POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

Executive Summary of the Thesis

STRATEGIE DI PORTAFOGLIO E PAIR TRADING: L'APPORTO DI RETI NEURALI LONG-SHORT TERM MEMORY

Laurea Magistrale in Mathematical Engineering - Ingegneria Matematica

Author: Guglielmo Zaffaroni

Advisor: Prof. Roberto Baviera

Academic year: 2020-2021

1. Introduction

Pairs-trading is a quantitative statistical arbitrage trading strategy based on the presumed profitability of situations where two assets with similar past performances begin to behave in a different manner, showing opposite market patterns. In the hypothesis that this trend is just a temporary deviation due to market inefficiencies, the undervalued stock is bought, while the overvalued one is sold. In this way, it is possible to make a profit with the reconvergence of the prices. It's important to underline that the long and the short positions are of the same value, so that the total strategy is market neutral and substantially self-funding.

In this work we study in deep the pairs-trading strategy proposed originally by Flori and Regoli in [2] based on a particular classification of stocks in the S&P 500 index described below. One of the most important peculiarities of their research is the use of machine learning techniques for the analysis of time series. These methods are spreading in the fields of algorithmic trading thanks to their capacity of extracting valuable information from financial data which helps to choose the best trades.

Flori and Regoli use the cointegration framework to select stocks that have moved together in the past as in [5]. Then they estimate the probability that the return difference between a stock and their similar ones will increase in the near future through an LSTM network. The values obtained are ordered, divided into deciles, and the first decile is sold while the tenth one is bought.

In this work, we aim initially to reproduce the trading algorithm, analyze the basics results, and enrich the study by evaluating the patterns in the stocks selected for trading and the strategy robustness with respect to some parameters. Afterwards, we evaluate the performance in the context of high-frequency trading. There is a growing interest and application of this practice in the finance community, but there are still few studies in the scientific literature about it. An interesting review about pairs-trading in a high-frequency framework can be found in [4]. In particular, we use data with minute frequency. Furthermore, in order to include trading costs, we analyze also bid and ask prices. In this way, we can exploit this additional information to better select stocks for trading and evaluate net strategy profitability.

2. The trading strategy

We consider all the stocks belonging to the S&P 500 index from January 2000 to June 2019 initially using daily data. We use cointegration to identify the stocks with similar past performances. Indeed, two series are cointegrated if the series of their difference follows a stationary process, meaning that the deviations from equilibrium are short-term and will be timely corrected. For every year of analysis, we test for cointegration every possible couple of stocks, using Engle-Granger's and Johansen's tests. Then we define the cointegration groups of a stock as the set of stocks cointegrated with the former one.

Then we use a new indicator, proposed in [2], in order to identify a pairs-trading opportunity. Differently from the classical procedure, we do not evaluate the gap in the prices or returns between two assets, but we compute the probability that the return difference between a stock and its peers (cointegration group) will increase in the next day. This is done thanks to the use of an LSTM neural network that solves a binary classification problem. In particular, defining r_t^i the return of stock i at day t, $r_t^{CG_i}$ the mean return of the cointegration group of stock i at time t and $\Delta r_t = r_t^i - r_t^{CG_i}$, the neural network computes the probability of $y_{t+1} = 1$, where

$$y_{t+1} = \begin{cases} 1 & \text{if} \quad d_{t+1} \geq 0 \\ 0 & \text{if} \quad d_{t+1} < 0 \end{cases}, \tag{1}$$

and $d_{t+1} = \Delta r_{t+1} - \Delta r_t$. The main peculiarity of the proposed procedure is that at all time t it exploits all the information contained in the time series up to period t. Thus it considers also the stock's recent market history and stock's relative behavior with respect to its peers in order to improve the prediction. To do so it uses data concerning returns, trading volumes, and discrepancies with respect to similar stocks for the previous 240 working days.

Every day we obtain a probability value for every stock belonging to at least one cointegration group. We order these values and split them into deciles, thus splitting the pool of stocks. Then the portfolio is created buying in the same amount the last decile (Top), which represents the stocks with the greatest probability to increase the returns with respect to the peers, and selling the first one (Bottom), which contains the stocks with the lowest probability.

3. Daily results

The cointegration procedure shows that on average, around 320 stocks are cointegrated with at least another one, and half of the groups contain less than 3 stocks. These results are fluctuating across the years, consistently with the meaning of cointegration, they increase in the crisis periods (dot-com bubble and 2008 financial crisis) and decrease in the other ones.

The LSTM network is perfectly able to perform the classification, showing very high values for the metrics used to evaluate it: accuracy, area under ROC, and logloss. The average results through the years of analysis for the whole sample of stocks, and the Top-Bottom portfolio (the one used for trading) are shown in Table 1.

Classification metrics

	Accuracy	Area ROC	Logloss
All	0.752	0.831	0.507
T-B	0.933	0.949	0.241

Table 1: Classification metrics for the whole sample of stocks, and the Top-Bottom portfolio.

This precision in the classification results determines positive abnormal return for the pairs-trading strategy, which gives an average annualized performance of 23.25%, both statistically and economically significant. Furthermore, evaluating the performance obtained by an illustrative investment strategy that equally invests in every decile, we show that to a greater probability of increasing return there correspond an actual greater absolute return. This behavior is shown in Figure 1. Finally, we show that the Top-Bottom strategy, though investing in more volatile stocks, is optimal also from the point of view of risk analysis. Indeed it has a much lower standard deviation of returns than any other decile.

The strategy turns out to be more profitable during the periods of market turmoils and its performance decreases in the recent period. These two aspects are coherent with the vast ma-

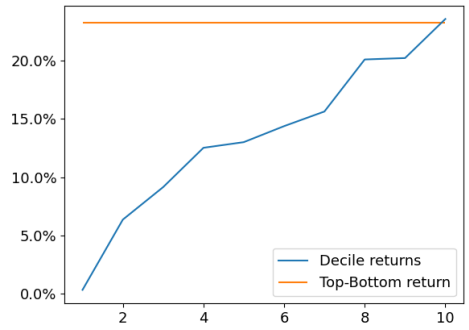


Figure 1: Annualized average returns of the strategy which equally invest in every decile portfolio and of the Top-Bottom strategy.

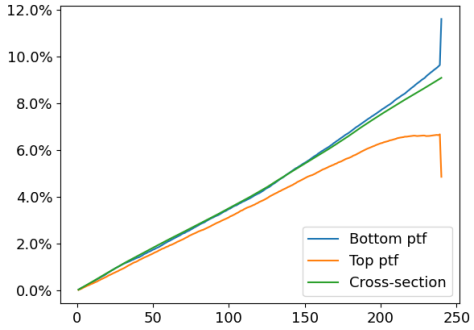


Figure 2: Stocks mean cumulative returns for the 240 days before trading.

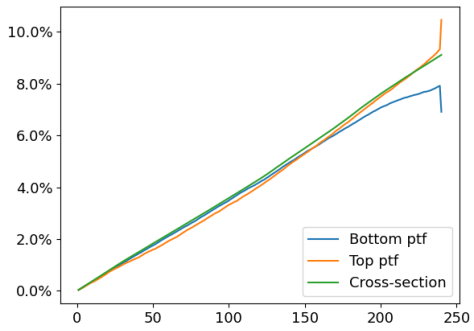


Figure 3: Cointegration groups mean cumulative returns for the 240 days before trading.

jority of the literature on pairs-trading. Indeed during down-market periods volatility increases and misvalued assets occur more often helping this kind of strategies. At the same time, there are more limits to arbitrage thus pairs-trading opportunities, which cannot be exploited, remain in historical data [3]. In the last years, it seems reasonable that the rise of computational power and its widespread availability caused a diffusion of this kind of strategies that eroded gradually their profitability [1].

Neural networks are often seen as black-box, so we investigate which are the common market patterns in the stocks selected for trading, and we check if the sources of profitability are linked to the short-term reversal of the relative value. To do so we compute the mean cumulative returns considering the last 240 instants for the stocks belonging to the first or tenth decile. We compute the same quantities for the mean cumulative returns for the cointegration groups. The results are plotted in Figure 2 and Figure 3 with respect to the cross-sectional average. The stocks belonging to the Top portfolio show a below mean momentum, and a crash in value in the last day, while the Bottom stocks show an average momentum and a sudden rise in the last day before trading. More or less the same behavior, but reversed, is observable in the cumulative returns of the cointegration groups. So the neural networks look for a double short-term reversal, in the Top and Bottom stocks, at the same time in the single asset and in its peers. It gains abnormal positive returns from these trends.

The proposed strategy proves to be profitable independently of the parameters. Indeed varying some parameters values it is still able to experience abnormally high returns. Particularly we notice that the cointegration specifications are the ones that influence the most the performance: the inclusion of a constant in the equation reduces the profitability to 14.2%, and an increase in the test's significance reduces it to 21.6%. Reducing the number of timestamps used in the time series to perform the prediction does not modify performance significantly. On the one hand, it confirms that the driving forces for pairs-trading are short-term, and on the other hand it allows to reduce the number of timestamps included in the model, thus decreasing the computational costs which can be very high.

In the end, by introducing a transaction cost of 5 bp, which is a common value in the scientific literature for such liquid stocks, we verify that the strategy's profitability is eroded, and the performance turns into negative values.

4. High-frequency, and trading costs

High-frequency trading is an increasingly common form of trading in the finance community nowadays, so we decided to evaluate the strategy using data with minute-frequency. In detail, we use the data relative to the stocks composing the S&P 500 index in the period from 01-03-2021 to 20-08-2021.

Besides we introduce trading costs in order to evaluate net profitability. To do so we exploit bid and ask prices that give additional important information to the LSTM network and beyond, and allow a more precise evaluation of the performance net to trading costs.

The strategy remains unchanged, apart from the inclusion of the bid-ask spread in the neural network input data. The number of cointegrated stocks and the groups median size increase to a mean value of 458 and 10 respectively. This behavior is coherent with the data frequency. Indeed during such a short time interval, there are no economic reasons that move the prices. It's the traders hedging that influence in such a strong way all the stocks in the same direction. The Top-Bottom precision metrics are even better, showing values of 0.947 for the accuracy, 0.961 for the area under ROC, and 0.202 for the logloss. The strategy produces a statistically and economically significant profit: the mean gross return for the single trade is 1.62e-4 that is equivalent to an astonishing 6.33% on a daily basis. The profit is even higher if we consider a longer holding period. For example, a 30-minute investment horizon generates a 2.089e-4 mean return per trade, while a 60 minute generates a 2.1e-4 mean return.

The indicator that the LSTM network tries to predict gives the probability that the return difference between a stock and its peers will increase in the near future. However, this quantity does not contain information about the effective profitability of the trade. Indeed it is a relative measure, not an absolute one. So we decided, using a double sorting procedure, to include the probability that a stock will increase in value. This is done using the indicator proposed in [1], which predicts if a stock return will be above or below the cross-sectional average in the next time instant. Doing so we construct a new strategy that buys the stocks with a high probability to overperform the peers, while having a positive return, and sells the stocks with a high probability to underperform the peers, while having a negative return. This strategy generates a mean excess return equal to 2.15e-4, confirming the idea that this new indicator can supplement the information contained in the original one, in order to improve performances.

It's interesting to verify if the abnormally positive returns survive the introduction of trading costs. From the point of view of hedge funds and investment banks, commissions and short-selling fees are negligible, the performance erosion comes exclusively from bid-ask spreads. So, in order to have a true estimate of the profits, we evaluate the returns using the bid and ask prices. We consider buying a stock at the momentaneous ask price, selling it at the bid for the long position, and vice versa. Doing so the strategy overall performance becomes negative: -1.39e-3, -1.36e-3 and -1.35e-3 respectively for 1, 30 and 60 minutes investment horizon. One of the main motivations is that the target variable doesn't reward a small spread or a positive net performance (d is evaluated using close prices). Moreover, we notice that the neural network tends to classify in the extreme deciles the more volatile stocks that are also the less liquid, increasing further the costs.

In light of this, we try to control the spread of the traded stocks. To do so we use a double sorting procedure, where one indicator is the usual one, and the other is the spread at the previous time instant. Clearly, we buy the stocks with high probability and small spread, and we sell the ones with low probability and small spread. Doing so we obtain a net performance of about -4.1e-4 that corroborates the idea of using liquid stocks in order to minimize costs. Since this procedure still does not reach positive performance, we modify the strategy, proposing a new target variable. It aims to find those stocks which show an increase in relative return higher than those of the cointegrated ones, and at the same time increasing more than a threshold. In particular, it is defined as:

$$y_{t+1} = \begin{cases} 0 & \text{if} \quad d_{t+1} < -\frac{S}{2} \\ 1 & \text{if} \quad |d_{t+1}| \leq \frac{S}{2} \\ 2 & \text{f} \quad d_{t+1} > \frac{S}{2} \end{cases}$$

where $d_{t+1} = \Delta r^i_{t+1} - \Delta r^i_t$, and S represents

the threshold. Clearly, we buy those stocks that show a high probability to belong to the class 2, and we sell those that show and high probability to belong to class 0. The threshold is estimated as the 50-th and 75-th percentile of a stock spread series. In this way, the LSTM selects for trading those stocks that have a high probability to increase or decrease in relative value with respect to their own spread. The results improve, indeed the returns are -1.19e-3 and -1.12e-3, but not enough to overcome trading costs.
In the end, we evaluate the impact of rebalancing the portfolio. Indeed in this kind of strategy, the asset turnover can be consistent and deteriorate the performance net to trading costs. We consider buying or selling a stock only if it changes decile from a time instant to the next one. In this way, the costs are not taken into account more than once. The profits get better, but not enough, reaching values of -1.26e-3, -1.18e-3, and -1.18e-3 for 1, 30, and 60 minutes. Moreover, we notice that there is no persistence in the decile membership: a stock remains in the previous group around 10% of the time, which is the value associated with the random probability.

5. Conclusions

The proposed strategy is able to generate positive excess returns, more than 23% on annual basis, thanks to the very high ability of the LSTM neural network to perform the classification, see Table 1. Also in the context of high-frequency trading, it gives an amazing 6.3% daily return, that can be improved taking into account other factors such as a longer investment horizon. Unfortunately, the strategy is not profitable beyond the trading costs, which are consistent in pairs-trading, because of the high portfolio turnover. Further modifications which consist in controlling for the spread, or looking for the stocks that gain more than the trading cost, improve the net performance but fail to obtain net profit.

References

[1] Thomas Fischer and Christopher Krauss. Deep learning with long short-term memory networks for financial market predictions. *European Journal of Operational Research*, 270(2):654–669, 2018.

[2] Andrea Flori and Daniele Regoli. Revealing pairs-trading opportunities with long short-term memory networks. *European Journal of Operational Research*, 295(2):772–791, 2021.

[3] Heiko Jacobs and Martin Weber. On the determinants of pairs trading profitability. *Journal of Financial Markets*, 23:75–97, 2015.

[4] Johannes Stübinger and Jens Bredthauer. Statistical arbitrage pairs trading with high-frequency data. *International Journal of Economics and Financial Issues*, 7(4):650–662, 2017.

[5] Ganapathy Vidyamurthy. *Pairs Trading: quantitative methods and analysis*, volume 217. John Wiley & Sons, 2004.

POLITECNICO DI MILANO
Scuola di ingegneria industriale e dell'informazione

Laurea Magistrale in Ingegneria Matematica:
Finanza Quantitativa

STRATEGIE DI PORTAFOGLIO E PAIR TRADING: L'APPORTO DI RETI NEURALI LONG-SHORT TERM MEMORY

Relatore:
Prof. Roberto Baviera

Candidato:
Guglielmo Zaffaroni - 928243

Anno Accademico 2020-2021

Abstract

Questo lavoro analizza in dettaglio una particolare strategia di pair trading, proposta originariamente da Flori e Regoli (2021). Essa fa uso della cointegrazione come tecnica per individuare le azioni con comportamento passato simile, e genera segnali di trading tramite una rete neurale Long-Short Term Memory. In particolare la rete utilizza dati giornalieri e predice la probabilità che un'azione abbia differenza di rendimenti crescente rispetto alle azioni simili.

L'analisi condotta in questa tesi da un lato conferma i risultati del lavoro originario, dimostrando che la strategia riesce a prevedere con elevata precisione la dinamica relativa dei titoli e genera rendimenti in eccesso rispetto al mercato, dall'altro lo arricchisce mostrando che i risultati non dipendono significativamente dai parametri scelti, e approfondendo lo studio dei rendimenti dei titoli selezionati per la negoziazione.

Inoltre l'analisi si concentra su alcuni aspetti non trattati in precedenza. Dato il crescente interesse del mondo finanziario rispetto al trading ad alta frequenza, si studiano i risultati della strategia utilizzando dati al minuto. Anche in questo contesto essa dimostra ottime performance di mercato, significative da un punto di vista sia economico che statistico. Essa viene poi migliorata prendendo in considerazione altri indicatori, con l'obiettivo di aumentare i profitti.

Un ultimo aspetto innovativo della tesi consiste in un'analisi dei risultati al netto dei costi di trading. Attraverso l'utilizzo dei prezzi bid e ask si mostra che a causa dell'elevato ricambio di titoli all'interno del portafoglio i profitti vengono erosi completamente, e inoltre che la profittabilità complessiva dipende fortemente dalla liquidità dell'azione che diminuisce lo spread.

Parole chiave: Pair trading, Machine learning, Neural networks, Trading ad alta frequenza.

Indice

Elenco dei simboli

$\frac{\partial L}{\partial \mathbf{W}}$	Matrice contenente tutte le derivate della loss rispetto ai pesi contenuti in $\mathbf{W}$
$\frac{\partial L}{\partial w_i}$	Derivata della funzione loss rispetto all'i-esimo peso
$\hat{x}$	Stimatore della quantità x
$\mathbf{x}'$	Vettore $\mathbf{x}$ trasposto
$\mathbb{E}$	Valore atteso di una variabile casuale
∇	Gradiente di un vettore
$\otimes$	Prodotto elemento per elemento di due vettori
$\Phi(\cdot)$	Generica funzione di attivazione
$\mathbf{w}$	Insieme dei pesi trainabili
h_t	Valore dello stato nascosto al tempo t
L	Funzione di loss
l_i	Funzione di loss valutata sull'i-esimo dato
$P(\Delta r \nearrow)$	Probabilità che la differenza di rendimenti aumenti
SSR	Somma dei residui al quadrato
T	Numero di istanti in una serie temporale
u_t	Termine di errore al tempo t
Y_t	t-esimo istante temporale della serie storica $\mathbf{Y}$

$\mathbf{B}$	Processo di Wiener multidimensionale
b	Bias relativo a un nodo della rete neurale

Acronimi

ADF Augmented Dickey-Fuller. 15, 20, 59, 60

AI Artificial Intelligence. 6

AIC Akaike Information Criterion. 13, 15

ANN Aritificial Neural Networks. 27

AR Autoregressive Model. 12, 17

AUC Area Under the Curve. 68

BIC Bayes Information Criterion. 13, 15, 17, 60

BPTT Back Propagation Through Time. vi, 43, 64

CAPM Capital Asset Pricing Model. 47, 48

EMH Efficient Market Hypothesis. 3

FN False Negative. 65

FNN Feed-forward Neural Network. 29, 42, 43

FP False Positive. 65

GLS Generalized Least Squares. 25

HAC Eteroskedasticity- and Autocorrelation-Consistent. 23, 25

HFT High-Frequency Trading. 6, 89

LSTM Long-Short Term Memory. vi, 46, 57, 61–64, 74, 75, 94

NN Neural Networks. 27, 28, 35, 41, 80

RNN Recurrent Neural Network. 41–43

ROC Receiver Operating Characteristic. 67

TN True Negative. 65

TP True Positive. 65

TPBTT Truncated Back Propagation Through Time. 43

VAR Vector Auto Regression. 16, 17

VECM Vector Error Correction Model. 19, 20

Elenco delle tabelle

Elenco delle figure

Introduzione

Il pair trading è una particolare strategia di investimento che rientra nella categoria degli arbitraggi statistici. Nella sua versione più semplice si basa sull'identificazione di una coppia di titoli su cui prendere una posizione lunga e una corta di pari importo, di modo che il portafoglio complessivo sia *self-financing*, ovvero non richiede nuovi apporti di capitale durante l'implementazione della strategia.

La strategia che viene analizzata in questo lavoro è stata proposta originariamente nel lavoro di Flori e Regoli (2021) in cui una delle più importanti peculiarità riguarda l'utilizzo di tecniche di *machine learning*. Esse stanno sempre di più prendendo piede nel mondo finanziario grazie al crescente sviluppo tecnologico, alla disponibilità di grandi quantità di dati e alla possibilità di usarle per effettuare previsioni consentendo una migliore selezione degli scambi. In particolare vengono utilizzate reti neurali *long-short term memory*, note per le potenzialità nell'analisi delle serie storiche.

L'obiettivo di questo lavoro di tesi è quello di approfondire la ricerca iniziata dagli autori. Più in dettaglio si cerca inizialmente di replicare l'algoritmo da loro ideato, valutandone le performance, e si mira ad arricchire lo studio approfondendo la dinamica dei titoli selezionati per il trading, individuando le cause di eventuali profitti e studiando la robustezza di questi ultimi rispetto ad alcune caratteristiche della strategia.

Le principali novità introdotte in questo studio sono due e riguardano l'utilizzo di dati ad alta frequenza, in particolare ogni minuto, pratica che ormai ha raggiunto un'importanza fondamentale nel mondo finanziario, e l'impiego dei prezzi bid e ask per valutare le performance del portafoglio al netto dei costi di intermediazione. Nella letteratura riguardante il pair trading esistono ancora pochi studi relativi a questi due aspetti che sono però imprescindibili nell'ambito del trading algoritmico.

Il lavoro è strutturato come segue:

Il Capitolo 1 offre una panoramica più dettagliata del contesto teorico in cui si pone il lavoro, definendo alcune nozioni fondamentali per il suo sviluppo e riesaminando parte dei principali contributi scientifici a riguardo.

Il Capitolo 2 è dedicato alla presentazione delle principali tecniche utilizzate nell'analisi delle serie storiche, in particolare si sofferma sulla cointegrazione, utilizzata per la selezione delle coppie per il pair trading.

Il Capitolo 3 dà un quadro complessivo sulle reti neurali, soffermandosi inizialmente sulla loro struttura generale, analizzando il funzionamento degli algoritmi di addestramento e i problemi che ne derivano, e infine presentando le reti *long-short term memory* utilizzate nel lavoro.

Il Capitolo 4 introduce brevemente i principali modelli finanziari, generalizzazioni successive del CAPM come quella introdotta in Fama e French (1993), utilizzate per valutare le performance della strategia.

Il Capitolo 5 illustra in dettaglio la strategia di trading utilizzata, fornendo i particolari necessari all'implementazione dell'algoritmo e descrivendo le tecniche usate per la sua valutazione.

Il Capitolo 6 presenta e commenta i principali risultati ottenuti nel trading giornaliero e approfondisce l'analisi condotta nel lavoro di riferimento.

Il Capitolo 7 mostra le modifiche necessarie per l'applicazione della strategia nel contesto dei dati ad alta frequenza e ne presenta i risultati principali. Inoltre propone diverse variazioni della strategia stessa al fine di massimizzare i profitti sia al lordo che al netto dei costi, analizzando quali sono i principali fattori che influiscono sulle performance.

Il Capitolo 8 riassume lo studio, offre dei commenti conclusivi e indirizza verso possibili ricerche future.

Capitolo 1

Inquadramento generale e revisione della letteratura

1.1 Teoria dei mercati efficienti

La teoria dei mercati efficienti EMH è un'ipotesi fondamentale nel contesto dell'economia finanziaria. Essa afferma che i mercati sono efficienti, cioè il prezzo delle attività scambiate sul mercato riflette e incorpora tutte le informazioni rilevanti disponibili (Fama, 1970). Pertanto, l'andamento storico dei prezzi delle azioni non può essere utilizzato per prevedere quello futuro. I movimenti del mercato azionario sarebbero determinati esclusivamente da informazioni future e non prevedibili. Ne consegue anche che i rendimenti azionari sono processi stocastici senza memoria e i prezzi reagiscono immediatamente quando nuove informazioni diventano disponibili al mercato. Non è realistica la prospettiva di "battere il mercato" in maniera sistematica, essendo le azioni scambiate al loro valore corretto (*fair value*).

Nel corso del tempo investitori e ricercatori hanno contestato ampiamente la teoria del mercato efficiente sia da un punto di vista teorico che empirico, dimostrando che il mercato è imperfetto perché nella realtà esistono fenomeni, definiti anomalie di mercato, attraverso i quali è possibile fare previsioni anche in mercati molto liquidi. Per esempio gli economisti comportamentali attribuiscono queste imperfezioni a una combinazione di pregiudizi cognitivi, quali eccessiva sicurezza, reazioni esagerate oppure altri comportamenti umani irrazionali nel processo di analisi delle informazioni (Jegadeesh e Titman, 1995). L'insieme di questi fattori può rendere prevedibile l'andamento dei

prezzi. Se il mercato è imperfetto, in alcune situazioni è possibile approfittare delle vulnerabilità di mercato per realizzare rendimenti anomali positivi.

Questo lavoro si sviluppa nell'ambito della teoria dei mercati imperfetti con l'obiettivo di predire l'andamento futuro dei prezzi nel mercato azionario, in particolare quello dell'indice S&P 500.

1.2 Short-term reversal

Uno dei fenomeni più studiati in questo contesto (cfr. Fama (1965), Jegadeesh (1990) e Jegadeesh e Titman (1993)) è il *reversal effect*, o effetto di inversione, secondo il quale le azioni che hanno avuto cattive performance recenti probabilmente avranno un'inversione di tendenza dei prezzi nel futuro a breve termine e viceversa. In tal senso gli andamenti di mercato passati possono influenzare le aspettative degli operatori di mercato. La spiegazione principale che viene fornita per questo fenomeno è quella per cui il prezzo delle azioni reagisce esageratamente a nuove informazioni relative alle società, il che porta a un *mispricing* momentaneo e a una conseguente correzione del prezzo nella direzione opposta (Jegadeesh e Titman, 1995). Esiste anche un'altra spiegazione relativa ad effetti di liquidità, la quale prevede che i profitti provenienti da queste reversioni siano un compenso per i *market makers* in cambio del servizio di fornitura di liquidità e del conseguente rischio di inventario (Avramov et al., 2006).

Il comune denominatore delle due motivazioni del reversal effect è che gli andamenti passati possono influenzare i comportamenti di chi opera sul mercato e rendere prevedibili gli andamenti futuri. Ciò permette quindi la creazione di *contrarian strategy*, ovvero di strategie di arbitraggio statistico, caratterizzate da acquisti e vendite in opposizione all'andamento momentaneo del titolo. In altre parole, un operatore è portato a vendere azioni che hanno avuto buone performance e comprare azioni che ne hanno avute di scarse, attendendosi un'inversione di tendenza.

1.3 Pair trading

Un'altra strategia d'investimento, che rientra nelle categoria degli arbitraggi statistici, è quella del *pair trading*. Questa strategia è neutrale rispetto all'andamento di mercato, in quanto permette all'operatore di guadagnare da

ogni situazione di mercato (al rialzo, al ribasso oppure stagnante). Si tratta di un arbitraggio statistico, ovvero una strategia prettamente quantitativa, che sfrutta il concetto di *mean reversion* per investire in un gruppo di azioni per un intervallo di tempo limitato. Il pair trading è in genere attribuito a un operatore di Morgan Stanley (Nunzio Tartaglia) che negli anni '80 si avvaleva dell'aiuto di un gruppo di matematici, fisici e analisti computazionali per sviluppare una strategia di arbitraggio quantitativa e automatizzata, che consisteva nello scambiare azioni in coppie (Vidyamurthy, 2004). Con la strategia del pair trading vengono individuate coppie di azioni affini, cioè con performance passate simili, e si sfruttano situazioni in cui le azioni della coppia mostrano andamenti di mercato opposti. In particolare, una volta individuate le coppie, vengono identificati i momenti in cui questa somiglianza si indebolisce e il prezzo di un'azione aumenta mentre quello dell'altra diminuisce. L'operatore compra pertanto l'azione che ha perso valore e vende allo scoperto quella che ne ha acquisito. La divergenza all'interno della coppia può essere causata da molteplici motivi, come cambiamenti temporanei di domanda/offerta, grossi ordini di acquisto/vendita per uno dei due asset, oppure dalla reazione a notizie importanti riguardanti una delle due società. Nell'ipotesi che le performance future assomiglino a quelle passate, questo prezzo relativo "errato" è solo temporaneo e al momento della riconvergenza dei due prezzi si genera un profitto. In tal senso queste situazioni possono essere interpretate come una reversione verso la media del prezzo relativo.

Il pregio di questa strategia è triplice: la posizione corta e lunga nella coppia per lo stesso controvalore consentono di costruire un portafoglio che è sostanzialmente autofinanziato. Inoltre, il portafoglio è neutrale alle dinamiche di mercato; se entrambi i titoli della coppia subissero uno stesso shock di segno positivo o negativo le perdite e i guadagni si compenserebbero. Infine, mentre i prezzi di una singola azione sono difficili da prevedere, c'è una forte evidenza per cui è possibile farlo per la differenza relativa dei prezzi di una coppia.

Per questi motivi nel corso del tempo ricercatori e professionisti hanno studiato numerose tecniche per costruire strategie di trading fondate sull'idea di valore relativo di due asset (cf. e.g. Vidyamurthy (2004), Gatev et al. (2006), Bertram (2009) Jacobs e Weber (2015) e Baviera e Baldi (2019)).

1.4 Intelligenza artificiale

L'intelligenza artificiale (AI) si sta sviluppando a una velocità sempre crescente. Negli ultimi decenni è stata ampiamente utilizzata in molti campi, entrando nella nostra vita quotidiana. Nell'industria finanziaria l'intelligenza artificiale sta avendo un forte impatto, riuscendo a produrre un valore addizionale superiore ai metodi tradizionali. In particolare le principali applicazioni sono quelle relative al *credit scoring*, alla prevenzione delle frodi, alla consulenza digitale, detta anche *robo-advisory*, e al trading algoritmico, tema portante di questo lavoro.

Gli algoritmi di trading sfruttano tecniche per la previsione di variabili finanziarie, utilizzando metodi quantitativi per determinare indicazioni operative poi eseguite da computer.

Negli anni più recenti, grazie all'aumento della potenza di calcolo delle macchine e della loro diffusa disponibilità, questi metodi sono diventati sempre più complessi e articolati, integrando anche tecniche di *machine learning*, grazie alle quali i risultati analitici e gli scambi conseguenti sono più profittevoli. Da questo deriva una loro crescente diffusione: addirittura si stima che il numero di scambi gestito da qualche tipo di sistema supportato da tecniche di intelligenza artificiale, per quanto riguarda il mercato equity americano nel 2020, sia compreso tra il 60 e il 73% del totale [1]. Per mezzo dell'AI sono sempre più frequentemente analizzati dati finanziari sia per sfruttare opportunità di arbitraggio statistico (Atsalakis e Valavanis (2009), Patel et al. (2015), Heaton et al. (2017) e Krauss et al. (2017)) che per ricerche più generiche in ambito finanziario (Kraus et al. (2020)).

Per tutti i motivi sopra elencati le analisi svolte in questo lavoro sono state eseguite anche attraverso mezzi di intelligenza artificiale, in particolare reti neurali.

1.5 High-Frequency Trading

Il trading ad alta frequenza HFT è un tipo di trading algoritmico caratterizzato da alte velocità di esecuzione, alti controvalori scambiati e orizzonti temporali di investimento molto brevi, che variano dalla manciata di millisecondi a qualche ora. Tutto ciò è gestito da algoritmi altamente sofisticati

[1] https://www.mordorintelligence.com/industry-reports/algorithmic-trading-market

che analizzano contemporaneamente dati di uno o più mercati ed eseguono gli ordini in base alle specifiche condizioni di mercato.

Il trading ad alta frequenza si è sviluppato lentamente dopo che nel 1983 il NASDAQ ha introdotto una forma di negoziazione puramente elettronica. Nei primi anni 2000 il tempo medio di esecuzione di un ordine era di parecchi secondi e la proporzione di ordini nel mercato equity gestita da algoritmi elettronici era inferiore al 10%. Negli anni successivi questi dati sono cambiati sensibilmente grazie soprattutto all'enorme sviluppo tecnologico e all'introduzione di incentivi per le compagnie che aumentavano la liquidità sul mercato. Si pensi che oggi i tempi di esecuzione sono crollati a qualche millisecondo o addirittura microsecondo, mentre il volume degli scambi elettronici ha superato il 70% del totale in alcuni mercati quotati (Aldridge (2013) e Haldane (2010)). La progressiva diffusione del trading ad alta frequenza ha fortemente influenzato il mercato, migliorandone sensibilmente la liquidità, rimuovendo gli spread tra prezzi bid e ask e riducendone la volatilità. D'altra parte però questa nuova realtà non è esente da critiche, in quanto utilizza modelli matematici e algoritmi per prendere decisioni e operare sui mercati, eliminando, o fortemente riducendo, la componente umana. Ciò può provocare importanti movimenti di mercato che non sono dovuti a ragioni economiche evidenti: si pensi al *flash crash* del 6 maggio 2010, durante il quale il Dow Jones Industrial Average ha perso il 10% in soli 20 minuti per poi risalire nuovamente.

Indipendentemente dagli effettivi vantaggi per il mercato, il trading ad alta frequenza è una pratica che si sta sempre di più diffondendo tra i principali operatori del settore (Goldman Sachs e Morgan Stanley tra i più noti) e sempre più è oggetto di studio nella letteratura scientifica (cf. e.g. Cartea et al. (2015) e Aldridge (2013)).

Per queste ragioni le simulazioni svolte nel presente lavoro sono state condotte utilizzando dati sia con frequenza giornaliera che al minuto.

1.6 Revisione della letteratura e obiettivo del lavoro

Il presente lavoro si basa su una particolare strategia di pair trading con compravendita di titoli dell'indice S&P 500 su base giornaliera. Questa strategia è stata originariamente proposta nel lavoro di Flori e Regoli (2021). Tra

i numerosi metodi citati in letteratura al fine di identificare azioni con un comportamento simile (si veda Krauss (2017) per una sintesi) gli autori utilizzano la cointegrazione, seguendo l'esempio di Vidyamurthy (2004) e Rad et al. (2016). La differenza tra due serie di prezzi di titoli cointegrati, per definizione, segue un processo stazionario ovvero, nel caso di deviazioni dalla media, la differenza tende asintoticamente a tornare al valore di equilibrio.

Mediante test di cointegrazione è possibile individuare i titoli che mostrano un andamento cointegrato ad almeno un'altra azione e dai quali possono scaturire opportunità di pair trading. Ovviamente solo questi sono interessanti per un'analisi basata su questo tipo di strategie di trading. D'ora in poi un titolo viene detto cointegrato se cointegrato con almeno un altro.

Nelle strategie più tradizionali, innanzitutto si identifica una coppia di titoli cointegrati e si effettua una negoziazione nei momenti in cui si osserva una divaricazione nei prezzi o nei rendimenti. In particolare si compra il titolo che ha sottoperformato e si vende quello che ha sovraperformato, dando per scontato che il primo sia destinato a salire e il secondo a scendere. La letteratura ha proposto svariate tecniche per identificare le condizioni di mercato che spingono a una compravendita, basate principalmente sul divario tra i prezzi e tra i rendimenti della coppia, come in Gatev et al. (2006), Rad et al. (2016) e Chen et al. (2019).

Le strategie di pair trading appena analizzate si basano su tecniche di analisi di serie storiche piuttosto tradizionali. Negli ultimi anni, attraverso tecniche di machine learning, è stato possibile identificare e sfruttare a vari scopi complesse strutture e relazioni non lineari all'interno delle serie temporali. Nel settore finanziario si è sviluppata una crescente letteratura che studia la capacità, in particolare delle reti neurali *long-short term memory*, di comprendere gli schemi sottostanti, al fine di predire futuri andamenti di mercato. Esempi interessanti di queste applicazioni si possono trovare in Bao et al. (2017) e in Fischer e Krauss (2018).

Seguendo questo indirizzo di pensiero, nel lavoro di Flori e Regoli (2021) è utilizzata tale specifica tecnica per predire gli andamenti futuri, quantificare le deviazioni e selezionare le azioni da comprare e da vendere. In dettaglio per ogni singolo titolo i si considerano i titoli ad esso cointegrati, si calcola il rendimento medio dei titoli di tale gruppo e si considera la differenza $\Delta r(t) = r_i(t) - r_i^{CG}(t)$ tra il rendimento in t del titolo e quello medio del gruppo. In seguito la rete sfrutta informazioni su rendimenti e volumi di scambio passati per calcolare la probabilità che un'azione presenti un incremento nella differenza tra l'istante $t+1$ e l'istante t, ossia $\Delta r(t+1) - \Delta r(t) > 0$. In questo

modo le azioni vengono ordinate in base al valore di probabilità ottenuto e si formano un portafoglio corto e uno lungo di ugual controvalore vendendo i titoli con probabilità più bassa (appartenenti al primo decile dei valori di probabilità) e comprando quelli con probabilità più alta (appartenenti al decimo decile). Così facendo si assume una posizione corta rispetto a quelle azioni che probabilmente avranno un rendimento minore rispetto alle loro simili e una posizione lunga rispetto a quelle azioni che probabilmente avranno un rendimento maggiore rispetto alle loro simili. Di conseguenza le azioni non vengono scambiate in coppie, anzi in linea di principio la relazione tra azioni dei due portafogli non non ha una particolare rilevanza. Una tecnica simile viene utilizzata da Chen et al. (2019). Si osservi che la strategia in esame prevede che le negoziazioni siano quotidiane e che vengano comprate e vendute delle azioni senza che si osservi per forza un divario nei prezzi o nei rendimenti, come accade invece nei lavori illustrati in precedenza.

Tornando all'articolo di Flori e Regoli (2021), la strategia da loro proposta viene messa a confronto con altre strategie di pair trading, che utilizzano metodi di ordinamento più classici, basati su differenze di prezzi o rendimenti momentanei, ed essa si dimostra paragonabile alle migliori altre da un punto di vista sia di performance, che di rischio. Il contributo interessante offerto dagli autori è che l'utilizzo della rete neurale fornisce segnali predittivi che vanno e al di là di quelli contenuti nei metodi più classici e che possono quindi essere sfruttati per generare strategie più redditizie. In particolare la probabilità che un'azione abbia rendimenti crescenti rispetto alle cointegrate permette di identificare se un possibile divario sia in diminuzione o in espansione, informazione solitamente non inclusa nei metodi tradizionali che individuano solo il gap, consentendo di generare segnali predittivi più precisi.

Occorre tuttavia osservare che anche la strategia più articolata e profittevole, tra le varianti proposte nell'articolo in questione, non permette l'ottenimento di un rendimento soddisfacente rispetto ai costi di transazione, a meno di considerare orizzonti temporali di investimento più lunghi, al fine di ridurre il ricambio di titoli nel portafoglio e a meno di considerare costi di transazione molto bassi.

L'obiettivo di questo lavoro è di approfondire la ricerca iniziata da Flori e Regoli (2021). Nella prima parte si cerca di riprodurre il loro algoritmo, valutandone le performance sia in termini di capacità predittive della rete neurale, sia in termini di rendimenti generati dalla strategia di trading. Il lavoro di tesi ha richiesto di specificare un numero consistente di "dettagli" espressi nel lavoro originario per sommi capi. Per esempio, prendendo le

stesse azioni nello stesso periodo temporale e stimando il numero di titoli cointegrati i risultati sono un poco diversi.

L'analisi giornaliera dei due autori viene arricchita su due fronti.

In una prima parte, sono studiate le caratteristiche principali dei titoli che la rete neurale classifica nel primo e nel decimo decile, ossia quelli che la strategia prevede di negoziare quotidianamente. Si cerca cioè di analizzare quali sono i motivi che guidano i profitti provenienti da tale strategia. Una volta individuati, si verifica se possono essere ricondotti all'idea di riconvergenza allo stato di equilibrio, implicita nel concetto di cointegrazione. In seguito si controlla se la strategia è robusta rispetto ad alcuni parametri, come le impostazioni dei test di cointegrazione, dalle quali dipendono fortemente i risultati dei test, il numero di dati che la rete analizza, che determina i tempi di calcolo, e infine una maggiore granularità nella formazione dei gruppi. Viene verificato quindi come le performance della strategia varino al variare di questi parametri e se esistono combinazioni più profittevoli.

Nella seconda parte, il focus dell'analisi è spostato sul trading ad alta frequenza. In particolare, la strategia è leggermente modificata in modo da valutarne le performance su dati con frequenza al minuto. Questa scelta è stata motivata dal crescente interesse e impiego del trading ad alta frequenza all'interno del mondo finanziario e dal fatto che finora sono pochi gli studi disponibili in letteratura, in quanto la maggior parte degli algoritmi sviluppati rimangono proprietà intellettuale dei creatori. Una panoramica dei principali risultati scientifici in termini di pair trading nel contesto dei dati ad alta frequenza si può trovare in Stübinger e Bredthauer (2017).

Gli obiettivi primari di questa seconda parte di analisi sono due. Il primo consiste nell'individuare i principali fattori che determinano i profitti, come gli orizzonti temporali di investimento, e anche nell'analizzare se l'inclusione di informazioni aggiuntive, scollegate da quelle riguardanti il rendimento relativo al gruppo, per esempio attraverso l'utilizzo di una strategia di pair trading più articolata, consentono di realizzare maggiori profitti. Il secondo considera invece l'influenza dei costi di transazione nel determinare l'effettiva profittabilità della strategia di trading. Per questo motivo nelle analisi svolte sono utilizzati i prezzi bid e ask osservati sul mercato. Questa informazione non solo consente di valutare in maniera più accurata il prezzo di negoziazione e il rendimento della strategia, ma anche può essere sfruttata al fine di selezionare i titoli che mostrano uno spread limitato e di conseguenza costi limitati, oppure selezionare quei titoli che probabilmente avranno un rendimento netto maggiore del costo stesso.

Capitolo 2

Analisi di serie storiche

Una serie storica (detta anche temporale) è la realizzazione di un processo stocastico discreto, e descrive la dinamica di certi fenomeni nel tempo. Può essere studiata per interpretare l'evoluzione di una variabile, individuare eventuali trend e ciclicità nei dati, oppure per provare a prevedere valori futuri della caratteristica misurata.

Questo capitolo introduce le principali tecniche econometriche per lo studio di serie temporali che verranno usate nel corso del lavoro, sottolineandone gli aspetti più importanti, e si sofferma in particolare sul concetto di cointegrazione, cruciale nella definizione della strategia di trading.

2.1 Nozioni fondamentali

Una serie storica Y viene rappresentata dall'insieme di valori che la variabile assume nel tempo. In particolare se si considera un periodo temporale caratterizzato da T istanti, la serie storica viene indicata da $Y = \{Y_1, Y_2, Y_3, ..., Y_T\}$, quindi Y_t rappresenta il valore della serie all'istante t. In questo contesto viene di solito utilizzata una terminologia speciale per indicare i valori passati di Y. In particolare, fissato un istante t, il valore di Y nel periodo precedente è chiamato *primo lag* ed è indicato da Y_{t-1}, e allo stesso modo si chiama *j-esimo lag* il valore assunto da Y j istanti prima, ovvero Y_{t-j}. Infine si definisce *differenza prima* la variazione nel valore di Y tra l'istante t e l'istante $t-1$, cioè $\Delta Y_t = Y_t - Y_{t-1}$.

Generalmente nelle serie storiche si suppone che esista un qualche tipo di dipendenza tra i valori assunti da Y in due istanti temporali vicini. Questa

dipendenza, o meglio correlazione, della serie con i suoi valori ritardati è chiamata *autocorrelazione* oppure *correlazione seriale*, e può essere definita rispetto a uno qualsiasi dei suoi lag:

$$\rho_j = \text{corr}(Y_t, Y_{t-j}) = \frac{\text{cov}(Y_t, Y_{t-j})}{\sqrt{\text{var}(Y_t)\,\text{var}(Y_{t-j})}}, \quad j = 1, 2, .., T.$$

Questa quantità esprime quindi quanto un valore Y_t della serie è influenzato dal valore della serie j istanti prima, e se questa influenza è in media positiva o negativa.

L'autocorrelazione sta alla base del concetto di *autoregressione*, cioè un modello che mette in relazione la serie storica con i suoi valori passati al fine di poter fare predizioni sui valori futuri. Il modello autoregressivo (AR) più semplice è detto del primo ordine (si indica con $AR(1)$) e consiste nell'eseguire la regressione OLS (*ordinary least squares*) dei valori Y_t della serie sui valori ritardati di un istante Y_{t-1}. E' detto autoregressivo proprio perché la regressione avviene sui suoi propri valori ritardati. In formule:

$$Y_t = \beta_0 + \beta_1 Y_{t-1} + u_t,$$

dove β_0 e β_1 sono i coefficienti veri della relazione autoregressiva e u_t è un termine di errore. Nel caso in cui si vogliano fare delle predizioni sui valori futuri della serie storica bisogna stimare i coefficienti incogniti β_0 e β_1 tramite gli stimatori OLS $\hat{\beta}_0$ e $\hat{\beta}_1$. Dopodiché il primo valore di predizione, che si riferisce quindi a una osservazione out-of-sample, si calcola come $\hat{Y}_{T+1|T} = \hat{\beta}_0 + \hat{\beta}_1 Y_T$.

Il modello $AR(1)$ usa però solo un valore ritardato e di conseguenza rischia di ignorare informazioni potenzialmente interessanti provenienti da istanti più indietro nel passato. Questo problema è facilmente risolvibile generalizzando il modello in modo tale che includa p valori ritardati. Quest'ultimo viene definito *modello autoregressivo del p-esimo ordine* e indicato come $AR(p)$. In formule:

$$Y_t = \beta_0 + \beta_1 Y_{t-1} + \beta_2 Y_{t-2} + ... + \beta_p Y_{t-p} + u_t,$$

con u_t termini di errore a cui si richiede che soddisfino $\mathbb{E}(u_t | Y_{t-1}, Y_{t-2}, ...) = 0$ per ogni t, e che non siano serialmente correlati tra loro.

Il modello autoregressivo usa quindi i dati del passato per quantificare relazioni storiche. L'idea che sta alla base è quella per cui se il futuro si

comporta allo stesso modo del passato, i dati storici possono essere utilizzati per prevedere quelli futuri. Se però il futuro differisse nettamente dal passato le relazioni storiche non sarebbero più affidabili guide per il futuro. Questo concetto è espresso dalla definizione di *stazionarietà* (cfr. Stock e Watson (2015), p.541):

Definizione 2.1.1 (stazionarietà)**.** Una serie storica è *stazionaria* se la sua distribuzione di probabilità non cambia nel tempo, cioè, se la distribuzione congiunta di $(Y_{s+1}, Y_{s+2}, ..., Y_{s+\tau})$ non dipende da s per ogni valore di τ; in caso alternativo Y_t è detta *non stazionaria*.

Per quanto detto finora, prima di iniziare ad analizzare una serie storica bisogna rispondere a due diverse domande: quanti lag devono essere inclusi nel modello? La serie storica da analizzare è stazionaria?

La risposta alla prima domanda è importante per due ragioni: se l'ordine p è troppo piccolo si rischia di omettere informazioni potenzialmente preziose, d'altra parte se è troppo alto si stimano più coefficienti del necessario introducendo quindi un maggiore errore di stima nella previsione. Esistono numerosi approcci per scegliere il parametro p, per una discussione dettagliata si veda Stock e Watson (2015). Qui vengono riportati solo i principali, che consistono nella minimizzazione dell'*information criterion*, che è una misura della qualità della stima di un modello statistico, la quale tiene conto sia della bontà di adattamento che della complessità del modello. Uno dei criteri di informazione è il *Bayes information criterion*, BIC, definito come:

$$BIC(p) = \log\left(\frac{SSR(p)}{T}\right) + (p+1)\frac{\log(T)}{T}, \tag{2.1}$$

dove $SSR(p)$ è la somma dei residui al quadrato del modello $AR(p)$ stimato. Lo stimatore di p, $\hat{p}$, è il valore che minimizza $BIC(p)$ tra tutte le possibili scelte di $p = 0, 1, .., p_{max}$. Il significato della formula è molto semplice: aumentando il numero di lag p il valore del primo termine della formula diminuisce in quanto diminuiscono i residui dei coefficienti stimati tramite OLS, allo stesso tempo il secondo termine aumenta di un fattore $\log(T)/T$. Così facendo BIC cerca un compromesso tra questi due fattori in modo tale che il numero di lag ottimale sia uno stimatore consistente del vero numero di lag.

Un altro criterio di informazione è l'*Akaike information criterion*, AIC, che è definito come:

$$AIC(p) = \log\left(\frac{SSR(p)}{T}\right) + (p+1)\frac{2}{T}.$$

L'unica differenza rispetto al BIC sta nel secondo termine, dove $\log(T)$ viene rimpiazzato da "2" in AIC, così che risulti più piccolo. In questo modo AIC è portato alla selezione di modelli che contengono più lag.

Per quanto riguarda la seconda domanda bisogna prima precisare che nelle serie storiche, soprattutto di carattere economico, possono esistere dei *trend*, ovvero dei movimenti, di lungo termine e persistenti nel tempo, della variabile rispetto al tempo. Questi possono portare la variabile a oscillare intorno al trend temporale. Si considerano due tipi di trend: deterministico e stocastico. Il primo è una funzione deterministica del tempo. Facendo per esempio riferimento alla funzione lineare, l'andamento della serie è espresso da:

$$Y_t = \beta_0 + \gamma t + \beta_1 Y_{t-1} + u_t.$$

E' utile individuare questo tipo di andamenti temporali perché possono contraddire apparentemente la stazionarietà della serie sebbene essa sia effettivamente stazionaria una volta rimosso questo andamento di lungo termine. Il secondo invece è casuale e varia senza nessuna relazione ben definita rispetto al tempo. Per questo motivo è spesso più difficili da individuare ma anche più interessanti da analizzare. Il modello più semplice di una variabile con trend stocastico è detto *random walk* ed è caratterizzato da variazioni indipendenti e identicamente distribuite (i.i.d.), cioè:

$$Y_t = Y_{t-1} + u_t \tag{2.2}$$

con u_t i.i.d. e tale che $\mathbb{E}(u_t|Y_{t-1}, Y_{t-2}, ...) = 0$. L'equazione (2.2) afferma che il valore della serie all'istante successivo è dato dal valore attuale più un termine imprevedibile, e quindi, la miglior previsione del valore futuro è data da quello attuale. Questo andamento casuale è chiaramente non stazionario in quanto la varianza della serie storica Y all'istante t dipende dal tempo secondo l'equazione $\text{var}(Y_t) = \text{var}(u_1 + u_2 + .. + u_t) = t\sigma_u^2$ e di conseguenza tutta la distribuzione del processo dipende dal tempo.

E' facile osservare che la random walk è un caso speciale del modello $AR(1)$ in cui $\beta_1 = 1$. Se invece $|\beta_1| < 1$ si può dimostrare (cfr. Stock e Watson (2015) appendice 14.2) che la distribuzione congiunta di Y_t e dei suoi lag non dipende dal tempo, rendendo il processo stazionario.

Nel caso di un modello autoregressivo di ordine p, la condizione da controllare è lievemente più complicata: per verificare la stazionarietà bisogna verificare che le radici del polinomio $1 - \beta_1 z - \beta_2 z^2 - ... - \beta_p z^p$ siano tutte maggiori di uno in modulo. Perciò in questo contesto si usa dire che una

serie storica per essere stazionaria non deve contenere radici unitarie, in caso alternativo è caratterizzata da un trend casuale che la rende non stazionaria.

2.2 Test statistici per la stazionarietà

I trend stocastici nelle serie temporali possono essere individuati con metodi informali, cioè non quantitativi, e formali. I primi consistono nello studio del grafico della serie temporale oppure nel calcolo del primo coefficiente di autocorrelazione. Infatti, quest'ultimo assume un valore molto vicino a 1 se la serie ha un trend stocastico al suo interno. In generale però, è meglio affidarsi a procedure statistiche quantitative per testare l'ipotesi di un trend casuale contro l'ipotesi alternativa per cui il trend non è presente. Il test principale usato a questo scopo è quello di Dickey-Fuller, molto frequentemente utilizzato per la sua affidabilità. Questo test è utilizzato per i modelli del tipo $AR(1)$ e testa la presenza o meno di una radice unitaria nel modello. Più in particolare testa:

$$H_0 : \beta_1 = 1 \quad \text{vs} \quad H_1 : \beta_1 < 1 \qquad \text{in} \quad Y_t = \beta_0 + \beta_1 Y_{t-1} + u_t.$$

Il test può essere riscritto sottraendo Y_{t-1} da entrambi i lati dell'equazione sulla destra, ottenendo:

$$H_0 : \delta = 0 \quad \text{vs} \quad H_1 : \delta < 0 \quad \text{in} \quad \Delta Y_t = \beta_0 + \delta Y_{t-1} + u_t,$$

dove $\delta = \beta_1 - 1$. La statistica di Dickey-Fuller è calcolata utilizzando errori standard non robusti, ovvero nell'ipotesi che siano omoschedastici.

Questo test può essere esteso anche a modelli del tipo $AR(p)$. In questo caso viene chiamato *Augmented Dickey-Fuller test*, ADF, e testa l'ipotesi per cui:

$$\begin{aligned} &H_0 : \delta = 0 \quad \text{vs} \quad H_1 : \delta < 0 \quad \text{in} \\ &\Delta Y_t = \beta_0 + \delta Y_{t-1} + \gamma_1 \Delta Y_{t-1} + \gamma_2 \Delta Y_{t-2} + .. + \gamma_p \Delta Y_{t-p} + u_t \end{aligned} \tag{2.3}$$

La statistica di ADF non ha una distribuzione normale, neanche in campioni molto grossi, i valori critici si trovano tabulati e dipendono anche dalla presenza o meno di trend deterministici nella serie. Nelle applicazioni pratiche, dove il numero di lag vero è sconosciuto, il parametro p da inserire nel modello può essere stimato attraverso *information criteria* come AIC o BIC.

In seguito al test di stazionarietà, se la serie in analisi dovesse presentare un trend stocastico, il miglior modo di gestirla è quello di trasformarla, in modo tale che questo trend venga eliminato. Questo obiettivo, nel caso della random walk per esempio, si raggiunge prendendone la differenza prima. Infatti se Y_{t-1} viene sottratto da entrambi i lati a $Y_t = \beta_0 + Y_{t-1} + u_t$, l'equazione diventa $\Delta Y_t = \beta_0 + u_t$, che è chiaramente stazionaria. In generale però bisogna prestare attenzione al fatto che un fallimento nel rifiutare l'ipotesi nulla non implica necessariamente che l'ipotesi nulla sia vera, ma piuttosto che non si hanno evidenze sufficienti per dire che è falsa.

2.3 Modello autoregressivo vettoriale

Il modello autoregressivo presentato sopra si focalizza sull'analisi di una singola serie temporale. Nella realtà però diverse serie storiche possono essere collegate tra loro e possono contenere informazioni utili per spiegare e prevedere le altre variabili. Il modello è stato quindi esteso, aggiungendo valori ritardati di più variabili, in modo tale che fosse in grado di analizzare, in maniera consistente, un vettore di serie temporali. Questo modello vettoriale è denominato *Vector AutoRegression* (VAR). Nel caso bidimensionale, cioè con due serie temporali, il modello è descritto da due equazioni: nella prima la variabile dipendente è $Y_t^{(1)}$, nella seconda la variabile dipendente è $Y_t^{(2)}$ e i regressori, in entrambe le equazioni, sono i valori ritardati di entrambe le variabili. In formule:

$$\begin{aligned} Y_t^{(1)} &= \beta_{10} + \beta_{11} Y_{t-1}^{(1)} + ... + \beta_{1p} Y_{t-p}^{(1)} + \gamma_{11} Y_{t-1}^{(2)} + ... + \gamma_{1p} Y_{t-p}^{(2)} + u_{1t} \\ Y_t^{(2)} &= \beta_{20} + \beta_{21} Y_{t-1}^{(1)} + ... + \beta_{2p} Y_{t-p}^{(1)} + \gamma_{21} Y_{t-1}^{(2)} + ... + \gamma_{2p} Y_{t-p}^{(2)} + u_{2t} \end{aligned} \tag{2.4}$$

Il modello può essere facilmente esteso a un set di k variabili diverse, e quindi k equazioni, senza particolari difficoltà. Esso diventa, in notazione matriciale:

$$\mathbf{Y}_t = \boldsymbol{\beta}_0 + \boldsymbol{\beta}_1 \mathbf{Y}_{t-1} + ... + \boldsymbol{\beta}_p \mathbf{Y}_{t-p} + \mathbf{u}_t.$$

Per $k = 2$ i coefficienti delle equazioni descritte in (2.4) vengono raccolti in questo modo:

$$\boldsymbol{\beta}_0 = \begin{pmatrix} \beta_{10} \\ \beta_{20} \end{pmatrix}, \quad \boldsymbol{\beta}_1 = \begin{pmatrix} \beta_{11} & \gamma_{11} \\ \beta_{21} & \gamma_{21} \end{pmatrix}, \quad .., \quad \boldsymbol{\beta}_p = \begin{pmatrix} \beta_{1p} & \gamma_{1p} \\ \beta_{2p} & \gamma_{2p} \end{pmatrix}, \quad \mathbf{Y}_t = \begin{pmatrix} Y_t^{(1)} \\ Y_t^{(2)} \end{pmatrix}.$$

Il significato del modello e i risultati chiave rimangono molto simili al caso unidimensionale del modello AR, tranne qualche modifica minore per riadattare quest'ultimo al caso multidimensionale. Per esempio, la formula per il BIC diventa:

$$BIC(p) = \log(\det(\hat{\mathbf{\Sigma}}_u)) + k(kp+1)\frac{\log(T)}{T},$$

dove $\mathbf{\Sigma}_u$ è la matrice delle covarianze degli errori del VAR e $\hat{\mathbf{\Sigma}}_u$ è la stima della matrice delle covarianze, cioè l'elemento (i,j) di $\hat{\mathbf{\Sigma}}_u$ è $\frac{1}{T}\sum_{t=1}^{T}\hat{u}_{it}\hat{u}_{jt}$ con $\hat{u}_{it}$ residuo OLS della i-esima equazione e $\hat{u}_{jt}$ della j-esima. La formula non è nient'altro che la generalizzazione al caso multidimensionale dell'equazione (2.1). Il primo termine esprime l'equivalente della quantità $\log(\frac{SSR(p)}{T})$, mentre il secondo rappresenta la penalizzazione per aggiungere un ulteriore regressore. Nel modello ci sono k equazioni, ognuna delle quali contiene p lag e un'intercetta, per k serie temporali. Per il resto la procedura per stimare l'ordine p rimane uguale al caso monodimensionale.

2.4 Ordine di integrazione

Fino ad adesso è stato presentato solo un tipo di trend stocastico, la random walk, che è in grado di descrivere il movimento di lungo termine di molte serie temporali. Nella realtà però esistono serie con trend più regolari, cioè che variano meno da un istante all'altro, di quanto descritto dall'equazione sopra. Servono quindi altri modelli per descrivere questo tipo di serie. Uno di questi è quello per cui la differenza prima segue una random walk, ovvero:

$$\Delta Y_t = \beta_0 + \Delta Y_{t-1} + u_t,$$

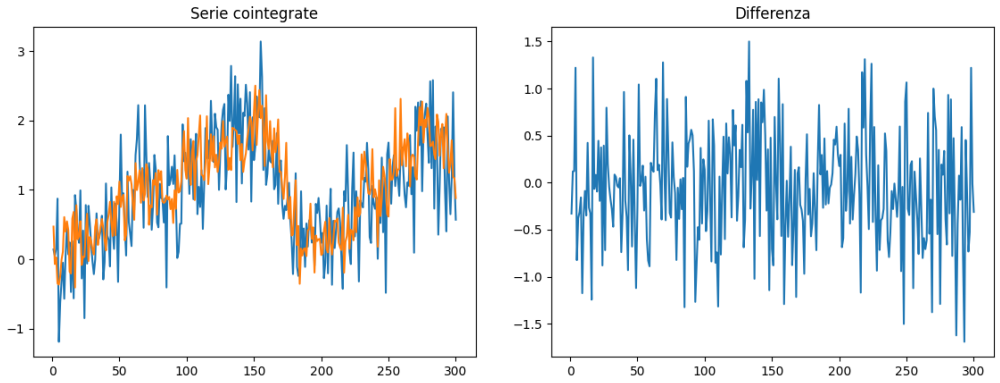
con u_t non autocorrelati. Ragionando come prima, se ΔY_t segue una random walk, allora la sua differenza, chiamata *differenza seconda* e denotata come $\Delta^2 Y_t$, è stazionaria. A questo punto è utile introdurre della terminologia addizionale per distinguere tra i diversi trend stocastici. Una serie che segue una random walk, la cui differenza quindi è stazionaria, viene definita *integrata di ordine* 1, oppure $I(1)$. Una serie la cui differenza prima segue una random walk viene definita *integrata di ordine* 2, oppure $I(2)$, e così via. Se la serie invece non presenta alcun trend stocastico, cioè è stazionaria, viene definita *integrata di ordine* 0 oppure $I(0)$. L'*ordine di integrazione* rappre-

senta quindi il numero di volte che la serie deve essere differenziata per essere stazionaria.

Da un punto di vista pratico individuare l'ordine di integrazione di una serie è molto semplice: si parte eseguendo un test di stazionarietà sulla serie, se il test rifiuta l'ipotesi nulla allora c'è evidenza che la serie sia stazionaria e di conseguenza l'ordine di integrazione è 0. In caso alternativo, cioè se viene accettata l'ipotesi nulla, si differenzia la serie e si testa l'ipotesi per cui ΔY_t ha una radice unitaria contro l'ipotesi alternativa per cui ΔY_t è stazionaria. Si procede in questo modo fino al primo rifiuto del test, dal quale si deduce facilmente l'ordine di integrazione della serie in analisi.

2.5 Cointegrazione

Tutte le nozioni e le tecniche presentate finora pongono le basi per il concetto della cointegrazione, idea chiave all'interno di questo lavoro. Può capitare nella realtà che due diverse serie temporali abbiano in comune lo stesso trend casuale e ciò le porta a muoversi insieme, cioè nella stessa direzione in un ampio intervallo temporale.



(a) Due serie storiche cointegrate

(b) Differenza di due serie cointegrate

Figura 2.1: Serie cointegrate

Dal grafico in Figura (2.1a) si vede che le due serie sono guidate dallo stesso andamento stocastico, ma la loro differenza, in Figura (2.1b), non sembra avere nessun tipo di trend. Per questo motivo si dice che hanno un

trend stocastico comune. Una definizione formale del concetto di cointegrazione è stata data dall'economista Clive Granger nella sua tesi di dottorato (Granger, 1983):

Definizione 2.5.1 (cointegrazione)**.** Date k serie storiche $\{Y^{(1)}, Y^{(2)}, ..., Y^{(k)}\}$, tutte integrate di ordine d, queste sono dette *cointegrate* se esistono k coefficienti $\theta^{(1)}, \theta^{(2)}, .., \theta^{(k)}$ tali per cui $\theta^{(1)}Y^{(1)} + \theta^{(2)}Y^{(2)} + .. + \theta^{(k)}Y^{(k)}$ è integrata di ordine minore di d. In questo caso $\theta^{(1)}, .., \theta^{(k)}$ sono chiamati *coefficienti di cointegrazione*.

La cointegrazione può essere utilizzata come metodo alternativo al calcolare le differenze di una serie per rimuovere un trend stocastico comune ad almeno due serie. In questo modo si rende la combinazione lineare stazionaria e quindi utilizzabile per l'analisi di regressione. Si supponga, per esempio, che $Y_t^{(1)}$ e $Y_t^{(2)}$ siano $I(1)$ e cointegrate con $[1, -\theta]$ vettore dei coefficienti di cointegrazione, allora $\Delta Y_t^{(1)}$, $\Delta Y_t^{(2)}$ e $Y_t^{(1)} - \theta Y_t^{(2)}$ sono $I(0)$. Il termine $Y_t^{(1)} - \theta Y_t^{(2)}$ è chiamato *error correction term* e il modello rappresentato dalle equazioni:

$$\begin{aligned}
\Delta Y_t^{(1)} &= \beta_{10} + \beta_{11}\Delta Y_{t-1}^{(1)} + ... + \beta_{1p}\Delta Y_{t-p}^{(1)} + \\
&\quad + \gamma_{11}\Delta Y_{t-1}^{(2)} + ... + \gamma_{1p}\Delta Y_{t-p}^{(2)} + \alpha_1(Y_t^{(1)} - \theta Y_t^{(2)}) + u_{1t} \\
\Delta Y_t^{(2)} &= \beta_{20} + \beta_{21}\Delta Y_{t-1}^{(1)} + ... + \beta_{2p}\Delta Y_{t-p}^{(1)} + \\
&\quad + \gamma_{21}\Delta Y_{t-1}^{(2)} + ... + \gamma_{2p}\Delta Y_{t-p}^{(2)} + \alpha_2(Y_t^{(1)} - \theta Y_t^{(2)}) + u_{2t}
\end{aligned} \tag{2.5}$$

è chiamato *Vector Error Correction Model*, VECM, e utilizza i valori passati dell'*error correction term* per predire i valori futuri di $\Delta Y_t^{(1)}$ e $\Delta Y_t^{(2)}$. Le equazioni (2.5), per semplicità di notazione, possono essere riscritte in forma matriciale come:

$$\Delta \mathbf{Y}_t = \boldsymbol{\beta}_0 + \boldsymbol{\beta}_1 \Delta \mathbf{Y}_{t-1} + .. + \boldsymbol{\beta}_p \Delta \mathbf{Y}_{t-p} + \boldsymbol{\alpha}\boldsymbol{\theta}' \mathbf{Y}_{t-1} + \mathbf{u}_t, \tag{2.6}$$

dove

$$\boldsymbol{\beta}_0 = \begin{pmatrix} \beta_{10} \\ \beta_{20} \end{pmatrix}, \quad \boldsymbol{\beta}_1 = \begin{pmatrix} \beta_{11} & \gamma_{11} \\ \beta_{21} & \gamma_{21} \end{pmatrix}, \quad .., \quad \boldsymbol{\beta}_p = \begin{pmatrix} \beta_{1p} & \gamma_{1p} \\ \beta_{2p} & \gamma_{2p} \end{pmatrix},$$

$$\boldsymbol{\alpha} = \begin{pmatrix} \alpha_1 \\ \alpha_2 \end{pmatrix}, \quad \boldsymbol{\theta}' = \begin{pmatrix} 1 & -\theta \end{pmatrix}, \quad \mathbf{Y}_t = \begin{pmatrix} Y_t^{(1)} \\ Y_t^{(2)} \end{pmatrix}.$$

Questo modello, come prima, è facilmente estendibile al caso k-dimensionale.

2.5.1 Metodi di verifica della cointegrazione

Da un punto di vista pratico ci sono essenzialmente tre metodi per determinare se due variabili possono essere modellate come cointegrate: plottare le serie e vedere se hanno un andamento stocastico comune, eseguire dei test statistici per la cointegrazione, o a volte è la conoscenza teorica delle variabili che dà un'indicazione in tal senso, come per esempio due tassi d'interesse con scadenze su orizzonti simili. Poiché gli ultimi due non sono sempre oggettivi, in seguito vengono presentati solo i due principali test statistici per la cointegrazione: il test di Engle-Granger e il test di Johansen.

Il test di Engle-Granger (Engle e Granger, 1987), la cui idea sottostante è molto semplice, si compone di due diversi passi: nel primo viene stimato il vettore di cointegrazione attraverso la seguente regressione OLS:

$$Y_t^{(1)} = \alpha + \theta Y_t^{(2)} + u_t. \tag{2.7}$$

L'equazione (2.7) costituisce un modello denominato *distributed lag model*, dove il valore corrente della variabile dipendente è basato sul valore attuale e su quelli ritardati di una variabile esplicativa. Le proprietà di questo modello sono molto simili a quelli presentati finora (per una trattazione completa si veda Stock e Watson (2015) capitolo 15).

Nel secondo passo viene eseguito un test ADF per la radice unitaria sui residui della regressione (2.7). Se i residui risultano stazionari significa che le due serie storiche hanno in comune lo stesso trend stocastico, che è stato eliminato tramite la differenza $Y_t^{(1)} - \theta Y_t^{(2)}$ e quindi sono cointegrate.

Il test di Johansen, proposto in Johansen (1991), è una procedura per testare la cointegrazione di ordine più generale rispetto a quella di Engle-Granger, in quanto si può applicare a molte variabili contemporaneamente, permettendo di rilevare più relazioni di cointegrazione. La teoria relativa al test viene di seguito presentata solo nel caso più semplice, dove non è presente alcun trend deterministico (per una trattazione più generale si veda Lütkepohl (2005) sezioni 8.2.2 - 8.2.5). Il test si basa sulla rappresentazione di un modello VECM e l'idea sottostante è quella di testare il rango r della matrice $\mathbf{\Pi}$ nell'equazione (2.8). Il rango in questo caso infatti rappresenta il numero di relazioni di cointegrazione presenti, ovvero il numero di vettori linearmente indipendenti con cui si può formare una combinazione lineare delle variabili del modello stazionaria. Più in dettaglio si consideri il seguente modello VECM senza termini deterministici:

$$\Delta \mathbf{Y}_t = \mathbf{\Pi} Y_{t-1} + \boldsymbol{\beta}_1 \Delta \mathbf{Y}_{t-1} + ... + \boldsymbol{\beta}_p \Delta \mathbf{Y}_{t-p} + \mathbf{u}_t, \tag{2.8}$$

dove $\mathbf{Y}_t$ è un processo di dimensione k e $\text{rk}(\mathbf{\Pi}) = r$ con $0 \leq r \leq k$ ignoto. Per determinare il valore di r si testa l'ipotesi:

$$H_0 : \text{rk}(\mathbf{\Pi}) = r_0 \quad vs \quad H_1 : r_0 < \text{rk}(\mathbf{\Pi}) \leq r_1.$$

In particolare l'ipotesi alternativa può essere formulata in due maniere differenti:

$$H_0 : \text{rk}(\mathbf{\Pi}) = r_0 \quad vs \quad H_1 : r_0 < \text{rk}(\mathbf{\Pi}) \leq k$$

e

$$H_0 : \text{rk}(\mathbf{\Pi}) = r_0 \quad vs \quad H_1 : \text{rk}(\mathbf{\Pi}) = r_0 + 1.$$

A seconda della formulazione, la statistica viene chiamata *trace statistic* nel primo caso e *maximum eigenvalue statistic* nel secondo, in quanto essa, sotto l'ipotesi nulla, converge in distribuzione rispettivamente alla traccia di $\boldsymbol{\mathcal{D}}$ oppure al massimo autovalore di $\boldsymbol{\mathcal{D}}$, dove

$$\boldsymbol{\mathcal{D}} = \left(\int_0^1 \mathbf{B} d\mathbf{B}' \right)' \left(\int_0^1 \mathbf{B}\mathbf{B}' ds \right)^{-1} \left(\int_0^1 \mathbf{B} d\mathbf{B}' \right) \tag{2.9}$$

$\mathbf{B}$ è un processo di Wiener standard a $k - r_0$ dimensioni, l'apice è il simbolo di trasposto, e gli integrali sono intesi come integrali di Ito. La teoria relativa agli integrali di Ito va oltre lo scopo di questo lavoro e non viene quindi affrontata. Per una trattazione esaustiva si veda Oksendal (2013).

La strategia per determinare il rango di cointegrazione di un sistema di k variabili è quella di testare una sequenza di ipotesi nulle:

$$H_0 : \text{rk}(\mathbf{\Pi}) = 0, \quad H_0 : \text{rk}(\mathbf{\Pi}) = 1, .., \quad H_0 : \text{rk}(\mathbf{\Pi}) = k - 1$$

e terminare i test quando l'ipotesi nulla non può essere rifiutata per la prima volta. Il rango della matrice è quindi dato dalla prima ipotesi non rifiutata. Per fare un esempio, si supponga di analizzare un sistema a $k = 3$ variabili: per prima cosa si testa $\text{rk}(\mathbf{\Pi}) = 0$, se l'ipotesi viene rifiutata si procede con il testare $\text{rk}(\mathbf{\Pi}) = 1$. Se questa ipotesi nulla non può essere rifiutata allora il rango sarà pari a 1.

Entrambe le statistiche, traccia e massimo autovalore, possono essere utilizzate per questi test e hanno proprietà abbastanza intercambiabili. Entrambe però richiedono che sia noto il numero di lag nel modello, che viene quindi stimato precedentemente tramite criteri d'informazione. I valori critici del test si trovano tabulati, per esempio in Johansen (1995) o in MacKinnon

et al. (1999), per i valori standard, ma possono essere facilmente simulati negli altri casi. E' infatti sufficiente costruire una random walk k-dimensionale come:

$$\mathbf{x}_t = \sum_{i=1}^{t} \mathbf{u}_i, \quad t = 1, 2, .., T \quad \mathbf{u}_i \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_k),$$

con $\mathbf{I}_k$ l'identità in $\mathbb{R}^{k \times k}$ e notare che:

$$T^{-2} \sum_{t=1}^{T} \mathbf{x}_{t-1} \mathbf{x}'_{t-1} \xrightarrow{d} \int_0^1 \mathbf{B}\mathbf{B}' ds$$

e

$$T^{-1} \sum_{t=1}^{T} \mathbf{x}_{t-1} \mathbf{u}'_t \xrightarrow{d} \int_0^1 \mathbf{B} d\mathbf{B}',$$

dove $\xrightarrow{d}$ indica la convergenza in distribuzione. Si può di conseguenza approssimare, per T grande, $\mathrm{tr}\,(\boldsymbol{\mathcal{D}})$ con $\boldsymbol{\mathcal{D}}$ definito nell'equazione (2.9) con:

$$\mathrm{tr}\left[\left(\sum_{t=1}^{T} \mathbf{x}_{t-1} \mathbf{u}'_t \right)' \left(\sum_{t=1}^{T} \mathbf{x}_{t-1} \mathbf{x}'_{t-1} \right)^{-1} \left(\sum_{t=1}^{T} \mathbf{x}_{t-1} \mathbf{u}'_t \right) \right].$$

2.6 Errori standard HAC

Nei modelli presentati sopra, in particolare *distributed lag* e VAR, spesso è presente l'ipotesi per cui i termini di errore u_t sono non autocorrelati. Nel caso in cui questa ipotesi non fosse soddisfatta è necessario fare alcune precisazioni; in linea generale infatti gli stimatori OLS dei coefficienti sono consistenti, ma gli errori standard OLS no. Ciò implica che le inferenze statistiche classiche, come test d'ipotesi e intervalli di confidenza, basate sugli errori standard, sono generalmente erronee. Per questo motivo è stato introdotto uno stimatore per la corretta formula della varianza dello stimatore OLS in caso di errori autocorrelati.

Si consideri un modello di questo tipo:

$$Y_t = \beta_0 + \beta_1 X_t + u_t,$$

dove $\mathbb{E}(u_t | X_{t-1}, X_{t-2}, ...) = 0$, X_t e Y_t sono stazionarie, (X_t, Y_t) e (X_{t-j}, Y_{t-j}) diventano indipendenti quando j diventa grande e infine outliers elevati sono

improbabili. La varianza dello stimatore di β_1 è data da:

$$\text{var}(\hat{\beta}_1) = \frac{\text{var}(\bar{v})}{(\sigma_X^2)^2}, \quad \text{con} \quad \bar{v} = \frac{\sum_{t=1}^{T} v_t}{T} \quad \text{e} \quad v_t = (X_t - \mu_x)u_t,$$

dove μ_x è la media della serie storica X e σ_X^2 è la sua varianza. Se i v_t sono i.i.d. allora $\text{var}(\bar{v}) = \text{var}(v_t)/T$ e la formula della varianza è equivalente a quella dello stimatore OLS classico. Se invece i v_t sono correlati nel tempo, la varianza di $\bar{v}$ vale:

$$\text{var}(\bar{v}) = \frac{\sigma_v^2}{T} f_T, \quad \text{con} \quad f_T = 1 + 2\sum_{j=1}^{T-1} \frac{T-j}{T}\rho_j \quad \rho_j = \text{corr}(v_t, v_{t-j}),$$

quindi

$$\text{var}(\hat{\beta}_1) = \frac{1}{T}\frac{\sigma_v^2}{(\sigma_X^2)^2} f_T, \tag{2.10}$$

cioè la varianza di $\hat{\beta}_1$ è data dalla varianza dello stimatore OLS di β_1 moltiplicata per un termine f_T che aggiusta la formula in presenza di autocorrelazione.
Il fattore f_T nella pratica va stimato attraverso lo stimatore $\hat{f}_T$, dove:

$$\hat{f}_T = 1 + 2\sum_{j=1}^{m-1} \frac{m-j}{m}\hat{\rho}_j, \quad \text{con} \quad \hat{\rho}_j = \frac{\sum_{t=j+1}^{T} \hat{v}_t \hat{v}_{t-j}}{\sum_{t=1}^{T} \hat{v}_t^2} \quad \text{e} \quad \hat{v}_t = (X_t - \bar{X})\hat{u}_t.$$

Il parametro m nell'equazione sopra è detto *parametro di troncamento* in quanto lo stimatore dell'autocorrelazione è accorciato in modo tale da includere solo $m-1$ termini di autocorrelazione al posto di $T-1$. Il valore di m va scelto con cura in quanto, da una parte includere troppe autocorrelazioni campionarie porterebbe lo stimatore ad avere una varianza troppo grande, dall'altra includerne troppo poche renderebbe lo stimatore non consistente. In genere m viene scelto seguendo la regola:

$$m = 0.75T^{1/3},$$

come suggerito in Stock e Watson (2015). Riprendendo l'equazione (2.10), si definisce lo stimatore della varianza di $\hat{\beta}_1$ che incorpora questi aggiustamenti come stimatore consistente con l'eteroschedasticità e l'autocorrelazione (HAC) e viene calcolato come:

$$\tilde{\sigma}^2_{\hat{\beta}_1} = \hat{\sigma}^2_{\hat{\beta}_1} \hat{f}_T. \tag{2.11}$$

Lo stimatore (2.11) spesso viene chiamato anche stimatore della varianza di *Newey-West* in onore degli econometristi Whitney Newey e Kenneth West che lo proposero in Newey e West (1986). Per una trattazione esaustiva si veda Stock e Watson (2015) capitolo 15.4.

2.7 Generalized Least Squares

Questa sezione offre qualche concetto teorico riguardante la regressione multipla e le modifiche necessarie quando vengono utilizzati dati storici. Per una trattazione rigorosa e dettagliata si consiglia di riferirsi a Stock e Watson (2015).

Il modello di regressione multipla può essere rappresentato in questo modo:

$$Y_i = \beta_0 + \beta_1 X_{1i} + \beta_2 X_{2i} + .. + \beta_k X_{ki} + u_i, \quad i = 1, .., n,$$

oppure in notazione matriciale come:

$$\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \mathbf{u},$$

dove

$$\mathbf{Y} = \begin{pmatrix} Y_1 \\ \vdots \\ Y_n \end{pmatrix} \quad \mathbf{u} = \begin{pmatrix} u_1 \\ \vdots \\ u_n \end{pmatrix} \quad \mathbf{X} = \begin{pmatrix} 1 & X_{11} & \dots & X_{k1} \\ \vdots & & \ddots & \\ 1 & X_{1n} & \dots & X_{kn} \end{pmatrix} \quad \boldsymbol{\beta} = \begin{pmatrix} \beta_0 \\ \vdots \\ \beta_k \end{pmatrix}.$$

Si pone inoltre $\mathbf{X}_i' = (1 \quad X_{1i} \quad ... \quad X_{ki})$. Le ipotesi del modello sono le seguenti: $\mathbb{E}(u_i|\mathbf{X}_i) = 0$ per ogni i, le coppie $(\mathbf{X}_i, Y_i)$, con $i = 1, .., n$, sono estrazioni indipendenti ed identicamente distribuite della loro funzione di probabilità congiunta, $\mathbf{X}_i$ e u_i hanno momenti quarti finiti diversi da 0, $\mathbf{X}$ ha rango pieno rispetto alle colonne, ovvero non c'è perfetta multicollinearità, $\text{var}(u_i|\mathbf{X}_i) = \sigma_u^2$, cioè gli errori sono omoschedastici condizionati a $\mathbf{X}_i$ e infine la distribuzione di u_i condizionata a $\mathbf{X}_i$ è normale. Lo stimatore del vettore incognito $\boldsymbol{\beta}$ che minimizza la somma degli errori al quadrato è quindi:

$$\hat{\boldsymbol{\beta}} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{Y}.$$

Le ipotesi descritte sopra sono adeguate nella maggior parte delle applicazioni. Nel caso di dati storici i residui presentano però un certo grado di

correlazione seriale. Di conseguenza lo stimatore può essere statisticamente inefficiente o addirittura portare a inferenze erronee. La soluzione a questo problema è quella di usare quindi errori standard che siano robusti rispetto all'eteroschedasticità e alla correlazione dei termini di errore tra le osservazioni, definiti sopra come HAC.

Ciò però non risolve il problema per cui la matrice di covarianza condizionata non sia diagonale, ovvero $\mathbb{E}(\mathbf{u}\mathbf{u}'|\mathbf{X}) = \sigma_u^2\mathbf{I}_n$, risultato che deriva dall'unione della prima, della seconda e della quinta ipotesi citata prima. Il modello *Generalized Least Squares* GLS oltrepassa queste difficoltà, sostituendo l'ipotesi di diagonalità, con quella per cui $\mathbb{E}(\mathbf{u}\mathbf{u}'|\mathbf{X}) = \boldsymbol{\Omega}(\boldsymbol{X})$, dove $\boldsymbol{\Omega}(\boldsymbol{X})$ è una matrice quadrata di dimensione n dipendente da $\mathbf{X}$: una tale descrizione della matrice delle covarianze consente delle dipendenze tra le osservazioni e il nuovo stimatore assume la forma:

$$\hat{\boldsymbol{\beta}}^{GLS} = (\mathbf{X}'\hat{\boldsymbol{\Omega}}^{-1}\mathbf{X})^{-1}(\mathbf{X}'\hat{\boldsymbol{\Omega}}^{-1}\mathbf{Y}),$$

dove $\hat{\boldsymbol{\Omega}}$ è uno stimatore di $\boldsymbol{\Omega}$.

Grazie alle sue proprietà il vettore $\hat{\boldsymbol{\beta}}^{GLS}$ può essere utilizzato per stimare i parametri di modelli lineari caratterizzati da osservazioni temporali senza che l'autocorrelazione ne indebolisca le proprietà.

In questo capitolo è stato definito il concetto della cointegrazione da un punto di vista teorico e sono stati presentati i principali metodi pratici per testare se due serie storiche sono legate da questa relazione, ponendo le basi per l'identificazione delle coppie per la strategia di pair trading. Infine sono stati presentati dei metodi per poter meglio analizzare i dati temporali che sono spesso autocorrelati e i cui errori non sono omoschedastici.

Capitolo 3

Reti Neurali

Le Reti Neurali Artificiali (ANN) dette anche Reti Neurali o Neural Networks (NN), sono tecniche di machine learning che si ispirano, in maniera semplificata, alla struttura e al funzionamento del cervello umano. Quest'ultimo infatti è costituito essenzialmente da cellule biologiche, i neuroni, interconnessi tra loro tramite fasci nervosi chiamati dentriti. In seguito a stimoli esterni si creano degli impulsi elettrici di diversa intensità che si propagano all'interno della struttura a seconda della densità di connessioni. La concentrazione e l'intensità di questi collegamenti vengono spesso modificate in seguito a impulsi esterni e proprio in questo modo avviene l'apprendimento da parte degli organismi biologici. Allo stesso modo le NN sono costituite da unità computazionali, i neuroni o nodi, connessi tra loro attraverso pesi che rappresentano l'intensità di questi collegamenti. La similitudine tra i neuroni biologici e quelli delle reti neurali si può osservare in Figura 3.1.

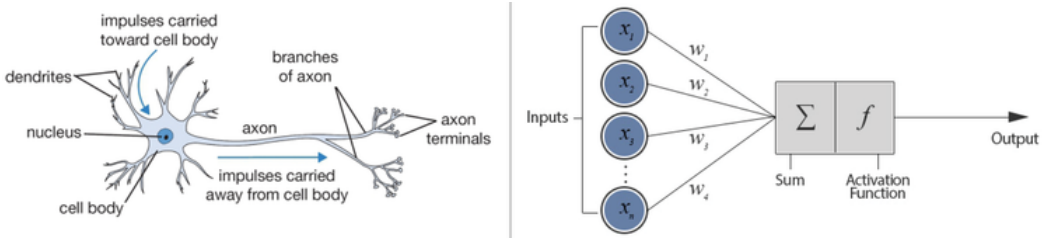


Figura 3.1: Similitudine tra la struttura di un neurone biologico (a sinistra) e uno di una ANN (a destra)

Una rete neurale artificiale calcola una funzione dei dati in ingresso e propaga, da un neurone all'altro, i valori ottenuti usando i pesi come intensità

dei collegamenti. Il processo di apprendimento della rete, proprio come gli organismi biologici variano la densità di connessioni, avviene modificando i pesi che connettono i neuroni. Questi cambiamenti avvengono in seguito a degli stimoli, che in questo caso sono costituiti da coppie di dati, input-output, rappresentanti rispettivamente le caratteristiche da imparare e la risposta associata a queste determinate caratteristiche (*supervised learning*). Da un punto di vista puramente matematico lo scopo di una rete neurale è quello di approssimare una funzione incognita f^* attraverso una mappa $y = f(x; w)$, dove x rappresenta i dati in input e w i parametri o pesi che l'algoritmo deve apprendere al fine di ottenere la migliore approssimazione possibile della funzione vera f^*.

Oggigiorno le NN e l'intelligenza artificiale hanno un ruolo sempre più centrale all'interno della nostra società e anno dopo anno sempre più sforzi ed energie vengono investiti per uno sviluppo ulteriore di queste tecnologie. Questa ricerca intensiva fa in modo che le reti neurali diventino sempre più potenti e performanti a tal punto da portare molte persone a interrogarsi sull'eticità e sui rischi che comportano.

In seguito vengono presentati gli aspetti principali di questi modelli computazionali, con un particolare focus sulle reti neurali *long-short term memory* utilizzate all'interno della strategia di trading. Le fonti utilizzate sono i due principali libri sul tema, ovvero Goodfellow et al. (2016) e Aggarwal (2018). Per una trattazione esaustiva sull'argomento si consiglia di riferirsi a queste ultime.

3.1 Composizione della rete

Una rete neurale è in grado di rappresentare una funzione complessa a piacere. La rete più semplice possibile è chiamata *perceptron neural network* ed è costituita da un singolo neurone, cioè una singola unità computazionale. Una rappresentazione sintetica si trova in Figura 3.1 a destra. I dati in input vengono direttamente mappati in uscita utilizzando un qualsiasi tipo di funzione, chiamata *funzione di attivazione*, seguendo la seguente relazione:

$$\mathbf{x} \mapsto \Phi(\mathbf{w}'\mathbf{x} + b),$$

dove $\mathbf{w}$ è il vettore dei pesi associati al nodo, b il suo *bias*, parametro che serve a traslare di una costante i dati in input, e Φ la sua funzione di attivazione.

Il neurone è l'unità computazionale di base di una rete neurale. Nella realtà esse sono infatti costituite da una molteplicità di questi neuroni che possono essere collegati tra loro in maniera molto diversa. L'intensità dei collegamenti è governata dai pesi **w** che mettono in relazione i diversi nodi presenti nella rete. Solitamente si costruiscono reti più complesse aumentando il numero di neuroni all'interno di uno strato, oppure aumentando il numero di strati stesso. I neuroni infatti vengono disposti secondo un modello stratificato per cui l'input e l'output sono separati da un gruppo di strati intermedi, chiamati strati nascosti, oppure *hidden layers*. Il numero di questi ultimi costituisce la profondità della rete. Per fare un esempio, tre funzioni $f^{(1)}, f^{(2)}, f^{(3)}$ collegate a cascata, cioè $y = f(x) = f^{(3)}(f^{(2)}(f^{(1)}(x)))$ costituiscono tre diversi strati: $f^{(1)}$ è il primo strato, $f^{(2)}$ è il secondo strato e infine $f^{(3)}$ è il terzo strato. $f^{(1)}$ e $f^{(2)}$ vengono chiamati strati nascosti in quanto non sono direttamente visibili nella relazione input-output, mentre $f^{(3)}$ viene chiamato strato output essendo quello finale. In generale ogni output di uno strato nascosto viene quindi utilizzato come input per lo strato successivo. Questo tipo di rete è chiamato *feed-forward neural network*, FNN, perché i diversi strati propagano le informazioni partendo dall'input e andando in avanti nella direzione dell'output senza formare cicli, come osservabile in Figura 3.2.

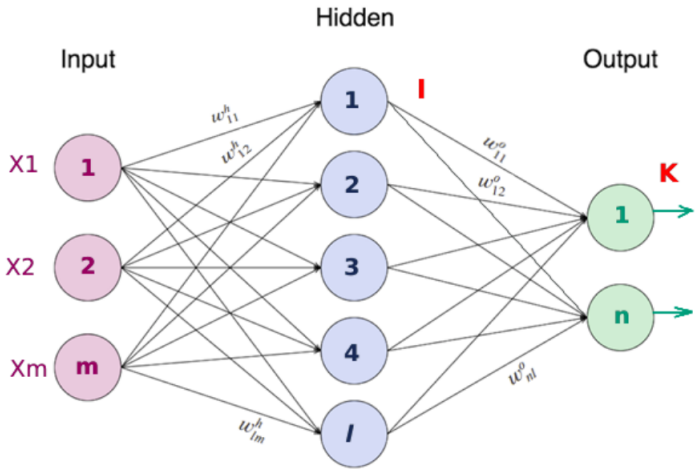


Figura 3.2: Esempio di una rete *feed-forward* con uno strato nascosto. I dati si propagano in avanti, partendo dall'input layer, passando per l'hidden layer e infine arrivando allo strato di output.

Questo tipo di reti sono state le prime messe a punto e le più semplici. Esse hanno proprietà e potenza differenti a seconda del numero di strati, del numero di nodi in ogni strato e del tipo di funzione di attivazione che viene scelto per ogni strato.

3.2 Funzione di attivazione e teorema di approssimazione universale

Solitamente tutti i nodi presenti all'interno di un medesimo strato sono caratterizzati dalla stessa funzione di attivazione, che però può cambiare in layer differenti, conferendo diverse caratteristiche alla rete. Lo scopo principale di una funzione di attivazione è quello di introdurre delle non-linearità in un algoritmo, che altrimenti sarebbe esclusivamente lineare, consentendo alla rete di apprendere schemi e relazioni complessi presenti nei dati. Infatti è stato dimostrato che una rete neurale con un sufficiente numero di neuroni e funzioni di attivazione *sigmoid* (Cybenko, 1989), oppure un sufficiente numero di strati (Hornik, 1991) può approssimare, con precisione arbitraria, una qualsiasi funzione continua tra due spazi euclidei. In maniera più generale e formale (cfr. Hornik et al. (1989)):

Teorema 1. Sia $\mathcal{NN}^{\Phi}_{d_0,d_1}$ l'insieme delle reti neurali con input di dimensione $d_0 \in \mathbb{N}$, output di dimensione $d_1 \in \mathbb{N}$ e funzione di attivazione $\Phi : \mathbb{R} \to \mathbb{R}$. Se Φ è continua e non costante, allora $\mathcal{NN}^{\Phi}_{d_0,d_1}$ è denso in $L^p(\mu)$ per ogni misura finita μ.

Questo risultato però ne garantisce solo l'esistenza, senza dare nessuna indicazione sul numero di neuroni, di strati oppure sulla struttura della rete da utilizzare nelle applicazioni reali per ottenere buone performance.

Esistono molti tipi di funzione di attivazione con caratteristiche diverse, che vengono quindi scelte a seconda dell'applicazione in esame. Per un quadro completo si veda Goodfellow et al. (2016). In questo lavoro ne verranno utilizzate 3 in particolare: *sigmoid, tanh* e *softmax*. La funzione sigmoid è così definita:

$$\Phi(x) = \frac{1}{1 + \exp(-x)}.$$

Essa ha come codominio l'intervallo $(0, 1)$ e quindi favorisce l'interpretazione per cui i valori in output siano valori di probabilità. La funzione tanh:

$$\Phi(x) = \frac{\exp(2x) - 1}{\exp(2x) + 1}$$

ha una forma simile alla sigmoid ma restituisce valori nell'intervallo $(-1, 1)$ e quindi è preferibile in quei casi in cui i risultati dei conti possono essere sia positivi che negativi. Per una rappresentazione grafica delle due si veda Figura 3.3.

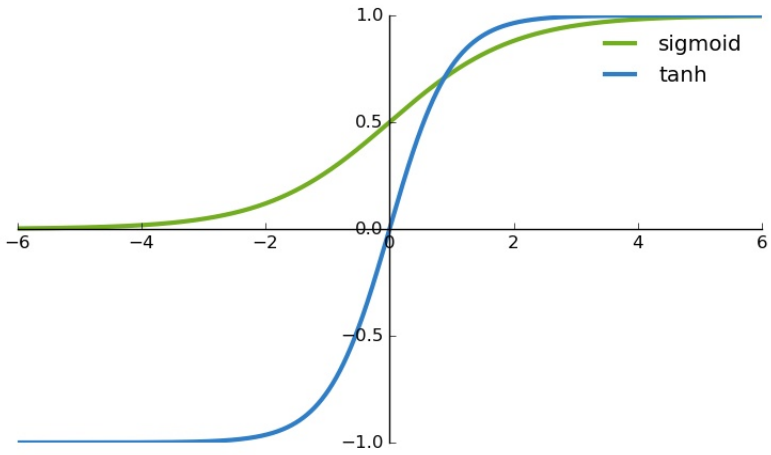


Figura 3.3: Rappresentazione delle funzioni sigmoid e tanh.

Infine la funzione softmax è così definita:

$$\Phi(x_i) = \frac{\exp x_i}{\sum_{j=1}^{k} \exp x_j} \quad \forall i \in \{1, 2, ..., k\},$$

dove x_i rappresenta il valore in output dell'i-esimo nodo nell'ultimo strato nascosto e k è il numero totale di nodi nell'ultimo strato. La funzione softmax viene solitamente utilizzata nello strato di output delle reti che mirano a risolvere un problema di classificazione categorico, dove cioè a ogni dato (oppure insieme di dati) in ingresso viene associata la probabilità di appartenenza a una delle k classi del problema. E' stata utilizzata all'interno di questo lavoro perché, come si vedrà in seguito, il trading viene eseguito in seguito al risultato di un problema di classificazione categorico.

3.3 Loss function

La *loss function*, in ottimizzazione matematica e statistica, è una funzione che ha il ruolo di misurare la distanza tra un valore stimato e un valore vero. Nel contesto delle reti neurali serve quindi a misurare la bontà dell'approssimazione della funzione f^* tramite la funzione f generata dalla rete. La scelta di questo tipo di funzione è quindi critica e altamente dipendente dal tipo di applicazione da svolgere. Per una trattazione generale e approfondita si faccia rifermento alle fonti.

In questo lavoro la loss function che viene utilizzata è chiamata *cross-entropy loss* ed è così definita:

$$L = -\frac{1}{N}\sum_{j=1}^{N} \mathbf{y}'_j \log(\hat{\mathbf{y}}_j) \tag{3.1}$$

dove $\mathbf{y}_j$ è un vettore codificato *one-hot*, cioè con tutti gli elementi nulli tranne quello nella posizione rappresentante la classe vera di appartenenza, posto uguale a 1. Il vettore $\hat{\mathbf{y}}_j$ viene restituito dallo strato output con attivazione softmax e N è il numero di elementi che sono stati classificati. Quindi la probabilità predetta di appartenenza a una determinata classe viene paragonata alla classe effettiva e viene calcolata una penalizzazione proporzionale a quanto la probabilità predetta è distante dal valore atteso. La penalizzazione introdotta è logaritmica e quindi produce valori elevati quando il valore vero è 1 ma viene predetto un numero vicino a 0 e valori piccoli quando il valore vero è 1 e viene predetto un numero molto vicino ad 1. Questa loss function è quindi in grado di misurare quanto siano distinguibili tra loro due distribuzioni di probabilità discrete.

3.4 Processo di apprendimento

L'utilizzo delle reti neurali è reso possibile da quello che viene definito processo di apprendimento (o *learning*). Durante questa fase la rete neurale impara a riconoscere quali sono le caratteristiche dei dati in ingresso x che ne determinano poi la risposta y. Per fare ciò assegna dei valori numerici, con l'obiettivo di minimizzare la loss function, a tutti i parametri $\mathbf{w}$ presenti nella rete e che servono a collegare tra loro i vari neuroni. Questo procedimento viene eseguito innanzitutto dividendo l'insieme di dati provenienti

dalla funzione incognita $y = f^*(\mathbf{x})$ in due diversi sottoinsiemi: il *training set* e il *test set*. Il primo viene usato per la calibrazione dei parametri e il secondo viene utilizzato per valutare la bontà dell'approssimazione raggiunta dalla rete neurale una volta istruita. Così facendo si possono valutare le performance di predizione della rete su un insieme di dati che non è stato utilizzato per il processo di learning. Al fine di calibrare i parametri bisogna risolvere il problema di ottimizzazione dato da:

$$\hat{\mathbf{w}} = \underset{w \in W}{\operatorname{argmin}} L(\{f(\mathbf{w}, \mathbf{x}_i)\}_{i=1}^{M}, \{f^*(\mathbf{x}_i)\}_{i=1}^{M}\}),$$

dove L è la loss function scelta, W rappresenta l'intero spazio dei parametri e M è il numero di dati presenti nel training set. Molto spesso però il numero elevato di strati nella rete, di nodi in ogni singolo strato e le funzioni di attivazioni non lineari rendono la loss function una composizione complicata di funzioni non lineari e questa struttura da luogo a un problema di ottimizzazione a molte dimensioni e altamente non lineare che è difficilmente risolvibile con il metodo standard di discesa del gradiente. Per calcolare il gradiente di questa composizione di funzioni è stato quindi sviluppato un algoritmo denominato *backpropagation algorithm* che sfrutta la regola della catena (chain rule) del calcolo differenziale. Esso calcola il gradiente come una somma, su tutti i possibili cammini da un neurone all'output, di prodotti di gradienti locali. Questa somma ha un numero esponenziale di cammini rispetto al numero di strati ma può essere facilmente calcolata attraverso una tecnica chiamata *dynamic programming*.

3.4.1 Backpropagation algorithm

Il *backpropagation algorithm*, o algoritmo di retropropagazione, consiste nell'utilizzo del dynamic programming e si compone di due diverse fasi: una *in avanti* e una *all'indietro*. Durante la prima fase i dati del training set vengono propagati, in gruppi, all'interno della rete dall'input all'output. In questo modo per ogni gruppo vengono calcolati i valori di attivazione di ciascuno strato nascosto utilizzando come pesi quelli correnti. Così facendo si ottengono dei valori in output; essi possono essere confrontati, attraverso la loss function, con i valori veri, e si possono quindi calcolare le derivate della funzione di loss rispetto ai valori in output. Dopodiché bisogna calcolare le derivate della loss rispetto a tutti i pesi presenti nella rete e ciò viene fatto nella seconda fase. L'obiettivo della seconda fase è infatti quello di calcolare

questi gradienti utilizzando la *chain rule*. Si parte calcolando il gradiente della loss rispetto all'output dell'ultimo strato, poi il gradiente rispetto ai dati in ingresso nell'ultimo strato e infine il gradiente rispetto ai pesi che collegano due diversi strati, e così via fino ad arrivare al primo strato della rete. Una volta ottenuto il gradiente rispetto a tutti i possibili pesi presenti si possono aggiornare gli stessi seguendo la normale procedura del metodo della discesa del gradiente. Una rappresentazione schematica si può osservare in Algoritmo 1.

Algoritmo 1 Backpropagation

```
for d in data do
    Forward phase: partendo dall'input layer propaga i dati attraverso la rete
    Backward phase: calcola le derivate della loss rispetto all'output layer
    for strato in stratiAlContrario: do
        calcola le derivate della loss rispetto agli input dei neuroni dello strato in esame
        calcola le derivate della loss rispetto ai pesi colleganti lo strato in esame e il precedente
        calcola le derivate della loss rispetto alle funzioni di attivazione al layer precedente
    end for
    aggiorna i pesi seguendo il metodo di discesa stocastica
end for
```

L'algoritmo di backpropagation è lo strumento che le reti neurali usano per apprendere ed è quindi lo step principale per il training. Qui ne è stata fatta una descrizione sintetica e non quantitativa, il lettore interessato ai dettagli e al rigore matematico è invitato a fare riferimento al capitolo 3.2 di Aggarwal (2018).

E' utile aggiungere che per quanto detto finora l'apprendimento viene eseguito utilizzando tutti i dati del training set una sola volta, cioè ogni dato viene propagato nella rete e usato per la stima del gradiente in un solo momento. Spesso però sono necessarie più iterazioni di questo algoritmo per fare in modo che la rete apprenda tutte le caratteristiche importanti che deve ricercare nei dati in ingresso. Questo processo può essere quindi ripetuto più volte, fino al raggiungimento di buone performance di predizione. In questo ambito ognuno di questi passaggi è denominato *epoch*.

3.4.2 Problemi pratici nel processo di apprendimento

Nonostante l'ottima reputazione delle reti neurali, rimangono moltissimi problemi a cui bisogna prestare attenzione, soprattutto durante la fase di apprendimento. Uno dei principali è quello dell'*underfitting* e dell'*overfitting* dei dati di training. Il primo significa che il modello non è sufficientemente complesso per imparare a riconoscere tutte le caratteristiche rilevanti nei dati. Per venire a capo di questa eventualità, di solito, è sufficiente aumentare la profondità della rete oppure il numero di nodi. Il secondo significa invece che la rete neurale è eccessivamente complessa e ha quindi ottime performance di predizione sui dati utilizzati per l'apprendimento, ma pessime su dati mai visti (test set). Ciò accade poiché, durante il processo di apprendimento, la rete impara delle caratteristiche casuali, specifiche dei dati utilizzati in questa fase, che non si generalizzano a nuovi dati. Un esempio di overfitting, underfitting e modello di corretta complessità è osservabile in Figura 3.4.

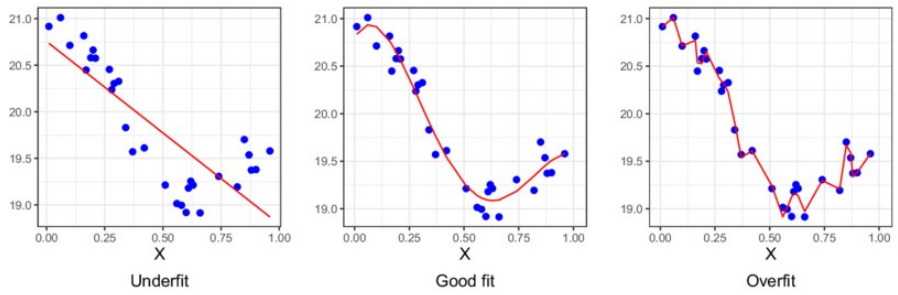


Figura 3.4: Esempio di underfitting e overfitting di polinomi. Underfitting (sinistra) è caratterizzato da un modello troppo semplice. Overfitting (destra) è caratterizzato da un modello troppo complesso. Il fit corretto è rappresentato in centro.

Per risolvere il problema dell'overfitting, nell'ambito delle NN, sono state sviluppate diverse tecniche, chiamate di regolarizzazione. Le due tecniche principali sono quelle *penalty-based* e *early-stopping*. La prima consiste nel correggere la funzione di loss aggiungendo delle penalità o altri tipi di vincoli al fine di favorire modelli semplici. Questa non verrà utilizzata all'interno del presente elaborato, motivo per il quale non sarà oggetto di ulteriore analisi. La seconda invece consiste nel terminare anticipatamente il processo di learning senza aver raggiunto la soluzione ottima sui dati di training. Più in dettaglio, si prende una piccola percentuale dei dati di training, chiamata *validation set*, che non viene usata per l'addestramento del modello, bensì per valutare l'errore tra il valore vero e quello predetto dalla rete, attraverso la

funzione di loss. In questo modo, alla fine di ogni epoch, si possono analizzare le performance della rete sia sui dati di training sia su quelli di validation. Così si determina il momento di fine dell'apprendimento come l'epoch in cui l'errore sulla porzione di dati esclusa (validation set) comincia a crescere, segnale che anticipa l'overfitting. Questo fenomeno è facilmente osservabile in Figura 3.5. Questa tecnica da una parte presenta degli svantaggi, in quanto

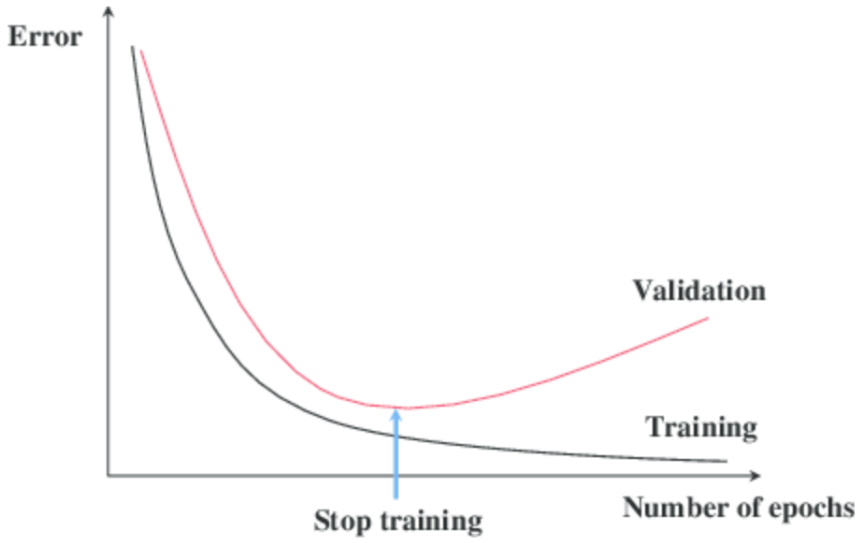


Figura 3.5: Esempio di early-stopping: la freccia in basso indica il momento in cui l'errore di validazione comincia a crescere ed è quindi ottimale fermare il processo di training.

riduce il numero di dati utilizzabili per il processo di apprendimento, dall'altra sembra funzionare bene in termini di predizioni out-of-sample, perché il momento di stop è determinato su una parte di dati non utilizzata per il training.

Come affermato in precedenza, aumentare la profondità della rete da una parte permette di accrescerne la potenza e le capacità predittive, ma dall'altra porta a un problema di stabilità degli aggiornamenti dei pesi nell'algoritmo di retropropagazione. Se infatti il numero di strati è molto elevato, gli aggiornamenti dei pesi nei primi strati possono essere o molto piccoli, *vanishing gradient*, oppure molto grandi, *exploding gradient*. Ciò si verifica a causa dei risultati delle produttorie derivanti dalla regola della catena, che possono aumentare o diminuire esponenzialmente lungo i cammini. Si consideri, per esempio, una rete composta da molti strati costituiti da un nodo ciascuno.

La derivata rispetto ai pesi nei primi strati è calcolata come prodotto di tutte le derivate rispetto ai pesi degli strati successivi. Di conseguenza, se il valore atteso di queste derivate locali è maggiore o minore di uno, il prodotto delle derivate aumenta o diminuisce in maniera esponenziale, causando degli aggiornamenti nei pesi eccessivamente grandi oppure eccessivamente piccoli.

In questo lavoro vengono usate due diverse funzioni di attivazione (sigmoid e tanh) negli strati nascosti e tutte e due favoriscono il problema del vanishing gradient: entrambe hanno valori della derivata molto piccoli nelle regioni di saturazione e come valori massimi 0.25 e 1 rispettivamente. Esistono numerose soluzioni al problema, per una trattazione completa si veda il capitolo 3.4 in Aggarwal (2018).

3.5 Algoritmi di training

Come già affermato l'algoritmo di apprendimento delle reti neurali si basa sulla minimizzazione della funzione di loss $L(\mathbf{w}, \mathbf{x}_i)$ e, per raggiungere questo obiettivo, viene utilizzata la tecnica di discesa del gradiente che aggiorna ricorsivamente i parametri $\mathbf{w}$ secondo l'equazione:

$$\mathbf{w}_{t+1} = \mathbf{w}_t - \alpha \nabla \boldsymbol{L}_t,$$

dove α rappresenta il passo di aggiornamento, detto *learning rate*, e, per la valutazione della loss, vengono utilizzati tutti i dati presenti nel training set. Questa procedura fa in modo che i pesi vengano aggiornati nella direzione opposta al gradiente, ovvero la direzione di massima decrescita della funzione loss rispetto allo spazio dei parametri. Questa direzione è però ottimale solo per passi infinitesimali, nella realtà può quindi portare a direzioni sbagliate oppure a un continuo zig-zag. Per provare a risolvere questo tipo di problema viene solitamente proposto un passo che diminuisce iterazione dopo iterazione, man mano che l'algoritmo si avvicina al minimo della funzione.

La versione classica del gradient descent, descritta sopra, nella realtà non viene quasi mai utilizzata per due diversi motivi: il primo dipende dal fatto che calcolare i gradienti per tutto il training set è spesso impossibile da un punto di vista di memoria della macchina. Il secondo invece è relativo alla problematica per la quale l'algoritmo della massima decrescita non converge necessariamente a un minimo globale, ma converge a un punto stazionario che corrisponde in genere a un minimo locale non ottimale. Esistono varie soluzioni per ovviare a questi problemi. Una di queste è detta *stochastic*

gradient descent, e consiste nel valutare la loss, calcolare il gradiente e aggiornare i pesi, dato per dato e non su tutto il training set. In questo modo viene ridotto drasticamente il consumo di memoria e il tempo necessario alla convergenza dell'algoritmo in un punto di minimo, ma allo stesso tempo la direzione di aggiornamento dei pesi è molto variabile a causa della forte approssimazione nel calcolo del gradiente.

Un'altra alternativa è quella di utilizzare la tecnica del *mini-batch gradient descent* che consiste nel dividere il training set in diversi gruppi, chiamati *mini-batch* e approssimare il gradiente vero, cioè calcolato su tutto il training set, con uno calcolato sul singolo batch. In formule:

$$\nabla L = \frac{1}{N}\sum_{i=1}^{N} \nabla l_i \approx \frac{1}{m}\sum_{j\in B} \nabla l_j \qquad B = \{j_1, j_2, .., j_m\}$$

dove $\{j_1, j_2, .., j_m\}$ sono gli indici appartenenti al batch in esame B e l_j indica la loss valutata solo sul j-esimo dato. Quest'ultima soluzione si rivela spesso essere il miglior compromesso tra stabilità, velocità dell'algoritmo e utilizzo di memoria.

Diversi aggiustamenti di questi algoritmi sono stati sviluppati nel corso del tempo al fine di evitare i numerosi minimi locali e accelerare il processo di apprendimento che può essere molto lungo. Essi si dividono principalmente in due categorie: da una parte ci sono le tecniche basate sul *momentum*, che aumentano il learning rate in quelle direzioni che puntano verso la soluzione ottimale, cioè quelle direzioni che rimangono coerenti tra un aggiornamento e un altro. Dall'altra le tecniche basate su *parameter-specific learning rate*, che assegnano diversi passi di aggiornamento ai diversi parametri a seconda della grandezza delle derivate parziali. L'idea è infatti quella per cui le direzioni dei parametri con elevate derivate parziali tendono ad oscillare mentre quelle con derivate parziali piccole tendono a rimanere consistenti. E' quindi utile diminuire i learning rate per le prime e aumentarli per le seconde.

Il metodo utilizzato nel presente lavoro è RMSprop (Tieleman e Hinton, 2012) in cui, in ogni iterazione dell'algoritmo del gradiente, vengono eseguiti i seguenti aggiornamenti:

$$A_i \Longleftarrow \rho A_i + (1-\rho)\left(\frac{\partial L}{\partial w_i}\right)^2$$

$$w_i \Longleftarrow w_i - \frac{\alpha}{\sqrt{A_i}}\frac{\partial L}{\partial w_i}.$$

La derivata parziale $\frac{\partial L}{\partial w_i}$ viene scalata per un fattore inversamente proporzionale alla radice quadrata della grandezza della derivata, senza considerarne il segno. In questo modo viene incoraggiato un movimento in quelle direzioni con pendenza più moderata e con direzione consistente e allo stesso tempo viene limitato quello nelle direzioni più inclinate, in cui c'è più rischio di oscillazioni. Il termine $\rho \in (0, 1)$ serve invece a fare in modo che i progressi non siano rallentati prematuramente da un fattore A_i costantemente crescente per cui l'importanza delle derivate passate decrescerebbe esponenzialmente. Infine i è un indice che varia su tutti i pesi addestrabili presenti nella rete neurale.

3.6 Batch-normalization

Avendo definito il concetto di *batch*, è interessante presentare un altro metodo molto utilizzato nel campo del deep learning: *batch normalization*. Batch normalization, presentato per la prima volta in Ioffe e Szegedy (2015), aiuta a ridurre il problema del vanishing/exploding gradient e riduce il fenomeno denominato *internal covariate shift*. Questo termine serve a descrivere il fenomeno per cui, durante il processo di apprendimento, i pesi continuano a cambiare e quindi cambiano anche i valori di attivazione. In altre parole, gli input dai primi strati agli ultimi continuano a modificarsi. La modifica degli input dagli strati iniziali a quelli successivi causa una convergenza più lenta poiché i dati di training per gli strati successivi non sono stabili. Batch normalization serve a introdurre "strati di normalizzazione", tra i diversi strati nascosti, col compito di modificare i dati in modo che abbiano una varianza confrontabile. In questo modo si riduce l'instabilità sopra descritta. Più in particolare, l'output di ogni nodo, proveniente dalla propagazione di un singolo mini-batch, viene modificato in modo tale che abbia media β_i e deviazione standard γ_i. Per raggiungere questo obiettivo su un particolare batch di numerosità m, i cui valori di attivazione valgono $v_i^{(1)}, v_i^{(2)}, .., v_i^{(m)}$, uno strato batch-normalization innanzitutto calcola la media μ_i e la varianza σ_i dei valori per il nodo i-esimo:

$$\mu_i = \frac{\sum_{r=1}^{m} v_i^{(r)}}{m}$$

$$\sigma_i^2 = \frac{\sum_{r=1}^{m} \left(v_i^{(r)} - \mu_i \right)^2}{m} + \epsilon,$$

dove il parametro ϵ serve a regolarizzare i casi in cui tutti i valori di attivazione sono uguali, il che comporterebbe una varianza nulla. Poi scala i valori di attivazione attraverso le formule:

$$\hat{v}_i^{(r)} = \frac{v_i^{(r)} - \mu_i}{\sigma_i} \quad \forall r$$

$$a_i^{(r)} = \gamma_i \hat{v}_i^{(r)} + \beta_i \quad \forall r.$$

Da un punto di vista pratico bisogna prestare attenzione durante la fase di predizione: i parametri μ_i e σ_i vengono calcolati su tutti i dati di training e considerati costanti nella fase di previsione, in questo modo non c'è dipendenza dal singolo batch. Per una trattazione dettaglia sulle modifiche necessarie da applicare al algoritmo di retropropagazione si veda Aggarwal (2018) capitolo 3.6.

3.7 Iperparametri

Per quanto detto finora le reti neurali, attraverso il processo di addestramento, sono in grado di assegnare i valori che minimizzano la loss function ai pesi associati alla densità di connessioni presenti nella rete. Esistono però una serie di parametri, detti *iperparametri* che la rete non è in grado di scegliere autonomamente e che devono essere impostati dall'esterno. Il termine *iperparametro* si riferisce in particolare a tutti quei parametri che regolano la struttura del modello, come il learning rate e il numero di nodi in uno strato, e che sono quindi fondamentalmente diversi dai pesi presenti nei vari strati della rete. Si può quindi affermare che c'è un'organizzazione dei parametri su due livelli diversi nelle reti neurali: nel primo si scelgono i valori degli iperparametri, e solo dopo i pesi vengono ottimizzati attraverso l'algoritmo di retropropagazione.

La scelta degli iperparametri può avvenire in due modalità differenti: possono essere impostati manualmente oppure si utilizzano delle procedure di *tuning*, che testano diverse combinazioni degli iperparametri in maniera automatizzata. I due algoritmi di tuning più utilizzati sono il *grid search* e *random search*. Grid search prevede che un insieme di valori venga selezionato per ogni iperparametro e vengano testate tutte le combinazioni possibili scegliendo poi quella ottimale, ovvero quella attraverso la quale la rete mostra le performance migliori. Il problema di questa tecnica è che il numero

di combinazioni può essere molto elevato e quindi richiedere un costo computazionale eccessivo. Random search invece testa casualmente solo alcune delle combinazioni possibili, riducendo quindi i tempi di calcolo.

Il tuning degli iperparametri è una fase cruciale per l'utilizzo delle reti neurali in quanto determina la precisione della rete nel portare a termine il suo compito. Per questo motivo le due tecniche appena presentate possono essere migliorate in diversi modi con l'obiettivo di ridurre il costo computazionale e aumentarne l'efficienza. Per una discussione dettagliata si faccia riferimento a Goodfellow et al. (2016) e Aggarwal (2018).

3.8 Rete Neurale Ricorrente

Le reti neurali feed-forward, descritte sopra, si sono dimostrate capaci di raggiungere obiettivi eccezionali in moltissimi casi pratici. Esse sono designate per dati multidimensionali le cui caratteristiche sono indipendenti l'una dall'altra, cioè trattano ogni osservazione come scorrelata dalle altre. Esistono però insiemi di dati, come le serie temporali e i testi, che presentano dipendenze sequenziali tra loro, e che quindi non si prestano ad essere analizzati da queste reti in quanto non tengono conto del "contesto". La necessità di modellare questo tipo di dati ha portato all'introduzione di un diverso modello di NN, chiamate *Recurrent Neural Networks* RNN, che mirano alla codificazione di relazioni sequenziali e temporali. Il singolo dato $\underline{x}$, che una RNN analizza, è rappresentabile come una serie storica: è infatti costituito da una successione di osservazioni $\{\underline{x}_t\}$. Ognuna di queste osservazioni è una parte essenziale dell'informazione, ma se analizzata separatamente dalle altre si trascurano tutte quelle informazioni provenienti dal contesto. La particolarità delle reti neurali ricorrenti è quella di avere connessioni retroattive grazie alle quali riescono a tenere conto del contesto. Nelle prossime pagine vengono presentate le caratteristiche principali di questo tipo di reti, per una discussione più dettagliata si faccia riferimento al capitolo 7 di Aggarwal (2018) oppure al capitolo 10 di Goodfellow et al. (2016).

3.8.1 Architettura

La RNN più semplice, introdotta in Elman (1990) e rappresentata in Figura 3.6, è costituita da uno strato nascosto e uno strato output. La particolarità di questo tipo di rete consiste nel *self-loop* dello strato nascosto h che consente

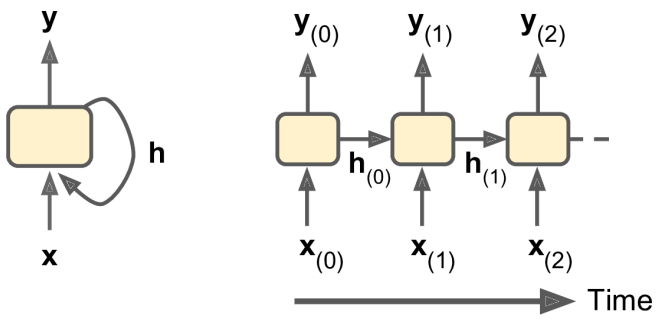


Figura 3.6: Esempio di Recurrent Neural Network: a sinistra una rappresentazione compressa, a destra una rappresentazione estesa temporalmente

a quest'ultimo di sfruttare la conoscenza dello stato all'istante precedente. Una RNN può essere rappresentata in maniera estesa attraverso una rete a strati temporali, che ricorda una normale FNN, dove ogni layer è costituito da un instante temporale differente. Dalla Figura 3.6, a destra, risulta chiaro come lo stato all'istante temporale $t-1$ influenzi lo stato all'istante t. Lo stato dello strato nascosto è quindi descritto dalla relazione:

$$h_t = \Phi(x_t, h_{t-1}; w).$$

E' importante notare che in ogni istante temporale la rete riceve un dato diverso x_t, relativo a quel determinato istante, lo stato dello strato interno è influenzato dal suo valore all'istante precedente e dal dato in input e di conseguenza anche il risultato in output dipende dal tempo. Allo stesso tempo però le matrici dei pesi e la funzione di attivazione sono le stesse in ogni istante temporale, ovvero non dipendono dal tempo.

L'output è definito come funzione dello stato h_t, cioè $y_t = g(h_t)$ ma, grazie alla natura ricorsiva dello strato nascosto, può essere riscritto come funzione dei soli dati in ingresso come:

$$y_t = F_t(x_1, x_2, .., x_t).$$

In questo modo la funzione $F_t(\cdot)$ varia con il valore di t sebbene la sua relazione con lo stato immediatamente precedente sia sempre la stessa.

Più in dettaglio, una *vanilla* RNN con uno strato nascosto e un output layer è descritta dalle seguenti equazioni, riportate qui in forma vettoriale:

$$\begin{aligned} \mathbf{h}_t &= \Phi(\mathbf{W}\mathbf{h}_{t-1} + \mathbf{U}\mathbf{x}_t + \mathbf{b}^h), \qquad t = 1, ..., T \\ \mathbf{y}_t &= \Phi^{output}(\mathbf{V}\mathbf{h}_t + \mathbf{b}^y). \end{aligned}$$

Si supponga che p sia la dimensione dello stato interno e d_0 la dimensione dei dati in ingresso e d_1 in uscita, allora $\mathbf{U}$ è una matrice di pesi (p, d_0), $\mathbf{W}$ è una matrice (p, p), $\mathbf{V}$ è una matrice (d_1, p) e infine $\mathbf{b}^h$ e $\mathbf{b}^y$ sono i due vettori di bias rispettivamente a p e d dimensioni per le due funzioni di attivazione Φ e Φ^{output}.

3.8.2 Backpropagation through time BPTT

Il processo di training di una RNN si basa sempre su algoritmi del gradiente ma è lievemente più complesso rispetto a quello di una FNN in quanto la connessione retroattiva modifica la topologia della rete. Se si osserva la rappresentazione di destra in Figura 3.6, si intuisce subito che la retropropagazione deve essere eseguita sia verticalmente, cioè da y_t a x_t, sia orizzontalmente, cioè andando all'indietro nel tempo. Un altro problema è dato dal fatto che, per usare la regola della catena nell'algoritmo di backpropagation, i pesi in strati differenti devono essere diversi tra loro. Ciò però non accade in questo tipo di reti: la matrice $\mathbf{W}$ è uguale per ogni strato temporale. L'algoritmo deve quindi essere modificato per poter gestire le due diverse direzioni di retropropagazione e i pesi condivisi dai diversi strati. La soluzione consiste nel fingere che i pesi nei diversi strati temporali siano indipendenti l'uno dall'altro, definendo cioè $\mathbf{W}^{(t)}$, $\mathbf{U}^{(t)}$ e $\mathbf{V}^{(t)}$ come dipendenti dal tempo. Si segue quindi il normale algoritmo di retropropagazione nelle due diverse direzioni, calcolando un aggiornamento per ogni peso di ogni strato temporale che in seguito vengono sommati per ottenere un solo aggiornamento per ogni parametro. Una rappresentazione schematica è fornita in Algoritmo 2.

Ci sono diversi problemi pratici nel processo di training delle RNN. Uno di questi è relativo alla lunghezza delle sequenze in input. Se le serie temporali sono costituite da un elevato numero di istanti la rete risultante avrà un numero elevato di strati temporali che può comportare delle difficoltà nella convergenza dell'algoritmo e a un eccessivo consumo di memoria computazionale. Questi problemi possono essere risolti attraverso la tecnica chiamata *truncated back-propagation through time* (TPBTT). Questa prevede che gli stati siano calcolati normalmente nella fase di propagazione in avanti ma gli aggiornamenti all'indietro siano calcolati sfruttando segmenti temporali di lunghezza limitata, per esempio 100. Solo una parte della loss function viene quindi sfruttata per calcolare le derivate parziali e per aggiornare i pesi.

Un'eccessiva profondità della rete, dipendente dalla lunghezza delle sequenze in input, può causare anche il problema del *vanishing/exploding gra-*

Algoritmo 2 Backpropagation through time

1: **for** d in data **do**
2: Forward phase: per ogni istante propaga i dati attraverso la rete e calcola la loss
3: Backward phase: calcola le derivate all'indietro rispetto a ogni peso di ogni strato temporale considerandoli indipendenti tra loro cioè calcolando: $\frac{\partial L}{\partial \mathbf{W}^{(t)}}$, $\frac{\partial L}{\partial \mathbf{U}^{(t)}}$ e $\frac{\partial L}{\partial \mathbf{V}^{(t)}}$
4: **end for**
5: Somma i pesi condivisi corrispondenti a diversi istanti temporali, ovvero:

$$\frac{\partial L}{\partial \mathbf{W}} = \sum_{t=1}^{T} \frac{\partial L}{\partial \mathbf{W}^{(t)}}, \qquad \frac{\partial L}{\partial \mathbf{U}} = \sum_{t=1}^{T} \frac{\partial L}{\partial \mathbf{U}^{(t)}} \qquad \frac{\partial L}{\partial \mathbf{V}} = \sum_{t=1}^{T} \frac{\partial L}{\partial \mathbf{V}^{(t)}}$$

6: aggiorna i pesi seguendo il metodo di discesa stocastica

dient accennato in precedenza. Se si considera una rete ricorrente, a uno strato nascosto, con funzione di attivazione Φ, con un peso condiviso w tra una coppia di nodi, e T istanti temporali, la derivata della loss function rispetto al valore di attivazione dello stato h_t nella fase di retropropagazione sarà:

$$\frac{\partial L}{\partial h_t} = \Phi'(h_{t+1}) w_{t+1} \frac{\partial L}{\partial h_{t+1}}.$$

Poiché i pesi nei diversi strati temporali sono condivisi il gradiente è moltiplicato sempre per la stessa quantità $w_t = w$. Nel caso $w > 1$ ci sarà un'inclinazione verso il problema dell'*exploding gradient*, mentre nel caso $w < 1$ l'inclinazione sarà nella direzione del *vanishing gradient*. Come affermato in precedenza, anche la funzione di attivazione gioca il suo ruolo in questo contesto, in quanto molte di quelle che vengono utilizzate nella pratica hanno un valore massimo della derivata pari a 1 se non addirittura più piccolo, favorendo quindi il vanishing gradient.

Numerose tecniche sono state proposte per provare a risolvere questi problemi ma vanno oltre lo scopo di questo elaborato, per una discussione dettagliata si veda Aggarwal (2018) capitolo 7.3. oppure Goodfellow et al. (2016) capitolo 10.11.

3.9 Long-short term memory networks

Un metodo più efficace per risolvere il problema relativo al gradiente è quello di inserire nella rete ricorrente una memoria interna in modo tale da portare una maggiore stabilità agli stati della rete e agli aggiornamenti dell'algoritmo del gradiente. A questo fine sono state introdotte le *long-short term memory network* (Hochreiter e Schmidhuber, 1997). L'architettura di queste reti è molto più complessa di quella di una vanilla RNN come si può osservare in Figura 3.7.

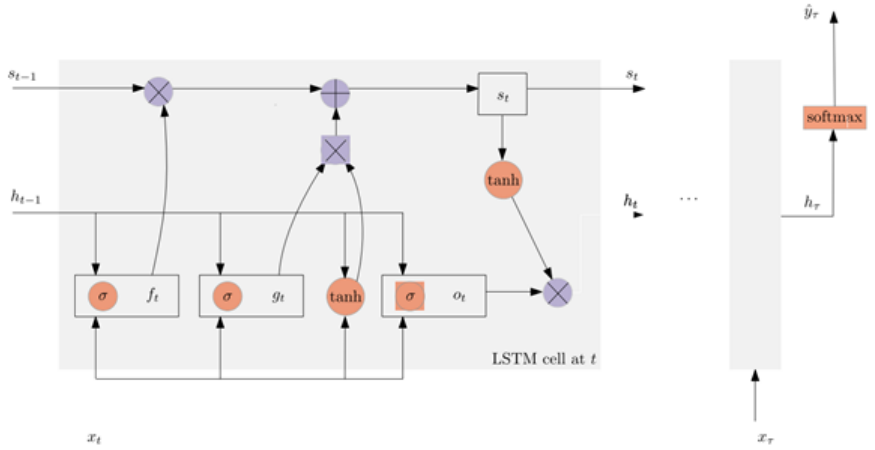


Figura 3.7: Rappresentazione schematica di una rete LSTM

Si può notare che è presente una memoria a lungo termine s_t chiamata *cell state* che ha il ruolo di conservare parte delle informazioni provenienti dagli stati precedenti. Questa memoria viene aggiornata nel tempo dall'algoritmo in una maniera delicata così che ci sia una maggiore persistenza nell'immagazzinamento di informazioni, evitando quindi le instabilità derivanti dal problema del vanishing/exploding gradient. Sono state aggiunte anche tre *gated units* f_t, g_t e o_t che vengono denominate rispettivamente *forget gate, input gate* e *output gate*. Le prime due hanno il ruolo di regolare il flusso di informazioni nel cell state rispettivamente facendo dimenticare e aggiungendo delle nuove informazioni istante dopo istante. La terza invece serve a modificare lo stato h_t dello strato nascosto da cui poi dipendono i valori in output. Le equazioni che regolano queste dinamiche, nell'ipotesi che la rete debba risolvere un problema di classificazione in K classi diverse, sono:

$$
\begin{aligned}
\mathbf{f}_t &= \sigma(\mathbf{W}^f \mathbf{h}_{t-1} + \mathbf{U}^f \mathbf{x}_t + \mathbf{b}^f), \qquad t = 1, ..., T \\
\mathbf{g}_t &= \sigma(\mathbf{W}^g \mathbf{h}_{t-1} + \mathbf{U}^g \mathbf{x}_t + \mathbf{b}^g), \qquad t = 1, ..., T \\
\mathbf{o}_t &= \sigma(\mathbf{W}^o \mathbf{h}_{t-1} + \mathbf{U}^o \mathbf{x}_t + \mathbf{b}^o), \qquad t = 1, ..., T \\
\mathbf{s}_t &= \mathbf{f}_t \otimes \mathbf{s}_{t-1} + \mathbf{g}_t \otimes \tanh(\mathbf{W}^s \mathbf{h}_{t-1} + \mathbf{U}^s \mathbf{x}_t + \mathbf{b}^s), \qquad t = 1, ..., T \\
\mathbf{h}_t &= \tanh(s_t) \otimes \mathbf{o}_t, \qquad t = 1, ..., T \\
\hat{\mathbf{y}}_t &= \text{softmax}(\mathbf{V}\mathbf{h}_t + \mathbf{b}^y),
\end{aligned}
\tag{3.2}
$$

dove σ rappresenta la funzione sigmoid, ovvero $\sigma(x) = \frac{1}{1+\exp -x}$ e dove $\otimes$ indica il prodotto elemento per elemento di due vettori. Le matrici $\mathbf{W}^j, \quad j \in \{f, g, o\}$ hanno dimensione (p, p) dove p rappresenta il numero di nodi nello strato LSTM, le matrici $\mathbf{U}^j, \quad j \in \{f, g, o\}$ hanno dimensione (p, d) con d dimensione del vettore di dati in ingresso, infine la matrice $\mathbf{V}$ ha dimensione (K, p). Di conseguenza il numero totale di parametri che il modello deve stimare è pari a $4(p(d+1)+p^2)+K(p+1)$ poiché si hanno 4 copie della tripletta $(\mathbf{W}, \mathbf{U}, \mathbf{b})$ più i parametri per la risposta al problema di classificazione in K classi.

Adesso si può meglio comprendere il significato delle varie parti di cui la rete si compone. I tre *gate* **f, g, o**, poiché la funzione sigmoid assume valori in $(0, 1)$, regolano se, e eventualmente quante delle informazioni presenti in ciascun neurone possono rispettivamente essere dimenticate dal *cell-state*: $\mathbf{f}_t \otimes \mathbf{s}_{t-1}$. Essere aggiunte al *cell-state*: $\mathbf{g}_t \otimes \tanh(\cdot)$. Infine fluire dal *cell-state* allo stato nascosto: $\tanh(\cdot) \otimes \mathbf{o}_t$. Il *cell-state*, in cui risiede la natura ricorrente della rete, immagazzina le informazioni più importanti che vengono poi utilizzate dallo strato nascosto per decidere cosa trasmettere all'output layer. Oltretutto la forma additiva degli aggiornamenti del cell-state aiuta ad evitare il problema del vanishing-gradient.

La rete neurale LSTM, grazie quindi alle sue caratteristiche che si prestano ad analizzare le serie storiche, sarà quella utilizzata nel corso di tutto questo elaborato.

In questo capitolo sono state presentate in breve le caratteristiche generali delle reti neurali, i principali metodi utilizzati nel processo di addestramento e i più frequenti problemi che possono scaturire da quest'ultimo. Infine sono state introdotte le reti neurali ricorrenti con un particolare focus su quella long-short term memory, utilizzata nella strategia di trading.

Capitolo 4

Modelli finanziari

In questo breve capitolo vengono presentati i principali modelli finanziari utilizzati nel lavoro e il loro significato.

4.1 Capital Asset Pricing Model

Il *Capital Asset pricing Model*, detto anche CAPM, è un modello, proposto in Sharpe (1964), che ha lo scopo di determinare il tasso di rendimento atteso delle attività finanziarie in funzione del rischio. Esso afferma che il rischio è costituito da due diverse componenti: il rischio sistematico, che è legato al tasso di rendimento del mercato e quindi non è diversificabile, e il rischio non sistematico, relativo a una particolare attività finanziaria e che quindi può essere diversificato creando un portafoglio costituito da molte attività finanziarie non correlate tra loro. In altre parole il modello CAPM stabilisce una relazione tra il tasso di rendimento di un asset e il suo livello di rischio. Questo concetto è espresso dalla seguente equazione:

$$\mathbb{E}\left[R_i\right] = R_f + \beta_i \left(\mathbb{E}\left[R_m\right] - R_f\right), \tag{4.1}$$

dove $\mathbb{E}\left[R_i\right]$ rappresenta il rendimento atteso del titolo considerato, R_f è il rendimento privo di rischio, $\mathbb{E}\left[R_m\right]$ è il rendimento atteso del mercato e infine β_i è la sensitività del rendimento atteso in eccesso rispetto al rendimento privo di rischio e può quindi essere definito come:

$$\beta_i = \frac{\mathrm{cov}(R_i, R_m)}{\mathrm{var}(R_m)}.$$

Questo modello, nonostante le numerose critiche ricevute relative principalmente all'esistenza di altre sorgenti di rischio, è molto utilizzato dai *practitioners* in finanza grazie proprio alla sua semplicità. Per una trattazione più approfondita sull'argomento si può vedere Barucci e Fontana (2003), capitolo 5.

Al fine di stimare il coefficiente β_i serve innanzitutto riscrivere l'equazione (4.1) nella forma di un modello lineare, cioè come:

$$R_{i,t} - R_f = \alpha_i + \beta_i(R_{m,t} - R_f) + \epsilon_{i,t},$$

dove $R_{i,t}$ è il rendimento del titolo i, osservato sul mercato al tempo t, R_f è il rendimento privo di rischio osservato e $R_{m,t}$ è il rendimento del portafoglio di mercato. Per ottenere quest'ultimo è pratica diffusa considerare un indice azionario ben diversificato, come per esempio S&P 500. Il termine α_i invece rappresenta l'intercetta del modello lineare. Essa non è presente nella formulazione classica del CAPM, ma viene qui aggiunta per rappresentare meglio i dati di mercato.

Da un punto di vista pratico α ha anche un significato ben preciso: è una misura del rendimento attivo di un investimento, cioè della performance di quell'investimento, paragonata all'andamento del mercato. Quindi un valore di α positivo, indica delle prestazioni superiori dell'investimento rispetto al mercato; un valore negativo delle prestazioni inferiori. Grazie a questa interpretazione, alfa viene utilizzata, nei mercati finanziari moderni, per misurare le performance di fondi o di strategie d'investimento. Infatti, soprattutto nel caso dei primi, che hanno costi di gestione, è importante mantenere il valore di α maggiore dei costi, al fine di offrire guadagni positivi rispetto a un fondo indicizzato, o a un ETF. Nel caso invece di una strategia di investimento alfa rappresenta la capacità di quest'ultima di battere sistematicamente il mercato e di generare rendimenti in eccesso. D'altra parte nel contesto di un mercato efficiente può essere dimostrato che il valore atteso di α sia nullo, in quanto è impossibile battere il mercato in maniera sistematica. I prezzi di mercato incorporano già tutte le informazioni disponibili, e quindi gli asset sono sempre prezzati correttamente. Detto in altra maniera non esiste un modo per identificare e sfruttare sistematicamente i prezzi errati in quanto essi non esistono.

Durante gli anni '70 le evidenze empiriche riguardanti il CAPM erano per lo più positive. In seguito, negli anni '80 una grande quantità di letteratura ha fatto vedere che il rendimento del portafoglio di mercato non era l'unico

fattore di rischio. In particolare determinate caratteristiche delle azioni si sono dimostrate significative nello spiegare i premi per le attività rischiose. Le principali sono il *price earnings ratio* in Basu (1977), la capitalizzazione di mercato, detta anche *size*, in Banz (1981), il *book to market value ratio* in Stattman (1980), il *momentum* e lo *short-term reversal* in De Bondt e Thaler (1985) e Jegadeesh e Titman (1993), e la liquidità in Amihud e Mendelson (1986). Queste considerazioni hanno quindi portato a diverse estensioni del modello CAPM che tengono conto di altri fattori di rischio.

4.2 Modello a 3 fattori di Fama e French

Eugene Fama e Kenneth French, dopo aver osservato che le azioni di società a bassa capitalizzazione e quelle con un elevato book to market ratio, cioè il rapporto tra il patrimonio netto contabile e la capitalizzazione (usato solitamente per misurare se e quanto una società sia sopravvalutata o sottovalutata), proposero in Fama e French (1993) un modello lineare a 3 fattori di rischio come estensione del CAPM. Tale modello è descritto dalla seguente equazione:

$$R_i = R_f + \alpha_i + \beta_i(R_m - R_f) + \beta_{i,s}SMB + \beta_{i,v}HML,$$

dove SMB, che sta per *Small Minus Big*, è un fattore che rappresenta la differenza di rendimenti delle società a bassa capitalizzazione e di quelle ad alta capitalizzazione, mentre HML, che sta per *High Minus Low*, è un fattore che rappresenta la differenza di rendimenti di società con un elevato rapporto tra patrimonio netto e capitalizzazione, solitamente definite *value stocks*, e quelle con rapporto modesto, *growth stocks*. Questi due fattori sono calcolati ordinando le azioni secondo le due nuove metriche nominate sopra (size e book to market ratio), dividendo le azioni in gruppi a seconda dell'ordinamento, combinando questi gruppi, e calcolando le differenze di rendimento dei portafogli creati. Per una descrizione più dettagliata si faccia riferimento all'articolo originale.

Il modello considera quindi il fatto che i rendimenti di value stocks a bassa capitalizzazione superano regolarmente quelli di mercato, mentre growth stocks ad alta capitalizzazione vengono battute dal mercato. Includendo questi due fattori addizionali, il modello viene aggiustato per queste tendenze, rendendolo uno strumento migliore per la valutazione delle performance.

4.3 Altri fattori di rischio

Nel corso del tempo sono stati definiti numerosi altri fattori di rischio, che possono essere aggiunti al modello in modo tale da estenderlo ulteriormente. I principali contributi sono dati da Fama e French (2015) che hanno proposto altri due fattori che tengono conto della profittabilità e dell'investimento. Il primo viene calcolato come differenza di rendimenti tra le società con elevata redditività operativa e le con modesta redditività operativa. Il secondo invece come differenza di rendimenti tra società con alti investimenti e società con bassi investimenti. Altri contributi interessanti sono stati dati da Jegadeesh e Titman (1993) e da Carhart (1997), i quali hanno rispettivamente proposto un indicatore per il fattore short-term reversal (ST_Rev) e uno per il fattore momentum (MOM). Il primo rappresenta l'anomalia per cui azioni con rendimenti relativamente bassi nel recente periodo guadagnano rendimenti più elevati nel periodo a seguire mentre azioni con rendimenti relativamente alti hanno in seguito rendimenti negativi o comunque più bassi. Questo fenomeno, da una parte sta chiaramente alla base delle strategie di pair trading in quanto può essere visto come un ritorno allo stato di equilibrio, ma non è l'unico fattore che le determina, poiché manca la parte di *relative pricing*. La ragione fondamentale che si attribuisce a questo fenomeno è spesso individuata dai ricercatori nella reazione eccessiva degli investitori all'uscita di nuove informazioni e nella successiva correzione della reazione in un breve intervallo di tempo. Il momentum invece rappresenta la tendenza per cui le azioni con prezzi crescenti crescono ulteriormente, e quelle con prezzi decrescenti continuano a decrescere. Questa anomalia, in disaccordo con la teoria dei mercati efficienti, non ha ancora trovato una spiegazione razionale nel mondo finanziario, la si attribuisce principalmente a pregiudizi cognitivi che appartengono all'ambito dell'economia comportamentale. Per una spiegazione dettagliata di come questi due fattori vengono calcolati si faccia riferimento ai testi originali.

Nel corso del lavoro verranno utilizzati modelli a fattori di rischio, in particolare il modello di Fama e French a tre fattori con l'aggiunta del fattore ST_Rev e del fattore MOM, che quindi può essere espresso come:

$$R_{i,t} - R_f = \alpha_i + \beta_i(R_{m,t} - R_f) + \beta_{i,s}SMB_t + \beta_{i,v}HML_t + \\ + \beta_{i,r}ST\_Rev_t + \beta_{i,m}MOM_t + \epsilon_{i,t}. \quad (4.2)$$

Questo breve capitolo ha presentato alcuni modelli lineari che hanno l'o-

biettivo di mettere in relazione i rendimenti ai rischi che li generano. Questi modelli sono utili all'interno del lavoro, sia per determinare quali sono le cause dei rendimenti della strategia di pair trading, sia per introdurre *alfa* come nuova misura di performance della strategia.

Capitolo 5

Metodologia

Questo capitolo ha lo scopo di presentare in dettaglio le tecniche utilizzate nel tentativo di replicare la strategia proposta in Flori e Regoli (2021). In particolare viene spiegata la procedura per eseguire i test di cointegrazione che servono a definire l'insieme di azioni che verranno utilizzate nella strategia. Viene definito il problema di classificazione utilizzato dalla rete neurale ed è illustrato tutto il procedimento necessario all'addestramento di quest'ultima. Infine si mostrano le metriche utilizzate per valutare sia la precisione delle previsioni sia il profitto che la strategia è in grado di generare.

5.1 Dati

All'interno di questo lavoro sono considerati due dataset. Un primo dataset con dati con frequenza giornaliera analogo a quello in Flori e Regoli (2021) ed un secondo con dati ad alta frequenza rilevati ogni minuto. In questo capitolo viene descritto il primo dataset mentre il secondo è descritto e analizzato nel Capitolo 7.

In questo primo dataset si considerano i prezzi di chiusura delle azioni appartenenti all'indice S&P 500 nel periodo che va dall'inizio di gennaio 2000 fino alla fine di giugno 2019. In particolare vengono utilizzate tutte quelle azioni che, all'inizio di ogni anno, dal 2003 al 2019, appartengono all'indice, e delle quali vengono utilizzati i dati dei tre anni precedenti e di quello in considerazione.

Per ciascuno dei 17 periodi di 4 anni che si possono formare non sono sempre disponibili tutti i dati di ogni azione che appartiene all'indice all'inizio

dell'anno: le ragioni possono essere diverse, come per esempio la quotazione avvenuta meno di tre anni prima, il fallimento entro l'anno, operazioni di M&A e così via. In questi casi vengono considerati solo i dati disponibili, e quindi su un periodo più corto. Si fa riferimento ai prezzi di chiusura e ai volumi di scambio giornalieri. Le informazioni sono state raccolte da Refinitiv, utilizzato tramite Eikon Data API di Python. Refinitiv fornisce la lista di azioni che costituiscono l'indice al momento, dette *constituents*, e la lista di *leavers* e *joiners*, cioè le azioni che escono ed entrano nell'indice con la rispettiva data. Con queste informazioni è possibile ricostruire facilmente l'insieme di azioni appartenenti all'indice all'inizio di ogni anno. Refinitiv fornisce anche i prezzi di chiusura aggiustati per variazioni di capitale, per esempio *stock-split*, ma non per *cash-dividends*. Quindi per fare in modo che i risultati della strategia di trading siano più attendibili, è importante aggiustare i prezzi per i dividendi. Prima di spiegare questa procedura è utile introdurre qualche concetto teorico sui rendimenti che verranno utilizzati ampiamente all'interno di questo lavoro.

5.1.1 Rendimenti e proprietà

In finanza sono solitamente usate due diverse tipologie di rendimenti, che hanno proprietà diverse, ma allo stesso tempo sono spesso intercambiabili: i rendimenti semplici, o *simple returns*, e i rendimenti logaritmici, o *log-returns*. I primi vengono definiti come:

$$R_t = \frac{P_t}{P_{t-1}} - 1$$

mentre i secondi come:

$$r_t = \log\left(\frac{P_t}{P_{t-1}}\right),$$

dove P_t è il prezzo dell'azione al giorno t. Essi sono utilizzabili in maniera alternativa perché lo sviluppo di Taylor del logaritmo mostra che per x piccoli vale la relazione $\log(1+x) \sim x$. Allo stesso tempo però, hanno proprietà differenti che possono essere utilizzate in diverse situazioni. I rendimenti semplici si aggregano sulle azioni: il rendimento di un portafoglio è infatti calcolabile come la somma pesata dei rendimenti delle diverse azioni. I log-rendimenti invece, grazie alle proprietà dei logaritmi, possono sia essere aggregati nel tempo, cioè il rendimento logaritmico di un periodo temporale è la somma

dei rendimenti logaritmici delle partizioni di quel periodo, in formule:

$$\log\left(\frac{P_{t+h}}{P_t}\right) = \sum_{i=0}^{h-1} \log\left(\frac{P_{t+i+1}}{P_{t+i}}\right),$$

sia anche permettono di calcolare il rendimento di una posizione corta come il negativo del rendimento della posizione lunga.

Infine è molto semplice convertire un rendimento semplice in uno logaritmico e viceversa, valgono infatti le seguenti relazioni:

$$r = \log(R+1) \qquad R = \exp(r) - 1.$$

5.1.2 Prezzi aggiustati

Per ottenere i prezzi di chiusura aggiustati per i dividendi, innanzitutto sono stati calcolati i log-rendimenti giornalieri sui prezzi di chiusura. Dopodiché sono stati scaricati, sempre tramite Refinitiv, i dividendi pagati nel periodo analizzato. Con questi ultimi è quindi possibile aggiustare i rendimenti nelle date di stacco cedola, cioè quelle date dopo le quali il compratore dell'azione non ha più diritto al dividendo successivo, attraverso la formula:

$$r_t = \log\left(\frac{P_t + D_t}{P_{t-1}}\right).$$

A questo punto tutti i rendimenti tengono conto dei dividendi e quindi, sfruttando le proprietà dei log-rendimenti, si ottengono i prezzi aggiustati come:

$$P_t = P_1 \exp\left(\sum_{j=2}^{t} r_j\right) \qquad \forall t \in \{2, 3, .., T\}.$$

5.1.3 Pulizia del dataset

Prima di iniziare l'analisi il dataset è stato analizzato e controllato, tramite la libreria Pandas di Python, al fine di integrare i dati mancanti, correggere errori oppure dati alterati e rimuovere eventuali problemi. In particolare, dopo aver aggiustato i prezzi, sono state rimosse dalle serie storiche tutte quelle date in cui è stato pagato un dividendo a mercato chiuso. Queste date riguardavano essenzialmente festività nazionali (Memorial day, Thanksgiving,

President day, e Natale), fine settimana e eventi straordinari (caduta delle Torri Gemelle e l'uragano Sandy) che hanno causato la chiusura dei mercati.

Nel caso di dati mancanti, i volumi di scambio sono stati integrati utilizzando il volume al giorno precedente, i prezzi di chiusura il valore di interpolazione lineare tra il giorno precedente e quello successivo e i rendimenti sono stati corretti in modo tale che fossero coerenti con l'interpolazione dei prezzi. Infine è stato creato un calendario con tutte le date di mercato aperto nel periodo di analisi per controllare che ogni azione avesse una serie storica completa nel periodo di appartenenza all'indice S&P 500. Nei pochi casi di data mancante, i valori sono stati integrati come descritto sopra.

Per quanto riguarda la correttezza dei dati sono stati applicati diversi filtri quantitativi per la ricerca di outliers, di rendimenti estremi e di eccessive differenze tra il prezzo di chiusura e il prezzo di chiusura aggiustato. E' noto che i rendimenti non seguono una distribuzione gaussiana, quindi gli outliers, che possono influenzare in maniera significativa l'analisi, sono stati definiti e ricercati come fatto in Benth et al. (2008), paragrafo 5.1.1: per ogni azione viene considerata la serie storica dei rendimenti, della quale si calcolano il primo quartile Q_1 e il terzo quartile Q_3. Si definisce l'intervallo interquartile come $IQR = Q_3 - Q_1$, e un'osservazione del rendimento r è identificata come outlier se $r \leq Q_1 - 3IQR$ oppure se $r \geq Q_3 + 3IQR$.

Per rilevare i rendimenti estremi e le eccessive differenze tra il prezzo di chiusura e quello di chiusura aggiustato, sono stati applicati dei filtri, ancora più semplici, per mostrare per esempio i rendimenti maggiori del 15% oppure quelle serie storiche tali per cui:

$$\frac{P_T^{adj} - P_T^{close}}{P_T^{close} * T} \geq q,$$

dove q è una soglia positiva. Questo indicatore caratterizza quelle azioni per cui la differenza tra prezzo aggiustato e prezzo di chiusura differisce di molto relativamente al prezzo di chiusura e alla lunghezza della serie storica. La soglia q viene determinata euristicamente in modo tale da selezionare un numero adeguato di azioni.

Qualora un'azione venisse rilevata da uno dei filtri appena descritti è stata controllata e paragonata con i dati di altri info provider, e nei pochi casi in cui è stato riscontrato un errore effettivo è stato corretto.

5.1.4 Dati dei Modelli a Fattori

I dati necessari per i Modelli a tre fattori di rischio di Fama e French e i fattori aggiuntivi, presentati nel precedente capitolo, sono quelli riportati sul sito di Kenneth R. French [1].

5.2 Software e Hardware

Tutte le analisi sono state eseguite utilizzando il software open source Python 3 (Van Rossum e Drake, 2009). In particolare, la pulizia dei dati è stata eseguita tramite Pandas (pandas development team, 2020), una libreria per la manipolazione e l'analisi dei dati. L'analisi econometrica delle serie storiche è stata fatta utilizzando il modulo Statsmodels (Seabold e Perktold, 2010) che fornisce classi e funzioni sia per la stima di molti modelli statistici e sia per l'esecuzione di test statistici. Le reti neurali LSTM sono implementate in Keras (Chollet et al., 2015), la libreria più comunemente utilizzata per l'apprendimento automatico e il deep learning, con Tensorflow (Abadi et al., 2015) come backend.

I programmi sono stati poi eseguiti su una CPU comune (AMD Ryzen 5, 2.00 GHz e 8 GB RAM), salvo la parte di training della rete che è stata eseguita su macchine più potenti (HPC cluster 4x Xeon E5-2640 v4 2.4GHz 384GB RAM e 4x Xeon E5-4610 v2 @2.3GHz 1.2TB RAM). Per ridurre i tempi di calcolo gli script relativi alla cointegrazione e all'addestramento delle reti sono stati eseguiti in parallelo.

5.3 Periodi di analisi

Come affermato sopra, i dati sono stati considerati nel periodo che va dall'inizio del 2000 alla fine di giugno 2019. Con questi 19 anni e mezzo di dati vengono eseguite 17 analisi su periodi di tempo parzialmente sovrapposti. Più in particolare, all'inizio di ogni anno, dal 2003 al 2019, si considerano tutte le azioni presenti nell'indice il primo giorno di mercato. Si utilizza il periodo di tre anni precedente per il training dei parametri (test di cointegrazione e apprendimento della rete neurale) e l'anno presente come anno su cui testare la strategia di trading, con un campione di dati out-of-sample. Le

[1] Vedi: https://mba.tuck.dartmouth.edu/pages/faculty/ken.french/

17 analisi sono quindi svolte su 17 periodi temporali di 4 anni (l'ultimo di 3 anni e mezzo) che traslano temporalmente all'inizio di ogni anno.

5.4 Analisi di cointegrazione

La strategia del pair trading si fonda sul comportamento di azioni che hanno una dinamica passata di rendimenti molto simile e che quindi possono essere considerate delle sostitute l'una dell'altra. In letteratura sono stati proposti diversi metodi per quantificare la similitudine di due, o più, titoli e quindi individuare le coppie candidate al trading. Si possono dividere in 4 principali categorie, come affermato in Krauss (2017): i metodi basati sulla minimizzazione della distanza relativa, i metodi basati sulla cointegrazione, quelli che sfruttano il concetto di ritorno verso la media dello spread e infine quelli basati sul controllo ottimo stocastico.

In generale, la maggior parte della letteratura propende per il primo, (si veda per esempio Gatev et al. (2006) e Do e Faff (2010)), o il secondo approccio, (Vidyamurthy (2004) e Rad et al. (2016)). I metodi basati sulla distanza, presentati qui per completezza, consistono essenzialmente nel misurare la distanza, tra i prezzi normalizzati, o tra i rendimenti, di ogni coppia di titoli in un determinato periodo di tempo. Cioè nel calcolare una delle due quantità:

$$\sum_{t=1}^{T} \left(P_t^i - P_t^j\right)^2 \qquad \sum_{t=1}^{T} \left(r_t^i - r_t^j\right)^2$$

per ogni coppia di titoli (i, j) e selezionare un certo numero, k, di coppie per cui tali quantità risultano più basse.

In questo lavoro, seguendo quanto fatto da Flori e Regoli (2021), si è optato per il metodo della cointegrazione come contesto teorico per l'identificazione di possibili coppie. Questa scelta è giustificata anche da Huck e Afawubo (2015), dove è stato dimostrato empiricamente che i metodi basati sulla distanza generano rendimenti in eccesso insignificanti, mentre quelli basati sulla cointegrazione, sia generano rendimenti positivi, stabili e robusti, sia diminuiscono sensibilmente il rischio di non convergenza delle coppie.

La cointegrazione, come discusso nel secondo capitolo, indica una relazione di comovimento di due variabili nel tempo. Infatti la differenza di due serie cointegrate si comporta come una serie stazionaria e dunque, così come un processo *mean-reverting*, le deviazioni dalla media sono solo temporanee e

col tempo torneranno al loro valore di equilibrio. Analogamente le differenze di prezzi che si creano tra titoli cointegrati sono percepite come deviazioni della differenza dei prezzi dal valore di equilibrio dovute ad inefficienze del mercato, quindi anche queste sono percepite come solo momentanee, e si prestano bene ad essere sfruttate attraverso le strategie di pair trading.

Da un punto di vista più strettamente economico, quando due titoli risultano cointegrati si può pensare che i fattori che influenzano la loro dinamica dei prezzi, e che rendono ciascuno dei due prezzi non stazionario, sono gli stessi; o in termini finanziari, i due asset hanno una simile esposizione al rischio, tale per cui i loro prezzi si muovono in maniera coordinata. Esempi di coppie che possono risultare cointegrate sono titoli che appartengono allo stesso settore oppure WTI crude oil e Brent crude oil; in altre parole, la cointegrazione viene percepita come una misura di somiglianza tra due diversi prodotti.

5.4.1 Metodologia utilizzata per la cointegrazione

La metodologia utilizzata per individuare le azioni cointegrate si compone di diversi step che sono descritti in dettaglio qui di seguito. Innanzitutto viene fissato l'anno di analisi t con $t \in \{2003, 2004, .., 2019\}$ e vengono considerati i prezzi di chiusura aggiustati dei 3 anni precedenti, ovvero nel periodo $[t-3, t-1]$, per ogni azione S^i presente nell'indice S&P 500 all'inizio dell'anno. I prezzi vengono normalizzati rispetto al primo valore disponibile nel periodo di analisi.

Tutte le serie storiche di lunghezza minore di 240 istanti temporali (giorni di mercato aperto) sono state scartate, in quanto la cointegrazione esprime il comovimento di due azioni nel lungo termine. E' quindi necessario avere a disposizione i dati su un periodo di tempo sufficientemente lungo (un anno corrisponde a circa 252 giorni di mercato) per individuare questo tipo di relazione. Oltretutto come si vedrà in seguito 240 è il *lookback period* utilizzato per la rete neurale: senza almeno 240 dati è quindi impossibile inserire le informazioni relative a quel titolo nel processo di apprendimento.

Per ogni serie storica viene testato l'ordine di integrazione, come descritto in Sezione 2.4, attraverso il test *Augmented Dickey-Fuller* presentato in equazione (2.3). ADF utilizza un massimo numero di lag pari a 5, per evitare eventuali stagionalità settimanali, il *Bayesian Information Criterion* come criterio di ricerca, e una soglia di accettazione pari all'1%. Le azioni vengono quindi raggruppate per ordine di integrazione, in particolare quelle di ordine

0 vengono scartate in quanto già stazionarie. I test di cointegrazione vengono eseguiti su tutte le possibili coppie di azioni integrate dello stesso ordine. Il numero di coppie da testare è molto alto, infatti se tutte le serie fossero $I(1)$, considerando 505 azioni[2] presenti nell'indice, dovrebbero essere eseguiti $\frac{505*504}{2} = 127260$ test. Ogni coppia di azioni (S_t^i, S_t^j) viene testata sia attraverso il test di Engle-Granger, sia attraverso quello di Johansen, presentati in Sezione 2.5.1.

Bisogna prestare attenzione al fatto che l'implementazione in Python di queste procedure presuppone che le serie siano integrate di ordine 1, quindi, in caso di ordine $d > 1$, è necessario eseguire i test sulle serie differenziate $d - 1$ volte.

Come per il test ADF, il numero massimo di lag utilizzati è pari a 5, il criterio di ricerca è BIC, la soglia di accettazione pari all'1%, però, seguendo l'esempio di Vidyamurthy (2004) e Rad et al. (2016), non viene inserita nessuna intercetta nell'equazione (2.7). Per ridurre i falsi positivi vengono ritenute cointegrate, da ora in poi $S_t^i \overset{ci}{\sim} S_t^j$, solo quelle coppie di azioni che risultano tali secondo entrambi i test. Infine per ogni azione i e anno t viene definito il *gruppo di cointegrazione* CG_t^i come l'insieme di tutte le azioni j che risultano cointegrate ad i in quell'anno, in formule:

$$CG_t^i = \left\{ S_t^j : S_t^i \overset{ci}{\sim} S_t^j \right\}.$$

E' interessante notare che la relazione di cointegrazione $\overset{ci}{\sim}$ in principio è simmetrica, cioè se $S_t^i \overset{ci}{\sim} S_t^j$ allora $S_t^j \overset{ci}{\sim} S_t^i$. Da un punto di vista computazionale però i test possono restituire un risultato asimmetrico. Nell'algoritmo, così come in Flori e Regoli (2021), è stato deciso di definire cointegrati due titoli se almeno una delle due relazioni $S_t^i \overset{ci}{\sim} S_t^j$ o $S_t^j \overset{ci}{\sim} S_t^i$ risulta soddisfatta.

Analogamente si potrebbe immaginare che la relazione di cointegrazione risulti transitiva, ma nemmeno questa proprietà si verifica nella realtà e quindi non è detto che per due titoli cointegrati di indici i e j risulti $CG_t^i = CG_t^j$.

I gruppi di cointegrazione CG_t^i, definiti per ogni azione i e per ogni anno t, vengono quindi utilizzati per addestrare le reti neurali e verificare le performance della strategia.

[2]L'indice è composto da 500 società diverse, ma alcune di queste emettono più di una tipologia di titoli.

5.5 Classificazione

Le occasioni di arbitraggio statistico, sfruttate dalle strategie di pair trading, sono viste come deviazioni temporanee dall'equilibrio. Per questo motivo, una volta che i gruppi di cointegrazione sono stati identificati, è importante costruire degli indicatori che abbiano una buona capacità nel rilevare queste deviazioni. Nel corso del tempo ricercatori e professionisti hanno proposto molti indicatori basati su concetti teorici differenti; di seguito sono riportate le principali correnti di pensiero. Per esempio Gatev et al. (2006) ha proposto l'idea di opportunità di pair trading basate su deviazioni in termini di prezzi, poiché questo approccio è quello che meglio approssima la descrizione di come i trader stessi scelgono le coppie. In alternativa, Chen et al. (2019) studia l'idea per cui le azioni che hanno avuto maggiori rendimenti e che quindi hanno sovraperformato, porteranno rendimenti più modesti in futuro, viceversa quelle che hanno sottoperformato ne porteranno di più elevati. Infine Blitz et al. (2013) dimostra che le strategie di reversione costruite sui residui dei rendimenti che non mostrano una dipendenza dinamica rispetto ai fattori di rischio generano rendimenti molto più elevati rispetto a strategie di mean-reversion classiche.

In questo elaborato, invece di limitarsi a rilevare le discrepanze di prezzo tra titoli cointegrati si usano le reti neurali per fare previsioni riguardanti la variazione dei rendimenti relativi, come in Flori e Regoli (2021). L'indicatore cattura la probabilità che il divario di rendimenti tra un'azione e il suo gruppo di cointegrazione aumenti o diminuisca nel futuro prossimo. Più in particolare la differenza di rendimenti al giorno t è calcolata come:

$$\Delta r_t^i = r_t^i - r_t^{CG,i},$$

dove $r_t^{CG,i}$ è la media aritmetica dei rendimenti delle azioni che appartengono a CG_t^i. La probabilità che questa differenza aumenti, in simboli $P(\Delta r \nearrow)$, è ottenuta attraverso una rete neurale LSTM, descritta in dettaglio in seguito, impostata per risolvere un problema di classificazione binario dove la variabile target y_{t+1} vale:

$$y_{t+1} = \begin{cases} 1 & \text{se} \quad \Delta r \geq 0 \\ 0 & \text{se} \quad \Delta r < 0 \end{cases} \tag{5.1}$$

e $\Delta r = \Delta r_{t+1}^i - \Delta r_t^i$. In questo modo, l'output della rete $\hat{y}_{t+1}$ rappresenta la probabilità che Δr aumenti in un orizzonte temporale di un giorno. La particolarità di questo indicatore, a differenza della maggior parte di quelli

utilizzati in letteratura, risiede nel fatto che è costruito non solo sfruttando le informazioni disponibili nel giorno in cui viene presa la decisione di investimento, ma anche considerando tutta la storia di mercato delle serie storiche coinvolte, al fine di arricchire le informazioni usate per la predizione. Inoltre, sempre grazie alle proprietà delle reti LSTM, l'indicatore impara dai dati passati e presenti gli schemi ripetitivi utili nel predire movimenti futuri, soppesando in maniera appropriata i differenti contributi temporali e non limitandosi a una semplice istantanea delle informazioni attuali.

5.6 Rete LSTM

Nella Sezione 3.9 è stato mostrato che le reti LSTM, grazie alle loro connessioni retroattive, riescono a analizzare le dipendenze di lungo termine nei dati, e sono quindi ben qualificate per classificare, processare e fare previsioni basate su serie storiche. Qui di seguito viene presentata la procedura dettagliata con la quale le reti neurali sono state addestrate per svolgere le analisi.

5.6.1 Training, validation e test set

Innanzitutto, in linea con quanto fatto nella procedura di cointegrazione, sono stati considerati periodi di analisi di 4 anni. Fissato un anno t, il periodo di analisi, $[t-3, t]$, è stato suddiviso in due sottoperiodi: il primo, che usa i dati dei tre anni precedenti, è utilizzato come training set, e serve quindi per la calibrazione dei parametri e l'apprendimento della rete, mentre il secondo, che utilizza i dati dell'anno in corso, viene utilizzato come test set, cioè sfrutta la rete neurale addestrata per calcolare $P(\Delta r \nearrow)$, fare predizioni e mettere in piedi la strategia di negoziazione. Il validation set è invece costruito, in ogni intervallo di 3 anni, selezionando in maniera casuale il 10% dei dati presenti nel training set.

5.6.2 Preprocessing dei dati

Le reti neurali ricorrenti sono addestrate per fare previsioni basate su una successione di valori consecutivi presi dai dati. Le serie storiche hanno però la forma di una lunga successione di osservazioni passate, ed è quindi necessario riorganizzare i dati in modo tale che possano essere maneggiati dalle reti

LSTM. Seguendo l'esempio di Fischer e Krauss (2018) quella denominata *sliding window*, per cui la serie storica completa viene divisa in una famiglia di sottosuccessioni di lunghezza fissata, ognuna traslata nel tempo rispetto alla precedente. Ognuna di queste sottosuccessioni, a cui si aggiunge la variabile target, rappresenta un singola successione di training utilizzata dalla rete. Una rappresentazione schematica è osservabile nella parte alta di Figura 5.1.

Quando si utilizza questa procedura bisogna scegliere il valore di un ulteriore iperparametro della rete: la lunghezza della sottosuccessione, detta anche *lookback period*. La scelta del numero di istanti temporali da includere è influenzata da due diversi aspetti contrastanti: da una parte se il numero è troppo piccolo si rischia di perdere qualche informazione importante della serie storica, dall'altra se la successione è troppo lunga il costo computazionale per il training può diventare molto elevato.

In questo lavoro, seguendo l'esempio di Flori e Regoli (2021), è stato utilizzato un lookback period τ di 240 giorni di mercato, quindi circa un anno. In particolare le sottosuccessioni multidimensionali in input sono composte da 3 diverse serie storiche: i valori passati dei rendimenti dell'azione i-esima r^i, quelli dei volumi di scambio V^i, e infine la differenza dei rendimenti tra l'azione i e la media dei rendimenti del gruppo di cointegrazione Δr^i. Queste sottosuccessioni vengono create per ogni azione che appartiene ad almeno un gruppo di cointegrazione e la traslazione temporale è giornaliera. Quindi per ogni azione nella finestra di training di durata pari a tre anni vengono create circa $750 - 240$ sottosuccessioni in input: 750 sono circa i giorni di mercato nei tre anni considerati e 240 sono i dati utilizzati per formare la prima successione. Una singola successione di training è quindi così costituita:

$$\begin{pmatrix} \Delta r^i_\theta & \Delta r^i_{\theta-1} & ... & \Delta r^i_{\theta-\tau+1} \\ V^i_\theta & V^i_{\theta-1} & ... & V^i_{\theta-\tau+1} \\ r^i_\theta & r^i_{\theta-1} & ... & r^i_{\theta-\tau+1} \end{pmatrix} \tag{5.2}$$

con il giorno θ che varia all'interno della finestra temporale di 3 anni. Più precisamente, se l'anno in considerazione è l'anno t e quindi la finestra di training è data da $[t-3, t)$, indicando con θ_0 il primo giorno dell'anno t, allora $\theta \in [\theta_0 - 3\,\text{anni} + \tau, \theta_0)$.

Sempre all'interno dell'intervallo di training di tre anni vengono calcolate su tutte le azioni presenti in almeno un gruppo di cointegrazione la media e la deviazione standard delle tre caratteristiche in ingresso. Questi valori vengono utilizzati per standardizzare i dati in ingresso, procedura che velocizza il processo di apprendimento (Bishop, 2006).

5.6.3 Architettura e addestramento

La rete neurale, come in Flori e Regoli (2021), è costituita da: i) uno strato in ingresso caratterizzato da 3 caratteristiche e una dimensione di lookback pari a 240 giorni, ii) un singolo strato nascosto del tipo LSTM con 35 nodi con funzioni di attivazione sigmoid e tanh (cfr. eq. (3.2)), iii) uno strato batch normalization, e infine iv) uno strato di output denso con due nodi e funzione di attivazione softmax. Quest'ultima, come affermato precedentemente, serve a trasformare i valori in uscita dallo strato precedente in probabilità di appartenenza a una delle due classi del problema di classificazione. La funzione di loss utilizzata, coerentemente con l'output softmax, è quindi *categorical cross-entropy*, descritta nell'equazione (3.1). Il numero totale di parametri da calibrare è quindi pari a 5602: 5460 per lo strato nascosto LSTM, 70 per lo strato batch normalization e 72 per lo strato di output.

Il processo di addestramento consiste in un problema di minimizzazione non convesso, che in Keras è risolto attraverso un metodo integrato che implementa l'algoritmo di BPTT descritto in Sezione 3.8.2. L'ottimizzatore utilizzato è RMSprop con un learning rate pari a 0.001, un fattore di decadenza ρ pari a 0.9; il batch size è impostato a 32. Viene anche impiegata la tecnica dell'early-stopping per scegliere il numero di epoche ottimale: il numero massimo di epoche è impostato a 50, e l'algoritmo viene fermato quando la *validation loss*, cioè la crossentropia calcolata sul validation set, non decresce per 10 epoche consecutive, in gergo *patience interval*. In caso di arresto prematuro del processo di training i parametri vengono ripristinati con quelli che presentavano il valore minimo della validation loss. In Tabella 5.1 vengono riassunti i valori degli iperparametri utilizzati. Infine

learning rate	decadenza	batch size	n° max epoche	patience
0.001	0.9	32	50	10

Tabella 5.1: La tabella mostra gli iperparametri utilizzati per l'architettura e per l'addestramento della rete.

un'altra tecnica impiegata durante l'addestramento consiste nel riordinare casualmente le diverse sottosuccessioni che vengono date in ingresso alla rete prima di ogni epoca. In questo modo, in epoche diverse vengono creati e analizzati batch diversi, con lo scopo di ridurre la varianza. Ciò comporta essenzialmente due vantaggi: il primo consiste nel ridurre il rischio di mini-batch molto differenti dal resto dei dati, i quali indirizzerebbero l'algoritmo

del gradiente verso minimi locali non ottimi; il secondo invece fa in modo di diminuire il rischio di overfitting, in quanto dopo ogni rimescolamento viene introdotta della nuova aleatorietà. In questo modo sia si velocizza la convergenza dell'algoritmo, sia si migliora la sua abilità nel generalizzare a dati mai visti.

5.6.4 Predizione

Durante la fase di predizione vengono sfruttate le sequenze in input, sempre in forma di matrice, come in (5.2), per calcolare le probabilità di appartenenza alle due diverse classi. In particolare la rete restituisce, per ogni successione, un vettore bidimensionale dove la prima componente rappresenta la probabilità per cui $y_{t+1} = 0$ e la seconda componente quella per cui $y_{t+1} = 1$ (cfr. equazione (5.1)). Questi valori di probabilità sono quindi usati per la classificazione.

Il numero di successioni, fissato il periodo di predizione $[t, t + 1)$, per ogni azione è di circa 250. Infatti, utilizzando la stessa notazione di Sezione 5.6.2, $\theta \in [\theta_0, \theta_f - 1]$, dove θ_f è l'ultimo giorno di mercato dell'anno t. Una rappresentazione schematica è data in Figura 5.1.

5.7 Metriche di precisione

Esistono molte metriche diverse che possono essere usate per misurare le performance out-of-sample di un classificatore o di un predittore; campi differenti hanno preferenze differenti dovute al perseguimento di diversi obiettivi. Nel contesto della classificazione una distinzione importante riguarda la frequenza con la quale ogni categoria si presenta nella popolazione, definita prevalenza. Esistono infatti sia metriche dipendenti dalla prevalenza, che indipendenti, ma hanno proprietà differenti. Dato un problema di classificazione binario, come nel caso definito dal predittore utilizzato, si possono definire quattro diverse combinazioni di categorie assegnate e categorie effettive: i veri positivi TP (assegnamento corretto e positivo), i veri negativi TN (assegnamento corretto e negativo), i falsi positivi FP (assegnamento positivo ed errato) e infine i falsi negativi FN (assegnamento negativo ed errato). La definizione di queste combinazioni è rappresentata in Figura 5.2.

In questo elaborato si è optato per metriche indipendenti dalla prevalenza, in quanto è naturale immaginare che l'indicatore y_{t+1} assuma metà delle volte

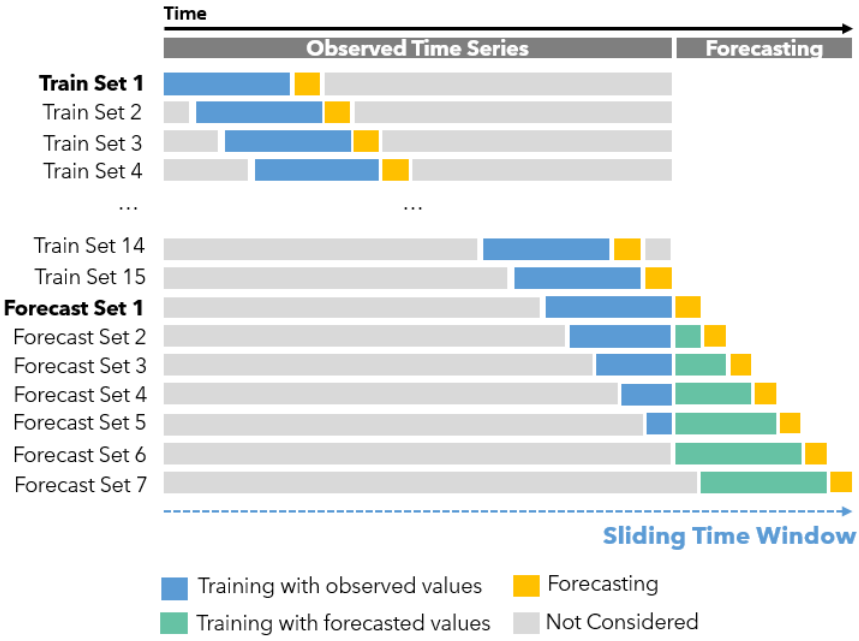


Figura 5.1: Generazione delle sottosuccessioni utilizzate nel processo di training e predizione, traslate nel tempo di un'unità. Le successioni di training (in blu) vengono costruite con le variabili esplicative e il valore target (in giallo). Man mano che la finestra si sposta verso destra si ottengono tutte le sequenze utili per l'addestramento. Una volta che la finestra arriva alla fine del periodo di training (observed time series) comincia la generazione delle serie di predizione: la prima successione di predizione utilizzerà gli ultimi 240 dati del periodo di osservazione per fare la prima previsione (Forecast set 1). La seconda previsione utilizza gli ultimi 239 dati più il primo dato del periodo di predizione (in verde). E così via.

valore 1 e l'altra metà valore 0. In particolare sono state utilizzate le seguenti metriche: *accuracy*, *area under ROC* e *log-loss*.

5.7.1 Accuracy

L'accuracy è una misura statistica della capacità di un classificatore nell'identificare o escludere una certa condizione, cioè è la proporzione di predizioni corrette rispetto al numero totale di casi esaminati. In formule:

$$Acc = \frac{TP + TN}{TP + TN + FP + FN}.$$

		Predicted Category	
		C_1 (+)	C_2 (–)
Actual Category	C_1 (+)	True Positive	False Negative
	C_2 (–)	False Positive	True Negative

Figura 5.2: Matrice di confusione: combinazioni di categorie effettive e predette dal classificatore.

Come paragone, nel caso in cui la frequenza delle due categorie è confrontabile, come in questo caso, il classificatore casuale, cioè che assegna in maniera totalmente casuale le classi, ha un valore di accuracy pari a 0.5. Inoltre bisogna specificare che questa è una metrica dipendente da una soglia: nel lavoro è stata utilizzata un'accuracy con una soglia di probabilità pari a 0.5, cioè sono state classificate come 1 tutte le osservazioni per cui la probabilità di un aumento della differenza dei rendimenti era maggiore o uguale a 0.5.

5.7.2 Area under ROC

La curva *receiver operating characteristic*, o curva ROC, è un grafico che mostra la bontà di un classificatore binario al variare della soglia di classificazione. Questa curva si ottiene disegnando il *true positive rate* TPR in funzione del *false positive rate* FPR, dove:

$$TPR = \frac{TP}{TP + FN} \quad \text{e} \quad FPR = \frac{FP}{FP + TN}.$$

La curva ROC è quindi definita da FPR e TPR rispettivamente come assi x e y, il che raffigura il trade-off tra i veri positivi e i falsi positivi. Per chiarire le idee, osservando la Figura 5.3, il miglior classificatore possibile produrrebbe un punto nell'angolo in alto a sinistra, di coordinate $(0, 1)$, che rappresenta il caso di assenza di falsi positivi e di falsi negativi. Un classificatore casuale restituirebbe un punto sulla diagonale principale: una popolazione con classi effettive bilanciate infatti avrebbe la stessa frequenza di veri positivi e di falsi positivi per ogni valore di soglia della classificazione. La diagonale che divide il primo quadrante fa anche da divisore rispetto alle buone e alle cattive classificazioni. Per passare dai singoli punti all'intera curva, una volta ottenuti i valori predetti dal classificatore, è sufficiente calcolare TPR

e FPR al variare della soglia di classificazione e rappresentare i vari punti ottenuti sul grafico.

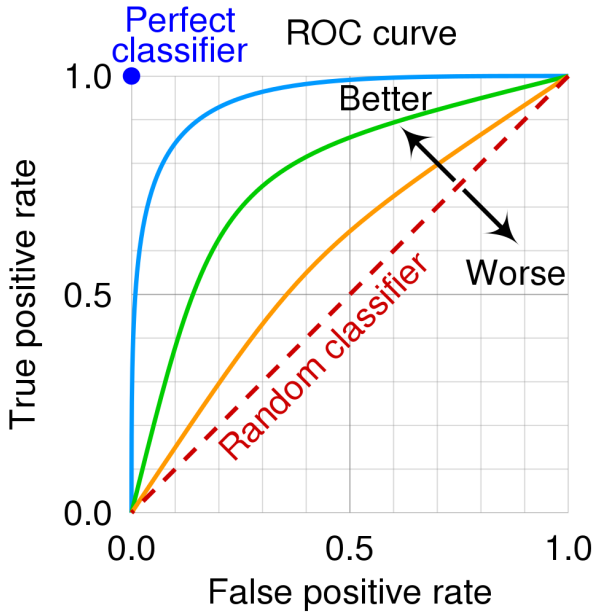


Figura 5.3: Esempi di curva ROC: il classificatore casuale è rappresentato dalla linea rossa tratteggiata. A classificatori migliori corrispondono curve sempre più piegate verso il punto (0,1).

L'*Area Under ROC*, o *Area Under the Curve* AUC, è una misura della capacità del classificatore di distinguere tra le classi ed è usata come valore riassuntivo della curva ROC: ovviamente maggiore è il valore AUC, maggiore è la bontà del classificatore. Il classificatore perfetto ha un valore pari a 1 e quello casuale pari a 0.5. Più il valore AUC è vicino a 1, più il classificatore ha buona probabilità di distinguere la classe positiva da quella negativa, in quanto è in grado di rilevare un numero maggiore di veri positivi e veri negativi rispetto al numero di falsi positivi e falsi negativi.

5.7.3 Logloss

La logloss, già introdotta nell'equazione (3.1), quantifica la precisione di un classificatore penalizzando le classificazioni sbagliate. Minimizzare la logloss è praticamente equivalente a massimizzare l'accuracy, ma con una sottile

differenza. Nel caso binario la formula può essere riscritta come:

$$L = -\frac{1}{N}\left(\sum_{j=1}^{N} y_j \log(\hat{y}_j) + (1-y_j)\log(1-\hat{y}_j)\right)$$

quindi per ogni classificazione solo la probabilità della classe corretta contribuisce alla somma. Se si considera il grafico del logaritmo, è chiaro che questa metrica penalizza molto le classificazioni che sono fiduciose rispetto a una classificazione errata. Infatti, nel caso in cui il classificatore assegna una probabilità molto piccola alla classe corretta $\hat{y}_j$, il contributo logaritmico sarà molto elevato. Per avere dei termini di paragone si consideri che il classificatore binario perfetto ha una logloss pari a 0, invece quello casuale pari a $\log(2) \sim 0.693$.

5.8 Costruzione del portafoglio

La strategia di trading utilizzata in questo lavoro, chiamata anche *Top-Bottom*, si basa innanzitutto sulla divisione dei titoli in gruppi usando i valori di probabilità restituiti dalla rete neurale. Più precisamente alla fine di ogni giorno vengono utilizzati i dati disponibili fino a quel momento per fare predizioni al giorno successivo le quali riguardano la probabilità di un aumento della quantità Δr. I valori delle probabilità vengono ordinati e divisi in decili e in questo modo si dividono in decili i titoli. Dopodiché si fissa un budget B e si divide B per la numerosità dei titoli del decimo decile, acquistando (posizione lunga) per ciascuno di essi lo stesso controvalore (portafoglio Top). Analogamente si vende allo scoperto (posizione corta) lo stesso controvalore per tutte le azioni appartenenti al primo decile (portafoglio Bottom). E' importante che i due portafogli abbiano lo stesso controvalore B, in modo tale che la strategia complessiva abbia un importo investito inizialmente nullo e sia neutrale rispetto all'andamento del mercato. L'ammontare di B risulta in tal modo irrilevante, a patto di non modificare le condizioni di mercato e trascurando gli impatti di liquidità.

Si hanno ora tutti gli elementi per valutare la strategia proposta nella sua totalità, ovvero la cointegrazione come cornice teorica per la selezione delle azioni simili, l'indicatore come predittore di rendimenti futuri, la rete neurale come tecnica di previsione e infine il portafoglio Top-Bottom come costruzione delle posizioni. Chiaramente, poiché vengono utilizzati dati con

frequenza giornaliera, i due portafogli vengono ribilanciati ogni giorno in cui il mercato è aperto.

Per analizzare la bontà del contesto teorico, a latere vengono stimate le performance di un'altra strategia di trading, che è solo fittizia e non relativa al pair trading. In particolare si considera di formare dieci portafogli diversi, ciascuno contenente in pari controvalore tutte le azioni di un decile diverso. In questo modo è possibile quantificare la relazione esistente tra i rendimenti dei titoli appartenenti a decili differenti e vedere se e come il rendimento del singolo portafoglio viene influenzato dal decile. L'analisi di questi portafogli serve a verificare l'idea per cui a maggiori probabilità di ottenere un incremento (di ammontare qualsiasi) corrispondano rendimenti effettivamente più alti.

Come per il resto delle procedure utilizzate, vengono considerate solo quelle azioni che risultano cointegrate con almeno un'altra azione nel periodo $[t-3, t)$. Questo campione è mantenuto per tutta la durata dell'anno t in analisi.

5.9 Misure di rendimento

Per valutare le performance della strategia di trading vengono usate due diverse misure: i rendimenti e le alfa dei modelli a fattori di rischio, presentati in Capitolo 4. La principale differenza tra queste due misure risiede nel fatto che la prima è una misura assoluta, infatti quantifica il profitto proveniente dalla strategia. La seconda invece è una misura relativa, quantifica infatti il profitto che la strategia ha portato in eccesso (o in difetto) rispetto al mercato. Oltretutto, come spiegato precedentemente, questa misura può essere aggiustata per vari fattori di rischio, in modo tale che rappresenti, in caso di α positiva e significativa per esempio, un profitto sistematico che va oltre a quello che ci si aspetterebbe per i rischi considerati. In particolare nel corso del lavoro viene utilizzata l'α derivante dal modello di Fama e French a 3 fattori di rischio con aggiunta di short-term reversal e momentum, descritta nell'equazione (4.2).

Per quanto riguarda i rendimenti vengono calcolati i log-rendimenti giornalieri dei portafogli costruiti sui vari decili come media aritmetica dei rendimenti semplici delle azioni appartenenti al decile j-esimo, indicato da dec_j

con $j \in \{1, .., 10\}$, in formule:

$$r_{dec_j} = \log\left(\frac{1}{n}\sum_{i \in dec_j} R_i + 1\right).$$

In questo modo si introduce una piccola approssimazione. Si suppone infatti che le azioni all'interno di un decile possano essere comprate in quantità non intere. Questa approssimazione non ha un peso eccessivo nel caso in cui i titoli vengano acquistati in quantità significative.

Il rendimento del portafoglio Top-Bottom viene calcolato, sfruttando le proprietà dei log-rendimenti, come:

$$r_{TB} = r_{dec_{10}} - r_{dec_1},$$

in quanto la posizione lunga dec_{10} ha lo stesso controvalore della posizione corta dec_1.

Le misure di α invece vengono calcolate attraverso regressione lineare, e la loro significatività è determinata dal valore della statistica t di Newey-West.

In questo capitolo è stato presentato lo schema generale necessario alla costruzione della strategia di trading, sono stati forniti i dettagli utilizzati per l'implementazione degli algoritmi e sono stati introdotti alcuni concetti teorici al fine di comprendere meglio i risultati.

Capitolo 6

Risultati

In questa sezione vengono riportati e commentati i risultati delle metodologie descritte nel capitolo precedente. Inoltre sono introdotte ulteriori tecniche relative allo studio dei fenomeni che guidano i profitti di tale strategia e ne vengono presentati i risultati. Infine è svolta un'analisi di robustezza dei parametri della strategia.

6.1 Cointegrazione

Innanzitutto sono riportati i risultati dell'analisi di cointegrazione. Il numero di azioni cointegrate per anno, cioè il numero di azioni che ogni anno appartiene ad almeno un gruppo di cointegrazione, e il numero mediano della dimensione dei gruppi, sono osservabili in Tabella 6.1. In media 322 azioni,

	Media	2003	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013	2014	2015	2016	2017	2018	2019
n° coint	321.8	428	408	400	317	299	282	318	321	421	367	264	300	304	264	271	238	269
dim gruppi	2.7	4	5	4	2	2	2	2	3	4	3	2	2	3	2	2	2	2

Tabella 6.1: La tabella mostra, per ogni anno di analisi e in media, il numero totale di azioni che appartengono ad almeno un altro gruppo di cointegrazione e la dimensione mediana dei gruppi, intesa come la mediana del numero di azioni che risultano cointegrate con l'azione considerata.

sulle circa 500 dell'indice, appartengono ogni anno ad almeno un gruppo e tipicamente metà dei gruppi hanno meno di 3 azioni, ovvero un titolo risulta cointegrato tipicamente con meno di 3 altri titoli. E' interessante osservare (cfr. Figura 6.1) che il numero di azioni cointegrate è più elevato negli anni di analisi che vanno rispettivamente dal 2003 al 2005 e dal 2010 al 2012.

Considerando che l'analisi è svolta sui dati dei tre anni precedenti, questi

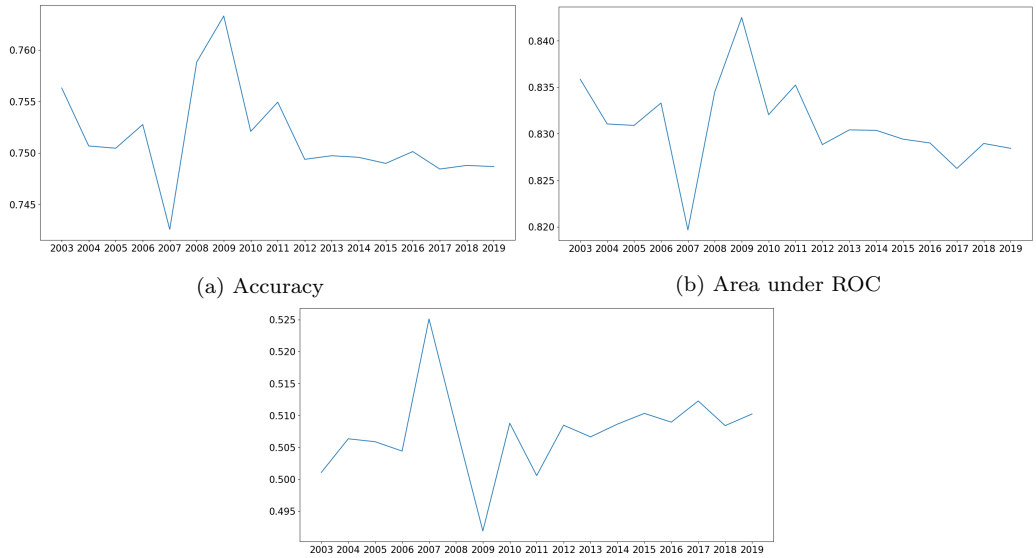
Figura 6.1: Numero di azioni che appartengono ad almeno un gruppo di cointegrazione per ciascun anno di analisi.

periodi corrispondono all'incirca alla crisi relativa alla bolla *dot-com* e alla crisi finanziaria del 2007-2009.

Il tempo computazionale per la stima dei gruppi di cointegrazione è di circa 2 ore per periodo di analisi.

6.2 Capacità predittive

Di seguito vengono discusse le capacità predittive della rete LSTM sui campioni di dati out-of-sample, ovvero i dati compresi nel periodo $[t, t+1)$. Le metriche di precisione sul campione totale sono riportate in Figura 6.2. Si può notare come la rete neurale mostri buoni valori di predizione per tutte e tre le metriche. I valori medi delle misurazioni negli anni equivalgono a 0.752 per l'accuracy, 0.831 per l'area under ROC e 0.507 per la log-loss. Come valori di riferimento, si ricorda che il classificatore casuale ha un valore di accuracy di 0.5, di area under ROC di 0.5 e di log-loss di 0.693. Questi risultati sono abbastanza stabili nel corso degli anni, con qualche oscillazione maggiore nel periodo della crisi finanziaria del 2008, rimanendo comunque in un range di valori contenuto. In particolare le performance prima si deteriorano nel 2007, nell'anno pre-crisi, poiché le caratteristiche dei dati che la rete neurale ha imparato in un periodo di stabilità finanziaria vengono riscontrate meno negli anni di crisi. Poi invece aumentano durante la crisi, quando anche i dati relativi alla crisi sono usati per l'addestramento della rete.



(a) Accuracy

(b) Area under ROC

(c) log-loss

Figura 6.2: Misure di precisione sul campione di dati out-of-sample per ogni anno di analisi rispetto al campione totale di azioni che appartengono ad almeno un gruppo di cointegrazione.

Osservando invece le misure di precisione calcolate sul portafoglio Top-Bottom (Figura 6.3), considerando quindi solo le azioni che ogni giorno appartengono al primo o all'ultimo decile della distribuzione di $P(\Delta r \nearrow)$, e che quindi sono quelle utilizzate nella strategia di trading, si nota che le performance di predizione sono ancora più alte. I valori medi negli anni sono pari a 0.933 per l'accuracy, 0.949 per l'area under ROC e 0.241 per la log-loss. I risultati ottenuti, in particolare quelli relativi al portafoglio Top-Bottom, dimostrano come la struttura teorica della cointegrazione come metodo di selezione delle azioni simili, le reti neurali LSTM come algoritmo di predizione e infine l'indicatore utilizzato si abbinino molto bene l'uno con l'altro con lo scopo di predire se la differenza di prezzo relativa di un'azione rispetto alle sue cointegrate aumenti o diminuisca da un giorno all'altro.

I tempi di calcolo per il training su una singola epoca si aggirano intorno ai 600 secondi utilizzando un un PC normale.

(a) Accuracy Top-Bottom

(b) Area under ROC Top-Bottom

(c) Log-loss Top-Bottom

Figura 6.3: Misure di predizione sul campione di dati out-of-sample per ogni anno di analisi rispetto al portafoglio Top-Bottom.

6.3 Performance della strategia

Di seguito viene valutata la capacità predittiva dell'algoritmo sfruttando una strategia che investe nei portafogli costituiti dalle azioni che giorno per giorno appartengono ai diversi decili della distribuzione di $P(\Delta r \nearrow)$. L'orizzonte temporale di investimento considerato è pari a un giorno. In Tabella 6.2 sono riportati, per ogni decile, i rendimenti giornalieri e le misure di α, calcolate attraverso il modello di Fama e French a 3 fattori con aggiunta di MOM e ST_Rev. Vengono riportati anche i valori della statistica di Newey-West per rendimenti e alfa in modo da poterne analizzare la significatività statistica. In particolare le statistiche per i rendimenti sono calcolate attraverso un modello lineare a sola intercetta: $r_t = a + u_t$.

Dalla tabella si osserva un andamento monotono dei rendimenti e dei valori di alfa lungo i decili, con il decimo decile che ha misure nettamente maggiori rispetto al primo. Ciò indica che l'algoritmo è in grado di predire correttamente il comportamento relativo della maggior parte delle azioni e che una maggiore probabilità di aumento del rendimento relativo è sintomo di un maggiore rendimento. In tabella è presente anche il rendimento della

Decile	1	2	3	4	5	6	7	8	9	10	TB
Rend	-3.5e-5	2.13e-4	3.17e-4	4.61e-4	4.85e-4	5.44e-4	5.84e-4	7.64e-4	7.68e-4	9.21e-4	9.56e-4
	(-0.18)	(1.19)	(1.86)	(2.76)	(2.89)	(3.25)	(3.41)	(4.23)	(4.22)	(4.49)	(7.18)
alpha	-0.0391	-0.015	-0.0054	0.0078	0.0109	0.016	0.0189	0.0375	0.0357	0.0493	0.0873
	(-4.843)	(-2.573)	(-0.979)	(1.514)	(2.252)	(3.522)	(3.9)	(4.794)	(5.980)	(5.606)	(6.464)

Tabella 6.2: La tabella mostra i rendimenti giornalieri medi e le misure di alfa medie dei portafogli costruiti investendo egualmente nelle azioni componenti i vari decili e del portafoglio Top-Bottom. Quest'ultima risulta pari a 9.6 basis point (bps). Le statistiche di Newey-West sono riportate in parentesi.

strategia Top-Bottom, la quale giornalmente assume una posizione corta sul primo decile e una posizione lunga sul decimo decile, costituendo così la strategia di pair trading a costo iniziale nullo. Dai valori riportati si osserva che la performance di questo portafoglio è significativa sia da un punto di vista economico che statistico. E' importante notare che l'indicatore $P(\Delta r \nearrow)$ rappresenta la probabilità di una certa azione di avere rendimenti crescenti nel breve termine rispetto al valore medio dei rendimenti delle azioni con cui risulta cointegrata. Questo significa che l'indicatore non si limita a prevedere i rendimenti di mercato, ma cerca anche quelle azioni con la probabilità maggiore (o minore) di avere un apprezzamento rispetto alle proprie simili, incorporando perfettamente l'idea di *relative pricing* che sta alla base del concetto di pair trading.

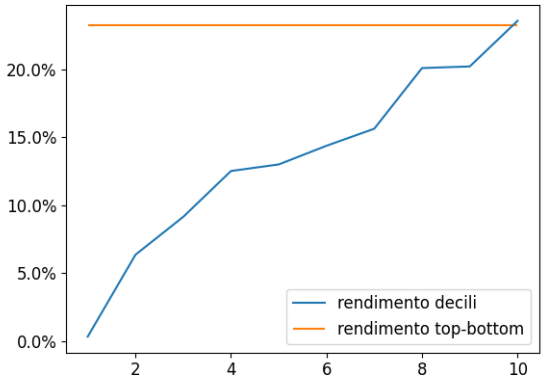


Figura 6.4: Rendimento annualizzato dei portafogli composti investendo egualmente nelle azioni componenti i vari decili e della strategia Top-Bottom.

In Figura 6.4 sono riportati i valori dei rendimenti annualizzati (su 252

giorni) dei portafogli costruiti sui vari decili. Annualizzando i rendimenti e rappresentandoli attraverso un grafico si apprezza meglio l'andamento monotono sui decili e si ha anche un miglior termine di paragone delle performance della strategia Top-Bottom. Questa infatti mostra performance annuali medie del 23.25% nel periodo che va dal 2003 a metà 2019 mentre il mercato americano mostra rendimenti annui tra il 7 e l'8% annuo.

Infine, per avere un'idea del rischio associato ai portafogli costruiti sui vari decili e alla strategia Top-Bottom nella Tabella 6.3 viene riportata la deviazione standard dei rendimenti giornalieri. E' interessante osservare che

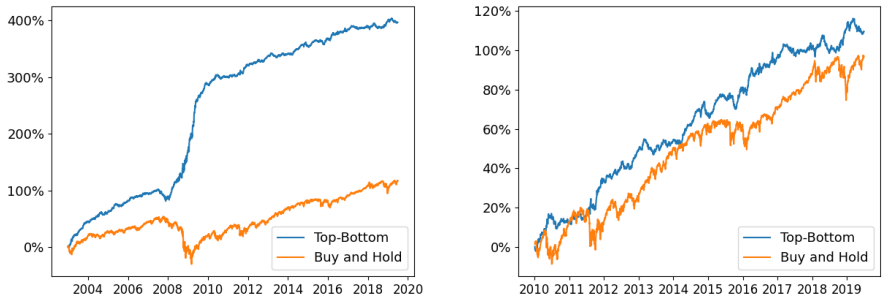
Decile	1	2	3	4	5	6	7	8	9	10	TB
dev std	0.01369	0.01266	0.01210	0.01212	0.01207	0.01203	0.01235	0.01298	0.01322	0.01509	0.00853

Tabella 6.3: La tabella mostra la deviazione standard dei rendimenti dei portafogli costruiti investendo egualmente nelle azioni componenti i vari decili e del portafoglio Top-Bottom.

la variabilità è massima nel primo e nel decimo decile, ovvero a livelli di probabilità più estremi sono associati maggiori rischi. D'altra parte però la deviazione standard del portafoglio Top-Bottom è di gran lunga inferiore rispetto a tutte le altre, dimostrando che la strategia, nonostante scambi i titoli più volatili, è vantaggiosa anche da un punto di vista di rischiosità dell'investimento.

6.4 Ulteriori analisi

In questa sezione vengono indagate ulteriormente le performance della strategia Top-Bottom. Innanzitutto vengono graficati i rendimenti della strategia nel periodo di analisi e confrontati con quelli di una strategia Buy & Hold sull'indice S&P 500. I risultati possono essere osservati in Figura 6.5a. Si osserva che negli anni della crisi finanziaria (2008-09) la strategia Top-Bottom guadagna rendimenti consistenti; la maggior parte dei profitti della strategia si generano in quel periodo. Questi risultati sembrano corroborare alcune opinioni note in letteratura sulle strategie d'investimento di pair trading (o più in generale quelle basate sullo short-term reversal effect), ossia che esse siano nettamente più profittevoli in periodi di instabilità del mercato. Ciò viene spiegato principalmente in base a due motivazioni: da una parte durante i periodi di mercato ribassista si creano opportunità di arbitraggio di valore relativo tra due asset (cfr. Clegg e Krauss (2018) e Jacobs e Weber (2015)), dall'altra nei momenti di elevata volatilità ci sono molti limiti contro



(a) Anni 2003-2019.

(b) Anni 2010-2019.

Figura 6.5: Rendimenti cumulati della strategia Top-Bottom e della stategia Buy and Hold su due diversi periodi di analisi.

le strategie di arbitraggio (*limits to arbitrage*) per cui è più difficile sfruttare questo tipo di strategie (Engelberg et al., 2018) e di conseguenza queste opportunità rimangono nei dati storici. Esempi di limitazioni possono essere elevati costi per le vendite allo scoperto o addirittura un divieto delle stesse.

Un altro aspetto interessante è relativo alle performance della strategia pre- e post-crisi. Si vede chiaramente infatti che dopo la crisi i rendimenti della strategia Top-Bottom sono notevolmente diminuiti e molto più simili a quelli della strategia Buy & Hold (vedi Figura 6.5b). Anche questo aspetto è in accordo con la letteratura per la quale l'ampia diffusione di queste strategie quantitative ne ha diminuito la profittabilità (per esempio Krauss et al. (2017) e Fischer e Krauss (2018)).

Fino a questo punto dell'analisi non sono stati presi in considerazione i costi di transazione, i quali includono principalmente commissioni, costi legati allo short-selling e bid-ask spreads. I costi sono un fattore importante in questo tipo di strategie, caratterizzate da un elevato numero di scambi, i quali portano a un continuo ribilanciamento del portafoglio, e quindi possono deteriorare consistentemente i profitti generati dalla strategia. Al fine di valutare l'impatto che essi hanno sulle performance bisogna innanzitutto notare che le strategie long-short implicano scambiare la stessa azione due volte, quando cioè la divergenza dei prezzi è rilevata e quando si richiude la posizione. Quindi il ribilanciamento del portafoglio è pari al 200% su base giornaliera. Ipotizzando un costo di trading pari a 5 basis point per trade, valore comune in letteratura quando si utilizzano azioni molto liquide quali

quelle dell'indice S&P 500 (si veda per esempio Avellaneda e Lee (2010)), si arriva a un valore di 10 bps per aprire e chiudere la posizione su un asset. Bisogna quindi sottrarre a ogni rendimento ottenuto un valore di 10 bps sia per la posizione lunga che per quella corta. Considerando questo fattore nella valutazione delle performance si arriva, per la strategia Top-Bottom, a un rendimento giornaliero medio negativo (-0.0011) e statisticamente significativo. La statistica t di Newey-West è pari a -7.979. Oltretutto è pratica abbastanza comune quella di considerare costi lievemente più elevati per la vendita allo scoperto. Cosa che in quest'analisi non è stata fatta portando quindi a una stima di redditività maggiore di quella che si otterrebbe in realtà.

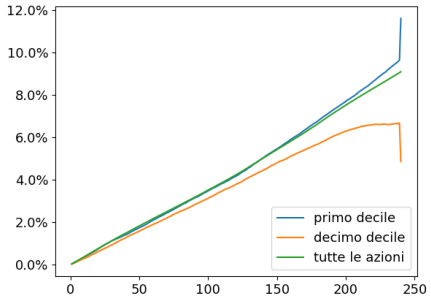
Si vedrà in seguito come questo tipo di analisi può essere resa più accurata analizzando la robustezza dei profitti rispetto ai costi di transazione.

La letteratura ha ampiamente indagato la profittabilità delle strategie basate sulla reversione nel breve termine al netto dei costi di transazione e, nonostante alcuni studi affermino che i profitti non vengono totalmente erosi dai costi, molti di questi giungono alla conclusione per cui i profitti sono nulli o addirittura negativi una volta che i costi vengono presi in considerazione (si veda per esempio Blitz et al. (2013) e Do e Faff (2012)). Quindi la questione per la quale sia o non sia possibile ottenere profitti significativi da strategie di questo tipo rimane quindi aperta e discussa.

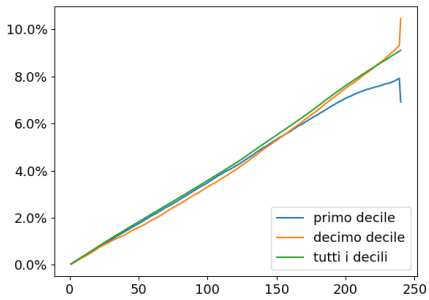
6.5 Studio dei fattori

Gli algoritmi basati sul *machine learning*, e più in particolare le NN, sono comunemente considerati delle *black-box*, cioè dei sistemi descrivibili solo attraverso il loro comportamento visibile. Tipicamente viene descritta la reazione del sistema (output) in seguito a un determinato stimolo in ingresso (input), ma senza comprenderne fino in fondo il funzionamento. Per questo motivo, seguendo l'esempio di Fischer e Krauss (2018), è stata svolta un'analisi sia qualitativa che quantitativa per comprendere più a fondo le caratteristiche comuni delle azioni che la rete neurale seleziona per il trading. Più in dettaglio sono state selezionate le serie storiche dei titoli e dei rispettivi cointegrati che giorno per giorno appartengono al primo o al decimo decile. Da queste sono stati estratti i rendimenti dei 240 istanti temporali prima del giorno di trading, ovvero quelle caratteristiche che la rete riceve in input e attraverso le quali determina se un titolo viene scambiato sul mercato o

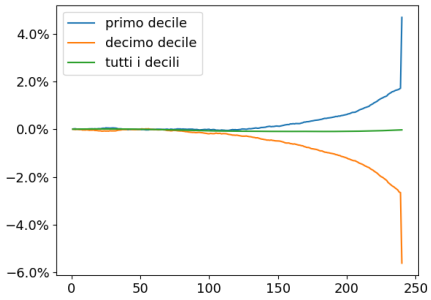
meno. Indicando con t il giorno di trading, per ogni azione appartenente al portafoglio Top o Bottom, vengono considerate tre diverse serie storiche: $\{r_{t-240}, .., r_{t-1}\}$, $\{r^{CG}_{t-240}, .., r^{CG}_{t-1}\}$, e $\{\Delta r_{t-240}, .., \Delta r_{t-1}\}$ rispettivamente la serie dei rendimenti dell'azione considerata, la serie della media dei rendimenti delle azioni cointegrate con quella considerata e infine la differenza tra le due. Fissando uno dei due portafogli, si considerano quindi tutti i titoli che gli appartengono in un giorno di analisi, e da questi vengono estratte le tre serie storiche dei 240 istanti pre-trading. Questa procedura viene ripetuta per tutti i giorni di analisi e viene calcolata la media istante per istante sulla totalità delle successioni estratte indipendentemente dal giorno. In questo modo per il portafoglio fissato si ottengono tre serie storiche da 240 istanti temporali rappresentanti le successioni medie delle tre caratteristiche analizzate. La stessa procedura viene eseguita per l'altro portafoglio e per tutte le azioni in generale. I risultati si osservano in Figura 6.6. Innanzitutto da Figura 6.6a si nota che le azioni appartenenti al portafoglio Top hanno un momentum al di sotto della media nei 240 giorni pre-trading, cioè tendono a sottoperformare se paragonate al campione trasversale, quelle appartenenti al portafoglio Bottom hanno invece un andamento molto simile fino al giorno $t-50$ circa e dopodiché mostrano un momentum lievemente superiore. Questi andamenti si fanno più estremi nel giorno $t-1$ dove i prezzi delle azioni del portafoglio Top crollano, perdendo il 27% di quanto avevano guadagnato in media nei primi 239 giorni. Al contrario i prezzi delle azioni del portafoglio Bottom mostrano un andamento quasi speculare guadagnando un ulteriore 20% negli ultimi giorni prima del trading rispetto a quanto guadagnato prima. Si osservano più o meno gli stessi andamenti, anche se ribaltati in Figura (6.6b): il rendimento medio delle azioni che risultano cointegrate a quelle del primo decile mostrano un momentum al di sotto della media negli ultimi 70 giorni fino ad arrivare a un vero e proprio crollo nell'ultimo; quello relativo al decimo decile ha un momentum molto simile alla media trasversale fino a mostrare un impennata nell'ultimo giorno. Il comportamento combinato delle due serie per ogni portafoglio è infine mostrato in Figura (6.6c). Queste figure mostrano quindi come la caratteristica principale da cui questa strategia di pair trading guadagna sia quella della reversione, per cui le deviazioni dei prezzi sono probabilmente temporanee e si correggeranno riconvergendo nuovamente all'equilibrio. In particolare in questo caso questa reversione di breve termine non solo caratterizza le azioni appartenenti ai decili più estremi ma anche le loro cointegrate, definendo la strategia Top-Bottom implementata come una combinazione di short-term reversal sia delle singole azioni



(a) Singola azione.



(b) Gruppo di cointegrazione.



(c) Differenza dei rendimenti tra la singola azione e il suo gruppo.

Figura 6.6: Serie storiche dei rendimenti cumulati nei 240 giorni precedenti a quello di trading, mediate sui diversi titoli. In ogni sottofigura è presente la serie per il primo decile, il decimo decile e la serie mediata su tutti i decili.

che delle loro simili.

Oltretutto è interessante notare come la rete neurale abbia estratto dai dati, senza nessun suggerimento riguardo a quali caratteristiche cercare, questi andamenti così forti e in comune alla maggior parte delle azioni, dimostrando quindi sia la potenza di questo strumento sia che l'effetto di reversione è fortemente presente nel mercato e può essere usato per predire rendimenti futuri.

L'analisi prosegue quindi valutando l'esposizione del portafoglio Top-Bottom rispetto alle più comuni fonti di rischio sistematico, in modo da poter verificare se effettivamente gli effetti di reversione sono significativi nel determinare i profitti della strategia, e se esistono altri fattori che la influenzano. Viene quindi calibrato il modello a 3 fattori di rischio di Fama e French con

aggiunta dei fattori di short-term reversal e momentum e i principali risultati si possono osservare in Tabella 6.4. Nonostante un coefficiente R^2 molto

Alpha	MKT	SMB	HML	MOM	ST_Rev	R^2	n°obs
0.0873^{***}	0.0672^{**}	0.0042	0.0044	-0.0311	0.2319^{***}	0.07	4151
(6.464)	(3.0)	(0.938)	(0.085)	(-0.972)	(4.889)		

Tabella 6.4: La tabella mostra l'esposizione alle sorgenti di rischio sistematico presenti nel modello lineare a 3 fattori di Fama e French con aggiunta di momentum e short-term reversal. Le statistiche di Newey-West sono riportate in parentesi. $^{***}p < 0.001$,$^{**}p < 0.01$,$^{*}p < 0.05$.

basso, il quale indica che solo una piccolissima parte dei profitti di questa strategia può essere spiegata da queste comuni fonti di rischio, si ritrovano altre caratteristiche attese. Innanzitutto la misura di alfa è positiva e statisticamente significativa (all'1‰), il che indica come questa strategia sia in grado di generare profitti extra in maniera sistematica. Come atteso non c'è una dipendenza rilevante rispetto ai fattori SMB, HML e MOM i quali non sono legati al concetto di reversione. Invece il fattore ST_Rev risulta positivo e statisticamente significativo, mostrando che le dinamiche di reversione dei prezzi a breve termine vengono catturate bene dalla strategia e che questa sorgente di rischio è sfruttata per generare profitti. Infine si nota anche una dipendenza significativa, sebbene in maniera inferiore (1%), dal fattore di mercato, il che contraddice in parte la neutralità della strategia rispetto al mercato.

6.6 Tuning degli iperparametri

Gli iperparametri della rete sono stati presi inizialmente uguali ai valori riportati nell'articolo originale di Flori e Regoli (2021), dopodiché è stato condotto il tuning di questi iperparametri per controllare che non esistessero combinazioni migliori. E' stata utilizzata una procedura di grid search dove gli iperparametri testati erano il numero di nodi nello strato LSTM e il learning rate. I tentativi eseguiti non hanno portato risultati migliori, neanche una struttura del tipo *stacked lstm* ha portato a risultati più convincenti, a fronte però di tempi di calcolo nettamente più elevati. Di conseguenza si è optato per lasciare le impostazioni originali.

6.7 Robustezza della strategia

Vengono qui presentati i risultati di analisi di robustezza rispetto ad alcuni parametri della strategia. Questa sezione serve quindi a dimostrare da una parte che i profitti in assenza di costi di transazione riportati precedentemente non sono frutto del caso e che variando alcuni parametri non si erodono totalmente, dall'altra che i parametri utilizzati costituiscono la combinazione migliore.

6.7.1 Parametri di cointegrazione

Innanzitutto sono stati testati i parametri relativi alla cointegrazione: l'inclusione o meno della costante nelle equazioni (2.6) e (2.7), e la significatività dei test. Nel primo caso è stata inclusa la costante nell'analisi di cointegrazione, come fatto in Huck e Afawubo (2015), e lasciato tutto il resto invariato. Il numero di azioni che risulta cointegrato è maggiore, come si osserva in Figura 6.7, passando da una media di 321.8 a una di 391, anche la grandez-

Figura 6.7: Numero di azioni che appartengono ad almeno un gruppo di cointegrazione ogni anno in presenza o meno della costante.

za mediana dei gruppi aumenta da 2.7 a 3.7. Le misure di precisione delle predizioni della rete rimangono grossomodo invariate, infatti le medie delle metriche equivalgono a 0.752, 0.831 e 0.506 per l'accuracy, l'area under ROC e la log-loss e a 0.933, 0.949 e 0.243 per le metriche del portafoglio Top-Bottom. Infine i rendimenti sui diversi decili e della strategia Top-Bottom si possono osservare in Figura 6.8. Includendo la costante nella relazione di cointegrazione, si osserva che l'andamento monotono dei rendimenti dei diversi decili è molto meno pronunciato e quindi le performance del portafoglio Top-Bottom molto più basse (14.2% annuo). Ciò è dovuto alla maggior

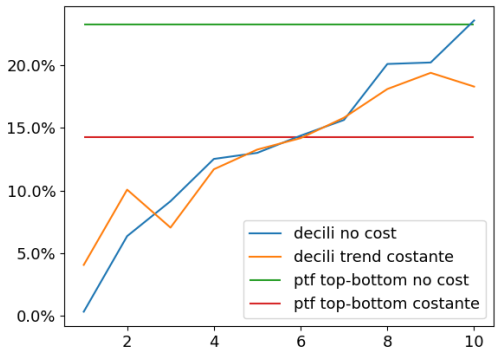


Figura 6.8: Rendimento annualizzato dei portafogli costruiti sui decili e rendimento del portafoglio Top-Bottom in presenza o meno della costante per l'analisi di cointegrazione.

libertà che viene lasciata alle serie storiche attraverso l'inclusione del termine costante che porta a un maggior numero di relazioni di cointegrazione che di conseguenza sono meno rilevanti.

Nel secondo caso invece è stato aumentato il livello di significatività delle relazioni di cointegrazione al 99.9% (escludendo il termine costante) con il fine di verificare se relazioni più significative comportassero una maggiore similitudine tra gli asset e quindi profitti maggiori. Per calcolare i valori critici del test di Johansen è stata seguita la procedura descritta in Sezione 2.5.1 in quanto essi non si trovano tabulati per valori non standard, a differenza di quelli di Engle-Granger.

In questo caso il numero di relazioni di cointegrazione è nettamente minore, come si osserva in Figura 6.9, il numero medio è pari a 111.6 e anche la mediana dei gruppi, ovvero il numero di azioni cointegrate con quella considerata, è più piccola (1.05). Le misure di precisione delle predizioni della rete rimangono circa invariate anche se lievemente inferiori, infatti le medie delle metriche equivalgono a 0.75, 0.829 e 0.511 per l'accuracy, l'area under ROC e la log-loss e a 0.924, 0.944 e 0.262 per le metriche del portafoglio Top-Bottom. Ciò è probabilmente dovuto alla minore numerosità dei dati disponibili per l'addestramento della rete. I rendimenti invece sono rappresentati in Figura 6.10, dove si osserva che in questo caso sono quasi equiparabili anche se lievemente inferiori (21.6% annuo) nel caso di significatività al 99.9%.

Questa analisi di robustezza rispetto ai parametri di cointegrazione quindi dimostra che non introdurre il termine costante e una significatività del

Figura 6.9: Numero di azioni che appartengono ad almeno un gruppo di cointegrazione ogni anno a seconda della significatività dei test.

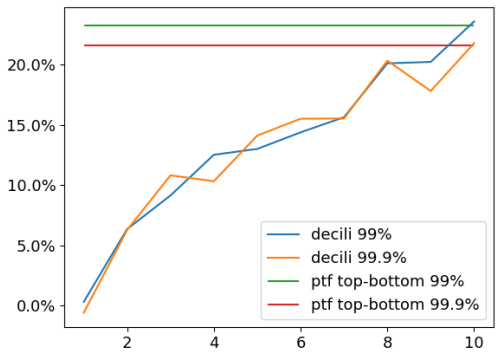


Figura 6.10: Rendimento annualizzato dei portafogli costruiti sui decili e rendimento del portafoglio Top-Bottom a seconda della significatività dei test di cointegrazione.

99% sono le impostazioni migliori per la strategia, in quanto viene raggiunto il miglior compromesso tra il numero di dati disponibili per il training e l'intensità delle relazioni di cointegrazione.

6.7.2 Lunghezza della finestra temporale

In questa sezione viene discusso l'impatto della lunghezza della finestra temporale (o lookback period). Come già affermato questo parametro regola il numero di istanti temporali che la rete utilizza per fare predizioni. Da una parte quindi conviene impostare una finestra temporale molto alta così che la rete abbia a disposizione più informazioni possibili per l'addestramento, dall'altra però una lunghezza molto elevata comporta costi computaziona-

li eccessivi. Oltretutto la Figura 6.6 mostra come i primi istanti temporali hanno un impatto meno importante nelle predizioni effettuate dalla rete.

Sulla base di queste considerazioni la lunghezza del lookback period è stata ridotta ad 80, ovvero circa 4 mesi di dati, e sono state rianalizzate le performance della rete e della strategia. In particolare le medie delle metriche di precisione valgono 0.752, 0.831 e 0.505 per accuracy, area under ROC e log-loss e 0.933, 0.95 e 0.238 nel caso di quelle del portafoglio Top-Bottom, quindi confrontabili con il caso della finestra più lunga. I tempi di calcolo allo stesso tempo si riducono sensibilmente, passando da circa 600 secondi per epoch a circa 150 secondi. I rendimenti si possono osservare in Figura 6.11; l'andamento sui decili è praticamente uguale e anche il rendimento della strategia Top-Bottom è perfettamente paragonabile al precedente, situandosi al 23%. Ciò è indice del fatto che non c'è praticamente nessuna differenza

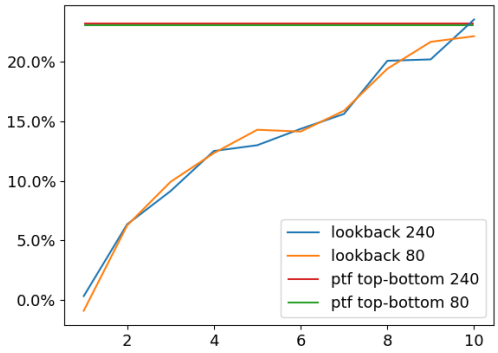


Figura 6.11: Rendimento annualizzato dei portafogli costruiti sui decili e rendimento del portafoglio Top-Bottom a seconda della lunghezza della finestra temporale utilizzata.

nell'utilizzare un lookback period di 240 o 80 istanti temporali, in quanto le caratteristiche che la rete impara a riconoscere sono conservate anche nella finestra più corta. D'altra parte in termini di tempi di calcolo il guadagno è notevole.

6.7.3 Decili

Infine è stata svolta un'analisi di robustezza rispetto all'uso dei decili come criterio di separazione per la creazione dei diversi portafogli. L'idea è quella di verificare se utilizzando una divisione più fitta, e quindi creando portafogli

contenenti meno azioni, l'algoritmo è in grado di selezionare meglio i titoli, e quindi generare profitti maggiori. In Figura 6.12 si osserva che i rendimen-

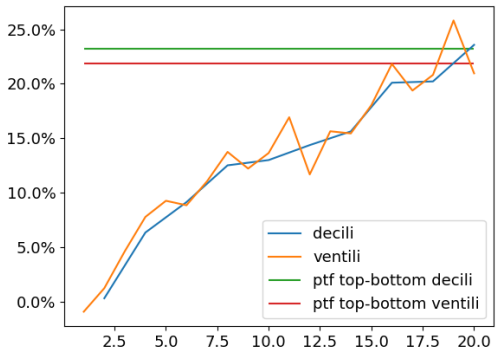


Figura 6.12: Rendimento annualizzato dei portafogli costruiti sui decili e sui ventili e rendimento del portafoglio Top-Bottom.

ti dei portafogli costruiti sui ventili non hanno un andamento strettamente monotono nonostante in media il rendimento cresca all'aumentare del decile considerato. Questo fatto evidenzia come la misura di probabilità $P(\Delta r \nearrow)$, una volta ordinata per valori, sia una buona misura della probabilità di crescita relativa ma con un livello di precisione limitato. Comunque l'algoritmo rimane robusto anche rispetto a una suddivisione più granulare delle azioni, infatti il rendimento complessivo del portafoglio Top-Bottom calcolato sui decili è pari a 21.87% annuo.

In questo capitolo sono stati presentati i principali risultati per l'analisi giornaliera della strategia. In particolare si è visto come essa sia in grado, attraverso ottimi valori in termini di metriche di precisione, di generare rendimenti positivi e statisticamente significativi. In accordo con la gran parte della letteratura si è osservato che la strategia genera la maggior parte dei profitti durante shock di mercato, che la profittabilità però si è erosa nei periodi più recenti, confermando che la profittabilità di questa strategia è collegata al reversal effect. Infine si è mostrato come la strategia sia robusta rispetto alla maggior parte dei parametri utilizzati.

Capitolo 7

Dati ad alta frequenza

Il focus del lavoro si sposta ora su un'analisi della strategia proposta nei capitoli precedenti, nel contesto del trading ad alta frequenza (HFT). La motivazione che guida questa scelta è duplice. Da una parte HFT è una pratica d'investimento sempre più diffusa, si pensi che già nel 2016 questo tipo di strategie raggiungevano volumi di traffico commerciale per azioni in media tra il 10 e il 40% (Aldridge e Krawciw, 2017) e che questi volumi arrivano ormai oltre il 70% del totale in alcuni mercati quotati. Dall'altra grazie all'utilizzo dei dati ad alta frequenza è possibile scaricare i prezzi bid e ask e quindi poter stimare con precisione i costi di trading.

Qui di seguito vengono presentate le modifiche alla strategia di pair trading utilizzata sul campione di dati giornalieri, in modo tale da riadattarla al contesto di HFT, e ne vengono presentati i risultati principali. In seguito vengono presentate delle variazioni della strategia, con lo scopo di massimizzare il profitto, e ne sono illustrati i risultati principali.

7.1 Presentazione dei dati

In questa seconda analisi vengono considerate le azioni che appartengono all'indice S&P 500 nel periodo che va dal 01/03/2021 fino al 20/08/2021, per un intervallo totale di quasi sei mesi. In particolare vengono utilizzati dati con frequenza al minuto nell'intervallo che va dalle 15:31 alle 22:00, orario italiano che corrisponde rispettivamente all'apertura e alla chiusura del mercato statunitense. Tutti i dati relativi ad orari precedenti e successivi vengono scartati in quanto relativi a prezzi di chiusura ritardati oppure a

scambi overnight, e quindi presenti solo per alcune azioni. Più nel dettaglio sono stati scaricati per ogni minuto i prezzi di chiusura (*close*), attraverso i quali è possibile calcolare i rendimenti, i prezzi bid (*denaro*), quelli ask (*lettera*), dai quali si ricavano facilmente i bid-ask spreads, e i volumi di scambio nel minuto.

Il processo di *data cleaning* è stato eseguito in maniera molto simile a quanto fatto per l'analisi giornaliera, le principali differenze riguardano la gestione dei dati mancanti: nel caso di assenza di prezzi di chiusura e presenza di quelli denaro e lettera, eventualità che indica l'assenza di scambi in quel minuto, il prezzo di chiusura è stato riempito con la media tra denaro e lettera. Tutti gli altri casi di prezzi mancanti sono stati riempiti con il valore precedente.

Per quanto riguarda i volumi invece, quelli mancanti sono stati riempiti con un valore nullo. Infine tutti i dati sono stati aggiustati per gli stock-split ma non per i dividendi; la mancanza di questo aggiustamento dovrebbe avere un impatto pressoché nullo, in quanto in termini di rendimenti, ipotizzando un dividendo pagato nei sei mesi di analisi, significa un'osservazione sbagliata su 50.000 circa.

7.2 Descrizione della strategia

Lo schema alla base della strategia di trading rimane praticamente identico a quello utilizzato per l'analisi giornaliera. Di seguito ne vengono riportati i punti chiave e le differenze.

I periodi di training e di testing sono costituiti da 3 e 1 giorno rispettivamente. In questo modo le proporzioni utilizzate nell'analisi precedente rimangono invariate e, in termini di numero di dati utilizzati, la differenza è piccola: si passa infatti da circa 252 dati per anno di analisi a 390 per giorno di analisi. In totale il dataset è costituito da 122 giorni di dati e in questo modo si ottengono 119 giorni in cui l'algoritmo può essere valutato su dati out-of-sample.

L'analisi di cointegrazione e l'indicatore $P(\Delta r \nearrow)$ rimangono praticamente invariati nell'impostazione. Fissato un giorno t, la cointegrazione viene studiata utilizzando i dati dei tre giorni precedenti, l'unica differenza consiste in un maggior numero di lag inclusi nel modello in quanto è aumentata la lunghezza della serie storica. L'indicatore invece viene calcolato sfruttando i dati del giorno successivo con l'obiettivo di predire se la differenza relativa

di rendimenti aumenterà nel minuto seguente. Anche la rete neurale rimane più o meno costruita nello stesso modo, con l'unica differenza che nei dati in ingresso viene inserita anche la serie storica dello spread relativo, calcolato come:

$$Spread_t = \frac{Ask_t - Bid_t}{Close_t} \tag{7.1}$$

che in linea di principio può aggiungere informazioni importanti alla rete neurale sui costi della strategia. Infine anche la costruzione del portafoglio giornaliero, la valutazione delle performance di predizione della rete neurale e di rendimento delle strategie sui decili e su quella Top-Bottom vengono eseguiti allo stesso modo.

7.3 Risultati

I risultati dell'analisi di cointegrazione mostrano che in media 458 azioni ogni giorno appartengono ad almeno un gruppo di cointegrazione e che la mediana della grandezza dei gruppi è di circa 10. Questi risultati sono molto più elevati rispetto al caso giornaliero, praticamente tutte le azioni risultano cointegrate ad almeno un'altra. Ciò indica che alle alte frequenze i titoli dell'indice S&P 500 hanno una tendenza molto più marcata a muoversi insieme. In tempi così brevi è difficile infatti che ci siano ragioni economiche che motivano un diverso comportamento dei prezzi, mentre ci sono importanti spiegazioni di carattere finanziario che spiegano questo risultato. Infatti una massa importante degli scambi viene effettuata da *hedge fund* e altri operatori istituzionali che fanno uso frequente di operazioni su tutto l'indice, influenzando tutti i titoli che lo compongono nella stessa direzione, amplificando quindi l'andamento cointegrato dei prezzi.

La rete neurale continua a dare ottimi risultati in termini di metriche di precisione: i valori medi ottenuti sono pari 0.762 per l'accuracy, 0.844 per AUC e 0.489 per la log-loss, mentre quelli per il portafoglio Top-Bottom sono pari a 0.947, 0.961, e 0.202. Quindi, rispetto all'analisi con i prezzi giornalieri, si ottiene addirittura un miglioramento in termini di precisione, dimostrando che, anche nel contesto dei dati ad alta frequenza, i prezzi relativi futuri sono prevedibili. Questa affermazione viene confermata ulteriormente dai rendimenti della strategia che investe egualmente sui diversi decili e di quella Top-Bottom, i quali sono riportati in Tabella 7.1. Si nota subito l'andamento monotono sui vari decili, che è ancora più pronunciato tra il primo e il secondo

Decile	1	2	3	4	5	6	7	8	9	10	TB
Rendimento	-7.5e-5	-3.69e-5	-2.21e-5	-1.16e-5	-2.23e-6	7.58e-6	1.43e-5	2.56e-5	4.13e-5	8.69e-5	1.62e-4
	(-34.1)	(-17.9)	(-11.3)	(-6.1)	(-1.16)	(3.91)	(7.6)	(13.4)	(20.9)	(39.1)	(70.6)

Tabella 7.1: La tabella mostra i rendimenti medi al minuto dei portafogli composti investendo egualmente nelle azioni componenti i vari decili, e del portafoglio Top-Bottom. Le statistiche di Newey-West sono riportate in parentesi.

decile e tra il nono e il decimo. Questo andamento crescente si concretizza in un rendimento per la strategia Top-Bottom significativo da un punto di vista statistico ed economico, che equivale a un sorprendente rendimento su base giornaliera del 6.33%.

7.3.1 Diversi orizzonti temporali di investimento

Nell'*intraday* trading possono essere presenti trend temporali di lungo termine nei prezzi delle azioni, causati per esempio da acquisti di pacchetti azionari eseguiti mediante ordini successivi. Queste tendenze, se sfruttate adeguatamente, possono avere un'influenza importante nelle performance della strategia. Per questo motivo in questa sezione viene presentato l'impatto di diversi orizzonti temporali d'investimento (*holding period*). In particolare sono stati scelti due intervalli temporali, rispettivamente di 30 e 60 minuti, per i quali detenere obbligatoriamente un titolo dopo averlo acquistato.

Le modifiche da apportare all'algoritmo sono minime e consistono essenzialmente nel ridefinire l'indicatore descritto dall'equazione (5.1) come:

$$y_{t+h} = \begin{cases} 1 & \text{se} \quad \Delta r \geq 0 \\ 0 & \text{se} \quad \Delta r < 0 \end{cases}, \quad \text{con} \quad \Delta r = \Delta r^i_{t+h} - \Delta r^i_t.$$

Così la risposta della rete $\hat{y}_{t+h}$ rappresenta la probabilità che Δr aumenti in un orizzonte temporale di h minuti. I risultati relativi alle metriche di precisione sono presentati nella Tabella 7.2, mentre quelli relativi ai rendimenti nella Tabella 7.3. Nella prima si nota che i due orizzonti temporali di investimento raggiungono più o meno gli stessi risultati in termini di precisione, e allo stesso tempo mostrano un lieve deterioramento delle metriche rispetto al caso con holding period di un minuto. Ciò è sicuramente dovuto alla maggiore difficoltà e alla maggiore incertezza nell'eseguire predizioni su un futuro più lontano rispetto a uno immediatamente prossimo. Per quanto riguarda il rendimento dei portafogli si osserva chiaramente l'andamento monotono,

Misure	Accuracy	AUC	Log-loss	Accuracy TB	AUC TB	Log-loss TB
30 minuti	0.75	0.832	0.503	0.928	0.954	0.242
60 minuti	0.75	0.833	0.5	0.922	0.955	0.250

Tabella 7.2: La tabella mostra le metriche di precisione medie del problema di classificazione per i due diversi orizzonti temporali, rispettivamente di 30 e 60 minuti.

Decile	1	2	3	4	5	6	7	8	9	10	TB
30 min	-3.08e-5	2.414e-5	4.576e-5	6.052e-5	7.861e-5	9.03e-5	1.005e-4	1.119e-4	1.343e-4	1.78e-4	2.089e-4
	(-0.812)	(0.672)	(1.297)	(1.715)	(2.236)	(2.547)	(2.84)	(3.12)	(3.65)	(4.532)	(22.027)
60 min	4.49e-5	9.5e-5	1.18e-4	1.399e-4	1.62e-4	1.64e-4	1.83e-4	1.92e45	2.14e-4	2.55e-4	2.10e-4
	(0.76)	(1.738)	(2.2)	(2.64)	(3.01)	(3.1)	(3.43)	(3.55)	(3.86)	(4.24)	(16.12)

Tabella 7.3: La tabella mostra i rendimenti medi per i due holding period di 30 e 60 minuti dei portafogli costruiti sui decili e di quello Top-Bottom. Le statistiche di Newey-West sono riportate in parentesi.

e le strategie Top-Bottom sono in grado di generare dei rendimenti positivi, altamente significativi da un punto di vista statistico, e maggiori rispetto al caso con holding period di un minuto. Quindi quel poco che viene perso in termine di correttezza delle classificazioni si recupera attraverso maggiori rendimenti poiché in orizzonti temporali di investimento più lunghi i prezzi hanno una maggiore possibilità di muoversi nella direzione desiderata.

7.3.2 Ordinamenti congiunti

Nel corso del lavoro è stato più volte ripetuto che l'indicatore $P(\Delta r \nearrow)$ rappresenta la probabilità che la differenza dei rendimenti cresca in un futuro prossimo, ed è stato mostrato che esso sia perfettamente in grado di predire le dinamiche di *relative pricing*, con ottimi risultati in termini di metriche di precisione. Allo stesso tempo bisogna considerare che la strategia viene costruita in modo da sfruttare la reversione del corso della singola azione rispetto al gruppo, e dell'andamento del gruppo rispetto all'azione. Ma questo secondo movimento relativo viene unicamente sfruttato dalla rete neurale per ottenere una predizione corretta e non dalla strategia di trading per aumentare il profitto, come invece succede nelle strategie di pair trading classiche. Così facendo la strategia rischia di perdere delle sorgenti di guadagno significative rischiando di compromettere la profittabilità finale della stessa.

In linea con un'ampia corrente di letteratura finanziaria viene quindi utilizzata una procedura di ordinamento congiunto delle azioni al fine di massi-

mizzare il profitto ottenibile dal comportamento del singolo titolo rispetto al gruppo. Questa tecnica si basa sull'ordinamento dei titoli, che come prima vengono divisi in gruppi, rispetto alle misure di un indicatore, dopodiché, condizionatamente a questo primo ordinamento, vengono ulteriormente ordinati e divisi in sottogruppi rispetto a un secondo indicatore. Così facendo si ha una misura che indica se la seconda dimensione di ordinamento fornisca o meno informazioni che arricchiscono quelle della prima dimensione, producendo preziosi rendimenti supplementari. In particolare il secondo indicatore utilizzato è quello proposto in Fischer e Krauss (2018), indicato con P_{FK}. Questo viene calcolato attraverso un problema di classificazione binaria che sfrutta una rete LSTM. Il valore target vale 1 se il rendimento all'istante successivo dell'azione considerata è al di sopra della mediana dei rendimenti di tutte le altre azioni all'istante successivo, vale 0 se al di sotto della mediana. L'indicatore proposto cerca quindi di prevedere se il rendimento dell'azione all'istante successivo sia più elevato o più modesto della mediana dei rendimenti dell'insieme di azioni considerate. Per una descrizione più dettagliata della procedura si rimanda all'articolo originale.

In prima approssimazione i rendimenti hanno una media nulla, e quindi questo indicatore predice se il valore dell'azione cresce o decresce. In questo modo, poiché non è possibile sfruttare il profitto proveniente dal movimento relativo del gruppo, si cerca di massimizzare il profitto della strategia comprando e vendendo quelle azioni che probabilmente avranno rispettivamente un rendimento positivo e negativo.

In Tabella 7.4 e 7.5, per completezza, vengono innanzitutto riportati i risultati della strategia di Fischer e Krauss valutata sullo stesso set di dati e allo stesso modo in cui è stata valutata la strategia con holding period di un minuto, ovvero le azioni vengono ordinate in base al valore dell'indicatore e viene venduto il primo decile e comprato il decimo.

Accuracy	AUC	Log-loss	Accuracy TB	AUC TB	Log-loss TB
0.522	0.529	0.6922	0.55	0.553	0.689

Tabella 7.4: La tabella mostra le metriche di precisione medie del problema di classificazione definito in Fischer e Krauss (2018) con un holding period di un minuto.

E' interessante notare che nonostante metriche di precisione quasi equivalenti a quelle di un classificatore casuale, l'indicatore è in grado di produrre un andamento monotono dei rendimenti sui decili e una performance per la strategia Top-Bottom significativa.

Decile	1	2	3	4	5	6	7	8	9	10	TB
Rend	-6.09e-5	-3.115e-5	-1.798e-5	-9.44e-6	-1.28e-6	4.79e-6	1.286e-5	2.11e-5	3.6373e-5	7.3e-5	1.34e-4
	(-26.8)	(-15.2)	(-9.04)	(-4.92)	(-0.67)	(2.5)	(6.87)	(10.7)	(18)	(32.1)	(56.2)

Tabella 7.5: La tabella mostra i rendimenti medi dei portafogli costruiti sui decili e di quello Top-Bottom utilizzando il problema di classificazione definito da Fischer e Krauss (2018). Le statistiche di Newey-West sono riportate in parentesi.

Per quanto riguarda l'approccio di ordinamenti congiunti, nelle Tabelle 7.6 e 7.7 si possono osservare i rendimenti ottenuti dividendo le azioni in quintili rispetto alla prima dimensione di ordinamento e, condizionatamente a questa, in ulteriori quintili rispetto alla seconda dimensione. In questo modo si ottengono $5 \cdot 5$ portafogli lunghi e 5 portafogli Top-Bottom. In particolare nella Tabella 7.6 la prima dimensione di ordinamento è data da $P(\Delta r \nearrow)$ e la seconda da P_{FK}, mentre nella Tabella 7.7 le dimensioni vengono invertite. Gli ordinamenti condizionati nella prima tabella mostrano

	Quintili P_{FK}					
Quintili $P(\Delta r \nearrow)$	1	2	3	4	5	T-B
1	-9.15e-5	-6.13e-5	-5.04e-5	-4.17e-5	-3.56e-6	5.59e-5
	(-35.8)	(-27.1)	(-23)	(-19.2)	(-14.5)	(25.3)
2	-2.8e-5	-2.01e-5	-1.56e-5	-1.35e-5	-6.94e-6	2.11e-5
	(-12.2)	(-9.6)	(-7.8)	(-6.7)	(-3.1)	(10.6)
3	-6.04e-6	-7.83e-7	2.61e-6	5.79e-6	1.2e-5	1.84e-5
	(-2.94)	(-0.37)	(1.31)	(2.8)	(5.27)	(9.4)
4	9.75e-6	1.71e-5	1.80e-5	2.34e-5	3.13e-5	2.16e-5
	(4.45)	(8.6)	(9.03)	(11.2)	(14)	(10.7)
5	3.58e-5	4.77e-5	5.57e-5	7.0e-5	1.11e-4	7.56e-5
	(15.3)	(22.3)	(25.1)	(31.8)	(43.1)	(33.7)
Avg	-1.34e-5	-2.87e-6	1.74e-6	7.35e-6	1.87e-5	3.21e-5
	(-7.8)	(-1.84)	(1.13)	(4.75)	(11)	(24.8)

Tabella 7.6: La tabella mostra i rendimenti medi di 25 portafogli, e quelli delle 5 strategie Top-Bottom nell'ultima colonna, ottenuti ordinando le azioni in quintili rispetto all'indicatore $P(\Delta r \nearrow)$, dopo, condizionatamente a questo ordinamento, ne è eseguito un altro rispetto a P_{FK} e le azioni vengono suddivise in ulteriori quintili. La riga Avg riporta il rendimento medio che si ottiene investendo egualmente in ogni quintile sulla seconda dimensione di ordinamento. Le statistiche di Newey-West sono riportate in parentesi.

che i rendimenti della strategia Top-Bottom, costruita sui quintili di P_{FK} variano tra 1.84e-5 e 7.56e-5 e sono tutti positivi e statisticamente significativi.

Quintili $P(\Delta r \nearrow)$						
Quintili P_{FK}	1	2	3	4	5	T-B
1	-9.9e-5	-5.7e-5	-4.03e-5	-2.68e-5	-6.6e-6	9.23e-5
	(-40.1)	(-25.6)	(-17.9)	(-12.2)	(-2.68)	(44.8)
2	-4.2e-5	-2.2e-5	-1.3e-5	-2.4e-6	1.2e-5	5.43e-5
	(-19)	(-11.1)	(-6.5)	(-1.13)	(5.3)	(30)
3	-2.38e-5	-7.1e-6	2.52e-6	1e-5	2.68e-5	5.06e-5
	(-11)	(-3.4)	(1.24)	(5.3)	(12.5)	(27.5)
4	-1e-5	7.36e-6	1.75e-5	2.5e-5	4.55e-5	5.58e-5
	(-4.6)	(3.54)	(8.8)	(12.3)	(21)	(29.3)
5	1.03e-5	3.26e-5	4.79e-5	6.6e-5	1.16e-4	1.06e-4
	(4.1)	(14.4)	(21.5)	(30.4)	(46.2)	(45.9)
Avg	-2.75e-5	-7.77e-6	2.47e-6	1.22e-5	3.24e-5	5.99e-5
	(-16.3)	(-5)	(1.61)	(7.8)	(19.2)	(48.3)

Tabella 7.7: La tabella mostra i rendimenti medi di 25 portafogli, e quelli delle 5 strategie Top-Bottom nell'ultima colonna, ottenuti ordinando le azioni in quintili rispetto all'indicatore P_{FK}, dopo, condizionatamente a questo ordinamento, ne è eseguito un altro rispetto a $P(\Delta r \nearrow)$ e le azioni vengono suddivise in ulteriori quintili. La riga Avg riporta il rendimento medio che si ottiene investendo egualmente in ogni quintile sulla seconda dimensione di ordinamento. Le statistiche di Newey-West sono riportate in parentesi.

Inoltre il rendimento medio, calcolato investendo equamente nei 5 portafogli Top-Bottom, rappresenta la misura più chiara del fatto che la seconda dimensione di ordinamento va ad integrare la prima fornendo informazioni aggiuntive, e ha un valore pari a 3.21e-5. I risultati di questa procedura di doppi ordinamenti suggeriscono (si osservi il valore medio per colonne) che le azioni appartenenti ai quintili più bassi di P_{FK} tendono a performare peggio di quelle appartenenti ai quintili alti, confermando l'andamento monotono osservato nella Tabella 7.5. Tutto ciò dimostra che l'indicatore proposto in Fischer e Krauss (2018) può essere utilizzato per arricchire le informazioni di $P(\Delta r \nearrow)$ e per costruire strategie d'investimento più redditizie. Allo stesso modo nella Tabella 7.7 si osserva come le informazioni contenute in P_{FK} possono essere arricchite da quelle contenute in $P(\Delta r \nearrow)$. Gli andamenti generali sono gli stessi e i risultati sono addirittura migliori. Si osserva infatti che i 5 portafogli Top-Bottom hanno rendimenti maggiori e più statisticamente significativi, mentre i portafogli medi, costruiti investendo egualmente nei quintili di P_{FK} (si osservi la riga Avg) mostrano ancora un andamento

monotono ma su un range di valori più grande. Infine il rendimento medio della strategia che investe in parti uguali nei 5 portafogli Top-Bottom è pari a 5.99e-5.

Da queste procedure di doppio ordinamento si evince quindi che le informazioni contenute in $P(\Delta r \nearrow)$ sono in grado di arricchire e completare quelle contenute in P_{FK} e viceversa. I due indicatori quindi forniscono informazioni complementari che possono essere utilizzate per costruire strategie più profittevoli migliorandone le performance. L'aspetto interessante dell'indicatore $P(\Delta r \nearrow)$ è quello per cui esso analizza i dati da un diverso punto di vista rispetto ai classici indicatori delle strategie di pair trading, cercando di capire se il divario di prezzi tra un'azione e le cointegrate è in aumento o in diminuzione, e consentendo la creazione di strategie più complesse. Per esempio, quella che assume una posizione lunga sul portafoglio (5,5) e corta su quello (1,1) produce dei rendimenti medi di 2.03e-4 e 2.15e-4, rispettivamente ordinando prima per $P(\Delta r \nearrow)$ e poi per P_{FK} e viceversa. I rendimenti ottenuti sono maggiori del 25.3% e 32.7% rispetto alla strategia Top-Bottom costruita con solo una dimensione di ordinamento e riportata nella Tabella 7.1. La strategia appena proposta è interessante anche da un punto di vista di significato. Infatti il portafoglio (5,5) rappresenta quelle azioni per le quali sia maggiore la probabilità che la differenza di rendimenti tra loro e le rispettive pari aumenti nel futuro, e allo stesso tempo hanno maggiore probabilità di avere un rendimento positivo (superiore alla mediana). Il portafoglio (1,1) ha invece il significato contrario, cioè con differenza in diminuzione e rendimento probabilmente negativo. In questo modo, questa strategia cerca di sfruttare sia i fattori relativi, sottostanti all'idea del pair trading, sia previsioni future sulle performance dell'azione in senso assoluto, senza cioè rapportarle ad altre.

Come ultima osservazione, nella Sezione 7.3.1 si è notato come un orizzonte temporale di investimento più ampio porti a benefici in termini di rendimento. Un'idea interessante è quindi quella di provare a sfruttare i due effetti combinati (doppi ordinamenti e holding period più lunghi) per massimizzare le performance. Nella pratica però l'indicatore P_{FK} non è in grado di fare predizioni a più istanti temporali di distanza. Già impostando un holding period di 30 minuti i risultati in termini di metriche di precisione mostrano performance totalmente equiparabili a quelle di un classificatore casuale, non si osserva nessun andamento monotono sui decili e la strategia Top-Bottom ha rendimenti all'incirca nulli, vanificando quindi questo tipo di analisi. Oltretutto, questa è un'ulteriore dimostrazione della capacità

predittiva dell'indicatore $P(\Delta r \nearrow)$.

7.4 Robustezza rispetto ai costi di transazione

Fino ad adesso è stato dimostrato che questa strategia, nel contesto del trading ad alta frequenza, porta a ottimi risultati sia dal punto di vista di precisione delle predizioni sia da quello dei rendimenti generati considerando solo i prezzi di chiusura. Si pensi per esempio che nella sua versione più semplice produce una performance di oltre il 6% su base giornaliera. L'attenzione del lavoro si sposta quindi sull'effettiva capacità di questa strategia nel generare profitti, una volta che i costi di trading vengono presi in considerazione. Come già affermato i costi in generale consistono in commissioni, bid-ask spreads e costi associati allo short-selling. In questo lavoro vengono considerate azioni molto liquide e di società a grande capitalizzazione. Dal punto di vista di hedge funds e banche d'investimento le commissioni e i costi associati allo short-selling sono trascurabili (Liu et al., 2017) e l'unica fonte di deterioramento del rendimento della strategia è individuata negli spread. Per avere quindi una stima precisa dell'impatto dei costi sono stati utilizzati i prezzi bid e ask per valutare le performance senza quindi utilizzare un valore parametrizzato dei costi, come fatto nella strategia giornaliera.

In particolare il rendimento semplice per una posizione lunga viene calcolato come:

$$R_t = \frac{Bid_t - Ask_{t-h}}{Ask_{t-h}},$$

in questo modo si considera di comprare l'asset al tempo $t-h$ con il prezzo che il mercato offre per venderlo e di rivenderlo al tempo t al prezzo che il mercato è disposto a pagare. Allo stesso modo il rendimento di una posizione corta viene calcolato come:

$$R_t = -\frac{Ask_t - Bid_{t-h}}{Bid_{t-h}}.$$

I risultati per le strategie Top-Bottom su una singola dimensione di ordinamento sono pari a -1.39e-3, -1.36e-3 e -1.35e-3 rispettivamente nei casi di orizzonte temporale di investimento pari a 1, 30, 60 minuti. I risultati dimostrano come la strategia non sia in grado di generare profitti una volta considerati i costi. Le ragioni possono essere molteplici; innanzitutto bisogna

considerare che la rete neurale ha informazioni relative all'andamento dello spread in ingresso, ma nella variabile target, calcolata attraverso i prezzi close, non viene premiato in nessun modo un valore di spread basso oppure un rendimento relativo calcolato con prezzi bid e ask che sia positivo. Anzi se si osserva un grafico (vedi Figura 7.1) dei rendimenti della sola posizione lunga sui vari decili, calcolata con i prezzi bid e ask, si nota che nel primo e nell'ultimo le performance al netto dei costi sono addirittura peggiori come se l'algoritmo assegnasse ai decili più estremi i titoli con spread più elevati. Questo osservazione è confermata dal calcolo degli spread medi nel periodo

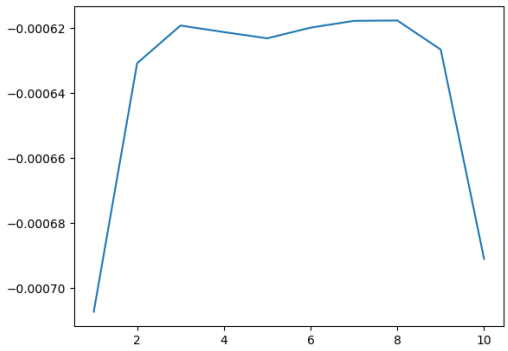


Figura 7.1: Rendimento dei portafogli costruiti sui decili calcolato utilizzando i prezzi bid e ask.

di analisi, si veda formula la (7.1). Infatti tutto il campione di azioni considerate nel periodo analizzato ha uno spread medio pari a 6.44e-4 mentre quelli delle azioni, classificate come appartenenti al primo e al decimo decile, sono pari rispettivamente a 7.05e-4 e a 7e-4. La rete quindi assegna valori di probabilità più estremi alle azioni con differenza tra prezzo denaro e prezzo lettera più ampio. Questa osservazione è coerente con la relazione esistente tra volatilità e spread. Infatti la volatilità solitamente aumenta nei periodi di ribasso e rialzo del mercato. In questi momenti aumentano anche gli spread in quanto dominati dall'incertezza, quando invece la volatilità è bassa e i rischi sono al minimo, lo spread si riduce.

In questo lavoro la rete neurale ha come obiettivo calcolare la probabilità che la differenza di rendimenti tra un'azione e le sue pari aumenti e quindi tra le tante caratteristiche nelle serie storiche ricerca anche volatilità per l'azione considerata, in modo tale da favorire il movimento relativo di rendimenti ri-

spetto alle azioni cointegrate. Dunque a causa della dipendenza tra volatilità e spread, poiché la prima è più elevata nei decili più estremi (come osservato anche nella Tabella 6.3) anche il valore del secondo è più elevato.

7.4.1 Impatto dello spread

Alla luce di ciò è interessante provare a rivalutare le performance della strategia controllando lo spread delle azioni scambiate. Per raggiungere questo scopo viene utilizzata una procedura di doppi ordinamenti usando come indicatori $P(\Delta r \nearrow)$ e lo spread stesso, con l'idea che uno spread relativo basso al tempo $t-1$ sia un buon indicatore di uno spread basso al tempo t. Esattamente come prima quindi le azioni vengono divise in quintili prima rispetto a $P(\Delta r \nearrow)$ e poi, condizionatamente, rispetto allo spread. Di seguito, in Tabella 7.8, vengono riportati solo i risultati relativi ai portafogli medi (Avg) costruiti sulla seconda dimensione di ordinamento, in quanto indice dell'effettiva utilità del controllo degli spread. L'andamento monotono decrescente sui

Quintili Spread	1	2	3	4	5
Avg	-1.71e-4	-2.98e-4	-4.35e-4	-6.18e-4	-1.13e-3
	(-104)	(-116)	(-116)	(-112)	(-111)

Tabella 7.8: La tabella mostra i rendimenti medi di 5 portafogli, ottenuti investendo egualmente in ogni quintile della seconda dimensione di ordinamento, ovvero lo spread. Le statistiche di Newey-West sono riportate in parentesi.

quintili mostra che mantenere basso il valore dello spread porta a rendimenti medi più elevati e che quindi lo spread integra le informazioni racchiuse nel primo indicatore. Le performance interessanti da valutare nel contesto della strategia Top-Bottom sono quelle relative al portafoglio lungo (5,1), cioè con un elevato valore di $P(\Delta r \nearrow)$ e un basso valore di spread, e corto (1,1), cioè con bassa probabilità e basso spread. Il rendimento di questa strategia è pari a -4.31e-4 ordinando prima per $P(\Delta r \nearrow)$ e poi per lo spread. Nel caso di ordinamenti invertiti, comprando quindi il portafoglio (1,5) e vendendo (1,1) i rendimenti sono pari a -4.09e-4. Essi sono sempre negativi ma molto maggiori rispetto a tutti quelli delle altre strategie Top-Bottom. Questo risultato indica quindi che il contenimento degli spread è cruciale al fine di ottenere una strategia profittevole anche se non sufficiente in questo specifico caso.

7.4.2 Indicatore alternativo

Poiché la limitazione degli spread non si è rivelata efficace nella costruzione di una strategia profittevole al netto dei costi di trading viene proposto un indicatore alternativo per il problema di classificazione. Esso mira alla ricerca di quelle azioni che presentano un rendimento crescente rispetto alle cointegrate in maniera maggiore rispetto allo spread in un intervallo temporale di un minuto. Per raggiungere questo obiettivo l'indicatore viene ridefinito in modo tale che Δr in valore assoluto sia maggiore di una soglia S, in formule:

$$y_{t+1} = \begin{cases} 0 & \text{se} \quad \Delta r < -\frac{S}{2} \\ 1 & \text{se} \quad |\Delta r| \leq \frac{S}{2} \\ 2 & \text{se} \quad \Delta r > \frac{S}{2} \end{cases}, \quad \text{con} \quad \Delta r = \Delta r^i_{t+1} - \Delta r^i_t.$$

In questo modo il problema di classificazione viene modificato da uno a due classi a uno a tre classi, dove la classe 0 indica quelle azioni il cui rendimento relativo decresce oltre la soglia, la classe 1 quelle azioni il cui rendimento relativo si muove all'interno di un determinato intervallo, infine la classe 2 quelle azioni il cui rendimento relativo cresce oltre la soglia. Chiaramente la nuova strategia di trading punta a comprare quelle azioni che mostrano un valore elevato di probabilità per la classe 2 e vendere quelle che mostrano un valore elevato per la classe 0. Il significato dell'indicatore rimane lo stesso, ma in questo modo vengono premiate quelle azioni che presentano un movimento relativo nella direzione desiderata maggiore rispetto alla soglia, al fine di favorire una strategia di trading profittevole al netto dei costi.

La soglia viene stimata attraverso lo spread che è la causa principale per cui i rendimenti della strategia sono negativi, ma la stima può avvenire in maniere differenti. Può essere utilizzato un singolo valore per tutte le azioni oppure può essere utilizzato un valore per ogni azione, in modo tale che il risultato raggiunga per ogni azione un buon compromesso tra la liquidità del titolo e il suo profitto futuro. In particolare nell'analisi si è optato per entrambi i metodi. Nel primo caso la soglia è stata calcolata come media degli spread di tutte le azioni cointegrate nel periodo di training, mentre nel secondo caso la soglia è stata calcolata come percentile degli spread dell'azione in considerazione anch'esso nel periodo di training.

Da un punto di vista della strategia rimane tutto praticamente invariato, è però necessario prestare attenzione al fatto che, non essendo più un problema di classificazione binario, in qualche istante temporale la stessa azione

potrebbe appartenere sia al decimo decile della classe 0 che al decimo della classe 2. Questa eventualità, mai avvenuta nel caso pratico, viene gestita attraverso la rimozione della determinata azione da entrambi i portafogli.

Di seguito vengono riportati i risultati di questo nuovo procedimento, iniziando dal caso per cui la soglia è uguale per tutte le azioni all'interno dello stesso blocco di analisi. In Tabella 7.9 sono riportate le metriche di precisione della classificazione. Le metriche si adattano facilmente al caso di classificazione a 3 classi. In particolare l'accuracy indica il numero di classificazioni esatte su quelle totali, la logloss da una misura di penalizzazione delle classificazioni sbagliate, infine AUC valuta l'area sotto la curva ROC e ne calcola la media per le diverse classi. Per una spiegazione dettagliata relativa valutazione della curva ROC nel caso multiclasse si veda Hand e Till (2001) oppure Fawcett (2006). I valori delle metriche di precisione ottenuti

Accuracy	AUC	Log-loss	Accuracy TB	AUC TB	Log-loss TB
0.633	0.725	0.795	0.872	0.785	0.419

Tabella 7.9: La tabella mostra la media delle metriche di precisione del problema di classificazione a 3 classi dove la soglia è calcolata come media degli spread di tutte le azioni nel periodo di training.

non possono essere paragonati a quelli del caso binario, in quanto non solo sono aumentate le classi del problema di classificazione ma anche non si ha più nessuna certezza che la prevalenza sia bilanciata tra le diverse classi. In ogni caso i valori riportati denotano una certa capacità della rete nel fare previsioni corrette. Il rendimento della strategia Top-Bottom però è pari a -1.45e-4, addirittura peggiore rispetto a quello della strategia originale. Ciò è probabilmente dovuto all'inadeguatezza dell'utilizzo di un singolo valore di soglia per tutti i titoli.

Di conseguenza si modifica nuovamente l'indicatore stimando un valore soglia per ogni azione, calcolato come il 50-esimo e il 75-esimo percentile degli spread nell'intervallo temporale di training, in modo tale che il risultato sia più adatto alla selezione dei titoli da negoziare. I risultati in termini di metriche di precisione sono riportati in Tabella 7.10. Si può notare che i valori sono lievemente peggiori rispetto al caso di soglia uguale per ogni azione e che peggiorano di poco passando dal 50° percentile al 75°, a causa della maggiore difficoltà nel predire un aumento grosso rispetto a uno più piccolo. I rendimenti medi delle due strategie sono pari a -1.19e-3 e -1.12e-3. Quindi esistono piccoli miglioramenti sia considerando la soglia sia aumentandone il

	Accuracy	AUC	Log-loss	Accuracy TB	AUC TB	Log-loss TB
50° percentile	0.596	0.688	0.882	0.862	0.782	0.465
75° percentile	0.571	0.688	0.921	0.812	0.771	0.565

Tabella 7.10: La tabella mostra la media delle metriche di precisione del problema di classificazione a 3 classi dove la soglia è calcolata come il 50-esimo e il 75-esimo percentile degli spread per ogni azione nel periodo di training.

valore ma anch'essi non sono sufficienti a rendere la strategia profittevole al netto dei costi.

7.4.3 Persistenza

Infine viene analizzato l'impatto del ribilanciamento del portafoglio. Infatti la dinamica delle performance al netto dei costi è collegata al ricambio di asset contenuti nel portafoglio che può essere consistente e peggiorare fortemente i risultati. Fino ad adesso non è stato preso in considerazione questo fattore, cioè le azioni sono sempre state negoziate in ogni istante. Nel caso in cui uno stesso titolo appartenga al portafoglio Top o a quello Bottom per più istanti consecutivi, è possibile considerare di non comprarlo e rivenderlo (o viceversa) negli istanti intermedi ma invece mantenerlo nel portafoglio, al fine di non pagare due volte i costi associati alle transazioni. In questo modo è possibile sia verificare se c'è un qualche tipo di persistenza temporale delle azioni all'interno dei decili più estremi, sia quantificare l'impatto sui profitti di questo fenomeno.

Le analisi sono state svolte per i 3 diversi intervalli temporali, ovvero 1, 30 e 60 minuti. La persistenza è stata quantificata come la percentuale di volte che un titolo appartenente a uno dei due decili più estremi appartiene allo stesso decile nell'istante successivo. La valutazione del rendimento del portafoglio nel caso di persistenza è invece più complicata e per stimarla sono state introdotte delle approssimazioni. Nel primo minuto del primo giorno un titolo è comprato al prezzo ask (venduto al bid), dopodiché, se la predizione per l'istante successivo prevede che non rimanga nello stesso decile, viene venduto al bid (comprato all'ask) ed è valutato il rendimento del primo istante. Se invece prevede che rimanga nello stesso decile viene venduto (comprato) al prezzo di chiusura, valutato il rendimento, e ricomprato (rivenduto) allo stesso prezzo, e così via. In questo secondo caso quindi non vengono pagati i costi associati agli spread, in quanto per la valutazione intermedia si consi-

dera che venga comprato e venduto allo stesso prezzo. Per fare un esempio se un'azione per due volte consecutive viene assegnata al decile Top il rendimento semplice per il primo scambio è pari a $r(t+1) = \frac{Close(t+1)}{Ask(t)} - 1$ e quello per il secondo scambio è pari a $r(t+2) = \frac{Bid(t+2)}{Close(t+1)} - 1$. I risultati sono riportati in Tabella 7.11. E' chiaro che anche questo fattore, da solo,

	Rendimento	Persistenza
1 minuto	-1.26e-3	8.6%
30 minuti	-1.18e-3	10.9%
60 minuti	-1.18e-3	10.9%

Tabella 7.11: La tabella mostra i rendimenti medi e la percentuale di persistenza delle azioni nei decili più estremi, su i 3 diversi intervalli temporali d'investimento.

non è in grado di rendere la strategia robusta rispetto ai costi di transazione. I rendimenti sono migliorati rispetto a quelli della valutazione "classica", ma rimangono ancora fortemente negativi. E' interessante anche osservare che la persistenza aumenta per orizzonti temporali d'investimento più lunghi ma il suo valore si aggira sempre intorno al 10%, che è il valore casuale. Ciò dimostra anche che non esiste una particolare tendenza del rendimento dei titoli nel continuare a crescere/decrescere relativamente a quelli dei loro cointegrati, ovvero di continuare ad avere le stesse caratteristiche per le quali sono stati classificati nei decili più estremi.

In questo capitolo è stata introdotta la strategia con i dati ad alta frequenza e ne sono stati presentati i risultati. Si è visto come i rendimenti al lordo dei costi siano ottimi già per la strategia più semplice e possono essere ulteriormente migliorati allungando gli orizzonti temporali di investimento oppure utilizzando tecniche di doppi ordinamenti. Dopodiché si sono osservate le performance calcolate utilizzando i prezzi bid e ask, ovvero al netto dei costi. I profitti si deteriorano totalmente e anzi diventano fortemente negativi. Il motivo principale è chiaramente dovuto agli spread, infatti selezionando quelle azioni con uno spread basso i risultati migliorano sensibilmente senza però portare guadagni. Neanche una nuova strategia che mira ad avere rendimenti maggiori dello spread è in grado di portare profitti.

Capitolo 8

Conclusioni

In questo lavoro si approfondisce lo studio proposto in Flori e Regoli (2021). Nel Capitolo 6, utilizzando dati con frequenza giornaliera, si vede come la strategia implementata ritrova molti risultati proposti sia dall'articolo originale che dal pair trading in generale. Essa infatti è in grado di generare rendimenti in eccesso positivi nel periodo temporale analizzato. In particolare le performance aumentano significativamente durante la crisi finanziaria del 2008 e invece negli anni successivi la profittabilità si è ridotta. Le cause verosimili sono il ritorno a condizioni di mercato più normali insieme a una maggiore diffusione di questo tipo di strategie.

Successivamente si analizza l'andamento dei titoli che la rete neurale individua per la formazione dei portafogli e si verifica che quelli negoziati al tempo t hanno subito delle forti variazioni di prezzo. In particolare le posizioni lunghe si riferiscono a titoli che all'istante precedente hanno mostrato una diminuzione di prezzo, al contrario le posizione corte sono su titoli che in $t-1$ hanno visto un apprezzamento. Questo dimostra che la profittabilità del portafoglio è legata a un comportamento di inversione, come ipotizzato nella teoria su cui si basano i metodi di pair trading. In particolare viene individuato un doppio *reversal effect*, sia del titolo considerato, sia del suo gruppo di cointegrazione, il che conferma ancora una volta che questi sbalzi di prezzo sono frequentemente dovuti a delle inefficienze di mercato e non a ragioni economico-finanziarie e possono essere quindi sfruttati per generare dei profitti attraverso una *contrarian strategy*.

La strategia si dimostra anche robusta rispetto alla maggior parte dei parametri presenti nel modello; l'unico che ne influenza in modo significativo i risultati è quello relativo all'inclusione della costante nell'equazione di

cointegrazione. Infatti l'inserimento di questo termine lascia una maggiore libertà al sistema, di conseguenza le azioni vengono ritenute cointegrate più facilmente, ma con una somiglianza più debole e ciò deteriora fortemente le performance. Infine, come già verificato dagli autori, la strategia non è in grado di generare rendimenti sufficientemente elevati da rimanere profittevole rispetto ai costi di transazione.

Nel Capitolo 7, in linea con il crescente interesse e sviluppo dell'*high-frequency trading*, il focus si sposta su dati con frequenza al minuto. La strategia continua a funzionare ottimamente in termini di rendimenti generati al lordo dei costi. In particolare si osserva che i profitti tendono ad aumentare sia allungando il periodo di detenzione sia aggiungendo informazioni riguardanti la probabilità che i titoli comprati/venduti abbiano rendimenti positivi/negativi al giorno successivo.

Anche in questo contesto però la strategia non mantiene la sua profittabilità considerando i costi di transazione che vengono introdotti attraverso l'uso dei prezzi bid e ask. Anzi un'analisi suddivisa sui decili segnala rendimenti netti peggiori proprio per il primo e il decimo, che sono quelli utilizzati per formare i portafogli. Questo perché la rete tende ad assegnare valori di probabilità più estremi ai titoli con uno spread più elevato.

Dopodiché sono osservati due comportamenti fondamentali. Il primo, più ovvio, riguarda il fatto che la selezione di azioni con valori bassi di spread giovi significativamente alle performance, che raggiungono attraverso questa semplice tecnica il loro valore più elevato anche se comunque negativo. La seconda osservazione rivela una diffusa variabilità intertemporale nella composizione dei decili in cui vengono suddivisi i titoli di mercato. I portafogli che si formano ogni minuto sono quindi composti da azioni sempre diverse e questo aumenta le spese di intermediazione e diminuisce la redditività.

Infine vengono presentate le performance di una strategia modificata che, attraverso la ridefinizione del problema di classificazione, premia quelle azioni il cui aumento di rendimento relativo è maggiore dello spread, in modo tale che i profitti delle posizioni siano maggiori rispetto ai costi di intermediazione. Anche in questo caso però il rendimento complessivo non supera i costi associati al trading.

Si arriva quindi alla conclusione per cui le strategie di pair trading, basate su una reversione del valore relativo, sono efficaci in quanto danno profitti lordi positivi ma non sufficienti a coprire i costi di transazione.

La ricerca su questo argomento potrebbe essere ulteriormente approfon-

dita in diverse direzioni, ne vengono qui proposte alcune.

Il punto fondamentale sarebbe capire se esistono, e come identificare modifiche, della strategia che la rendono profittevole al netto dei costi. In particolare sarebbe utile non dover per forza assumere una posizione di mercato in ogni istante temporale ma farlo solo se si ha un qualche grado di certezza che il rendimento generato possa essere superiore al costo della posizione, attraverso l'introduzione di una soglia sul valore di probabilità stimato. Oppure assumere la posizione solo se nel contempo di osserva un divario consistente tra i prezzi di azioni cointegrate, in linea con le strategie di pair trading più classiche. Un'altra modifica potrebbe essere quella di studiare il comportamento relativo rispetto a un singolo titolo, e non a un gruppo, e poi scambiare le azioni in coppie in modo tale da sfruttare il rendimento generato dall'eventuale ritorno alla media in entrambe le direzioni.

Infine sarebbe interessante modificare la strategia attraverso la variazione di altri aspetti come la calibrazione di diverse reti neurali per diversi gruppi di azioni e non una singola per tutte oppure, una volta ottenuti dei profitti effettivi, uno studio più sistematico della grandezza del lookback period da cui dipendono fortemente i tempi di calcolo.

Bibliografia

M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. URL `https://www.tensorflow.org/`. Software available from tensorflow.org.

C. C. Aggarwal. *Neural networks and deep learning*. Springer, 2018.

I. Aldridge. *High-frequency trading: a practical guide to algorithmic strategies and trading systems*, volume 604. John Wiley & Sons, 2013.

I. Aldridge and S. Krawciw. *Real-time risk: What investors should know about FinTech, high-frequency trading, and flash crashes*. John Wiley & Sons, 2017.

Y. Amihud and H. Mendelson. Asset pricing and the bid-ask spread. *Journal of financial Economics*, 17(2):223–249, 1986.

G. S. Atsalakis and K. P. Valavanis. Surveying stock market forecasting techniques–part ii: Soft computing methods. *Expert Systems with applications*, 36(3):5932–5941, 2009.

M. Avellaneda and J.-H. Lee. Statistical arbitrage in the us equities market. *Quantitative Finance*, 10(7):761–782, 2010.

D. Avramov, T. Chordia, and A. Goyal. Liquidity and autocorrelations in individual stock returns. *The Journal of Finance*, 61(5):2365–2394, 2006.

R. W. Banz. The relationship between return and market value of common stocks. *Journal of financial economics*, 9(1):3–18, 1981.

W. Bao, J. Yue, and Y. Rao. A deep learning framework for financial time series using stacked autoencoders and long-short term memory. *PloS one*, 12(7):e0180944, 2017.

E. Barucci and C. Fontana. *Financial markets theory*. Springer, 2003.

S. Basu. Investment performance of common stocks in relation to their price-earnings ratios: A test of the efficient market hypothesis. *The Journal of Finance*, 32(3):663–682, 1977.

R. Baviera and T. S. Baldi. Stop-loss and leverage in optimal statistical arbitrage with an application to energy market. *Energy Economics*, 79: 130–143, 2019.

F. E. Benth, J. S. Benth, and S. Koekebakker. *Stochastic modelling of electricity and related markets*, volume 11. World Scientific, 2008.

W. K. Bertram. Optimal trading strategies for itô diffusion processes. *Physica A: Statistical Mechanics and its Applications*, 388(14):2865–2873, 2009.

C. M. Bishop. Pattern recognition. *Machine learning*, 128(9), 2006.

D. Blitz, J. Huij, S. Lansdorp, and M. Verbeek. Short-term residual reversal. *Journal of Financial Markets*, 16(3):477–504, 2013.

M. M. Carhart. On persistence in mutual fund performance. *The Journal of Finance*, 52(1):57–82, 1997.

Á. Cartea, S. Jaimungal, and J. Penalva. *Algorithmic and high-frequency trading*. Cambridge University Press, 2015.

H. Chen, S. Chen, Z. Chen, and F. Li. Empirical investigation of an equity pairs trading strategy. *Management Science*, 65(1):370–389, 2019.

F. Chollet et al. Keras. `https://keras.io`, 2015.

M. Clegg and C. Krauss. Pairs trading with partial cointegration. *Quantitative Finance*, 18(1):121–138, 2018.

G. Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals and systems*, 2(4):303–314, 1989.

W. F. De Bondt and R. Thaler. Does the stock market overreact? *The Journal of Finance*, 40(3):793–805, 1985.

B. Do and R. Faff. Does simple pairs trading still work? *Financial Analysts Journal*, 66(4):83–95, 2010.

B. Do and R. Faff. Are pairs trading profits robust to trading costs? *Journal of Financial Research*, 35(2):261–287, 2012.

J. L. Elman. Finding structure in time. *Cognitive science*, 14(2):179–211, 1990.

J. E. Engelberg, A. V. Reed, and M. C. Ringgenberg. Short-selling risk. *The Journal of Finance*, 73(2):755–786, 2018.

R. F. Engle and C. W. Granger. Co-integration and error correction: representation, estimation, and testing. *Econometrica: Journal of the Econometric Society*, pages 251–276, 1987.

E. F. Fama. The behavior of stock-market prices. *The Journal of Business*, 38(1):34–105, 1965.

E. F. Fama. Efficient capital markets: a review of theory and empirical work. *The Journal of Finance*, pages 383–417, 1970.

E. F. Fama and K. French. French, 1993, common risk factors in the returns on stocks and bonds. *Journal of financial economics*, 33(1):3–56, 1993.

E. F. Fama and K. R. French. A five-factor asset pricing model. *Journal of financial economics*, 116(1):1–22, 2015.

T. Fawcett. An introduction to roc analysis. *Pattern recognition letters*, 27 (8):861–874, 2006.

T. Fischer and C. Krauss. Deep learning with long short-term memory networks for financial market predictions. *European Journal of Operational Research*, 270(2):654–669, 2018.

A. Flori and D. Regoli. Revealing pairs-trading opportunities with long short-term memory networks. *European Journal of Operational Research*, 295 (2):772–791, 2021.

E. Gatev, W. N. Goetzmann, and K. G. Rouwenhorst. Pairs trading: Performance of a relative-value arbitrage rule. *The Review of Financial Studies*, 19(3):797–827, 2006.

I. Goodfellow, Y. Bengio, and A. Courville. *Deep learning*. MIT press, 2016.

C. W. Granger. *Co-integrated variables and error-correcting models*. PhD thesis, Discussion Paper. Department of Economics, University of California at San Diego, 1983.

A. Haldane. Patience and finance. In *Speech to Oxford China Business Forum, Beijing, Bank of England*, volume 2, 2010.

D. J. Hand and R. J. Till. A simple generalisation of the area under the roc curve for multiple class classification problems. *Machine learning*, 45(2): 171–186, 2001.

J. B. Heaton, N. G. Polson, and J. H. Witte. Deep learning for finance: deep portfolios. *Applied Stochastic Models in Business and Industry*, 33 (1):3–12, 2017.

S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

K. Hornik. Approximation capabilities of multilayer feedforward networks. *Neural networks*, 4(2):251–257, 1991.

K. Hornik, M. Stinchcombe, and H. White. Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5):359–366, 1989.

N. Huck and K. Afawubo. Pairs trading and selection methods: is cointegration superior? *Applied Economics*, 47(6):599–613, 2015.

S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. PMLR, 2015.

H. Jacobs and M. Weber. On the determinants of pairs trading profitability. *Journal of Financial Markets*, 23:75–97, 2015.

N. Jegadeesh. Evidence of predictable behavior of security returns. *The Journal of Finance*, 45(3):881–898, 1990.

N. Jegadeesh and S. Titman. Returns to buying winners and selling losers: Implications for stock market efficiency. *The Journal of Finance*, 48(1): 65–91, 1993.

N. Jegadeesh and S. Titman. Overreaction, delayed reaction, and contrarian profits. *The Review of Financial Studies*, 8(4):973–993, 1995.

S. Johansen. Estimation and hypothesis testing of cointegration vectors in gaussian vector autoregressive models. *Econometrica: Journal of the Econometric Society*, pages 1551–1580, 1991.

S. Johansen. *Likelihood-based inference in cointegrated vector autoregressive models*. Oxford University Press, 1995.

M. Kraus, S. Feuerriegel, and A. Oztekin. Deep learning in business analytics and operations research: Models, applications and managerial implications. *European Journal of Operational Research*, 281(3):628–641, 2020.

C. Krauss. Statistical arbitrage pairs trading strategies: Review and outlook. *Journal of Economic Surveys*, 31(2):513–545, 2017.

C. Krauss, X. A. Do, and N. Huck. Deep neural networks, gradient-boosted trees, random forests: Statistical arbitrage on the s&p 500. *European Journal of Operational Research*, 259(2):689–702, 2017.

B. Liu, L.-B. Chang, and H. Geman. Intraday pairs trading strategies on high frequency data: The case of oil companies. *Quantitative Finance*, 17 (1):87–100, 2017.

H. Lütkepohl. *New introduction to multiple time series analysis*. Springer Science, 2005.

J. G. MacKinnon, A. A. Haug, and L. Michelis. Numerical distribution functions of likelihood ratio tests for cointegration. *Journal of applied Econometrics*, 14(5):563–577, 1999.

W. K. Newey and K. D. West. A simple, positive semi-definite, heteroskedasticity and autocorrelation consistent covariance matrix. *National Bureau of Economic Research*, 55:1–14, 1986.

B. Oksendal. *Stochastic differential equations: an introduction with applications*. Springer Science & Business Media, 2013.

T. pandas development team. pandas-dev/pandas: Pandas, 2020.

J. Patel, S. Shah, P. Thakkar, and K. Kotecha. Predicting stock and stock price index movement using trend deterministic data preparation and machine learning techniques. *Expert systems with applications*, 42(1):259–268, 2015.

H. Rad, R. K. Y. Low, and R. Faff. The profitability of pairs trading strategies: distance, cointegration and copula methods. *Quantitative Finance*, 16(10):1541–1558, 2016.

S. Seabold and J. Perktold. statsmodels: Econometric and statistical modeling with python. In *9th Python in Science Conference*, 2010.

W. F. Sharpe. Capital asset prices: A theory of market equilibrium under conditions of risk. *The Journal of Finance*, 19(3):425–442, 1964.

D. Stattman. Book values and stock returns. *The Chicago MBA: A journal of selected papers*, 4(1):25–45, 1980.

J. H. Stock and M. W. Watson. *Introduction to econometrics*, volume 3. Pearson, 2015.

J. Stübinger and J. Bredthauer. Statistical arbitrage pairs trading with high-frequency data. *International Journal of Economics and Financial Issues*, 7(4):650–662, 2017.

T. Tieleman and G. Hinton. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural networks for machine learning*, 4(2):26–31, 2012.

G. Van Rossum and F. L. Drake. *Python 3 Reference Manual*. CreateSpace, 2009.

G. Vidyamurthy. *Pairs Trading: quantitative methods and analysis*, volume 217. John Wiley & Sons, 2004.