IMES DISCUSSION PAPER SERIES

A Review of New Developments in Finance with Deep Learning: Deep Hedging and Deep Calibration

Yuji Shinozaki

Discussion Paper No. 2024-E-2

IMES

INSTITUTE FOR MONETARY AND ECONOMIC STUDIES

BANK OF JAPAN

2-1-1 NIHONBASHI-HONGOKUCHO
CHUO-KU, TOKYO 103-8660
JAPAN

You can download this and other papers at the IMES Web site:

https://www.imes.boj.or.jp

IMES Discussion Paper Series 2024-E-2
April 2024

A Review of New Developments in Finance with Deep Learning: Deep Hedging and Deep Calibration

Yuji Shinozaki*

Abstract

The application of machine learning to the field of finance has recently become the subject of active discussions. In particular, the deep learning is expected to significantly advance the techniques of hedging and calibration. As these two techniques play a central role in financial engineering and mathematical finance, the application to them attracts attentions of both practitioners and researchers. *Deep hedging*, which applies deep learning to hedging, is expected to make it possible to analyze how factors such as transaction costs affect hedging strategies. Since the impact of these factors was difficult to be assessed quantitatively due to the computational costs, deep hedging opens possibilities not only for refining and automating hedging operations of derivatives but also for broader applications in risk management. *Deep calibration*, which applies deep learning to calibration, is expected to make the parameter optimization calculation, which is an essential procedure in derivative pricing and risk management, faster and more stable. This paper provides an overview of the existing literature and suggests future research directions from both practical and academic perspectives. Specifically, the paper shows the implications of deep learning to existing theoretical frameworks and practical motivations in finance and identifies potential future developments that deep learning can bring about and the practical challenges.

Keywords: Financial engineering; Mathematical finance; Derivatives; Hedging; Calibration; Numerical optimization

JEL classification: C63, G12, G13

* Deputy Director, Institute for Monetary and Economic Studies, Bank of Japan (currently, Associate Professor, Musashino University, E-mail:y-shino@musashino-u.ac.jp)

The author appreciates comments from and discussions with Yasufumi Osajima (SMBC Nikko Securities), Yoshihiro Tawada (MUFG Securities EMEA), Toshiyuki Nakayama (MUFG Bank), and Syoiti Ninomiya (Tokyo Institute of Technology). The author also thanks FORTE Science Communications (https://www.forte-science.co.jp/) for English language editing. The views expressed in this paper are those of the author and do not necessarily reflect the official views of the Bank of Japan.

Contents

1 Introduction

In recent years, deep learning has been applied in a variety of fields, including image and natural language processing. The field of finance is no exception. Indeed, many innovative methods using deep learning have been proposed to solve problems in financial engineering and mathematical finance. In particular, research on *hedging* and *calibration*, which are crucial in practical applications, has garnered attention both theoretically and practically. Given this trend, this paper summarizes the application of deep learning in the field of finance, with a focus on research related to hedging and calibration. It also discusses possible future developments deep learning might bring to the field of finance.

One of the main objectives of finance theory is to guide decision-making of market participants under uncertainty such as price fluctuation risks of financial assets. Among the prominent concerns in the field of finance is how to price derivatives (financial products whose payoffs depend on the future prices of underlying assets such as stocks or foreign exchange) and manage their risk. Financial engineering and mathematical finance, which mainly focus on these issues, have developed under strong influence of practical applicability. In particular, the problem of hedging, which addresses how to mitigate the future price fluctuation risks of financial assets held, and calibration, which is the process of adjusting model parameters to ensure consistency between observed market prices and theoretical model prices, are central research themes in financial engineering and mathematical finance.

Deep learning is expected to significantly advance techniques related to hedging and calibration. First, with regard to hedging, deep learning has the potential to derive hedging strategies that take into account the impact of transaction costs and other actual market frictions. While numerous theoretical studies have been conducted under the traditional framework of risk-neutral valuation, in practice there are various constraints that prevent the full implementation of hedging strategies that are derived theoretically. Recent approaches using deep learning for hedging aim to overcome these constraints, potentially leading to more refined hedging strategies and to the automation of hedging transactions. Second, with regard to calibration, it has been pointed out that deep learning has the potential to significantly accelerate the process of parameter adjustments, which would lead to faster and more stable valuation of derivatives. In this paper, we summarize such prospects in the field of finance and clarify the implications of deep learning to existing theoretical frameworks and practical motivations.

The reminder of the paper is organized as follows: First, in section 2, we summarize the concepts of financial engineering and mathematical finance related to the pricing and risk management of derivatives, and formulate the problems of hedging and calibration. The application of deep learning to hedging, on which we focus in this paper, is formulated under a framework referred to as *convex risk minimization*. Since this framework differs from the conventional risk-neutral valuation framework, it is beneficial to understand the relationship between these two frameworks in order to understand the current trend of research on the application of deep learning to financial engineering and mathematical finance. With this in mind, we summarize the history of the development of each framework and subsequently formulate the hedging and calibration problems.

In section 3, we provide an overview of deep learning. A building block of deep learning is the neural network, a system of functions in which linear and non-linear functions are alternately composed. Deep learning is one of machine learning methods that involve deep layers of neural networks and has achieved remarkable results in various fields in recent years. To provide an overview of these recent developments, we discuss the positioning of deep learning within the broader machine learning field and explain its historical development. We also provide brief explanations of the techniques used for efficiently training deep learning models and of the theoretical characteristics of neural networks. Subsequently, the notation adopted in the following sections is introduced.

In section 4, we review the literature on *deep hedging*, a technique that applies deep learning to the problem of hedging. As explained in section 2, the conventional approach in current practice is built on the risk-neutral valuation framework, where hedging strategies are derived based on the concept of the *replication* of derivatives. However, this approach has difficulty in quantitatively incorporating real-world market frictions, such as transaction costs, into pricing and hedging strategies because it assumes hypothetical frictionless markets. An alternative to the replication-based approach is the framework of convex risk minimization. In this approach, the agent seeks to minimize the risk associated with profit and loss taking realistic elements such as transaction costs and hedging errors into consideration. Deep hedging employs deep learning to perform the optimization calculations in the convex risk minimization framework, making it practically feasible to derive effective hedging strategies. Specifically, it derives optimal hedging strategies by describing a hedging strategy using a neural network and training its parameters to minimize *loss risk* (measured based on a convex risk measure) under given future market scenarios. Note that deep hedging is a general method to derive portfolio management strategies that minimize profit and loss risk and thus has potential applications beyond issues related to derivatives, such as asset liability management. In section 4, after explaining the deep hedging framework, we summarize current research trends and the outlook for potential future developments.

In section 5, we review the literature on *deep calibration*, a technique that applies deep learning to calibration in derivative pricing. Calibration is a calculation to adjust model parameters so that the model prices (the prices of the instruments based on risk-neutral valuation) match the market prices of highly liquid financial instruments. Thus, it is an important process to ensure that a model does not violate the *no-arbitrage* condition among instruments, that is, to ensure that the pricing of illiquid complex financial instruments is consistent with the prices of liquid instruments. Conventionally, iterative optimization methods such as the Newton–Raphson method have been used in calibration; however, these methods come with computational challenges, especially when calibrating numerous financial products. In contrast, deep calibration bypasses iterative calculations by pre-training the relationship between model parameters and model prices using neural networks. Some studies have reported that deep calibration accelerates and stabilizes derivative price calculations, and, accordingly, practitioners have been paying close attention to it, as it may open the possibility of using computationally expensive pricing models. Deep calibration essentially replaces the optimization calculations within the risk-neutral valuation framework with deep learning techniques, aiming to improve the computational techniques within current derivative practices. In section 5, we describe the deep calibration framework, summarize current research trends, and outline the practical advantages and challenges of the method.

To summarize, deep hedging focuses on the convex risk minimization framework, while deep calibration applies deep learning within the risk-neutral valuation framework. The fact that these two techniques are formulated under the two distinct frameworks indicates that there is a following significant difference in terms of their impact on practice. Deep hedging has the potential to significantly transform derivative practices in the long run, although it requires further theoretical and practical studies as there are hurdles to use it in finance practices, such as how to measure loss risk and how to generate future market scenarios. On the other hand, deep calibration primarily targets improving computational techniques within the risk-neutral valuation framework. Therefore, it is expected to provide promising solutions to specific challenges in current derivative practices.

This paper can be read in various manners depending on the readers' knowledge and interest. Section 2 summarizes the background of the finance issues addressed in sections 4 and 5. Thus, the readers familiar with financial engineering and derivative practices may choose to skip it. Similarly, section 3 provides a basic discussion of the deep learning techniques used in sections 4 and 5, and thus the readers familiar with deep learning may find it unnecessary to read. Sections 4 and 5 can be read independently, so that the readers can select only the content that matches their interest.

2 Derivative pricing and risk management: Hedging and calibration

In this section, we summarize the conceptual framework of financial engineering and mathematical finance related to derivative pricing and risk management, and then formulate the problems of hedging and calibration.

Historically, the conventional theories of financial engineering and mathematical finance have evolved around the framework of risk-neutral valuation. On the other hand, deep hedging, one of main focuses of this paper, is formulated based on a framework known as convex risk minimization. To understand the trend of applying deep learning to financial engineering and mathematical finance, it is important to understand the relationship between these two frameworks.

To this end, in this section, we first summarize the historical development of the frameworks of risk-neutral valuation and convex risk minimization, keeping in mind the issues of derivative pricing and risk management. We subsequently address the specific issues related to convex risk minimization and formulate the problem of deep hedging which is discussed in section 4. Using the Black–Scholes model as an example, we then outline the methods based on risk-neutral valuation and formulate the calibration problems discussed later in section 5. In the present section, our principal aim is to clarify the practical motivations behind hedging and calibration. Therefore, we introduce each concept in an inductive way, starting from practical motivations, taking an approach that differs from the usual explanations provided in financial engineering and mathematical finance.

For standard explanations of topics in mathematical finance, such as the framework of risk-neutral valuation, readers may refer to introductory books such as Björk [2009], Shreve et al. [2004], and Baxter et al. [1996]. For literature featuring practical challenges, books such as Andersen and Piterbarg [2010] and Hull [2014] are also useful references. Föllmer and Schied [2016] provides a detailed explanation of the theory of *incomplete markets*, in which the framework of convex risk minimization is covered.

2.1 Development history of derivative pricing and risk management

2.1.1 Framework of risk-neutral valuation

Financial engineering and mathematical finance, whose main topics include derivative pricing and risk management, have evolved around two concepts: *replication* and *no-arbitrage*. Replication refers to generating cash flows identical to a derivative by appropriately trading the underlying assets. No-arbitrage means that one cannot profit without initial capital and without bearing any risk of making losses. If the market satisfies the no-arbitrage condition and a derivative is replicable, then the fair value of that derivative is equivalent to the initial cost required for replication. This is a fundamental principle in financial engineering and mathematical finance.[1] From a practical standpoint, the equivalence between replicability and hedging of derivatives is crucial. Namely, if a derivative is replicable, taking an opposite position through replication can fully eliminate the uncertainty in profit and losses (P&Ls). Consequently, the trading strategy for replication (replication strategy) is identical to a hedging strategy for the derivative.

The principles of replication and no-arbitrage were theoretically developed within the framework of risk-neutral valuation. In this framework, the fair value of a derivative equals the expected value of future cash flows under a probability measure called the risk-neutral measure, discounted by the risk-free interest rate. The following is an intuitive explanation of why derivatives with uncertain future cash flows can be priced by discounting them at the risk-free interest rate. First, for a derivative that can be replicated or hedged, we can eliminate the risks of underlying price fluctuations and construct a risk-free portfolio that consists of the derivative and the hedging instrument. Moreover, if the no-arbitrage condition holds,

[1] For instance, if the price of a derivative exceeds the initial cost required to replicate it, selling the derivative would enable one to make profit without bearing any risk through replication, thereby violating the no-arbitrage condition. Similarly, it can be demonstrated that the opposite scenario also leads to the violation of the no-arbitrage condition. Hence, it becomes evident that the price of the derivative must be equal to the initial cost required for replication.

the return rate on the risk-free portfolio must be equal to the risk-free interest rate. Thus, based on these discussions of replication and no-arbitrage, derivatives can be priced under risk-neutral valuation.

In the seminal work on the Black–Scholes model (Black and Scholes [1973]), the option pricing formula was derived in a heuristic way based on the ideas of replication and no-arbitrage. In section 2.3, we formulate the framework of risk-neutral valuation from this perspective. Following the emergence of the Black–Scholes model, Harrison and Kreps [1979] and Harrison and Pliska [1981] established a rigorous mathematical proof for the fact that an arbitrage-free market admits risk-neutral valuation, specifically, that a risk-neutral probability measure exists when the no-arbitrage condition is satisfied.

There are two reasons why the theoretical framework of risk-neutral valuation based on the principles of no-arbitrage and replication has been accepted by practitioners in the financial industry and developed further. First, in the risk-neutral valuation framework, one can obtain replication strategies explicitly based on rigorous mathematical tools such as Itô calculus. For instance, models in mathematical finance, including the Black–Scholes model, describe asset prices using stochastic differential equations,[2] which is the theoretical basis for the techniques used in practice. Second, there is no need to estimate expected returns on the underlying assets. The expected returns on financial assets are notoriously volatile and challenging to estimate.[3] Therefore, derivative models requiring estimates of expected growth rates of underlying asset prices were impractical for pricing and risk management. In contrast, within the risk-neutral valuation framework, the estimation of expected returns is unnecessary, because observable risk-free interest rates are used for pricing. As elaborated in section 2.3, complex derivative pricing relies on observable market prices of highly liquid financial instruments. This implies that with risk-neutral valuation, one can derive pricing and hedging strategies based solely on currently observable information (see Table 1 for a comparison with deep hedging, which will be discussed in section 4). This is a significant advantage in option pricing and risk management.

2.1.2 Framework of convex risk minimization

As mentioned above, while the framework of risk-neutral valuation has significant advantages, there are also practical difficulties. Most theoretical models, including the Black–Scholes model, are based on the assumption of an ideal financial market, where no market frictions exist (e.g., risk-free assets can be borrowed and lent freely, and the underlying asset can be bought or sold without any transaction cost). However, in a real market, transaction costs exist, and various constraints such as short-selling constraints and borrowing constraints are present. Particularly, after the financial crisis of 2007-08, the impact of counterparty credit risks and funding costs on transactions has become significant, making valu-

[2]For instance, in the Black–Scholes model, it is assumed that the price of the underlying asset follows a stochastic process called geometric Brownian motion. Under this stochastic process, the price of the underlying asset at a certain point in time follows a log-normal distribution.

[3]For example, see Cochrane [2011] and Martin [2016].

ation adjustments like xVAs (x-Valuation Adjustments) crucial.[4] These market frictions and associated transaction terms affect optimal hedging strategies and also influence the pricing of derivatives through their impact on hedging costs.

In situations where market frictions exist, replication and hedging errors are inevitable as perfect replication is not possible, which induces uncertainty into the total P&L after hedging. Therefore, it is necessary to consider frameworks other than risk-neutral valuation, leading to numerous attempts to develop theories for incomplete markets, where factors such as transaction costs prevent perfect replication. Leland [1985] extended the Black–Scholes model to propose methods for modifying replication strategies according to the amount of transaction costs and constraints such as trading frequency, demonstrating that the fair value of derivatives remains within a certain range even when market frictions exist. Building on this, Hodges and Neuberger [1989] incorporated the utility functions to formulate the optimality of hedging strategies, quantifying the uncertainty in total P&Ls after hedging.[5] In addition, Davis et al. [1993] refined the work of Hodges and Neuberger [1989] and formulated a setting consistent with the Black–Scholes model, and many studies have examined price behavior under this setting (e.g., Shreve [1995], Rogers [2004], Whalley and Wilmott [1997], and Barles and Soner [1998]). More recently, Xu [2006] and Ilhan et al. [2009] introduced the idea of minimizing a convex risk measure instead of maximizing utility. This approach offers the advantage of providing a more realistic representation of trader preferences and a clearer relationship between price and hedging strategies, as described in section 2.3.1.

This paper employs the formulation by Hodges and Neuberger [1989], which deals with the framework of convex risk minimization. Specifically, it aims to quantify the uncertainty in total P&Ls after hedging using convex risk measures and seeks hedging strategies that minimize this risk. For instance, when adopting expected shortfall (ES) as a convex risk measure, the problem becomes that of minimizing the area under the distribution of total losses (i.e., minimizing the potential for significant losses). In this framework, the derivative price can be set as the price at which an agent has the same expected utility (or risk measure) when trading the derivative and when not trading it, known as the *indifference price*.

Despite certain theoretical advancements, the derivation of hedging strategies considering

[4]xVAs are adjustments to the market price of derivatives that take into account various costs associated with derivative trading, such as counterparty credit risks and funding costs.

[5]In the context of incomplete markets, the primary formulations for the fair pricing of derivatives include approaches based on utility functions, dominant replication, and the minimal equivalent martingale measure (MEMM). Dominant replication involves trading the underlying asset in such a way that the total P&Ls at maturity are non-negative regardless of the outcome. The fundamental idea is to set the fair value of the derivative as the minimum cost required to achieve dominant replication. However, strategies for dominant replication often become trivial (for instance, holding one unit of the underlying asset from the start to replicate a single call option), making this approach impractical due to resulting high fair prices (see Kramkov [1996] and Föllmer and Kabanov [1997]). The MEMM approach involves searching for a measure within the set of equivalent martingale measures (or risk-neutral probability measures) that minimizes hedging errors. However, challenges arise, such as the lack of practically feasible hedging strategies under the identified measure (see Föllmer and Schweizer [1991] and Miyahara [2001]). For a survey on the impact of transaction costs on derivatives and other financial assets, Muhle-Karbe et al. [2017] is a valuable reference.

transaction costs requires complex optimization calculations, preventing the practical application of incomplete market theories for a long time. Thus, in derivative practices relying on risk-neutral valuation as a benchmark, it is customary to make adjustments to cover errors incurred from market frictions and numerical inaccuracies in model calculations. Since most of these adjustment methods rely on simplified formulas and trader experience, more objective methods based on incomplete market models have long been called for.

In this historical context, deep hedging, a hedging technique which utilizes deep learning, has been rapidly progressing from both theoretical and practical perspectives in recent years. Deep hedging, proposed by Buehler et al. [2019a], performs optimization calculations under the framework of an incomplete market such as convex risk minimization using deep learning, allowing for pricing and risk management that considers transaction costs and other factors.

There is a significant difference between deep hedging based on convex risk minimization and hedging derived from risk-neutral valuation. In the latter approach, one does not need to specify the traders' risk preferences explicitly, because under the assumption that a complete replication is possible, uncertainty in payoffs is eliminated and thus traders' risk preferences toward uncertainty do not matter. As a result, the hedge strategy is uniquely determined irrespective of trader preference and utility function. On the other hand, deep hedging solves the convex risk minimization problem by explicitly considering trader utility function (or risk measure that quantifies loss risk). As mentioned above, it is necessary to specify how traders evaluate uncertainty regarding possible losses in the case where complete replications and hedges are not possible. One may regard deep hedging as a revival of conventional asset pricing theory in the sense that it derives optimal trading strategies by taking the traders' utility function (risk preference) into consideration.[6]

2.2 Formulation of hedging: Convex risk minimization problem

In this section, we formulate the hedging problem for derivatives as a convex risk minimization problem. In derivative hedging, the objective is to mitigate the uncertainty of the total P&L at maturity by trading hedging instruments (assets used for hedging) up to the derivative's expiration. Here, we formulate the P&L at maturity as a random variable. The convex risk minimization problem aims to minimize the risk of the P&L at maturity, which is quantified by a convex risk measure. Subsequently, following Buehler et al. [2019a], we formulate this concept in a discrete-time model.

[6] Among the foundational results in modern finance theory, Markowitz [1952] portfolio selection theory and the Capital Asset Pricing Model (CAPM) by Sharpe [1964]; Lintner [1965]; Mossin [1966] stand out. Markowitz's theory analyzes investors who optimally choose their portfolio based on mean-variance utility, considering the expected return (mean) and risk (variance) of portfolios. CAPM identifies the determinants of risk premiums for individual risky assets under the assumption of a representative investor maximizing expected utility. Thus, research that assumes investors maximize expected utility (or minimize expected loss) has become a primary approach in finance, providing insights into optimal investment behavior and asset pricing.

2.2.1 P&L at maturity

Let $t_0 = 0$ be the initial time and T be the maturity of a derivative. We assume that market transactions are possible at times $t_0 < \cdots < t_k < \cdots < t_n = T$. Let $(\Omega, \mathcal{F}, P)$ be a probability space and $\{\mathcal{F}_k\}_{k=0,\ldots,n}$ be a discrete filtration. For simplicity, we consider only one underlying asset, and let its asset price $\{S_k\}_{k=0,\ldots,n}$ be an $\mathcal{F}_k$-adapted real-valued stochastic process. Furthermore, let the payoff Z_T of the derivative be an F_n-measurable real-valued random variable. For example, for a European call option with a strike price K, $Z_T = \max\{S_n - K, 0\}$.[7]

For simplicity, we consider a setting where only the underlying asset is used as a hedging instrument.[8] In this case, a hedging strategy is determined by the holding amounts of the underlying asset at each time, which is an $\mathcal{F}_k$-adapted real-valued stochastic process $\delta = \{\delta_k\}_{k=0,1,\ldots,n}$. Moreover, the transaction costs determined by the trading volume of the hedging instrument at each time t_k are denoted as $c_k(\delta_k - \delta_{k-1})$.[9] For instance, if transaction costs are proportional to trading volume, with a constant percentage ε of the transaction amount, then $c_k(\delta_k - \delta_{k-1}) = \varepsilon|\delta_k - \delta_{k-1}|S_k$. Let p_0 denote the current price of this derivative.

Given this setup, the P&L at maturity for an investor with a short position in the derivative is as follows:

$$\mathrm{PL}^{(Z_T, p_0, \delta)} = p_0 - Z_T + \sum_{k=0}^{n-1} \delta_k (S_{k+1} - S_k) - \sum_{k=0}^{n} c_k (\delta_k - \delta_{k-1}). \tag{1}$$

The first and second terms on the right-hand side represent the profit from selling the derivative at the initial time t_0 and the payoff that must be paid at maturity T, respectively. The third term on the right-hand side represents the cumulative hedging profit or loss resulting from the hedging transactions. The final term on the right-hand side represents the cumulative transaction costs incurred for adjusting the hedge position at each possible trading point.[10]

2.2.2 Risk measurement of P&L

The P&L at maturity, formulated in the previous section, is a random variable. A real-valued function that maps this random variable to a measurement of risk is simply called a risk measure. Particularly in mathematical finance, convex risk measures that satisfy the

[7]The subsequent discussions also hold even when Z_T is an $\mathcal{F}_n$-measurable random variable, making them suitable for cases of payoffs depending on other than S_n, including path-dependent payoff cases. The framework is also easily extendable to scenarios with multiple underlying assets.

[8]The approach can also be extended to settings where other assets serve as hedge instruments.

[9]Here, we set $\delta_{-1} = \delta_n = 0$, so that transaction costs for purchasing δ_0 units of the hedge instrument at the beginning of the hedge operation and selling δ_{n-1} units at the maturity are reflected in the P&L.

[10]In Buehler et al. [2019a], the notation $\mathrm{PL}_T(Z_T, p_0, \delta)$ is used for the P&L at maturity. In this paper, since the P&L at maturity is formulated as a random variable, we denote the P&L at maturity as $\mathrm{PL}^{(Z_T, p_0, \delta)}(\omega)$ for $\omega \in \Omega$. Note that the arguments to be modified in subsequent discussions are (Z_T, p_0, δ), and given that only the P&L at maturity are considered, we adopt the notation $\mathrm{PL}^{(Z_T, p_0, \delta)}$.

following definition are frequently considered. For a more detailed explanation of convex risk measures, see Föllmer and Schied [2016].

Definition 1 (Convex risk measure)**.** *Let χ be the space of all random variables on Ω. A function $\rho : \chi \to \mathbb{R}$ is called a convex risk measure if it satisfies the following conditions:*

1. *Monotonic Decrease: For $X_1, X_2 \in \chi$, if $X_1 \geq X_2$, then $\rho(X_1) \leq \rho(X_2)$.*
2. *Convexity: For $\alpha \in [0, 1]$ and $X_1, X_2 \in \chi$, $\rho(\alpha X_1 + (1-\alpha)X_2) \leq \alpha\rho(X_1) + (1-\alpha)\rho(X_2)$.*
3. *Cash Invariance: For $X \in \chi$ and $c \in \mathbb{R}$, $\rho(X + c) = \rho(X) - c$.*

The intuitive meaning of each condition for a convex risk measure is as follows: Firstly, a monotonic decrease implies that if the P&L of portfolio 1, X_1, is greater than or equal to the P&L of portfolio 2, X_2, then the risk measure of portfolio 1 will be less than or equal to that of portfolio 2 (indicating that portfolio 1 is less risky). Secondly, convexity formulates what is known as the diversification effect, stating that the risk of a portfolio formed by diversifying between two portfolios is less than or equal to a weighted average of the individual portfolio risks. Lastly, cash invariance means that a certain payoff c will reduce the risk measure by the same amount. These properties are deemed essential for a risk measure, and convex risk measures, including ES, have long been a topic of study in finance; see Föllmer and Schied [2016].[11]

2.2.3 Convex risk minimization problem

Given the above setting, the convex risk minimization problem is formulated as follows.

Definition 2 (Convex risk minimization problem)**.** *Given a convex risk measure ρ, the convex risk minimization problem for hedging a derivative with payoff Z_T using a hedging instrument S is defined as the problem of finding a hedging strategy $\delta^* = \{\delta_k^*\}_{k=0}^n$ that solves the following minimization problem:*

$$\delta^* = \arg\min_{\delta \in \bar{\mathcal{H}}} \rho\left[\mathrm{PL}^{(Z_T, p_0, \delta)}\right]. \tag{2}$$

Here, the minimum is sought over the appropriate set of hedging strategies $\bar{\mathcal{H}}$ (a subset of the space of real-valued stochastic processes).

Remark 1 (Exploration space of hedging strategies)**.** *In Definition 2, it was assumed that within the appropriate set of hedging strategies $\bar{\mathcal{H}}$, there exists a δ^* that minimizes $\rho\left(\mathrm{PL}^{(Z_T, p_0, \delta)}\right)$. However, the conditions for $\bar{\mathcal{H}}$ to satisfy this assumption are not trivial.[12] Therefore, in*

[11]The expected shortfall $\mathrm{ES}_\alpha(X)$ is defined as $\mathrm{ES}_\alpha(X) = -E\left[X \mid X \leq \mathrm{VaR}_\alpha(X)\right]$. It is essential to note that while Value at Risk (VaR) is a commonly used risk measure in financial practice, it does not satisfy convexity (i.e., it is not a convex risk measure). For research examining the properties of risk measures, refer to works such as Föllmer and Leukert [2000] and Föllmer and Schied [2002].

[12]Lemma 2.1 in Godin [2016] provides the conditions in the case of the ES risk measure.

Buehler et al. [2019a], the authors defined the set of selectable hedging strategies $\mathcal{H}$ without assuming the existence of a minimum and discuss the infimum $\inf_{\mathcal{H}} \rho\left(\mathrm{PL}^{(Z_T, p_0, \delta)}\right)$ *rather than the minimum (refer to section 4.1). In formulating deep hedging, the existence of a* δ^* *that minimizes* $\rho\left(\mathrm{PL}^{(Z_T, p_0, \delta)}\right)$ *is not an issue. Hence, in this paper, we place strong assumptions on* $\bar{\mathcal{H}}$ *to explicitly formulate the convex risk minimization problem.*

2.2.4 Utility-indifference price

Under the framework of convex risk minimization, it is common to price derivatives as utility-indifference prices defined below. Here, the minimized risk quantity is denoted by $\pi(Z_T, p_0) = \min_{\delta \in \bar{\mathcal{H}}} \rho\left(\mathrm{PL}^{(Z_T, p_0, \delta)}\right)$.

Definition 3 (Utility-indifference price)**.** *The utility-indifference price* p_0^* *for a derivative with payoff function* Z_T *satisfies*

$$\pi(Z_T, p_0^*) = \pi(0, 0).$$

It is thus the case that the utility-indifference price is a price such that the risk quantity when one does not trade the derivative at all equals the risk quantity when one trades the derivative with an optimal hedging trade. Further, using equation (1) and the cash invariance of the convex risk measure (Condition 3 of Definition 1), we can derive

$$\begin{aligned}
\pi(Z_T, p_0^*) &= \min_{\delta \in \bar{\mathcal{H}}} \rho\left[p_0^* - Z_T + \sum_{k=0}^{n-1} \delta_k (S_{k+1} - S_k) - \sum_{k=0}^{n-1} c_k (\delta_k - \delta_{k-1})\right] \\
&= -p_0^* + \min_{\delta \in \bar{\mathcal{H}}} \rho\left[-Z_T + \sum_{k=0}^{n-1} \delta_k (S_{k+1} - S_k) - \sum_{k=0}^{n-1} c_k (\delta_k - \delta_{k-1})\right] \\
&= -p_0^* + \pi(Z_T, 0).
\end{aligned}$$

From Definition 3, it follows that

$$p_0^* = \pi(Z_T, 0) - \pi(0, 0).$$

2.3 Formulation of calibration: Estimation of model parameters in risk-neutral valuation

Calibration refers to the process of adjusting model parameters to ensure consistency between observed market prices and theoretical prices based on the model. In general derivatives practice, after calibrating the model parameters to align with highly liquid financial instruments whose prices are considered to be informative, models are used for pricing and deriving hedging strategies for complex financial instruments that are not actively traded in the market or have low liquidity. Therefore, the development of techniques that achieve rapid and stable calibration is crucial for derivatives practice and risk management. In this paper, we focus on calibration within the framework of risk-neutral valuation. Specifically, we address the problem of adjusting model parameters to ensure that the model prices (the risk-neutral price) are consistent with market prices. In the following sections, we first discuss the intuitive concept of risk-neutral valuation, considering its relationship with convex risk minimization. Subsequently, we formulate the calibration problem.

2.3.1 Derivation of risk-neutral valuation

In mathematical finance, risk-neutral valuation employs continuous-time models, often under the assumption of an hypothetical frictionless market. This ideal setup postulates that transactions, including short selling, can be executed at any time without incurring trading costs. To distinguish the P&L in the discrete-time model, $\mathrm{PL}^{(Z_T,p_0,\delta)}$, discussed in the previous section, we denote the P&L in the continuous model as $\widetilde{\mathrm{PL}}_t^{(Z_T,p_t,\delta)}$. Here, for $t \in [0,T]$, $\widetilde{\mathrm{PL}}_t^{(Z_T,p_t,\delta)}$ represents the P&L from time t to T.[13]

In this notation, replication is formulated as a pair consisting of a risk-neutral price $p_t^{(\mathrm{RN})}$ and a corresponding replication strategy $\{\widetilde{\delta}_u^{(\mathrm{RN})}\}_{u\in[0,T]}$ to hedge the uncertainty associated with the derivative's payoff that satisfies the following condition:

$$P\left(\widetilde{\mathrm{PL}}_t^{(Z_T,p_t^{(\mathrm{RN})},\widetilde{\delta}^{(\mathrm{RN})})} = 0\right) = 1 \quad \text{for } t \in [0,T]. \tag{3}$$

In other words, if replication is feasible, the uncertainty associated with the derivative's payoff can be eliminated through hedging, ensuring that the P&L at maturity T can be almost surely guaranteed to be zero. Thus, replication serves as a method to eliminate loss risk related to the derivative's payoff without explicitly assuming a risk measure.

The pioneering work by Black and Scholes [1973] and Merton [1973] utilized Itô calculus to formulate a specific replication technique for derivatives. The derivation approach proposed by them differs somewhat from recent standard explanations in financial engineering and mathematical finance. However, it is useful for understanding the relationship between hedging and calibration, so we present an outline as follows. Consider a European option based on an underlying asset price $\widetilde{S}_t$. Namely, the payoff at maturity T is $Z_T = f(\widetilde{S}_T)$, where f is a deterministic payoff function and $\widetilde{S}_T$ is the underlying asset price at maturity T. Let us denote the option price at time t as p_t. The underlying asset price follows the stochastic differential equation:

$$\widetilde{S}_t = \widetilde{S}_0 + \int_0^t \mu \widetilde{S}_u \, du + \int_0^t \sigma \widetilde{S}_u \, dW_u,$$

where W_t represents the standard Brownian motion under the probability measure P, and μ and σ are constant parameters denoting drift and volatility, respectively. This stochastic process is known as geometric Brownian motion, where the price of the underlying asset at a fixed time follows a log-normal distribution. Then, the dynamics of the P&L are formulated

[13]In discrete-time models, $\mathrm{PL}^{(Z_T,p_0,\delta)}$ represents the P&L occurring from time 0 to T.

as a stochastic process, with

$$\widetilde{\mathrm{PL}}_t^{(Z_T,p_t,\widetilde{\delta})} = p_t - Z_T + \int_t^T \widetilde{\delta}_u \, d\widetilde{S}_u.$$

Let us assume that the option price p_t at time t is expressed as a function of t and $\widetilde{S}_t$ as

$$p_t = v(t, \widetilde{S}_t).$$

The dynamics of $\widetilde{\mathrm{PL}}_t^{(Z_T,p_t,\widetilde{\delta})}$ with respect to t can be described using Itô formula as

$$\begin{aligned} d\widetilde{\mathrm{PL}}_t^{(Z_T,p_t,\widetilde{\delta})} &= \frac{\partial}{\partial t}v(t,\widetilde{S}_t)\,dt + \frac{\partial}{\partial x}v(t,\widetilde{S}_t)\,d\widetilde{S}_t + \frac{1}{2}\frac{\partial^2}{\partial x^2}v(t,\widetilde{S}_t)\,d\widetilde{S}_t \cdot d\widetilde{S}_t - \widetilde{\delta}_t\,d\widetilde{S}_t \\ &= \left(\frac{\partial}{\partial t}v(t,\widetilde{S}_t) + \frac{\partial}{\partial x}v(t,\widetilde{S}_t)\mu\widetilde{S}_t + \frac{1}{2}\frac{\partial^2}{\partial x^2}v(t,\widetilde{S}_t)\sigma^2\widetilde{S}_t^2 - \widetilde{\delta}_t\mu\widetilde{S}_t\right)dt + \left(\frac{\partial}{\partial x}v(t,\widetilde{S}_t) - \widetilde{\delta}_t\right)\sigma\widetilde{S}_t\,dW_t. \end{aligned}$$

Here, $\frac{\partial}{\partial x}$ represents partial differentiation with respect to the second argument S_t of v. Then, as a condition for eliminating the risk of this P&L process, that is, setting the diffusion term to zero, we obtain the following:

$$\widetilde{\delta}_t^{\mathrm{Del}} = \frac{\partial}{\partial x}v\left(t, \widetilde{S}_t\right).$$

This is the hedging strategy known as delta hedging. The right-hand side of the above equation is the sensitivity of the derivative price to changes in the price of the underlying asset. Therefore, the intuitive idea behind delta hedging is to hold an amount of the underlying asset equal to the sensitivity (delta) per one unit of the derivative, so that the price changes in the underlying and hedge position completely offset each other.[14].

Further, given that (1) a risk-free asset is traded, (2) the market is arbitrage-free, and (3) the P&L at maturity are zero ($\widetilde{\mathrm{PL}}_t^{(Z_T,p_t,\widetilde{\delta})} = 0$), it can be demonstrated that v satisfies the Black–Scholes partial differential equation:

$$\begin{cases} \dfrac{\partial}{\partial t}v(t,x) + rx\dfrac{\partial}{\partial x}v(t,x)\,d\widetilde{S}_t + \dfrac{\sigma^2 x^2}{2}\dfrac{\partial^2}{\partial x^2}v(t,x) = rv(t,x), \\ v(T,x) = f(x). \end{cases} \tag{4}$$

Here, r represents the continuously-compounded risk-free interest rate of the risk-free assets. For the solution v of this equation, setting $p_t = v\left(t, \widetilde{S}_t\right)$, we have

$$P\left(\widetilde{\mathrm{PL}}_t^{(Z_T,p_t,\widetilde{\delta}^{\mathrm{Del}})} = 0\right) = 1 \quad \text{for } t \in [0,T].$$

[14]If we assume more complex models for the underlying asset price $\widetilde{S}_t$, this argument does not hold, and thus condition (3) can fail. In such a case, one seeks a hedging strategy $\{\widetilde{\delta}_u^{(\mathrm{RN})}\}_{u\in[0,T]}$ such that condition (3) is approximately satisfied, taking into account delta, vega, gamma, and other higher-order greeks. Such a strategy is referred to as *greeks hedging* or *parameter hedging*.

Furthermore, v can be expressed in terms of expectation using the Feynman–Kac formula as follows:

$$v(t, \widetilde{S}_t) = e^{-r(T-t)} E^Q \left[f(\widetilde{S}_T) \;\middle|\; \widetilde{S}_t \right]. \tag{5}$$

Here, the probability measure Q is the risk-neutral probability measure. Under this probability measure, $\widetilde{S}_t$ follows the stochastic differential equation

$$\widetilde{S}_t = \widetilde{S}_0 + \int_0^t r\widetilde{S}_u \, du + \int_0^t \sigma\widetilde{S}_u \, dW_u^Q.$$

Equation (5) shows that the option value v is equal to the present value of the expected payoff under the risk-neutral probability measure discounted by the risk-free interest rate. This is called the risk-neutral valuation framework because derivative prices can be obtained based on this equation.

When computing option prices based on risk-neutral valuation, it is essential to predetermine parameters like r and σ, which describe the fluctuations of S_t under the probability measure Q. Moreover, in models more complex than the Black–Scholes model, determining multiple model parameters becomes necessary. This parameter determination is discussed in the subsequent section on calibration.

2.3.2 Calibration problem

When pricing derivatives and deriving hedging strategies based on risk-neutral valuation, it is necessary to determine the values of the parameters (for instance, r and σ in the Black–Scholes model).

As mentioned at the beginning of section 2, the general approach in current derivative practice for determining these parameters is to assume that the market prices of financial instruments with high market liquidity (the instruments to be calibrated) are correct and to set the parameters so that the theoretical model prices fit those market prices. Adjusting the parameters of the *model* in this way is referred to as calibration in the theory of derivative pricing. Using calibrated models, practitioners price more complex derivative instruments whose market prices are unobservable. Such procedures allow for pricing complex derivatives so as to be consistent with the market prices of highly liquid instruments; that is, there is no-arbitrage between the complex derivatives and the highly liquid instruments. Thus, accurate and efficient calibration is extremely important in derivative pricing and risk management.

Below, we formulate the calibration problem. First, consider the following three parameter groups:

- Model parameters (parameters of the stochastic differential equation governing the underlying asset price): $\sigma \in \mathbb{R}^Q$
- Product-specific parameters (such as maturity and strike price): $\tau \in \mathbb{R}^p$

- External factors (e.g., risk-free interest rates): $\phi \in \mathbb{R}^m$

Under this framework, let $\mathrm{QP}(\tau)$ denote the market price of a financial product with product-specific parameters τ, and let $\mathrm{MP}(\sigma; \tau, \phi)$ represent the model price under risk-neutral valuation.

Given R calibration target products with market prices $\{\mathrm{QP}(\tau_i)\}_{i=1,\dots,R}$, the calibration problem can be formulated as a minimization problem to determine the model parameters:

$$\sigma^* = \operatorname*{argmin}_{\sigma \in \mathbb{R}^Q} \mathrm{Error}\left(\sigma; \{\mathrm{QP}\left(\tau_i\right)\}_{i=1,\dots,R}, \phi\right), \tag{6}$$

where Error is an error function, that is, a function that penalizes the error between the market price $\mathrm{QP}(\tau)$ and the model price $\mathrm{MP}(\sigma; \tau, \phi)$. For example, the following function is used:

$$\mathrm{Error}\left(\sigma; \{\mathrm{QP}\left(\tau_i\right)\}_{i=1,\dots,R}, \phi\right) = \sum_{i=1}^{R} \left(\mathrm{MP}\left(\sigma; \tau_i, \phi\right) - \mathrm{QP}(\tau_i)\right)^2.$$

A typical practical problem is to adjust about 10 model parameters for tens to hundreds of calibration targets.[15] For such high-dimensional optimization problems, methods like the Newton–Raphson method are commonly employed. However, these optimizations are often computationally expensive and may not necessarily converge. Especially when convergence issues arise, it is difficult to determine whether the problem lies with the optimization algorithm or the set of model parameters that can represent the market prices with in an acceptable error range is empty. Hence, calibration stands as one of the most intricate stages in the development of derivative pricing models. To address this, deep calibration, introduced in section 5, aims to replace these optimization computations with deep learning techniques, thereby eliminating the need for optimization calculations with each valuation, leading to faster and more stable calculations.

3 Deep learning

In this section, we provide an overview of deep learning. Neural networks utilize a system of functions with a vast number of parameters, composed alternately of linear and non-linear functions. Machine learning methods that utilize functions with deep layers of neural networks are referred to as deep learning. While deep learning has empirically proven to be highly effective in numerous practical applications, recent studies have gradually elucidated its theoretical foundations.

In section 3.1, we provide a brief overview of deep learning and recent research trends in the field. In section 3.2, we introduce the notation used in the explanations in sections 4 and 5. For more detailed information on deep learning, see monographs such as Goodfellow et al. [2016].

[15] A typical example of calibration in practice is a calibration of the volatility of a Hull-White model, which has roughly 10 parameters, calibrated to tens to hundreds of swaptions data.

3.1 Background of success

3.1.1 Classification of machine learning and the positioning of deep learning

Deep learning is a kind of machine learning. Therefore, to understand the characteristics and potential applications of deep learning, it is beneficial to comprehend its positioning within machine learning as a whole. In the following, we provide a general classification of machine learning methods. We also provide detailed explanations of the positioning and characteristics of deep learning within the broader context of machine learning.[16]

In machine learning, one first trains a statistical model, called a machine learning model, using training data. These trained models are then employed for tasks such as decision-making, classification, and prediction. Here, training refers to estimating the parameters of a function system (statistical model) that links input and output data. Generally, machine learning is said to offer a more inductive and precise understanding of the relationships between data points as compared to classical statistical models, especially when vast amounts of data are used for training.

Given the above, machine learning can be classified from several perspectives. The first axis of classification is the type of objective function set in the learning (i.e., parameter estimation). In *supervised learning*, given a training dataset consisting of known output data, the model parameters are trained to replicate the given outputs as closely as possible. Here, for instance, a machine learning model might be trained with images of dogs and images of animals that are not dogs, with each image correctly labeled (e.g., as a dog or not a dog). The trained model is then used to judge whether a given arbitrary image is that of a dog. In *unsupervised learning*, tasks like data clustering are performed without labeled output. A third objective function-type category of machine learning is *reinforcement learning*. In reinforcement learning, there is an agent who seeks to maximize the rewards that will be receive in the future; the rewards depend on the agent's actions and the state of the model, which changes sequentially. The maximization result implies an optimal policy, which is a mapping from the state to the agent's optimal action. The second axis of classification is the purpose of the machine learning. Machine learning models can be developed for any number of purposes; for example, one might use a model to judge whether an email is spam, to classify handwritten numbers as 0 to 9, or to predict future stock prices and unemployment rates from various kinds of feature variables. The third axis of classification is the function system employed in the model's formulation. In machine learning, depending on the objective function of the optimization and on the purpose of the model, a specific function system with a suitable structure for the given model setup is employed. For example, logistic regression is used to predict relatively simple phenomena, random forest and k-nearest neighbor methods are used for classification and regression, and neural networks (multilayer perceptrons) are used to describe more complex phenomena.

[16]Machine learning is a vast and rapidly evolving field that encompasses various models, learning approaches, and applications. Consequently, there might be instances where the definitions and classes are not entirely clear.

Deep learning is a category under the third axis, in which a function system with a very large number of parameters that repeatedly synthesizes linear and nonlinear functions alternately (e.g., neutral network) is used. In particular, *deep reinforcement learning*, which uses neural networks in reinforcement learning, has solved many problems in recent years and is attracting widespread attention.

3.1.2 History and current status of deep learning

The history of deep learning began in the 1950s, and is said to currently be in its third wave of enthusiasm. The first boom (in the 1950s) was propelled by the development of the perceptron (Rosenblatt [1958]), which is often considered to be the prototype of neural networks. The perceptron modeled a nervous system, sparking high expectations for artificial intelligence. However, by the late 1960s, Minsky and Papert [1969] had highlighted the limitations of the perceptron in recognizing even simple shapes such as linearly separable figures, leading to a decline in interest.

The second boom began in the 1980s with the introduction of backpropagation (Rumelhart et al. [1988]). As explained in detail later, backpropagation is a method for efficiently performing the calculations required to update the parameters in the training of a multilayer neural network. This technique made learning possible for neural networks with several layers and greatly advanced the study of deep learning. Despite these advancements, however, issues like overfitting and the challenges in parameter updates led to another ebb in enthusiasm.[17]

The ongoing third boom gained momentum when Hinton et al. [2006] successfully applied pre-training to multilayer neural networks. Multilayer neural networks have since attracted widespread attention, leading to significant developments. For example, AlexNet by Krizhevsky et al. [2017] achieved high performance in the ImageNet Large Scale Visual Recognition Challenge (ILSVRC), and further improvement ensued. Deep learning has brought about breakthroughs such as AlphaGo's victory over the Go world champion (Silver et al. [2016]) and the development of data generation techniques (Goodfellow et al. [2020]). These achievements indicate that deep learning has expanded its applications to now include control problems and data generation tasks, in addition to traditional prediction and regression tasks.

Multiple reasons have been given for the success of deep learning in the current third boom. First are the advancements in the techniques and algorithms used for learning. As noted above, the second boom lost momentum because of the difficulties encountered in multilayer neural network learning due to such problems as overfitting. In the third boom, the development of breakthrough technologies that overcome this issue have played an important role. Second is rapid development of infrastructure to support numerical computations for deep learning. This includes the availability of powerful Graphics Processing

[17] Overfitting occurs when a learning model becomes overly tailored to the training data, leading to a decrease in performance when applied to unknown data.

Units (GPUs) and libraries like PyTorch and TensorFlow; these developments in hardware and software make it possible to execute large-scale, computationally expensive numerical calculations. Furthermore, accumulating experience in solving various problems with deep learning has fostered a positive feedback loop, encouraging further exploration and innovation. In other words, the accumulation of rules of thumb for solving problems using deep learning has made it easier to generate ideas for solving new problems with deep learning. Recent theoretical research has shed further light on why deep learning has been so successful in practical terms. Specifically, as we will shortly discuss in section 3.1.4, it is becoming clear that neural networks have theoretically desirable natural performance with respect to learning and function representation. Such theoretical ground might serve to effectively distinguish deep learning from its machine learning predecessor.

This trend has led to a growing body of research on the application of deep learning to financial engineering and mathematical finance. The main reasons for this are rooted in three properties that are specific to problems in these fields. First, as discussed in section 2, the problem of hedging derivatives is a control problem in that it requires a hedging strategy that controls the uncertainty of a P&L. For this reason, empirical rules of reinforcement learning, which have developed rapidly in recent years, are useful.[18] Second, derivative prices and hedging strategies are often described by functions with nonlinearity due to nonlinear payoff functions and other factors. Therefore, they are compatible with neural networks that have high representational capacity for the functions of interest for estimation purposes. Third, given that techniques in the derivatives domain are grounded in financial engineering and mathematical finance theory, precise numerical calculations and theoretical assurances are paramount. With recent advancements in deep learning technologies and infrastructure, achieving the required precision for practical applications in the derivatives domain is becoming feasible, although it still requires careful validation, as discussed in sections 4 and 5.

3.1.3 Technique for learning

As previously discussed, in machine learning, a statistical model called a machine learning model is first trained with appropriate data and then used for various tasks. Here, the term *training* refers to solving an optimization problem to estimate the parameter θ in the function $y = f(x; \theta)$ that connects the input data x to the output data y.

Specifically, parameter θ is estimated by minimizing the objective function, which is referred to as the loss function J. For instance, in supervised learning, where training data comprising pairs of input data x_i and corresponding output data y_i denoted as $\{(x_i, y_i)\}_{i=1,\dots,q}$ are provided, the error (difference) between the training data and the model's output is captured as loss and the problem is formulated so as to minimize its value. If the loss function J

[18]Indeed, the deep hedging introduced in section 4 draws inspiration from the policy gradient method of reinforcement learning.

is defined in terms of the squared error, it can be expressed as follows:

$$J(\theta) = \frac{1}{q}\sum_{i=1}^{q} e\left(\omega_i;\theta\right), \quad e\left(\omega_i;\theta\right) = \left(y_i - f(x_i;\theta)\right)^2 . \tag{7}$$

For model learning for specific problems such as calibration or hedging, it is essential to set an appropriate loss function.

The basic algorithm to solve the aforementioned minimization problem is to search for the optimal solution θ^* by iteratively updating the parameters and terminating the iteration when the update in θ_i becomes smaller than a certain threshold. Specifically, the update is given by

$$\theta_{i+1} = \theta_i + \nabla J(\theta_i).$$

Here, $\nabla J(\theta)$ represents the gradient, defined by $\nabla J(\theta) = \left(\frac{\partial J(\theta)}{\partial \theta^{(1)}}, \ldots, \frac{\partial J(\theta)}{\partial \theta^{(k)}}\right)$, where $\theta^{(k)}$ represents the k-th element of θ. This approach follows the idea of the Newton–Raphson method, where the gradient of the loss function is computed, and the parameters are updated in the direction in which the loss decreases.

In machine learning, θ often becomes high-dimensional, ranging from thousands to trillions of dimensions, making this iterative approach extremely challenging. For instance, issues such as the computational burden of computing the gradient $\nabla J(\theta)$ and convergence to saddle points or local minima (gradient vanishing problem) frequently arise (Figure 1).

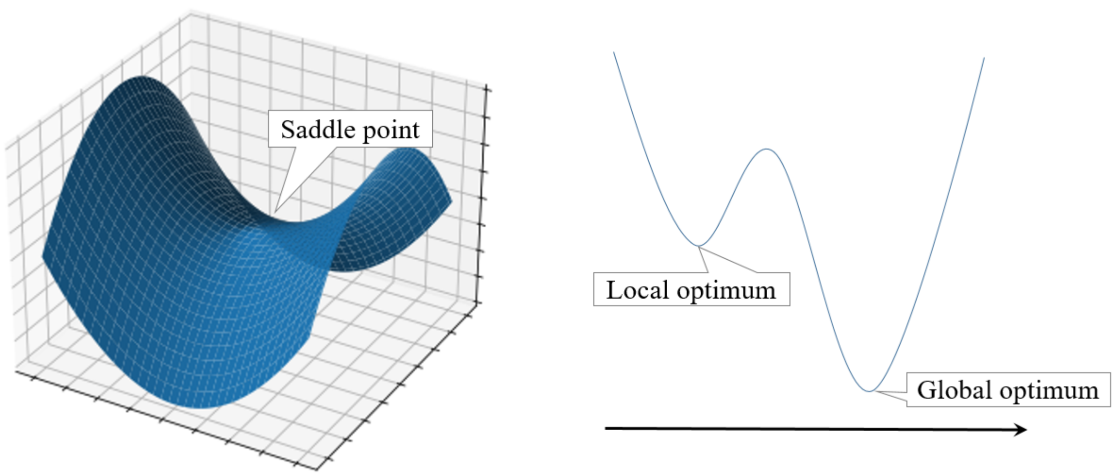

Figure 1: saddle point and optimal point

To address these challenges, various techniques have been proposed. The following three are among the most prominent of these techniques:

1. Setting the learning rate: Multiplying $\nabla J(\theta)$ by an appropriate constant ν_i prevents getting stuck in local minima or making excessive updates.

2. Stochastic gradient descent (SGD): Rather than using all the training data to compute

$\nabla J(\theta)$, SGD approximates it using a randomly sampled subset of the training data.[19] Here, the amount of training data extracted is q_B, referred to as the number of batches. For example, if the loss function is given by equation (7), one stochastically samples q_B data points as $\left\{(x_{i(j)}, y_{i(j)})\right\}_{j=1,\ldots,q_B} \subset \{(x_i, y_i)\}_{i=1,\ldots,q}$, and then approximates as follows.

$$\nabla J(\theta) = \frac{1}{q}\sum_{i=1}^{q} \nabla e\left(\omega_i;\theta\right) \approx \frac{1}{q_B}\sum_{j=1}^{q_B} \nabla e\left(\omega_{i(j)};\theta\right). \tag{8}$$

Note that in doing so, we take advantage of the fact that the loss function $J(\theta)$ can be represented in terms of the sum of the errors $e\left(\omega_i;\theta\right)$ with respect to individual data samples.[20]

3. Backpropagation: This is a method to efficiently compute $\nabla J(\theta)$ using the fact that neural networks are layered or composite functions. By using the chain rule of composite functions, partial differential coefficients for each parameter can be efficiently calculated in the order from the layer on the output side to the layer on the input side (in the opposite direction of input-output). Note that, if the error $e\left(\omega_i;\theta\right)$ with respect to individual training data samples is differentiable with respect to the parameter θ (i.e., if the analytical solution of the derivative exists), the method becomes more efficient.

Additionally, various techniques such as skip connections, momentum methods, automatic learning rate adjustments, regularization layers, data augmentation, and dropout have been introduced to address the issue of not converging to global optima. Notably, the Adam method proposed by Kingma and Ba [2015] combines the principles of SGD, momentum, and RMSprop, representing one of the most widely used learning algorithms. For detailed insights, the readers are referred to the aforementioned literature.

3.1.4 Natural performance of neural networks

As discussed in section 3.1.2, theoretical research on neural networks has progressed in recent years, gradually revealing the reasons why deep learning works so effectively. Specifically, it has become evident that neural networks possess desirable properties in terms of representation, optimization, and generalization abilities. This section provides an overview of these properties.

Firstly, representation ability refers to how well a neural network can approximate the function it aims to represent. Studies on representation ability have a long history, with the universal approximation property of neural networks (i.e., functions that are important for applications can generally be approximated by neural networks) being particularly renowned.

[19] If one can select a sample of sufficient size from the training data, using only a subset of the training data can significantly reduce the computational cost of $\nabla J(\theta)$. Additionally, by stochastically sampling the training data, the problem of vanishing gradients is less likely to occur.

[20] As mentioned in section 4.1.2, in deep hedging where the risk measure is learned as a loss function, efficient learning can be achieved by representing the loss function as a sum over samples of the training data, as given in equation (7), and applying techniques such as SGD and backpropagation.

Recent research has highlighted the *adaptive approximation* capability of neural networks, demonstrating that they can effectively adjust the resolution based on the input, thereby efficiently approximating functions (Suzuki [2018]). Intuitively, adaptive approximation means that the network can adjust its resolution depending on the input, especially increasing the resolution where the function has large variations.

Secondly, optimization ability relates to the predictive power on training data and the capability to explore global solutions θ^*. As previously mentioned, $J(\theta)$ is a high-dimensional function, making it susceptible to becoming stuck in local minima or saddle points. However, neural networks exhibit favorable properties in the shape of the objective function $J(\theta)$, making such issues less likely. For instance, Garipov et al. [2018] experimentally demonstrated the connectedness of regions containing optimal solutions, and Kawaguchi [2016] theoretically showed that if the activation functions are restricted to linear functions, all local minima are also global minima.

Lastly, generalization ability pertains to the predictive power on unseen data, ensuring that the network does not overfit to training data and can adaptively represent functions based on problem complexity. Neural networks are known to possess an implicit regularization capability that adjusts model complexity based on problem complexity. Concrete examples of this include *norm minimization* (Hastie et al. [2022]), *flat minima* (Soudry et al. [2018]), and the *lottery ticket hypothesis* (Frankle and Carbin [2018]). This automatic adjustment is realized, for instance, by certain parameters θ^* approaching zero, effectively nullifying some input values.

3.2 Neural network notation

Definition 4. *Let $L \in \mathbb{N}$ with $L \geq 2$ be the number of layers in a neural network, and let $N_0, N_1, \ldots, N_L \in \mathbb{N}$ denote the dimensions of the intermediate layers. Let $W_l : \mathbb{R}^{N_{l-1}} \to \mathbb{R}^{N_l}$ represent an affine transformation. In other words, there exist matrices $A^l \in \mathbb{R}^{N_l \times N_{l-1}}$ and vectors $b^l \in \mathbb{R}^{N_l}$ such that $W_l(x) = A^l x + b^l$. Further, let $\eta : \mathbb{R} \to \mathbb{R}$ be an activation function.*

Then, the composite function $F^\theta : \mathbb{R}^{N_0} \to \mathbb{R}^{N_L}$ defined by the following is referred to as a (feed-forward) neural network:

$$F^\theta(x) = W_L \circ F_{L-1} \circ \ldots \circ F_1,$$

where $F_l = \eta \circ W_l$ (with η applied element-wise). Here, θ represents the parameters of the neural network, that is, the set of components of matrices A^l and vectors b^l for $l = 1, \ldots, L$.

Figure 2 shows a schematic diagram of a neural network. Each layer is composed of an affine function followed by a non-linear activation function. By stacking these layers L times, the function operates on input data x by alternating between linear and non-linear transformations, producing an output y.

Hereafter, we denote the set of feed-forward neural networks with input dimension N_0, output dimension N_L, parameter dimension M as $\mathcal{NN}_{M;N_0,N_L}$, and the set of parameters as

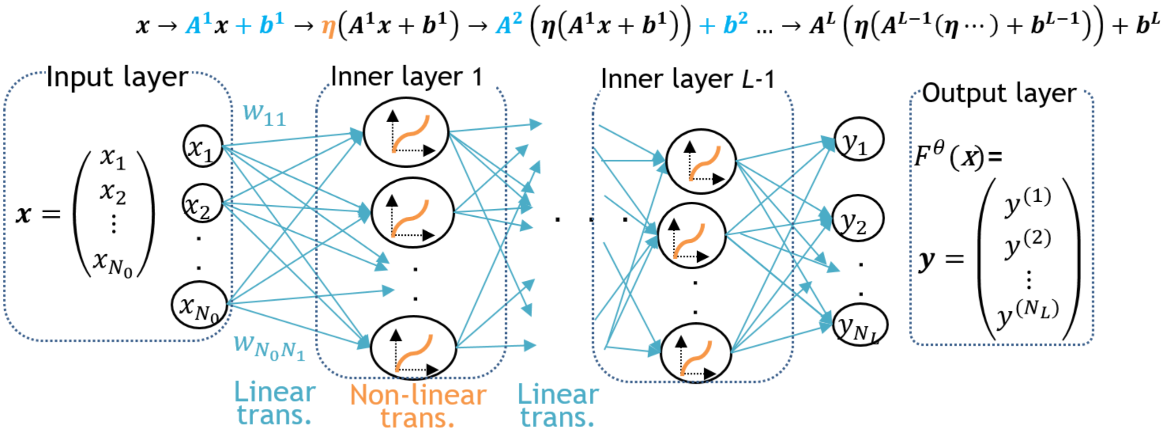

Figure 2: Neural Network

Θ_M.[21]

4 Deep hedging

In this section, we introduce the framework of deep hedging presented by Buehler et al. [2019a]. Deep hedging is a framework that utilizes deep learning to numerically solve convex risk minimization problems. As mentioned in section 2, it bypasses risk-neutral valuation and directly solves the convex risk minimization problem. This approach has gained attention due to its ability to quantify the impact of market frictions such as trading costs.[22]

In section 4.1, we introduce the framework of deep hedging proposed by Buehler et al. [2019a]. In section 4.2, we discuss the subsequent research trends in this area. In section 4.3, we conclude with future prospects.

4.1 Algorithm proposed in Buehler et al. [2019a]

Deep hedging consists of a series of algorithms to determine an appropriate hedging strategy using a neural network. A hedging strategy is first modeled by a neural network. The model is then trained to minimize the loss risk using deep learning. The learning results are used to determine the hedging strategy. Using the notation introduced in section 2, we can formulate this procedure as shown below. For simplicity, we consider an underlying asset (hedging instrument) price process $\{S_k\}_{k=0,\dots,n-1}$ that is a Markov process; that is, the future price depends only on the current price and not on past prices.

[21]The dimension M is given as $M = \sum_{l=1}^{L}(N_{l-1} \times N_l + N_l)$ because it is the sum of the numbers of the elements of matrix A^l and vector b^l for $l = 1, \dots, L$.

[22]According to *Risk Magazine*, an industry publication, some financial institutions appear to be actively utilizing deep hedging. Furthermore, Hans Buehler (XTX Markets), one of the authors of Buehler et al. [2019a], was awarded *Quant of the year 2022* for this achievement (Risk.net [2022]).

Hedging strategies are modeled using the following neural network functions:

$$\begin{cases} \hat{\delta}_0^\theta = F^{\theta_0}(0, S_0), \\ \hat{\delta}_k^\theta = F^{\theta_k}\left(\hat{\delta}_{k-1}^\theta, S_k\right) \quad (k = 1, \ldots, n-1). \end{cases} \tag{9}$$

Here, $F^{\theta_k} \in \mathcal{NN}_{M;2,1}$ (with two inputs $\hat{\delta}_{k-1}^\theta, S_k$ and one output $\hat{\delta}_k^\theta$) typically consists of a shallow neural network with two to three layers. We assume the parameters $\theta_k \in \Theta_M$ for each neural network F^{θ_k} describing the hedging strategy are distinct for each period. The neural network is trained using data across all periods. Furthermore, the parameters describing the entire hedging strategy up to maturity are denoted as $\theta = (\theta_0, \ldots \theta_{n-1}) \in \prod_{k=0}^{n-1} \Theta_M$. In essence, the hedging strategy of each period is modeled by a shallow-layer neural network whose input data are the hedge strategy in the previous period and the current underlying asset price (as shown in one block in Figure 3), and the strategy across periods is modeled by a recursive deep-layer neural network (as shown in the entire Figure 3). In section 4.1.1, we explain the validity (convergence) of modeling the hedging strategy by such a neural network to minimize the loss risk.

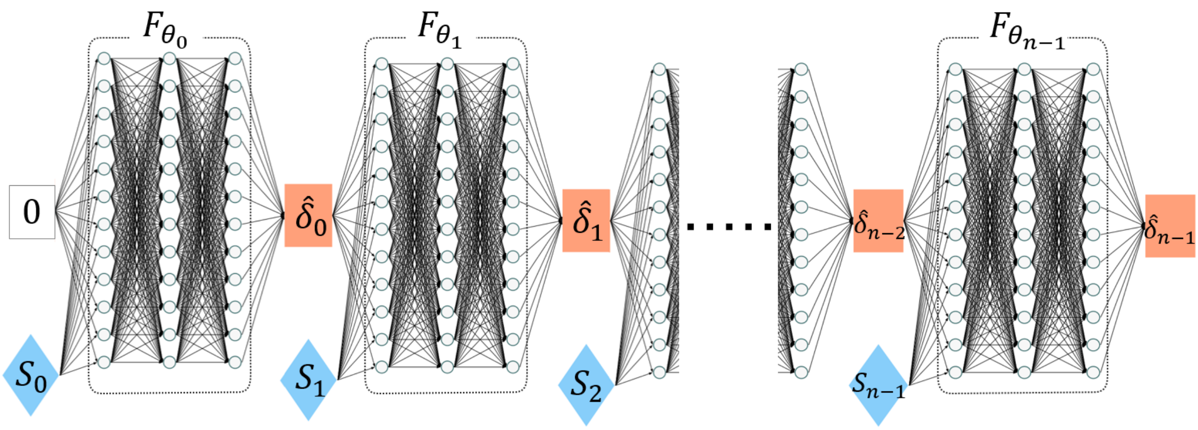

Figure 3: Modeling of hedging strategies in deep hedging

Next, in the learning of parameters θ, given the future scenarios of $\{S_k\}_{k=0,\ldots,n}$ (probability distribution and its realization values), $\{S_k(\omega_i)\}_{k=0,\ldots,n, \, i=1,\ldots,q}$, the P&Ls $\left\{\mathrm{PL}^{(Z_T, p_0, \delta^\theta)}(\omega_i)\right\}_{i=1,\ldots,q}$ for scenarios $i = 1, \ldots, q$ are calculated. Then, we calculate the optimal θ^* that minimizes the loss risk $\rho\left(\left\{\mathrm{PL}^{(Z_T, p_0, \delta^\theta)}(\omega_i)\right\}_{i=1,\ldots,q}\right)$. Here, the deep learning technique introduced in section 3.1.3 is used. Importantly, the loss function needs to be represented in an appropriate form, which is discussed in section 4.1.2.

In the execution of the hedging, the realized asset prices $\{S_k(\widetilde{\omega})\}_{k=0,\ldots,n}$ are sequentially fed into the trained neural network $F^{\theta_k^*}$. Thus, we produce the hedging strategy $\left\{\hat{\delta}_k^{\theta^*}\right\}_{k=1,\ldots,n-1}$ by deep hedging as follows:

$$\begin{cases} \hat{\delta}_0^{\theta^*} = F^{\theta_0^*}(0, S_0), \\ \hat{\delta}_k^{\theta^*} = F^{\theta_k^*}\left(\hat{\delta}_{k-1}^{\theta^*}, S_k(\widetilde{\omega})\right) \quad (k = 1, \ldots, n-1). \end{cases}$$

4.1.1 Approximation theorem

In the framework described above, the process is described, for simplicity, for the case where the asset price $\{S_k\}_{k=0,\dots,n}$ follows a one-dimensional Markov process. In this subsection, following Buehler et al. [2019a], we discuss the validity (approximation theorem) of modeling hedging strategies using neural networks for more general $\mathcal{F}_k$-adapted $\mathbb{R}^d$-valued stochastic processes $\{S_k\}_{k=0,\dots,n}$.

Before presenting the approximation theorem, additional notation and assumptions are required. Firstly, the transaction cost function c_k in the $\mathrm{PL}^{(Z_T,p_0,\delta)}$(equation (1)) is assumed to be right-continuous; that is, for $x \in \mathbb{R}$, $\lim_{x\downarrow a} c_k(x) = c_k(a)$ holds.[23] Additionally, $\{I_k\}_{k=0,\dots,n-1}$ represents market information at each time and I_k is assumed to generate $\mathcal{F}_k$. Moreover, let $\mathcal{H}$ be a subset of $\mathbb{R}^d$-valued stochastic processes representing possible hedging strategies.[24] $\mathcal{H}^M$ is defined as the set of hedging strategies expressible by neural networks $\mathcal{NN}_{M;N_0,N_1}$ as follows:

$$\mathcal{H}_M = \left\{ \left\{\hat{\delta}_k^{\theta}\right\}_{k=0,\dots,n-1} \mid \hat{\delta}_k^{\theta} = F^{\theta_k}\left(\delta_{k-1}, I_0, \dots, I_k\right),\ \theta_k \in \Theta_M, F^{\theta_k} \in \mathcal{NN}_{M;r(k+1)+d,d} \right\}.$$

Note that when $\{S_k\}_{k=0,\dots,n}$ is a Markov process, this definition aligns with equation (9). Let X be an $\mathcal{F}_n$-measurable random variable, and define

$$\pi(X) = \inf_{\delta\in\mathcal{H}} \rho\left(\mathrm{PL}^{(X,0,\delta)}\right), \quad \pi_M(X) = \inf_{\delta\in\mathcal{H}_M} \rho\left(\mathrm{PL}^{(X,0,\delta)}\right).$$

Here, $\pi(X)$ represents the risk when choosing the optimal hedging strategy from all possible strategies, while $\pi_M(X)$ represents the risk when choosing the optimal strategy from the set of strategies expressible by neural networks.

Theorem 1 (Approximation theorem for deep hedging)**.**

$$\lim_{M\to\infty} \pi_M(X) = \pi(X).$$

The proof of Theorem 1 can be established using the universal approximation theorem of neural networks under assumptions such as the right-continuity of c_k. For more details, refer to Proposition 4.2 in Buehler et al. [2019a]. It is worth mentioning that this theorem only guarantees convergence; showing the precision or rate of convergence analytically remains challenging.

Theorem 1 indicates that the risk $\pi_M(X)$ minimized within the hedging strategy $\mathcal{H}_M$ approximated by a neural network converges to the risk $\pi(X)$ minimized within the set of possible hedging strategies $\mathcal{H}$ as the number of parameters M (parameters describing the neural network for a one-period hedging strategy) increases. This observation suggests that model-

[23] In this case, S_k for $k = 0, \dots, n$ is considered as part of the market information ($\{S_k\}_{k=0,\dots,n}$ generating a σ-algebra contained within the σ-algebra generated by $\{\mathcal{F}_k\}_{k=0,\dots,n}$).

[24] In section 2.2 of Buehler et al. [2019a], the set of possible hedging strategies is defined, taking into account the transaction constraints. For more details, refer to the original paper.

ing hedging strategies using neural networks to minimize loss risk is a viable approach.[25]

4.1.2 A key in learning neural network weight parameters

To efficiently learn the weight parameters θ^* that minimize the loss function $J(\theta)$, techniques such as SGD and backpropagation, as introduced in section 3, are employed. It is desirable that the loss function $J(\theta)$ be the sum of errors concerning a single training data sample and that it be differentiable with respect to θ. Therefore, in the deep hedging, one considers a class of risk measures for which the loss function can be represented as the expected value of a random variable $J(\theta) = \rho\left(\left\{\mathrm{PL}^{(Z_T,p_0,\delta^\theta)}(\omega_i)\right\}_{i=1,\ldots,q}\right)$.[26]

Attempts to represent risk measures such as ES as expected values began with Rockafellar et al. [2000]; and were further developed by Ben-Tal and Teboulle [2007] through the introduction of the concept of Optimized Certainty Equivalence (OCE).[27] OCE is formulated as follows: Let $l : \mathbb{R} \to \mathbb{R}$ be a continuous, non-decreasing, convex function, and for a real-valued random variable X,

$$\kappa(X) = \inf\left\{w + \mathbb{E}[l(-X - w)]\right\}, \quad w \in \mathbb{R}.$$

Then, κ satisfies the definition of a convex risk measure and is called a risk measure with OCE. For instance, ES can be expressed by setting $l(x) = (1/(1-\alpha))\max\{x, 0\}$ as below, and thus ES has the property of OCE:

$$\mathrm{ES}_\alpha(X) = \inf_{w\in\mathbb{R}}\left\{w + E\left[\frac{1}{1-\alpha}\max\{-X - w, 0\}\right]\right\}.$$

A risk measure with OCE can be efficiently learned. Namely, using the representations via OCE, we transform the argument in optimization from θ to a two-dimensional variable $\theta \times w$. Using the formulation of OCE, efficient learning is possible by transforming the variable of the minimization problem from θ to $\theta \times w$:

$$\begin{aligned}\inf_\theta \rho\left(\mathrm{PL}^{(Z_T,p_0,\hat{\delta}^\theta)}\right) &= \inf_\theta \inf_{w\in\mathbb{R}}\left\{w + E\left[l\left(\mathrm{PL}^{(Z_T,p_0,\hat{\delta}^\theta)} - w\right)\right]\right\} \\ &= \inf_{\bar{\theta}=\theta\times w}\left\{w + E\left[l\left(\mathrm{PL}^{(Z_T,p_0,\hat{\delta}^\theta)} - w\right)\right]\right\}.\end{aligned}$$

By this transformation, the loss function for learning is set as

$$J\left(\bar{\theta}\right) = \left\{w + \sum_{i=1}^{q}\left[l\left(\mathrm{PL}^{(Z_T,p_0,\hat{\delta}^\theta)}(\omega_i) - w\right)\right]P(\omega_i)\right\}.$$

As mentioned in section 3.1.3, representing the loss function in this form allows the use of

[25] In Buehler et al. [2019a], it is not directly stated that the modeled hedge strategy $\left\{\hat{\delta}_k^\theta\right\}_{k=0,\ldots,n-1}$ converges to the solution δ^* of the convex risk minimization problem. As mentioned in Remark 1, the reason for this is to avoid discussing the existence of hedge strategies that minimize risk.

[26] It is known that Value-at-Risk cannot be represented as the expectation of a random value.

[27] In decision theory, OCE was introduced as an extension of certainty equivalence.

SGD. Moreover, if the derivative with respect to θ on the right-hand side is analytically available, it can be used to accelerate backpropagation (refer to Buehler et al. [2019a] Proposition 4.6).

4.1.3 Summary of numerical results

In section 5 of Buehler et al. [2019a], the authors provide empirical evidence of the validity of the learning outcomes through numerical experiments. Here, we present a summary of this evidence.

Setup

- Target option: European Call Option (1-month maturity, at-the-money strike) $Z_T = \max\left\{S_T^{(1)} - K, 0\right\}, \quad T = \frac{30}{365}$.
- Distribution for hedging instruments: Heston model (Heston [1993])

$$\begin{cases} d\widetilde{S}_t^{(1)} = \sqrt{\widetilde{S}_t^{(2)}}\widetilde{S}_t^{(1)} dW_t^{(1)}, \\ d\widetilde{S}_t^{(2)} = \alpha\left(b - \widetilde{S}_t^{(2)}\right) dt + \sigma\sqrt{\widetilde{S}_t^{(2)}}\left(\rho dW_t^{(1)} + \sqrt{1-\rho^2} dW_t^{(2)}\right), \\ \left(\widetilde{S}_0^{(1)}, \widetilde{S}_0^{(2)}\right) = \left(s_0^{(1)}, s_0^{(2)}\right), \end{cases} \tag{10}$$

  with parameters set as $\alpha = 1$, $b = 0.04$, $\rho = -0.7$, $\sigma = 2$, $s_0^{(1)} = 100$, and $s_0^{(2)} = 0.04$.
- Daily hedging: $t_k = \frac{k}{365}, n = 30$
- Risk measure: ES
- Generation of future scenarios: By discretizing equation (10) and using Monte Carlo simulation, generate q future scenarios $\left\{\left(S_i^{(1)}(\omega_j), S_i^{(2)}(\omega_j)\right)\right\}_{\substack{i=1,\dots,n \\ j=1,\dots,q}}$.
- Learning neural network weights: Using a learning rate $\nu_i = 0.0005$ and batch size $q_B = 256$, compute the optimal weights θ^* that minimize $\rho\left(\left\{\mathrm{PL}^{(Z_T, p_0, \delta^\theta)}(\omega_i)\right\}_{i=1,\dots,q}\right)$ via Adam optimization.

Results

1. Comparison of deep hedging and delta hedging (no transaction costs)
   Figure 2 in Buehler et al. [2019a] shows the frequency distribution of the realized P&Ls for both deep hedging and delta hedging in no-transaction-cost setting.[28] Here, the risk measure is set as 50%-ES. The graph shows that the behavior of the realized P&Ls in deep hedging closely aligns with the P&Ls in delta hedging in the case of no

[28]This is the frequency distribution obtained from an out-of-sample test with a sample size $q' = 10^6$, namely, the frequency distribution of the P&Ls when applying the hedging strategy derived from the learning outcomes under scenarios $\{S_k(\widetilde{\omega}_i)\}_{\substack{k=0,\dots,n \\ i=1,\dots,q'}}$ as distinct from those used for learning.

transaction costs.[29] In addition, the greeks are also closely align.[30] This indicates the validity of deep hedging in the case of no transaction costs.[31]

2. Impact of risk measure parameter setting on hedging strategy learning
Figure 7 in Buehler et al. [2019a] displays the frequency distribution of the realized P&Ls for deep hedging when the risk measures are set as 50% and 99%-ES, respectively. From the graph in the upper part of this figure, in the case of 50%-ES, the distribution of the realized P&Ls is higher around the median; that is, the hedging strategies are trained to avoid losses below the mean. The table at the bottom of the figure also shows that in the case of 99%-ES, the distribution of the realized P&Ls is smaller than the case of 50%-ES; that is, hedging strategies that avoid large losses are learned. These facts indicate that the learning outcome of deep hedging changes as expected depending on the parameter setting of the risk measure.

3. Impact on utility-indifference pricing due to transaction costs
Figure 11 in Buehler et al. [2019a] shows the utility-indifference prices p_ε for different ε, setting transaction cost as $c_k(\delta_k - \delta_{k-1}) = \varepsilon \sum_{i=1}^{2} \left|\delta_k^{(i)} - \delta_{k-1}^{(i)}\right| S_k^{(i)}$. The graph indicates that
$$p_\varepsilon - p_0 = O\left(\varepsilon^{0.71}\right)$$
holds. This is consistent with the theoretical results (Whalley and Wilmott [1997], Rogers [2004]) for the case in which the price process of the underlying asset follows a one-dimensional diffusion process such as the Black–Scholes model. This fact indicates that the utility-indifference prices calculated through deep hedging vary with respect to the transaction costs, as theoretically predicted, and that the hedging error converges with a reasonable degree of accuracy.

4.2 Research trends

4.2.1 Future scenario generation

In section 4.1, we presented the framework of deep hedging, without specifying how the future scenarios of the underlying assets (hedging instruments) are generated.[32] Developing methods for generating future scenarios is a central issue in the subsequent research on deep hedging, since generating realistic future scenarios is essential for the practical implementation of deep hedging.

[29] The horizontal axis of this graph represents the hedging error as a percentage of one unit of the underlying asset. As mentioned above, this is because the current price of the underlying asset is set to $s_0^{(1)} = 100$.

[30] Refer to Figure 3 in Buehler et al. [2019a].

[31] As explained in section 2, in the case of no transaction costs, a delta hedge can eliminate hedging errors through replication. However, due to minimal numerical computation errors, the frequency distribution of the realized P&Ls from the delta hedge also has a slight spread.

[32] In Buehler et al. [2019a], future scenarios are generated using a stochastic differential equation with fixed parameters.

Among the various methodologies, techniques employing generative models based on deep learning have been extensively investigated. For instance, Wiese et al. [2019] proposed a method to generate future scenarios for vanilla options[33] using Generative Adversarial Networks (GANs).[34] When evaluated against the quasi-maximum likelihood method and vector autoregressive models, the scenarios generated by GANs better captured the characteristics of the given past data across multiple metrics. Additionally, Buehler et al. [2020, 2021a] proposed an approach using a deep generative model called the *variational autoencoder* and features termed *signatures* to generate future scenarios from limited market data.[35]

It is apparent that if we use a scenario that gives a hedging strategy where the expected return genuinely becomes positive upon increasing the number of scenario samples, that is, a statistical arbitrage strategy, the hedging strategy learned via deep hedging can be significantly influenced by this arbitrage strategy. This influence may result in a biased hedging strategy, such as monotonically increasing the holdings of specific assets. Buehler et al. [2022a, 2021b] extended the notion of risk-neutral valuation in the presence of transaction costs and introduced a method to remove statistical arbitrage from future scenarios. Essentially, this method transforms the given future scenarios to ensure the absence of hedging strategies with genuinely positive expected returns. Moreover, numerical results demonstrate that by eliminating statistical arbitrage from the future scenarios, the performance of hedging strategies trained via deep hedging improves; that is, hedging errors are reduced.

Furthermore, in empirical studies, Zhang and Huang [2021] and Mikkilä and Kanniainen [2021] reported numerical experiments using historical data for option prices to generate future scenarios. Horvath et al. [2021b] discussed scenarios that follow non-Markovian stochastic processes, that is, the rough volatility model (Gatheral et al. [2018]). Additionally, Lütkebohmert et al. [2021] examined scenarios governed by stochastic differential equations with uncertain coefficients.

4.2.2 Validation of deep hedging under different settings

In this paper, following the framework of Buehler et al. [2019a], we formulate the hedging problem as a minimization problem of the risk quantified by convex risk measures of the P&Ls at maturity, and use utility indifference pricing. In practice, it is essential to modify and extend the formulation of the hedging problem depending on the specific situation or objective.

Kolm and Ritter [2019] and Cao et al. [2021] formulated the risk measure using the mean and variance of the P&Ls at maturity as an optimization objective. Additionally, Carbonneau [2021] incorporated hedge errors as the optimization objective without resorting to

[33] These are basic derivatives such as European options.

[34] This is the generative model of deep learning proposed by Goodfellow et al. [2020]. The method uses neural networks to generate pseudo datasets with features similar to the input data. By training two networks for generation and discrimination in competition, high-quality datasets can be generated.

[35] Signatures are features that generalize high-order correlation structures between the prices of multiple assets.

convex risk measures. The conditions of the derivatives change daily due to factors such as shortening expiration times. Buehler et al. [2022b] considered this aspect and proposed a framework to formulate the hedging problem using the Bellman equations and solve it using the actor-critic method of reinforcement learning.

Moreover, Carbonneau and Godin [2021] discussed the concept of Equal Risk Pricing as an alternative to utility indifference pricing (Definition 3).[36]

Note that even before Buehler et al. [2019a], researchers had explored the derivation of hedging strategies using deep learning and reinforcement learning, under specific conditions such as the absence of transaction costs. For instance, Halperin [2019] proposed a method to derive hedging strategies under the Black–Scholes model in a cost-free setting using the Q-learning framework.[37] Specifically, the author formulated the replication via adjustments in the portfolio, the basic idea of the Black–Scholes model, in a discrete time setting. Using the Markov decision process in reinforcement learning, the author then proposed a method for formulating a hedging strategy as a solution to the dynamic optimization problem and calculating an optimal hedging strategy analytically. In Halperin [2019], the author discussed the numerical computations for this method. Notably, research works such as Mikkilä and Kanniainen [2021] and Buehler et al. [2019b] have adopted the formulation using Q-learning.

4.2.3 Numerical experiments for more complex derivatives

Whereas the numerical experiments in Buehler et al. [2019a] were conducted on relatively simple short-dated derivatives, there are a number of studies on more complex derivatives. For example, Carbonneau and Godin [2021] and Imaki et al. [2021] dealt with exotic derivatives such as look-back options and Asian options. Carbonneau [2021] dealt with long-dated derivatives. In another study, Shi et al. [2021] compared the method proposed in Buehler et al. [2019a] with the method based on numerical solutions of forward-backward stochastic differential equations (Weinan et al. [2017]), and concluded that the method in Buehler et al. [2019a] is more effective for long-term optimization problems.

4.2.4 Modifications in modeling hedging strategies

In Buehler et al. [2019a], a method was introduced to model hedging strategies using feedforward neural networks as described in equation (9). As discussed in section 4.2.3, it is considered that this model can adequately learn hedging strategies for specific problems. However, for more intricate problems, where an efficient derivation of hedging strategies is required, it is beneficial to modify the hedging strategies model.

[36]This is the price formulated by Guo and Zhu [2017], where the risk amount for the seller of a derivative equals that of the buyer. Generally, while utility indifference prices may differ between the seller and the buyer, risk-equivalent prices are the same for both sides.

[37]Q-learning is a method in reinforcement learning that represents future cumulative rewards as a function of the current environment's state and action and estimates this function.

Zhang and Huang [2021] presented a model using Long Short Term Memory Recurrent Neural Networks (LSTM-RNNs) that utilizes historical hedging strategy data to determine future-period hedging strategies. Through numerical experiments, they verified that the hedging error in P&Ls is improved, compared to classical methodologies like Leland [1985]. Similarly, Carbonneau [2021] adopted the use of LSTM-RNNs and discussed long-term hedging for derivatives in settings where the underlying asset price process includes jumps.

Tawada and Sugimura [2020] proposed a technique using Gaussian Process Regression (GPR) rather than neural networks for modeling hedging strategies. GPR offers the advantage of high interpretability for the trained parameters, rendering the derived hedging strategies more comprehensible. Making use of this property, the authors proposed a method to transition to alternative hedge techniques (such as greeks hedging) when the scenarios used for learning significantly deviate from the realized scenarios. Imaki et al. [2021] proposed a model that includes the additional hypothesis that trading does not occur if the trading volume remains below a certain threshold and offered a method to estimate this threshold by using deep learning. The authors not only theoretically demonstrated that the hedging strategy under this model becomes optimal even with its approximation, but they also showed a significant reduction in the computational costs required for learning hedging strategies in numerical experiments. Additionally, Son and Kim [2021] conducted numerical experiments utilizing various deep learning models, including regression neural networks and convolutional neural networks, among others.

4.2.5 Application to general risk management

Since the convex risk minimization problem is a general optimization problem given P&L measures, we can apply the same solution approaches not only to derivative hedging but also to various risk management situations. Consequently, there is a growing number of discussions on the application of deep hedging frameworks to broader risk management problems. For instance, the application of deep hedging to risk management in the banking sector has been discussed in Krabichler and Teichmann [2020], while its application to risk management in insurance companies is explored in Fernandez-Arjona and Filipović [2022]. Additionally, Shimizu [1998] utilized neural networks within a supervised learning framework to consider feedback effects in stress tests. It seems possible to conduct such stress tests based on deep hedging.

4.3 Future prospects

4.3.1 Development potential

Deep hedging offers a framework for determining optimal trading strategies given the amount of P&L risk. Due to its flexibility, it can be applied to a wide range of problems, including derivative hedging and risk management. Below, we summarize its development potential in the field of derivatives and elsewhere.

In the field of derivatives, both the refinement and automation of hedging strategies are expected. Regarding refinement, deep hedging makes it possible to incorporate market frictions such as transaction costs. Additionally, by considering information other than financial market observations—for example, news—in generating future scenarios for hedging instruments, we can reflect such information in the hedging strategy. Importantly, by setting risk measures according to the risk preference of the entity hedging the derivative, a strategy tailored to that risk preference can be derived. These could represent new analytical methods in derivative risk management. The automation of hedging operations may also be advanced by the refinement of hedging strategies. As discussed in section 2.1.2, in hedging strategies derived via risk-neutral valuation, adjustments such as additional cost charges are usually applied to cover inevitable errors in replicating the derivative. Such price adjustments are one of the main obstacles to automating hedging operations, as the size of the price adjustments are often determined in an ad hoc manner based on the subjective judgements of the traders. On the other hand, by virtue of deep hedging, such qualitative adjustments become unnecessary because deep hedging can directly take into account market frictions, thus making the automation of hedging operations more feasible. Additionally, by using deep learning models for future scenario generation, it becomes possible to repeatedly perform the learning of hedging strategies and scenario generation, which opens the possibility of deriving hedging strategies purely from data without any model dependency.

Outside the field of derivatives, as discussed in section 4.2.5, applications to general risk management and broader decision-making based on data such as corporate management strategy become feasible. However, compared to the hedge problems in the field of derivatives, the formulation and validation of such broader problems represent a bigger challenges. Thus, it would seem appropriate that we first develop the techniques of deep hedging for derivatives.

4.3.2 Practical challenges

As noted above, discussions of the challenges associated with deep hedging is ongoing, as the method is still quite novel. Below, we summarize the practical challenges of deep hedging by categorizing the problem settings of deep hedging and the technical nature of deep learning.

As discussed in section 4.2.2, for the practical use of deep hedging, expanding or modifying the problem settings according to the specific situation or purpose is necessary. Especially since there is no established method for setting the parameters of the risk measures, comparing strategies learned under different risk measure parameters would seem practically useful for risk management. Moreover, when calculating derivative prices using deep hedging (especially for counterparty transactions or fair value calculations), one needs to be careful since external parameters (like risk measure parameters) can significantly affect the price.

In addition to the extension and modification of the problem settings mentioned in section 4.2.2, there are other settings closely related to the practical application of deep hedging. For instance, optimizing not only the risk amount at maturity T of the derivative but also the risk amount at intermediate points $t \in [0, T]$ is, in practice, considered essential. Additionally, incorporating value adjustments called xVAs, which have become important in derivative practices post the 2007-08 financial crisis, into the framework of deep hedging is another significant challenge. Operationally, it is crucial to re-evaluate the hedging strategy after significant changes in one's portfolio or market environment.

Regarding deep learning techniques, as mentioned in section 3, compared to established financial engineering techniques such as the Monte Carlo method, there is no theoretical guarantee regarding numerical behavior, such as the accuracy of learning. Hence, we should be careful in practical applications. Moreover, similar to applying deep learning in other fields, enhancing the speed and stability of learning and increasing the interpretability of results present serious challenges.

In summary, while deep hedging has significant development potential, it also poses numerous challenges. Thus, it is crucial that we explore how best to utilize deep hedging by comparing it with the already established hedging approach based on risk-neutral valuation. Especially when one takes a model-based approach in practical applications, it is critically important to understand the underlying data and calculation principles and to use the model-derived results cautiously. Table 1 summarizes the relationship between deep hedging and hedging based on risk-neutral valuation.

Table 1: Comparison of deep hedging and risk-neutral valuation

	Deep hedging	Hedging via risk-neutral valuation (Greeks hedging)
Principle	Convex risk minimization (Incomplete market) — Physical measure	Replication / No-arbitrage (Risk neutral valuation) — Risk neutral measure
Main inputs	1. Future market scenario 2. Risk preference (Risk measure)	Information on the current market environment (Market prices of liquid assets)

5 Deep calibration

In this section, we introduce deep calibration (Hernandez [2017], Horvath et al. [2021a], etc.). As noted in section 2.3, the calibration process is computationally intensive and prone to problems such as with convergence issues in the optimization calculations. Deep calibration is an algorithm that trains a neural network to learn the relationship between the model parameters and model prices in advance, and then replaces the optimization calculations of the model parameters which are a step in the pricing calculations (risk-neutral valuation). This approach is expected to accelerate and stabilize the pricing calculations.

In section 5.1, we introduce the deep calibration algorithm proposed in Hernandez [2017]. In section 5.2, we discuss the trends in subsequent research. We then summarize, in section 5.3, the practical advantages and challenges.

5.1 Algorithm proposed in Hernandez [2017]

Deep calibration is executed in the following two steps: First, a neural network is trained by taking the model prices of the calibration target products as the input and the model parameters as the output. Then, in the daily pricing tasks, the market prices of the calibration target products are input into the trained neural network to approximate the model parameters (Figure 4). Using the notation from section 2.3, we describe the algorithm below. Note that the algorithm belongs to the supervised learning classification as described in section 3.1.1.

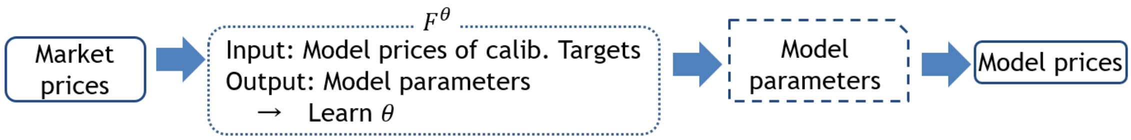

Figure 4: Process of deep calibration

Here, we consider the problem of determining model parameters $\sigma \in R^Q$ with given R calibration target product market prices $\{\text{QP}(\tau_i)\}$ for $i = 1, ..., R$.

1. Training data generation
   Randomly generate external factors $\hat{\phi}_j \in \mathbb{R}^m$ and model parameters $\hat{\sigma}_j \in \mathbb{R}^Q$.[38] Next, compute the model price $\left\{\text{MP}\left(\hat{\sigma}_j; \tau_i, \hat{\phi}_j\right)\right\}_{i=1,...,R}$. Repeat this q times to generate training data $\left\{\hat{\sigma}_j, \hat{\phi}_j, \text{MP}\left(\hat{\sigma}_j; \tau_i, \hat{\phi}_j\right)\right\}_{\substack{i=1,...,R \\ j=1,...,q}}$.

2. Learning
   Train $K^\theta \in \mathcal{NN}^\theta_{M;m+R,Q}$ with inputs $\left(\hat{\phi}_j, \left\{\text{MP}\left(\hat{\sigma}_j; \tau_i, \hat{\phi}_j\right)\right\}_{i=1,...,R}\right) \in \mathbb{R}^{m+R}$ for $i = 1, ..., R$ and output $\hat{\sigma}_j$. For instance, minimize the loss function defined as

$$J(\theta) = \sum_{j=1}^{q} \left(\hat{\sigma}_j - K^\theta\left(\hat{\phi}_j, \text{MP}\left(\hat{\sigma}_j; \tau_i, \hat{\phi}_j\right)\right)\right)^2$$

   to determine θ^*.

[38] When generating external factors and model parameters randomly, upper and lower bounds are determined by referencing past model parameters and market data.

3. Application
   In the daily valuation of derivatives, given the market price $\{\mathrm{QP}_i\}_{i=1,\ldots,R}$ of the calibration target product and external factors $\phi \in \mathbb{R}^m$, use $K^{\theta^*}\left(\phi, \{\mathrm{QP}_i\}_{i=1,\ldots,R}\right)$ as model parameters.

Hernandez [2017] applied the above algorithm to the Hull-White model (with swaptions as calibration targets) and compared it with conventional calibration methods using historical data. For the setup of this numerical experiment, refer to Table 2. Additionally, Hernandez [2017] reported that in the generation of pairs $\hat{\phi}_j, \hat{\sigma}_j$, the calculations become stable if the correlations between model parameters and external factors using past data are taken into account.

Table 2: Numerical examples of deep calibration

Reference	Model	Calibration targets	Setting of neural networks	Acceleration
Hernandez [2017]	Hull–White model - Num params. $Q = 2$	Swaption 6M-Libor - Num targets $R = 200$	Num layers $L = 5$ (4 interm. layers) Dim of interm. layers $N_l = 64$ Num training data $q = 150,000$	Around 225 times faster
Horvath et al. [2021a]	(Rough) Bergomi model - Num params. $Q = 8$	Volatility surface - Num targets $R = 88$	Num layers $L = 5$ (4 interm. layers) Dim of interm. layers $N_l = 30$ Num training data $q = 68,000$	Around 9,000 to 16,000 times faster

Note: For more details, refer to the indicated references.

5.2 Research trends

5.2.1 Improvements in the algorithm

Horvath et al. [2021a], Bayer et al. [2019], and Liu et al. [2019a] proposed a method, called here the two-step method, to enhance the speed and stability of the algorithm in Hernandez [2017]. This method consists of an algorithm that trains a neural network to learn the forward function $\mathrm{MP}(\cdot; \tau_i, \phi)$ of the model price and then calibrates the model parameters using conventional optimization methods. According to Horvath et al. [2021a] and Bayer et al. [2019], the primary advantage is that by training the neural network on the forward function of the model price, the results are easier to validate and interpret relative to the case where the inverse function is trained.[39]

Subsequently, Itkin [2020] modified a part of the algorithm in Hernandez [2017] by introducing a method that uses the forward function of the model price for training data generation, akin to the two-step method. Itkin [2020] pointed out that by using a method which ensures that the function maintains the no-arbitrage condition and smooth greeks during the learning of the price function's forward function, stable and rapid calibration becomes feasible.

Furthermore, Horvath et al. [2021a] and Bayer et al. [2019] discussed methods to minimize errors across various maturities and strikes by carefully designing the error function. These methods are useful for swaptions and the volatilities of foreign exchange rates among other products, as the calibration targets of these products can be seen as a multi-dimensional matrix of maturities and strikes.

5.2.2 Applications to various models

Various numerical experiments have been conducted with the aim of putting these algorithms into practice. In particular, in Horvath et al. [2021a], the calibration of the rough volatility model using the two-step method was discussed. For the setting of the numerical experiments in Horvath et al. [2021a], refer to Table 2.

Calibration problems for other models have also been discussed, including the Heston model in Dimitroff et al. [2018], the SABR model in Lokvancic [2020], the interest rate models in Sabbioni et al. [2021], and the general HJM model in Benth et al. [2020]. In addition, in Liu et al. [2019b], the generation of training data for deep calibration under the Black–Scholes and Heston models is discussed. In Brigo et al. [2021], the interpretability of deep calibration under the Heston model is addressed.

5.3 Practical advantages and challenges

Deep calibration offers several desirable features for practical operations. Firstly, it operates within a supervised learning framework. This is advantageous for enhancing and verifying the stability of learning because one can prepare as much training data and test data as needed by randomly generating the model parameters and associated model prices. Moreover, several studies reported that deep calibration can achieve satisfactory accuracy at a relatively modest computational expense. In addition, by validating the learned function, one can verify the attainable range of the calibration targets' prices and the behavior of the

[39] Note that research on learning model prices with neural networks began with Hutchinson et al. [1994] and has been extensively studied since then. In particular, Garcia and Gençay [2000] and Dugas et al. [2009] have presented methods for learning model prices using neural networks by considering important conditions in finance such as the absence of arbitrage as constraint conditions. Ruf and Wang [2020], a review paper, provides a summary of these studies.

greeks prior to the daily valuations, which can render the calibration more robust compared to existing optimization methods.[40]

It should be noted that it is crucial to regularly inspect the results obtained by deep calibration. In particular, if the market environment undergoes significant changes, the relationship structure between the model parameters and external factors may also change significantly, therefore changing the shape of the learned function. In such cases, prompt validation of the model or switching to existing optimization methods may be required.

6 Concluding remarks

This paper provides an overview of the trends in research related to deep hedging and deep calibration, which are applications of deep learning in the field of finance, from both practical and academic perspectives. Deep hedging applies deep learning to solve convex risk minimization problems, thereby making it possible to solve problems that were not wholly solvable analytically and to quantify the impact of market frictions such as transaction costs on derivative pricing and hedging strategies. On the other hand, deep calibration makes the calculations in calibration faster and more stable, which is a part of the derivative pricing process under the risk-neutral valuation framework. As such, deep learning has the potential to advance a variety of techniques in the field of finance over both the short and long term.

Beyond the applications to hedging and calibration which are discussed in this paper, there have been extensive discussions of other potential applications of deep learning in finance. Many problems in finance, including hedging, are formulated as *control problems*, which seek optimal actions under given constraints. The compatibility of control problems with reinforcement learning has inspired numerous reinforcement learning-based methods to solve problems such as portfolio optimization (Jiang et al. [2017]), optimal execution (Ning et al. [2021]), market-making (Spooner et al. [2018]), and algorithmic trading (Deng et al. [2016]). In addition, studies on predicting asset prices (Patel et al. [2015]) and generating financial time series data using Generative Adversarial Networks (Wiese et al. [2020]) have been progressing. There have also been studies on the application of deep learning to general computational problems, including applications to forward-backward stochastic differential equations (Weinan et al. [2017]) and optimal stopping problems (Becker et al. [2019]). The former study (application to forward-backward stochastic differential equations) has direct applications in finance, including the calculation of adjustments to credit valuation adjustment (CVA) and initial margin (Henry-Labordère [2020]), as well as xVA calculations (Gnoatto et al. [2020]). Several survey papers have discussed the application of machine learning and deep learning in finance, including Hambly et al. [2021], Ruf and Wang [2020], Emerson et al. [2019], Warin [2021], and Mashrur et al. [2020].

As both deep learning and finance are fast-developing research fields, it is crucially important to keep up with the most current cutting-edge discussions in these fields.

[40]For instance, if anomalies are detected in the process of calibration, it becomes easier to identify the cause.

References

Andersen, L. B. and Piterbarg, V. V., *Interest Rate Modeling 1: Foundations and Vanilla Models*, Atlantic Financial Press, 2010.

Barles, G. and Soner, H. M., "Option pricing with transaction costs and a nonlinear Black-Scholes equation", *Finance and Stochastics*, 2(4), pp.369–397, 1998.

Baxter, M., Rennie, A., and Rennie, A. J., *Financial calculus: an introduction to derivative pricing*, Cambridge university press, 1996.

Bayer, C., Horvath, B., Muguruza, A., Stemper, B., and Tomas, M., "On deep calibration of (rough) stochastic volatility models", *arXiv preprint arXiv:1908.08806*, 2019.

Becker, S., Cheridito, P., and Jentzen, A., "Deep optimal stopping", *The Journal of Machine Learning Research*, 20(1), pp.2712–2736, 2019.

Ben-Tal, A. and Teboulle, M., "An old-new concept of convex risk measures: the optimized certainty equivalent", *Mathematical Finance*, 17(3), pp.449–476, 2007.

Benth, F. E., Detering, N., and Lavagnini, S., "Accuracy of deep learning in calibrating HJM forward curves", *arXiv preprint arXiv:2006.01911*, 2020.

Björk, T., *Arbitrage theory in continuous time*, Oxford university press, 2009.

Black, F. and Scholes, M., "The pricing of options and corporate liabilities", *Journal of political economy*, 81(3), pp.637–654, 1973.

Brigo, D., Huang, X., Pallavicini, A., and Borde, H. S. d. O., "Interpretability in deep learning for finance: a case study for the Heston model", *arXiv preprint arXiv:2104.09476*, 2021.

Buehler, H., Gonon, L., Teichmann, J., and Wood, B., "Deep hedging", *Quantitative Finance*, 19(8), pp.1271–1291, 2019a.

Buehler, H., Gonon, L., Teichmann, J., Wood, B., Mohan, B., and Kochems, J., "Deep Hedging: Hedging Derivatives Under Generic Market Frictions Using Reinforcement Learning", *Risk Management & Analysis in Financial Institutions eJournal*, 2019b.

Buehler, H., Horvath, B., Lyons, T., Arribas, I. P., and Wood, B., "A data-driven market simulator for small data environments", *arXiv preprint arXiv:2006.14498*, 2020.

Buehler, H., Horvath, B., Lyons, T., Perez Arribas, I., and Wood, B., "Generating financial markets with signatures", *Risk.net cutting edge*, 2021a.

Buehler, H., Murray, P., Pakkanen, M., and Wood, B., "Deep hedging:learning to remove drift", *Risk.net, cutting edge*, 2022a.

Buehler, H., Murray, P., Pakkanen, M. S., and Wood, B., "Deep Hedging: Learning Risk-Neutral Implied Volatility Dynamics", *arXiv preprint arXiv:2103.11948*, 2021b.

Buehler, H., Murray, P., and Wood, B., "Deep Bellman Hedging", *arXiv preprint arXiv:2207.00932*, 2022b.

Cao, J., Chen, J., Hull, J., and Poulos, Z., "Deep hedging of derivatives using reinforcement learning", *The Journal of Financial Data Science*, 3(1), pp.10–27, 2021.

Carbonneau, A., "Deep hedging of long-term financial derivatives", *Insurance: Mathematics and Economics*, 99, pp.327–340, 2021.

Carbonneau, A. and Godin, F., "Equal risk pricing of derivatives with deep hedging", *Quantitative Finance*, 21(4), pp.593–608, 2021.

Cochrane, J. H., "Presidential address: Discount rates", *The Journal of finance*, 66(4), pp.1047–1108, 2011.

Davis, M. H., Panas, V. G., and Zariphopoulou, T., "European option pricing with transaction costs", *SIAM Journal on Control and Optimization*, 31(2), pp.470–493, 1993.

Deng, Y., Bao, F., Kong, Y., Ren, Z., and Dai, Q., "Deep direct reinforcement learning for financial signal representation and trading", *IEEE transactions on neural networks and learning systems*, 28(3), pp.653–664, 2016.

Dimitroff, G., Roeder, D., and Fries, C. P., "Volatility model calibration with convolutional neural networks", *Available at SSRN 3252432*, 2018.

Dugas, C., Bengio, Y., Bélisle, F., Nadeau, C., and Garcia, R., "Incorporating Functional Knowledge in Neural Networks.", *Journal of Machine Learning Research*, 10(6), 2009.

Emerson, S., Kennedy, R., O'Shea, L., and O'Brien, J., "Trends and applications of machine learning in quantitative finance", In *8th international conference on economics and finance research (ICEFR 2019)*, 2019.

Fernandez-Arjona, L. and Filipović, D., "A machine learning approach to portfolio pricing and risk management for high-dimensional problems", *Mathematical Finance*, 32(4), pp.982–1019, 2022.

Föllmer, H. and Kabanov, Y. M., "Optional decomposition and Lagrange multipliers", *Finance and Stochastics*, 2(1), pp.69–81, 1997.

Föllmer, H. and Leukert, P., "Efficient hedging: cost versus shortfall risk", *Finance and Stochastics*, 4(2), pp.117–146, 2000.

Föllmer, H. and Schied, A., "Convex measures of risk and trading constraints", *Finance and stochastics*, 6(4), pp.429–447, 2002.

Föllmer, H. and Schied, A., "Stochastic finance", In *Stochastic Finance*, de Gruyter, 2016.

Föllmer, H. and Schweizer, M., *Hedging of contingent claims under incomplete information*, volume 5, pp. 19–31, 1991.

Frankle, J. and Carbin, M., "The lottery ticket hypothesis: Finding sparse, trainable neural networks", *arXiv preprint arXiv:1803.03635*, 2018.

Garcia, R. and Gençay, R., "Pricing and hedging derivative securities with neural networks and a homogeneity hint", *Journal of Econometrics*, 94(1), pp.93–115, 2000.

Garipov, T., Izmailov, P., Podoprikhin, D., Vetrov, D. P., and Wilson, A. G., "Loss Surfaces, Mode Connectivity, and Fast Ensembling of DNNs", *Advances in neural information processing systems*, 31, 2018.

Gatheral, J., Jaisson, T., and Rosenbaum, M., "Volatility is rough", *Quantitative finance*, 18(6), pp.933–949, 2018.

Gnoatto, A., Picarelli, A., and Reisinger, C., "Deep xVA solver–A neural network based counterparty credit risk management framework", *arXiv preprint arXiv:2005.02633*, 2020.

Godin, F., "Minimizing CVaR in global dynamic hedging with transaction costs", *Quantitative Finance*, 16(3), pp.461–475, 2016.

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

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y., "Generative adversarial networks", *Communications of the ACM*, 63(11), pp.139–144, 2020.

Guo, I. and Zhu, S.-P., "Equal risk pricing under convex trading constraints", *Journal of Economic Dynamics and Control*, 76, pp.136–151, 2017.

Halperin, I., "The QLBS Q-Learner goes NuQLear: fitted Q iteration, inverse RL, and option portfolios", *Quantitative Finance*, 19(9), pp.1543–1553, 2019.

Hambly, B., Xu, R., and Yang, H., "Recent advances in reinforcement learning in finance", *arXiv preprint arXiv:2112.04553*, 2021.

Harrison, J. M. and Kreps, D. M., "Martingales and arbitrage in multiperiod securities markets", *Journal of Economic theory*, 20(3), pp.381–408, 1979.

Harrison, J. M. and Pliska, S. R., "Martingales and stochastic integrals in the theory of continuous trading", *Stochastic processes and their applications*, 11(3), pp.215–260, 1981.

Hastie, T., Montanari, A., Rosset, S., and Tibshirani, R. J., "Surprises in high-dimensional ridgeless least squares interpolation", *The Annals of Statistics*, 50(2), pp.949–986, 2022.

Henry-Labordère, P., "CVA and IM: welcome to the machine", *Risk.net, cutting edge*, 2020.

Hernandez, A., "Model calibration with neural networks", *Risk.net*, 2017.

Heston, S. L., "A closed-form solution for options with stochastic volatility with applications to bond and currency options", *The review of financial studies*, 6(2), pp.327–343, 1993.

Hinton, G. E., Osindero, S., and Teh, Y.-W., "A fast learning algorithm for deep belief nets", *Neural computation*, 18(7), pp.1527–1554, 2006.

Hodges, S. and Neuberger, A., "Optimal replication of contingent claims under transaction costs", *Review Futures Market*, 8, pp.222–239, 1989.

Horvath, B., Muguruza, A., and Tomas, M., "Deep learning volatility: a deep neural network perspective on pricing and calibration in (rough) volatility models", *Quantitative Finance*, 21(1), pp.11–27, 2021a.

Horvath, B., Teichmann, J., and Žurič, Ž., "Deep hedging under rough volatility", *Risks*, 9(7), pp.138, 2021b.

Hull, J. C., *Options, Futures, and other Derivatives (9th edition)*, Pearson prentice hall, 2014.

Hutchinson, J. M., Lo, A. W., and Poggio, T., "A nonparametric approach to pricing and hedging derivative securities via learning networks", *The journal of Finance*, 49(3), pp.851–889, 1994.

Ilhan, A., Jonsson, M., and Sircar, R., "Optimal static-dynamic hedges for exotic options under convex risk measures", *Stochastic Processes and their Applications*, 119(10), pp.3608–3632, 2009.

Imaki, S., Imajo, K., Ito, K., Minami, K., and Nakagawa, K., "No-transaction band network: A neural network architecture for efficient deep hedging", *Available at SSRN 3797564*, 2021.

Itkin, A., "Deep learning calibration of option pricing models: some pitfalls and solutions", *Risk.net*, 2020.

Jiang, Z., Xu, D., and Liang, J., "A deep reinforcement learning framework for the financial portfolio management problem", *arXiv preprint arXiv:1706.10059*, 2017.

Kawaguchi, K., "Deep learning without poor local minima", *Advances in neural information processing systems*, 29, 2016.

Kingma, D. P. and Ba, J., "Adam: A Method for Stochastic Optimization", In *ICLR*, 2015.

Kolm, P. N. and Ritter, G., "Dynamic replication and hedging: A reinforcement learning approach", *The Journal of Financial Data Science*, 1(1), pp.159–171, 2019.

Krabichler, T. and Teichmann, J., "Deep Replication of a Runoff Portfolio", 2020.

Kramkov, D. O., "Optional decomposition of supermartingales and hedging contingent claims in incomplete security markets", *Probability theory and related fields*, 105(4), pp.459–479, 1996.

Krizhevsky, A., Sutskever, I., and Hinton, G. E., "Imagenet classification with deep convolutional neural networks", *Communications of the ACM*, 60(6), pp.84–90, 2017.

Leland, H. E., "Option pricing and replication with transactions costs", *The journal of finance*, 40(5), pp.1283–1301, 1985.

Lintner, J., "The Valuation of Risk Assets and the Selection of Risky Investments in Stock Portfolios and Capital Budgets", *The Review of Economics and Statistics*, pp. 13–37, 1965.

Liu, S., Borovykh, A., Grzelak, L. A., and Oosterlee, C. W., "A neural network-based framework for financial model calibration", *Journal of Mathematics in Industry*, 9(1), pp.1–28, 2019a.

Liu, S., Oosterlee, C. W., and Bohte, S. M., "Pricing options and computing implied volatilities using neural networks", *Risks*, 7(1), pp.16, 2019b.

Lokvancic, M., "Machine learning SABR model of stochastic volatility with lookup table", *Available at SSRN 3589367*, 2020.

Lütkebohmert, E., Schmidt, T., and Sester, J., "Robust deep hedging", *Available at SSRN 3869616*, 2021.

Markowitz, H., "Portfolio Selection", *Journal of Finance*, 7(1), pp.77–91, 1952.

Martin, I., "What is the Expected Return on the Market?*", *The Quarterly Journal of Economics*, 132(1), pp.367–433, 2016.

Mashrur, A., Luo, W., Zaidi, N. A., and Robles-Kelly, A., "Machine Learning for Financial Risk Management: A Survey", *IEEE Access*, 8, pp.203203–203223, 2020.

Merton, R. C., "Theory of Rational Option Pricing", *The Bell Journal of Economics and Management Science*, 4(1), pp.141–183, 1973.

Mikkilä, O. and Kanniainen, J., "Empirical Deep Hedging", *Derivatives eJournal*, 2021.

Minsky, M. and Papert, S., "An introduction to computational geometry", *Cambridge tiass., HIT*, 479, pp.480, 1969.

Miyahara, Y., "Model and Related Estimation Problems", *Asia-Pacific Financial Markets*, 8, pp.45–60, 2001.

Mossin, J., "Equilibrium in a capital asset market", *Econometrica: Journal of the econometric society*, pp. 768–783, 1966.

Muhle-Karbe, J., Reppen, M., and Soner, H. M., "A primer on portfolio choice with small transaction costs", *Annual Review of Financial Economics*, 9, pp.301–331, 2017.

Ning, B., Lin, F. H. T., and Jaimungal, S., "Double deep q-learning for optimal execution", *Applied Mathematical Finance*, 28(4), pp.361–380, 2021.

Patel, J., Shah, S., Thakkar, P., and Kotecha, K., "Predicting stock market index using fusion of machine learning techniques", *Expert Systems with Applications*, 42(4), pp.2162–2172, 2015.

Risk.net, "Quant of the year: Hans Buehler", *Risk.net*, 2022.

Rockafellar, R. T., Uryasev, S., et al., "Optimization of conditional value-at-risk", *Journal of risk*, 2, pp.21–42, 2000.

Rogers, L., *Why is the effect of proportional transaction costs $O(\delta^{2/3})$?*, pp. 303–308, American Mathematical Society, 2004.

Rosenblatt, F., "The perceptron: a probabilistic model for information storage and organization in the brain.", *Psychological review*, 65(6), pp.386, 1958.

Ruf, J. and Wang, W., "Neural networks for option pricing and hedging: a literature review", *Journal of Computational Finance*, 2020.

Rumelhart, D. E., McClelland, J. L., Group, P. R., et al., *Parallel distributed processing*, volume 1, IEEE New York, 1988.

Sabbioni, L., Restelli, M., and Prampolini, A., "Fast Direct Calibration of Interest Rate Derivatives Pricing Models", In *Proceedings of the First ACM International Conference on AI in Finance*, ICAIF '20 New York, NY, USA: Association for Computing Machinery, 2021.

Sharpe, W. F., "Capital Asset Prices: A Theory Of Market Equilibrium Under Conditions Of Risk", *Journal of Finance*, 19(3), pp.425–442, 1964.

Shi, X., Xu, D., and Zhang, Z., "Deep Learning Algorithms for Hedging with Frictions", *arXiv preprint arXiv:2111.01931*, 2021.

Shimizu, T., "Dynamic macro stress testing with feedback effects (in Japanese)", *IMES Discussion Paper Serie No.1998-J-1*, 1998.

Shreve, S. E., "Liquidity premium for capital asset pricing with transaction costs", *Institute for Mathematics and Its Applications*, 65, pp.117, 1995.

Shreve, S. E. et al., *Stochastic calculus for finance II: Continuous-time models*, volume 11, Springer, 2004.

Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., Van Den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., et al., "Mastering the game of Go with deep neural networks and tree search", *nature*, 529(7587), pp.484–489, 2016.

Son, G. and Kim, J., "Neural Networks for Delta Hedging", *arXiv preprint arXiv:2112.10084*, 2021.

Soudry, D., Hoffer, E., Nacson, M. S., Gunasekar, S., and Srebro, N., "The implicit bias of gradient descent on separable data", *The Journal of Machine Learning Research*, 19(1), pp.2822–2878, 2018.

Spooner, T., Fearnley, J., Savani, R., and Koukorinis, A., "Market Making via Reinforcement Learning", 2018.

Suzuki, T., "Adaptivity of deep ReLU network for learning in Besov and mixed smooth Besov spaces: optimal rate and curse of dimensionality", In *International Conference on Learning Representations*, 2018.

Tawada, Y. and Sugimura, T., "Machine learning hedge strategy with deep Gaussian process regression", *Risk.net, cutting edge*, 2020.

Warin, X., "Deep learning for efficient frontier calculation in finance", *arXiv preprint arXiv:2101.02044*, 2021.

Weinan, E., Han, J., and Jentzen, A., "Deep Learning-Based Numerical Methods for High-Dimensional Parabolic Partial Differential Equations and Backward Stochastic Differential Equations", *Communications in Mathematics and Statistics*, 4(5), pp.349–380, 2017.

Whalley, A. E. and Wilmott, P., "An asymptotic analysis of an optimal hedging model for option pricing with transaction costs", *Mathematical Finance*, 7(3), pp.307–324, 1997.

Wiese, M., Bai, L., Wood, B., and Buehler, H., "Deep hedging: learning to simulate equity option markets", *Available at SSRN 3470756*, 2019.

Wiese, M., Knobloch, R., Korn, R., and Kretschmer, P., "Quant gans: Deep generation of financial time series", *Quantitative Finance*, 20(9), pp.1419–1440, 2020.

Xu, M., "Risk measure pricing and hedging in incomplete markets", *Annals of Finance*, 2(1), pp.51–71, 2006.

Zhang, J. and Huang, W., "Option hedging using LSTM-RNN: an empirical analysis", *Quantitative Finance*, 21(10), pp.1753–1772, 2021.