



WORKING PAPER N.02.06

Maggio 2002

RETI NEURALI ARTIFICIALI PER LA VALUTAZIONE DI OPZIONI

M. Corazza^{a,b}

M. Gobbo^b

a. Università Ca' Foscari, Venice

b. GRETA, Venice.

RETI NEURALI ARTIFICIALI PER LA VALUTAZIONE DI OPZIONI

Marco CORAZZA

E-mail: corazza@unive.it

Michele GOBBO

E-mail: mgobbo@greta.it

Dipartimento di Matematica Applicata
Università Ca' Foscari di Venezia

G.R.E.T.A.
Venezia

ABSTRACT – In questo lavoro si considera un particolare strumento quantitativo non lineare e non parametrico, le cosiddette reti neurali artificiali, applicato alla valutazione di opzioni finanziarie. Inizialmente, si procede alla presentazione di un particolare tipo di rete neurale artificiale, il *multilayer perceptron* ad apprendimento supervisionato, evidenziandone caratteristiche fondamentali quali la struttura architettonica, la procedura di stima, nonché di settaggio e di sviluppo. Successivamente, dopo aver brevemente riassunto alcuni approcci seguiti da altri Autori, si applica lo strumento presentato alla valutazione di una particolare categoria di strumenti derivati: le opzioni finanziarie *call* di tipo europeo scritte sull'indice azionario inglese FTSE-100. Infine, si evidenziano vantaggi e limiti della modellistica proposta.

KEYWORDS – Opzioni finanziarie, reti neurali artificiali, *multilayer perceptron* ad apprendimento supervisionato, FTSE-100.

1 INTRODUZIONE

Le opzioni finanziarie sono strumenti derivati che hanno conosciuto negli ultimi decenni una rapida diffusione e che vengono utilizzate sia a fini di copertura rispetto a varie forme di rischio, sia a fini speculativi. Tuttavia, è innegabile che parte del loro iniziale successo e della loro successiva diffusione sia legato all'esistenza di un modello di valutazione semplice ed efficiente: la nota formula di valutazione in forma chiusa proposta da Black e Scholes (nel seguito: B-S) nel 1973 (per maggiori dettagli si veda [Black et al., 1973]), formula estesa in varie direzioni da Merton nello stesso anno (per maggiori dettagli si veda [Merton, 1973]).

Questa formula si basa su numerose ipotesi che, nella realtà, non sempre risultano soddisfatte. In particolare, molti studi empirici hanno evidenziato come la distribuzione dei rendimenti logaritmici di numerose serie finanziarie si discostino significativamente dalla quella Normale ipotizzata da Black e Scholes, presentando caratteristiche di leptocurtosi (cioè, le "code" della distribuzione risultano più spesse di quelle della Normale) e di asimmetria. In secondo luogo, il mercato in cui si opera è

caratterizzato da frizioni quali, ad esempio, i costi di transazione, l'impossibilità di operare allo scoperto, il fatto che le contrattazioni non avvengono con continuità,

La violazione di una, o più, delle ipotesi alla base del modello di B-S porta inevitabilmente quest'ultimo a commettere sistematici errori rispetto ai prezzi di mercato. La maggior parte degli studi empirici¹ condotti al fine di verificare la bontà della formula di B-S concorda nell'individuare prevalentemente due tipologie d'errore: una in riferimento al tempo mancante a scadenza, ed una in riferimento al prezzo d'esercizio (o, più precisamente, al rapporto tra il valore dell'attività sottostante ed il prezzo d'esercizio). In particolare, in quest'ultimo caso rappresentando in un diagramma le volatilità implicite dalla formula di B-S si ottengono quelli che sono noti in letteratura come effetti *smile*.

Ciò nonostante, la formula di B-S risulta utilizzata in pratica in virtù della sua semplicità: è necessaria, infatti, la stima di un solo parametro ignoto, la volatilità, ricavabile empiricamente dal mercato. Comunque, è da sottolineare come gli operatori tendano a "correggere" i valori ottenuti con il modello di B-S calibrando opportunamente i prezzi delle opzioni, con riferimento al tempo mancante a scadenza ed al prezzo d'esercizio, al fine di replicare l'effetto *smile*. Si può quindi osservare come gli operatori, nell'utilizzare la formula di B-S così modificata, stiano in realtà approssimando un più complesso modello per la valutazione di opzioni. Il problema è, ovviamente, individuare il modello che riesce a catturare meglio le caratteristiche dei prezzi di mercato.

Poiché è realistico congetturare che l'errore compiuto dal modello di B-S sia legato alla violazione contemporanea di più ipotesi, allora risulta preferibile un approccio alla valutazione che riduca al minimo le ipotesi di partenza. A tal fine, per la valutazione di opzioni finanziarie di tipo europeo in questo lavoro si considera un particolare modello non lineare e non parametrico: le reti neurali artificiali (nel seguito: RNA).

Il seguito di questo articolo è organizzato come segue. Nella sezione 2 illustriamo i principali metodi e strumenti relativi alle RNA. Nella sezione 3 proponiamo la specificazione di un modello il cui "motore" è costituito da una RNA ad apprendimento supervisionato. Nella sezione 4 introduciamo il problema della valutazione delle opzioni finanziarie mediante RNA. Nella sezione 5 presentiamo i risultati di una valutazione neurale di opzioni *call* di tipo europeo scritte sull'indice azionario inglese FTSE-100. Infine, nella sezione 6 si presentano anche alcune considerazioni ed alcune osservazioni finali.

2 IL MULTYLAYER PERCEPTRON AD APPRENDIMENTO SUPERVISIONATO: ASPETTI TEORICI ED OPERATIVI

In questa sezione si introducono i metodi e gli strumenti quantitativi propri della *soft-artificial intelligence* che prendono il nome di reti neurali artificiali - o reti neuronali - di tipo *MultyLayer Perceptron* (nel seguito: MLP) ad apprendimento supervisionato. In particolare, dapprima si provvederà a fornirne una sintetica presentazione di natura qualitativa e, successivamente, si provvederà a fornirne i principali elementi teorici.

In generale, la neurocomputazione è quella disciplina che, fra le altre sue potenzialità, prendendo ispirazione dalle discipline della neurobiologia *strictu sensu*,

¹ Si veda ad esempio [Rubinstein, 1994].

permette di inferire efficacemente le relazioni eventualmente esistenti tra variabili potenzialmente esplicative (le variabili di *input*) e variabili potenzialmente esplicanti (le variabili di *output*). Fin d'ora è da porre bene in evidenza che, nonostante la citata ispirazione ai principi di funzionamento dei sistemi nervosi propri degli esseri viventi superiori, la neurocomputazione è una disciplina quantitativa nel senso tradizionale del termine; pertanto, i metodi e gli strumenti offerti dalla neurocomputazione, al pari di quelli offerti dalle altre discipline quantitative, si fondano sul classico metodo scientifico deduttivo [Hertz *et al.*, 1991]. In particolare, le RNA di tipo MLP ad apprendimento supervisionato sono dei sistemi di elaborazione di dati e di informazioni caratterizzati da un elevato numero di elementi elaborativi (elementi detti "neuroni artificiali" o, più semplicemente, "nodi"); ognuno di questi elementi elaborativi, analogamente a quanto si verifica per i neuroni biologici,

- è in grado di effettuare poche e semplici operazioni di elaborazione,
- ed è collegato a molti altri elementi elaborativi dai quali riceve dei segnali di *input* od ai quali trasmette dei segnali di *output* determinati trasformando opportunamente dei segnali di *input* precedentemente ricevuti.

È anche da porre in evidenza che, in generale, il numero di neuroni artificiali che, in media, costituiscono una RNA di tipo MLP è un infinitesimo del numero (tipicamente, 10.000 miliardi) di quelli biologici che, in media, costituiscono il cervello di un essere vivente superiore.

Nonostante questa pallida analogia con i sistemi neurali biologici, le RNA di tipo MLP risultano comunque formalmente caratterizzate da proprietà analoghe a quelle che caratterizzano gli stessi sistemi neurali biologici, principali fra le quali:

- la capacità di inferire le relazioni (anche complesse) eventualmente esistenti tra segnali di *input* e segnali di *output* relativi ad uno stesso fenomeno; più in dettaglio, al fine di realizzare tali inferenze, non è necessario fornire alla RNA di tipo MLP particolari conoscenze a priori inerenti al fenomeno stesso, ma, più semplicemente, è sufficiente presentarle più volte delle coppie costituite da degli *input* e dai corrispondenti *output*: a partire dall'analisi di questi ultimi - mediante dei meccanismi formali di aggiornamento progressivo e convergente dei propri parametri - il modello neuronale "apprende" (seppure in forma non parametrica) le relazioni di cui in oggetto (proprietà nota come capacità di apprendimento);
- la capacità di modificare opportunamente la rappresentazione formale relativa a relazioni già "apprese" (secondo le modalità di cui al punto precedente) tra segnali di *input* e segnali di *output*, quando alla RNA di tipo MLP si presentino nuovi *pattern* di *input-output* (proprietà nota come capacità di adattamento o plasticità);
- la capacità di saper operare efficacemente e correttamente anche in presenza di segnali di *input* e/o di *output* fortemente distorti (proprietà nota come capacità di robustezza al rumore);
- e la capacità di saper correttamente associare a dei segnali di *input* mai "visti" in precedenza (ovvero, non utilizzati durante la fase di apprendimento) i corrispondenti segnali di *output* (proprietà nota come capacità di generalizzazione).

Queste, ed altre proprietà di cui godono le RNA di tipo MLP, spiegano il perché, in particolare in tempi recenti, la neurocomputazione abbia conseguito un evidente successo nell'ambito delle discipline socio-economiche. Infatti, a tal riguardo è da porre in evidenza come, in generale, molti degli "oggetti" studiati da queste discipline siano caratterizzati

- dalla inadeguatezza delle informazioni disponibili a priori e
- dalla pochezza delle relazioni funzionali che è possibile stabilire *ex ante*,

caratteristiche (entrambe negative) sostanzialmente affrontabili mediante il ricorso a metodologie e ad approcci che si ispirano alle capacità di *problem solving* tipiche degli esseri viventi superiori quali, per l'appunto, sono le RNA di tipo MLP.

2.1 Gli aspetti teorici di base

Nei loro tratti generali, le RNA di tipo MLP costituiscono degli interessanti e flessibili strumenti quantitativi in grado di rappresentare, in maniera tipicamente non parametrica, relazioni non lineari della forma:

$$\mathbf{y} = f(\mathbf{x}): A \subseteq \mathbf{R}^m \rightarrow B \subseteq \mathbf{R}^n$$

dove

A è lo spazio degli *input* e

B è lo spazio degli *output*.

Più precisamente da un punto di vista formale, modelli di questo tipo presentano la medesima struttura di grafi orientati ad archi pesati, i cui nodi siano disposti secondo uno strato di *input* (costituito da m nodi), uno o più strati intermedi (denominati strati nascosti od anche *hidden layer*) ed uno strato finale di *output* (costituito da n nodi). È da sottolineare il fatto che, in generale, il numero dei nodi presenti in ciascuno degli strati nascosti di un modello MLP non può essere determinato a priori. Ciascuno strato risulta, inoltre, completamente connesso con gli strati ad esso immediatamente successivi (lo strato di *input* con il primo strato nascosto, il primo strato nascosto con il secondo strato nascosto e così via fino all'ultimo strato nascosto con lo strato finale di *output*), senza alcuna connessione tra unità appartenenti ad un medesimo strato (denominate connessioni intrastrato). Grazie a questa morfologia strutturale, il segnale emesso in *input* viene propagato in modo unidirezionale dallo strato di *input* stesso, attraverso gli strati nascosti, fino allo strato finale di *output*.

Ciascun nodo della rete, ad eccezione degli m nodi appartenenti allo strato di *input*, è in grado di eseguire operazioni di computazione e di trasformazione. In particolare, due sono le funzioni che caratterizzano ognuno di questi nodi:

- la prima determina l'*input* del nodo, calcolando una "aggregazione" pesata dei segnali confluenti nel nodo medesimo. La funzione di "aggregazione" più comunemente utilizzata, a questo proposito, è la cosiddetta funzione somma, avente la forma seguente:

$$a_{l,k} = \sum_{i=1}^{I_{l-1}} w_{i,l,k} \tau_{i,l,k} - b_{l,k}, \quad l = 2, \dots, L \text{ e } k = 1, \dots, K_l$$

dove

$a_{l,k}$ è l'*input* del k -esimo nodo dell' l -esimo strato,

$w_{i,l,k} \in \mathbf{R}$, con $i = 1, \dots, I_{l-1}$, sono i pesi associati agli archi che connettono gli I_{l-1} nodi dell' $(l-1)$ -esimo strato con il k -esimo nodo dell' l -esimo strato,

$\tau_{i,l,k} \in \mathbf{R}$ sono i corrispondenti segnali che confluiscono nel nodo stesso provenienti da ciascuno dei nodi presenti nell' $(l-1)$ -esimo strato e

$b_{l,k} \in \mathbf{R}$ è un parametro di soglia denominato *bias* (per maggiori dettagli si veda, ad esempio, [Hecht-Nielsen, 1990]). Il *bias* può essere considerato come il peso associato ad un (fittizio) $(I_{l-1} + 1)$ -esimo arco confluyente nel nodo, vale a dire $w_{I_{l-1}+1,l,k} = b_{l,k}$ che propaga il segnale costante $\tau_{I_{l-1}+1,l,k} = -1$;

- la seconda funzione determina l'output del nodo, trasformando l'input $a_{l,k}$ nel valore di output $\tau_{k,l+1,h}$, con $h = 1, \dots, K_{l+1}$. In particolare, se si verifica che $l=L$, allora gli output considerati sono esattamente quelli associati ai nodi appartenenti allo strato di output, cioè sono gli n elementi del vettore di output del modello MLP preso in esame. Una delle funzioni di trasformazione che viene più comunemente utilizzata è la cosiddetta funzione di trasferimento (o di attivazione) logistica, avente la forma seguente:

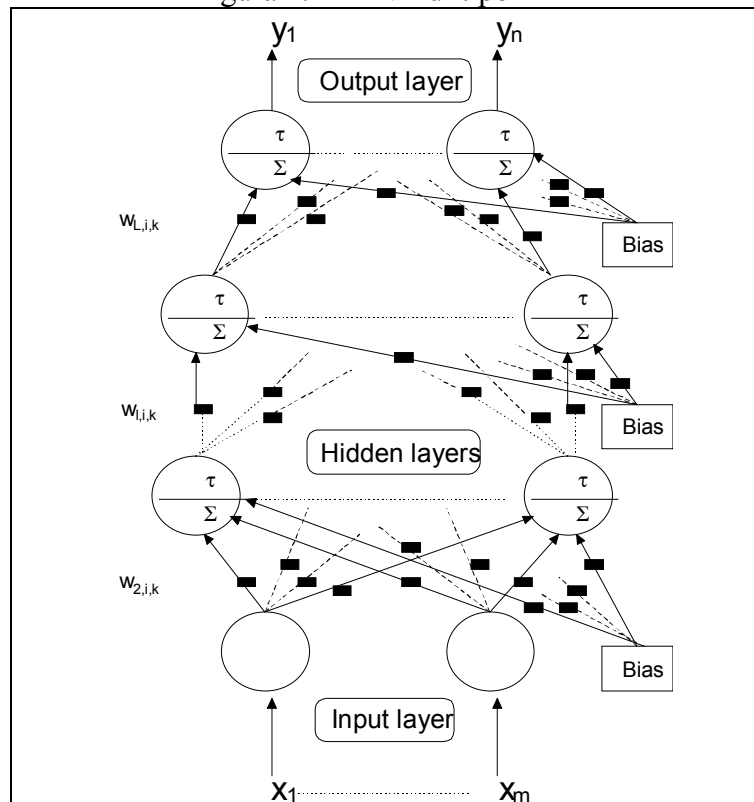
$$\tau_{k,l+1,h} = \frac{s_{l,k}}{1 + \exp(-g_{l,k}a_{l,k})}, \quad l = 2, \dots, L, \quad k = 1, \dots, K_l \text{ e } h = 1, \dots, K_{l+1},$$

dove

$s_{l,k} \in]0, +\infty[$ è un parametro denominato *shift* e

$g_{l,k} \in]0, +\infty[$ è un parametro denominato *gain*, entrambi associati al k -esimo nodo dell' l -esimo strato (per maggiori dettagli si veda, ad esempio, [Hecht-Nielsen, 1990]).

Figura 2.1 - RNA di tipo MLP



Degno di nota è il fatto che una funzione di questo tipo può essere considerata come una *smooth version* di un classificatore multicategoria.

La sostanziale abilità già dimostrata dai modelli MLP in varie tipologie di applicazioni è proprio dovuta alla presenza di una struttura fondamentale di questo tipo, espressamente diffusa e parallela. Naturalmente, in questa classe di modelli rivestono un ruolo essenziale i parametri liberi, cioè i pesi $w_{i,l,k}$, con $i = 1, \dots, I_{l-1}$, $l = 2, \dots, L$ e $k = 1, \dots, K_l$. Allo scopo di determinare i loro valori "ottimali" si utilizza un algoritmo fondato sulla metodologia (classica) di EBP.

2.2 L' algoritmo di *Error Back Propagation* (EBP)

In termini generali, i valori ottimali dei pesi associati ad un modello MLP sono determinati sulla base di una procedura estimativa iterativa, denominata algoritmo di addestramento (o di apprendimento). Gli algoritmi di addestramento maggiormente utilizzati si fondano sulla metodologia EBP, alla cui base, a sua volta, si pone la metodologia di Discesa Rapida del Gradiente (DRG). Nell'ambito di algoritmi di addestramento di questo tipo, dopo aver effettuato un'operazione di inizializzazione casuale dei pesi, le coppie di *input-output* appartenenti ad un prefissato insieme di dati $D = \{ \mathbf{x}_p, \mathbf{y}_p \}$, $p = 1, \dots, P$, vengono iterativamente presentate al modello MLP fintantoché, attraverso opportune operazioni di modifica (aggiornamento) dei pesi, non si raggiunga il minimo (assoluto) di una funzione di costo predefinita, funzione che esprima una misura della distanza che intercorre tra gli *output* desiderati e i corrispondenti *output* che il modello MLP calcola. Una tipica funzione di costo è la seguente:

$$E = \frac{1}{2P} \sum_{p=1}^P \sum_{j=1}^n \left[y_{p,j}^* \left(w_{1,1,1}, \dots, w_{I_{l-1}+1, L, K_L} \right) - y_{p,j} \right]^2 \quad (2.2.1)$$

dove

P è il numero delle coppie di *input-output* appartenenti al prefissato insieme di dati, n è la dimensione dello spazio di output,

$y_{p,j}^* \left(w_{1,1,1}, \dots, w_{I_{l-1}+1, L, K_L} \right)$ o, più semplicemente, $y_{p,j}^*$, con $j = 1, \dots, n$, sono gli elementi del p -esimo vettore degli output calcolati dal modello MLP (il cui valore dipende da quello assunto dai pesi di iterazione in iterazione) e

$y_{p,j}$ sono, infine, gli elementi del p -esimo vettore degli output desiderati.²

In modo particolare, l'aggiornamento dei pesi si ottiene retropropagando opportunamente il valore assunto dalla funzione di costo, cioè il livello di errore negli output, dallo strato di output, attraverso gli strati nascosti, fino allo strato iniziale di input, nel modo seguente:

$$w_{i,l,k}(t+1) = w_{i,l,k}(t) + \Delta w_{i,l,k}(t+1) = w_{i,l,k}(t) - \eta \left. \frac{\partial E}{\partial w_{i,l,k}} \right|_{(\mathbf{x}_1, \mathbf{y}_1), \dots, (\mathbf{x}_P, \mathbf{y}_P)},$$

$$i = 1, \dots, I_{l-1} + 1, \quad l = L, \dots, 2, \quad k = 1, \dots, K_l \quad \text{e} \quad t = 0, \dots, T,$$

dove

$w_{i,l,k}(q)$ è il valore, alla q -esima iterazione (od epoca) dell'algoritmo, del peso associato all'arco che connette l' i -esimo nodo dell' $(l-1)$ -esimo strato con il k -esimo nodo dell' l -esimo strato,

$\Delta w_{i,l,k}(q)$ è il valore, alla medesima iterazione, dell'aggiornamento del peso stesso,

$\eta \in]0,1]$ è un parametro denominato tasso di apprendimento,

² Per questi motivi, tali procedure iterative di stima sono anche denominate procedure (a carattere *batch*) di apprendimento supervisionato.

$\left. \frac{\partial E}{\partial w_{i,l,k}}(q) \right|_{(x_1, y_1), \dots, (x_P, y_P)}$ è il valore, sempre alla stessa iterazione, assunto dalla derivata

parziale della funzione di costo rispetto al peso stesso e

T , infine, è il numero di iterazioni richieste dall'algorithmo d'addestramento (numero che risulta implicitamente definito da un'opportuna regola di *stop-learning* fissata inizialmente).

In particolare, è possibile provare che la forma funzionale dell'aggiornamento parametrico

$$\left. \frac{\partial E}{\partial w_{i,l,k}}(q) \right|_{(x_1, y_1), \dots, (x_P, y_P)} \quad \text{con } i = 1, \dots, I_{l-1} + 1, \quad l = L, \dots, 2, \quad k = 1, \dots, K_l \text{ e } q = 1, \dots, T + 1,$$

è diversa se si considerano pesi associati ad archi che confluiscono in nodi dello strato finale di *output* (cioè se $l=L$) oppure pesi i cui archi confluiscono in nodi appartenenti ad uno degli strati nascosti (cioè se $l = L - 1, \dots, 2$).

La procedura iterativa di stima utilizzata per eccellenza può essere riassunta nel modo che segue:

Step 1 Prendi in considerazione l'insieme di dati D contenente le coppie di *input-output*.

Step 2 Inizializza $t = 0$ ed inizializza casualmente il valore dei pesi $w_{i,l,k}(t)$, con $i = 1, \dots, I_{l-1} + 1$, $l = L, \dots, 2$ e $k = 1, \dots, K_l$.

Step 3 Aggiorna il valore dei parametri retropropagando il livello d'errore così come descritto nella procedura (a carattere batch) di apprendimento supervisionato.

Step 4 Aggiorna t ponendo $t_{\text{ser}} = t + 1$ e $t = t_{\text{ser}}$.

Step 5 Se il criterio di *stop-learning* non è soddisfatto allora ritorna allo Step 3 altrimenti termina.

In un algoritmo di apprendimento come questo è particolarmente rilevante il ruolo assunto dalla regola di arresto dell'apprendimento. La versione classica di regola *stop-learning* risulta soddisfatta a quella T -esima iterazione dell'algorithmo di apprendimento in cui il livello d'errore rilevato $E(T)$ non sia superiore al valore di una prefissata soglia suddetta \bar{E} . Ad una scelta procedurale di questo tipo, tuttavia, può accompagnarsi un fenomeno sfavorevole denominato iperapprendimento, consistente nella possibilità che l'algorithmo di apprendimento individui relazioni funzionali, tra i vettori di *input* \mathbf{x} ed i vettori di *output* \mathbf{y} delle coppie *input-output* appartenenti al prefissato insieme di dati D , in realtà inesistenti. Allo scopo di evitare il manifestarsi di possibili inconvenienti di questo tipo, alcuni autori suggeriscono di utilizzare un criterio di *stop-learning* basato sulla Metodologia di Discesa Concorrente dell'errore (MDC) (ad esempio si veda [Hecht-Nielsen, 1990] e [Hertz *et al.*, 1991]). In un approccio così definito, l'insieme di dati iniziale contenente le coppie di *input-output* viene, prima di tutto, convenientemente suddiviso in due sottoinsiemi ad intersezione vuota, il *training set* (D_T) ed il *testing* (o *validation*) set (D_V) e, successivamente, l'algorithmo di apprendimento viene applicato iterativamente al *training set* D_T , finché non sia raggiunto il minimo (assoluto) della funzione di costo calcolata sul *testing set* D_V .

3 LA COSTRUZIONE DEL MODELLO NEUROCOMPUTAZIONALE

Secondo quanto richiesto all'approccio quantitativo di cui in analisi e date le caratteristiche dei metodi e degli strumenti neurocomputazionali considerati (così come presentati nella sezione 2), in questa terza sezione si propone la puntuale specificazione di un articolato modello il cui "motore" è costituito da una RNA ad apprendimento supervisionato di tipo MLP. Più in dettaglio, il seguito di questa sezione si propone la specificazione vera e propria del modello neuronale.

In generale, nell'ambito dell'implementazione e dello sviluppo di una efficace RNA di tipo MLP ad apprendimento supervisionato, vi sono vari aspetti ai quali si deve obbligatoriamente prestare attenzione, aspetti quali:

- quelli provenienti dalla teoria della neurocomputazione come, ad esempio, la metodologia di discesa rapida del gradiente (ad esempio si riveda l'algoritmo EBP alla sottosezione 2.2),
- e quelli emergenti durante la fase di implementazione operativa, come, ad esempio, il problema della selezione di un'opportuna classe di modelli neurali capaci di affrontare lo specifico problema considerato.

Con riferimento all'analisi qui in via di sviluppo, l'insieme di tali aspetti può essere raggruppato nelle tre seguenti principali classi operative:

- una prima classe in cui si prende in considerazione la selezione delle variabili di *input* e di quelle di *output* e si prendono in considerazione le eventuali operazioni di pre- e post-processamento dei dati;
- una seconda classe operativa in cui esamina l'aspetto del *design* architetturale del modello;
- ed una terza ed ultima classe in cui si prevede l'analisi della procedura di *training* e di quella di *testing*.

Tabella 3.1 - Classi operative

<u>Classe 1</u> Selezione degli <i>input</i> e degli <i>output</i> Pre- e post-processamento dei dati	- Selezione delle variabili di <i>input</i> ed <i>output</i> - Processamento dei dati <ul style="list-style-type: none"> • Filtraggio dei dati • Trasformazione dei dati
<u>Classe 2</u> <i>Design</i> del modello MLP	- Strutture architettoniche <ul style="list-style-type: none"> • Struttura della regione nascosta • Numero dei nodi presenti nello strato di <i>input</i> e di <i>output</i> e negli strati nascosti • Morfologia delle connessioni
<u>Classe 3</u> Procedure di <i>training</i> e di <i>testing</i>	- Selezione delle funzioni di trasferimento - Procedure di <i>training</i> <ul style="list-style-type: none"> • Algoritmo di <i>training</i> • Gestione del tasso di apprendimento e del <i>momentum</i> - Regole di <i>Stop Learning</i> - Inizializzazione casuale - <i>Shuffling</i> dei dati

In particolare, con riferimento ad una RNA di tipo MLP ad apprendimento supervisionato, l'insieme degli aspetti elencati è opportunamente dettagliato nella Tabella 3.1.

È da porre in evidenza come l'obiettivo principale che si sta perseguendo, ovvero la specificazione, la progettazione e l'implementazione di un modello neuronale capace di investigare opportunamente il fenomeno indagato, sia in completo accordo con il carattere specifico dei metodi e degli strumenti neurocomputazionali.

In relazione al perseguimento dell'obiettivo qui in analisi, si è deciso, in accordo con la letteratura specializzata (si veda, ad esempio, [Hecht-Nielsen, 1990] e [Hertz *et al.*, 1991]), di "privilegiare" alcuni degli aspetti operativi elencati nella precedente tabella secondo i tre seguenti distinti livelli di "attenzione": un livello di attenzione media (primo livello), un livello di attenzione medio-alta (secondo livello) ed un ultimo livello di attenzione alta (terzo livello).

Attenzione a livello medio

In primo luogo, si è dedicata attenzione a livello medio

- alle operazioni di selezione delle variabili di *input* e di quelle di *output* e
- alle operazioni di processamento dei dati.

Gli aspetti relativi ad entrambe queste operazioni vengono trattati in dettaglio nella sezione 4.

Attenzione a livello medio-alto

In secondo luogo, si è prestato un livello di attenzione medio-alto

- alle operazioni di selezione delle funzioni di trasferimento, ciò sempre in conformità alle principali linee guida presenti, al riguardo, nella letteratura specializzata. In particolare, per ogni nodo-processore della RNA di tipo MLP ad apprendimento supervisionato (cioè per tutti i suoi nodi ad eccezione di quelli di *input*) si utilizza la classica funzione logistica (quella presentata nella sottosezione 2.1) nella quale si pone $s_{l,k} = g_{l,k} = 1$, con $l = 2, \dots, L$ e $k = 1, \dots, K_l$. È da porre in evidenza come alcuni Autori, in relazione all'aggiornamento dei pesi nel corso della fase di *training*, suggeriscano di utilizzare la seguente versione modificata della regola di apprendimento supervisionato (sempre di tipo *batch*):

$$w_{i,l,k}(t+1) = (1 + \beta)w_{i,l,k}(t) - w_{i,l,k}(t-1) - (1 - \beta)\eta \frac{\partial E}{\partial w_{i,l,k}}(t+1) \Bigg|_{(x_1, y_1), \dots, (x_P, y_P)}$$

$$i = 1, \dots, I_{l-1} + 1, l = L, \dots, 2, k = 1, \dots, K_l \text{ e } t = 1, \dots, T$$

dove

$\beta \in [0,1]$ è un parametro denominato *momentum* (o fattore di smussamento) in base al quale l'addestramento può risultare più rapido, grazie all'introduzione di un meccanismo di memoria dei valori assunti dai pesi al termine delle precedenti iterazioni dell' algoritmo di apprendimento.

È prestato un livello medio-alto di attenzione anche

- alla gestione del tasso di apprendimento η .

In accordo alle metodologie classiche (ad esempio si veda [Hecht-Nielsen, 1990]), nel corso della fase di addestramento il tasso di apprendimento deve essere sottoposto con continuità ad azioni di aggiustamento (che ne facciano aumentare o diminuire il valore), ciò allo scopo di ottimizzare il processo di apprendimento stesso del modello neuronale considerato.

Anche

- alle operazioni di inizializzazione casuale e di *shuffling* dei dati (ovvero di riordinamento casuale dei *pattern* di *input-output*)

si è dato un livello di attenzione medio-alto. In particolare, alcuni Autori propongono di addestrare una RNA di tipo MLP ad apprendimento supervisionato sulla base di una serie di diverse configurazioni casuali iniziali, cosicché sia possibile procedere allo sviluppo di più fasi di addestramento, i cui esiti possano in seguito venir tra loro confrontati, allo scopo di prevenire e/o porre rimedio all'insorgere di possibili difficoltà, quali, ad esempio, quelle connesse a situazioni in cui si raggiunga soltanto un minimo relativo della funzione di costo considerata, anziché uno assoluto (questo fenomeno è generalmente noto sotto la denominazione di *Local Minimum Pitfall* o LMP). Anche lo *shuffling* dei dati è una proposta che viene comunemente utilizzata nella letteratura specializzata al fine di prevenire il verificarsi di problemi derivanti da uno sviluppo scorretto dell'addestramento, quali, ad esempio, l'apprendimento di relazioni funzionali non corrette tra le variabili di *input* e quelle di *output*.

Attenzione a livello alto

Infine, è stato riservato un livello di attenzione alta ai due seguenti aspetti:

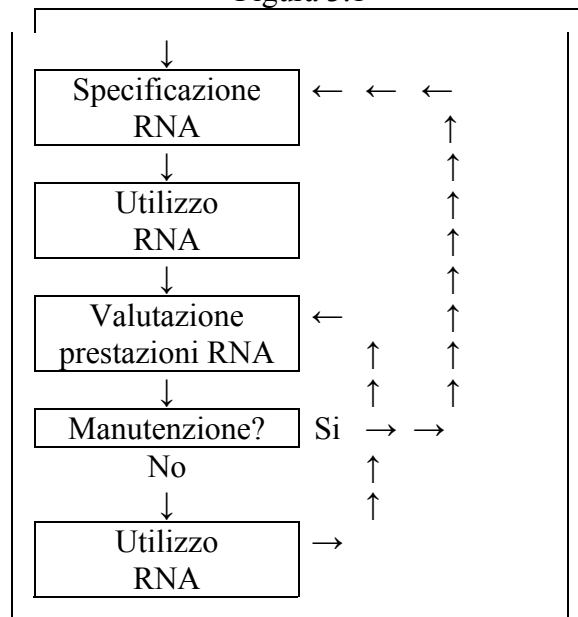
- il *design* architetturale del modello neuronale e
- la regola di *stop-learning*,

ciò perché entrambi questi aspetti si pongono come elementi essenziali nell'ambito della progettazione modellistica (si veda, ad esempio, [Hecht-Nielsen, 1990]).

Per ciò che riguarda il primo aspetto, cioè la struttura architetturale del modello neuronale, si intende considerare attentamente il dimensionamento di una regione "interna" del modello stesso costituita da un unico strato nascosto di nodi, operazione questa che costituisce uno dei punti critici della progettazione di una rete MLP. A tal riguardo si intende specificare e sottoporre a verifica differenti soluzioni architetture.

In quanto al secondo aspetto meritevole di un livello di attenzione alta, cioè la regola di *stop-learning*, si è inteso verificare la "bontà" operativa di una versione opportunamente modificata della regola di *stopping* di tipo MDC presentata nella sottosezione 2.2.

Figura 3.1



In conclusione è da notare che una RNA di tipo MLP non è per sempre. In altri termini, affinché il suo funzionamento prosegua correttamente ed efficacemente nel

tempo necessita di un'opportuna "manutenzione". Con quest'ultima affermazione si intende porre in evidenza il fatto che il modello neuronale, al fine di perseguire con continuità nel tempo i propri obiettivi, necessita di essere sottoposto a periodici nuovi addestramenti. In particolare, la finalità di questi nuovi addestramenti risiede nell'esigenza di "aggiornare" la conoscenza della RNA di tipo MLP sugli eventuali cambiamenti strutturali che, relativamente al fenomeno indagato, potrebbero essere intercorsi dal periodo dell'ultimo aggiornamento. Ovviamente, in sede di nuovo addestramento deve essere effettuata *ex novo* tutta la specificazione del modello neuronale così come presentata in questa sezione. Il "ciclo della vita" del modello neuronale qui proposto si può rappresentare graficamente mediante la diagrammazione a blocchi riportata in Figura 3.1.

Riassumendo, sono molteplici le RNA di tipo MLP da implementare ed indagare al fine di pervenire alla costruzione del modello neuronale ottimale. In particolare, date le varie opzioni operative da considerare così come presentate nella Tabella 3.1, è agevole evincere che questa fase possa essere una di quelle che richieda un elevato numero di risorse sia umane, sia computazionali.

4 LE RNA PER LA VALUTAZIONE DELLE OPZIONI FINANZIARIE

In questa sezione si introduce il problema della valutazione delle opzioni finanziarie mediante RNA.

Con riferimento al problema di cui in oggetto, i vantaggi dell'approccio neurale risiedono nella non necessità di particolari ipotesi a priori, lasciando che sia la RNA ad inferire la dinamica eventualmente esistente tra le variabili di interesse ed il valore dell'opzione. In secondo luogo, l'approccio non parametrico riduce il problema dell'errata specificazione del modello e si può adattare a cambiamenti strutturali nel processo generatore dei dati. Infine, mediante tale approccio si ottiene una formula di valutazione in forma chiusa che risulta relativamente facile da trattare. Tuttavia, tali modelli possono porre problemi di interpretabilità dei risultati ottenuti.

4.1 Introduzione alla valutazione di opzioni finanziarie

Prima di affrontare nello specifico lo sviluppo della RNA secondo i punti precedentemente elencati, rivediamo brevemente la derivazione della formula di Black e Scholes. I due Autori, sulla base di determinate "condizioni ideali", derivano un'espressione in forma chiusa per determinare il prezzo di una opzione *call* di tipo europea scritta su di un'azione. Le ipotesi sulle quali si basa lo specifico modello da noi considerato sono le seguenti:

- il tasso d'interesse a breve termine è noto e costante per tutta la durata dell'opzione,
- non si considerano costi di transazione,
- si può dare e prendere a prestito denaro illimitatamente ed allo stesso tasso d'interesse a breve termine,
- si può operare allo scoperto e
- il prezzo del bene sottostante si evolve in modo continuo secondo il seguente andamento diffusivo

$$dS = \mu S dt + \sigma S dz$$

dove

S indica il prezzo del bene sottostante al tempo corrente,

μ indica il rendimento istantaneo atteso del bene sottostante,

t indica la variazione del tempo,

σ indica lo scarto quadratico medio (misura della volatilità) del rendimento istantaneo del bene sottostante,

z indica un particolare processo di Markov, noto come processo di Wiener.

Quest'ultima ipotesi sulla dinamica del prezzo dell'attività sottostante è particolarmente importante, in quanto implica che, a scadenza, il prezzo del bene sottostante segua una distribuzione di probabilità di tipo log-Normale e, di conseguenza, che i rendimenti istantanei si distribuiscono Normalmente con varianza costante nel tempo.

Il valore dell'opzione *call* nel caso il titolo sottostante paghi un tasso di dividendo nel tempo uguale a q , parametro supposto noto e costante, è:

$$C_{B-S} = S_t e^{-q\tau} \Phi(d_1) - X e^{-r\tau} \Phi(d_2)$$

dove

$$\Phi(d) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^d e^{-0.5t^2} dt,$$

$$d_1 = \left[\log(S_t/X) + (r - q + \sigma^2/2)\tau \right] / (\sigma\sqrt{\tau}),$$

$$d_2 = d_1 - \sigma\sqrt{\tau}.$$

Gli *input* del modello di B-S per la valutazione di opzioni su indici azionari sono quindi:

- il prezzo del bene sottostante,
- il prezzo d'esercizio,
- il tempo mancante a scadenza,
- il tasso d'interesse privo di rischio utilizzato,
- il tasso di dividendo e
- la volatilità.

Lo sviluppo di una RNA ai fini della valutazione di opzioni finanziarie è un problema complesso da affrontare in quanto molti sono gli elementi da prendere in considerazione e che influenzano in modo significativo la prestazione finale della RNA; nella sezione 3 si è proposto lo schema che si segue nel proseguo.

4.2 La specificazione delle variabili di *input* e di *output*

L'individuazione delle variabili di *input* e di quelle di *output* nel caso dell'applicazione delle RNA alla valutazione di opzioni finanziarie è un'operazione relativamente semplice. Un punto di partenza è dato dalla stessa formula di B-S che individua cinque variabili (nel caso standard) che influenzano la variabile in *output* data dal prezzo dell'opzione. Non sorprende, quindi, che in letteratura la maggior parte degli approcci individui le variabili di *input* tra le variabili elencate precedentemente. Si veda ad esempio [Hutchinson *et al.*, 1994], in cui gli Autori considerano il valore del sottostante, il prezzo d'esercizio³ ed il tempo mancante a scadenza. In [Lajbcygier *et al.*, 1997] e [Anders *et al.*, 1998] si prende in considerazione anche il tasso d'interesse e varie stime della volatilità; altri Autori ancora propongono come ulteriore variabile esplicativa

³ Queste due variabili vengono congiuntamente considerate mediante il rapporto S/X , detto *moneyness* (per maggiori dettagli si veda la sotto-sezione successiva).

l'*open interest*, indicatore che evidenzia le posizioni aperte con riferimento ad un particolare contratto d'opzione.

In termini generali, possiamo indicare la funzione che la RNA va ad apprendere nel modo seguente:

$$C_t = f(S_t, X, T-t, \sigma_t, r_t, d) \quad (4.2.1)$$

L'analisi dei risultati ottenuti dai precedenti Autori sono tutti incoraggianti nell'individuare le RNA come un valido strumento di valutazione alternativo alla classica formula di B-S.

4.3 La fase di pre-processamento dei dati

Consideriamo ora le operazioni necessarie alla costruzione del *set* di dati da utilizzare. Come evidenziato, la fase di processamento dei dati prevede:

- la selezione di un opportuno sottoinsieme dei dati disponibili (operazione di filtraggio dei dati);
- l'opportuna ridefinizione dei *pattern* di *input* e di quelli di *output* (operazione di trasformazione dei dati).

La prima operazione è necessaria al fine di escludere, secondo opportuni criteri, esempi ritenuti non significativi e, quindi, aventi scarso contenuto informativo. I criteri utilizzati sono stati i seguenti (per maggiori dettagli si veda [Anders *et al.*, 1998]):

- vengono escluse le opzioni eccessivamente *deep-in-the-money* oppure *deep-out-of-the-money* perché, generalmente, risultano scarsamente trattate;
- vengono escluse le opzioni con scadenza inferiore ad un prefissato numero di giorni; infatti, tali opzioni risultano trattate nel mercato al loro valore intrinseco e possono implicare errori rispetto ai prezzi teorici;
- vengono escluse le opzioni trattate a prezzi troppo bassi; infatti, si può empiricamente notare come, soprattutto per i prezzi delle opzioni più vicini allo zero, vi sia un numero elevato di opzioni trattate allo stesso valore. Quindi, in questi casi la RNA può avere, delle difficoltà nell'associare a *pattern* di *input* anche molto diversi fra di loro lo stesso output⁴.

Le variabili che influenzano il prezzo di un'opzione, avente come attività sottostante un indice azionario, secondo il modello B-S sono sei. In particolare, dalla relazione (4.2.1) si ha che il valore di un'opzione *call* di tipo europeo (C_t) è funzione del valore dell'indice (S_t), del prezzo di esercizio (X), del tempo mancante a scadenza ($T-t$), della volatilità (σ), del tasso d'interesse privo di rischio (r) e del tasso di dividendo (d). In particolare, la formula B-S può essere riscritta considerando le seguenti semplificazioni:

- si utilizza come tasso d'interesse privo di rischio la differenza tra l'originario tasso d'interesse privo di rischio ed il tasso di dividendo, cioè $r-d$;
- si utilizza il rapporto tra il valore dell'indice ed il prezzo d'esercizio S_t/X , ciò al fine di determinare il rapporto tra il prezzo dell'opzione ed il prezzo d'esercizio C_t/X .

Pertanto, gli *input* effettivamente necessari sono quattro. In particolare, l'ultima semplificazione considerata si ottiene ipotizzando che la distribuzione del rendimento dell'attività sottostante l'opzione sia indipendente dal livello del prezzo dell'attività stessa. Secondo il Teorema 9 enunciato da Merton in [Merton, 1973] questo implica che la funzione $f(\cdot)$ sia omogenea di grado 1 rispetto al valore del sottostante ed al prezzo

⁴ Tale criterio ha come ulteriore vantaggio anche quello di ridurre in modo drastico il totale esempi.

d'esercizio. Poiché la formula B-S rientra in tale categoria, utilizzando tali *input* per la RNA molti Autori ipotizzano che anche la formula di valutazione dell'opzione lo sia. Le motivazioni alla base di tale semplificazione sono le seguenti:

- viene utilizzata una RNA con nodo di *input* in meno, riducendo il numero dei parametri e quindi la complessità del problema,
- si ritiene che questo permetta alla RNA di apprendere più facilmente la struttura della volatilità implicita di mercato (il così detto effetto *smile* si rappresenta graficamente mettendo in relazione la volatilità implicita e, appunto, il rapporto S_t/X).

Quindi, la funzione (incognita) che andiamo ad approssimare è:

$$\frac{C_t}{X} = f\left(\frac{C_t}{X}, T-t, \sigma_{IMP}, r-d\right).$$

Ciascun esempio sottoposto alla rete sarà, quindi, composto da cinque valori, di cui quattro relativi all'*input* ed 1 relativo all'*output* desiderato.

4.4 La fase di apprendimento

Le RNA di tipo MLP ad apprendimento supervisionato considerate prevedono un solo strato nascosto, caratterizzato da un numero "opportuno" di nodi. L'*output* generato dalla rete sulla base degli *input* fornite è determinato come segue:

$$\frac{C_{NN,t}}{X} = \beta_0 + \sum_{j=1}^q g\left(\gamma_{0j} + \frac{S_t}{X}\gamma_{1j} + (T-t)\gamma_{2j} + \sigma_{IMP}\gamma_{3j} + (r-d)\gamma_{4j}\right)\beta_j$$

dove

q indica il numero dei nodi nascosti,

β_0 , β_j e γ_{ij} , con $j = 1, \dots, n$ e con $i = 1, \dots, 4$, indicano i parametri che caratterizzano la RNA e

$g(\cdot)$ indica la funzione di trasferimento.

Le funzioni di trasferimento considerate sono quella logistica per i nodi dello strato nascosto e quella lineare per il nodo di *output*.

Come funzione di costo si assume la (2.2.1) e come algoritmo di ricerca del punto di minimo (assoluto) si utilizza quello di Levenberg-Marquand. Nella fase applicativa il problema dell'*overfitting* non è stato quasi mai riscontrato; la ragione di ciò può ragionevolmente essere imputata al filtraggio iniziale fatta sui dati al fine di eliminare gli esempi meno significativi, ed alle caratteristiche dei dati oggetto di studio che, in sostanza, si possono considerare poco rumorosi.

5 L'APPLICAZIONE ED I RISULTATI

Presentiamo ora alcuni risultati tratti da [Billio *et al.*, 2001]. L'impostazione seguita è quella delineata precedentemente nelle sezioni 3 e 4. In particolare i dati si riferiscono ad opzioni *call* di tipo europeo scritte sull'indice azionario FTSE-100 per il periodo che va dal 04.01.2000 al 24.12.2000 (per un totale di 7741 esempi). Da tale insieme di *pattern* se ne sono opportunamente estratti 2500 utilizzati nella fase di addestramento e di selezione della RNA, esempi che a loro volta sono stati suddivisi come segue nei seguenti insiemi di dati:

- *training set*: insieme composto da 2000 esempi scelti casualmente dall'insieme delle opzioni trattate dal 04.01.2000 al 31.10.2000; tali dati vengono utilizzati al fine di stimare i valori dei pesi liberi delle RNA;
- *validation set*: insieme composto da 500 esempi scelti casualmente dall'insieme delle opzioni trattate dal 04.01.2000 al 31.10.2000 una volta estratto il *training set*; tali dati vengono utilizzati al fine di individuare il momento in cui terminare la fase di apprendimento della RNA;
- *running set*: insieme composto da esempi costruiti a partire dalle opzioni trattate dal 01.11.2000 al 24.12.2000; tali esempi vengono utilizzati al fine di verificare le prestazioni della RNA in fase applicativa.

Il tempo mancante a scadenza è stato calcolato facendo riferimento ai soli giorni lavorativi.

I dati sui tassi d'interesse utilizzati al fine di ottenere una buona stima di tale parametro sono i seguenti tassi interbancari: tasso a brevissimo termine (una settimana), tassi a 1, 3, 6, 12 mesi⁵. Il tasso d'interesse che si accorda con il tempo a scadenza di ciascuna opzione viene ottenuto interpolando linearmente i due tassi interbancari più vicini.

Il tasso di dividendo è stato ricavato dal sito Internet della borsa londinese, il *London Stock Exchange*⁶. Nella nostra applicazione consideriamo tale parametro costante non solo per tutta la vita residua della singola opzione, ma anche per tutto il periodo considerato e, quindi, per tutte le opzioni valutate⁷.

Infine, la stima della volatilità è ottenuta implicitamente dai prezzi di mercato delle opzioni medesime. In particolare, è stata effettuata una media semplice delle volatilità implicite legate alle due opzioni i cui prezzi di esercizio sono più vicini al prezzo del bene sottostante, data una certa scadenza.

Andiamo ora a presentare le prestazioni in termini di RMSE delle RNA addestrate, ciò al fine di individuare, sulla base dell'errore commesso, la RNA più idonea ad essere utilizzata a fini applicativi. Si ricorda che le prove sono state effettuate considerando diverse configurazioni iniziali dei pesi, come suggerito dalla letteratura.

Tabella 5.1 - RMSE · 10⁻³

N. di nodi dello strato nascosti	RMSE sul <i>training set</i>	RMSE sul <i>validation set</i>
2	3.796	3.680
3	1.818	1.909
4	1.597	1.736
5	1.459	1.604
6	1.440	1.604
7	1.376	1.514
8	1.341	1.451
9	1.331	1.466

Come si può vedere dalla terza colonna della Tabella 5.1, la rete che ottiene i risultati migliori nel *validation set* è quella rete caratterizzata da otto nodi nello strato nascosto (nel seguito: Rete 8) che, quindi, sarà quella utilizzata a fini applicativi.

⁵ In particolare, viene utilizzato il *London Interbank Offer Rate* (LIBOR) sulla sterlina.

⁶ L'indirizzo *web* di tale sito è: "<http://www.londonstockexchange.com>".

⁷ Infatti, si è osservato che durante il periodo considerato il parametro non ha subito grandi oscillazioni, mantenendosi vicino al livello considerato.

Come si può vedere dalla Tabella 5.2, l'errore compiuto dal modello neurale sul *running set* risulta minore non solo dell'errore compiuto dal modello di B-S, ma anche dell'errore compiuto dalla stessa RNA quando calcolato sul *training set*. Verosimilmente, questo risultato è dovuto al fatto che il *running set* è composto da un numero limitato di opzioni e soprattutto appartenenti a particolari categorie.

Tabella 5.2 - RMSE nella fase applicativa

	Rete 8	B-S
<i>Running set</i>	1.155	4.242

Dopo aver verificato che le prestazioni *out of sample* sono soddisfacenti, andiamo ad analizzare le prestazioni *in sample*, ovvero il comportamento del modello neurale nell'insieme di dati utilizzati nella sua fase di implementazione e sviluppo. Al fine di verificare se le prestazioni del modello varino in funzione del valore assunto dagli *input*, si suddividono come di seguito indicato le opzioni considerate nel periodo che va dal 04.01.2000 al 31.10.2000:

- con riferimento al rapporto tra il valore dell'indice ed il prezzo di esercizio, individuando cinque categorie (Tabella 5.3),
- con riferimento al tempo mancante alla scadenza, individuando tre categorie (Tabella 5.4).

Nel presentare i risultati ottenuti proponiamo anche un confronto con i corrispondenti errori compiuti dal modello B-S, ciò al fine di permetterne una migliore comprensione e lettura.

Tabella 5.3 - RMSE con riferimento a *S/X*

	Rete 8	B-S
<i>Deep-in-the-money</i>	1.543	7.603
<i>In-the-money</i>	1.404	6.460
<i>Near-the-money</i>	1.241	3.263
<i>Out-of-the-money</i>	1.445	4.595
<i>Deep-out-of-the-money</i>	1.531	9.169
Totale <i>In-sample</i>	1.429	6.402

Tabella 5.4 - RMSE con riferimento al tempo mancante a scadenza

	Rete 8	B-S
Breve termine	1.311	3.764
Medio termine	1.415	6.560
Lungo termine	1.529	7.759
Totale <i>In-sample</i>	1.429	6.402

L'analisi dei risultati riportati nelle Tabelle 5.1-5.4 porta alle seguenti considerazioni:

- le prestazioni della RNA nell'analisi *in the sample* sono sempre migliori di quelle del modello di B-S per tutte le categorie di opzioni considerate;
- il comportamento risulta qualitativamente simile per i due modelli; l'errore tende ad aumentare all'aumentare del tempo mancante a scadenza e tende ad essere più elevato per prezzi di esercizio che si allontanano dal valore dell'indice.

Inoltre, l'analisi condotta non sembra evidenziare, con riferimento alle varie categorie di opzioni considerate, una particolare struttura dell'errore, non rivelando una tendenza, da parte della RNA, a generare valori delle opzioni sistematicamente inferiori o superiori a quelli di mercato. Piuttosto, l'errore compiuto può essere imputato in buona

parte a "rumore", particolarmente presente nel caso di opzioni scarsamente trattate quali, appunto, quelle *deep-in-the-money* oppure quelle *deep-out-of-the-money*.

In conclusione, i risultati si possono considerare sicuramente soddisfacenti ed in linea con quanto ottenuto dagli altri Autori citati in questo lavoro.

6 CONSIDERAZIONI ED OSSERVAZIONI FINALI

L'approccio neurocomputazionale per la valutazione di opzioni finanziarie rappresenta una valida alternativa a tecniche di valutazione più classiche quali quelle basate su espressioni in forma chiusa (ove disponibili) e a tecniche numeriche proposte in letteratura (ad esempio si veda [Boyle *et al.*, 1997] per una rassegna sulle tecniche di simulazione di tipo Monte Carlo e si veda [Wilmott *et al.*, 1995] per un'introduzione ai metodi alle differenze ed agli elementi finiti).

Comunque, tali metodi neurocomputazionali presentano alcune limitazioni, quali

- la richiesta di una considerevole quantità di dati per ottenere dei risultati soddisfacenti;
- il problema dell'*overfitting* a causa del quale il modello tende a recepire come strutturale anche la parte "rumorosa" dei dati (tuttavia, come sottolineato precedentemente tale problema può essere limitato attraverso un'opportuna fase di *shuffling* dei dati);
- inoltre, poiché le RNA sono delle quasi-*black box*, non risulta sempre finanziariamente interpretabile la relazione che lega ciascuna variabile esplicativa al valore dell'opzione e, di conseguenza, la sua significatività ai fini della valutazione.

Particolarmente promettente sembra poi il cosiddetto "approccio ibrido", in cui la RNA è chiamata ad apprendere solamente le deviazioni dei prezzi di mercato rispetto ad un modello parametrico che si è dimostrato sufficientemente accurato per le opzioni considerate (quale, ad esempio, il modello di B-S). Un'altra tipologia di approccio ibrido consiste nel fornire alla RNA degli input elaborati preventivamente da altri modelli. In particolare, essendo la stima della volatilità un *input* di fondamentale importanza, un possibile approccio potrebbe prevedere l'utilizzo di un particolare modello parametrico per ottenere delle migliori stime della volatilità. Ad esempio, la modellistica di tipo ARCH e GARCH ed i modelli a volatilità stocastica possono essere in grado di catturare importanti caratteristiche delle serie storiche finanziarie, quali l'effetto *cluster* e l'effetto *leverage* come, ad esempio, illustrato in [Bollerslev *et al.*, 1992].

Futuri possibili sviluppi di tale approccio potrebbero considerare:

- l'utilizzo di *input* aggiuntivi, o diversi, da quelli che si basano sulla formula di B-S;
- la specificazione di particolari misure di *performance* collegate al fenomeno analizzato al fine di specificare RNA che assicurino una buona applicabilità pratica (soprattutto in relazione alle operazioni di copertura);
- la definizione di un'analisi complessiva del fenomeno che consideri, oltre al prezzo, anche altre caratteristiche della funzione di valutazione (tipicamente i parametri greci).

BIBLIOGRAFIA

- [1] ANDERS U., KORN O. E SCHIMITT C., Improving the Pricing of Options: a Neural Network Approach, *Journal of Forecasting*, 17, 369-388, 1998.

- [2] BELCARO P.L., CANESTRELLI E. e CORAZZA M., Artificial Neural Network Forecasting Models: an Application to the Italian Stock Market, *Badania Operacyjne i Decyzje*, 3/4, 29-48, 1996.
- [3] BILLIO M., CORAZZA M. e GOBBO M., Modelli Neuronali e Modelli Switching Regime per la Valutazione di Opzioni Finanziarie, *Quaderno del Dipartimento di Matematica Applicata - Università Ca' Foscari di Venezia*, 102/2001, 2001.
- [4] BOLLERSLEV T., CHOU R.Y. e KRONER K.F., ARCH Modeling in Finance: a Review of the Theory and Empirical Evidence, *Journal of Econometrics* 52, 5-59, 1992.
- [5] BOYLE P., BROADIE M., GLASSERMAN P., Monte Carlo Methods for Security Pricing, *Journal of Economic Dynamics and Control* 2, 1267-1321, 1997.
- [6] BLACK F. e SCHOLES M., The Pricing of Options and Corporate Liabilities, *Journal of Political Economy*, 81, 637-659, 1973.
- [7] FELDMAN K. e KINGDON J., Neural Networks and some Applications to Finance, *Applied Mathematical Finance*, 2, 17-42, 1995.
- [8] HANKE M., Option Pricing Using Neural Networks vs. Black/Scholes: an Empirical Comparison of Two Fundamentally Different Option Pricing Methods, *Journal of Computational Intelligence in Finance*, 5(1), 1999.
- [9] HANKE M., Neural Network Approximation of Option Pricing Formulas for Analytically Intractable Option Pricing Models, *Journal of Computational Intelligence in Finance*, 5(5), 1997.
- [10] HECHT-NIELSEN R., *Neurocomputing*. Addison-Wesley, 1990.
- [11] HERTZ J., KROGH A. e PALMER R., *Introduction to the theory of neural computing*, Addison-Wesley, 1991.
- [12] HORNIK K., STINCHCOMBE M. e WHITE H., Multilayer Feedforward Networks are Universal Approximators, *Neural Networks* 2, 359-366, 1989.
- [13] HULL J.C., *Opzioni, Futures e Altri Derivati*, Il Sole 24 Ore Libri, 2000.
- [14] HUTCHINSON J.M., LO A. e POGGIO T., A Nonparametric Approach to Pricing and Hedging Derivatives Securities via Learning Networks, *Journal of Finance*, 49, 851-889, 1994.
- [15] LAJBCYGIER P.R. e CONNOR J.T., Improved Option Pricing Using Artificial Neural Networks and Bootstrap Methods, *International Journal of Neural System*, 8(4), 457-471, 1997.
- [16] MALLIARIS M. e SALCHENBERGHER L., Neural Networks for Predicting Options Volatility, in Trippi R.R., Turban E.: "Neural Networks in Finance and Investing".
- [17] MERTON R.C., Theory of Rational Option Pricing, *Bell Journal of Economics and Management Science* 4, 141-183, 1973.
- [18] RUBINSTEIN M., Implied Binomial Tree, *Journal of Finance* 49, 771-818, 1994.
- [19] WILMOTT P., DEWYNNE J. e HOWISON S. D., *The Mathematics of Financial Derivatives: a Student Introduction*, Cambridge University Press, 1995.