

ALMA MATER STUDIORUM · UNIVERSITÀ DI  
BOLOGNA

---

SCUOLA DI SCIENZE  
Corso di Laurea in Informatica per il Management

## Reti neurali in ambito finanziario

Relatore:  
Chiar.ma Prof.ssa  
Elena Loli Piccolomoni

Presentata da:  
Alessandro Fabbri

Anno Accademico 2018/2019

# Indice

<b>Introduzione</b>	<b>4</b>
<b>1 Breve introduzione alle reti neurali</b>	<b>5</b>
1.1 Alcuni cenni storici . . . . .	5
1.2 Reti Neurali Biologiche e Artificiali . . . . .	7
1.3 Applicazioni . . . . .	8
1.4 Classificazione delle reti neurali . . . . .	9
1.5 Tipo di algoritmo di apprendimento . . . . .	9
1.5.1 Apprendimento supervisionato . . . . .	9
1.5.2 Apprendimento non supervisionato . . . . .	10
1.5.3 Apprendimento per rinforzo . . . . .	10
1.6 Struttura . . . . .	10
1.6.1 Struttura <i>Feed-forward</i> . . . . .	11
1.6.2 Struttura ricorrente . . . . .	12
<b>2 Reti neurali in ambito finanziario</b>	<b>13</b>
2.1 Previsioni di serie temporali . . . . .	13
2.1.1 Scelta delle variabili di input . . . . .	14
2.1.2 Scelta del livello di apprendimento . . . . .	14
2.1.3 Scelta dell'orizzonte temporale di riferimento per la previsione . . . . .	14
2.2 Classificazione e Discriminazione . . . . .	14
2.3 Approssimazione di funzioni . . . . .	14
2.4 Costruzione della rete neurale . . . . .	15
2.4.1 Individuazione dell'obiettivo . . . . .	15
2.4.2 Costruzione del set di dati . . . . .	17
2.5 Apprendimento . . . . .	18
2.5.1 Suddivisione temporale dell'archivio dei dati . . . . .	19
2.5.2 Numero di strati nascosti e numero di neuroni . . . . .	19
2.5.3 Connessione tra gli strati differenti . . . . .	20
2.5.4 Funzione di attivazione . . . . .	21

2.5.5	Regole di apprendimento . . . . .	22
2.5.6	Aggiornamento dei pesi di connessione dei neuroni . . .	22
2.5.7	Indicatore di errore . . . . .	23
2.6	Generalizzazione . . . . .	23
<b>3</b>	<b>Reti neurali per la previsione dei prezzi in finanza</b>	<b>24</b>
3.1	Reti neurali per la previsioni dei prezzi . . . . .	24
3.1.1	Backpropagation . . . . .	24
3.1.2	Radial Basis Function . . . . .	25
3.1.3	General Regression . . . . .	25
3.1.4	Support Vector Machines . . . . .	25
3.1.5	Risultati . . . . .	26
3.2	Rete neurale per la costruzione di un portafoglio di titoli . . .	27
	<b>Conclusioni</b>	<b>31</b>

# Elenco delle figure

1.1	Confronto tra elementi biologici e artificiali . . . . .	8
1.2	I due tipi di struttura feed-forward . . . . .	11
1.3	Confronto tra rete ricorrente e una rete feed-forward . . . . .	12
2.1	Architettura a uno strato nascosto con connessioni standard . . . . .	20
2.2	Architettura a uno strato nascosto con connessioni a salto . . . . .	20
2.3	Architettura a uno strato nascosto con connessioni ripetute . . . . .	21
3.1	Risultati dei 5 modelli . . . . .	26
3.2	Media e deviazione standard per 100 processi di addestramento . . . . .	26
3.3	Procedimento di ottimizzazione . . . . .	28
3.4	Struttura della rete . . . . .	29
3.5	Risultati (MAE e MAPE) . . . . .	29
3.6	Risultati (Regressione) . . . . .	30
3.7	Confronto fra valori stimati e valori reali . . . . .	30

# Introduzione

Le reti neurali artificiali sono oggi diventate uno strumento estremamente efficace nell'analisi di situazioni non "predicibili" analiticamente, e per questo motivo si prestano in maniera estremamente efficace a modellare problemi di varia natura.

In particolare in questo lavoro cercherò di analizzare l'utilizzo di reti neurali in ambito economico-finanziario in quanto alcuni dei temi che si riscontrano in economia ben si prestano ad un'analisi attraverso le reti neurali.

In particolare nel primo capitolo di questo elaborato descriverò le origini delle reti neurali e alcuni criteri attraverso i quali oggi si classificano le reti stesse. Nel secondo capitolo mi occuperò invece di approfondire quali sono i passaggi da seguire al fine di costruire una rete neurale concentrandomi sulla risoluzione di problemi legati all'ambito economico-finanziario.

Infine, nell'ultimo capitolo, mi dedicherò all'analisi di due articoli ("*Neural networks for stock price prediction*"[4] e "*Forecasting Portfolio Optimization using Artificial Neural Network and Genetic Algorithm*"[3]) nei quali vengono confrontati i risultati ottenuti tramite l'utilizzo di diversi tipi di reti neurali accennando anche ad approcci diversi attraverso algoritmi di *machine learning*.

# Capitolo 1

## Breve introduzione alle reti neurali

### 1.1 Alcuni cenni storici

Nello studio di alcune tipologie di problemi è possibile rendersi conto di come, in alcuni casi, sia particolarmente difficile elaborare algoritmi di tipo analitico in grado di risolvere tali problemi. In particolare esistono problemi, come ad esempio il riconoscimento dei volti, che sono in apparenza molto semplici per un essere umano ma risultano di particolare difficoltà quando devono essere risolti da un elaboratore.

Da qui l'idea di approcciare i problemi cercando di imitare i meccanismi degli esseri viventi più evoluti utilizzando anche le crescenti conoscenze sulle dinamiche biologiche alla base dell'elaborazione dell'informazione.

Le più antiche teorie sul funzionamento del cervello e dei processi mentali sono state concepite dai filosofi greci, in particolare da Platone e Aristotele. In seguito queste stesse teorie furono riprese da Cartesio e, successivamente, nel XVIII secolo dai filosofi empiristi.

Le prime realizzazioni di macchine cibernetiche, categoria alla quale appartengono i sistemi neurali, appaiono verso il 1940, col nascere di una scienza nuova, la cibernetica, che viene definita come "scienza che studia i processi intelligenti" e viene fondata da Norbert Wiener nel 1947. In realtà i primi a parlare realmente di reti neurali e, in particolare, di neuroni artificiali sono stati il neurofisiologo Warren Sturgis McCulloch e il matematico Walter Pitts con la pubblicazione nel 1943 di "A logical calculus of the ideas immanent in nervous activity".

In questa pubblicazione venivano presentati i neuroni artificiali come dei modelli di neuroni biologici facenti parte di reti caratterizzate da un neurone

in entrata ed un neurone in uscita. Inoltre, in questo particolare modello, veniva richiesto che tutti i dati fossero binari. Attraverso tale modello si era in grado di calcolare delle semplici funzioni booleane quali la funzione AND e la funzione OR.

Successivamente, nel 1948, Ross Ashby, un altro padre della cibernetica, costruisce l'omeostato, uno dei primi sistemi con connessioni interne regolabili, capace di variare la sua configurazione interna adattandola a stimoli esterni. Nel 1949 lo psicologo canadese Donald Olding Hebb, nel libro "The organization of behaviour", cerca di spiegare i complessi modelli di funzionamento del cervello ipotizzando i primi meccanismi di apprendimento delle reti neurali. A lui si deve l'apprendimento *hebbiano*, uno dei fondamentali algoritmi di apprendimento che si basa sul principio secondo il quale se due neuroni si attivano contemporaneamente, allora la loro interconnessione deve essere rafforzata.

Nel 1958 Von Neumann, nella sua opera "The computer and the brain", confronta il cervello umano con la tecnologia degli elaboratori del tempo, osservando come le reti neurali non potessero essere precise nello svolgere operazioni complesse a causa della mancanza di strutture.

Nello stesso anno Frank Rosenblatt del Cornell Aeronautical Laboratory sviluppa l'idea del cosiddetto *perceptron*, un'entità caratterizzata da un *layer* di ingresso ed uno di uscita, la cui regola di apprendimento si basa sulla minimizzazione dell'errore. Il sistema confronta l'uscita effettiva prodotta dalla rete rispetto a quella prevista e, di conseguenza, aggiorna i pesi delle varie connessioni. È con questa eccezionale scoperta che si inizia a intuire la potenzialità di un sistema in grado di adattarsi alla realtà e quindi di apprendere.

Alcuni anni dopo, nel 1969, Minsky e Papert pubblicano il testo "An introduction to computational geometry", nel quale vengono dimostrati i limiti delle reti proposte da Rosenblatt in grado di calcolare solamente funzioni linearmente separabili. Il fatto che, date queste limitazioni, neanche una funzione booleana semplice come l'XOR fosse realizzabile, porta ad un forte rallentamento nelle ricerche in questo campo.

Passati alcuni anni, si ha un nuovo impulso nelle ricerche, dopo che Hopfield, nel 1982, rilancia lo studio cercando di confutare le tesi di Minsky e Papert. Nel 1986, David E. Rumelhart, G. Hinton e R. J. Williams danno vita al famoso algoritmo di addestramento di *Backpropagation* il quale prevede la correzione dei pesi delle connessioni fra i nodi in base alla correttezza o meno del risultato della rete. Si iniziano poi ad introdurre dei livelli intermedi fra i neuroni di ingresso e uscita (detti anche nascosti o *hidden layer*) e, tramite un processo che sfrutta la discesa di un gradiente, si riesce così a addestrare la rete affinché riesca a trovare il minimo locale di una funzione appartenente

ad un particolare spazio. In questa maniera la rete acquista la capacità di calcolare delle risposte plausibili anche per degli input che la rete stessa non ha mai visto. Attraverso questa tecnica è stato così possibile superare la limitazione delle funzioni linearmente separabili, potendo finalmente includere l'XOR, donando un nuovo impulso allo studio di questa tipologia di reti. [2]  
[1]

## 1.2 Reti Neurali Biologiche e Artificiali

Le reti neurali artificiali sono costruite in modo da cercare di imitare, per quanto possibile, il funzionamento di quelle biologiche.

L'elemento fondamentale di una rete neurale biologica è il neurone il quale è composto da tre elementi fondamentali:

- soma: è il corpo centrale del neurone
- dendriti: rappresentano l'ingresso e sono fibre minori che si ramificano a partire dal soma e attraverso le sinapsi, giunzioni di comunicazioni, raccolgono gli input da neuroni afferenti e li propagano verso il soma
- assone: è la fibra principale che parte dal soma e si allontana da esso per portare l'output ai neuroni vicini.

Questi neuroni possono trovarsi in due stati: attivo o a riposo.

Il soma è in grado di elaborare una somma pesata dei segnali in ingresso e, se il risultato supera un certo valore di soglia, allora il neurone si attiva e produce un "potenziale di azione" che viene inviato all'assone. Quando invece il risultato non supera la suddetta soglia, il neurone rimane in uno stato di riposo. Ogni neurone è normalmente connesso ad un migliaio di altri neuroni. In un normale cervello umano il numero di sinapsi può superare  $10^{14}$ . Grazie alla presenza di un numero molto elevato di neuroni e sinapsi in grado di operare contemporaneamente l'effettiva potenza di elaborazione risulta essere molto elevata.

Il neurone artificiale è un elemento ideato per comportarsi in modo simile a quello biologico. Di questi ne esistono diversi tipi a seconda del modello matematico adottato. Tale modello descrive il modo nel quale il segnale in uscita varia in funzione dei segnali in ingresso. Un insieme di questi neuroni artificiali compone una rete neurale che per quanto sopra detto può essere vista come un insieme di funzioni matematiche non lineari che trasformano un insieme di variabili indipendenti  $x = (x_1, \dots, x_d)$ , definiti ingressi della rete, in un insieme di variabili dipendenti  $y = (y_1, \dots, y_c)$ , dette uscite della rete. La precisa forma di queste funzioni dipende dalla struttura interna della rete



e da un insieme di valori  $w = (w_1, \dots, w_d)$  chiamati pesi.

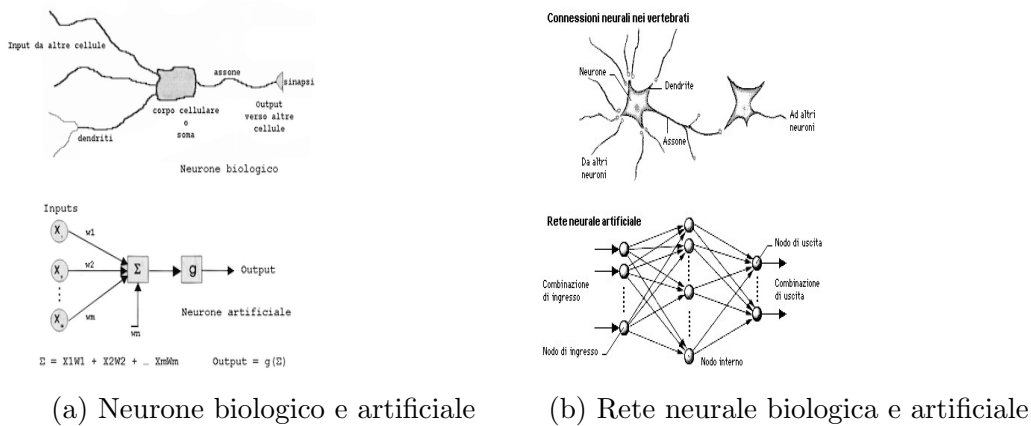


Figura 1.1: Confronto tra elementi biologici e artificiali

## 1.3 Applicazioni

È importante sottolineare come gli ambiti di applicazione delle reti neurali siano numerosi ed eterogenei. Ad esempio possono essere utilizzate per controllare il flusso del traffico in una rete stradale, per modellare il trasporto di beni e persone, nell'ambito delle telecomunicazioni, della logistica, della manutenzione preventiva. Più in generale tali reti vengono solitamente adoperate in quei contesti nei quali i dati possono essere parzialmente errati oppure ove non esistano modelli analitici in grado di risolvere un problema. In questo senso un loro tipico utilizzo è nei *software* di OCR, nei sistemi di riconoscimento facciale e nei sistemi che si occupano di trattare dati soggetti a errori o rumore. Le reti neurali vengono oggi utilizzate anche come strumento di previsione delle condizioni meteorologiche e, negli ultimi anni, è aumentata notevolmente la loro importanza anche nella bio-informatica, nella quale sono usate per la ricerca di *pattern* funzionali e strutturali di proteine e acidi nucleici.

Tuttavia, in questa sede si prenderà in esame esclusivamente l'ambito economico-finanziario.

Più in generale, in questo campo le applicazioni sono adatte a risolvere problemi di previsione, classificazione, controllo e ottimizzazione, grazie alla loro estrema compatibilità con i modelli di scelta discreta, fondati sulla teoria microeconomica delle utilità casuali. Inoltre, secondo la classificazione proposta dalla "International Conference on Neural Networks in the Capital Markets",

queste reti trovano un utilizzo nella gestione dei portafogli, nella stima dei modelli di curva dei rendimenti, nella valutazione dei titoli obbligazionari e azionari, oltre che nelle strategie di *trading*, di copertura e di arbitraggio e, infine, nella previsione. In quest'ultimo caso, l'oggetto dell'analisi può riguardare i prezzi azionari dei titoli obbligazionari o quelli dei contratti *futures*, dei tassi di interesse o quelli di cambio.

## 1.4 Classificazione delle reti neurali

Vista la grande tipologia di reti neurali, non esiste un vero e proprio sistema di classificazione. Due criteri sono maggiormente utilizzati in questo senso: il tipo di algoritmo di apprendimento e il tipo di struttura della rete.

## 1.5 Tipo di algoritmo di apprendimento

Il criterio più utilizzato è quello relativo al tipo di algoritmo di apprendimento. L'apprendimento può essere supervisionato, non supervisionato o per rinforzo.

### 1.5.1 Apprendimento supervisionato

Si parla di apprendimento supervisionato quando si ha a disposizione un set di dati comprendente esempi tipici di input con i relativi output. Analizzando questi dati la rete può imparare ad inferire la relazione che li lega in modo da aumentare la sua capacità di previsione. Fra le varie metodologie utilizzate per migliorare le prestazioni della rete nella fase di apprendimento spesso si utilizza la tecnica di *backpropagation*. Tale tecnica usa questi dati allo scopo di modificare i pesi ed altri parametri della rete stessa, così da minimizzare l'errore di previsione relativo all'insieme di addestramento. Se l'addestramento ha successo, la rete impara a riconoscere la relazione incognita che lega le variabili d'ingresso a quelle d'uscita, e diventa in grado di fare previsioni anche laddove l'uscita non sia nota a priori. L'obiettivo finale dell'apprendimento non è solo la previsione del valore d'uscita per ogni valore valido dell'ingresso, basandosi soltanto su un numero limitato di esempi di corrispondenza (coppie di valori input-output), ma soprattutto quello di identificare una regola generale che colleghi i dati in ingresso con quelli in uscita, così da evincere un metodo riutilizzabile per altri compiti simili. Questo tipo algoritmo di apprendimento viene utilizzato principalmente per problemi di classificazione, riconoscimento e regressione.

## 1.5.2 Apprendimento non supervisionato

A differenza dell'apprendimento supervisionato, questo tipo di apprendimento assume di avere solamente un set di dati in ingresso senza alcuna indicazione del risultato desiderato. L'algoritmo cercherà quindi di classificare i dati sulla base di caratteristiche comuni per cercare di effettuare ragionamenti e previsioni sugli input successivi. Gli algoritmi che appartengono a questa classe non sempre sono efficienti, poiché molto dipende dalla tipologia dei dati in ingresso. In particolare, si cerca di utilizzare dati numerici, in quanto, se vengono forniti dei dati non numerici, che sono difficilmente ordinabili e quindi difficili da confrontare, non è detto che l'algoritmo trovi un risultato soddisfacente. Lo scopo di questo secondo metodo di apprendimento è quello di trovare schemi e modelli nascosti, ossia identificare negli input una struttura logica, senza che questi siano preventivamente etichettati. Questo tipo di apprendimento viene utilizzato principalmente per problemi di *clustering* e regole di associazione.

## 1.5.3 Apprendimento per rinforzo

L'apprendimento per rinforzo è un caso particolare di apprendimento supervisionato nel quale non si conosce l'esatto risultato desiderato, poiché non vengono presentate le coppie input-output di esempi noti, ma si fornisce solo feedback sull'esito positivo o negativo di una risposta. Si basa solo sulle informazioni per stabilire se l'output effettivo è vicino o meno alla stima. L'apprendimento per rinforzo è una procedura di apprendimento che premia la rete neurale per il suo buon risultato di output o la punisce per il cattivo risultato. Questo tipo di rete presenta un processo di apprendimento più lento. L'apprendimento di rinforzo viene utilizzato nei casi in cui non è possibile specificare *pattern* di ingresso-uscita, come per i sistemi ad apprendimento supervisionato, ma viene fornito un rinforzo al sistema, che lo interpreta come un segnale positivo/negativo sul suo comportamento e aggiusta i propri parametri di conseguenza.

## 1.6 Struttura

Un'ulteriore classificazione delle reti neurali può essere fatta in base al tipo di architettura, cioè in base al modo nel quale sono collegati tra di loro i neuroni. Le strutture principali sono *Feed-forward* e ricorrente.

### 1.6.1 Struttura *Feed-forward*

Le reti *feed-forward* possono essere di due tipi *Single-layer feed-forward* e *Multi-layer feed-forward* (figura 1.2). Le reti neurali con flusso in avanti sono state tra le prime ad essere implementate e sono anche quelle con struttura tra le più semplici. In una rete di questo tipo le connessioni tra le unità non formano cicli e le informazioni si muovono solo in una direzione partendo dai nodi d'ingresso, attraversando nodi nascosti, se presenti, fino ad arrivare ai nodi d'uscita. Le reti *feed-forward* non hanno memoria di input avvenuti in tempi precedenti, per cui l'output è determinato solamente dall'attuale input.

#### *Single-layer feed-forward* (a propagazione in avanti)

Il *single layer perceptron* (perceptrone a singolo livello) è la più semplice architettura *feed-forward*. La struttura di questa rete è formata semplicemente da un *layer* d'ingresso seguito direttamente da uno d'uscita, in modo tale che i dati si propagano in un unico senso, entrando in un punto e uscendo da un altro. La vera differenza di questa rete rispetto alle altre è dovuta al fatto di essere dotata di un unico *layer* che effettua l'elaborazione dei dati e non presenta nodi nascosti. Inoltre, data la sua semplicità dovuta allo scarso numero di connessioni, l'architettura risulta essere molto limitata, essendo capace di combinare i dati in ingresso una sola volta.

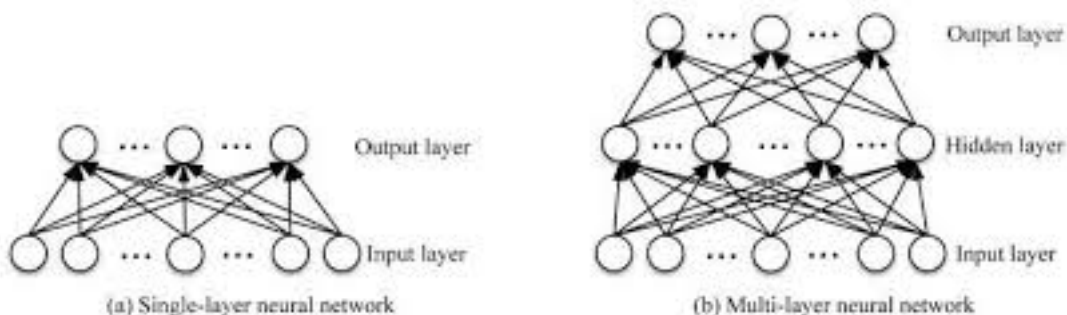


Figura 1.2: I due tipi di struttura feed-forward

#### *Multi-layer feed-forward* (propagazione in avanti a più livelli)

Questa classe di reti *feed-forward* si distingue dalla precedente per il fatto che tra il *layer* di input e quello di output si possono trovare uno o più *layer* di neuroni, nascosti detti *hidden*. Ogni *layer* ha connessioni in entrata e in uscita ed è connesso con il precedente e con quello successivo, proprio

come nelle reti *single-layer*. Anche in questo caso la propagazione del segnale avviene in avanti senza cicli e senza connessioni trasversali. Questo tipo di struttura fornisce alla rete una prospettiva globale, in quanto aumentano le interazioni tra neuroni e di conseguenza possono essere combinate più informazioni.

## 1.6.2 Struttura ricorrente

Una rete ricorrente si distingue dalle precedenti in quanto ciclica. Questo significa che i valori di uscita di uno strato di un livello superiore vengono utilizzati come valori d'ingresso per uno strato inferiore. Tale caratteristica è molto importante poiché permette l'utilizzo di uno degli strati come memoria di stato, consentendo di fornire dei valori temporali in ingresso e di modellarne un comportamento dinamico rispetto alle informazioni ricevute in precedenza. Tale caratteristica rende questo tipo di architettura adatta a compiti di analisi predittiva su sequenze di dati, come il riconoscimento della grafia o quello vocale.

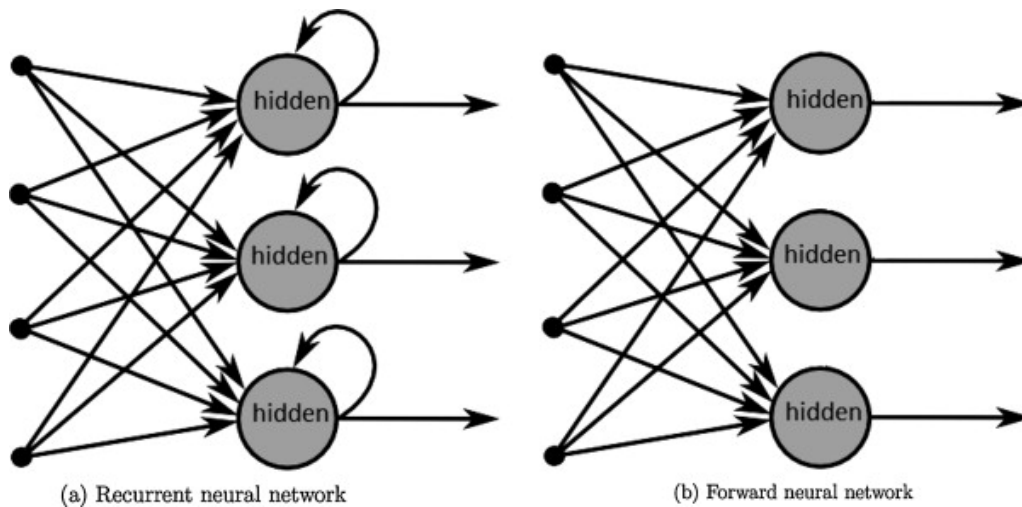


Figura 1.3: Confronto tra rete ricorrente e una rete feed-forward

## Capitolo 2

# Reti neurali in ambito finanziario

Come descritto in precedenza, gli ambiti di sviluppo applicativi delle reti neurali vedono coinvolte diverse componenti del mondo finanziario. Questo grande interesse nei confronti di tali tecnologie rende le applicazioni delle reti neurali talmente numerose da non riuscire a classificarle [2]. Tuttavia possiamo suddividerle, in base ai campi applicativi, in tre grandi categorie:

- previsioni di serie temporali
- classificazione
- approssimazioni di funzioni

### 2.1 Previsioni di serie temporali

In quest'ambito rientrano la maggior parte delle applicazioni delle reti neurali nel settore finanziario, il cui obiettivo è quello di realizzare profitti speculativi tramite *trading* di breve o brevissimo periodo di attività finanziarie quotate, vale a dire, azioni, tassi di cambio e *future*. Il tipo di dato su cui avviene l'elaborazione è solitamente una serie storica come, ad esempio, un indice finanziario. A prescindere dalle possibili applicazioni, da un punto di vista operativo, è necessario dividere la serie storica in due parti: la prima, composta dalle cosiddette osservazioni *in-sample*, funge da base per l'addestramento mentre la seconda, composta dalle osservazioni *out-of-sample*, ha lo scopo di verificare la validità dell'addestramento stesso. Inoltre la rete può essere addestrata per fornire previsioni per più ampi orizzonti temporali, utilizzando le sue stesse previsioni a breve termine come input per le previsioni

a lungo termine. Per questo tipo di applicazioni è molto importante la cura di alcuni aspetti che verranno descritti di seguito.

### **2.1.1 Scelta delle variabili di input**

Tale scelta deve avvenire considerando che la rete non è in grado di fornire alcuna funzione esplicativa e per questo potrebbe utilizzare variabili non significative. Infatti, le relazioni tra variabili cambiano nel tempo e di conseguenza input significativi oggi potrebbero non esserlo più in futuro.

### **2.1.2 Scelta del livello di apprendimento**

Nel processo di apprendimento è necessario tener conto del fatto che un processo di *training* troppo corto potrebbe non consentire alla rete di cogliere le relazioni tra le variabili, mentre un processo di *training* troppo lungo potrebbe rendere la rete incapace di generalizzare rischiando di cadere nel cosiddetto *overtraining*.

### **2.1.3 Scelta dell'orizzonte temporale di riferimento per la previsione**

Tale scelta è un fattore molto importante in quanto orizzonti temporali di previsione molto brevi accrescono il numero di previsioni corrette, portando un profitto medio più elevato, al contrario di quelle di lungo periodo che risultano mediamente meno corrette.

## **2.2 Classificazione e Discriminazione**

Applicazioni tipiche in questo campo riguardano la valutazione del rischio di credito come, ad esempio, la suddivisione in classi di *rating* o le decisioni di affidamento. Nei casi di classificazione la rete ha il compito di assegnare gli input a un certo numero di categorie predefinite cui corrispondono altrettanti output mentre nei modelli destinati alla discriminazione la rete deve anche creare le classi stesse nelle quali suddividere i dati di input.

## **2.3 Approssimazione di funzioni**

In questo caso, le reti vengono applicate in tutte le funzioni avanzate di *pricing* e di *risk management* nelle quali manca una forma funzionale precisa

per la valutazione degli strumenti. Si pensi, ad esempio, alle opzioni di tipo americano, alle opzioni esotiche e ai portafogli di opzioni.

## 2.4 Costruzione della rete neurale

Il processo di costruzione di una rete neurale per applicazioni finanziarie si può articolare in quattro fasi:

- individuazione dell'obiettivo
- costruzione del set di dati su cui effettuare l'apprendimento della rete neurale
- apprendimento
- generalizzazione dell'output

### 2.4.1 Individuazione dell'obiettivo

La prima fase consiste nella definizione dell'obiettivo dell'analisi. Una volta deciso l'ambito dell'indagine, è necessario stabilire i confini della misurazione del fenomeno. Di un certo dato, infatti, possono interessare diversi aspetti. Se ad esempio si individua come obiettivo la previsione del tasso di cambio, è possibile predirne il valore puntuale, il tasso di variazione su un periodo precedente o anche solo la tendenza di periodo. In questo senso la scelta dell'indicatore più utile per la previsione dipende da fattori specifici dell'operatore e da fattori esterni. Nel primo caso, il risultato della rete neurale viene utilizzato per l'implementazione del *trading system*: l'indicazione *buy*, *hold* o *sell* viene mostrata mediante frecce che segnalano il successivo rialzo o ribasso della quotazione. Nel secondo caso, quello che interesserà all'analista sarà, non solo sapere che un mercato è in rialzo o in ribasso, ma principalmente avere un'idea dell'entità della fluttuazione per poterla confrontare con le dinamiche previste sugli altri mercati. Tra i fattori esterni che possono condizionare la scelta vi sono l'eventuale incompletezza o irregolarità di una serie storica, il rumore dovuto ad elementi di inefficienza del mercato che possono impattare negativamente su alcuni prezzi. Va inoltre valutata la scelta della frequenza dei dati su cui ottenere l'output tenendo conto del fatto che poter lavorare su dati a bassa piuttosto che ad alta frequenza consente di ridurre la componente di rumore. Un ultimo aspetto riguarda l'orizzonte temporale della previsione. L'accuratezza delle stime tende in genere a diminuire con l'aumentare del periodo richiesto assegnato alla rete. I risultati più frequenti



mostrano come un'accettabile lunghezza della previsione sia compresa fra i 6 e i 10 anni.

## 2.4.2 Costruzione del set di dati

Nella costruzione della rete neurale deve essere attribuita una grande importanza alla costruzione dell'archivio, cioè del set di informazioni che verranno utilizzate per riconoscere l'eventuale evoluzione della variabile di output. In relazione ai dati, le fasi rilevanti sono due:

- raccolta
- selezione delle variabili di input e output

### Raccolta

La raccolta delle informazioni deve rispondere ad alcuni principi fondamentali. In primo luogo, queste devono essere recuperate dai mercati con regolare frequenza di rilevazione, al fine di garantire la continuità della serie storica. Tale frequenza non deve essere minore della frequenza scelta per la previsione. Questo perché non è possibile fare apprendere ad una rete neurale informazioni la cui rilevazione è avvenuta in periodi fortemente caratterizzati da trasformazioni strutturali di mercato. Per esempio, si pensi a quanto successo sul mercato primario dei titoli di Stato italiani, con meccanismi di asta che sono mutati radicalmente nel tempo. Prevedere oggi il prezzo d'asta utilizzando dati registrati nel periodo in cui esisteva il prezzo base o in cui la Banca d'Italia acquistava tutti i titoli non collocati sopra un certo valore ritenuto corretto, sarebbe altamente improbabile. Inoltre è fondamentale prevedere il rinnovamento periodico dell'archivio dei dati, al fine di aggiornare i parametri della serie. In aggiunta la rete dovrebbe essere messa nelle medesime condizioni conoscitive dell'analista di mercato. Questo significa che i fattori che determinano le scelte sui mercati devono poter essere riconosciuti dalla rete.

## **Selezione delle variabili di input e output**

È Fondamentale una continua calibrazione dell'archivio dei dati in modo da eliminare i dati che risultano non significativi in relazione al contenuto informativo della rete. In termini operativi, la procedura dovrebbe articolarsi nelle seguenti fasi:

- definizione ampia del primo set di dati
- primo apprendimento della rete
- valutazione dell'apporto informativo delle singole variabili
- analisi della matrice di correlazione fra le variabili di input
- eliminazione delle variabili meno significative
- successivo apprendimento della rete con il database ridotto

Seguendo questa serie di operazioni è possibile arrivare all'individuazione dell'archivio ottimale in funzione del problema analizzato.

## **2.5 Apprendimento**

La fase dell'apprendimento è da considerarsi una delle fasi più importanti, poiché è la fase che consentirà alla rete di essere in grado di trovare una soluzione al problema. La scelta dell'architettura della rete è, in questo senso, un fattore estremamente importante. I parametri che devono essere determinati per la costruzione di una buona architettura sono:

- la suddivisione temporale dell'archivio dei dati
- il numero di neuroni da inserire in ciascuno strato
- la funzione di attivazione
- l'aggiornamento dei pesi di connessione dei neuroni
- indicatore di errore

### 2.5.1 Suddivisione temporale dell'archivio dei dati

È necessario suddividere la serie storica in diversi *set* per le diverse fasi del processo di apprendimento. Il primo *set*, detto *training set*, viene usato per l'addestramento vero e proprio. I restanti dati possono a loro volta essere divisi in due *set* detti *test set* e *generalisation set*. La rete impara cercando di riconoscere la dinamica del *training set* e verifica come si adatta sul *test set*. Infine si applica al *generalisation set*, un insieme di dati che non ha mai potuto osservare. Non esiste una regola precisa per la suddivisione della serie storica da analizzare, ma le soluzioni più adottate sono (60% 20% 20%) e (60% 30% 10%) rispettivamente per il *training set*, il *test set* e il *generalisation set*.

### 2.5.2 Numero di strati nascosti e numero di neuroni

Per quanto riguarda il numero di strati nascosti, potrebbe essere sufficiente un solo strato, poiché anche un solo strato è in grado di approssimare funzioni non lineari con elevato grado di accuratezza. Tuttavia, questo approccio richiede un elevato numero di neuroni, andando a limitare il processo di apprendimento. Si è notato che l'utilizzo di reti con due strati nascosti risulta essere il più efficace. Un numero di strati nascosti superiore a due non produce infatti miglioramenti nei risultati ottenuti dalla rete. Per quanto riguarda il numero di neuroni, bisogna trovare un giusto equilibrio poiché un numero troppo basso di neuroni riduce la capacità di apprendimento della rete, mentre un numero molto alto può portare al fenomeno del cosiddetto *overfitting*, cioè il modello tende a specializzarsi troppo sui dati utilizzati nell'addestramento diminuendo la sua capacità di elaborare correttamente su dati diversi da quelli dell'apprendimento. Questo fenomeno è in contrapposizione con quello che si vuole ottenere da una rete neurale cioè la capacità di generalizzazione definibile come l'attitudine ad offrire buone performance su dati diversi dal *training set*. Questo equilibrio si può ottenere utilizzando una delle seguenti quattro formule:

$$h = 2 \cdot n \cdot i \quad (2.1)$$

$$h = 2 \cdot n \quad (2.2)$$

$$h = n \quad (2.3)$$

$$h = \frac{n + m}{2} \sqrt{t} \quad (2.4)$$

dove:

- $h$  è il numero di neuroni nascosti

- $n$  è il numero di neuroni in input
- $m$  è il numero di neuroni in output
- $t$  è il numero di osservazioni ottenute nel *training set*

Secondo gli esperti i risultati ottenuti dimostrano come nessuna di queste formule possa esser generalizzata e quindi utilizzata per risolvere il problema generico, sebbene si preferisca l'utilizzo della 1.3.

### 2.5.3 Connessione tra gli strati differenti

Esistono tre tipi di connessione per le architetture: standard, a salti e ripetute.

#### Connessione standard

Questa tipologia prevede connessioni fra input e output che passano attraverso uno o più strati nascosti, ai quali sono assegnati dei neuroni che facilitano l'apprendimento della rete. Le connessioni sono dirette e non prevedono né salti né ritorni su se stessi (cicli).

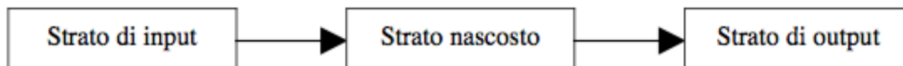


Figura 2.1: Architettura a uno strato nascosto con connessioni standard

#### Connessione a salti

Questa tipologia prevede che la rete assegni dei pesi connettivi anche a neuroni presenti in strati non adiacenti. Nel caso più semplice lo strato di input presenta una serie di connessioni non solo con lo strato nascosto, ma anche con quello di output. Questa ramificazione connettiva si articola ulteriormente nel caso in cui il numero di strati nascosti aumenti.

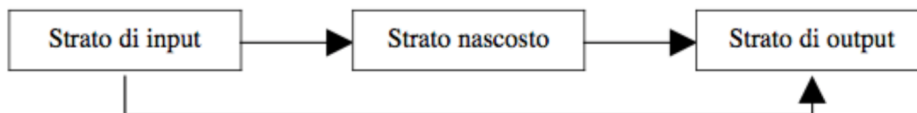


Figura 2.2: Architettura a uno strato nascosto con connessioni a salto

## Connessione ripetute

Questa tipologia prevede la possibilità che i neuroni assegnati agli strati nascosti possano ritornare sulle variabili di input con processi iterativi così da quantificare in modo preciso il peso connettivo.

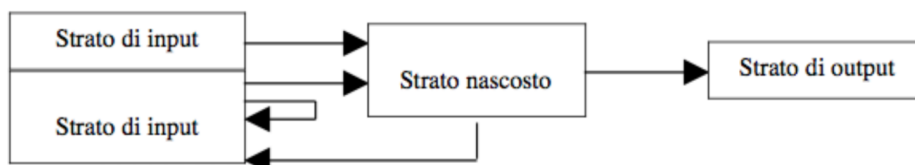


Figura 2.3: Architettura a uno strato nascosto con connessioni ripetute

### 2.5.4 Funzione di attivazione

La funzione di attivazione (*activation function*) è una funzione che trasforma i valori di ingresso nel neurone nei valori di uscita. Esistono otto tipi di diverse di funzioni di attivazione, ed è possibile definire una funzione diversa per ogni strato della rete.

#### La funzione lineare

$$f(x) = x \quad (2.5)$$

Questa funzione risulta efficace quando viene utilizzata per lo strato che contiene l'output, perché, pur essendo più rigida delle altre, evita che il risultato tenda verso il minimo o il massimo, mentre non è consigliata per l'utilizzo negli strati nascosti, soprattutto se questi sono caratterizzati da un elevato numero di neuroni, che risulterebbero così connessi proprio su una base funzionale che si vuole superare con l'utilizzo della rete stessa. Il limite rilevante della funzione lineare è quello di non consentire un *fitting* adeguato per serie storiche caratterizzate da *trend* persistente.

#### La funzione logistica/logistica simmetrica

$$f(x) = \frac{1}{1 + e^{-x}} \quad (2.6)$$

$$f(x) = \left( \frac{2}{1 + e^{-x}} \right)^{-1} \quad (2.7)$$

Entrambe queste funzioni presentano la caratteristica di variare nell'ambito di un intervallo, rispettivamente  $[0, 1]$  e  $[-1, 1]$ . La funzione logistica risulta

essere utile negli strati nascosti delle reti applicate alle serie storiche finanziarie, mentre la simmetrica negli strati di input e nascosti nei problemi di carattere dinamico.

### La funzione tangente iperbolica

$$f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (2.8)$$

Tale funzione diventa affidabile se utilizzata negli strati nascosti nelle reti a tre strati oppure nel caso in cui si sia stata scelta una funzione logistica o lineare per l'output.

### La funzione sinusoidale

$$f(x) = \sin(x) \quad (2.9)$$

Questa funzione viene utilizzata nei problemi di ricerca e si suggerisce di normalizzare input e output all'interno di un range  $[-1, 1]$ .

### La funzione gaussiana

$$f(x) = e^{-x^2} \quad (2.10)$$

La funzione è utile in particolari processi dinamici in cui l'architettura è a due strati nascosti paralleli, con una funzione tangente nel secondo strato.

## 2.5.5 Regole di apprendimento

È necessario definire i criteri di arresto dell'apprendimento a seconda dell'obiettivo che ci si è posti. Se si vuole costruire una rete neurale con finalità previsionale è consigliato valutare l'apprendimento sul *test set* altrimenti, se lo scopo è lo studio di un fenomeno, è più indicato l'apprendimento tramite *training set*.

## 2.5.6 Aggiornamento dei pesi di connessione dei neuroni

Bisogna decidere con quale tasso di cambiamento la rete deve modificare la definizione dei pesi dei neuroni rispetto alla significatività dell'errore commesso. Un altro approccio è quello di utilizzare il cosiddetto *momentum*. In questo caso viene aggiunto al nuovo peso la variazione dell'ultimo peso

raggiunto dalla rete neurale. Tramite l'uso del *momentum* la rete può apprendere anche a un tasso elevato senza rischiare di oscillare, poiché recupera un'elevata quota dell'ultimo peso raggiunto.

### 2.5.7 Indicatore di errore

Per valutare la bontà della rete è necessario utilizzare una funzione che quantifichi lo scostamento del risultato ottenuto da quello aspettato. In questo senso esistono vari indicatori. Tali indicatori vengono utilizzati per variare i parametri di apprendimento. La qualità dei parametri individuati è tanto migliore quanto è più basso l'indice di errore. Durante il processo di apprendimento è importante che il valore di errore non oscilli ma si stabilizzi così da attestare la bontà dei parametri individuati (convergenza).

## 2.6 Generalizzazione

Una volta determinato il valore dei parametri della rete è necessario valutarne le capacità previsionali. Questo perché anche se la rete è stata correttamente costruita potrebbe non essere in grado di offrire un output adeguato nella fase di previsione. La rete va quindi testata sul *generalisation set* calcolando gli indicatori di errore su questa serie storica mai osservata dalla rete. Potrebbe infatti succedere che un modello riesca a descrivere ottimamente il *training* e il *test set* ma poi risulti del tutto inadeguato per quanto riguarda la sua generalizzazione. In questo caso sarà necessario procedere ad un ulteriore addestramento della rete.

Al termine di questo processo la rete neurale sarà verosimilmente in grado di ricostruire la legge che descrive un determinato fenomeno, riconoscendo in esso forme di regolarità e di struttura al fine di elaborare delle previsioni. Tuttavia, il più grande limite delle reti neurali è l'incapacità di fornire spiegazioni riguardo ai risultati raggiunti. Inoltre non bisogna scordarsi che la progettazione di tali reti si ottiene attraverso un procedimento che comporta molte prove e talvolta anche errori.



# Capitolo 3

## Reti neurali per la previsione dei prezzi in finanza

### 3.1 Reti neurali per la previsioni dei prezzi

Da sempre la previsione dei prezzi delle azioni nei mercati finanziari suscita grande interesse, sia dal punto di vista teorico che pratico, tanto che attualmente sono stati inventati diversi modelli per migliorare il potere predittivo degli analisti. In particolare la previsione del prezzo azionario è considerata un compito impegnativo, poiché il mercato azionario è estremamente dinamico e non rispetta un andamento lineare, potendo essere influenzato da molti fattori macroeconomici difficilmente prevedibili, quali eventi politici, scelte nelle politiche aziendali, condizioni economiche generali, l'indice dei prezzi delle materie prime, tassi di interesse e fattori psicologici degli investitori.

Nell'elaborato dal titolo *Neural networks for stock price prediction*[4] del Maggio 2018, vari studiosi confrontano i risultati predittivi ottenuti da cinque modelli di intelligenza artificiale, applicati ai prezzi di chiusura settimanali di tre singoli titoli: Bank of China, Vanke A e Kweichou Moutai.

Nel dettaglio vengono analizzati tre tipi di reti neurali, *backpropagation* (BP), *radial basis function* (RBF) e *general regression* (GRNN), oltre a due modelli di *vector machine: support vector machine regression* (SVMR) e *lest square support machine regression* (LS-SVMR).

#### 3.1.1 Backpropagation

Generalmente una rete BP è formata da un livello di input, uno o più livelli nascosti e un livello di output. In questo lavoro come funzione di attivazione viene utilizzata la sigmoide, mentre per quanto riguarda l'apprendimento, al fine di aumentare la velocità dello stesso, si utilizza un approccio casuale che

fornisca un sottoinsieme dei dati di input nel processo di diminuzione del gradiente.

### 3.1.2 Radial Basis Function

Questo tipo di reti ha tipicamente tre strati: uno strato di input, uno strato nascosto con una funzione di attivazione RBF non lineare ed uno strato di output lineare. Nella funzione di attivazione RBF i valori di ingresso ad ogni neurone sono pesati in relazione alla distanza del neurone che fornisce l'input. In questo senso i neuroni localmente più vicini contribuiscono maggiormente alla funzione di trasferimento.

### 3.1.3 General Regression

La rete neurale a regressione generalizzata (GRNN) è un caso particolare di una rete RBF. GRNN è stato suggerito da D.F. Specht nel 1991. Questo tipo di rete è essenzialmente una RBF nella quale il numero di neuroni nello strato nascosto è lo stesso della dimensione del campione  $n$  dei dati di *training*. Una particolarità di questo tipo di reti è che non necessitano del processo di apprendimento. Nel caso in cui il numero di neuroni nascosti non cambi si parla di rete GRNN statica. Se invece il numero di neuroni nascosti aumenta man mano che si amplia l'archivio di dati si parla di rete GRNN dinamica. In questo lavoro gli autori hanno deciso di utilizzare la versione dinamica della rete poiché ha una maggiore capacità predittiva.

### 3.1.4 Support Vector Machines

Le Support Vector Machines (SVM), che nascono negli anni 90 grazie alla ricerca di Vladimir Vapnik e dei suoi collaboratori, cercano di risolvere il problema dell'apprendimento a partire da un *training set* di dati sperimentali di cui sono noti i parametri caratteristici. L'obiettivo è quello di costruire un sistema che impari da dati già correttamente classificati e che da essi poi riesca a costruire una funzione di classificazione in grado di catalogare dati anche al di fuori di questo insieme. L'algoritmo che permette di addestrarle si può ricondurre ad un problema di programmazione quadratica con vincoli lineari e trova applicazione in svariati ambiti, tra i quali i più comuni sono il riconoscimento di *pattern*, la catalogazione di testi e l'identificazione di volti all'interno di immagini.

### 3.1.5 Risultati

In questo lavoro viene studiato il prezzo di chiusura settimanale di tre singoli titoli: Bank of China (601988), Vanke A (000002) e Kweichou Moutai (600519). La dimensione del campione è pari a 427 dati, rappresentativi del periodo compreso fra il 3 gennaio 2006 e l'11 marzo 2018. L'intero set di dati è stato diviso in un set di allenamento (80%) e un set di test (20%). Per valutare la bontà della rete sono stati utilizzati gli indicatori MAPE (*Mean Absolute Percentage Error*) e MSE (*Mean Squared Error*).

Come possiamo vedere dai risultati dei cinque modelli presenti nella tabella 3.1, i metodi analizzati hanno un buon potere predittivo, poiché anche la rete GRNN che ha un valore di errore più alto, ha comunque un MAPE non superiore al 5%. In questo ambito tale risultato è molto soddisfacente considerando che viene effettuata una previsione sul prezzo delle azioni piuttosto che la volatilità.

Method		BP	RBF	GRNN	SVMR	LS-SVMR
Bank of China	MSE	0.009	0.014	0.020	0.012	0.018
	MAPE	0.019	0.025	0.024	0.023	0.028
Vanke A	MSE	2.976	4.686	6.036	3.422	5.472
	MAPE	0.049	0.065	0.067	0.059	0.072
Kweichou Moutai	MSE	395.1	740.1	1103.6	407.4	405.5
	MAPE	0.026	0.036	0.048	0.029	0.027

Figura 3.1: Risultati dei 5 modelli

Dai risultati appare comunque evidente come la rete BP fornisca delle previsioni decisamente migliori rispetto agli altri tipi di reti neurali. Per quello che riguarda queste ultime è però difficile stabilire quale abbia comportamento migliore poiché l'accuratezza delle previsioni è abbastanza uniforme. Quello che si può comunque affermare è che la rete di tipo GRNN è quella che si comporta nel modo peggiore.

Statistics		Mean	Std.
Bank of China	MSE	0.009	$4.8 \times 10^{-5}$
	MAPE	0.019	0.0001
Vanke A	MSE	2.976	0.0067
	MAPE	0.049	0.0001
Kweichou Moutai	MSE	395.1	1.3728
	MAPE	0.026	$7.7 \times 10^{-5}$

Figura 3.2: Media e deviazione standard per 100 processi di addestramento

Un ulteriore interessante aspetto che emerge da questa ricerca è la stabilità della rete BP. In generale essendo i valori con cui vengono inizializzati i pesi scelti casualmente, ci si trova spesso davanti a situazioni di instabilità. In questo caso, effettuando il processo di addestramento 100 volte, come si può vedere nella tabella 3.2 i valori di deviazione standard si mantengono comunque piuttosto contenuti rispetto ai valori medi.

## 3.2 Rete neurale per la costruzione di un portafoglio di titoli

Nell'ambito degli investimenti finanziari, se per un investitore esiste il rischio del singolo titolo, per un'azienda è importante considerare il rischio di un insieme di titoli (portafoglio) che devono essere scelti in maniera concertata così da massimizzare il rendimento complessivo. Se infatti non esistesse alcuna correlazione, il rischio del portafoglio sarebbe analogo a quello dei singoli titoli. In questo senso il rendimento di un portafoglio si ottiene prendendo in considerazione il contributo che ciascun titolo in esso incluso apporta. In termini economici, il rendimento atteso di un portafoglio è la media ponderata dei rendimenti attesi dei singoli componenti, i cui pesi sono le quote dei vari titoli sul totale del portafoglio.

Previsioni accurate sui prezzi delle azioni aiutano gli investitori/aziende a costruire buone previsioni di portafoglio ottimali. Da qui la necessità di avere previsioni accurate sui prezzi per aiutare il processo decisionale degli investitori/aziende. A questo scopo possono essere utilizzate con profitto le reti neurali. Molti studi affermano infatti che una rete neurale è il metodo di previsione più accurato disponibile in questo momento rispetto ad altri metodi di previsione quali la regressione logistica, autoregressivo vettoriale e media mobile integrata autoregressiva.

Nell'articolo *Forecasting Portfolio Optimization using Artificial Neural Network and Genetic Algorithm* [3] viene descritto come creare una rete neurale in grado di prevedere il prezzo di chiusura futuro di titoli azionari con l'obiettivo di costruire un portafoglio azionario ottimo. In particolare per l'ottimizzazione del portafoglio vengono utilizzati algoritmi genetici. In figura 3.3 viene schematizzato il procedimento seguito per l'ottimizzazione del portafoglio, nel quale sono evidenziate le attività appartenenti al processo suddividendo quelle di *management* finanziario da quelle di *machine learning*. Inoltre, in un processo di questo tipo, è particolarmente importante la preparazione dei dati (normalizzazione) in modo da rendere più efficiente il processo di apprendimento, poiché i pesi degli ingressi dei neuroni sono

spesso compresi in piccoli intervalli (es.  $[-1, 1]$ ).

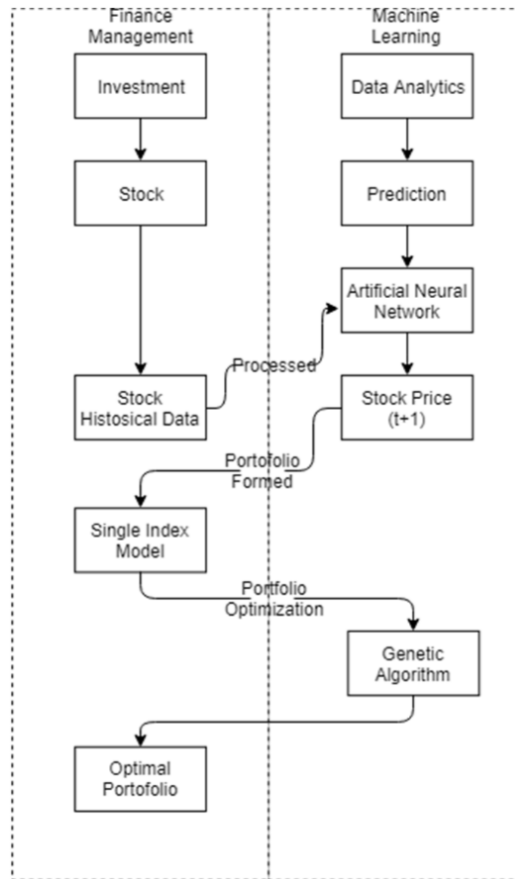


Figura 3.3: Procedimento di ottimizzazione

Per completare lo studio vengono presi in considerazione un campione di 38 titoli azionari dei 45 inclusi nell'indice LQ45 del mercato azionario indonesiano. Questo campione di dati comprende 2464 valori relativi a 10 anni (2008-2018).

Per le previsioni è stata utilizzata una rete *multilayer perceptron* di tre livelli, con algoritmo di apprendimento *backpropagation*. La rete presenta 10 ingressi, 40 neuroni nascosti ed un'uscita (figura 3.4). Gli ingressi rappresentano i valori di apertura, chiusura, massimo e minimo del prezzo e volume di scambio relativi al giorno  $T$  e giorno  $T - 1$ . L'uscita è la previsione del valore di chiusura al giorno  $T + 1$ . La scelta di utilizzare 40 neuroni nel livello nascosti è stata fatta analizzando i risultati ottenuti con 20, 30 o 40 neuroni e notando che all'aumentare del numero di neuroni anche l'accuratezza della rete aumentava.

L'intero set di dati è stato diviso in un set di allenamento (80%) e un set di test (20%).

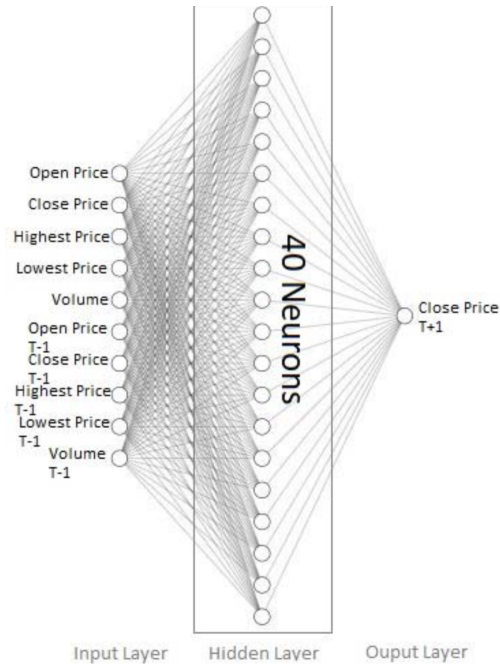


Figura 3.4: Struttura della rete

Per valutare la bontà della rete sono stati utilizzati gli indicatori MAPE (*Mean Absolute Percentage Error*) e MAE (*Mean Absolute Error*). In figura 3.5 vengono riportati i valori di tali indicatori.

Average 38 Stock Error Ratio	
MAE (Rupiah)	MAPE
485.96	5.60%

Figura 3.5: Risultati (MAE e MAPE)

È possibile anche verificare, come mostrato in figura 3.6, come la maggior parte dei risultati siano accurati. Essi infatti sono quasi tutti disposti sulla *fit line*.

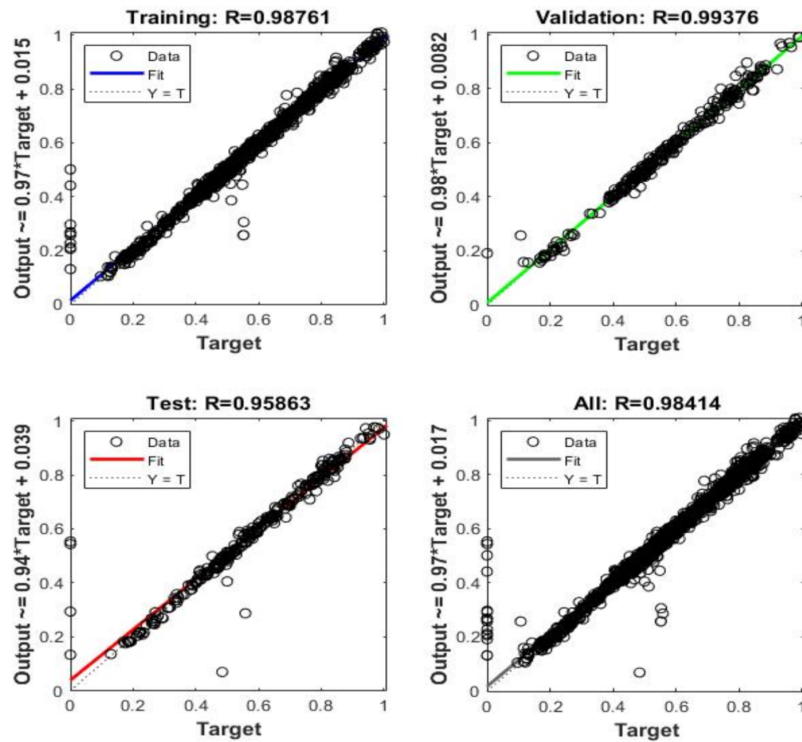


Figura 3.6: Risultati (Regressione)

Nella figura 3.7 è infine possibile vedere come i risultati di previsione della rete si avvicinino molto ad i valori reali.

SAMPLE OF EMPIRICAL RESULTS USING 10-40-1 NEURAL NETWORK PREDICTIVE MODEL

Sample Period	SMGR Stock	
	Actual Price	Predicted Price
21/10/2016	9975	9988.78
24/10/2016	10150	9922.67
25/10/2016	10075	10099.62
26/10/2016	10075	10053.7
27/10/2016	9975	10061.94
28/10/2016	9850	9902.99
31/10/2016	9500	9813.49
1/11/2016	9450	9431.39
2/11/2016	9275	9632.4
3/11/2016	9275	9337.17

Figura 3.7: Confronto fra valori stimati e valori reali

# Conclusioni

Dall'analisi di questi due articoli emerge come utilizzando in modo appropriato le reti neurali sia possibile ottenere degli ottimi risultati di previsione in ambito economico-finanziario.

In particolare fra le varie tipologie di reti neurali appaiono particolarmente efficaci, al fine di previsioni basate su serie storiche, reti neurali di tipo *feed-forward* con algoritmo di apprendimento *backpropagation*, grazie anche alla loro relativa semplicità. Questo dimostra l'efficacia di questo approccio nei confronti di una tipologia di dati che per loro natura è priva di struttura evidente. Naturalmente per poter ricavare dei risultati affidabili non bisogna trascurare il processo di addestramento che può essere costoso in termini di tempo e risorse.

Rimane comunque vero che sebbene una rete neurale sia in grado di ricostruire la legge che descrive un certo fenomeno, riconoscendo in esso forme di regolarità e di struttura che le consentono di produrre delle previsioni, il suo più grande limite è l'incapacità di fornire spiegazioni riguardo ai meccanismi di relazione fra dati in ingresso e previsione. Per questo motivo è probabile che tale limite un giorno porterà allo sviluppo di nuove tecniche algoritmiche in grado di fornirci non solo risultati accurati ma anche la comprensione delle leggi sulle quali tali risultati sono basati.



# Bibliografia

- [1] K.-L Du e M.N.s Swamy. *Neural Networks and Statistical Learning*. Ott. 2013. ISBN: 978-1-4471-5570-6. DOI: 10.1007/978-1-4471-5571-3.
- [2] Crescenzo Gallo. «Reti Neurali Artificiali: Teoria ed Applicazioni Finanziarie». In: *Dipartimento di Scienze Economiche, Matematiche e Statistiche, Università di Foggia, Quaderni DSEMS* (gen. 2007).
- [3] Mohammad Solin, Andry Alamsyah, Brady Rikumahu e Muhammad Saputra. «Forecasting Portfolio Optimization using Artificial Neural Network and Genetic Algorithm». In: lug. 2019, pp. 1–7. DOI: 10.1109/ICoICT.2019.8835344.
- [4] Yue-Gang Song, Yu-Long Zhou e Ren-Jie Han. «Neural networks for stock price prediction». In: (mag. 2018).