

Introduzione al Soft Computing

**Algoritmi Genetici
Reti Neuro-Fuzzy
Conclusioni**



Cristian Randieri



www.intellisystem.it

Algoritmi genetici

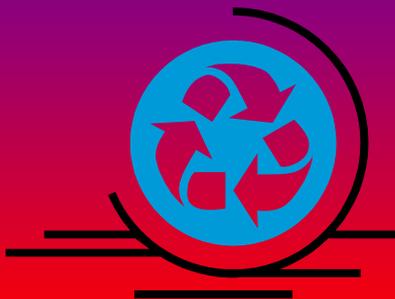
Gli algoritmi genetici (GA = Genetic Algorithms) furono proposti inizialmente da J.H. Holland nel 1975. Da allora sono stati oggetto di molti studi e recentemente si è iniziato ad utilizzarli anche per applicazioni pratiche.



Essi costituiscono un modello computazionale idealizzato dell'evoluzione darwiniana, basata sui due principi della variazione genetica e della selezione naturale. Gli individui di una popolazione hanno patrimoni genetici più o meno differenti (variazione genetica). Ad ogni generazione, coppie di individui si uniscono per generare altri individui, che saranno dotati di un patrimonio genetico risultante dalla combinazione dei patrimoni dei genitori.



L'adattamento degli individui all'ambiente ("fitness") dipende dal loro patrimonio genetico. Gli individui con maggiore fitness sono mediamente favoriti (selezione naturale) rispetto agli altri: essi tendono a vivere di più, a riprodursi di più, trasmettendo alla discendenza parte del loro patrimonio genetico, già rivelatosi competitivo.



Negli algoritmi genetici il patrimonio genetico di un individuo è rappresentato generalmente da una stringa di n simboli o geni. Ogni gene è un simbolo tratto da un alfabeto di p simboli; in particolare, i simboli possono essere 0, 1 ($p = 2$), avendosi allora stringhe binarie. Si possono pertanto avere individui diversi che, nel caso binario, sono vertici di un iper-cubo di n dimensioni.



Bontà delle soluzioni

La bontà delle soluzioni è misurata dalla fitness delle stringhe. Per ottenere tale evoluzione, gli algoritmi si valgono di operazioni genetiche, ciascuna eseguita su uno o più individui, scelti con criteri probabilistici.



Operazioni genetiche più comuni

Riproduzione



In una popolazione di M individui (generazione attuale) se ne scelgono (con probabilità proporzionale alla loro fitness) solo $m < M$, che costituiscono la generazione successiva.

Mutazione



Opera su una sola stringa A nel modo seguente: dopo aver scelto, in modo casuale, un indice k si sostituisce al simbolo un nuovo simbolo tratto in modo random dall'alfabeto dei geni. Si ottiene così la trasformazione.

001.011.101  010.011.101
↑

Cross-over

010.001.111
↑
010.100.101  010.100.111

Questa operazione coinvolge due stringhe genitrici, Dopo aver scelto in modo random un indice k, si effettua uno scambio di geni tale da produrre le stringhe figlie. Una variante di questa operazione sceglie in modo random due indici k ed l ed effettua lo scambio

Algoritmo genetico elementare

- . Si ha una popolazione U di m stringhe, ordinate in funzione della loro fitness.
- . Si sceglie con probabilità P_{mut} (molto piccola) un gene di una stringa e si opera la mutazione.
- . Si scelgono con probabilità . due stringhe e si opera il cross-over.
- . Si riordinano le stringhe in funzione della loro fitness e se ne scelgono m , con probabilità proporzionale alla fitness, costituendo così la nuova popolazione U .
- . Si ritorna al punto 1 fino a che non è stato soddisfatto un predefinito criterio di stop (per esempio un numero massimo di iterazioni).



Soluzione.....



Il problema sarà risolto se almeno una delle stringhe della popolazione finale costituisce una soluzione vicina all'ottimo assoluto. Sostanzialmente l'algoritmo genetico è un metodo di ricerca di soluzioni ottimali nel "paesaggio" multidimensionale della fitness. Tale metodo è considerato più efficiente di quello denominato "hill climbing" che presenta l'inconveniente di offrire soluzioni che sono minimi relativi e non assoluti. Per l'algoritmo genetico la mutazione introduce, così come il coefficiente di momentum per le reti neurali, un disturbo che tende ad evitare la convergenza prematura su un ottimo relativo.

Schemi



E' importante anche rilevare che, nell'ambito di una stringa, possono emergere, in un certo momento, determinate sequenze di geni che le conferiscono una fitness superiore alla media.

Un esempio di schema, in una stringa binaria con $n=13$ è

01###100#1101

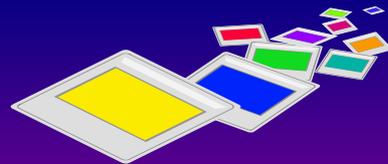
dove # indica un qualunque simbolo legale.

Si definisce "ordine" di uno schema il numero di simboli definiti in esso (9 nel precedente esempio).

Si ritiene che l'operazione di cross-over faciliti la combinazione di schemi buoni di ordine basso ("building blocks") in schemi migliori di ordine elevato. Un teorema dovuto ad Holland assicura, sotto determinate condizioni, la convergenza dell'algoritmo.

Gli schemi con fitness superiore alla media tendono a diffondersi esponenzialmente nella popolazione.

...continua



Tuttavia, se esistono soluzioni diverse, ugualmente buone ma con schemi contrastanti, l'operazione di cross-over può produrre, da genitori buoni, ibridi cattivi. Generalmente nel passaggio da una generazione all'altra che, come abbiamo visto, tende a riprodurre gli m individui migliori, si preferisce comunque copiare l'individuo migliore della generazione precedente, per non rischiare la perdita del suo prezioso patrimonio genetico.

Un esempio

Per risolvere un problema mediante l'algoritmo genetico è necessario formalizzare le ipotetiche soluzioni con stringhe di simboli aventi la stessa lunghezza. Come primo esempio consideriamo il semplice problema di trovare la regola che sintetizza i casi accettabili della tabella

Caso	Prezzo	Qualità	Accettabilità
1	450	3	si
2	500	3	si
3	600	3	no
4	450	2	si
5	500	2	si
6	600	2	no
7	450	1	no
8	500	1	no
9	600	1	no

La regola cercata sarà del tipo $(Op_1 VP) Op_2 (Op_1 VQ)$ con la seguente codifica binaria

Op_1 = operatore 1 (0 = maggiore o uguale, 1 = minore o uguale)

VP = valore prezzo (01 = 450, 10 = 500, 11 = 600, 00 non usato)

Op_2 = operatore 2 (0 = AND, 1 = OR)

VQ = valore qualità (01 = 1, 10 = 2, 11 = 3, 00 non usato)

Per esempio la regola 0111101, una stringa di 7 bit equivale a
(Prezzo 600) OR (Qualità 1)

Fitness



La fitness di una regola è uguale al rapporto tra il numero di casi accettabili che esso soddisfa e il numero totale di tali casi (la regola precedente ha fitness nulla). Partendo allora da una popolazione sufficientemente grande di stringhe (nel nostro caso ne sono possibile 128), applicando l'algoritmo genetico, si giunge, in un ragionevole numero di generazioni, alla migliore soluzione che ha fitness 1 ed è

1100010=(Prezzo 500) OR (Qualità 2)



Per completare la breve trattazione sugli algoritmi genetici occorre sottolineare che esistono una serie di operatori genetici non standard che permettono di migliorare le prestazioni dell'algoritmo genetico sia in termini di velocità di convergenza che in termini di accuratezza della soluzione.



Sistemi Intelligenti

Expert Systems

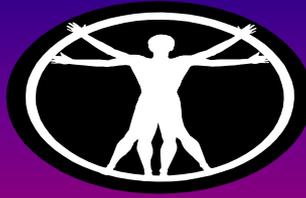
Fuzzy Systems

Artificial Neural Network

Hybrid Systems



Expert Algorithms



- *Alcuni ricercatori (soprattutto dell'area informatica) ritengono che i sistemi esperti, costituiti da una base di regole e da un meccanismo di inferenza siano simili ai metodi umani di risoluzione dei problemi.*



IE
Click!

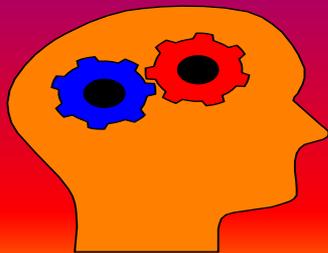


Nell'area dei sistemi di controllo il controllo esperto può essere applicato ai sistemi di controllo adattativo basato sulla conoscenza, alla diagnosi dei guasti, allo scheduling ed al planning.

Fuzzy Systems

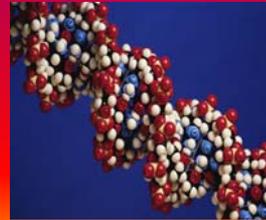
A differenza dei tradizionali sistemi esperti, le regole fuzzy, essendo una conoscenza di tipo strutturato, si prestano ad una precisa codifica in forma numerica.

.....Sin da quando è stata proposta la teoria della fuzzy logic (Zadeh 1965) i sistemi fuzzy sono stati visti come una possibile strategia di sintesi, in particolare per sistemi complessi, non lineari con imprecise od incerte conoscenze sul sistema.



Reti neurali artificiali

- *Una rete neurale artificiale può essere assimilata ad un modello dinamico a larga scala, non lineare, consistente di un certo numero di semplici elementi di calcolo (nodi o neuroni) i quali sono tra loro interconnessi con intensità tra le connessioni aggiustabili mediante appropriati algoritmi (es. backpropagation)*



I benefici del controllo fuzzy

- *Considerando le applicazioni esistenti della logica fuzzy, che spaziano dai piccoli sistemi domestici ai sistemi di controllo a larga scala, i vantaggi possono essere così classificati:*
 - ◆ *Implementazione di sistemi con un elevato grado di automazione*
 - ◆ *Robustezza rispetto a nuove situazioni*
 - ◆ *Riduzione dei tempi di sviluppo e di manutenzione*
 - ◆ *Minor costo di sviluppo*



Anche la logica Fuzzy ha i propri limiti

In alcuni articoli la logica fuzzy
viene presentata come una tecnica
di controllo FAVOLOSA. Le affermazioni più comuni sono:

**I controllori fuzzy consentono un
più elevato grado di interazione in
sistemi complessi e ben strutturati**

sono più robusti



*Bisogna però ricordarsi che non
sempre si possono ottenere i risultati
voluti, di conseguenza è più proficuo
utilizzarli laddove non si trovano
soluzioni ottimali con i metodi
classicamente usati*



Reti Neuro-fuzzy

Un'altra caratteristica
del modo umano di
ragionare (oltre al parallelismo)
è la capacità di trattare
informazioni "vaghe"



La fuzzy logic così come è stata concepita, esula completamente dal concetto di apprendimento automatico che invece è intrinseco, per esempio, alle reti neurali. Quindi se si rimane nell'ambito della teoria fuzzy originaria, non è possibile utilizzare la fuzzy logic in tutti quei campi applicativi nei quali non si abbiano già a disposizione degli esperti che formulino le regole. Per diversi anni, questa caratteristica negativa della fuzzy logic è stato l'aspetto più fortemente limitante per la sua diffusione in molti campi applicativi.

Il sorgere delle prime tecniche di apprendimento automatico sviluppate apposta per la fuzzy logic è da considerarsi una pietra miliare nel campo della ricerca sul soft computing. Attualmente una percentuale elevata dei lavori nell'ambito di tale disciplina è basata sulla commistione fra le varie tecniche. Questa interazione è ormai così stretta che, talvolta, appare veramente difficile individuare quali siano le reali frontiere fra le reti neurali, la logica fuzzy e gli algoritmi genetici.



Per realizzare dei sistemi che riflettessero da un lato il parallelismo di elaborazione e dall'altro la capacità di elaborare informazioni vaghe sono state create le reti NEURO-FUZZY

Teoria delle reti Neuro-fuzzy

Le FNN sono particolari reti neurali nelle quali i pesi delle connessioni rappresentano i parametri di un insieme di regole fuzzy.

In questa sede per brevità si riporta esclusivamente la struttura di una rete neuro-fuzzy di tipo I, caratterizzata dall'aver i conseguenti costanti, cioè le regole hanno forma del tipo:

R^i : If x_1 is $A_{i1}(x_1)$ and x_2 is $A_{i2}(x_2)$ then $y=wf_i$ ($i=1,2,..n$)

$$y^* = \frac{\sum_{i=1}^n \mu_i \cdot wf_i}{\sum_{i=1}^n \mu_i}$$

dove:

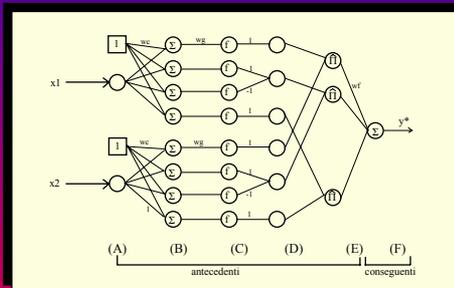
- R^i rappresenta l' i -esima regola, A_{i1} e A_{i2} sono le variabili fuzzy della premessa;

- wf_i è una costante, n è il numero di regole;

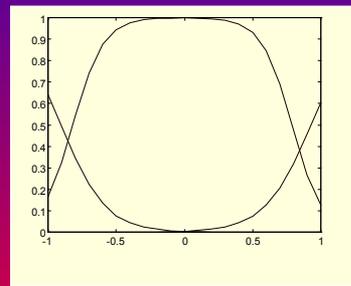
- y^* è il valore defuzzificato dell'uscita.



Struttura di una rete Neuro-fuzzy



Struttura di una rete neuro-fuzzy



Funzioni di membership



La rete in figura fa riferimento a una FNN con due ingressi (x_1, x_2), un'uscita (y^*) e tre funzioni di membership in ogni premessa. Le notazioni $w_c, w_g, w_f, 1, -1$ tra le unità sono i pesi di connessione. Le unità con un simbolo 1 sono bias con uscita unitaria. Le relazioni ingresso-uscita delle unità con i simboli f, Σ, Π sono definiti dalle seguenti equazioni:

$$\begin{array}{ll}
 I_j^{(n)} = \sum_k w_{jk}^{(n,n-1)} O_k^{(n-1)} & f : O_j^{(n)} = \frac{1}{1 + \exp(-I_j^{(n)})} \\
 \Sigma : O_j^{(n)} = I_j^{(n)} & I_j^{(n)} = \prod_k w_{jk}^{(n,n-1)} O_k^{(n-1)} \\
 \Pi : O_j^{(n)} = I_j^{(n)} & \bar{H} : O_j^{(n)} = \frac{I_j^{(n)}}{\sum_k I_k^{(n)}}
 \end{array}$$

Dove:

$I_j^{(n)}$ e $O_j^{(n)}$ sono rispettivamente l'ingresso e l'uscita dell'unità j -esima dello strato n ;

$w_{jk}^{(n,n-1)}$ è il peso della connessione tra la k -esima unità dello strato $(n-1)$ -esimo e la j -esima unità dello strato n -esimo.

Il processamento delle informazioni da parte di una FNN prevede due fasi distinte:

**il calcolo dell'attivazione degli antecedenti
il calcolo dell'attivazione del conseguente.**

Nella topologia mostrata, la parte delle premesse è costituita dagli strati da A ed E mentre i gradi di verità delle premesse sono calcolati negli strati da A a D.

I pesi w_c e w_g rappresentano i parametri che determinano rispettivamente la posizione centrale ed il gradiente della funzione sigmoidale nelle unità dello strato C.

Inizializzando opportunamente i pesi le funzioni di membership $A_{ij}(x_j)$ possono essere allocate nell'universo del discorso come riportato ad esempio nella figura precedente.

In questo tipo di implementazione di una base di regole fuzzy con reti neurali i connettivi AND e OR sono sostituiti rispettivamente dagli operatori di prodotto e somma.



Pertanto l'attivazione della regola corrispondente a ciascun sottospazio è data dai prodotti dei gradi di appartenenza degli antecedenti presenti nella regola, ai corrispondenti fuzzy set.

Questa operazione viene realizzata dalle unità dello strato E.

$$\text{inputs : } \mu_i = \prod_j A_{i,j}(x_j) \quad \text{outputs : } \bar{\mu}_i = \frac{\mu_i}{\sum_k \mu_k} \quad (3)$$

dove μ_i è l'attivazione della regola i -esima e $\bar{\mu}_i$ è il valore normalizzato di μ_i .

La fase di defuzzificazione viene normalmente eseguita con il metodo del centroide definito dall'equazione d'uscita delle (3).

Il processo di identificazione di un modello fuzzy può essere suddiviso nelle due fasi di identificazione dei parametri strutturali (di topologia) della rete e, successivamente, di identificazione delle premesse e dei conseguenti, con apprendimento del tipo back-propagation modificata sulla base delle (2). La prima fase viene normalmente eseguita attraverso una procedura di fuzzy clustering, ovvero attraverso il metodo di Sugeno.

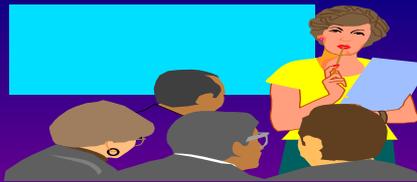
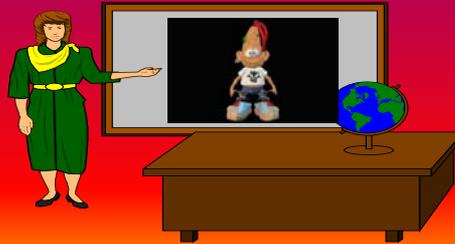
Principali Conferenze su Fuzzy Logic e Neurofuzzy del 1996



- *EUFIT 96, Aachen, September 1996*
- *FUZZ-IEEE 1996, New Orleans, Sep. 1996*
- ▲ *World Congress on Neural Networks, San Diego, September 1996*
- ▼ *IEEE Int. Conf. on System, Man and Cybernetics, Beijing, China, October 1996*
- ◆ *Intenational Conference on Industrial Fuzzy Control and Intelligent Systems, Cancun, Mexico, Nov. 1996.*

Temi di una tipica conferenza sulla Fuzzy Logic nella scienza

- *Gli algoritmi basati su logica Fuzzy sono usati in vasti campi della ricerca quali: applicazioni spaziali, automazione, predizione finanziaria, elaborazione dei segnali audio e video. Sono stati pertanto introdotti nuovi tipi di fuzzy logic toolboxes e software di sviluppo disponibili nel mercato.*



- *Gli obiettivi sono quelli di focalizzare l'attenzione degli utenti, degli sviluppatori, delle industrie e delle università sulle ultime novità in questa area. Con ciò si vuole delineare lo stato d'arte corrente nelle applicazioni reali basate sulla fuzzy logic.*

EUFIT '96: topics

- *To Basics of Fuzzy Set Theory and Uncertainty Modeling*
- *Neural Networks - Basics and Applications*
- *Genetic Algorithms and Evolutionary Computing*
- *Fuzzy Sets in Artificial Intelligence*
- *Neural Fuzzy Systems and Learning*
- *Fuzzy Databases and Information Retrieval*
- *Fuzzy Control - Methodology and Applications*
- *Decision Support Systems*
- *Data Analysis and Signal Processing*
- *Pattern Recognition and Image Processing*
- *Applications in Process Industry and Production Planning*
- *Financial Engineering and Forecasting*
- *Software, Hardware, and Standards*
- *Panel Discussions on Intelligent Applications in Industry*



Riviste con ampi spazi sul Soft Computing

- *IEEE Transaction on Fuzzy Systems*
- *IEEE Transaction on Neural Networks*
- *IEEE Control System Magazine*
- *IEEE Transaction on System Man and Cyb.*
- *IEEE Transaction on Control System Tech.*
- *Control Engineering Practice, Pergamon Press*



