Deep Science Publishing
https://doi.org/10.70593/978-93-49910-91-1_10

Chapter 10: Algorithmic trading and portfolio optimization using deep learning and high-frequency market data

10.1. Introduction

Algorithmic trading has become a dominating form of trading in equity and derivatives markets. A recent analysis of trading data from the U.S. equity market revealed that 70% of all trading activity is now handled by computers using various algorithmic strategies and trading systems. These strategies range from traditional execution, liquidity shaping and smart order routing strategies to more sophisticated high-frequency trading strategies that are often based on statistical arbitrage and market making principles. Autonomous in their performance, proprietary trading firms typically run high-frequency trading strategies on collocated servers. These high-performance trading infrastructure are expensive and involve a highly-technical staff that build custom software intrusions, write often complex algorithmic trading strategies in low level programming languages, and configure powerful computing networks and fast internet connectivity. Not surprisingly, many proprietary trading firms have generated astronomical profits from the skyrocketing growth in trading volume, particularly in equity options (O’Brien, 2025; Pereira, 2025; Quinn, 2025).

The term algorithmic trading is used to describe any trading that is based on using specific instructions derived from pre-defined criteria to execute trades. These criteria are based on benchmarks, average price, and arrival price, for example. There are different approaches to implementing algorithmic trading. The simplest approach is to break a large order into smaller chunks and execute them at different times. This is called the slicing strategy. However, this does not guarantee a successful outcome as market price may drift during the time intervals between executions. Algorithmic trading involves devising a set of rules that will automate trading activity. This is done through the implementation of algorithmic trading systems that allow traders to determine the criteria that will generate buy and sell signals. The systems are also designed to execute trades using the predefined rules for entering and exiting trades.

The research and academic effort has focused on the predictability of financial market excess returns, which had been the research goal of early signal generation studies. AI is directly relevant to these problems, as it is now possible to train AI systems to solve the temporal or spatiotemporal problems posed by these studies directly, simply and effectively. Unlike traditional time series methods, AI systems do not require market data stationarity or normalization, nor do they rely heavily on feature engineering to convert market data into particular domains or components (Rajan, 2025; Santos & Xu, 2025).

Fig 10.1: Algorithmic Trading and Portfolio Optimization

10.2. Literature Review

AI systems have emerged as vital tools for algorithmic trading-related tasks. However, we have not yet encountered such comprehensive coverage of the third revolution of AI — the deep learning renaissance — in algorithmic trading research. This literature review highlights the AI heritage of algorithmic trading, deep learning applied to financial time series forecasting, deep learning applied to order book dynamics, and algorithmic trading with AI equities and assets. The current AI methods lack resilience and transparency: AI-assisted algorithmic trading can result in catastrophic losses for market participants and financial markets. These problems are surmountable thanks to practical constraints embedded in AI systems and encouraged transparency.

Studies in the time series and forecasting literature have long been at the forefront of applying statistical techniques, machine learning, and AI to algorithmic trading design. Financial time series forecasting has a long standing and rich academic literature.

10.3. High-Frequency Market Data

High-frequency market data play a major role when bridging modern deep learning techniques with financial markets. They contain a wealth of information about a growing list of assets, trading decisions and their consequences, as well as the influence of third parties. The ever-increasing supply of various types of high-frequency data in combination with recent innovations in financial theory, artificial intelligence, and the increase in computing power and bandwidths has opened a new era of designing trading systems and optimizing portfolios. In this context, we first review the major types of high-frequency data one encounters in finance. We then describe current data acquisition techniques, motivations, and issues. Coherent models are only as good as their underlying data, and as such, we devote a special section on cleaning, preprocessing, and replenishing high-frequency data.

We naturally focus on high-frequency (intraday) trading and related portfolio strategies. High-frequency data usually refers to trading data on an extreme frequency, such as 1-second or even 1-millisecond frequency. In particular, we cover ask, bid, spread, volume, depth, tick size, and return data. We also touch briefly on other types of data such as news articles, trading ideas, and fundamental signals such as earnings. High-frequency data usually refers to price data on a very high frequency, such as one or more milliseconds. Such trading data consists of ask price, bid price, and volume data. Not denying the importance and added value of data of lower frequencies, such as 1-minute or even 1-hour data, the extreme frequency of high-frequency data provides much richer trading and portfolio signals, with trading decision horizons possibly being even less than one second.

10.3.1. Types of High-Frequency Data

High-frequency market data come in several types, which can be classified into the following primary categories:

Price and Volume Series: This includes the actual price and volume series as a function of time. The price series can be in the form of bids, asks, midprices, or transaction prices. There are also a variety of volume measures, such as transaction volume or volume reshaped using other methods. All of these price and volume series can be recorded at different sampling frequencies and in different terms.

Order Book Data: This is the multilevel order book that records all limit and market orders at each level of the bid-ask spread. There are a variety of sampling frequencies for order book data, as well as various terms. The order book data can also be reshaped into other bid-ask spread measures.

Order Flow Data: This includes the sequence of all market orders and limit orders, as well as other combinations of market and limit orders over time. Orders can take the form of market orders, limit orders, or even cancellation orders. Mostly, however, they are either market orders or limit orders, and they can be referenced in many ways. For example, they can be counted or represented as a sum of values or a change in terms of percentage changes, raw values, or both. They can also be provided in terms of the direction of the transaction, either buyer-initiated or seller-initiated. These transactions can also be aggregated over time, and the resulting by-respective market-side transacting measures can be smoothed over time using various bandwidth smoothing techniques and sampling frequencies.

10.3.2. Data Acquisition Techniques

There are several ways to gather high-frequency market data. Many data vendors specialize in selling high-frequency data that has been collected previously. Some of these vendors may supply historical data for a price, while others provide tick data for a fee. Some high-frequency traders can download historical high-frequency market data at no cost. Other free data sources include various platforms. The drawback to using free data sources, however, is that the downloaded market data may not include all of the required financial instruments, instruments of interest may have missing tick data, or the data could end up being wrong.

Some high-frequency traders may want additional data that vendors do not provide, or that they may find more beneficial in the holistic trading decision-making process. If, for example, proprietary tick data is needed, the trading system must have a way to access financial terms called Application Programming Interfaces (APIs) provided by financial trading firms or clearinghouses, as well as brokerage firms and financial market exchanges. APIs are useful for something other than just collecting tick data. They can be employed by trading software to place a buy or sell limit order into the market or retrieve a triggered stop-loss order. An API is an indirect connection to the Web that transmits the request over the Internet and returns any requested data in an appropriate format.

10.3.3. Data Preprocessing and Cleaning

This chapter focuses on tick data or order-book data and discusses data selection and preparation necessary for algorithmic trading and portfolio strategy design. Using data without necessary cleaning and preprocessing affects backtesting and simulation. The term “garbage-in, garbage-out” applies aptly to backtesting simulations where the data for training a model and the data for testing the model are not chosen correctly. Applying high-frequency data or data from a different market regime will yield misleading results about the model's predictive power and the expected Sharpe ratio of backtested results. It is thus imperative to know in detail how the data has been generated and what adjustments might be necessary.

Preprocessing and cleaning of high-frequency trading data requires domain-specific knowledge of the type of data-generating process and what stylized facts of price data should be expected. For example, minute-frequency close-to-close series would not exhibit the same kind of autocorrelation structure as limit order book data. Preprocessing for the first case would remove duplicates, adjust for corporate actions, fill missing values if necessary, and correct for outliers. In addition to these adjustments, trading phenomena unique to lower frequencies must also be considered. The data analyst would check for price rallies, bubbles, stock splits, and overnight drifts. Stylized facts available for limit order book data would include order imbalance, volatility jumps, queue dynamics, order waiting times, order execution probabilities, and adverse selection effects due to asymmetric information, especially in the market regime and their dependence on order size.

10.4. Deep Learning Techniques

Despite their remarkable success, traditional ML techniques often require addressing some critical issues such as the need for careful feature engineering, choices of scale, sparsity, and normalization, and selection of hyper-parameters. Realizing that the above challenges have to be addressed every time for a new task, the idea of using deep learning techniques to represent a diverse and complex range of variations, many of which may be unknown until now, led to revolutionary advances in various fields by massively improving speed and accuracy, sometimes even by an order of magnitude. Also deserves mention the growing availability of high-performance computational facilities, including GPUs and TPUs, increase in the availability of large datasets, and the emergence of powerful visualized deep learning software frameworks, thus helping practitioners focus more on approaches than implementation. In brief, deep learning enables better performance on varied AI tasks, improving on traditional ML methods and becoming the foundation of increased AI automation. It is believed by some that deep learning may even render ML unnecessary for many applications. Traditional ML techniques and deep learning methods are thus not necessarily seen as mutually exclusive; rather, deep learning acts as a scalable and state-of-the-art approach to both specific and general tasks in many AI applications. The innovation is thus by making hard problems easier again. With the rapidly evolving financial technology ecosystem, deep learning techniques have emerged as an important new class of algorithms in financial applications. The appropriate use of deep learning methods requires careful consideration of the massive amount of financial data, its structure, representation, the tasks in question, theoretical understanding, and domain knowledge, in tandem with building on the success of traditional approaches. Indeed, traditional techniques and deep learning methods are again synergistic.

Fig 10.2: Deep Learning Techniques of Algorithmic Trading

10.4.1. Neural Networks Overview

Deep learning is widely believed to be the most successful branch of machine learning. As part of a larger class of models referred to as neural networks, deep networks use additional levels of successive non-linear transformation, so-called hidden layers, allowing them to ultimately learn complex structure in high-dimensional data provided that sufficient amounts of labeled data is available for supervised training. Neural networks themselves draw inspiration from the structure and organization of the human brain. Each single neuron receives a sum of inputs from nearby neurons, sending a single output forward to potentially many more other neurons. The degree of contribution corresponds to the strength of the connection, a real-valued weight which is then changed during the supervised learning process using gradient-based optimization. The single neuron implements a non-linear transfer function, with the activation function specifying the amount of activation as a function of the total input. Standard choices are the logistic function or variations on the rectifier. The first hidden layer performs thus a local weighted sum and the non-linear activation of its inputs. The output of a single neuron in one layer can thus act as the input to several neurons in the next layer, providing a way to jointly model more complex transformations, especially as it is repeated several times.

Given enough data, restrictions concerning the capacity of modeling hidden patterns and structure in the data can be controlled adaptively by simply varying the number of hidden layers or the number of neurons in the layers, as well as communicating their mutual influence by sharing weights between layers or within layers. The associated learning task of gradient descent on the error with respect to the training data labels has become parallelizable across several CPU and GPU architectures, allowing the practical training of deep architectures for many different learning tasks, from image processing to speech recognition and natural language processing. Hybrid with probabilistic graphical techniques, neural networks are also a true generative model capable of generating realistic samples that mimic the real data distribution.

10.4.2. Recurrent Neural Networks (RNNs)

Recurrent Neural Networks (RNNs) are crucial in deep learning for analyzing sequential data due to their unique capability to maintain context through loop connections, characterizing them as the first deep learning technology incorporating memory. RNNs generate time-dependent additions to the layer-hybrid output of traditional neural networks, allowing them to learn patterns in sequential or time-series data. By using time-dependent results to calculate the network error, the Backpropagation Through Time (BPTT) process allows RNNs to update network weights. The BPTT process calculates the total network error by addition across all timesteps, which unfortunately limits the length of sequential data that can be processed due to memory constraints. Therefore, RNNs are typically used for time-series problems with limited durations.

Unfortunately, simple RNNs suffer from the vanishing or exploding gradient problems, which are amplified for multi-time step sequential data. Long Short-Term Memory (LSTM) and Gated Recurrent Neural Networks (GRUs) are examples of extended RNN architectures that facilitate vastly flexible architectures without the vanishing or exploding gradient problems. LSTMs expand the memory capabilities of the Vanishing Gradient RNN through the addition of gated units that modulate the flow of information into and out of memory cells. GRUs will also do this using gating units, but do not have memory cells which make for a simpler architecture. BPTT can still be used for LSTM and GRU architectures, but the use of distinct gates for controlling memory states also facilitate additional gradient efficiency advantages.

10.4.3. Convolutional Neural Networks (CNNs)

Deep Learning techniques are Machine Learning algorithms which learn the features hierarchically. These algorithms consist of multiple layers: the first layer learns simple patterns and then passes them to the next layer to learn from patterns produced by the previous layer, and so on until the final layer is reached. The last layer uses different signals which are results of features from previous layers, to make a prediction of the target well tuned. These feature learning algorithms are called feature extraction algorithms which extract higher level features from the input data which makes learning the final prediction function more effective.

Deep Learning techniques are inspired by the way the human brain works and its ability to recognize and distinguish high level features from low level features. Different from classical algorithms, these Deep Learning methods can learn features hierarchically from the input for various different data types like images, video, text and audio and combined together have the capacity to surpass the performance of classical models. In recent years, due to the massive amount of available data and the great advances of the capacity of Graphics Processing Units, Deep Learning techniques have become popular in practice and have gained their advocates among practitioners and researchers. The following sections briefly discuss some popular Deep Learning techniques: Convolutional Neural Networks, Recurrent Neural Networks, and Generative Adversarial Networks.

In the field of Machine Learning, CNNs are a category of Neural Networks that have proven very effective in areas such as image recognition and image classification tasks. As a specialized kind of Neural Networks for processing data that has a known grid-like topology, CNNs are particularly used to analyze visual imagery but have also proven useful in other applications that involve spatial data. For these reasons, CNNs are also called Shift Invariant or Space Invariant Networks since the parameters are defined to be shared across space. This is mainly achieved by using filters to perform convolutions. CNNs are similar to ordinary Deep Neural Networks, except that they make use of CNN architecture to exploit the 2D structure of the data.

10.4.4. Generative Adversarial Networks (GANs)

Generative Adversarial Networks (GANs) have since become one of the most successful recent developments in neural networks. A GAN consists of two neural networks – a generator and a discriminator – which are trained simultaneously through a process similar to a minimax two-player game. The generator is trained to produce fake data resembling the training data in order to fool the discriminator into thinking that the data was real. The discriminator is trained to go through batches of real data and fake data produced by the generator, and to output probabilities of how likely the data is real. Once both these networks have been trained, the generator can produce unseen high-dimensional data that looks similar to the training data.

GANs have been widely applied to image, text, and video generation. Since financial data is often viewed as high-dimensional data, GANs can be applied to high-dimensional financial data. As GANs consist of two networks, they require large amounts of training data. Moreover, GANs are notoriously prone to mode collapse where the generator learns to capture only a small fraction of the data distribution. Such limitations make their application to trading systems and portfolio optimization challenging. However, the benefits of GANs such as the ability to augment small financial datasets and to produce unseen high-dimensional financial data have made GANs attractive to research in finance.

In finance, GANs have so far been predominantly used to augment financial datasets, especially images such as stock price charts or candlesticks. GANs have also been used to generate time series data such as stock prices, and to generate trading signals. However, GANs have not been used to model and optimize high-dimensional portfolio return distributions, which is one of the key contributions of this work. Due to the limitations of GANs mentioned earlier, GANs have also not been applied to trading systems or portfolio optimization.

10.5. Algorithmic Trading Strategies

Algorithmic trading is the use of specialized computer programs to enter a series of orders to execute a financial trading strategy without human intervention. It is a very powerful technique allowing trading at much higher speeds with lower costs than human trading while eliminating the risks of human error.

There are many different specific trading strategies that may be incorporated within an algorithmic trading program. Typically, an algorithmic trading program would be structured to use multiple trading strategies simultaneously for a more diversified approach to generate income from market trading during different market environments. A single strategy may also be optimized to be applied to different securities, and more sophisticated algorithms may adapt the strategies dynamically as market conditions change. Empirical evidence suggests substantial profits can be generated using algorithmic trading with relatively low downside risk and such strategy may form the basis of an investment fund. In this chapter, we will detail some of these commonly used algorithmic trading strategies.

Market making involves placing bids and offers in a market to profit from the bid-offer spread. Market making normally involves taking and providing liquidity into the market by placing bids and offers at the consensus market prices. A market maker would normally make money over time from the bid-offer spread. Algorithmic trading building blocks for market making normally consist of a price updating engine and a bid-offer spread control module. The price updating engine ensures the bid and offer prices are reflective of the latest market prices based on trade and market data while the bid and offer spread control module sets the bid-offer spread depending on current market conditions.

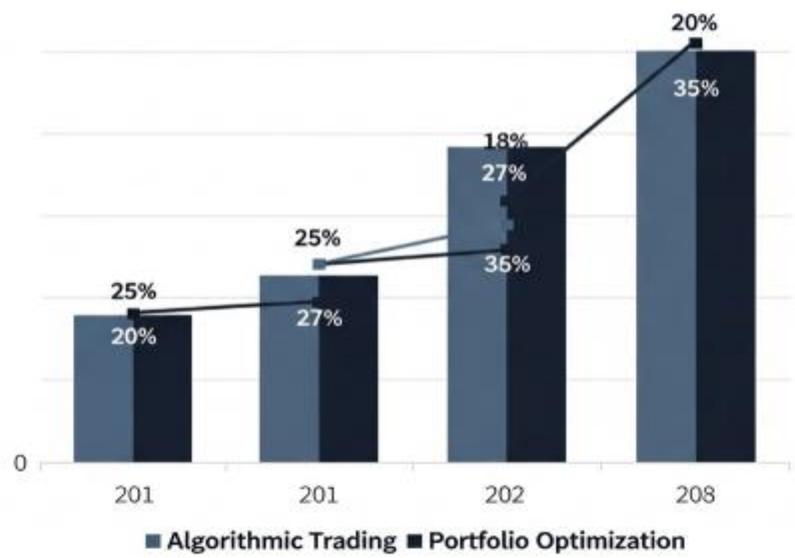

Fig 10.3: Deep Learning and High-Frequency Market Data

Arbitrage strategies seek to profit from mispricing between closely related products or relationships by taking offsetting positions in these products or using products or related securities with defined price or payout relationships. Typically, arbitrage strategies have specific entry and exit signals and allocation of capital across the products being arbitraged.

The example above is a simple case of statistical arbitrage where we simply took advantage of the pricing discrepancy of a single asset across two exchanges. However, there are various sophisticated methods of arbitrage. Below are a couple of commonly employed arbitrage trading strategies. The first one is called cash and carry arbitrage. In this arbitrage, two positions are taken, one in the cash market and the other in the derivative market, i.e., futures market or options market. The trader buys the underlying asset from the cash market and simultaneously sells the derivative on the same underlying security.

10.5.1. Market Making

Market making consists of supplying liquidity to the limit order book by placing limit orders on both sides of the book. Market makers capture the bid-ask spread of an asset or security selling at the ask price and subsequently buying to close at the bid price. Market makers have private information about their clients' demand and supply, and they engage in a vast amount of transactions at little profit with the aim of profiting from illiquid markets. Retail and institutional investors are the main consumers of this service. The longer the distance between the bid and ask price the higher the cost to consumers. Yet market making is not risk-free, especially in the eventuality of an adverse selection cost when market makers trade against informed traders.

10.5.2. Arbitrage

Arbitrage is frequently considered the most basic form of trading. An arbitrageur is said to exploit price differences in a specific financial instrument in different markets. Of course, in an efficient market, such pricing discrepancies only exist for a short time. Let us consider an example to show the concept of arbitrage.

Suppose that a stock is traded on two exchanges. The stock is currently trading at $100 in one exchange, and $103 in another. The trader simultaneously buys the stock at one exchange for $100 and sells it at the other for $103, pocketing a $3 profit. However, as soon as the trader engages in this transaction, the price of the stock on the first exchange will increase (due to the buying pressure), and the price of the stock on the second exchange will decrease (due to selling pressure). Eventually, the prices on the two exchanges will converge to a single price, eliminating the possibility of profits from the trade. If such price discrepancies existed for a longer time, traders would exploit them and make significant profits, thus increasing market efficiency since they would have provided liquidity by buying the stock in one exchange and selling it in another.

10.5.3. Trend Following

Trend Following strategies buy an asset after a price increase, and short-sell the asset after a price decrease, with the expectation these directions would continue. The goal of these strategies is to find price movements that can persist over a certain period of time, with the hope that they persist over the order execution time in the case of high-frequency trading. Trend Following is a less contrarian strategy than Mean Reversion and is actually a more liked strategy by trend followers, for the simple fact that it would reduce their lifetime risk of investing and help them realize their annualized returns. CTAs and hedge funds have employed diversified Trend Following for systematic trading for decades, holding long and short positions on equity indexes, interest rates, foreign exchange, and commodity futures. Such trading strategies, for the most part, have been developed using non-machine learning tools, model-free reinforcement learning, and deep reinforcement learning to develop the trading strategy later. Trend Following has a very large literature with dozens of models for volatility and risk controls with different types of orders; price, spread, market, limit, stop; and different objectives and different markets. Although work has been done using Long Short-Term Memory neural networks, especially when using sentiment variables and recommendation systems, LSTMs have been used mainly for sequential prediction of asset prices. Unlike LSTMs, which are trained for sequential asset price prediction on daily data, the high-frequency trading strategy developed in this research will use day-of-the-week and time-of-the-trading-day periodic functions incorporated in the architecture of the model. This will enable the proposed model to be trained directly to optimize the order execution between the two critical trading times of the trading day; the market open and market close.

10.5.4. Mean Reversion

Mean reversion is one of the oldest algorithmic trading strategies. The rationale is that, in the short term, prices fluctuate and sometimes move far from their long-term moving averages. After trading for a time just outside their normal price range, stocks become an overbought and oversold situation, retracing back to their average price level. The determination of the mean-reverting zone proximity is calculated via a short-term moving average or another indicator, such as Bollinger Bands or the relative strength index.

Mean Reversion Strategy Algorithm

The algorithm of the trading mean reversion strategy is the following:

Input Kernel Function, K(s,t), and stock price, S(t)..

1. Step 1. Prepare Data

The training data are D8(t0) containing D8(t0 − 1), D8(t0 − 2), D8(t0 − 3), ⋯ and D8(t0 + 1), D8(t0 + 2), D8(t0 + 3), ⋯, with Qt of length Q as inputs and K8, M(1), and D8(t0 + 8) as output.

The tested data contain D8(t1), D8(t1 − 1), D8(t1 − 2), D8(t1 − 3), ⋯ across days are loaded with, K8, M(2),(4), and D8(t1 − 8) as output.

2. Step 2. Submit Task

A kernel model is generated from the training data, and the mean reversion trading model is submitted along with the task. The task is to generate the return prediction for day t1... to minimize error.

3. Step 3. Execute Task

At the end of the day, the mean reverting signal is generated for the distance from the training data center. The input mode for performing the forecasting exercise on test data consists of K8, M(1), D8(D8(t0 − 8)). During the day every half hour, the kernel model is executed. A binary trading strategy is implemented.

The mean reversion method consists of selling overbought shares and purchasing oversold ones as part of a trading algorithm. Although many refer to this rationale, it is very challenging to define either a single signal throughout the day or a binary signal that proceeds to the repeated distances across the day.

10.6. Portfolio Optimization Techniques

Constructing a portfolio out of one or more assets is an essential step after predicting market trends. Portfolio optimization is a critical task in asset trading and allocation management since a portfolio with a maximum return for a minimum acceptable risk and given constraints is desired. Optimization of a portfolio is normally done by employing historical data of assets and indexes to find suitable correlation values among the assets. Although associated with the Capital Assets Pricing Model, the optimal portfolio is more often obtained by utilizing the Modern Portfolio Theory. In MPT, a static asset allocation consisting of prescribed proportions of assets in a portfolio over a long-term horizon is often utilized based on uniformly weighted historical sample data.

Thereafter, this allocation is reevaluated after a certain time interval in order to provide the best risk-return characteristics. In this approach, historical return is used to compute the optimal portfolio. Generally, these optimal portfolios may be optimal only at the historical time period in which they were computed. In fact, financial datasets of stock prices are characterized by chaos and nonlinearity, for which traditional static models of forecasting are not supposed to work. As a result, the actual returns may often deviate considerably from the forecast return values.

There are several techniques available to optimize a portfolio. We discuss below three of the most popular techniques.

10.6.1. Modern Portfolio Theory

Introduced and elaborated by the Nobel laureate Harry Markowitz, the Modern Portfolio Theory (MPT) addresses key questions related to the capital allocation problem, central to the theory of finance. Suppose our portfolio consists of n risky assets, denoted by $(S^1, S^2, \ldots, S^n)$, and our portfolio weight vector is given by $w = (w^1, w^2, \ldots, w^n)$. For different investments at different maturity dates, we need to take into consideration the evolution of the value of the portfolio over time. Thus, we only consider the returns instead of the (absolute) levels, which is consistent with most economic arguments. We are also assuming for now that the risk-free interest rate is constant over time. The return on our portfolio then is a linear combination of the returns on the individual assets, thus given by the equation:

$$r^p = w^T r + r^F,$$

where r denotes the vector of returns on the investment assets, r^F is the risk-free return, and r^p is the return on our portfolio. The risk associated with the investment return comes from the risks associated with the investment in each security; that is, our portfolio volatility depends upon the volatilities associated with the individual securities as well as their covariances.

We first analyze the capital allocation problem with two risky assets, and later generalize it to the case with more than two risky assets. Suppose that the portfolio consists of two risky securities S1 and S^2, which have returns given by the equations:

$$r^1 = \mu^1 + \varepsilon^1(t), \text{ and } r^2 = \mu^1 + \varepsilon^2(t),$$

where the return vectors given by ε^1 and ε^2 are both normal distributions with mean zero and ≥ 0 with variances σ^2_1 and σ^2_2 and correlation, respectively. The return volatility associated with portfolio investment during the horizon is given by:

$$\sigma^2 p = w^2_1 \sigma^2_1 + w^2_2 \sigma^2_2 + 2 w_1 w_2 \rho \sigma_1 \sigma_2,$$

where w_1 + w_2 = 1. Both portfolio returns and volatilities have the same structural meaning as the individual security returns and volatilities, but enlarged from the one-dimensional case to the two-dimensional (risky assets) case. Thus, the investment portfolio is no longer given by naïve diversification. Instead, it follows the theory of finance.

10.6.2. Black-Litterman Model

The Black-Litterman model is an extension of the original CAPM model and has been widely recognized in finance literature as a rational method for creating a portfolio of assets. In this approach, an investor can generate his/her expectations on asset returns based on both the risk of the asset and an equilibrium assumed to be defined by the market. The Black-Litterman model has several advantages: it allows the inclusion of an investor's opinions on specific assets; it reflects the investor's confidence in the estimations; it incorporates a method of combining different estimation sources; and it overcomes the problem that a small error of estimation could make huge variance during the optimization process. The realistic problems implied in this model could be generated and properly solved using the tools of deep learning, including input-output operation and attention.

The Black-Litterman model is typically constructed as a normal distribution. Aside from scratches and uncertainty, its distribution could be classified as irrational as it fixes the expected return of a specific portfolio to the risk-budgeting portfolio with confidence, which is typically not the case for an unreasonable investor who creates a portfolio without inward consideration of the world market, his/her past investment experiences, and the objective market state including unknown structural breaks. However, the structure of the Black-Litterman model indicates that one could construct distorted estimations of the expected return by adjusting the input variables and applying deep learning. Using these advantages over a wide asset universe, we propose a recursive approach to decorated estimations of the expected return, extract a reasonable portfolio without unreasonable inputs, and generate a probabilistic rather than a deterministic portfolio using random guessing.

10.6.3. Risk Parity Approach

Another approach to portfolio construction has lately gained popularity, appealing to a group of portfolio managers and hedge funds. This group believes that market capitalizations are not a good measure of risk and the overall investment weighting should take into consideration the risk contribution of each asset in a portfolio. The approach is called risk parity, which is a relatively new concept that is the radical opposite of the capital asset pricing model and modern portfolio theory in that it would allocate capital to each asset class based on risk, not correlation. The basic concept contends that risk and its contribution to a portfolio's total risk, rather than expected return, is the most important factor to consider when devising which assets to put in a portfolio and how to weight them. Risk parity aims to allocate a portfolio's risk equally among the various asset classes contained in the portfolio. The conceptual foundation of risk parity is relatively simple. If you create a portfolio where both equity and bond positions contribute equally to overall portfolio risk, the portfolio should perform better than one that is long equities and short bonds, since equities are, on a historical basis, a much riskier asset class than bonds.

Risk finding has become much easier for portfolio managers because computer technology now allows for simultaneous monitoring of the risk factor exposures of hundreds of portfolios. Generating risk concentrations, such as geographic exposure, sector exposure, currency exposure, and liquidity exposure, has become a common practice in the hedge fund industry. As hedge fund styles have become more defined and more transparent, the risk separation process permits investors to constrict risk and protect against possible blow-ups from their hedge fund investments.

10.7. Conclusion

In this book, we introduced us to how to find alpha in financial markets from the lens of deep learning. We focused on using deep learning methods on high-frequency financial data due to its readily available information and the profit potential coming from its low market price impact. The result of applying deep learning to the problem of predicting mid-price change was streamed spatially on a daily basis and temporally on a tick-by-tick basis and showed a very high precision and a very low maximum drawdown. Based on it, we built an execution strategy based on only market orders and back-tested it on actual high frequency data over a year of S&P 500 E-Mini futures. We showed it beat the transaction costs towards which we built the strategy. And because it was built on raw market data, we presented its profitability as 0.10% of the market depth. The strategy was also very resilient to the widening of the market depth. We also learnt a lot about deep learning using high-frequency financial data, such as batch generations and hyperparameter tuning. Our learnings enable us to build deep learning models that had a streaming data mode and that could generalize to unseen data.

10.7.1. Future Trends

Advances in all aspects of machine learning methods are leading to rapid adoption by an increasing number of the retail market actors, as well as by other participants in generally every sector of the economy. The exponential growth of available datasets and the democratization of the tools necessary to analyze them are pushing the boundaries of formerly posited limits for machine learning methods. The future is, therefore, very exciting but also ambiguous: will, e.g., reinforcement learning-based approaches shift the market dynamics, or on the contrary, the machine learning models will prevail, allocating portfolios in such a way that the reinforcement strategies are prevented from being successful?

In light of the fast pace of innovation, what conclusions can be drawn from the analysis shown in this study? Generally, we suggest that a wide array of different approaches should be tried and allocated to different market conditions, relying in this way on their inherent strength and expertise. There is no single best approach for machine learning-based portfolio optimization and the generic question of model selection is not trivial. The underlying multi-objective decision process is wide, as decisions considering transaction costs and bid-ask spreads, risk preferences while optimizing on return or optimizing on the Sharpe ratio, or richness of tradeable strategies while optimizing on the turnover or optimizing on the portfolio cardinality uniquely influence performances. These types of decisions depend not only on a market actor's profile but also on the prevailing market conditions. A broad basket of strategies, models, and configurations focused on specific market scenarios is very helpful in effectively exploiting the rich sources of information provided in high-frequency datasets..

References

O'Brien, K. (2025). The role of AI in customer onboarding and KYC. In The New Frontiers of Financial Services (pp. 282–295).

Pereira, L. J. (2025). AI-powered decision-making in investment banking. In The New Frontiers of Financial Services (pp. 296–310).

Quinn, T. (2025). Transforming insurance through AI and automation. In The New Frontiers of Financial Services (pp. 311–326).

Rajan, A. (2025). AI governance and stakeholder trust in financial systems. In The New Frontiers of Financial Services (pp. 327–340).

Santos, G., & Xu, L. (2025). Personal financial management with conversational AI. In The New Frontiers of Financial Services (pp. 341–358).