

ALGORITMI DI
OTTIMIZZAZIONE E
APPROSSIMAZIONE
(SOFT COMPUTING)

A CURA DI:
ALESSANDRO FILISETTI

SOMMARIO

- Algoritmi di ottimizzazione
- Algoritmi di ottimizzazione e di ricerca
- Algoritmi di approssimazione, ricerca, simulazione (soft computing)
 - Algoritmi genetici
 - Reti neurali
- Esempi di applicazione

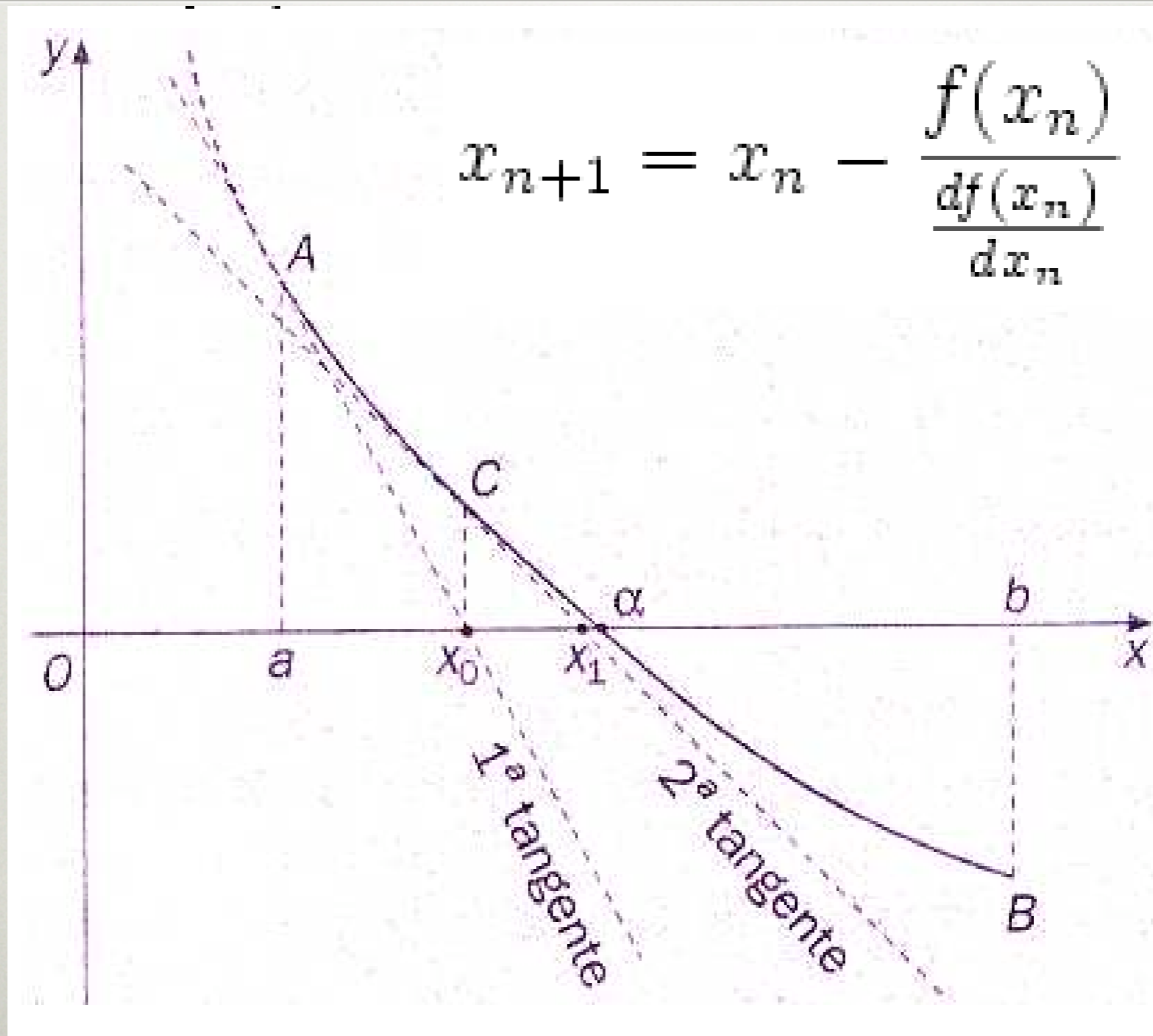
OTTIMIZZAZIONE

L'**ottimizzazione** è la branca della matematica applicata che studia teoria e metodi per la ricerca dei punti di **massimo** o **minimo** di un **modello matematico** che traduce in termini matematici un dato problema (non occupandosi quindi direttamente di come tali modelli siano stati costruiti). L'ambito di ricerca privilegiato dell'ottimizzazione sono i modelli esprimibili in termini di **funzioni** in più variabili, nei quali i punti di **ottimo** vengono ricercati ponendo anche vincoli di qualitativi espressi in termini di **derivate successive**.

ALGORITMI DI OTTIMIZZAZIONE

Gli algoritmi di ottimizzazione fanno parte di quella classe di funzioni aventi il compito di trovare il minimo o il massimo di una funzione per la maggior parte dei casi non lineare.

ESEMPIO: L'ALGORITMO DI NEWTON



ALGORITMI DI OTTIMIZZAZIONE Vs. ALGORITMI DI RICERCA

Un qualunque algoritmo di ottimizzazione può essere usato per fare ricerca e un qualunque algoritmo di ricerca può essere usato per fare ottimizzazione. Le tecniche di ricerca però non puntano ad ottimizzare ma cercano e basta. Se un algoritmo dovesse cercare la pallina verde in un insieme di palline analizzerebbe la differenza della luce riflessa e cercherebbe il minimo della differenza tra lo spettro memorizzato in esso e quello osservato, quindi ecco un algoritmo di ottimizzazione che va a risolvere un problema di ricerca (es. riconoscimento di immagini, impronte digitale ...).

TECNICHE DI OTTIMIZZAZIONE

- **Tecniche dirette:** Fanno riferimento esplicito alle funzioni e al problema su cui stiamo indagando. Funzionano bene quando il punto di partenza è vicino alla soluzione, è fortemente condizionato dalle condizioni iniziali;
- **Tecniche enumerative:** Si basa sul costruire schemi regolari (es. griglia). Aumenta di efficienza all'aumentare del numero di passi;
- **Tecniche casuali:** invece di seguire step regolari si estraggono a caso i punti di ricerca. Hanno più o meno le stesse proprietà di quelle enumerative ma sono più efficienti dal punto di vista computazionale. I fenomeni casuali possono essere comunque guidati ed in questo modo prendono il nome di *tecniche casualizzate*.

TECNICHE DI OTTIMIZZAZIONE: SCHEMA DI EFFICIENZA

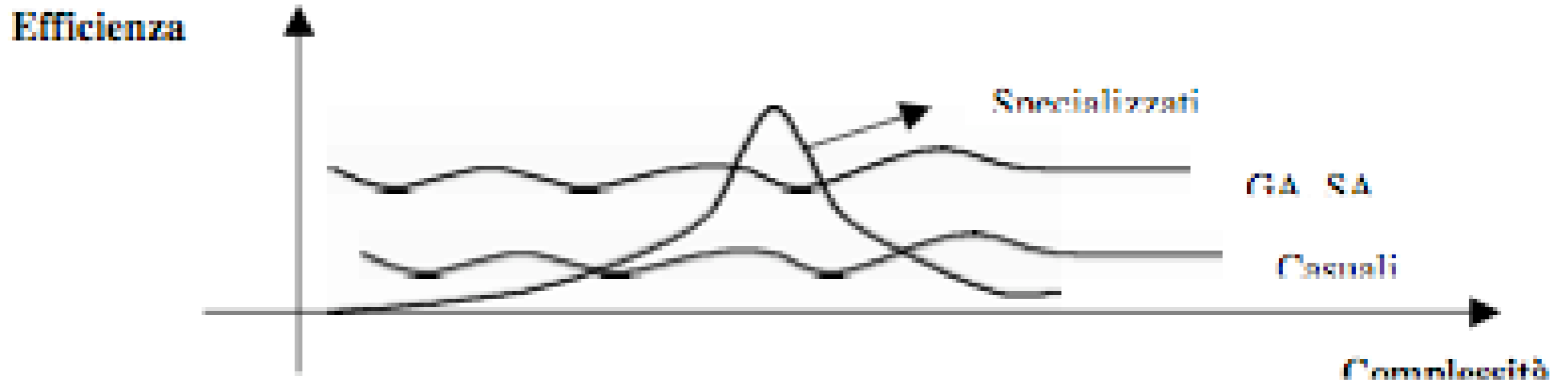
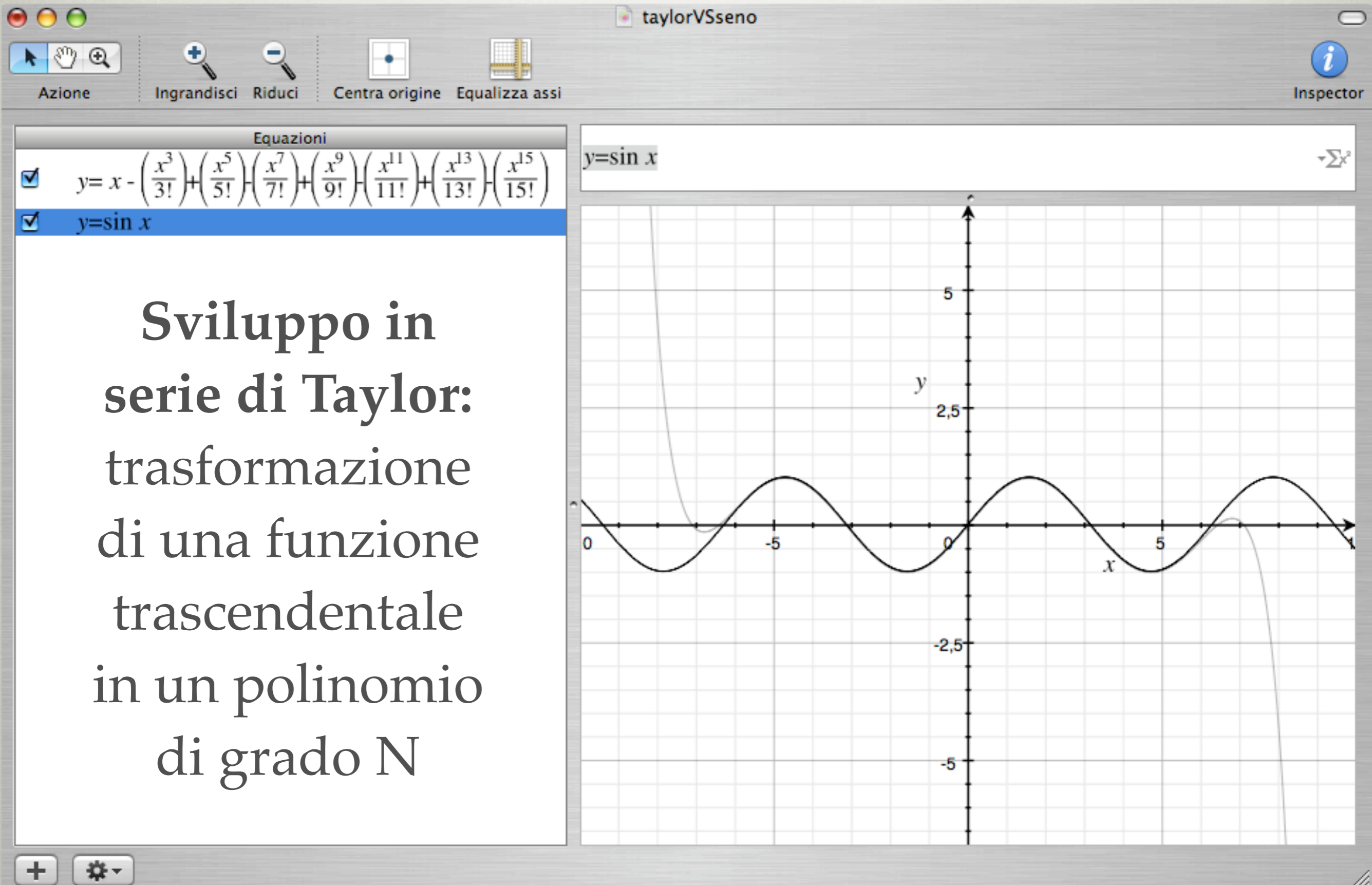


Fig. 7.7. Efficienza vs Complessità per alcune classi di algoritmi di ottimizzazione.

AGIRE DI FANTASIA

1. Quando ci troviamo di fronte ad algoritmi dobbiamo agire di fantasia, possono essere nati per risolvere una certa classe di problemi ma niente ci vieta di utilizzarli per risolvere nuovi tipi di problemi.
2. Pensiamo sempre a come avremmo risolto noi il problema. Dobbiamo pensare a tecniche biomorfe legate ad aspetti biologici, come un organismo affronterebbe questo problema? Pensando in questi termini ci avviciniamo al concetto di intelligenza artificiale.

ESEMPIO DI APPROSSIMAZIONE



Sviluppo in serie di Taylor: trasformazione di una funzione trascendentale in un polinomio di grado N

“SOFT COMPUTING”

Le discipline appartenenti al campo del “*Soft Computing*” nascono con l'intento di fornire un insieme di tecnologie più flessibili, più idonee per affrontare i problemi del mondo reale, dove niente è assolutamente certo, ma spesso si hanno a disposizione molteplici possibili soluzioni. Attraverso le metodologie classiche, la formulazione dei problemi porta a definire sistemi perfettamente modellati o controllati, la cui configurazione risulta essere un procedimento molto complesso, data la mancanza di una informazione certa, precisa e completa. Da questa constatazione emerge allora chiaramente la necessità che ha portato all'affermazione di un diverso sistema di approcci, in un certo senso approssimati, capaci di trattare l'informazione imperfetta e incompleta ottenibile dal mondo reale.

ALGORITMI EVOLUTIVI

ALGORITMI EVOLUTIVI

Gli Algoritmi Genetici costituiscono una classe di tecnologie di ricerca probabilistica che si ispirano all'evoluzione biologica. L'algoritmo fornisce uno strumento per l'interrogazione di insiemi molto estesi di dati e per analizzare relazioni funzionali complesse. Il processo di ricerca si basa su una versione simulata dell'evoluzione in senso darwiniano, nella quale una popolazione di soluzioni candidate vengono manipolate e condizionate, per mezzo di una strategia artificialmente generata di sopravvivenza dei soggetti più adatti dal punto di vista evolutivo.

ALGORITMI EVOLUTIVI

CLASSI DI APPLICAZIONE

- Elevato numero di soluzioni possibili
- Elevato numero di variabili interessate
- Elevato grado di complessità
- Soluzioni di tipo discreto o comunque discretizzabili

ALGORITMI EVOLUTIVI

CARATTERISTICHE

- Fanno parte della classe degli algoritmi casualizzati (evolvono in maniera stocastica seguendo però una funzione obiettivo);
- Riescono ad esplorare tutto lo spazio delle soluzioni;
- Indipendenza dal fenomeno di path dependency;
- Lavorano in parallelo e non in serie

ALGORITMO EVOLUTIVO

I VARI STEP

- Definire lo spazio di ricerca
- Decidere il sistema di codifica / decodifica
- Stabilire la struttura del cromosoma
- Impostare i parametri genetici
- Costruire una popolazione iniziale composta da un certo numero di individui soluzioni (codificate)
 - decodificare il valore della stringa dell'individuo e valutare quale è la soluzione corrispondente a quell'individuo
 - Valutazione del **fitness value** per ogni soluzione

STRUTTURA LOGICA DI UN ALGORITMO GENETICO

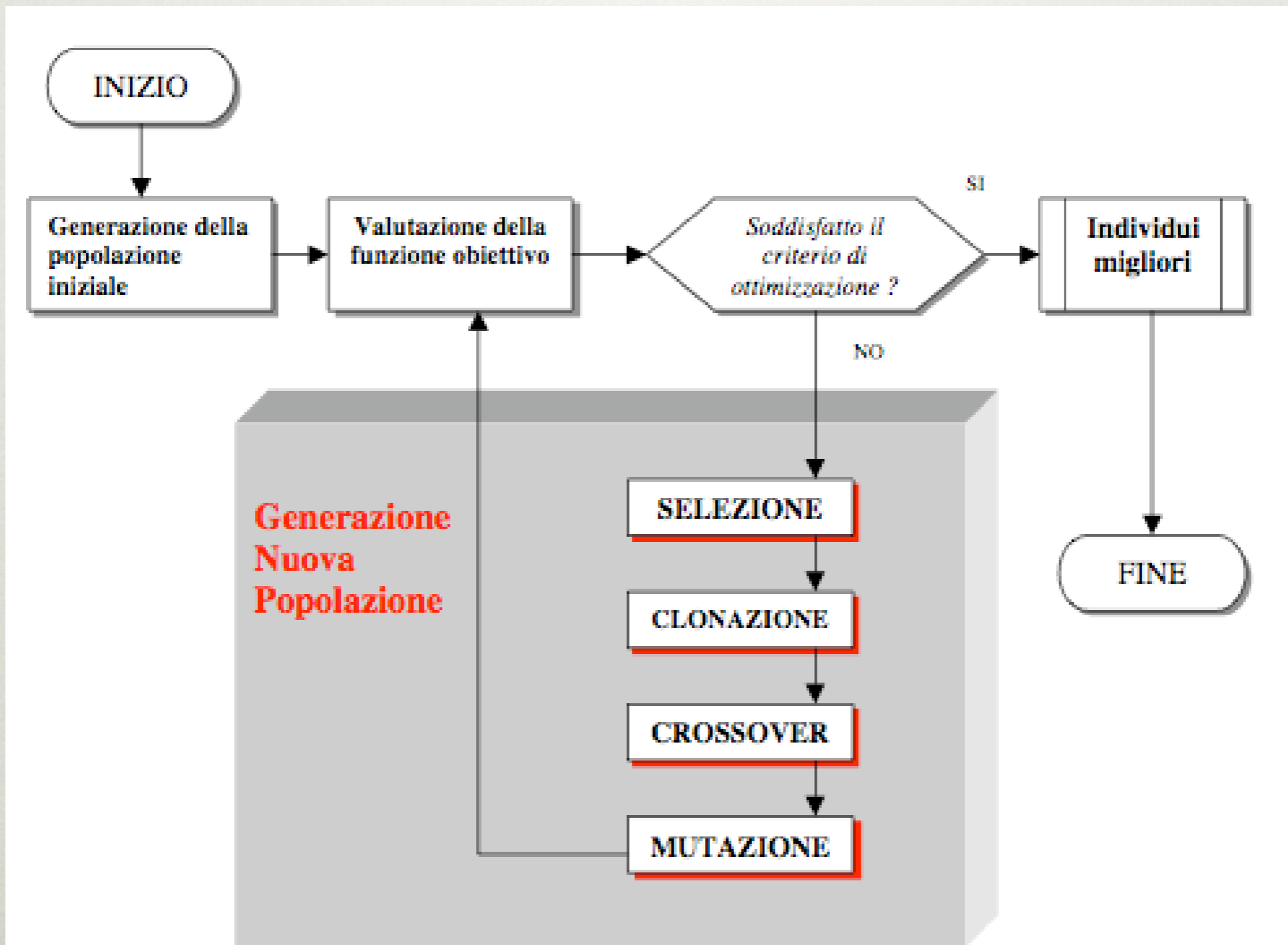
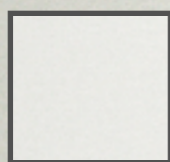


Fig. 4.3. Struttura logica di un algoritmo genetico.

STRUTTURA DI UN ALGORITMO GENETICO

Cromosoma (individuo Soluzione)



Gene (caratteristica dell'individuo)

STRUTTURA DI UN ALGORITMO GENETICO

Generazioni

Durata dell'algoritmo

STRUTTURA DI UN ALGORITMO GENETICO

Funzione Obiettivo (fitness function)

Ogni cromosoma (individuo soluzione) viene valutato secondo una “funzione di fitness” che indica il suo grado di adattamento e che costituisce il criterio in base al quale determinare la probabilità di sopravvivenza dell’individuo nella generazione successiva. La ricerca dell’individuo migliore sarà fatta cercando il cromosoma con fitness maggiore.

STRUTTURA DI UN ALGORITMO GENETICO

Principali Operatori Genetici

SELEZIONE

Gli individui vengono ordinati in base alla loro fitness, i migliori (% sul totale) verranno selezionati per partecipare alla creazione della generazione successiva.

STRUTTURA DI UN ALGORITMO GENETICO

Principali Operatori Genetici

CROSSOVER

Effettua l'incrocio tra i geni di due individui genitori finalizzato alla creazione di due individui figli.

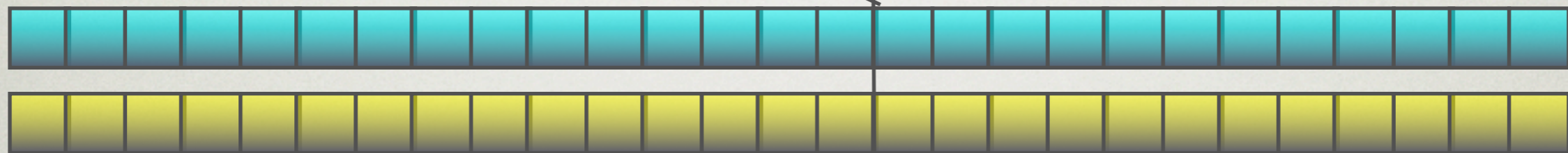
PERMETTE DI SPOSTARSI ALL'INTERNO DELLO SPAZIO DELLE SOLUZIONI IN BASE ALLE INFORMAZIONI PASSATE

STRUTTURA DI UN ALGORITMO GENETICO

Principali Operatori Genetici

Single Point Crossover

Punto di taglio



Applicazione dell'operatore



STRUTTURA DI UN ALGORITMO GENETICO

Principali Operatori Genetici

Multiple Point Crossover



STRUTTURA DI UN ALGORITMO GENETICO

Principali Operatori Genetici

MUTAZIONE

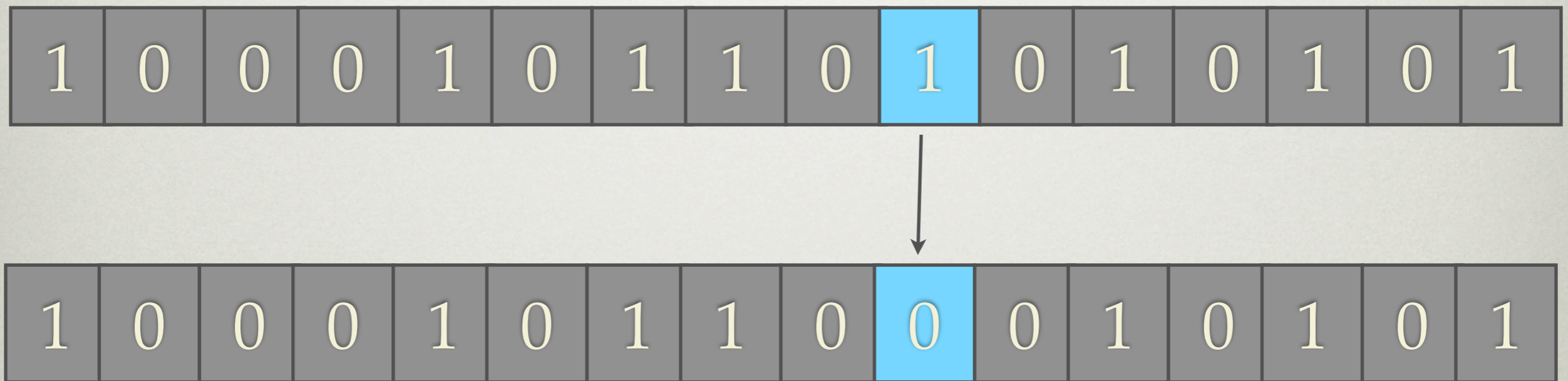
Secondo una certa probabilità modifica in maniera casuale uno dei geni del cromosoma

PERMETTE DI SALTARE DA UN PUNTO ALL'ALTRO DEL
DOMINIO DELLE SOLUZIONI. RAPPRESENTA
L'INNOVAZIONE

STRUTTURA DI UN ALGORITMO GENETICO

Principali Operatori Genetici

MUTAZIONE



STRUTTURA DI UN ALGORITMO GENETICO

Principali Operatori Genetici

ELITISMO (CLONAZIONE)

I migliori individui passano automaticamente alla generazione successiva.

Attenzione: può portare ad una prematura conversione dell'algoritmo verso una soluzioni sub-ottimale

STRUTTURA LOGICA DI UN ALGORITMO GENETICO

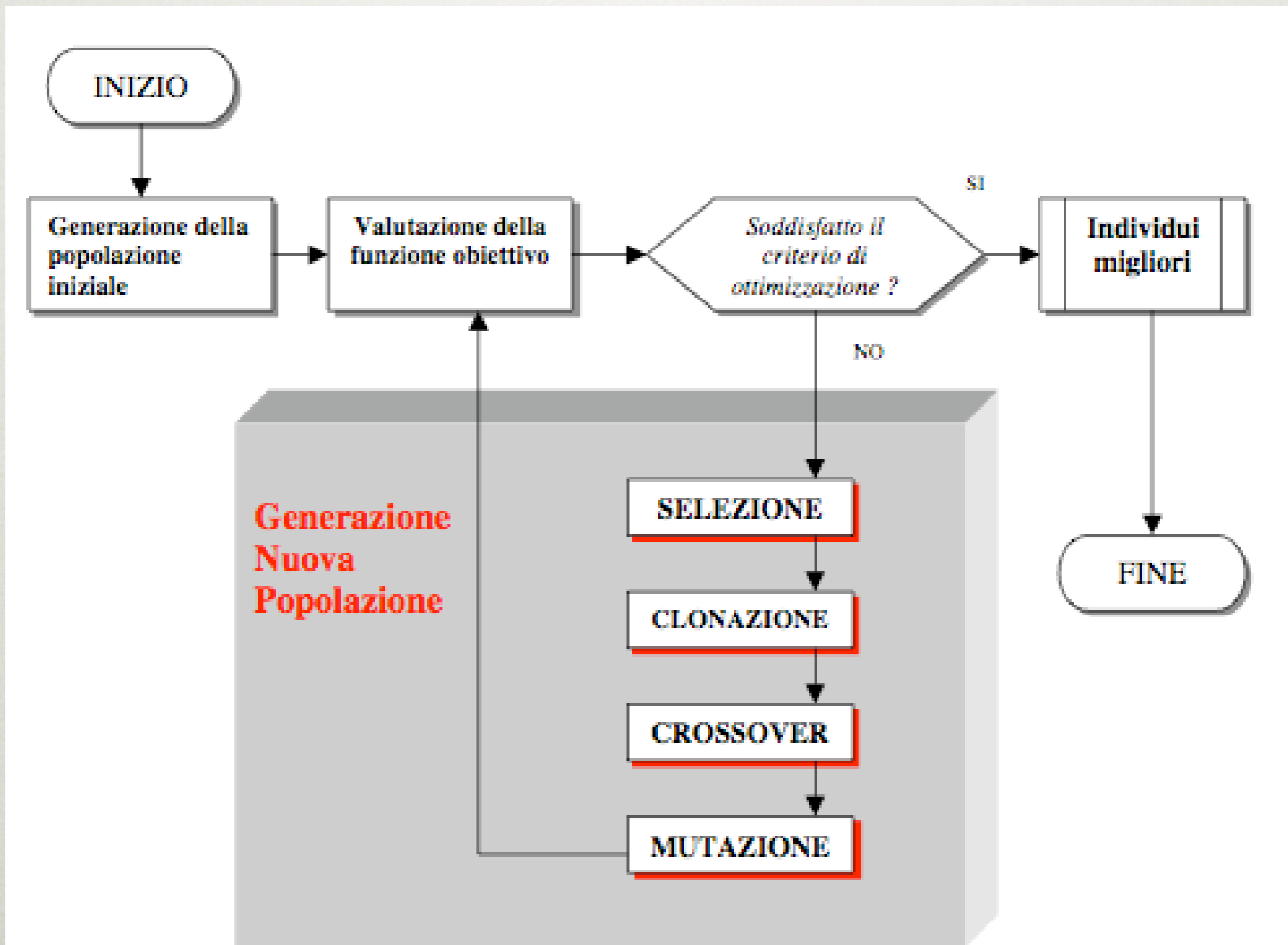


Fig. 4.3. Struttura logica di un algoritmo genetico.

RETI NEURALI

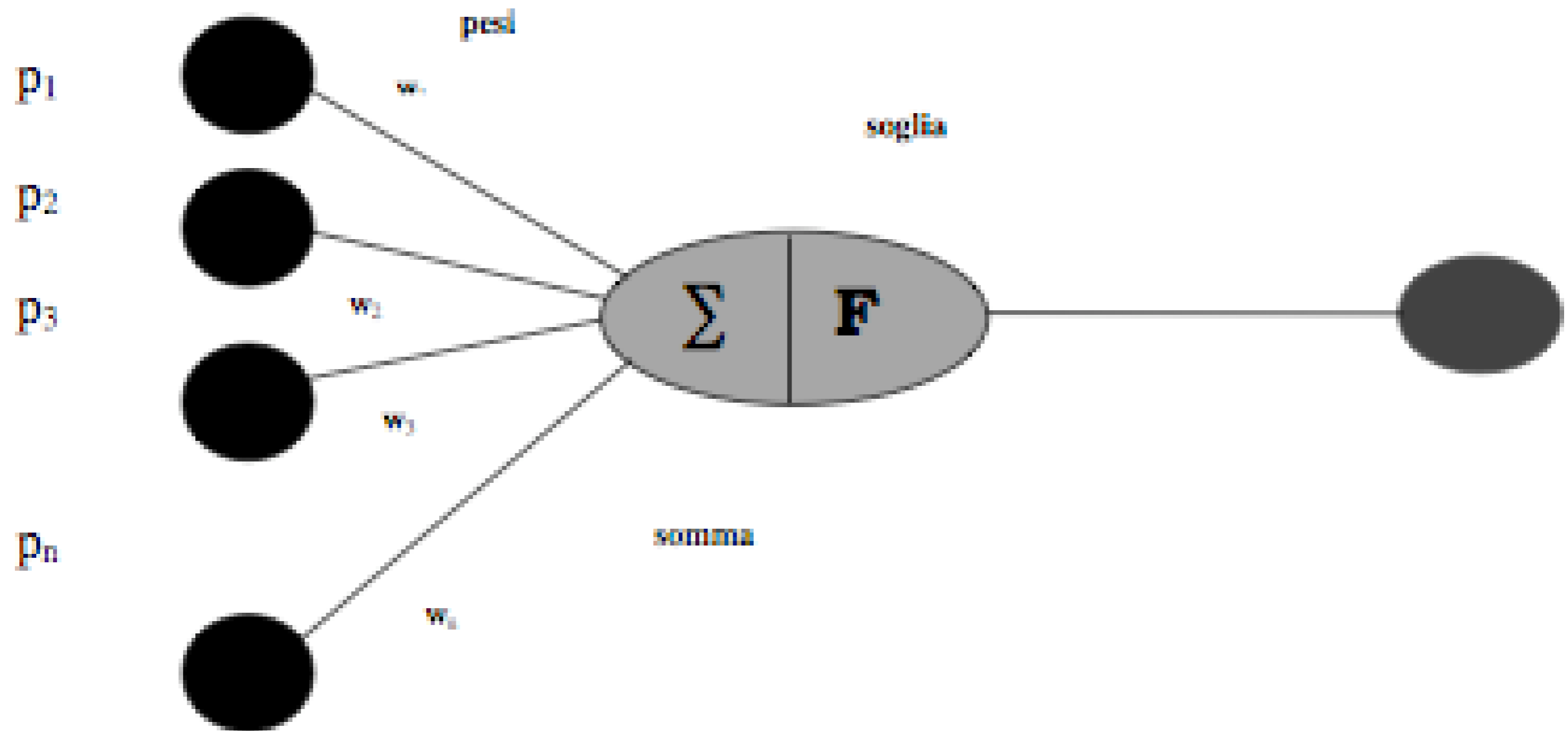
RETI NEURALI

Le Reti Neurali sono strumenti per l'elaborazione distribuita, originariamente progettati come metafore computazionali del cervello umano. Fanno parte di una classe più ampia di algoritmi capaci di apprendimento con applicazioni principali nei compiti di riconoscimento, di associazione e di manipolazione di strutture (pattern).

Sono costituite da una moltitudine di unità elementari di elaborazione (processing elements, PE, neuroni) interconnesse a formare una rete. Ciascuna unità compie una sequenza limitata di operazioni numeriche semplici e ne comunica i risultati alle unità adiacenti tramite una connessione cui è applicato un peso (detto peso sinattico).

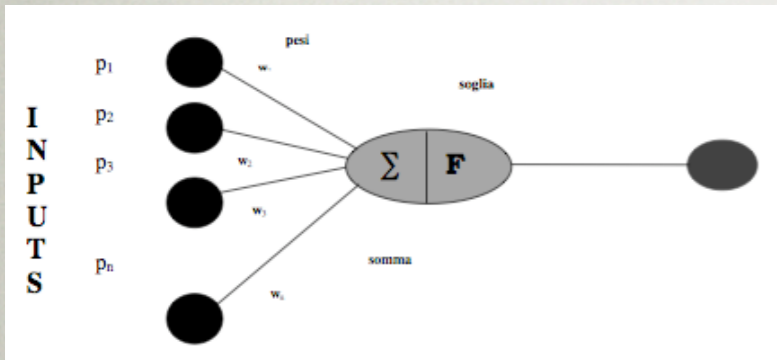
RETI NEURALI

Schema di un neurone artificiale



RETI NEURALI

Il neurone e le sue componenti



- p_1, p_2, \dots, p_n --> **input**, corrispondono ai segnali che arrivano dall'esterno;
- w_1, w_2, \dots, w_n --> **connessioni sinaptiche**, indicano il grado di importanza del segnale ricevuto;

$$a = F\left(\sum_{i=1}^n w_i p_i + b\right)$$

Funzione di trasferimento --> gli input (opportunamente pesati) sono aggregati a formare un unico valore, a cui di norma si aggiunge un parametro di distorsione (b).

Questo costituirà l'argomento della funzione di trasferimento che emetterà un segnale diverso a seconda del valore passato. La funzione di trasferimento ha il compito di controllare se il valore calcolato superi o meno una determinata soglia e di fornire l'output che costituisce il segnale per altri neuroni.

RETI NEURALI

Più nello specifico

PESI

- Intensità della forza di impulso presente a livello delle sinapsi. Ogni neurone può ricevere più input simultaneamente, e ognuno di essi può avere una diversa importanza (molto importante se l'intensità associata è alta, poco importante se l'intensità associata è bassa).
- Misurano la forza di connessione fra gli input e il neurone. Sono parametri scalari che possono essere variati in modo tale che la rete mostri un comportamento desiderato o interessante per la risoluzione del problema.
- Possono essere aggiustati esogenamente o dalla rete stessa mentre questa viene allenata.

RETI NEURALI

Più nello specifico

FUNZIONE DI SOMMA (1 / 2)

- Calcola la somma di tutti gli input opportunamente pesati. Geometricamente gli input e i pesi possono essere rappresentati da due vettori, (p_1, p_2, \dots, p_n) e (w_1, w_2, \dots, w_n) .
- Il segnale totale si ottiene moltiplicando ogni componente del vettore di input per il corrispondente elemento del vettore dei pesi $(w_1 * p_1, w_2 * p_2, \dots, w_n * p_n)$ e sommando i diversi risultati ottenuti dalla suddetta moltiplicazione.

$$(w_1 * p_1 + w_2 * p_2 + \dots + w_n * p_n) = \sum_{i=1}^n w_i p_i$$

- Di norma viene anche aggiunto un parametro di distorsione (b), che va a moltiplicare un parametro fornito dall'esterno in ingresso alla rete, solitamente posto uguale a 1, per cui il segnale complessivo diventa:

$$\sum_{i=1}^n w_i p_i + b$$

RETI NEURALI

Più nello specifico

FUNZIONE DI SOMMA (2 / 2)

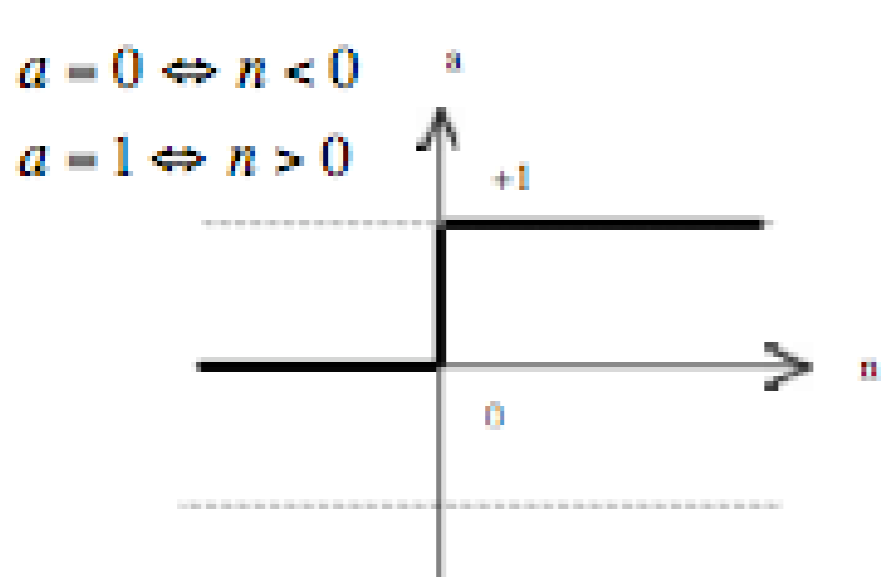
- Questa somma costituirà l'argomento che sarà passato alla funzione di trasferimento.
- Non è l'unica opportunità possibile di passaggio degli input: i pesi e gli input possono infatti essere combinati in molti modi differenti prima di diventare argomento della funzione di trasferimento. Questa scelta sarà lasciata al ricercatore, che dovrà tenere conto del contesto di applicazione della rete neurale.

RETI NEURALI

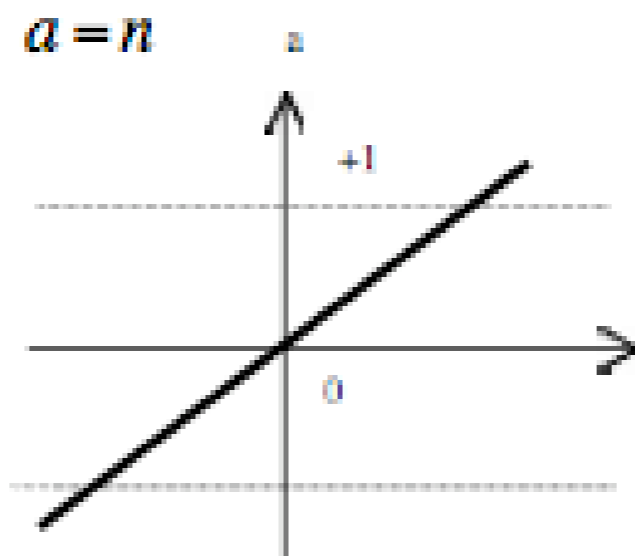
Più nello specifico

FUNZIONE DI TRASFERIMENTO

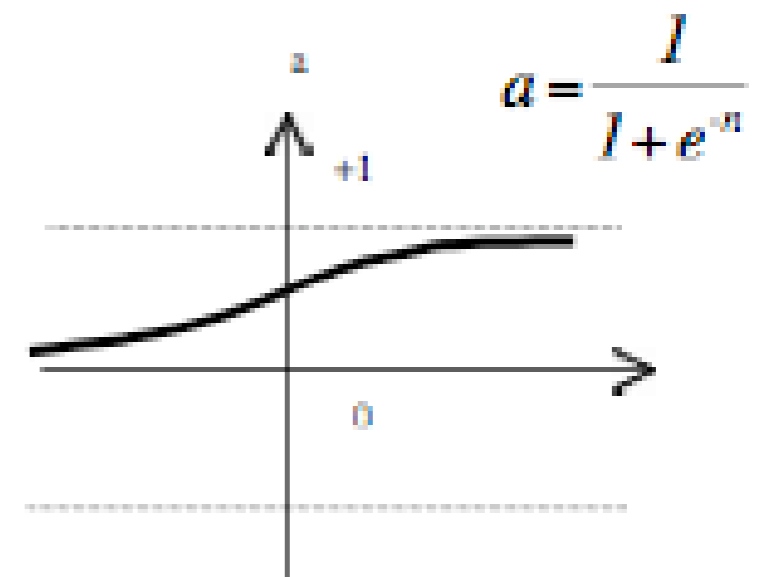
- Strettamente dipendente dalla funzione scelta. Ne esistono di diversi tipi (lineare, “hard limit”, sigmoide logaritmica, tangente sigmoide, etc...);
- Argomento della funzione sarà il valore somma degli inputs (slide precedente);
- Questo valore viene confrontato con un valore soglia: se è superiore si avrà la generazione di un segnale, se invece è inferiore sarà prodotto un segnale inibitorio o nessun segnale. Entrambi i tipi di risposta sono significativi.



Funzione Hard Limit



Funzione Lineare



Funzione Sigmoide

RETI NEURALI

Più nello specifico

SCALA E LIMITI

- Il risultato ottenuto dalla funzione di trasferimento può essere poi opportunamente modificato, per ottenere un valore più adatto al tipo di ricerca che si deve affrontare: si potrà quindi moltiplicare per un fattore di scala e/o aggiungere un determinato valore, mantenendosi però all'interno di limiti prefissati.

RETI NEURALI

Più nello specifico

FUNZIONE DI OUTPUT (COMPETIZIONE)

- L'output ottenuto da ogni neurone è proporzionale al valore determinato dalla funzione di trasferimento;
- ciò non preclude la facoltà di modifiche ulteriori: potrebbe, per esempio, essere modificato per evidenziare le sue capacità rispetto a quelle dei neuroni vicini.
- Dopo un opportuno confronto, è possibile determinare quali sono i neuroni maggiormente attivi che, perciò, "meritano" di partecipare in misura maggiore al processo di *learning* o di adattamento.

RETI NEURALI

Più nello specifico

FUNZIONE DI ERRORE

- Le reti sono adatte a problemi in cui è tollerata una certa componente di errore.
- Nell'apprendimento supervisionato (vedremo tra poco) il termine di errore è normalmente calcolato attraverso lo scarto quadratico medio (MSE).
- Esso è di fondamentale importanza per l'addestramento della rete, in quanto le variazioni dei pesi sono calcolate in modo tale da minimizzare questa componente di errore. (**ottimizzazione**)
- Fondamentale anche per prevenire fenomeni di overfitting nei quali la rete potrebbe essere stata sovrapparametrizzata o fatta allenare troppo

RETI NEURALI

Più nello specifico

FUNZIONE DI LEARNING

- il motore base per raggiungere la risoluzione del problema. Essa rappresenta matematicamente il metodo seguito per modificare i pesi di connessione attribuiti agli input, in modo tale che la rete "impari" a risolvere il problema, "adattandosi" alle diverse informazioni ricevute.

RETI NEURALI

Algoritmo di apprendimento

Le Reti Neurali non vengono programmate in modo diretto: si fa uso, invece, di un **algoritmo di apprendimento**. L'algoritmo di apprendimento è un meccanismo sistematico per la regolazione dell'intensità dei pesi della rete, con lo scopo di ottenere una determinata risposta funzionale, fissata come obiettivo per lo strato di unità di uscita, in corrispondenza di un dato insieme di input.

2 paradigmi

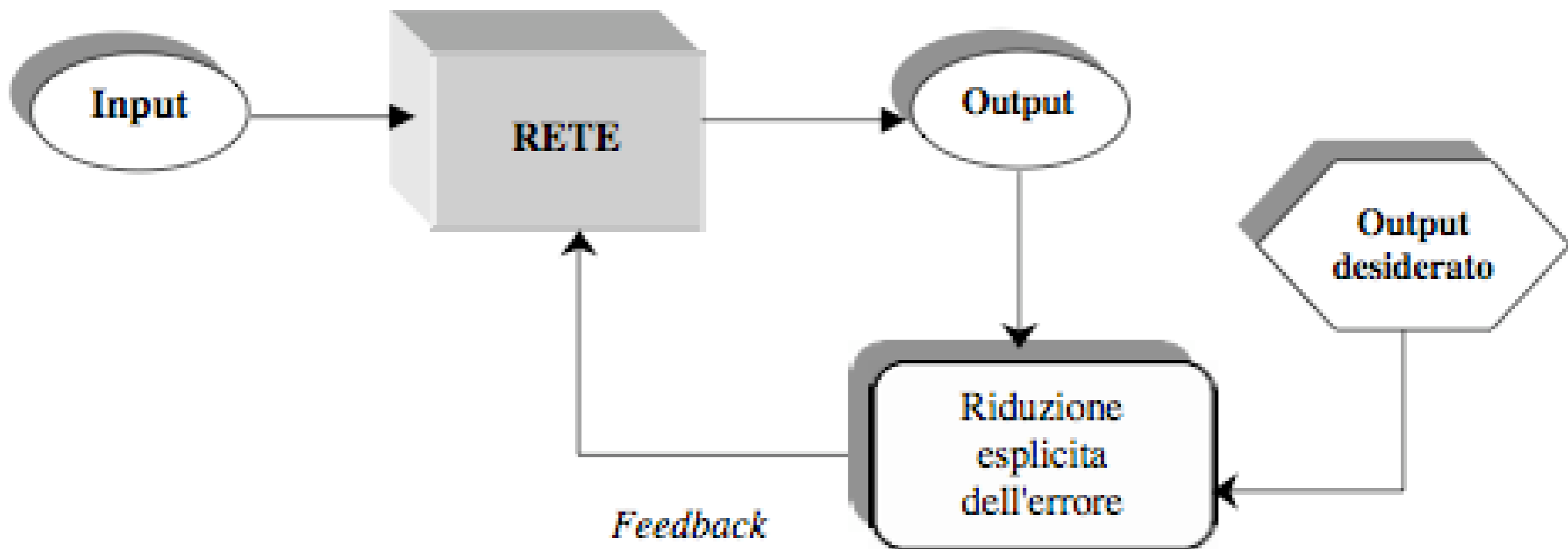
**Apprendimenti
supervisionato**

**Apprendimenti
NON
supervisionato**

RETI NEURALI

Apprendimenti supervisionato

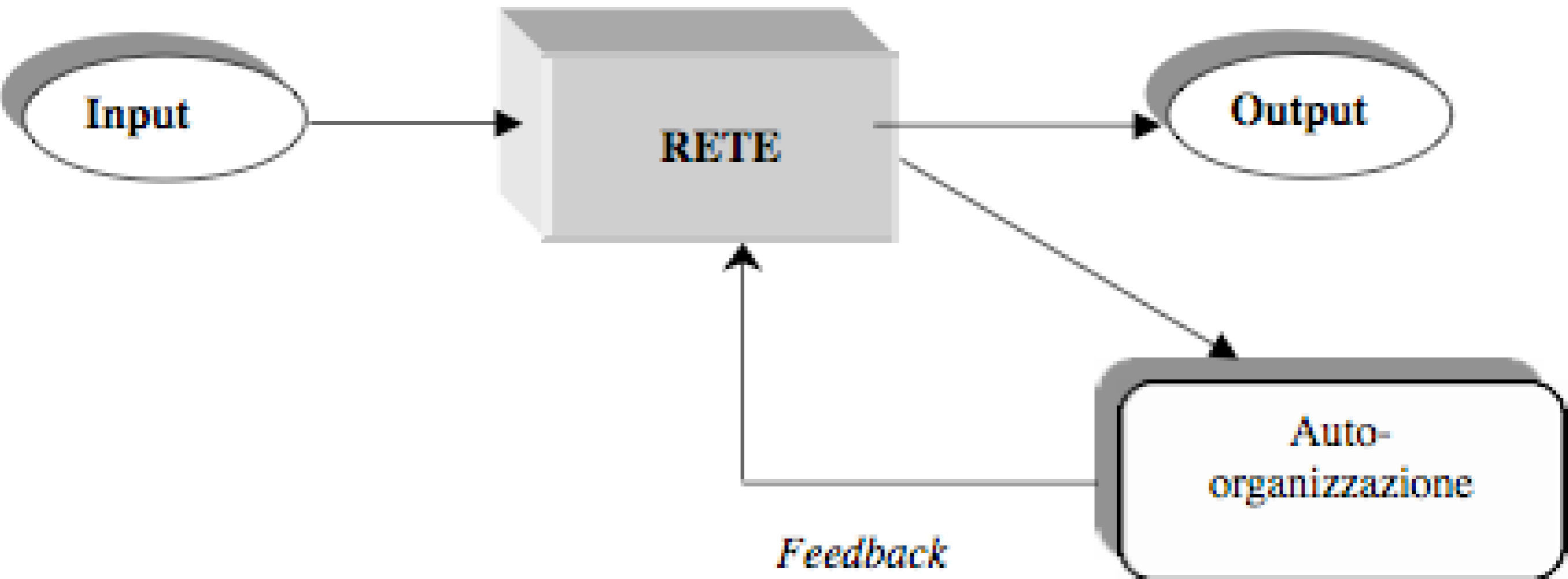
La rete come un approssimatore di funzioni



RETI NEURALI

Apprendimenti NON supervisionato

Configurazioni significative all'interno dei dati



RETI NEURALI

In termini più attuali le reti neurali

PRO

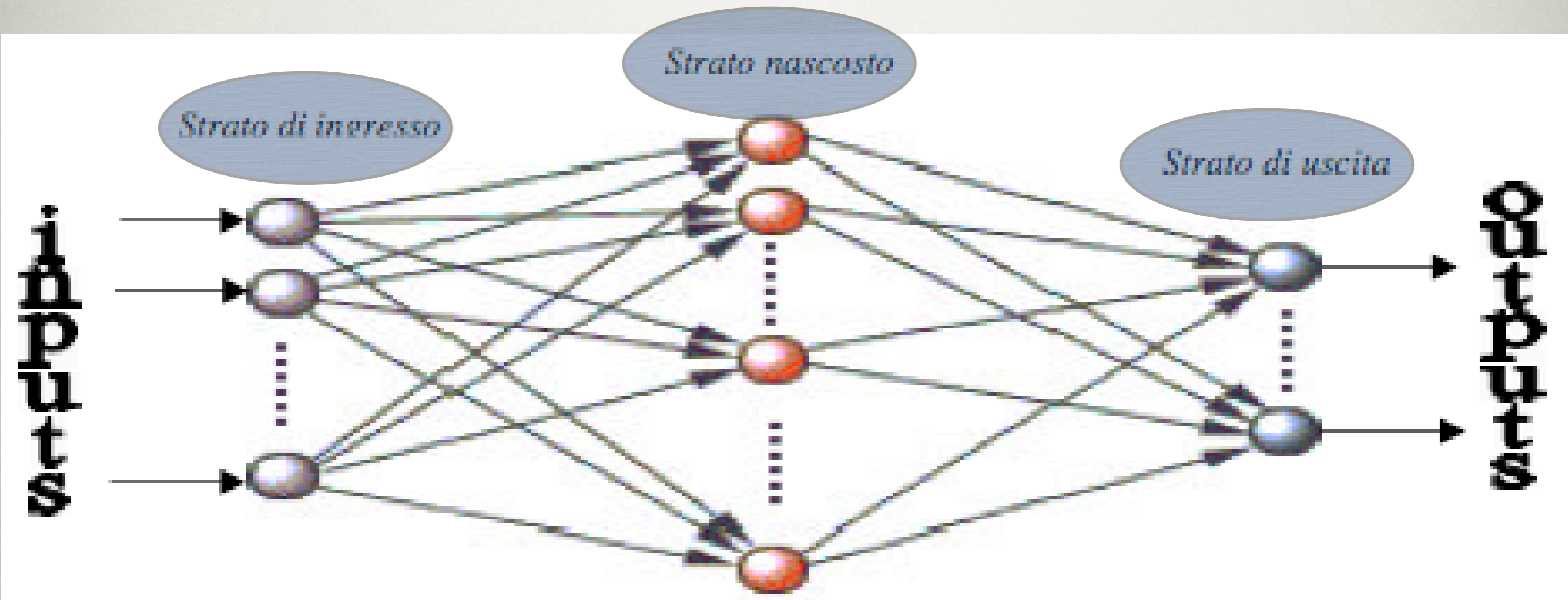
- Sono strumenti di modellizzazione non parametrica e non lineare
- Offrono la possibilità di generalizzare concetti
- Operano su set di dati mai visti in precedenza
- Imparano da una insieme di dati più piccolo dell'intero insieme

CONTRO

Non spiegano la dinamica del problema ma si limitano a descriverla

RETI NEURALI

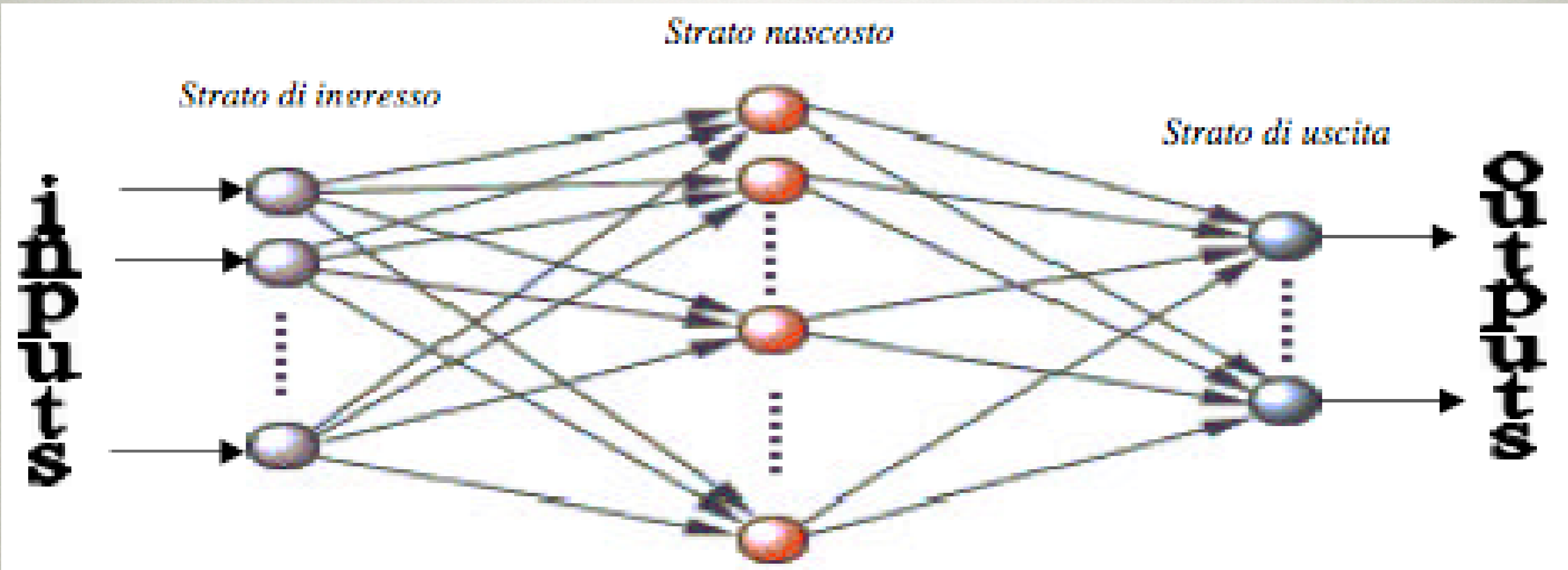
Topologia di una rete



RETI NEURALI

TOPOLOGIE

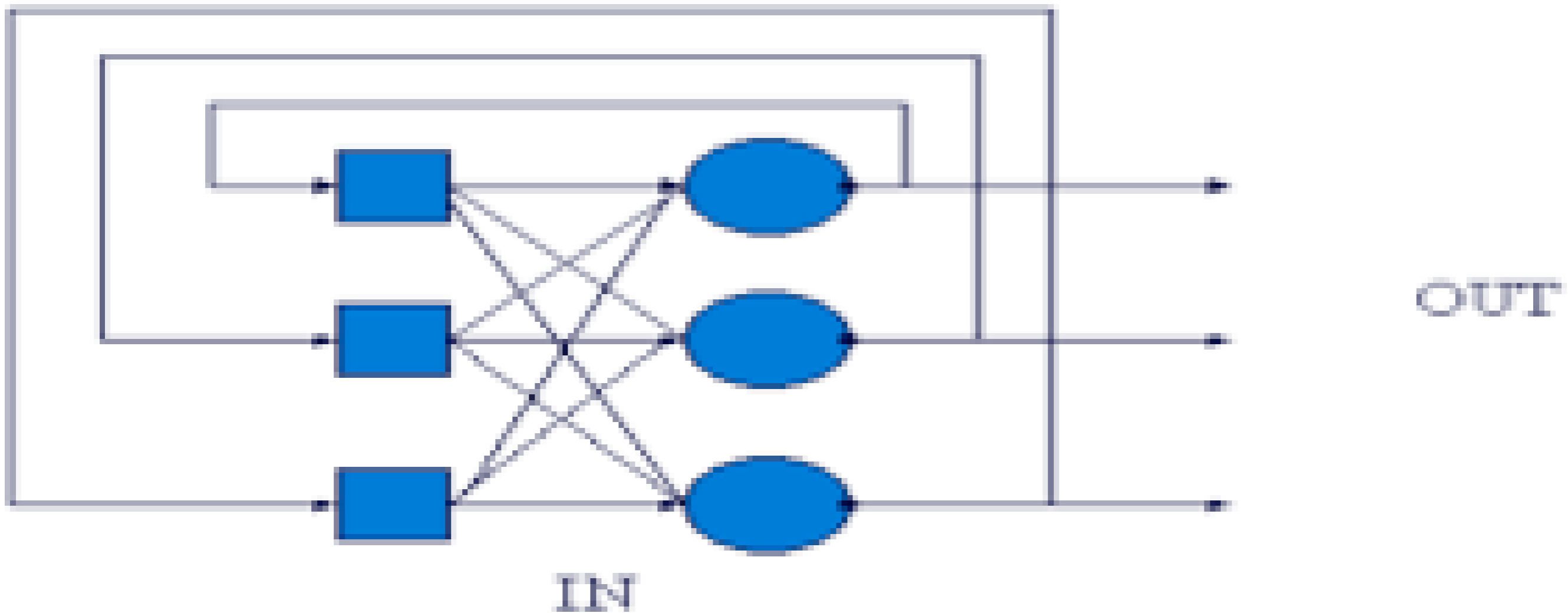
Layered feed-forward neural network



RETI NEURALI

TOPOLOGIE

Recurrent neural network



RETI NEURALI

TOPOLOGIE

Layered connected neural network

Essa è costituita da unità di input *feed-forward* e da uno strato di neuroni collegati lateralmente ognuno ai rispettivi vicini. Ogni neurone riceve input dall'esterno, ma anche da tutti i neuroni che lo circondano. Un esempio è dato dalla rete di Kohonen.

RETI NEURALI

TOPOLOGIE

Hybrid network

Si denominano in tal modo quelle reti che raggruppano peculiarità tipiche delle reti sopra esaminate, nella stessa architettura. Un esempio può essere dato dalla macchina di *Boltzmann*

RETI NEURALI

esempi di topologie

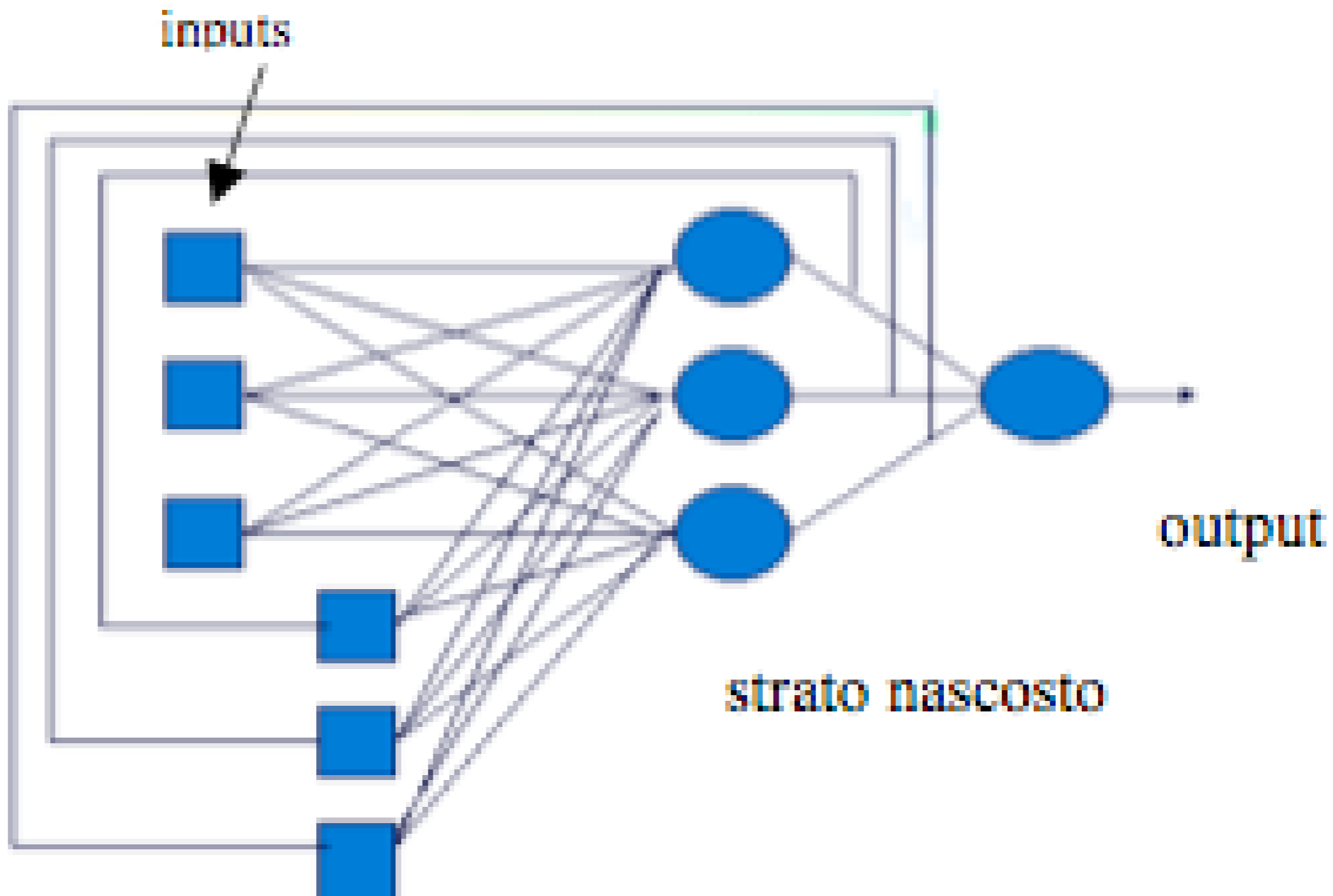


Fig. 6.7. Rete di Elman.

RETI NEURALI

esempi di topologie

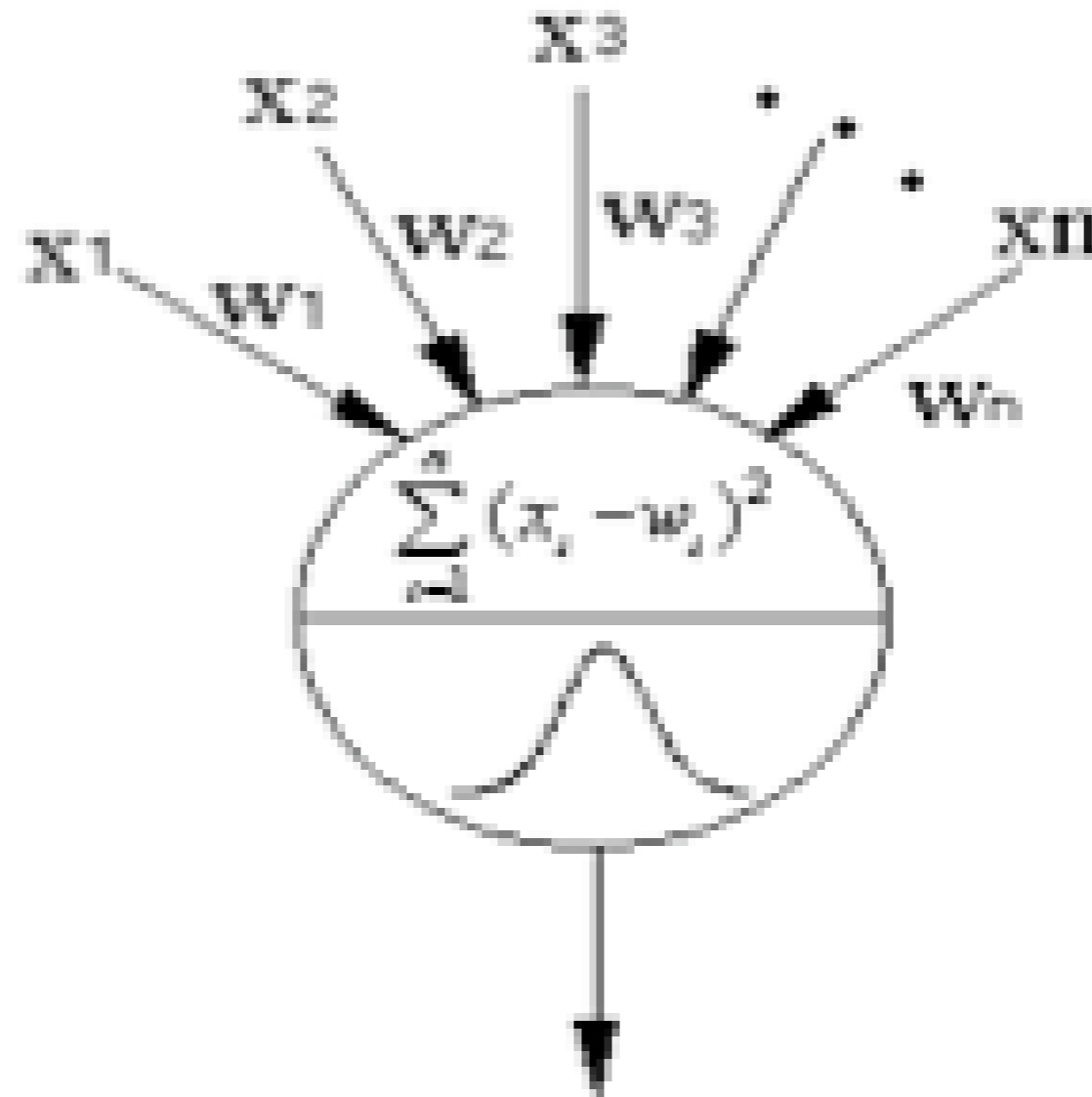
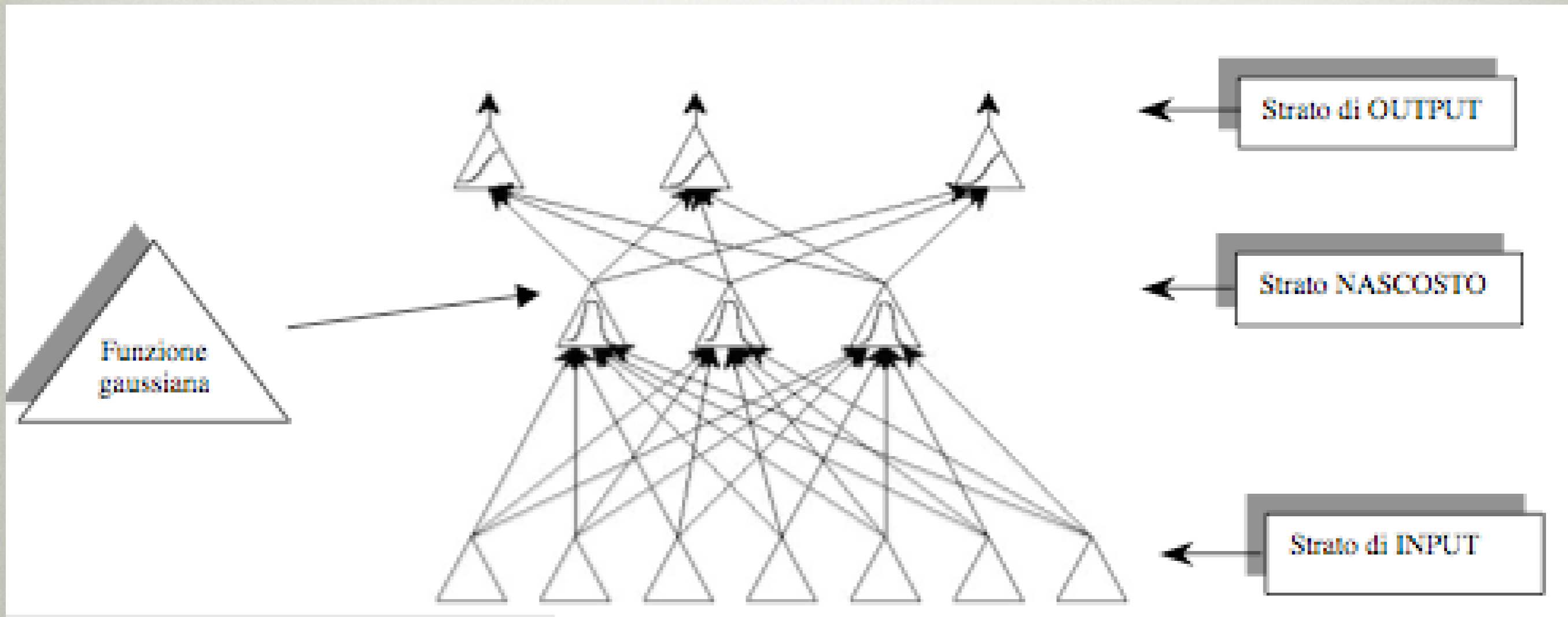


Fig. 6.8. Neurone artificiale usato nelle reti RBF.

RETI NEURALI

esempi di topologie



ESEMPI DI APPLICAZIONE

Algoritmi evolutivi

Selezione di un modello geneticamente evoluto

Il dataset contiene come insieme di variabili indipendenti gli spettri di assorbimento di radiazione elettromagnetica misurata a 21 frequenze diverse su 264 campioni di sangue e come variabile dipendente il livello di colesterolo presente nei 264 campioni.

Ci si chiede: **“Quali sono le frequenze tra le 21 disponibili che determinano il valore di Y ?”**.

SELEZIONE DI UN MODELLO STATISTICO

- X_p = numero di variabili

Noi vogliamo:

- $Y = f(X_k) + \epsilon$ dove $k \leq p$
- Quante sono le variabili statisticamente significative
- Quali sono le variabili statisticamente significative
- Quale è la forma della funzione f
- Come è strutturata la componente d'errore
 - Variabili non misurate
 - Elementi stocastici non individuabili

IPOTESI

- $\epsilon = N(0, \sigma^2)$ i.i.d (white noise)
componente stocastica con media 0 e
deviazione standard σ^2
- f sia più possibile lineare

CONDIZIONI LIMITE

Modello con tutte le variabili

$$y = \alpha_1 x_{k1} + \alpha_2 x_{k2} + \alpha_3 x_{k3} + \alpha_4 x_{k4} + \dots + \alpha_n x_{kn} + \epsilon$$

Soluzioni Possibili

$$2^n - 1 = nPoints$$

DEFINIRE LO SPAZIO DI RICERCA

$$2^n - 1 = nPoints$$

Soluzioni Possibili

21 variabili

X

364 osservazioni

Il data set viene diviso in due sotto-dataset:

- Training Set :: prova del modello
- Validation Set :: valutazione del modello

STABILIRE IL SISTEMA DI CODIFICA E DECODIFICA

Modello con tutte le variabili

$$y = \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \beta_4 x_4 + \dots + \beta_n x_n + \epsilon$$

Creazione della struttura del cromosoma

1 = considero la presenza della variabile

0 = non considero la presenza della variabile

1	0	1	1	1	0	1	0	0	0	0	0	1	0	0	1	0	0	0	1	1
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

$$Y = \beta_1 X_1 + \beta_3 X_3 + \beta_4 X_4 + \beta_5 X_5 + \beta_7 X_7 + \beta_{13} X_{13} + \beta_{16} X_{16} + \beta_{20} X_{20} + \beta_{21} X_{21}$$

PARAMETRI GENETICI E % DI MAPPATURA DELL'ALGORITMO

- CROSSOVER :: Single Point crossover
- SELEZIONE :: Il 80% della popolazione migliore partecipa al processo genetico, ogni individuo ha l'80% di probabilità di riprodursi.
- MUTAZIONE :: puntual mutation (0.2%)
- ELITISMO :: 2% degli individui migliori passano automaticamente alla generazione successiva
- Gli individui mancanti ad ogni popolazione verranno generati casualmente come per la generazione di partenza

CICLO EVOLUTIVO DELL'ALGORITMO GENETICO

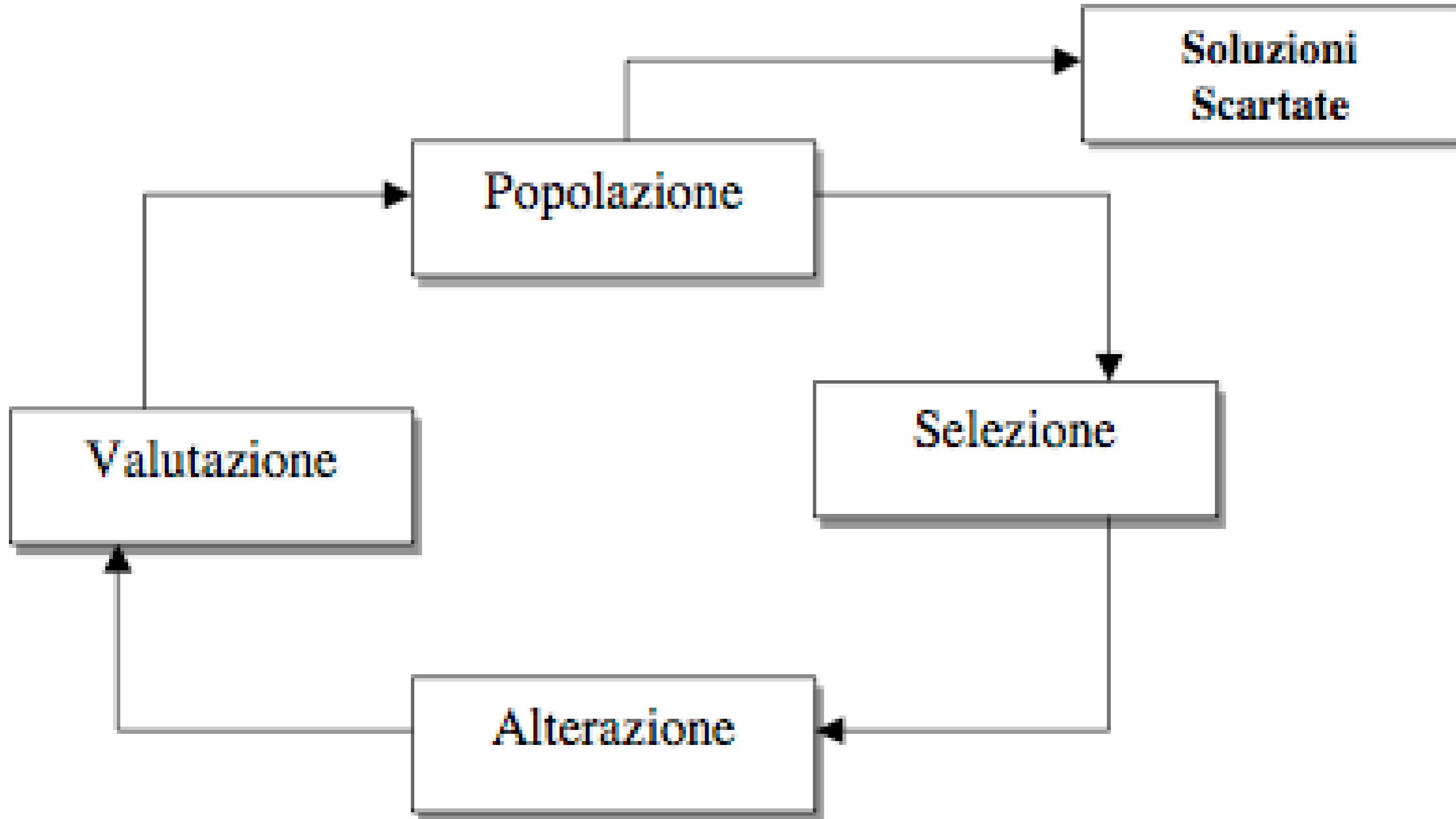


DIAGRAMMA DI FLUSSO DEL MODELLO

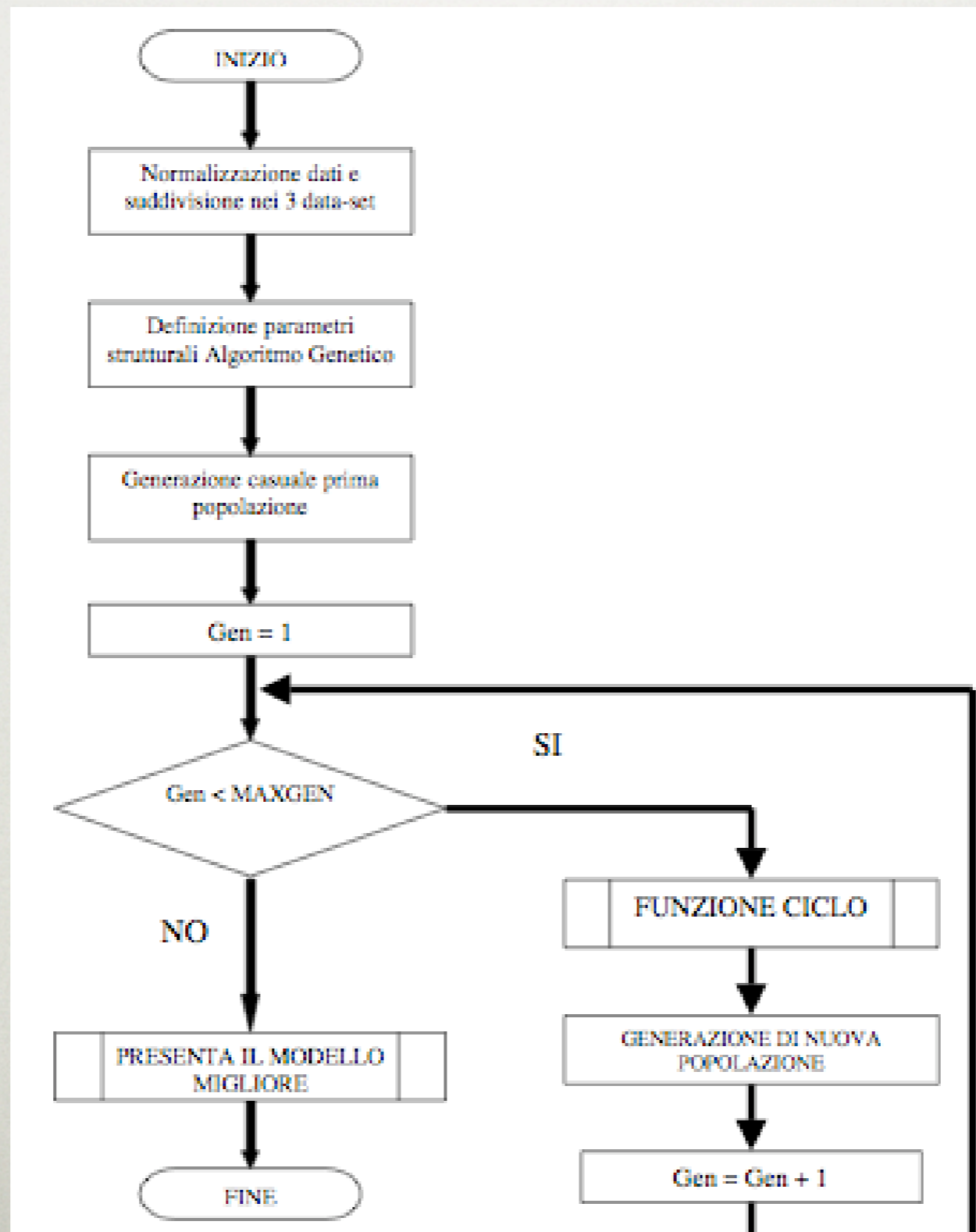
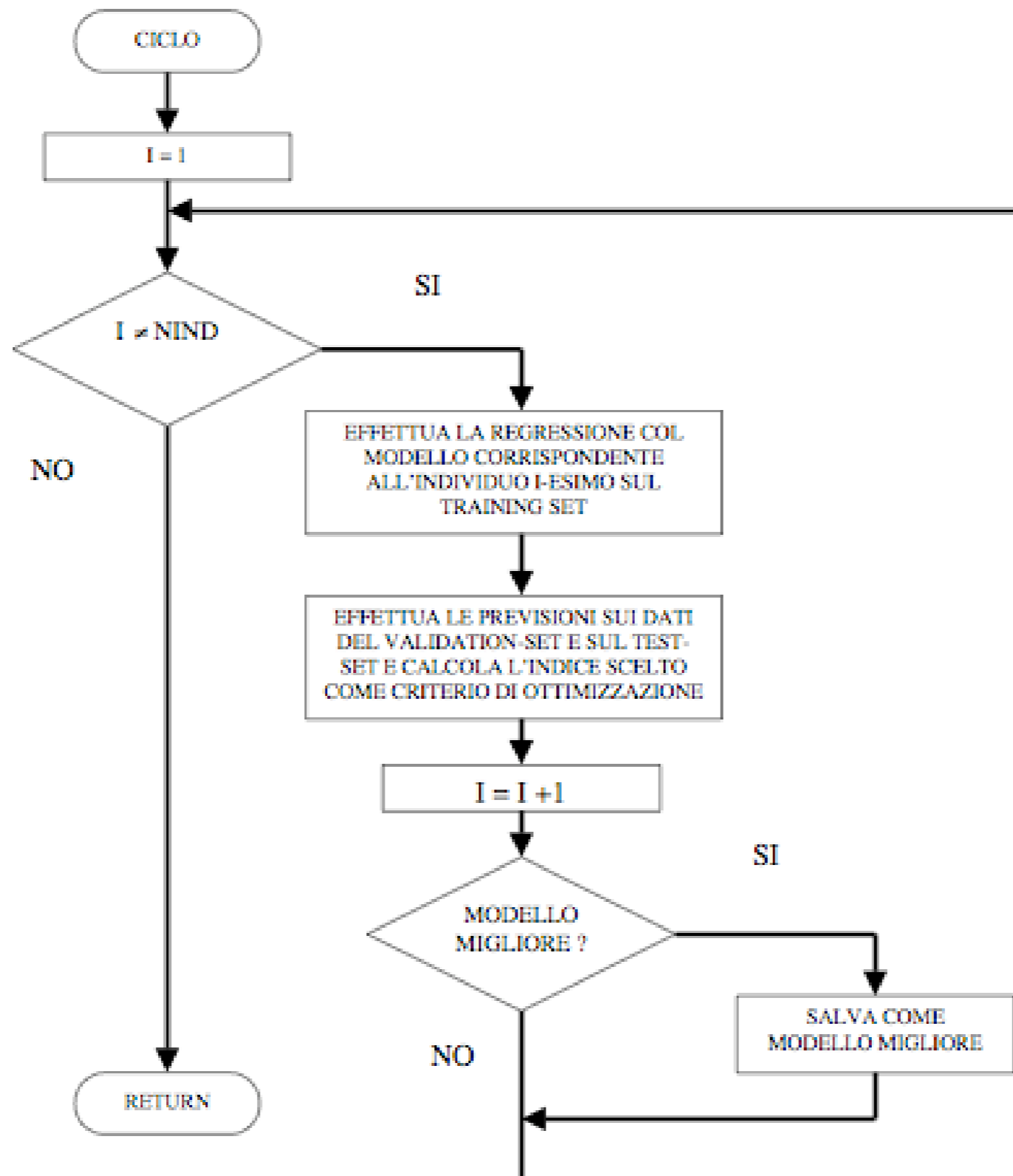


DIAGRAMMA DI FLUSSO DELLA FUNZIONE CICLO



RISULTATI GALMS

GENETIC ALGORITHM FOR LINEAR MODEL SELECTION

METODO	CRITERIO	RSQ_{test}	MSEP	N.V.	N_t
GALMS	<i>RSQ</i>	0.804	5.13	15	7
GALMS	<i>AIC</i>	0.808	3.87	5	-

CONCLUSIONI

- Ottimi risultati ottenuti con il metodo di valutazione fitness AIC
- Valore minore rispetto al metodo forward
- La valutazione in parallelo fornisce un modello più efficiente da un punto di vista previsivo

ESEMPI DI APPLICAZIONE

Reti neurali

**Il modello previsionale delle maree di
Venezia**

MODELLO PERVISIONALE DELLE MAREE DI VENEZIA

Il singolare andamento delle maree nel mare Adriatico ha, da molti anni, attratto fisici e meteorologi impegnandoli in più direzioni per la realizzazione di schemi numerici in grado di interpretare il più fedelmente possibile il livello di marea.

Elemento di particolare attrazione si è dimostrato il fenomeno dell'acqua alta che interessa principalmente le lagune venete, soprattutto quella di Venezia con l'allagamento del centro storico e delle isole. Questo evento provoca, da molti anni, danni rilevanti agli operatori economici veneziani, ai trasporti pubblici e privati, alla viabilità pedonale, all'attività portuale e da ultimo, ma non meno importante, causa condizioni di vita insalubri.

Gli studi per la previsione del livello di marea sono iniziati con la formulazione fisico-matematica del fenomeno mareale impostando le relazioni matematiche (equazioni dell'idrodinamica) dell'interazione aria-mare attraverso le condizioni iniziali e al contorno.

Si ottiene così la distribuzione nel tempo e nello spazio di alcune variabili quali per esempio la velocità di corrente e il livello marino.

Questo metodo, nonostante la buona affidabilità nella previsione, ha dimostrato ben presto grossi limiti, come la necessità di conoscere in tempo reale alcuni parametri meteomarinari o la necessità di un potente mezzo di calcolo per ottenere velocemente i risultati.

VARIABILI D'INTERESSE

- L'esistenza delle sesse (oscillazioni libere del mare) con un periodo di 22 ore per la frequenza fondamentale, suggerisce l'autoregressione, o l'uso dei livelli osservati nelle ore precedenti come predittori. In altre parole, questo significa che se l'Adriatico fosse già in oscillazione, se non ci fossero nuovi disturbi e se non ci fosse smorzamento, il suo livello risulterebbe lo stesso di 22 ore prima.
- Fattore meteorologico: sia la pressione atmosferica (effetto barometro) che il vento col suo trascinamento hanno effetti notevoli; poiché il vento dipende soprattutto dalla pressione, tanto vale provare solo l'uso di quest'ultima per il nostro scopo.
- La dipendenza non lineare tra la velocità del vento e il suo impatto sul mare può essere approssimata con la linearità, almeno come tentativo.

STRUTTURA DEL MODELLO

- 182 predittori: : si cerca tra tutti i possibili $218^{(2)}-1$ modelli quello “migliore” per ogni anticipo previsivo effettuato (1, 3, 6, 9, 12, 15 e 24 ore).
 - 132 livelli di marea osservata a Venezia, all’indietro nel tempo, da una certa ora T all’ora T-131 (cosiddetti parametri marini),
 - 50 valori di pressione atmosferica osservati a Bari, Falconara, Genova, Pesaro, Pescara, Ravenna, Rimini, Teramo, Trieste, Venezia, ogni 3 ore dal tempo T al tempo T-12 (cosiddetti parametri metereologici).

SELEZIONE DEL MODELLO MIGLIORE

Grazie all'utilizzo del programma GALMS studiato precedentemente viene selezionato il modello migliore

UTILIZZO DELLA RETE

Mediante l'utilizzo di Reti Neurali si cercherà una possibile componente non lineare tra i dati autoregressivi di marea e di pressione con il valore previsto di marea.

Le Reti Neurali con addestramento supervisionato sono infatti considerate buoni approssimatori di funzioni non lineari.

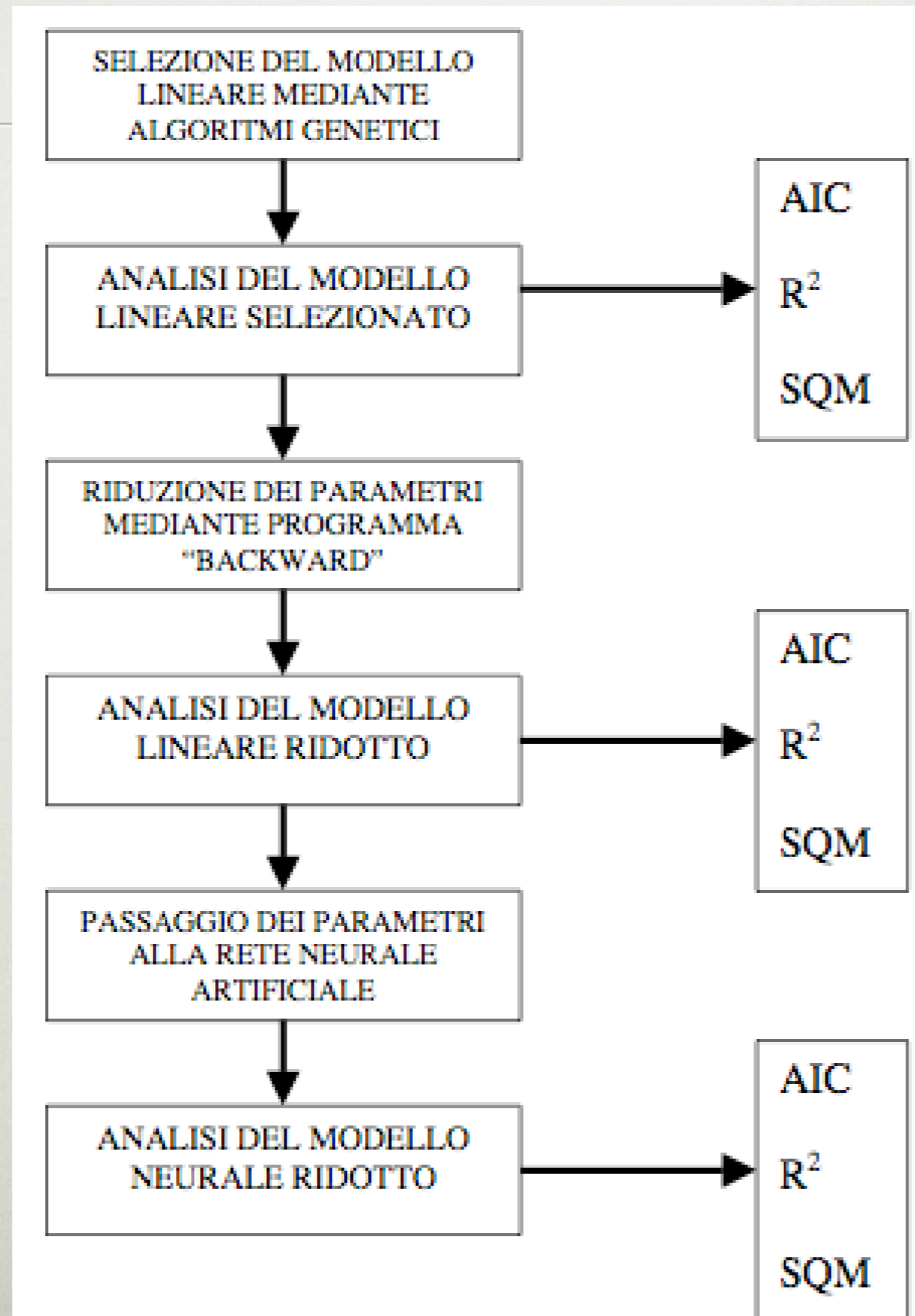
Viene passato alla rete il modello selezionato dall'algoritmo genetico

TRAINING, VALIDATION E TEST SET

L'intero set di dati è stato qui campionato al fine di poter applicare la metodologia di "Early Stopping".

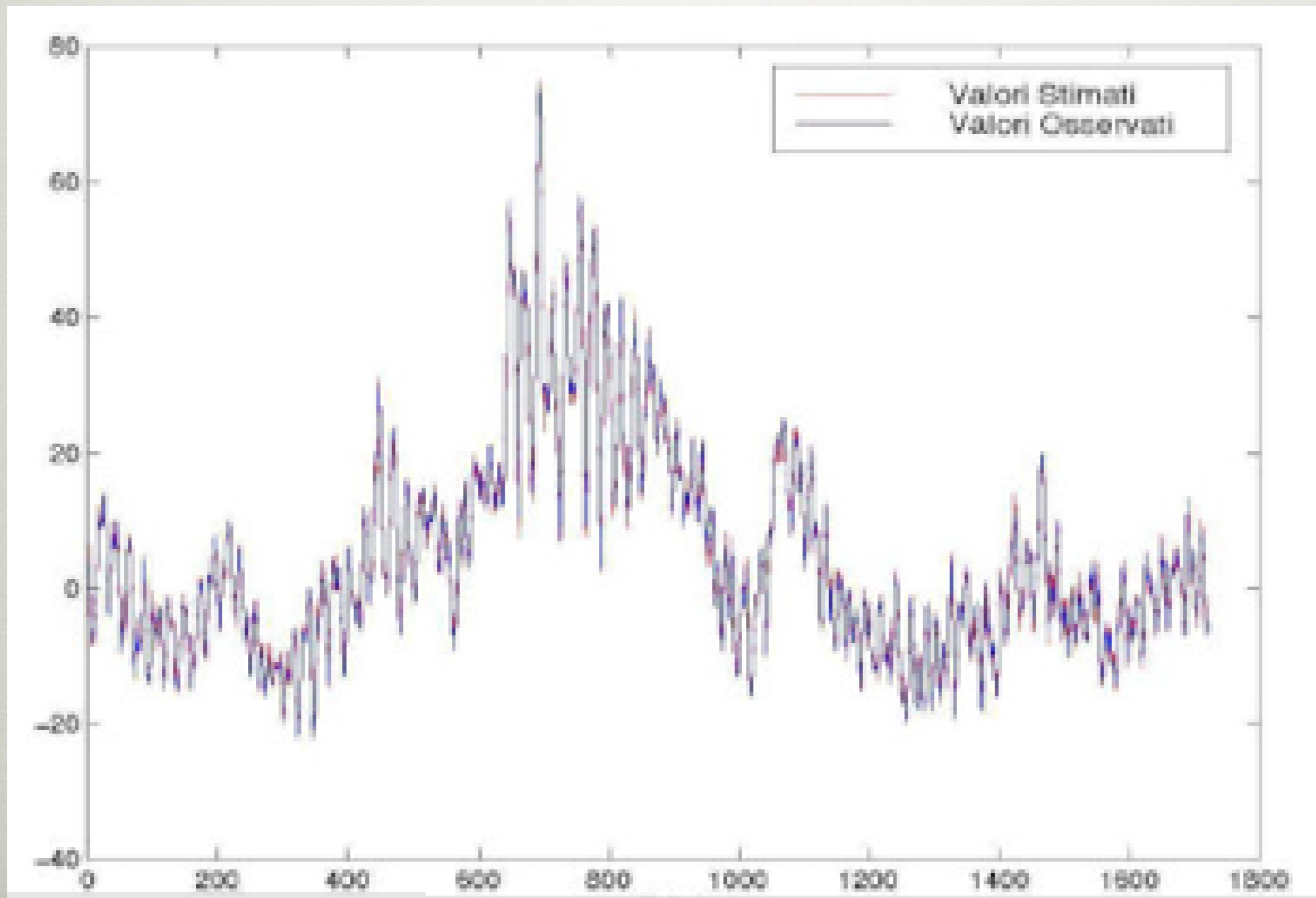
- **Training set** (circa 5000 dati corrispondenti al periodo 6/1/84 - 1/8/84); utilizzato esclusivamente durante l'allenamento per il calcolo del gradiente e l'aggiornamento della matrice dei pesi sinattici;
- **Validation set** (circa 1700 dati corrispondenti al periodo 2/8/84 – 19/10/84); utilizzato dall'algoritmo per il calcolo della funzione di errore associata alla struttura neurale implementata;
- **Test set** (la restante parte corrispondente al periodo 19/10/84 – 31/12/84). non viene utilizzato in fase di allenamento ma serve soltanto per il confronto di modelli diversi.

PROCEDIMENTO DI ANALISI



RISULTATI

VALORI PREVISTI VS. VALORI OSSERVATI



FINE