

# L'Intelligenza Artificiale

di Raffaello De Masi

## Cosa è una tecnica di intelligenza artificiale

### Quarta parte

L'intelligenza artificiale copre, l'abbiamo detto più volte, un ampio spettro di problemi. Proprio per questo appare quanto mai necessario stabilire una procedura di base indicante le strutture inequivocabili destinate a risolvere il problema e per dare ad una domanda una ed una sola risposta.

Purtroppo i problemi trattati dalla A.I. hanno il difetto di essere tanto differenziati tra loro da possedere ben pochi fattori in comune e generalizzabili. La domanda è: ci sono tecniche comuni a più campi d'interesse tali da poter essere generalizzabili e utilizzabili almeno preliminarmente per la soluzione dei problemi stessi? E, in caso contrario, come è possibile mutare e modificare metodologie di altri campi per produrre risultati efficaci anche in tale nuovo orizzonte? E, ancora, una volta generalizzato un algoritmo, quali sono i controlli da adottare per verificare se questo può essere utilizzato efficacemente per la soluzione di un problema analogo, ma non del tutto identico?

Fare questo significa esaminare tecniche, magari già adottate ed individualmente testate da altri sperimentatori, ed adattarle alle nostre esigenze.

Uno dei principi fondamentali della A.I. è il possesso di conoscenze. Ma questa caratteristica irrinunciabile di qualsiasi sistema più o meno intelligente, possiede una serie di proprietà indesiderabili, almeno in relazione alle caratteristiche operative di un computer:

- un discreto bagaglio di conoscenze, infatti,

- è voluminoso

- è difficile da caratterizzare, vale a dire che è estremamente disagiata, per certe categorie di informazioni (ad esempio concetti astratti), una loro classifica rigida e sequenziale.

- è costantemente in evoluzione, ovvero, va continuamente aggiornato e, talora, del tutto sovvertito in alcuni particolari anche fondamentali.

Come è, allora, possibile con questi handicap, giungere a tecniche generalizzate ed algoritmi ottimali in intelli-

genza artificiale? Occorre stabilire uno standard, un metodo che sfrutti e raccolga conoscenze secondo processi logici tali da renderle:

- generalizzabili: in altre parole, non deve essere necessario che ogni situazione, azione, od informazione debba essere rappresentata separatamente, ma è invece necessario che situazioni, condividenti importanti proprietà possano essere raggruppate insieme. Se la conoscenza non potesse sfruttare tale possibilità di sintesi, occorrerebbe disporre di molta più memoria di quella disponibile da qualsiasi essere pensante, sia uomo o macchina.

- comprensibili da ognuno (in effetti si tratta, qui, di un'altra faccia della generalizzabilità della informazione stessa)

- facilmente modificabili, per poter correggere errori, od aggiornare le tecniche di intervento specifico in funzione di nuove accresciute conoscenze del problema.

- deve essere utilizzabile in diverse situazioni, anche se, ovviamente, in modo non totalmente accurato e completo.

Sebbene ogni problema di intelligenza artificiale sia risolvibile senza far ricorso a tali tecniche (ad esempio, il programma in Basic della ricerca dei menu, presentato un paio di puntate or sono, potrebbe non essere generalizzabile), appare ovvio come sia più conveniente, nello stesso esempio, utilizzare la tecnica indicata per il riconoscimento dei pasti richiesti per altri scopi, magari simili, come la gestione di una gelateria o di una paninoteca. Ecco allora che si passa dal semplice programma in Basic alla tecnica di Intelligenza Artificiale, intesa come insieme di tool, di metodologie destinate a rendere indipendente, almeno in parte, la macchina ed il suo programma dal problema che deve trattare. Nel caso particolare (menu del ristorante) viene soprattutto tenuta d'occhio la generalizzazione, visto che, per forza di cose, altri aspetti, come corredo di informazioni (nome e tipologia dei pasti serviti), non sono riassumibili, ed occuperanno irrimediabilmente grossi quantitativi di memoria.

Passiamo ad un esempio in cui viene applicato quanto finora detto e vediamo come il riconoscimento delle particolarità sia esemplificabile in un

algoritmo, di uso abbastanza generalizzato, modificabile, se necessario (attenzione, non nel principio, ma in certi particolari applicativi), ed utilizzabile in molte altre situazioni, anche se, come vedremo, con un minimo di elasticità, dovuta alla non perfetta analogia dei problemi, o di particolari ad essi afferenti.

Immaginiamo di dover costruire un programma per una macchina destinata a comprendere la scrittura. Sorvoleremo qui sul problema pratico di come fisicamente vadano lette le lettere stesse (il caso più ovvio potrebbe essere un digitizzatore), e fermiamoci sul dilemma di come le lettere vadano effettivamente interpretate, vale a dire, più precisamente, come vada rappresentato l'input. La grandezza della minima unità digitizzata, ovviamente, dipende dalla macchina e dalla sua risoluzione di lettura. In ogni caso, l'approccio più logico al problema ci pare essere quello della lettura della lettera come un pattern, un reticolato con maglie aperte o cieche, corrispondenti, in pratica, a parti piene (valore 1) o vuote (valore 0) d'inchiostro. La figura 1 mostra un esempio di input.

Leggere, pertanto, un carattere, almeno nelle condizioni da noi imposte inizialmente, vuol dire riconoscere la griglia, la scacchiera di punti bianchi e neri, e da essa risalire alla lettera di partenza. Non è facile, ma non impossibile: vediamo come la macchina può operare per il riconoscimento e quali sono i principali handicap per l'ottenimento di un risultato ottimale.

Faremo, di seguito tre esempi, via via più complessi, caratterizzati, l'uno rispetto all'altro, dall'inserimento di una o più variabili penalizzanti, riguardanti, ad esempio, la forma, la grandezza, l'orientamento, ecc. delle lettere o dei simboli da riconoscere.

Il primo caso, il più semplice (rigido) impone l'uso di caratteri, destinati ad essere riconosciuti, sempre eguali ad un modello di base. È il caso, ad esempio, di uno scritto redatto sempre nello stesso formato e corpo (es. un elaborato tipografico o dattilografico). Il riconoscimento è delegato ad una primitiva operazione di archiviazione delle caratteristiche singole di ogni carattere; in pratica ogni lettera, numero, simbolo, anche personalizzato, di cui si desidera il riconoscimento, viene

inserito, preventivamente, in un comune pattern, diciamo, tanto per identificare, di dimensioni pari a  $10 \times 10$  (attenzione: la matrice, quadrata, avrà ingombro massimo pari alla distanza del più alto punto di una lettera maiuscola, tipicamente la A, ed il più basso di un carattere discendente, tipicamente la q). Trasformeremo tale sequenza, ovviamente esprimibile come numero binario, in un numero, che inseriremo in una array, e, a questo punto, il gioco è fatto! Numeri e lettere saranno legati biunivocamente.

Questa routine ha diversi problemi, abbastanza intuibili, di cui il maggiore appare legato al grosso numero, ancorché binario, destinato a rappresentare la nostra matrice di dimensioni  $10 \times 10$ . Tradotto in decimale, il numero passerebbe in notazione esponenziale (il solo  $2^{100}$  già darebbe  $1.26E30$ ) e si perderebbe qualsiasi valore di dettaglio. Né pare utilizzabile, per ovvi motivi di memoria, il criterio di immagazzinare il valore in matrici bidimensionali.

Generalmente, se si desidera usare questo primo metodo, il problema viene semplificato, utilizzando una riduzione delle righe e delle colonne lette. Usualmente, nella griglia, vengono lette solo tre righe, corrispondenti, nella maggior parte dei casi, a quella centrale, ed a quelle del primo e del secondo medio. Ogni carattere va riferito ad  $8 \times 3 = 24$  bit di pattern. L'unica precauzione da adottare è quella di verificare, ovviamente, che, per un particolare caso, due lettere non abbiano sequenze perfettamente eguali; in tal caso è sufficiente spostare le righe di analisi, di parsing, per risolvere agevolmente il problema.

Il procedimento ha, comunque, una serie di difetti, peraltro ovvi anche al profano. La sequenza di caratteri da leggere deve avere sempre le stesse dimensioni, e deve essere, ancora, sempre uguale a se stessa. È perciò adottabile solo su scritti estremamente standardizzati, come, l'abbiamo già detto, elaborati meccanicizzati. Ma, anche in tal caso, la pur più piccola sbavatura dell'inchiostro di stampa porta ad errori, che, nel caso di lettura di numeri e codici può condurre a conseguenze disastrose. Ancora, l'introduzione di un nuovo carattere nella array porta alla riprogrammazione totale o, almeno, al controllo minuzioso ed estenuante di tutta la serie già presente, onde verificare che la nuova sequenza non si sovrapponga a qualcuna già esistente. D'altro canto, aumentando il numero di caratteri da leggere diviene sempre più ridotto lo spazio operativo, visto che, generalmente, in una lettera, esistono addensamenti preferenziali di punti.

Vediamo, allora, di affinare la tecni-

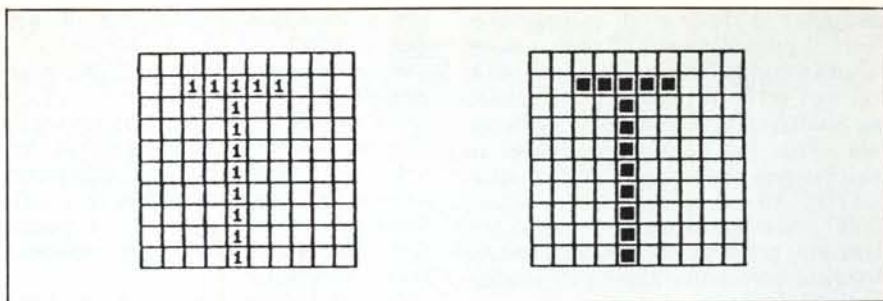


Figura 1 - Rappresentazione di una lettera tramite pattern di punti su una griglia di  $10 \times 10$ , a destra come punti, a sinistra in termini di bit.

ca, e di rendere, una volta tanto, meno sensibile ai particolari la macchina.

Un sistema più raffinato di affrontare il problema è dato dall'algoritmo che segue: si parte ancora da una griglia di bianchi e neri (0 ed 1), come nel caso precedente; il dizionario di confronto, per così dire, è costruito immagazzinando le varie lettere campione in un set di vettori, ognuno di lunghezza N; ciascuno di tali vettori è calcolato prendendo la matrice di bit corrispondente alla griglia, e dividendola in N regioni (quadrate o rettan-

golari, a scelta). Di qui il numero di 1 di ogni regione è calcolato e registrato nella posizione del vettore corrispondente a tale regione. N deve essere grande abbastanza (la lettera deve essere piuttosto sminuzzata) in modo che ogni griglia risulti rappresentata da vettori differenti.

Operativamente, per classificare una matrice in input si procede in questo modo:

- si converte la matrice in un vettore N-dimensionale
- si calcola la distanza, nello spazio N-dimensionale, del vettore di input da ognuno dei vettori della griglia.
- si classifica l'input come corrispondente alla griglia a cui più si avvicina.

Quest'ultimo concetto è di grande importanza: il fatto che si vada, per analogia, a ricercare il carattere più prossimo a quello di analisi permette di superare, infatti, il problema delle piccole differenze tra carattere analizzato e di raffronto. Ovviamente, il primo scotto che si paga è quello di una maggiore lentezza del programma operativo; ma non è tutto.

L'arma della elasticità di analisi è a doppio taglio: è vero che piccole differenze vengono perdonate, ma ciò comporta, inevitabilmente, dei rischi. Tanto per intenderci, se viene mantenuta una notevole elasticità nell'accettare la differenza tra lettere, cosa necessaria specie quando viene analizzata scrittura manuale, si può cadere nella confusione per caratteri simili (v. fig. 2a). In altre parole, l'algoritmo non distingue tra differenze significative e non significative. Ancora, come nella figura 2b, vengono lette come diverse, lettere eguali, ma aventi diverso rapporto altezza-larghezza. Una soluzione teorica è l'introduzione di griglie specifiche per ogni possibile combinazione di caratteristiche, ma esiste il grave problema dell'aumento esponenziale delle combinazioni per ogni variante introdotta (la fig. 2c mostra come due possibili variazioni portino a quattro caratteri diversi).

Un terzo algoritmo di soluzione affronta il problema in maniera diversa. Premesso che la scacchiera bianca o

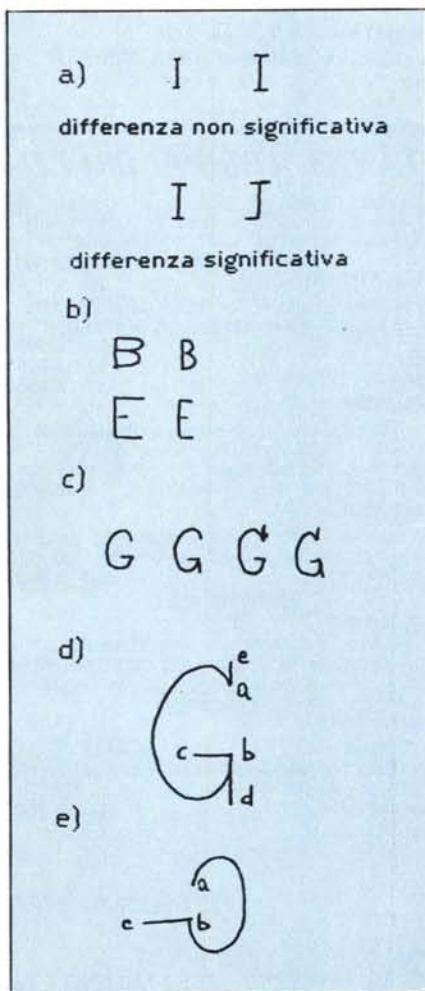


Figura 2

nera resta la stessa degli esami precedenti, il procedimento di analisi viene eseguito non già per confronto diretto, con una serie di pattern precostituiti, ma analizzando la geometria della lettera stessa, che viene rappresentata attraverso una serie booleana di caratteristiche. Ad esempio, la stessa lettera G del precedente esempio, meglio evidenziata nella figura 2d, può essere descritta come una coppia di routine grafiche. La prima, che raccoglie le caratteristiche salienti della figura può essere rappresentata da

ROUTINE arco cerchio {a-b} [almeno 270°]

ROUTINE tratto verticale {b-d} [astina distinguente la C dalla G] mentre la seconda, contenente le strutture opzionali potrebbe essere così articolata:

ROUTINE rapporto ellissoidale: [analizzante il rapporto assiale delle due dimensioni principali, e, in pratica, analizzante se l'arco stesso viene tracciato come di cerchio o di ellisse]

ROUTINE angolo di tracciamento [che può essere reso flessibile stabilendo un range di validità dell'ampiezza dell'angolo stesso]

ROUTINE segmenti opzionali {b-d / a-e} [del tutto accessorie per la com-

prensione e la individuazione univoca della lettera] che, in linguaggio informale può essere tradotta:

«È una lettera G qualunque segno che sia composto da un arco di cerchio, in un intorno di tre quadranti, avente un segmento orizzontale presente all'estremo dell'arco, e possedente due ulteriori segmenti, opzionali...» e così via.

Questo algoritmo, una volta tradotto in programma in un linguaggio dedicato (è più semplice di quanto non sembri), consentirà una maggiore flessibilità degli altri due. È possibile, così riconoscere differenze relative tra le dimensioni delle lettere; ha, inoltre, il pregio di essere relativamente conciso, è facilmente aggiornabile, non dovendosi intervenire su dati memorizzati, ma solo sul programma stesso. Inoltre appare, per la prima volta, la utilizzazione del concetto di generalizzabilità: il programma può essere adattato per riconoscere lettere simili (come la I, la J e la l, o la C e la stessa G). Anche qui, ovviamente esistono problemi; se non si esegue un perfetto progetto delle routine di individuazione, può accadere che la lettera della figura 2e sia scambiata con la G.

Quanto abbiamo finora detto aveva,

solo in parte, lo scopo di evidenziare come è possibile la lettura della parola scritta (anche se la tecnica effettiva si basa proprio su un algoritmo del terzo tipo). Ci interessava invece per evidenziare l'uso delle strutture basilari descritte precedentemente; la fase di completamento dell'algoritmo è infatti passata attraverso queste tre fasi:

— ricerca, destinata a fornire una strada di approccio al problema (inquadramento del carattere in un pattern, fatto comune a tutti e tre gli algoritmi presentati).

— conoscenza del problema, che ci ha permesso di suddividere lo stesso in altri più piccoli (es. lettura delle stringhe binarie, individuazione delle strutture geometriche, come l'arco di cerchio od i segmenti descritti nell'analisi della lettera G, individuazione delle possibilità di sovrapposizione di pattern per caratteri simili, ecc.)

— astazione, in cui si è eseguita una separazione (terzo algoritmo) tra le caratteristiche indispensabili di un carattere e quelle accessorie.

— generalizzazione.

Ogni problema, nel campo della A.I. può essere risolto efficacemente attraverso l'uso di questa tecnica. Avremo modo di vederne, in seguito, numerosi esempi. **MC**

## LE PIÙ GRANDI NOVITÀ DEL 1986 AI PREZZI PIÙ BASSI D'EUROPA

### PERIFERICHE PER QL

**QINTERAM - Espansione interna di memoria da 256 o da 512 Kb con Eprom da 16 Kb.**

Incrementa la memoria del QL a 384 o a 640 Kb, si inserisce all'interno del computer senza effettuare nessuna saldatura, dissaldatura o manomissione, non invalida pertanto la garanzia ed il montaggio è velocissimo ed accessibile a tutti. Il connettore di espansione resta libero e si possono inserire contemporaneamente altre periferiche.

Inoltre sulla stessa scheda è presente una Eprom che aggiunge 16 Kb di nuovi e potenti comandi al QDOS, fra i quali la gestione del RAM come RAM Disk e il multitasking che permette ad esempio di caricare in memoria i 4 programmi della PSION e passare istantaneamente dall'uno all'altro. Dissipazione termica e assorbimento irrisoni grazie all'impiego di integrati con tecnologia TTL C-MOS.

La gestione della RAM è indipendente con conseguente incremento sino al 30-40% della velocità di esecuzione dei programmi.

**QINTERAM 256 Kb L. 199.000**  
**QINTERAM 512 Kb L. 259.000**

**QL DISK - Sistema per floppy drives**

Il sistema è composto da un'interfaccia per floppy drives e da drives da 3,5" e 1 Mb. L'interfaccia progettata anch'essa dall'autore del QDOS Tony Tebby possiede la piena compatibilità con tutto il software e hardware SINCLAIR.

**+ Drive 3.5" + 1 Mb L. 490.000**

**+ Drive 3.5" + 1 Mb L. 250.000**

**QINTERCOM - Interfaccia di comunicazione multifunzione.**

E munita di una uscita Centronics per stampanti-

parallele, più una uscita Cannon a 9 poli per joystick e di una uscita per Modem con possibilità di selezionare la velocità da 75 a 1.200 Baud.

**L. 79.000**

**QL CARPROM**

Cartuccia porta EPROM completa di EPROM da 128 Kb. Si inserisce nella porta ROM esterna.

**1 L. 15.000**

**10 L. 12.000**

**100 L. 8.900**

**QL PROM**

Elaboratissimo e professionale programmatore di EPROM per il QL che si inserisce nel connettore di espansione. Sistema operativo residente su EPROM per una rapidissima programmazione. **L. 300.000**

**QL UVPROM**

Compatto cancellatore di EPROM a UV. Cancella fino a tre EPROM contemporaneamente. Timer automatico da 15". **L. 110.000**

### SOFTWARE PER QL

**QL MOON**

Superbo e professionale programma sull'astrologia lunare, informazioni e statistiche su oltre 300 punti, banca dati di oltre 50 Kb. Gestito da menu con grafica eccezionale.

**Su cartuccia L. 34.000**

**3D SLIME**

Gioco con grafica tridimensionale, scritto dagli stessi autori di M-PAINT.

**Su cartuccia L. 34.000**

**QL PLAGIO**

Professionale copiatore universale per QL. Copia

sette per settore qualsiasi tipo di cartuccia. Scritto completamente in L/M. Funziona con qualsiasi tipo di programma e versione ROM. Il programma viene venduto esclusivamente per proprio uso personale.

**Su EPROM L. 90.000**

**Cartucce per -**

**Microdrive 1 L. 5.900**

**50 (cad.) L. 5.400**

**100 (cad.) L. 4.900**

**Floppy disk 3,5" DD - SD**

**135 TPI Basf L. 7.000**

**Floppy disk 3,5 DD - DD - 135**

**TPI Panasonic L. 9.000**

**Coppia integrati versione JS L. 35.000**

### ATTENZIONE PER TUTTI I POSSESSORI DELLO SPECTRUM

#### Interfaccia Duplex

Permette di duplicare e di trasferire su: NASTRO - MICRODRIVE - FLOPPY DISK qualsiasi tipo di programma commerciale oggi esistente sul mercato; TURBO - TURBO-PULSANTI - MAXI - CON L/M NEL LOADER, ecc.

Semplicissima da usare, si collega l'interfaccia al connettore di espansione, al termine premendo un tasto di break si ottiene una copia a velocità normale che si carica in maniera autonoma senza interfaccia collegata.

I possessori dell'interfaccia 1 potranno scegliere l'opzione microdrive al momento del trasferimento ed ottenere su cartidge una copia del programma preferito.

Il prezzo dell'INTERFACCIA DUPLEX, con il manuale e le spese di spedizione contrassegno è di **L. 95.000**

**ATTENZIONE I PREZZI SOVRAESPONDI SONO IVA COMPRESA**

**COMPUTER CENTER**

PER INFORMAZIONI E/O ORDINAZIONI

VIA FORZE ARMATE, 260 - 2052 MILANO - TEL. 02-4890213

# Nella giungla dei compatibili oggi c'è un nuovo re:

# Quasar



## UN RE IN PRIMO PIANO

Il prezzo di un compatibile è importante. Ma noi vi garantiamo un prodotto il cui rapporto tra prezzo, qualità ed affidabilità è il migliore in assoluto.

## UN RE CHE NON TEME CONFRONTI

È MS DOS compatibile (sa utilizzare tutti i maggiori programmi esistenti sul mercato dei personal computers). Possiamo paragonarlo al PC XT, ma con qualche caratteristica in più. Raggiunge la massima espansione di memoria - 640 Kbytes - direttamente su piastra madre e può passare dal clock standard di 4 MHz a quello, molto più redditizio, di 7 MHz, tramite un semplice comando da tastiera.

## UN RE UNA DINASTIA

Ecco i diversi allestimenti che differenziano le macchine:

- con hard disk da 10 o da 20 Mbytes
- con la scheda per la rete locale
- con il modem completamente automatico o con il modemphone, che comprende anche l'apparecchio telefonico
- con la scheda color/graphic o la hercules o la monochrome
- con il mouse
- con l'A/D D/A converter
- con la scheda 8255 per 48 linee programmabili di I/O oppure senza drivers per applicazioni diverse
- con monitor monocromatico da 12 o 15 pollici o con quelli a colori da 14 pollici in media e alta risoluzione
- con stampante da 80 col/100 cps sino al top della gamma, la stampante laser.

## UN RE PARTICOLARMENTE DOTATO

La dotazione standard è realizzata per soddisfare anche i più esigenti: due floppy disk drivers (256 Kbytes di memoria), scheda color/graphic, porta parallela e cavo per stampante. Ma l'ulteriore crescita del vostro reale amico dipende soltanto da voi.

## UN GRANDE RE UN BEL RISPARMIO

Sua maestà ha un'altra grande dote, l'economicità. Interpellateci perché da noi i fatti non sono parole.

**Quasar**  
QUASAR SRL - Via Diagonale 319 - 13050 Pratrivero (VC)  
Tel. (015) 778804 - Tlx. 211401 MILFIL I