

Relazione progetto SOAR

Andrea Cacioli Matricola: 914501

Escape Game

In questo progetto é stato necessario utilizzare l'architettura cognitiva **SOAR** per modellare il comportamento di un agente rinchiuso in una stanza con l'obiettivo di scappare attraverso una finestra. Tale agente ha a disposizione diversi oggetti di cui però all'inizio non sa il funzionamento, tuttavia attraverso un meccanismo di **Reinforcement Learning** l'agente imparerà a fare le giuste combinazioni di oggetti al fine di creare una fionda, rompere la finestra e scappare.

Modellazione della conoscenza

Le assunzioni che sono state fatte (per non rendere troppo complicato il progetto modellando anche la fisica del mondo) sono le seguenti:

- Il robot può in ogni momento trovarsi in una di quattro posizioni: **nord, sud, est, ovest**.
- Il robot sa fin da subito dove si trova la **finestra** e dove si trovano i **tronchi**. (Si può immaginare che la stanza sia sufficientemente piccola da avere tutto a vista).
- Il robot sa che può **combinare** gli elementi che trova (molla, rametto, sassolini).
- Il robot sa che può sparare alla finestra in due modi diversi:
 - Ai bordi
 - Al centro
- Il robot **non** sa né quale sia la migliore combinazione di oggetti che gli permetterà di avere una fionda, né quale sia il punto giusto da sparare quando si spara alla finestra.
- Il robot sa qual é il movimento migliore da fare nelle varie situazioni.

Risoluzione Impasse

L'impasse é risolta in modi diversi a seconda di cosa sta succedendo intorno al robot.

Impasse per il movimento

- L'operatore che ti muove verso i tronchi é sempre il migliore quando si hanno i tronchi.
- L'operatore che ti muove verso la finestra é sempre il migliore quando si hanno i tronchi ma la finestra non é ancora rotta.

Impasse per la combinazione di oggetti

Qui la scelta dell'operatore all'inizio é casuale (numeric indifferent) dopodiché la scelta é presa grazie al reinforcement learning.

Impasse per lo sparo alla finestra

Qui la scelta dell'operatore all'inizio é casuale (numeric indifferent) dopodiché la scelta é presa grazie al reinforcement learning.

Ulteriori Operatori

Il robot dispone di altri due operatori: uno per costruire la torre e uno per arrampicarsi sulla torre e scappare.

Performance

In questa sezione verrà analizzato il numero di passi medio che impiega l'agente a imparare i rinforzi.

Valore di epsilon	Numero di Esecuzioni	Valore di Stabilità
default (0.1)	1	8
0.3	2	8
0.5	2	8
0.9	3	8

Osservazioni

Attenzione: Sotto il campo “Numero di Esecuzioni” é inteso come caso peggiore perché in realtà spesso se l'agente indovina subito l'operazione giusta, esso tende a continuare per quella strada anche con valori di epsilon più alti (forse perché ha anche un rinforzo negativo quando sbaglia).

Lavoro futuro

Sarebbe interessante vedere fino a dove si può spingere il meccanismo del RL provando a eliminare ogni tipo di conoscenza del mondo eccetto che per i meccanismi di RL per vedere come un agente complesso impara in maniera completamente behaviouristica.

Sarebbe interessante capire se tali “pesi” imparati possono essere utilizzati per task più complessi ed eventualmente passati di generazione in generazione aggiungendo delle mutazioni in stile algoritmo genetico. Tale sistema non é biologicamente ispirato poiché i rinforzi sono tipicamente parte del fenotipo ed in quanto tale non é passato geneticamente.