



Intelligenza artificiale



WIKIBOOKS
Libri liberi per un mondo aperto

INTELLIGENZA ARTIFICIALE

it.wikibooks.org

2020

Questo testo proviene dal sito
https://it.wikibooks.org/wiki/Intelligenza_artificiale
ed è stato scritto collettivamente dagli utenti di tale sito

Questo e-book è aggiornato al
23 aprile 2020

In copertina:
Coppia di robot NAO. Immagine rilasciata con licenza Creative Commons
Attribuzione-Condividi allo stesso modo 4.0 International [https://commons.wikimedia.org/wiki/File:NAO_Robot_\(bleu_et_rouge\)_\(cropped\).jpg](https://commons.wikimedia.org/wiki/File:NAO_Robot_(bleu_et_rouge)_(cropped).jpg)

Wikibooks non dà garanzie sulla validità dei suoi contenuti. Per i dettagli vedi:
https://it.wikibooks.org/wiki/Wikibooks:General_disclaimer

Quest'opera è soggetta alla licenza Creative Commons Attribuzione-Condividi allo stesso modo 3.0 Unported. Per leggere una copia della licenza visita il sito:
<https://creativecommons.org/licenses/by-sa/3.0/deed.it>

Indice

1	Definizione	1
2	Storia	3
2.1	I primi passi	3
2.2	La rivoluzione informatica	4
2.3	Il punto di svolta	4
2.4	Verso l'IA moderna	5
2.5	IA moderna	5
3	Filosofia	7
3.1	Test di Turing	9
3.2	Stanza cinese	10
4	Rappresentazione della conoscenza	17
4.1	Linguaggi formali	17
4.2	Alberi di decisione	18
5	Apprendimento automatico	21
5.1	Apprendimento non supervisionato	22
5.2	Apprendimento supervisionato	24
5.3	Apprendimento con rinforzo	26
6	Logica Fuzzy	41
7	Problema di soddisfacimento di vincoli	43
7.1	Algoritmi	44
8	Applicazioni	45
	Fonti	47

Nota

Questo libro intende trattare gli aspetti riguardanti l'Intelligenza artificiale sia dal punto di vista teorico che dal punto di vista pratico. Il libro non approfondisce i singoli argomenti lasciando ad altri testi gli opportuni approfondimenti, ma punta a realizzare una panoramica della materia. Essendo la materia estremamente estesa il libro tratta solo i temi ritenuti classici dell'intelligenza artificiale quindi prevalentemente l'apprendimento automatico tralasciando molte tecniche e paradigmi legati alla materia.

Definizione

Con il termine intelligenza artificiale (spesso abbreviato in AI, dall'inglese Artificial Intelligence), si intende generalmente l'abilità di un computer di svolgere funzioni e ragionamenti tipici della mente umana.

Esistono almeno quattro diversi punti di vista sull'intelligenza artificiale, quattro modi diversi di intendere l'intelligenza artificiale. Questi approcci sono:

- Pensare come un essere umano - Approccio del test di Turing
- Agire come un essere umano - Approccio della simulazione
- Pensare razionalmente - Approccio della logica
- Agire razionalmente - Approccio degli agenti razionali

L'intelligenza artificiale è una disciplina dibattuta tra scienziati e filosofi, la quale manifesta aspetti sia teorici che pratici.

Nel suo aspetto puramente informatico, essa comprende la teoria e le tecniche per lo sviluppo di algoritmi che consentano alle macchine (tipicamente ai calcolatori) di mostrare un'abilità e/o attività intelligente, almeno in domini specifici. Per attività intelligente si intende la capacità di estrapolare da conoscenze precedenti delle linee guida da utilizzare per risolvere nuovi problemi, problemi che il sistema non ha mai affrontato sebbene possa averne affrontati di simili in passato.

Storia

L'espressione "Intelligenza Artificiale" (Artificial Intelligence) è stata coniata nel 1956 dal matematico americano John McCarthy, in seguito ad uno storico seminario interdisciplinare svoltosi nel New Hampshire. Secondo le parole di Marvin Minsky, uno dei "pionieri" della I.A., lo scopo di questa nuova disciplina sarebbe stato quello di "far fare alle macchine delle cose che richiederebbero l'intelligenza se fossero fatte dagli uomini".

2.1 I primi passi

In tal senso la storia ha inizio nel XVII sec. quando Blaise Pascal (scienziato, scrittore e filosofo francese) inventa la cosiddetta "Pascalina" per aiutare il padre, incaricato dall'amministrazione fiscale della Normandia di eseguire un difficile lavoro di calcolo. La macchina era capace di eseguire automaticamente addizione e sottrazione; questa "macchina aritmetica" fu la capostipite dei calcolatori ad ingranaggi.

In età vittoriana Charles Babbage creò macchine calcolatrici a rotelle: "The difference engine" (la macchina alle differenze) riusciva a fare calcoli differenziali ed arrivò a progettarne una intelligente che però, per problemi tecnici, non riuscì mai a funzionare; avrebbe dovuto essere programmata con schede perforate, un po' come accadde in seguito con i primi calcolatori elettronici. Le schede perforate, cartoncini forati a secondo della necessità, furono ampiamente usati per esempio per il funzionamento dei telai Jacquard, di pianole meccaniche, quindi dei primi calcolatori. Hermann Hollerith (statistico U.S.A. di origine tedesca) ideò le schede perforate applicate a calcolatrici attorno al 1885; questo sistema fu usato per la prima volta per i calcoli relativi all'11° censimento U.S.A., nel 1891. Il sistema meccanografico adottato da Hollerith riscosse tale successo da indurlo a fondare la Tabulating Machine corporation. In seguito i calcolatori furono molto usati dai militari per regolare il tiro della loro artiglieria.

2.2 La rivoluzione informatica

Se la teoria dell'IA evolve indipendentemente dai progressi scientifici, le sue applicazioni sono fortemente legate agli avanzamenti della tecnologia informatica. Infatti, solo nella seconda metà del XX secolo è possibile disporre di dispositivi di calcolo e linguaggi di programmazione abbastanza potenti da permettere sperimentazioni sull'intelligenza.

La struttura dei calcolatori viene stravolta con la sostituzione dei "relè", usati per i primi calcolatori elettromeccanici, con le "valvole" o tubi elettronici. Nel 1946 nasce ENIAC (Electronic Numerical Integrator And Calculator), concepito come calcolatore moderno nel '45 da John von Neumann; faceva l'elaborazione a lotti (batch) nell'ordine di migliaia di informazioni al minuto. La programmazione avveniva comunque tramite schede.

La seconda generazione di computer si ha nei '60, "The time sharing", sistemi basati sulla divisione di tempo e quindi più veloci; più terminali, soprattutto telescriventi, sono collegati ad un calcolatore centrale. L'innovazione in questo periodo sta nel passaggio dalle valvole ai transistor. A quell'epoca i programmi erano fortemente condizionati dai limiti dei linguaggi di programmazione, oltre che dai limiti di velocità e memoria degli elaboratori. Il punto di svolta si ha proprio tra gli anni '50 e '60, con linguaggi di manipolazione simbolica tali l'ILP, il LISP e il POP. Nei primi anni '70 inizia la microelettronica su circuiti stampati. Di questo periodo i Very large scale integration (VLSI), i primi minicomputer, i PDP 11, più piccoli, collegabili a distanza, con un costo che all'epoca si aggirava sui 100 milioni di vecchie lire. Si inizia a parlare di "informatica distribuita", tramite cavi telefonici, sia per le telescriventi ma soprattutto per i computer intelligenti, essendo l'elaborazione indipendente dai grossi computer centrali.

2.3 Il punto di svolta

Uno dei punti di svolta della materia si ha con un famoso articolo di Alan Turing sulla rivista *Mind* nel 1950. Nell'articolo viene indicata la possibilità di creare un programma al fine di far comportare un computer in maniera intelligente. Quindi la progettazione di macchine intelligenti dipende fortemente dalle possibilità di rappresentazione simbolica del problema.

Nel 1952 Arthur Samuel presenta il primo programma capace di giocare a Dama, un risultato molto importante perché dimostra la possibilità di superare i limiti tecnici (il programma era scritto in Assembly e girava su un IBM 701) per realizzare sistemi capaci di risolvere problemi tradizionalmente legati all'intelligenza umana. Per di più, l'abilità di gioco viene appresa dal programma scontrandosi con avversari umani. Nel 1956, alla conferenza di Dartmouth (la stessa conferenza a cui l'IA deve il suo nome), viene mostrato un programma che segna un'altra importante tappa dello sviluppo dell'IA. Il

programma LT di Allen Newell, J. Clifford Shaw e Herb Simon rappresenta il primo dimostratore automatico di teoremi.

La linea seguita dalla giovane IA si basa quindi sulla ricerca di un automatismo nella creazione di un'intelligenza meccanica. L'approccio segue essenzialmente un'euristica di ricerca basata su tentativi ed errori oltre che investigare su tecniche di apprendimento efficaci.

2.4 Verso l'IA moderna

Secondo le parole di Minsky, dopo il 1962 l'IA cambia le sue priorità: viene data minore importanza all'apprendimento, mentre si pone l'accento sulla rappresentazione della conoscenza e sul problema ad essa connesso del superamento del formalismo finora a disposizione e liberarsi dalle costrizioni dei vecchi sistemi.

"Il problema della ricerca efficace con euristiche rimane un presupposto soggiacente, ma non è più il problema a quale pensare, per quanto siamo immersi in sotto-problemi più sofisticati, ossia la rappresentazione e modifica di piani" (Minsky, 1968).

I punti cardine di questa ricerca sono gli studi di Minsky sulla rappresentazione distribuita della conoscenza, quella che viene chiamata la "società delle menti", e il lavoro di John McCarthy sulla rappresentazione dichiarativa della conoscenza. Quest'ultima viene espressa formalmente mediante estensioni della logica dei predicati e può quindi essere manipolata facilmente. Con i suoi studi sul "ragionamento non monotono" e "di default", McCarty contribuisce a porre gran parte delle basi teoriche dell'IA.

Il punto di vista psicologico non viene assolutamente trascurato. Ad esempio il programma EPAM (Feigenbaum e Feldman, 1963) esplora la relazione tra memoria associativa e l'atto di dimenticare. Alla Carnegie Mellon University vengono sperimentati programmi per riprodurre i passi del ragionamento, inclusi eventuali errori.

L'elaborazione del linguaggio naturale sembra essere un campo destinato a un rapido sviluppo. La traduzione diretta di testi porta però a insuccessi iniziali che influenzeranno per molti anni l'attribuzione di fonti a tale campo. Malgrado questo, viene dimostrato abbastanza presto che si possono ottenere buoni risultati in contesti limitati.

2.5 IA moderna

Durante gli anni novanta vi fu una progressiva espansione del settore legato all'intelligenza artificiale. L'avvento di personal computer sempre più potenti permise di implementare algoritmi di intelligenza artificiale senza dover utilizzare hardware dedicato, inoltre approfonditi studi su sistemi come gli automi cellulari e le reti neurali permisero di ottenere risultati pratici interessanti

in settori dove lo sviluppo di algoritmi deterministici è molto difficoltoso se non impossibile, poiché praticamente tutti i sistemi di riconoscimento della scrittura comprendono un modulo a rete neurale più o meno sviluppato.

Filosofia

Le attività e le capacità dell'Intelligenza Artificiale comprendono:

- l'apprendimento automatico (machine learning), utile in contesti quale il gioco degli scacchi
- la rappresentazione della conoscenza e il ragionamento automatico in maniera simile a quanto fatto dalla mente umana
- la pianificazione (planning)
- la cooperazione tra agenti intelligenti, sia software che hardware (robot)
- l'elaborazione del linguaggio naturale (Natural Language Processing)
- la simulazione della visione e dell'interpretazione di immagini, come nel caso dell'OCR.

Queste attività se svolte da un uomo implicano l'utilizzo di un'intelligenza e di capacità di raziocinio. Questo porta naturalmente a formulare la domanda che sta al centro di tutti i dibattiti legati all'intelligenza artificiale, cioè: "I computer possono pensare?".

Le risposte sono varie e discordi, ma perché abbiano un senso bisogna prima determinare cosa significhi pensare. Ironicamente, nonostante tutti siano d'accordo che gli esseri umani sono intelligenti, nessuno è ancora riuscito a dare una definizione soddisfacente di intelligenza; proprio a causa di ciò, lo studio dell'AI si divide in due correnti:

la prima, detta intelligenza artificiale forte, sostenuta dai funzionalisti, ritiene che un computer correttamente programmato possa essere veramente dotato di una intelligenza pura, non distinguibile in nessun senso importante dall'intelligenza umana. L'idea alla base di questa teoria è il concetto che risale al filosofo empirista inglese Thomas Hobbes, il quale sosteneva che ragionare

non è nient'altro che calcolare: la mente umana sarebbe dunque il prodotto di un complesso insieme di calcoli eseguiti dal cervello;

la seconda, detta intelligenza artificiale debole, sostiene che un computer non sarà mai in grado di eguagliare la mente umana, ma potrà solo arrivare a simulare alcuni processi cognitivi umani senza riuscire a riprodurli nella loro totale complessità.

Rimanendo nel campo della programmazione "classica", basata su linguaggi simbolici e lineari, in cui la grande velocità di calcolo dei processori moderni supplisce alla carenza di parallelismo, sicuramente assume una posizione dominante l'AI debole, in quanto si può facilmente constatare come un computer elabori una serie di simboli che non comprende e che si limiti ad eseguire i suoi compiti meccanicamente.

Bisogna tuttavia riconoscere che, con la diffusione sempre maggiore di reti neurali, algoritmi genetici e sistemi per il calcolo parallelo, la situazione si sta evolvendo a favore dei sostenitori del connessionismo.

A detta di alcuni esperti del settore, però, è improbabile il raggiungimento, da parte di un computer, di una capacità di pensiero classificabile come "intelligenza", in quanto la macchina stessa è "isolata" dal mondo, o, al massimo, collegata con esso tramite una rete informatica, in grado di trasmettergli solo informazioni provenienti da altri computer. La vera "intelligenza artificiale", perciò, potrebbe essere raggiungibile solo da robot (non necessariamente di forma umanoide) in grado di muoversi (su ruote, gambe, cingoli o quant'altro) ed interagire con l'ambiente che li circonda grazie a sensori ed a bracci meccanici. Spesso, difatti, anche nell'uomo, l'applicazione dell'intelligenza deriva da qualche esigenza corporea, perciò è improbabile riuscire a svilupparne un'imitazione senza un corpo.

Inoltre, finora, nel tentativo di creare AI, si è spesso compiuto un errore che ha portato i computer all'incapacità di applicare il buonsenso e alla tendenza a "cacciarsi nei pasticci". L'errore consiste nel non considerare a sufficienza il fatto che il mondo reale è complesso e quindi una sua rappresentazione lo sarà altrettanto. Non solo sarà complessa, ma sarà anche incompleta, perché non potrà mai includere tutti i casi che il robot potrà incontrare. Perciò, o immettiamo nel cervello artificiale una quantità enorme di informazioni corredate da altrettante regole per correlarle (il che originerà, probabilmente, un vicolo cieco logico alla prima difficoltà incontrata), oppure lo mettiamo in condizione di imparare. La chiave dell'AI, sembra proprio essere questa: l'imitazione della sua analogia naturale, tenendo ben presente l'importanza dei processi evolutivi nello sviluppo delle caratteristiche morfologiche e comportamentali di un individuo e nella formazione di ciò che viene definito "senso comune".

Il dibattito sull'intelligenza artificiale debole e forte ha prodotto moltissima letteratura. Tra i saggi libri e articoli spiccano due test, il Test di Turing (\rightarrow 3.1) e la stanza cinese (\rightarrow 3.2). Il primo test perora l'intelligenza artificiale forte mentre il secondo punta all'intelligenza artificiale debole.

3.1 Test di Turing

Il Test di Turing è un criterio, introdotto dal matematico Alan Turing nell'articolo *Computing machinery and intelligence*, apparso nel 1950, per determinare se una macchina sia in grado di pensare.

Il test consiste in un gioco, noto come gioco dell'imitazione, a tre partecipanti: un uomo, A, una donna, B, e una terza persona C. Quest'ultimo è tenuto separato dagli altri due e tramite una serie di domande deve stabilire qual è l'uomo e quale la donna. Dal canto loro anche A e B hanno dei compiti: A deve ingannare C e portarlo a fare un'identificazione errata, mentre B deve aiutarlo. Affinché C non possa disporre di alcun indizio (come l'analisi della calligrafia o della voce), le risposte alle domande di C devono essere dattiloscritte o similmente trasmesse.

Il test di Turing si basa sul presupposto che una macchina si sostituisca ad A. In tal caso, se C non si accorgesse di nulla, la macchina dovrebbe essere considerata intelligente, in quanto -in questa situazione- è stata in grado di comportarsi come un uomo.

Per macchina intelligente Turing ne intende una in grado di pensare, e il ragionamento alla base del test è fondato sull'idea di pensiero come concatenazione di idee e capacità di esprimerle. Per Turing, quindi, tutto si limita alla produzione di espressioni non prive di significato. Nell'articolo si legge: secondo la forma più estrema di questa opinione, il solo modo in cui si potrebbe essere sicuri che una macchina pensa è quello di essere la macchina stessa e sentire se si stesse pensando. [...] Allo stesso modo, la sola via per sapere che un uomo pensa è quello di essere quell'uomo in particolare. [...] Probabilmente A crederà "A pensa, mentre B no", mentre per B è l'esatto opposto "B pensa, ma A no". Invece di discutere in continuazione su questo punto, è normale attenersi alla educata condizione che ognuno pensi.

Le macchine di Turing sono macchine a stati discreti in grado di simulare altre macchine a stati discreti. Una macchina per sostenere il test dev'essere programmata considerando la descrizione di un uomo in termini discreti (stati interni, segnali, simboli). Dalla complessità del software, si legge tra le righe dell'articolo, emergeranno le funzioni intellettuali. Su questa aspettativa si fonda una disciplina nota come intelligenza artificiale, il cui scopo è la costruzione di una macchina in grado di riprodurre le funzioni cognitive umane. Sebbene le previsioni di Turing fossero che entro il 2000 sarebbe stata realizzata una macchina intelligente, finora nessuna ha superato il test.

Prove a confutazione del test

Il filosofo John Rogers Searle, ha proposto una variante al test di Turing, che ha preso il nome di stanza cinese (\rightarrow 3.2) e, tramite questa, ha messo in dubbio l'attendibilità del Test di Turing come prova sufficiente a dimostrare

che una macchina o un qualsiasi sistema informatico siano sistemi dotati di vera intelligenza, sia che questi abbiano superato o meno tale test.

3.2 Stanza cinese

La Stanza cinese è un esperimento mentale ideato da John Searle. Esso è un controesempio della teoria dell'intelligenza artificiale forte. Alla base del ragionamento di Searle è che la sintassi (grammatica) non è equivalente alla semantica (significato). Questo esempio prende spunto dal test di Turing (→ 3.1) ma lo modifica cercando di confutarlo.

Searle presentò l'argomentazione della Stanza cinese nel suo articolo "Minds, Brains and Programs" (Menti, cervelli e programmi) pubblicato nel 1980. Da allora, è stato un pilastro del dibattito sull'ipotesi chiamata da Searle intelligenza artificiale forte.

Descrizione dell'esperimento

I sostenitori dell'intelligenza artificiale forte sostengono che un computer opportunamente programmato non sia solo la simulazione o un modello della mente, ma che esso possa essere una mente. Esso cioè capisce, ha condizioni conoscitive e può pensare. L'argomento di Searle (o meglio, l'esperimento mentale) si oppone a questa posizione. L'argomentazione della stanza cinese è la seguente:

Si supponga che, nel futuro, si possa costruire un computer che si comporta come se capisse il cinese. In altre parole, il computer prenderebbe dei simboli cinesi in ingresso, consulterebbe una grande tabella che gli consentirebbe di produrre altri simboli cinesi in uscita. Si supponga che il comportamento di questo computer sia così convincente da poter facilmente superare il test di Turing. In altre parole, si supponga che il computer possa convincere un uomo, che parli correttamente cinese (per esempio un cinese), di parlare con un altro uomo che parla correttamente cinese, mentre in realtà sta parlando con un calcolatore. A tutte le domande dell'umano il computer risponderebbe appropriatamente, in modo che l'umano si convinca di parlare con un altro umano che parla correttamente cinese. I sostenitori dell'intelligenza artificiale forte concludono che il computer capisce la lingua cinese, come farebbe una persona, in quanto non c'è nessuna differenza tra il comportamento della macchina e di un uomo che conosce il cinese.

Ora, Searle chiede di supporre che lui si sieda all'interno del calcolatore. In altre parole, egli si immagina in una piccola stanza (la stanza cinese) dalla quale riceve dei simboli cinesi, e una tabella che gli consenta di produrre dei simboli cinesi in uscita. Searle fa notare che egli non capisce i simboli cinesi. Quindi la sua mancanza di comprensione dimostra che il calcolatore non può comprendere il cinese, poiché esso è nella sua stessa situazione. Il calcolatore è un semplice manipolatore di simboli, esattamente come lo è lui nella stanza

cinese - e quindi i calcolatori non capiscono quello che stanno dicendo tanto quanto lui.

Esperimenti mentali

Nel 1980, John Searle pubblicò "Minds, Brains and Programs" (Menti, Cervelli e Programmi) nella rivista *The Behavioral and Brain Sciences*. In questo articolo, Searle espose la sua argomentazione e di seguito rispose alle principali obiezioni che erano state sollevate durante le sue presentazioni a diversi campus universitari (vedi la prossima sezione). Inoltre, l'articolo di Searle fu pubblicato nella rivista insieme ai commenti ed alle critiche di 27 ricercatori di scienze cognitive. Questi 27 commenti furono seguiti dalle risposte di Searle ai suoi critici.

Negli ultimi due decenni del XX secolo, l'argomentazione della Stanza cinese fu oggetto di moltissime discussioni. Nel 1984, Searle presentò l'argomentazione della Stanza cinese in un libro (*Minds, Brains and Science*). Nel gennaio 1990, il popolare periodico *Scientific American* portò il dibattito all'attenzione del mondo scientifico: Searle incluse l'argomentazione della Stanza cinese nel suo articolo "Is the Brain's Mind a Computer Program?". Il suo pezzo era seguito da un articolo di risposta, "Could a Machine Think?", scritto da Paul e Patricia Churchland. Poco tempo dopo, fu pubblicato un confronto sulla Stanza cinese tra Searle ed un altro eminente filosofo, Jerry Fodor (in Rosenthal (ed.) 1991).

Il cuore dell'argomentazione è una teorica simulazione umana di un computer simile alla Macchina di Turing. L'essere umano nella Stanza cinese segue istruzioni in inglese per manipolare simboli cinesi, mentre un computer esegue un programma scritto in un linguaggio di programmazione. L'uomo crea l'apparenza della comprensione del cinese seguendo le istruzioni di manipolazione dei simboli, ma non giunge per questo a capire il cinese. Poiché un computer non fa altro che ciò che fa l'uomo - manipolare simboli in base alla loro sola sintassi - nessun computer, semplicemente eseguendo un programma, giunge a comprendere realmente il cinese.

Questa argomentazione, basata strettamente sulla Stanza Cinese, è rivolta contro la teoria che Searle chiama intelligenza artificiale forte. Secondo tale teoria, un computer adeguatamente programmato (o lo stesso programma) può comprendere il linguaggio naturale e possedere effettivamente altre capacità mentali simili a quelle degli uomini che imita. Secondo l'intelligenza artificiale forte, un computer può giocare a scacchi in modo intelligente, fare una mossa astuta, o capire il linguaggio. Per contro, l'intelligenza artificiale debole è la teoria secondo la quale i computer sono semplicemente utili nella psicologia, nella linguistica ed in altre aree di studio, in parte perché possono simulare capacità mentali. Ma l'intelligenza artificiale debole non afferma che i computer possano effettivamente capire o che siano intelligenti. L'argomentazione della Stanza cinese non è rivolta all'intelligenza artificiale debole, né ha lo sco-

po di dimostrare che le macchine non possono pensare - per Searle i cervelli sono proprio macchine in grado di pensare. È volta a confutare la teoria che calcoli formali svolti su simboli possano generare il pensiero. Potremmo riassumere l'argomentazione più ristretta come una *reductio ad absurdum* contro l'intelligenza artificiale forte nel modo seguente. Sia L un linguaggio naturale e chiamiamo un "programma per L" un programma per conversare correntemente in L. Un sistema computazionale è qualsiasi sistema, umano o meno, che può eseguire un programma.

- Se l'intelligenza artificiale forte è vera, allora esiste un programma per il cinese tale che se un qualsiasi sistema computazionale esegue quel programma, il sistema arriva in tal modo a capire il cinese.
- Potrei eseguire un programma per il cinese senza con questo arrivare a capire il cinese.
- Pertanto l'intelligenza artificiale forte è falsa.

La seconda premessa è sostenuta dall'esperimento mentale della Stanza cinese. La conclusione di questa argomentazione è che eseguire un programma non può generare comprensione. L'argomentazione più ampia include la tesi che l'esperimento mentale dimostra più in generale che non si può ottenere semantica (significato) dalla sintassi (manipolazione di simboli formali).

Il punto centrale dell'argomento di Searle è la distinzione tra sintassi e semantica. La stanza è in grado di combinare i caratteri secondo le regole, cioè si può dire che la stanza si comporta come se seguisse regole sintattiche. Ma, secondo Searle, essa non conosce il significato di ciò che ha fatto, cioè non ha contenuto semantico. I caratteri non rappresentano neppure simboli perché non sono interpretati in nessuna fase del processo.

Argomentazioni formali

Nel 1984 Searle fornì una versione più formale dell'argomentazione della quale la Stanza Cinese è una parte. Elencò quattro premesse:

- Premessa 1: I cervelli causano le menti
- Premessa 2: La sintassi non è sufficiente per la semantica
- Premessa 3: I programmi per calcolatore sono interamente definiti dalla loro struttura formale e sintattica
- Premessa 4: Le menti hanno contenuto semantico.

La seconda premessa sarebbe supportata dall'argomentazione della Stanza cinese, dato che Searle sostiene che la stanza segue solo regole sintattiche formali e non "comprende" il cinese. Searle ne deriva che queste premesse conducono direttamente a tre conclusioni:

- Conclusione 1: Nessun programma per calcolatore è sufficiente di per sé stesso a dare una mente ad un sistema. I programmi non sono menti.
- Conclusione 2: Il modo con il quale le funzioni del cervello causano le menti non può avvenire esclusivamente in virtù dell'esecuzione di un programma per calcolatore.
- Conclusione 3: Qualsiasi altra cosa sia ciò che genera le menti, essa dovrebbe avere poteri causali almeno equivalenti a quelli del cervello.

Searle descrive questa versione come "eccessivamente grezza". È stato considerevolmente dibattuto se questa argomentazione sia realmente valida. Queste discussioni si focalizzano sui vari modi nei quali le premesse possono essere analizzate. Si può interpretare la premessa 3 nel senso che i programmi per calcolatore hanno contenuto sintattico ma non semantico e così le premesse 2, 3 e 4 conducono validamente alla conclusione 1. Ciò porta a dibattere circa l'origine del contenuto semantico di un programma per calcolatore.

Critiche

Esistono molte critiche all'argomentazione di Searle. La maggioranza di esse rientra nella 'risposta del sistema' o nella 'risposta del robot'.

La risposta del sistema

Sebbene l'individuo nella Stanza cinese non comprenda il cinese, forse lo capiscono la persona e la stanza considerati insieme come sistema. La persona sarebbe proprio come un singolo neurone del cervello, e, come un singolo neurone da solo non può capire, ma può contribuire alla comprensione del sistema complessivo, così la persona non capisce, ma il sistema complessivo sì. La risposta di Searle è che qualcuno potrebbe teoricamente memorizzare il libro delle regole e che lavori all'aperto; ciò lo renderebbe capace di interagire come se capisse il cinese, ma ciononostante continuerebbe a seguire un insieme di regole, senza comprendere il significato dei simboli che sta utilizzando, infatti in questo caso non c'è nulla nel sistema che non sia anche nella persona e poiché la persona non capisce il cinese, non lo capisce neanche il sistema. Questo conduce all'interessante problema di una persona capace di conversare correntemente in cinese senza "sapere" il cinese, e secondo il contro-argomento tale persona capisce effettivamente il cinese anche se questi sosterrrebbe il contrario. Un argomento simile è che la persona non sa il cinese ma lo sa il sistema che comprende sia la persona che il libro delle regole.

La risposta del robot

Supponiamo che, invece che in una stanza, il programma sia collocato in un robot che possa muoversi ed interagire con il suo ambiente. Allora capirà cer-

tamente ciò che sta facendo? La risposta di Searle è di supporre che, senza che l'individuo nella Stanza cinese ne sia a conoscenza, alcuni segnali in ingresso che sta ricevendo provengano direttamente da una telecamera montata su un robot, e alcuni dei segnali in uscita siano utilizzati per muovere le braccia e le gambe del robot. Ciononostante, la persona nella stanza sta sempre seguendo le regole e non sa che cosa significhino i simboli. Supponiamo che il programma corrispondente al libro delle regole simuli in estremo dettaglio l'interazione dei neuroni nel cervello di un parlante cinese. Allora si dovrà dire che il programma capisce sicuramente il cinese? Searle replica che tale simulazione non avrà riprodotto le caratteristiche importanti del cervello - i suoi stati causali ed intenzionali. Ma cosa accadrebbe se una simulazione del cervello fosse connessa al mondo in modo tale da possedere il potere causale di un vero cervello - forse collegata ad un robot del tipo descritto prima? Allora sarebbe certamente capace di pensare. Searle concorda che è teoricamente possibile creare un'intelligenza artificiale, ma puntualizza che tale macchina dovrebbe avere gli stessi poteri causali di un cervello. Essa sarebbe più di un semplice programma per calcolatore.

Critica dei coniugi Churchland (Paul e Patricia)

Searle dà per scontato che i simboli formali non interpretati (come i simboli cinesi per l'uomo) non coincidono con i contenuti mentali. Per rispondere Searle ipotizza che i programmatori della stanza cinese sappiano il cinese e che abbiano costruito il sistema in modo da fargli elaborare informazioni in cinese (i simboli). Searle immagina che l'uomo nella stanza si stufi di mescolare soltanto questi simboli, che per lui sono privi di significato. Suppone dunque che l'uomo decida d'interpretare i simboli secondo le mosse di una partita a scacchi. Searle si chiede quale semantica esprime ora il sistema: quella del cinese, degli scacchi o entrambe? Si spinge oltre: suppone che un uomo dall'esterno decida che le manipolazioni dei simboli possano essere interpretate come previsioni dell'andamento della borsa. I simboli per l'uomo (formali) non hanno una semantica intrinseca, ma arbitrariamente assegnata.

Altre risposte

Il linguaggio naturale non è semplicemente una serie di eventi di impulso-risposta, ma piuttosto un'interazione. Una persona che esca dalla Stanza cinese potrebbe essere capace di capire effettivamente il cinese se le domande poste si riferissero a lui. Per esempio, la risposta alla domanda scritta in cinese "Qual è il tuo colore preferito?" può essere data solo con riferimento alla persona a cui viene chiesta. Se la persona può rispondere, allora capisce il cinese; se non può, allora persino per un osservatore esterno quella persona non parla cinese. Si può discutere se la semantica in un qualsiasi linguaggio naturale richieda una correlazione tra le parole ed oggetti del mondo reale. In una Stanza cinese,

non può esistere una relazione tra le parole o i simboli ed il mondo reale perché non esiste niente a cui riferirsi all'interno della Stanza cinese se non alle regole che vi sono scritte.

- supponiamo che un italiano faccia una vacanza lavoro a Londra per imparare la lingua, e quante difficoltà trovi nel comunicare le prime settimane.
- supponiamo che la sua vacanza diventi una permanenza definitiva nella città inglese, diventerà padrone della lingua inglese tanto da arrivare a PENSARE in inglese senza aver più bisogno di tradurre mentalmente quello che gli viene detto ed il suo pensiero dalla madre lingua all'inglese. Un'intelligenza artificiale per quanto evoluta e sempre più veloce possa essere rimane comunque un elaboratore di dati, che usa un metodo "stanza cinese" ad esempio per comunicare con un cinese.

Le ultime due note sembrano comunque presupporre che la mente umana non si comporti col metodo "stanza cinese" per comprendere inizialmente (stando all'esempio) la lingua italiana. A priori è invece possibile ed anzi probabile che il sistema di "decodifica" sia composto inizialmente dal solo meccanismo di traduzione [significante (italiano) → significato (mentale)] e che nelle prime fasi dell'apprendimento della lingua inglese l'individuo tenda a sviluppare un meccanismo intermedio di traduzione [significante (inglese) → significante (italiano)], ma che col tempo riesca a superare la necessità di questo passaggio in più sostituendogli un più efficace meccanismo di traduzione diretta [significante (inglese) → significato].

Rappresentazione della conoscenza

La rappresentazione della conoscenza è quella branca della matematica che si preoccupa di definire dei linguaggi che permettano di formalizzare la conoscenza al fine di potervi fare dell'inferenza. Lo studio della rappresentazione della conoscenza è fondamentale per l'intelligenza artificiale dato che questa prevede l'utilizzo delle precedenti conoscenze per migliorare e affinare le tecniche di risoluzione dei problemi. Senza un adeguato metodo di memorizzazione della conoscenza molte tecniche dell'intelligenza artificiale sarebbero inutili o molto limitate dato che non sarebbero in grado di apprendere dagli stimoli esterni e quindi non potrebbero evolversi.

Esistono principalmente due metodologie per rappresentare la conoscenza:

- Linguaggi formali (\rightarrow 4.1)
- Alberi di decisione (\rightarrow 4.2)

4.1 Linguaggi formali

Per rappresentare la conoscenza si può far uso dei linguaggi formali. Dal punto di vista matematico per linguaggio formale si intende un insieme di stringhe di lunghezza finita costruite sopra un alfabeto finito, cioè sopra un insieme finito di oggetti tendenzialmente semplici che vengono chiamati caratteri, simboli o lettere.

Di un linguaggio formale può far parte o meno la stringa muta o parola vuota, cioè la sequenza costituita da zero caratteri: questa spesso viene denotata come ϵ , ϵ o λ : qui preferiamo usare μ .

Un linguaggio può essere finito o infinito in quanto non si pongono limiti alla lunghezza delle stringhe.

Per rappresentare la conoscenza quasi sempre si fa uso della logica del primo ordine dato che questa logica consente di costruire enunciati relativamente

complessi che tramite le regole della logica possono essere analizzati per definire se rispetto alle premesse sono degli enunciati veri oppure no. Questa logica in sostanza permette di costruire dei risolutori automatici che analizzano gli enunciati e ne scartano quelli falsi senza bisogno dell'intervento umano.

I linguaggi formali risultano utili quando ci si trova a dover lavorare in un ambiente deterministico che possa essere descritto in modo agevole tramite enunciati logici. Agenti (\rightarrow 5.3) informatici che si muovono in un ambiente virtuale spesso si avvantaggiano di descrizioni dell'ambiente o delle sue regole effettuate tramite linguaggi formali. Invece ambienti non deterministici e dove si trova un elevato rumore di fondo come possono essere gli ambienti reali esplorati da robot mal si prestano all'utilizzo delle regole dato che una descrizione accurata del mondo sarebbe estremamente complessa e comunque incompleta. Inoltre il rumore di fondo insito in misure effettuate da sensori reali può rendere molto difficile la realizzazione di regole basate su logiche del primo ordine.

4.2 Alberi di decisione

Nella teoria dell'intelligenza artificiale un albero di decisione è un grafo di decisioni e delle loro possibili conseguenze, (incluso i relativi costi, risorse e rischi) utilizzato per creare un piano di azioni (plan) mirato ad uno scopo (goal). Un albero di decisione è costruito al fine di supportare l'azione decisionale (decision making).

Nel machine learning un albero di decisione è un modello predittivo, dove ogni nodo interno rappresenta una variabile, un arco verso un nodo figlio rappresenta un possibile valore per quella proprietà e una foglia il valore predetto per la variabile obiettivo a partire da i valori delle altre proprietà, che nell'albero è rappresentato del cammino (path) dalla nodo radice (root) al nodo foglia. Normalmente un albero di decisione viene costruito utilizzando tecniche di apprendimento a partire dall'insieme dei dati iniziali (data set), il quale può essere diviso in due sottoinsiemi: il training set sulla base del quali si crea la struttura dell'albero e il test set che viene utilizzato per testare l'accuratezza del modello predittivo così creato.

Nel data mining un albero di decisione viene utilizzato per classificare le istanze di grandi quantità di dati (per questo viene anche chiamato albero di classificazione). In questo ambito un albero di decisione descrive una struttura ad albero dove i nodi foglia rappresentano le classificazioni e le ramificazioni l'insieme delle proprietà che portano a quelle classificazioni. Di conseguenza ogni nodo interno risulta essere una macro-classe costituita dall'unione delle classi associate ai suoi nodi figli.

Il predicato che si associa ad ogni nodo interno (sulla base del quale avviene la ripartizione dei dati) è chiamato condizione di split.

In molte situazioni è utile definire un criterio di arresto (halting), o anche criterio di potatura (pruning) al fine di determinarne la profondità massima. Questo perché il crescere della profondità di un albero (ovvero delle sue dimensioni) non influisce direttamente sulla bontà del modello. Infatti, una crescita eccessiva della dimensione dell'albero potrebbe portare solo ad aumento sproporzionato della complessità computazionale rispetto ai benefici riguardanti l'accuratezza delle previsioni/classificazioni.

Apprendimento automatico

L'Apprendimento automatico (noto in letteratura come Machine Learning) rappresenta una delle aree fondamentali dell'Intelligenza Artificiale e si occupa della realizzazione di sistemi che si basano su osservazioni o esempi come dati per la sintesi di nuova conoscenza (classificazioni, generalizzazioni, riformulazioni).

Sono numerose le situazioni di difficile soluzione mediante algoritmi tradizionali. Queste tipicamente sono dovute alla presenza di uno o più dei seguenti fattori:

- Difficoltà di formalizzazione. Per esempio ognuno di noi sa riconoscere se una certa immagine contenga la faccia di un amico ma probabilmente nessuno sa descrivere una sequenza di passi computazionali che, eseguita sui pixel, consenta di rispondere alla domanda.
- Elevato numero di variabili in gioco
- Mancanza di teoria. Per esempio non esistono leggi matematiche note che regolino con esattezza l'andamento dei mercati finanziari.
- Necessità di personalizzazione. Se per esempio vogliamo classificare documenti come interessanti o non interessanti, la distinzione può dipendere significativamente dal particolare utente.

Gli algoritmi di apprendimento automatico sono tradizionalmente divisi in tre principali tipologie:

- Apprendimento supervisionato (→ 5.2)
- Apprendimento non supervisionato (→ 5.1)
- Apprendimento con rinforzo (→ 5.3)

5.1 Apprendimento non supervisionato

Le tecniche di apprendimento non supervisionato mirano ad estrarre, in modo automatico, della conoscenza a partire da basi di dati. Questo avviene senza una specifica conoscenza dei contenuti da analizzare. Un esempio tipico di questi algoritmi lo si ha nei motori di ricerca. Questi programmi, data una o più parole chiave, sono in grado di creare una lista di link rimandanti alle pagine che l'algoritmo di ricerca ritiene attinenti alla ricerca effettuata. La validità di questi algoritmi è legata alla utilità delle informazioni che riescono ad estrarre dalla base di dati, nell'esempio sopracitato è legata all'attinenza dei link con l'argomento cercato. Questi algoritmi lavorano confrontando i dati e ricercando similarità o differenze. Sono molto efficienti con elementi di tipo numerico, dato che possono utilizzare tutte le tecniche derivate dalla statistica, ma sono molto meno efficienti con dati non numerici. Se i dati sono dotati di un ordinamento intrinseco gli algoritmi riescono comunque ad estrarre informazioni, ma se i dati in ingresso non sono dotati di un qualche tipo di ordinamento spesso gli algoritmi falliscono. Se i dati non sono dotati di ordinamento cercare di ordinarli imponendo una graduatoria arbitraria non risolve il problema. Questo si può facilmente capire con un esempio. Supponiamo di disporre di un database con l'elenco dei colori utilizzati da uno stilista. Si potrebbe cercare di associare ad ogni colore uno specifico numero e su quello fare delle analisi di tipo statistico. Ma dato che l'associazione tra colore e numero è arbitraria si possono pensare ad infinite associazioni che darebbero infiniti risultati diversi. Questi algoritmi in conclusione lavorano correttamente in presenza di dati contenenti un ordinamento o un raggruppamento netto e chiaramente identificabile.

Principali algoritmi:

- Clustering (→ 5.1)
- Regole di associazione (→ 5.1)

Clustering

Il Clustering o analisi dei cluster o analisi di raggruppamento è una tecnica di intelligenza artificiale volta alla selezione e raggruppamento di elementi omogenei in un insieme di dati. Tutte le tecniche di clustering si basano sul concetto di distanza tra due elementi. Infatti la bontà delle analisi ottenute dagli algoritmi di clustering dipende essenzialmente da quanto è significativa la metrica e quindi da come è stata definita la distanza. La distanza è un concetto fondamentale dato che gli algoritmi di clustering raggruppano gli elementi a seconda della distanza e quindi l'appartenenza o meno ad un insieme dipende da quanto l'elemento preso in esame è distante dall'insieme. Le tecniche di clustering si possono basare principalmente su due tecniche.

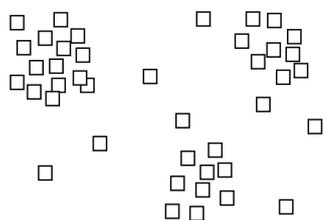


Fig. 5.1: Esempio di dati da analizzare

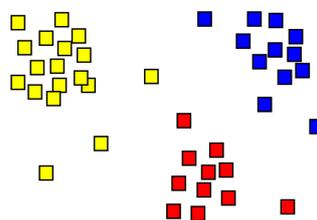


Fig. 5.2: Dati suddivisi in tre cluster

Tecniche

Dal basso verso l'alto Questa filosofia prevede che inizialmente tutti gli elementi siano considerati cluster a sé e poi l'algoritmo provvede ad unire i cluster più vicini. L'algoritmo continua ad unire elementi al cluster fino ad ottenere un numero prefissato di cluster oppure fino a che la distanza minima tra i cluster non supera un certo valore.

Dall'alto verso il basso All'inizio tutti gli elementi sono un unico cluster e poi l'algoritmo inizia a dividere il cluster in tanti cluster di dimensioni inferiori. Il criterio che guida la divisione è sempre quello di cercare di ottenere elementi omogenei. L'algoritmo procede fino a che non ha raggiunto un numero prefissato di cluster. Questo approccio è anche detto gerarchico.

Regole di associazione

Nella teoria dell'intelligenza artificiale le regole di decisione sono delle formule logiche che permettono di associare delle scelte o dei fatti a altri fatti o scelte. Queste regole permettono di individuare collegamenti in ampi insiemi di dati infatti vengono spesso utilizzate per analizzare grandi quantità di dati alla ricerca di associazioni utili. Strutture come i supermercati da molti anni analizzano gli acquisti dei clienti che utilizzano tessere a punti per realizzare pubblicità mirate e per migliorare l'organizzazione dei prodotti.

Le regole di associazioni partono dal presupposto di possedere un insieme di dati e che l'utente voglia cercare eventuali correlazioni interne. Spesso si vuole individuare se un certo numero di scelte possano portare con una certa probabilità minima ad effettuare una specifica scelta. Le regole hanno due parametri che definiscono la loro bontà, questi parametri sono il supporto e la confidenza. Il supporto identifica quante volte la specifica regola appare percentualmente nell'insieme di dati di test. La confidenza invece specifica con quale percentuale nell'insieme di dati di esempio la regola fa una previsione corretta.

Date queste premesse usualmente si richiede una confidenza di almeno il 90% mentre il supporto può essere anche di solo pochi punti percentuali.

5.2 Apprendimento supervisionato

L'Apprendimento supervisionato è una tecnica di apprendimento automatico che mira a istruire un sistema informatico in modo da consentirgli di risolvere dei compiti in automatico.

- Si definiscono i dati in ingresso come insieme I , (tipicamente vettori)
- Si definisce l'insieme dei dati in uscita come insieme O (gli output possono essere valori continui (regressione), o una etichetta numerica)
- Si definisce una funzione h che associa ad ogni dato in ingresso (I) la sua risposta corretta (O)

Tutti gli algoritmi di apprendimento supervisionato partono dal presupposto che se forniamo all'algoritmo un numero adeguato di esempi l'algoritmo sarà in grado di creare una funzione h_1 che approssimerà la funzione h . Se l'approssimazione di h risulterà adeguata quando proporremo ad h_1 dei dati in ingresso mai analizzati la funzione dovrebbe essere in grado di fornire delle risposte in uscita simili a quelle fornite da h e quindi corrette o quasi. Molti di questi algoritmi in sostanza lavorano in un mondo lineare, presupponendo che a ingressi simili necessitino di uscite simili. Nel nostro mondo questo in generale non è vero, basta vedere le dinamiche caotiche legate al tempo, ma esistono molte condizioni in cui questa semplificazione è accettabile. Si può facilmente intuire che il buon funzionamento di questi algoritmi dipende in modo significativo dai dati in ingresso; se si forniscono pochi ingressi l'algoritmo potrebbe non aver abbastanza esperienza, mentre molti dati in ingresso potrebbero rendere eccessivamente lento l'algoritmo, dato che la funzione h_1 generata dagli ingressi potrebbe essere molto complicata. Questi algoritmi sono molto sensibili al rumore, anche pochi dati errati potrebbero rendere l'intero sistema non affidabile e condurlo a decisioni errate. Tradizionalmente i principali algoritmi sono stati:

- Alberi di decisione (\rightarrow 4.2)
- Regole di decisione (\rightarrow 5.2)
- Sistemi esperti (\rightarrow 5.2)

Regole di decisione

Nella teoria delle dell'intelligenza artificiale le regole di decisione sono delle formule logiche che permettono di associare delle scelte o dei fatti a altri fatti o scelte. Queste regole permettono di individuare collegamenti in ampi insiemi di dati infatti vengono spesso utilizzate per analizzare grandi quantità di dati alla ricerca di associazioni utili. Strutture come i supermercati da molti anni analizzano gli acquisti dei clienti che utilizzano tessere a punti per realizzare pubblicità mirate e per migliorare l'organizzazione dei prodotti.

Nella teoria dell'intelligenza artificiale le regole di associazione sono delle formule logiche che cercano di legare alcuni parametri di ingresso a delle decisioni. Queste regole svolgono lo stesso compito svolto dagli alberi di decisione (\rightarrow 4.2) ma con un approccio diverso.

Le regole infatti analizzano direttamente lo spazio di esempi e cercano di individuare le formule logiche in grado di riprodurre correttamente le scelte. Esistono molti algoritmi per implementare le regole di decisione ma in generale questi algoritmi cercano di individuare i parametri più significativi ed associano a questi le scelte più comuni. Gli esempi che vengono correttamente categorizzati da queste regole vengono rimossi dall'insieme di analisi e poi si ripete l'analisi fino ad ottenere un insieme di dati da analizzare vuoto.

Procedendo con questa metodologia vengono generate molte regole che spesso coprono solo dei casi particolari e quindi le regole a questo punto non mostrano un comportamento abbastanza generale. Quindi usualmente si procede a effettuare una fase di potatura delle regole dove queste vengono analizzate e raggruppate in regole più generali in modo da ottenere delle regole effettivamente utili.

Finita la fase di *potatura* delle regole queste vengono applicate ad un insieme di test in modo da poter verificare l'effettiva validità delle regole nell'analizzare situazioni nuove.

A differenza degli alberi di decisione le regole soffrono di un problema di sovracopertura dato che per uno specifico caso possono attivarsi più regole quindi anche l'ordine delle regole è importante, prima vanno poste le regole più generali e in seguito vanno poste le regole che intercettano i casi particolari dato che questi sono meno comuni.

Sistemi esperti

Il termine sistema esperto identifica una categoria di programmi informatici legati all'intelligenza artificiale che, dopo essere stati opportunamente istruiti, sono in grado di dedurre nuove informazioni da un insieme di informazioni di partenza.

Un sistema esperto si fonda sulla competenza umana registrata nella cosiddetta base di conoscenza (ad esempio sotto forma di regole), aggiornabile in base all'esperienza. Come avviene per l'esperto umano, il sistema esperto

può operare su dati qualitativi e incompleti. Può infatti utilizzare forme di ragionamento approssimato, attraverso tecniche probabilistiche o facendo ricorso alla cosiddetta "fuzzy logic" ($\rightarrow 6$), reso spesso come logica sfumata, un tipo di logica a più valori.

Ciò che rende diversi i sistemi esperti da altri algoritmi di intelligenza artificiale è che un sistema esperto è sempre in grado di spiegare logicamente le sue decisioni, mentre ciò non è vero per esempio nelle reti neurali. Tuttavia, se una rete neurale si limita a produrre le regole di una fuzzy cognitive machine, un sistema fuzzy, allora si ha il vantaggio della flessibilità delle reti neurali associato alla possibilità di risalire alla logica delle decisioni, tramite l'esame delle regole fuzzy prodotte.

Categorie

I sistemi esperti si dividono in due categorie principali:

Sistemi esperti basati su regole I sistemi esperti basati su regole sono dei programmi composti da regole della forma IF condizione THEN azione. Dati una serie di fatti i sistemi esperti, grazie alle regole di cui sono composti, riescono a dedurre nuovi fatti.

Per esempio: Supponiamo di avere un problema di salute, forniamo al sistema esperto i seguenti fatti: 1) ho mal di testa 2) sono raffreddato; 3) la temperatura corporea è di 38 gradi centigradi il sistema esperto assume i fatti e sceglie una regola così formata: IF ((mal di testa) AND (raffreddore) AND (temperatura = 38)) THEN (Il suo problema è INFLUENZA)

Sistemi esperti basati su alberi Un sistema esperto basato su alberi, dato un insieme di dati ed alcune deduzioni, creerebbe un albero che classificherebbe i vari dati. Nuovi dati verrebbero analizzati dall'albero e il nodo di arrivo rappresenterebbe la deduzione.

È da notare che un sistema esperto non è "intelligente" nel senso comune della parola, ossia in modo creativo. Le deduzioni di un sistema esperto non possono uscire dall'insieme di nozioni immesse inizialmente e dalle loro conseguenze. Ciò che li rende utili è che, come i calcolatori elettronici, possono maneggiare una gran quantità di dati molto velocemente e tenere quindi conto di una miriade di regole e dettagli che un esperto umano può ignorare, tralasciare o dimenticare.

5.3 Apprendimento con rinforzo

L'apprendimento per rinforzo è una filosofia di programmazione che punta a realizzare algoritmi in grado di apprendere e adattarsi alle mutazioni dell'ambiente. Questa tecnica di programmazione si basa sul presupposto di potere

ricevere degli stimoli dall'esterno a seconda delle scelte dell'algoritmo. Quindi una scelta corretta comporterà un premio mentre una scelta scorretta porterà ad una penalizzazione del sistema. L'obiettivo del sistema è il raggiungimento del maggior premio possibile e di conseguenza del migliore risultato possibile. Le tecniche legate all'apprendimento per rinforzo si dividono in due categorie:

- **Algoritmi ad apprendimento continuo:**
Queste tecniche partono dal presupposto di disporre di un meccanismo semplice in grado di valutare le scelte dell'algoritmo e quindi premiare o punire l'algoritmo a seconda del risultato. Queste tecniche sono in grado di adattarsi anche a delle modifiche sostanziali dell'ambiente. Un esempio sono i programmi di riconoscimento del parlato o i programmi di OCR che con l'utilizzo migliorano le loro prestazioni.
- **Algoritmi ad addestramento preventivo:**
Questi algoritmi partono dalla constatazione che valutare costantemente le azioni dell'algoritmo può essere un procedimento non automatizzabile oppure molto costoso e in questo caso si applica una prima fase in cui si istruisce l'algoritmo e quando il sistema viene ritenuto affidabile viene cristallizzato e reso non più modificabile. Molti componenti elettronici usano delle reti neurali al loro interno, e i pesi sinaptici di queste reti non sono modificabili dato che sono fissati durante la realizzazione del circuito.

Da notare che le categorie sopraelencate sono delle scelte implementative più che delle differenze concettuali dell'algoritmo e quindi spesso un algoritmo può ricadere della prima o nella seconda categoria a seconda di come viene implementato dal progettista.

Principali algoritmi:

- Reti neurali (→ 5.3)
- Algoritmi genetici (→ 5.3)
- Classificatori (→ 5.3)

Agente

In intelligenza artificiale un agente è un componente hardware o software che cerca ameno in parte di simulare il comportamento umano o il comportamento animale di interazione tra l'organismo e l'ambiente.

A livello generale un agente è composto da tre componenti, un sistema sensoriale (S), un sistema decisionale (D) e degli attuatori (A).

- Il sistema sensoriale analizza l'ambiente in cui è immerso l'agente.

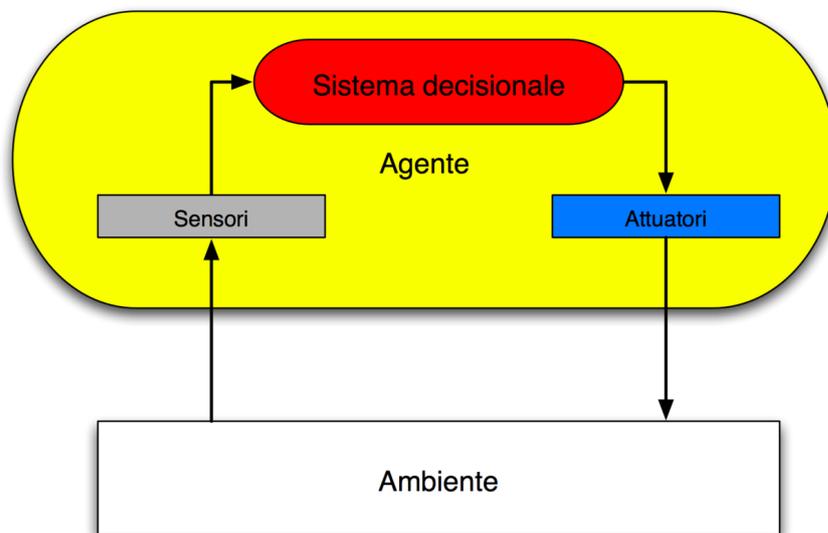


Fig. 5.3: Rappresentazione schematica di un agente

- Il sistema decisionale basandosi sui dati provenienti dall'ambiente su informazioni interne dell'agente e su un insieme di regole o procedure decisionali stabilisce come deve comportarsi l'agente.
- Gli attuatori sono i componenti che vengono gestiti dal sistema decisionale e trasmettono gli stimoli all'ambiente.

Tipologie di agente

Esistono varie tipologie di agenti. Il più semplice agente è l'agente stimolo-risposta SR. Questo agente ad uno specifico stimolo dell'ambiente risponde sempre con la stessa risposta.

Subito dopo si trova l'agente stimolo-stato-risposta SXR. Questo agente tiene uno stato interno che può influenzare l'uscita e quindi la risposta dell'agente dipende dallo stato presente e dagli stati passati che vengono memorizzati dallo stato interno.

In terzo tipo di agente è chiamato deliberativo. Questi agenti dispongono di un modello più o meno complesso dell'ambiente e utilizzano questo modello e gli stati di ingresso per decidere lo stato di uscita.

Apprendimento per rinforzo

L'agente riceve degli stimoli dal mondo esterno. Il suo sistema decisionale comanda gli attuatori per svolgere delle azioni e in base a come queste azioni

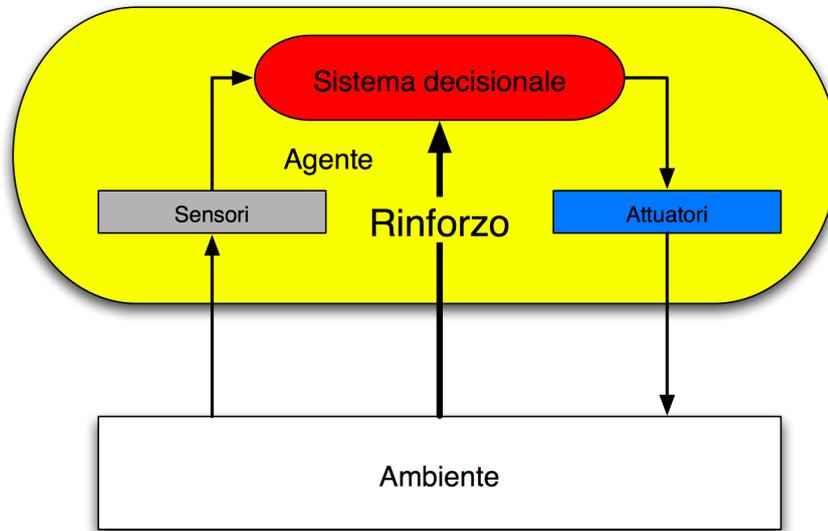


Fig. 5.4: Schema di un agente con rinforzo

impattano sul mondo esterno l'agente può ricevere degli stimoli dal mondo esterno che lo premiano se l'azione è favorevole o che lo puniscono se invece l'azione non è favorevole per l'obiettivo dell'agente.

Descrizione dell'ambiente

Un agente per funzionare non necessita di una descrizione completa dell'ambiente anche perché una descrizione realmente esaustiva dell'ambiente è impossibile, dato che richiederebbe troppi dati da memorizzare e troppi sensori. Ad un agente basta conoscere solo alcune proprietà dell'ambiente per poter operare correttamente. Usualmente si assume che l'ambiente possa essere descritto come un processo di decisione Markoviano. Un processo di decisione Markoviano è formato da:

- Un insieme finito di stati.
- Un insieme finito di azioni
- Un insieme finito di transizioni che associano ad ogni coppia di elementi stato-azione una certa distribuzione di probabilità.
- Una funzione di rinforzo che associa ad ogni transizione un valore numerico (il rinforzo)

La caratteristica importante dei processi di decisione Markoviani è che lo stato attuale dipende solamente dallo stato precedente e non da tutta la

storia passata. Questo ovviamente non è sempre vero ma ciò semplifica molto la realizzazione degli agenti fornendo spesso buone prestazioni.

L'obiettivo dell'agente è di massimizzare il rinforzo però, essendo che ogni istante l'agente riceve un rinforzo ci troveremo di fronte ad una somma che tende all'infinito se si somma il rinforzo su tutti gli istanti temporali (che sono infiniti). Esistono diversi metodi per aggirare il problema, il più semplice prevede di limitare la somma ad un certo numero di istanti (per esempio a cento istanti a partire dall'attuale). La seconda prevede di effettuare la media del rinforzo atteso e quindi di dividere la somma per il numero di istanti. Comunque la formulazione più comune è il rinforzo atteso scontato. In questa formulazione il valore del rinforzo viene moltiplicato per una funzione che ha un valore elevato quando il tempo è prossimo al tempo presente e che decresce all'allontanarsi dell'istante temporale. Questa formulazione dà maggior peso al tempo presente rispetto al tempo futuro.

Gestione dell'agente

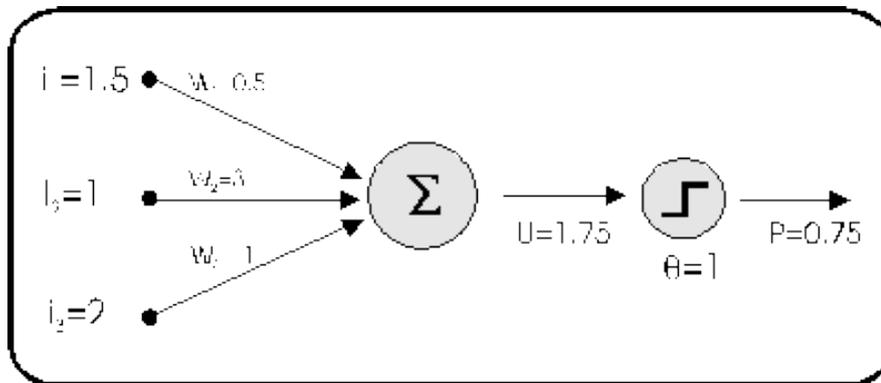
Gli agenti sono guidati nelle loro azioni da una politica. La politica associa ad ogni coppia di stato azione un indice di probabilità. Questo indice indica quanto è probabile che in quello stato si esegua quella azione. Un agente in ogni stato può sapere che alcune azioni porteranno un certo risultato e può avere delle azioni che non ha ancora eseguito e di cui quindi non conosce il rinforzo. La politica gli dice se intraprendere la stessa strada e quindi scegliere tra le azioni note o se provare una delle azioni incognite alla ricerca di un rinforzo maggiore. La realizzazione di una politica ottimale è un problema difficilmente risolvibile e dipende dallo specifico problema; in generale si è visto che è meglio effettuare inizialmente una politica molto aggressiva esplorando spesso stati incogniti alla ricerca di rinforzi molto consistenti per poi spostarsi gradatamente verso una politica più conservativa che sfrutti gli stati conosciuti per ottenere un rinforzo costante.

Reti neurali

Una rete neurale è un insieme di neuroni biologici tra loro interconnessi. Nell'uso moderno si intende però di solito con rete neurale una rete di neuroni artificiali, che cerca di simulare il funzionamento dei neuroni all'interno di un sistema informatico. Può essere composta sia da programmi che da hardware dedicato. Spesso viene utilizzata in congiunzione alla logica fuzzy e funge da base per molti sistemi di intelligenza artificiale.

Fondamenti biologici

In quasi tutti gli organismi viventi sono presenti complesse organizzazioni di cellule nervose, con compiti di riconoscimento delle configurazioni assunte dall'ambiente esterno, memorizzazione e reazione agli stimoli provenienti dallo



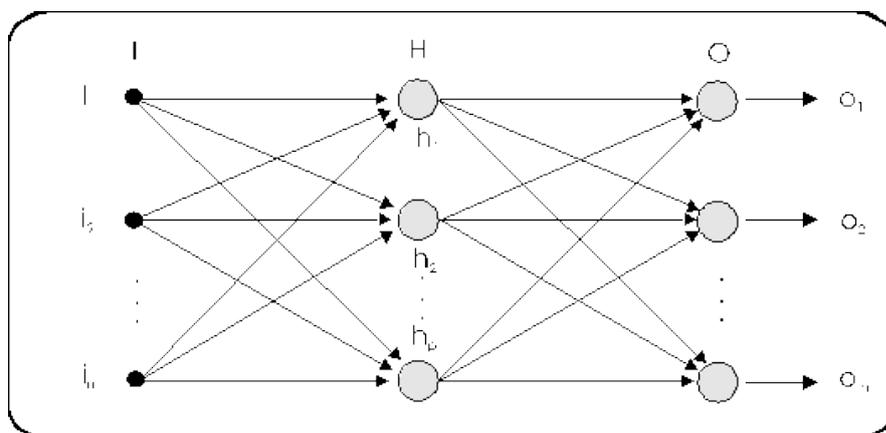
stesso. Il cervello umano rappresenta probabilmente il più mirabile frutto dell'evoluzione per le sue capacità di elaborare informazioni. Al fine di compiere tali operazioni, le reti biologiche si servono di un numero imponente di semplici elementi computazionali (neuroni) fittamente interconnessi in modo da variare la loro configurazione in risposta agli stimoli esterni: in questo senso si può parlare di apprendimento ed i modelli artificiali cercano di catturare questo tratto distintivo della biologia.

Tipicamente un neurone è costituito di 3 parti principali: il soma (corpo cellulare), l'assone (linea di uscita del neurone unica ma che si può diramare anche in migliaia di rami) e il dendrite (detto anche albero dendritico in quanto la sua estesa ramificazione ne ricorda la forma, il quale funge da linea di entrata del neurone che riceve segnali in ingresso da altri assoni tramite sinapsi). Il corpo cellulare fa una "somma (algebraica) pesata" (integrazione) dei segnali in ingresso. Se il risultato eccede un certo valore di soglia allora il neurone si attiva ed è prodotto un "potenziale di azione" che è mandato all'assone. Se il risultato non eccede il valore di soglia, il neurone rimane in uno stato di riposo. Una rete neurale artificiale riceve segnali esterni su uno strato di nodi (unità di elaborazione) d'ingresso, ciascuno dei quali è collegato con numerosi nodi interni, organizzati in più livelli. Ogni nodo elabora i segnali ricevuti e trasmette il risultato a nodi successivi.

Funzionamento di una rete neurale feedforward

Le reti neurali si basano principalmente sulla simulazione di neuroni artificiali opportunamente collegati. Il modello rappresentato in figura è quello proposto da McCulloch e Pitts.

I suddetti neuroni ricevono in ingresso degli stimoli e li elaborano. L'elaborazione può essere anche molto sofisticata ma in un caso semplice si può pensare che i singoli ingressi vengano moltiplicati per un opportuno valore detto peso, il risultato delle moltiplicazioni viene sommato e se la somma supera una certa soglia il neurone si attiva attivando la sua uscita. Il peso indica l'efficacia sinaptica della linea di ingresso e serve a quantificarne l'importanza,



un ingresso molto importante avrà un peso elevato, mentre un ingresso poco utile all'elaborazione avrà un peso inferiore. Si può pensare che se due neuroni comunicano fra loro utilizzando maggiormente alcune connessioni allora tali connessioni avranno un peso maggiore.

I singoli neuroni vengono collegati alla schiera di neuroni successivi, in modo da formare una rete di neuroni. Normalmente una rete è formata da tre strati. Nel primo abbiamo gli ingressi (I), questo strato si preoccupa di trattare gli ingressi in modo da adeguarli alle richieste dei neuroni. Se i segnali in ingresso sono già trattati può anche non esserci. Il secondo strato è quello nascosto (H, hidden), si preoccupa dell'elaborazione vera e propria e può essere composto anche da più colonne di neuroni. Il terzo strato è quello di uscita (O) e si preoccupa di raccogliere i risultati ed adattarli alle richieste del blocco successivo della rete neurale. Queste reti possono essere anche molto complesse e coinvolgere migliaia di neuroni e decine di migliaia di connessioni.

Storia

L'ampia varietà di modelli non può prescindere dal costituente di base, il neurone artificiale proposto da W.S. McCulloch e W. Pitts in un famoso lavoro del 1943, il quale schematizza un combinatorio lineare a soglia, con dati binari multipli in entrata e un singolo dato binario in uscita: un numero opportuno di tali elementi, connessi in modo da formare una rete, è in grado di calcolare semplici funzioni booleane.

Nel 1958, F. Rosenblatt introduce il primo schema di rete neurale, detto perceptron (perceptrone), antesignano delle attuali reti neurali, per il riconoscimento e la classificazione di forme, allo scopo di fornire un'interpretazione dell'organizzazione generale dei sistemi biologici. Il modello probabilistico di Rosenblatt è quindi mirato all'analisi, in forma matematica, di funzioni quali l'immagazzinamento delle informazioni, e della loro influenza sul riconoscimento dei patterns; esso costituisce un progresso decisivo rispetto al modello

binario di McCulloch e Pitts, perché i suoi pesi sinaptici sono variabili e quindi il perceptrone è in grado di apprendere.

L'opera di Rosenblatt stimola una quantità di studi e ricerche, e suscita un vivo interesse e notevoli aspettative nella comunità scientifica, destinate tuttavia ad essere notevolmente ridimensionate allorché nel 1969 Marvin Minsky e Seymour A. Papert mostrano i limiti operativi delle semplici reti a due strati basate sui perceptroni, e dimostrano l'impossibilità di risolvere per questa via molte classi di problemi, ossia tutti quelli non caratterizzati da separabilità lineare delle soluzioni: questo tipo di rete neurale non è abbastanza potente, infatti non è in grado di calcolare neanche la funzione *or esclusivo* (XOR). Di conseguenza, a causa di queste limitazioni, ad un periodo di euforia per i primi risultati della cibernetica (come veniva chiamata negli anni sessanta), segue un periodo di diffidenza durante il quale tutte le ricerche in questo campo non ricevono più alcun finanziamento dal governo degli Stati Uniti d'America; le ricerche sulle reti tendono, di fatto, a ristagnare per oltre un decennio, e l'entusiasmo iniziale risulta fortemente ridimensionato.

Il contesto matematico per addestrare le reti MLP (Multi-Layers Perceptron, ossia perceptrone multistrato) fu stabilito dal matematico americano Paul Werbos nella sua tesi di dottorato (Ph.D.) del 1974. Uno dei metodi più noti ed efficaci per l'addestramento di tale classe di reti neurali è il cosiddetto algoritmo di retropropagazione dell'errore (error backpropagation), proposto nel 1986 da David E. Rumelhart, G. Hinton e R. J. Williams, il quale modifica sistematicamente i pesi delle connessioni tra i nodi, così che la risposta della rete si avvicini sempre di più a quella desiderata. L'algoritmo di backpropagation (BP) è una tecnica d'apprendimento tramite esempi, costituente una generalizzazione dell'algoritmo d'apprendimento per il perceptrone sviluppato da Rosenblatt nei primi anni sessanta. Mediante questa tecnica era possibile, come detto, trattare unicamente applicazioni caratterizzabili come funzioni booleane linearmente separabili.

L'algoritmo di apprendimento si basa sul metodo della discesa del gradiente che permette di trovare un minimo locale di una funzione in uno spazio a N dimensioni. I pesi associati ai collegamenti tra gli strati di neuroni si inizializzano a valori piccoli e casuali e poi si applica la regola di apprendimento presentando alla rete dei pattern di esempio. Queste reti neurali sono capaci di generalizzare in modo appropriato, cioè di dare risposte plausibili per input che non hanno mai visto.

L'addestramento di una rete neurale di tipo BP avviene in due diversi stadi: *forward-pass* e *backward-pass*. Nella prima fase i vettori in input sono applicati ai nodi in ingresso con una propagazione in avanti dei segnali attraverso ciascun livello della rete (*forward-pass*). Durante questa fase i valori dei pesi sinaptici sono tutti fissati. Nella seconda fase la risposta della rete viene confrontata con l'uscita desiderata ottenendo il segnale d'errore. L'errore calcolato è propagato nella direzione inversa rispetto a quella delle connessioni sinaptiche. I pesi sinaptici infine sono modificati in modo da minimizzare la

differenza tra l'uscita attuale e l'uscita desiderata (backward-pass).

Tale algoritmo consente di superare le limitazioni del perceptrone e di risolvere il problema della separabilità non lineare (e quindi di calcolare la funzione XOR), segnando il definitivo rilancio delle reti neurali, come testimoniato anche dall'ampia varietà d'applicazioni commerciali: attualmente la BP rappresenta un algoritmo di largo uso in molti campi applicativi.

Algoritmo di Backpropagation

L'algoritmo di backpropagation è utilizzato nell'apprendimento con supervisione. Esso permette di modificare i pesi delle connessioni in modo tale che si minimizzi una certa funzione errore E . Tale funzione dipende dal vettore h-esimo di output \overline{out}^h restituito dalla rete, dato il vettore h-esimo di ingresso \overline{x}^h e dal vettore h-esimo di output \overline{y}^h che noi desideriamo (che fa parte del **training set**). Il training set è dunque un insieme di N coppie di vettori $(\overline{x}^h, \overline{y}^h)$, con $h = 1, \dots, N$. La funzione errore che si deve minimizzare si può scrivere come: $\frac{1}{2} \sum_h \sum_k (out_k^h - y_k^h)^2$, dove l'indice k rappresenta il valore corrispondente al k -esimo neurone di output. E è una funzione dipendente dai pesi (che in generale variano nel tempo). Per minimizzare $E(w)$ si può usare l'algoritmo della discesa del gradiente (**gradient-descent**). L'algoritmo parte da un punto generico $\overline{x}(0)$ e calcola il gradiente $\nabla f(\overline{x}^0)$. Il gradiente dà la direzione verso cui muoversi lungo la quale ho il massimo incremento (o decremento se considero $-\nabla$). Definita la direzione ci si muove di una distanza η predefinita a priori e si trova un nuovo punto $\overline{x}(1)$ sul quale è calcolato nuovamente il gradiente. Si continua iterativamente finché il gradiente non è nullo.

L'algoritmo di backpropagation può essere diviso in due passi:

- Forward pass: l'input dato alla rete è propagato al livello successivo e così via ai livelli successivi (il flusso di informazioni si sposta in avanti, cioè forward). Si calcola dunque $E(w)$, l'errore commesso.
- Backward pass: L'errore fatto dalla rete è propagato all'indietro (backward) e i pesi sono aggiornati in maniera appropriata.

Pregi

Le reti neurali per come sono costruite lavorano in parallelo e sono quindi in grado di trattare molti dati. Si tratta in sostanza di un sofisticato sistema di tipo statistico dotato di una buona immunità al rumore; se alcune unità del sistema dovessero funzionare male, la rete nel suo complesso avrebbe delle riduzioni di prestazioni ma difficilmente andrebbe incontro ad un blocco del sistema. I software di ultima generazione dedicati alle reti neurali richiedono comunque buone conoscenze statistiche; il grado di apparente utilizzabilità immediata non deve trarre in inganno, pur permettendo all'utente di effettuare da subito previsioni o classificazioni, seppure con i limiti del caso. Da un

punto di vista industriale, risultano efficaci quando si dispone di dati storici che possono essere trattati con gli algoritmi neurali. Ciò è di interesse per la produzione perché permette di estrarre dati e modelli senza effettuare ulteriori prove e sperimentazioni.

Difetti

I modelli prodotti dalle reti neurali, anche se molto efficienti, non sono spiegabili in linguaggio simbolico umano: i risultati vanno accettati "così come sono", da cui anche la definizione inglese delle reti neurali come "black box". Come per qualsiasi algoritmo di modellazione, anche le reti neurali sono efficienti solo se le variabili predittive sono scelte con cura. Non sono in grado di trattare in modo efficiente variabili di tipo categorico (per esempio, il nome della città) con molti valori diversi. Necessitano di una fase di addestramento del sistema che fissi i pesi dei singoli Neuroni e questa fase può richiedere molto tempo, se il numero dei record e delle variabili analizzate è molto grande. Non esistono teoremi o modelli che permettano di definire la rete ottima, quindi la riuscita di una rete dipende molto dall'esperienza del creatore.

Algoritmi genetici

L'algoritmo genetico appartiene ad una particolare classe di algoritmi utilizzati in diversi campi, tra cui l'intelligenza artificiale. È un metodo euristico di ricerca ed ottimizzazione, ispirato al principio della selezione naturale di Charles Darwin che regola l'evoluzione biologica.

Questo tipo di algoritmi è detto genetico perché mutua terminologia dalla genetica, branca della biologia.

Gli algoritmi genetici sono applicabili alla risoluzione di un'ampia varietà di problemi d'ottimizzazione non indicati per gli algoritmi classici, compresi quelli in cui la funzione obiettivo è discontinua, non derivabile, stocastica, o fortemente non lineare.

Principi di funzionamento

Un tipico algoritmo genetico parte da un certo numero di possibili soluzioni (individui) chiamate popolazione e provvede a farle evolvere nel corso dell'esecuzione: a ciascuna iterazione, esso opera una selezione di individui della popolazione corrente, impiegandoli per generare nuovi elementi della popolazione stessa, che andranno a sostituire un pari numero d'individui già presenti, e a costituire in tal modo una nuova popolazione per l'iterazione (o generazione) seguente. Tale successione di generazioni evolve verso una soluzione ottima del problema assegnato.

La loro evoluzione viene ottenuta attraverso una parziale ricombinazione delle soluzioni, ogni individuo trasmette parte del suo patrimonio genetico ai propri discendenti, e l'introduzione di mutazioni casuali nella popolazione

di partenza, sporadicamente quindi nascono individui con caratteristiche non comprese tra quelle presenti nel corredo genetico della specie originaria.

Finita la fase di evoluzione la popolazione delle soluzioni viene analizzata e vengono tenute solo le soluzioni che meglio risolvono il problema: gli individui con le qualità più adatte all'ambiente in cui si trovano hanno quindi maggiori possibilità di sopravvivere e riprodursi. Queste soluzioni subiranno una nuova fase di evoluzione e così via.

Alla fine ci si aspetta di trovare una popolazione di soluzioni che riescano a risolvere adeguatamente il problema posto. Non vi è modo di decidere a priori se l'algoritmo sarà effettivamente in grado di trovare una soluzione accettabile. Di norma gli algoritmi genetici vengono utilizzati per problemi di ottimizzazione per i quali non si conoscono algoritmi di complessità lineare o polinomiale.

Un caso particolare di applicazione di algoritmi genetici è Acovea, un software studiato per trovare il profilo migliore delle opzioni di ottimizzazione del compilatore gcc: un problema di elevata complessità.

Dettaglio del funzionamento

La soluzione del problema viene codificata in una struttura, di solito una stringa, detta gene.

Inizialmente viene creato un certo numero di geni in maniera casuale e si definisce una funzione che restituisce la "bontà" di un gene come soluzione del problema, detta funzione di fitness.

L'algoritmo consiste nell'applicazione di operazioni, che tendono a modificare la popolazione dei geni, nel tentativo di migliorarli in modo da ottenere una soluzione sempre migliore.

L'evoluzione procede quindi in passi, per ognuno di questi viene per prima cosa eseguito un ordinamento dei geni sulla base del risultato della funzione di fitness. Vengono poi eseguite le operazioni su un numero di geni stabilito dai parametri dell'algoritmo, che in generale determinano quanti geni devono subire crossover e mutazioni, e in quale misura.

L'algoritmo evolve quindi attraverso i seguenti punti:

- generazione, in maniera casuale, una popolazione iniziale;
- creazione di una sequenza di nuove popolazioni, o generazioni. In ciascuna iterazione, gli individui della popolazione corrente sono usati per creare la generazione successiva, e a questo scopo si compiono degli ulteriori passi:
 - ciascun membro della popolazione corrente è valutato calcolandone il rispettivo valore di fitness (idoneità);
 - si determina un opportuno ordinamento di tali individui sulla base dei valori di fitness;

- gli individui più promettenti sono selezionati come genitori;
- a partire da tali individui si genera un pari numero di individui della generazione successiva, e ciò può avvenire secondo due modalità distinte, vale a dire effettuando cambiamenti casuali su un singolo genitore (mutazione) oppure combinando opportunamente le caratteristiche di una coppia di genitori (incrocio);
- gli individui così generati vanno a sostituire i genitori consentendo la formazione della generazione successiva;

infine, l'algoritmo s'interrompe quando uno dei criteri d'arresto è soddisfatto.

Crossover In base a un coefficiente stabilito inizialmente, alcune parti dei geni risultati migliori vengono scambiate, nell'ipotesi che questo possa migliorare il risultato della funzione di fitness nel successivo "passo evolutivo".

Mutazione La mutazione consiste nella modifica casuale di alcune parti dei geni con valore di fitness più basso, in base a coefficienti definiti inizialmente. Queste modifiche puntano a migliorare il valore della funzione per il gene in questione.

Basi teoriche e storia

La traduzione e l'estensione dei principi esposti, validi per i sistemi biologici e per i sistemi artificiali, si deve storicamente a John Holland. Dopo un non breve periodo di tempo, in cui il rilievo di tale lavoro non fu pienamente riconosciuto, l'impiego degli algoritmi genetici si è andato consolidando in ambito informatico, ingegneristico, finanziario ed ovviamente nel campo delle scienze sociali e naturali.

Un teorema, dovuto a Holland, assicura che, sotto determinate ipotesi, gli individui con alti valori di fitness tendono a crescere esponenzialmente nella popolazione attraverso il meccanismo dell'incrocio, assicurando così la convergenza dell'algoritmo genetico verso una soluzione ottimale. Nel suo teorema sugli schemi (schema theorem), detto anche "teorema fondamentale degli algoritmi genetici", egli dimostra che uno schema (ossia una particolare combinazione di geni che occupano posizioni precise all'interno di un cromosoma) prolifera più rapidamente se, oltre ad avere un alto valore di fitness, contiene un piccolo numero di geni specifici non lontani l'uno dall'altro. Ciò, infatti, riduce la probabilità di distruggere lo schema durante la fase di riproduzione.

Brevi successioni di geni, che assumono particolari valori, definiscono i cosiddetti blocchi costitutivi (building blocks): favorendo l'incrocio dei cromosomi meglio adattati, in cui si riscontra statisticamente la presenza di peculiari blocchi costitutivi, l'algoritmo aumenta la probabilità che blocchi costituenti

opportuni, provenienti da cromosomi diversi, si ritrovino in uno stesso cromosoma. Assumendo che l'associazione di siffatti blocchi sia dunque vantaggiosa, dovrà anche ritenersi probabile la comparsa di un cromosoma (soluzione) eccellente per il problema in esame, in un tempo ragionevole.

La dimostrazione del teorema degli schemi è basata sull'ipotesi di codifica binaria, ma Wright (1991) l'ha estesa al caso di codifica con numeri reali; lo stesso Wright ha mostrato che una codifica reale è da preferirsi nel caso di problemi continui d'ottimizzazione. Herrera e Lozano (1998) hanno poi presentato un'ampia rassegna di operatori genetici applicabili a cromosomi codificati mediante numeri reali, compresi vari tipi di operatori di crossover (incrocio).

Pertanto, il campo dei numeri reali costituisce ormai un'appropriata e consolidata forma di rappresentazione per gli algoritmi genetici in domini continui. Tuttavia, a causa di complessi fenomeni di interazione non lineare (epistaticità) tra gruppi di valori di una stringa rappresentante un individuo, non si può affermare con certezza che la combinazione di blocchi costitutivi altamente performanti sia sempre destinata a produrre individui ancora migliori. In altri termini, non sempre l'operazione genetica di crossover produce risultati accettabili, e anzi a volte accade che, a partire da due genitori estremamente promettenti, si ottenga un discendente decisamente meno valido.

Programmazione Evolutiva

In questi casi, la cosiddetta programmazione evolutiva, sviluppata principalmente da David B. Fogel, anch'essa ispirata all'evoluzione naturale ma non alla genetica, può essere impiegata con successo nelle applicazioni. Tale metodologia differisce dagli algoritmi genetici in quanto non utilizza l'operazione genetica di crossover, che invece per essi risulta imprescindibile.

Programmazione Genetica

La programmazione genetica, elaborata fondamentalmente ad opera di John R. Koza, è invece un metodo per la generazione automatica di programmi, a partire da una descrizione ad alto livello del task da svolgere, e basato sul principio darwiniano della selezione naturale allo scopo di sviluppare una popolazione di programmi migliorativi nell'arco delle successive generazioni. Essa si avvale di operazioni capaci di alterare l'architettura di detti programmi e di prendere decisioni sull'uso delle subroutine, dei loop, della ricorsione e della memoria. Da ciò si nota che la programmazione genetica costituisce in sostanza un'estensione degli algoritmi genetici al caso di popolazioni costituite da programmi di dimensione variabile; la programmazione genetica sostituisce in altri termini alla stringa di lunghezza costante, codificata in vario modo, un programma con struttura ad albero, il cui corpo (radice, nodi intermedi) è costituito da funzioni aritmetiche o logiche, mentre i nodi ter-

minali rappresentano variabili o costanti numeriche. Pertanto la popolazione risulta ora composta da un numero opportuno di programmi, i quali mediante le operazioni genetiche di riproduzione (non è prevista alcuna mutazione) producono, in un certo numero di generazioni, il programma che risolve al meglio un problema assegnato, in forma topologica parametrizzata.

Classificatori

I sistemi a classificatori in intelligenza artificiale sono dei modelli che cercano di simulare l'apprendimento e l'evoluzione di agenti adattativi.

Questi sistemi si basano su un insieme di regole. Le regole esprimono dei fatti o delle condizioni usualmente con costrutti del tipo *se X allora esegui Y* sebbene non vi siano vincoli teorici per l'utilizzo di costrutti più complessi.

Queste regole vengono valutate in base a un meccanismo di retroazione che misurando la risposta dell'ambiente permette di definire la bontà o utilità delle regole. Le regole che meglio si adattano all'ambiente vengono mantenute mentre le altre regole vengono eliminate.

Utilizzando gli algoritmi genetici vengono realizzati nuovi set di regole utilizzando le regole sopravvissute come base di partenza. Questo nuovo set di regole viene utilizzato dal classificatore per interagire con l'ambiente e per essere a loro volta valutate in modo da poter procedere iterativamente.

I sistemi basati su classificatori possono adattarsi rapidamente alle mutazioni dell'ambiente grazie agli algoritmi genetici che forniscono costantemente mutazioni ed evoluzioni per provare nuove strade. In un ambiente statico comunque la presenza delle regole sopravvissute alla fase di retroazione permette al classificatore di mostrare un buon adattamento all'ambiente.

Sistemi basati su classificatori tendono a sviluppare un insieme di regole utilizzate per i problemi generici e poi piccoli sottoinsiemi (isole) di regole che entrano in azione al presentarsi di particolari situazioni.

Logica Fuzzy

La teoria Fuzzy nasce nella metà degli anni 60, quando il professore Lotfi A. Zadeh espone la necessità di utilizzare concetti imprecisi nella risoluzione di problemi della vita reale. In un articolo del 1965 Zadeh pubblicò il primo articolo che introduceva il concetto di insiemi fuzzy, i quali, a differenza degli insiemi tradizionali, inglobano il concetto di grado di appartenenza di un elemento all'insieme. Con questo concetto, quindi, si è resa possibile la rappresentazione di concetti imprecisi che tutti noi utilizziamo normalmente nel linguaggio naturale. Ad esempio nella frase "Luigi è alto" il concetto di alto risulta impreciso poiché non ci è data una definizione rigorosa di quanto Luigi sia alto, con gli insiemi fuzzy, quindi, possiamo definire il concetto di alto ad esempio con il seguente insieme:

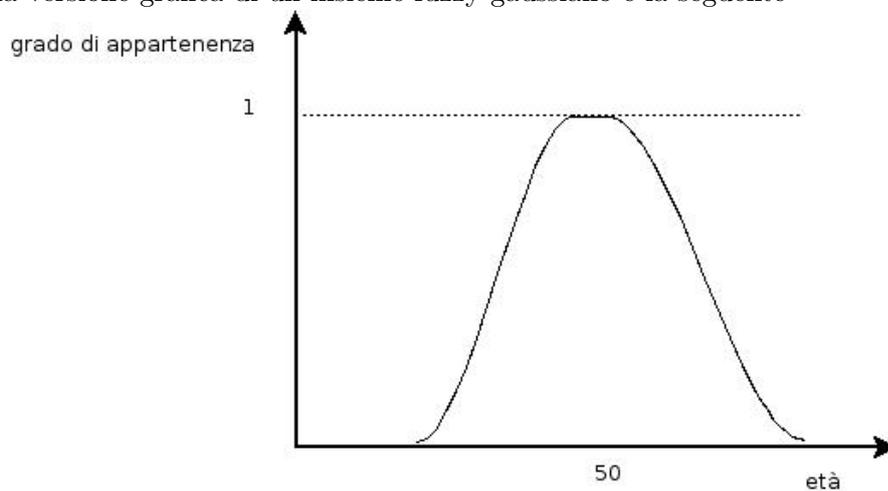
$$0,1/150 + 0,25/160 + 0,5/170 + 0,8/180 + 0,9/190 + 1/200.$$

L'insieme fuzzy visto prima è rappresentato da una serie di valori, che indicano l'altezza di una persona in centimetri, e da un numero compreso nell'intervallo $[0,1]$ che rappresenta il grado di appartenenza dell'elemento all'insieme, quindi, scrivendo $0,5/170$, stiamo dicendo che una persona alta 170 cm è di altezza media poiché il grado di appartenenza dell'elemento 170 è 0,5, mentre una persona alta 200 cm è una persona decisamente alta, poiché il grado di appartenenza dell'elemento 200 è 1. È possibile rappresentare un insieme fuzzy tramite una funzione, che indica la funzione di appartenenza degli elementi all'insieme, le funzioni più utilizzate sono le funzioni triangolari, poiché è possibile creare sistemi fuzzy efficienti utilizzando solo insiemi fuzzy triangolari, si possono anche creare insiemi fuzzy trapezoidali, gaussiane e così via. Un esempio di funzione di appartenenza può essere la seguente:

$$\mu(x) = \frac{1}{1 + \left(\frac{50-x}{10}\right)^2}$$

Questa è una funzione gaussiana che può essere utilizzata, ad esempio, per rappresentare l'insieme fuzzy "persone con età vicina ai 50", poiché esse forma

una curva che raggiunge il picco massimo (di valore 1) con x uguale a 50.
Una versione grafica di un insieme fuzzy gaussiano è la seguente



Sugli insiemi fuzzy è possibile applicare tutte le operazioni fondamentali degli insiemi classici quali unione, intersezione e negazione, tali operazioni vengono effettuate mediante l'utilizzo di speciali funzioni di appartenenza che sono: per l'unione funzioni s-norm, ad esempio il massimo, per l'intersezione funzioni t-norm, ad esempio il minimo e per la negazione si applica la formula $\mu(x) = 1 - x$.

Problema di soddisfacimento di vincoli

Molti problemi nell'ambito dell'intelligenza artificiale sono classificabili come problemi di soddisfacimento di vincoli (Constraint Satisfaction Problem o CSP); fra questi citiamo problemi combinatori, di allocazione di risorse, pianificazione e ragionamento temporale. Questi problemi possono essere risolti efficientemente attraverso tecniche ben note di risoluzione di CSP.

Formalmente, un CSP può essere definito su un insieme finito di variabili (X_1, X_2, \dots, X_n) i cui valori appartengono a domini finiti di definizione (D_1, D_2, \dots, D_n) e su un insieme di vincoli. Un vincolo su un insieme di variabili è una restrizione dei valori che le variabili possono assumere simultaneamente. Concettualmente, un vincolo può essere visto come un insieme che contiene tutti i valori che le variabili possono assumere contemporaneamente: un vincolo tra k variabili $c(X_{i_1}, X_{i_2}, \dots, X_{i_k})$ è un sottoinsieme del prodotto cartesiano dei domini delle variabili coinvolte $D_{i_1}, D_{i_2}, \dots, D_{i_k}$ che specifica quali valori delle variabili sono compatibili con le altre. Questo insieme può essere rappresentato in molti modi, per esempio per mezzo di matrici, equazioni, disuguaglianze o relazioni. Una soluzione ad un CSP è un assegnamento di valori alle variabili che soddisfi tutti i vincoli.

Un classico esempio di problema che può essere visto come CSP è il rompicapo delle otto regine. Il problema consiste nel disporre su una scacchiera otto regine che non si attacchino a vicenda; naturalmente, non è possibile disporne più di otto, perché questo implicherebbe posizionarne due su una stessa riga. Per formalizzare il problema come CSP dobbiamo identificare un insieme di variabili, un insieme di domini ed un insieme di vincoli. Poiché su ogni riga e su ogni colonna della scacchiera dovrà essere posta esattamente una regina, una possibile formalizzazione del problema è quella di considerare come variabili ciascuna delle otto righe della scacchiera; l'insieme delle variabili sarà quindi $V = \{R_1, R_2, \dots, R_8\}$. Ciascuna di queste variabili può assumere un valore che rappresenta la colonna su cui si trova la regina corrispondente, per cui i domini delle variabili sono le otto possibili colonne: $D_1 = D_2 = \dots = D_8 = \{1, 2, 3, 4, 5, 6, 7, 8\}$. Abbiamo quindi già inserito in

questa formalizzazione il concetto che due regine non possono trovarsi sulla stessa riga; resta da imporre che due regine non possono trovarsi né sulla stessa colonna, né sulla stessa diagonale. L'insieme dei vincoli sarà dunque:

$$\forall i, j. R_i \neq R_j$$

ad indicare che due regine non possono trovarsi sulla stessa colonna e:

$$\forall i, j, \text{ se } R_i = a \text{ e } R_j = b, \text{ allora } i - j \neq a - b \text{ e } i + j \neq a + b$$

per imporre che due regine non possono trovarsi sulla stessa diagonale.

7.1 Algoritmi

Per risolvere problemi di soddisfacimento di vincoli, vengono usati diversi algoritmi, fra cui algoritmi che valutano i vincoli a posteriori:

- Backtracking
- Backjumping
- Backmarking

Algoritmi che valutano i vincoli a priori

- Forward Checking
- Partial Look-Ahead
- Full Look-Ahead

Algoritmi per semplificare il problema

- Arc-Consistenza
- Path-Consistenza

Applicazioni

Le applicazioni dell'intelligenza artificiale sono molto varie. I sistemi a linguaggi formali vengono utilizzati in contesti dove le regole base sono chiare e certe. Diversi robot di ricerca utilizzano definizioni basate su linguaggi formali per definire gli scopi e gli obiettivi.

Gli alberi di ricerca vengono utilizzati quando è possibile organizzare la conoscenza chiaramente in modo da poterla utilizzare per risolvere in automatico i problemi.

I sistemi di apprendimento non supervisionato come il clustering e le regole di associazione vengono utilizzati per analizzare imponenti moli di dati alla ricerca di particolari associazioni o di particolari gruppi di utenti che deviano da un preciso profilo. Questi strumenti vengono molto utilizzati dalle società di analisi per classificare gli utenti e fornire pubblicità o prodotti personalizzati. Le società spesso utilizzano questi strumenti per individuare comportamenti devianti come utenti che stanno per diventare morosi o che sono probabilmente dei truffatori dato che normalmente questi utenti hanno un comportamento nettamente diverso rispetto alla maggioranza degli utenti e le tecniche di clustering riescono sovente ad individuarli o comunque definiscono un cluster dove vengono concentrati tutti gli utenti che hanno un'elevata probabilità di diventare utenti morosi.

I sistemi di apprendimento supervisionato vengono utilizzati per organizzare una base di conoscenza spesso fornita da un esperto umano in modo da poter risolvere problemi in automatico. Questi sistemi vengono utilizzati per esempio dalle compagnie petrolifere per effettuare analisi preliminari delle prospezioni petrolifere e dagli ospedali per fornire diagnosi preliminari.

I sistemi ad apprendimento con rinforzo come le reti neurali, gli algoritmi genetici e i classificatori sono utilizzati in ambienti incerti dove le condizioni sono soggette a continui mutamenti o dove i dati possono essere affetti da errori a volta anche molto rilevanti. Le reti neurali sono spesso utilizzate per riconoscere le immagini, la voce e i percorsi. Gli algoritmi genetici invece sono in grado di analizzare ampi spazi degli stati alla ricerca di soluzioni che se non

sono le migliori spesso non prossime a quelle ottime. I classificatori cercano di riunire i vantaggi delle due precedenti tecnologie.

Fonti

Immagini

5.1 - *licenza*: pubblico dominio; *fonte*:
<https://commons.wikimedia.org/wiki/File:Cluster-1.svg>

5.2 - *licenza*: pubblico dominio; *fonte*:
<https://commons.wikimedia.org/wiki/File:Cluster-2.svg>

5.3 - *licenza*: pubblico dominio; *fonte*:
<https://commons.wikimedia.org/wiki/File:Agente.png>

5.3 - *licenza*: pubblico dominio; *fonte*:
https://commons.wikimedia.org/wiki/File:Agente_con_rinforzo.png

5.3 - *licenza*: pubblico dominio; *fonte*:
<https://commons.wikimedia.org/wiki/File:Neurone.artificiale.gif>

5.3 - *licenza*: pubblico dominio; *fonte*:
<https://commons.wikimedia.org/wiki/File:Rete-Neurale.gif>

6 - *licenza*: GFDL, CC bySA 3.0; *fonte*:
<https://it.wikibooks.org/wiki/File:Fuzzy.jpg>

Testi

Intelligenza artificiale/Definizione - *fonte*: https://it.wikibooks.org/w/index.php?title=Intelligenza_artificiale/Definizione&oldid=85610

Intelligenza artificiale/Storia - *fonte*: https://it.wikibooks.org/w/index.php?title=Intelligenza_artificiale/Storia&oldid=371332

Intelligenza artificiale/Filosofia - *fonte*: https://it.wikibooks.org/w/index.php?title=Intelligenza_artificiale/Filosofia&oldid=221742

Intelligenza artificiale/Test di Turing - *fonte*:
https://it.wikibooks.org/w/index.php?title=Intelligenza_artificiale/Test_di_Turing&oldid=321754

Intelligenza artificiale/Stanza cinese - *fonte*:
https://it.wikibooks.org/w/index.php?title=Intelligenza_artificiale/Stanza_cinese&oldid=181105

Intelligenza artificiale/Rappresentazione della conoscenza - *fonte*:
https://it.wikibooks.org/w/index.php?title=Intelligenza_artificiale/Rappresentazione_della_conoscenza&oldid=211648

Intelligenza artificiale/Linguaggi formali - *fonte*:
https://it.wikibooks.org/w/index.php?title=Intelligenza_artificiale/Linguaggi_formali&oldid=85616

Intelligenza artificiale/Alberi di decisione - *fonte*:
https://it.wikibooks.org/w/index.php?title=Intelligenza_artificiale/Alberi_di_decisione&oldid=321755

Intelligenza artificiale/Apprendimento automatico - *fonte*:
https://it.wikibooks.org/w/index.php?title=Intelligenza_artificiale/Apprendimento_automatico&oldid=85617

Intelligenza artificiale/Apprendimento non supervisionato - *fonte*:
https://it.wikibooks.org/w/index.php?title=Intelligenza_artificiale/Apprendimento_non_supervisionato&oldid=202089

Intelligenza artificiale/Clustering - *fonte*: https://it.wikibooks.org/w/index.php?title=Intelligenza_artificiale/Clustering&oldid=220096

Intelligenza artificiale/Regole di associazione - *fonte*:
https://it.wikibooks.org/w/index.php?title=Intelligenza_artificiale/Regole_diAssociazione&oldid=158076

Intelligenza artificiale/Apprendimento supervisionato - *fonte*:
https://it.wikibooks.org/w/index.php?title=Intelligenza_artificiale/Apprendimento_supervisionato&oldid=85621

Intelligenza artificiale/Regole di decisione - *fonte*:
https://it.wikibooks.org/w/index.php?title=Intelligenza_artificiale/Regole_di_decisione&oldid=85622

Intelligenza artificiale/Sistemi esperti - *fonte*:
https://it.wikibooks.org/w/index.php?title=Intelligenza_artificiale/Sistemi_esperti&oldid=343270

Intelligenza artificiale/Apprendimento con rinforzo - *fonte:*
https://it.wikibooks.org/w/index.php?title=Intelligenza_artificiale/Apprendimento_con_rinforzo&oldid=85624

Intelligenza artificiale/Agente - *fonte:* https://it.wikibooks.org/w/index.php?title=Intelligenza_artificiale/Agente&oldid=211664

Intelligenza artificiale/Reti neurali - *fonte:*
https://it.wikibooks.org/w/index.php?title=Intelligenza_artificiale/Reti_neurali&oldid=333442

Intelligenza artificiale/Algoritmi genetici - *fonte:*
https://it.wikibooks.org/w/index.php?title=Intelligenza_artificiale/Algoritmi_genetici&oldid=211665

Intelligenza artificiale/Classificatori - *fonte:*
https://it.wikibooks.org/w/index.php?title=Intelligenza_artificiale/Classificatori&oldid=85628

Intelligenza artificiale/Fuzzy - *fonte:* https://it.wikibooks.org/w/index.php?title=Intelligenza_artificiale/Fuzzy&oldid=212339

Intelligenza artificiale/Problema di soddisfacimento di vincoli - *fonte:*
https://it.wikibooks.org/w/index.php?title=Intelligenza_artificiale/Problema_di_soddisfacimento_di_vincoli&oldid=85631

Intelligenza artificiale/Applicazioni - *fonte:*
https://it.wikibooks.org/w/index.php?title=Intelligenza_artificiale/Applicazioni&oldid=335085