Politecnico di Torino

Politecnico di Torino

Corso di Laurea Magistrale in Ingegneria Gestionale
A.a. 2022/2023
Sessione di Laurea Dicembre 2022

Intelligenza artificiale e deep learning applicati al mercato finanziario

Relatori:
Dottoressa Laura Rondi
Dottor Franco Varetto

Candidato:
Marco Barbero

Sommario

1. Analisi tecnica dei mercati finanziari 7

1.1. Analisi tecnica vs analisi fondamentale 7

1.1.1. Critiche all'analisi tecnica 8

1.1.2. La Teoria di DOW 9

1.2. Analisi dei Grafici 10

1.2.1. Tipologie di grafici 10

1.2.2. Il concetto di trend 12

1.2.3. Supporti e Resistenze 13

1.2.3.1. Inversione 15

1.2.4. Trendlines 16

1.2.5. Percentuale di ritracciamento 20

1.2.6. Reversal & Price gap 21

1.2.7. Formazioni principali 22

1.2.7.1. Formazioni d'inversione 23

1.2.7.1.1. Testa e spalle & testa e spalle rovesciato 23

1.2.7.1.2. Tripli massimi e minimi 25

1.2.7.1.3. Doppio top e doppio bottom 25

1.2.7.1.4. Soucer e spikes 26

1.2.7.1.5. Formazioni di inversione con Candlestick 26

1.2.7.2. Formazioni di continuazione 27

1.2.7.2.1. I triangoli 27

1.2.7.2.2. Wedge 29

1.2.7.2.3. Rettangolo 30

1.2.7.2.4. Testa e spalle come figura di continuazione 30

1.2.7.2.5. Formazioni di continuazione nelle Candlesticks 31

1.2.8. Volume 32

1.2.8.1. Indicatori di volume 33

1.2.9. Ordine nell'analisi: prima i grafici di lungo periodo. 33

1.2.10. Medie mobili 34

1.2.10.1. Utilizzo di più medie mobili 36

1.2.10.2. Envelopes e bande di Bollinger 37

1.2.11. Oscillatori 37

1.2.11.1. Tipi di Oscillatori 38
1.2.11.2. MACD: Moving average convergence/divergence 42
1.2.12. Teoria delle onde di Elliott 43
1.2.12.1. Il pattern 43
1.2.12.1.1. Price channeling 44
1.2.12.2. Ratio: la percentuale di Fibonacci e il ritracciamento 45
1.2.13. I Cicli temporali 46
1.2.13.1. Cicli temporali nell'analisi tecnica 49
1.3. Tattiche operative 50
1.3.1. Money management 50
1.3.2. Timing 51
1.3.3. Tipologie di ordini 52
1.3.4. Legami tra mercati 52
1.3.5. Analisi del respiro del mercato 54
2. Intelligenza artificiale, Machine Learning e Deep learning 56
2.1. Un po' di storia 60
2.2. Costruzione di sistemi di machine learning 63
2.2.1. Terminologie e componenti necessari 63
2.2.2. Tensori 64
2.2.2.1. Operazioni sui tensori 65
2.2.2.2. Tensori nel deep learning 67
2.2.3. Il deep learning 70
2.2.3.1. Modelli di reti di layer 70
2.2.3.2. Funzioni obiettivo e ottimizzatori 71
2.2.4. Le principali fasi della costruzione del sistema 72
2.2.4.1. Definizione del problema 72
2.2.4.2. Pre-elaborazione 72
2.2.4.3. Creazione della rete 74
2.2.4.4. Addestramento 75
2.2.4.5. Valutazione 76
2.2.4.6. Ottimizzazione iperparametri e regolarizzazione modello 78
2.2.4.7. Predizione 78
2.3. Problemi del machine learning 78

2.3.1. Funzioni di attivazione 81
2.4. Tipologie di apprendimento 81
2.4.1. Apprendimento con supervisione 81
2.4.2. Apprendimento con auto-supervisione 82
2.4.3. Reinforcement learning 82
2.4.4. Apprendimento senza supervisione 83
2.5. Tipologie di reti neurali 83
2.5.1. Convnet 84
2.5.1.1. Funzionamento layer Convnet 84
2.5.1.2. Come procedere con pochi dati a disposizione 86
2.5.1.3. Visualizzare le rappresentazioni apprese da convnet 88
2.5.2. Reti neurali ricorrenti e convnet 1D 89
2.5.2.1. Reti neurali ricorrenti 92
2.5.2.2. Migliorie alle reti neurali ricorrenti 94
2.5.2.3. Combinazione di CNN e RNN 95
2.6. Algoritmi per il machine learning 95
2.6.1. Perceptron 96
2.6.2. Adaline 98
2.6.3. La regressione lineare 99
2.6.4. La regressione logistica 100
2.6.5. SVM, Support Vector Machine 102
2.6.6. Alberi decisionali 103
2.6.7. K-Nearest Neighbor 104
3. Sistemi di trading informatici 106
3.1. Python 106
3.2. Trading algoritmico 107
3.3. L'utilizzo dei dati 108
3.3.1. Archiviazione efficiente 109
3.4. Backtesting vettorizzato 109
3.5. Applicazione Strategie in Python 110
3.5.1. Strategie basate su SMA 110
3.5.2. Strategie basate su Momentum 114
3.5.2.1. A serie storiche 114

3.5.3. Strategie basate su Mean Reversion 119
3.5.4. Possibili problematiche nel backtesting 121
3.5.5. IA, Machine learning e Reti neurali 122
3.5.5.1. Regressione lineare 122
3.5.5.2. Machine learning: regressione logistica 126
3.5.5.3. Deep learning 129
4. Conclusioni 131
Bibliografia 134
Ringraziamenti 135

Introduzione

Ogni fenomeno o evento che avviene attorno a noi è descrivibile attraverso un'enorme quantità di caratteristiche. Oggigiorno la raccolta e la gestione di questi dati risulta più facile grazie ad hardware e reti sempre più prestanti, perciò diventa fondamentale saperli gestire e soprattutto elaborare.

L'analisi dei dati può fornirci preziose indicazioni permettendo di individuare tempestivamente, a volte prevedere, possibili sviluppi come:

- malattie in ambito medico
- vendite per un'azienda
- previsioni meteo

Si potrebbe continuare all'infinito l'elenco ma l'aspetto più importante è che risulta quasi impossibile effettuare manualmente tutte le osservazioni necessarie in quanto l'analisi anche di un solo evento richiederebbe molto tempo e per questo motivo vengono usati calcolatori in grado di automatizzare e velocizzare il lavoro.

Nella programmazione classica i computer fanno ed eseguono i compiti assegnati e individuati dagli utenti, se però a quest'ultimi sfuggisse una certa tipologia di analisi o uno schema presente nei dati utile allo scopo, la mancanza si propaga a tutta l'attività.

Per ovviare a queste problematiche è stato sviluppato il concetto di intelligenza artificiale, per fare in modo che sia il calcolatore ad apprendere in autonomia schemi dai dati che possono sfuggire o essere sfuggiti all'uomo e applicare queste scoperte nelle analisi.

I dati di partenza possono essere di diverso tipo ma indipendentemente da questo vengono tutti "tradotti" in numeri in modo che un computer possa elaborarli.

Un ambito descritto quasi totalmente da numeri dove questo nuovo approccio ha visto una rapida espansione negli ultimi anni è quello finanziario in cui riuscire ad anticipare anche di poco futuri movimenti di mercato può fare un'enorme differenza tra profitti e perdite.

Lo scopo di questa tesi è proprio sviluppare un sistema di trading basato sull'intelligenza artificiale, e in particolar modo quel ramo chiamato deep learning dove è il calcolatore stesso a cercare di capire e individuare possibili schemi o relazioni tra i dati.

Il primo capitolo della tesi introduce le basi di quel che si definisce analisi tecnica, cioè quei metodi e strategie sviluppate nel corso del tempo per lo studio degli strumenti finanziari i cui output possono essere utilizzati come input nell'intelligenza artificiale.

Il secondo capitolo descrive e approfondisce gli aspetti che riguardano l'intelligenza artificiale, il machine learning e il deep learning spiegando come avviene la creazione di una rete neurale.

La maggior parte del lavoro svolto per il terzo capitolo non è visibile nella tesi in quanto riguarda lo sviluppo di diversi algoritmi in linguaggio Python per la predizione dei comportamenti di strumenti finanziari. Ovviamente il principale è quello inerente all'intelligenza artificiale e al deep learning ma per valutare in modo più ampio i risultati, non solo rispetto l'andamento normale di borsa, sono stati sviluppati e testati anche altre tipologie di algoritmi basati su alcune semplici tecniche dell'analisi tecnica. Il terzo capitolo presenta quindi i diversi risultati che si sarebbero ottenuti con queste applicazioni su cui vengono tratte le conclusioni.

1. Analisi tecnica dei mercati finanziari

"*L'analisi tecnica è lo studio del movimento di mercato tramite l'uso sistematico di grafici allo scopo di prevedere le tendenze future dei prezzi*."[1]

Nonostante si possa applicare sia in ambito futures che in quello azionario, in questo testo viene trattata solamente in ottica di quest'ultimo. Le tre fonti più importanti ai fini dell'analisi tecnica sono:

- Prezzo
- Volume
- Open interest (solo per i futures)

Si parla di movimenti dei prezzi o **Price Action**, quando di considera solamente il prezzo, mentre si usano i termini movimenti dei prezzi di mercato o **Market Action**, quando vengono inclusi tutti i fattori.

1.1. Analisi tecnica vs analisi fondamentale

All'analisi tecnica si contrappone l'analisi fondamentale che cerca di determinare il giusto prezzo di mercato esaminando tutti i fattori rilevanti delle forze economiche per poi confrontarlo con il valore attuale: così si individuano i beni sopravvalutati (Prezzo attuale > Prezzo analisi) che andrebbero venduti, o sottovalutati (Prezzo attuale < Prezzo analisi) che andrebbero acquistati.

Analisi Fondamentale→ studio delle cause

Analisi Tecnica→ studio degli effetti

La differenza tra le due tipologie si osserva soprattutto quanto nel mercato avvengono cambiamenti importanti di prezzo dovuti a motivazioni fondamentali non ancora emerse storicamente. In questi casi gli analisti fondamentali non hanno modo di intercettare l'anomalia e non possono ipotizzarne le conseguenze, mentre un'analista tecnico tramite l'interpretazione di grafici e indicatori **avrebbe** potuto riscontrare l'irregolarità per agire di conseguenza.

Questo perché l'analisi tecnica parte da 3 presupposti:

I. **Il mercato sconta tutto:** i prezzi di borsa incorporano già tutti i fattori fondamentali che li hanno determinati (es. politico, psicologico, etc….) e l'andamento dei prezzi rispecchia i cambiamenti quantitativi della domanda e dell'offerta.
   Domanda > Offerta → Prezzo sale
   Offerta > Domanda → Prezzo Scende
   Il prezzo risulta essere l'indicatore per i fondamentali e l'analista tecnico quindi non si preoccupa di capire o approfondire i motivi fondamentali esterni che han portato alla variazione dei prezzi, perché tramite lo studio dei grafici e l'uso di indicatori, questi aspetti risultano indirettamente osservati. Può capire anche la direzione che il mercato intende prendere senza effettuare un'analisi delle motivazioni esterne al prezzo.

[1] J. J. Muprhy; Analisi tecnica dei mercati finanziari; 2° edizione; pag. 1

II. **I prezzi si muovono dentro un trend** creato dal mercato stesso. Come vedremo più avanti l'obiettivo principale dell'analisi è individuare il prima possibile un trend per poterlo sfruttare investendo nella stessa direzione, in quanto è più probabile che questo prosegua nella stessa direzione fino al presentarsi di determinati segnali.
III. **La storia si ripete:** i movimenti di mercato risultano spesso ricorrenti e tendono a rispecchiare la psicologia umana del mercato in un determinato momento storico.

L'analisi fondamentale è sempre necessaria nei primi approcci decisionali ma l'analisi tecnica risulta essere:

- **più flessibile**: permette di seguire diversi mercati, approfondendo quello che risulta più promettente solo nel momento opportuno.
- **più adattabile**: può occuparsi di diverse dimensioni temporali (brevi, medie e lunghe) ed è utilizzabile per azioni, features, opzioni e beni.

Come verrà discusso più avanti tra i diversi mercati (azionario, obbligazionario, commodities e valute) esistono dei legami e l'analisi tecnica applicata su ognuno di essi permette di individuare meglio l'azione o i gruppi di azioni su cui conviene operare in un determinato momento.

1.1.1. Critiche all'analisi tecnica

Negli anni sono state mosse una serie di critiche all'approccio di tipo tecnico che possiamo riassumere in 3 punti:

- **La teoria dell'autoalimentazione**: ipotizza che vi sia una sorta di fattore di autoalimentazione e cioè tramite l'analisi dei grafici si innescherebbero ondate di acquisti o vendite in base alle figure individuate. L'interpretazione delle figure è soggettiva e vengono individuate dall'osservatore, f*inora non è stato possibile quantificarle matematicamente con certezza* e resta difficile pensare che tutti gli operatori vedano la stessa forma nello stesso momento. Sta quindi nell'abilità dell'analista individuare tempestivamente una figura e agire di conseguenza attendendo il momento giusto in base al proprio intervallo di tempo di riferimento. Anche l'espandersi dell'uso di sistemi informatici non dovrebbe preoccupare in questo senso in quanto tendenzialmente programmati per seguire i trend primari, e anche iniziassero tutti a eseguire gli stessi movimenti i software verrebbero adattati di conseguenza.
- **Studiare il passato per predire il futuro:** la seconda critica riguarda lo studio dei prezzi passati per capire i possibili movimenti futuri. Criticare questo approccio però vuol dire mettere in dubbio tutti i sistemi di previsione utilizzati, analisi fondamentale compresa, in quanto basati sullo studio delle serie storiche del passato. L'analisi dei grafici è fondamentalmente un'analisi delle serie storiche.
- **Teoria del random walk**: basata sull'ipotesi che in un mercato efficiente i prezzi fluttuano vicino al loro valore intrinseco, quindi il variare dei prezzi è casuale e imprevedibile e la serie storica dei prezzi non è un indicatore utilizzabile per il futuro. Secondo questa teoria il miglior approccio è il buy and hold invece di tentare di anticipare il mercato. Anche se è innegabile la presenza di casualità è impensabile pensare che tutti i movimenti di mercato siano casuali e inoltre osservando un qualsiasi tipo di grafico è evidente la presenza di trend non giustificati con il random walk. Resta difficile pensare che quello che è successo ieri, o nelle settimane precedenti, non abbia conseguenze sul presente.

1.1.2. La Teoria di DOW

L'analisi tecnica vede la sua origine dalle teorie di Charles Dow che nel 1882 assieme a Edward Jones fondarono la Dow Jones & Company.

Dow riteneva che tramite lo studio degli indici era possibile valutare le condizioni economiche di una nazione e per questo definì una serie di indici:

- 1884 pubblicò il primo indice contenente 9 titoli di compagnie di trasporti e 2 industriali
- 1897 divise l'indice in due parti, una con 12 titoli di società industriali e la seconda con 20 titoli di società ferroviarie
- 1928 il primo indice racchiudeva 30 titoli di società industriali, *com'è ancora oggi.*

Dow pubblicò le sue idee in diversi articoli ma una forma sistemica alle sue idee fu data solo dopo la sua morte ed è riassumibile in sei concetti generali che risultano adatti ancora oggi nell'analisi tecnica degli indici:

1. **Gli indici scontano tutto**: ogni possibile fattore prevedibile del mercato è riflesso negli indici di borsa, e immediatamente assimilato nei prezzi. Il mercato sconta tutto
2. **Il mercato ha 3 trends:** distinti in base all'estensione temporale
   - Primario: trend di durata maggiore, anni.
   - Secondario: di durata media (settimane o mesi) rappresenta una correzione del trend primario
   - Minore: breve fluttuazione del trend secondario, solitamente poche settimane
3. **Il trend primario ha 3 fasi:**
   - Accumulazione: quando agiscono sul mercato gli investitori più informati che ritengono che le notizie negative sono già state scontate dal mercato
   - Partecipazione: in questa fase prendono posizione i cosiddetti trend followers, con aumento dei prezzi e delle notizie positive
   - Distributiva: ulteriore aumento delle news positive e aumento dei volumi speculativi. In quest'ultima fase gli investitori entrati nello step di accumulazione iniziano la vendita
4. **Gli indici si devono confermare a vicenda:** secondo Dow per avere conferma di un segnale rialzista o ribassista entrambi gli indici che aveva ipotizzato, industriale e ferroviario, dovevano dare le stesse indicazioni
5. **Il volume deve confermare il trend:** il volume deve espandersi nella direzione del trend primario. Ad esempio se rialzista il volume aumenta con l'aumentare dei prezzi e diminuisce con l'abbassarsi di questi ultimi
6. **Un trend si considera in atto fino al manifestarsi di un segnale di inversione:** anche se non è facile, l'individuazione di questi segnali avviene tramite l'uso di alcuni indicatori ma risulta comunque difficile distinguere un'inversione di trend da una normale correzione

La teoria di Dow considera solo i prezzi di chiusura, per questo motivo viene considerata poco reattiva e si stima che applicandola si possa perdere fino al 25% del movimento iniziale, in quanto i suoi segnali risultano evidenti solo nella seconda fase di un trend. Questo però non era un problema in quanto non era nata con l'obbiettivo di anticipare i movimenti, bensì con lo scopo di catturarne il più possibile e di usare gli indici per determinare lo stato di salute dell'economia.

1.2. Analisi dei Grafici

1.2.1. Tipologie di grafici

Un grafico utilizzato nell'analisi tecnica riporta il tempo sull'asse delle ascisse X e il prezzo sull'asse delle ordinate Y.

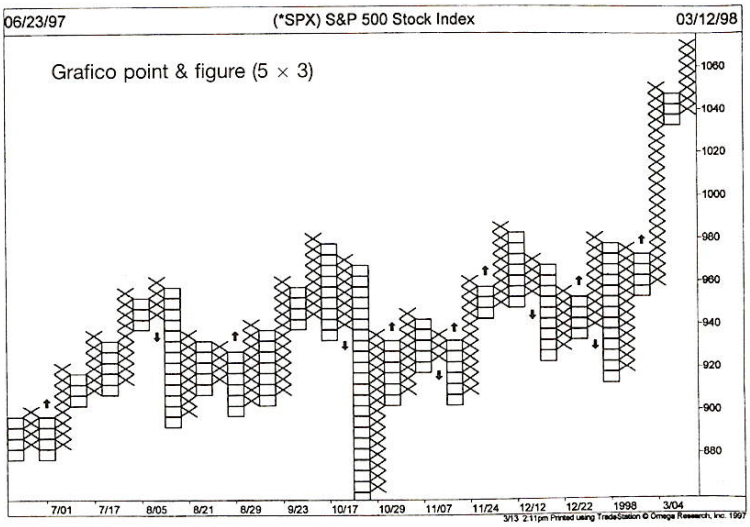
Il fattore tempo può essere rappresentato con intervalli molto diversi: anni, mesi, settimane, giorni o addirittura minuti e l'utilizzo di diverse scale temporali permette all'analista di avere una visione più completa dei movimenti di mercato. Da questo punto in avanti, se non specificato diversamente, negli esempi si considerano intervalli di tempo giornalieri.

Sui grafici si registra giorno per giorno il valore del prezzo, e ai fini dell'analisi può essere utilizzato il prezzo:

- massimo
- minimo
- di apertura
- di chiusura

Esistono diversi tipi di grafici:

- **Point&figure**: prima forma utilizzata nell'analisi tecnica, costituito da barre verticali rappresentate con X, in caso di prezzo al rialzo, e con O, in caso di prezzo al ribasso.



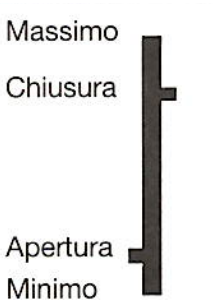
[2]

Questa tipologia non prende in considerazione il fattore tempo e registra solo le variazioni di prezzo, infatti se in un determinato intervallo non si verificano cambiamenti il grafico non cambia. Il volume non viene riportato a parte ma è incorporato nella quantità di variazioni che sono avvenute definendo una dimensione per ogni box. Ogni volta che una colonna X si muove sopra la colonna precedente si ha un segnale di acquisto e ogni volta che una colonna 0 scende al di sotto della colonna 0 precedente si ha un segnale di vendita. È possibile variare le dimensioni dei box per modificare la sensibilità del grafico: maggiori dimensioni implicano un numero minore di segnali e viceversa box più piccoli risultano utili per trading di breve periodo grazie a una maggiore sensibilità

- **Linear**: in questo tipo di grafico vengono considerati solo i prezzi di chiusura e si tratta semplicemente di una linea che unisce tutte le chiusure giornaliere

[2] J. J. Muprhy; Analisi tecnica dei mercati finanziari; 2° edizione; pag. 203

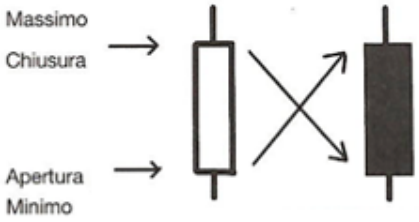
- **Bar chart:** il daily è il più utilizzato e prende in considerazione sia il fattore tempo che prezzo. Ogni movimento è rappresentato da una barra verticale che racchiude l'informazione relativa al prezzo massimo, minimo, di apertura e di chiusura nell'unità di tempo considerata.



[3]

- **Candlestick**: versione giapponese dei bar char che incorpora le stesse informazioni ma rappresentate in modo da essere di più rapida e facile comprensione. Ogni candela è così composta:

[4]



  - Real body: rettangolo che misura la distanza tra il prezzo di apertura e quello di chiusura, se è lungo si parla di *long days* e in caso contrario di *short days*. Il colore del body cambia a seconda che il prezzo di apertura sia maggiore o minore del prezzo di chiusura (bianco-nero oppure verde-rosso)

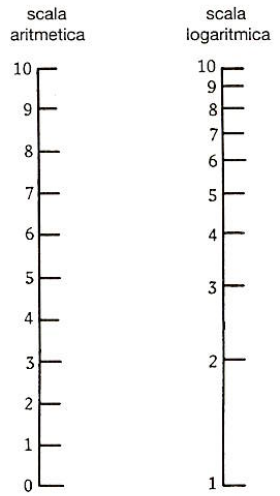
  - Shadow: linea sottile, situata sopra e/o sotto il body, che mostra l'intervallo dei prezzi nell'unità di tempo, dal massimo al minimo. Se estese sono sintomo di un mercato molto indeciso e in caso tutta o quasi la candlestick sia composta da ombre, prezzo di chiusura vicino a quello di apertura, si parla di candela *doji* di cui ne esistono diversi tipi*:*
    - *Long legged*: ombra superiore molto estesa che implica grossa indecisione di mercato
    - *Gravestone*: ha solamente un'ombra superiore molto estesa, più lunga è più risulta essere un segnale negativo
    - *Dragonfly*: unica ombra inferiore estesa ed è vista come un possibile segnale positivo rialzista

  Questa tipologia può essere usata per diversi archi temporali, solitamente il massimo è di una settimana (5 giorni), e l'analisi della singola candela può dare importanti indicazioni operative al trader.

Nelle analisi che seguiranno si considereranno come tipologie di grafici i Bar chart e Candlestick.

Per rappresentare i prezzi sui grafici possono essere utilizzati scale aritmetiche oppure logaritmiche.

[3] J. J. Muprhy; Analisi tecnica dei mercati finanziari; 2° edizione; pag. 229
[4] J. J. Muprhy; Analisi tecnica dei mercati finanziari; 2° edizione; pag. 203

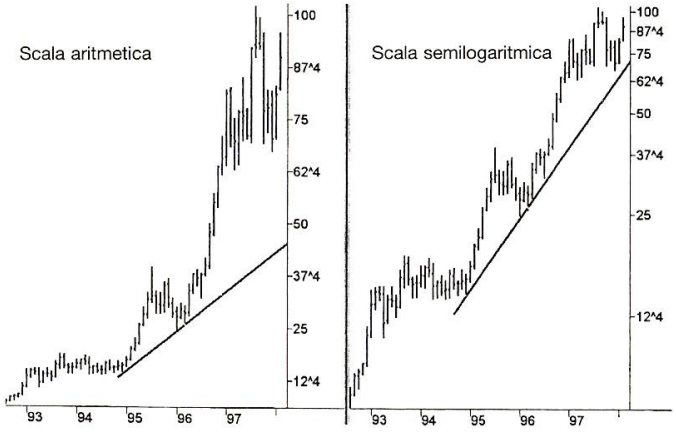


[5]

La scala aritmetica (più utilizzata per i futures) è caratterizzata da spazi costanti ed equidistanti tra i valori rappresentati, la distanza tra 5 e 10 uguale a quella tra 20 e 25.

Nella scala logaritmica (più utilizzata per azioni) l'ampiezza è uguale per le stesse percentuali di variazione, ad esempio tra 1-2 e 5-10 dove in entrambi i casi il delta è del 100%.

Ovviamente questa differenza si ripercuote anche nelle trendlines, spiegate nel prossimo paragrafo, che vengono tracciate sui grafici. Nell'esempio a fianco sullo stesso grafico utilizzando la scala logaritmica la trendline tracciata sui medesimi punti risulta più inclinata evidenziando un andamento migliore



1.2.2. Il concetto di trend

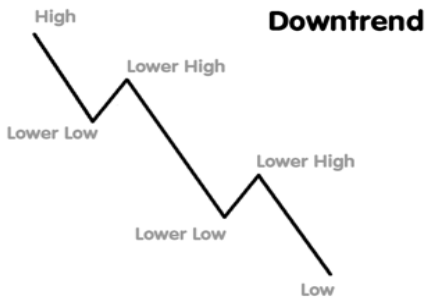
I grafici dei mercati generalmente non seguono un andamento lineare ma sono rappresentati da onde a zig-zag in cui si susseguono una serie di massimi e minimi. *"La direzione di questi massimi e minimi costituisce il trend di mercato"*.[6]

Il concetto di trend è il concetto fondamentale dell'analisi tecnica, il suo monitoraggio è lo scopo principale degli analisti per cercare di capire il miglior momento per posizionarsi nel mercato.

Il trend può avere 3 direzioni:

- **Rialzista(uptrend):** serie di massimi e minimi crescenti dove ogni massimo e minimo successivi risultano più alti dei precedenti. Un trader tende ad acquistare con questo tipo di formazione.

[5] J. J. Muprhy; Analisi tecnica dei mercati finanziari; 2° edizione; pag. 31
[6] J. J. Muprhy; Analisi tecnica dei mercati finanziari; 2° edizione; pag. 37



- **Ribassista(downtrend)**: serie di massimi e minimi decrescenti. Se riconosce l'inizio di un trend ribassista il trader vende la sua posizione.

- **Laterale(trendless):** serie di massimi e minimi orizzontali, riflette un periodo di equilibrio nei livelli del prezzo dove la domanda e l'offerta si bilanciano. In questi casi l'operatore solitamente non prendere posizioni.

Come visto in precedenza il trend può avere numerose classificazioni in base all'arco temporale utilizzato, ma la maggior parte degli analisti tendono a considerarne principalmente 3 secondo quanto indicato nella teoria di Dow:

- Primario: periodo maggiore di un anno
- Secondario(intermedio): dalle 3 settimane a diversi mesi, utilizzato per strategie trend-following
- Minore (di breve periodo): periodo inferiore alle 2/3 settimane, utilizzato per identificare il corretto timing di entrata nel mercato

[7]

Ogni trend fa parte del superiore e ne rappresenta una correzione se di segno opposto.

Ciascun trader può optare per una propria classificazione in base all'arco temporale di cui si occupa (breve o lungo periodo): ad esempio per un trader che opera daily archi temporali di giorni costituiscono trend primari mentre risultano insignificanti per operatori di lungo periodo.

1.2.3. Supporti e Resistenze

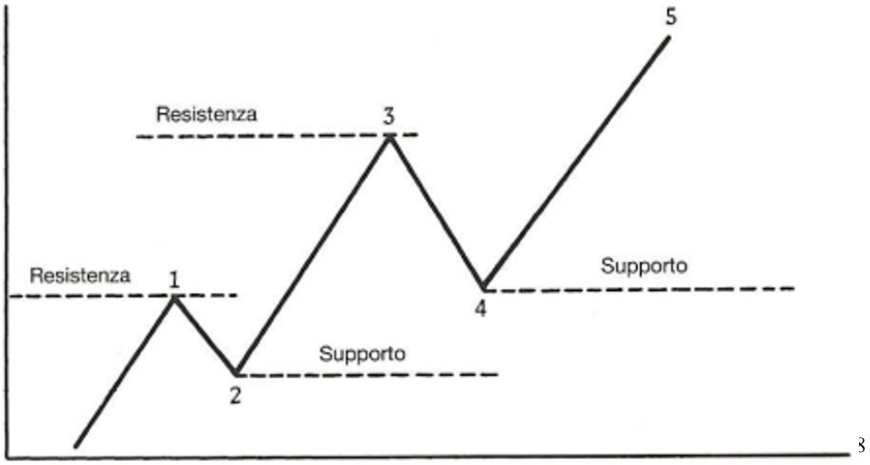
I massimi e minimi che ci permettono di individuare un trend possono essere definiti rispettivamente resistenze e supporti.

I minimi vengono anche chiamati supporti perché individuano un'area del grafico in cui l'interesse dei compratori è superiore a quello dei venditori. Di conseguenza essendoci più domanda rispetto all'offerta i prezzi smettono di scendere e iniziano a salire formando un livello di supporto.

[7] J. J. Muprhy; Analisi tecnica dei mercati finanziari; 2° edizione; pag. 41

I valori di resistenza (massimi) funzionano invece all'opposto, in quanto rappresentano i punti in cui la pressione di vendita è maggiore di quella di acquisto.

Esempio di supporti e resistenze in Uptrend:



Supporti e resistenze sono usati per individuare l'andamento dei diversi trend infatti:

- Durante un uptrend se:
  - avviene una correzione tale che il minimo precedente
    - viene raggiunto→ potrebbe indicare l'inizio di un trend laterale
    - viene violato→ potrebbe indicare un'inversione di trend
  - una precedente resistenza viene testata
    - ma non superata→potrebbe indicare un cambiamento di trend

- Durante un downtrend se:
  - avviene una correzione tale che il massimo precedente
    - viene raggiunto→ potrebbe indicare l'inizio di un trend laterale
    - viene violato→ potrebbe indicare un'inversione di trend
  - un precedente supporto viene testato
    - ma non rialzato→potrebbe indicare un cambiamento di trend

La presenza di supporti e resistenze permette di individuare delle figure specifiche sui grafici, approfondite più avanti, che aiutano l'analista ad anticipare inversioni o lateralizzazioni.

La permanenza dei prezzi in queste specifiche aree può diventare molto significativa ai fini dell'analisi ed è caratterizzata da 3 parametri:

- tempo trascorso nell'area: tanto più a lungo si estende un'area, di supporto o resistenza, più importante questa diventa
- volume di contrattazione: più il volume risulta essere forte, molte azioni sono state scambiate, più il livello è importante
- momento delle contrattazioni: più le attività sono recenti, più è da considerarsi forte l'area

[8] J. J. Muprhy; Analisi tecnica dei mercati finanziari; 2° edizione; pag. 42

1.2.3.1. Inversione

Supporti e resistenze sono dei concetti "dinamici", cioè quando un livello viene violato subisce un'inversione di ruolo: quello che prima era un supporto diventa una resistenza e viceversa.

Questo processo e dovuto alla psicologia che sta alla base di questi livelli. Per spiegarlo è necessario prima suddividere gli utenti del mercato come segue, ipotizzando un mercato al rialzo iniziato da un'area di supporto dove i prezzi hanno oscillato:

1. **longs**: coloro che hanno già comprato ma che pensano di non aver acquistato a sufficienza, quindi attendono che il prezzo torni verso il supporto per aumentare le posizioni
2. **short**: operatori cha hanno aperto operazioni in vendita, con il mercato al rialzo si trovano nella direzione sbagliata tanto più il prezzo si allontana dal supporto e non vedono l'ora che il prezzo si riavvicini al supporto per acquistare posizioni dove hanno impostato le vendite
3. **liquidi**:
    a. utenti che avevano posizioni ma le ha liquidate troppo presto al prezzo di supporto: attendono un altro retest per comprare nuovamente
    b. utenti non ancora entrati(uncommitted): una volta intuito il trend rialzista aspettano la prossima opportunità per entrare nel mercato

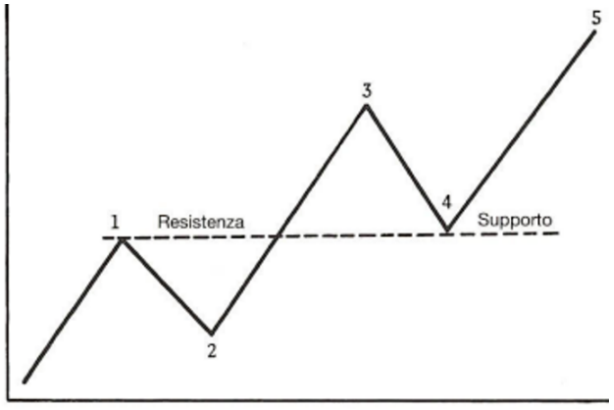
Gli interessi di tutti i gruppi elencati coincidono, cioè vogliono tutti acquistare al riavvicinarsi nuovamente dell'area di supporto, quindi è possibile che per un breve periodo anche se il trend è in rialzo i prezzi tendano a scendere fino a quando l'aumento degli ordini di acquisto attorno al prezzo di supporto farà proseguire l'uptrend.

Se al contrario i prezzi riuscissero a violare l'area di supporto si attiveranno i comportamenti opposti:

- **longs**: accortisi di aver fatto un errore, cercano di liquidare le loro posizioni, anche se in perdita, o di piazzare ordini di vendita non appena il prezzo torni verso il precedente livello di supporto

- **short**: capendo di stare dalla parte giusta del mercato tentano di aumentare le loro posizioni di vendita

Di conseguenza un'area di supporto in seguito ad un calo del prezzo sotto i livelli chiave si tramuta in una resistenza e viceversa come si vede dalla seguente immagine:



[9]

dopo un deciso rialzo al punto 3 il ritracciamento si ferma al punto 4 quasi in corrispondenza del punto 1. Qui tutte le vendite che avevano creato la resistenza si sono trasformate in altrettanti acquisti cercando di seguire il trend rialzista, di conseguenza il livello di resistenza precedente diventa un supporto con il passare del tempo.

[9] J. J. Muprhy; Analisi tecnica dei mercati finanziari; 2° edizione; pag. 47

I supporti e le resistenze rappresentano fedelmente quello che gli investitori fanno sul mercato, e permettono agli analisti di determinare le reazioni e comportamenti di questi in base a determinati movimenti di prezzo.

L'analisi tecnica quindi altro non è che lo studio della psicologia e del comportamento umano al variare delle condizioni di mercato.[10]

Per far si che il processo di inversione avvenga non è sufficiente che l'area considerata venga violata ma è anche necessario che questo accada con un elevato livello di penetrazione tale da convincere gli operatori di aver commesso un errore di valutazione. La quantificazione di questo è soggettiva ma tendenzialmente gli analisti la determinano in base all'importanza del livello di supporto violato (secondo i parametri tempo, volume e momento): 3% in caso di livelli molto importanti e 1% in caso di livelli poco importanti.

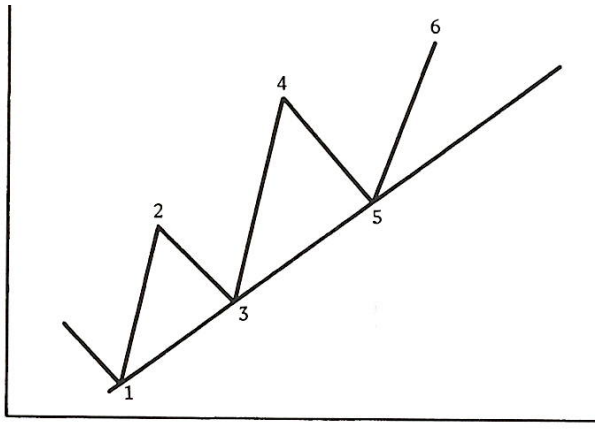
Altro aspetto psicologico da tenere in considerazione nell'analisi di supporti e resistenze sono i numeri tondi! Una volta determinata un'area di interesse spesso sono i numeri tondi in essa contenuti a rappresentare i livelli di prezzo a cui generalmente si fermano i rialzi o i ribassi, questo perché rappresentano dei livelli di supporto/resistenza anche psicologici. I trader approfittano di questi per anticipare i movimenti effettuando degli ordini di vendita/acquisto poco sotto/sopra tali valori.

1.2.4. Trendlines

Le linee di tendenza, o trendlines, sono uno strumento tecnico utilizzato durante le analisi negli approcci chiamati trend-following e sono rappresentate da linee rette che congiungono una serie di punti.

Come per i trend possiamo individuare trendlines:

- rialziste: quando la linea passa per i punti di minimo del grafico e l'inclinazione risulta essere verso l'alto. Solitamente si considerano importanti con una pendenza di almeno di 45°, e cioè quando i prezzi variano nella medesima percentuale del tempo.

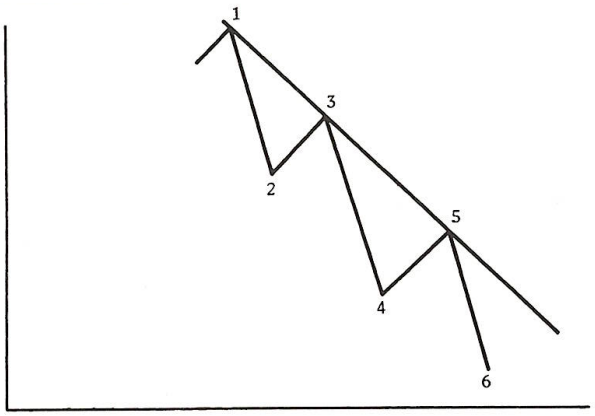


11

[10] J. J. Muprhy; Analisi tecnica dei mercati finanziari; 2° edizione; pag. 47
[11] J. J. Muprhy; Analisi tecnica dei mercati finanziari; 2° edizione; pag. 50

- ribassiste: quando la linea congiunge i punti di massimo e la pendenza è rivolta verso il basso.

12



Un'inclinazione eccessiva può significare che i prezzi si stanno muovendo troppo rapidamente e altrettanto velocemente potrebbero ridiscendere, l'eventuale rottura di questa linea può comportare un ritorno dei prezzi sulla linea di 45 gradi. Tuttavia si è osservato negli anni che quando ci si trova in questa casistica è meglio utilizzare come indicatori le medie mobili che altro non sono che trendlines curvilinee, quindi a seconda del tipo di trend è necessario individuare lo strumento più corretto per procedere all'analisi.

Ovviamente non è possibile tracciare una linea che passi per tutti i punti di interesse e una delle skills dell'analista consiste proprio nel saper individuare la linea corretta, solitamente seguendo i seguenti steps:

1. individuare il trend del grafico per capire quale linea dovrà tracciare
2. individuare due punti di massimo/minimo dove i secondi devono essere inferiori/superiori ai primi. Se si utilizzano i grafici a barre andrebbero considerati tutti i prezzi di ogni barra e non solo le chiusure
3. tracciare la linea che congiunge i punti. Ad esempio in un uptrend alcuni analisti tendono ad aspettare che il precedente massimo sia raggiunto o superato prima di procedere. Le linee che congiungono solamente due punti sono considerate provvisorie e per confermarne la validità devono essere ritestate almeno una terza volta rimbalzando e proseguendo il trend.

Le trendlines individuano delle aree di supporto o resistenza e come quest'ultime sono da considerarsi di importanza diversa in base:

- all'ampiezza dell'arco temporale per cui si estende
- al numero di volte in cui è stata testata positivamente, cioè il prezzo rimbalza e prosegue nella stessa direzione del trend

In entrambi i casi maggiore è il valore più importante risulterà la trendline e di conseguenza una sua rottura risulterà essere più significativa.

Si possono verificare delle rotture momentanee di trendline per brevi archi temporali che rappresentano dei falsi segnali e che possono essere individuati utilizzando diversi criteri, ad esempio:

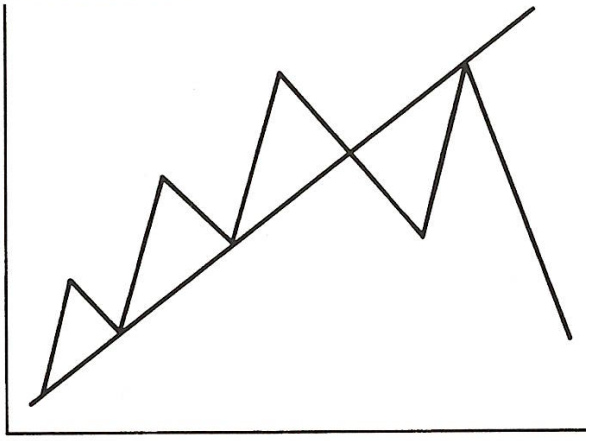
- filtri di tempo: se si tratta di breve periodo. Ad esempio per trendline giornaliere potrebbe essere un importante segnale se il prezzo chiude sotto/sopra la linea per 2 giorni consecutivi
- filtri di prezzo: quando la penetrazione della rottura non supera un determinato livello, solitamente il 3% sul prezzo di chiusura per le trendlines di lungo periodo. Più gli intervalli

[12] J. J. Muprhy; Analisi tecnica dei mercati finanziari; 2° edizione; pag. 51

di tempo sono ristretti più dovrebbe essere ridotta la percentuale di penetrazione da considerare

Per valutare le rotture l'analista deve essere bravo nell'individuare i corretti parametri da usare come filtri in base ai mercati e alla situazione del momento.

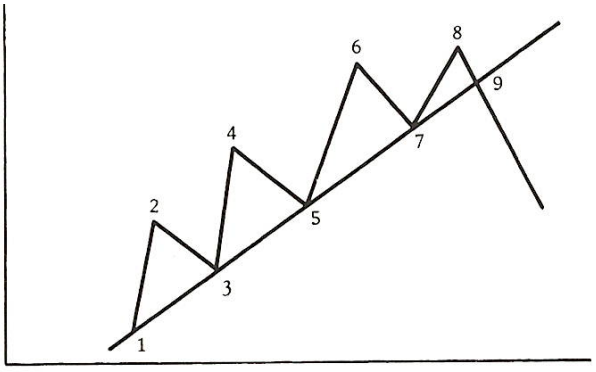
Così come i supporti e le resistenze anche le trendlines una volta violate definitivamente invertono i propri ruoli.



[13]

Le trendlines sono utilizzate nell'analisi tecnica per:

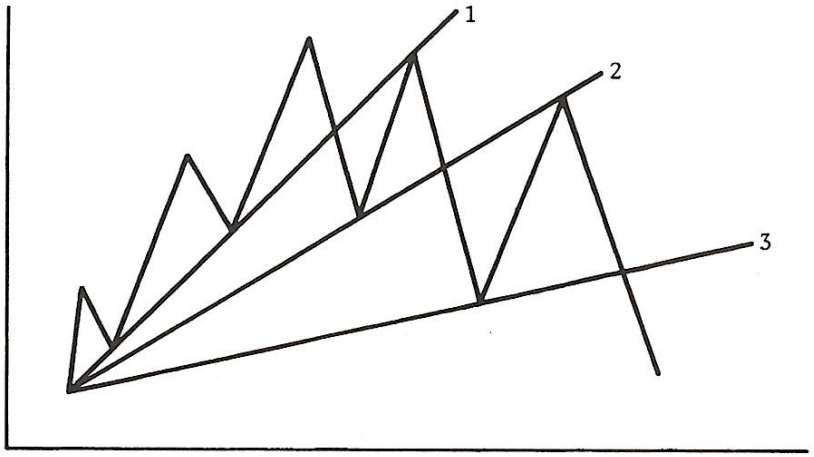
- capire la direzione e la velocità del trend in base all'inclinazione della linea tracciata, perché un trend per continuità tende a mantenere le sue caratteristiche nel tempo
- rappresentando dei veri e proprio supporti e resistenze:
    - individuare un'inversione di trend quando vengono violate
    - aree ideali per acquisti o vendite



[14]

[13] J. J. Muprhy; Analisi tecnica dei mercati finanziari; 2° edizione; pag. 56
[14] J. J. Muprhy; Analisi tecnica dei mercati finanziari; 2° edizione; pag. 53

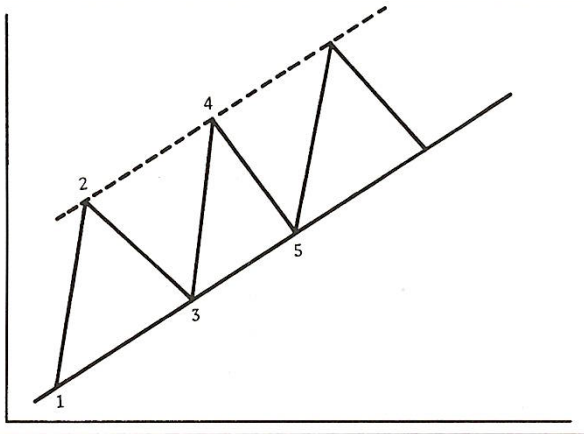
- (*Principio del ventaglio*) quando una trendlines viene rotta e il prezzo rimbalza sulla precedente senza però tornare sopra la prima e questo processo si ripete nuovamente è possibile disegnare delle trendlines in successione la cui forma e assimilabile a un ventaglio.[15]



Questo indica la direzione che sta prendendo il trend e la violazione anche della terza linea viene vista come importante segnale di un'inversione di tendenza.

- individuare degli obiettivi di prezzo: una volta oltrepassata con decisione la linea si è osservato che i prezzi tendono ad allontanarsi di uno spazio equivalente alla distanza verticale tra la precedente linea e il punto di inversione di trend

- determinazione di un canale: i prezzi a volte fluttuano all'interno di un vero e proprio canale individuato da un lato dalla linea di trend principale e dall'altro dalla linea di canale che viene disegnata partendo dal primo massimo/minimo e tracciando una linea parallela alla trendline principale.[16]



L'individuazione di un canale è molto importante in quando un trader può utilizzare la trendline principale per aprire delle posizioni e il range individuato per cercare profitti di breve periodo vendendo quando i prezzi si avvicinano al top del canale.

Il canale poi eredita anche le proprietà delle trendlines che lo generano e quindi:

  - risulterà più importante più si estende nel tempo e quanto più è stato testato positivamente
  - la rottura della trendline di un canale rilevante può rappresentare un'importante inversione di tendenza così come l'incapacità del prezzo di raggiungere e/o superare la linea di canale per diverso tempo
  - la rottura di una linea di canale può essere vista come un'importante accelerazione di trend, utilizzandola per un incremento di posizioni long. *Una volta che un canale è stato rotto i prezzi tendono a percorrere una distanza uguale alla profondità del canale*[17]

[15] J. J. Muprhy; Analisi tecnica dei mercati finanziari; 2° edizione; pag. 58
[16] J. J. Muprhy; Analisi tecnica dei mercati finanziari; 2° edizione; pag. 64
[17] J. J. Muprhy; Analisi tecnica dei mercati finanziari; 2° edizione; pag. 67

1.2.5. Percentuale di ritracciamento

Dopo un importante movimento di mercato i prezzi sono soliti ritornare in parte indietro prima di proseguire nel trend. Proprio per questa caratteristica è possibile cercare di individuare dei livelli chiave in cui piazzare degli ordini di acquisto/vendita e questo avviene con l'utilizzo della tecnica del ritracciamento, che definisce dei livelli fissi individuati con il passare del tempo che variano in base alla teoria considerata:

Teoria di Dow	Rapporti di Fibonacci/teoria delle Onde di Elliot	W.D.Gann(1/8,2/8,3/8.....8/8)
33%	38%	3/8(38%)
50%	50%	4/8(50%)
66%	62%	5/8(62%)

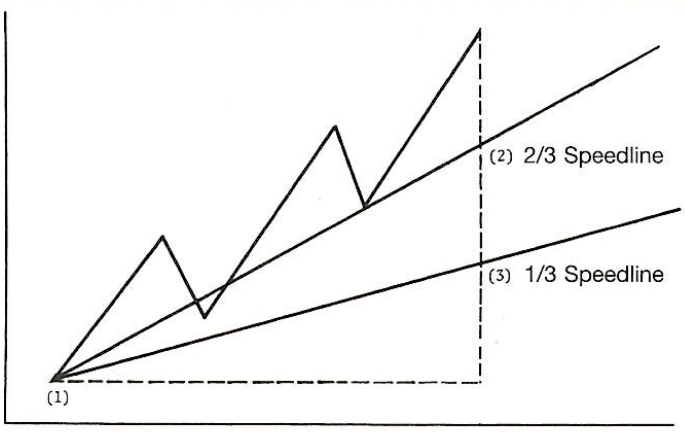
Nonostante la teoria di Gann suddivide la variazione in ottavi i più importanti sono le frazioni che corrispondono alle percentuali indicate.

Questo metodo può essere applicato a tutti i tipi di trend e viene utilizzato dai traders per individuare aree di acquisto/vendita convenienti.

Se i prezzi rompono anche il livello più alto di ritracciamento 62-66%, potrebbe essere interpretato come un possibile segnale di debolezza e/o di inversione di tendenza con conseguente ritracciamento del 100%.

Combinando la percentuale di ritracciamento e le trendlines è stata sviluppata la ***tecnica delle speedlines:*** [18]

Consiste nel dividere il trend in terzi per mezzo di una linea verticale con massimo nel punto più alto del trend e minimo in corrispondenza del punto più basso. Si tracciano poi linee dal punto di inizio del trend che ricadono sui valori 1/3 e 2/3 della verticale. Ogni volta che si verifica un nuovo massimo o minimo sarà necessario apportare degli accorgimenti. Come tutte le trendlines e le aree di supporto/resistenza anche le speedlines invertono i loro ruoli una volta una volta rotte.



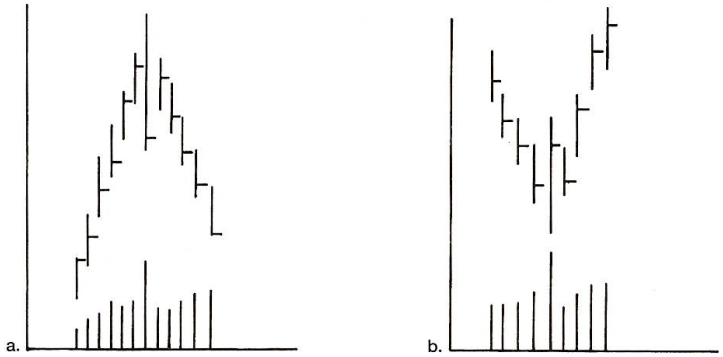
[18] J. J. Muprhy; Analisi tecnica dei mercati finanziari; 2° edizione; pag. 70

1.2.6. Reversal & Price gap

I reversal sono formazioni particolari che vengono utilizzate nell'analisi di altre figure tecniche.

Si parla di reversal day quando in un giorno si ha un massimo/minimo seguiti da una chiusura inferiore/superiore:

- top reversal day: uptrend, nuovo massimo seguito da una chiusura inferiore
- bottom reversal day: downtrend, nuovo minimo seguito da una chiusura superiore
- buy climax: top reversal day con elevati volumi di acquisto. Può rappresentare punto di inversione o un punto di massimo rilevante



[19]

- selling climax: bottom reversal day con elevati volumi di liquidazione. Può rappresentare punto di inversione o un punto di minimo rilevante
- key reversal day: uptrend, prezzo di apertura individua un nuovo massimo ma poi chiude sotto il prezzo di chiusura del giorno precedente. In downtrend è il contrario

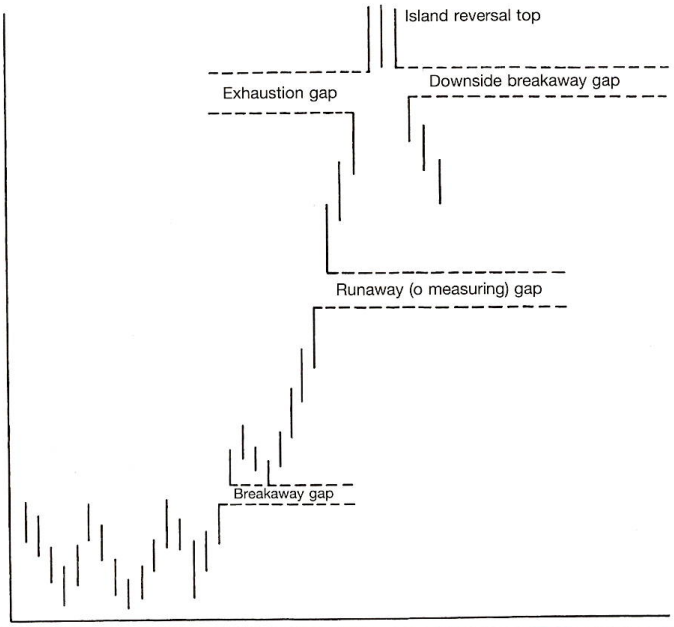
Queste figure "**reversal**" hanno una maggiore importanza quando individuate su grafici con intervalli temporali di settimane o mesi (es. upside weekly reversal corrisponde al bottom reversal day).

I **price gap** sono dei veri e propri buchi sul grafico che si manifestano quando non avviene nessuno scambio e i prezzi aprono la seduta di mercato sopra il massimo del giorno precedente in un trend rialzista, upside gap, o sotto i minimi in un trend ribassista, downside gap. Entrambi rappresentano un importante segnale di forza o di debolezza del mercato, soprattutto quando compaiono su grafici settimanali o mensili, e delineano rilevanti aree di supporto o resistenza.

[19] J. J. Muprhy; Analisi tecnica dei mercati finanziari; 2° edizione; pag. 73

Esistono 3 tipi di gaps:

1. **breakaway**: accompagnati normalmente da forti volumi, segnalano in genere l'inizio di un importante movimento di mercato e si verificano solitamente sul completamento di importanti formazioni (es. sulla rottura della neckline di un head and shoulder). È possibile che questi gap vengano ritestati o addirittura colmati (in parte o totalmente) da eventuali ritracciamenti del prezzo, rappresentando un segnale di debolezza del trend. Più il volume risulta elevato più difficilmente questo avverrà.[20]



2. **runaway**: si manifesta quando un movimento è stato in atto per un certo arco temporale e la formazione sta flettendo con volumi moderati di mercato. In questi casi si tendono a formare dei gap nella direzione del trend che in uptrend indicano un mercato molto forte mentre in downtrend il contrario. I runaway gap fungono da supporti/resistenze forti per future correzioni e difficilmente vengono colmati. Se avviene in uptrend è considerato un segnale molto negativo per il mercato.
3. **exhaustion**: si verifica al termine di un movimento di mercato. L'analista, dopo che gli altri tipi di gaps si sono verificati, attende l'exhaustion gap in quanto i prezzi al termine della formazione tendono a avere un ultimo balzo nella direzione del trend seguito da una correzione. La chiusura dei prezzi al di sotto(uptrend) o al di sopra(downtrend), cioè quando il gap viene colmato, indica la fine del movimento. A seguito della formazione di questo tipo di gap può succedere che i prezzi tendano a muoversi all'interno di un range per giorni o settimane seguiti poi da un ulteriore gap contrario, dando vita alla formazione chiamata "Island reversal" che indica un'inversione di trend molto importante.

1.2.7. Formazioni principali

L'obiettivo principale di un analista è di individuare tempestivamente eventuali cambiamenti o continuazioni di trend e questo avviene con l'utilizzo di alcune formazioni specifiche identificabili sui grafici.

Esistono due categorie di formazione:

- d'inversione: indicano un'importante inversione di trend
- di continuazione: rappresentano una pausa nel movimento

Verranno ora analizzate le diverse figure e anche in questo caso il volume risulta avere un ruolo fondamentale per la conferma della struttura individuata.

[20] J. J. Muprhy; Analisi tecnica dei mercati finanziari; 2° edizione; pag. 75

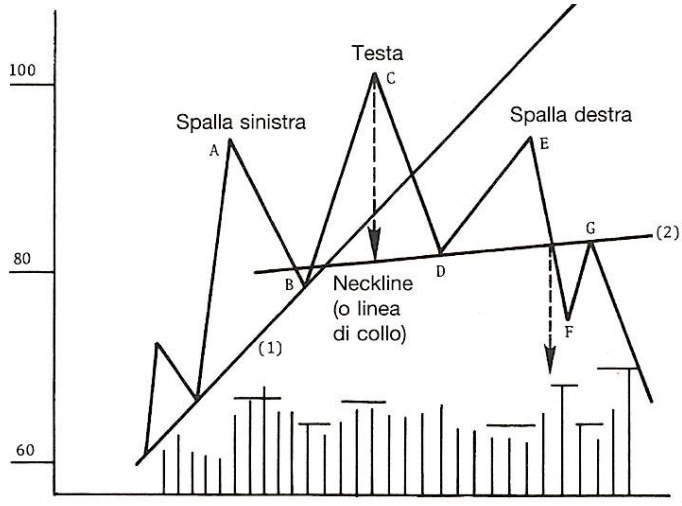
1.2.7.1. Formazioni d'inversione

Le formazioni di inversione sono caratterizzate dai seguenti fattori:

- la presenza di un trend primario: la formazione deve avere qualcosa da invertire altrimenti il suo presentarsi è da considerarsi sospetto
- viene rotta un'importante trendline: non sempre indica un'inversione ma da un primo segnale che è in atto un cambiamento nel trend
- più la figura è ampia (duratura nel tempo) e alta (volatile) più il movimento dei prezzi che ne segue sarà forte
- le formazioni di topping sono più brevi e più voltatili di quelle di fondo
- i fondi hanno un range di prezzi più limitato
- il volume risulta più importante nella fase rialzista: solitamente il volume aumenta nella direzione del trend fino a un massimo al completamento della figura. Nelle prime fasi di inversione del trend rialzista non risulta così fondamentale anche se un incremento delle attività viene interpretato come un segnale di un ribasso dei prezzi. L'aumento di volume risulta invece molto importante quando avviene l'inversione di un trend ribassista.

1.2.7.1.1. Testa e spalle & testa e spalle rovesciato

Partendo come sempre dal concetto di trend primario questa figura si crea quando le serie di massimi e minimi cominciano a perdere di intensità, a causa del bilanciamento della domanda e dell'offerta, dando vita a una formazione orizzontale, fase di distribuzione, che si conclude con la rottura dei livelli di supporto/resistenza invertendo il trend.

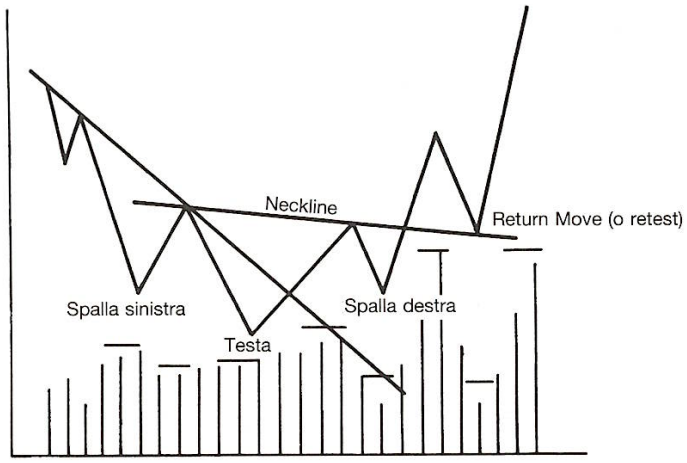


[21]

In **uptrend** la creazione della formazione "testa e spalle"(head and shoulders) avanza senza segnali di top, cioè con massimi successivi (punto C ed E) con volumi inferiori ai precedenti (punto A). Si forma una figura con 3 massimi riconoscibili di cui il centrale C, superiore agli altri, rappresenta la testa e i laterali, A ed E, le spalle. Il secondo massimo C tende ad avere volumi inferiori rispetto alla spalla sinistra A, segnale che la pressione di acquisto sta diminuendo. L'indicazione più importante che permette di riconoscere la struttura è che il volume sul terzo massimo E deve essere molto inferiore ai due precedenti. Superata la testa i massimi successivi non riescono a raggiungere o superare questo livello e se viene rotto il supporto principale, trendline, si può già parlare di inizio di un trend laterale sufficiente per liquidare posizioni lunghe ma non abbastanza per andare short. A questo punto si può disegnare la nuova linea di supporto, neckline o la linea di collo, che unisce i due minimi successivi. Se anche quest'ultima viene rotta in chiusura **con aumento di volumi** è possibile prevedere l'inizio di un downtrend, con una caduta del prezzo **minima** sotto quest'ultima stimabile almeno pari alla distanza verticale tra la testa e il collo dal punto di rottura oppure al doppio della lunghezza della prima onda al ribasso. Se

[21] J. J. Muprhy; Analisi tecnica dei mercati finanziari; 2° edizione; pag. 82

questi target minimi vengono superati il prossimo obiettivo di prezzo corrisponderà con il valore da cui è iniziato tutto il movimento tenendo sempre però in considerazione eventuali zone di supporto/gaps presenti sul grafico più ampio. Gli eventuali rimbalzi dovranno avere volumi ridotti e la formazione può considerarsi completa se si verifica il criterio di penetrazione di 1-3% o la permanenza sotto la linea per 2 giorni consecutivi, se intervalli giornalieri.



[22]

In **downtrend** invece si parla della formazione detta "testa e spalle rovesciato" (inverse head and shoulders). Come per il testa e spalle classico abbiamo 3 distinti picchi di bottoms con il secondo, la testa, inferiore alle spalle e anche qui per il completamento della figura è necessario che il prezzo oltrepassi la neckline con volumi. In questo caso il volume ha un'importanza ancora maggiore rispetto al caso precedente ed è assolutamente necessario che la rottura avvenga con elevati volumi perché il mercato può scendere anche solo per mancanza di interesse o inerzia mentre sale solamente quando la pressione di acquisto è superiore a quella di vendita. Per la prima parte della figura funziona in modo simile al caso in uptrend, quindi il volume sulla testa deve essere inferiore rispetto a quello sulla spalla sinistra, ma il rimbalzo prima della spalla destra fino alla neckline deve avvenire con un volume superiore rispetto alla spalla sinistra. L'eventuale rottura della neckline poi avverrà con un ulteriore aumento di volumi. Anche qui i movimenti di ritorno dovrebbero essere accompagnati da bassi volumi e le tecniche utilizzate per calcolare specifici obiettivi di prezzo rimangono le stesse.

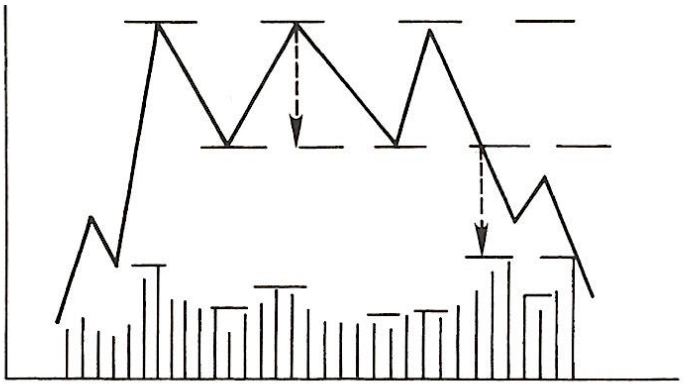
Queste formazioni non sono infallibili, in molti casi funzionano ma possono anche rappresentare solamente delle fasi di consolidamento e può accadere che il prezzo continui nella direzione del trend principale. Per questo motivo gli analisti tendono a usare un sistema di filtri per cercare di avere una maggior probabilità di individuare ed eliminare i falsi positivi, ad esempio:

- considerare solamente il prezzo di chiusura per eventuali rotture escludendo quelle intra-day
- filtro di prezzo: utilizzare delle percentuali di penetrazione, dal 1% al 3%
- filtro temporale: il superamento dei supporti o resistenza dovrebbe avvenire per almeno un certo arco temporale (almeno 2 giorni se daily)

Nonostante queste precauzioni è possibile che avvengano delle situazioni anomale, ad esempio in uptrend un nuovo massimo prima dell'inversione e in questi casi si parla di "Bull Trap".

[22] J. J. Muprhy; Analisi tecnica dei mercati finanziari; 2° edizione; pag. 87

1.2.7.1.2. Tripli massimi e minimi

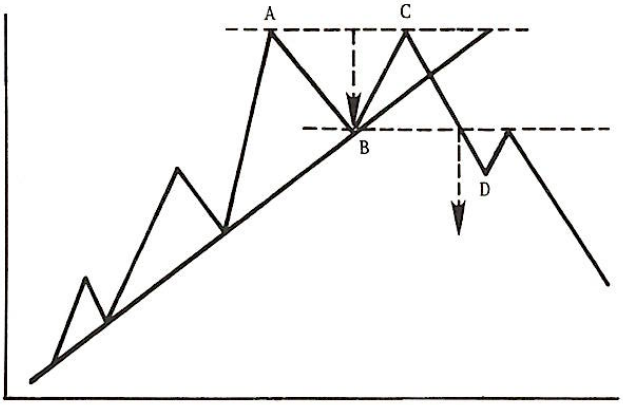


Queste figure riprendono molte particolarità dei "testa e spalle" e "testa e spalle rovesciato" viste in precedenza e sono una sorta di variante particolare dove la differenza principale sta nel fatto che i 3 top e i 3 bottom si trovano sullo stesso livello.

[23]

1.2.7.1.3. Doppio top e doppio bottom

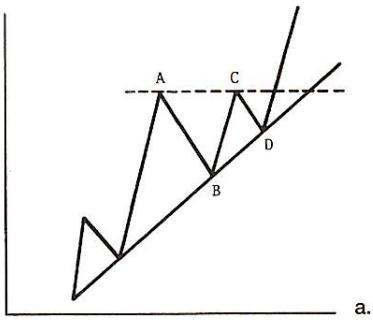
Dopo gli "head and shoulders" sono le figure più comuni e riconoscibili, riconducibili a delle M in caso di top e W in caso di bottom. Si distinguono dalle precedenti proprio per il fatto che i punti di massimo o minimo sono 2 invece di 3.

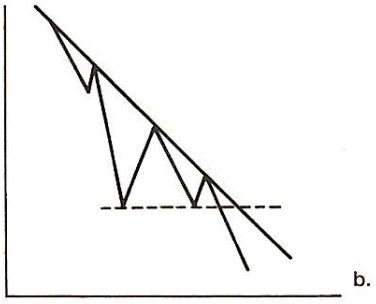


[24]

La formazione del volume e la misurazione dei prezzi target resta la stessa. Le figure sono considerate potenziali e di consolidamento finché i supporti/resistenze non vengono violati.

Molti movimenti naturali, ad esempio il rimbalzo dei prezzi da un minimo o un ritracciamento da un massimo, vengono spesso confusi con queste figure di inversione ed è per questo motivo che sarebbe sempre meglio aspettare che le figure si completino prima di aprire posizioni. Un periodo di intervallo indicativo è di almeno un mese.[25]

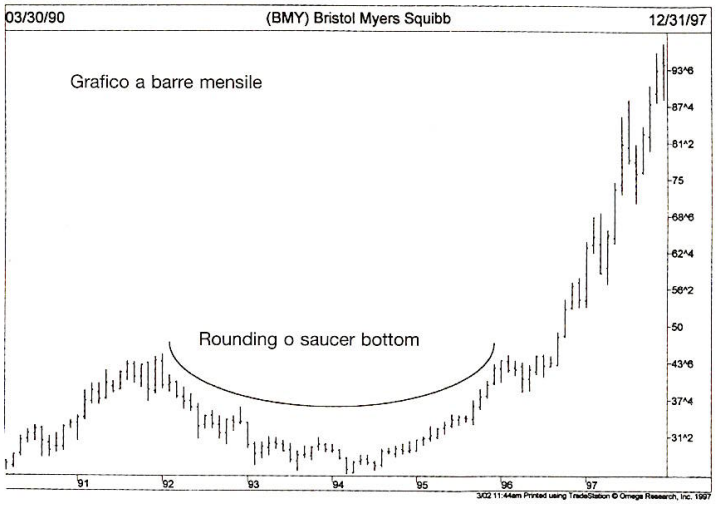




[23] J. J. Muprhy; Analisi tecnica dei mercati finanziari; 2° edizione; pag. 91
[24] J. J. Muprhy; Analisi tecnica dei mercati finanziari; 2° edizione; pag. 93
[25] J. J. Muprhy; Analisi tecnica dei mercati finanziari; 2° edizione; pag. 97

1.2.7.1.4. Soucer e spikes

Le ultime figure di inversione che consideriamo sono i soucer e gli spikes. I primi caratterizzati da un movimento lento laterale che poi diventa crescente, sono individuati su grafici settimanali o mensili.



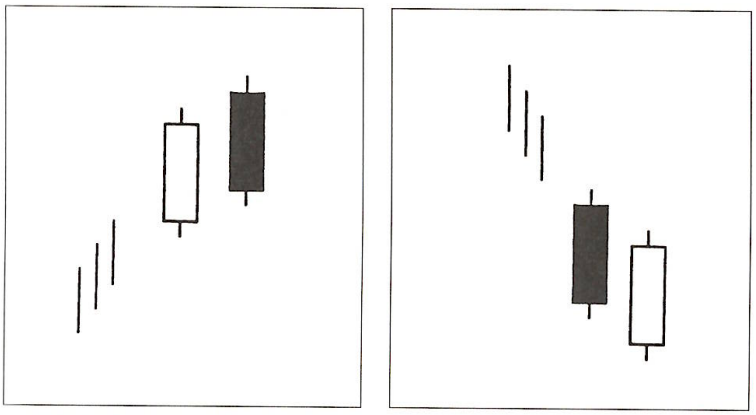
Gli Spikes invece sono formazioni a V che si verificano in modo improvviso, con periodi di transizione brevissimi o addirittura assenti, in seguito a notizie negative in un mercato molto tirato.

[26]

1.2.7.1.5. Formazioni di inversione con Candlestick

Con l'utilizzo delle candlestick sono state classificate 40 potenziali forme di inversione, le più comuni sono le seguenti:

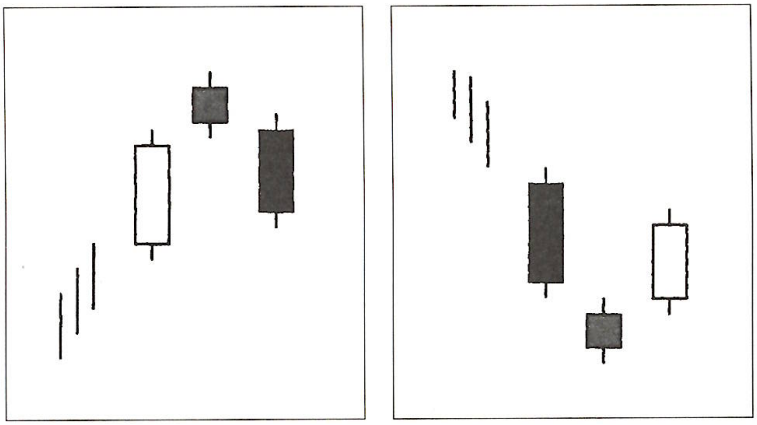
- **Dark cloud clover**: formazione negativa che si sviluppa in 2 giorni, in cui il primo prevede una candela positiva con corpo molto esteso a conferma di un uptrend in corso. Nel secondo giorno dopo un'apertura positiva, che supera il massimo precedente, i prezzi girano al ribasso oltrepassando la metà della candlestick del giorno precedente, creando così un corpo negativo che corrisponde a un possibile segnale di vendita.[27]



- **Piercing line:** formazione opposta alla precedente con una candela negativa seguita da una positiva che dopo l'apertura inferiore chiude al rialzo sopra la metà della candlestick precedente. Possibile segnale rialzista.

[26] J. J. Muprhy; Analisi tecnica dei mercati finanziari; 2° edizione; pag. 99
[27] J. J. Muprhy; Analisi tecnica dei mercati finanziari; 2° edizione; pag. 233

- **Evening star e morning star**: formazioni d'inversione (ribassista e rialzista) su 3 giorni molto potenti. Ad esempio per le evening star, ribassista, nel primo giorno è presente una candela positiva con corpo molto esteso seguita da un secondo giorno che inizia con un gap al rialzo ma prosegue creando una candela ribassista con corpo ristretto, detta *star*, che tende a raggiungere l'apertura del giorno precedente. Il terzo giorno si apre con un ulteriore gap al ribasso e prosegue con una candela negativa che percorre quasi tutto il corpo della candlestick del primo giorno.



[28]

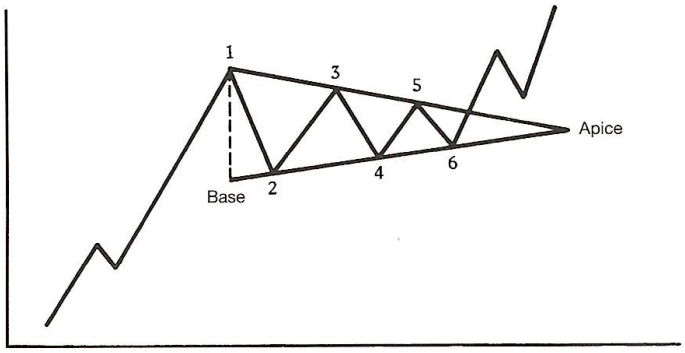
1.2.7.2. Formazioni di continuazione

Figure che solitamente indicano una pausa rispetto al trend principale del movimento dei prezzi prima che questi riprendano nella direzione principale.

1.2.7.2.1. I triangoli

Figure delimitate da 2 trendlines, il cui punto di incontro è chiamato apice, e una linea verticale sulla sinistra chiamata base, ne esistono di 4 tipi:

- **Simmetrico(coil):** le trendlines sono convergenti, ciascuna passa per almeno due punti di reazione (1,2,3,4…) di massimo/minimo che risultano essere minore/maggiore del precedente con volumi via via in diminuzione. Il tempo limite per la risoluzione del triangolo si trova in corrispondenza dell'apice e solitamente verso i 2/3 o 3/4 della sua lunghezza i prezzi rompono la figura nella stessa direzione del trend precedente con volumi importanti.

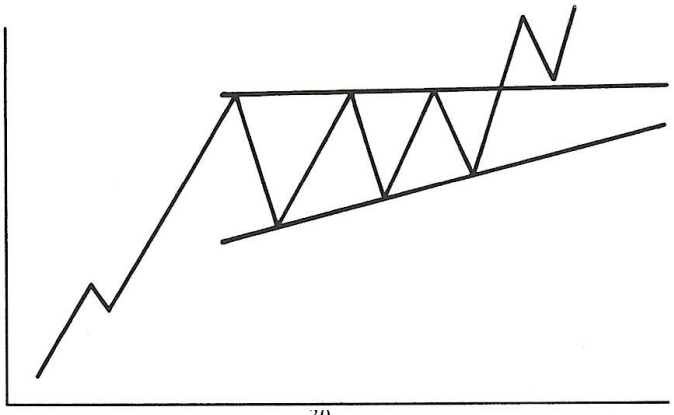


[29]

Per calcolare eventuali prezzi target vengono utilizzate 2 tecniche:
  - Misurare l'altezza della base e proiettarla nel punto di rottura nella direzione del trend
  - Disegnare una trendline rialzista parallela, in caso di uptrend, al supporto del triangolo che individuerà i possibili obiettivi di prezzo

[28] J. J. Muprhy; Analisi tecnica dei mercati finanziari; 2° edizione; pag. 234
[29] J. J. Muprhy; Analisi tecnica dei mercati finanziari; 2° edizione; pag. 102

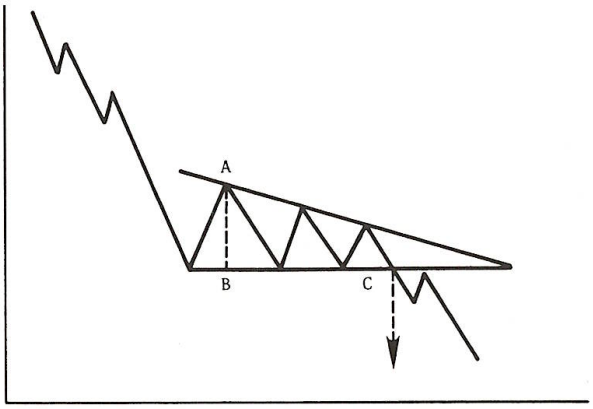
- **Ascendente**: in questa tipologia la trendline superiore è piatta e quella inferiore ascendente, il che significa una maggior pressione di acquisto rispetto alla vendita. Solitamente questa figura è da considerarsi rialzista e viene rotta verso l'alto con notevole aumento di volume. Per calcolare il prezzo target viene misurata l'altezza nel momento di maggior volatilità della formazione, punto più ampio del triangolo, e questa viene proiettata verso l'alto nel punto di rottura.



30

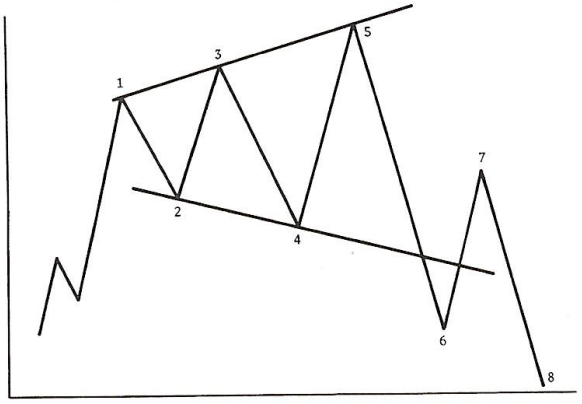
Molto spesso questa figura si presenta in una formazione di bottoming e rappresenta la possibile fine di un downtrend.

- **Discendente**: la trendline inferiore è piatta mentre quella superiore è inclinata verso il basso, maggior aggressività da parte dei venditori, ed è considerato una figura ribassista. La rottura con incremento dei volumi del supporto inferiore della figura rappresenta un forte segnale ribassista e di proseguo di downtrend. Il calcolo di obiettivi di prezzo avviene esattamente come per il triangolo ascendente e al contrario di quest'ultimo questa figura a volte si trova al top di un mercato rialzista e ne indica la perdita di slancio.



31

- **In espansione (broadening):** figura piuttosto rara in cui le trendlines sono divergenti creando una specie di triangolo aperto in espansione dove il volume invece di diminuire nella formazione tende ad aumentare, sintomo di un mercato molto emotivo. Solitamente è una figura ribassista che compare al termine di un mercato fortemente rialzista.



32

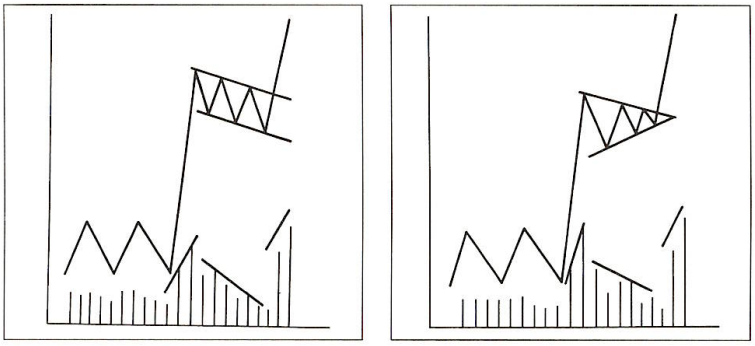
[30] J. J. Muprhy; Analisi tecnica dei mercati finanziari; 2° edizione; pag. 106
[31] J. J. Muprhy; Analisi tecnica dei mercati finanziari; 2° edizione; pag. 108
[32] J. J. Muprhy; Analisi tecnica dei mercati finanziari; 2° edizione; pag. 110

I triangoli sono figure la cui formazione solitamente richiede da 1 a 3 mesi almeno, se si individua in un tempo minore probabilmente può trattarsi di altre figure come le **bandiere e i pennants**.

Quest'ultime sono figure abbastanza comuni che si formano in procinto dell'esaurimento di un trend e rappresentano una pausa dei prezzi da un forte movimento di mercato avvenuto precedentemente(flagpole). Sono ritenute comunque figure di continuazione che solitamente appaiono a metà del movimento e vengono rotte nella stessa direzione del trend principale con esplosione di volumi.[33]

Le **Flags** sono riconducibili a dei rettangoli delimitati da trendlines parallele con pendenza opposta alla direzione del trend principale.

Il **Pennant** è delineato da 2 trendlines convergenti come i triangoli simmetrici.

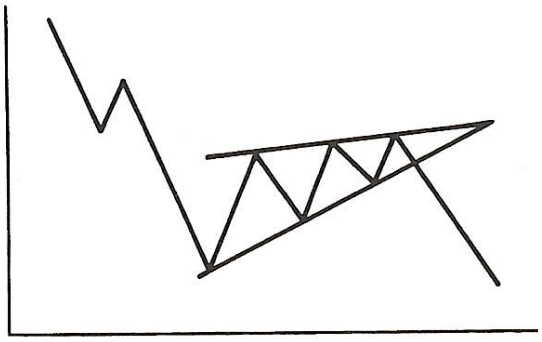


In entrambi i casi il volume diminuisce notevolmente durante la formazione, sono figure di breve periodo, solitamente 1-3 settimane, e tendono a essere più rapide in downtrend.

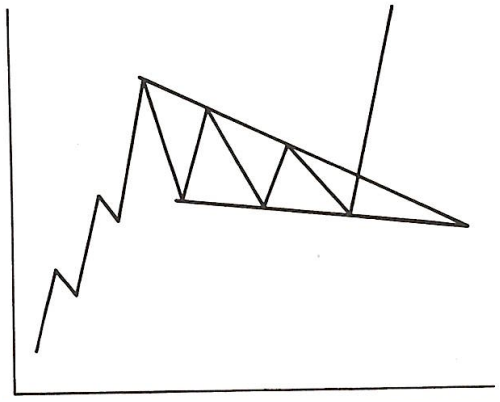
Gli obiettivi di prezzo vengono calcolati in modo simile per entrambe: viene misurata la distanza tra la rottura dell'ultimo importante supporto/resistenza e il bottom/top precedenti alle figure. Quest'ultima viene poi riportata nel punto di rottura nella direzione del trend

1.2.7.2.2. Wedge

Questa formazione di continuazione è simile al triangolo simmetrico sia per il tempo di formazione, 1-3 mesi, che per la conformazione, delineato da 2 trendlines convergenti in un apice. Caratterizzata da un'inclinazione rialzista (rising wedge, segnale bearish) o ribassista (falling wadge, segnale bullish) in direzione opposta al trend.



[34]

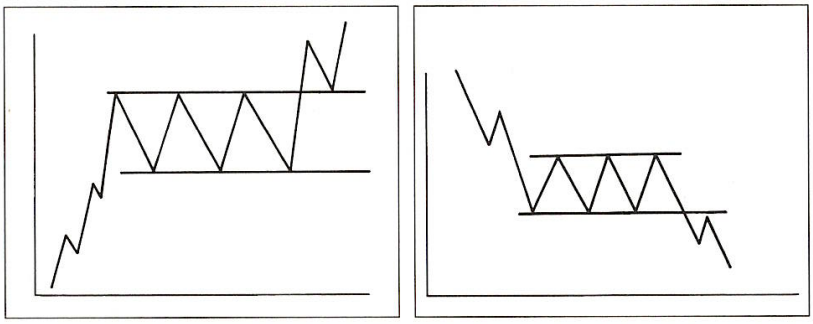


Solitamente si presentano durante un trend principale ma se si manifestano in un momento di top/bottom del mercato possono indicare un'inversione di tendenza.

[33] J. J. Muprhy; Analisi tecnica dei mercati finanziari; 2° edizione; pag. 111
[34] J. J. Muprhy; Analisi tecnica dei mercati finanziari; 2° edizione; pag. 114-115

1.2.7.2.3. Rettangolo

Figura più semplice da individuare dove i prezzi in pausa dal trend tendono a muoversi all'interno di due linee parallele orizzontali ben definite che fungono da supporto e resistenza.



[35]

Forma di consolidamento in cui è necessario prestare attenzione in quanto i massimi/minimi potrebbero essere visti come segnali di inversione. Per questo motivo è molto importante osservare i volumi: in uptrend le correzioni dovrebbero avvenire con bassi volumi mentre i rimbalzi con elevati volumi per dare luogo a una continuazione del trend sulla rottura, se avvenisse il contrario potrebbe trattarsi di un'inversione.

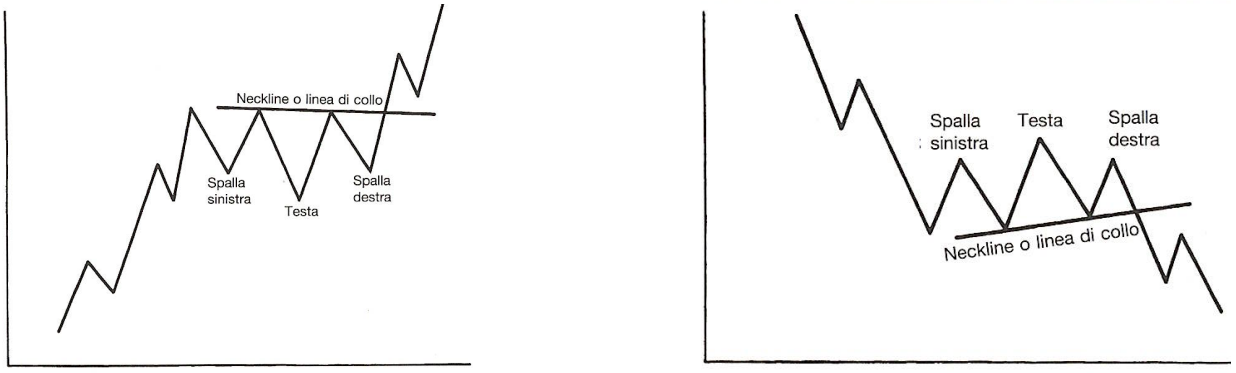
Uno degli aspetti più positivi di questa figura è che si può sfruttare il range di prezzi individuato per acquisti attorno ai minimi e vendite all'incirca sui massimi, oppure il contrario in caso si operasse allo scoperto, per effettuare profitti rapidi in breve periodo con basso rischio.

Il completamento di questa figura richiede 1-3 mesi come le precedenti ma la distribuzione del volume è differente: le ampie oscillazioni del prezzo ostacolano la caduta delle attività.

Per calcolare il prezzo target si misura l'altezza del range e la si proietta nel punto di rottura nella direzione del trend.

1.2.7.2.4. Testa e spalle come figura di continuazione

Il testa e spalle in certi casi può rappresentare anche una formazione di continuazione: come nel caso precedente anche in questo caso i prezzi tendono a formare una sorta di rettangolo con il secondo minimo/massimo inferiore/superiore agli altri 2 e danno luogo a dei testa e spalle rovesciati in uptrend e downtrend rispetto a quelli di inversione prima descritti



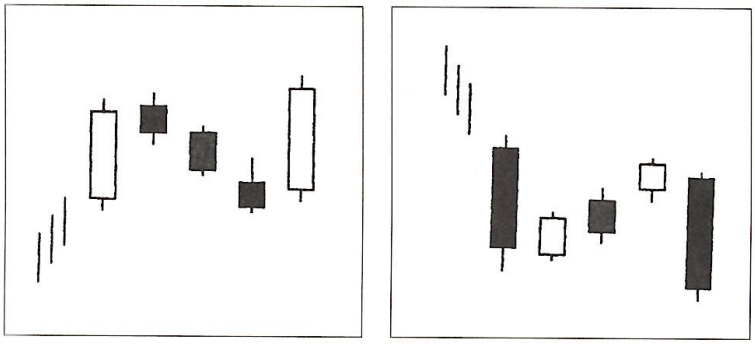
[36]

[35] J. J. Muprhy; Analisi tecnica dei mercati finanziari; 2° edizione; pag. 116
[36] J. J. Muprhy; Analisi tecnica dei mercati finanziari; 2° edizione; pag. 120-121

1.2.7.2.5. Formazioni di continuazione nelle Candlesticks

Per questa tipologia sono state identificate 16 formazioni candlesticks e la più importante è la seguente:

- Rising e falling three methods: la rising three method è una figura rialzista che si sviluppa in uptrend.



[37]

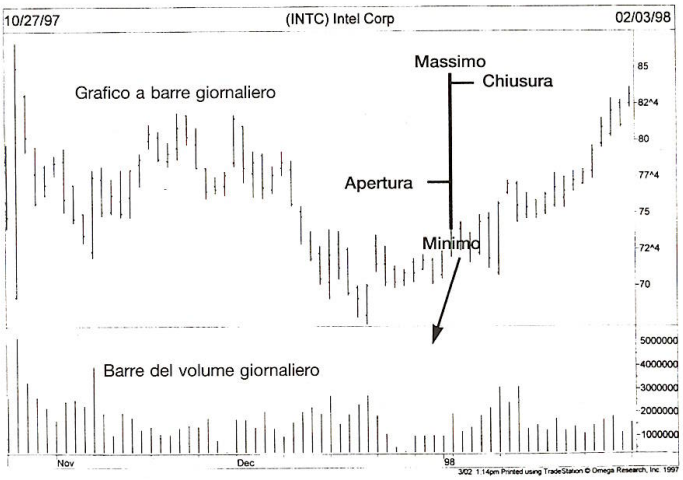
La prima giornata è costituita da un lungo candlestick a conferma del trend seguita da 3 giornate che tendono a correggere il movimento restando dentro il range della prima candela. Il 5° giorno della formazione ripresenta una candlestick positiva con corpo molto esteso che raggiunge un nuovo massimo e prosegue l'uptrend. Il periodo di riposo, correzione, può durare più di 3 giorni.
La falling three methods è il contrario di quella appena descritta.

[37] J. J. Muprhy; Analisi tecnica dei mercati finanziari; 2° edizione; pag. 235

1.2.8. Volume

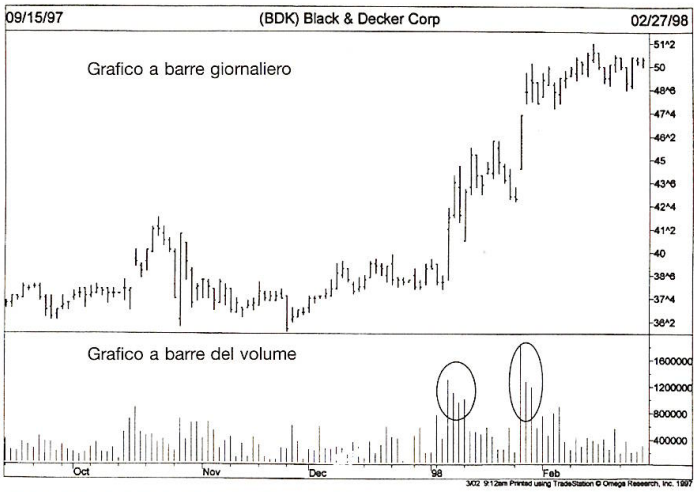
Il volume *è il numero delle entità trattate durante il periodo osservato*[38], rappresenta il totale degli scambi azionari in un determinato giorno di un dato titolo, e viene rappresentato da una serie di barre verticali situate sotto il grafico dei prezzi corrispondente.

Sul mercato azionario questo dato è immediatamente disponibile in ogni momento (mentre con i futures il giorno dopo) e molti analisti utilizzano un'analisi multidimensionale basata sul prezzo e sul volume quando si parla di azioni. (Gli operatori dei features aggiungono anche gli open interest).



39

Il volume viene considerato solitamente secondario nell'analisi rispetto al prezzo ma, come si è visto, diventa il fattore di primaria importanza quando usato come indicatore di conferma (ad esempio nel testa e spalle volumi decrescenti con il formarsi della figura che poi aumentano sulla rottura).



40

Un volume dovrebbe espandersi nella stessa direzione del trend dominante e contrarsi nei rimbalzi o ritracciamenti. Si ritiene inoltre che i volumi anticipino i prezzi, e cioè che una diminuzione della pressione in direzione del trend anticipi un'inversione delle tendenze di mercato.

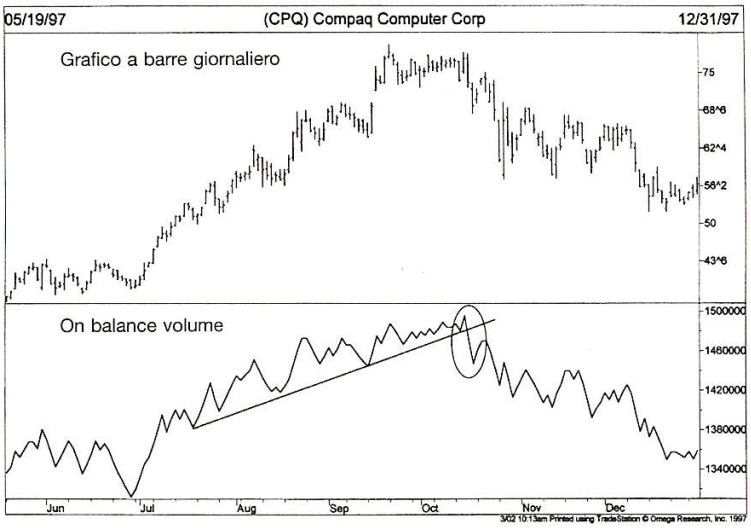
Tendenzialmente la relazione tra volumi e prezzi può essere riassunta come segue:

Prezzo	Volume	Mercato
sale	alto	forte
sale	basso	debole
scende	alto	debole
scende	basso	forte

[38] J. J. Muprhy; Analisi tecnica dei mercati finanziari; 2° edizione; pag. 123
[39] J. J. Muprhy; Analisi tecnica dei mercati finanziari; 2° edizione; pag. 32
[40] J. J. Muprhy; Analisi tecnica dei mercati finanziari; 2° edizione; pag. 124

1.2.8.1. Indicatori di volume

Con il fine di quantificare la pressione di acquisto o vendita gli analisti utilizzano degli indicatori di volume per elaborare delle conclusioni più ponderate. L'indicatore più conosciuto e usato è l'OBV (On balance volume) che forma una curva del volume sotto il grafico dei prezzi.



[41]

Al volume giornaliero viene assegnato un segno positivo o negativo in base al prezzo di chiusura rispetto al giorno precedente e il risultato è aggiunto/sottratto al volume totale.

La curva ottenuta viene analizzata con gli stessi indicatori impiegati per i grafici dei prezzi ed è usata per:

- conferma della qualità del trend, andamento congruente al grafico dei prezzi
- segnale di inversione, andamento divergente al movimento dei prezzi

Tale indicatore ha però dei limiti in quanto può assegnare un valore positivo o negativo al volume anche se le contrattazioni sono state solo leggermente positive o negative è questo ne pregiudica l'attendibilità.

1.2.9. Ordine nell'analisi: prima i grafici di lungo periodo.

Finora si è parlato di grafici prendendo come base i grafici giornalieri che risultano essere i più utilizzati da parte dei trader. I grafici di lungo periodo, settimanale o mensile, sono ugualmente se non ancora più importanti perché permettono di studiare la tendenza del mercato, non osservabile sui quelli di breve periodo. I trend su questi archi temporali sono più chiari e di più facile individuazione, le figure che si possono utilizzare nell'analisi sono le stesse ma alcune di queste, come i reversal, risultano essere più importanti e più efficaci nelle previsioni.

Durante un'analisi quindi è meglio partire dai grafici di lungo periodo in modo avere un quadro d'insieme per poi spostarsi su quelli di breve. In linea generale si individuano le trendlines, i supporti e le resistenze su:

1. Grafico ventennale: più significativi se in scala logaritmica, non utili ai fini del trading ma servono nell'analisi di mercato a scopo previsionale in modo da individuare il trend principale e determinare gli obiettivi di prezzo
2. Grafico settimanale: ultimi 5 anni
3. Grafico giornaliero: ultimi 6-9 mesi, usati nell'analisi per individuare il corretto timing di entrata e uscita

[41] J. J. Muprhy; Analisi tecnica dei mercati finanziari; 2° edizione; pag. 130

Sempre per il motivo che i prezzi del mercato scontano tutto non si ritiene necessario aggiornare i prezzi presenti sui grafici in base all'inflazione, dato che quest'ultimi l'avranno già inglobata al loro interno.

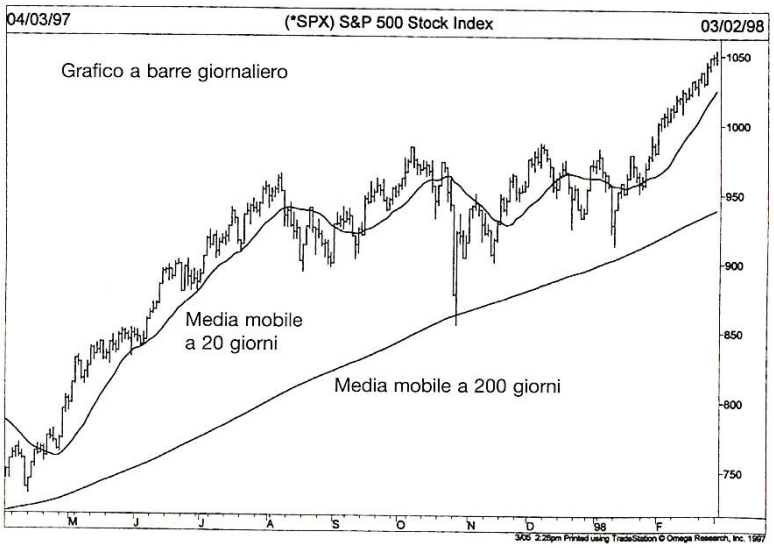
1.2.10. Medie mobili

Le medie mobili sono gli indicatori maggiormente utilizzati nell'analisi tecnica in quanto le loro regole possono essere facilmente utilizzate in programmi informatici, generando dei segnali non soggetti all'interpretazione e alla soggettività dell'analista, e possono essere facilmente testate o modificate.

Come dice il nome stesso vengono definite:

- "medie" perché raggruppano le ultime n chiusure (prezzo più utilizzato in questi casi) di nostro interesse e ne fanno una media estraendone un valore
- "mobili", il blocco di giorni/settimane/anni considerato cambierà spostandosi in avanti nel tempo

Lo scopo principale di chi utilizza le medie mobili è quello di identificare e segnalare l'inizio di un trend monitorandone poi gli sviluppi. Non servono a prevedere i movimenti di mercato in quanto per sua natura matematica seguono solamente lo sviluppo dei prezzi, ed essendo sempre in ritardo permettono di identificare un trend solo dopo che è già iniziato.



Mediando i prezzi si ottiene una linea smussata che permette una più facile identificazione del trend:

La media mobile di breve periodo sarà più vicina ai movimenti dei prezzi di una di lungo periodo, l'utilizzo di una o dell'altra dipende dal tipo di mercato e dalla sensibilità che si vuole avere.

[42]

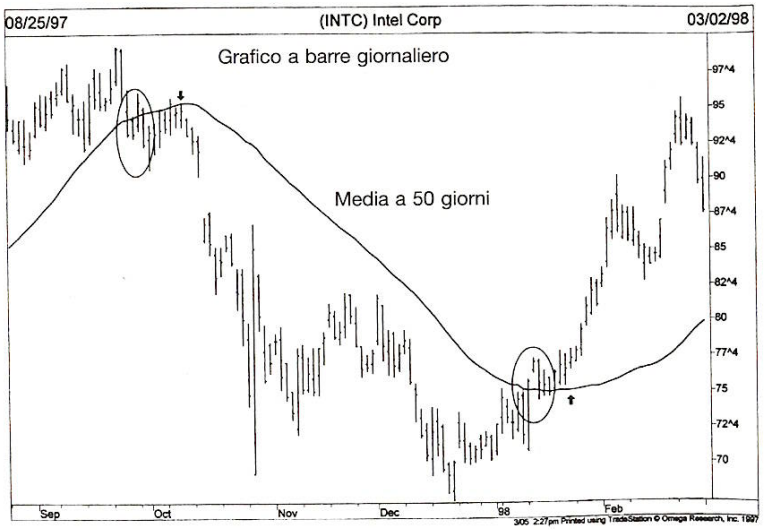
Esistono diversi tipi di medie mobili:

- **Semplice:** la più utilizzata in assoluto, prende in considerazione solo l'ultimo periodo e dà lo stesso peso a ogni prezzo in ogni intervallo valutato
- **Ponderate linearmente:** utilizzata per ovviare al problema relativo al peso delle medie semplici. Ogni prezzo viene pesato per l'intervallo di tempo a cui appartiene, ad esempio in caso giornaliero se si trattasse del valore del 10° giorno viene moltiplicato per 10 e così via,

[42] J. J. Muprhy; Analisi tecnica dei mercati finanziari; 2° edizione; pag. 157

il tutto viene poi diviso per la somma dei pesi. Anche questa tipologia però non riesce a fornire un'idea precisa di quello che sta per succedere nel mercato

- **Ponderata esponenziale:** sistema più sofisticato di calcolo che cerca di superare i limiti e le carenze delle due precedenti. Anche questa media ha lo scopo primario di dare un peso maggiore ai prezzi più recenti ma include anche i prezzi passati, ovviamente con pesi minori
- **Adattiva:** le medie mobili tendono a essere più precise in fase di tendenza seguendone la velocità ma meno quando i movimenti di mercato sono laterali creando molti falsi segnali. La media mobile adattiva invece aggiusta automaticamente la propria velocità in base al livello disturbo che riscontra, muovendosi più lentamente se il mercato è laterale e più velocemente quando viene individuato un trend. Questo avviene con l'utilizzo di un coefficiente chiamato ***efficiency ratio*** che confronta il movimento dei prezzi con la volatilità. Ad esempio se è alto, il movimento è maggiore della volatilità, viene favorita una media più veloce.



[43]

Gli analisti applicano il concetto di media mobile ai grafici, solitamente a barre o candlesticks, e tendono a interpretare come segnale di acquisto quando il prezzo di chiusura si trova sopra la media mobile considerata, di vendita il contrario. Spesso si attende come conferma che la media mobile prenda la stessa direzione del trend dei prezzi.

È necessario prestare molta attenzione all'intervallo di tempo considerato per la creazione di questi indicatori. Medie mobili di breve o brevissimo periodo sono molto sensibili, incrociandosi continuamente con i prezzi, forniscono anticipatamente informazioni relative al trend ma possono generare un numero elevato di falsi segnali. In ambito azionario si è visto che alcuni numeri presenti nella sequenza di Fibonacci, 13-21-34-55, risultano comportarsi molto bene come intervalli temporali da utilizzare anche se vengono usati sistematicamente le medie mobili a 50 giorni e 10 settimane sul breve periodo oppure 30-40 settimane sul lungo periodo. È possibile utilizzare dei software per analizzare il mercato e determinare quale siano le medie mobili che sono risultate più adatte nel tempo.

Le medie mobili sembrano avere inoltre un'importante relazione con i cicli temporali di mercato che verranno spiegati più avanti.

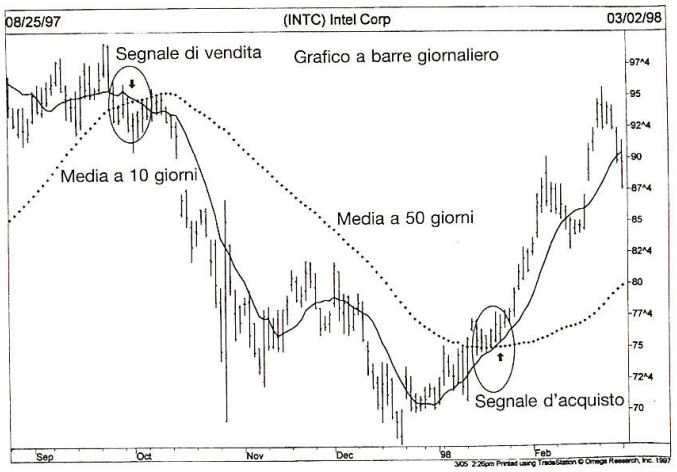
[43] J. J. Muprhy; Analisi tecnica dei mercati finanziari; 2° edizione; pag. 159

1.2.10.1. Utilizzo di più medie mobili

Singole si comportano bene fintanto che il trend rimane in essere ma non sono adatte a individuare tempestivamente le inversioni di tendenza. Per questo motivo e per eliminare alcuni falsi segnali, vengono solitamente utilizzate in gruppi di 2 o 3 contemporaneamente prestando attenzione ai reciproci comportamenti, incroci o ***crossover***. Gli intervalli più utilizzati sono 5-10-20-50 giorni e per costruzione in uptrend le medie più brevi son quelle maggiori, più vicine ai prezzi e sotto di esse vi saranno quelle più lunghe mentre in downtrend sarà il contrario e quelle più brevi saranno le più basse sul grafico.

Si parla di ***doppio crossover*** quando due medie mobili, solitamente con coppie temporali 5-20 o 10-50, si incrociano creando:

- un segnale di acquisto, se la media mobile più breve incrocia al rialzo quella di più lungo periodo
- un segnale di vendita allo scoperto: se la media più breve incrocia al ribasso quella più lunga



[44]

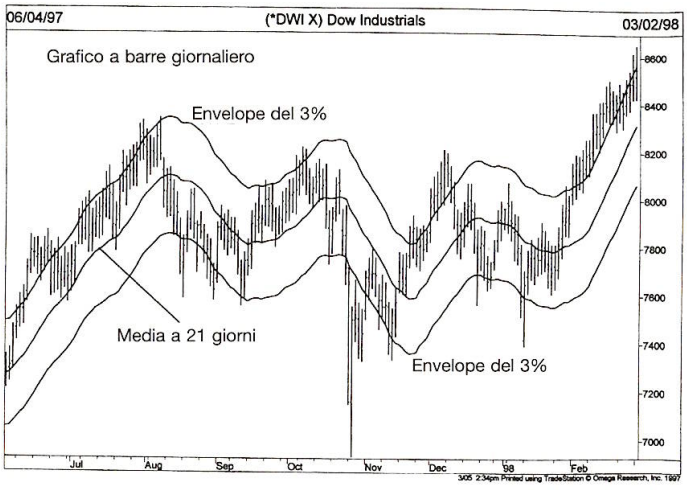
Si parla invece di ***triplo crossover*** (medie a 4-9-18 o 5-10-20 giorni) quando le medie mobili interessate sono 3:

- segnale di acquisto: in un downtrend quando la media più breve incrocia al rialzo le altre due. La conferma avviene quando anche quella di medio periodo incrocia quella più lunga
- segnale di vendita: quando in uptrend la media più breve incrocia le altre due al ribasso e anche in questo caso la conferma avviene quando la media mobile intermedia incrocia al ribasso quella di lungo peridio. Un primo possibile segnale di inversione, utilizzato dai trader per liberarsi di alcune posizioni, avviene quando la media più breve incrocia al ribasso quella media.

[44] J. J. Muprhy; Analisi tecnica dei mercati finanziari; 2° edizione; pag. 162

1.2.10.2. Envelopes e bande di Bollinger

Per migliorare l'utilità di una singola media mobile, oltre all'utilizzo di più medie mobili in contemporanea, vengono usati altri strumenti come gli Envelopes percentuali, che servono per capire quando un mercato è diventato troppo tirato (i prezzi si distaccano troppo dalla media mobile). Sono collocate a percentuali fisse sopra e sotto la media, ad esempio 3-5-10% per medie di 21 giorni,10 e 40 settimane.



[45]

Un'altra tecnica simile è quella delle bande di Bollinger. Anche qui due bande di trading vengono collocate intorno alla media mobile ma, invece che a una percentuale fissa, a distanza di deviazione standard (solitamente di 20 giorni, in questo modo il 95% dei dati sui prezzi sarà compreso nelle bande). Questa caratteristica di costruzione permette anche di avere informazioni sulla volatilità avuta nel periodo in questione e le bande saranno tanto più distanti quanto maggiore è stata e viceversa. Solitamente le bande tendono ad alternare fasi di contrazione e espansione, e quando l'espansione diventa eccessiva è speso segno di una possibile inversione. I limiti superiori e inferiori delle bande di Bollinger vengono inoltre utilizzati per identificare degli obiettivi di prezzo.

In entrambi i casi il mercato è considerato ipercomprato o ipervenduto quando le linee che delimitano le bande sono raggiunte e/o superate dai prezzi.

1.2.11. Oscillatori

Gli oscillatori sono indicatori secondari che risultano essere delle importanti aggiunte all'analisi del trend principale. I loro segnali sono considerati molto rilevanti soprattutto quando si opera nella direzione del trend e quest'ultimo è maturo e/o nella sua fase finale mentre nelle prime fasi, per costruzione, anticiperanno il trend e non devono avere la stessa importanza. Sono collocati solitamente al fondo dei grafici, si muovono all'interno di una barra orizzontale che spesso è suddivisa in due sezioni da una linea centrale in corrispondenza di un valore intermedio che dipende dalla formula utilizzata (solitamente 0 o 50).

Esistono diversi tipi di oscillatori ma le principali regole di interpretazione rimangono le stesse:

- quando viene raggiunto o ci si avvicina a un valore estremo dell'oscillatore, superiore o inferiore della banda, può significare che i prezzi si son mossi troppo velocemente in una direzione (mercato ipercomprato o ipervenduto) e quindi può anticipare una correzione di mercato o una

[45] J. J. Muprhy; Analisi tecnica dei mercati finanziari; 2° edizione; pag. 164

fase di consolidamento. Se si è in uptrend e si utilizza una strategia di acquisto, gli oscillatori possono indicare il timing giusto di ingresso individuando le fasi di ipervenduto e lo stesso avviene in downtrend, dove gli oscillatori possono indicare il momento di vendita o di short quando si è in ipercomprato

- bisognerebbe acquistare quando l'oscillatore è vicino alla barra inferiore e vendere quando è vicino alla banda superiore

- l'incrocio con la linea centrale viene utilizzato per determinare segnali di acquisto o vendita, tenendo sempre ben presente l'indicatore principale, il trend. Vengono aperte posizioni lunghe se la linea è incrociata al rialzo in un uptrend e posizioni short in caso contrario.

Gli oscillatori sono delle linee che si muovono all'interno di questa barra, e che uniscono i punti individuati utilizzando diverse formule.

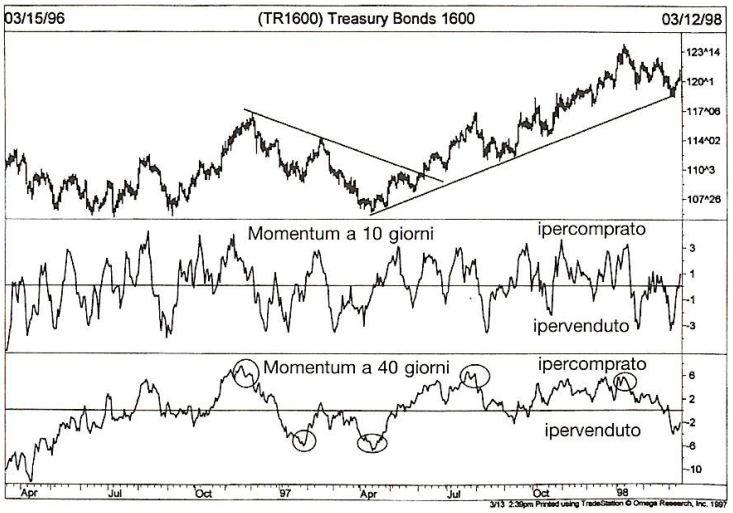
1.2.11.1. Tipi di Oscillatori

Come anticipato esistono diversi tipi di oscillatori, i principali sono:

- **Momentum:** questo oscillatore misura il tasso di variazione dei prezzi rispetto ai loro effettivi livelli e viene calcolato intervallo per intervallo sottraendo dall'ultimo prezzo di chiusura il prezzo individuato negli X precedenti con la formula:

$$M = V - V^{X}$$

Possono essere utilizzati diversi archi temporali, quello maggiormente impiegato in grafici giornalieri è di 10 giorni, e più corto è il periodo più la linea risulta essere sensibile con maggiori oscillazioni.



[46]

Se il risultato del calcolo è positivo viene posizionato sopra la linea intermedia, in corrispondenza dello 0, mentre sotto se è negativo.

Esiste un'importante correlazione tra la linea di momentum e la direzione del trend dei prezzi:

- se il momentum si trova al di sopra dello 0 e il trend dei prezzi è rialzista→ il trend sta accelerando
- se il momentum inizia una fase discendente mentre l'uptrend è ancora in corso → il trend si sta indebolendo

[46] J. J. Muprhy; Analisi tecnica dei mercati finanziari; 2° edizione; pag. 176

- se scende al di sotto dello 0 e il prezzo di chiusura è inferiore a quello dei 10 giorni precedenti→ inizio downtrend di breve periodo, la cui forza aumenterà più il momentum continua a scendere
- l'oscillatore incrocia lo 0 al rialzo in ipervenduto→segnale di acquisto
- l'oscillatore incrocia lo 0 al ribasso in ipercomprato→ segnale di vendita

I prezzi possono continuare a muoversi nella direzione del trend anche se la linea di momentum inizia ad appiattirsi, questo significa solamente che la velocità si sta stabilizzando.

Un segnale di pericolo si ha quando si verifica una divergenza tra l'oscillatore e il trend dei prezzi, e la violazione della linea centrale dà informazioni sulla possibile tendenza del prezzo.

Solitamente per determinare i limiti superiori e inferiori della barra viene effettuata un'analisi visiva dello storico del mercato considerato e per questo motivo quest'ultime devono essere ridefinite periodicamente.

- **Rate of change (ROC):** ogni punto è calcolato con al formula $100(V/V^x)$, dove il rapporto tra l'ultima chiusura e il prezzo degli X intervalli precedenti viene moltiplicato per cento che diventa il punto che individua la linea intermedia.
- **Commodity Channel Index (CCI)**: il prezzo attuale è confrontato con una media mobile relativa a un certo intervallo temporale e il risultato viene normalizzato con un divisore basato sulla deviazione media. Grazie a questo tipo di costruzione viene individuato un intervallo costante, con dominio [-100,100], utilizzato dai trader come limite oltre al quale il mercato risulta ipercomprato o ipervenduto.
- **Relative Strength Index (RSI):** uno degli oscillatori più usati dai trader. È stato creato per ovviare al problema delle barre limitrofe non costanti e alle possibili distorsioni nei risultati che emergono dai calcoli di altri oscillatori quando improvvisi e bruschi cambiamenti nei valori possono causare delle false inversioni di linea. La formula utilizzata per i calcoli è:

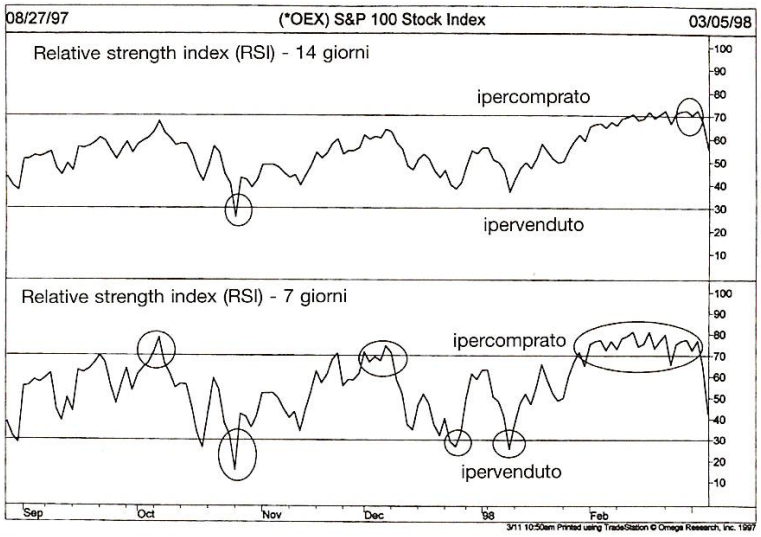
  $$RSI = 100 - \left[\frac{100}{1+RS}\right] \text{ con}$$

  RS= media delle chiusure al rialzo di x giorni / media delle chiusure al ribasso di x giorni

  media delle chiusure al rialzo di x giorni= Σ(Δchiusura al rialzo)/(n° di periodi)

  media delle chiusure al ribasso di x giorni= Σ(Δchiusura al ribasso)/(n° di periodi)

  come indicato precedentemente l'oscillatore risulta più sensibile su periodi minori, evidenziando maggiormente le oscillazioni, e viene utilizzato su 5-7-9-21-28 giorni (originariamente su un periodo di 14 giorni).



[47]

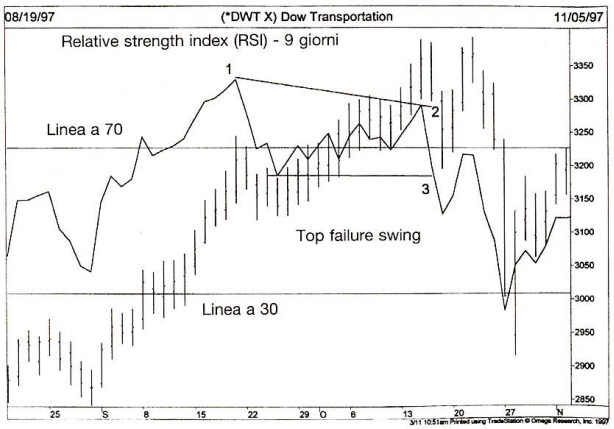
Per costruzione individua un range verticale costante che va da 0 a 100, dà indicazioni più rilevanti quando i valori si avvicinano all'estremità, il mercato si considera ipercomprato quando i movimenti sono al di sopra di 70-80(in bull market) e ipervenduto quando sono al di sotto di 30-20(in bear market).

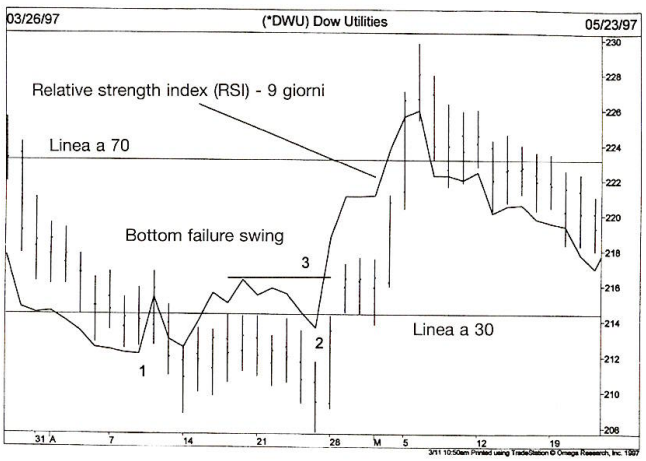
Si parla di

- Failure swings: quando l'RSI si trova sopra 70 o sotto 30. In questi casi le divergenze con il trend dei prezzi sono un segnale preoccupante e se l'oscillatore rompe un precedente massimo o minimo la formazione si considera confermata. Bisogna però prestare molta attenzione perché in caso di forte trend l'oscillatore, che anticipa i prezzi, produce segnali prematuri di ipercomprato o ipervenduto e per questo motivo si tende ad aspettare un secondo movimento in questa zona per una conferma

- Top failure swing: in uptrend quando un nuovo massimo di RSI, sopra 70, è incapace di superare il precedente massimo e successivamente avviene una rottura al ribasso di un minimo precedente

[48]





- Bottom failure swing: in downtrend, quando l'RSI, sotto i 30, finisce per superare un precedente massimo senza formare un nuovo minimo.

[49]

[47] J. J. Muprhy; Analisi tecnica dei mercati finanziari; 2° edizione; pag. 184
[48] J. J. Muprhy; Analisi tecnica dei mercati finanziari; 2° edizione; pag. 185
[49] J. J. Muprhy; Analisi tecnica dei mercati finanziari; 2° edizione; pag. 185

- **Processo dello stocastico**: basato sull'osservazione che in uptrend i prezzi di chiusura tendono ad avvicinarsi al massimo del price range di un determinato periodo di tempo e al contrario in downtrend tendono ad avvicinarsi al minimo. Allo scopo di determinare la relazione tra il prezzo di chiusura recente e il price range (più comune 14 giorni) vengono utilizzate due linee, la %K più veloce e %D più lenta:
  1. %K= $100\left[\frac{C-L14}{H14-L14}\right]$

     C=ultima chiusura

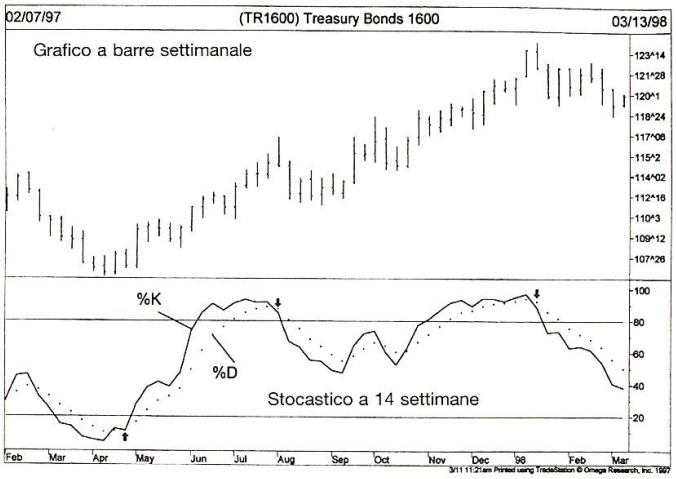
     L14= il minimo più basso degli ultimi 14 periodi

     H14=il massimo più alto degli stessi n periodi

     Un livello superiore a 80 indica una chiusura vicino al massimo del price range mentre sotto il 20 vicino al minimo.

  2. %D, la più importante tra le due e genera i segnali principali, è una media mobile a 3 periodi della linea %K. La divergenza di questa linea con il prezzo del mercato nelle zone di ipercomprato e ipervenduto rappresentano il segnale principale e si parla di divergenza:
     a. ribassita, quando sopra agli 80 si verificano 2 massimi decrescenti con prezzi al rialzo
     b. rialzista, quando sotto i 20 si formano due minimi crescenti con prezzi al ribasso

     la conferma di questi segnali che così possono essere usati per acquisti o vendite avviene quando questa linea incrocia la %K più veloce



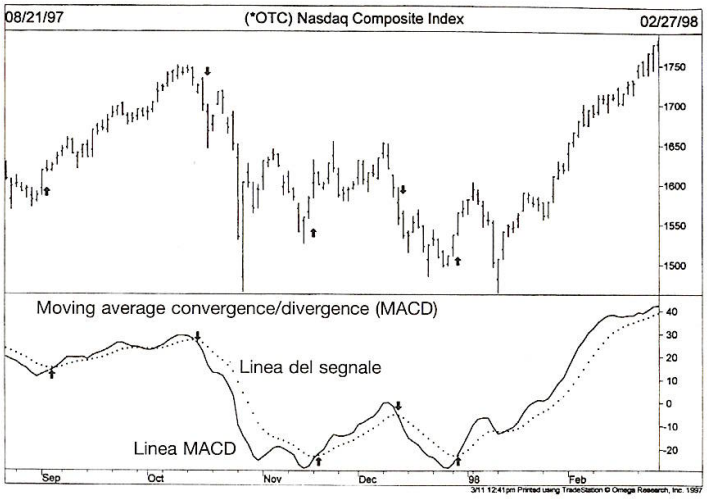
[50]

Ai fini dell'analisi per avere segnali più precisi è possibile combinare la stocastico settimanale, per determinare direzione del mercato, con quella giornaliera, per indicazioni sul timing, oppure con l'RSI.

- **%R, oscillatore Larry Williams:** il prezzo di chiusura giornaliero viene sottratto dal prezzo massimo del range di un determinato periodo, il tutto diviso per il range totale del periodo. La lunghezza dell'oscillatore può essere collegata ai cicli del mercato e solitamente si usa ½ di quest'ultimo, i più diffusi 5-10-20.

[50] J. J. Muprhy; Analisi tecnica dei mercati finanziari; 2° edizione; pag. 189

1.2.11.2. MACD: Moving average convergence/divergence



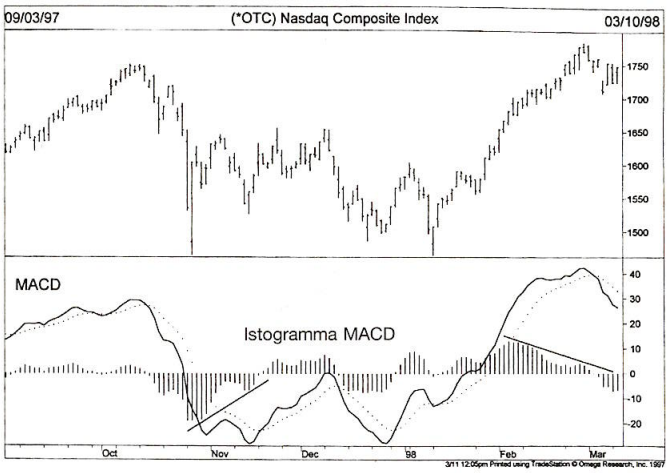
[51]

Indicatore che a livello grafico mostra 2 linee, anche se quelle usate per i calcoli sono 3. La più veloce è la linea MACD, è data dalla differenza di 2 medie mobili esponenziali dei prezzi di chiusura, solitamente gli ultimi 12 e 26 periodi. La seconda linea è chiamata linea di segnale è una media mobile esponenziale a 9 periodi della linea MACD.

Quando le linee sono troppo lontane dallo 0 si parla di ipervenduto o ipercomprato e i segnali più importanti di acquisto/vendita vengono dati quando le due curve si incrociano:

- MACD al di sopra della linea di segnale→ acquisto
- MACD al di sotto della linea di segnale→ vendita

Come negli oscillatori anche per le MACD vengono osservate le divergenze con il trend dei prezzi per determinare punti di ingresso o uscita dal mercato: ad esempio una divergenza negativa ribassista avviene quando le linee in ipercomprato iniziano a mostrare debolezza nel mentre che i prezzi continuano a salire, e questo solitamente indica un massimo di mercato. Al contrario una divergenza positiva rialzista quando le linee di MACD si trovano molto al di sotto dello zero, ipervenduto, e iniziando a risalire prima dei prezzi, questo spesso indica un bottom di mercato.



[52]

È anche possibile trasformare le due linee in un istogramma a barre verticali che mostrano i delta tra le linee delle medie mobili che sarà positivo se la linea MACD si trova sopra quella di segnale, negativo in caso contrario. Quando l'istogramma al di sopra dello zero inizia a scendere significa che l'uptrend si sta indebolendo e se quando si trova al di sotto inizia a risalire potrebbe indicare la fine del downtrend. Gli incroci dell'istogramma con la linea dello zero coincidono con i segnali di acquisto/vendita citati prima.

[51] J. J. Muprhy; Analisi tecnica dei mercati finanziari; 2° edizione; pag. 193
[52] J. J. Muprhy; Analisi tecnica dei mercati finanziari; 2° edizione; pag. 195

Così come gli oscillatori anche gli istogrammi sono utilizzati per individuare il timing corretto di uscita mentre sono meno indicati per l'apertura di nuove posizioni.

Come per il processo stocastico, su timeframe settimanali forniscono informazioni più importanti utili a determinare direzioni di trend mentre quelli giornalieri aiutano a perfezionare i punti di ingresso o uscita e quest'ultimo tipo di segnale viene ritenuto confermato solo se in linea con il trend settimanale.

1.2.12. Teoria delle onde di Elliott

Teoria fondata da Ralph Nelson Elliot, che lui stesso definiva complemento della teoria di DOW, con l'obiettivo di definire la sua personale visione del comportamento del mercato azionario.

La teoria ha 3 aspetti fondamentali:

- **Il pattern:** le figure delle onde e delle formazioni
- **La ratio**: la percentuale che risulta utile per determinare i punti di ritracciamento o gli obiettivi di prezzo
- **Il tempo:** usato per confermare le figure e le percentuali

1.2.12.1. Il pattern

Applicato inizialmente agli indici del Dow Jones sosteneva che il mercato azionario in uptrend seguisse un andamento ripetitivo di 5 onde rialziste, 3 al rialzo chiamate impulse waves e 2 al ribasso chiamate corrective waves, seguite da 3 ribassiste.

[53]

Le onde possono riferirsi a periodi di diversa lunghezza e ogni onda può essere suddivisa in onde di grado inferiore, 5 se l'onda suddivisa è nella stessa direzione del trend oppure 3 se si tratta di un'onda di correzione.

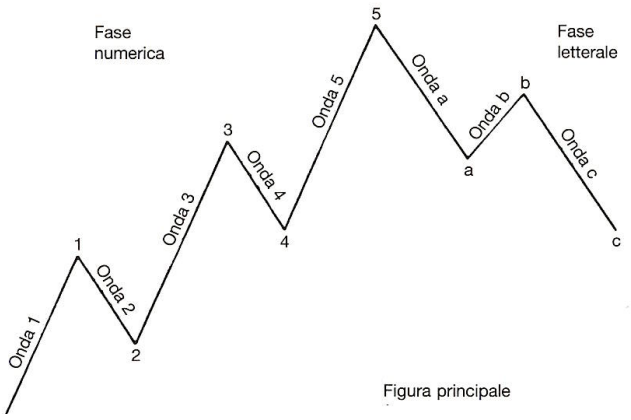


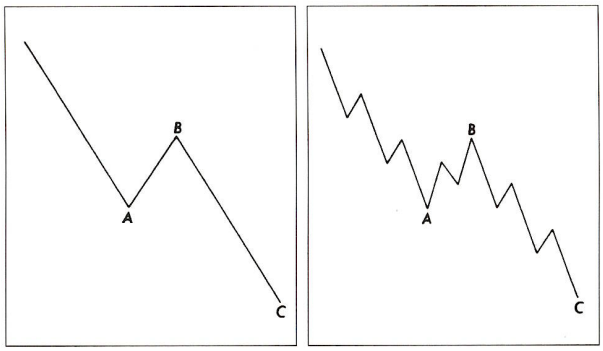
Figura principale

[53] J. J. Muprhy; Analisi tecnica dei mercati finanziari; 2° edizione; pag. 248

Le corrective waves sono classificate in:

- Zig-zags: figura a 3 onde in controtendenza al trend primario con una sequenza di 5-3-5

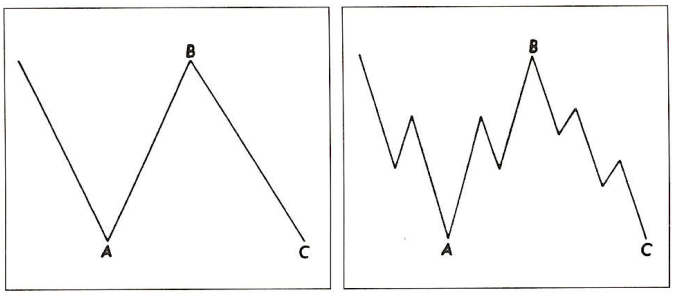
54



- Flats: come il precedente è una figura composta da 3 onde in controtendenza al trend primario ma con una sequenza 3-3-5.

55

In genere è considerato un segnale di forza in bull market e rappresenta più un consolidamento che una correzione.



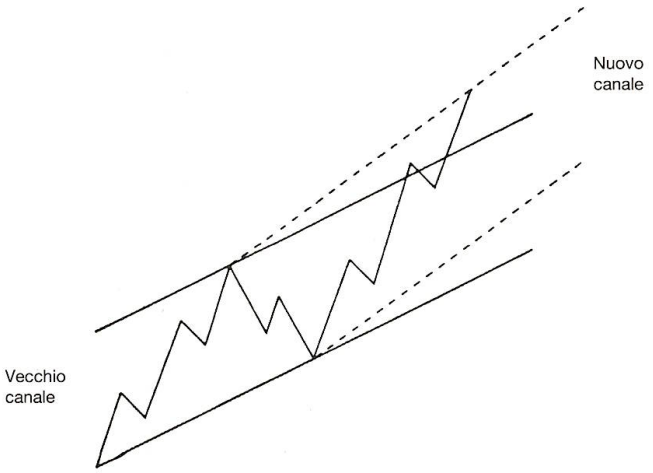
- I triangoli: si verificano solitamente sulla quarta onda e precedono il movimento finale nella direzione del trend primario. Le figure sono le stesse viste in precedenza ma Elliott le definisce di consolidamento la cui rottura avviene in 5 onde, ciascuna formata a sua volta da 3 onde.

1.2.12.1.1. Price channeling

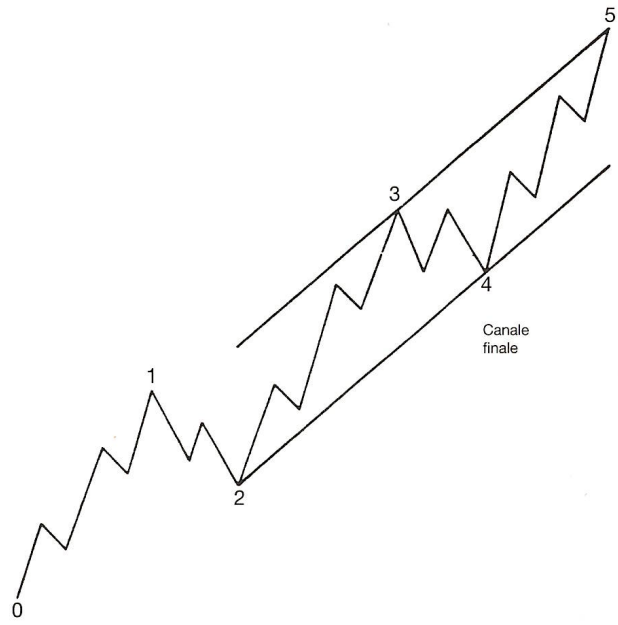
Come si evince dal nome consiste nella creazione di canali con il fine di calcolare degli obiettivi di prezzo.

Si costruisce un trend channel come in figura, tracciando la trendlines rialzista lungo i bottoms delle prime due onde e una parallela passante per il top:

56



[54] J. J. Muprhy; Analisi tecnica dei mercati finanziari; 2° edizione; pag. 252
[55] J. J. Muprhy; Analisi tecnica dei mercati finanziari; 2° edizione; pag. 254
[56] J. J. Muprhy; Analisi tecnica dei mercati finanziari; 2° edizione; pag. 259-260



Quando la terza onda accelera aumentando la pendenza e supera il precedente top è necessario ridisegnare il canale congiungendo i nuovi bottom e top.

Una volta che le 5 onde rialziste sono state completate e inizia il bear market solitamente i prezzi non scendono sotto il livello della quarta onda che diventa quindi un supporto per le correzioni.

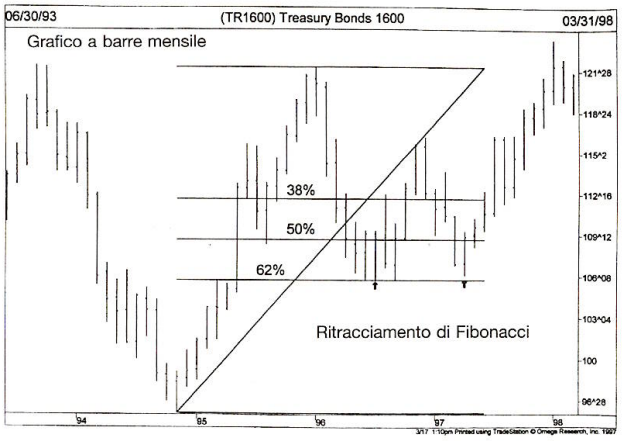
1.2.12.2. Ratio: la percentuale di Fibonacci e il ritracciamento

La teoria delle onde si basa sui numeri della sequenza di Fibonacci:1,1,2,3,5,8,13,21,34,55,89…

Questo è già visibile nel numero delle onde di cui si è discusso nel paragrafo precedente riferito al pattern: 8 onde per un ciclo completo di cui 5 principali e 3 di correzione.

E inoltre molto utilizzata nel calcolo di obiettivi di prezzo secondo le seguenti formule:

- Target minimo di top della 3° onda: (lunghezza prima onda*1,618) + Bottom 2° onda
- Target top 5° onda
  - minimo: (lunghezza prima onda*2*1,618) + bottom 1° onda
  - massimo: (lunghezza prima onda*2*1,618) + top 1° onda
  - se 1° e 3° onda uguali: [(top 3°-bottom1°) * 1,618] + bottom 4°
- Lunghezza 3° onda di correzione:
  - uguale alla 1° onda di correzione se classica figura a zig-zag 5-3-5
  - (0,618*1° onda di correzione) - bottom 1° onda di correzione
  - Figura zig-zag 3-3-5, dove 2° onda raggiunge/supera top 1° di correzione 1,618*lung 1°)



Questi calcoli possono risultare poco pratici e per questo motivo viene usato il metodo delle percentuali di ritracciamento. I valori più utilizzati sono 38%, in un trend forte, 50% e 61,8%(arrotondato a 62), in un trend debole.

[57]

[57] J. J. Muprhy; Analisi tecnica dei mercati finanziari; 2° edizione; pag. 264

I numeri di Fibonacci sono stati utilizzati anche per la definizione di obiettivi temporali: a partire da un significativo punto di inversione vengono contati i tops e bottoms importanti nell'unità di tempo considerata con la "speranza" che questi avvengano in un'unità che fa parte della serie di Fibonacci (ad esempio 13° giorno, 21° giorno ….). Per questo tipo di approccio l'aspetto temporale è considerato il meno importante della teoria di Elliott.

Nei mercati azionari le figure di Elliot tendono a essere chiare, solitamente la terza onda risulta la più estesa e una regola quasi ferrea è che la quarta onda non supera mai la prima nel ritracciamento. La teoria delle onde di Elliot risulta più efficace se applicata agli indici del mercato azionario.

1.2.13. I Cicli temporali

Il fattore tempo riveste un ruolo molto importante nell'analisi tecnica in quanto:

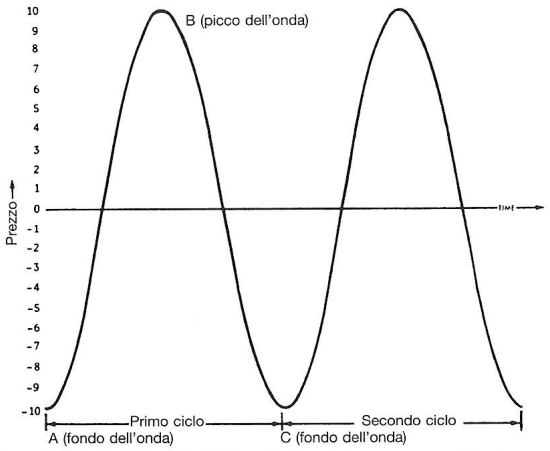
- viene utilizzato per cercare di stimare quando i prezzi calcolati con i metodi indicati precedentemente raggiungeranno tali obiettivi
- più una formazione si protrae nel tempo, più questa avrà delle potenzialità nei suoi successivi movimenti
- più a lungo una trendline di supporto/resistenza è valida, più sarà importante
- caratteristica principale delle medie mobili
- usati per obiettivi temporali di Fibonacci

Per valutare correttamente anche questa caratteristica si è introdotta l'analisi dei cicli, che aggiunta alle medie mobili, ai picchi di prezzo e alle trendlines permette di avere una visione più chiara della situazione.

Prima di capire come utilizzare i cicli nell'analisi tecnica è necessario fare una piccola introduzione.

I Cicli

Gli autori dell'analisi dei cicli temporali studiando il passato sono stati in grado di definire i cicli per quasi tutti gli aspetti della nostra vita (dalle macchie solari ai conflitti mondiali, etc…). Si è notato che cose molto diverse tendono ad avere cicli che terminano negli stessi momenti e che se sono simili tendono anche ad avere gli stessi ritmi. Si è così iniziato ad applicare questo tipo di analisi anche ai mercati finanziari.



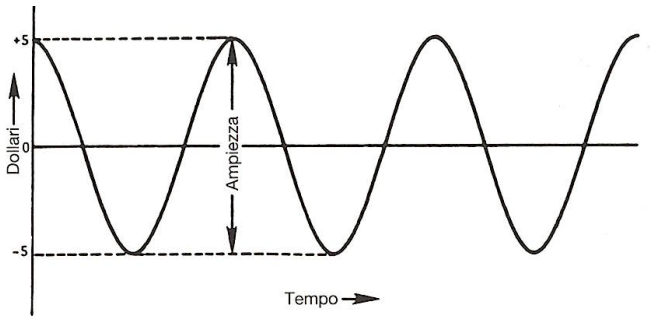
I cicli sono rappresentati da onde in cui i massimi vengono chiamati picchi e i minimi fondi. Nell'analisi delle onde la durata del ciclo è misurata tra i fondi.

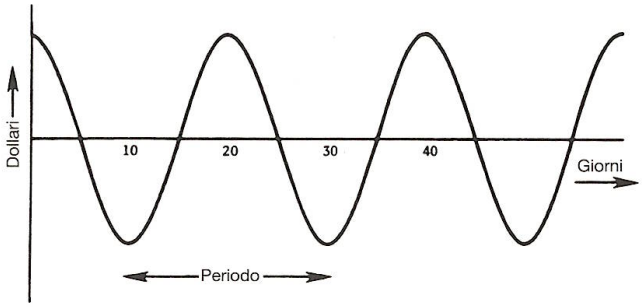
58

[58] J. J. Muprhy; Analisi tecnica dei mercati finanziari; 2° edizione; pag. 274-275

I cicli sono contraddistinti da 3 caratteristiche principali:

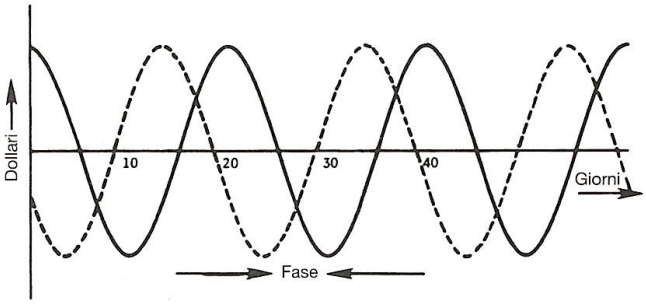
- l'ampiezza: misura dell'altezza dell'onda e nell'analisi dei mercati è espressa nella valuta utilizzata, centesimi o punti





- il periodo: tempo che in intercorre tra due minimi

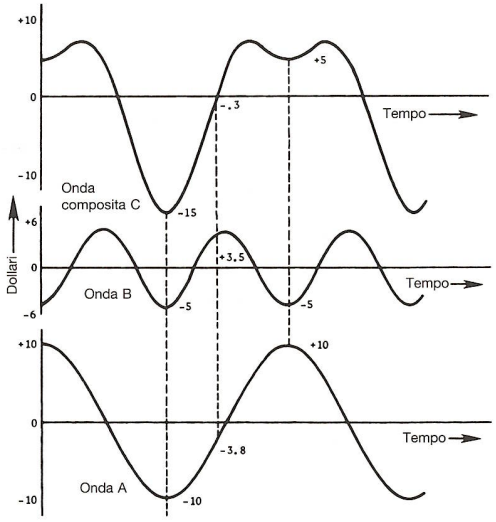
- la fase: misura del tempo in cui si è formato un minimo osservando la differenza tra onde. Permette di studiare le relazioni tra le estensioni di diversi cicli, quello effettivo e quello stimato. Questo consente di determinare una stima per la data di successivi punti di minimo in base a quelli effettivi realizzatisi in precedenza considerando anche il periodo e l'ampiezza.



Se un ciclo resta costante può essere usato per stimare successivi punti di massimo o minimo

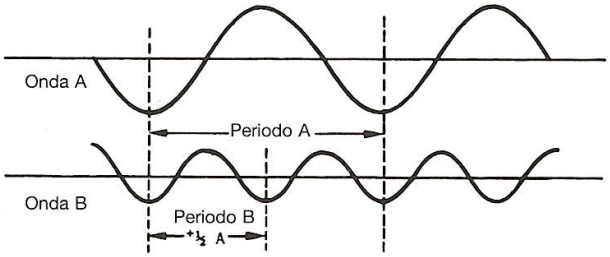
I principi dei cicli

- principio della somma: i movimenti dei prezzi sono una somma di almeno 2 o 3 cicli attivi, ciascuno dei quali può essere misurato per determinare degli obiettivi di prezzo. Questo principio aiuta a spiegare i canali come somma di un'onda flat e una linea rialzista



[59]

[59] J. J. Muprhy; Analisi tecnica dei mercati finanziari; 2° edizione; pag. 276



a. Esempio di armonia e sincronicità

- principio dell'armonia: onde vicine sono tra loro in rapporto ben definito, solitamente 2. Se si considera ad esempio un ciclo di 20 giorni, l'onda minore sarà di 10 e quella maggiore di 40giorni.
[60]

- principio della sincronicità: le onde di lunghezze differenti tendono a creare minimi nello stesso momento. Corrispondenza molto importante perché indica che mercati differenti possono avere inversioni nello stesso istante.

- principio della proporzionalità: relazione tra periodo e ampiezza, periodi più lunghi corrispondono a ampiezze maggiori in modo proporzionale

- principio della variazione: i primi 3 principi elencati rappresentano delle tendenze ma non regole fisse, sono sempre presenti delle variabili non considerate

- principio della nominatività: le lunghezze dei cicli nominali teorici, indicati nella tabella, possono essere usate come base di partenza per differenti mercati.
[61]

Anni	Mesi	Settimane	Giorni
18			
9			
	54		
	18		
		40	
		20	
			80
			40
			20
			10
			5

[60] J. J. Muprhy; Analisi tecnica dei mercati finanziari; 2° edizione; pag. 277
[61] J. J. Muprhy; Analisi tecnica dei mercati finanziari; 2° edizione; pag. 279

1.2.13.1. Cicli temporali nell'analisi tecnica

Sono presenti molti cicli nei mercati ma solo quelli dominanti sono facilmente identificabili, hanno un valore previsionale e sono in grado di influenzare in modo importante i prezzi. Come per il trend anche per i cicli è bene partire da quelli più lunghi per poi spostarsi in quelli di breve periodo utili per determinare i corretti timing di entrata o uscita dal mercato.

Le tipologie di cicli sono:

- Di lungo periodo (>2 anni): permettono di capire la direzione del trend primario
- Stagionale (1 anno o meno): periodi ben definiti intra-anno in cui si verificano delle precise formazioni con massimi e minimi stagionali. Ogni mercato è influenzato da cicli stagionali che permettono di individuare delle specifiche tendenze. Ad esempio nel mercato del grano i minimi si verificano vicino al periodo del raccolto per poi aumentare con l'incremento della domanda. Nel mercato azionario invece solitamente i mesi forti sono tra Novembre-Gennaio, Marzo-Aprile e Luglio mentre il mese più debole tende ad essere Settembre
- Primario/intermedio (da 9 a 26 settimane): i cicli intermedi rappresentano il trend secondario e aiutano a determinare la direzione del mercato per i segnali che si ricavano dal ciclo operativo
- Operativo (4 settimane): risulta il più utile ai fini del trading, grazie al quale vengono definiti i timing di ingresso e uscita

Ad esempio se il trend primario è in rialzo, i minimi fatti dal ciclo operativo vengono usati per ingresso mentre se è al ribasso i massimi vengono usati per uscite o short. Anche per i cicli quindi è di fondamentale importanza partire dall'analisi del trend principale e solo dopo si possono interpretare correttamente i segnali dei cicli di più breve periodo.

Un importante concetto dell'analisi dei cicli è quello della traslazione: mentre per la lunghezza dei cicli si fa riferimento ai minimi in questo caso vengono considerati i massimi che dovrebbero cadere nel mezzo dell'onda. Si è osservato infatti che in un mercato di lungo periodo rialzista i massimi tendono a spostarsi a sinistra del valore centrale dell'onda mentre se il trend è ribassista a destra.

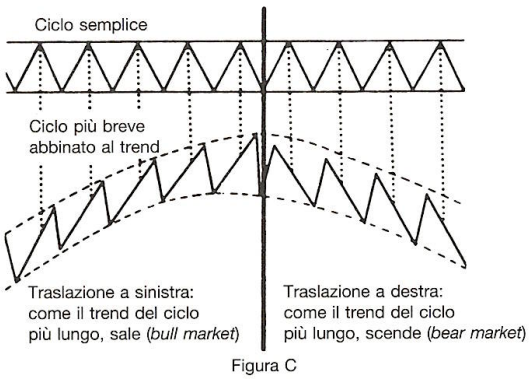


Figura C

Quindi in un mercato rialzista, i prezzi tendono a rimanere di più nella fase di rialzo. Se invece il mercato è ribassista i prezzi rimangono di più nella fase al ribasso.

[62]

Per analizzare l'influenza del singolo ciclo sui mercati è necessario isolare quest'ultimo visivamente o con il supporto di appositi sistemi informatici. Storicamente tra i cicli più noti si è visto che sul mercato azionario hanno influenza:

- L'andamento dei primi giorni dell'anno e del mese di Gennaio

[62] J. J. Muprhy; Analisi tecnica dei mercati finanziari; 2° edizione; pag. 286

- Il ciclo quadriennale presidenziale (nel mercato americano): 1° e 4° anno forte, 2° e 3° debole

Come anticipato i cicli risultano molto utili se abbinati ad altri indicatori come le medie mobili o gli oscillatori. Ad esempio se si individua che il ciclo operativo ha una lunghezza di 20 giorni l'oscillatore viene determinato su 10 giorni e possono venir create due medie mobili di 20 e 10 per studiarne eventuali segnali di incrocio.

I cicli però non hanno una lunghezza statica e per cercare di individuarla caso per caso, è stato definito un approccio statistico chiamato MESA (maximum entropy spectral analisys): permette di apportare aggiustamenti agli indicatori tecnici, come le medie mobili o gli oscillatori, e determinare la tipologia di mercato in cui ci si trova (se trending è meglio un indicatore in grado di seguire il trend, come le medie mobili, mentre se il mercato è ciclico meglio indicatori come gli oscillatori).

1.3. Tattiche operative

I software di analisi tecnica possono essere utilizzati per qualsiasi mercato se si riesce a reperire i dati su cui operare. Ogni sistema di trading deve partire da 3 fattori:

- la previsione del prezzo: primo passo essenziale che dice all'operatore cosa fare, se comprare o vendere, in base alla sua previsione della direzione del mercato. Se questa è sbagliata nessuno degli aspetti che ne seguono potrà funzionare
- timing: determinato in modo puramente tecnico, aiuta a stabilire i momenti di entrata e uscita
- gestione patrimoniale: riguarda l'allocazione del capitale e come deve essere suddiviso tra mercati/azioni per impostare le operazioni

1.3.1. Money management

È uno degli aspetti fondamentali che vanno affiancati all'analisi puramente tecnica. È infatti importante definire con precisione quanto e come ci si può esporre in un'operazione perché, per quanto accuratamente analizzata, possono verificarsi degli imprevisti e conseguenti perdite. Un'accurata gestione patrimoniale aumenta le possibilità di essere vincenti sul lungo periodo.

Per iniziare vengono suggerite delle linee guida di base che possono essere poi perfezionate con l'uso di accurate statistiche:

- Il totale investito non dovrebbe superare il 50% del capitale a disposizione. Questo in modo da poter sfruttare determinate occasioni o mediare al ribasso eventuali posizioni aperte erroneamente
- Ogni singola operazione non dovrebbe superare il 10-15% del capitale
- La cifra che si intende investire in un'operazione rischiosa non deve superare il 5%, o comunque deve essere una cifra che il trader è disposto a perdere
- La cifra investita in un determinato settore di mercato non deve superare il 20/25% del capitale in quanto i titoli simili tendono a muoversi in modo uniforme

Alla base di queste linee guida vi è il concetto di diversificazione per limitare l'esposizione al rischio.

Alcune metodologie di supporto alla gestione patrimoniale sono:

- **gli stop di protezione:** consiste nel definire degli ordini di vendita/short o di acquisto/long a determinati livelli in modo da proteggere le proprie posizioni limitando le perdite in caso i movimenti di mercato smentiscano le nostre previsioni. La bravura dell'operatore si vede proprio nella definizione di questi livelli che vengono stabiliti combinando diversi fattori tecnici in modo che siano più le volte che limiti le perdite da quelle che chiudono erroneamente delle posizioni a causa di falsi breakout. Maggiore è la volatilità più gli stop devono essere lontani dai livelli individuati, per evitare i falsi segnali, ma abbastanza vicini da limitare le perdite. È importante che questi siano piazzati sopra un'importante resistenza per le posizioni short e sotto un supporto per quelle long.
- **percentuale di gestione del rischio:** essendo che non si possono chiudere tutte le operazioni in profitto, anzi più della metà spesso risultano in perdita, l'obiettivo di un trader è che l'ammontare incassato dalle operazioni vincenti superi l'ammontare di quelle perdenti.
  Per ogni operazione si calcolano un obiettivo di profitto e le potenziali perdite con un rapporto che solitamente è di 3:1 in modo da cercare di tagliare quasi subito le posizioni in perdita e lasciar correre le posizioni in profitto in assenza di segnali particolari (es la divergenza tra la direzione del trend e gli oscillatori).
  Un altro metodo è suddividere l'operazione, o aprire posizioni multiple, definite di trading e di trending. La parte trading è quella di breve periodo e pone obiettivi più facilmente raggiungibili (es resistenza in ipercomprato) che permettono di fare rapidi e piccoli profitti, lasciando la possibilità all'utente di riaprire la posizione se il trend riprende. Le operazioni di trending invece sono long e vengono mantenute sul lungo periodo fino al raggiungere degli obiettivi o gli stop originali.

1.3.2. Timing

La decisione finale di come e dove entrare nel mercato è l'ultima a livello operativo e deve essere presa con il supporto dei principali metodi utilizzati dall'analisi tecnica:

- **Tattiche sulle rotture**: in questi casi il trader può scegliere se
  - anticipare la rottura→più alto il possibile guadagno ma anche una maggior probabilità di errore
  - agire su di essa→ più probabilità di essere dalla parte giusta ma l'operazione avrà un costo maggiore
  - agire dopo la rottura→ aspettando una correzione o una reazione che però non sempre avvengono

  Per coprire tutte le casistiche e diversificare limitando il rischio si può agire aprendo posizioni diverse su ogni momento.

- **Rottura delle trendlines**: comprare sulla rottura al rialzo di una trendline di lungo periodo o vendere se la rottura avviene al ribasso.
- **Uso di supporti e resistenze**: come per le trendlines la rottura di una resistenza è un importante segnale di acquisto mentre quando avviene su un supporto è di vendita. I supporti risultano importanti per la definizione degli stop che possono essere inseriti poco sotto il livello individuato dal supporto. La stessa cosa vale per le resistenze quando sono rotte al rialzo e si trasformano in supporti. In downtrend il rimbalzo verso la resistenza o in uptrend il ritracciamento verso il supporto rappresentano dei momenti ideali per l'apertura di posizioni.
- **Uso del ritracciamento percentuale:** come visto nelle precedenti trattazioni determinate percentuali di ritracciamento o rimbalzo, 40-60% (o i numeri individuati con le percentuali di Fibonacci), rappresentano degli ottimi punti di ingresso o di vendita.

- **Uso dei gaps:** essendo che i gaps rappresentano dei veri e propri livelli di supporto o di resistenza si possono usare le stesse logiche definite sopra, acquistando a livello del gap o poco sopra e inserendo degli stop poco sotto.

Il modo migliore per scegliere i punti di entrata e uscita più corretti è di combinare tutti questi fattori tra loro, considerando i vincoli del money management, e per un migliore timing tutti questi aspetti vengono applicati su grafici di breve periodo, giornalieri o intra-day.

1.3.3. Tipologie di ordini

Un trader può eseguire diversi tipi di operazioni:

- **Al mercato:** ordine di acquisto o vendita al prezzo di mercato corrente. Utilizzati in mercati veloci, garantiscono la posizione ma lasciano l'operatore in balia del mercato.
- **Limit order:** si definisce un prezzo limite. Se si intende acquistare è al di sotto dell'attuale, è indica il prezzo più alto che si intende pagare. Se si vuole vendere si trova al di sopra e indica il prezzo più basso che si accetta. Il rischio è che il limite venga raggiunto ma che il prezzo rimbalzi/ritracci così velocemente che l'ordine non viene eseguito.
- **Stop order:** specifica un prezzo al quale l'ordine deve essere eseguito. Si parla di buy stop (per sfruttare la rottura) o sell stop (per limitare le perdite) a seconda che si trovi sopra o sotto il prezzo di mercato. Una volta che il livello di stop viene raggiunto l'ordine stop diventa un ordine al mercato e viene eseguito alla miglior condizione possibile.
- **Stop limit:** combinazione dei precedenti in cui viene indicato sia il momento di attivazione dell'ordine con lo stop, sia la condizione limite che deve essere soddisfatta. Rischiano di non essere eseguiti in caso di gap dei prezzi.
- **Market if touched**: simile al limit order che però diventa effettivo solo se il limite viene raggiunto eseguendo l'ordine alla miglior condizione possibile dopo l'evento.

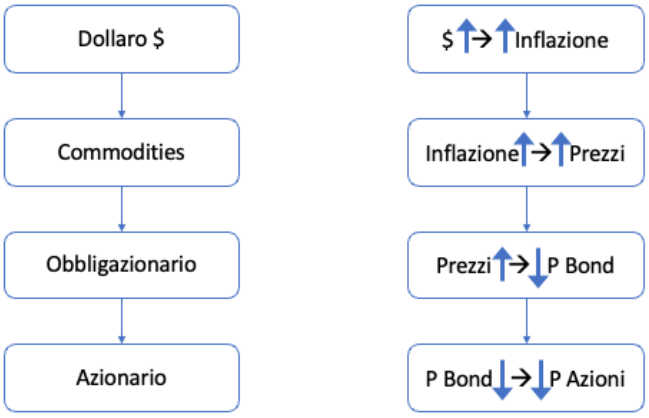
1.3.4. Legami tra mercati

Nel corso degli anni si è osservato come diversi mercati risultano legati tra loro, per comprendere meglio cosa succede in quello azionario è utile capire gli altri mercati e i diversi legami che li legano.

- **Mercato azionario e mercato dei futures:** intorno agli anni 70 con l'introduzione dei futures sulle valute, sui treasury bond e sugli indici, con i futures si iniziò ad operare anche sul mercato azionario e obbligazionario. Questo ha fatto in modo che i movimenti del mercato dei futures iniziassero a condizionare il mercato azionario e non solo quello delle commodities, anticipando addirittura tramite appositi segnali l'inflazione e le politiche monetarie. Il legame appare evidente quando si osserva la relazione tra S&P500 cash e S&P500 futures. Questi futures trattano un premio dall'indice cash determinato dai suoi tassi d'interesse a breve. Quando il premio supera il fair value, premio intrinseco calcolato giornalmente, di una percentuale si attivano delle operazioni di arbitraggio con programmi di trading in cui vengono venduti i features e si comprano le azioni dell'indice, il che aiuta ulteriormente la salita dell'indice.
- **Mercato azionario e obbligazionario:** il mercato azionario è influenzato dai tassi d'interesse che possono essere osservati indirettamente dal mercato obbligazionario, in quanto i bonds si muovono in maniera opposta a questi ultimi. Quando scende il prezzo del bond, cioè sale lo yield, è un possibile segnale che il mercato azionario possa anch'esso scendere. I due mercati

sono quindi positivamente correlati (questo rapporto potrebbe invertirsi quando ci si trova in deflazione).

- **Bond e commodities:** i prezzi dei bond sono influenzati dalle aspettative di inflazione spesso anticipate dai prezzi delle commodities che quindi viaggeranno in direzione opposta ai prezzi dei bond. I due mercati sono negativamente correlati.
- **Commodities e dollaro:** l'aumento del dollaro è considerato un campanello di allarme inflazionistico, con conseguenze negative sui prezzi delle commodities. Mercati positivamente correlati. Il trend del dollaro consente una valutazione sulla forza dell'economia americana, questo permette a sua volta una quantificazione degli utili delle società e di conseguenza la loro valutazione in borsa.

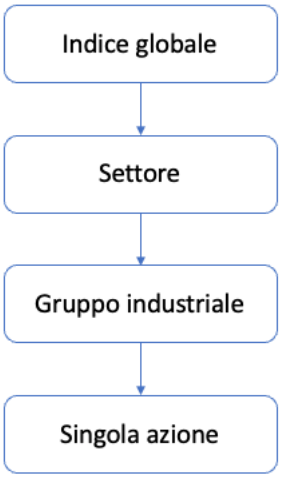


Questi concetti danno una visione globale del sistema economico e i software permettono di identificare il grado di correlazione tra diversi mercati.

Quello azionario può essere ancora suddiviso in settori, a loro volta frammentati in gruppi industriali, che risentono di ciò che avviene a negli altri mercati ad esempio:

- **mercati obbligazionari forti e merci deboli**: i gruppi azionari sensibili ai tassi d'interesse (utilities, finanziari e beni produttori di prime necessità) hanno performance migliori dei restanti e quelli sensibili all'inflazione ne soffrono
- **mercati obbligazionari deboli e merci forti**: avviene il contrario, i gruppi azionari sensibili ai tassi d'interesse ne soffrono mentre quelli sensibili all'inflazione ne beneficiano
- **dollaro e titoli molto capitalizzati:** un dollaro forte può avere conseguenze negative sui titoli di grandi multinazionali in quanto vedrebbero penalizzate le loro esportazioni a causa dell'aumento dei prezzi mentre può aiutare le piccole compagnie che operano a livello nazionale.

Per osservare questi legami e decidere a livello operativo viene fatta un'analisi della forza relativa in cui sono messi a rapporto due prezzi, ad esempio un indice delle commodities e i futures sui bonds. Se il rapporto è crescente un operatore azionario dovrebbe comprare titoli sensibili all'inflazione e vendere quelli collegati ai tassi d'interesse.



Per quanto riguarda invece i settori può essere utile confrontarli graficamente con un indice generale come lo S&P500 in modo da individuare quali stanno performando meglio (miglior forza relativa) e tramite l'analisi tecnica individuare quelli su cui stanno avvenendo cambiamenti importanti come inversioni di tendenza, cercando poi di investire in singoli titoli con forza relativa maggiore rispetto al settore.

Questi legami tra mercati, settori, gruppi industriali, indici e azioni possono essere studiati più facilmente con dei software che utilizzano l'intelligenza artificiale e le reti neurali, spiegate più avanti.

1.3.5. Analisi del respiro del mercato

Tramite i passaggi indicati precedentemente gli utenti possono arrivare ad individuare le singole azioni su cui operare con l'analisi tecnica. Esistono però ulteriori indicatori per l'analisi dei mercati che ne determinano lo stato individuandone il cosiddetto "respiro".

La misura del respiro del mercato è possibile da un'analisi di confronto tra:

- **il numero dei titoli al rialzo(advance) rispetto a quelli al ribasso(decline):** ogni giorno vengono conteggiati il numero dei titoli che salgono, che scendono e che rimangono invariati. La loro sommatoria (o il loro rapporto percentuale) viene inserita su un grafico giornaliero e tutti i risultati ottenuti vengono uniti da una line chiamata Advance-Decline(AD). È l'indicatore che rappresenta meglio la salute del mercato e viene poi confrontato con l'andamento dell'indice di mercato: se hanno la stessa direzione è visto positivamente mentre un'eventuale divergenza è una possibile indicazione di cattivo respiro del mercato. Questa linea risulta essere molto utile a livello previsionale in quanto solitamente raggiunge il suo massimo in anticipo rispetto agli indici di mercato. Le indicazioni dell'AD giornaliera valgono per il breve termine e vengono considerate valide anche per medi/lunghi periodi quando sono confermate dall'AD settimanale.
  Spesso vengono anche applicati degli oscillatori alla linea AD per determinare casi di ipercomprato/ipervenduto e individuare gli estremi di mercato. Uno dei più utilizzati è quello di McClellan: ricavato dalla differenza tra la media mobile esponenziale a 19 giorni e quella a 39. L'oscillatore considera un range da -100 a 100, quando il valore supera gli estremi si è in ipervenduto o ipercomprato e l'attraversamento del valore 0 centrale indica un possibile segnale rialzista, verso l'alto, o ribassista, verso il basso
- **nuovi massimi rispetto a nuovi minimi:** viene effettuato un conteggio delle azioni che hanno raggiunto nuovi massimi e nuovi minimi nella giornata di contrattazione rispetto ai valori dell'ultimo anno. I valori vengono poi analizzati in 2 modi:
  - riportati tramite delle medie mobili, solitamente a 10 giorni, separate su un grafico: se la linea dei massimi sale e quella dei minimi diminuisce è indice di un mercato rialzista, se quest'ultima incrocia al rialzo quella dei massimi indica un mercato ribassista.
  - calcolata la differenza si crea una linea simile a quella AD che può essere confrontata con i principali indici di mercato, confermandoli (stessa direzionalità), smentendoli (in caso di divergenza) o anticipandoli (eventuali inversioni). Normalizzando i valori della linea e riportandoli su un grafico con valore centrale 0 restano di più facile lettura le indicazioni di questo indice
- **il volume al rialzo rispetto al volume al ribasso:** come per i massimi e minimi il confronto tra i volumi di una giornata di operazioni può avvenire tramite linee separate o per differenza, e in entrambi i casi se i volumi rialzisti superano quelli ribassisti è un segnale positivo del mercato mentre negativo in caso contrario

Confrontando i rapporti dei valori AD e quelli relativi ai volumi è stato creato l'indice di Arms o TRIN con lo scopo di individuare se esiste un maggior volume sui titoli al rialzo o al ribasso:

$$\text{Arms index}=\frac{\frac{Adavancing}{declining}}{\frac{VolUp}{VolDown}}$$

Segnale positivo se <1, volume in crescita delle azioni rialziste

Segnale negativo se >1, volume in crescita azioni ribassiste

Per ottenere un indice meno sensibile alla variazione del singolo periodo, solitamente viene usata una versione "Open" dove si calcola una media mobile di quest'ultimo (estrapolata dalle medie mobili dei 4 componenti che compongono il prezzo e cioè apertura, chiusura, minimo e massimo) a 10/21 periodi e si considera ipervenduto se il risultato è >1,2 oppure ipercomprato se <0,7.

Quest'ultimo può essere combinato con la versione intra-day dell'indice AD, chiamato TICK, e in questo caso se il TICK è al rialzo con il TRIN<1 è un segnale positivo.

Il respiro del mercato può essere misurato confrontando anche gli indici principali del mercato stesso che hanno caratteristiche differenti ad esempio:

- Dow Jones e S&P500: contengono un numero ristretto di tioli ad alta capitalizzazione
- NYSE (New York Stock Exchange): contiene tutti i titoli della borsa di NY
- NASDAQ: con 5000 titoli è l'indice più diversificato ma dopo l'esplosione di capitalizzazione delle aziende tech il valore è guidato da poche aziende
- Russel 2000: contiene 2000 titoli di minor capitalizzazione

Per avere una conferma della direzione del mercato tutti gli indici dovrebbero tendenzialmente allinearsi al Dow Jones e lo S&P500. Se i titoli minori risultassero eccessivamente deboli rispetto a questi due indici potrebbe essere un segnale di indebolimento del mercato.

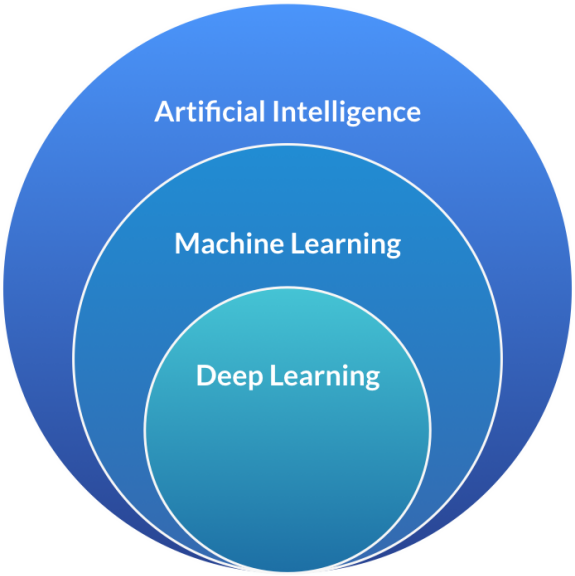
Perché segnali di rottura vengano considerati rilevanti occorre che i primi 3 subiscano rotture simili quasi contemporaneamente.

2. Intelligenza artificiale, Machine Learning e Deep learning

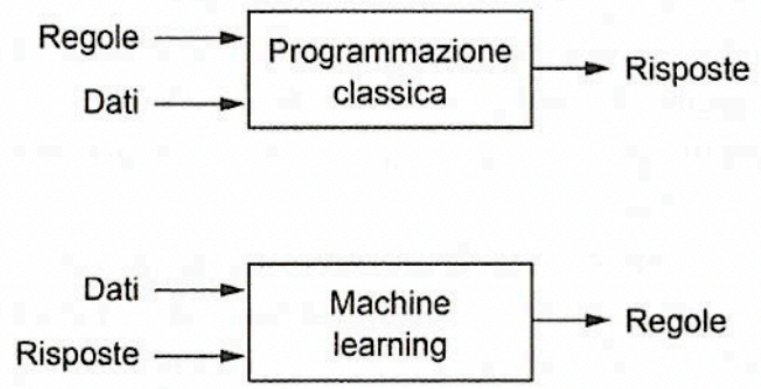
L'intelligenza artificiale è nata intorno agli anni 50 quando un gruppo di studiosi della neonata *computer science* ha iniziato a domandarsi se i computer potessero "pensare" nel *tentativo di automatizzare i compiti intellettivi svolti dagli esseri umani*.[63]

Intelligenza artificiale è il nome generale che si dà a diversi approcci con l'obiettivo di automatizzare i processi cognitivi umani e può andare da elementi semplici come un foglio di lavoro Excel a altri avanzati, come i robot umanoidi che camminano e parlano.

Il machine learning è un sottoinsieme dell'IA e deep learning è uno dei rami del machine learning.



I primi programmi di scacchi agivano solo in base a regole inserite nel codice dagli sviluppatori, non contemplavano il machine learning, in quanto si riteneva che il modo migliore per avvicinare il cervello umano era coprire il maggior numero di casistiche esplicitando le rispettive regole a codice. Questa visione, chiamata *intelligenza artificiale simbolica*, dominante dagli anni 50 a 80 circa, risultava però inefficace nella risoluzione di problemi più complessi, come la classificazione di immagini e il riconoscimento vocale. Nacque così il nuovo approccio chiamato ***machine learning*** con lo scopo di dotare le macchine di capacità di apprendimento e originalità.



[64]

[63] Francois Chollet; Deep Learning con Python; 2020; pag. 4
[64] Francois Chollet; Deep Learning con Python; 2020; pag. 5

Con il machine learning si passa quindi dalla pura e semplice programmazione, dove lo sviluppatore indica le regole, a un vero e proprio addestramento in cui vengono sottoposti degli esempi(dati e risultati) alle macchine per svolgere un determinato compito, e quest'ultime dopo aver rilevato delle strutture statistiche producono regole per automatizzarlo.

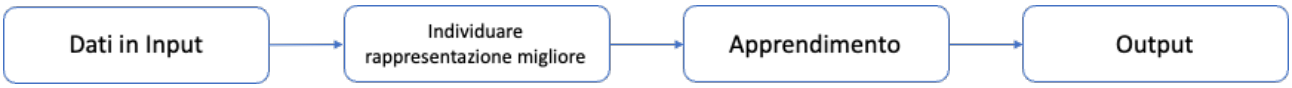
Il machine learning si è diffuso notevolmente a partire dagli anni 90 grazie all'introduzione di hardware sempre più prestanti che hanno consentito di analizzare più facilmente le enormi moli di dati necessari, perché sebbene sia un campo molto legato alla statistica si tratta di un procedimento più tecnico che matematico, che basa i suoi calcoli sulle nozioni teoriche ma produce risultati e idee in modo empirico.

Per sviluppare un modello di machine learning abbiamo quindi bisogno di 3 cose:

- Dati in input(punti)
- Esempi di output attesi
- Un modo per valutare se l'algoritmo sta lavorando correttamente(feedback)

Il feedback viene usato per regolare il funzionamento dell'algoritmo e rappresenta il core dell'apprendimento.

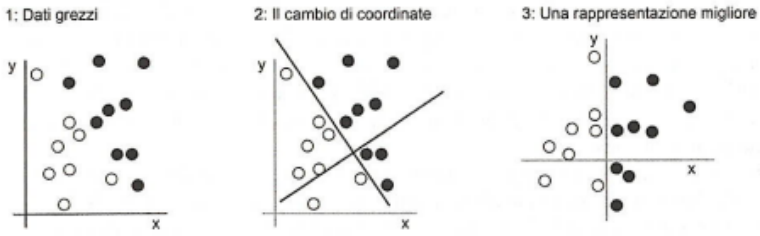
L'obiettivo principale del machine learning è quello di imparare a trasformare i dati di input in output, il più vicino possibili a quelli attesi. Questo avviene individuando le "rappresentazioni" utili tra i dati di input, cioè determinando quale sia il modo migliore di codificare, rappresentare e interpretare i dati per il nostro scopo: gli stessi dati rappresentati in modo diverso possono semplificare o complicare notevolmente un compito. Ad esempio individuare tutti i pixel rossi di un'immagine risulta più semplice se questa è codificata in formato RGB.



Esempio:

- Obiettivo: partendo dalle coordinate di un punto su un grafico individuare la corretta classificazione di colore, se si tratta di un punto nero o uno bianco
- Input: coordinate dei punti
- Output: colore dei punti
- Valutazione: percentuale di punti classificati correttamente

Ricevuto i dati di input il primo step consiste nell'individuare la miglior rappresentazione in modo che la separazione tra i punti colorati sia netta e si possa definire una regola di classificazione precisa:



questo può essere fatto attraverso una trasformazione delle coordinate, come nell'immagine, cosicché tutti i punti bianchi si trovino nell'area del piano individuata da X<0 e quelli neri in quella X>0.

65

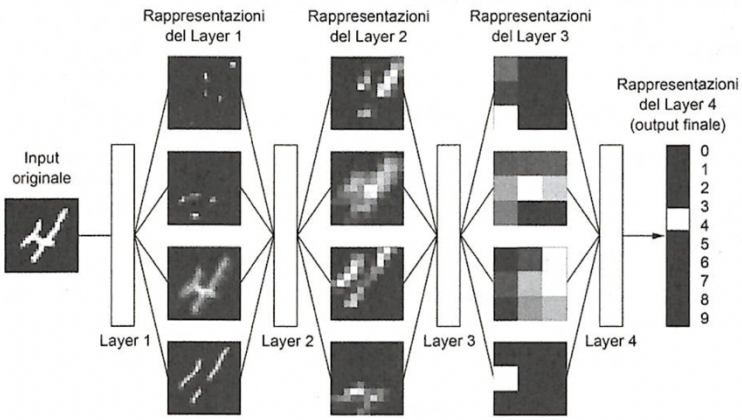
[65] Francois Chollet; Deep Learning con Python; 2020; pag. 7

Questa trasformazione può essere fatta manualmente con la programmazione tradizionale oppure tramite il machine learning, usando il feedback per individuare automaticamente la tipologia di modifica dei dati da utilizzare.

Gli algoritmi di machine learning non sono molto creativi nella ricerca delle trasformazioni da effettuare e selezionano l'operazione da uno spazio predefinito chiamato "spazio delle ipotesi".

Il machine **learning** tecnicamente quindi è *la ricerca delle rappresentazioni più utili dei dati di input in uno spazio di possibilità predefinito usando come guida un segnale di feedback* [1][66], e questo permette di risolvere una vasta gamma di compiti mappando degli input su target sfruttando molti esempi di input e output.

Il deep learning, anche chiamato apprendimento a rappresentazioni gerarchiche o tramite rappresentazioni a layer, è una branca del machine learning in cui il termine "deep" indica l'obiettivo di individuare layer(livelli) di rappresentazione dati sempre più profondi e significativi. Il numero di layer, solitamente decine o centinaia, che contribuiscono a formare un modello dei dati e detto depth (profondità) del modello. Quando ci si basa solamente su uno o due layer si parla di apprendimento superficiale o shallow learning o machine learning. Le rappresentazioni a layer vengono apprese automaticamente con l'uso dei dati di addestramento tramite modelli chiamati reti neurali strutturate attraverso strati di layers sovrapposti. I layer quindi altro non sono che lunghe catene di operazioni geometriche applicate in successione.

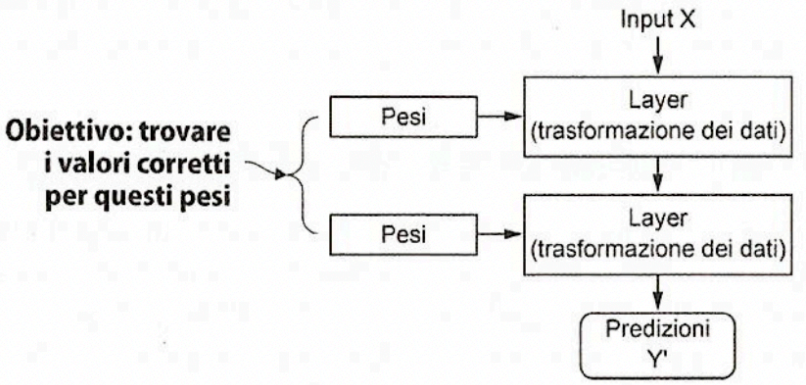


[67]

Ecco un esempio di rete neurale con 4 layer di profondità con lo scopo di capire la cifra rappresentata nell'immagine. Si può immaginare come un sistema di filtri multi-fase, individuati dai layer, in ciascuno dei quali avviene una rappresentazione differente dell'immagine che risulta maggiormente purificata più si procede in profondità.

[66] Francois Chollet; Deep Learning con Python; 2020; pag. 8
[67] Francois Chollet; Deep Learning con Python; 2020; pag. 10

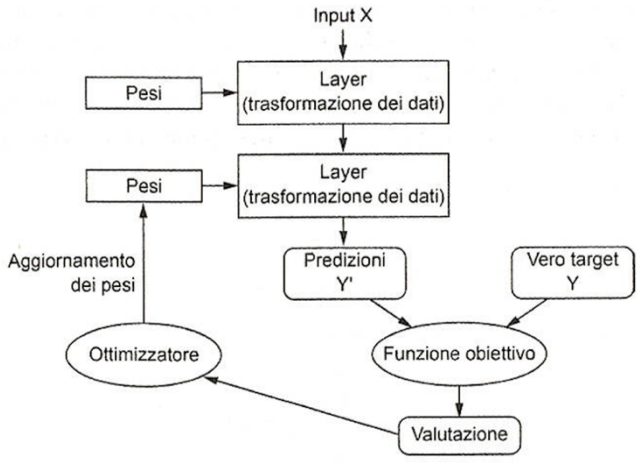
Le operazioni che vengono effettuate sui dati in un determinato layer sono memorizzate nei così detti pesi(weight) o parametri del layer. Con il termine learning si intende proprio il cercare l'insieme dei valori dei pesi di tutti i layer coinvolti in un determinato compito in modo da mappare correttamente gli input con i rispettivi target attraverso delle combinazioni lineari.



[68]

La vera conoscenza di un modello è proprio conservata e risiede nei pesi.

Una rete neurale può contenere milioni di parametri e calcolare il valore corretto di ciascuno di essi è molto dispendioso soprattutto considerando che la modifica di uno solo di essi influenza il comportamento degli altri. È necessario quindi disporre un controllo sulla rete in grado di misurare quanto ci si sta avvicinando al risultato corretto e questo viene effettuato con la funzione obiettivo o loss function.



[69]

Questa prende in input le previsioni della nostra rete Y', il risultato effettivo che dovrebbe essere prodotto Y e ne calcola la "distanza" tramite un punteggio, feedback. La valutazione è poi passata all'ottimizzatore che implementa l'algoritmo di retropropagazione, algoritmo centrale nell'apprendimento automatico, che ha lo scopo di regolare i pesi al fine di diminuire la differenza tra il desiderato e il predetto, cioè si vuole minimizzare la funzione obiettivo.

Senza informazioni inizialmente ai pesi è assegnato un piccolo valore *casuale* e vengono usati dalla rete per effettuare operazioni *casuali* sui dati. L'insieme di tutti questi passaggi è chiamato ciclo di addestramento e una volta che la funzione obiettivo fornisce un valore minimo tale per cui le previsioni sono molto vicine ai target si parla di ***rete addestrata.***

L'idea di cercare di addestrare una macchina con il fine di creare un'intelligenza artificiale simil-umano vede la sua origine nei primi anni 50 e con il passare del tempo ha attraversato diversi cicli, in cui fasi di ottimismo con conseguenti elevati investimenti si sono alternate a delusioni per il mancato raggiungimento di obiettivi e il conseguente declino dei finanziamenti, i così detti inverni

[68] Francois Chollet; Deep Learning con Python; 2020; pag. 10
[69] Francois Chollet; Deep Learning con Python; 2020; pag. 12

della AI (metà anni 70 e primi anni 90). In questo momento si ritiene che ci troviamo all'apice di entusiasmo del terzo ciclo dove però per gli addetti ai lavori c'è più consapevolezza sul fatto che sebbene è necessario limitare le aspettative sul breve periodo le prospettive sul lungo periodo sono importanti. Negli ultimi anni son stati fatti passi da gigante in questa direzione ma solo poche soluzioni vengono applicate in modo pratico e questo ne limita le potenzialità in tutti i campi della nostra vita, soprattutto in quello medico.

Il deep learning non sempre però è da considerarsi lo strumento giusto per risolvere i problemi e talvolta in base ai dati o al problema è meglio procedere con altri algoritmi del machine learning.

2.1. Un po' di storia

Il machine learning è *l'applicazione e la scienza degli algoritmi che traggono informazioni utili dai dati*[70] **trasformando i dati in conoscenza** ed è nato con lo scopo di velocizzare la creazione di regole e modelli predittivi per l'analisi dei dati. (es Siri per Iphone)

Una delle prime forme di machine learning ancora utilizzata, antecedente addirittura ai computer, è la modellazione probabilistica, cioè l'applicazione dei principi statistici all'analisi dei dati. L'algoritmo classificatore di Naive Bayes è uno dei più noti di questa tipologia, è basato sul teorema di Bayes e parte dal presupposto che le caratteristiche dei dati di input siano tutte indipendenti.

Un ulteriore passo in avanti venne fatto a metà degli anni 80 con l'introduzione dell'algoritmo di retropropagazione, che utilizza l'ottimizzazione a discesa del gradiente(spiegato più avanti) per l'addestramento di catene di operazioni parametriche, anche se la prima vera applicazione pratica nelle reti neurali arrivò nel 1989 nell'ambito della classificazione delle cifre scritte a mano e la rete neurale risultante LaNet è stata utilizzata dal servizio postale degli Stati Uniti negli anni 90.

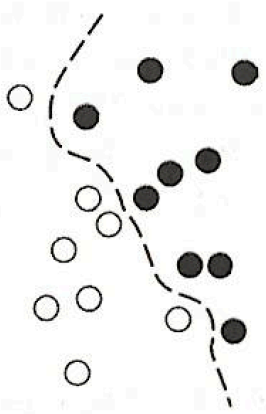
Dopo i primi successi nacque un nuovo approccio nel machine learning composto da metodi chiamati metodi Kernel: costituiscono un gruppo di algoritmi di classificazione, il più famoso è il Support Vector Machine (SVM), con l'obiettivo di risolvere problemi di classificazione cercando di individuare dei confini decisionali fra 2 insiemi di punti di 2 diverse categorie nei dati di addestramento. Il confine decisionale può essere espresso come un iperpiano(che collassa in una linea in caso di dati bidimensionali), una linea o una superficie e per classificare un nuovo dato sarà sufficiente vedere in quali delle aree individuate dai confini ricade.

Il SVM si svolge in due passi:

I. Mappatura dei dati in una nuova rappresentazione multidimensionale in cui viene individuato il confine decisionale
II. Il confine decisionale deve essere calcolato con l'obiettivo di massimizzare la distanza tra l'iperpiano e i punti più vicini, *massimizzazione del margine*, in modo che le classificazioni siano più chiare

[70] S. Raschka & V. Mirjalili; Machine Learning con Python; 2020; pag. 1

Lato computazionale l'identificazione di un buon confine decisionale può essere molto difficile e in questi casi interviene il Kernel trick: afferma che non è necessario nel nuovo spazio calcolare le coordinate dei nostri punti ma è sufficiente calcolare la distanza fra coppie di punti attraverso la funzione di kernel. Questa operazione viene fatta solitamente a mano e mappa due punti casuali nello spazio iniziale nello spazio target alla medesima distanza evitando di effettuare nuovamente il calcolo nella nuova rappresentazione, con SVM viene appreso solo l'iperpiano di separazione.

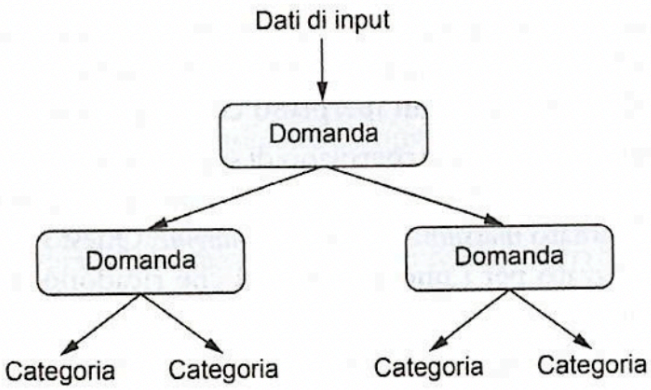


[71]

Le SVM risultarono ottime per semplici problemi di classificazione ma erano di difficile estensione a grossi dataset e non davano buoni risultati per problemi percettivi come la classificazione immagini.

Un altro tipo di algoritmo applicato a partire dagli anni 2000 sono gli alberi decisionali, simili a flowchart, che consentono di classificare dati in input o predire valori in base ad altri input.

Alcune di queste tipologie sono delle volte preferite al metodo Kernel, come la *foresta casuale* che rappresenta un approccio solido e pratico in cui si creano un elevato numero di alberi decisionali specializzati con output che vengono valutati tutti assieme. Applicabili ad un'ampia gamma di problemi sono considerati il secondo miglior algoritmo per molte soluzioni di machine learning.



[72]

Dal 2014 però all'approccio a foreste casuali fu preferito quello delle *macchine ad amplificazione del gradiente*. Questa tecnica è basata sull'assemblaggio di più modelli predittivi deboli ed utilizza l'amplificazione del gradiente, cioè un metodo per migliorare i modelli di machine learning addestrando iterativamente i nuovi modelli in modo da specializzarli nel risolvere i punti deboli di quelli precedenti. Applicandolo agli alberi decisionali vengono generati modelli che spesso risultano migliori delle foreste casuali e per questo motivo assieme al deep learning è uno degli algoritmi migliori per compiti non percettivi.

Fino al 2010 le reti neurali rimasero un po' ai margini del machine learning ma poi tornarono alla ribalta fornendo risultati importanti rispetto agli altri algoritmi di machine learning in competizioni accademiche incentrate sulla classificazione delle immagini. Dal 2012 le reti neurali profonde convolutive divennero le preferite per compiti di tipo percettivo e raggiunsero il picco di popolarità

[71] Francois Chollet; Deep Learning con Python; 2020; pag. 16
[72] Francois Chollet; Deep Learning con Python; 2020; pag. 18

nel 2015 quando il vincitore di una gara ha raggiunto il 96,4% di correttezza nella classificazione di immagini. In questo modo iniziarono ad essere applicate in molti altri ambiti, sostituendo le SVM e gli alberi decisionali, com'è avvenuto al CERN dove per l'analisi dei dati delle particelle rivelate viene usato la rete neurale profonda Keras.

La veloce ascesa del deep learning è dovuta al fatto che:

- offre migliori prestazioni per numerosi compiti
- semplifica le attività di problem solving automatizzando il feature engineering (cioè l'attività manuale fatta dagli operatori al fine di rendere i dati di input più adatti all'elaborazione).

Il modello studia contemporaneamente tutti i layer di rappresentazione, mentre con gli altri algoritmi avveniva in sequenza, avviene quindi un apprendimento congiunto delle funzionalità e ogni volta che una di queste viene aggiustata anche tutte le altre dipendenti vengono adattate avvicinandosi all'obiettivo finale più velocemente. In questo modo si sviluppano rappresentazioni sempre più complesse e le rappresentazioni incrementali intermedie vengono apprese congiuntamente, ogni layer deve assecondare le esigenze di rappresentatività del sottostante e del superiore.

Machine learning e deep learning hanno visto la loro maggior espansione soprattutto negli ultimi anni anche grazie all'aumento di disponibilità e capacità delle loro 3 componenti principali:

- l'hardware
- dataset e benchmark
- miglioramenti algoritmici

Come detto prima si tratta di una scienza prettamente ingegneristica, né matematica né statistica, dove le idee vengono messe alla prova testando con maggior efficacia e velocità nuovi algoritmi grazie ad hardware più evoluti e maggiori dati a disposizione.

Il deep learning è considerato una rivoluzione nel campo dell'intelligenza artificiale e anche se le reti neurali potranno essere sostituite in futuro le sue proprietà principali serviranno da base per i nuovi metodi:

- semplicità: automatizzando la fase di feature engineering
- scalabilità: adatto alla parallelizzazione su CPU e GPU anche di piccoli lotti di dati
- versatilità: adatti per l'addestramento continuo con dati aggiuntivi senza ripartire da zero, riutilizzabili e ri-orientabili per compiti simili

2.2. Costruzione di sistemi di machine learning

2.2.1. Terminologie e componenti necessari

Prima di procedere è necessario fare un po' di chiarezza sui termini che verranno usati:

- Classi: insieme di possibili etichette fra cui scegliere in problemi di classificazione
- Etichetta: istanza di un'annotazione appartenente a una classe in problemi di classificazione. Ad esempio se ho delle immagini di cani e gatti, se un'immagine è annotata come contente la classe cane, questa è l'etichetta
- Annotazioni o verità di base: insieme dei target di un dataset, valori con cui possono essere identificati/descritti gli oggetti del dataset
- Campione(sample) o input: singolo punto dei dati nel modello o dataset
- Predizione o output: risultato prodotto dal modello
- Target: risultato effettivo che dovrebbe generare il modello in base agli input
- Errore nella predizione o valore loss: distanza tra predizione e target
- Classificazione:
    - Binaria: ogni campione di input può essere catalogato su 2 categorie mutuamente esclusive
    - Multi-etichetta: a ogni campione di input può essere assegnata più di una etichetta
- Regressione
    - scalare: compito in cui il target è un valore scalare continuo
    - vettoriale: compito in cui il target è un insieme di valori scalari continui
- Mini-batch o batch(lotto): piccolo set di campioni, solitamente una potenza di 2 per facilitare allocazione su memoria, elaborati simultaneamente nel modello
- Iperparametri: parametri propri della rete come il numero di layer e le loro dimensioni
- Parametri: pesi della rete
- Layer: elemento base che costituisce le reti, da immaginare come un filtro in cui i dati di ingresso vengono elaborati e estratti in una rappresentazione più significativa per il nostro compito. Esempio di layer in Python
    - Dense: densamente connessi, ogni layer è connesso al successivo attraverso un peso. **Codifica**: *network.add(layers.Dense(10, activation= 'softmax')* vuol dire che è di tipo 10-way softmax, cioè restituisce un array di 10 punteggi di probabilità a somma 1. Ogni punteggio è la probabilità che la nostra immagine appartenga a una delle classi

Il deep learning consiste in una concatenazione di layer con il compito di raffinare sempre di più i dati in ingresso. Oltre ai layer per impostare l'addestramento di una rete è necessario aggiungere altre 3 cose riguardanti il passo di compilazione:

i. Una funzione obiettivo: utilizzata dalla rete per misurare le prestazioni durante l'addestramento in modo da capire se ci si sta muovendo nella corretta direzione
ii. Un ottimizzatore: meccanismo con cui la rete si aggiorna in base al risultato della funzione obiettivo e ai dati di ingresso
iii. Metriche da monitorare in addestramento e testing: definiscono l'accuratezza dei risultati, per esempio la percentuale di punti classificati correttamente in un compito di classificazione

I modelli di deep learning non lavorano su un intero dataset alla volta ma li suddivide in piccoli lotti, batch, per poi elaborarli.

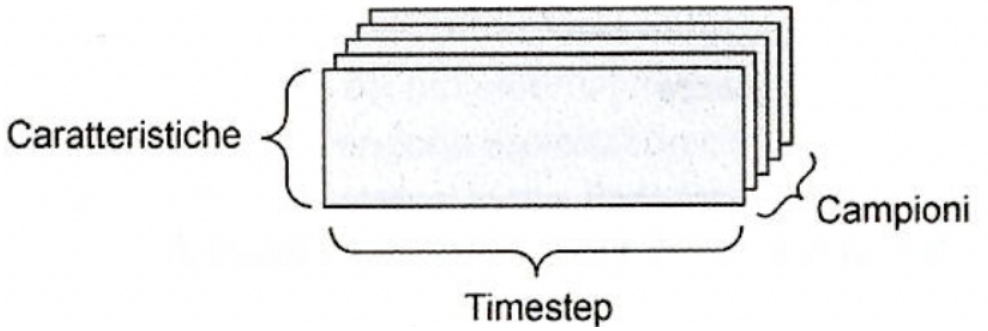
2.2.2. Tensori

Per essere utilizzati dalle reti neurali i dati sono memorizzati come struttura di base in array multidimensionali chiamati ***tensori***. Un tensore è un contenitore di dati, quasi sempre numerici, sono una generalizzazione delle matrici per un numero n di dimensioni(assi).

I tensori sono definiti da 3 attributi:

- Numero di assi(rango): dimensione del tensore. Generalmente il primo asse di tutti i tensori, l'asse 0, è detto asse dei campioni (dimensione dei campioni) o asse di batch (dimensione dei batch) se si parla di un tensore di batch. In base al rango abbiamo i tensori:
    - 0D: contiene un solo numero, è uno scalare e ha 0 dimensioni
    - 1D: è un array di numeri, vettore, ed ha un solo asse. Il numero di valori che contiene ne determina la dimensionalità
    - 2D: vengono così chiamate le matrici che sono dotate di 2 assi, righe e colonne
    - 3D: ottenuti inserendo le matrici in un nuovo array che visivamente è assimilabile a un cubo di numeri

  Procedendo come per i 3D si possono creare tensori sempre con maggiori dimensioni, solitamente nel deep learning si lavora con tensori da 0D a 4D, 5D solo in caso di elaborazioni video.
- Forma(shape): tupla di interi che descrive quante dimensioni/valori ha il tensore lungo ogni asse. Ad esempio per una matrice 3x5 la forma sarà (3, 5).
- Tipo di dati: tipo di dati contenuti e solitamente sono
    - Dati vettoriali: i più comuni, in cui ogni singolo punto può essere codificato come un vettore, un lotto di punti quindi sarà un tensore 2D in cui il primo asse è quello dei campioni e il secondo è delle caratteristiche. Ad esempio un dataset di dati di 10 persone contente età, cap e città può essere memorizzato in un tensore 2D con forma (10,3)
    - Dati di serie temporali o sequenziali: utilizzati per raffigurare tutti quei dati in cui il tempo rappresenta un fattore importante e in questi casi vengono usati tensori 3D in cui il secondo asse è per convenzione quello temporale



[73]

Ad esempio in un dataset di prezzi azionari, ogni minuto viene codificato come un vettore 3D dove memorizziamo il prezzo più alto, quello più basso e quello aggiornato mentre una giornata come un tensore 2D con forma (390(minuti), 3)

- Dati di immagini: si utilizzano sempre tensori 3D per una singola immagine in quanto è caratterizzata da altezza, larghezza e profondità dei colori. Quindi un dataset di immagini è rappresentato con un tensore 4D per cui ci sono 2 convenzioni di forma:
    - channels-last, asse profondità dei colori alla fine
    - channels-first, asse profondità dei colori subito dopo il primo

[73] Francois Chollet; Deep Learning con Python; 2020; pag. 37

- Dati di video: richiedono tensori 5D in quanto considerato come una sequenza di frame, ciascuno memorizzabile con tensori 3D, a cui si aggiunge il fattore tempo(4D) e l'utilizzo di insiemi di frame 5D

2.2.2.1. Operazioni sui tensori

Tutte le trasformazioni che vengono fatte nelle reti neurali possono essere ridotte a poche operazioni su tensori.

Se prendiamo ad esempio un layer di tipo dense della libreria Keras di Python, questo è definito come segue:

keras.layers.Dense(512, activation= 'relu')

questa funzione prende in input un tensore 2D e ne restituisce una nuova rappresentazione generando un output di questo tipo:

output= relu(dot(W, input)+b)

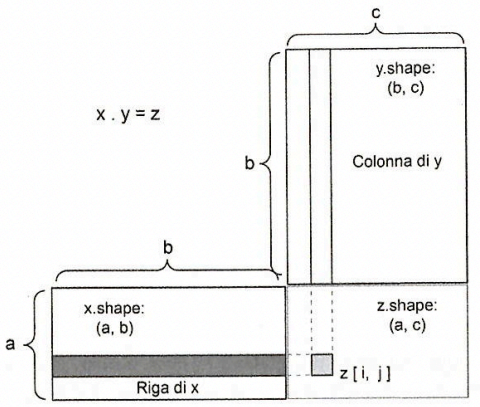
dove il prodotto(dot) tra il tensore in input e un tensore W, matrice dei pesi, viene sommato a un vettore b, bias che rappresenta l'errore di approssimazione tra il vero risultato dei dati di ingresso e quello calcolato mediante l'algoritmo.

W e b sono gli attributi del layer e sono chiami pesi o parametri addestrabili.

Le operazioni somma e relu sono dette *element-wise* perché vengono applicate in modo indipendente a ciascun elemento dei tensori, adatte quindi a implementazioni a parallelismo massivo. Ecco alcuni tipi di operazioni:

- Relu: sceglie il massimo tra il valore risultante e 0, cioè relu(x)= max(x,0)
- Broadcasting: viene effettuata quando vengono sommati tensori con forme differenti come un tensore 2D e un vettore. Il tensore minore viene elevato in broadcast per adattarlo a quello maggiore in 2 steps
  - Al più piccolo vengono aggiunti degli assi, assi broadcast, fino ad avere la stessa dimensione del più grande
  - Il tensore più piccolo è ripetuto lungo questi nuovi assi fino a pareggiare la forma

  Questi due passaggi vengono fatti solo virtualmente al momento del calcolo in quanto creare un nuovo tensore completo 2D risulterebbe inefficiente a livello computazionale
- Dot: effettua il prodotto di tensori e a differenza delle operazioni *element-wise* combina gli elementi dei tensori in input. Come si vede dall'immagine vale la medesima regola del prodotto tra matrici in cui la prima dimensione di x deve essere uguale alla 0-esima dimensione di y.
  Per tensori multidimensionali vale la stessa regola dei 2D e per X sarà considerata l'ultima dimensione



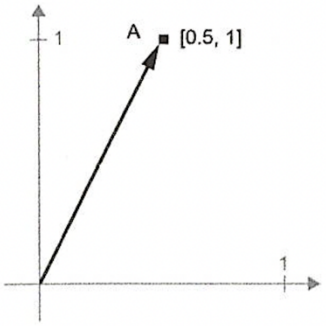
[74]

[74] Francois Chollet; Deep Learning con Python; 2020; pag. 43

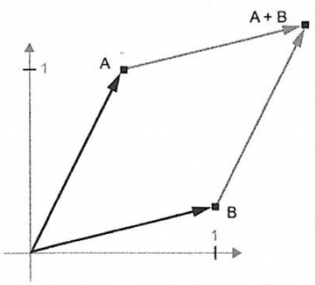
- Reshaping: eseguire il reshaping di un tensore significa ridisporre le sue righe e colonne in modo abbia una specifica forma mantenendo lo stesso numero di coefficienti. Un esempio è la trasposizione delle matrici

I tensori possono essere visti come coordinate di punti dello spazio geometrico e anche le relative operazioni hanno la medesima interpretazione.

Per esempio un vettore A = [0.5, 1] è un punto sullo spazio geometrico 2D collegato all'origine degli assi da un versore orientato al punto.



[75]



Anche la somma con un altro vettore può essere vista geometricamente e tutte le operazioni geometriche elementari possono essere utilizzate come operazioni su tensori.

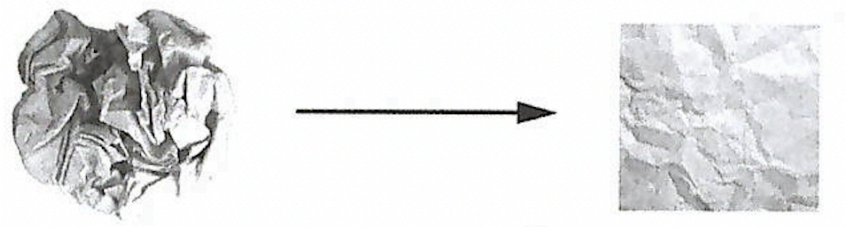
Il primo argomento passato a ciascun layer alla creazione, ad esempio Dense(512), è il numero di unità nascoste del layer. Ogni unità nascosta è una dimensione nello spazio di rappresentazione del layer e significa che la matrice W avrà la forma(input_dimension, 512), quindi il nostro dataset di input viene proiettato su uno spazio 512-dimensioni.

La dimensionalità dello spazio indica la libertà che è lasciata alla rete nell'apprendimento delle rappresentazioni, cioè ogni punto del dataset può essere "descritto" attraverso n attributi. Un elevato numero di unità nascoste permettono l'apprendimento di rappresentazioni più complesse ma anche di schemi superflui che peggiorano le prestazioni nei test e richiedono maggiori costi computazionali.

[75] Francois Chollet; Deep Learning con Python; 2020; pag. 45

2.2.2.2. Tensori nel deep learning

Essendo che le reti neurali sono costituite da catene di operazioni sui tensori ne consegue che è possibile interpretare una rete neurale come una trasformazione geometrica molto complessa in uno spazio multidimensionale.



In particolare quando si parla di immagini 3D si può immaginare il nostro insieme di dati in input come due fogli di carta di diversi colori accartocciati in modo da formare una pallina.

[76]

L'obiettivo di ogni modello di machine learning è quello di trasformare la pallina con il fine di poter ridistendere i fogli di carta così da poter identificare chiaramente le classi rappresentate dai fogli rendendole separabili.

Il compito di ogni modello è quindi trovare le rappresentazioni più semplici per i nostri dati e il deep learning lo fa in modo eccellente decomponendo in modo incrementale una complessa trasformazione geometrica in una lunga serie di trasformazioni elementari.

Le operazioni, che avvengono nei layer, agiscono su quelli definiti parametri addestrabili W e b, anche chiamati attributi kernel e bias, che contengono le informazioni apprese dalla rete dai dati di addestramento. Tramite uno step chiamato inizializzazione casuale gli viene assegnato un valore iniziale piccolo e casuale e gradualmente vengono aggiustati in base ai feedback della funzione obiettivo, ***learning***.

Tutto questo fa parte di quello che è chiamato ciclo di addestramento:

I. Estrae un lotto(batch) di campioni X e i corrispondenti target Y
II. Tramite il passo di avanzamento applica la rete a X ed estrae le predizioni pred_Y
III. Calcola il loss della rete sull'input di addestramento, cioè la differenza tra i valori predetti pred_Y e quelli target Y
IV. Aggiorna i pesi della rete con il fine di minimizzare la distanza misurata con il loss

Lo step principale è l'aggiornamento dei pesi che può essere fatto in diversi modi:

- il primo è di congelare tutti i pesi della rete tranne uno alla volta che viene aggiornato nella direzione di minimizzazione del loss. È un approccio decisamente inefficiente perché è necessario eseguire lo stesso calcolo per ogni coefficiente
- il processo migliore sfrutta il fatto che tutte le operazioni della rete sono differenziabili e calcola il gradiente del valore loss considerando i coefficienti della rete, aggiornando in un'unica operazione tutti i coefficienti nella direzione desiderata.

[76] Francois Chollet; Deep Learning con Python; 2020; pag. 46

Derivata:

in una funzione regolare e continua f(x)= y, che associa a un valore reale x un numero reale y, una piccola variazione ε_x di x comporta una piccola variazione ε_y su y. Essendo una funzione regolare, curva senza angoli netti, se ε_x è abbastanza piccolo nei dintorni di un determinato punto p è possibile approssimare f con una funzione lineare di pendenza a dove $\varepsilon_y = a*\varepsilon_x$

$$f(x+\varepsilon_x)=y + a*\varepsilon_x$$

la pendenza a è la derivata di f in p e se:

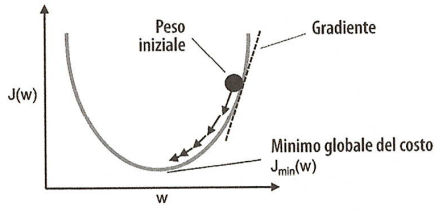
- <0→ piccola variazione positiva di x attorno a p produce una riduzione di f(x) e viceversa
- >0→ piccola variazione positiva di x attorno a p produce un aumento di f(x) e viceversa

E il valore assoluto |a| dice quanto velocemente queste variazioni si verificano.

Per ogni funzione differenziabile esiste la funzione derivata f'(x) che associa ai valori di x la pendenza dell'approssimazione lineare della curva nel punto e ci permette di capire come si muove la nostra funzione a seguito della modifica del parametro. Conoscendo questi aspetti per ridurre il valore di una funzione f(x) sappiamo che è sufficiente modificare x nella direzione opposta della derivata.

Gradiente:

Generalizzazione del concetto di derivate su funzioni con input multidimensionali(tensori). Il gradiente fornisce direzione e verso nei quali la funzione cresce più velocemente.



Nel nostro caso quindi è la derivata di un'operazione sul tensore e considerando un input x, una matrice W, un target y e una funzione obiettivo *loss* si può

- Calcolare la previsione y_pred = dot(W,x)
- Il valore loss_value= loss(y_pred , y)

Essendo bloccati x e y in quanto dati noti, possiamo vedere i due step come un'unica funzione che associa dei valori loss ai pesi W: loss_value= f(W).

La derivata di f in un punto W0 è un tensore gradient(f)(W0), con la stessa forma di W, dove ciascun coefficiente gradient(f)(W0)[i,j] indica la direzione e la magnitudine l'entità della variazione del loss_value alla modifica di (W0)[i,j]. Il gradient(f)(W0) è il gradiente della funzione f in W0 e come la derivata di una funzione di un singolo coefficiente è interpretata come la pendenza della curva, anche il gradiente può essere visto come il tensore che descrive la curvatura di f(W) e di conseguenza si può ridurre il valore della funzione f(x) facendo variare x nella direzione opposta alla derivata:

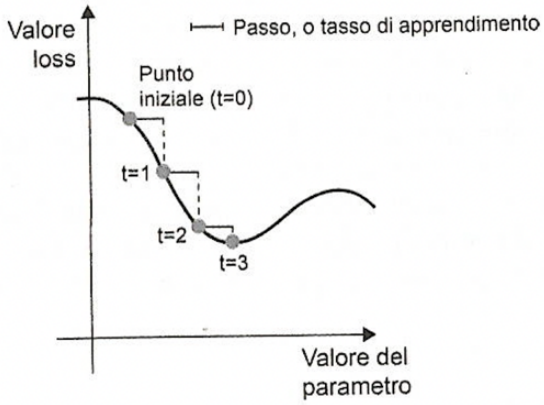
$$W1= W0 - step*gradient(f)(W0) \text{ con } \quad step= \text{fattore di scala}$$

77

[77] S. Raschka & V. Mirjalili; Machine Learning con Python; 2020; pag.34

Questo avviene tramite l'operazione chiamata **discesa stocastica del gradiente (SGD)** che sfrutta il fatto che per trovare il minimo di una funzione è sufficiente individuare i punti in cui la derivata è uguale a 0. Nel caso della rete neurale quindi lo scopo è individuare la combinazione dei pesi che minimizza la funzione obiettivo risolvendo l'equazione polinomiale gradient(f)(W)=0 e aggiornando tutti i pesi *W= -step*gradient*. Il termine stocastico si riferisce al fatto che ogni lotto viene scelto causalmente dal dataset.

[78]



Nell'immagine vediamo un esempio grafico per un tensore 1D.

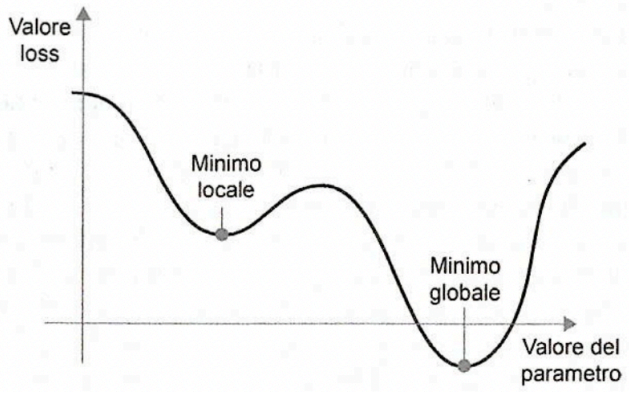
È importante scegliere un valore *step* ragionevole in quanto:

- Se troppo piccolo: la discesa della curva richiederà troppe iterazioni con il rischio di incappare in un minimo locale
- Se troppo grande: rischio scelta di posizioni casuali sulla curva

È inoltre possibile operare su batch, algoritmo mini-batch SGD, utilizzare un unico campione e relativo target a ogni iterazione oppure ancora usare tutti i dati disponibili. Ovviamente più dati si utilizza più l'aggiornamento dei pesi è accurato a discapito però di un maggior costo a livello computazionale. Il compromesso migliore resta quindi quello di utilizzare l'approccio a mini-batch cercando di individuare la dimensione migliore.

Uno dei vantaggi della discesa stocastica del gradiente è che può essere utilizzata per l'addestramento online, dove il nostro modello riceve continuamente nuovi dati di addestramento, particolarmente utile quando si stanno accumulando molti dati da applicazioni web in quanto permette al sistema di adattarsi immediatamente ai cambiamenti.

Esistono più varianti della SGD che differiscono per il fatto di tenere in considerazione gli aggiornamenti dei pesi precedenti nel calcolo del successivo invece di considerare solo il valore attuale. Queste varianti sono dette ottimizzatori e una di queste è la **SGD momento,** dove il termine momento si riferisce ai 2 problemi prima indicati:



[79]

- Velocità di convergenza
- Minimi locali

Un passo di apprendimento troppo breve bloccherebbe l'aggiornamento nel minimo locale senza poter trovare il globale in quanto ad ogni step sembrerebbe di allontanarsi dal target di minimizzazione. Va immaginano come una pallina che scende lungo un pendio in modo non sufficientemente veloce da superare gli avvallamenti momentanei che incontra. Il momento è implementato considerando a ogni passo sia la pendenza che

[78] Francois Chollet; Deep Learning con Python; 2020; pag. 50
[79] Francois Chollet; Deep Learning con Python; 2020; pag. 52

la velocità attuale derivata dall'accelerazione precedente, cioè tenendo conto del gradiente corrente e anche del precedente aggiornamento del parametro w.

Nella pratica la funzione di una rete neurale è composta da molte operazioni concatenate sui tensori, su ciascuna delle quali è possibile calcolare una derivata semplice. Per ottimizzare il valore del loss si applica la regola della catena

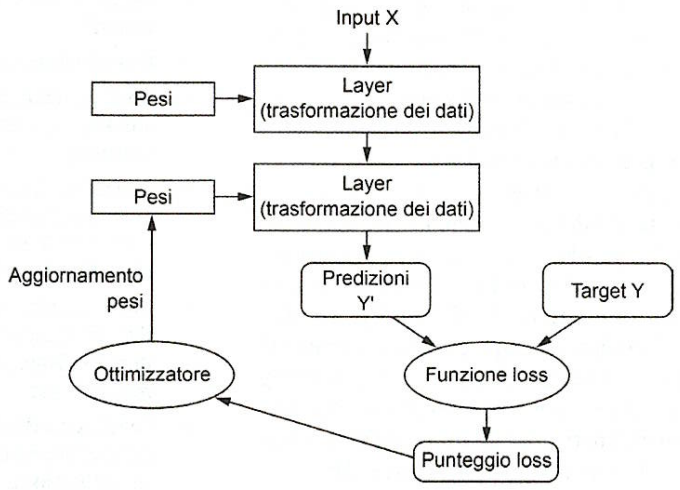
$$f(g(x)) = f\,'\,(g(x))^{*}\, g\,'\,(x)$$

al calcolo dei valori del gradiente del loss procedendo all'indietro dai layer superiori a quelli inferiori in modo da calcolare il contributo dato da ciascun parametro alla minimizzazione, ***algoritmo di retropropagazione.***

Questo algoritmo è implementato a codice tramite framework, come TensorFlow, che eseguono tutto il procedimento.

2.2.3. Il deep learning

Ricapitolando quanto anticipato per addestrare una rete neurale sono necessari:



- dati di addestramento, input e target
- layer concatenati in un modello, "filtrano" i dati per determinare delle predizioni
- la funzione obiettivo/loss, confronta le predizioni con i target calcolando un punteggio che deve essere minimizzato durante l'addestramento
- l'ottimizzatore, modo in cui la rete viene aggiornata modificando i pesi in base al feedback della funzione obiettivo (es SGD)

[80]

2.2.3.1. Modelli di reti di layer

Il layer è l'unità base delle reti neurali, una sorta di mattoncino lego che funziona da modulo di elaborazione dati di input, 1 o più tensori, e produce uno o più tensori in output. Quasi tutti i layer posseggono uno stato rappresentato dai pesi (1 o più tensori), appresi con tecniche come la discesa stocastica del gradiente, e il loro insieme forma la conoscenza della rete.

Esistono diversi tipi di layer che si possono utilizzare in base ai differenti formati dei tensori e alla tipologia di elaborazione dei dati:

- Densamente connessi: per semplici dati vettoriali conservati in tensori 2D

[80] Francois Chollet; Deep Learning con Python; 2020; pag. 58

- Ricorrenti: ad esempio LSTM, per dati sequenziali conservati in tensori 3D
- Convolutivi 2D: dati di immagini conservati in tensori 4D

In alcuni framework, come Keras, la costruzione di un modello di deep learning è possibile solo mettendo insieme dei layer compatibili, in quanto ogni layer può ricevere tensori in input o produrne in output solo in determinate forme. Il tensore in output di un layer solitamente rappresenta l'input di un altro layer che deve essere in grado di ricevere in ingresso la medesima forma di tensore.

Un modello di deep learning è un grafo diretto e aciclico di layer, i più popolari sono:

- Stack lineare di layer: la versione più comune in cui a ogni input è associato un singolo output. Ideale quando le etichette sono numeri scalari, 0-1, e i dati di input sono vettori
- Reti a due rami
- Reti multihead
- Inception block

La topologia di una rete è molto importante in quando definisce uno spazio delle possibilità che vincola il modello a una specifica serie di operazioni sui tensori. Queste a loro volta permettono di mappare i dati in input su dati di output sempre con l'obiettivo di definire un buon insieme di valori per i tensori dei pesi. La scelta di una rete specifica è quindi molto importante e tramite alcune regole di best practice e all'esperienza si impara a scegliere quella corretta.

2.2.3.2. Funzioni obiettivo e ottimizzatori

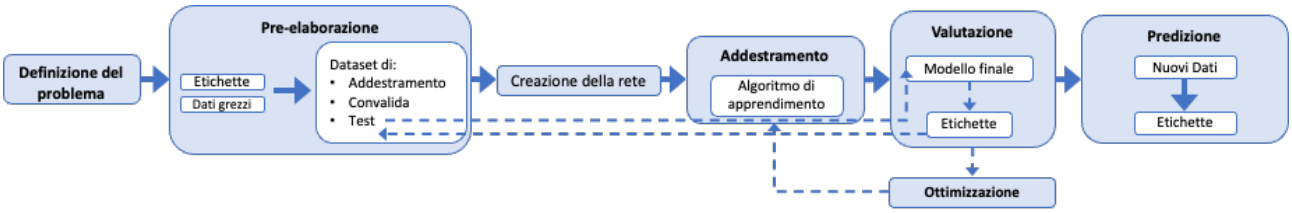
Una rete neurale può avere più output e in alcuni di questi casi più funzioni obiettivo, 1 per ciascun output. Il metodo SGD deve basarsi però su un unico valore scalare loss e quindi quando si ha una rete multi-loss tutti i valori loss vengono combinati tramite media in un'unica quantità scalare. Come per la topologia anche la scelta di una funzione obiettivo è fondamentale in quanto la rete farà il possibile per cercare di minimizzare il loss e se la funzione non è in linea con il compito da raggiungere non si avrà l'effetto desiderato. Si sa però che per alcuni scopi si devono usare determinate funzioni, ad esempio per un problema di

- classificazione binaria→ si usa l'entropia incrociata binaria
- classificazione multi-classe→ si usa l'entropia incrociata categorica
- regressione→ si usa l'errore quadratico medio
- apprendimento di una sequenza→ si usa la *connectionist temporal classification* CTC

l'entropia incrociata è la scelta migliore quando gli output sono di tipo probabilistico in quanto misura la distanza tra la vera distribuzione di probabilità e le predizioni.

2.2.4. Le principali fasi della costruzione del sistema

In linea di massima si può individuare uno schema per la costruzione di un sistema di machine learning composto dalle seguenti fasi:



2.2.4.1. Definizione del problema

Prima di tutto è importante definire con precisione il problema determinando:

- Quali saranno i dati in input
- Quali saranno i dati di output, cioè quello che si vuole predire
- Qual è il problema da affrontare (classificazione binaria, classificazione multi-classe, regressione scalare, regressione vettoriale….)

Non si può procedere con gli altri step se non si trova una risposta a queste domande ed è inoltre necessario assicurarsi la disponibilità dei dati, la cui correttezza in termini di ricchezza di informazioni rilevanti al nostro problema sarà verificata a modello funzionante. Ad esempio in un sistema di trading la cronologia dei prezzi non è detto che sia sufficientemente rilevante in quando da sola non contiene sufficienti informazioni predittive.

2.2.4.2. Pre-elaborazione

In questa fase di fondamentale importanza vengono organizzati e formattati i dati grezzi in modo che siano leggibili e analizzabili da un algoritmo. Spesso le caratteristiche sono correlate e risultano essere ridondanti per questo motivo sono utilizzate delle tecniche per ridurne la dimensionalità. Molte tecniche di pre-elaborazione dei dati e di manipolazione delle caratteristiche (feature engineering) che vengono utilizzate dipendono dal dominio dei dati. Per esempio in caso il dataset sia composto da parole queste vengono prima "convertite" in numeri interi le cui liste sono poi convertite in tensori.

Le operazioni sui dati più utilizzate sono:

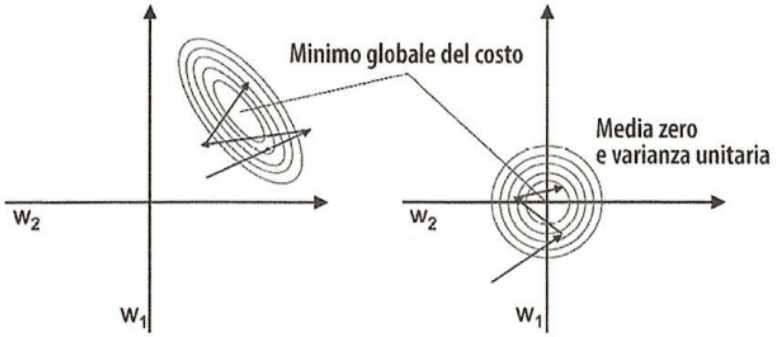
- **Vettorizzazione:** è la trasformazione di tutti gli input e target di una rete neurale in tensori di dati in virgola mobile, qualunque sia la tipologia di dati (es. suoni, testi e immagini), in quanto le reti neurali lavorano sui tensori

- **Normalizzazione:** trasforma i valori in input in modo che siano compresi in un intervallo da [0, 1] dato che a volte ogni caratteristica, o la stessa caratteristica di campioni diversi, assume valori in scale differenti. È un caso speciale della riduzione in scala min-max applicata con la seguente formula:

$$X_{norm}^{i} = \frac{x^{i} - x_{min}}{x_{max} - x_{min}}$$

con x_{min} valore più piccolo della colonna e x_{max} valore massimo

- **Standardizzazione:** come per la normalizzazione viene usata per uniformare i valori presenti nei campioni. Utilizzata specialmente in algoritmi di ottimizzazione come la discesa del gradiente, centra le colonne delle caratteristiche sullo 0 trasformando i valori in una distribuzione normale standard con media pari a 0 e varianza unitaria.



$$X_{std}^{i} = \frac{x^{i} - \mu_x}{\sigma_x}$$

[81]

Questo processo è necessario ai fini della velocità e dell'efficacia di apprendimento in modo da evitare di fornire alla rete:

a. valori troppo diversi tra loro e dai pesi iniziali di una rete (es eccessivamente grandi)
b. non omogenei, con intervalli di valore differenti

se così non fosse si rischierebbe che gli aggiornamenti dei gradienti non riescano mai a convergere.
Un altro vantaggio della standardizzazione consiste nella sua capacità di conservare le informazioni relative ai valori anomali rendendo l'algoritmo meno sensibile ad essi.

- **Gestione valori mancanti:** può succedere che all'interno dei dati a disposizione vi siano dei valori mancanti nelle caratteristiche dei campioni e molto spesso gli strumenti di calcolo non sono in grado di gestire questi casi e/o producono risultati imprevedibili. Si può procedere in diversi modi:
  - inserire 0 se si tratta di un valore non significativo e la rete automaticamente imparerà che 0 indica un dato mancante in modo da ignorarlo
  - eliminare le righe o le colonne contenti tali valori, tutti o anche solo uno. Bisogna usare però con cautela questo secondo approccio in quanto c'è il rischio di rimuovere troppi campioni e rendere inefficace l'analisi
  - usare tecniche d'interpolazione come l'attribuzione della media per stimare i valori mancanti sostituendoli con la media calcolata utilizzando gli altri esempi presenti nel campione

- **Estrazione/manipolazione delle caratteristiche (feature engineering):** viene usata la conoscenza sui dati e sull'algoritmo utilizzato per effettuare delle trasformazioni manuali ai dati, prima che vengano visti dal modello, in modo da semplificare la rappresentazione del problema così che la rete operi al meglio, perché non sempre è in grado di apprendere bene dai dati grezzi. Prima dell'avvento del deep learning questa operazione era fondamentale perché gli spazi delle ipotesi degli algoritmi classici (superficiali) del machine learning non erano abbastanza ricchi per imparare le caratteristiche dai dati. Ora l'operazione può essere automatizzata con il deep learning anche se viene ancora usata come supporto dove possibile in quanto l'utilizzo di buone caratteristiche già definite permette di risolvere i problemi utilizzando meno risorse, meno dati e in modo più elegante

[81] S. Raschka & V. Mirjalili; Machine Learning con Python; 2020; pag. 40

- **Padding:** tecnica utilizzata nelle reti per analisi di immagini o testi, trasforma le liste di valori in modo che abbiano la stessa lunghezza e siano più facilmente analizzabili

In questa fase vengono anche codificate per categorie le etichette, vettorizzandole come un tensore di interi oppure utilizzando la codifica one hot, cioè inserendo l'etichetta in un vettore con tutti 0 tranne un 1 in corrispondenza dell'indice dell'etichetta.

I valori ricavati vengono anche utilizzati per i dati di test in quanto non si deve **mai** utilizzare questi ultimi per calcolare delle quantità ma solo come verifica.

Inoltre con il fine di determinare se il comportamento dell'algoritmo è corretto o meno, è necessario suddividere il dataset iniziale in:

- dataset di addestramento, utilizzato per l'apprendimento
- dataset di convalida, sufficientemente numeroso per evitare valutazioni errate dovute ad un'elevata varianza del set
- dataset di test, utilizzato per la verifica, perché non andrebbero mai utilizzati gli stessi dati per apprendimento e test

Le predizioni fatte dall'algoritmo "addestrato" sul dataset di addestramento e di convalida vengono verificate su quello di test.

È importante scegliere un rapporto appropriato per il partizionamento dei dati in quanto non si vuole allocare troppe informazioni nel dataset di test ma al contempo quest'ultimo non può essere eccessivamente piccolo altrimenti la stima della generalizzazione dell'algoritmo sarà inaccurata. Le suddivisioni dipendono dalla grandezza del dataset e le più comuni sono 60:40,70:30 e 80:20 ma con dataset molto grossi vengono usati anche 90:10 e 99:1.

2.2.4.3. Creazione della rete

Per la creazione della rete occorre prendere importanti decisioni:

- Quanti layer usare. L'ultimo in caso di classificazioni di dati su N classi deve essere un Dense di dimensioni N
- Quante unità nascoste per ogni layer, il loro numero non deve essere inferiore a quello delle dimensioni degli output finali in modo da evitare colli di bottiglia
- La funzione di attivazione di ogni layer. L'ultima attivazione deve essere un Softmax se si tratta di una classificazione mono-etichetta multi-classe
- Metrica di misurazione successo: utilizzata dall'utente per misurare il risultato del modello e determinarne la performance ed è scelta in base alla tipologia di problema:
    - classificazione bilanciati, i più usati sono l'accuratezza e l'errore $ACC = \frac{\sum predizioni\ corrette}{\sum predizioni}$ e $ERR = \frac{\sum predizioni\ errate}{\sum predizioni}$ cioè ACC = 1 - ERR
    - di classificazione sbilanciati (alcune classi più comuni di altre), si usano valori precision e recall
    - di valutazione o classificazione multi etichetta, mean average precision
    - di regressione, mean absolute error(mae), valore assoluto della differenza fra predizioni e target

una prassi abbastanza comune è definire delle metriche personalizzate in base ad esigenze specifiche

- La funzione obiettivo: scelta influenzata dalla metrica e dalla rappresentazione dei dati
  - categorical_crossentropy se codifica etichette One-hot
  - sparse_categorical_crossentropy se etichette codificate come interi
  - mse, mean square error, se problemi di regressione
- Un ottimizzatore: modo in cui la rete viene aggiornata in base alla funzione obiettivo

Ecco uno schema che mette in relazione il tipo di problema, la funzione obiettivo e l'attivazione dell'ultimo livello:

Tipo di problema	Attivazione ultimo livello	Funzione obiettivo
Classificazione binaria	sigmoid	binary_crossentropy
Multi classe, classificazione mono-etichetta	softmax	categorical_crossentropy
Multi classe, classificazione multi etichetta	sigmoid	binary_crossentropy
Regressione a valori arbitrari	Nessuna	mse
Regressione a valori compresi fra 0 e 1	sigmoid	mse o binary_crossentropy

[82]

2.2.4.4. Addestramento

È il core del processo con l'obiettivo di aumentare la potenza statistica, cioè sviluppare un modello in grado di battere un livello base determinato dal parametro di valutazione per la misura del successo selezionato. Il confronto con un livello base, spesso calcolato manualmente quando si affrontano problemi di cui non esiste una soluzione nota, è molto utile per esempio in compiti di classificazione non bilanciati, dove alcune classi sono più comuni di altre. Se dopo alcune iterazioni non si riesce a superare il livello minimo potrebbe essere necessario tornare a uno degli steps precedenti e rivalutare l'architettura utilizzata o i dati di input, potrebbero essere non adatti per il compito.

In caso questi due aspetti risultino corretti si può procedere:

- inserendo vincoli utili all'output per l'attivazione dell'ultimo layer
- modificando la funzione obiettivo
- scegliendo un nuovo ottimizzatore o modificando la configurazione dell'ottimizzatore attuale regolando per esempio il tasso di apprendimento

Si possono usare diversi algoritmi per ogni problema, ciascuno con i suoi pregi e i suoi difetti, ed è quindi necessario confrontarne un piccolo gruppo per determinare quale si comporta meglio nel nostro caso specifico e proseguire nell'addestramento solamente di questo. Per un confronto si deve utilizzare una metrica in grado di misurare le varie prestazioni descritta nel paragrafo successivo.

Una volta scelto il dataset di apprendimento si addestra il modello per un numero n di epoche in mini-batch da m campioni. Per essere sicuri di trovarsi nella giusta direzione contemporaneamente si monitora il valore loss e l'accuratezza tramite grafici che ne dimostrino l'andamento intercettando così tempestivamente eventuali problemi come l'overfitting, spiegato più avanti.

[82] Francois Chollet; Deep Learning con Python; 2020; pag. 115

2.2.4.5. Valutazione

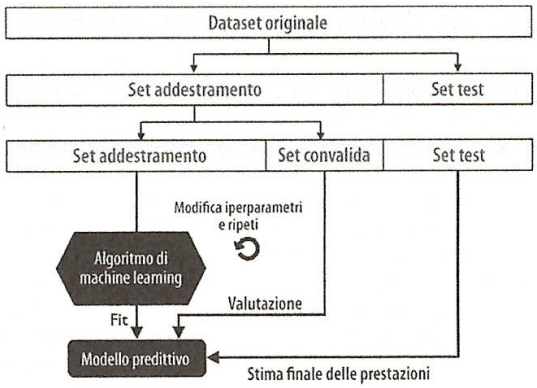
In questo step del ciclo di addestramento vengono fatte le verifiche con il dataset di test e se la valutazione è positiva, loss minimizzati in base ai criteri identificati prima, si può iniziare ad utilizzare il modello.

È sempre meglio non valutare i modelli sugli stessi dati di addestramento altrimenti si può incappare nel problema dell'overfitting. Per questo motivo una delle prassi più comuni è di suddividere i dati iniziali in un set di addestramento, uno di convalida e uno di test. Il set dei dati di convalida è usato per quello che è definito tuning della configurazione del modello: il feedback derivante da questi dati è utilizzato per determinare i valori corretti degli iperparametri della rete.

Avviene quindi un vero e proprio addestramento anche della configurazione della rete, ma in questo step può verificarsi facilmente l'overfitting sul set di convalida collegato al concetto di fuga di informazioni, l'aspetto secondo cui ogni volta che ottimizziamo un iperparametro sulla base delle prestazioni del modello sul set di dati di convalida, una parte delle informazioni entrano a far parte del modello stesso. Se questo si ripete più volte si verifica un'eccessiva fuga di informazioni dai dati con il risultato che il modello si comporterà bene sul set di convalida ma non su quelli nuovi. La valutazione finale avviene sul set di dati di test mai visti prima dal modello e se risulta ottimizzato allora sarà sufficientemente generalizzato per essere usato.

È importante quindi scegliere correttamente i set di dati di addestramento, di convalida e di test, e i principali metodi utilizzati sono:

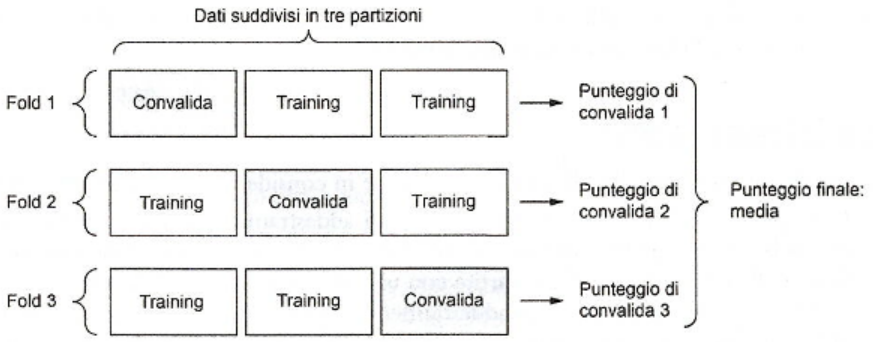
- convalida semplice holdout: prevedeva inizialmente l'"accantonamento" (si trattiene hold) di una parte dei dati come set di test, utilizzati anche per la valutazione, addestrando il modello su quelli rimanenti.



Al fine di ottimizzare efficacemente il modello è necessario però prevedere anche un set di convalida.
Il protocollo di valutazione può soffrire di un difetto quando si hanno pochi dati a disposizione: in questi casi infatti i set di convalida e test possono contenere troppi pochi campioni per essere statisticamente rappresentativi e il problema è facilmente riconoscibile quando mescolamenti casuali dei dati forniscono misure molto differenti delle prestazioni del modello
[83]

- convalida K-fold: la convalida incrociata K-fold consiste nel suddividere i dati in k partizioni di pari dimensioni, nell'istanziare K modelli identici e nell'addestrare il modello su K-1 partizioni utilizzando la rimanente per valutazione. Il punteggio di convalida finale sarà poi la media dei K punteggi ottenuti. L'approccio è utile quando le prestazioni del modello mostrano una varianza significativa e anche in questo caso è necessario utilizzare un set di convalida

[83] S. Raschka & V. Mirjalili; Machine Learning con Python; 2020; pag. 174



84

- convalida K-fold iterata con shuffling: consiste nell'applicare più volte la convalida k-fold mescolando ogni volta i dati prima di suddividerli nelle k partizioni. La valutazione finale è la media dei punteggi ottenuti in ogni iterazione k-fold. Metodo di convalida adatto alle situazioni in cui si hanno pochi dati a disposizione ed è necessario ottenere la valutazione del modello più precisa possibile, si finisce con addestrare e valutare P x K modelli dove P è il numero delle iterazioni

Per scegliere il protocollo corretto è bene considerare alcuni fattori:

- Rappresentatività dei dati: si vuole avere set di addestramento e di test il più rappresentativi possibile dei dati a disposizione, per questo motivo è consigliabile mescolare casualmente i dati per evitare pre-elaborazioni come gli ordinamenti
- Trascorrere del tempo: quando si prova a predire il futuro in base al passato, es movimenti azionari, non si deve mescolare casualmente i dati prima di suddividerli in quanto bisogna sempre assicurarci che tutti i dati del set di test siano posteriori a quelli di addestramento per evitare il crearsi di scarti temporali
- Ridondanza presente nei dati: una cosa molto comune è che alcuni punti dei dati compaiono più volte e in caso di mescolamento con successiva suddivisione questa ridondanza sarà presente anche nei vari subset compresi quelli di test che sarebbero così composti in parte anche da dati di addestramento, cosa assolutamente da evitare. È fondamentale che i set di addestramento e test rimangano disgiunti

Un'altra considerazione che viene effettuata in questo passaggio riguarda la potenza statistica del nostro modello con l'uso dei concetti di overfitting e underfitting spiegati più avanti. In poche parole è necessario fare in modo che il modello vada in overfitting:

- aggiungendo layer
- aumentando la dimensione dei layer
- svolgendo l'addestramento per molte epoche

non appena i valori di loss di addestramento e convalida tendono a peggiorare, si è raggiunto l'overfitting e si inizia a regolarizzare e ottimizzare il modello.

[84] Francois Chollet; Deep Learning con Python; 2020; pag. 99

2.2.4.6. Ottimizzazione iperparametri e regolarizzazione modello

Se il responso della valutazione non è soddisfacente si passa a una fase di ottimizzazione iperparametri e regolazione del modello. Per la spiegazione delle problematiche che si possono riscontrare e delle possibili soluzioni si rimanda al capitolo 2.3 ma indicativamente alcune delle tecniche che possono essere applicate sono:

- Aggiunta del dropout
- Modifica dell'architettura
- Aggiunta regolarizzazioni
- Cambiamento iperparametri
- Aggiungere o rimuovere caratteristiche
- Modifica della procedura di convalida qualora le prestazioni sui set di test risultano molto peggiori di quelle sui set di addestramento

Per agire direttamente sugli iperparametri però una nota tecnica di ottimizzazione è la ricerca a griglia, che ha lo scopo di individuare la combinazione ottimale dei valori degli iperparametri.

Si tratta di una ricerca esaustiva a forza bruta nella quale vengono specificati una lista di valori per gli iperparametri e il computer restituisce la combinazione di valori migliore dopo aver valutato ogni combinazione.

Questo approccio è molto costoso in termini computazionali e perciò a volte viene utilizzata la ricerca casuale che si comporta in modo simile alla ricerca a griglia me è meno dispendiosa.

2.2.4.7. Predizione

Si effettuano delle predizioni su nuovi dati, su cui vengono applicate le stesse modifiche/filtraggi utilizzati nella fase di pre-elaborazione sui dataset di addestramento.

2.3. Problemi del machine learning

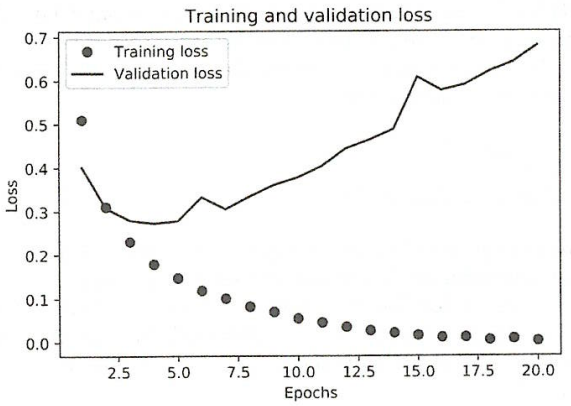
In un processo di machine learning è importante imparare a gestire il compromesso tra ottimizzazione e generalizzazione:

- **Ottimizzazione:** parte del processo che possiamo controllare in cui avviene l'aggiustamento del modello per migliorare le prestazioni di apprendimento. Rappresentata dal parametro "*bias*", un valore elevato indica che l'algoritmo non è in grado di catturare adeguatamente le relazioni tra le caratteristiche e gli output di destinazione
- **Generalizzazione:** capacità del modello addestrato di operare sui nuovi dati, rappresentato dalla varianza. Un'elevata varianza indica che il modello è troppo sensibile alle fluttuazioni nei dati, può rappresentare bene il set di dati di addestramento ma non quelli di test

Se questo non avviene con gli algoritmi di apprendimento possono emergere i seguenti problemi:

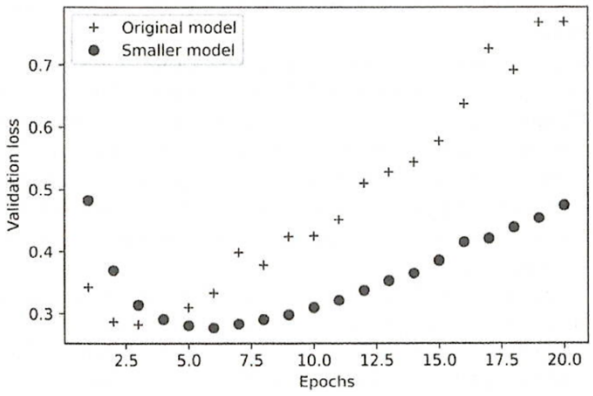
- **Underfitting(bias alto),** quando il modello non ha ancora incorporato o non è abbastanza complesso per individuare gli schemi presenti nei dati di addestramento. A inizio training l'ottimizzazione e la generalizzazione sono correlate, più basso è il valore loss sui dati di addestramento più basso è anche quello sui dati di test

- **Overfitting(alta varianza),** quando un algoritmo riesce a individuare correttamente gli schemi presenti nei dati di addestramento ma non si comporta altrettanto bene su quelli nuovi. Dopo un certo numero di epoche la generalizzazione smette di migliorare, le metriche di convalida si appiattiscono o degradano in quanto il modello inizia ad eseguire dei pattern specifici dei dati di addestramento irrilevanti su quelli di convalida/test. Per evitare che un modello impari schemi fuorvianti e/o insignificanti non necessari la migliore soluzione è di dotarsi di più dati di addestramento al fine di aumentare la capacità di generalizzazione. L'overfitting indica che il modello utilizzato è troppo complesso per il dataset di addestramento.[85]



Per cercare di limitare l'overfitting è utilizzato un processo chiamato regolarizzazione che può essere composto da diverse tecniche:

- **Ridurre le dimensioni del modello:** è la tecnica più semplice, consiste nel limitare la quantità di parametri "apprendibili", determinati dal numero di layer e unità per layer. Più parametri significa che il modello rischia di apprendere una mappatura quasi perfetta tra campioni e target, con schemi fuorvianti e superflui, facendo venire meno la capacità di generalizzazione.[86]



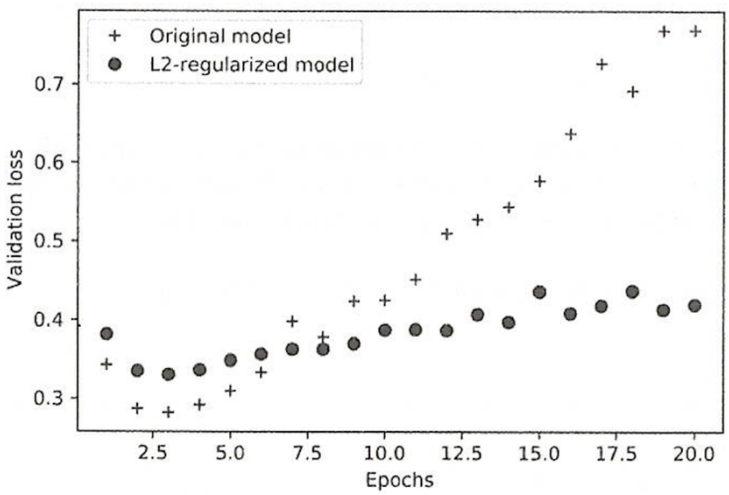
Limitare le risorse di memorizzazione costringe il modello a ricercare delle rappresentazioni compresse dei dati che hanno una maggiore capacità predittiva ma non bisogna esagerare per evitare di ricadere nell'underfit. Per questo motivo inizialmente si parte con pochi layer e parametri alla ricerca del giusto compromesso di capacità valutato sul set di dati di convalida. Si aggiungono layer e modificano le dimensioni fino a quando non si evidenzia una diminuzione del valore loss. L'insieme dei parametri "apprendibili" è chiamato capacità del modello.
Quando si hanno pochi dati di addestramento è meglio usare reti piccole in quanto anche se sono più lente nell'ottimizzazione vanno in overfitting più tardi e presentano anche un minor rumore del valore loss

- **Aggiunta alla regolarizzazione dei pesi:** applica il principio del rasoio di Occam(tra 2 spiegazioni quella più semplice solitamente è anche la più corretta) alla scelta del modello (insieme dei valori dei pesi) più adatto al compito. Quindi normalmente i modelli più semplici, ad esempio vincolando i pesi ad assumere solamente valori piccoli, tendono ad essere quelli meno soggetti all'overfit data la minor entropia nella distribuzione dei valori dei parametri.

[85] Francois Chollet; Deep Learning con Python; 2020; pag. 74
[86] Francois Chollet; Deep Learning con Python; 2020; pag. 106

La regolarizzazione dei pesi avviene aggiungendo alla funzione obiettivo un costo che va a penalizzare i valori estremi. Vi sono 2 tipi di costo:

- Regolarizzazione L1: costo proporzionale al valore assoluto dei coefficienti dei pesi
- Regolarizzazione L2(abbattimento dei pesi): costo aggiunto proporzionale al quadrato del valore dei coefficienti dei pesi solitamente nella forma $(\frac{\lambda}{2}||w||^2)$ dove λ è il parametro di regolarizzazione che viene modificato per aumentare l'aderenza del modello ai dati, se lo facciamo crescere aumenta l'intensità di regolarizzazione



[87]

Come si vede nell'immagine di esempio il modello con l'utilizzo della regolarizzazione risulta più resistente all'overfitting

- Aggiunta del dropout: viene applicato a un layer e consiste nell'azzerare in modo casuale durante l'addestramento un certo numero di caratteristiche in output per rompere le correlazioni casuali presenti nei dati di addestramento. La frazione di caratteristiche azzerate è chiamato tasso di dropout, solitamente tra il 20% e 50%. In fase di test non vengono azzerate le caratteristiche ma i valori di output vengono ridotti in scala di un fattore uguale al tasso di dropout. È una delle tecniche più utilizzate ed efficaci perché l'eliminazione casuale introduce del rumore negli output che fa diminuire gli avvenimenti casuali non significativi
- Data-augmentation: utilizzata nella visione computerizzata quando l'overfitting deriva dall'avere a disposizione pochi campioni che impediscono una corretta generalizzazione sui nuovi dati. Con il processo di data-augmentation vengono generati più dati di addestramento partendo da quelli esistenti, aumentando il campione con trasformazioni casuali che creano nuove credibili immagini. In questo modo la rete non vedrà mai lo stesso input 2 volte e potrà individuare maggiori dettagli anche se non è detto che risolva completamente l'overfitting in quanto i campioni di addestramento risulteranno comunque correlati e se questo accedesse è possibile aggiungere un layer dropout

I modelli che si generano con il machine learning devono essere più generalizzabili possibile, cioè devono essere in grado di adattarsi anche ai dati mai visti prima.

In generale è impossibile creare un modello con basso bias e bassa varianza.

[87] Francois Chollet; Deep Learning con Python; 2020; pag. 109

2.3.1. Funzioni di attivazione

Senza le funzioni di attivazione il layer potrebbe effettuare solo nelle operazioni supportate. Per esempio in quello *dense* indicato prima

$$output= dot(W, input)+b$$

avrebbe solo 2 operazioni lineari, di prodotto e somma, e il layer potrebbe apprendere solo trasformazioni affini dei dati di input: lo spazio delle ipotesi del layer sarebbe l'insieme di tutte le possibili trasformazioni lineari dei dati in uno spazio n dimensionale. Aggiungere layer in uno stack profondo non aumenterebbe lo spazio di possibilità e per questo motivo vengono aggiunte le funzioni di attivazione, o di non linearità, che permettono di accedere a uno spazio delle ipotesi più ricco.

Eccone alcune:

- Relu: REctified Linear Unit come visto prima è una funzione che azzera i valori negativi
$$relu(x)= max(x,0)$$
- Sigmoide: produce una probabilità in output, comprimendo i valori nell'intervallo [0, 1], rappresenta la probabilità con cui il campione ha un determinato target
- Softmax: restituisce un output di distribuzione di probabilità, un array di n punteggi di probabilità a somma 1. Ogni punteggio è la probabilità che un punto appartenga a una delle classi e quella con probabilità maggiore rappresenta la predizione

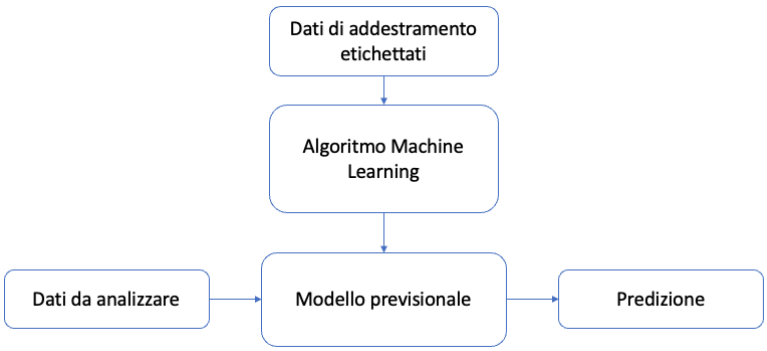
2.4. Tipologie di apprendimento

Esistono 4 diversi tipi di apprendimento per il machine learning:

- con supervisione
- con auto-supervisione
- reinforcement learning
- senza supervisione

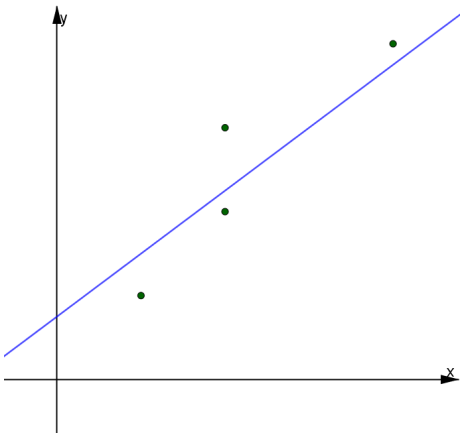
2.4.1. Apprendimento con supervisione

In questo tipo di approccio l'obiettivo è di istruire un modello per analisi e previsione partendo da dati di "addestramento" etichettati, cioè con risultati noti. Il termine "con supervisione" indica proprio questo aspetto, cioè si riferisce all'utilizzo di questi dati di esempio come base, dai quali il sistema ricava un modello previsionale la cui correttezza si verifica sui risultati già conosciuti. Una volta che l'algoritmo si ritiene corretto viene applicato su dati con risultati ancora ignoti. Quindi in concreto questo tipo di apprendimento non fa altro che mappare i dati di input sui target noti, detti etichette o annotazioni.



Esistono diverse sottocategorie di apprendimento con supervisione:

- **Compito di classificazione:** quando come dati di addestramento vengono usate delle classi discrete con l'obiettivo di assegnare ai nuovi dati inseriti delle etichette, rappresentate da categorie discrete non ordinate. Ad esempio i sistemi di filtraggio automatico della posta analizzano una serie di mail già classificate come spam e non spam e creano un modello di classificazione binaria per andare ad assegnare i nuovi messaggi a una delle 2 categorie
- **Compito di analisi a Regressione:** utilizzato per la predizione di risultati continui dove si cerca di trovare una relazione tra un certo numero di variabili descrittive predittrici, chiamate "*feature o caratteristiche"*, e una variabile di risposta continua, *"risultato o variabile target"*.



La regressione lineare è una linea tracciata tra i punti che rappresentano le relazioni che intercorrono tra una variabile delle caratteristiche X e una variabile target Y, con l'obiettivo di minimizzare la distanza quadratica media fra i punti dei dati e la linea stessa. Una volta tracciata viene usata la pendenza e l'intersezione della linea per cercare di predire la variabile target dei nuovi dati.

- **Generazione di sequenze:** predire un titolo che descriva un'immagine
- **Predizione dell'albero sintattico:** ricevendo in input una frase predice la sua decomposizione in un albero sintattico
- **Rilevamento di oggetti:** tracciare un riquadro intorno a determinati oggetti contenuti in un'immagine di input
- **Segmentazione di immagini:** data un'immagine disegnare la maschera di pixel su un oggetto

Ricapitolando le caratteristiche principali di questo tipo di apprendimento sono:

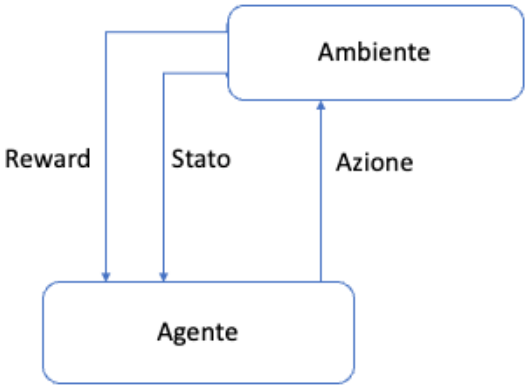
- Dati di input etichettati
- Feedback diretto
- Predizione risultato

2.4.2. Apprendimento con auto-supervisione

Variante dell'apprendimento con supervisione in cui le etichette non sono preparate manualmente ma generate dai dati di input solitamente tramite algoritmi euristici. Un esempio di questa tipologia è il cercare di predire la prossima parola di un testo sulla base di quelle precedenti.

2.4.3. Reinforcement learning

Altro tipo di machine learning tornato alla ribalta in seguito alla sua applicazione da parte di Google per l'apprendimento di giochi Atari. L'obiettivo è sviluppare un sistema, ***agente***, in grado di migliorarsi con l'interazione con l'ambiente.



L'agente recupera le informazioni di input dall'ambiente ed esegue un'azione che viene ritrasmessa all'ambiente. Quest'ultimo risponde con una misura della qualità dell'azione attraverso un segnale di ricompensa(reward) che può essere immediato o ritardato, se necessita di una serie di azioni. Tramite un approccio esplorativo, trial and error, **l'agente continua a interagire con l'ambiente con lo scopo di massimizzare la reward (reinforcement learning).**

Un esempio sono i sistemi automatici del gioco degli scacchi dove l'ambiente è la scacchiera, il risultato di una mossa è il nuovo stato dell'ambiente e la ricompensa è il risultato della partita.

Le caratteristiche principali di questo tipo di apprendimento sono:

- Processo decisionale
- Sistema di ricompense
- Apprendimento di sequenze di azioni

2.4.4. Apprendimento senza supervisione

A differenza dei tipi di apprendimento visti fino ad ora, in quello senza supervisione vengono utilizzati dei dati senza etichette e con struttura ignote. L'obiettivo è quindi l'individuazione di strutture e l'estrazione di informazioni utili tramite l'esplorazione dei dati ricevuti. Una delle tecniche utilizzate per fare questo è chiamata *clustering,* o classificazione senza supervisione: i dati vengono organizzati in sottogruppi, cluster, che contengono tutti gli oggetti individuati che presentano un certo livello di similarità, e tramite questa nuova struttura risulta più semplice definire delle relazioni tra i dati.

Un altro sottogruppo dell'apprendimento senza supervisione è la riduzione della dimensionalità: al giorno d'oggi vi è un elevato numero di dati da analizzare per ogni osservazione con sonseguenti problemi di spazio di memoria e prestazioni dei calcolatori per gli algoritmi di machine learning. L'obiettivo di questo sottogruppo è di eliminare il rumore(denoising) di fondo nei dati riducendone le dimensioni mantenendo le informazioni importanti.

Le caratteristiche principali di questa tipologia di apprendimento sono:

- Nessuna etichetta
- Nessun feedback
- Ricerca di strutture nascoste nei dati

2.5. Tipologie di reti neurali

In base al compito che dobbiamo svolgere si possono identificare delle reti neurali migliori di altre, alcune vengono descritte nei prossimi paragrafi:

- Visione computerizzata→ reti neurali convolutive, convnet
- Elaborazione testi, serie temporali e dati sequenziali→ reti neurali ricorrenti convnet 1D

2.5.1. Convnet

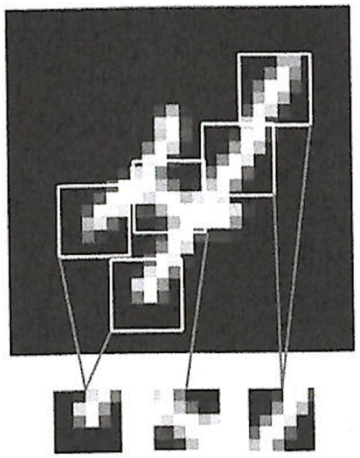
Modello di deep learning utilizzato per applicazioni di visione computerizzata e sono il miglior strumento per risolvere i problemi di classificazione di natura visuale.

Una semplice rete convnet è composta da:

- uno stack di layer di convoluzione (Conv2D) e di pooling (MaxPooling2D), chiamato base convolutiva del modello. I tensori in input hanno forma "*image_height, image_widht, image_channels"* e ciascuno layer produce in output un tensore 3D con forma *"height, widht, channels"*. La larghezza e l'altezza tendono a diminuire con la profondità della rete e l'obiettivo dello stack è arrivare ad alimentare l'ultima parte della rete
- un classificatore densamente connesso, stack di layer dense, che elaborano vettori 1D. È necessario quindi appiattire gli output da 3D in 1D prima dell'aggiunta degli ultimi layer dense classificatori

2.5.1.1. Funzionamento layer Convnet

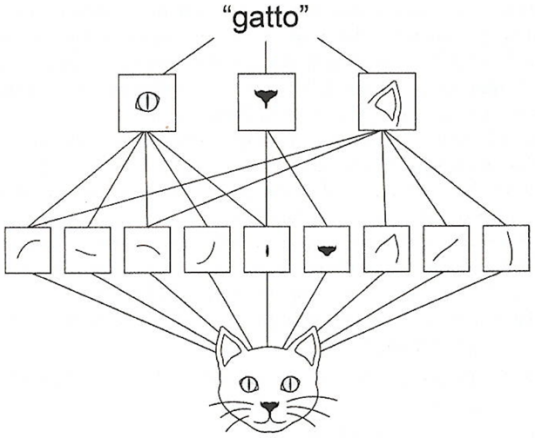
A differenza dei layer densamente connessi, che apprendono pattern globali nel loro spazio di caratteristiche di input, i layer di convoluzione apprendono pattern locali, cioè piccole finestre 2D degli input.



[88]

Questo fatto permette alle reti convnet di acquisire 2 proprietà:

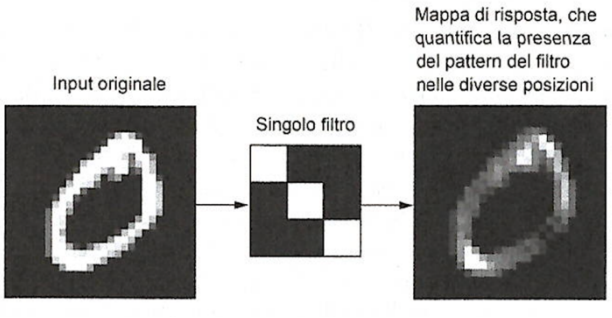
- i pattern appresi sono invarianti alla traduzione: se apprende un pattern in un angolo dell'immagine il modello è in grado di riconoscerlo anche in altre posizioni mentre una rete connessa dovrebbe apprenderlo nuovamente. Per questo motivo le reti convnet richiedono meno campioni di addestramento
- possono apprendere gerarchie spaziali di pattern: il primo layer apprende piccoli pattern locali, il secondo pattern più grandi composti da caratteristiche dei primi layer e così via permettendo al modello di apprendere efficacemente concetti visuali sempre più complessi e astratti. Ad esempio nell'immagine si vede come gli elementi locali si combinano in occhi o orecchie che a sua volta si combinano in concetti di più alto livello fino alla classificazione, gatto.[89]



[88] Francois Chollet; Deep Learning con Python; 2020; pag. 122
[89] Francois Chollet; Deep Learning con Python; 2020; pag. 123

I tensori 3D utilizzati dalle reti sono chiamati mappe delle caratteristiche, comprendono 2 assi spaziali e un asse delle profondità, detto anche asse dei canali. Ad esempio se l'immagine è RGB l'asse delle profondità ha un valore di 3, red green e blue.

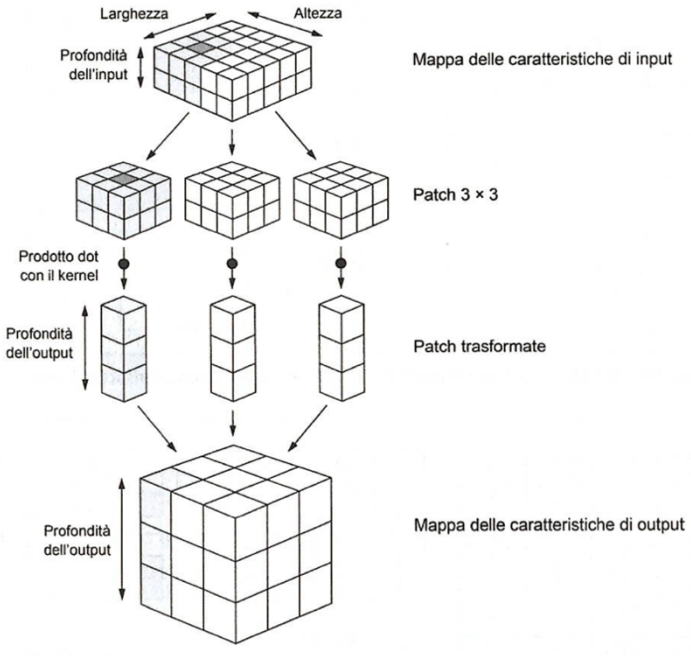
L'operazione di convoluzione estrae delle patch dalla mappa delle caratteristiche in input, applica la stessa trasformazione a tutte le patch producendo la mappa delle caratteristiche di output che risulta essere ancora un tensore 3D con altezza, larghezza e profondità. Quest'ultima però non è più legata ad una particolarità dell'immagine come il colore, bensì è un parametro arbitrario del layer che rappresenta il numero di filtri/caratteristiche utilizzati dal layer, ciascuno dei quali codifica determinati aspetti dei dati di input. Ogni tensore 3D della mappa delle caratteristiche, è quindi composto dall'asse delle profondità, e in ciascuna caratteristiche/filtro vi è un tensore 2D che è la mappa spaziale 2D della risposta del filtro sull'input.



[90]

Sono 2 i parametri fondamentali per le convoluzioni:

- dimensioni delle patch estratte dagli input: solitamente 3x3 o 5x5
- profondità della mappa della caratteristiche di output: numero di filtri calcolati dalla convoluzione



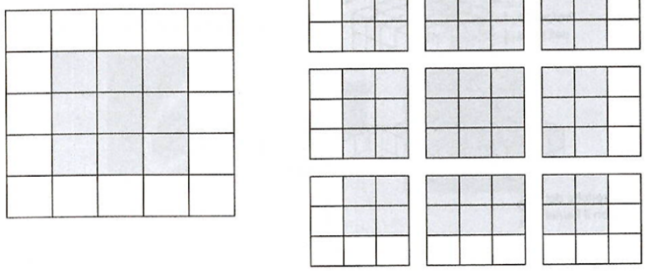
A livello pratico il sistema fa scorrere queste finestre sulla mappa 3D delle caratteristiche in input, e in ogni posizione possibile estrae la patch 3D delle caratteristiche circostanti. Ciascuna di queste con forma *"altezza, larghezza, profondità output"* viene poi trasformata in un vettore 1D con forma *"profondità output"* tramite un prodotto di tensori con la matrice dei pesi appresi, chiamata *Kernel della convoluzione*. L'insieme di questi vettori forma la mappa delle caratteristiche degli output 3D che mantiene la posizione spaziale delle caratteristiche in input.

[91]

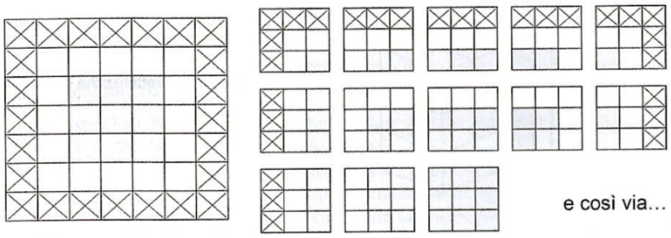
[90] Francois Chollet; Deep Learning con Python; 2020; pag. 124
[91] Francois Chollet; Deep Learning con Python; 2020; pag. 125

La larghezza e l'altezza tra output e input possono differire a causa:

- dell'effetto bordo: ad esempio se abbiamo una mappa delle caratteristiche 5x5 si avranno 25 tessere in totale. In questa però è possibile centrare una finestra 3x3 solo in 9 occasioni in cui si ottiene una mappa di output 3x3 rinunciando a due tessere in ogni dimensione, effetto bordo.



Volendo mantenere le stesse dimensioni si deve utilizzare una tecnica chiamata padding, in cui vengono aggiunte righe e colonne in modo da poter centrare le finestre di convoluzione su ogni tessera di input. [92]



- del concetto dei passi convolutivi: il passo è la distanza tra le tessere centrali di due finestre successive, di default è consigliato 1. Maggiore è il passo più i valori della mappa, altezza e larghezza, saranno sub-campionati dello stesso fattore. Per effettuare questa operazione invece di modificare il passo effettuando trasformazioni lineari, solitamente si ricorre all'operazione max-pooling, avviene nei layer MaxPooling, che sub-campiona più aggressivamente le mappe, estraendo finestre dalle mappe delle caratteristiche in input e producendo in output il valore massimo di ogni canale, tramite un'operazione max sui tensori. Il sub-campionamento è necessario perché se non venisse effettuato:
    - Si rischia di perdere dimensioni, mentre le caratteristiche fornite dall'ultimo layer di convoluzione devono contenere informazioni relative a tutto l'input
    - La mappa delle caratteristiche tende a contenere troppi coefficienti con possibili conseguenze di overfitting

2.5.1.2. Come procedere con pochi dati a disposizione

Quando non abbiamo a disposizione molti dati su cui lavorare possiamo agire in 3 modi:

- utilizzare la data-augmentation
- utilizzare una rete pre-addestrata:
    - da cui estrarre delle caratteristiche
    - su cui eseguire fine tuning

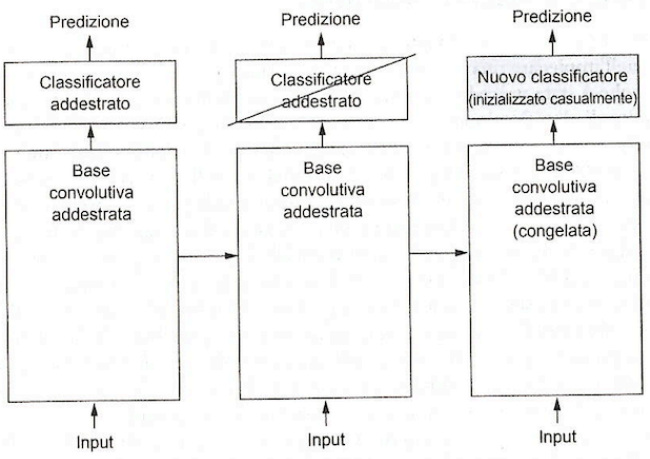
Una rete pre-addestrata è una rete salvata e addestrata in precedenza su un grosso dataset, solitamente per un compito di classificazione su immagini. Se il dataset era sufficientemente esteso e generale, la gerarchia spaziale delle caratteristiche apprese può fungere correttamente da modello generico per altri compiti, anche se questi possono riguardare classi completamente differenti. Uno dei vantaggi

[92] Francois Chollet; Deep Learning con Python; 2020; pag. 126

più importanti del deep learning è proprio la portabilità delle caratteristiche apprese su problemi differenti.

Estrazione delle caratteristiche

Utilizza le rappresentazioni apprese da una precedente rete per estrarre caratteristiche importanti dai nuovi campioni. A livello pratico sottopone i nuovi dati alla base convolutiva di una rete addestrata e istruisce un nuovo classificatore sull'output prodotto. Vengono usate le rappresentazioni apprese dalla base convolutiva perché hanno una maggior probabilità di essere generiche e riutilizzabili mentre quelle apprese dal classificatore tendono ad essere specifiche dell'insieme di classi su cui è stato addestrato il modello perdendo le informazioni relative alla posizione.



[93]

Inoltre se il nostro nuovo dataset differisce molto da quello utilizzato per l'addestramento del modello è meglio utilizzare solo pochi primi layer della base convolutiva che contengono mappe delle caratteristiche locali più generalizzabili, come spigoli colori e trame, mentre i layer più profondi estraggono concetti più astratti, come orecchio di gatto o occhio di cane.

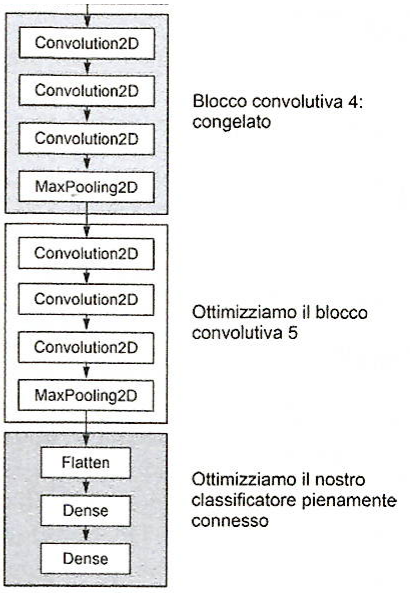
Una volta estratta la mappa delle caratteristiche viene collocato il nuovo classificatore da addestrare, 2 modi per procedere:

- eseguire la base convolutiva sul dataset e utilizzare il suo output come input per un classificatore densamente connesso indipendente. Soluzione veloce ed economica ma che non permette di utilizzare la data augmentation
- aggiungere al modello della base convolutiva i layer dense ed eseguire il risultato dall'inizio alla fine sui dati di input. In questo moto è possibile utilizzare la data-augmentation ma risulta più costosa in termini di risorse. Prima di compilare e addestrare il modello è importante congelare la base convolutiva, cioè evitare che i pesi dei layer vengano aggiornati, così da evitare che le rappresentazioni precedentemente apprese vengano modificate nell'addestramento.

[93] Francois Chollet; Deep Learning con Python; 2020; pag. 144

Fine tuning

Tecnica complementare alla precedente che consiste nello "scongelare" alcuni layer superiori (rappresentazioni più astratte) del modello base congelato, utilizzato per l'estrazione delle caratteristiche, e addestrare congiuntamente sia la nuova parte del modello(classificatore) che questi layer superiori, ma solo dopo che l'ottimizzatore è già stato addestrato altrimenti un eventuale segnale di errore si propagherà in modo eccessivo distruggendo le precedenti rappresentazioni.



I passi principali quindi sono:

- aggiungere la rete personalizzata a quella di base pre-addestrata
- congelare la base
- addestrare l'ottimizzatore aggiunto
- scongelare alcuni layer della rete di base
- addestrare congiuntamente questi ultimi e la rete personalizzata [94]

Vengono scongelati i layer superiori perché è più utile riaddestrare le caratteristiche più specializzate, che risultano quelle più importanti per il problema, e inoltre solo alcuni layer (i primi 2 o 3) per contenere il numero di parametri da addestrare in modo da limitare l'overfitting.

2.5.1.3. Visualizzare le rappresentazioni apprese da convnet

Molti considerano i modelli di deep learning delle black box che presentano le rappresentazioni apprese in una forma non comprensibile agli uomini. In alcuni casi può essere vero ma per le convnet sono stati sviluppate delle tecniche per visualizzare e interpretare le rappresentazioni, le principali:

- Visualizzazione degli output intermedi (attivazioni intermedie): utile per vedere come i vari layer trasformano gli input e capire il significato dei singoli filtri. I primi layer conservano quasi tutte le informazioni presenti nell'immagine iniziale e fungono da collettori dei vari rilevatori di spigoli. Procedendo in profondità gli output (o attivazioni) dei layer diventano sempre più astratti aumentando la sparsità delle attivazioni, cioè non tutti i filtri utilizzati inizialmente vengono "attivati" ma molti risultano vuoti, in quanto contengono sempre meno informazioni sullo specifico input visivo e sempre più informazioni sul classe target dell'immagine.
- Visualizzazione dei filtri delle convnet: utile per capire a quali pattern o concetti visuali è sensibile ogni filtro. È sufficiente definire una funzione obiettivo che massimizza il valore di un filtro di un determinato layer di convoluzione e poi utilizzare la discesa stocastica del gradiente per massimizzare la risposta di un filtro.
- Visualizzazione delle mappe termiche di attivazione delle classi di un'immagine: per comprendere come e quali parti di un'immagine conducono la convnet alla decisione finale in termini di classificazione. Utile per la localizzazione di oggetti nelle immagini e il debugging del processo decisionale della rete. Questa tipologia di tecniche è chiamata CAM, Class Activation Map, e consiste nel generare mappe termiche delle attivazioni delle classi sulle immagini di input. Queste mappe sono delle griglie 2D in cui sono associati dei punteggi a una specifica classe di output per ogni posizione dell'immagine di input.

[94] Francois Chollet; Deep Learning con Python; 2020; pag. 153

2.5.2. Reti neurali ricorrenti e convnet 1D

Utilizzate per problemi di:

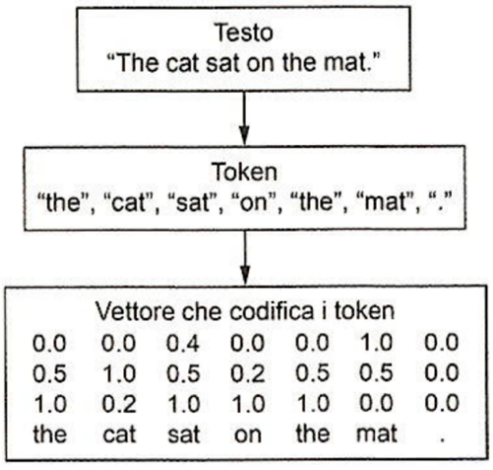
- classificazione di documenti e serie temporali, ad esempio identificando l'argomento di un articolo
- confronti tra serie temporali, stima della correlazione fra due documenti o due valori azionari
- apprendimento frase per frase, es. decodifica lingue
- analisi del sentiment, attraverso la classificazione di tweet o recensioni di film
- previsione di serie temporali, come le condizioni metereologiche

Il testo è forse la forma di dati sequenziali più diffusa e può essere visto come sequenza di caratteri o parole. Questi modelli di deep learning non sono in grado di comprendere il testo nel senso umano ma mappano le strutture statistiche della lingua con il compito di riconoscere un pattern applicato alle parole.

Come tutte le reti neurali anche questa tipologia di rete opera solo su tensori di numeri ed è quindi necessario trasformare il testo in tensori numerici tramite un processo chiamato vettorizzazione che può avvenire in diversi modi:

- suddividendo il testo in parole e trasformando ogni parola in un vettore
- suddividendo il testo in caratteri e trasformando ogni carattere in un vettore
- estrarre gruppi sovrapposti di parole o caratteri, n-grammi, e trasformare ciascun n-gramma in un vettore

La suddivisione del testo è chiamata *tokenizzazione* e ogni unità in cui viene suddiviso *token*. A ciascuno di questi è associato un vettore numerico che viene inserito in un tensore di sequenze per poi essere inviato alle reti neurali profonde.



L'associazione token- vettore può avvenire principalmente in due modi:

- codifica one-hot
- token (o word) embedding

[95]

[95] Francois Chollet; Deep Learning con Python; 2020; pag. 180

Codifica one-hot

Metodo più semplice per trasformare token in vettori, associa ad ogni parola un indice intero ***i*** univoco e poi la trasforma in un vettore binario di dimensioni N, dimensioni del vocabolario, con tutti 0 tranne l'***i***-esimo elemento 1, one-hot. Lo stesso calcolo può avvenire a livello caratteri dell'alfabeto. Si tratta quindi di vettori binari, sparsi e ad alta dimensionalità (numero di parole contenute nel dizionario) arrivando ad avere più di 20.000 dimensioni.

Molti framework di programmazione implementano utility in grado di eseguire questa operazione automaticamente eliminando caratteri e parole speciali in modo da evitare di avere spazi molto ampi per i vettori in input.

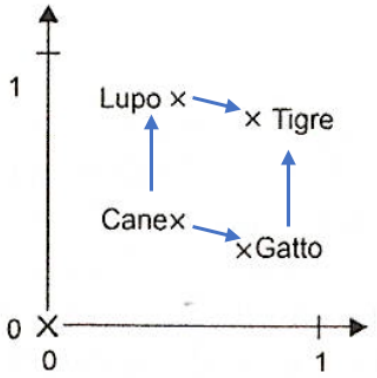
Quando il numero di token univoci del vocabolario è troppo ampio viene usata una variante chiamata one-hot-hashing trick: invece di assegnare esplicitamente un indice a ciascuna parola e mantenere i riferimenti degli indici in un dizionario, viene applicata una codifica hash delle parole in vettori di lunghezza fissa. Questo approccio, evitando di utilizzare indici espliciti, permette di risparmiare memoria e di codificare i dati, generare i vettori di token, prima di conoscere tutti i dati disponibili. Di contro però è suscettibile alle collisioni fra codici hash cioè quel fenomeno quando 2 punti dei dati, parole distinte, finiscono con l'avere la stessa codifica e in questi casi un algoritmo di machine learning non saprebbe come comportarsi. Questa probabilità si può ridurre aumentando la dimensionalità dello spazio di hashing in modo che sia molto più grande del numero totale dei token univoci codificati.

Word embedding

Potente metodo utilizzato per associare un vettore a una parola, utilizza vettori di parole densi chiamati *vettori di word embedding*. Al contrario dei precedenti questi vettori sono in virgola mobile e a bassa dimensionalità, cioè densi e non sparsi. Questi vettori sono appresi dai dati utilizzando dimensionalità di 256, 516 o 1024 e quindi sono in grado di accorpare le informazioni in meno spazio rispetto ai precedenti.

Il word embedding viene effettuato in due modi:

- **imparando assieme al compito da svolgere**, partendo da vettori di parole casuali e apprendendo nuovi vettori di parole nello stesso modo in cui si apprendono i pesi di una rete. L'obiettivo è di mappare il linguaggio umano in uno spazio geometrico, in cui le relazioni geometriche dovrebbero riflettere quelle semantiche con parole simili o sinonimi che si trovano vicine tra loro. Altro parametro importante oltre alla distanza è la direzione, vista come un vettore che mostra la tipologia di relazione tra valori, trasformazione da-a (ad esempio aggiungendo il vettore femminile alla parola principe si otterrebbe principessa).



[96]

Nell'immagine i vettori delle relazioni geometriche verticali possono indicare "da animale domestico a selvatico" mentre quelli orizzontali "da canino a felino".

[96] Francois Chollet; Deep Learning con Python; 2020; pag. 185

Partendo però da vettori casuali lo spazio di embedding non è strutturato e risulta rumoroso portando parole simili o addirittura sinonimi ad occupare posizioni molto differenti. Non esiste un qualcosa che possiamo definire linguaggio umano e la bontà di uno spazio di word embedding dipende dal compito che bisogna svolgere, per questo motivo è consigliabile apprendere un nuovo spazio di embedding ad ogni nuovo compito.
Dato che si tratta di apprendere i pesi di un *layer Embedding* si usa la retropropagazione. A livello pratico questo layer è un dizionario che associa indici interi, che rappresentano parole, a dei vettori densi.

[97]

Prende in input un tensore 2D di interi con forma *"samples, sequence_length"* dove ogni punto è una sequenza di interi ed è in grado di gestire lunghezze variabili anche se tutte quelle di un lotto batch devono avere la stessa lunghezza perché andranno inserite in un unico tensore e per questo motivo quelle più corte subiranno un padding con 0 mentre quelle più lunghe un troncamento. Il layer restituisce poi un tensore 3D in virgola mobile con forma *"samples, sequence_length, embedding_dimensionality"* che verrà lavorato da un altro layer specifico, RNN o 1D.
Inizialmente i pesi, dizionario interno dei vettori dei token, del layer Embedding saranno casuali e vengono aggiustati durante l'addestramento con la retropropagazione. Al termine lo spazio embedding sarà ben strutturato per il problema di cui si deve occupare il modello.

- **caricare un vettore di word embedding precedentemente calcolato** per un altro compito nel modello attuale, word embedding pre-addestrati. Come in precedenza i vettori pre-addestrati vengono usati quando abbiamo carenza di dati di addestramento, che risultano insufficienti per un apprendimento efficace delle caratteristiche che ci aspettiamo essere piuttosto generiche. Si caricano dunque vettori di embedding da uno spazio precalcolato, ben strutturato e con proprietà utili in grado di catturare gli aspetti generici. È possibile scaricare e utilizzare molti database di word embedding precalcolati, i più famosi sono Word2vec e GloVe.

Gli output dei layer embedding vengono poi "appiattiti" per essere utilizzate dai layer Dense finali della rete: bisogna però prestare molta attenzione in questo passaggio perché appiattire le sequenze embedded e addestrare un singolo layer dense finale porta ad avere un modello che tratta separatamente le parole della sequenza senza considerare le relazioni tra parole e la struttura della frase. Ad esempio due recensioni differenti di un film:

- questo film è una bomba
- questo film è la bomba

potrebbero essere classificate allo stesso modo come negative mentre la prima non lo è.

È sempre meglio quindi aggiungere dei layer ricorrenti RNN o convolutivi 1D sopra le sequenze embedded per imparare le caratteristiche che considerano l'intero contenuto della sequenza.

[97] Francois Chollet; Deep Learning con Python; 2020; pag. 186

2.5.2.1. Reti neurali ricorrenti

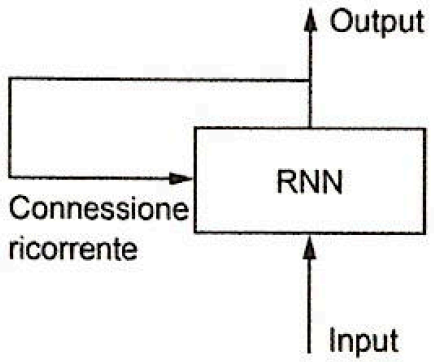
Finora sono state affrontate solo reti neurali "senza memoria" anche dette feed-forward, che elaborano i dati di input senza alcuna conservazione di stato tra i vari input. Per queste tipologie di reti quando bisogna lavorare su una sequenza o serie temporale di punti di dati è necessario fornire alla rete la sequenza intera in una volta sola, trasformandola in un singolo punto dei dati.

Al contrario l'intelligenza biologica elabora le informazioni in modo incrementale, come quando leggiamo un libro:

- conservando quello che è stato osservato
- costruendo un modello in base alle informazioni acquisite
- e aggiornandolo continuamente con le nuove informazioni in ingresso

L'obiettivo delle reti neurali ricorrenti RNN è proprio di emulare questo processo in modo semplificato:

- elabora le sequenze eseguendo iterazioni sugli elementi della sequenza
- conserva uno stato con informazioni relative a tutto quello visto in precedenza, che si resetta solo fra elaborazioni di sequenze differenti

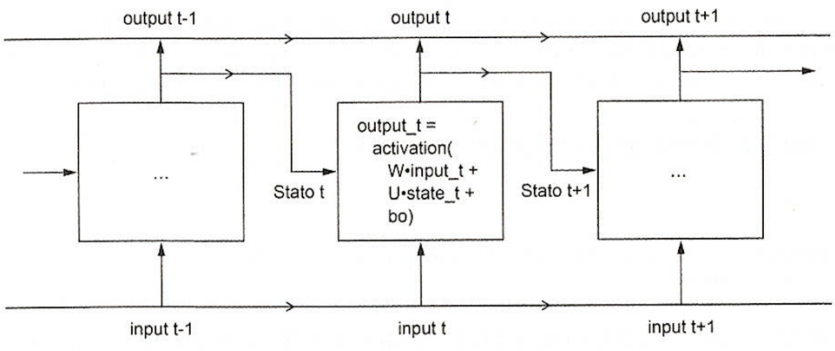


[98]

In questo modo considera ancora le sequenze come singolo punto indipendente dei dati, però la sequenza non è elaborata in un unico passo ma tramite una serie di cicli, ciascuno dei quali lavora su un elemento diverso della sequenza. L'input della RNN è una sequenza di vettori codificati come un tensore 2D di forma *"timesteps, input_features"*. L'algoritmo opera sui timestep combinando il relativo stato in ***t*** con l'input in ***t*** in modo da produrre l'output di ***t*** che diventerà l'input del ciclo in ***t+1***.

Non essendoci precedenti output per il primo timestep lo stato corrente viene inizializzato con un vettore di 0, stato iniziale della rete, che si aggiorna ad ogni iterazione assegnando allo stato (t+1) il valore dell'output di t. In pratica le RNN sono dei cicli for che riutilizzano le quantità calcolate nell'iterazione precedente caratterizzate da una funzione di passo indicata in figura:

[98] Francois Chollet; Deep Learning con Python; 2020; pag. 197



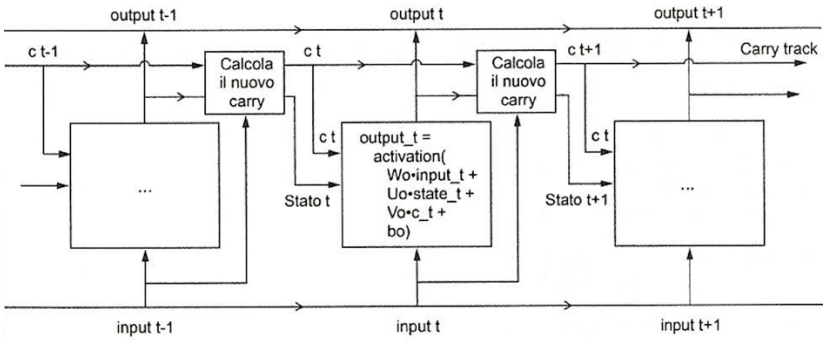
[99]

La trasformazione dell'input in output è parametrizzata da due matrici, W e U, e da un vettore di bias.

Per questi compiti si possono utilizzare diversi tipi di layer in base alla libreria che si ha a disposizione.

Ad esempio in Keras si hanno a disposizione i seguenti layers, anche se a livello pratico vengono usati principalmente il secondo e il terzo:

- SimpleRNN: elabora lotti di sequenze ricevendo tensori in input di forma (batch_size, timesteps, input_features). Può restituire o intere sequenze di output per ogni timesteps, tensore 3D di forma *"batch_size, timesteps, output_features"*, oppure solo l'ultimo output per ogni sequenza di input, tensore 2D con forma *"batch_size, output_features"*. Per aumentare la potenza di rappresentazione può essere utile alle volte sovrapporre più layer ricorrenti, in questo caso tutti i layer intermedi devono restituire l'intera sequenza degli output. Non sono adatti all'elaborazione di lunghe sequenze ed essendo troppo semplici, soffrono del problema della scomparsa del gradiente, cioè quando l'eccessivo numero di layer rende inaddestrabile la rete
- LSTM (long short-term memory): variante del SimpleRNN a cui è stato aggiunto un metodo che salva le informazioni per poterle usare in un secondo momento evitando che i segnali più vecchi vadano persi.



[100]

In pratica viene aggiunto un nuovo flusso che trasporta (carries) le informazioni ***C*** tra i vari timestep in modo da combinarle con l'input e la connessione ricorrente, tramite una trasformazione densa, per generare l'output che diventerà lo stato successivo.
Il valore carry **Ct**, al tempo ***t***, è calcolato come combinazione di 3 trasformazioni con forma di una semplice cella SimpleRNN ciascuna con le loro matrici dei pesi. A livello pratico rappresentano dei veri e propri vincoli allo scopo di:
  - eliminare le informazioni irrilevanti del flusso carry
  - aggiornare il valore con le nuove informazioni

  in modo da supportare l'algoritmo nella ricerca. La scelta di questi "vincoli" viene demandata a degli algoritmi di ottimizzazione.

- GRU (Gated Recurrent Unit): sviluppato nel 2014, utilizza lo stesso principio di quelli LSTM ma in modo più ottimizzato rendendoli più economici in termini computazionali.

[99] Francois Chollet; Deep Learning con Python; 2020; pag. 198
[100] Francois Chollet; Deep Learning con Python; 2020; pag. 205

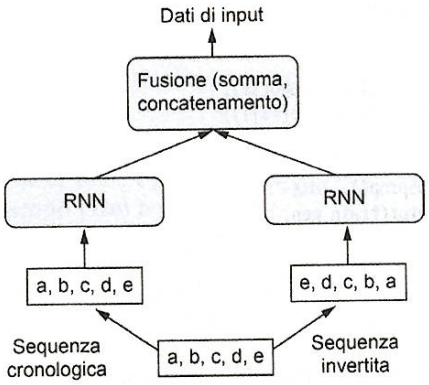
2.5.2.2. Migliorie alle reti neurali ricorrenti

È possibile con l'uso di alcune tecniche migliorare le prestazioni e la capacità di generalizzazione di questa tipologia di rete. Le principali sono 3:

- **Dropout ricorrenti:** per contrastare l'overfitting nei layer ricorrenti viene utilizzato il dropout, spiegato pocanzi, dove alcune unità di input vengono azzerate casualmente per "rompere" le correlazioni casuali presenti nei dati di addestramento. È necessario seguire alcune regole:
  - evitare di applicarlo prima di un layer ricorrente in quanto può pregiudicare l'apprendimento invece di aiutare nella regolarizzazione
  - applicare la stessa maschera di dropout (pattern di unità scartate) a ogni timestep, in questo modo la rete è in grado di propagare correttamente l'errore di apprendimento nel tempo
  - applicare alle attivazioni ricorrenti interne dei layer una maschera di dropout costante nel tempo (ricorrente) per regolarizzare le rappresentazioni formate dai passaggi ricorrenti dei layer

  Ogni layer ricorrente nella libreria Keras ha 2 parametri di dropout: dropout, numero in virgola mobile che specifica il tasso di dropout per unità di input del layer, e recurrent_dropout, tasso di dropout delle unità ricorrenti

- **Layer ricorrenti sovrapposti:** aumenta la potenza di rappresentazione della rete (maggiore carico computazionale) cercando di eliminare i colli di bottiglia della rete. La regola generale è di aumentare la capacità prestazionale della rete fino a quando l'overfitting non diventa il problema principale, nonostante si siano già prese le corrette contromisure. Se questo non diviene eccessivo probabilmente si ha un problema di limitata capacità che si prova a risolvere aumentando il numero di unità nei layer oppure aggiungendo layer.
  La sovrapposizione di più layer ricorrenti è un metodo utile per realizzare reti ricorrenti più efficaci (es. Google Translate si basa su uno stack di 7 grandi layer LSTM) ma per effettuare questo è necessario che tutti i layer intermedi restituiscano l'intera sequenza degli output tramite un tensore 3D e non solo l'ultimo
- **Layer ricorrenti bidirezionali:** varianti della RNN normale, monodirezionale, che presentano le stesse informazioni alla rete ricorrente in modi differenti, col fine di aumentare l'accuratezza e limitare i problemi di memoria sui dati meno recenti.



Utilizzati principalmente per l'elaborazione del linguaggio naturale sfrutta l'alta sensibilità dei layer RNN all'ordinamento utilizzando 2 normali RNN che elaborano la sequenza di input ciascuna in una direzione differente, cronologica e anti-cronologica, mettendo poi insieme le rappresentazioni in modo da ottenerne di più ricche.
[101]

[101] Francois Chollet; Deep Learning con Python; 2020; pag. 223

2.5.2.3. Combinazione di CNN e RNN

Sebbene nel capitolo precedente si è detto che le reti convolutive siano adatte soprattutto per risolvere problemi di visione computerizzata, possono essere usate anche per l'elaborazione delle sequenze.

Il fattore tempo può essere visto come una dimensione spaziale, come altezza e larghezza di un'immagine 2D, e le proprietà delle convnet di:

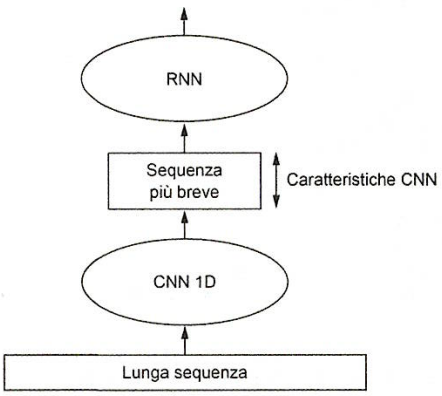
- estrarre caratteristiche da elementi di input locali
- garantire la modularità della rappresentazione
- l'utilizzo efficiente dei dati

rende le convnet 1D comparabili alle reti RNN per compiti semplici, necessitando inoltre di carichi computazionali molto inferiori.

Le convoluzioni 1D funzionano allo stesso modo delle 2D viste per le immagini, estraggono patch 1D dalle sequenze locali, individuano dei pattern interni in determinate posizioni che possono però essere riconosciuti in altre dato che a ogni patch viene applicata la stessa trasformazione di input, rendendo le convnet 1D invarianti alla traslazione.

La stessa cosa avviene con le operazioni di pooling 2D che ha un equivalente 1D, dove vengono estratti patch 1D (sottosequenze) dagli input e ne viene prodotto un output del valore massimo, max pooling, o medio, average pooling.

L'utilizzo delle convnet risulta molto utile quando i pattern sulle sequenze risultano indipendenti da dove si trovano come posizione mentre non funzionano bene in caso contrario, quando ad esempio dati recenti devono essere interpretati in modo differente da quelli più lontani perché più importanti.



In questi casi una strategia che combina la velocità e leggerezza delle reti convnet CNN, utilizzando una convnet 1D in pre-elaborazione, con la sensibilità all'ordinamento di quelle RNN può portare numerosi vantaggi. Ad esempio quando si ha a che fare con sequenze così lunghe da non poter essere elaborate con la sola RNN, la convnet 1D le trasformerà in sequenze più brevi di caratteristiche di più alto livello subcampionandole.

[102]

2.6. Algoritmi per il machine learning

Esistono numerose tipologie di algoritmi che possono essere utilizzati nel machine learning ma ai fini di questa trattazione vengono approfonditi quelli di classificazione.

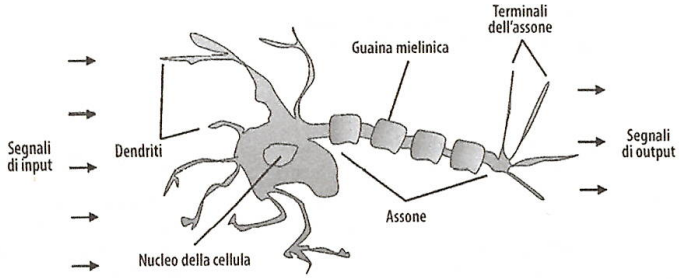
Predire dei prezzi futuri di strumenti finanziari è molto difficile basandosi solamente su serie storiche mentre è sicuramente più semplice e affidabile cercare di individuare la direzione del movimento di uno strumento finanziario, trasformando il problema in un compito di classificazione.

[102] Francois Chollet; Deep Learning con Python; 2020; pag. 231

Non esiste un algoritmo di classificazione migliore per tutte le situazioni, ciascuno ha le sue peculiarità ma tramite pratica ed esperienza si impara a scegliere quello più adatto al compito da svolgere. Questa valutazione viene fatta su ogni algoritmo in base agli errori presentati al termine dell'addestramento dove ad esempio un errore di classificazione di 1 su 100 può anche essere letto una percentuale di accuratezza del 99%.

2.6.1. Perceptron

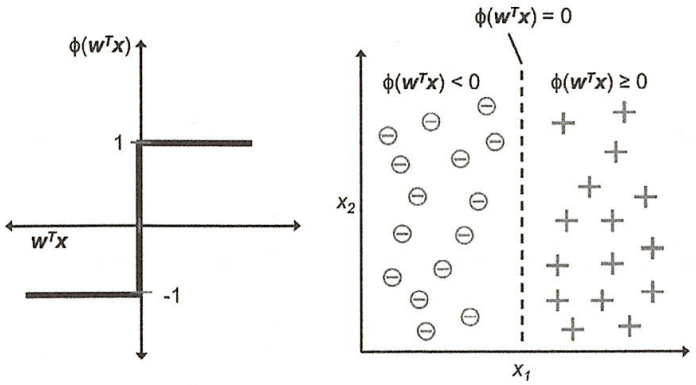
Il termine rete neurale deriva dalla neurobiologia in quanto si è preso spunto dal funzionamento del cervello umano allo scopo di creare un'intelligenza artificiale cercando di insegnare l'apprendimento a una macchina. Nel 1943 Warren McCulloch e Walter Pitts, partendo dal concetto di cervello umano composto da cellule nervose interconnesse (neuroni) che elaborano dati e comunicano con segnali chimici ed elettrici, compararono una cellula nervosa con i suoi dendriti, nucleo e assone ad una semplice porta logica con un output binario, 0 o 1, che viene attivato solo se l'accumulo dei segnali in ingresso supera una certa soglia.



[103]

Questa idea è alla base dell'algoritmo "perceptron", ideato pochi anni dopo dallo psicologo Frank Rosenblatt con il fine di creare una specie di rete neuronale artificiale in cui i neuroni reagiscono o meno agli impulsi, con l'obiettivo di riconoscere forme geometriche.

Fondamentalmente si tratta del primo algoritmo classificatore di tipo binario che vuole modellizzare un neurone artificiale e per questo contiene una serie di parametri adattivi, i coefficienti di peso, che fungono da moltiplicatori per le caratteristiche recepite. La combinazione lineare dei prodotti delle caratteristiche x e dei pesi w rappresenta l'input z del neurone sul quale è applicata una funzione decisionale $\Phi(z)$ a gradino che a seconda del valore di z, maggiore o minore di una certa soglia θ, restituisce un output di predizione 1 o -1 con cui determina se attivare un neurone o meno.



[104]

$z = w_0 x_0 + \ldots.. + w_n x_n = w^t x$

con: $w_0 = -\theta$(bias unit) e $x_0 = 1$

$$\Phi(z) = \begin{cases} 1 \text{ se } z \geq 0 \\ -1 \; altrimenti \end{cases}$$

Per apprendere e aggiornare i pesi l'algoritmo li inizializza a 0, o a piccoli numeri casuali, e poi la rete viene alimentata con un esempio alla volta. Ad ogni classificazione errata durante l'addestramento i pesi vengono aggiornati in base al valore di output con le seguenti formule:

[103] S. Raschka & V. Mirjalili; Machine Learning con Python; 2020; pag. 18
[104] S. Raschka & V. Mirjalili; Machine Learning con Python; 2020; pag. 20

$$w_j = w_j + \Delta w_j \qquad \text{con} \qquad \Delta w_j = \eta(\text{output effettivo}^i - \text{output predetto}^i)\, x_j^i$$

η=tasso di apprendimento [0, 1]

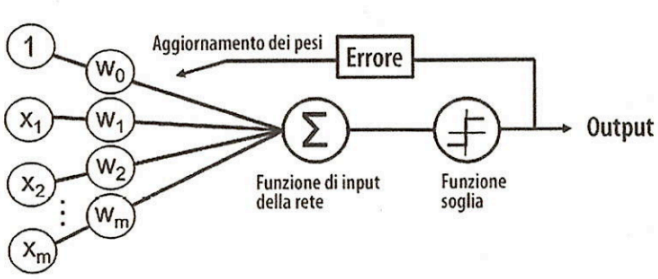
Se la predizione è corretta il valore tra parentesi sarà sempre uguale a 0 quindi il $\Delta w_j = 0$ lascia invariati i pesi, mentre se è errata vengono modificati nella direzione della classe target in modo da far convergere l'algoritmo, riducendo il numero di errori ad ogni epoca e trovando un confine decisionale definito in grado di classificare i dati.

Questa prima rete di neuroni era composta da strati un po' come i layer attuali:

- quello di input, dove i neuroni ricevono direttamente i dati in ingresso
- strati nascosti, che utilizzano come input gli output di altri neuroni
- quello di output, produce l'output della rete nel suo complesso

L'algoritmo del percettrone può essere usato per individuare i pesi di un perceptron ma quando si hanno dei multistrati per l'apprendimento è necessario utilizzare la funzione error back-propagation, retropropagazione dell'errore. Con l'obiettivo di minimizzazione dell'errore si confronta la valutazione effettiva in uscita con quella predetta e tramite la funzione a passo unitario e si aggiustano i pesi di conseguenza.

In questo tipo di algoritmo è necessario che le due classi, in cui si identificano i segnali, siano separabili linearmente e il loro tasso di apprendimento sia sufficientemente piccolo, in caso contrario bisogna definire un numero massimo di passi(epoche) che vanno effettuati sul dataset di addestramento per terminare il processo. (oppure una soglia per il numero di errate classificazioni)



Riassumendo graficamente il perceptron riceve in input gli esempi x e li combina con i pesi w per calcolare l'input della rete z. Questo viene passato alla funzione soglia $\Phi(z)$ che restituisce la predizione 1 o -1 usata per quantificare l'errore della predizione e aggiornare i pesi.

105

Nell'algoritmo perceptron, base delle attuali reti neurali, il tasso di apprendimento e il numero di interazioni per limitarne l'esplorazione sono considerati iperparametri, o anche parametri di ottimizzazione

La versione originale di questo tipo di algoritmo era molto limitata in quanto:

- l'aggiornamento dei pesi avviene su una funzione a passo unitario
- era in grado di prendere in input e produrre come output solo 2 valori, -1 e 1
- non converge mai se le classi non sono perfettamente separabili utilizzando un confine decisionale lineare, perché i pesi vengono continuamente aggiornati dato che in ciascuna epoca vi è sempre almeno un esempio classificato erroneamente

Per i limiti sopra esposti poco tempo dopo è stato ipotizzato da Bernard Widrow e Tedd Hoff un nuovo tipo di rete neurale mono-layer basato sull'algoritmo **Adaline, ADAptive Linear NEuron**.

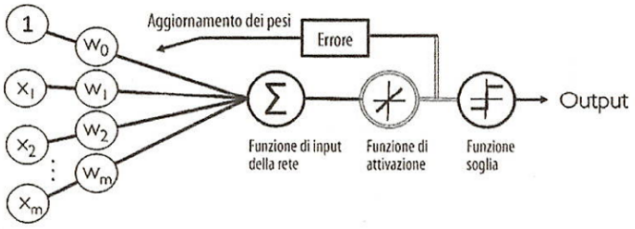
[105] S. Raschka & V. Mirjalili; Machine Learning con Python; 2020; pag. 23

2.6.2. Adaline

L' ADAptive Linear NEuron rappresentò un importante passo in avanti rispetto al Perceptron, in quanto si iniziò a utilizzare l'algoritmo di retropropagazione che impiega l'ottimizzazione a discesa del gradiente per l'addestramento di catene di operazioni parametriche. Come detto in precedenza la prima applicazione ufficiale fù attorno agli anni 90' nella rete LaNet nel servizio postale degli Stati Uniti.

In questa tipologia i pesi vengono aggiornati sulla base di una funzione di attivazione lineare che è la funzione identità dell'input della rete:

$$\Phi(w^t x) = w^t x$$



Adaptive Linear Neuron (Adaline)

[106]

A differenza del perceptron dove dopo aver valutato un singolo esempio di addestramento si confrontano le etichette effettive con quelle predette, con Adaline vengono confrontate le etichette effettive delle classi con l'output a valori continui prodotto dalla funzione di attivazione $\Phi(w^t x)$ utilizzando tutti gli esempi presenti nel dataset di addestramento, su cui viene calcolato il gradiente utilizzato per il calcolo dei pesi. Si continua in ogni caso ad utilizzare una funzione soglia per la produzione della predizione finale.

Nell'algoritmo Adaline è presente uno dei concetti chiave del machine learning con supervisione, e cioè la funzione obiettivo che deve essere ottimizzata nell'apprendimento. Solitamente si tratta di una funzione di costo continua che va minimizzata: ad esempio in Adaline può essere definita la funzione di costo J per apprendere i pesi come la somma dei quadrati degli errori fra il risultato calcolato e l'etichetta effettiva:

$$J(w) = 1/2 \sum (y^i - \Phi(z^i))^2 \qquad \text{con} \qquad y^i = \text{output effettivo}$$

Essendo differenziabile e convessa si applicava un algoritmo di ottimizzazione chiamato *discesa del gradiente batch* (spiegato nei paragrafi precedenti) per calcolare i pesi che minimizzano la funzione di costo

$$wj = wj + \Delta wj \qquad \text{con} \qquad \Delta wj = -\eta \nabla J(w)$$

facendo un passo nella direzione opposta rispetto al gradiente a ogni iterazione, calcolato come la derivata parziale per ciascun peso:

$$\partial J / \partial w_j = -\sum (y^i - \Phi(z^i))\, x_j^i$$

Per la prima volta in questo approccio venivano anche introdotti:

- il tasso di apprendimento, che come visto in precedenza deve cercare di essere ne troppo ampio ne troppo piccolo
- la *standardizzazione* delle caratteristiche in modo da velocizzare la convergenza

[106] S. Raschka & V. Mirjalili; Machine Learning con Python; 2020; pag. 33

2.6.3. La regressione lineare

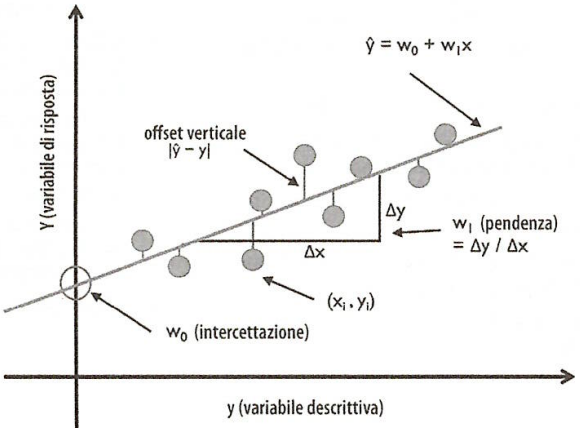
I modelli basati sull'analisi a regressione vengono utilizzati per predire variabili target su una scala continua. Molto utili in ambito scientifico in quanto permettono di comprendere le relazioni tra le variabili, valutare le tendenze ed effettuare previsioni.

L'obiettivo della regressione lineare è quello d modellare la relazione fra una o più caratteristiche e una variabile target continua cercando di predire gli output su una scala continua invece che su delle etichette. Non sono dei classificatori ma vengono trattati in questo capitolo in quanto ai fini delle previsioni in ambito di strumenti finanziari è utilizzata per predire la direzione del mercato e alla fine i suoi risultati sono convertiti in 2 soli valori -1 e 1.

Il tipo più elementare di regressione è la regressione lineare semplice, univariata, che modella la relazione tra una singola caratteristica, variabile descrittiva x, e una variabile target a valore continuo y.

$$y = w_0 + w_1x$$

Il peso w_0 rappresenta l'intercetta dell'asse y e w_1 è il coefficiente del peso della variabile descrittiva. Ai fini della modellazione l'obiettivo finale della regressione è quello di apprendere i pesi dell'equazione che meglio descrivono la relazione fra la variabile descrittiva e quella target in modo da poterli utilizzare per predire le risposte a future variabili descrittive non presenti nel dataset iniziale.



[107]

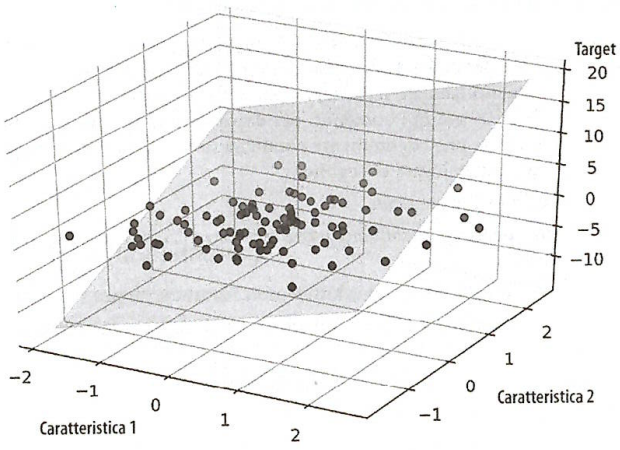
A livello grafico può essere rappresentata come la miglior linea retta, *best fit*, che attraversa gli esempi di addestramento con le linee verticali che la congiungono agli esempi che raffigurano gli errori della predizione, residui o offset.

Quella che però viene utilizzata nell'analisi degli strumenti finanziari solitamente è la versione generalizzata con più variabili descrittive, ***la regressione lineare multipla.***

$$Y = w_0 + w_1x_1 + \ldots + w_mx_m = \sum_{i=0}^{n} w_i x_i = w^T x$$

Il peso w_0 rappresenta sempre l'intercetta dell'asse y con $x_0 = 1$.

[107] S. Raschka & V. Mirjalili; Machine Learning con Python; 2020; pag. 282



A livello grafico per esempio in caso si avessero due caratteristiche può essere rappresentato attraverso un iperpiano bidimensionale come in figura.

108

Per questa tipologia di algoritmo può essere utile visualizzare sotto forma di grafici o con una matrice di correlazione, le relazioni lineari che intercorrono tra le coppie di variabili in modo da configurare il modello considerando solamente quelle caratteristiche che risultano avere un'elevata correlazione con la variabile target.

Il calcolo dei parametri della regressione può avvenire applicando l'algoritmo di ottimizzazione discesa stocastica del gradiente sulla funzione di costo utilizzata per i minimi quadrati vista nel modello Adaline senza la funzione a passo unitario in modo da avere valori target continui.

$$J(w)=1/2\ \sum(y^i-\Phi(z^i))^2$$

Un'ottima misura quantitativa per determinare la bontà del modello è il cosiddetto errore quadratico medio, mean square error:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y^i - \Upsilon^i)^{\,2}$$

2.6.4. La regressione logistica

Algoritmo di modello probabilistico per la classificazione lineare e binaria che introduce il concetto di *odd* cioè le probabilità in favore di un determinato evento:

$\frac{p}{(1-p)}$ p= probabilità dell'evento che vogliamo predire

Applicando il logaritmo naturale a questo rapporto(log-odds) otteniamo la funzione *logit*:

$$logit(p) = log\,\frac{p}{(1-p)}$$

Che trasformando i valori di input compresi tra 0 e 1 in valori sull'intervallo dei numeri reali, può essere usata per esprimere una relazione lineare tra i valori delle caratteristiche e log-odds

$$logit\big(p(y=1|x)\big) = \sum_{i=0}^{m} w_i x_i \;=\; w^t x$$

Con $p(y=1|x)$ probabilità condizionata al fatto che un determinato esempio appartenga alla classe 1 in base alle sue caratteristiche x.

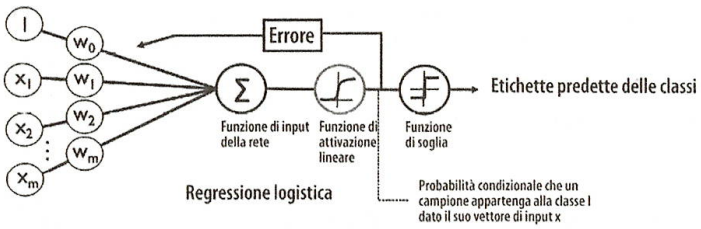
[108] S. Raschka & V. Mirjalili; Machine Learning con Python; 2020; pag. 283

In alcune applicazioni si è interessati a predire la probabilità che un esempio appartenga a una determinata classe utilizzando la forma inversa della funzione logit chiamata *funzione sigmoide logistica:*

$$\phi(z) = \frac{1}{1+e^{-z}} \text{ con dominio [0,1]}$$

dove z è l'input della rete uguale a $w^t x$ e l'output è interpretabile come la probabilità che un determinato esempio appartenga a una classe o meno in base alle sue caratteristiche parametrizzate per i loro pesi, es $p(y = 1|x; \boldsymbol{w})$.

Pertanto questa funzione altro non è che la corrispondente funzione di attivazione dell'algoritmo Adaline, prende in input dei valori reali e li trasforma in valori compresi nell'intervallo [0,1] che si possono convertire in un risultato binario per la stima della probabilità di appartenenza a una classe, tramite la funzione soglia:



$$y = \begin{cases} 1 \; se \; z \geq 0 (\; \phi(z) \geq 0{,}5) \\ 0 \; altrimenti \end{cases}$$

[109]

Grazie a questa sua caratteristica la regressione logistica è molto utilizzata in campo medico per determinare la probabilità che un paziente abbia una determinata malattia sulla base dei sintomi presentati.

Per apprendere i pesi corretti derivando la funzione di costo è necessario definire innanzi tutto la probabilità L che si vuole massimizzare in un modello di regressione logistica. Risulta più facile però massimizzare il logaritmo naturale di questa equazione, funzione *log-likelihood,* che riduce anche i rischi di underflow numerico con probabilità molto piccole, cioè tutti quei casi di errore dove un risultato è troppo piccolo per essere rappresentato con i bit a disposizione e viene confuso con il valore 0.

$$\log L(w) = \sum_{i=0}^{n} [y^i \log\left(\phi\left(z^i\right)\right) + \left(1 - y^i\right) \log\left(1 - \phi\left(z^i\right)\right)]$$

Ora si può utilizzare un algoritmo di ottimizzazione, come l'ascesa del gradiente, per massimizzare la funzione o riscrivendo la funzione *log-likelihood* come una funzione di costo J da minimizzare con la discesa del gradiente.

$$\mathrm{J}(w) = \sum_{i=0}^{n} [-y^i \log\left(\phi\left(z^i\right)\right) - \left(1 - y^i\right) \log\left(1 - \phi\left(z^i\right)\right)]$$

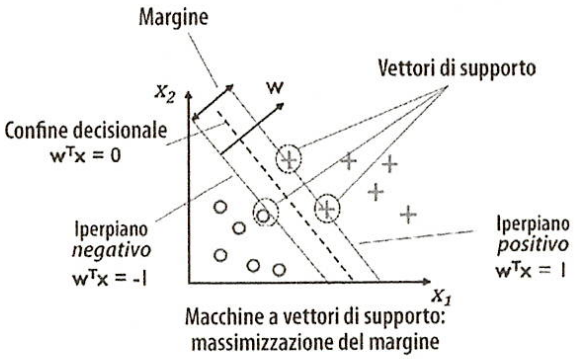
$$\mathrm{J}(\phi(z), y, w) = -y \log\big(\phi(z)\big) - (1 - y) \log\big(1 - \phi(z)\big) = \begin{cases} -\log\big(\phi(z)\big) \; se \; y = 1 \\ -\log\big(1 - \phi(z)\big) \; se \; y = 0 \end{cases}$$

Con queste operazioni il costo si avvicina a 0 se la predizione è corretta mentre se la predizione è errata tende all'infinito.

[109] S. Raschka & V. Mirjalili; Machine Learning con Python; 2020; pag. 57

2.6.5. SVM, Support Vector Machine

Considerato come un'estensione del perceptron con la differenza che invece di minimizzare gli errori di classificazione cerca di massimizzare il margine, cioè la distanza tra l'iperpiano decisionale (linea di separazione dei valori) e gli esempi di addestramento, vettori di supporto, che più si avvicinano a questo.



L'idea alla base è che con grandi margini si hanno bassi errori di generalizzazione.

[110]

Dopo una serie di calcoli si arriva a determinare la formula del margine, distanza iperpiano positivo da quello negativo, da massimizzare come $\frac{2}{||w||}$, oppure $\frac{1}{2}\left||w|\right|^2$, con i seguenti vincoli:

$$w_0 + w^t x^i \geq 1 \; se \; y^i = 1$$

$$w_0 + w^t x^i \leq -1 \; se \; y^i = -1$$

che indicano che ogni esempio, positivo o negativo, deve ricadere oltre i rispettivi iperpiani.

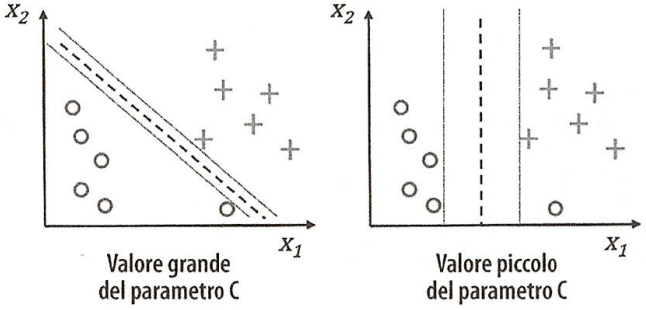
I vincoli lineari per dati non separabili linearmente devono però essere "allentati" a volte in modo da consentire la convergenza dell'ottimizzazione anche con errate classificazioni, considerando delle penalizzazioni in termini di costi.

Per questo motivo viene introdotta la variabile di slack ξ aggiungendola ai vincoli lineari e alla funzione da minimizzare:

$$w_0 + w^t x^i \geq 1 - \xi^i \; se \; y^i = 1$$

$$w_0 + w^t x^i \leq -1 + \xi^i \; se \; y^i = -1$$

$$\frac{1}{2}\left||w|\right|^2 + c(\Sigma\xi^i)$$

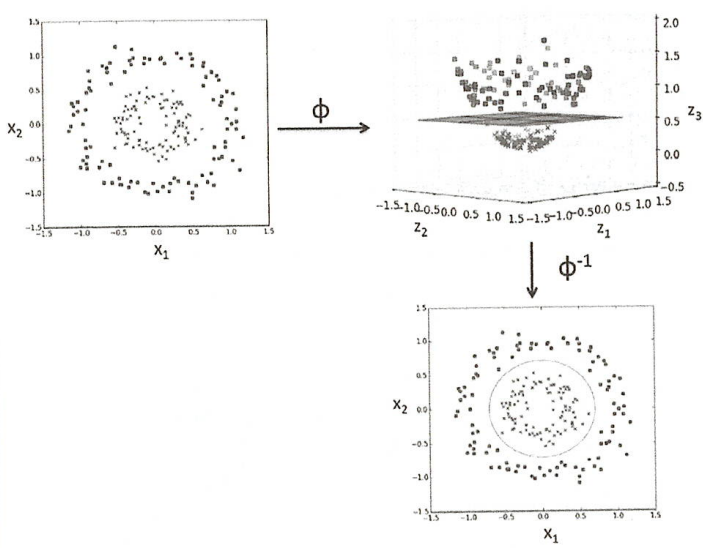


C determina in modo direttamente proporzionale la penalizzazione per errata classificazione: più è grande più saranno penalizzati gli errori e viceversa. Come si vede dall'immagine C controlla la larghezza del margine, regola il compromesso tra bias e varianza.[111]

Per la risoluzione di problemi di classificazione non lineari dove non è possibile separare adeguatamente gli esempi viene "Kernelizzato" il classificatore che prenderà il nome SVM Kernel. Il termine indica che per cercare di identificare un buon confine decisionale vengono create combinazioni lineari delle caratteristiche non lineari e proiettati i risultati attraverso una funzione di

[110] S. Raschka & V. Mirjalili; Machine Learning con Python; 2020; pag. 70
[111] S. Raschka & V. Mirjalili; Machine Learning con Python; 2020; pag. 72

mappatura ϕ su uno spazio con maggiore dimensionalità in cui i dati possano essere separati linearmente.



[112]

come si vede nell'esempio è possibile così individuare un piano decisione lineare su cui un modello SVM può essere addestrato per classificare i dati in questo nuovo spazio.

Una controindicazione di questo approccio sta nel fatto che la creazione delle nuove caratteristiche in spazi con maggior dimensionalità richiede molta potenza di calcolo.

2.6.6. Alberi decisionali

I classificatori ad albero decisionale sono modelli utilizzati per suddividere i dati prendendo una decisione sulla base di una serie di domande.

[113]

L'algoritmo decisionale parte dalla radice dell'albero, suddivide i dati in base alla caratteristica che produce il maggior *guadagno informativo (IG- information gain)* e ripete questa operazione ad ogni nodo dell'albero fino al presentarsi di sole foglie pure, cioè quando tutti gli esempi di addestramento appartengono alla stessa classe. Per evitare il proliferare di nodi/foglie rischiando di ricadere in overfitting, solitamente viene determinata una profondità massima dell'albero.

Per suddividere i nodi in base alle caratteristiche più informative è necessario definire una funzione obiettivo da ottimizzare, massimizzazione dell'IG, tramite l'algoritmo di apprendimento ad albero.

$$IG(D_p, f) = I(D_p) - \sum_{j=1}^{m} \frac{N_j}{N_p} I(D_j)$$

f= caratteristica utilizzata per la suddivisione
D_p *e* D_j= dataset del nodo genitore(p) e del j-esimo figlio
I= misura di impurità
N_p *e* N_j= numero totale di esempi di addestramento al nodo padre e al j-esimo nodo figlio

[112] S. Raschka & V. Mirjalili; Machine Learning con Python; 2020; pag. 76
[113] S. Raschka & V. Mirjalili; Machine Learning con Python; 2020; pag. 80

Il guadagno informativo altro non è che la differenza tra l'impurità del nodo genitore e la somma delle impurità dei nodi figli ed è tanto maggiore quando sono basse queste ultime.
Per semplicità e vantaggi di calcolo vengono utilizzati degli alberi decisionali binari in cui ogni nodo padre viene suddiviso in 2 nodi figli e su questa tipologia viene effettuata la misura dell'impurità con uno dei seguenti 3 criteri:

- L'entropia I_H: calcolata per tutte le classi non vuote dove p(i|t) è la proporzione degli esempi per il nodo t che appartengono alla classe i

$$I_H(t) = -\sum_{i=1}^{c} \mathrm{p(i|t)}\, log_2 \mathrm{p(i|t)}$$

  Sarà uguale a 0 ($\mathrm{p(i = n|t)} = 1$ o 0) se tutti gli esempi del nodo appartengono alla stessa classe ed è massima, 1, se vi è una distribuzione uniforme tra le classi ($\mathrm{p(i = n|t)} = 0{,}5$). Questo criterio tenta di massimizzare la distribuzione.

- L'impurità di Gini I_G: questo criterio cerca di minimizzare la probabilità di un'errata classificazione ed è massimo se le classi sono perfettamente miscelate.

$$I_G(t) = 1 - \sum_{i=1}^{c} \mathrm{p(i|t)}^2$$

- L'errore di classificazione I_E: criterio utile per la "potatura".

$$I_E(t) = 1 - \max\,[p(i|t)]$$

Una volta scelto un criterio si può costruire un albero decisionale su cui è possibile tracciare confini decisionali, determinati dall'albero e rappresentati da rettangoli contenenti le caratteristiche, partendo dal dataset di addestramento.

Gli alberi decisionali presi singolarmente soffrono di un'elevata varianza e sono molto sensibili all'overfitting, per questo motivo con il fine di costruire un modello più solido in grado di offrire migliori prestazioni di generalizzazione è stato creato l'algoritmo a *foresta casuale*.

Facile da utilizzare e dotato di buona scalabilità mette assieme una serie di alberi decisionali:

- Addestra ciascun albero scegliendo casualmente gli esempi dal dataset e le caratteristiche, utilizzando poi quella che offre un maggior vantaggio per calcolare la predizione
- Assegna l'etichetta della classe in base una votazione a maggioranza delle predizioni

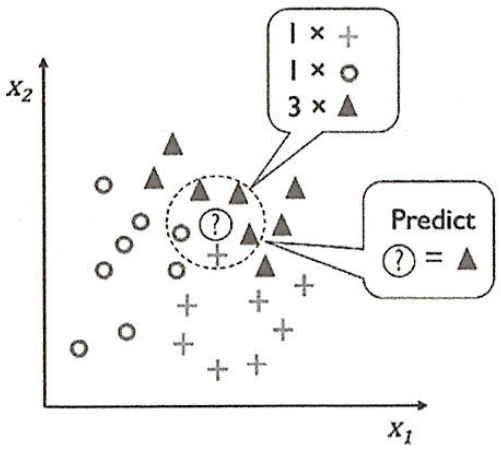
Questo processo limita il rumore contenuto nei singoli alberi e permette di non preoccuparsi troppo dei vari iperparametri, come la dimensione del campione o il numero delle caratteristiche. L'unico vero parametro importante da considerare è proprio il numero di alberi anche se solitamente maggiore è meglio è.

2.6.7. K-Nearest Neighbor

L'algoritmo K-Nearest Neighbor (KNN), utilizzato nell'apprendimento con supervisione, è un tipico esempio di lazy learner. Lazy, pigro, perché non apprende una funzione discriminativa dai dati di addestramento ma memorizza il dataset di addestramento.

I passi dell'algoritmo sono piuttosto semplici:

- Scegliere il numero di k e una metrica per la distanza
- Trovare i k vicini più prossimi fra i record dei dati da classificare
- Assegnare l'etichetta della classa in base a un voto a maggioranza



[114]

In base alla metrica per la distanza scelta e al numero k di esempi del dataset più prossimi e simili al nuovo punto "?", nell'immagine viene assegnato l'etichetta Δ che risulta in maggioranza fra i punti più vicini.

Il vantaggio di questo approccio è che il classificatore si adatta immediatamente a nuovi dati di addestramento.

Lo svantaggio è che nei casi peggiori la complessità computazionale per classificare i nuovi esempi cresce in modo lineare con il numero di esempi del dataset di addestramento. Questo problema può essere limitato con dataset di poche dimensioni/caratteristiche o quando l'algoritmo implementato utilizza una struttura efficiente per la memorizzazione.

Infine non potendo scartare degli esempi di addestramento non essendoci nessun passo di addestramento, lo spazio di memoria diventa un problema con grossi dataset.

Se nella risoluzione dell'algoritmo il voto a maggioranza dà come risultato un pareggio il KNN prediligerà i vicini con minor distanza dal punto da classificare.

K è fondamentale per dare al nostro algoritmo un buon equilibrio tra overfitting e underfitting mentre la metrica deve essere appropriata per le caratteristiche presenti nel dataset.

[114] S. Raschka & V. Mirjalili; Machine Learning con Python; 2020; pag. 92

3. Sistemi di trading informatici

In questo capitolo verranno presentati i risultati delle implementazioni informatiche di alcune delle tecniche viste in precedenza. Ogni tecnica valutata è stata testata sulle seguenti azioni o indici azionari (media ponderata di diversi titoli inclusi nel paniere dell'indice):

- **FTSE MIB:** al momento contiene 40 società quotate sulla Borsa italiana, selezionate in base alla capitalizzazione, al volume degli scambi e al settore
- **S&P500:** è il più importante indice azionario nordamericano, include le azioni delle principali 500 grandi aziende contrattate al New York Stock Exchange (NYSE), all'American Stock Exchange (AMEX) e al NASDAQ, selezionate da un apposito comitato
- **Dow Jones:** 30 titoli azionari principali quotati sul mercato statunitense scelti dalla redazione del "The Wall Street Journal"
- **NASDAQ:** National Association of Securities Dealers Automated Quotation è invece l'indice dei principali titoli tecnologici della Borsa Americana
- **DAX:** 30 titoli a maggior capitalizzazione della Borsa di Francoforte in Germania
- **FTSE 100:** 100 principali società quotate al London Stock Exchange
- **Nikkei:** contiene i titoli delle maggiori 225 compagnie quotate alla Borsa di Tokyo (TSE)
- **Apple**
- **Stellantis**
- **Banca Generali**
- **Enel**
- **Intesa Sanpaolo**

Gli intervalli di tempo considerati sono giornalieri e si è deciso di utilizzare i dati degli ultimi 10 anni suddivisi come segue:

- 01/01/2012 - 01/01/2020, in modo da verificare le performance prima della pandemia Covid19 e dell'attuale conflitto in corso
- 01/01/2012- 01/10/2022, arco di 10 anni completo in modo da constatare eventuali conseguenze di pandemia e guerra
- 01/01/2017- 01/10/2022, ultimi 5 anni

Per non rendere troppo lunga la trattazione vengono inseriti solo i risultati grafici dell'indice FTSE MIB mentre tutti i risultati sono presentati al termine di ciascun paragrafo in forma tabellare.

I risultati vengono poi comparati con un benchmark di riferimento che per semplicità sarà il classico keep and hold.

3.1. Python

Il linguaggio di programmazione utilizzato per gli sviluppi è Python. Nato nel 1991, nel 1994 fu rilasciata la versione 1.0 ma solo dopo due decenni, dal 2011 in poi, ha iniziato a essere usato su larga scala affermandosi come uno dei principali linguaggi di programmazione nel settore finanziario grazie alla sua sintassi molto chiara e concisa che permette di svolgere compiti anche complessi con poche righe di codice.

Una svolta importante in questa direzione è avvenuta nel 2006 con il rilascio della prima versione del package NumPy(numerical Python) che ha fornito un approccio più efficace e veloce nell'uso di

strutture dati ai fini del calcolo, soprattutto grazie al concetto di vettorizzazione che vedremo più avanti. In questo modo è possibile scrivere codice più compatto e fino ad 8 volte più veloce degli approcci classici nella risoluzione di problemi relativi all'analisi dati. Ha quindi subito trovato una naturale applicazione nel trading algoritmico, dove la gestione di dataset di serie storiche finanziarie di enormi dimensioni risulta fondamentale, soprattutto:

- nelle sessioni di backtesting, che permettono di valutare le strategie di trading prima di implementarle
- nell'elaborazione dei flussi dati real time. Qui entra in gioco un ulteriore package specifico per l'analisi Pandas, il cui sviluppo è iniziato nel 2008 per poi essere reso open source dal 2009.

3.2. Trading algoritmico

Con il termine "trading algoritmico" ci si riferisce al trading di strumenti finanziari basati su algoritmi formali, cioè un insieme di operazioni svolte in sequenza per raggiungere un determinato obiettivo. Quest'ultimo è chiaro e univoco quando si tratta di risolvere un'equazione ma in ambito finanziario c'è ne possono essere diversi che corrispondo a diverse strategie di trading:

- Beta trading: guadagnare per il rischio di mercato investendo ad esempio in ETF, Exchange Traded Fund, che replicano le performance del S&P500
- Alpha generation: guadagnare per rischi indipendenti dal mercato, ad esempio vendendo azioni allo scoperto quotate nello S&P500 o in ETF collegati. Queste strategie vengono valutate in base al loro rendimento nei confronti di un benchmark nello stesso periodo, ad esempio rispetto allo S&P500.
- Static hedging: copertura contro i rischi di mercato acquistando opzioni
- Dynamic edging: copertura contro i rischi di mercato delle opzioni operando sui futures e altri strumenti di liquidità. Il primo algoritmo di questa tipologia, delta hedging, è stato pubblicato nel 1973 da Black, Scholes e Merton mostra come coprire tutti i rischi di mercato in un mondo ideale, semplificato e continuo, ma è risultata efficace anche nella realtà tanto è vero che è ancora utilizzata
- Asset-liability management: negoziazione di azioni ed ETF per coprire passività derivanti per esempio da stipula assicurazioni sulla vita
- Market making: fornire liquidità alle opzioni acquistando e vendendo a prezzi bid e ask

Questi sono solo alcuni degli esempi di operazioni che possono essere implementati con approcci discrezionali, con algoritmi che supportano e in alcuni casi sostituiscono l'essere umano nelle decisioni.

Esistono gestori di portafogli che basandosi principalmente su esperienza e ricerca sono riusciti a battere i benchmark per lunghi periodi di tempo, es. Warren Buffet, ma statisticamente la maggior parte dei gestori attivi non ci riesce: come quando nel 2015 lo S&P500 ha avuto un rendimento del 1,4% e il 66% dei fondi azionari gestiti attivamente non è stato in grado di fare meglio, la percentuale poi sale addirittura al 82% se si guarda periodi di 5-10 anni. Uno studio del 2016 ha comparato i due approcci, discrezionali e sistematici, ed è arrivato alla conclusione che gli hedge fund sistematici azionari battono i corrispettivi discrezionali in merito al rapporto di stima del rendimento però per esempio in un anno come il 2017 han reso solamente l'8,5% rispetto al 21,8% dello S&P500.

Nell'ambito di questa trattazione verranno utilizzate strategie di trading definite Alpha Seeking, che hanno l'obiettivo di generare rendimenti positivi e superiori al mercato indipendentemente dalla direzione di quest'ultimo. Ogni strategia tratterà un solo strumento finanziario alla volta, indice azionario o singola azione, i cui segnali sono derivati da dati temporali strutturati e non da altre fonti come news e social media.

Le quattro strategie che verranno implementate sono:

- Simple moving average: basata su medie mobili semplici SMA per generare segnali di posizionamenti nel mercato. A livello generale si ritiene un segnale di acquisto quando una media breve incrocia al rialzo una media di più lungo periodo e di vendita in caso contrario
- Momentum: si basa sul concetto di momentum, che prevede che uno strumento finanziario si comporti in conformità con le sue performance recenti
- Mean Reversion: basata sull'idea che uno strumento finanziario tende ad avvicinarsi a un valore medio tanto più si trova lontano da esso
- Machine e deep Learning: vengono addestrati degli algoritmi di deep learning principalmente attraverso l'osservazione delle serie storiche dei rendimenti, con il fine di predire i movimenti del mercato azionario. Quelli testati sono
    - la regressione lineare
    - machine learning: regressione logistica
    - deep learning: reti neurali

3.3. L'utilizzo dei dati

Senza i dati nessun algoritmo potrebbe avere successo o può essere valutato. Generalmente nello sviluppo di applicativi finalizzati al trading algoritmico si ha a che fare con 4 tipi di dati, riassumibili nella tabella seguente:

	Strutturati	Non Strutturati
Storici	Prezzi di chiusura	Articoli di notizie
Real-time	Prezzi bid/ask FX	Post Twitter

115

Come si evince dalla tabella la tipologia dei dati è definita dal tempo, se storici o real time, e dalla struttura, strutturati o meno, e ciascuna deve essere trattata in modo differente.

A prescindere dell'implementazione è sempre necessario avere a disposizione dei dati storici, meglio se strutturati, con cui eseguire i cosiddetti backtest per la valutazione dell'algoritmo utilizzato.

Nello sviluppo di un sistema i dati la fanno da padrone lungo tutto il processo in quanto:

- Il primo passo consiste proprio nella loro importazione seguita dalla lettura dei dati finanziari di interesse, che possono essere recuperati da file o siti web
- Nella fase intermedia questi devono poi essere gestiti ed elaborati nel modo migliore possibile
- Come ultimo step devono essere archiviati nel modo più efficiente possibile

Per svolgere questi tre punti in Python viene solitamente utilizzata la libreria "pandas", adatta soprattutto alla lettura dati di diverse fonti come CSV, XLXS, Json, servizi web, etc….

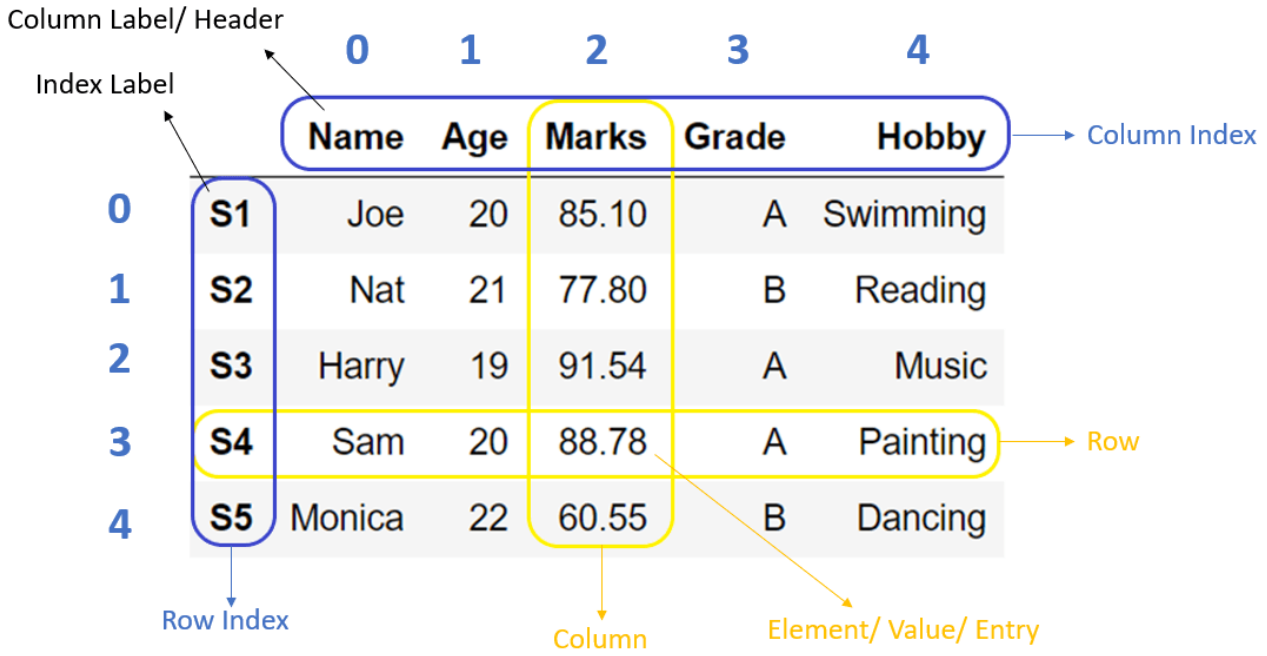
[115] Yves Hilpisch, Trading con Python; 2021; pag. 41

3.3.1. Archiviazione efficiente

Uno degli aspetti principali nello sviluppo di un software per il trading riguarda proprio l'archiviazione dei dati in modo da poterli recuperare e riutilizzare all'occorrenza. Ad esempio quando questi vengono recuperati da un servizio web possono essere gestiti in due modi:

- memorizzati su una copia temporanea in un preciso momento
- memorizzati su disco e aggiornati continuamente

L'oggetto più comunemente usato per memorizzare i dati, grazie alla sua versatilità e facilità delle operazioni, è il DataFrame della libreria pandas che come si vede nell'immagine è assimilabile a una tabella con record indicizzati:



In caso di volumi di dati molto grandi e immutabili al fine di velocizzare il recupero delle informazioni è utile memorizzare gli oggetti DataFrame in altri standard di memorizzazione binari, es.HDF5, che permettono di aumentare la velocità di elaborazione avvicinandosi alla massima teorica dell'elaboratore.

Per analisi più sofisticate tramite query in linguaggio SQL è possibile scrivere i dati delle serie storiche contenuti negli oggetti DataFrame in un database relazionale. Anche se la velocità non è paragonabile ai formati di archiviazione binari questo rende possibile l'utilizzo di istruzioni standardizzate con compiti analitici già a DB.

3.4. Backtesting vettorizzato

Un altro step fondamentale nello sviluppo di un'applicazione di trading è il backtesting, che come dice la parola stessa permette di testare e verificare le idee implementate nel programma su dati passati prima di iniziare l'effettivo utilizzo.

La vettorizzazione è un concetto del calcolo scientifico "prestato" alla programmazione, fa riferimento allo stile in cui le operazioni sui valori scalari sono generalizzate a vettori, matrici e array

multidimensionali. Nell'ambito dell'analisi finanziaria, soprattutto nel contesto del backtesting, questo semplifica e velocizza le procedure che interessano serie storiche di dati come nel caso dei prezzi azionari.

Questo approccio è molto utilizzato nei test di alcune strategie (es SMA, Momentum e Mean reversion) quando si tratta di:

- Strategie di trading semplici
- Esplorazione interattiva della strategia
- Visualizzazione come obiettivo principale
- Completi programmi di backtesting

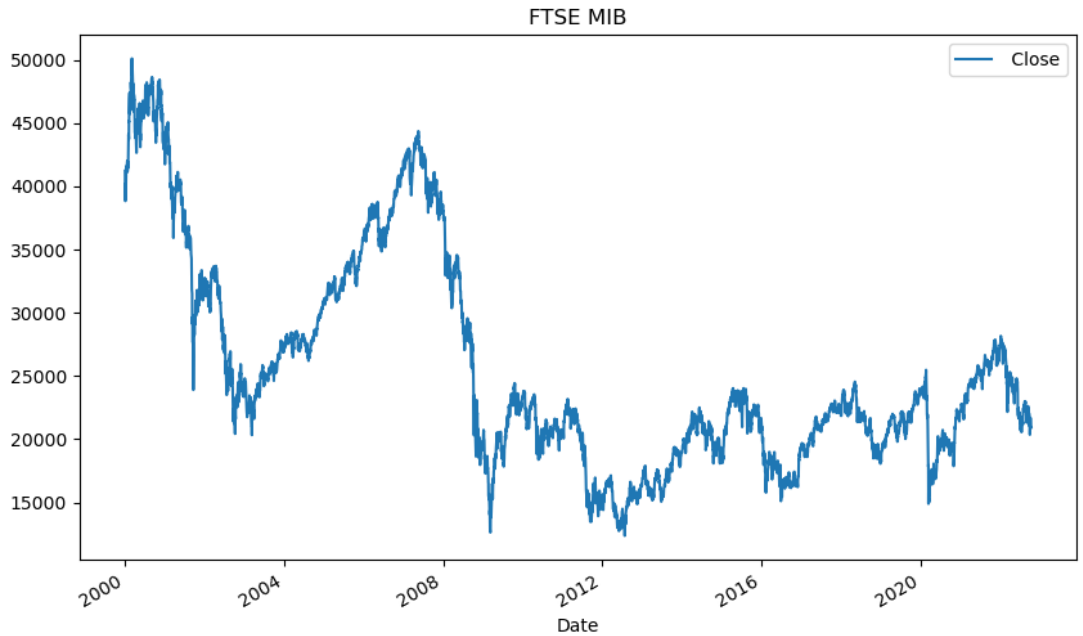
3.5. Applicazione Strategie in Python

I risultati che vengono presentati da qui in avanti si basano sul backtesting appena spiegato e ipotizzano un budget iniziale di 10K.

Oltre ai parametri elencati a inizio capitolo per ogni strategia viene indicato, se possibile in base all'algoritmo, un costo di transazione per ogni operazione che come vedremo può avere un peso molto alto sul profitto finale.

3.5.1. Strategie basate su SMA

La prima strategia testata è la Simple Moving Average applicata all'indice FTSE MIB il cui grafico dal 2000 ad oggi è presentato nella seguente immagine:

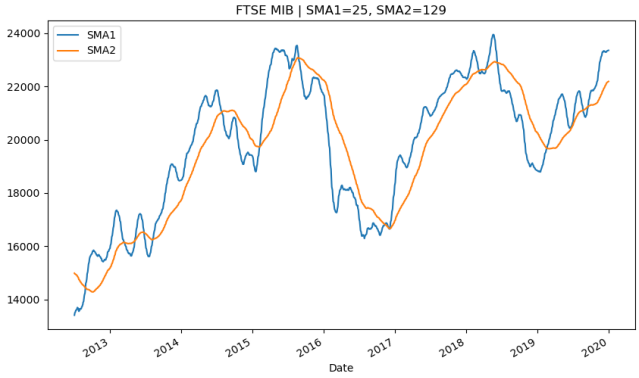


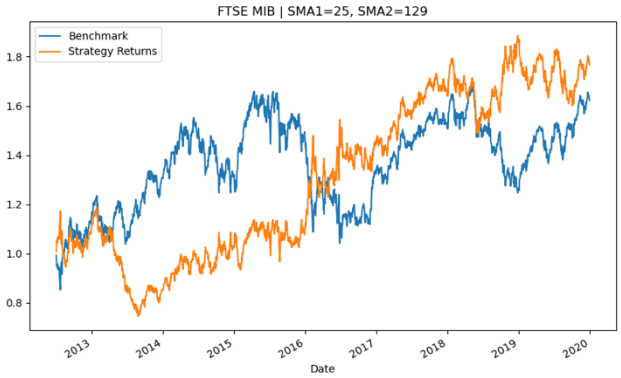
L'algoritmo utilizzato utilizza l'incrocio di 2 medie mobili per generare i segnali di acquisti/vendita e dispone inoltre di un metodo in grado di individuare la coppia di medie mobili che massimizzano i

ritorni in un range di valori definito dall'utente: 5-30 giorni per la SMA minore e 30-200 per la maggiore.

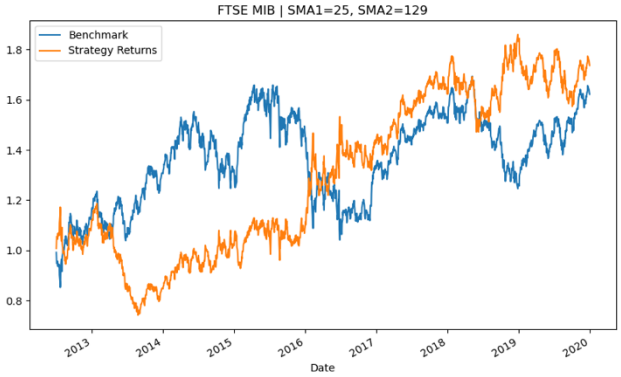
I risultati sono i seguenti:

- Intervallo temporale = 01/01/2012 – 01/01/2020 e SMA ottimizzate = 25 e 129
  Benchmark: € End of period =16.200€ con un rendimento annualizzato del 6,22%



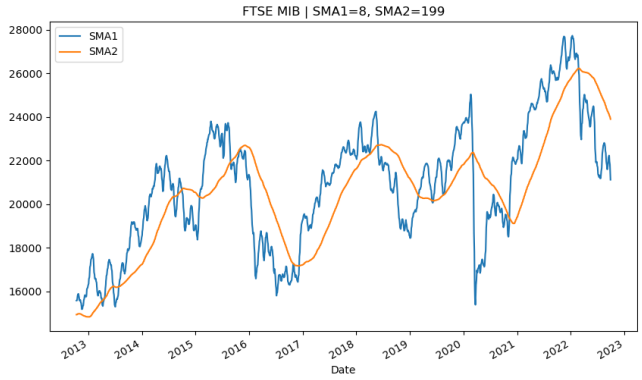


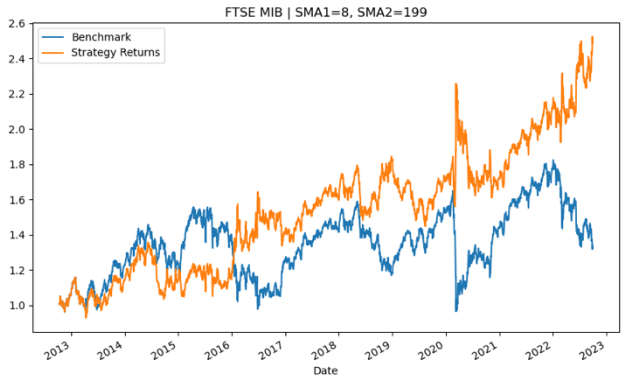
Transaction cost =0
€ End of period = 17.700 €
Annual Return = 7,40%



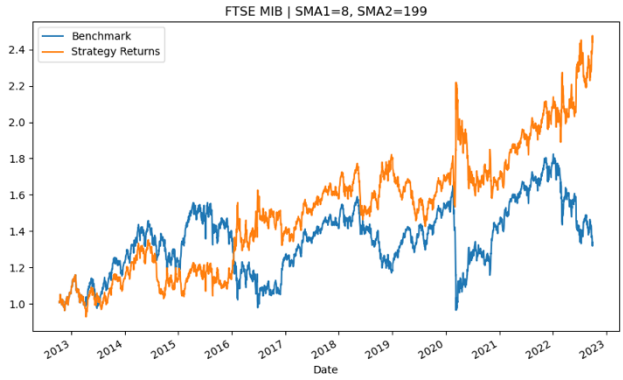
Transaction cost =0,001(0,1%)
€ End of period = 17.400 €
Annual Return = 7,17%

- Intervallo temporale = 01/01/2012 – 01/10/2022 e SMA ottimizzate = 8 e 199
  Benchmark: € End of period =13.400€ con un rendimento annualizzato del 2,52%



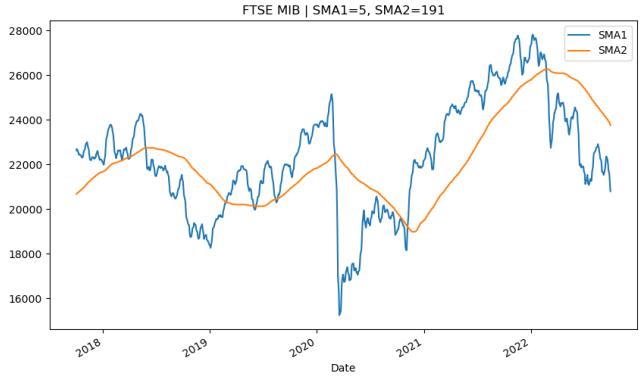


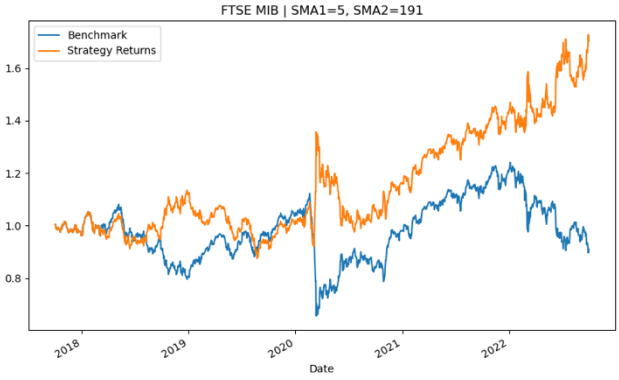
Transaction cost =0
€ End of period = 24.900 €
Annual Return = 8,07%



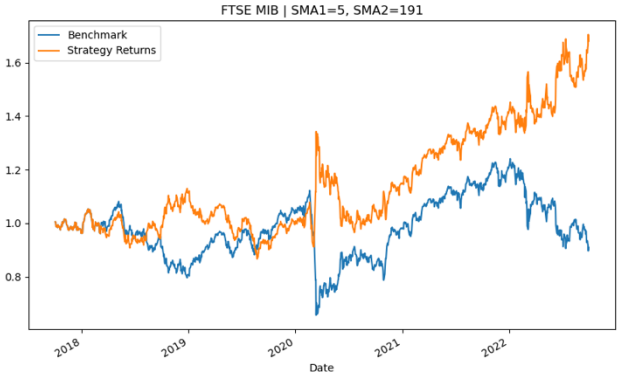
Transaction cost =0,001(0,1%)
€ End of period = 24.400 €
Annual Return = 7,89%

- Intervallo temporale = 01/01/2017 – 01/10/2022 e SMA ottimizzate = 5 e 191
  Benchmark: € End of period =9.100€ con un rendimento annualizzato di -1,63%





Transaction cost =0
€ End of period = 17.000 €
Annual Return = 9,67%



Transaction cost =0,001(0,1%)
€ End of period = 16.800 €
Annual Return = 9,44%

In tutti i casi esaminati la strategia batte i valori di benchmark e sembra comportarsi bene anche nei periodi più critici, anzi proprio in questi sembra ottenere le performance migliori.

Qui di seguito ecco un recap per gli altri risultati dove sono evidenziati in verde quelli migliori del benchmark e in rosso i peggiori.

Start = 10K € Annual Return= R		Intervallo														
		01/01/2012-01/01/2020					01/01/2012-01/10/2022					01/01/2017-01/10/2022				
		SMA1	SMA2	TC= 0	TC=0.001	Benchmark	SMA1	SMA2	TC= 0	TC=0.001	Benchmark	SMA1	SMA2	TC= 0	TC=0.001	Benchmark
FTSE MIB	€ EOP	25	129	17700	17400	16200	10	213	24900	24400	13400	5	191	17000	16800	9100
	R			7,40%	7,17%	6,22%			8,07%	7,89%	2,52%			9,67%	9,44%	-1,63%
S&P500	€ EOP	21	167	20400	20300	22900	21	164	28800	28500	25400	10	81	19100	18700	15000
	R			9,32%	9,25%	10,91%			9,42%	9,32%	8,26%			11,91%	11,50%	7,31%
Dow Jones	€ EOP	28	199	20100	20000	22200	5	186	28700	28200	24400	8	167	21600	21500	14200
	R			9,12%	9,05%	10,48%			9,39%	9,22%	7,89%			14,33%	14,24%	6,29%
NASDAQ	€ EOP	29	186	23600	23400	29000	5	168	38500	37800	34700	5	168	29700	29600	16500
	R			11,33%	11,21%	14,24%			12,16%	11,98%	11,17%			20,84%	20,77%	9,10%
DAX	€ EOP	16	82	14600	14100	19700	20	32	18000	16400	18000	20	32	17100	16300	10300
	R			4,84%	4,39%	8,84%			5,13%	4,30%	5,13%			9,78%	8,87%	0,52%
FTSE 100	€ EOP	29	166	9600	9400	12900	22	185	9700	9500	11700	5	31	11800	11100	9500
	R			-0,51%	-0,77%	3,23%			-0,26%	-0,44%	1,35%			2,92%	1,83%	-0,89%
Nikkei 225	€ EOP	29	139	25100	24700	27900	29	139	20400	20000	30600	13	30	14100	13400	13300
	R			12,19%	11,97%	13,68%			6,26%	6,08%	9,99%			6,16%	5,22%	5,08%
Apple	€ EOP	17	30	46100	43300	40300	5	35	91200	82800	75400	5	30	50700	48300	40900
	R			21,05%	20,10%	19,03%			20,70%	19,71%	18,76%			32,62%	31,51%	27,76%
Stellantis	€ EOP	29	198	35400	35100	46700	7	198	93200	90900	45700	10	34	46200	44300	12100
	R			17,12%	16,99%	21,25%			20,92%	20,66%	13,81%			30,49%	29,54%	3,37%
Banca Generali	€ EOP	27	158	95200	94600	31000	10	145	89300	87800	34000	5	157	20800	20500	9900
	R			32,53%	32,43%	15,19%			20,48%	20,31%	10,98%			13,58%	13,30%	-0,17%
Intesa Sanpaolo	€ EOP	6	106	29400	28400	23400	6	111	25800	24900	15000	5	173	14600	14400	6000
	R			14,43%	13,94%	11,21%			8,40%	8,07%	3,51%			6,80%	6,55%	-8,50%
Enel	€ EOP	12	30	19000	17700	21800	7	40	27300	24900	13800	6	69	28900	28300	9600
	R			8,35%	7,40%	10,23%			8,92%	8,07%	2,78%			20,27%	19,83%	-0,71%

Se si osservano i risultati fino a prima degli inizi del COVID nel 2020, nella maggior parte dei casi il keep and hold risultava vincente ma se si considera il triennio 2020-2022 gli esiti sono a favore della strategia anche se in alcuni casi di poco. Dai test effettuati, e come si evince anche dai grafici FTSE MIB sopra riportati, però i profitti maggiori si verificano in casi particolari cioè in corrispondenza di movimenti importanti dei prezzi degli strumenti finanziari.

Nella strategia i costi di transazione dello 0,1% ipotizzati risultano avere una bassa incidenza sul risultato finale a causa del numero limitato di operazioni effettuate.

3.5.2. Strategie basate su Momentum

Esistono due tipi fondamentali di strategie momentum:

- Momentum cross-essential, si concentra sui rendimenti relativi dei titoli in modo trasversale, acquistando asset che recentemente hanno avuto performance migliori rispetto ai concorrenti o a un benchmark e vendendo quelli con performance inferiori
- Momentum a serie storiche, utilizza il rendimento passato dei titoli acquistando quelli che hanno avuto un buon rendimento e vendendo quelli con rendimenti scarsi.

In questa trattazione viene approfondito quest'ultimo.

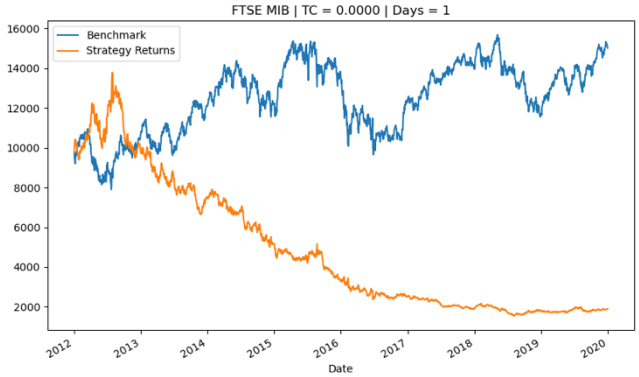
3.5.2.1. A serie storiche

Nella versione più semplice acquista titoli con ultimo rendimento positivo e vende in caso contrario. È possibile però utilizzare una finestra temporale a piacere portandola per esempio a 3 o più giorni in modo che la decisione avvenga sulla base di un maggior numero di informazioni.

È stato implementato inoltre un metodo che permette di confrontare i risultati della strategia su diversi intervalli così da poter scegliere direttamente il più remunerativo e i risultati saranno presentati solo in quest'ottica.

A titolo di esempio qui di seguito ecco gli esiti della strategia sull'indice FTSE MIB per intervalli di tempo 1, 3, 7, 9 giorni con un investimento iniziale di 10K €.

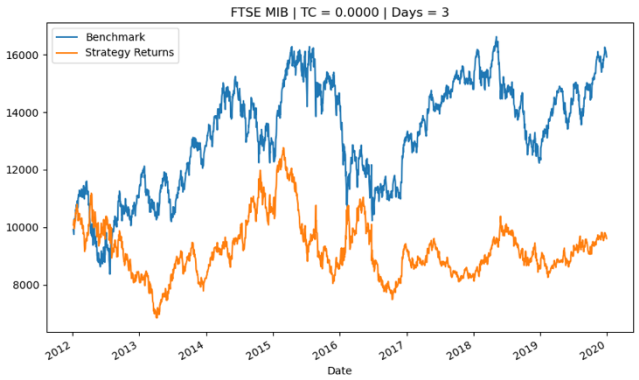
- Intervallo temporale = 01/01/2012 – 01/01/2020
  Benchmark: € End of period =15.600€ con un rendimento annualizzato del 5,77%



Intervallo(days) = 1

Transaction cost =0
€ End of period = 1.899 €
Annual Return = -18,75%

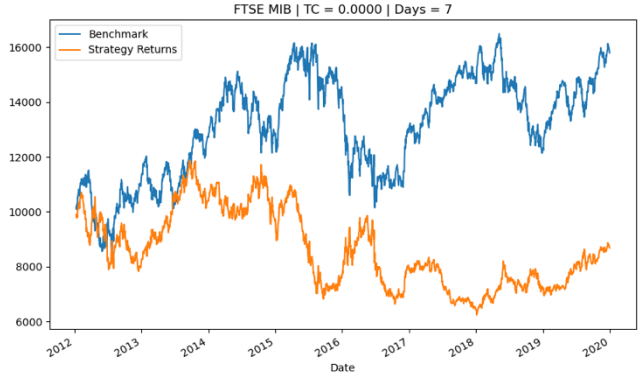
Transaction cost =0,001(0,1%)
€ End of period = 652 €
Annual Return = -28,91%



Intervallo(days) = 3

Transaction cost =0
€ End of period = 9.606 €
Annual Return = -0,5%

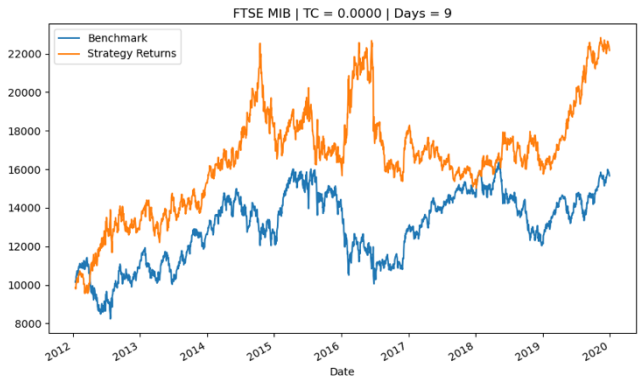
Transaction cost =0,001(0,1%)
€ End of period = 5.542 €
Annual Return = -7,11%



Intervallo(days) = 7

Transaction cost =0
€ End of period = 8.683 €
Annual Return = -1,75%

Transaction cost =0,001(0,1%)
€ End of period = 6.143 €
Annual Return = -5,91%

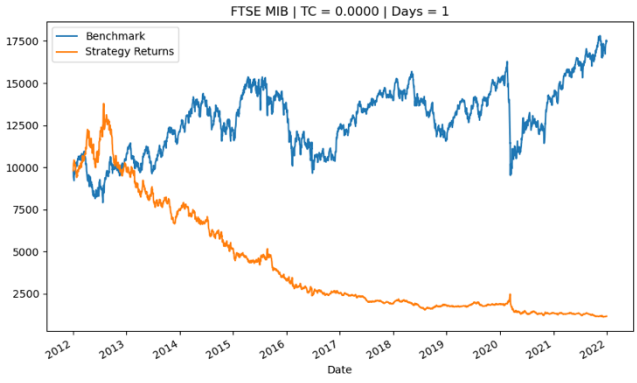


Intervallo(days) = 9

Transaction cost =0
€ End of period = 22.169 €
Annual Return = 10,46%

Transaction cost =0,001(0,1%)
€ End of period = 16.572 €
Annual Return = 6,52%

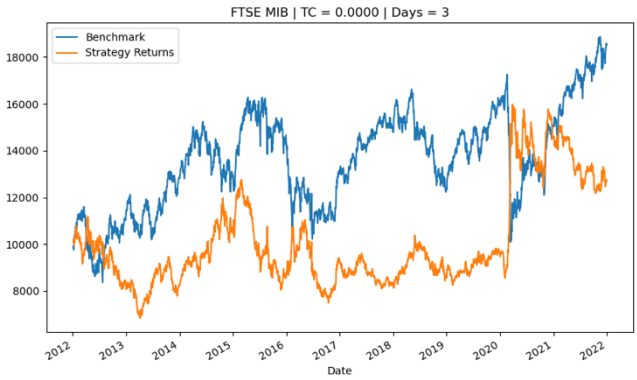
- Intervallo temporale = 01/01/2012 – 01/10/2022
  Benchmark: € End of period =13.500€ con un rendimento annualizzato del 2,59%



Intervallo(days) = 1

Transaction cost =0
€ End of period = 1.589 €
Annual Return = -14,49%

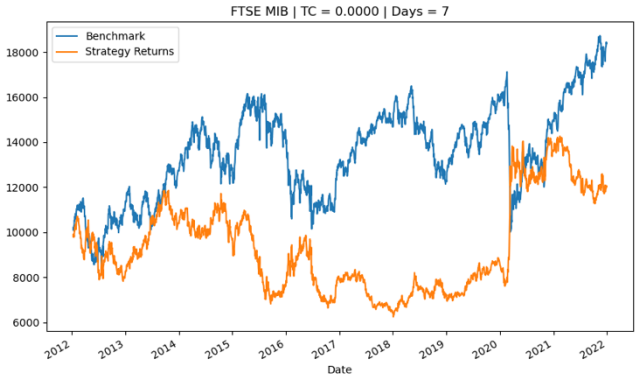
Transaction cost =0,001(0,1%)
€ End of period = 377 €
Annual Return = -24,35%



Intervallo(days) = 3

Transaction cost =0
€ End of period = 14.456 €
Annual Return = 3,19%

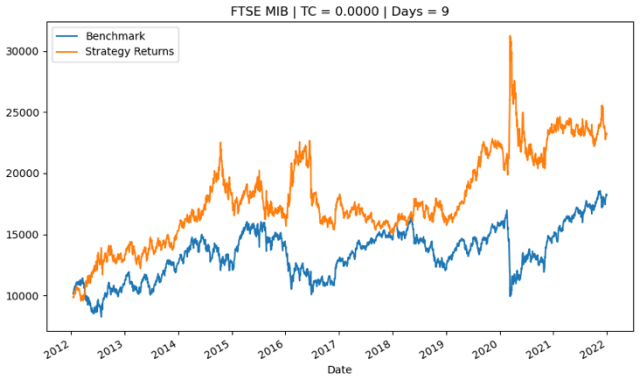
Transaction cost =0,001(0,1%)
€ End of period = 6.842 €
Annual Return = -3,18%



Intervallo(days) = 7

Transaction cost =0
€ End of period = 15.530 €
Annual Return = 3,82%

Transaction cost =0,001(0,1%)
€ End of period = 9.765 €
Annual Return = -0,2%

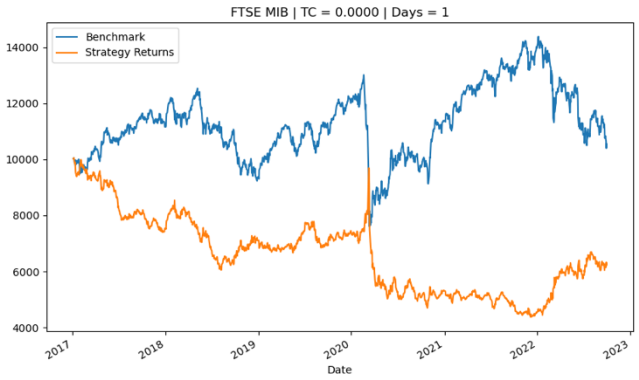


Intervallo(days) = 9

Transaction cost =0
€ End of period = 26.296 €
Annual Return = 8,58%

Transaction cost =0,001(0,1%)
€ End of period = 17.591 €
Annual Return = 4,92%

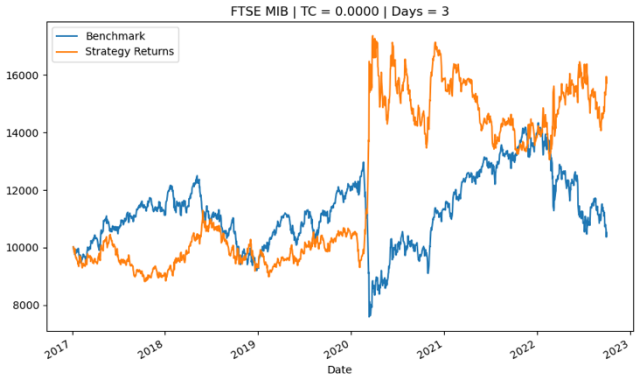
- Intervallo temporale = 01/01/2017 – 01/10/2022
  Benchmark: € End of period =10.596€ con un rendimento annualizzato di 1,01%



Intervallo(days) = 1

Transaction cost =0
€ End of period = 6.234 €
Annual Return = -7,89%

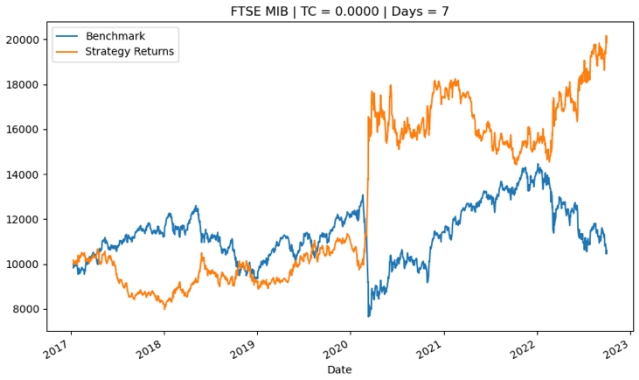
Transaction cost =0,001(0,1%)
€ End of period = 2.904 €
Annual Return = -19,35%



Intervallo(days) = 3

Transaction cost =0
€ End of period = 15.718 €
Annual Return = 8,18%

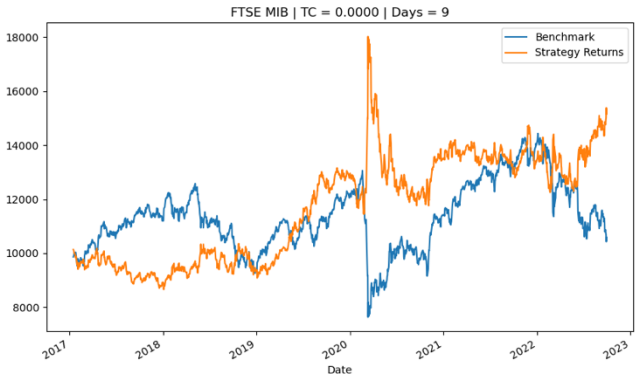
Transaction cost =0,001(0,1%)
€ End of period = 10.525 €
Annual Return = 0,89%



Intervallo(days) = 7

Transaction cost =0
€ End of period = 19.871 €
Annual Return = 12,68%

Transaction cost =0,001(0,1%)
€ End of period = 15.507 €
Annual Return = 7,93%



Intervallo(days) = 9

Transaction cost =0
€ End of period = 15.158 €
Annual Return = 7,5%

Transaction cost =0,001(0,1%)
€ End of period = 12.214 €
Annual Return = 3,54%

Nei diversi scenari intervalli troppo brevi hanno spesso avuto risultati davvero pessimi in quanto basandosi su un numero limitato di informazioni(giorni) spesso si rischia di seguire nell'operatività falsi segnali temporanei.

Con questa strategia inoltre si nota maggiormente l'incidenza che possono avere i costi di transazione: in alcuni casi addirittura un +3% si trasforma in un risultato negativo se si include il costo di transazione ipotizzato dello 0,1%.

Dal suo utilizzo negli anni si è riscontrato che questo tipo di approccio risulta più efficiente se utilizzato su frame temporali orari invece che giornalieri.

Nella tabella riassuntiva contente tutte le simulazioni emerge inoltre che la strategia ha un miglior risultato se applicata alle singoli azioni delle compagnie:

Start = 10K € Annual Return= R		Intervallo											
		01/01/2012-01/01/2020				01/01/2012-01/10/2022				01/01/2017-01/10/2022			
		Days	TC= 0	TC=0.001	Benchmark	Days	TC= 0	TC=0.001	Benchmark	Days	TC= 0	TC=0.001	Benchmark
FTSE MIB	€ EOP	9	22169	16572	15659	9	26296	17591	13755	7	19871	15507	10596
	R		10,46%	6,52%	5,77%		8,58%	4,92%	2,75%		12,68%	7,93%	1,01%
S&P500	€ EOP	24	12114	10510	23984	24	17220	14169	27259	12	27793	23969	15786
	R		2,43%	0,62%	11,56%		4,73%	3,01%	8,91%		19,46%	16,42%	8,26%
Dow Jones	€ EOP	23	11940	10236	23529	11	21499	15441	26447	11	27867	23699	15343
	R		2,24%	0,29%	11,29%		6,73%	3,77%	8,63%		19,51%	16,19%	7,73%
NASDAQ	€ EOP	43	16393	14833	30830	20	25503	20610	37129	12	29766	25851	19036
	R		6,37%	5,05%	15,11%		8,29%	6,35%	11,81%		20,89%	17,96%	11,85%
DAX	€ EOP	33	15858	10926	19622	49	14100	10720	17607	49	14893	13706	10126
	R		5,93%	1,11%	8,79%		2,97%	0,59%	4,93%		7,17%	5,64%	0,22%
FTSE 100	€ EOP	21	8753	7052	13275	7	8817	6094	12133	30	11571	10049	9470
	R		-1,65%	-4,27%	3,60%		-1,07%	-4,13%	1,66%		2,57%	0,09%	-0,94%
Nikkei 225	€ EOP	9	17791	13595	27666	6	22830	14269	30514	6	20548	16067	13447
	R		7,47%	3,91%	13,56%		7,28%	3,07%	9,96%		13,34%	8,60%	5,29%
Apple	€ EOP	21	53650	45809	45164	21	159388	130105	85023	21	65447	59634	43000
	R		23,37%	20,95%	20,74%		26,57%	24,40%	19,98%		38,64%	36,42%	28,87%
Stellantis	€ EOP	11	87859	68424	47021	11	498734	355339	46100	11	54891	44941	14611
	R		31,21%	27,17%	21,35%		39,48%	35,51%	13,89%		34,46%	29,87%	6,82%
Banca Generali	€ EOP	9	59900	45181	42682	44	53909	45755	29109	45	22172	20143	12165
	R		25,08%	20,75%	19,89%		15,42%	13,82%	9,52%		14,85%	12,95%	3,47%
Intesa Sanpaolo	€ EOP	8	13643	9818	18943	8	37489	24632	13733	8	47644	38930	7008
	R		3,96%	-0,23%	8,31%		11,90%	7,97%	2,74%		31,19%	26,67%	-6,00%
Enel	€ EOP	48	14360	12851	24675	48	22325	19292	14738	49	28712	27448	10105
	R		4,63%	3,19%	11,95%		7,07%	5,75%	3,36%		20,13%	19,20%	0,18%

Questo è ovvio in quanto il momentum funziona un po' da analisi del sentiment degli operatori nei confronti dello strumento finanziario ed è più facile intercettare questo aspetto su una singola zione che su un paniere come gli indici.

3.5.3. Strategie basate su Mean Reversion

Si può dire che le strategie mean reversion si basano su un approccio opposto a quelle momentum: se uno strumento finanziario ha avuto performance troppo buone rispetto a un normale trend viene venduto mentre in caso contrario viene acquistato in quanto si pensa possa seguire una “reversione alla sua media”. Quindi mentre la strategia momentum presuppone una correlazione positiva tra i rendimenti la mean reversion la presuppone negativa.

La mean reversion utilizza una media mobile semplice SMA per approssimare una tendenza del prezzo e a seconda della distanza tra il prezzo e la media si ricevono dei segnali di long o di short.

Applicando la strategia sull’indice FTSE MIB con un investimento inziale di 10K€, i parametri che vengono fatti variare nei test sono:

- La deviazione assoluta dal prezzo corrente
- L’intervallo su cui viene costruita la SMA

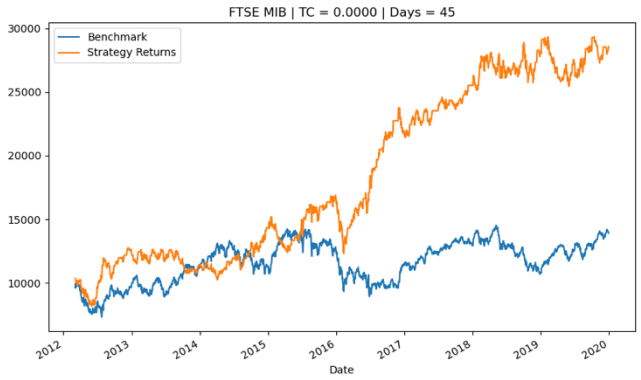
Anche in questo caso è stato implementato un algoritmo in grado di estrapolare la miglior combinazione varianza – SMA in base a un range definito dall’utente:

- SMA: 9 – 21 – 33 – 45 – 55 giorni per gli indici e 3 – 5 – 7 – 9 – 13 per le azioni
- Varianza: 320 – 520 € (circa 1,5% - 2,5% del prezzo attuale)

e i risultati finali verranno presentati in quest'ottica.

Esiti della strategia sull'indice FTSE MIB:

- Intervallo temporale = 01/01/2012 – 01/01/2020
  Benchmark: € End of period =15.600€ con un rendimento annualizzato del 5,77%

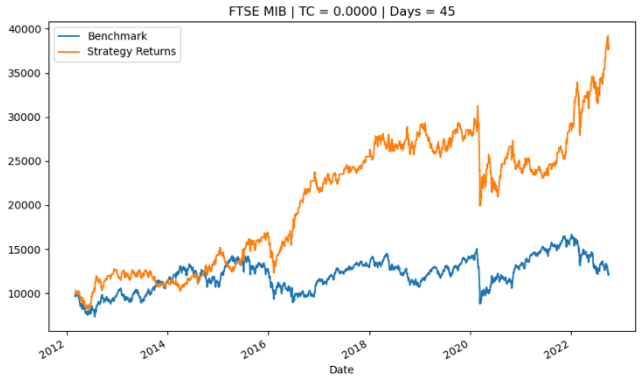


SMA = 45
Varianza =427(21.500)

Transaction cost =0
€ End of period = 28.553 €
Annual Return = 14,01%

Transaction cost =0,001(0,1%)
€ End of period = 24.282 €
Annual Return = 11,73%

- Intervallo temporale = 01/01/2012 – 01/10/2022
  Benchmark: € End of period =13.500€ con un rendimento annualizzato del 2,59%

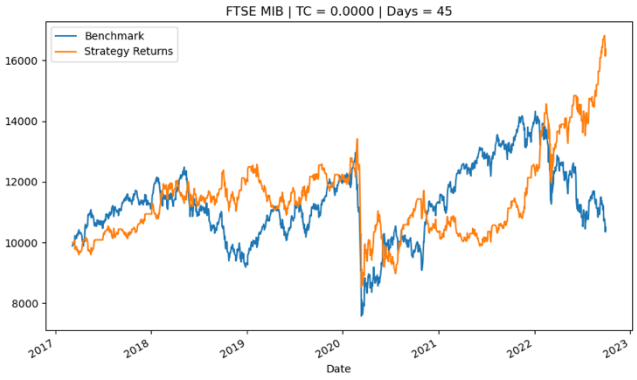


SMA = 45
Varianza =427(21.500)

Transaction cost =0
€ End of period = 38.182 €
Annual Return = 12,08%

Transaction cost =0,001(0,1%)
€ End of period = 30.276 €
Annual Return = 9,89%

- Intervallo temporale = 01/01/2017 – 01/10/2022
  Benchmark: € End of period =10.596€ con un rendimento annualizzato di 1,01%



SMA = 45
Varianza =427(21.500)

Transaction cost =0
€ End of period = 16.375 €
Annual Return = 8,96%

Transaction cost =0,001(0,1%)
€ End of period = 14.508 €
Annual Return = 6,69%

Dalla tabella risultante di tutti i test la strategia non sembra dare risultati soddisfacenti:

Start = 10K € Annual Return= R		Intervallo															Prezzo usato per calcolo varianza
		01/01/2012-01/01/2020					01/01/2012-01/10/2022					01/01/2017-01/10/2022					
		SMA	Varianza	TC= 0	TC=0.001	Benchmark	SMA	Varianza	TC= 0	TC=0.001	Benchmark	SMA	Varianza	TC= 0	TC=0.001	Benchmark	
FTSE MIB	€ EOP	45	422	28553	24282	13907	45	427	38182	30276	12216	45	427	16375	14508	10500	21500
	R			14,01%	11,73%	4,21%			12,08%	9,89%	1,72%			8,96%	6,69%	0,85%	
S&P500	€ EOP	55	87	15258	14645	23029	21-9	55-88	12924	11596	27079	9	84-88	12788	11629	15736	3700
	R			5,42%	4,88%	10,99%			2,21%	1,27%	8,85%			4,37%	2,66%	8,20%	
Dow Jones	€ EOP	33	466	15650	14005	23250	33	466	13361	11194	25406	9	776	9551	8784	15250	31080
	R			5,76%	4,30%	11,12%			2,50%	0,96%	8,26%			-0,80%	-2,23%	7,62%	
NASDAQ	€ EOP	45	232	17155	16205	30563	9	257	15908	13816	39014	9	257	15889	13854	18972	10800
	R			6,98%	6,22%	14,99%			4,03%	2,79%	12,28%			8,39%	5,83%	11,78%	
DAX	€ EOP	21-55	304-226	18167	16095	20558	55	226	22004	18490	16924	33	235	14108	12743	10271	12700
	R			7,75%	6,13%	9,43%			6,94%	5,37%	4,58%			6,17%	4,31%	0,47%	
FTSE 100	€ EOP	45	106	25182	21757	12759	55	115	35991	29467	11555	55	115	18873	16907	9284	6900
	R			12,24%	10,20%	3,09%			11,52%	9,63%	1,24%			11,68%	9,56%	-1,28%	
Nikkei 225	€ EOP	9	662	14264	13220	27941	45	472	18155	15134	27085	21	558	14161	12580	13712	26800
	R			4,54%	3,55%	13,71%			5,21%	3,59%	8,85%			6,24%	4,07%	5,64%	
Apple	€ EOP	5	2,35	12001	11811	48740	5	3,52	26598	23614	91755	5	3,52	26598	23614	46457	147
	R			2,31%	2,10%	21,89%			8,68%	7,59%	20,76%			18,55%	16,12%	30,62%	
Stellantis	€ EOP	5	0,195-0,3185	17698	13763	50188	3	0,299	19879	15206	52154	5	0,3185	24934	19206	13120	13
	R			7,40%	4,07%	22,34%			6,02%	3,63%	15,09%			17,22%	12,02%	4,84%	
Banca Generali	€ EOP	9	0,6815	18105	14142	42525	9	0,6815	17019	11921	41997	9	0,6815	9058	7445	12196	29
	R			7,70%	4,43%	19,83%			4,63%	1,51%	12,99%			-1,71%	-5,00%	3,51%	
Intesa Sanpaolo	€ EOP	3	0,028	25506	15530	18308	3	0,04	15537	9657	13273	3	0,044	6160	5166	6704	1,83
	R			12,42%	5,66%	7,85%			3,82%	-0,30%	2,44%			-8,08%	-10,85%	-6,72%	
Enel	€ EOP	7	0,094	35583	26839	22637	7	0,094	34330	22131	13521	7	0,075	13000	9605	10247	4,19
	R			17,19%	13,13%	10,75%			11,07%	6,99%	2,60%			4,67%	-0,70%	0,43%	

Questo però non ci deve ingannare in quanto può essere dovuto ai vincoli applicati ai parametri per questione di tempo nelle analisi che altrimenti sarebbe stato davvero elevato. In particolare si è deciso di limitare le SMA a soli 5 valori e la varianza in un intervallo ben definito dal 1,5 al 2,5 % con scatti incrementali dello 0,05%.

Nelle colonne SMA e varianza quando sono riportati 2 valori, il secondo fa riferimento al migliore possibile in presenza dei costi di transazione. Questo indica che la loro incidenza è tale da impattare sui parametri ottimali della strategia.

3.5.4. Possibili problematiche nel backtesting

Le tecniche viste fino ad ora sono solo alcune delle versioni più semplici a disposizione per questi tipi di algoritmi ma nel backtesting avrebbero comunque dato risultati sufficienti, addirittura sempre molto positivi per il nostro FTSE MIB.

Non bisogna però prendere per assodati questi algoritmi applicandoli indistintamente a diversi strumenti finanziari perché potrebbero non generalizzarsi altrettanto bene in molti casi o modificando anche solo un parametro che li caratterizza.

È molto importante tenere a mente questo aspetto per non ricadere nel cosiddetto *Snooping*, cioè quel fenomeno che si verifica *quando un determinato insieme di dati viene utilizzato più volte per scopi di inferenza o scelta del modello*.[116] In questi casi si utilizzano gli stessi dati e la medesima strategia modificandone i parametri al fine di ottenere risultati soddisfacenti per vendere il prodotto con un determinato potenziale economico ma irrealistico nel contesto reale.

L'altro problema in cui si può rilevare è quello, già precedentemente spiegato, dell'overfitting dove il modello ha buone performance sui dati di test ma risulta scadente sui nuovi dati futuri.

[116] Yves Hilpisch, Trading con Python; 2021; pag. 101

3.5.5. IA, Machine learning e Reti neurali

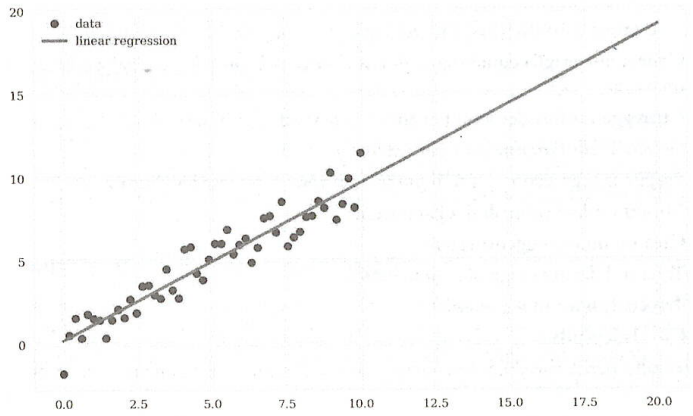
Con gli enormi progressi avuti negli ultimi anni nelle aree dell'IA, machine learning e deep learning anche l'industria finanziaria sta cercando di trarre vantaggio da questi ultimi approcci all'analisi dei dati con lo scopo di predire possibili movimenti futuri e anticiparli.

In questo capitolo verranno testate le seguenti strategie:

- Strategie basate sulla regressione lineare
- Strategie basate sul machine learning
- Strategie basate sul deep learning

3.5.5.1. Regressione lineare

Questa strategia utilizza la tecnica, recuperata dalla statistica, della regressione lineare, o dei minimi quadrati ordinari (OLS, Ordinary Least Squares), per cercare di individuare una tendenza o derivare la possibile direzione futura di uno strumento finanziario.



Negli esempi il grado più elevato dei monomi della regressione lineare è 1. L'allargamento dell'intervallo della variabile dipendente rispetto al dominio del dataset originale permette inoltre, dati i parametri ottimali della regressione, di predire valori per la variabile dipendente Y oltre il dominio del dataset mediante un'estrapolazione.

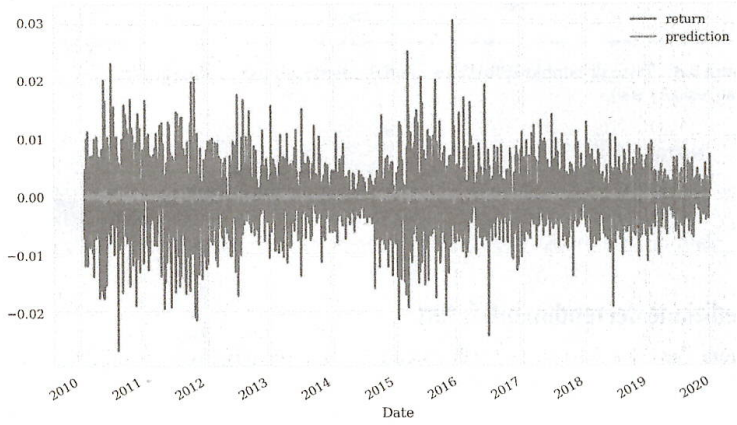
117

Come in tutte le strategie viste fino ad ora l'obiettivo è di predire il livello dell'indice nei giorni a venire in base ai valori passati e il numero di giorni utilizzati come input viene chiamato *lag*.

Per esempio prende il nome di *tre lag* quando viene utilizzato l'indice odierno e di 2 giorni precedenti, la regressione utilizzata avrà quindi 3 variabili indipendenti e una dipendente.

I parametri ottimali individuati dall'algoritmo con l'utilizzo dei prezzi come variabili indipendenti, illustrano quella che viene chiamata ipotesi a percorso casuale, afferma che i prezzi seguono un percorso casuale con la conseguenza che il miglior predittore per il prezzo di domani è quello di oggi. Utilizzare al posto dei prezzi i rendimenti logaritmici potrebbe essere la scelta migliore per questi casi a causa della loro caratteristica di rendere stazionari i dati delle serie storiche limitando gli effetti di uno shock improvviso su eventuali calcoli di medie etc... Come si vede nella seguente immagine

[117] Yves Hilpisch, Trading con Python; 2021; pag. 116



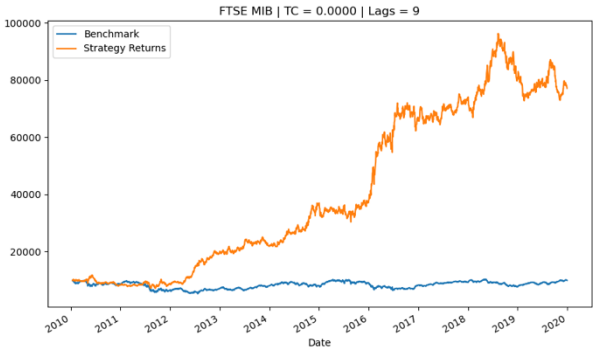
[118]

in questo modo non può essere predetto in modo significativo il rendimento futuro ma si può riuscire a individuare meglio la direzione del mercato basando la regressione lineare sul segno dei rendimenti logaritmici che servono come valori della variabile dipendente, utilizzando valori di segno 1 e -1 come passo di regressione.

Per una strategia basata sulla predizione oltre al tasso di successo riveste un ruolo importantissimo anche il market timing, cioè la capacità di individuare tempestivamente i giorni migliori e peggiori in modo da agire di conseguenza.

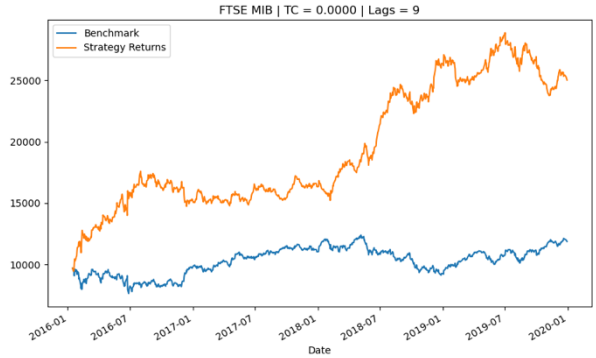
Nel codice utilizzato per questo algoritmo è anche possibile avere una visione più chiara dei problemi indicati per i tre approcci più semplici dei paragrafi precedenti, Snooping e Overfitting, in quanto l'algoritmo è addestrato su un set di dati campione e poi valutato sui restanti.

Questo problema risulta molto evidente nel seguente esempio dove nel caso 1 viene addestrato e testato l'algoritmo sullo stesso set di dati (medesimo arco temporale) mentre nel secondo caso no. In entrambi i casi è stato utilizzato un lags di 9 giorni e un costo transazione nullo:



Dataset addestramento = Dataset valutazione

Capitale fine periodo strategia= 77.166,45 €



Dataset addestramento ≠ Dataset valutazione

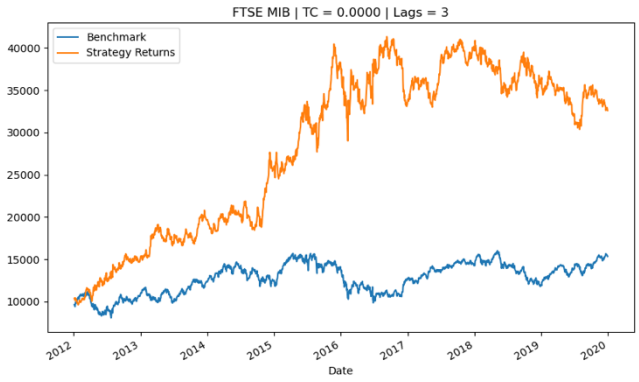
Capitale fine periodo strategia= 25.035,09 €

[118] Yves Hilpisch, Trading con Python; 2021; pag. 123

Nei nostri test si avevano a disposizione, per i diversi indici e azioni, dati dal 01/01/2000 al 01/10/2022, e perciò si sono utilizzati come dati di addestramento tutti quelli antecedenti e non compresi nell'intervallo di test. Essendo che il valore di lags può essere scelto a piacere dall'utente si è predisposto un algoritmo che individua il valore ottimale di lags in un range da 3 a 33.

Qui di seguito ecco i primi 3 risultati in ordine di performance per l'indice FTSE MIB, sempre con un budget iniziale di 10K €.

- Intervallo temporale = 01/01/2012 – 01/01/2020
  Benchmark: € End of period =15.600€ con un rendimento annualizzato del 5,77%

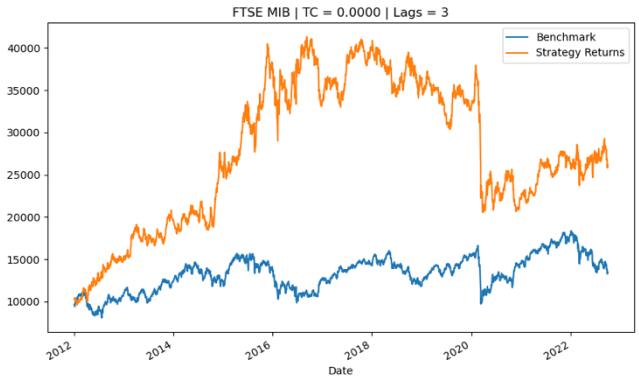


Lags = 3

Transaction cost =0
€ End of period = 32.603 €
Annual Return = 15,92%

Transaction cost =0,001(0,1%)
€ End of period = 18.054 €
Annual Return = 7,66%

- Intervallo temporale = 01/01/2012 – 01/10/2022
  Benchmark: € End of period =13.500€ con un rendimento annualizzato del 2,59%

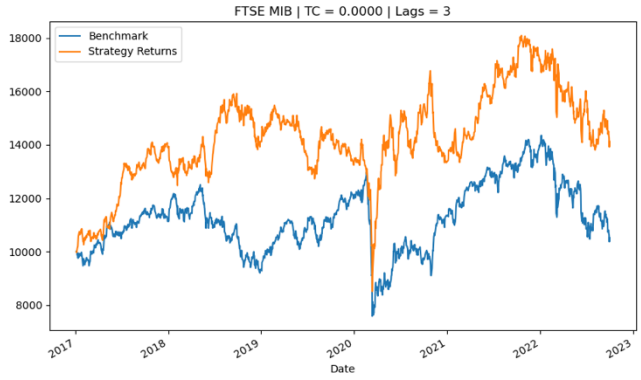


Lags = 3

Transaction cost =0
€ End of period = 26.192 €
Annual Return = 8,54%

Transaction cost =0,001(0,1%)
€ End of period = 11.757 €
Annual Return = 1,39%

- Intervallo temporale = 01/01/2017 – 01/10/2022
  Benchmark: € End of period =10.596€ con un rendimento annualizzato di 1,01%



Lags = 3

Transaction cost =0
€ End of period = 14.117 €
Annual Return = 6,18%

Transaction cost =0,001(0,1%)
€ End of period = 7.740 €
Annual Return = -4,36%

Tabella riassuntiva dei test:

Start = 10K € Annual Return= R		Intervallo											
		01/01/2012-01/01/2020				01/01/2012-01/10/2022				01/01/2017-01/10/2022			
		Lags	TC= 0	TC=0.001	Benchmark	Lags	TC= 0	TC=0.001	Benchmark	Lags	TC= 0	TC=0.001	Benchmark
FTSE MIB	€ EOP	3	32603	18054	15336	3	26192	11757	13472	3	14117	7740	10520
	R		15,92%	7,66%	5,49%		8,54%	1,39%	2,57%		6,18%	-4,36%	0,89%
S&P500	€ EOP	3-10	15042	6623	25219	3	29273	8977	27989	3	19977	10576	15800
	R		5,24%	-5,02%	12,26%		9,57%	-0,91%	9,15%		12,79%	0,98%	8,28%
Dow Jones	€ EOP	6	16094	6563	24665	6	33616	9934	26952	5	24315	12026	15309
	R		6,13%	-5,13%	11,95%		10,87%	-0,06%	8,80%		16,71%	3,26%	7,69%
NASDAQ	€ EOP	3-13	14567	6064	33607	9-20	11361	4613	39014	19	15939	10012	18838
	R		4,81%	-6,06%	16,36%		1,09%	-6,37%	12,28%		8,45%	0,02%	11,64%
DAX	€ EOP	10-14	10804	4018	21567	23-11	19959	5601	18308	17	16863	8142	10447
	R		0,97%	-10,77%	10,08%		6,06%	-4,81%	5,28%		9,51%	-3,51%	0,76%
FTSE 100	€ EOP	6-3	12340	6951	13439	3	18743	7792	12161	3	17090	10376	9588
	R		2,66%	-4,44%	3,76%		5,49%	-2,10%	1,68%		9,77%	0,64%	-0,73%
Nikkei 225	€ EOP	6	11515	5845	28211	21	14505	4838	29219	3-7	10927	5798	13332
	R		1,78%	-6,49%	13,84%		3,22%	-5,99%	9,55%		1,55%	-9,04%	5,13%
Apple	€ EOP	7	20421	9792	48646	4-8	68975	24652	91609	12	97324	54328	46151
	R		9,34%	-0,26%	21,87%		17,86%	7,98%	20,74%		48,55%	34,22%	30,47%
Stellantis	€ EOP	5	85666	34620	50188	33	47602	13802	42819	33	18825	9867	12005
	R		30,80%	16,79%	22,34%		14,20%	2,78%	13,18%		11,63%	-0,23%	3,23%
Banca Generali	€ EOP	7	12465	4271	44281	10	10626	3162	42151	6	25957	16601	12434
	R		2,79%	-10,09%	20,44%		0,52%	-9,33%	13,03%		18,04%	9,22%	3,86%
Intesa Sanpaolo	€ EOP	12	17888	8655	18569	4	5156	1881	14322	3	7580	4164	6704
	R		7,54%	-1,79%	8,04%		-5,48%	-13,25%	3,10%		-4,70%	-14,13%	-6,72%
Enel	€ EOP	3	100209	49267	22183	6	99567	38506	13858	32	18191	9234	10507
	R		33,39%	22,06%	10,47%		21,60%	12,16%	2,82%		10,97%	-1,38%	0,86%

Anche in questo caso i costi di transazione risultano avere un elevato impatto tanto da influenzare il valore ottimale della colonna lags dove il secondo valore, se presente, è il lags che fornisce le performance migliore considerando i costi di transazione pari allo 0,1%.

Si vede inoltre come i risultati sembrano migliorare con l'aumentare della % dei dati usati per l'addestramento.

3.5.5.2. Machine learning: regressione logistica

Come detto in precedenza questa è una strategia di semplice modello probabilistico per la classificazione binaria.

Nel trading algoritmico solitamente è sufficiente predire la direzione del movimento più che l'entità assoluta e con questo ragionamento il problema della predizione si semplifica diventando un problema di classificazione risolvibile a livello base con l'algoritmo della regressione logistica.

Per lo sviluppo di un sistema basato su di essa i passaggi principali sono 3:

I. Selezione del modello da istanziare
II. Addestramento del modello
III. Predizioni

Come nella regressione lineare si tratta in fondo di un problema di classificazione in cui ogni nuova osservazione deve essere associata a una categoria che può avere 2 valori, 1 o -1(1 o 0).

Nel machine learning si parla di caratteristiche, feature, invece di variabili indipendenti o descrittive come nella regressione semplice. Ogni caratteristica può essere identificata da un lag, quindi se si lavora con 3 lag indica che stiamo lavorando con 3 caratteristiche.

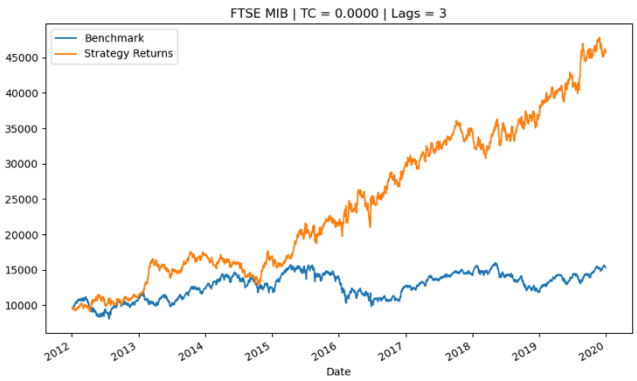
Solitamente la regressione logistica ha un tasso di successo, scelte corrette su errate, maggiore rispetto alla lineare ma performance peggiori e la percentuale di successi tende a diminuire con l'aumento del numero di lags ma di contro migliorano le performance.

Non sempre questo è vero però ad esempio nei test effettuati si è considerato un intervallo di lag da 3 a 33 e per l'indice FTSE MIB il risultato migliore in tutti gli intervalli si è riscontrato un lag pari a 3.

Come per i test sulla regressione lineare anche in questo caso si avevano a disposizione per i diversi indici/azioni dati dal 01/01/2000 al 01/10/2022 e si è utilizzato come dati di addestramento tutti quelli antecedenti e non compresi nell'intervallo di test. Inoltre come sopra si è predisposto un algoritmo in grado di individuare il valore ottimale di lags in un range da 3 a 33.

Qui di seguito ecco alcuni dei migliori risultati ottenuti per l'indice FTSE MIB, sempre con un budget iniziale di 10K €.

- Intervallo temporale = 01/01/2012 – 01/01/2020
  Benchmark: € End of period =15.600€ con un rendimento annualizzato del 5,77%

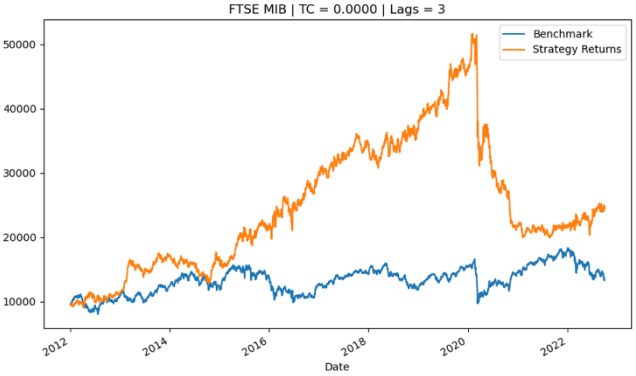


Lags = 3

Transaction cost =0
€ End of period = 45.705 €
Annual Return = 20,92%

Transaction cost =0,001(0,1%)
€ End of period = 17.818 €
Annual Return = 7,49%

- Intervallo temporale = 01/01/2012 – 01/10/2022
  Benchmark: € End of period =13.500€ con un rendimento annualizzato del 2,59%

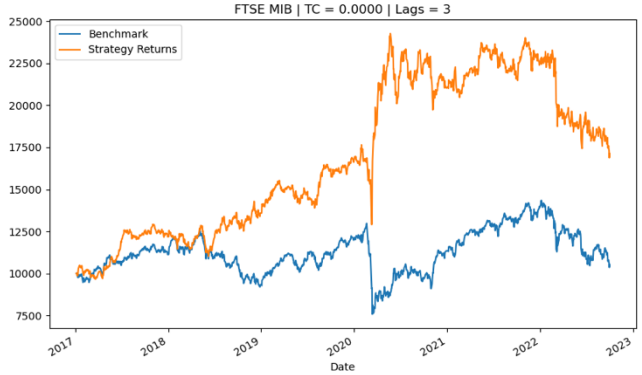


Lags = 3

Transaction cost =0
€ End of period = 24.617 €
Annual Return = 7,97%

Transaction cost =0,001(0,1%)
€ End of period = 7.010 €
Annual Return = -2,98%

- Intervallo temporale = 01/01/2017 – 01/10/2022
  Benchmark: € End of period =10.596€ con un rendimento annualizzato di 1,01%



Lags = 3

Transaction cost =0
€ End of period = 17.126 €
Annual Return = 9,81%

Transaction cost =0,001(0,1%)
€ End of period = 9.139 €
Annual Return = -1,55%

Con questa strategia nella maggior parte dei casi considerati sembrano avere un peso ancora maggiore i costi di transazione a causa di un numero più elevato di operazioni, portando ottimi rendimenti addirittura in negativo.

Dalla tabella sottostante si evince che la regressione logistica si comporta meglio rispetto a quella lineare anche se ci sono alcuni casi di performance eccessivamente buone che vanno verificate su futuri nuovi valori in modo da constatare l'eventuale presenza di fenomeni particolari, come l'overfitting.

Anche qui i risultati tendono a migliorare con l'aumento delle quantità di dati disponibili per l'addestramento.

Start = 10K € Annual Return= R		Intervallo											
		01/01/2012-01/01/2020				01/01/2012-01/10/2022				01/01/2017-01/10/2022			
		Lags	TC= 0	TC=0.001	Benchmark	Lags	TC= 0	TC=0.001	Benchmark	Lags	TC= 0	TC=0.001	Benchmark
FTSE MIB	€ EOP	3	45705	17818	15336	3	24617	7010	13472	3	17126	9139	10520
	R		20,92%	7,49%	5,49%		7,97%	-2,98%	2,57%		9,81%	-1,55%	0,89%
S&P500	€ EOP	19-3	33182	23640	24605	3	56187	55876	27989	4	34391	30199	15800
	R		16,18%	11,35%	11,91%		15,82%	15,77%	9,15%		23,96%	21,19%	8,28%
Dow Jones	€ EOP	3	26729	25630	24605	3	65017	57547	27226	5	35645	30193	15309
	R		13,08%	12,48%	11,91%		17,27%	16,06%	8,90%		24,74%	21,19%	7,69%
NASDAQ	€ EOP	18-9	29317	24562	31856	18	91131	59638	37548	19	27852	20387	18838
	R		14,39%	11,89%	15,58%		20,69%	16,41%	11,92%		19,50%	13,19%	11,64%
DAX	€ EOP	32-3	24135	10281	19691	33	28902	6592	17926	20	15031	10980	10447
	R		11,64%	0,35%	8,84%		9,45%	-3,48%	5,09%		7,34%	1,64%	0,76%
FTSE 100	€ EOP	8-29	16656	5648	13300	14	22168	4658	12007	8	17177	9417	9588
	R		6,59%	-6,89%	3,63%		7,01%	-6,30%	1,57%		9,87%	-1,04%	-0,73%
Nikkei 225	€ EOP	13	15442	5807	26987	13	13190	3580	29588	3	12235	10700	13332
	R		5,58%	-6,57%	13,21%		2,38%	-8,37%	9,67%		3,57%	1,18%	5,13%
Apple	€ EOP	9	25388	12888	48963	3	21999	10074	92567	10	135439	76517	46151
	R		12,35%	3,22%	21,96%		6,94%	0,06%	20,85%		57,33%	42,46%	30,47%
Stellantis	€ EOP	4	112700	43109	51362	7	355836	105053	49744	12-9	28748	16097	12005
	R		35,36%	20,04%	22,70%		35,53%	22,16%	14,63%		20,16%	8,63%	3,23%
Banca Generali	€ EOP	33	22695	9975	30956	29	14141	4284	32334	29-3	16275	13830	12434
	R		10,79%	-0,03%	15,17%		2,99%	-6,96%	10,50%		8,84%	5,80%	3,86%
Intesa Sanpaolo	€ EOP	4	27817	20628	19756	3	35806	26341	13273	4	4461	2607	6704
	R		13,64%	9,47%	8,88%		11,47%	8,59%	2,44%		-13,10%	-20,85%	-6,72%
Enel	€ EOP	3	95272	25835	23202	18	220164	38914	13858	28	25783	11997	10507
	R		32,55%	12,60%	11,09%		30,10%	12,26%	2,82%		17,91%	3,22%	0,86%

3.5.5.3. Deep learning

La strategia basata sul deep learning è simile a quella del machine learning, con l'obiettivo quindi di individuare i movimenti di mercato utilizzando le caratteristiche (informazioni) ricavate da altri giorni, ma utilizza le reti neurali spiegate in precedenza per affrontare il problema di classificazione.

Anche qui è presente il concetto di lags che indica il numero di caratteristiche considerate per le predizioni. Ci sono molti altri fattori da dover configurare solitamente tramite un procedimento di trial and error come la struttura della rete neurale, cioè il numero di layer nascosti e la quantità di neuroni per ogni layer, e le epoche, quante volte la rete analizza l'intero nostro set di dati.

Ai fini della trattazione si è cercato di ottimizzare i parametri della rete per le analisi e le predizioni inerenti all'indice FTSE MIB e per gli altri indici o azioni si è riutilizzata la medesima struttura di rete individuando però la miglior combinazione di lags ed epoche.

Essendo inoltre che una rete neurale tendenzialmente ha un apprendimento migliore tanto maggiore è la quantità di dati di addestramento, si è deciso di testarla solamente sull'ultimo intervallo temporale considerato nella trattazione 01/01/2017 – 01/10/2022 in modo da avere a disposizione il dataset più grande possibile per il learning.

In prima battuta come caratteristiche sono stati considerati solamente i prezzi di chiusura delle varie giornate con il seguente miglior risultato per l'indice FTSE MIB:

Lags=125 & Epoche=3

Transaction cost =0
€ End of period = 42.560 €
Annual Return = 28,64%

Transaction cost =0,001(0,1%)
€ End of period = 20.190 €
Annual Return = 13,00%

Dai risultati dei test emerge che una rete neurale:

- sembra essere più stabile nei redimenti rispetto agli altri algoritmi utilizzati fino ad ora in quanto fornisce risultati positivi e migliori del benchmark in tutti i casi con costi nulli, senza picchi irregolari che potrebbero essere dovuti ad anomalie o overfitting
- risulta però più sensibile ai costi di transazione a causa del maggior numero di operazioni effettuate

Start = 10K € Annual Return= R		Intervallo 01/01/2017-01/10/2022				
		Lags	Epoch	TC= 0	TC=0.001	Benchmark
FTSE MIB	€ EOP	125	3	42560	20190	10870
	R			28,64%	13,00%	1,46%
S&P500	€ EOP	85	11	26330	15160	16020
	R			18,34%	7,50%	8,54%
Dow Jones	€ EOP	120-60	4-13	24410	15060	15510
	R			16,79%	7,38%	7,93%
NASDAQ	€ EOP	110	9	26900	14770	19650
	R			18,78%	7,02%	12,47%
DAX	€ EOP	30	3	24250	12420	11090
	R			16,66%	3,84%	1,82%
FTSE 100	€ EOP	110	3	18410	8510	9650
	R			11,20%	-2,77%	-0,62%
Nikkei 225	€ EOP	45	8	22060	11020	13570
	R			14,75%	1,70%	5,45%
Apple	€ EOP	65	13	66720	33950	47730
	R			39,11%	23,69%	31,24%
Stellantis	€ EOP	80	11	28370	17000	14990
	R			19,88%	9,67%	7,29%
Banca Generali	€ EOP	60	6	40860	22580	12920
	R			27,74%	15,22%	4,56%
Intesa Sanpaolo	€ EOP	125	6	41670	27880	7020
	R			28,17%	19,52%	-5,97%
Enel	€ EOP	124	4	32370	19670	10090
	R			22,66%	12,49%	0,16%

Il grosso vantaggio di una rete neurale consiste però nel fatto che è possibile inserire ulteriori caratteristiche nell'addestramento per cercare di migliorare la capacità predittiva e le performance del modello.

Sono stati eseguiti dei test anche in quest'ottica aggiungendo features come momentum, volatilità e distanza:

Start = 10K € Annual Return= R		Intervallo 01/01/2017-01/10/2022									
		Caratteristiche= chiusure giornaliere					Caratteristiche= chiusure, momentum, volatilità, distanza				
		Lags	Epoch	TC= 0	TC=0.001	Benchmark	Lags	Epoch	TC= 0	TC=0.001	Benchmark
FTSE MIB	€ EOP	125	3	42560	20190	10870	135	8	49280	23940	10870
	R			28,64%	13,00%	1,46%			31,97%	16,40%	1,46%
S&P500	€ EOP	85	11	26330	15160	16020	75	6-10	38540	30690	16020
	R			18,34%	7,50%	8,54%			26,44%	21,53%	8,54%
Dow Jones	€ EOP	120-60	4-13	24410	15060	15510	30	7-11	30710	21040	15510
	R			16,79%	7,38%	7,93%			21,55%	13,81%	7,93%
NASDAQ	€ EOP	110	9	26900	14770	19650	75	10	31650	16940	19650
	R			18,78%	7,02%	12,47%			22,19%	9,60%	12,47%
DAX	€ EOP	30	3	24250	12420	11090	35	13	15600	10450	11090
	R			16,66%	3,84%	1,82%			8,04%	0,77%	1,82%
FTSE 100	€ EOP	110	3	18410	8510	9650	35	13	15050	7980	9650
	R			11,20%	-2,77%	-0,62%			7,37%	-3,85%	-0,62%
Nikkei 225	€ EOP	45	8	22060	11020	13570	100	13	19660	9630	13570
	R			14,75%	1,70%	5,45%			12,48%	-0,65%	5,45%
Apple	€ EOP	65	13	66720	33950	47730	30	9	75170	49990	47730
	R			39,11%	23,69%	31,24%			42,02%	32,30%	31,24%
Stellantis	€ EOP	80	11	28370	17000	14990	20	8	32830	20400	14990
	R			19,88%	9,67%	7,29%			22,97%	13,20%	7,29%
Banca Generali	€ EOP	60	6	40860	22580	12920	55	8	44240	24400	12920
	R			27,74%	15,22%	4,56%			29,51%	16,78%	4,56%
Intesa Sanpaolo	€ EOP	125	6	41670	27880	7020	125	4-3	42790	29490	7020
	R			28,17%	19,52%	-5,97%			28,77%	20,69%	-5,97%
Enel	€ EOP	124	4	32370	19670	10090	110	6	29890	18720	10090
	R			22,66%	12,49%	0,16%			20,98%	11,52%	0,16%

Nella tabella riassuntiva si può vedere che nei due terzi dei casi i valori sono migliorati, evidenziati in verde scuro, in seguito all'aggiunta di queste caratteristiche.

4. Conclusioni

Confrontando i diversi algoritmi utilizzati sull'arco temporale più recente e di nostro interesse 2017-2022, otteniamo la seguente tabella riassuntiva dove sono evidenziati in verde i risultati migliori.

Start = 10K € Annual Return= R		SMA		Momentum		Mean Reversion		Regression lineare		Regression logistica		Rete neurale		Rete neurale con più feature	
		TC= 0	TC=0.001	TC= 0	TC=0.001	TC= 0	TC=0.001	TC= 0	TC=0.001	TC= 0	TC=0.001	TC= 0	TC=0.001	TC= 0	TC=0.001
FTSE MIB	€ EOP	17000	16800	19871	15507	16375	14508	14117	7740	17126	9139	42560	20190	49280	23940
	R	9,67%	9,44%	12,68%	7,93%	8,96%	6,69%	6,18%	-4,36%	9,81%	-1,55%	28,64%	13,00%	31,97%	16,40%
S&P500	€ EOP	19100	18700	27793	23969	12788	11629	19977	10576	34391	30199	26330	15160	38540	30690
	R	11,91%	11,50%	19,46%	16,42%	4,37%	2,66%	12,79%	0,98%	23,96%	21,19%	18,34%	7,50%	26,44%	21,53%
Dow Jones	€ EOP	21600	21500	27867	23699	9551	8784	24315	12026	35645	30193	24410	15060	30710	21040
	R	14,33%	14,24%	19,51%	16,19%	-0,80%	-2,23%	16,71%	3,26%	24,74%	21,19%	16,79%	7,38%	21,55%	13,81%
NASDAQ	€ EOP	29700	29600	29766	25851	15889	13854	15939	10012	27852	20387	26900	14770	31650	16940
	R	20,84%	20,77%	20,89%	17,96%	8,39%	5,83%	8,45%	0,02%	19,50%	13,19%	18,78%	7,02%	22,19%	9,60%
DAX	€ EOP	17100	16300	14893	13706	14108	12743	16863	8142	15031	10980	24250	12420	15600	10450
	R	9,78%	8,87%	7,17%	5,64%	6,17%	4,31%	9,51%	-3,51%	7,34%	1,64%	16,66%	3,84%	8,04%	0,77%
FTSE 100	€ EOP	11800	11100	11571	10049	18873	16907	17090	10376	17177	9417	18410	8510	15050	7980
	R	2,92%	1,83%	2,57%	0,09%	11,68%	9,56%	9,77%	0,64%	9,87%	-1,04%	11,20%	-2,77%	7,37%	-3,85%
Nikkei 225	€ EOP	14100	13400	20548	16067	14161	12580	10927	5798	12235	10700	22060	11020	19660	9630
	R	6,16%	5,22%	13,34%	8,60%	6,24%	4,07%	1,55%	-9,04%	3,57%	1,18%	14,75%	1,70%	12,48%	-0,65%
Apple	€ EOP	50700	48300	65447	59634	26598	23614	97324	54328	135439	76517	66720	33950	75170	49990
	R	32,62%	31,51%	38,64%	36,42%	18,55%	16,12%	48,55%	34,22%	57,33%	42,46%	39,11%	23,69%	42,02%	32,30%
Stellantis	€ EOP	46200	44300	54891	44941	24934	19206	18825	9867	28748	16097	28370	17000	32830	20400
	R	30,49%	29,54%	34,46%	29,87%	17,22%	12,02%	11,63%	-0,23%	20,16%	8,63%	19,88%	9,67%	22,97%	13,20%
Banca Generali	€ EOP	20800	20500	22172	20143	9058	7445	25957	16601	16275	13830	40860	22580	44240	24400
	R	13,58%	13,30%	14,85%	12,95%	-1,71%	-5,00%	18,04%	9,22%	8,84%	5,80%	27,74%	15,22%	29,51%	16,78%
Intesa Sanpaolo	€ EOP	14600	14400	47644	38930	6160	5166	7580	4164	4461	2607	41670	27880	42790	29490
	R	6,80%	6,55%	31,19%	26,67%	-8,08%	-10,85%	-4,70%	-14,13%	-13,10%	-20,85%	28,17%	19,52%	28,77%	20,69%
Enel	€ EOP	28900	28300	28712	27448	13000	9605	18191	9234	25783	11997	32370	19670	29890	18720
	R	20,27%	19,83%	20,13%	19,20%	4,67%	-0,70%	10,97%	-1,38%	17,91%	3,22%	22,66%	12,49%	20,98%	11,52%
Rendimento medio annualizzato		14,95%	14,38%	19,58%	16,49%	6,30%	3,54%	12,45%	1,31%	15,83%	7,92%	21,89%	9,85%	22,86%	12,67%

È stato inoltre calcolato il rendimento medio annualizzato delle performance ottenute dai diversi modelli su tutti gli indici e le azioni analizzate. Si può affermare:

- Se non si considerano i costi di transazione in 8 casi su 12 gli algoritmi di machine learning e deep learning risultano migliori ma inserendo le commissioni sulle operazioni le performance ne risentono di molto
- Dal rendimento medio annualizzato, la rete neurale con più features risulta la migliore. Considerando i costi prevale la strategia momentum
- La mean reversion è la strategia che ha performato peggio e risulta la migliore solamente per il FTSE100 anche se di poco rispetto alla rete neurale
- I costi di transazione hanno un peso molto elevato sui risultati: anche se quelli considerati risultano solo dello 0,1% dell'operazione si possono osservare impatti notevoli
- La rete neurale ha ottenuto i risultati mediamente migliori seppur la struttura utilizzata sia stata ottimizzata solamente per l'indice FTSE MIB

Tra le 3 strategie più semplici basate esclusivamente sull'analisi tecnica sia il Momentum che la SMA hanno registrato buone performance con il backtesting ma è necessario procedere con cautela nell'interpretare questi risultati in quanto si è lavorato sull'intero set di dati identificando i migliori parametri. Solo con test su nuovi dati futuri si può constatare l'eventuale presenza di casi di *Snooping* e *Overfitting*.

Queste problematiche sono state in parte attenuate nel machine learning e nel deep learning grazie alla suddivisione dei dati in subset di addestramento e di test durante la preparazione. È stato così possibile verificare su questi ultimi le performance degli algoritmi cercando di individuare eventuali anomalie, per questo motivo i risultati ottenuti con il machine learning e il deep learning sono da considerarsi più affidabili nell'ottica di questa trattazione.

In particolare l'ultimo test eseguito sul deep learning ci fa capire come questo strumento possa essere potente per il fatto che può inglobare anche i risultati di altri algoritmi al suo interno e utilizzarli come parametri per determinare delle predizioni più complete e precise (es. aggiunta momentum).

Queste ultime verifiche consentono di affermare che, anche se è uno dei molti approcci al machine learning, il deep learning è un grande successo grazie al numero di parametri che possono essere considerati che permettono di riprodurre qualunque ipersuperficie delle soluzioni. Lo dimostrano i risultati straordinari ottenuti negli ultimi anni in un'ampia gamma di compiti considerati molto difficili in passato: ad esempio dando la possibilità ai computer di estrarre informazioni da immagini, video e suoni adeguatamente etichettati li si sta dotando di una sorta di percezione sintetica, seppur limitata.

L'aspetto più interessante del deep learning risiede nella sua semplicità: ogni osservazione viene trasformata in un vettore(vettorizzazione), che rappresenta un punto nello spazio geometrico, e tramite l'apprendimento incrementale di semplici trasformazioni geometriche viene mappata su un nuovo spazio. Ogni layer quindi non esegue altro che una semplice operazione geometrica e la catena di layer da vita a una complessa trasformazione, che in quanto differenziabile può essere ottimizzata.

Come visto nel corso della trattazione applicare tutte queste operazioni ai dati è un po' come ridistendere due fogli di colore differente accartocciati insieme: ***i modelli di deep learning sono quindi macchine matematiche in grado di ridistendere complessi problemi con l'utilizzo di dati a elevata dimensionalità.***[119]

Il termine rete neurale è da attribuirsi prettamente alla sua origine storica ma a poco a che fare con il cervello. Alcuni nomi più appropriati potrebbero essere:

- ***Apprendimento tramite rappresentazioni a livelli o gerarchiche***
- ***Modelli differenziabili profondi***
- ***Trasformazioni geometriche concatenate***[120]

Se da un lato questa semplicità è un vantaggio dall'altro fa emergere i punti deboli delle reti neurali:

- richiedono una potenza di calcolo decisamente maggiore rispetto agli altri algoritmi e più si complica il modello più questa necessità aumenta
- la definizione della miglior struttura della rete nella maggior parte dei casi è ancora affidata a dei processi di trial and error che richiedono molto tempo. Esistono alcuni metodi che permettono di migliorare e individuare alcuni iperparametri, come il tasso di apprendimento durante la discesa del gradiente, ma richiedono ulteriore capacità di calcolo
- il principale problema è che al momento nella maggior parte dei casi sono ancora considerate come delle black box per il motivo che si conosce poco delle trasformazioni che avvengono all'interno e questo rende difficile interpretare il loro funzionamento. Come detto in precedenza sono stati sviluppati negli anni procedure informatiche per le reti Convet, utilizzate per la risoluzione di problemi di classificazione di natura visuale, che con output visivi permettono di capire alcune trasformazioni e caratteristiche individuate dai singoli layer ma si è ancora ben lontani dal comprendere a pieno il loro funzionamento

[119] Francois Chollet; Deep Learning con Python; 2020; pag. 315
[120] Francois Chollet; Deep Learning con Python; 2020; pag. 316

Per il momento i limiti più grandi risultano essere quindi il carico computazionale disponibile, le tempistiche per ottimizzare il modello e la comprensione del lavoro della rete, anche se negli ultimi anni si son fatti passi da gigante soprattutto per i primi due aspetti.

In conclusione il machine learning e deep learning sono strumenti davvero potenti che presentano ancora molti margini di miglioramento e se usati correttamente possono supportare l'uomo in diverse scelte velocizzando il progresso in quasi ogni campo. Fino a quando il calcolo computazionale e il tempo investito non smetteranno di essere un limite sarà difficile procedere anche nell'interpretazione della rete e toccherà all'uomo scegliere il giusto trade-off tra risultati e complessità del modello: in alcuni casi se la rete neurale garantisse solo un piccolo vantaggio potrebbe essere meglio applicare un algoritmo più semplice, che richiede meno potenza e tempo per il calcolo, a più casi invece che ottenere il massimo da uno solo.

Bibliografia

- J. J. Muprhy, Analisi tecnica dei mercati finanziari; 2° edizione
- S. Raschka & V. Mirjalili; Machine Learning con Python; 2020
- Francois Chollet; Deep Learning con Python; 2020
- Yves Hilpisch, Trading con Python; 2021

Ringraziamenti

Ringrazio la Dottoressa Rondi e il Dottor Varetto per avermi seguito e per avermi indicato questo argomento come tesi. Mi ha permesso di aprire gli occhi su un mondo nuovo in evoluzione che non mancherò di continuare ad approfondire e studiare in futuro.

Ringrazio i miei responsabili aziendali passato e presente, Giuseppe e Silvia, per avermi insegnato molto e aver capito e assecondato le mie necessità tra esami/tesi, spero di aver ricambiato dando sempre il massimo durante il lavoro.

Ringrazio il gruppo "Sbarra Sbarra Sbarra", compagni di tante serate davanti al pc dove tra chiacchere e risate riuscivo a distrarmi per un attimo dai numerosi impegni e pensieri.

Ringraziamento principale alla mia famiglia: mia Madre che anche se non è più tra noi son sicuro sia stata al mio fianco aiutandomi nei momenti più difficili, mio Padre la persona più forte che conosco che nonostante i molti problemi e difficoltà avuti ha sempre trovato il modo per aiutarmi e dedicarmi parole buone e di conforto, il mio fratellino che nonostante la sua giovane età ha sempre trovato il tempo per regalarmi un sorriso nonostante tutto quello che ha passato e la mia futura moglie Martina per essermi stata accanto in tutti questi anni condividendo da vicino mie gioie e dolori. Tutti voi mi avete supportato e soprattutto "sopportato" anche nei momenti più difficili, avete sempre creduto in me, appoggiato le mie scelte e siete stati la forza che mi ha permesso di portare a termine questo difficile percorso di lavoro-studio.