Apprendimento automatico in ambiente Pacman

January 5, 2026

Giulia Macarra

Il progetto, basato sul gioco pac-man, utilizza l'ambiente del Berkeley Pacman Project. Tale framework è una risorsa open-source implementata in Python 3.

1. Reinforcement Learning

Nel RL l'agente apprende, tramite l'interazione con l'ambiente, come mappare situazioni e azioni con l'obiettivo di massimizzare un reward cumulativo. Le azioni o le informazioni sull'ambiente non sono fornite a priori, e influenzano sia il reward immediato che la sottosequenza successiva. Un modello di RL richiede un equilibrio tra esplorazione ed exploitation. I problemi di RL possono essere formalizzati mediante un processo decisionale markoviano (MDP), che modella problemi di decisione sequenziale in ambienti stocastici. Un MDP è definito come una quintupla

$$(\mathcal{S}, \mathcal{A}, P, R, \gamma),$$

dove $\mathcal{S}$ rappresenta l'insieme degli stati, $\mathcal{A}$ l'insieme delle azioni, P la funzione di transizione di stato, R la funzione di ricompensa e $\gamma \in [0, 1)$ il fattore di sconto. L'obiettivo dell'agente è massimizzare il ritorno atteso G_t, definito come la somma scontata delle ricompense future. Poiché tale quantità è aleatoria, vengono introdotte le funzioni di valore, che forniscono una stima deterministica della qualità delle decisioni. Nel modello considerato, la funzione di riferimento è la funzione di valore azione–stato $Q(s, a)$, che rappresenta il ritorno atteso associato all'esecuzione dell'azione a nello stato s e al successivo comportamento dell'agente secondo una politica π.

$$Q^*(s,a) = \sum_{s'} P(s' \mid s,a)\Big[R(s,a,s') + \gamma \max_{a'} Q^*(s',a')\Big]$$

La funzione di valore ottimale Q^*, espressa in forma ricorsiva attraverso l'equazione di Bellman, può essere calcolata in modo iterativo.

$$Q^*(s,a) = \sum_{s'} P(s' \mid s,a)\Big[R(s,a,s') + \gamma \max_{a'} Q^*(s',a')\Big].$$

Tale formulazione si basa sulla proprietà di Markov; una volta nota Q^*, è possibile costruire una politica ottimale selezionando, in ogni stato, l'azione che ne massimizza il valore. La relazione di Bellman afferma che il valore associato all'esecuzione dell'azione a nello stato s è dato dalla somma della ricompensa immediata attesa e del valore futuro atteso, assumendo che l'agente segua la politica π.

2. pacman risolto con Q-learning tabellare

Il primo modello è basato sul Q-learning, un algoritmo di RL di tipo temporal-difference (TD) e off-policy, in cui la behaviour policy e la target policy sono distinte. L'algoritmo implementa un aggiornamento incrementale, confrontando il valore stimato $Q(s_t, a_t)$ con un target derivato dall'equazione di Bellman; la differenza tra questi due termini costituisce l'errore di differenza temporale (TD error), utilizzato per correggere progressivamente la stima corrente. Nel caso specifico di Pacman l'apprendimento è di tipo locale: confronta la stima presente $Q(s_t, a_t)$ con una stima futura a un passo, ottenuta combinando la ricompensa immediata osservata e il valore stimato dello stato successivo. La natura off-policy del Q-learning emerge dal fatto che l'agente può selezionare le azioni secondo una politica esplorativa ε-greedy; tuttavia, l'aggiornamento dei Q-values assume sempre un comportamento greedy ottimale nel passo successivo, indipendentemente dalla natura casuale dell'azione eseguita. L'utilizzo della politica ε-greedy consente di esplorare un numero sufficiente di coppie stato-azione, condizione necessaria per l'apprendimento di una politica greedy ottimale.

$$Q(s,a) \leftarrow Q(s,a) + \alpha\Big[r + \gamma \max_{a'} Q(s',a') - Q(s,a)\Big]$$

come descritto in (Sutton & Barto, 2018; Dusi, 2020).

statistiche e analisi dei risultati

Tipo di fantasma	Training WR	Avg steps	Q	Test WR	Avg score
Random Ghost	87.33%	134	552	86.67%	620.0
Directional Ghost	76.40%	124	558	88.00%	777.0

Table 1. Analisi quantitativa del primo modello

In una prima fase il modello è stato addestrato e testato esclusivamente sul labirinto small classic, sia con un fantasma di tipo random che directional. In entrambi i

Email: Giulia Macarra <macarra2095848@studenti.uniroma1.it>.

Machine Learning 2025, Sapienza University of Rome, 2nd semester a.y. 2024/2025.

casi il modello è risultato ottimo con prestazioni molto simili, mostrando un comportamento stabile nella fase di test. Si evidenzia un lieve miglioramento nei risultati nel caso con il fantasma directional; si potrebbe ipotizzare dovuto alla diversa strategia di movimento, considerando che quest'ultimo cerca di inseguire pacman, rendendo i suoi spostamenti facilmente interpretabili cosicché, l'agente possa adattarsi e muoversi di conseguenza. Per valutarne la stabilità e verificarne la robustezza e la capacità di generalizzazione, l'agente è stato testato anche su labirinti differenti. Sul labirinto medium classic si è riscontrato un win rate del 92% mentre sul labirinto original classic del 32%. Questo calo significativo è attribuibile alle differenze strutturali tra i labirinti: small classic, medium classic presentano una topologia simile, mentre original classic introduce una complessità che rende difficile all'agente adattarsi, seguendo la politica ottimale ottenuta su di un ambiente diverso. L'agente è stato esclusivamente testato sui diversi labirinti e non addestrato nuovamente, pertanto, la tabella dei Q-values è rimasta inalterata. Dal punto di vista statistico, al fine di valutare la stabilità del modello rispetto alla casualità, il test è stato ripetuto - su small classic - adottando diversi valori di seed (42, 128, 456). Le prestazioni finali registrano una variazione dell'ordine di circa il 2-4%, coerente con la natura stocastica del processo di apprendimento. Infine, il modello è stato addestrato e testato su di un ambiente più complesso, introducendo nel labirinto 2 fantasmi di tipo directional. Sia in fase di training che di test il win rate è crollato al 7%, rendendo evidente il limite intrinseco del Q-learning tabellare: l'aumento della complessità dell'ambiente rende il problema difficilmente gestibile con una rappresentazione discreta, poiché lo spazio degli stati non riesce a catturare la strategia ottimale da utilizzare.

3. pacman risolto con DQN

Le Deep Q-Networks (DQN) estendono il Q-learning tabellare introducendo l'uso di una rete neurale profonda, che approssima la funzione di valore azione–stato. La rete, parametrizzata da un insieme di pesi θ, restituisce una stima dei Q-values associati alle azioni disponibili per uno stato dato. In questo contesto, l'apprendimento non avviene più tramite l'enumerazione esplicita delle coppie stato–azione, ma attraverso la capacità di generalizzazione della rete neurale sullo spazio degli stati. Il secondo modello si basa su una Double DQN, introdotta per mitigare il fenomeno dell' *overestimation bias*. Tale struttura separa concettualmente la selezione dell'azione dalla sua valutazione, associandole a due reti neurali distinte: la policy network e la target network. La policy network, aggiornata tramite backpropagation, viene utilizzata per selezionare l'azione greedy nel prossimo stato, mentre la target network, aggiornata tramite un meccanismo di *soft update*, fornisce una stima più stabile del valore associato a tale azione.

$$y_t^{\text{DoubleDQN}} = r_{t+1} + \gamma\, Q\Big(s_{t+1}, \arg\max_{a'} Q(s_{t+1}, a'; \theta), \theta^-\Big)$$

Il target così ottenuto viene impiegato nella funzione di loss per aggiornare i parametri della policy network, mantenendo invariata la natura temporal-difference e off-policy dell'algoritmo. come descritto in (Mnih et al., 2015; Buzzoni, 2018)

statistiche e analisi dei risultati

È stato definito un agente, basato su Double DQN, addestrato e testato sul layout small classic sia in presenza di un fantasma di tipo random sia di tipo directional. Analogamente a quanto osservato nel modello tabellare, anche in questo caso il comportamento dell'agente risulta stabile e le prestazioni sono complessivamente elevate con entrambi i fantasmi. Le differenze emergono principalmente in fase di test, dove si osservano valori medi di punteggio più elevati e un numero inferiore di passi nel caso del fantasma directional, indicando una maggiore efficienza della politica appresa in presenza di una dinamica più strutturata dell'avversario.

Metrica	Random Ghost	Directional Ghost
Train – Win rate	84.00%	84.80%
Test – Win rate	93.33%	90.00%
Test – Avg score	860.1	1016.4
Test – Avg steps	167.9	131.4

Table 2. Analisi quantitativa del secondo modello (Double DQN)

Al fine di valutare la capacità di adattamento del modello, l'addestramento e il test sono stati ripetuti con un nuovo agente introducendo una politica, che includesse l'utilizzo dei power-up. In questo scenario si registra un ulteriore miglioramento delle prestazioni in fase di test, con un win rate che raggiunge il 94%, valori medi di score più elevati e una riduzione del numero medio di passi. Tali risultati suggeriscono che la Double DQN è in grado di integrare efficacemente informazioni aggiuntive nello stato e di sfruttarle per apprendere strategie più efficienti.

Metrica	Valore
Win rate	94.00% (47/50)
Avg score	1038.4
Avg steps	103.8
Avg capsules eaten	1.08
Avg ghosts eaten (scared)	1.04

Table 3. Risultati di test con 1 Directional Ghost e power-up (Double DQN)

References

Buzzoni, M. Reinforcement learning per giochi a informazione completa. Master's thesis, Università di Bologna, 2018.

Dusi, L. Deep reinforcement learning applied to game environments. Master's thesis, Università degli Studi di Padova, 2020.

Mnih, V., Kavukcuoglu, K., Silver, D., et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015.

Sutton, R. S. and Barto, A. G. *Reinforcement Learning: An Introduction*. MIT Press, 2 edition, 2018.