Intelligenza artificiale

Mattia Nicolis

A.A. 2025-26

Indice

Introduzione all'intelligenza artificiale

Il compito dell'**intelligenza artificiale** (IA) non è solo quello di *comprendere*, ma anche di *costruire* entità intelligenti: macchine in grado di calcolare come agire in modo efficace e sicuro in un'ampia varietà di situazioni nuove.

In passato, gli studiosi hanno proposto diverse versioni di IA: alcuni hanno definito l'intelligenza in termini di fedeltà alla **prestazione umana**, mentre altri preferiscono una definizione formale di intelligenza come **razionalità**.

I metodi utilizzati sono necessariamente diversi: l'approccio che persegue un'intelligenza simile a quella umana deve essere in parte una scienza empirica correlata alla psicologia, richiedendo osservazioni e ipotesi riguardo al comportamento umano e ai processi di pensiero.

Un approccio razionalista, invece, sfrutta una combinazione di matematica e ingegneria e si collega alla statistica, alla teoria del controllo e all'economia.

Test di Turing

Il **test di Turing**, proposto da Alan Turing nel 1950, mira a comprendere se una macchina è in grado di pensare.

Un computer supera il test se un esaminatore umano, dopo aver posto alcune domande in forma scritta, non è in grado di distinguere se le risposte provengano da una persona o da una macchina.

Per superare il test, il computer deve possedere le seguenti capacità:

- **interpretazione del linguaggio naturale**, per comunicare con successo nel linguaggio umano
- **rappresentazione della conoscenza**, per memorizzare ciò che conosce o apprende
- **ragionamento automatico**, per rispondere alle domande e trarre nuove conclusioni
- **apprendimento automatico** (*machine learning*), per adattarsi a nuove circostanze, individuare ed estrapolare schemi

Esiste, inoltre, un cosiddetto **test di Turing totale**, che richiede l'interazione con oggetti e persone nel mondo reale.

Per superare il test di Turing totale, un robot deve anche possedere:

- **visione artificiale** e riconosciemtno vocale, per percepire il mondo

- **robotica**, per manipolare gli oggetti e muoversi fisicamente

Tipologie di intelligenza artificiale

Agenti autonomi

Sono sistemi che percepiscono l'ambiente e agiscono in modo autonomo per raggiungere obiettivi specifici.

Data analysis

Consiste nell'utilizzo di algoritmi per analizzare grandi quantità di dati ed estrarre informazioni utili e correlazioni complesse.

Machine Learning

È lo sviluppo di algoritmi che permettono ai modelli di apprendere dai dati di esempio e migliorare le proprie prestazioni nel tempo, senza essere esplicitamente programmati.

Un esempio è il riconoscimento di immagini.

L'apprendimento automatico si divide in tre categorie principali:

- **unsupervised learning**: l modello viene addestrato su dati non etichettati, con l'obiettivo di scoprire strutture nascoste o pattern nei dati senza risposte predefinite.
- **supervised learning**: l modello viene addestrato su dati etichettati, dove ogni input è associato a una risposta corretta. L'obiettivo è imparare a mappare correttamente gli input alle risposte
- **reinforced learning**: il modello apprende attraverso interazioni con l'ambiente, ricevendo ricompense o penalità in base alle azioni compiute. L'obiettivo è massimizzare la ricompensa totale nel tempo

Time series analysis

L'analisi delle serie temporali è un'area dell'apprendimento automatico che si concentra sull'analisi di dati raccolti nel tempo.

Le serie temporali sono sequenze di dati misurati a intervalli regolari, come la temperatura giornaliera, i prezzi delle azioni o i dati di vendita mensili.

L'obiettivo è identificare pattern, tendenze e stagionalità per effettuare previsioni future.

Gli approcci comuni includono:

- **riconoscimento di anomalie e cause**: identificazione di eventi che si discostano dal comportamento normale, potenzialmente indicativi di errori o problemi.
- **generative transformers**: modelli in grado di predire il prossimo elemento in una sequenza di dati (ad esempio la parola successiva in una frase o il pixel successivo in

un'immagine), utilizzando il concetto di **attenzione** per pesare l'importanza delle diverse parti della sequenza di input.

Agenti intelligenti

Un agente intelligente è un sistema che percepisce l'ambiente circostante attraverso sensori e agisce su di esso per raggiungere un obiettivo specifico.

Gli elementi chiave di un agente intelligente sono:

- **performance measure**: misura il successo dell'agente nel raggiungere i propri obiettivi
- **rationality**: l'agente deve agire in modo da massimizzare la performance attesa

Markov Decision Process (MDP)

Un MDP è un modello matematico utilizzato per rappresentare problemi di decisione sequenziali. I suoi elementi principali sono:

- **state**: rappresenta l'ambiente in un determinato momento
- **actions**: nsieme delle azioni che l'agente può intraprendere
- **transition model**: descrive l'effetto delle azioni sull'ambiente (può essere parzialmente incognito)

$$T : (state, action) \rightarrow next\_state$$

- **reward**: vvalore immediato associato all'esecuzione di un'azione

$$R : (state, action, next\_state) \rightarrow real\_number$$

- **Policy**: strategia con cui l'agente decide quale azione intraprendere in ogni stato per massimizzare la ricompensa totale attesa nel tempo

$$\pi : (state) \rightarrow action$$

Generative AI

L'intelligenza artificiale generativa comprende una classe di modelli in grado di creare nuovi contenuti — testo, immagini, musica o video — a partire da dati di addestramento.

Questi modelli possiedono miliardi di parametri e sono pre-addestrati su enormi quantità di dati.

In sostanza, "predicono il futuro" basandosi sui dati su cui sono stati **addestrati** e su un **prompt** (input dell'utente).

Agenti intelligenti

Un **agente** è qualsiasi entità che possa essere vista come un sistema che percepisce il proprio ambiente attraverso dei **sensori** e agisce su di esso mediante **attuatori**.

Un agente umano possiede come **sensori** gli occhi e altri organi, e può utilizzare come **attuatori** mani, gambe, corde vocali ecc.

Un agente robotico può invece disporre di telecamere e telemetri a infrarossi come sensori, e di diversi motori come attuatori.

Definizione formale di agente razionale

Per ogni possibile sequenza di percezioni, un agente razionale dovrebbe scegliere un'azione che massimizzi il valore atteso della sua misura di performance, date le informazioni fornite dalla sequenza e da ogni ulteriore conoscenza posseduta dall'agente.

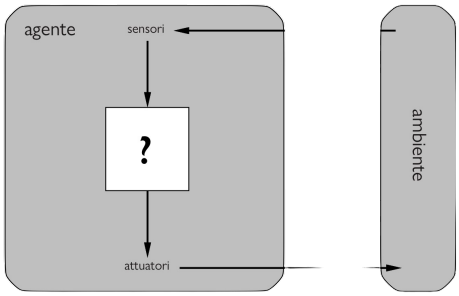

Figura 1: agente razionale

Useremo il termine **percezione** per indicare i dati che i sensori di un agente percepiscono.

La **seuquenza percettiva** di un agente è la storia completa di tutto ciò che esso ha percepito nella sua esistenza.

Sèecificando l'azione prescelta dell'agente per ogni possibile sequenza percettiva, abbiamo descritto l'agente in modo completo.

In termini matematici diciamo, quindi, che il comportamento di un agente è descritto dalla **funzione agente**, che descrive la corrispondenza tra una qualsiasi sequenza percettiva e una specifica azione:

$$f : \mathcal{P}^* \rightarrow \mathcal{A}$$

Possiamo immaginire di rappresentare in forma di tabella la funzione agente che descrive un certo agente.

Nella maggior parte dei casi questa tabella sarebbe molto grande, di fatto infinito, a meno di non specificare una lunghezza massima delle sequenza percettive che vogliamo prendere in considerazione.

La tabella è una descrizione esterna dell'agente.

Internamente, la funzione agente di un agente artificale sarà implementata da un **programma agente**.

E' importante tenere distinti questi due concetti:

- *funzione agente*: descrizione matematica stratta
- *programma agente*: implemenyazione concreta della funzione agente, in esecuzione all'interno di un sistema fisico.

Strategie di ricerca non informata

Un **algoritmo di ricerca non informata** non riceve alcuna informazione su quanto uno stato sia vicino all'obiettivo.

Es: consideriamo una persona che si trova ad Arad con l'obiettivo di raggiungere Burarest.

Una persona non informata, priva di conoscenza della geografia romena, non ha modo di sapre se sia meglio come primo passo andare a Zerind o Sibiu.

Ricerca in ampiezza

Quando tutte le azioni hanno lo stesso costo, una strategia appropriata è la **ricerca in ampiezza**, in cui si espande prima il nodo radice, poi tutti i suoi sucessori, poi i sucessori di quest'ultimi, e così via.

Potremmo implementare la ricerca in ampiezza, chiamata **RICERCA-BEST-FIRST**, con la funzione di valutezione $f(n)$ è uguale alla profondità del nodo, cioè al numero di azioni necessario per ragiungerlo.

Una coda FIFO (First In First Out) risulterà più veloce di una coda con priorità e ci fornirà l'ordine corretto dei nodi: i nuovi nodi vanno in fondo alla coda e quelli vecchi, a profondità minore, vengono espansi per primi.

Inoltre, raggiunti può essere un insieme di stati, anzichè una corrispondeza da stati a nodi, perchè una volta raggiunto uno stato, non possiamo più trovare un cammino migliore per raggiungerlo.

LA ricerca in amppiezza trova sempre una soluzione con un numero minimo di azioni, perchè quando genre nodi di profondità d, ha già generato tutti i nodi di profondità $d-1$, perciò se uno di essi fosse una soluzione, sarebbe stato stato trovato.

Ciò significa che è ottimale rispetto al costo per problemi in cui tutte le azioni hanno lo stesso costo, ma non per problemi che non hanno tale proprietà.

E' completa in ogni caso.

In termini di tempo di tempo e spazio, supponiamo di effettuare una ricerca in un albero uniforme dove ogni stato ha b sucessori.

Allora il numer totale di nodi generati è:

$$1 + b + b^2 + b^3 + \cdots + b^d = O(b^d)$$

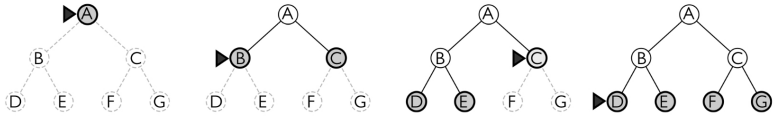

Figura 2: algoritmo BFS

```
function BFS(problem) returns a solution, or failure
  nodo <- nodo(problema.statoIniziale)
```

Tutti i nodi rimangono in memoria, perciò le complessità temporale e spaziale sono entrambe $O(b^d)$, quindi, esponenziali.

Nella ricerca in ampiezza, i requisiti di memoria rappresentano un problrma più importante rispetto al tempo di esucuzione.

Tuttavia, il tempo rimane comunque un fattore importante.

In generale, i problemi di ricerca con complessità esponenziale non possono essere risolti mediante ricercche non informate, se non nelle istante più pisccole.

Ricerca a costo minimo (algoritmo Dijkstra)

Quando le azioni hanno costi diversi, si utilizza BFS in cui la funzione di valutazione è il costo del cammino della radice al nodo corrente: si tratta dell'**algoritmo di Dijkstra** o **ricerca a costo uniforme**.