

Table of Contents

Reti neurali evolutive con topologia a rete complessa	1
<i>Mauro Annunziato, Ilaria Bertini, Matteo De Felice, Stefano Pizzuti</i>	

Reti neurali evolutive con topologia a rete complessa

Mauro Annunziato¹, Ilaria Bertini¹, Matteo De Felice², and Stefano Pizzuti¹

¹ Energy, New technology and Environment Agency (ENEA), Via Anguillarese 301, 00123 Rome, Italy, {mauro.annunziato, ilaria.bertini, stefano.pizzuti}@casaccia.enea.it

² Dipartimento di Informatica ed Automazione, Università degli Studi di Roma “Roma Tre”, Via della Vasca Navale 79, 00146 Rome, Italy
matteo.defelice@casaccia.enea.it

Abstract. Le reti complesse sono state osservate in molti sistemi biologici reali e l'applicazione di tale topologia alle reti neurali artificiali porta a delle considerazioni interessanti. In questo articolo presentiamo uno studio preliminare sull'evoluzione di reti neurali a topologia complessa. Questa struttura viene utilizzata per affrontare un problema di modellazione di un processo chimico in cui sono presenti degli ingressi non conosciuti (disturbi). L'algoritmo evolutivo da noi utilizzato considera una popolazione iniziale di individui ognuno dei quali ha nel proprio genotipo una rete scale-free differente ed al termine abbiamo osservato ed analizzato la topologia delle reti che hanno mostrato le migliori prestazioni. Le simulazioni mostrano che le performance delle reti con topologia complessa sono simili a quelle delle reti feed-forward ma l'analisi della topologia delle reti più efficienti porta alla conclusione che le prestazioni di una rete (in questo caso specifico la capacità di modellare) sono influenzate dalla configurazione dei nodi in input.

1 Introduzione

Le reti neurali artificiali e gli algoritmi evolutivi sono ambedue astrazioni di processi naturali. Sono formulati in un modello computazionale cosicchè sia la potenza di apprendimento delle reti neurali che le capacità adattive dei processi evolutivi sono racchiusi in un ambiente di vita artificiale. L'apprendimento adattivo produce risultati che dimostrano come può essere indotto un comportamento complesso e significativo in un sistema tramite la variazione casuale della topologia e delle regole che lo governano. Gli algoritmi evolutivi possono aiutare per determinare delle architetture delle reti neurali ottimizzate dando così inizio ad una nuova branca delle reti neurali, le Reti Neurali Evolutive [1]. È assodato [2] che nella maggior parte dei casi la combinazione di algoritmi evolutivi e reti neurali porta a prestazioni pari (in termini di accuratezza) a reti neurali progettate ad hoc ed addestrate con algoritmi di back-propagation [3] ed oltretutto, di fronte ad alcuni problemi, la prima combinazione porta anche a prestazioni superiori. Questo significa che in alcune applicazioni dove l'accuratezza è fondamentale,

potrebbe essere conveniente sperimentare combinazioni tra algoritmi evolutivi e reti neurali. Un'area di ricerca nuova e interessante che è nata recentemente è quella delle *Reti Complesse*. Questo tipo di reti sta ricevendo grande attenzione nella comunità dei fisici, perchè sembra essere la struttura alla base di molte reti naturali ed artificiali come quelle presenti nelle proteine, nel metabolismo, nel sistema di specie animali, in molte reti biologiche [4–6], nella rete Internet, il WWW, le reti metaboliche e molte altre [7].

In questo contesto, le reti neurali con topologie complesse gestite da meccanismi evolutivi sono un'idea nuova e noi le abbiamo usate per modellare dei processi complessi.

2 La metodologia

L'obiettivo di questo lavoro è lo studio di reti neurali evolutive con una topologia basata su grafi diretti, ottenuta usando un algoritmo iterativo simile a quello proposto da Barabasi-Albert nel 1999 [8].

2.1 Le reti complesse

Non esiste una definizione univoca di “rete complessa”, questo termine si riferisce ad una rete con una topologia non banale e con un alto numero di nodi e connessioni. Comunque le reti complesse possono essere classificate, in accordo ad alcuni descrittori della topologia, in due grandi classi: le reti *Small World* e le reti *scale-Free*. I più importanti descrittori della topologia sono: la distribuzione del grado dei nodi, la lunghezza del cammino minimo più breve e il coefficiente di clustering. Le proprietà di queste reti sono spesso confrontate con quelle dei grafi random [9] che possono senza dubbio essere considerati delle strutture “semplici”. I grafi di tipo random presentano una distribuzione del grado dei nodi di tipo poissoniana, il valore del più breve cammino minimo è piccolo così come presentano un piccolo coefficiente di clustering. Le reti di tipo “small-world” [4, 10] presentano anch'esse una distribuzione del grado dei nodi poissoniana, hanno il più breve cammino minimo di valore piccolo ma un alto coefficiente di clustering. Queste reti si trovano a metà tra le reti regolari e le reti random (vedi figura 2.1) ed è stato dimostrato [4] che questa topologia risulti ottimale per i compiti di comunicazione.

Il modello scale-free [11] presenta una distribuzione del grado dei nodi che segue una legge di potenza. Ciò significa che sono presenti pochi nodi ad alta connettività (*hubs*) e molti nodi con pochi collegamenti (vedi figura 2.1). Queste reti presentano la proprietà “small-world” [4], infatti qualsiasi coppia di nodi presenti nella rete sono distanziati da un numero relativamente piccolo di archi. Questi tipi di reti stanno ricevendo una grande attenzione nella comunità dei fisici, perchè recentemente sono state osservate molte reti con una distribuzione dei nodi scale-free. Per esempio citiamo la rete Internet, il WWW, la rete delle e-mail, le reti metaboliche, le reti di fiducia (*trust*) e molte altre [7]. I nodi di queste reti hanno la probabilità di ottenere un nuovo collegamento proporzionale

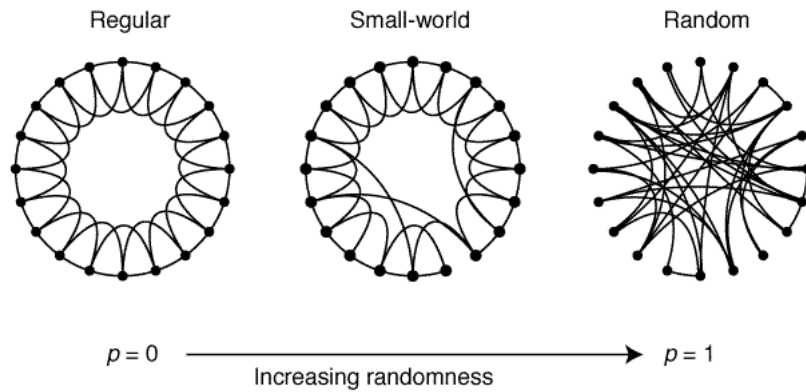


Fig. 1. Topologie di reti complesse

al numero di collegamenti che già hanno, per questo tali reti si dice che presentano il fenomeno riassunto con la frase “the rich gets richer”.

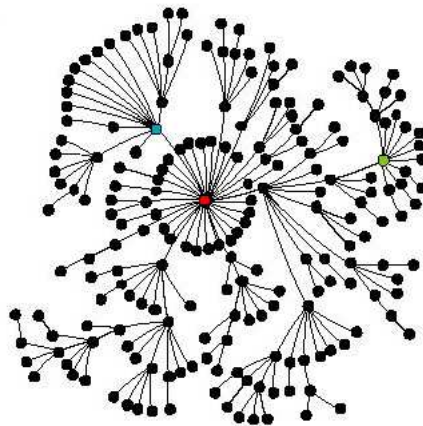


Fig. 2. Rete a topologia scale-free

In questo lavoro ci siamo concentrati sulla reti scale-free.

2.2 L'algoritmo

Abbiamo utilizzato delle reti neurali basate su reti complesse creati con l'algoritmo il cui pseudo-codice è visibile in tabella 1. In questo algoritmo abbiamo considerato due tipi di connessioni tra la rete e i nodi di input (nello pseudo-codice questi metodi sono indicati dalla variabile `INPUT_CONNECTION_TYPE`. Nel caso A

abbiamo tutti i nodi di input connessi a m nodi della rete. Diversamente nel caso B abbiamo ogni nodo di input con m archi ad altrettanti nodi non di input della rete. Come si può vedere nello pseudo-codice in tabella 1, c'è una funzione di *preferential-attachment*: questa funzione definisce la probabilità che un nodo della rete riceva un arco da un nodo appena inserito [8,12]. La forma analitica di questa funzione è la seguente:

$$P(k_i) = \frac{k_i^\alpha}{\sum_j k_j^\alpha} \quad (1)$$

Questa funzione è crescente monotona, il parametro α influenza il numero di hub con alta connettività e k_i rappresenta il grado del nodo i -esimo. Nelle figure 2.1 e 2.1 mostriamo la distribuzione del grado dei nodi di reti con 4000 nodi costruite con l'algoritmo presentato con due valori di α differenti, nei grafici è mostrata anche una funzione di fitting del tipo $k^{-\gamma}$.

Algorithm 1 Algoritmo di creazione delle reti neurali complesse

```

for all  $i$  tale che  $0 \leq i \leq m_0$  do
  ADD node $_i$ 
  CONNECT node $_i$  to ALL
end for
for all  $i$  tale che  $1 \leq i \leq \text{TOTAL\_NODES}$  do
  ADD node $_i$ 
  for all  $j$  tale che  $1 \leq j \leq m$  do
     $x \leftarrow$  GET\_NODE\_WITH\_PREFERENTIAL\_ATTACHMENT
    CONNECT node $_i$  to node $_x$ 
    CONNECT node $_x$  to node $_i$ 
  end for
end for
 $x \leftarrow$  RANDOM(TOTAL\_NODES)
OUTPUT\_NODE  $\leftarrow$  node $_x$ 
if INPUT\_CONNECTION\_TYPE = A then
  ADD ALL\_INPUT\_NODES
  for all  $i$  tale che  $1 \leq i \leq m$  do
     $x \leftarrow$  RANDOM(TOTAL\_NODES)
    CONNECT ALL\_INPUT\_NODES to node $_x$ 
  end for
end if
if INPUT\_CONNECTION\_TYPE = B then
  ADD input\_node $_i$ 
  for all  $j$  tale che  $1 \leq j \leq m$  do
     $x \leftarrow$  GET\_NODE\_WITH\_PREFERENTIAL\_ATTACHMENT
    CONNECT input\_node $_i$  to node $_x$ 
  end for
end if

```

Una rete creata con questo algoritmo è mostrata in figura 2.2.

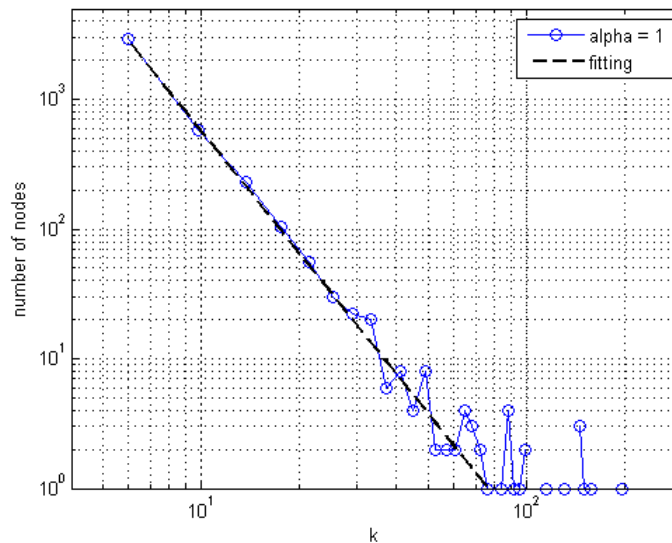


Fig. 3. Distribuzione del grado dei nodi di una rete di 4000 nodi creata con $\alpha = 1$ e fitting eseguito con una legge di potenza con $\gamma = 3.1$ (linea tratteggiata)

Come parametri dell'algoritmo abbiamo usato i valori mostrati nella tabella 2.3.

2.3 L'ambiente evolutivo

L'ambiente evolutivo da noi implementato è un ambiente di Vita Artificiale (ALIFE) [13]. Questo tipo di approccio è stato testato su problemi statici di ottimizzazione molto conosciuti come il Problema del Commesso Viaggiatore (TSP, *Travelling Salesman Problem*), il circuito di Chua, il sistema dinamico di Kuramoto [14] così come su numerosi casi reali [15–18]. Il contesto di ALIFE consiste in un reticolo bidimensionale (*life space*) che rappresenta uno spazio fisico dove gli individui artificiali (o agenti autonomi) possono muoversi liberamente. Ad ogni iterazione (*life cycle*), gli individui si muovono nello spazio e nell'eventualità di un incontro tra due individui avviene un'interazione. Ogni individuo possiede un particolare insieme di regole che determinano le sue interazioni con gli altri agenti, principalmente basate sul confronto tra la loro energia e il loro valore di performance. Gli individui si possono auto-riprodurre tramite una mutazione di tipo aploide che avviene solamente se l'individuo possiede un'energia maggiore del valore di energia fissato per la nascita. Infatti durante la nascita avviene che dal genitore al figlio venga trasferita una quantità di energia predefinita, chiamata energia di nascita. Nella riproduzione aploide viene effettuato ad ogni life cycle un test di probabilità per un eventuale auto-riproduzione e la mutazione

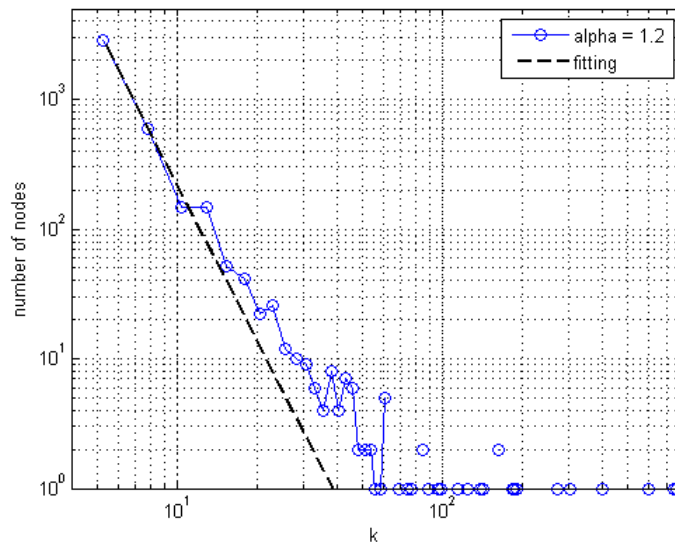


Fig. 4. Distribuzione del grado dei nodi di una rete di 4000 nodi creata con $\alpha = 1.2$ e fitting eseguito con una legge di potenza con $\gamma = 4$ (linea tratteggiata)

avviene nei geni rispettando il tasso e l'ampiezza di mutazione, anch'essi stabiliti in maniera evolutiva [19]. Quando due individui si incontrano avviene uno scontro. Il vincitore è l'individuo che possiede un valore di performance maggiore, lo sconfitto trasferisce un quantitativo di energia (*fighting energy*) al vincitore. Ad ogni life cycle l'età di ogni individuo viene incrementata e quando raggiunge un valore vicino al tempo di vita medio, la probabilità di morte naturale aumenta. Il meccanismo di invecchiamento è molto importante per garantire la possibilità di perdere memoria delle soluzioni più vecchie e seguire così l'evoluzione del processo. La morte di un individuo avviene anche nel momento in cui la sua energia al termine di una riproduzione o di uno scontro diventa nulla. Una descrizione dettagliata di questa metodologia è riportata in [15, 16]. L'ambiente di Vita Artificiale da noi utilizzato si chiama "Artificial Societies" [20] con alcune modifiche rispetto l'originale. Nella nostra implementazione ogni individuo della popolazione di partenza è inizializzato con una topologia di rete differente nel proprio genotipo. Le reti di partenza sono create con un valore del parametro m che varia da 2 a 6. I pesi degli archi e le funzioni di attivazione, ma non la topologia, sono soggetti a mutazioni casuali mentre non è stato implementato un meccanismo di crossover tra reti con differenti topologia dal momento che nell'algorithm da noi utilizzato non è utilizzato un meccanismo di riproduzione bisessuale.

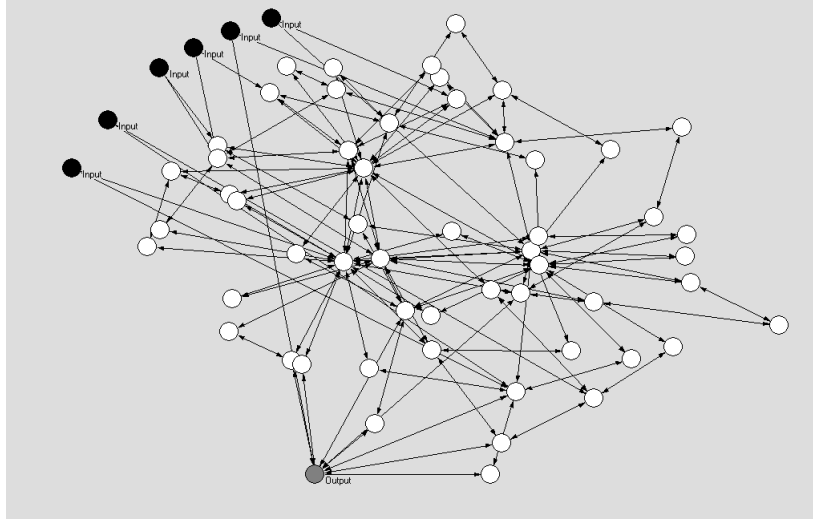


Fig. 5. Esempio di una rete neurale a topologia complessa. I nodi di input sono di colore nero mentre quelli di output sono di colore grigio

Table 1. Parametri dell'algoritmo

m_0	4
m	2 – 6
α	1.2
Dimensione dello spazio	25×25
Individui della popolazione iniziale	215

3 Il modello di benchmark

Il modello che abbiamo usato consiste in un processo composto da due CSTR (*Continuous flow Stirred Tank Reactor*) linearizzati posti in parallelo (vedi figura 2.2). Il modello del CSTR descrive una reazione del primo ordine esotermica diabatica ed è comunemente studiato per le sue caratteristiche [21, 22]. Le equazioni, scritte in forma adimensionale [23], sono le seguenti:

$$\begin{aligned}
 \dot{x}_1 &= q(x_{1f} - x_1) - \Phi x_1 \kappa(x_2) \\
 \dot{x}_2 &= q(x_{2f} - x_2) - \delta(x_2 - x_3) + \beta \Phi \kappa(x_2) \\
 \dot{x}_3 &= \delta_1 [q_c(x_{3f} - x_3) + \delta \delta_2 (x_2 - x_3)]
 \end{aligned} \tag{2}$$

dove x_1 , x_2 e x_3 sono rispettivamente la concentrazione, la temperatura di reazione e la temperatura del cooling-jacket tutte in forma adimensionale. Il primo ingresso è q_c e rappresenta il flusso del cooling-jacket mentre q , il flusso del reagente, rappresenta il secondo ingresso. L'uscita del sistema è la concentrazione adimensionale x_1 . Gli altri parametri con i valori che abbiamo usato

con relativa spiegazione sono presentati nella tabella 2. Il modello complessivo da noi usato è rappresentato nello schema in figura 3. Come detto prima, il processo globale è composto da due CSTR linearizzate in parallelo, ognuna con un differente numero di Damkholer così da avere due differenti inerzie (la lunghezza del transitorio), uno rapido e uno più lento. Al fine di simulare un ambiente non-stazionario (come molti dei casi reali) abbiamo fatto sì che l'ingresso q (il flusso del reagente) del modello CSTR sia alla stregua di un disturbo, infatti non viene fornito al modello neurale per la modellazione. L'output complessivo del sistema, $s(t)$, è la somma delle uscite dei due modelli CSTR, il valore di ognuna di queste è normalizzato tra 0 ed 1 ed è campionata ogni 0,3 s.

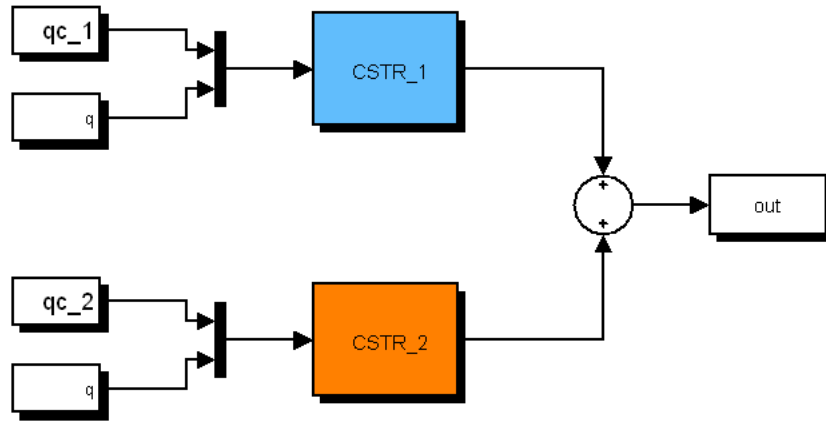


Fig. 6. Schema del modello

Le matrici di sistema del modello con “disturbo” sono le seguenti:

$$\mathbf{A} = \begin{pmatrix} -q_s - \Phi K & -\Phi x_{1s} K' & 0 \\ \beta \Phi K & -q_s - \delta - \beta \Phi x_{1s} K' & \delta \\ 0 & \delta \delta_1 \delta_2 & -\delta_1 q_{cs} - \delta \delta_1 \delta_2 \end{pmatrix} \quad (3)$$

$$\mathbf{B} = \begin{pmatrix} 0 & (x_{1f} - x_{1s}) \\ 0 & (x_{2f} - x_{2s}) \\ \delta_1 (x_{3f} - x_{3s}) & 0 \end{pmatrix} \quad (4)$$

Con $K = e^{\frac{x_{2s}}{1 + \frac{x_{2s}}{\gamma}}}$ e $K' = e^{\frac{x_{2s}}{1 + \frac{x_{2s}}{\gamma}}} \frac{1}{(1 + \frac{x_{2s}}{\gamma})^2}$. La portata del reattore q (disturbo) è modellata come un treno di impulsi di frequenza 0.1 Hz, ampiezza 0.15 e ampiezza dell'impulso del 40%. Ambedue i modelli CSTR hanno in ingresso un gradino ad ampiezza casuale variabile tra 0 e 0.5 ogni 6.5 secondi. Il gradino passa in un filtro con funzione di trasferimento $\frac{1}{s+1}$.

Table 2. Parametri del modello del CSTR

x_{1s}	0.2028	Punto di equilibrio della concentrazione
x_{2s}	5	Punto di equilibrio della temperatura di reazione
x_{3s}	0.4079	Punto di equilibrio della temperatura del cooling-jacket
x_{1f}	1	Concentrazione adimensionale dell'ingresso del reattore
x_{2f}	0	Temperatura adimensionale dell'ingresso del reattore
x_{3f}	-1	Temperatura adimensionale del cooling-jacket
Φ	0.25-0.53	Numero di Damkohler, descrive la velocità di reazione rispetto alla turbolenza
q_{cs}	0.9785	Punto di equilibrio della portata del fluido refrigerante
q_s	1	Punto di equilibrio della portata del reattore
β	8	Calore della reazione
δ	0.3	Coefficiente di trasferimento termico
δ_1	10	Rapporto tra il volume del reattore e del cooling-jacket
δ_2	1	Rapporto tra la densità di capacità termica tra il reattore e il cooling-jacket
γ	20	Energia di attivazione del processo

Il termine $\kappa(\cdot)$ rappresenta la velocità di reazione (adimensionale) corrispondente alla legge di Arrhenius:

$$\kappa(x) = e^{\frac{x}{1+\frac{x}{\gamma}}} \quad (5)$$

4 Risultati

L'uscita della rete neurale corrisponde alla predizione del segnale $s(t+h)$, con h pari all'orizzonte di predizione, gli ingressi della rete sono invece pari ai sei campioni passati del segnale $s(t-1), \dots, s(t-6)$. Tutti gli esperimenti sono stati svolti con un valore di orizzonte di predizione pari a 5. Abbiamo eseguito una serie di 10 test e calcolato il valore di RMSE medio:

$$E_{\text{rmse}} = \sqrt{\frac{\frac{1}{2} \sum_{i=1}^M (y(i) - \bar{y}(i))^2}{M}} \quad (6)$$

In tabella 3 è mostrato il confronto tra le performance di diverse tipologie di reti neurali: reti feed-forward, reti fully-connected e reti scale-free addestrate con algoritmi di vita artificiale e reti feed-forward addestrate con un classico algoritmo di back-propagation.

Alla fine dei test è stata effettuata una analisi della topologia delle reti con le migliori performance. Infatti il meccanismo di selezione naturale insito negli algoritmi evolutivi porta ad una drastica riduzione del numero di differenti topologie presenti nelle soluzioni della popolazione di individui. In figura 4 è visibile un esempio di tale meccanismo: sull'asse orizzontale è mostrato il tempo, sull'asse verticale ci sono le differenti topologie delle reti. Nel grafico è visibile chiaramente il processo che porta alla scomparsa della gran parte delle topologie delle reti

Table 3. Confronto tra le performance di diverse tipologie di reti neurali

Tipologia di Rete Neurale	RMSE
Rete feed-forward con ALIFE	0.039
Rete feed-forward con back-propagation	0.072
Rete fully-connected con ALIFE	0.039
Rete a topologia scale-free con ALIFE	0.041

di partenza. Un'analisi della connettività media (il parametro m delle tabelle 1 e 2.3) delle reti più efficienti mostra che, al termine di 10 simulazioni, tale valore è pari a 3.9, un risultato molto vicino alla media aritmetica (infatti il parametro m varia da 2 a 6). Questo semplice risultato sembra dimostrare che la connettività non influenzi la performance della rete neurale. È stata inoltre eseguita un'analisi sulla tipologia di connessione tra la rete ed i nodi di input e si è osservato che l'80% delle reti più efficienti usano una connessione con i nodi di input di tipo B (vedi la variabile `INPUT_CONNECTION_TYPE` nello pseudo-codice presentato in tabella 1) per questo motivo possiamo ipotizzare che l'informazione degli input distribuita su più nodi porta a migliori performance rispetto ad un'informazione "mirata" ad un insieme di piccoli nodi.

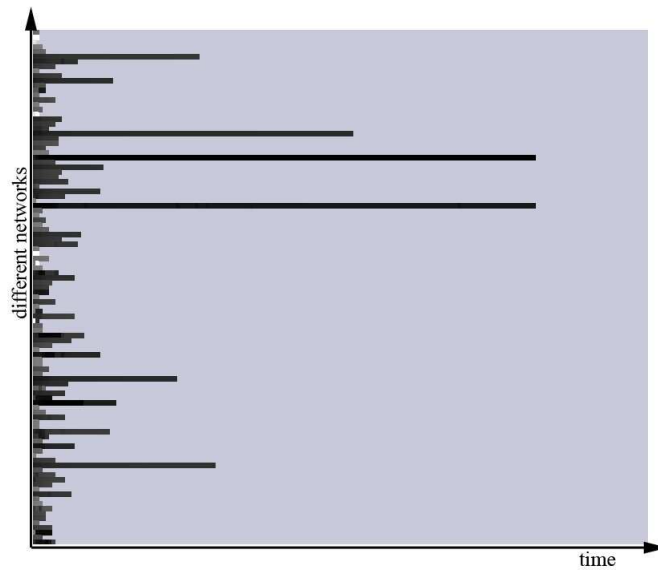


Fig. 7. Esempio del meccanismo di selezione naturale

5 Conclusioni

Le reti complesse come il modello scale-free proposto da Barabasi-Albert sono osservate in molti sistemi biologici e l'applicazione di queste topologie alle reti neurali porta a considerazioni interessanti. In questo articolo abbiamo presentato uno studio preliminare su come possano evolvere delle reti neurali con topologie complesse ed in particolare ci siamo concentrati sul modello scale-free. Come benchmark abbiamo affrontato il problema della modellazione di un processo chimico in cui sono presenti negli ingrossi sconosciuti (disturbo). L'ambiente evolutivo parte con una popolazione di individui con differenti reti scale-free nel loro genotipo ed al termine dell'esecuzione dell'algoritmo abbiamo osservato ed analizzato le reti con le migliori performance. Gli esperimenti oggetto di questo articolo mostrano che le performance delle reti complesse nell'affrontare il problema di modellazione proposto sono simili a quelle che si ottengono con le classiche reti feed-forward. La possibilità di effettuare analisi topologiche al termine del processo evolutivo può essere considerata la base per un'unione fruttuosa tra i due paradigmi per eccellenza dei sistemi cognitivi: le reti neurali e le reti complesse. Infatti l'analisi al termine dell'insieme dei test mostra d'un lato che la connettività ha poca influenza sulle performance della rete, mentre d'altro lato la connettività dei nodi di input sembra essere più influente. Questi risultati suggeriscono una direzione da seguire verso l'obiettivo di creare reti in grado di eseguire complessi compiti di modellazione.

References

1. Yao, X., "Evolving Artificial Neural Networks", *Proceedings of the IEEE*, 87(9): (1999)1423-1447
2. Alander, J. T., "An indexed bibliography of genetic algorithms and neural networks", Technical Report 94-1-NN, University of Vaasa, Department of Information Technology and Production Economics (1998)
3. Cant-Paz, E. and Kamath, C. , "An empirical comparison of combinations of evolutionary algorithms and neural networks for classification problems", *IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics* (2005) 915-927.
4. Watts DJ, Strogatz SH., "Collective dynamics of 'small-world' networks", *Nature*. 1998 Jun 4;393(6684):440-2.
5. Gully A. P. C. Burns, Malcolm P. Y Oung, "Analysis of the connectional organization of neural systems associated with the hippocampus in rats", *Philosophical Transactions of the Royal Society B: Biological Sciences*, Volume 355, Issue 1393, 29 Jan 2000, Pages 55 70
6. Victor M. Eguiluz, Dante R. Chialvo, Guillermo A. Cecchi, Marwan Baliki, and A. Vania Apkarian, "Scale-Free Brain Functional Networks", *Phys. Rev. Lett.* 94, 018102 (2005)
7. Dorogotvsev, S. N., Mendes, J. F. F., "Evolution of Networks", *Advances in Physics*, Vol. 51, n. 4, pp. 1079-1187, 2002
8. Barabasi, A.-L. and Albert R. 1999, "Emergence of scaling in random networks", *Science*, 286 509512.
9. P. Erdos, A. Renyi, "On random graphs", *Publ. Math. Debrecen*, 1959

10. Watts, D. J., "Small Worlds: The Dynamics of Networks Between Order and Randomness", Princeton University Press, 1999
11. Coven, R., Havlin, S., ben-Avraham, D., "Structural Properties of Scale-Free Networks, chap. 4 in "Handbook of graphs and networks", Eds. S. Bornholdt and H. G. Schuster, Wiley-VCH, 2002
12. H. Jeong, Z. Neda and A.-L. Barabasi, "Measuring preferential attachment for evolving networks", Euro. Phys. Lett. 61 567 (2003)
13. Langton, C., "Artificial Life", Addison-Wesley, Redwood City/CA, USA, 1989
14. Annunziato M., Bertini I., Lucchetti M., Pannicelli A., Pizzuti S. "Adaptivity of Artificial Life Environment for On-Line Optimization of Evolving Dynamical Systems", in Proc. EUNITE01, Tenerife, Spain, 2001
15. Annunziato M. , Bertini I. , Pannicelli A., Pizzuti S. , Tsimring L., "Complexity and Control of Combustion Processes in Industry", Proc. of CCSI 2000 Complexity and Complex System in Industry, Warwick, UK, 2000
16. Annunziato M. , Lucchetti M., Orsini G., Pizzuti S. , "Artificial life and on-line flows optimisation in energy networks", IEEE Swarm Intelligence Symposium, Pasadena (CA), USA, 2005
17. Annunziato M., Bertini I., Pannicelli A. and Pizzuti S. "A Nature-inspired-Modeling-Optimization-Control system applied to a waste incinerator plant", 2nd European Symposium NiSIS06, Puerto de la Cruz, Tenerife (Spain), 2006
18. Annunziato M., Bertini I., Pannicelli A., Pizzuti S. "Evolutionary Control and On-Line Optimization of an MSWC Energy Process", Journal of Systemics, Cybernetics and Informatics, Vol.4, Num. 4, 2006
19. Annunziato M., Bertini I., Iannone R., Pizzuti S. "Evolving feed-forward neural networks through evolutionary mutation parameters", 7th International Conference of Intelligent Data Engineering and Automated Learning (IDEAL 06), Burgos, Spain, 2006, 554-561
20. Annunziato M. ,Bruni C., Lucchetti M. , Pizzuti S. : "Artificial life approach for continuous optimisation of non stationary dynamical systems", Integrated Computer-Aided Engineering, vol. 10, n.2, 111-125, Giugno 2003
21. Russo L.P., Bequette B.W. "Impact of process design on the multiplicity behaviour of a jacketed exothermic CSTR", AIChE Journal, Vol. 41, 1995, pp 135-147
22. Saraf V.S., Bequette B.W., "Auto-tuning of cascade controlled open-loop unstable reactors", American Control Conference, 2001. Proceedings of the 2001, Vol.3, Iss., 2001, pp. 2021-2026
23. Russo, L.P. and Bequette, B.W., "State-Space versus Input/Output Representations for Cascade Control of Unstable Systems", Ind. Eng. Chem. Res., 36, 6, 1997, 2271 2278