritmi senza risultati. Data l'assenza di risultati abbiamo deciso di partire con un ambiente semplificato: una griglia di 0 e 1 in cui l'obbiettivo dell'agente era quello di cliccare le celle col valore 1. In questo periodo abbiamo provato ad applicare diverse techinche, ad esempio aggiungere una memoria al modello, dandole in input una griglia in più con indicate le celle già cliccate. Nonostante gli ottimi risultati però, questo input con doppia griglia non si addattava correttamente con prato fiorito, in quanto il modello avrebbe dovuto avere dei layer convoluti. Così dopo diverse settimane e molti test siamo riusciti ad ottenere un modello che riuscisse a cliccare correttamente le celle col valore 3 (in un range di valori da 0 a 3) senza l'utilizzo della memoria.

Di fianco il grafico del modello -->



Modelli e Tecniche

Torna all'indice

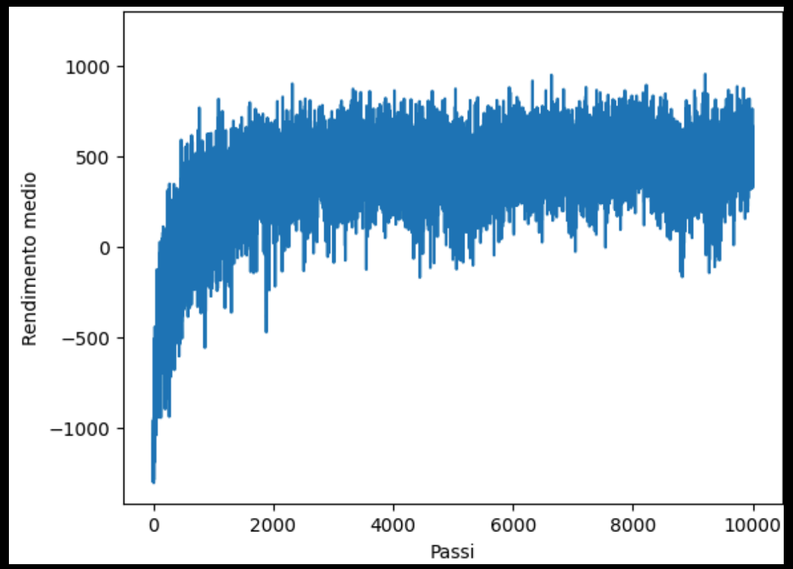
Storico dello sviluppo

Una volta che abbiamo ottenuto buoni risultati con un ambiente semplificato, abbiamo cercato di convertire il modello per l'ambiente di prato fiorito. I due ambienti erano più differenti di quanto ci aspettassimo infatti pure qui prima di avere dei risultati abbiamo fatto diverse settimane di training. La prima strategia utilizzata è stata quella del policy based. I parametri più importanti che hanno permesso al modello di funzionare sono stati il ratio_clipping, che ha reso il modello più stabile e il numero di step raccolti nel replay buffer. Meno problemi abbiamo avuto nel passare da policy based a DQN, la variabile più importante che abbiamo dovuto modificare è stata grandezza del replay buffer, in quanto non ne conoscevamo l'importanza.

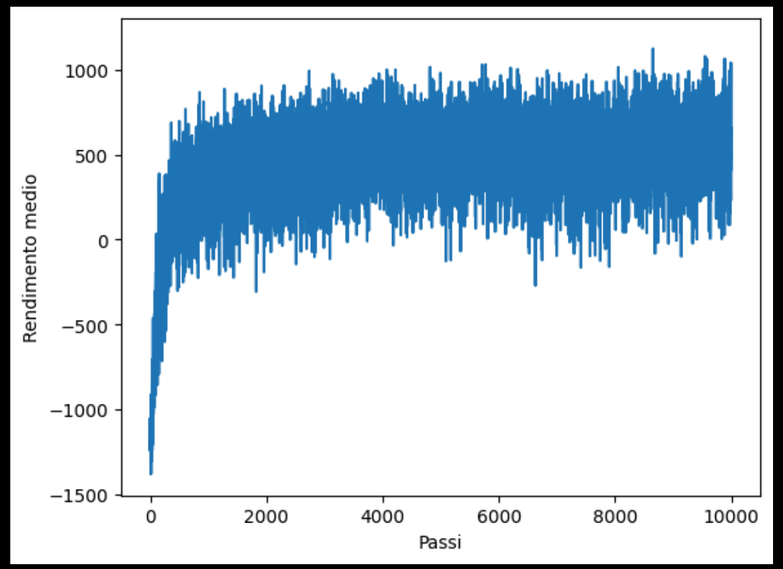
Analisi e Valutazione

Torna all'indice

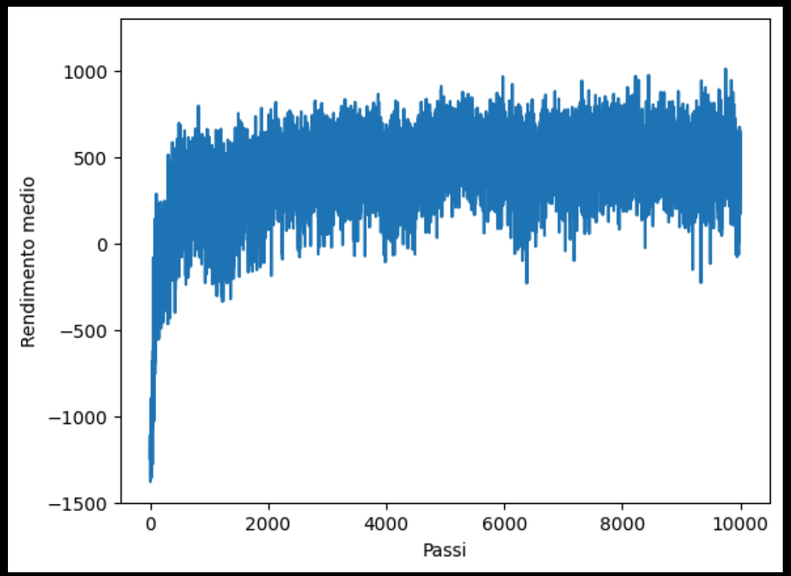
Policy Based



Dense



1 Conv

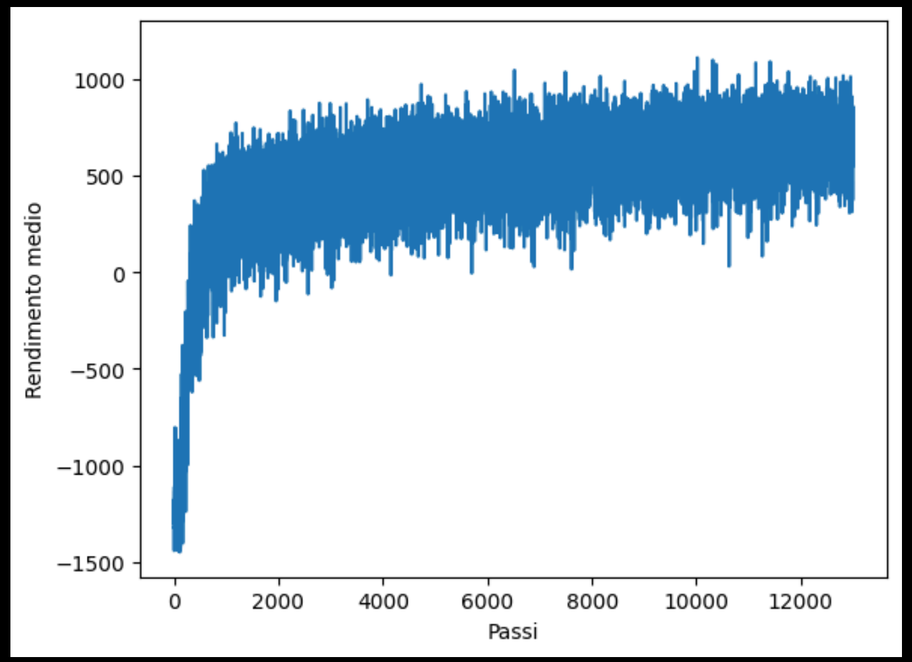


More Conv

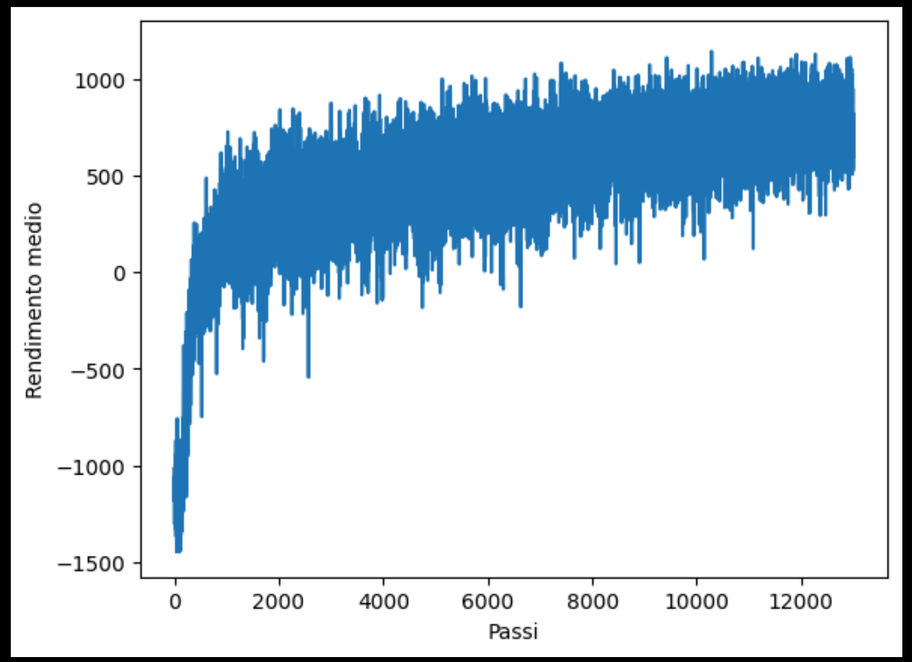
Analisi e Valutazione

Torna all'indice

DQN



Dense



1 Conv

Analisi e Valutazione

Torna all'indice

Win-Rate (media di 5mila episodi)

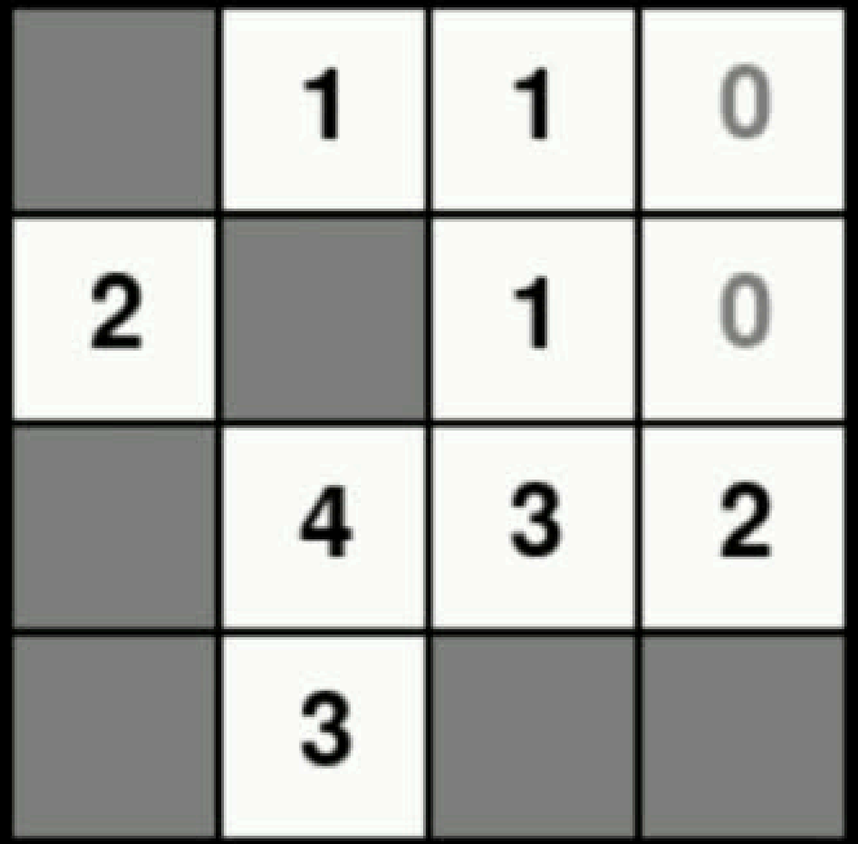
	Dense	1 Conv	M Conv
Policy-Based	0.52	0.65	0.49
Dqn	0.67	0.75	

Il modello migliore è il Dqn con 1 layer conv. Interessante notare come il modello con più layer conv policy-based performi peggio.

Applicazione

Torna all'indice

Qua abbiamo una dimostrazione video del nostro miglior modello trainato a lavoro, in funzione sull’ambiente con la difficoltà massima.



Futuro e Conclusione

Torna all'indice

Nonostante le iniziali difficoltà riscontrate, dopo un mese e mezzo di lavoro, siamo riusciti ad arrivare a dei risultati soddisfacenti, infatti il modello riesce a vincere circa l’80% delle partite in una difficoltà casuale. Per questioni di tempo ci siamo dovuti fermare con questa ottima baseline, ma continueremo lo sviluppo per provare a migliorare il modello con tecniche più avanzate. Una di queste è il curriculum learning, dove il modello viene addestrato progressivamente con un ambiente sempre più difficile. Un’altra nostra idea che vorremmo testare è quella di addestrare il modello solo su una griglia 3x3 e di conseguenza applicarlo a sole griglie di dimensioni più grandi, scomponendo la griglia in sotto griglie 3x3, poi passarle al modello per ottenere la probabilità delle azioni e, successivamente, il modello sceglierà la cella con la probabilità tra tutte più alta.